BLENDER SECRETS

EVERYTHING YOU NEED FOR YOUR BLENDER JOURNEY

by Jan van den Hemel (updated Sept 13th 2024)

Why this e-book?

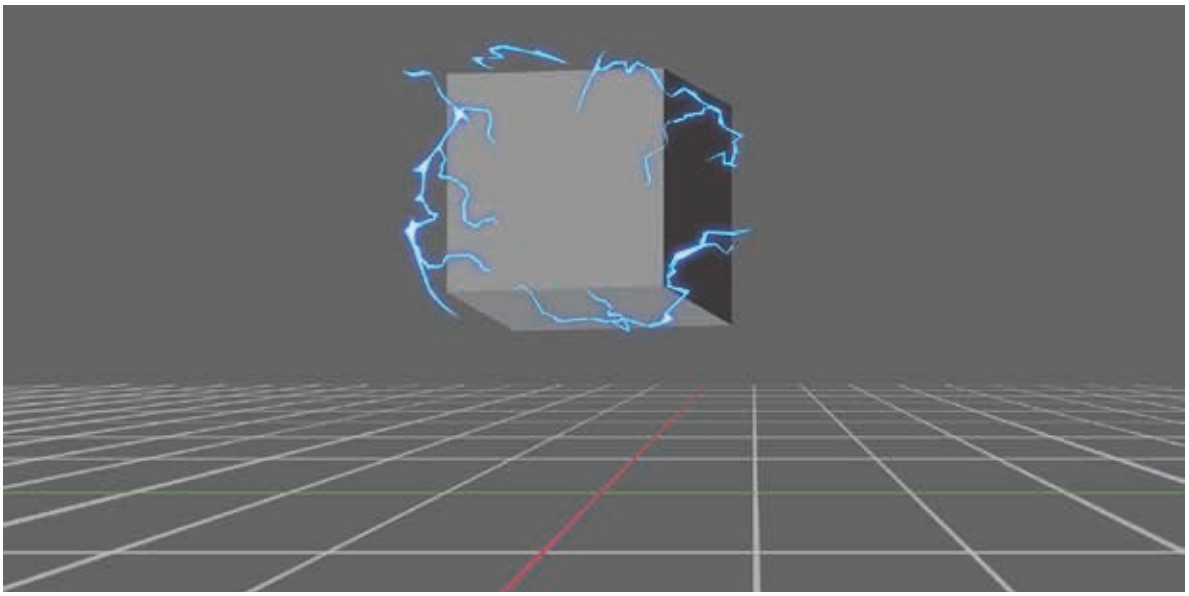
Hi! Allow me to introduce myself. **My name is Jan** and you may know me from my "Daily Blender Tips". The reason I started making these tips is because even as a professional Blender user, I still sometimes need to look stuff up ("what was that shortcut again?") and could not find a good source of really short tutorials.

For things you only do once in a while (like baking normal maps), even the professionals will sometimes go to Youtube to patiently endure a 40-minute tutorial where the actual explanation comes at the 37-minute mark. Wouldn't it be great if these topics were addressed in under 1 minute?

I couldn't find any channel with short, one-minute video tutorials for Blender, so I decided to make them myself. Of course, not everything can be shown in 1 minute, but a surprising amount of things can! This e-book collects these Blender tips in written form, so you can learn them at your own pace.

I hope it will serve you well and that you will enjoy reading it.

Now let's start and see on what journey we can take this default cube! 🔥



Best wishes,
Jan

TABLE OF CONTENTS

MODELING BASICS 18

Loop Cut and Slide Tips 19
LoopTools - Curve and GStretch 22
LoopTools - Bridge, Loft and Circle 27
LoopTools - Relax, Space and Flatten 33
How to use Shade Auto Smooth in Blender 4.3 36
Create a hole from a Vertex 39
Perfect Holes with Quad Topology in Curved Surfaces 46
Image file to geometry using Knife Project 52
More hole-creating methods 63
Support Loops 71
Select Connected Faces 73
Inset individual faces 73
Make selections circular with To Sphere 74
Concave Bevel Profile 76
Make connected holes and smooth surfaces with Bridge 77
Make a coffee mug handle - Extrude to Cursor 79
Modeling using Spin 81
Shrink / Fatten 84
Convert Triangles to Quads 86
Move Edge Loops Without Destroying Your UV Layout 87
Scale Tubes and Pipes Proportionally 87
Knife Project 88
Offset Edge Slide 90
Flatten Faces 91
Collapse Edges 92
Crease Edges 93
Symmetrize 94
Angle Constraint 95
Randomize Transform 96
Sphere With Quads 97
Slide Vertices Along Edges 98
Sliding Vertices to where there's no edge 98
Extrude faces along Normals 99
Using Empties For Transformations And Mirroring 100
Import Reference Images For Modeling 102
Select Rows Of Faces Easily And Quickly 103
Separate a part of the mesh 103
Moving edge loops to center 104
F2 Add-on 105
Easily fix Edge Loops 106
Rip Fill 107
Dissolve Edges 108
Bisect tool 109

Click here to return to the index.

Merge vertices by distance . . . 110
Mirror without a mirror modifier . . . 111
Smooth Shortcuts and Round Cubes . . . 112
Selecting faces by size . . . 114
Boundary Protection - 6 ways to add support loops . . . 115
3 Scaling Tips you may not know! . . . 118
3 inset modeling tips . . . 120
5 Reasons your Bevel may not be working . . . 122
Bevel Tips (Poles, Miter Outer, Shape) . . . 125
5 ways to add Vertices . . . 129
Select Linked - Every Option Explained . . . 132
The Scale Cage . . . 135
3 ways to Flatten Faces . . . 138
3 ways to Smoothen areas . . . 141
Clever way to add Support Loops . . . 144
3 ways to Straighten Edge Loops . . . 145
Edge Flow add-on . . . 147
2 ways to Dissolve Edges . . . 148
Connected Proportional Editing . . . 149
The Weld Modifier . . . 150
How to use Creasing . . . 152
Interactive Primitives . . . 154
Modeling with Primitives . . . 156
Car Modeling Basics part 1 - Reference . . . 160
Car Modeling Basics part 2 - Mirror modifier . . . 163
Car Modeling Basics part 3 . . . 165
Snapping with Base Points in Blender 4 . . . 169

MODELING . . . 174

Cylinder with Holes / Barrel . . . 175
Camera Projection modeling part 1 - fSpy to Blender . . . 180
Camera Projection modeling part 2 - Create an Environment . . . 188
Modeling from Photos with the Knife Tool - Basics . . . 194
Modeling from Photos with the Knife Tool - Perspective . . . 197
Modeling from Photos - Why is my texture distorted? . . . 200
Super Quick Circular Array . . . 202
How to use Edge Slide . . . 203
Triangular Fill method . . . 205
3D Scanning with Meshroom and Blender . . . 206
Custom Transform Orientation . . . 213
Twisted Bridge With LoopTools Add-On . . . 215
Linked Duplicates . . . 217
Clipping Border . . . 218
Inset Boundary . . . 218
Limited Dissolve (Make a Sci-fiSphere) . . . 219
Cast Modifier (Or How To Make Another Weird Sphere) . . . 221
Tissue Add-on: Experiment 1 . . . 223

Click here to return to the index.

Tissue Add-on: Experiment 2 . . . 226
How to use Nasa's free models and textures . . . 230
Quad Topology Tips . . . 235
Circular hole in a sphere . . . 242
Unsubdivide and poke faces . . . 243
Making a microphone mesh with modifiers . . . 245
Circular Array of Cubes . . . 247
Modeling Propeller Blades . . . 252
Array along a path . . . 255
Discombobulator . . . 256
Align Car Rim Parts . . . 257
Filling many holes at once . . . 260
Making welds in two ways . . . 261
Making a screw the easy way . . . 262
Math Mesh add-on . . . 264
Modeling with math . . . 265
Hide vertices from editing in Edit Mode . . . 266
Balloon Text . . . 267
Make balloon objects with Cloth pressure . . . 268
Two ways to unsubdivide . . . 271
Add an Edge Loop on triangles . . . 273
Make a pipe with cloth wrinkles . . . 274
Metaball Array . . . 278
Group instances . . . 280
Text on a Curve . . . 281
Modeling with Hard Surface Normal Stamps . . . 286
Create your own Hard Surface Normal Stamps . . . 288
Modeling holes with Quad topology . . . 291
Perfect topology on Curved surfaces . . . 294
Do you know all 5 of these Bevel methods? . . . 297
Turn N-gon pipes into Quad pipes in a couple of clicks! . . . 300
Easily Extrude Thick Panels . . . 302
Grease Pencil Strokes to Mesh . . . 305
Wrapping hard surface meshes on surfaces . . . 309
Modeling Circular Hard Surface Detail . . . 312
Modeling Sci-fi Greebles on a Sphere . . . 317
Complex Mode for Solidify Modifier . . . 321
Easy Non-Destructive Panel Cuts . . . 323
Custom Fonts in Blender . . . 326
Non-destructive bridge . . . 328
Surface patch from Curves . . . 331
Model an airplane wing . . . 334
Beveling and Tapering Curves . . . 338
Filling complex N-gon shapes like leaves . . . 343
Using the remesh modifier on a flat plane . . . 347
Convert Normal Map to Geometry (for 3D printing) . . . 349
The default cubes strike back! (easy tank tracks) . . . 352

Click here to return to the index.

Circular Array using Screw modifier . . . 355
Circular Array using Instancing . . . 358
Fast Circular Array . . . 361
Circular Array using Empty . . . 362
Auto Weld Vertices . . . 365
Make an arch using the Warp tool . . . 366
Model shapes with Screw modifier . . . 368
Find the triangles in a dense mesh . . . 370
Kitbashing details . . . 371
Easy hard surface details using Floaters . . . 374
Baking Floaters to Normal maps . . . 377
Array of Floaters . . . 380
Basics of Nurbs Surfaces . . . 388

BOOLEANS . . . 395
3 ways to slice objects . . . 396
The Fastest Boolean Workflow . . . 402
Fixing Boolean Issues . . . 408
Modeling a scifigun with Carver . . . 410
Modeling Broken Pillars . . . 416
Slice Cuts . . . 426
Boolean Basics: Non-Destructive Workflow . . . 431
How To Use the Bool Tool Add-on . . . 437

MODELING CLOTHING . . . 440
Cloth Simulation Sewing basics . . . 441
Posed Cloth Simulation for Sculpting . . . 445
Mixing Armatures and Simulations for Clothing . . . 448
Detailed Clothes simulated Easily . . . 452
Model a Simple Dress . . . 455
Modeling Stitches . . . 457
Modeling clothes . . . 461
Model a Skirt using Select Next Active . . . 465
Interactively adjust cloth sims . . . 467
Seams to Sewing Pattern . . . 470

ARCHITECTURE MODELING . . . 475
Create Decorative Edges on sofas and cushions . . . 476
How to make Custom Bevel Profiles . . . 480
Modeling a designer door handle with Knurl Pattern . . . 483
Modeling a window . . . 488
Chair modeling: Wireframe backrest . . . 498
Extruding complex shapes . . . 502
Align three furniture legs . . . 504
Easy Pillow using the Cloth modifier . . . 507
How to make diamond tufting patterns . . . 511
Modeling folds in cloth . . . 517
How To Make A Curtain In One Minute . . . 520

Click here to return to the index.

Make a curtain using Shape Keys and a Cloth simulation . . . 527
Apply Shape keys and Cloth sim . . . 531
8 Tips for modeling Decorative Elements . . . 533
Model a Spiral Staircase . . . 538
Model a basket using modifiers . . . 544
Medieval Tower with Wall Factory . . . 547
Extrude at 90° angles using Shear . . . 552
Drawing tapered ornamental details on surfaces . . . 554
Ornament Modeling with Custom Curve Bevels . . . 558

ENVIRONMENTS . . . 560
Draw in 3D with Grease Pencil and Convert to Mesh . . . 561
Draw in 3D with Grease Pencil - Part 2 - Sculpting . . . 564
Easily Create Textured Mountains . . . 569
Create the Earth's Moon . . . 573
Creating Clouds using Mesh to Volume . . . 582
G Scatter - Free Geometry Nodes-based scatter add-on . . . 588
Scatter Add-on . . . 591
Another Noise Tool (A.N.T. Landscape) . . . 594
Node-based tree modeling . . . 595
MTree node-based tree add-on Materials and presets . . . 601
Easy Tree Wind Animation with free Mtree add-on . . . 606
Easy tree and leaves wind animation with Sapling . . . 609
Make a forest . . . 612
Custom Tree or Plant part 1 - The Trunk . . . 616
Custom Tree or Plant part 2 - The Twigs . . . 619
Custom Tree or Plant part 3 - The Leaves . . . 625
Mixing 3D scanned and node-based Trees . . . 630
Mixing photogrammetry and hand-made trunks . . . 637
Selecting lots of tiny things . . . 640
Add ivy with IvyGen add-on . . . 642
Billboard with Normal Maps . . . 644
Create a mysterious city with Blender and fSpy . . . 652

LOW POLY MODELING . . . 681
Low-poly tree in one minute . . . 682
Even lower poly tree . . . 686
Low poly typography . . . 688
When to rotate edges . . . 691

CABLES, PIPES and ROPES . . . 692
Drawing cables on surfaces . . . 693
How To Make Pipes . . . 694
Tubetool add-on . . . 697
Pipe Nightmare add-on . . . 699
Detailed Cylinder Using Modifiers . . . 703
Easy Textured Pipes . . . 706
Start and End Caps . . . 709

Click here to return to the index.

How to add multiple cables at once . . . 711
Hanging cables (Catenary Curves) . . . 714
Model a Rope along a Curve . . . 717
Create Corrugated Pipes . . . 722
Cable Physics . . . 726
Cables with collision . . . 729
Grappling hook and rope physics . . . 730
Adding Cables with Physics that are attached to moving objects . . . 733
Textured Cable with Physics connected to objects . . . 736
Complex Rope Mesh Physics with the Surface Deform modifier . . . 739

SCULPTING . . . 744

Improve Sculpting Performance Instantly . . . 745
Create a Sculpting base mesh with the Skinify add-on . . . 747
Create a Sculpting base mesh with Metaballs . . . 752
Sculpting Adding Detail . . . 755
Sculpting with Textures and Stencils . . . 757
Creating a Robot base mesh with Voxel remesh . . . 761
Sculpting a robot concept . . . 765
Face Sets . . . 770
Hard Surface Sculpting: Refining the rough sculpt . . . 774
Hard Surface Sculpting: Masking and Mesh Filter . . . 778
Hard Surface Sculpting: Using Curves . . . 781
Hard Surface Sculpting: Dyntopo . . . 783
Sculpting Sword Details Using Alpha Textures . . . 786
Creating Alphas for Sculpting . . . 791
Sculpting details using Masking . . . 793
Placing Hard Surface Details in Sculpt Mode . . . 796
Radial Symmetry in Sculpt Mode . . . 800
Using the Layer Brush . . . 803
Extruding in Sculpt Mode . . . 809
Inflate Mesh Filter . . . 816
Sculpting Tentacles with Snake Hook and Curve Stroke Method . . . 821
Seamless Tiling Sculpting . . . 827
Creating a robot helmet in Sculpt Mode . . . 833
Zbrush-like Morph Target in Blender . . . 840
Mask Extract . . . 843
Line Project . . . 852
Voxel and Quad Remesh in Sculpt Mode . . . 858
Pose Brush . . . 861
Sculpting Auto Masking with Cavity . . . 863
How to create VDM Brushes in Blender 4 . . . 868
Sculpting with Vector Displacement Maps - the VDM brush . . . 874
Baking Vector Displacement Maps . . . 876
Slicing Objects in Sculpt Mode . . . 881
Slicing Objects using Face Sets . . . 884
How to make stylized ripples . . . 888

Click here to return to the index.

Move the Pivot Point in Sculpt Mode . . . 897

SCULPTING CLOTHING . . . 899

Sculpt clothing folds with the Cloth Brush . . . 900
Cloth Brush Tips and Tricks . . . 902
8 Types of Cloth Brush Deformation . . . 906
Cloth Sculpting with Masks, Face Sets and the Cloth Filter . . . 911
Making a Padded Jacket using the Expand feature . . . 915
Create a Stitches sculpting brush in Blender . . . 921
How to create Ruffled Skirts . . . 926
Marvelous Designer-like Cloth pulling / grabbing . . . 934

RETOPOLOGY . . . 941

Solving Edge Flow problems . . . 942
Retopology tips - Settings . . . 945
Retopology tips - Creating geometry and the F2 add-on . . . 949
Retopology tips - LoopTools and Zbrush-like Smoothing . . . 951
Retopology tips - The magic of Grid Fill . . . 953
Retopology tips - How to retopo . . . 955
Bevel and Crease Hard Surface Panels . . . 957
Quick topology fix for ugly edges . . . 960
Retopology Visibility Tip: Draw Xray . . . 962
Fixing Shrinkwrap Problems . . . 966
Copy / Paste Geometry for Faster Modeling . . . 969
BSurfaces - Drawing Retopology with Annotation . . . 972
Use BSurfaces with Grease Pencil to sketch geometry . . . 976
Easy trick to turn N-gons and Triangles into Quads . . . 982
The Poly Build Tool . . . 986
Reconstruct Lower Subdivisions from a 3D scan . . . 988

UV UNWRAPPING . . . 990

Tips for Marking Seams . . . 991
Unwrap like a pro using Live Unwrap . . . 994
UV Unwrapping hard surface panels . . . 997
UV Unwrapping Transfer and pack UV's . . . 1000
Solving 3 common texture & UV issues . . . 1003
Stitch UV's . . . 1007
Unwrap a Cup - Follow Active Quads . . . 1009
Exporting the UV Layout . . . 1013
Project from View - avoiding stretched UV's . . . 1017
Easy Straight Unwrapping . . . 1019
Mirror UV-maps . . . 1022
Combine UVs from Multiple Objects . . . 1023
UDIM - Monster-sized textures . . . 1024
UV Unwrapping Cylindrical Objects like a Cyber Donut . . . 1031
Easily mark seams on complex shapes . . . 1037

RIGGING . . . 1039

Easy rigging of clothes in 30 seconds . . . 1040

Click here to return to the index.

Ragdoll Rigging part 1 - The easy method 1043
Ragdoll Rigging part 2 - More Realistic Joints 1047
Ragdoll Rigging part 3 - Connecting it to an Armature 1051
Install and use Rigid Bodys Generator add-on 1054
Automatic Muscle Deformations 1057
Rigging a simple character with Rigify 1062
Auto-Rigging with Mixamo 1072
Rigify FK to IK Snap 1074
Rigging a paper plane 1077
Adding Props To Your Character (like a stylish hat) 1089
Rigging With Separate Objects Or Not? 1092
Bendy Bones 1094
Bendy Bones part 2 1097
Better rigging with extra edge loops 1102
Action Bone constraint 1103
Auto IK (Inverse Kinematics) 1106
Manually Rigging Legs 1107
Adding extra bones 1111
Manually fixing weight paint issues 1114
Blending actions 1117
Constraints and Physics 1118
Constraint transfer 1121
The nuts and bolts of Transformation Constraints 1123
Robot Rigging (Part 1 - The Armature) 1127
Robot Rigging Part 2 Parenting to Bones 1130
Robot Rigging Part 3 Constraints 1134
Adding Objects to a rigged Character 1136
How to have a character interact with Physics Simulations 1138
How to have a character interact with an object 1141
How to change any pose to T-pose for retargeting 1144
Easy Piston Rigging 1147
Adding and rigging a tail for your character 1152

TEXTURING **1156**

Manually Paint Edge Wear 1157
Texture Painting with Quick Edit and Krita 1162
Texture Clone Painting 1165
Texture Clone Painting - Clone from one object to another 1167
Texture Painting Basics 1170
Texture Painting with Custom Brushes 1172
Texture Painting with Gradients 1176
Adding Bump Height to Texture Masks 1179
Texturing a head 1185
Make Seamless Textures Easily In Gimp 1196
My Painting Workflow In Blender 1199
How to use Leaf textures 1202
Texture Painting with Stencils 1213

Texture mixing . . . 1218
Rust Texture that works on any object in Eevee . . . 1220
Texturing a cyberpunk computer with photos . . . 1228
Fixing missing textures . . . 1238
One-click texturing with Dream UV . . . 1239
Turn a Photograph into a Tiling Texture . . . 1245
Create an Albedo texture in Blender . . . 1252
Seamless textures in one click with Anti-Seam . . . 1255
Texture Ripper . . . 1258

MATERIALS . . . 1260

Blending objects using Rust with the Ambient Occlusion node . . . 1261
Mixing Shader Masks with Displacement Maps . . . 1265
Micro-displacement . . . 1272
Higher realism through imperfections . . . 1278
Procedural bump maps . . . 1284
Texturing Without UV-Unwrapping . . . 1289
Drag And Drop Materials . . . 1292
Using Shrink Wrap To Project A Sticker To A Surface . . . 1294
Glass Transparency With Alpha Channel . . . 1296
Double-sided material . . . 1300
Add a decal using only nodes . . . 1302
Glass in Eevee . . . 1304
Quickly Add Material Nodes With Node Wrangler Add-On . . . 1306
Easy PBR Textures With Node Wrangler . . . 1308
Node Presets . . . 1311
Material Basics: Pointiness . . . 1313
Material Basics: Snow . . . 1318
How to use normal maps . . . 1323
Bump maps or Normal maps? . . . 1325
Random colors with the Object Info node . . . 1327
Easy wireframe shader . . . 1330
Select materials from characters . . . 1332
Using Empties to control textures . . . 1333
Single-sided Emission shader . . . 1334
Emission shader in Eevee . . . 1335
Rotate box-mapped textures . . . 1337
Using the Separate RGB Node . . . 1339
Substance Material Add-on for Blender . . . 1342
How to bake Bevel and Occlusion maps . . . 1347
Blur Node . . . 1353
Substance Painter-like Edge Wear Effects . . . 1357
Dirty Metal material . . . 1361
Append free materials . . . 1363
Perfect Displacement maps with low poly models . . . 1365
Low-poly hair cards . . . 1373
Procedural Leather Material . . . 1377

Click here to return to the index.

Wireframe Effect using modifiers . . . 1380
Random colors using nodes . . . 1383
The Reroute Node . . . 1386
How to avoid Texture stretching . . . 1387

BAKING . . . 1390
Baking Normal Maps from the Multires Modifier . . . 1391
Baking Displacement maps . . . 1393
Bake 3D scan textures to Low Poly meshes . . . 1397
Baking normal maps . . . 1399
Baking textures with the free TexTools add-on . . . 1402
Make a Normal Map using a Matcap . . . 1407
Baking with a Cage . . . 1411
Baking Displacement from Camera View . . . 1414

CHARACTERS . . . 1418
Modeling from Photos - Face Builder . . . 1419
Easy Anime Hair . . . 1423
Create characters in just a few clicks in Blender . . . 1426
How to Model Stylized Hair . . . 1431

GREASE PENCIL . . . 1436
Grease Pencil Basics . . . 1437
Grease Pencil for Concept Art . . . 1443
2D Animation Quick Start . . . 1452
2D Animation From 3D Animation . . . 1456
Grease Pencil Build Modifier . . . 1460
Grease Pencil Sculpt Stroke Thickness . . . 1461
Curve Tool in Grease Pencil . . . 1462
Grease Pencil Stroke Interpolation . . . 1465
Grease pencil guides . . . 1467
Grease pencil cutter tool . . . 1469
Installing textured brushes . . . 1470
Creating custom textured brushes . . . 1472
Draw Grease Pencil On Surfaces without offset distance issues . . . 1474
How to use the Grease Pencil Drawing Plane Canvas option . . . 1478

MODIFIERS . . . 1482
Array modifier with object offset . . . 1483
Corrective smooth modifier . . . 1484
Toggle Subdivision Levels . . . 1485
Apply All Modifiers At Once . . . 1486
Add hooks and Laplacian Deformer . . . 1487
Lattice Deform Modifier . . . 1490
Weird stuff with the Build Modifier . . . 1492
Mesh Deform modifier . . . 1495
Weighted Normals modifier . . . 1497
Editing Normals: Data Transfer . . . 1499
Smooth Edges with Average Normals . . . 1501

Click here to return to the index.

Smooth Edges with harden normals . . . 1502
UV Project modifier . . . 1503

ANIMATION . . . 1505
3 ways to Bounce a Ball . . . 1506
Animated Flag . . . 1509
Slow-motion Cloth Simulation . . . 1512
Copy Animation From One Object To Another . . . 1514
Perpetual Controlled Speed Tire Rotation . . . 1515
Easily Animate Objects With Lattices . . . 1520
Animate A Fish Without Rigging . . . 1523
Using Empty To Animate Displacement Modifier In A Loop . . . 1528
Dupliverts For Motion Graphics . . . 1532
Shape Keys: Join As Shapes . . . 1534
How To Launch A Cube . . . 1536
Copy animation and Edit separately . . . 1540
Easy animated leaves . . . 1542
Camera rotation without keyframes . . . 1545
Pulsating light without keyframes . . . 1547
Wiggle Expression . . . 1548
Liquid Colorful Text Reveal Motion Graphics . . . 1549
Shoelace animation . . . 1553
Using Sound to animate objects . . . 1556
How to control the speed of image sequences . . . 1558

CHARACTER ANIMATION . . . 1561
Importing Mixamo character animation in Blender . . . 1562
Mixamo walk cycle and how to create an Action from keyframes . . . 1563
Combining Mixamo Animations . . . 1565
Adjusting Mixamo animation in Blender . . . 1568
Cleaning up mocap and adding an adjustment layer . . . 1570
Retargeting with the free Rokoko add-on . . . 1574
Two-handed grip animation using Inverse Kinematics . . . 1577
Shape Keys: Basics . . . 1581
Simple Character Walk Cycle . . . 1584
Character Walking Along a Path . . . 1590
How To Have Characters Interact With Physics Simulations . . . 1594
Shape keys for facial expressions . . . 1599
Facial Expressions . . . 1602
Walk cycle markers . . . 1604
Motion Paths . . . 1606
Rotating your character . . . 1607

NATURE ANIMATION . . . 1609
Flowing water with Flow Maps . . . 1610
Animating clouds with Flow Maps . . . 1612
Moving Cloud Shadows . . . 1616
Animated growing Plant . . . 1620

Click here to return to the index.

Grass moving in the wind . . . 1628
Asteroid field . . . 1639

SIMULATIONS, HAIR AND PARTICLES . . . 1643
OpenVDB Quickstart . . . 1644
Vacuum Packing Objects using Cloth Pressure . . . 1648
Boat Floating on Ocean Waves . . . 1652
Rain with Dynamic Paint . . . 1657
Seamlessly Looping Ocean . . . 1665
Particles Follow A Path . . . 1673
Quick Hair / Fur and How to Comb and Weight Paint . . . 1677
2 Types Of Quick Fluids . . . 1683
Quickly Fracture Stuff . . . 1687
More Fracture Stuff! . . . 1692
Slow motion particles . . . 1695
How to set the centre of weight in a rigid body . . . 1698
Volumetric Cloud . . . 1699
Fluid Simulations part 1 (basics) . . . 1701
Fluid Simulations part 2 (finetuning) . . . 1704
Connecting things for simulations . . . 1707
Heroic Default Cube . . . 1708
Reveal a Photo with a Physics sim . . . 1714
Fun with Planks (Rigid Bodies) . . . 1717
Cell Fracture Explosion . . . 1720
Cell Fracture Inner Material . . . 1723
Annotate Cell Fracture . . . 1724
More realistic destruction and collision detection . . . 1726
Exploding Default Cube . . . 1729
Distribute Weight of a Rigid Body . . . 1733
Hide cracks pre-fracture . . . 1736
Cell Fracture Basics . . . 1740
Quick Mantaflow Explosion . . . 1745
How to make chains . . . 1747
Ocean Modifier basics . . . 1750
Ocean shader with Foam . . . 1754
Make a Dancing Hair Monster with Hair Dynamics . . . 1757
Quick Liquid with Spray, Foam and Bubbles . . . 1759
Particles along a curve with random colors in Eevee . . . 1762
Liquid stream following a curve . . . 1766

LIGHTING . . . 1769
Sunset Timelapse with Sky Texture . . . 1777
Volumetric background to HDRI transitions . . . 1782
How to use IES lights . . . 1785

CAMERA TIPS . . . 1788
Different cameras per viewport & adjusting lights . . . 1789
3 Camera Tips . . . 1790
Easily make smooth camera moves . . . 1791

Click here to return to the index.

Camera Shift 1794
Camera track to constraint 1795
Camera turnaround 1796
Camera switching while rendering using markers 1797
Camera crane rig add-on 1798
Set Viewport to Camera View 1799
Fly Mode 1800

RENDER TIPS 1801
Faster Render Times 1802
Added realism with Textured Lights 1805
More Realistic Renders in Cycles and Eevee using Gobos 1809
Create masks with Holdout collections 1814
Shallow Depth of Field in Eevee 1816
Randomize Grain 1820
Bevel Node 1821
Use HDRI only for foregrounds 1823
How to use motion blur 1825
Easy animated volumetric fog in Eevee 1828
Shadow Catcher in Cycles 1832
Keyframe render visibility 1835
Viewport Animation Render 1836
Cycles Rendering Basics 1838
Quickly open your renders in a browser 1843
Track Billboards to the camera 1844
Particle System Billboards 1845
Optimize Environment Scenes with the LOD add-on 1847
Proxies for Image Sequences on Planes 1851

NON-PHOTOREALISTIC RENDERING 1857
Basic Toon Shader 1858
Easy Outline with inverted hull 1860
Control outline thickness 1862
Sketchy moving toon outline 1863
Anime Style Ramen in Blender 1865
Inverted Hull Outline in Cycles 1871
Create an outline with the Line Art Modifier 1875
Add noise to Line Art Modifier strokes 1878
Baking the Line Art Modifier 1882
Pixelated Renders 1886
Correctly display pixel art textures 1888
Freestyle calligraphy render 1889
Pencil lines with freestyle 1893
Easily Add Camera Movement To A 2D Painting 1896
Cut-out Animation 1900
Anime tree in Krita and Blender 1905
Low poly texture atlas 1909

Click here to return to the index.

COMPOSITING 1911
Rendering passes as separate files 1912
Render Passes 1914
Compositing: Mist pass 1916
Compositing: Cryptomatte 1919
Compositing: Color Correction 1922
Compositing: Vignette 1924

WORKFLOW TIPS 1926
2 tips for navigating in a big environment scene 1927
Flip Node 1928
Top 3 solutions to Blender issues 1930
Make linked duplicates unique with one click 1933
Drag and Drop with the Asset Browser 1934
Use Collections and parented objects in the Asset Browser 1937
Asset Browser: How to mark Node Groups as Assets 1940
Drag-and-drop Decals from the Asset Browser 1942
Node Groups 1947
Concept Art Workflow - Trace image to Grease Pencil 1949
Linking Characters 1955
Recover Lost Blend-Files 1956
Purge All Orphan Data 1958
Custom Startup Settings 1960
Import SVG vectors with colors intact 1961
Import SketchUp models 1963
Workflow tip: Linked Libraries 1965
Editing Linked Libraries 1967
Copy Rotation with Copy Attributes Menu Add-On 1968
Get 6000 free assets and an Asset Manager with Blenderkit 1970
Switching between different cameras 1975
Stored Views 1977
Snapping and Origin transform shortcuts 1979
Using Simplify, Camera Culling and Clipping Regions 1981
High Quality Eevee Viewport by Default 1984

INTERFACE TIPS 1986
Quickly Switch Objects While Sculpting and Texture Painting 1987
3 ways to isolate objects 1989
Hide Everything Except Selected 1992
Isolation Mode 1992
Simplify Setting 1993
Copy Objects From One Blender Window To Another 1994
Hold CTRL To Snap Objects 1995
Viewport Visibility Tips 1996
Quad View Toggle 1998
Zoom To Selected 1999
Toggle Fullscreen or Maximize Area 2000

Click here to return to the index.

Quick Favorites 2001
Quickly Find Objects In The Outliner 2002
Snap to orthographic view 2003
Random Object Colors in the viewport 2004
Colorful Wireframes 2004
Incrementing your file names 2005
Retopology visibility tips 2006
Node labels and colors 2008
More Node Tips! 2009
Even more Node Tips! 2011
Set in- and out-points in the timeline 2012
Grid snapping 2013
Draw select and lasso select 2014
Align objects to camera view 2015
Interface scaling 2016
Scaling Tips 2017
Blender UI Value scrolling 2018
Lock to 3D cursor 2019
Maximum undo & Picker shortcut 2020
Swap interface windows 2021
How to hide lights in Eevee 2022
Eevee Viewport Color Management 2023
3D Vocabulary 2026

To go to a tip, click on the title in the Index. To watch the original video on Youtube, click on the **green header title** on the top of the page.

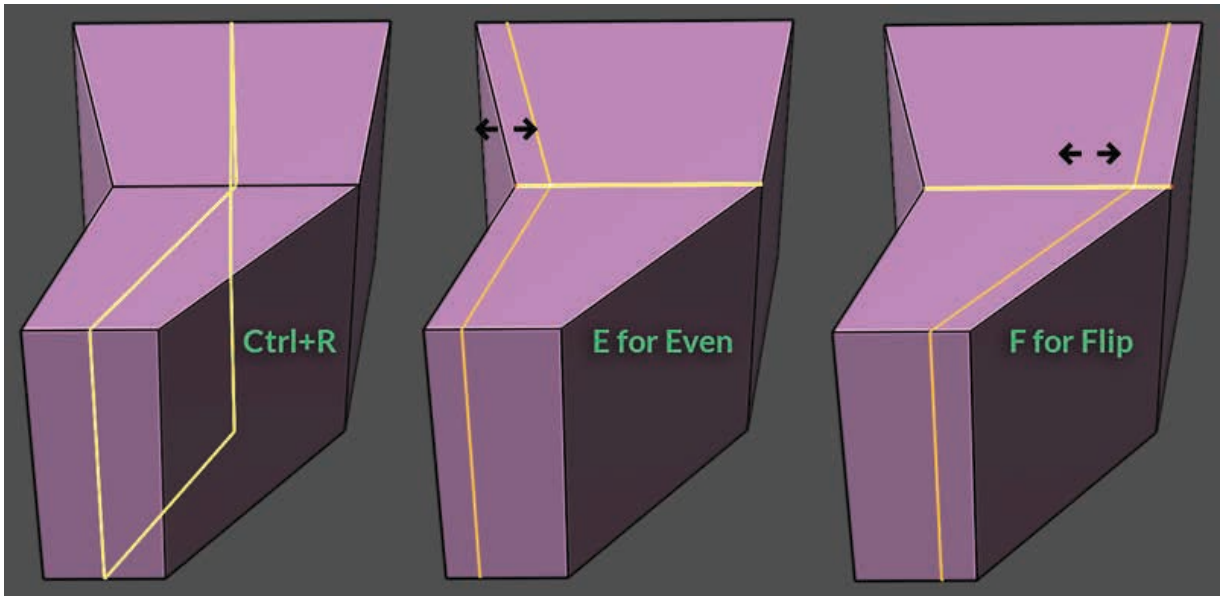
Blender Secrets is written by Jan van den Hemel for Video Publishing BV in Belgium. ISBN number: 9789464337303

MODELING BASICS

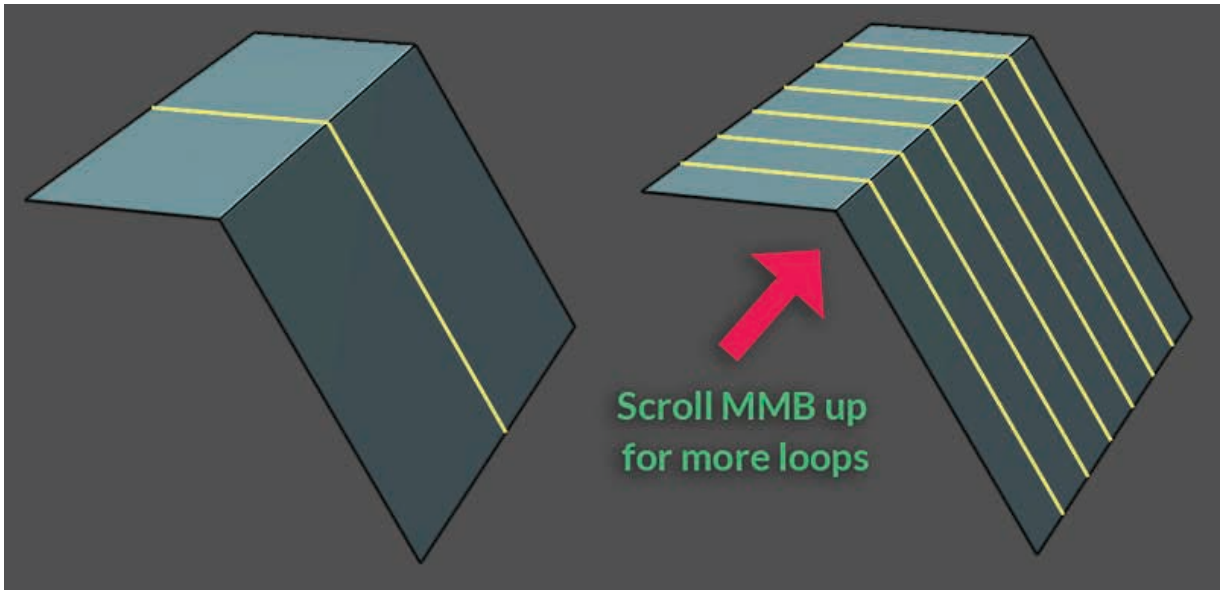
Loop Cut and Slide Tips

Having to add Loop Cuts is something you'll come across often when modelling. There's quite a few options that you can use to your benefit.

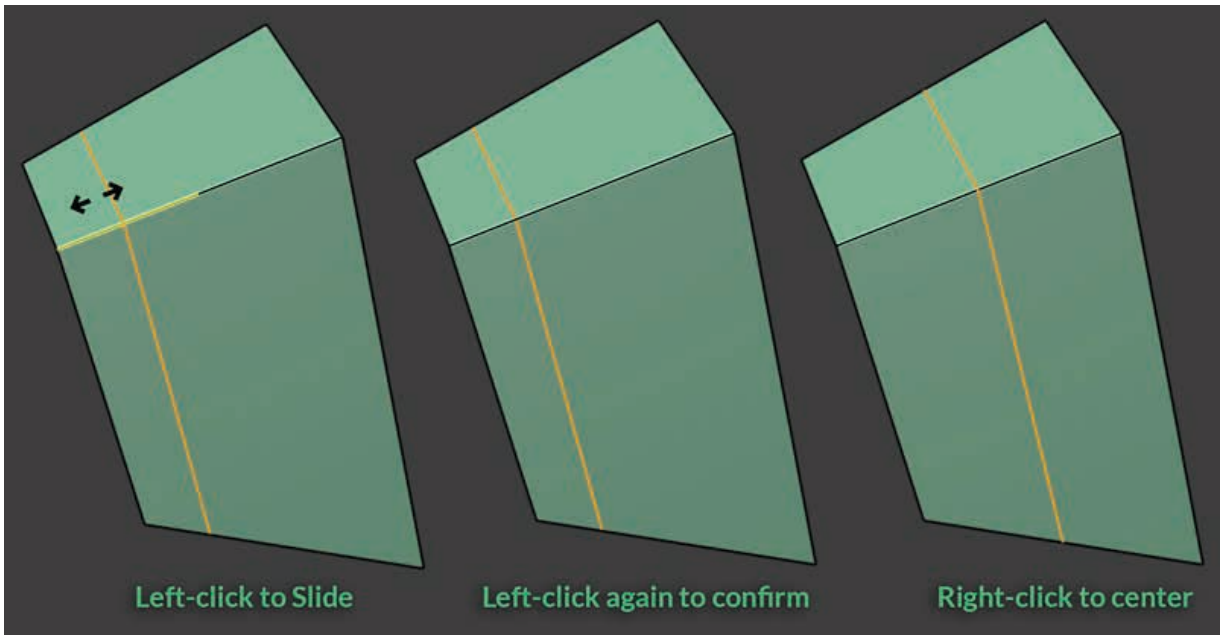
To add Loop Cuts, **press Ctrl+R** in Edit Mode. **Press E** if you want it to evenly match one of the adjacent edges. **Pressing F** flips which edge it matches.



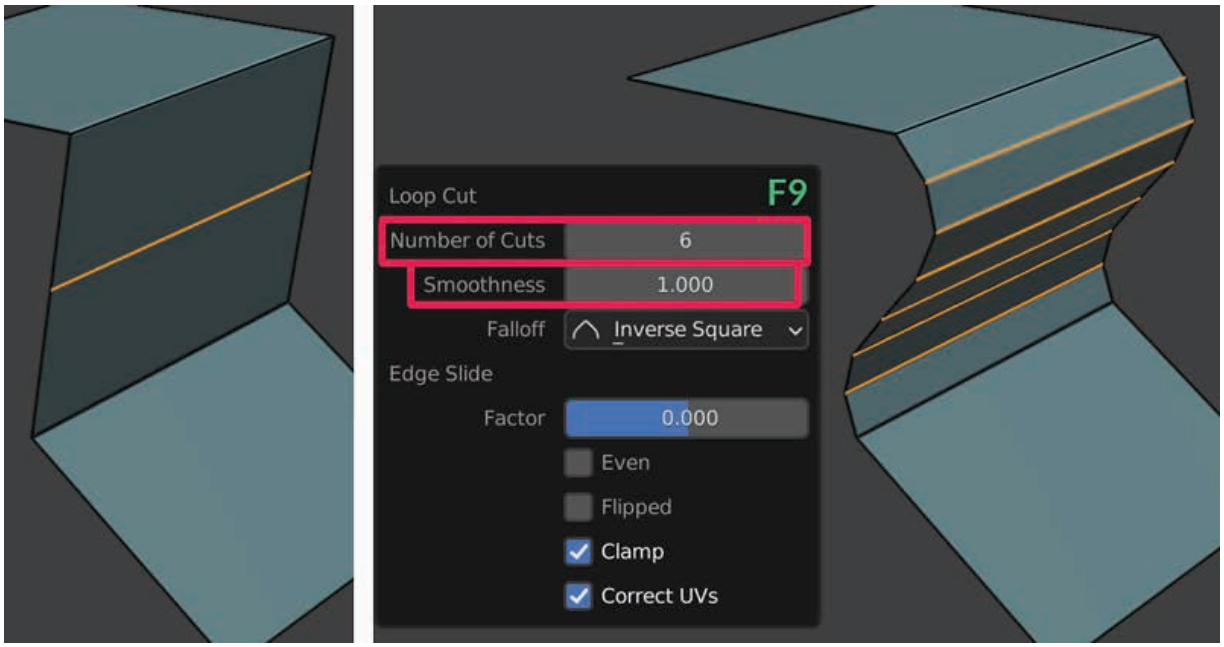
You can add more cuts by **scrolling the mouse wheel up**. Note that you can't do this anymore after pressing E, but you can still adjust the amount of loops in the panel that opens after confirming the loop cut placement (more on this later).



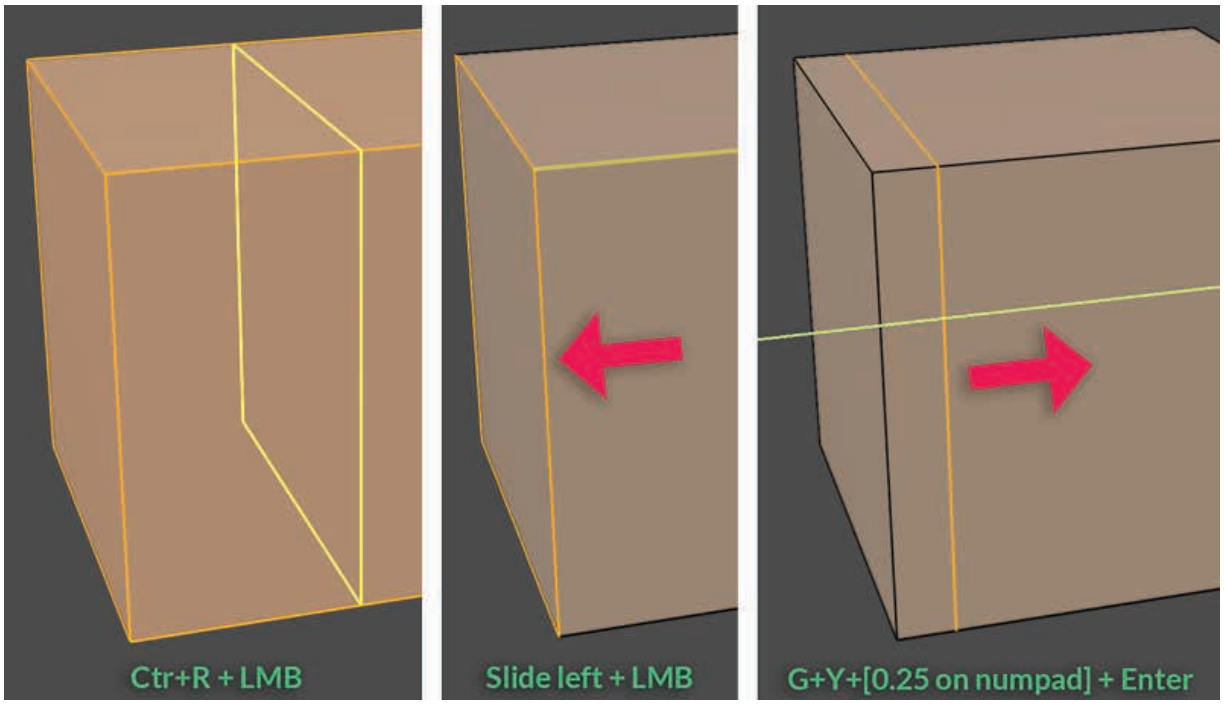
After **left-clicking**, you can slide the loop cuts up or down and **left-click again** to confirm. Or **right-click** to place the loop cut in the middle.



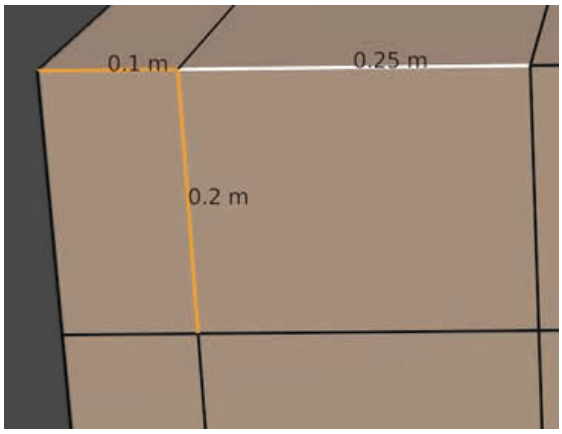
You then get the **Adjust Last Operation** panel where you can still some values. **Press F9** if you accidentally closed this menu. The **Smoothness** value scales the selected loop cuts using a specific Falloff profile.

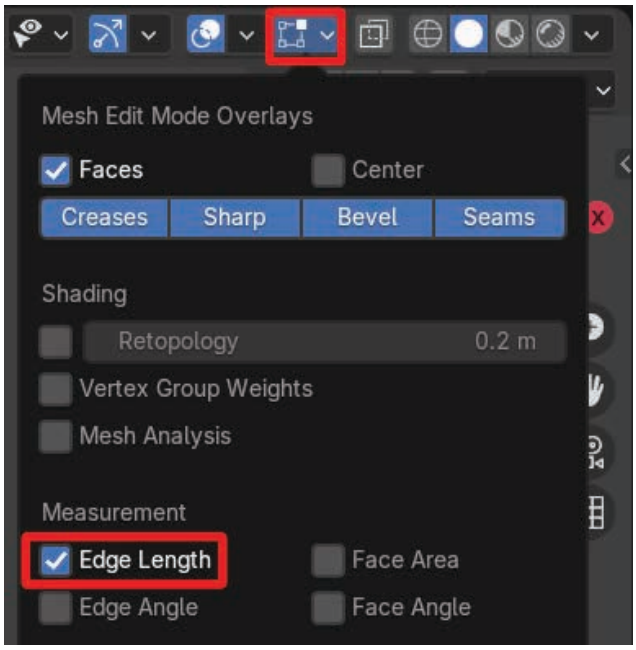


In case you want to add a Loop cut at a very precise distance from an edge, create the loop cut and slide it to that edge. Then, press G+the axis you want to move it on, and the distance on the Numpad. For example, **G+Z+[0.25]** to move it .25 units on the Z-axis from that edge. This can be useful when working on an architectural scene, where you need to follow precise real-world measurements.



You can also turn on **Edge Length** in Mesh Edit Mode Overlays and hold Shift to slide more carefully, for precise placement of the loops. Select an edge to see how long it is.





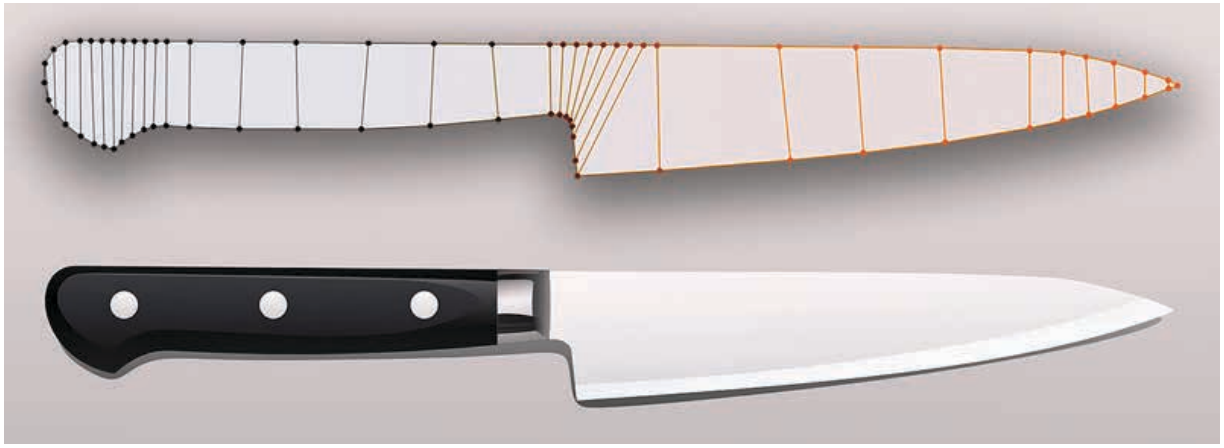
Some additional tips: If you want to slide edge loops at a later point, you can always do so by pressing G twice. Then you can slide them until they hit the edge closest to them.

After that, you can press G+[the axis you want to move them on]+[the distance on the Numpad], as explained above. To toggle between a positive and negative value you can simply press the [-] (minus) key on the numpad.

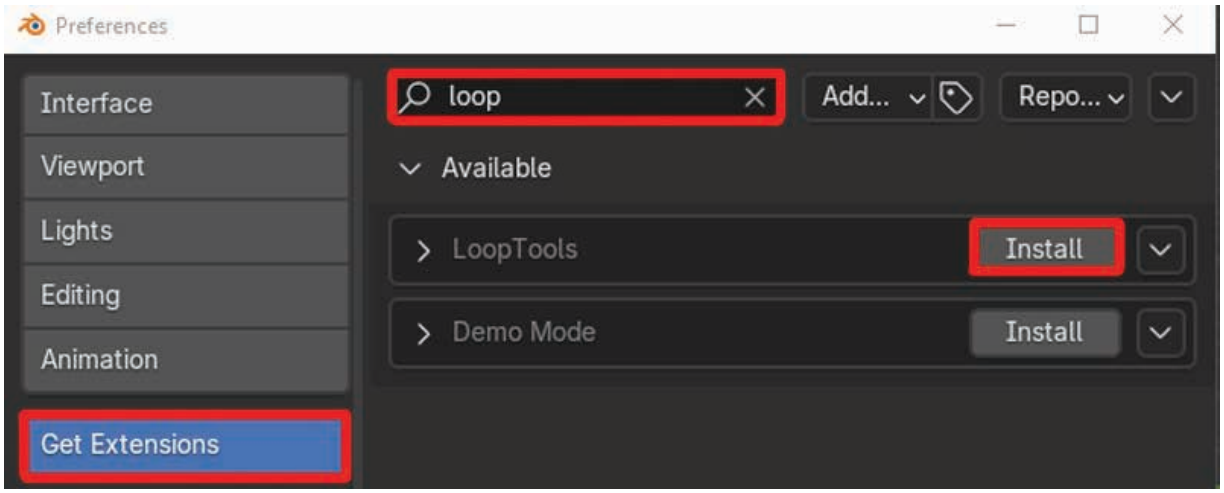
Fun fact, pressing the other [-] key (the one not on the Numpad on your keyboard) while hovering the mouse cursor above any value field in Blender, will turn that value negative (or positive, if it was already negative).

LoopTools - Curve and GStretch

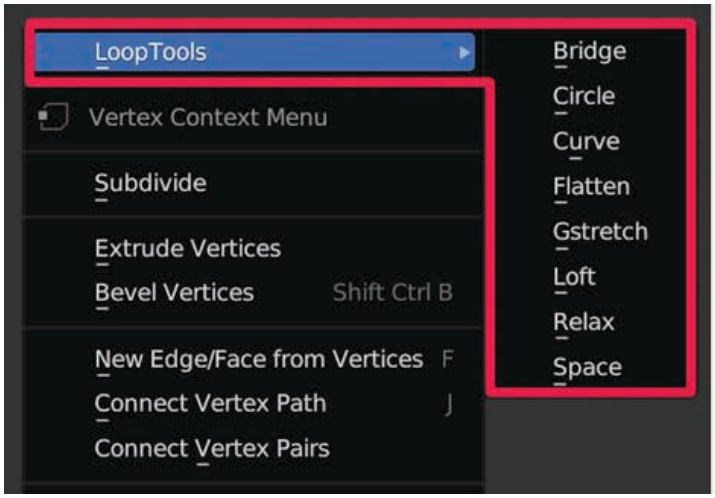
LoopTools are a set of essential modelling options that need to be enabled in Preferences. The LoopTools **Curve** and **GStretch** options in particular, are useful tools in Blender for creating and manipulating curved surfaces.

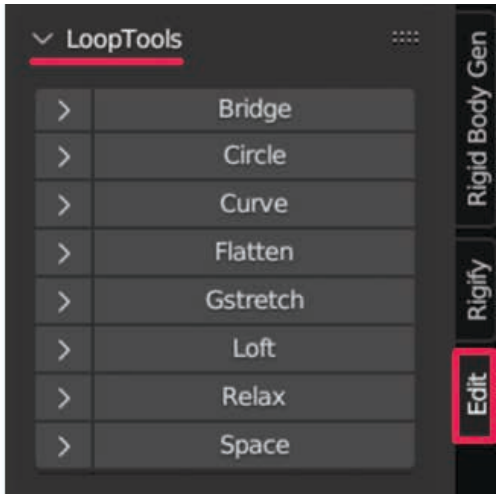


Go to Preferences and enable the Extension called LoopTools.



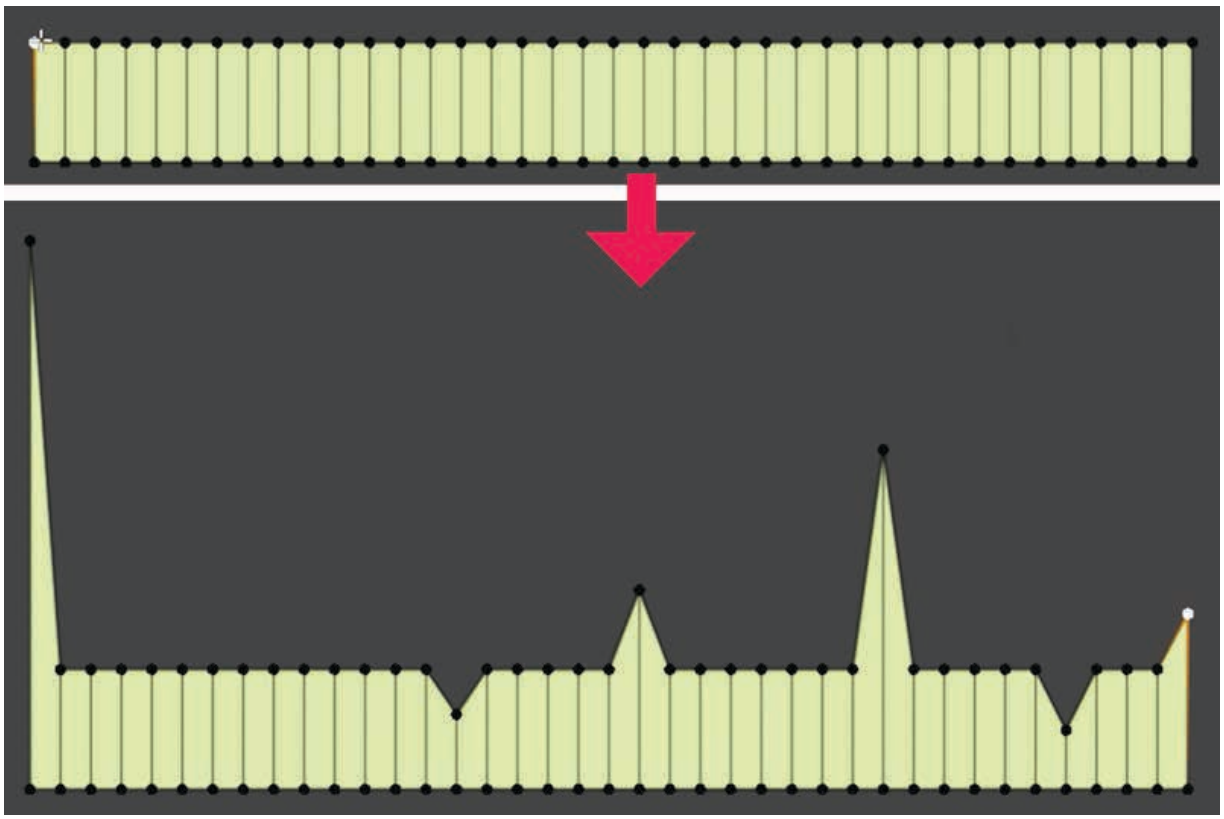
After enabling the add-on, you can find the options in Edit Mode when **right-clicking**. Alternatively you can find them in the **Option panel** in the **Edit tab**. Note that it will only appear there, when you're in Edit Mode.



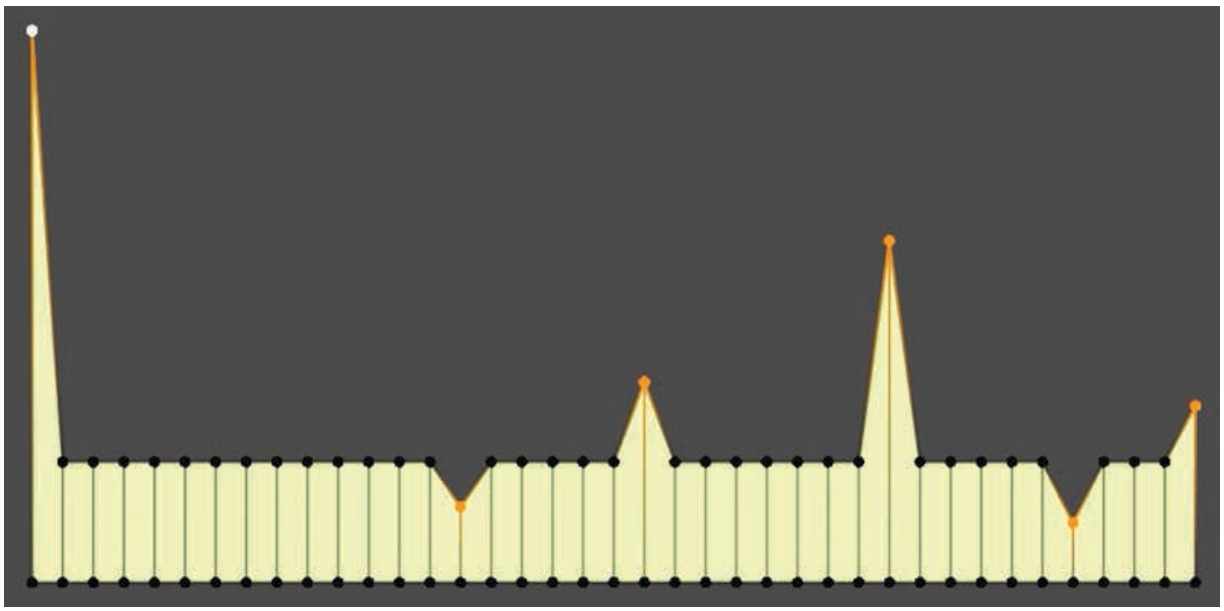


Click here to return to the index.

Select three or more vertices and move them up along the Z-axis. You can do this, by selecting them one by one and pressing G (move) and then Z to constrain the movement along the Z-axis.

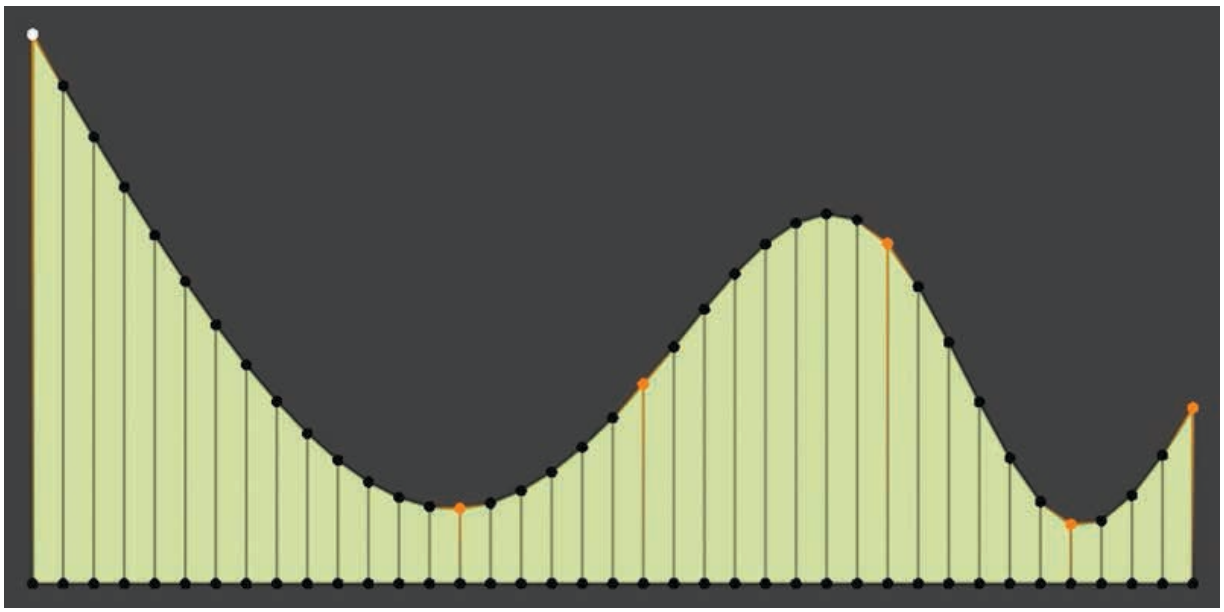


Select all of the moved Vertices while holding Shift.

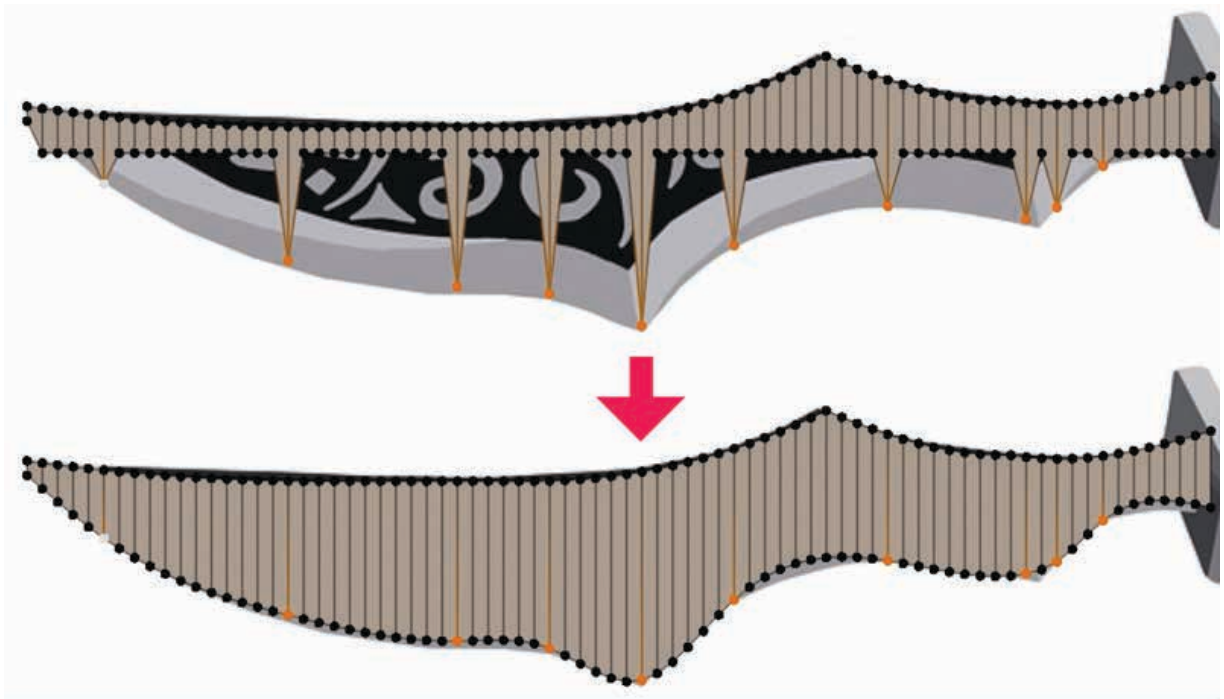


Use the Curve tool by right-clicking, and choosing **Curve** from the LoopTools options.

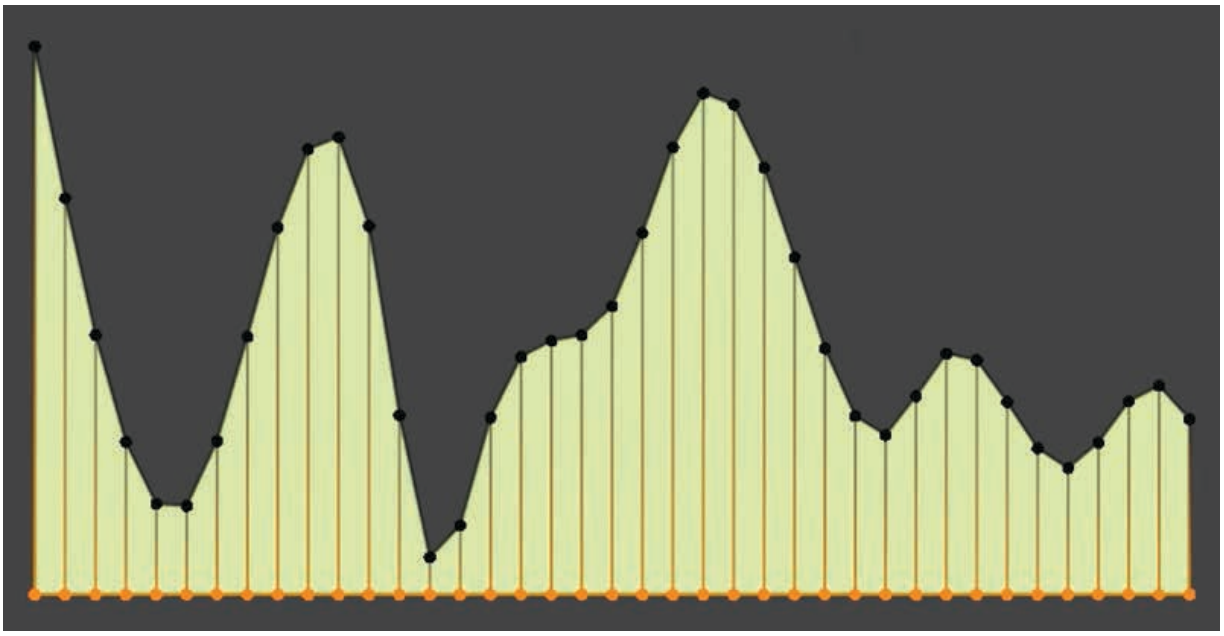
As you can see this interpolates the vertices in between the selected ones.



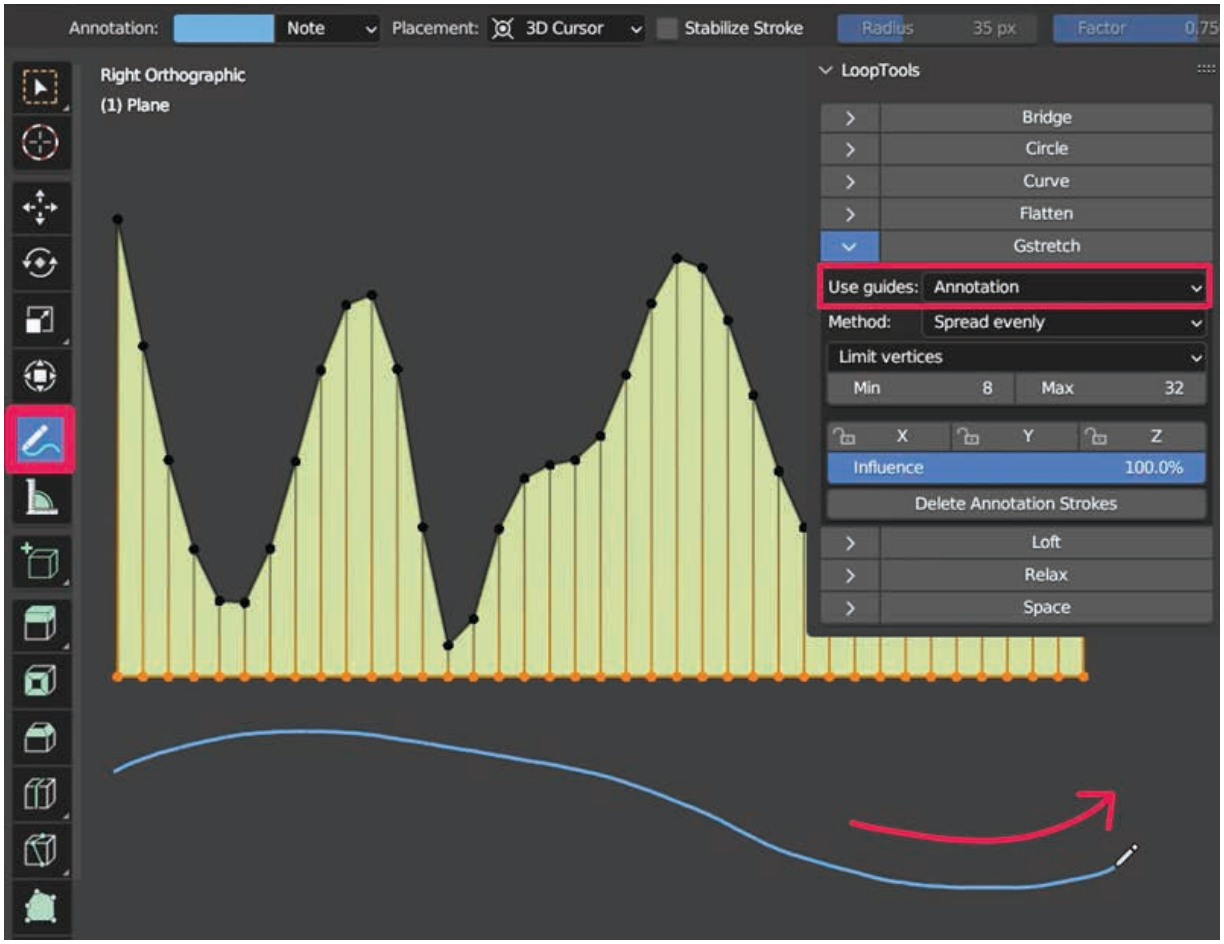
We can use this to quickly create interesting shapes, like a knife or dagger.



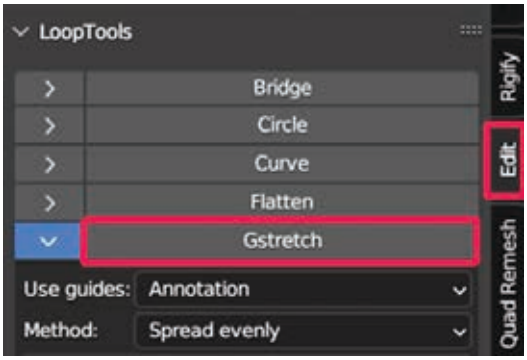
The **GStretch** option uses the **Annotate** tool. First, select a row of vertices.



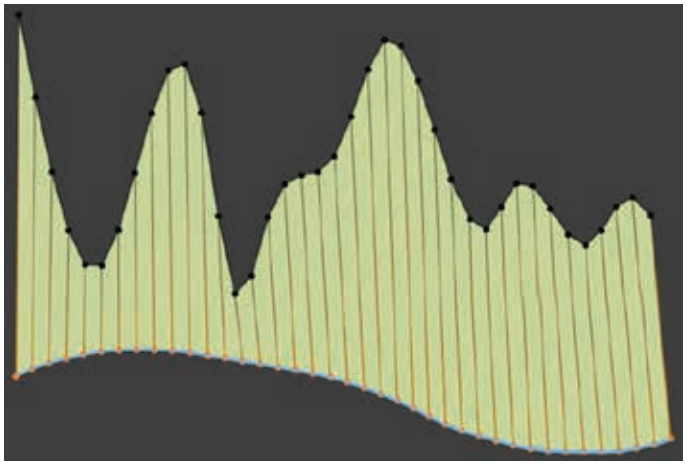
In the GStretch options, choose **Annotation** as Guides. Then draw an annotation. You can draw annotations at any time in Blender by **holding D** and drawing with the left-mouse-button. Holding D and the right-mouse button acts as an eraser.



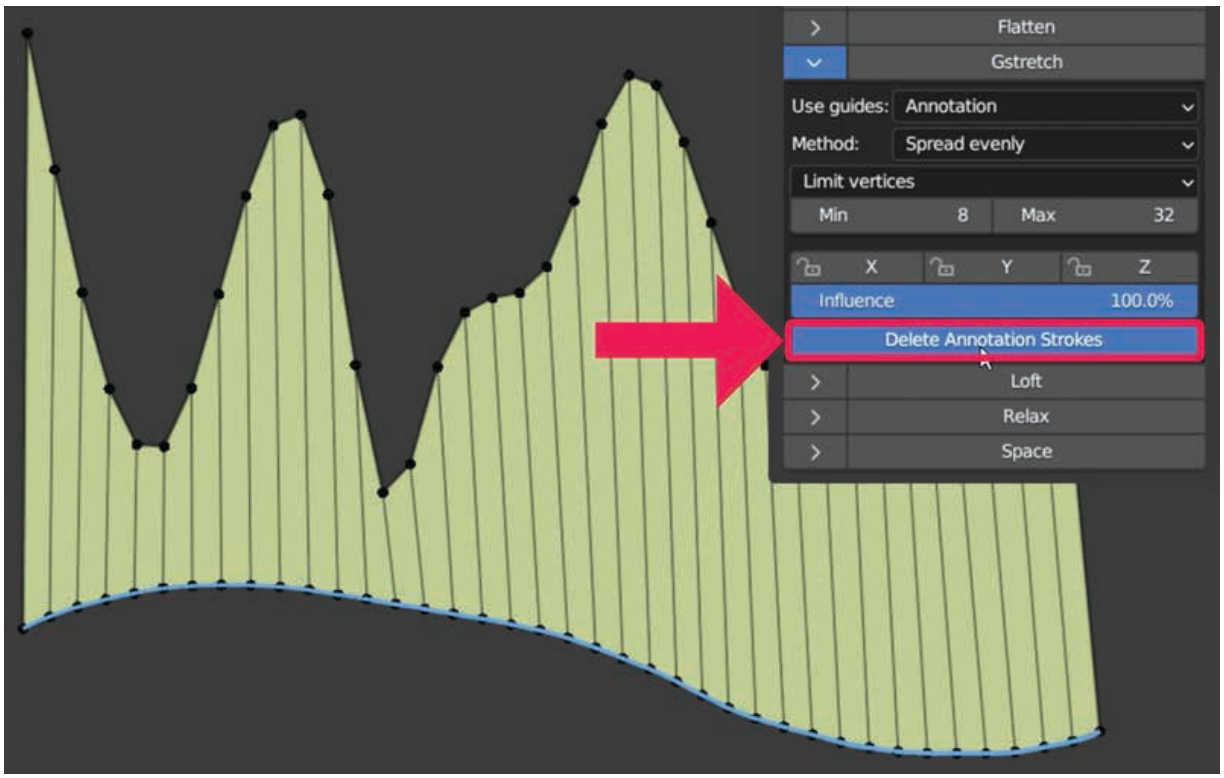
Then click on **GStretch**.



As you can see this stretches the selected vertices to the annotation line.

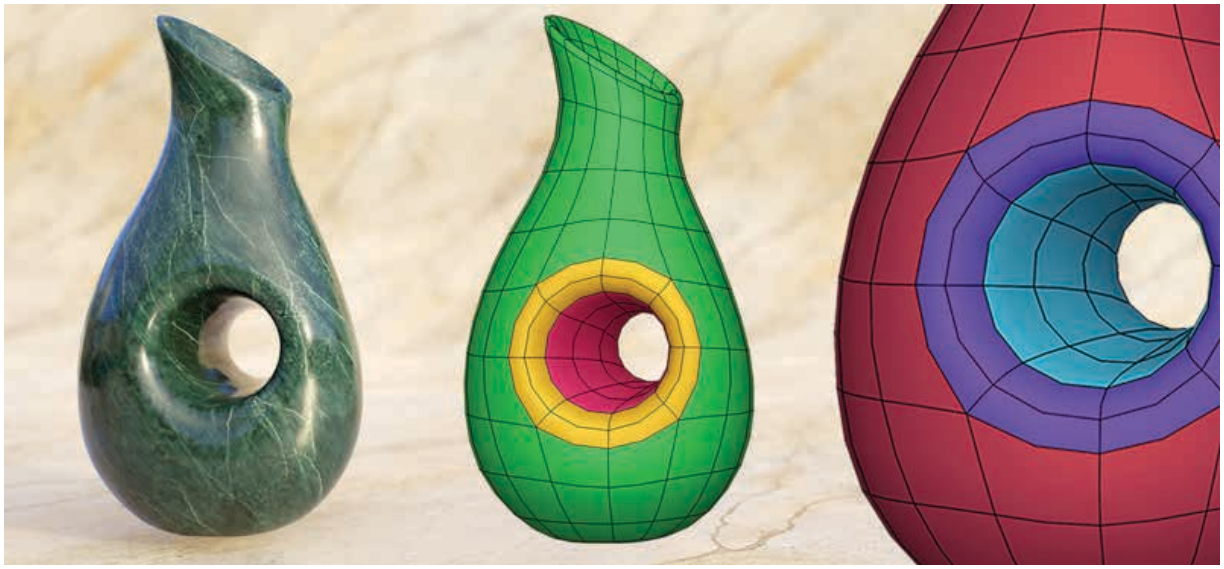


Deselect all Vertices by **pressing Alt+A**. Then you can click on **Delete Annotation Strokes** to remove the annotation stroke.

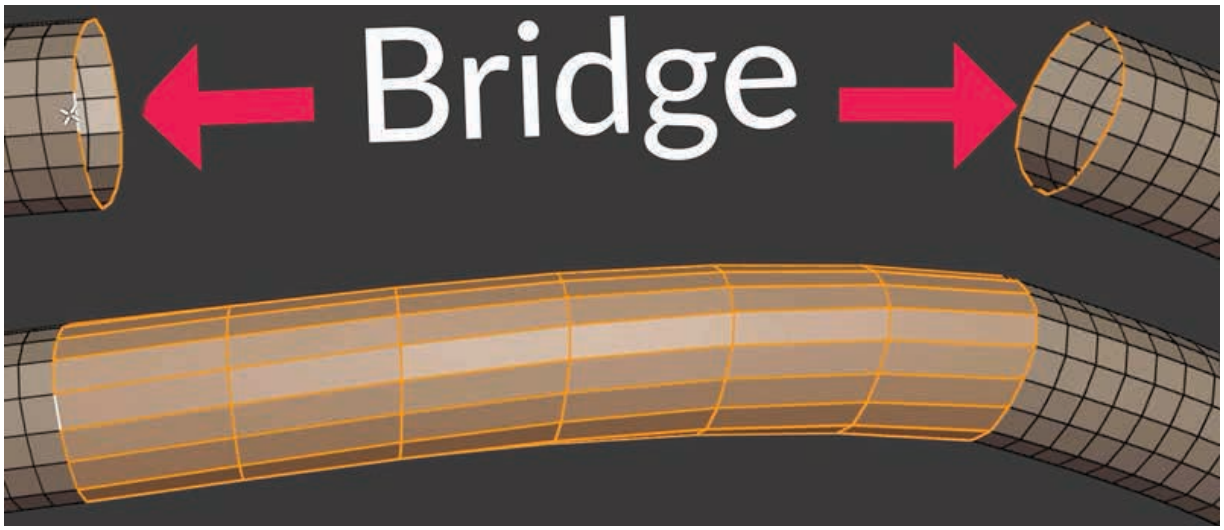


LoopTools - Bridge, Loft and Circle

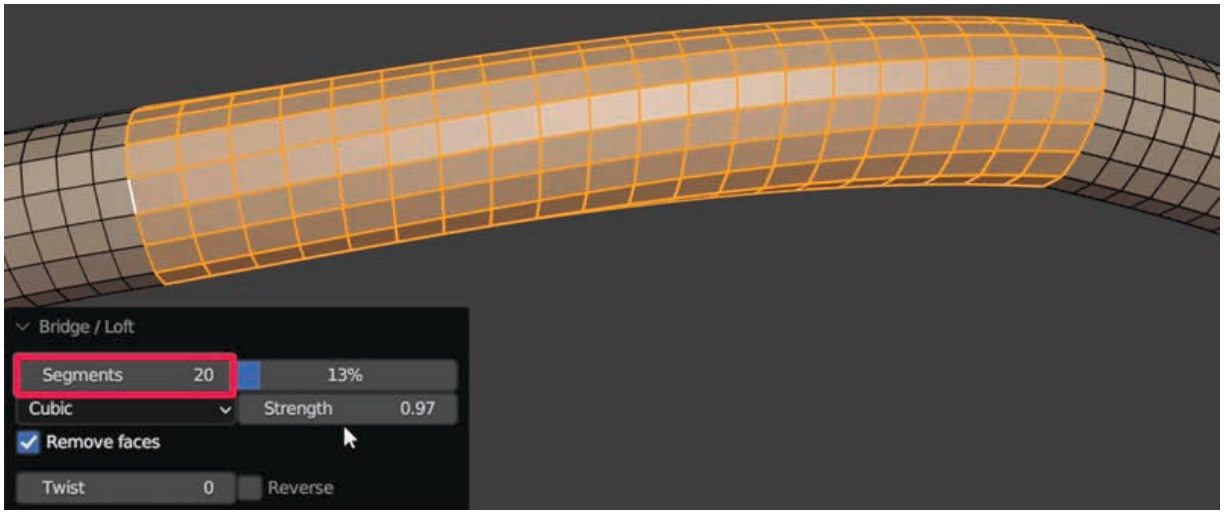
Continuing with the LoopTools extension that comes by default in Blender, using the options Bridge, Circle and Loft allows us to create some interesting shapes, like a vase with a hole in it.



The Bridge Tool bridges two Edge selections with new geometry.

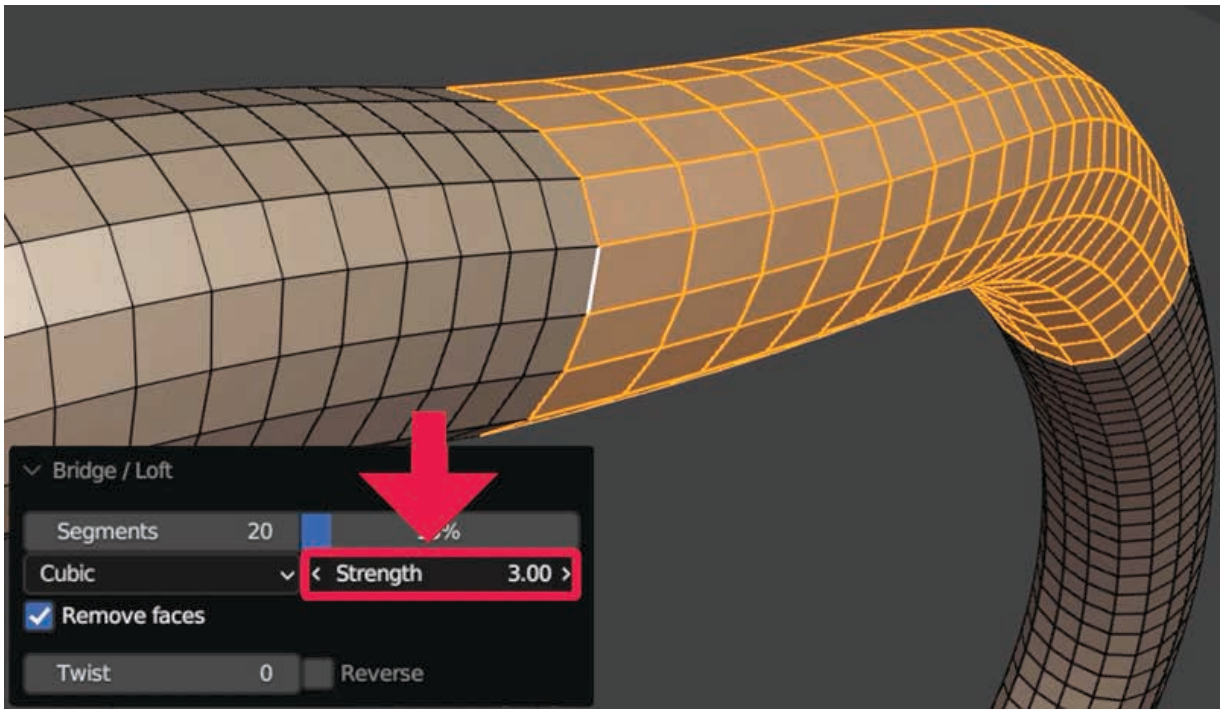


You can then increase the amount of segments as needed.

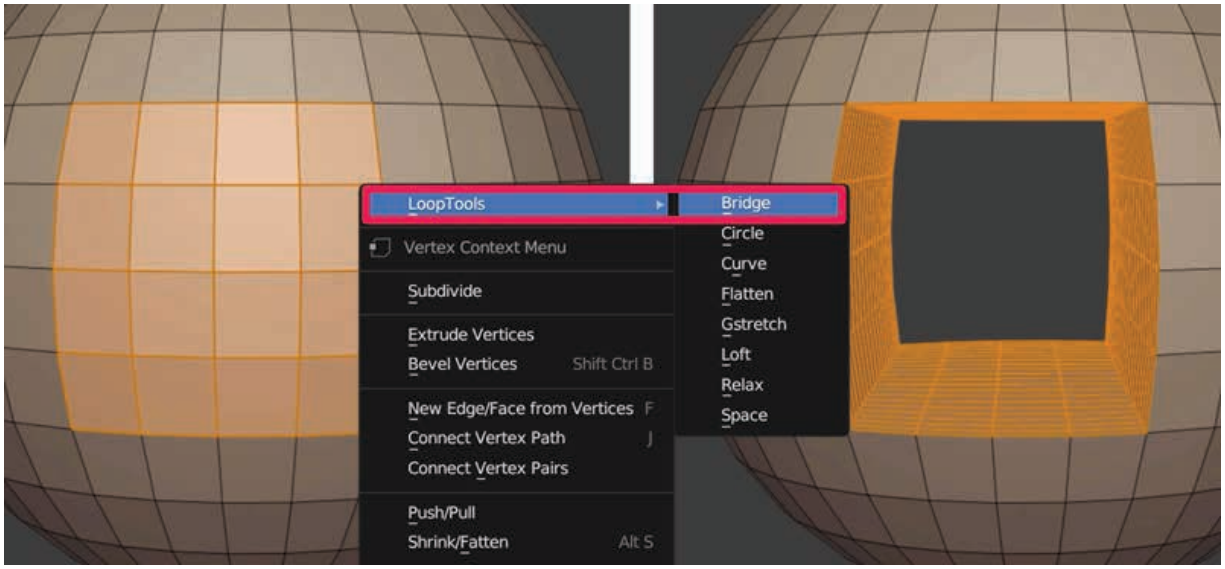


Click here to return to the index.

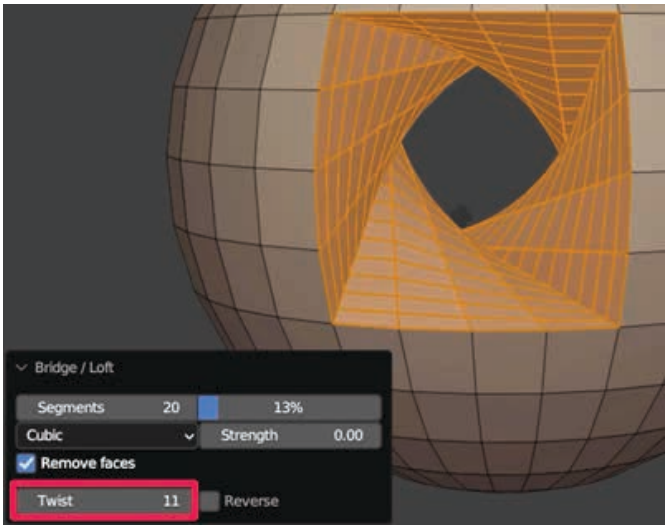
As well as adjust the shape of the bridge by changing the Strength value.



Bridge can also be used to create a hole in an object.

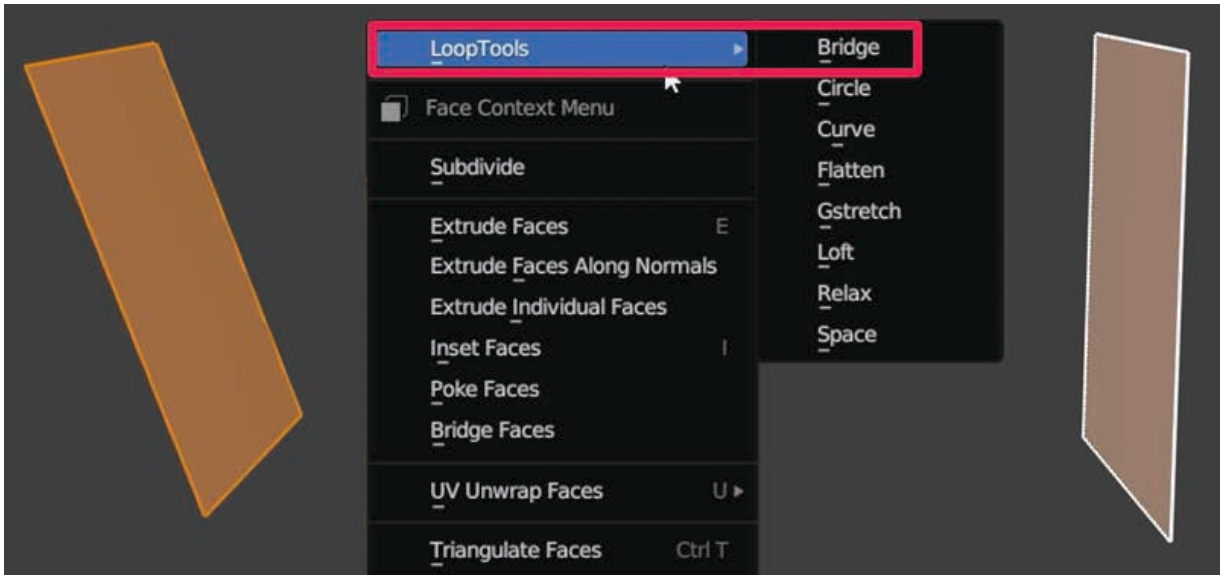


The **Twist** value twists the bridging geometry.

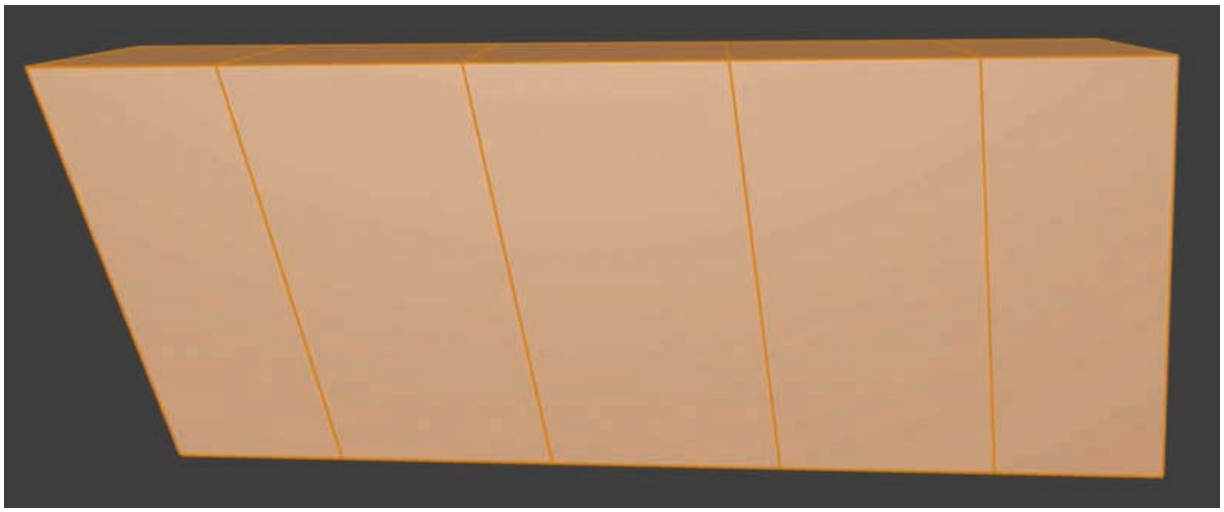


Click here to return to the index.

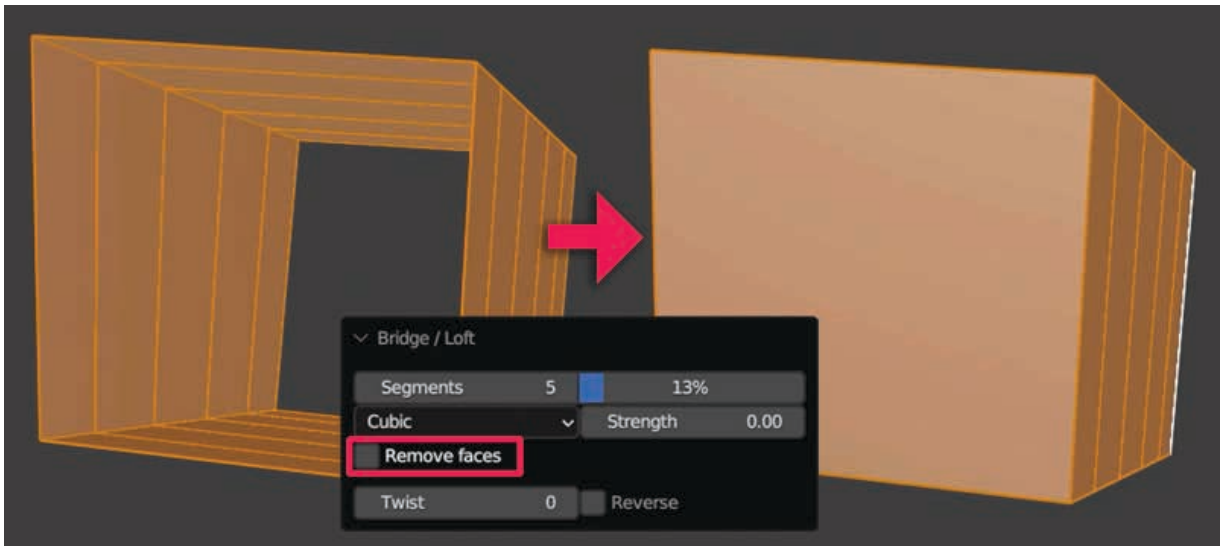
You can also connect two parts of an object as long as both are selected in Edit Mode.



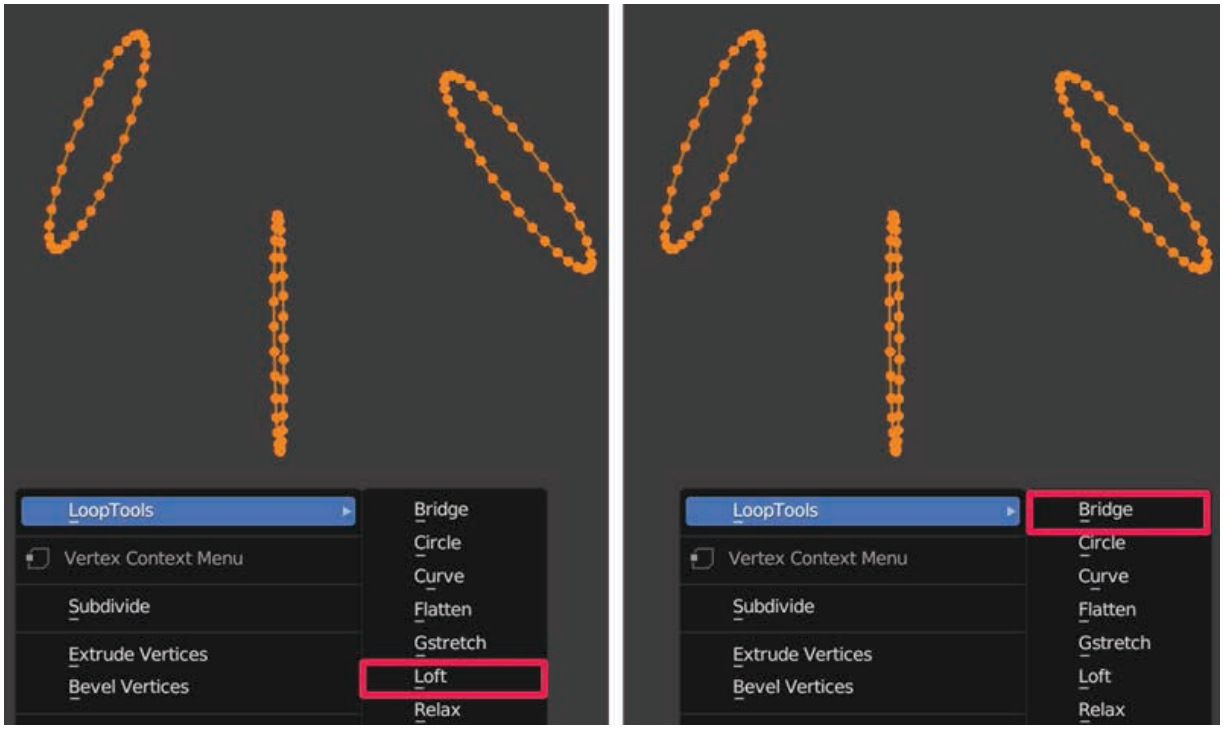
New geometry is created to bridge the selection. Note, that the original selected geometry has been automatically deleted.



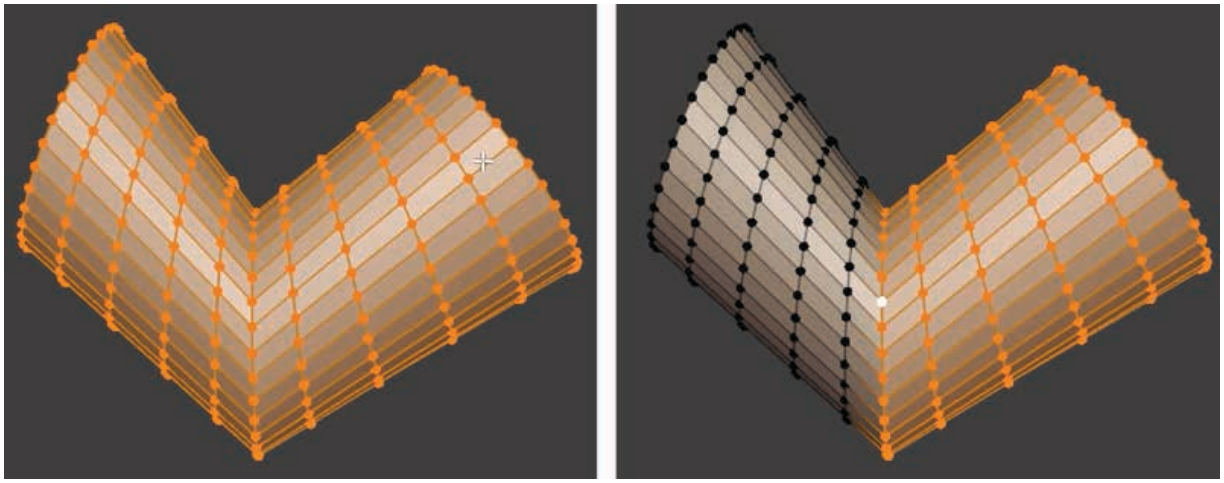
Turn off "**Remove faces**" to stop the original selection from being deleted.



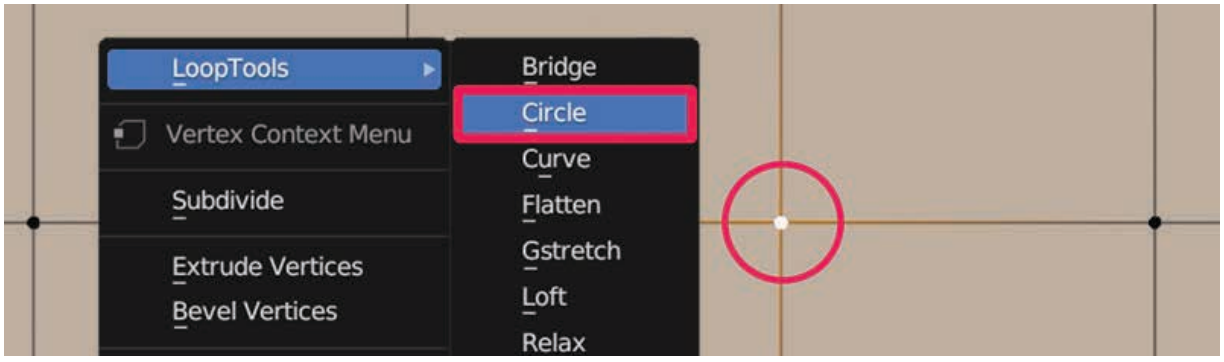
The **Loft** tool is quite similar to the **Bridge** tool in that it creates new geometry between selected edge loops.



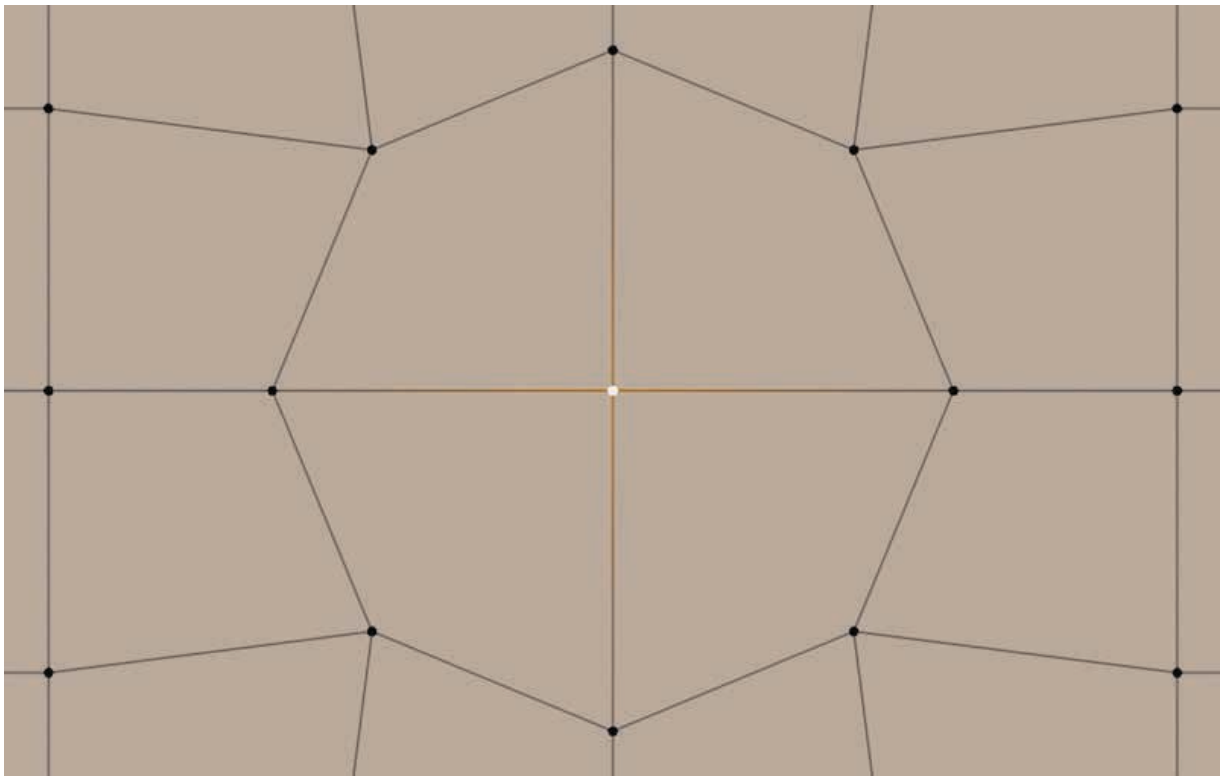
However with Bridge you'll need to do the same action several times to bridge all the geometry. In other words, Loft is faster in some cases.



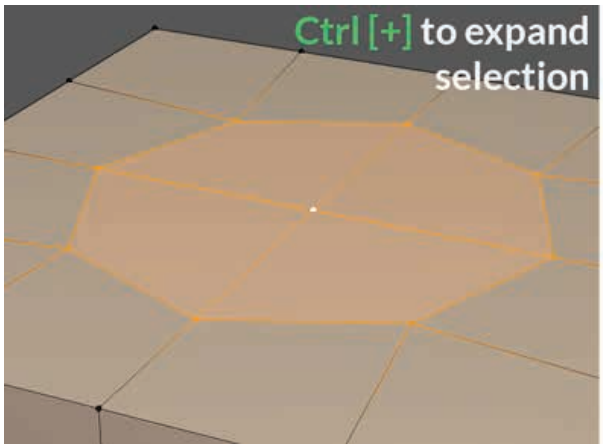
Select a single Vertex, **right-click and choose Circle.**

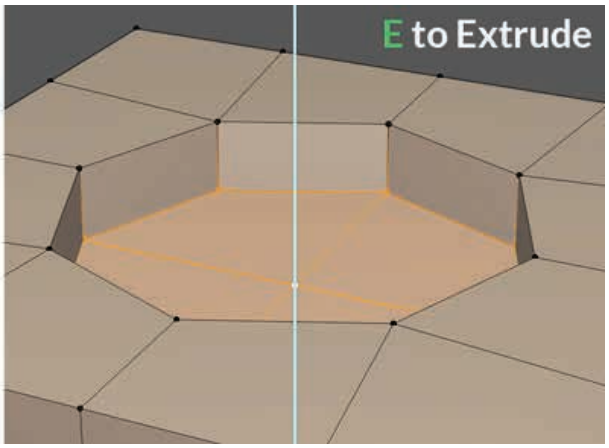


This turns the surrounding vertices into a circular shape.

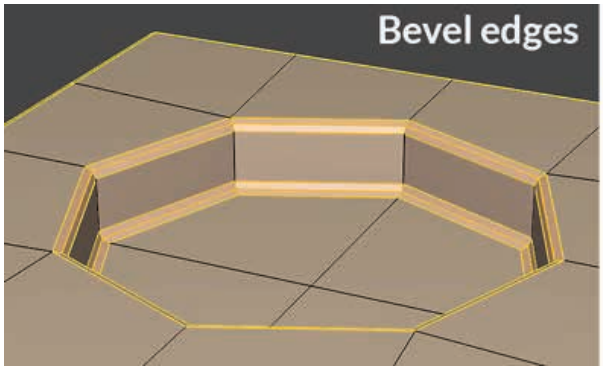


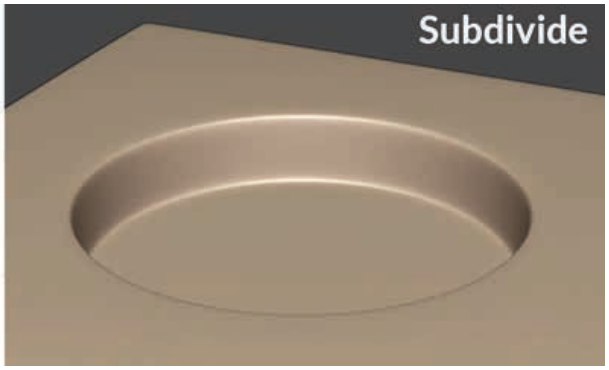
By **pressing Ctrl and [+ on the Numpad]**, you can increase the selection. Then you can extrude this to create a hole.



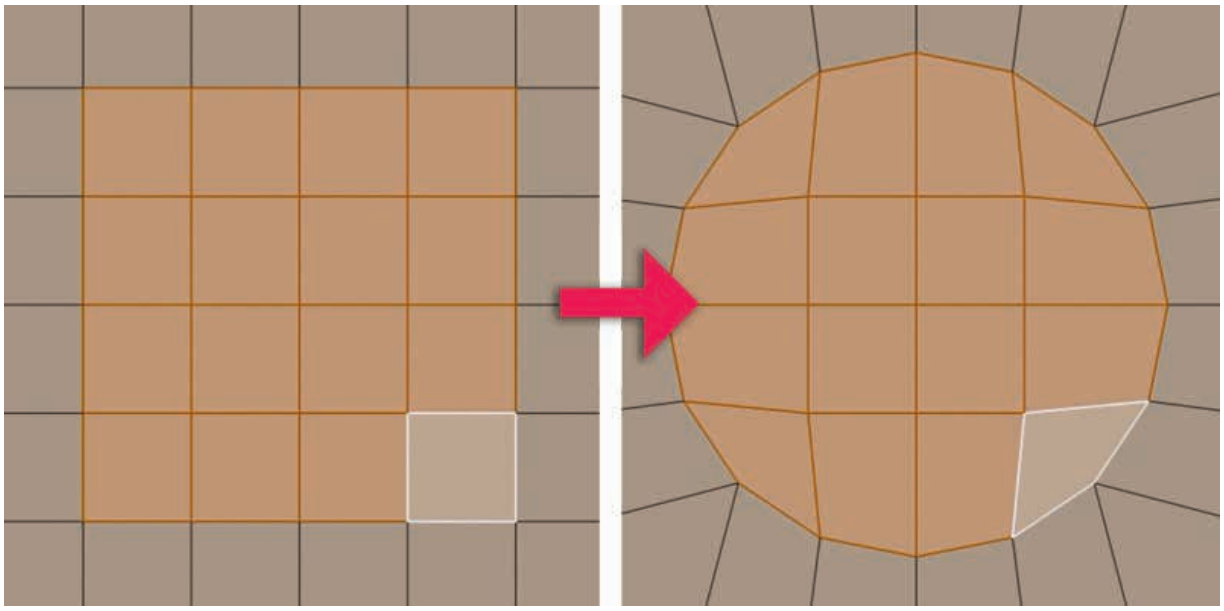


If you Bevel the edges with **Ctrl+B** and add a **Subdivision Surface modifier** to it, you get a nice smooth circular gap.





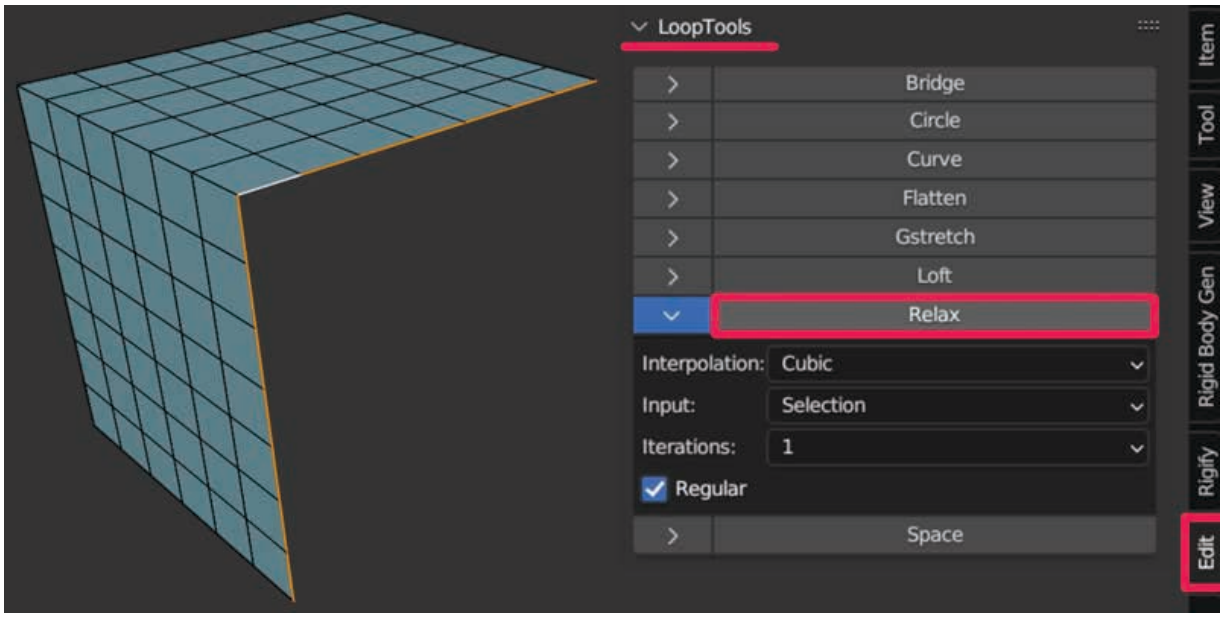
Instead of a single vertex, you can also select a number of Faces and make those circular.



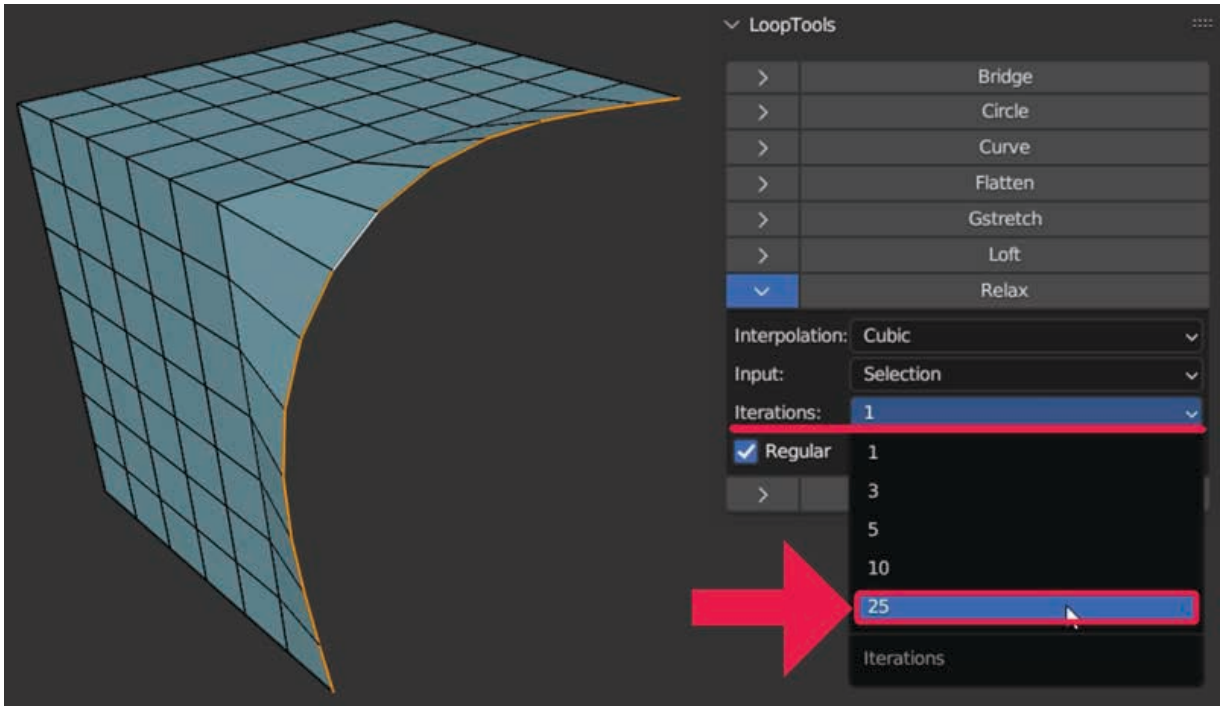
LoopTools - Relax, Space and Flatten

Continuing with LoopTools, let's have a look at the Relax, Flatten, and Space options.

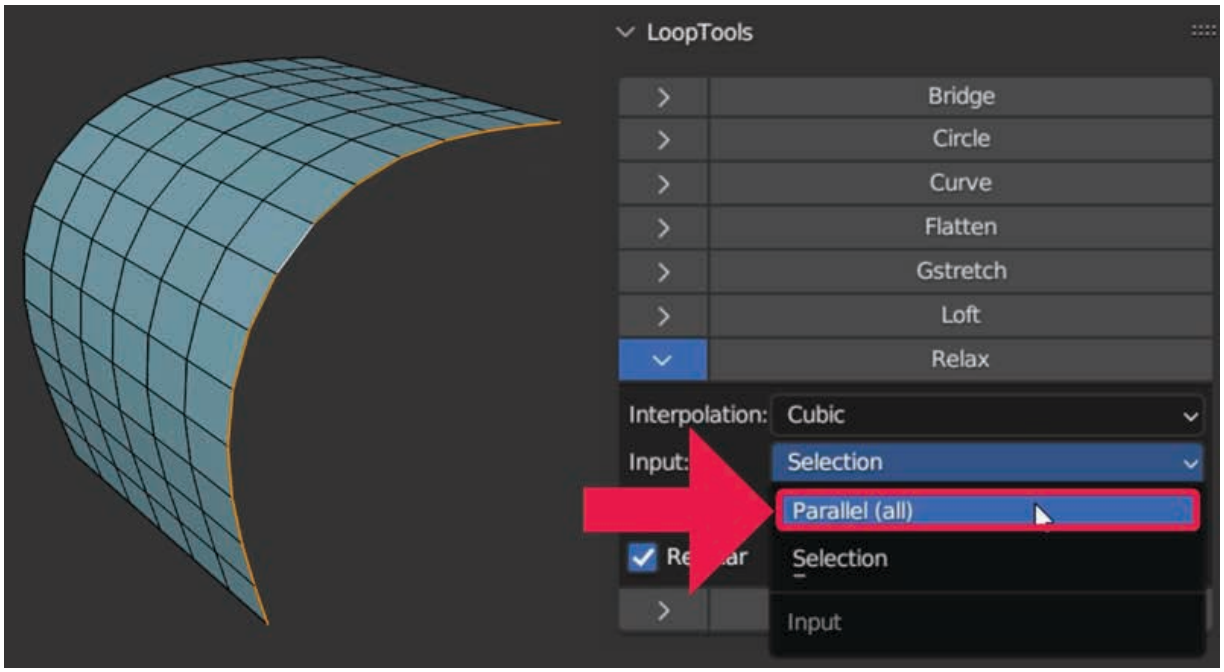
The relax tool makes a selected Edge Loop smoother.



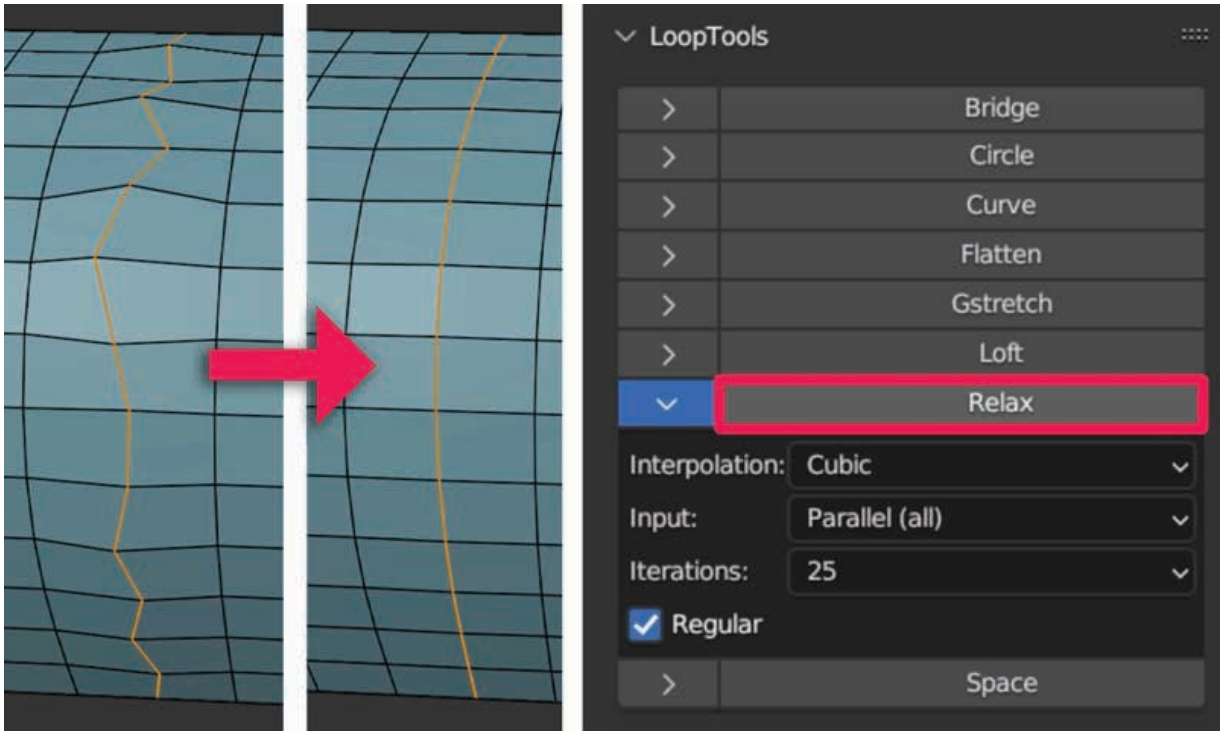
You can do this in small increments, or set the Iterations to a higher value to have a stronger effect once you click on Relax.



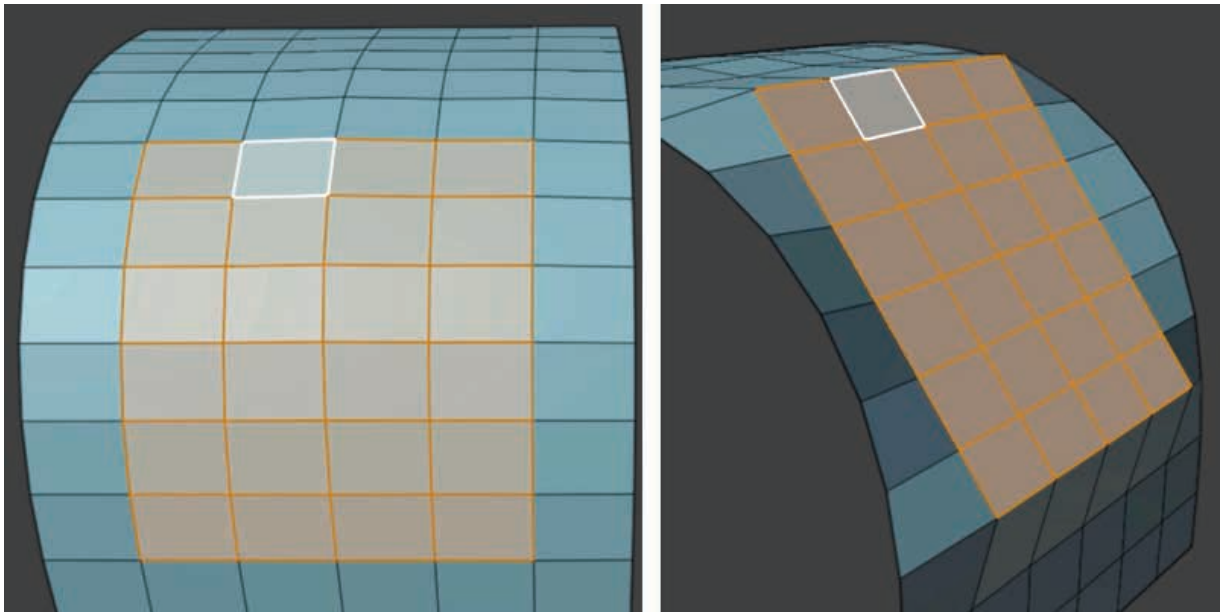
With Input set to **Parallel**, the Edge Loop on the other side will also be influenced when you click on Relax.



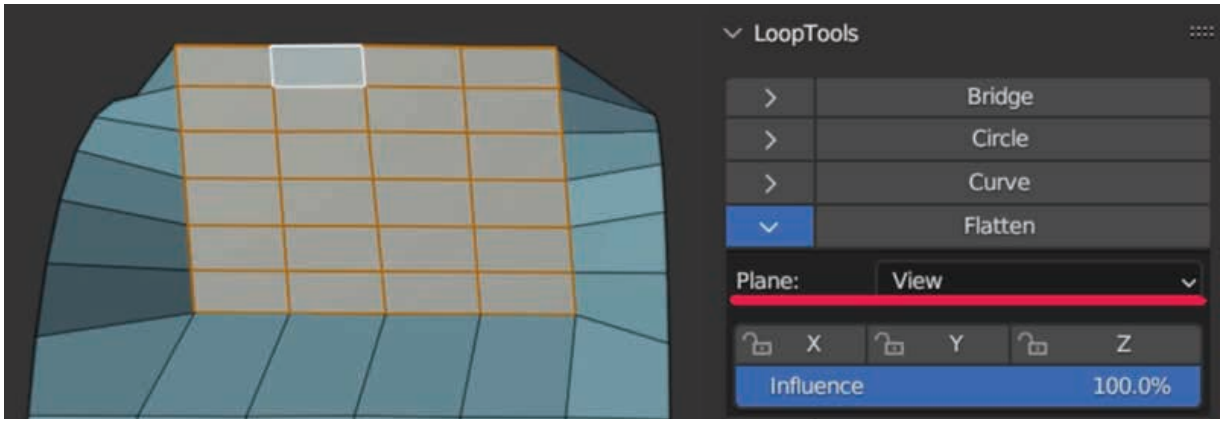
Relax can be useful to make a messy Edge Loop more smooth.



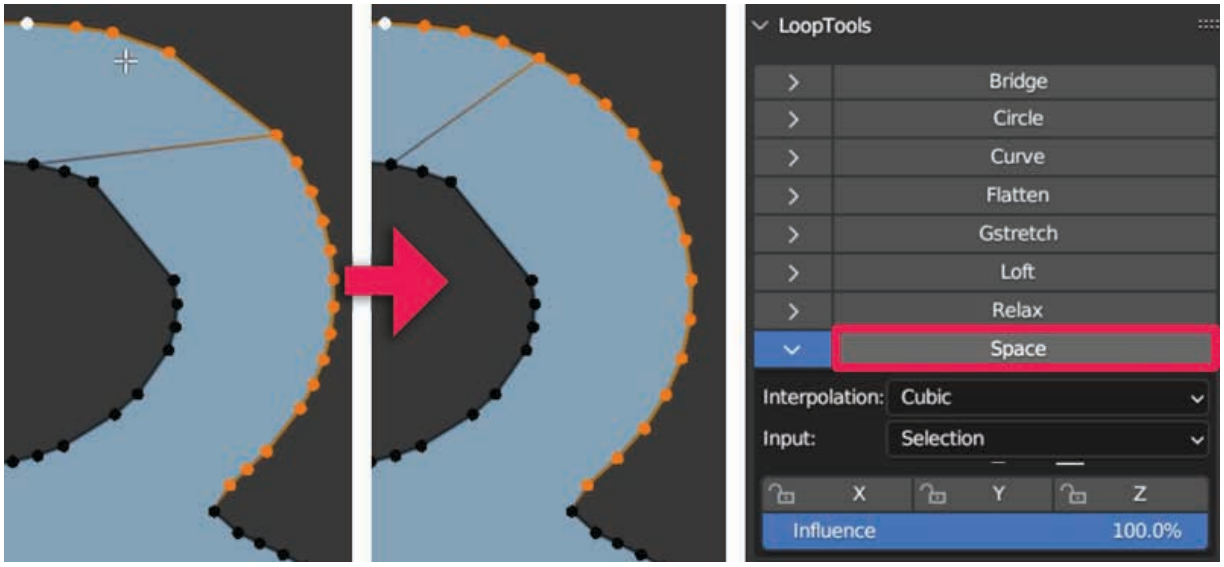
The **Flatten** tool unsurprisingly, flattens a selection.



Besides flattening along the Normal axis, you can set the flattening plane to **View**. Then, it will flatten from the viewing angle.



When you select vertices and press Space, it evens them out by equalizing the spaces between them. This can be useful for text objects that have been turned into meshes, for example.

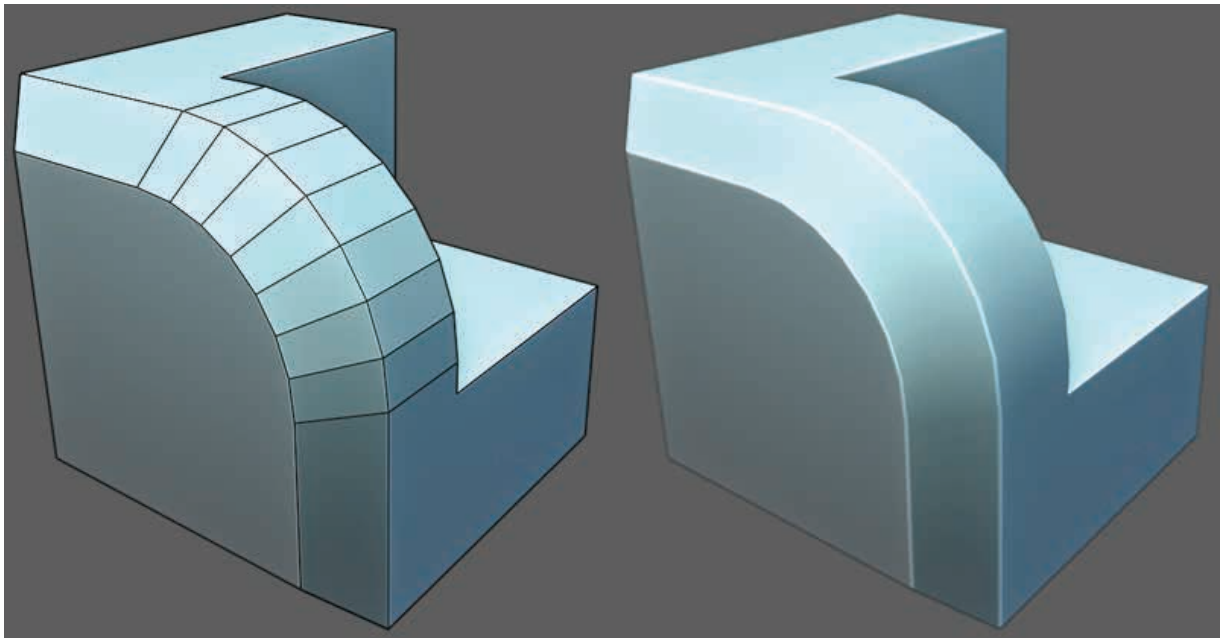


Click here to return to the index.

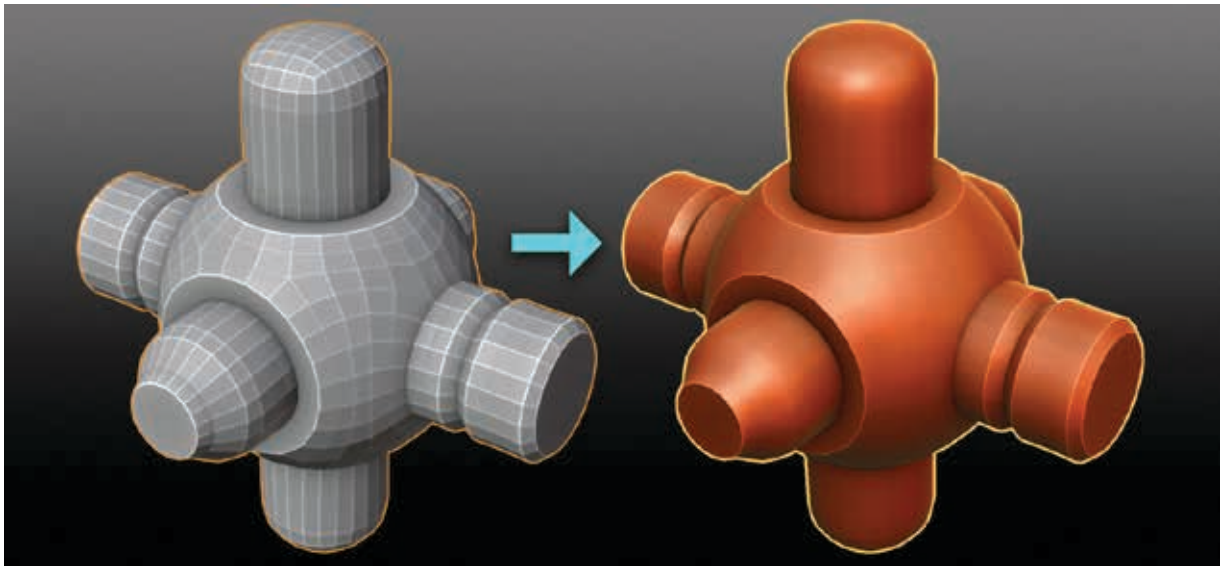
How to use Shade Auto Smooth in Blender 4.3

Shade Auto Smooth is a great option for quick modeling without Subdivisions. You can make really fast models using boolean cutting, N-gons, everything that would cause trouble with Subdivisions, and just get a smooth / sharp look by right-clicking in Object Mode and choosing Shade Auto Smooth.

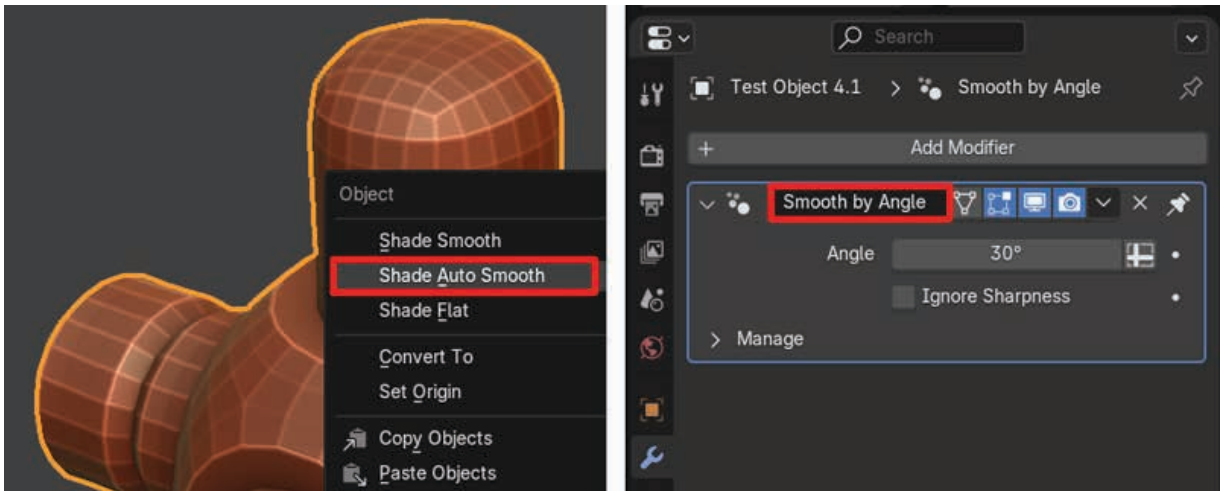
Not for good topology, but great for quick concepting and the early blocking out stage.



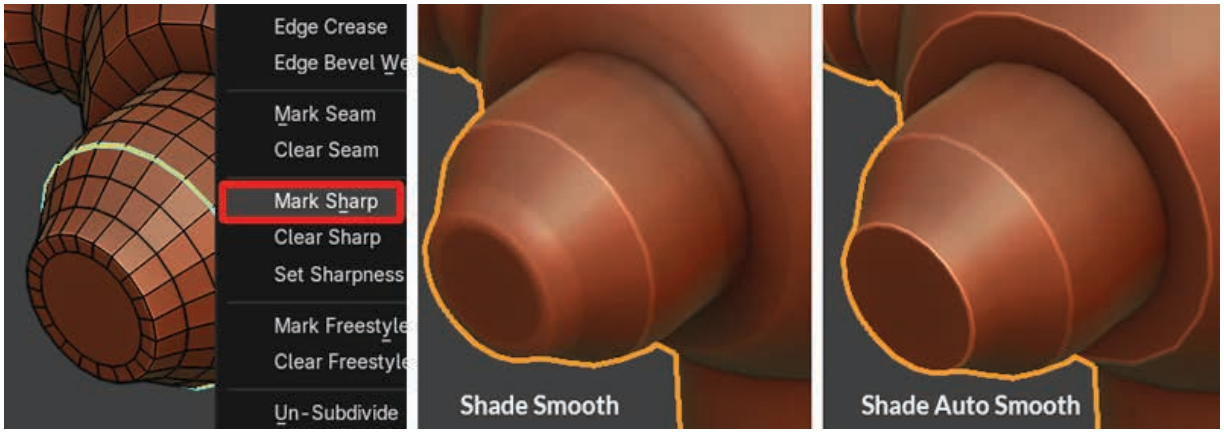
The **Shade Auto Smooth** option has been replaced since Blender 4.1 by **Shade Smooth by Angle**. Then after that, they renamed it to **Shade Auto Smooth** again… but how it works has still changed from what you may have been used to from before. Let's see how it works now.



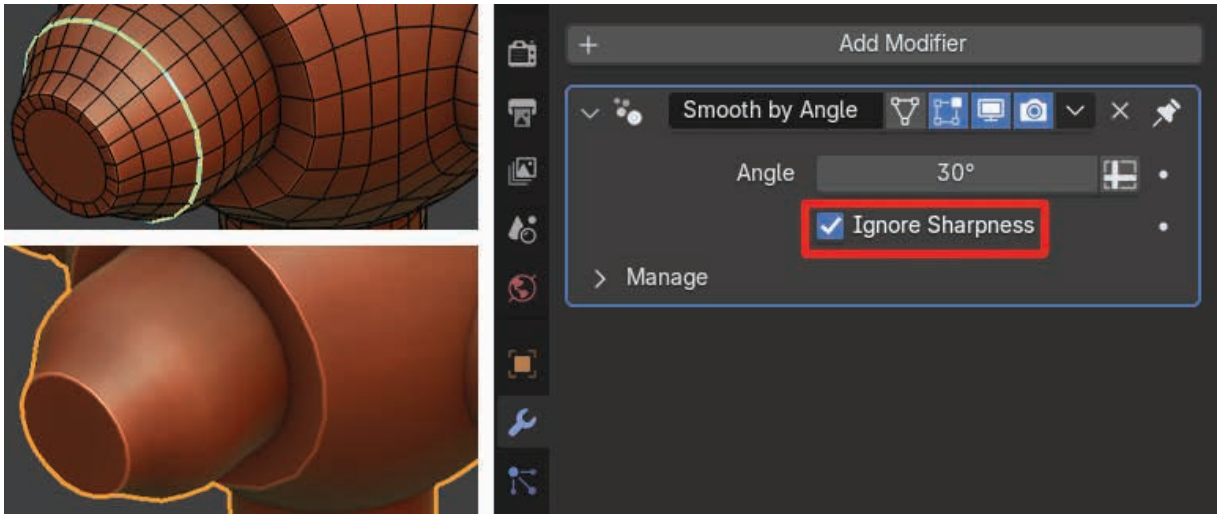
The result is the same, except now a modifier called **Smooth by Angle** is added as well.



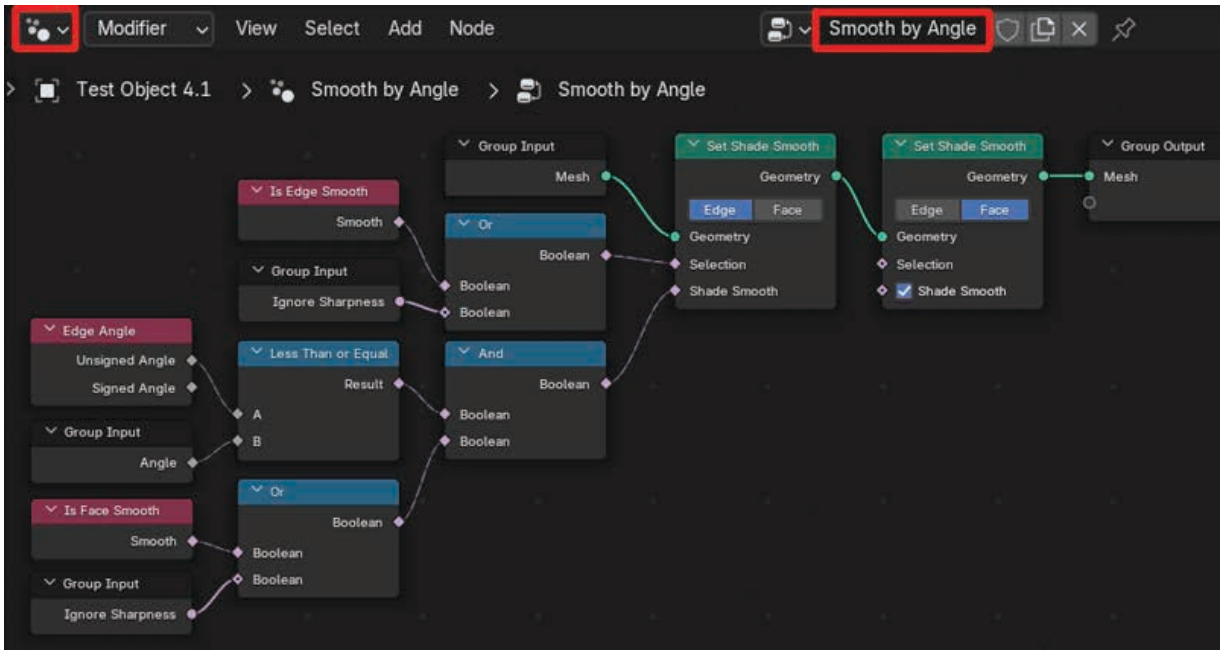
If you mark Edges as Sharp manually, this will work both with **Shade Smooth** and **Shade Auto Smooth**.



However if you check the option **Ignore Sharpness** in the modifier, the Sharp Edges will be ignored. The **Smooth by Angle modifier** makes the process non-destructive. You can choose under which angle things are sharp. Generally the default of 30 degrees tends to be a good choice.

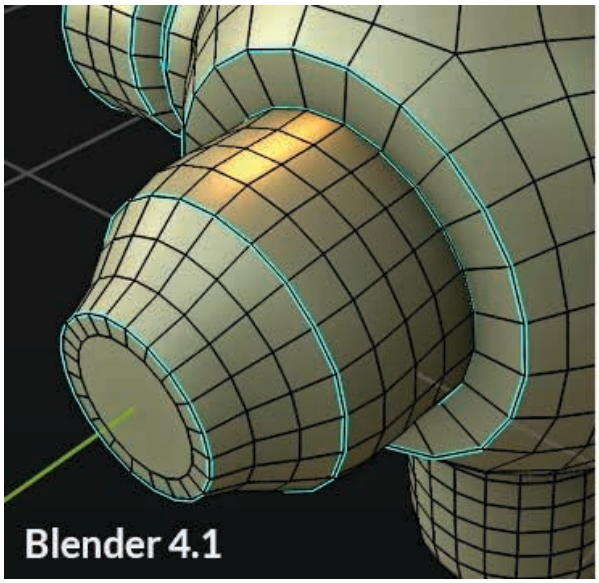


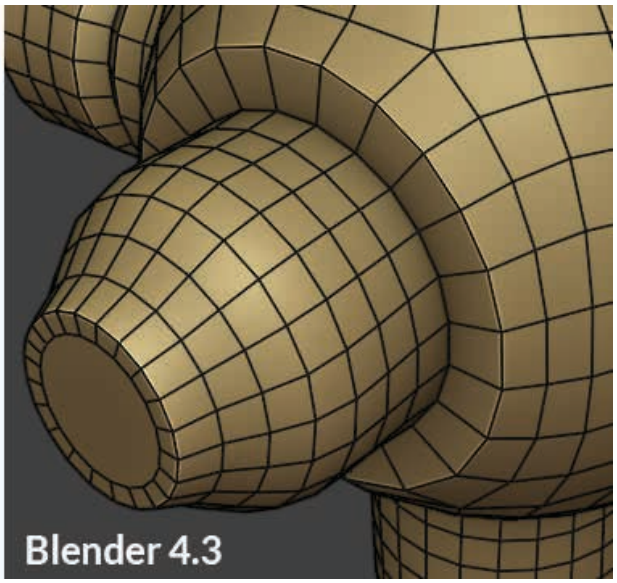
Interestingly, the modifier is made with Geometry Nodes. So in the **Geometry Node Editor**, you can have a look at how it works under the hood.



This modifier replaces the Normals option that we had previously, where you could also set the angle value. So while that option is gone, we now have a more functional non-destructive geonodes modifier based method, for shading auto smooth.

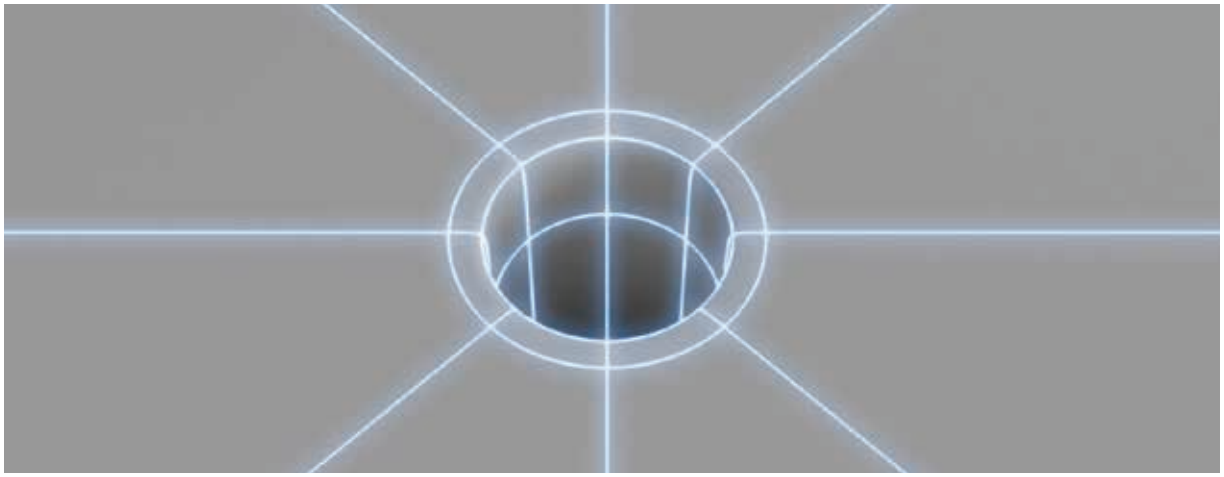
And the (controversial) feature of automatically marking Edges as Sharp when choosing Shade Auto Smooth, which was introduced in Blender 4.1, has been rolled back after Blender 4.2. Making it almost the same as it was before, but Geometry Nodes based.



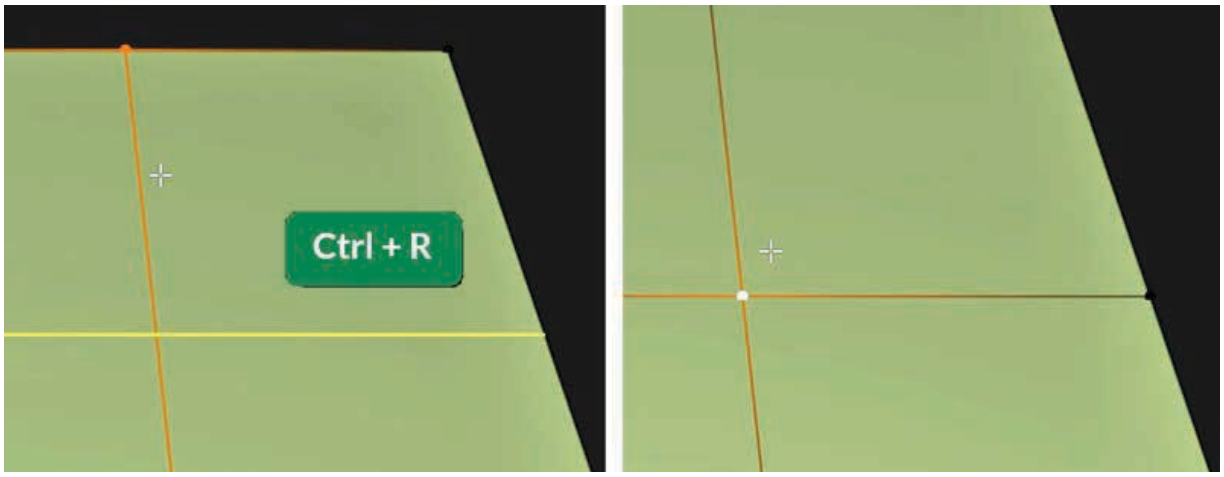


Create a hole from a Vertex

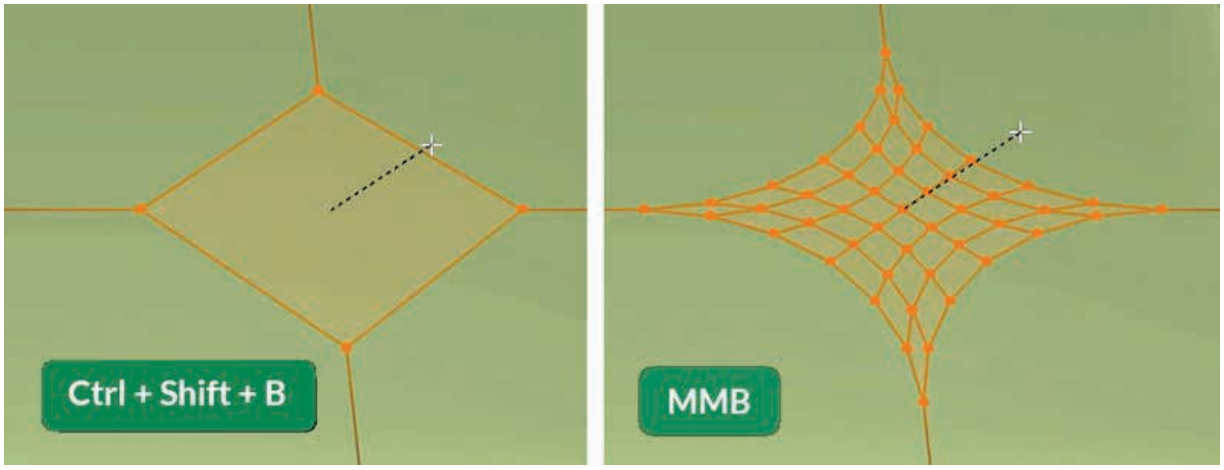
The easiest way to add a circular hole is by beveling a Vertex. Watch a longer video about this topic here.



First add a couple of intersecting Edge Loops with **Ctrl+R**, so that you get a Vertex at the location where you want to place a hole. Of course, in case your geometry already has a Vertex in the location where you want the hole to be, you don't need to do this step.

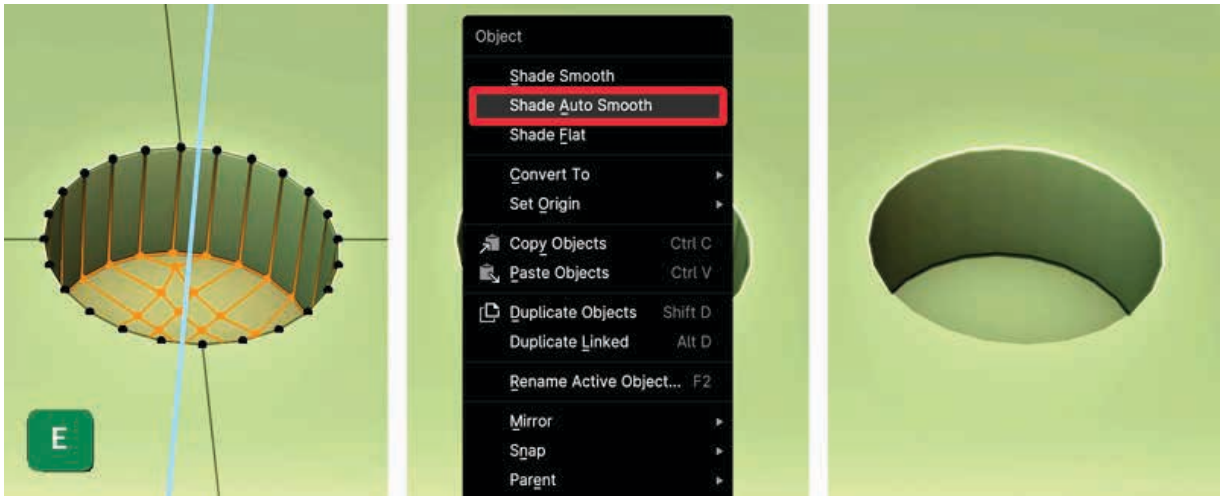


Then select the Vertex and **press Shift+Ctrl+B** to bevel it. Scroll the Middle Mouse Button up to add Segments.

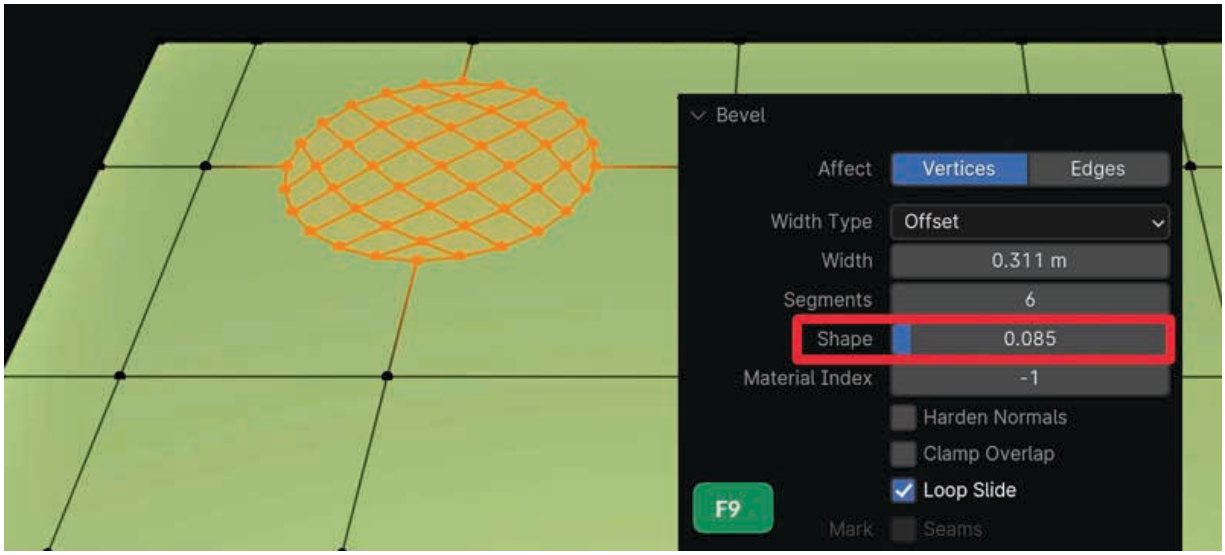


The easiest way to make it circular is by going to **LoopTools > Circle**. If you don't have these options, make sure LoopTools is enabled in Preferences > Add-ons.

Then you can **press E** to extrude it inwards. Right-click and choose Shade Auto Smooth to make it look smooth.

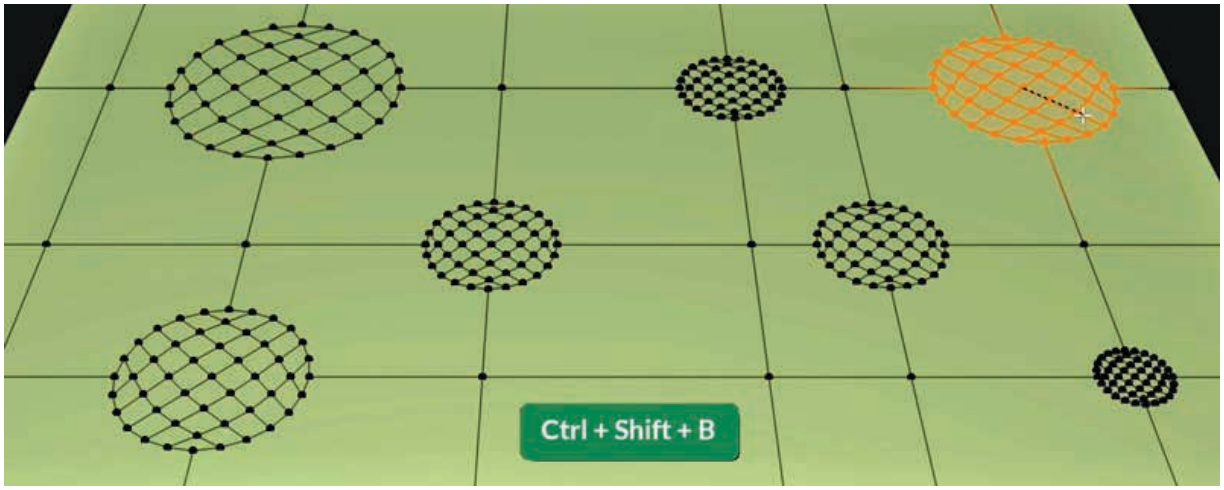


Instead of using LoopTools to make it circular, you can also adjust the **Shape** value in the last operator panel. A value of **0.085** seems to be the value closest to a circle, although it's not 100% perfectly circular. This panel appears in the lower left corner, but you can place it anywhere (and get it back if you accidentally close it) by pressing F9.



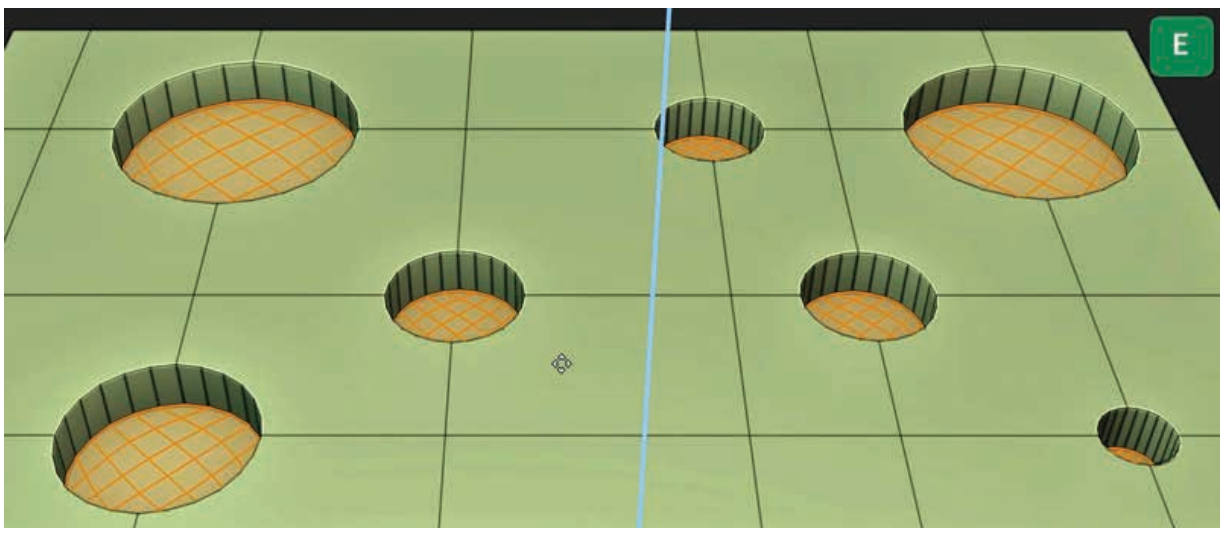
Click here to return to the index.

The advantage of this is that Blender remembers that value for each subsequent bevel, so that can save some time. The values are reset to default, the next time you start Blender.

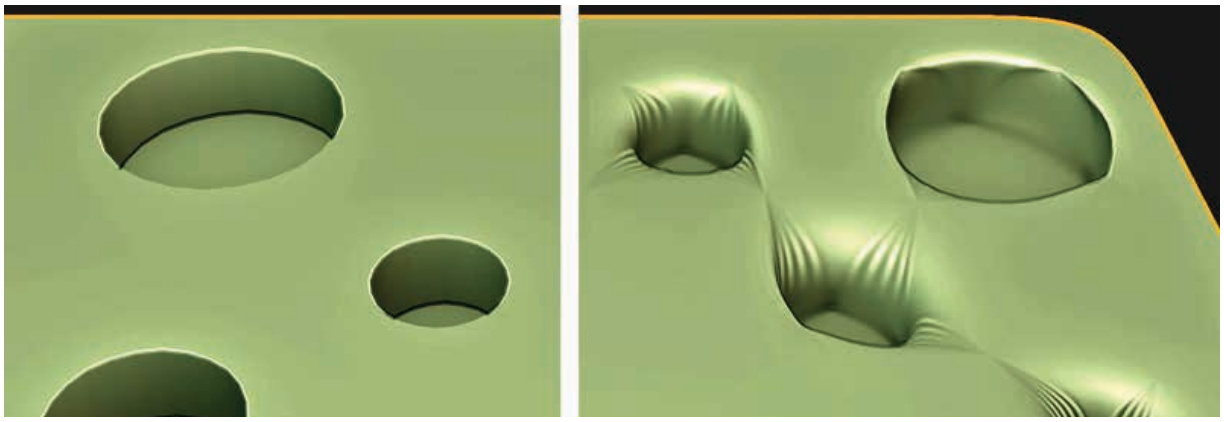


To select all the other circles, first go to **Face Selection** mode. Press **Shift+G** for the Select Similar menu and choose **Face Regions**.

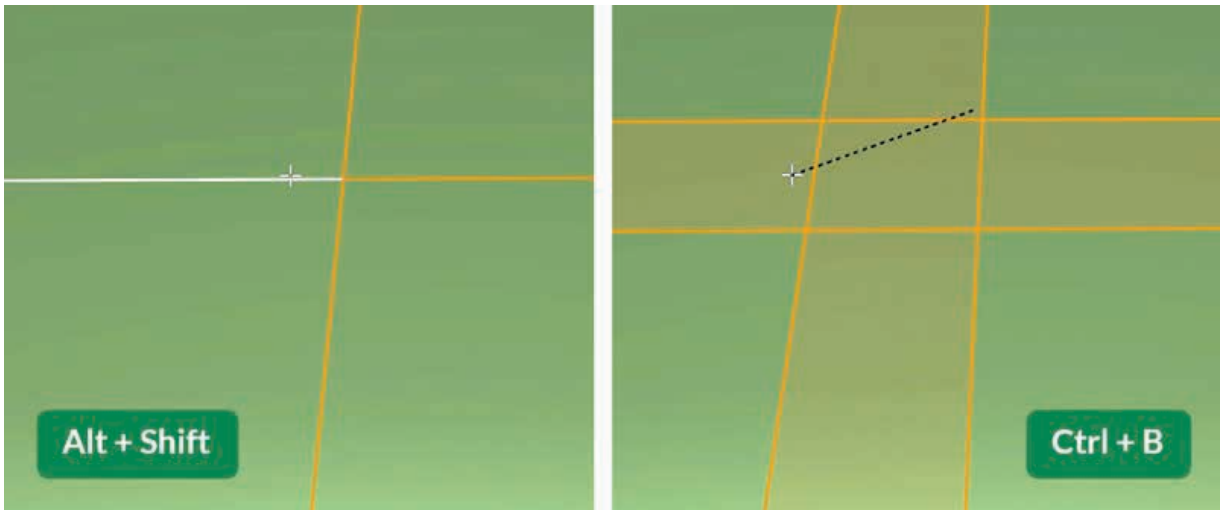
Then you can extrude them all at once by **pressing E**.



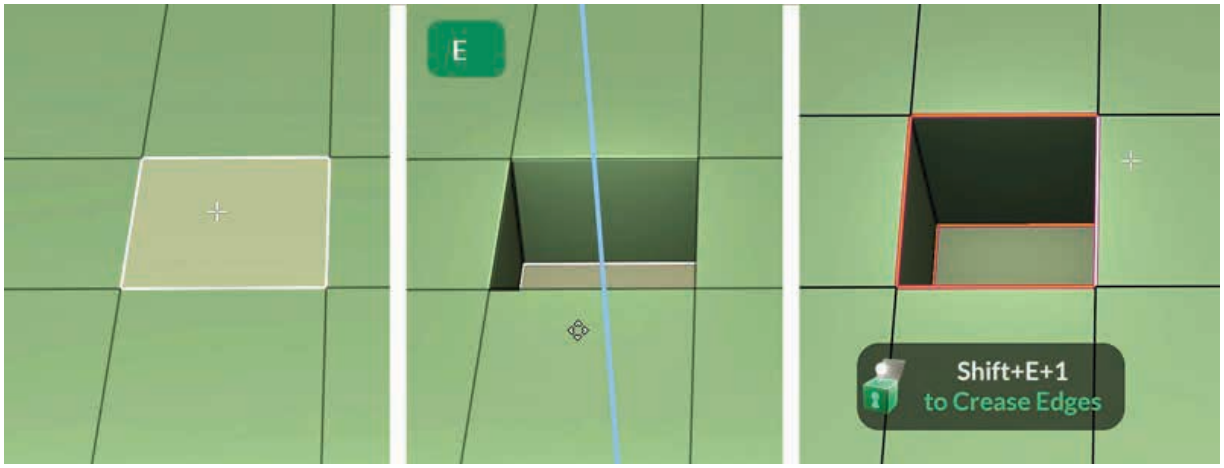
Shade Auto Smooth to make it look better. This is fine for a model that doesn't get deformed or subdivided, in a workflow where you use Shade **Auto Smooth** *(below left)*. But once you **subdivide** it, you get this kind of mess *(below right)*. It could be interesting for some kind of cloth object, but in this case it's not what we're after.



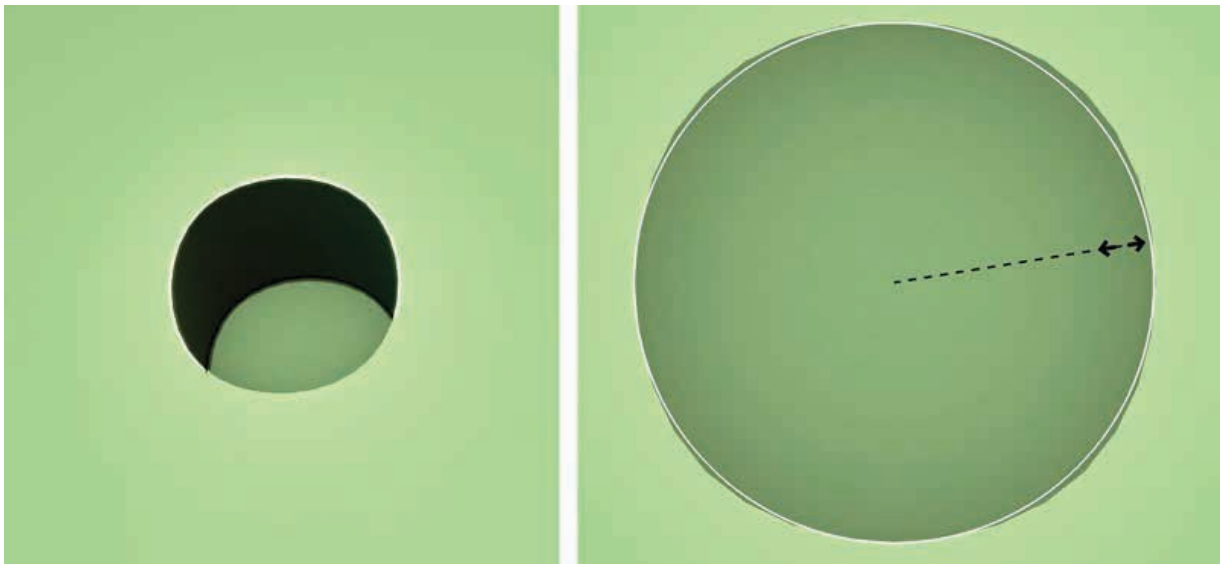
To get a hole you can use in a subdivision workflow, you'll need more Edge Loops. First, select both Edge Loops by **holding Alt and Shift** and clicking on them. Bevel the selected Edges by **pressing Ctrl+B**.



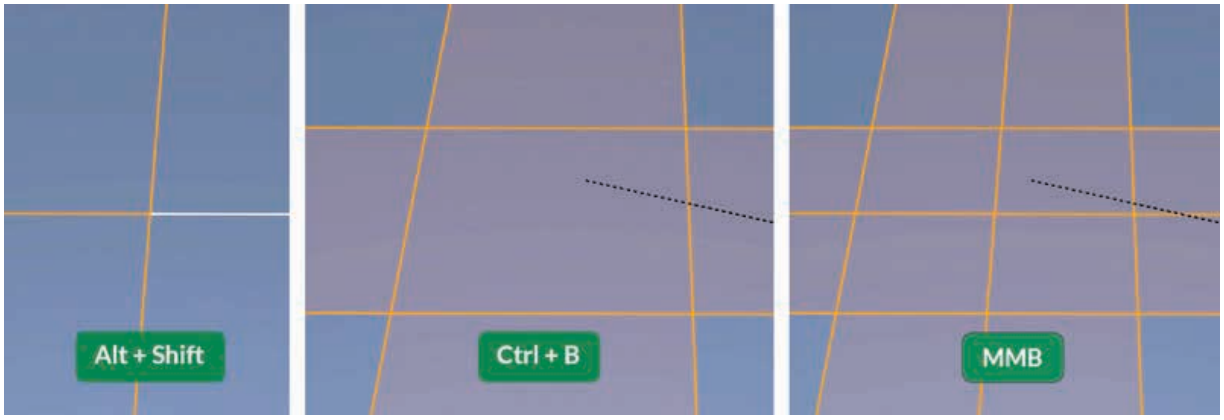
At the very least you'll need two Edge loops intersecting so that you get one Face. This Face can then be extruded to create a simple hole. We can hold Shift and select the edges that we want to keep sharp and Crease them with **Shift+E+1** on the Numpad.



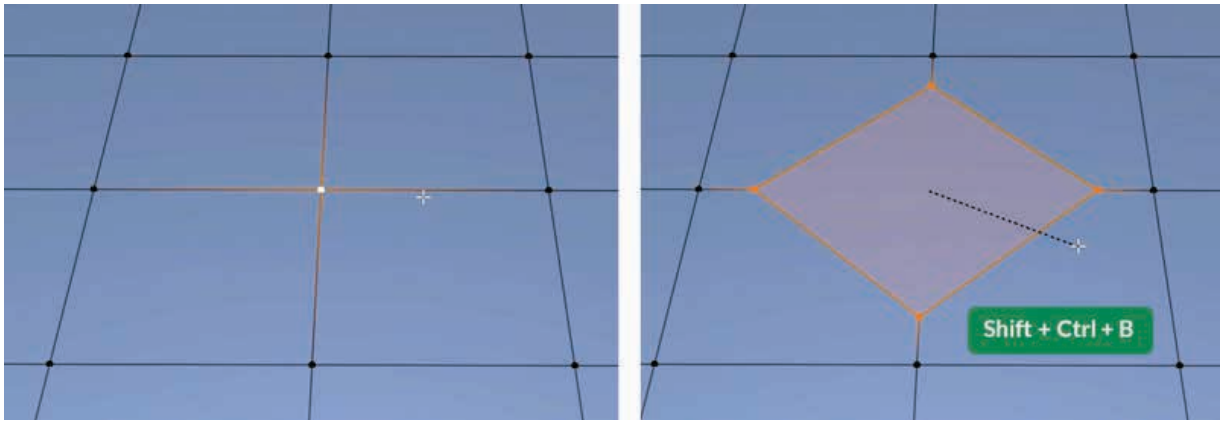
If you add a Subdivision modifier now, it will look circular enough. However if we compare this to a Circle (Shift+A > Mesh > Circle), we can see that it's not 100% circular. If you want it to be perfect, you'll need more geometry than this.



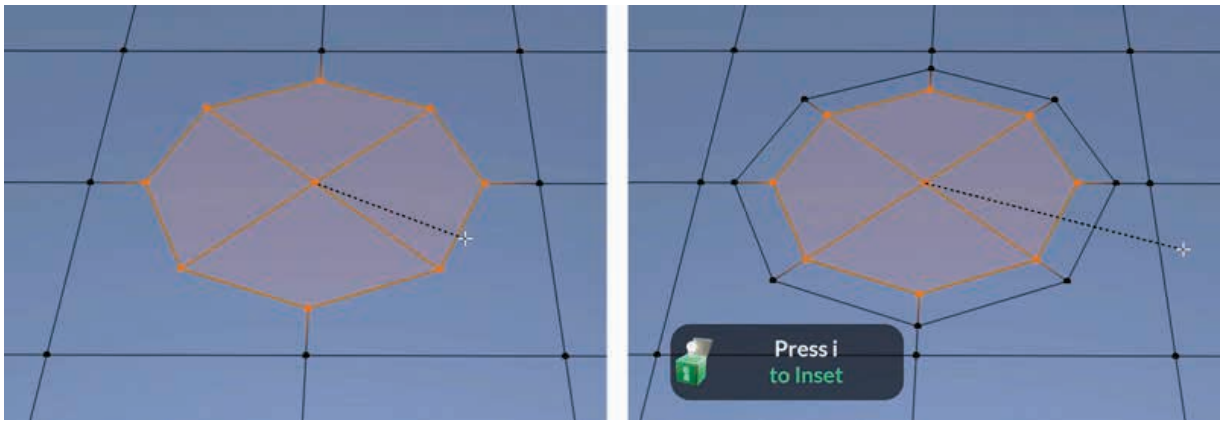
Let's look at a method with slightly more geometry. As before, starting with a Plane **press Ctrl+R** to add some Edge Loops. **Hold Alt and Shift** to be able to select multiple Edge Loops. For a perfectly circular hole we'll need to add one more Edge Loop when beveling these loops. So bevel with **Ctrl+B** and then scroll the mouse wheel up to add an additional loop.



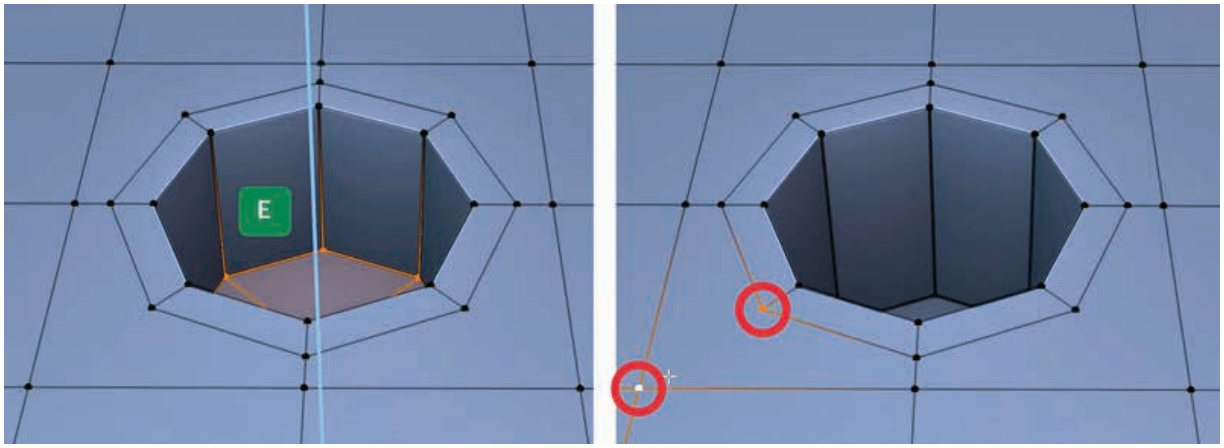
Switch to **Vertex selection mode**. Then we can select and bevel this Vertex in the center with **Shift+Ctrl+B.**



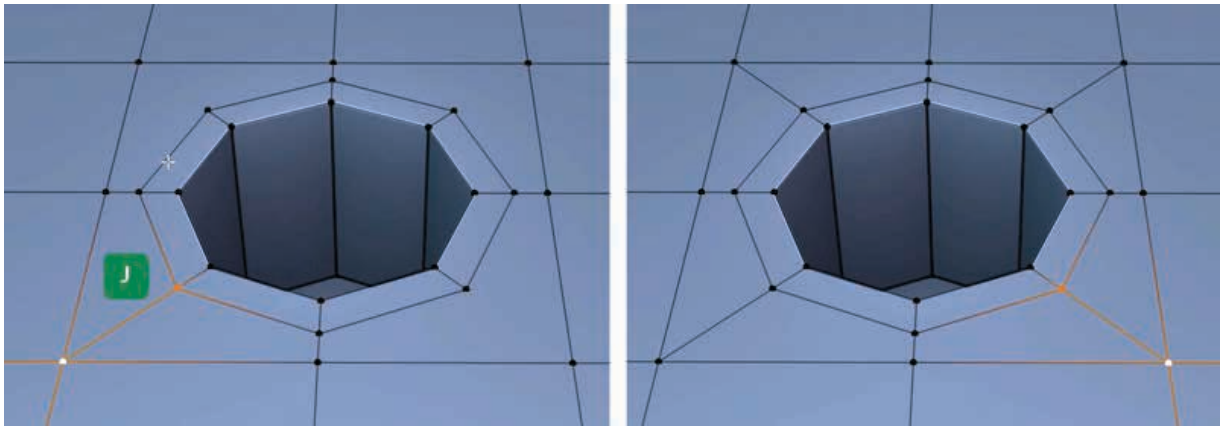
Scroll the mouse wheel up to add one more segment. Blender remembered my **Shape** value from the previous time I beveled a Vertex. Otherwise you can use **LoopTools > Circle** to get the circular shape. I'll inset this with **i** to add a loop of Faces to support the geometry around the hole.



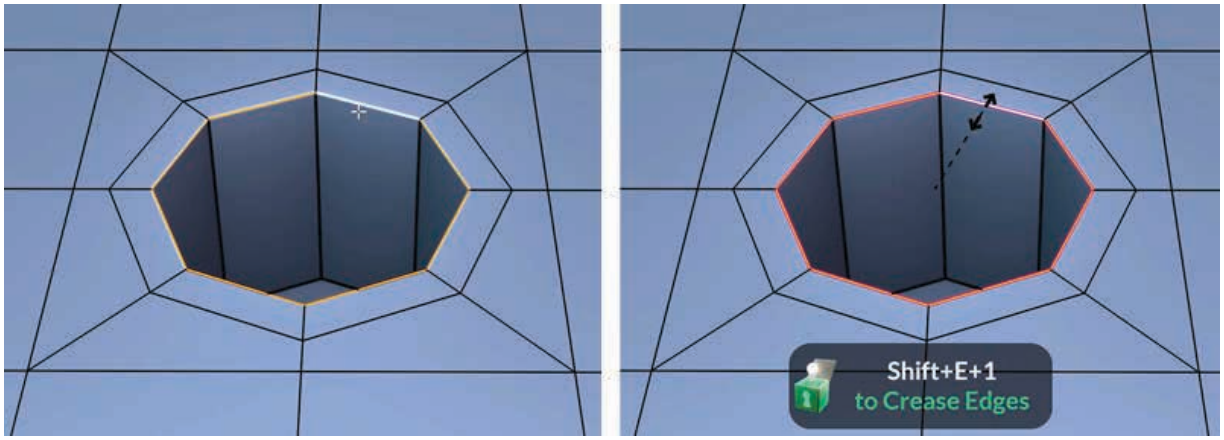
And now we can extrude that down. To get all quads, we can connect these Vertices by selecting them and **pressing J**.



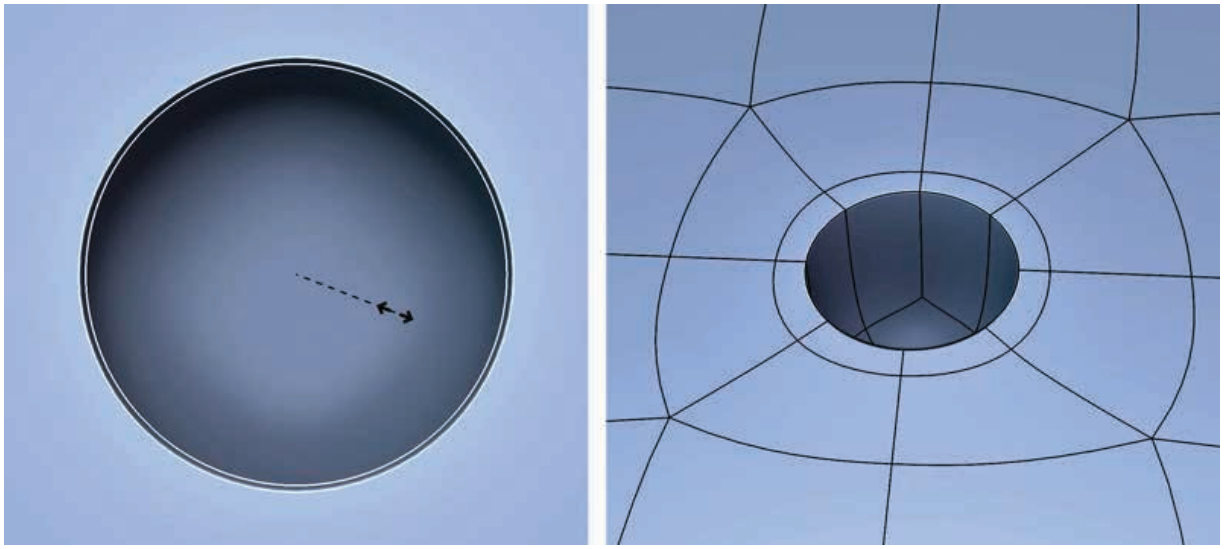
Repeat for the other Vertices until they are all connected and there are no N-gons, only Faces with 4 Edges ("Quads").



And we can **crease** these edges to keep them sharp.



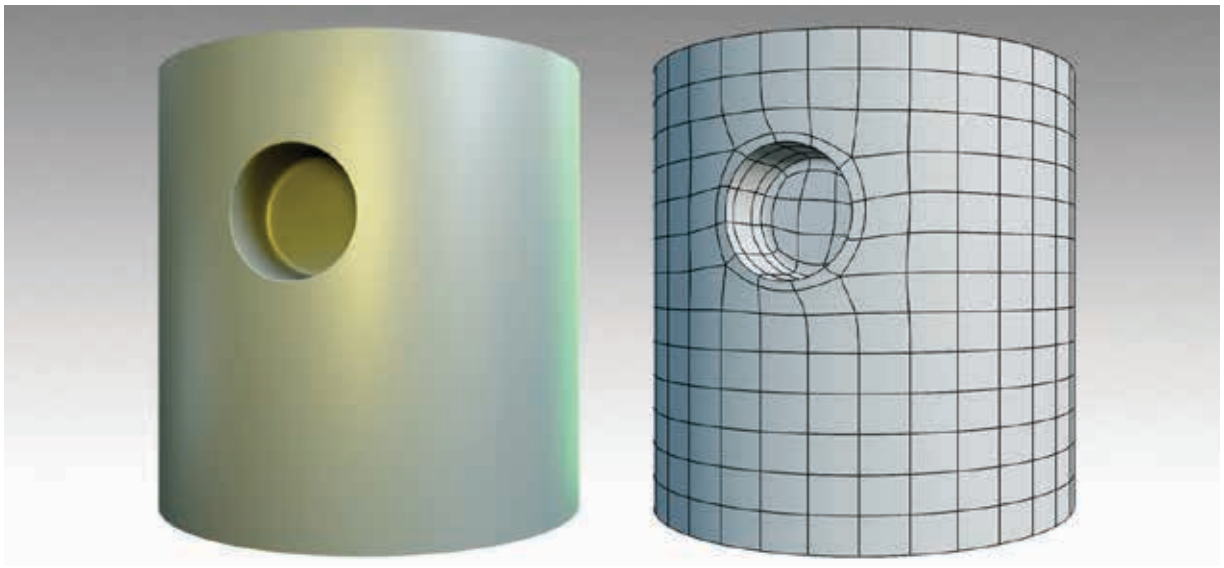
Press Ctrl and 2 (or 3, depending on how much geometry you want) to give it a Subdivision modifier. After subdividing, the hole is perfectly circular. We can verify the shape by comparing it to a Circle (Shift+A > Mesh > Circle).



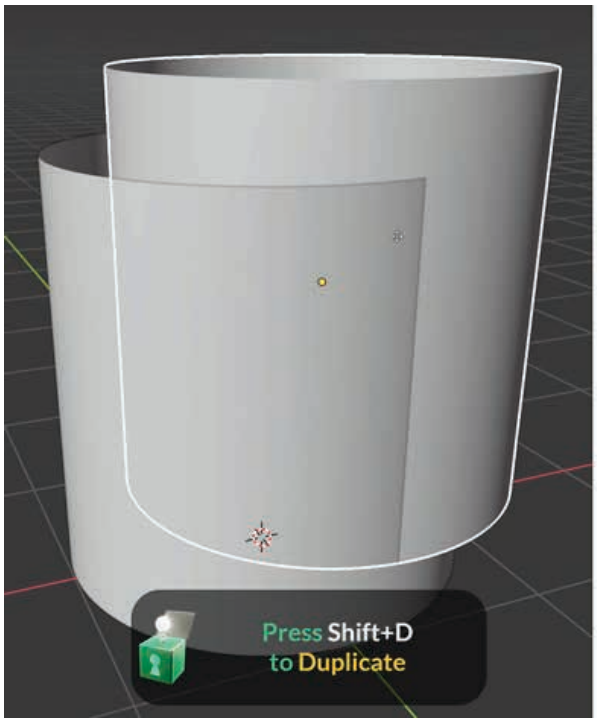
And that's how you make a perfect quad-topology hole from a Vertex.

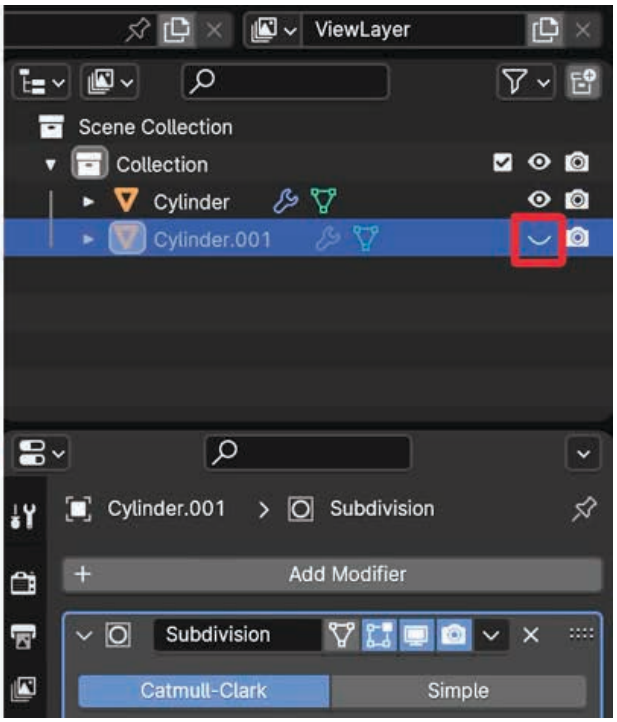
Perfect Holes with Quad Topology in Curved Surfaces

This technique is useful for when you want a perfectly circular hole in a warped or curved surface. To watch a longer version of this topic, click here.

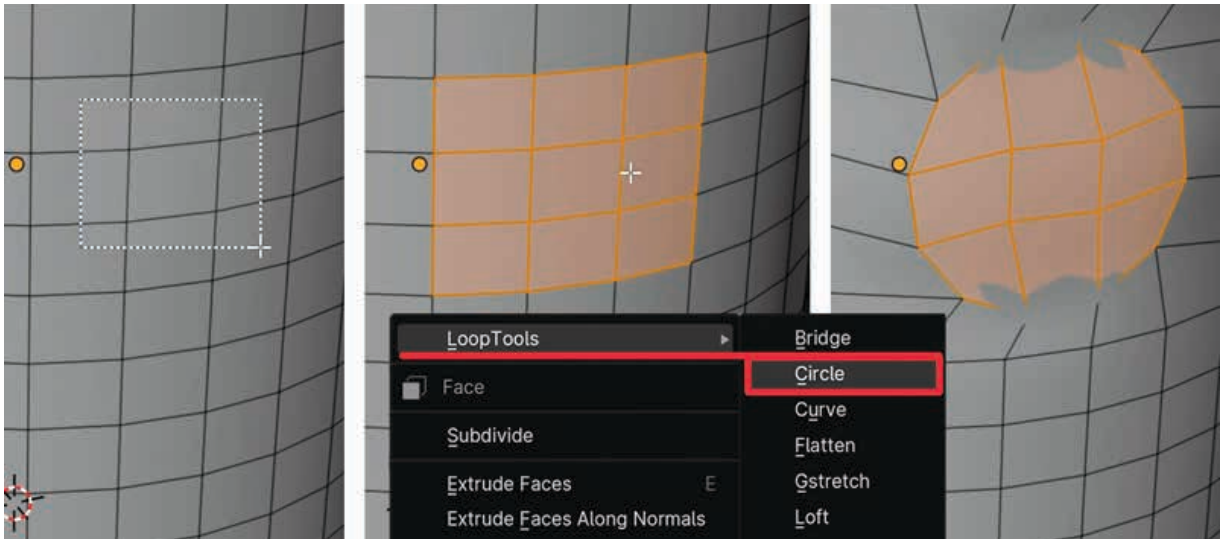


Duplicate the surface you want to add a hole to with **Shift+D**, and right-click to cancel the transformation. It's important that the duplicated Cylinder is in the exact same location as the original one. Disable the visibility of the duplicate in the Outliner.

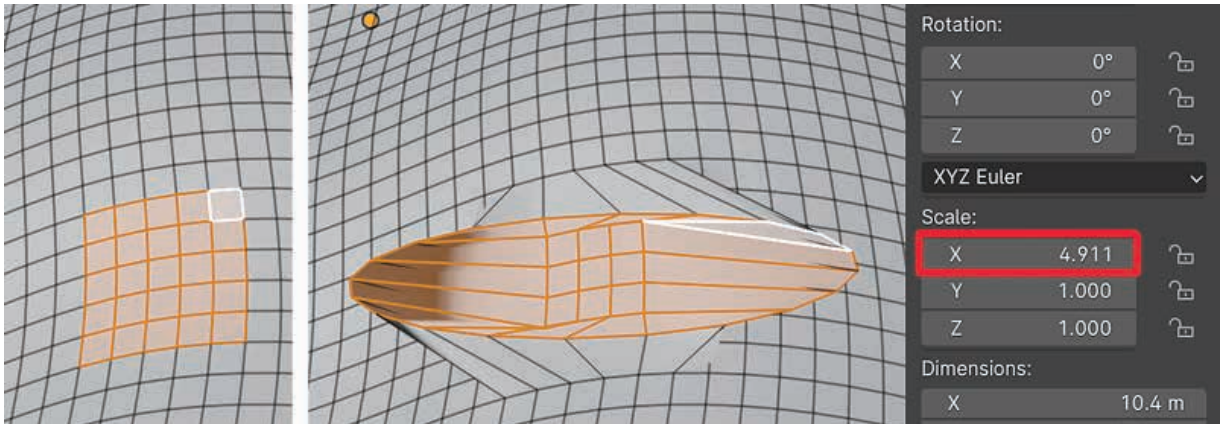




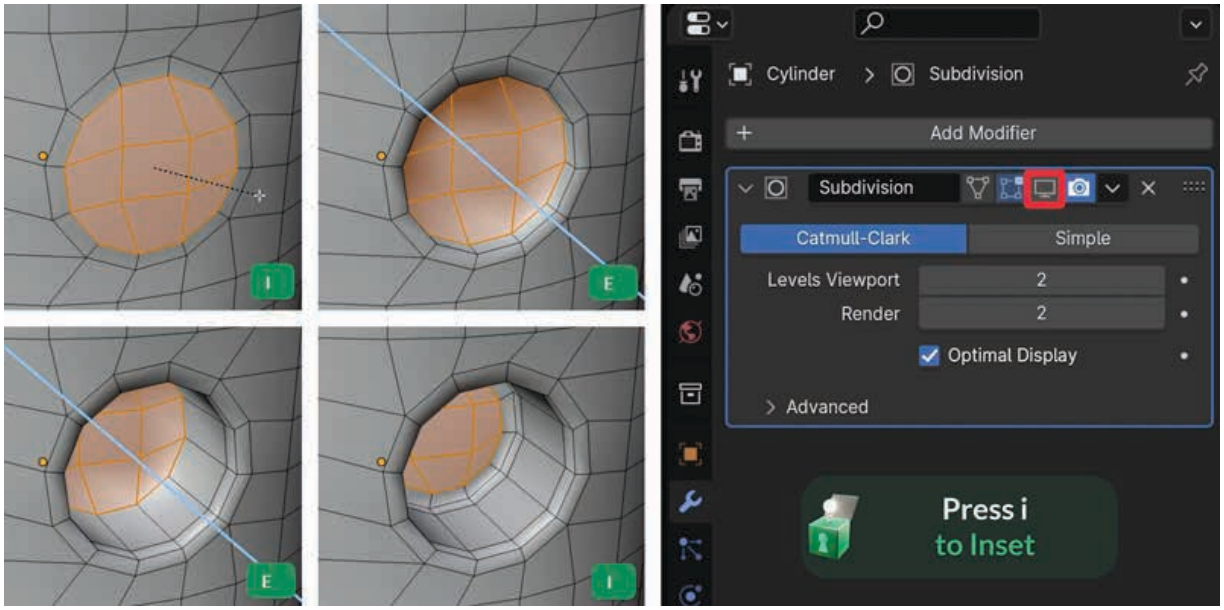
Select some Faces and make them circular with **LoopTools > Circle**. Note, that you'll have to enable LoopTools in Preferences > Add-ons if you haven't already.



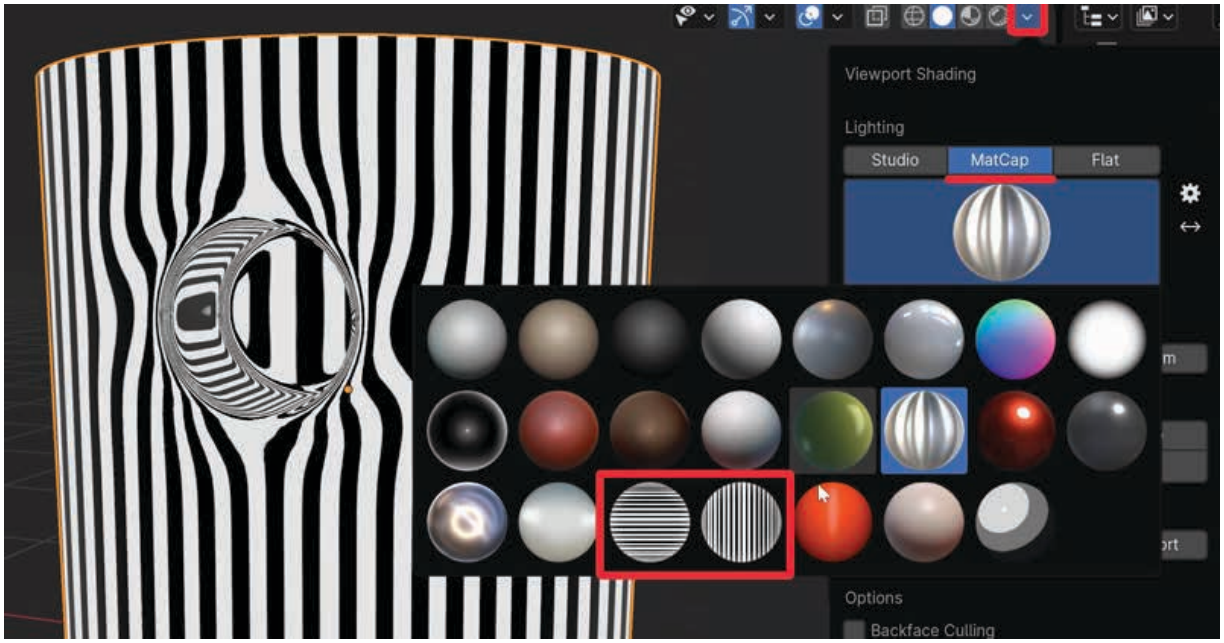
Note, that if you get a weird stretched circle, that means the scale of your object in uneven. You can fix this by selecting the object in Object Mode, pressing Ctrl+A and choosing Apply Scale.



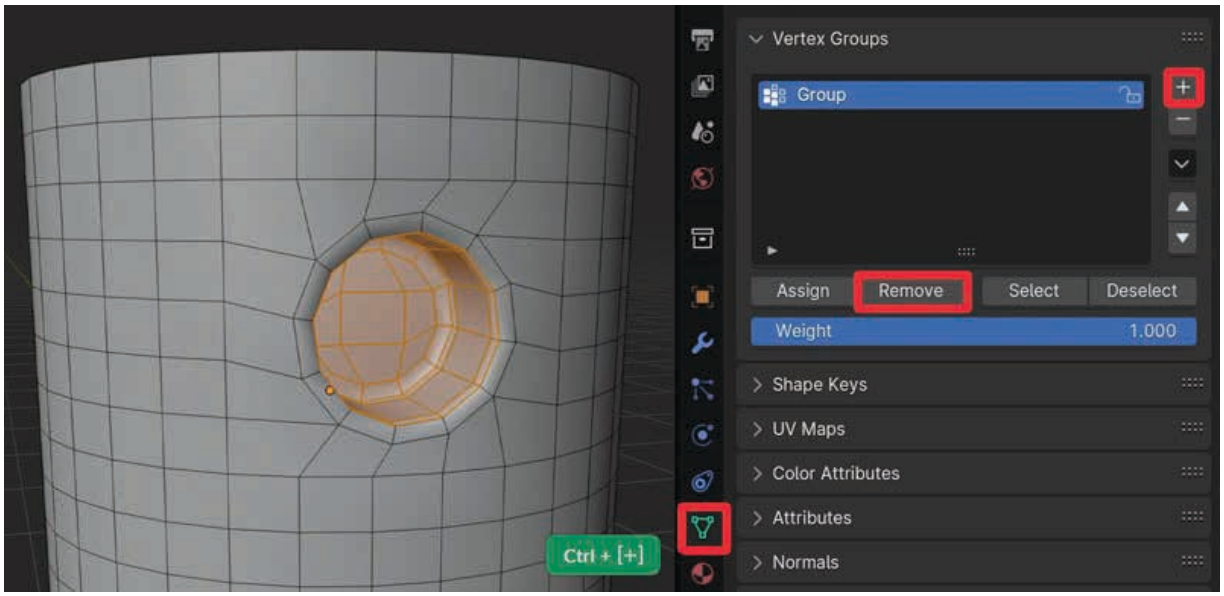
Inset (i) and **extrude (e)** to create a hole with some boundary protection loops. Alternatively, you can crease the edges you wish to keep sharp with **Shift+E+1**. By turning off the Subdivision modifier temporarily, you can see better what you're doing with the geometry.



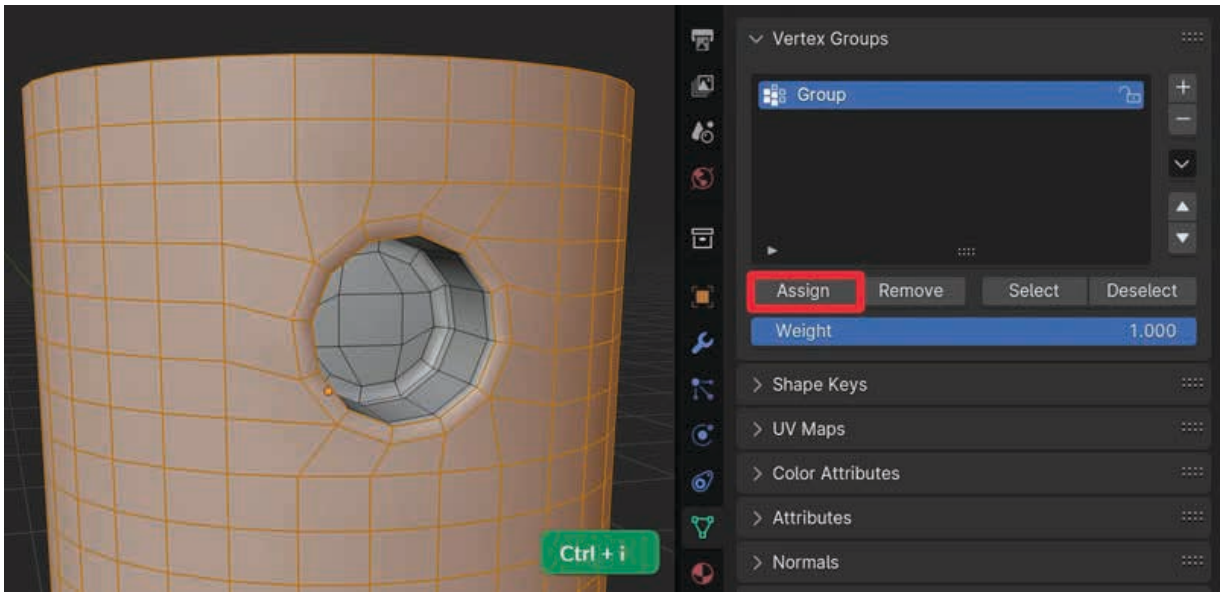
Using a **Matcap** it's clear, that the surface around the hole is quite 'lumpy'. Especially these two matcaps that have black lines, are useul for spotting uneven surfaces.



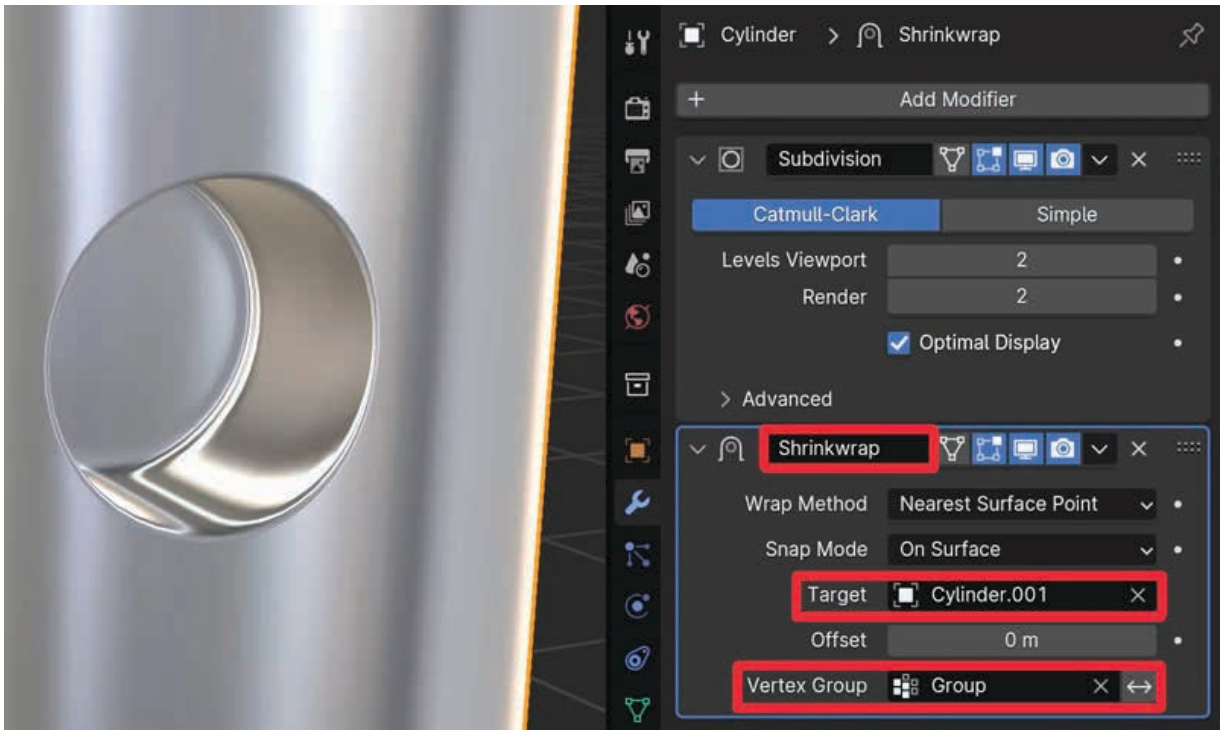
Increase the selection to include the entire hole, then make a new **Vertex Group** by clicking on the + icon. You can also p**ress Ctrl+G** to create the Vertex group. Then click on **Remove**, to give that part a "Weight" of zero in the Vertex Group.



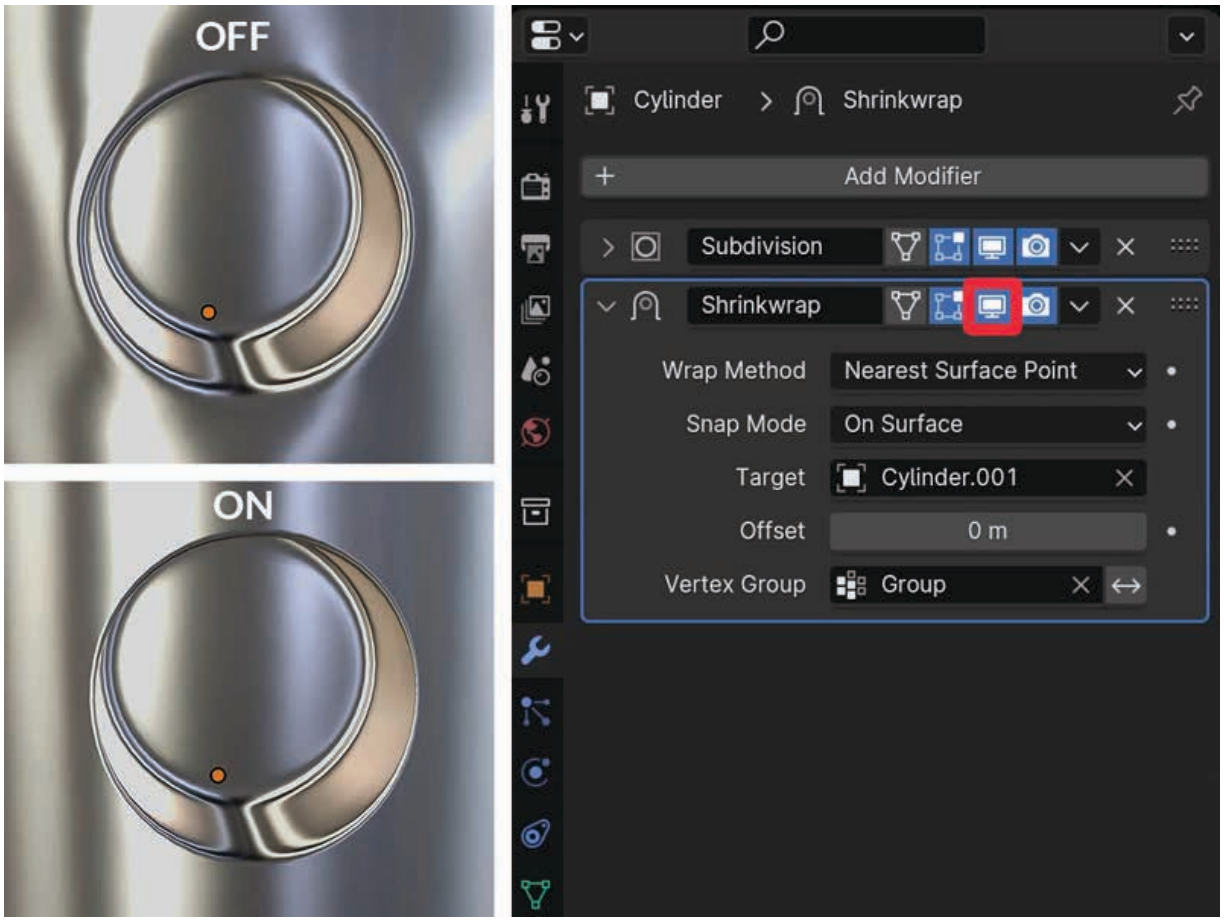
The rest of the Cylinder should have a Weight of 1, so invert the selection with **Ctrl+i**. And then click on **Assign** to give it a Weight of 1.



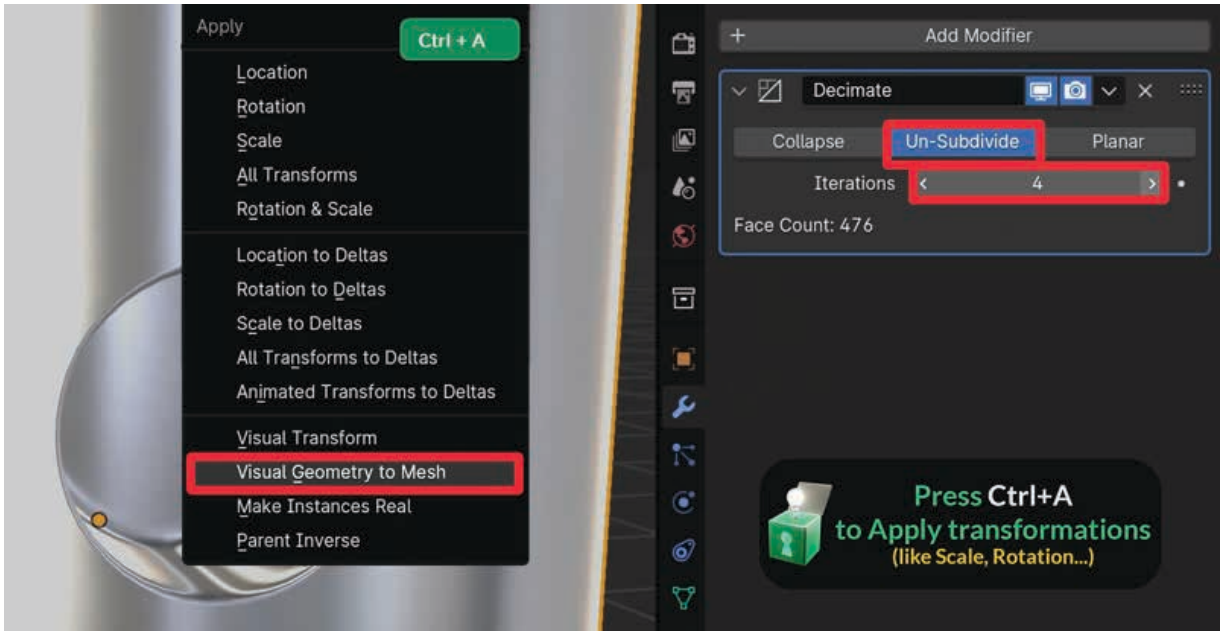
Add a **Shrinkwrap modifier** with the duplicate (Cylinder.001) as its **Target**. It will look weird, until you use the **Vertex Group** ("Group") to exclude the hole. Then it will look perfect.



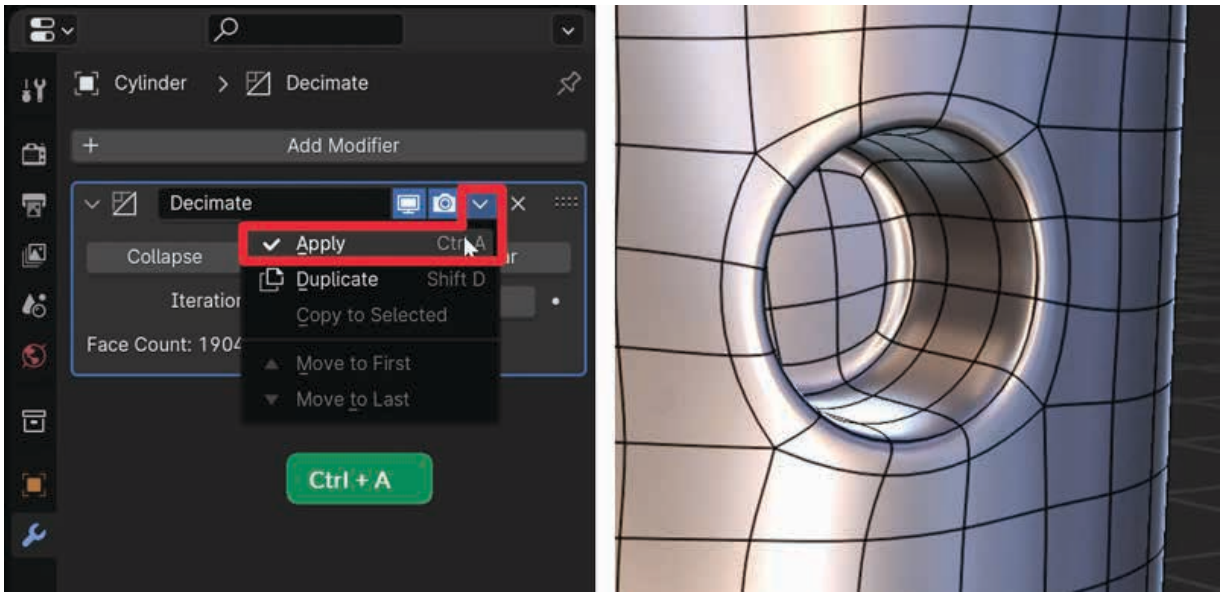
If you turn the Shrinkwrap modifier on and off, you can see the difference it makes. Of course, in some cases having a surface that isn't perfect can actually be desirable. In the real world, surfaces are sometimes imperfect. That's something to consider.



Press Ctrl+A and choose **Visual Geometry to Mesh**, to apply all the modifiers at once. Then add a **Decimate modifier** set to **Un-Subdivide**.



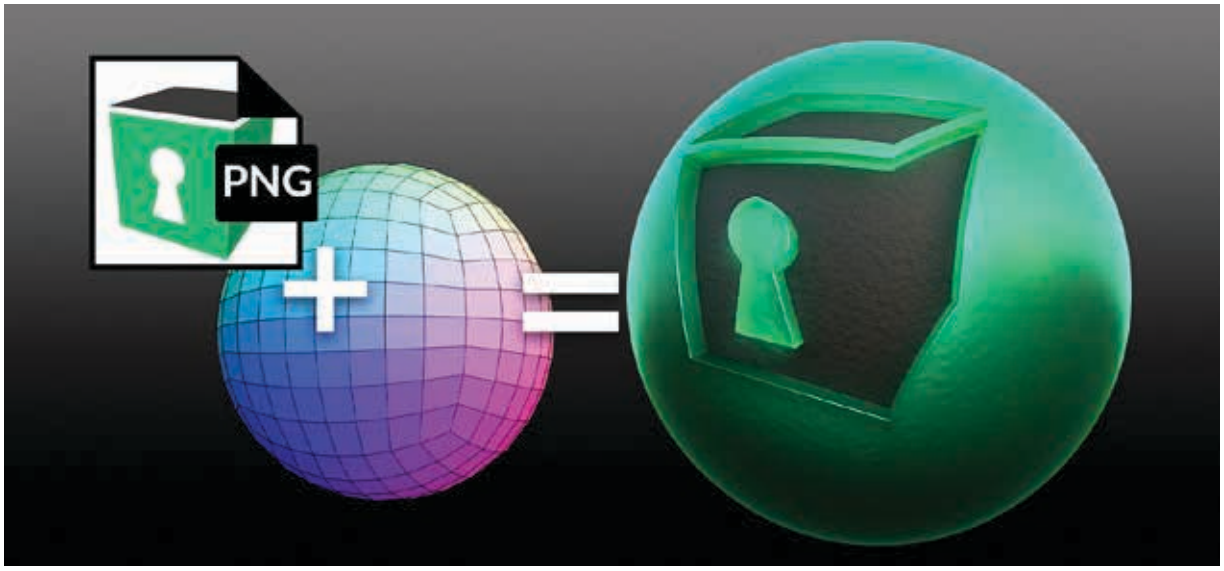
Increase the **Iterations** to an even number and apply the modifier.



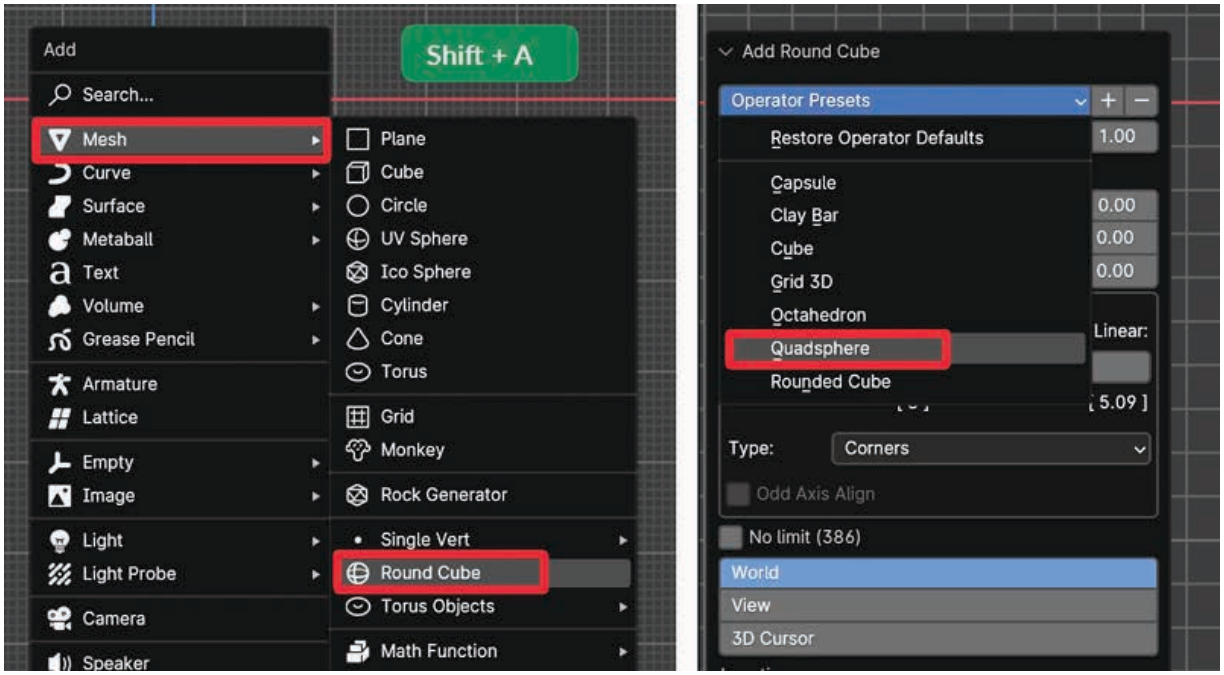
Now you have a perfect, quad-topology circular hole in a Cylinder.

Image file to geometry using Knife Project

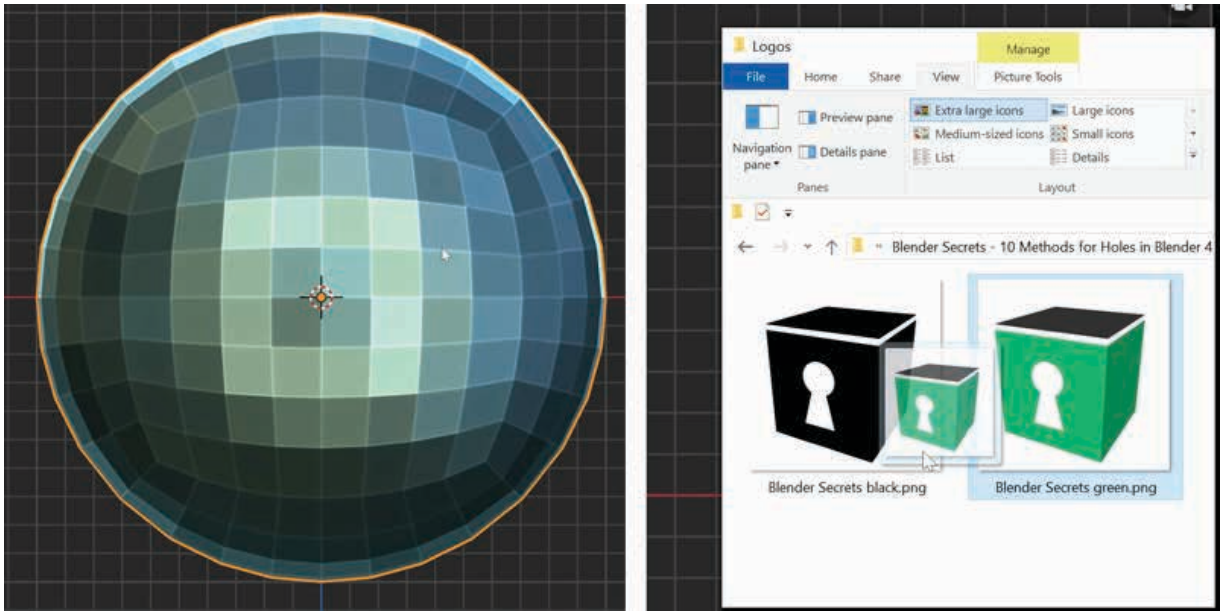
Knife Project is another interesting tool for making holes. You can watch a longer video about this topic here.



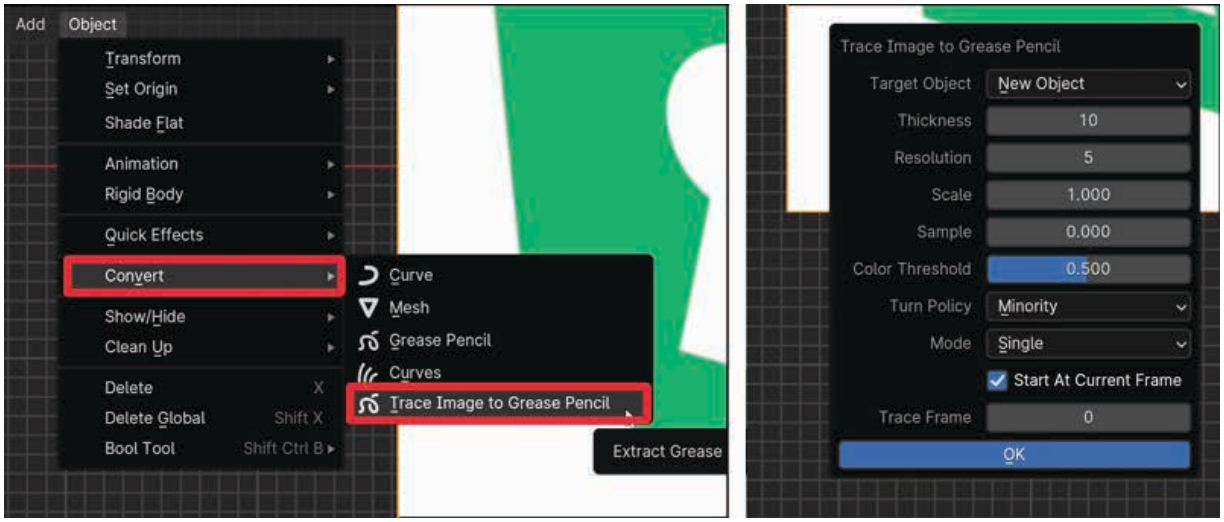
To demonstrate it, I'll use a **Round Cube**. This is the **Quadsphere** preset. Note, that you won't have these extra objects unless you enable the Add Mesh: Extra Objects add-on in Preferences.



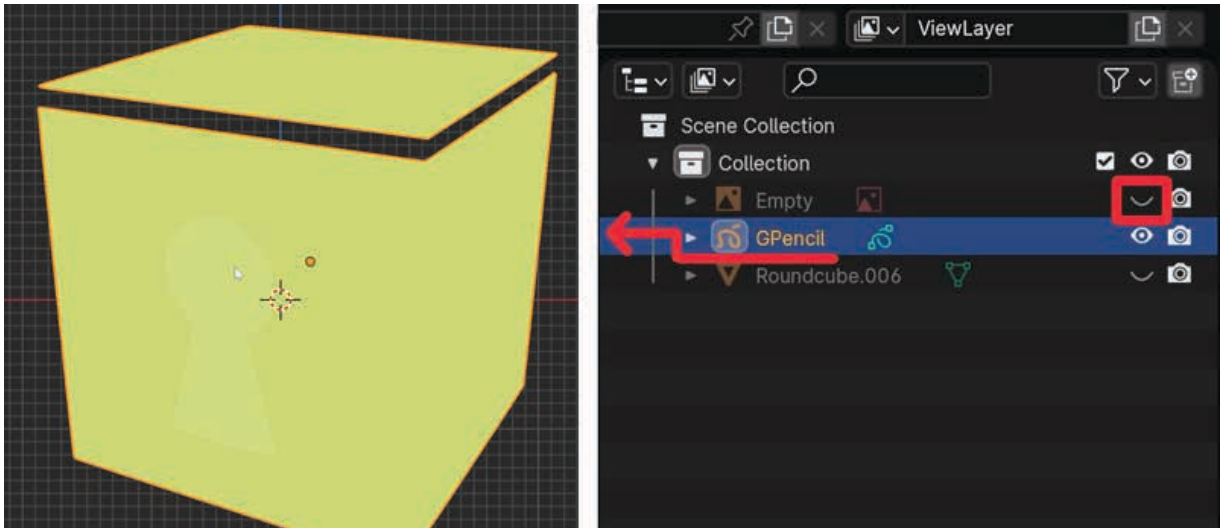
Hide the Quadsphere for now by **pressing H** in Object Mode. Next we'll import a logo image. You can just **drag and drop** images from a file explorer window onto the 3D viewport. This image is imported as an Empty object.



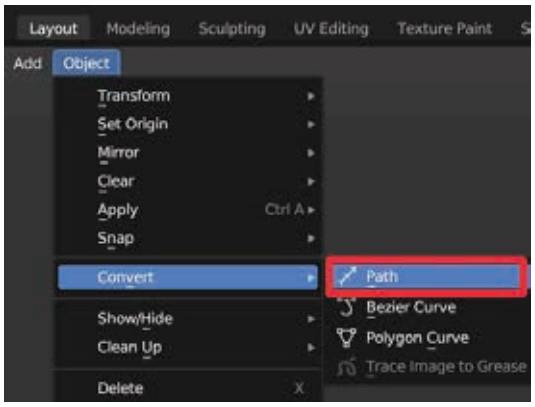
Convert it to a **Grease Pencil** object by going to this menu. I recommend the default Resolution, which is 5.



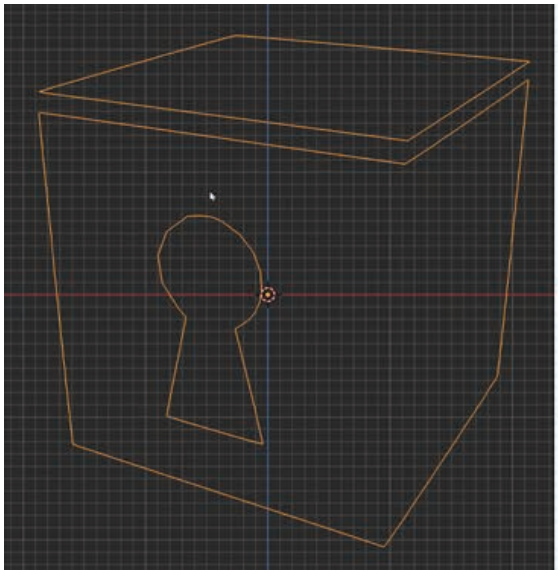
This creates a new Grease Pencil object, or **GPencil** as it's called in the Outliner. We can hide the original image (called "Empty" in the Outliner). It's a bit difficult to see, but it also kept the keyhole of the logo.

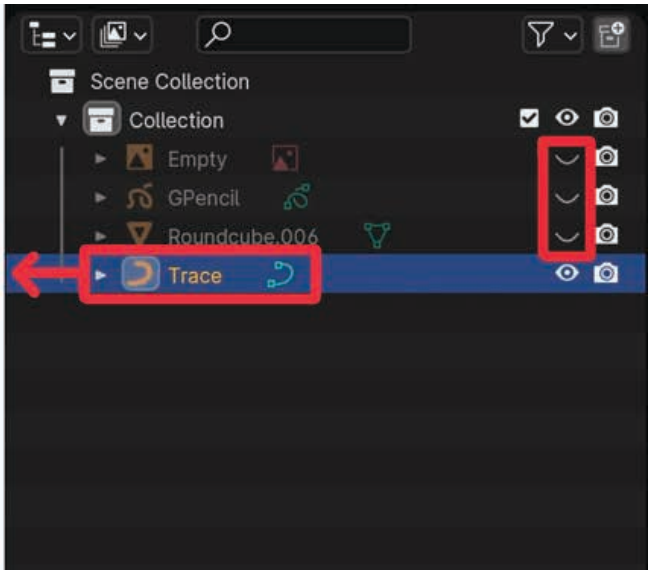


But we actually still need to convert this to a Path. Make sure the GPencil object is selected, and go to **Object > Convert > Path**.

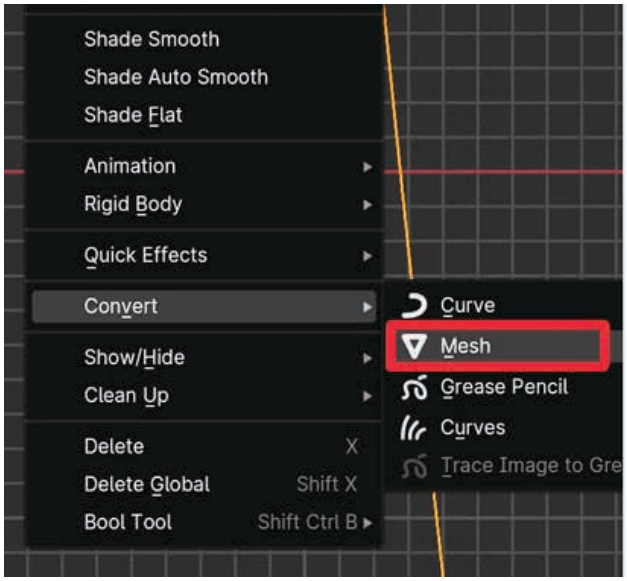


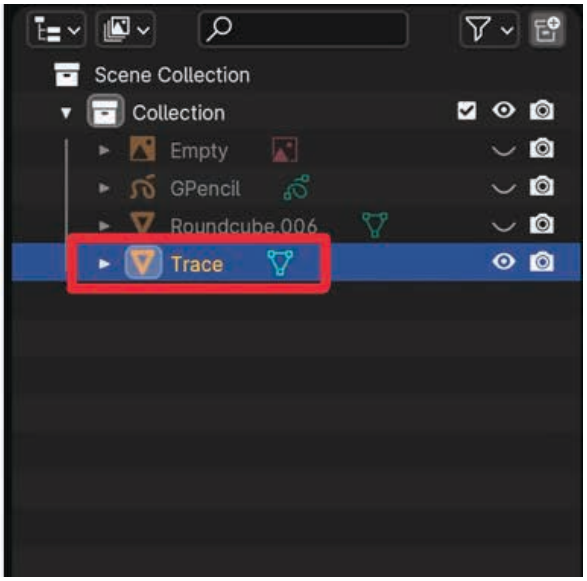
If we disable the Grease Pencil object, we can see the Path. Confusingly, it's called "**Trace**". You can see from the icon that it's a Path object. Everything else is hidden.





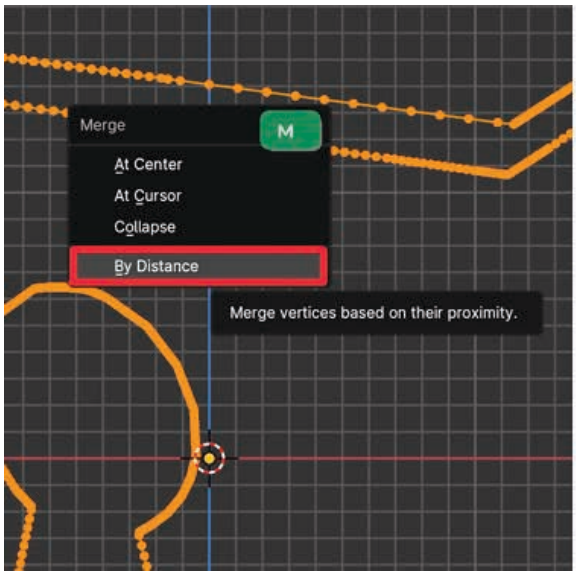
Finally, we convert it to a Mesh by going to **Object > Convert > Mesh**. I hope in the future we can convert directly from Grease Pencil to Mesh. I recommend renaming this mesh to "Logo" or something else that is clear to you. Everything else is hidden.

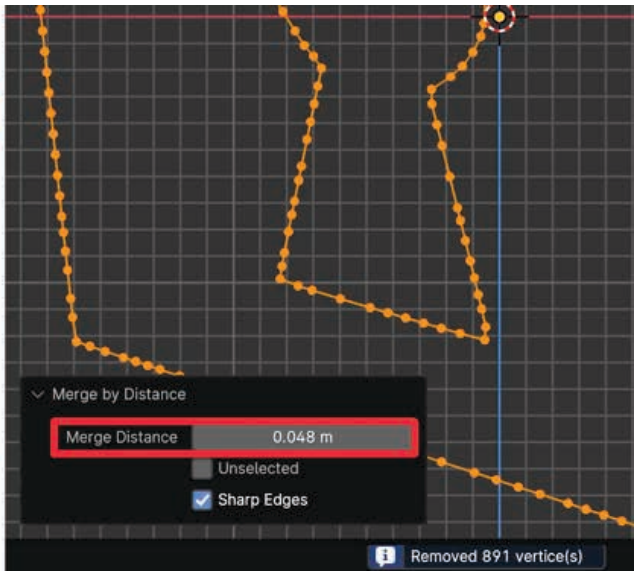




Click here to return to the index.

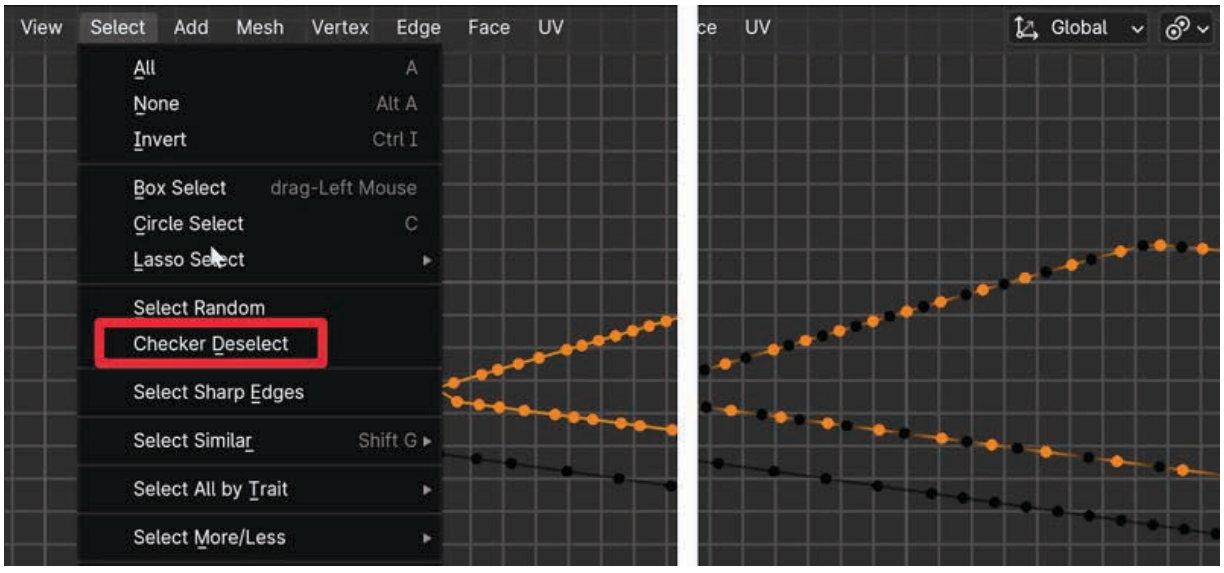
Now as you can see, this has a lot of Vertices. That will make it difficult later to get a nice smooth result when we cut into the Quadsphere. So what we can do is select all the Vertices and **press M** for the Merge menu. Then Merge by **Distance**, and experiment with the Distance value.



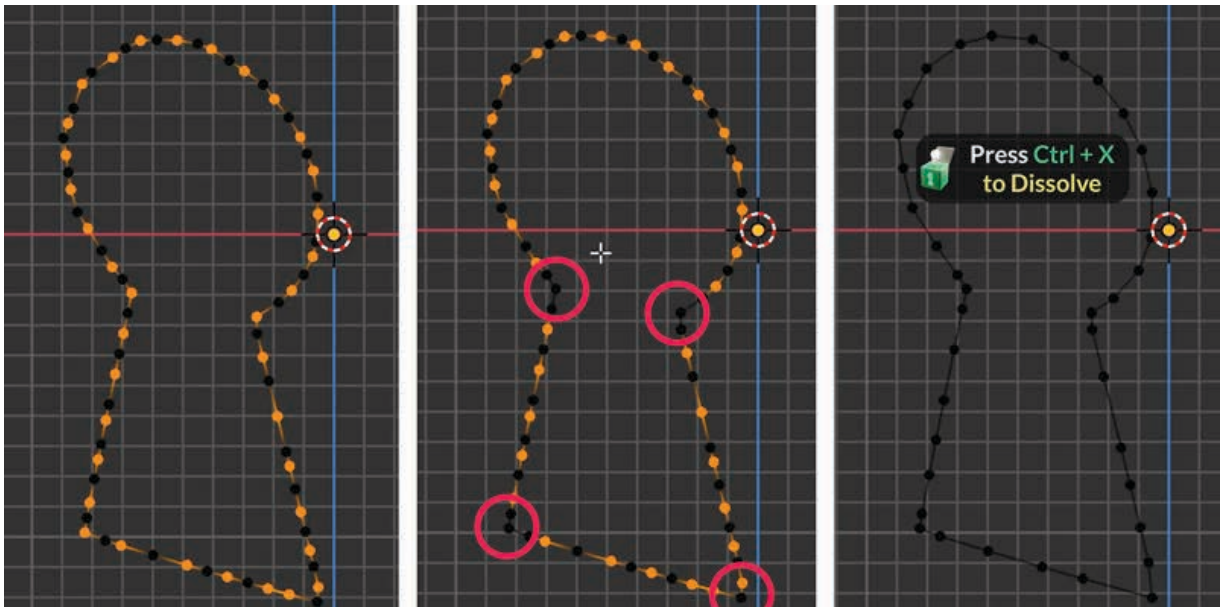


This is about as far as we can go without destroying the shape too much. The value you need to use, will depend on the specific mesh you're working on, so it requires some experimentation to arrive at the right **Distance**.

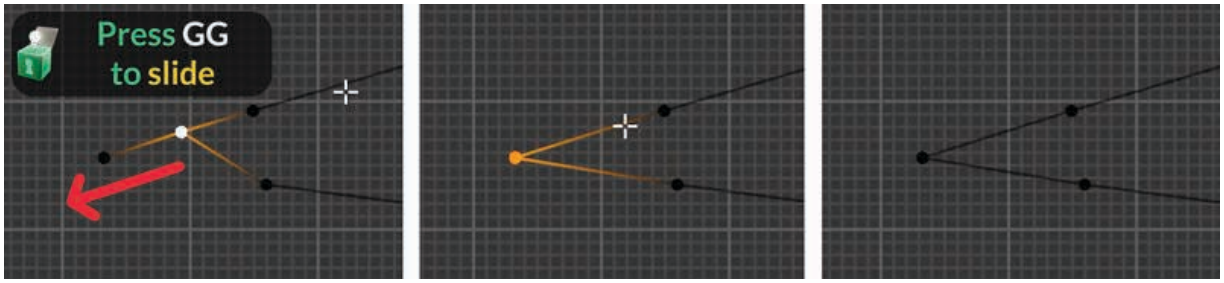
To further get rid of Vertices, we can use Checker Deselect. Unfortunately we need to do it per individual mesh island. So select one mesh island by hovering the cursor over it and **pressing L**. Then go to **Checker Deselect**. This selects every other Vertex.



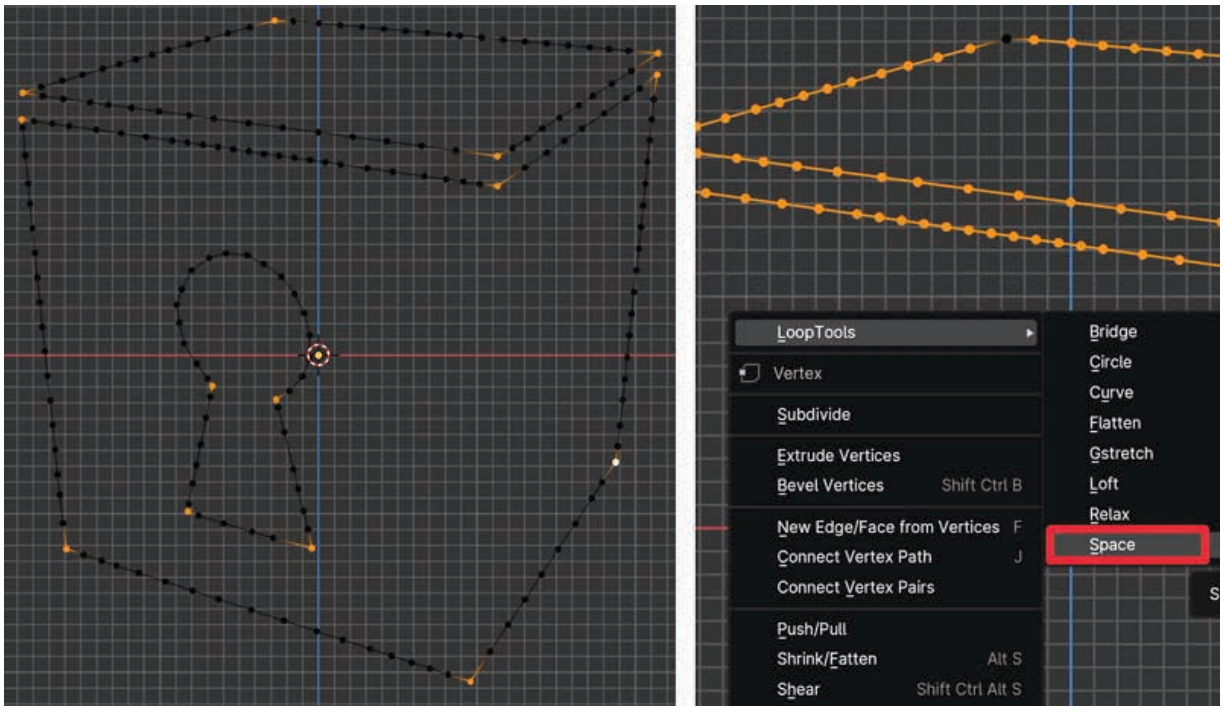
Then you can **press Ctrl+X** to dissolve the selected Vertices. Repeat these steps for each mesh island. I do recommend first deselecting the corner Vertices, as these are important for the shape.



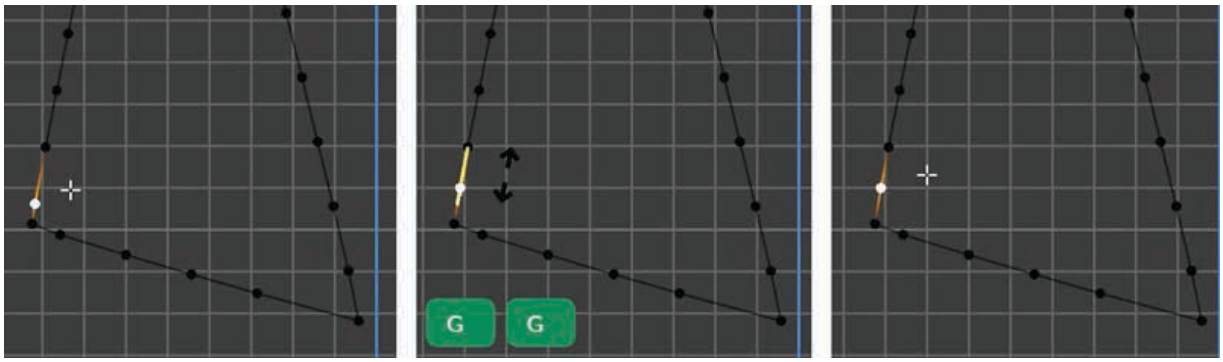
We can manually **slide** or move some Vertices to repair these corners, where the shape got destroyed a bit.



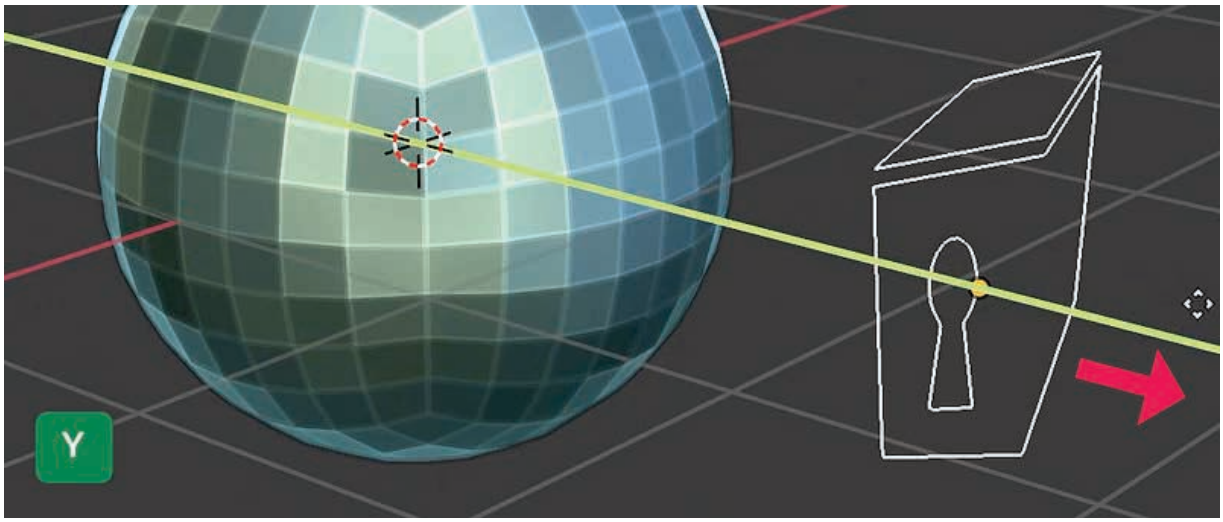
For the next step, first select all the corner Vertices. Then invert the selection by **pressing Ctrl+i**. Then you can use the **LoopTools > Space** tool to evenly space out those Vertices.



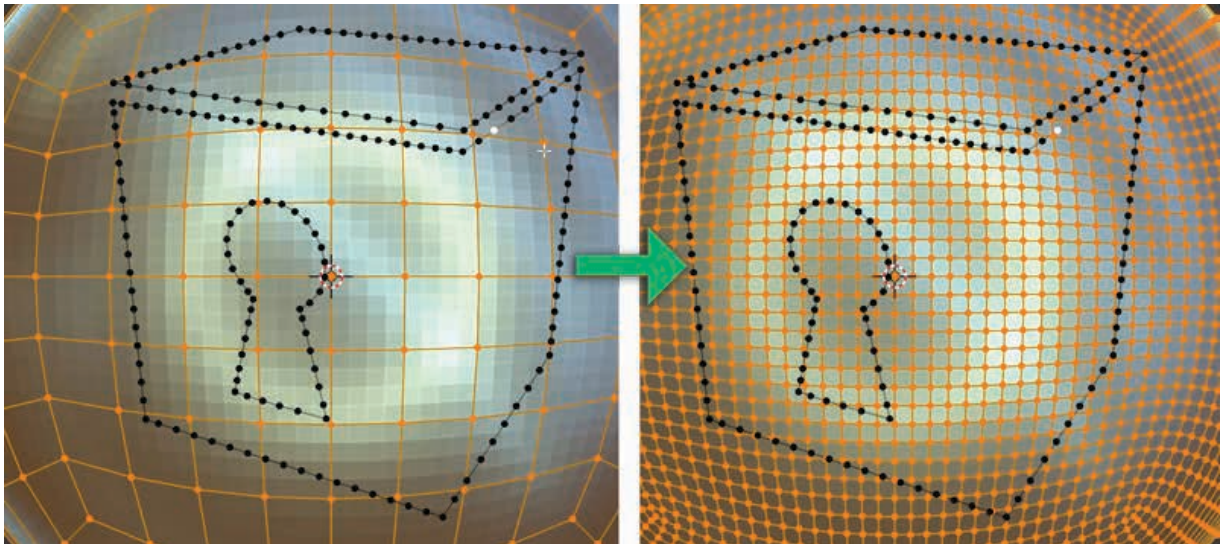
Now the Vertex distribution looks a lot better. We can manually slide (GG) and move (G) some Vertices for some final improvements.



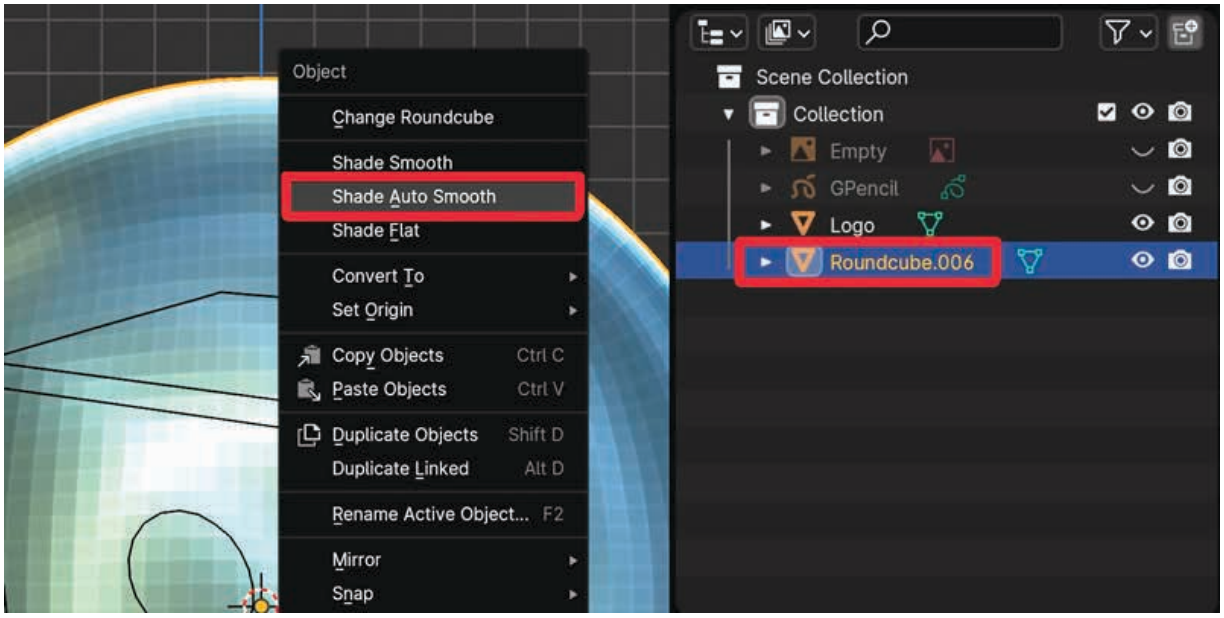
Unhide the Quadsphere. Move the logo along the Y-axis. The distance doesn't matter, as long as it's not inside of the sphere.



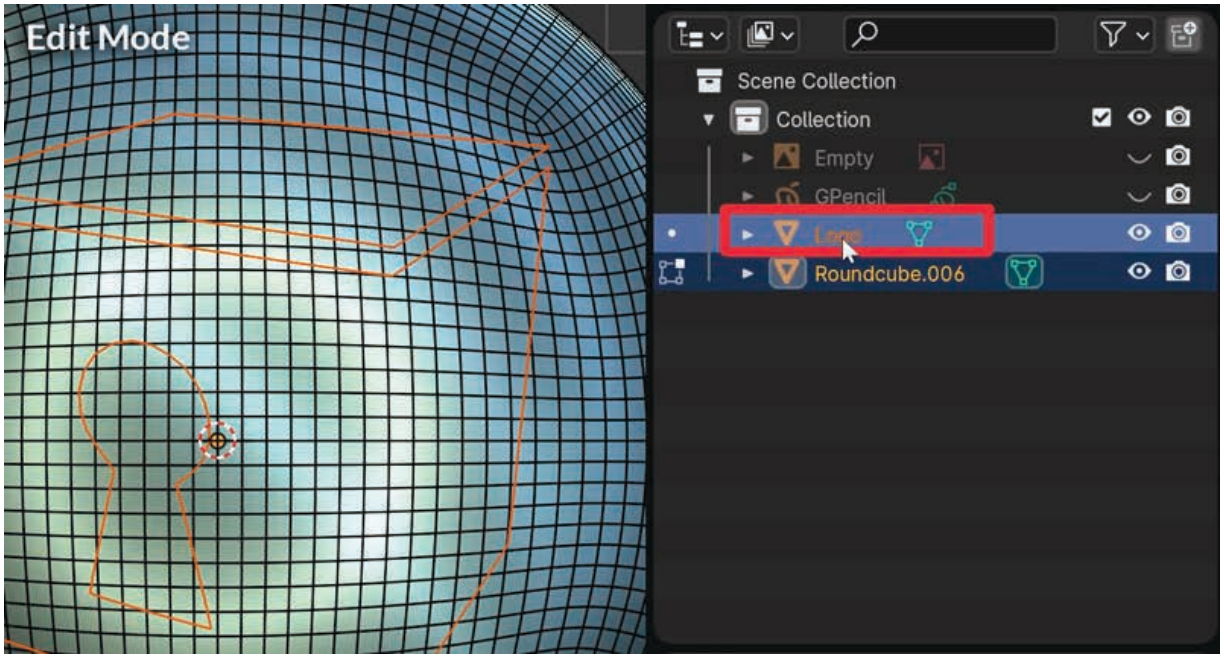
If we select both objects and go to **Edit Mode**, we can inspect and compare the vertex distribution. Usually it's a good idea to have the resolution of both objects fairly similar, to make things easier later on. In this case the sphere is much lower resolution than the logo. We can solve that by adding a **Subdiv modifier** to it in Object Mode, and **applying** the modifier. We can see that the objects then match better in terms of resolution.



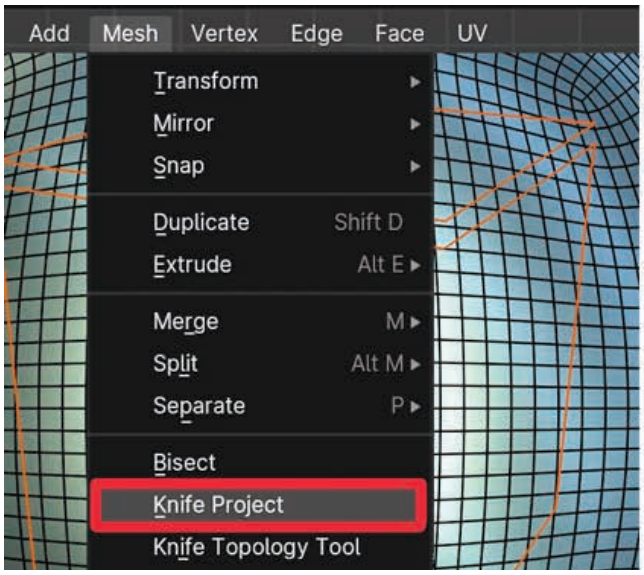
I'll just enable **Shade Auto Smooth** so it looks a bit better. Now we're ready to use Knife Project. Select only the Quadsphere (called Roundcube in the Outliner), in **Object Mode**.

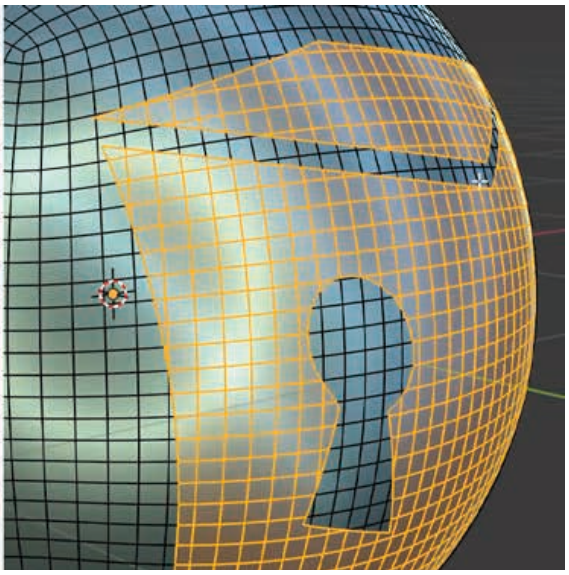


Go to **Edit Mode**. It doesn't matter if you're in Vertex, Edge or Face mode, or if everything or nothing is selected. Then **hold Ctrl** and in the Outliner select the Logo. The reason I recommend selecting it in the Outliner instead of the viewport, is because it's quite tricky to select it in the viewport.

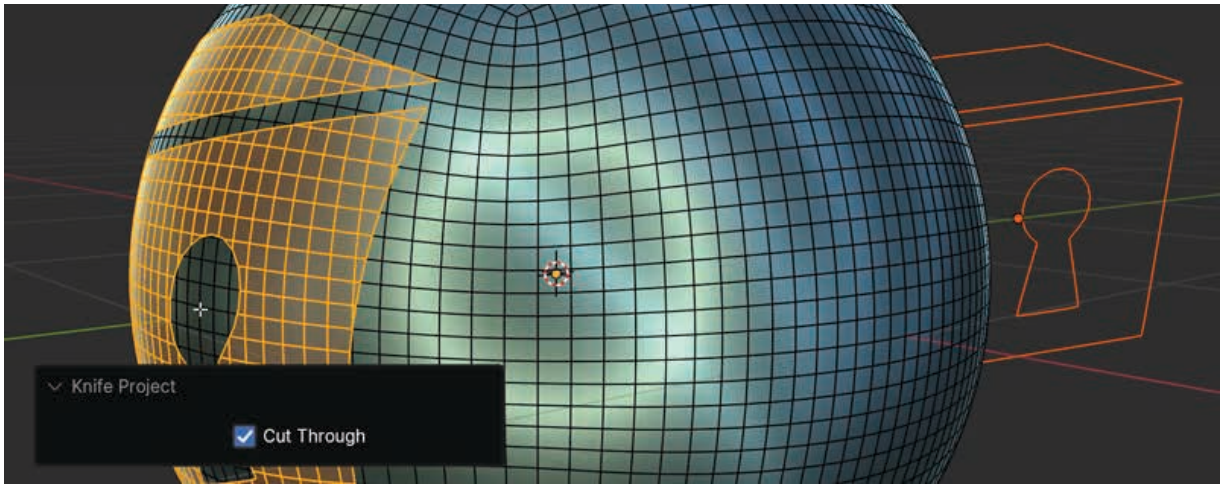


Make sure you're in **Front** view. Go to **Mesh > Knife Project**. As you can see this has cut the logo shape into the Sphere.

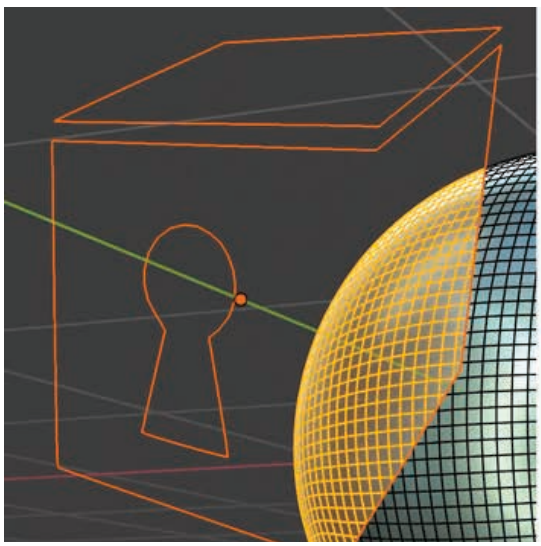


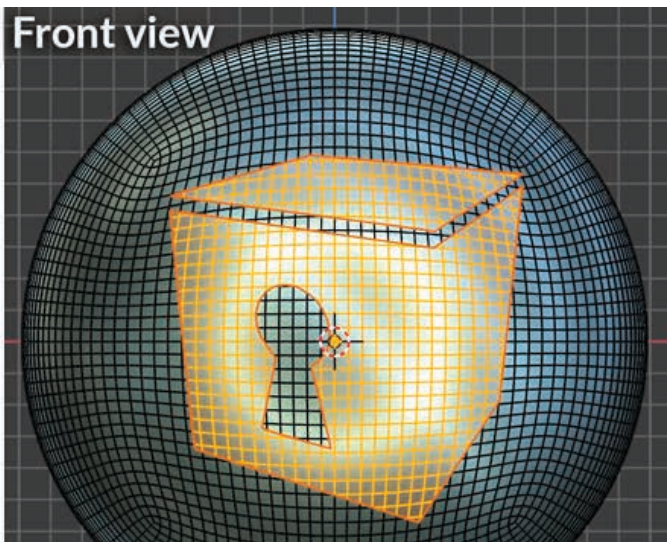


We can enable **Cut Through** in the last operator panel to cut all the way to the other side of the sphere.

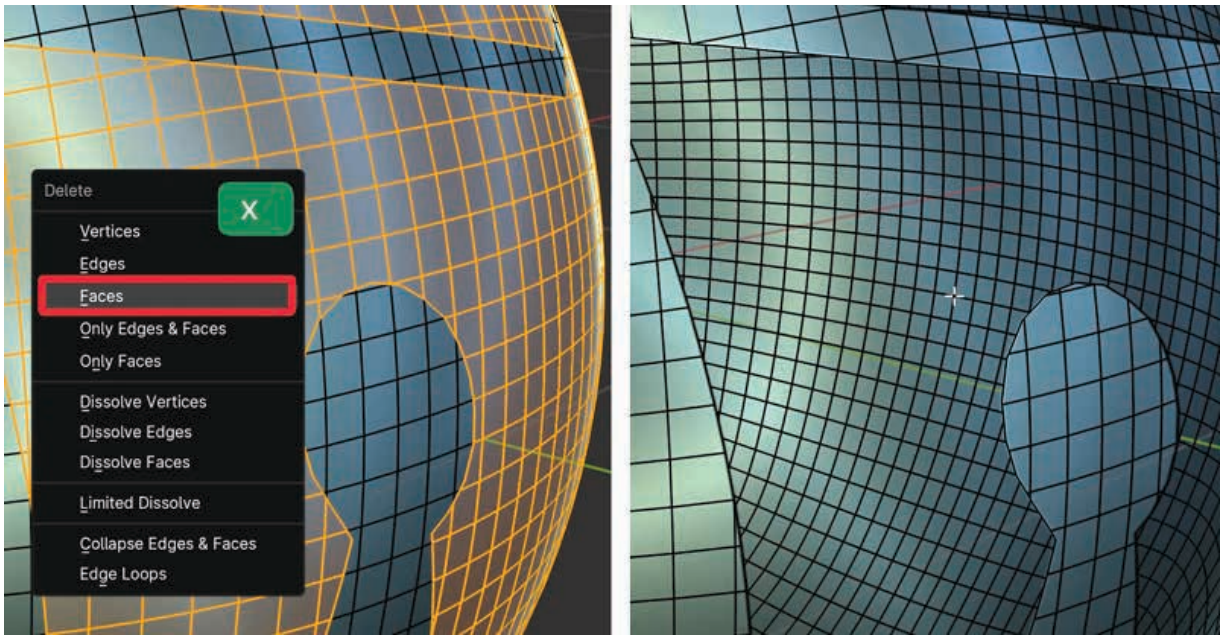


One thing that's important to know, is that the **viewing angle** determines how you cut. So for example, below on the left you see that we only cut using the overlapping part of the logo. So it's best to cut from the **Front orthographic view** in this case, as seen below on the right. While the last operator panel is active, we can still cut from different angles by checking / unchecking that option.



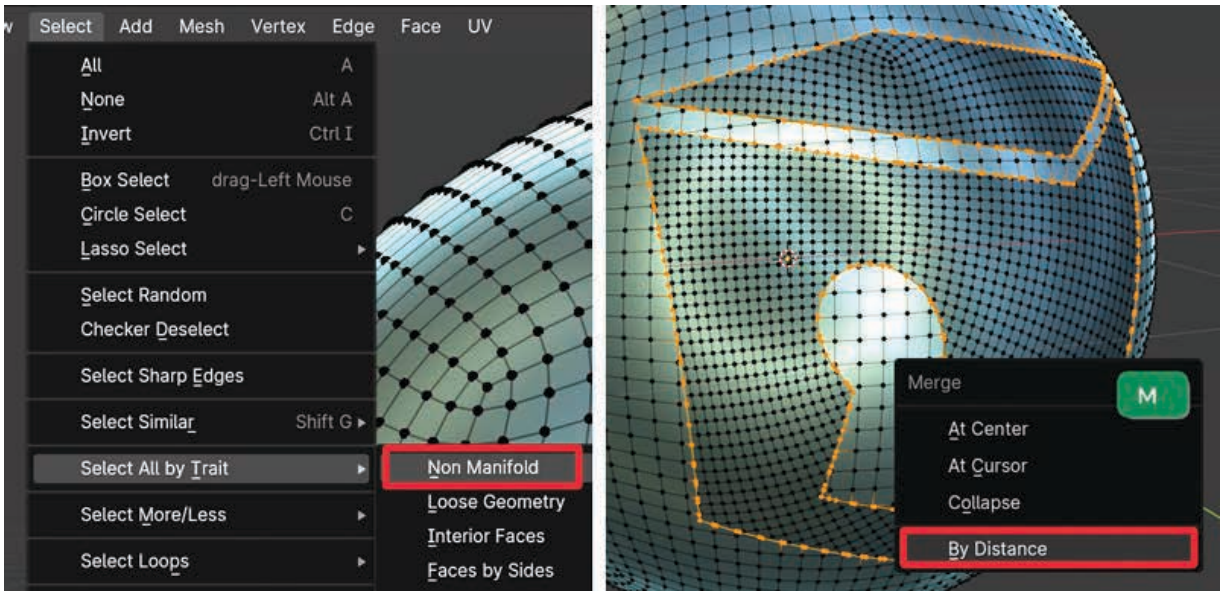


In this case we cut from the front, and without cutting through the sphere. Since the Faces inside the cut region are all selected after the cut, we can easily delete them.

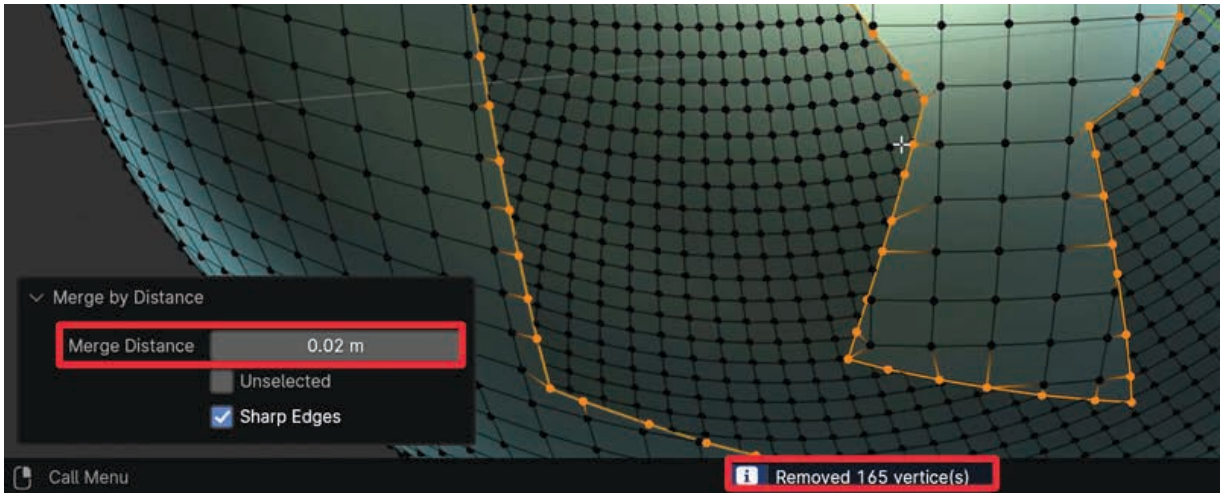


The knife project operation left all these unwanted Vertices. You could enable **Auto Merge Vertices** and then slide them by **pressing G twice** to clean this up manually.

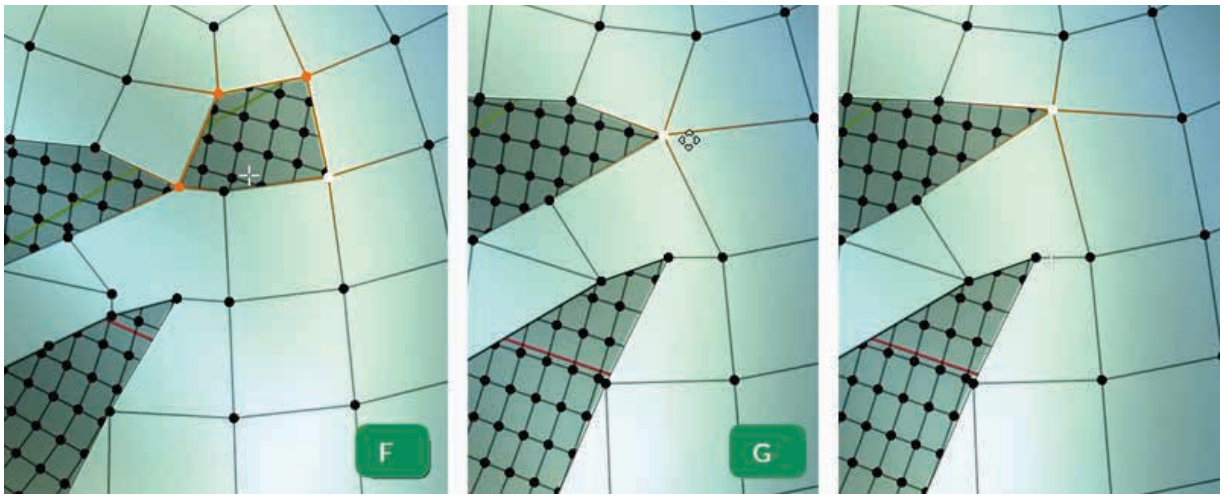
However that's quite time-consuming. We can take advantage of some nice time-saving steps. First, select the **Non Manifold geometry**, which is just the Vertices around the hole. Then **merge (M)** by **Distance**.



Experiment with the **Merge Distance** value until you get a good result. This takes care of a lot of Vertices.



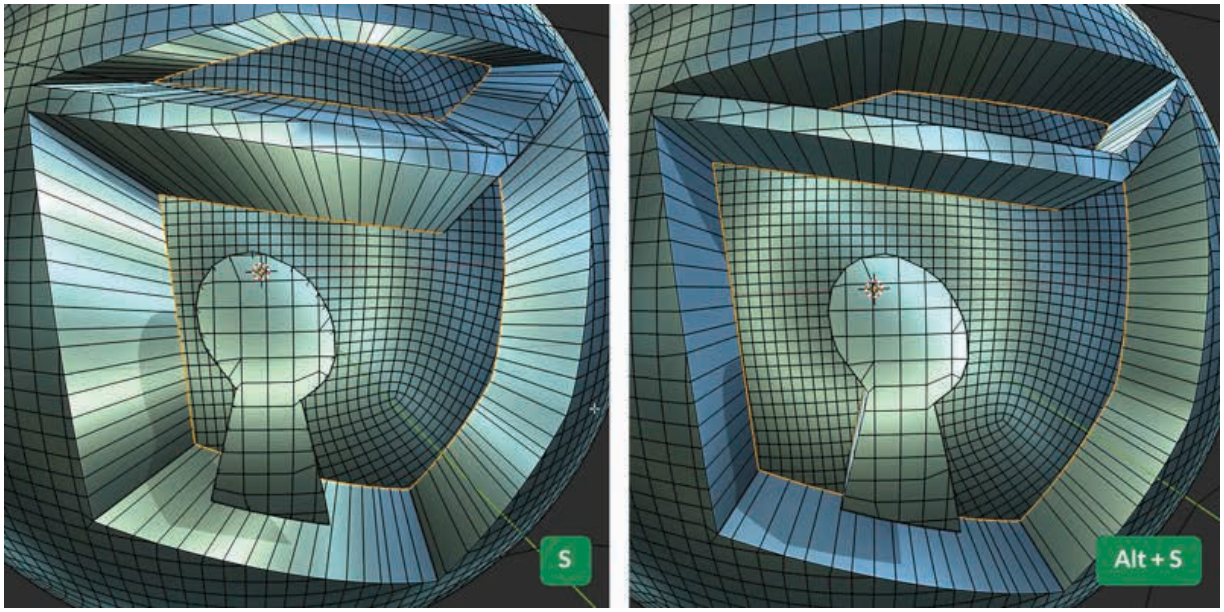
We can still manually clean the left over ones by sliding them with GG and Auto Merging them, as shown a couple of steps previous. This is also an opportunity to fix the shape of the hole where necessary. If there's a hole, select the Vertices around it and press F to fill it.



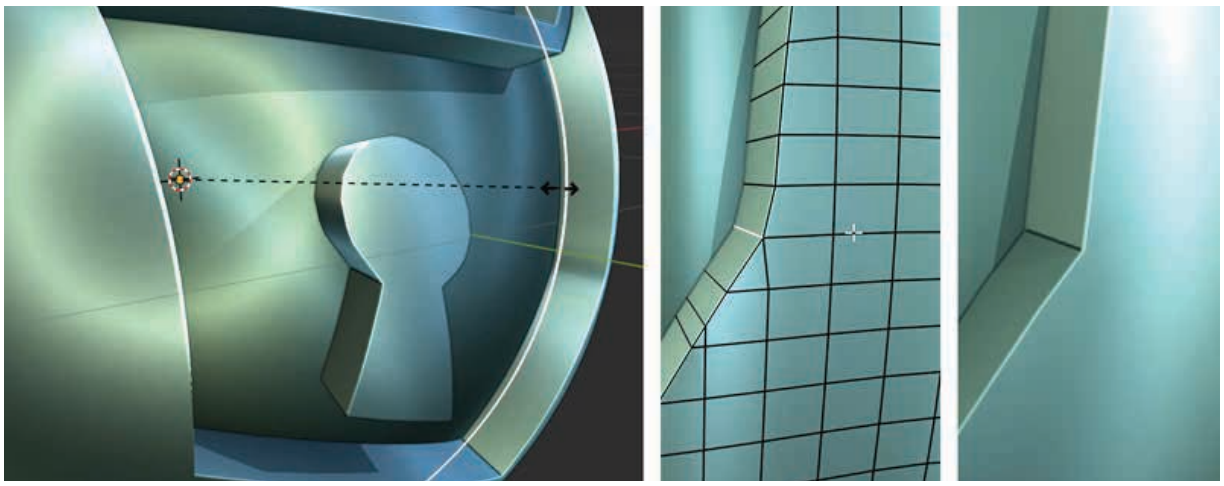
Click here to return to the index.

Select the **Non Manifold** Vertices again. Let's switch to **Edge mode** so we can deselect these Edges. Make sure only Edges around the hole are selected.

Now if we **Extrude with E** and then **scale with S**, we get a not so great result. Instead, after extruding, scale with **Alt+S**. This scales in the direction of the Normals.



This gives a pretty clean and smooth result. In **Object Mode**, let's add another Quadsphere (or UV Sphere or Icosphere). Subdivide and shade it smooth, then scale it down. This gives us an easy and non-destructive way to control the depth of the logo. If necessary you can still adjust some Edges to improve the shape.

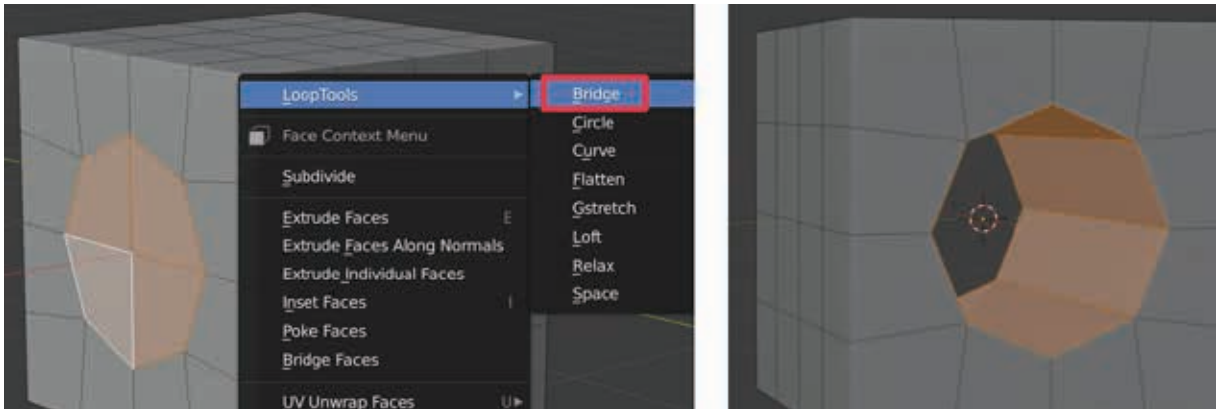


More hole-creating methods

Besides the methods mentioned in detail before, there are several more methods for creating holes in 3D models in Blender.

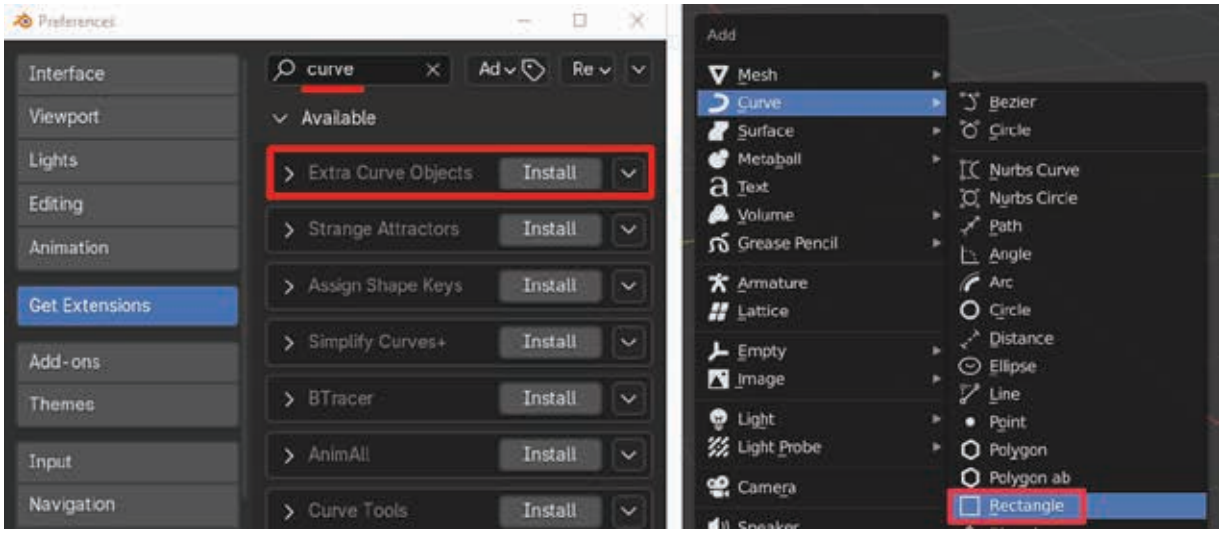
Loop Tools Bridge

Make a selection of at least 4 faces circular using **LoopTools > Circle** (LoopTools is an Extension you'll need to enable). Then, do the same on the opposite side of the cube. **Holding Shift**, select these circular faces at both sides. **Right-click** and choose **Loop Tools > Bridge**. This creates a hole through the cube. Alternatively, right-click and choose **Bridge Faces**.



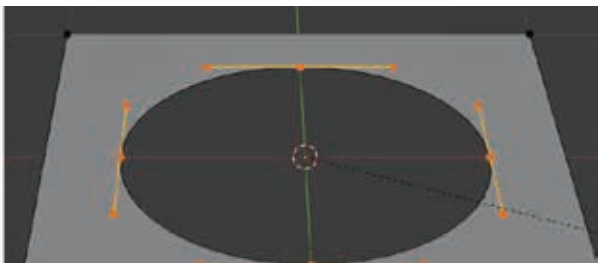
Curves

Install the **Extra Curve Objects** Extension on in Preferences so you can add a Curve **Rectangle** in the 3D Viewport.

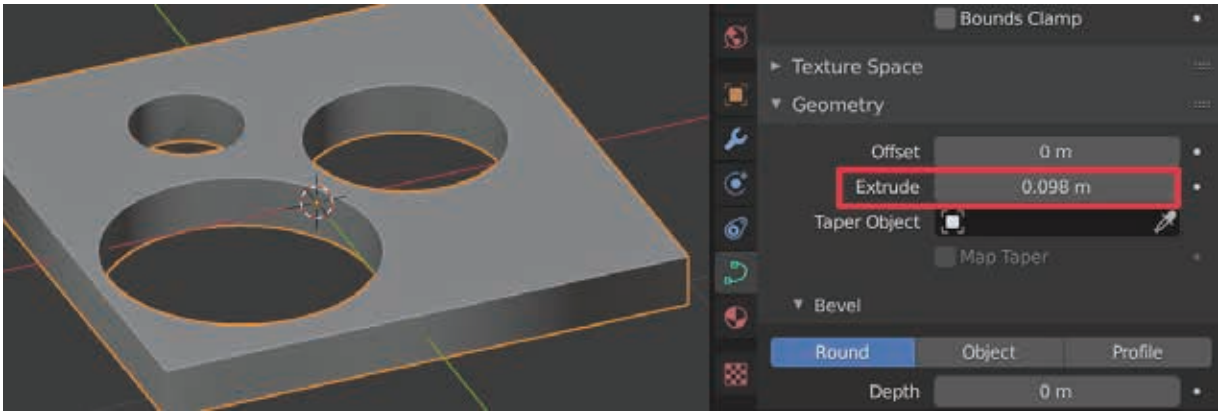


While still in Edit Mode, **press Shift+A** and add a **Circle**.

You can scale the circle by **pressing S** and move it by **pressing G**.

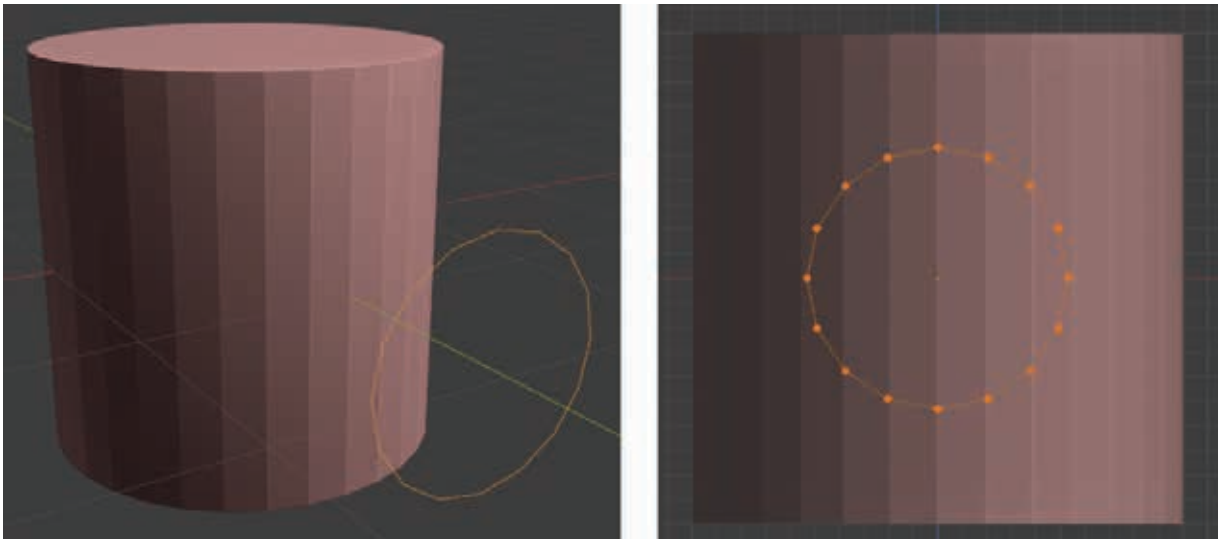


By **pressing Shift+D** you can duplicate the circle to create more holes. To add some thickness, increase the **Extrude** value of the Rectangle.

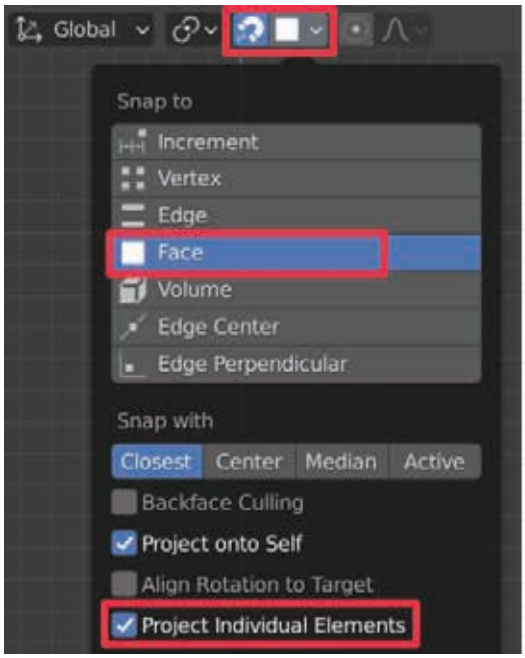


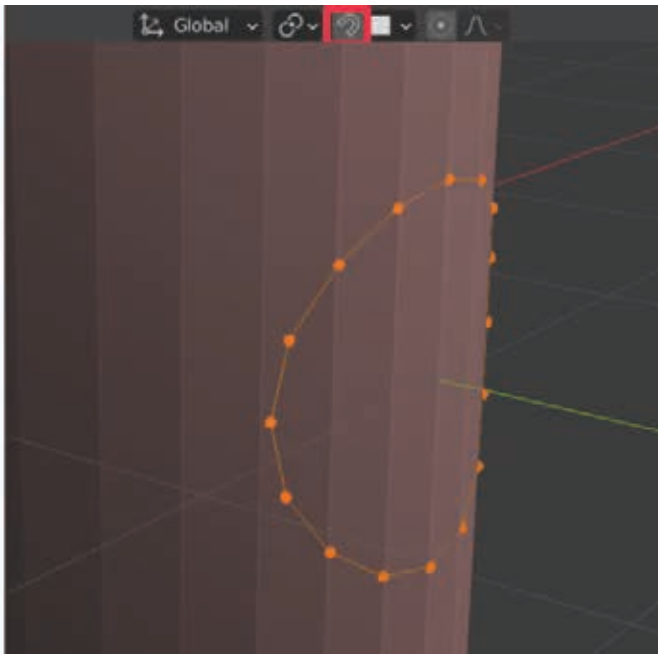
Project

In this example, we use this method to create a hole in a cylinder. **Press Shift+A** and choose **Mesh > Circle**. Move and rotate the Circle so that it is placed in front of the cylinder and scale it down in Edit Mode. I recommend not giving the Circle too many vertices, 12 is enough (less is fine, too, if the Cylinder is also more low poly).



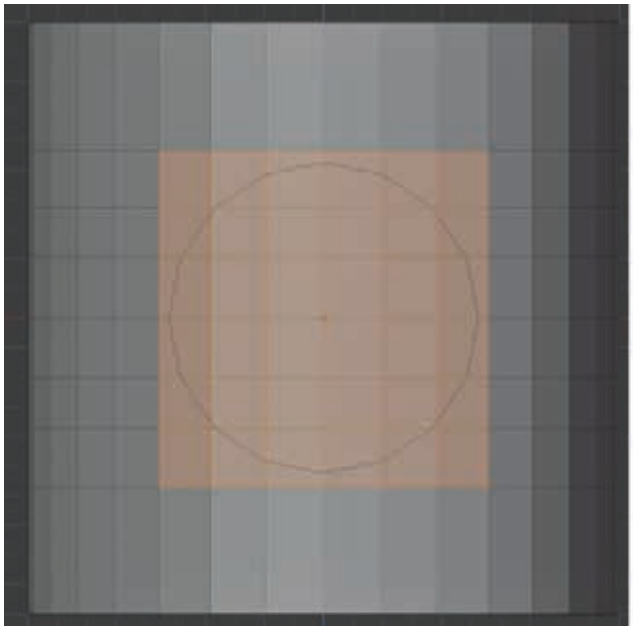
Turn on snapping to **Face** and check **Project Individual Elements**. In orthographic view, with all vertices of the Circle selected in Edit Mode, **press G and [Enter]** - don't move the Circle.

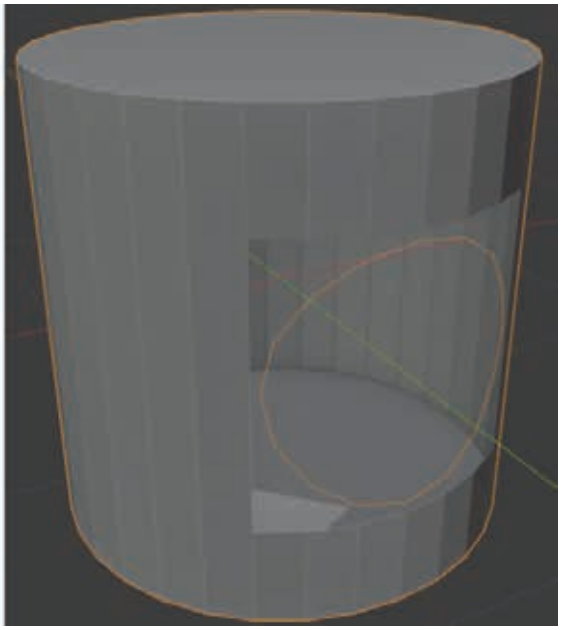




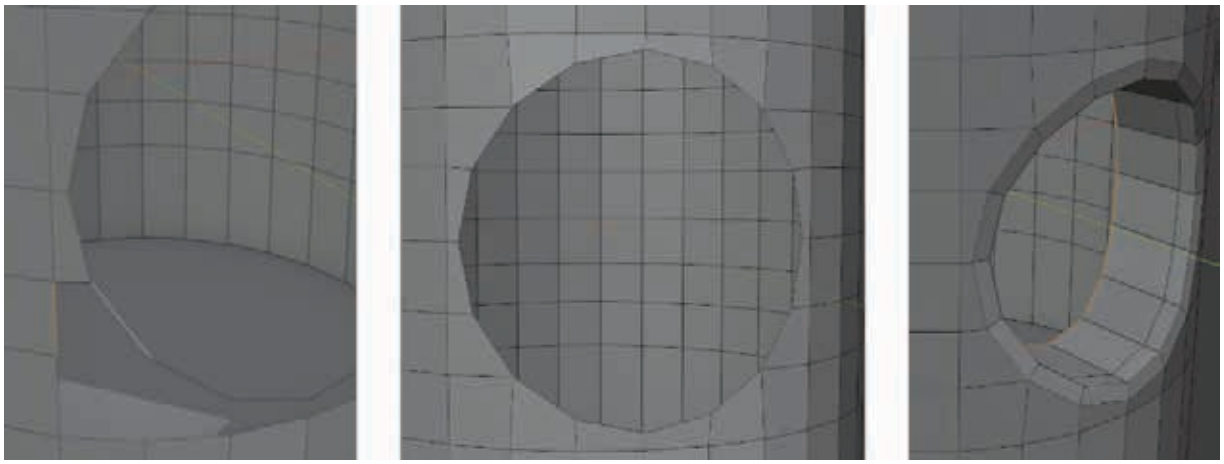
By briefly going into the move tool by **pressing G**, the vertices snap to the faces of the Cylinder. **Turn off Snapping**.

In Object Mode, select both the Cylinder and the Circle and **press Ctrl+J** to join them. Add some edge loops to the Cylinder by **pressing Ctrl+R**. You can add them manually one by one, to match the vertices of the Circle. Then remove the faces underneath the Circle by selecting them and **pressing X > Faces**.



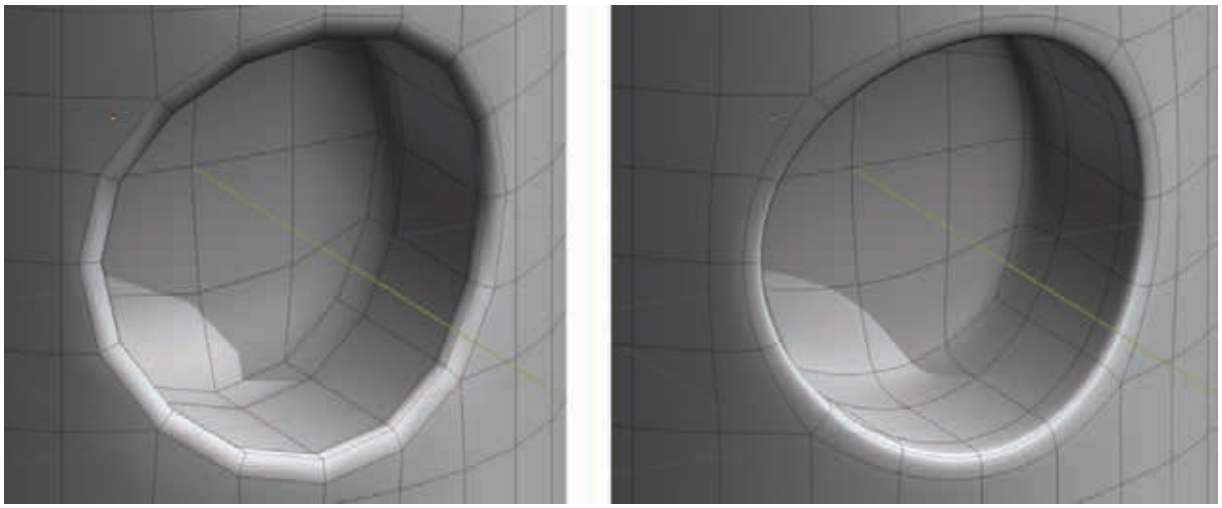


Fill in the gaps between edges of the Cylinder and the Circle by selecting two opposing edges and **pressing F** (for Fill). Create faces this way to connect the Circle to the Cylinder and then select the edge loop of the Circle by **Alt+double-clicking** on it. **Press E+S** to extrude it and scale it down, then create some more extrusions inward.



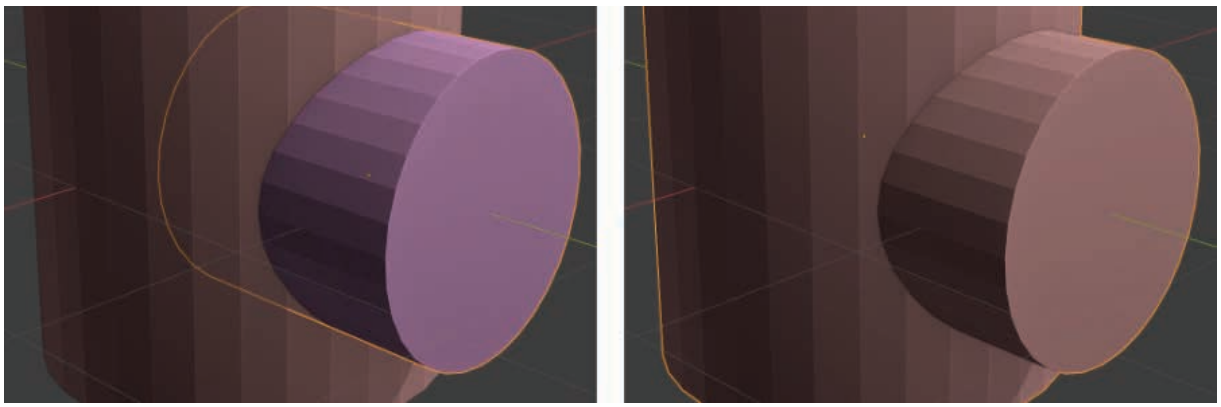
To close the hole, **press Ctrl+F+G** for Grid Fill (or **press Ctrl+F** and choose **Grid Fill)**. Then you may have to change the **Offset** value to rotate the faces.

With this method using support loops (the extra edge loops around the edges) you can add a **Subdivision modifier** to add extra smoothness.

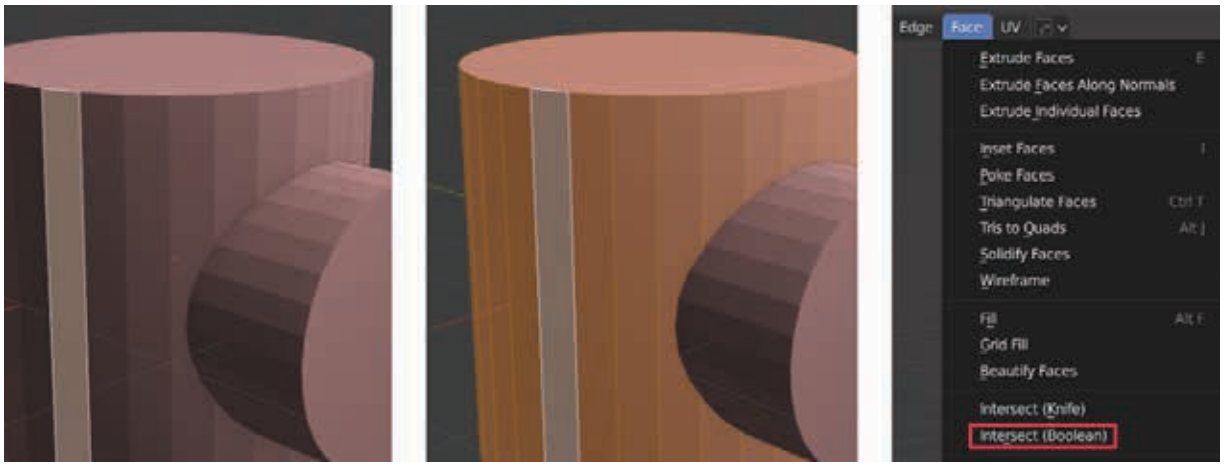


Intersect

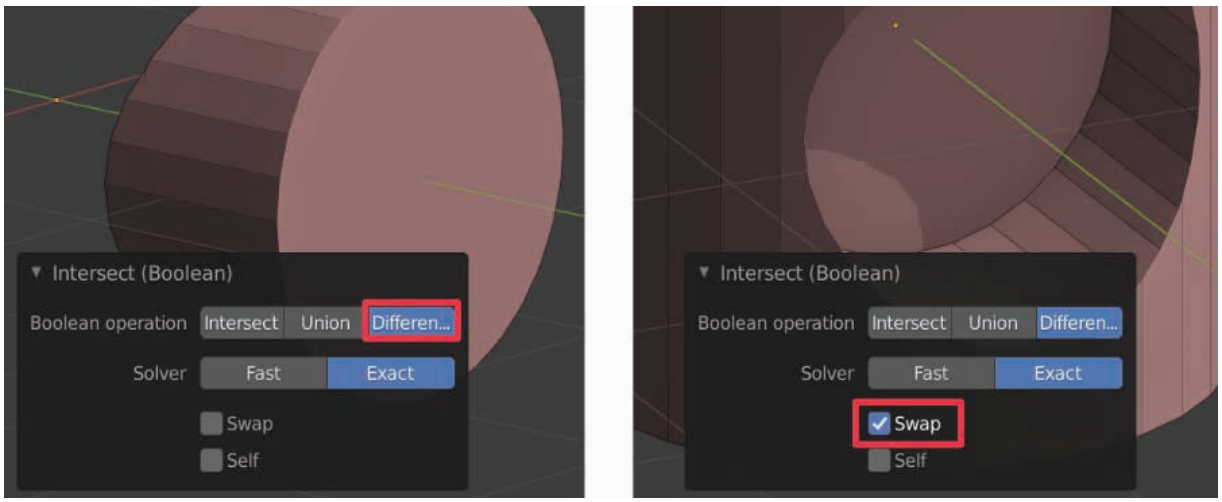
Add a Cylinder and rotate, move and scale it so that it intersects with the first Cylinder. Select both objects and **press Ctrl+J** to join them.



In Edit Mode select a face of the object you're going to cut out of, then with the mouse cursor hovering over that object **press L** to select the rest of the faces. Then from the **Face menu** choose **Insersect (Boolean)**.

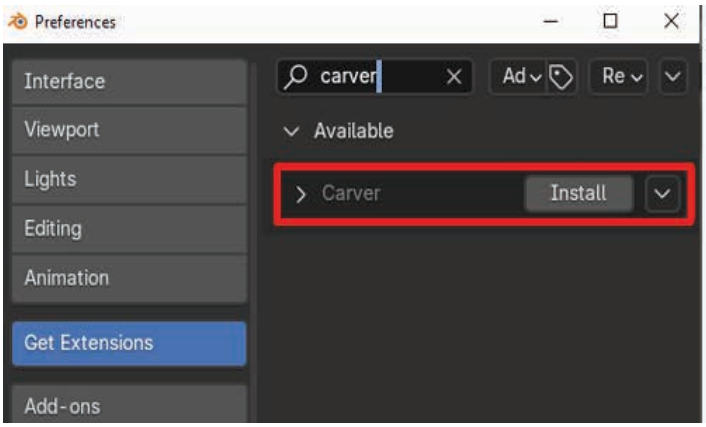


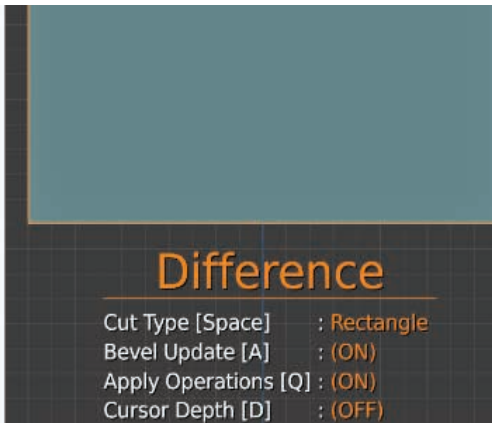
Then choose the Boolean option **Difference** and check **Swap**.



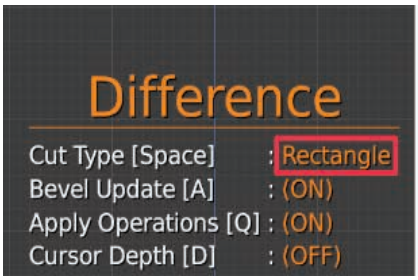
Carver

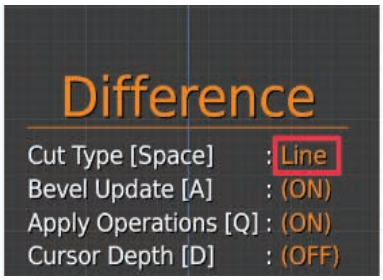
Install the Carver add-on in Preferences > **Get Extensions**. To activate it in the 3D Viewport, **press Shift+Ctrl+X**.

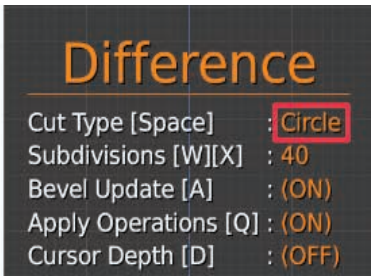




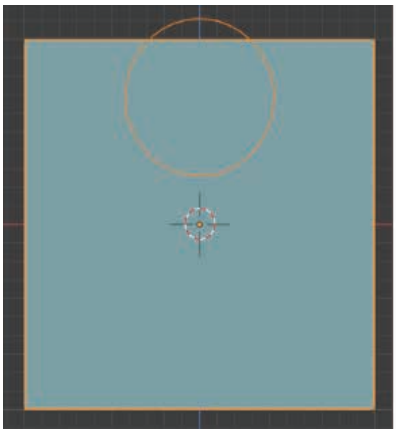
Pressing the Spacebar cycles through the different cutting shapes: Rectangle, Line and Circle.

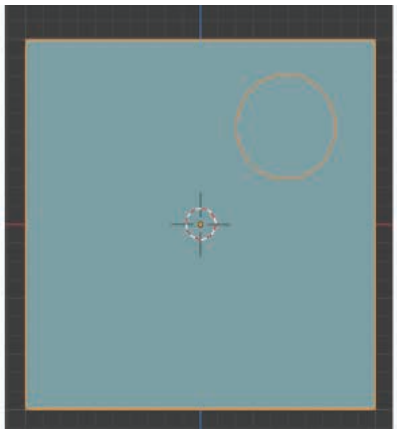


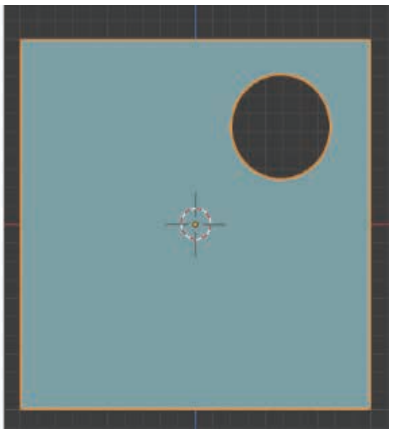




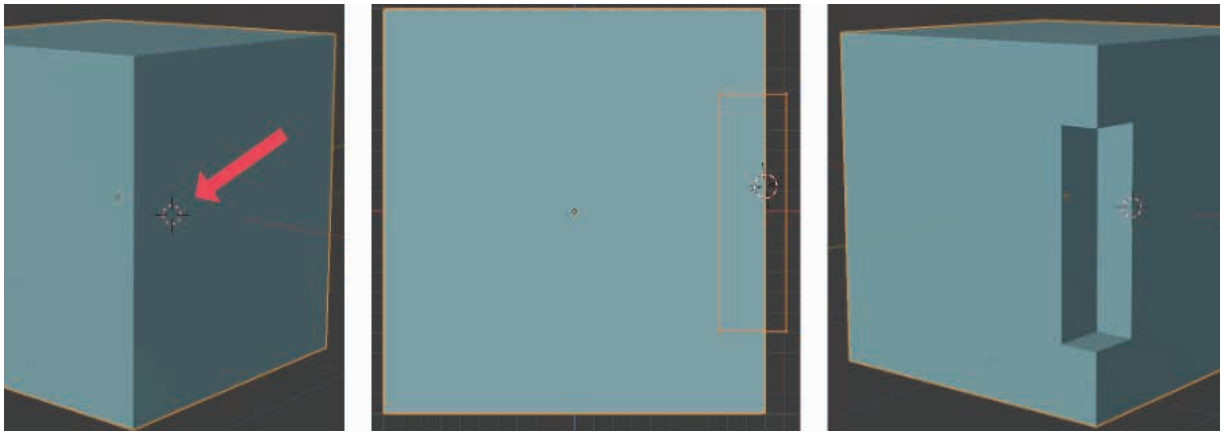
To create a circular hole, **press the Spacebar** until you get the **Circle** cut type and **click on the object** you want to cut into. Then **drag the mouse** to increase the size of the Circle. **Click again** to confirm the cut. If you want to move the circle before cutting, **hold Alt** before clicking the second time.







By default Carver will cut all the way through the object. But you can also cut to a specified depth. **Press Shift+Ctrl+X** to activate the Carver tool. **Ctrl+left-click** on the object to move the **3D cursor**and **Press D** to toggle **Cursor Depth** on. Make the cut the usual way by clicking, dragging the cutter shape and clicking again. It may not look like you've cut anything until you look at the object from the side.

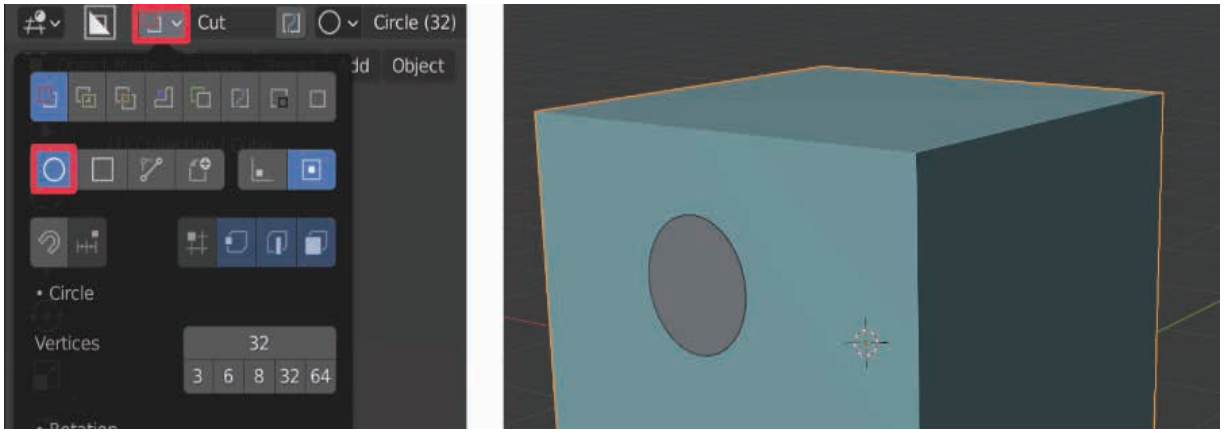


Boxcutter

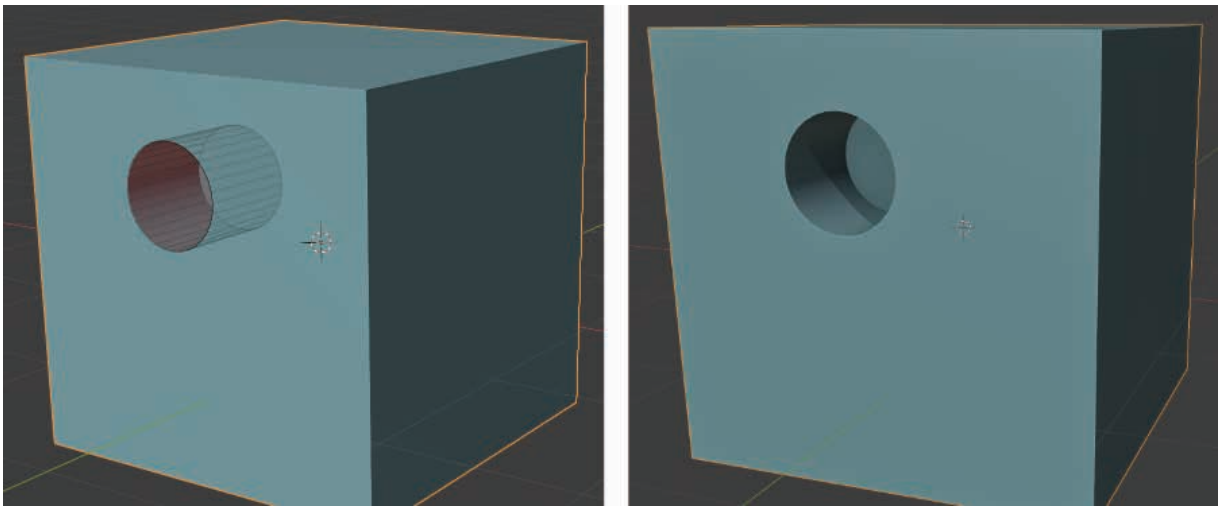
Finally I'd like to mention Boxcutter. Quite similar to the free Carver add-on, Boxcutter is however slightly more user-friendly. It's also a commercial add-on (it costs 20 USD) so you may want to stick to Carver if you want a free option.

I'd say the main difference is that with Boxcutter you don't need to be in an orthographic view to cut from that angle, so it's a bit more fun to use (though it does take a while to learn all the options and keyboard shortcuts).

To activate Boxcutter (once the add-on is installed) **Press Alt+W**. You can choose the cutter shape from the drop-down menu. Click on the object to draw the cutter.



Once you let go of the left-mouse button, you can change the depth of the cut. **Click again to confirm**.

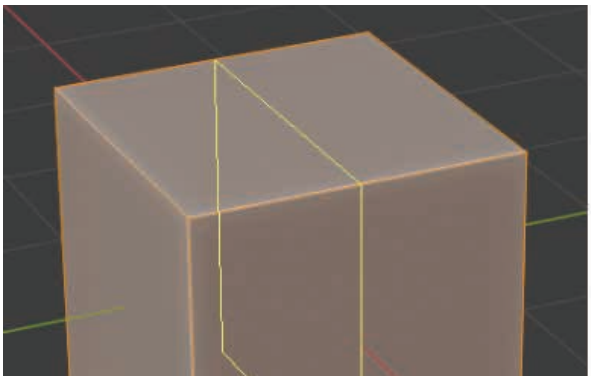


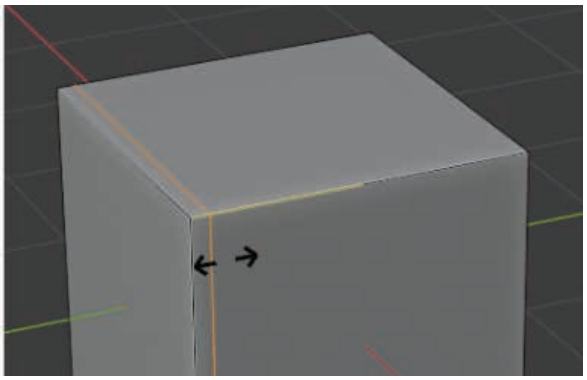
Boxcutter is just one of many commercial add-ons that facilitate fast Boolean-based workflows in Blender. Some other notable examples are Fluent and Meshmachine.

Support Loops

Support Loops are a fundamental part of polygonal modeling. Add a Loop Cut with the shortcut **Ctrl+R**. Left-click to confirm.

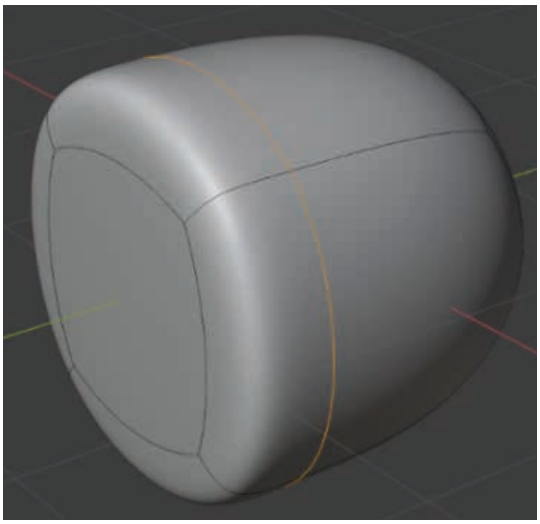
Then move it to where you want. **Left-click again** to confirm the position.

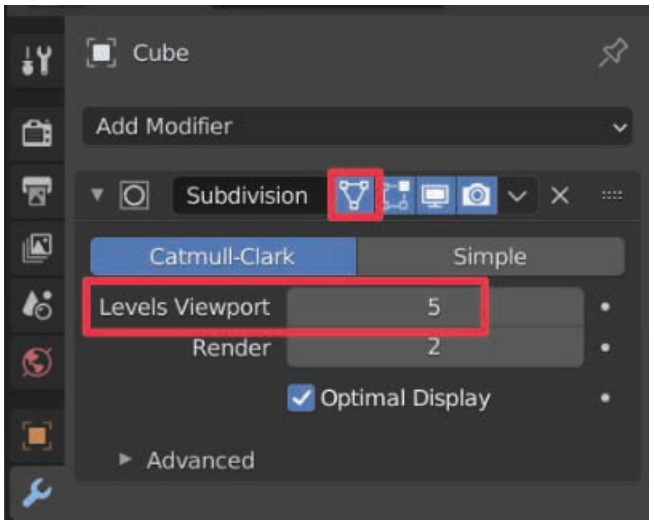




You can see what the loopcut does when you add a **Subdiv modifier** to it.

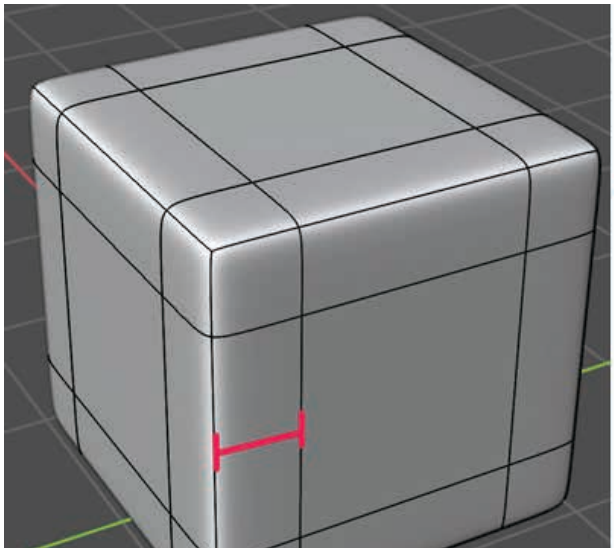
Click the **On Cage** button to see the effect of the modifier on the edges in Edit Mode.

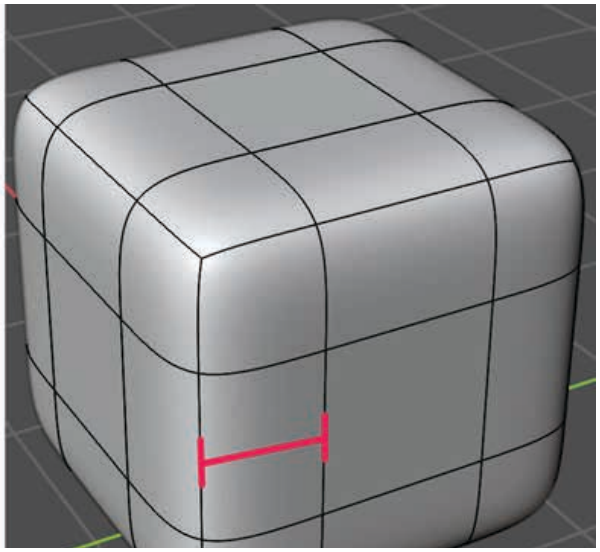




The closer the support loops are to the edge of the model, the sharper the edges.

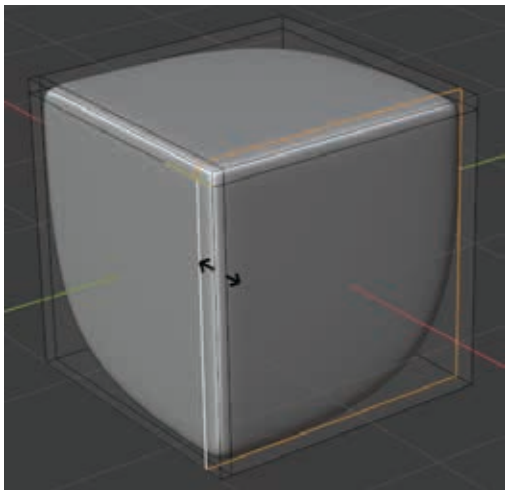
If they are further from the edge, the edges become smoother.





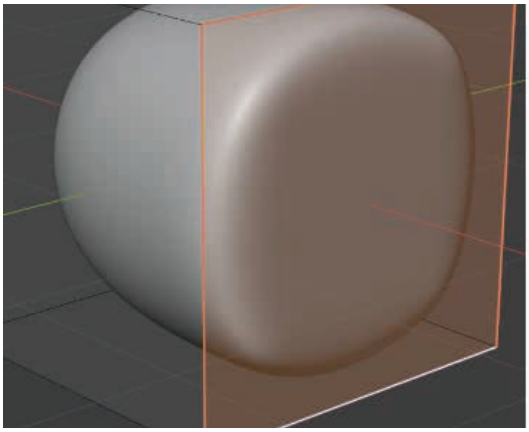
Click here to return to the index.

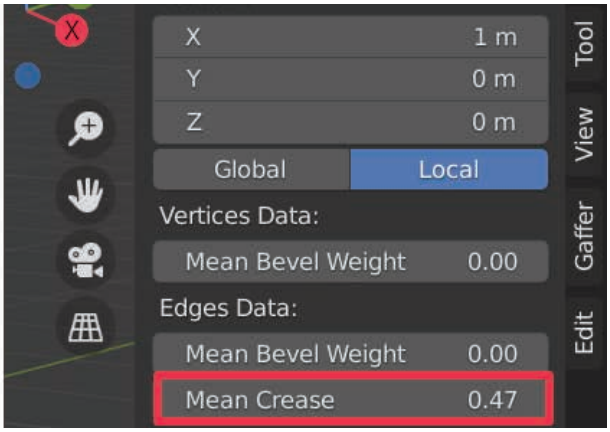
To adjust how sharp you want the edge of the model to be you can move it closer to the edge of the model. To do this, select it by **holding Alt and double-clicking** and **press G twice** to slide the loop.



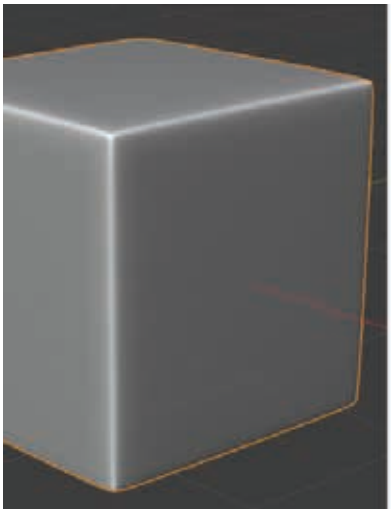
Note: if you have "enable 3-button mouse" enabled in Preferences you only need to double-click on the loop to select it.

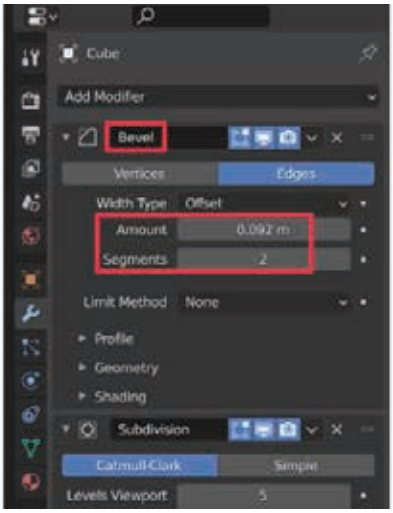
For a non-destructive alternative, you can select edges and **press Shift+E** and move the mouse (or enter a numerical value up to 1) to Crease edges. You can also change the **Mean Crease** value in the Option panel for the same result.





You could also add a **Bevel modifier** to get non-destructive sharp edges on your model. If you apply the modifier you can see in Edit Mode that it results in support loops.

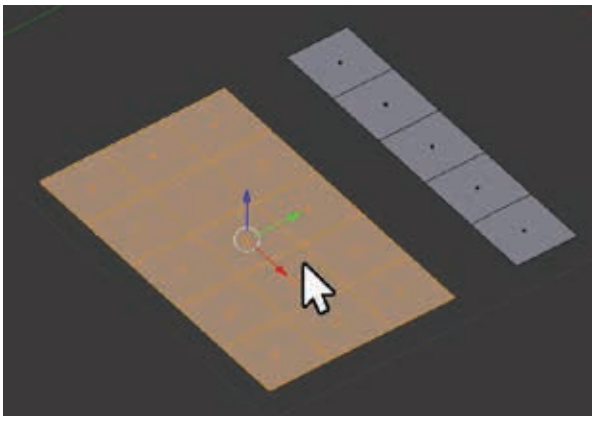




Select Connected Faces

While hovering the mouse cursor over a face, **press L** to select all the faces that are connected to it.

Press L again when hovering over another set of faces to add those to the selection (no need to hold Shift, in other words).

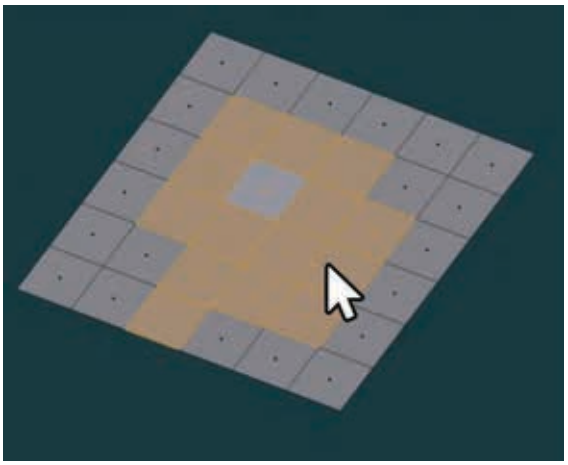


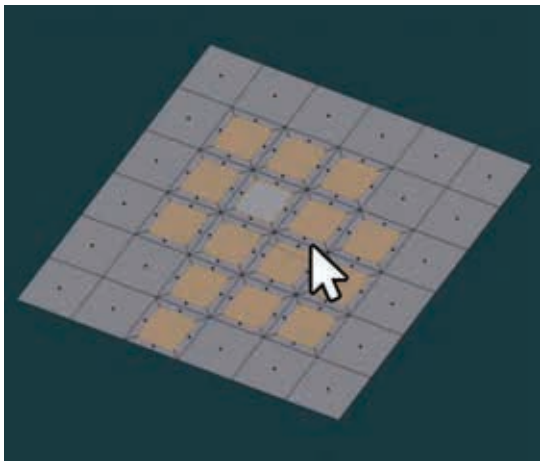
Inset individual faces

Ever wondered how to "inset" individual faces? It's very easy… as long as you know how. You just **press i twice!** First select the faces you want to inset...

Press i, then **press i again** to inset each individual face.

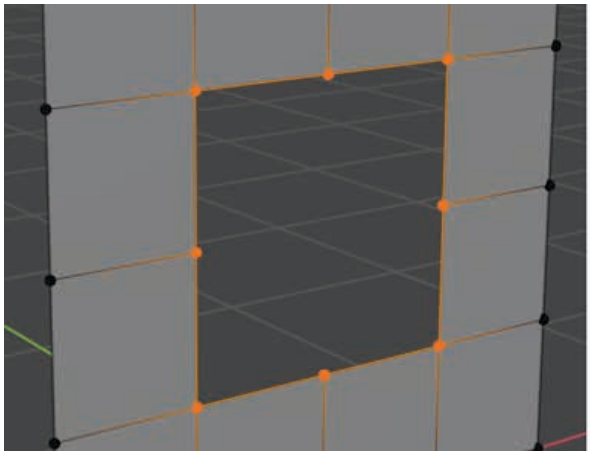
You can toggle back to insetting all faces at once by pressing i again.

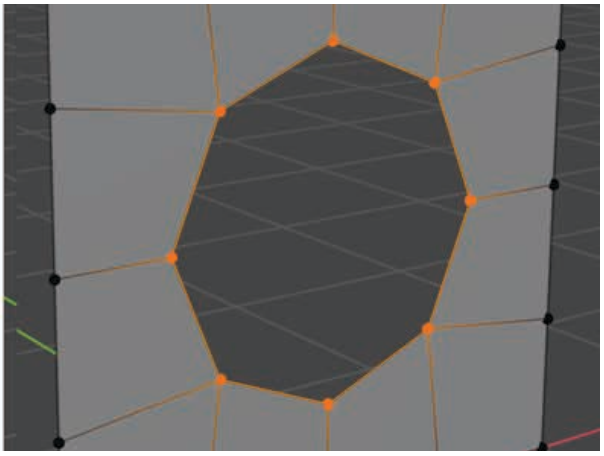




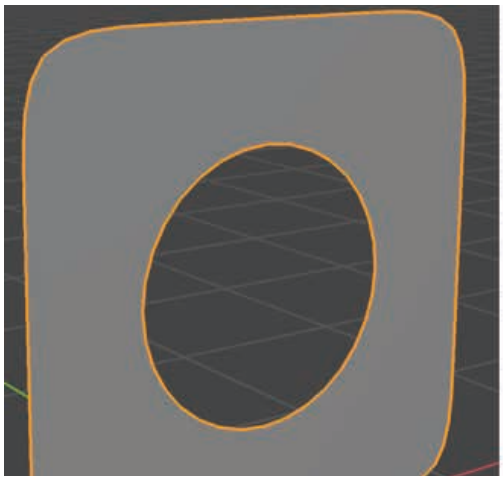
Make selections circular with To Sphere

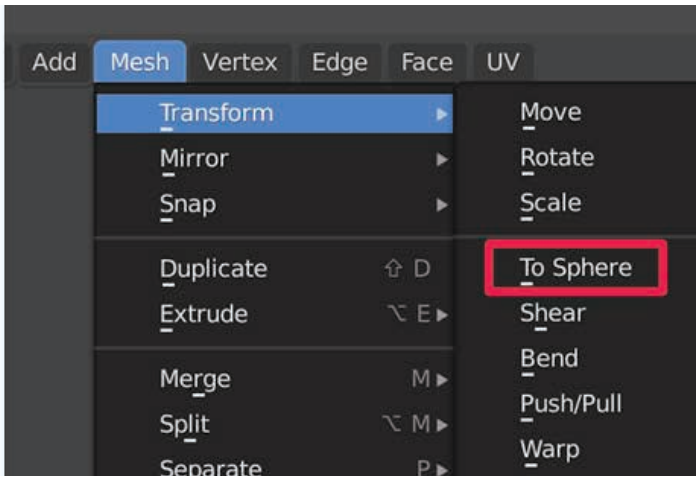
To make this square hole circular, select the vertices or edges and **press Shift+Alt-S** then drag the mouse. Or **press Shift+Alt+S and press 1 on the numpad** for the maximum value.



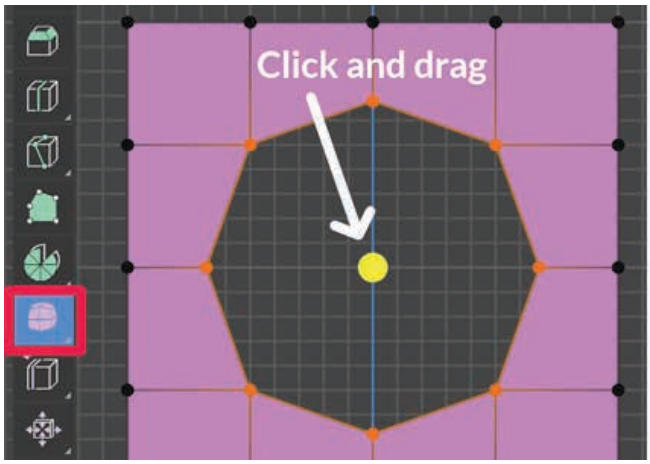


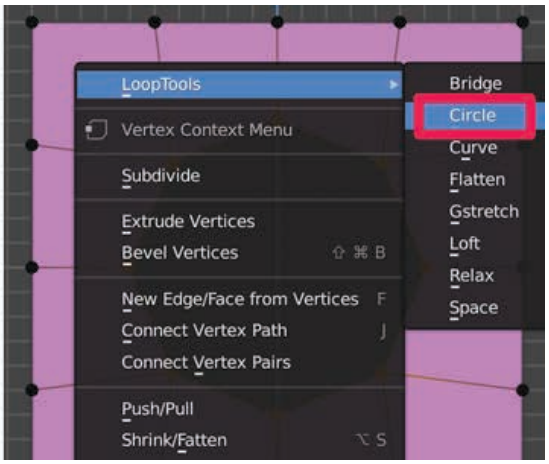
Add a Subdivision modifier in Object Mode by **pressing Ctrl+2** if you want to make the hole smooth. You can also find the To Sphere tool under **Mesh > Transform > To Sphere**.



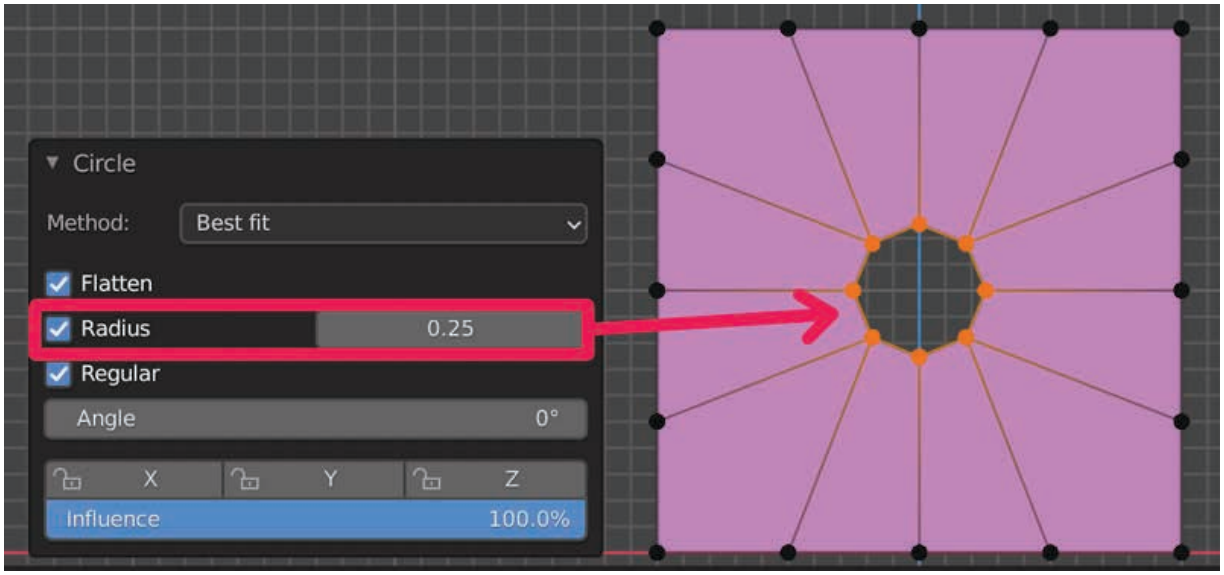


Alternatively, you can use the **Smooth** tool from the Tool panel (press T to toggle the visibility of the Tool panel in the 3D viewport). Or, use the **Circle** option from the LoopTools menu.

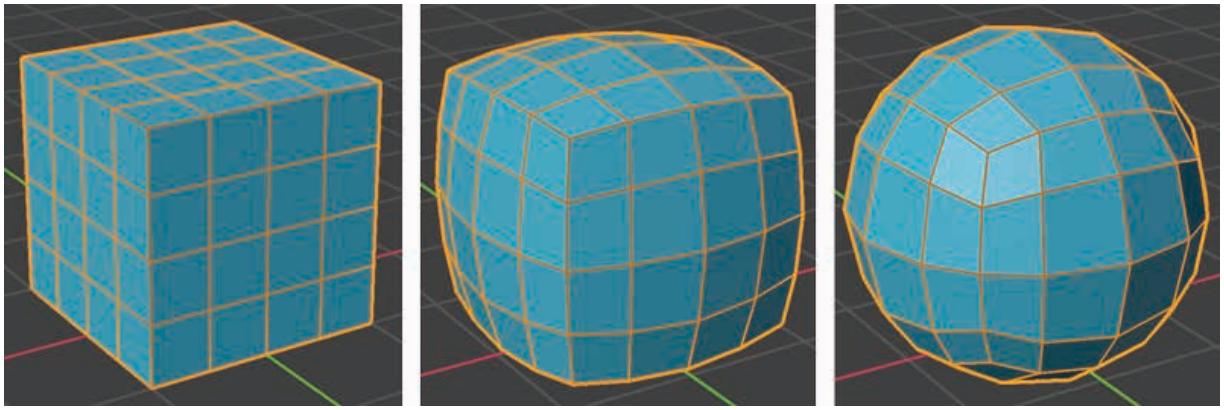




To have this LoopTools > Circle option, you need to enable **LoopTools** in Preferences. The benefit of LoopTools is that you can change the **Radius** easily as well.



To Sphere can also be used to make selections not just round but **spherical**, as the name implies.

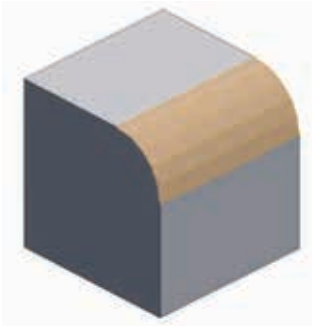


Concave Bevel Profile

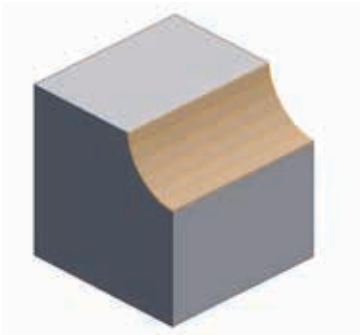
When beveling an edge, you can choose to add more detail, but also to have it be concave instead of convex.

First, press **Ctrl+B** to bevel the selected edge.

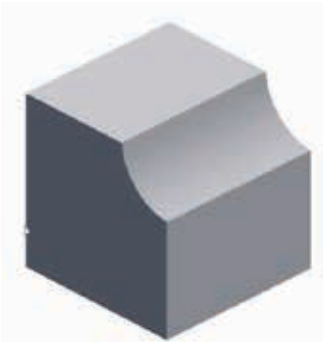
Next, scroll the mouse wheel to add more edges (for more control, use the + key)

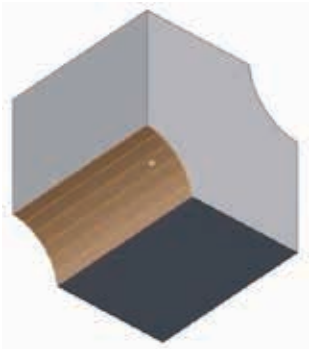


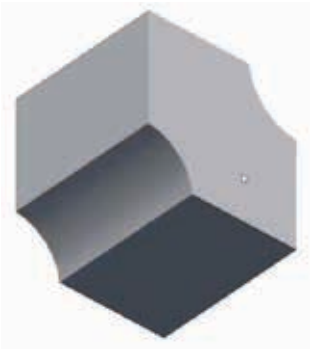
Press P (for "profile") and move the mouse to change the profile from convex to concave.



Now the profile of the bevel is "concave".

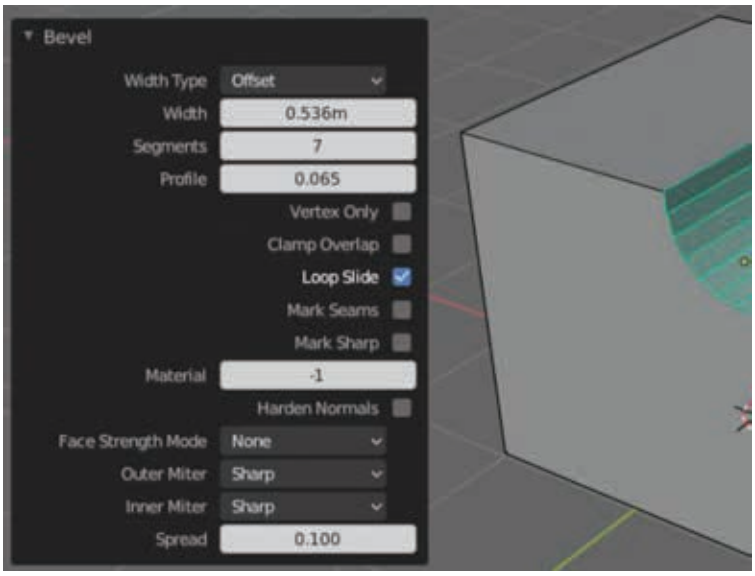






Blender remembers the last setting you used for your bevel and will automatically apply it the next time you bevel an edge.

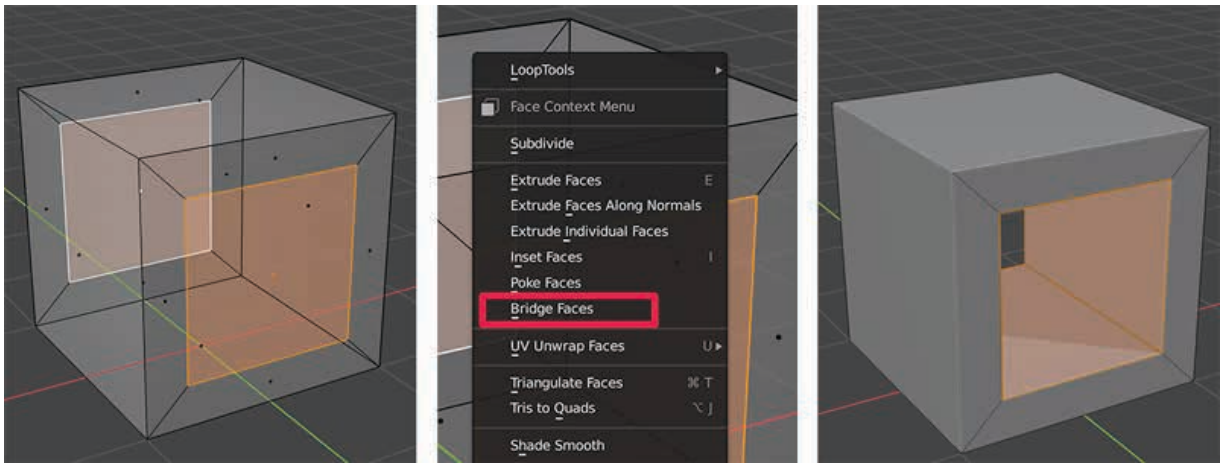
When beveling, Blender also shows a Bevel menu in the viewport (you can expand it).



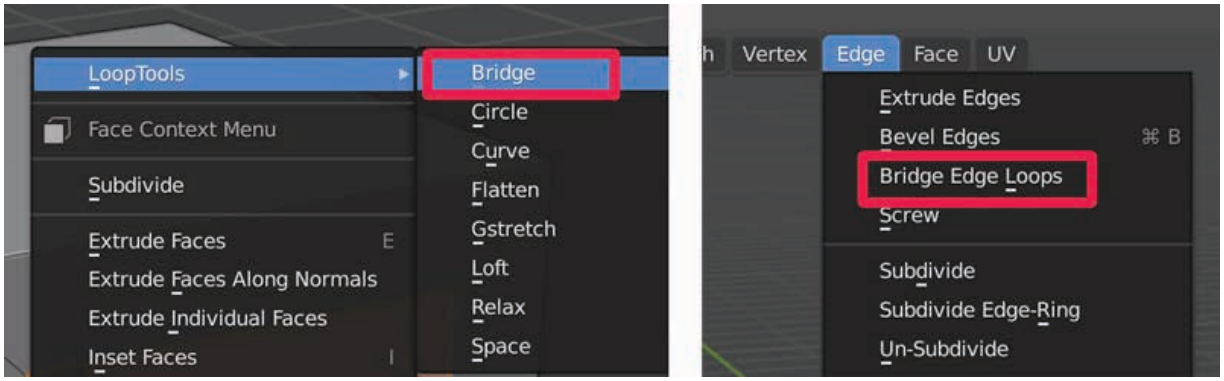
Click here to return to the index.

Make connected holes and smooth surfaces with Bridge

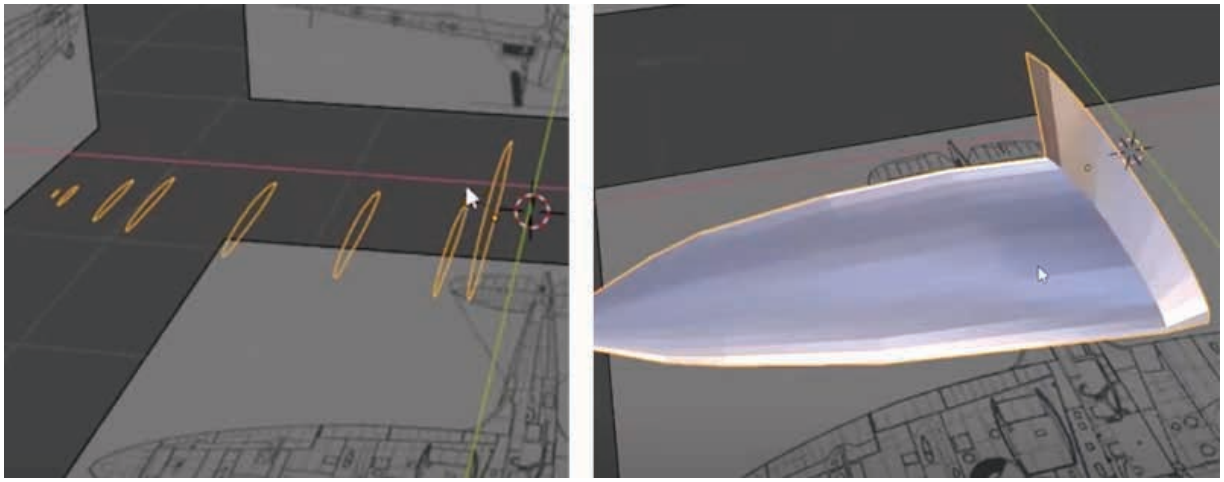
If you've inset two faces on either side of a cube you can create a tunnel through it. To do this, right-click in Edit Mode and with those opposing Faces selected choose **Bridge Faces**.



A very similar **Bridge** option is also available in the **LoopTools** menu, if that add-on is enabled (in the Get Extensions tab in Preferences). It can also be found in the Edge Menu, where it's called Bridge Edge Loops.

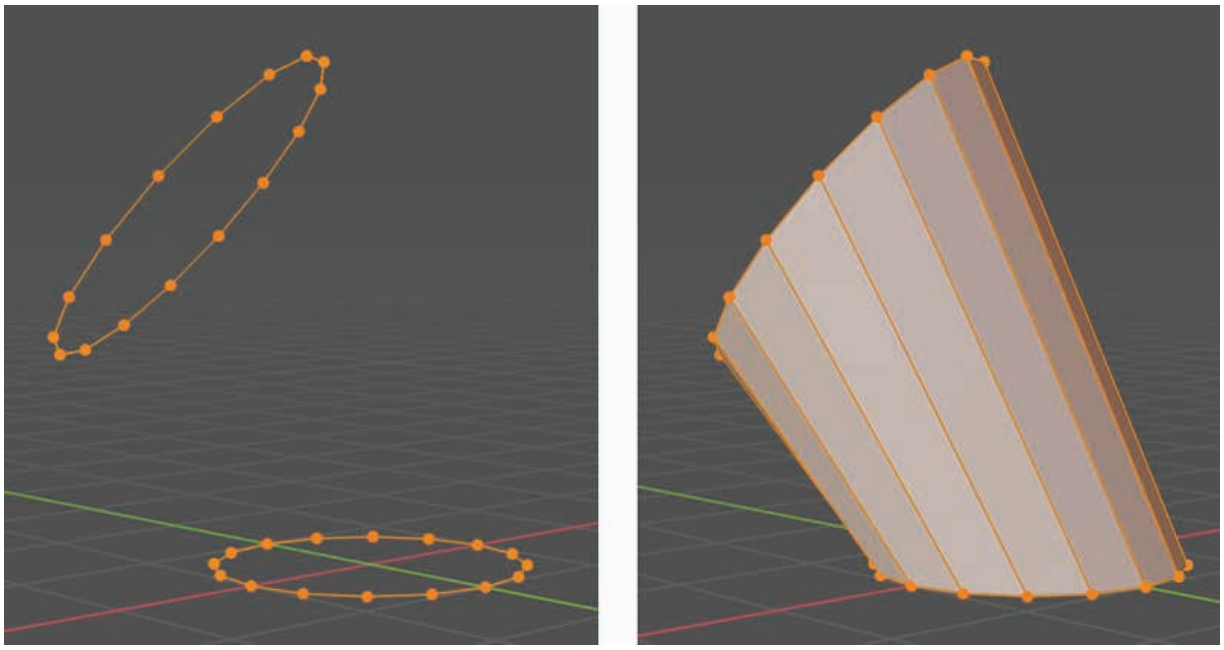


With the **Bridge Edge Loops** option you can also connect selected edge loops to make an airplane wing for example, as shown in another video.

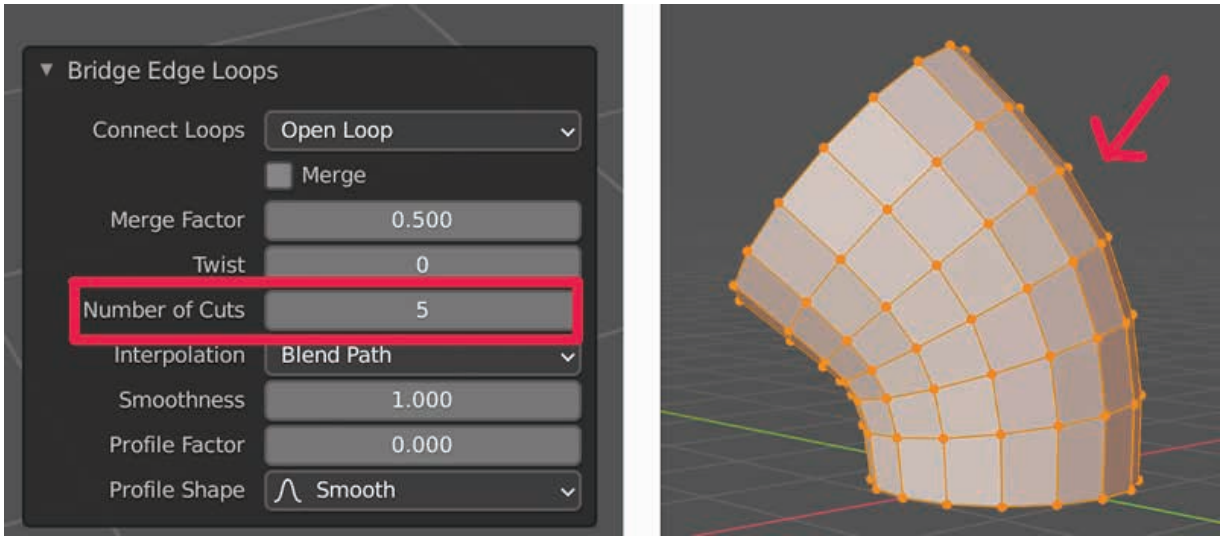


Click here to return to the index.

What's really interesting is that when connecting edge loops...

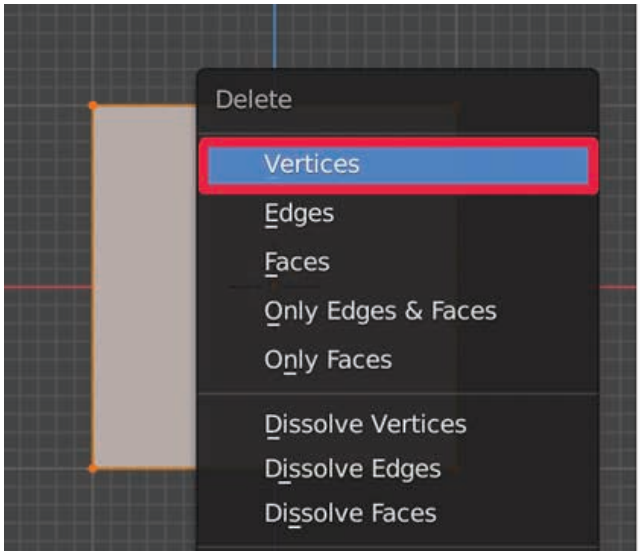


...if you add more **cuts** it will add curvature to the new geometry. This way you can quickly create organic shapes or smooth surfaces.

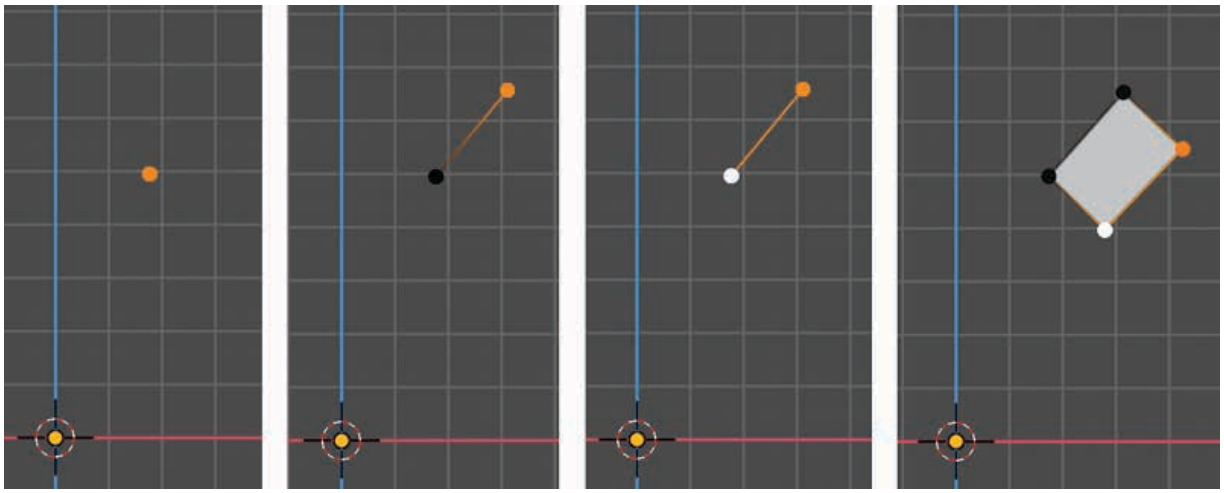


Make a coffee mug handle - Extrude to Cursor

Press Tab to enter Edit Mode and **press 1** to enter Vertex Selection mode. Select all by **pressing A** and **press X** and choose **Vertices** to delete all. Now, you can create new vertices by **holding Ctrl and right-clicking**.

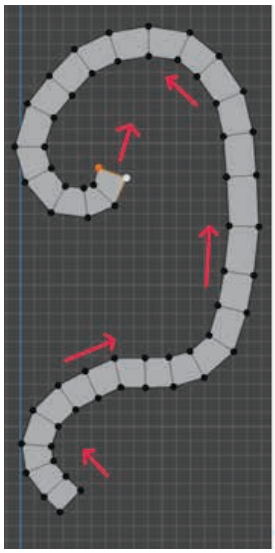


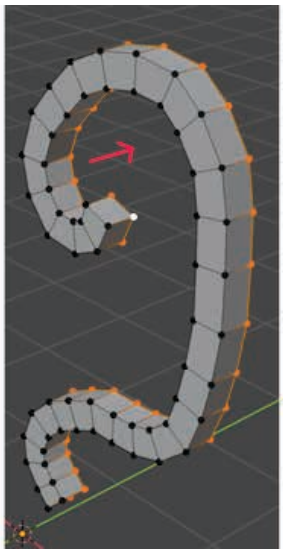
Similarly, with a vertex selected, you can extrude it by **holding Ctrl and right-clicking**. It will be extruded to where you click with the mouse cursor. With two or more vertices selected, the same action will create new faces.

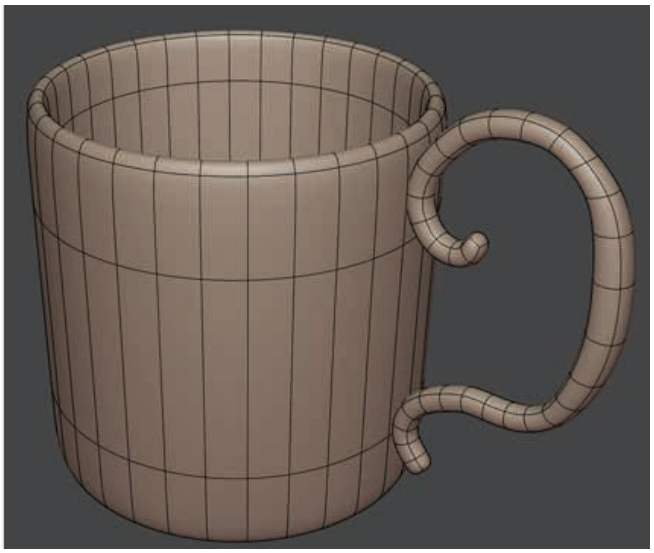


Click here to return to the index.

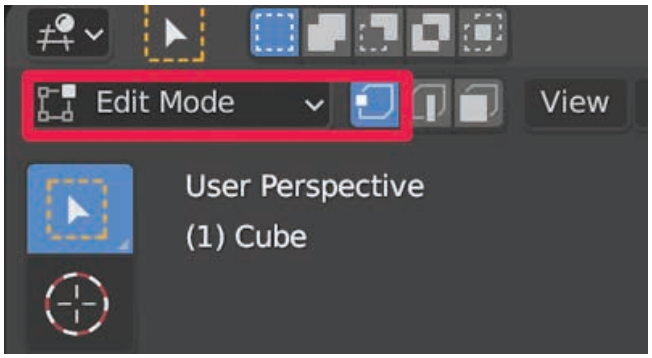
The new Face will be rotated based on where the mouse cursor is when you click. It's easiest to use it in an **Orthographic view**. For example to create the handle of a coffee mug.





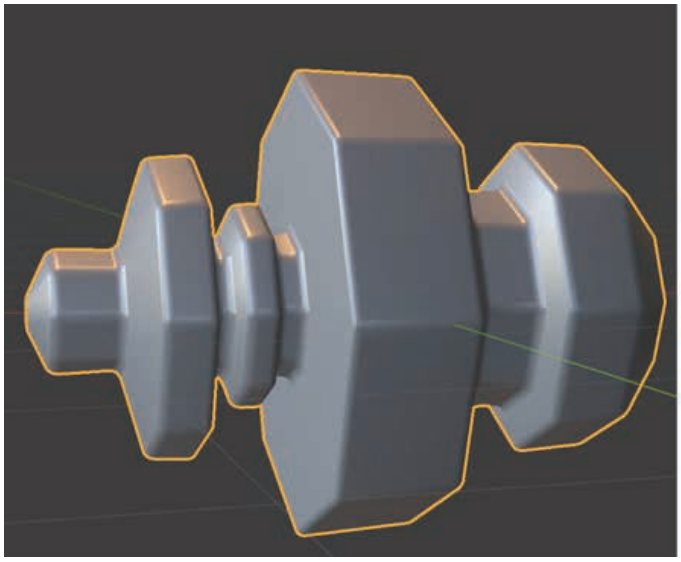


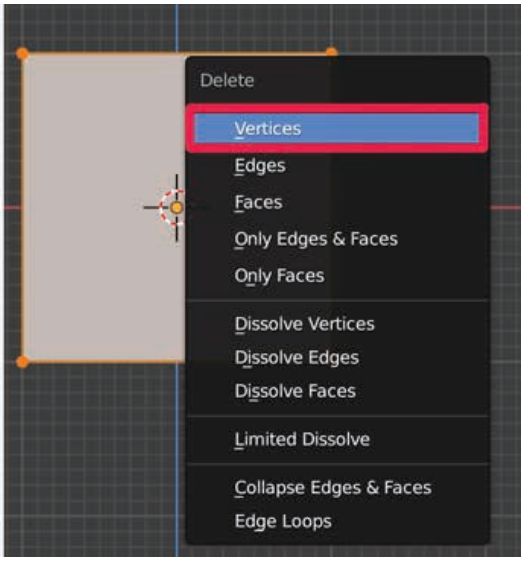
If nothing happens, check that you are definitely in Edit Mode and that **Vertex Selection** mode is active.



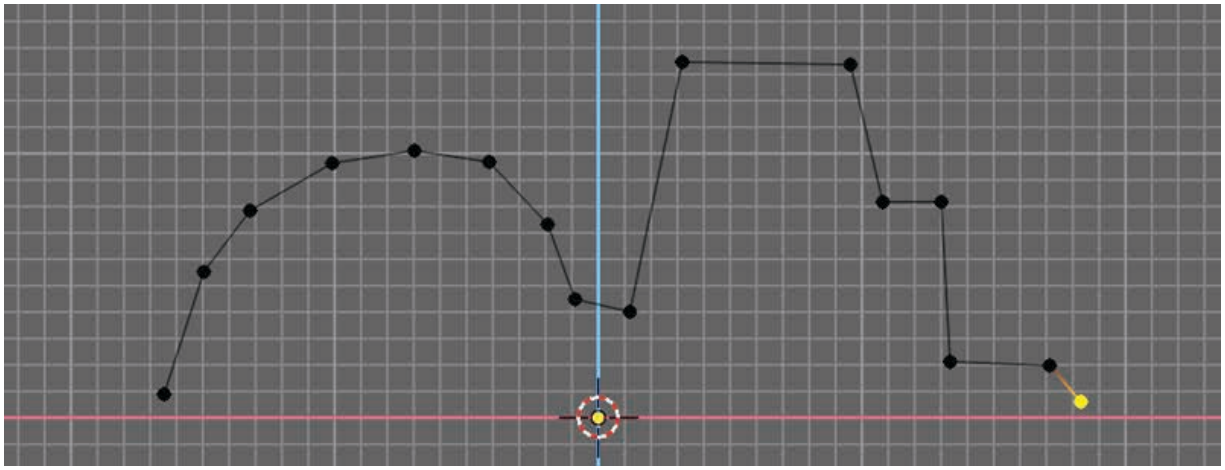
Modeling using Spin

Spin lets us easily create cylindrical objects. **Press 1 on the Numpad** to go to Front orthographic view. With the Default Cube selected **press Tab** to enter Edit Mode. **Press X** and choose **Vertices** to delete all.



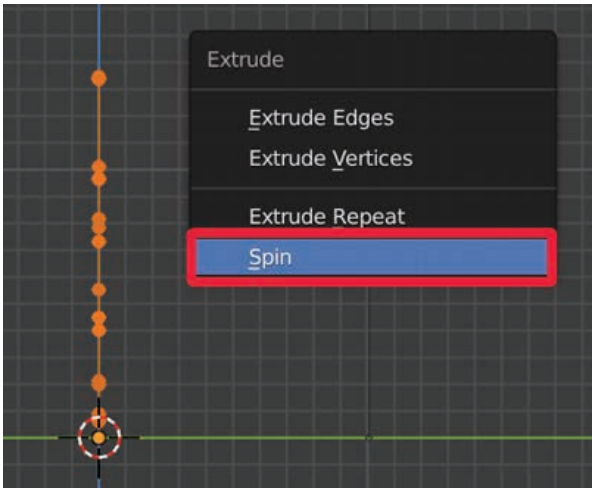


Then **while holding Ctrl, right-click** to create new vertices. Repeat to draw a profile. The 3D cursor acts as the center point, so stay above that.

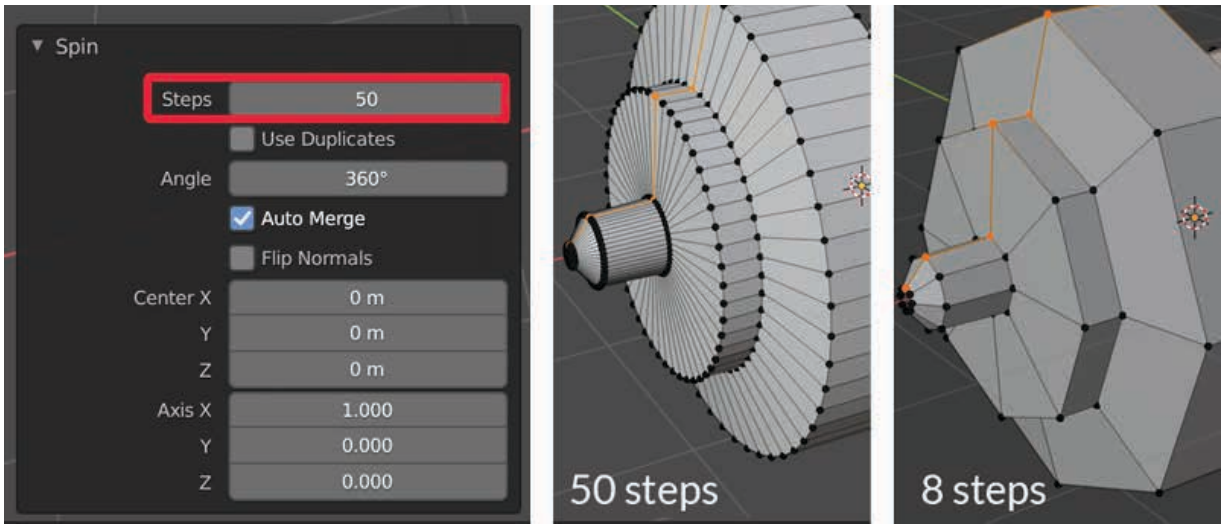


Go to the Right orthographic view by **pressing 3 on the Numpad.**

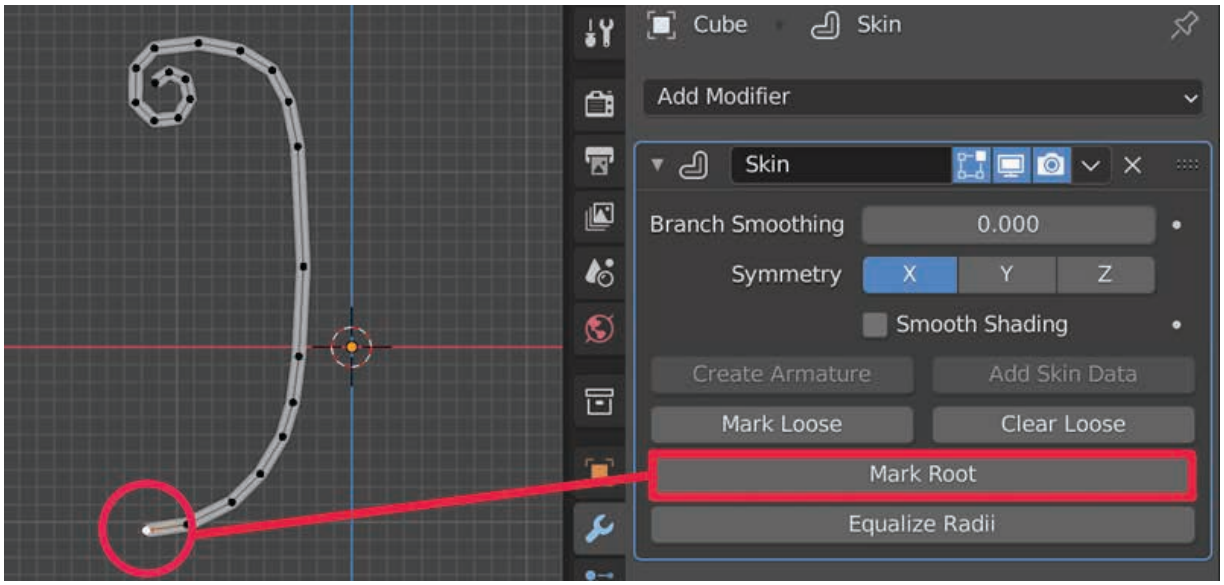
Press A to select all, then p**ress Alt+E** and choose **Spin**.



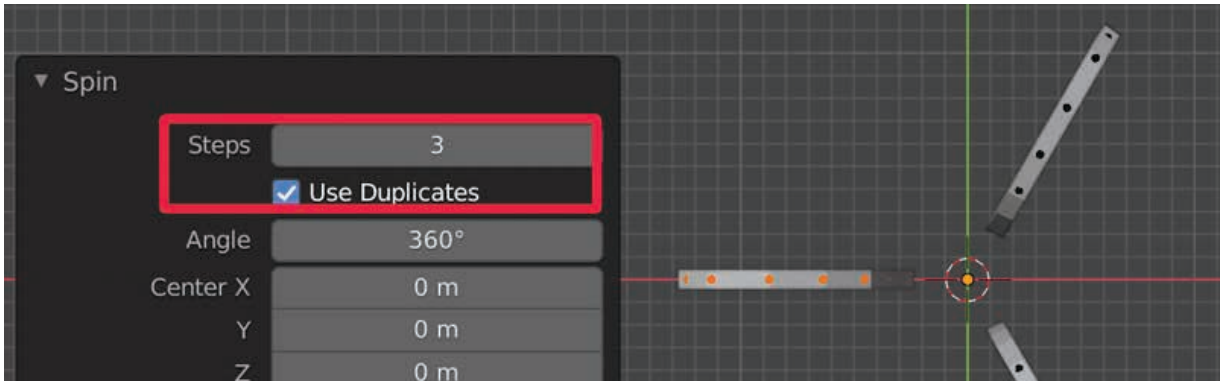
A menu appears where you can change some options like how detailed the object is (the **Steps** value) and whether you want to close it (**Angle**).



You can also choose not to fill the shape, by enabling **Use Duplicates**. This can be useful to make objects like a little table with legs, for example. To do this, add a **Skin modifier** and mark one vertex as **Root**.

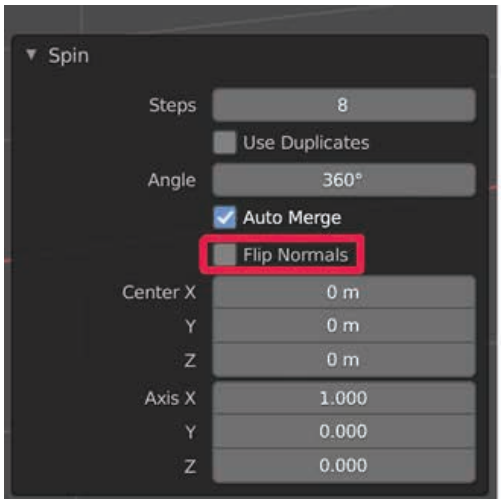


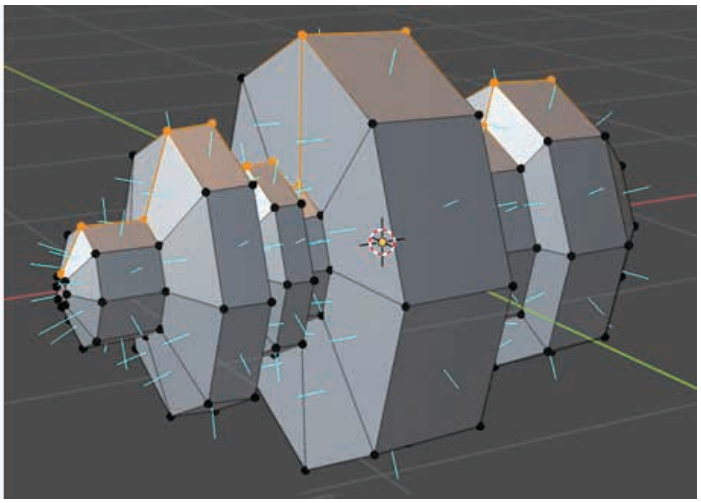
Then do the Spin operation by selecting all vertices, **pressing Alt+E** and choosing **Spin** (in this case from the Top orthographic view).



Not we get some “table legs” which we can then add a Subdivision modifier to, and a Cylinder for the tabletop.

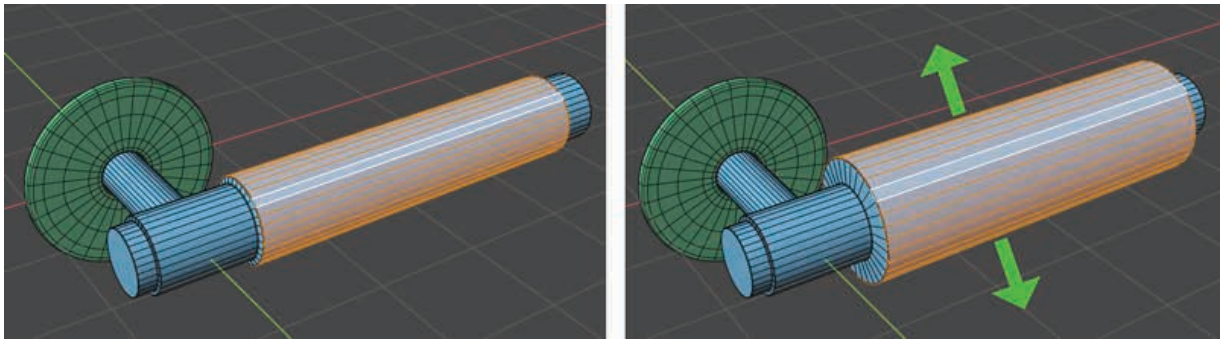
Bonus tip: If you create the profile from left to right, the Normals will point outwards after using the Spin option. If you created it from right to left, the normals will point inwards. In that case you can use the **Flip Normals** checkbox to flip the normals.





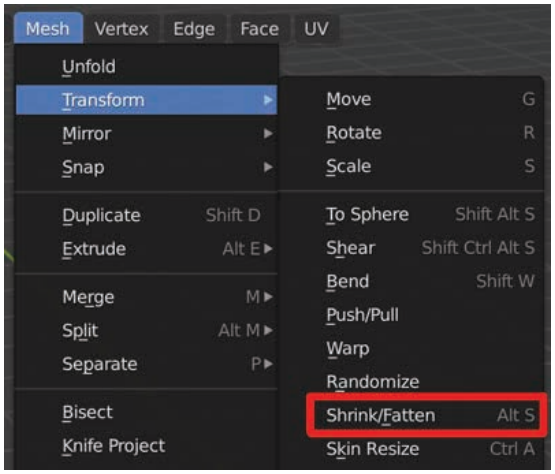
Shrink / Fatten

The **Shrink / Fatten** transform option is useful for when you want to make a Cylinder thicker without making it longer or shorter.

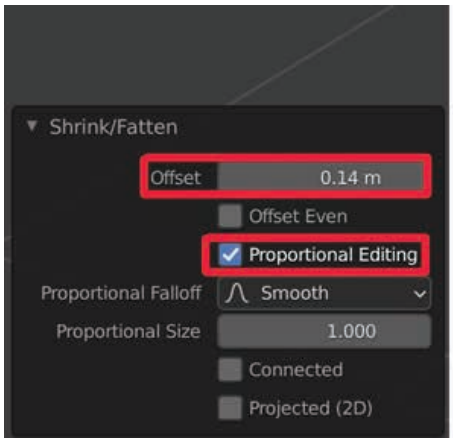


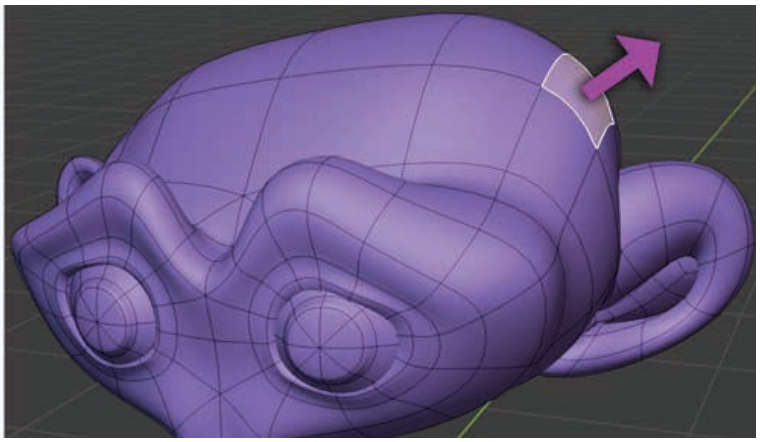
In the above example, only select the Faces you want to transform. Not the ones on either end of the Cylinder. Otherwise it will just scale as usual.

You can shrink / fatten a selection in Edit Mode by **pressing Alt+S** or find it in the **Mesh > Transform** menu.



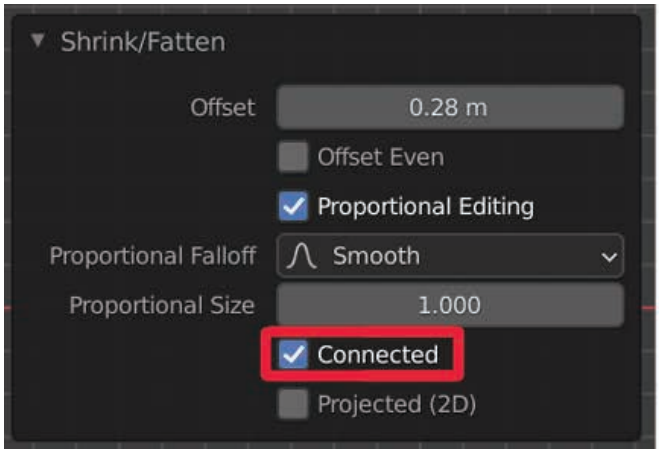
After **pressing Alt+S and left-clicking**, a small menu with options will appear in the bottom left corner of the viewport. Turning on **Proportional Editing** allows you to shrink and fatten a selection with proportional falloff. You can use the **Offset** value here to control the amount of shrinking / fattening.

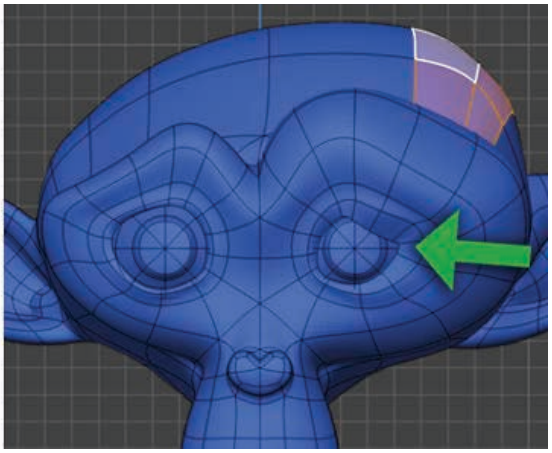




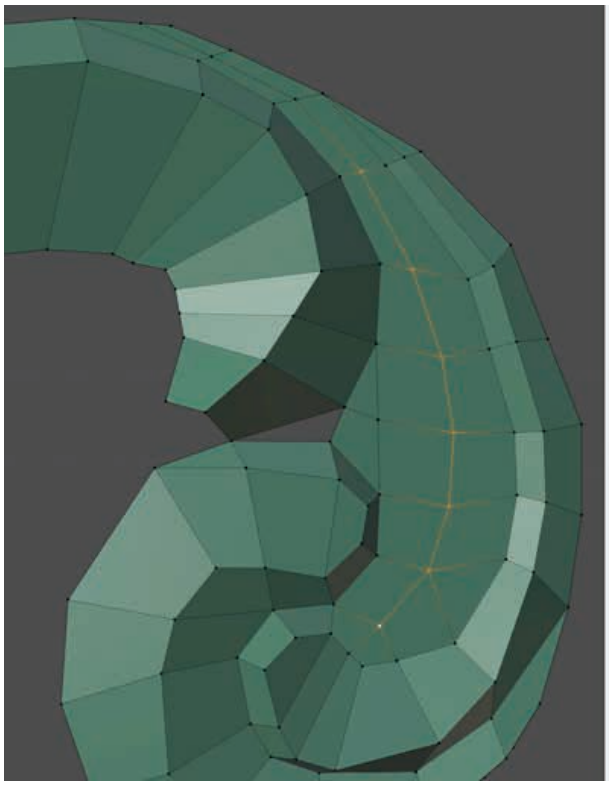
Click here to return to the index.

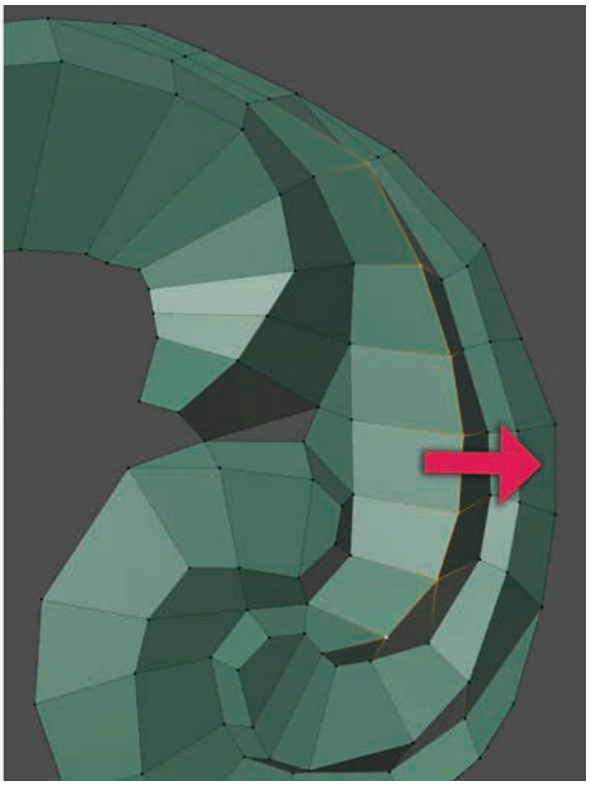
By checking the **Connected** option parts of the mesh that are not connected like the eyes in this monkey will not be affected.





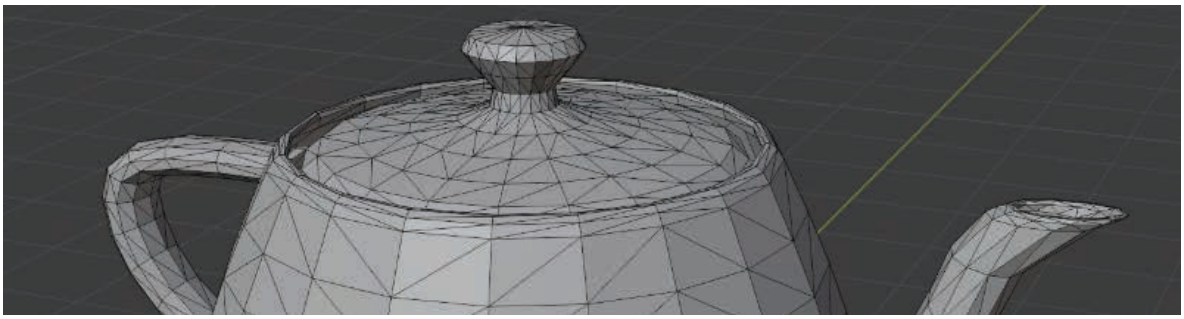
It can also be a useful way to move vertices for example the selection in this ornament model.



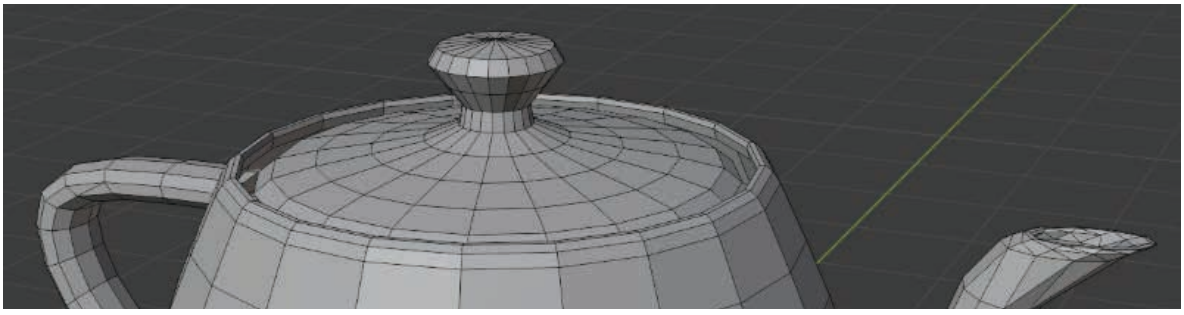


Convert Triangles to Quads

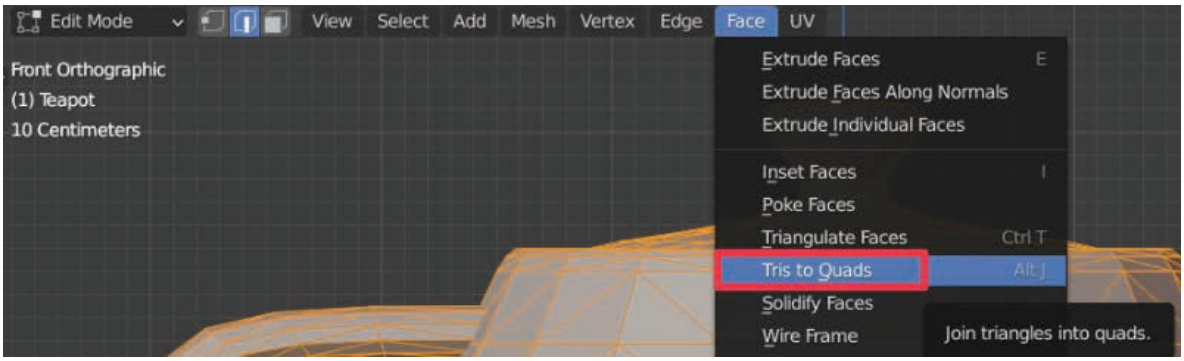
Let's say you've downloaded this model somewhere. When you go to Edit Mode you discover that it's all triangles!



We can fix that quickly. Select all in Edit Mode by **pressing A.** Then **press Alt+J** to convert the triangles to quads.



You can also do this through the face menu. Just make sure the faces are all selected. Note that it won't work in all cases, but it will at least take care of most of the triangles.



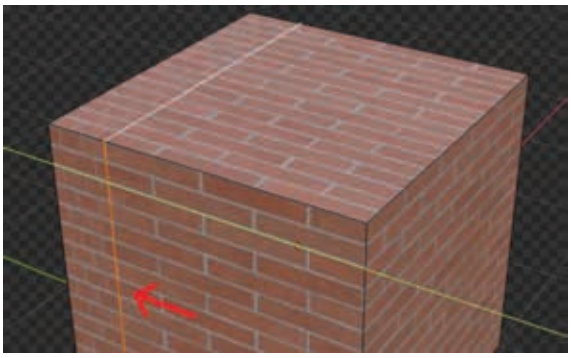
To do it the other way around: Convert quads to triangles by **pressing Ctrl+T**.

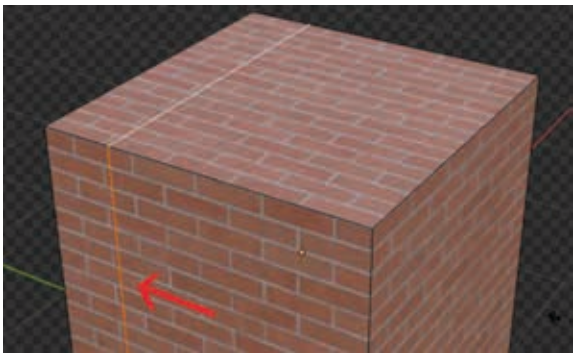
Move Edge Loops Without Destroying Your UV Layout

Want to move Edge Loops without warping your UV Layout?

Moving the Edge Loop by **pressing G** (or G + Y in this case, to keep the movement on the Y-axis), will stretch and squeeze textures.

The solution is to use the "edge slide" function, activated by **pressing G twice** instead of once.





Now the texture remains unstretched while you move the edge loop.

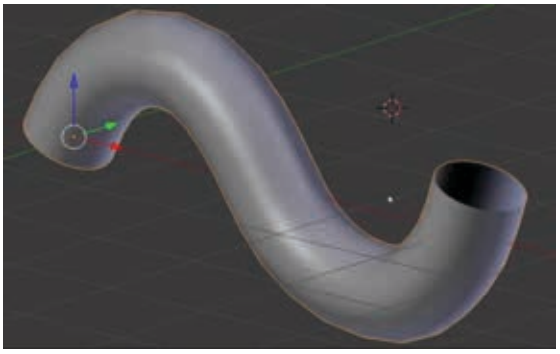
Scale Tubes and Pipes Proportionally

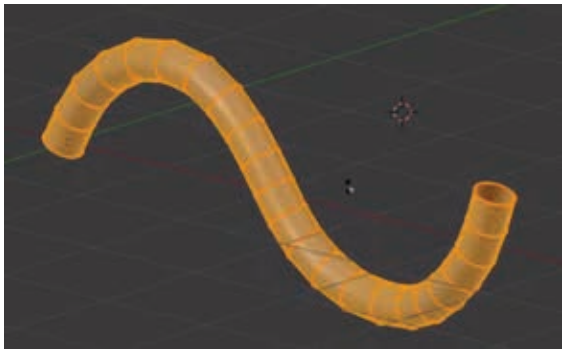
It's smart to make pipes with the Skin modifier and keep the modifiers in the stack so you can still make changes later.

But sometimes we don't have that option and we still want to change the thickness of a pipe object which is a normal mesh.

How do we do that?

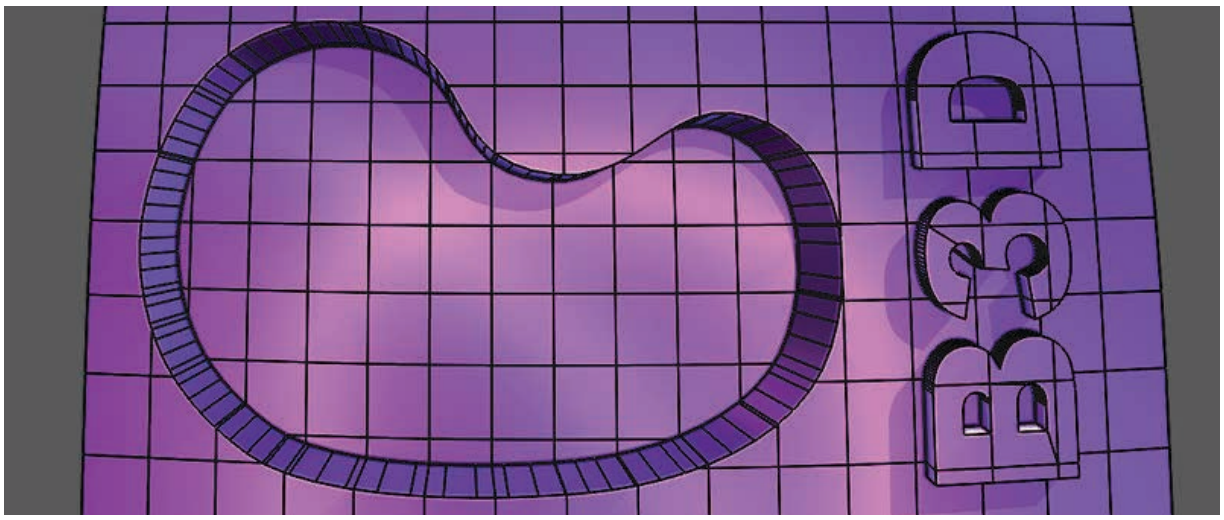
To proportionally scale pipes and similar objects, go to Edit Mode, **press Alt+S** and drag the mouse. This will not affect the overall size of the pipe, only its thickness.



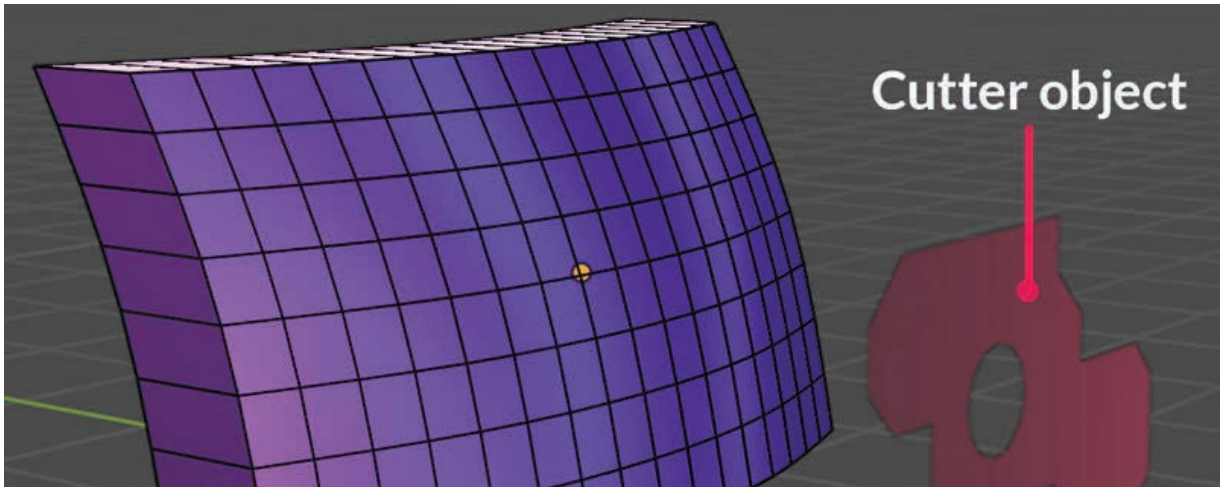


Knife Project

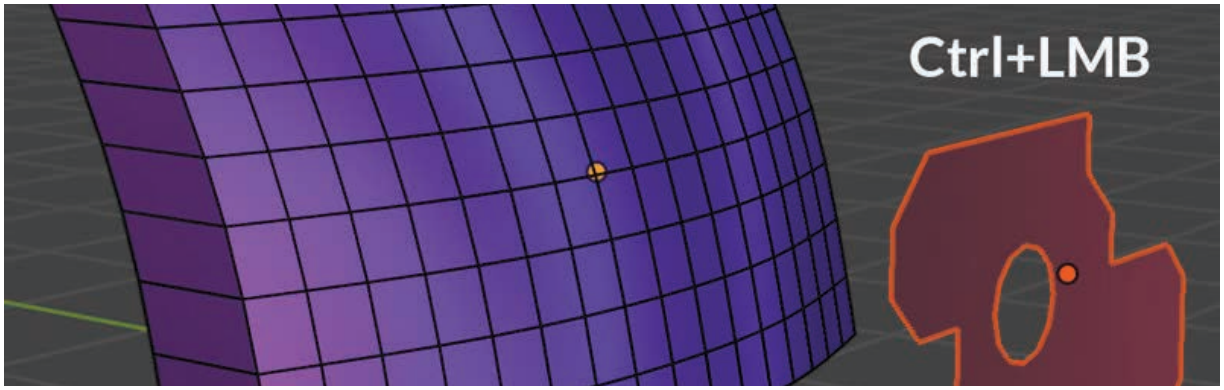
The way Knife Project works has changed since **Blender 2.93.4**. This is how it works currently (last checked in **Blender 4.3**). You no longer have to select both objects in Edit Mode. Here's a quick look at how it works. If you're looking for the old workflow instead, check this video.



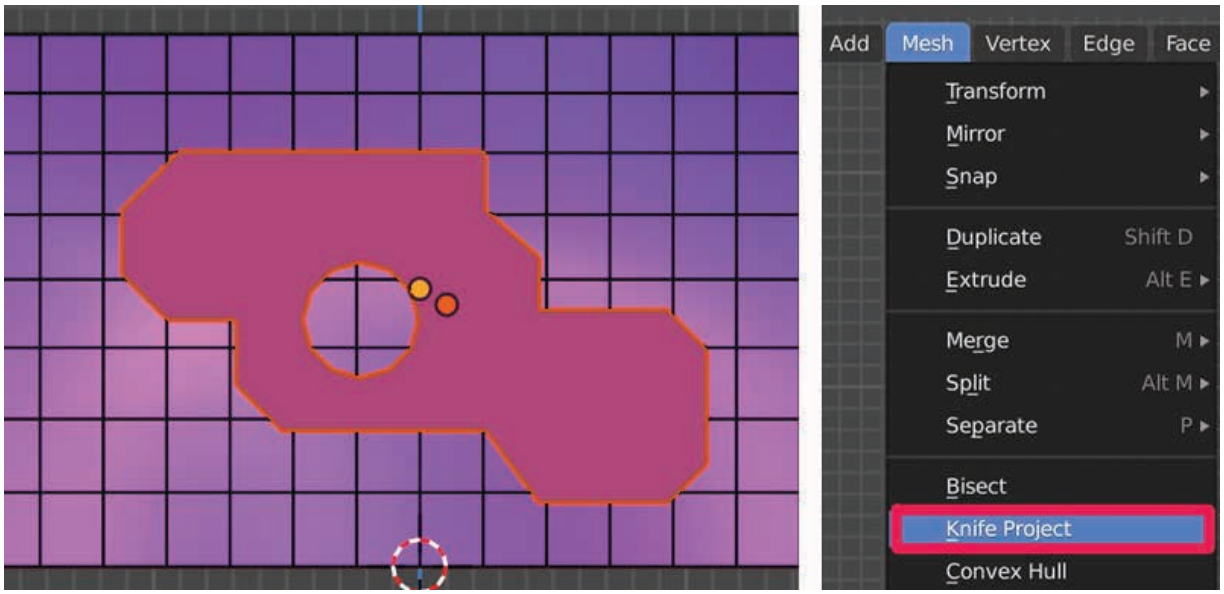
You'll need an object to cut into, as well as a cutter object. In this case I'm using a mesh as the cutter object. Select the object that you want to cut and **press Tab** to go to Edit Mode.



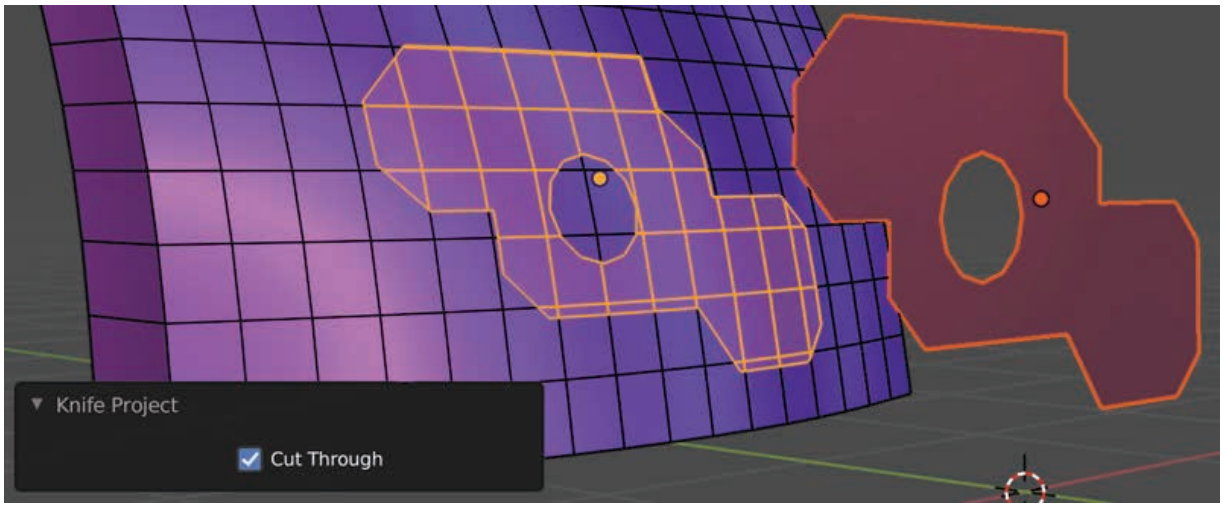
Then **hold Ctrl and left-click** on the cutter object to select it. This is easier if the cutter object is not visually overlapping the other object.



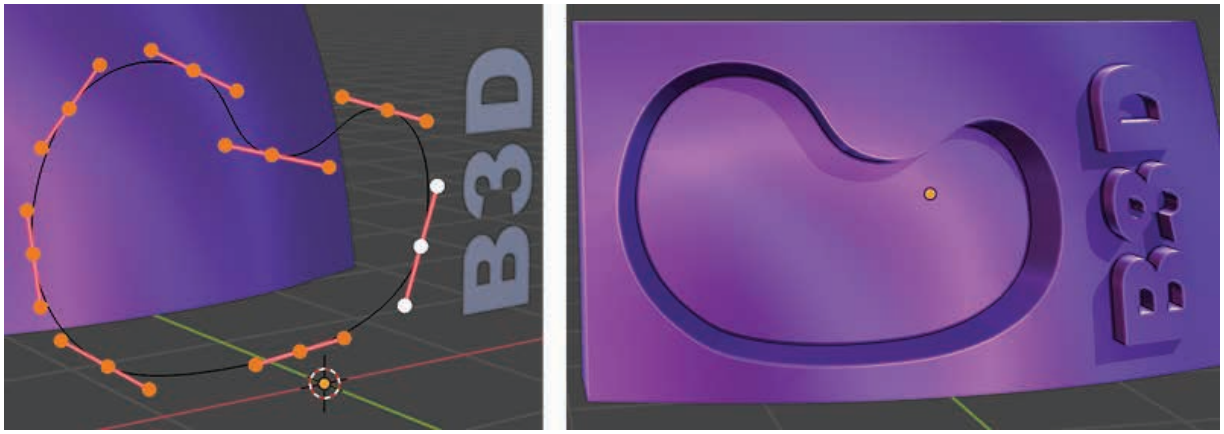
Then rotate the view so the cutter is placed where you want. Orthographic views work well for this. Go to **Mesh > Knife Project** to place the cut.



The part where the cut was made will be selected, which is convenient if you want to extrude or inset it, for example. If you want the cut to go all the way through the model, enable **Cut Through**. Otherwise, only the front-facing Faces will be cut.



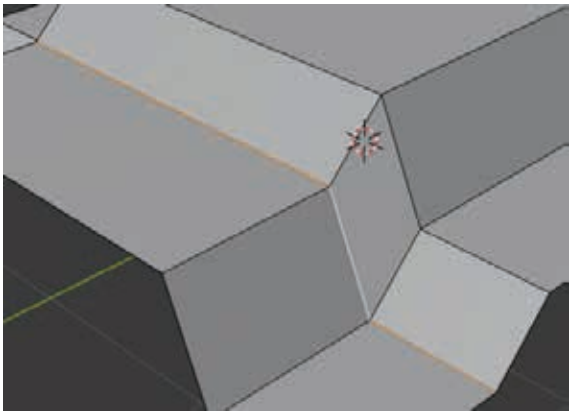
Besides meshes you can also use **Curves** as cutter objects. **Text** objects work as cutters as well, without having to convert them to meshes.



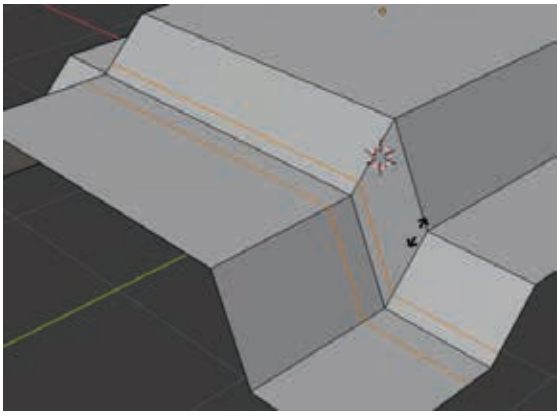
Offset Edge Slide

This is a quick way to add support loops when doing Subdivision surface smoothing where you want to control the sharpness of some edges.

Alt-click on an edge to select the entire loop.

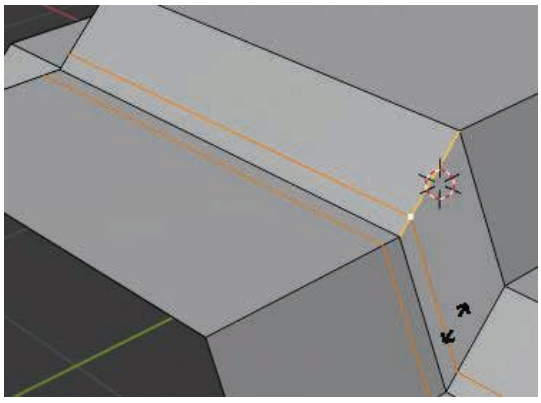


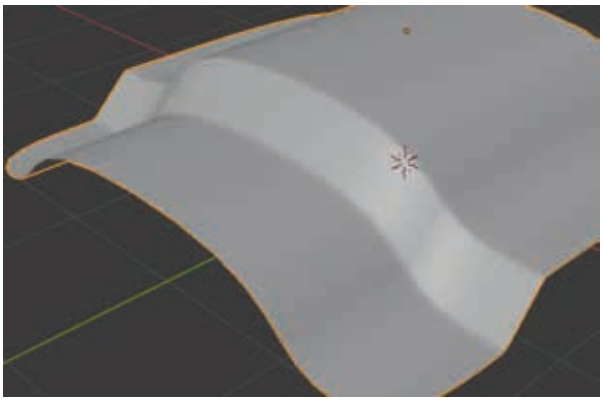
Press Ctrl+Shift+R to create two edge loops on either side of an edge.



Press E to make sure they are the same distance from the original loop.

These are called "support loops" and when combined with a subdivision smoothness modifier will keep the sharpness of the edge they are close to.

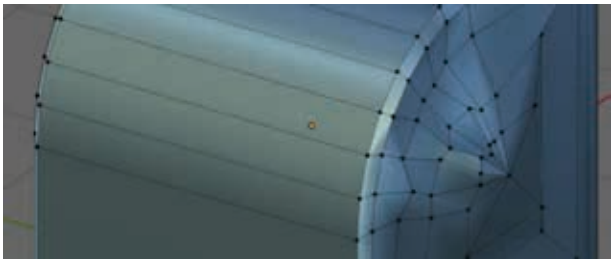




The same surface with a Subdivision Surface modifier and Smooth Shading turned on. ***Right-click*** *on the object and choose "****Shade Smooth****" and* ***press Ctrl+3*** *to add a Subdivision Surface modifier set to 3 subdivisions.*

Flatten Faces

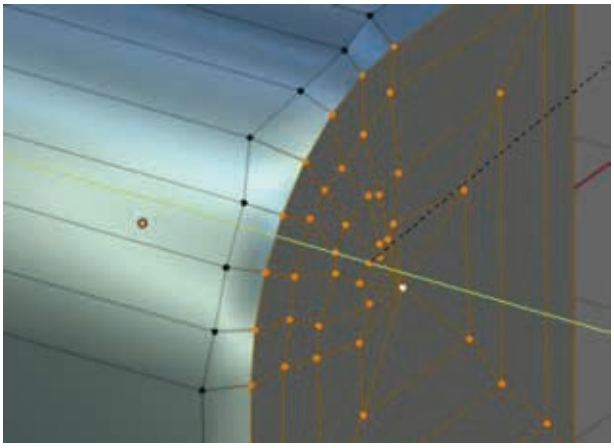
Sometimes you need to flatten several polygonal faces together.

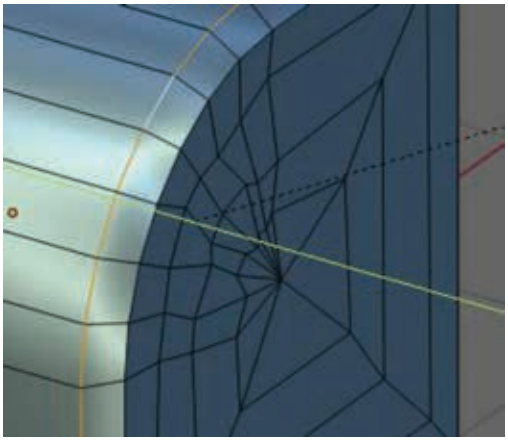


In this case I select one vertex (the one in the center) and expand with **Ctrl+numpad "+"** to expand the selection.

Press S (scale) and **X** (so it's restricted to the X-axis) and **0** (zero) so that the selection is scaled to zero on the X-axis.

To take care of the edge, go to Edge select mode (Ctrl+2) and double-click on one edge to select the entire loop. **Press S+X+0** to scale that edge loop to zero.



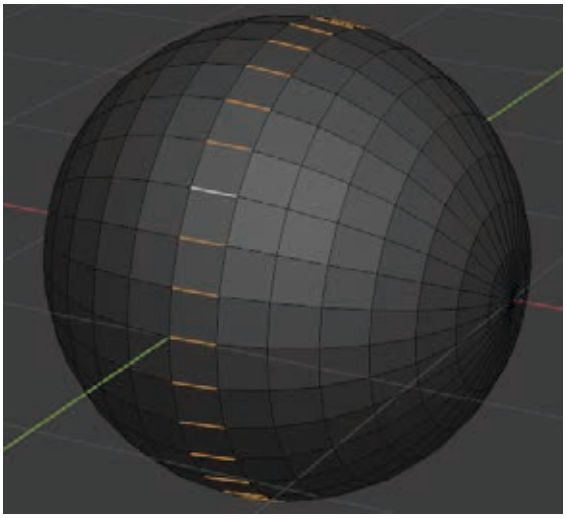


Alternatively, you can use the **LoopTools extension / add-on** which comes with Blender by default and use the "Flatten" function when you have all the faces selected.

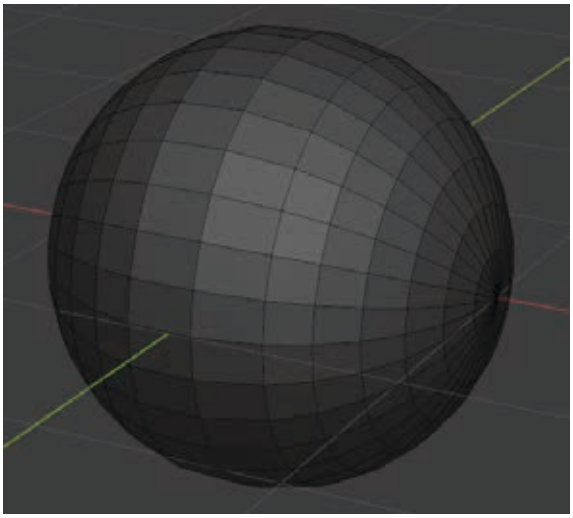
Collapse Edges

Holding Ctrl, double-(left)-click on an edge to select the ring of edges. *(Note: this is Ctrl+Alt+Left-click if you don't have "enable 3-button mouse" enabled in Preferences).*

Press X, then C to collapse those edges.



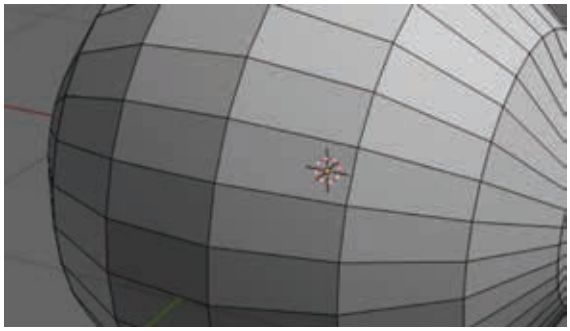
Now the edges have been collapsed.

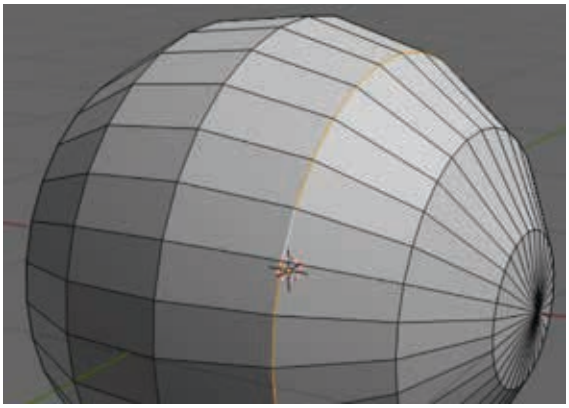


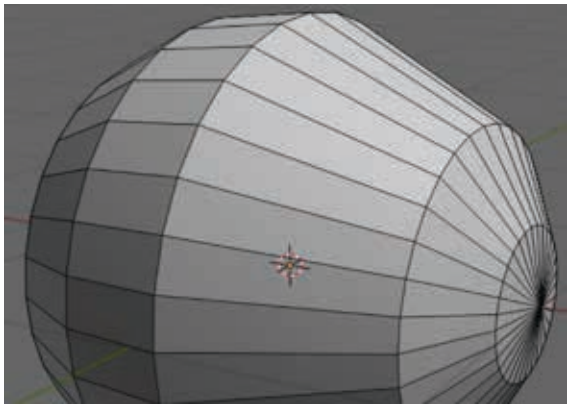
Let's repeat that. Select edges, **Press X** and then C.

As you can see, this reduces the geometry.

Alternatively, you could select the edge loops going in the other direction by double-clicking on it, **press X** and choose "dissolve edges" or just **press Ctrl+X** to skip the menu.



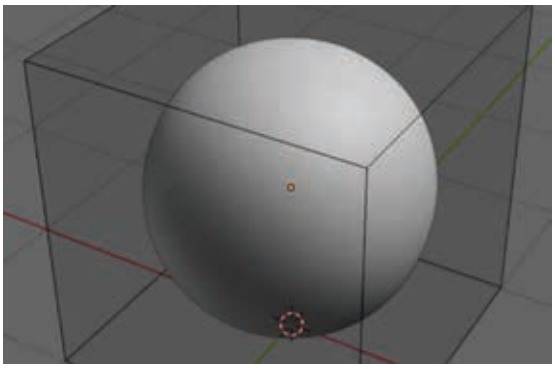




Crease Edges

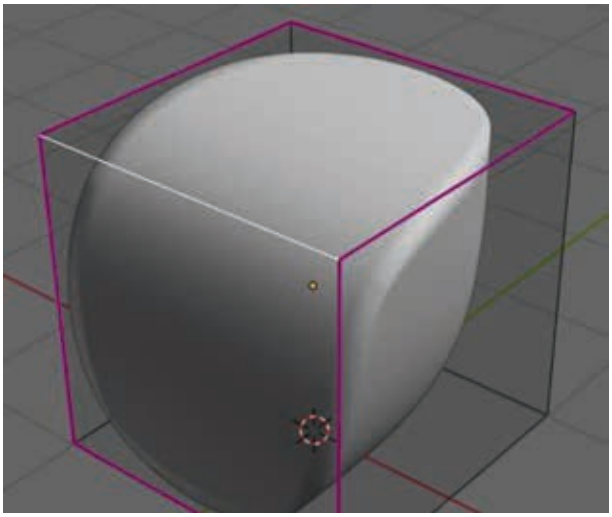
This is the default cube, with a Subdivision Surface modifier set to 4 iterations and Smooth Shading, so it's become an almost perfect sphere.

In Edit Mode you can still see and select the edges that make up the cube.



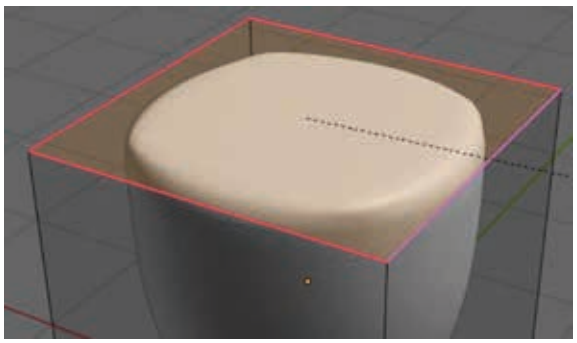
If you want the edge to be 100% sharp, you can also **press Shift+E and then 1**. The creased edges will have a purple color.

To set the creasing value of an edge back to zero, **press Shift+E and -1**.



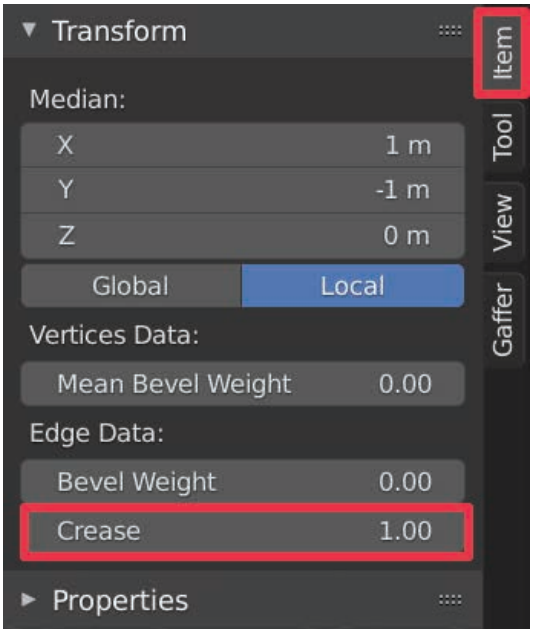
You can make edges sharp in a non-destructive way with the Crease option.

Select the edge(s) you want to sharpen and **press Shift+E** and then move the mouse to increase the creasing.



You can also find the crease value in the Option Panel.

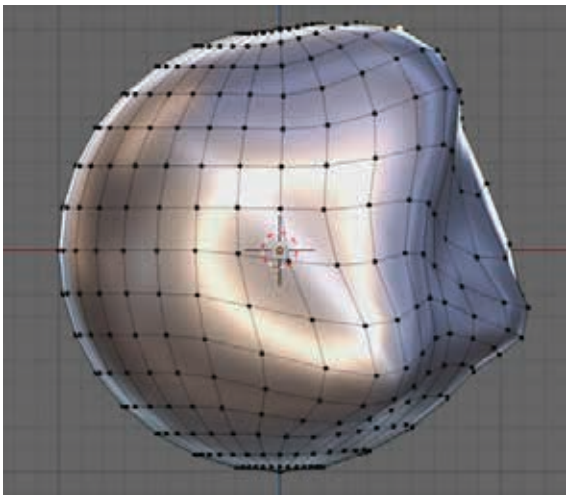
Press N to open it and under **Item** find the **Crease** value.



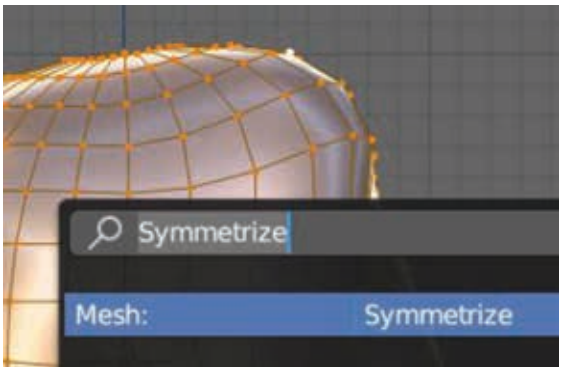
The good thing about this is that you can adjust it later, by **pressing Shift+E** and dragging the mouse in the different direction.

Symmetrize

If your mesh needs to be symmetrical, you can fix it easily with the Symmetrize function.

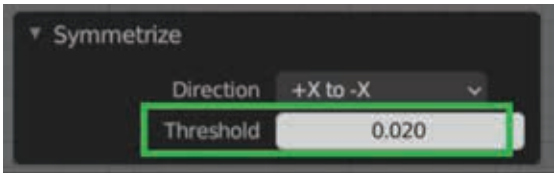


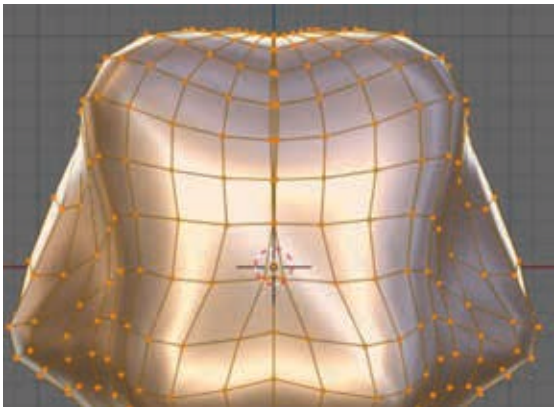
Select all the vertices in Edit Mode and **Press F3.** Search for “Symmetrize”.



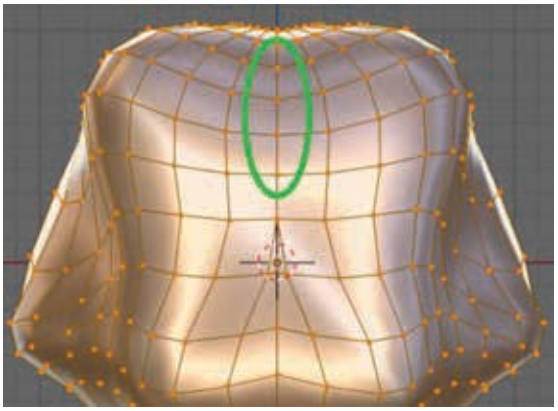
You can change the direction in which the symmetrize action is performed in the little tool window that temporarily pops up in the viewport. In this case, +x to -x worked.

There are some overlapping vertices in the middle of the model.





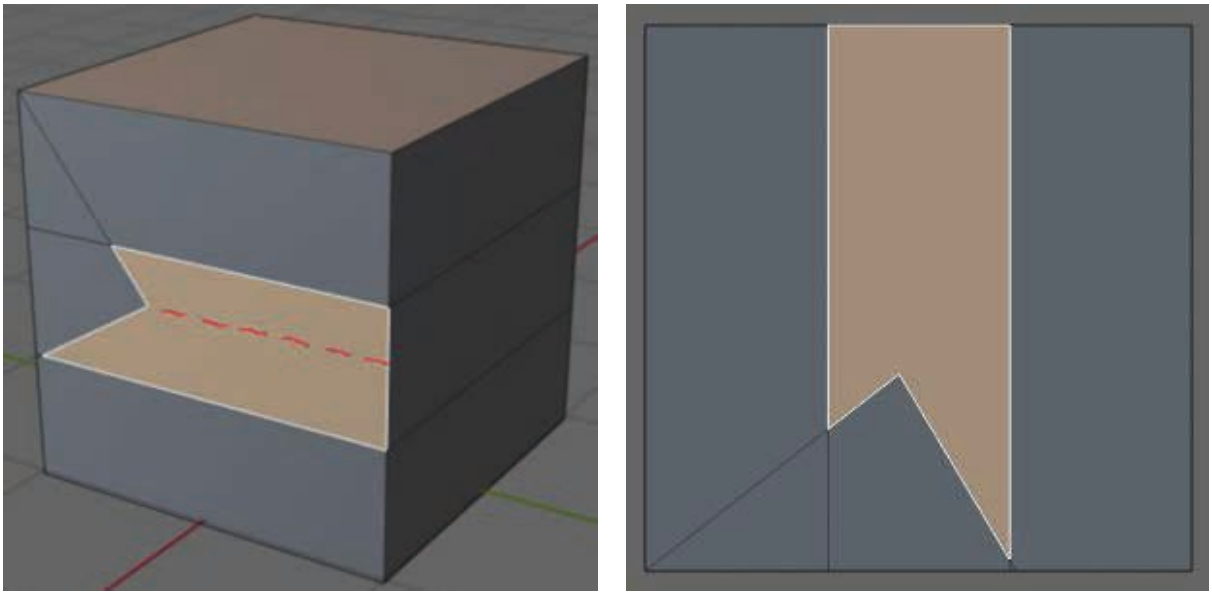
Increasing the Threshold value merges these vertices.



Of course, you could also delete half of the model and add a Mirror Modifier for a more non-destructive workflow.

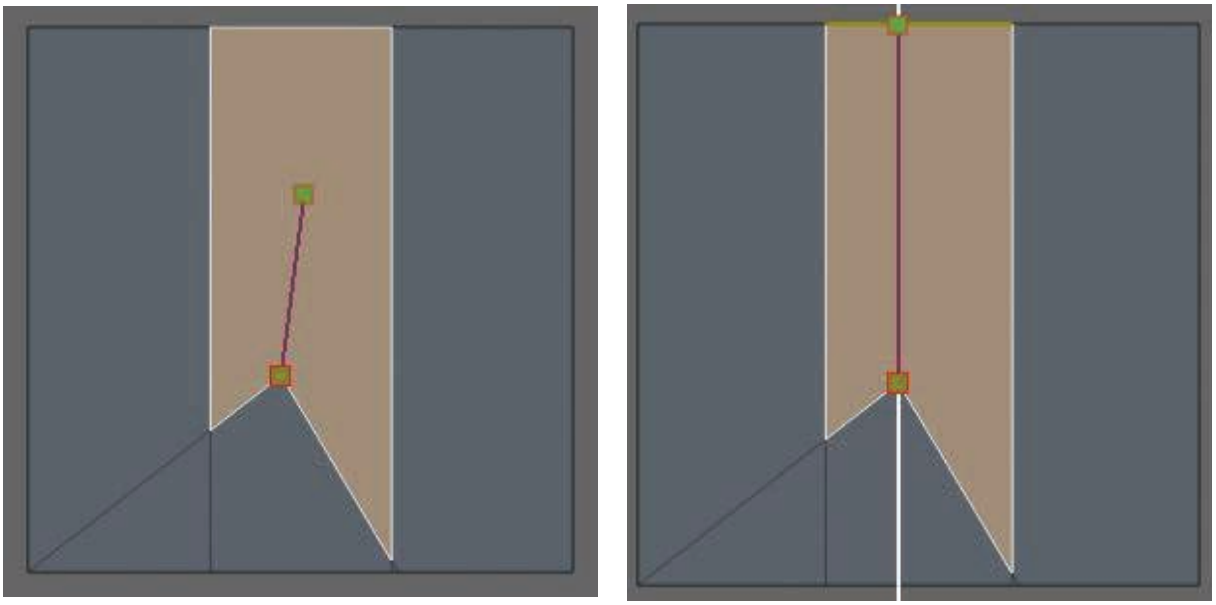
Angle Constraint

If you need to cut a straight line using the Cut Tool (shortcut: **K**), align the view to the selected face of an object (**press Shift+7**) first.



Left: the red line indicates where we want to cut. Right: The object aligned to view.

Then **Press K** to start making your cut and **press A** *(Note: in older versions of Blender the shortcut was C)* to constrain the angle while cutting.

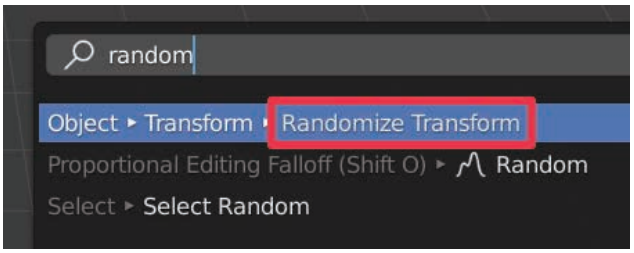


Left: Starting to cut. Right: The white constrain line helps us cut straight.

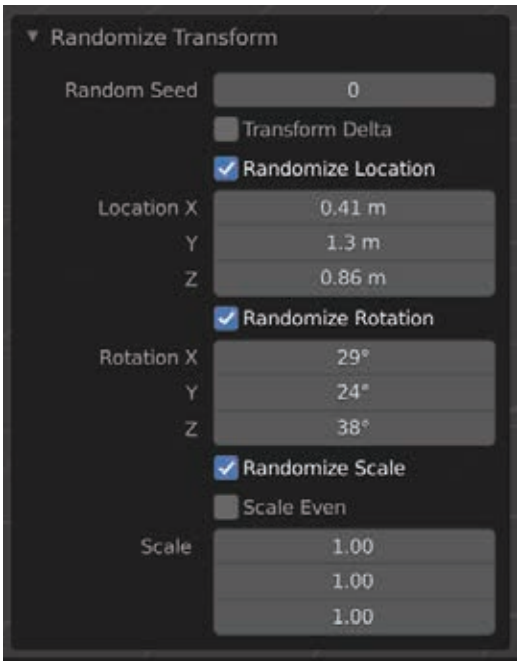
To use this Constrain trick to its benefit, you should always use it from an ortographic view (front, side, top, bottom). **Bonus Tip**: You know you can press for example "3" on the numpad to go to the Right ortographic view. But press **Ctrl+3** to go to the opposite side, the Left ortographic view.

Randomize Transform

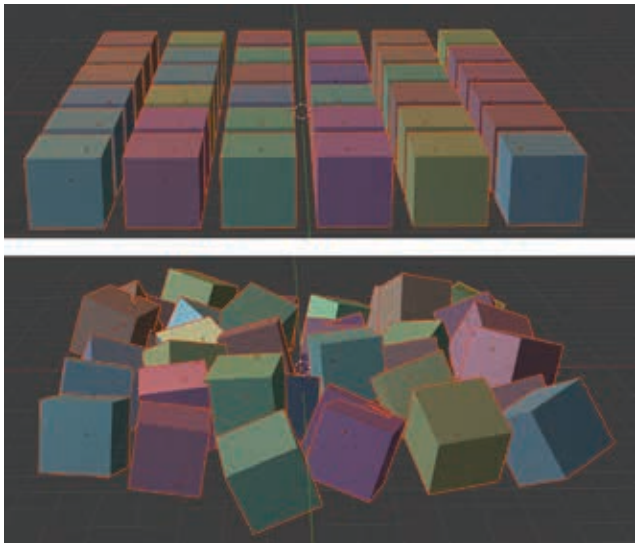
If you want to randomly move (“transform”) some objects, **press F3** and search “Randomize transform”. Select it from the search results andpress **Enter**.



You can enter values of how strongly you want to randomize the transformations.



You can Randomize the Location, Rotation and Scale separately or together.



You can also choose different **Random Seed** values to get some different results. What is this good for? Well, for example - imagine you have a bunch of spaceship

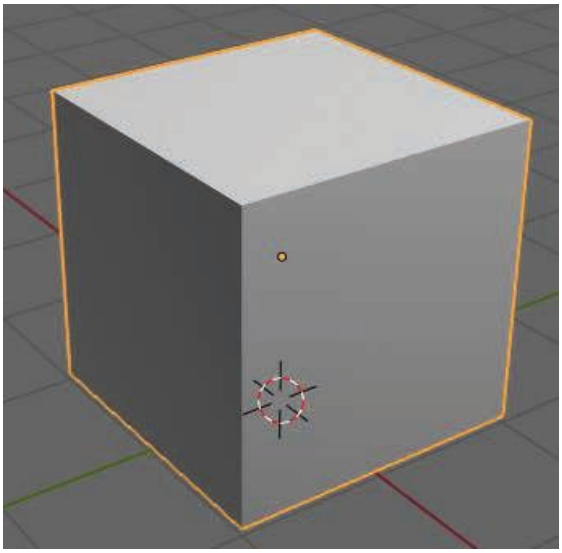
parts or random tech parts and you want to quickly conceptualize something randomly.

You can quickly get them in some random order and go through different Seed values to see different random configurations of those objects until you get inspiration for some spaceship or robot design.

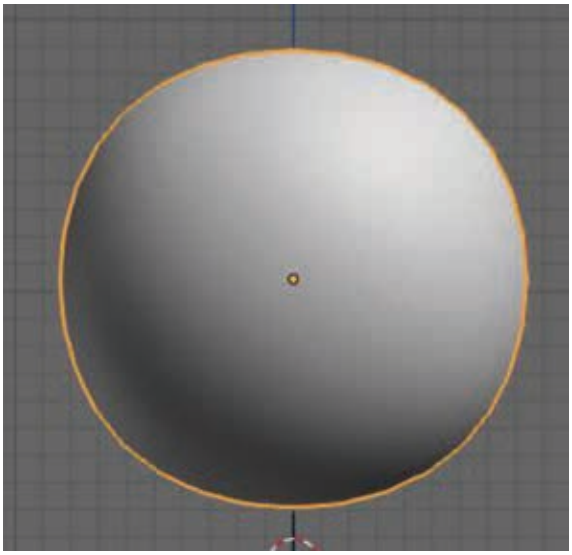
You can use that as the basis for concept art, for example.

Sphere With Quads

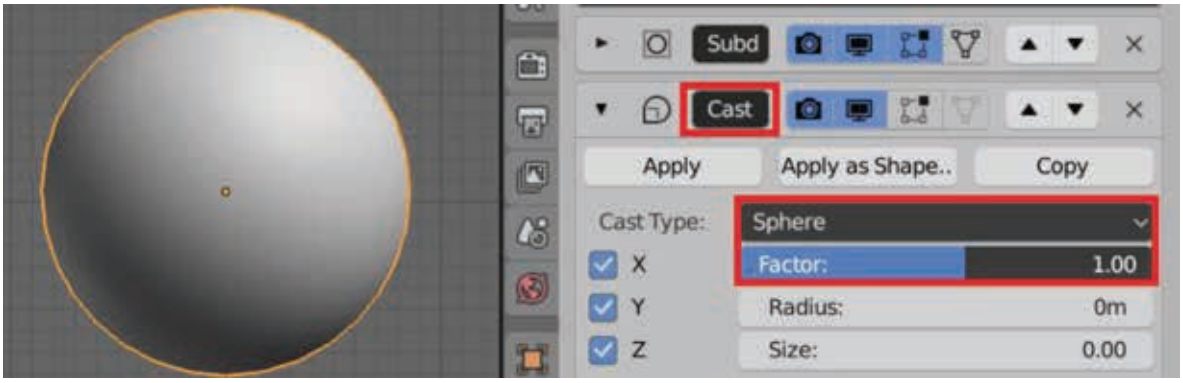
Start with a default cube. Add a Subdiv modifier to the cube to get a sphere.



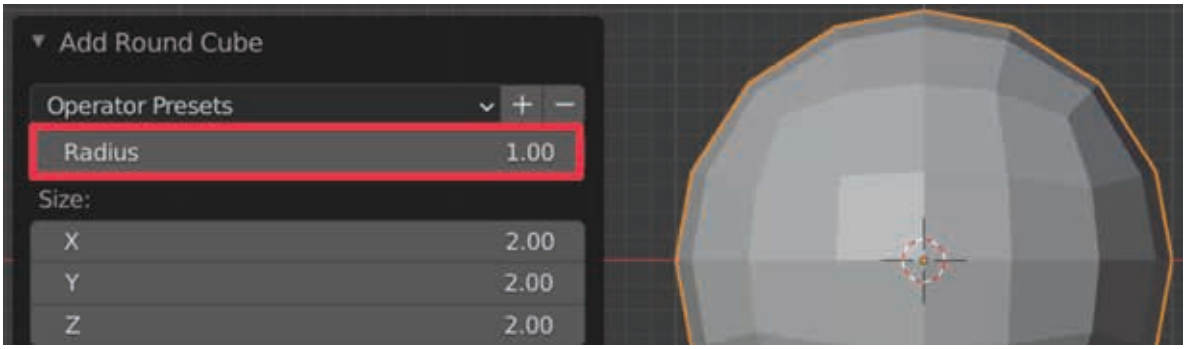
However, it is not perfectly round. As you can see it's still a bit square.



Adding a **Cast Modifier** set to **Sphere** and **Factor: 1** makes the cube perfectly round.



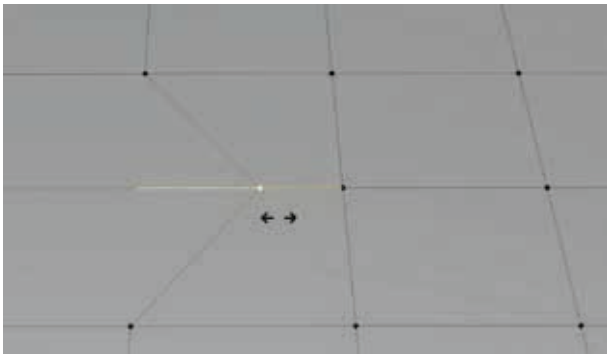
Bonus Tip: By enabling the "**Extra Mesh Objects**" Extension in Preferences, you get a "round cube" as one of the extra primitive objects. You still have to set the **Radius to 1** to make it perfectly spherical.



Slide Vertices Along Edges

To slide a vertex along an edge, first select the vertex.

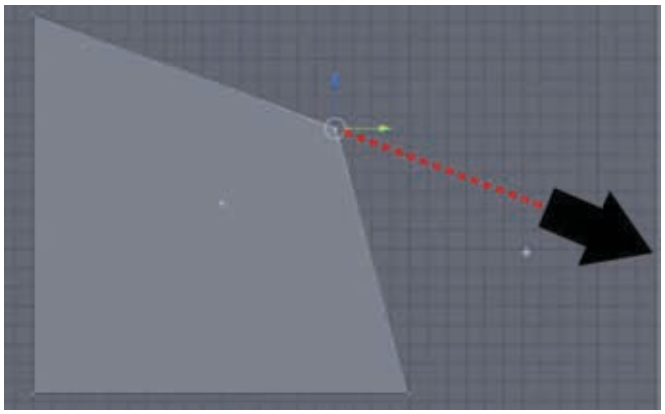
Then, **press G twice** to slide it along an edge.



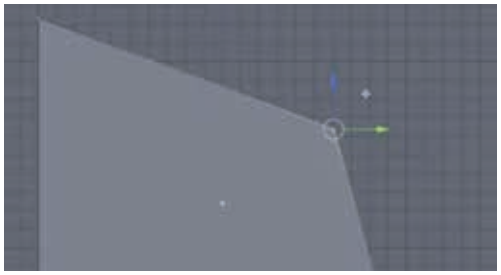
Once you've pressed G twice, you can change the direction of the sliding by moving the mouse cursor.

Sliding Vertices to where there's no edge

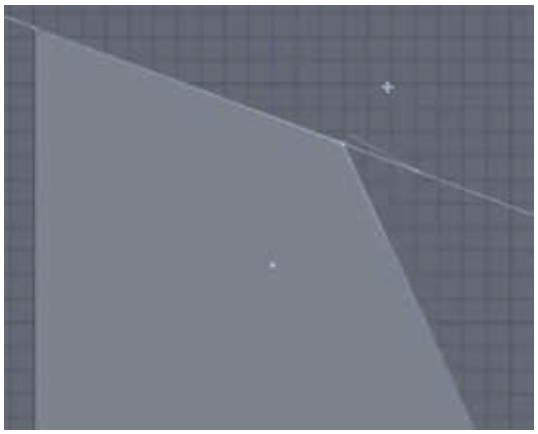
Or how to edge-slide a vertex along an edge that doesn't exist...



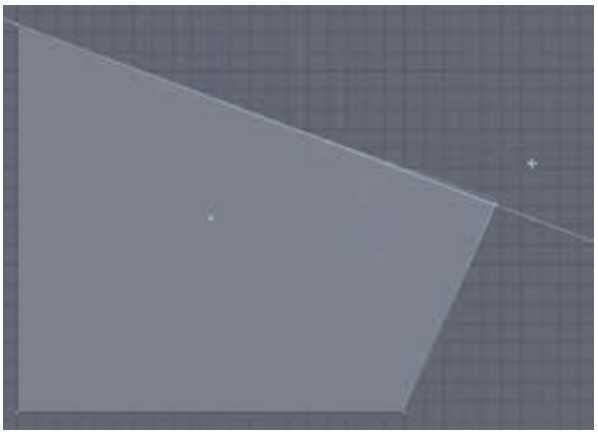
Select the vertex that you wish to move.



Press G twice and move it up along the edge, then **press C** and move it down in the direction where there is no edge.

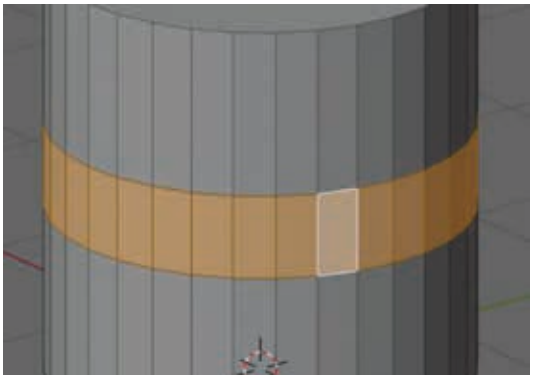


Pressing C shows the line that continues the edge in the direction you want to slide the vertex. Now you can slide it in that direction.

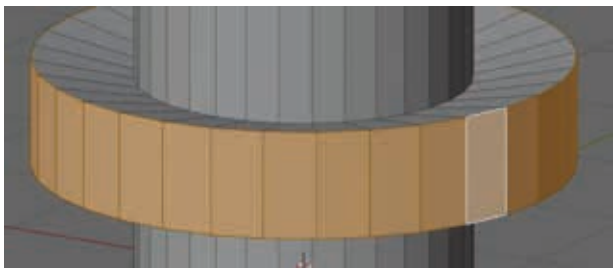


Extrude faces along Normals

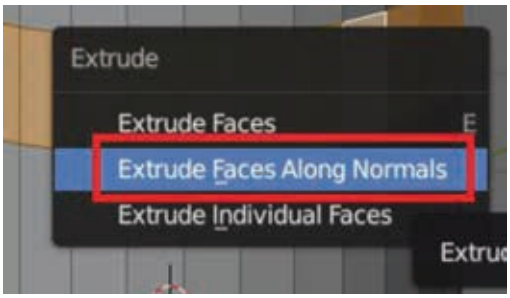
To select a loop of faces, first go to Face select mode. Now **Alt+Left-click** on one face in the loop to select the entire loop.



You can extrude the loop outwards by pressing **E** to extrude and then **S** to scale. As you can see, this not only scales the extrusion outwards, but the faces themselves are scaled up which may not be desired.

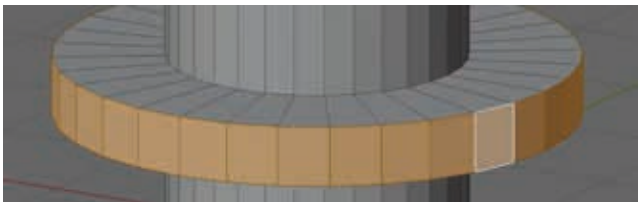


Undo twice (**Ctrl + Z**) to go back to having just the selection of the face loop. Now instead of pressing **E** to extrude, **press Alt + E**. This menu appears:



Choose **Extrude Faces Along Normals**. Now, the extruded faces remain consistent in scale (at least on the Z-axis).

Alternatively, you could scale the faces back along the Z-axis after extruding, but it would be less precise.

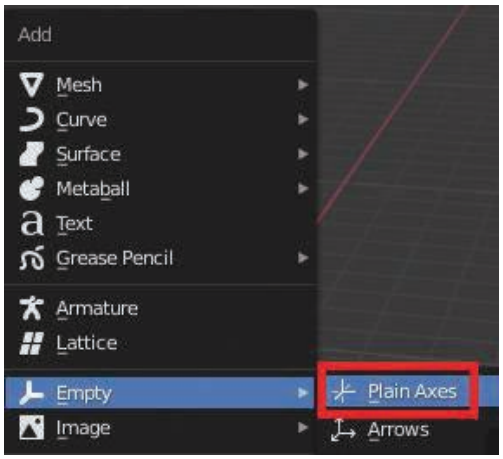


Click here to return to the index.

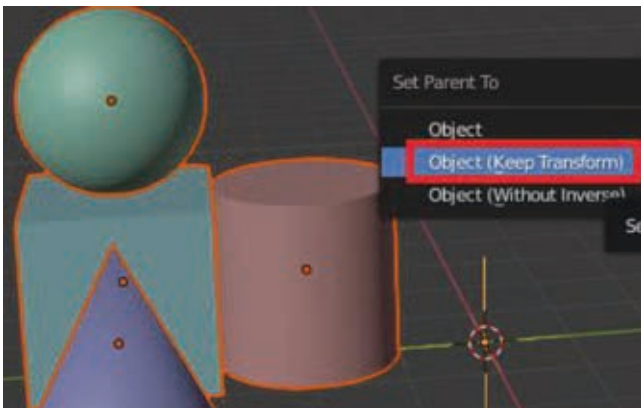
Using Empties For Transformations And Mirroring

Let's say we want to mirror or transform (move/scale/rotate) all these objects at the same time. We can use an Empty to do this more easily.

Create an Empty: **press Shift+A** > Empty > Plain Axes (the type doesn't really matter).

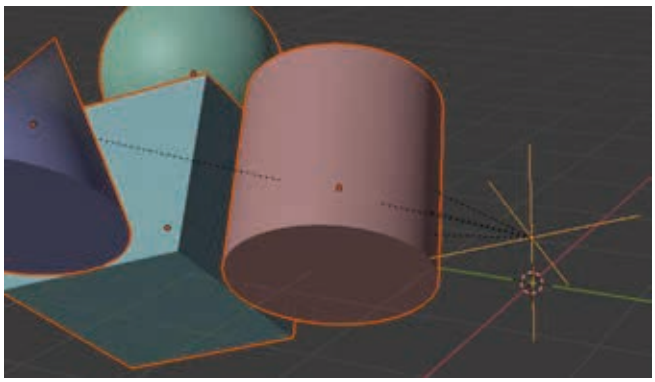


Select the objects you want to transform while holding Shift and select the Empty last. **Press Ctrl+P** to parent. Choose "Object (Keep Transform)".



Now you can use the empty to transform the other objects all at once. The empty itself won't render. Also useful is to use an Empty as the Center Point for a Mirror Modifier.

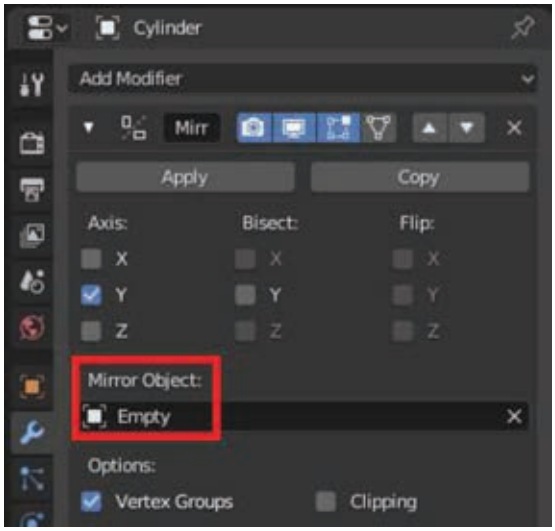
You can see the black, dotted "relationship lines" between the Empty and the objects that are parented to it. These also won't render, it's just for reference in the Viewport.



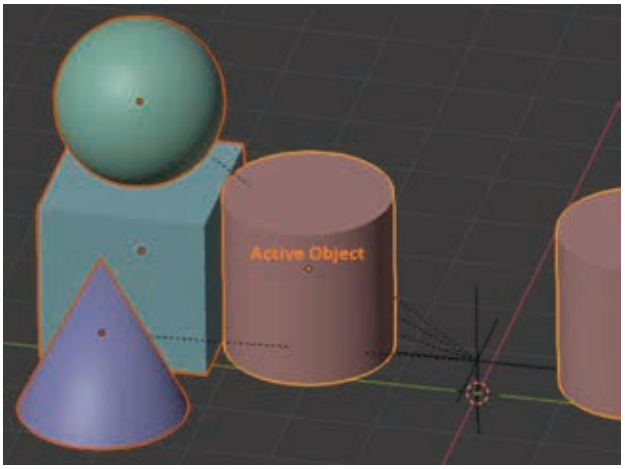
Note: make sure to clear the parenting to the Empty if you want to try the next steps:

Select the objects, **press Alt+P** and choose **Clear and Keep Transformation**.

Add a **Mirror Modifier** to an object. In the modifier choose the Empty as the **Mirror Object**.

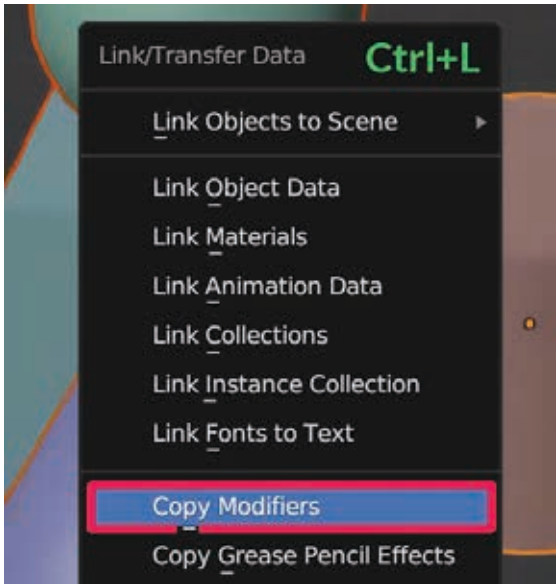


Here I select the other objects while **holding Shift**, and the object with the mirror modifier on it is selected last. Being selected last makes it the "Active" object.

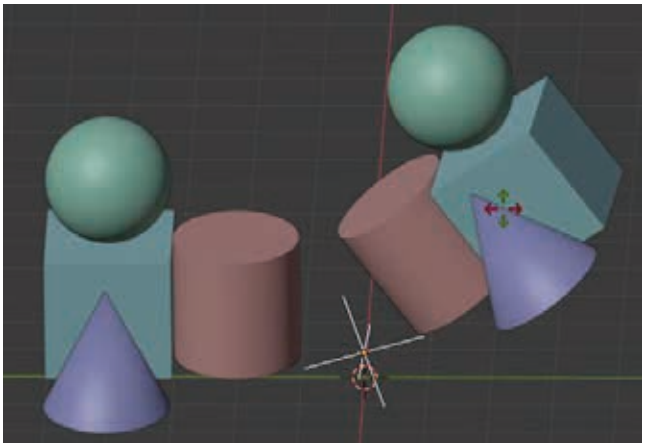


This is especially handy if you need to mirror several objects along the same point without having to adjust the origin point of those objects.

Press Ctrl+L and choose **Modifiers**. This copies the modifiers of the active object to the other selected objects.

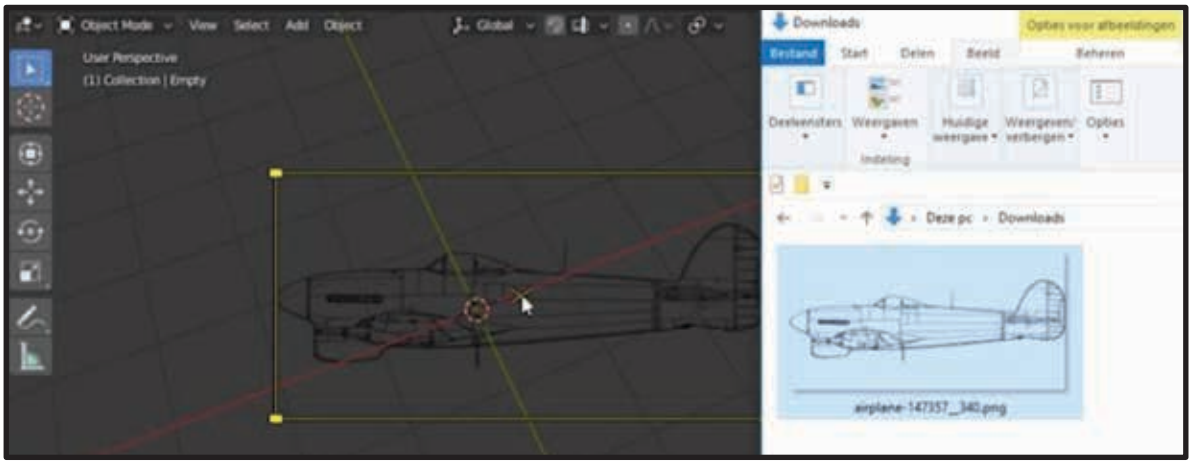


You can now use the empty to control which point the mirror modifier uses as its centre.

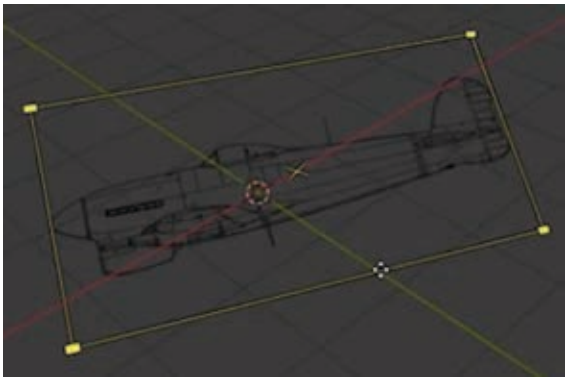


Import Reference Images For Modeling

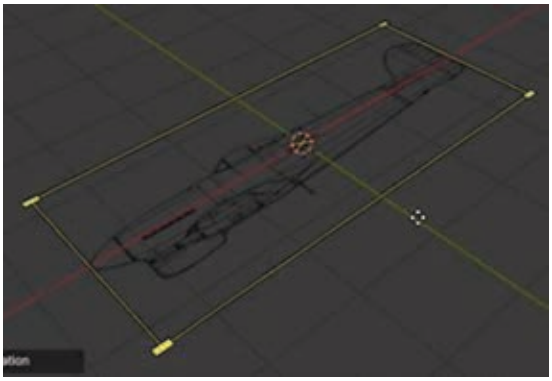
To quickly import reference images into Blender you can drag the image from a folder window to the Blender interface.



Select it and reset the rotation (**Alt+R**) and location (**Alt+G**).



This places the reference image flat and centered onthe3D plane.



If you drag the image into an orthographic view, you won't have to reset the rotation. You can resize it with the yellow handles at the edges. This object is actually a kind of Empty.

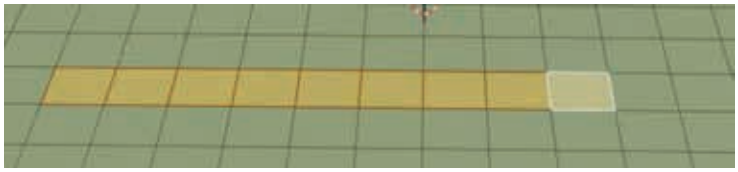
Now you can use this as modeling reference. If you have reference images for the front- and top-view, that would be even better.

Select Rows Of Faces Easily And Quickly

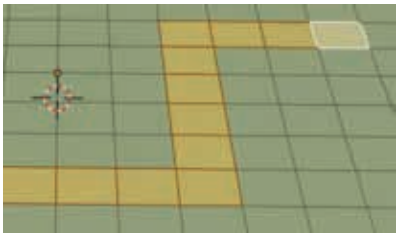
Instead of holding Shift and selecting individual faces one by one...

Click on one face to select it.

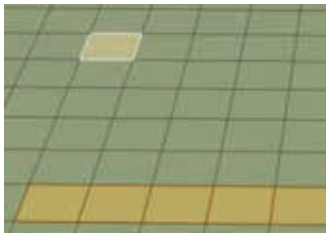
Click on another face while **holding down Ctrl**. The faces in between are also selected.

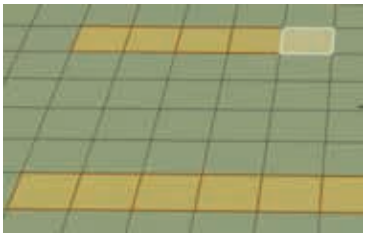


You can keep Ctrl-clicking and adding rows of faces to your selection.



To add to the selection without adding the faces in between, **hold Shift** and click on a face. You can then continue holding Ctrl and selecting more faces.

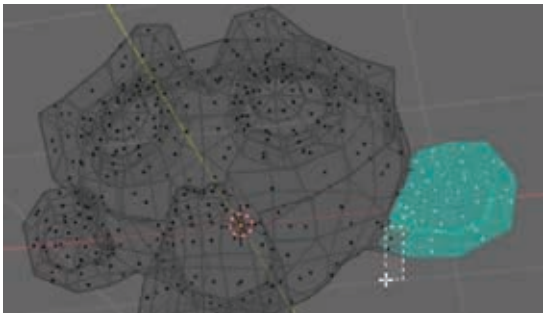




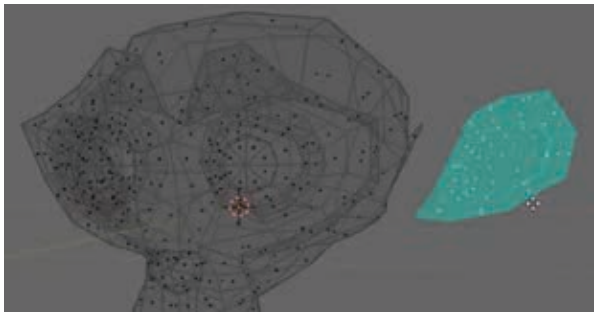
Separate a part of the mesh

In **Edit Mode** select the part you want to separate. Drag while holding the left mouse button to make a box selection.

Pressing W cycles through different selection modes like Lasso select.



Press Y to separate the selected faces.
Press G to move the selection.

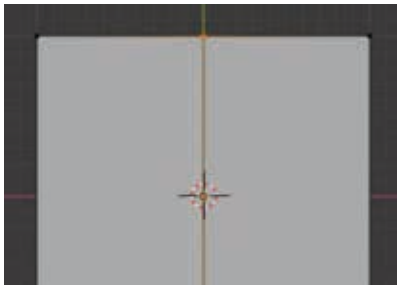


Moving edge loops to center

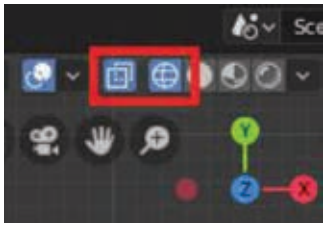
If you want to mirror an object, typically you will add a loop cut in the middle, delete one half and use a mirror modifier to create the other half.

Adding a loop cut in the middle by **pressing Ctrl+R.**

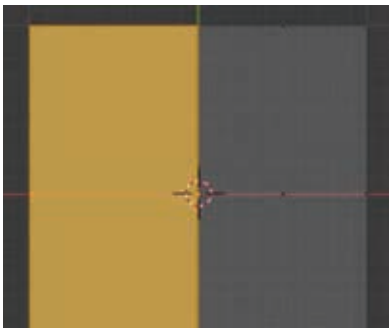
Go to Face selection mode by **pressing 3**.



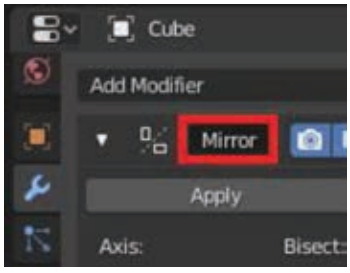
It helps to have **Wireframe** shading and **X-ray** turned on to select the faces on the back as well.



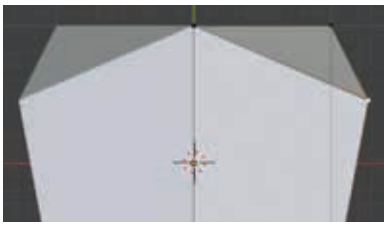
Select one half of the faces and delete them by **pressing Ctrl+X**.



Then you add a **Mirror modifier** in Object Mode to recreate the deleted half of the model.

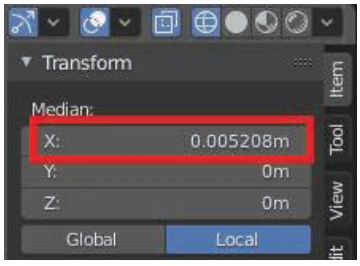


Now, any changes on one side will be mirrored on the other side, which is a huge time-saver on symmetrical objects (like characters).

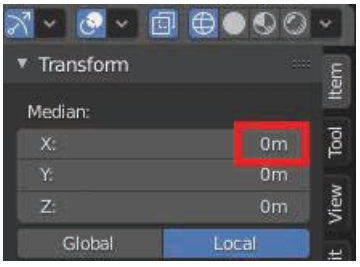


Here you can see that the middle edge loop is just a bit removed from the world center so the mirror modifier doesn't close the gap.

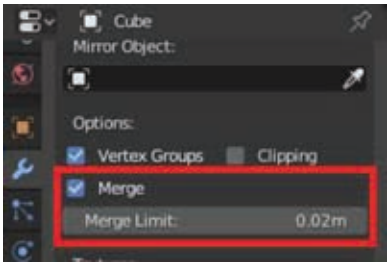
In the option panel (**N**) you will see which axis the edge is moving on.



Setting this axis value back to zero fixes the location issue.



Alternatively increase the **Merge Limit** in the modifier until the edges merge.



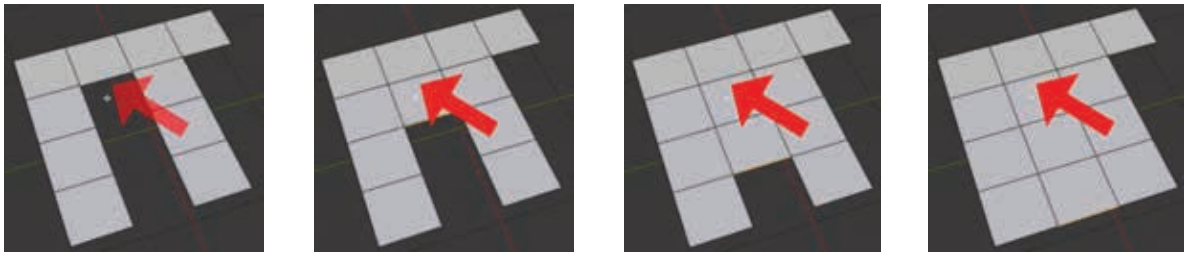
Click here to return to the index.

F2 Add-on

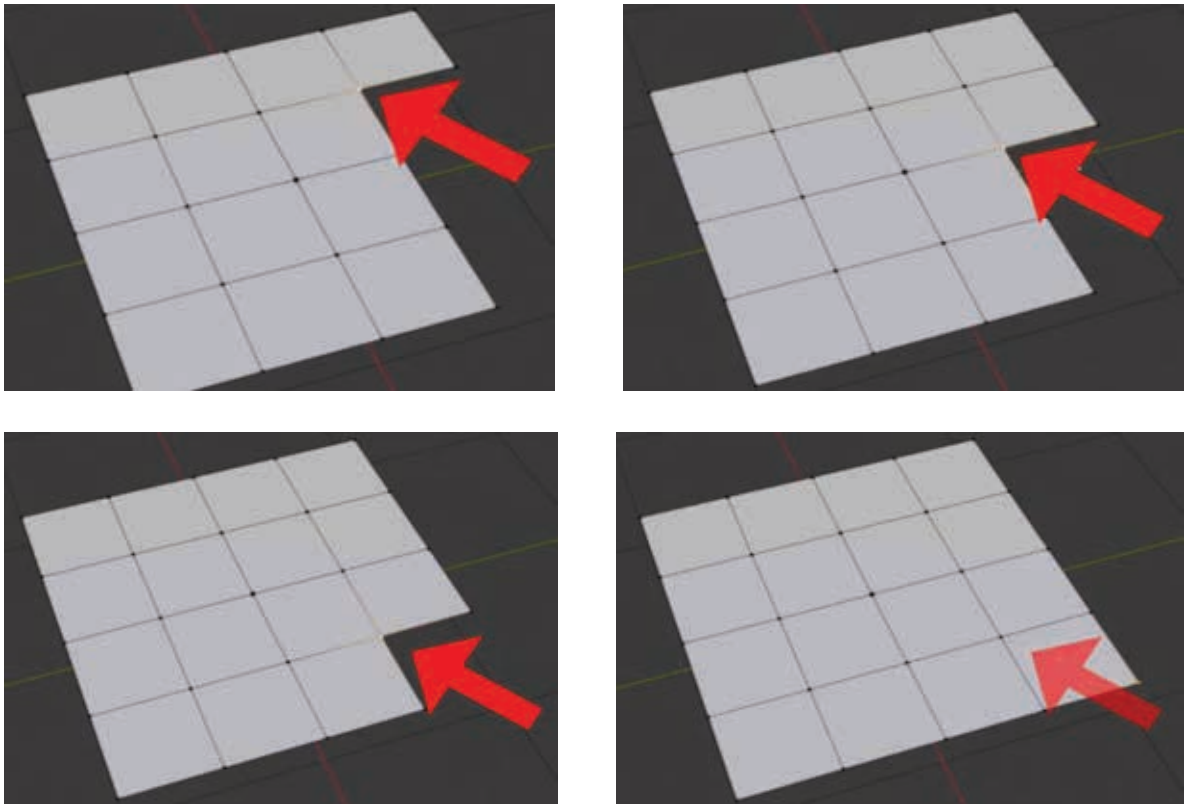
When retopologizing we often have to fill faces. This means a lot of selecting edges and **pressing F**. The **F2 add-on** speeds up this process. The F2 add-on comes with Blender by default.

You just need to activate it in Preferences. To do this, go to **Get Extensions** and search for F2. Then click on **Install**.

Then, to use it select one edge like the one pointed out below, and **press F**, in this case 3 times, to add the missing faces. You only need to select the edge once.



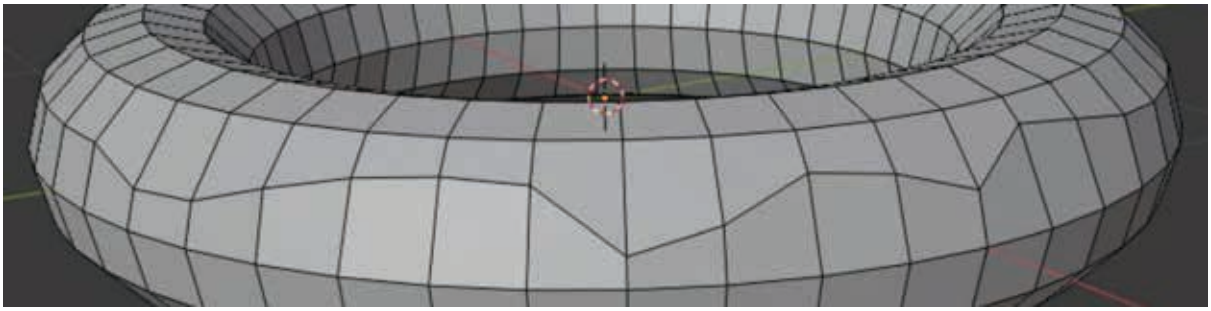
Or select a vertex (like the one pointed at below) and **press F** to add a face. Right-click to confirm.



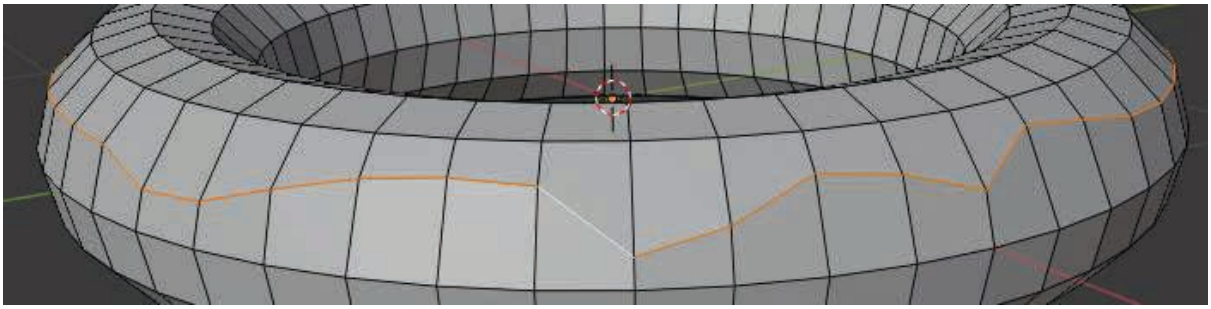
It takes a little while to get used to this add-on, but it's really worth using. Saves a lot of time!

Easily fix Edge Loops

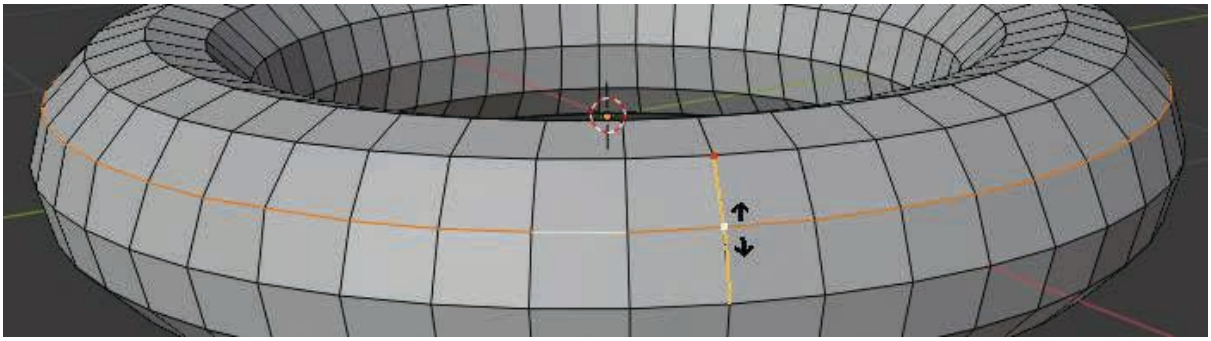
Here is a quick way to fix this messed up edge loop.



Select the edge loop by **holding Alt+ Left Mouse Button clicking** on it.



Move the edge loop by **pressing G twice**. Follow this by **pressing E** to make the line straight.

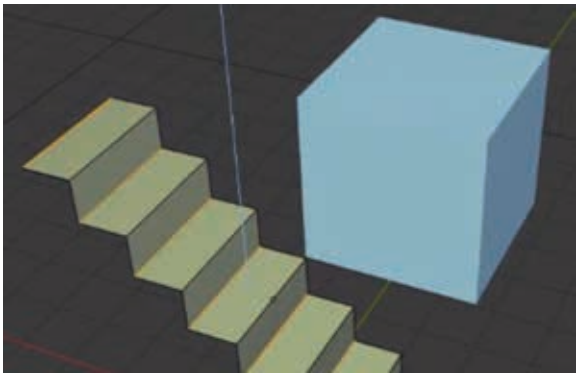


Hold Ctrl to snap the edge incrementally while moving it.
You can also choose which neighbouring edge to follow the orientation of by **pressing F**.

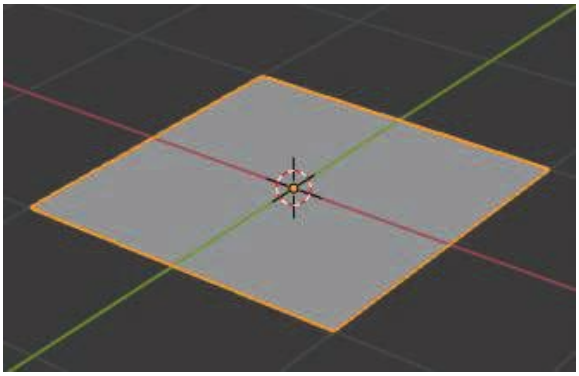
Rip Fill

Rip Fill is a way to "rip" (break the connection between vertices or edges) where a new face is created in between automatically. It's similar to the Rip command (**press V** to rip a vertex into two) except that it also fills the hole that is created.

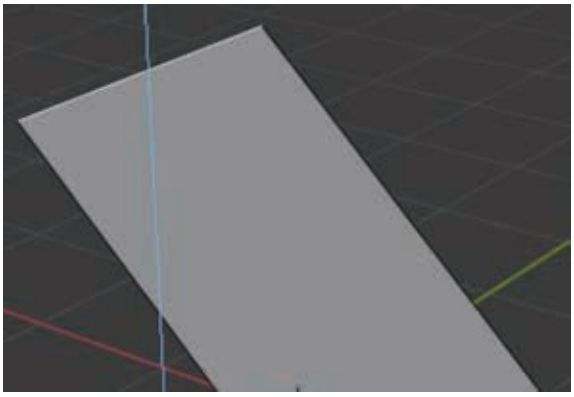
It allows us to quickly make this staircase for example:



First, create a simple plane. **Press TAB** to go into Edit Mode.

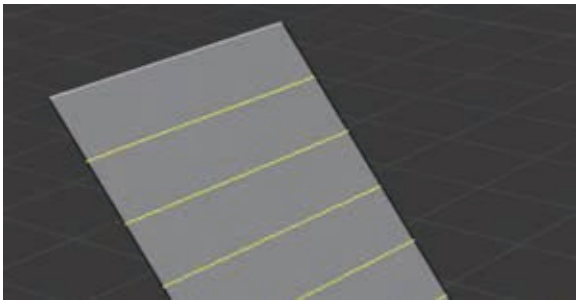


Press 2 for Edge select mode and move one of the edges up in the Z-axis by **pressing G** and Z and moving the mouse.

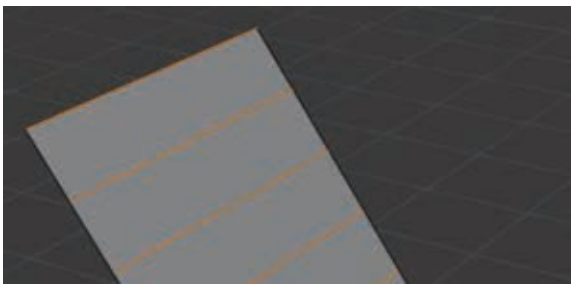


Now **press Ctrl+R** and scroll the mouse-wheel up to create several loop cuts.

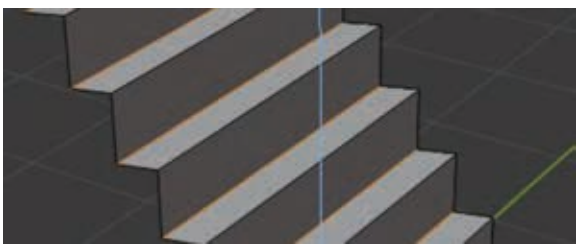
Press left mouse button to confirm and right mouse button to exit the Loop cut tool without moving the new edges.



Select the top and bottom edge as well: **hold Shift and left-click** on them.

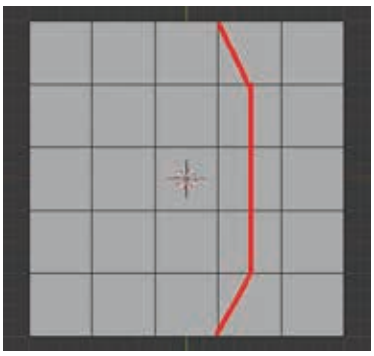


Press Alt+V to Rip Fill and **press Z** to move the new edged down. There you go, you've made some stairs!

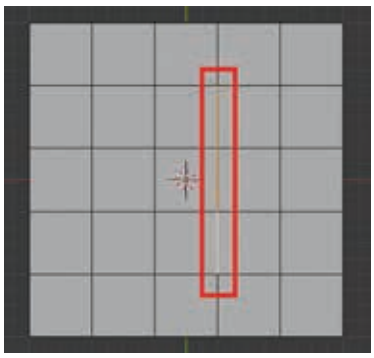


I've noticed that the behaviour of this tool can be unpredictable. Sometimes you need to try a couple of times to get what you want. When would you like to use this, other than for making stairs?

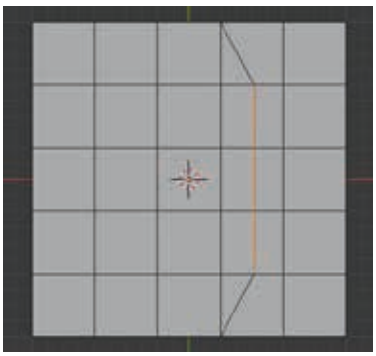
Let's say you want to add these extra cuts:



First select these edges, then **Press Alt+V.**

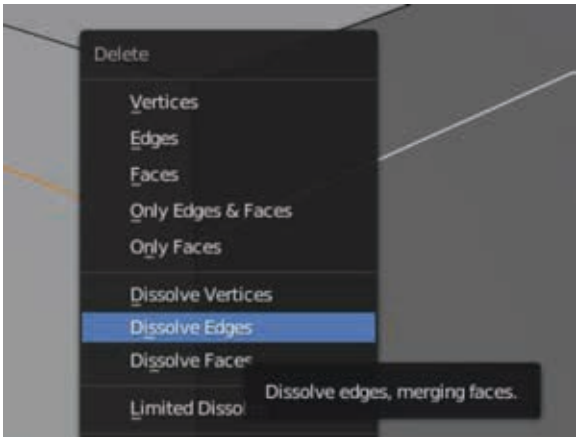


Move the new edges on the X-axis by **pressing X**.

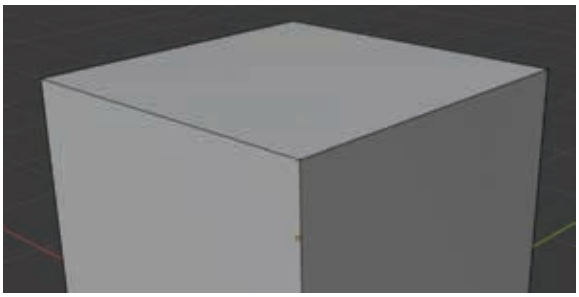


Dissolve Edges

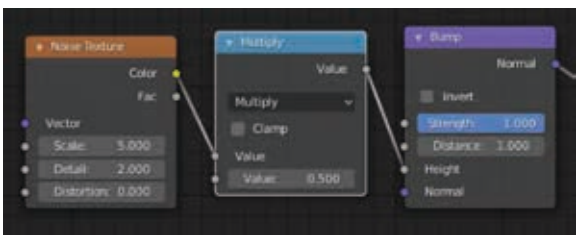
To dissolve an edge, normally you do this: select the edge (by double-clicking on it), then **press X** and choose **Dissolve Edges**.



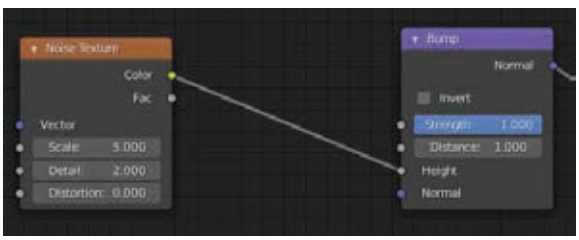
But there is a quick keyboard shortcut for this, so you don't have to use this menu. **Press Ctrl+X** instead of going through the Delete menu. Alternatively you can also press **Ctrl+Delete**. The edge will be dissolved instantly.



This works with nodes as well, for example to remove a node between two other nodes, select it and **press Ctrl+X**.

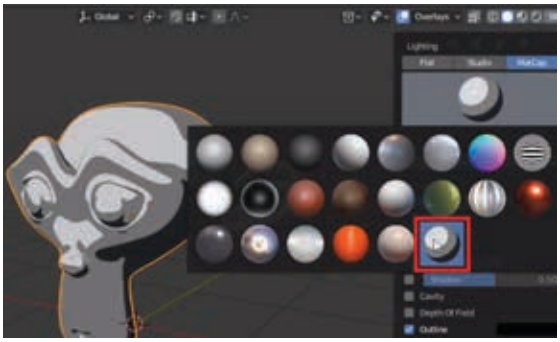


The node will be dissolved and the connection between the other two nodes will remain.

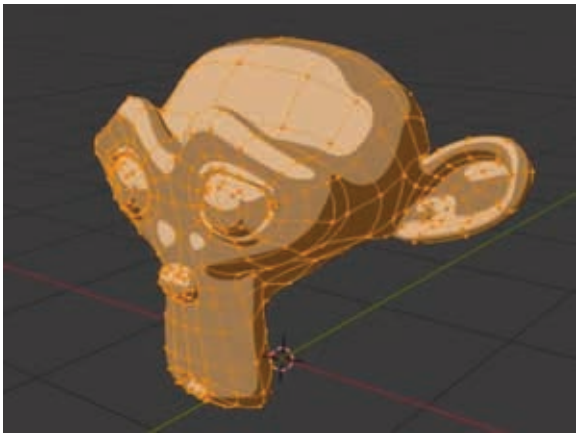


Bisect tool

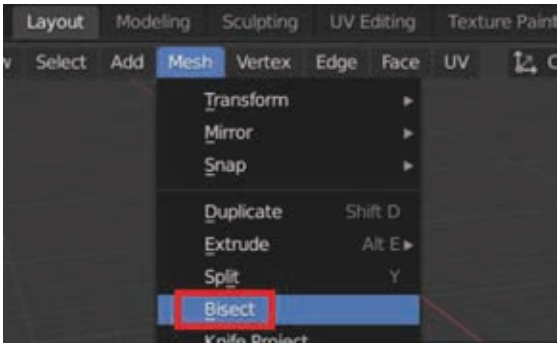
The Bisect tool allows us to slice into a model. For this demonstration I'm using this nice toon shader matcap.



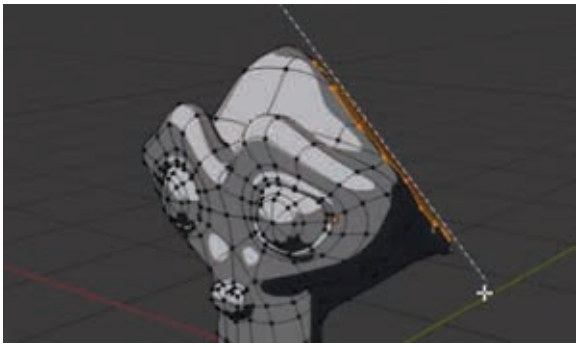
Go to Edit Mode, select all by **pressing A**.



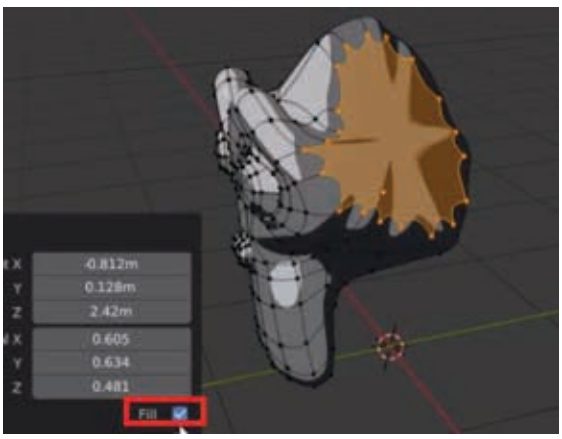
Go to **Mesh > Bisect** or **press F3** and search for Bisect.



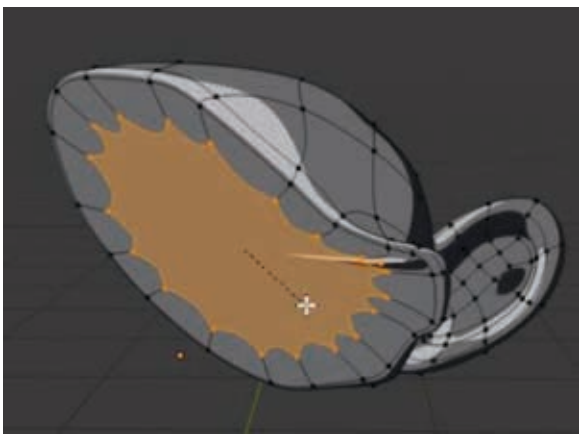
Now you can cut through the model.



The Bisect menu will become available in the viewport. Expand it to see the options.



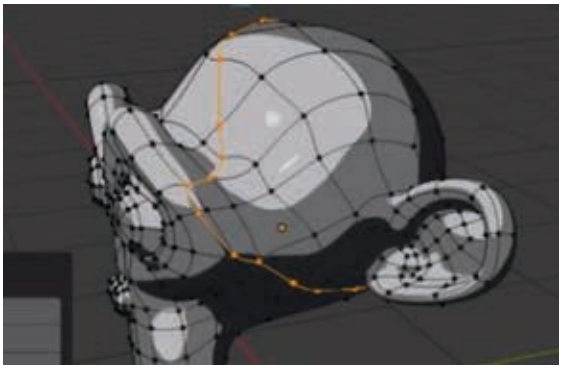
You can choose whether or not to Fill the hole.



Click here to return to the index.

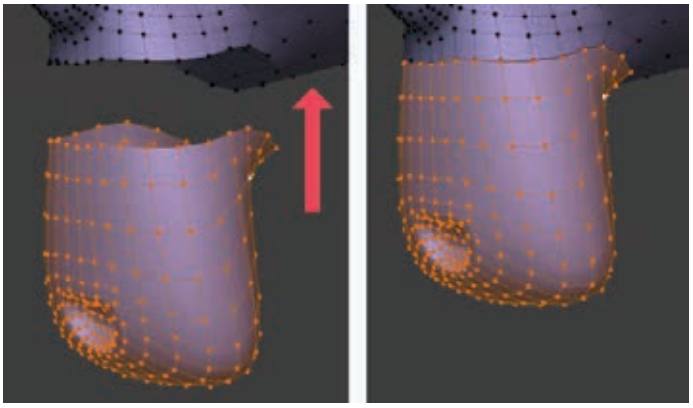
You can also choose whether to remove the inner or outer part after slicing.

If you choose neither Inner or Outer, the Bisect tool will create a cut along the surface of the model without removing any part.

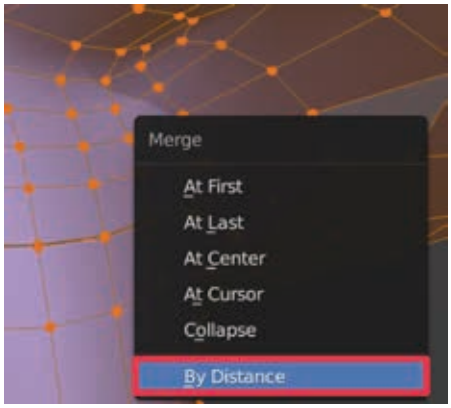


Merge vertices by distance

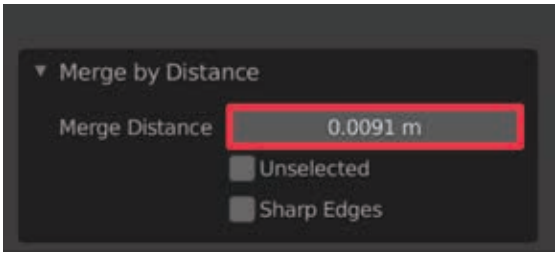
If necessary **press G** to move the faces towards each other. **Hold Shift** for more precision.

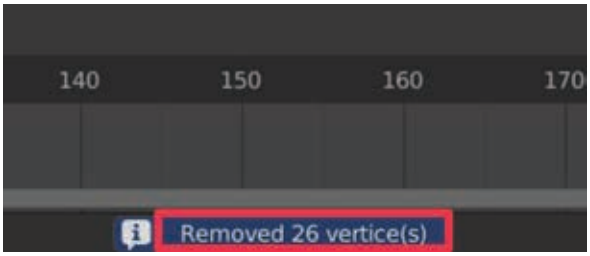


Press M+B or **press M** and choose **Distance**.



Holding Shift, change the **Merge Distance** value until you get an information message in the bottom of the viewport saying that a certain amount of vertices have been merged.

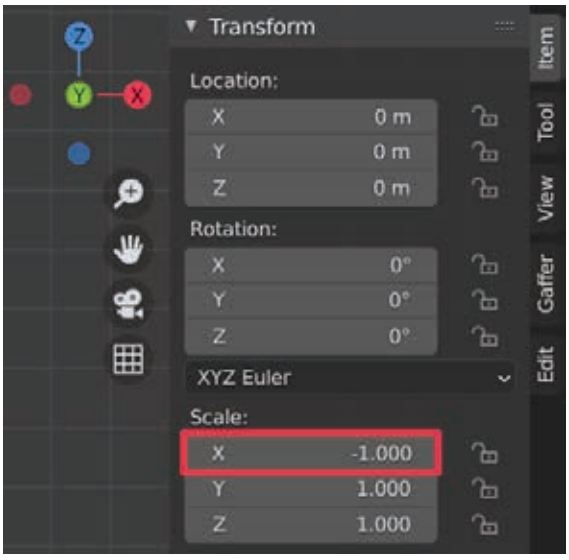




Mirror without a mirror modifier

You can mirror an object by scaling it -1 along an axis.

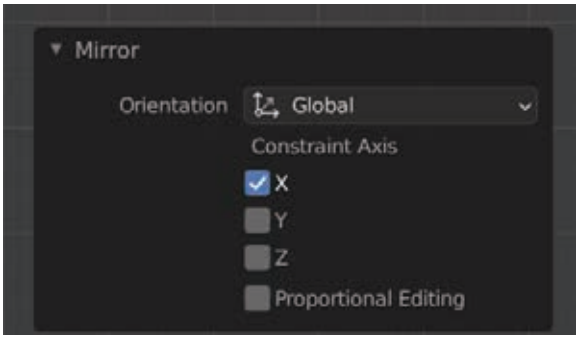
You can see in the Transform panel of the Option menu (which you can open by **pressing N**) that the X scale value is now -1.



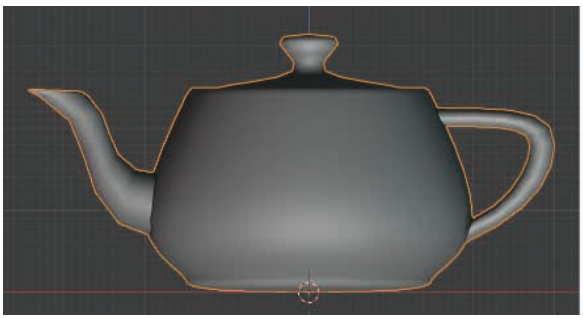
In this example I typed **S, X, -1** in Object Mode to scale it -1 along the X-axis.

You can also **press Ctrl+M** in Object Mode to mirror. For example **press Ctrl+M+X** to mirror along the Y-axis.

Left-click after **pressing Ctrl+M** to get the Mirror menu. There you can choose an axis.



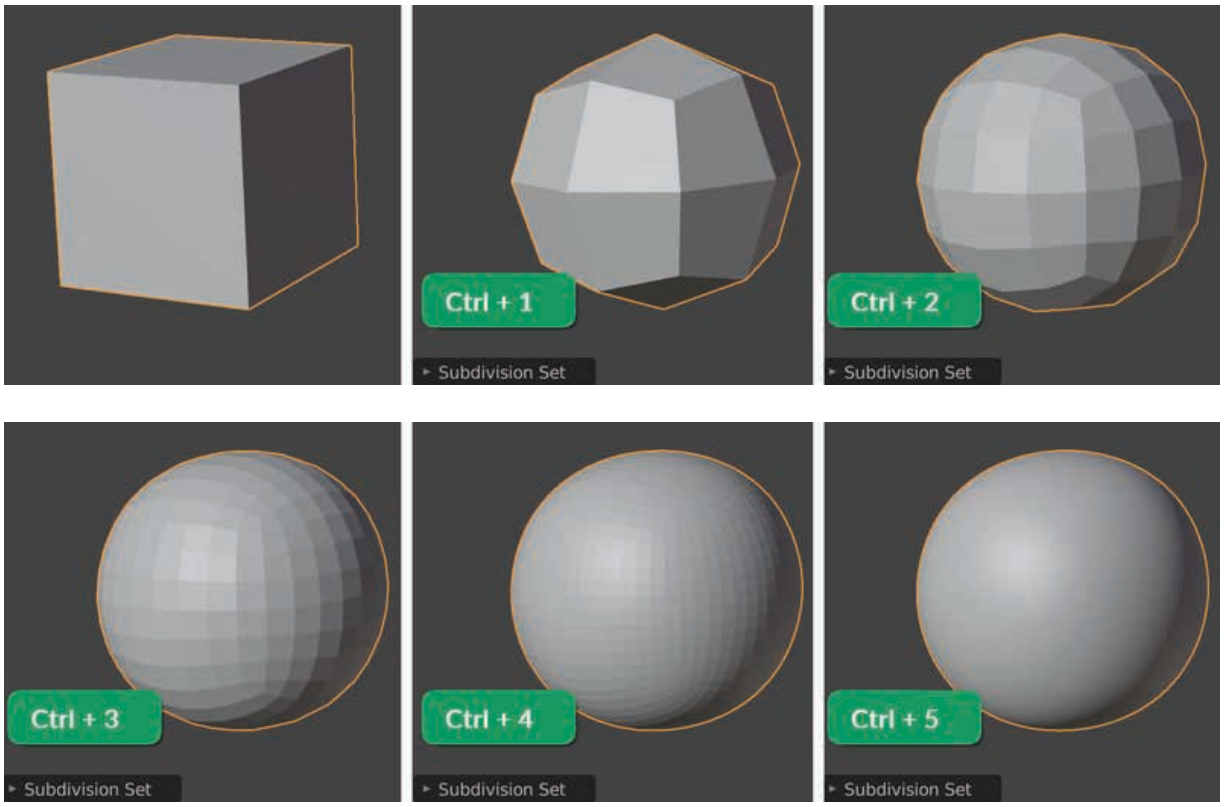
Press Ctrl+A and choose Scale to apply the scale. This unfortunately flips the normals inside out (pictured below left). That's why the object suddenly looks kind of dark. In Edit Mode **press A** to select all, then **press Shift+N** to recalculate normals.



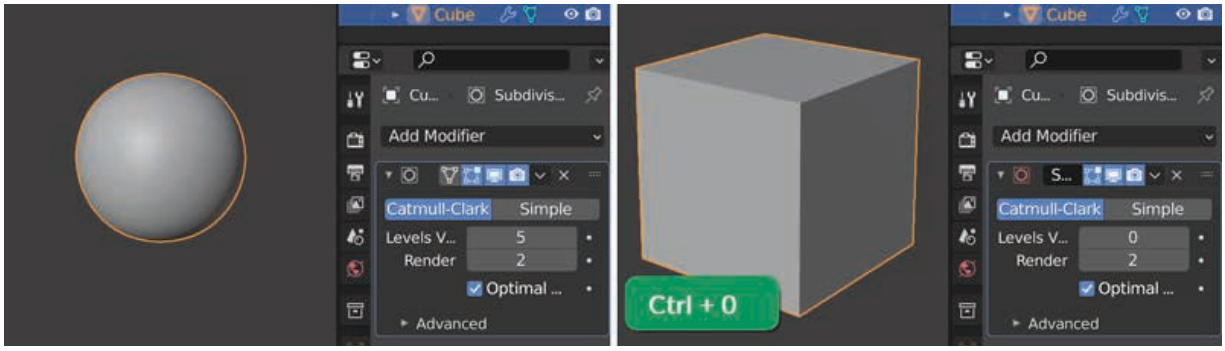
Click here to return to the index.

Smooth Shortcuts and Round Cubes

To quickly add a Subdivision modifier, instead of adding it in the modifier stack **press Ctrl and a number from 1 to 5**.



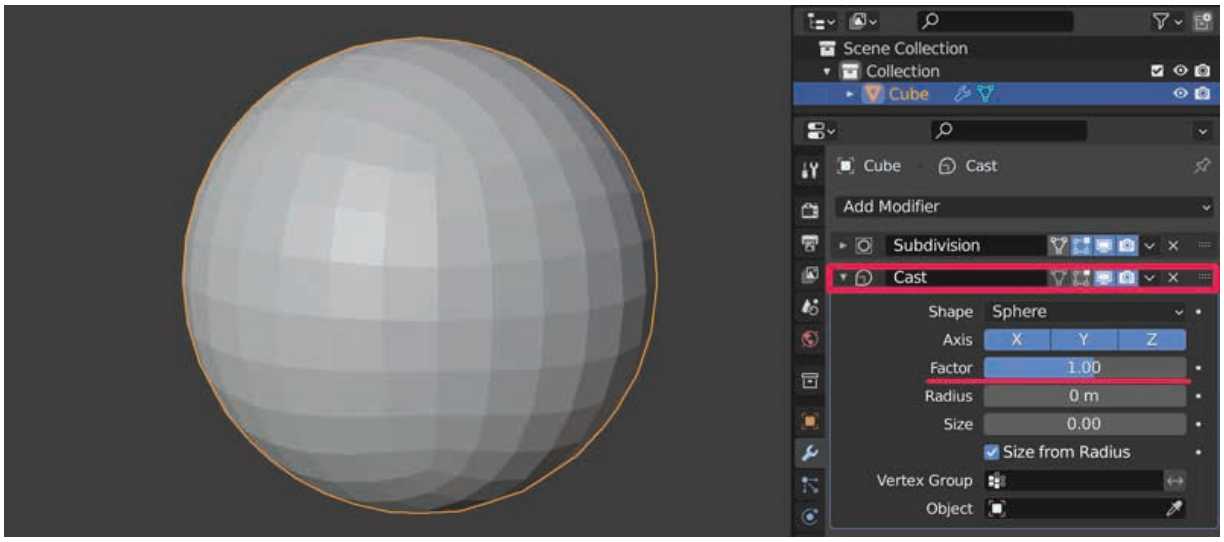
You can then use this keyboard shortcut to toggle between the subdivision levels and **Ctrl+0** turns the modifier off.



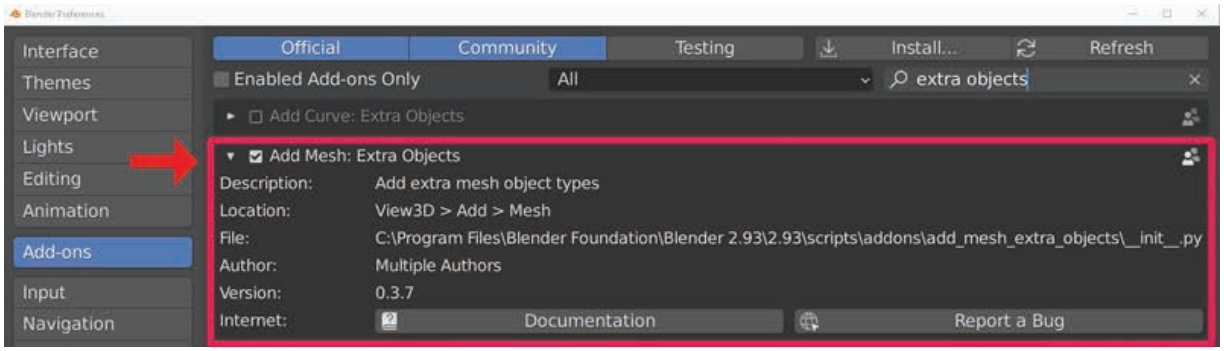
It does not work with the numpad keys, only the top row of numbers from 0 to 5.

Click here to return to the index.

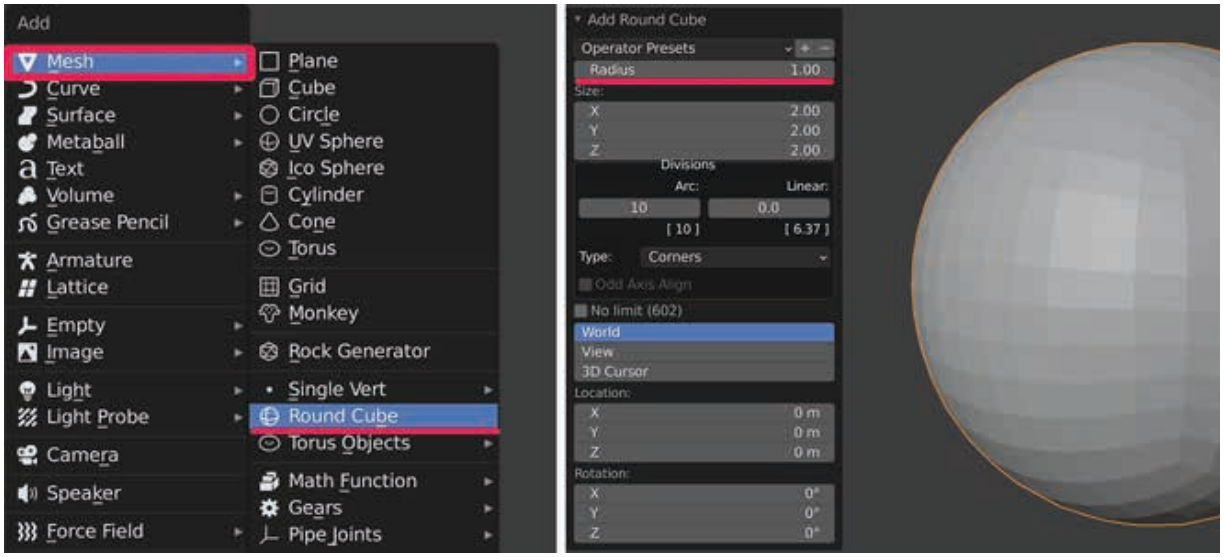
Bonus tip: if your goal is to make a perfectly round sphere a subdivided cube is not the way to go. Unless you add a **Cast modifier** and set to **Sphere,** Factor 1.



Alternatively, you can activate the **Extra Objects add-on** that comes with Blender by default.

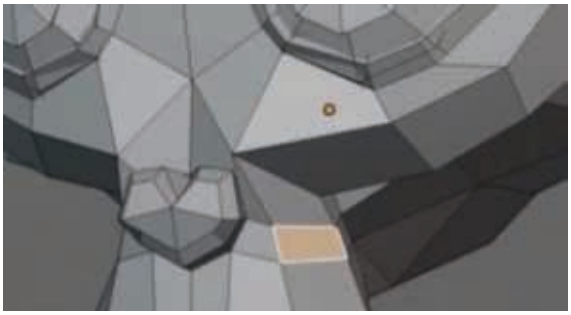


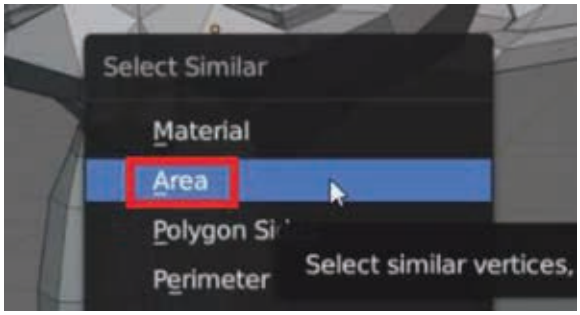
It adds the **Round Cube** to the primitives.



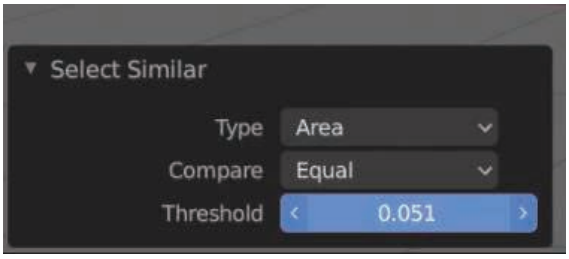
Selecting faces by size

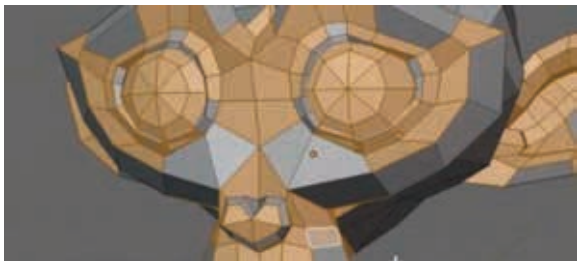
To select faces with a similar surface area, select the face and **press Shift+G** then choose **Area**.





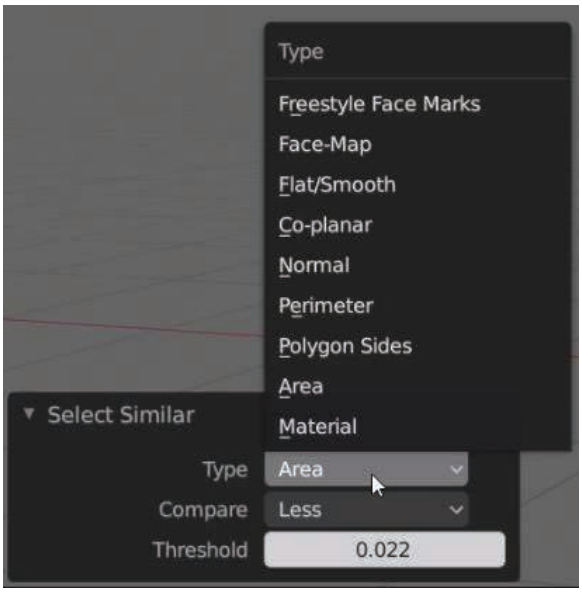
In the corner you will get a menu with a **Threshold** value to increase or decrease.





You can also select faces with greater or lesser surface area than the one selected.

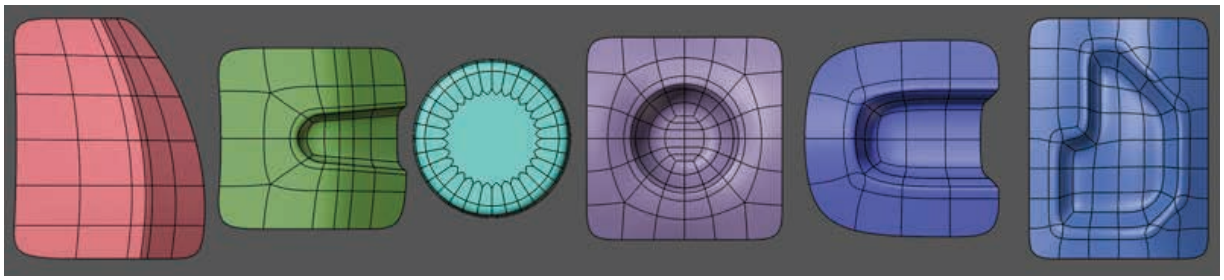
As well as other selection criteria instead of the surface area.



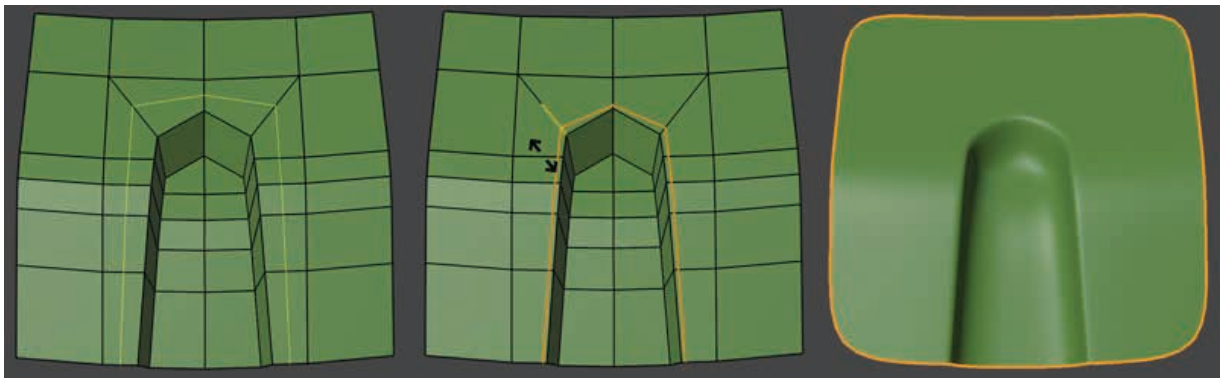
Click here to return to the index.

Boundary Protection - 6 ways to add support loops

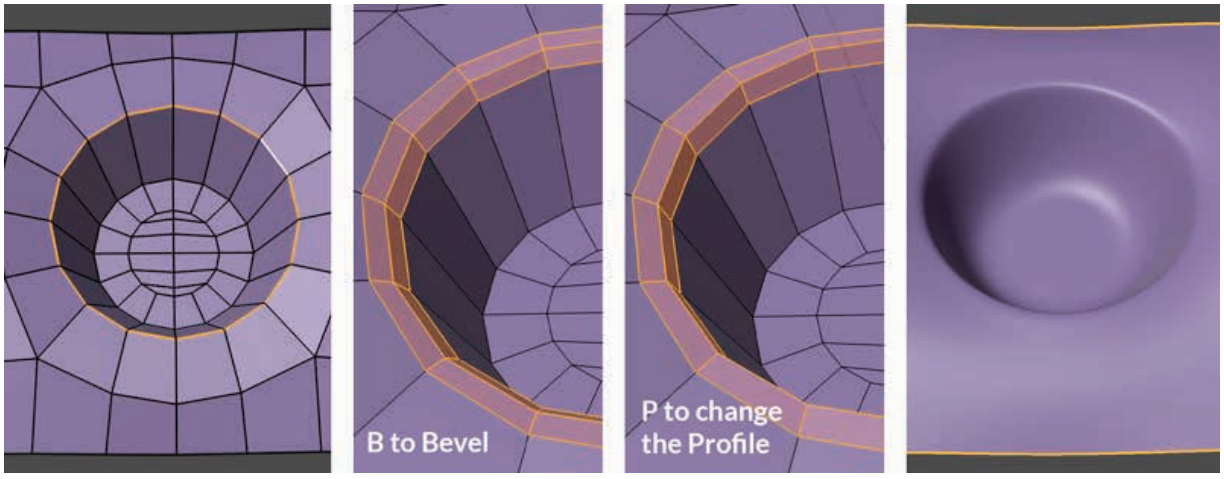
In a **Subdivision modeling workflow** where you use a Subdivision modifier to add smoothness to your model, you'll want to add boundary support wherever you want sharp edges. If you're not using Creasing, that means adding actual edge loops for support. There's several methods for adding these loops.



Loop Cut and Slide
By **pressing Ctrl+R** you can add a Loop Cut. Then you can move the Loop Cut closer to the boundary, to get a sharper corner. You can **press E** to conform the loop to the neighbouring geometry, and **F to flip** which side it conforms to.

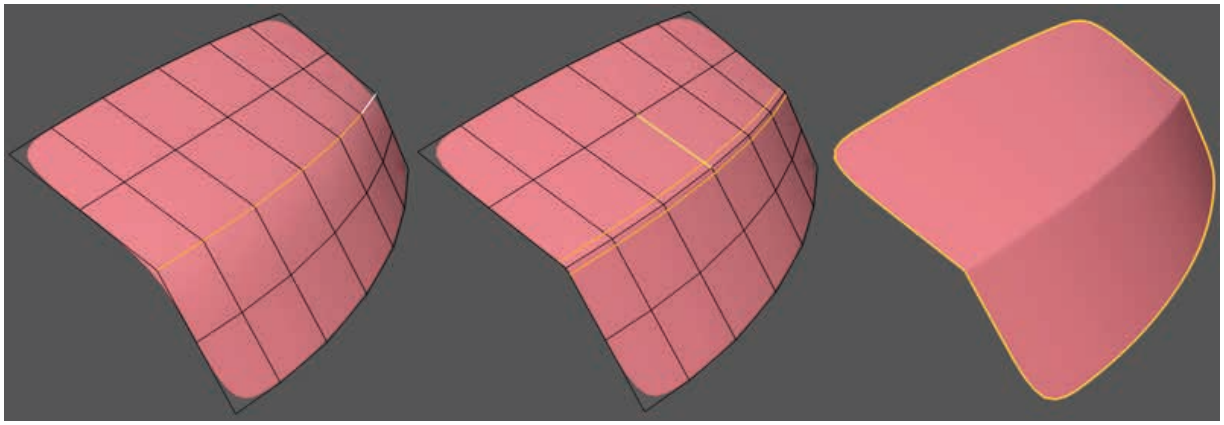


Bevel
Select the boundary edges, **press Ctrl+B** to Bevel, scroll the mouse wheel until you have 3 loops. **Press P** (for Profile) then drag the mouse to change the profile to a 90° angle.

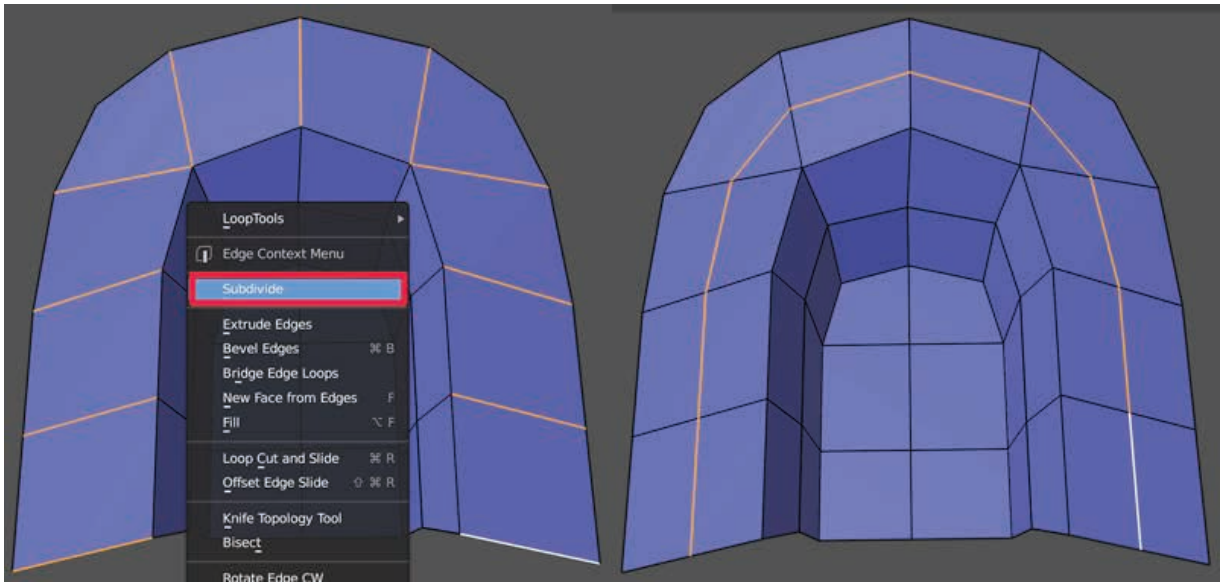


Click here to return to the index.

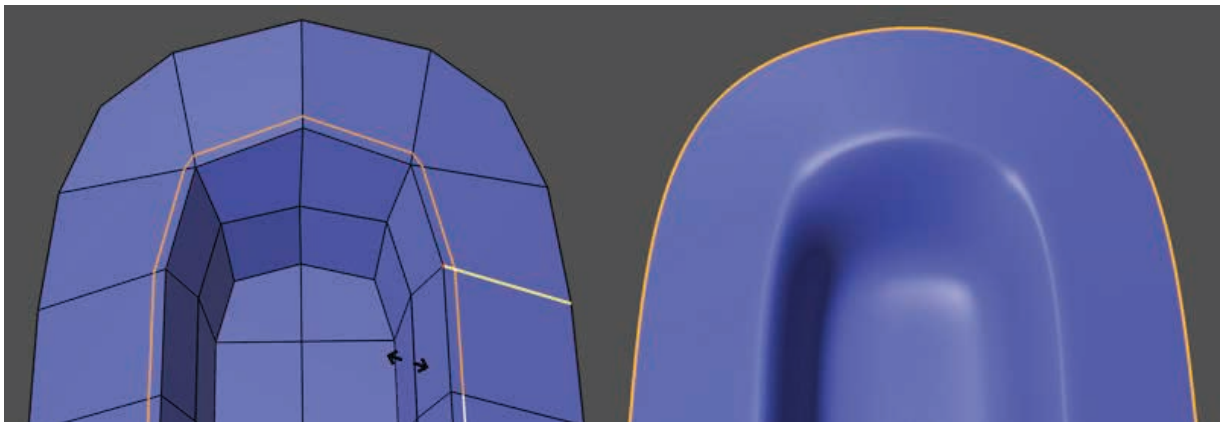
Offset Edge Loop
To add Loop Cuts on on both sides of a selected boundary at once, **press Shift+Ctrl+R**.



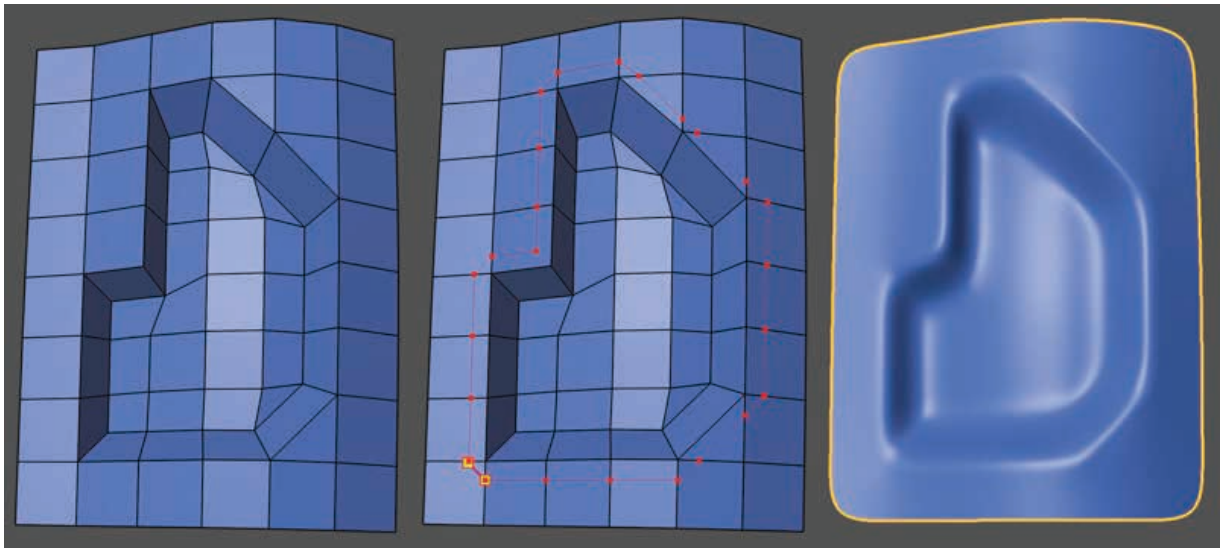
Subdivide
You don't always have a boundary that you can easily select, or you just want to add a very specific edge flow. One solution in such a case is to select the edges where you want to force the loop, **right-click** and choose **Subdivide**.



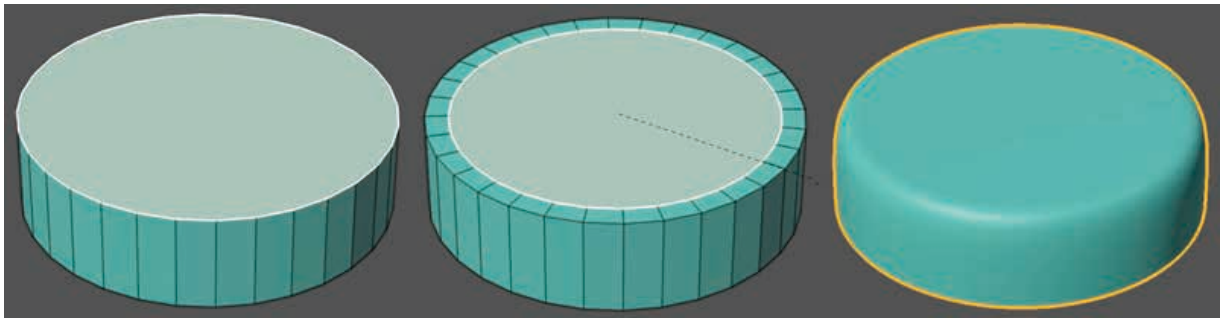
Then you can select the newly created edge loop, slide it by **pressing G twice** and and **press E and F** to conform it to the neighbouring edges.



Knife Tool
Similarly, you can create any kind of loop cut manually by using the Knife Tool, activated by **pressing K**. Confirm the cuts by **pressing Enter**. With this method, you have complete control over the creation of the cuts, but you usually will have to clean the result up a bit to retain good edge flow.

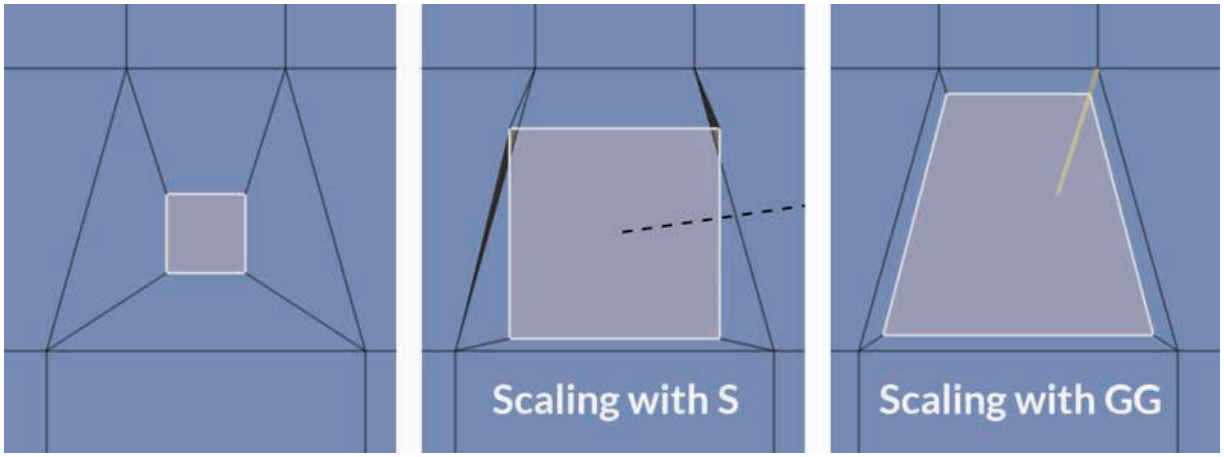


Inset
In some cases, for example a panel or the cap of a cylinder, you can add boundary protection by insetting the Faces that make up one side.

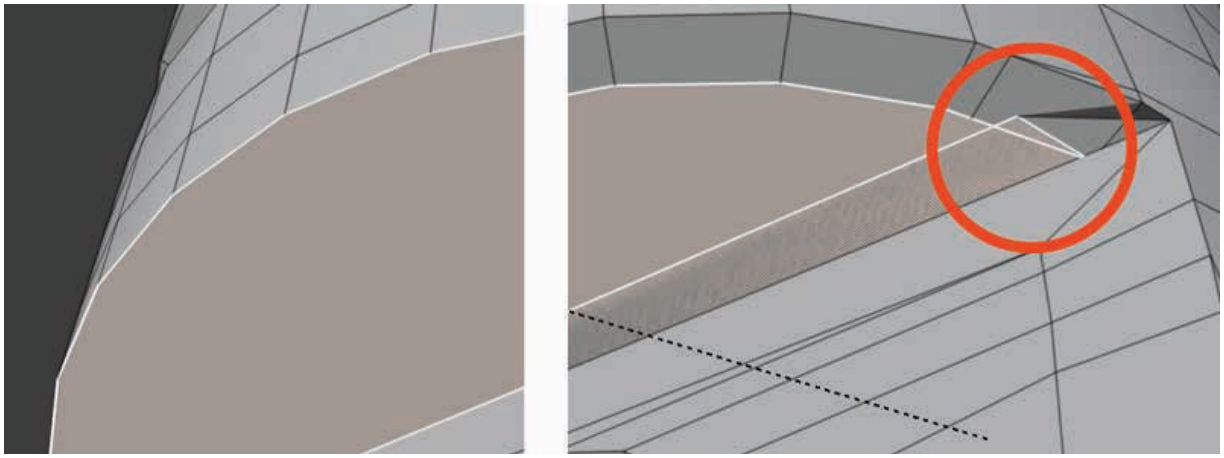


3 Scaling Tips you may not know!

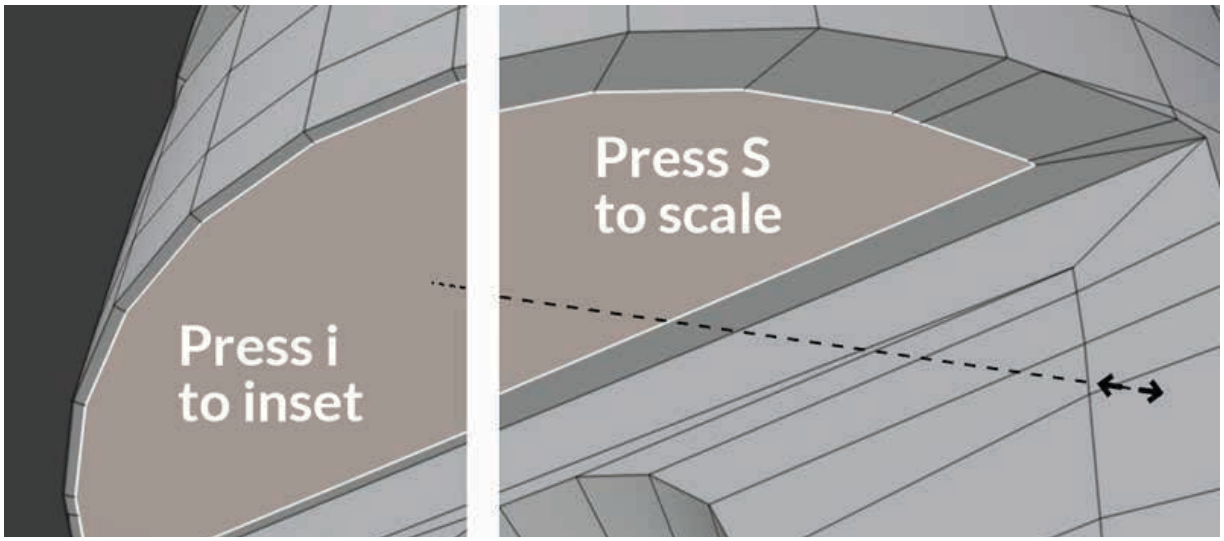
In case you have an inset Face that you want to scale up in a way that conforms the edges around it, **pressing S** may not work unless the edges around it have the same shape as the Face. **Pressing G twice** conforms the inset Face to the surrounding edges as you scale it up.



If you want to inset a Face, but **pressing Ctrl+i** creates overlapping geometry...

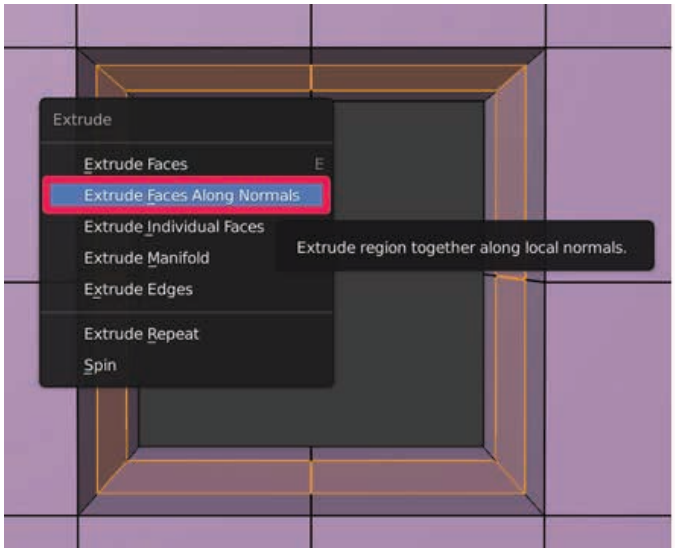


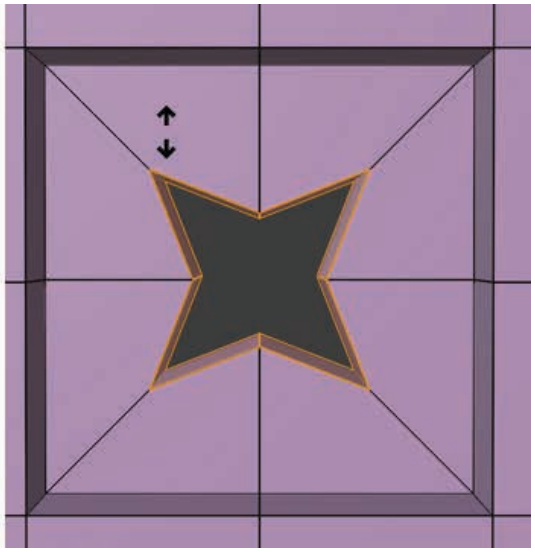
Inset only a tiny bit, **left-click and then press S** to scale it further.



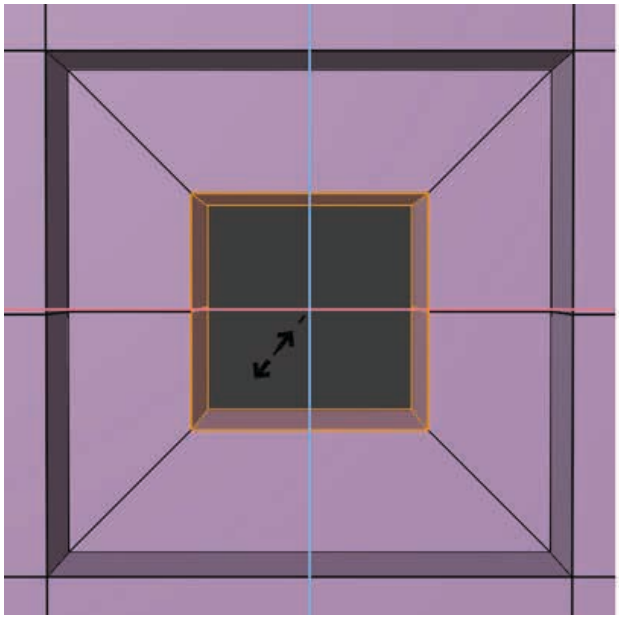
Click here to return to the index.

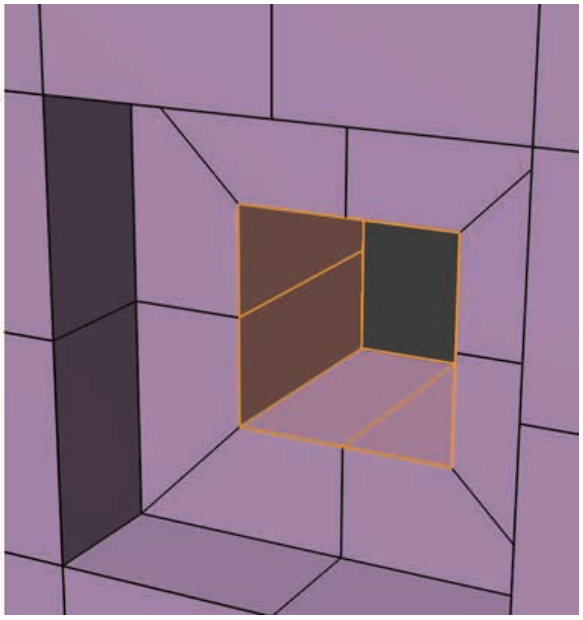
When extruding these Faces, we can **press Alt+E** for the extrude menu and choose **Extrude Faces Along Normals**. However scaling down the extrusion doesn't give a desirable result in this case.





Instead, after Extruding we **left-click** to confirm without scaling the extruded Faces. Then, **press S** and then while scaling, **press Shift** and the axis you want to exclude from the scaling operation, in this case Y. In other words, **press S and then Shift+Y** in this case.

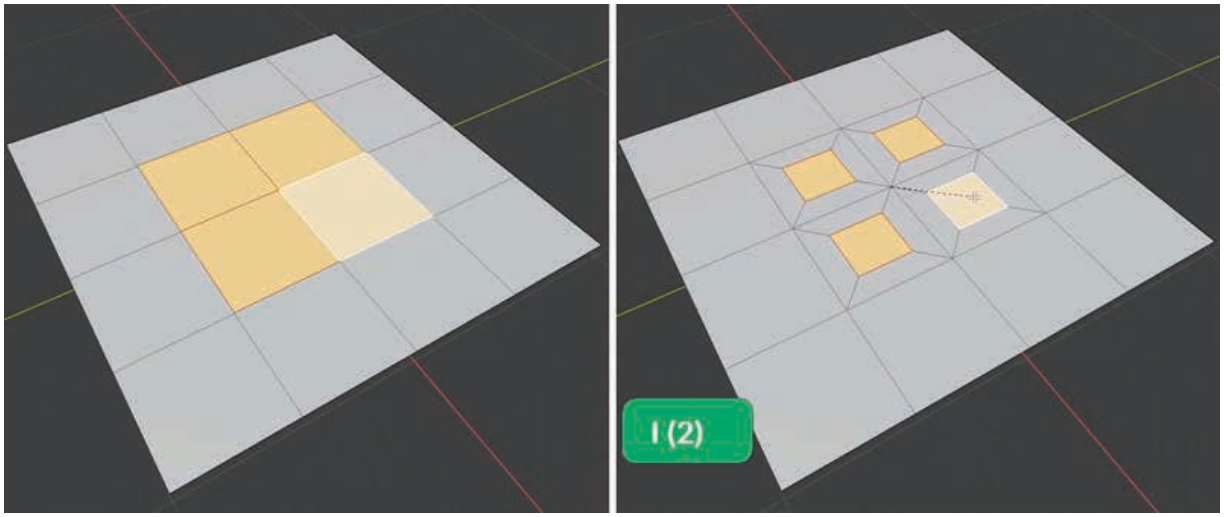




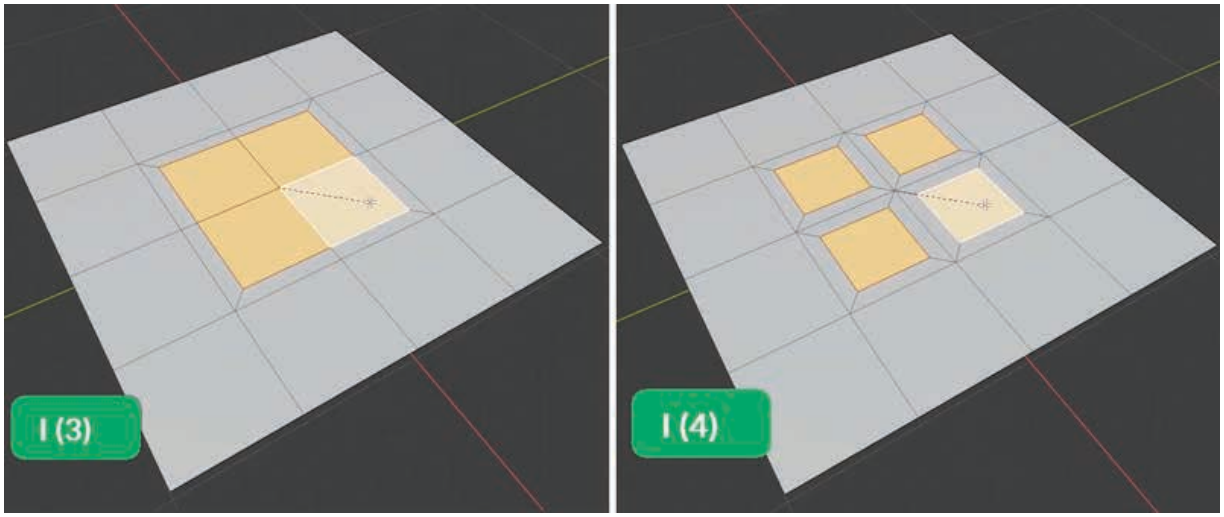
Now it's extruded inwards perfectly!

3 inset modeling tips

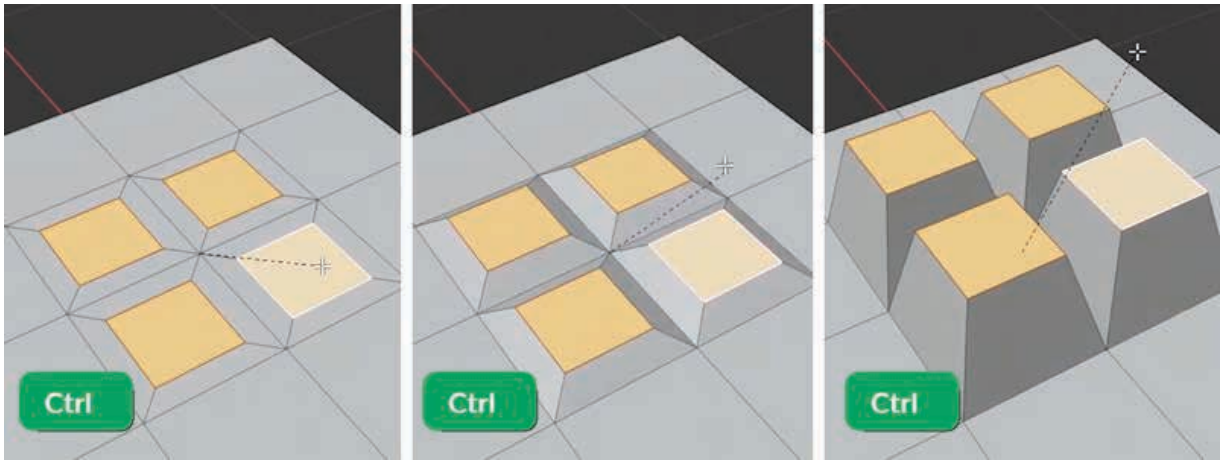
To inset individual faces, **press i twice**.



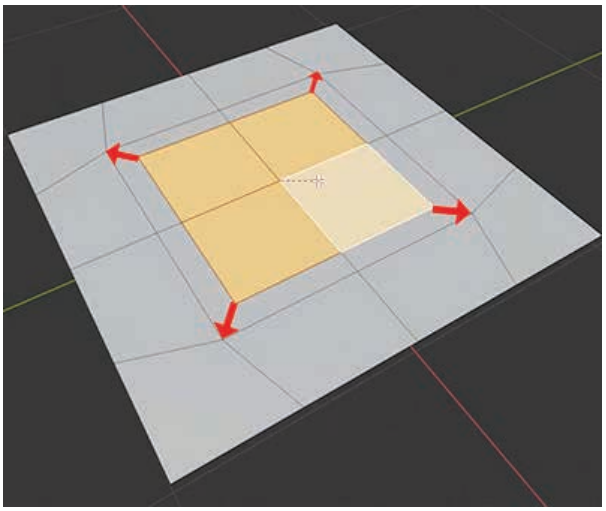
Toggle this behaviour by **pressing i**.



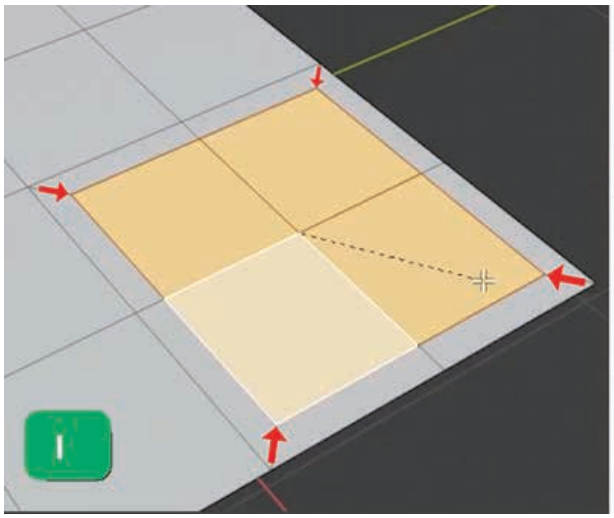
You can even turn it into an **extrude** action by **holding Ctrl**.

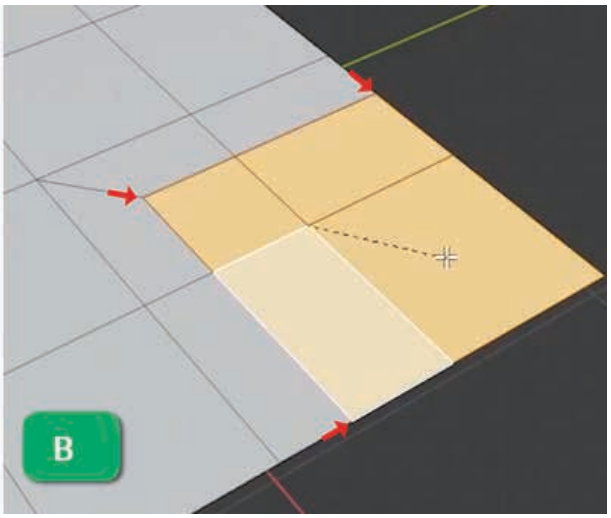


Press o if you want the extra edges to be created outside of your selection instead inside, this is called an “outset”.

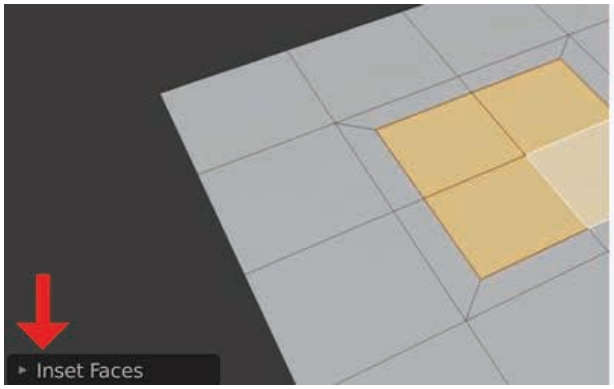


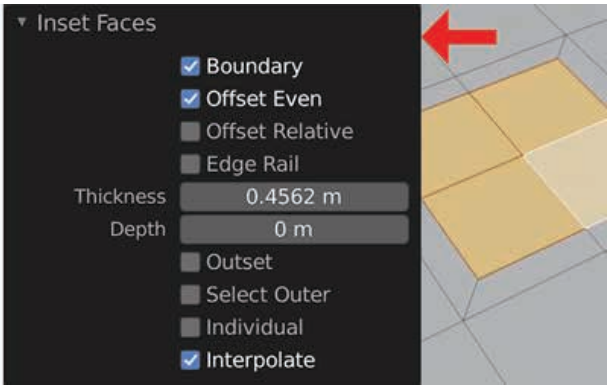
Pressing B toggles whether the **boundary** is also inset or not.





You can also use the Inset menu for these options and more. This should appear in the lower left corner of the viewport. If you accidentally lose it, **press F9** to get it back.

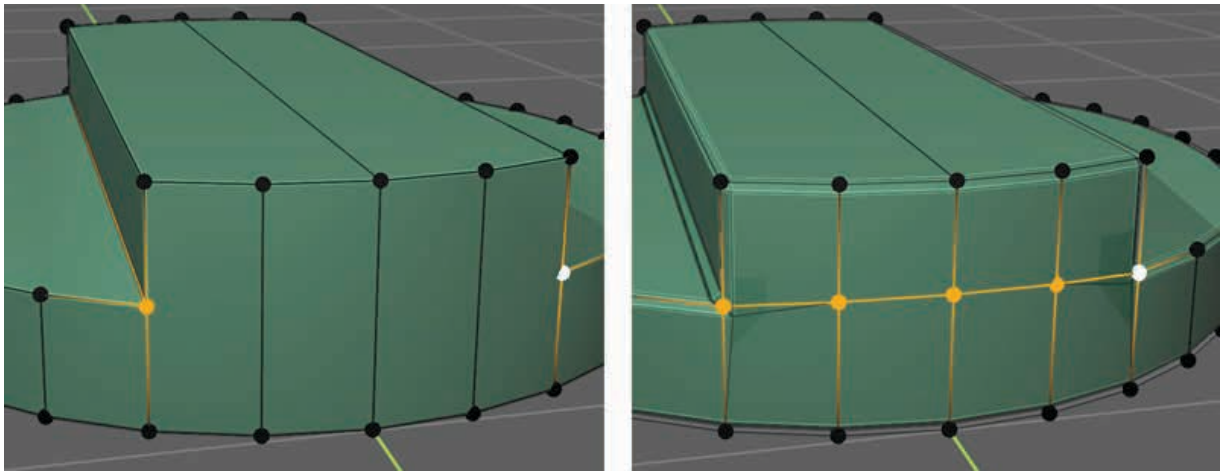




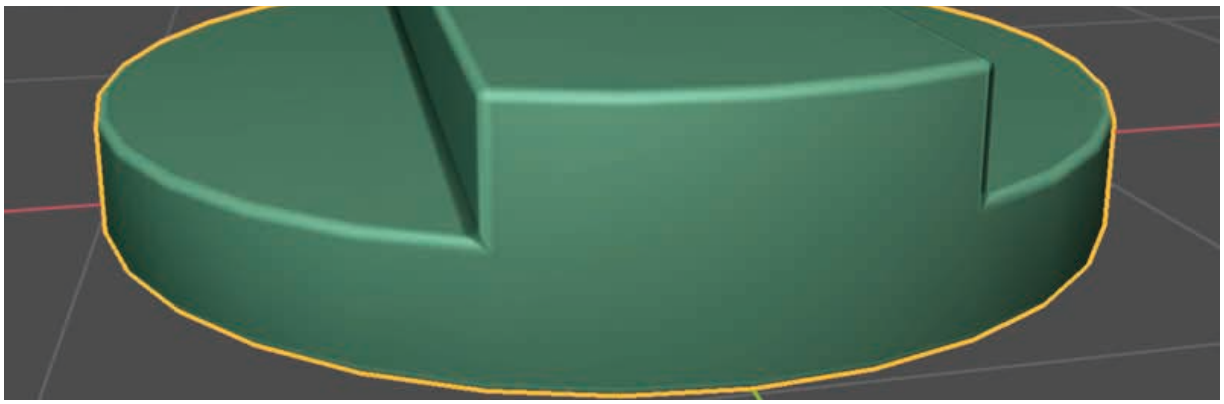
5 Reasons your Bevel may not be working

It can be frustrating when you try to add a Bevel modifier, but nothing seems to happen. Let's look at some cases where you might have Bevel issues, and how to solve them!

Ngons
The below cylindrical object has some Faces which have more than 4 vertices. This is stopping the Bevel modifier from doing its thing. Luckily, by selecting the two vertices at opposing ends and pressing J we can connect them, adding an Edge in between and thus easily getting a Quad Faces mesh.

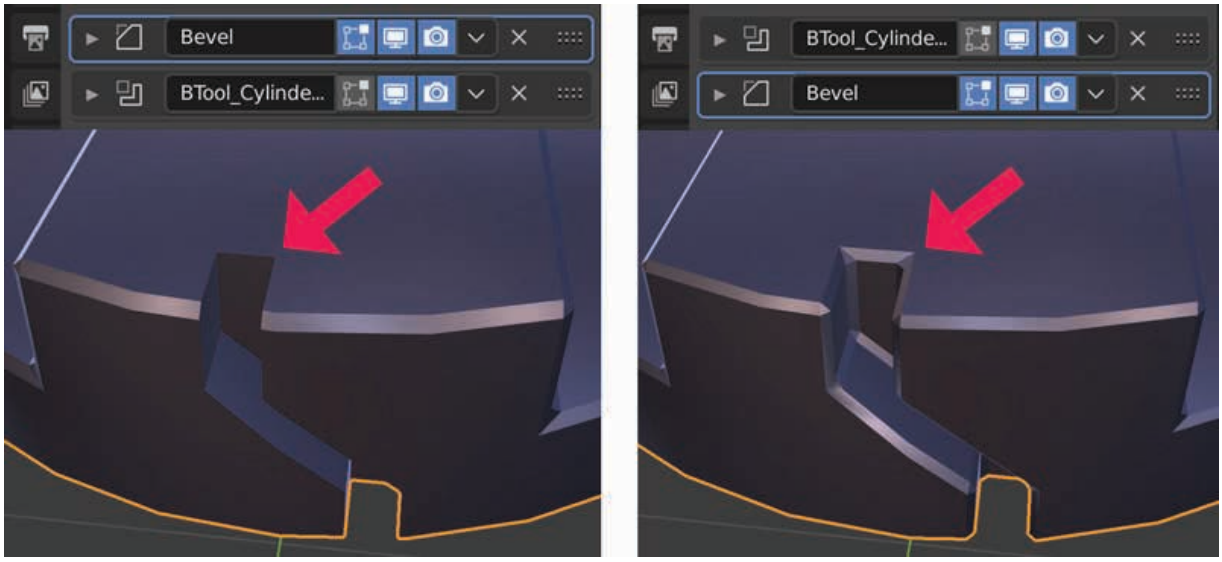


This results in the Bevel modifier working as intended:



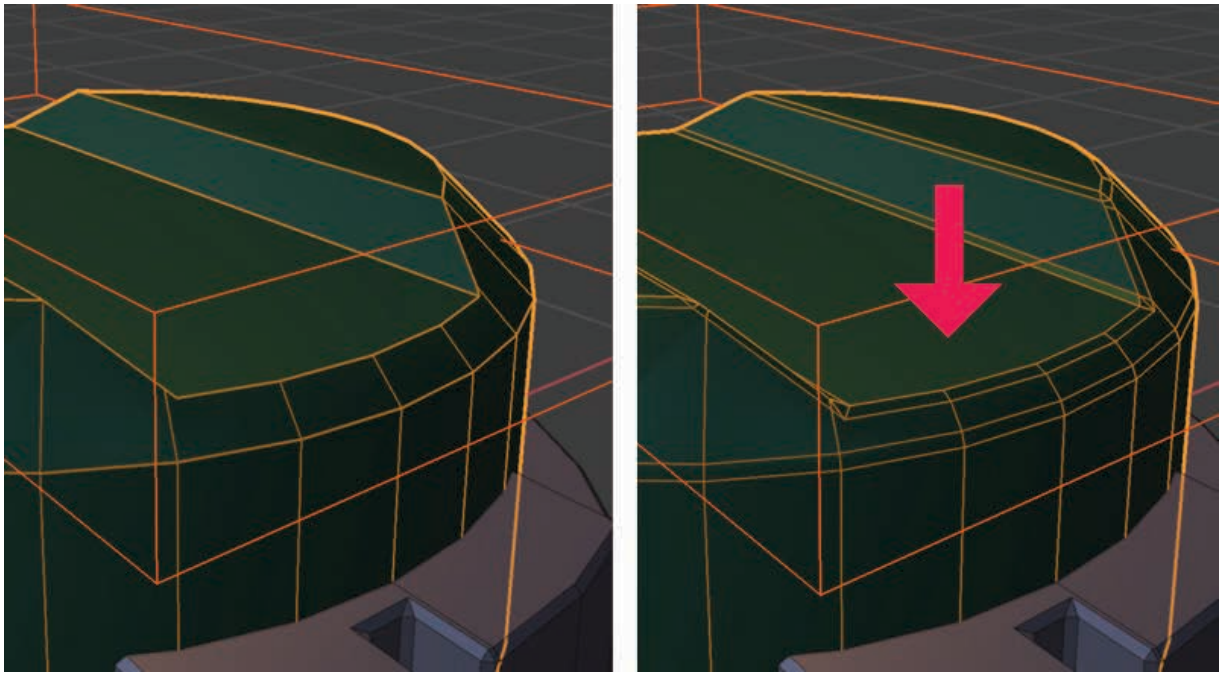
Wrong modifier order
In case you're using a Boolean modifier to cut a shape out of your model, it's important that the Bevel modifier comes before the Boolean modifier in the modifier stack (which actually means that it's visually below it in the modifier stack, because the modifiers are read from bottom to top and not - as you might expect - from top to bottom).

Below you see the difference the modifier order makes. Neither is "wrong" or "right", it just depends on what you want to achieve (Note that the boolean cutter object is hidden).



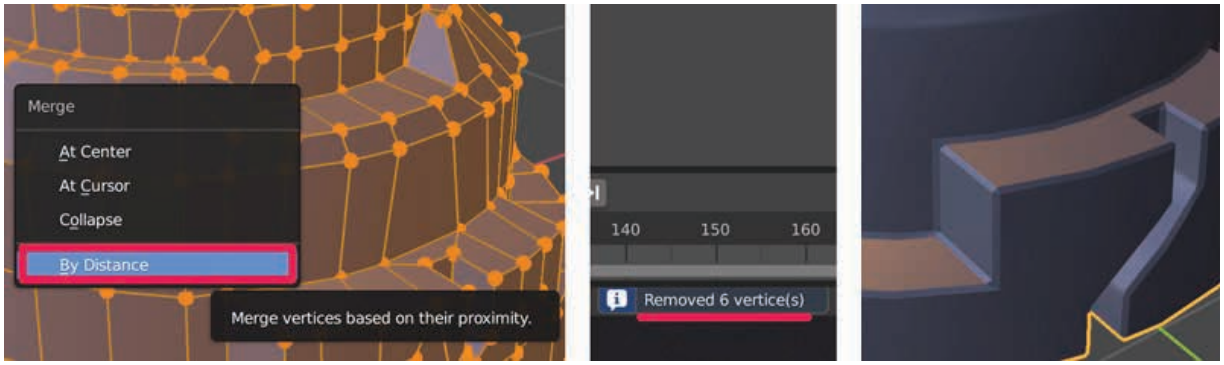
Boolean cutter placement
Sometimes, a boolean cutter object can be placed in such a way, that it creates some vertices or edges which are very close together, closer than the Bevel Amount value. That makes it impossible for the Bevel to do a nice job.

Just moving the cutter object slightly can easily solve the problem in such a case. In this example I moved it down slightly.



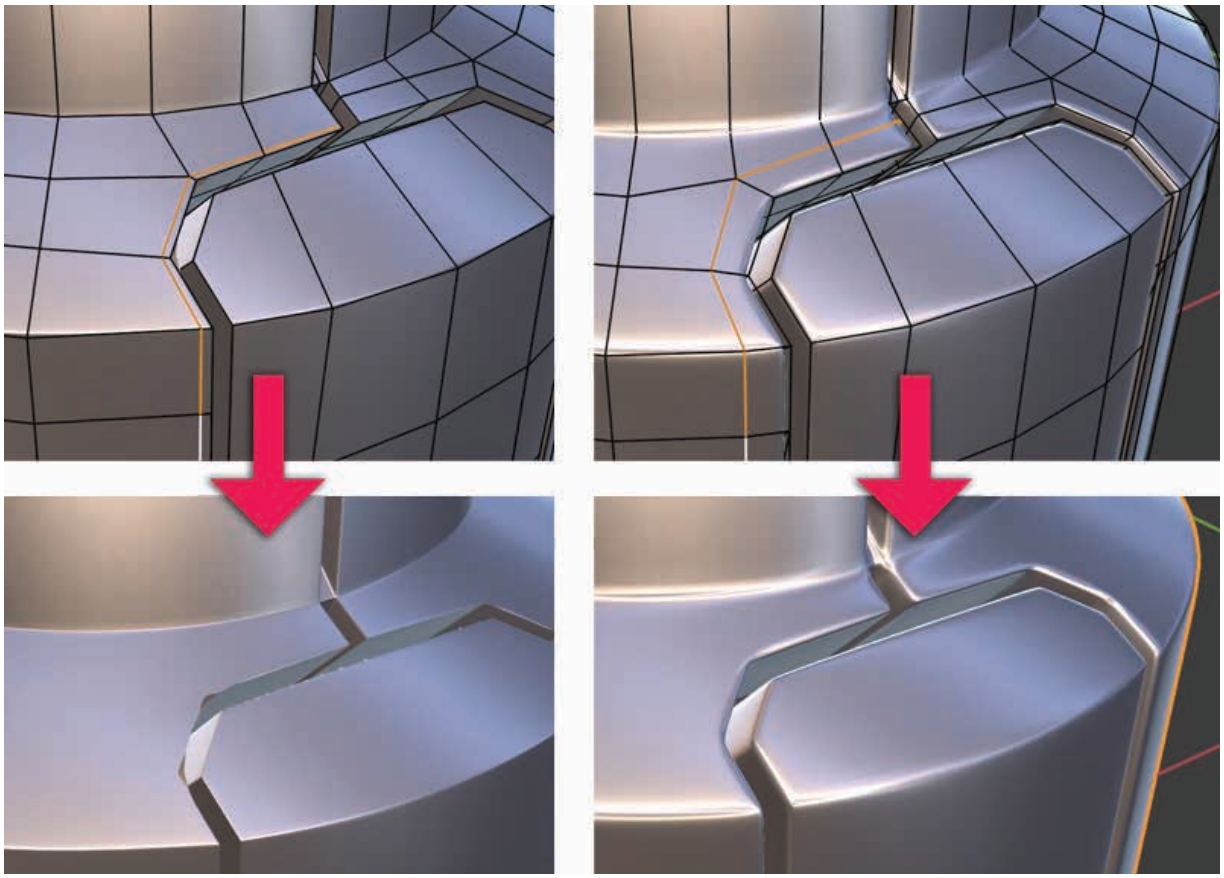
Double vertices
Another thing that might be stopping your bevel from working is overlapping or double vertices. This can be solved by selecting all vertices in Edit Mode, pressing M and choosing Distance. Then, adjust the slider until you see a message at the bottom notifying you of merged double vertices.

Be careful though, slide the value to far and you start merging geometry that you're not supposed to merge.



Edges too close to each other
Sometimes, geometry on your mesh is just too close together to allow for a bevel. If the distance between edges (for example with edge loops that are meant to support a perimeter) is smaller than the bevel size, your bevel won't work as intended.

The solution in that case is to make some space. **Alt+click** on the edge loop to select it, then **press G twice** to slide it.

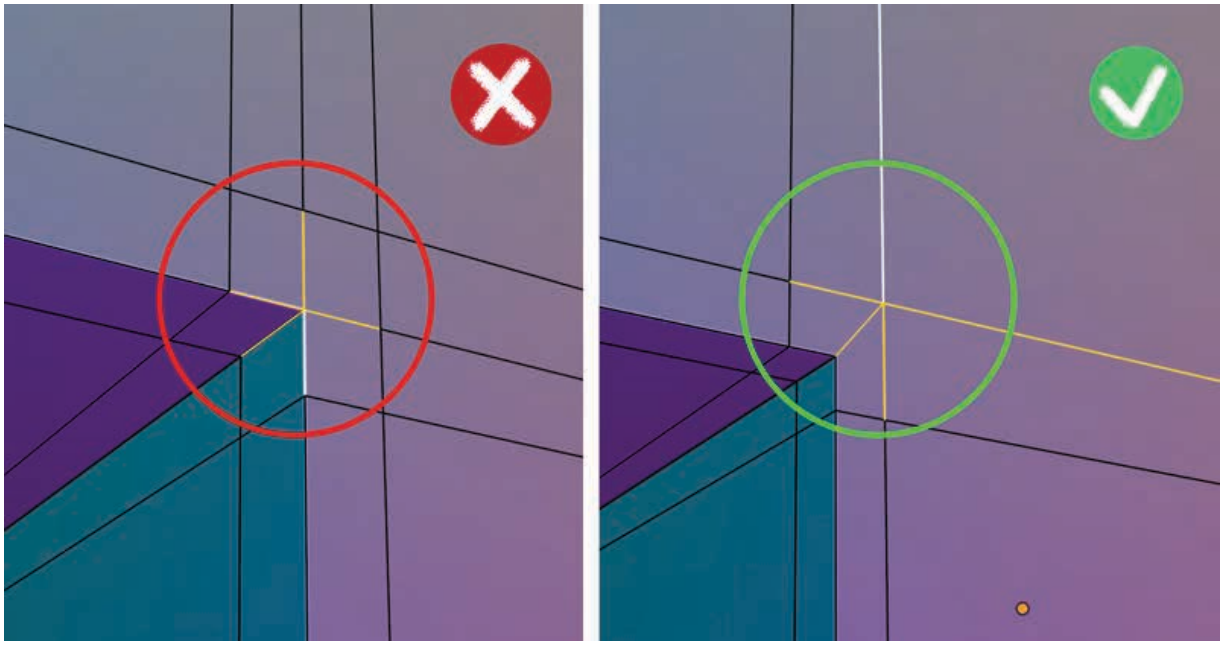


Bevel Tips (Poles, Miter Outer, Shape)

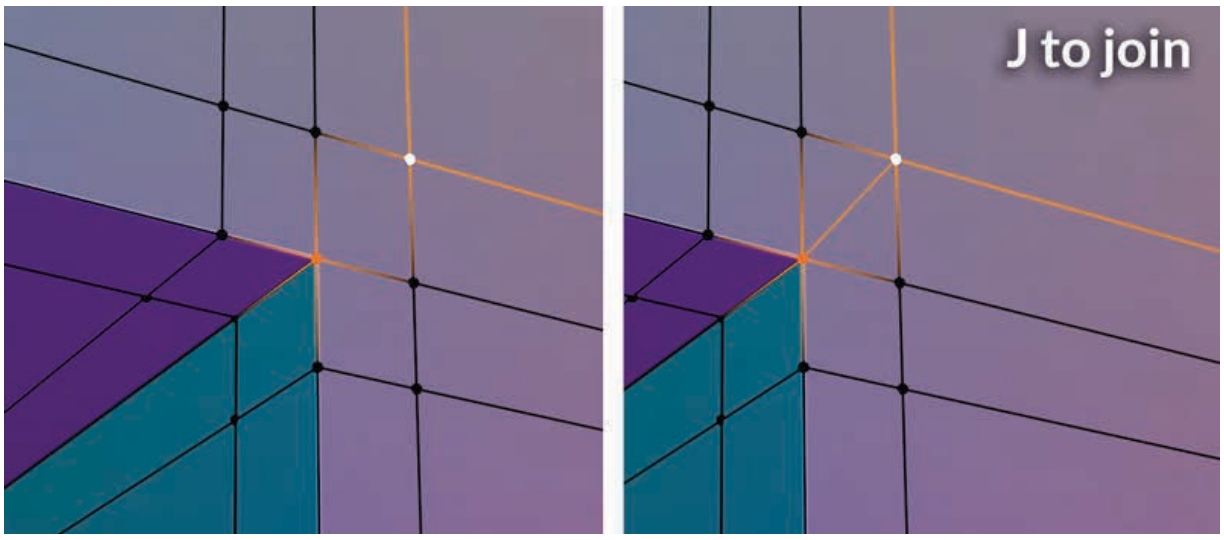
Poles
It's normal and unavoidable, that when you're 3D modeling you end up with poles. Poles are when edges meet in a single Vertex. For example, at the base of an extruded Face.

But although unavoidable, it's desirable to place them in such a way that they cause minimal shading issues.

For example, if you're adding a bevel to sharp corners by adding support loops manually, be careful that you don't have poles in those corners. Otherwise that corner may have some shading artefact when subdivided.



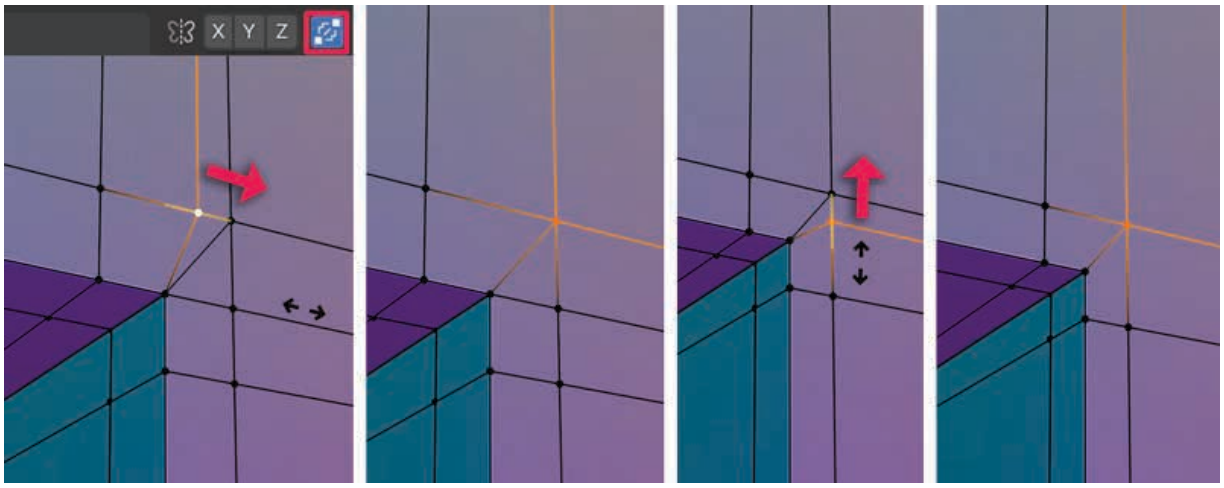
In order to avoid shading artefacts, it's best to move that pole to a flat surface. In this example, it's easy to move the pole to the flat part of the mesh, away from the corner. First, select these two Vertices and create an Edge between them by **pressing J** (for Join).



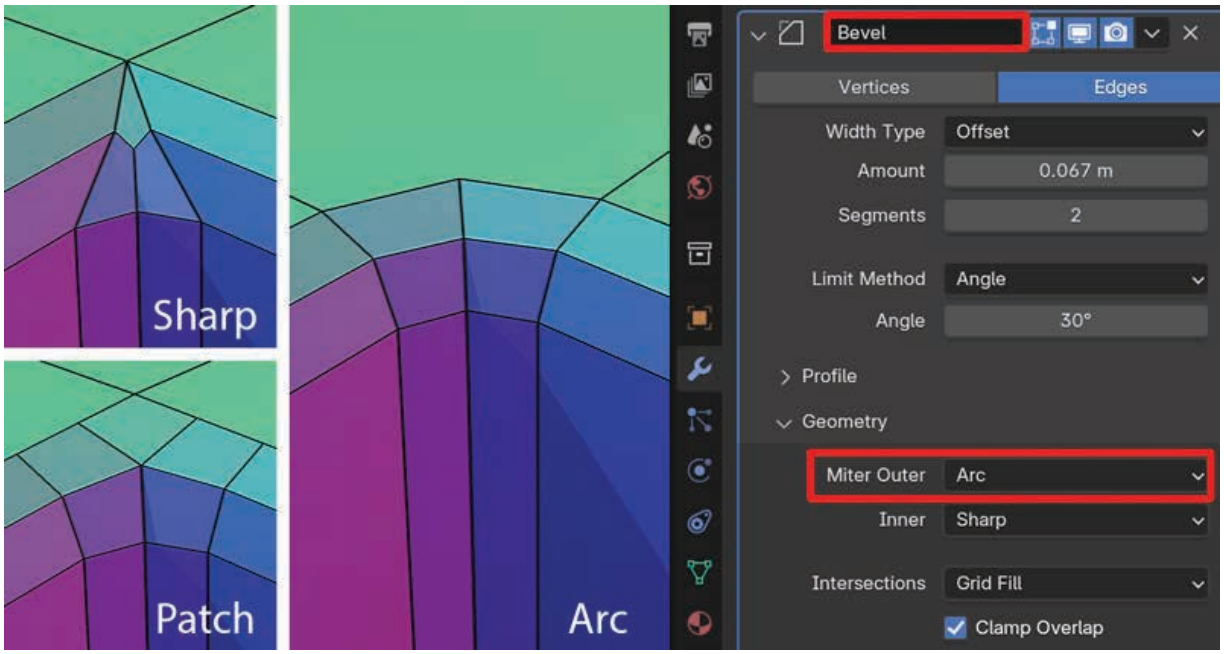
With **Auto Merge Vertices** temporarily enabled, slide the Vertices so that the geometry is simplified and the pole is moved to the flat part of the geometry.

You can slide Vertices by **pressing G twice**. Make sure you select also the other Vertices of that Edge, or select and slide the Edge in Edge selection mode.

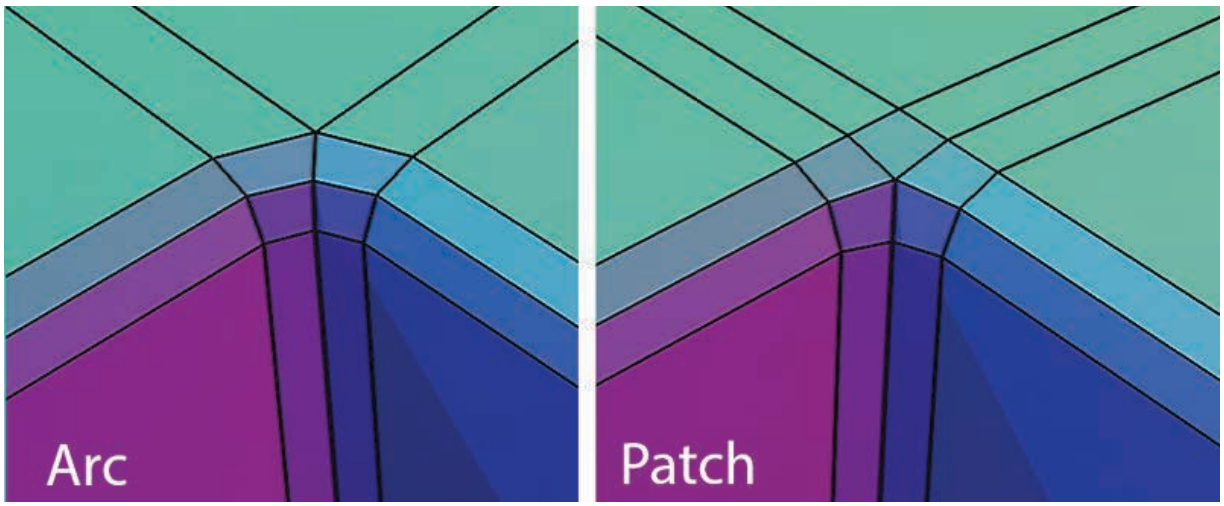
In fact, you don't even really have to join those 2 vertices in the beginning. But it can be handy to see that diagonal Edge for reference so you know where you need to slide things.



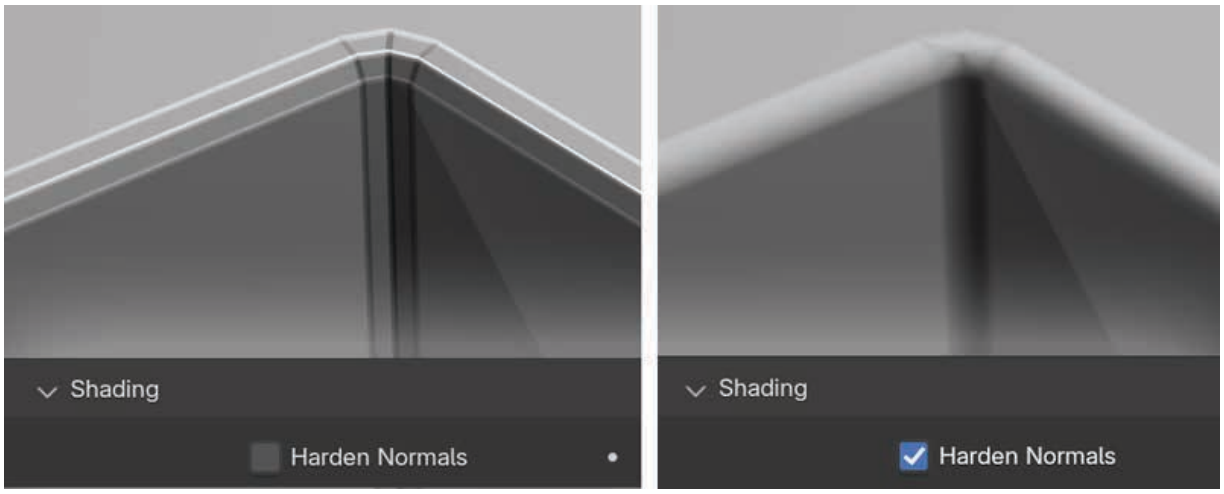
Miter Outer
When using the bevel tool or modifier on concave shapes, set the Miter Outer option to **Arc**. That will give a nicer result than the default **Sharp**. (However, the benefit of Sharp is that you don't need additional supporting Edge loops.)



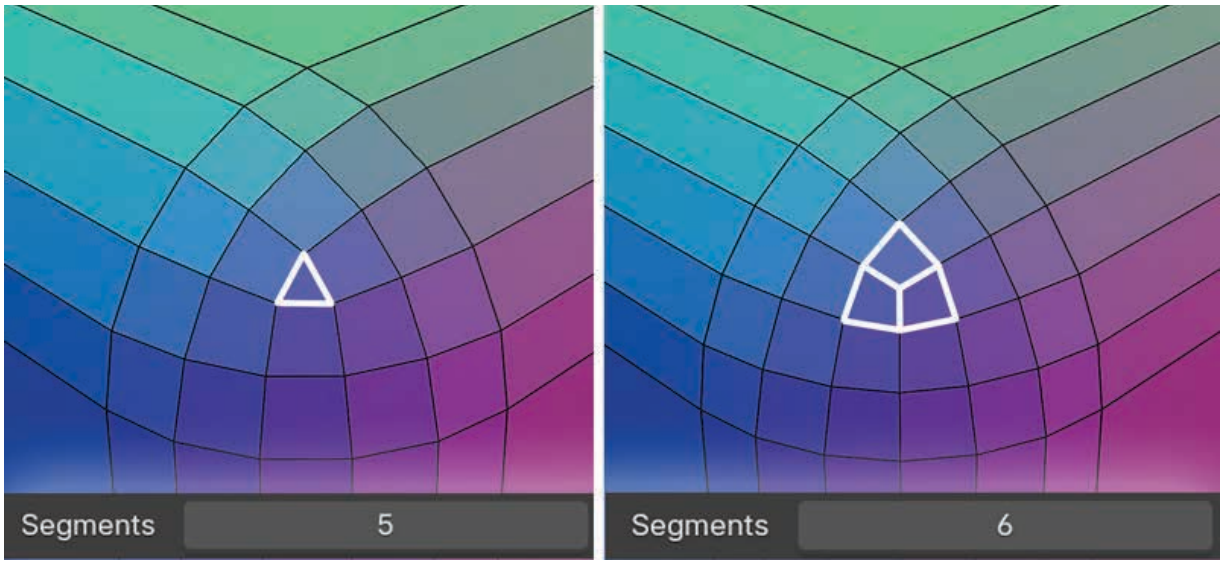
There's also the **Patch** option. But as you can see, this requires more supporting loops if you want to have quad topology. So in this case, Arc is more efficient.



You can then check the **Harden Normals** option to get smooth shading.

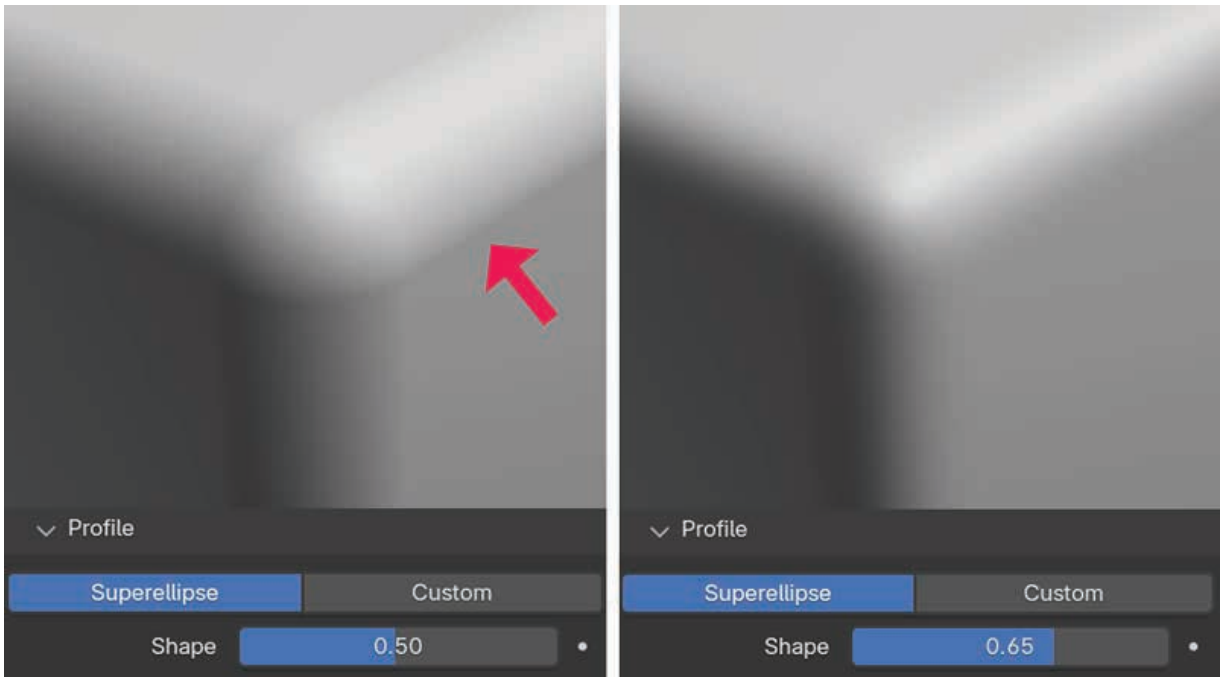


To avoid triangles and keep your bevels all quads, make sure you use an **even number** of **Segments**. Using even numbers in general is a good idea when modeling for quads, by the way.



Shape
For nicer looking bevels, set the **Shape** value higher than the default of 0.5. A value of **0,6 to 0,7** gives a nicer, smoother looking bevel.

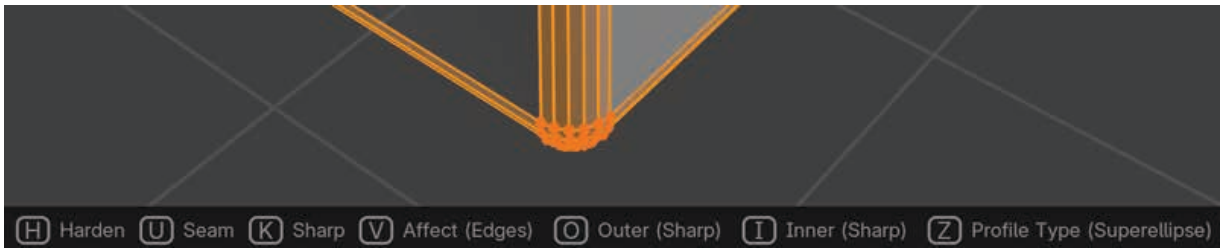
With the default value, you get a very clear and ugly border between the rounded part and the flat surface. With the value of 0.65, it's much more smooth without a clear border between flat and rounded.



If you want to read more about why companies like Apple use this kind of bevel profile in their products, you can read this interesting article.

As a final tip: You can use the **Bevel modifier** in Object Mode, or the **Bevel tool** in Edit Mode. The modifier of course has the benefit of being non-destructive, but if you're in a hurry the Bevel Tool can be very useful.

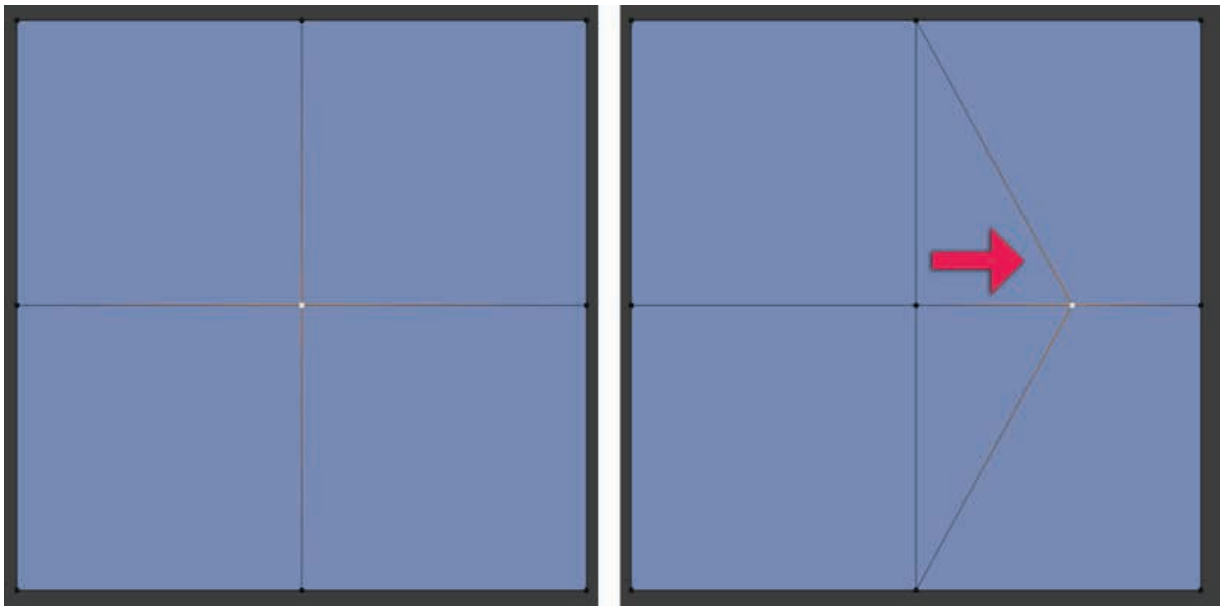
If you forget the many keyboard shortcuts this tool has, remember that you can see them at the bottom of the screen while using it.



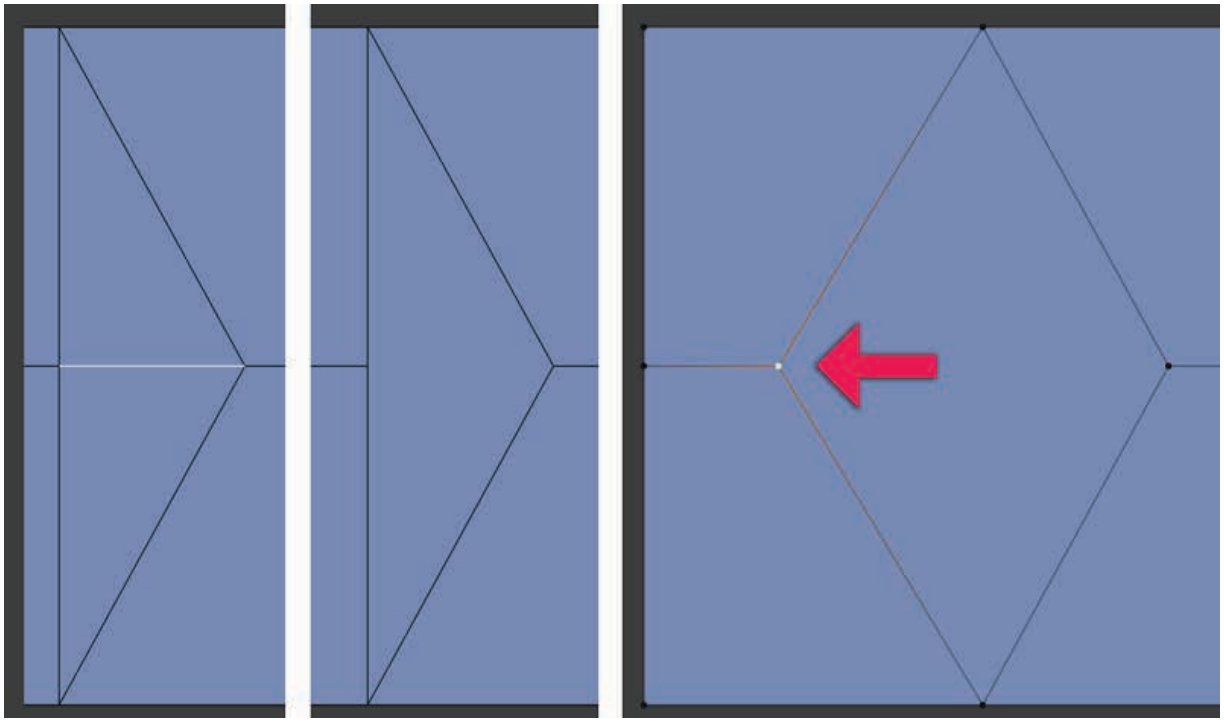
5 ways to add Vertices

Sometimes you need to add a vertex to create some extra geometry. This can be especially useful during the retopology stage. Below are five methods for adding vertices.

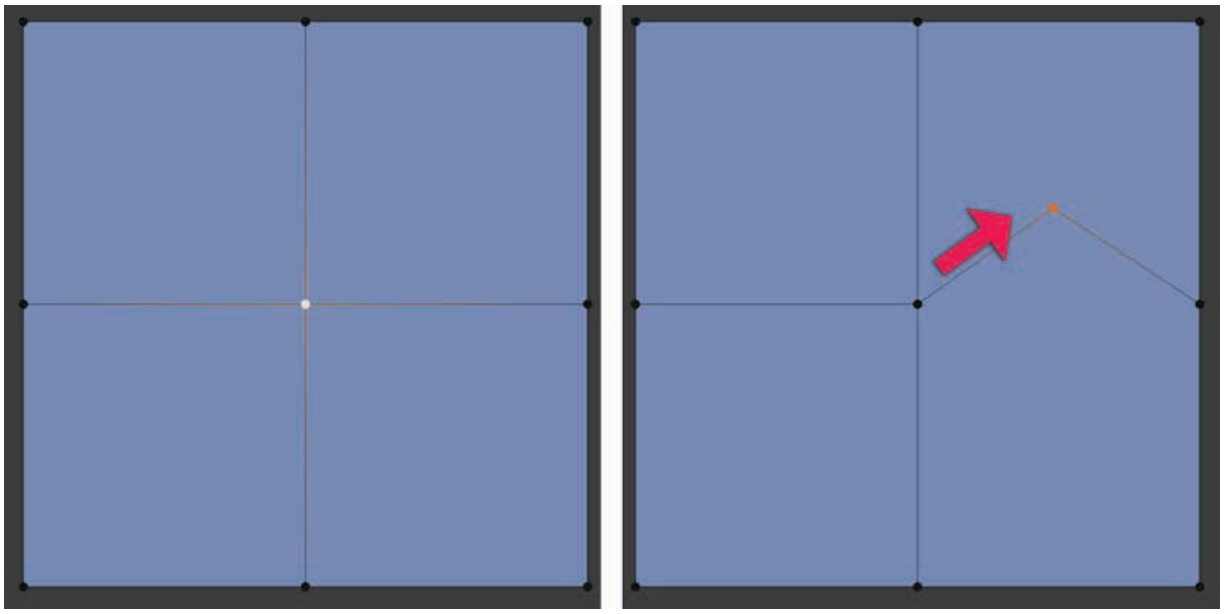
Rip Vertex
Select a vertex and **press Alt+V**. This “rips” the selected vertex, creating a new vertex and also two triangles.



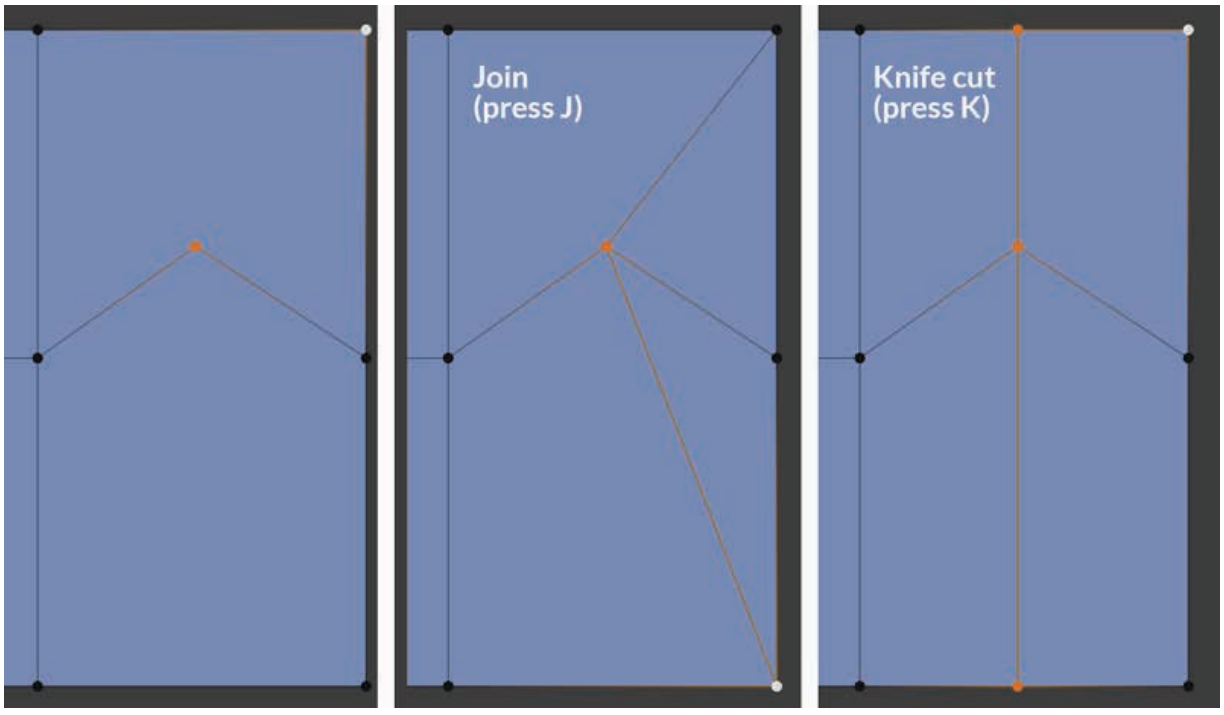
If you select the edge between the triangles and **press Ctrl+X** you dissolve the edge. This way it becomes a quad Face again. You can move the vertices further apart to improve the quad shape.



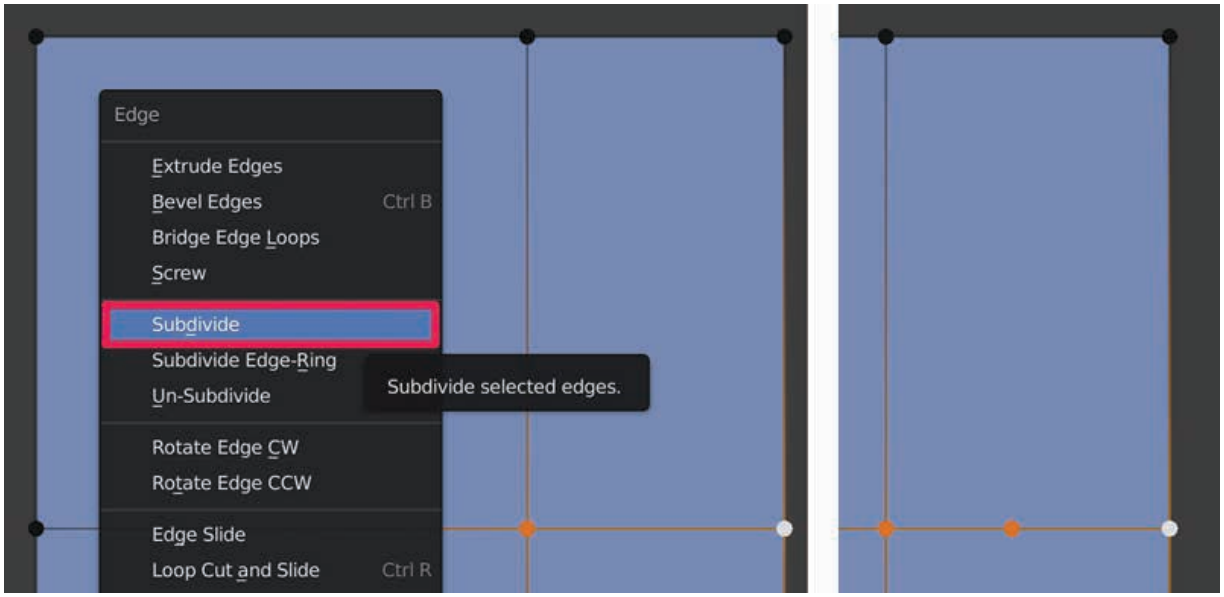
Extend Vertex
Select a vertex and **press Alt+D**. This creates a new vertex and nothing else, but it's still connected to everything.



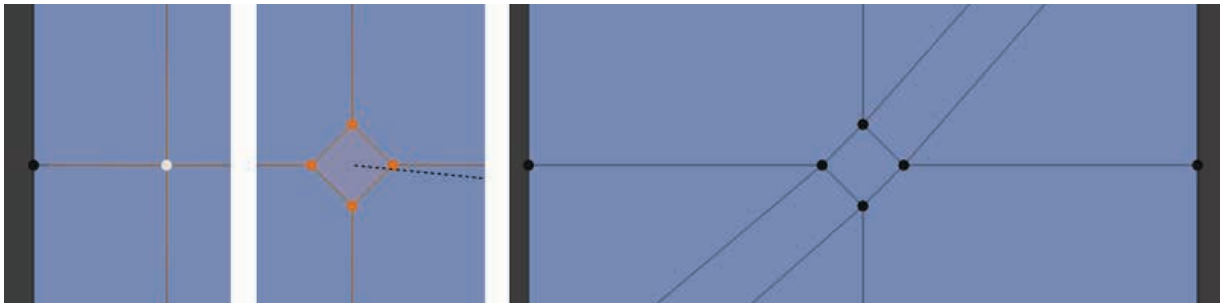
To solve N-gons that come into life this way, select two vertices and **press J** to join them. Or depending on the situation, you can also use the **Knife tool** (activate by **pressing K**) to cut some new edges, maintaining an all-quad surface.



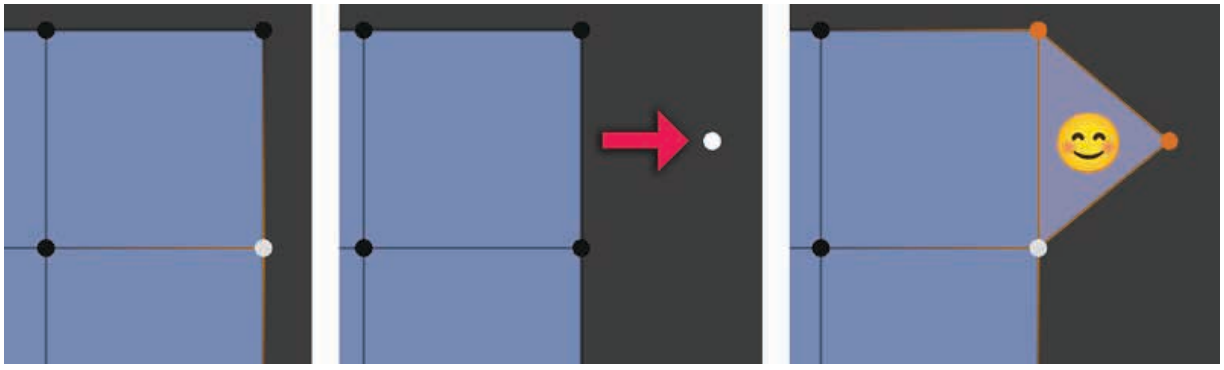
Subdivide Edge
Select an Edge or the two vertices that make the Edge, **right-click** (or **press Ctrl+E** for the Edge menu) and choose **Subdivide**. As for how to solve the N-gons this creates, it's just like the previous method: Join or Knife Cut.



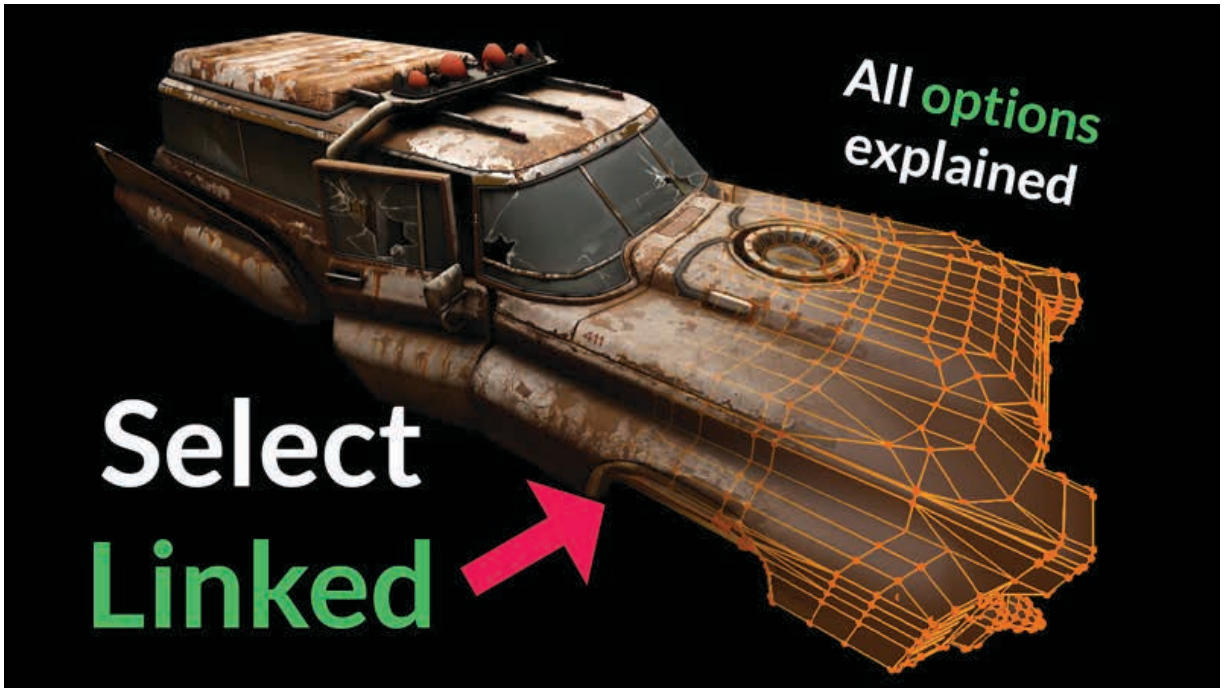
Bevel Vertex
Select a Vertex and **press Shift+Ctrl+B**, then move the mouse to the right. **Scroll the mouse wheel** up or down to influence the amount of vertices. This can be useful for creating holes, or for supporting the flow of some boundary reinforcement edges for example.



Duplicate vertex
Select a Vertex and **press Shift+D**, then move the mouse. Now the vertex is all lonely without any other vertex friends, so select a couple of nearby vertices and **press F**.

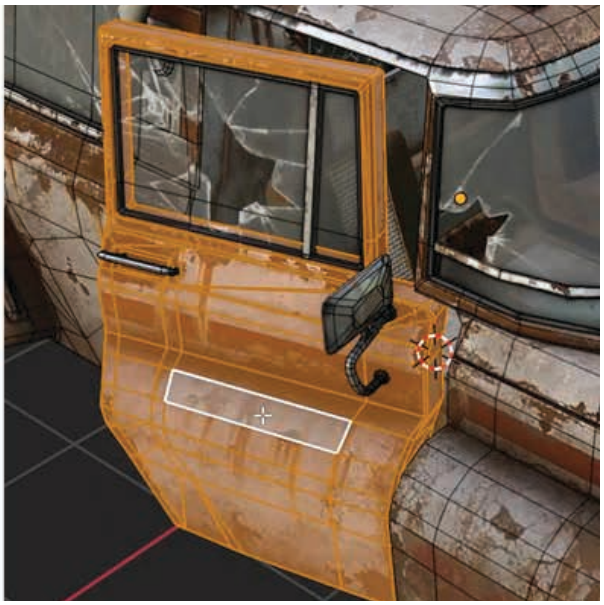


Select Linked - Every Option Explained

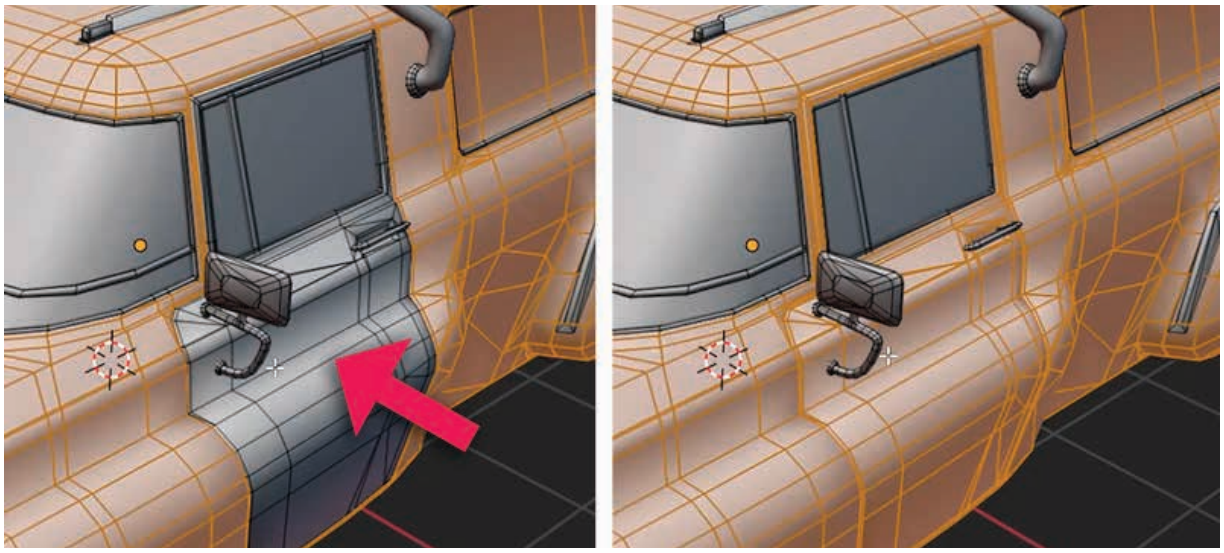


Select Linked helps you select parts of your model faster in Edit Mode. Instead of manually selecting a part of the mesh with for example **Circle Select** (below left), you just select one **Face** and then **press L** to select all the connected Faces of that mesh island.

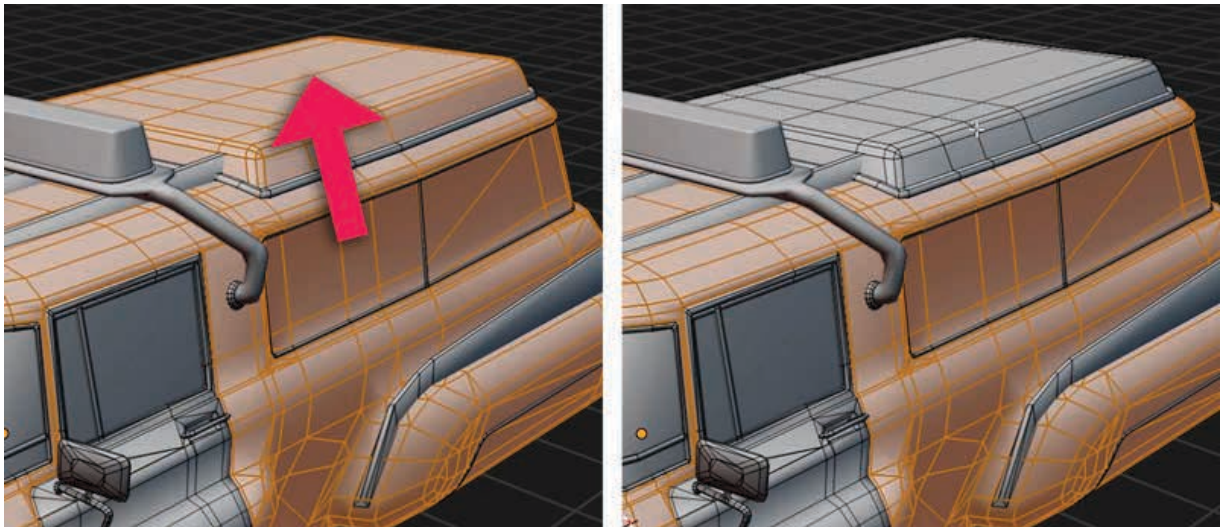
Ctrl+L selects the rest of a mesh island if part of it is already selected.



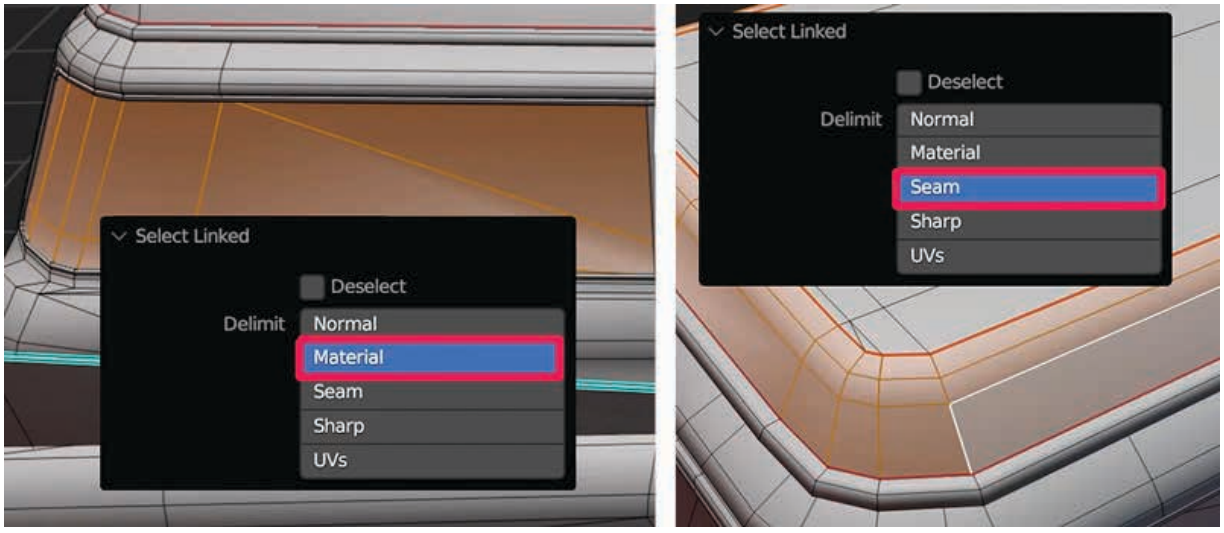
Pressing L selects the linked geometry under the cursor.



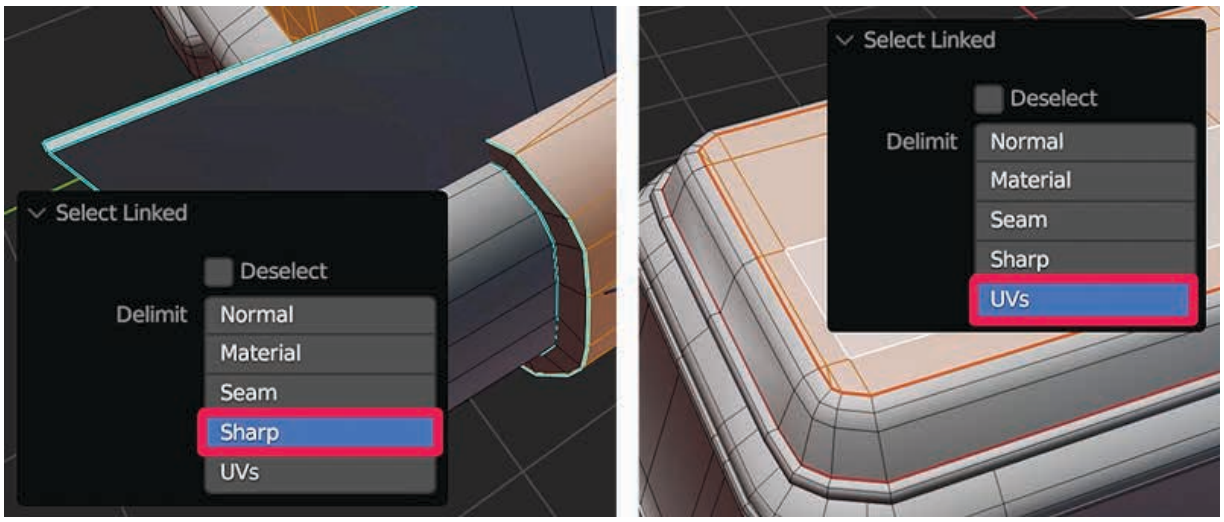
Shift+L deselects, rather than selects, the linked geometry.



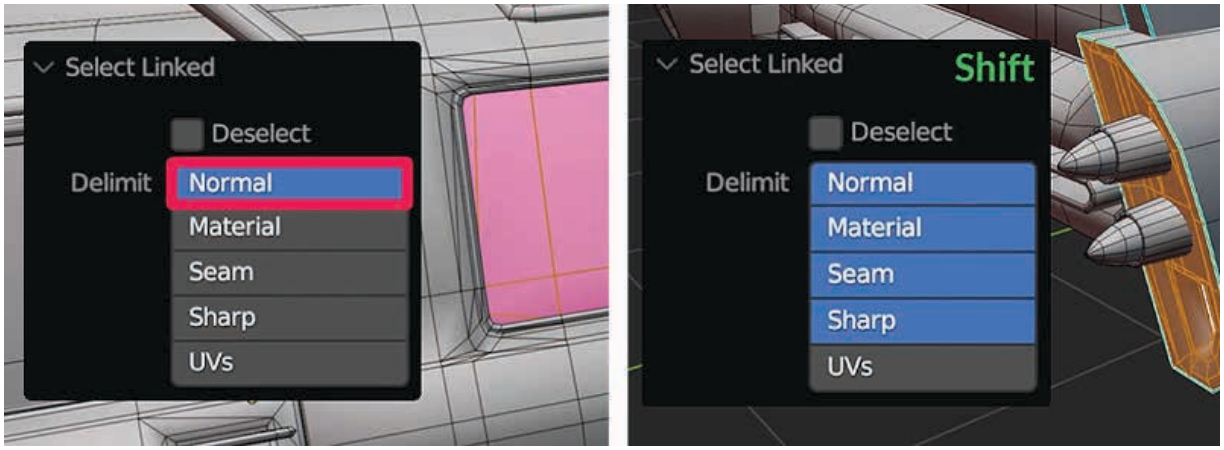
You can further limit the selection: By **materials**, By Edges marked as **Seam**…



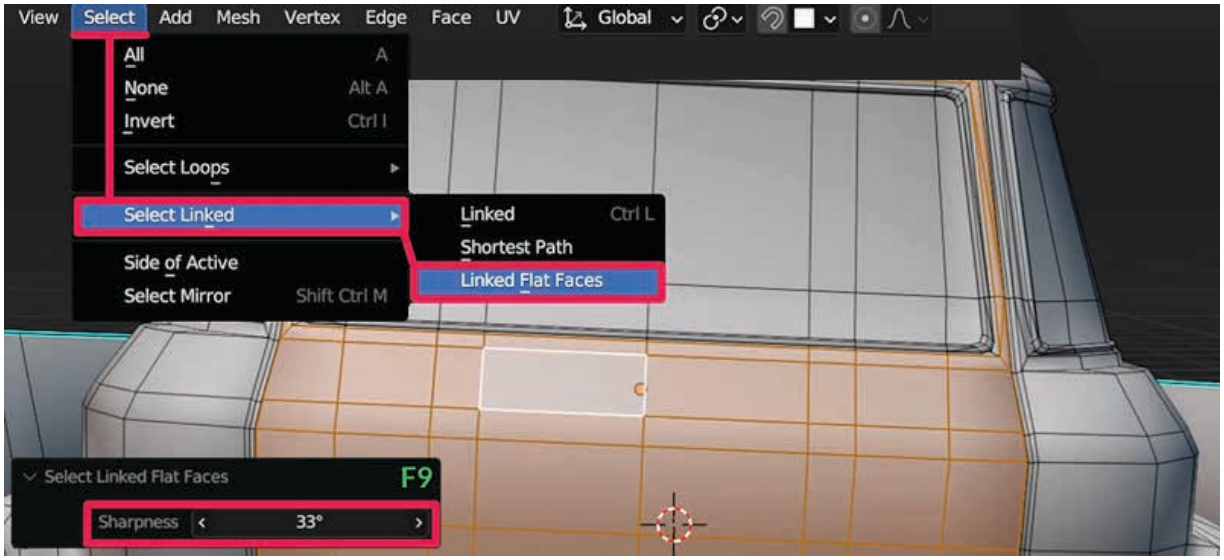
Edges marked as **Sharp**, by **UV** islands…



Or by flipped **Normals**. Hold **Shift** if you want to select or deselect multiple of these options.

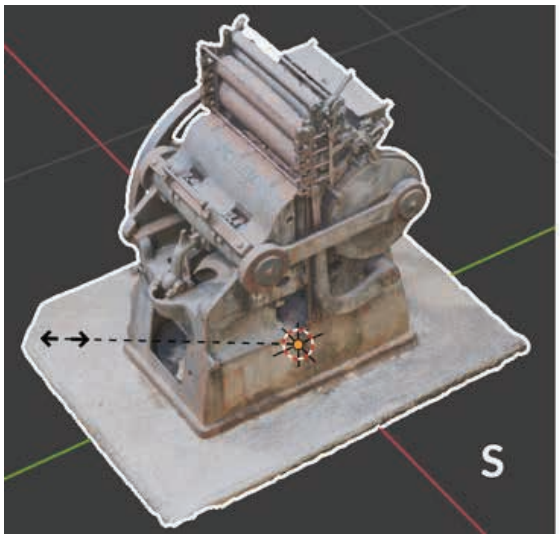


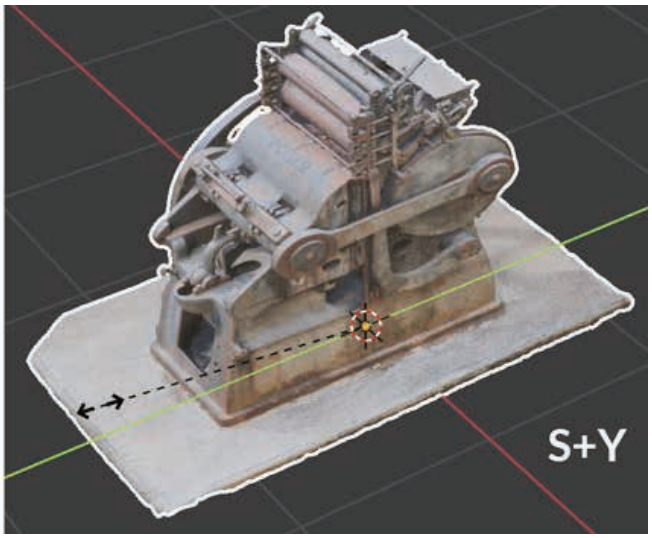
You can also go to **Select > Select Linked > Linked Flat Faces**. This selects all other Faces with the same Normal angle. You can then adjust the angle under which you want to select other Faces, in order to expand or contract the selection.



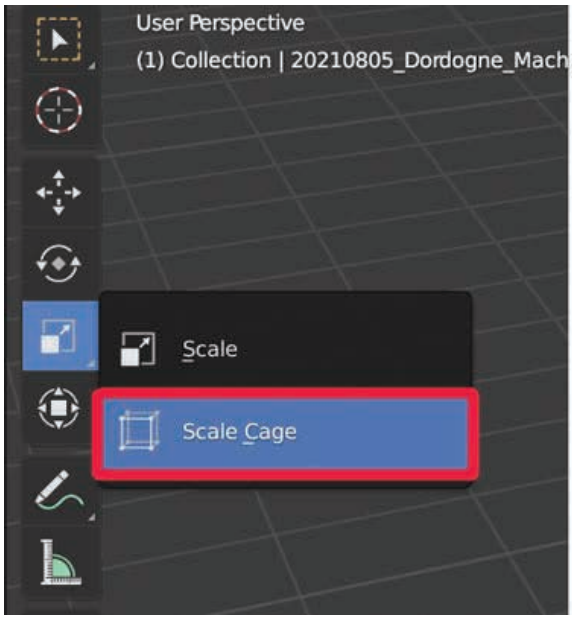
The Scale Cage

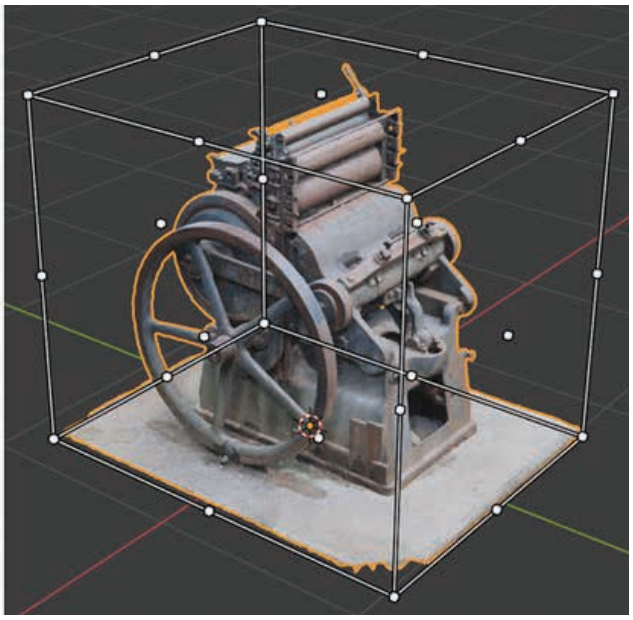
You probably know that you can scale objects uniformly by **pressing S**, or on an axis by pressing S and that axis, like S+Y for example to scale along the Y-axis.



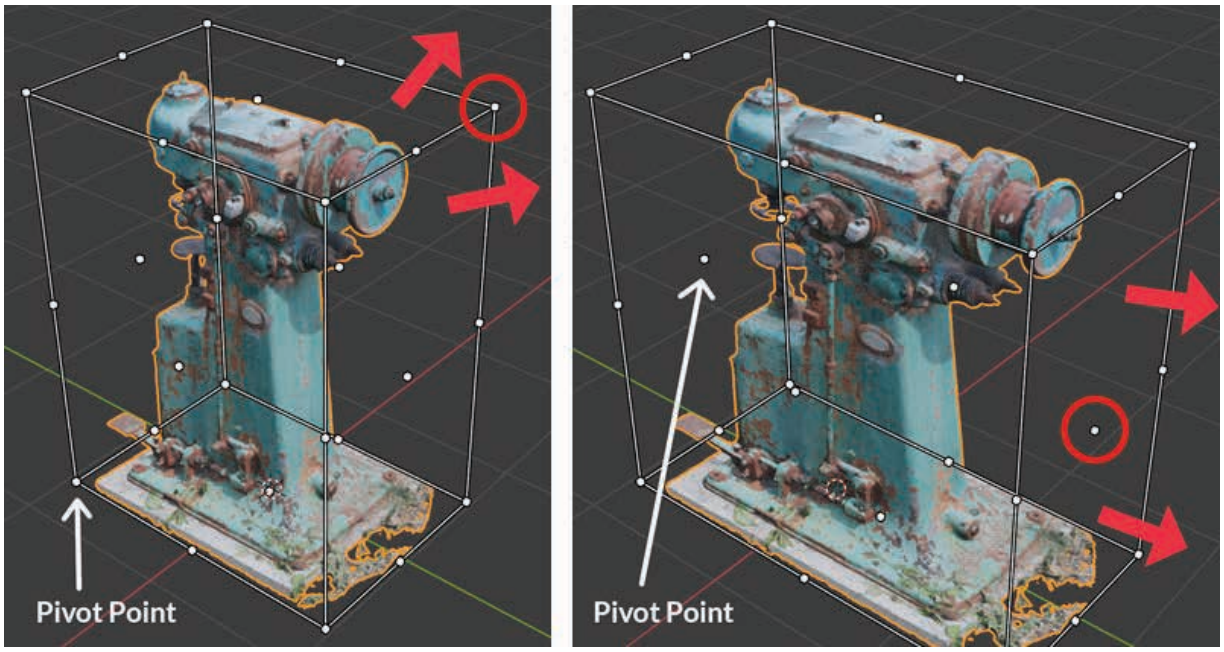


A lesser known tool for scaling objects is the **Scale Cage**, which is accessed from the **Tool panel**. This adds a cage around the selected object or multiple objects which you can use to scale them.

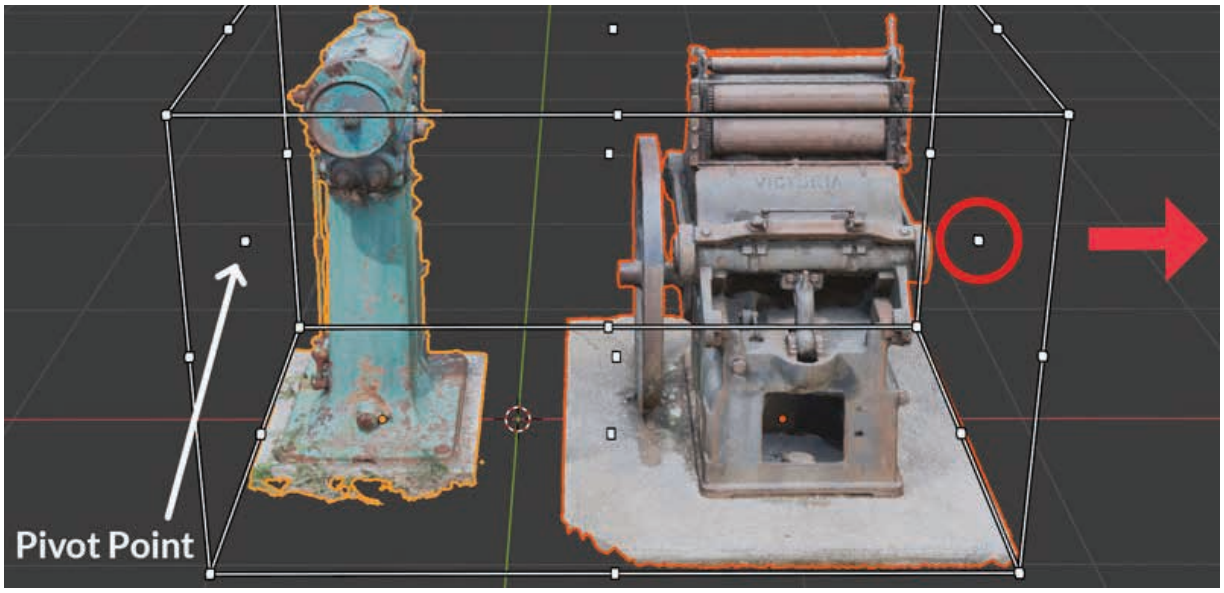




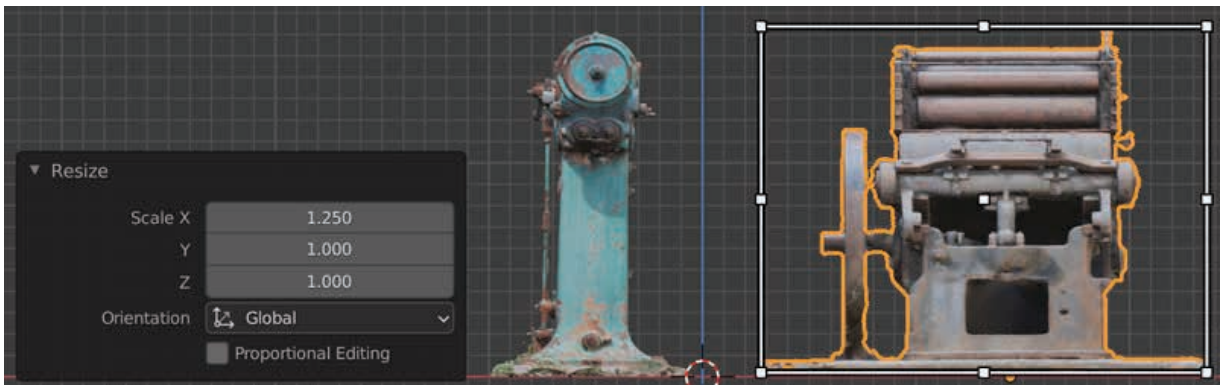
you can then scale either uniformly by dragging the corner handles, or along an axis by dragging the handles on the sides. It always uses the opposite handle as the pivot point.



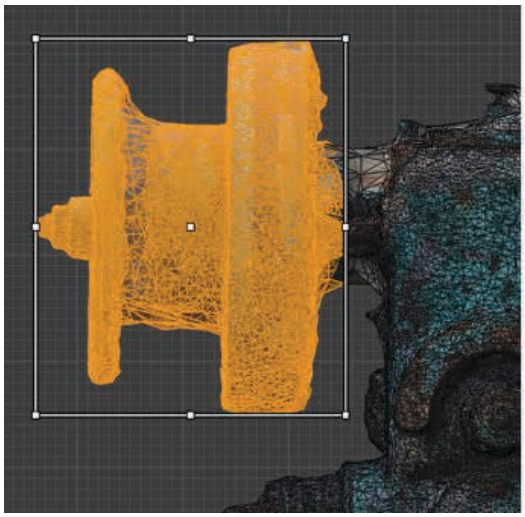
This can be very handy if you want to scale multiple objects away from a specific point, for example.

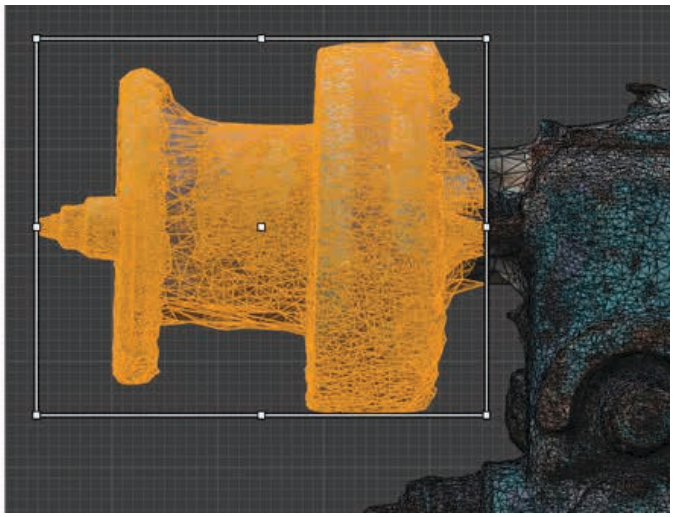


The tool has some options, for example you can type a scale value for each axis or use Proportional Editing. These options become available after you use the tool.



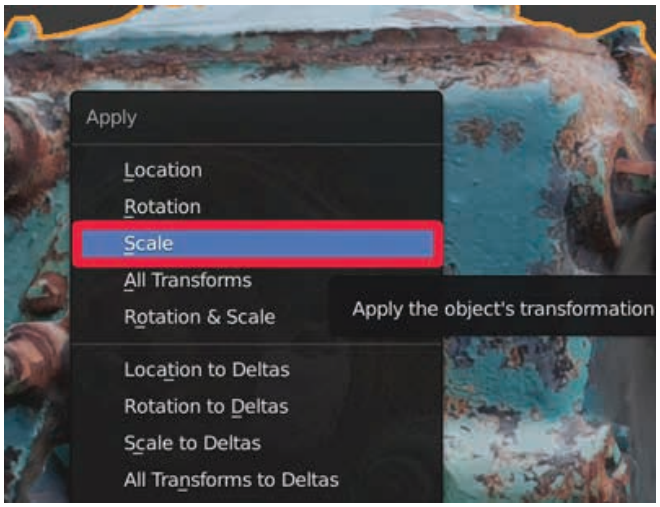
It works in Edit Mode as well.





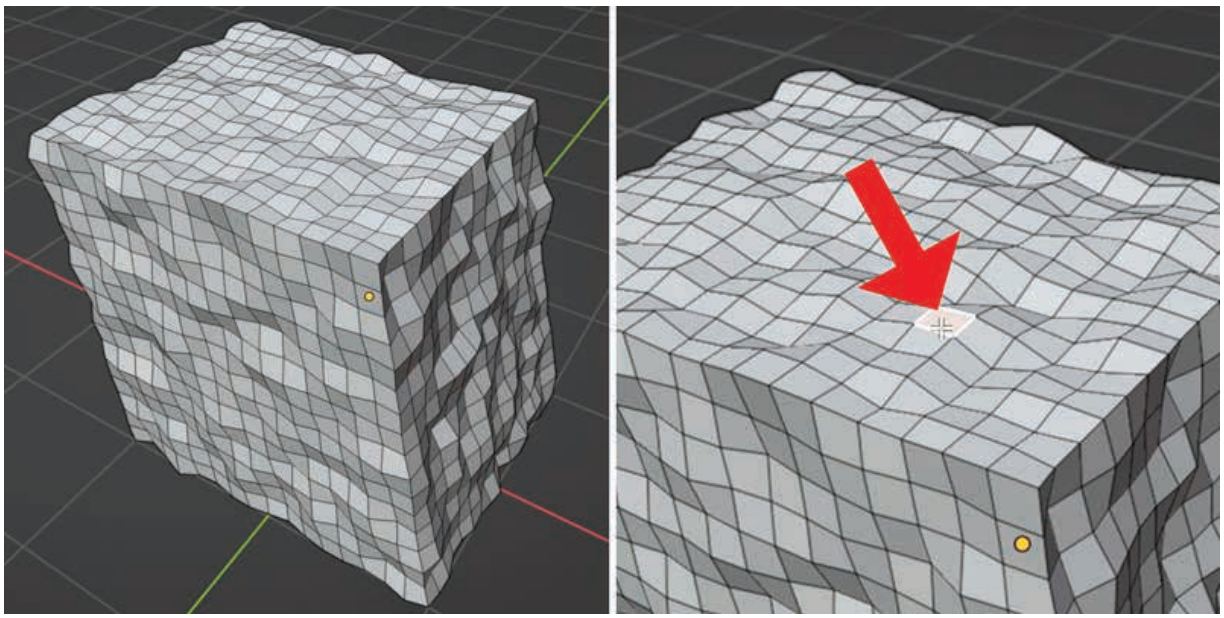
Don't forget to apply scale after scaling objects in Object Mode by **pressing Ctrl+A > Scale**.

Otherwise, you may run into issues later, like bevels not working as desired.



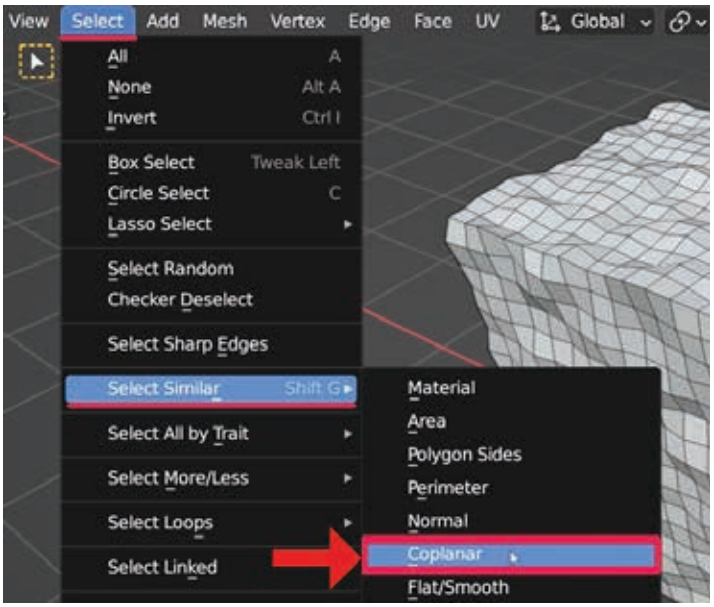
3 ways to Flatten Faces

Select the faces that you want to flatten.

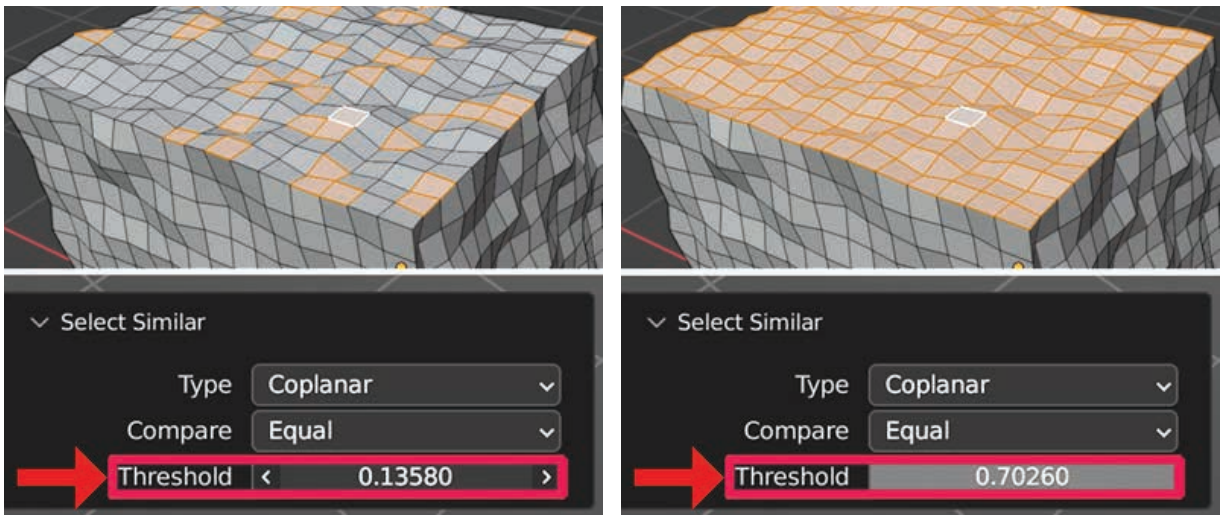


Go to **Select Similar**.

Choose **Coplanar**.

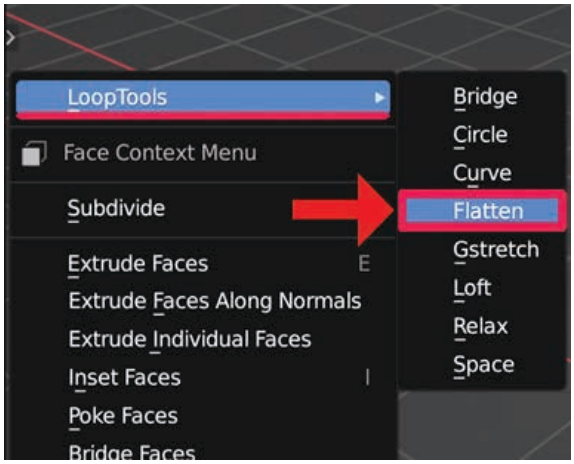


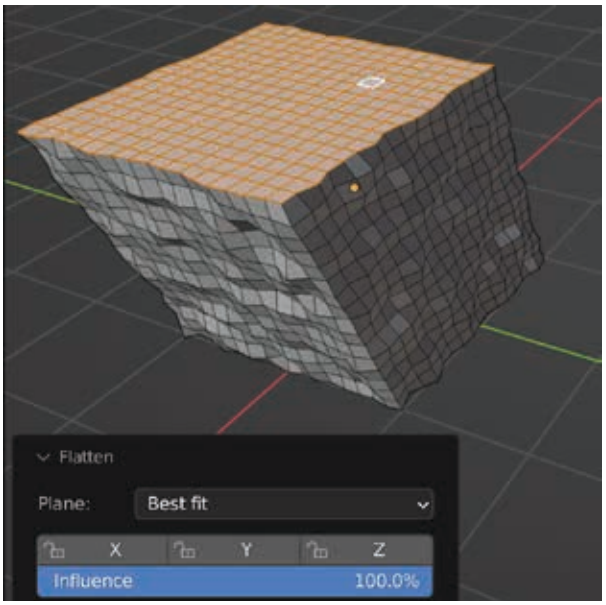
Then increase the **Threshold** if necessary.



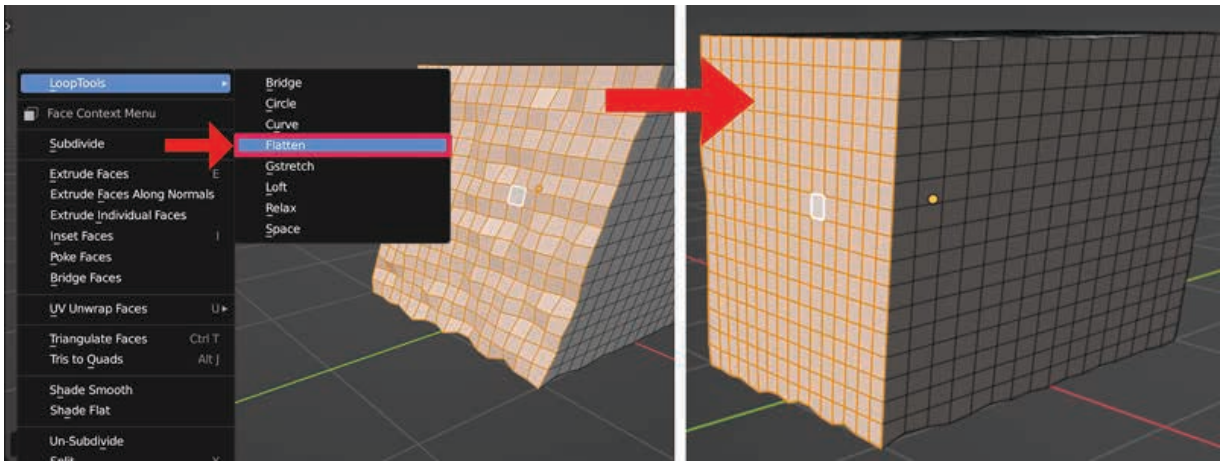
Click here to return to the index.

Right-click, select **LoopTools** and choose **Flatten** (LoopTools needs to be installed in Get Extensions in Preferences).

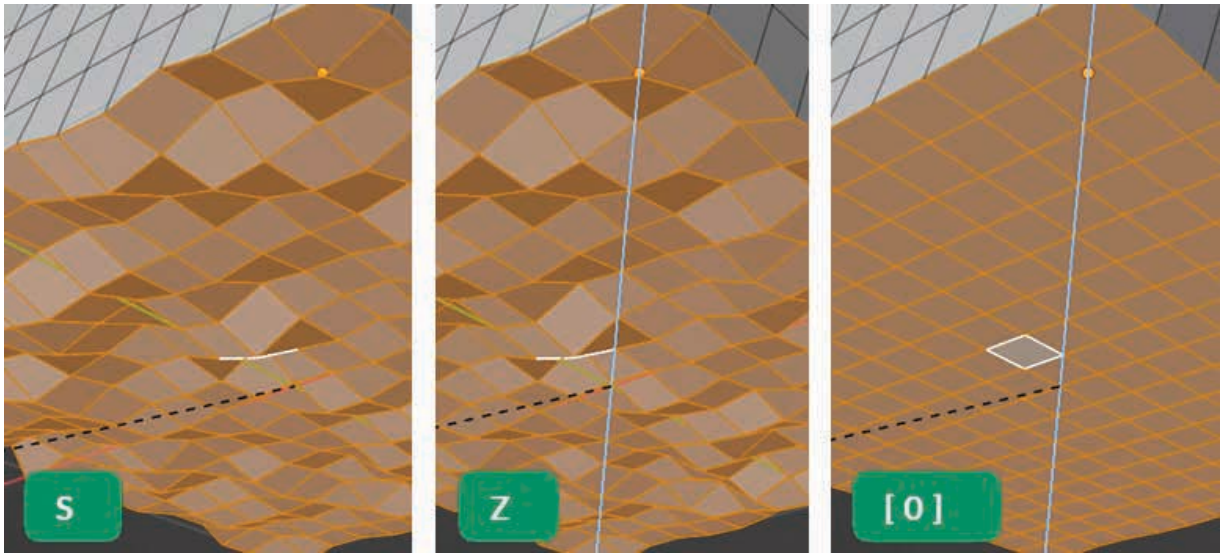




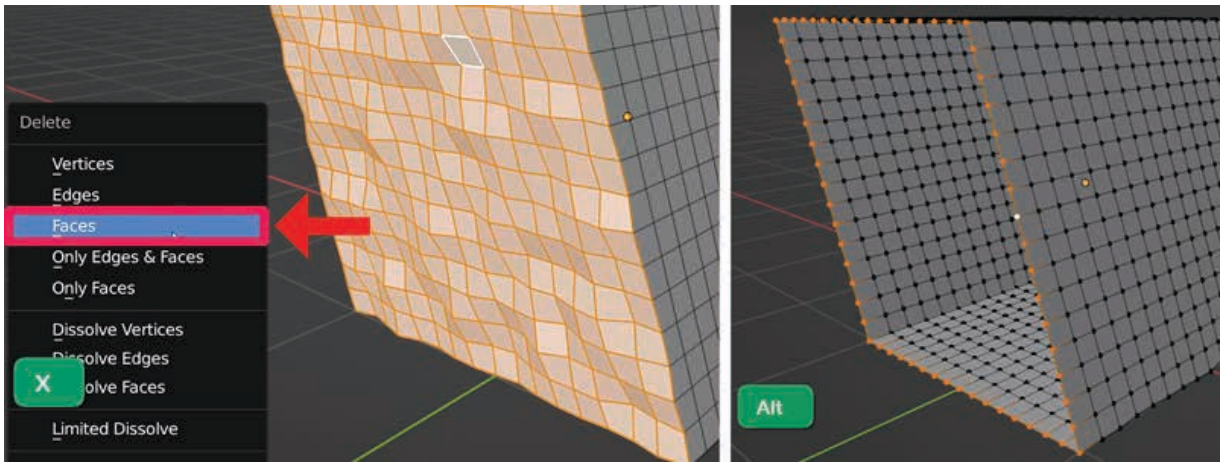
The good thing about this method is that it works no matter the angle of the faces.



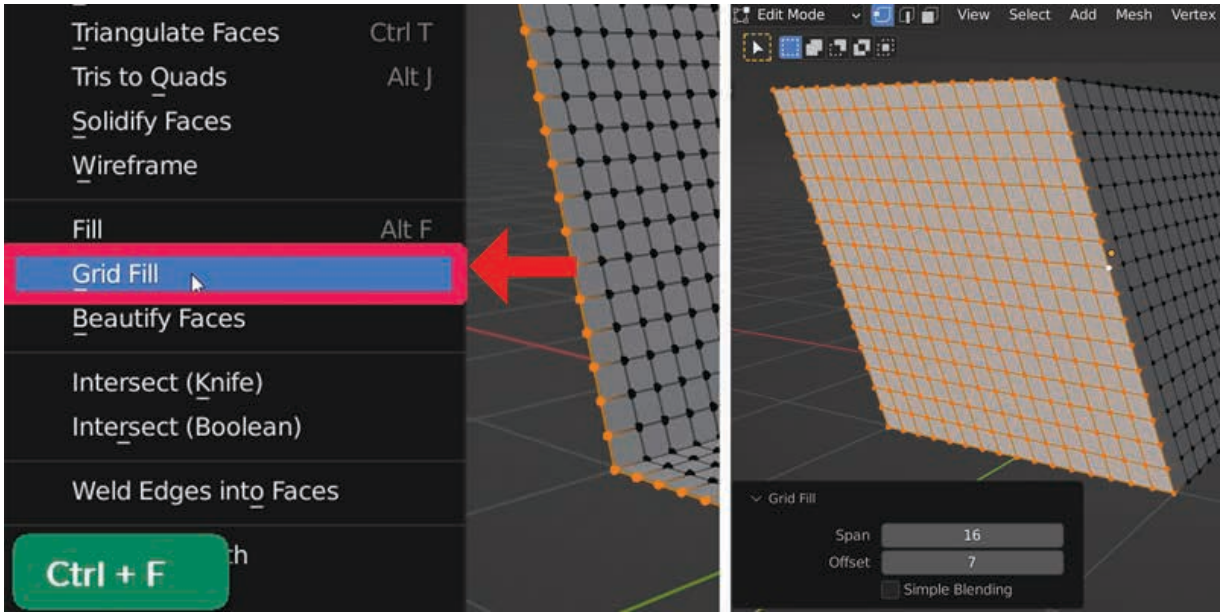
Another way is to scale things to zero. In other words, select the faces, then **press S + Z** (for the **Z**-axis) and zero. This works best if the faces are aligned perpendicular to the exact axis you are scaling to.



As a third method, you can delete the selected faces, select the edges...

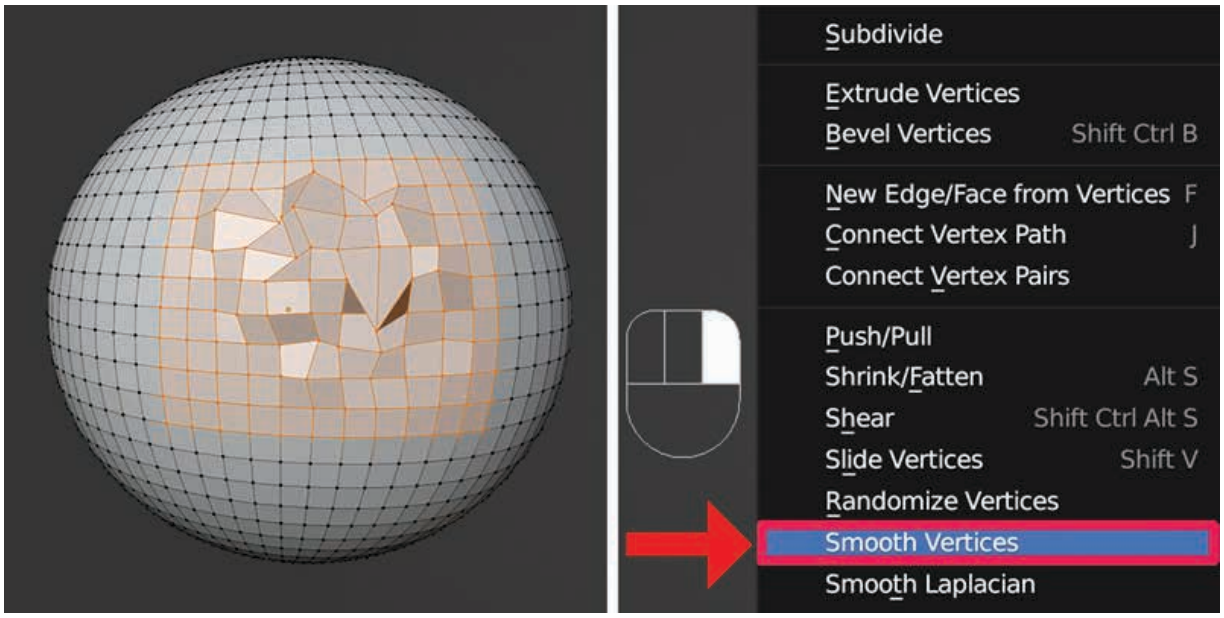


...and then fill the hole by **pressing Ctrl+F** and choosing **Grid Fill**.

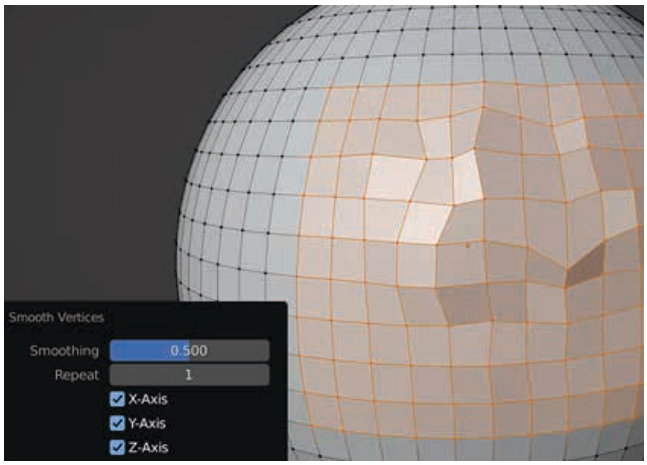


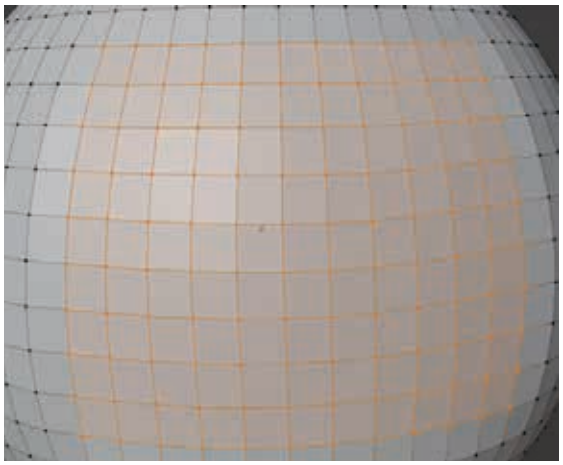
3 ways to Smoothen areas

If your model has some areas that need to be smoothened out, you have 3 options. You can select the offending vertices, right-click and choose **Smooth Vertices**.

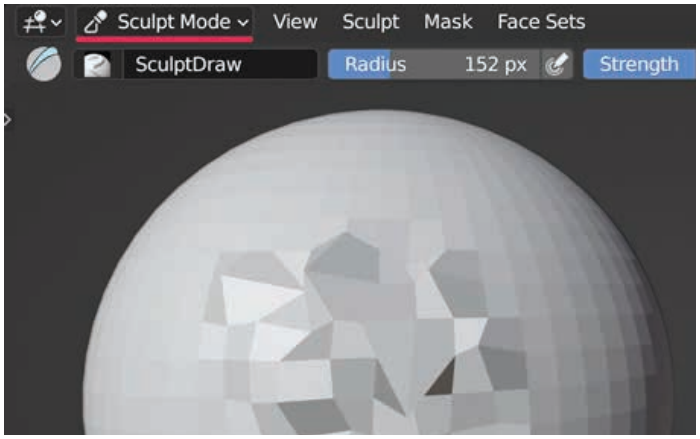


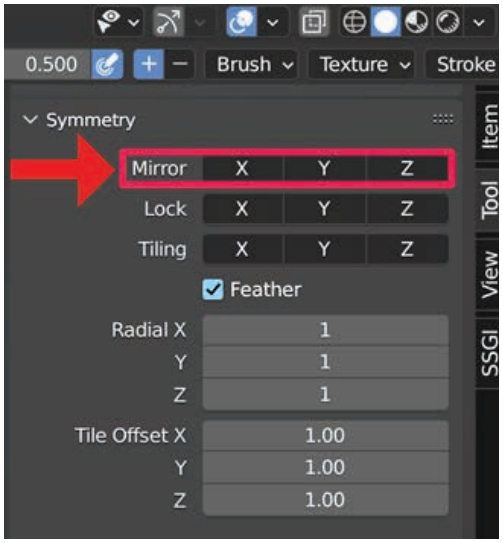
Then hit **Shift+R** a few times, to repeat the last action until it looks smooth enough.



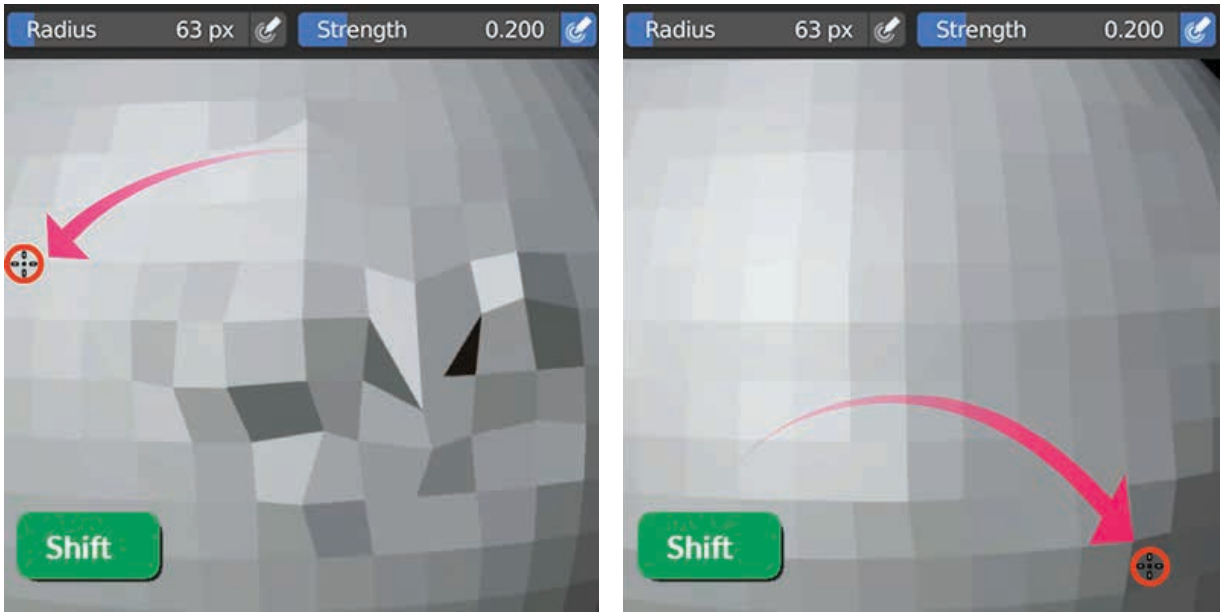


Another option is to go over to **Sculpt mode** - be sure to turn off Symmetry...

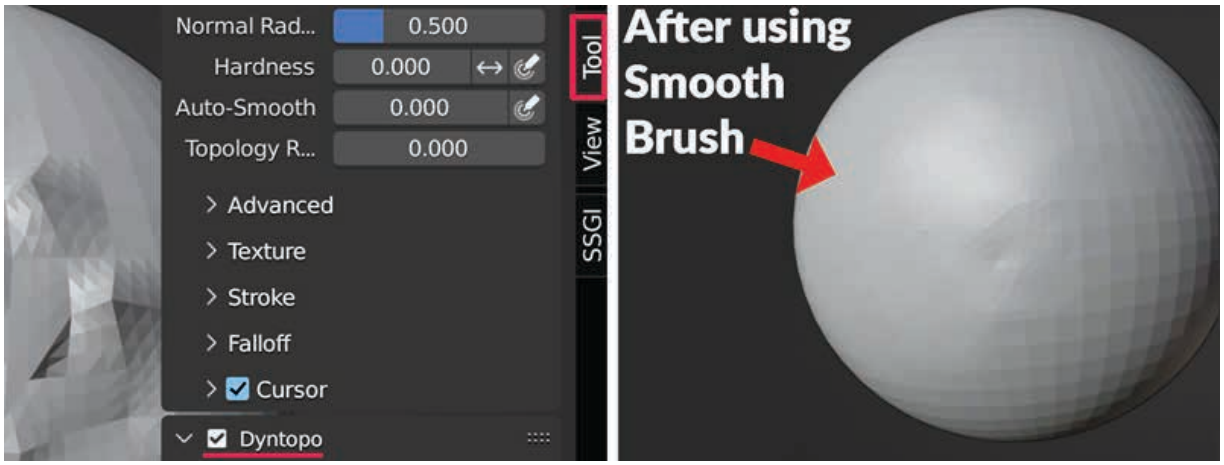




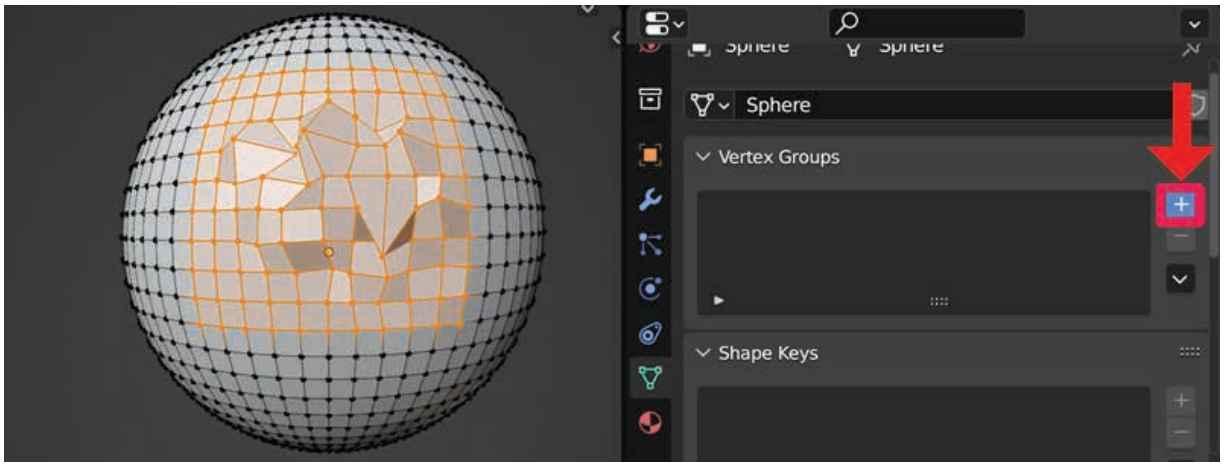
...and sculpt on the model while **holding down Shift**.



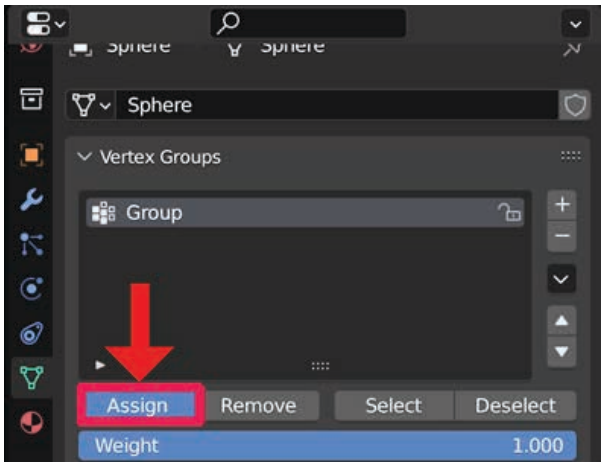
This smooths out the surfaces as well but works best when you have more subdivisions or with **Dyntopo** enabled. It takes some practice to get it right.

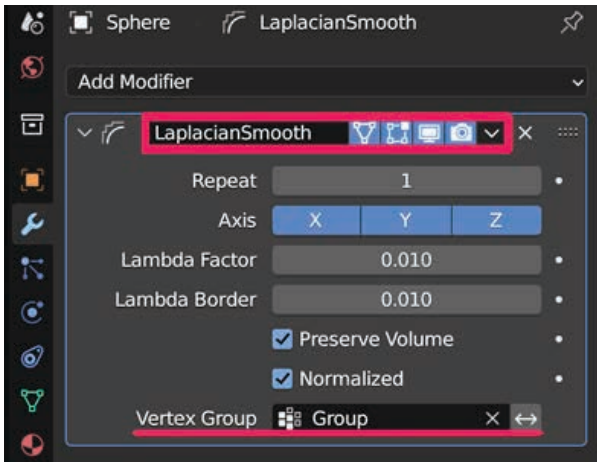


You can also add these vertices to a **Vertex Group** and then use that Vertex Group in a **Smooth modifier** or a **Laplacian Smooth modifier**. To do this, first select the vertices you want to smooth out, then **click on the + icon** in the Vertex Groups list.

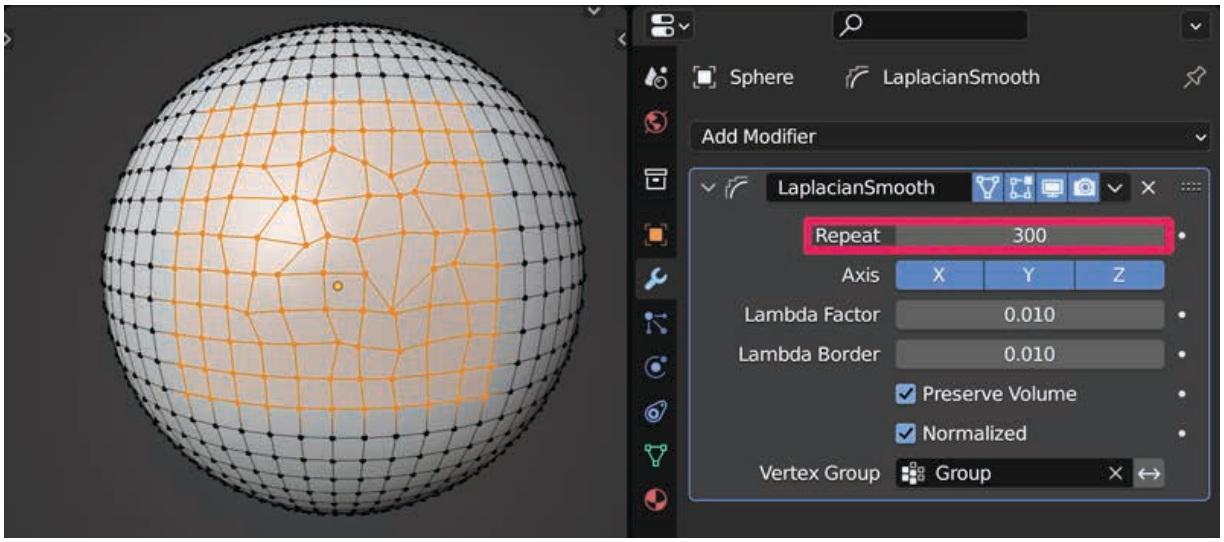


Then click on **Assign** with a Weight of 1. Add a **Laplacian Smooth modifier** and in the **Vertex Group field**, choose the Vertex Group (called “Group” by default).



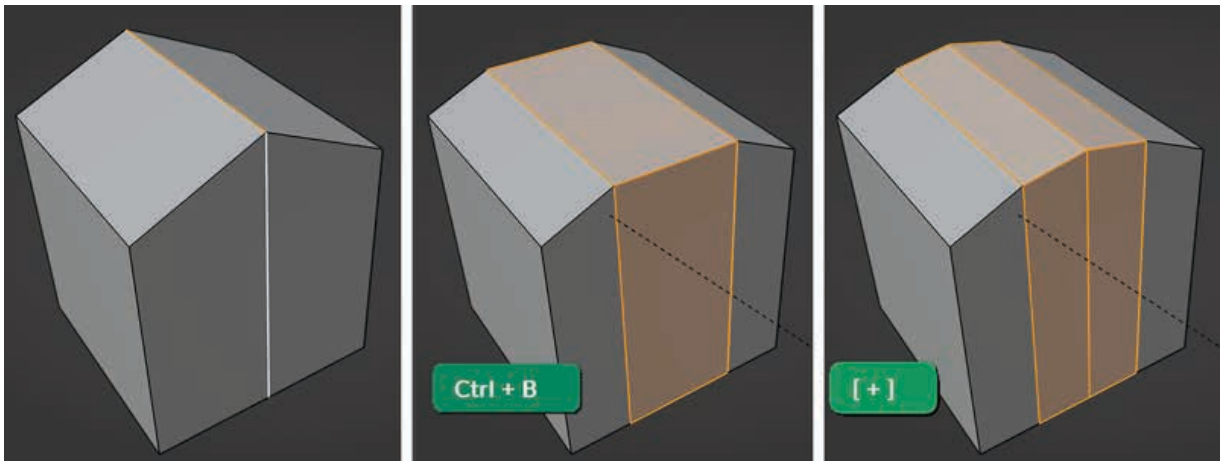


By increasing the **Repeat** value, you smooth the area out further.

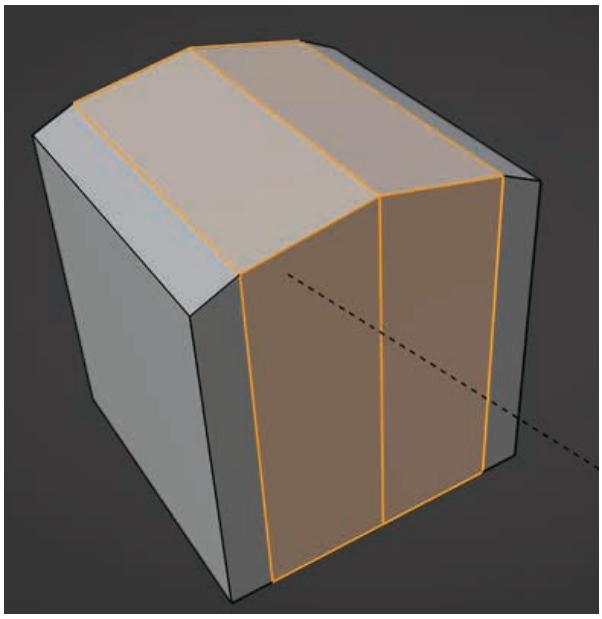


Clever way to add Support Loops

Method 1: To add support loops on both sides of an edge you could simply bevel that edge by p**ressing Ctrl+B** and increase the loops to 3 by scrolling the middle mouse button or **pressing the + (plus) key** on the numpad.

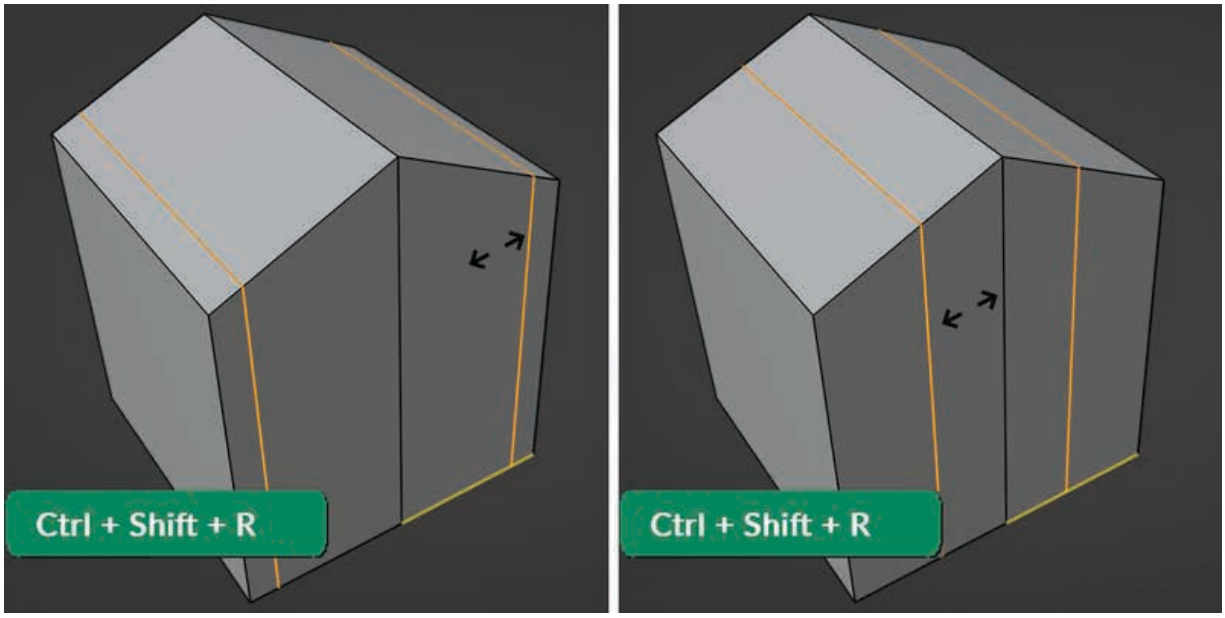


But as you can see, this transforms the middle edge as well, bringing it down.



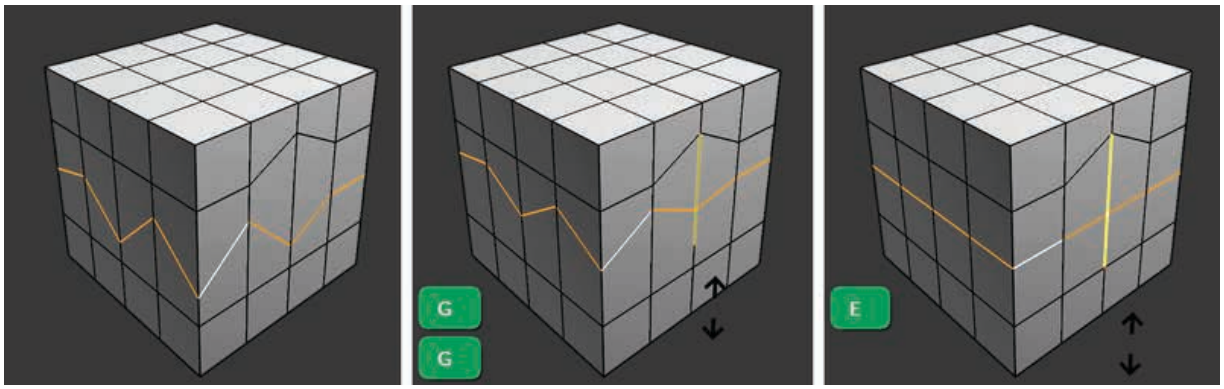
Method 2: If you don't want that, a better method is to **press Shift+Ctrl+R**.

This adds new edge loops at both sides of the selected edge without transforming it. Move the mouse to slide the edges before confirming with the left-mouse-button.

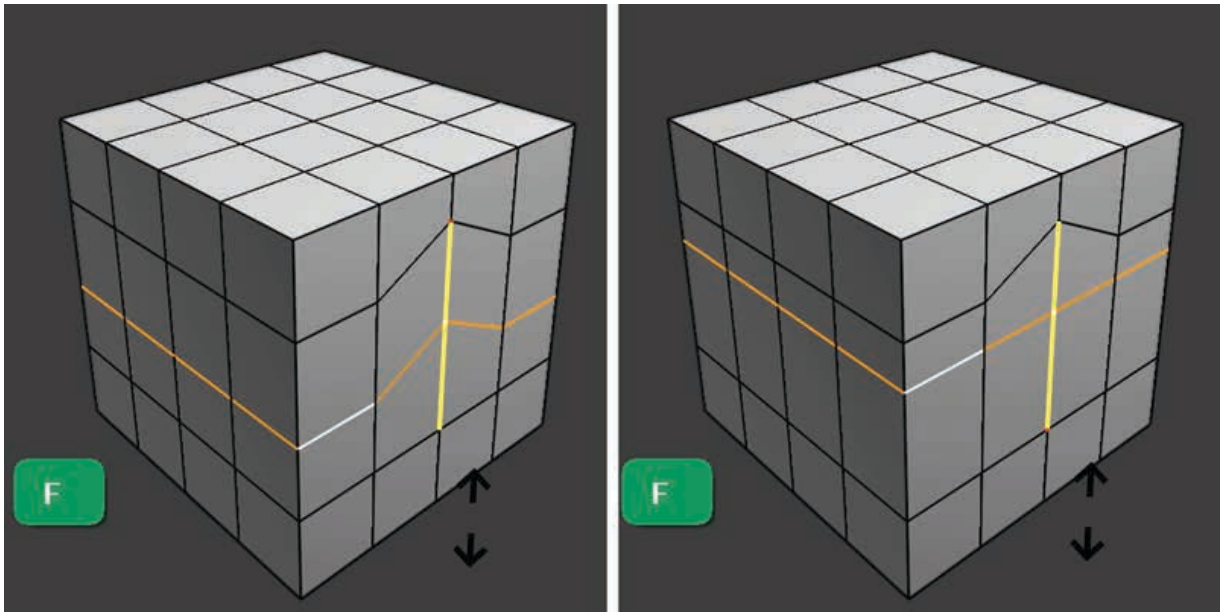


3 ways to Straighten Edge Loops

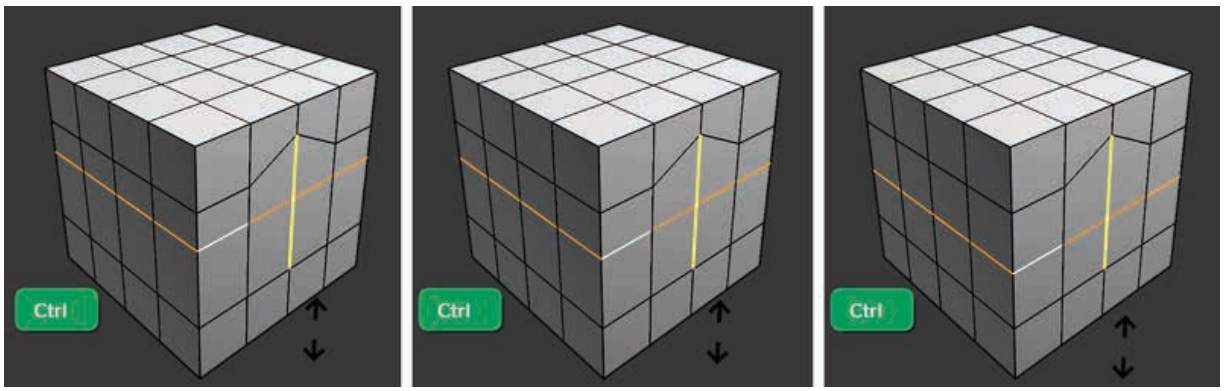
Method 1: In case you have an edge loop that you need to make perfectly straight you can **press G twice and then press E**.



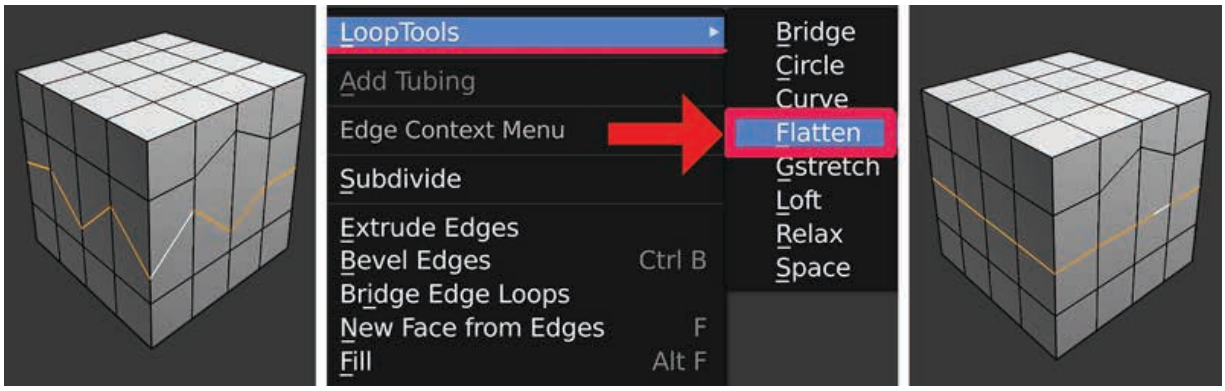
Pressing E conforms the edge loop to its neighbouring edge loop so it only works if there's a perfectly straight edge loop next to it. **Press F** to toggle which edge loop to confirm the shape to.



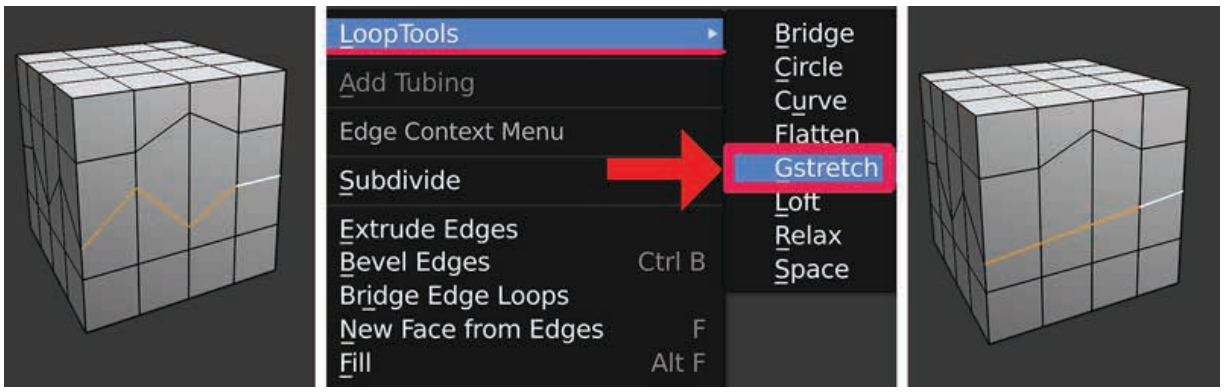
If you want to make sure you place the edge loop in the middle **hold down CTRL** to move it in increments.



Method 2 is to use the **LoopTools** addon (this needs to be installed in **Get Extensions** in Preferences). Select the edges and right-click choose LoopTools and then **Flatten**.

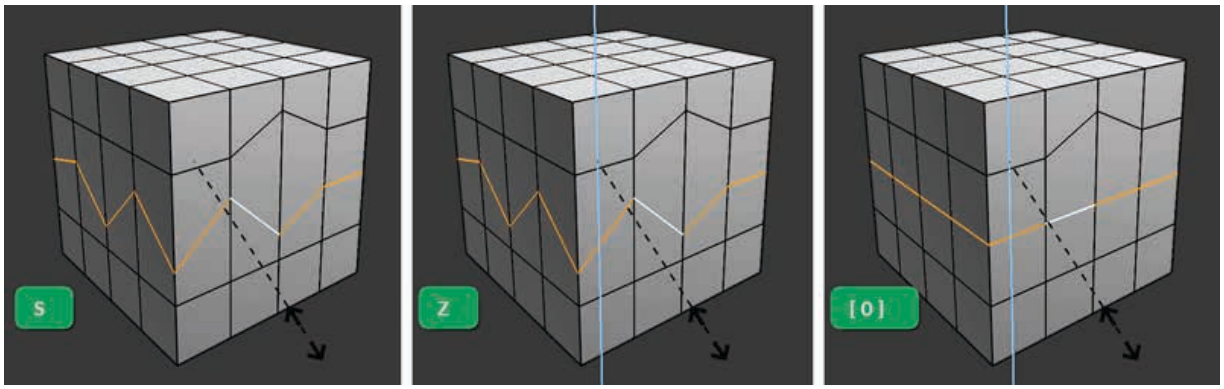


If you only want to straighten out the edges on one side of the model or a limited amount of edges instead of the entire loop use the **Gstretch** option instead.



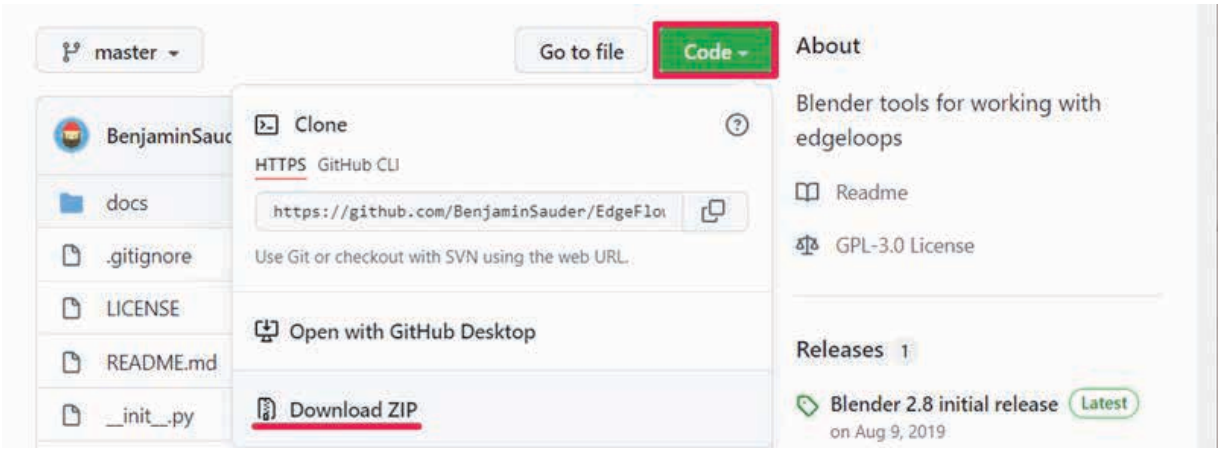
Method 3 is to scale to 0 (Zero).

In this case we scale along the Z-axis. **Press S, Z and then 0** on the numpad.

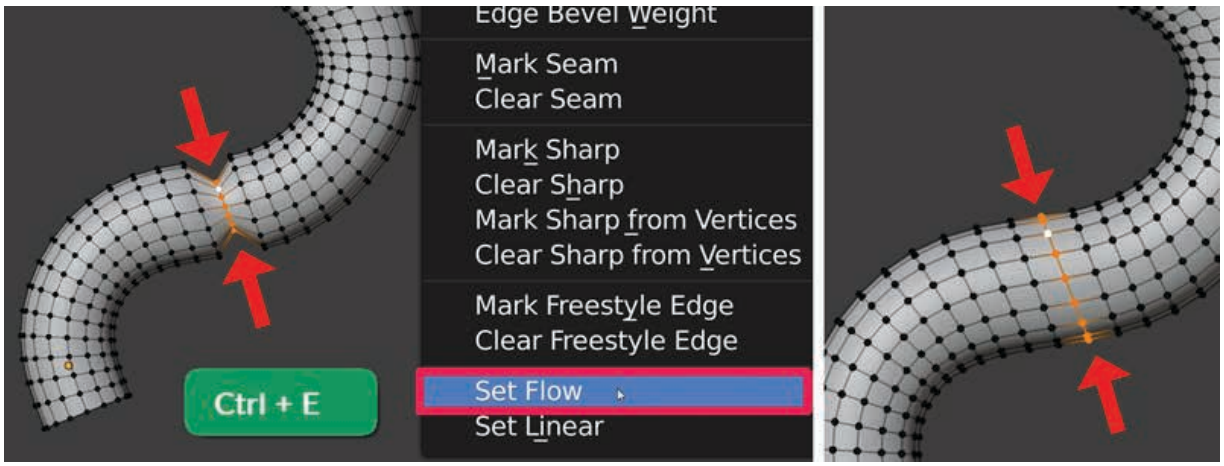


Edge Flow add-on

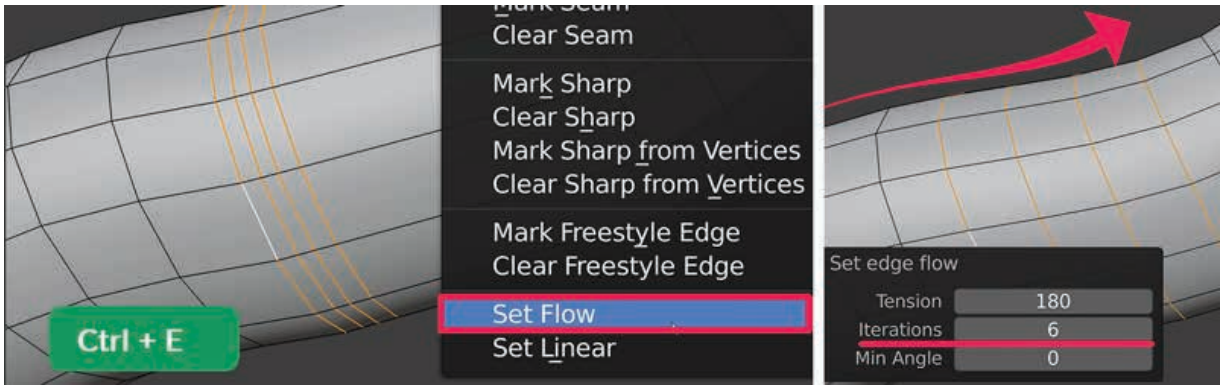
Edge Flow is a free add-on that you can find on Github and install as a zip file in Preferences.



Once activated, it can be found i**n Edit Mode** at the bottom of the Edge menu which you can open with **Ctrl+E**. This is how it works - for example you select an edge loop use the Edge Flow tool and it will make the edge loop follow the flow of the other edge loops.

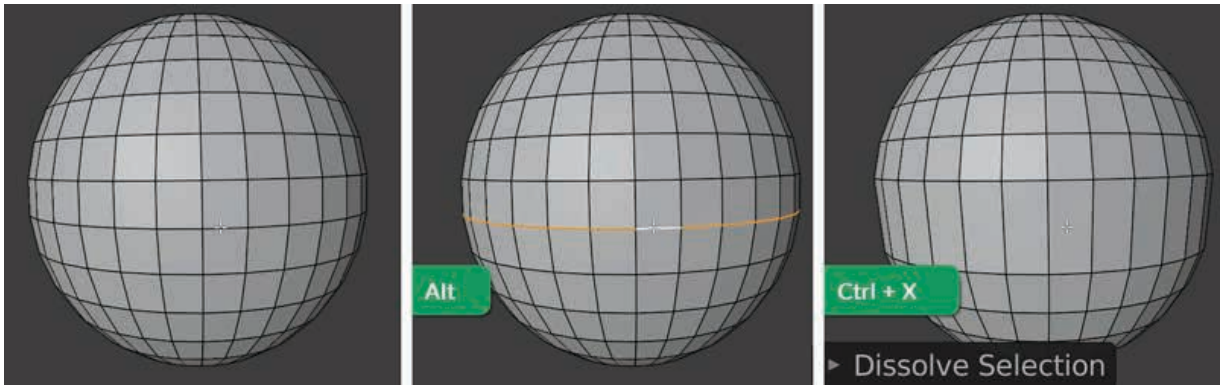


If you have edge loops that need to be distributed more evenly like these instead of **pressing G twice** and sliding them each individually you can use the Edge Flow tool here as well to do this all with one click.

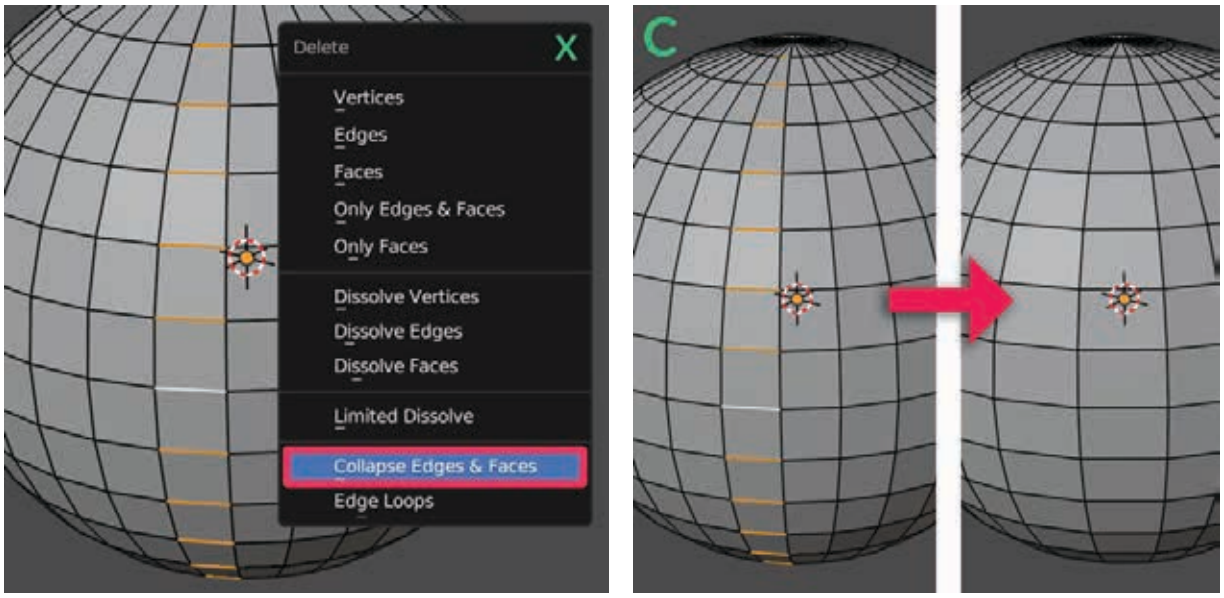


2 ways to Dissolve Edges

There's a couple of ways to remove edges without removing any faces. First, you can select edges or edge loops by holding Alt and left-clicking. Then **press Ctrl+X**. This dissolves the selected edges.



You can also **hold Ctrl+Alt and left-click** on an edge to select a row instead of a loop of edges. Then **press X and C** to collapse that row of edges.

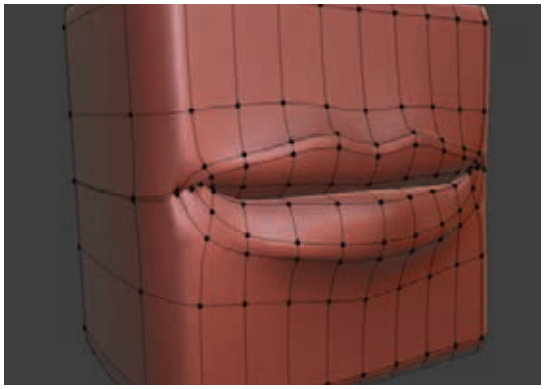


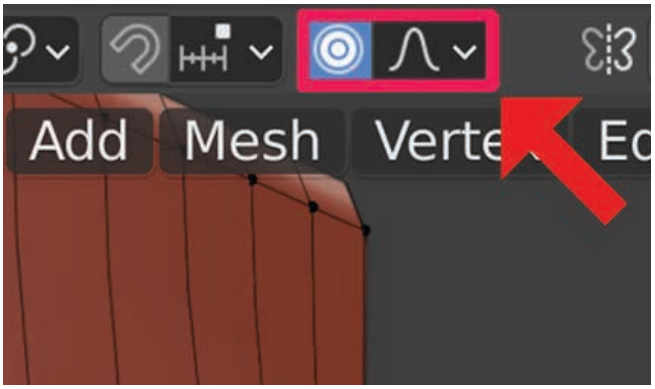
Please note, that if you have "Emulate 3-button mouse" enabled in Preferences under Input, these shortcuts are different.

	Default	**Emulate 3-button mouse**
Select Edge Loop	Alt + LMB	double-click LMB
Select Edge Row	Ctrl + Alt + LMB	Ctrl + double-click LMB

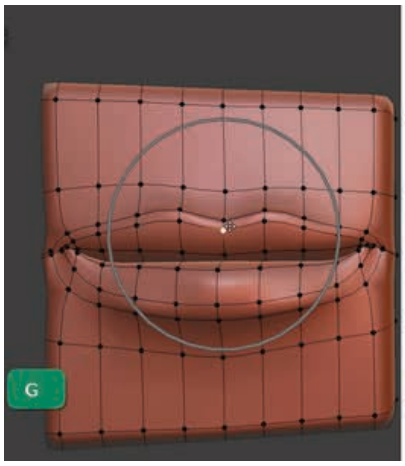
Connected Proportional Editing

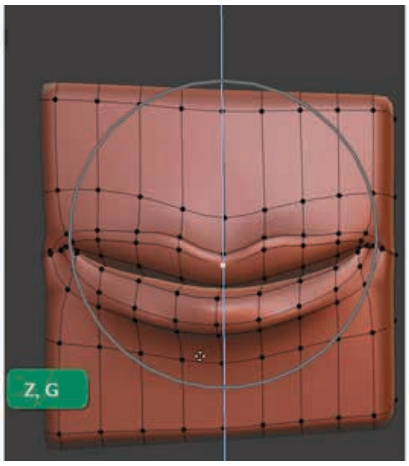
Let's say that you've modeled this terrifying default cube mouth for some reason. And you want to do some tweaking using the proportional editing tool.

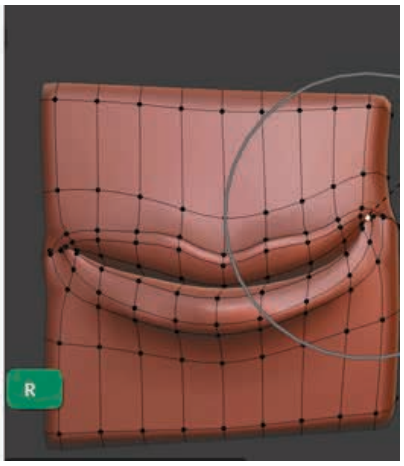




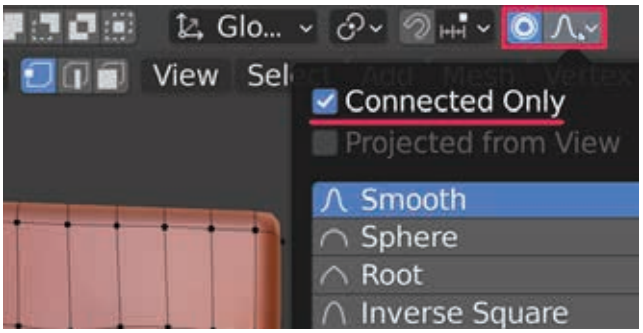
Right now it moves both the top and bottom part, although these are in fact two separate mesh islands. The top part and the bottom part.





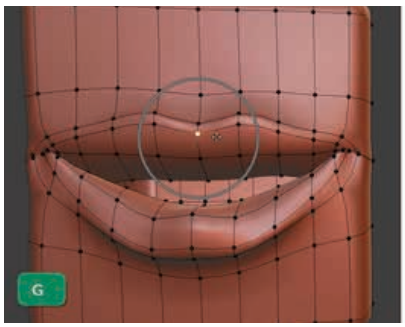


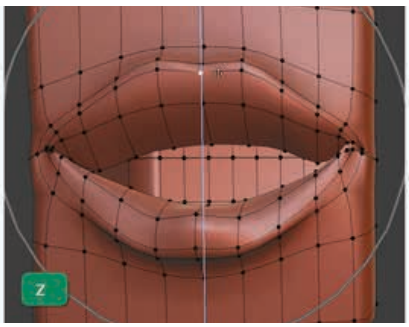
This means that we can take advantage of the "**Connected Only**" feature in the Proportional Editing Menu.

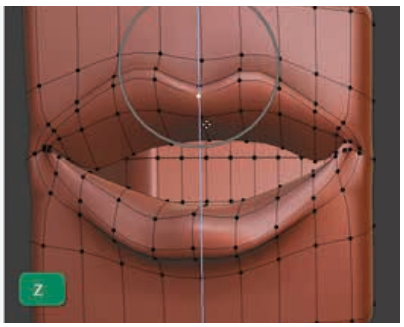


This makes sure only the connected vertices are affected by proportional editing.

This can come in handy when you want to edit one part organically, without affecting the other.

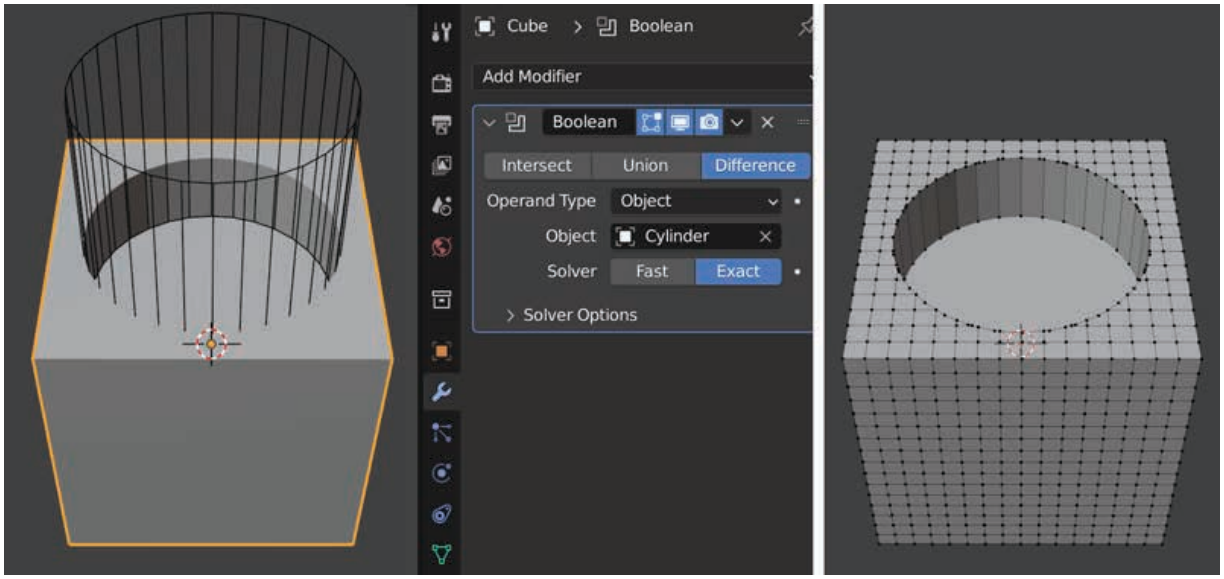




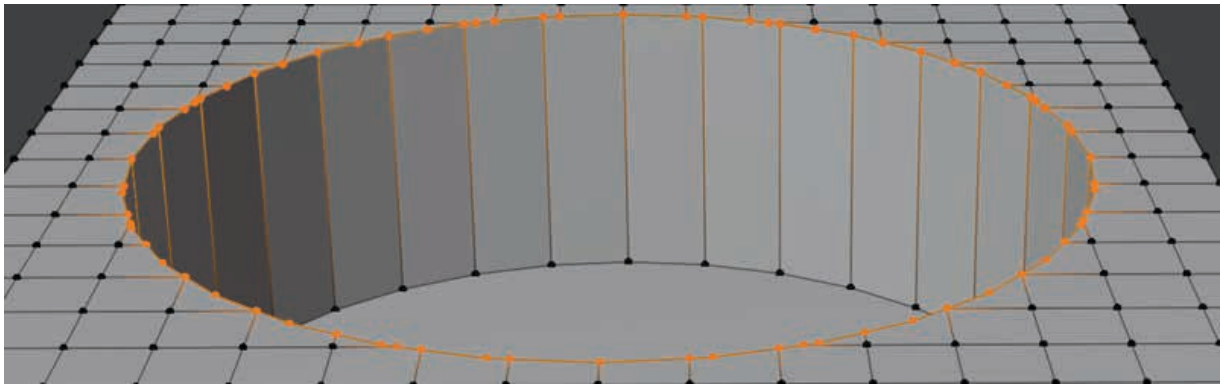


The Weld Modifier

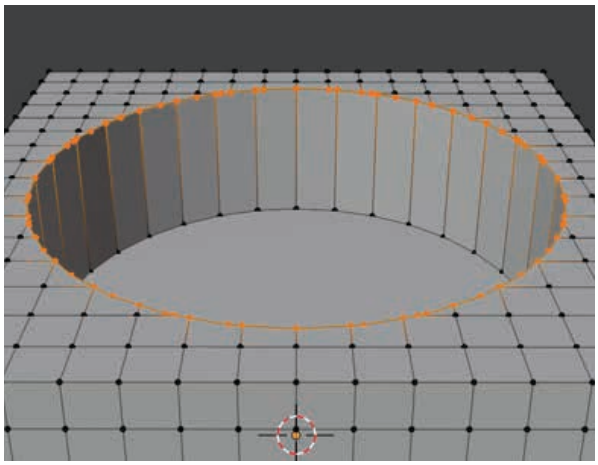
The **Weld modifier** comes in handy when cleaning up unnecessary vertices left over from Boolean cutting.

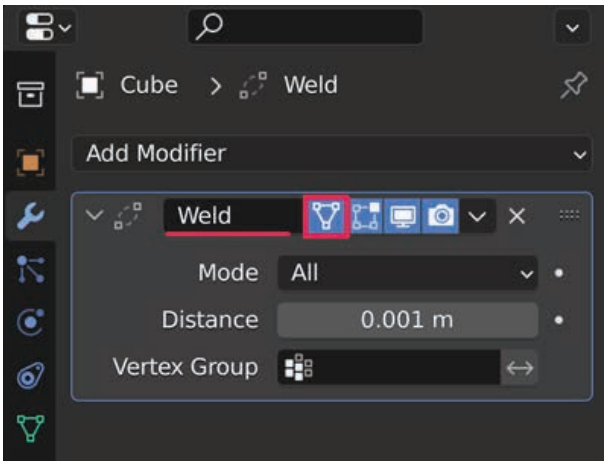


As you can see a lot of stray vertices are created from the boolean operation.



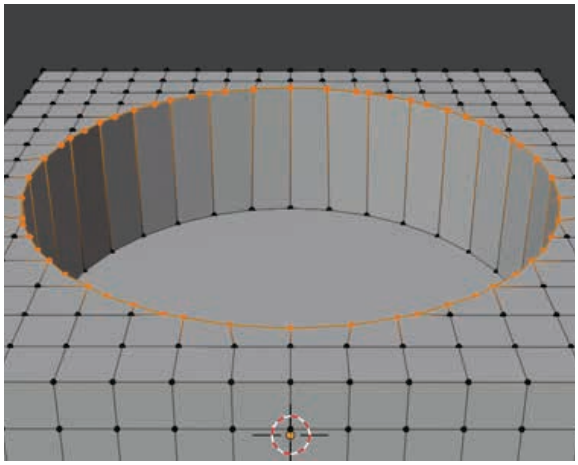
Simply add the Weld modifier and turn on the "**On Cage**" button so you can see the result in Edit Mode.

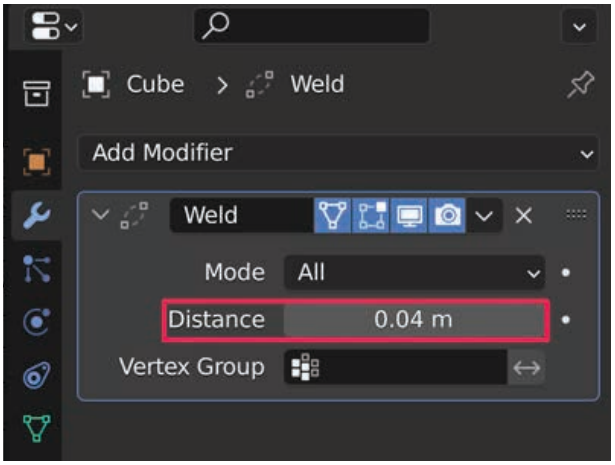




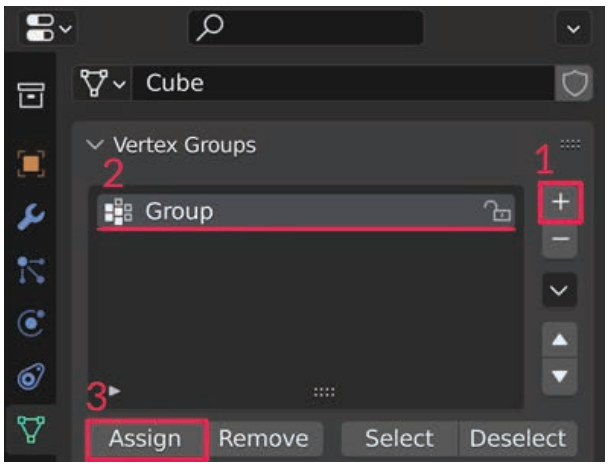
Click here to return to the index.

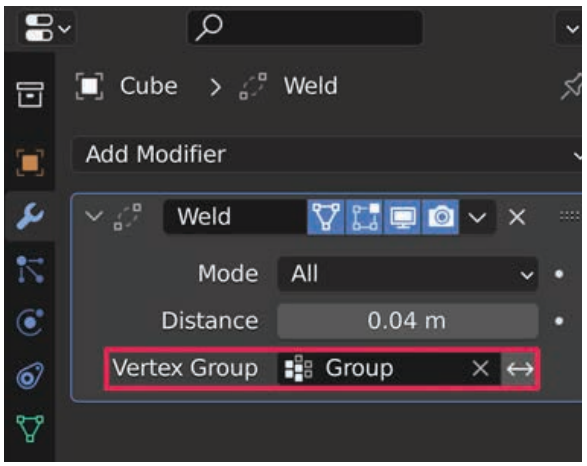
Then increase the value until you get something that works.





If you want to limit the modifier to a specific area you can also use a **Vertex Group**.

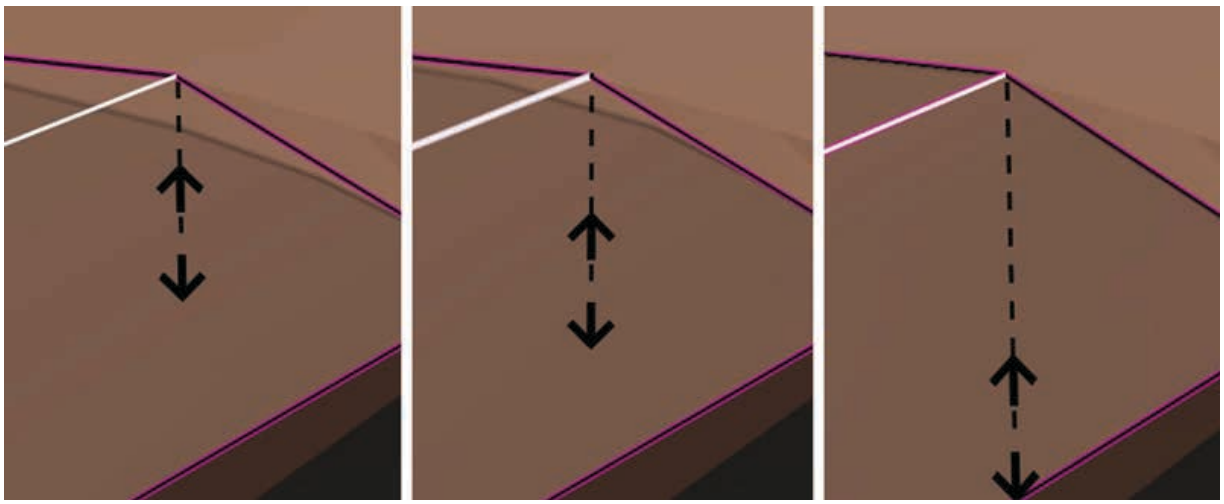




How to use Creasing

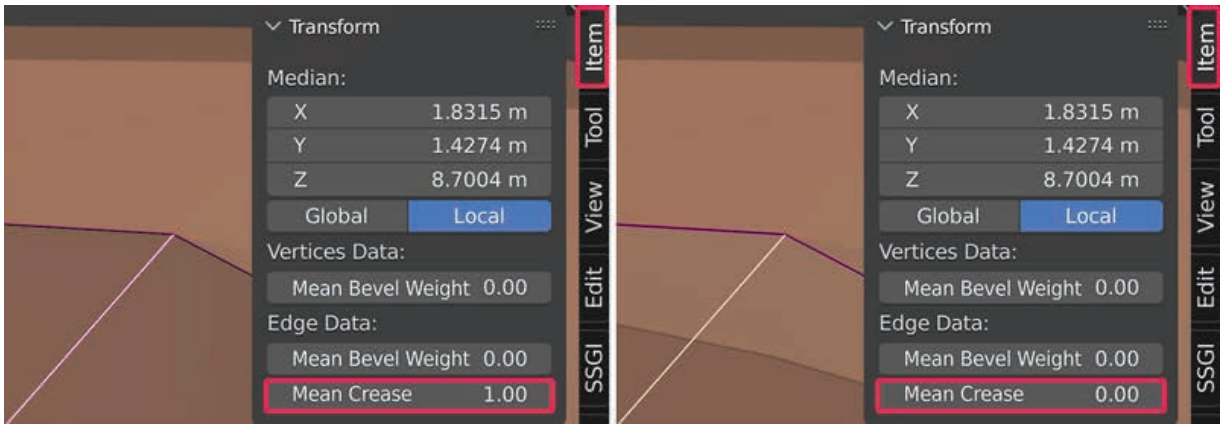
With a smooth, subdivided object like this, you can make edges sharper in a non-destructive way using Creasing.

Select an edge in Edit Mode and **press Shift+E**, then **move the mouse** to control the creasing amount. The benefit of this method is that you can still change this later by pressing Shift+E again and dragging the mouse.

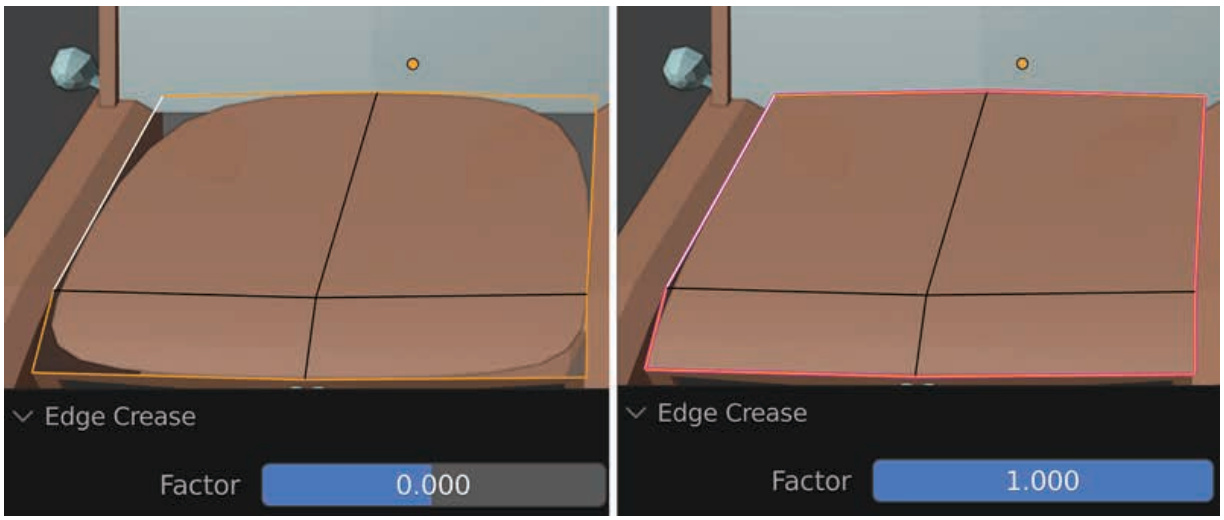


If you quickly want to do this with a keyboard shortcut, it's **Shift+E+1 on the numpad**. To remove the creasing **Press Shift+E+[-1 on the numpad]**.

Alternatively you can open the option panel by **pressing N** and change the **Mean Crease** value between 0 and 1.



Creasing is also useful to make the borders of an object sharp so that they are not affected by the subdivision modifier.

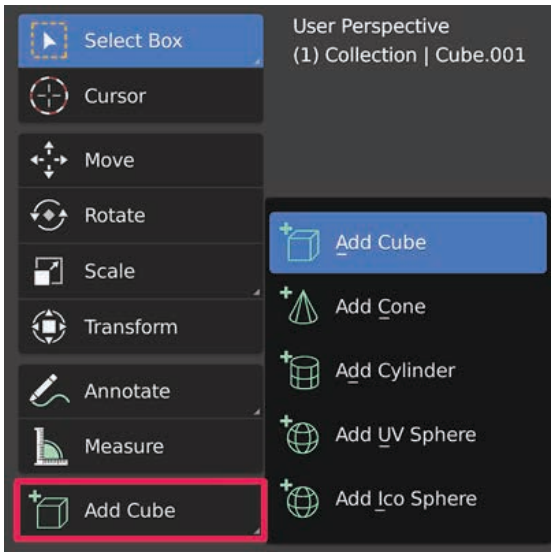


Interactive Primitives

Blender 2.9.2 introduced a new way to interactively add primitives. Primitives are objects such as the default cube, the cone, cylinder, UV sphere and so on.

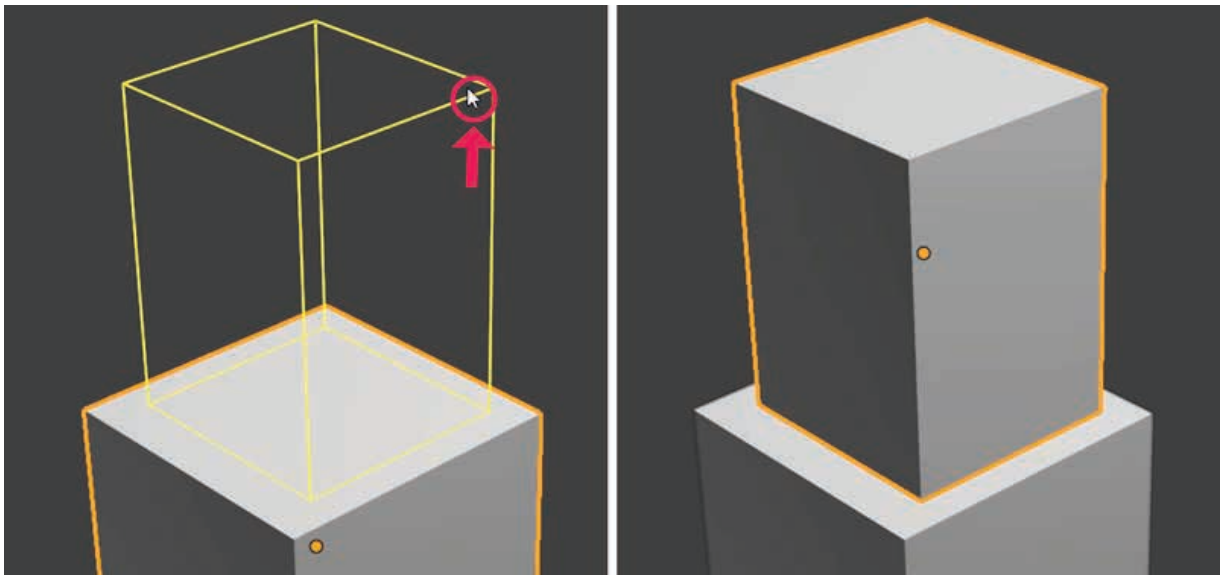
To use it, enable the **Add primitive tool** by clicking on it in the Tool panel.

If you **click and hold**, you can choose between the usual primitive shapes.



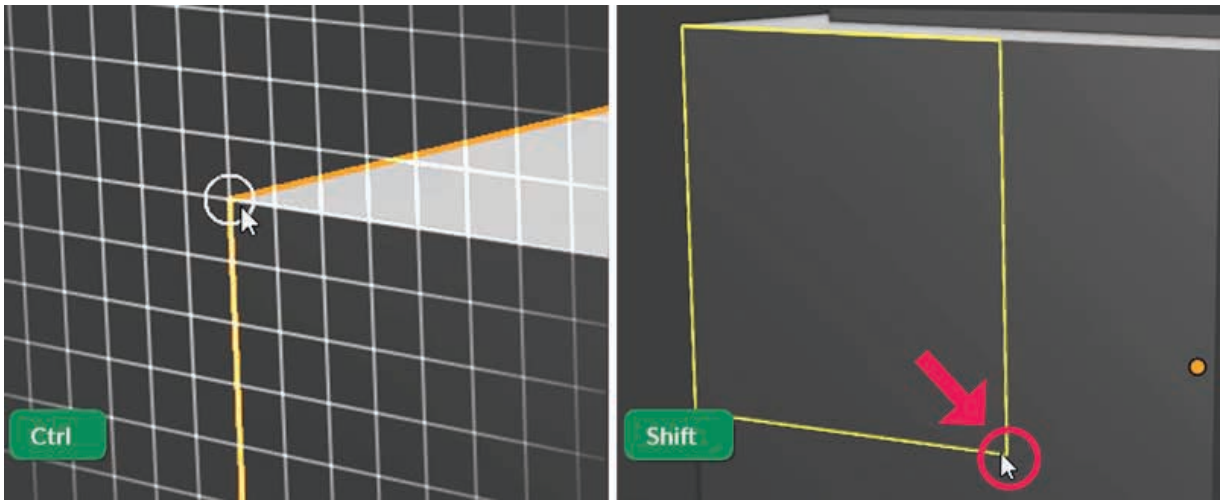
Select an object and left click, drag it in two dimensions...

then let go of the mouse button to drag it in the third dimension, and finally click again to confirm.

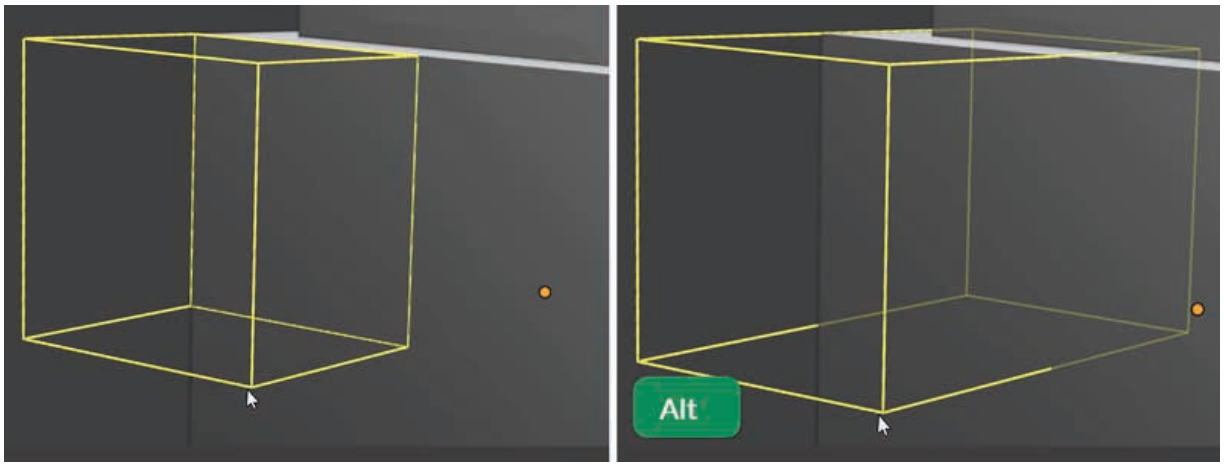


Click here to return to the index.

If you **hold Ctrl**, you can snap to the corners and edges of the selected object. **Holding Shift and Ctrl** keeps the original aspect of the primitive.

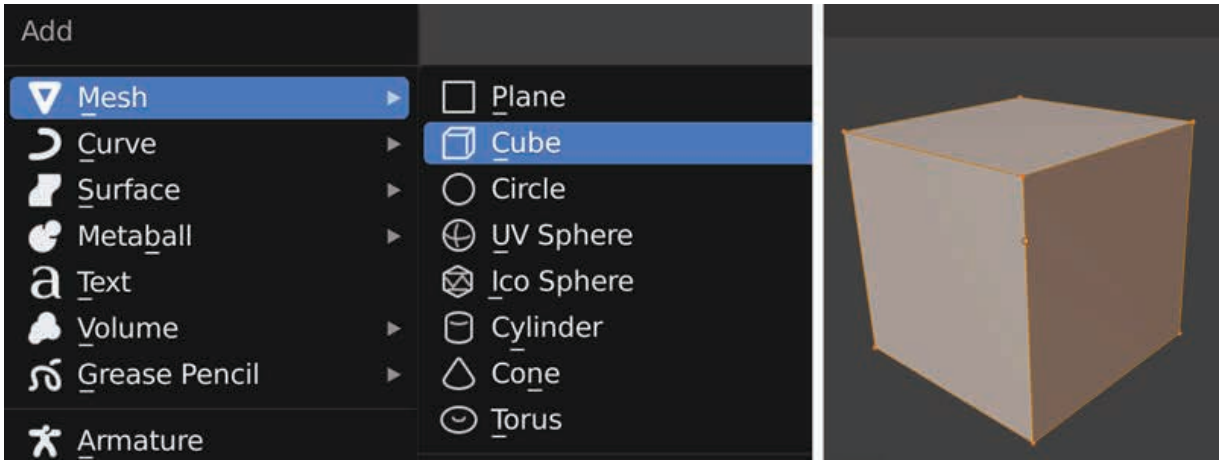


If you **hold Alt** after clicking to create a primitive, it will use that original point where you clicked as the centre.

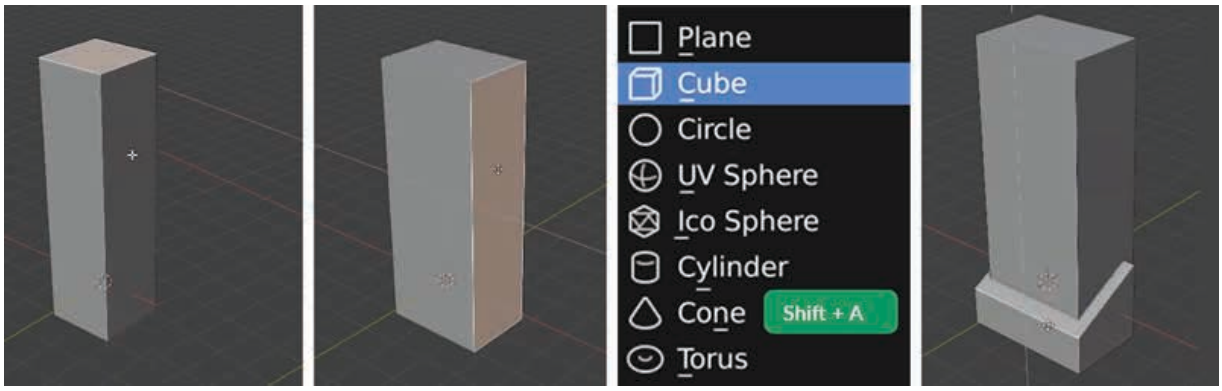


Modeling with Primitives

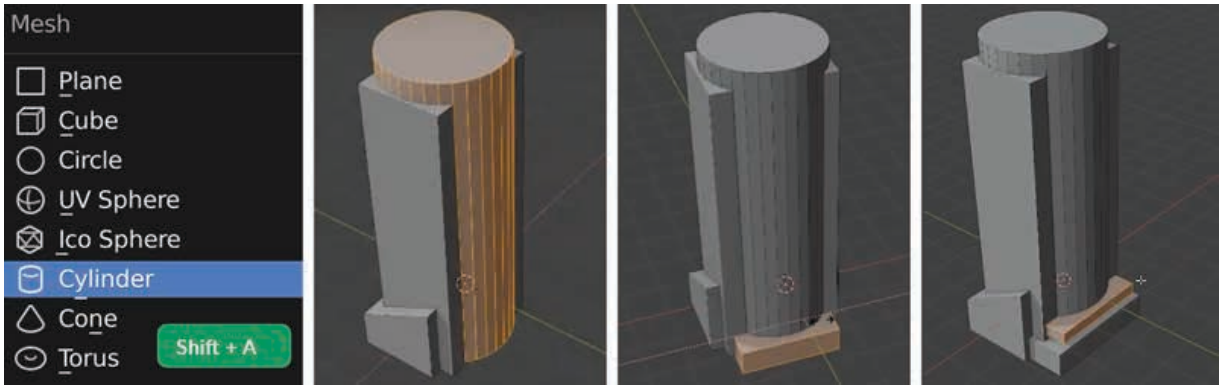
Blender comes with a bunch of primitives that can be accessed by **pressing Shift+A** and going to the **Mesh** menu.



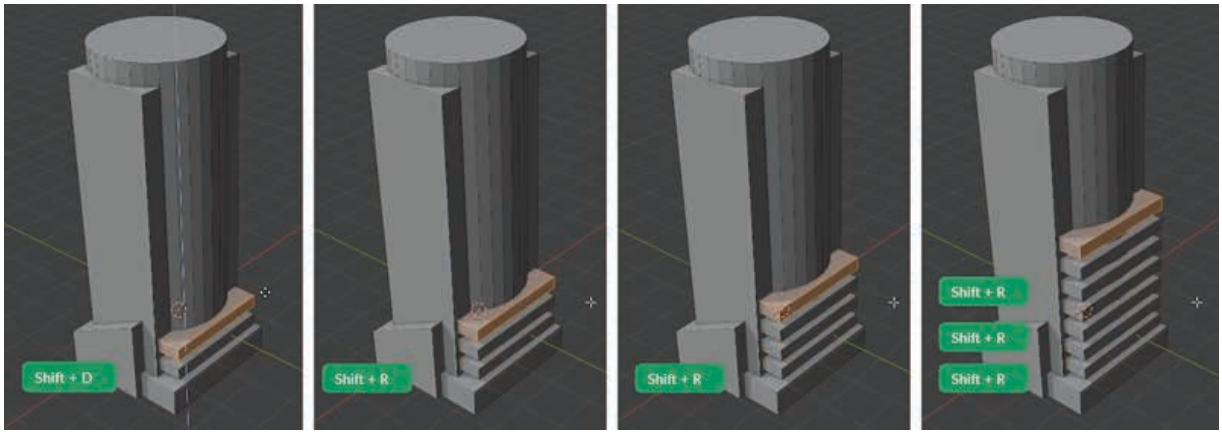
Here I move some faces of a cube in **Edit Mode**. You can also move edges to create angular shapes.



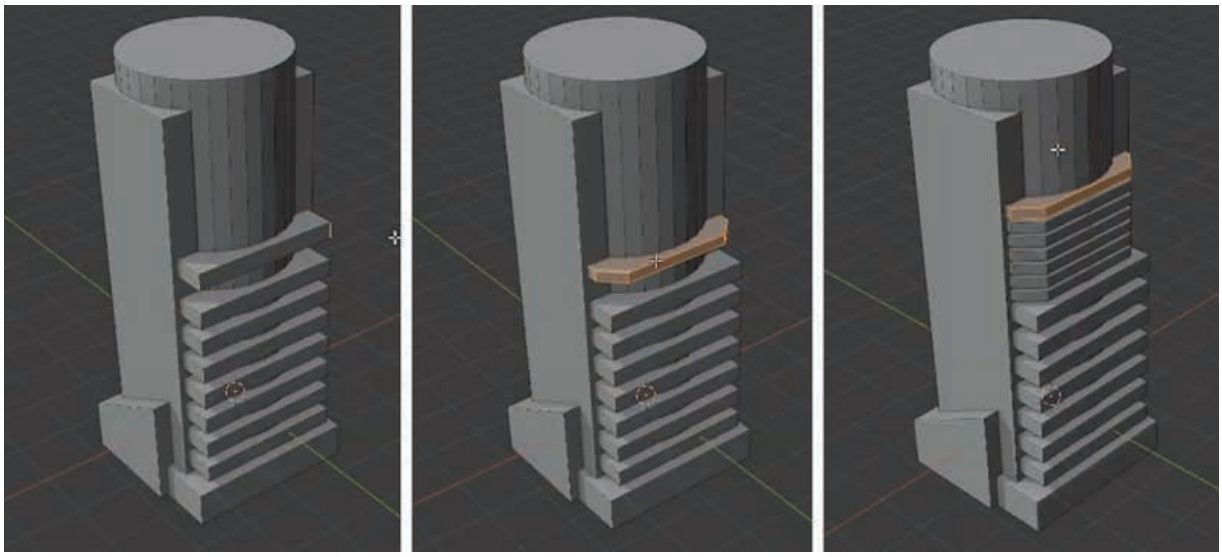
Just using **Cubes** and **Cylinders** can get you pretty far in quickly blocking out shapes.



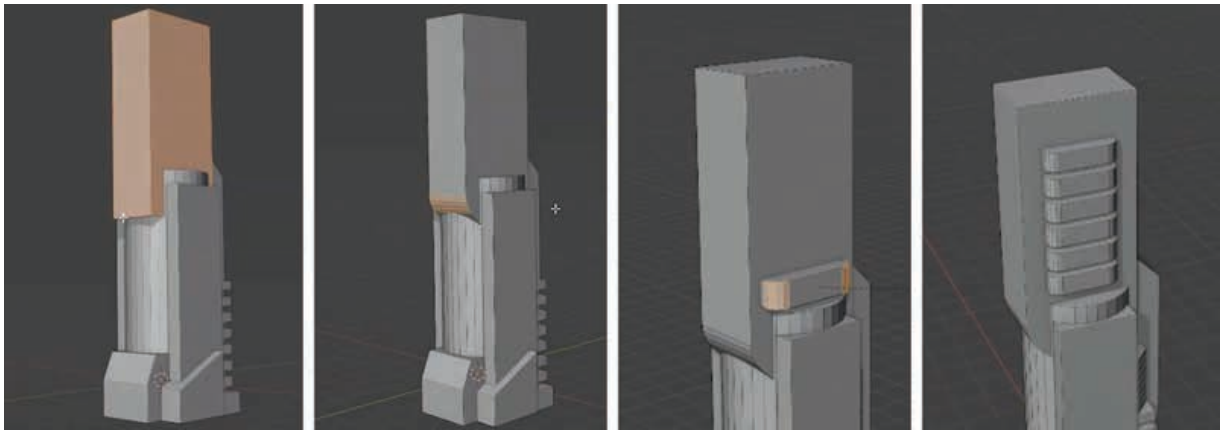
Here I'm duplicating a cube with **Shift+D and Z** so I can move it along the Z-axis as part of the duplication action. Then, by **pressing Shift+R** I can repeat those exact steps.



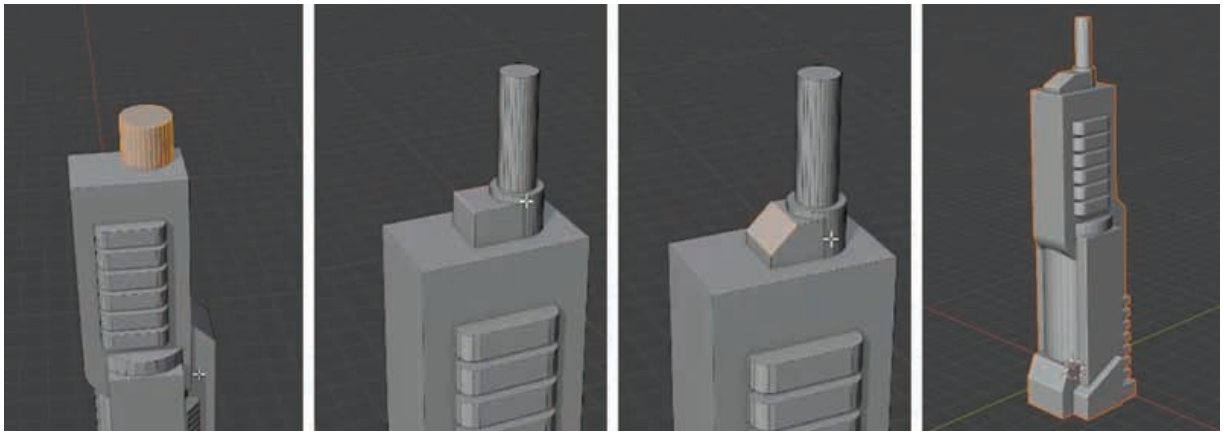
Selecting some edges and beveling them with **Ctrl+B** can also create interesting shapes. Which can then by duplicated as in the previous step.



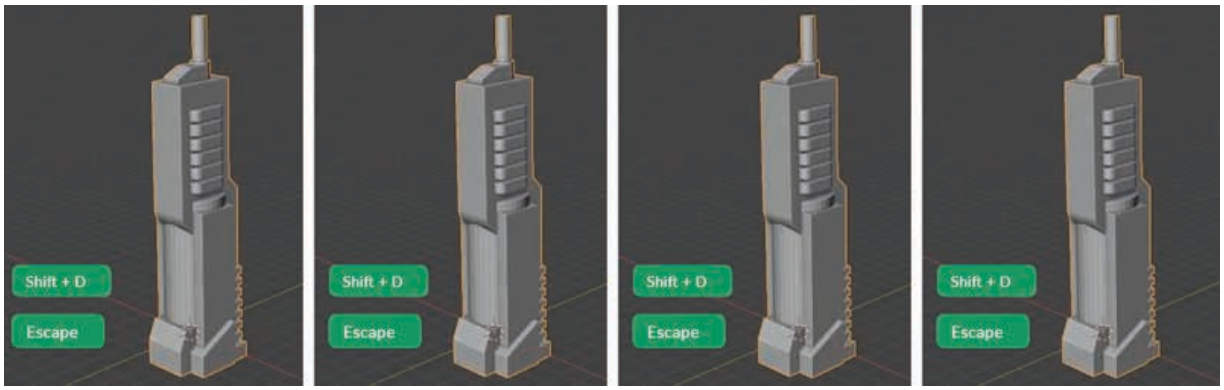
By **scrolling the mouse wheel up** during a bevel action, you can make the bevel more rounded by adding more segments to it.



As long as you stay in Edit Mode and add more primitives it remains a single object.



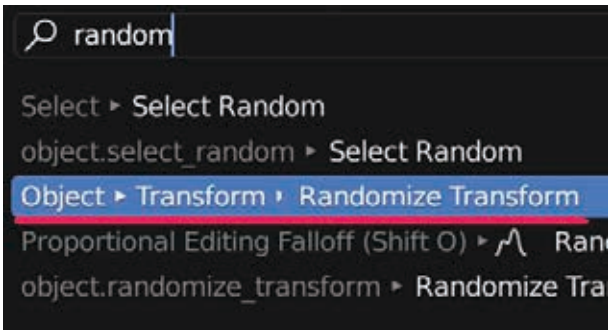
To create a city, go to Object Mode and duplicate the object with **Shift+D**, and then repeat that by **pressing Shift+R** a few times so you get a few duplicates. You don't need to move the duplicates, we'll do that in the next step.



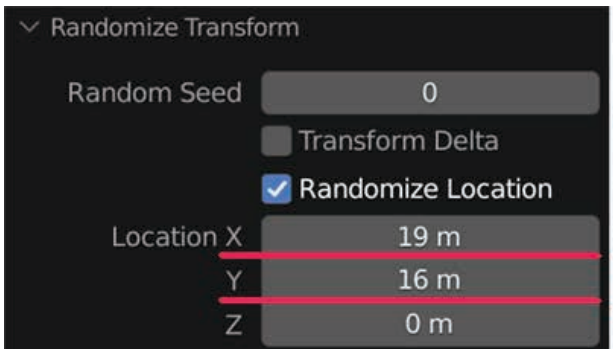
Select them all by **pressing A.**

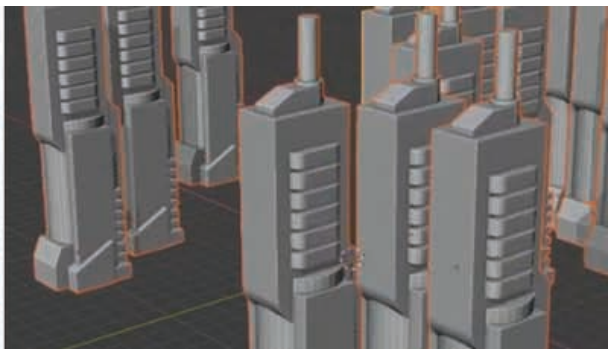
Press F3 to search for **Randomize Transform**.

Or go to Object > Transform > Randomize Transform.

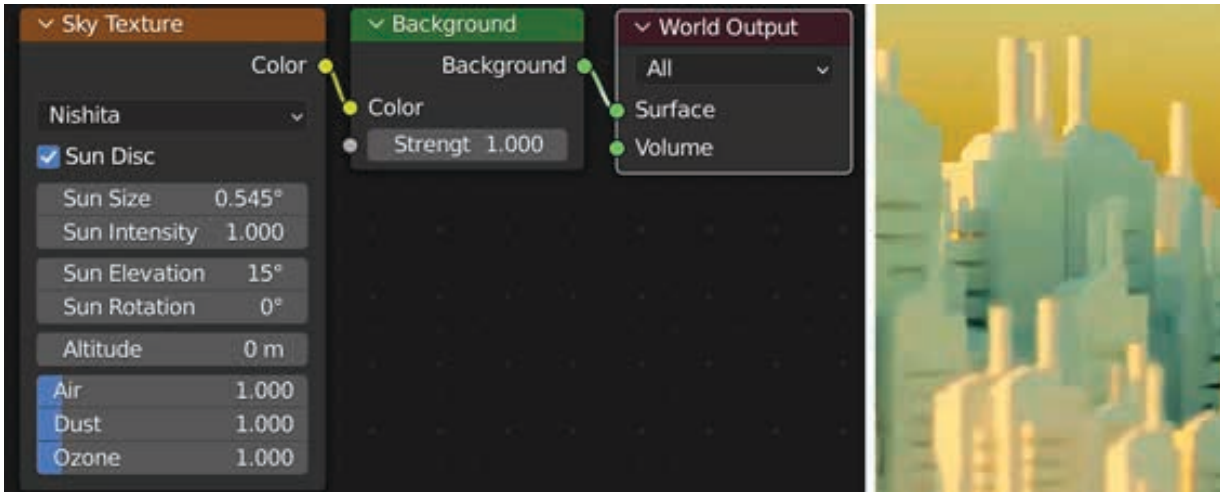


Use this to create some variation in location and scale.





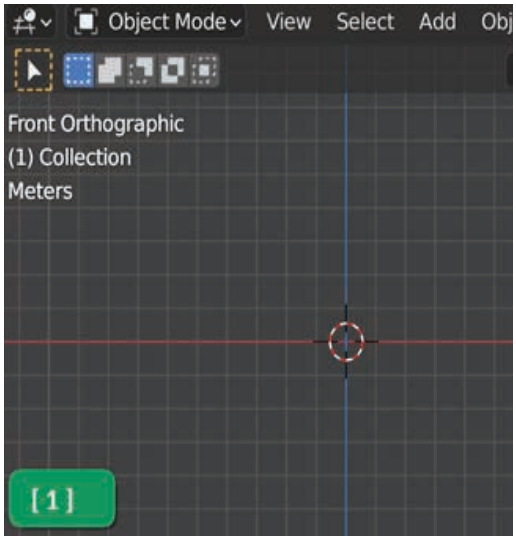
Using Blender version 2.9 or later you can use the **Sky Texture** to light the environment.

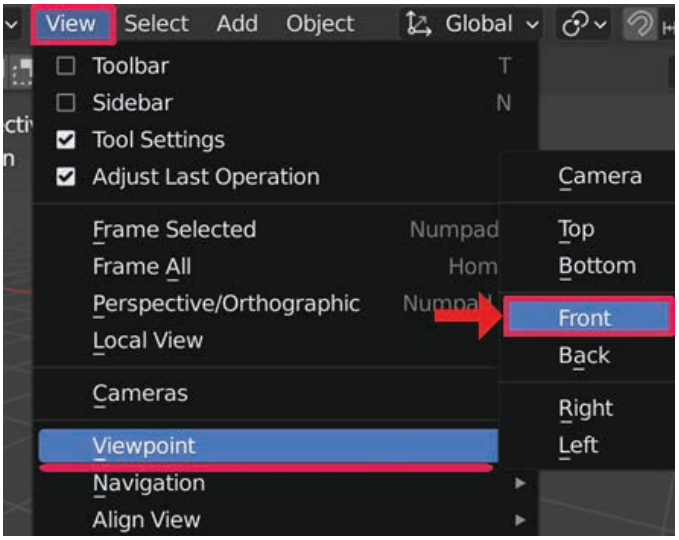


Finally, use Box Mapping to add a photographic image texture on the objects, and you're all done.

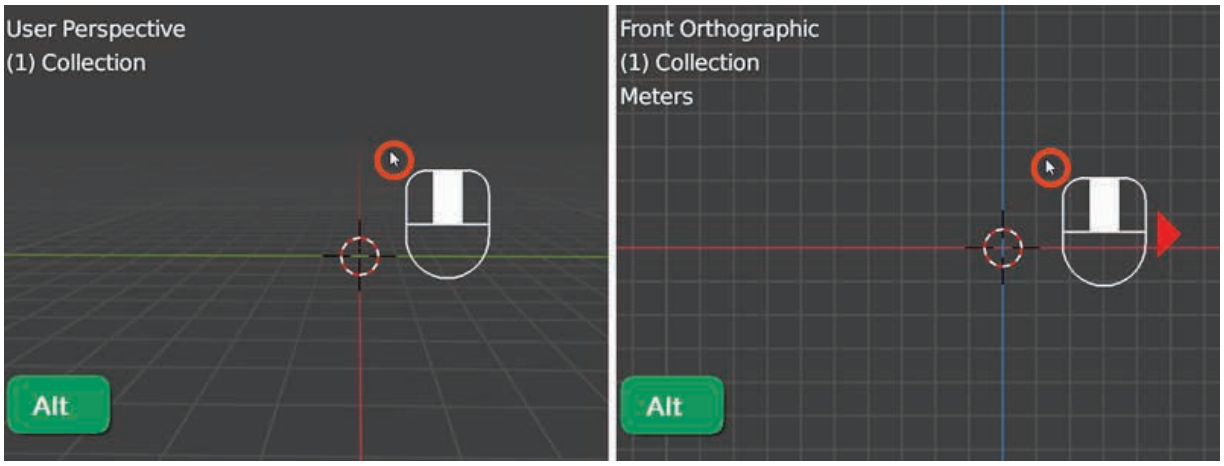
Car Modeling Basics part 1 - Reference

Press 1 on the Numpad to go to the front view. If you don't have a Numpad on a laptop for example you can go to View > Viewport > **Front**.

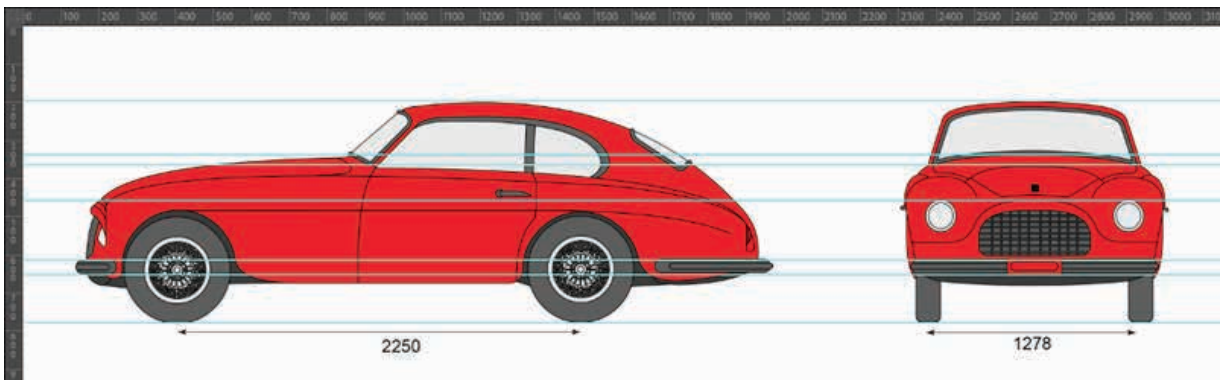




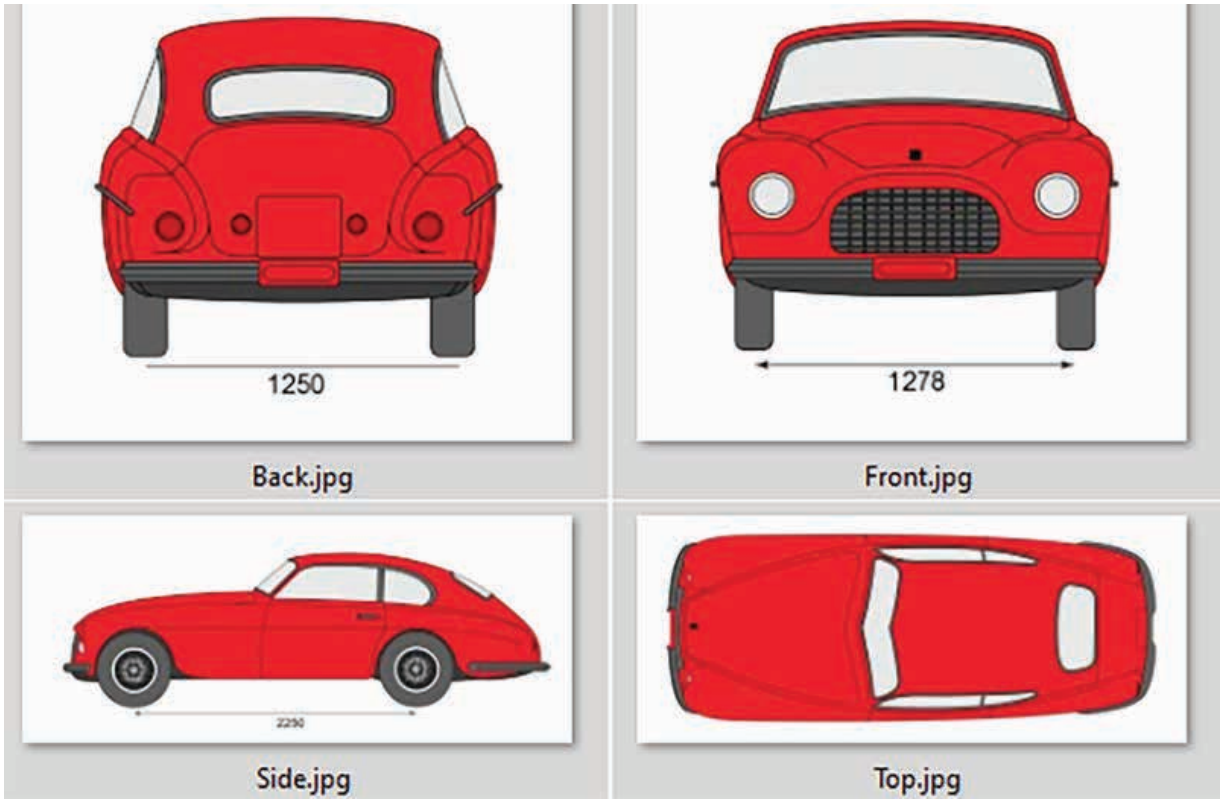
Alternatively, you can hold Alt while orbiting the viewport until it snaps into the front orthographic view.



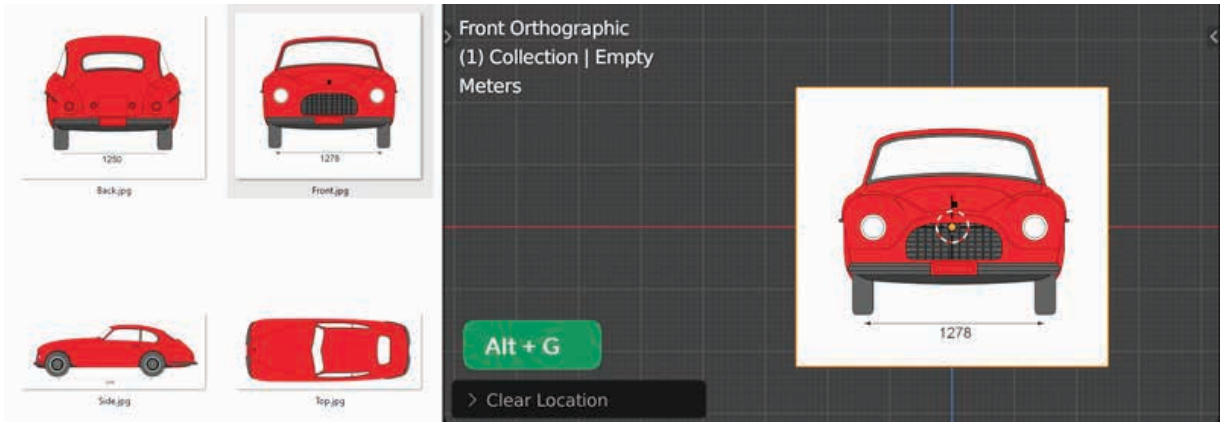
Before dropping reference images into Blender, make sure they align in the image editing software you like.



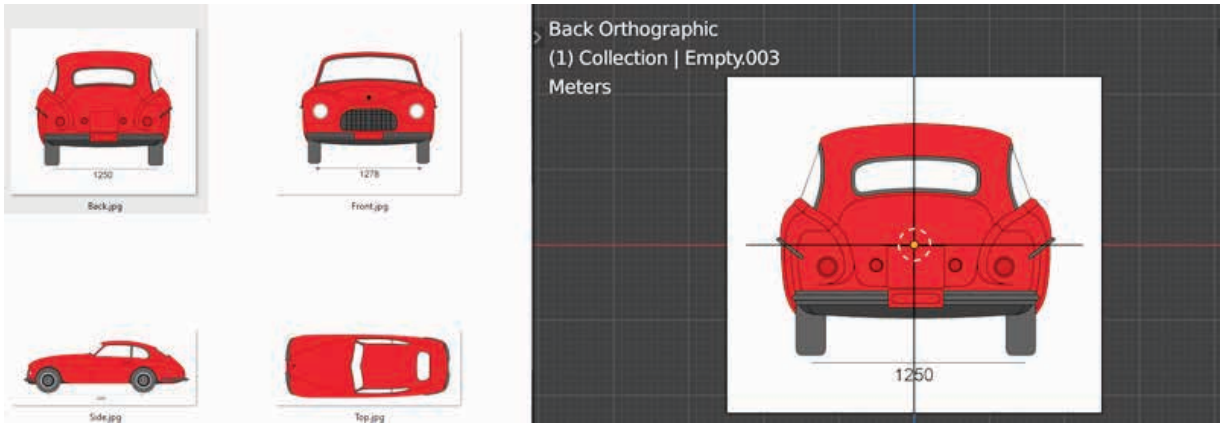
Save them as separate, clearly named images.



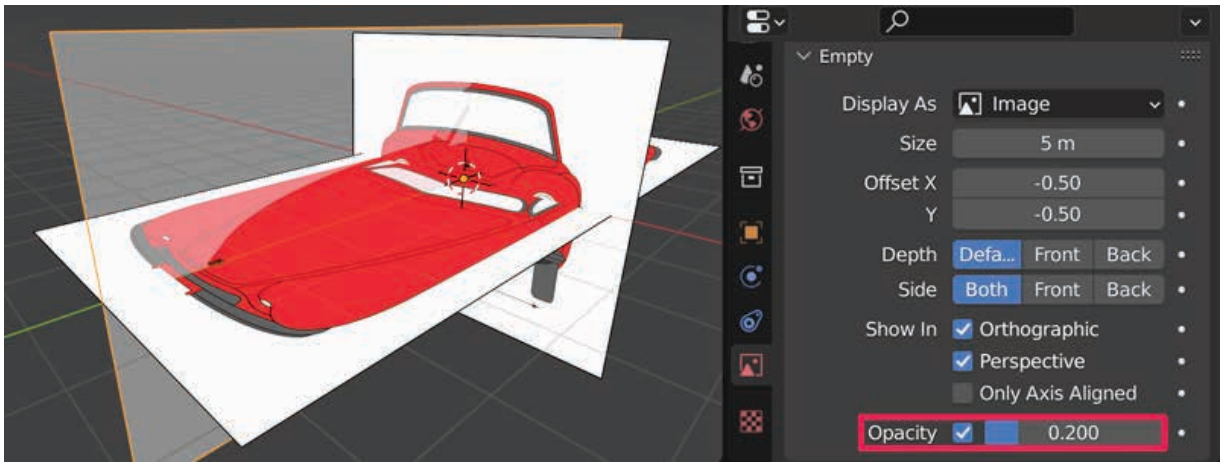
Drag the front reference image into the front viewport. **Press Alt+G** to reset its location.



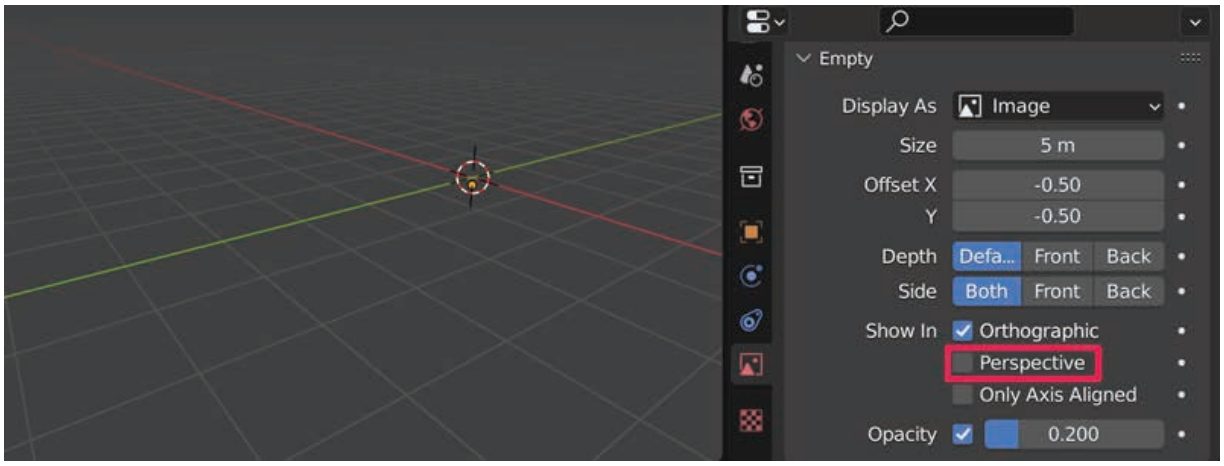
Do the same for the other views as well. To get the Back orthographic view, **press Ctrl+1** on the numpad instead of just 1 on the numpad. The more views the better.



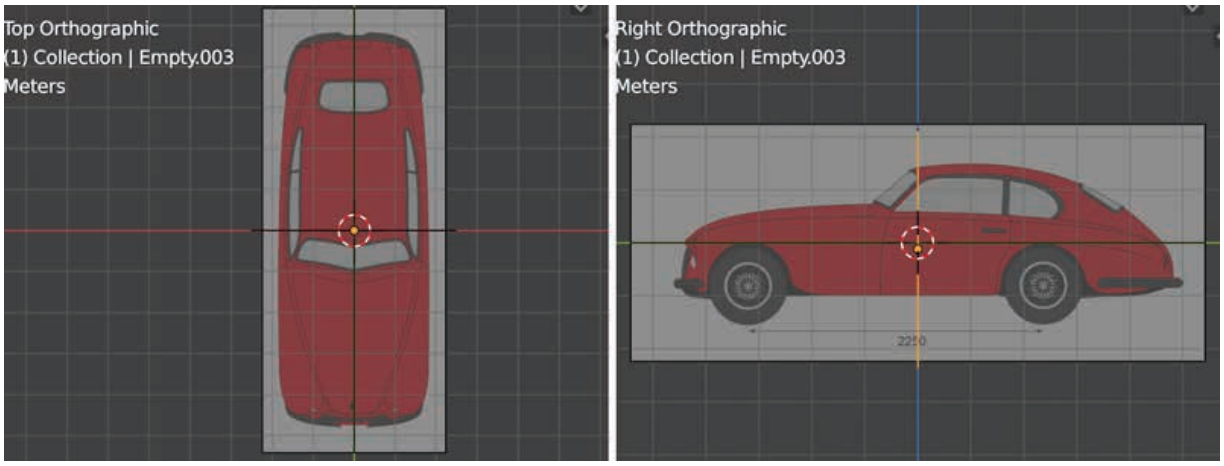
I personally like to enable **Opacity** and reduce the value so the image becomes see-through.



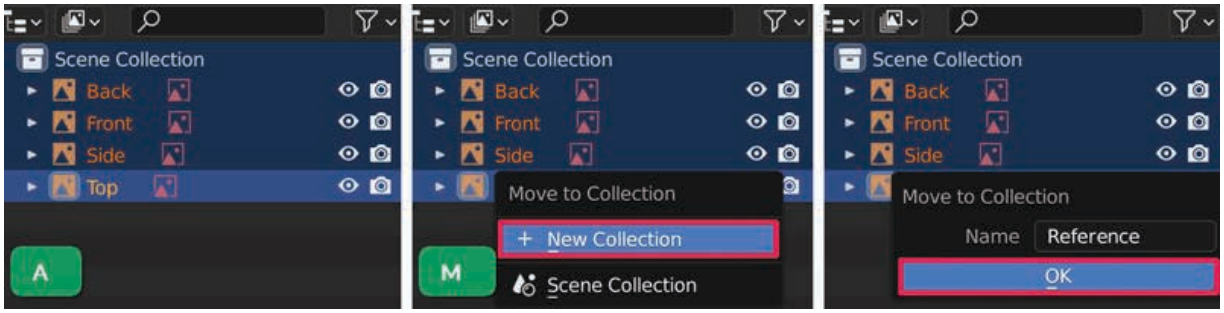
If you don't want to see the reference images in perspective view, simply uncheck **Perspective**. Then it's not visible in Perspective view.



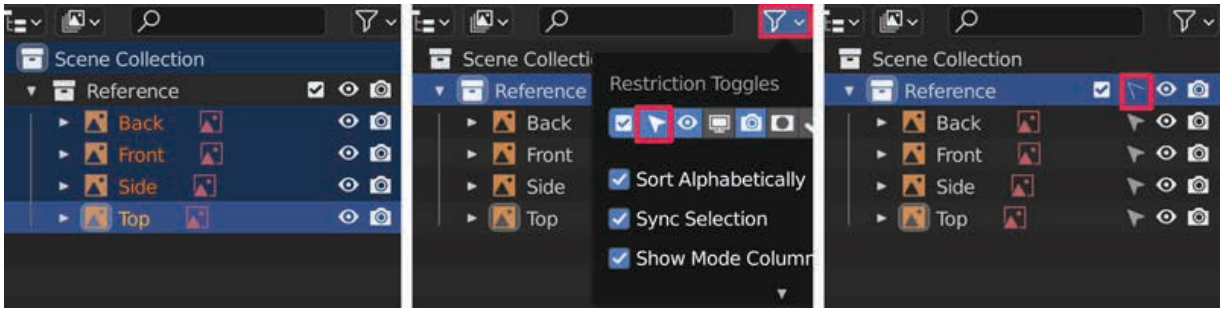
That way they are only visible in the orthographic views.



Select the reference images and **press M**, then choose **New Collection**. Click **OK**.

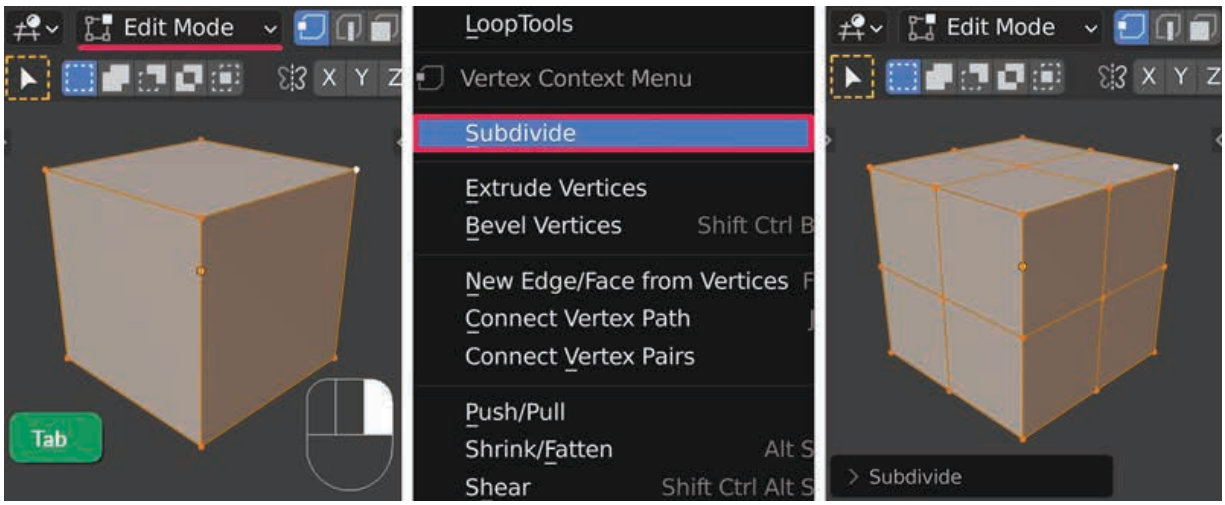


This way you can turn off the selectability so you don't keep accidentally selecting them while modeling.

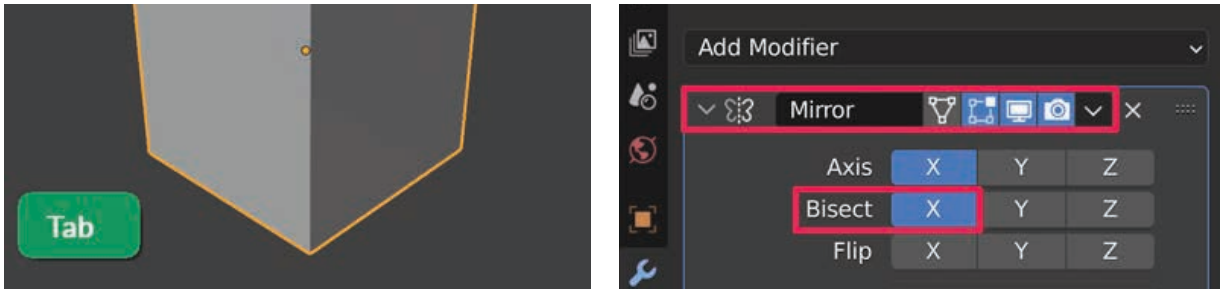


Car Modeling Basics part 2 - Mirror modifier

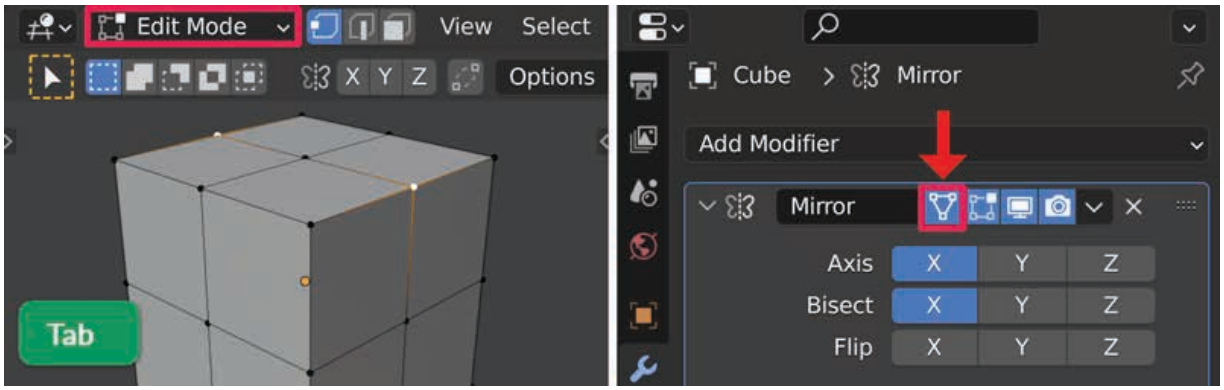
Press Tab to go into Edit Mode, then subdivide the cube once.



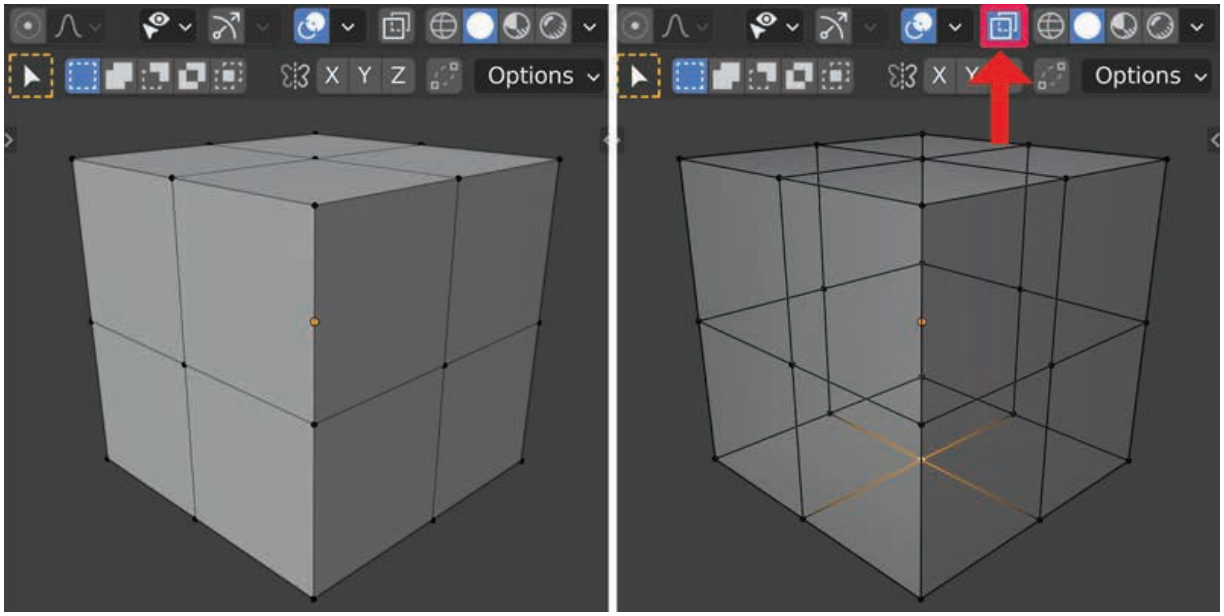
Exit the Edit Mode by **pressing Tab** and add a **Mirror modifier** to the default cube. Turn on **Bisect** so that half of the cube is removed automatically.



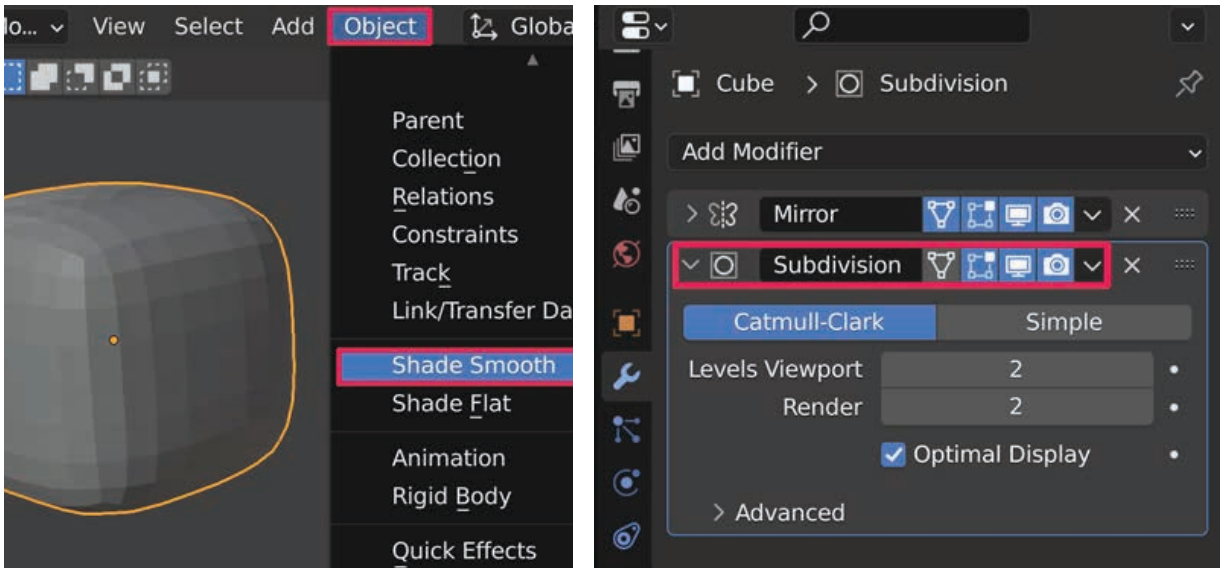
Enable **on cage** so you see the modifier result in the viewport.



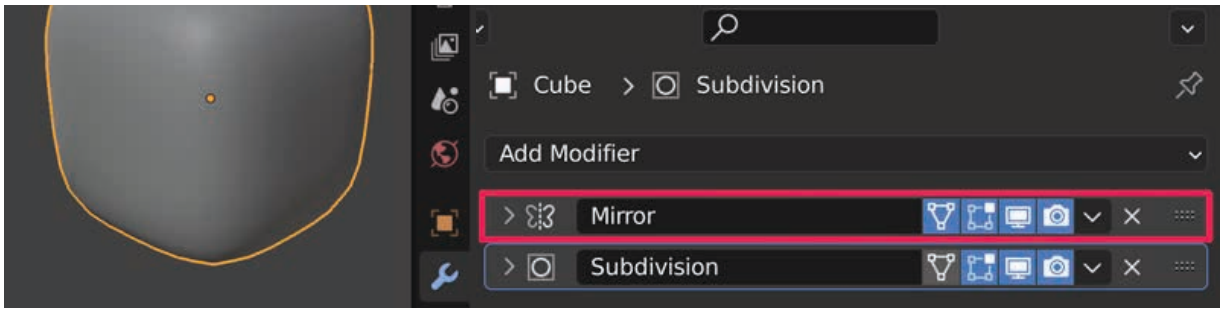
Toggle X-Ray on so that you can select vertices on the other side of the mesh as well.



Add a **Subdivision modifier** to the cube. Make sure **Smooth Shading** is turned on.

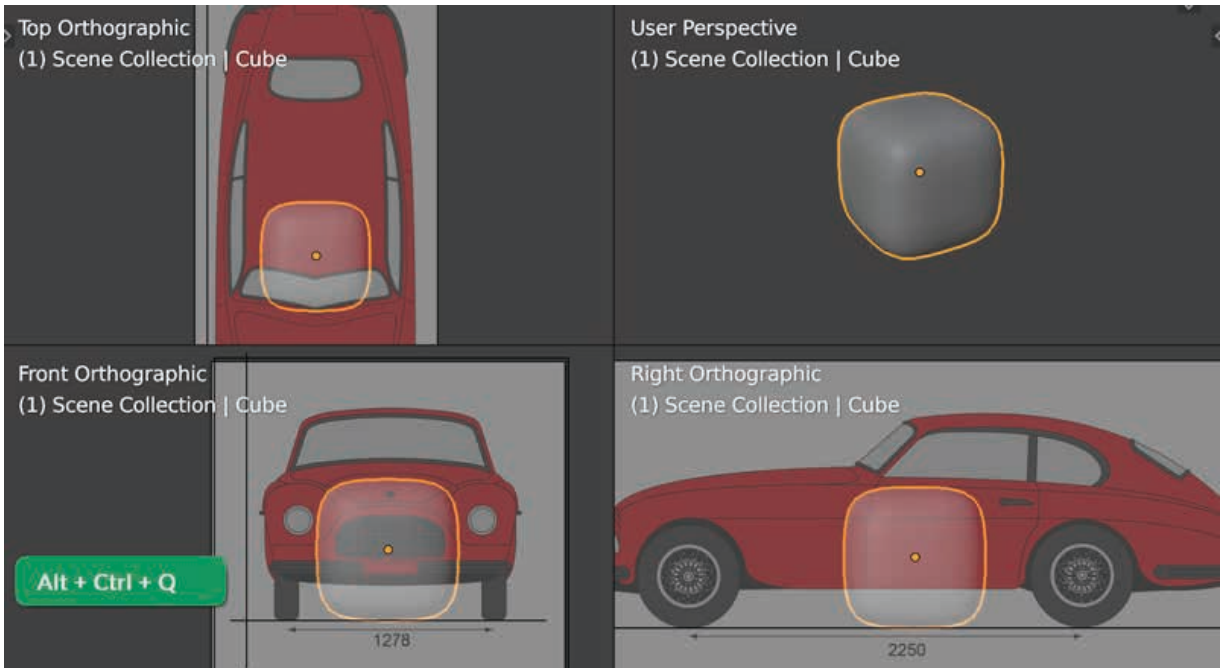


Make sure that if you add more modifiers the **Mirror modifier** is always on top for the correct result.

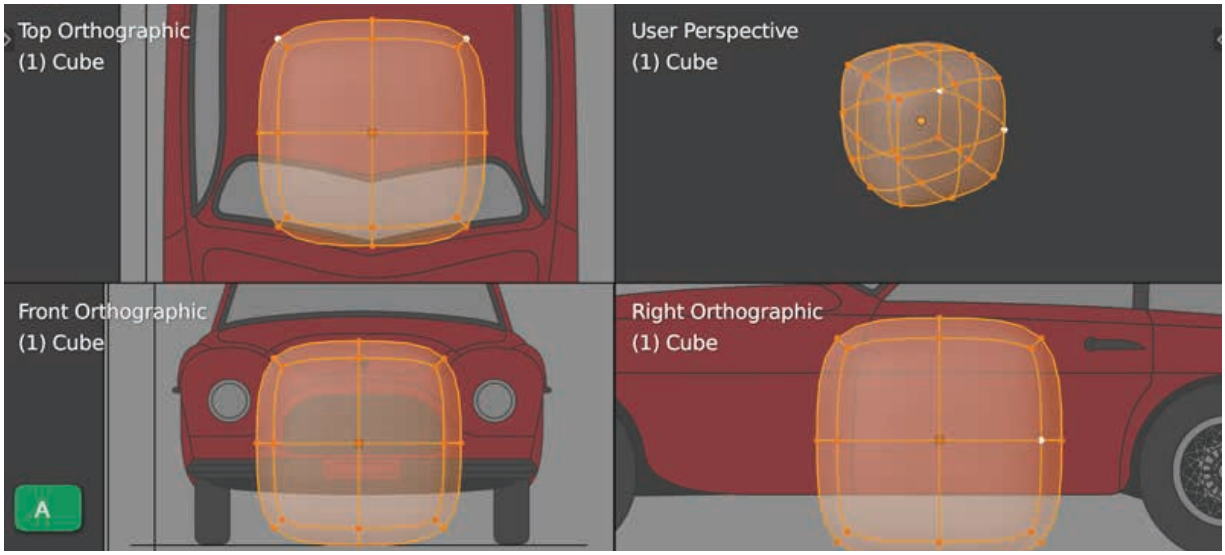


Car Modeling Basics part 3

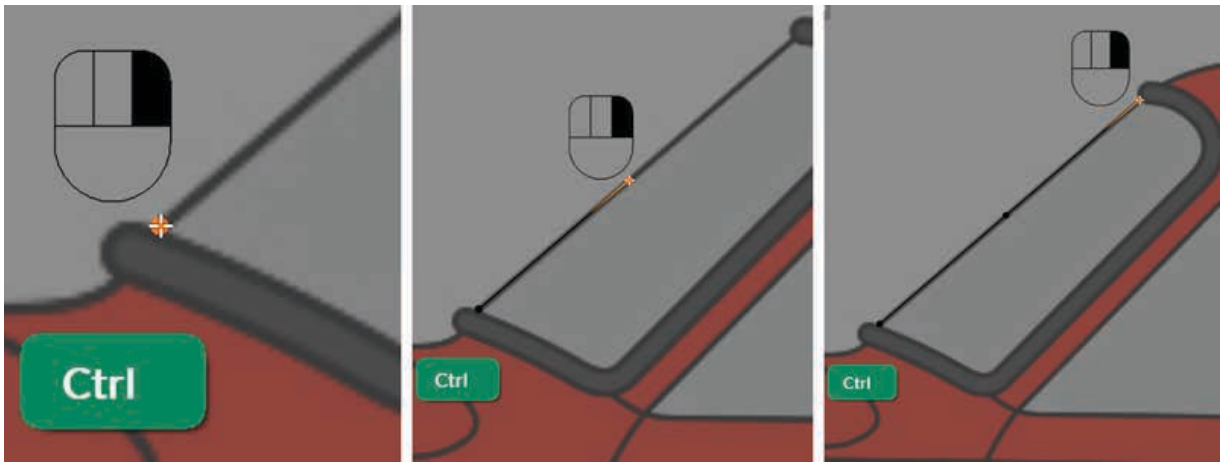
Enable Quad view by **pressing Alt+Ctrl+Q** so we can check all angles at the same time.

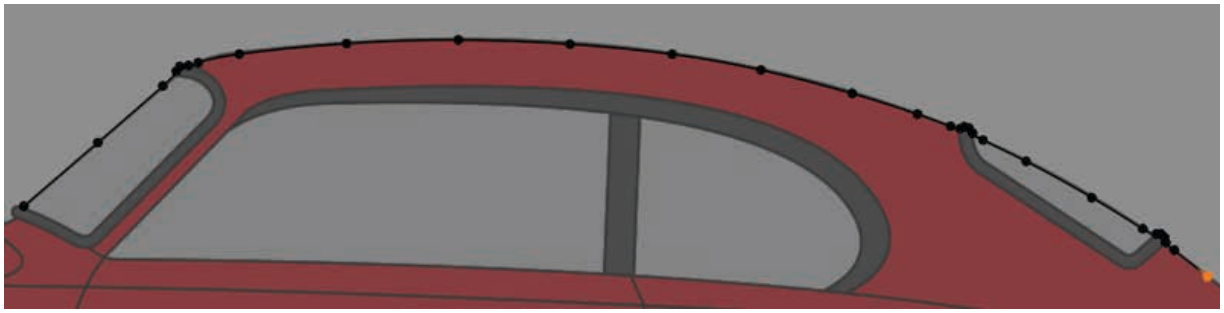


Just like with a real car, we will create this model out of separate parts. In Edit Mode select all and the delete all the faces.



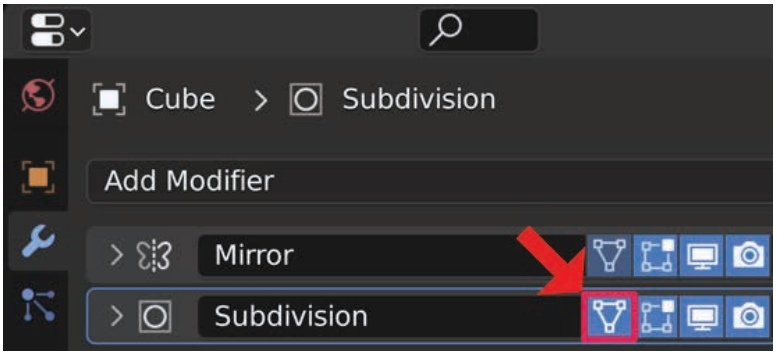
Now we can create a new vertex by **holding Ctrl and right-clicking**.

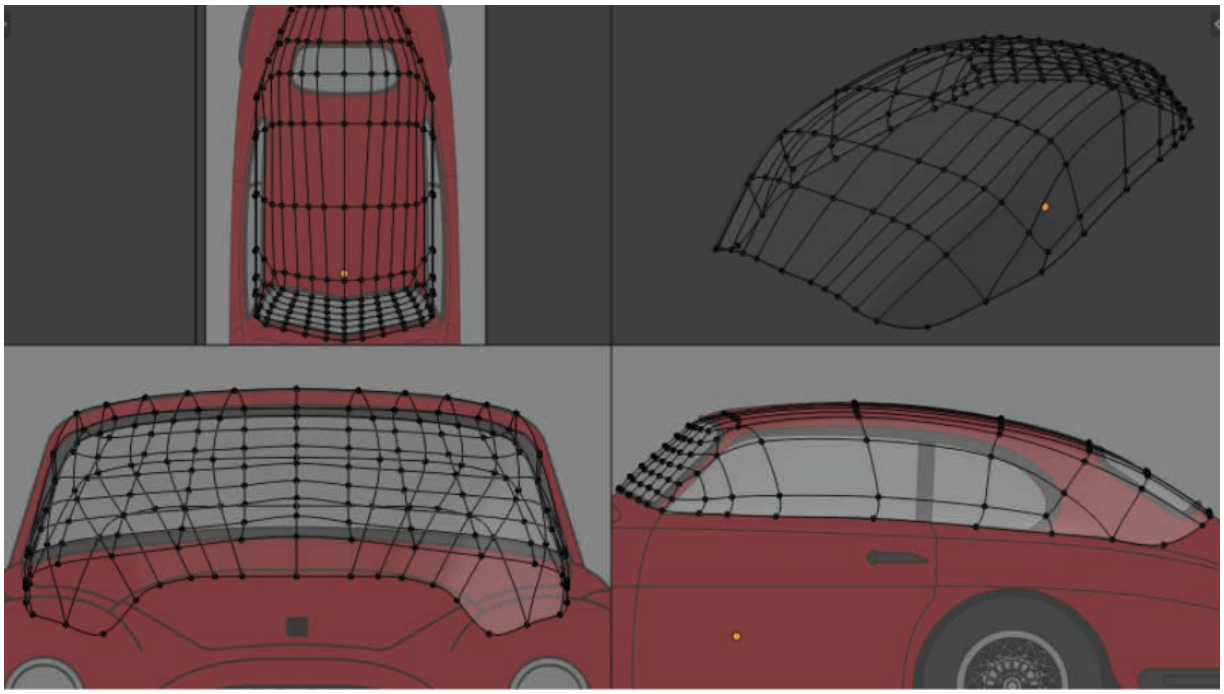




Extrude and fill faces until you've got the whole part blocked out.

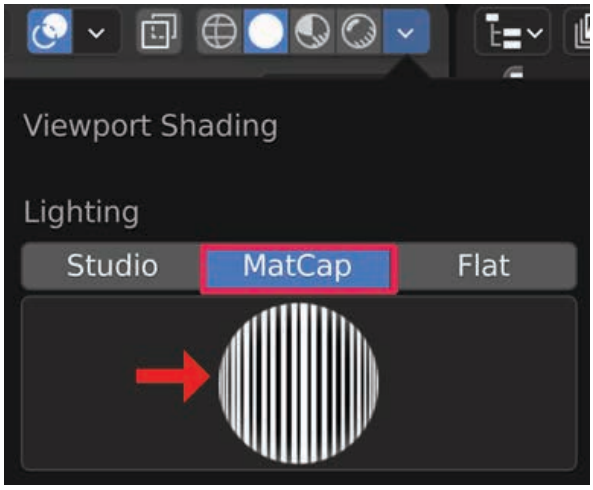
Enable **On Cage** on the Subsurf modifier and fine-tune the shape.

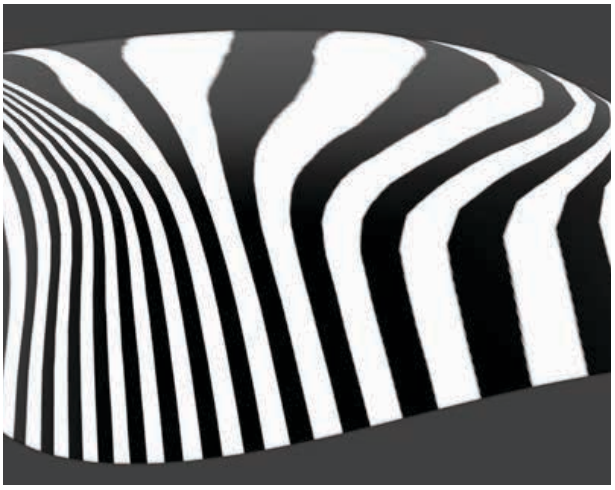




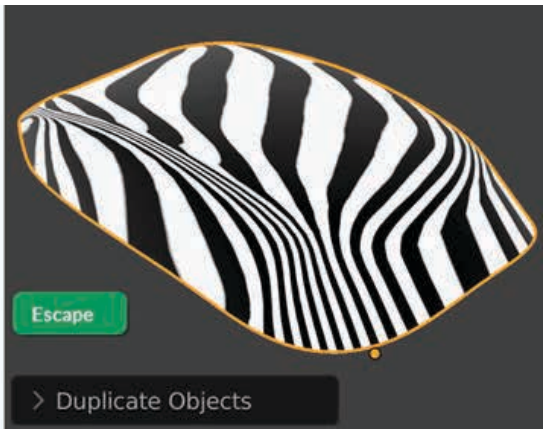
Click here to return to the index.

Use Matcaps to check the smoothness. The black and white Matcaps in particular are made for this purpose.

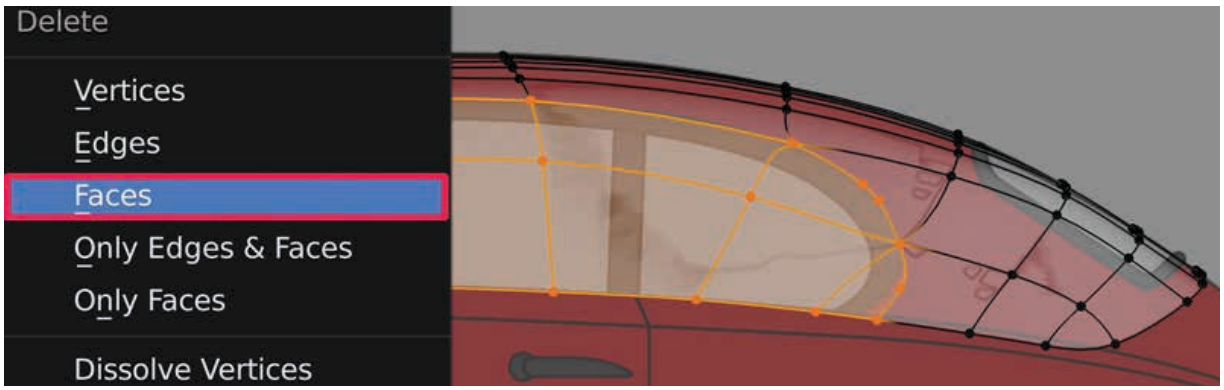




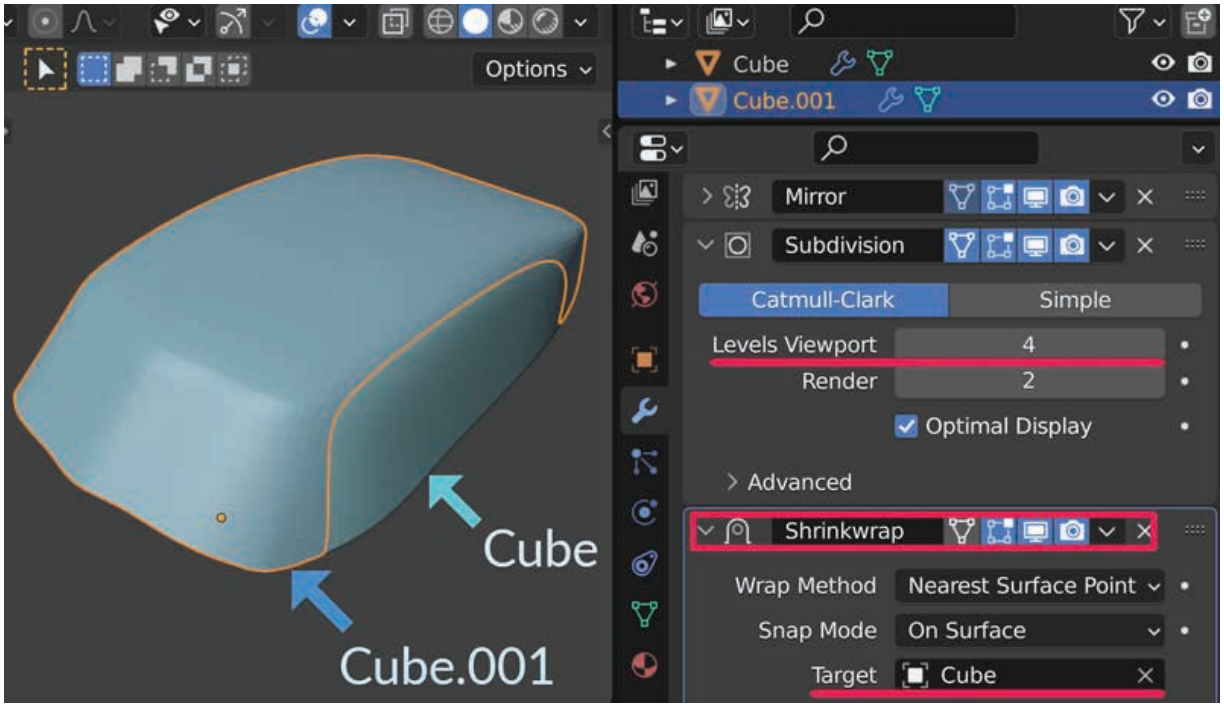
To add holes, duplicate it in object mode and create the holes in the duplicate.



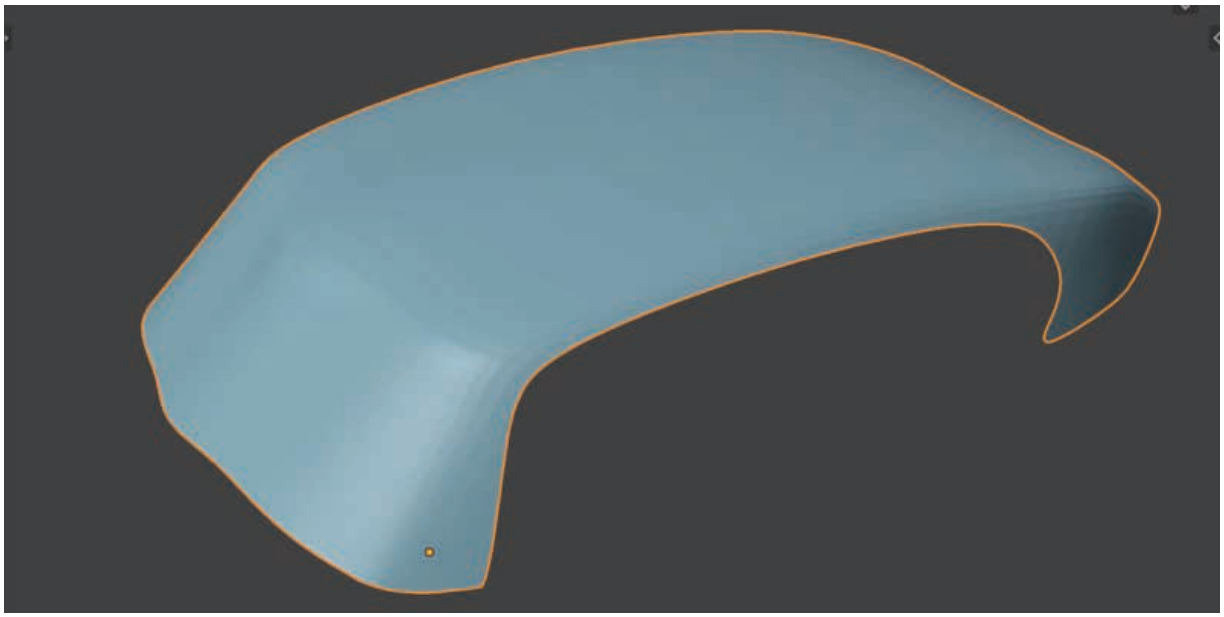
Select and delete some Faces, for example for the windows, on the duplicate object.



Increase the subdivisions on the original part then use a **Shrinkwrap modifier** so the part with the holes follows the original topology. Make sure the Shrinkwrap modifier is placed underneath the Subsurf modifier.

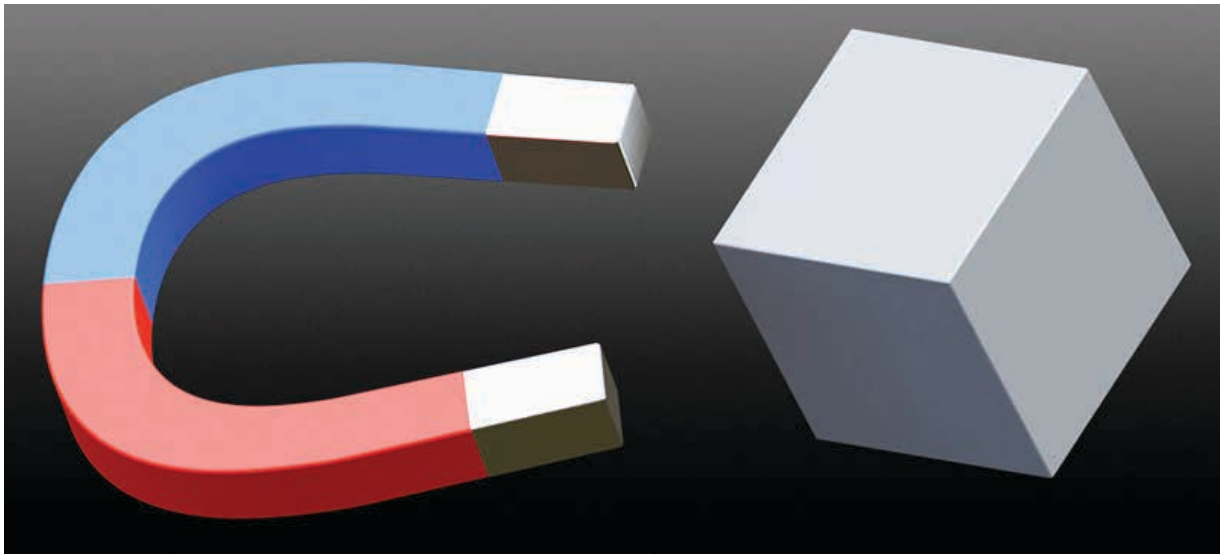


By using the Shrinkwrap modifier and the underlying perfectly smooth geometry, you can change the duplicate geometry without it losing its smooth topology.



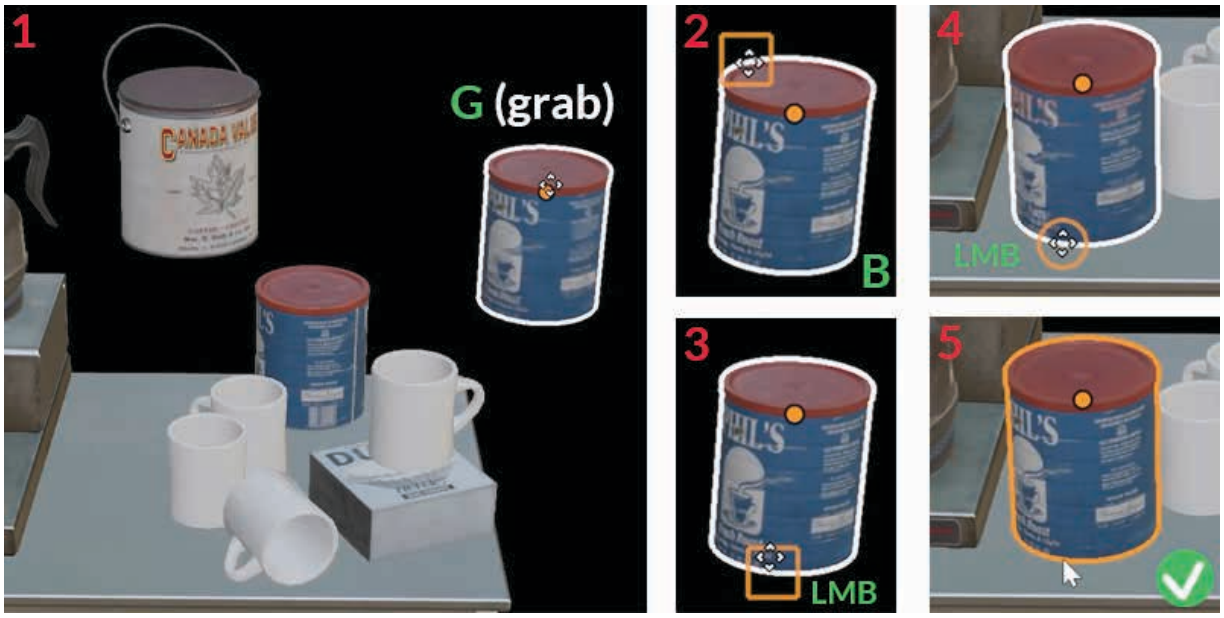
Snapping with Base Points in Blender 4

The Base Point feature in Blender 4 allows for fast and accurate snapping.



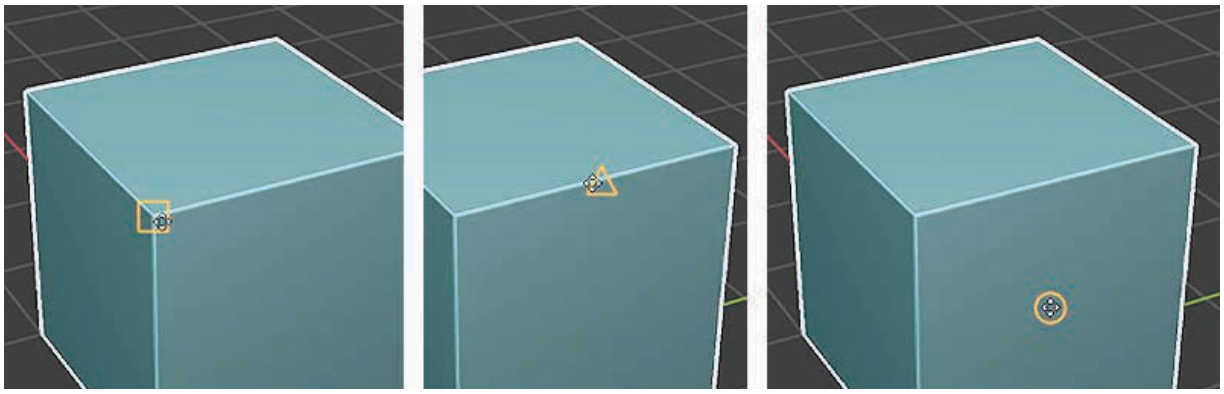
Before Blender 4.0, placing things accurately was quite time-consuming. You had to for example place the 3D cursor somewhere, then select the object. Then press Shift+S and choose Selection to Cursor. But it would only work if the Object origin was at its bottom, otherwise you'd have to change that first... The improvement in snapping Blender 4.0 brings will save literally hours of unnecessary steps and clicks, for many 3D artists.

Press G to start moving an object, then **press B** (for **Base Point**). **Left-click** to select the Base Point on that object. Then choose a target and **left-click** on it to snap the object to it. Done.

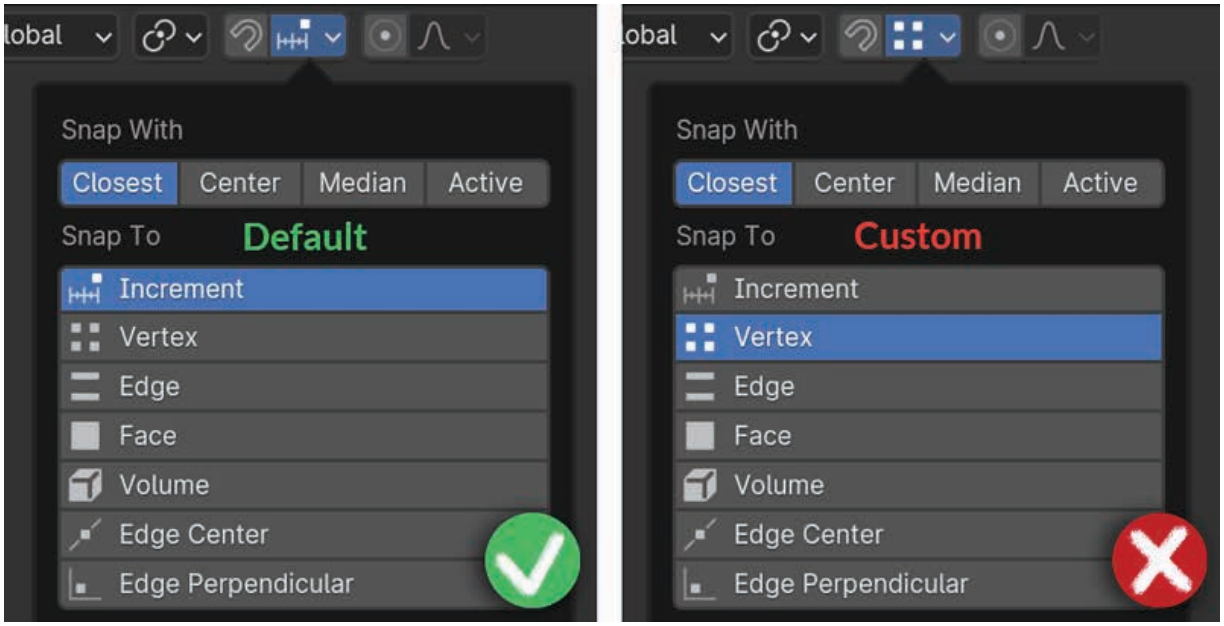


Coffee cart model by Joe Seabuhr on Poly Haven

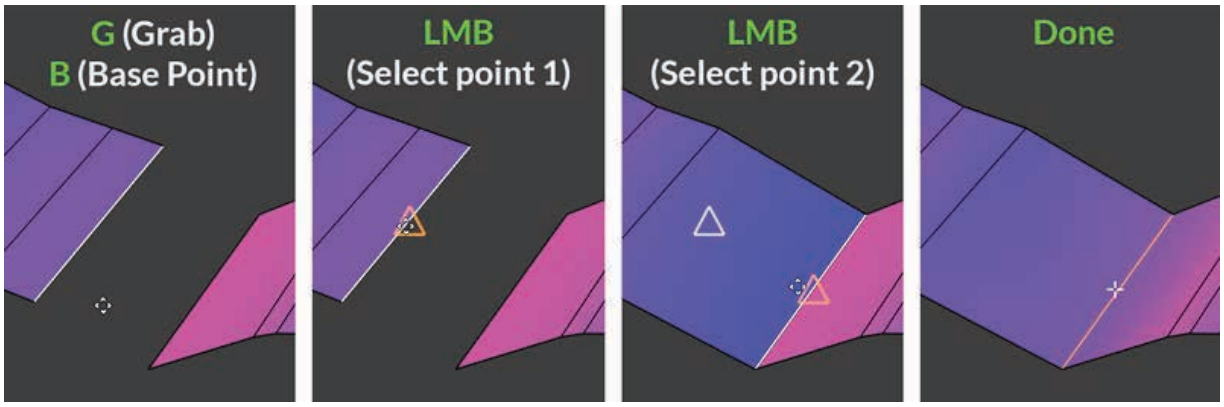
The Base Point can be a corner, the middle of an edge or the center of a Face. This is represented by the cursor, which will turn into a square, triangle or circle.



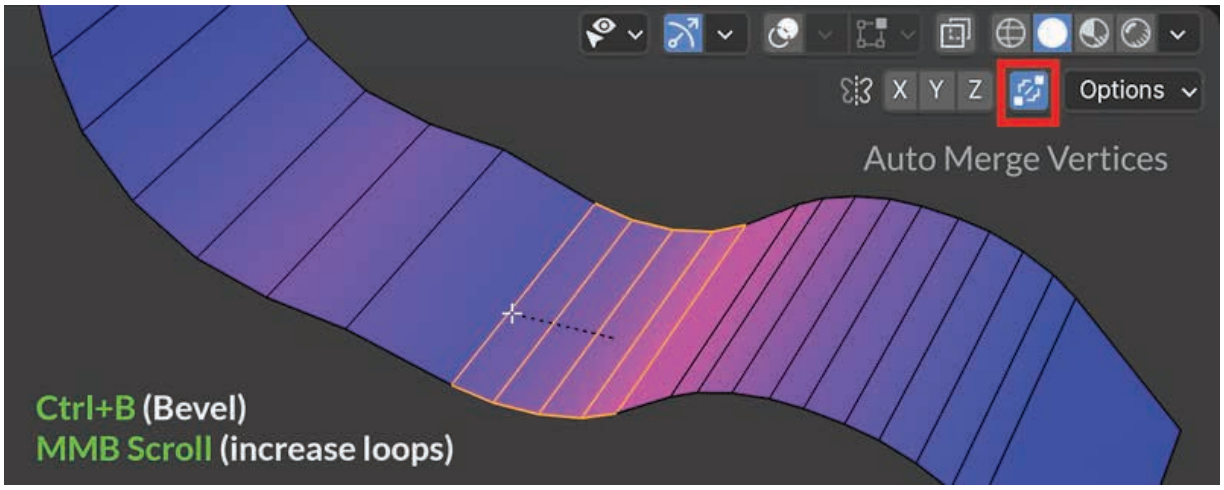
Note, that it's influenced by any custom settings you may already be using in the Snapping menu. If it's set to **Increment** , you can snap to all three of the points shown in the image above. However if you've set snapping to Vertex (for example), the Base Point snapping will also be limited to that selection. So keep that in mind.



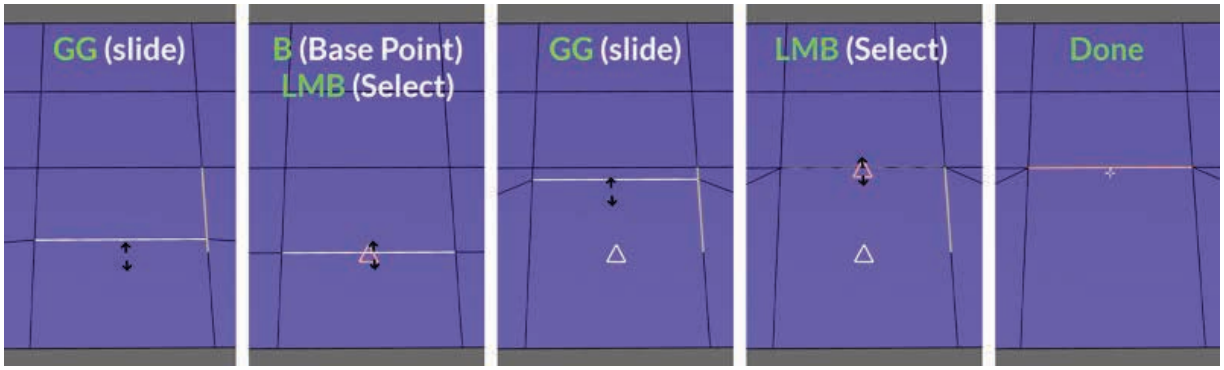
It also works in **Edit Mode** . Select a Vertex, Edge or Face, and **press G** . Then **press B** to choose the Base Point, **left-click** and then choose the target. And **left-click** to snap it there. Done.



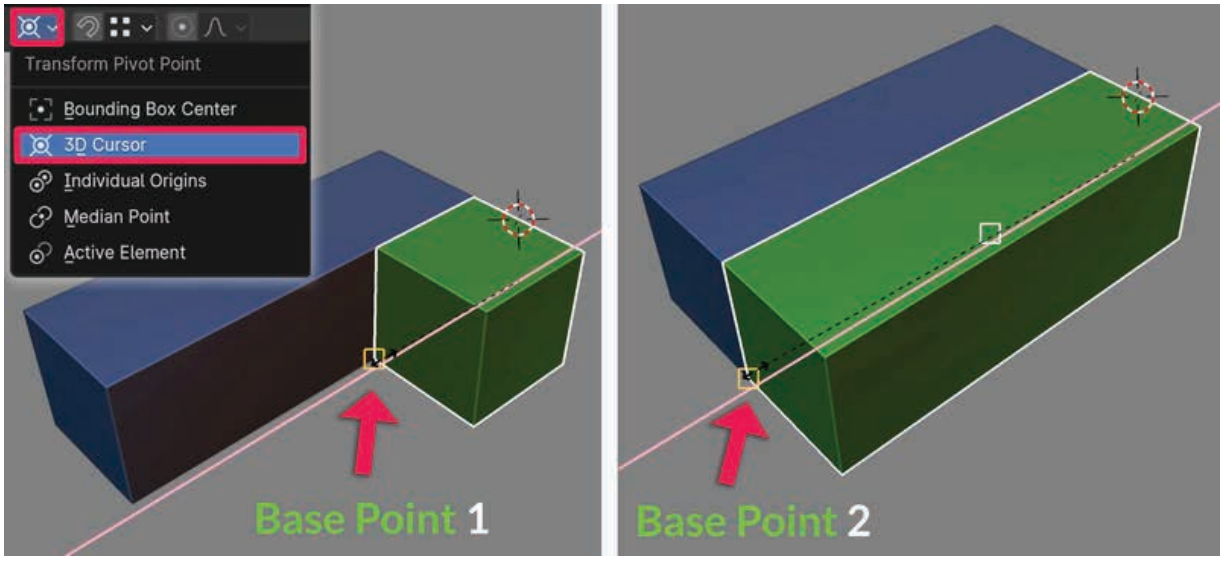
If you've got **Auto Merge Vertices** enabled, the snapped Edges will automatically be merged. Then you can immediately bevel it if you want, by pressing Ctrl+B.



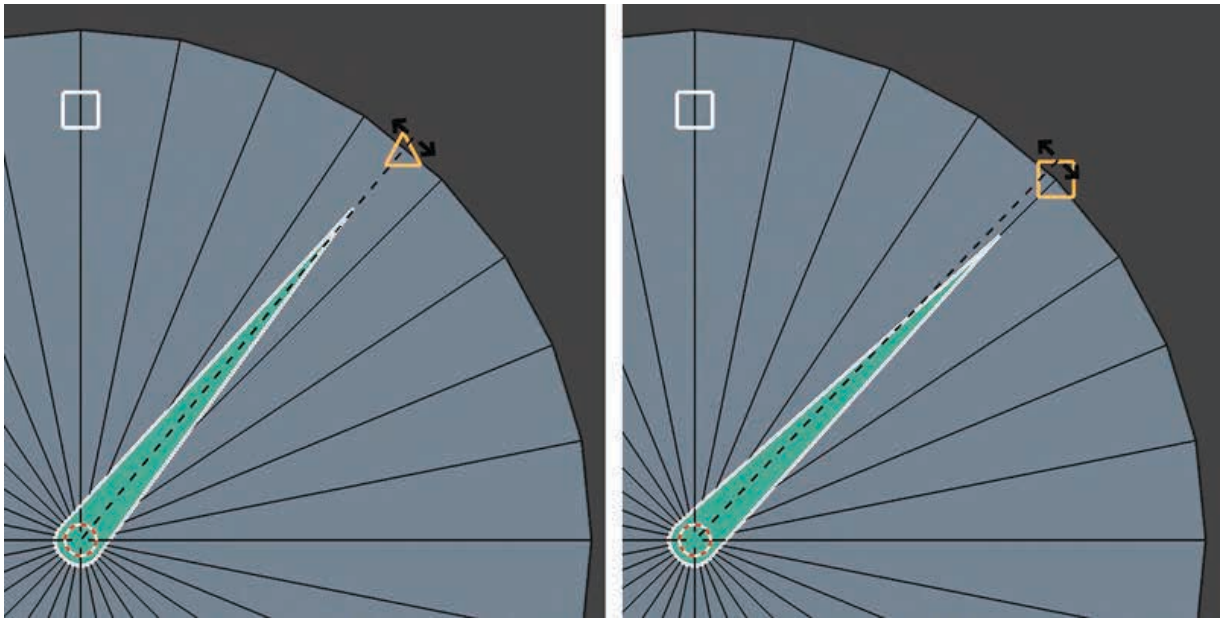
This also works when sliding an Edge or Vertex. Sliding is achieved by **pressing G** twice.



Base Point snapping is also available when **scaling** . This is especially handy when scaling away from the 3D Cursor.



And it works when **rotating** as well.



Sometimes you have to select a Base Point on the bottom of an object, and then it's hard to see the Base Point you want to snap it to on another object. In that case, you'll need to rotate the 3D view. This is possible since Blender 4.0.

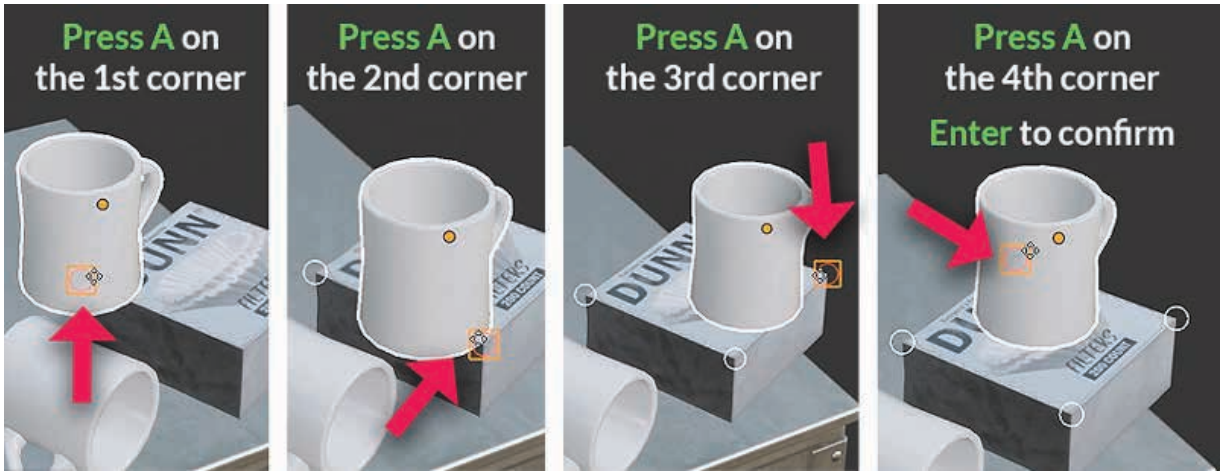
To do this, after pressing G and B to set the first Base point, **hold down Alt and the Middle Mouse Button** and navigate in your 3D scene. Then set the target Base point.

Pressing A after pressing B for Base Point snapping gives you the option to select multiple base points . It can be convenient to select the corners of an object, and get the median point of those.

You can watch a short video clip about it here. Below you can see the steps: First, set the Base Point on your object as usual. Then if necessary, adjust the view with Alt+MMB.

Then hover over each corner you want to use as a Base Point, and **press A** each time. You'll see a small circle being added there. Unfortunately the original object will move along with your mouse cursor which can make it a bit harder to see what you're doing. But you can tell that you're hovering over a corner when the mouse cursor becomes a square. You can set as many points as you want.

Press Enter to confirm, and the object will snap to the median point of all the Base Points you selected.



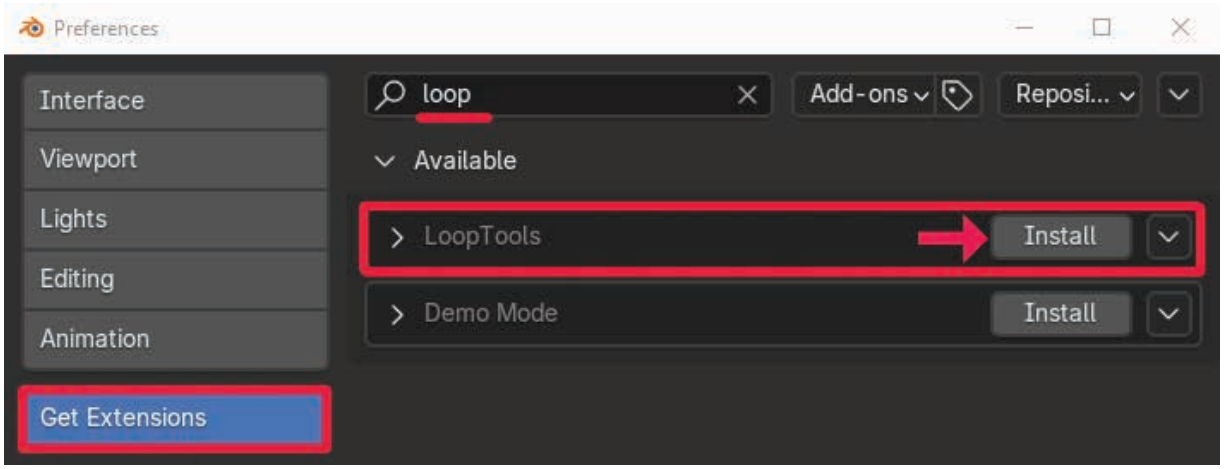
MODELING

Cylinder with Holes / Barrel

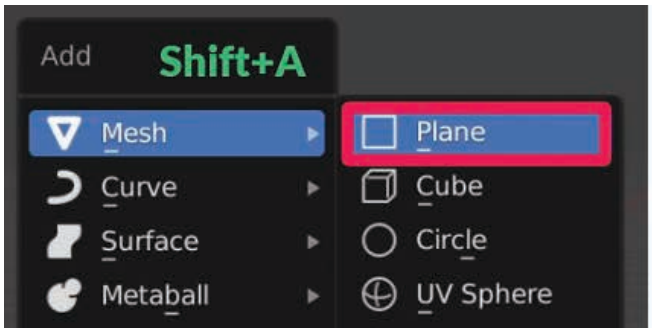
This method of adding holes to a cylinder takes advantage of modifiers.

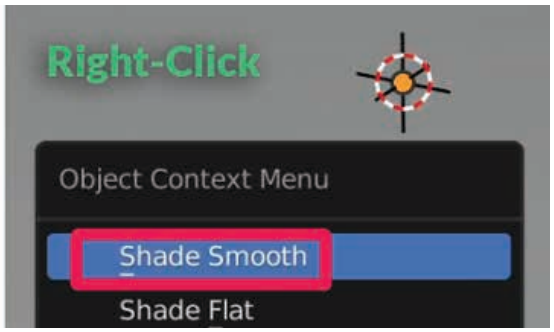
The benefit is, that we only have to make one hole, and then create the rest non-destructively.

Install **LoopTools** in Preferences > Get Extensions.

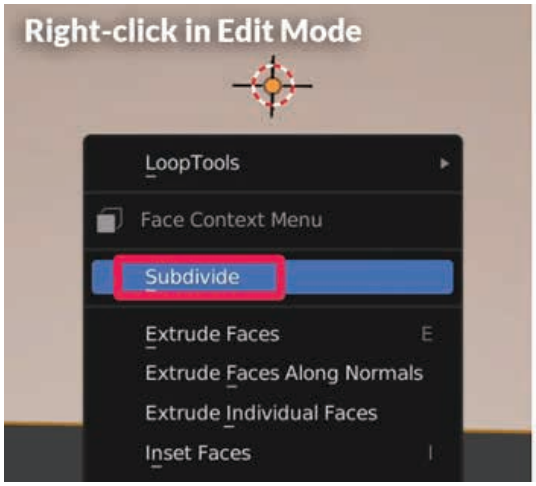


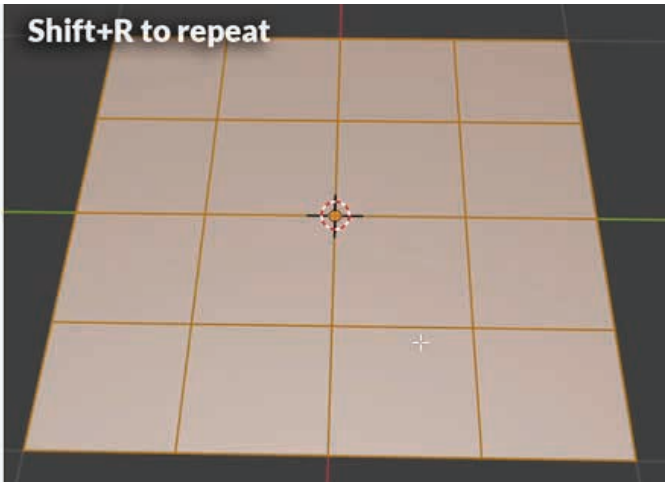
Add a **Plane** and set it to **Shade Smooth**.



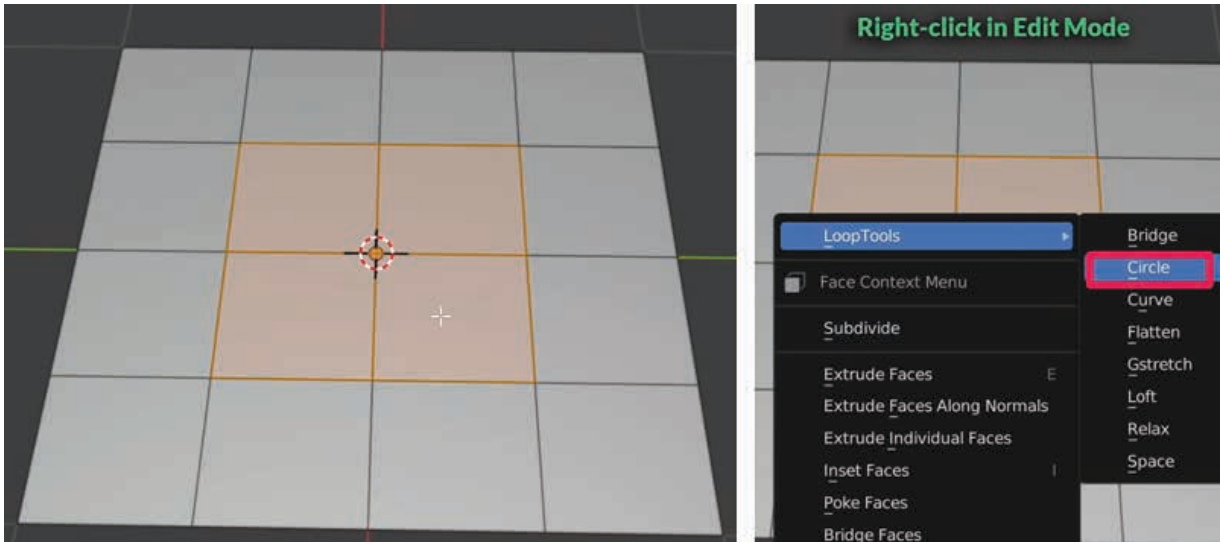


Subdivide the Plane twice in Edit Mode.

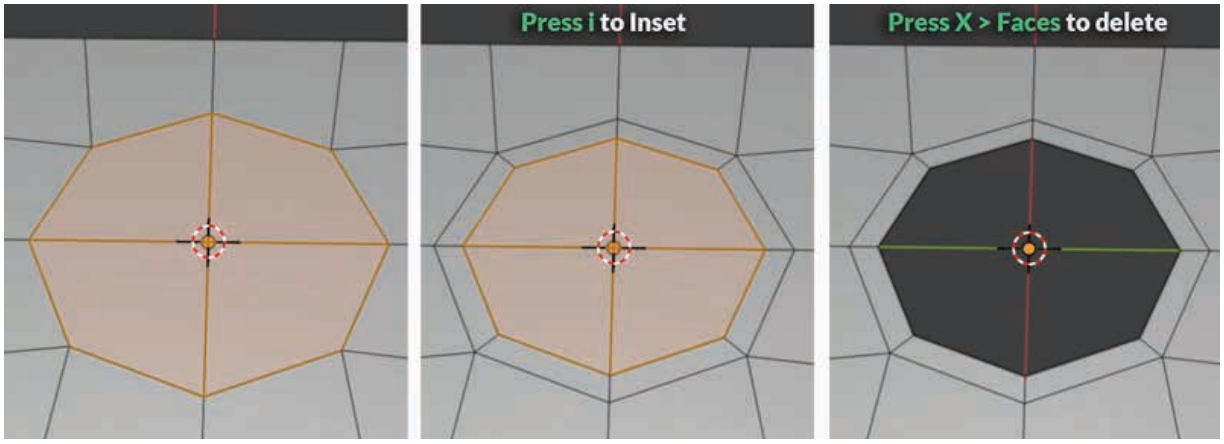




Select the 4 middle Faces, right-click and use **LoopTools > Circle**.

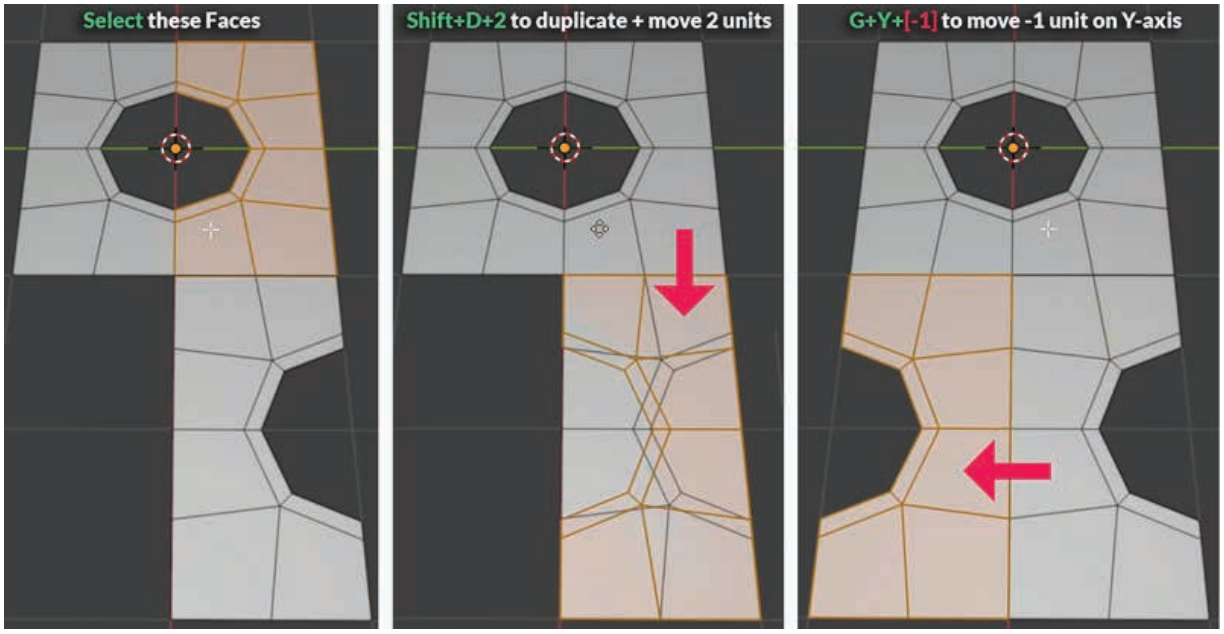


Inset the Faces by **pressing i**, then delete them by **pressing x**.

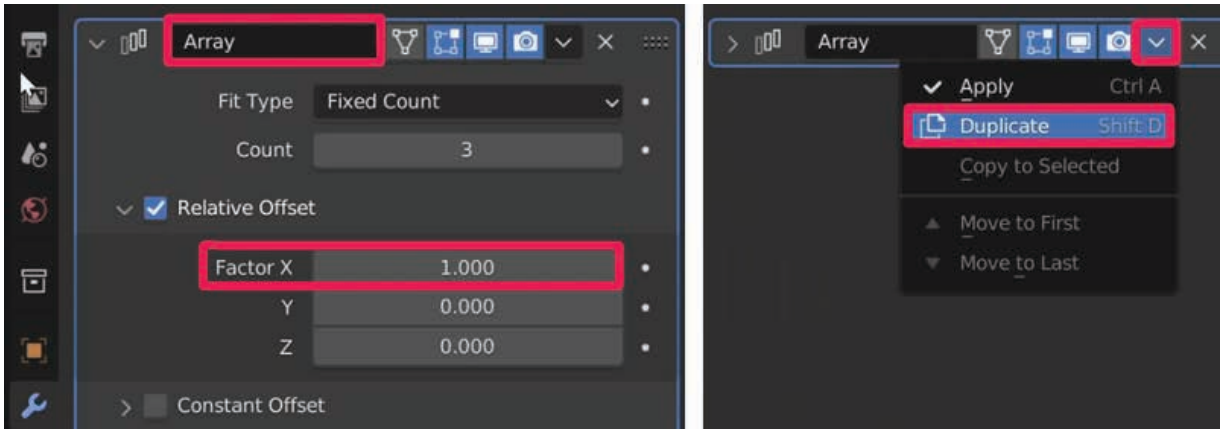


Duplicate and move selected Faces to create an uneven look. **Press Shift+D+[2 on the Numpad]** to duplicate and immediately move the duplicate 2 units on the X-axis. Then **press G+Y+[1 on the Numpad]** to move it 1 unit on the Y-axis.

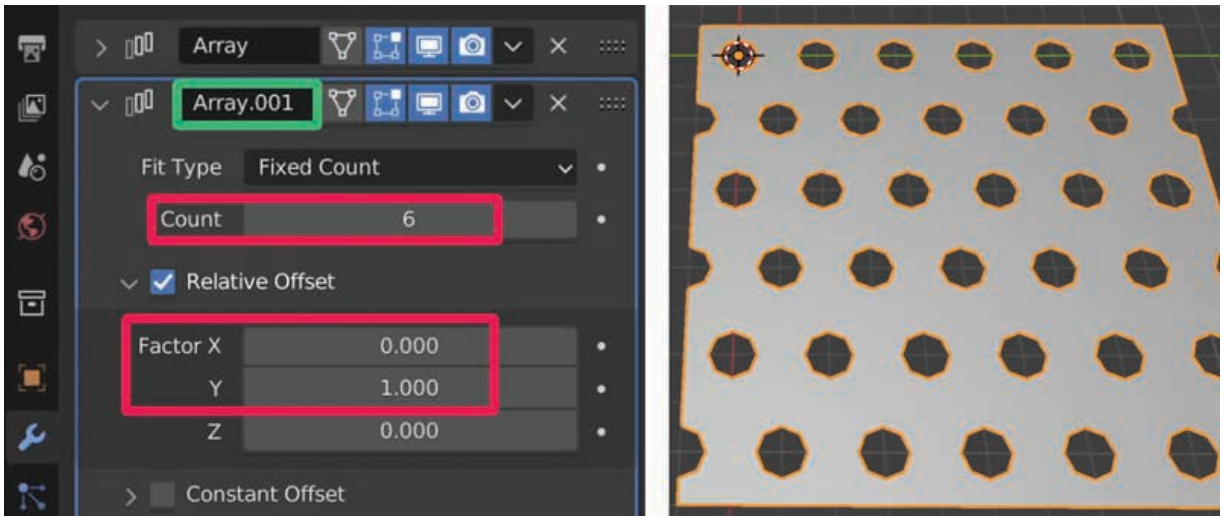
Repeat those steps for the other half, but move it -1 unit on the Y-axis instead of 1 positive unit. In other words, **press Shift+D+[2 on the Numpad]** to duplicate and immediately move the duplicate 2 units on the X-axis. Then **press G+Y+[-1 on the Numpad]** to move it -1 unit on the Y-axis.



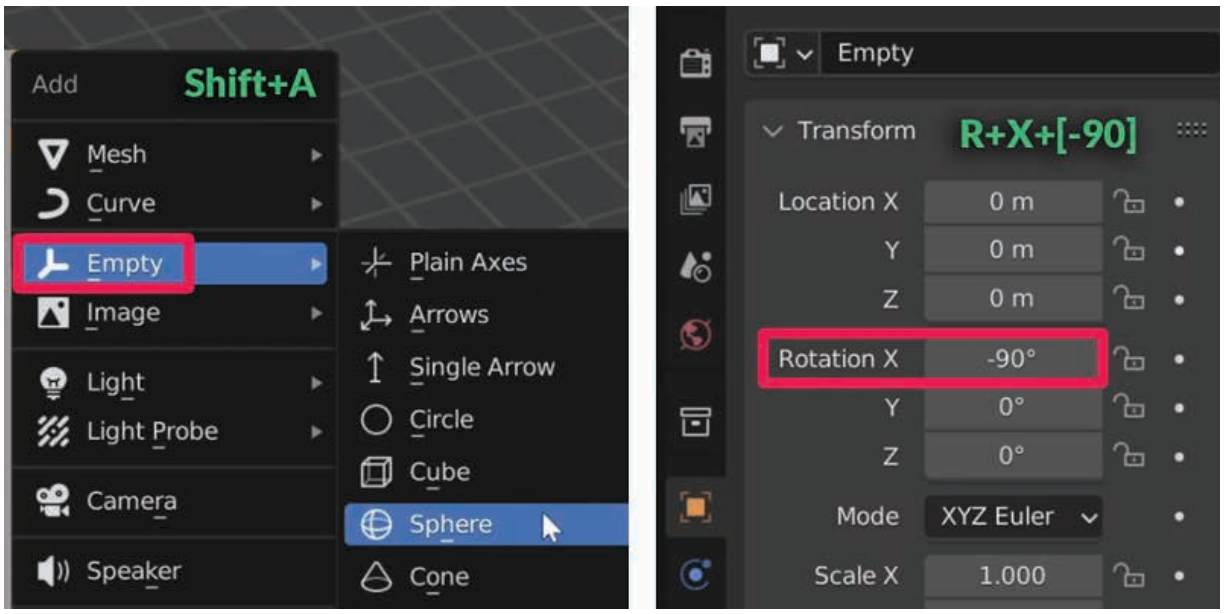
In Object Mode add an **Array modifier**. Duplicate it.



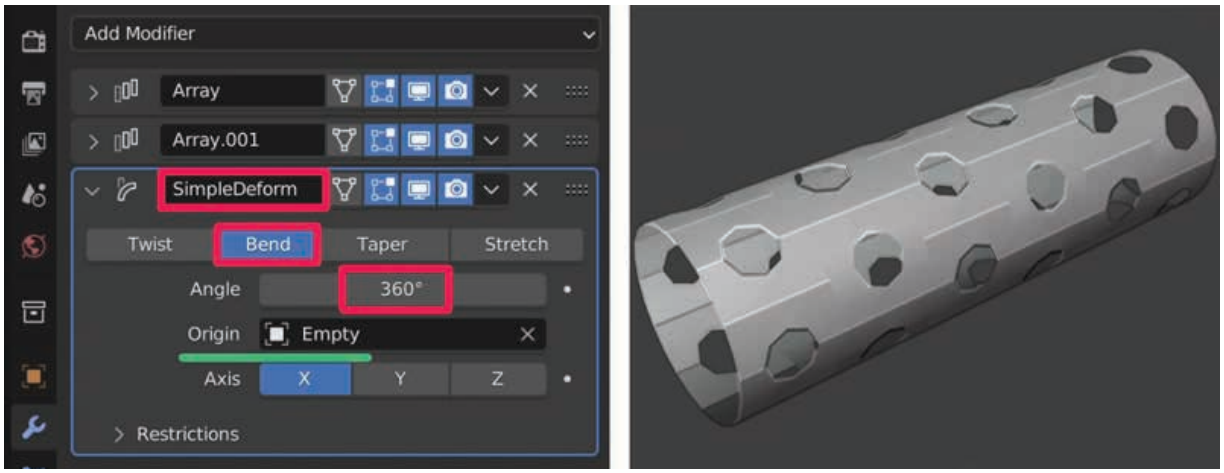
In the duplicated modifier, set the **Relative Offset to Y** instead of X and increase the **Count**.



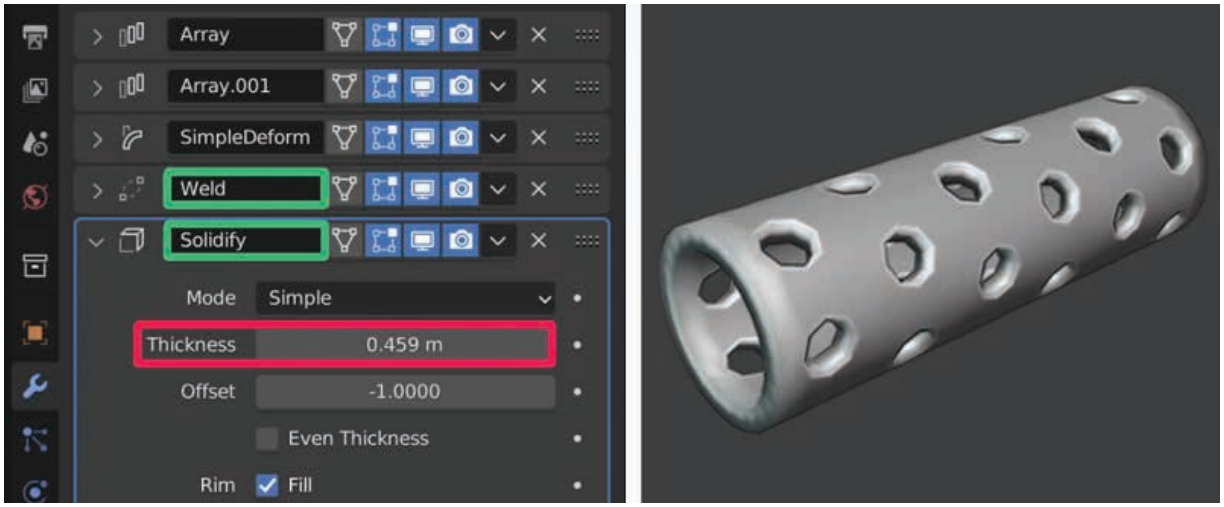
Add an **Empty** (any shape) and set **X Rotation to -90**. You can do this either by **pressing R+X+[-90 on the Numpad]** or by entering -90 in the Rotation X field.



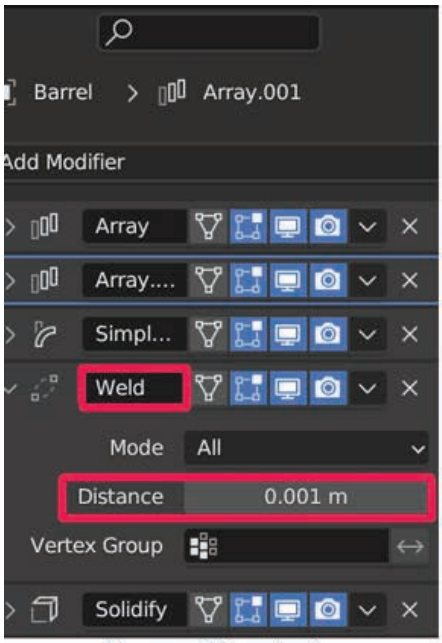
Give the Plane a **SimpleDeform modifier** set to a **360° Bend** with the Empty as the **Origin**.



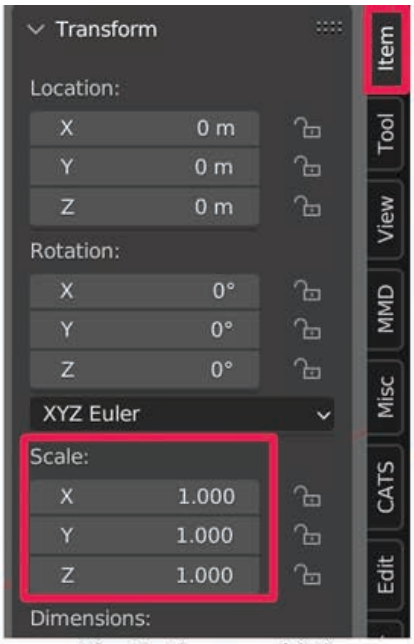
Add a **Weld modifier** to merge vertices that are close to each other and a **Solidify modifier** to give it some Thickness.



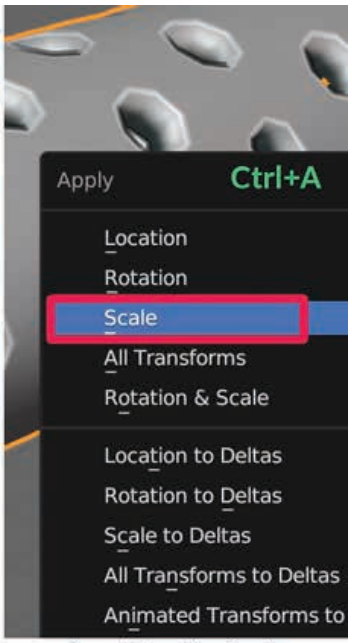
Note that you may have to increase the **Distance** value of the Weld modifier to get it to weld the vertices. However, if this is the case you may also want to check the **Scale** of your object in the Option panel (N). If the Scale isn't 1, that could cause unpredictable behaviour with modifiers. You can always apply the scale by **pressing Ctrl+A** and choosing **Scale**.



The modifier stack

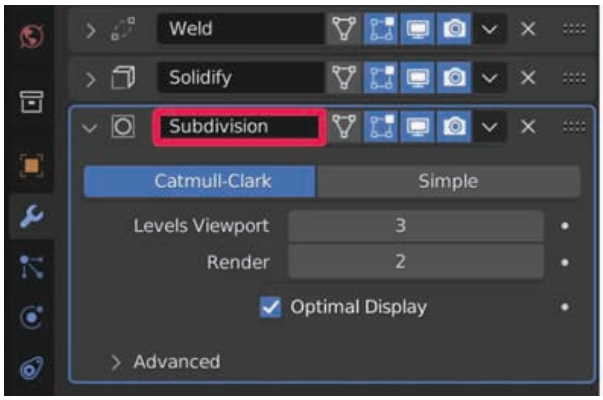


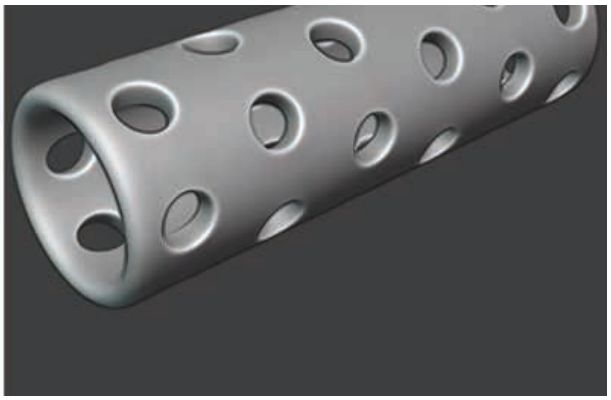
The Option panel (N)



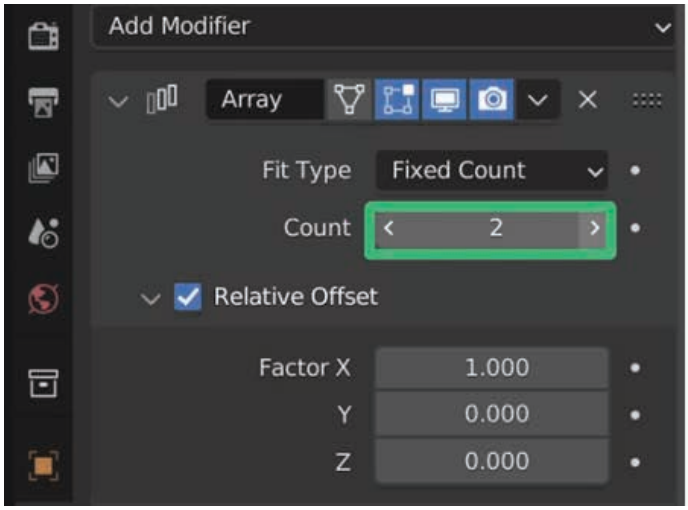
Applying the Scale

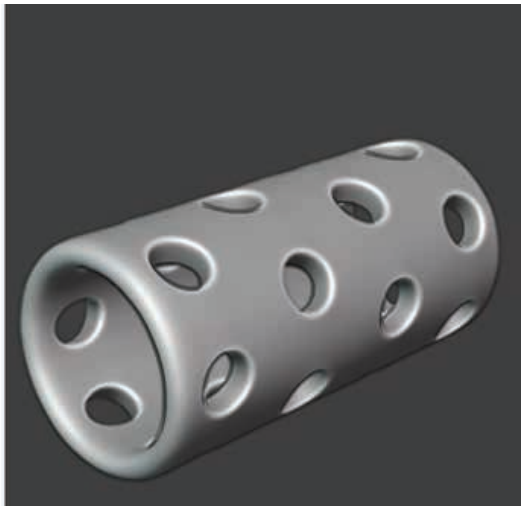
Finally add a **Subdiv modifier**.





You can change the **Array Counts** to adjust the shape as needed.

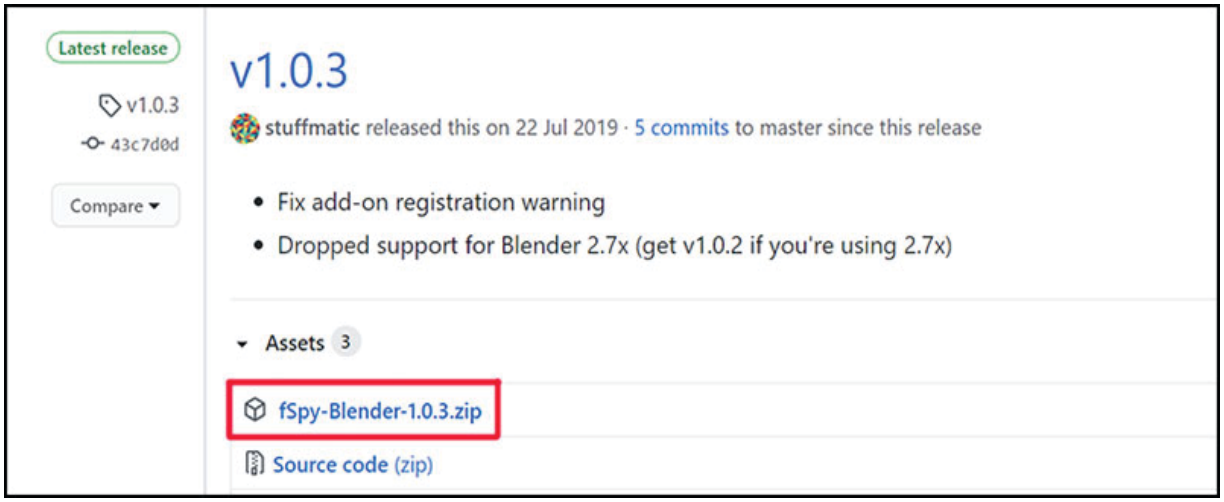




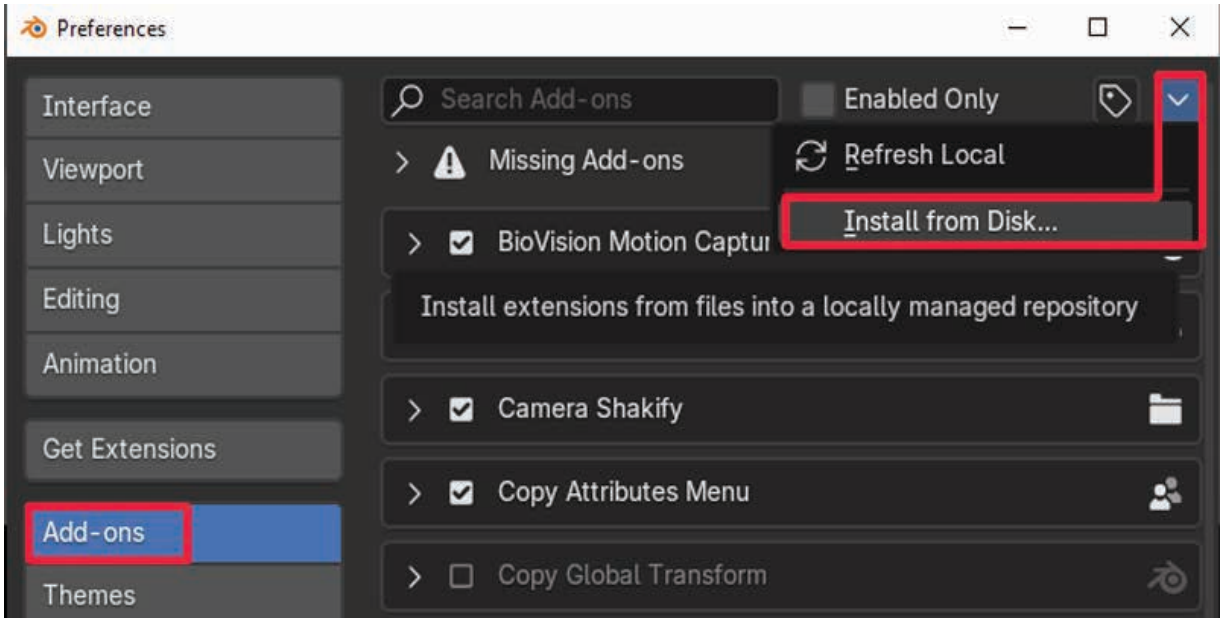
Camera Projection modeling part 1 - fSpy to Blender

The free **fSpy add-on** for Blender makes it easy to turn any photograph into a 3D scene, so that you can seamlessly place objects in there or animate the camera. This is a technique often used in visual effects for movies, like in the Matrix for example.

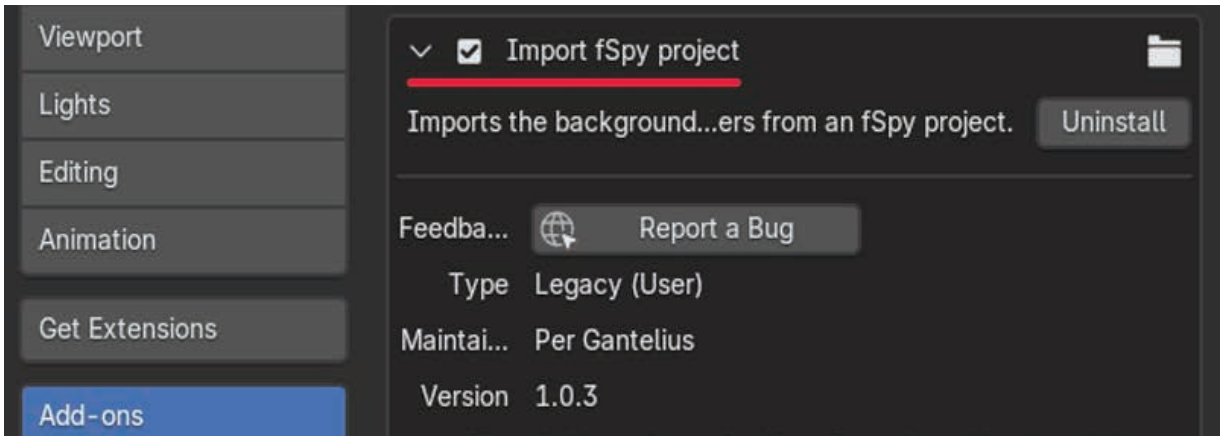
To install the add-on, go to this Github page and **download the fSpy Blender add-on**. The version I'm using is **version 1.0.3.** Download, but don't unpack, the zip file. If you're on a Mac, make sure the zip doesn't automatically get unpacked.



Install the add-on by going to Blender Preferences and installing the unpacked zip. In the **Add-on** tab, click on the drop-down menu to find the **Install from Disk** option.



After that, the add-on is automatically enabled. What this add-on does, is that it adds an "**fspy**" option to the **File > Import** menu in Blender.

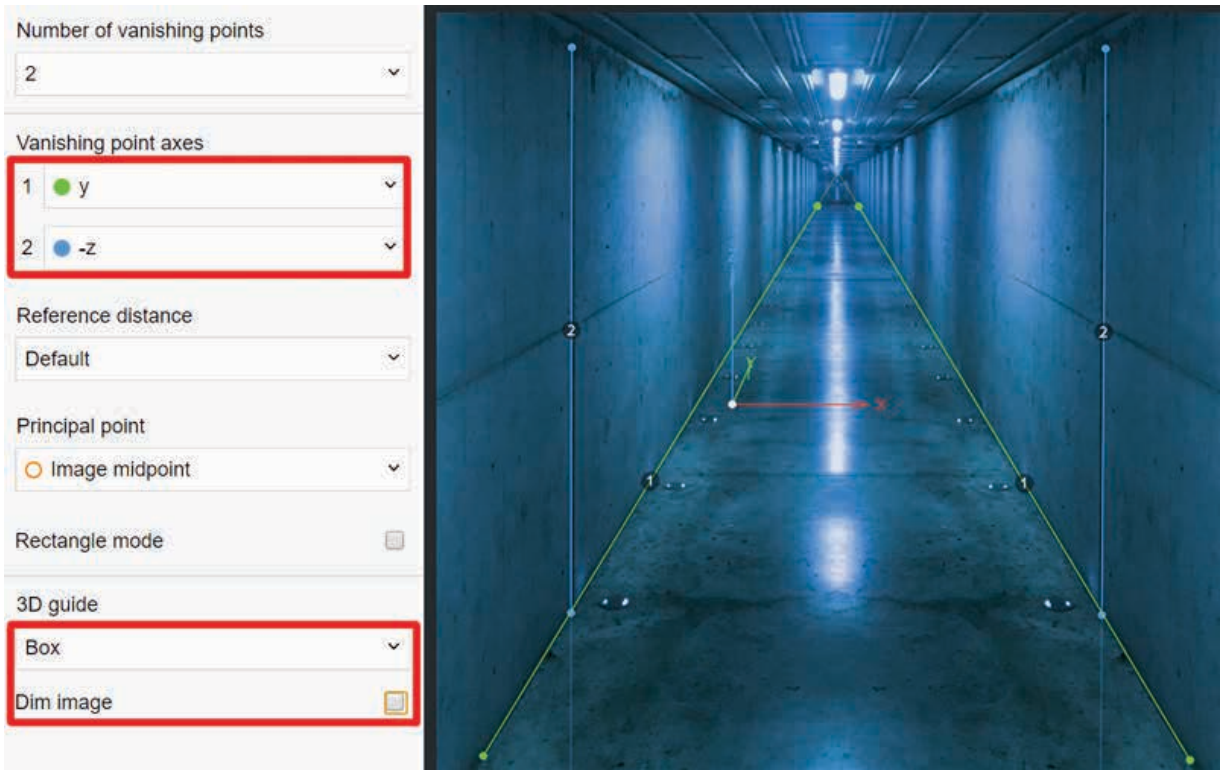


Now we can import fspy files in Blender. However, we still need to be able to make those files. For this, we need to download something else. We need the stand-alone fSpy software. Go to https://github.com/stuffmatic/fSpy/releases/tag/v1.0.3. If you're on Windows like me, download and run the file **fspy-setup-1.0.3.exe**.

That's all we have to install. Now when running the fSpy app (which should run automatically after you've installed it) you can drag a photograph to it. The photograph I'll be using in this example is one taken by Elti Meshau on Unsplash (a website where you can find a lot of nice free photographs).

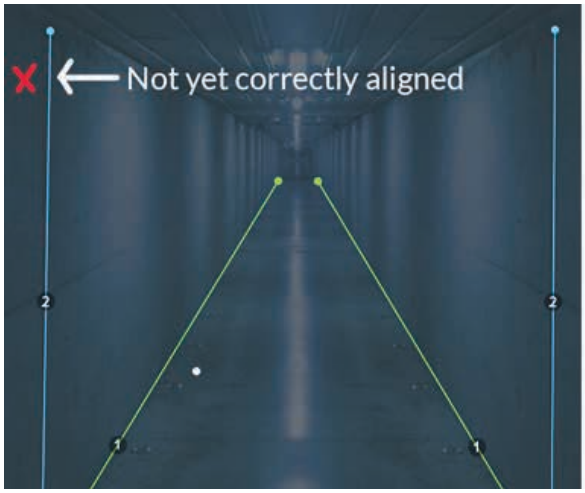
Drag and drop the photograph on the fSpy app canvas. Blender uses the **Z-axis as the vertical axis**, and the Y-axis as the axis that goes to the distance, so you'll need to change these setting in fSpy. Set **axis 1 to Y** and **axis 2 to -Z**.

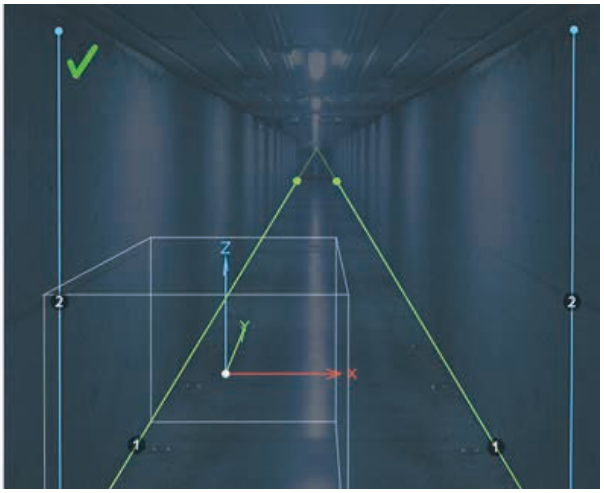
Also under 3D guide choose "**Box**". I also like to **uncheck "dim image"**, that way the photograph is easier to see. After this, you can move the blue and green lines by grabbing their end vertices and place them so that they follow vertical lines (for the blue Z-axis) and horizontal/perspective lines (for the green Y-axis).



Fortunately, this photograph contains some very clear perspective lines as well as vertical lines created by the groove where the slabs of concrete meet. We can use these visual cues to align the red and blue lines.

Once the perspective lines are lined up well enough, a box will appear. If it doesn't, keep trying to align the perspective lines to the photographic perspective as best you can until it does.



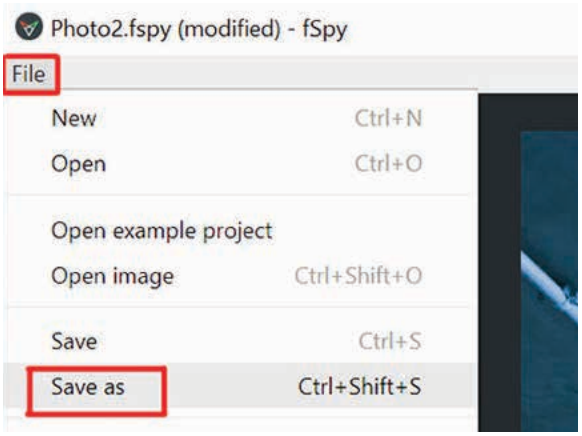


By **holding Shift** while dragging the red and blue axis handles, a magnified circle will appear that lets you fine-tune where you place the handles.

Another tip for more accuracy is to simply work with the fspy window maximized.

The **Box** 3D guide option places a simple wireframe cube inside your scene which you can move by grabbing the white origin point of the Y+Z arrows. This way you can move it around to check if it lines up with the photographic perspective. If it doesn't, try placing the blue and red lines along different visible lines in the photograph, until the box lines up better.

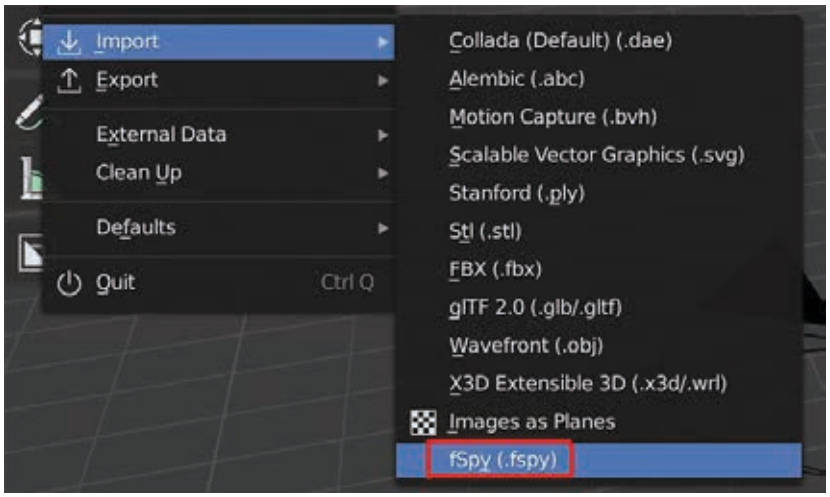
Save the fSpy file once you think the perspective is lined up well enough.



Now we can open it in Blender.

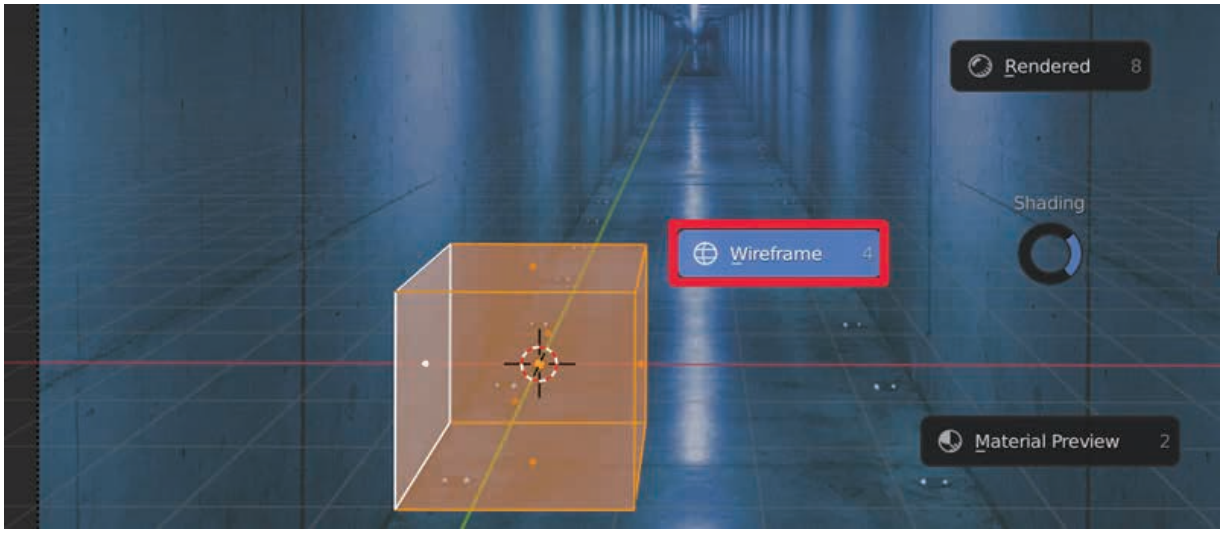
You may want to select and delete the default Blender camera first, as fSpy loads its own camera. That way there's no confusion.

Go to **File > Import > fSpy** and find the fSpy file you saved.

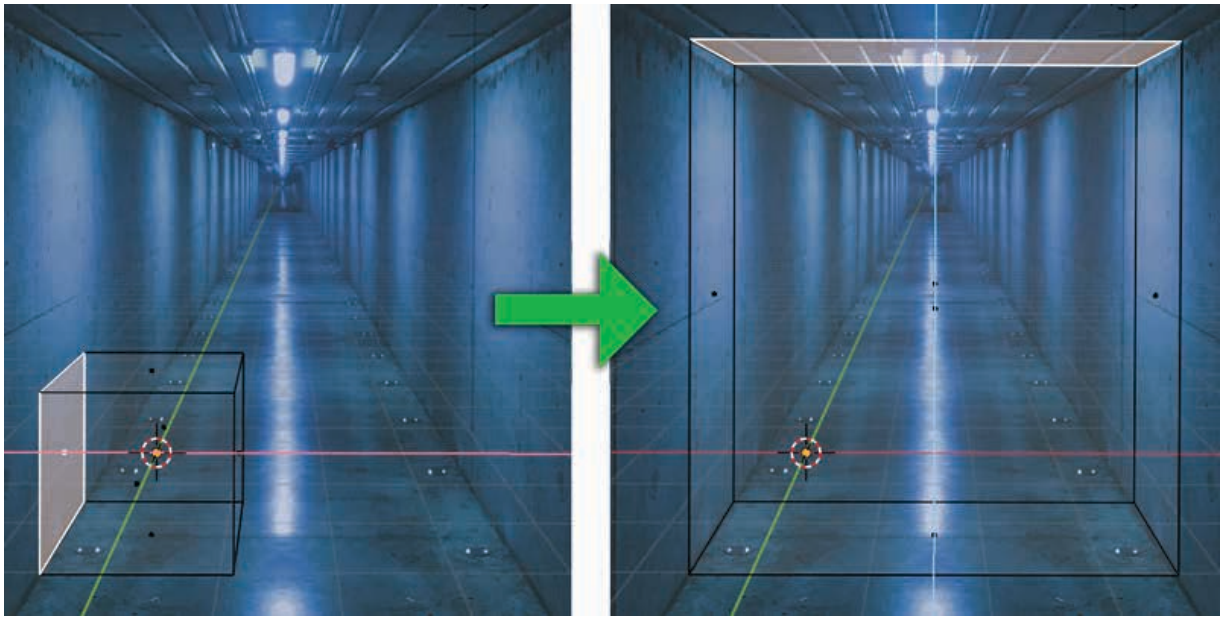


Now the photograph has been loaded into Blender as a Camera background, and the Scene dimensions match the photograph.

The default cube now rests inside the world of the photograph. Although just placing 3D objects in the photograph is one possibility, another is to recreate the photograph in 3D as an environment. **Press Z and choose Wireframe.**



In Edit Mode, move the faces of the cube one by one to match the photograph. Be careful to only move them along either the **X and Z axis** initially (by pressing G+X and G+Z), until they line up to the walls, floor and ceiling in the photo like this:

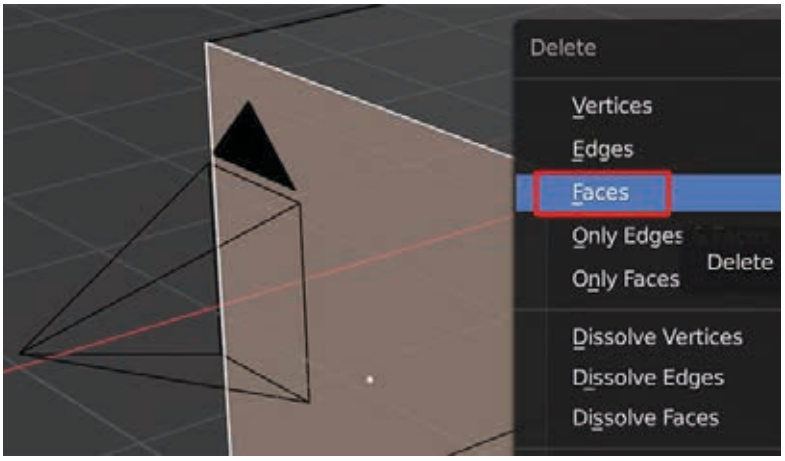


Now you can also extend the front and back Face of the Cube further on the **Y-axis**. Thanks to how fSpy has correctly identified the camera settings and perspective, you can see that the edges line up perfectly with the perspective.

Now, if we want to give this cube the projected texture of the protograph, we need to do a couple of things.

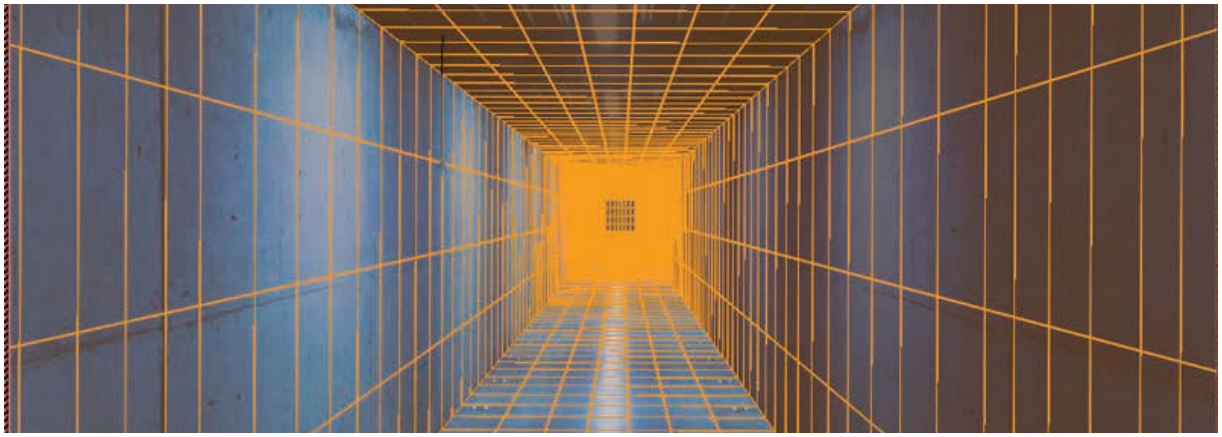
first, remove the Face that's closest to the camera.

Next, you have to add some more edge loops.

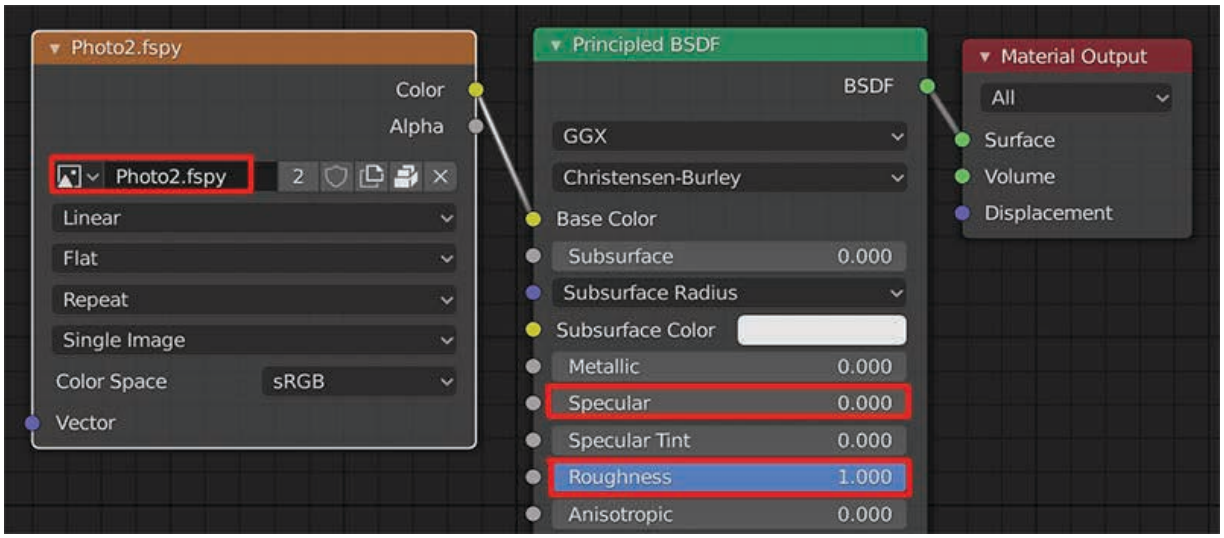


If you don't do this, the texture will unfortunately look all "wobbly". **Press Ctrl+R** to add edge loops.

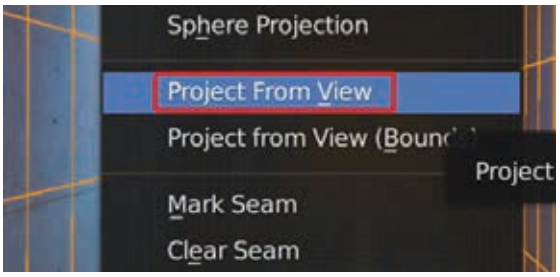
Next, add some subdivisions by **right-clicking and choosing "Subdivide"**. You can then repeat this by **pressing Shift+R** a few times.



Now, add a material to the object and as the Image Texture, use the photograph. Set **Specular to 0** (zero) and **Roughness to 1**. This is to make sure the material isn't shiny.



Then, select all in Edit Mode and **press U and choose Project From View**.



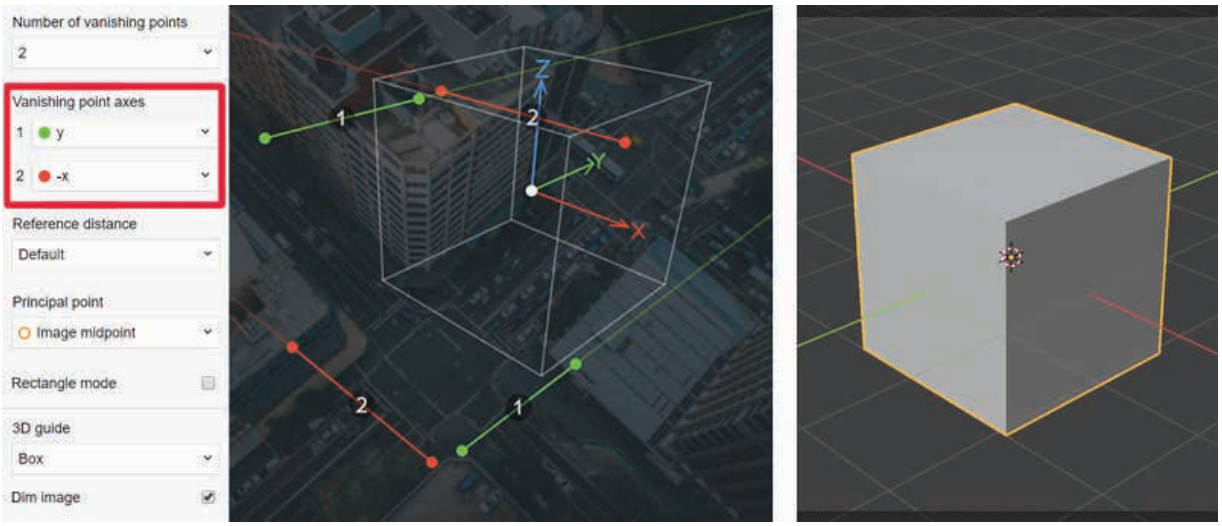
Now the 3D corridor has the photographic texture, and you can adjust the camera to find new angles.

If the texture still looks wobbly, you may need to add extra edge loops here and there.

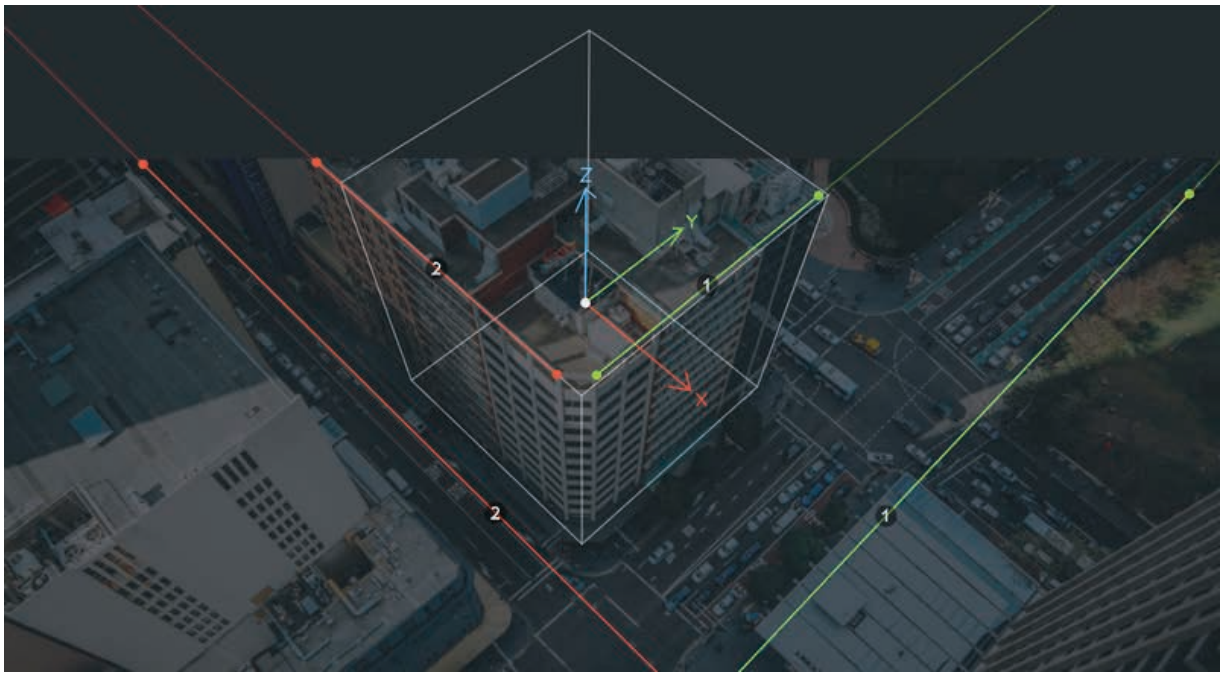
Camera Projection modeling part 2 - Create an Environment

Previously we looked at how to use **fSpy** to calibrate the camera so we can bring a photo into Blender with the correct camera settings that match the photo. Let's look a bit more at how to use that for 3D modeling.

Here I'm using this free aerial photograph of busy city streets originally found on Pixabay. This photo differs from the previous one in terms of the angle from which it was taken and where the vanishing points are. So we'll need to set vanishing point 1 to Y and 2 to -X. When in doubt, just compare it to the default camera view in Blender. It should be roughly the same.

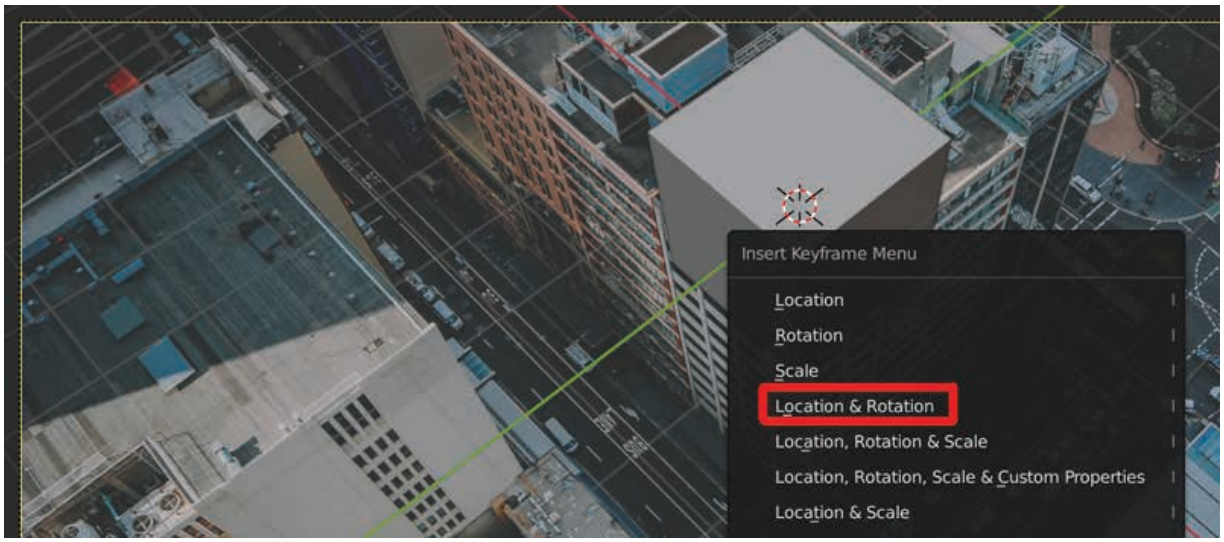


After finding some perspective lines to follow with the vanishing lines, you can see if the Box 3D guide overlaps with one of the buildings. It doesn't have to be 100% perfect.

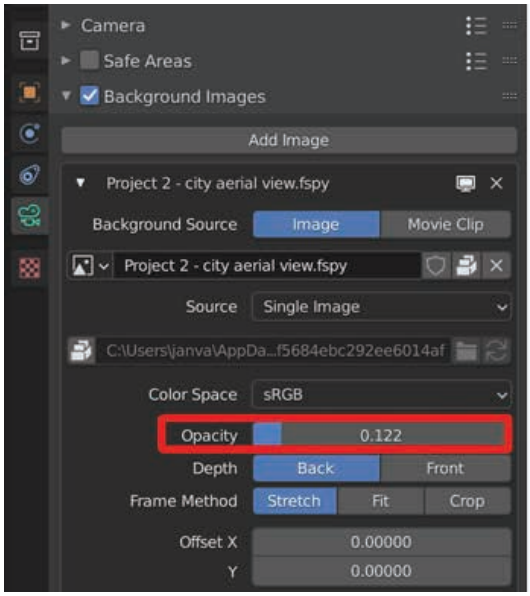


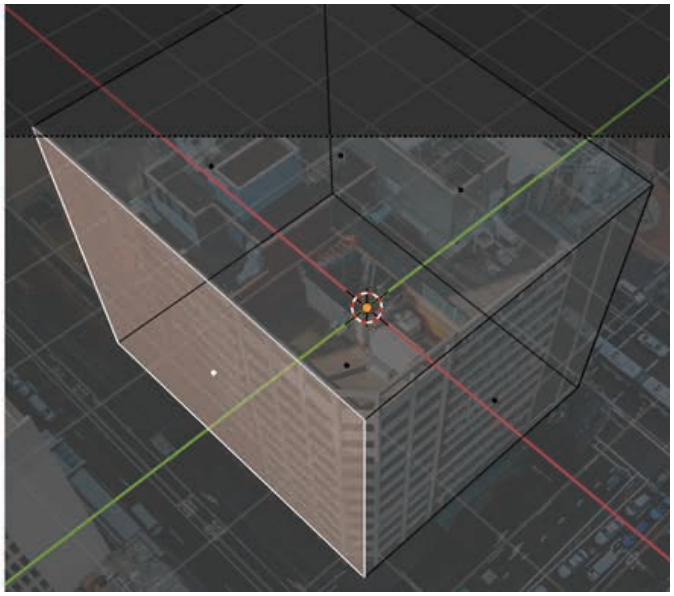
Save the file and import it in Blender (for more details, check the previous topic). You should get this camera view, with the default cube already almost in place of one of the buildings.

Before you do anything I recommend selecting the camera and **pressing i then choosing Location & Rotation** to set a keyframe for the calibrated camera position, just in case you accidentally move the camera.

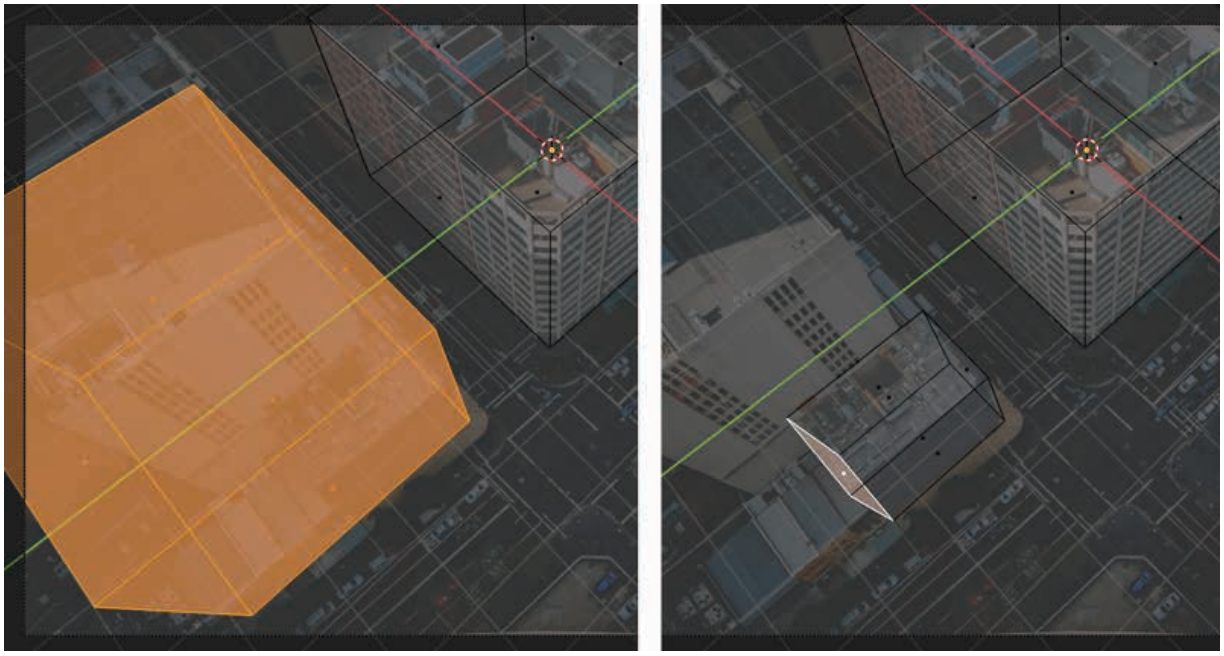


It can be convenient to set the Opacity of the camera background a bit lower. Select the correct camera in the Outliner (it will have the name of your fSpy file) and lower the Opacity. Switch to Wireframe mode by **pressing Z and choosing Wireframe**, then **press 3** for Face Selection mode and start moving Faces along the X, Y and Z axis by **pressing G+X, G+Y and G+Z** to block out the building.





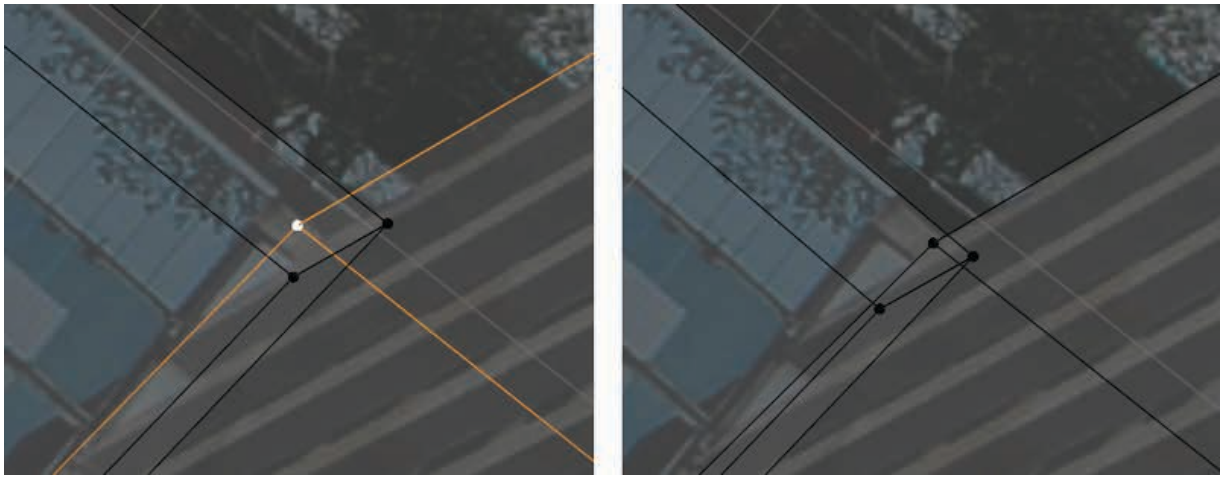
Press A to select all Faces, **press Shift+D** to duplicate the Cube and then **Shift+Z** to move the duplicate on all axis except Z. **Left-click** to confirm once it's in the spot where you want it. Then, move Faces again to match the building in the photo.



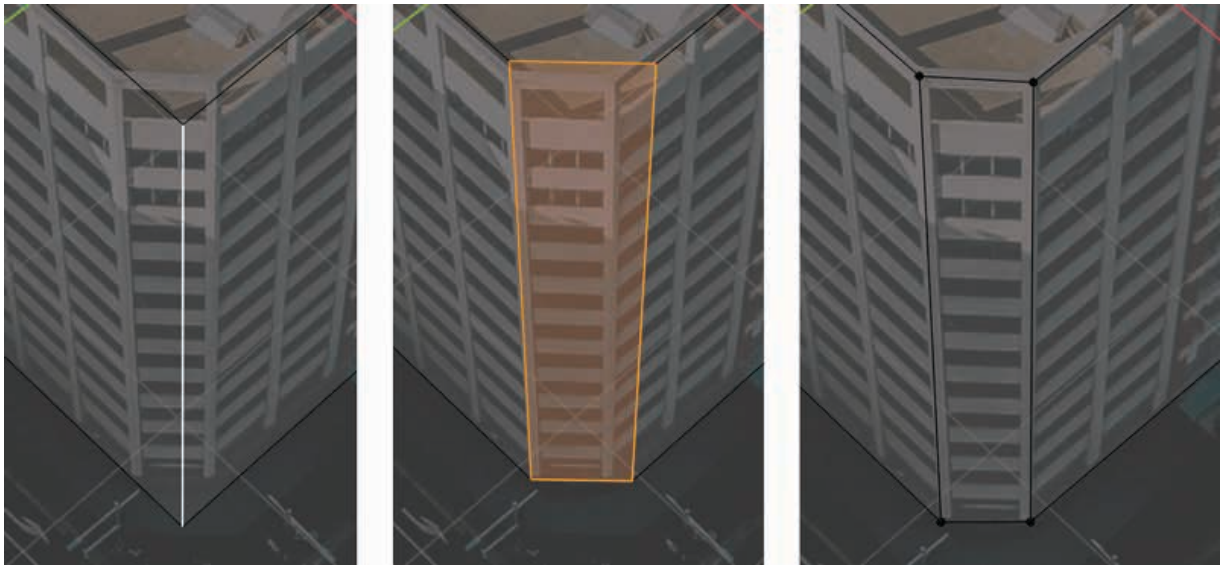
Now repeat these steps for the other buildings. To select a Cube to duplicate, just hover over it with the mouse cursor and **press L**. Or select a Face and **press Ctrl+L**.

Finally select a bottom Face and duplicate it, then scale it up until it fills the entire background photo. This will be what we project the street onto. Now you've got this primitive city made up of some boxes and a ground plane, all one object.

The camera calibration wasn't perfect, so some of the buildings don't exactly line up. We can solve this easily enough in **Edit Mode** by **pressing 1** for Vertex selection mode, and then selecting and moving some vertices so they line up better.

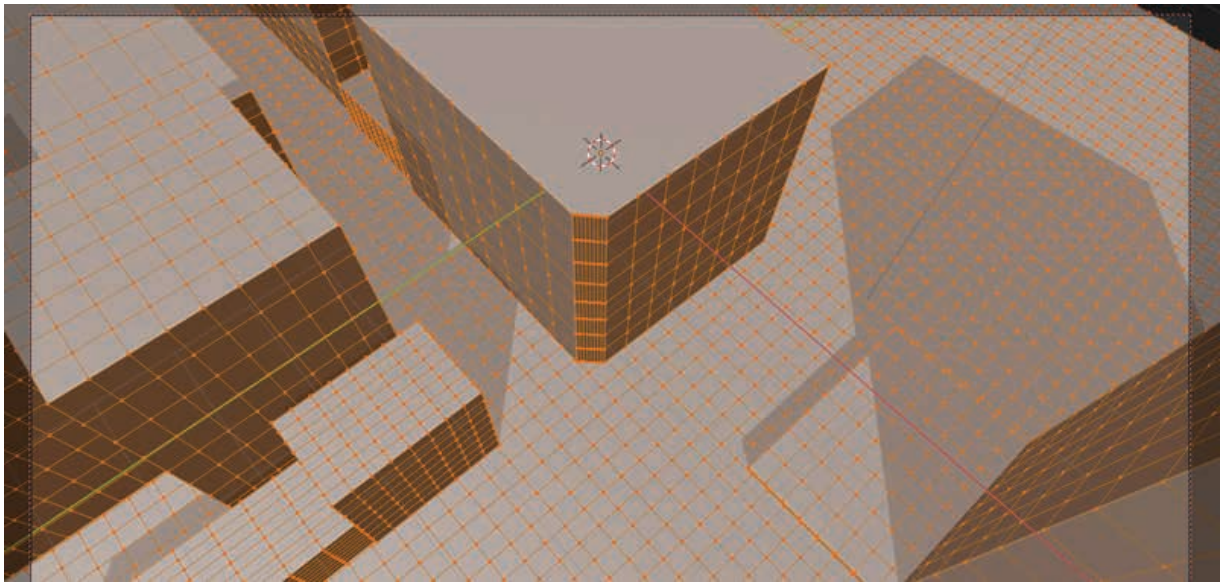


You can also further refine the model a bit more by adding bevels where needed like on this corner, by selecting an edge and **pressing Ctrl+B** and moving the mouse, then fine-tune it by selecting and moving individual vertices.

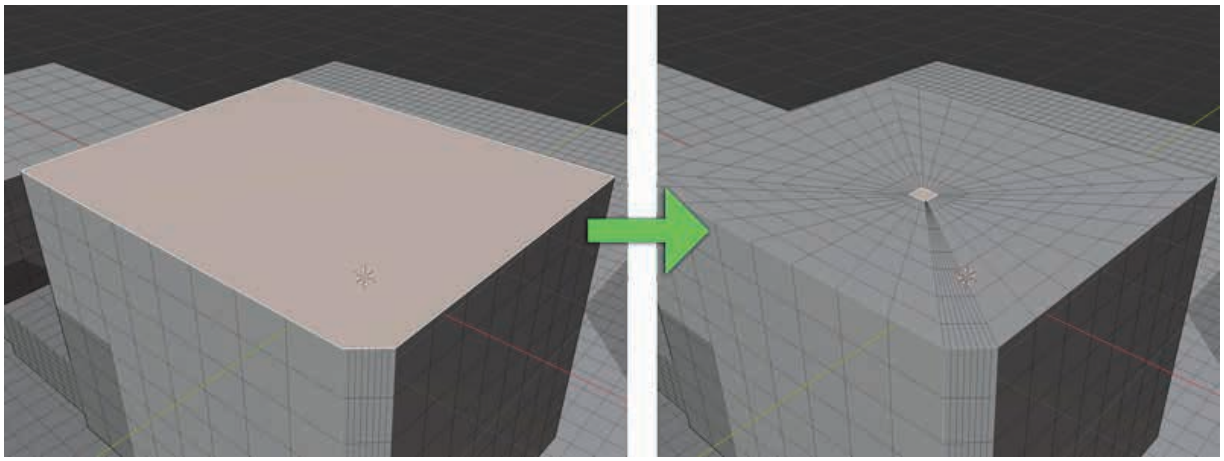


Subdivide the ground plane a few times and then select all and subdivide all the geometry a few more times. If you don't do this, the textures will not project well.

The reason we subdivided the ground plane first is because it's much bigger than the other geometry, and we want to end up with a more or less uniform distribution of geometry.



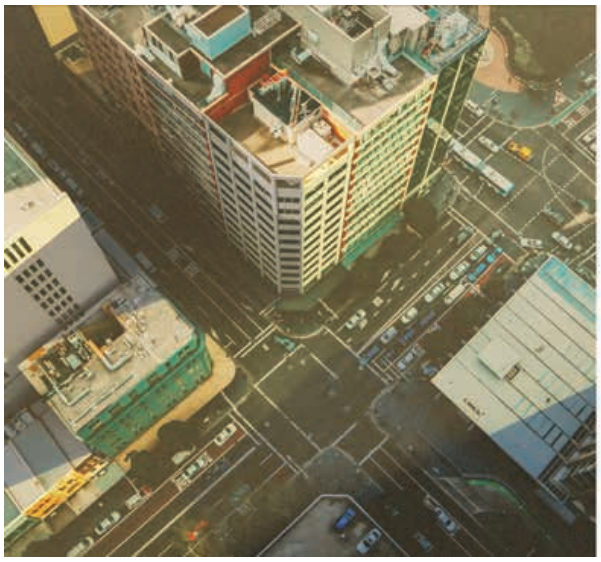
As you can see the top of the building is still one big N-gon. We can solve this by selecting it and **pressing E for extrude, followed by S and moving the mouse** to scale the new geometry down. Repeat this a few times until you end up with only a small face in the middle.



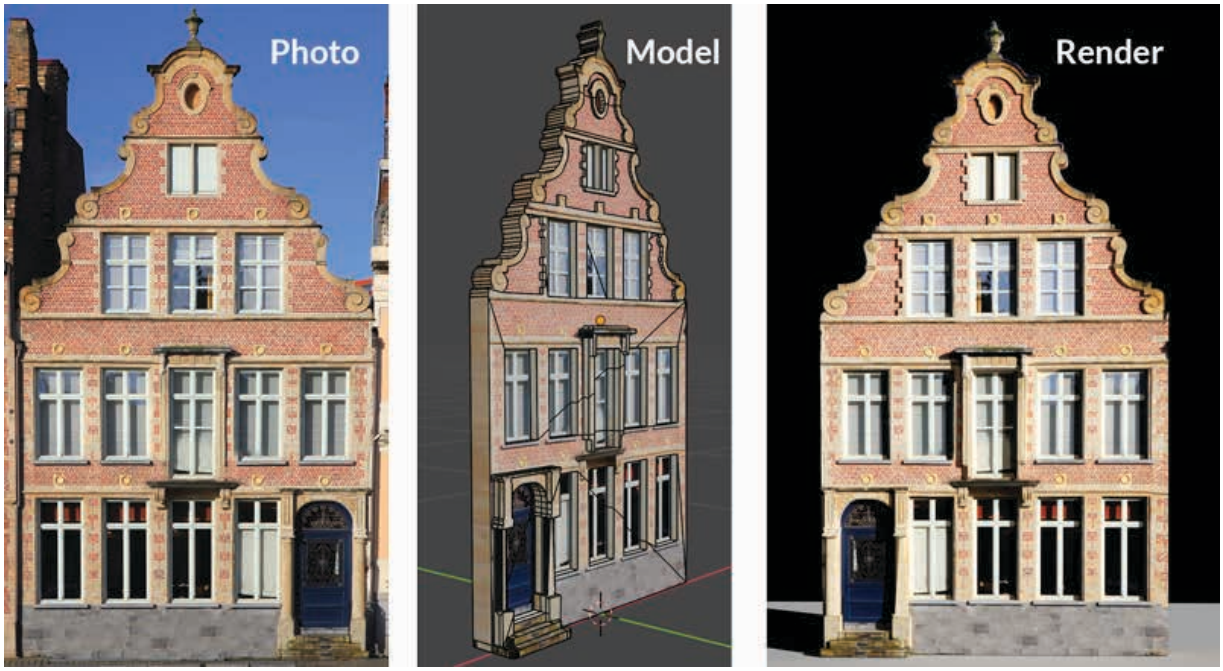
Now that we have a somewhat even distribution of subdivided geometry, give the object a material, unwrap the geometry with **Project from View** (for more detailed steps check the previous chapter).

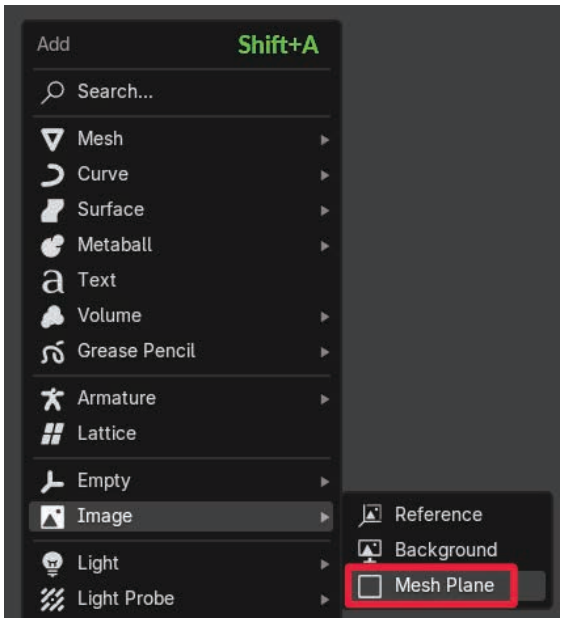
Set the image texture extension to Clip if you don't want the texture to repeat outside of the camera projection region.

That's all! Now you can add some subtle camera movements and add some objects and maybe even some volumetrics to the scene.



Modeling from Photos with the Knife Tool - Basics

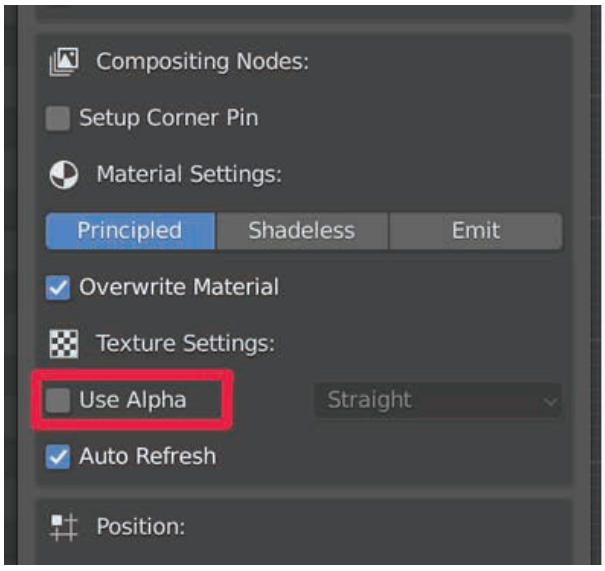


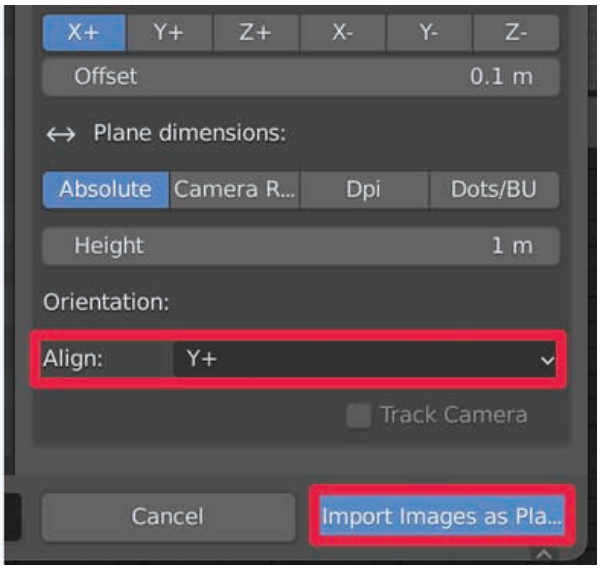


Using the Knife tool, we can quickly turn a photo into a textured 3D model.

Import an image by **pressing Shift-A** and choosing Image > Mesh Plane.

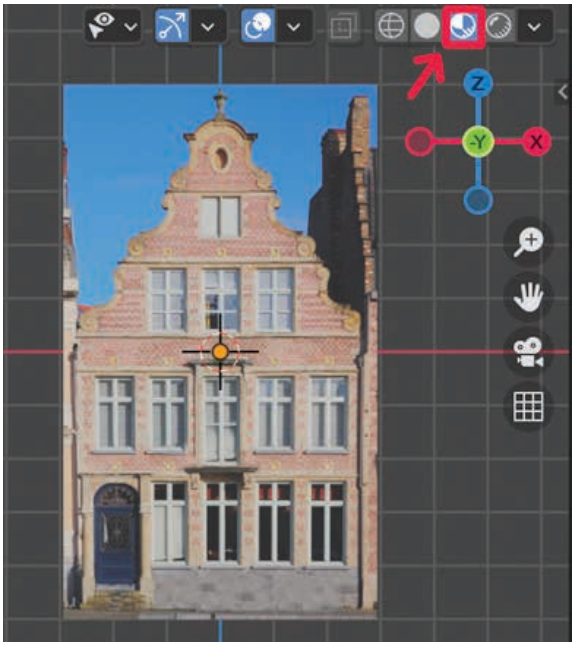
Select an image and in the import options, **disable Use Alpha** and set **Orientation Align to Y+** so we don't have to rotate the plane after importing it.





Click here to return to the index.

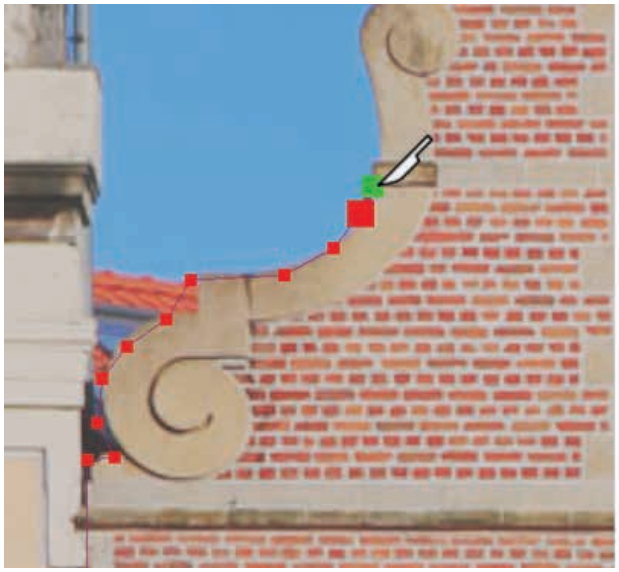
After importing the image, **press Z** and choose **Material Preview** viewport mode so we can see the image texture in the viewport.

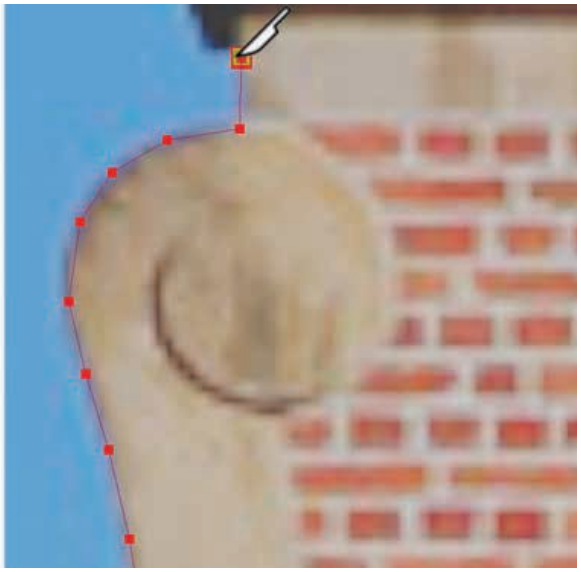


Press 1 on the numpad to see the image in the Front orthographic view.

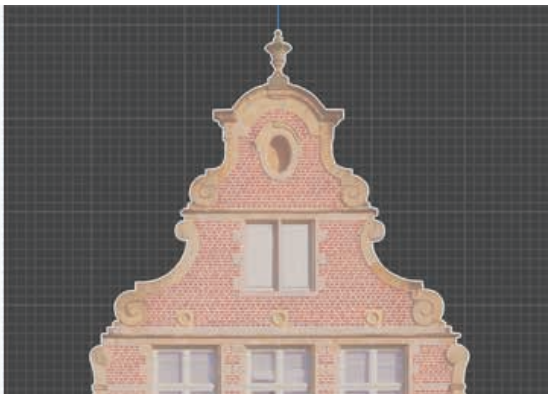
Press Tab to enter Edit Mode and **press K to activate the Knife Tool**.

Cut out the part you want to keep. Scroll the mouse wheel to zoom in or out while cutting.

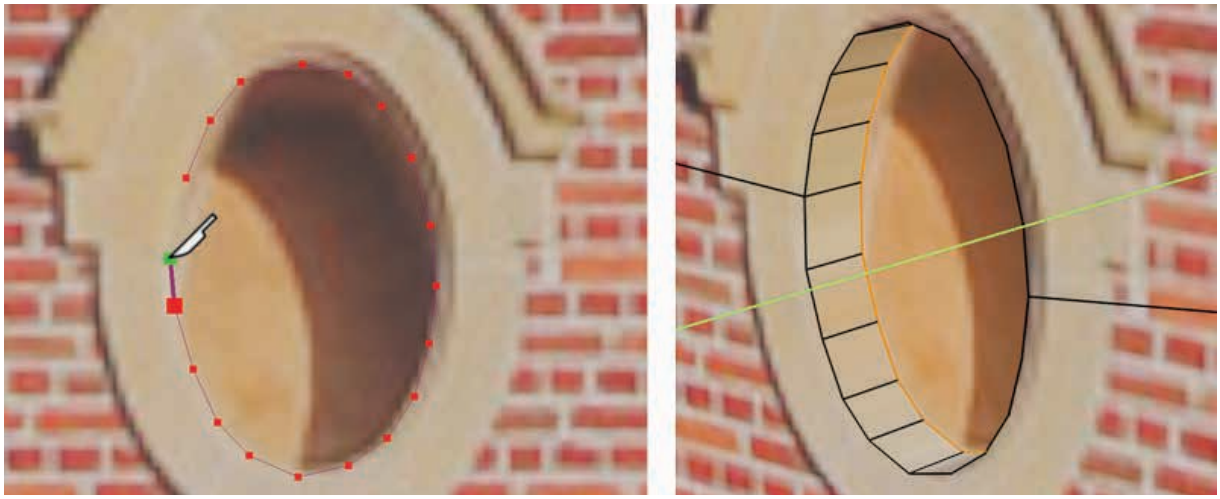




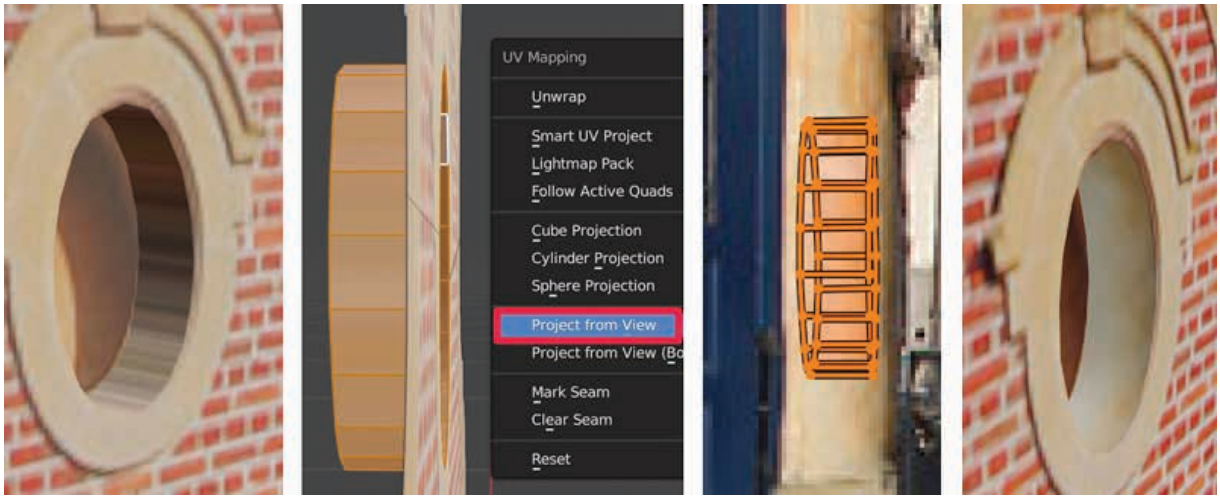
Go all the way around the part you want to keep and close the loop, then **press Enter** to confirm. Select and delete the Faces you don't need by **pressing X > Faces**.



Use the Knife tool to add additional edges for parts you want to Transform or Extrude.



If the side parts look too noticeably stretched, select them and from the side **press U > Project from view**, then move them in the UV Editor to a place where they fit visually.



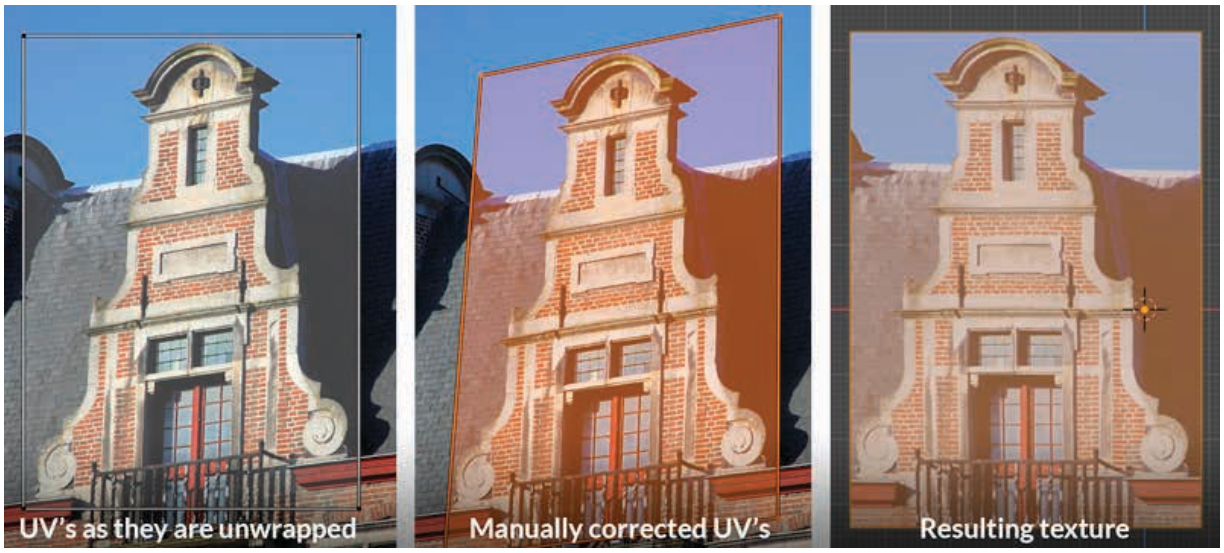
Repeat these steps for each part that you want to give more depth. After cutting out and extruding enough details, it looks pretty nice!

Modeling from Photos with the Knife Tool - Perspective

In this second look at modeling from photographs using the Knife tool, we look at how to use a photo that already has perspective distortion.

As you can see this photo has a bit of perspective. First, I add some loop cuts and remove the parts I don't want.

Drag the UV's in the UV Editor until the perspective in the texture has been corrected to look as if the photo was taken from the Front.



Now you can simply cut out the shape and details as shown in the previous topic about the basics of Knife Tool modeling from photographs.

If you run into a problem like this part where we have a kind of balcony railing, instead of cutting it out (which would be time consuming and difficult) we can just select that area, and move the bottom UV points up in the UV Editor window to remove that area of the texture.

Below we see how the selected UV points in the UV Editor are moved up.

The result is that the model no longer shows the part with the metal railing there. If we want we can now model our own railing and put it there.

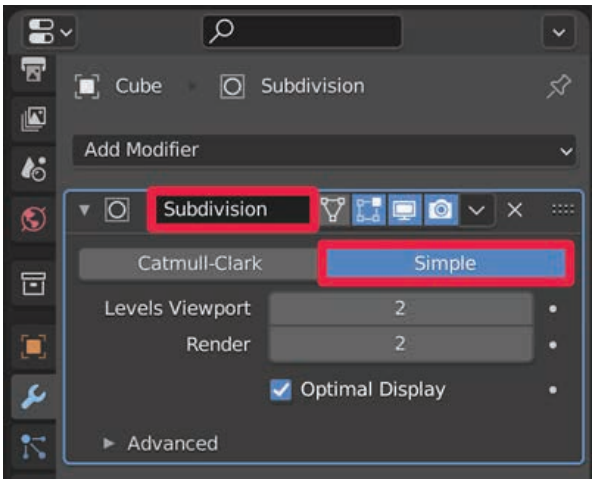
Modeling from Photos - Why is my texture distorted?

This might happen to you: a texture on a low poly mesh looks all wobbly and distorted. What's up with that? The reason a texture gets warbled is because Blender uses the "Affine" texture method, which is computationally efficient but causes distortion whenever the shape of the UV's doesn't match the shape of the corresponding Faces.

You can see in this example that the UV's don't match the shape of the selected Face, causing distortion:

But once the UV's are scaled so that it matched the shape of the Face, the texture is no longer distorted:

This is easily solved by adding more geometry, however you may not want to add extra geometry destructively by adding edge loops, so another way to solve it is by using a **Subdiv modifier** set to **Simple**.

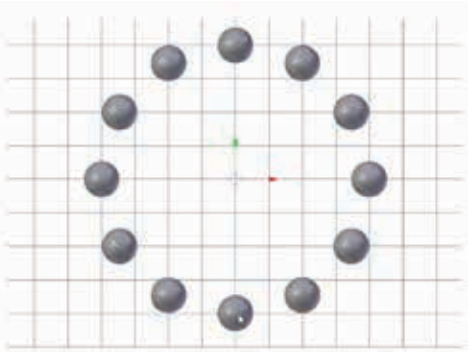


Below you can see the difference: while the geometry remains the same in Edit Mode, you can see the model without the Subdiv modifier set to Simple has a distorted texture (left) and the one with the modifier looks fine (right).

When adding more details, the extra geometry usually solves the texture distortion problem and you won't need the Subdiv modifier anymore. But if your model is meant to stay very low poly, you can bake the texture from the model with the Subdiv modifier to the model without any modifier on it, and the texture will look fine once baked.

Super Quick Circular Array

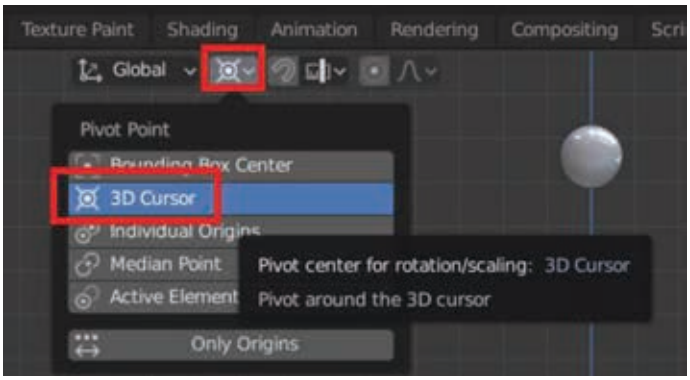
There's no default way (yet) for making an Array **circular** in Blender.



But there's a quick trick that you can use to achieve this anyway.

First, create your first object.

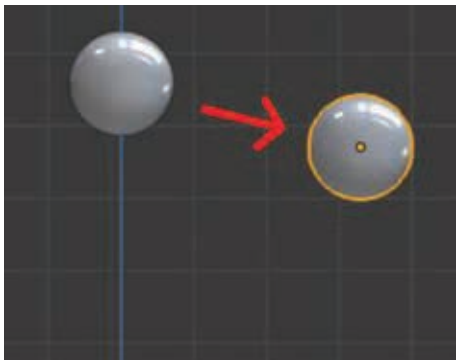
Next, make sure that **Pivot Point** is set to **3D cursor**.



Make sure the 3D cursor and object don't overlap.

The Pivot Point will be the point around which the object is rotated.

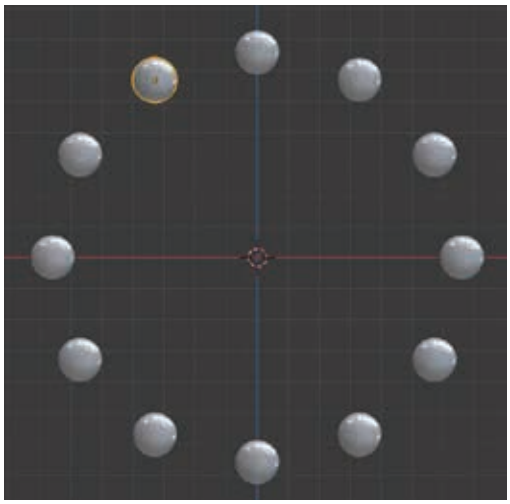
With the object selected, **press Shift+D** to duplicate.



Then **press R + 30 + Enter**. This rotates the duplicate 30 degrees, in this case around

the position of the 3D cursor because we set the orientation that way.

Press **Shift+R** to repeat this action (repeat x 10). You can choose different angles than 30°,

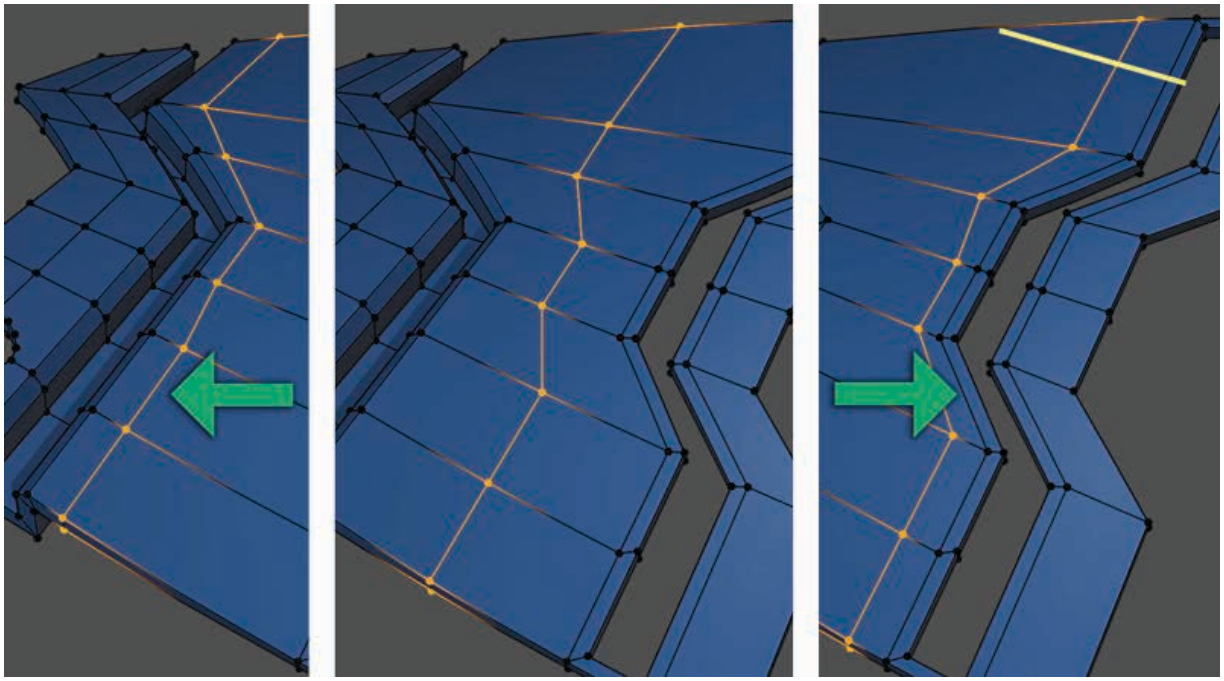


depending on how many duplicates you want of your object. In this case: 360° divided by 30° yields 12 objects.

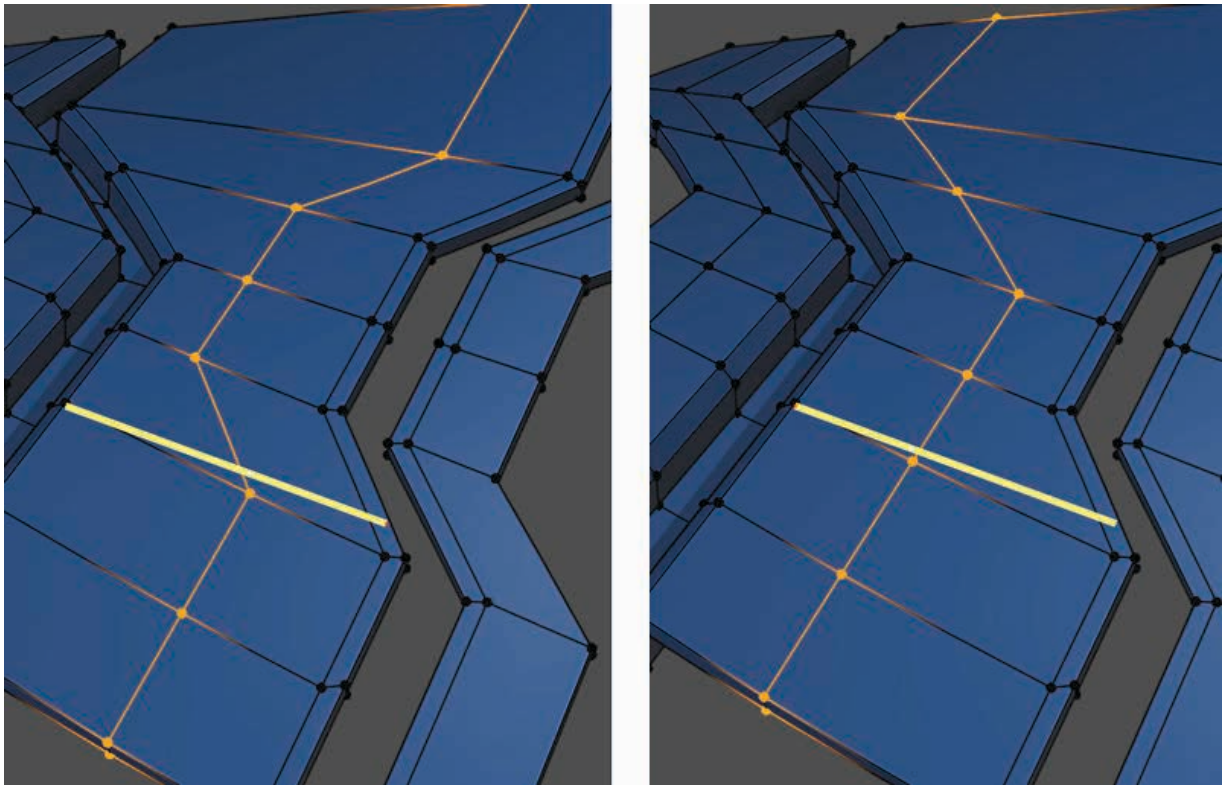
To know the angle you need to rotate around, divide 360 by the number of objects you want in your array. For example, 360 / 12 = 30.

How to use Edge Slide

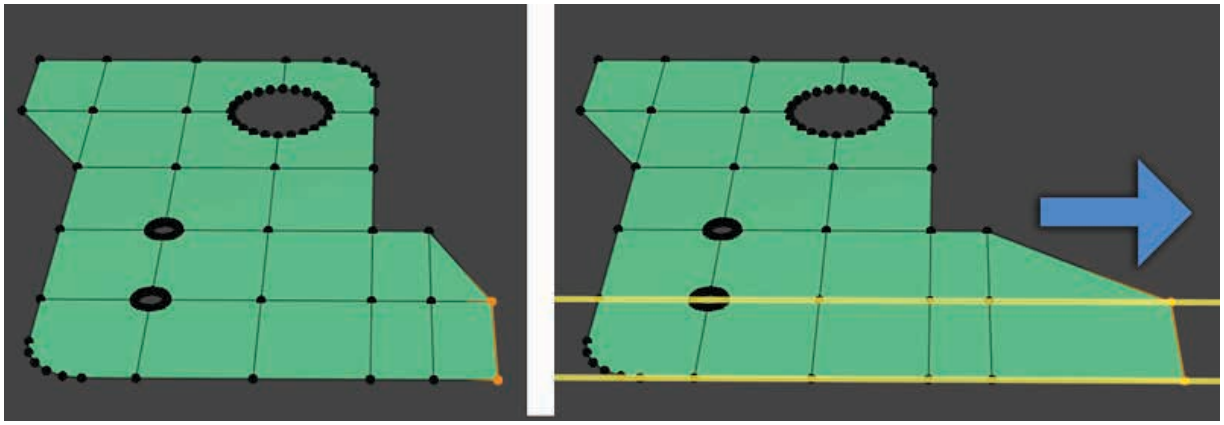
By **pressing G twice**, you can slide a selection. By default, the closer it will get to other vertices, the more it will match that shape. Alternatively you can **press Shift+V** to slide a vertex as well.



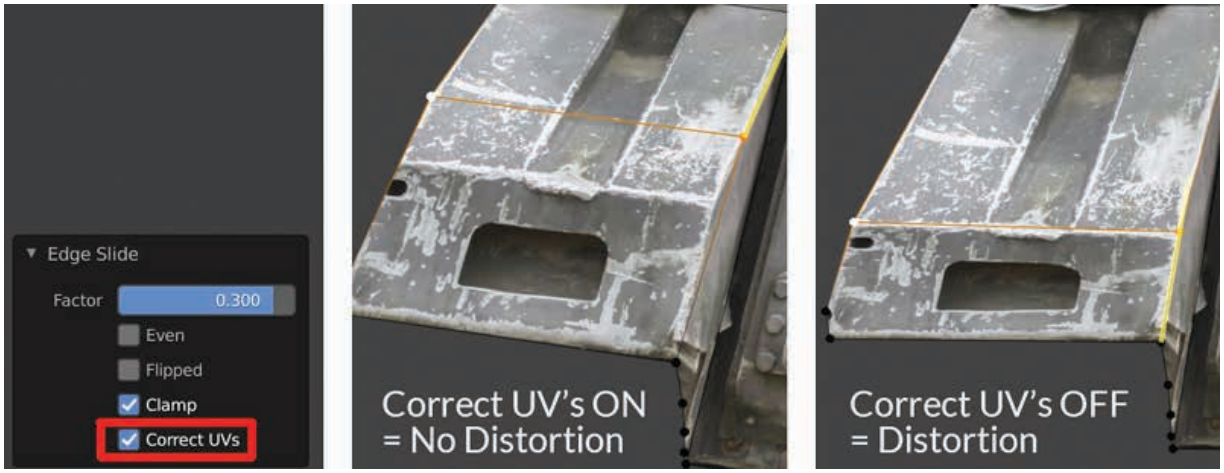
Pressing E will force the selection to match the shape of the adjacent vertices 100%. **Pressing F** lets you Flip between which edge loop to match, and will match it exactly in terms of shape.



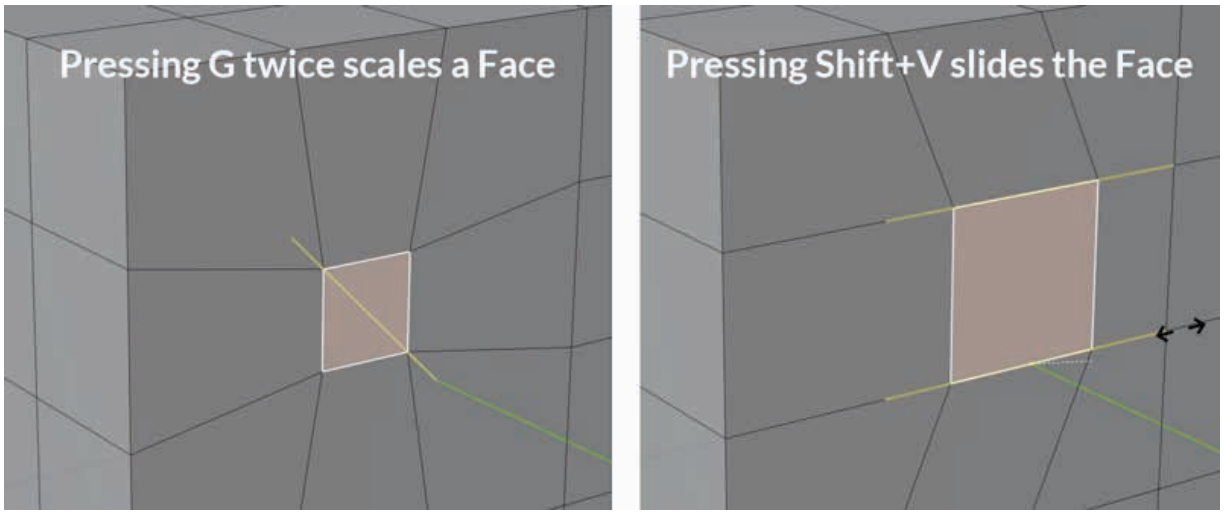
Pressing C lets you go outside of the mesh bounds if needed. After left-clicking to confirm the slide, a menu also pops up with the same options.



The **Correct UVs** option makes sure that the textures don't get distorted when sliding.



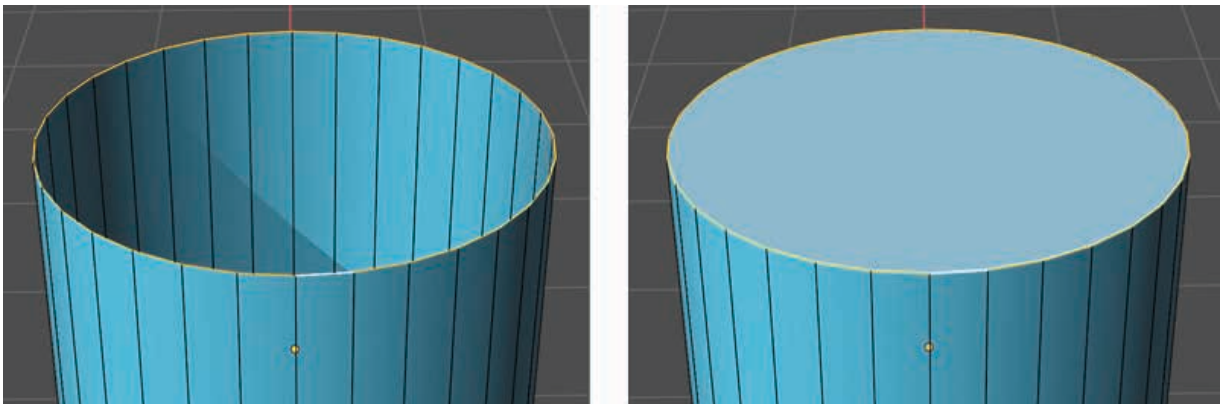
As mentioned at the start, **Shift+V** is an alternative way to slide. In fact, for Faces this is the only method of sliding as pressing G twice scales the Face instead.



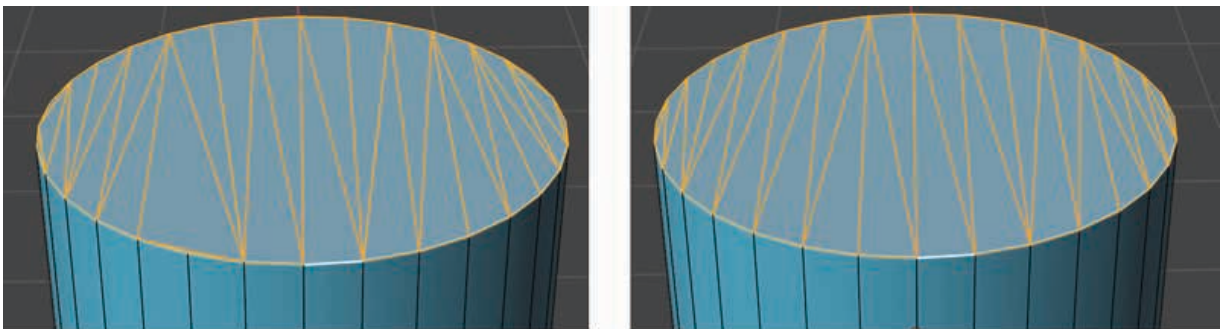
Triangular Fill method

Let's cap this cylinder. Select the loop of edges by **Alt+left-clicking** on an edge. **Press "F"** to fill the hole.

This creates an N-gon or a Face with more than 4 sides. This is usually not ideal as it can cause problems with subdivisions and deformations later.

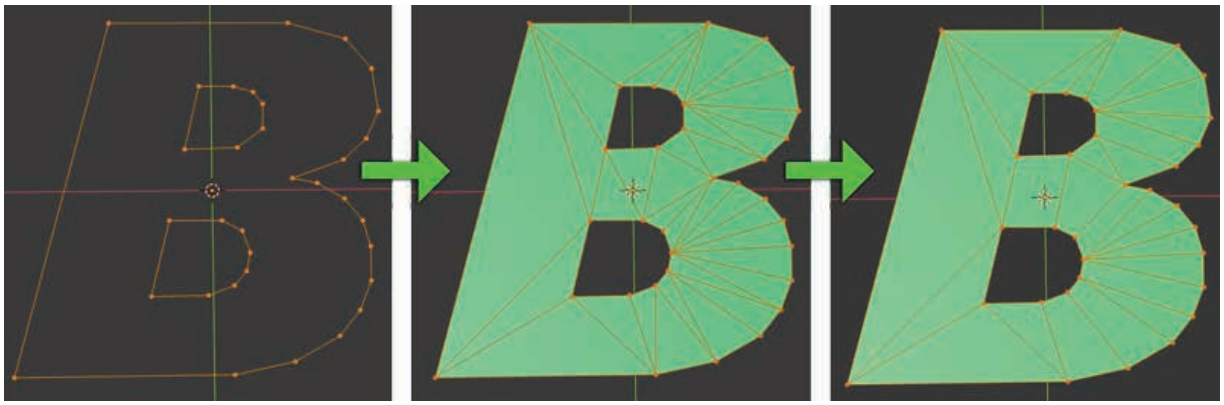


Let's undo that by **pressing Ctrl+Z**. Instead, **press Alt+F** for a **Triangular Fill.** You have one option here: you can enable or disable "**Beauty**". You can try both, to see which one works best in your specific case.



An example of where **Triangular Fill** works really well is for when you have geometry with holes. In this case, enabling the **Beauty** option gives the best result.

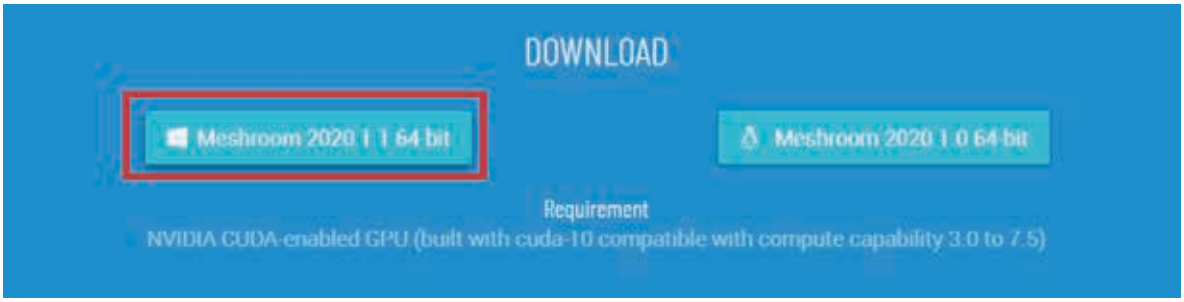
To further improve the geometry, you can select all by **pressing A** and then **press Alt+J** to automatically turn triangles into quads where possible.



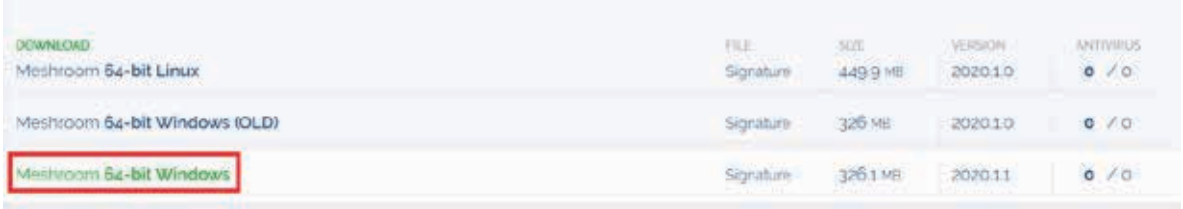
3D Scanning with Meshroom and Blender

With Blender and the free photogrammetry software Meshroom as well as a camera or even your smartphone, you can 3D scan objects. This pumpkin is a 3D model made with Meshroom and rendered in Blender.

On the Meshroom website, click on the latest version to go to the download page.

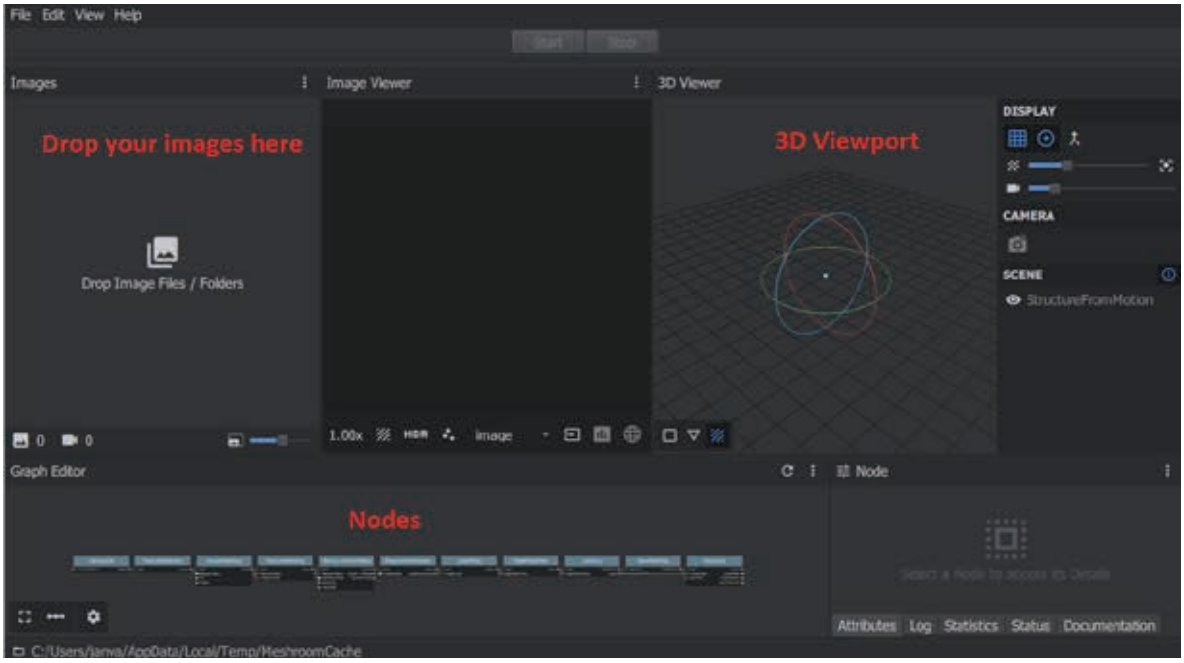


On the download page, choose the version you want. I'm using a Windows system, so I choose the most recent Windows version which is "2020.1.1".



Unzip the downloaded file. Open Meshroom by clicking on **Meshroom.exe** in the unzipped folder. In some cases you may need to give Windows permission to execute the file (this only happens the first time).

You'll get the below interface which actually looks a little bit like Blender: You've got a 3D viewport, a window with nodes, an Image viewer window… you can **zoom in and out in these viewports with the mouse scroll wheel**. This may be necessary in the nodes window. The 3D viewport can be rotated by holding down the left mouse button.



To get started with 3D scanning, you'll need to **take some photographs** of your subject first, from various angles around the object.

Make sure the subject isn't too shiny for best results. If it's wet, dry it with some paper towels. You can also make shiny objects duller with something called "**dulling spray**". This is a product usually sold in photography shops, specifically for making shiny objects appear less shiny on photographs. It can be quite convenient for 3D scanning as well.

(note: please don't use dulling spray on human skin. There are specific tissues for making skin less shiny, which most good photography shops will also have in stock.)

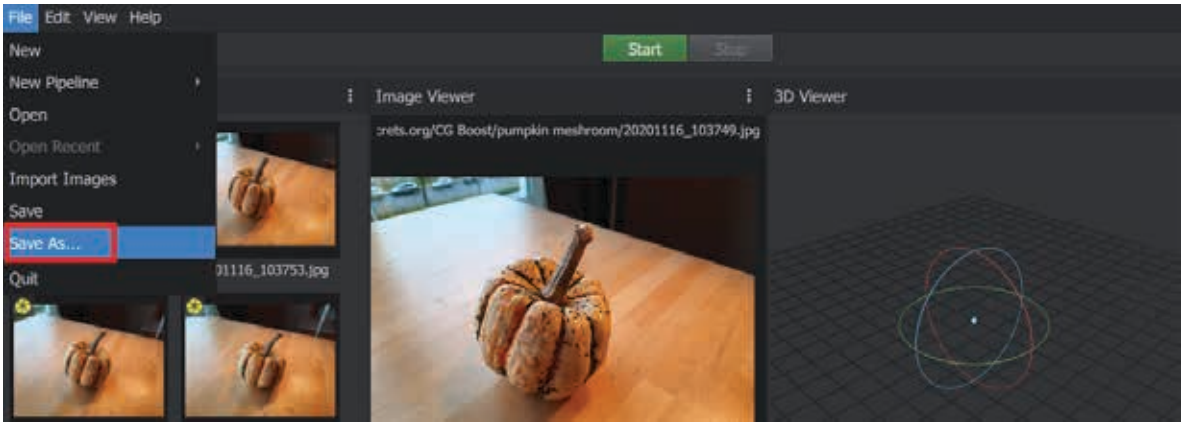
I took about **70 photographs** of this pumpkin, walking around it 3 times and taking photos with my smartphone. If you have a good DSLR camera, I recommend using that instead as the photos will be more consistent.

As you can see from the picture, I have warm artificial light from inside mixed with cold autumn sunlight coming from outside.

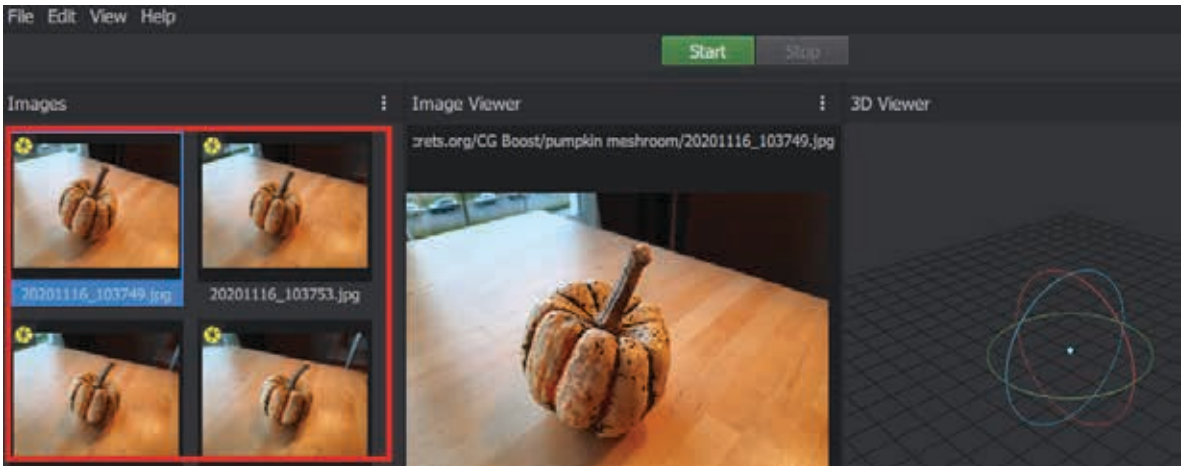
Not ideal! What you really want, is to have diffuse light of the same color temperature (in other words, either daylight or artificial light, not both) all around the object, so that there are no hard shadows or light differences.

At the very least, make sure there's **enough light**. If I hadn't added that lamp it had just been really dark on the left side. And make sure that everything is in focus.

Once you've copied your photographs from your phone to your computer, **drag them into the Images window in Meshroom**. Then save the project somewhere by going to **File>Save As**.



I recommend using **at least 50 photographs**, but not too many - it may give a better result, but it will take very long.

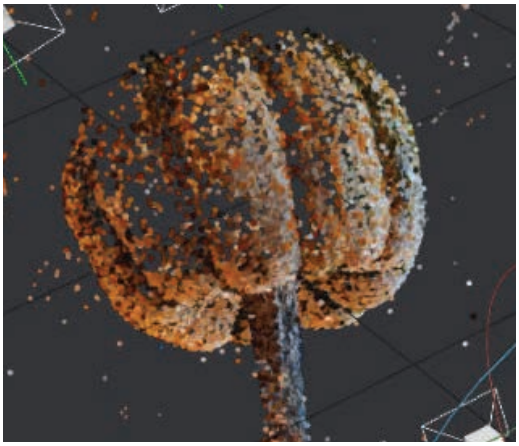


The default settings of Meshroom are pretty good. I'd recommend selecting the Texturing node and **unchecking the UDIM option** (as UDIMs are not really needed for a simple 3D pumpkin in my opinion).

Then all you need to do is press Start! (and wait… and wait some more…) Seriously though, I'd recommend hitting the Start button just before you're going to do something else anyway like eat dinner, file your taxes, walk your dog, reflect on the choices you've made in life... it will take a while.

That's the downside of Meshroom.
I often use Reality Capture (a paid photogrammetry app) when I don't feel like waiting, as it's much faster (but you pay a little bit for each export).

After being patient you get a point cloud:



To get the mesh model to show up rather than just the point cloud, **double-click on the Texture node**.

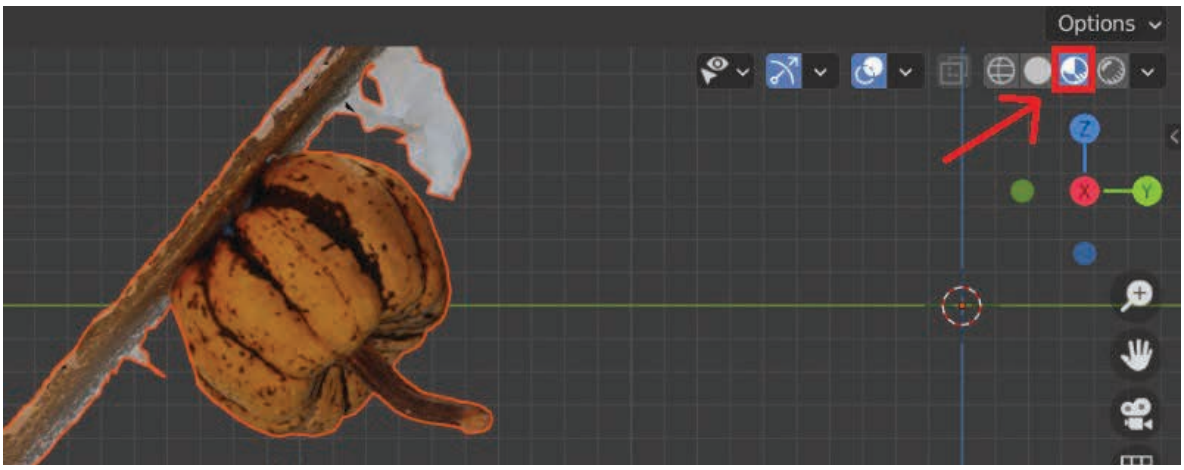
Now you see the mesh as well as the point cloud.

Looks like a successful 3D scan, despite the quality (or lack thereof) in the smartphone images.

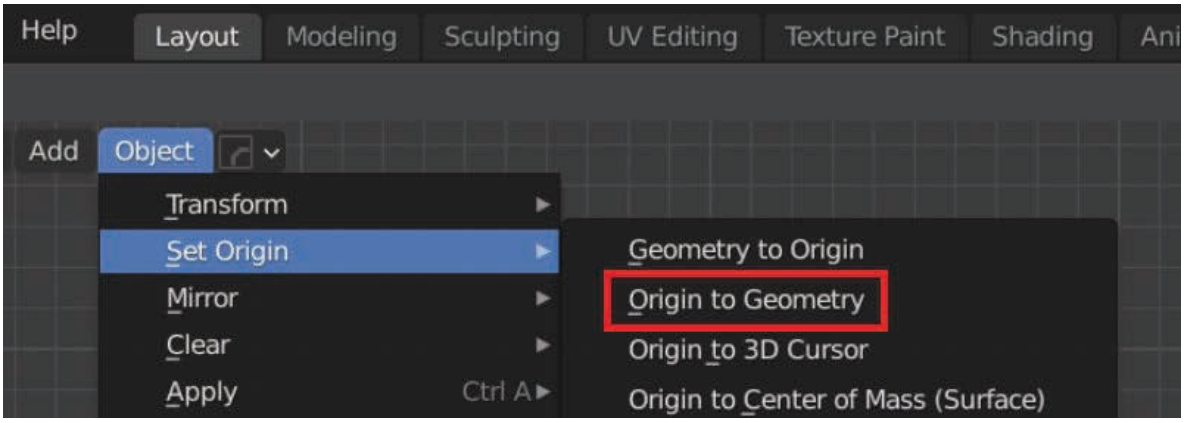
Now, the good news is that you don't need to export anything. Meshroom has already automatically exported an OBJ file, as well as a texture. These can be found in subfolders of where you saved the project file initially.

If you go to your project folder you'll find a new subfolder called **Meshroom Cache**, which contains a subfolder called **Texturing**. That's where you'll find both the mesh (texturedMesh.obj), the material file (.mtl) and the images textures.

Go to Blender and Import **texturedMesh.obj** by going to **File > Import > OBJ**. Turn on Material view so you can see the texture.



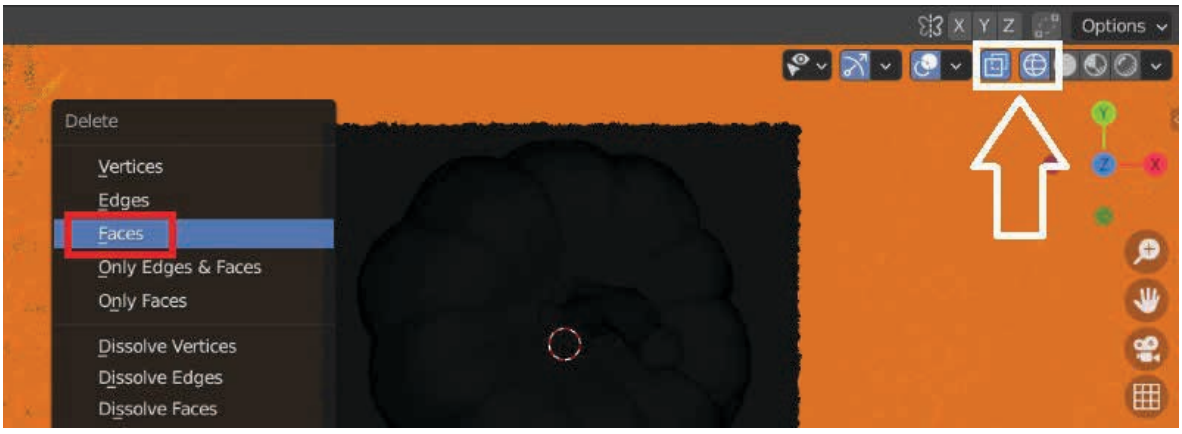
Looks good, we just need to rotate the imported object a few times to get it oriented correctly. This is done easily in the orthographic views. But first, **set the origin to the geometry**.



Rotate the pumpkin from the front and side view until it looks correctly oriented.

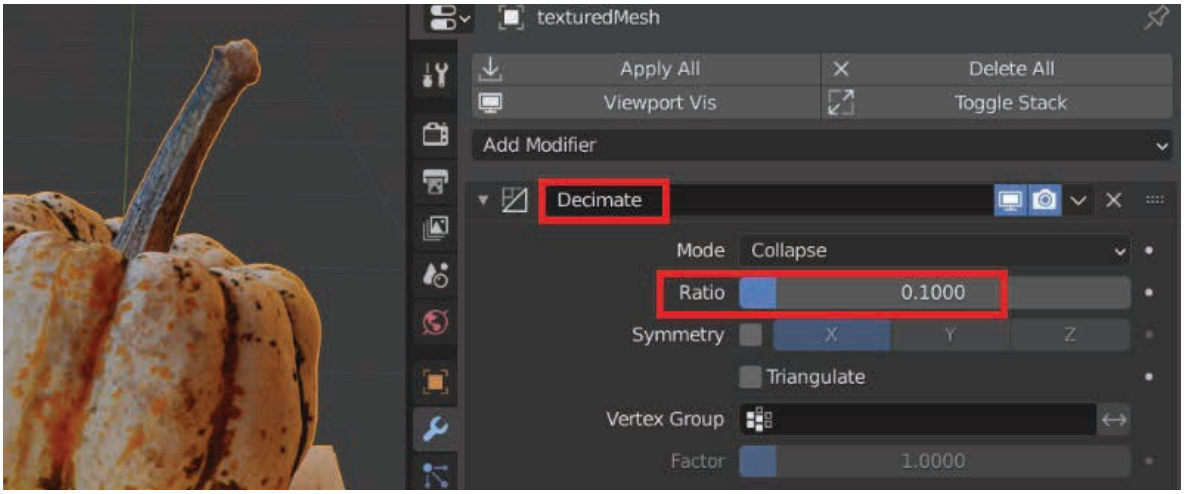
That's better, but now it becomes clear that there's some additional geometry that we don't need.

Go to Top orthographic view and in **X-ray and Wireframe mode** select just the pumpkin and a bit of the ground plane it rests on. Then **press Ctrl+i** to invert the selection, **press X and choose Faces** to remove those faces.



Now we're left with just the pumpkin and some of the table underneath it.

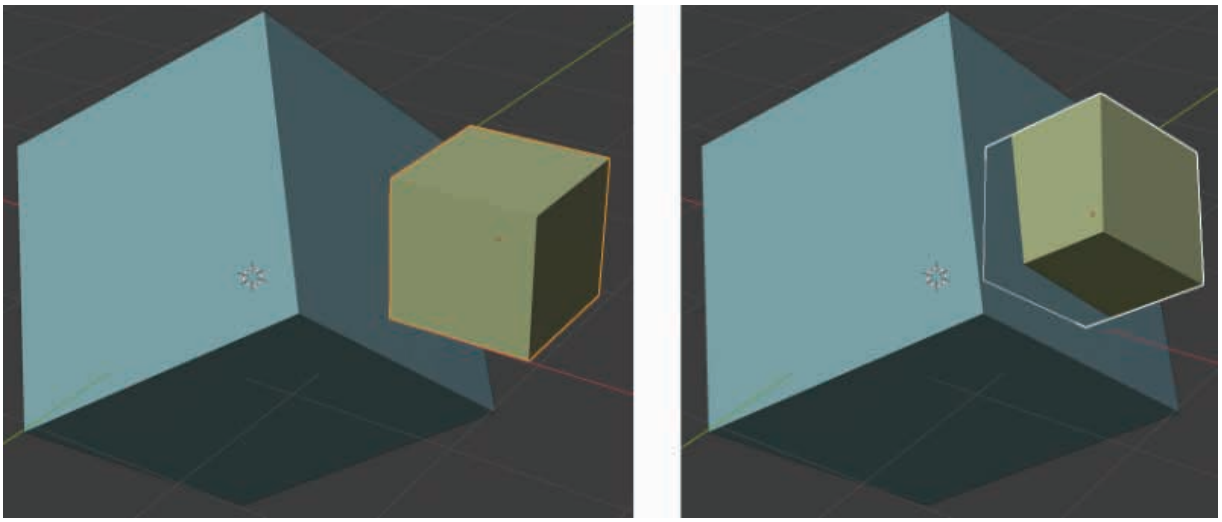
In case the pumpkin has too many vertices and it's making your computer slow down, you can use a **Decimate modifier** on it. I tend to go with **0.1** as the **Ratio**, with good results. You can turn on "Statistics" in the Viewport Overlay options to see how many vertices you have.



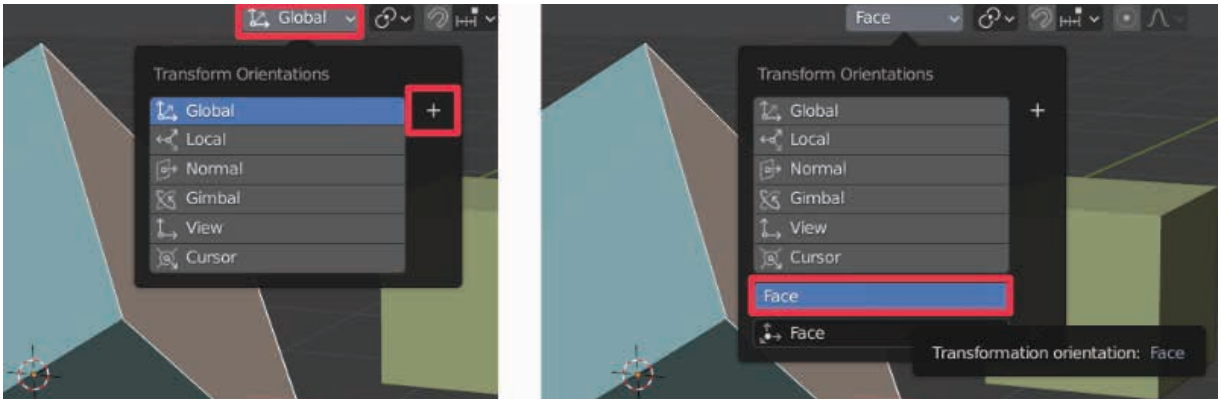
In this case the pumpkin goes from 738.726 triangles to 73.871 triangles, without really any visible difference. I could probably decimate it some more if I wanted to before noticing any real quality loss.

Custom Transform Orientation

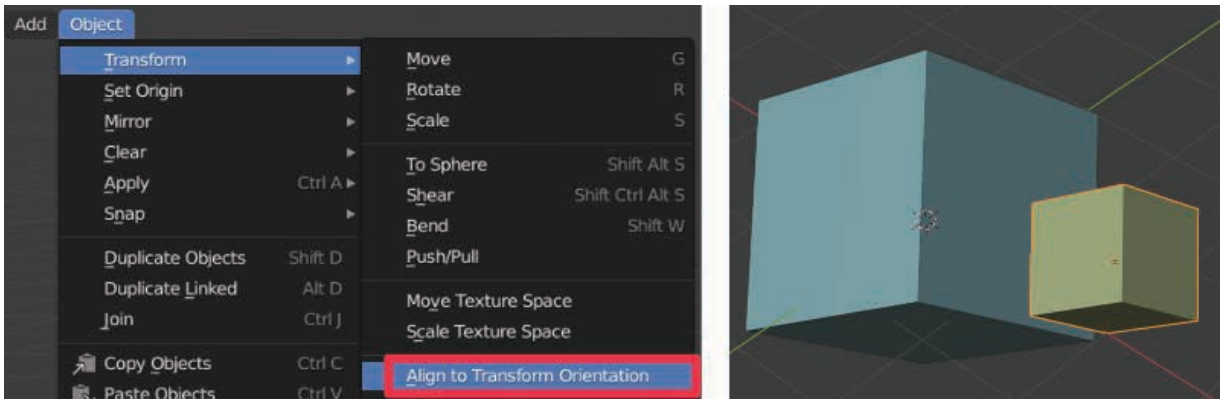
Let's say you want to slide this cube along the surface of this other cube. Manually rotating one cube to match the other is impractical. So is moving it along the surface accurately.



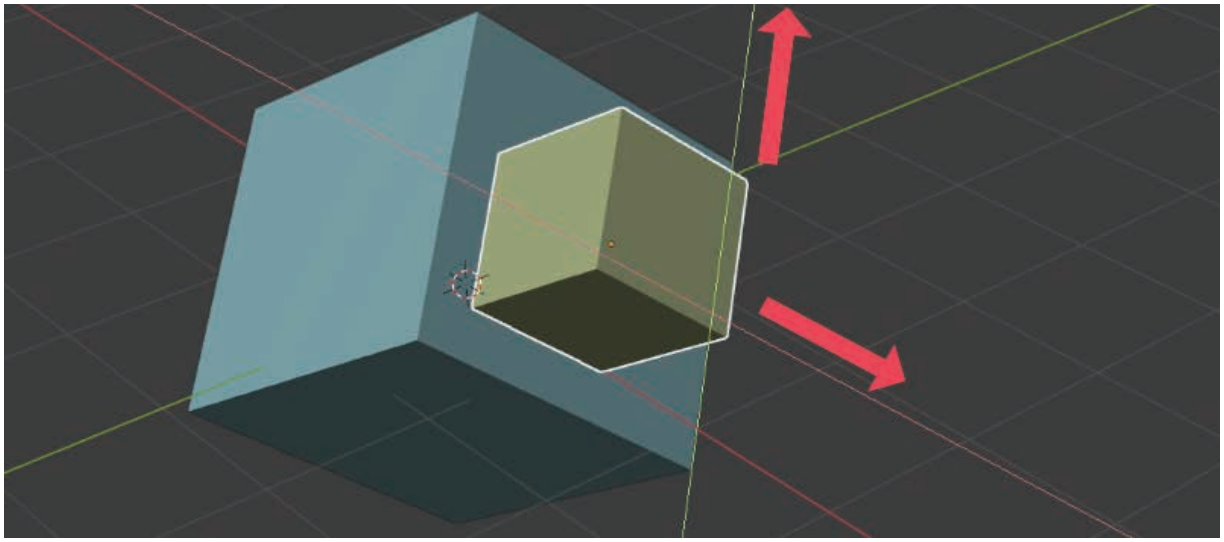
For this you can create a "custom transform orientation". In Edit Mode **select a face** as the basis of the orientation. In the Transform Orientations menu **click on the + icon**. Give your new transform orientation a name if you want (by default it will be named “Face”).



Exit Edit Mode by pressing Tab and select the other object. We can align its rotation to the other cube now by going to Object > Transform > **Align to Transform Orientation**.

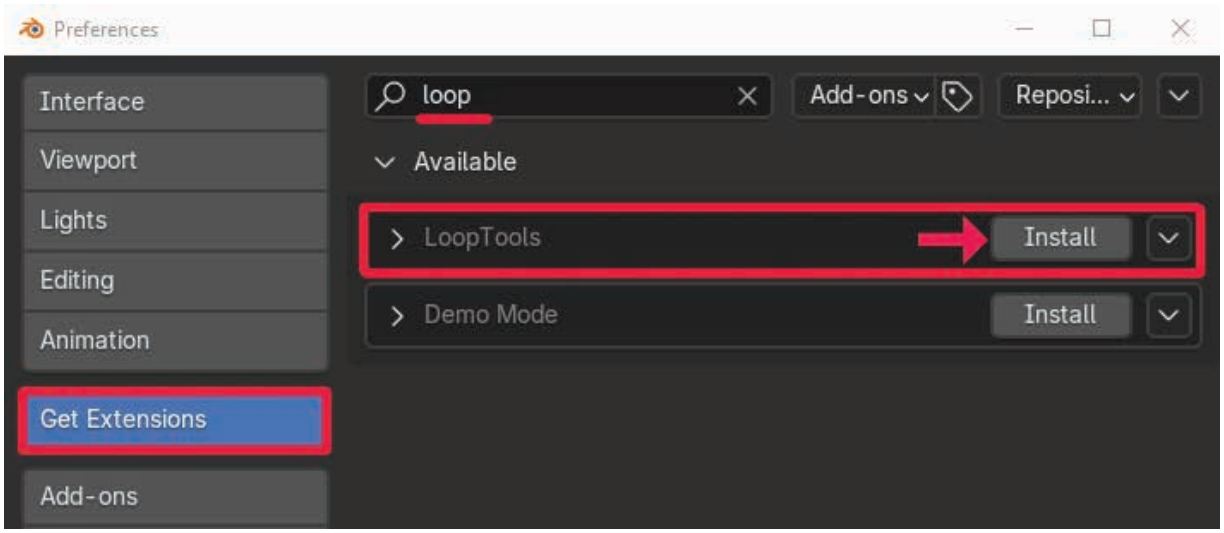


Now **pressing G and then X or Y** will move the object along the X or Y-axis of that particular transform orientation. **Press G then Shift+Z** to exclude the Z-axis.



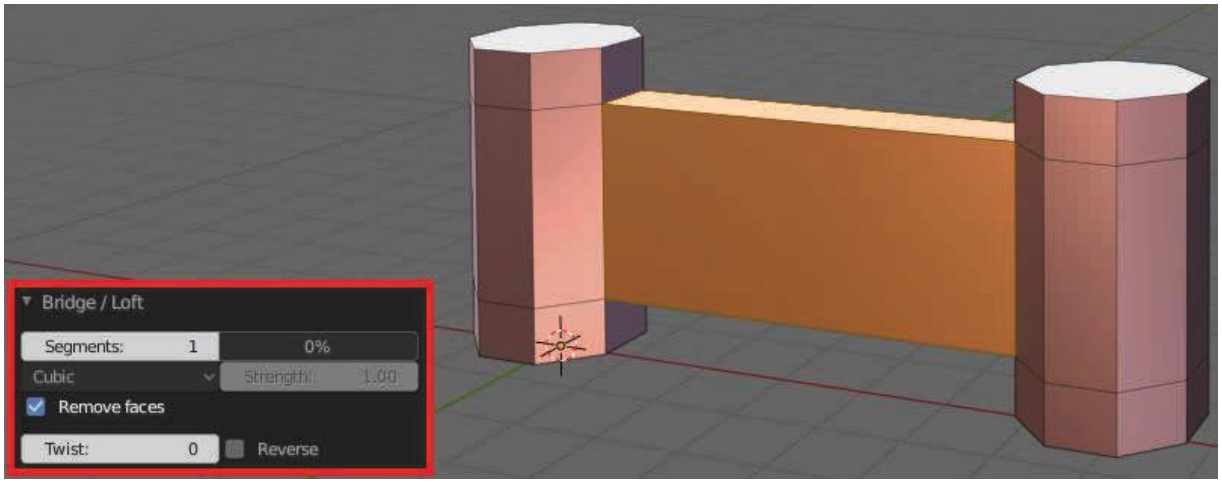
Twisted Bridge With LoopTools Add-On

For this, you need to activate the **LoopTools add-on**, which comes standard with Blender. Search for it in the **Get Extensions** section of the Preferences and click on **Install**.

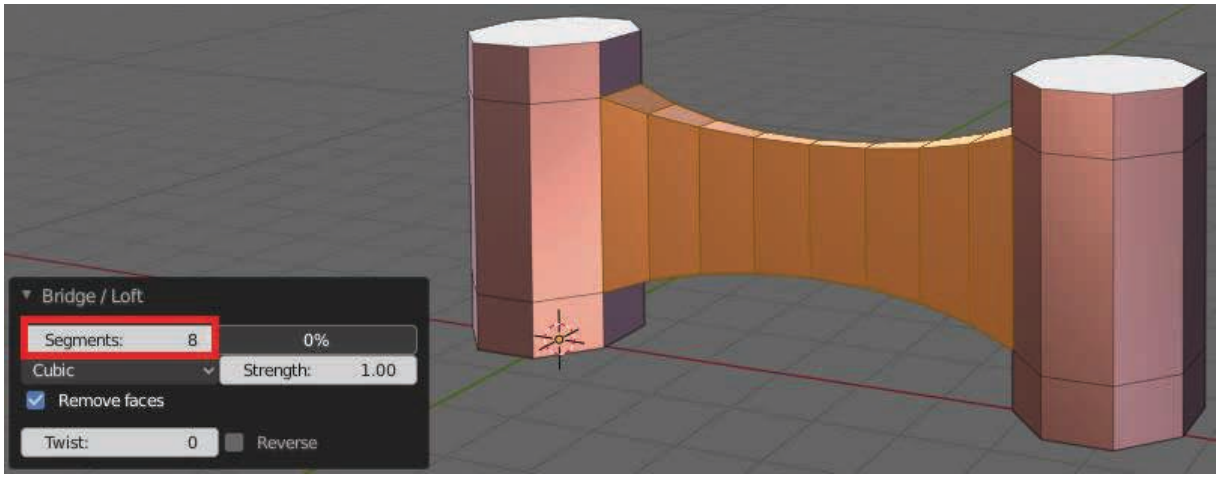


In Edit Mode select two opposing faces and right-click, select **LoopTools > Bridge**.

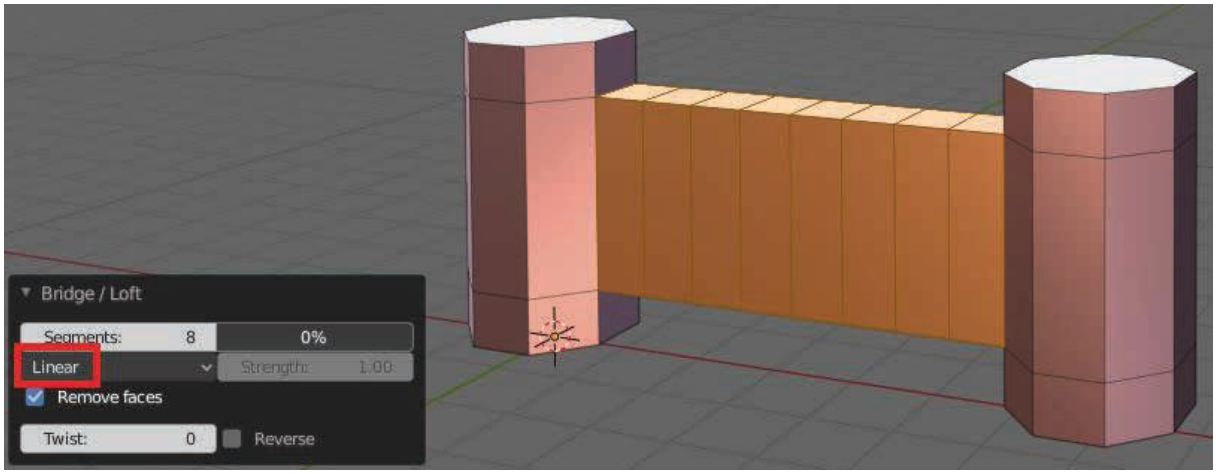
This creates a bridge between the two selected faces, and opens the Bridge menu.



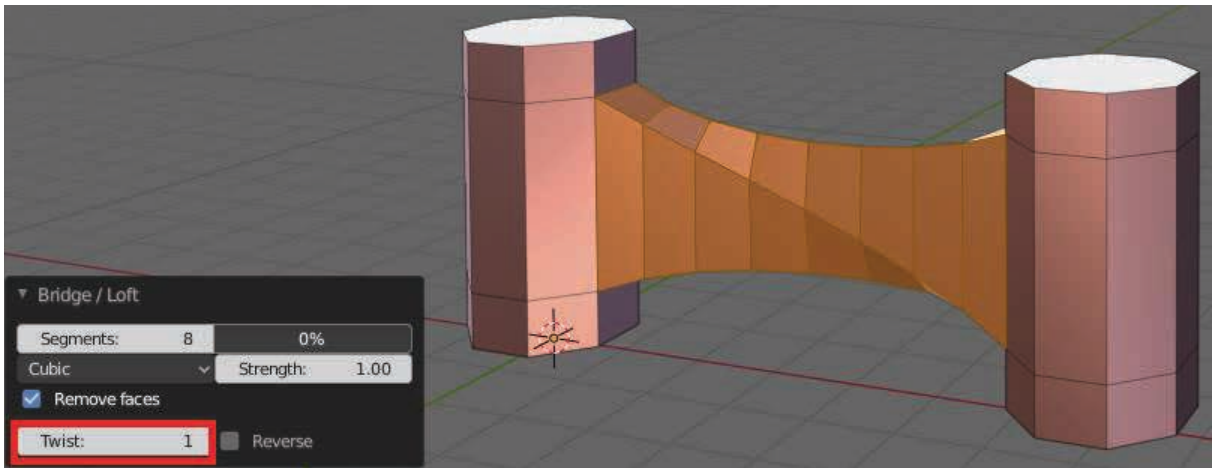
You can now choose how many segments the bridge has, and add a Twist value. If you set the interpolation type to Cubic, the bridge will be a more rounded or curving shape.



If you set **Interpolation type** to **Linear**, the bridge will just be straight without any curving.



You can increase the **Twist** value to make the bridge twist around itself. Here it is set to Twist value 1, with **Interpolation type Cubic** to give it some curve.

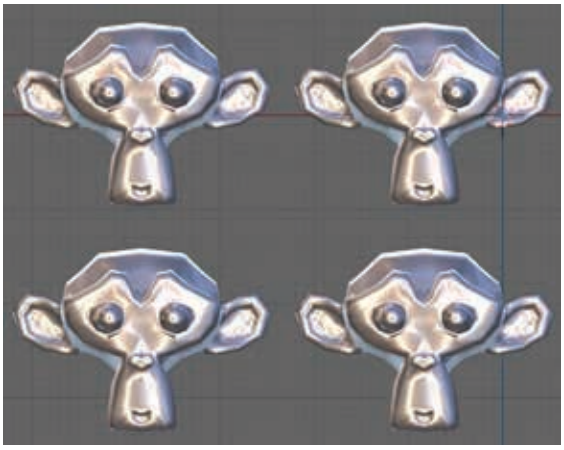


This could be interesting for making a piece of furniture like a chair for example.

Linked Duplicates

Instead of duplicating with **Shift+D** you can duplicate with **Alt+D** instead. This has big advantages in some cases.

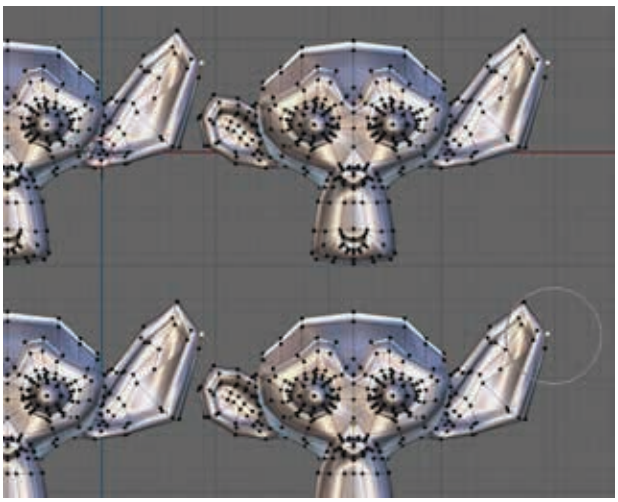
Here I duplicated a monkey by **pressing Alt+D.**



You will notice that if you go to Edit Mode, all the duplicates are also in Edit Mode.

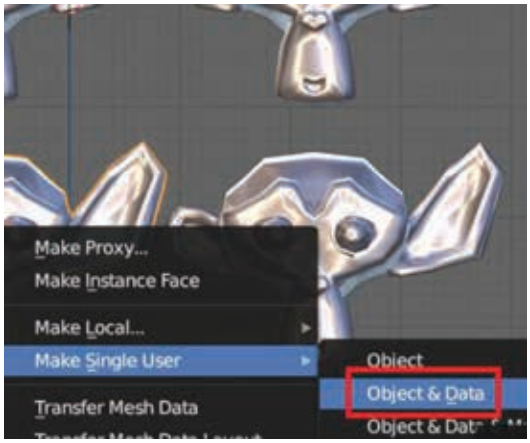
The duplicates are “linked”, which means that if you edit any of them in Edit Mode the others will update as well.

This is great if you have a lot of similar objects and they all need the same changes. It saves a lot of time.



To make an object “Single User” or unique, you can go to the **Object Menu > Relations > Make Single User > Object & Data.**

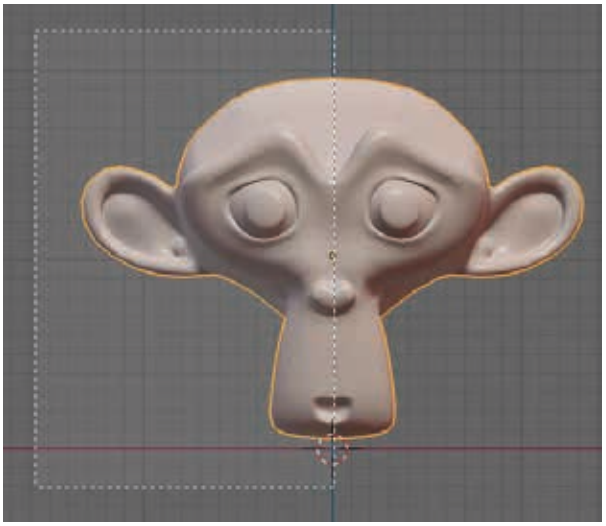
Now that object can be edited individually without affecting any other objects.



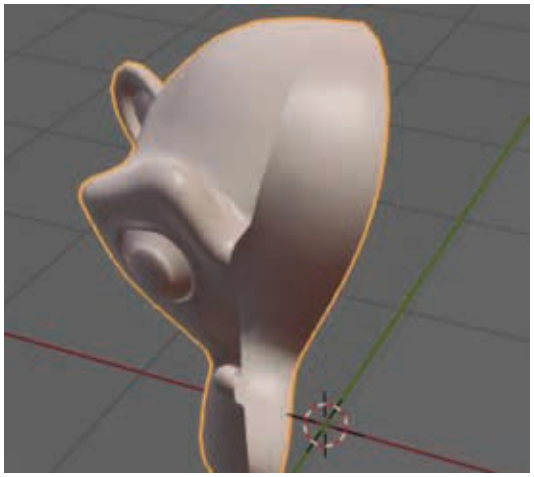
Clipping Border

To edit some things inside another model (like the inside of the eyeballs here) you can use “Clipping Border”.

Simply draw a box selection with the keyboard shortcut **Alt+B**.



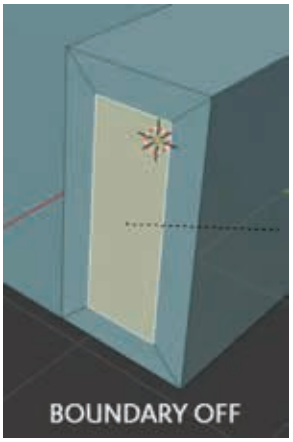
As you can see the rest of the model is now hidden. Press **Alt+B** again to reveal it.

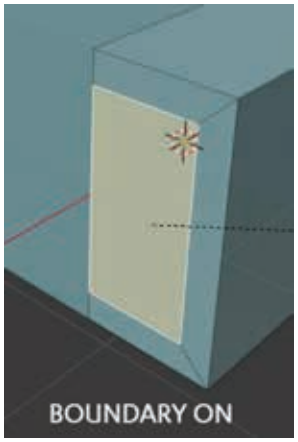


Inset Boundary

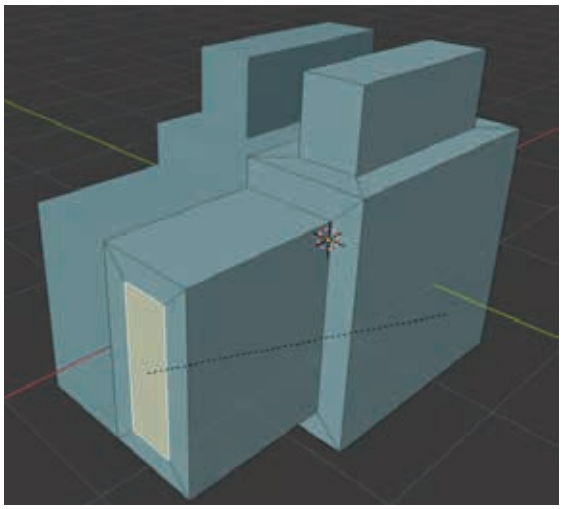
There are several ways to inset a face. **Press “i”** to get the regular inset, which is the same as extruding and scaling down, without changing the location.

But to get the inset to respect a Mirror modifier, you need to use the **Boundary** option by **pressing “b”** after your inset.

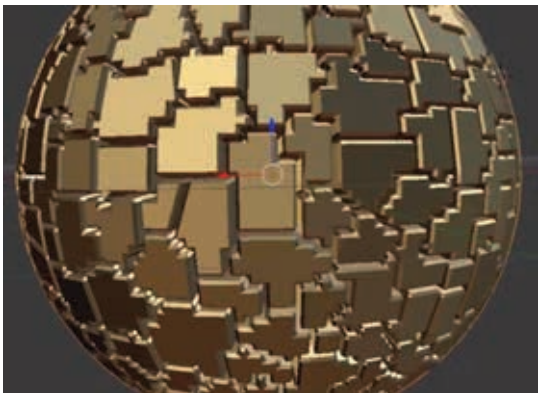




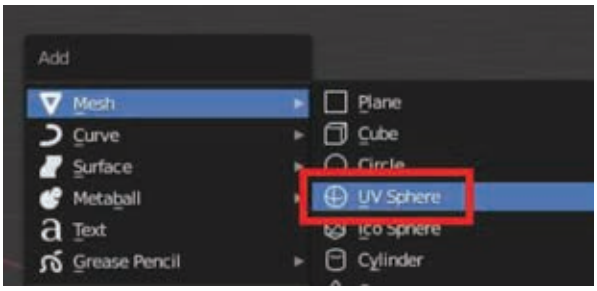
You can toggle Boundary on/off as needed by **pressing“b”**.



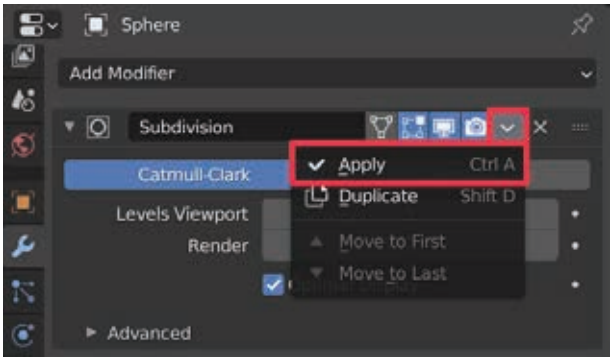
Limited Dissolve (Make a Sci-fiSphere)



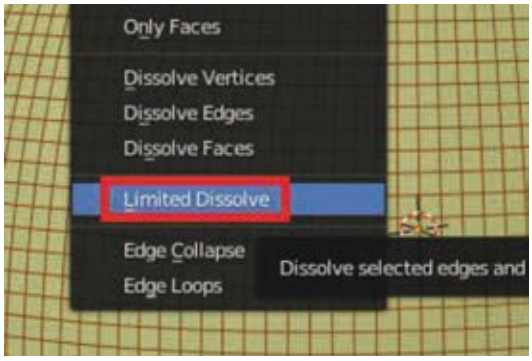
Let's make this… thing! Start with a Sphere. **Press Shift+A**, select **UV Sphere**.



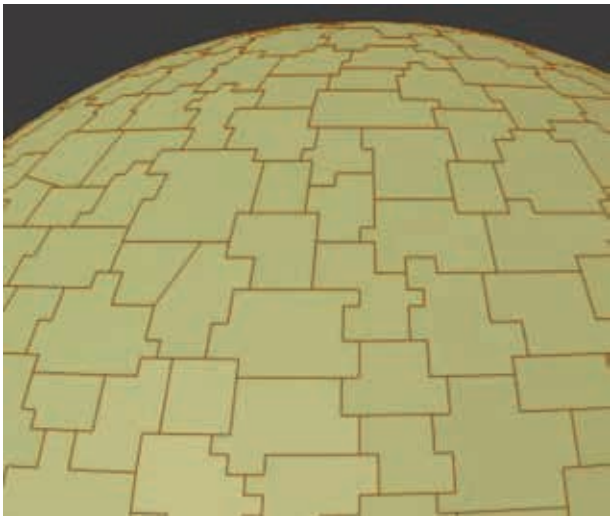
Add some subdivisions by pressing **Ctrl+3**. This adds a Subdiv modifier with 3 subdivisions. Apply the modifier.



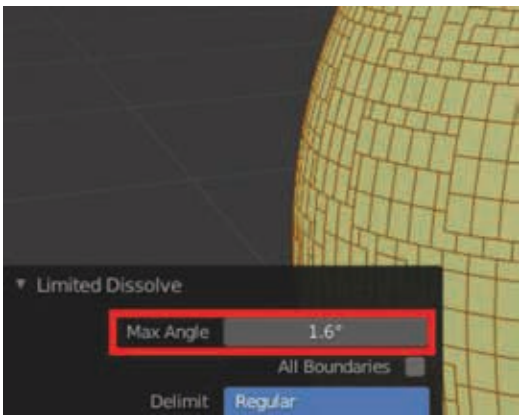
In Edit Mode select all and **press X** and choose **Limited Dissolve** from the options.



This creates an interesting pattern.

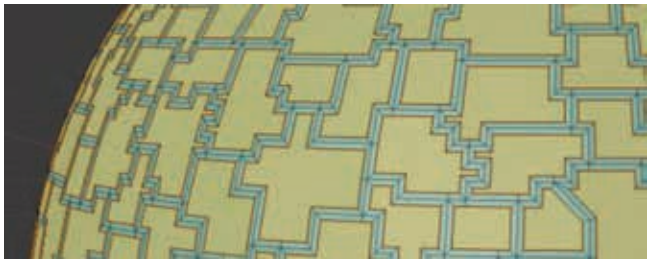


The Max Angle value controls the way it looks. I think the default value of 5° is pretty good.

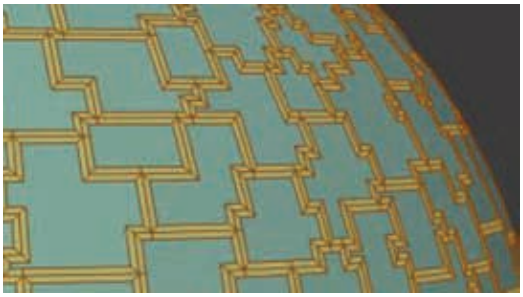


Click here to return to the index.

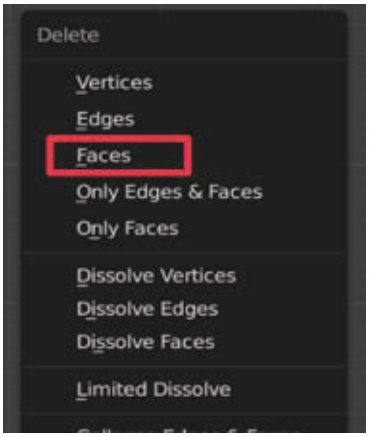
Let's have some fun! **Press i twice** to inset the individual faces.



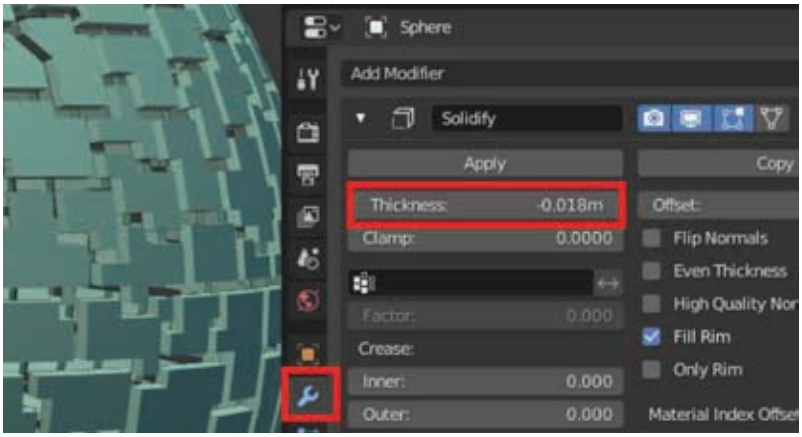
Press Ctrl+i to invert the selection.



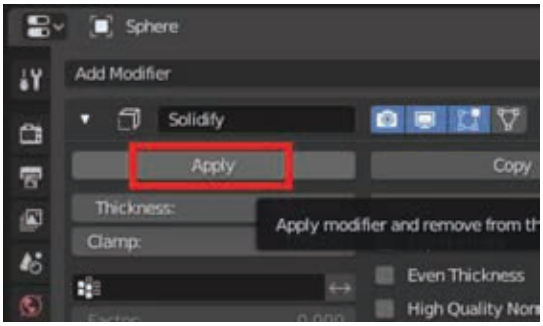
Press x to delete the selected faces.



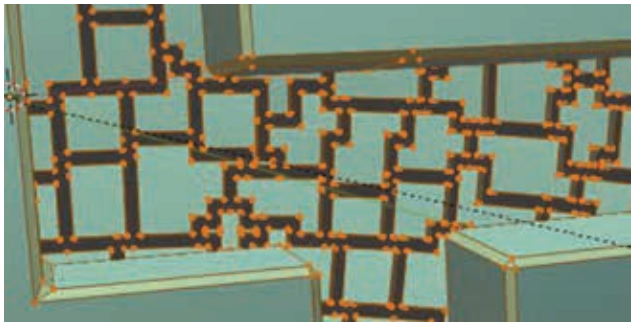
Add a **Solidify** modifier to add thickness to the remaining faces. Adjust the **Thickness** value to your liking.



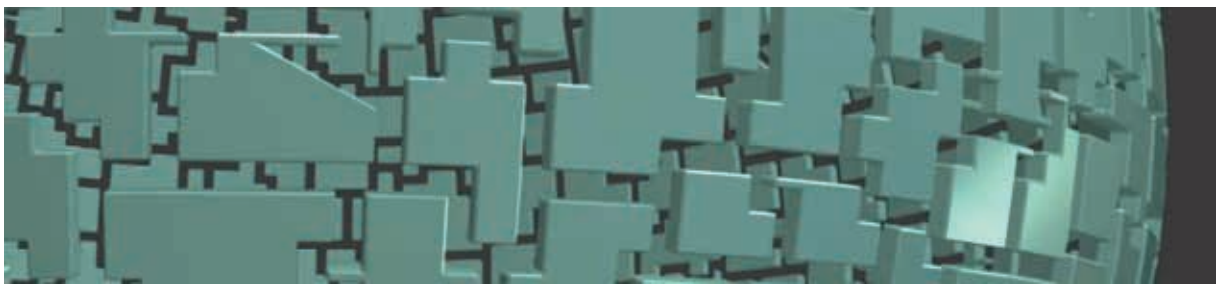
In case you want a slight bevel, apply the Solidify modifier in Object Mode by clicking on**Apply**.



Then select all faces and **press Ctrl+B**. Hold down **Shift** to get more precise control as you make the bevel profile.



Interesting shapes! And all done without really modeling anything. Note: I'm using a **Matcap** to make the object a bit more shiny in the viewport.

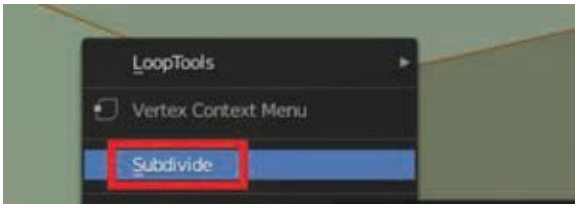


Cast Modifier (Or How To Make Another Weird Sphere)

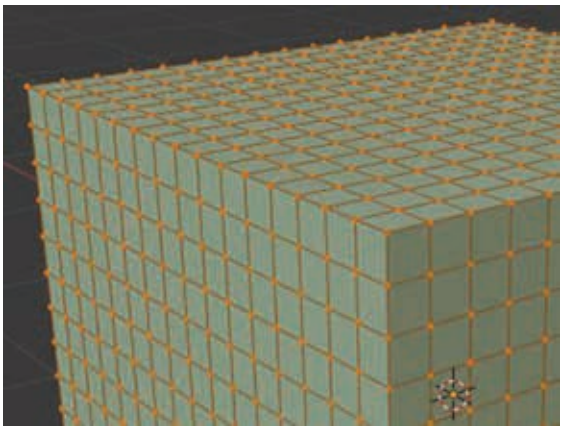
Modeling without actually pushing around any polygons or vertices is fun too. As another example, let's make this sphere.

In Edit Mode subdivide your default cube a few times. Right-click and choose **Subdivide**, then **press Shift+R** (repeat last action) 3 times.

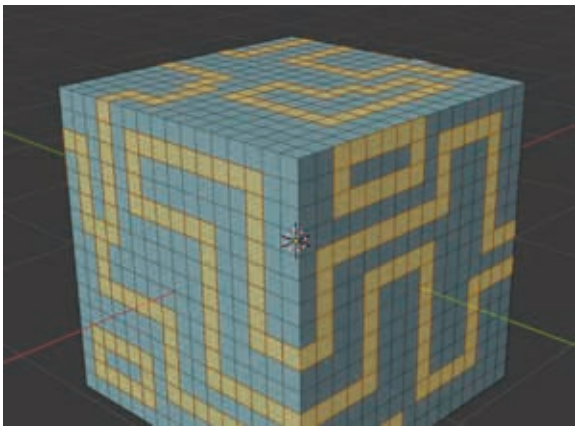
The reason I'm not using a Subdivision modifier for this (like I did in the previous tip) is because I don't want the cube to be smoothened.



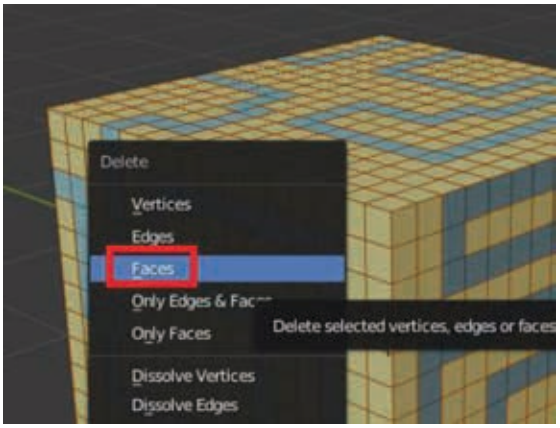
Now it should look like this:



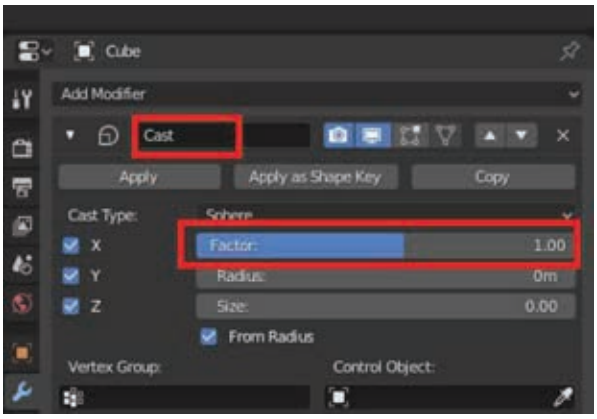
Select an interesting pattern of faces.



Select an interesting pattern of faces. Invert the selection and delete the other faces.



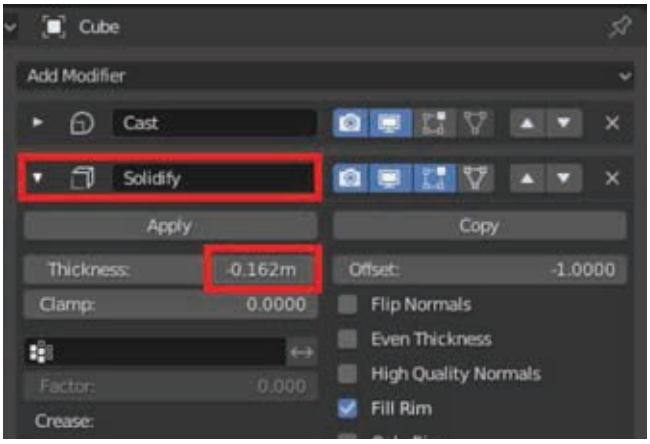
TAB back into Object Mode, add a **Cast modifier,** set it to "Sphere". Set the "Factor" value to 1 (completely spherical).



Click here to return to the index.

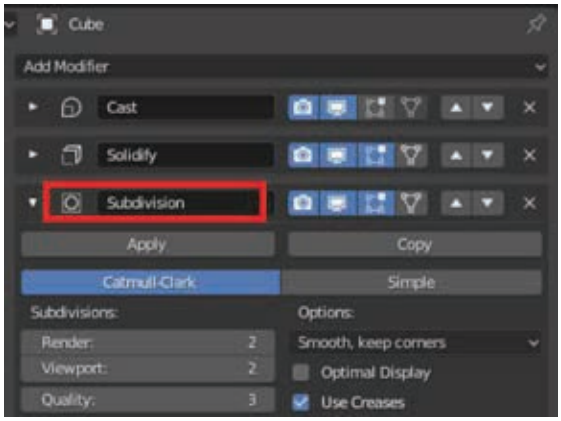
Also, play around with higher values than 1 and -1, to see some weird shapes!

Add a **Solidify modifier** to add thickness. Make sure the Cast modifier is above the Solidify modifier otherwise it won't work.



Experiment with the Thickness value until you get something you like. You don't need to copy my values.

Add a **Subdivision modifier**.

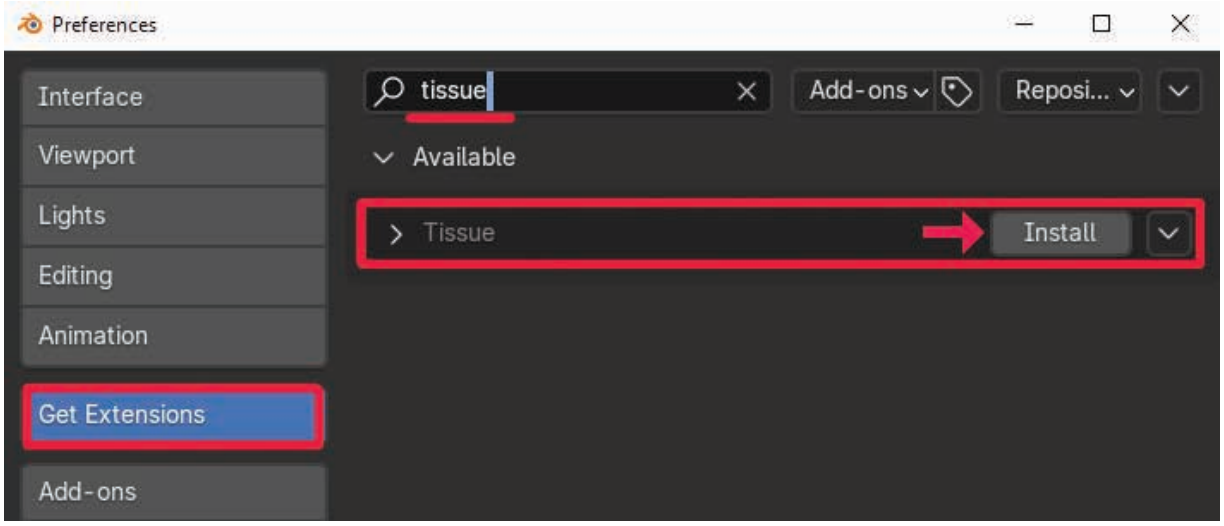


Right-click on the object and choose **Shade Smooth**.

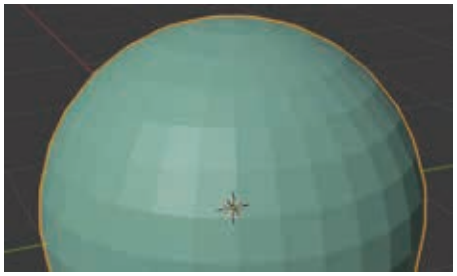
Press Shift+A and to Mesh > UV Sphere to add a sphere inside and scale it up. You'll need to right-click on that sphere and choose **Shade Smooth** as well.

Tissue Add-on: Experiment 1

The **Tissue addon** made by Alessandro Zomparelli gives us the ability to make very interesting mesh patterns. It comes with Blender by default by needs to be enabled in Preferences. Go to the Get Extensions tab and search for Tissue, then click on Install.

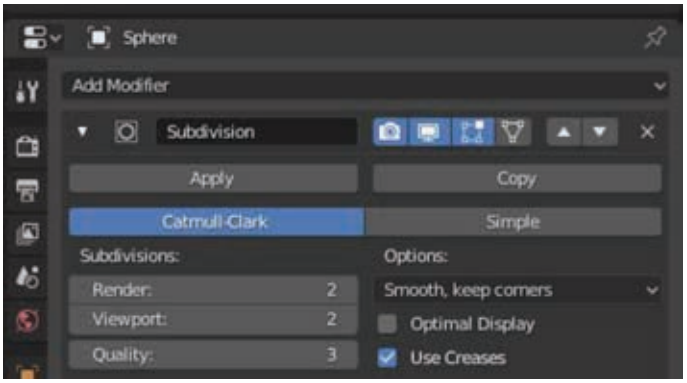


Press Shift+A, choose Mesh > **UV Sphere**.

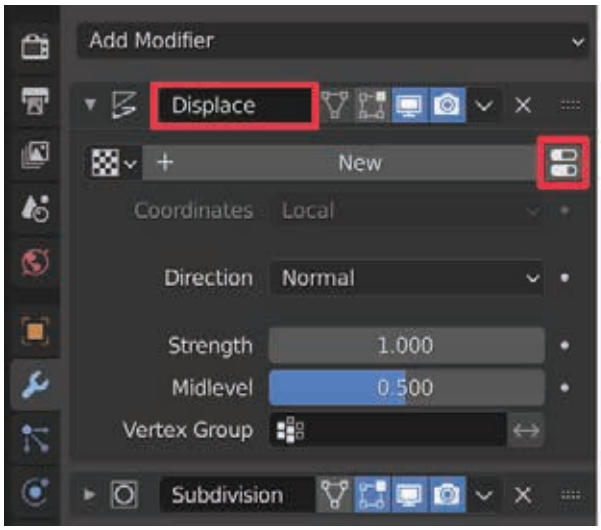


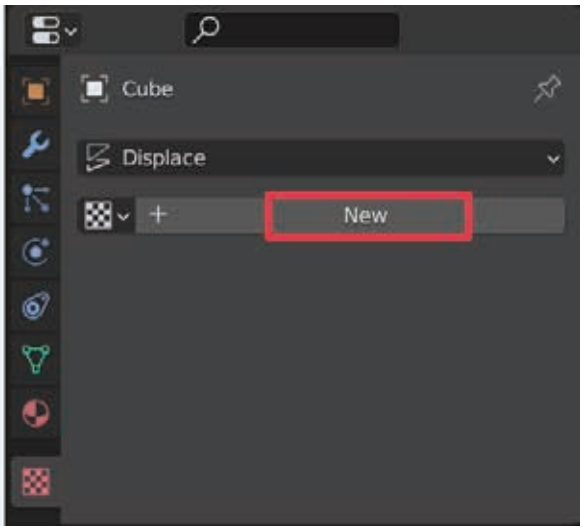
Add a **Displacement modifier** above the Subdivision modifier and click on the Texture icon.

Press Ctrl+2 to add a Subdivision modifier with 2 levels of subdivisions.

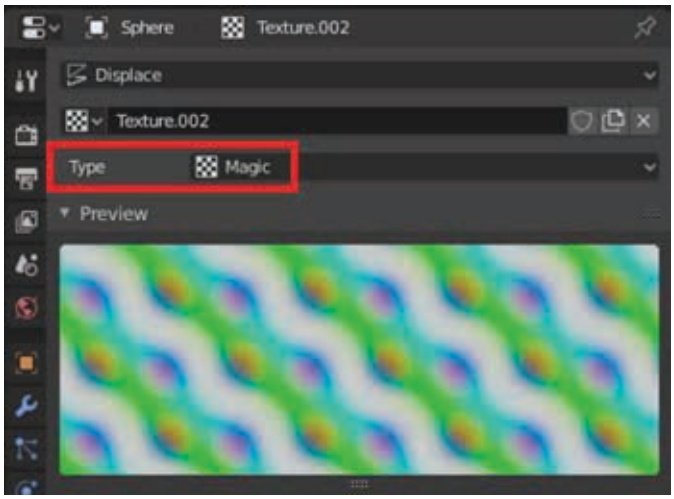


In the Texture tab, click on **New**.

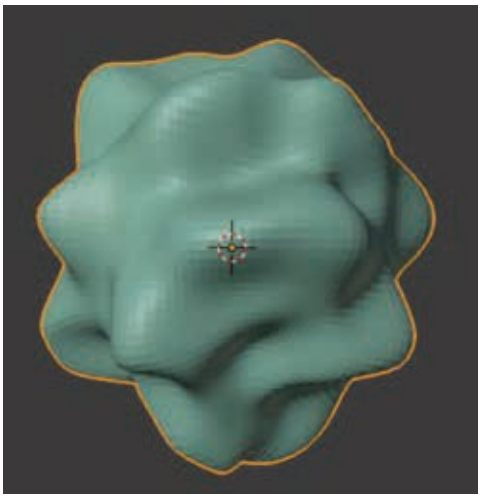




Choose an interesting procedural texture under **Type**, for example **Magic**.



You will see the resulting displacement in your object.



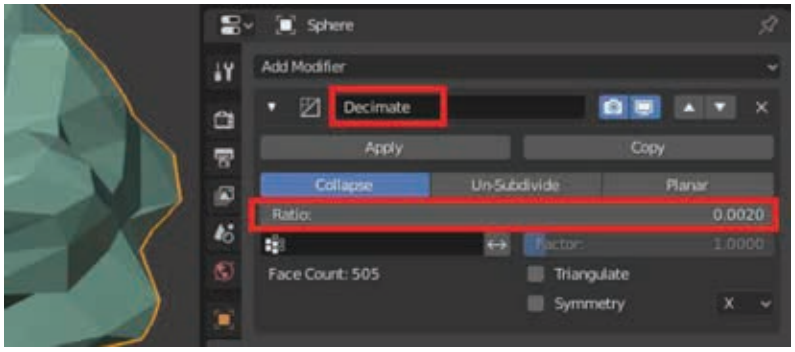
If you want you can experiment with the settings of the texture, in this case the **Magic texture** gives us control over Depth and Turbulence values.

When you are satisfied with the settings, subdivide the object some more and apply the modifiers.

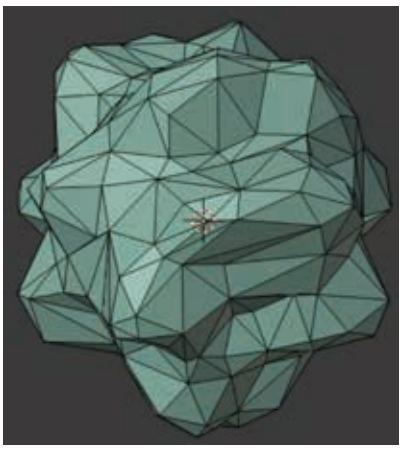
Alternatively, **press F3** and search for "Convert". Choose **Mesh from Curve**to apply all modifiers at once.

Now, add a **Decimate modifier**.

Setting the **Ratio** very low actually produces new, interesting geometry. Note that it's better to type a value than to use the slider, which can be very slow.

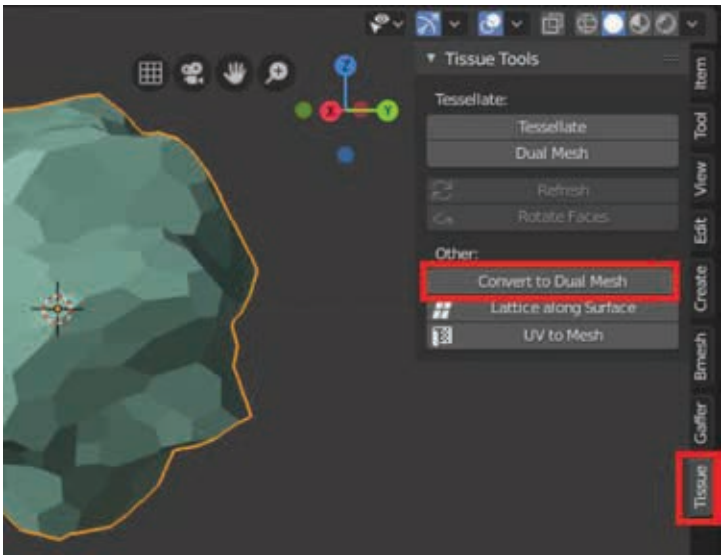


Apply the Decimate modifier.

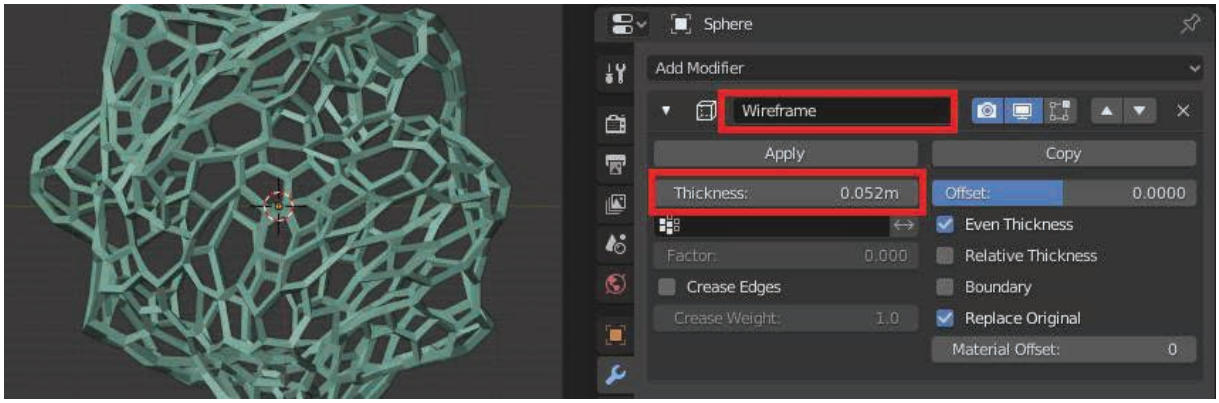


Now let's use the Tissue addon. Open the Option panel by **pressing N** and click on **Convert to Dual Mesh**.

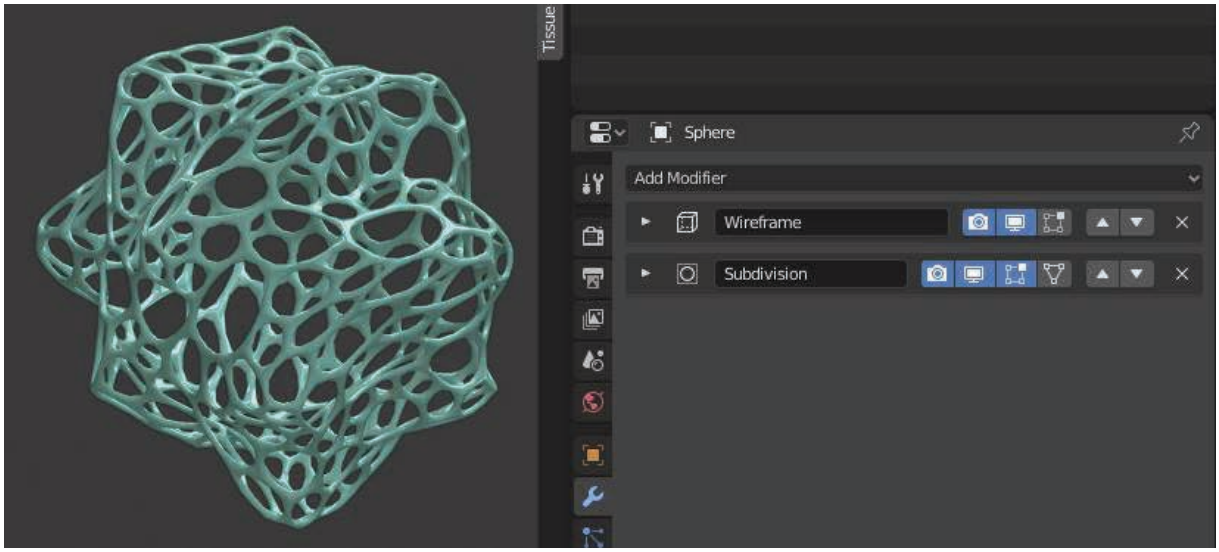
You see the result, in this case a kind of honeycomb structure. If you prefer to keep the original mesh and create a copy instead, click on **Dual Mesh**.



Add a **Wireframe modifier**and tweak the **Thickness** to your liking.

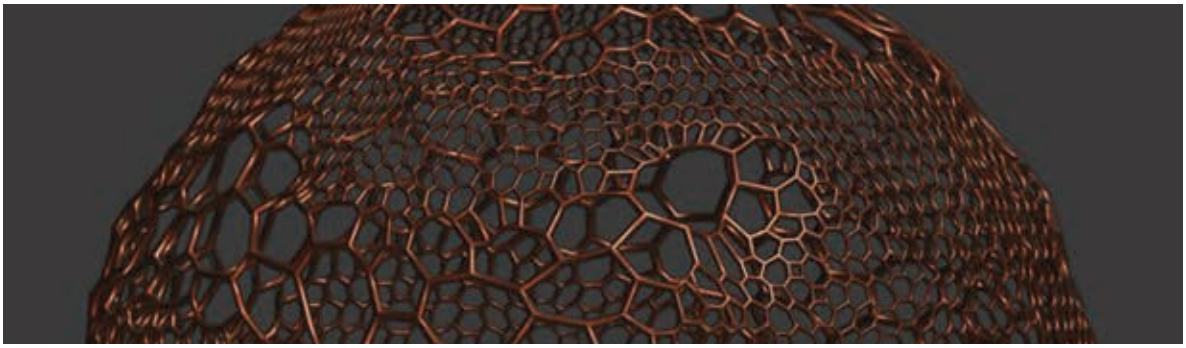


Add a **Subdivision modifier** below the wireframe modifier to make the result smooth. Note that I'm using a Matcap and Smooth Shading here to make it look even better.

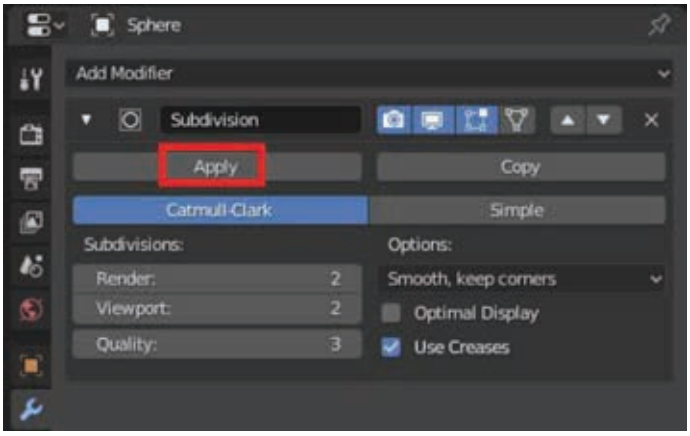


Tissue Add-on: Experiment 2

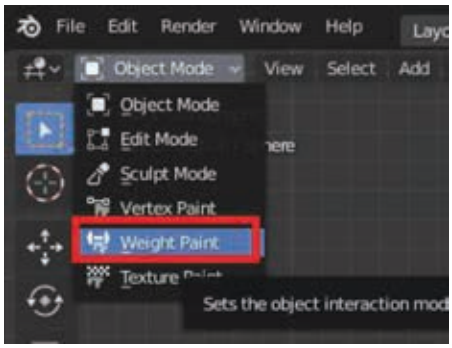
Certain modifiers can use a weight painted map to influence how they work. This way, interesting varied results can be achieved without doing any traditional modeling. This tutorial requires the included **Tissue add-on** (see previous topic).



Create a **UV sphere** (Shift+A > Mesh > UV Sphere) and **press Ctrl+2** to add a Subdiv modifier set to 2 levels of subdivision. **Apply** the modifier in the stack.

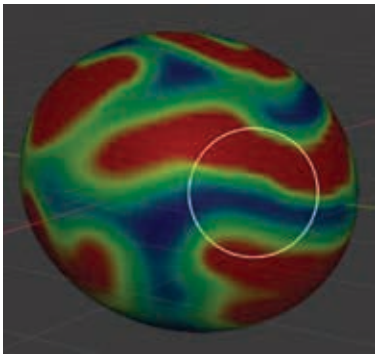


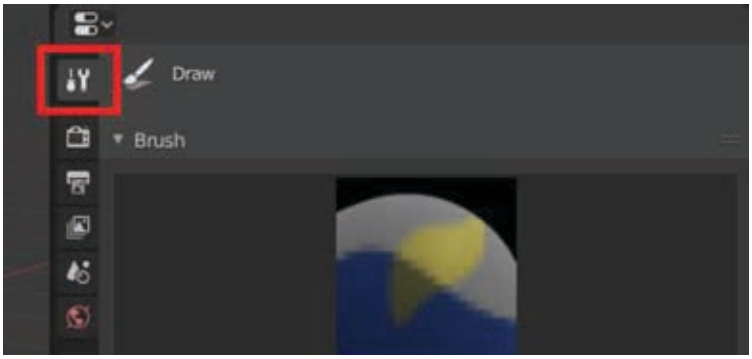
Go into Weight Paint mode.



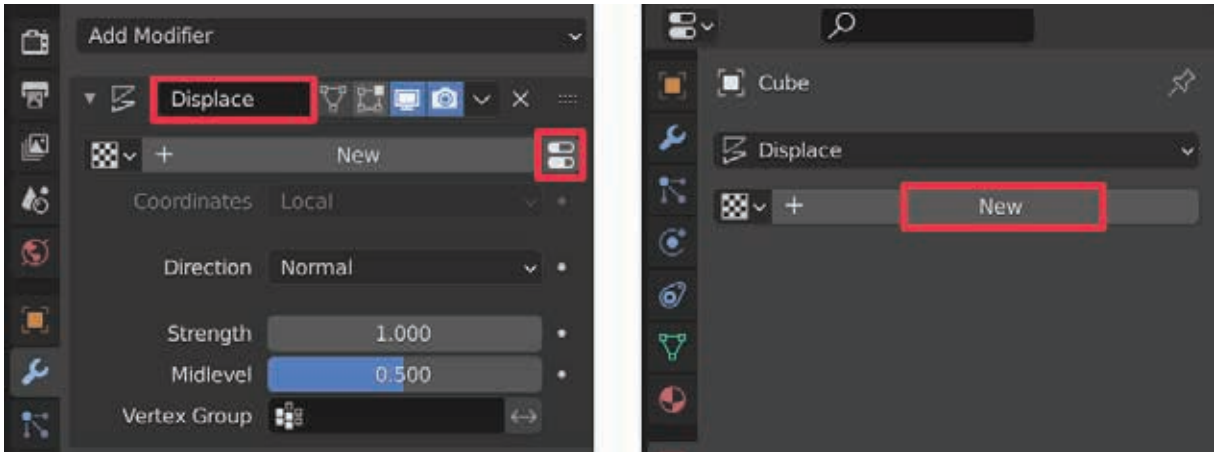
You can find the settings for weight painting in the **Active Tool settings tab**. For this example I will use the default settings.

Paint the weight map by dragging the mouse cursor over the object while holding the left mouse button. Repeatedly painting on one area increases the “weight”, shown by increasing the color towards red (blue being zero weight).

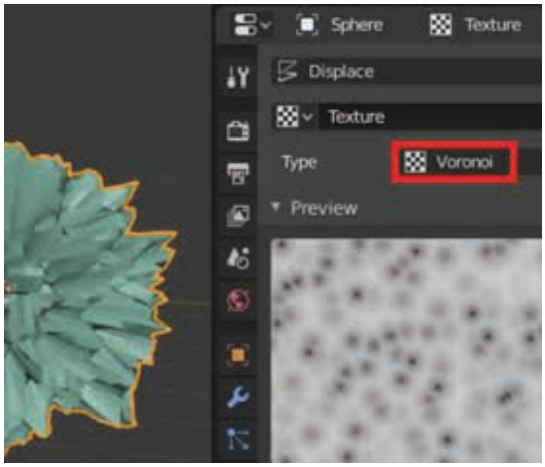




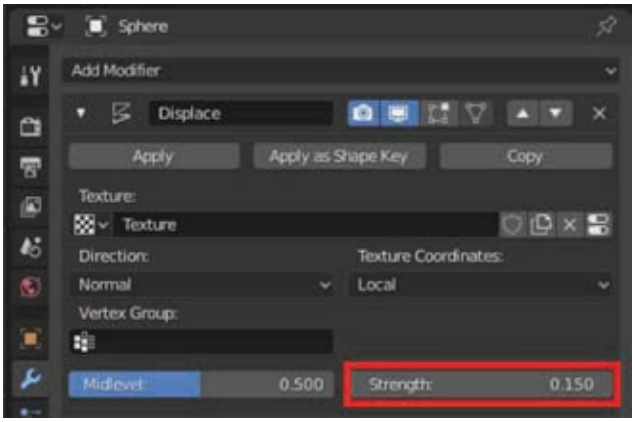
Add a **Displacement modifier** and click on the Texture icon. In the Texture tab, click on **New**.



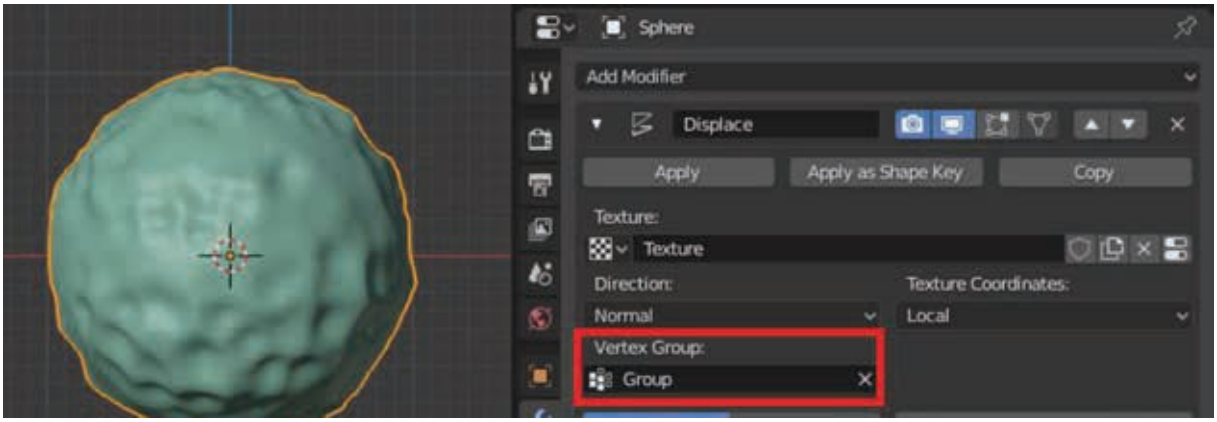
In this case I used a **Voronoi** type procedural texture.



As you can see the displacement is too strong. We'll need to lower it by decreasing the **Strength** value.

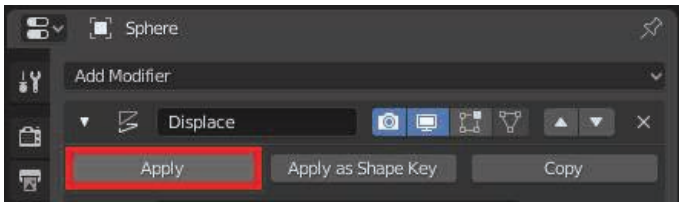


Use the **weight map** you painted earlier on the displacement modifier by selecting it in the Vertex Group option. (The weight map is called "Group" by default)

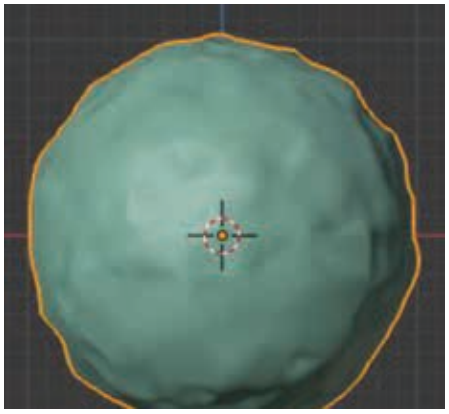


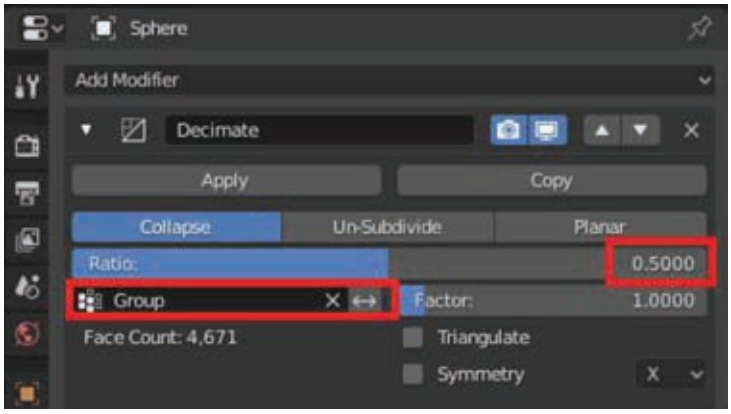
As you can see, the weight map influences where the modifier influences the model and where it doesn't.

Apply the displacement modifier and **Apply** it.



Add a decimation modifier. Set the decimate **Ratio** to about half, use the weight paint group again.

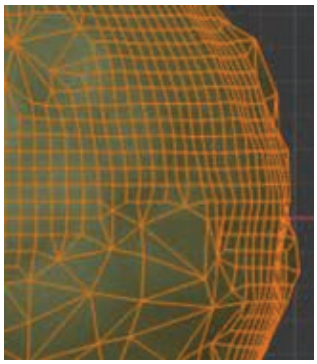


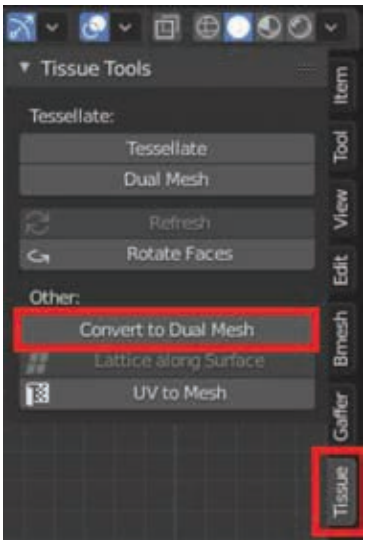


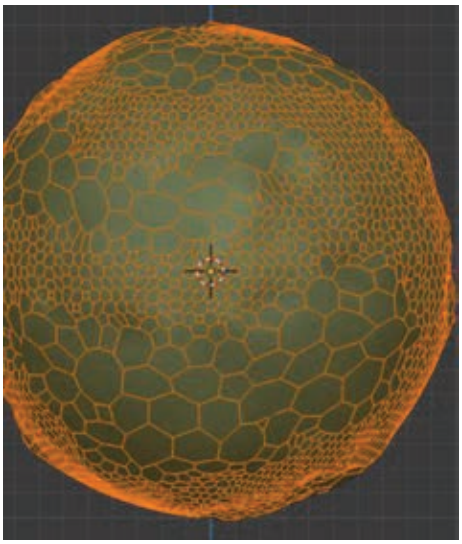
Apply the Decimation modifier.

In Edit Mode you can see that using the weight paint again influenced where the modifier did its thing. In this case, some areas are more decimated than others.

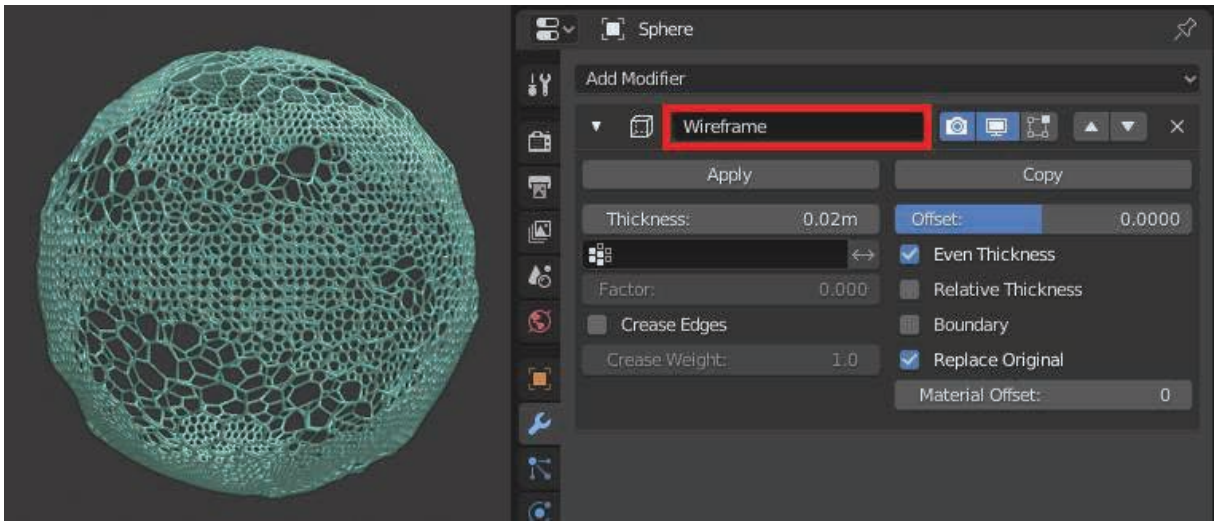
Click on **Convert to Dual Mesh** in the Tissue panel.



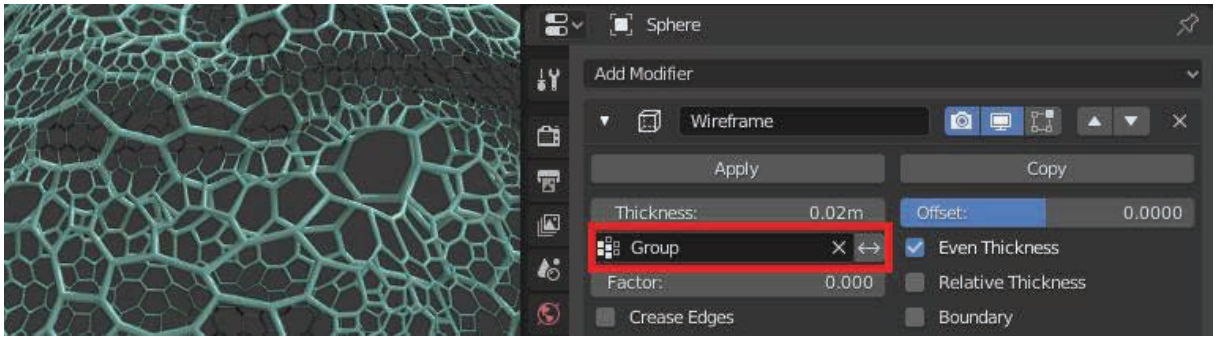




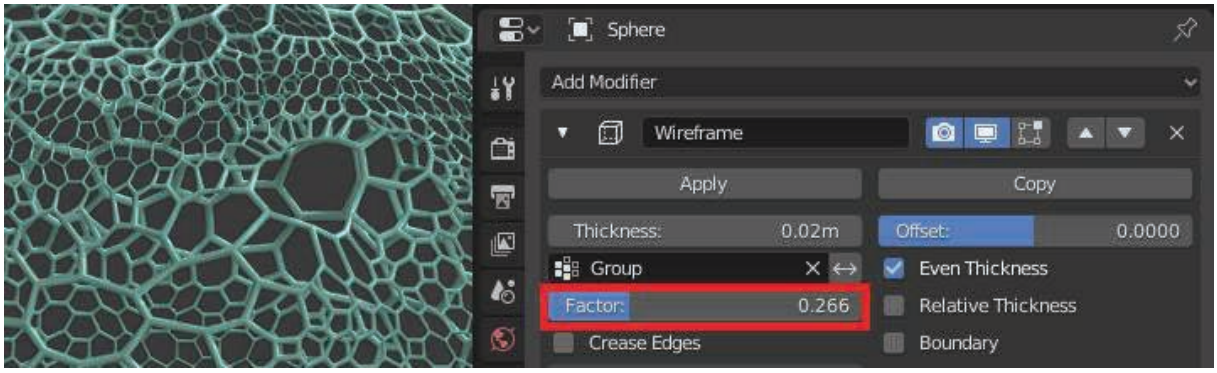
Let's add a **Wireframe modifier**like we did in the previous tip.



The weight paint map can still be of use for a third time. Add it as the **Thickness** factor in the modifier. The result is that some of the wires are thicker than others.



Maybe the thin wires are a bit too thin. We can change how much the modifier needs to listen to the weight map by changing the **Factor value**.

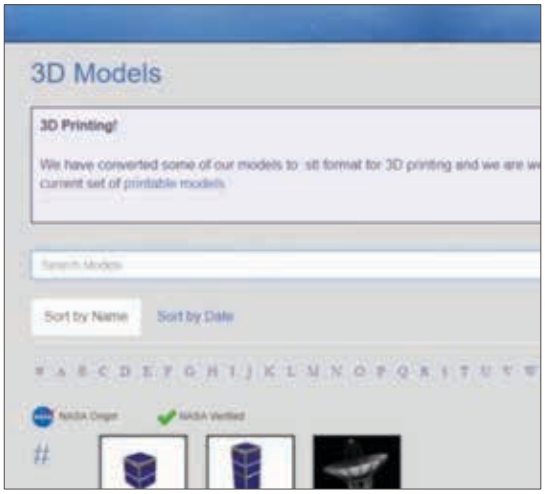


For bonus points, you can animate some of these settings (like the Thickness) to make some weird animation.

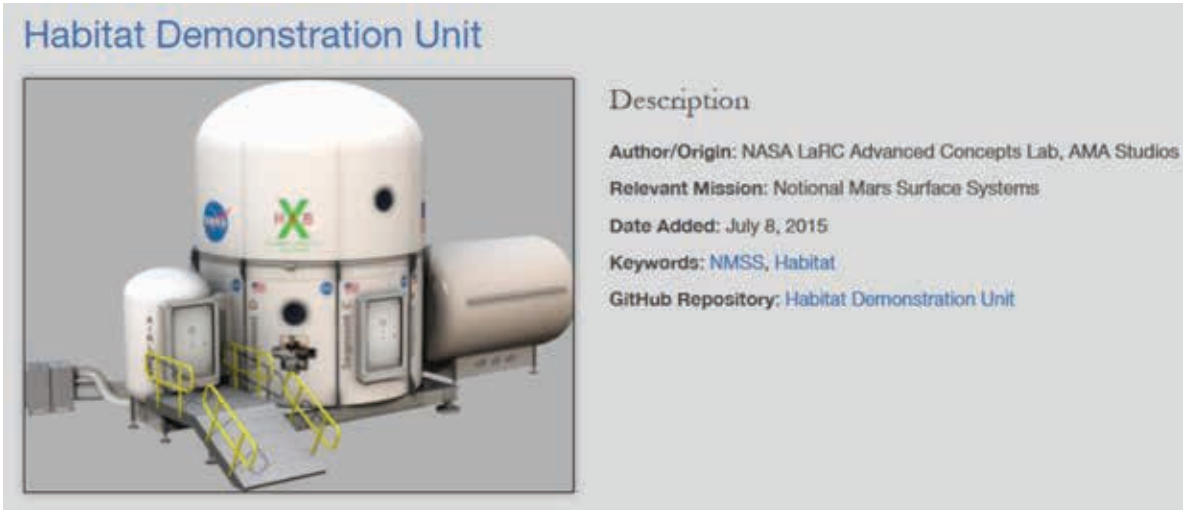
How to use Nasa's free models and textures

Nasa has a nice library of 3D models as well as textures for planets and moons.

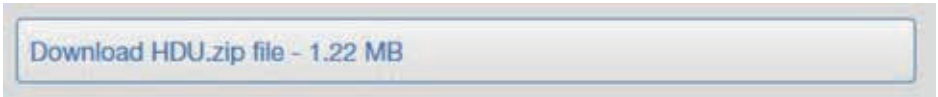
They can be found in this online archive: https://nasa3d.arc.nasa.gov/



This looks like a cool model. Imagine living inside this for a while...

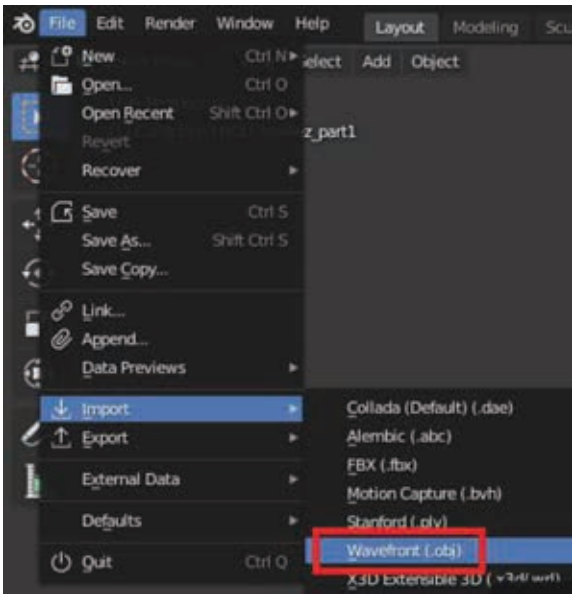


Click on the Download button below the description to download the model.



After downloading the file, unzip it and you'll find an OBJ file as well as some texture.

Click here to return to the index.



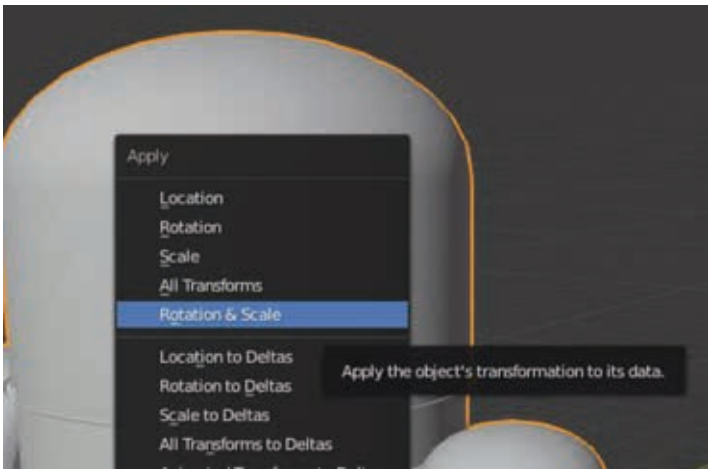
Import the **.OBJ** file that you downloaded. File > Import > OBJ.

In the case of this habitat, there are two OBJ files, so we'll import one after another. The object is probably too big soyou may not evenbeabletoseeit. Select the object in the **Outliner**.

Press S+[0.01] to scale the object to 1% of its original size.

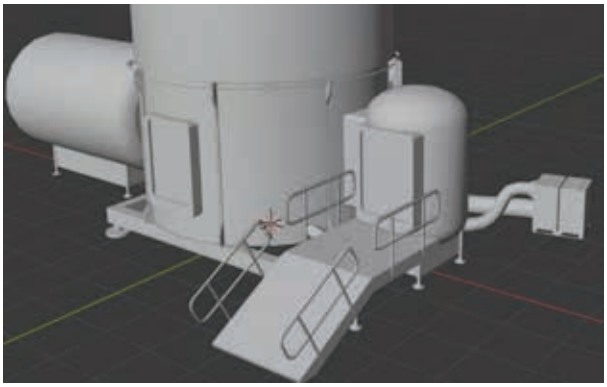
Also, rotate it 180° on the X-axis by **pressing R+X+180**.

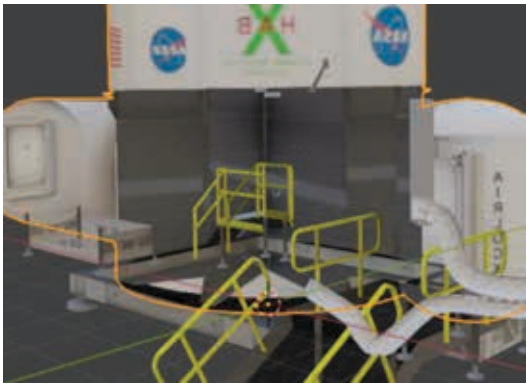
Press Ctrl+A and choose Rotation & Scale to apply the rotation and scale.



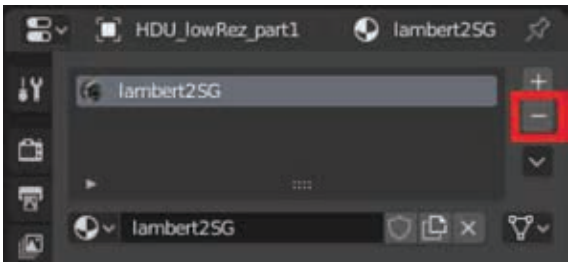
Import the second OBJ and repeat the step of scaling it down to 1% and rotating it 180° on the X-axis. Apply the rotation and scale. Now we should check the texture. Go to **Material view** or Rendered view.

Well, that looks pretty weird. It seems the normals are flipped, but flipping them back (by selecting everything in Edit Mode and **pressing Shift+N**) doesn't always help.

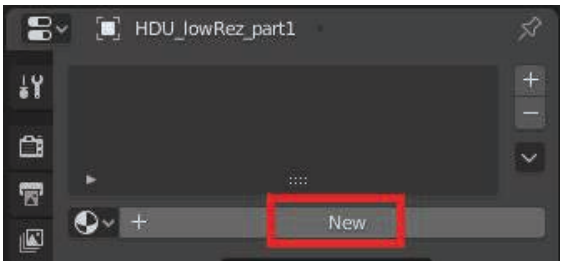




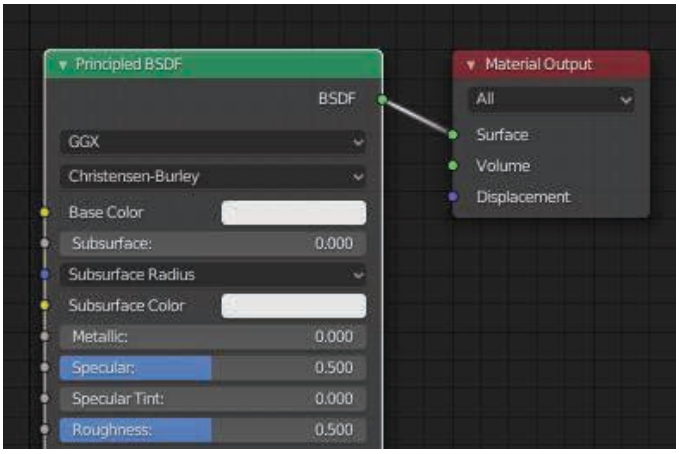
Press the [-] icon to delete the material.



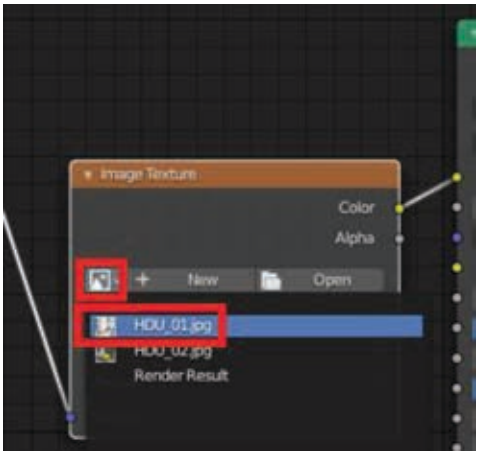
Create a new material.



Select the Principled BDSF node in the Shader Editor and **press Ctrl+T** to add the nodes for the image texture. (This assumes you have the **Node Wrangler add-on** enabled in Preferences).



In the Image texture node, choose the texture which is still loaded in memory (in this case it's HDU_01. jpg).

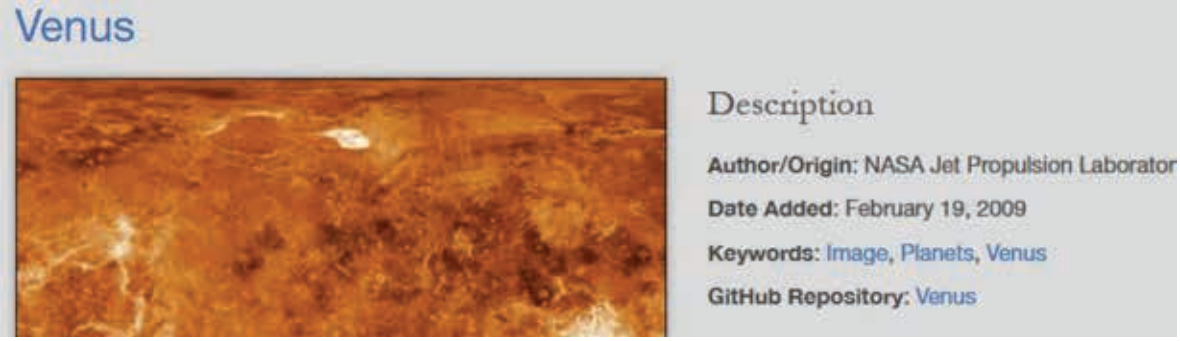


Now it's starting to look better!

Do the same for the other part of the model.

Besides models, you can find many images and textures here: https://nasa3d.arc.nasa.gov/images

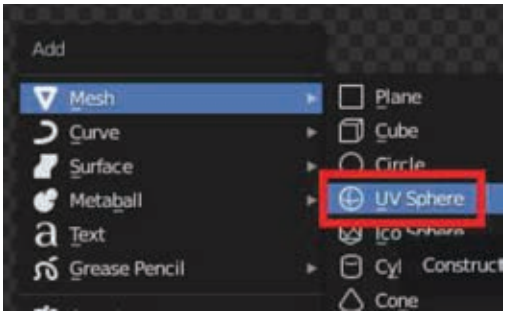
Let's download this texture of Venus. Click on the download button below the description on the website. Unzip the archive once it's finished downloading.



It's in a format that is ready to be wrapped onto a UV sphere.

Create a UV sphere by Pressing Shift+A and choosing Mesh > **UV Sphere**.

Press Ctrl+3 to give it 3 levels of subdivision.



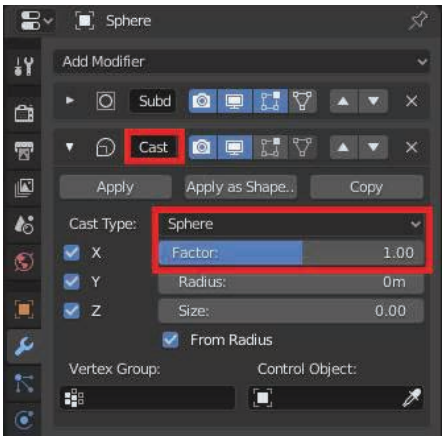
Due to the mesh topology of the UV sphere, you may see this kind of pinching effect at the poles.

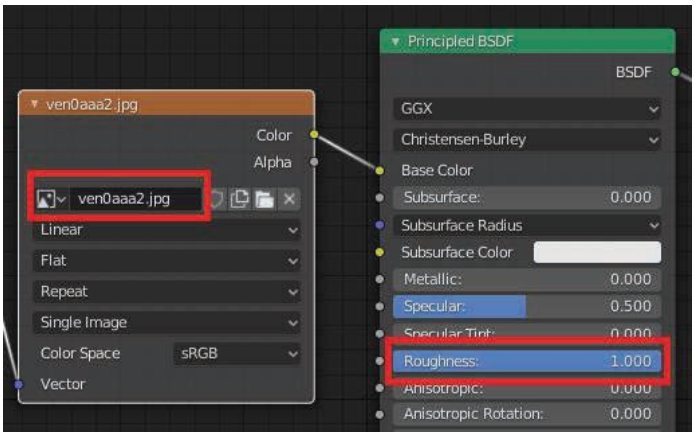
This can be solved by adding a **Cast modifier**.

Set it to Cast Type: **Sphere**. **Factor: 1**. This should make the sphere perfectly round, with smooth poles. Make sure it's underneath the Subdiv modifier.

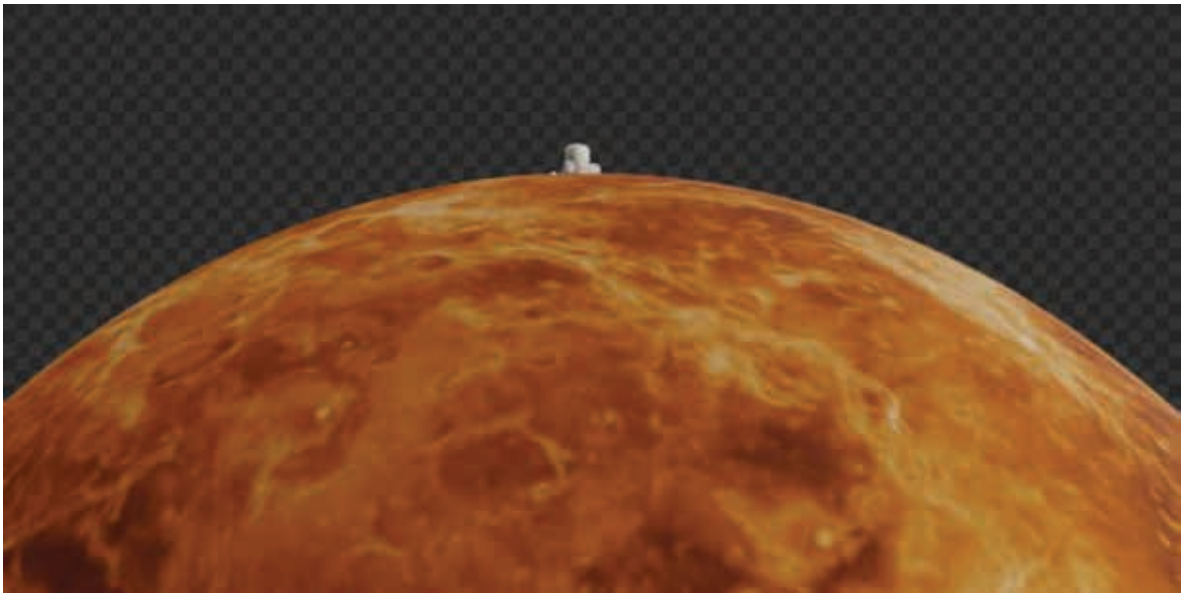
Just like with the habitat model, create a material and give it an image texture.

Select the Venus image you downloaded. Now, turn the **Roughness** value up so the planet is not shiny.





Add the little habitat on top and we're all set! (Habitat not to scale)

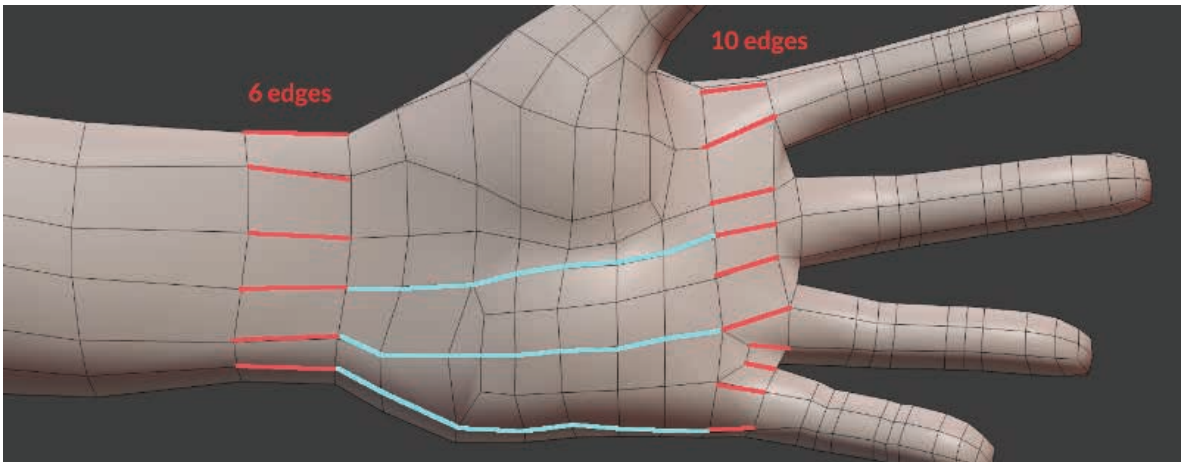


Click here to return to the index.

Quad Topology Tips

Quads are Faces that have four edges. A 3D model that is required to deform, such as a character, and that uses a Subdivision modifier, looks best when made entirely of quads.

However this can lead to some modeling challenges. When modeling a hand for example, you inevitably need more geometry for the fingers than for the arm. You therefore need to know how to reduce the amount of edges flowing from the fingers to the arm through the hand, while avoiding using triangles (faces with less than 4 edges) or N-gons (faces with more than 4 edges). Below you see an example of how a model goes from 10 edges where the fingers meet the hand to only 6 for the arm.

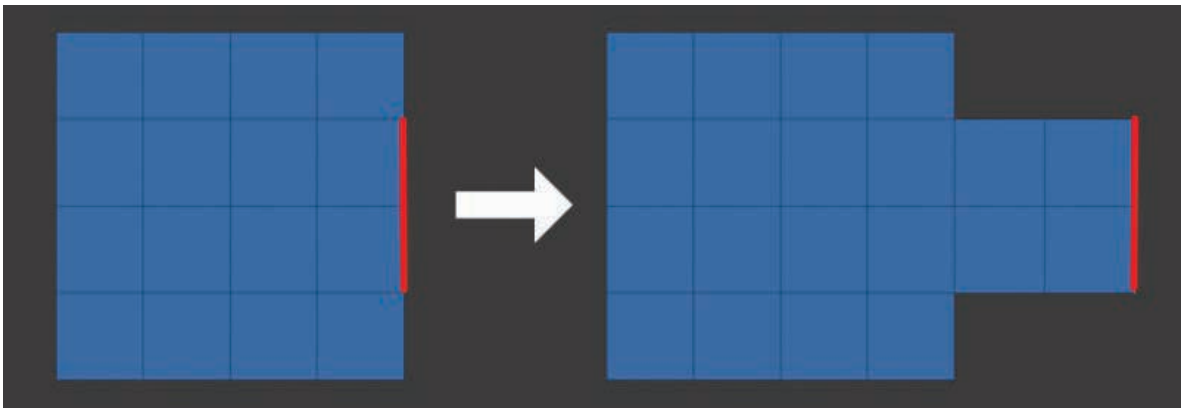


Now that we understand what quads are for and when to use them, let's look at some situations that you might run into, and how to solve them.

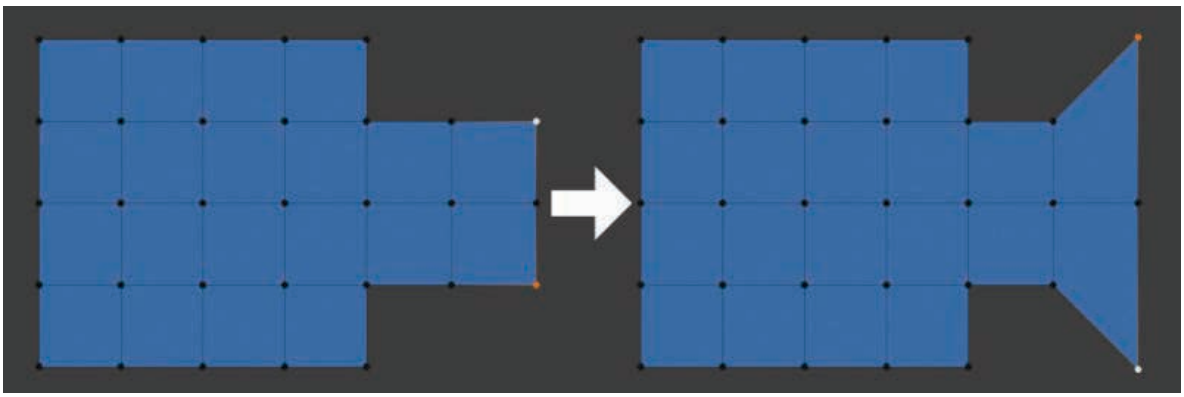
From 4 to 2 quads

In Edit Mode, go to Edge Select Mode by **pressing 2**. First, select two edges. Then **press E** and extrude them to the side.

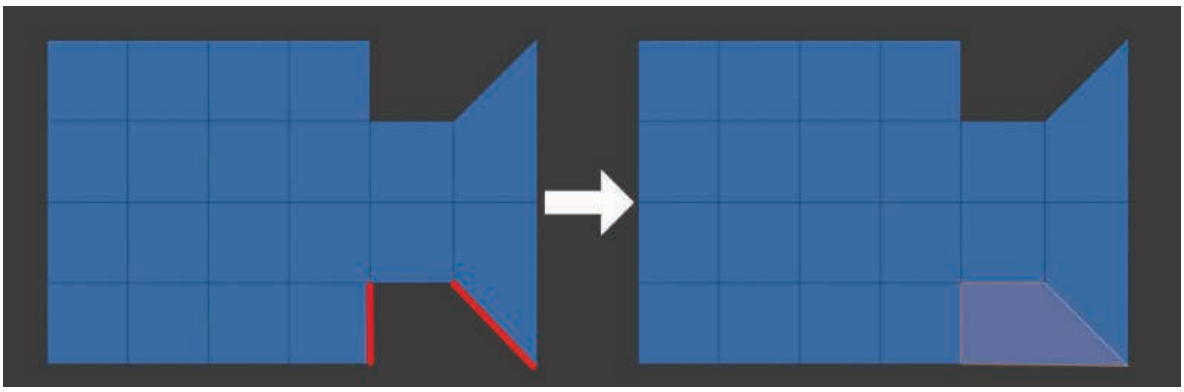
In my case I **pressed E+X** to extrude them along the X-axis.



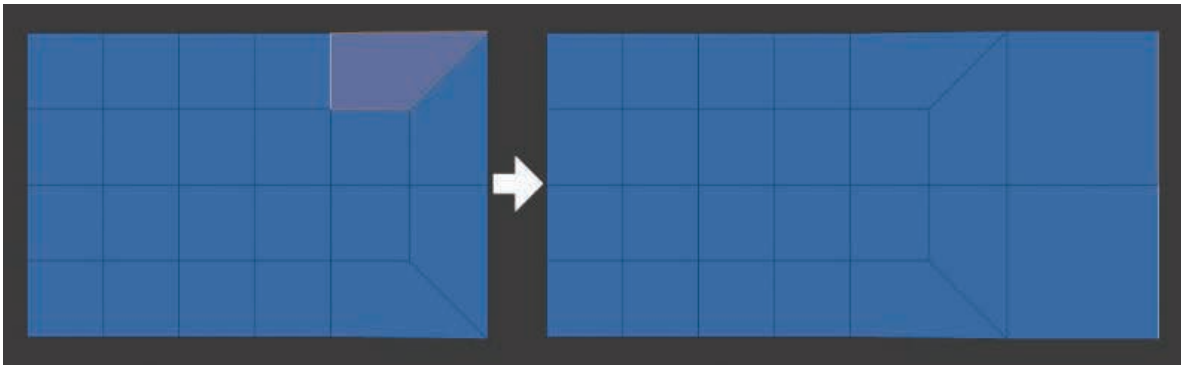
Next, go to Vertex Select Mode by **pressing 1** and select the vertices at the corners of the extruded faces, then scale them away from each other by **pressing S**. This way you can move them both at once.



Now **press 2** to go back to Edge Select mode and select two opposing edges and **press F** to fill the hole with a Face. In case you happen to have the **F2 add-on** (found in Extensions) enabled, you only have to select one edge (the one between the two other edges) and **press F** to fill the hole.

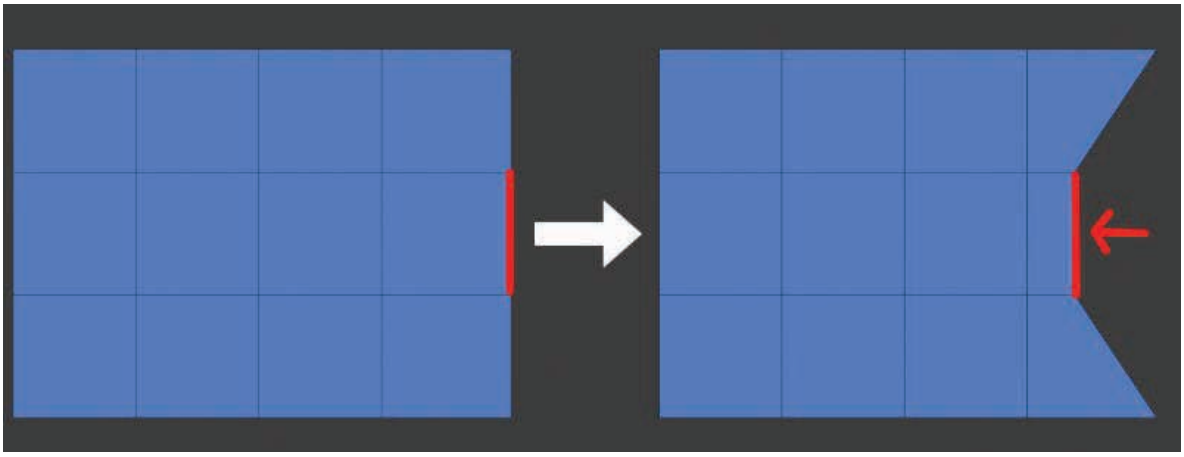


Fill the missing Face at the top in the same way as before. Then select the two edges at the right and extrude them by **pressing E** (or **E+X** to extrude perfectly along the X-axis). Now you've effectively gone from 4 quads to 2 quads, without losing the "all quad" topology.

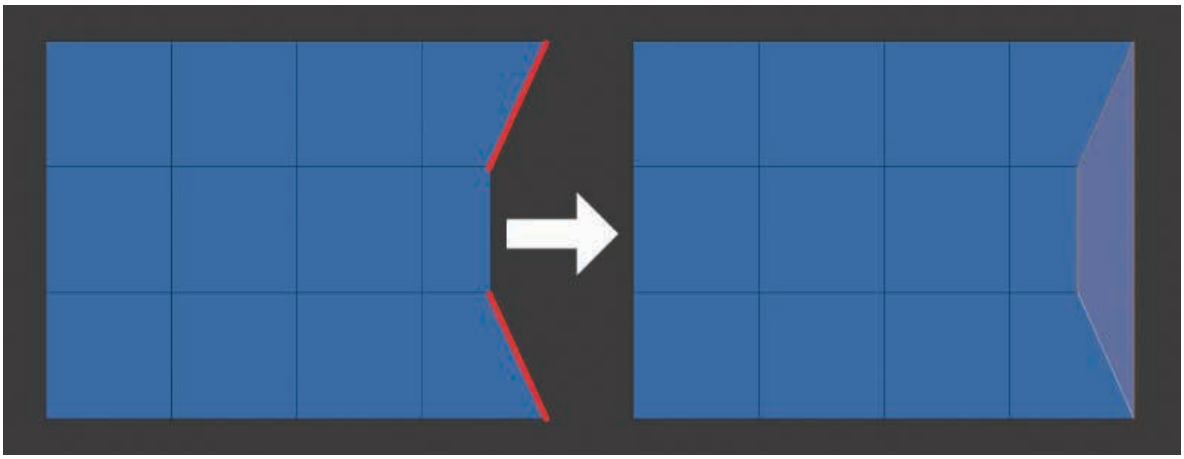


From 3 quads to 1 quad

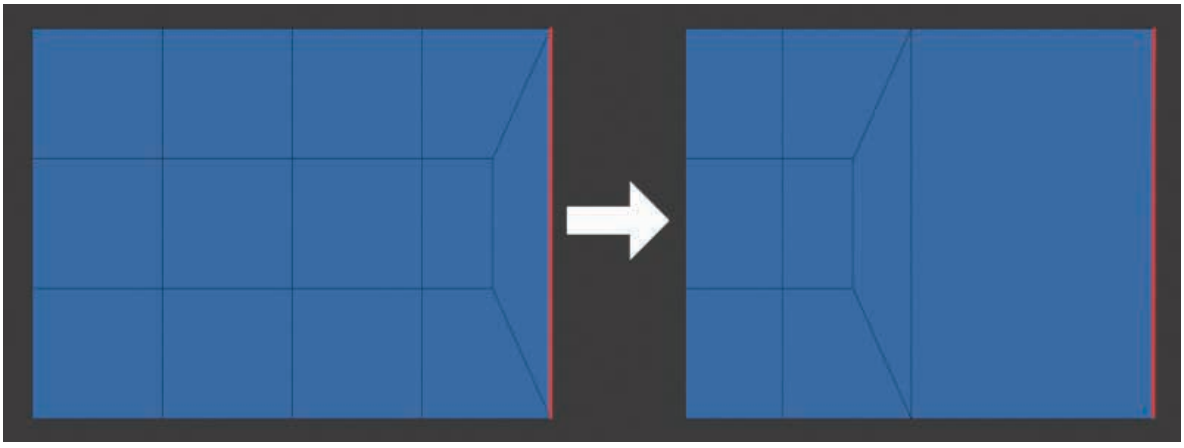
In a different situation, you might want to go from 3 quads to just one. First, select the middle edge on the right. Then, **press G+X** and move it to the left along the X-axis.



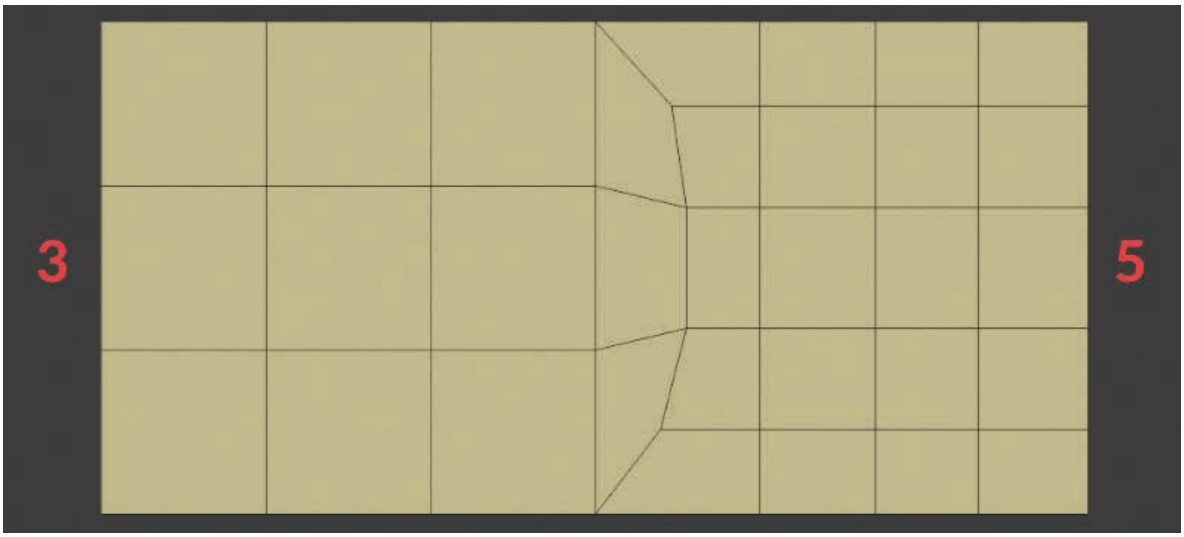
Now select the top and bottom edge on the right, then **press F** to fill with a Face.



Finally you can extrude the edge on the right by **pressing E** and you'll have your topology going from 3 faces to just one face, while still having all quads.

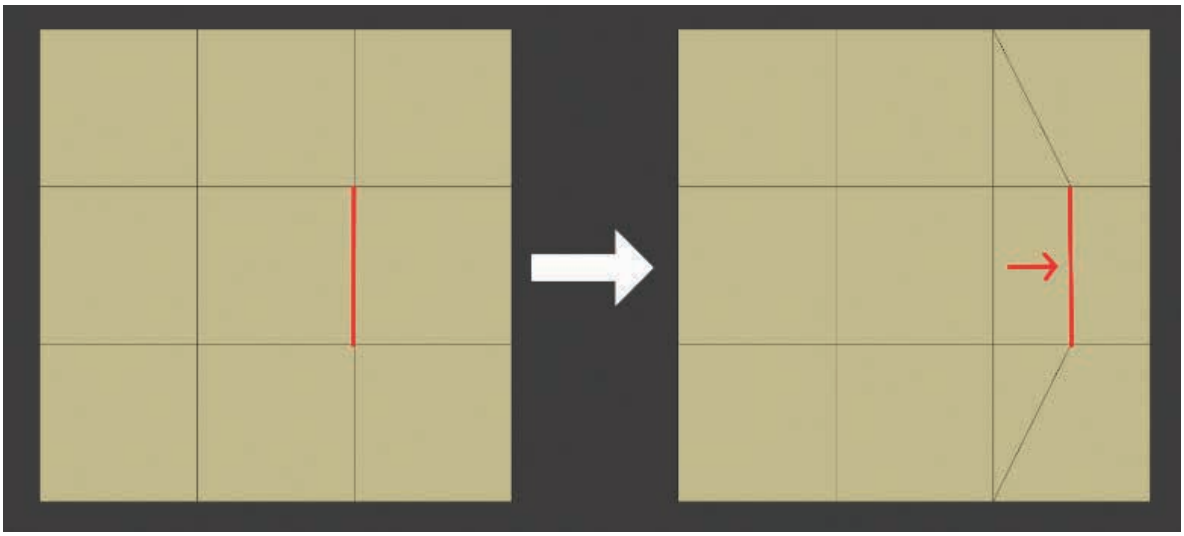


These two methods can help you go from more to less quads. But what about going from less to more? Here's a trick that you can use, for example to go from 3 to 5 quads.



From 3 to 5 quads

In Edit Mode, **press 2** for Edge Select Mode. First, select one edge, as shown. Next, **press Alt+V+X** to "**Rip**" that edge on the X-axis without leaving a hole (just pressing V for Rip without pressing Alt would have left a hole).

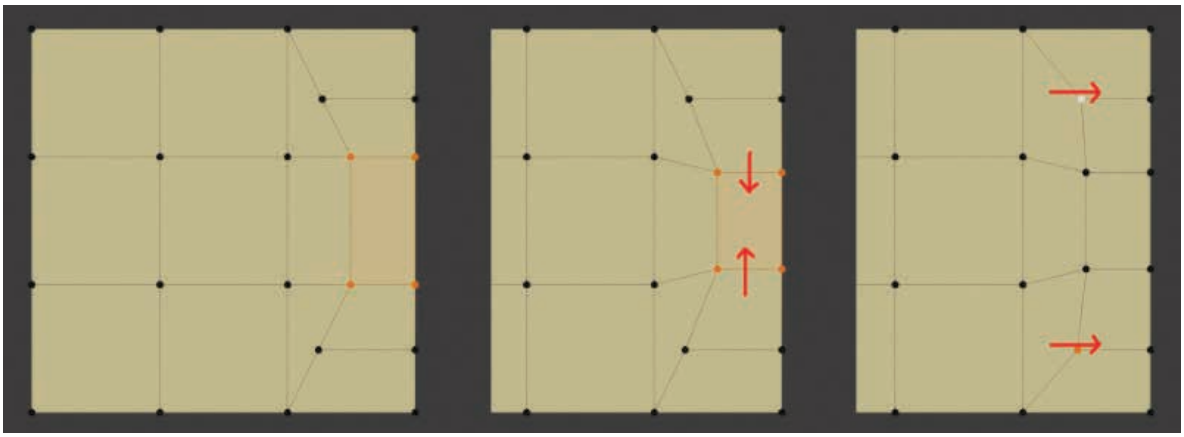


As you can see, this creates two diagonal edges, however these also introduce triangles, which we don't want in our "all quads" topology. This is fixed by using the **Knife Tool**. To activate it, **press K** and click in the middle of a diagonal edge, then **press A** to "**constrain**" the angle of the cut and click on the edge to the right.

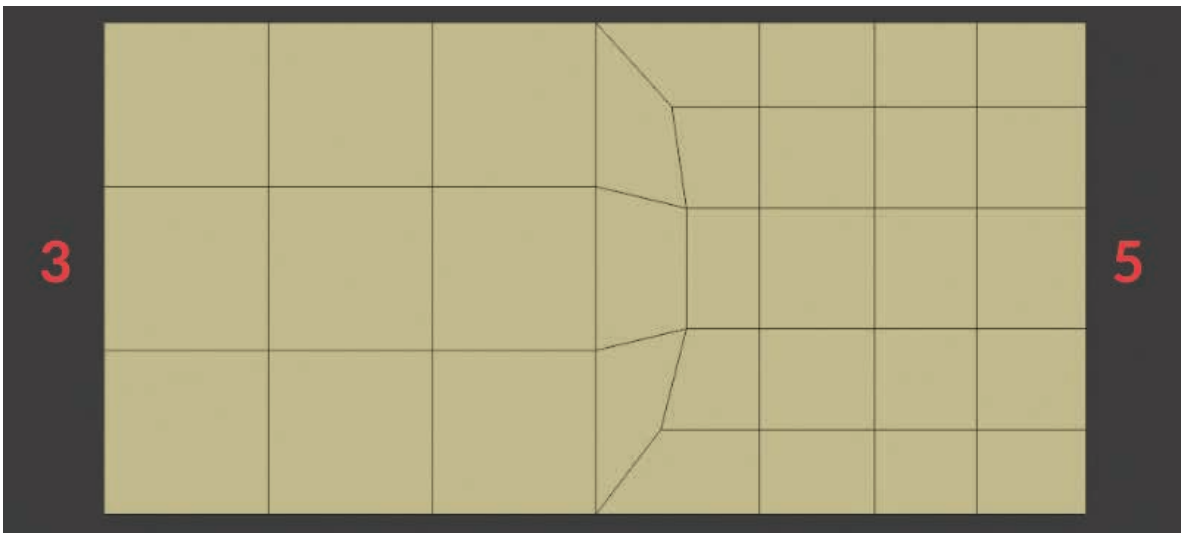
Then **press Enter** to confirm.

Do the same for the diagonal edge at the lower row of faces. Now you can select the vertices of the middle Face on the right and scale them towards each other on the Z-axis by **pressing S+Z**, this creates a more even topology where that Face has the same size as the other faces around it.

You can also slide some vertices by **pressing G+X** to move them on the X-axis to the right, so those Faces look less like triangles and more like quads (which they are).



Now you've got the mesh flowing from 3 to 5 faces without destroying the "all quads" topology.



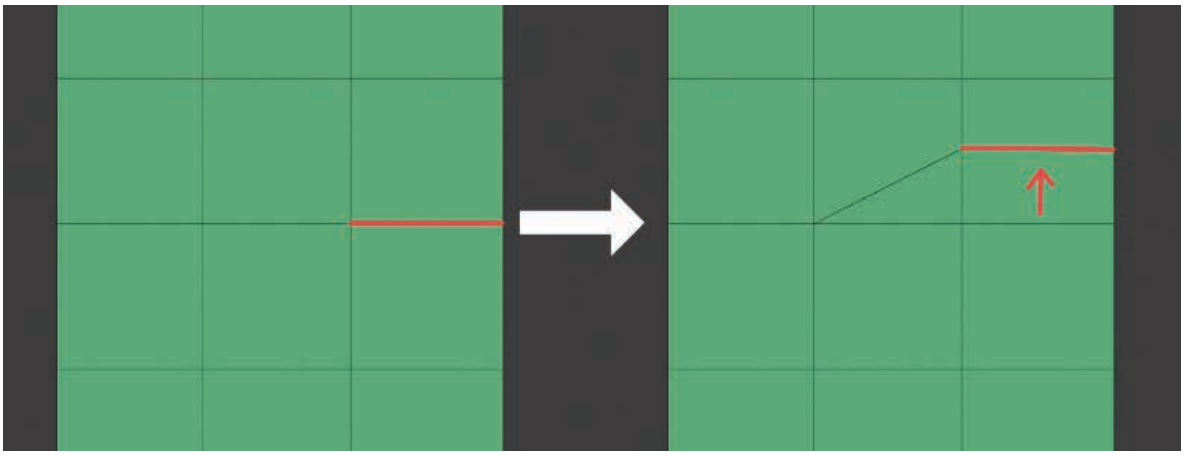
Tip: a surprisingly good resource for topology examples is Pinterest.com, if you search for something like "3D character topology" you can find a lot of good examples. Topologyguides.com is another good resource.

Similarly, you can use this "ripping" trick to go from 2 to 4 quads. It's always good to know a trick for even numbers, as well as uneven numbers.

From 4 to 6 quads

In Edit Mode **press 2** to activate Edge Select Mode, then select the middle edge on the right.

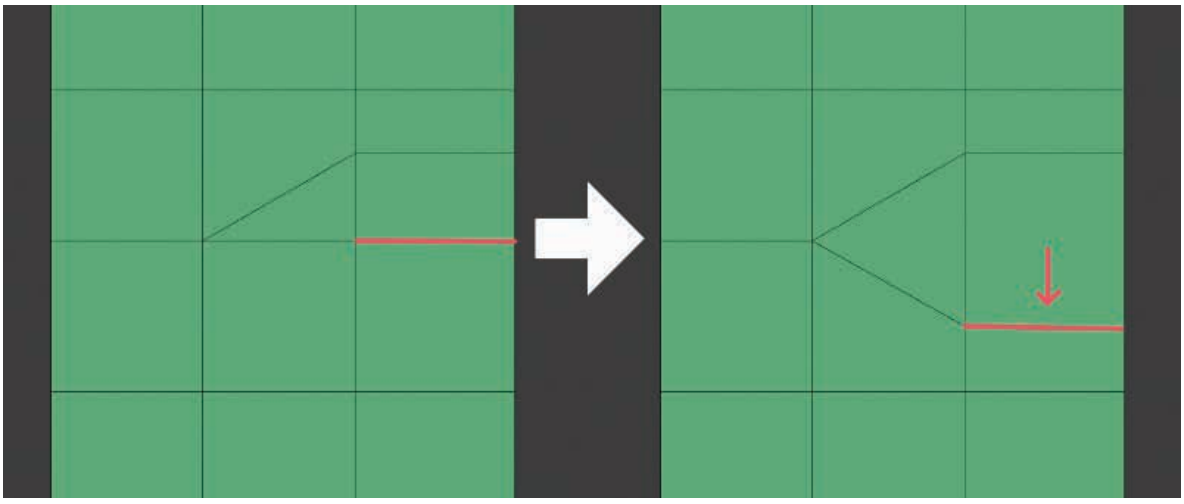
Press Alt+V to rip and move that edge, then move it up or down by **pressing Z.**



***Note**: Before constraining the movement to the Z-axis by pressing Z, it can be a good idea to freely move the edge around to see whether you should move it up or down.*

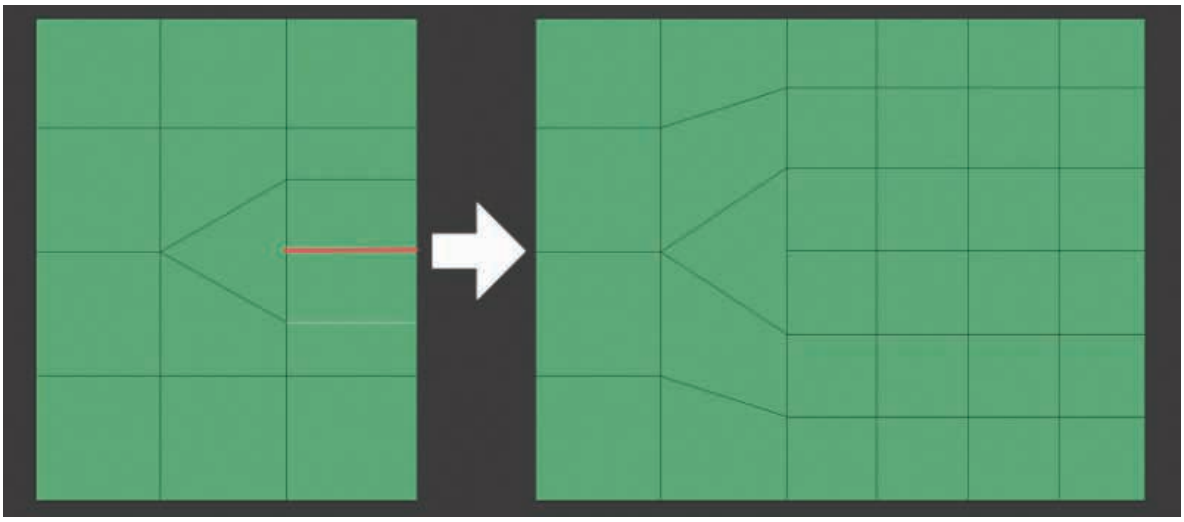
Blender somewhat randomly decides which direction your ripped edge should go in, and if you go in the wrong direction it actually overlaps with other geometry, which is not good.

Now that you've ripped that edge, select the edge in the middle of the right column of edges and move it up or down (depending on where you moved the ripped edge) by **pressing G**.



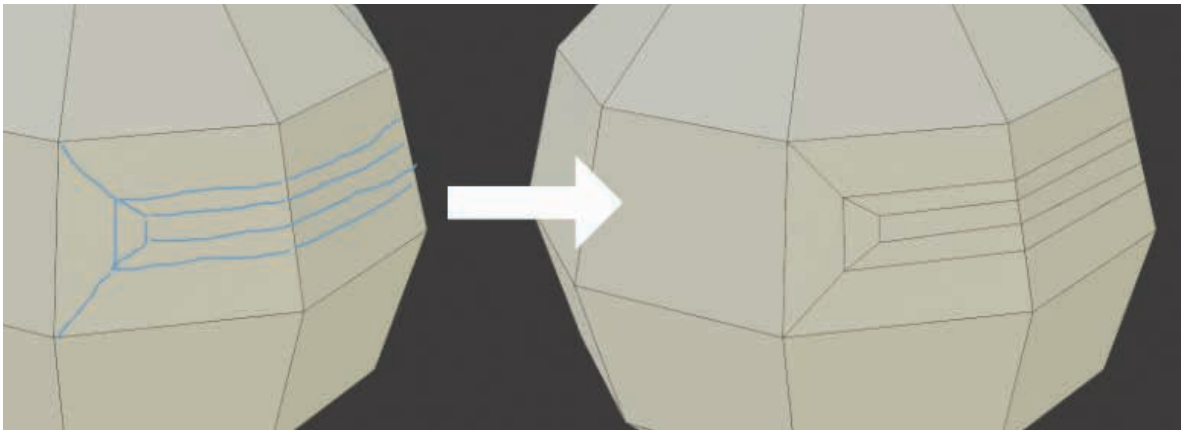
Now, you can cut a new edge in the middle by **pressing K** for the **KnifeTool** and constraining it to a straight line by **pressing A**. Confirm the knife cut by **pressing Enter**.

After that, you can rearrange the vertices of the 6 quad Faces on the right so they are more equal of size, and then extrude the edges to the right. Now you have topology going from 4 to 6 faces.



As long as you know these methods to go from more to less quads, both from even and uneven numbers, you can tackle many retopology situations.

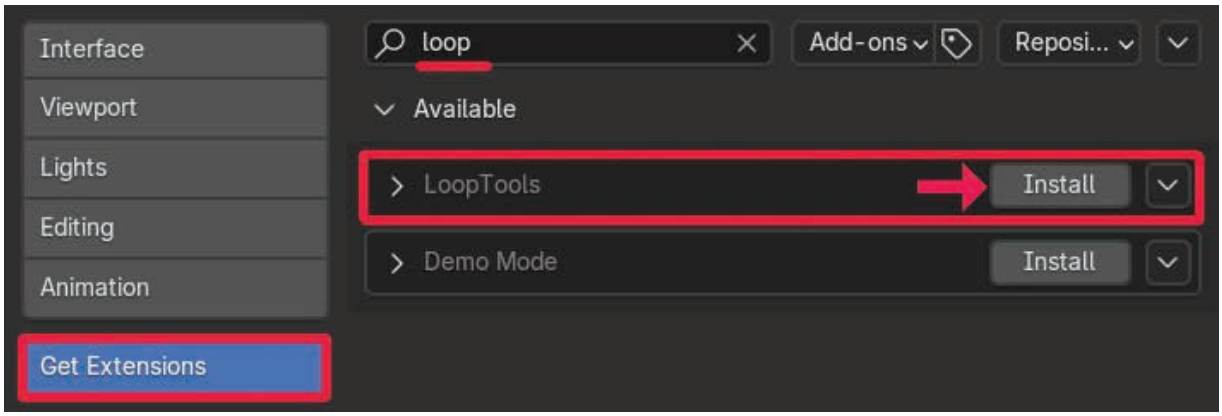
It can be useful to draw the new edges with the **Annotate Tool** first, so you don't have to do it all in your imagination. The Annotate Tool is activated by pressing and **holding D** while drawing with the **Left Mouse Button**. To erase it, **hold D** and use the **Right Mouse Button**.



An example of drawing the retopology with the Annotate tool before modeling.

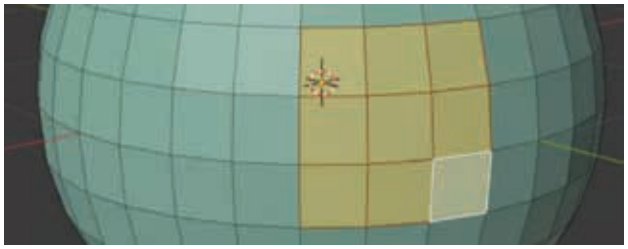
Circular hole in a sphere

This tutorial uses the **LoopTools add-on** which comes with Blender by default. Make sure it's enabled in Preferences.

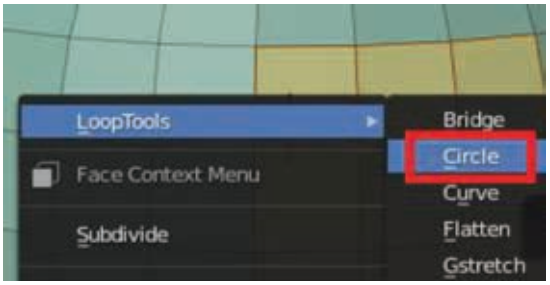


For this example I'll use a UV Sphere. Create it by **pressing Shift+A** and choosing Mesh > UV Sphere. It has 32 segments and 16 rings.

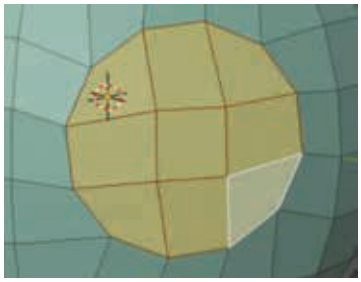
Press TAB to go into Edit Mode and **press 3** for Face selection mode. Select 9 faces forming a square.



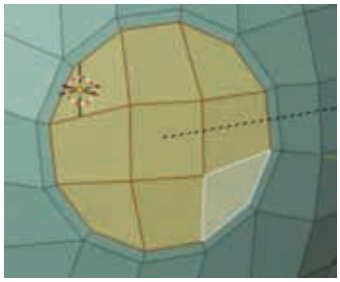
Right-click and from the LoopTools menu choose **Circle**.



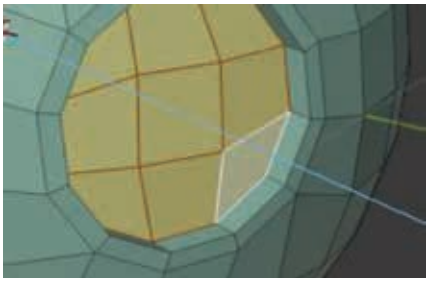
The faces selection will have become circular.



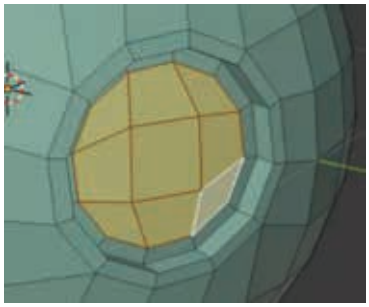
Press i to inset the selection.



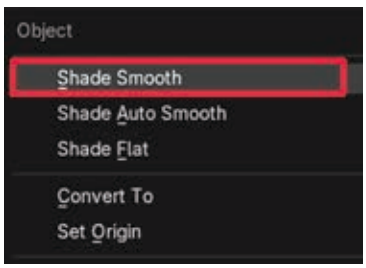
Then **press e** to extrude and move the extrusion inwards.



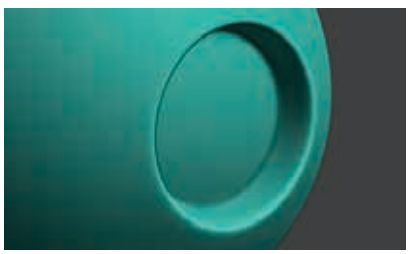
Press i to inset again.



In Object Mode, right-click and choose **Shade Smooth**.



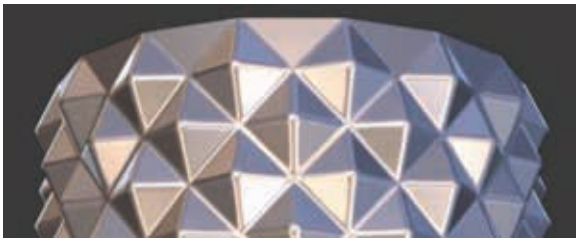
Add some subdivisions by **pressing Ctrl+3**. This adds a Subdiv modifier at level 3.



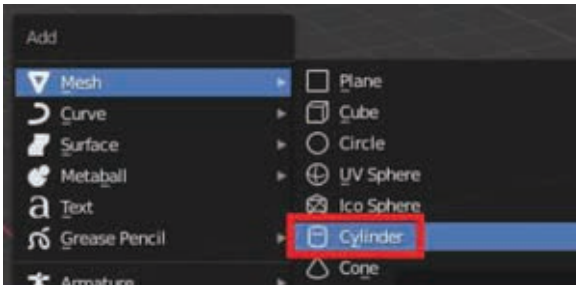
Click here to return to the index.

Unsubdivide and poke faces

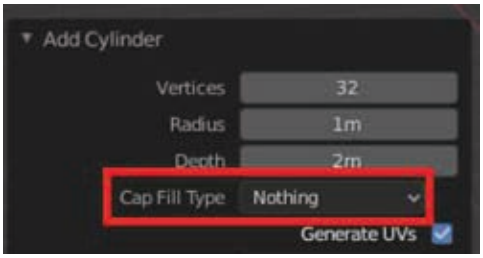
This is a good technique for making some kind of mechanical handle, like on a surgical instrument.



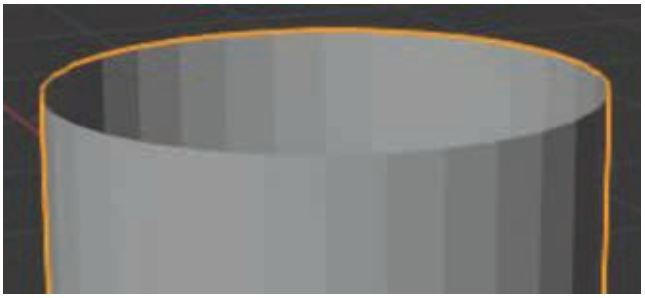
Create a cylinder by pressing **Shift+A** and choosing Mesh > **Cylinder**.



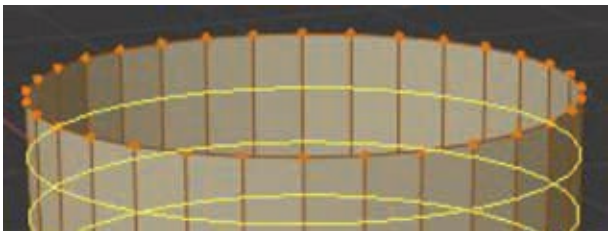
In the Cylinder options, choose **Nothing** as Cap Fill Type.



The cylinder will normally look like this.



Press TAB to go into Edit Mode and add some more edge loops by **pressing Ctrl+R** and scrolling the mouse wheel up.

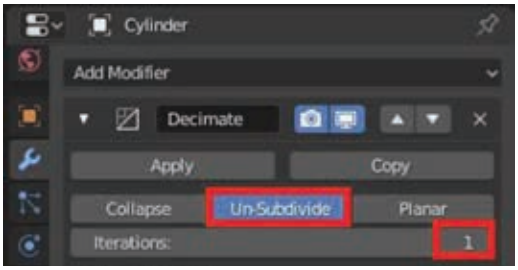


Here I created **9** more edge loops. It's easier to use uneven numbers.

You can even numbers, but you'll have to deselect the top and bottom row later.

Press TAB to go back into Object Mode. Add a **Decimate modifier** to this cylinder.

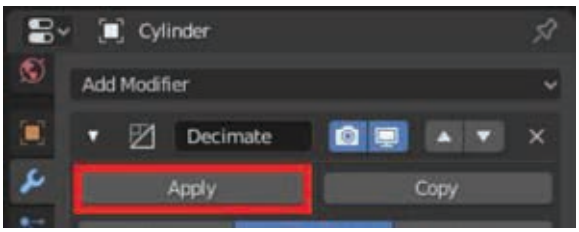
Set it to **Un-Subdivide**, 1 iteration.



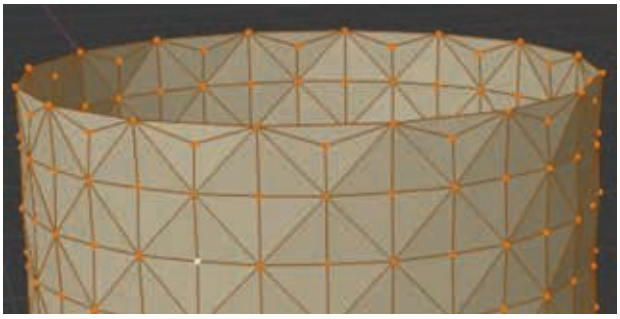
1 iteration will give you this result.

However you can also use a higher value and get different results, so experiment with that if you want.

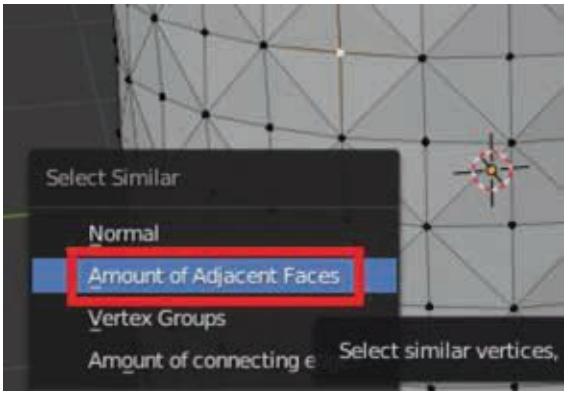
Apply the modifier. **Press TAB** to go back into Edit Mode. **Press A** to make sure everything is selected.



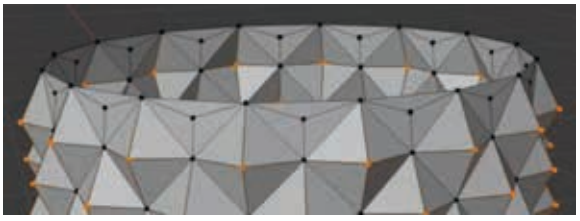
You will get this crazy mess.



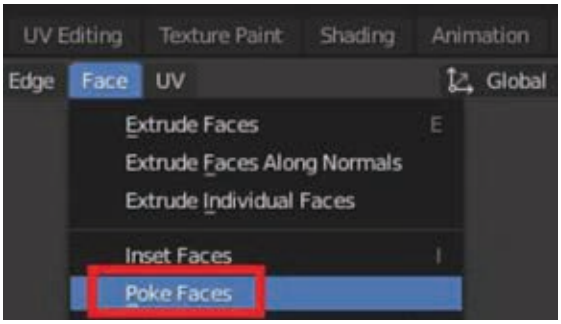
Press Shift+G and choose Adjacent Faces.



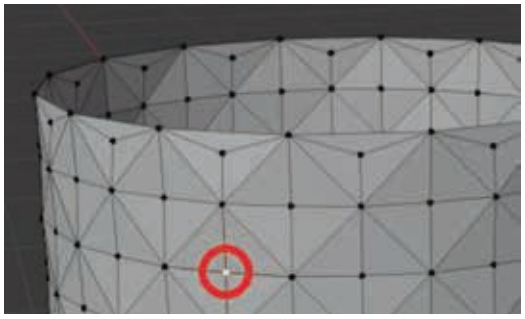
Press Alt+S and move the mouse to scale the selected vertices outwards. In Object Mode right-click on the object and choose **Shade Smooth.**



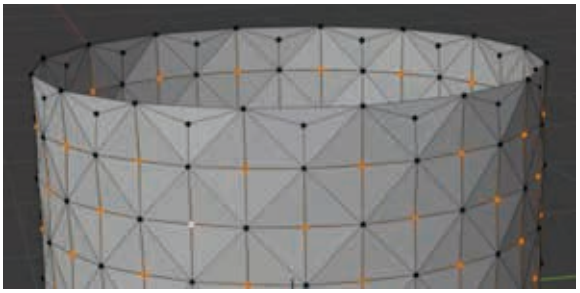
Go to Face > **Poke Faces**.



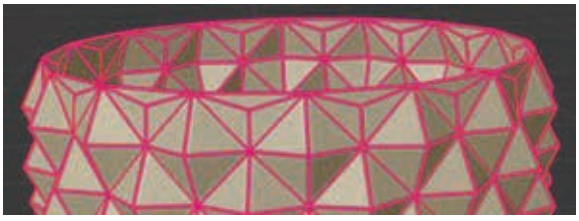
Select one vertex like the one shown.



Now your selection will look like this. If you used an even number of edge loops before, you'll have to deselect the top and bottom row of vertices at this point.

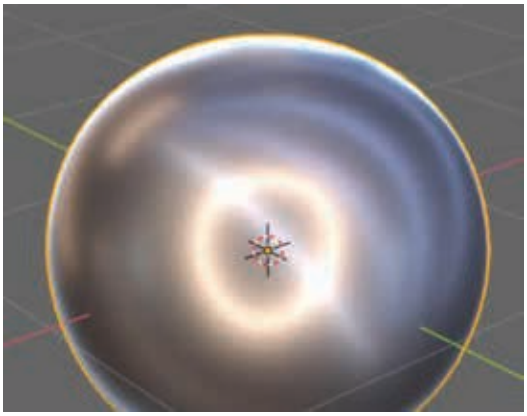


Add a Subdivision modifier. In Edit Mode **press 2** for Edge select mode and **press A** to select all. **Press Shift+E,** then move the mouse to crease the edges.

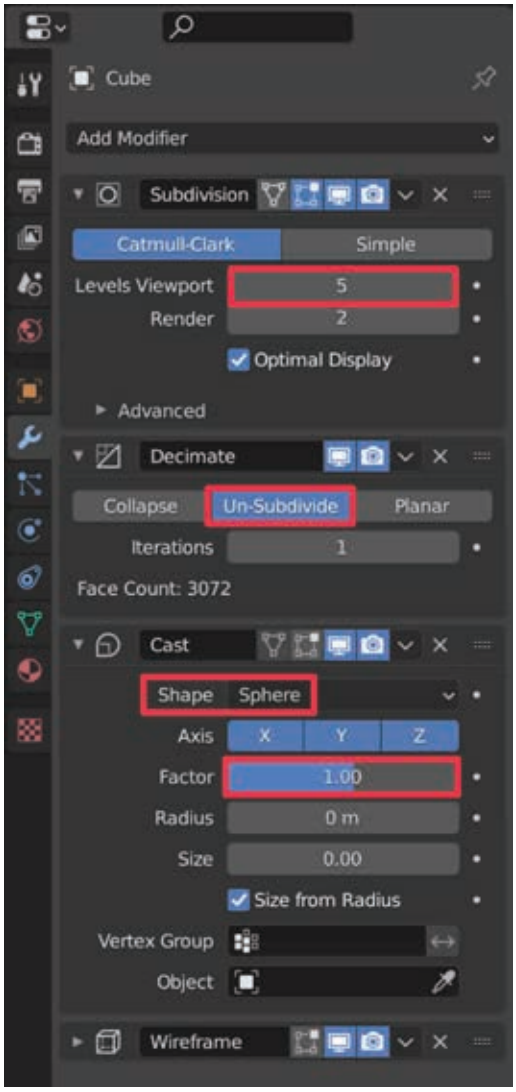


Making a microphone mesh with modifiers

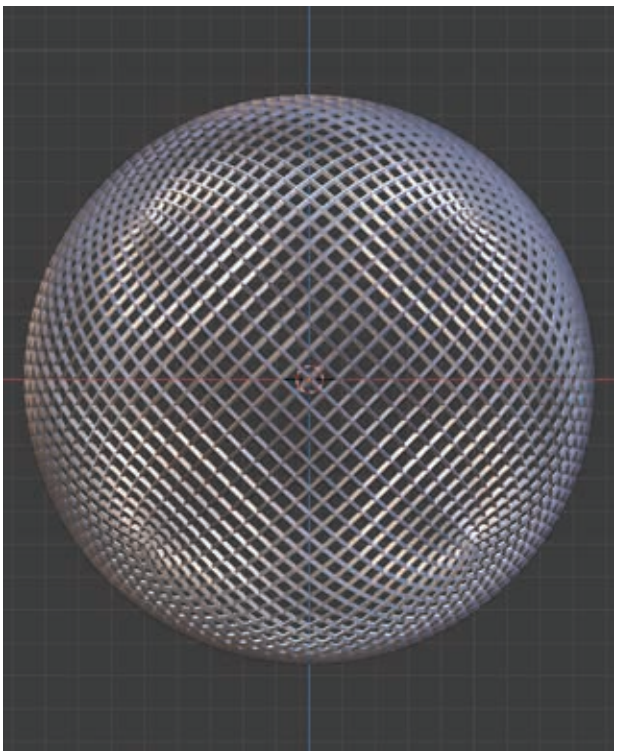
Press **Shift+A**. Add a cube. Add a Subdivision modifier to the cube. You can do this quickly by **pressing Ctrl+5** - this adds the Subdivision modifier with 5 levels of subdivision. I'm using a **Matcap** to make it look a bit more interesting in the viewport.



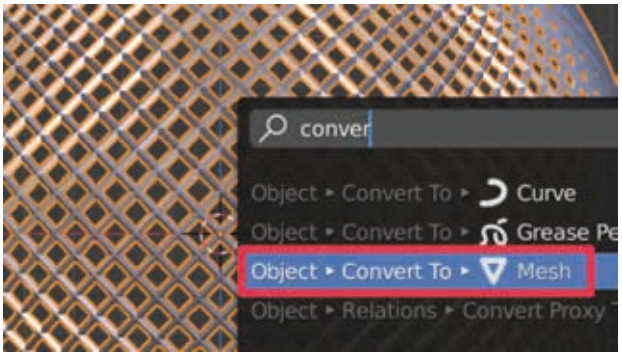
Add a **Decimate, Cast and Wireframe modifier** underneath the Subdivision modifier in this order and with these settings:



Now the cube will look like this:

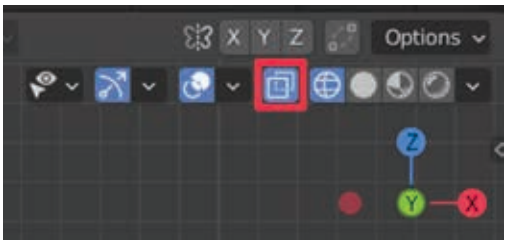


With the object selected in Object Mode **press F3** and type “convert”, choose **Convert to Mesh**. This applies to all modifiers at once.

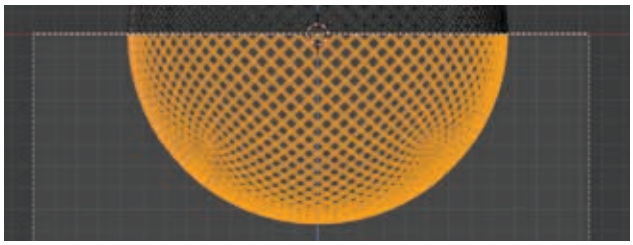


Go to Wireframe mode by **pressing Z** and choosing **Wireframe**.

Make sure **X-ray**is also turned on, so you will select faces on the back side of the object as well.

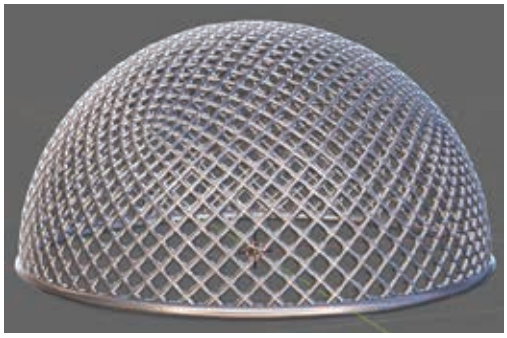


Left-click and drag a box selection to select the lower half and delete the lower half by **pressing Ctrl+X**.



Now you have the most difficult part of the microphone modeled and you can start adding more parts.

Note that this creates a fairly high-poly mesh, good for close-ups.

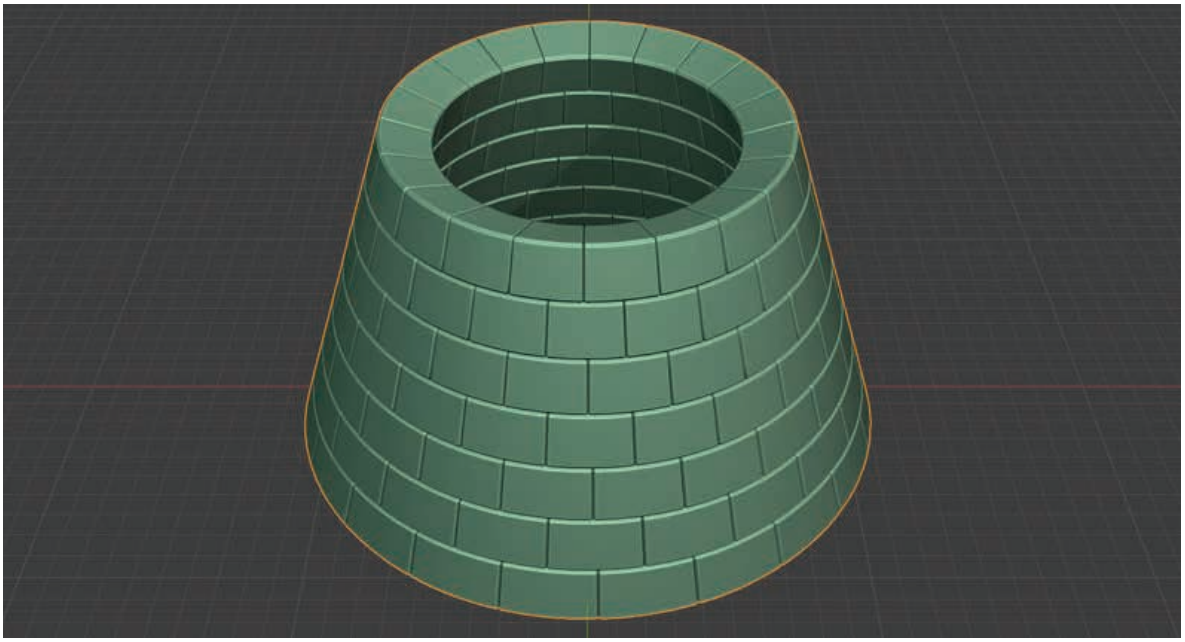


If your microphone is not so close to the camera, you could alternatively fake the holes by using a black and white pattern of dots, and using that as an alpha to control the transparency of a material. The black dots would make the material transparent.

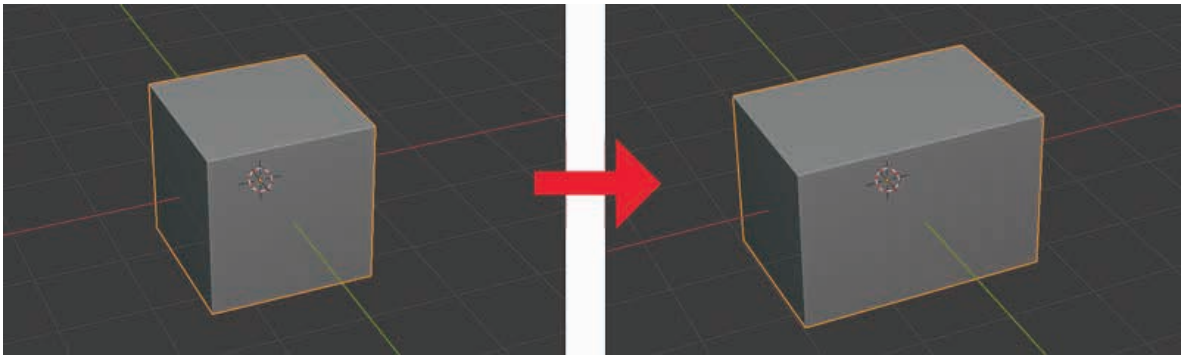
Circular Array of Cubes

Here we look at an interesting way to do a circular array using a method which is completely non-destructive and therefore very flexible.

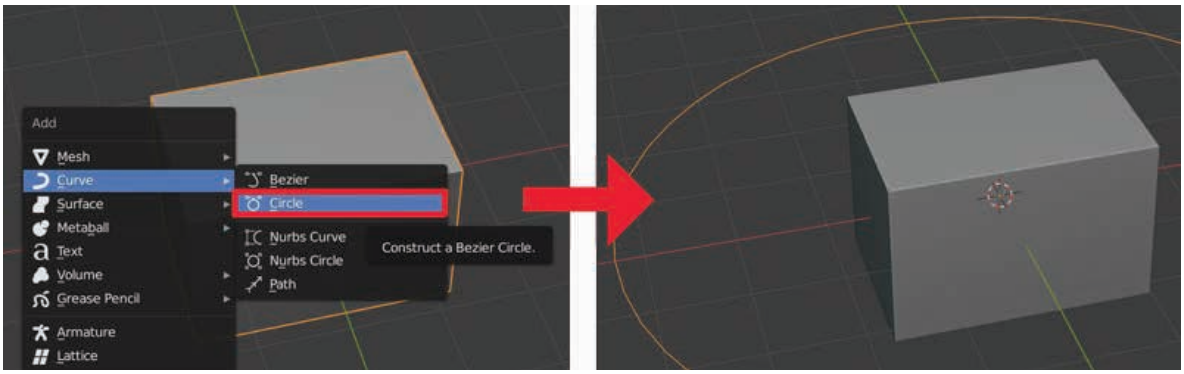
Since it deforms the objects in the array, it is especially useful for specific cases where that is required, like making a cylindrical wall like you would see in a well or a chimney.



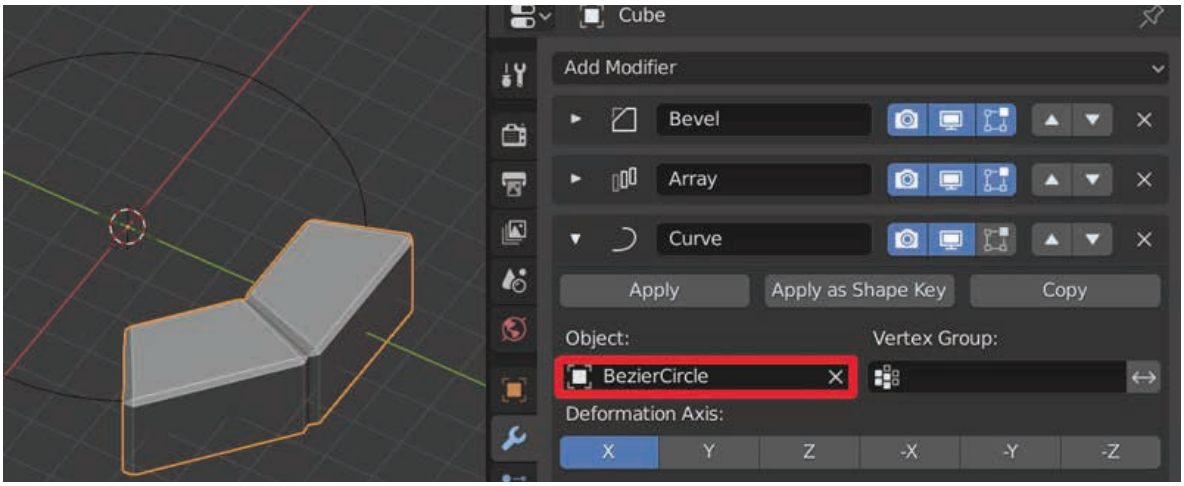
Scale the default Cube on the X-axis by **pressing S+X** until you get something brick-shaped. That's right, today the life of the default cube is spared… Be careful that you scale along the X-axis and not the Y-axis.



Create a Bezier Circle by **pressing Shift+A** and choosing **Curve > Circle**. Scale it up by **pressing S** and moving the mouse (it's probably hidden inside the Cube initially).

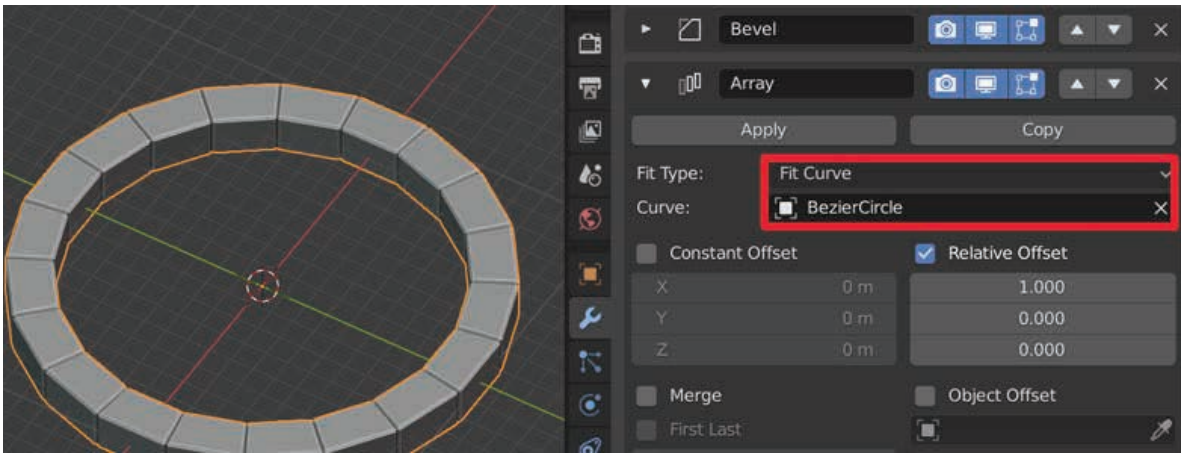


Select the Cube and add a **Bevel modifier**, an **Array modifier** and a **Curve modifier** to it. Choose the Bezier Circle as the object in the Curve modifier.

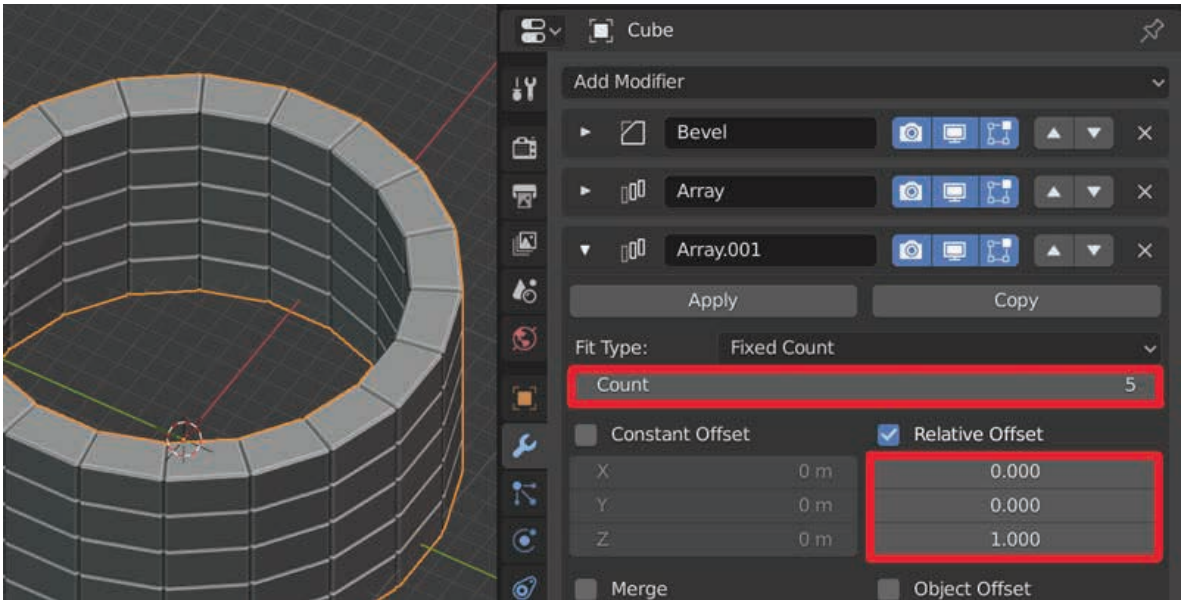


In the Array modifier you can either set **Fit Type** to **Fixed Count** and manually increase the Count until you get enough duplicates to complete the loop, or Set **Fit Type** to **Fit Curve** so the Array modifier sets the amount that is needed automatically, saving you some clicks. You can increase the scale of the Curve if needed.

Select the Bezier Circle as the **Curve object** if you use the **Fit Curve** option. Scale the cube slightly to close the last gap of the loop.



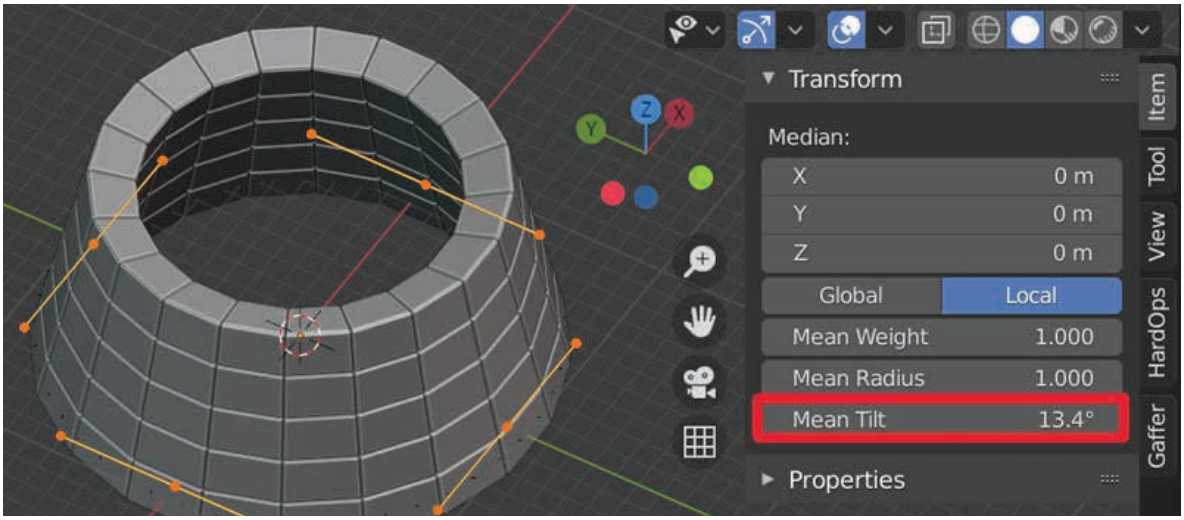
Add another **Array modifier** to stack the bricks vertically by Setting the Z-value of the **Relative Offset** to 1. Increase the **Count** value until you get the height you want.



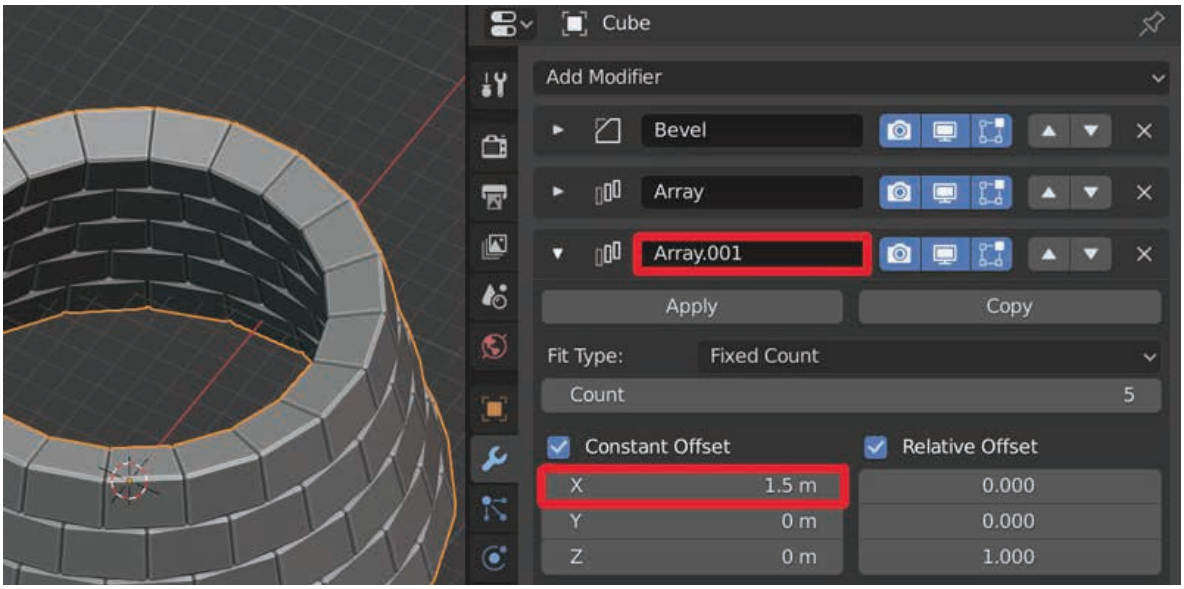
If you want to have this construction taper towards the top, select all the vertices of the Curve in Edit Mode by **pressing A** and then "tilt" it with **Ctrl+T** until you get the result you want. The easiest way to select the Curve is in the **Outliner**, since it is hidden from view in the Viewport.

This value is called the **Mean Tilt** and can also be found in the Option panel when in Edit Mode.

Tip: In case the Mean Tilt doesn't taper your stone cylinder, it's most likely because the Curve modifier is not below the two Array modifiers but sandwiched in between them.

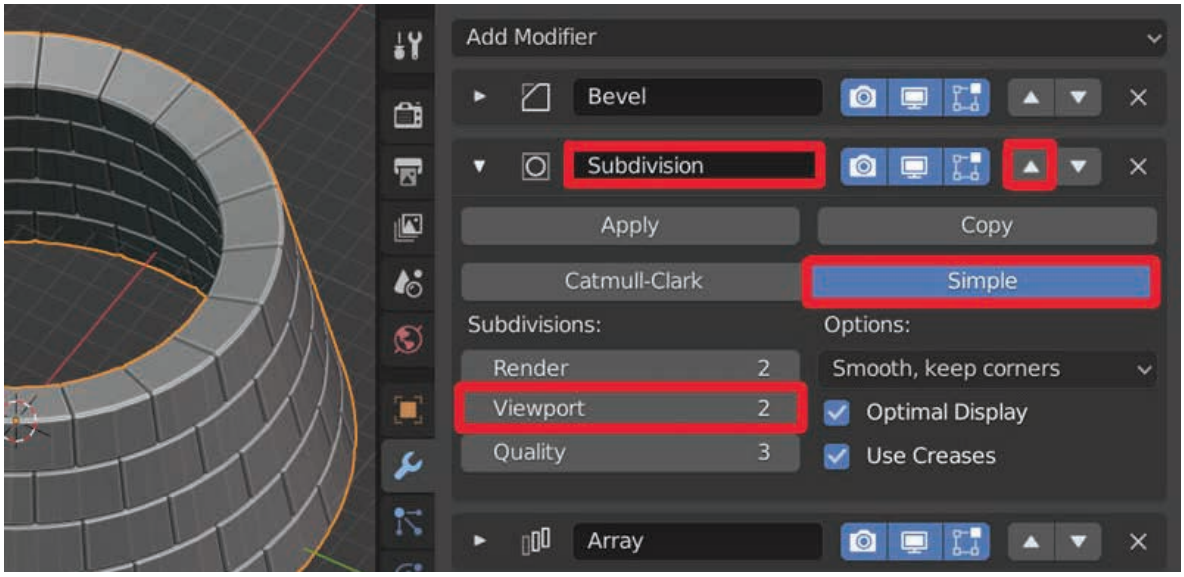


You can offset the stones by turning on **Constant Offset** in the Second Array modifier and increasing the **X value**.

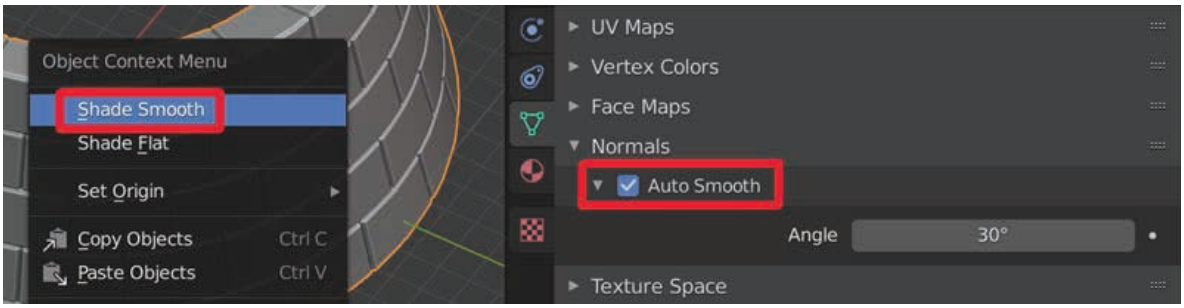


If you want the stones to follow the curve more smoothly, add a **Subdivision modifier** between the Bevel and Array modifiers. You can move the modifier up the stack after adding it by pressing the up-arrow button. Set it to **Simple** and increase the Viewport amount to 2.

***Interface tip**: In Blender 2.8, to move the modifier you can press the **Up arrow** once and then repeat this by **pressing Shift+R** as many times as needed. I prefer to do it this way because it avoids accidentally clicking on the "Copy" button on the modifier... However, in Blender 2.9 you can simply drag the modifiers up and down.*

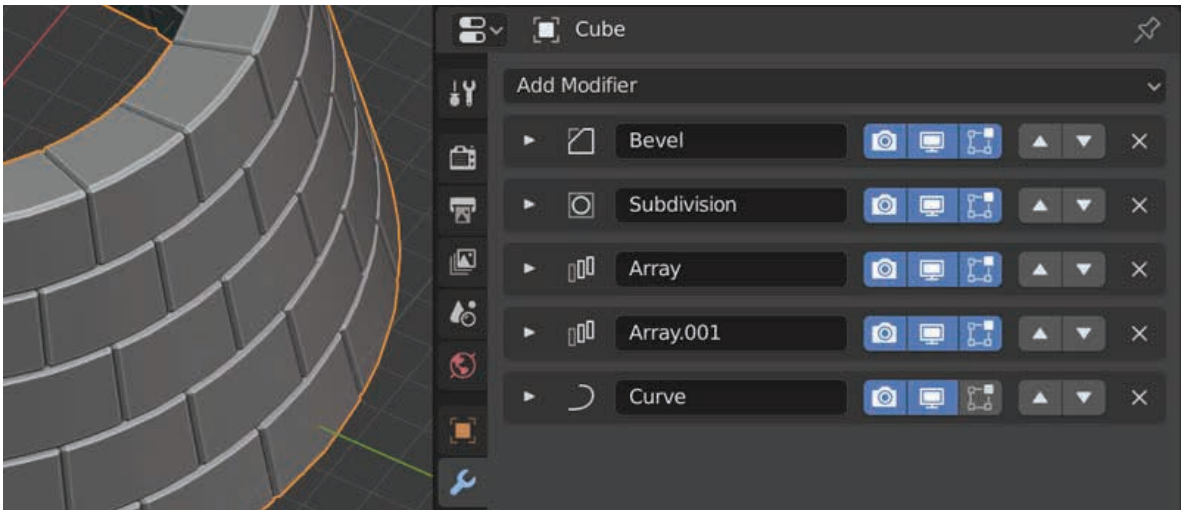


You may want to turn on **Smooth Shading** (right-click on the object and choose Shade Smooth) and **Auto Smooth** at this point.

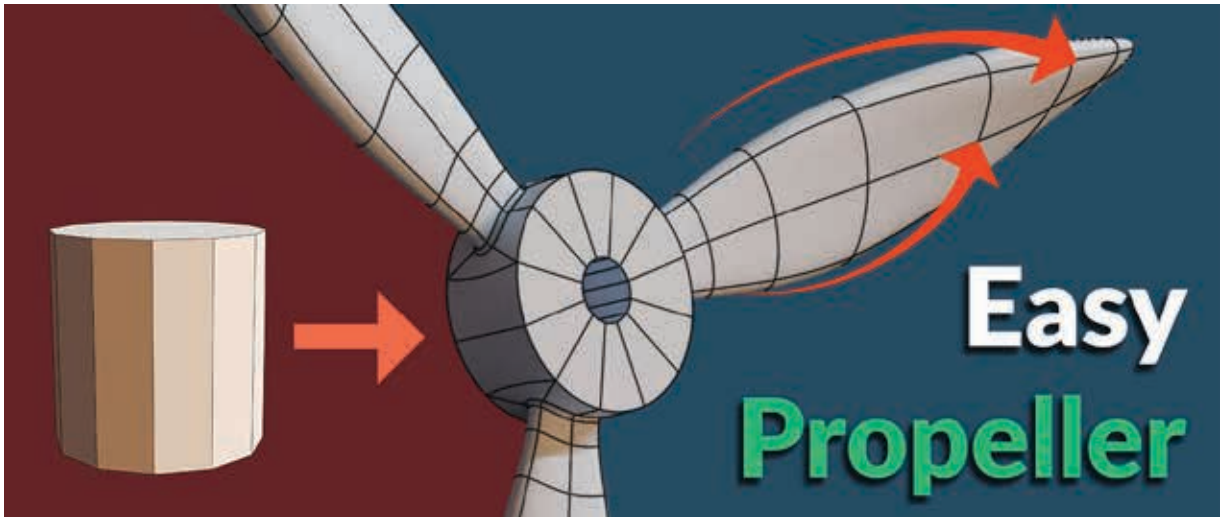


Adding a **Subdivision modifier**set to **Simple** creates more geometry so the cube can bend more smoothly around the Curve. Alternatively, in Edit Mode add more Edge Loops by **pressing Ctrl+R** and scrolling the mouse wheel up. The benefit of using the Subdivision modifier is that it keeps everything non-destructive, allowing you to still change things easily later.

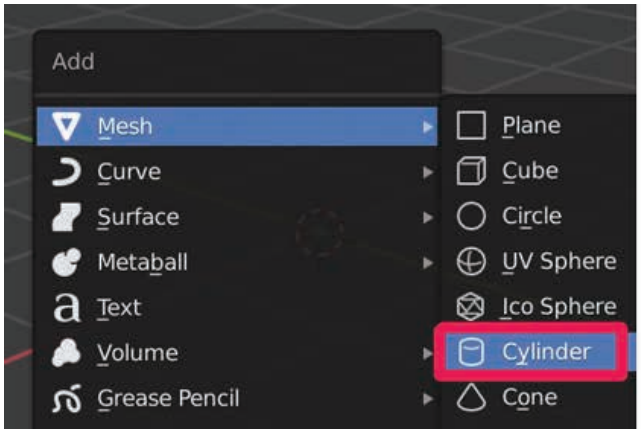
Make sure to arrange the modifiers in the right order from top to bottom: Bevel, Subdivision, both Array modifiers and finally the Curve modifier.

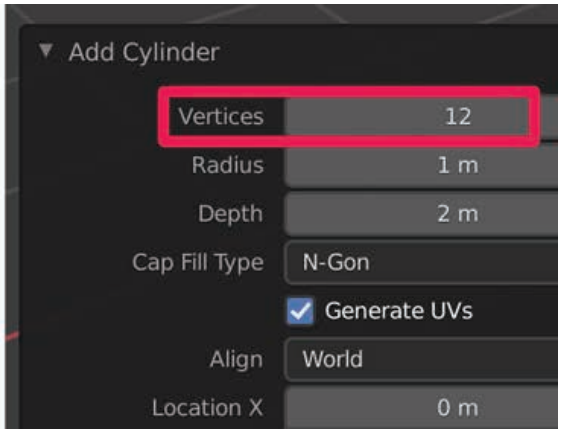


Modeling Propeller Blades

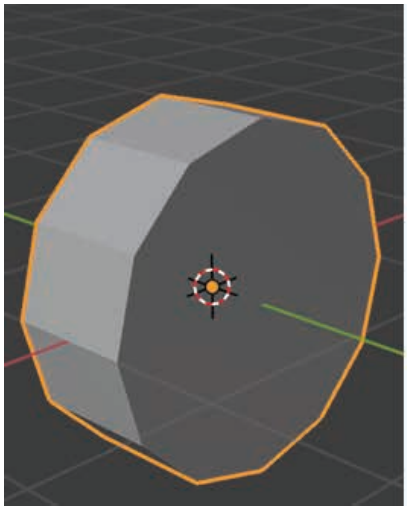


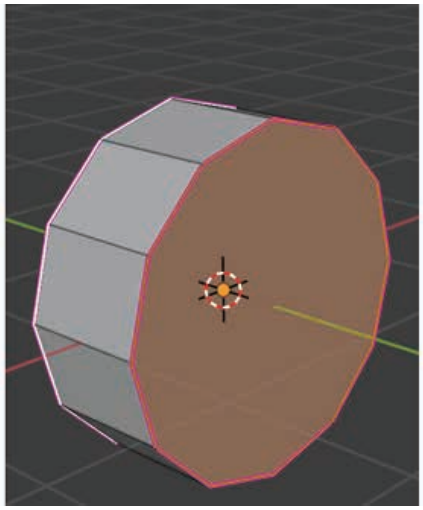
Add a Cylinder with **Shift+A > Mesh > Cylinder** and set it to 12 vertices.

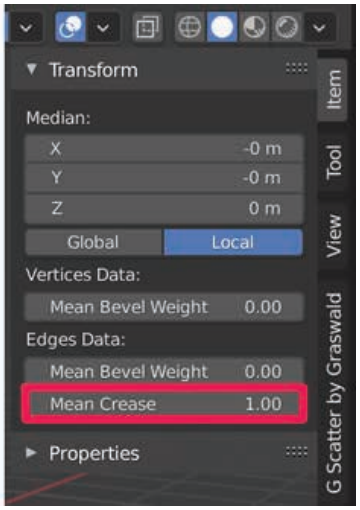




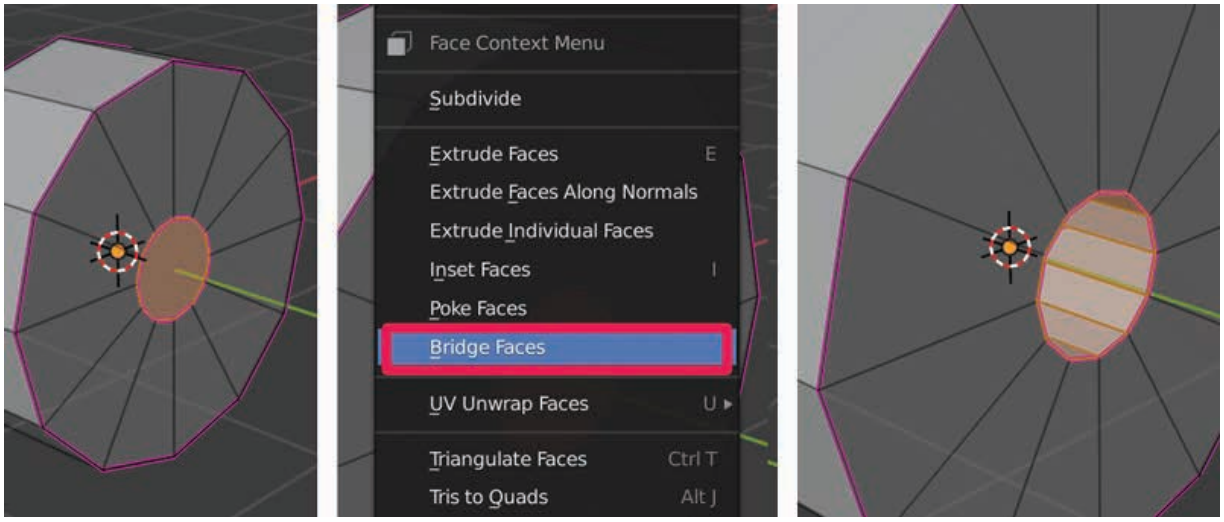
Rotate it 90° on the X-axis and scale it on the Y-axis. Select the side Faces and set Mean Crease to 1.



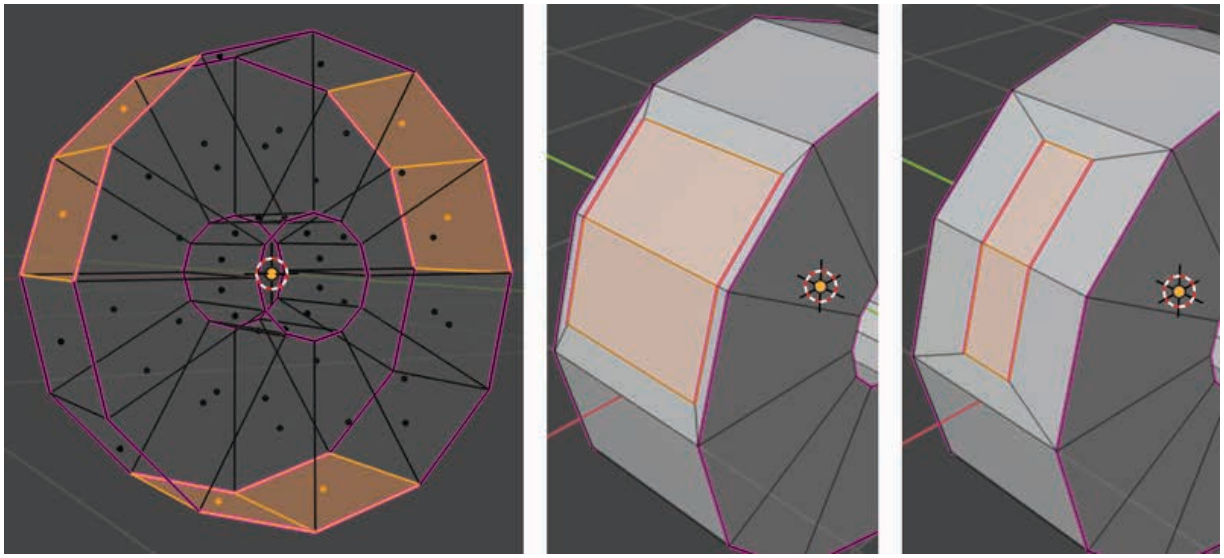




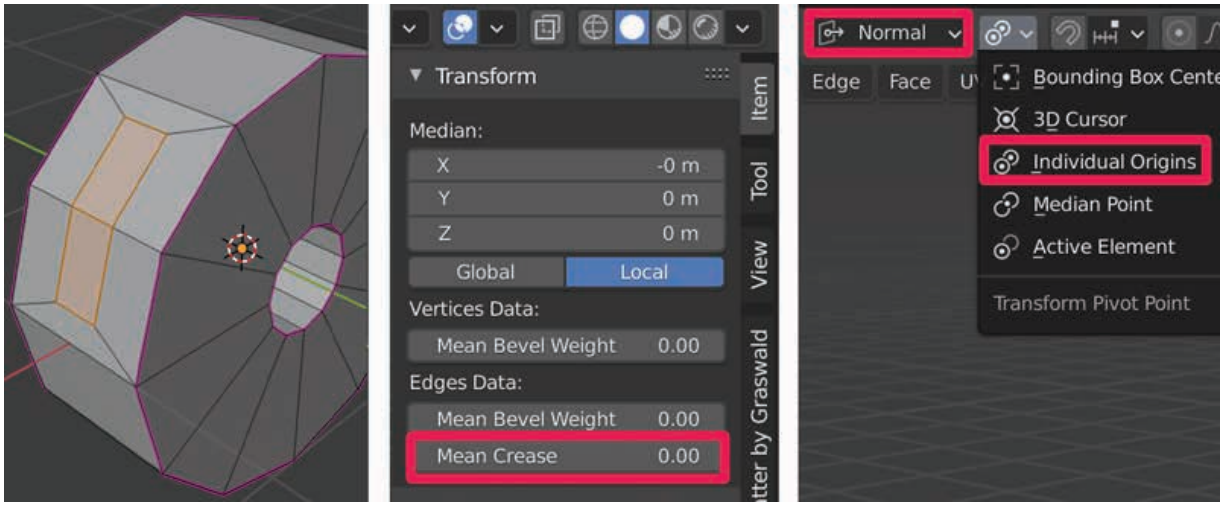
Press i to inset, then **Bridge** the selection to make a hole.



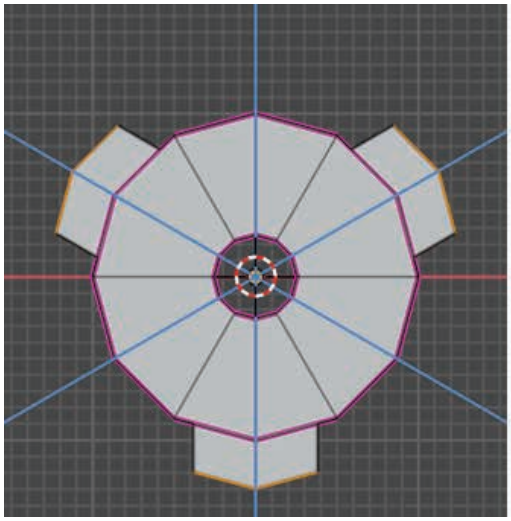
Select the faces you want to extrude propeller blades from. Inset the selected Faces and scale them on the Y-axis.

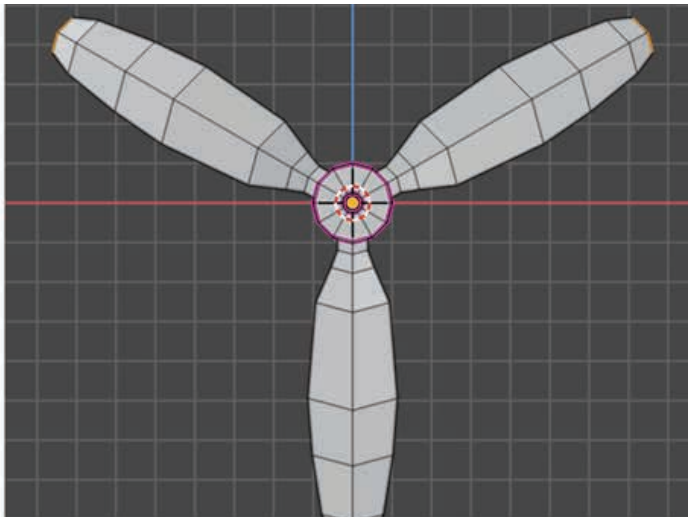


Set **Mean Crease** to **0**. Set Transform Pivot Point to **Individual Origins**. Choose **Normal** as the Transform Orientation.

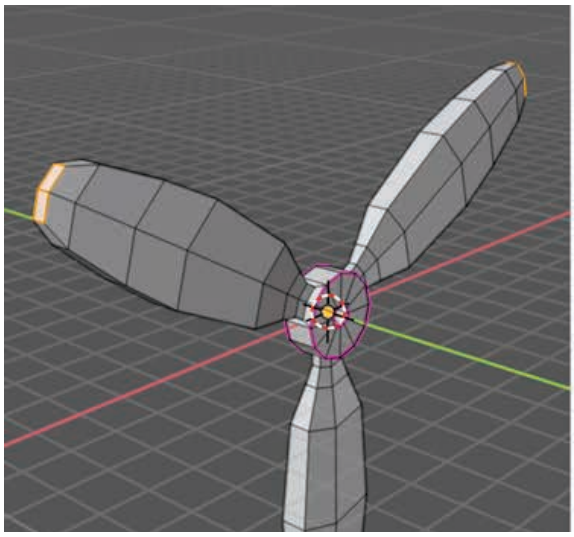


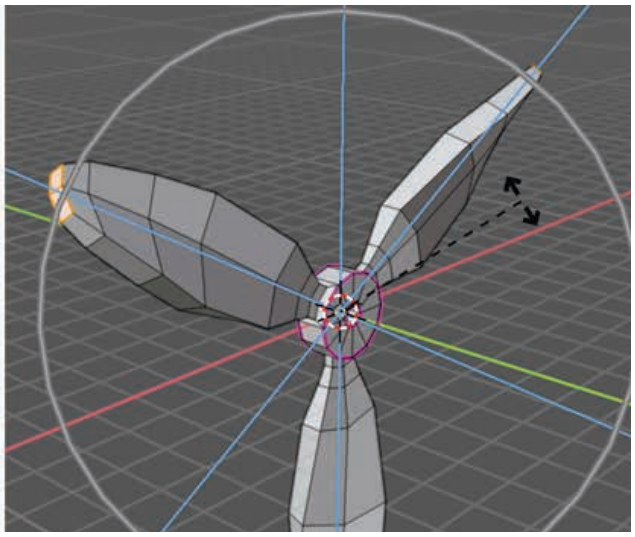
Press E and move the mouse to extrude the beginning of the propellors. Extrude and Scale to create the rest of the blades.



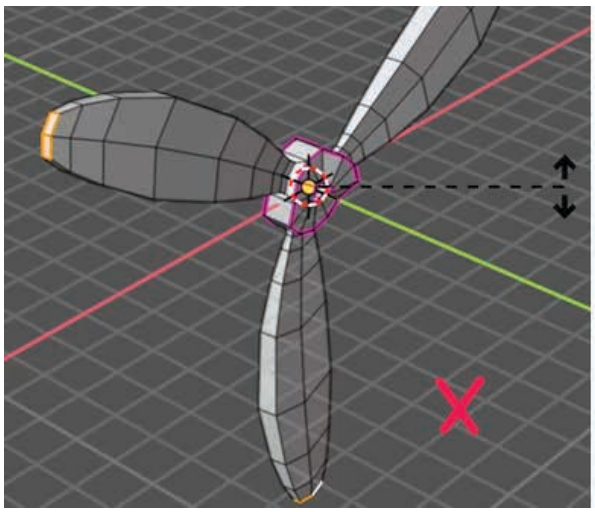


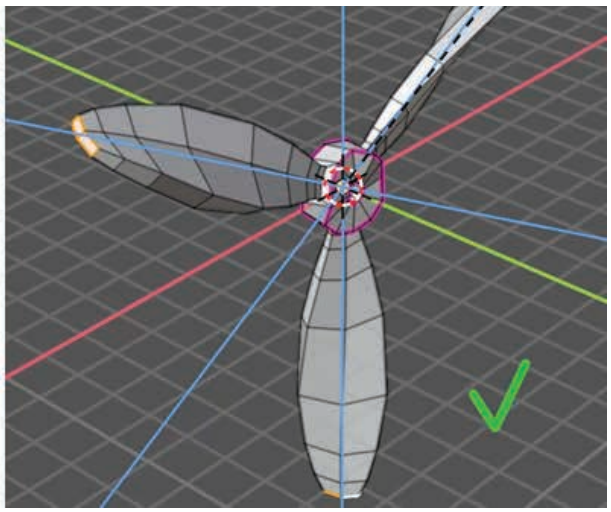
Turn on Proportional Editing by **pressing o** (the letter o, not zero) and rotate the blades. Increase the influence radius with the middle mouse wheel.





If the blades don't rotate this way but get all wonky, **press Z** once or more to cycle through different orientations until you get the correct one.

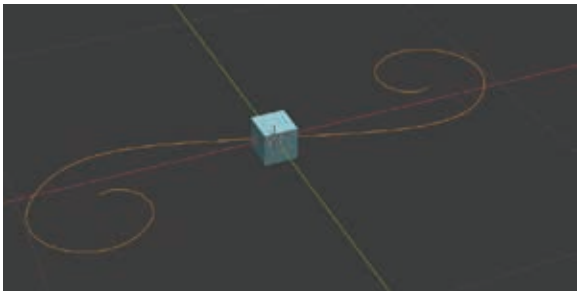




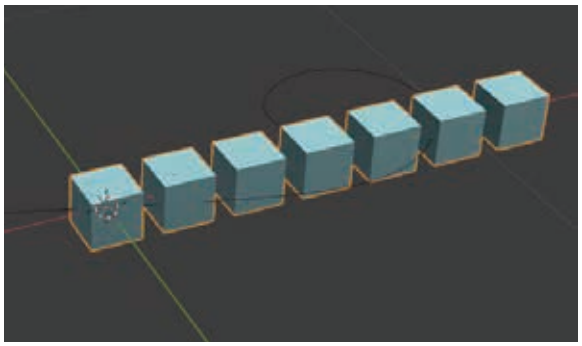
Array along a path

When creating an array of objects along a Curve, the objects are slightly deformed along the curve of the path they are following.

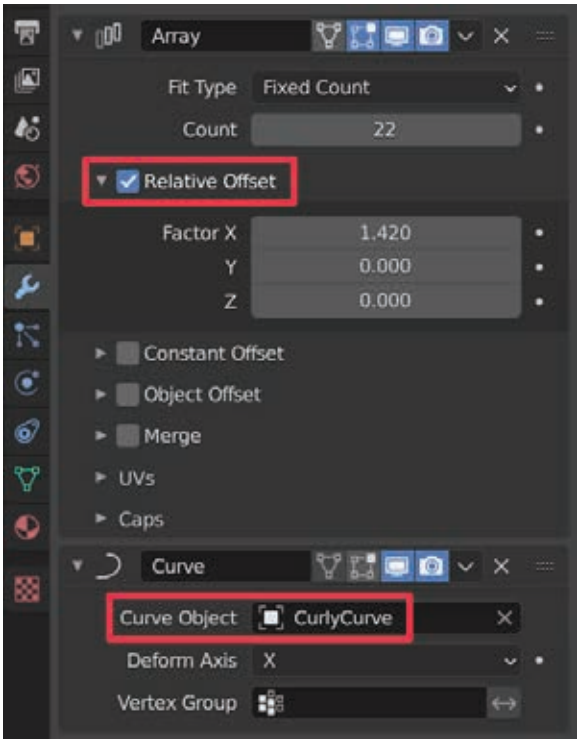
This can be useful in some cases. This Curve is a **"Curly Curve"**, which is added to the primitives if you enable "**Add Curves**" in add-ons.



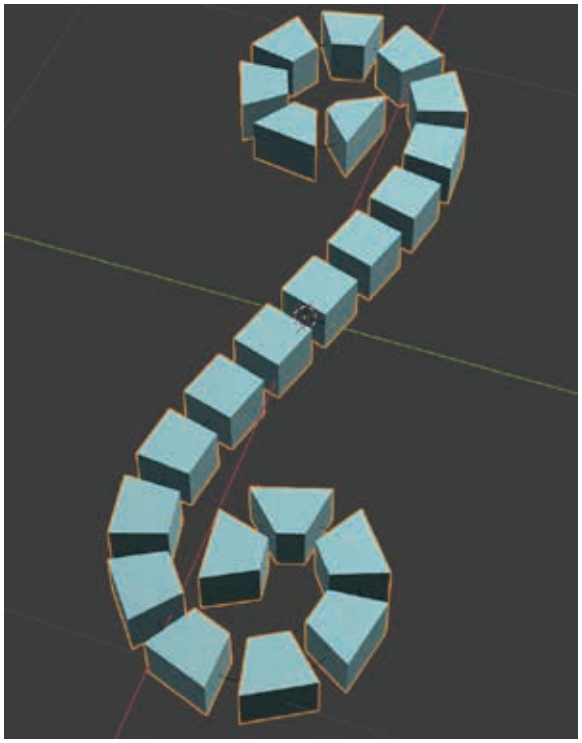
Add an **Array modifier** to the Cube. Increase the **Count** and **Relative Offset.**



Add a **Curve modifier**, choose the Curve. You may have to adjust the **Deform Axis**.

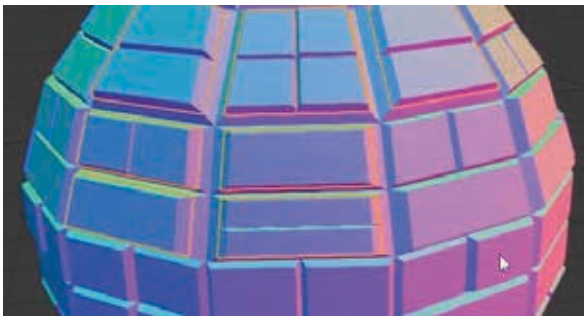


Now the Array follows the Curve. Increase the **Array Count** if needed.



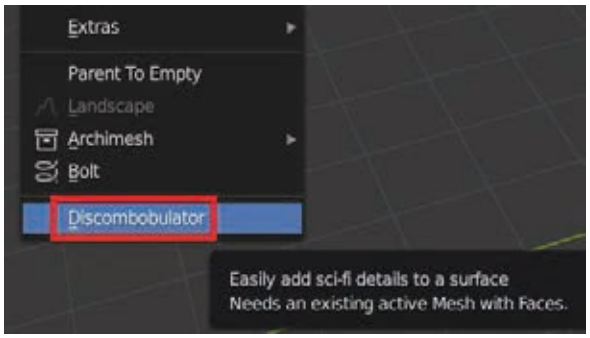
Discombobulator

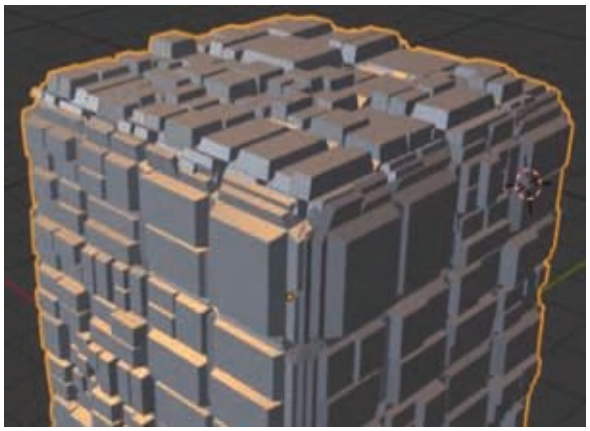
The **Discombobulator add-on** comes with Blender by default. It adds some random sci-fi details to any mesh object.



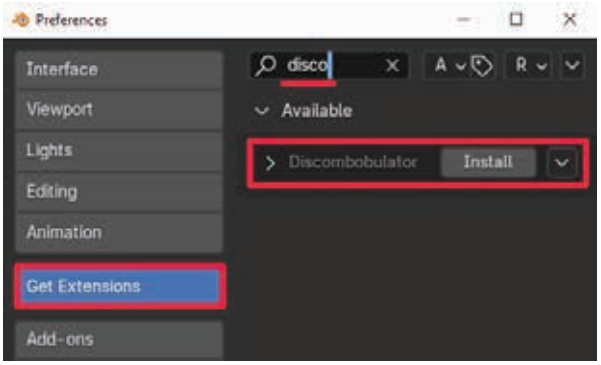
To use it, **press TAB** to go into Edit Mode and select all the faces of an object that you want to have discombobulated. **Press Shift+A** and select the option from the create menu.

If that seems odd, it's because you are adding geometry to the model - the discombobulator actually creates a separate mesh.



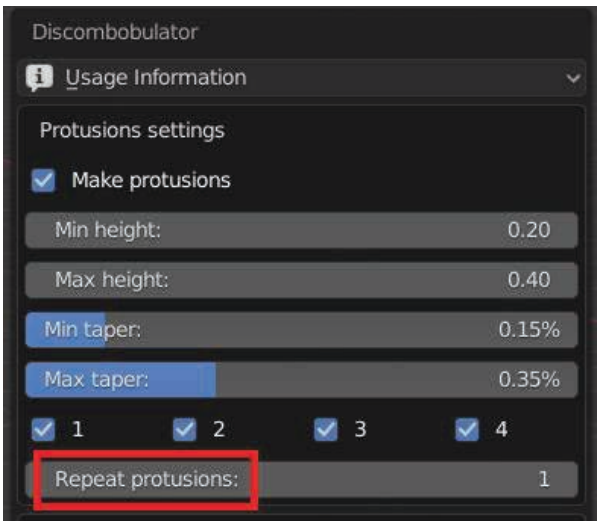


Make sure the add-on is enabled. You can find it in the **Get Extensions** tab.



It has plenty of values that you can tweak to get just the right… combobulation. Okay, let's call them **greebles**, which is a more common term for these random "sci-fi details".

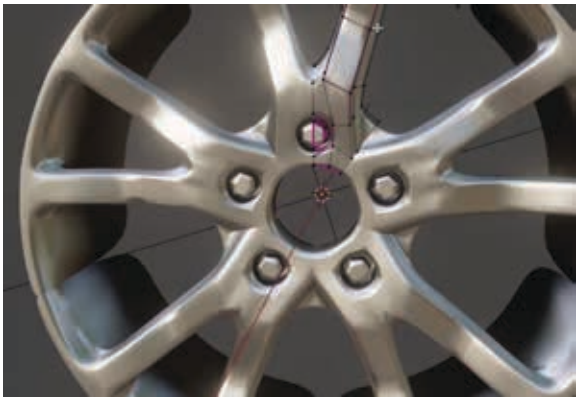
Min and Max height controls the extrusion height. Min and Max taper controls the tapering of the extrusions. One of the most interesting options is the (misspelled) **Repeat protrusions** button. This creates greebles upon greebles, so it quickly creates more detail in other words.



On left you'll see the Borg cube, I mean default cube, with some protrusions. The protrusions value needs to be set higher than 1 for it to do something.

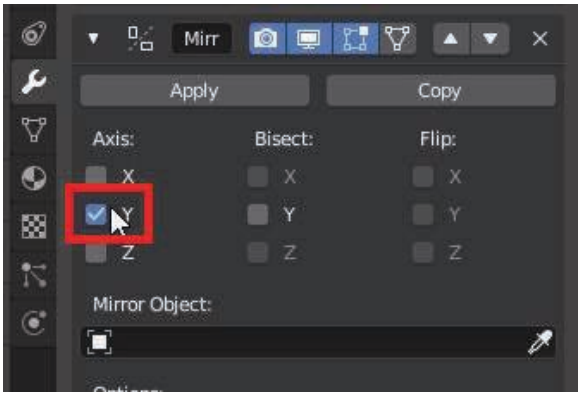
Align Car Rim Parts

In this tutorial we'll look at making a car rim by just modeling a small part and duplicating it.



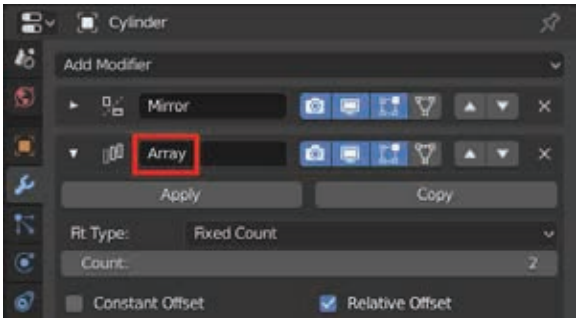
Model one tenth of the rim and add a Mirror modifier.

Set the correct Axis (Y in this case, but it may be different for you depending on the rotation of you object).

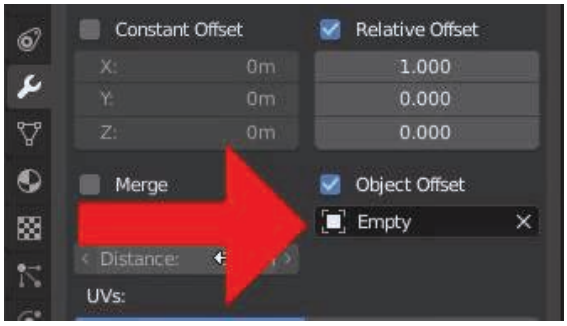


Set the correct Axis (Y in this case, but it may be different for you depending on the rotation of you object).

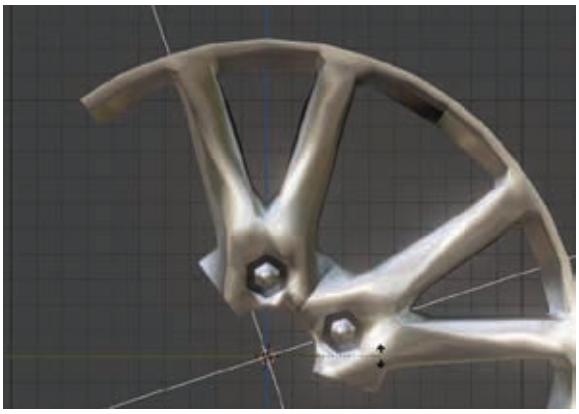
Add an Empty and use it as the Array Offset Object.



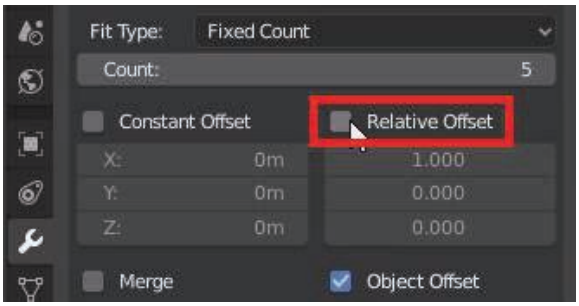
Add an Empty and use it as the Array Offset Object.



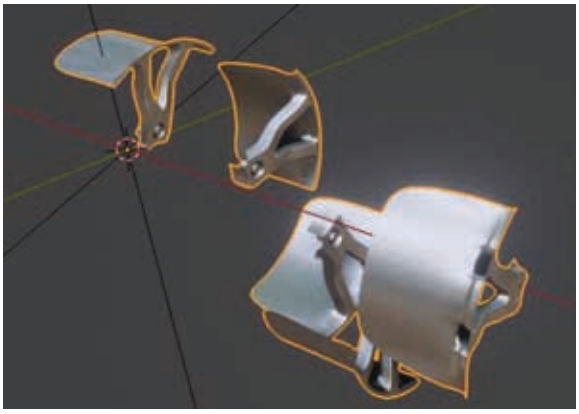
Rotate the Empty 72° and increase the array count to 5. To rotate something a specific angle, press R, the axis of rotation (Y in this case) and then the number, so **R+Y+72** in this case, looking at it in the front orthographic view.

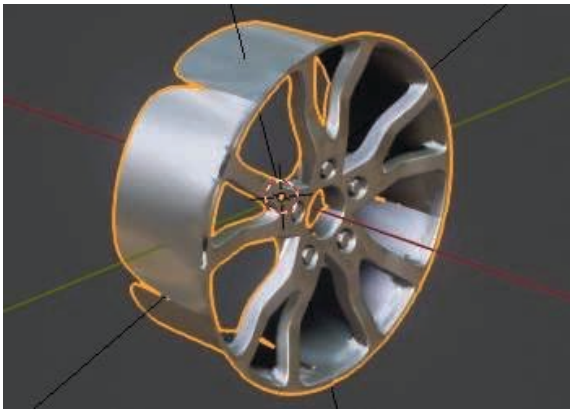


Turn off **Relative Offset** in the array modifier.



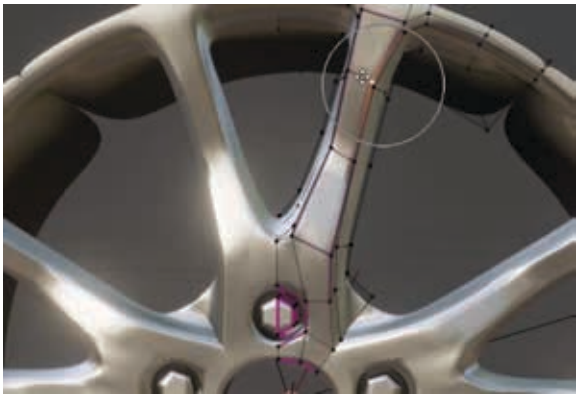
This is to make sure the duplicates are only rotated, not moved (see below left/right for the difference)

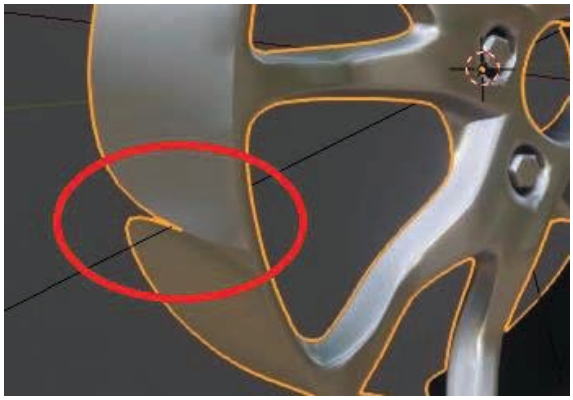




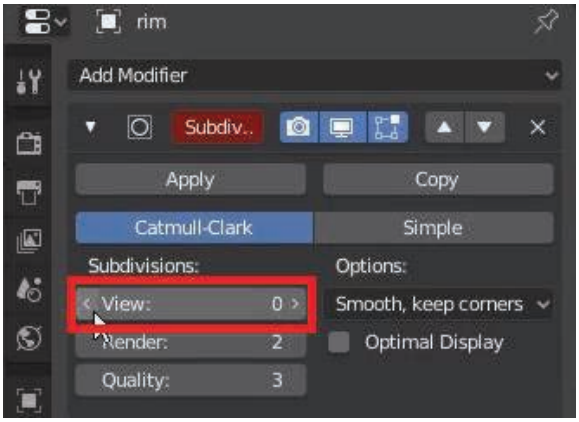
This is a non-destructive workflow. As long as you have these modifiers set up, changes to the original object will also be visible in the duplicates.

It is non-destructive until the end, when you have to remove doubles with the Merge by Distance command. That is so you don't get weird gaps like this:

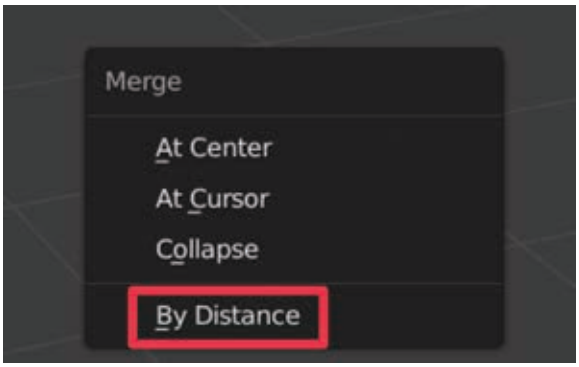




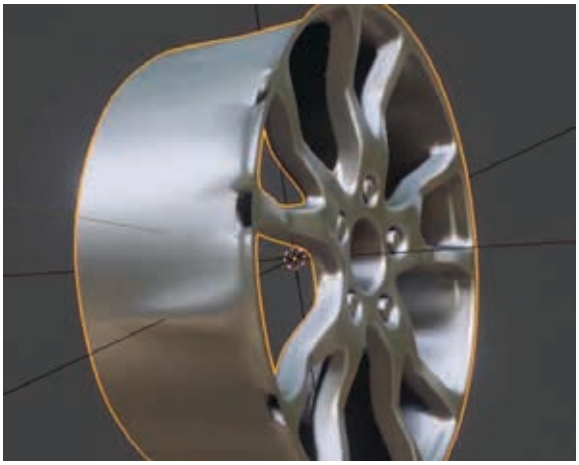
Turn off the Subdivision surface modifier by setting the View value to 0 and Apply the other modifiers.



Press M, choose Merge **By Distance**.



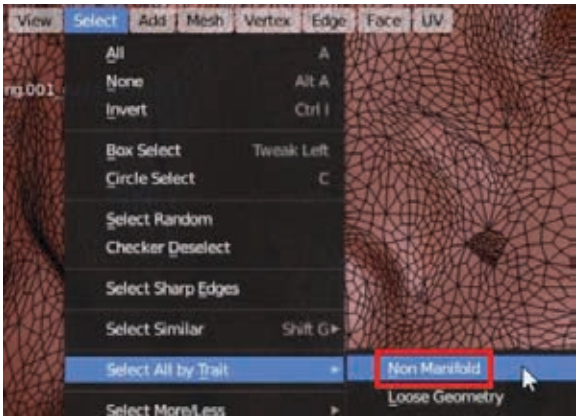
Now the vertices are merged, removing the gaps.



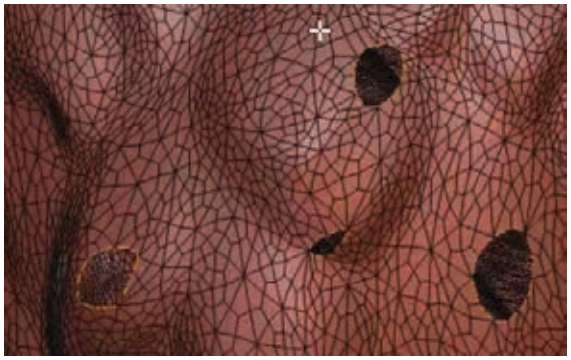
Filling many holes at once

This model, a 3D scan, has many little holes that need to be fixed.

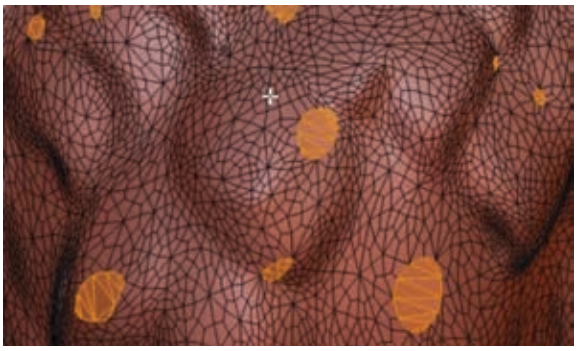
In Edit Mode, Select > By Trait > **Non Manifold**.



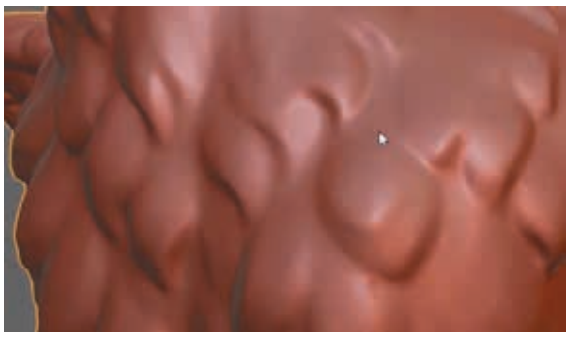
Now the edges of the holes are selected. **Press Alt+F**.



This fixes the holes. The new faces are shaded Flat, which makes them stand out.



Right-click and choose **Shade Smooth**. Now those new faces are also shaded smooth.



Making welds in two ways

The hard way to make welds is to model them by hand. Model a cylindrical ring around a cylinder.

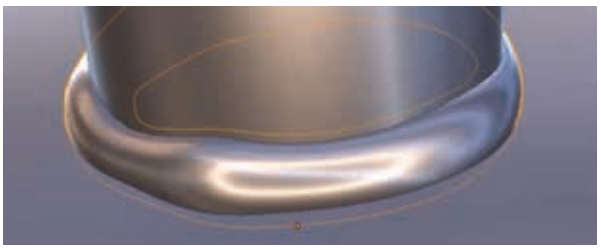
Play with the noise size and modifier value to get the result you want.

It adds a little menu on the **Tool panel** (open with N). Just select two intersecting objects and press **Weld**.

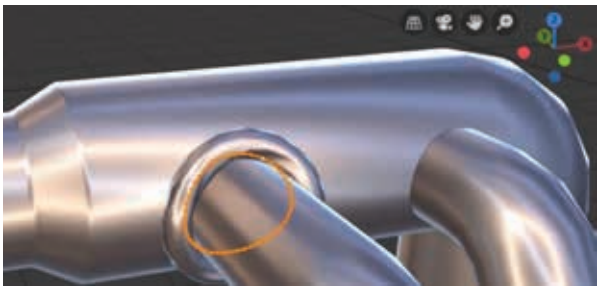
It's made with a curve so you need to press **Alt+S** to scale it.

More information as well as the download link here. This has been tested in **Blender 4.3** and still works.

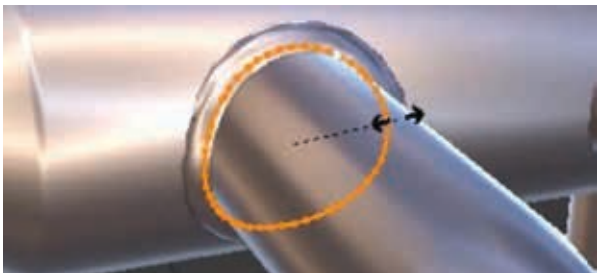
Add a **Displacement modifier** to make it uneven and a **Subdivision modifier** to make it smooth.



But there's an easier way to do this, using the Welder add-on. See below for more info.



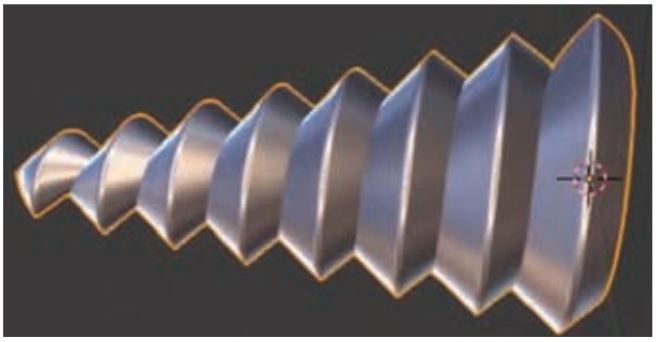
You may have to press Rotate weld, if the direction the add-on chooses is wrong.



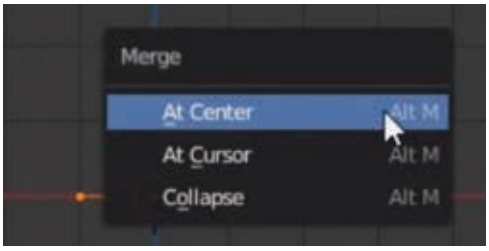
Click here to return to the index.

Making a screw the easy way

Making screws is easy with this technique.

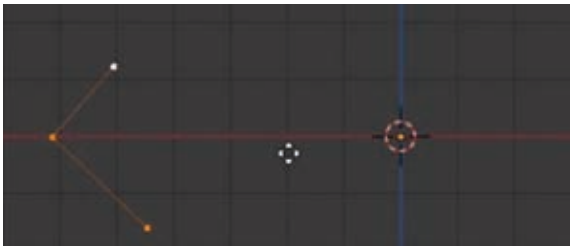


Press Shift+A and add a **Plane**. In Edit Mode select all (A) and merge vertices to center **pressing M**.

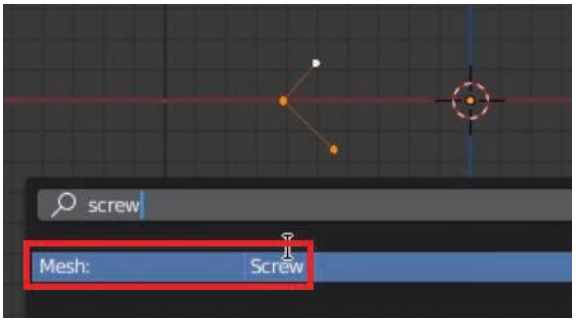


Extrude 2 vertices (**press E** to extrude).

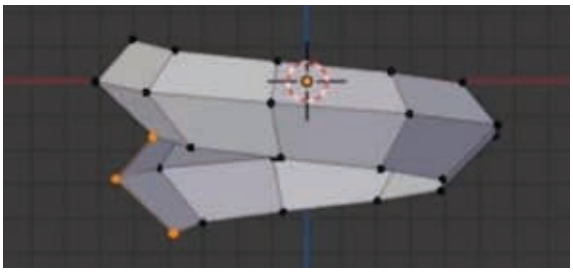
Move the vertex at the bottom further down and to the right.



Press F3, search for **Screw**.

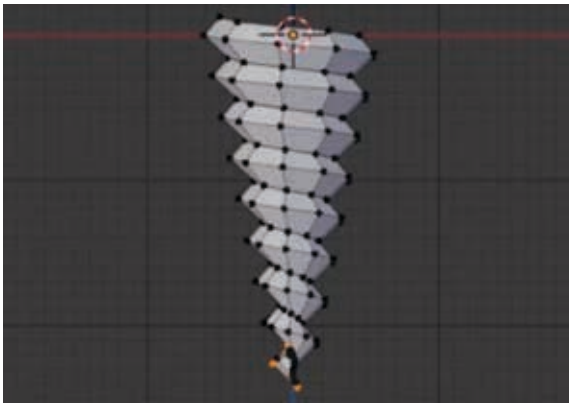


This turns the vertices into the first part of the screw.



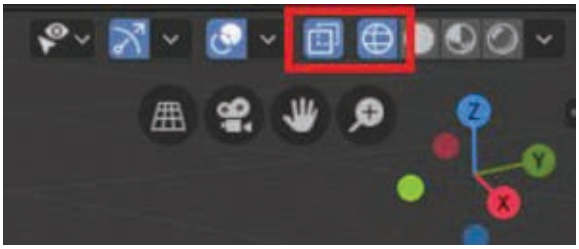
Pressing Shift+R will repeat the Screw command.

Repeat it again, until the entire screw is made.

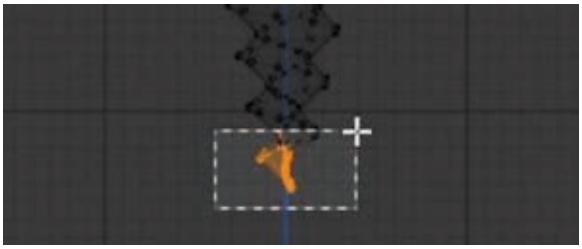


Click here to return to the index.

Turn on Wireframe and X-ray so you can select through the object.



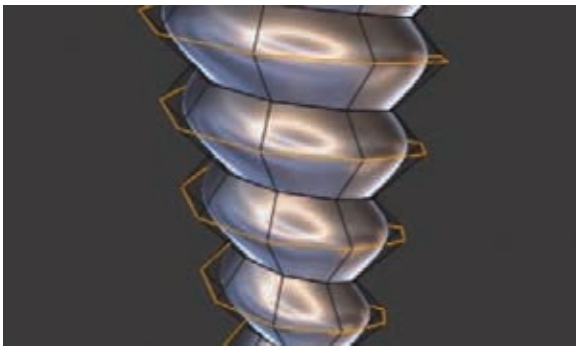
Select and merge the bottom vertices by pressing **M**.



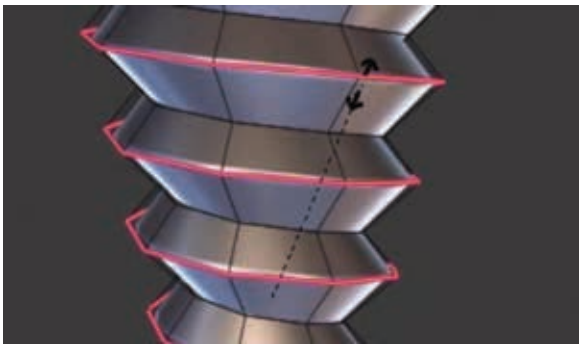
In Object mode **right-click** and choose **Smooth Shading**. **Press Ctrl+2** to add two levels of subdivision.

It's nice and smooth, but what we really want is for the edges to be sharper.

Select the middle edge loop by **holding Alt and double-clicking** on an edge.

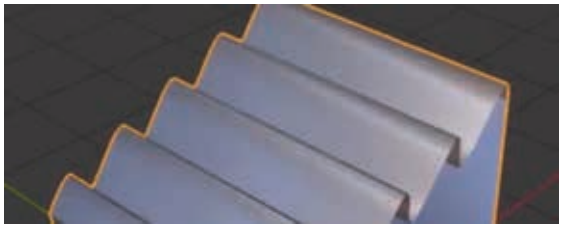


Press Shift+E+1 to crease it. That's it, all done!



Math Mesh add-on

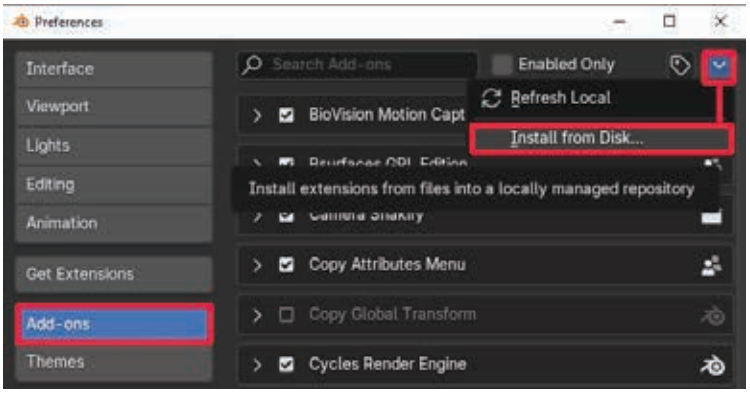
Math Mesh is an interesting add-on made by Ryan Southall. It allows us to use mathematics to create shapes such as waves.



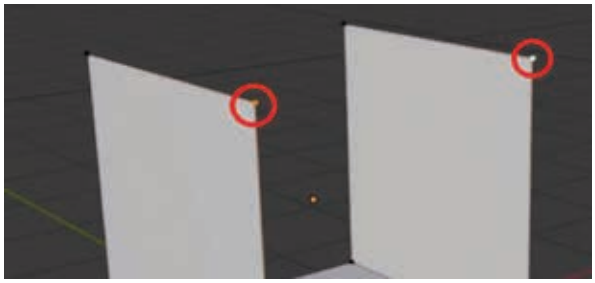
Download it from his site, or from Github.

Don't unzip the file, instead choose **Install from Disk** in Preferences.

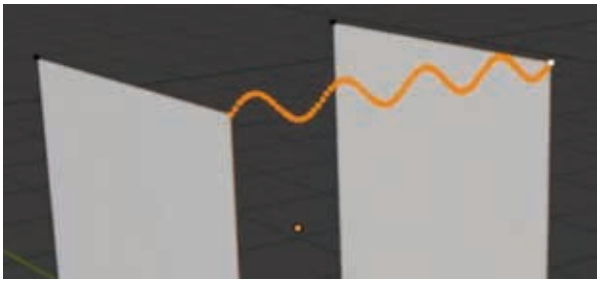
After that, it will be automatically enabled.



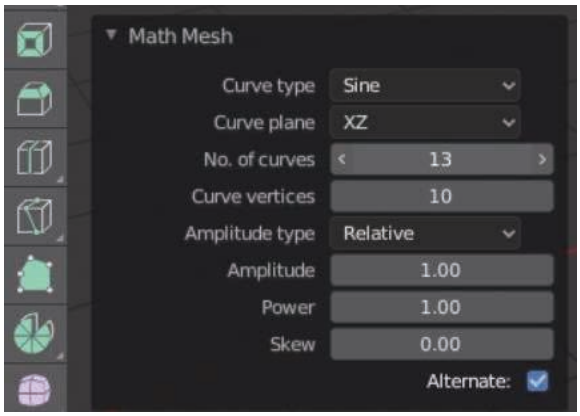
Select two vertices, press **Alt+Shift+M** to activate the add-on.



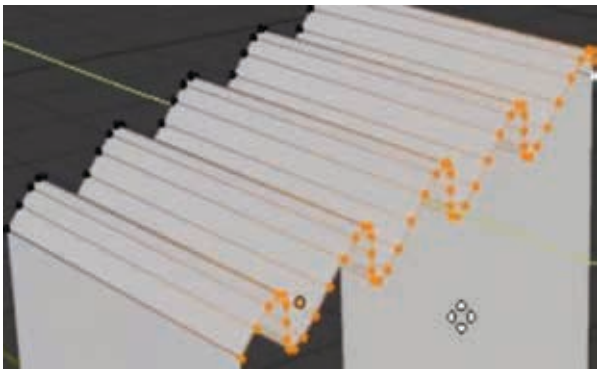
As you can see it's created a wave of vertices in-between.



You can play with the settings and see the results.

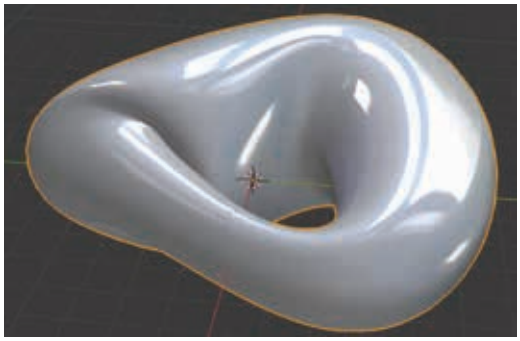


It's a good way to make roofs for example. Extrude by **pressing E**, snap to vertex (**hold Ctrl** to snap), remove doubles.

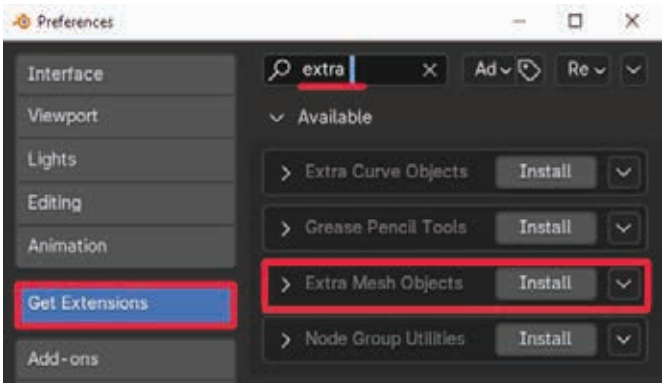


Modeling with math

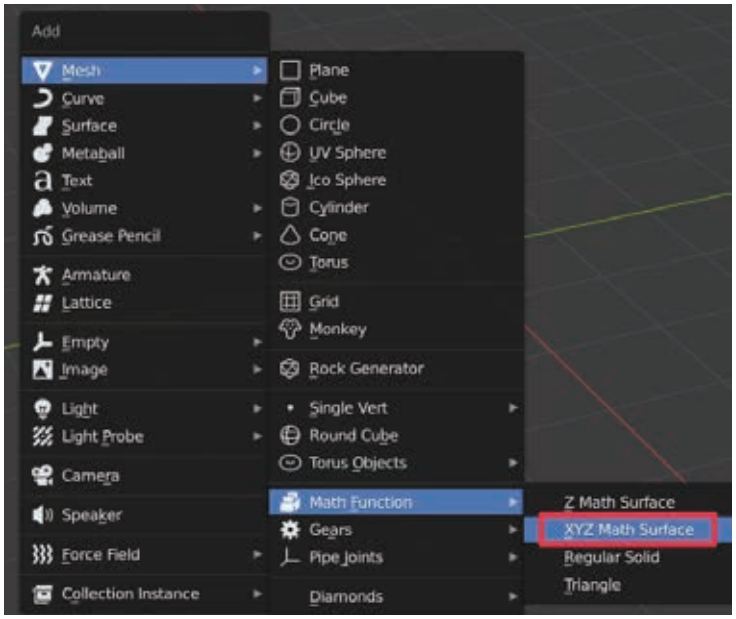
Some very interesting shapes can be created by using Mathematics.



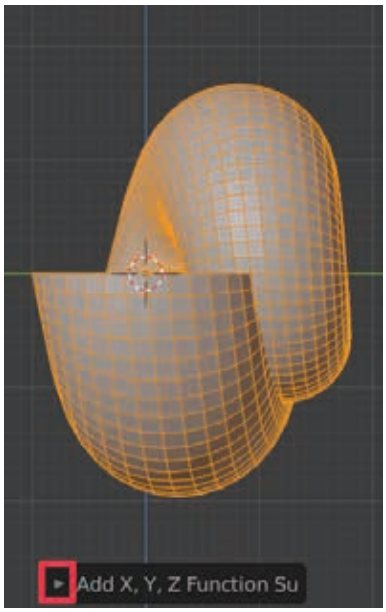
In preferences, type Extra to find the **Extra Mesh Objects** extension. Install it.



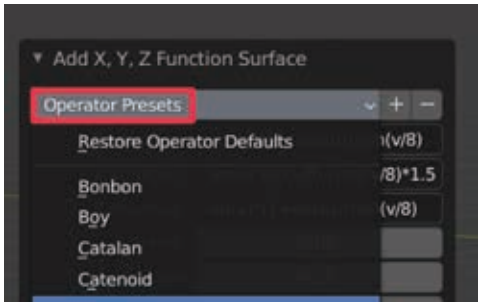
Press Shift+A and go to Mesh > Math Function>**XYZ Math Surface**.



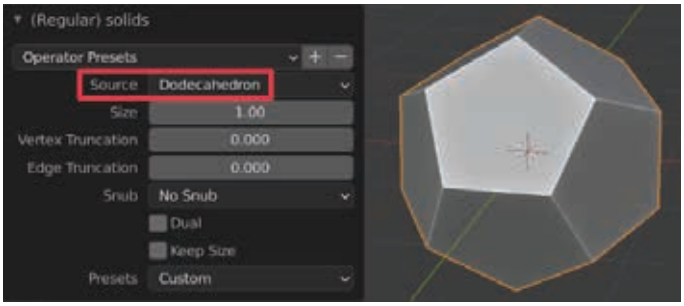
Expand the menu.



There are many different operator presets.

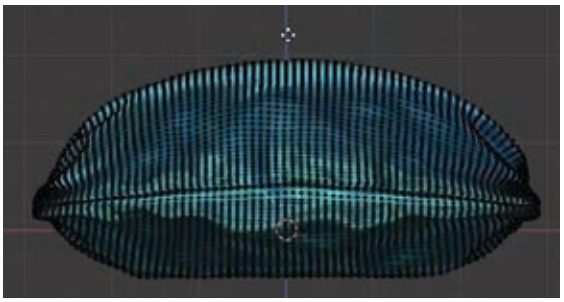


You can also add a **Regular Solid**, with such shapes as the **Dodecahedron**.



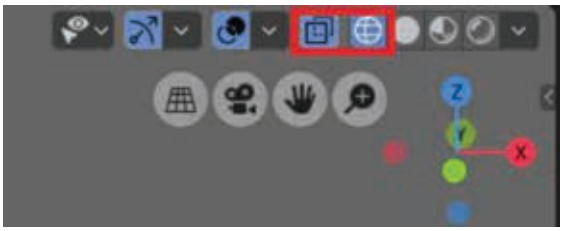
Hide vertices from editing in Edit Mode

With Proportional Editing in Edit Mode you may affect parts that you don't want to affect.

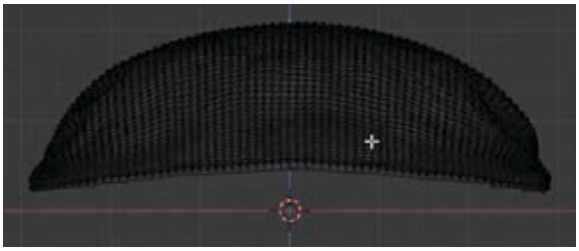


The simple solution is to just hide the vertices you don't want to be affected. In Edit Mode drag a selection around the part you want to hide, then **press H**.

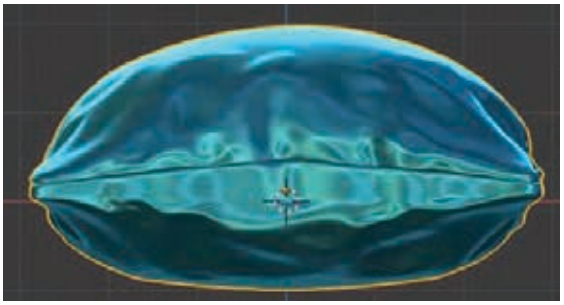
Make sure **Wireframe** mode and **X-ray** are turned on, so you select all the vertices, not just the ones in front.



Now only the visible vertices are affected by the editing.



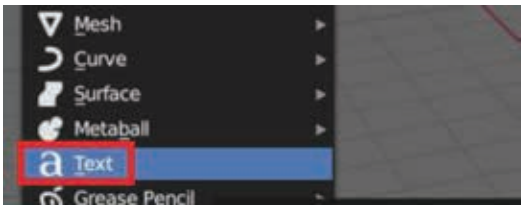
When we unhide by **pressing Alt+H** and go to Object Mode, we see that the bottom vertices have remained unaffected.



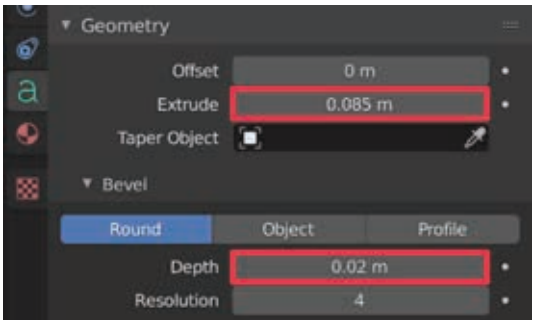
Balloon Text

In this tutorial we will look at making an "inflated balloon" style text.

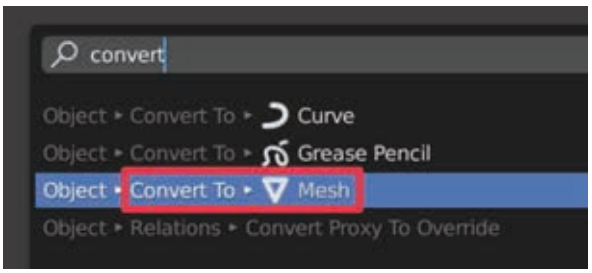
Press shift+A to add a text, hit **TAB** to edit the text.



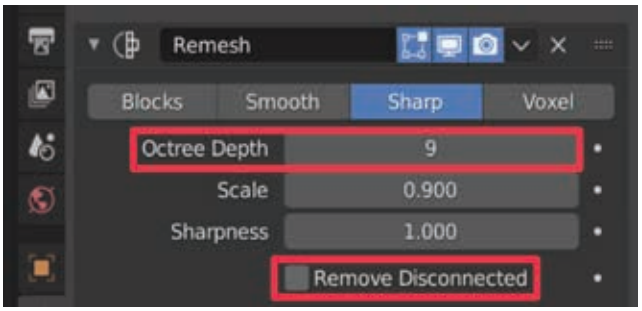
Extrude it in the Geometry options and give it some depth in the Text tab.



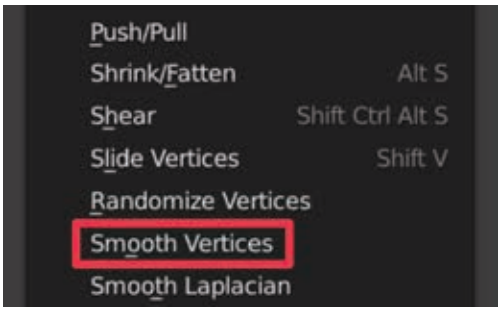
Press F3 and search for **Convert to Mesh**. This converts it from a Text object to a regular mesh.



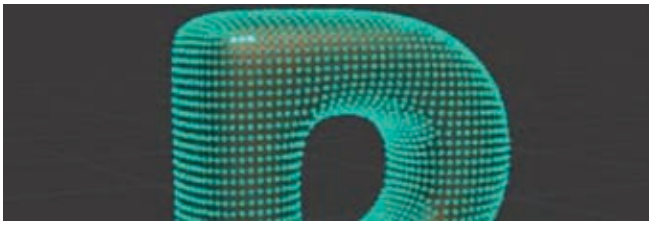
Add a **Remesh modifier** and apply it by **pressing Ctrl+A**.



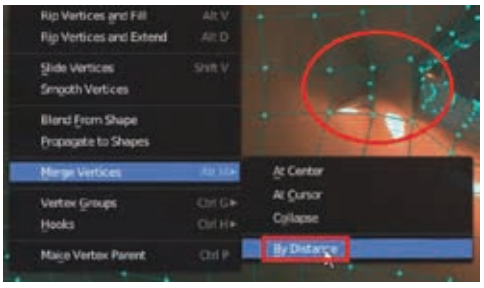
In Edit Mode select all, go to Vertex > **Smooth Vertices**.



Repeat Smooth Vertices by pressing **Shift+R** many times (say, 60 times) until it becomes smooth enough to your liking.

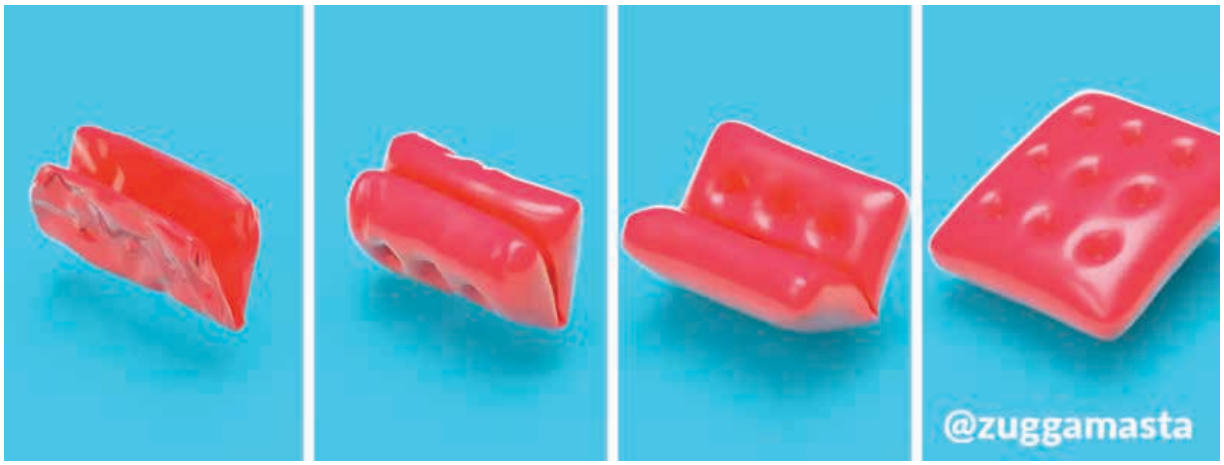


You may need to do some cleanup with Merge by Distance.

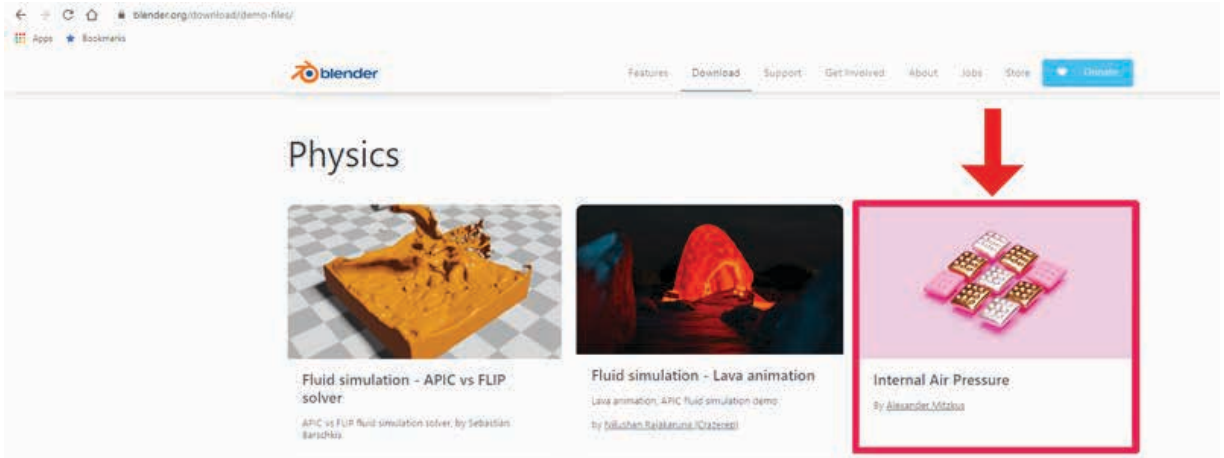


Make balloon objects with Cloth pressure

@zuggamasta on Twitter (Alexander Mitzkus) shared this animation that he made using Cloth pressure on the Blender website (scroll down to "Physics" to find it), allowing us a closer look at how he created it.

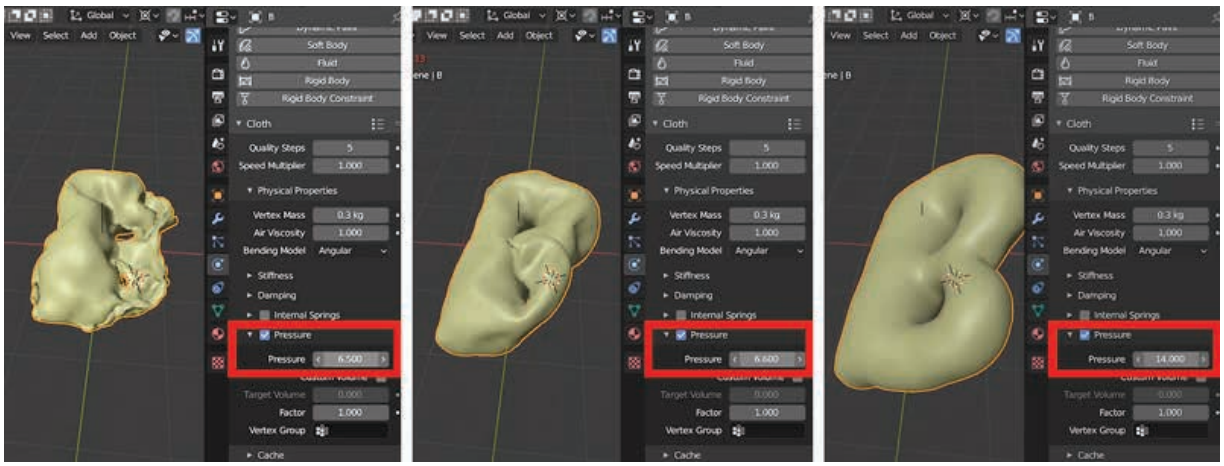


Fortunately, he shares the file on the Blender website. So we can analyse it and learn how to reproduce this effect.

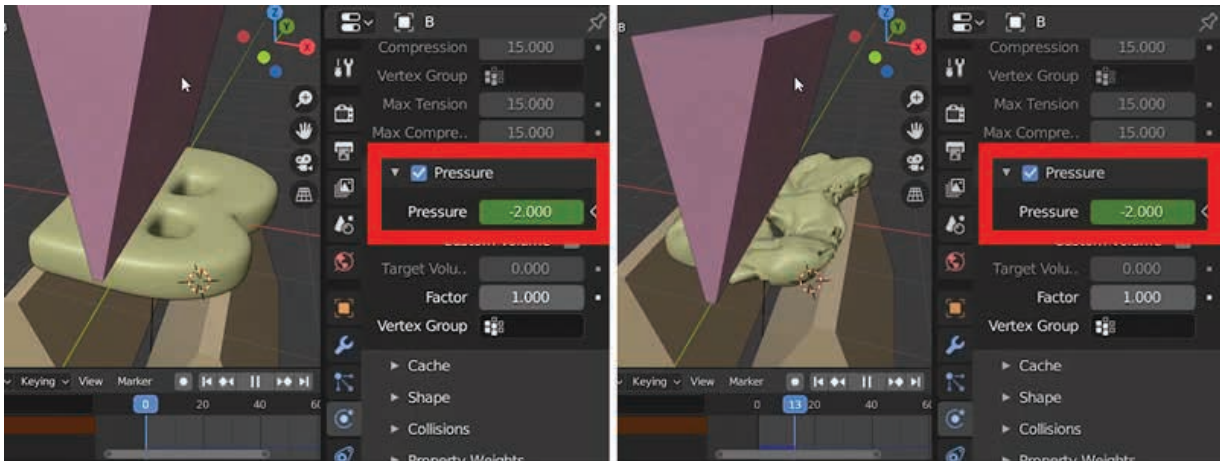


First of all, it uses the new Cloth physics, which include something called **Pressure**. Pressure is a keyframable value which deflates or inflates objects.

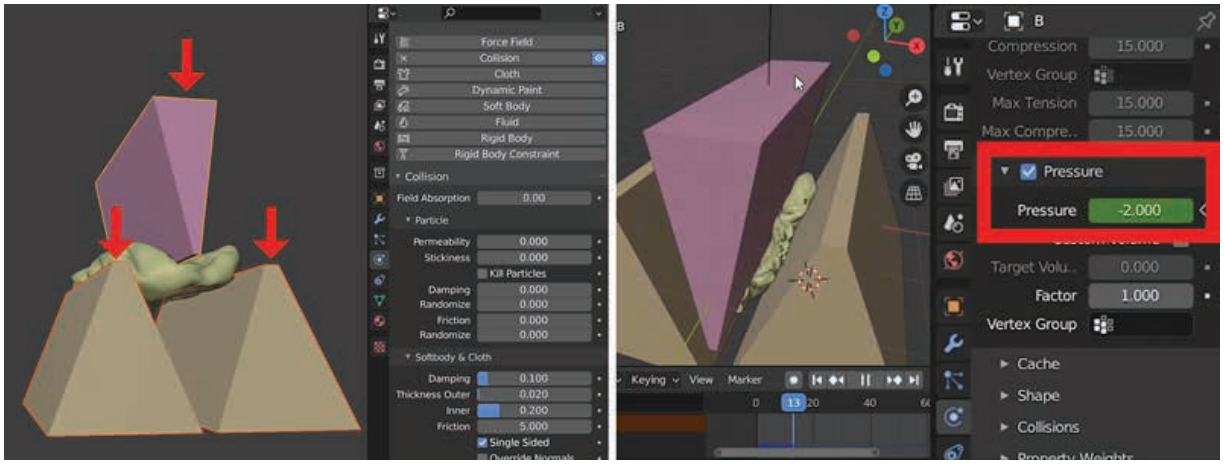
When we fill in positive value and **press the Spacebar**, the simulation starts and the model gets inflated and starts to float a bit.



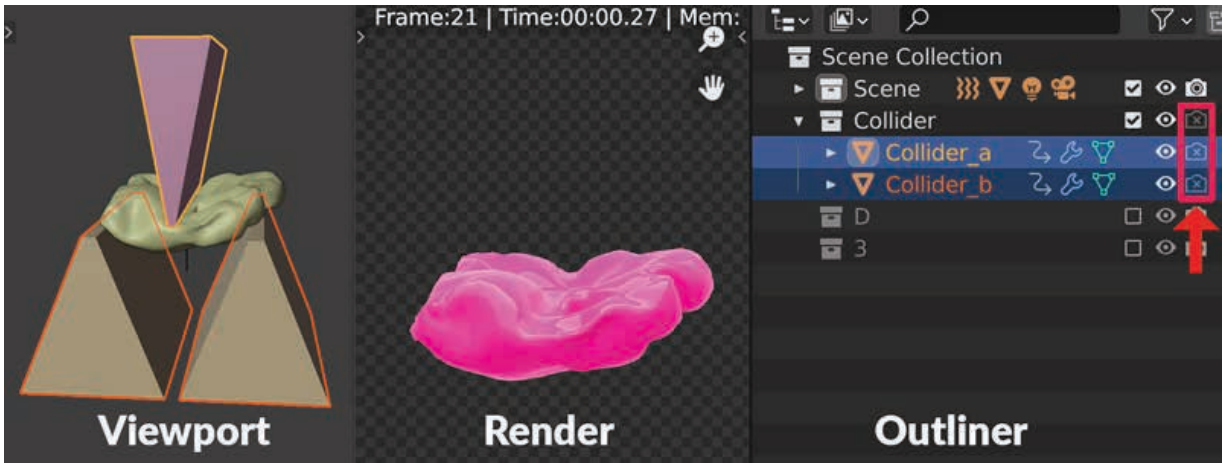
The amount of pressure you need depends on your object. In the beginning this object has pressure of -2, so it gets kind of sucked into itself.



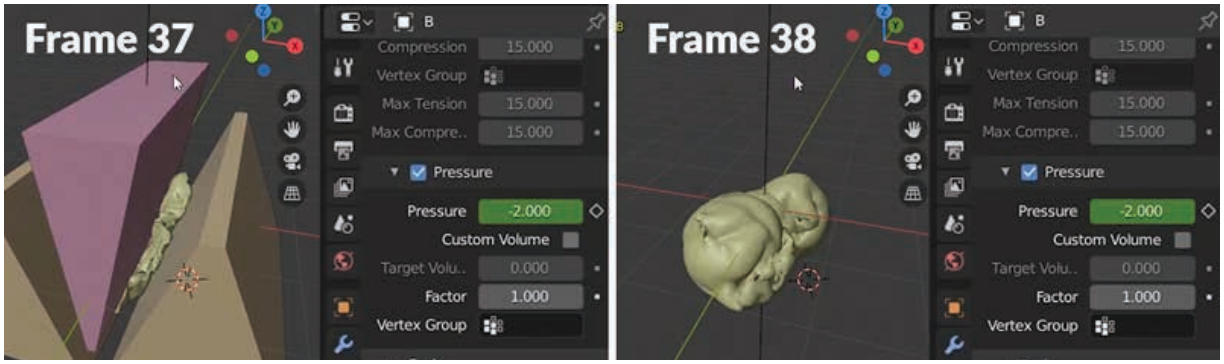
Then these collision objects squash it.



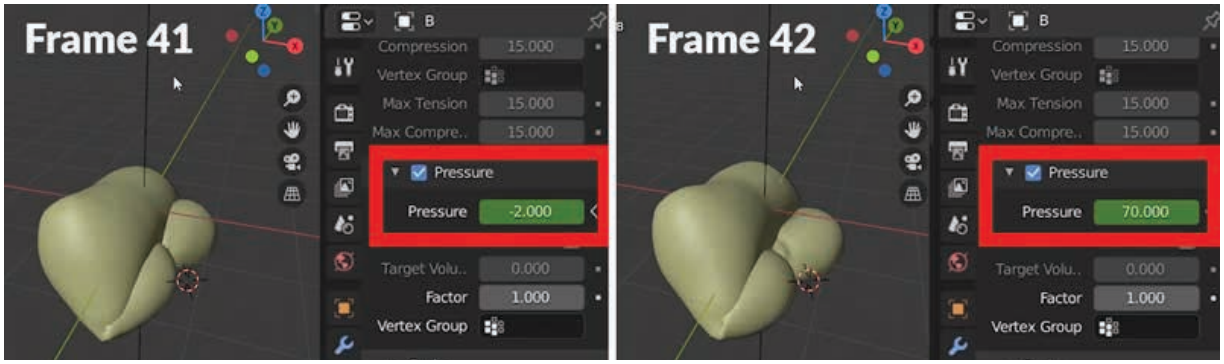
They are not visible in the render.



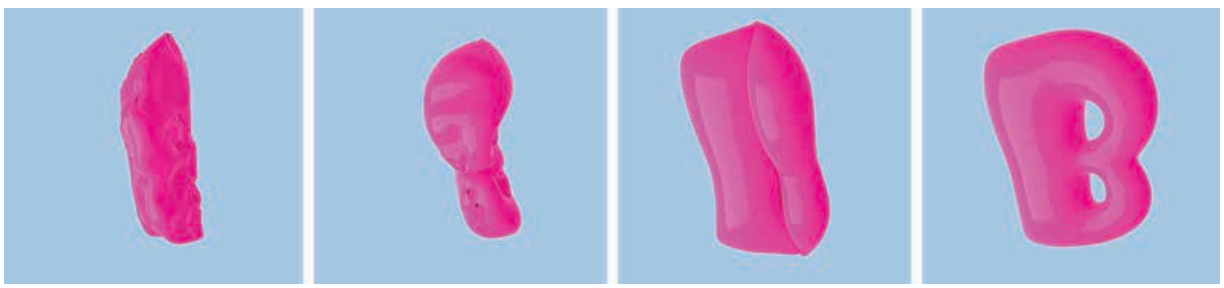
Next, they move away from one frame to the next.



At the same time, the pressure is increased from -2 to 70.

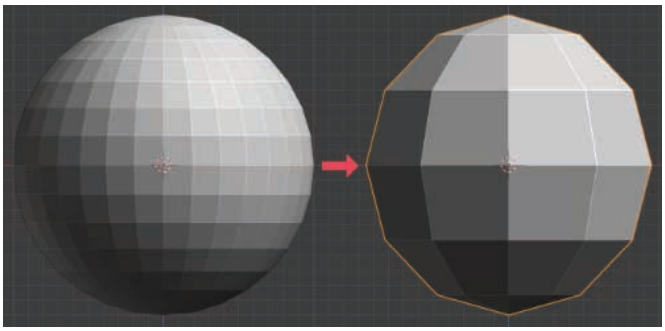


This causes the object to inflate and unsquash itself.

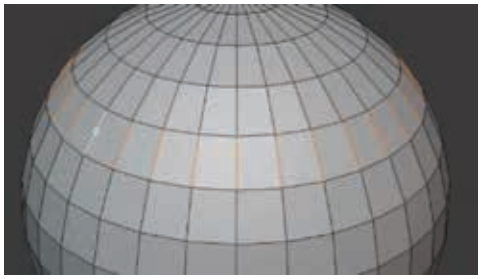


Two ways to unsubdivide

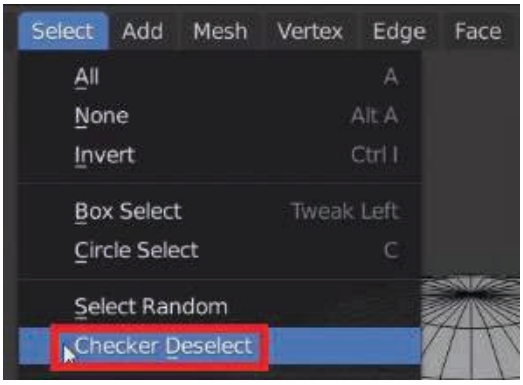
Here we look at two ways to reduce detail on your mesh. Manually, and with a modifier.



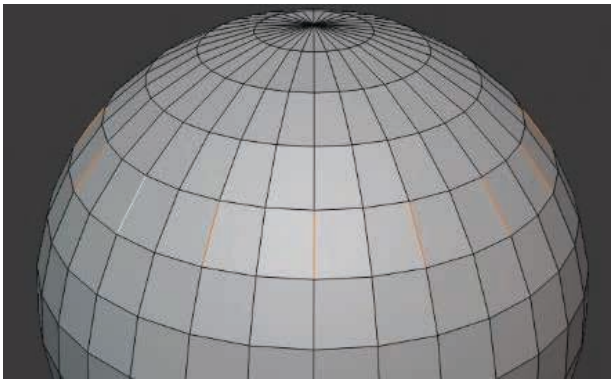
Hold Ctrl and Alt, then left-click on a vertical edge to select a loop of similar edges.



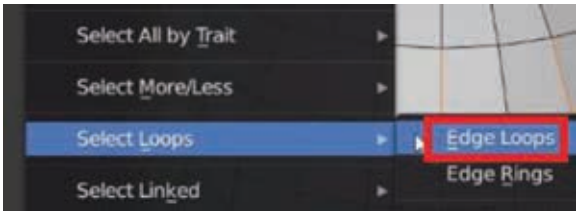
Select > **Checkered Deselect**.



Now every other edge is deselected.

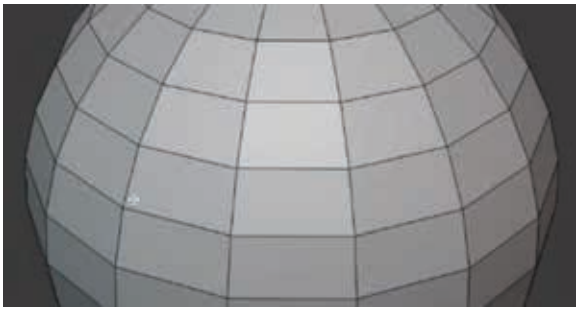


Select > Select Loops > **Edge Loops**.

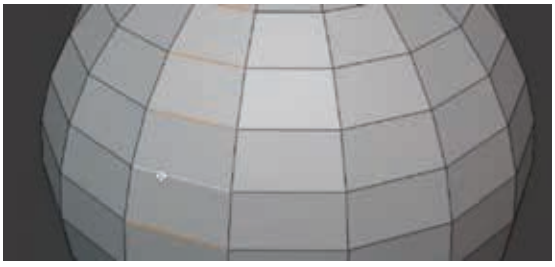


Now the selection is extended from those edges to their edge loops.

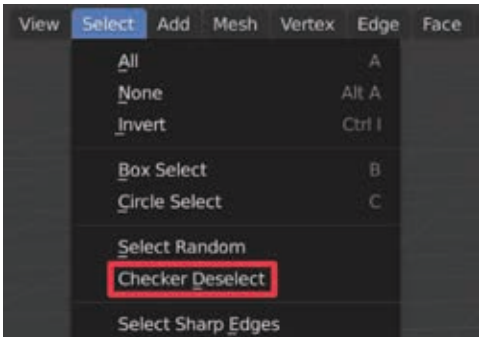
Press Ctrl+X to **dissolve** those edge loops.



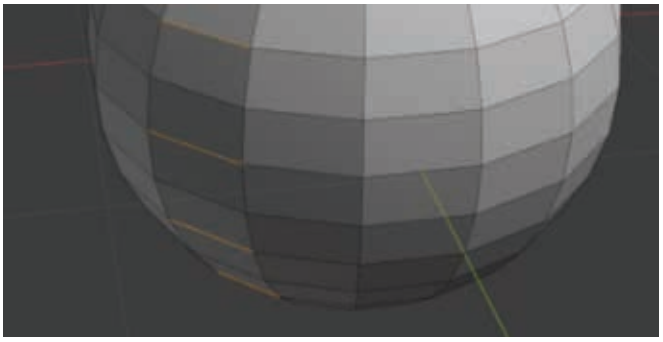
Hold Ctrl and Alt, then left-click on a horizontal edge to make a new selection of edges.



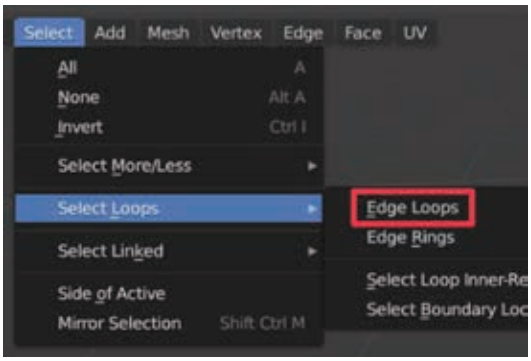
Select > **Checkered deselect**.



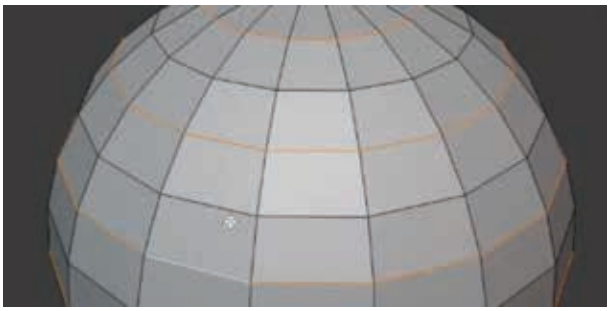
Now every other edge is deselected.



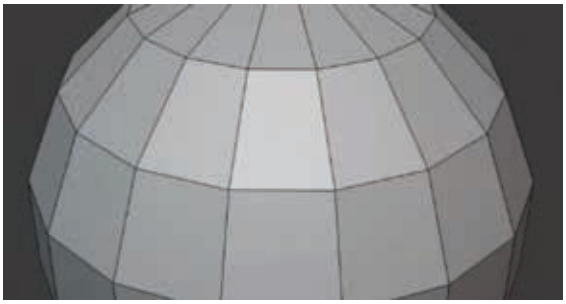
Select > Select Loops > **Edge Loops**.



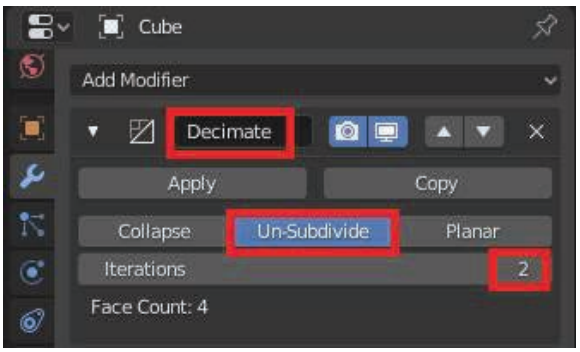
Now the selection is extended to include the entire edge loop of the selected edges.



Press Ctrl+X to dissolve the selection. This is one way to reduce detail on a mesh.

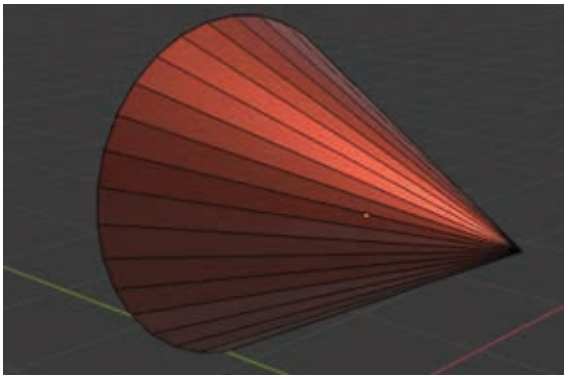


Alternatively, add a **Decimate modifier**. Set it to **Unsubdivide**, 2 iterations.

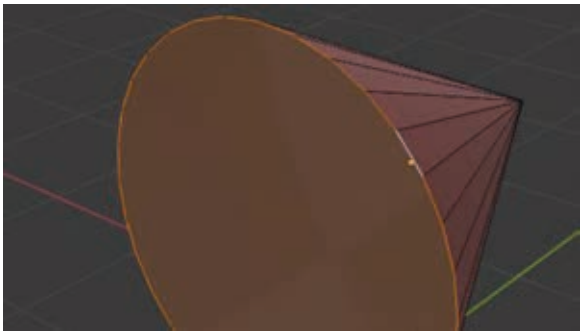


Add an Edge Loop on triangles

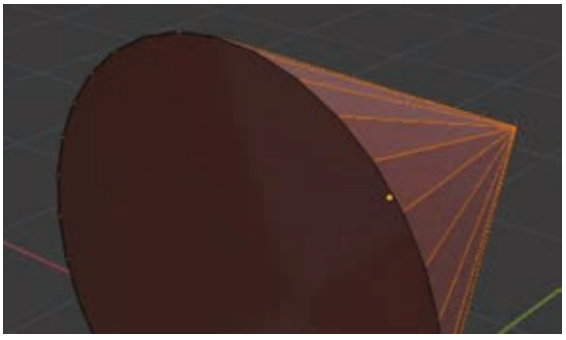
Normally you can add an extra edge loop with **Ctrl+R**, but that doesn't work in this case because these faces are triangles.



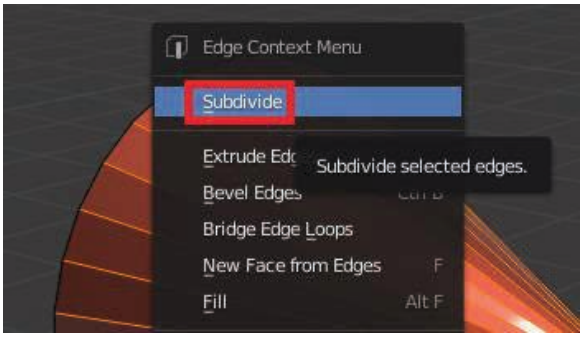
Here's how to do it. Select the bottom edge loop by **holding Alt and left-clicking** on it (or just double-clicking on it if Emulate 3-button mouse is enabled in Preferences).



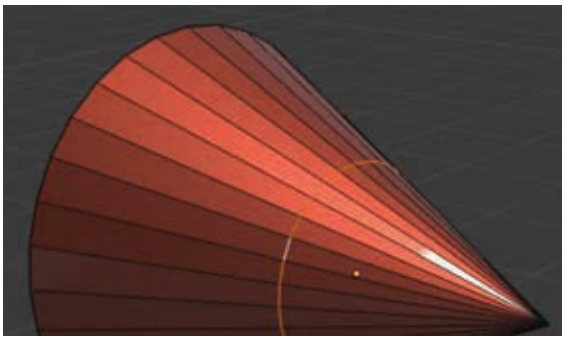
Invert the selection by **pressing Ctrl+i**.



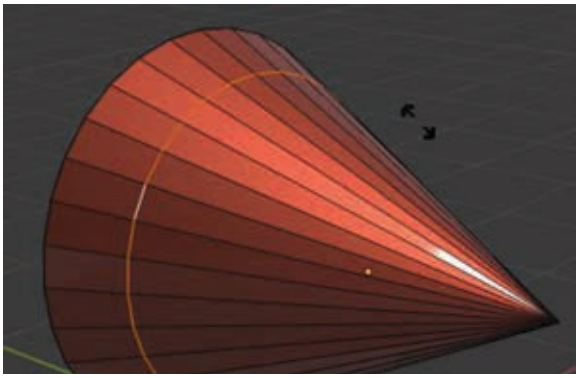
Right-click and choose **Subdivide**.



Now you have an extra edge loop. Select it by **holding Alt and left-clicking** on it.



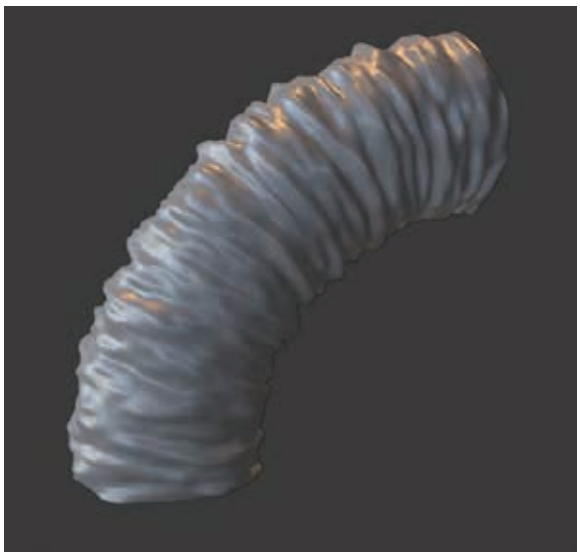
Then you can slide it by **pressing G** twice.



Click here to return to the index.

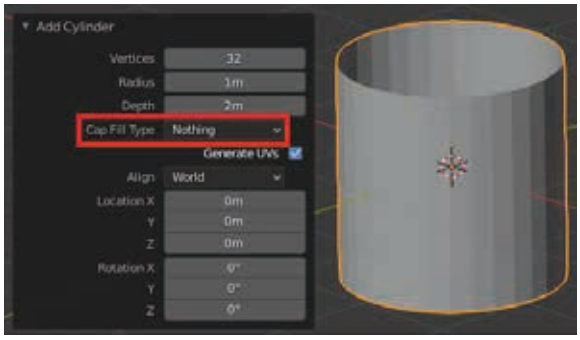
Make a pipe with cloth wrinkles

In this tutorial we'll look at making a pipe that bends at a 90° angle and has cloth-like wrinkles or folds.



First of all, we create a cylinder by **pressing Shift+A** and choosing Mesh > Cylinder.

As **Cap Fill Type** choose **Nothing**.



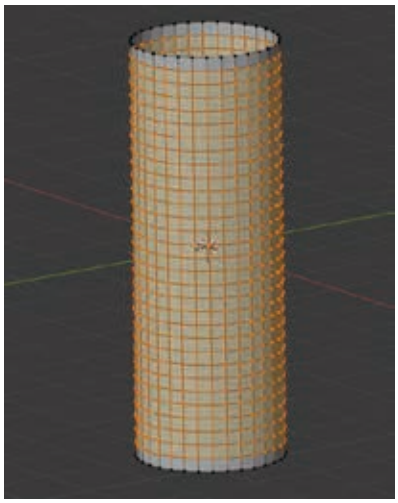
Scale it up along the Z-axis by **pressing S+Z**.

Then add some horizontal edge loops by **pressing Ctrl+R** and scrolling up on the mouse wheel.

Left-click to confirm, then right-click to make sure the selection doesn't move.

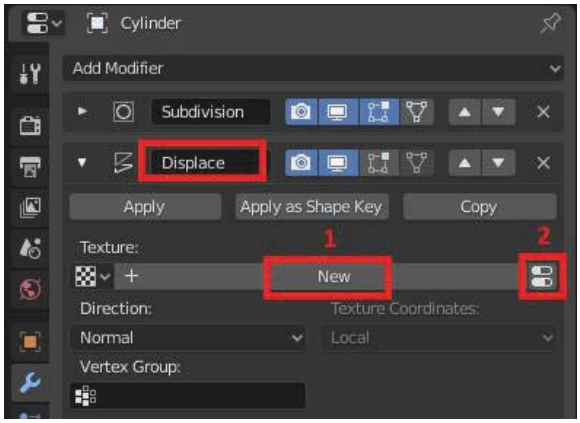
Press TAB to go to Object Mode.

Press Ctrl+2 to add a Subdivision surface modifier with two levels of subdivision.



Add a **Displacement modifier**.

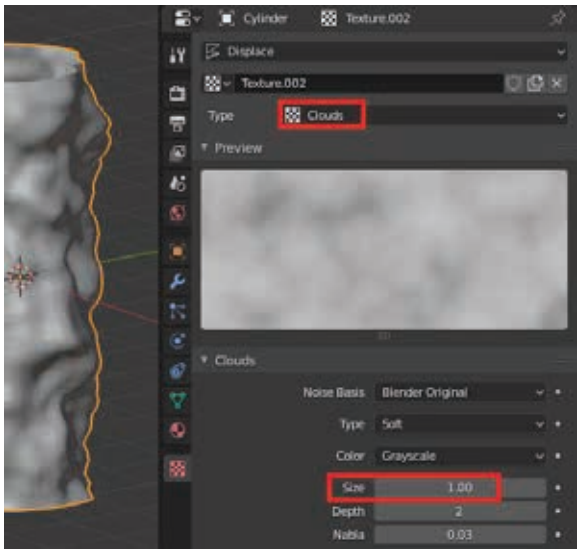
Make sure it is below the Subdivision surface modifier, so that it uses the extra geometry created by the Subdivision modifier.



Click on New and then on the Texture tab icon.

As **Type** set the texture to **Clouds**.

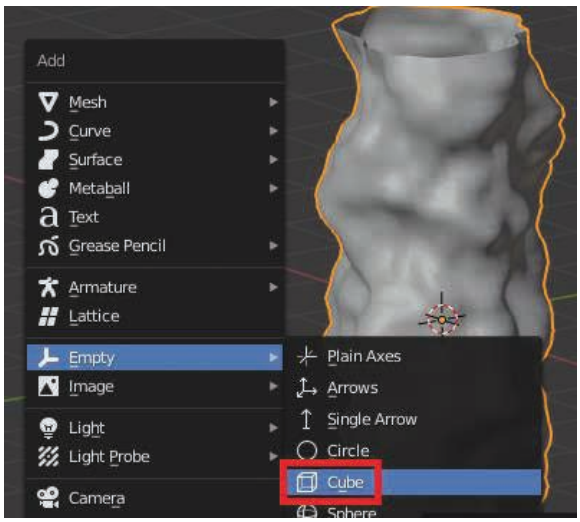
In this case I set the size to 1, otherwise the result was too spiky.



Right-click on the cylinder and choose **Shade Smooth** to get a better visual representation of what the Displacement is doing.

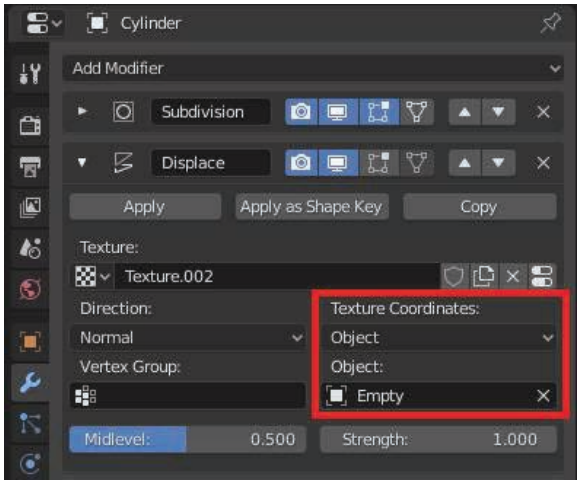
Press Shift+A and choose Empty > **Cube** to add a Cube-shaped Empty.

Make sure the 3D cursor is still in the middle of the scene so that the cursor is in the middle of the Cylinder when you create it.

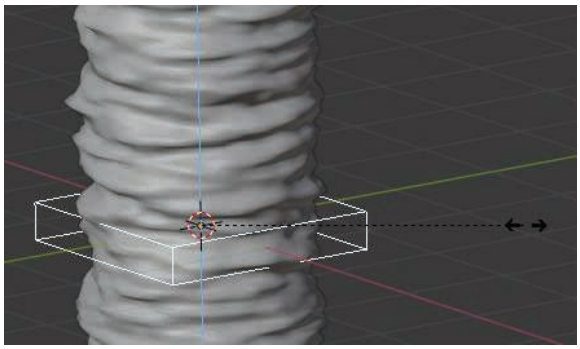


Select the Cylinder and in the Displacement modifier, set Texture Coordinates to **Object**.

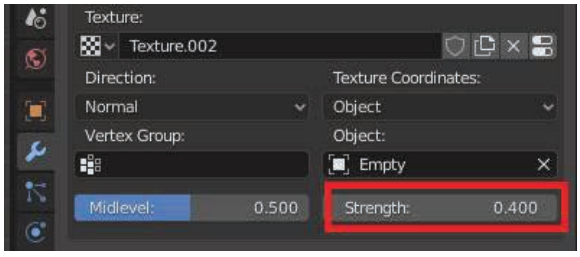
Choose the **Empty** as the Texture Coordinates Object.



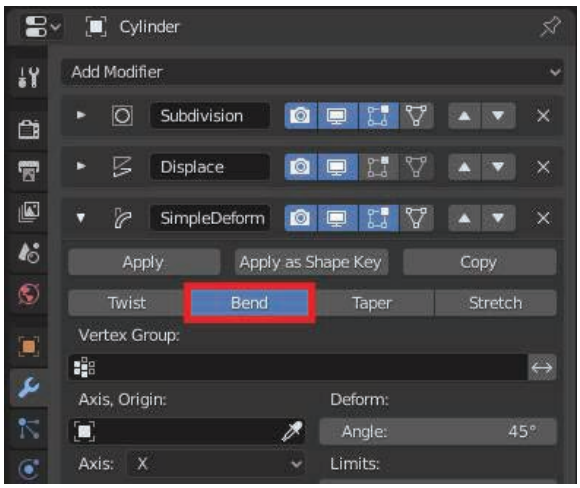
Select the Empty and scale it down along the Z-axis by **pressing S+Z** and moving the mouse.



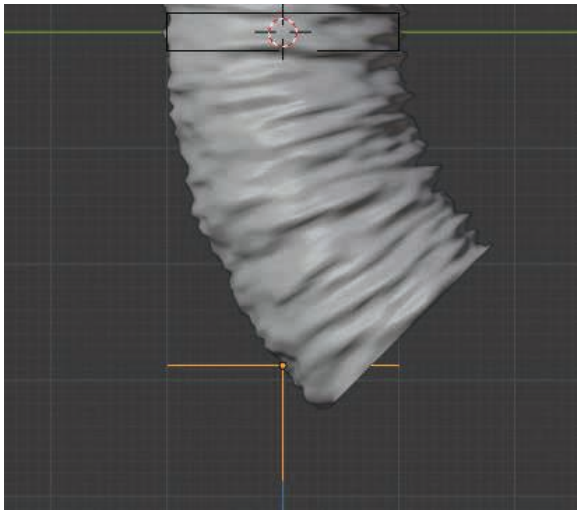
In this case I noticed I had to reduce the **Strength** of the Displacement modifier to get a better looking result.



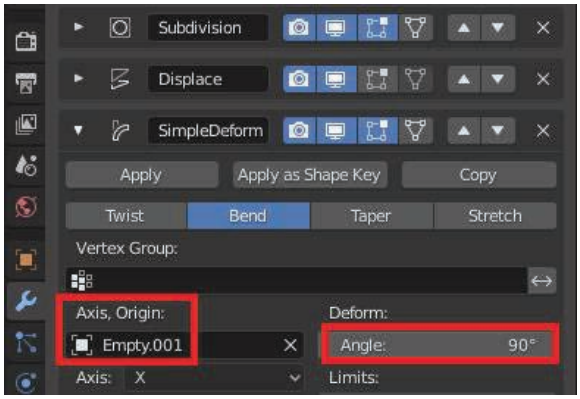
Add a **Simple Deform modifier**, set it to **Bend**.



Press Shift+A and add another **Empty** (any shape is fine) and move it down to the bottom of the Cylinder.



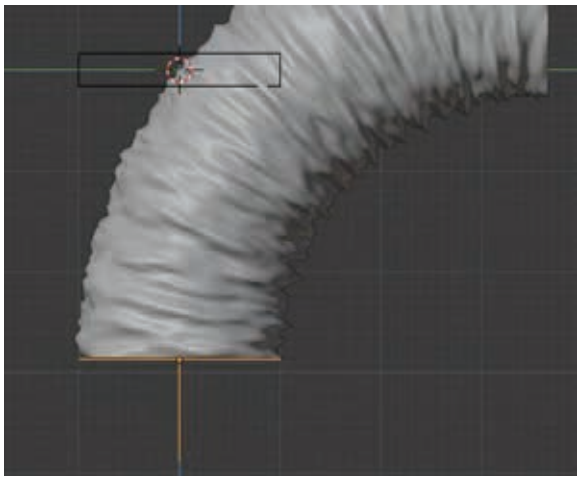
Use the new **Empty** as origin for the Simple Deform modifier.



You may still need to fine-tune the position of the Empty at the base of the Cylinder to get it to line up exactly with the base.

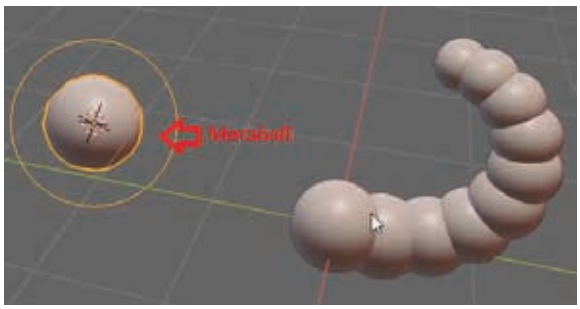
At this stage you can fine-tune the look by tweaking the values of the Displacement modifier and Cloud texture.

If you want to you can Apply all the modifiers so that you can for example sculpt on the mesh.

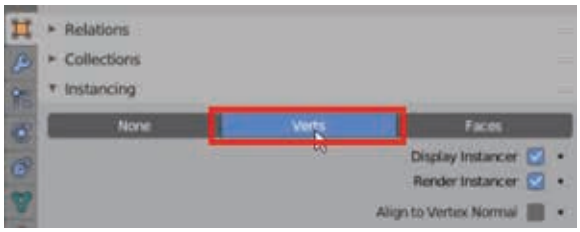


Metaball Array

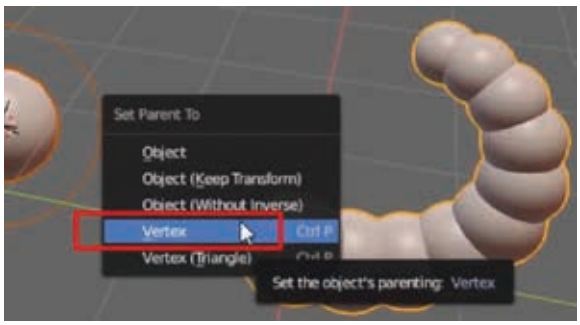
Press Shift+A > Metaball > **Ball** to add a Metaball to your scene (see the previous tip on how to make the array of spheres).



Turn on Instancing: **Verts** for the normal sphere object.



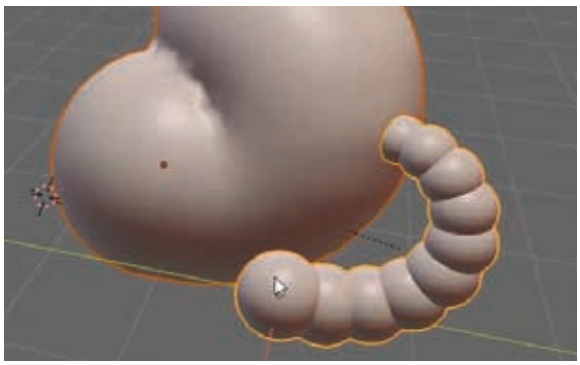
Shift-select first the Metaball, then the normal Sphere and **press Ctrl+P** and choose **Vertex**.



Now select the Empty that drives the Object Offset of the array (see previous tip).

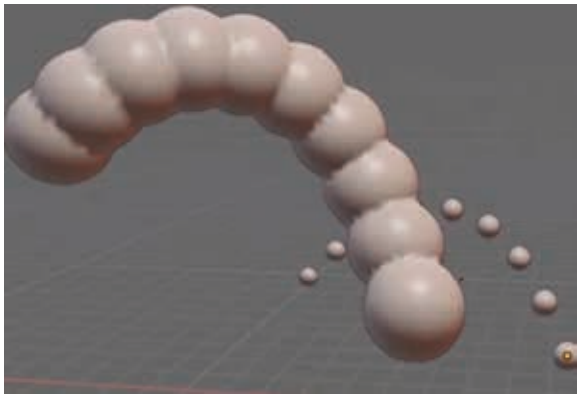
It now controls the Metaball array as well.

We need to scale the empty down otherwise the Metaballs are too big and create this accidental butt-shape.

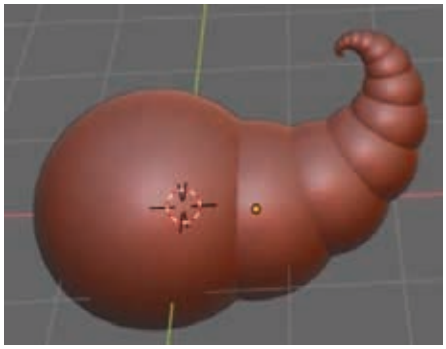


As you can see the array controls the Metaballs as well.

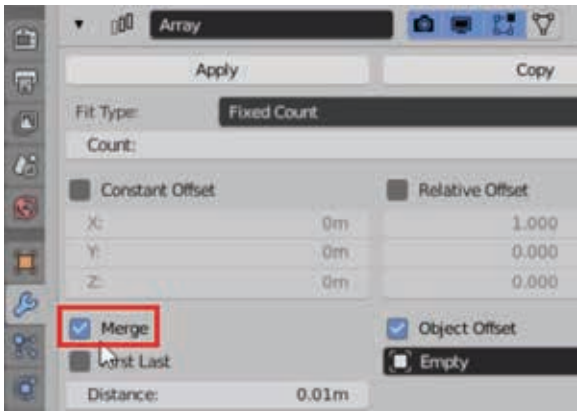
The benefit of the Metaballs is that they are automatically welded together.



An alternative workflow is to just use the **Merge** setting in the Array modifier.

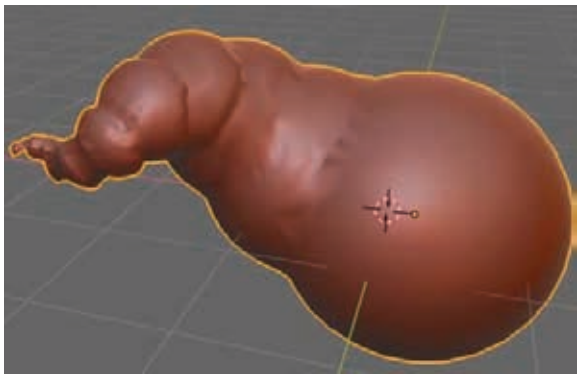


Turn on **Merge** and experiment with the value until the Spheres merge.



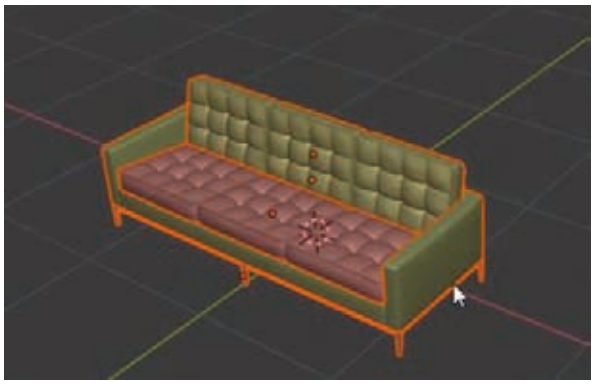
The result is different than with the Metaballs solution. Both could be useful in different occasions.

Of course they create messy topology, so you may want to remesh them, for example with a Remesh modifier.



Group instances

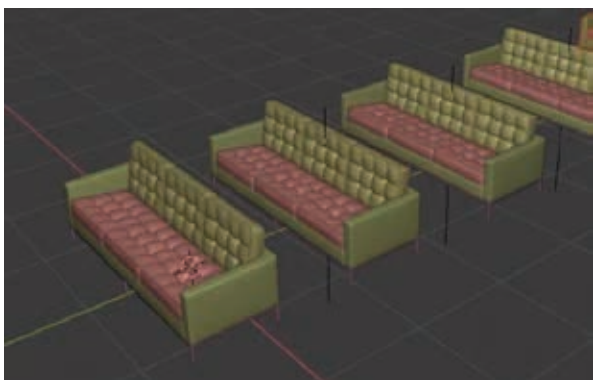
When I duplicate this object using **Shift+D**, the face count and memory use increase quickly.

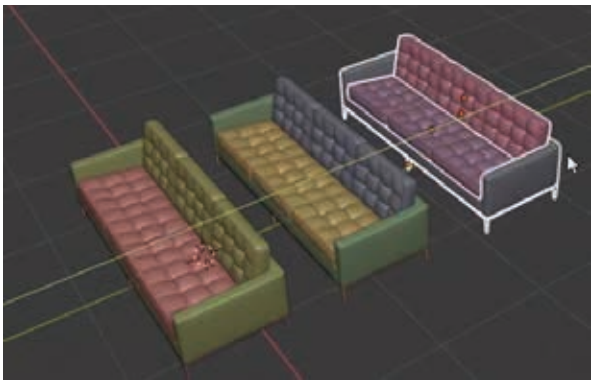


Instead add the Object or objects to a Collection by **pressing M** and giving it a name, like “Sofa”.

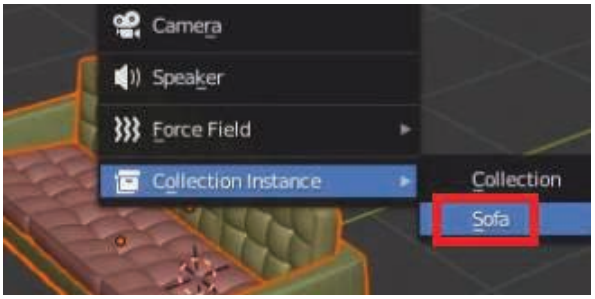
These instances can now be duplicated with **Shift+D**, but notice there's no increase in Geometry.

It’s the same as **pressing Alt+D** to make a Linked Duplicate, except you’ve added it to the Create menu as a Collection Instance.



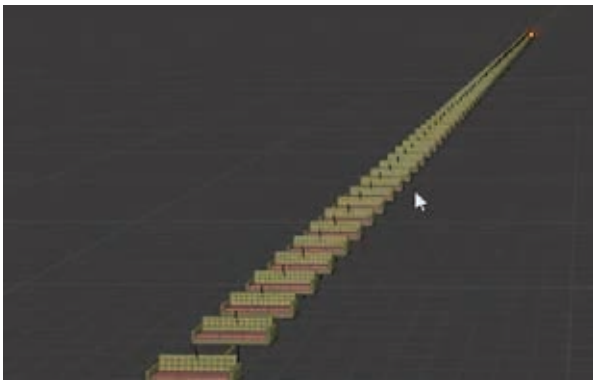


Now **press Shift+A** and choose **Collection Instance** and choose the group you named Sofa.



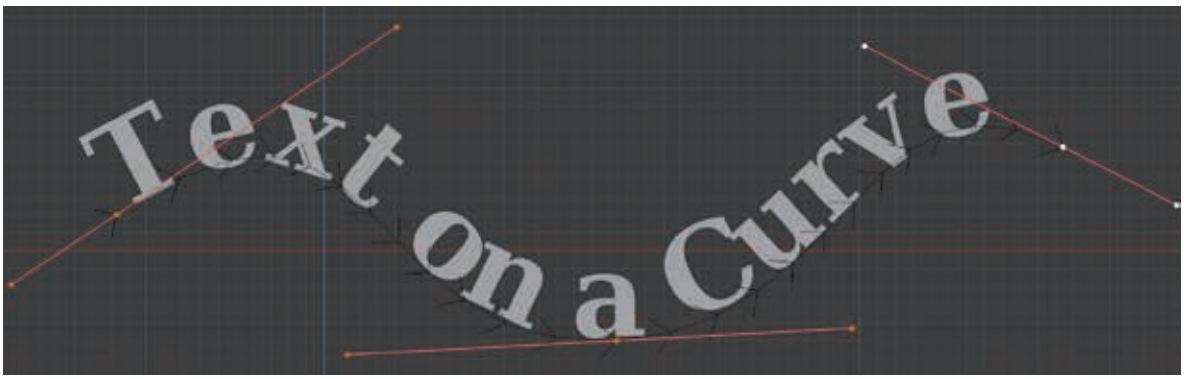
I can keep adding sofas with this method, without an increase in memory usage.

Infinite sofas!



Text on a Curve

Did you know you can place text on a Curve in Blender? This is a nice way to create text for motion graphics, for example.



To add text to a curve object in Blender, first create both a BezierCurve and a Text object. To create a BezierCurve **press Shift+A** and choose Curve > **Bezier** (or Circle, if you want circular text). Then **press Shift+A** again and choose **Text**.

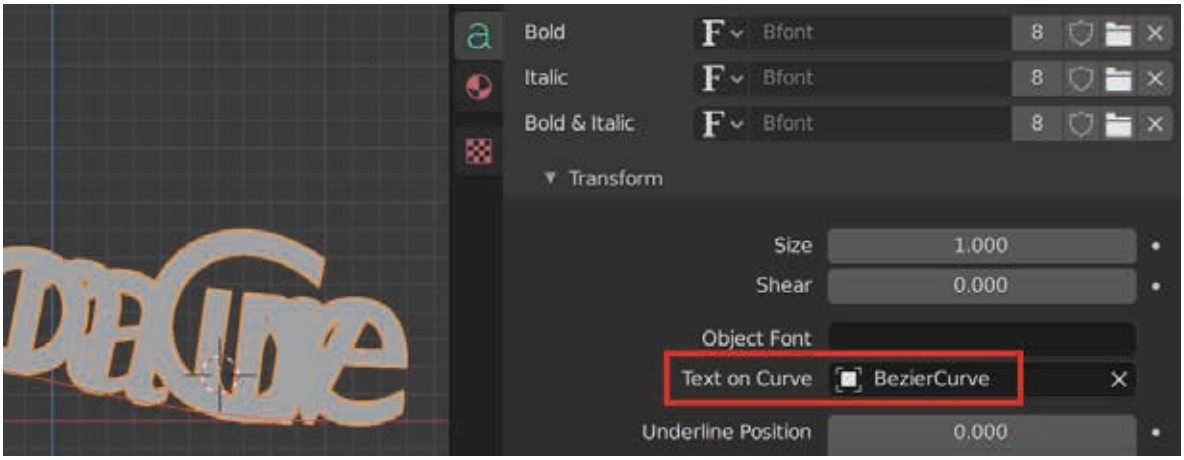
By default Blender creates these things lying down flat, so you may want to select both and **press R+X+90** to rotate them both 90 degrees on the X-axis.

To edit the Text object **press TAB** (to go into Edit Mode) and type your text. **Press Backspace** to remove characters or **Ctrl+Backspace** to remove entire words.

To change the spacing between two characters, press the left or right keys until the cursor is between the two characters, then **hold ALT** and press the left/right keys. This is called the **kerning** in typography.

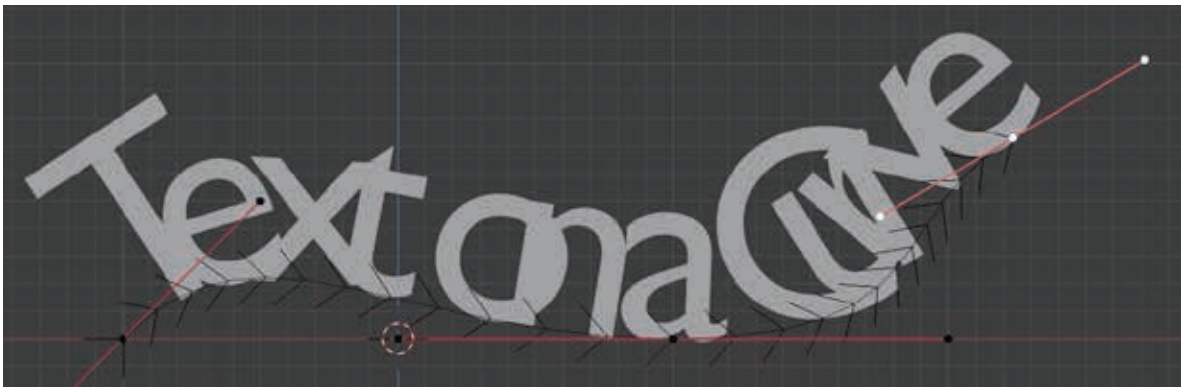
I guess this is personal preference, but I always feel I can improve the kerning. Below left: the text without edited kerning (notice the difference in distance between the R and V, and the V and the E). On the right the same text, with edited

Select the Text object and then under Font > **Transform** choose the **BezierCurve**.



You will notice that with the default BezierCurve, the text is now all bunched up to fit on the small Curve. Oh no, our precious kerning! But don't worry, we just have to make the Curve longer and edit its shape to our liking.

First, let's add another vertex to the BezierCurve. Select it and **press TAB** to go into Edit Mode. Select the vertex on the right and then **click somewhere to its right while holding Ctrl**. This extrudes another vertex to that location.

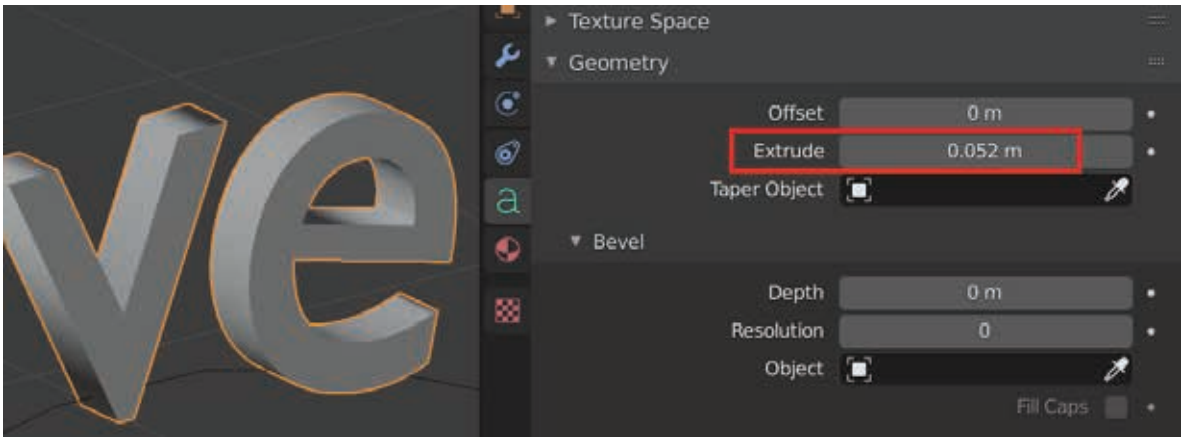


Alternatively you can **press E** to extrude the selected vertex. Whatever's most convenient to you.

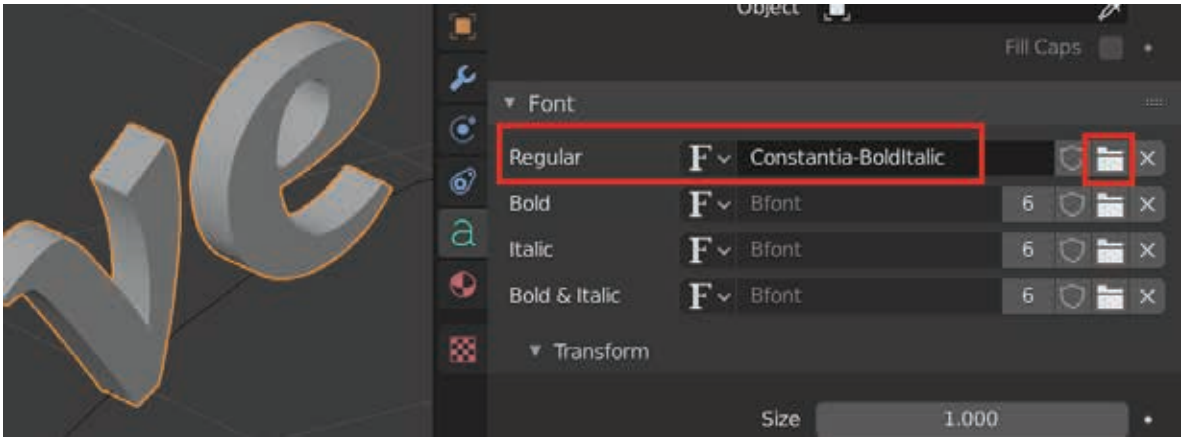
Now that we have enough vertices let's edit the BezierCurve so the text is readable again. Just like with a regular mesh, you can select vertices and **press G** to move them and **press R** to rotate them.

You can also **press S** to scale them - what this does is it changes the "weight" of the vertices. This changes the roundness of the curve based on how long the handles of that vertex are. It's best to try it in order to really understand what that means.

To give thickness to the text, you can enter an Extrude value.

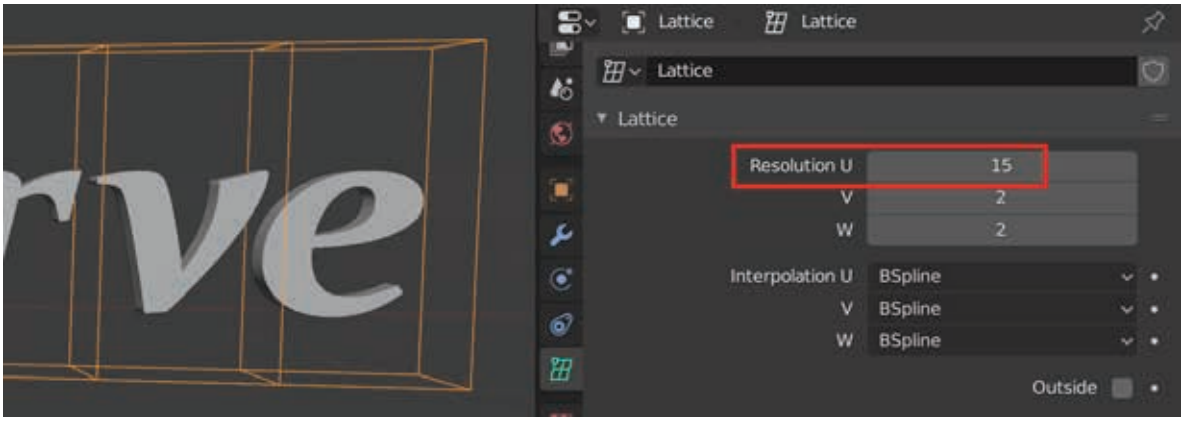


You can also still change the font of your text by clicking on the folder icon next to the font, and selecting a font there. You can visually see previews of the fonts so it's easy to choose.

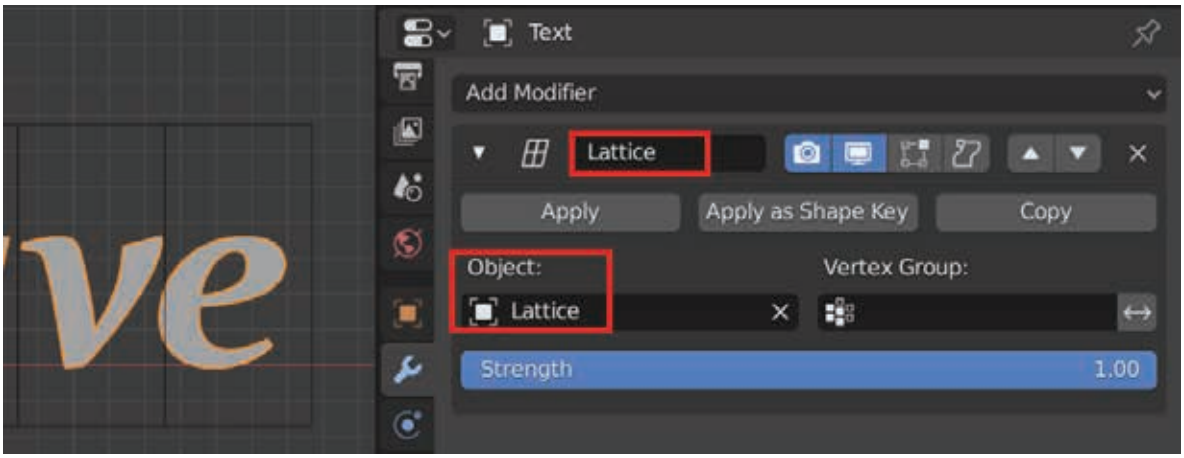


Alternatively, you could also deform the text using a Lattice. To do this, **press Shift+A** and choose **Lattice**. Scale and move the lattice until it fits snugly around the text. Be careful - do this in Object Mode, not in Edit Mode.

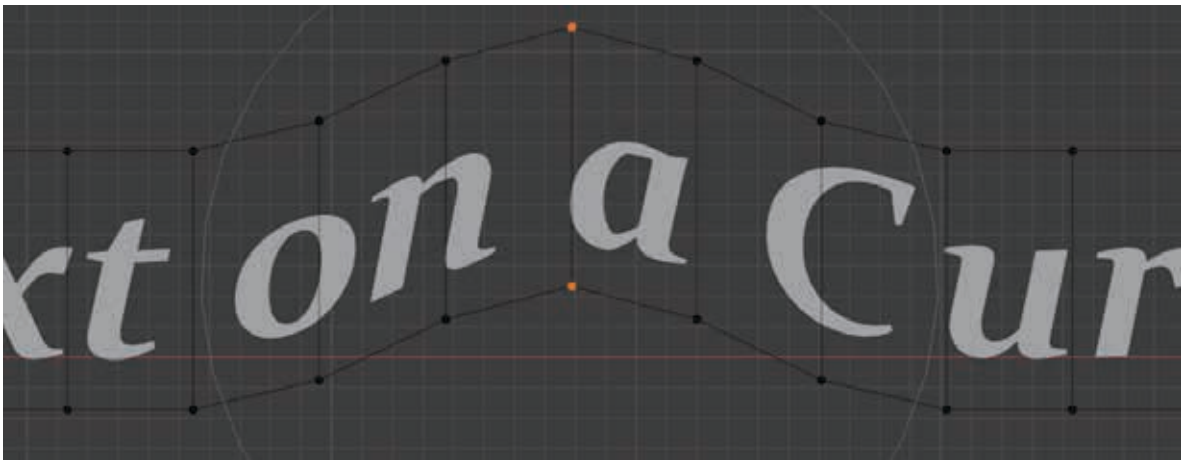
You will need to add some resolution to the Lattice in order to be able to do deformations. Increase the **U resolution**.



Select the Text again and add a **Lattice modifier**. As the object choose the **Lattice**.

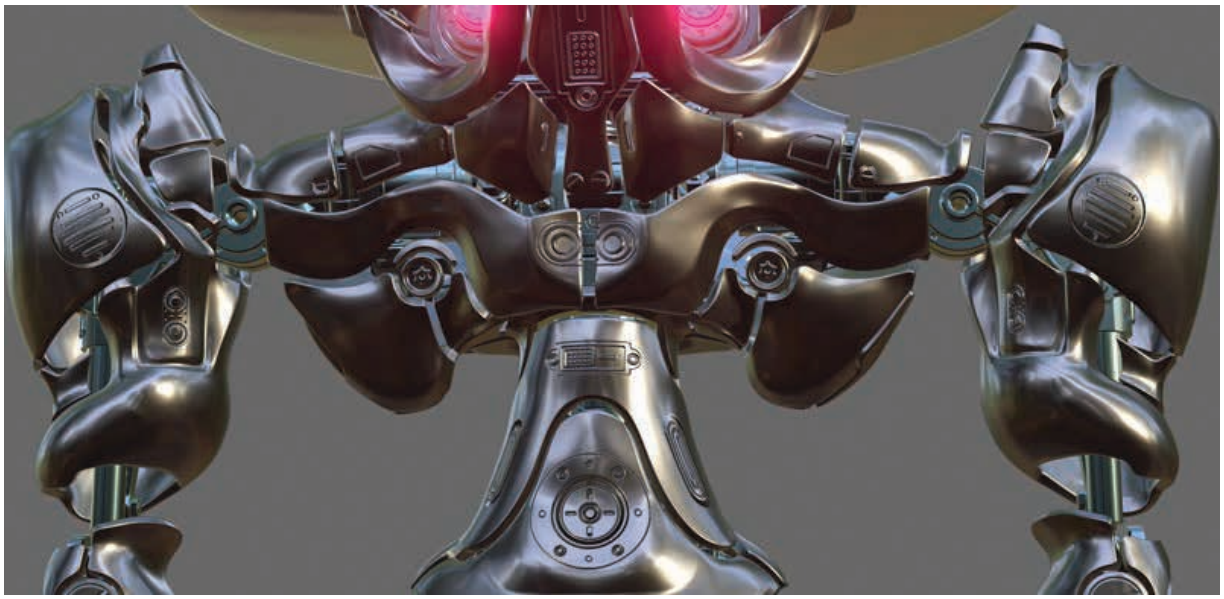


Now you can select the Lattice and **press TAB** to go into Edit Mode. If you select and move vertices, the text will be deformed along with it. It works best to activate **Proportional Editing** here by **pressing o and scrolling the mouse wheel** to increase or decrease the proportional falloff.



Modeling with Hard Surface Normal Stamps

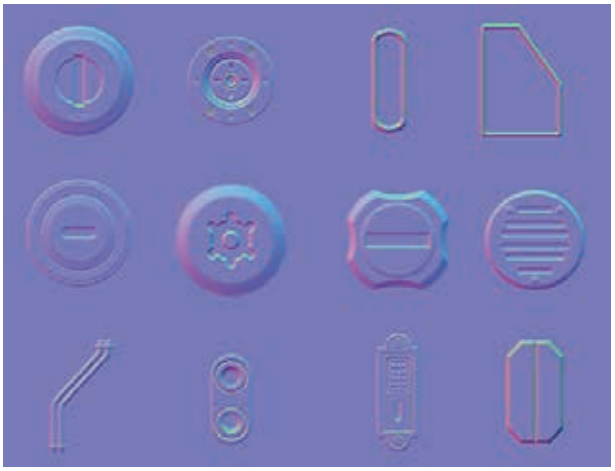
To add bolts, small panels and other small details we can use Normal stamps.



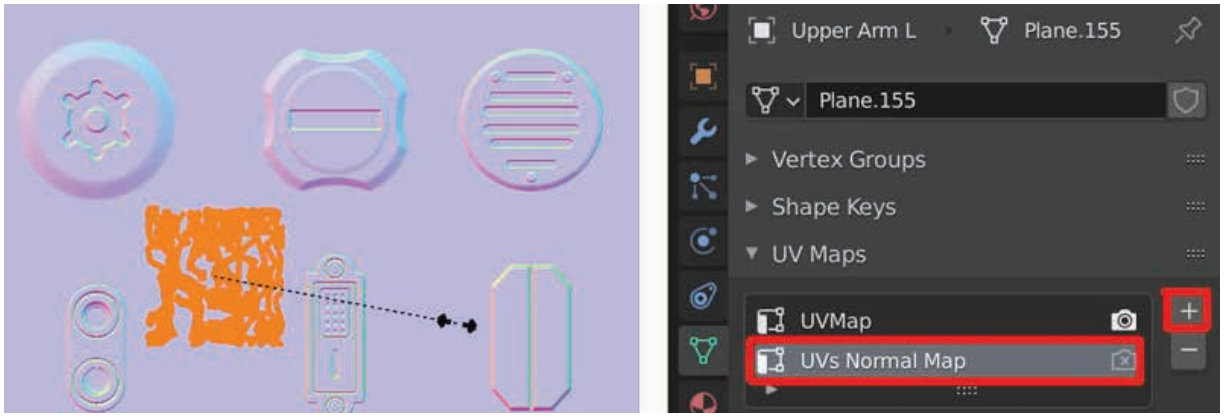
You can make these yourself by baking normal maps or you can save a lot of time buying them online, they're pretty cheap. I used these by Johathan Ching on Artstation.

Black and white alphas are more commonly found than normal stamps, and are used as sculpt brushes. You can use these too, by converting them to normal maps online on this website.

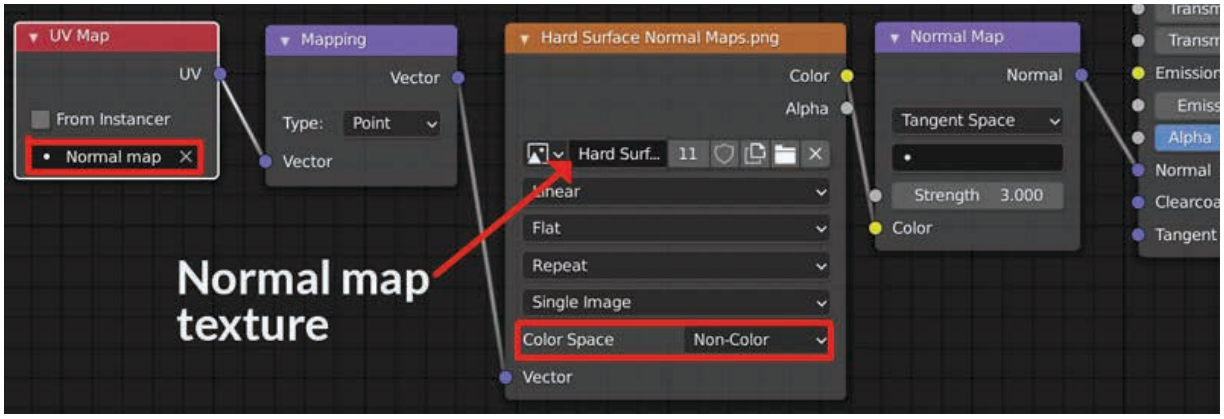
Contrary to the black and white sculpt alphas, the most efficient way to use Normal stamps is to place all of them onto one texture atlas.



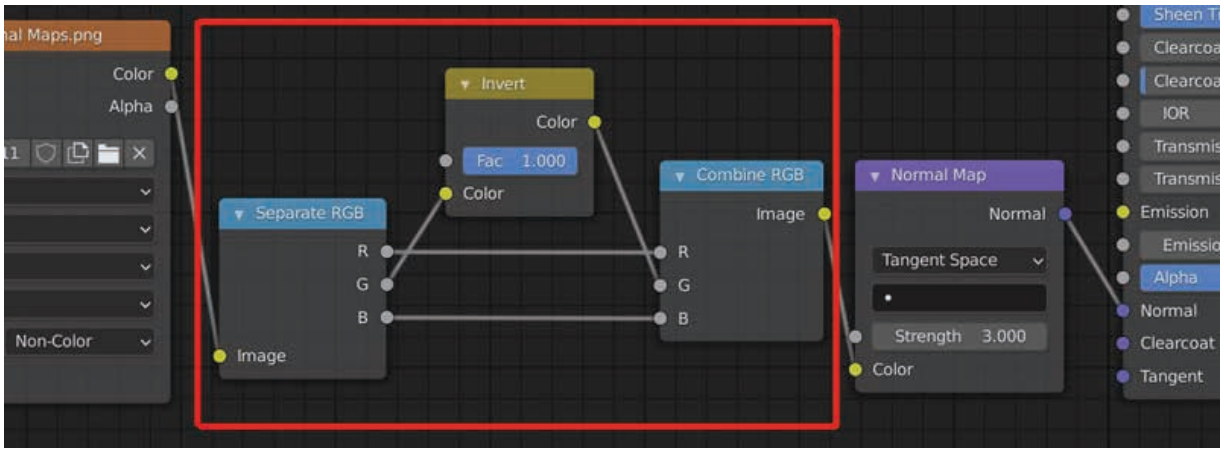
Load the Normal map texture in the UV Editor window so you can see what you're doing. **Create a second UV map** for the Normal map by c**licking on the + icon and giving it a name**. Now in Edit Mode scale all the UV's of this new UV map down to a point on the normal map where there's nothing.



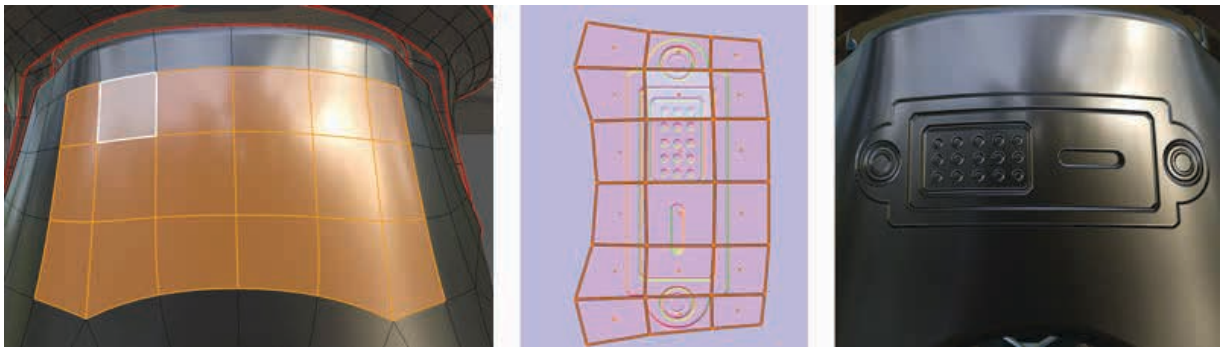
To use the normal maps is very easy. First, **add a Normal Map node** to the material of the object and connect the **Normal map texture** to it. Make sure the Color Space is set to **Non-Color**! Add a **UV Map node** to tell the material that it needs that specific UV map for the Normal map.



Just keep in mind that that website creates **Direct X** normal maps, which are slightly different than **Open GL** normal maps (which is what Blender uses). Also the chance that normal maps you buy online are Direct X instead of Open GL is pretty high. To "convert" them, simply add these nodes:



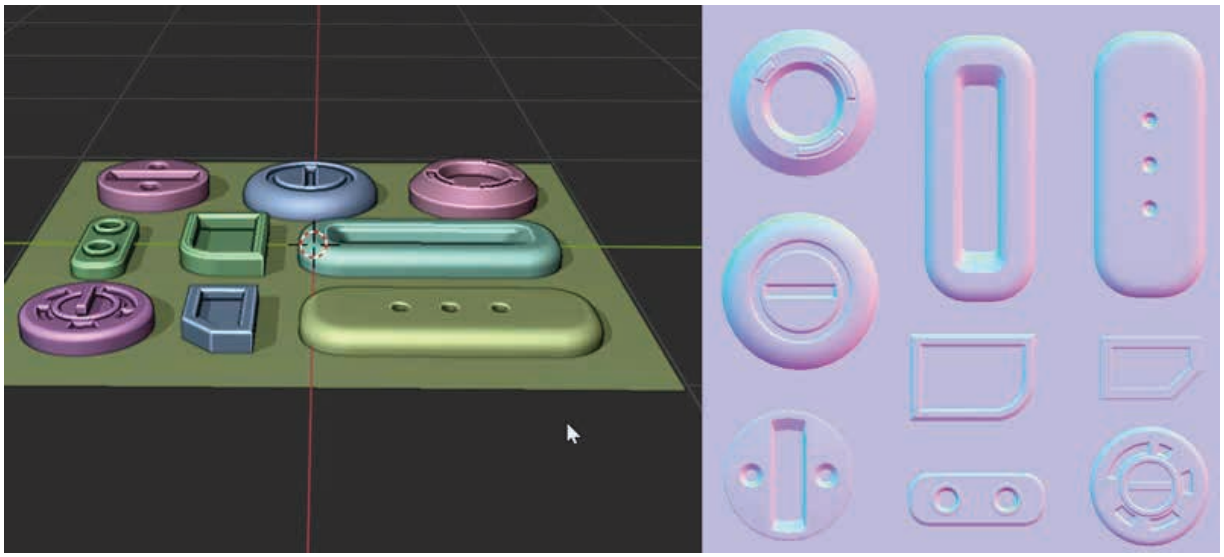
They will invert the green channel, which is what you need to display them correctly in Blender. If you've made the normal maps yourself in Blender, you don't need to do this. Now select some faces where you want to add a normal map and unwrap it, then move it to that part of the Normal map on the UV map.



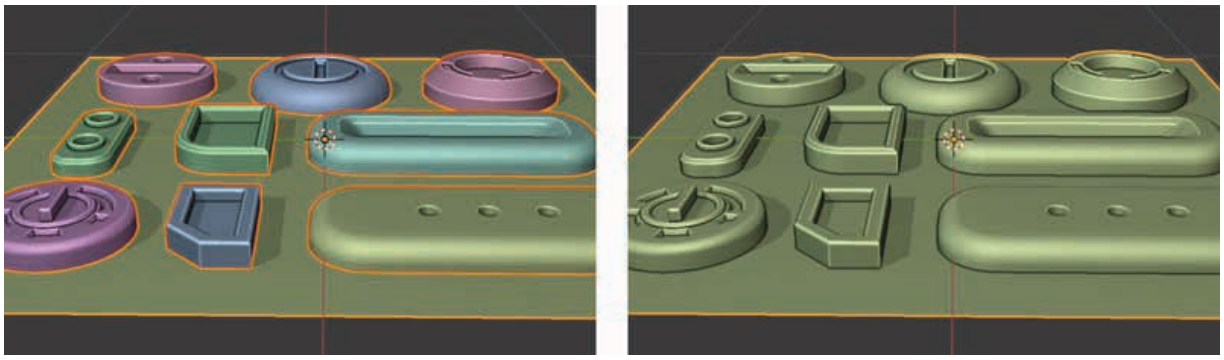
In other words, you only need to unwrap the faces where you want to add some details with Normals. Repeat this until you feel that you've got enough details. Play with the Strength of the Normal Map node until it looks right to you.

Create your own Hard Surface Normal Stamps

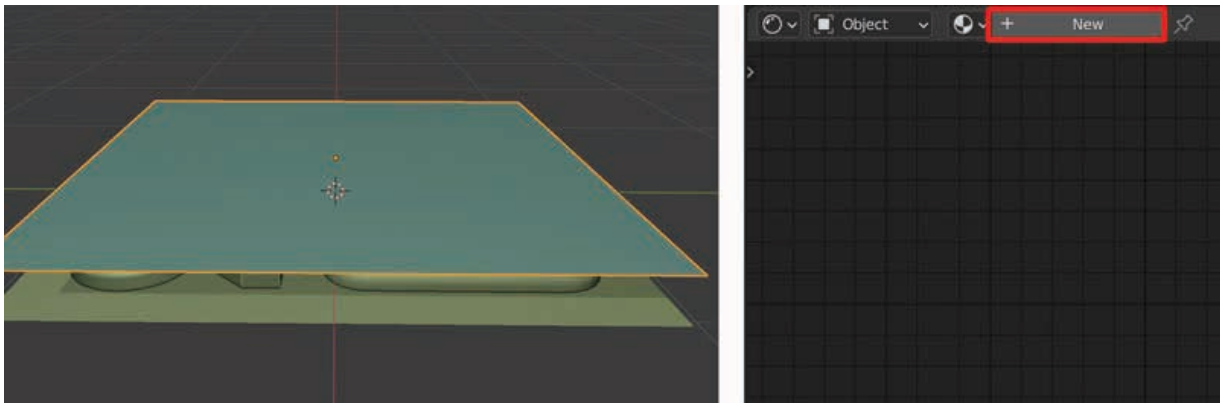
Normal maps can be used to add fake geometry as demonstrated in the previous topic. Now let's look at how to bake these kinds of normal maps.



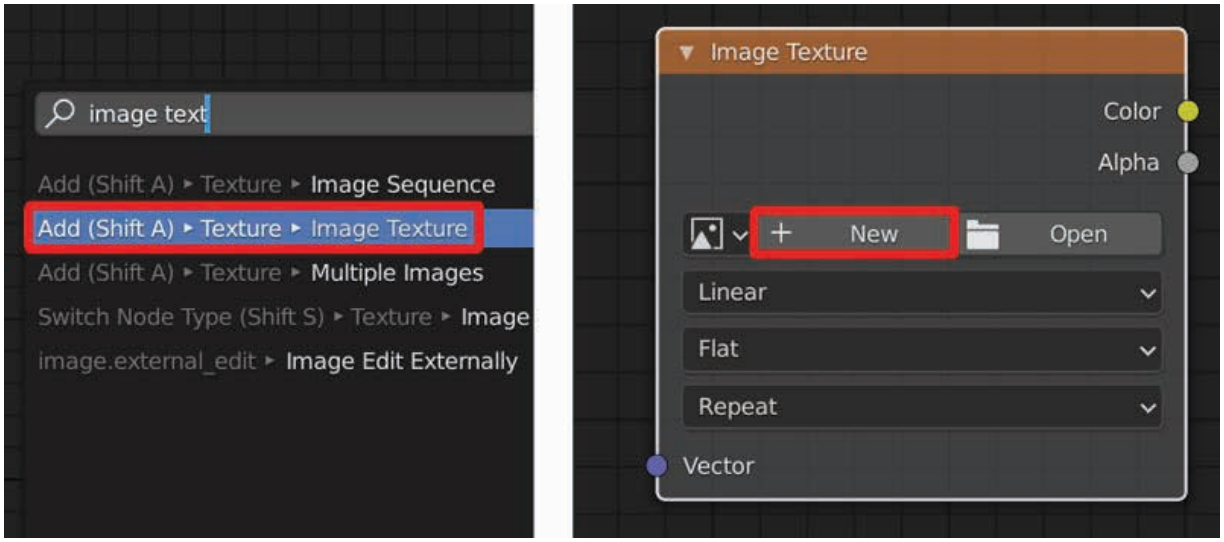
First, model some details on top of a Plane. Don't worry about the topology or n-gons, just go nuts! Select and join all the objects with **Ctrl+J**.



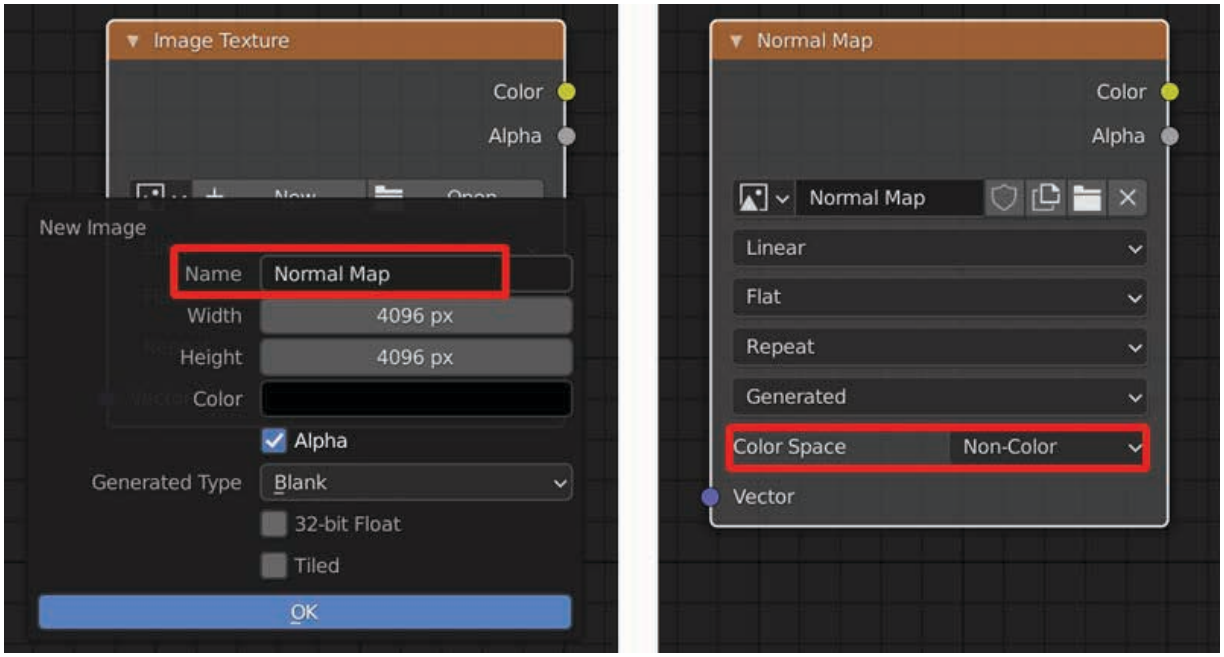
Note that I enabled **Auto Smooth** and **Shade Smooth** for these objects. **Press Shift+A** and add another Plane, place it above the plane with the details. Give that Plane a new material.



Press F3 in the Shader Editor window and search for "image texture". Create a new texture. Don't connect the Image Texture node to the material.

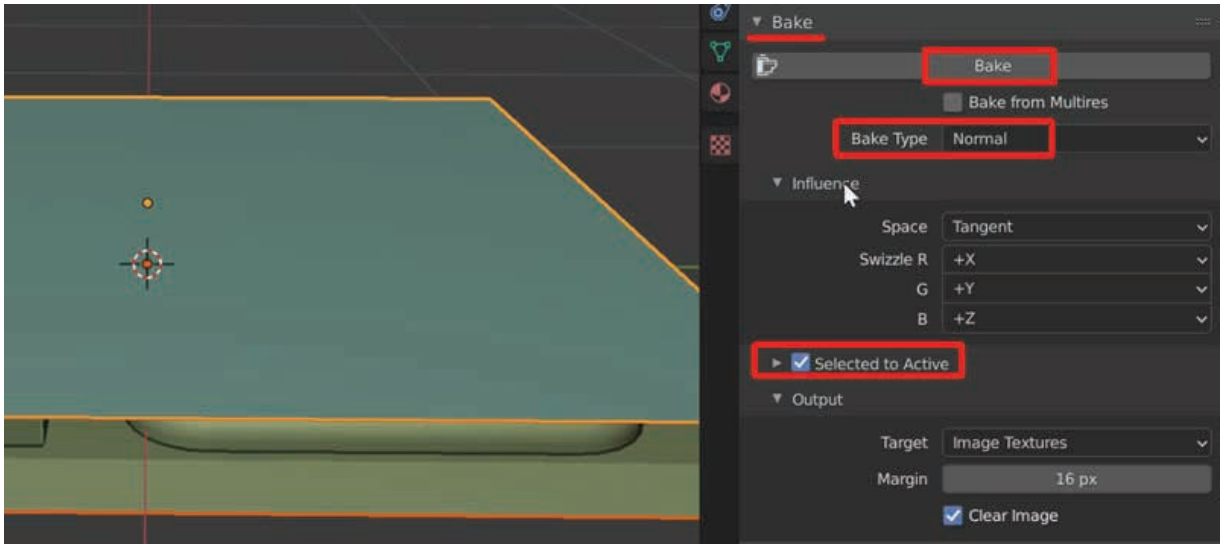


Name it something clear like “Normal Map”. Set it to **Non-color data**.



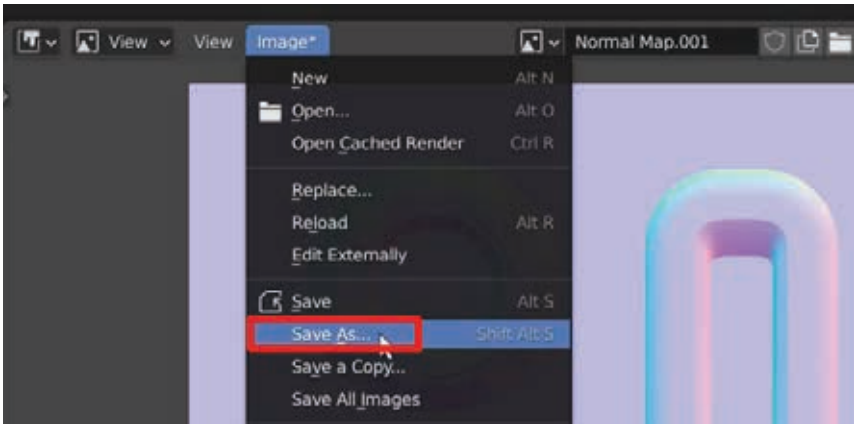
Select the detailed mesh, **hold Shift** and select the other Plane. In Cycles in the Render tab, go to the Bake settings. Set **Bake Type** to **Normal** and check “**Selected to Active**”.

Click on **Bake**.



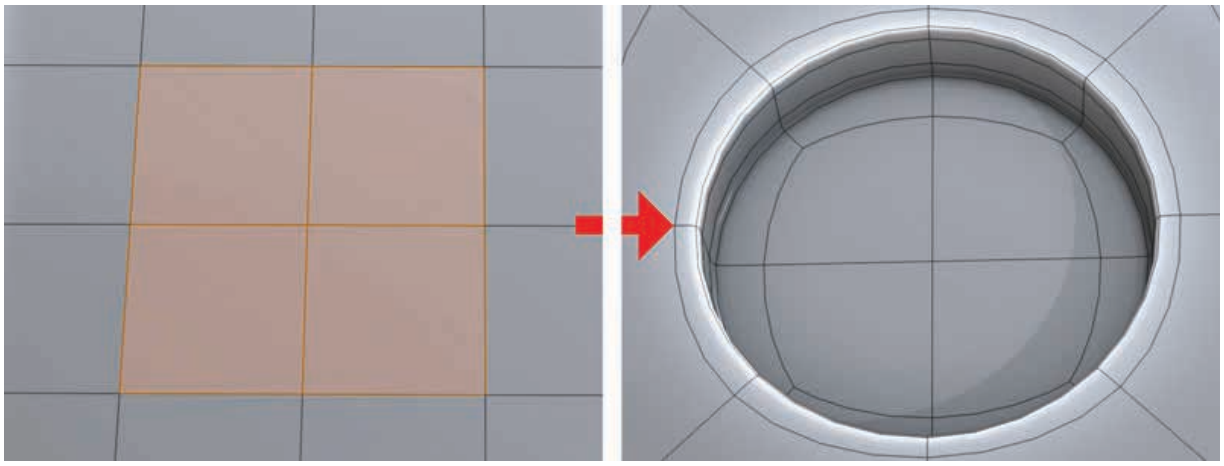
That’s it, you’re done!

Don’t forget to save your Normal map texture.

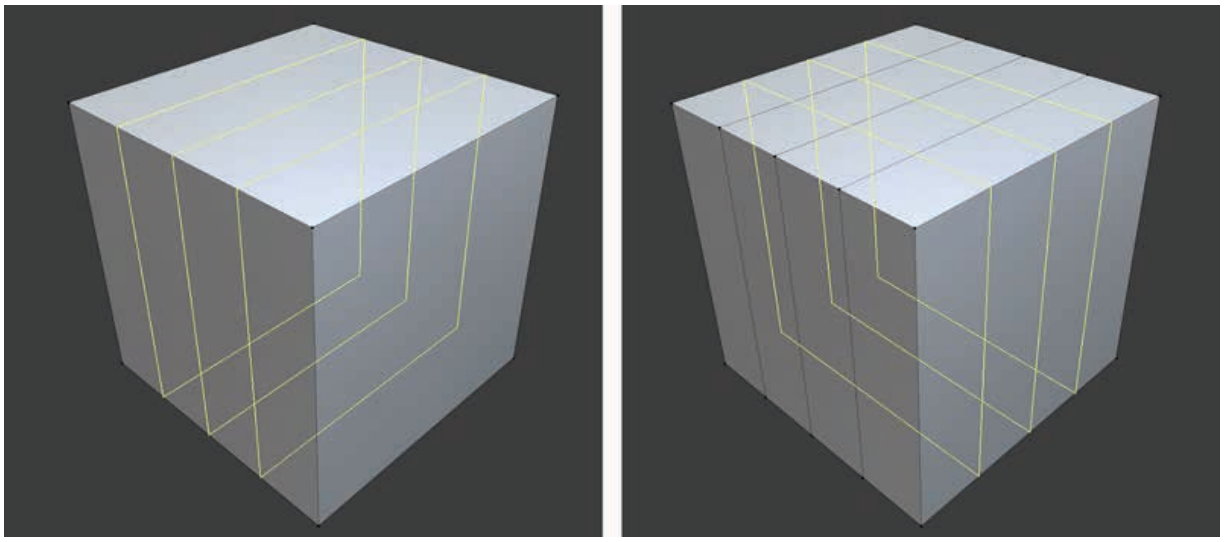


Modeling holes with Quad topology

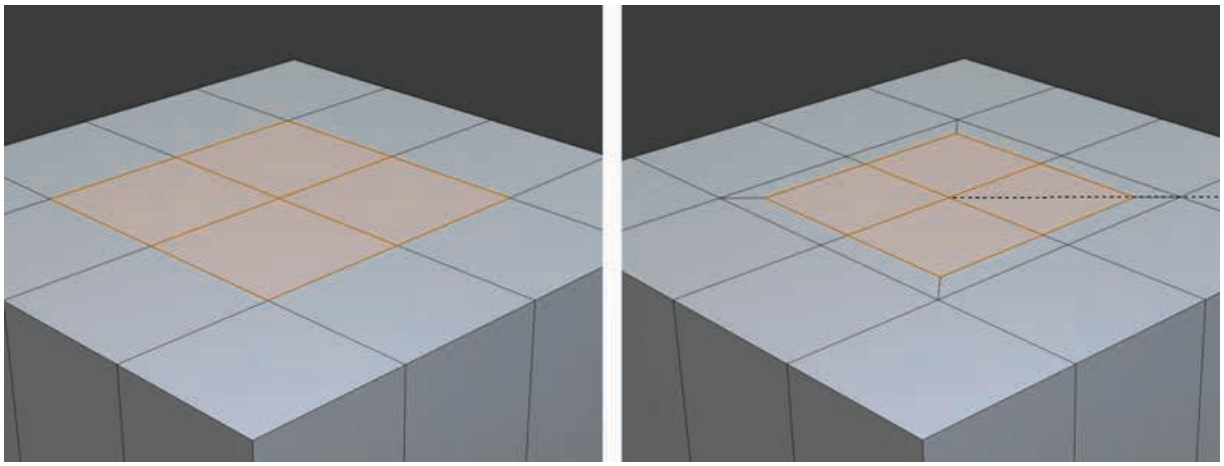
If you want to avoid n-gons and boolean artefacts, modeling with quads is the way to go.



For a round hole, you don't actually need that many loops. 3 edge loops per side of this cube will do. **Press Ctrl+R** and **press 3** on the numpad to add 3 edge loops.

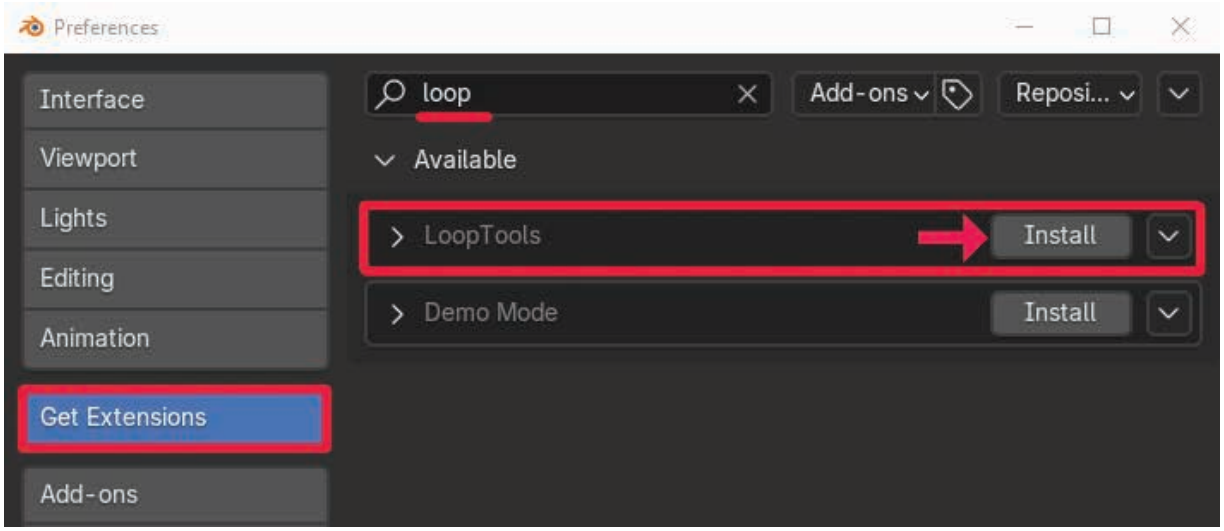


Now select 4 faces in the middle and inset them by **pressing i** and moving the mouse.

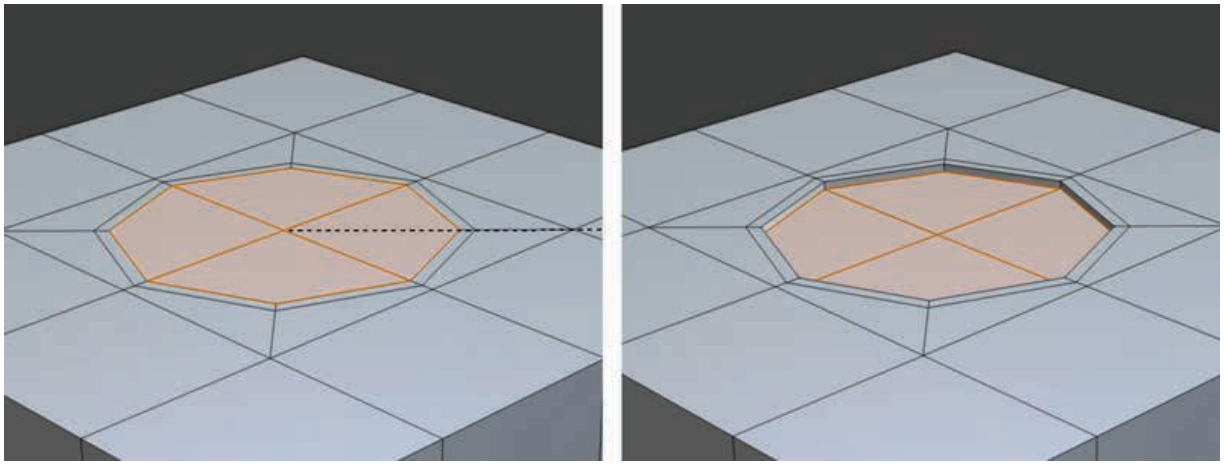


Right-click and from the **LoopTools** menu choose **Circle**.

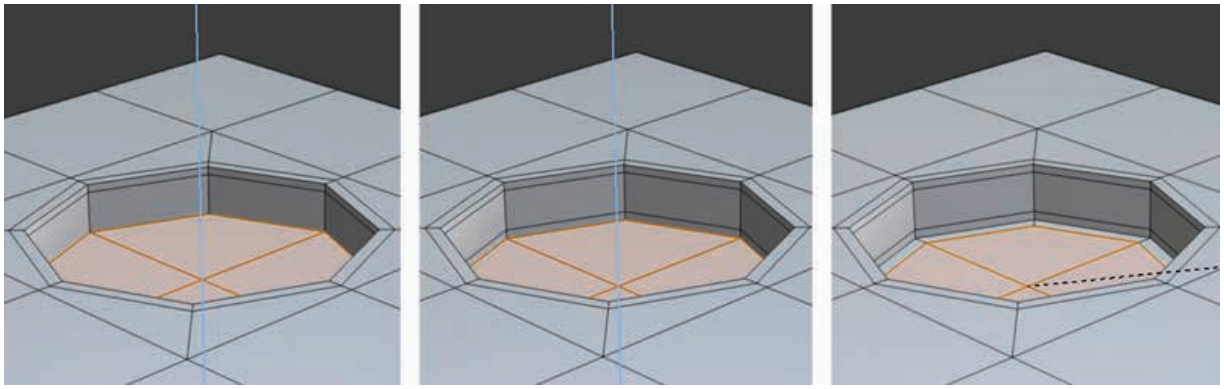
LoopTools add-on needs to be installed in Preferences in the Get Extensions tab.



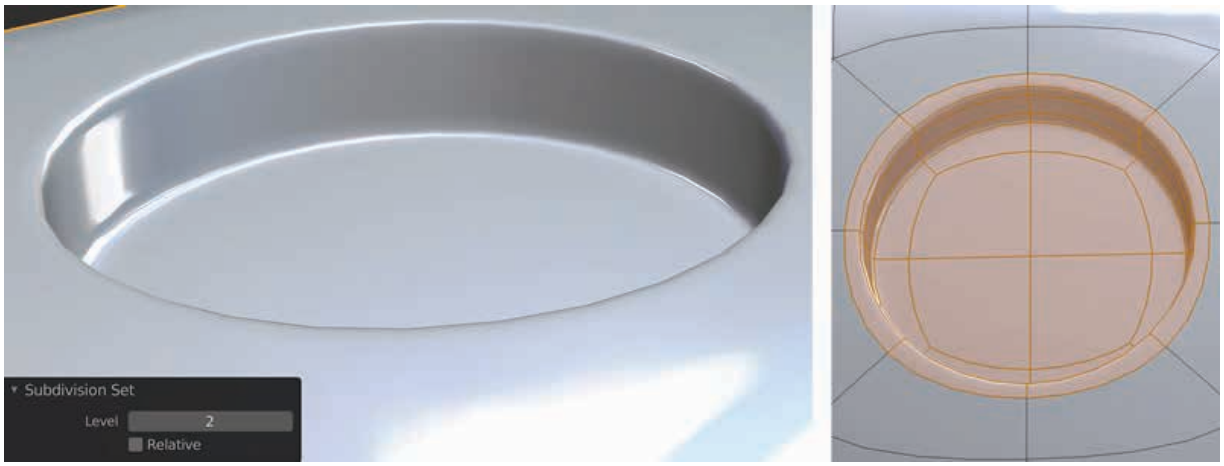
Inset the selection again slightly and extrude down by **pressing E and Z** to create some support loops.



Extrude further down and create some support loops there as well, if you like.



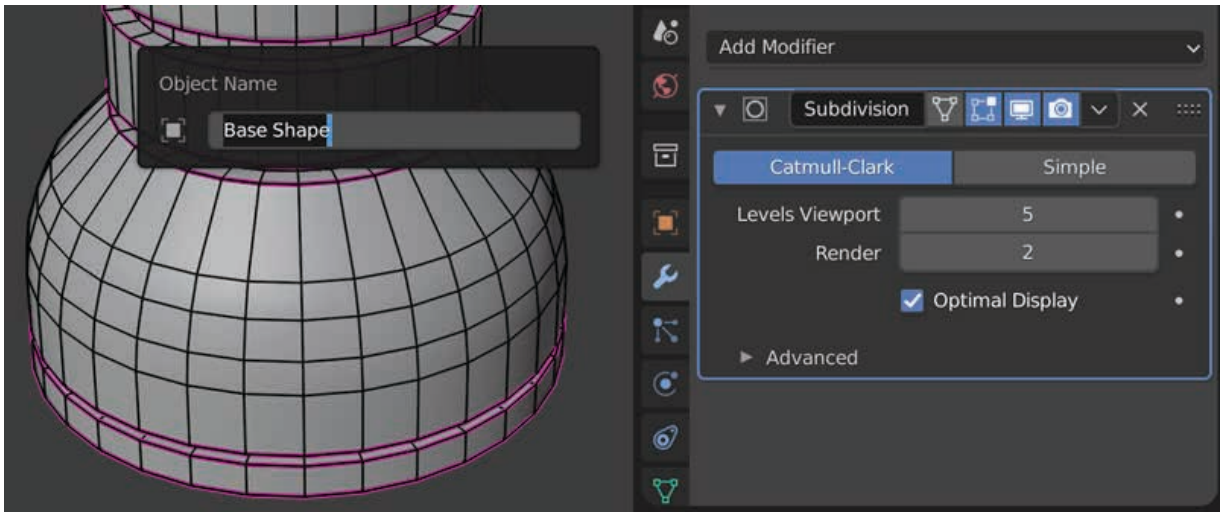
Now we've got a hole with perfect quad topology.



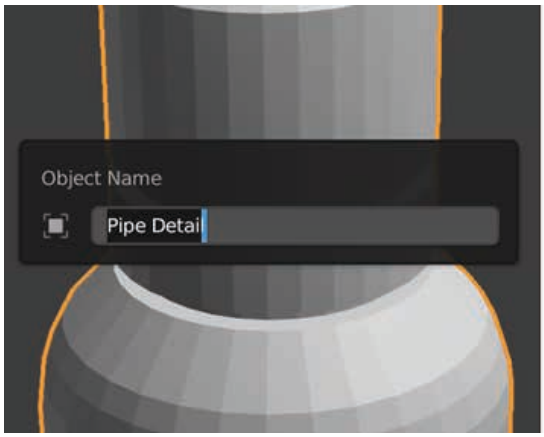
Perfect topology on Curved surfaces

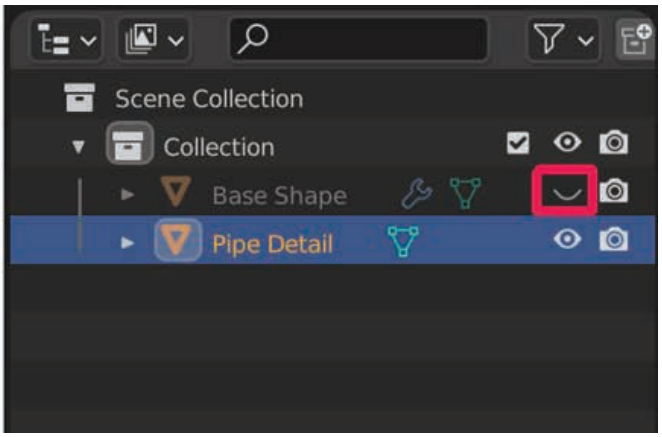
When modeling a circular object or other non-flat surface, you can add holes and other details using Knife Cut and Loop Tools. But it will usually break the surface smoothness.

The solution is to keep a smooth subdivided version of the base shape, duplicate it and hide this base shape. You can quickly rename objects in the viewport by **pressing F2** and typing a name. In this case I've added Creases by selecting edges and **pressing Shift+E+[1 on the numpad]** and added a **Subdivision modifier** to make the object smooth and sharp where I want it, in a non-destructive way.

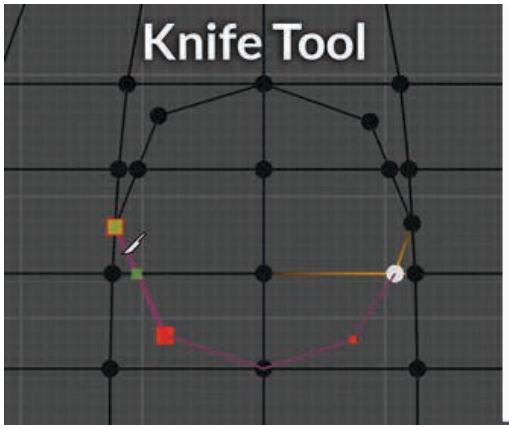


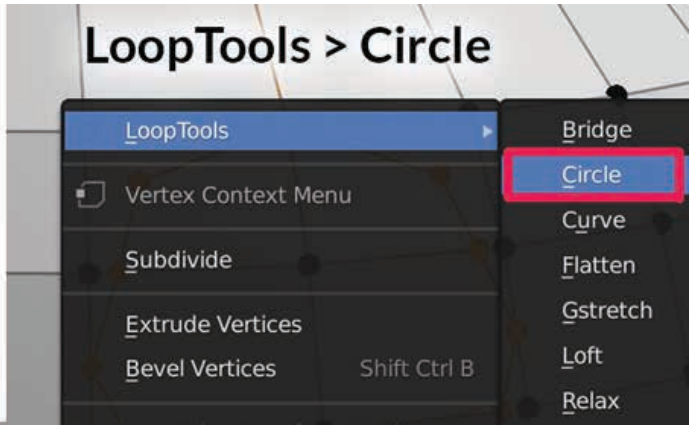
In Object Mode **press Shift+D** to duplicate the object and hide the "base shape" by disabling its visibility in the Outliner (see image below right). Give the duplicate a name as well. You can remove the Subdiv modifier from the duplicate object for now, it's just important that the base shape is very smooth.



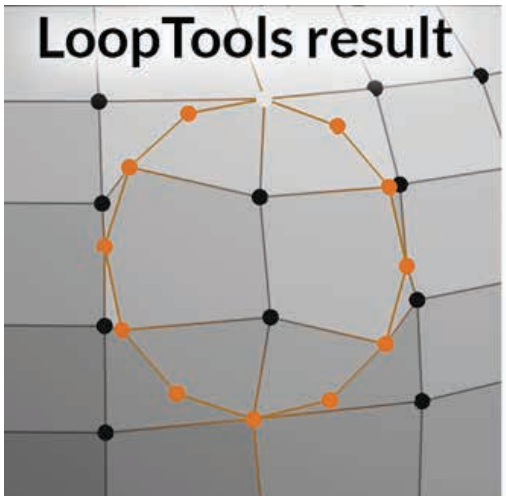


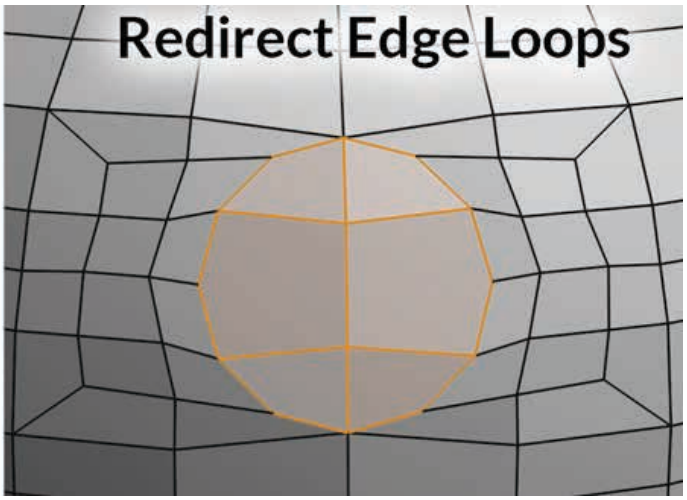
Then add your cuts and details to the duplicate. You can **press K** and use the **Knife Tool** to cut a circle, then right-click and use **Loop Tools > Circle** to make it perfectly circular. Note that Loop Tools needs to be enabled in Preferences under Add-ons.



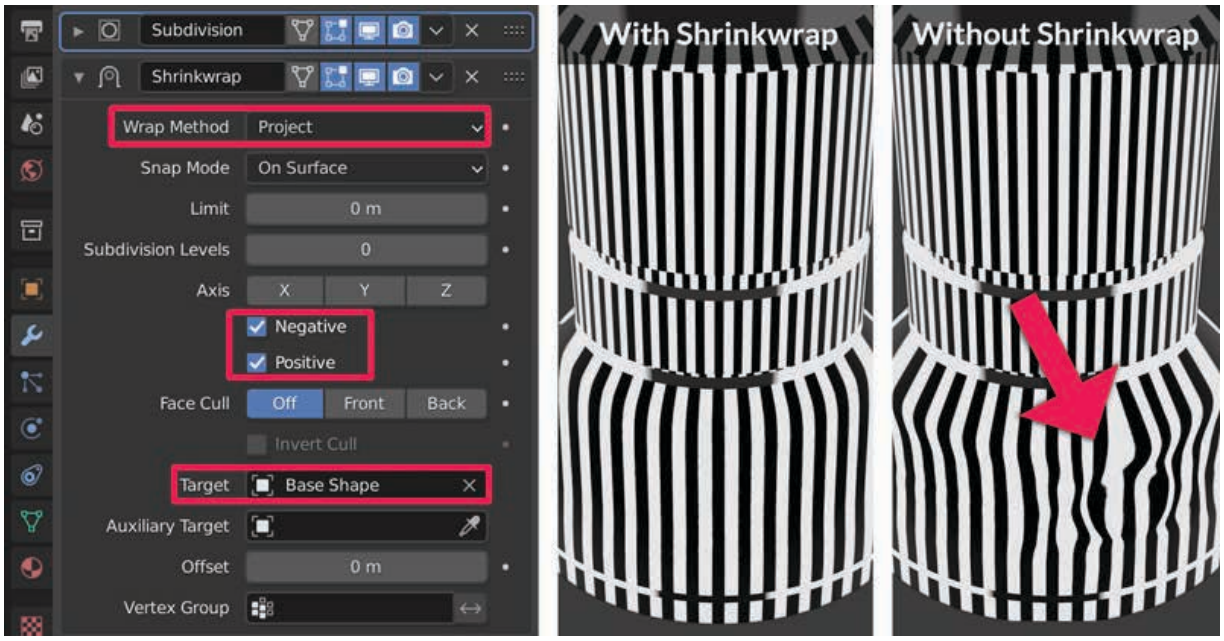


After that, you can add some cuts to redirect loops so as to avoid N-gons and triangles.

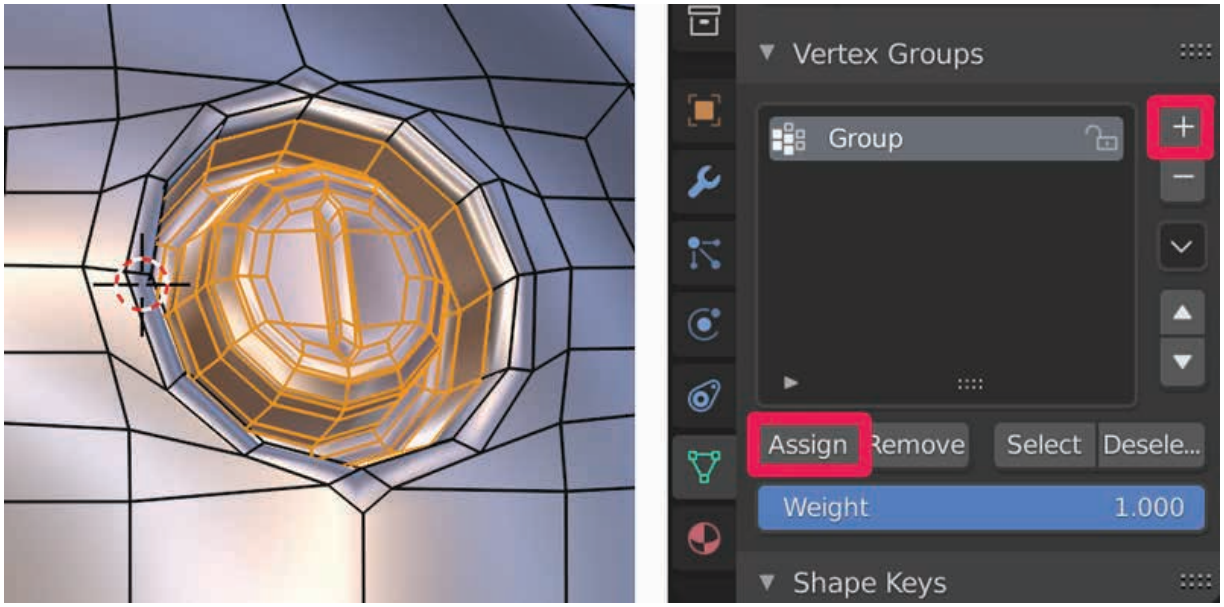




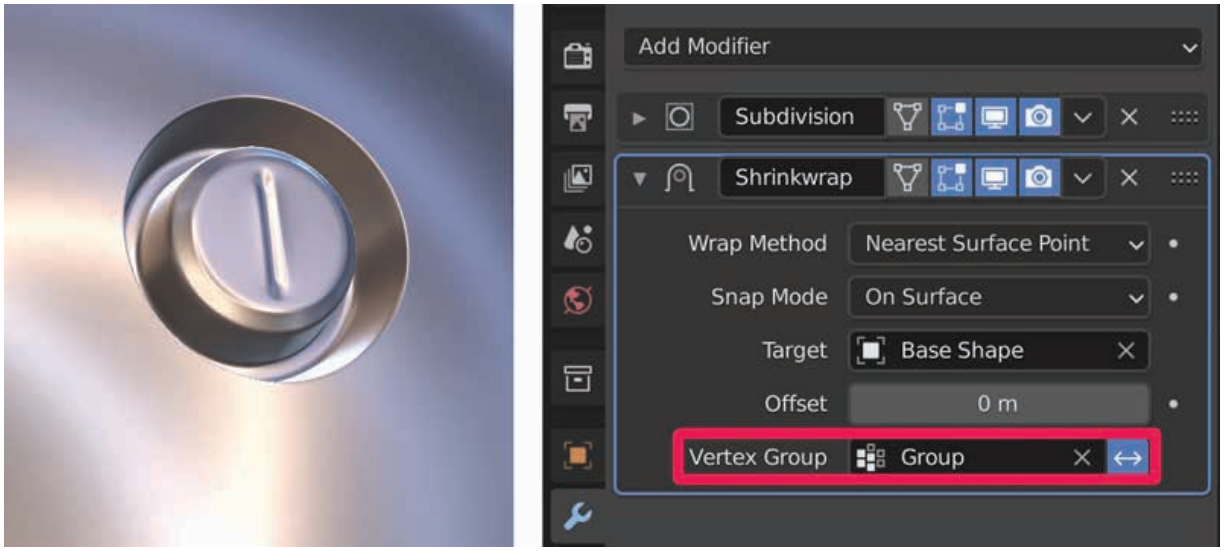
Using a **Shrinkwrap modifier** with the **base shape** as the **Target**, you can keep it perfectly smooth. Note that I'm also using a **Subdivision modifier** to add some geometry necessary for smoothness. Below you can see the difference with and without Shrinkwrap modifier, using a high-contrast Matcap meant to be used for finding surface imperfections:



Before you start extruding geometry you can either apply the Shrinkwrap modifier or keep it, but turn it off temporarily and add the extruded parts to a **Vertex Group** so you can exclude them from the modifier when you turn it on again.



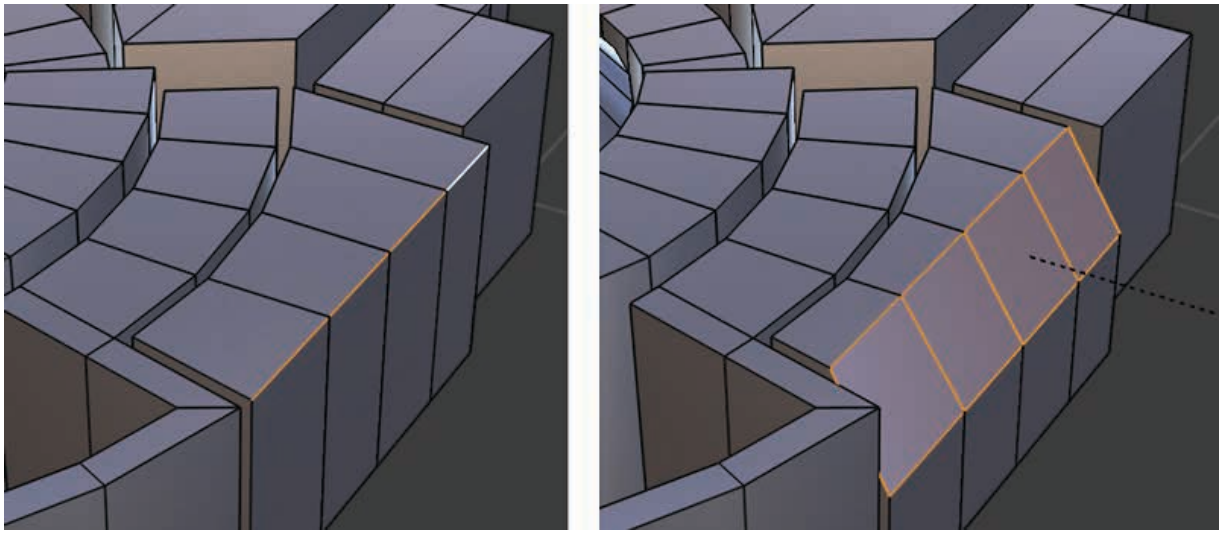
Add the Vertex Group to the Shrinkwrap modifier and **click the <-> button** to invert it. I used the default Wrap Method "**Nearest Surface Point**" instead of "Project" as it gave a better result in this case.



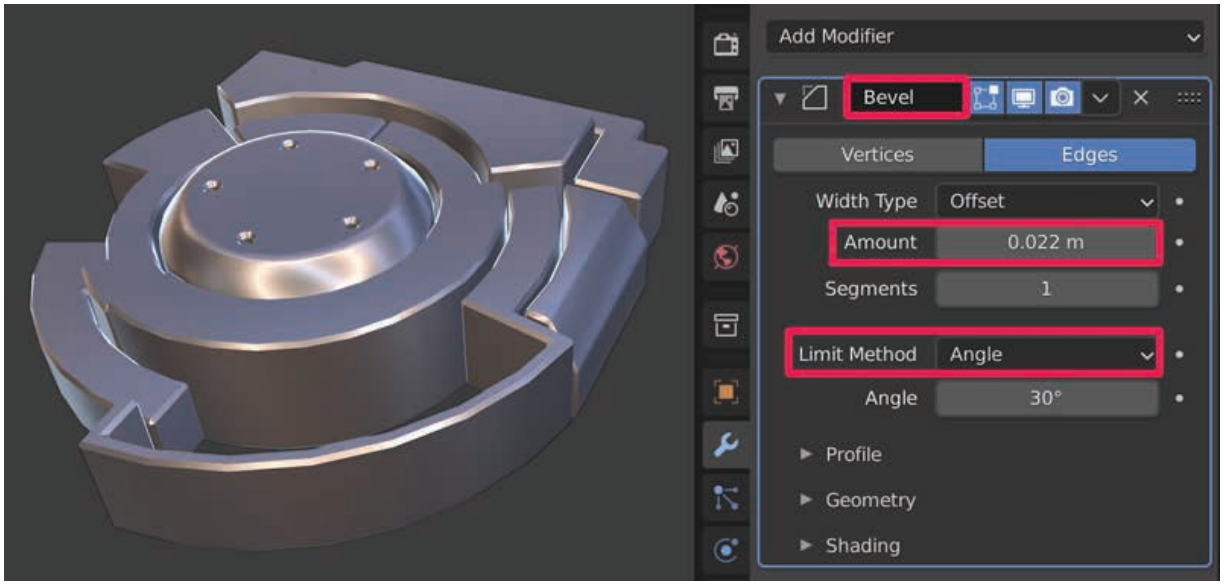
Note: Both the Base Shape and the duplicate object need to be in the same exact location for the Shrinkwrap modifier to work correctly.

Do you know all 5 of these Bevel methods?

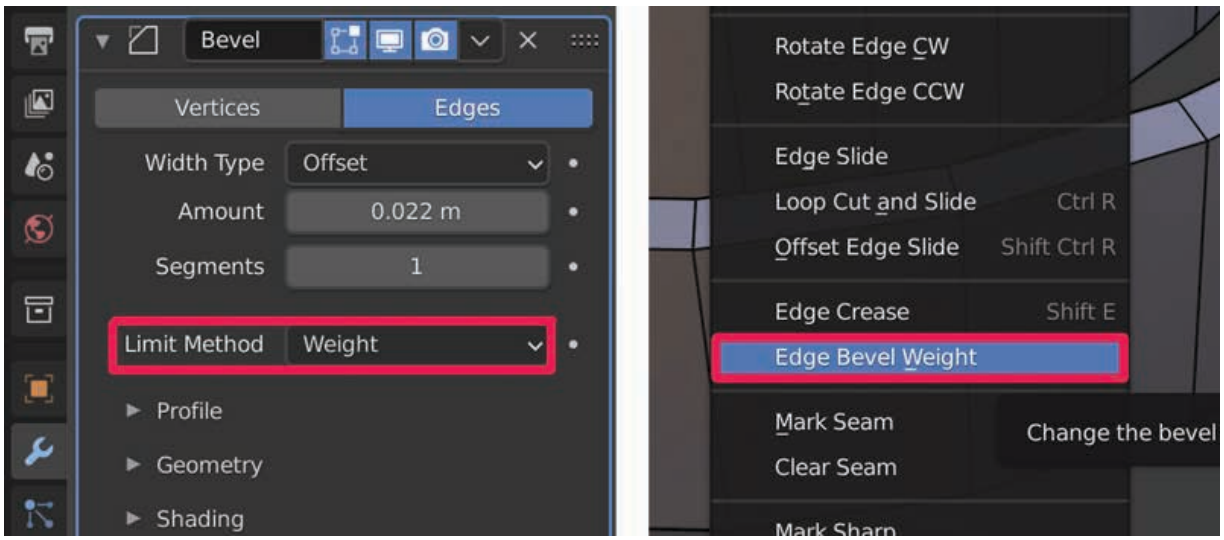
1. Beveling with Ctrl+B
Select the edges you want to bevel and **press Ctrl+B**. This is mainly recommended for big bevels that really affect the object shape.



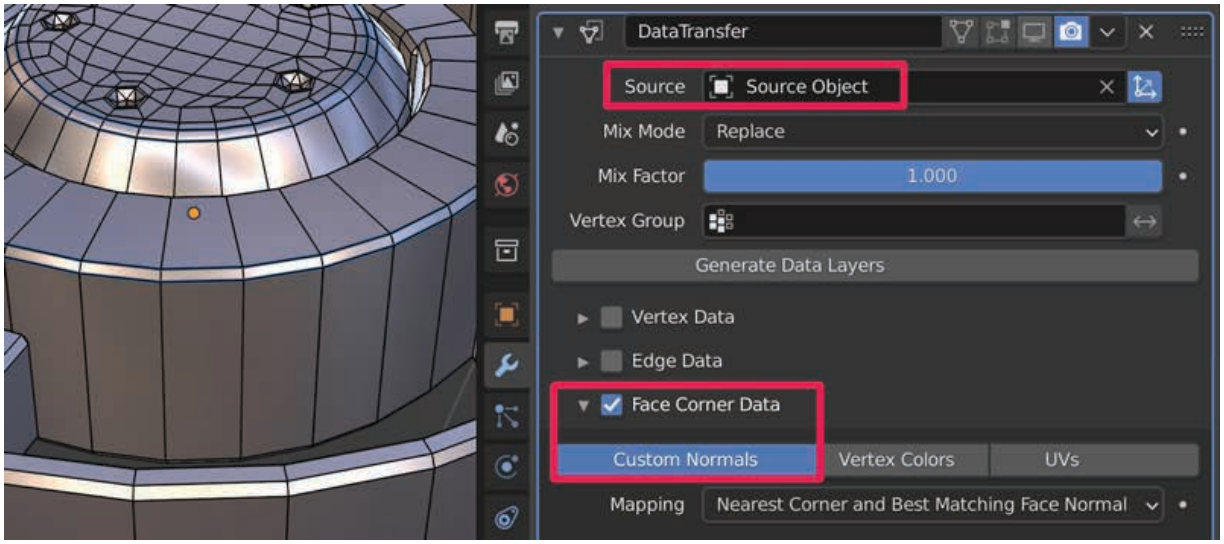
2. The Bevel modifier: Angle
The Bevel modifier on the other hand can be used for smaller bevels. Set it to **Angle** and change the value for an overall bevel. Increase the **Segments** if you want a smoother bevel.



3. Bevel Edge Weight
Set the Bevel modifier to **Weight** and select the edges you want to bevel, **press Ctrl+E** and choose **Bevel Edge Weight**. Then drag the mouse. The benefit of this is that you can get different bevels from a single Bevel modifier.

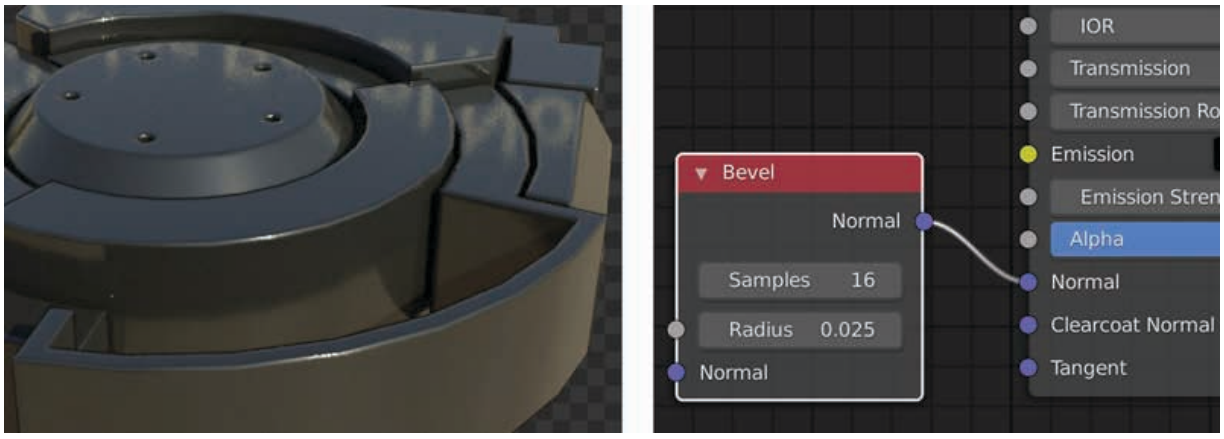


4. Data Transfer modifier
If your object has a simple two-edges bevel, you can make it look much smoother using a **Data Transfer modifier** and using a smoother beveled object as the **Source**. Note that both objects need to be in the same location. **Face Corner Data** needs to be checked, set to **Custom Normals**.



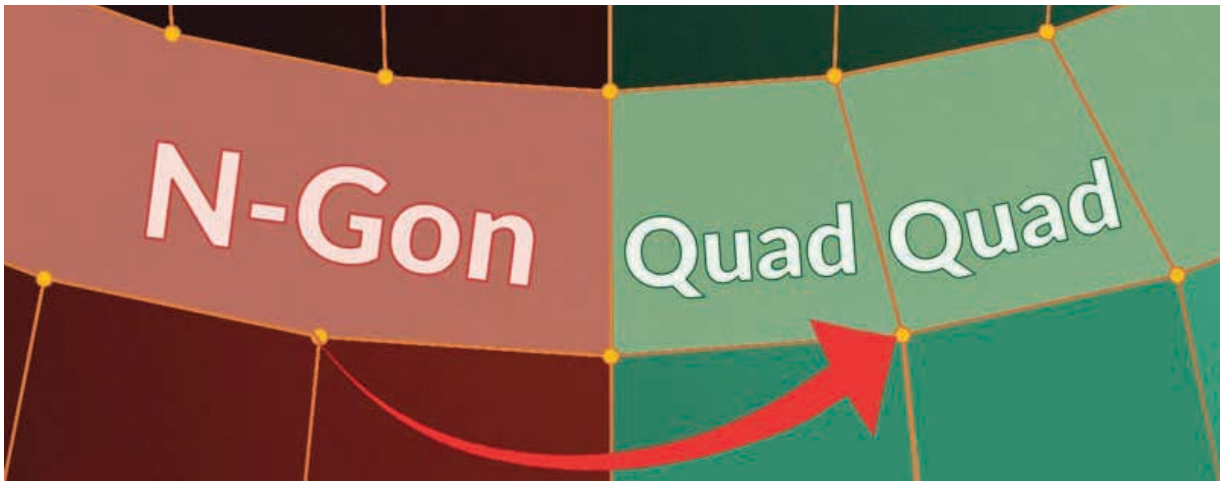
Also, **Auto Smooth** needs to be enabled for this to work. You can keep the Source object hidden, or once you've applied the modifier you can simply delete the Source object.

5. Cycles Bevel Node
Finally you can use a Bevel Node on the material if you're rendering in Cycles. Note that you won't see this bevel in the viewport, only in the render.

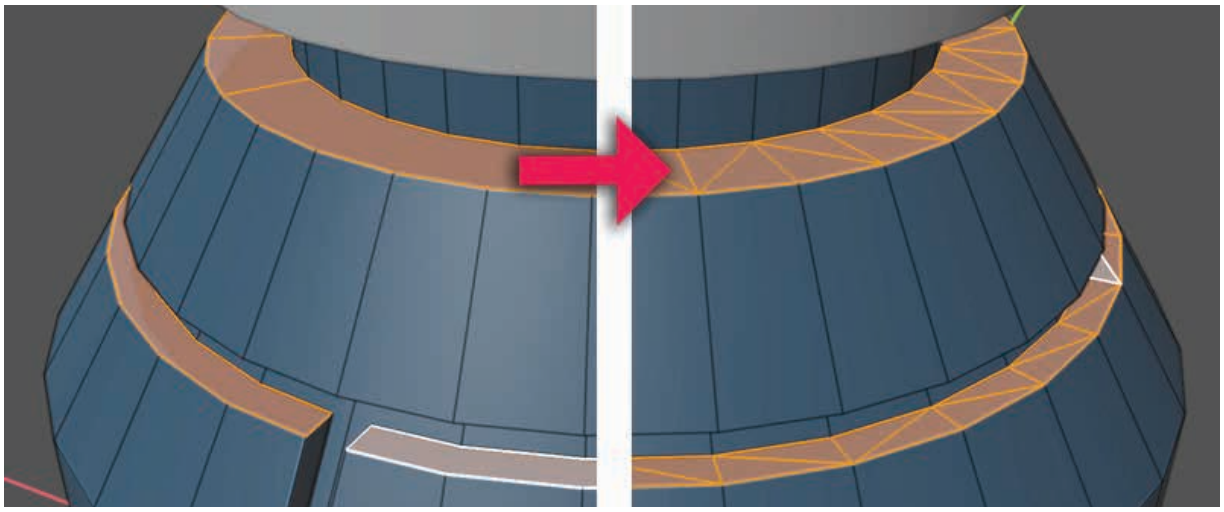


Turn N-gon pipes into Quad pipes in a couple of clicks!

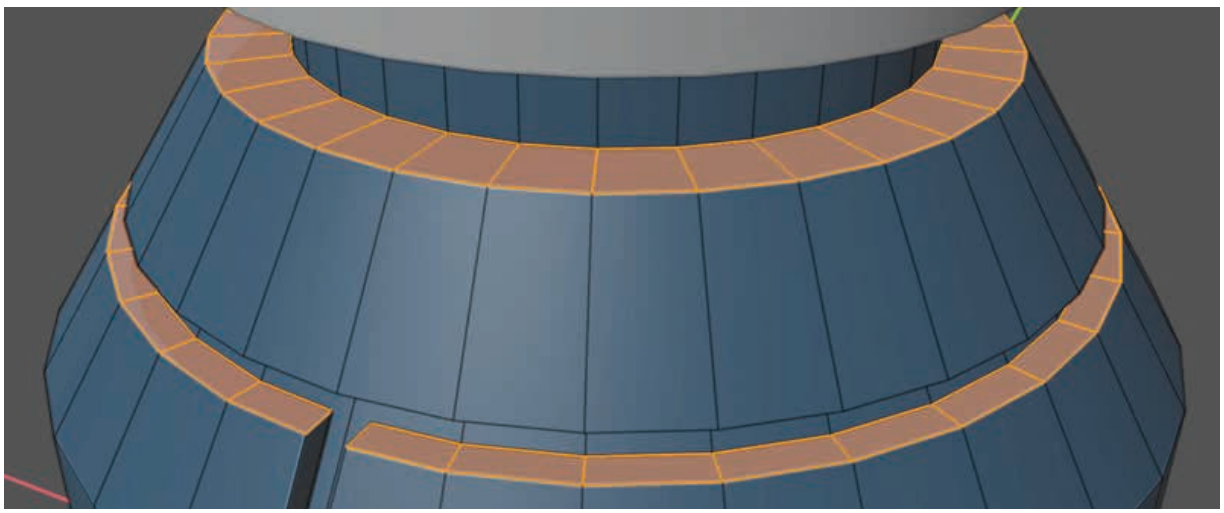
When modeling with Booleans, you sometimes will get into this situation where you've got some big N-gons surrounded by quad faces. Here's a quick way to turn them into all quads.



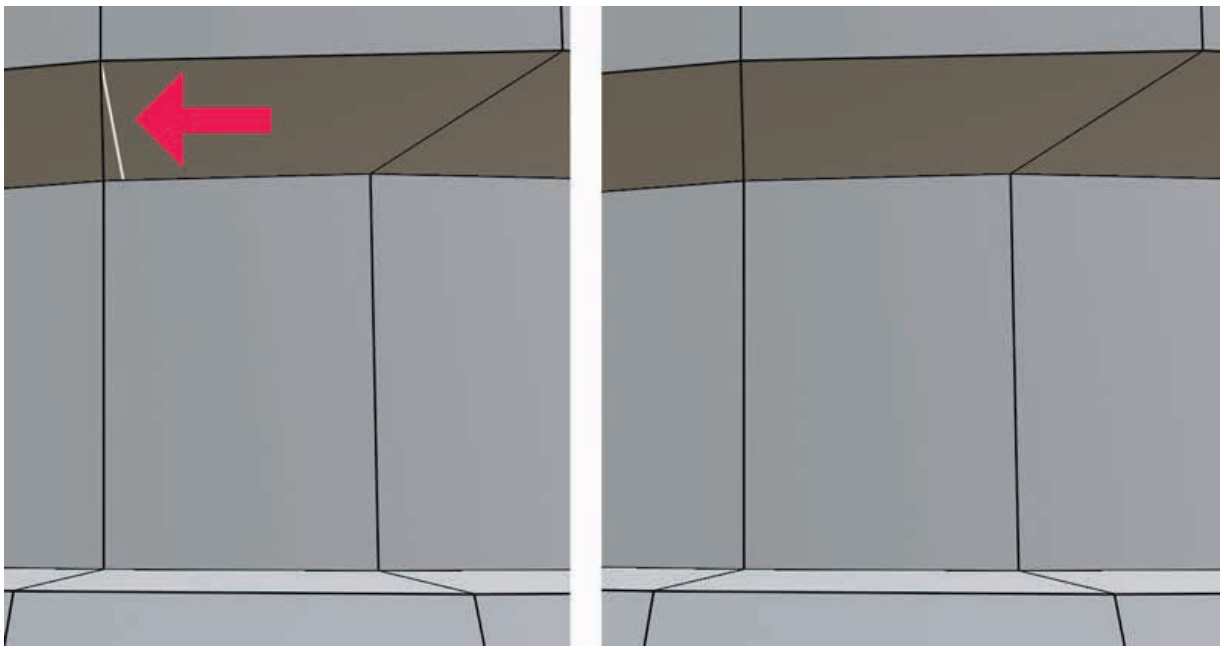
Select those N-gons and **press Ctrl+T** or go to **Face > Triangulate Faces.**



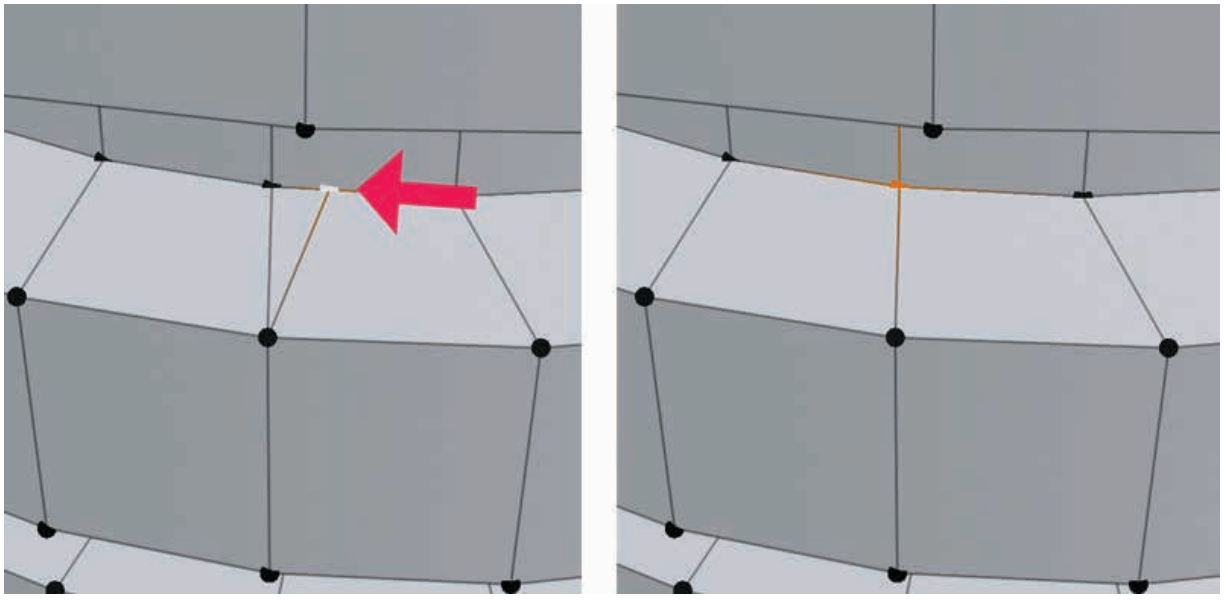
Then **press Alt+J** or go to **Face > Tris to Quads**. That's all!



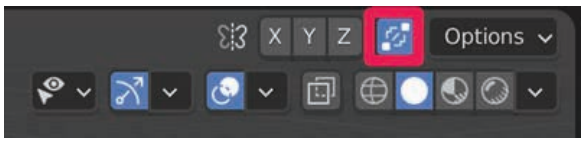
In some cases you may still need to manually do some cleanup. For example by selecting unnecessary edges and dissolving them by **pressing Ctrl+X**.



Or by sliding vertices by **pressing G twice** and merging them automatically by having **Auto Merge Vertices** enabled.

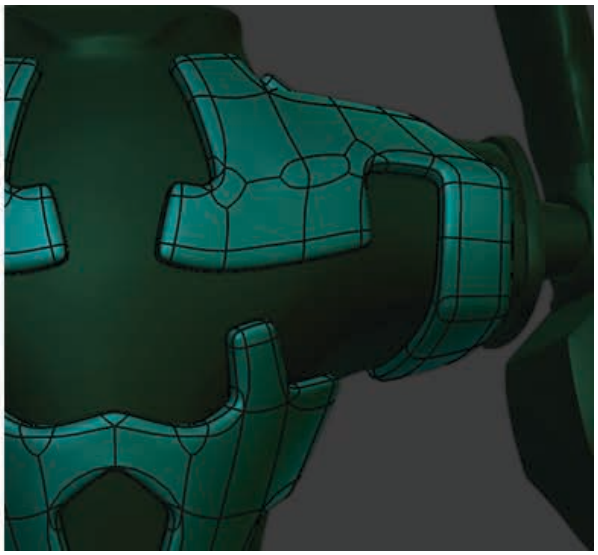


The **Auto Merge Vertices** button is located in the top right corner of the viewport.

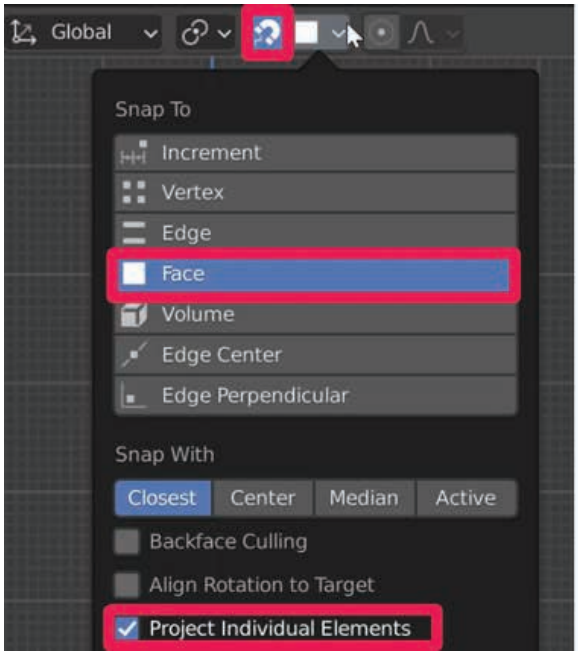


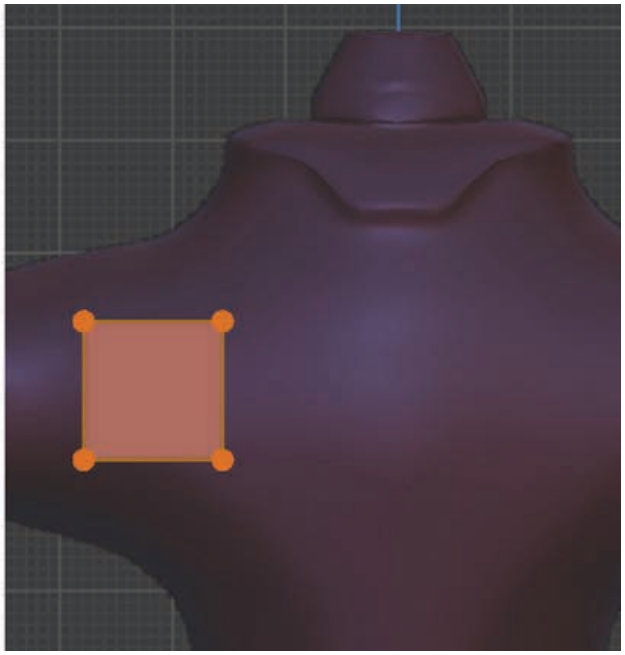
Easily Extrude Thick Panels

By adding a few modifiers to a Plane, you can turn it into a nice beveled panel that you can quickly extrude to add details to, for example, a robot model.

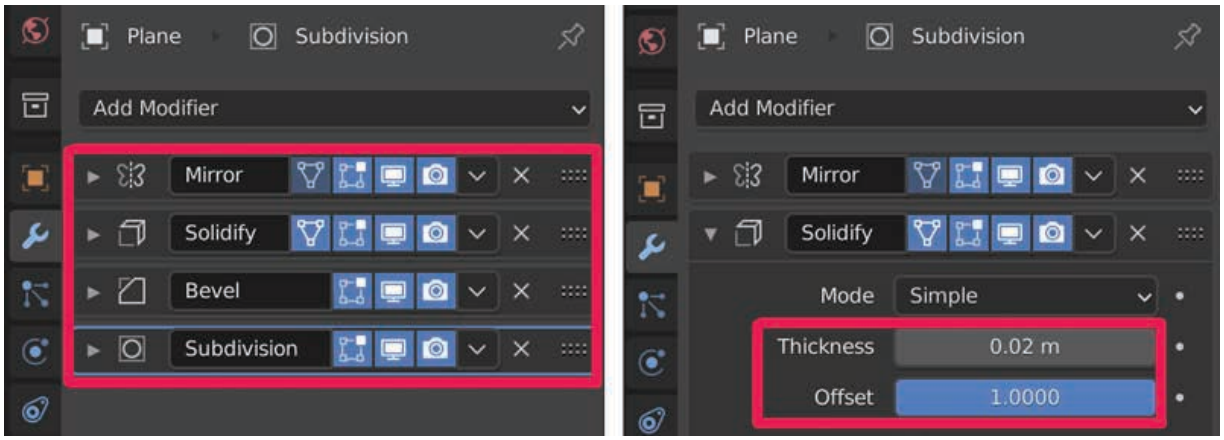


Turn on **Snapping** to **Face** and enable **Project Individual Elements**. Add a **Plane** and in Edit Mode select all by **pressing A** and move it by **pressing G** and moving the mouse, to snap to the surface.

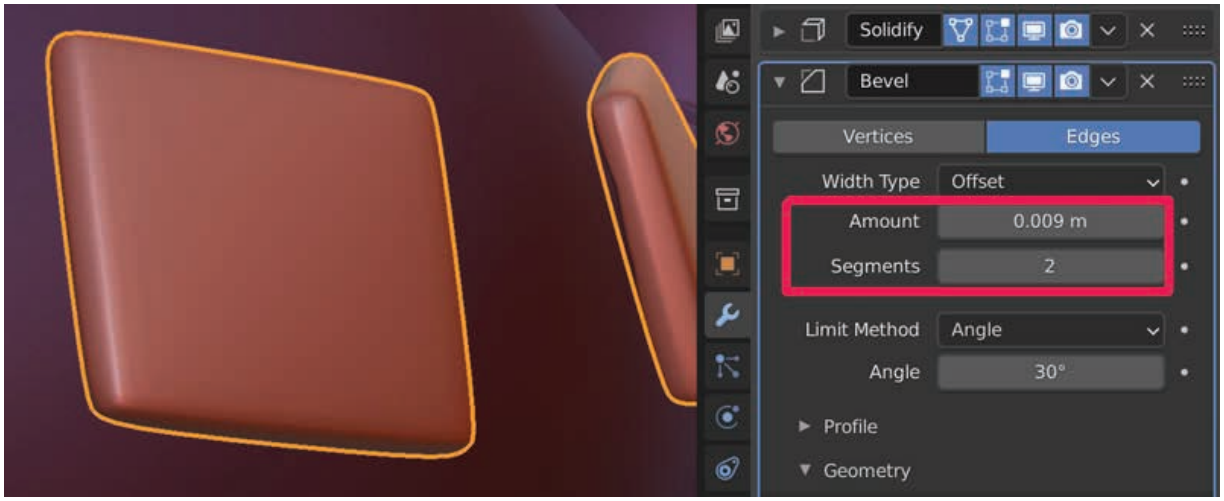




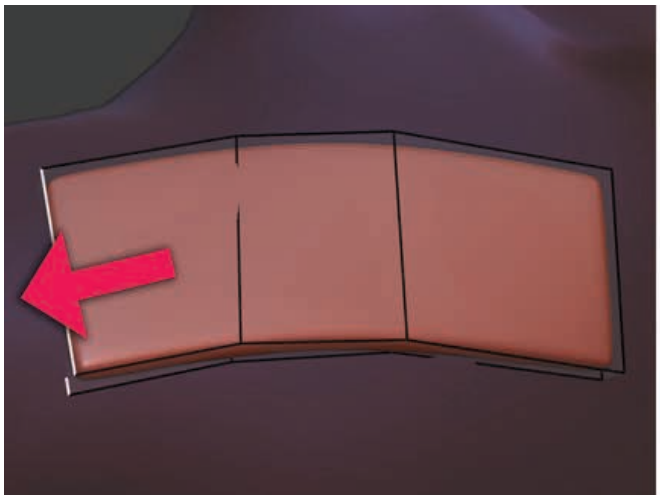
Add a **Mirror, Solidify, Bevel and Subdivision modifier** to the Plane (in that order). Increase the **Thickness** value of the Solidify modifier. You can set the **Offset to 1** if you don't want to use a negative value for the Thickness.

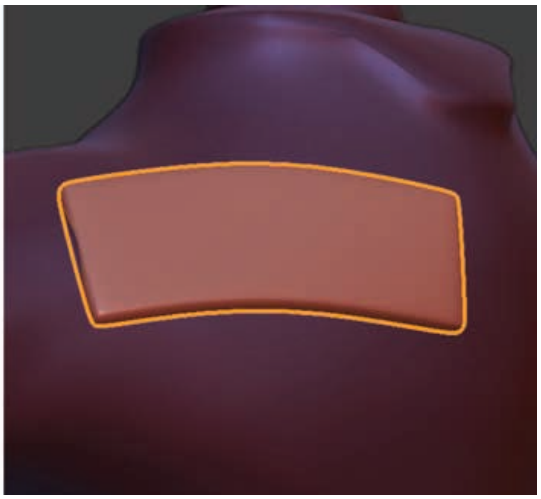


Decrease the **Bevel Amount**. Set **Segments to 2** if you want a sharper bevel. Increase **Subdivision Levels to 2**. Right-click on the object in Object Mode and choose **Shade Smooth**. Now you should get something like this result, a thick plane with beveled edges.

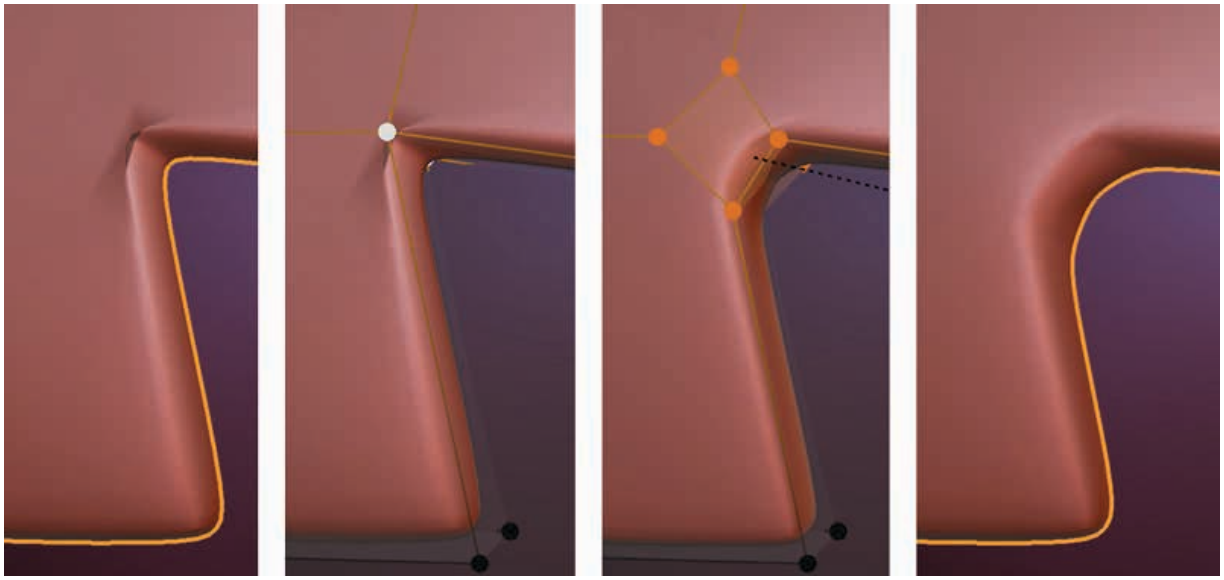


With the modifiers set up, select an Edge and either **press E** to extrude... Or **hold Ctrl and click the right mouse button** to where you want to extrude to.

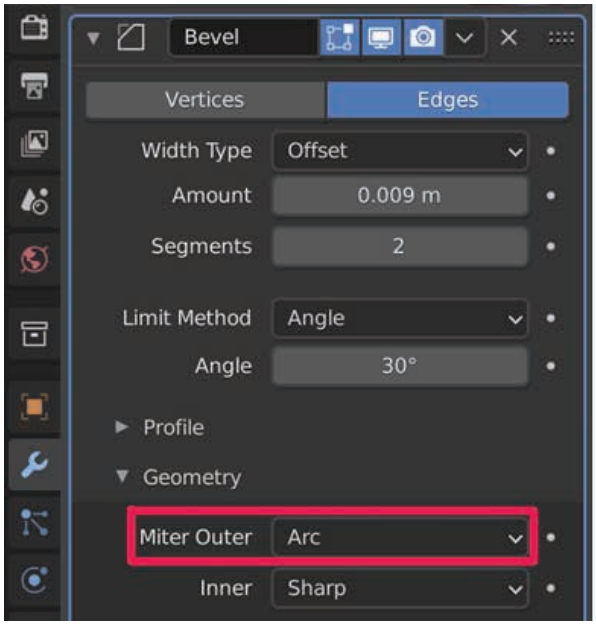




To fix ugly corners, select the vertex and **press Shift+Ctrl+B** to add supporting geometry.

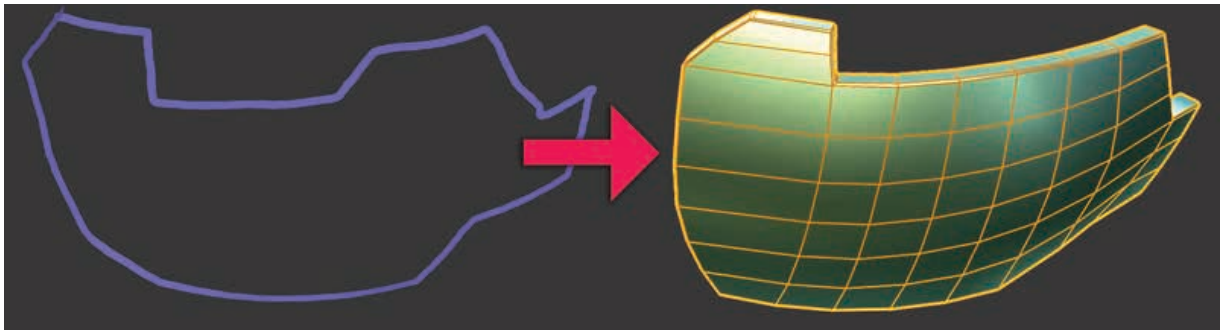


Alternatively you can set **Miter Outer to Arc** instead of **Sharp**. Keep in mind this may create a visual look that you may not want in the rest of the panels, so choose the method that fits your specific case.

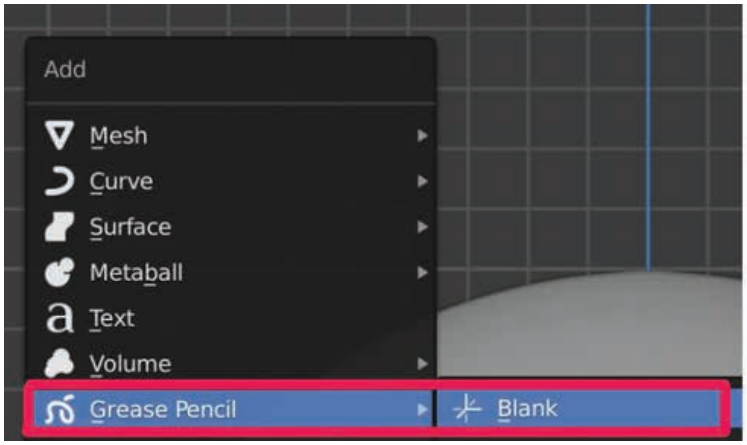


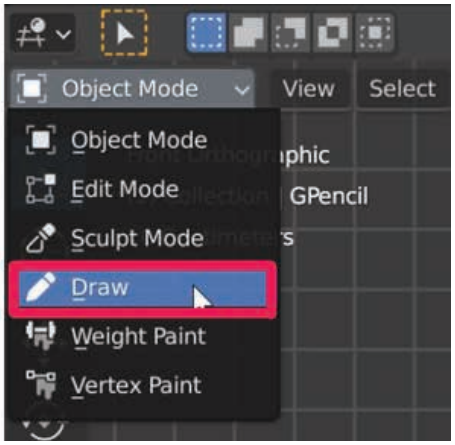
Grease Pencil Strokes to Mesh

Instead of using the Annotate tool to plan your retopology or design hard surface panels, you can also use Grease Pencil to draw a design on top of your mesh. One advantage of this is that you can convert Grease Pencil strokes to meshes, which opens up the possibility of at least having some geometry when starting to make the mesh.

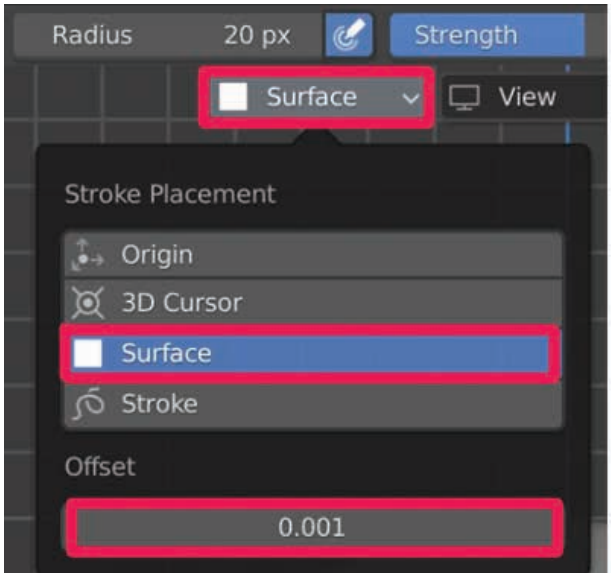


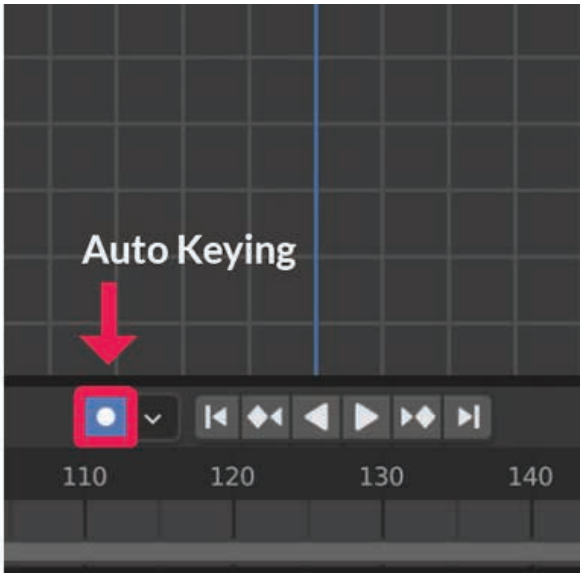
The downside is that it's more work to set up, in my opinion, as the Grease Pencil-originated mesh requires quite a bit of cleanup to be useful. However it's good to know this method exists, it may come in handy in some cases. **Press Shift+A** and add a **Blank Grease Pencil** object. Go to **Draw Mode**.



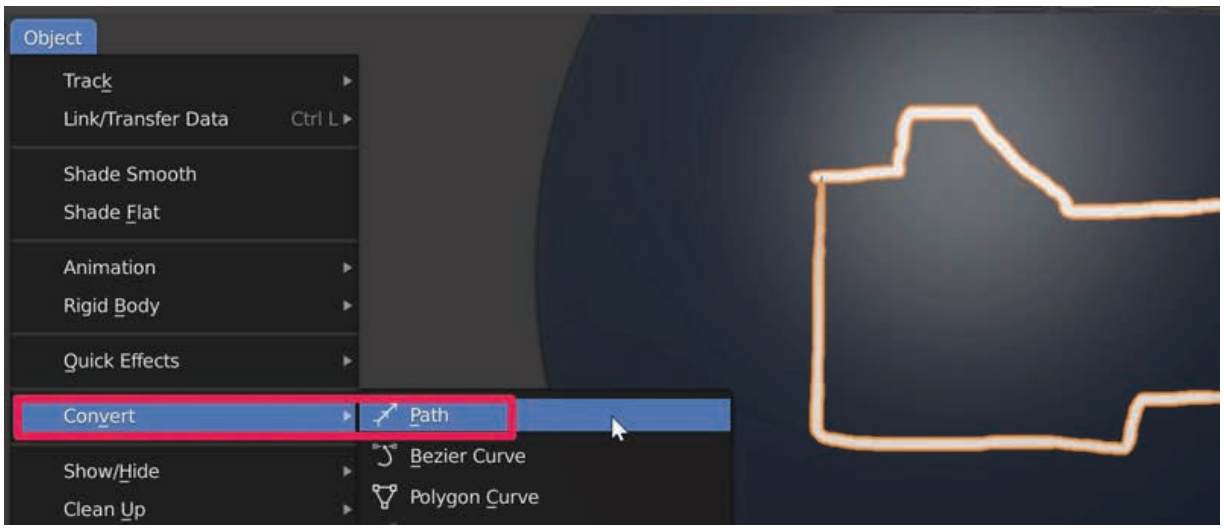


Set Stoke Placement to **Surface** and reduce the **Offset** to about 0.001. Enable **Auto Keying** to see the stroke (You can turn this off after drawing the first stroke).

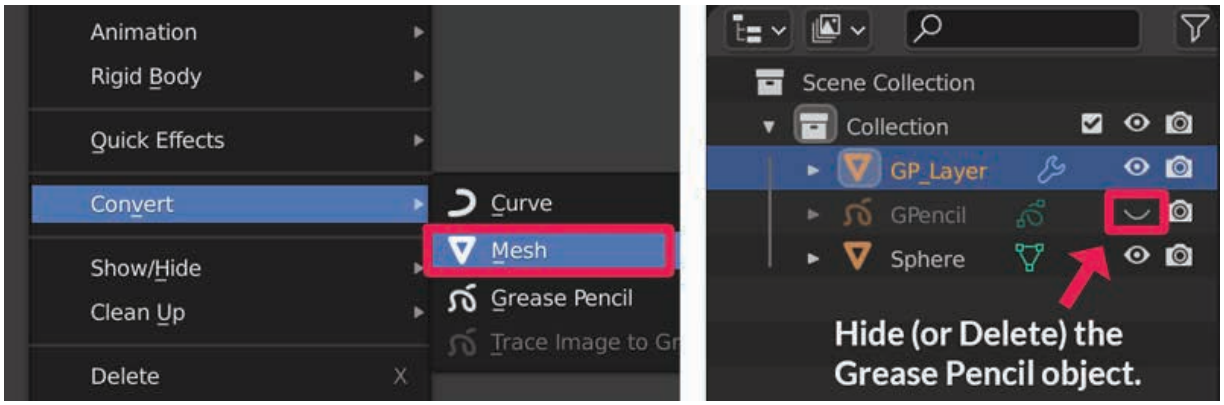




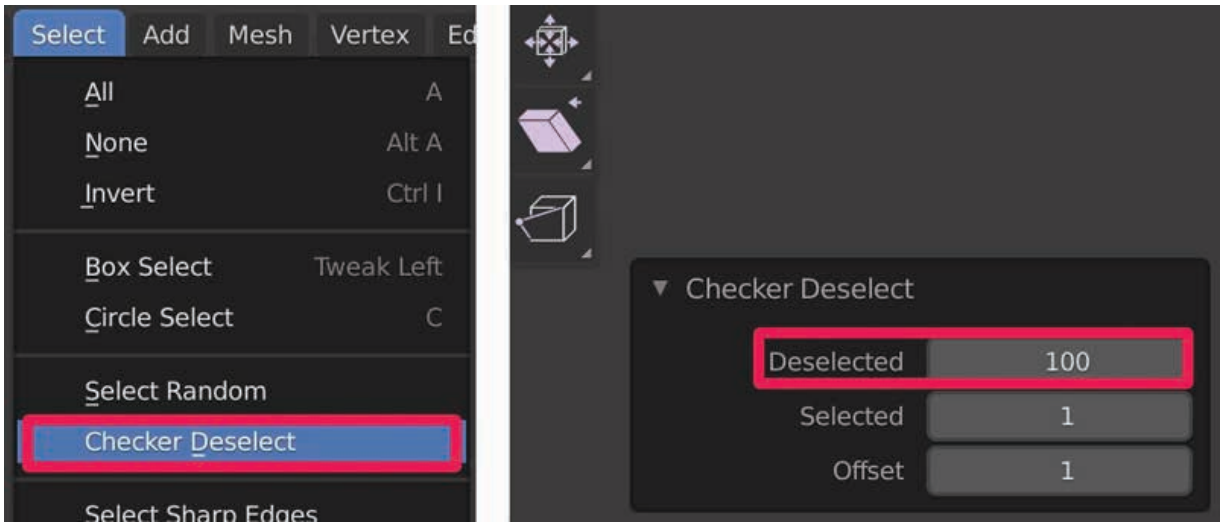
Now you can draw on top of a mesh. Draw your design, then in Object Mode **convert it to a Path**.



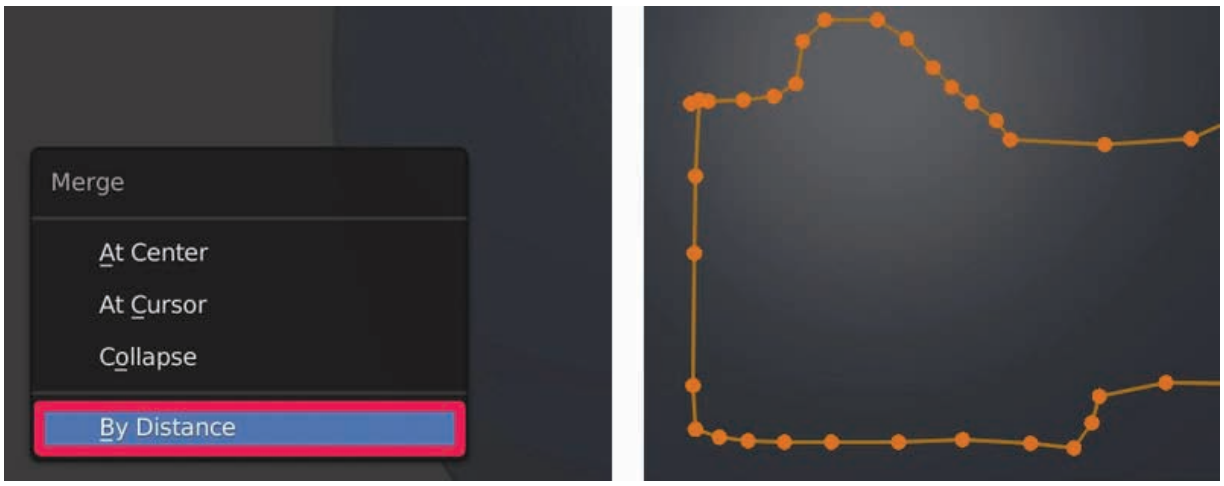
Then convert the Path to a **Mesh**. Hide the Grease Pencil object in the Outliner (or just Delete it).



Select all vertices of the the converted Mesh in Edit Mode and go to **Select > Checker Deselect**. Set Deselect value to **100** (the maximum).



Press Ctrl+i to invert, then **Ctrl+X** to dissolve the selection. **Press M** and merge **By Distance** to get rid of any doubles or unconnected parts.

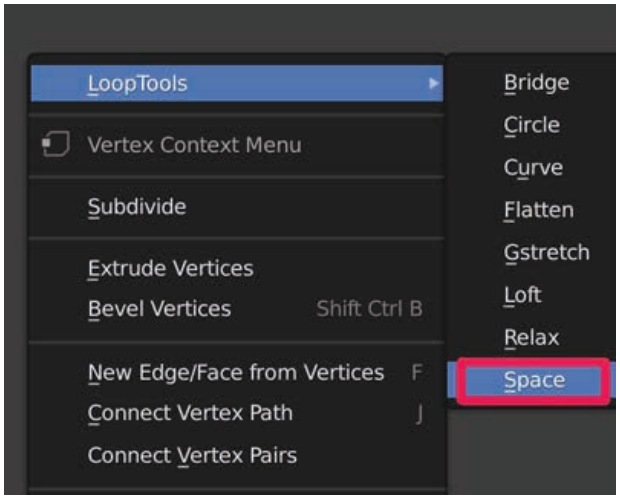


If you want you can do another **Checker Deselect** with Deselected value of 1.

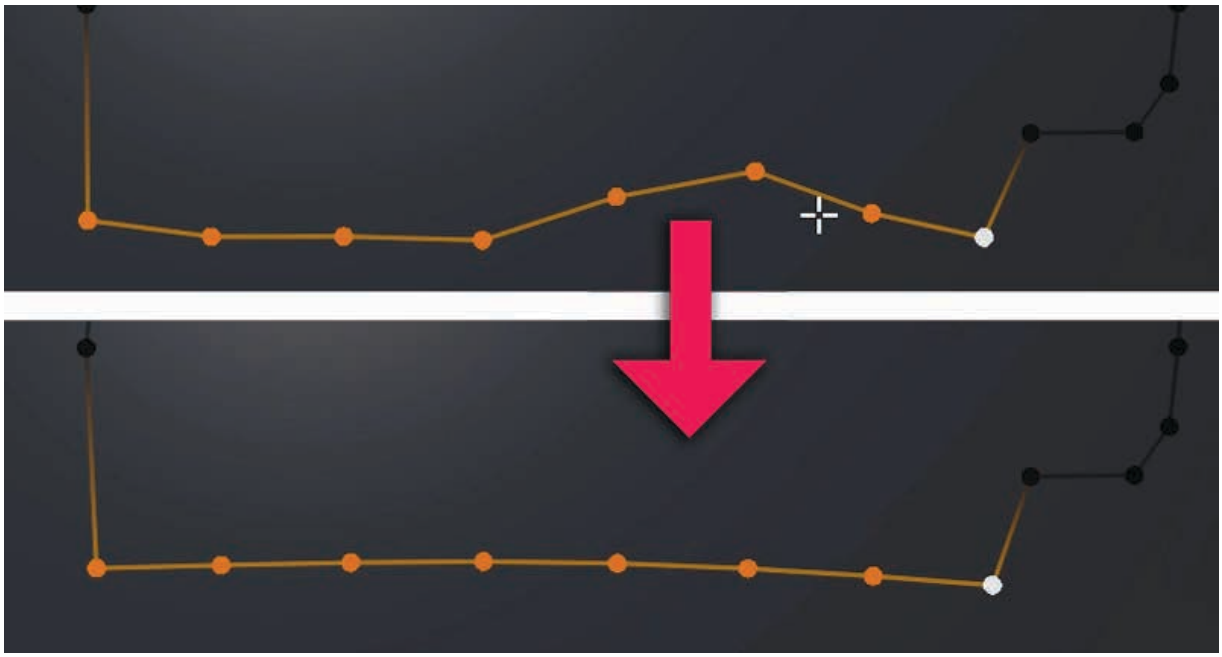
Select an edge by selecting the first and last vertex while **holding Ctrl**.

Space out the selected vertices with **LoopTools > Space**.

Note that you need to have the LoopTools add-on enabled in Preferences to have this option.

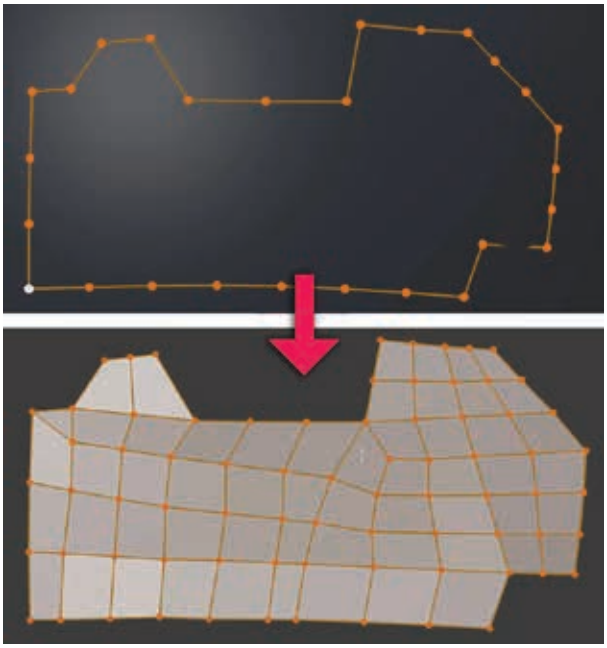


LoopTools > Relax improves the flow of the vertices. **Press and hold Shift+R** to repeat until the edge loops better.



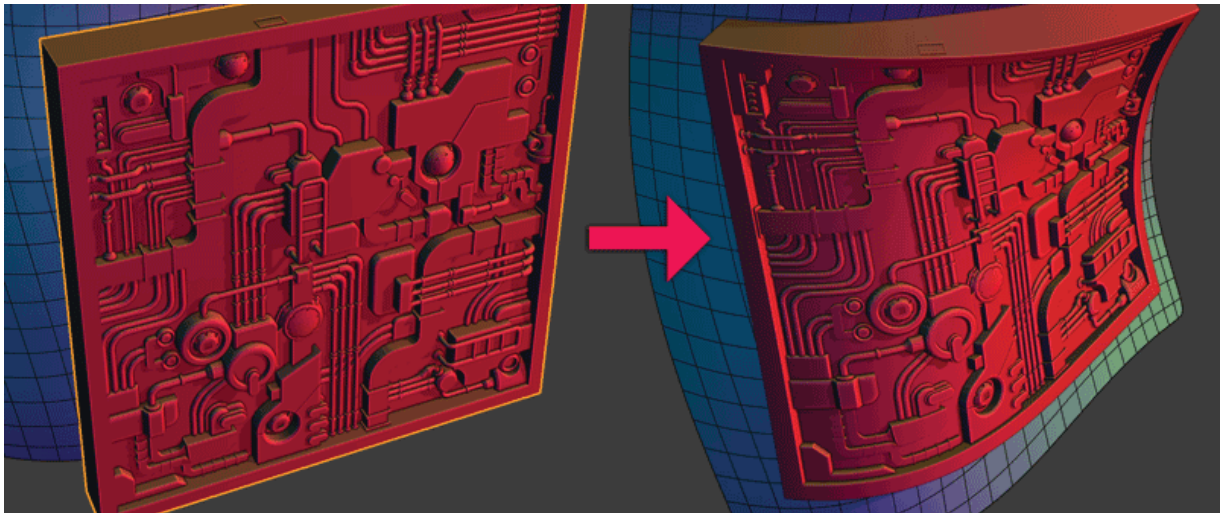
Now you can use this as a nice starting point for retopology.

Check the chapters about retopology for tips about filling the vertex outline with faces, using **F2 add-on** and other methods.

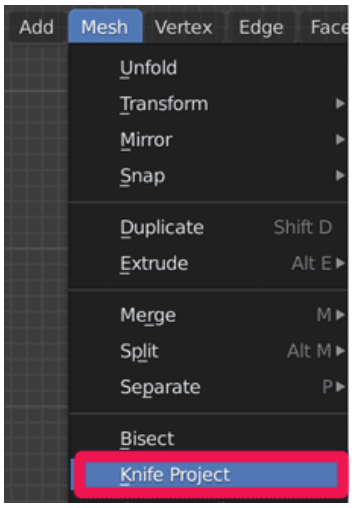


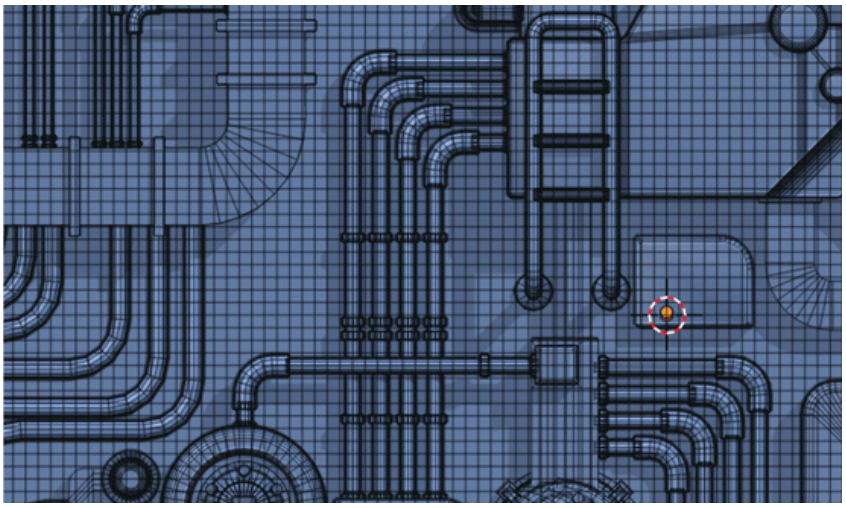
Wrapping hard surface meshes on surfaces

Previously we looked at adding hard surface details (or “greebles”) to a Sphere, and deforming it to a circular object. But what if you have a more complex warped surface and want to add details to that?

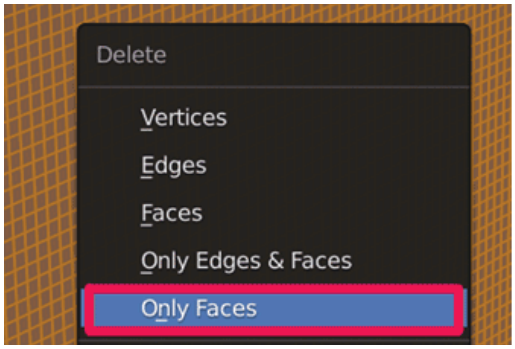


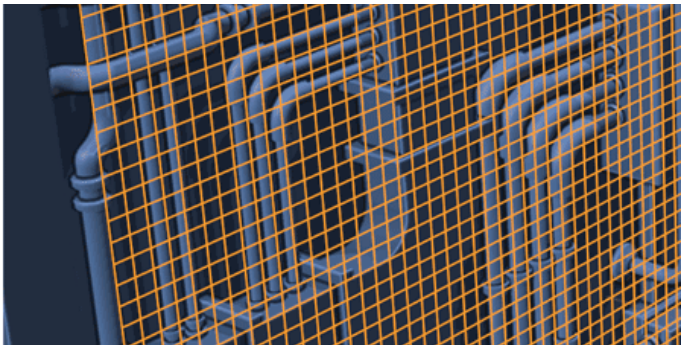
As with the previous topic, we slice up (or dice, as it’s called in the add-on Hard Ops) the mesh using a subdivided plane and **Knife Project**.



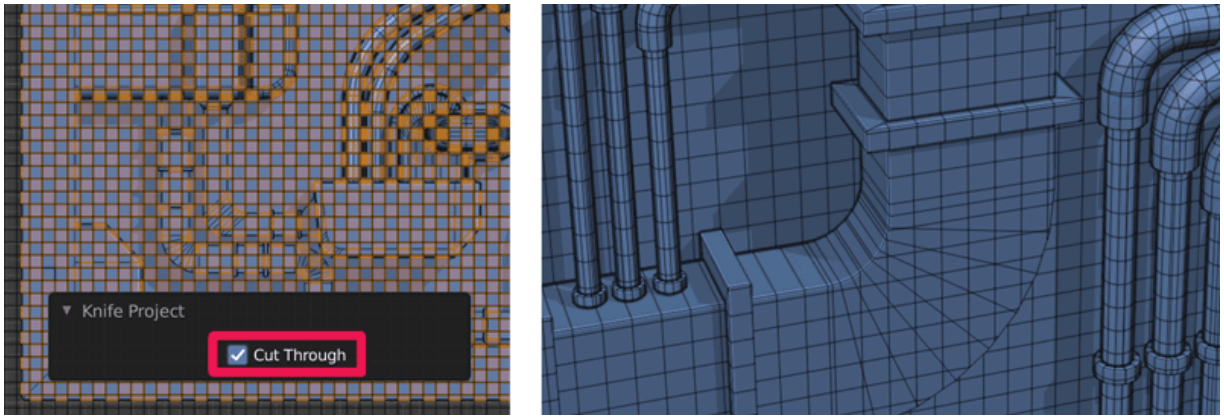


Depending on how much geometry your object has, it might take a minute to process, so save your project before doing this step. If it takes too long or crashes, try again with slightly less subdivisions.

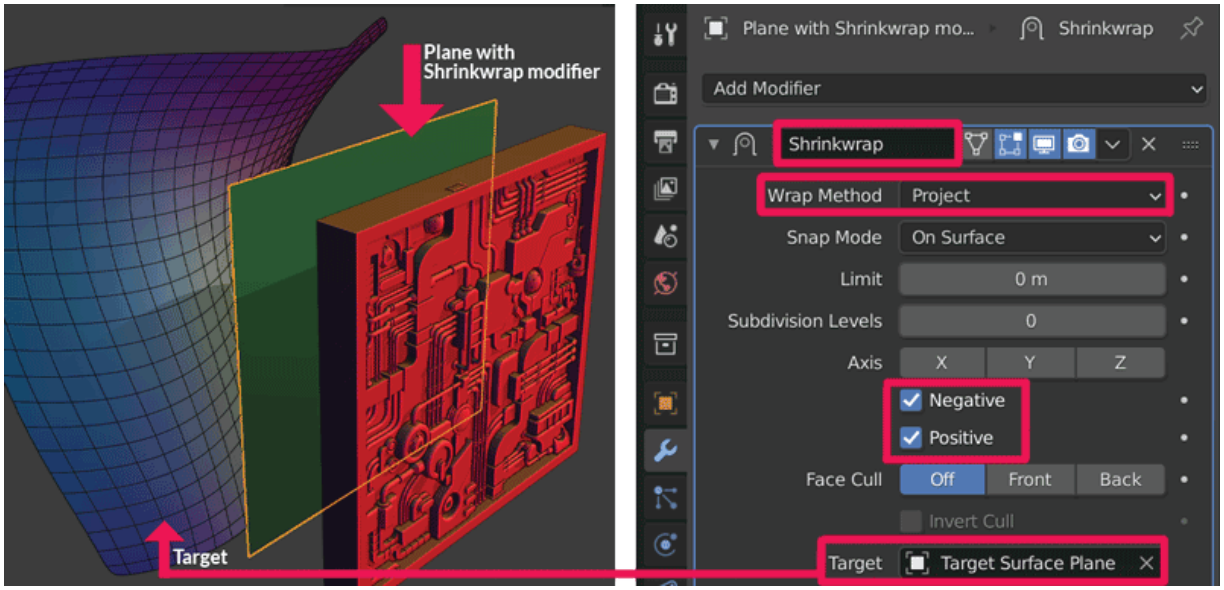




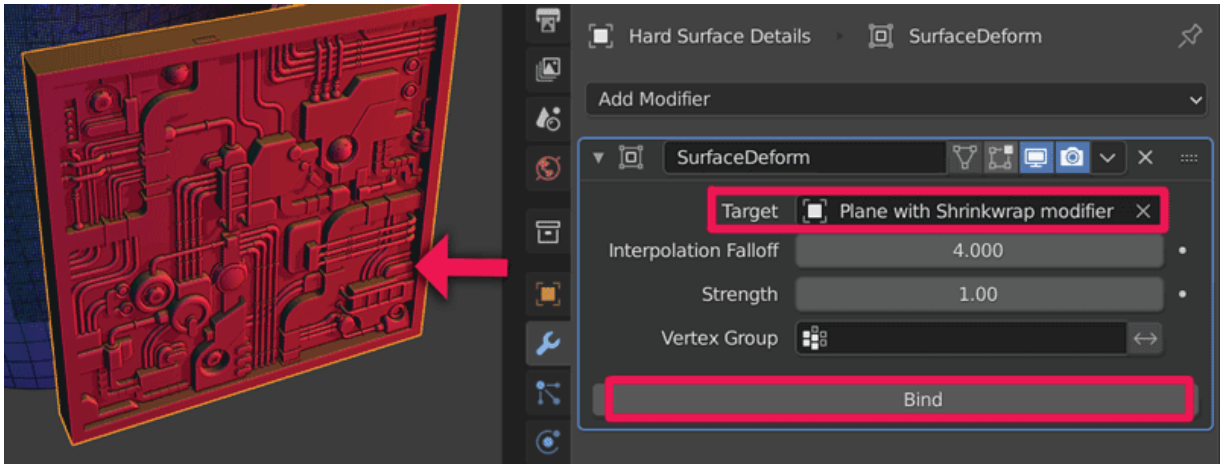
Make sure you check the Knife Project option "**Cut Through**".



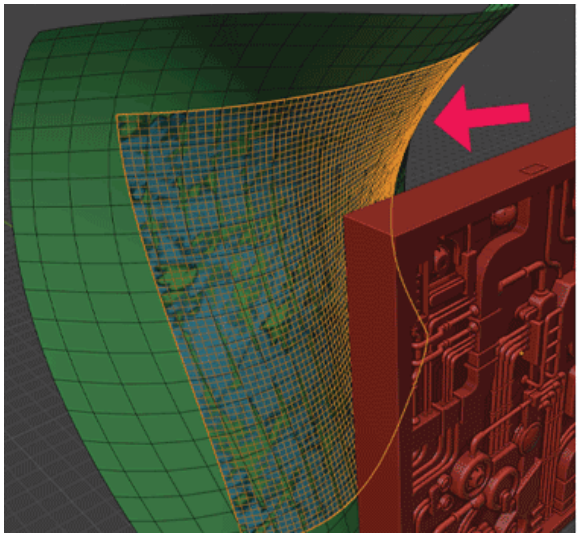
Now with the object sliced, create another **Plane** that's slightly bigger than the hard surface object and subdivide it. You'll add a Shrinkwrap modifier to this later (not yet!) so you may as well name it "Plane with Shrinkwrap modifier" to make it clear later which Plane is which.

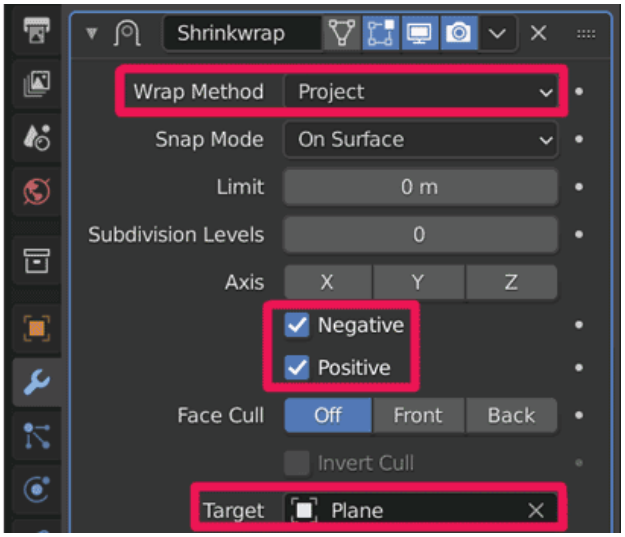


Add a **Surface Deform modifier** to the hard surface object, and set the **Target** to the second Plane (the one named "Plane with Shrinkwrap modifier"). **Click on Bind** (and wait a moment for Blender to do its thing).

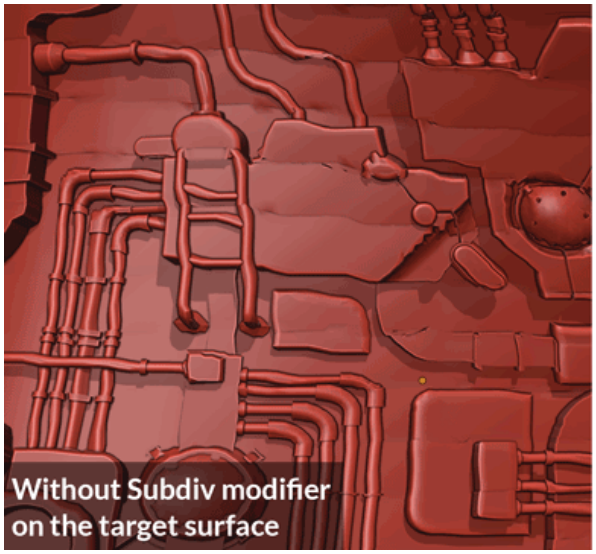


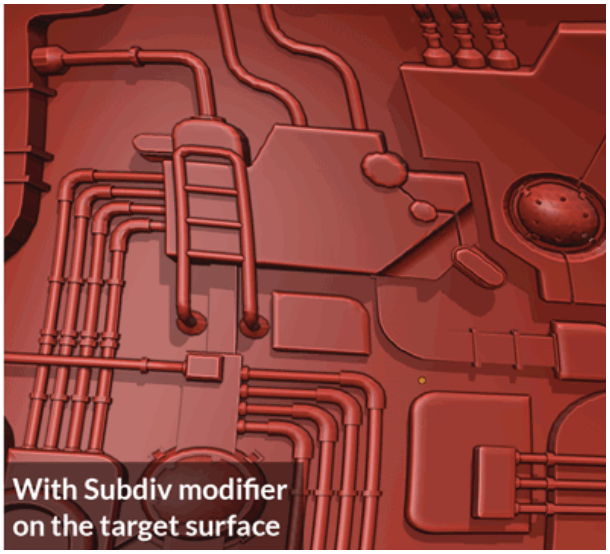
Add a **Shrinkwrap modifier** to this second Plane, and use that to project it onto the surface that you want to project the hard surface details onto.





Now, you'll see that the hard surface details are wrapped around the warped surface. To make it smoother, add a Subdivision surface modifier to the surface that everything is Shrinkwrapped onto.



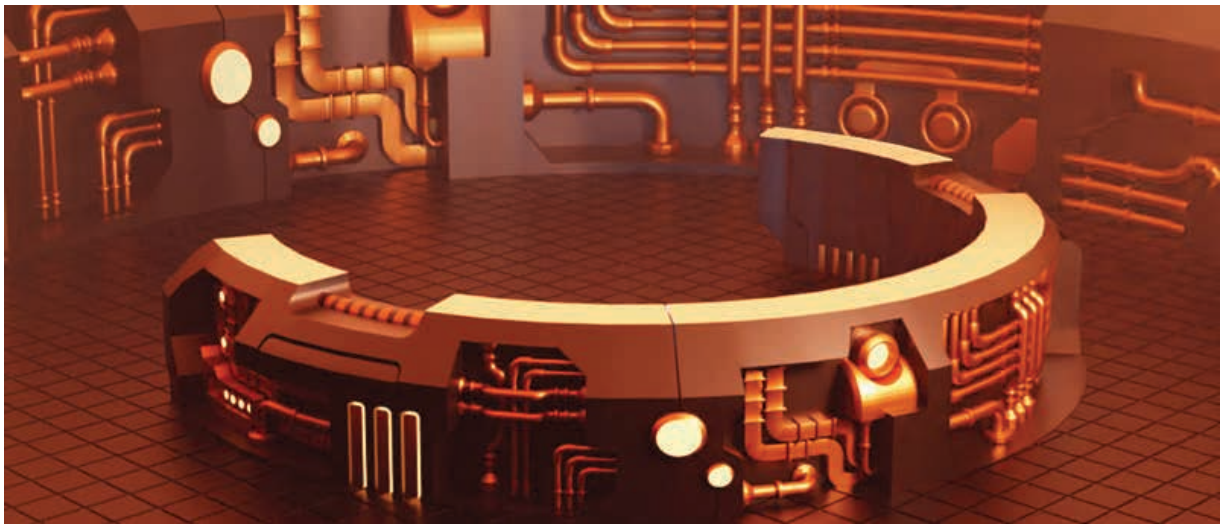


The default settings for the Surface deform modifier work well, however I've noticed that the order in which you do these steps is quite important.

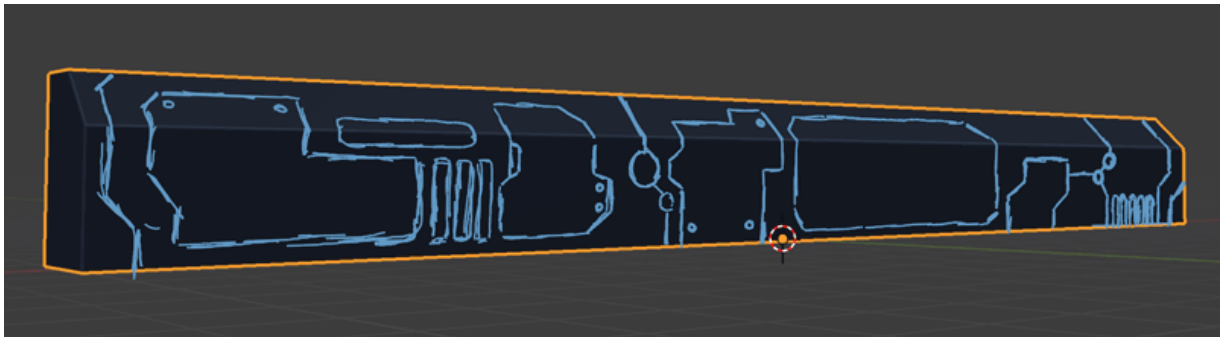
Bind the object to the plane with the Surface Deform modifier **before** adding the Shrinkwrap modifier to the Plane. If you do it the other way around, you may run into trouble.

Modeling Circular Hard Surface Detail

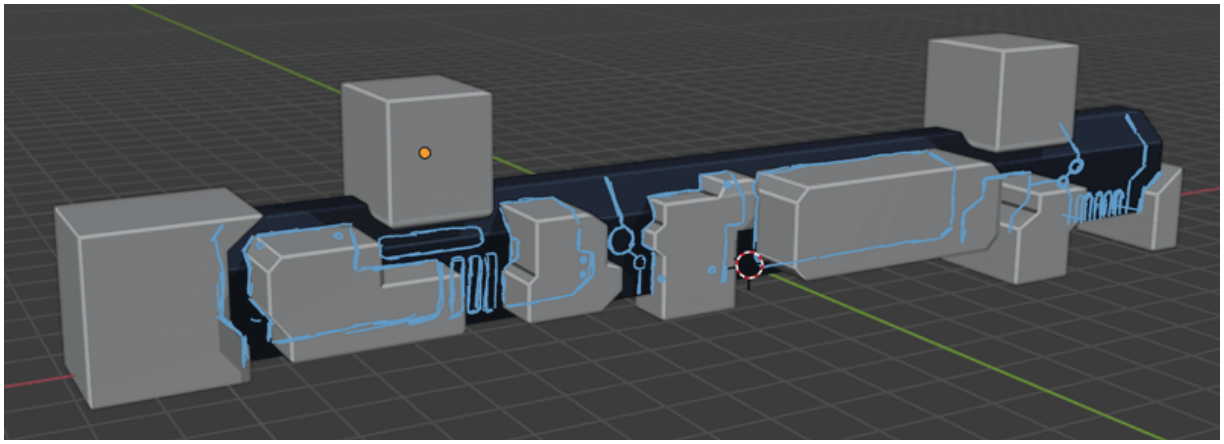
To create circular hard surface details, the easiest way is to start with a straight object that we later bend into a circle.



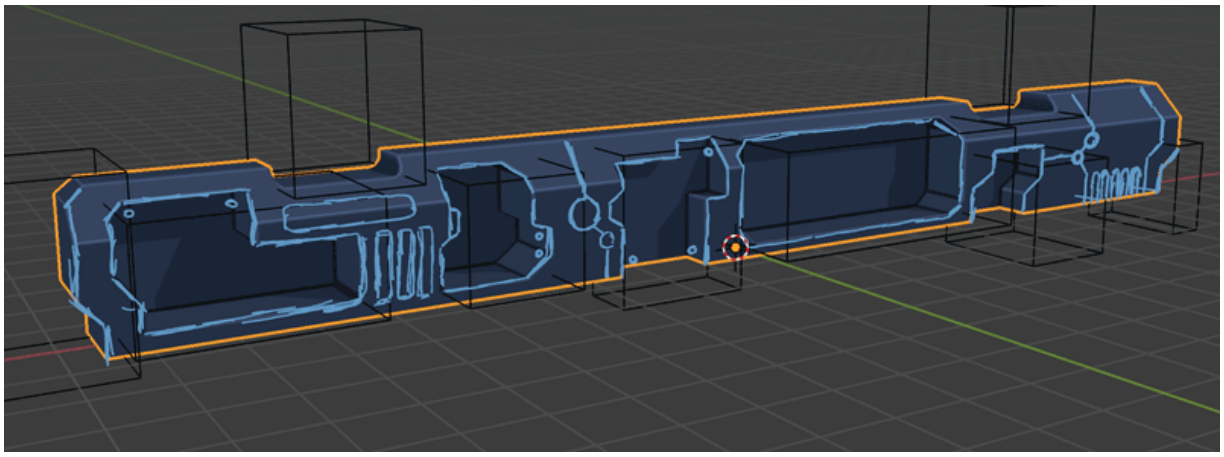
First I model the object just as a straight piece, by scaling a Cube along the X-axis and then drawing on it using the **Annotate tool**, to sketch the idea of the details. As you can see I made the viewport color of the object a bit darker to see the annotation lines better.



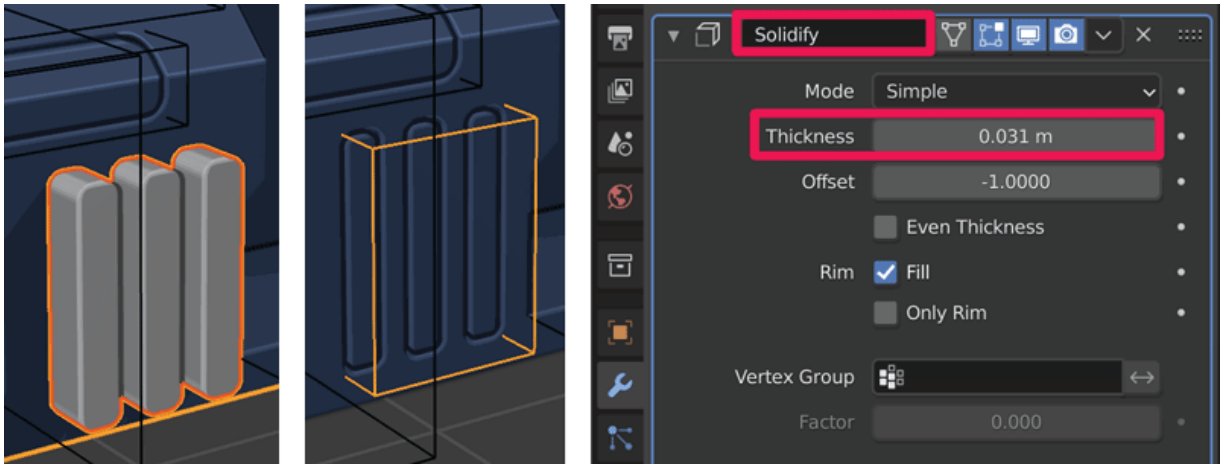
I then add and edit some more cubes, to create cutter shapes that I can use to cut out with a **boolean difference** operation. Of course I could skip the annotation sketch step, but I personally find it a bit stressful to start with actual geometry right away. Splitting things up into smaller steps always makes it easier and more relaxed.



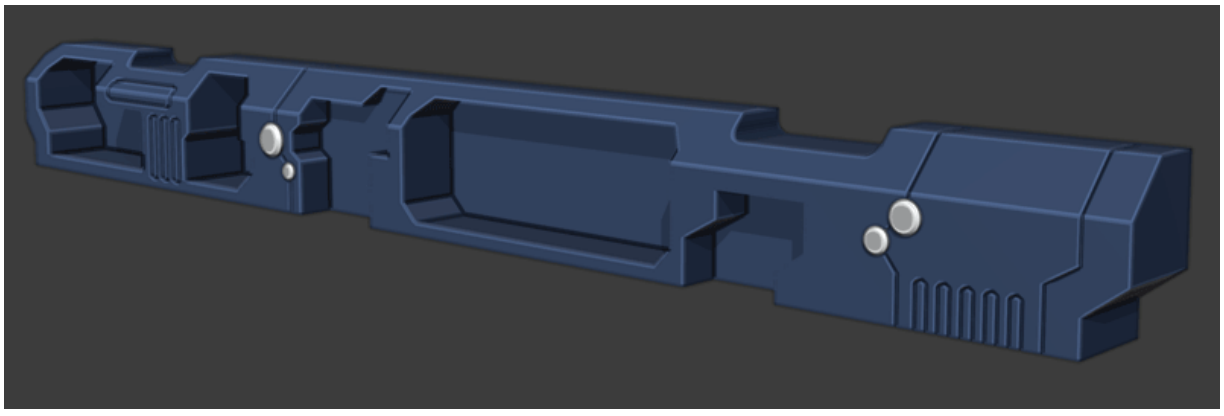
Using the **BoolTool add-on** I can just easily cut those shapes out by selecting the cutter and the main object and **pressing Ctrl and - on the numpad**.



There are some lines that are meant as slices, I do those separately to keep things clear and easy. Creating slices is pretty much the same as difference cuts, except you need to add a **Solidify modifier** to the cutter object. The **Thickness** value on the **Solidify modifier** then controls the thickness of the slice cut.

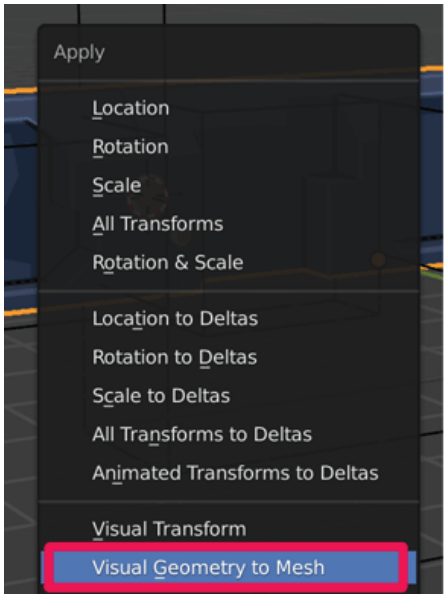


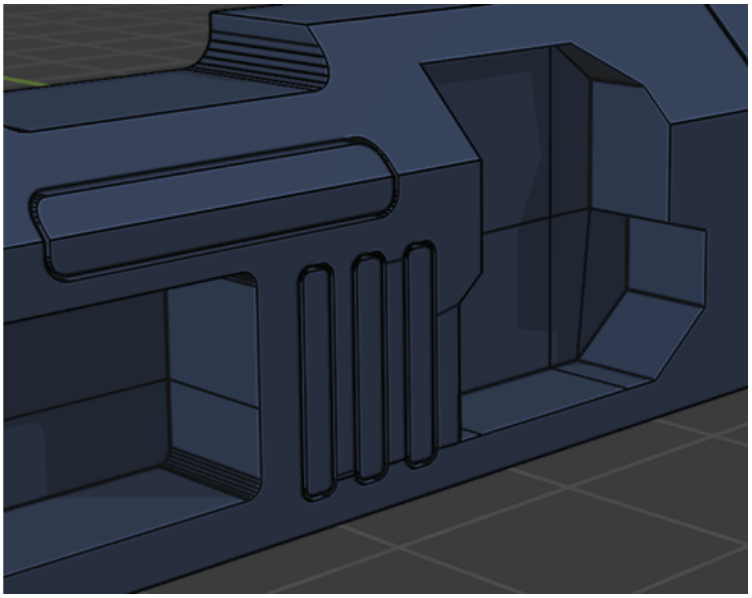
This is the result with all the cuts and slices, as well as some beveled cylinders added as separate objects.



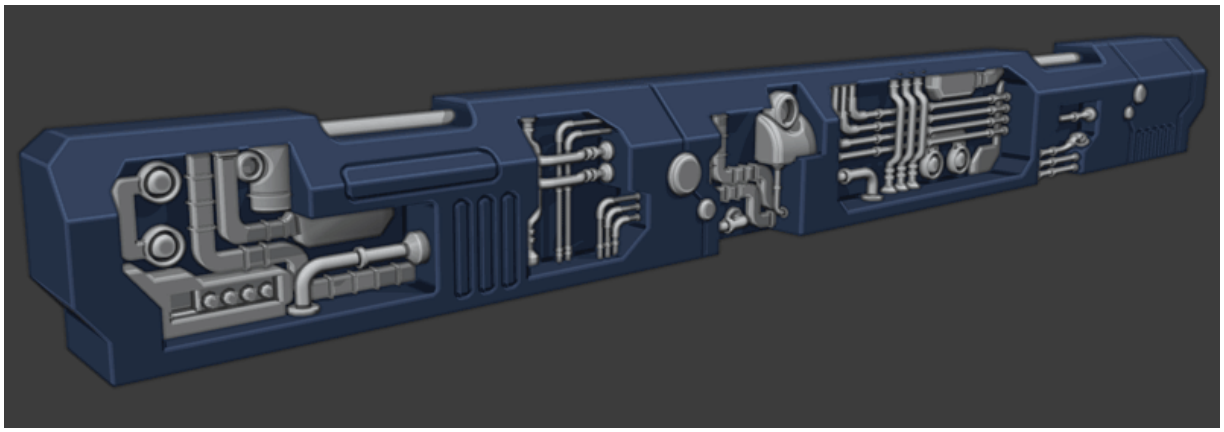
Although it's nice to have a non-destructive workflow and we can still adjust the depth of these boolean cutters if we want, at some point it starts to bog down the computer and it's time to apply all the boolean modifiers.

To do this simply select the main object, **press Ctrl+A** and choose **Visual Geometry to Mesh**. Now it's just a mesh object, with all the cuts applied as geometry.

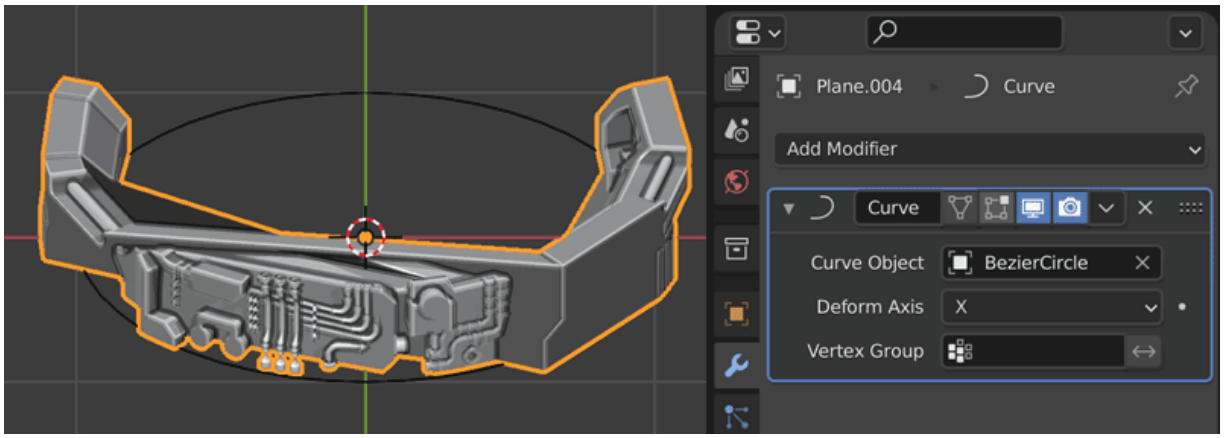




Of course we can keep adding more detail, like greebles and pipes inside of the cut out parts.

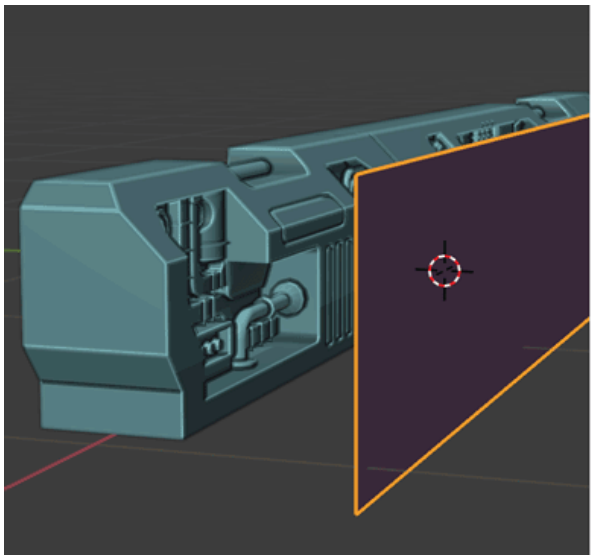


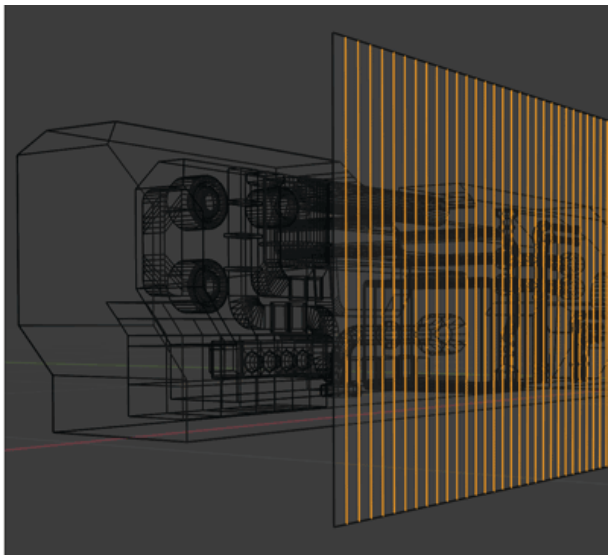
With this done, let's look at how we can bend this into a circle. The obvious method would be to add a Bezier Circle by **pressing Shift+A > Curve > Circle**. Then add a **Curve modifier** to the object and use the **BezierCircle** as the **Object**. However, as you can see the object doesn't bend very well at all.



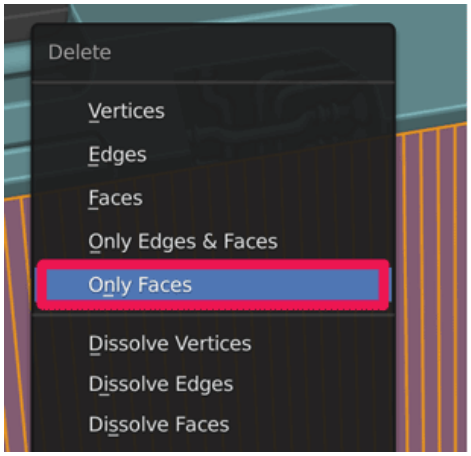
The reason for this is that the object simply doesn't have enough geometry. It can only bend where there's vertical edges. So, the solution is to **add more vertical cuts**. But how?

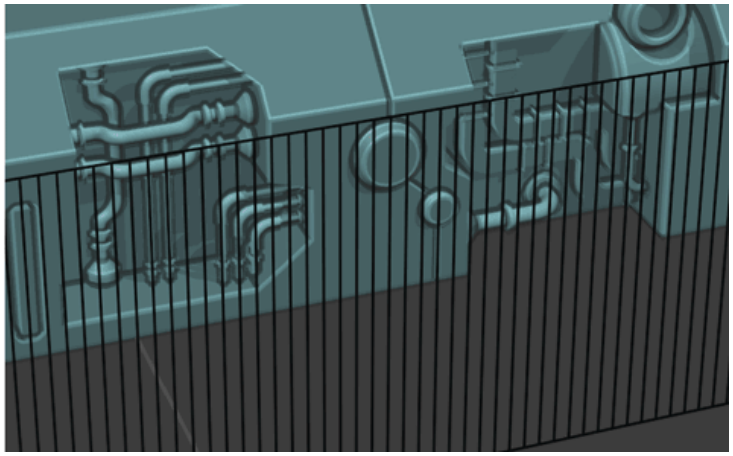
First, add a **Plane** by p**ressing Shift+A > Mesh > Plane**. Scale it in Front orthographic view to be bigger than the object. Place it slightly in front of the object. Then in Edit Mode, add lots of vertical cuts.





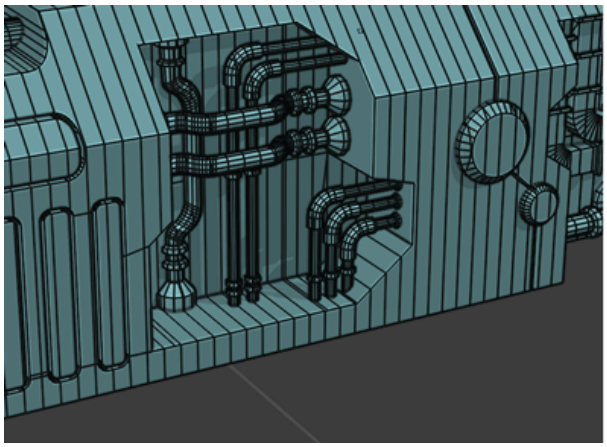
Now in Edit Mode, select all the Faces of the Plane and **press X**, then choose **Only Faces**. This leaves you with a Plane consisting only of edges.

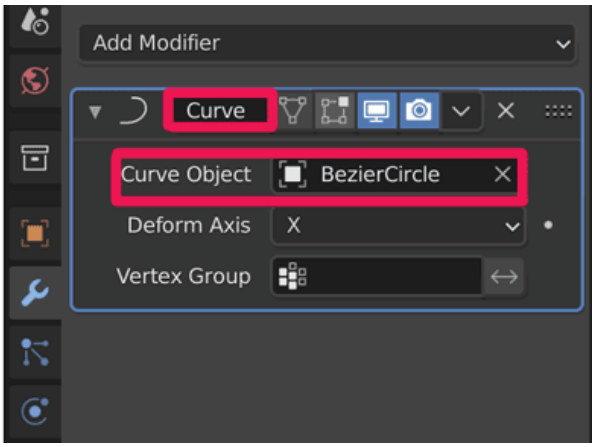




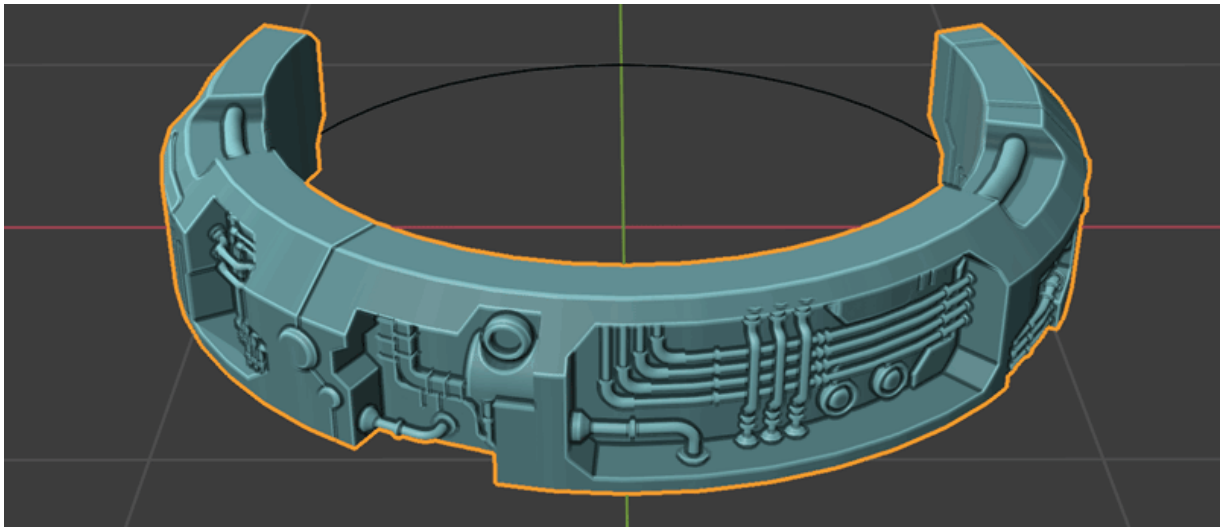
In Object Mode, select only the hard surface object. Go to Front orthographic view. **Press Tab** to enter Edit Mode and **while holding Ctrl, left-click** on the Plane. Go to **Mesh > Knife Project**. If it's difficult to select the Plane in the viewport, you can also do that in the Outliner.

You can hide (or delete) the Plane now. In Edit Mode, you can see that the object has been sliced vertically.





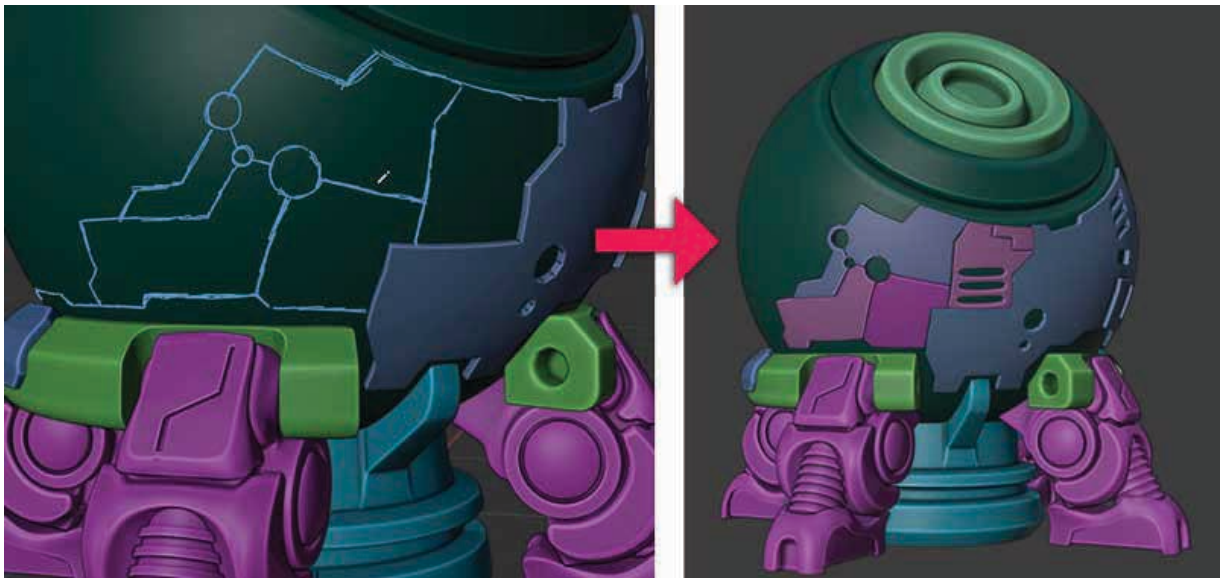
Now you can try the Curve modifier again. In this case, it works! You may need to change the **Deform Axis** on the Curve modifier, or rotate the Curve itself to get the result you want.



Modeling Sci-fi Greebles on a Sphere

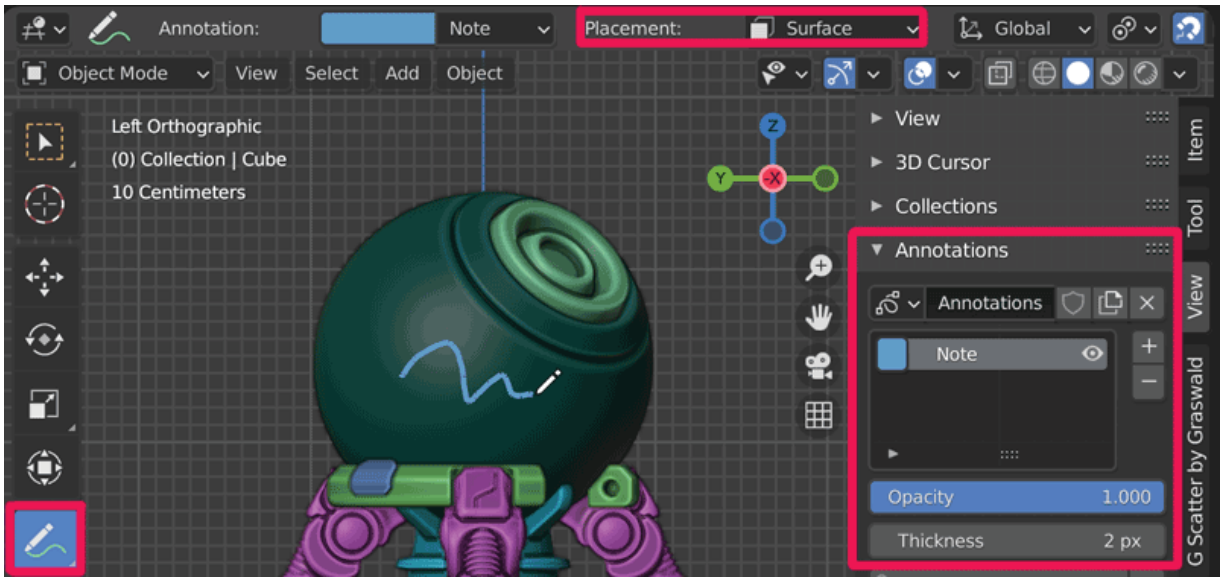
If you want to model sci-fi details (or "greebles") on a spherical object, it can be useful to design them first using the **Annotate tool**.

The benefit of this workflow, is that you can think purely about the design first in a way that's easy to try, erase and try again without having to go through the time-consuming modeling process. Then, once you're happy with the design you can focus purely on modeling.

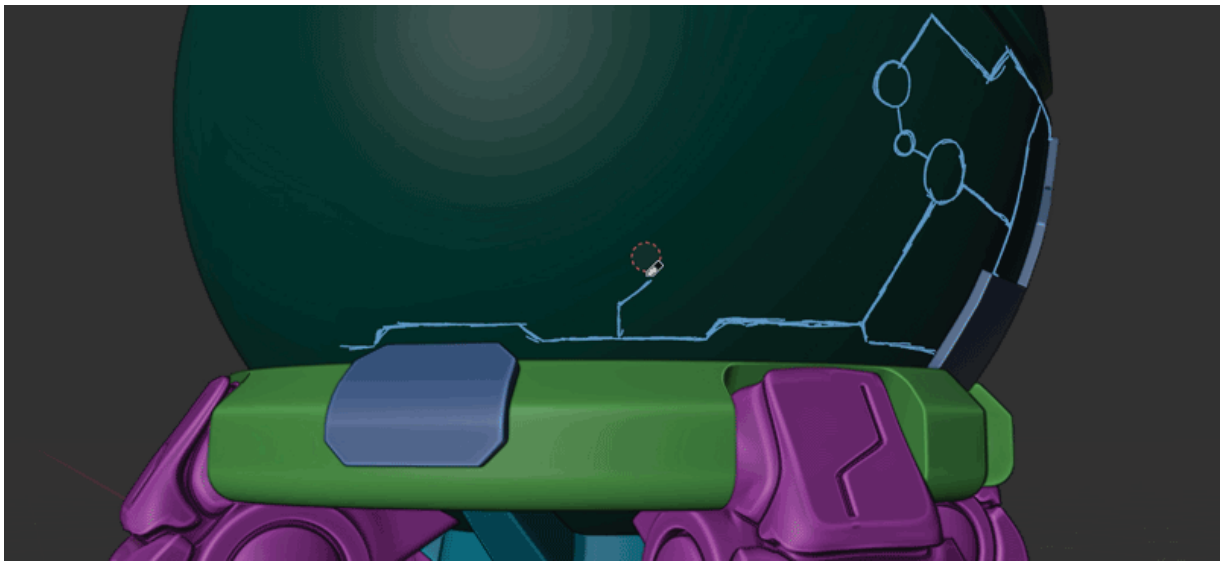


To start, enable the **Annotate** tool by clicking on it in the Tool panel. Then, set **Stroke Placement to Surface** so the strokes will be on top of whatever object you're drawing on.

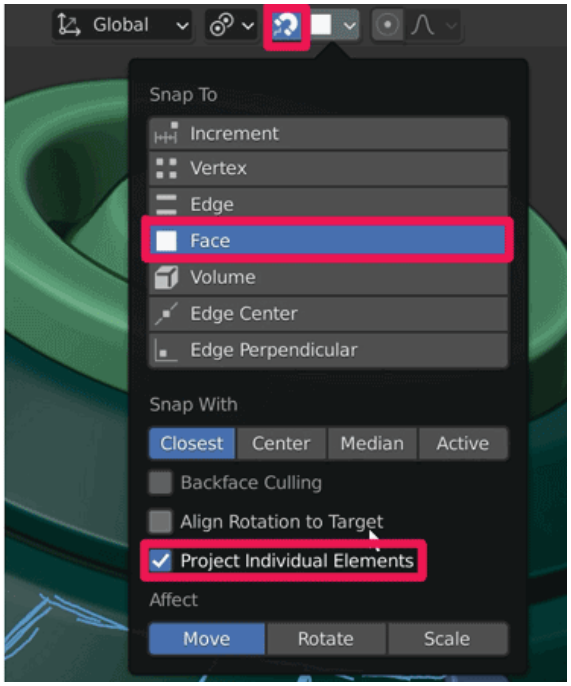
In the Option Panel in the View tab under Annotations, you can hide the annotation by clicking on the eye icon, or lower the **Opacity**.



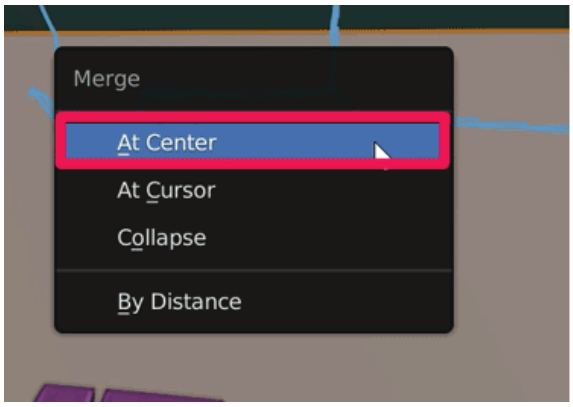
Hold Ctrl to erase the annotation strokes. The keyboard shortcut for using the Annotate tool is **holding D** and drawing a stroke with the left mouse button. To erase, **hold D and draw with the right mouse button**. To use stroke stabilization, **hold Shift**.

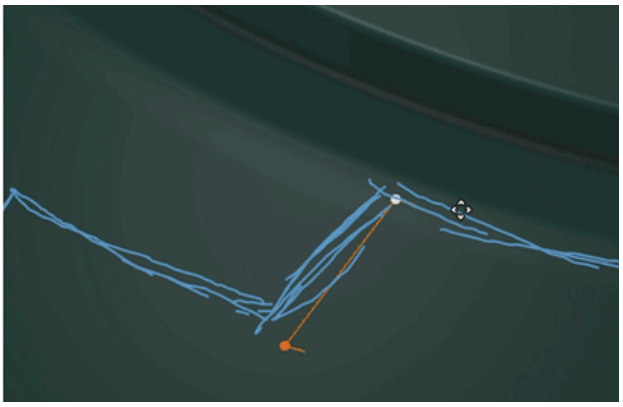


I recommend doing the design in parts instead of everything at once. Once the design is drawn with the Annotate tool, we can start with the modeling part. First, turn on **Snapping** with **snap to Face** and **Project Individual Elements** enabled.

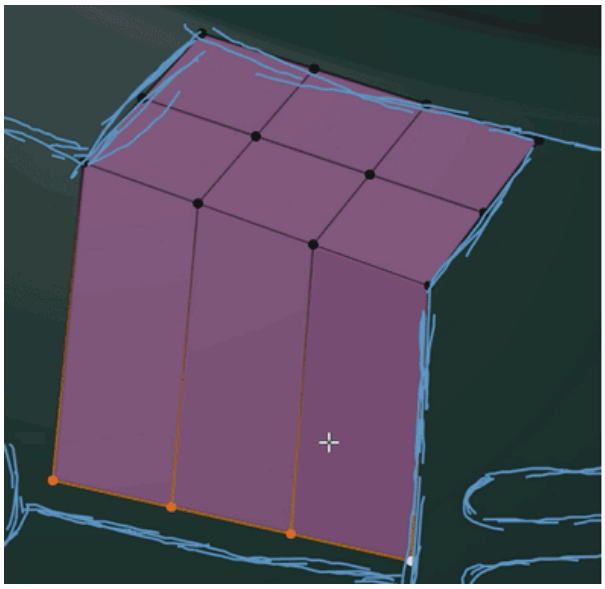


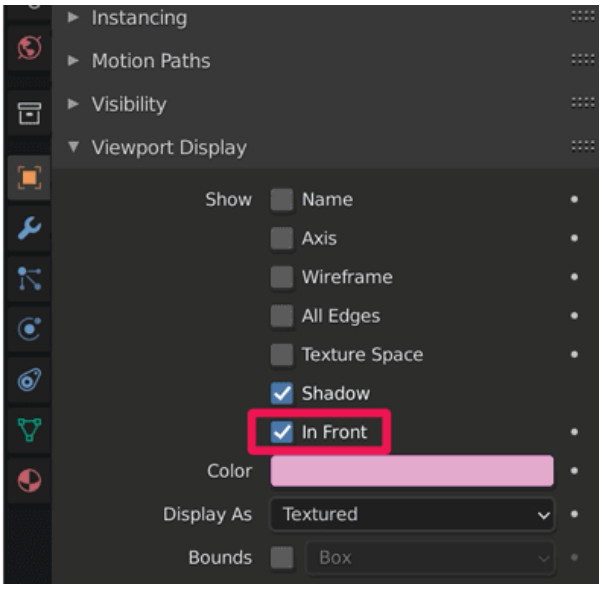
Add any primitive object like a Cube (by pressing Shift+A and choosing Mesh > Cube) and in Edit Mode switch to Vertex Select and **press M**, then choose **At Center**. Now you've got one vertex that you can move on the surface of the object to where you want to start modeling. **Press E** to extrude the vertex to create the first edge.



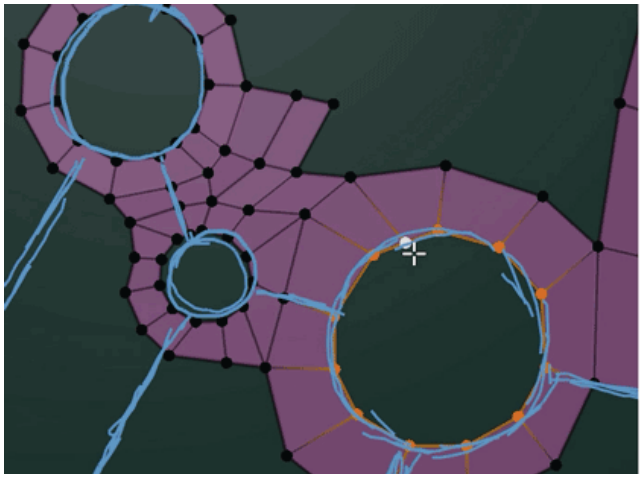


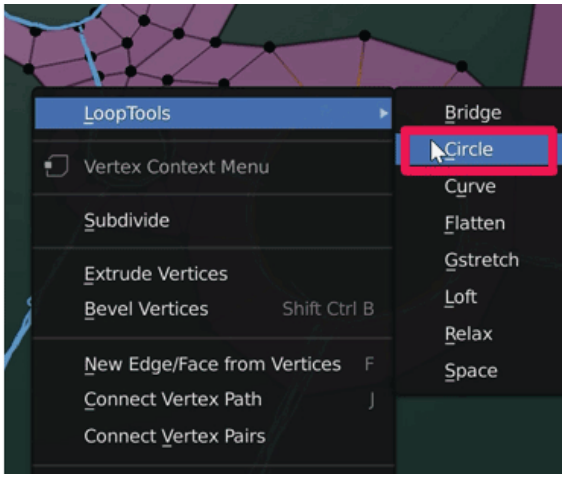
Now the problem is that you can't really see the new geometry well, as it's mostly hidden by the other object. This is easily solved by checking **In Front**. While you're there, you may as well change the viewport display **Color**. I find it pleasant to have everything in a different color as opposed to just grey.



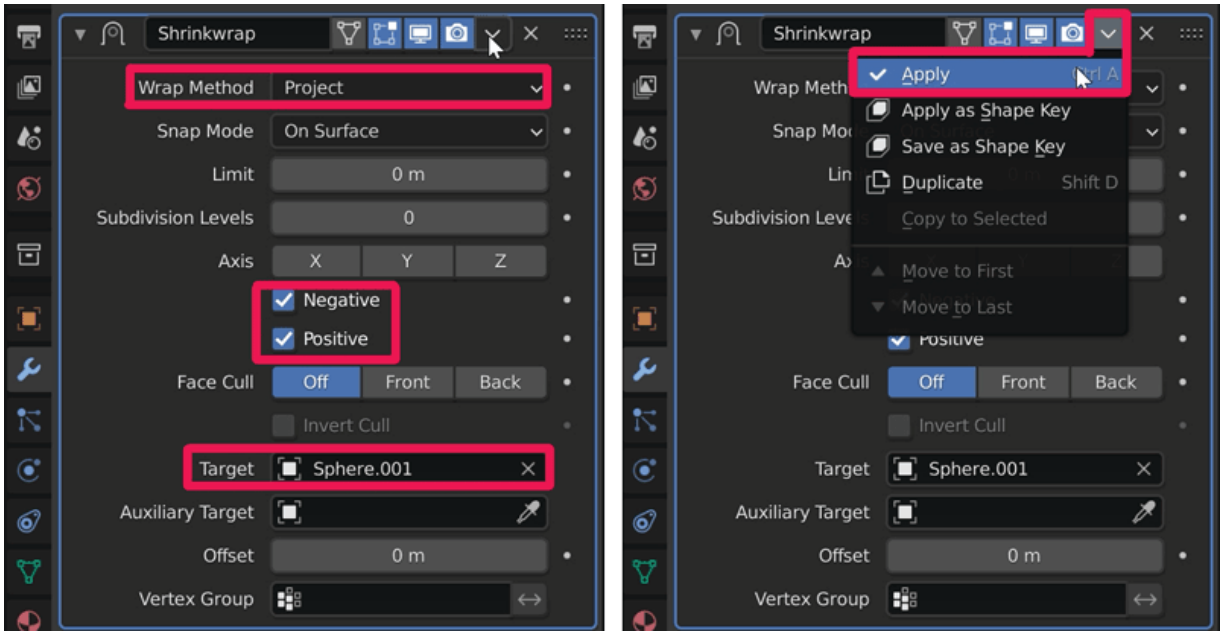


After this it's just a matter of extruding edges, adding edge loops and merging vertices as described in more detail in the chapters about retopology. If you model circular holes, you can model them roughly by hand first and then use the **LoopTools > Circle** function to make them perfectly circular (you'll need to enable LoopTools in Preferences > Add-ons for this).



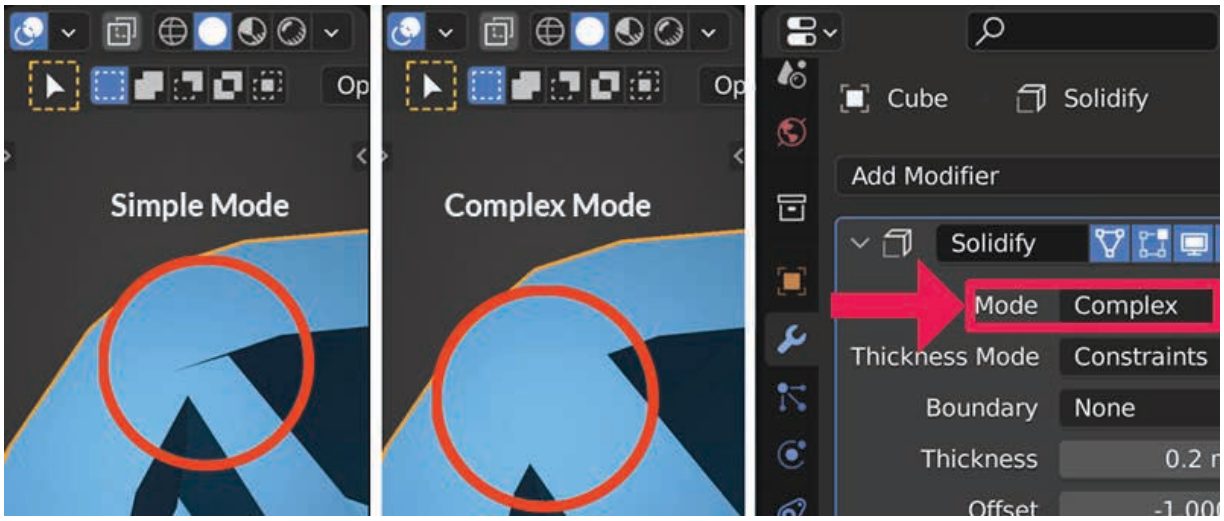


After you finish modeling the geometry, to make sure it really snaps perfectly to the surface below it, add a **Shrinkwrap modifier** to it and set it to **Wrap Method: Project**. Then check both **Negative and Positive** and as the **Target**, choose the Sphere it's snapping to. Then, you can **apply** the Shrinkwrap modifier.

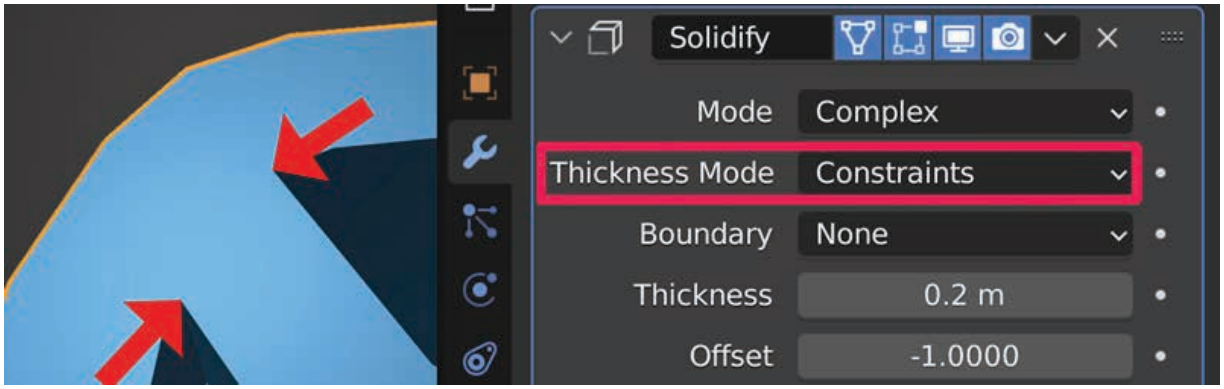


Complex Mode for Solidify Modifier

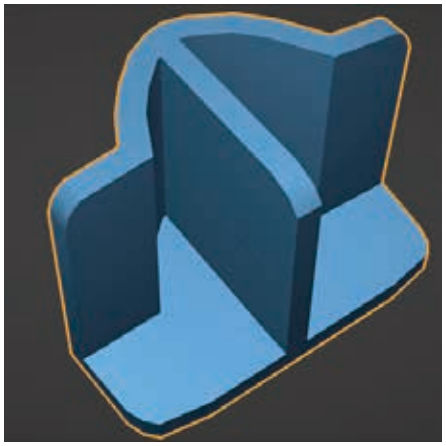
The Thickness modifier now has a new mode called “Complex Mode”. It allows you to give more complex meshes thickness where Simple mode would fail. To activate it, simply switch from Simple to Complex mode in the modifier.



Thickness Mode: **Constraints** seem to give the most even thickness.

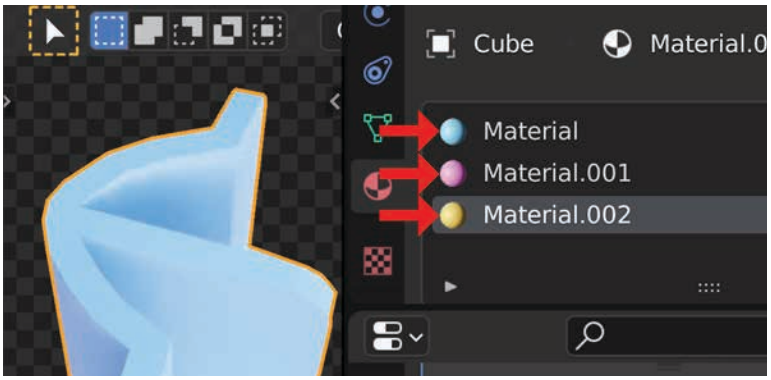


This is great news if you have an architectural plan and want to create walls for example.

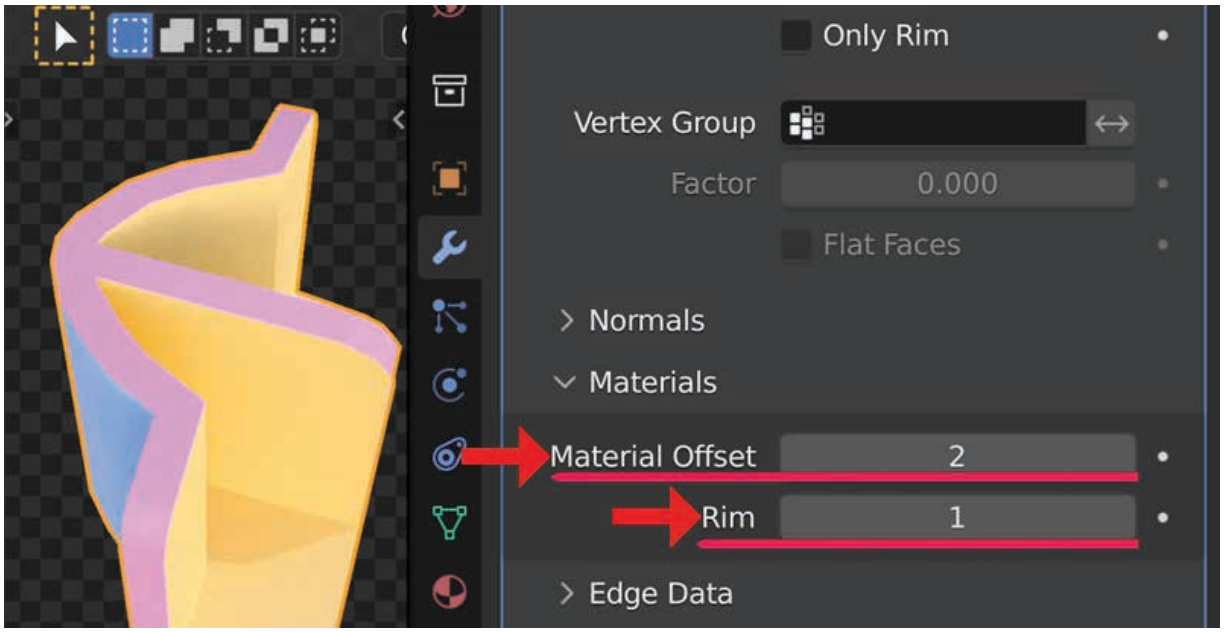


Click here to return to the index.

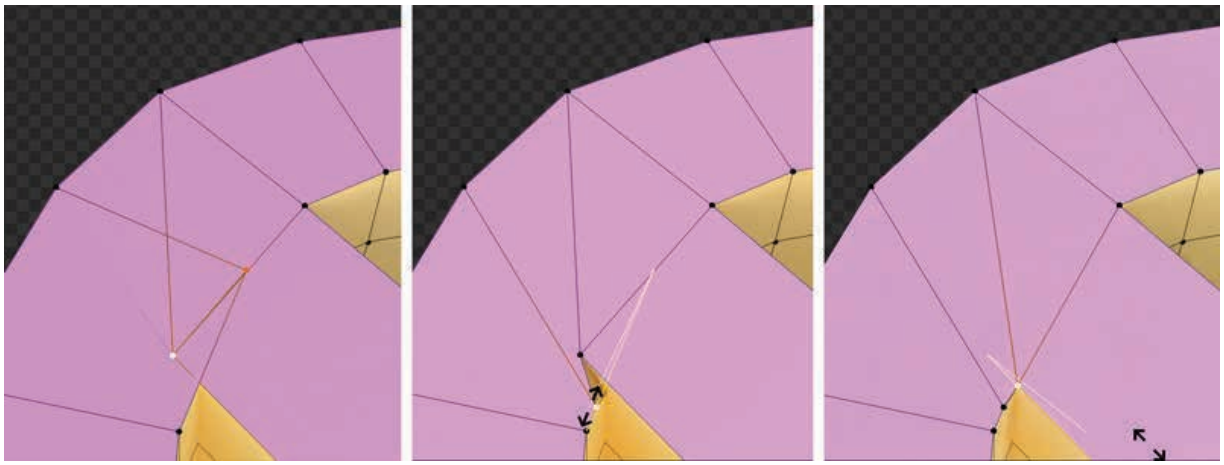
If your object has several materials, you can use the material Index Offset to create a separate material for the newly created faces.



Changing The **Index Offset** value changes the colour of the faces generated by the modifier. Changing the **Rim** value, changes the colour of the faces in between both sides.

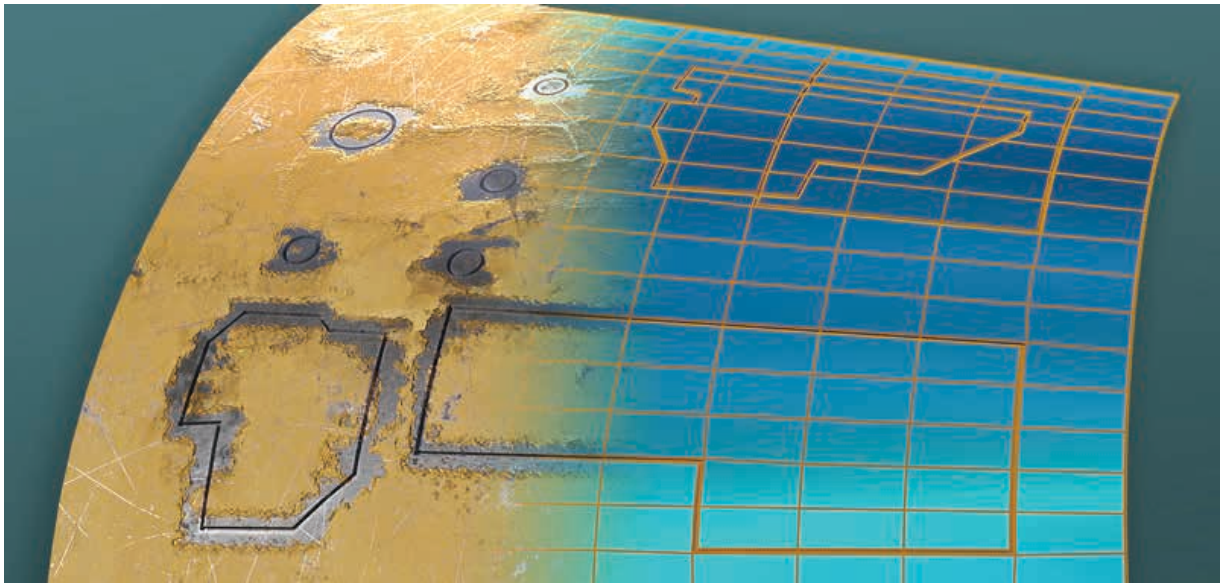


The material for the new faces seems to be ignored in Complex mode in my experience. This new Complex Mode is also good news for 3D printing work although you would have to apply the modifier and fix any weird geometry before exporting your model.

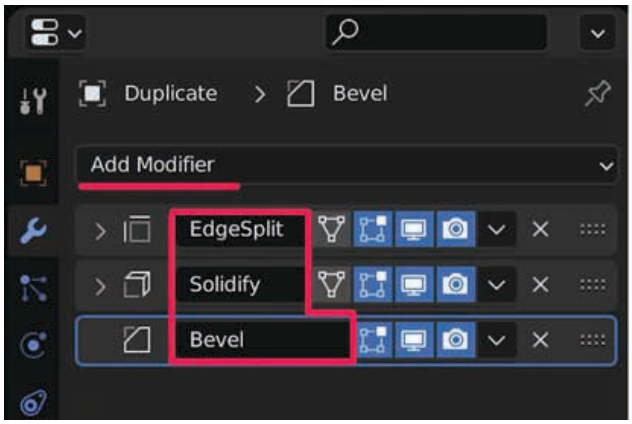


Easy Non-Destructive Panel Cuts

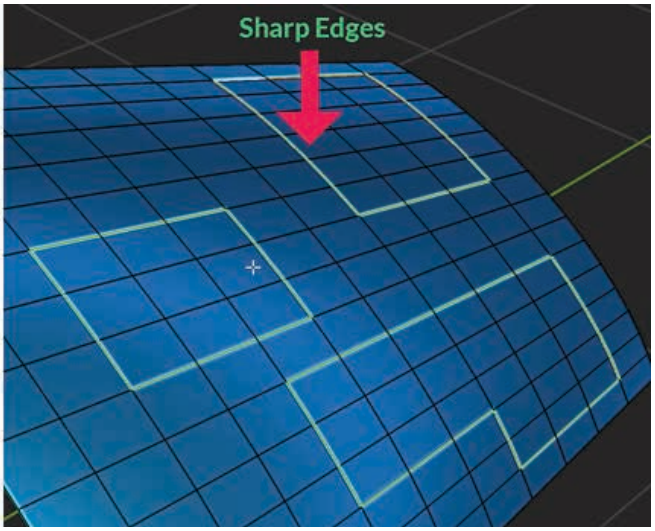
This method can be useful if you plan to bake the panel cuts to a Normal map.



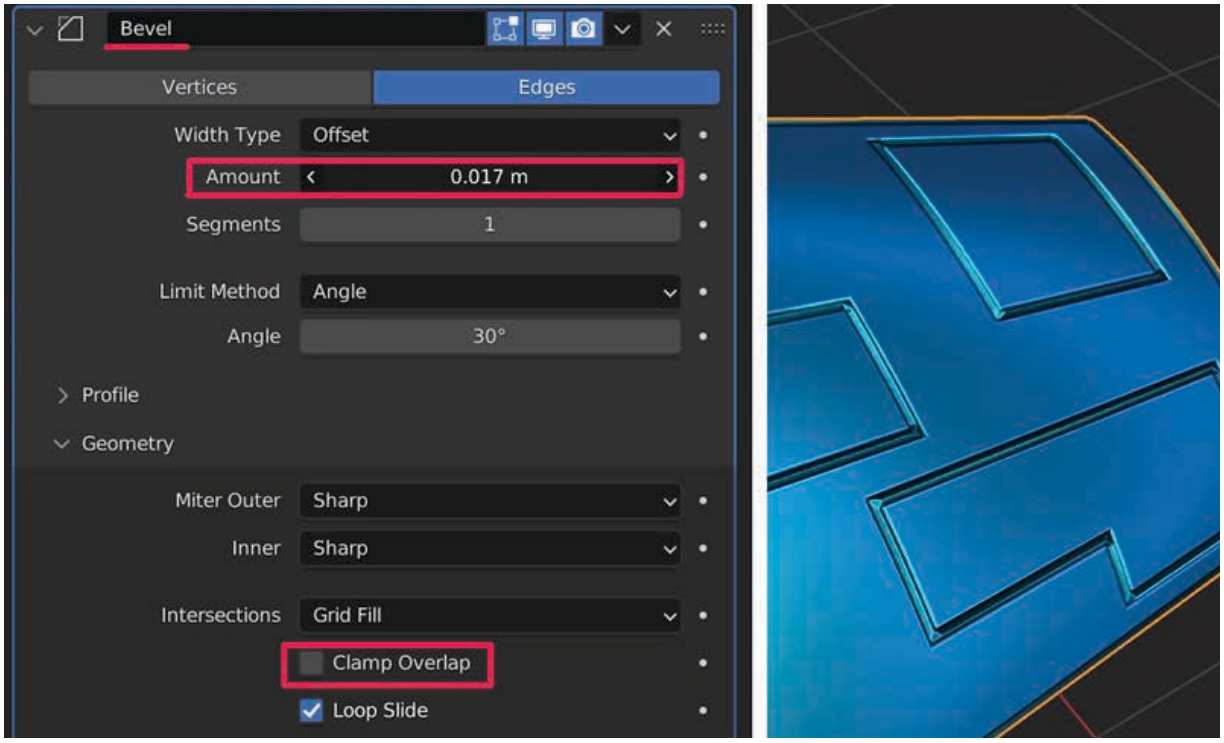
Add an **EdgeSplit** modifier, a **Solidify** modifier and a **Bevel** modifier to your object. Right-click and choose **Shade Auto Smooth**.



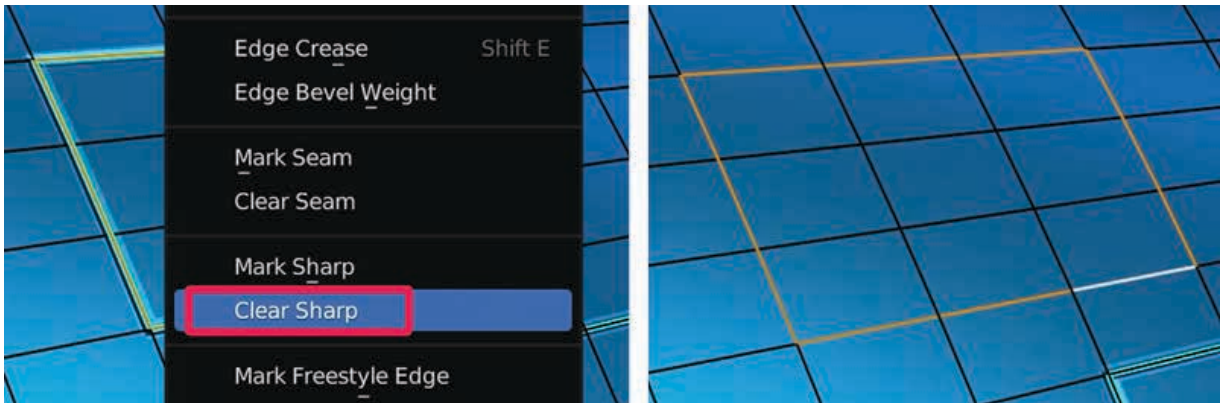
Select some edges in Edit Mode, right-click and choose **Mark Sharp**. These Edges will be turned into panel cuts by the modifiers.



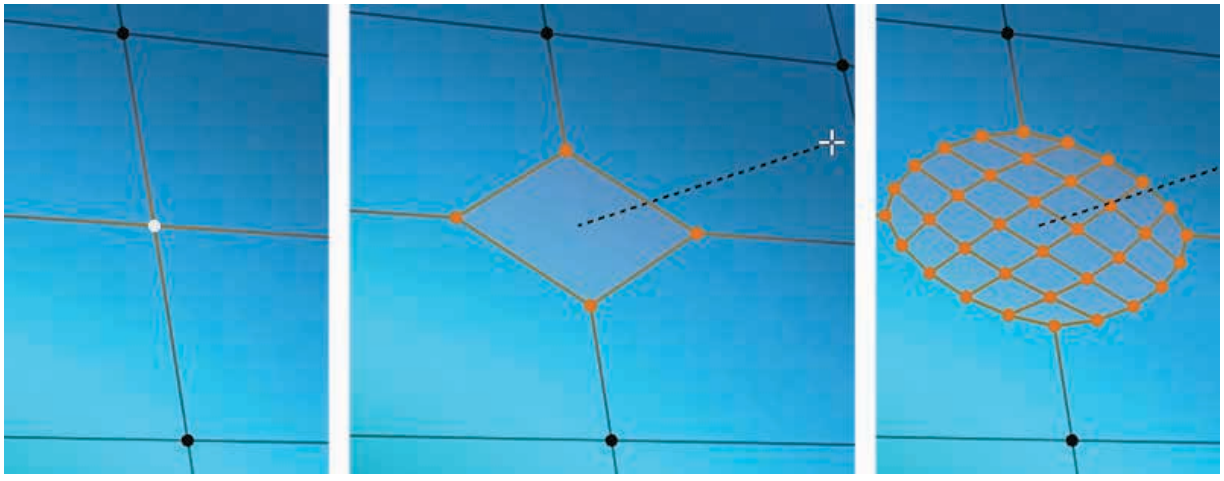
Turn off **Clamp Overlap** in the Bevel modifier to be able to increase the Bevel **Amount**.



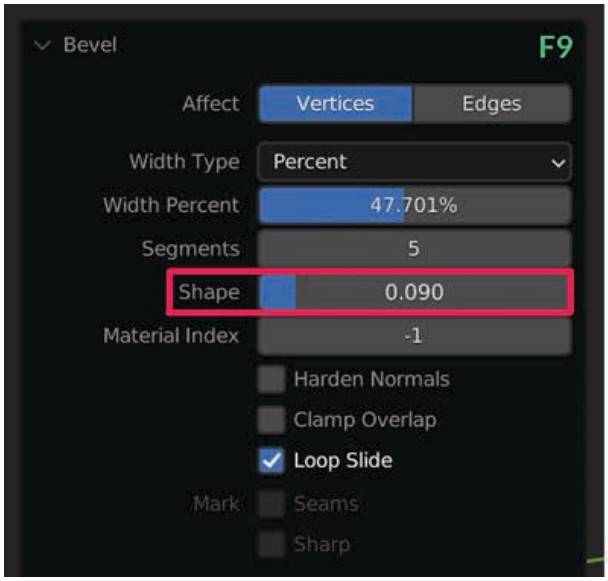
To remove the panel cuts, simply select those edges in Edit Mode, right-click and choose **Clear Charp**.

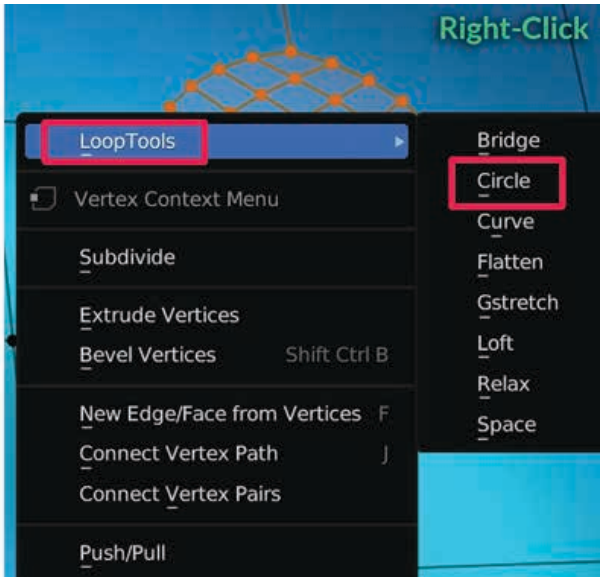


To make a circular panel cut, simply select and Bevel a Vertex with **Shift+Ctrl+B**. Scroll the mouse wheel up to increase the vertex count. Don't worry if the shape isn't circular.

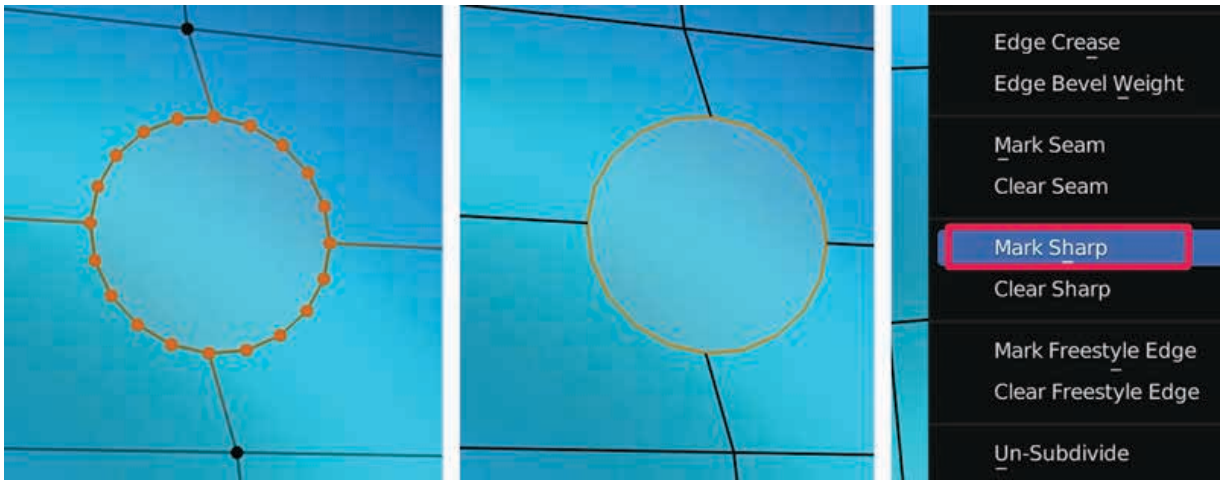


Then adjust the **Shape** value to 0.09 in the Bevel options that pop up (if it's gone, press F9), or right-click and use **LoopTools > Circle** to make it circular.





Press F to fill the selection. Switch to Edge selection, right click and **Mark Sharp**.

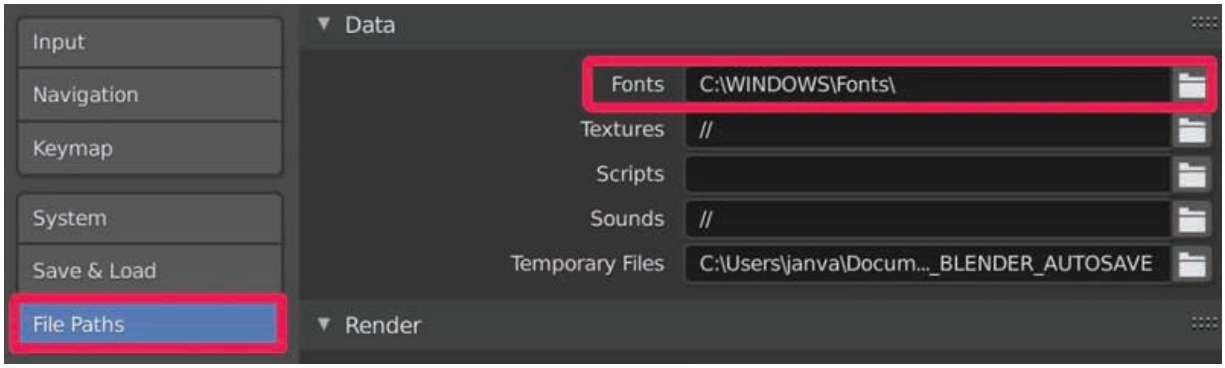


Before the N-gon police shows up, quickly **press K** and make some Knife cuts to create an interesting shape. And again right-click and Mark Sharp.

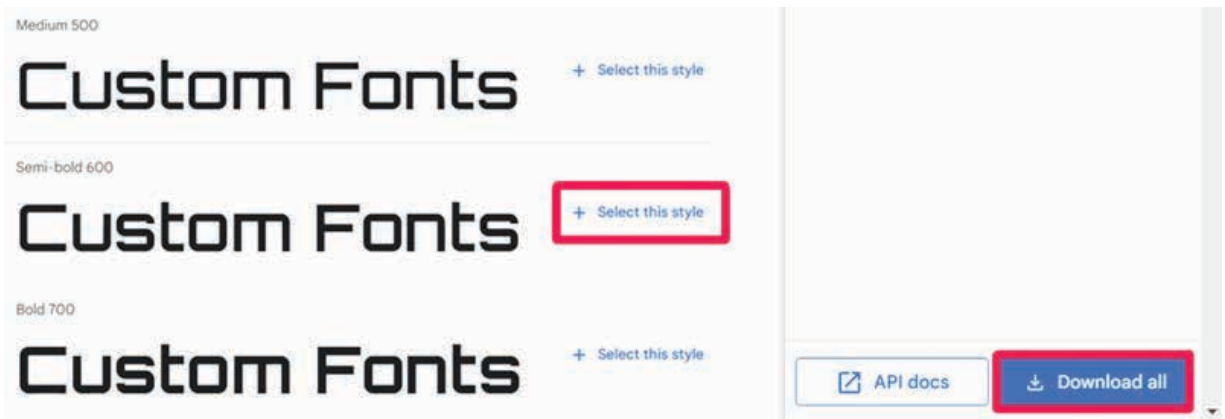
Custom Fonts in Blender

Using the default font in Blender for your 3D text is kind of lazy... the bad kind of lazy! What you want is to get an interesting font that fits in the context of what you're making. Luckily, you can get hundreds of free fonts and use them in Blender.

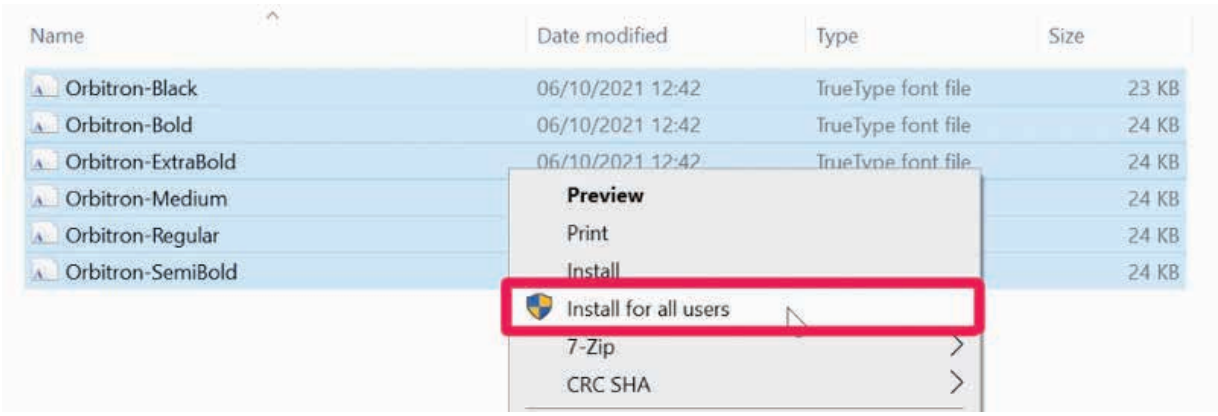
In Preferences, under File Paths verify that the Fonts path is set to where you keep your fonts. On Windows, by default this is C:\WINDOWS\Fonts\.



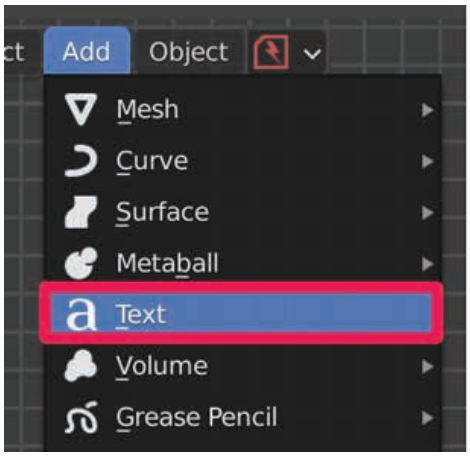
The website https://fonts.google.com has a lot of nice and free fonts that you can use. Once you've found a font you like, click on "**+ Select this style**". Then click on **Download all.** Alternatively you can click on **Download family**.

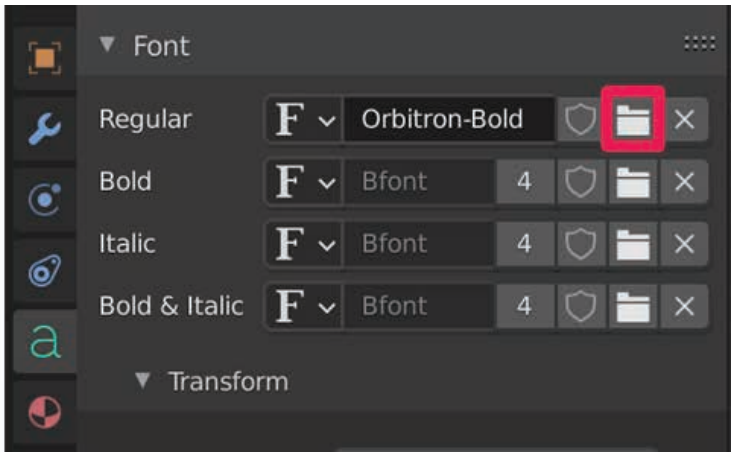


Extract the downloaded ZIP file. Select all the fonts in the unpacked folder, **right-click** and choose **Install for all users**. *This is very important - if you click **Install** instead of **Install for all users**, it will **NOT** go to C:\WINDOWS\Fonts\ and Blender can't find it.*



Now add some text in Blender by going to **Add > Text** and choose your new font in the Font tab.

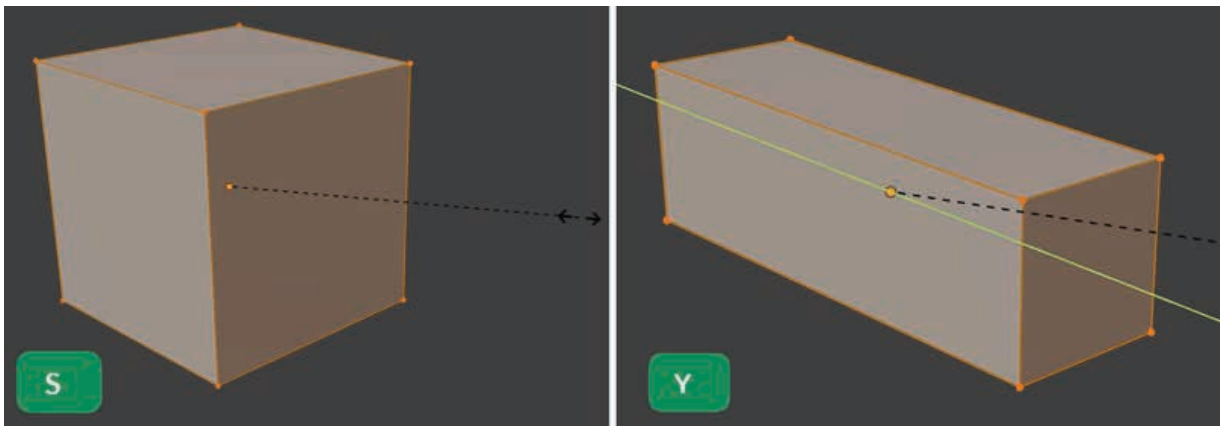




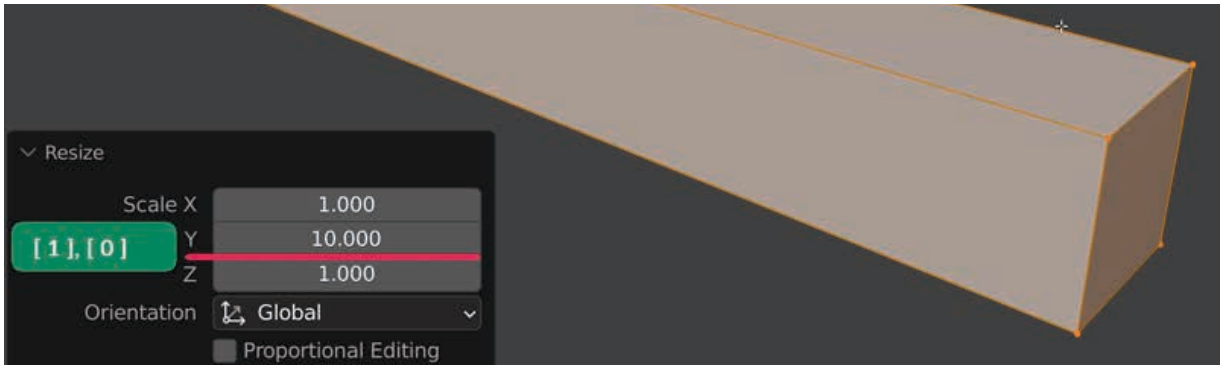
Give it some thickness and a cool material, and you've got a cool title screen for your sci-fi movie, for example!

Non-destructive bridge

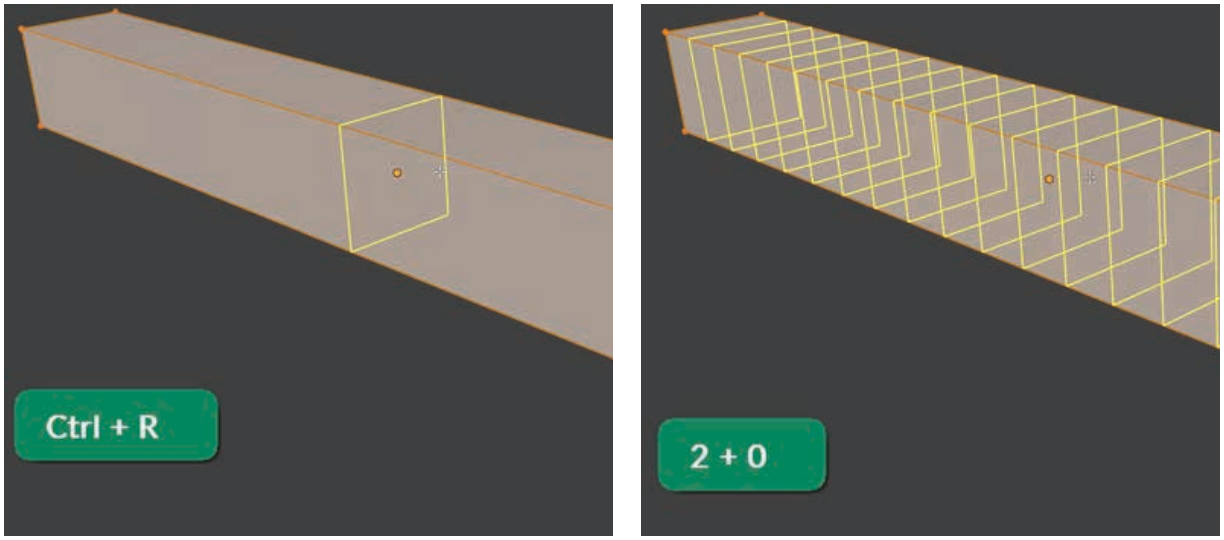
First, don't delete the default cube. Go to **Edit Mode** and **press S+Y+10** on the Numpad to scale the Cube 10 units on the Y-axis.



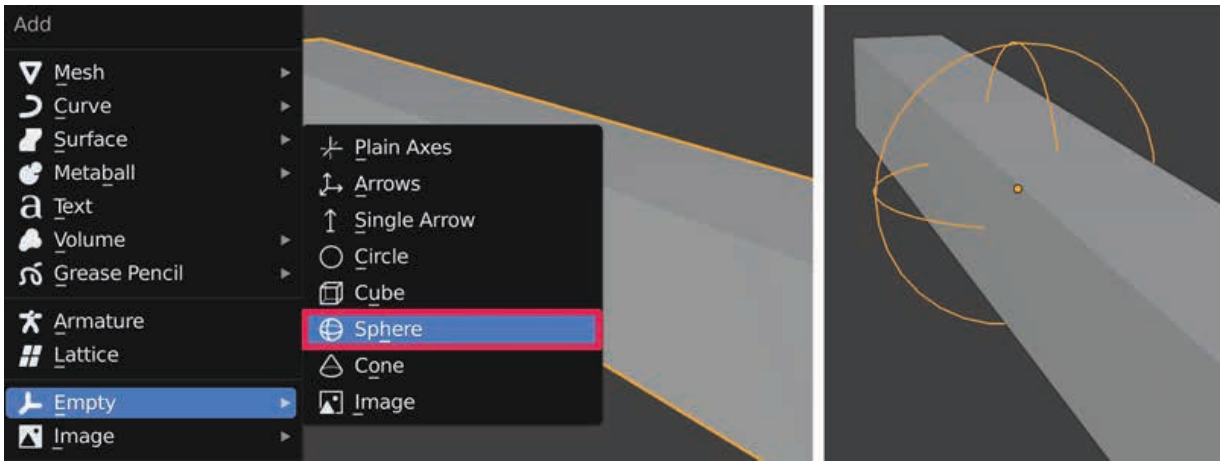
You can also change the scale in the **Resize menu** that's temporarily available just after you scale it in Edit Mode.



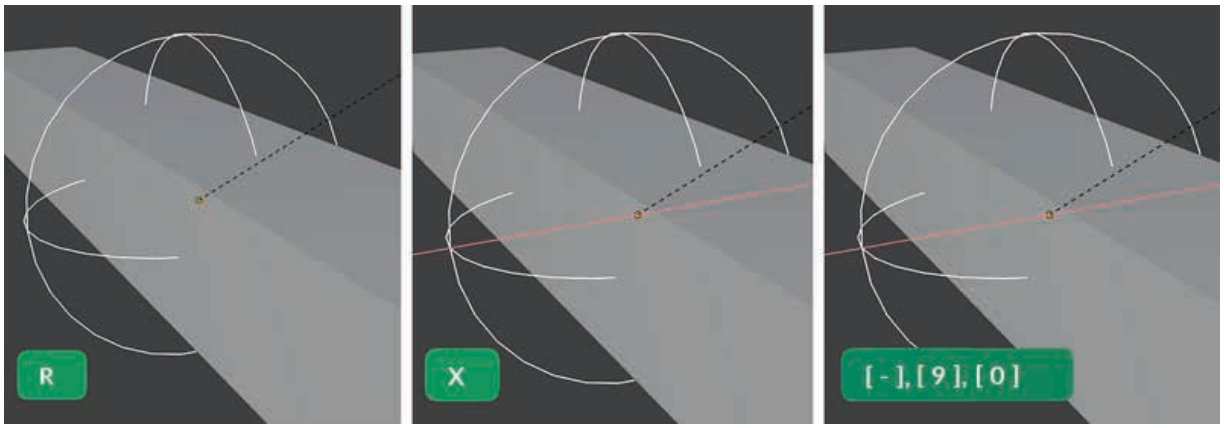
Press Ctrl+R and 20 on the Numpad to add 20 loop cuts. **Press Enter** twice to confirm.



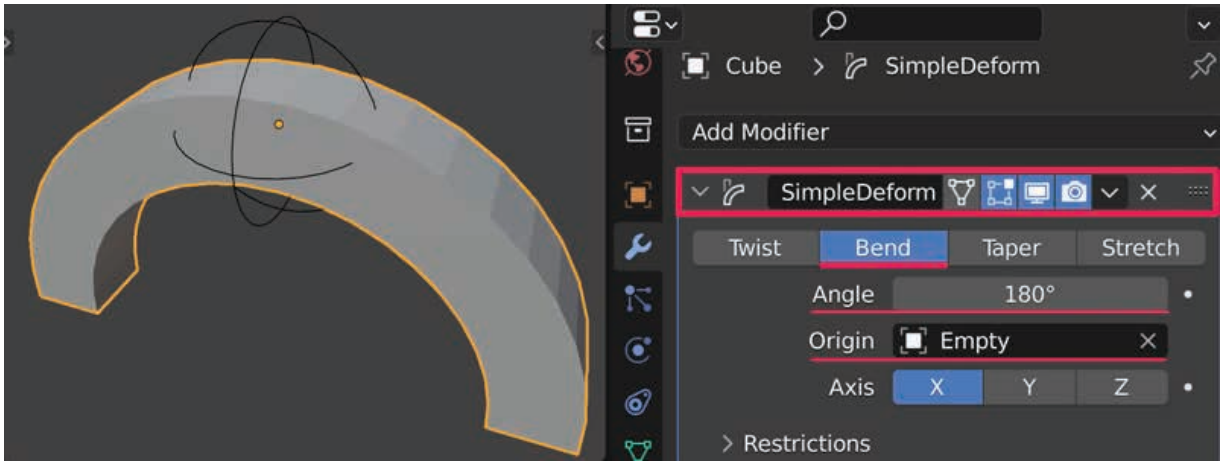
In Object Mode, create an **Empty**. It doesn't matter which type of Empty you choose.



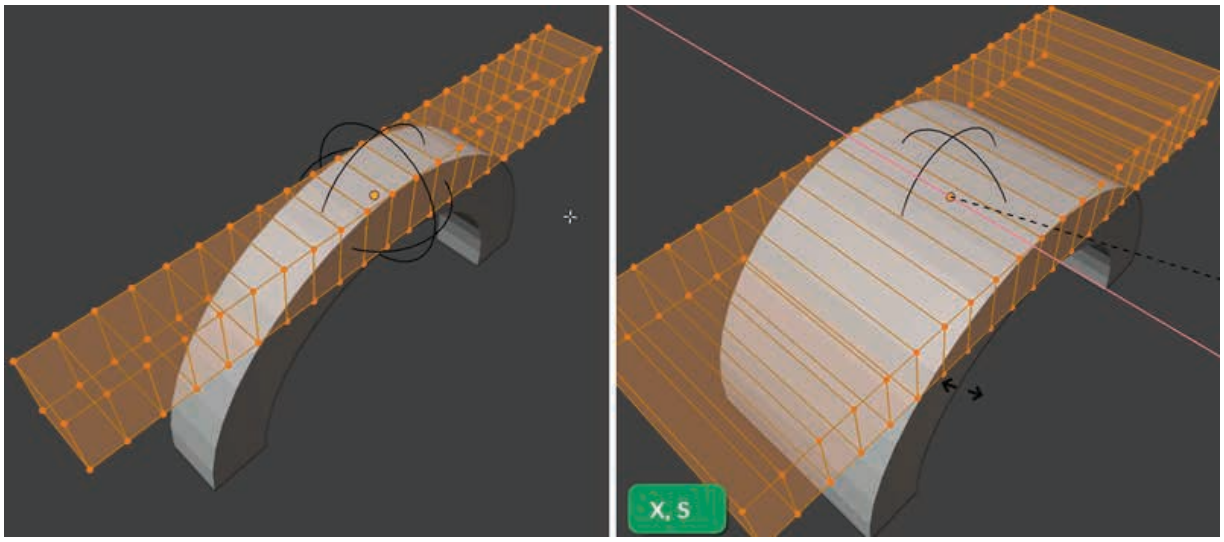
Press R and X and -90 to rotate the Empty -90° on the X-axis.



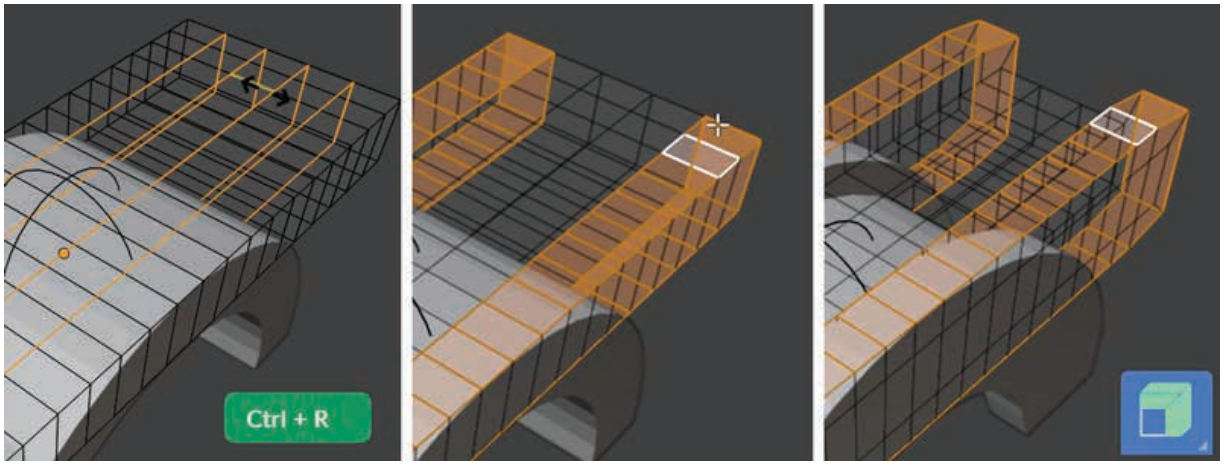
Add a **Simple Deform** modifier to the stretched Cube. Set it to **Bend** at 180° **Angle**. Choose the Empty as the **Origin**.



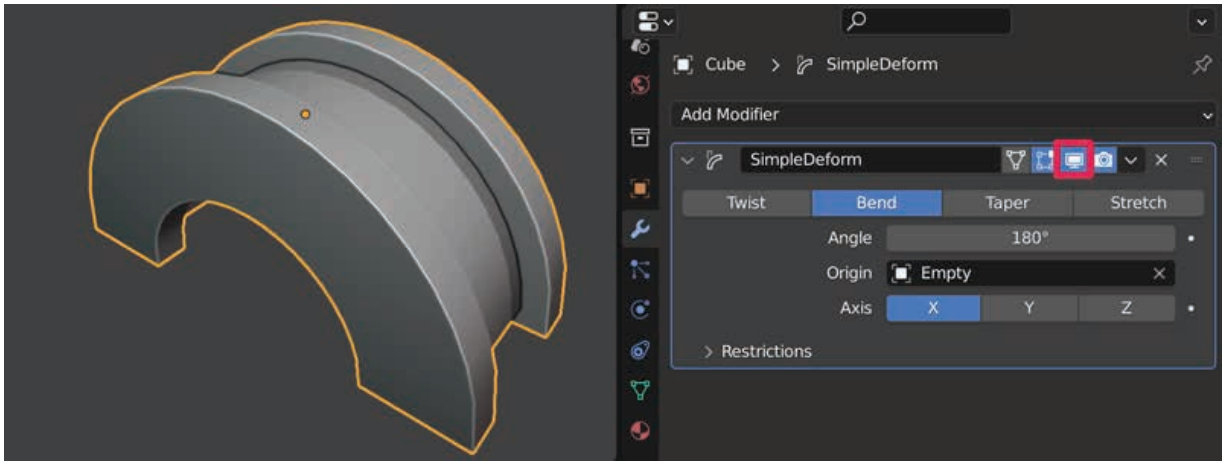
Now you have the basic bridge shape and you can add more detail in Edit Mode. Here I'm scaling it on the X-axis by **pressing S + X** and moving the mouse.



Now you have the basic bridge shape and you can add more detail in Edit Mode. Below I'm adding some loops by **pressing Ctrl+R** and scrolling the mouse wheel up.

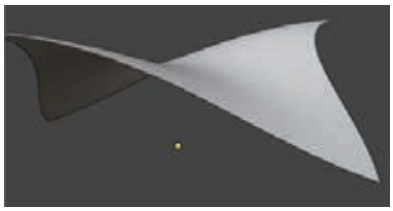


You can turn the modifier off, to make it easier to work on it and then turn it back on later to see what the final result looks like.

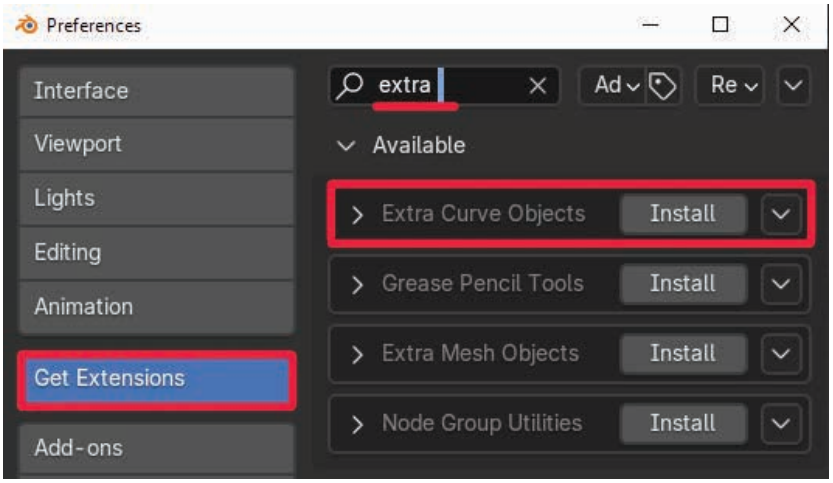


Surface patch from Curves

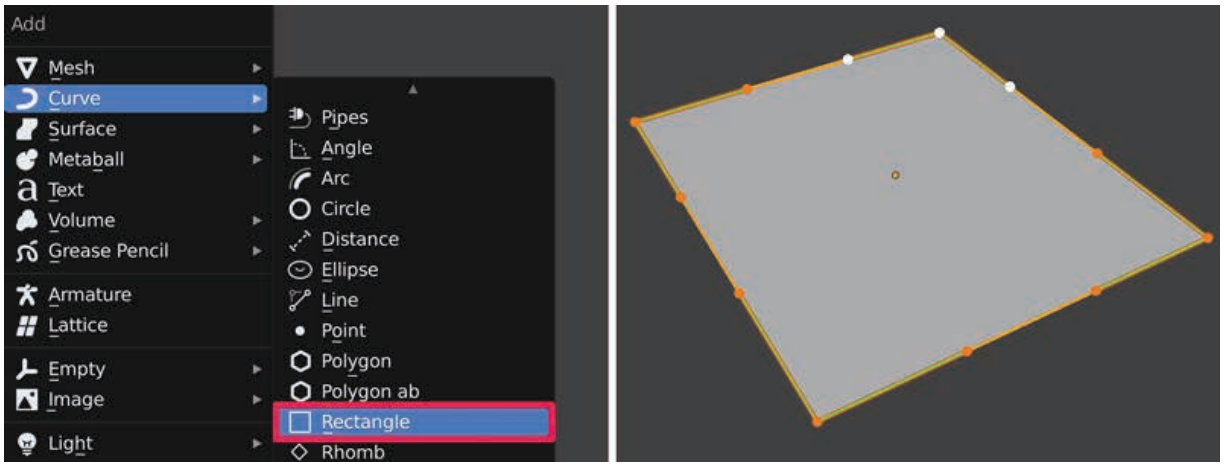
Using Curves, you can create surfaces with perfect smoothness.



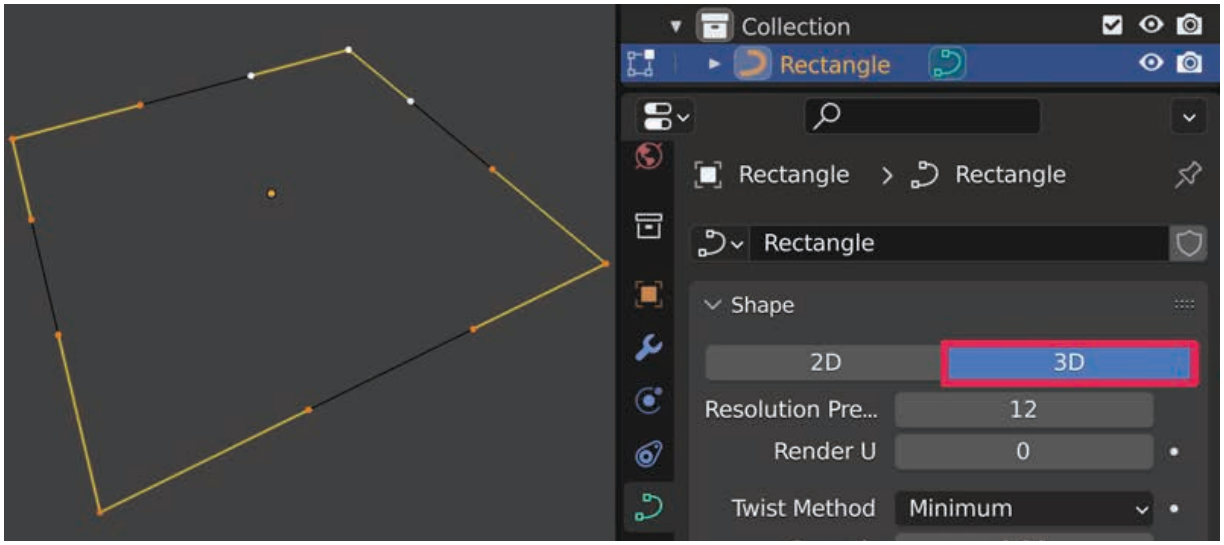
Enable the **Extra Curve Objects** add-on.



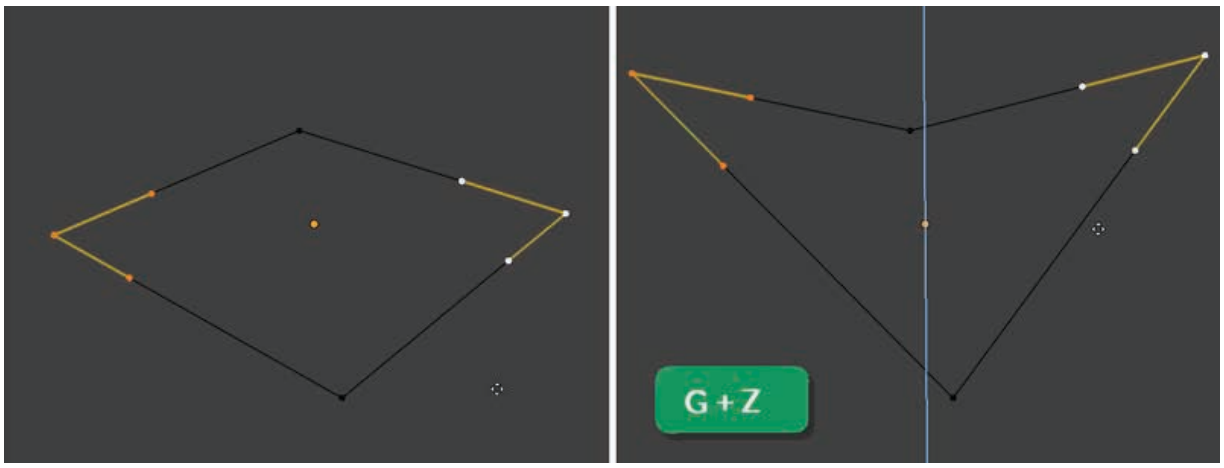
Add a Curve > Rectangle.



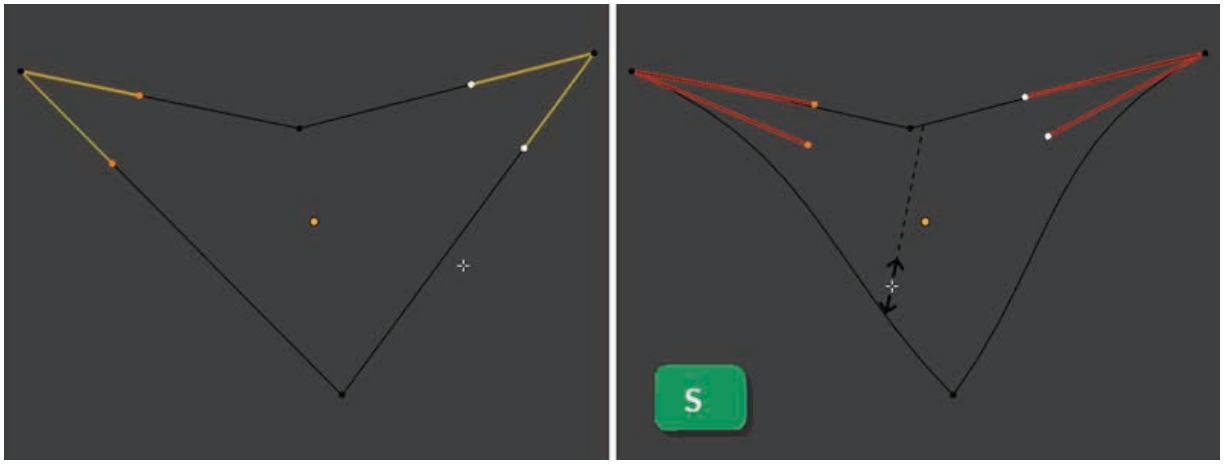
Turn on 3D Shape.



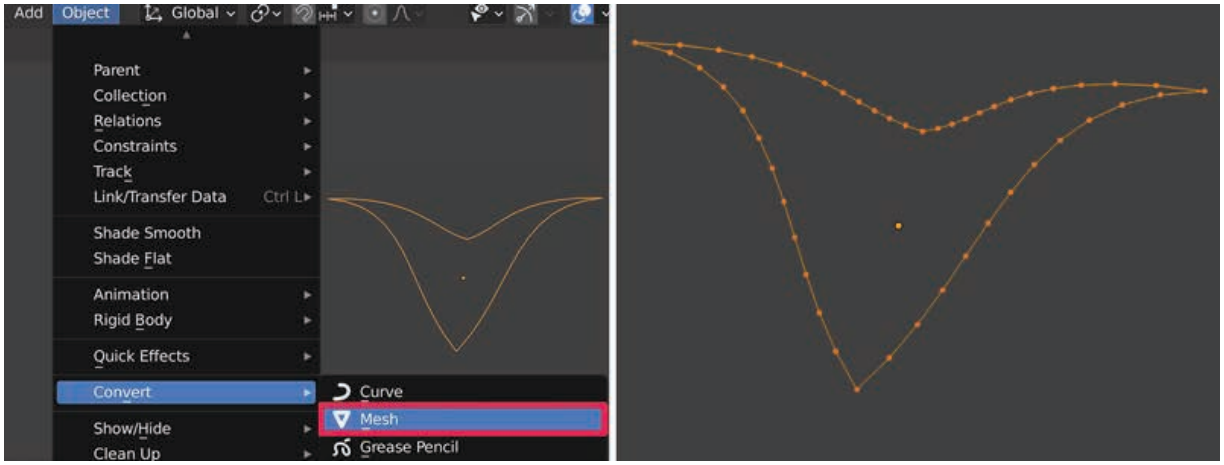
Move the curves into the shape you want.



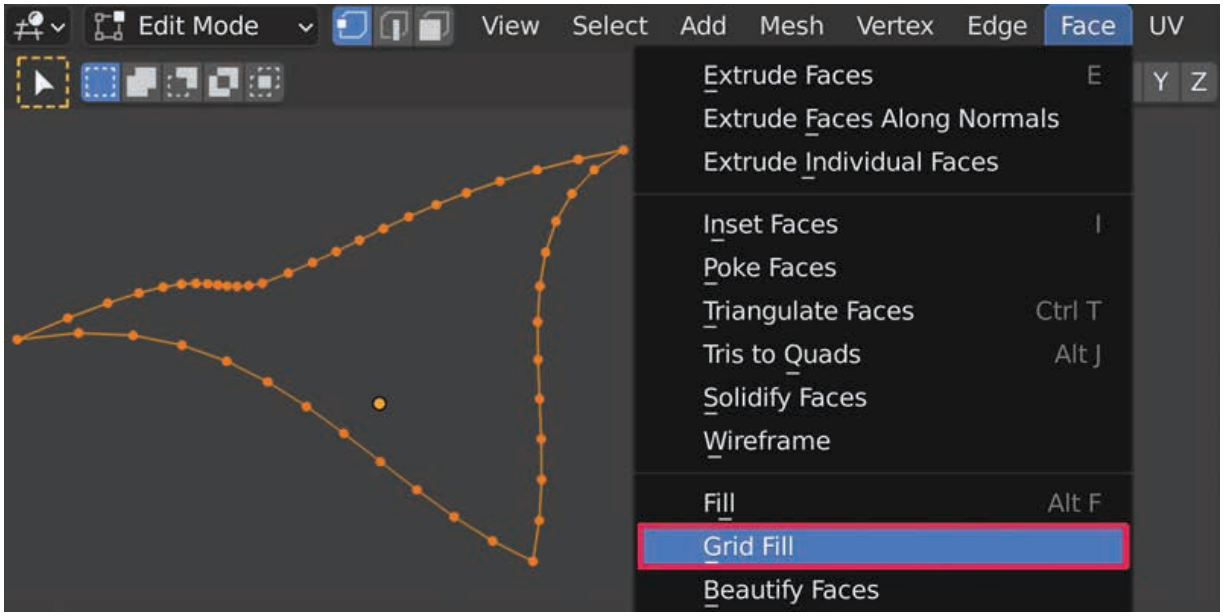
You can select individual handles, and move them or scale them.



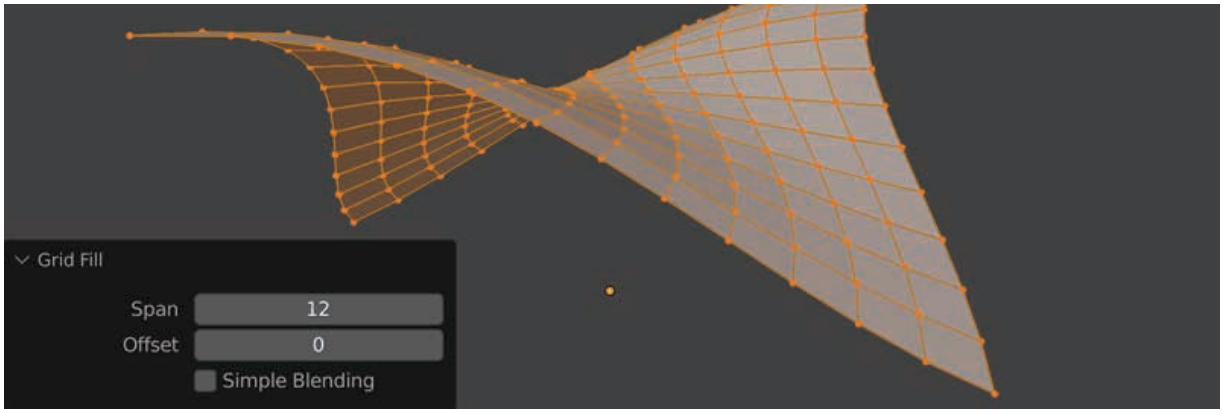
Go to Object > Convert to > **Mesh from Curve**.



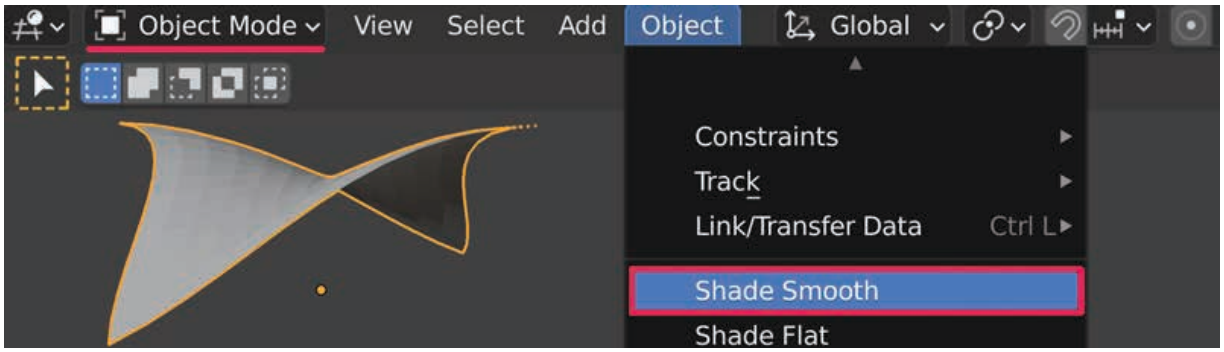
Select all vertices in Edit Mode and go to Face > **Grid Fill**.



In the Grid Fill menu, you can change the distribution of edge loops.



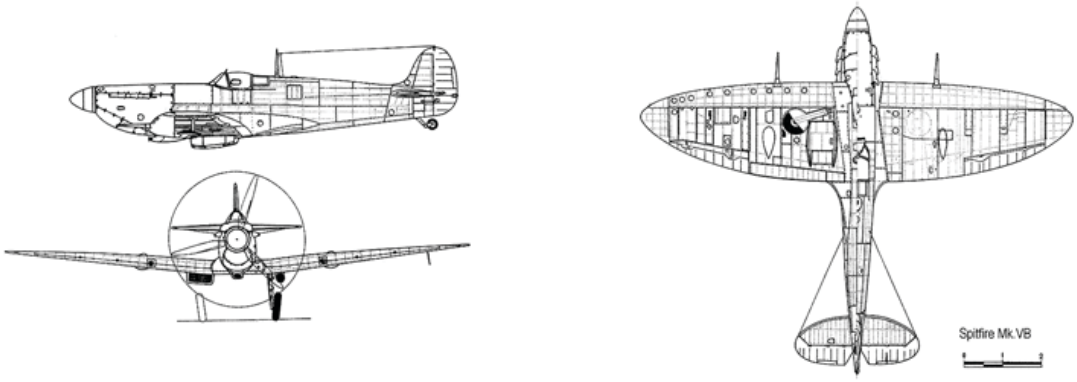
Turn on **Smooth Shading**.



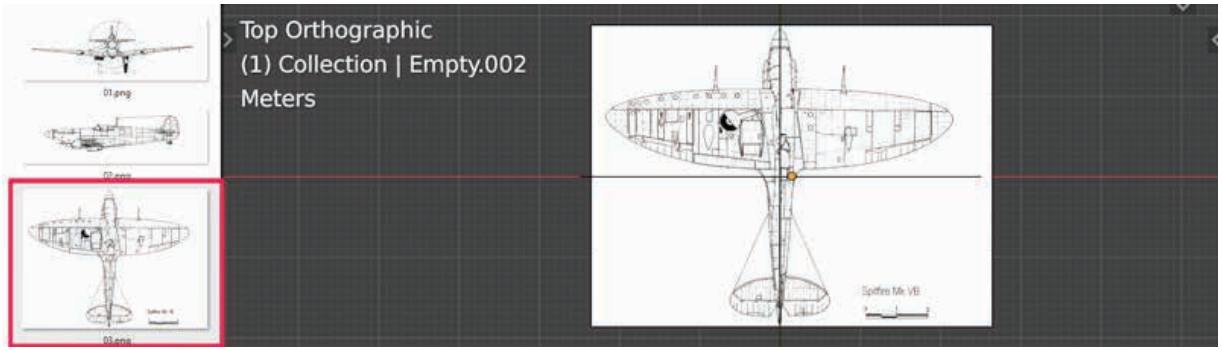
Now the surface looks smooth.

Model an airplane wing

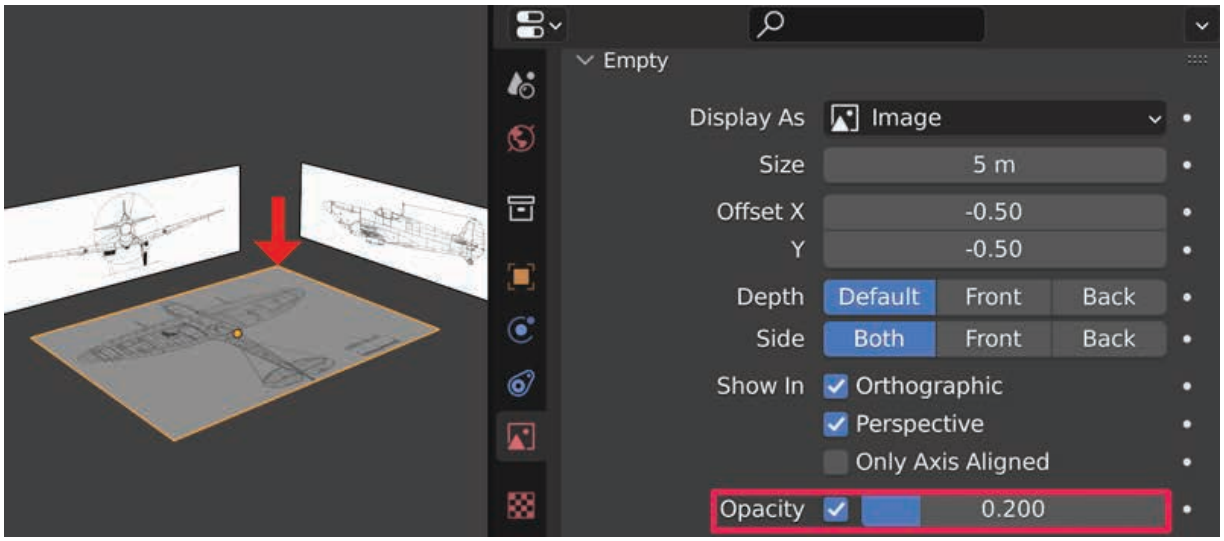
To model an airplane wing which is based on a real aircraft, find some blueprints online. I'm using these Spitfire blueprints.



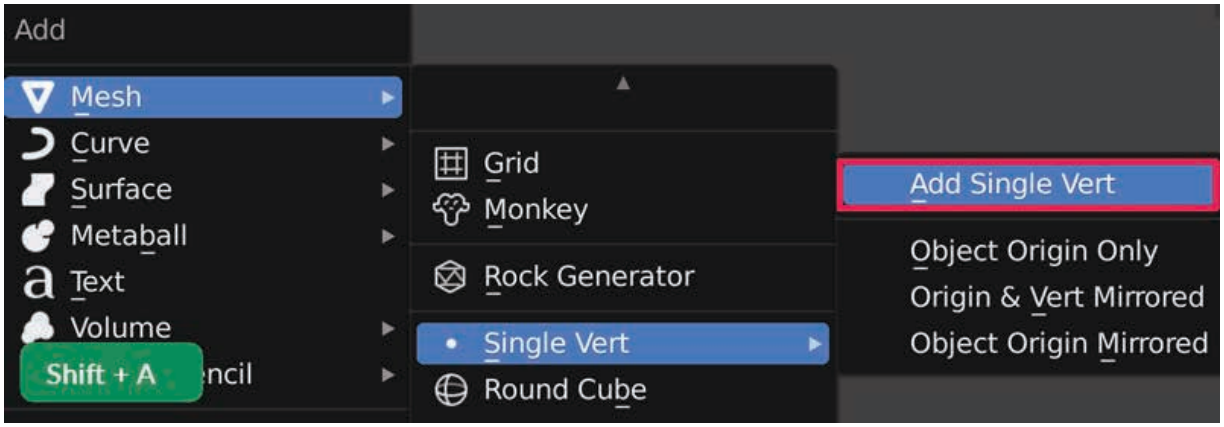
Drag-and-drop the blueprints in your viewport. The easiest way is to have separate images for the front, back and top and then drag them into the corresponding orthographic views. Note that not all image formats are supported, check the list here.



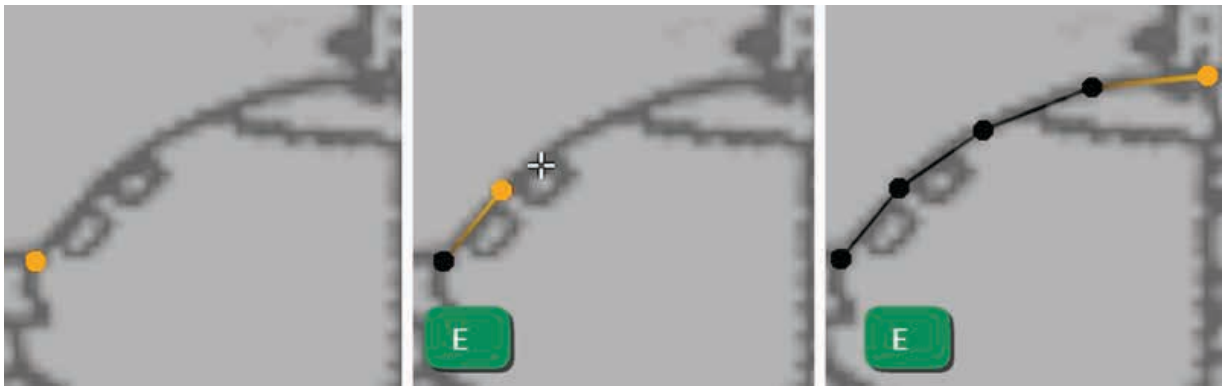
You can then enable **Opacity** and reduce the transparency of the reference images. Try to line them up as much as possible.



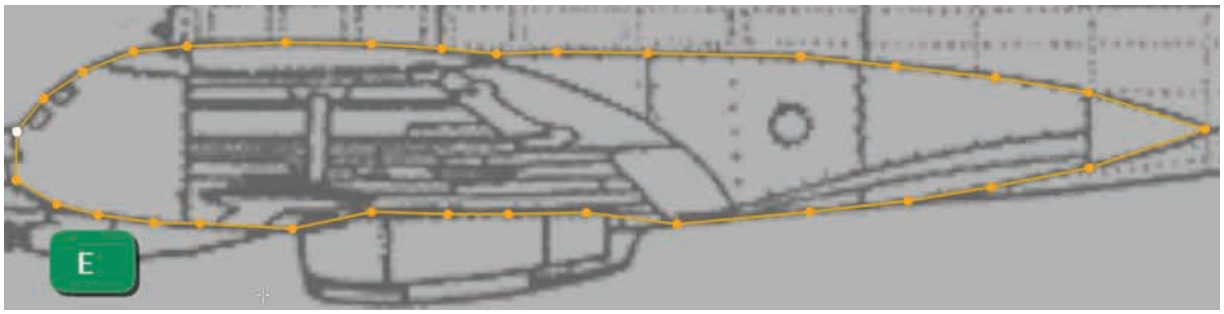
Then create a single vertex and extrude it to create the shape of the wing. You need to have the Extra Objects add-on enabled in Preferences to have this option.



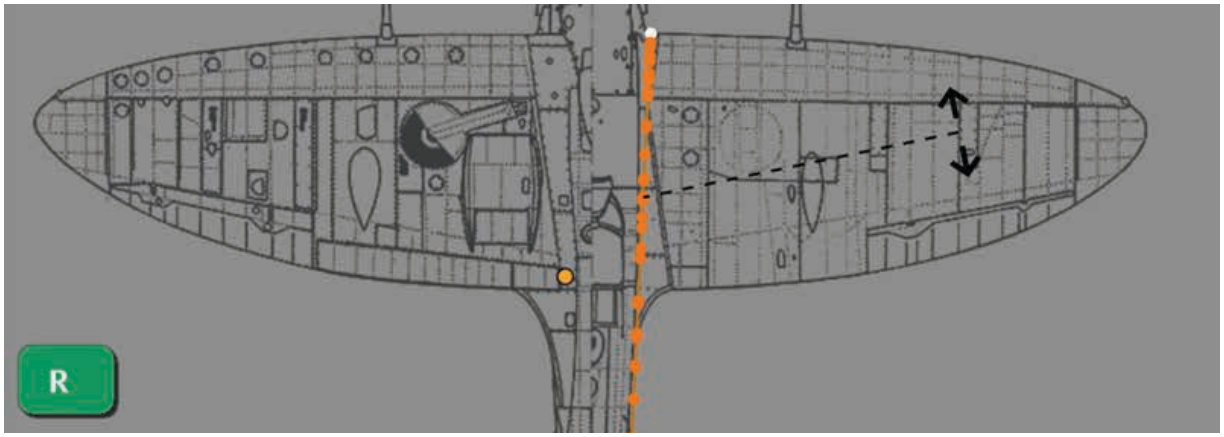
Press E to extrude the vertex, following the wing shape.



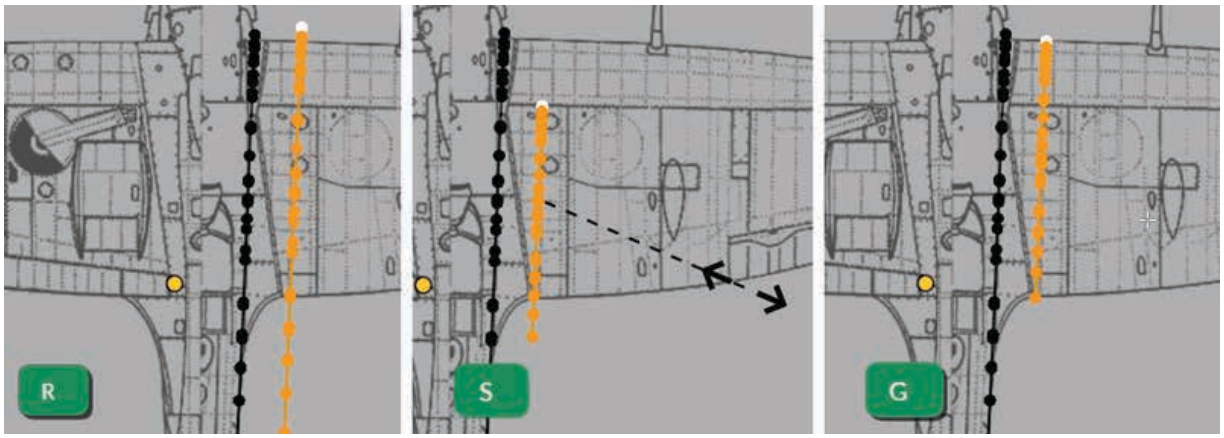
When the last vertex reached the first one, select both and merge them by **pressing M**.



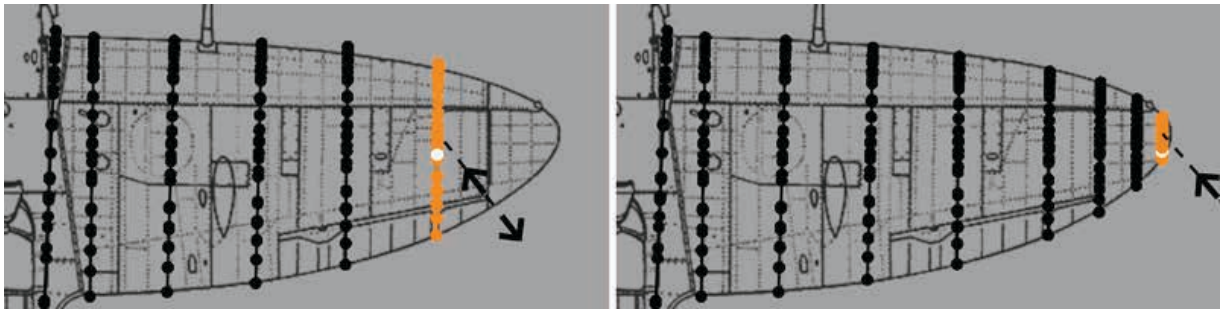
Check the different views and make the necessary adjustments.



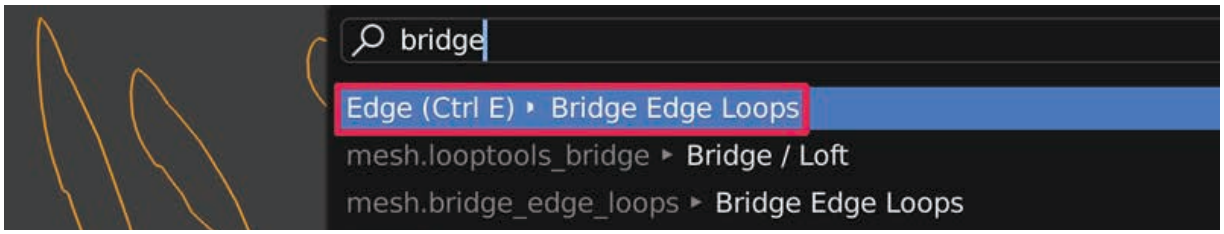
In Edit Mode, **press Shift+D** to duplicate this geometry. And move it over to create the next slice of the wing. Transform it to fit that part of the wing by scaling it and moving the vertices.



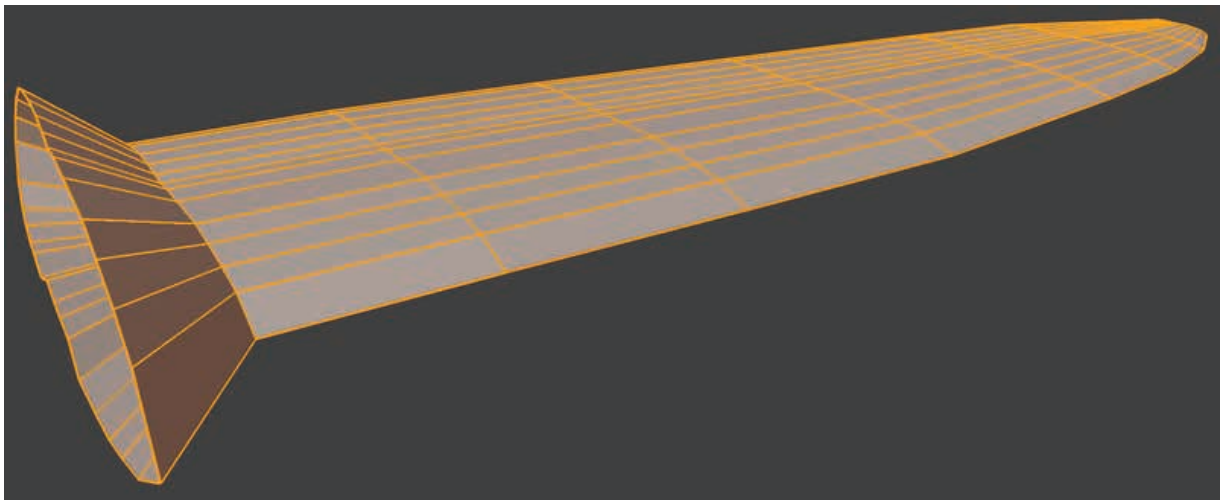
Repeat this until you've got enough slices to make the entire wing.



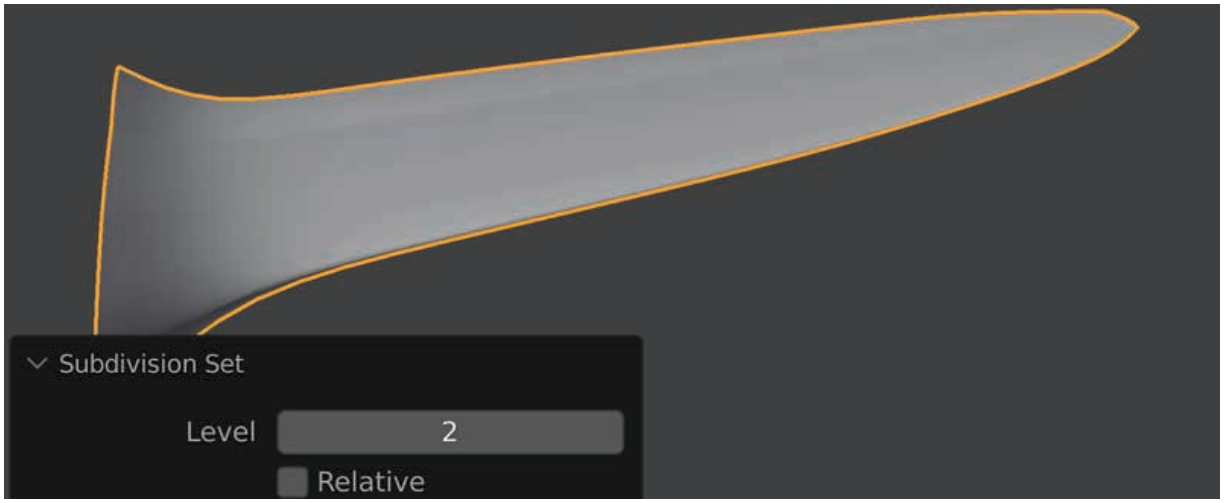
Select all the edges and **press F3**, search for "**Bridge Edge Loops**". Or, in Edge selection mode, right-click and choose "Bridge Edge Loops".



As you can see, the bridges are now connected by faces.



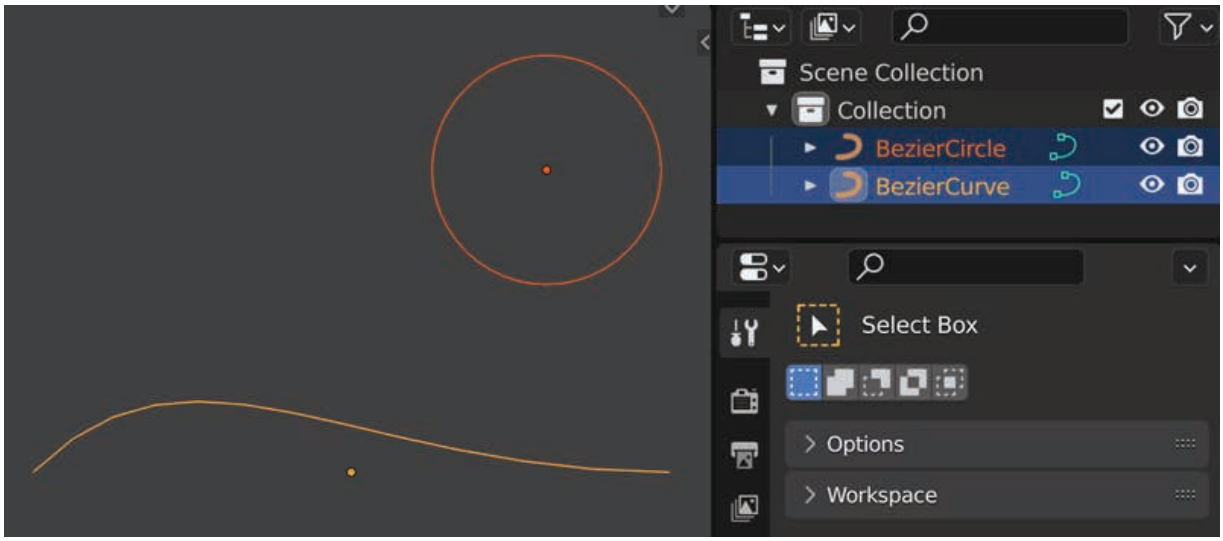
Press Ctrl+2 to add a Subdiv modifier set to level 2.



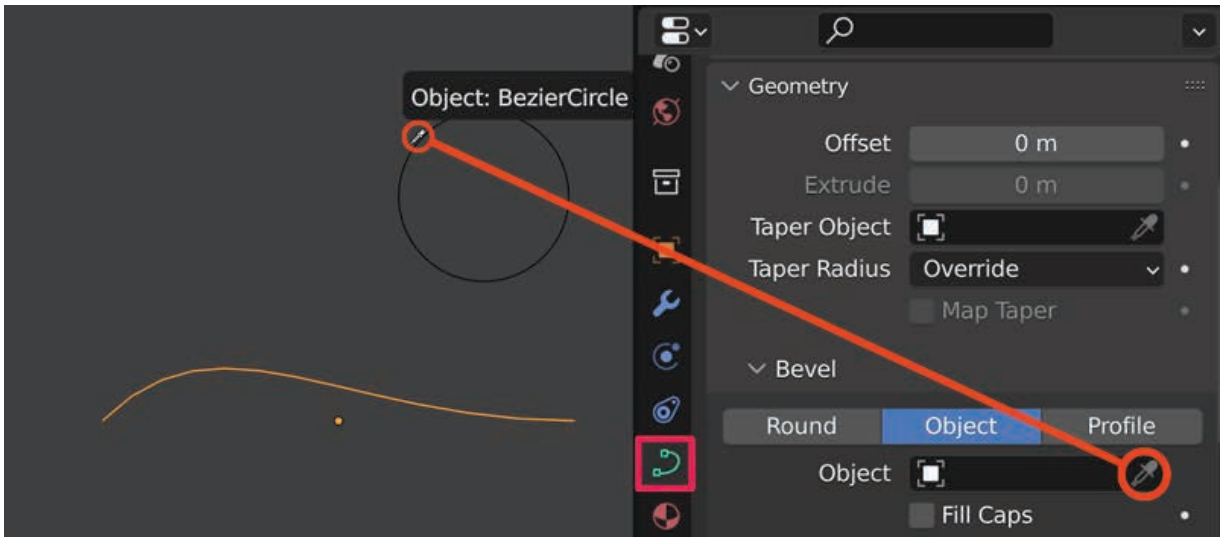
Beveling and Tapering Curves

Using Curves you can create complex and smooth shapes such as this pipe.

Add a Bezier Curve and a Circle Curve.

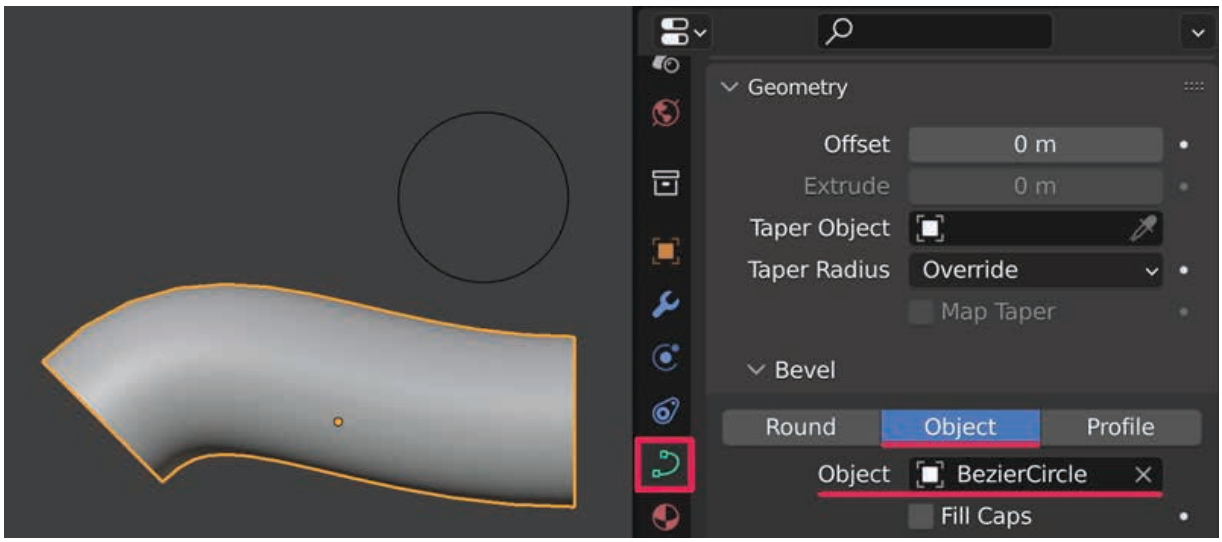


With the Bezier Curve selected, choose the Circle as the Bevel object.

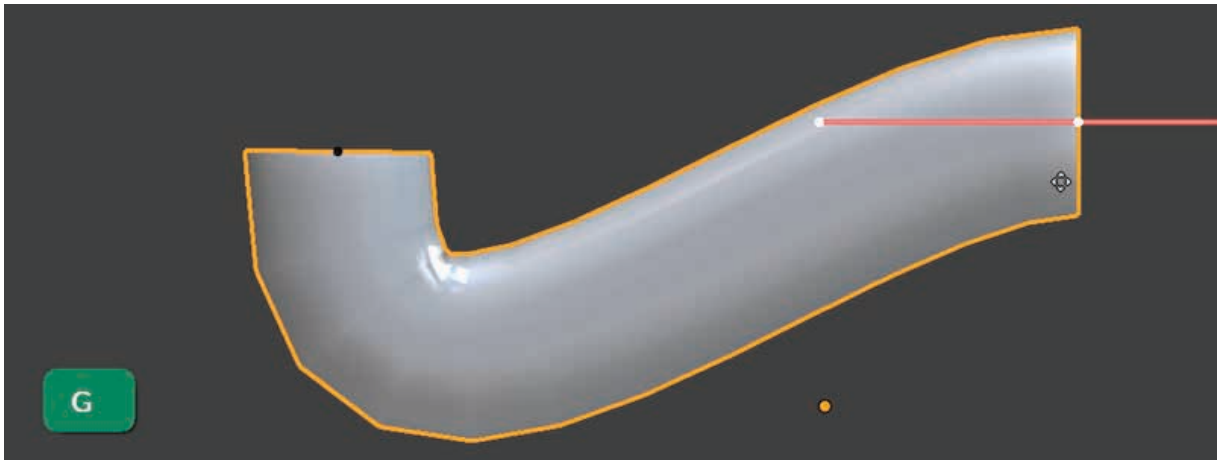


Click here to return to the index.

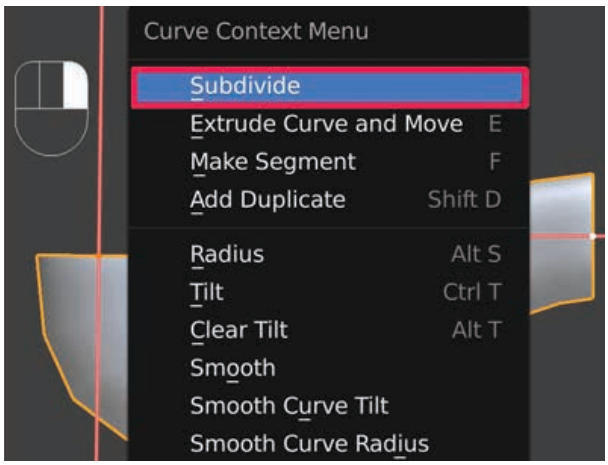
As you can see, this creates geometry for the Curve.

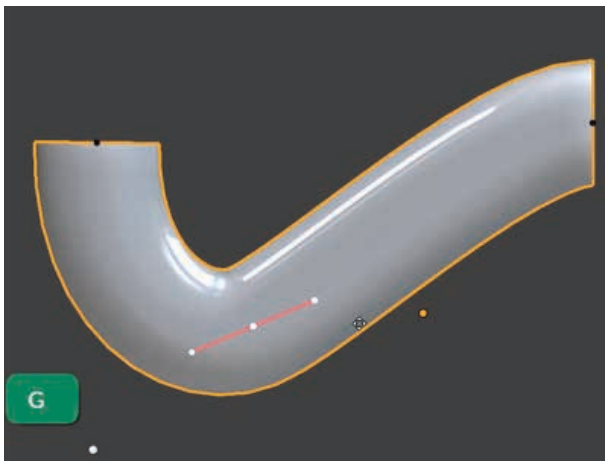


Go to **Edit Mode** and change the shape of the Bezier Curve. You can move the vertices with **G,** rotate with **R** and scale them with **S**.

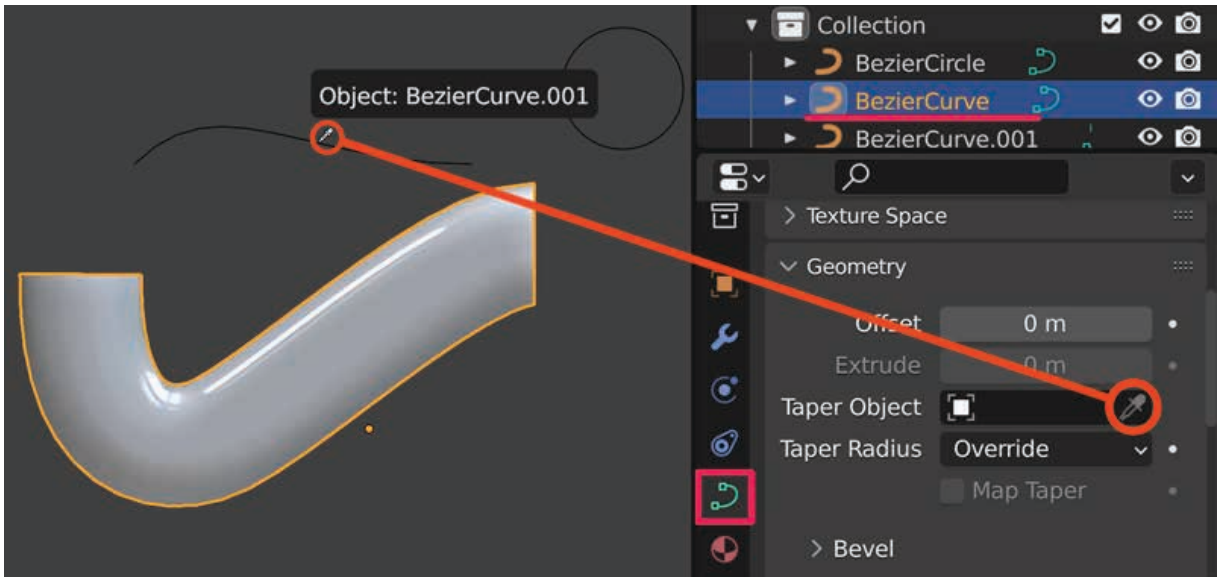


Select two vertices, right-click and choose **Subdivide** to add another vertex in between.

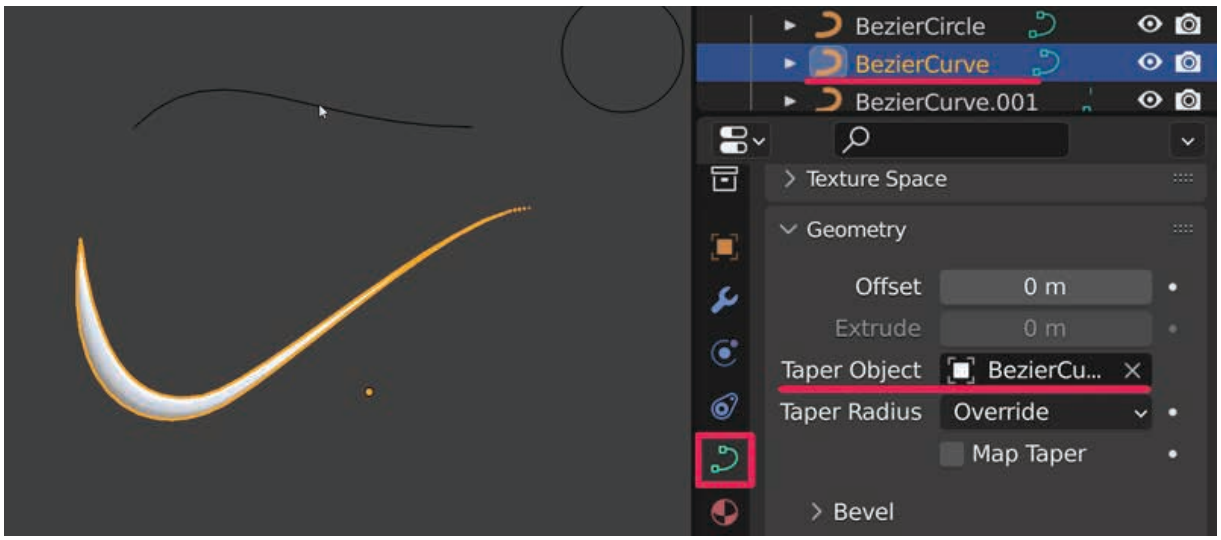




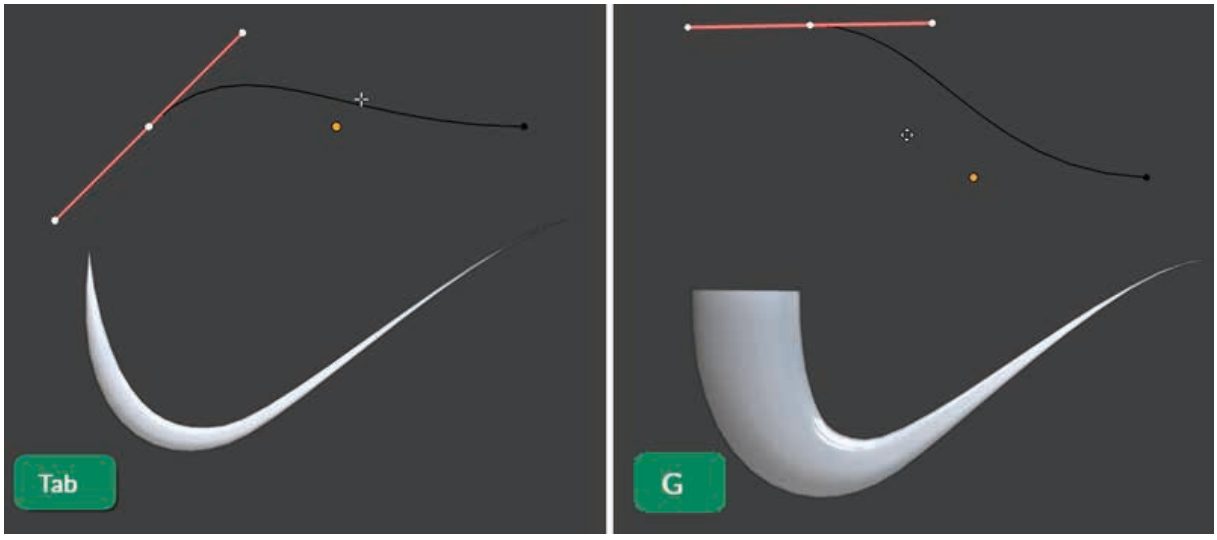
Add another Bezier Curve and use it as a Taper object for the pipe.



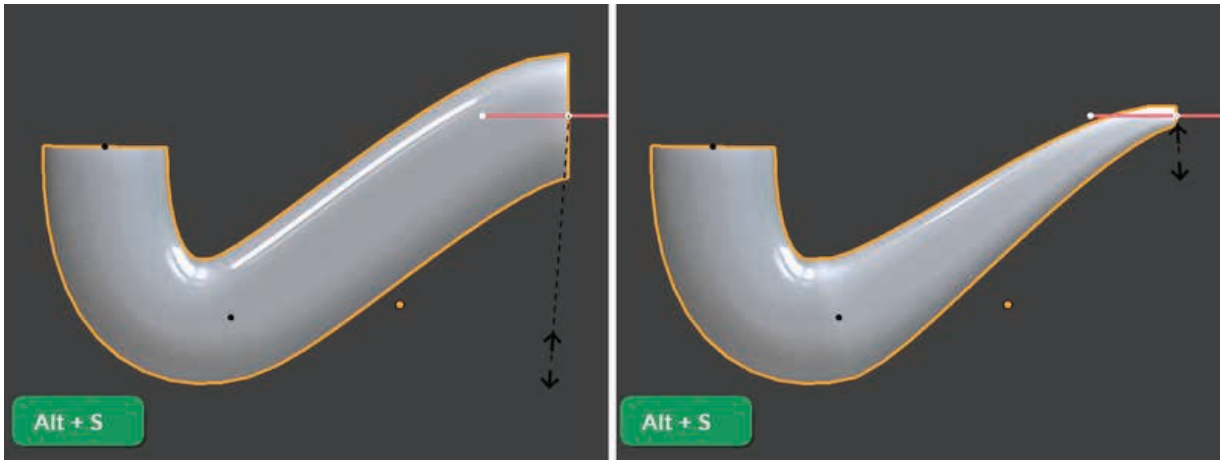
As you see, it influences the shape of the geometry.



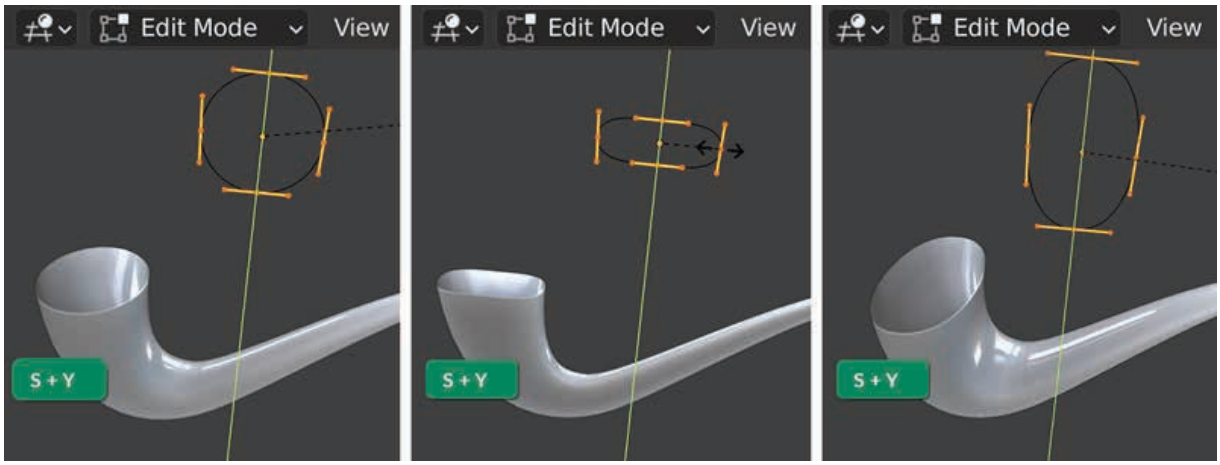
Move the vertices up or down to get the shape you need.



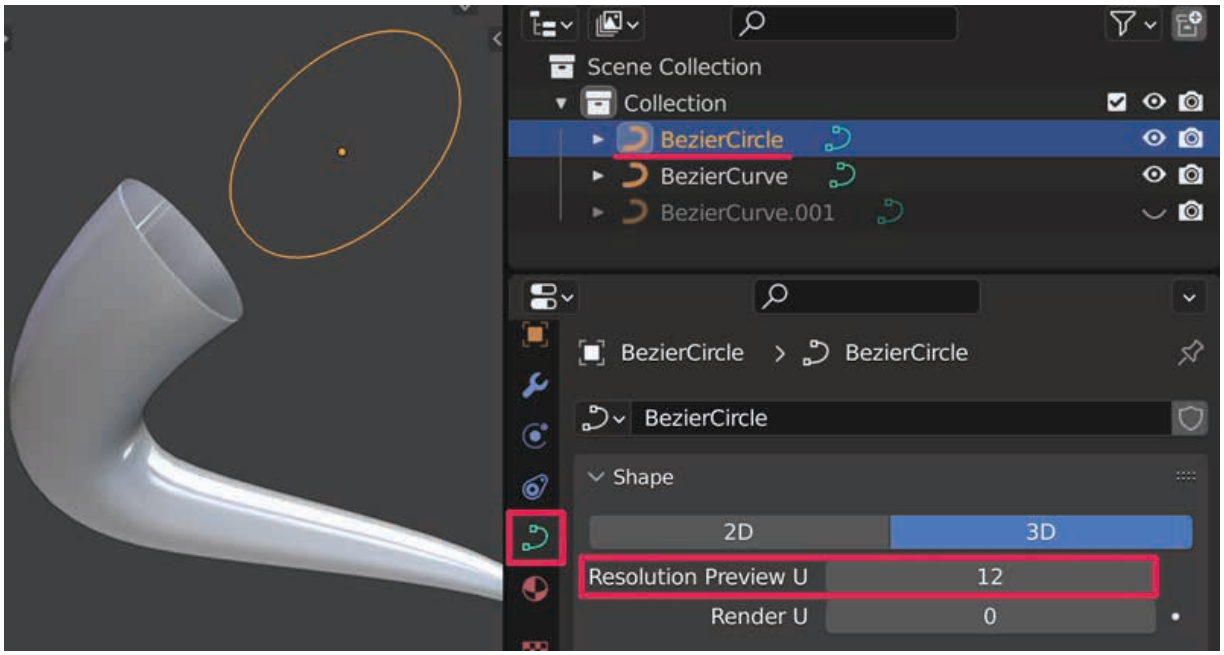
You can do this without the Taper object by selecting a vertex and **pressing Alt+S**. This changes the **weight** of the vertex.



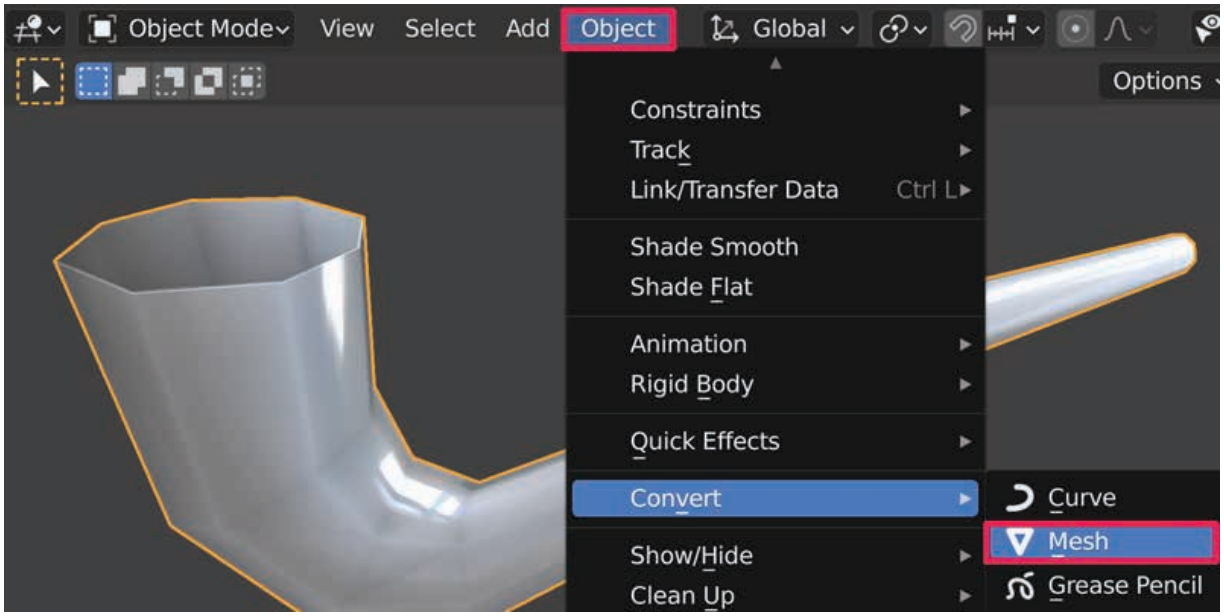
You can also change the shape of the bevel by adjusting the Circle in Edit Mode.



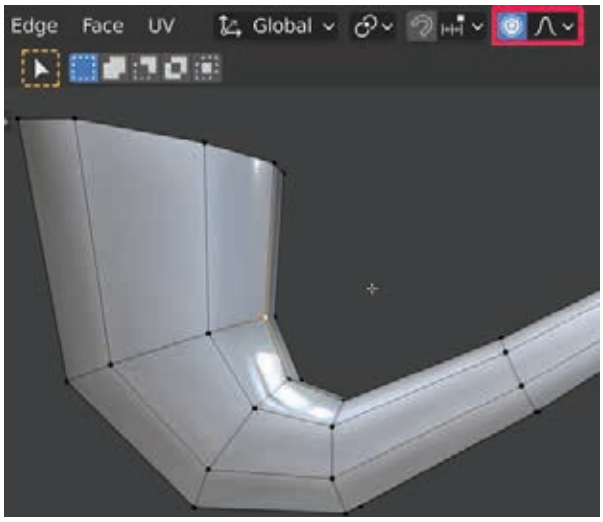
If you want a more low-poly pipe, you can change the resolution. Do this for both the bezier curve as well as the bevel object.

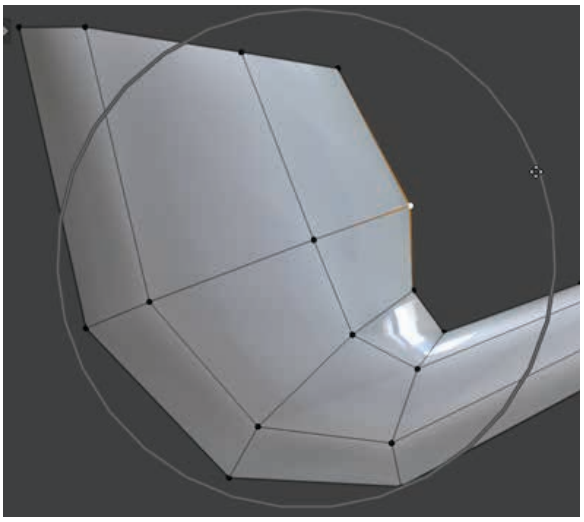


For fine-tuning you can **convert it to Mesh** and use proportional editing.

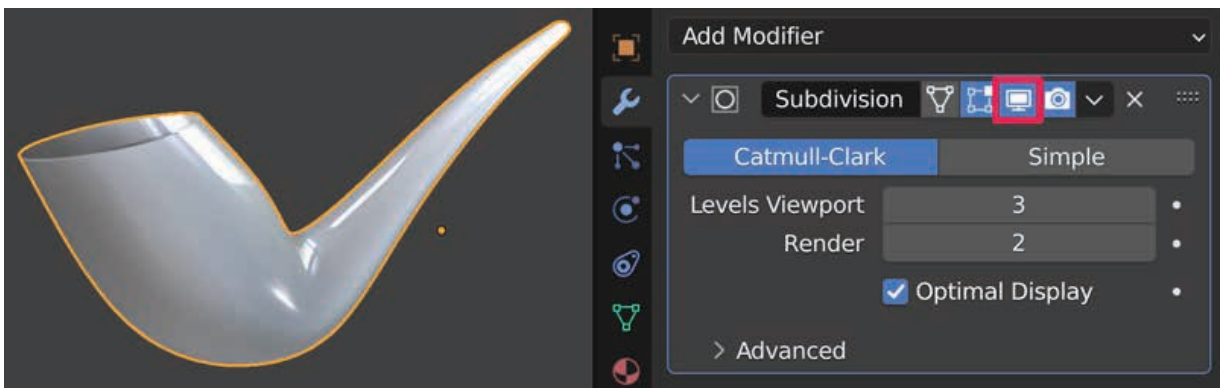


For fine-tuning you can **convert it to Mesh** and use proportional editing.



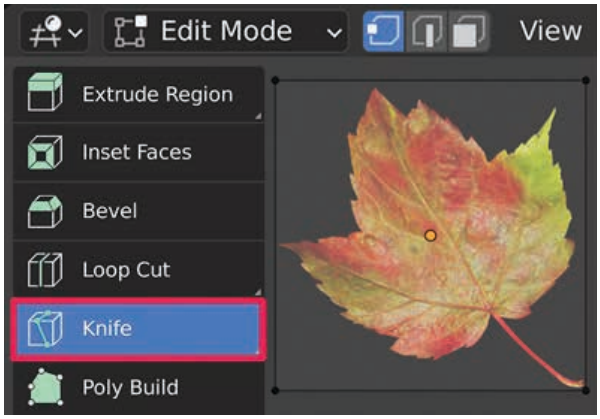


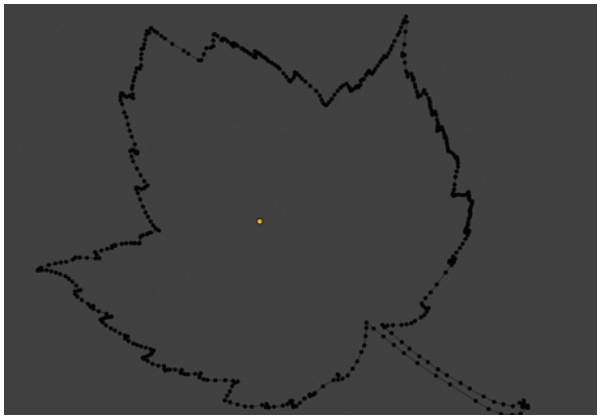
Using a Subdivision modifier, it will look smooth.



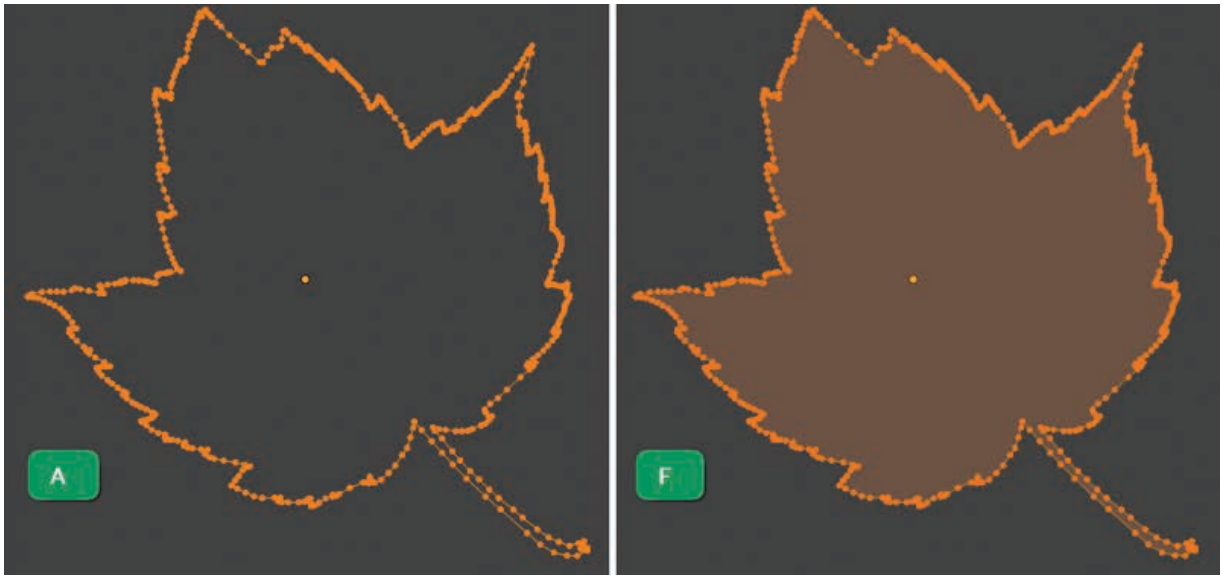
Filling complex N-gon shapes like leaves

In this example, I've traced the outline of a leaf.

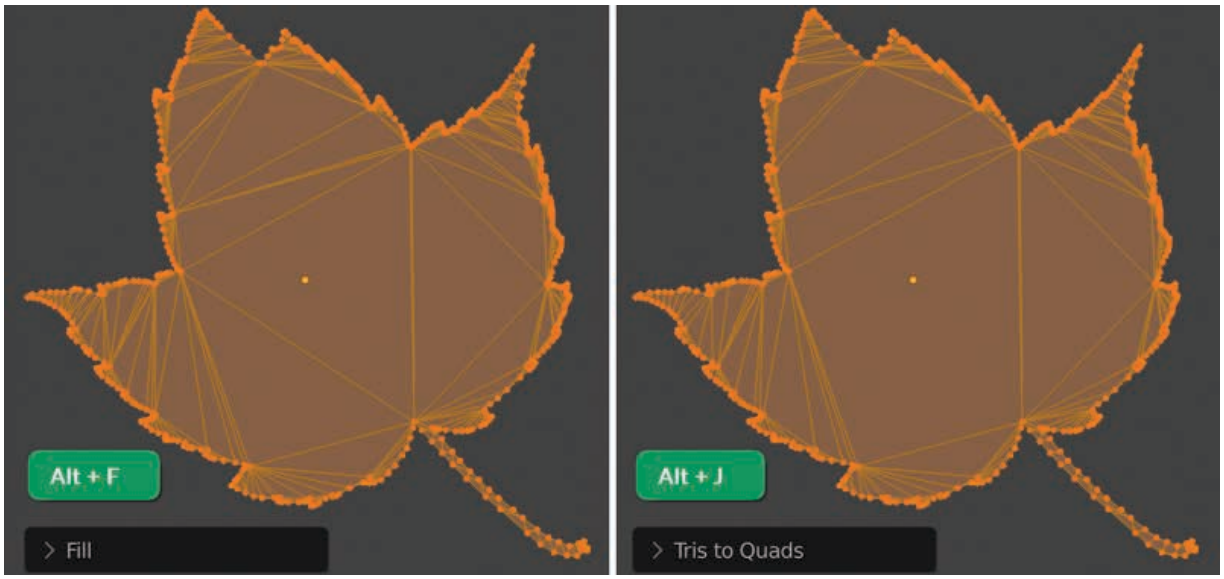




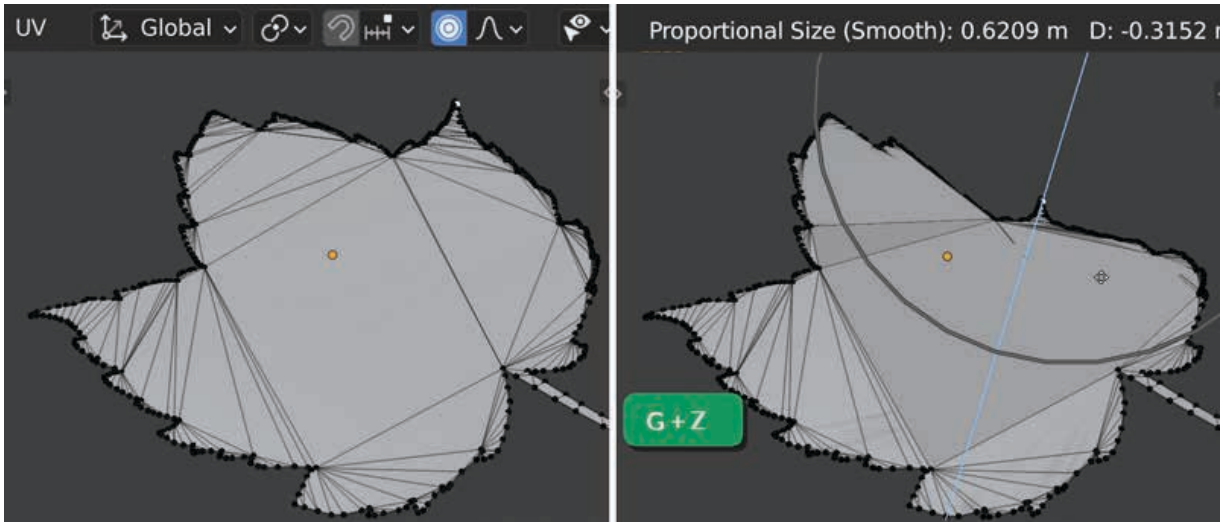
Pressing F fills it with one N-gon. It is perfectly flat and can't bend at all.



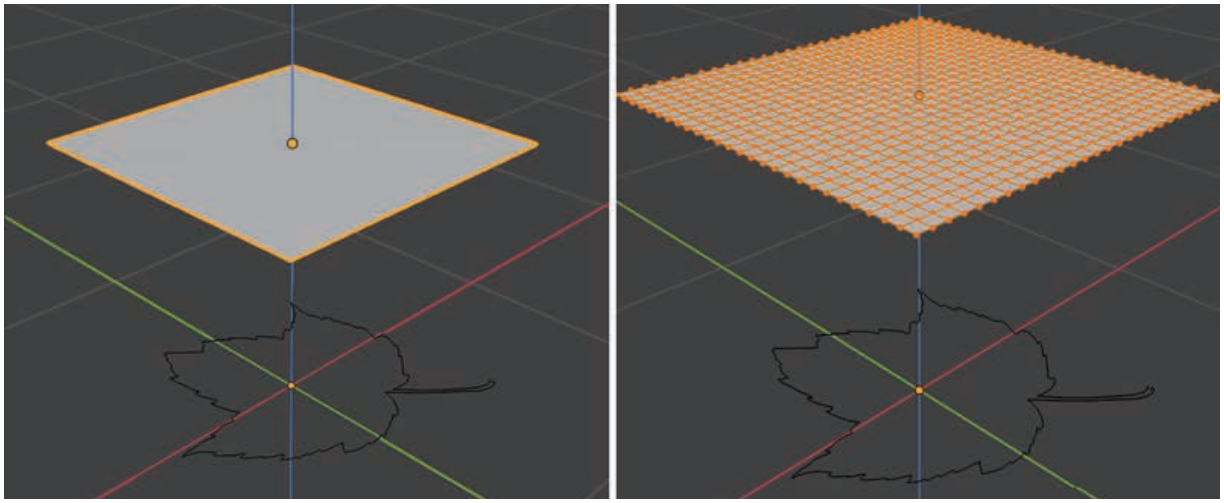
Pressing Alt+F Fills it with triangles. Follow that up with **Alt+J** to remove some unnecessary triangles.



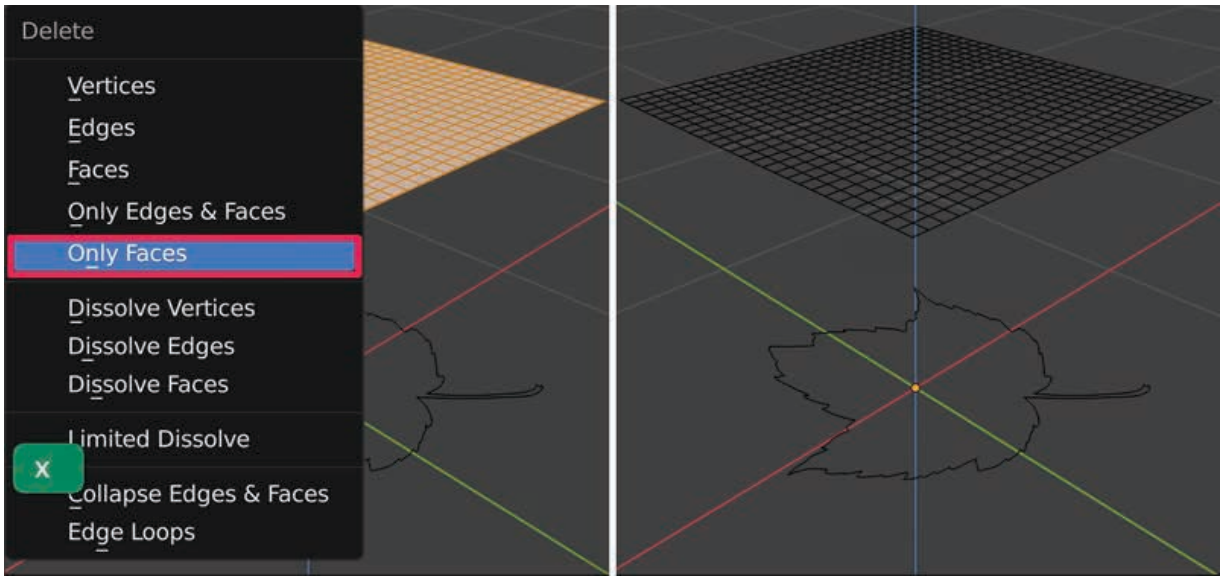
Now we can bend it a bit, but you get these artefacts.



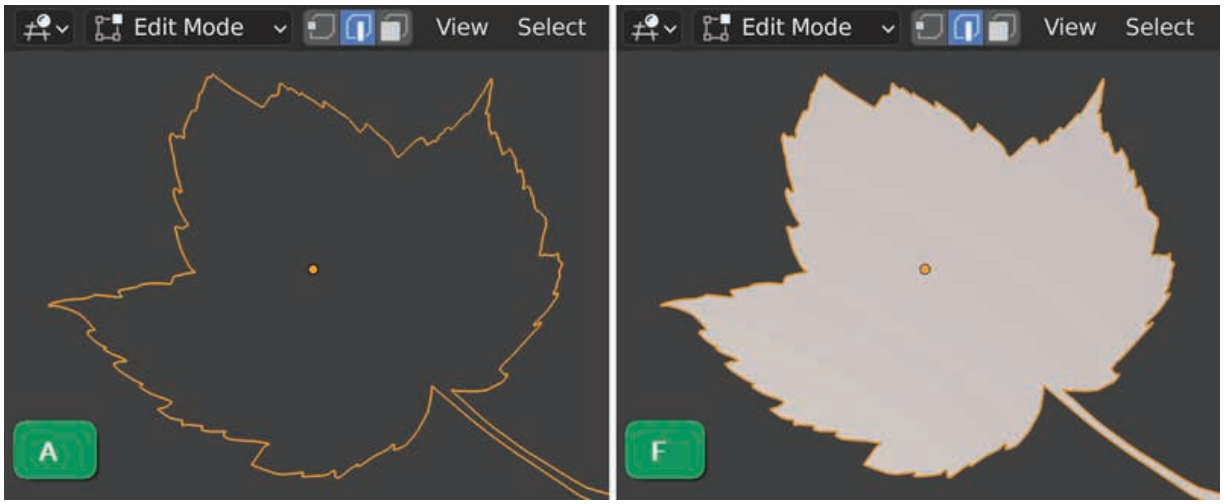
A third method is by first creating a Plane with subdivisions and then placing it above the leaf.



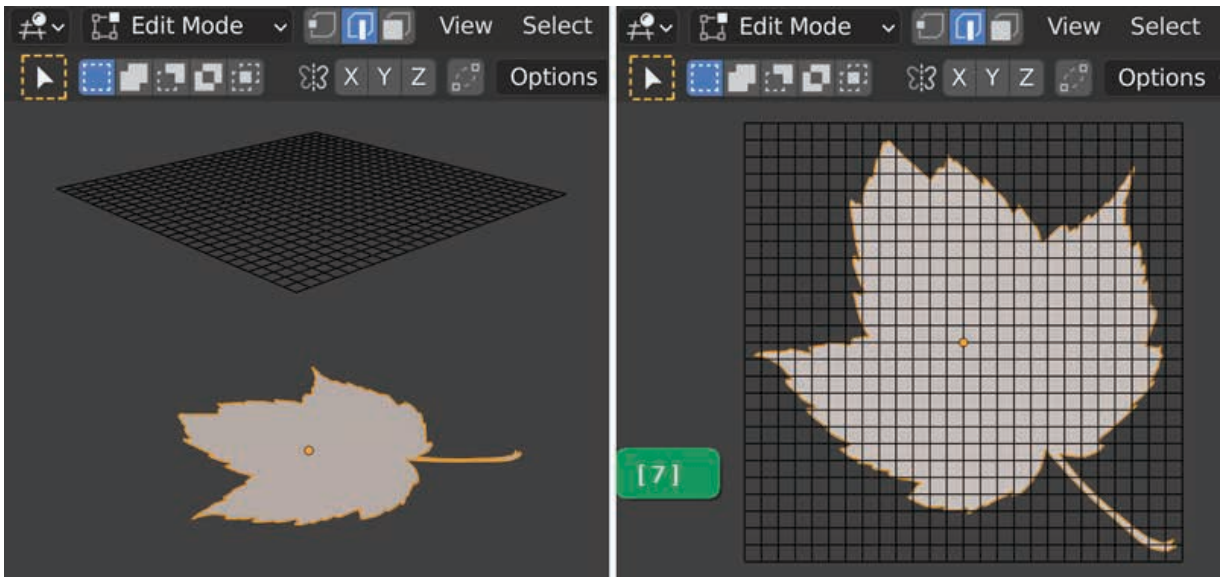
Then in Edit Mode, delete only the Faces by selecting all and **pressing X > Only Faces**, from the subdivided plane. This gives you a kind of grid of Edges.



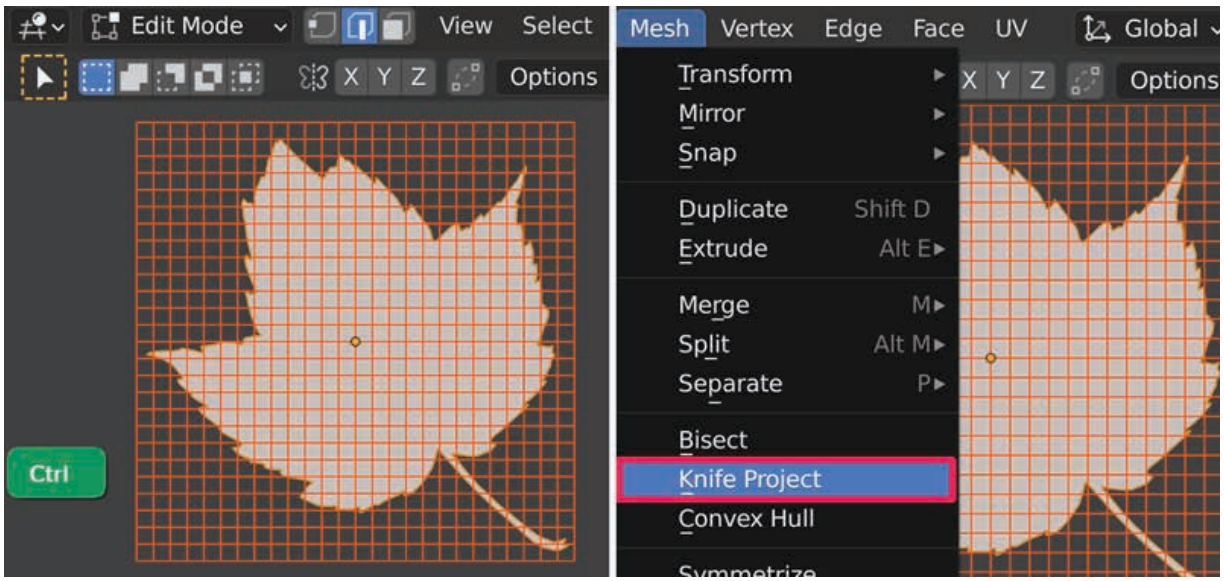
Fill the Leaf with a single N-gon in Edit Mode by selecting all and by **pressing F**.



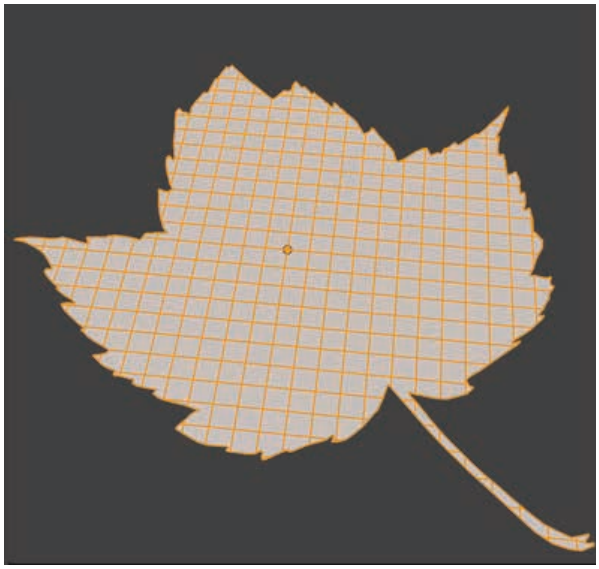
Select the leaf, go to **Edit Mode.** Switch to top view by **pressing 7** on the numpad to see everything from the top.

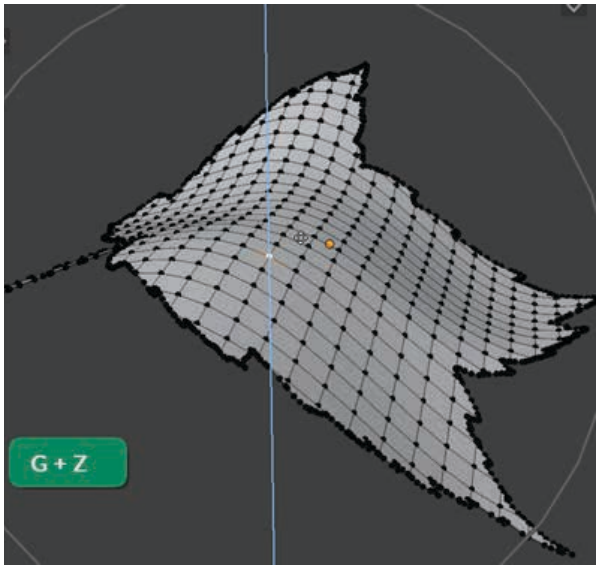


Hold Ctrl and left-click on the plane to select it. Go to Mesh > **Knife Project**.



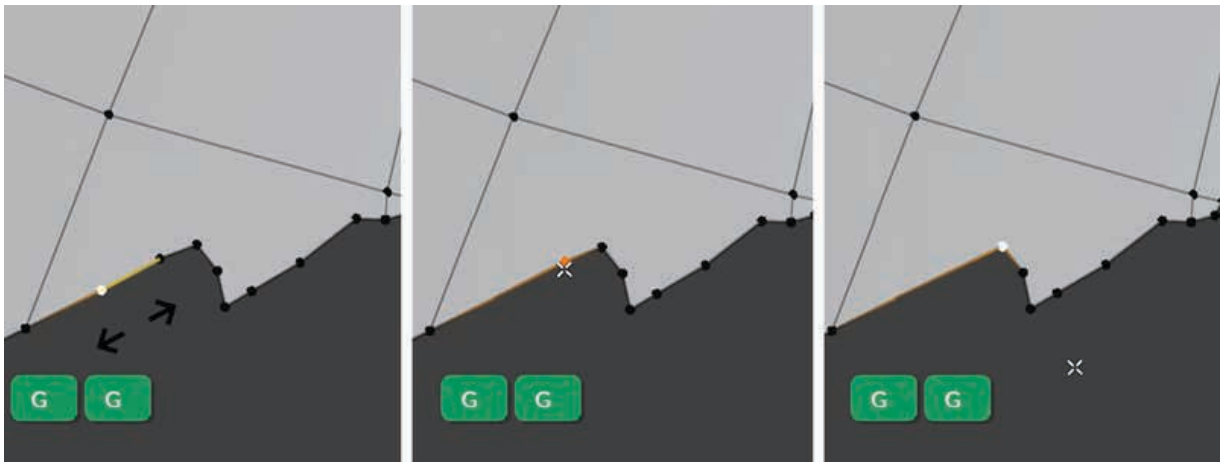
Now the Leaf has many square cuts and can bend well. You can delete the Plane.





There are still small N-gons but they don't cause shading issues at this scale.

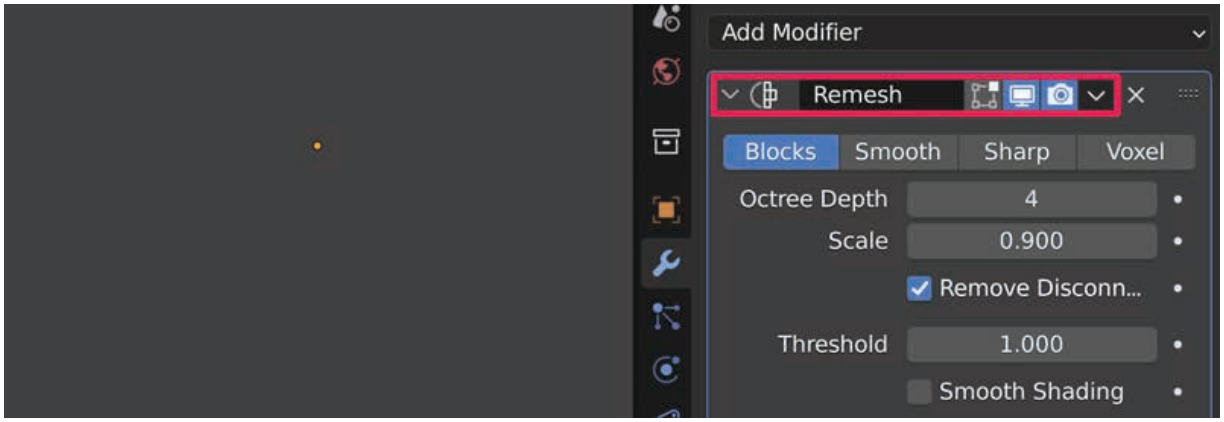
You can turn on **Auto Merge** and slide vertices by **pressing G twice** in order to clean up these parts, if you're into that sort of thing.



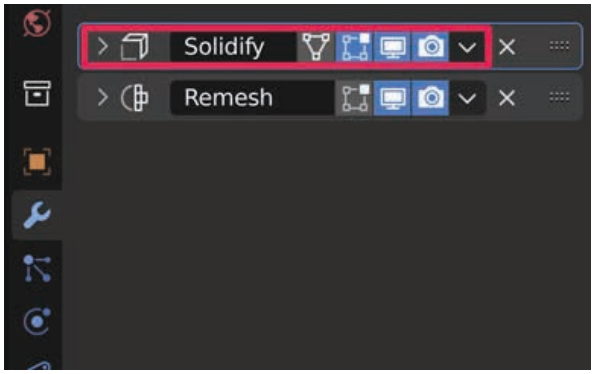
Using the remesh modifier on a flat plane

If you have a flat object with N-gons that you want to convert to quads, the Remesh modifier by itself doesn't really do anything.

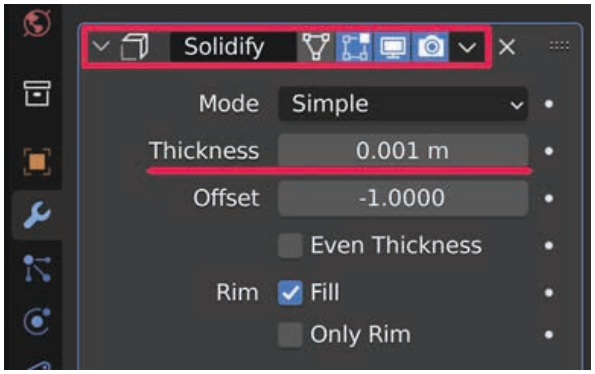
That's because it needs more geometry than just a flat plane to work. A flat plane simply disappears.



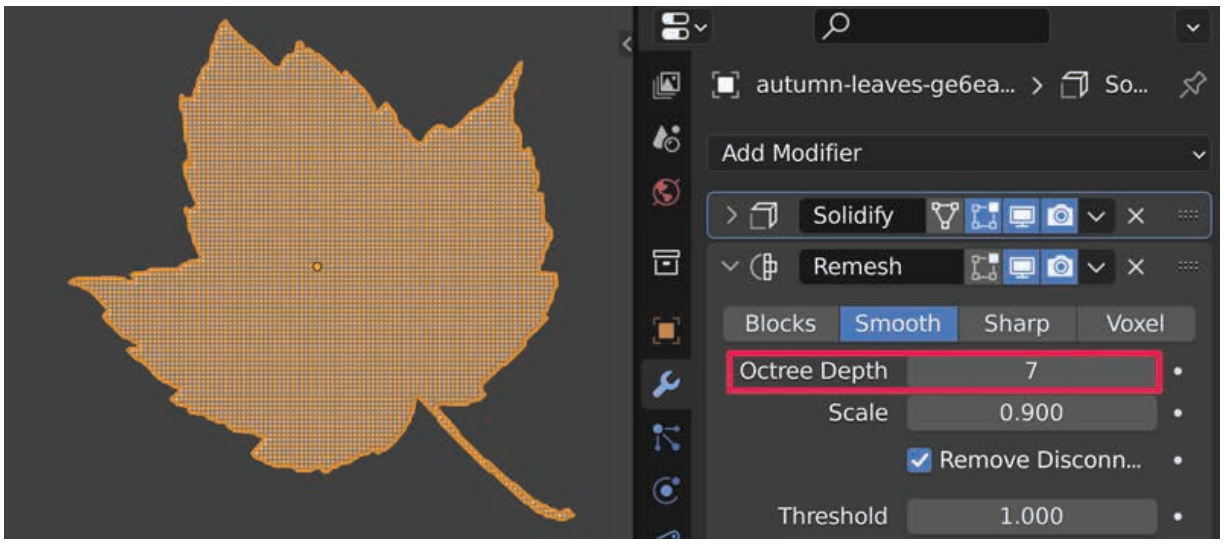
Turn on **Wireframe** in the Object Viewport Properties so that you can see better what's happening.



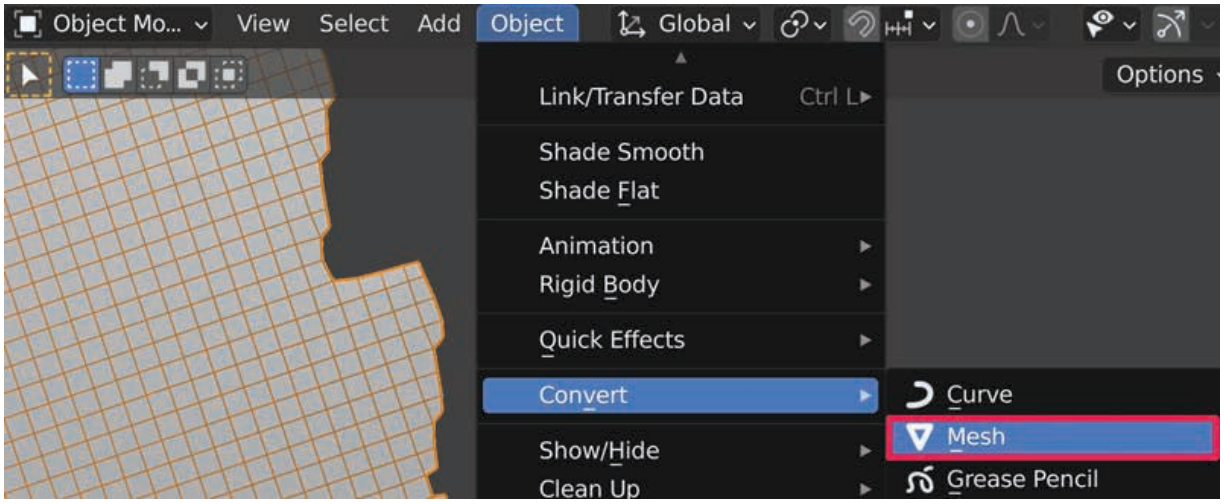
Add a Solidify modifier and set the **Thickness** very low, like to 0.001. Make sure it is placed before the Remesh modifier.



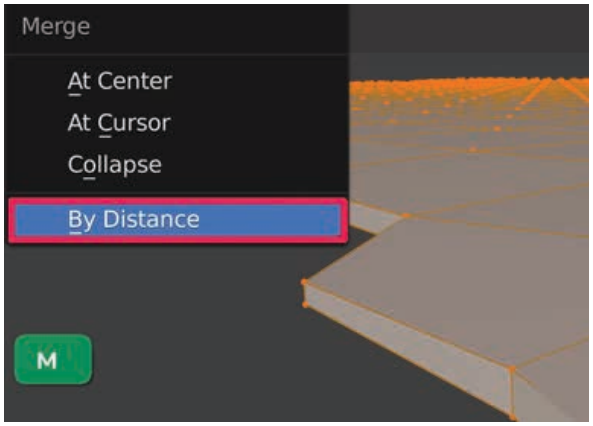
Change the Octree Depth until you get the amount of Faces that you need. Be careful with these kind of values: if you go too high, it might freeze Blender for a while. Try Octree Depth 4, then 5, 6 ... see how high you can go step by step.



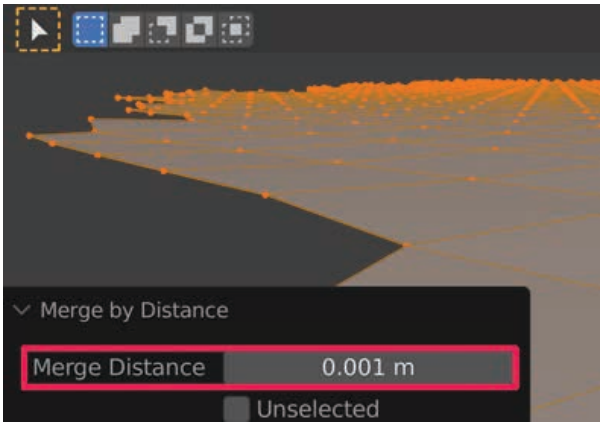
Convert it to Mesh to apply both modifiers at once.



In Edit Mode select all and **press M** and choose Merge > **By Distance**.



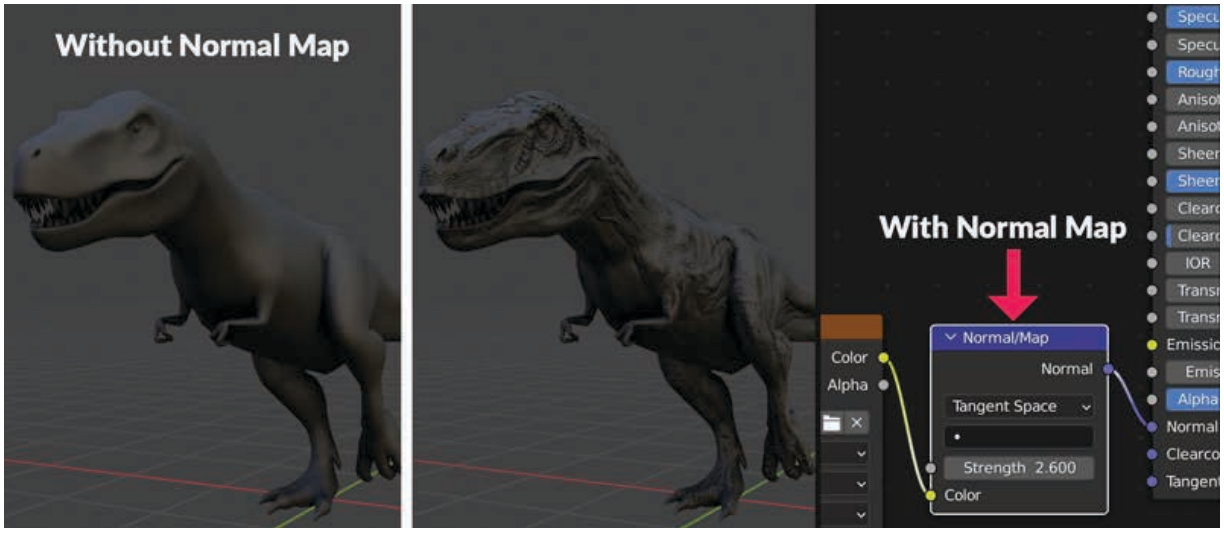
Increase the **Merge Distance** until the thickness disappears again.



In this example we used a leaf, but this can also be a very convenient way of remeshing text, which tends to have really bad topology.

Convert Normal Map to Geometry (for 3D printing)

Normal maps are a great way to add fine detail to a model. However, it's a visual cheat that doesn't actually change the geometry. Which is great for visual effects or games, but if you want to print the model out as a 3D print, you'll lose that detail.

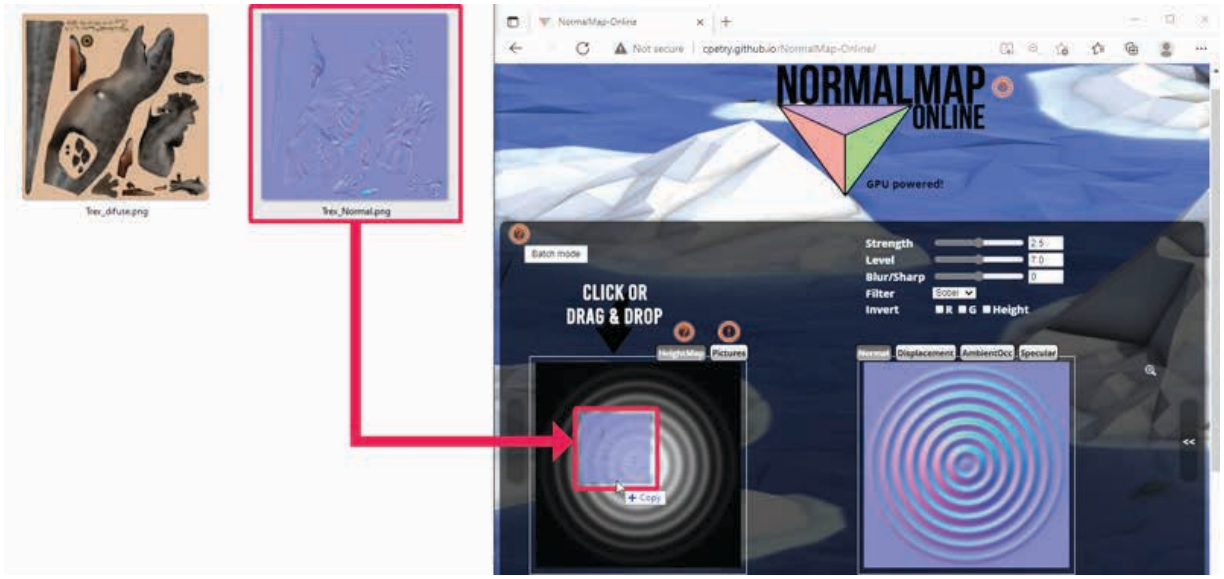


If you want to 3D print the model with the Normal map detail included as actual height differences, you'll need to convert the Normal map to 'real' geometry.

To do this, you'll first need to convert the Normal map to a **Displacement map**.

You can use the website Normalmap Online for this purspose. It's free to use.

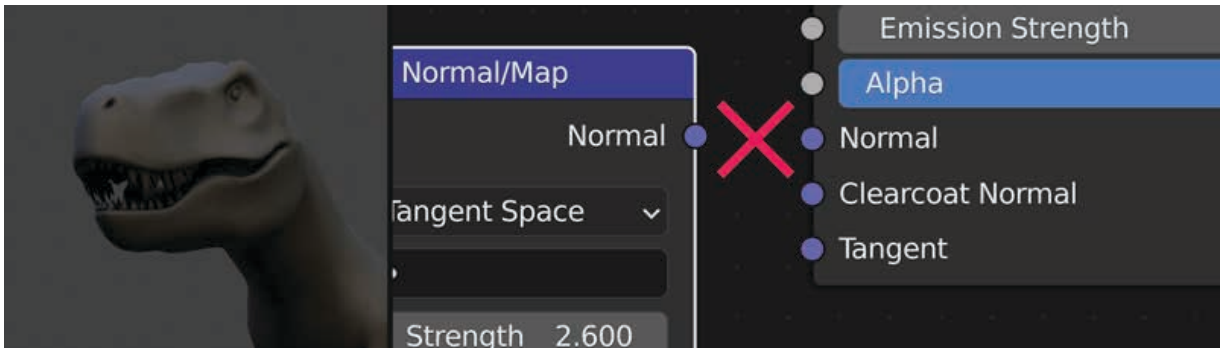
Drag the Normal map to the box underneath where it says **Drag & Drop**.



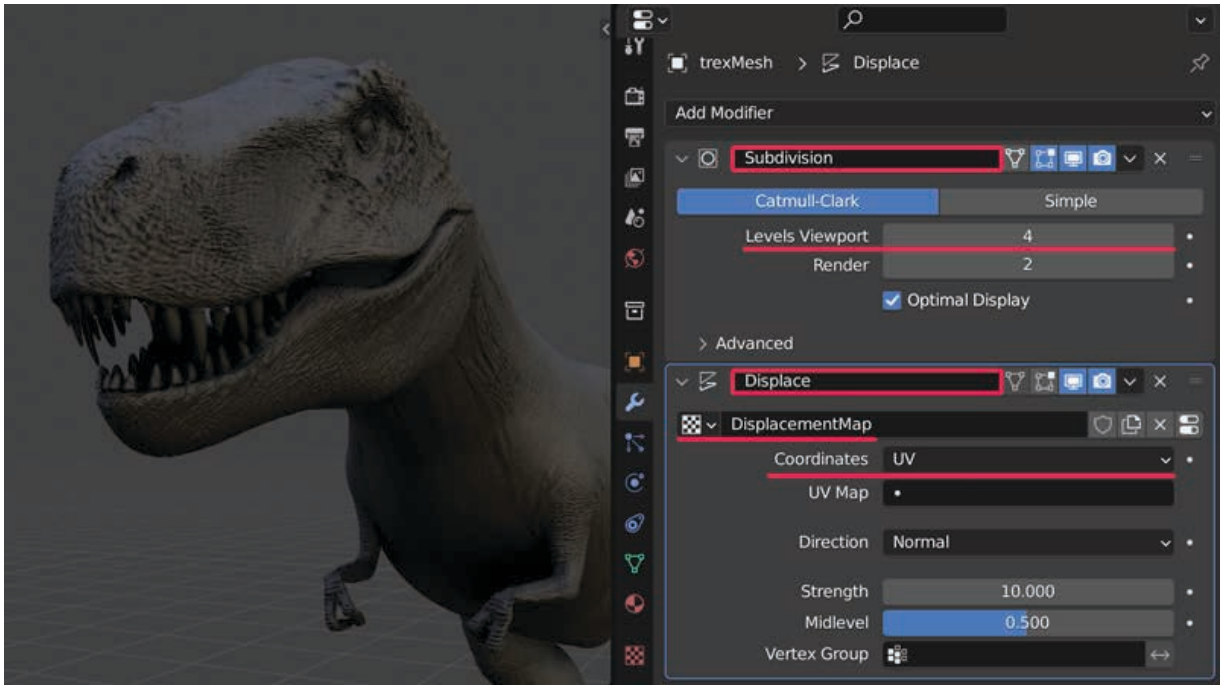
Click here to return to the index.

Then, download the resulting **Displacement** map by selecting it and clicking on **Download**.

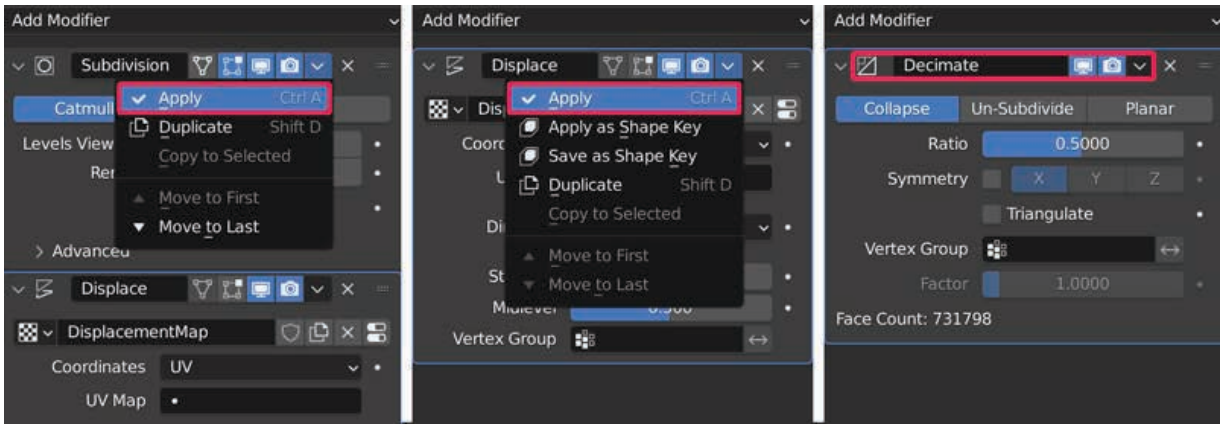
Now you can remove the Normal map from the material of the 3D object.



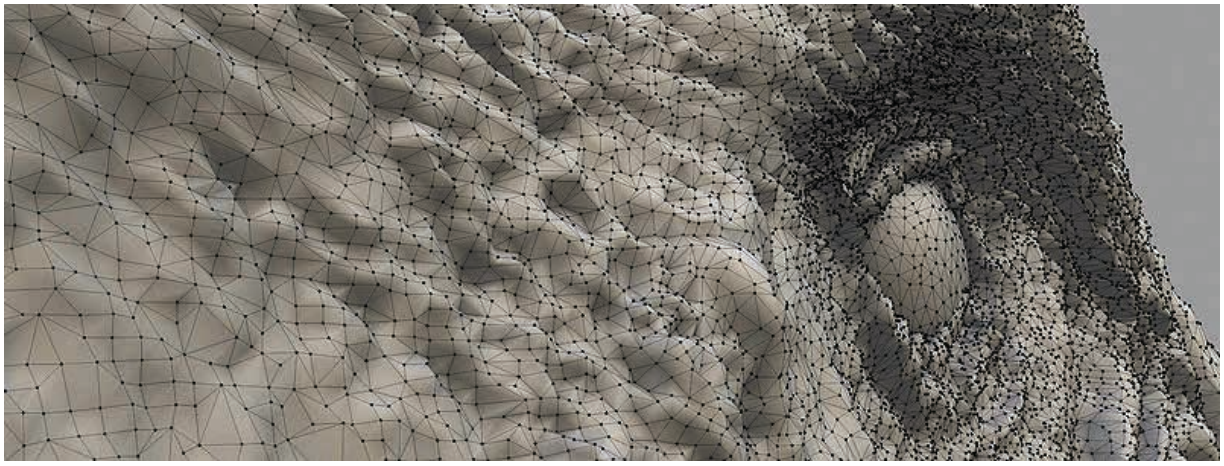
Add a **Displacement modifier** with the Displacement texture. As well as a **Subdivision modifier** with plenty of levels. Set Texture coordinates for the Displacement to **UV**.



Apply both modifiers by choosing that option from their drop-down menu, then add a **Decimate modifier**, then apply that. This will make the file less heavy for printing. Note that these steps might take a while, depending on your computer. You'll have to be patient. Eventually the process will finish.



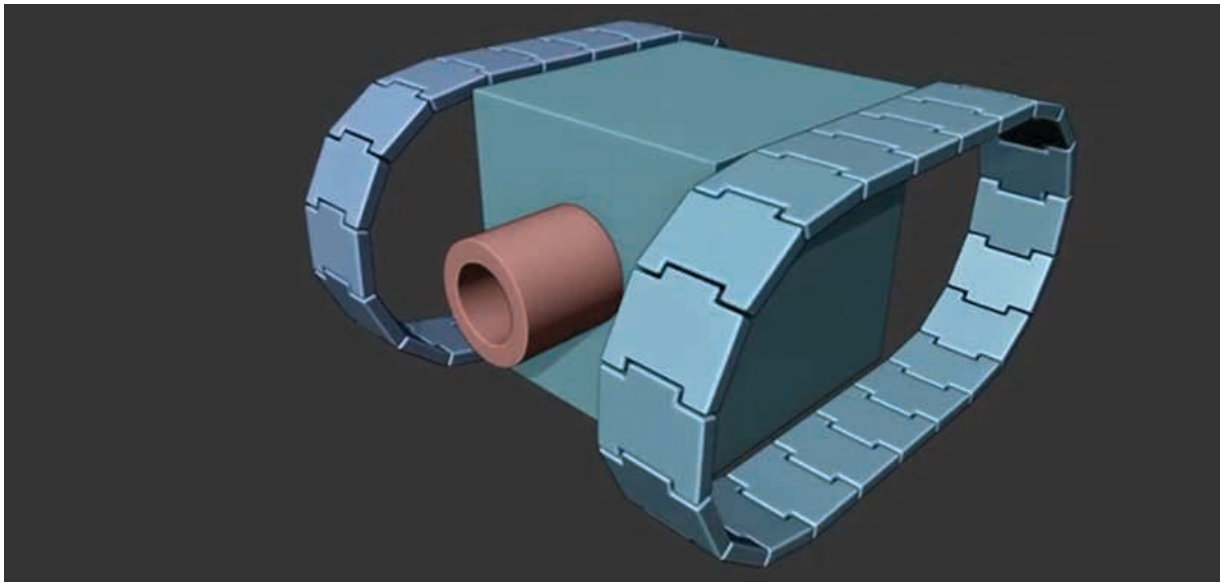
Now the detail from the Normal map is part of the geometry.



For 3D printing, these arms might be a bit too thin. So add some thicker arms instead.

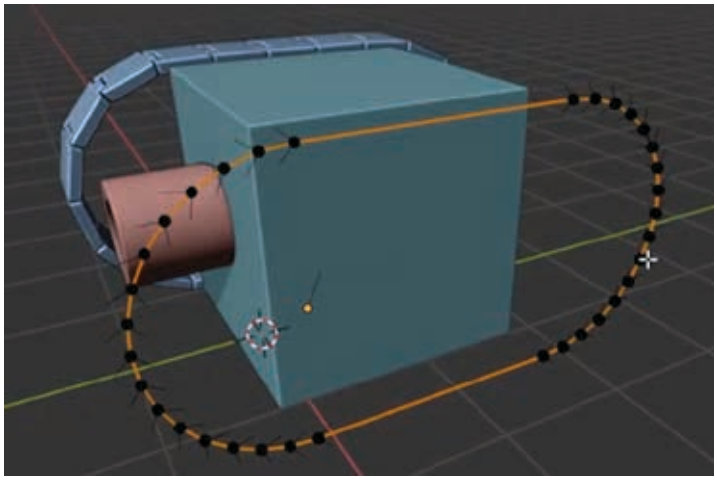
The default cubes strike back! (easy tank tracks)

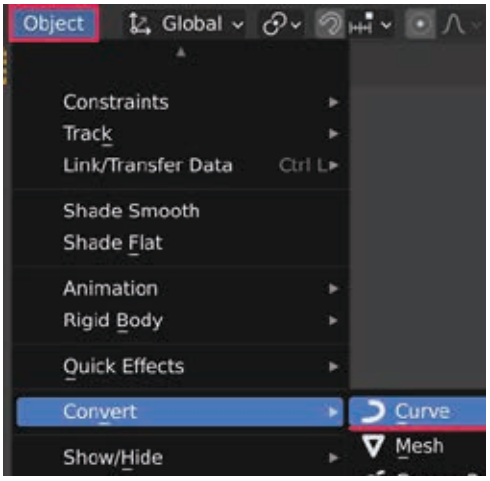
The default cubes are fighting back! Let's look at adding some simple tank tracks.



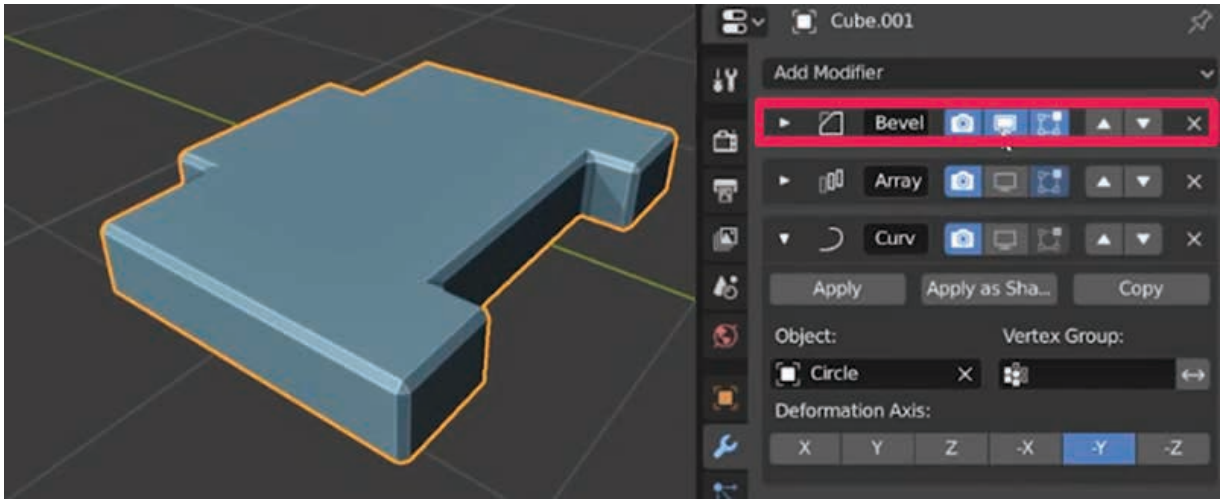
This tank track is actually easy to set up. You need a Curve in the shape of the track.

In this case I started with a mesh circle and then converted that to a Curve.



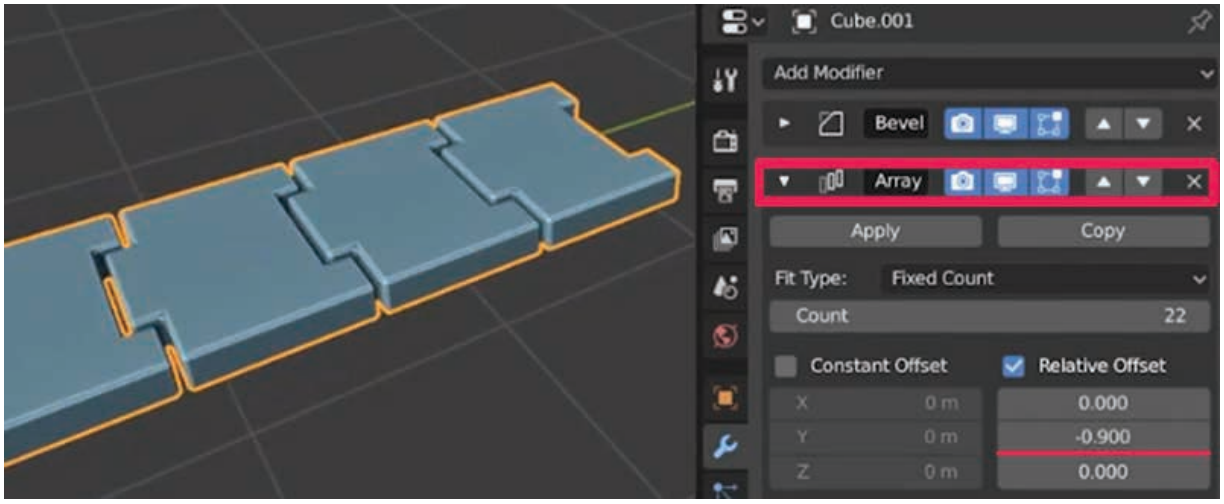


Then create one of the track parts and add a **Bevel modifier** for esthetic purposes.

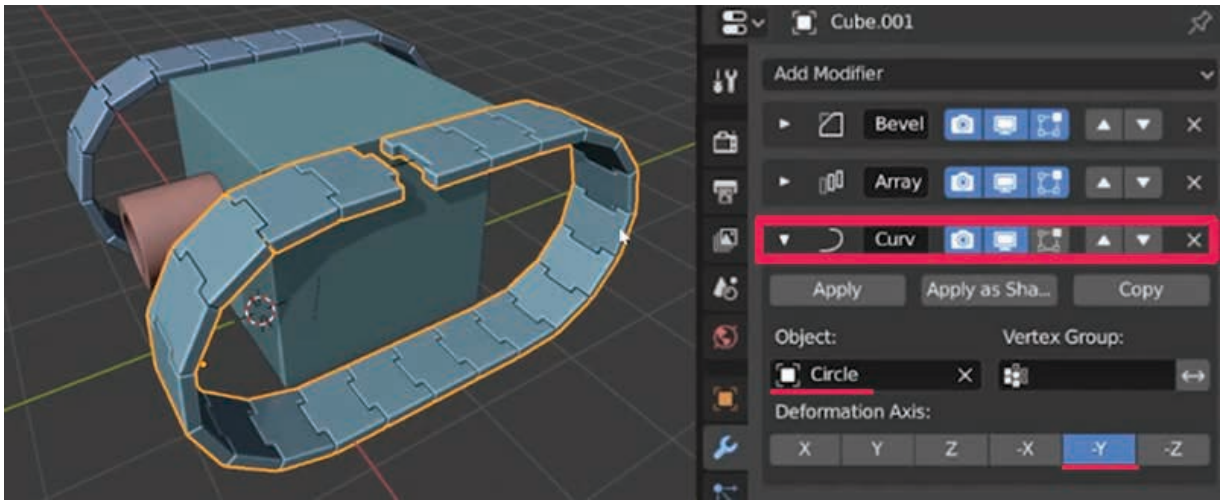


Click here to return to the index.

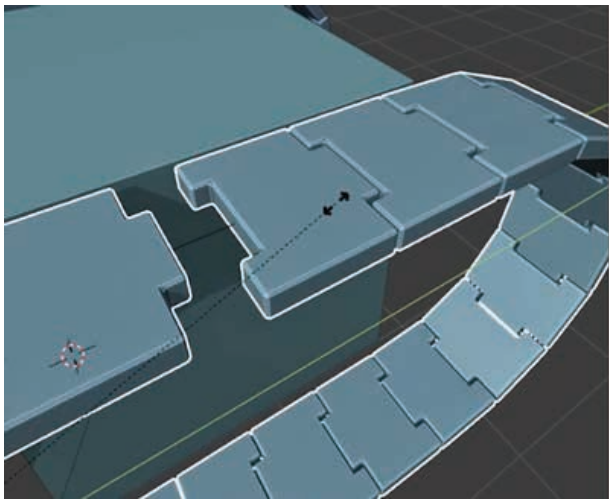
As well as an **Array modifier** set to the correct Axis (Y in this case).

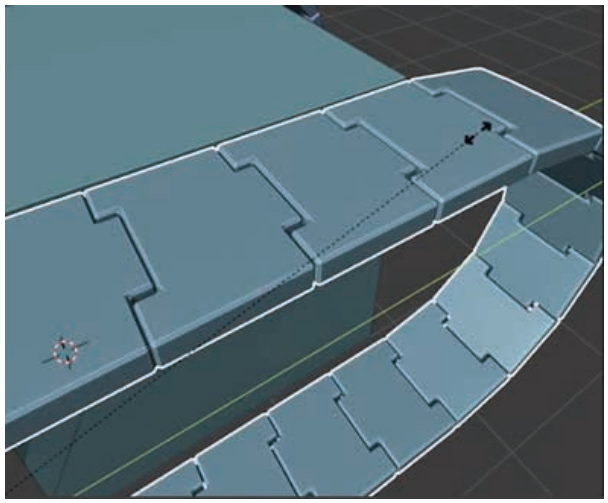


Also add a **Curve modifier** with object set to the Curve (named Circle, because it was originally converted from a Mesh Circle).

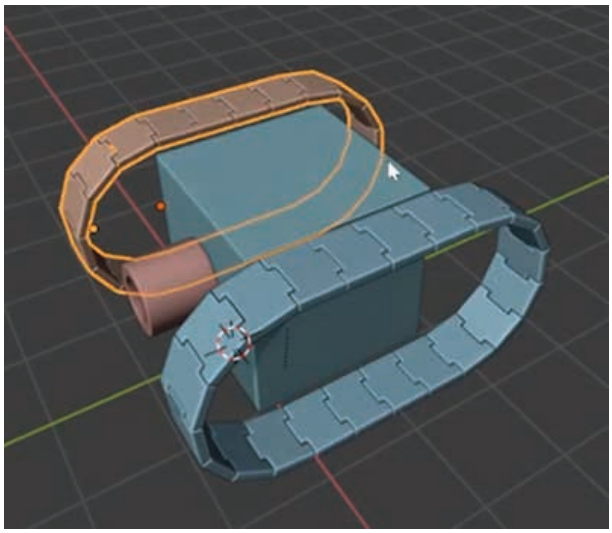


Increase the array **Count** and scale the track by **pressing S** so the loop is closed.

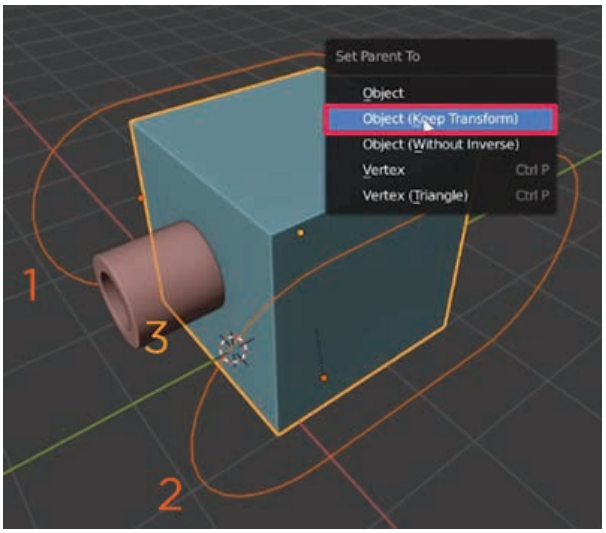




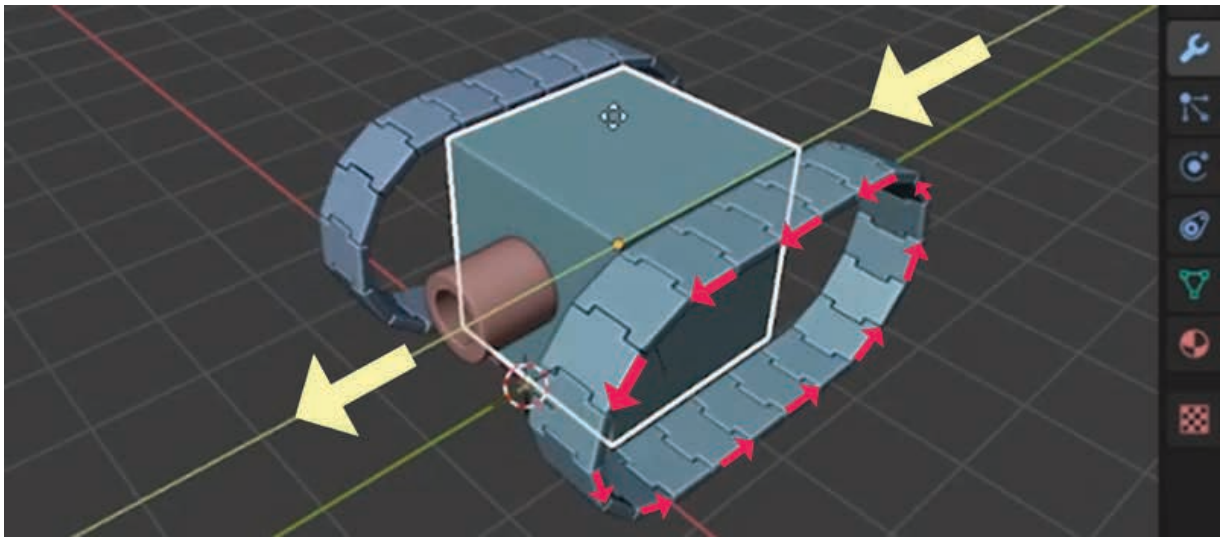
I created the track shape with an N-Gon on purpose, so that it won't deform along the curve. Duplicate the curve and the track to the other side with **Shift+D**.



Parent both curves to the default cube with **Ctrl+P** > Object (Keep Transform).

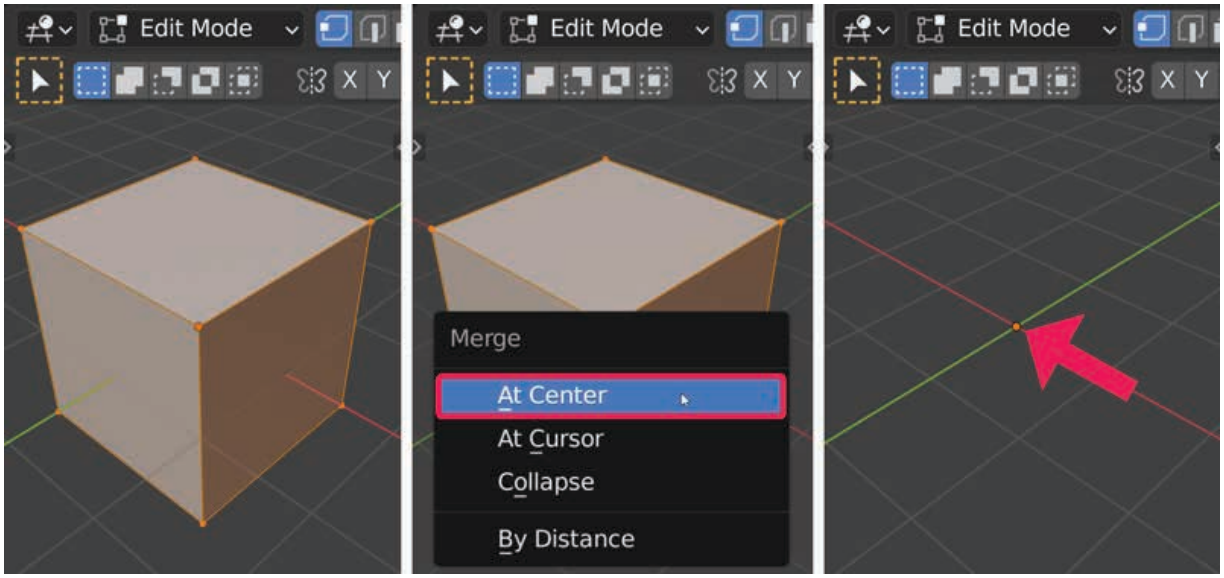


Now when you move the cube the track will move as expected.

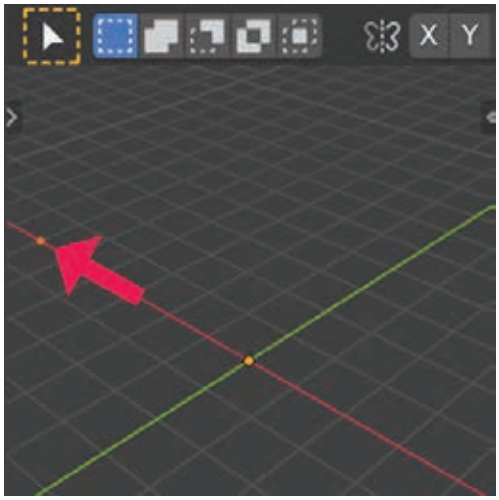


Circular Array using Screw modifier

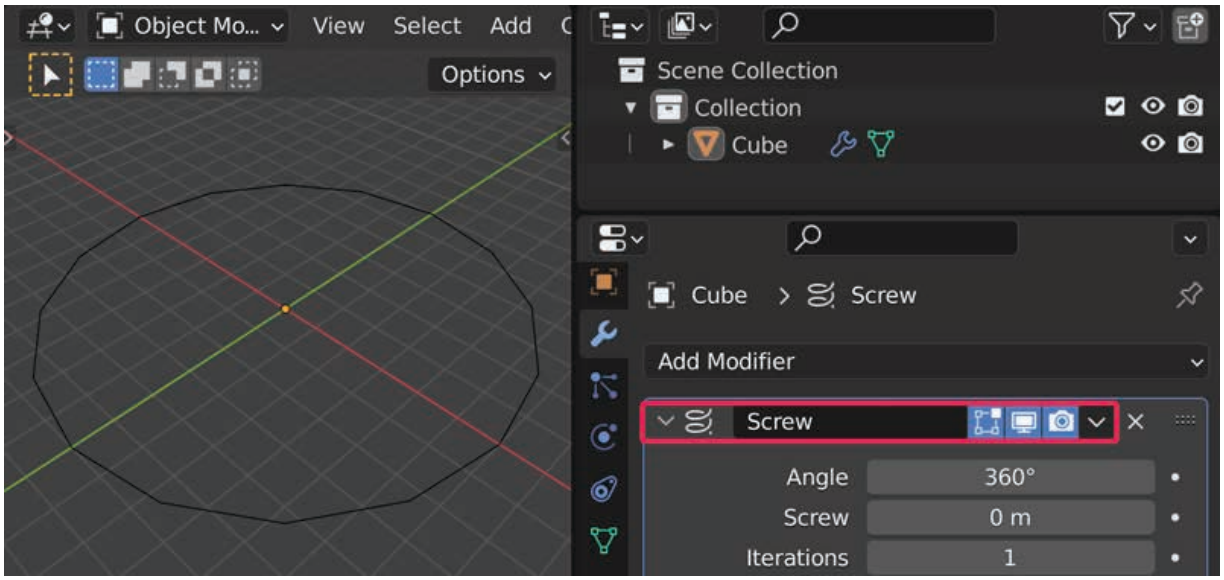
Select the default cube and go into Edit Mode by **pressing Tab**. Then **Press M** and choose **To Center**. Now you have one vertex.



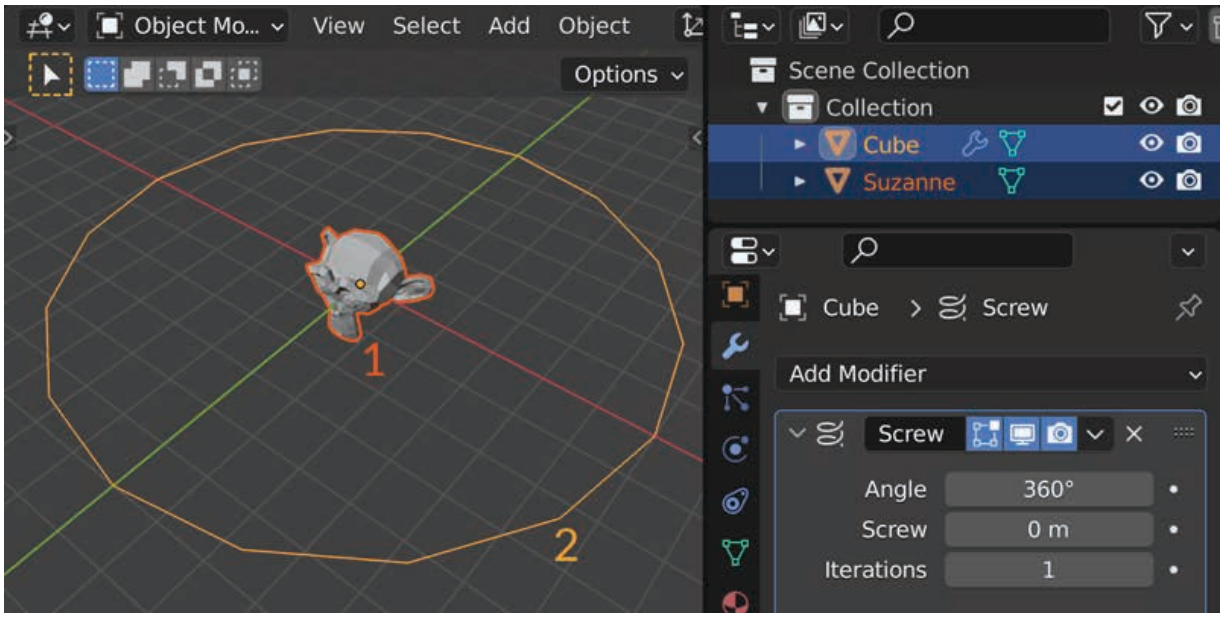
While still in Edit Mode, make sure the Vertex is selected by **pressing A**, then move it away from its origin.



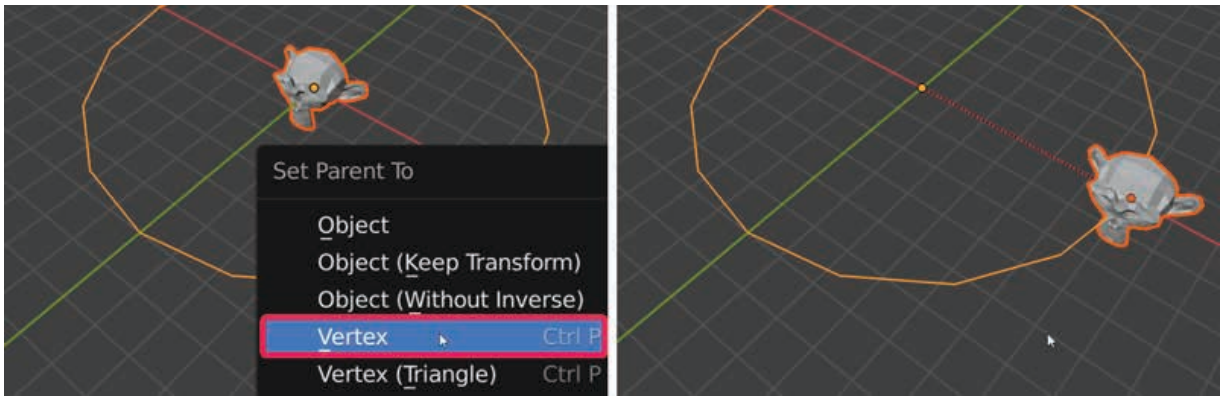
In Object Mode, add a **Screw modifier** to it. Now you've got a circle.



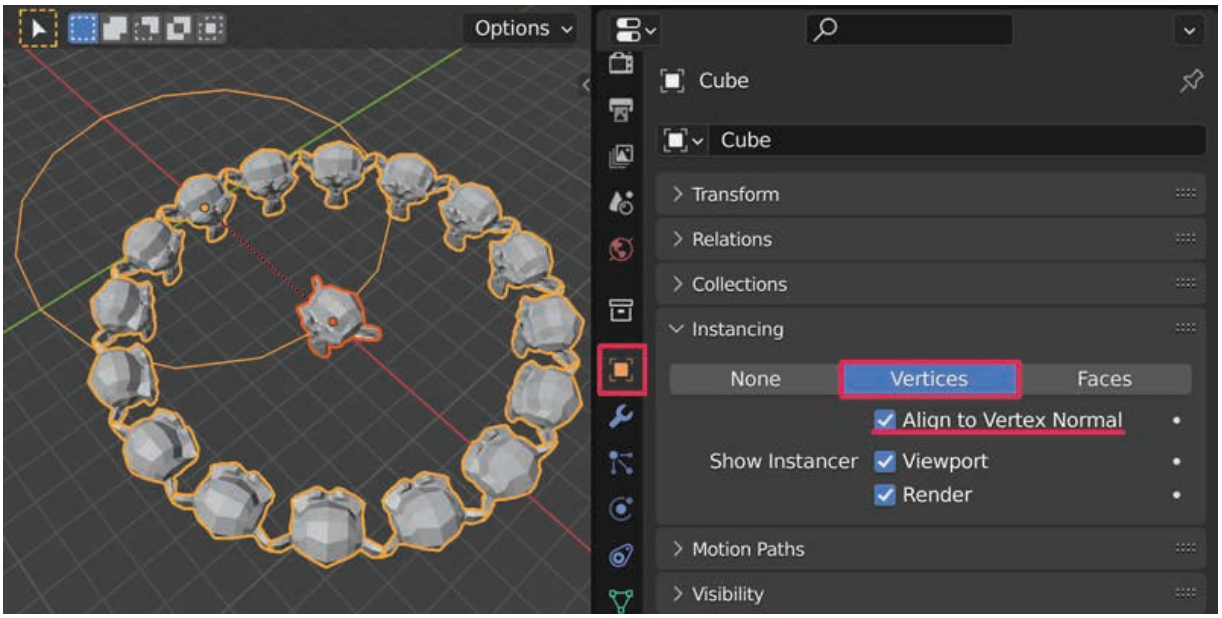
Now select the object that you want to have in an array and **hold Shift** and select the vertex.



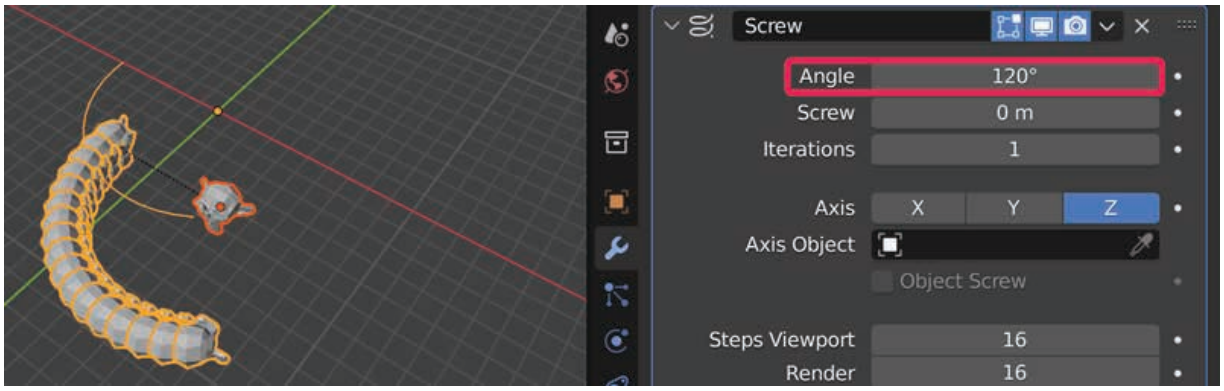
Then **press Ctrl+P** to parent to vertex. It might be easier to select these things in the **Outliner** than in the viewport.



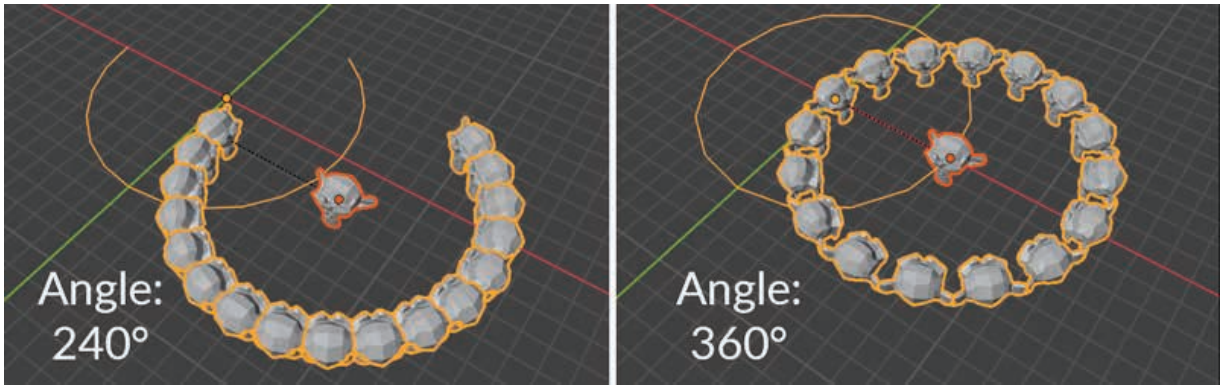
With the vertex selected, under **Instancing** choose **Vertices** and enable **Align to Vertex Normal**.



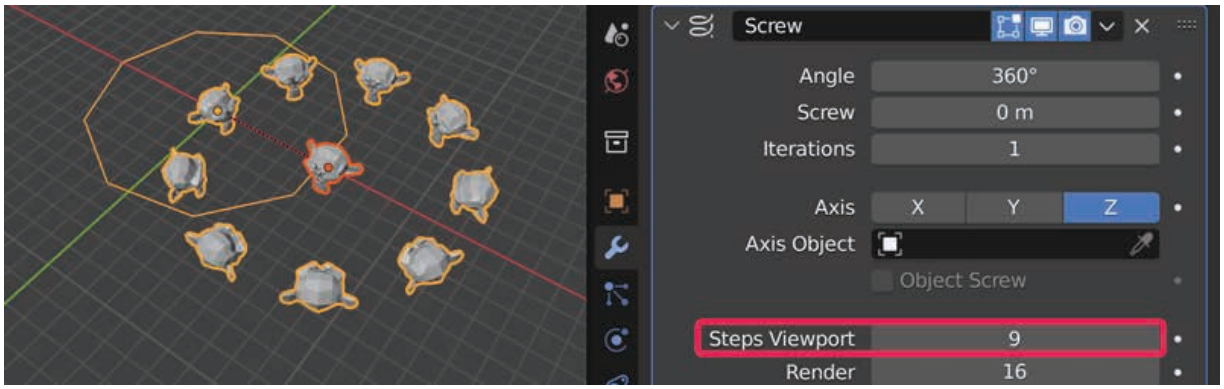
Now you can control the size of the array by moving the vertex in Edit Mode. The Screw modifier defaults to 360 degrees and 16 steps.



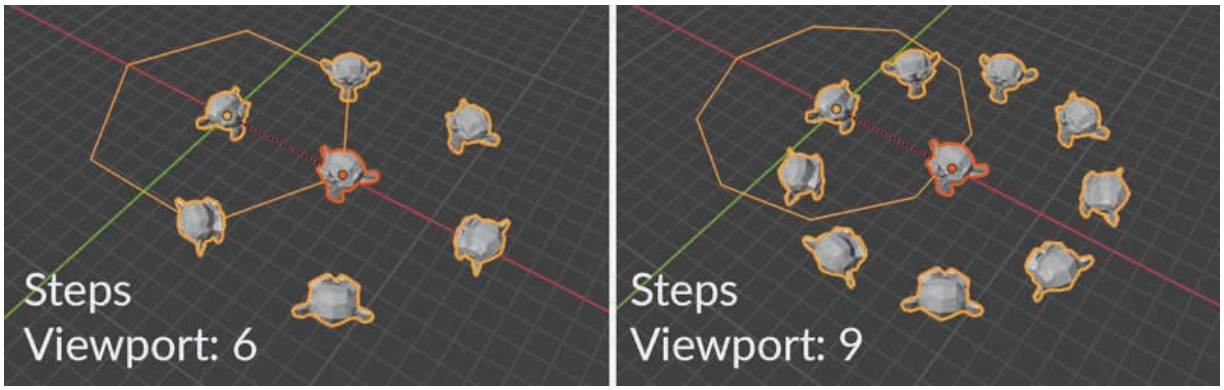
You can change these values as needed.



By changing the **Steps** value, you change the amount of instances.

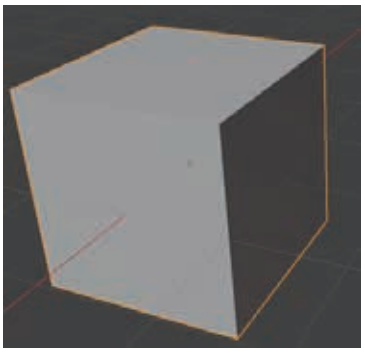


For example, 6 steps or 9 steps:



Circular Array using Instancing

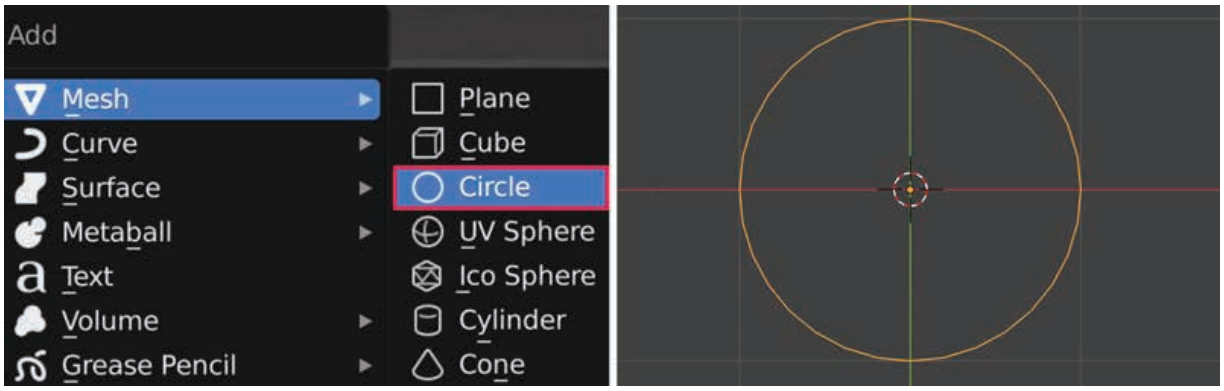
If you know exactly how many duplicates you want on an array, this is an easy and fast trick.



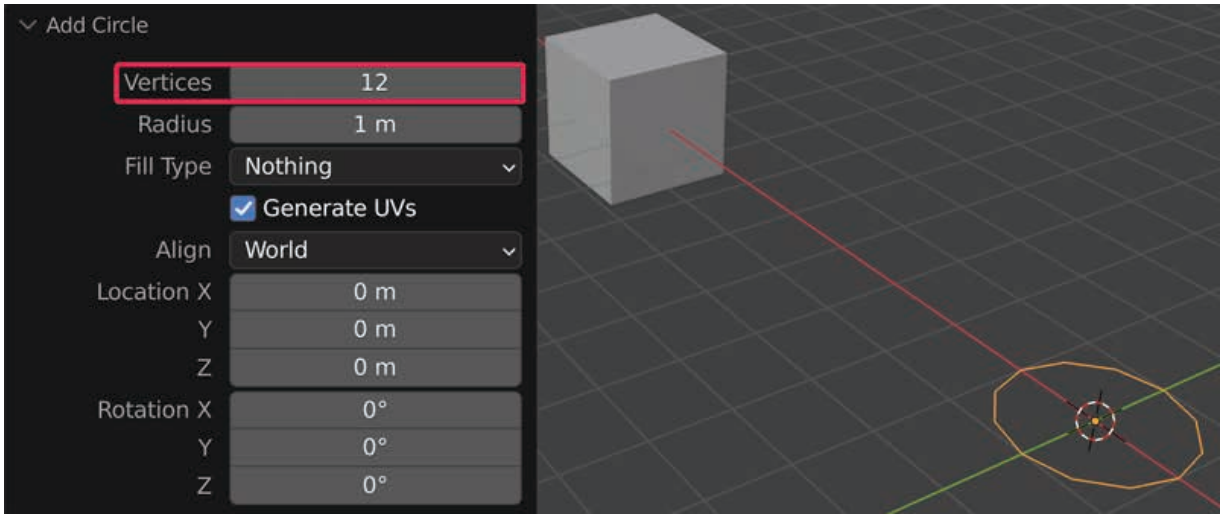
First, create the object you want to have in an array.

For this example we'll use the default cube. Scale it down to 20% of its size by pressing S+[0.2 on the numpad].

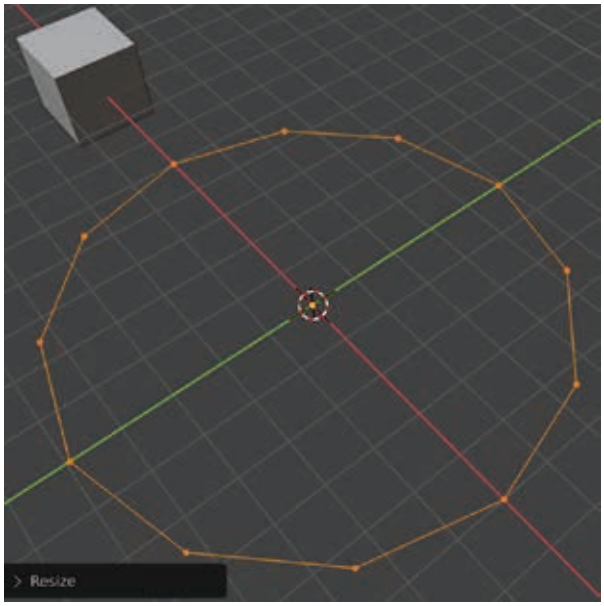
Create a Mesh **Circle** - not a Curve!



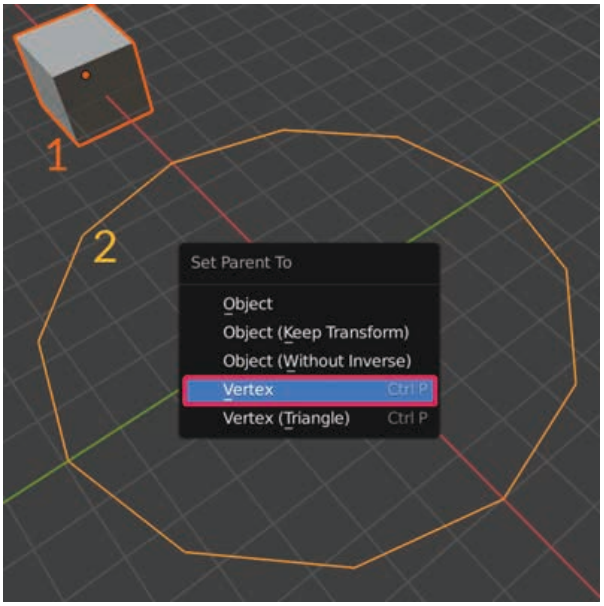
Set it to have the amount of **Vertices** that you want the array to have 12 for example.



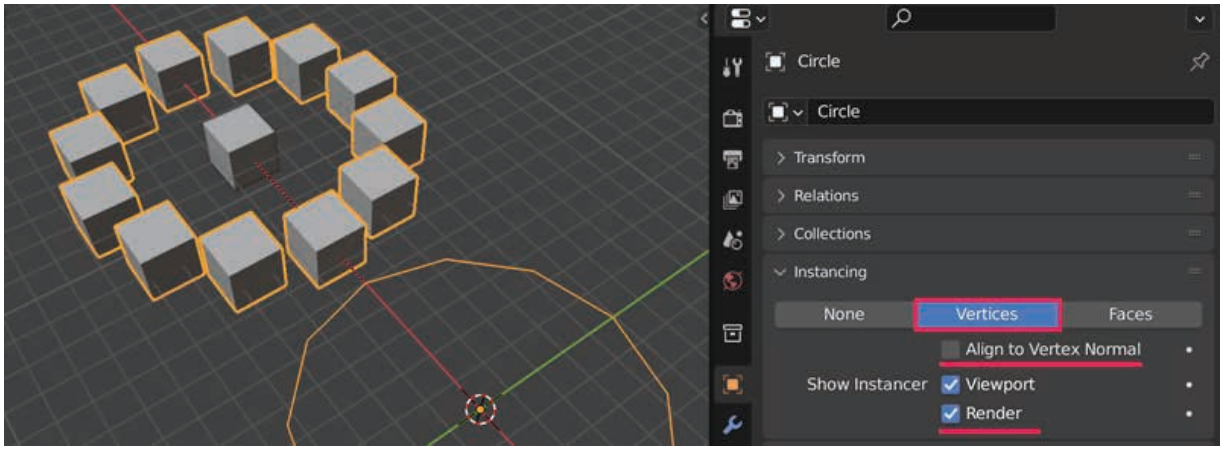
Scale the circle to the size that you want the array to have.



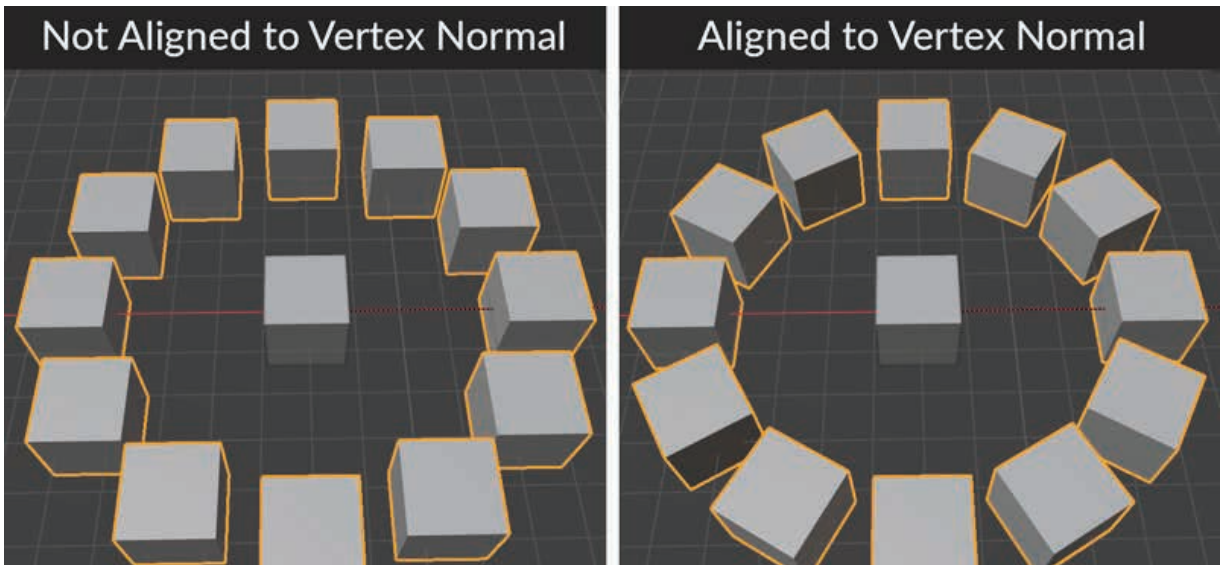
Now parent the object to the circle with **Ctrl+P** and choose parent to vertex.



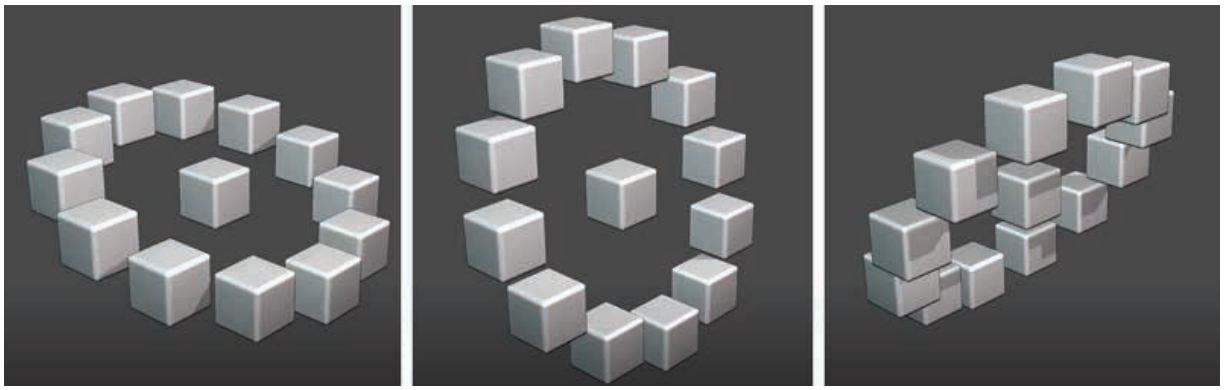
Select the Circle and under **Instancing** choose **Vertices**. You can choose whether to align the objects to the circle or not, and if you want to have the original object show up in renders or not. Select and scale the original cube down if the instances are too big.



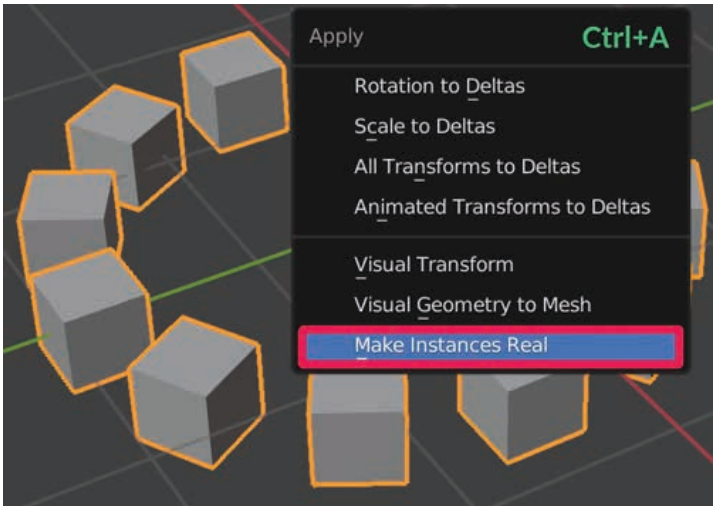
Below you see the difference between not aligning or aligning to Vertex Normal.



Now you can still manipulate the circle. If you don't align the objects to the circle, you can create some interesting animation like this…

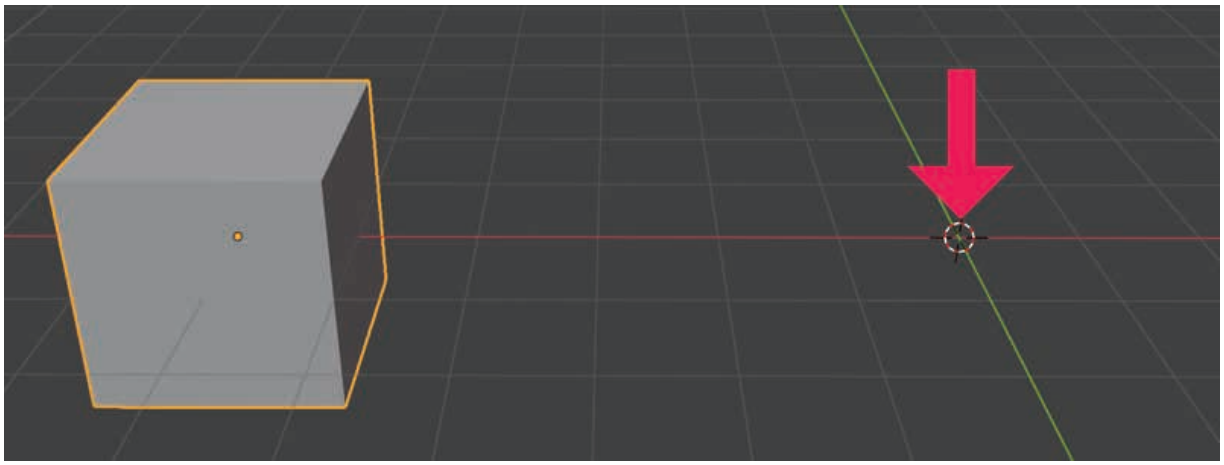


In case you want to make these instances into real mesh objects that you can transform individually, select them Circle with the instances in Object mode, **press Ctrl+A** and choose **Make Instances Real**.

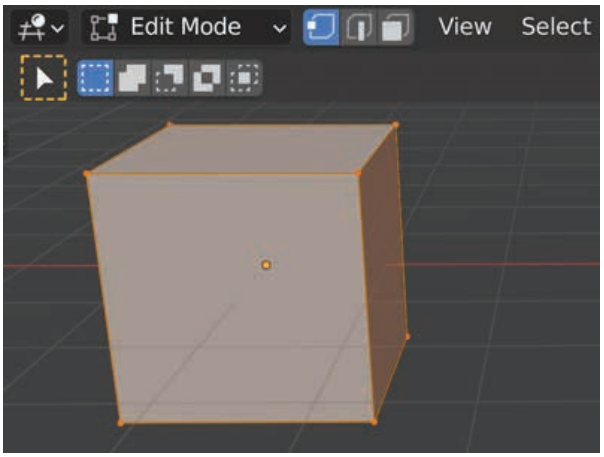


Fast Circular Array

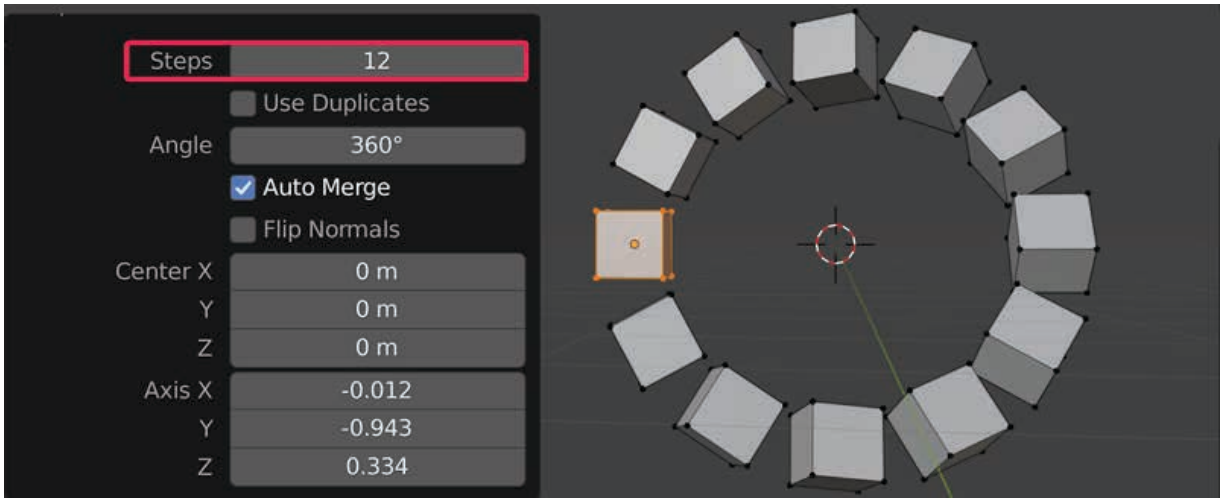
Place the 3D cursor where you want the center of your circular array to be.



In edit mode, select the whole mesh by **pressing A**.



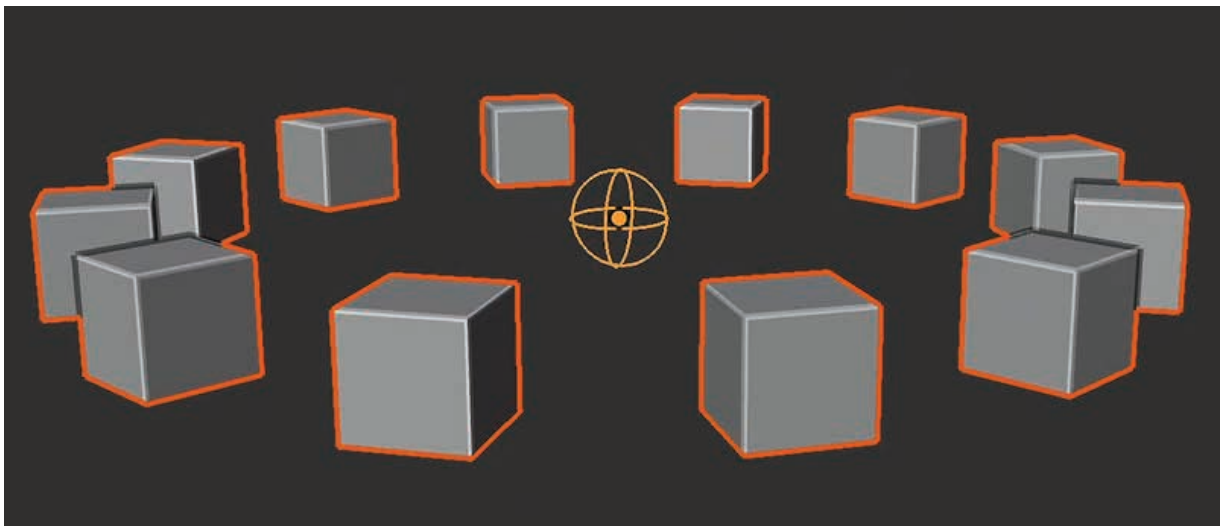
Then **Press Alt+E and S** to create the array. You can then use the menu to change the amount of duplicates on your array.



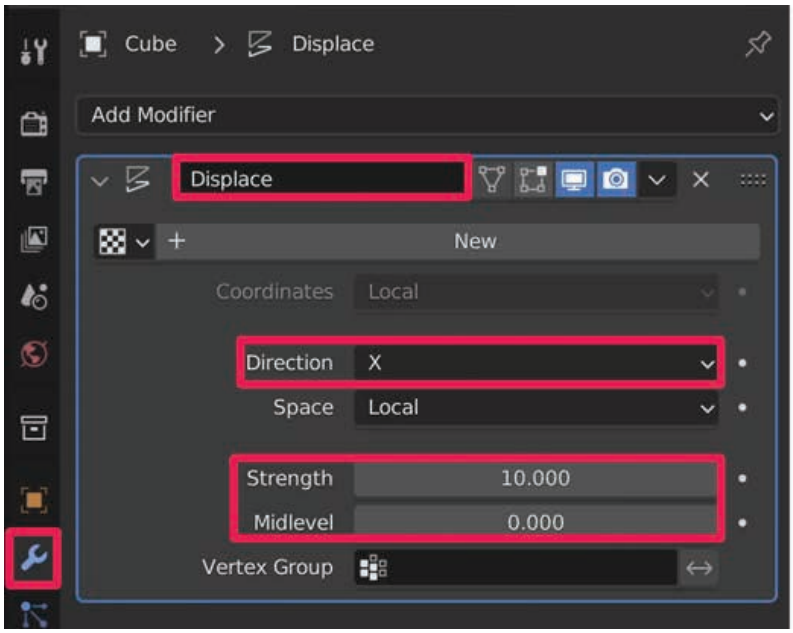
Some day, I'm sure Blender will have a simple "Circular" checkbox in the Array modifier! But until then, you have the above methods to get it done.

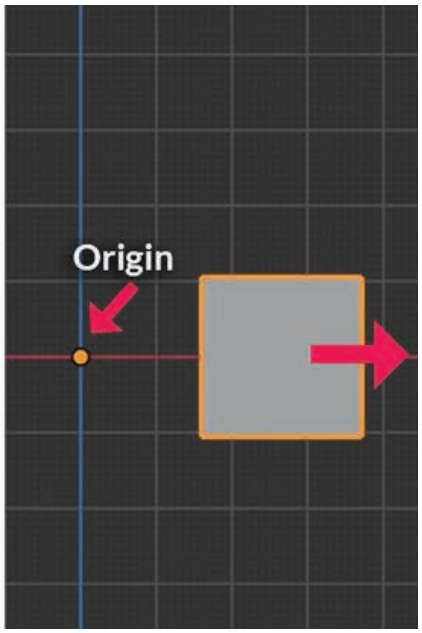
Circular Array using Empty

This method of creating an array uses an **Empty** to control rotation, as well as two modifiers that make it a fairly non-destructive method.

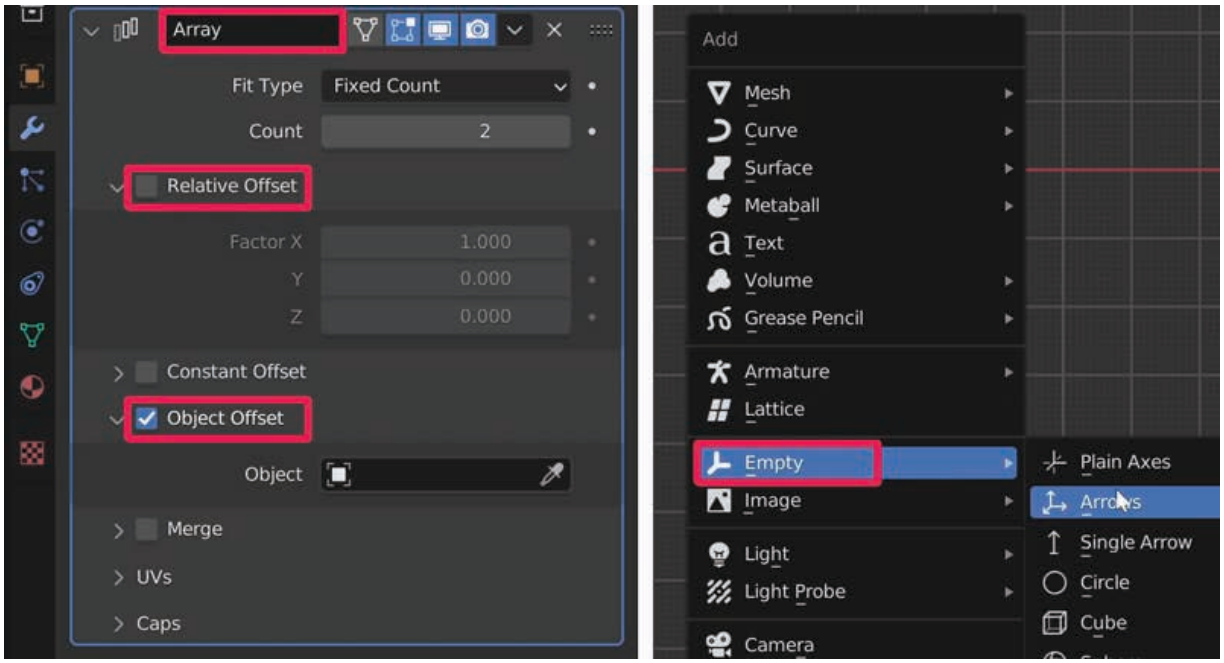


Add a **Displace modifier** to your object and set **Direction to X**. Decrease **Midlevel** to zero.. Now you can non-destructively move the object away from its origin by increasing the **Strength** value.

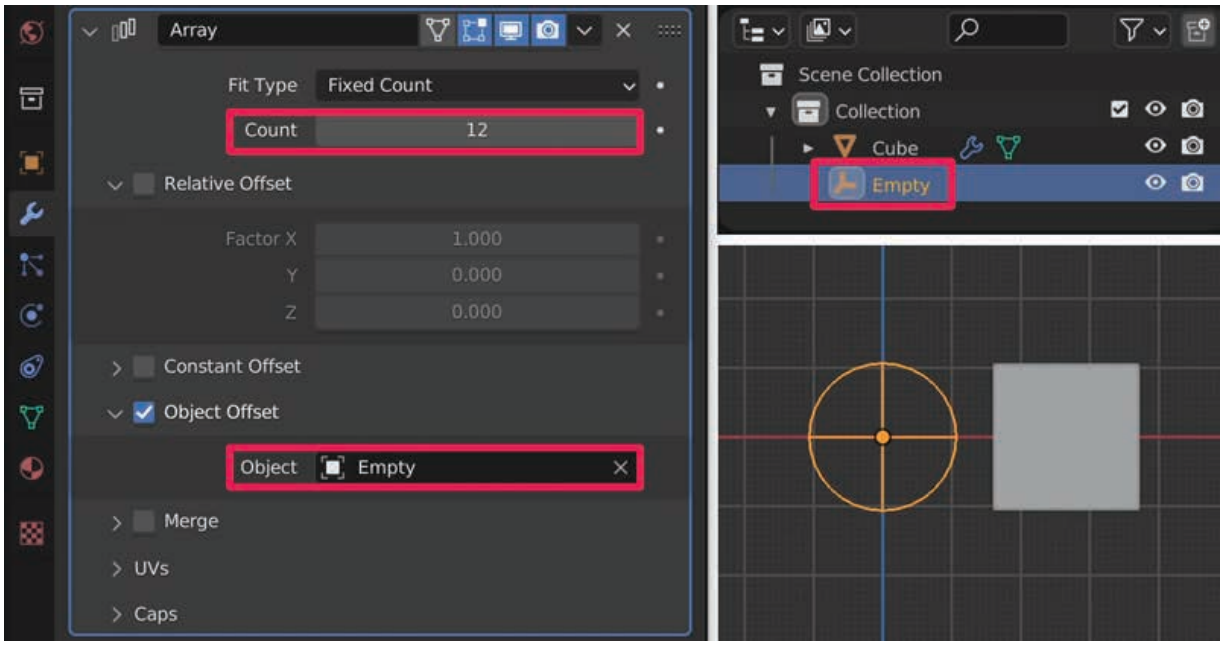




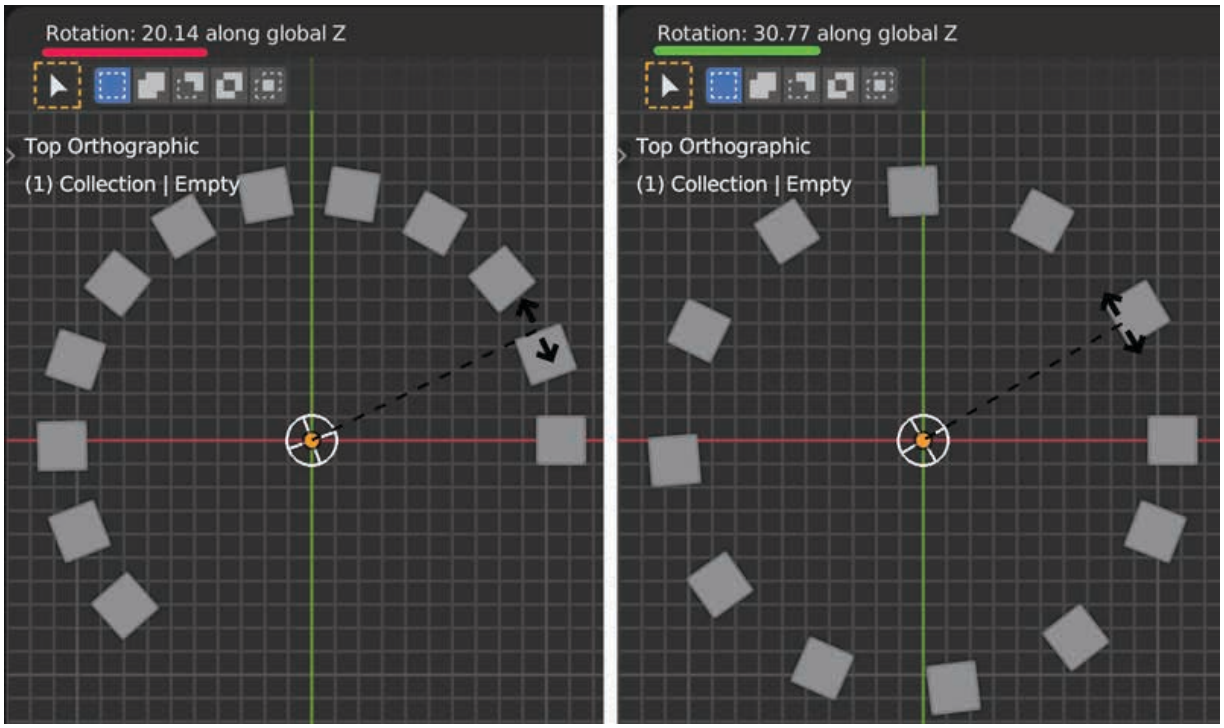
Next add an **Array modifier**. Enable **Object Offset** instead of Relative Offset. Add an **Empty** by **pressing Shift+A** and choosing any kind of Empty.



In the Array modifier use the Empty as the Offset **Object**. Increase the **Count** to the amount of instances you want. Select the **Empty** in the 3D viewport or in the Outliner.

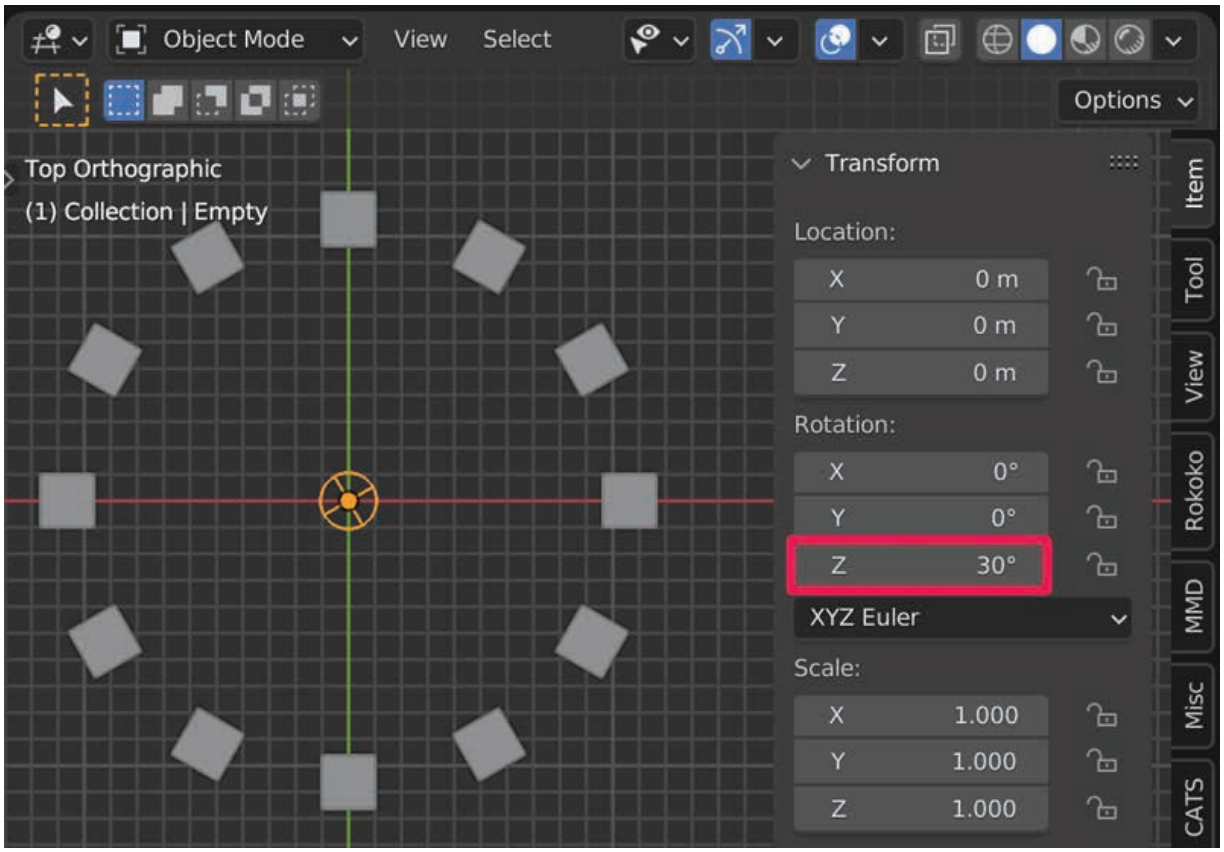


Press R and Z to rotate along the Z-axis, and eyeball the rotation until it looks okay. You can see the rotation value in the top left corner.



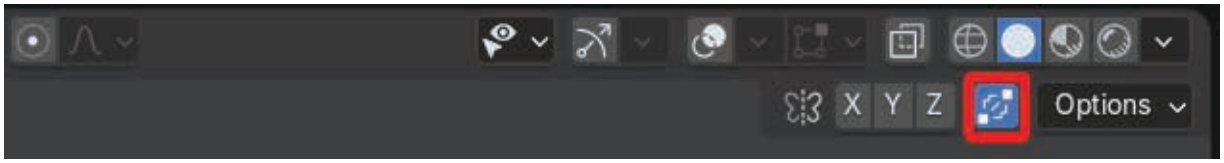
Round that number off and type it on the Numpad, followed by **Enter**.

So in this example, type 30 on the Numpad and <Enter>. You can also type the Rotation value in the Transform field, in the Option panel. If you divide 360 by 12 you get 30, so that's how you can calculate how much rotation you need for your amount of instances. Personally, I prefer to eyeball it!

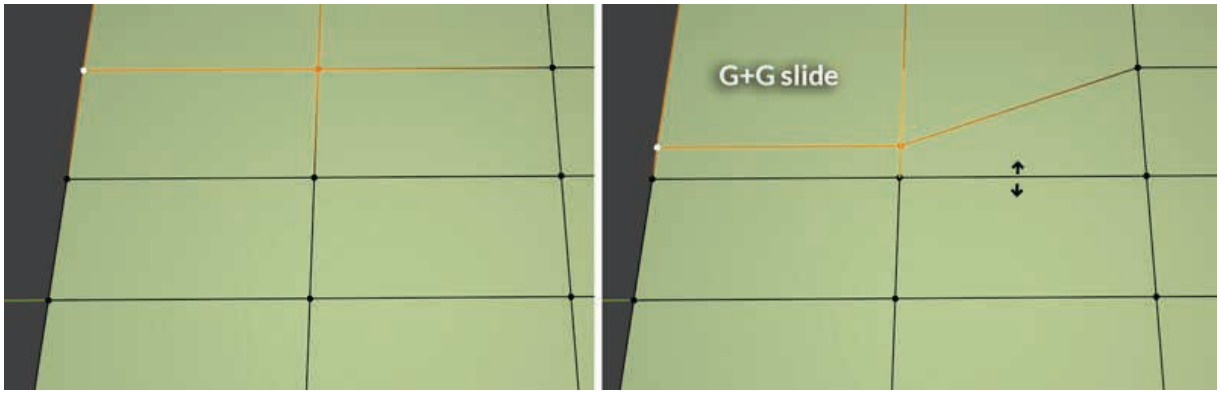


Auto Weld Vertices

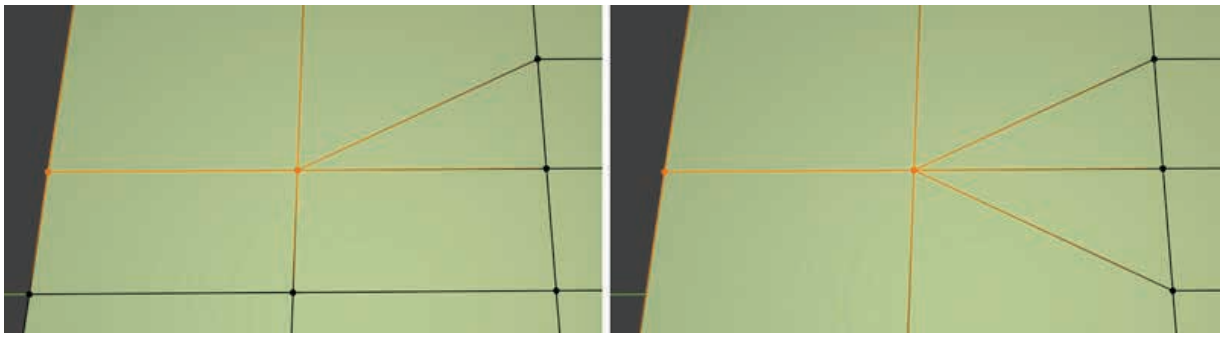
Instead of manually merging vertices together one option is to turn on **Auto Merge Vertices**. To activate it, click on its icon which is available in **Edit Mode.**



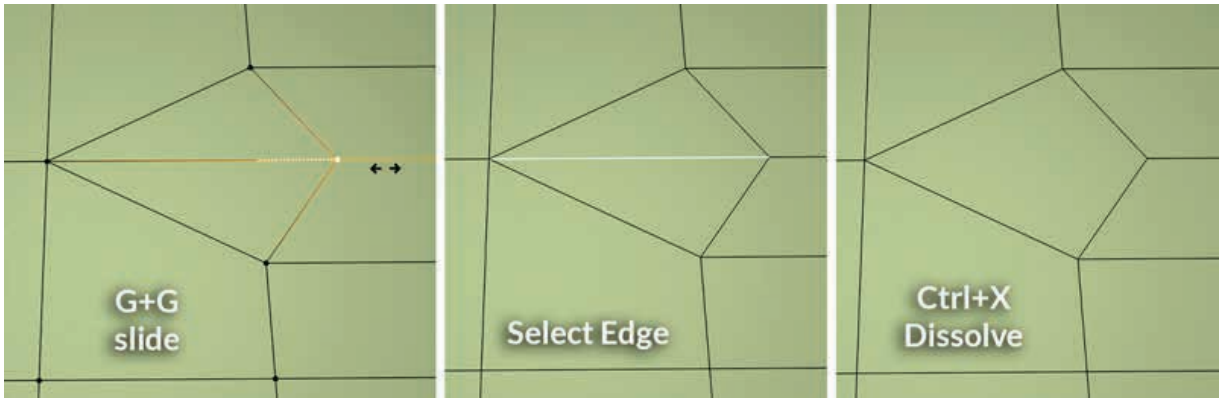
As an example of how you can take advantage of this, here we do a reduction of 4 to 2 quads. First, select Vertices or an Edge, and press G twice to slide it towards another Edge.



As soon as they're close enough together, they will merge automatically. Do the same for the other side, until you get the result on the right:



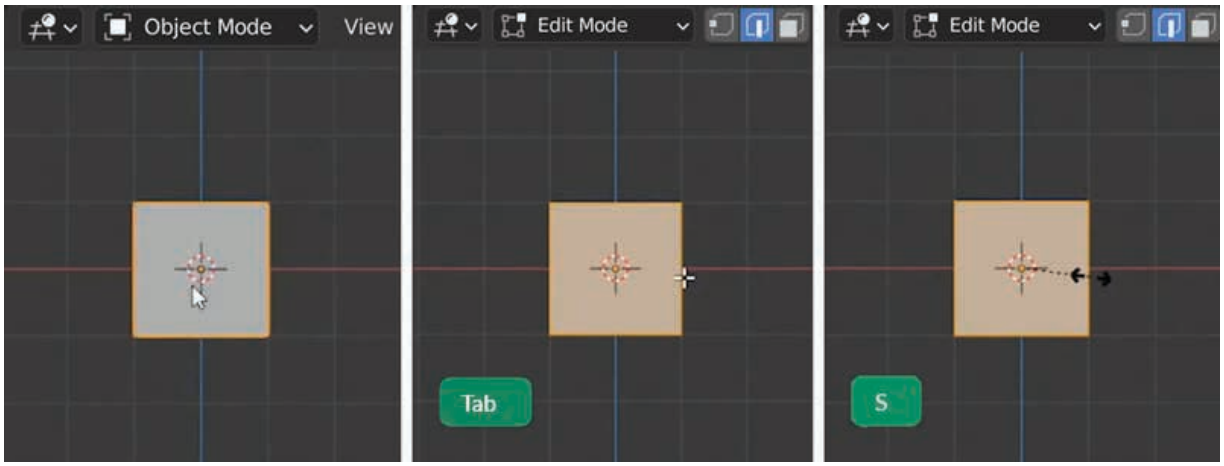
Now you can move a Vertex to get a more square-shaped quad, and then dissolve the Edge in the middle by selecting it and pressing Ctrl+X.

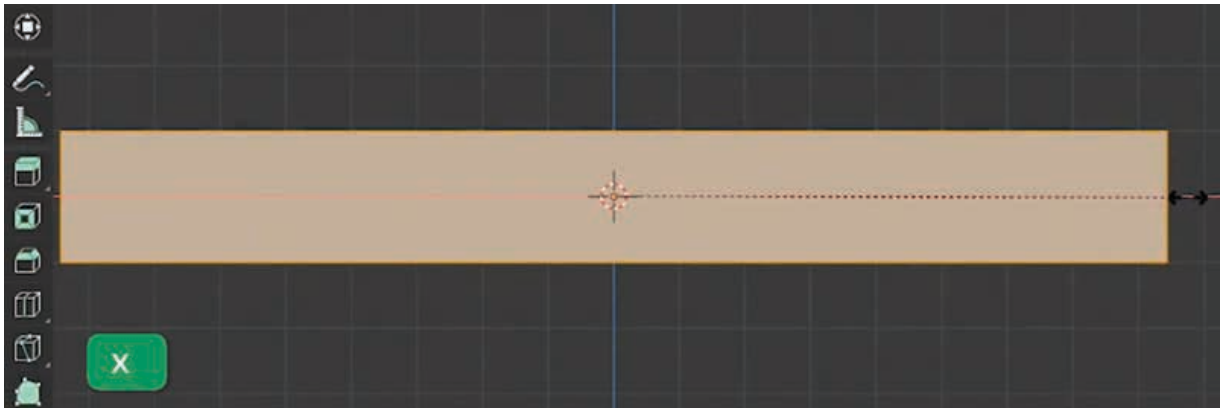


One way to quickly move two vertices together is by selecting both and scaling them to the same point by **pressing S + 0** (zero).

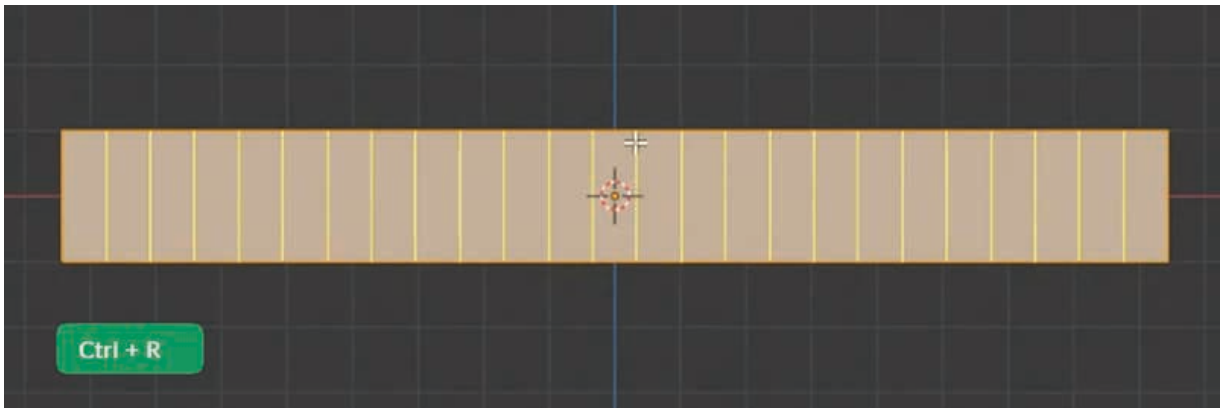
Make an arch using the Warp tool

To make an arch, **press Tab** to go into Edit Mode and scale the default cube along the X-axis **pressing S and X** and moving the mouse.

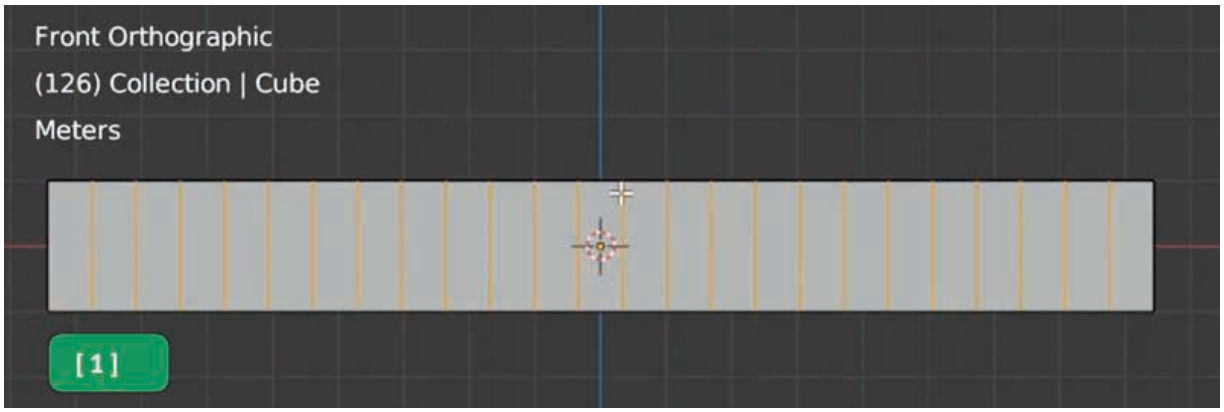




Add some loop cuts by **pressing Ctrl+R** and scrolling the mouse wheel up.

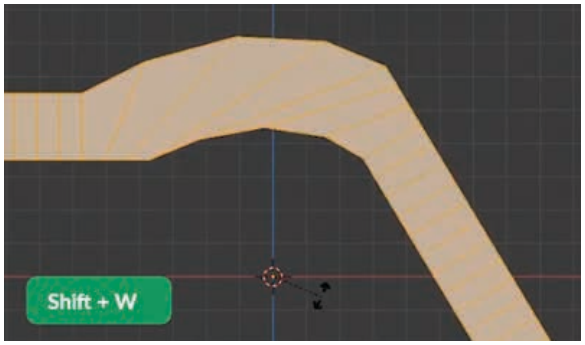


The result of the **Warp Tool** depends on the position of the **3D cursor** as well as the view, so make sure you are in **Front Orthographic view** by **pressing 1 on the numpad**.

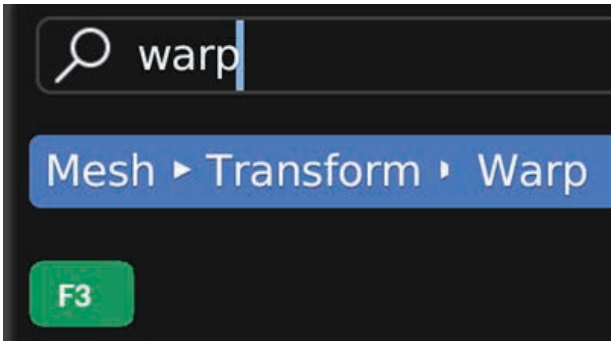


Move the mesh up by **pressing G and Z** and moving it up.

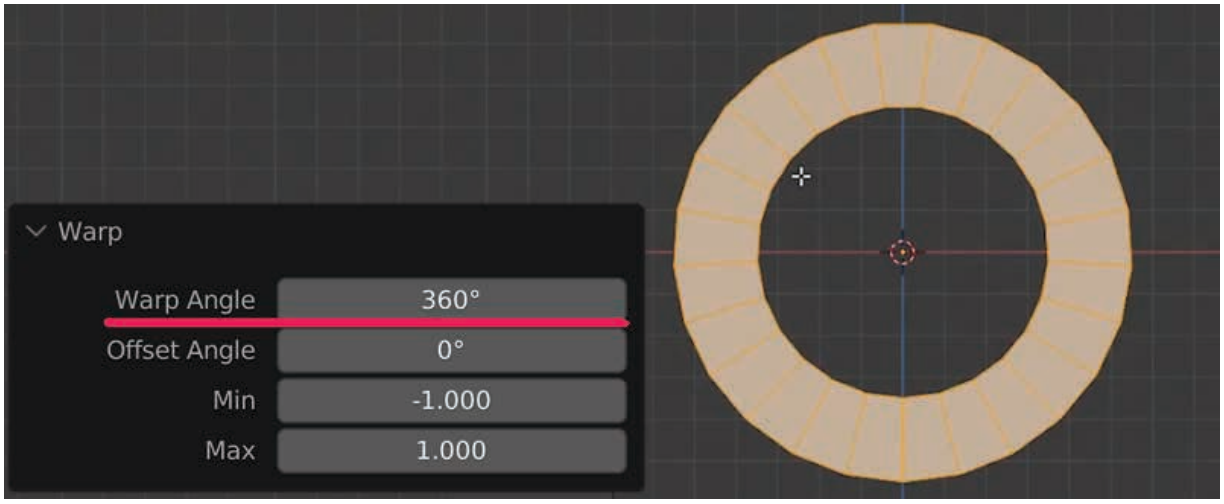
You can start warping by **pressing Shift+W** but that doesn't give us what we want in this case.



Instead, **press F3 and type Warp** to find the Warp tool there.

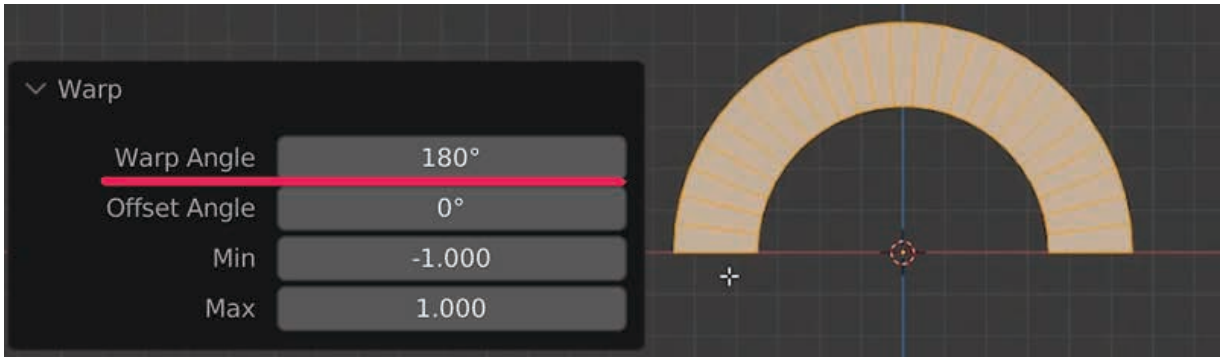


It defaults to 360° but you can change it if needed.

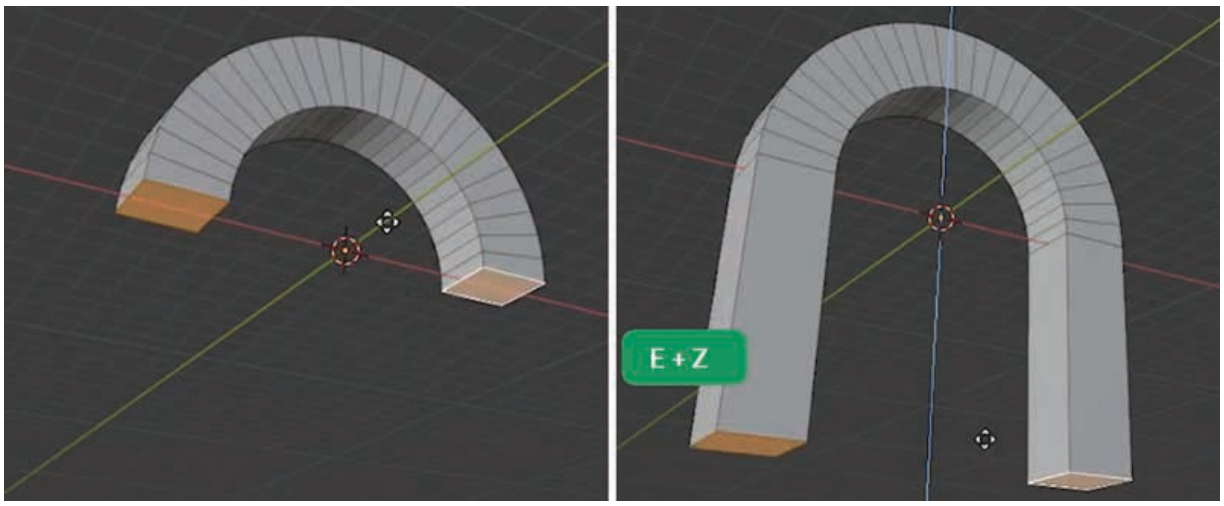


Click here to return to the index.

In this case change it to 180°.

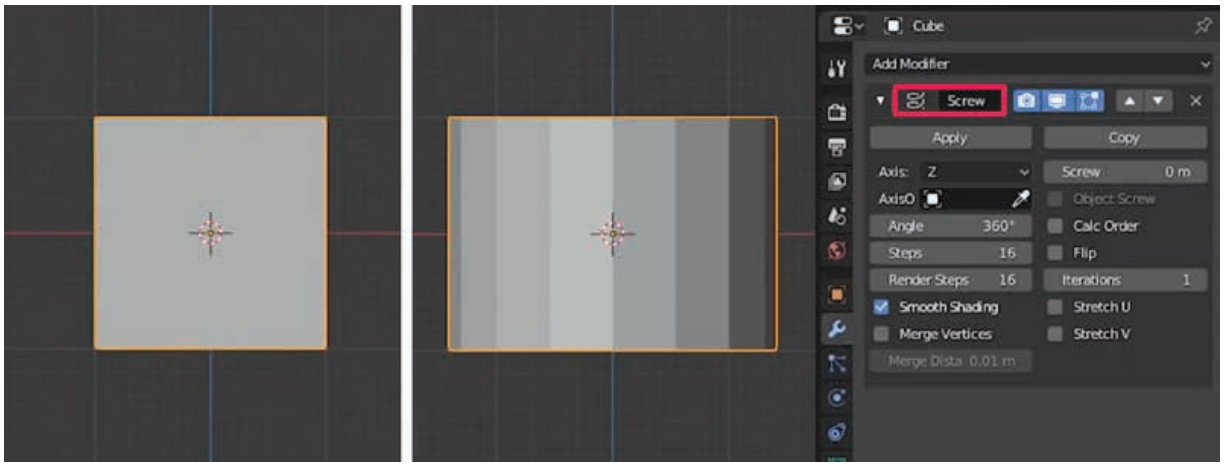


Now select the bottom faces and extrude them by **pressing E and Z** to move them along the Z-axis.

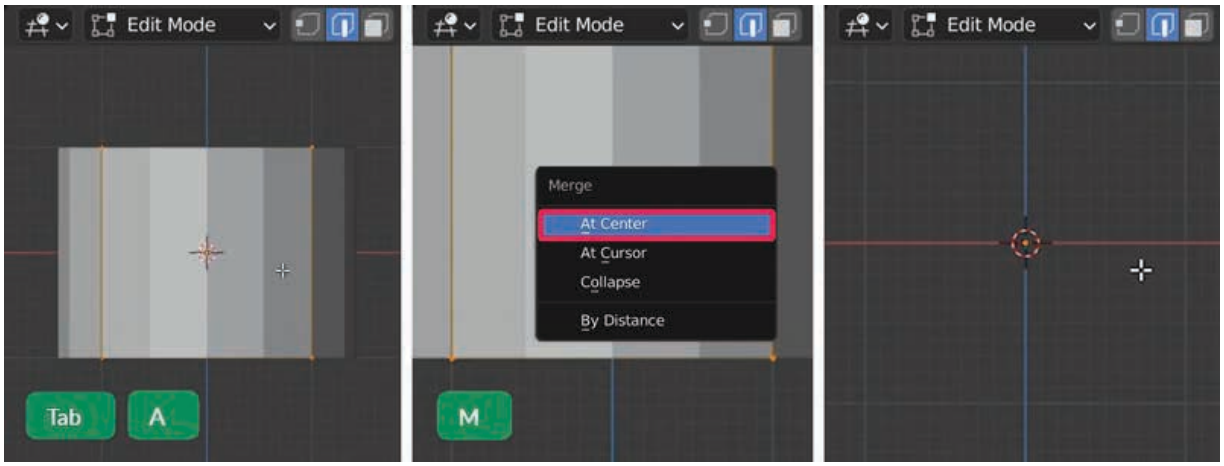


Model shapes with Screw modifier

Select the Default Cube. Add a **Screw modifier**.

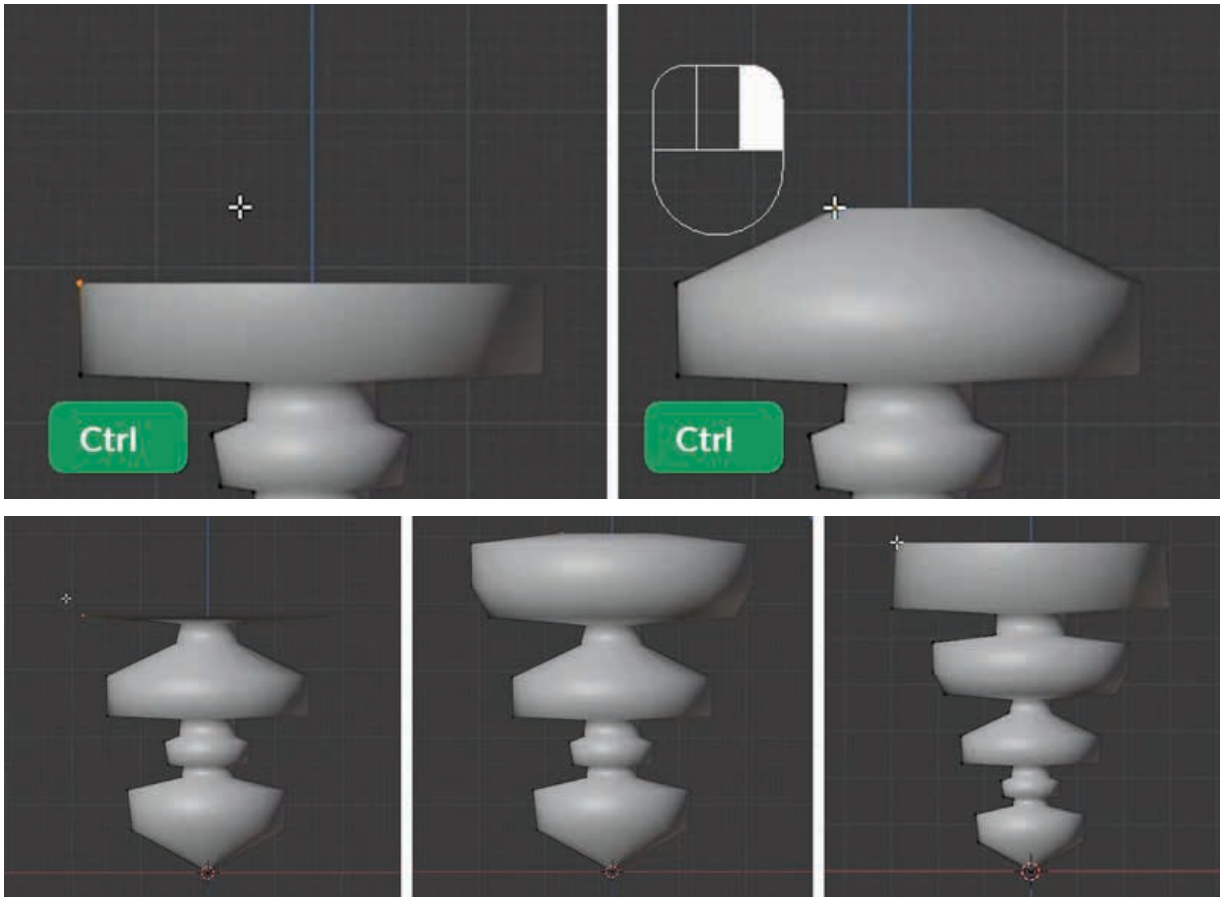


Press TAB to go into Edit Mode. Select all by **pressing A** and **press M** and merge **At Center**.



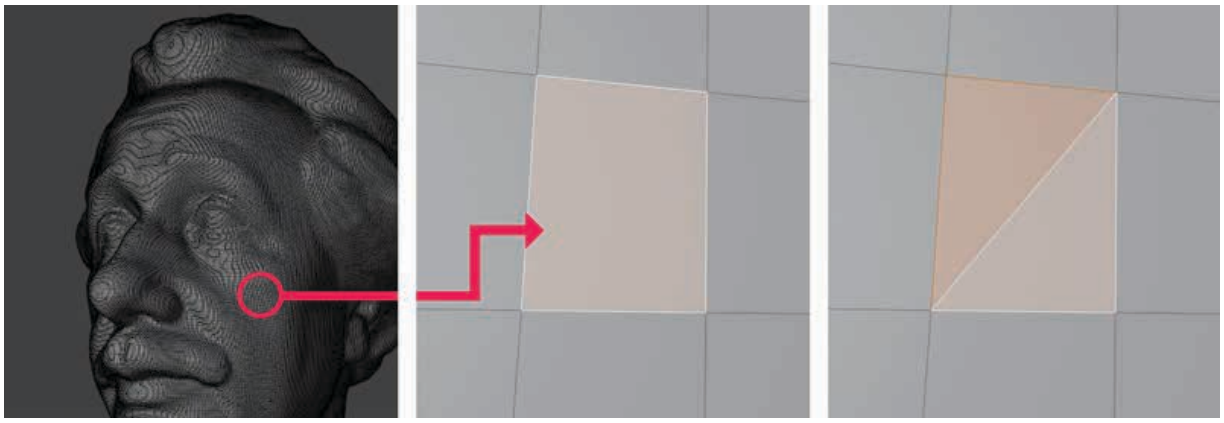
Now you can **press E** to extrude the vertex and model with the Screw modifier.

Instead of **pressing E** you can also **hold CTRL and right click** to extrude.

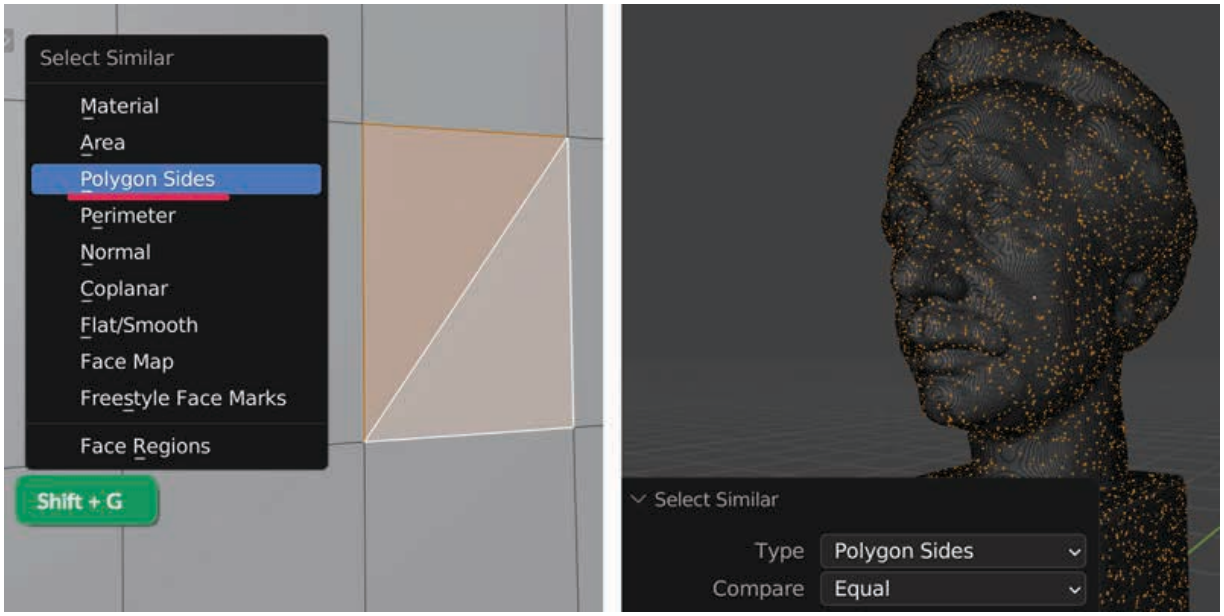


Find the triangles in a dense mesh

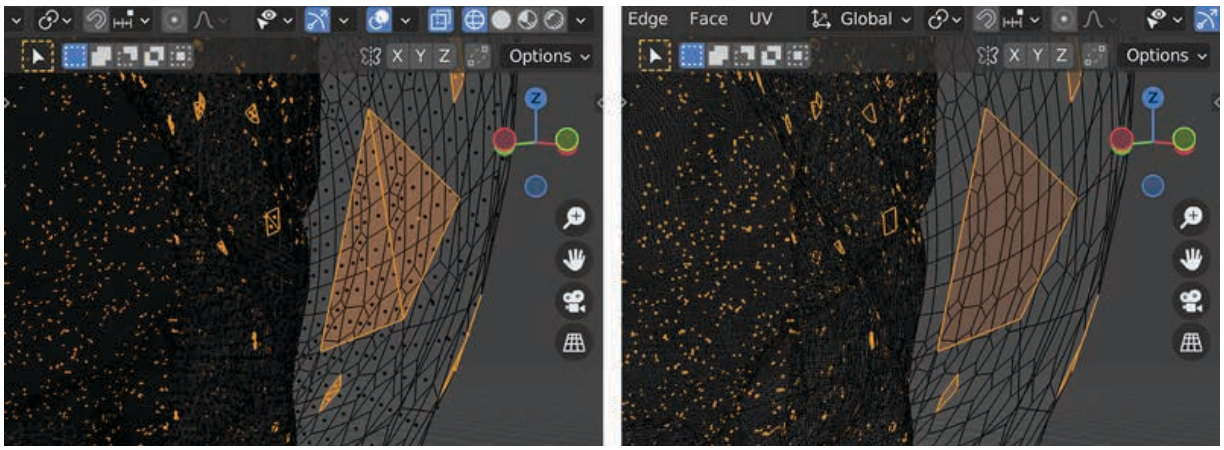
If you need to find a few triangles in a mesh with a lot of faces, this is how to do it. Select one quad face and **press Ctrl+T** to turn it into two triangles.



Now select one of the triangles and **press Shift+G** and choose **Polygon sides**. This way, any other triangles in the mesh will become selected.



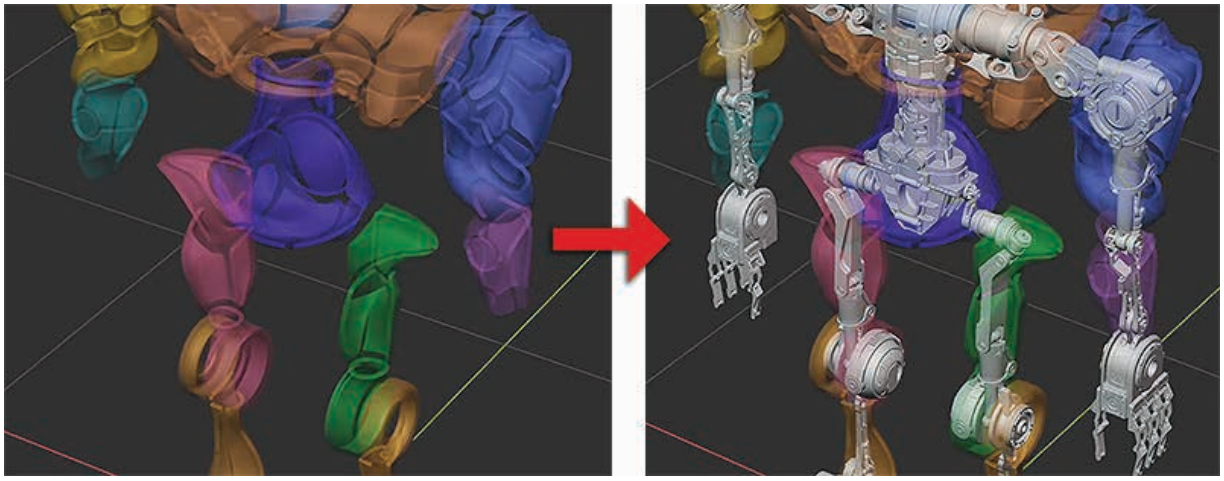
In Wireframe mode you can see them more clearly. You can then **press Alt+J** if you want to turn them into quads.



Click here to return to the index.

Kitbashing details

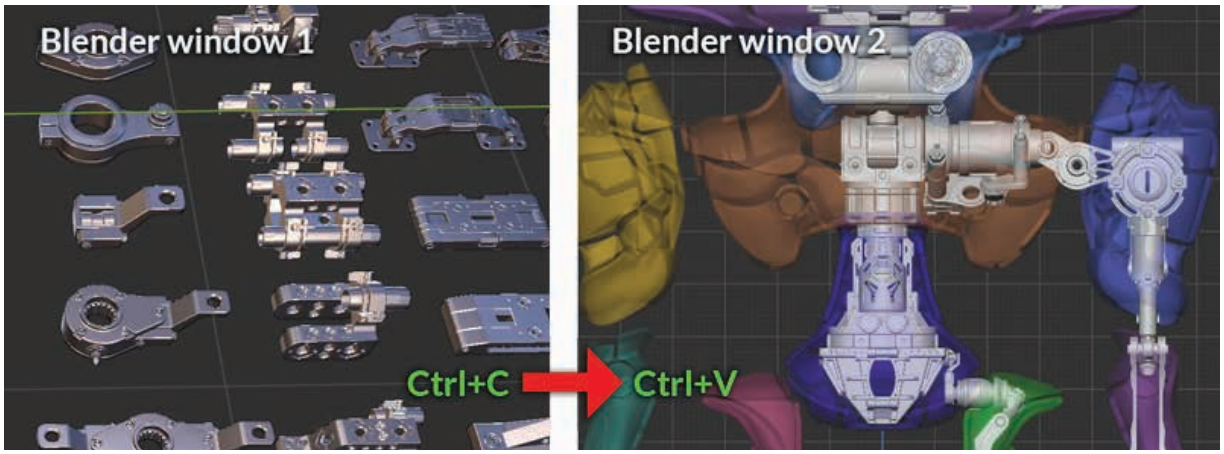
After creating the "outer shell" of a robot, we can add mechanical details using kitbashing.



You can choose to model these details yourself, which is a great way to make sure the chosen design language is carried through. The downside of this is that it's another big time investment, however you can always recycle these details in later projects.

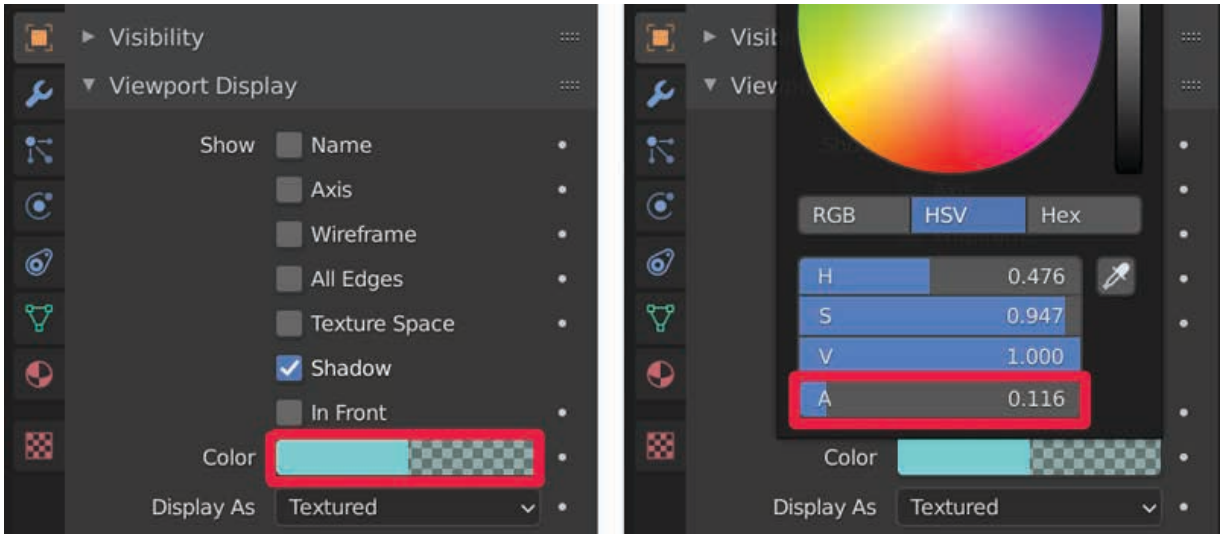
Alternatively you can take advantage of the many kitbashing detail packs sold online. These will save you many hours of work, the downside being that they come with their own design language that may not match your project, and they can be very high-poly and aren't always UV-unwrapped, so you may still need to do some work on them anyway. The kitbash details I used in this case are by Oleg Ushenok.

Instead of appending all the kitbash details that you download, you can open a second Blender instance and copy/paste the details you need to the Blender instance with your project scene (with **Ctrl+C** and **Ctrl+V**). That way when you save your scene, the unused geometry won't be in there.



If you look on Artstation or Gumroad you can find a lot of these kitbashing packs and they're very cheap. It would take you months to model the hundreds of parts that you can buy for very little money.

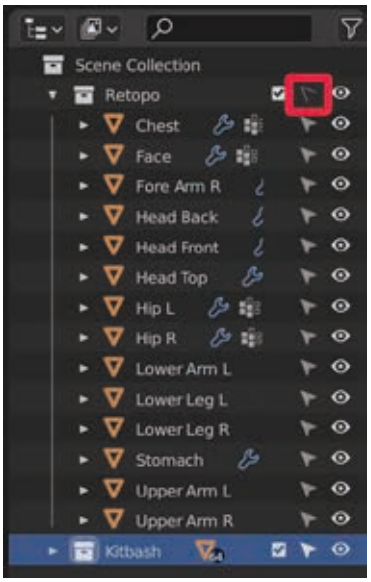
To place the details inside the shell of the robot, you can make the objects partially transparent by lowering the alpha value of the **Viewport Display** Color in **Solid mode**.



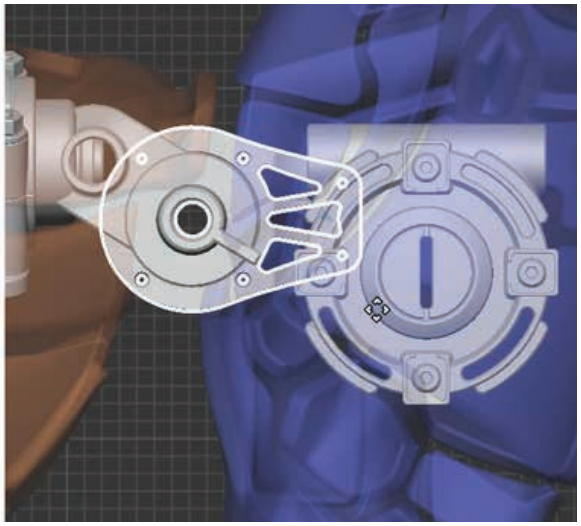
Then turn off selectability for those objects in the Outliner.

The most efficient way to do this is to put all these objects in a Collection and turn off selectability for that Collection, instead of for all the objects one by one.

Then create a new Collection for the kitbash pieces and select it so that anything you add to the scene is added to that new selection.

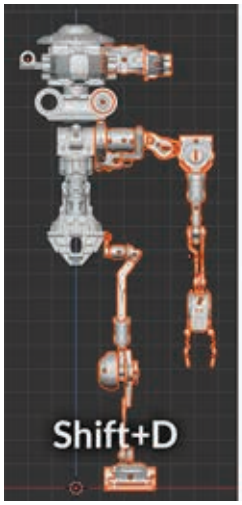


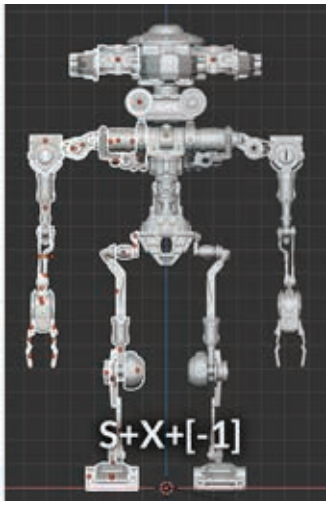
With the retopo objects made transparent in the viewport you can easily see where you place the details inside of the robot. Copy/paste a part from the Kitbash scene to the robot scene and **Press Alt+G** to reset the location of the pasted object. Then move, rotate and scale as necessary to place it where you want.



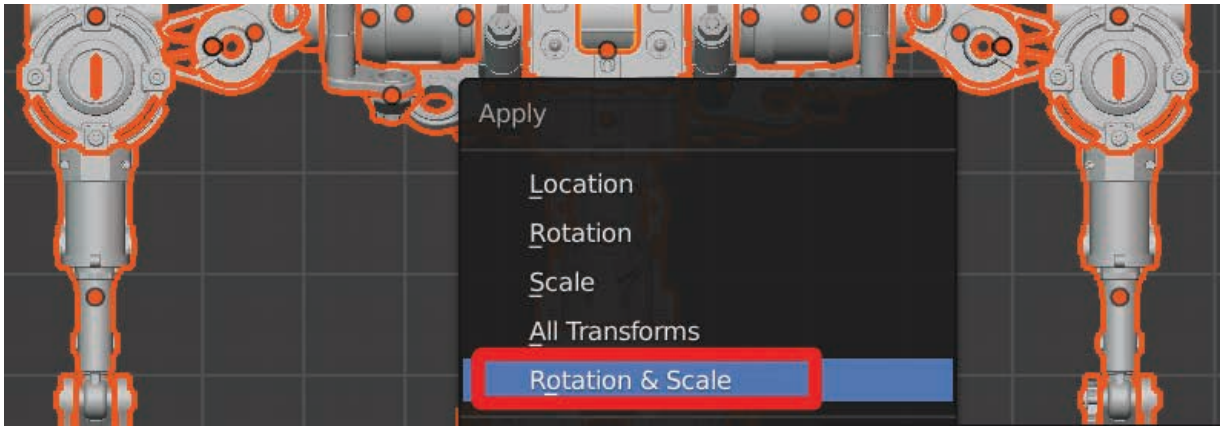
To mirror all the parts, simply select the parts you want to duplicate, **set Transform Pivot Point to 3D cursor** and with the 3D cursor at the world origin **press Shift+D+[-1 on the numpad].**

This will duplicate them to the other side of the X-axis.



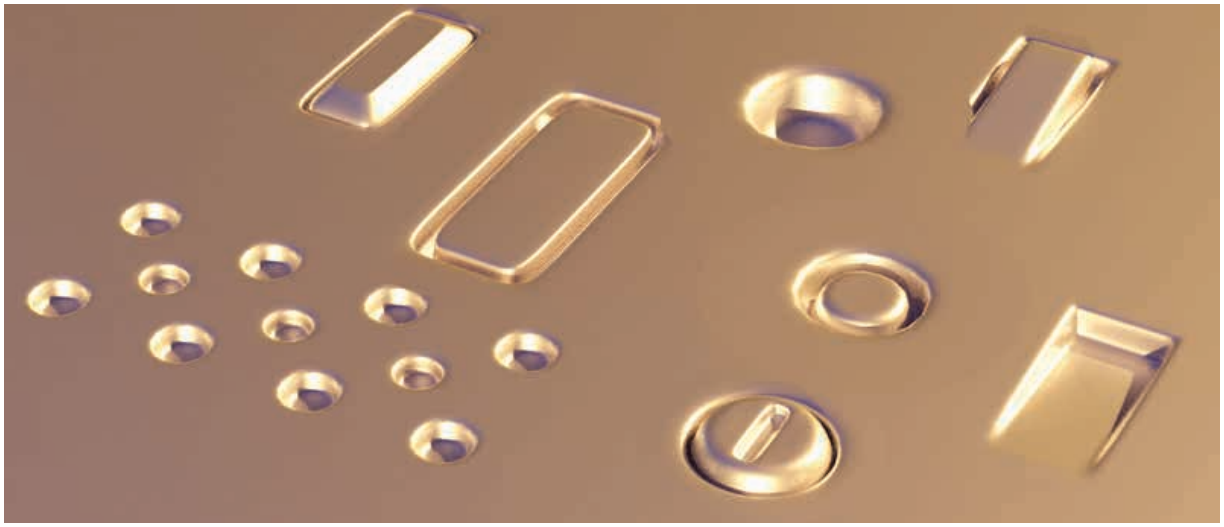


After finishing the kitbashing, make sure all the rotations and scales are applied by selecting all the kitbash parts, **pressing Ctrl+A** and choosing **Rotation & Scale**. After this, some of the normals will be flipped (the ones of the duplicated objects with negative scale) so select all in Edit Mode and **press Shift+N**.



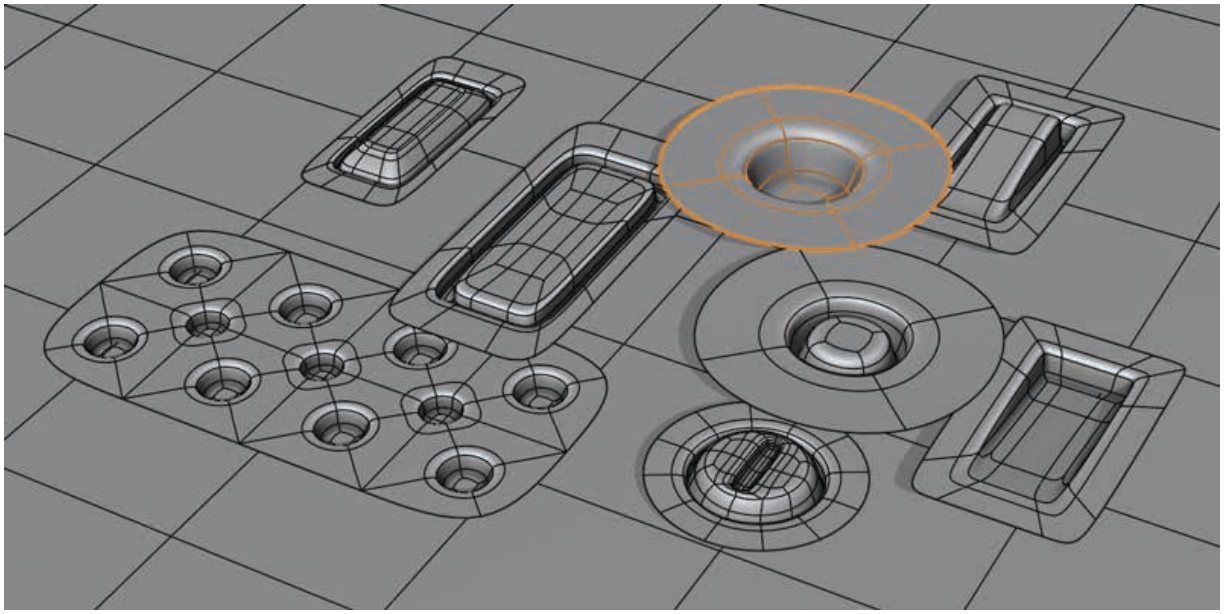
Easy hard surface details using Floaters

"Floaters" are small details that are placed on top of a 3D object, in such a way that it seems like they are connected to that object. They can create the illusion of depth in the object they are placed on, without having to do any boolean operations or anything like that.



Floaters in Rendered preview.

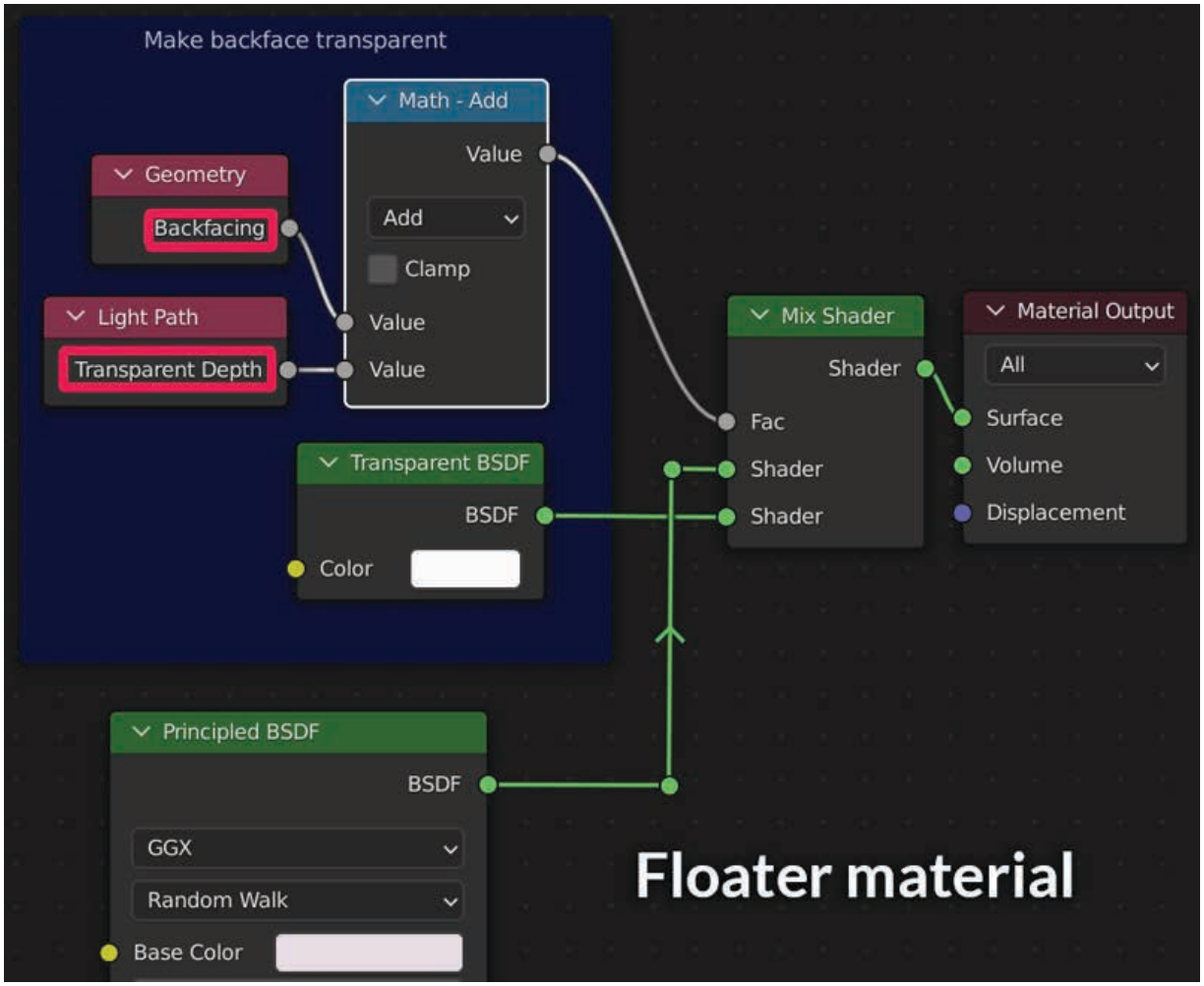
However they are just "floating" on top of that geometry, as you can see when you exit Rendered preview. The effect only works in renders. It also works best on flat surfaces.



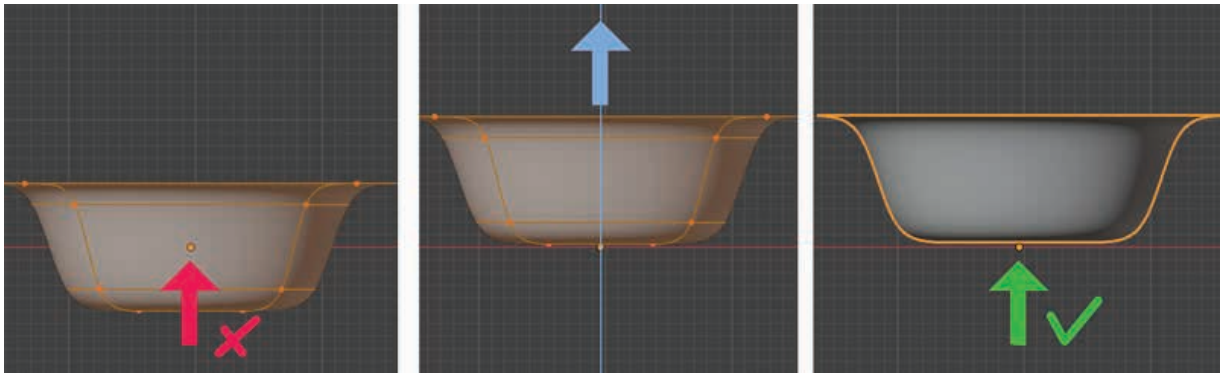
The same floaters, but in Object preview.

The way floaters work is simply by making their backface transparent, which is done by using the below nodes, which I learnt about from this video by Penfinity.

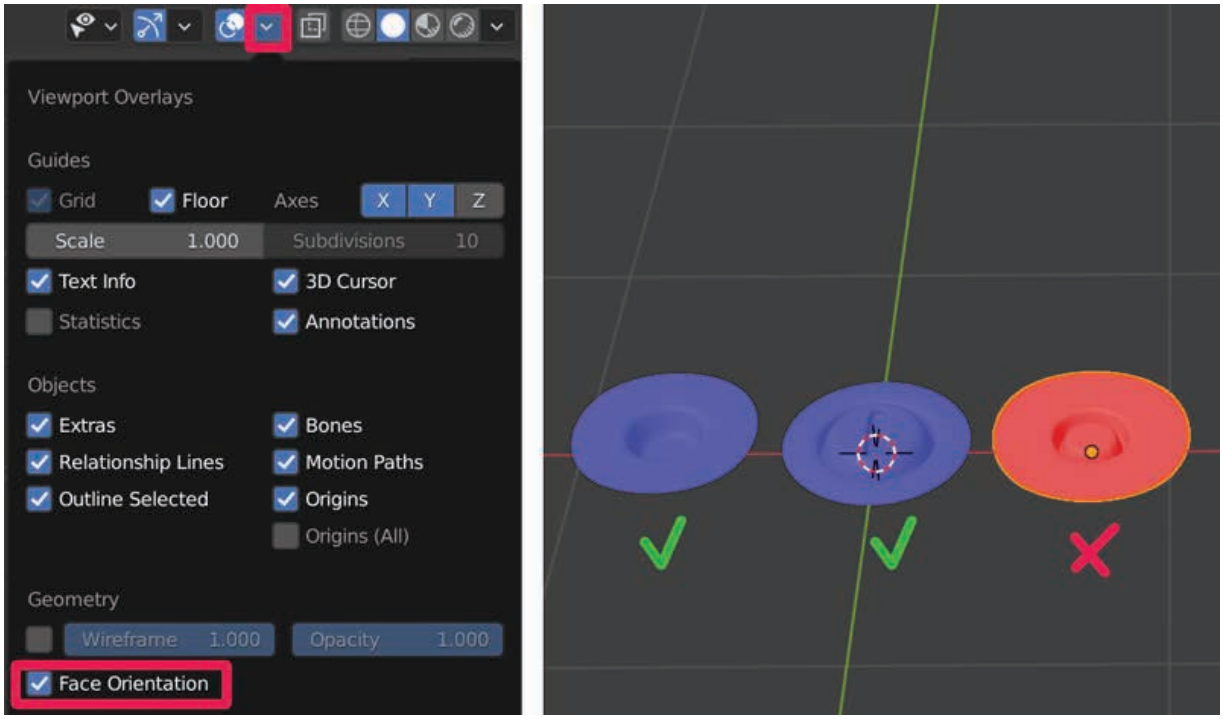
Note that I've simplified some nodes like the Geometry and Light Path nodes by selecting them and pressing Ctrl+H, otherwise these screenshots would get very big. The important thing is that you use the correct outputs: **Backfacing** for the Geometry node, and **Transparent Depth** for the Light Path node. The floater material is for the floater objects only, not the underlying surface.



The position of the **Origin** of the floaters is important: it should be at the bottom of the object. You can select all in Edit Mode and move just the geometry up if you want to move the origin down.



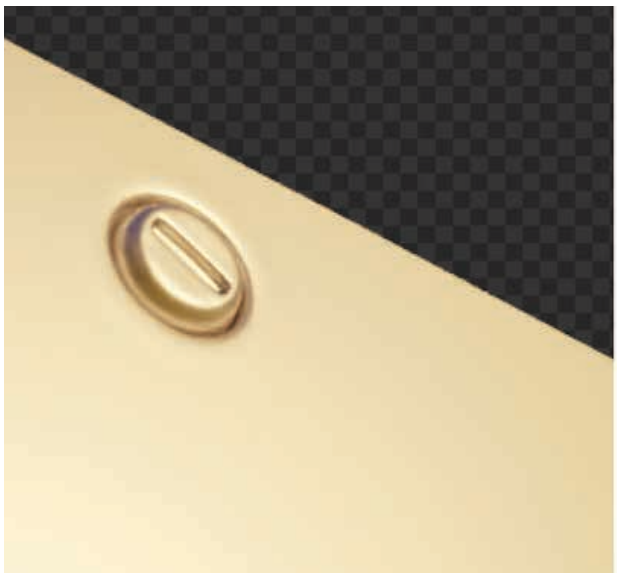
Also, the **Face orientation** (normal orientation) is important, it should be facing the same way as the rest of the geometry, so you should see blue, not red, when you check **Face orientation** in the viewport overlay options. You can reorient the normals by selecting the faces in Edit Mode and **pressing Shift+N**.

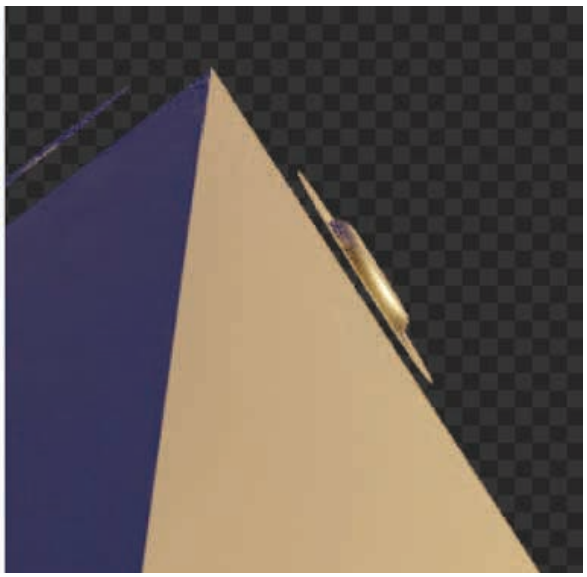


Snapping is set to **Face**, with **Align Rotation to Target** enabled.
Snapping can be turned off, so that it's only enabled when you hold down Ctrl while moving objects. As long as those options are chosen, it will allow you to drag the floaters on the surface with the correct rotation. Also, the smoother the underlying geometry is, the more smoothly the floaters will rotate aligned to that surface. So you may want to add a Subdivision modifier to that surface if it has some curve to it.

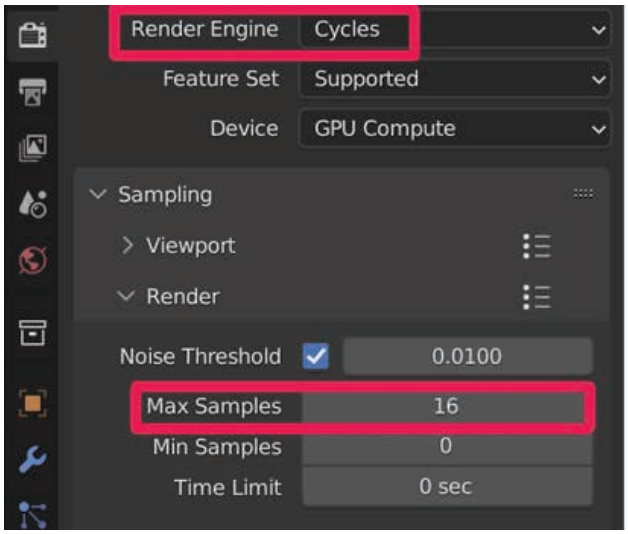
Baking Floaters to Normal maps

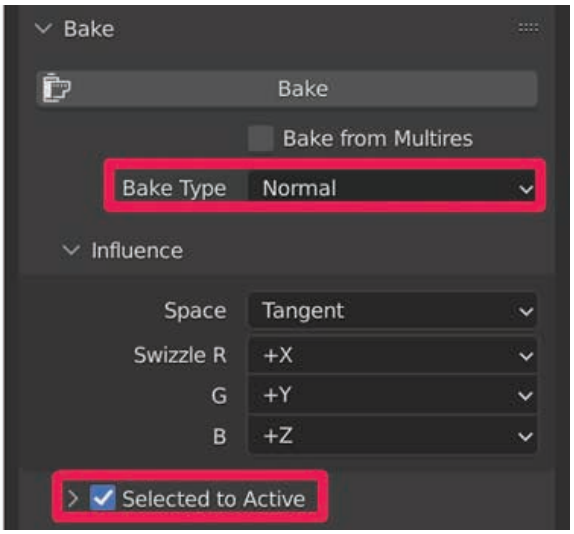
Previously we looked at using floaters to add small details. While it works well, as soon as you look at it from a certain angle, you see that they are in fact above the surface. To solve this, we can bake the floaters.



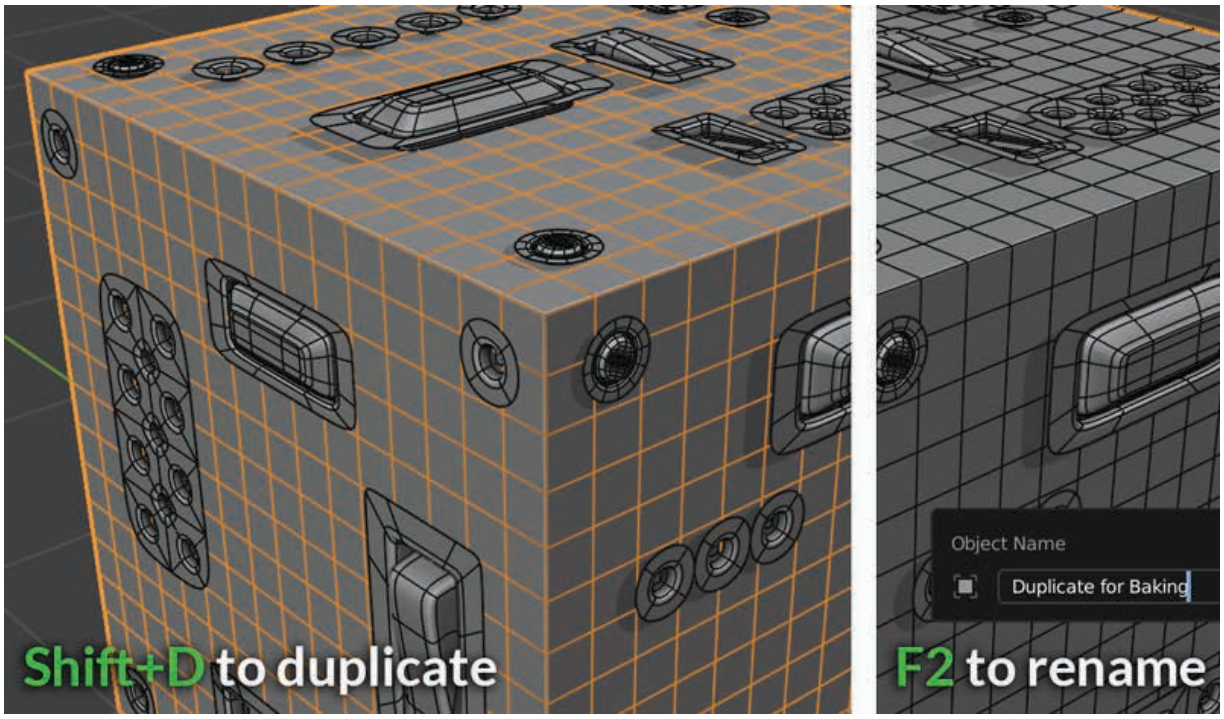


To bake the floaters to a Normal map, you need **baking options**, and for that we need **Cycles**. Set Render samples to 16, otherwise you'll spend a long time waiting. Set Bake Type to **Normal** and check **Selected to Active**.

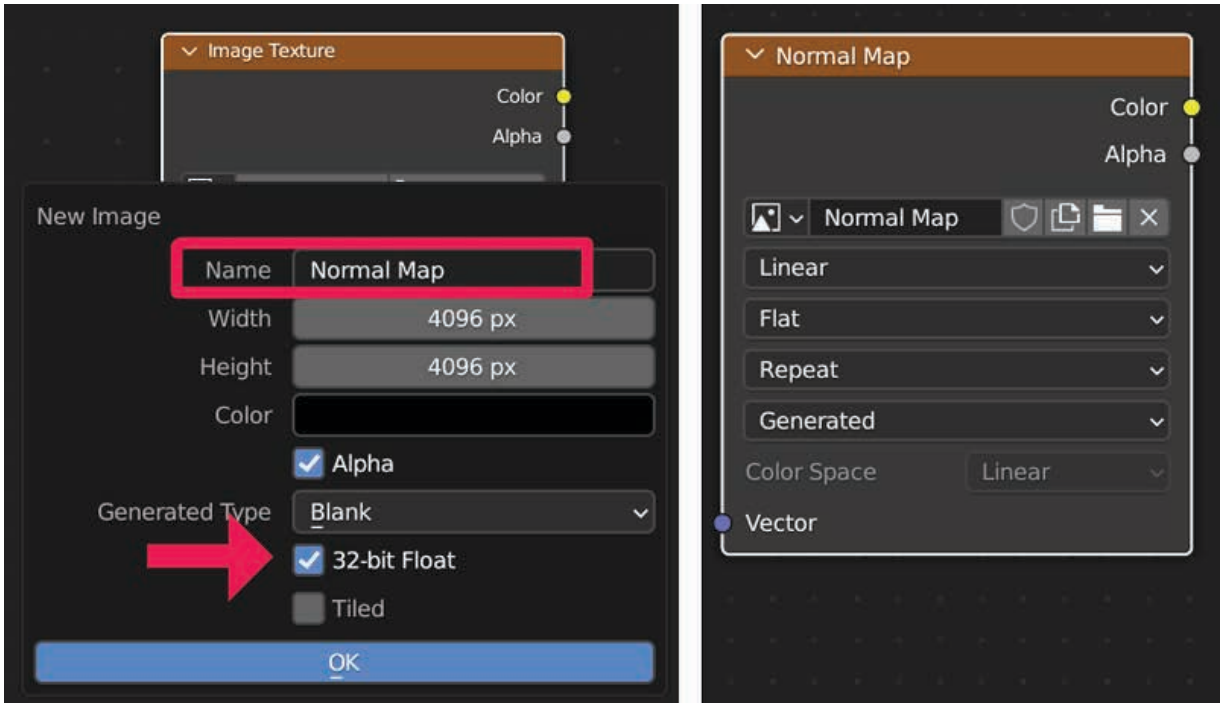




Duplicate the surface object by **pressing Shift+D**. Then rename it something clear like "Duplicate for Baking".

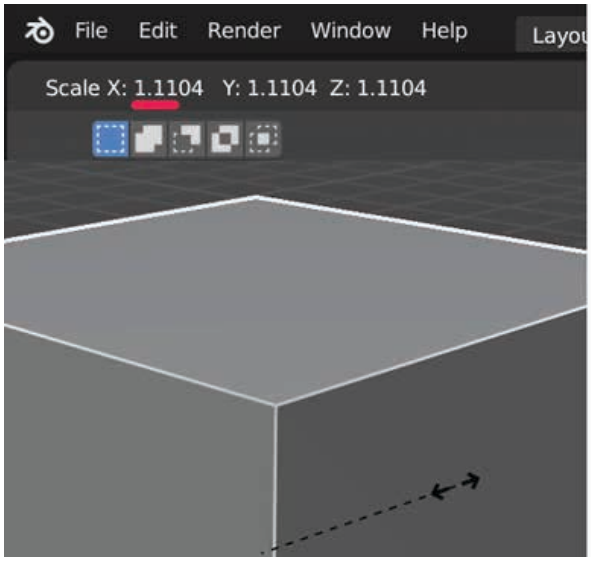


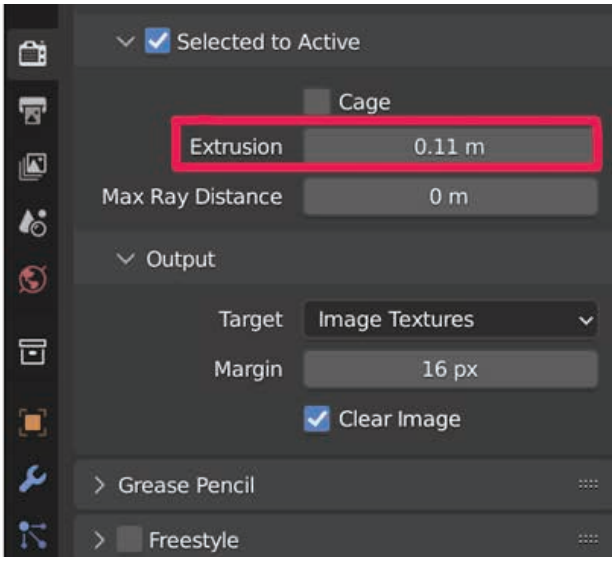
Give the **Duplicate for Baking** a material and add an **Image Texture node**, but don't connect it. Create a new texture in this node, and name it something clear like "**Normal map**". I like to use **32-bit Float** for maximum quality - you can always convert it to 8-bit later if you want.



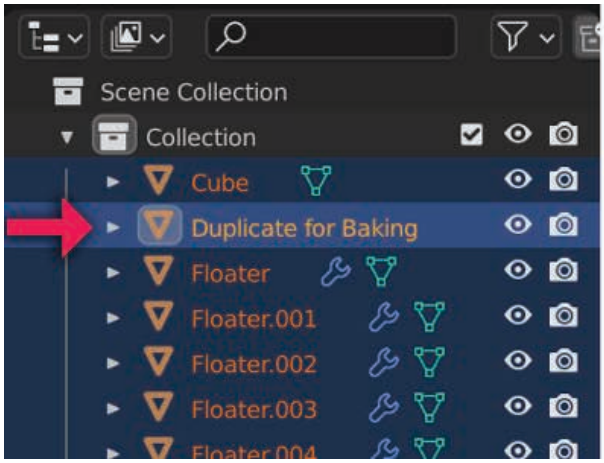
If you expand the "Selected to Active" options, you'll see the Cage settings. But instead of using a Cage, we'll use the "**Extrusion**" option in this case.

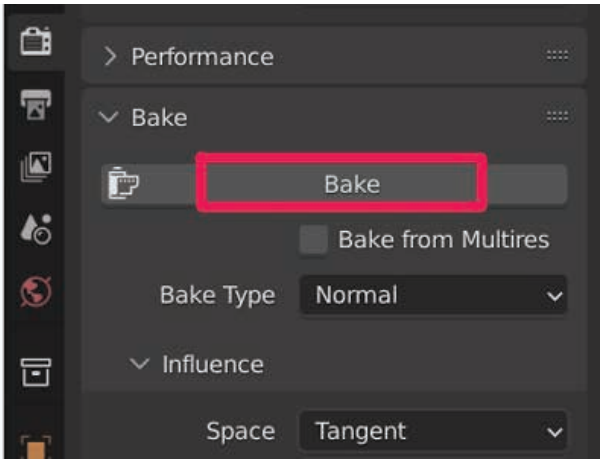
To see what value you need, select the cube (either the original or the duplicate) and scale it up, while keeping an eye on the **Scale** values in the top left corner. The cube envelops the floaters at about a scale of 1,**11** so wel need an Extrusion value of 0,**11** in this case. **Right-click to cancel the scaling operation**.



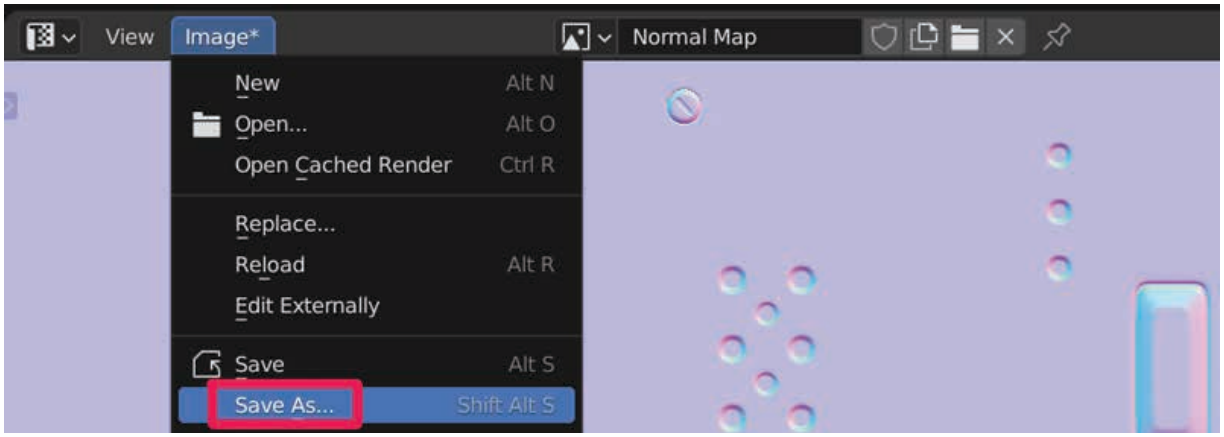


Select all the Floater objects in the Outliner, as well as the Cube. You can select all the floaters quickly by **holding Shift** and selecting the first and last one. Then **while holding Ctrl** select the Duplicate for Baking last, so it is the Active object (it should have a lighter shade of orange/yellow to indicate this). Then **click on Bake**.

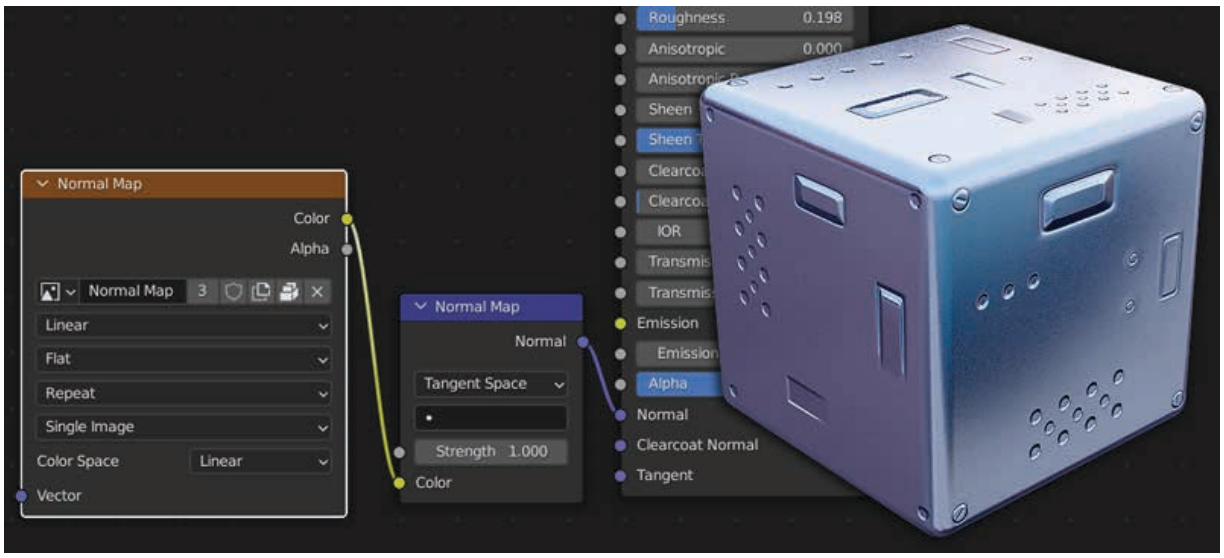




After a while, you'll get a nice Normal map that you can (and should) save!



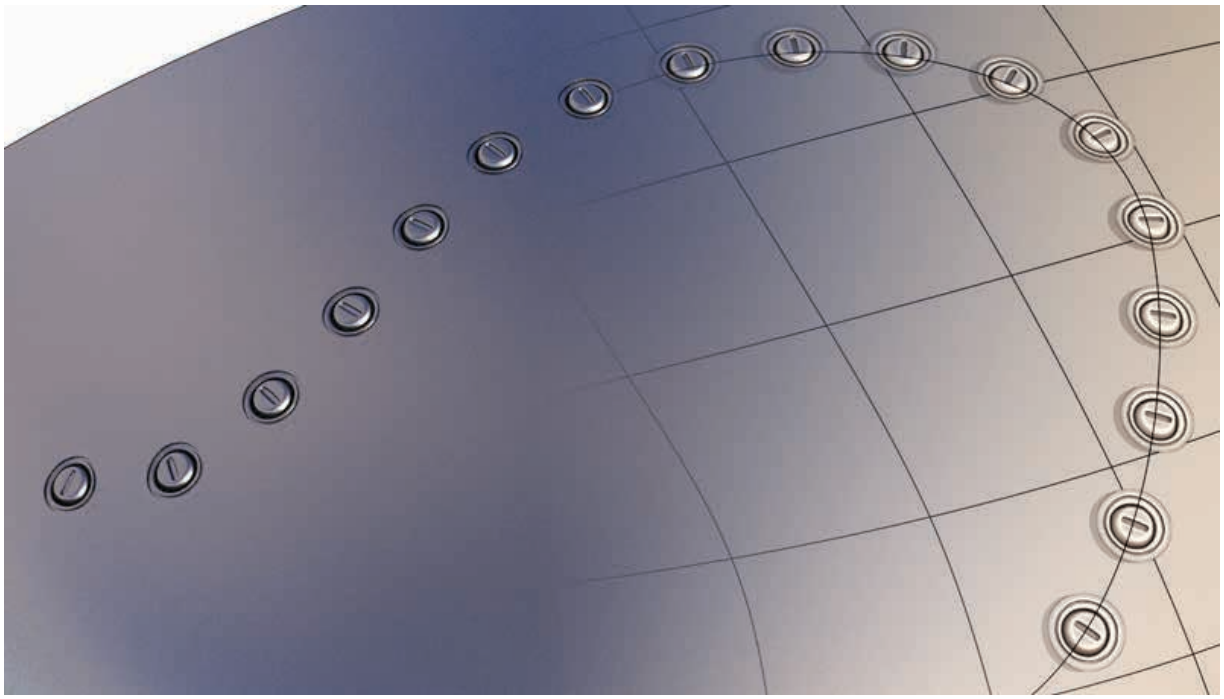
To test the normal map, just connect it through a Normal Map node to the Principled BSDF node.



If you want a **Displacement map** instead of a Normal map, what you can do is first bake a Normal map as described above, and then convert it on the website NormalMap-Online. This is also discussed in this video. This is a bit of a hack and the website does a lot of guesswork, since Normal Maps don't contain height information. But it's better than nothing.

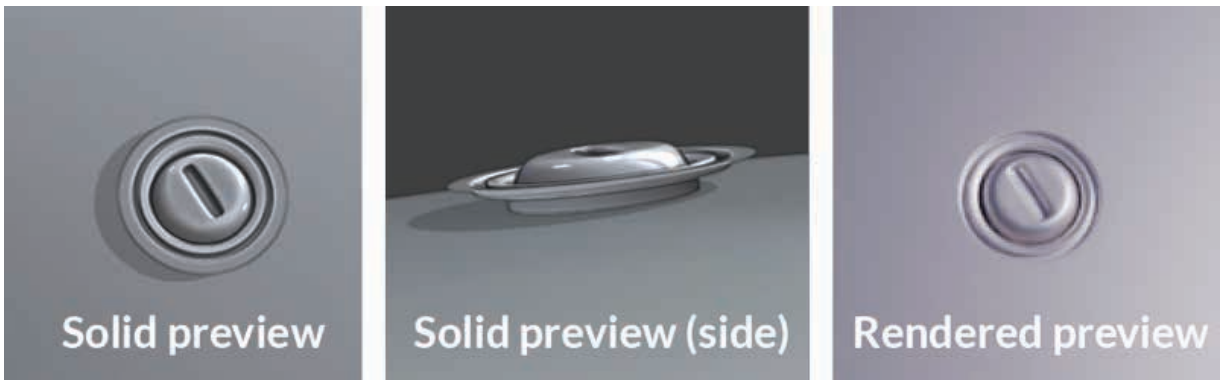
Array of Floaters

Now that we know all about floaters, how about we get fancy with them? And by fancy, I mean making an array along a curve. As always in Blender, there are several ways to do this. Here we'll look at the two most useful methods. First, by using a **Curve modifier.**

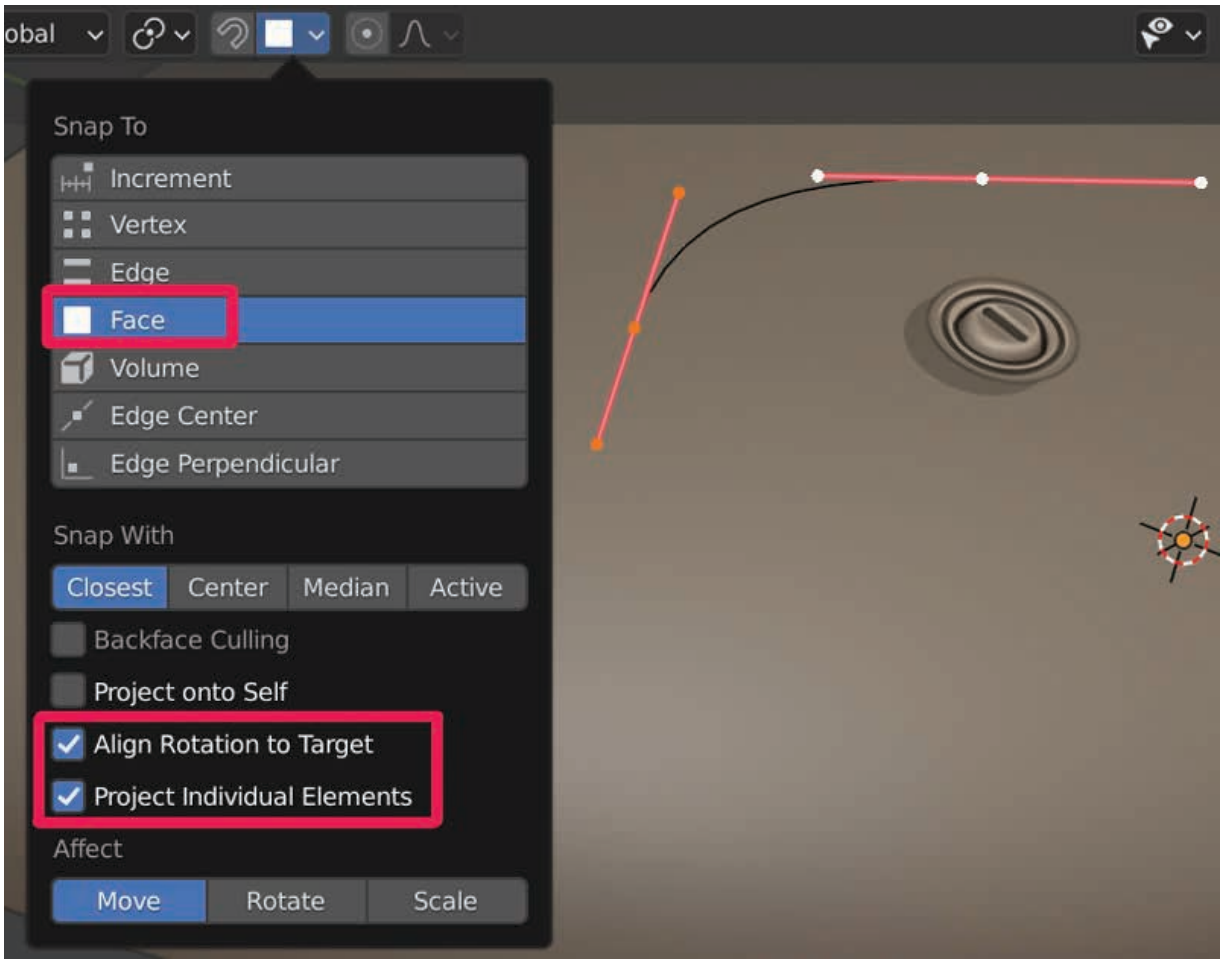


An array of floaters along a Curve

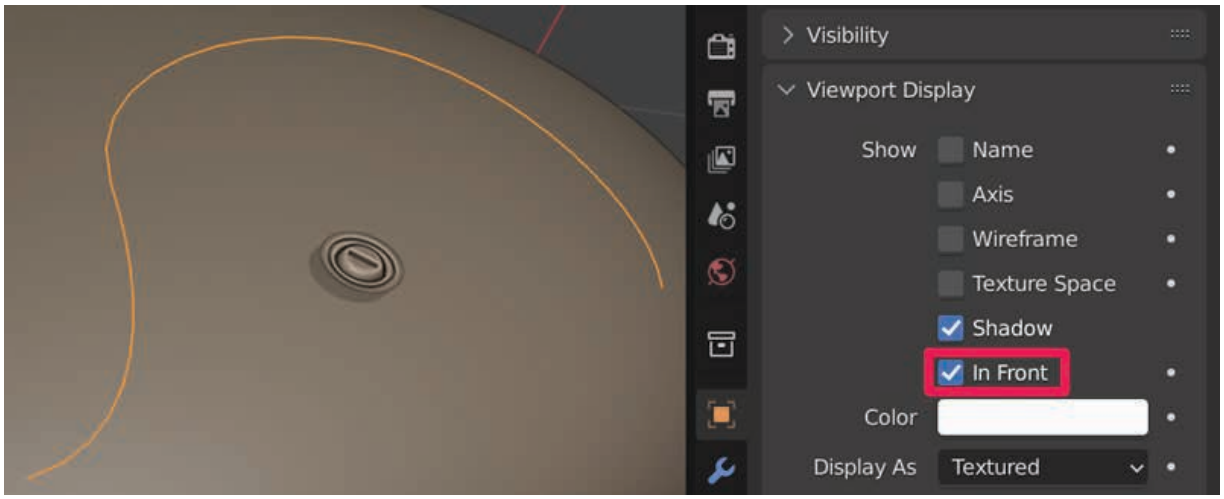
I've got this screw floater on a curved surface. It helps if the surface object has a **Subdivision modifier** so that it's smooth enough.



Press Shift+A. Add a curve by choosing **Curve > Bezier**. With the below settings enabled, you can snap the curve to the surface by **holding Ctrl** when you move its vertices in Edit Mode. Scale them with S and move them with G along the surface.

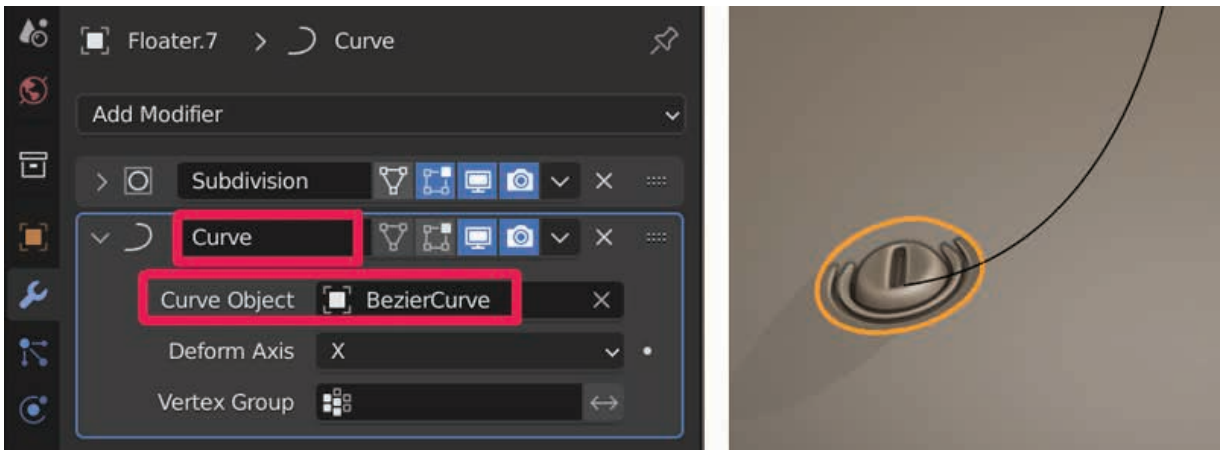


You can also rotate, move and scale the individual vertices and handles. Don't forget to **hold Ctrl** to keep it snapped to the surface. By enabling **In Front** it's easier to see the curve.



In Edit Mode right-click and choose **Subdivide** and repeat that step (or press Shift+R to repeat) to subdivide the curve a few times, so that it has enough vertices to really snap to the surface. With them all selected **press G** to make sure they're snapped to the surface.

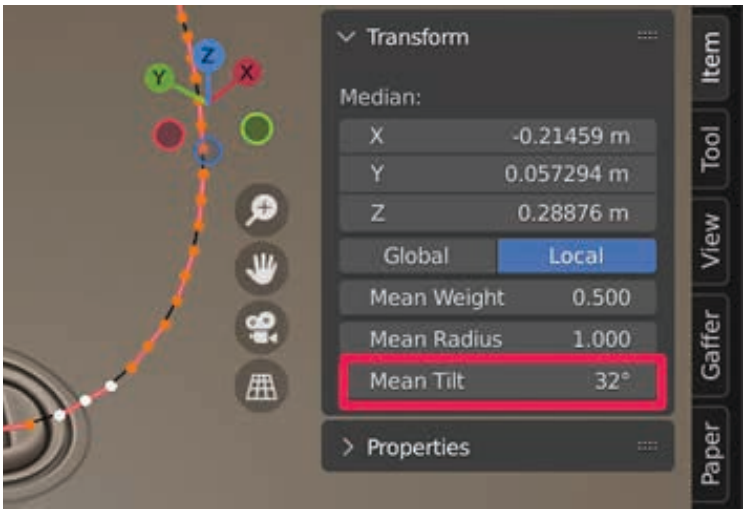
Reset the location and rotation of the floater by selecting it and **pressing Alt+G** (clear location) and **Alt+R** (clear rotation). Add a Curve modifier to it and choose the Bezier Curve as the Curve Object.



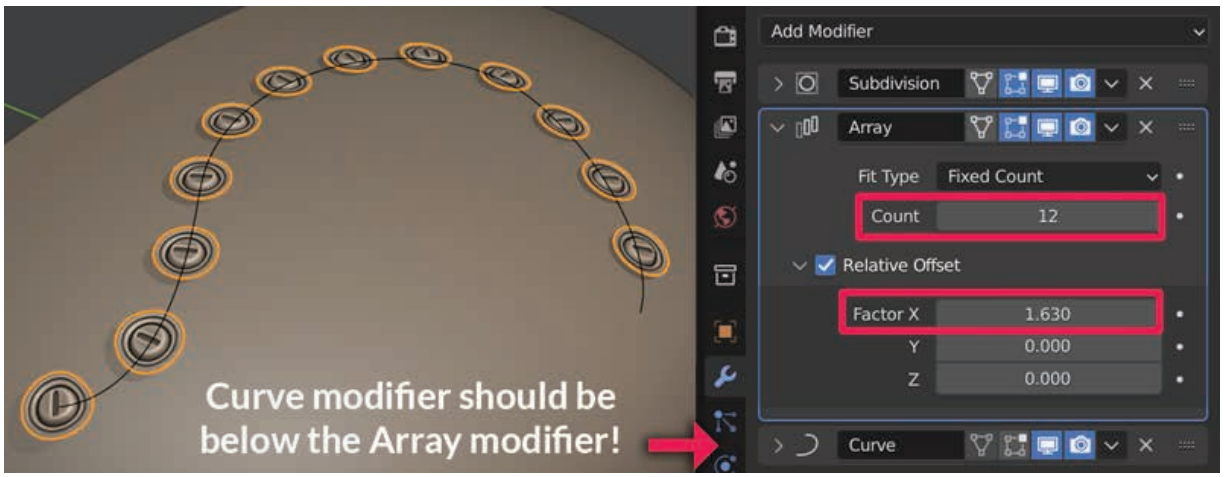
As you can see, it kind of works - but the floater isn't rotated correctly.

To solve this, select the Curve and in Edit Mode, **press Ctrl+T** and move the mouse until it looks good.

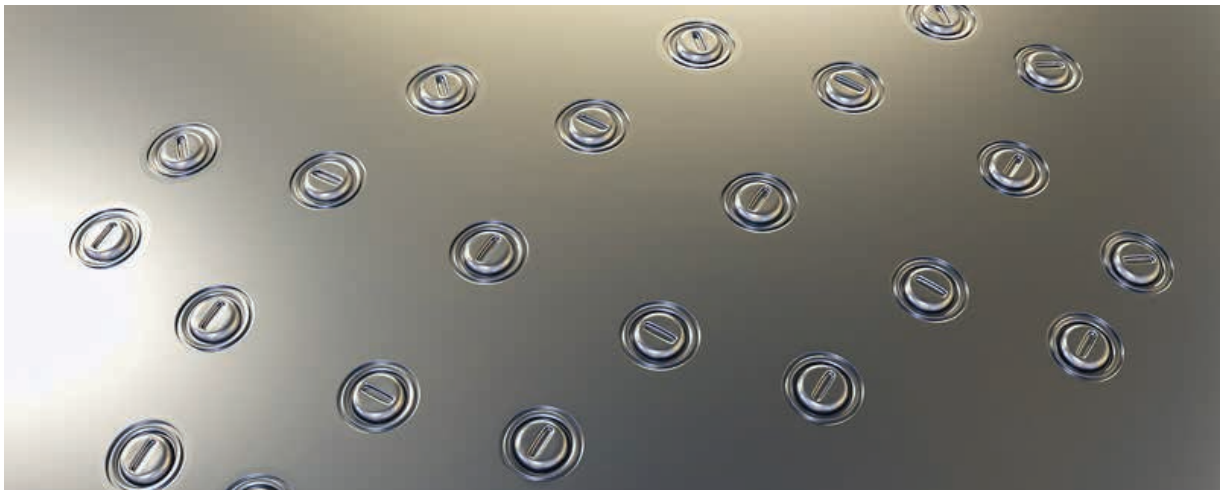
Or adjust the **Mean Tilt** value in the Option panel.



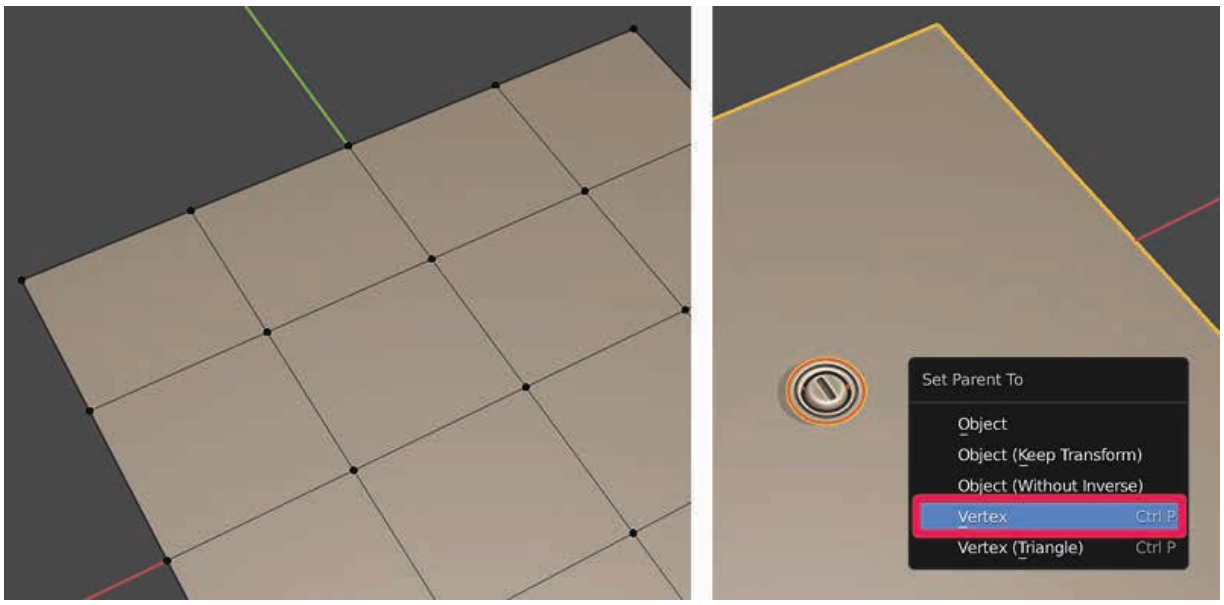
Then, add an **Array modifier**. It should be above the Curve modifier for it to work properly. Increase the Count to get more floaters. Then increase the **Factor X** to more than 1 to add some distance between them.



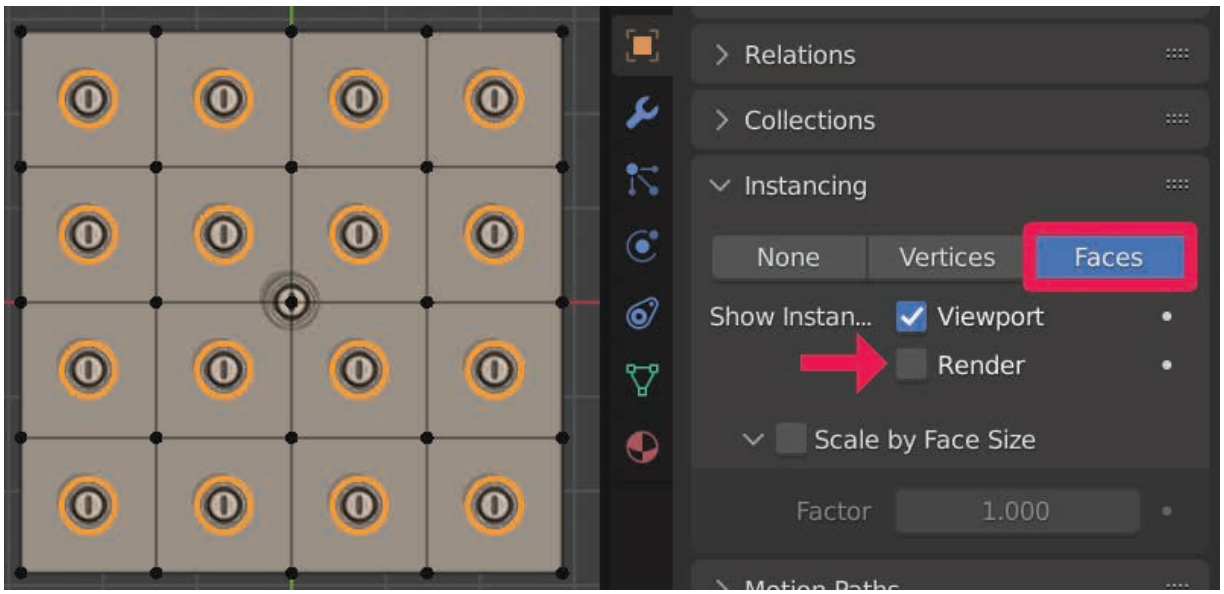
That's all, now you've got an array of floaters! However, if you look closely you'll notice that the Curve modifier deforms the shape of the floaters slightly. This is how the Curve modifier is meant to behave, but it may not be desirable. Fortunately there is another method that doesn't have this problem. And that method, is using **Dupliverts**. This method also allows for more complex arrays, like this:



Dupliverts require an intermediate object with vertices. We can use a Plane, which is subdivided a couple of times. In Object Mode, select the floater and then the Plane while holding Shift. **Press Ctrl+P** and choose **Vertex.**

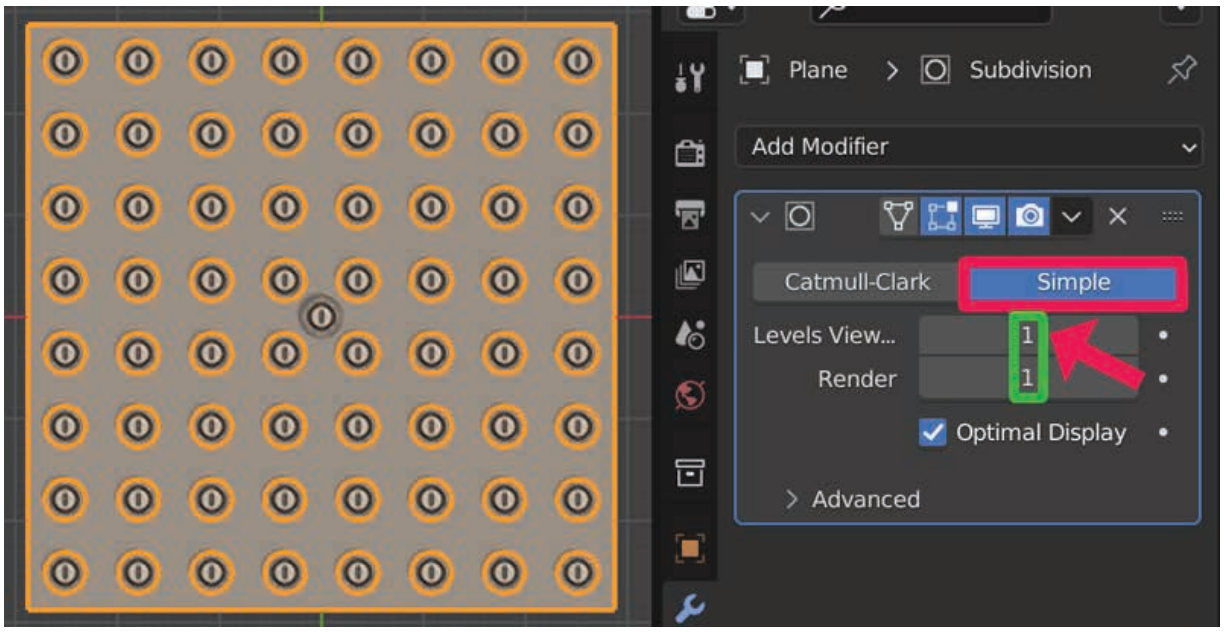


In the **Object Properties** tab of the Plane, expand the **Instancing** options and select **Faces**. This will show one floater per Face. You can also choose Vertices if you prefer. But the Faces option has the advantage that the orientation of the individual floaters is aligned to the normal of the Face. Also, **uncheck Render visibility** for the Plane so it doesn't show up in the render.

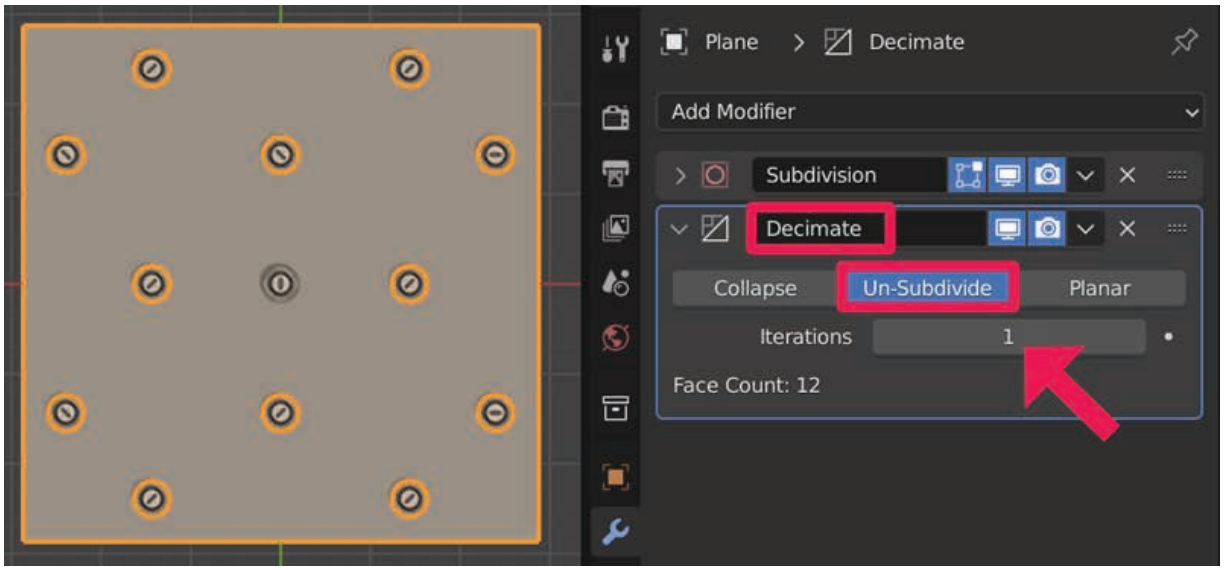


Now we have some interesting options. For example, by adding a **Subdivision modifier** to the Plane, we can increase the amount of floaters (since we're increasing the amount of Faces with the modifier).

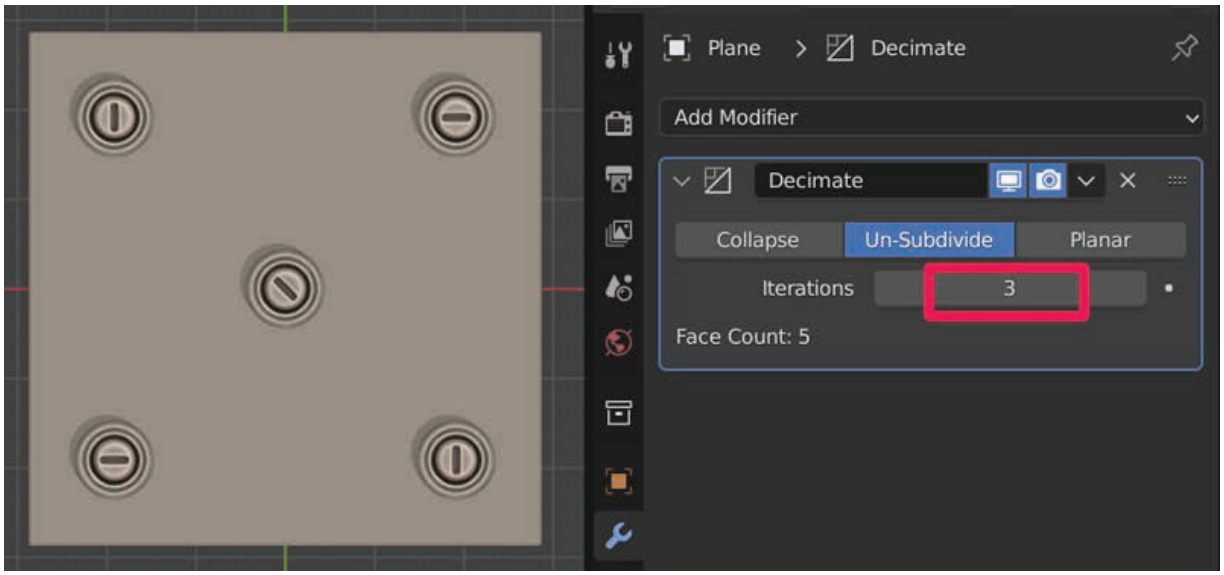
Set it to **Simple** so the Plane doesn't become rounder. Also, make sure the Viewport and Render **Levels** are identical. Otherwise, you'll get a different amount of instances in the render.



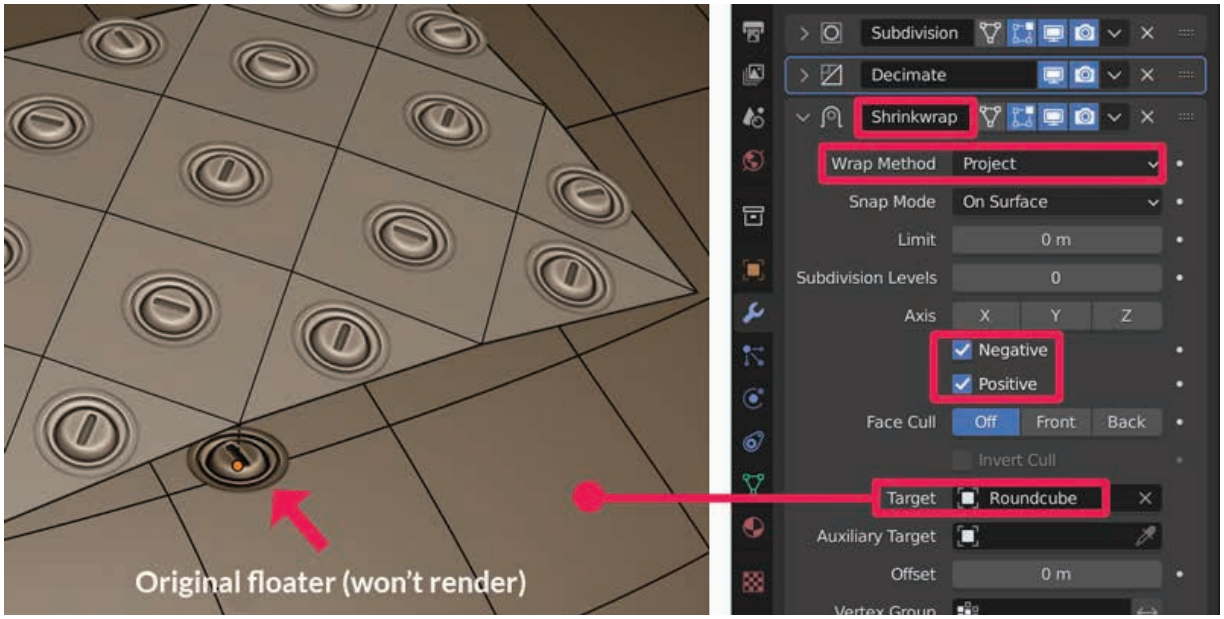
Adding a **Decimate modifier** set to **Un-Subdivide** can also lead to interesting effects.



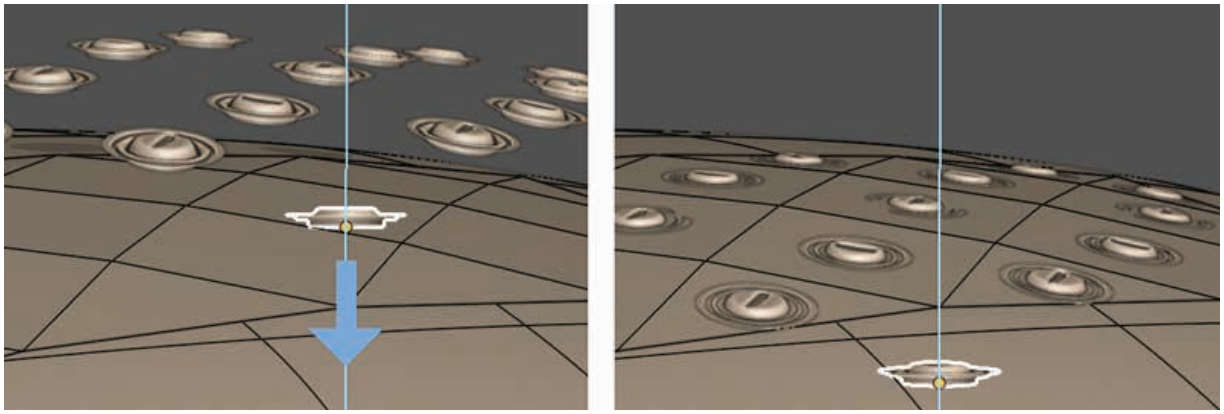
You can experiment with different uneven values for the Decimate modifier and different amounts of Subdivision (or none at all) for different distributions of floaters. Note, that the original floater won't be rendered. You can scale it up and down to scale the instances up and down.



To wrap the Plane to a surface which has some curvature to it, you can use a **Shrinkwrap modifier** with the below settings: Wrap Method **Project, Negative** and **Positive** checked, and the surface object as the **Target**.



In my experience, you may have to move the original floater up or down to get the instances to align correctly to the surface when using a Shrinkwrap modifier.



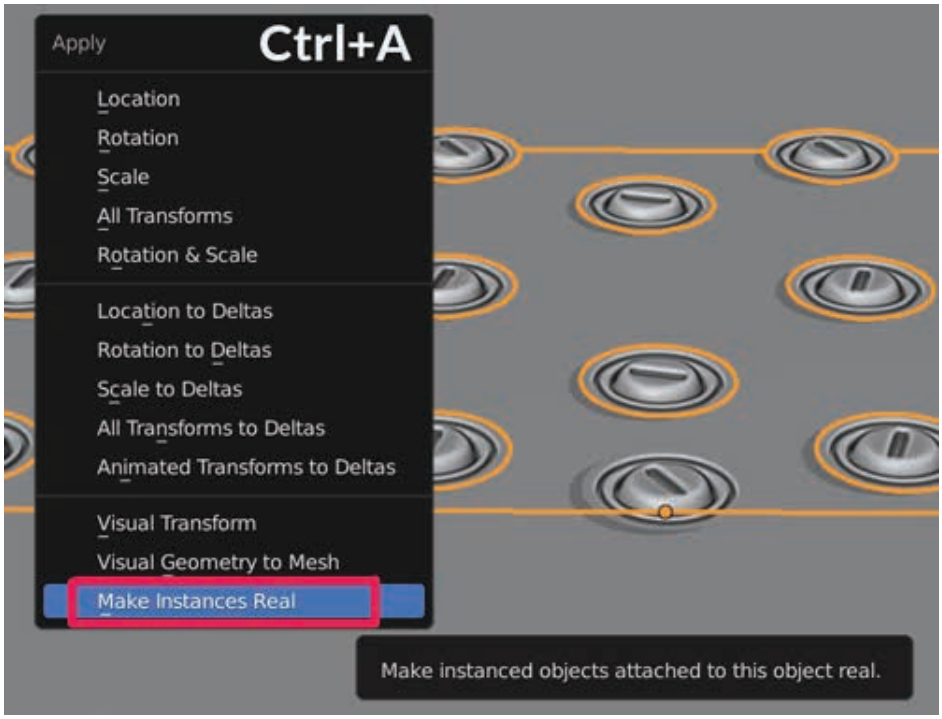
In the end, when we render we get an array of floaters.

Cycles Render of the floaters array

Of course, we can still bake this to a Normal Map as shown before. Especially with curved surfaces underneath the floaters, that is advisable as the illusion created by the floater shader is more likely to break the less flat the underlying geometry is.

However, to bake the instances you need to make them "real" first - by **selecting the Plane and pressing Ctrl+A** and choosing **Make Instances Real**.

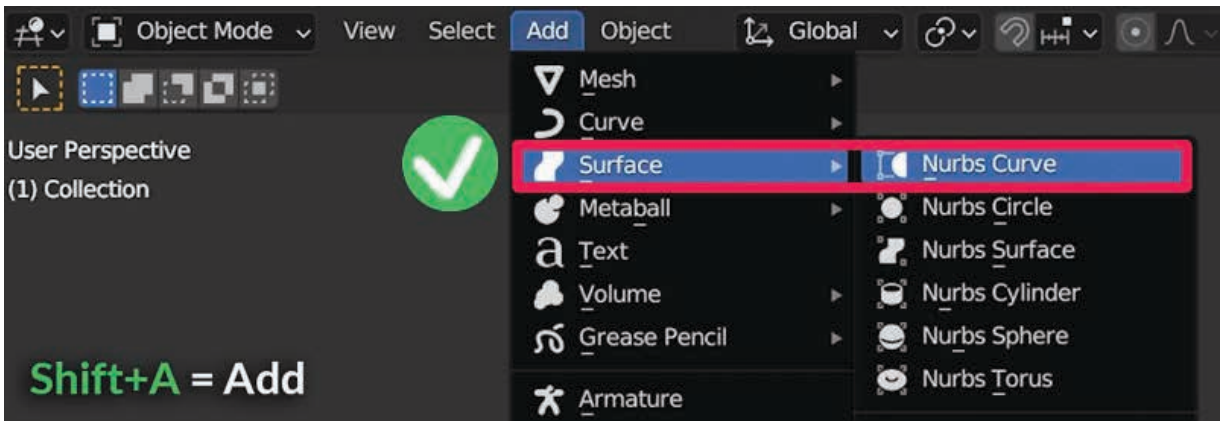
After making them "real", you'll need to remove the original floater and the Plane, as you don't need them anymore and to avoid that they show up in the Render.



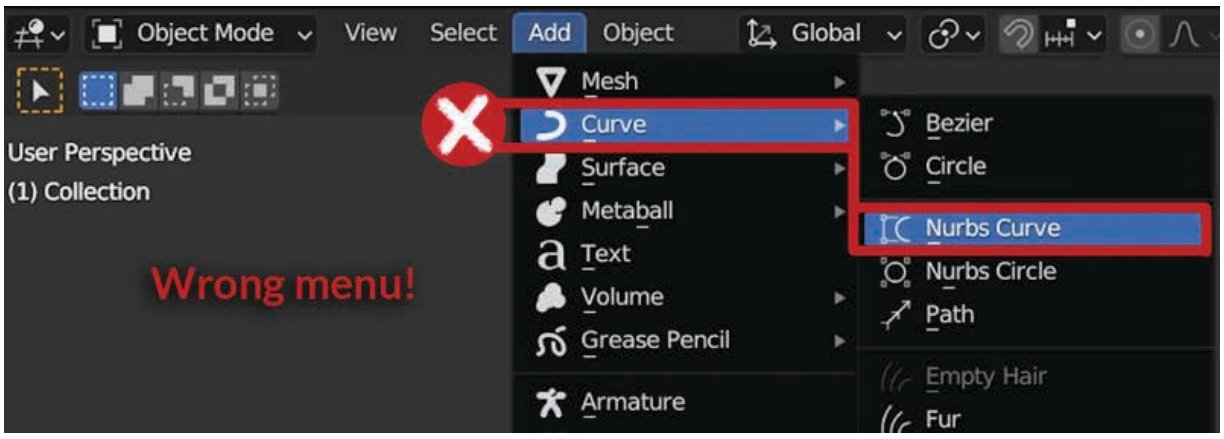
Basics of Nurbs Surfaces

NURBS (Non-Uniform Rational B-Splines) Surfaces can be handy for precision modeling. They are generally used in car design, CAD, and industrial prototyping for their precise handling of curved surfaces. Though the NURBS tools in Blender are quite limited, they're enough to make for example a smooth-looking mouse.

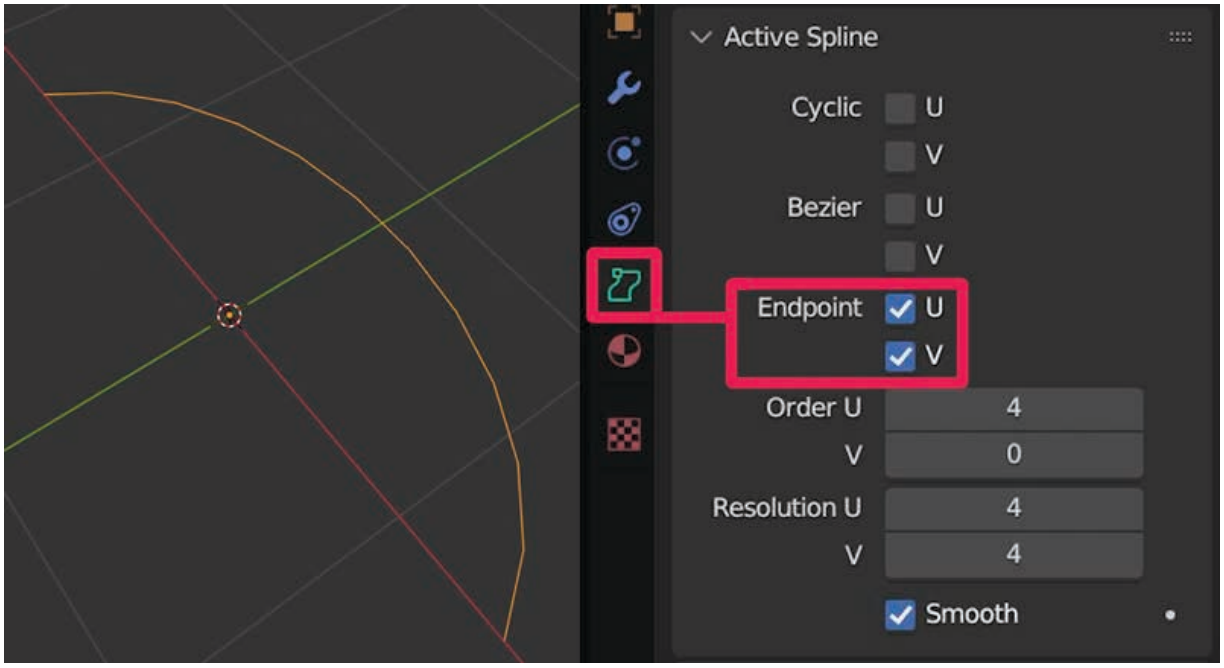
Add a Nurbs Surface by going to Add > Surface > Nurbs Curve.



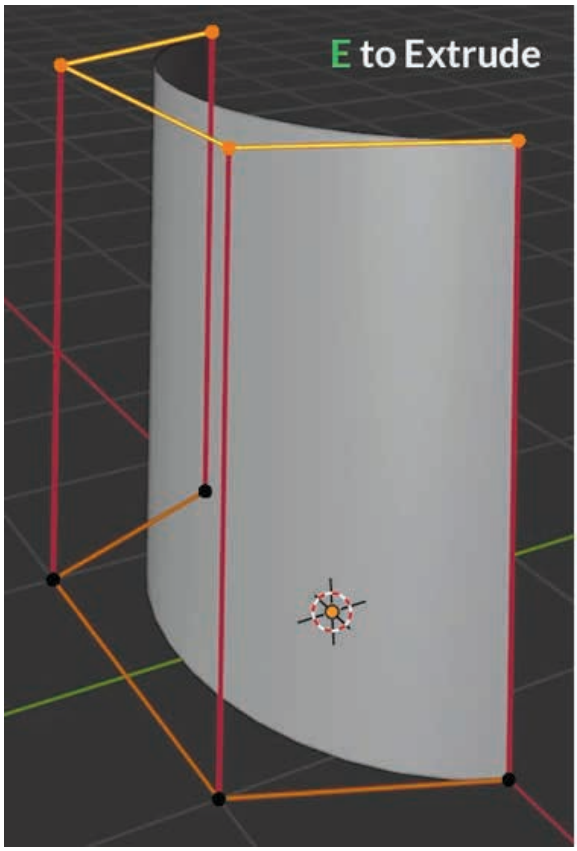
Be careful that you don't go to the wrong menu.

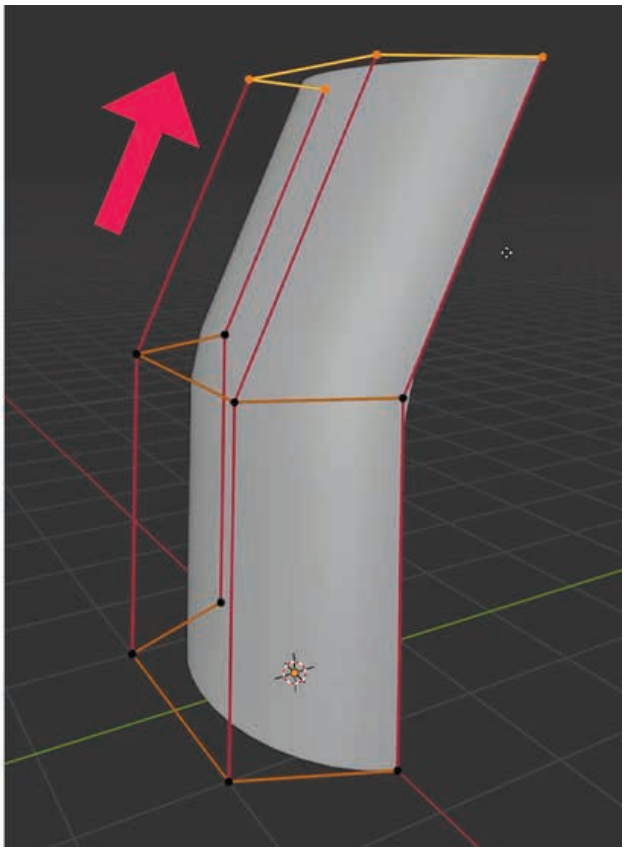


Under Active Spline options, check **Endpoint U** and **V**.

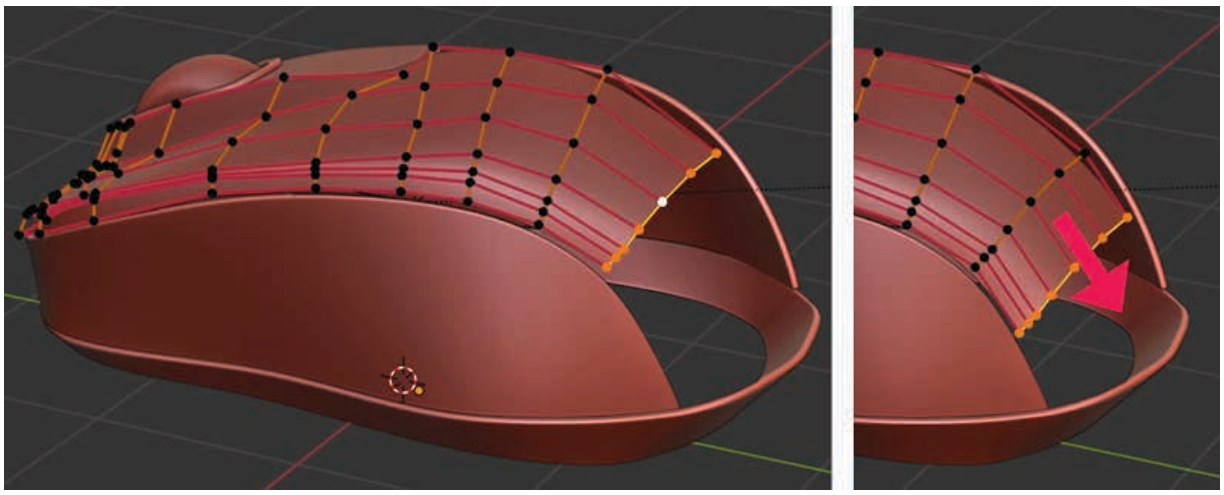


Now you can manipulate individual Vertices. Or you can select vertices and **press E to extrude a surface**. Note, that you need to select vertices along an entire edge. You can't select just a few vertices for extrusion, like you can with Subdivision modeling.

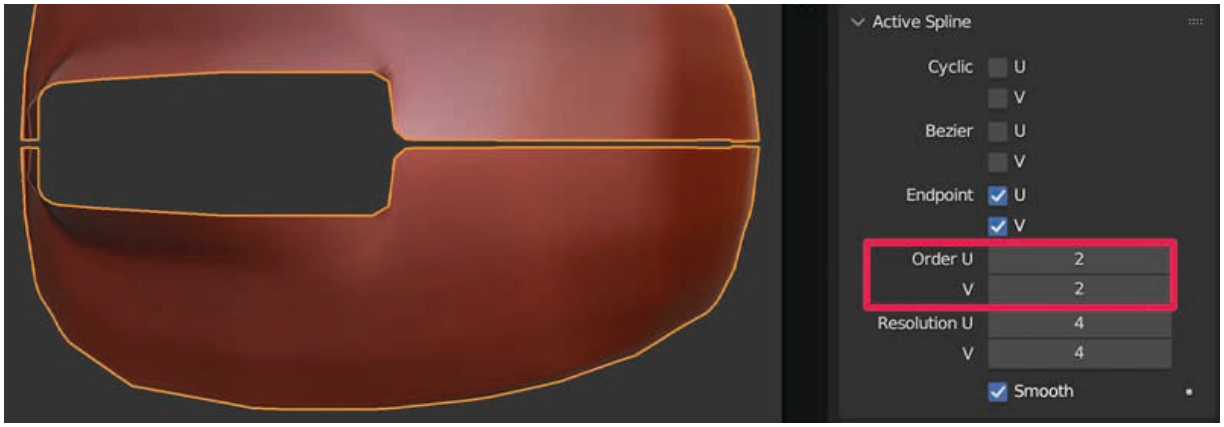




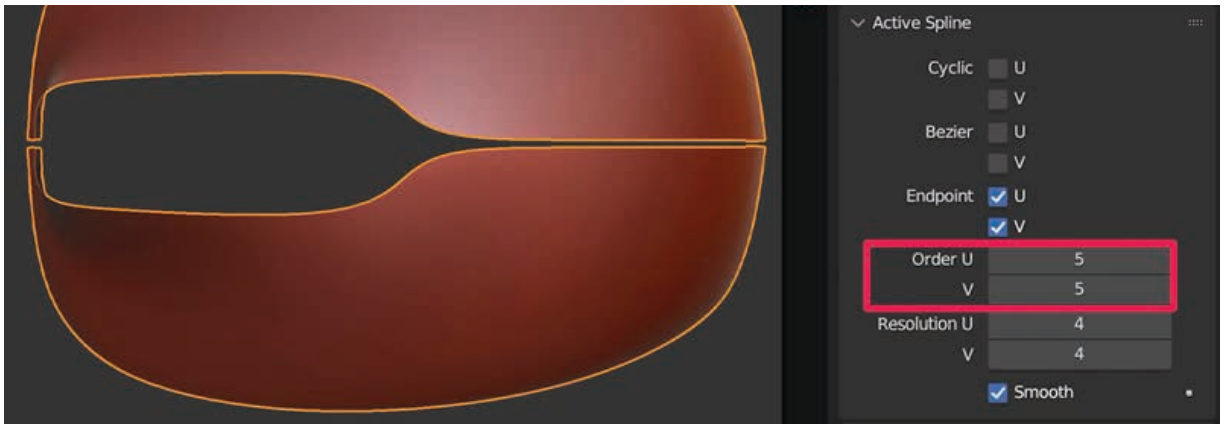
This is also how I modeled this mouse. Of course, you need to tweak the individual vertices to keep a nice smooth flow.



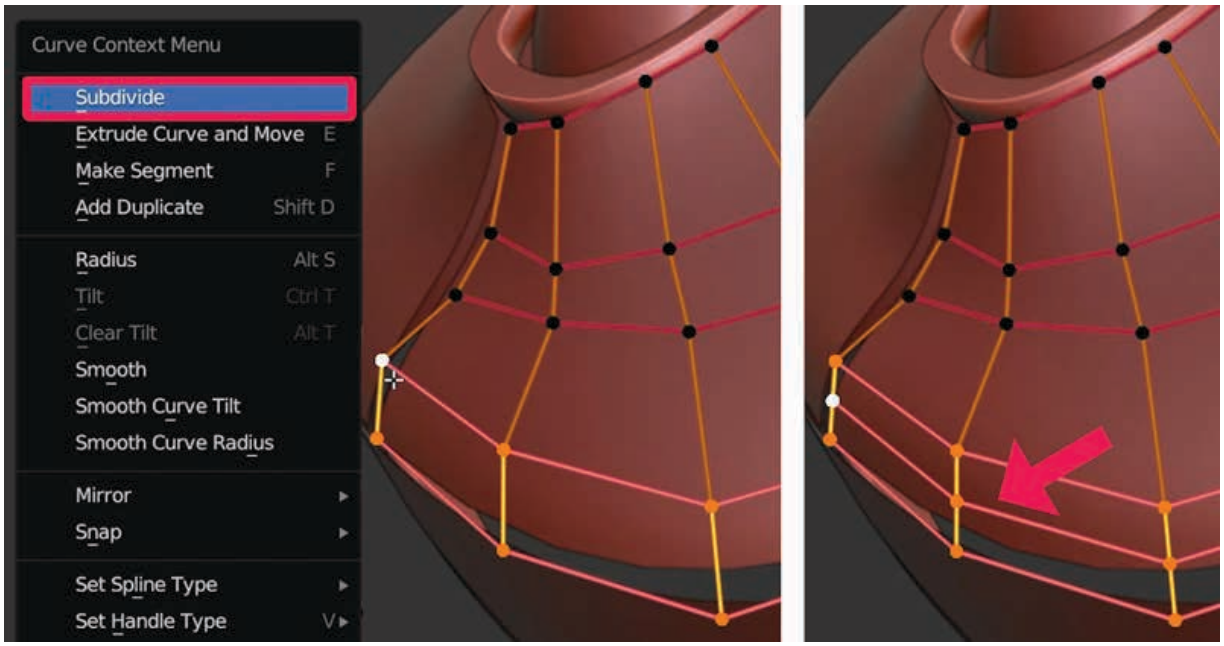
Grow the influence of the Vertices on the surface area by **increasing the Order U and V values**.



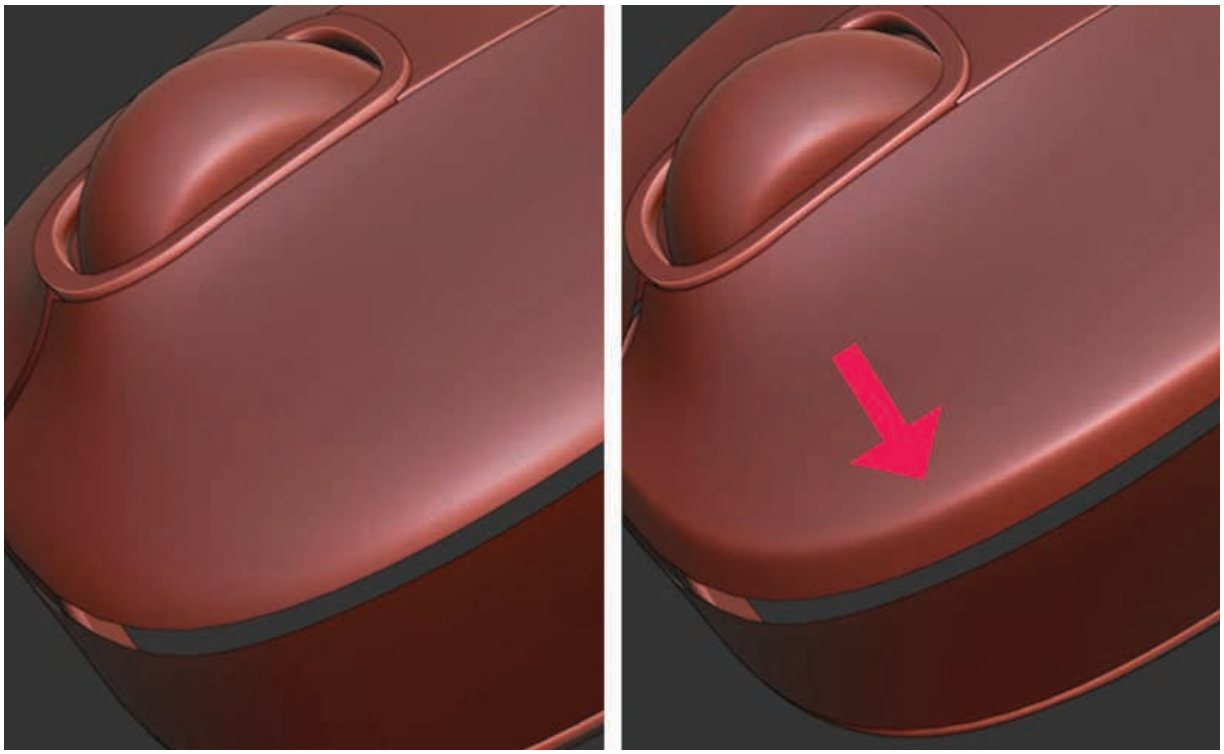
This makes the surface more smooth. Note, that it also influences the shape. You can't set this value higher than the amoun of vertices you have on either the **U (yellow)** and **V (pink)** direction. So for example, if you only have 3 vertices in the U direction, you can only set the U value to 3 at the most.



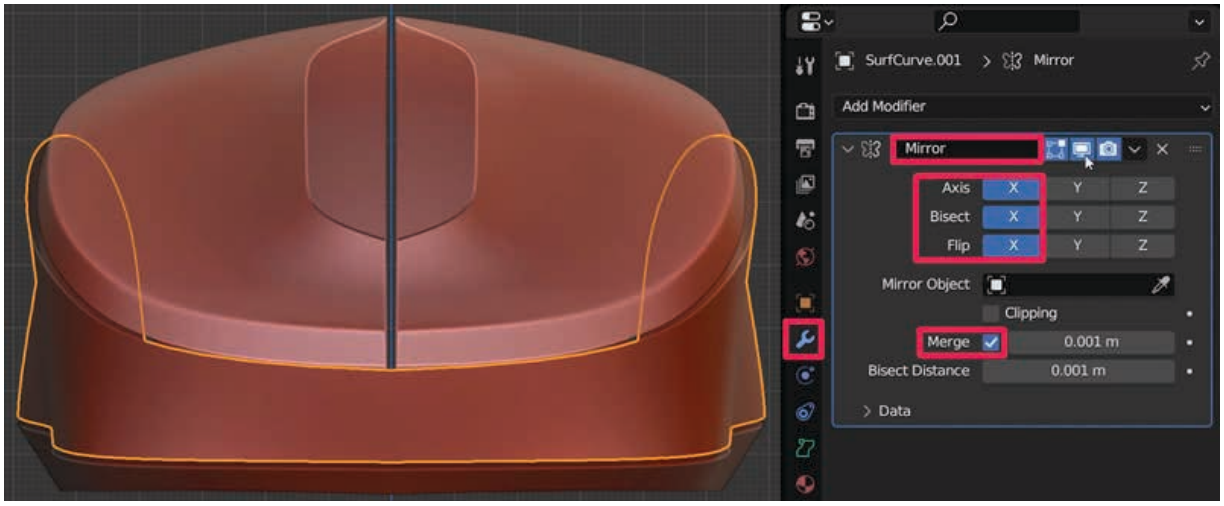
To add more detail, select Vertices, **right-click and choose Subdivide**.



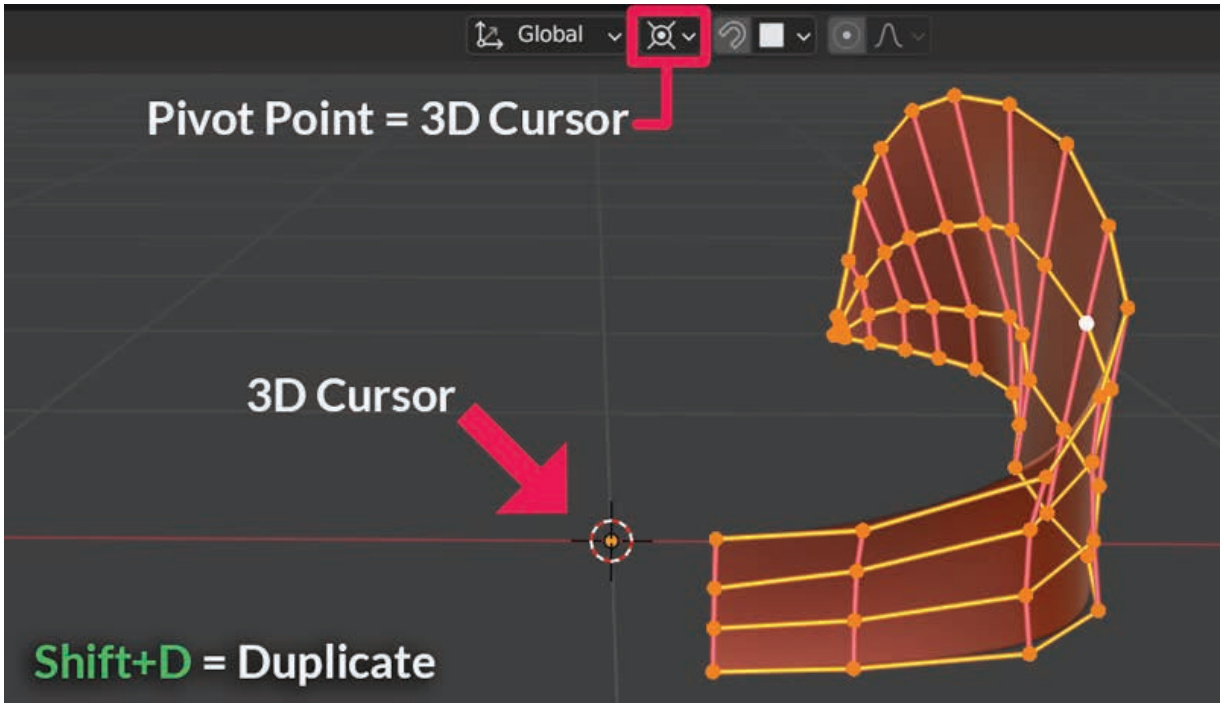
By having more vertices close together, you can create sharper, creased surfaces.



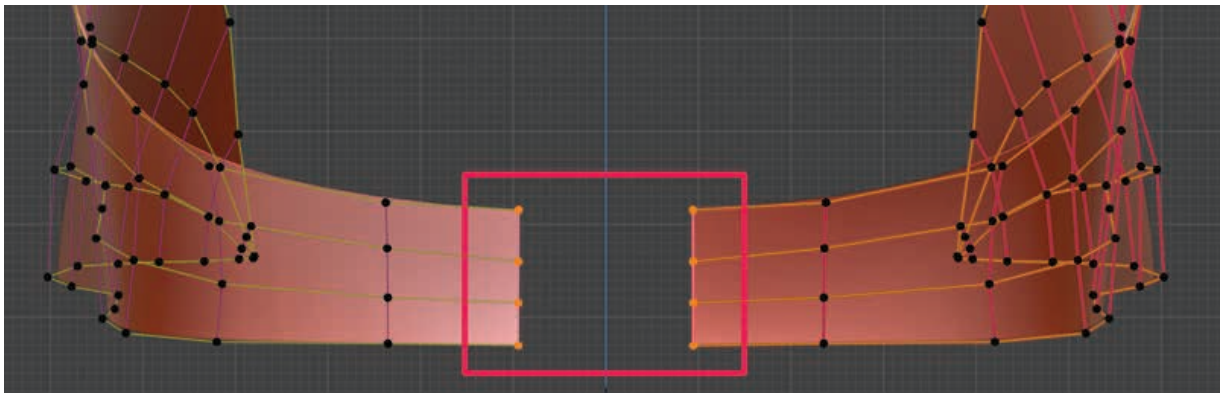
Mirror modifiers work, as long as Axis, Bisect, Flip and Merge are all enabled.



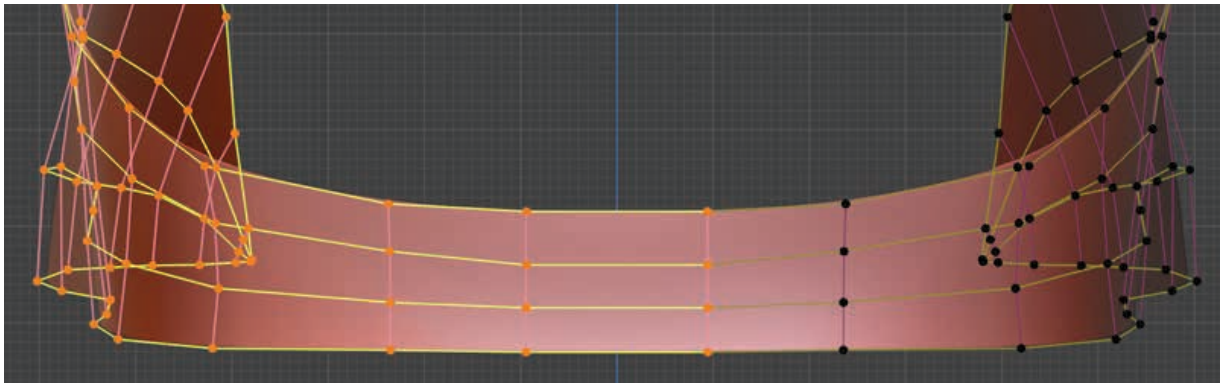
Alternatively, to mirror the Nurbs surface you can duplicate part of it with **Shift+D**, and scale it by -1 on the X-axis by **pressing S+X+[-1] <Enter>**. Do this with the pivot point set to 3D cursor, and the 3D cursor at the World Origin.



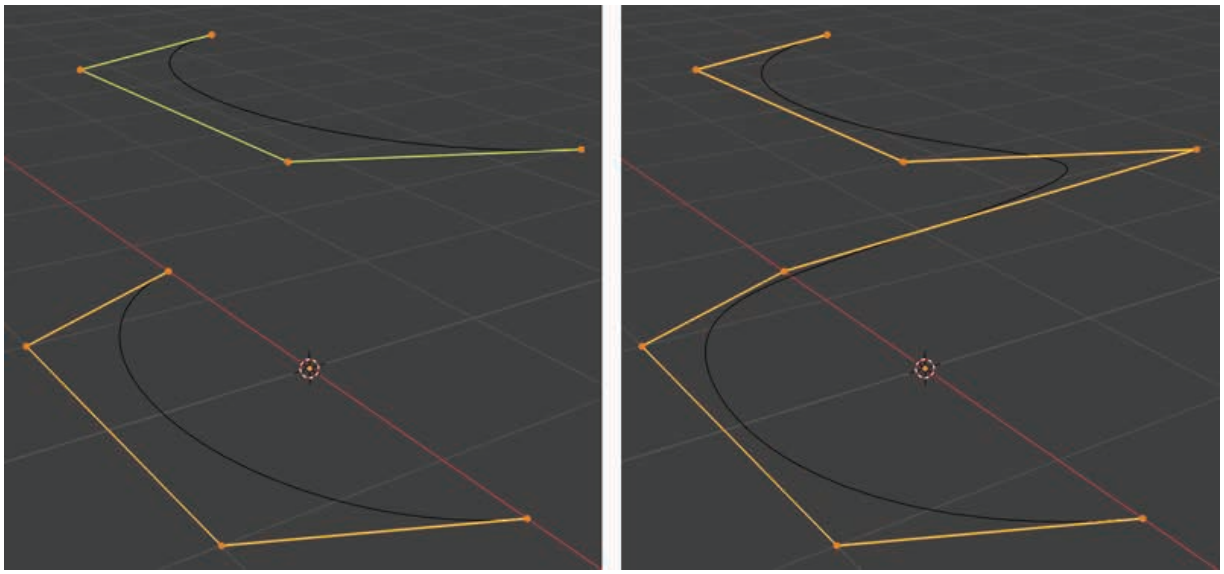
Then fill what's in between by selecting the vertices…



… and **pressing F**.

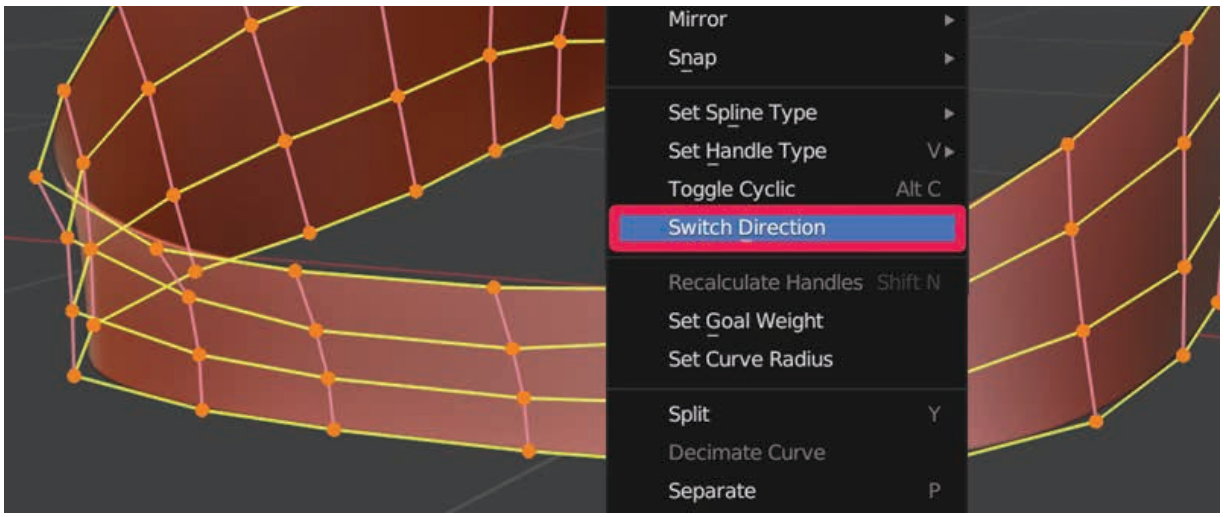


If this doesn't work, you haven't used a **Surface > Nurbs** but instead a **Curve > Nurbs**. This is the wrong menu and it won't work. This is the question I get most about this topic, so definitely play close attention to getting the correct menu. Pressing F will only connect a Nurbs Curve, it won't fill it like it does a Nurbs Surface.

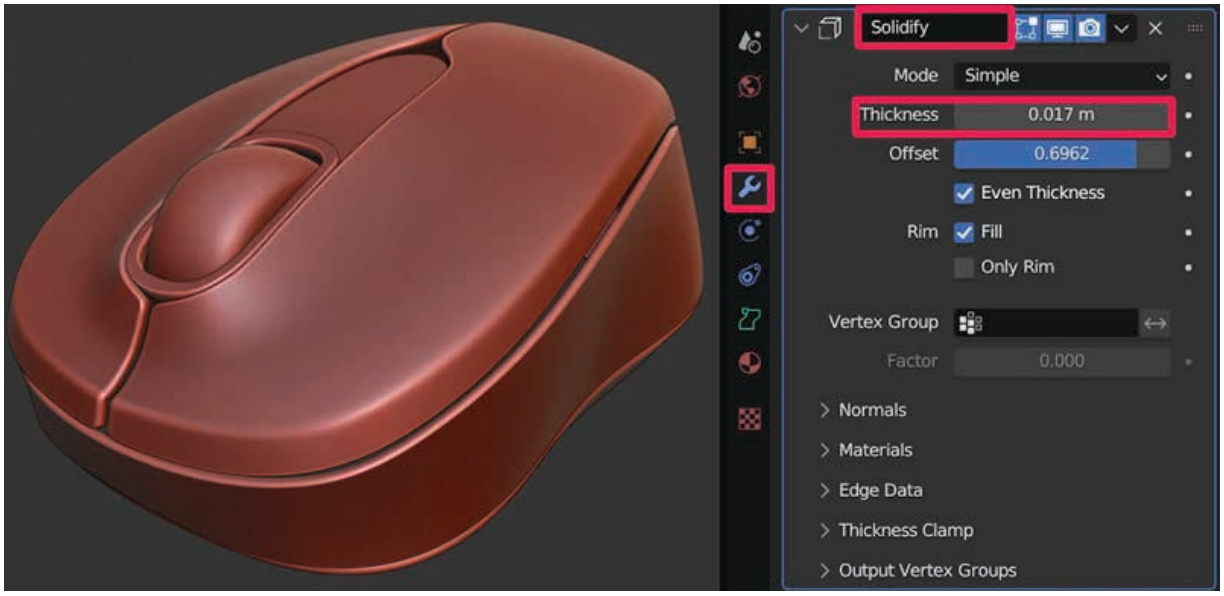


Pictured above: Pressing F only connects a Nurbs Curve, it won't fill it.

Note that filling an area may flip the Normals in an undesirable way. That may affect how the modifiers work and how the renders look. To recalculate Normals, select all and right-click, then choose Switch Direction.



For thickness, use a **Solidify modifier**. You can also add a Subdivision Surface modifier before it, to have everything a little smoother. It also helps the Solidify modifier give a better result.



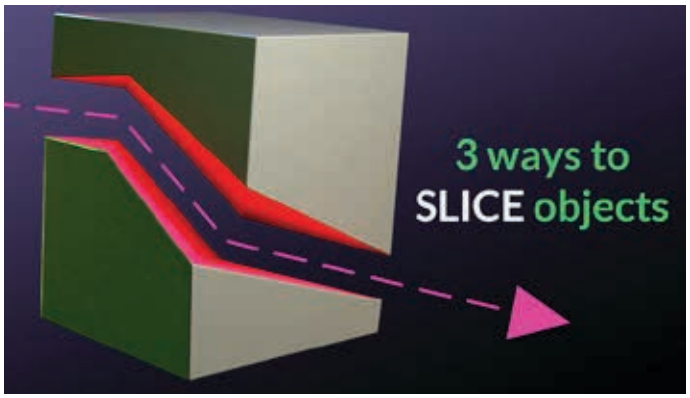
For the old video about this topic, go here. It's still relevant, but I remade it because many viewers misssed the fact that you need to go to Surface > Nurbs and not to Curve > Nurbs. Instead of answering all their comments, I decided to make a better video.

To download the mouse I modeled for this topic, click here.

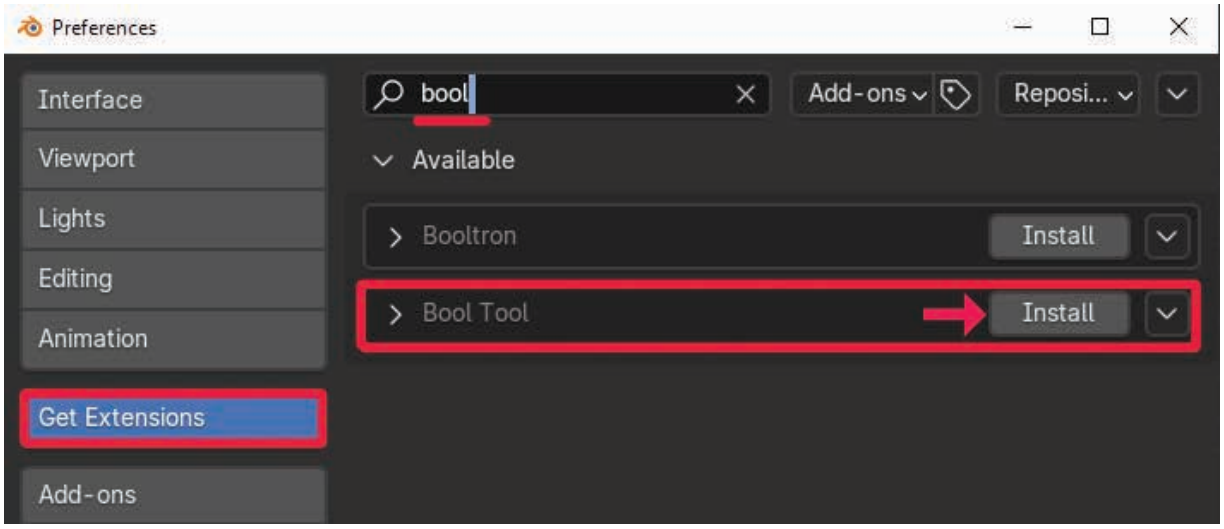
BOOLEANS

3 ways to slice objects

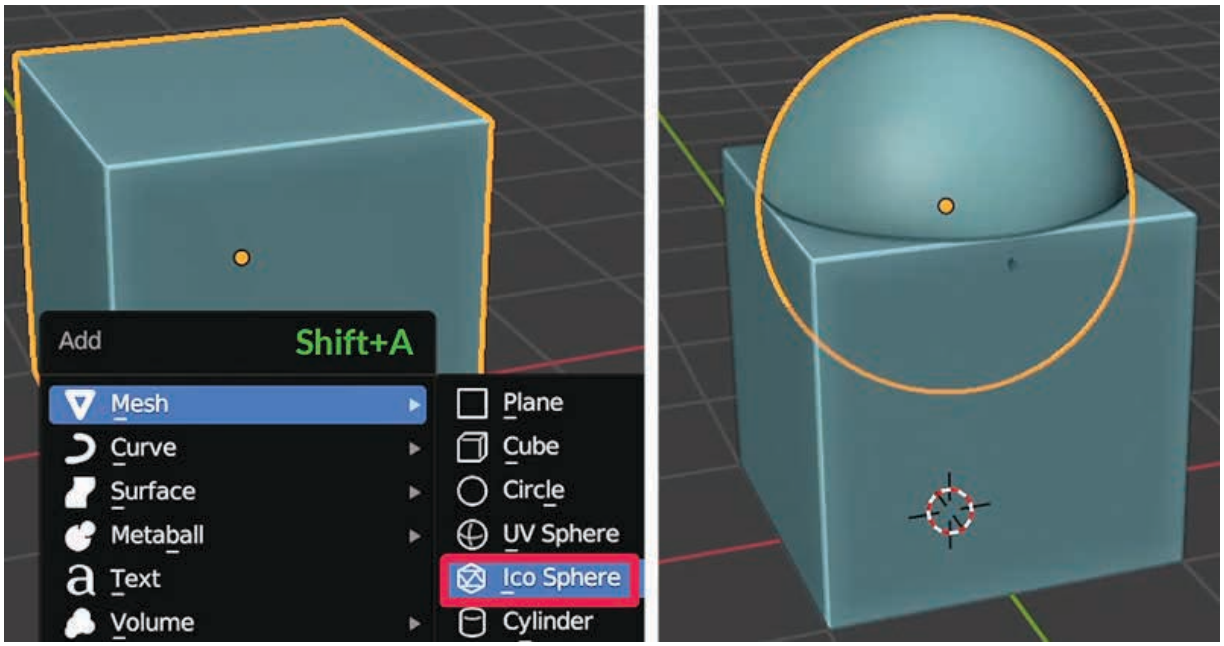
Let's look at several ways to "slice" objects in Blender, using add-ons that come with Blender by default: Bool Tool and Carver.



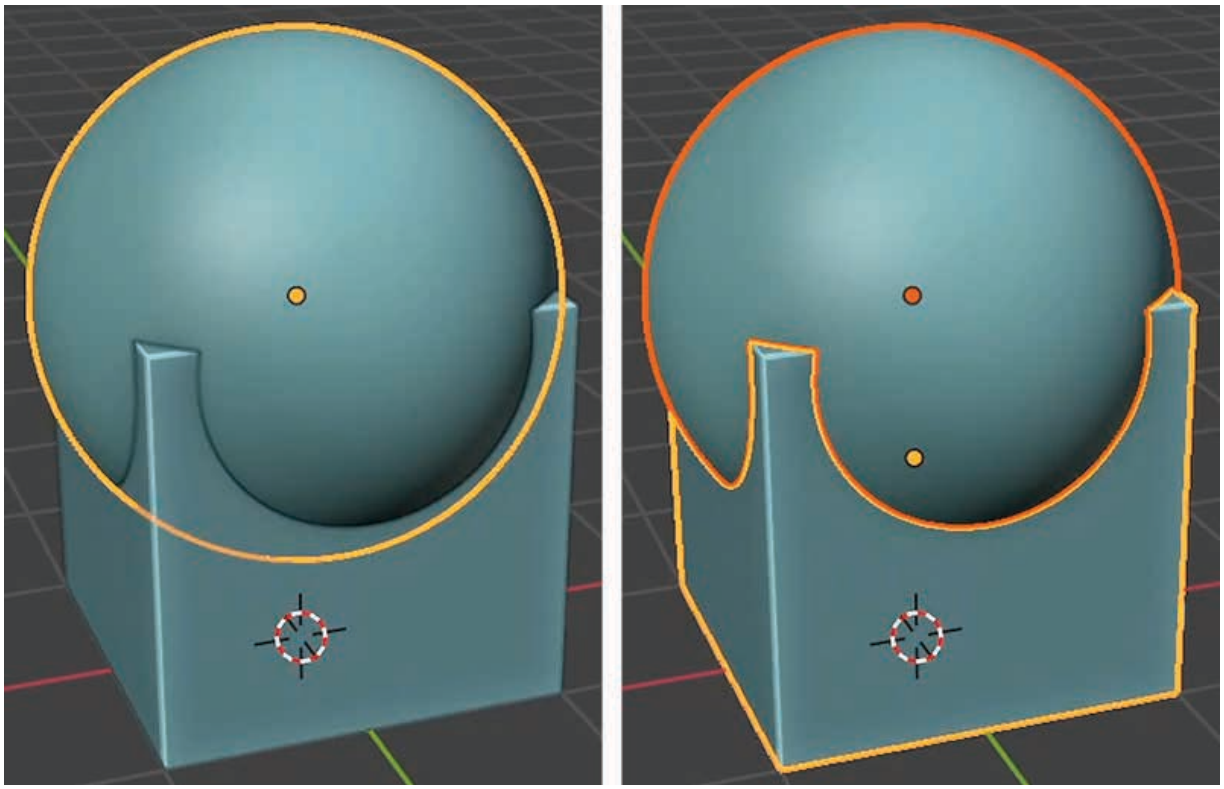
Make sure the **Bool Tool** cutter add-on that comes with Blender, is enabled in Preferences.



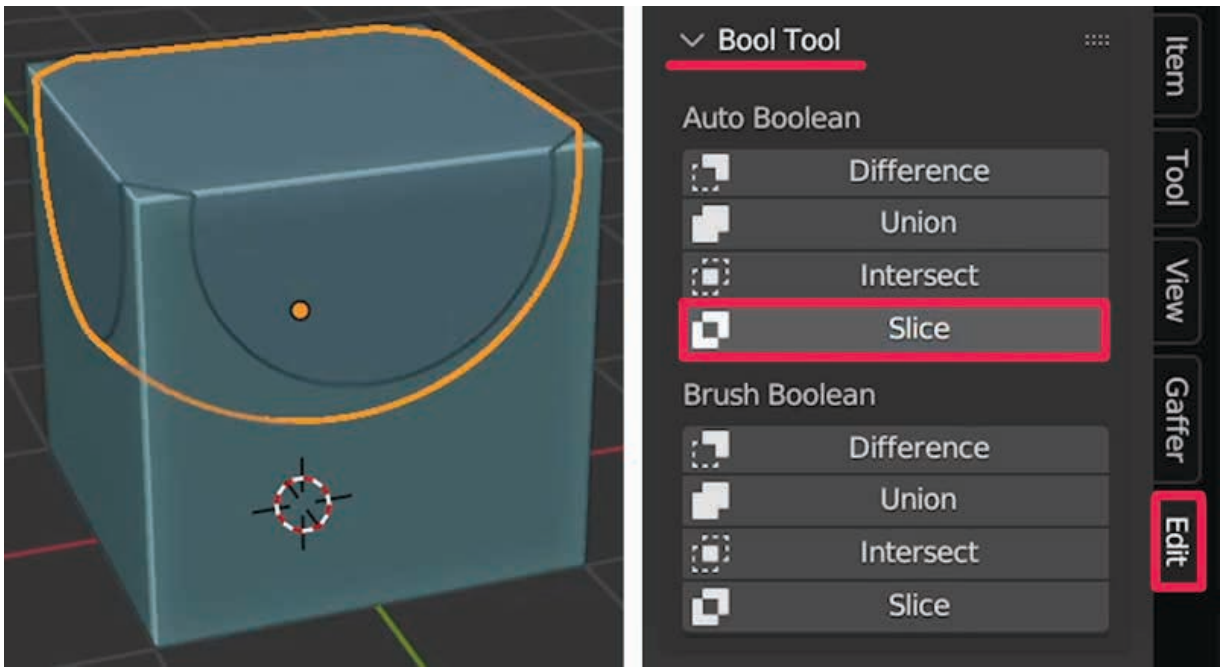
Add any object to use as a cutter, and transform it to correspond with how you want to slice. When I say transform, I just mean move, scale and/or rotate it. Although you could also go further and change its shape in Edit Mode. For this example, I will just move and scale an **Icosphere**.



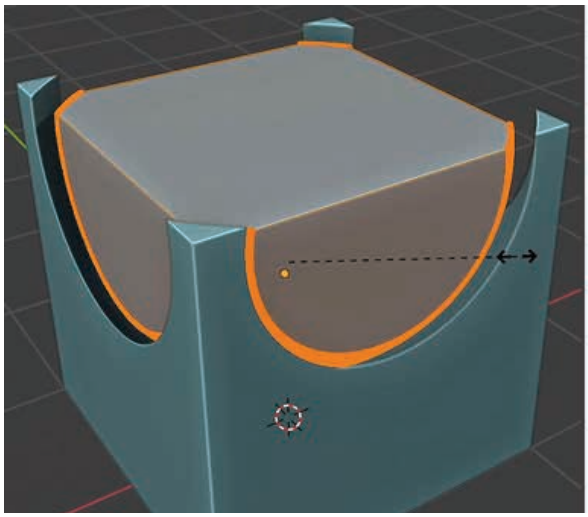
Select the cutter (the object you want to use to cut, in this example the Icosphere), **hold Shift** and select the object that you want to slice into, in this example the Cube.

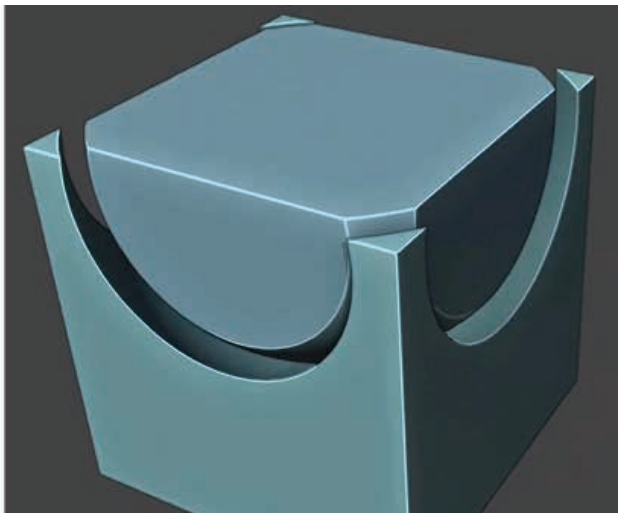


You can find the **Bool Tool** options in the Option Panel under the Edit tab. In the Bool Tool menu under the **Auto Boolean** options, click on **Slice**.



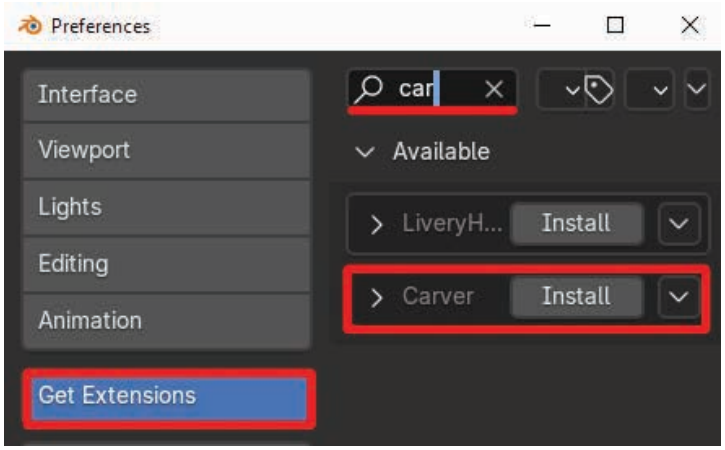
This creates a sliced effect on the geometry. You can scale down what's left of the Icosphere to create a bigger gap between both objects.



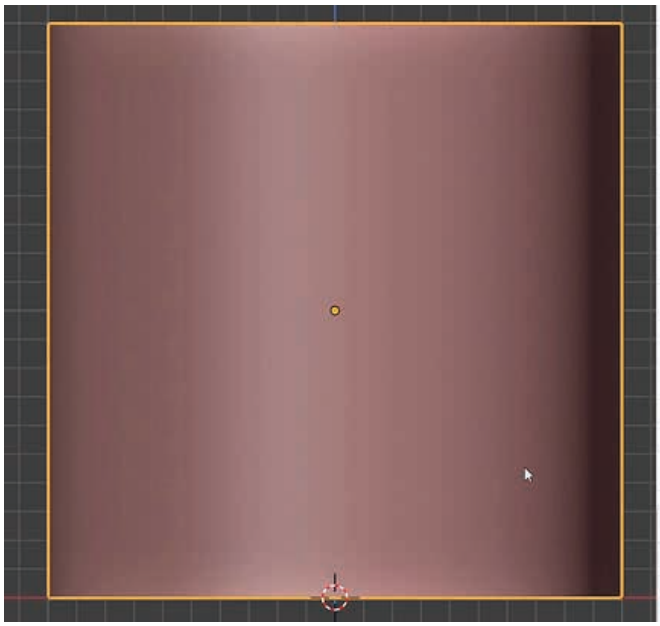


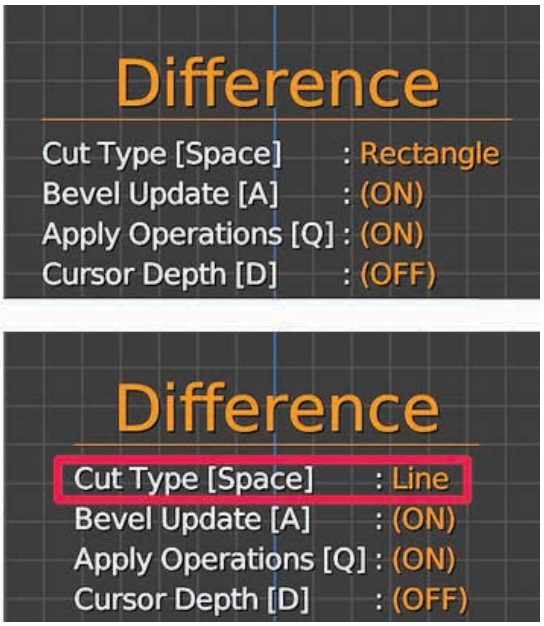
This next method requires that you enable the add-on **Carver**, which also comes with Blender by default.

This needs to be installed in the **Get Extensions** tab in Preferences.

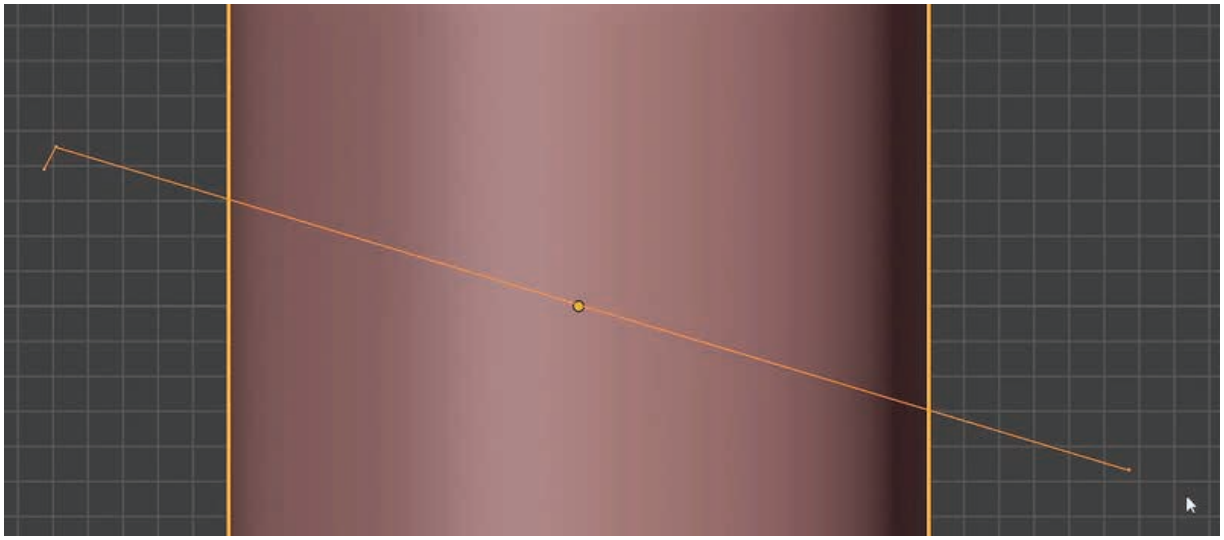


Select an object to cut into. **Press Shift+Ctrl+X** to call the Carver add-on. By default it uses a **Rectangle** to cut. **Pressing the Spacebar** once, enables **Line** cutter mode.

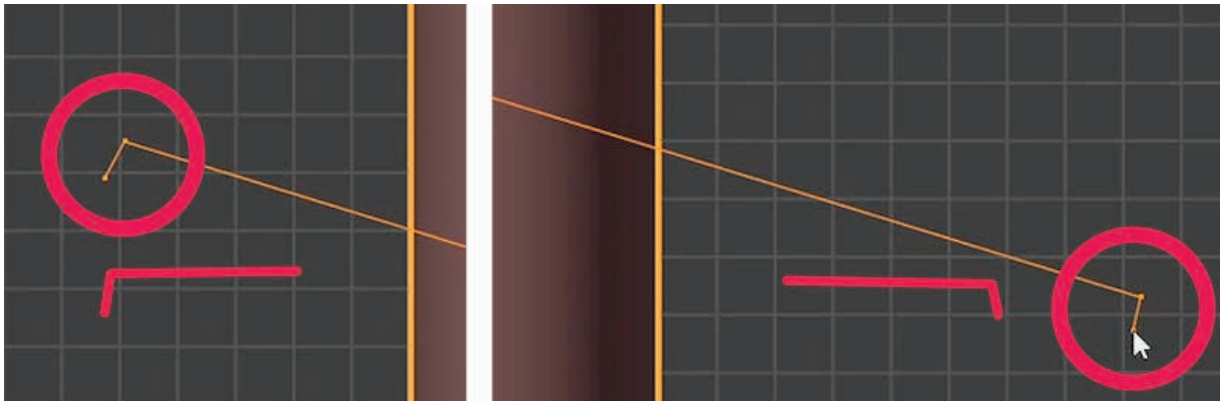




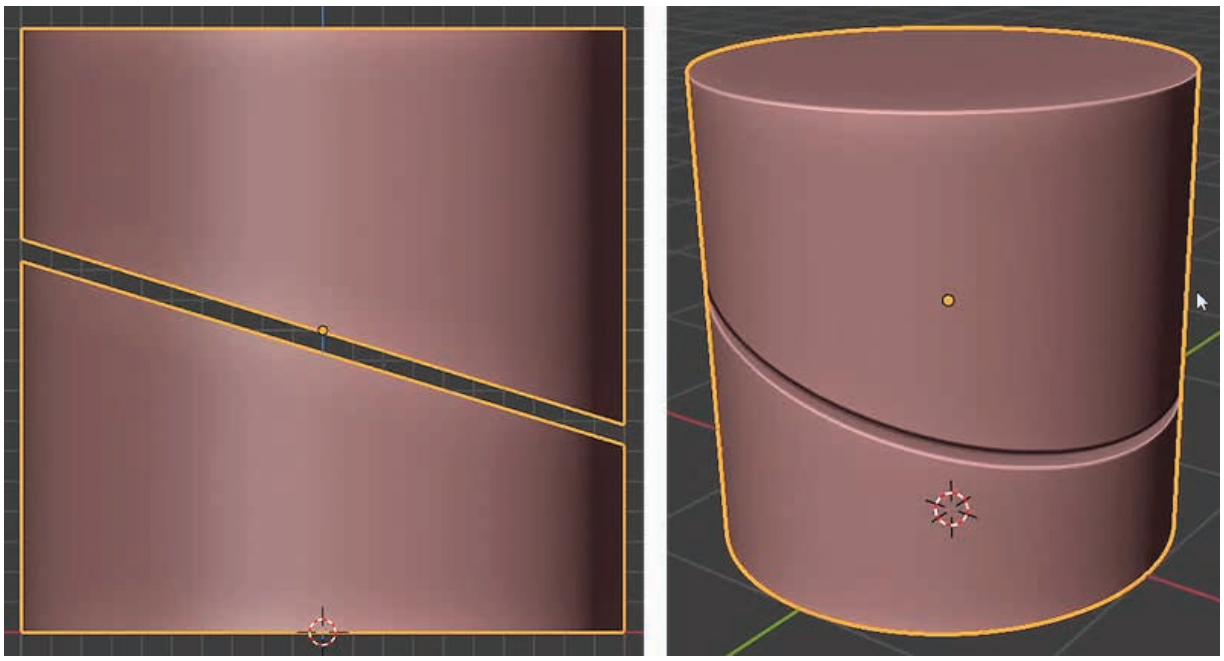
Left-mouse-click to draw a cutter line.



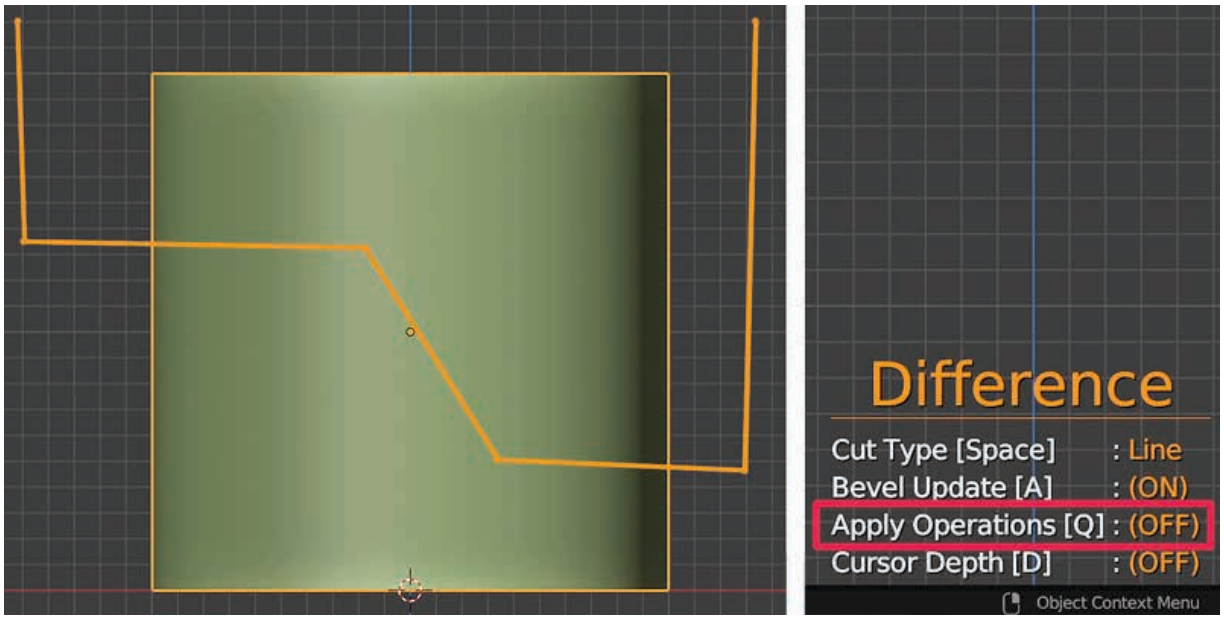
You can add some thickness to the cut by adding short lines at the start and end at roughly 90° angles.



Press the **Spacebar** again to use the line as a cutter object. Exit the Carver tool by pressing **Escape**. Now there's a gap where you used the cutter line.

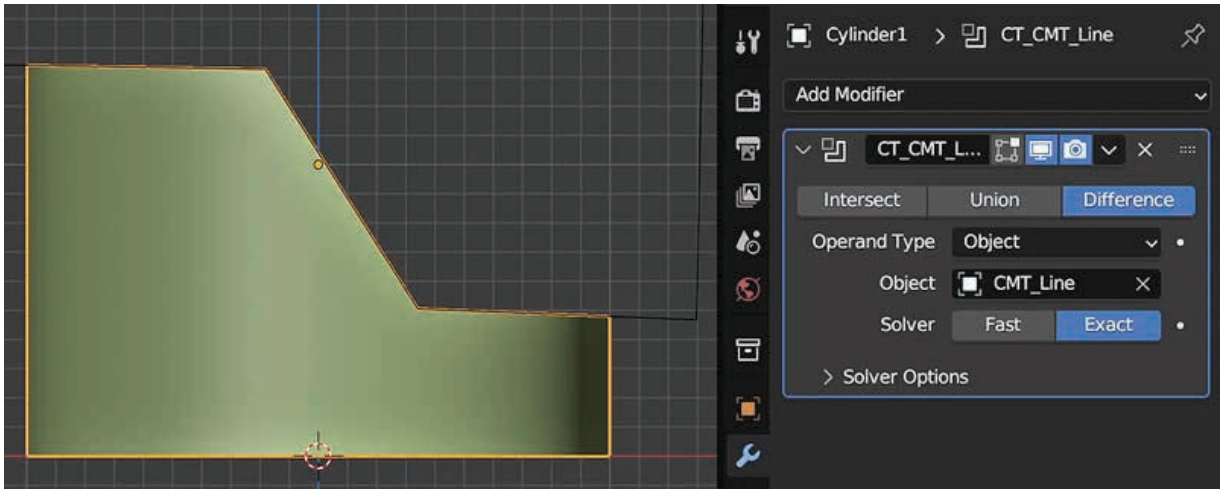


If you want to cut a line with several angles, **press Q** to make it non-destructive. This toggles the "**Apply Operations**" option on / off.



Now we get a **Boolean modifier** on the object, and the top half has been cut way, with the line turned into Cutter object that is displayed as a a wireframe.

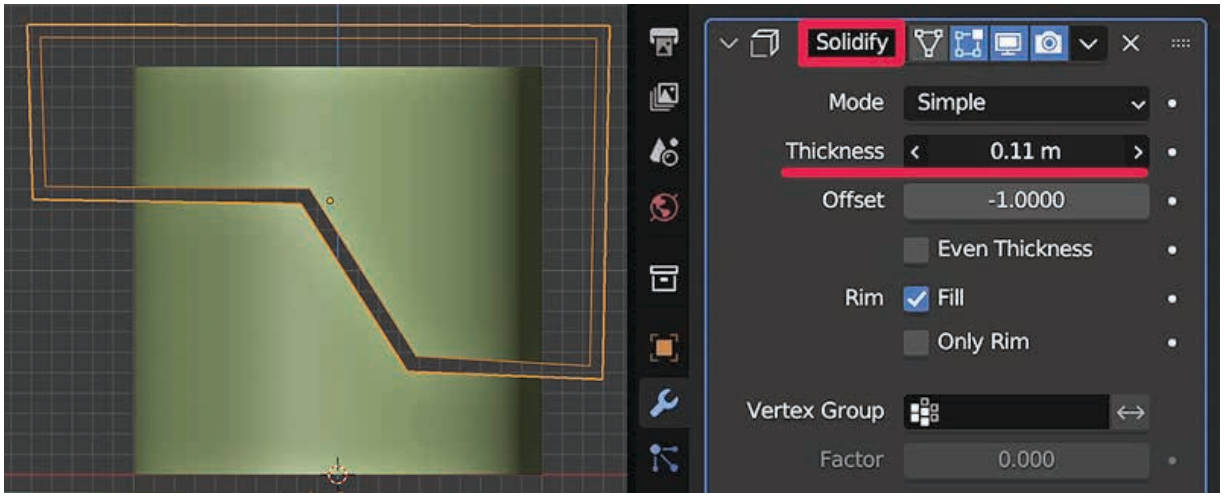
The Boolean modifier has some kind of special name because we used the Carver add-on, but other than that, it's just a normal Boolean modifier like what you could add manually from the Add Modifier drop-down menu.



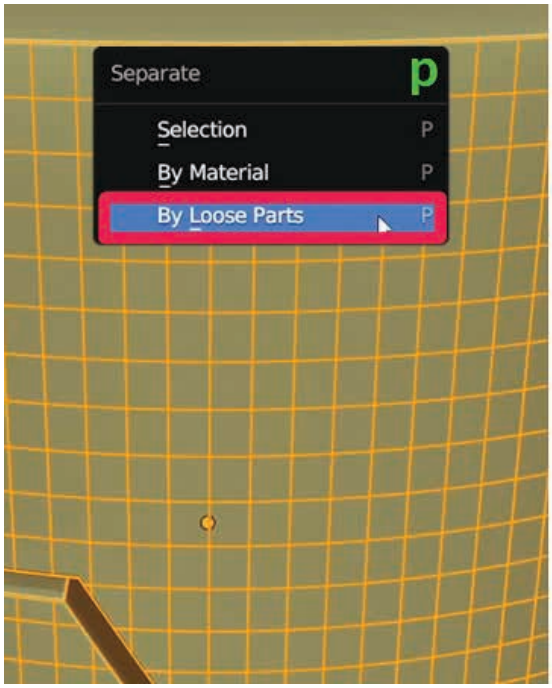
Note: I've tested all these steps in **Blender 4.3,** and it all still works. However, I get a lot of confused comments about this video.

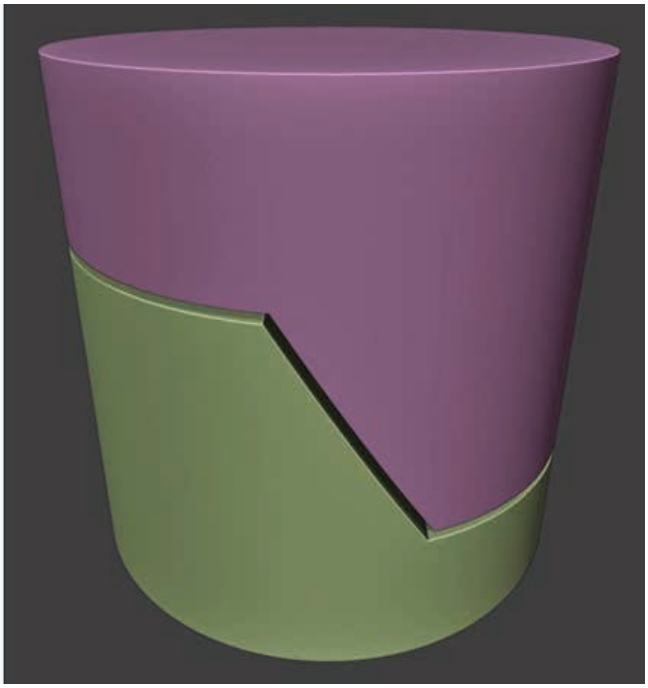
So if there's something that doesn't work for you, please leave a comment describing the issue. If there's extra steps that I need to clarify, I'd love to know!

Next, add a **Solidify modifier** to the cutter object. This restores the top half of the object. Changing the **Thickness** value further gives us control over how big the gap is between the two halves of the object.



If you want to have separate objects, apply the modifier and select all in Edit Mode. Then **press P** for the **Separate** menu, and choose **By Loose Parts**.



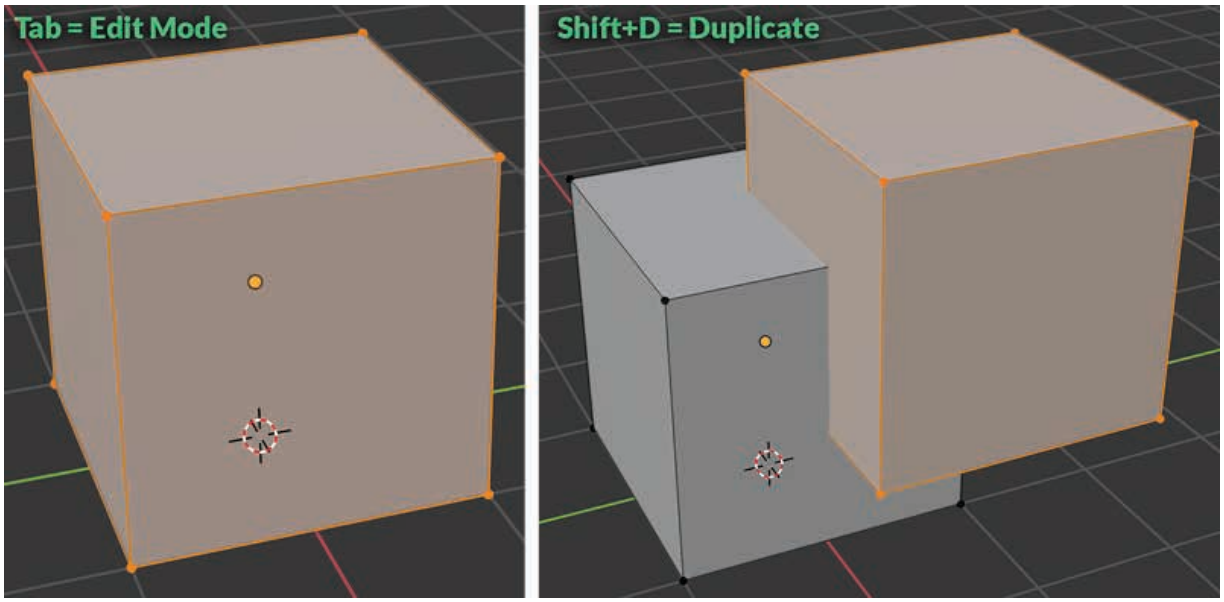


The Fastest Boolean Workflow

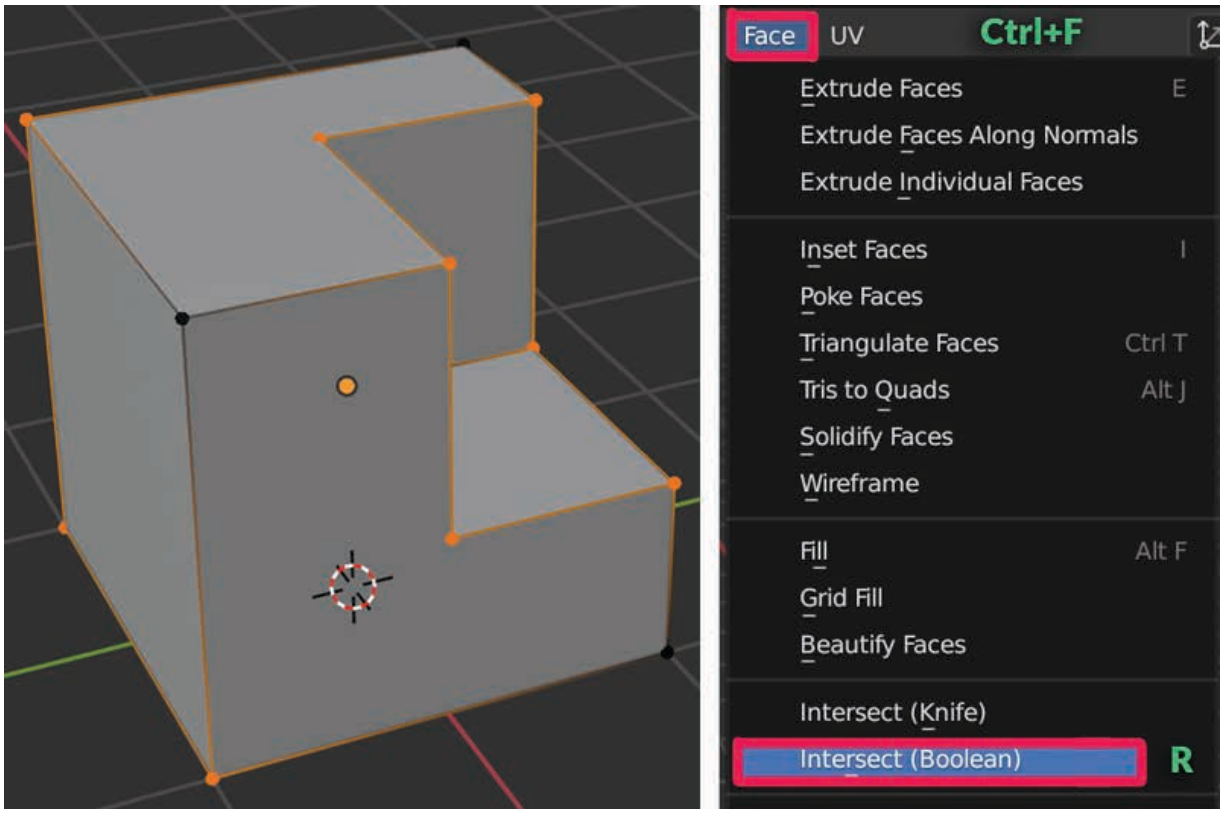
This relatively unknown technique for boolean cutting is fast - one keyboard shortcut and that's it. No modifiers to apply, just keep cutting and create something quickly.

Example made with this technique by Tushar Ganesh.

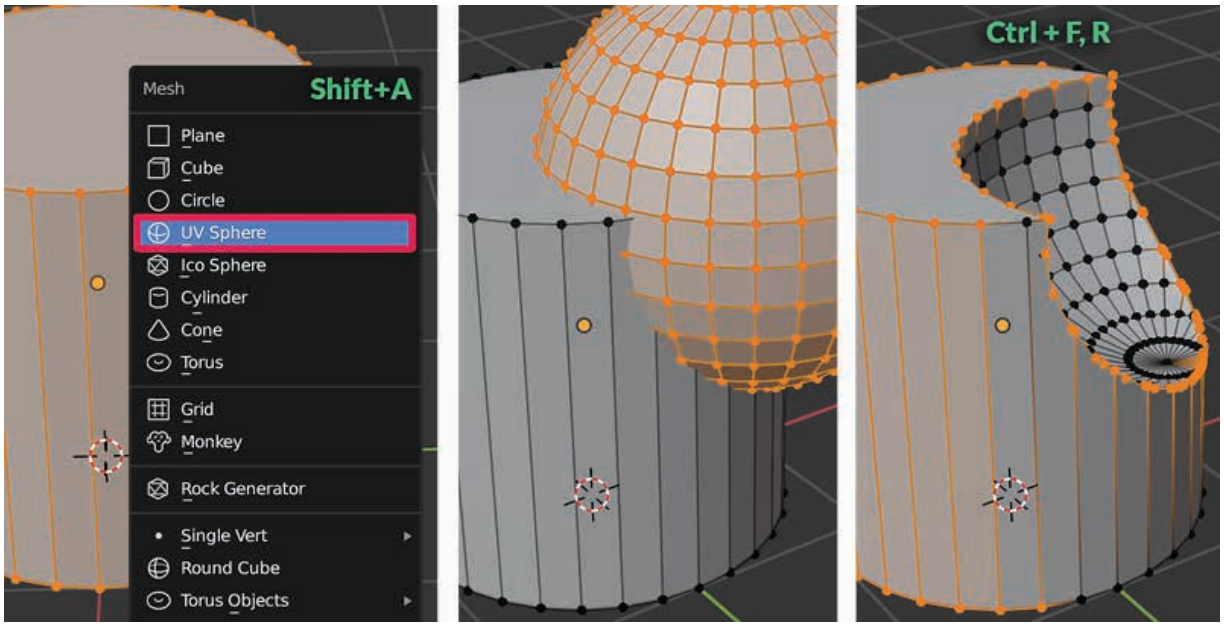
In Edit Mode, duplicate a mesh with **Shift+D** and move it so that it partly overlaps the orginal mesh.



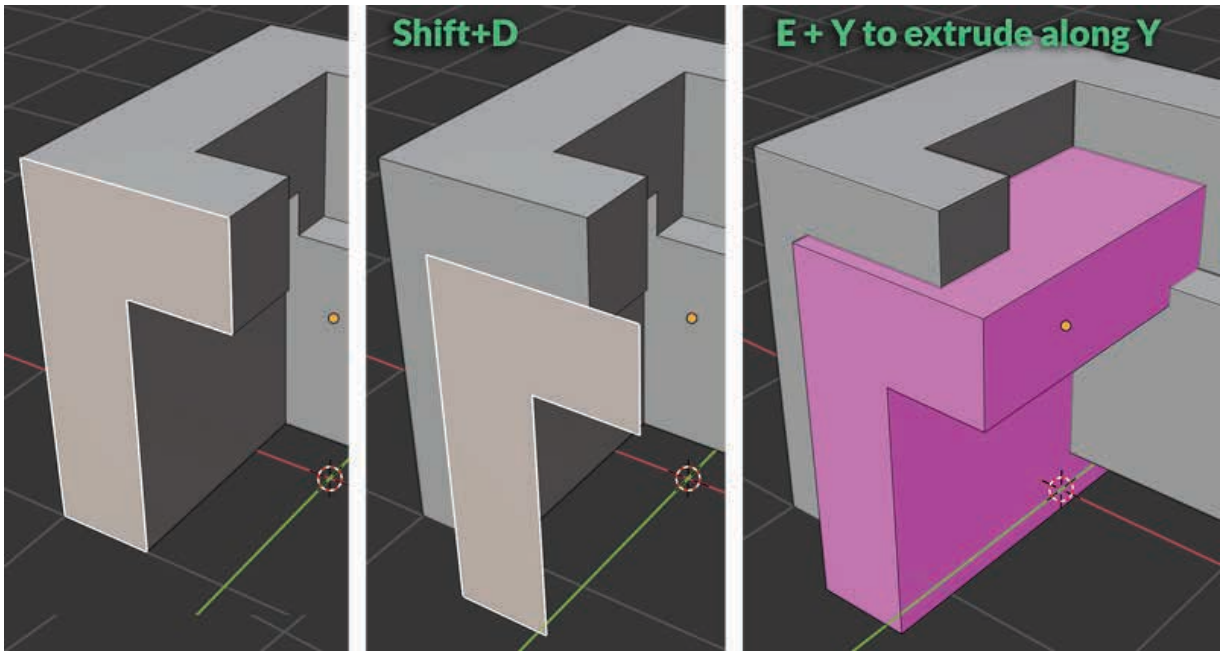
Press Ctrl+F and R to do a Boolean cut. As you can see in the screenshot below right, the keyboard shortcut Ctrl+F simply opens the Face options, then R is the underlined part of inteRsect. Of course you can also just press Ctrl+F and choose that option.



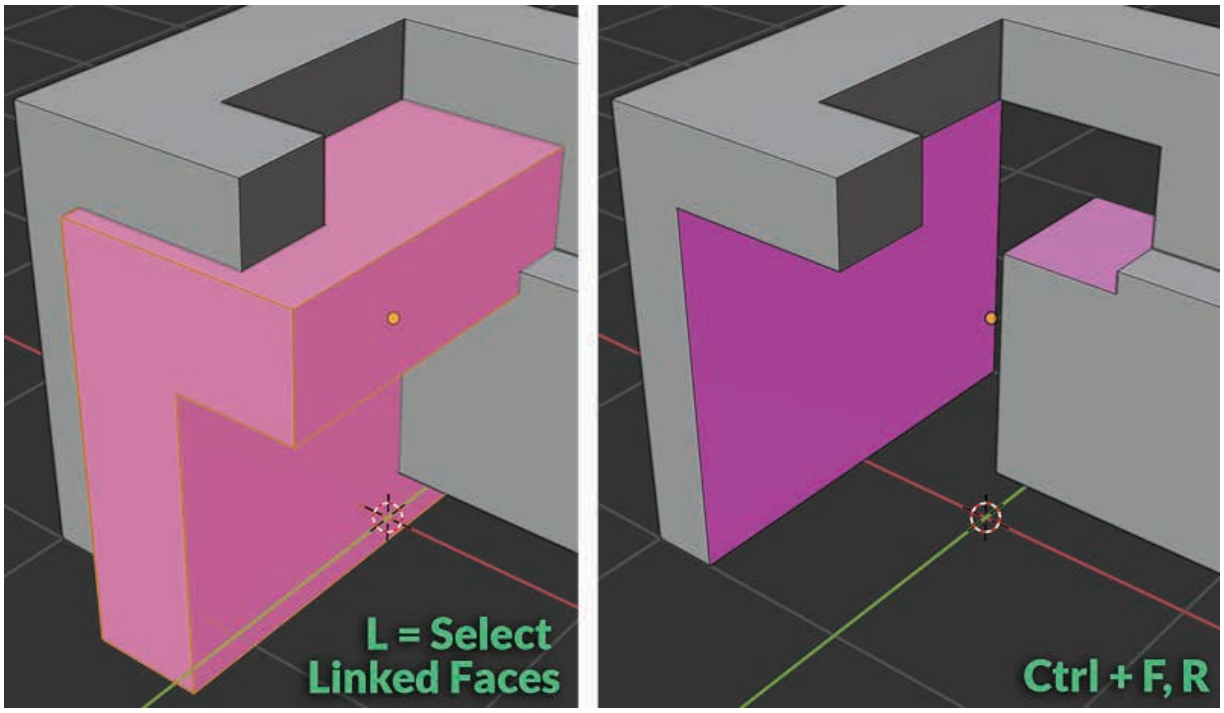
That's the fastest method for doing a boolean cut in Blender. Of course, it's also destructive as it doesn't use any modifiers. If you prefer a non-destructive Boolean workflow, check out the BoolTool add-on that comes with Blender by default. You can also add other meshes in Edit Mode by **pressing Shift+A**.



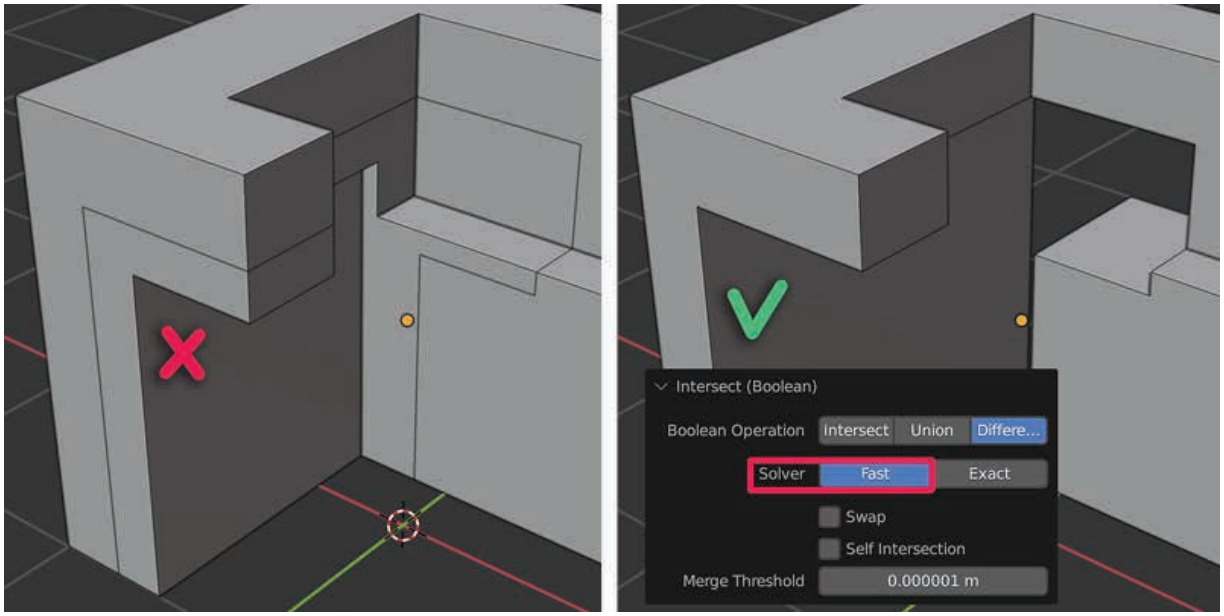
Although destructive, this is a fast way to work and it's especially convenient that you can just use a weirdly shaped Face, extrude it and then use it as a cutter object. To do that, select the Face, **press Shift+D** to duplicate and then **E to extrude** it.



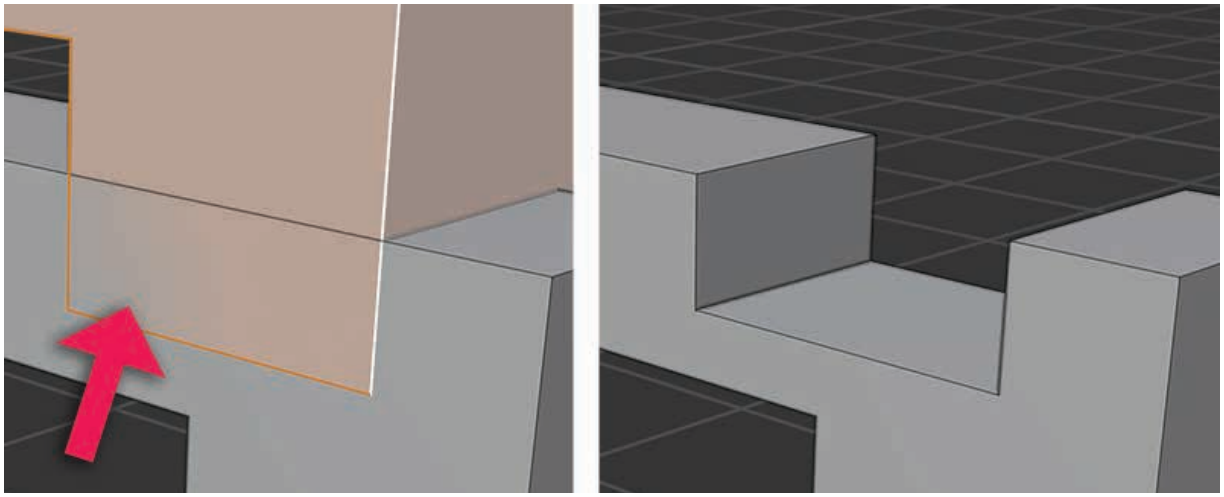
Then hover the mouse cursor over it and **press L** to select the entire mesh island (Ctrl+L works too, if you've already got a Face selected like in this case - pressing just L works also when you've got no Faces selected as long as you hover over the mesh).



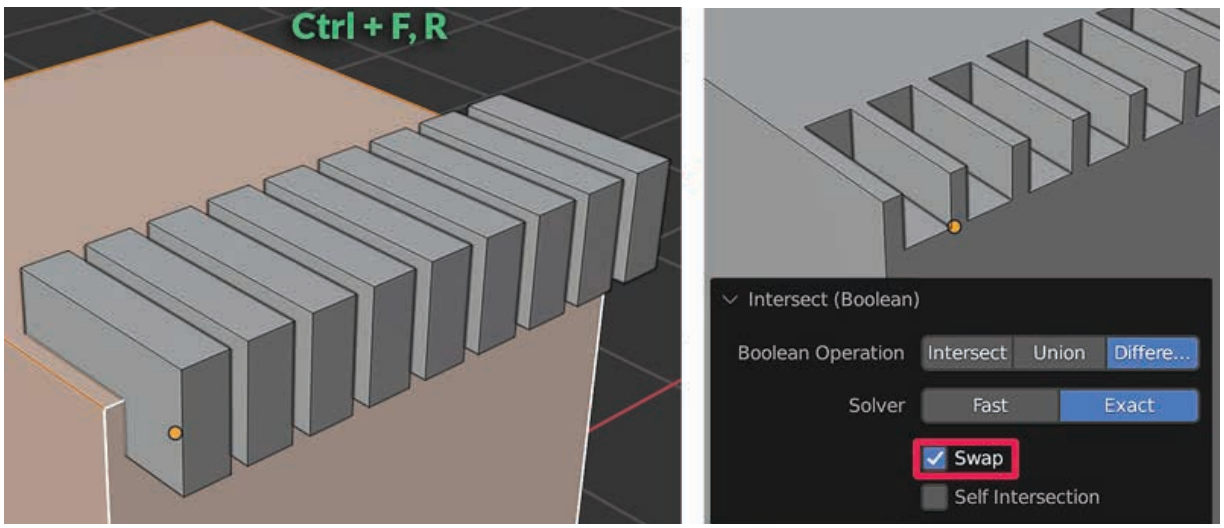
Note, that the reason for the colorful Faces is that the normals were flipped because of extruding in the "wrong" direction (more on this below). You can try using **the "Fast" Solver** in case you run into trouble using a mesh with flipped normals, like the result below left.



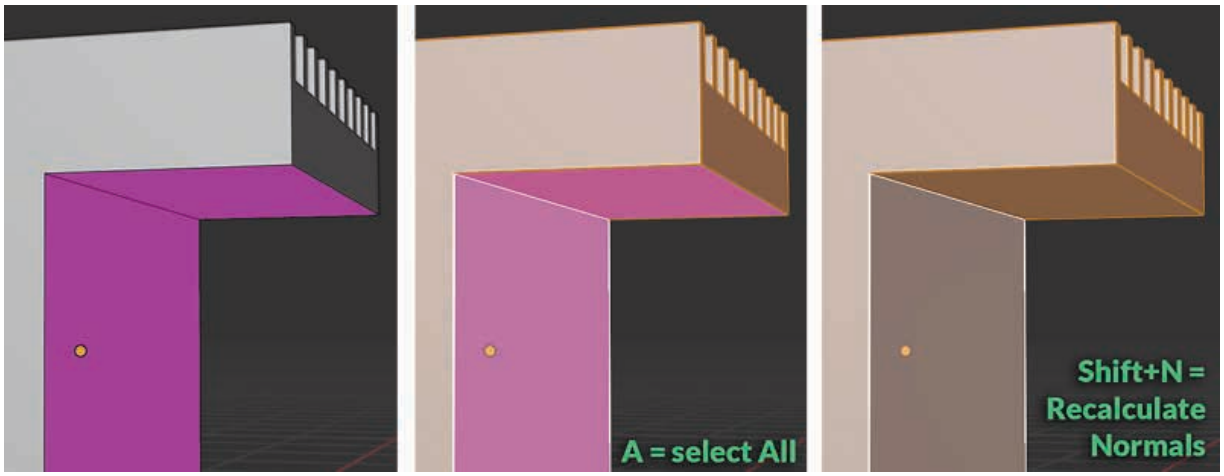
Trying out a different solver can also be useful when trying to do a Boolean Difference operation with meshes that are overlapping. In general, if it doesn't work: trying the other solver usually is the answer.



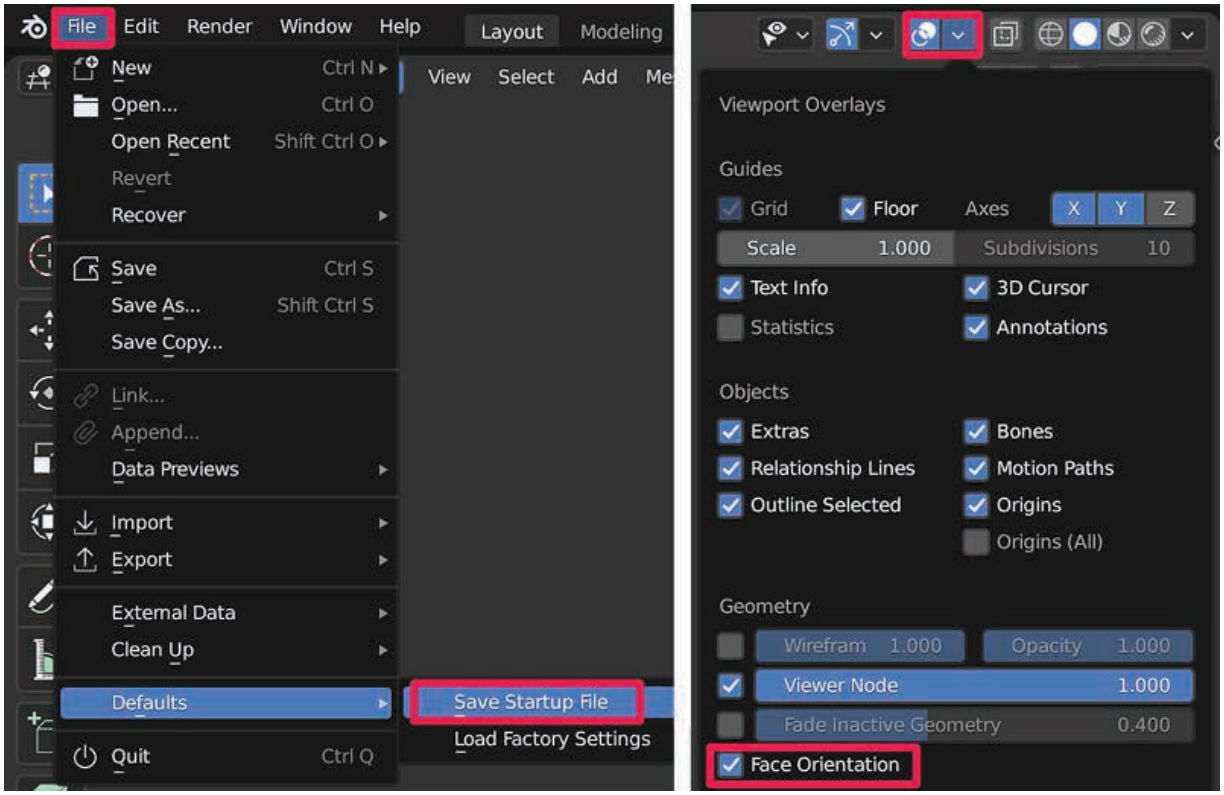
If you repeat a duplication with **Shift+R** a few times, use the other object to cut instead of selecting all those duplicates. Then check "**Swap**" after the boolean operation.



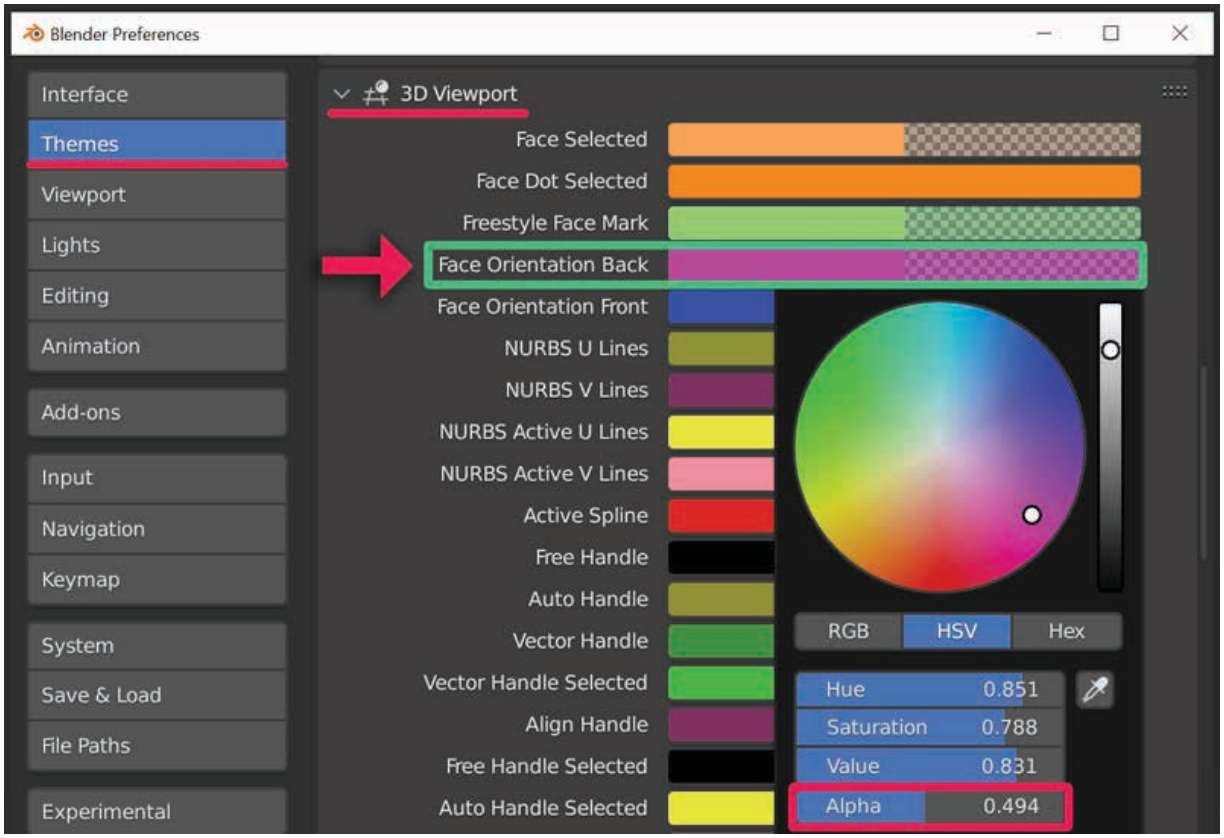
Select all with A and **press Shift+N** to Recalculate Normals if you've got flipped normals after boolean cutting.



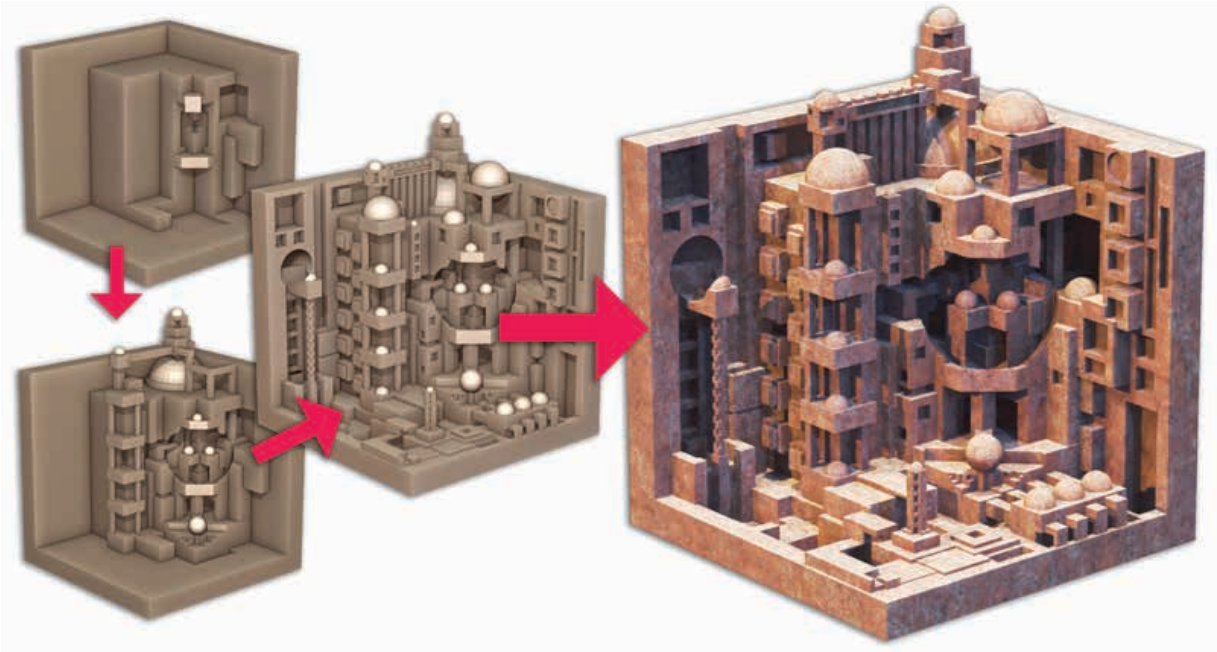
Bonus tip, if you want it to be really obvious when Normals are flipped, turn on **Face Orientation** in a fresh scene and **save the Startup file**. That way it'll be on every time you start up Blender.



Secondly, give "**Face Orientation Back**" a festive color and raise the Alpha value to something like 0.5.



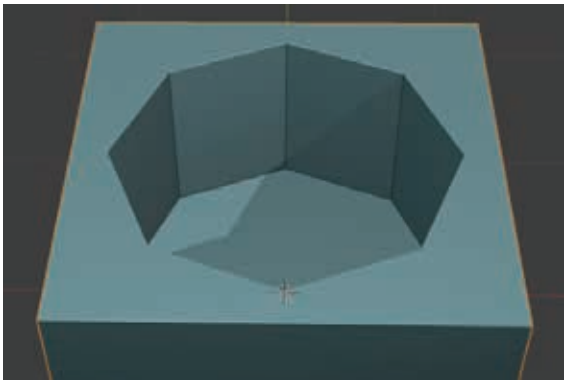
With this technique, you can really quickly come up with complicated shapes.



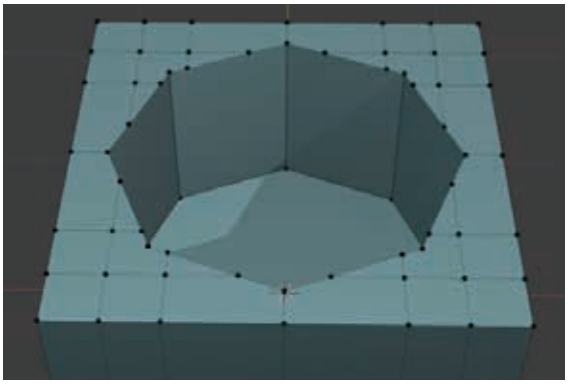
Check out the greeble workshop from Lino Thomas from Blender Conference 2022 for a great demo of this fast Boolean technique (as well as some other mind-blowing tricks).

Fixing Boolean Issues

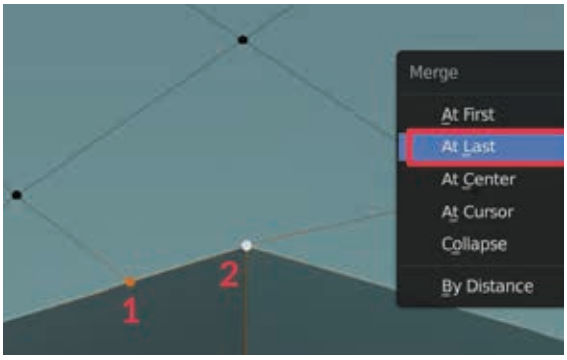
This hole is the result of a Boolean operation.



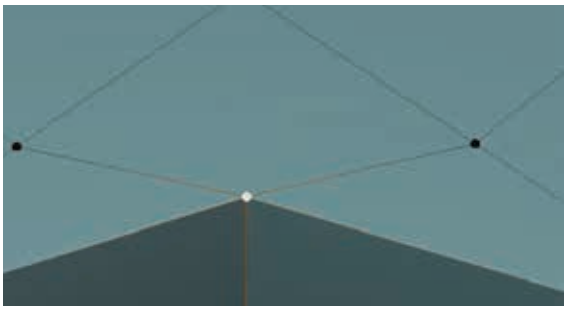
As you can see, some unnecessary vertices are left over.



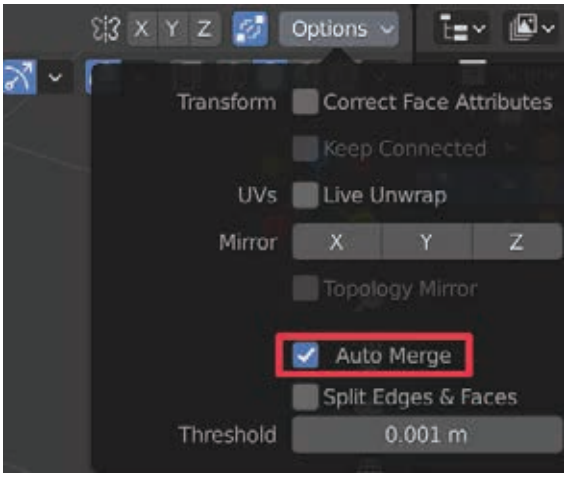
One way to get rid of these extra vertices is to merge them by **pressing M**.



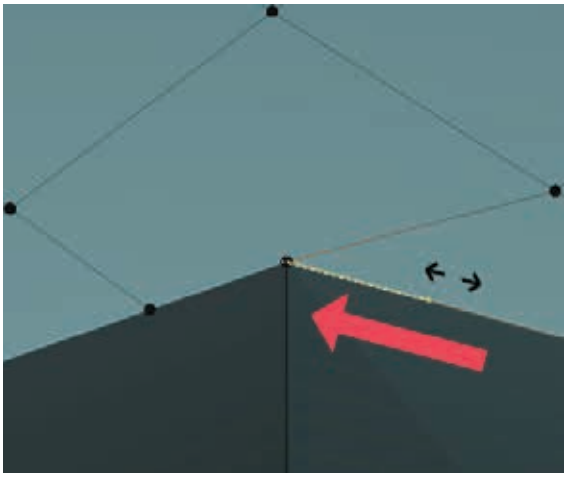
To keep the shape of the mesh from distorting, select the unnecessary vertex first, then choose merge **At Last**.



You can also enable **Auto Merge** in the Options drop-down (only in Edit Mode).

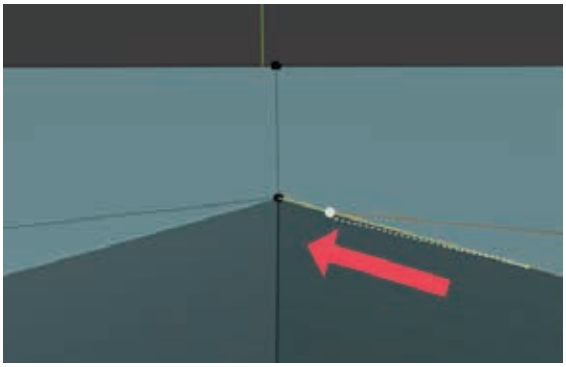


Now **press G twice** to slide a vertex towards another vertex. It auto merges.



Click here to return to the index.

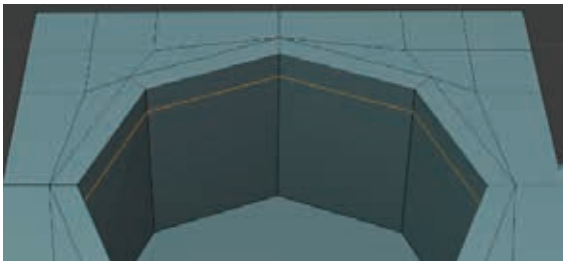
Now it's just a matter of cleaning up all the unnecessary vertices this way by sliding them towards other vertices.



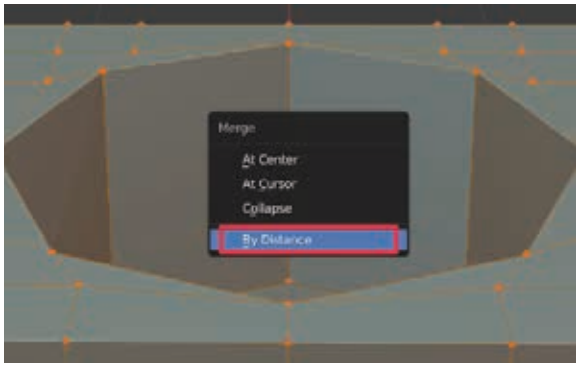
As you see there's some triangles, which would cause problems when using a Subdivision modifier. To solve this I'll create ring of quads around the hole.

Select the inner faces by selecting the bottom face and **pressing Ctrl+[+** on the numpad]. Then **press i to inset**, and right-click before moving the inset.

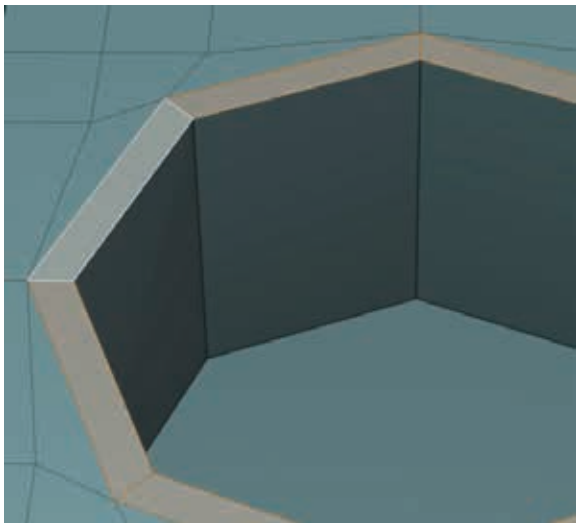
Press Ctrl+R to add a support loop on the inside and slide it up so it's closer to the edge.



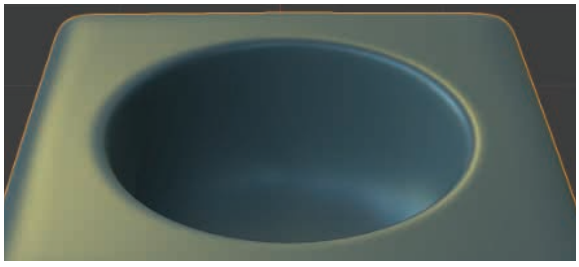
Just in case some vertices didn't merge automatically, **press M** and choose **By Distance** to remove any doubles.



It looks like nothing has changed. **Press Alt+S** to scale the inset inwards. Now you've got a ring of quad faces around the hole.



In Object Mode **press Ctrl+3** to add a Subdiv modifier with 3 levels of subdivision. Perfectly smooth!

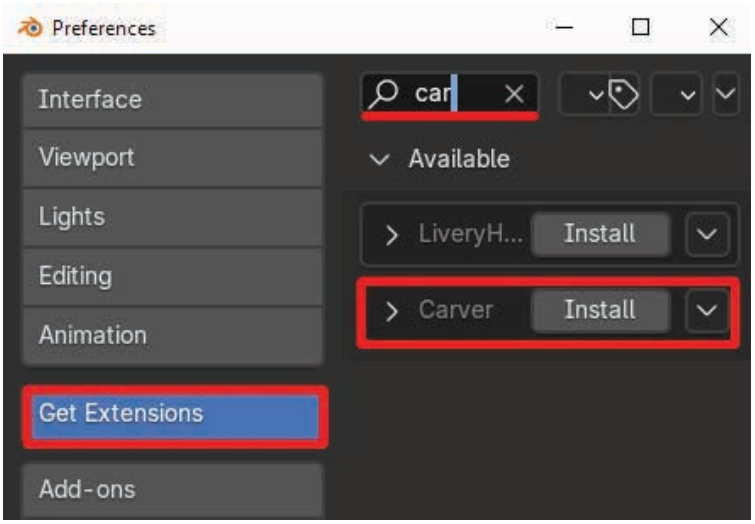


Modeling a scifigun with Carver

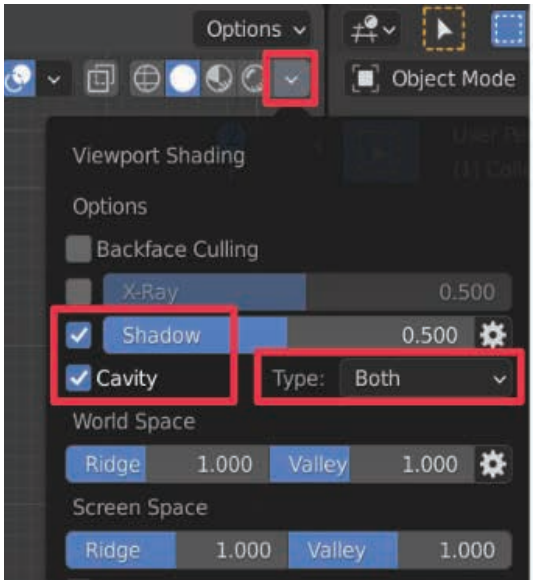
In this tutorial we'll have a look at Blender's built-in Carver tool.

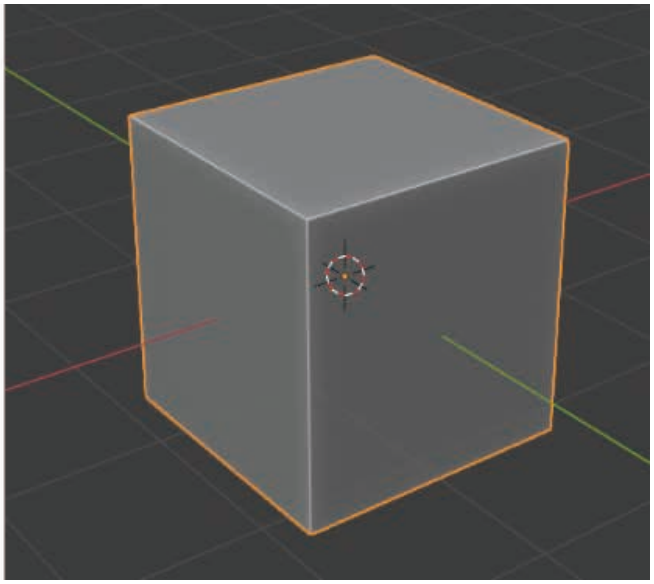
Carver allows you to quickly make boolean cuts, including some preset shapes that allow us to quickly create a cool sci-figun.

Install **Carver** in the Get Extensions tab of **Preferences**.

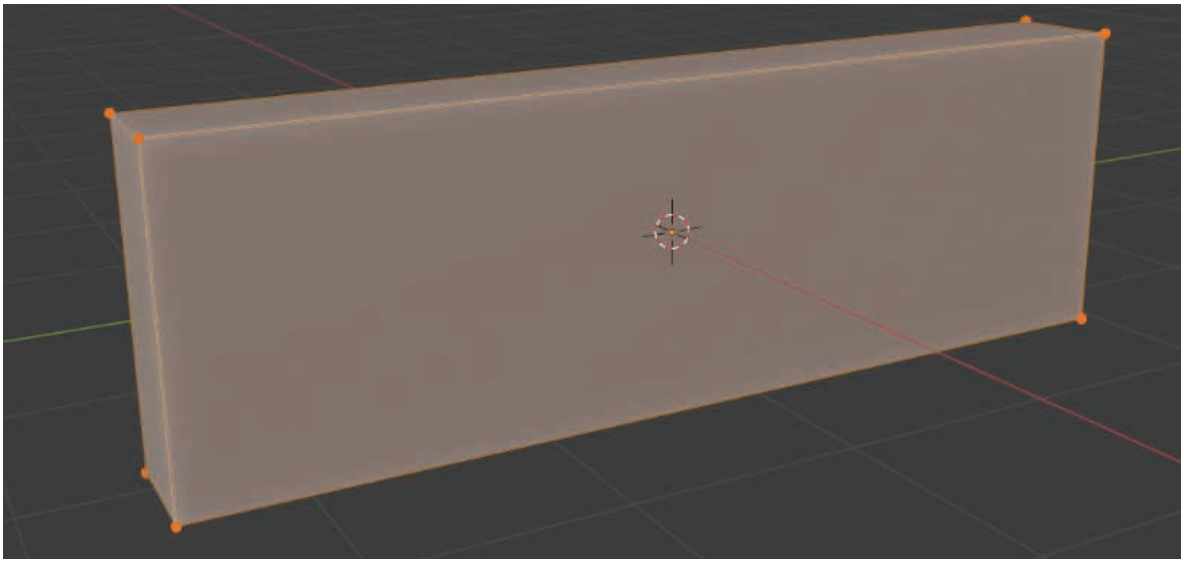


In the **Viewport Shading** drop-down menu, check the boxes for **Shadow** and **Cavity**. This way it's much easier to see the cuts you made, especially from orthographic views. Set Cavity type to **Both**.

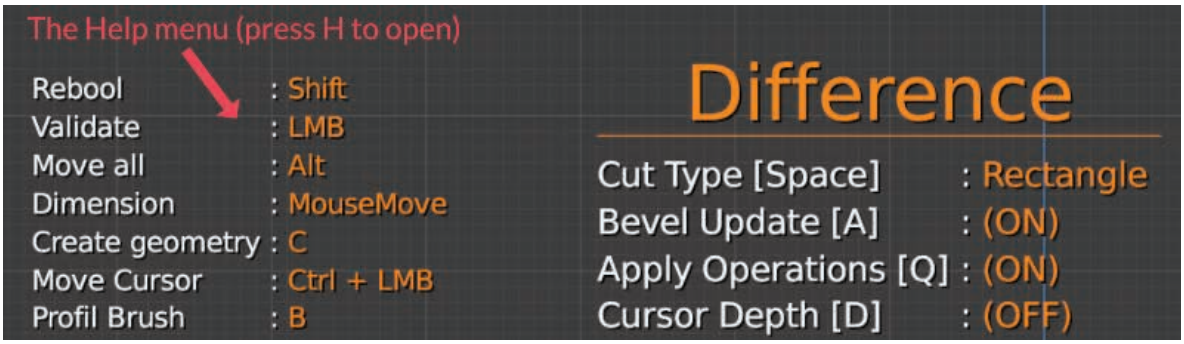




Press TAB to enter Edit Mode and scale the default cube along the Y-axis by **pressing S+Y**, then make it a bit flatter along the X-axis by **pressing S+X**. Then **press TAB** again to exit Edit Mode.

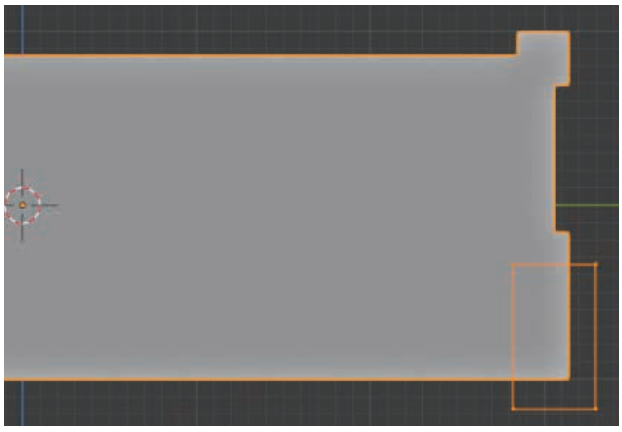


With the object selected, **press Shift+Ctrl+X** to activate Carver in cutting mode.
In the center of the screen you see which options are currently active, by default it is set to Difference with a Rectangle shape. **Pressing H** (for Help) shows the keyboard shortcuts you can use on the left.



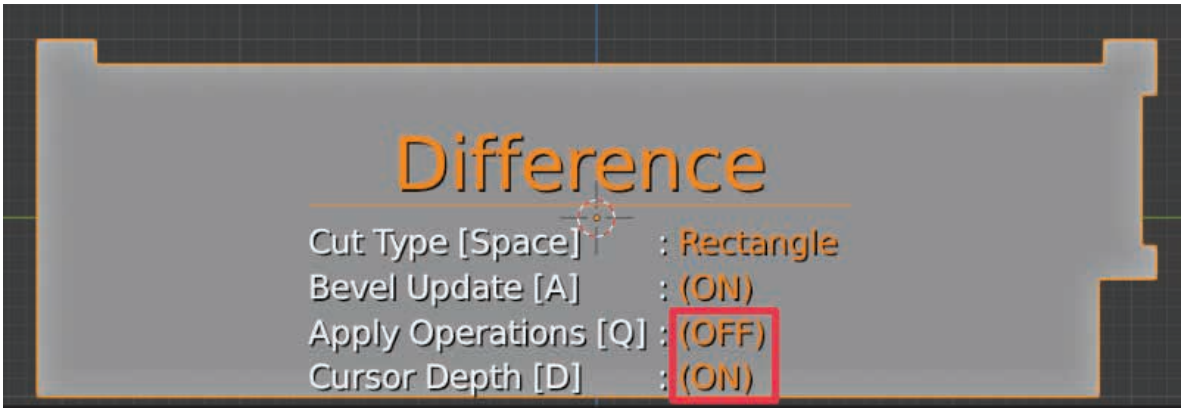
Now you can **left-click** to start drawing rectangle shapes, and **left-click again** to confirm the cut.

If you want to move the rectangle before confirming the cut, **hold Alt** before confirming the cut.

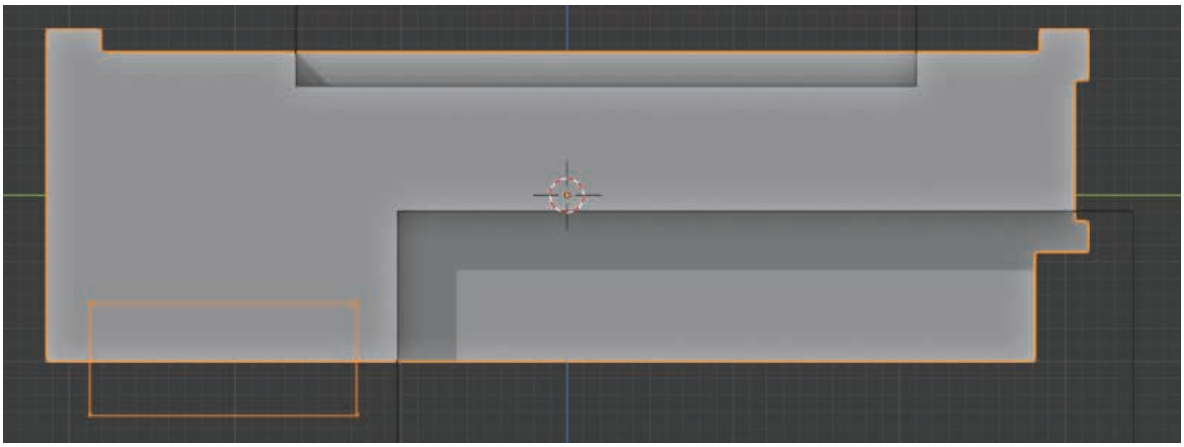


By default, this is a destructive method, meaning that once the cut is made, you can't edit the cutter object as it's automatically deleted.

By **pressing Q**, Carver goes to non-destructive mode. This means that the cutter objects remain after the boolean operation, so you can edit or move them.

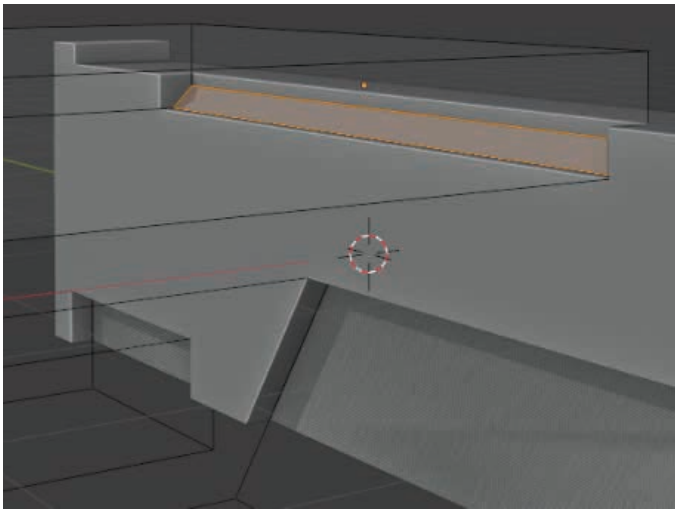


Pressing D lets you cut to the depth of the 3D cursor (which is at the world center in this case). As you can see in the below picture, the cutter shapes remain as wireframes.

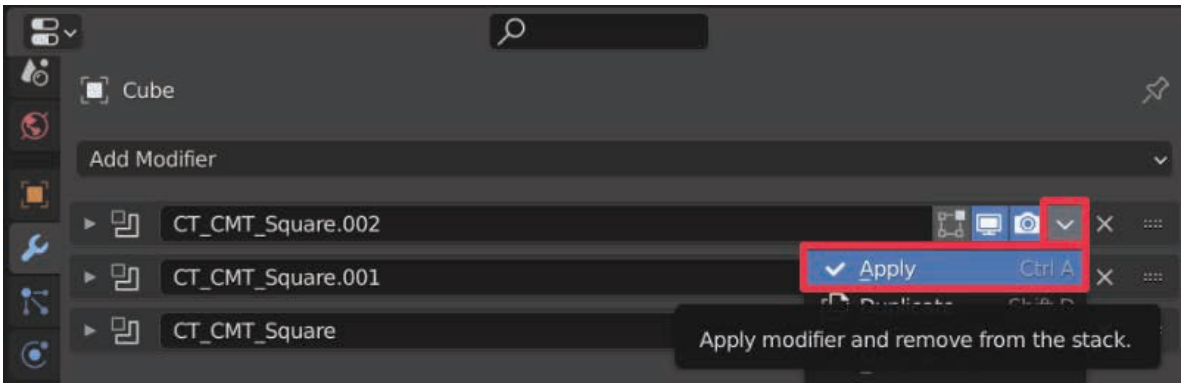


Now you can exit the Carver tool (by **clicking in an empty area of the viewport**) and select one of the cutter shapes.

This allows you to bevel edges in Edit Mode with **Ctrl+B**, and move, rotate or scale them. In this case I also moved them all slightly along the X-axis, away from the world origin.

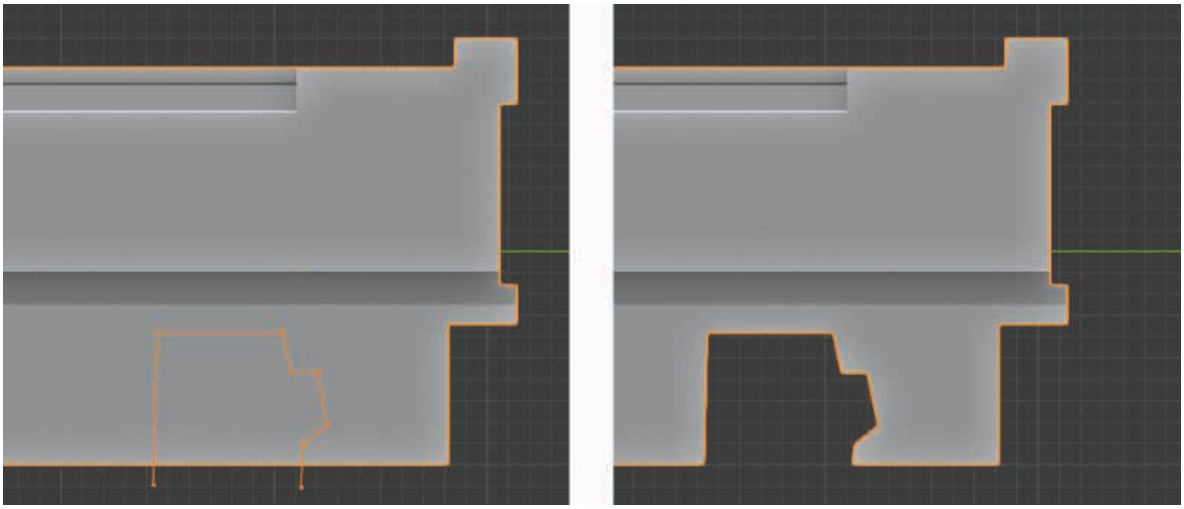


After you're done with editing the cutter shapes, you can get rid of them after you apply the modifiers on the cube. You can also **press Ctrl+A** to apply modifiers while hovering your mouse cursor above the specific modifier.

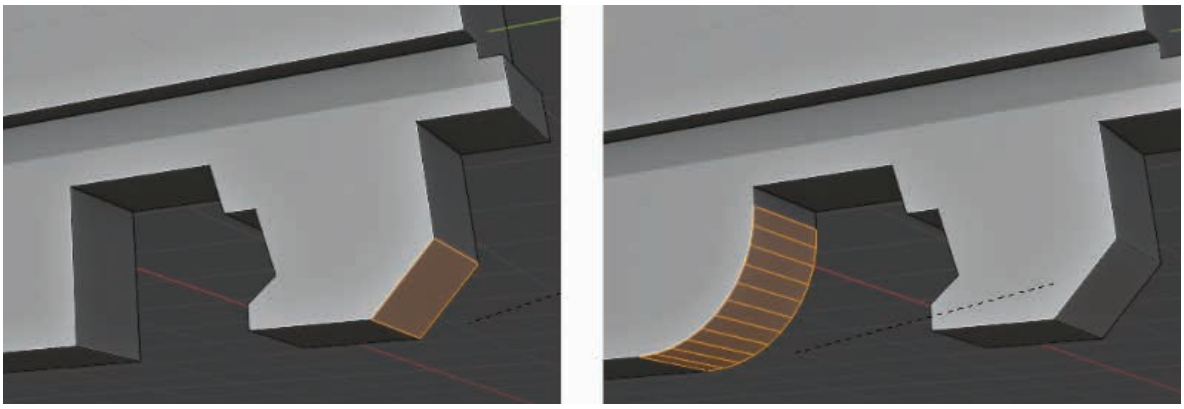


Another interesting cutting method is **Line**. You can cycle through cut types by **pressing the Spacebar**. This allows you to create custom shapes, which you then confirm by **pressing the Spacebar**. I returned Carver to destructive mode by **pressing Q** and turned off depth by **pressing D**.

Here I use the **Line** cut type to cut out the part in front of what will be the grip of the gun.

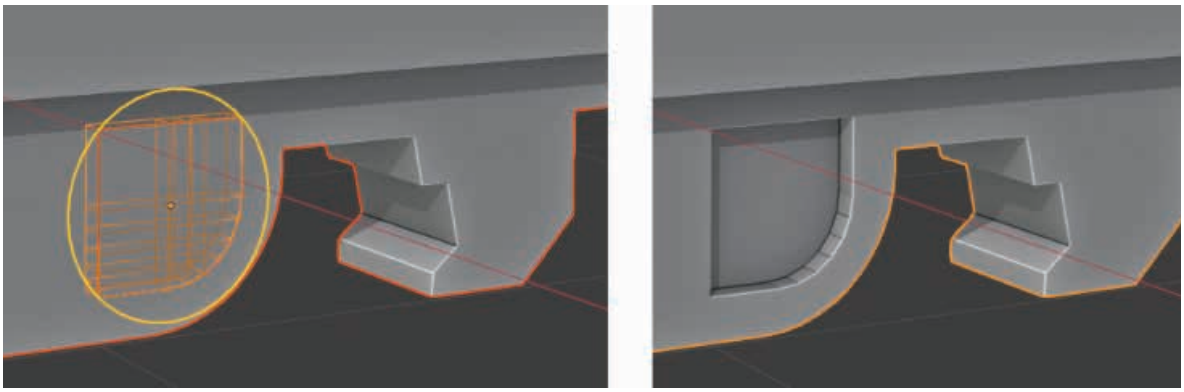


You can always go into Edit Mode and bevel some edges with **Ctrl+B**. **Scroll the mouse wheel up** to increase the amount of segments.

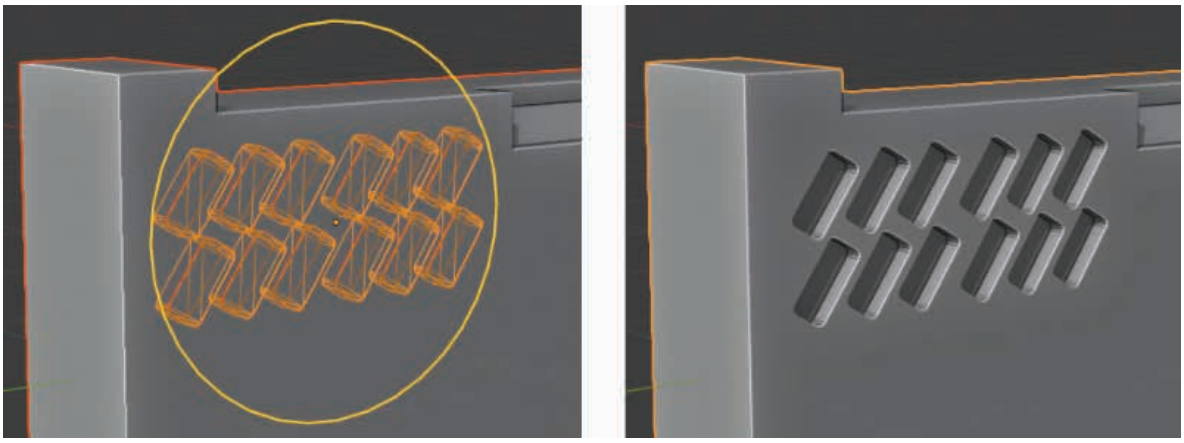


The **Profil Brush**, which is activated in Carver tool by **pressing B**, allows you to use some preset cutter shapes. There are quite a few options, so I recommend opening the Help menu by **pressing H**.

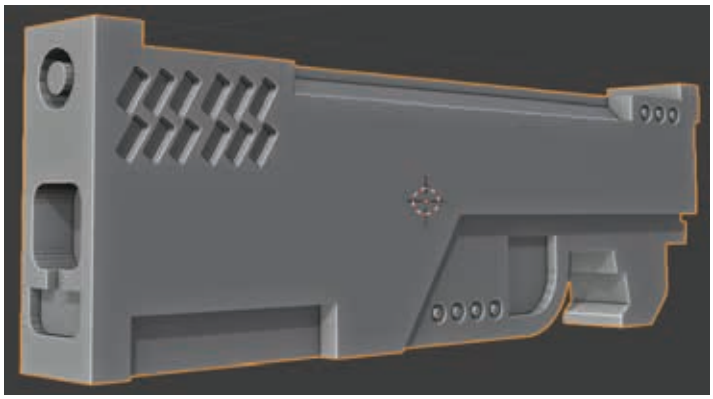
Pressing X(or W) cycles through the preset shapes. Scale the shape down by **pressing S**, and rotate by **holding down the left mouse button while moving the mouse**. You can control the thickness by **pressing D** and moving the mouse.



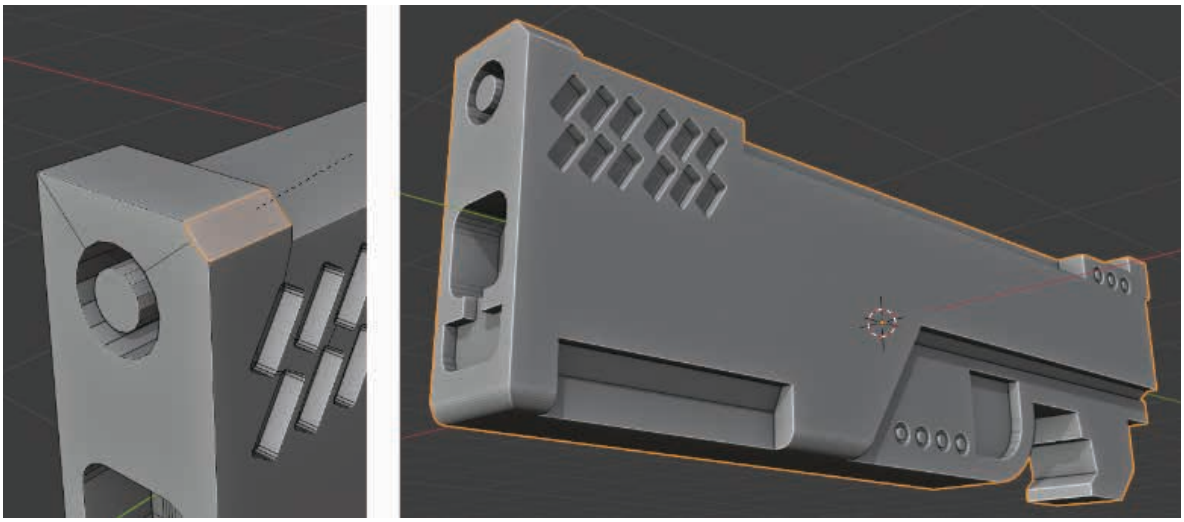
What's nice about the Profil Brush is that you don't have to use it in Orthographic view, it will align itself to the object's face normals. You can also create an array by **pressing the arrow keys** to increase the horizontal and vertical count.



As you can see below, I added some details by using some of the preset cutter shapes of the Profil Brush.



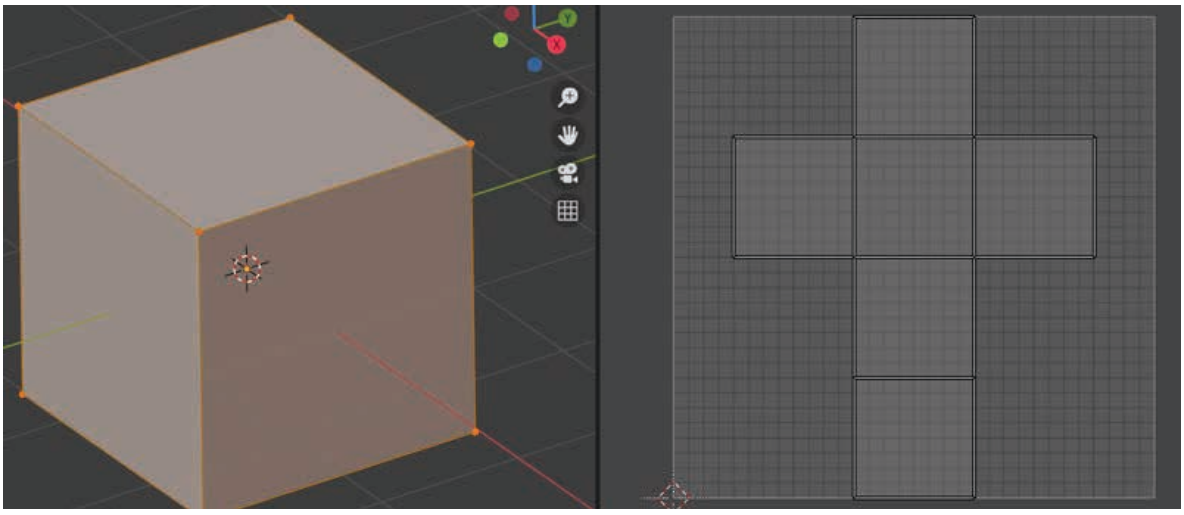
As a final modeling touch I Beveled some more edges, where the geometry allows it. You can never have enough beveled edges!



Modeling Broken Pillars

In this tutorial, we'll turn the default cube into a realistic textured broken pillar mostly by using modifiers and textures. Start of by not deleting the Default Cube! You can watch this tutorial in two videos here: Part 1, Part 2.

Open a **UV Editor** window so you can see the UV layout of the Default Cube. Like most primitive objects, the cube already has a UV layout that you can see when selecting all in Edit Mode.

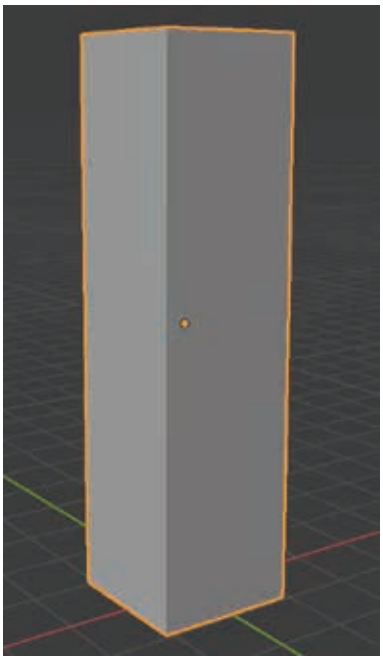


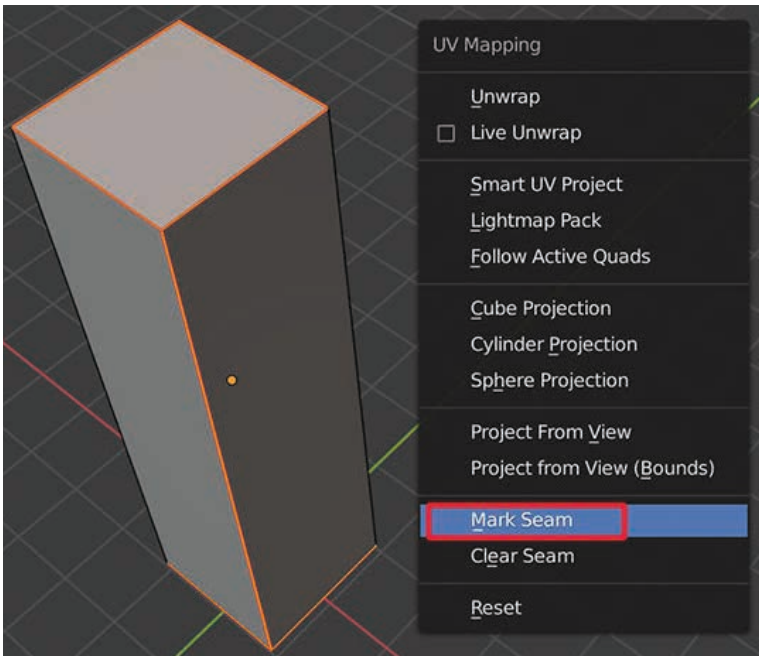
However we will create a more convenient UV layout that we can update as we change the shape of the model.

In Edit Mode, select all by **pressing A** and stretch the cube out by **pressing S+Z** and moving the mouse to make the cube more like a pillar.

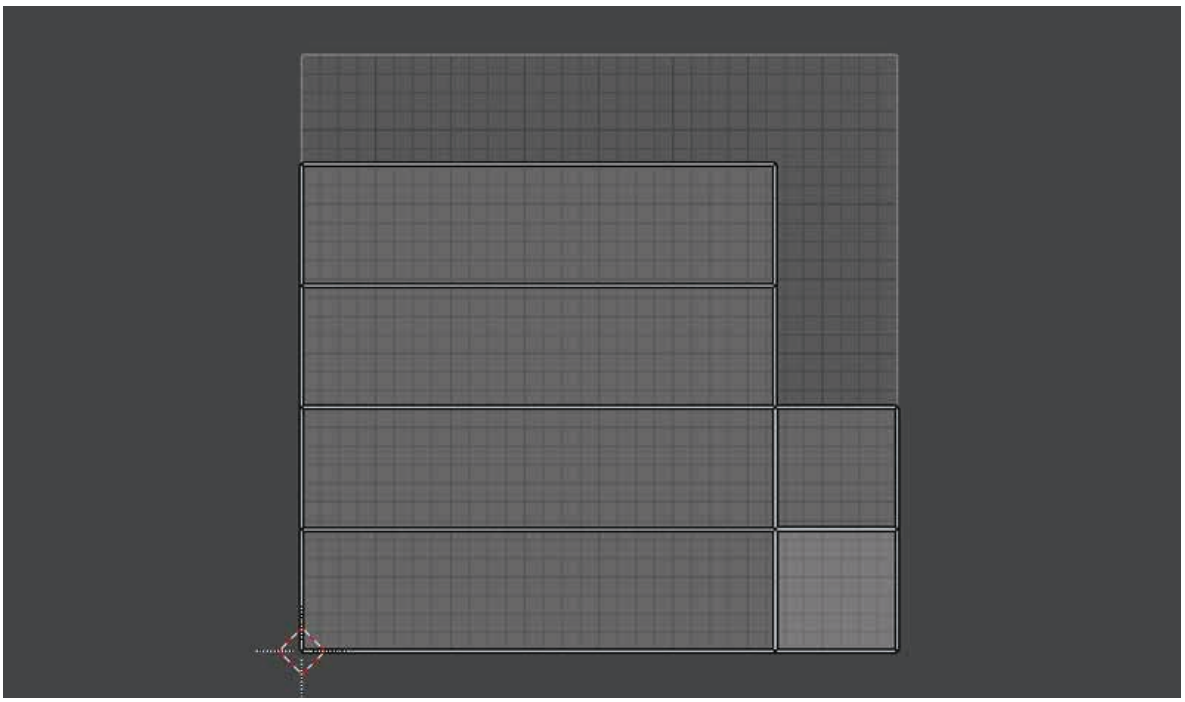
Press Alt+A to deselect everything.

While still in Edit Mode, **press 2** for **Edge Selection mode** and select only the edges of the top and bottom plane. **Press U** then choose **Mark Seams**. Now select one vertical edge of the cube and mark it as a seam as well.



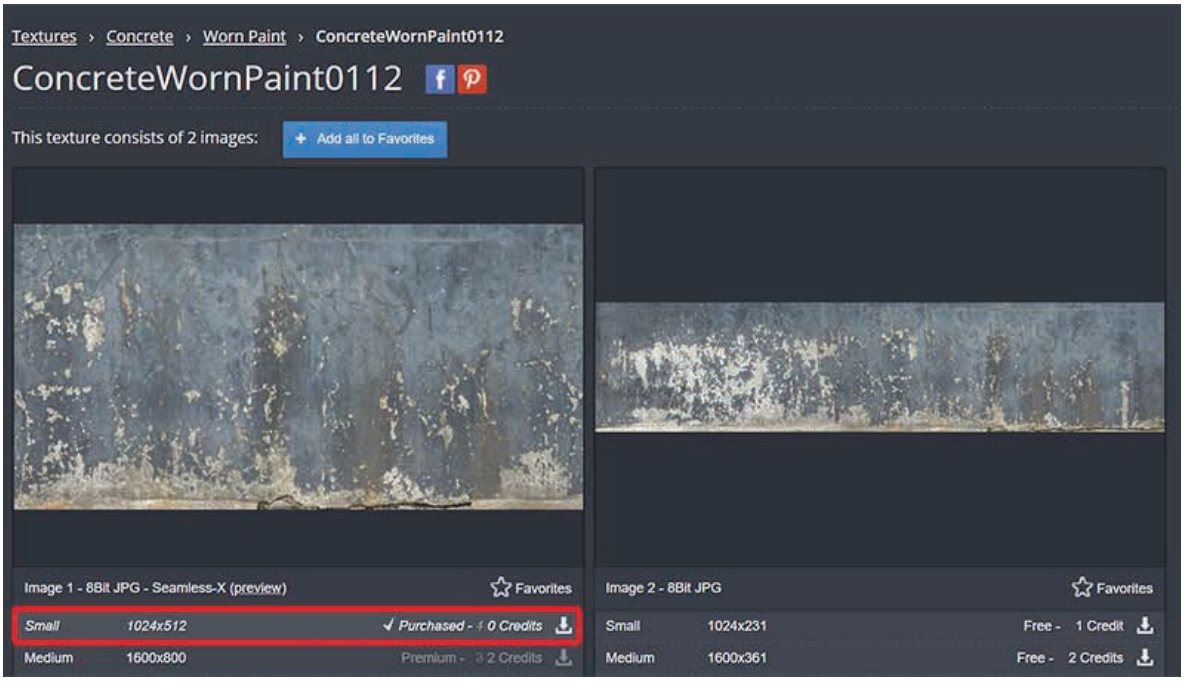


Press A to select all and **press U** then choose **Unwrap** (or press U twice). You should get something like this in the UV Editor window:



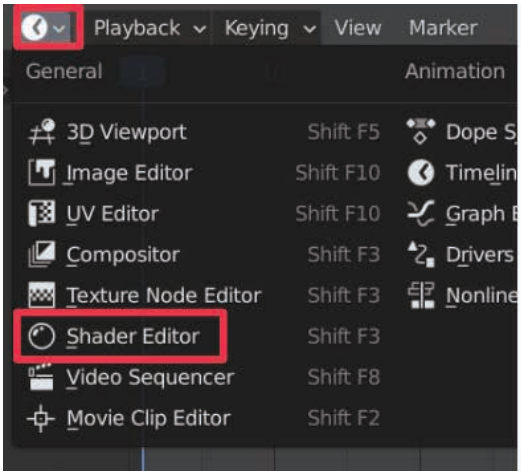
Now we are going to give it a texture. Photographic textures are a great short-cut to achieving photorealism. Go to Textures.com (or a different website of your choice) and click on the category "Concrete".

There are further subcategories like "Worn Paint" for example. I like the texture called "ConcreteWornPaint0112". You can download the small version for free.

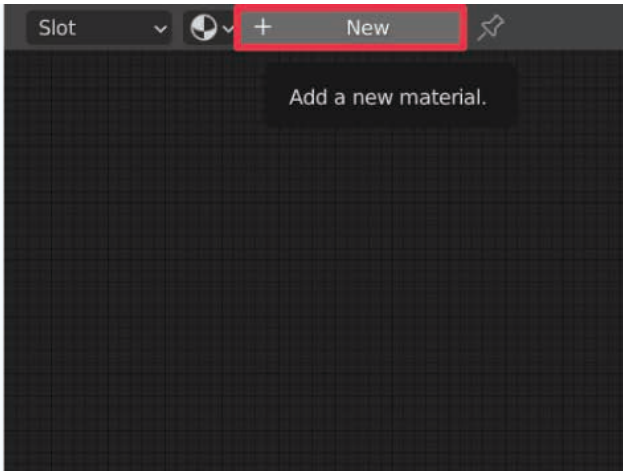


Download the texture and place it in a "Textures" **subfolder** of the folder in which you save your Blend file. That way you don't need to search for your textures later.

To add the texture to the material of the pillar, open a Shader Editor window by clicking on the Timeline icon and changing it to Shader Editor.

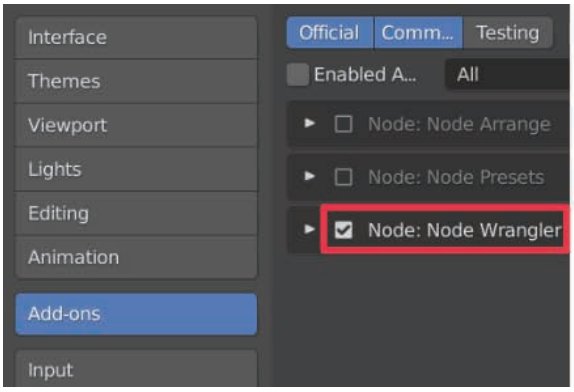


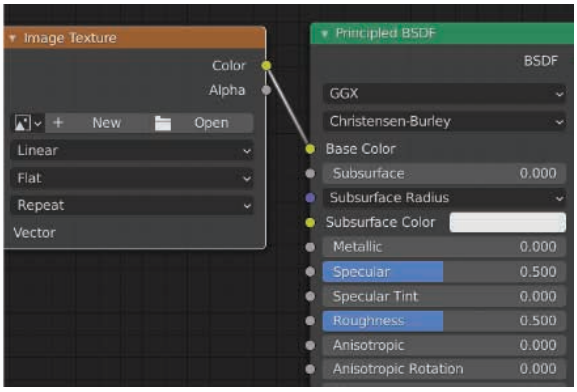
Drag the window border up a bit to make it bigger. **Click on New** to create a new Material.



Now you can zoom in on the newly created nodes by scrolling the mouse wheel up. Make sure **Node Wrangler** is enabled in Preferences.

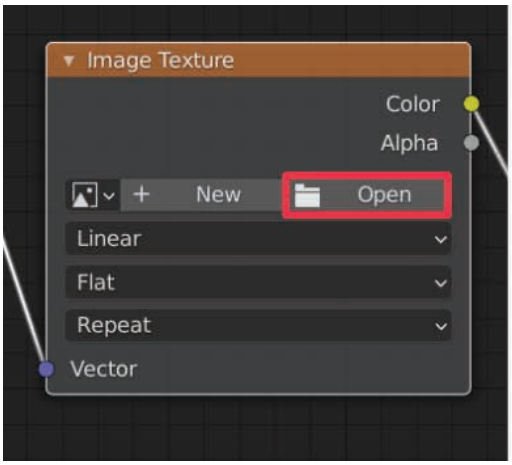
Then click on the Principled BSDF node to select it and **press Ctrl+T** to add image texture nodes. You may have to reposition these nodes by **pressing G** and moving the mouse.

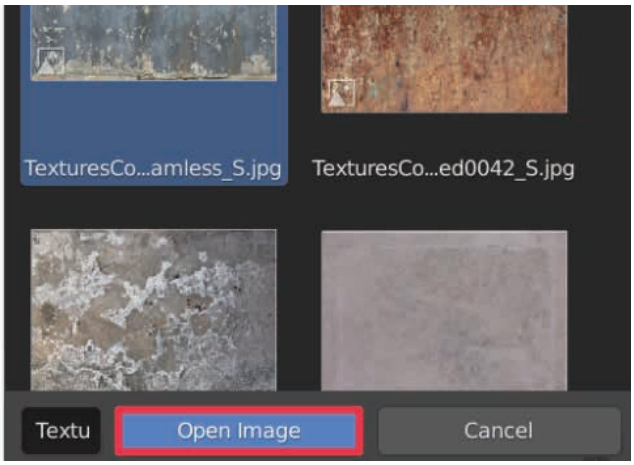




Click on **Open** on the Image Texture node.

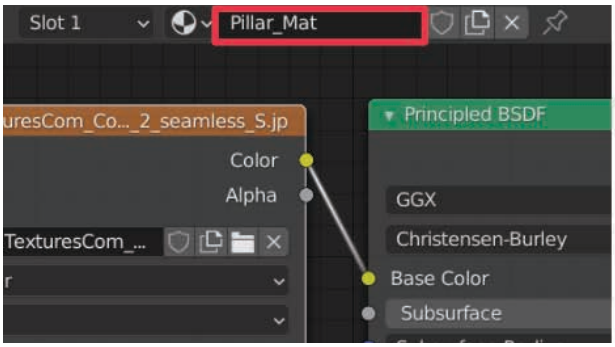
Browse to the texture, select it and click on **Open Image**.

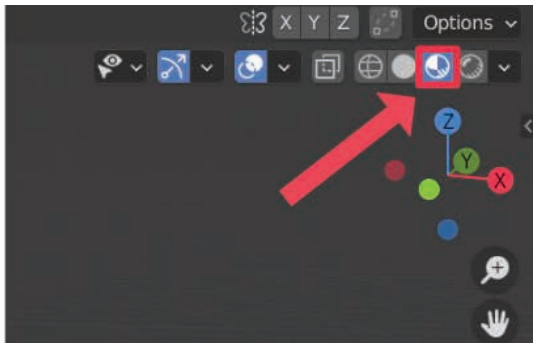




Name the material "**Pillar_Mat**" or something else that helps you recognise it as the Pillar material.

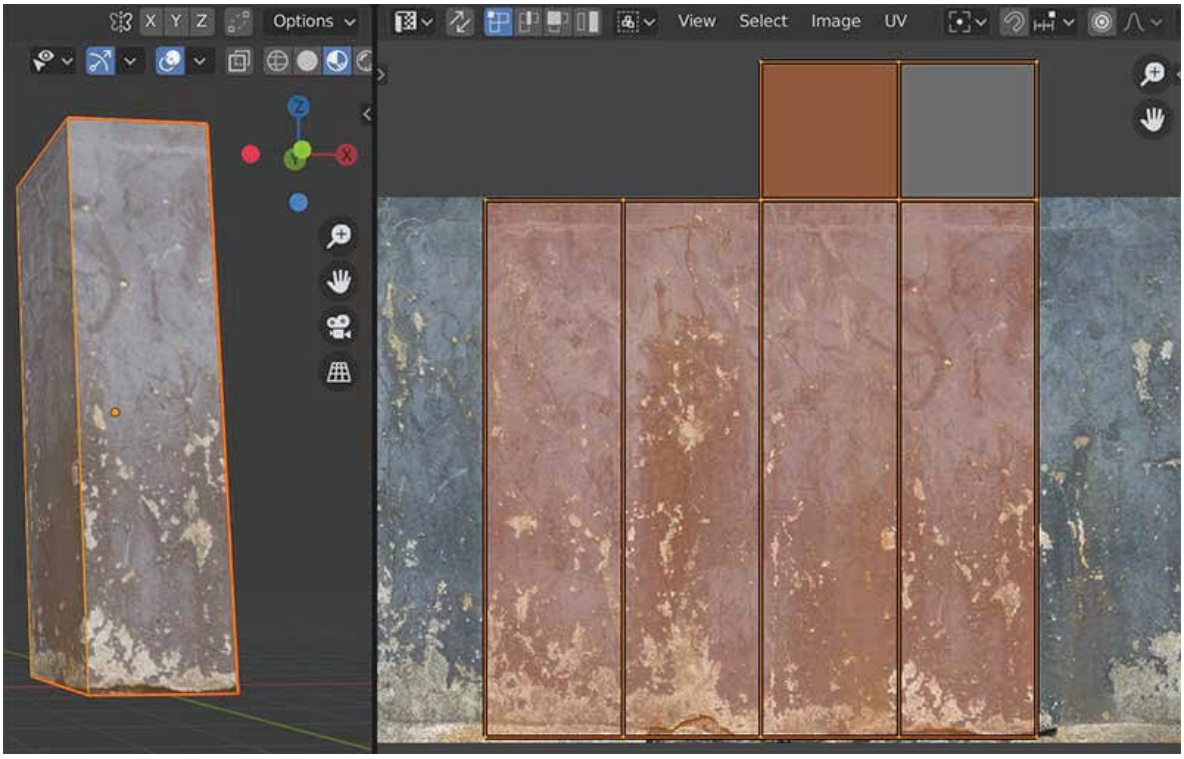
You will need to switch to Material view to see the texture in the viewport.



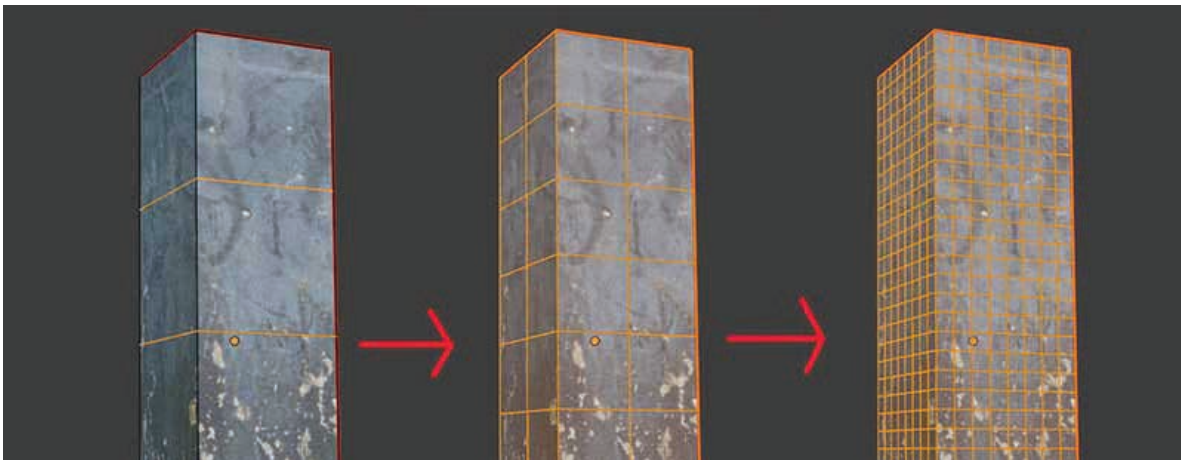


We may have to rotate and scale the UV's to fit the part of the texture that we want to use. In some cases like with this rectangular texture, you need to unwrap the object again so the aspect ratio is correct.

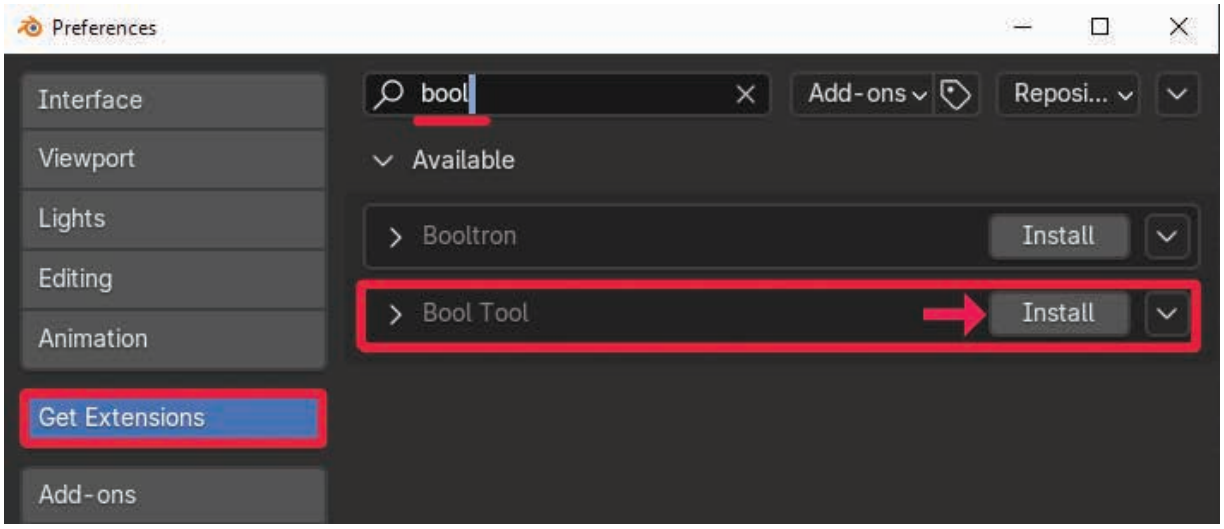
You can select the vertices in the UV layout window and rotate them incrementally by **pressing R and holding Ctrl**. Or you can **press R and 90** for example, to rotate them 90 degrees. **Press S** to scale the UV's. Transform the UV's until you get this kind of effect where the most damage to the pillar seems to be at the bottom.



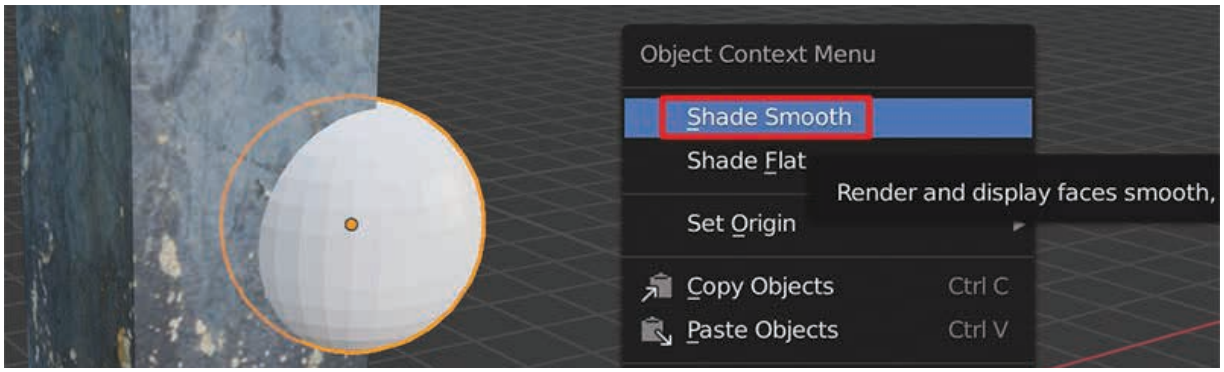
In **Edit Mode**, add some more horizontal Edge Loops with **Ctrl+R** and then **Subdivide** the pillar a couple of times to add some more geometry. This comes in handy when using booleans later.



Make sure the **Bool Tool** is enabled in Preferences.

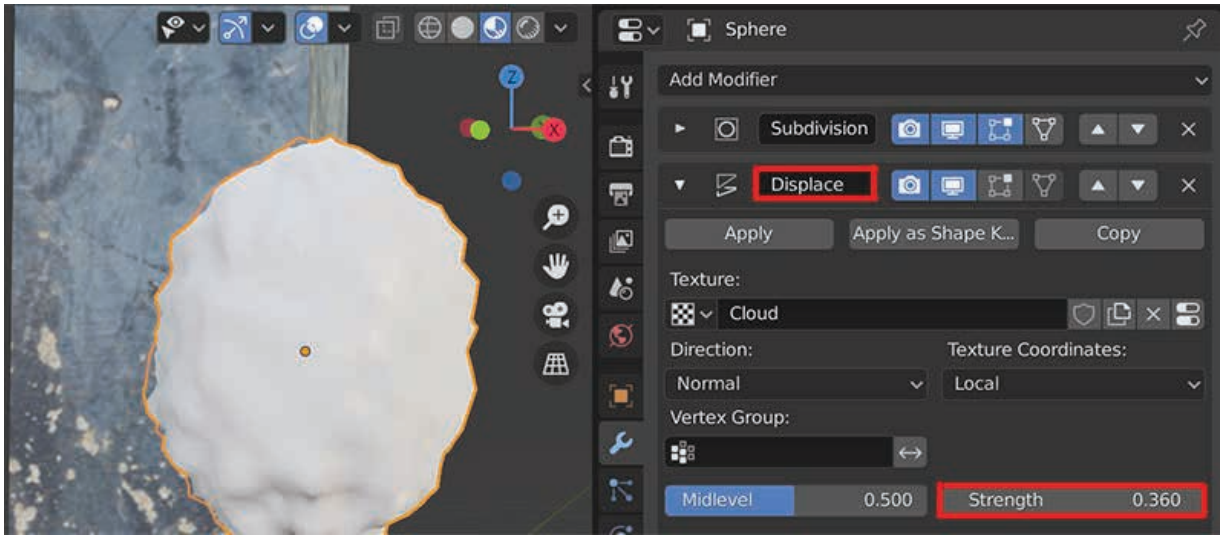


Create a **UV Sphere** from the Add menu and set it to **Shade Smooth**.



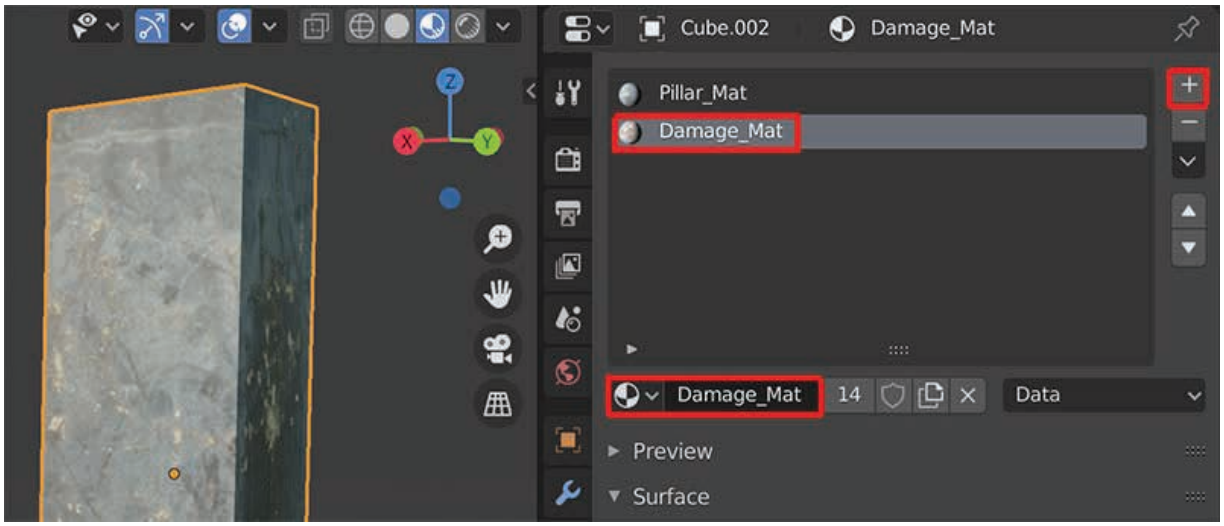
Give the Sphere a **Displacement modifier** with a procedural **Cloud texture**. Experiment with the Strength value. Add a **Subdivision modifier**before the Displacement modifier to get more detail.

This is the object which will take a "bite" out of our pillar. If you want to scale the Sphere along the Z-axis for example, it's best to do so in Edit Mode.



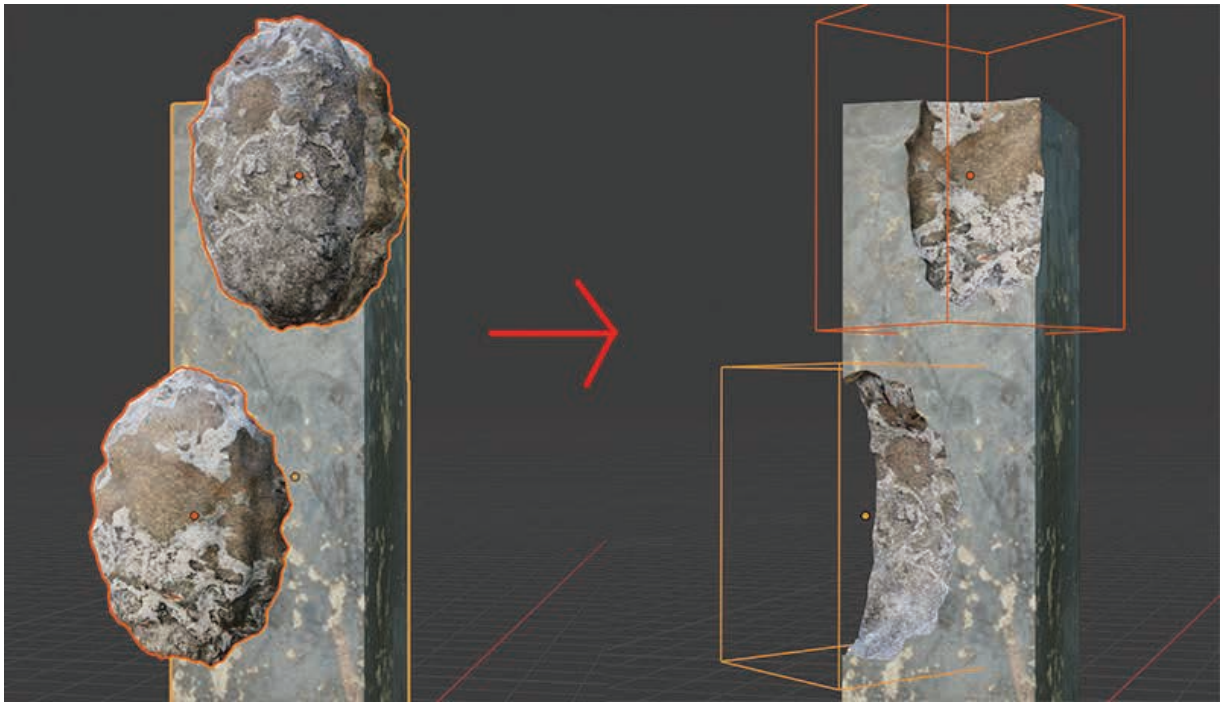
Download another texture, this time a rough concrete that will represent the damages parts of the pillar. Add a material to the sphere and use the texture as the Base Color. I used the texture "ConcreteFloorsDamaged0025" from Textures.com.

Name this material "Damaged_Mat" for ease of use. Add this material to the pillar as well (as a second material) by clicking on the + icon and selecting it. This is necessary for the pillars to show an "inner" material after the Boolean operations.



Move the UV Sphere so it overlaps somewhat with the pillar and with the Sphere selected, hold Shift and select the pillar as well. Then **press Ctrl + minus on the keypad**. The sphere will turn into a Bounding Box and the pillar will have a chunk bitten out of it.

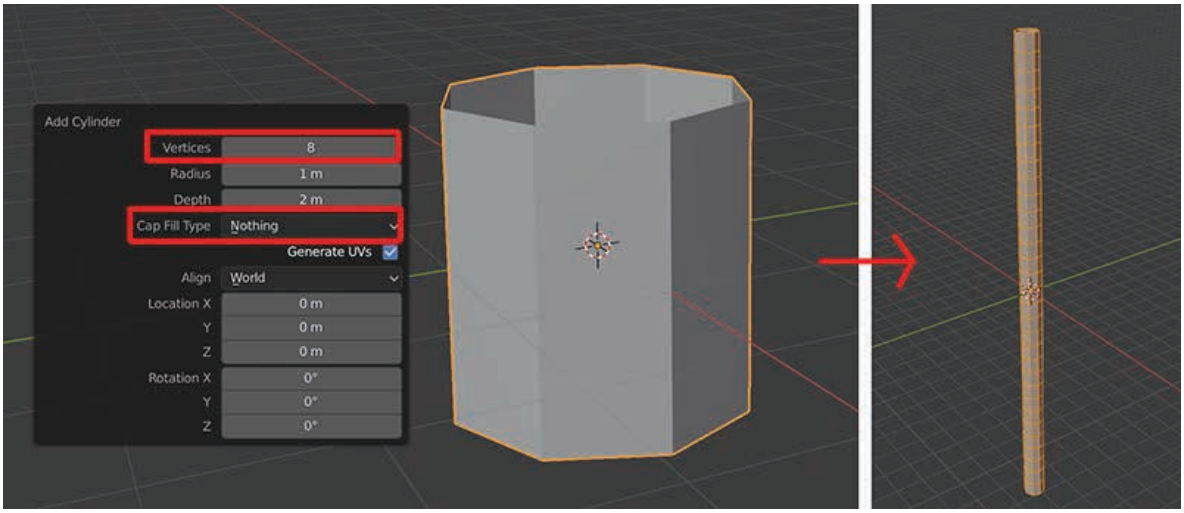
If you want to take several “bites” out of the pillar you can duplicate the Sphere first, by selecting it and **pressing Shift+D**, and moving it to where you want.



If it happens that you see the seam of the texture in the cut out parts, simply rotate the bounding box along the Z-axis until the seam is gone.

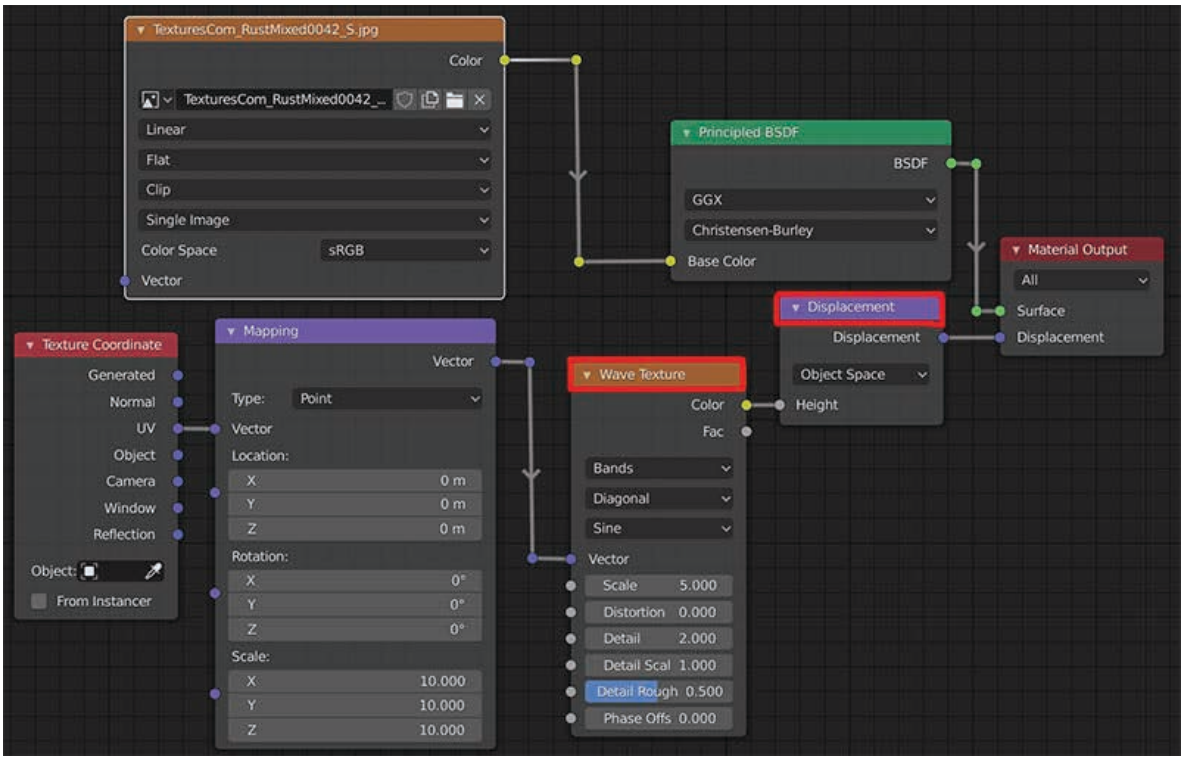
Now let's add some **rusted metal rebar**. These are the metal cylinders that reinforce concrete pillars. Add a **Cylinder** and set **Cap Fill Type** to **Nothing**. Vertices can be set to 8.

Scale it along the Z-axis to almost the length of the pillar. **Press Ctrl+R** and scroll the mouse wheel up to add some edge loops, which are necessary for the next step. In Object Mode set it to **Shade Smooth**.

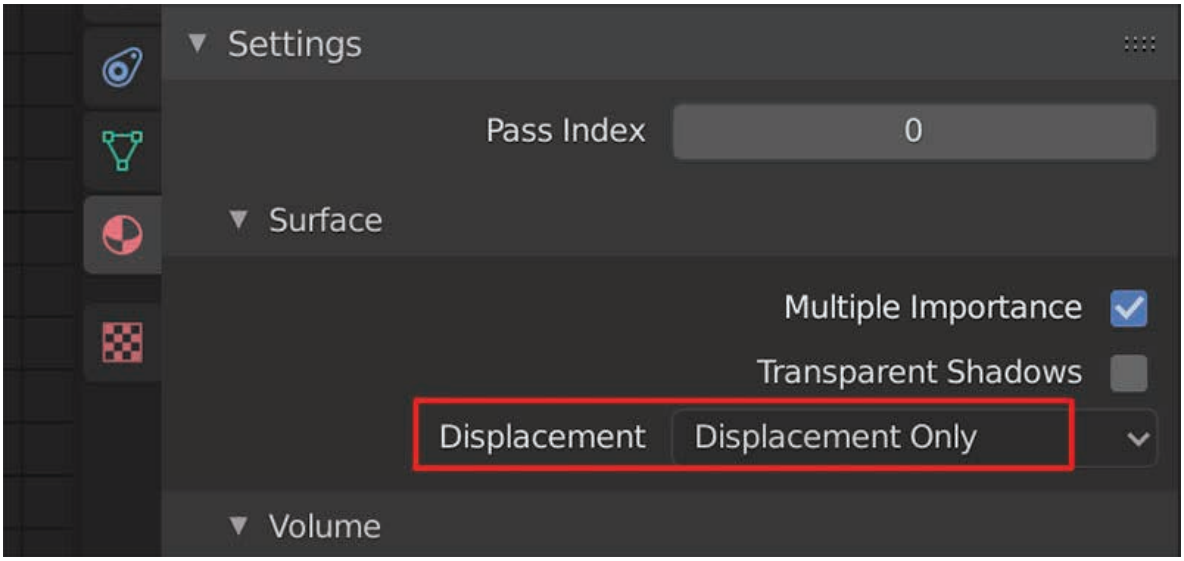


To create the material for the rebar, download a rust texture. I used "**RustMixed0042**" from Textures.com.

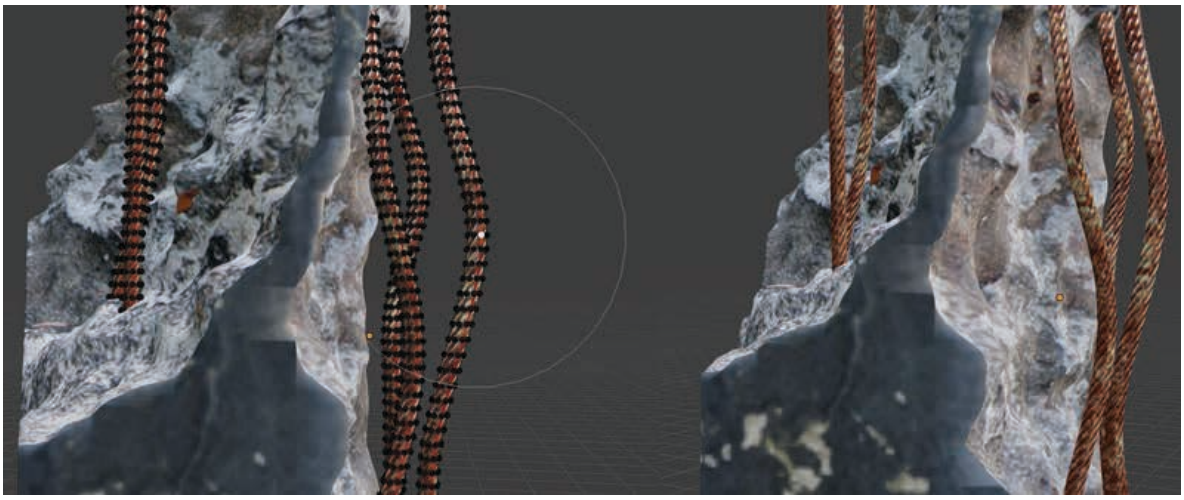
Rebar has this specific kind of look where it has a kind of thread running along its length. We can add some nodes to the material to create this. Mainly we'll use a **Wave Texture** to create a black-and-white displacement map. Using a **Displacement node**, we can use the Wave texture to create actual displacement on the model.



In the Material **Settings**, turn on **Displacement Only** as the type of Displacement (please note: this option is only visible with Cycles set as the render engine).

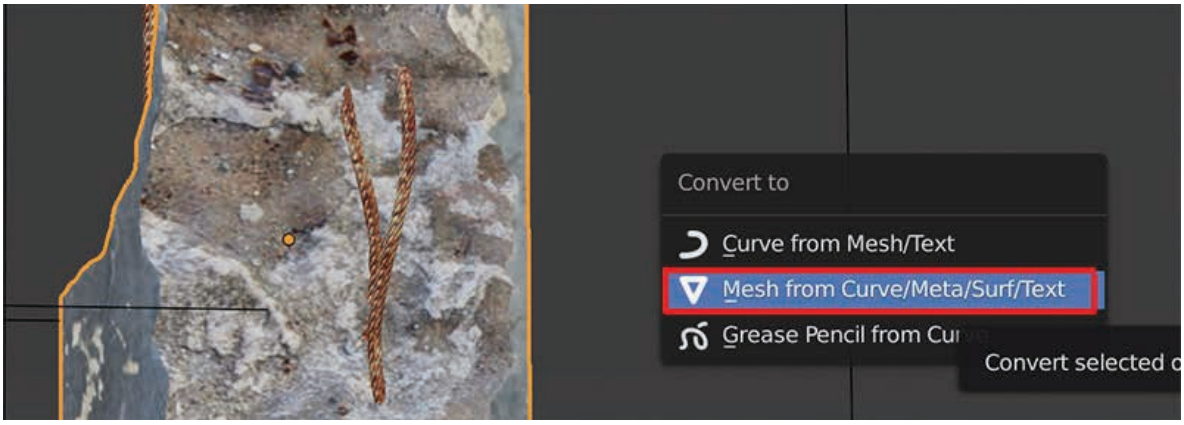


In Edit Mode, select all and duplicate the rebar a few times with **Shift+D** and move the duplicates with **Shift+Z** (this means you can move it along all axis except the Z-axis). Turn on **Proportional Editing** and check "**Connected Only**". This checkbox allows us to edit the individual cylinders. Move them to make them look more bent and destroyed.



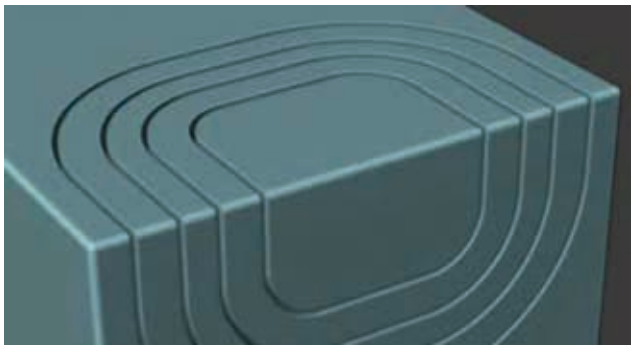
This is the basic workflow for creating the damaged concrete pillars with rusted rebar sticking out of them. To create some variations, you can simply duplicate them including the boolean cutter objects, and move those around to remove chunks of the pillar at different locations.

To make these more low poly you can apply the modifiers and then add a **Decimate modifier** to it. The fastest way to apply all modifiers is to select the pillar, **press F3** and search for "**convert to**", then choose "**Mesh**". After that all the modifiers have been collapsed and you can add the Decimate modifier to make the model smaller.



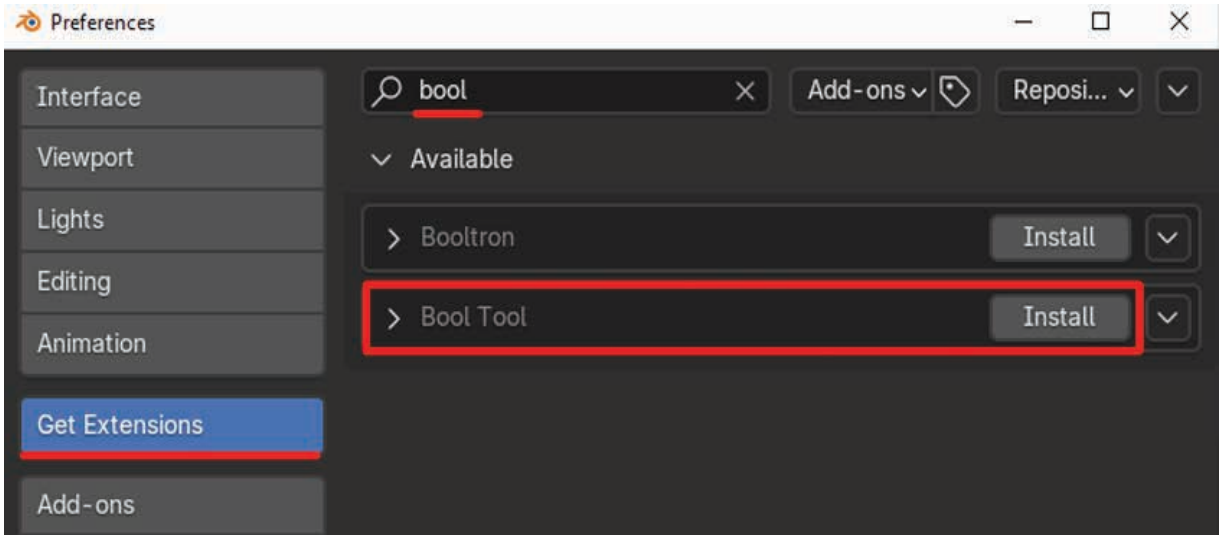
Slice Cuts

Commercial hard-surface add-ons are very popular right now.

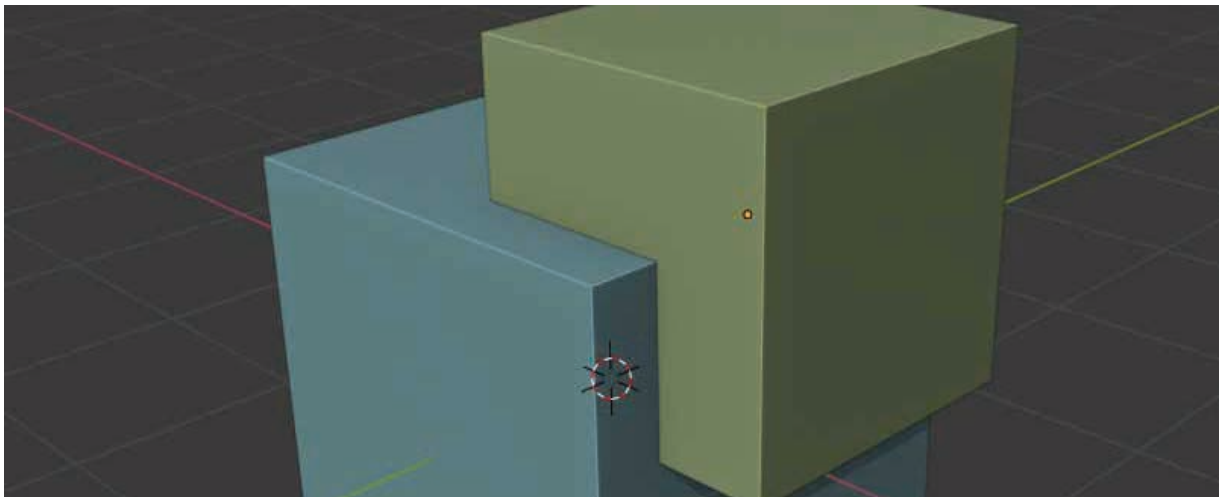


They are all based on modifiers that we can find in Blender by default, so let's have a look at creating nice slice cuts like shown here, with the default tools of Blender.

Before we begin: This tip assumes you have the **Bool Tool add-on** enabled in Preferences, which comes with Blender by default. Although you don't strictly need it, Bool Tool speeds up boolean cutting. Search for "bool" in Preferences > **Get Extensions** and click on **Install**.

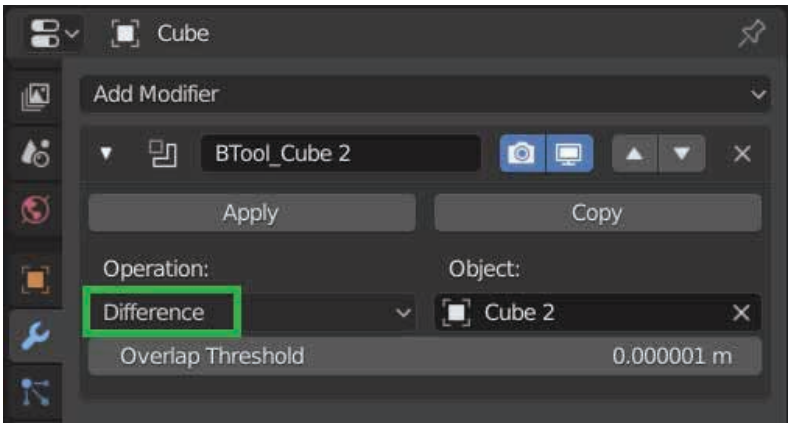


To make slice cuts in the default cube, create a second object that acts as a cutter object. Scale and move it (**S to scale and G to move**) so that it intersects the first cube where you want to slice.

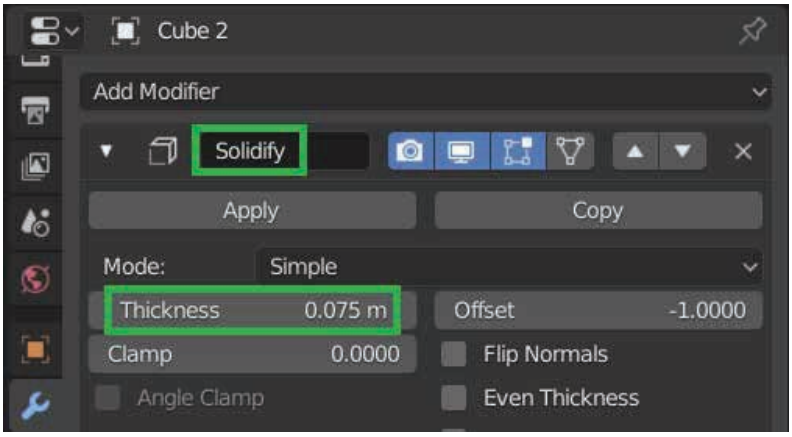


Select the object you want to cut with, hold shift and select the object you want to cut into. Then **press Ctrl + [- on the numpad]** to make a "difference" cut using the cutter object to cut into the cube (this shortcut only works if Bool Tool is enabled).

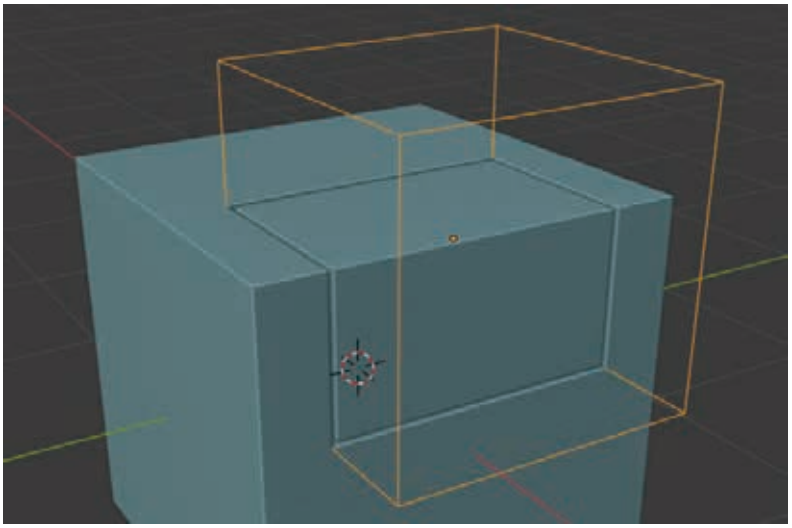
You'll notice a Boolean modifier has been added to the cube, set to **Difference**.



Now, select the cutter object (which now looks like a bounding box) in the viewport and **add a Solidify modifier** to it.



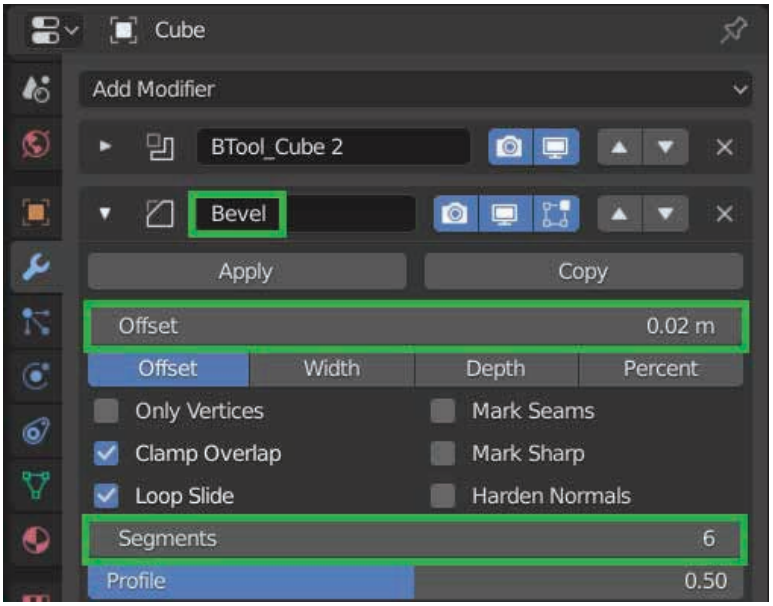
Change the **Thickness** Value to change the thickness of the slice.



Then you can add a **Bevel modifier** to the cutter object in order to give the slice rounded corners.

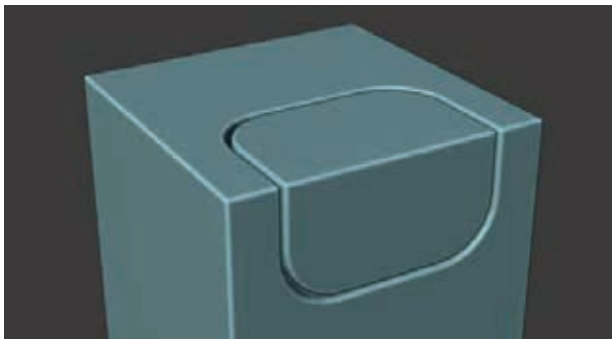
Play with the **Offset** value until you like the result.

You can add more **Segments** to make the bevel rounder.



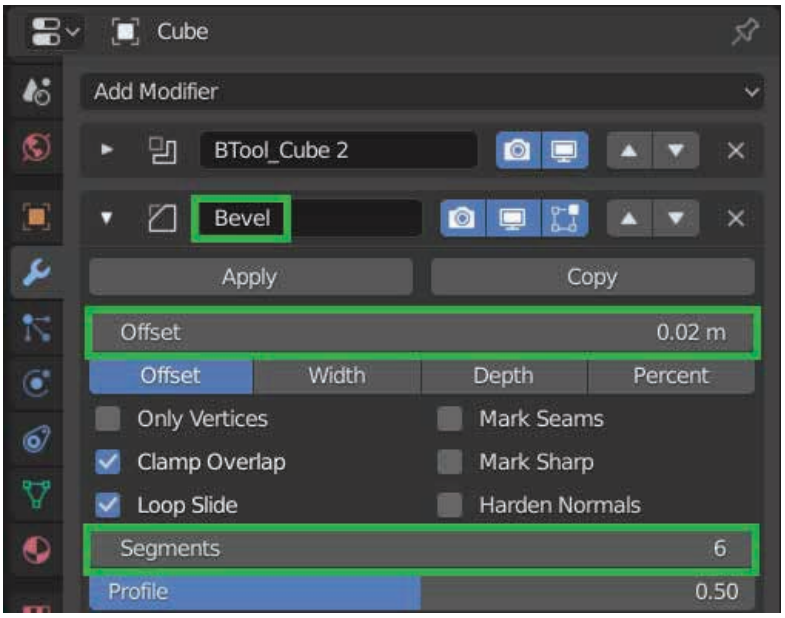
Now it looks like this. You'll notice that the round corners are not smooth. To achieve a nice smooth look, with the cutter object selected right-click and choose **Shade Smooth**.

Then turn on **Auto Smooth** as well. Now the corners are smooth.

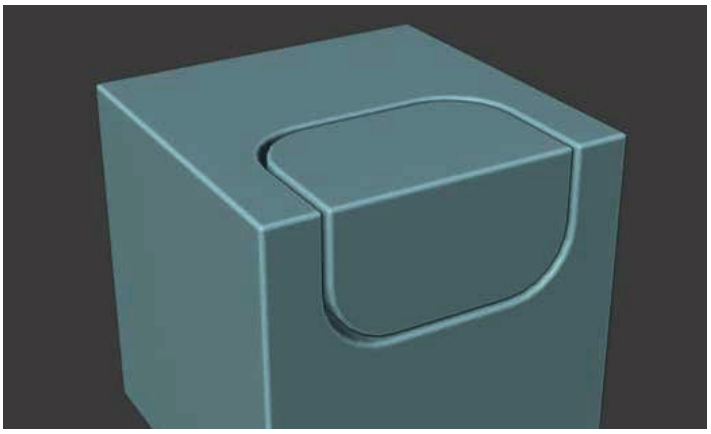


You can also add a **Bevel modifier** to the first cube.

Note that you will get different results from placing the **Bevel modifier** above or below the **Boolean modifier**.



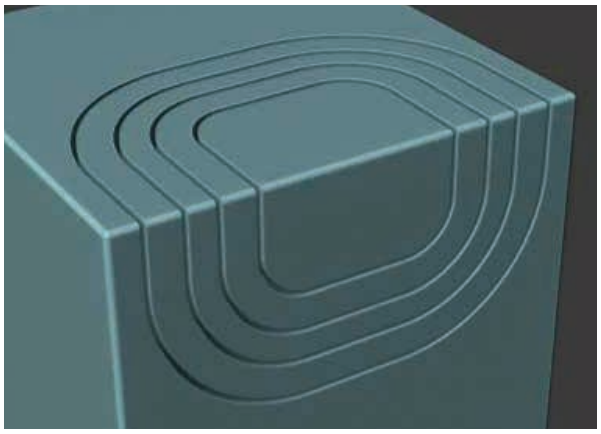
Neither is right or wrong, it just depends on what look you want to achieve.



To create more slices you can add more **Solidify modifiers** to the cutter object.

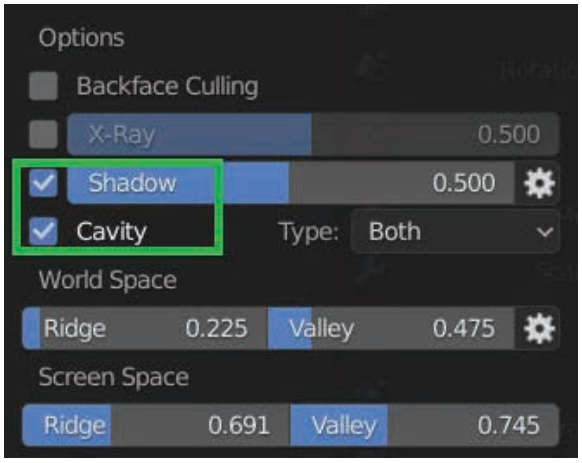
Click here to return to the index.

It takes some experimenting with the **Thickness** values of the different Solidify modifiers until you get it right, but after some tweaking you can end up with something like this:



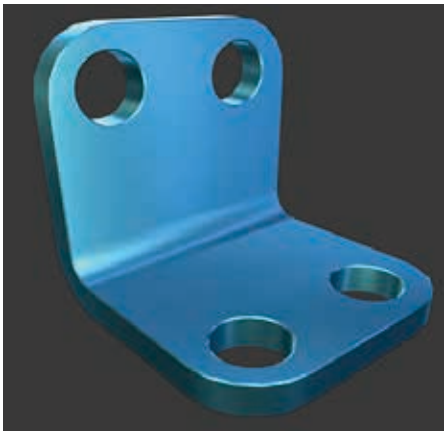
With this many Solidify modifiers, the shading may look better with **Shade Smooth** and **Auto Smooth** turned off.

In case you're curious about how I created the beveled look in the viewport before the objects had Bevel modifiers-this is done by turning on the **Cavity** option in **Viewport Shading** options. I also turned on **Shadow** because I think it looks nice.



Boolean Basics: Non-Destructive Workflow

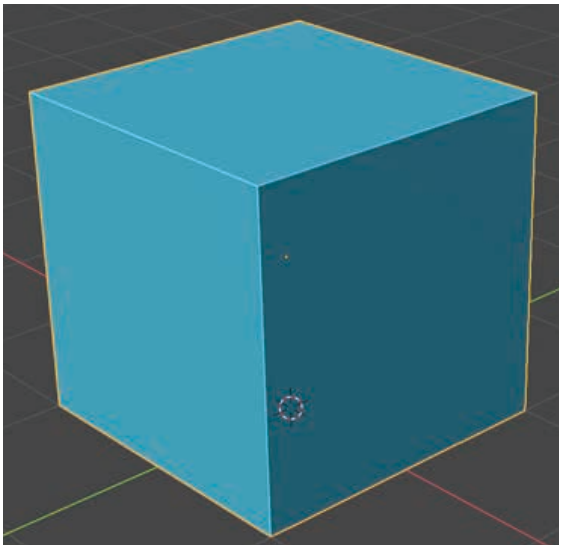
Hard-surface objects like this are not hard to model, if you use **booleans**.



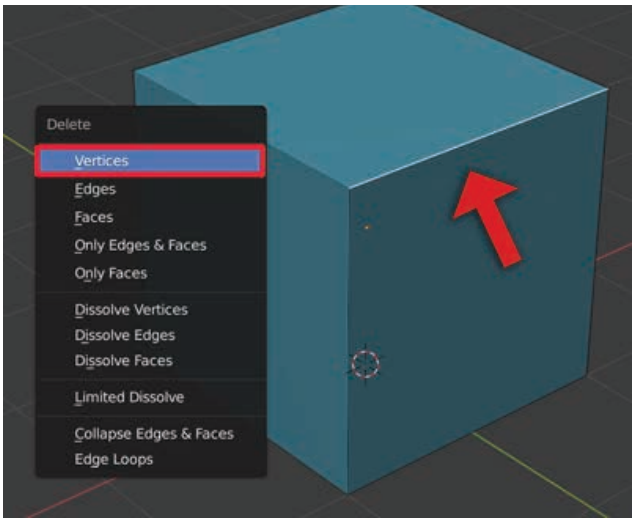
You can even keep the scene in such a way that changes are not destructive but can be altered later.

Please note that there is also an add-on that speeds up this process, to learn more check the next chapter, "**How to use the Bool Tool add-on**".

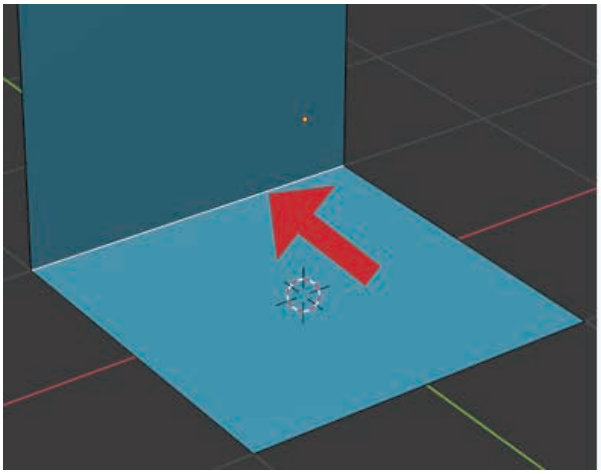
First, let's move the default cube up one unit by **pressing G+Z+1**.



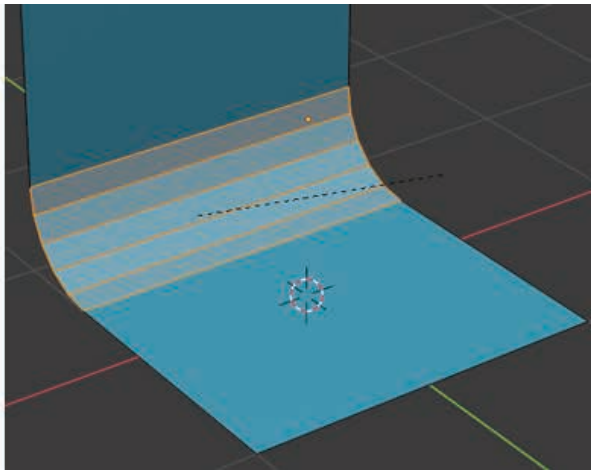
Press Tab to enter Edit Mode and **press 2** for Edge Selection mode. Select this edge, **press X** then choose **Vertices**.



That gets rid of a lot of geometry. Select the rear corner edge and **press Ctrl+B,** then move the mouse to bevel the edge.

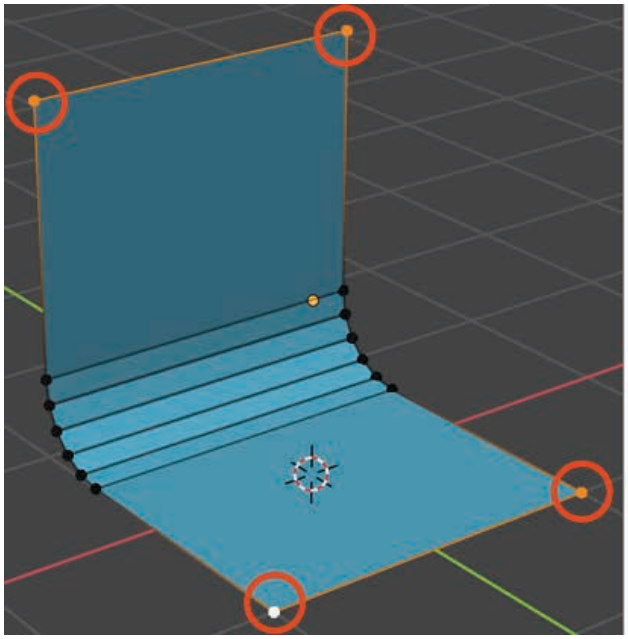


Then, **scroll the mouse wheel up** to increase the amount of segments. **Left-click to confirm.**

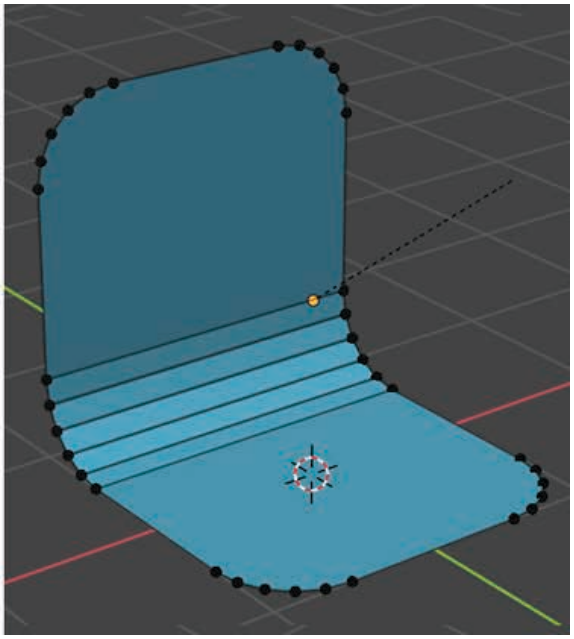


Alternatively, you can use the Bevel menu that pops up in the lower left corner of the viewport after using the Bevel function. But I recommend learning this shortcut.

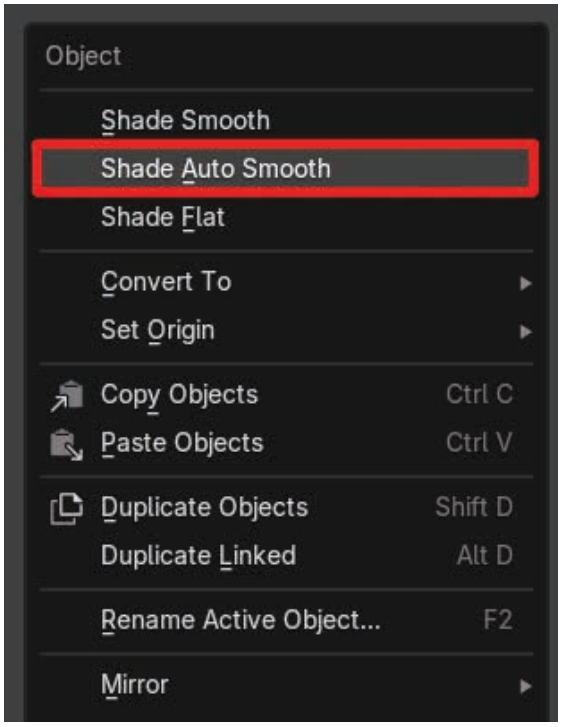
Press 1 to switch to **Vertex** select mode and select only these corner vertices while **holding Shift**.



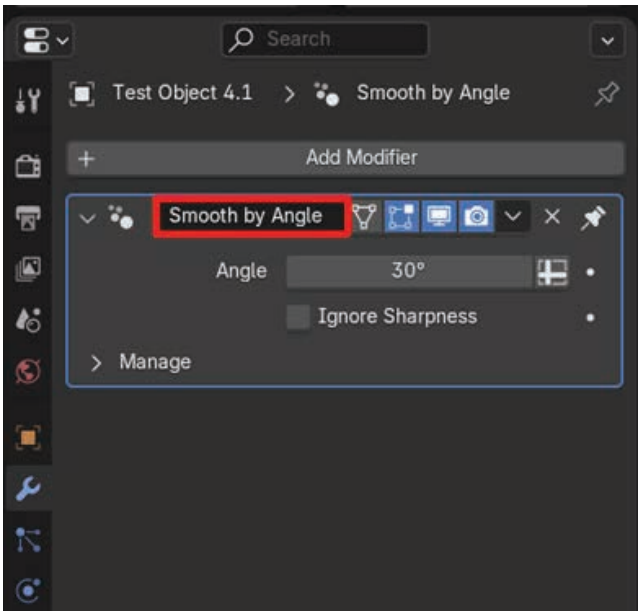
Press Shift+Ctrl+B and move the mouse laterally to bevel these vertices. You can control the amount of vertices with the mouse scroll wheel. Left-click to confirm.



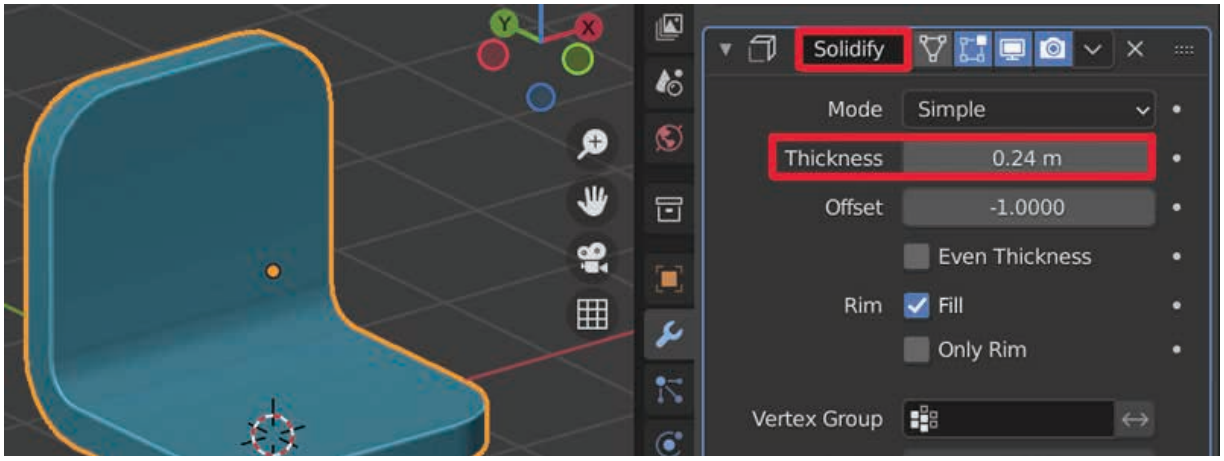
Press TAB to get out of Edit Mode. Right-click on the object and choose **Shade Auto Smooth**.



This will make sure that corners over 30 degrees are sharp, and under 30 degrees are smooth. It also adds a Geometry Nodes modifier where you can adjust this Angle value as needed.



Add a **Solidify Modifier** and adjust the **Thickness** value to get the desired thickness.

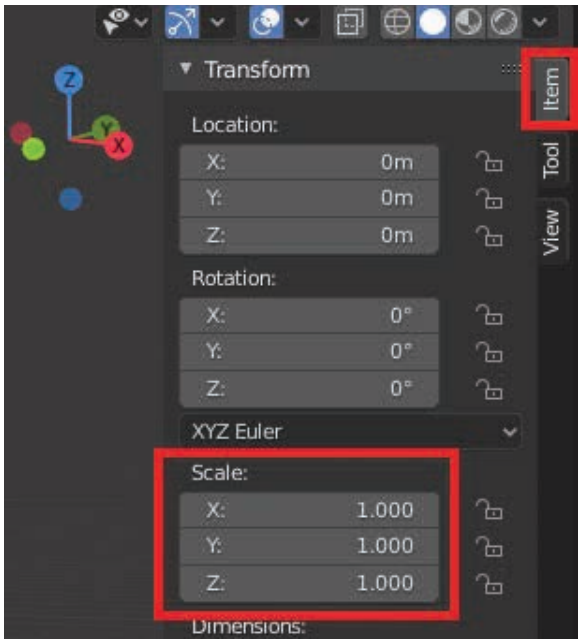


In case the thickness looks very different than what you'd expect, check the Scale of the object. **It should be 1** for each axis.

If not, apply the scale by **pressing Ctrl+A > Scale**.

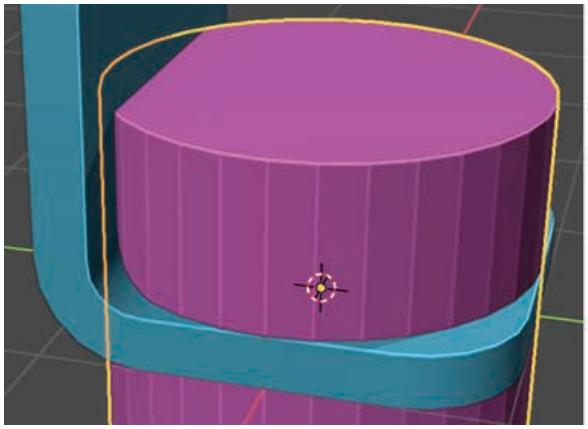
When modifiers or parent-child relationships in Blender act weird, you should make it a habit to check the Scale. It can cause a lot of issues.

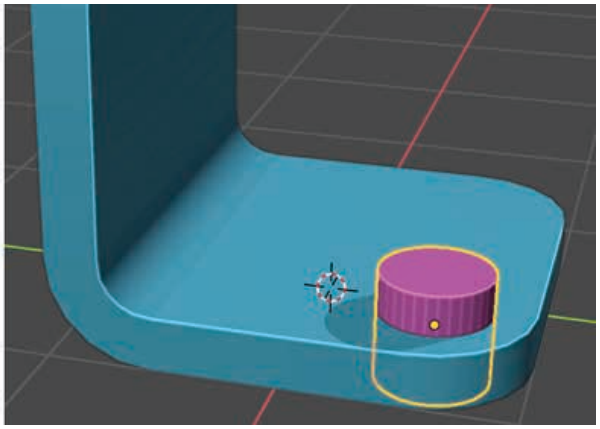
Open this panel (the Option panel) by **pressing N** if needed.



In Object Mode **Press Shift+A** and add a cylinder. Scale it down by **pressing S** and moving the mouse.

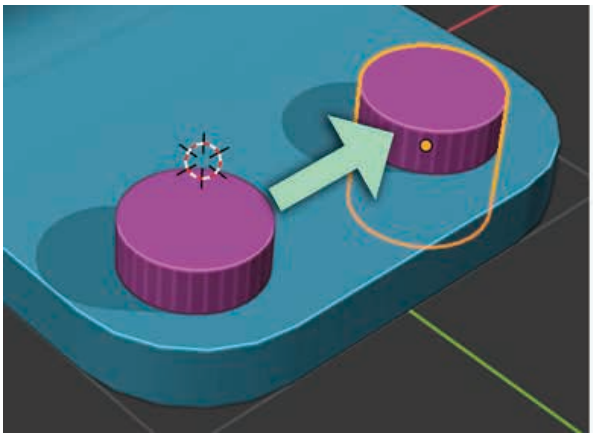
Then move it along the X and Y axis by **pressing G+X** moving the mouse, and same for **G+Y**. As with the previous object, turn on **Shade Auto Smooth**.



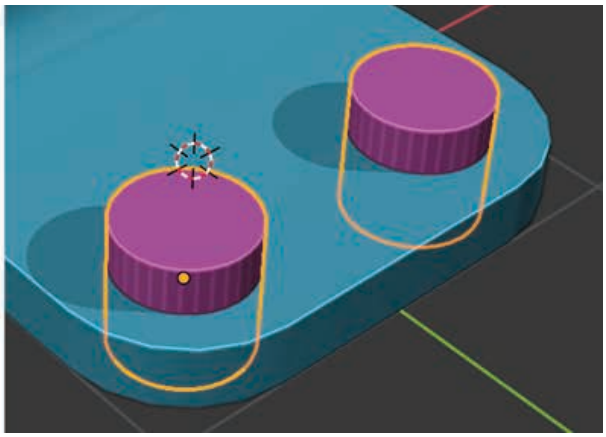


Note that you can also add objects in Edit Mode, but they will become part of the same object you are editing. In this case we want to have separate objects so we can use one to cut through another. So it's essential that you add the Cylinder in **Object Mode**.

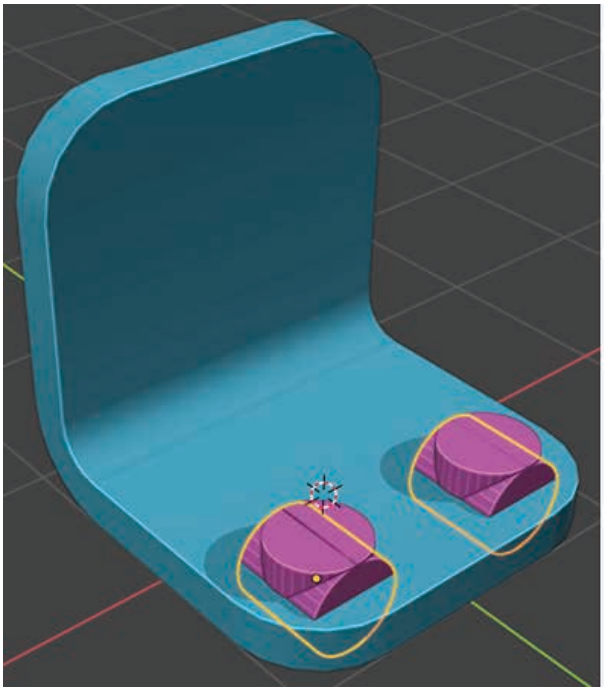
Press **Shift+D** to duplicate the cylinder, move it (G+X) and join both cylinders by holding Shift while selecting both and **pressing Ctrl+J** (join).



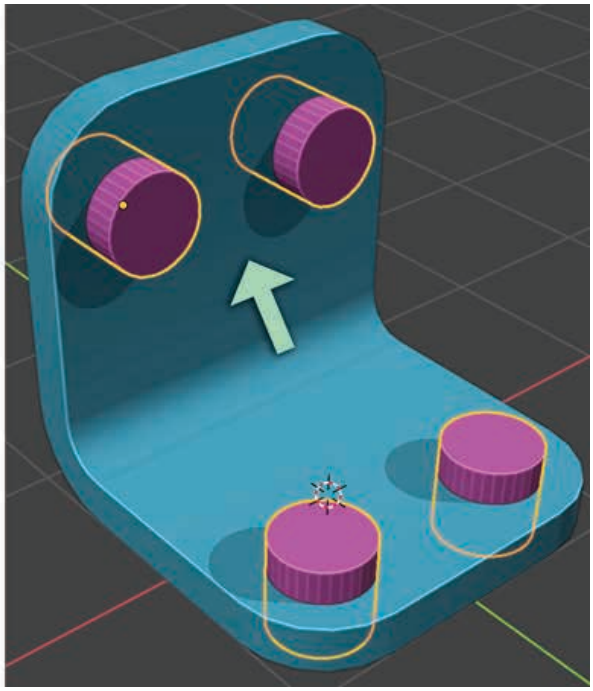
Press Shift+D again to duplicate the joined cylinders, rotate them 90° on the X-axis by **pressing R+X+90** and move them along the Z and Y-axis by **pressing G+Z** and **G+Y**.



Press **Shift+D** to duplicate the joined cylinders, then rotate them 90 degrees on the X-axis by **pressing R+X+90**.

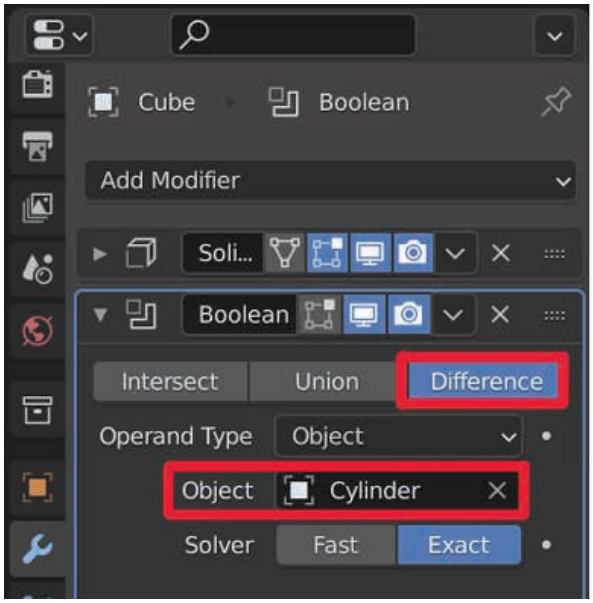


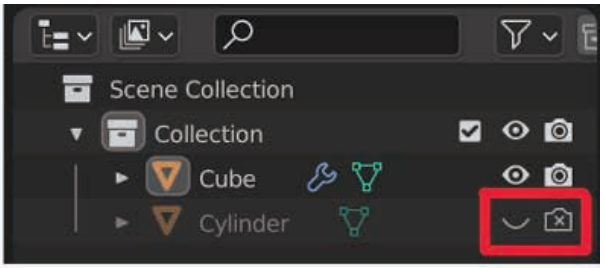
Move them up with **G+Z** and moving the mouse and **G+Y** and moving the mouse until they intersect the other object. Join them with the other cylinders with **Ctrl+J**.



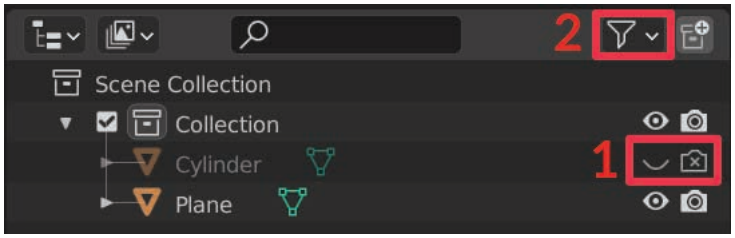
Select the original object (the Cube). Add a **Boolean Modifier** to the Cube, it should be set to **Difference**, which is the case by default. Choose the cylinders as the **Object**.

The cylinders should disappear, but don't worry if they don't: you can simply turn their visibility off in the **Outliner** by disabling the icons that look like an eye and a camera. That ensures you both can't see them in the viewport and that they won't show up in the render.

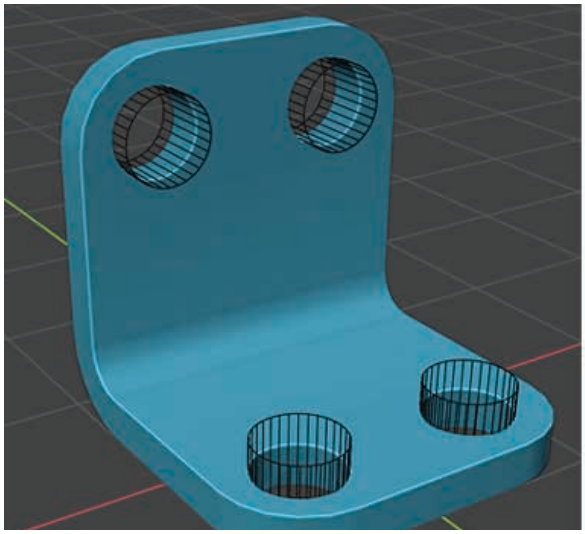


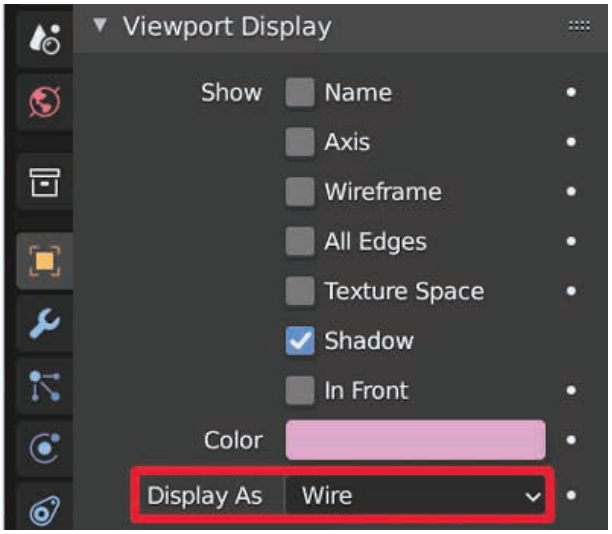


In case you don't see one of the icons (**1**), you can add it from the Filter drop-down menu (**2**).

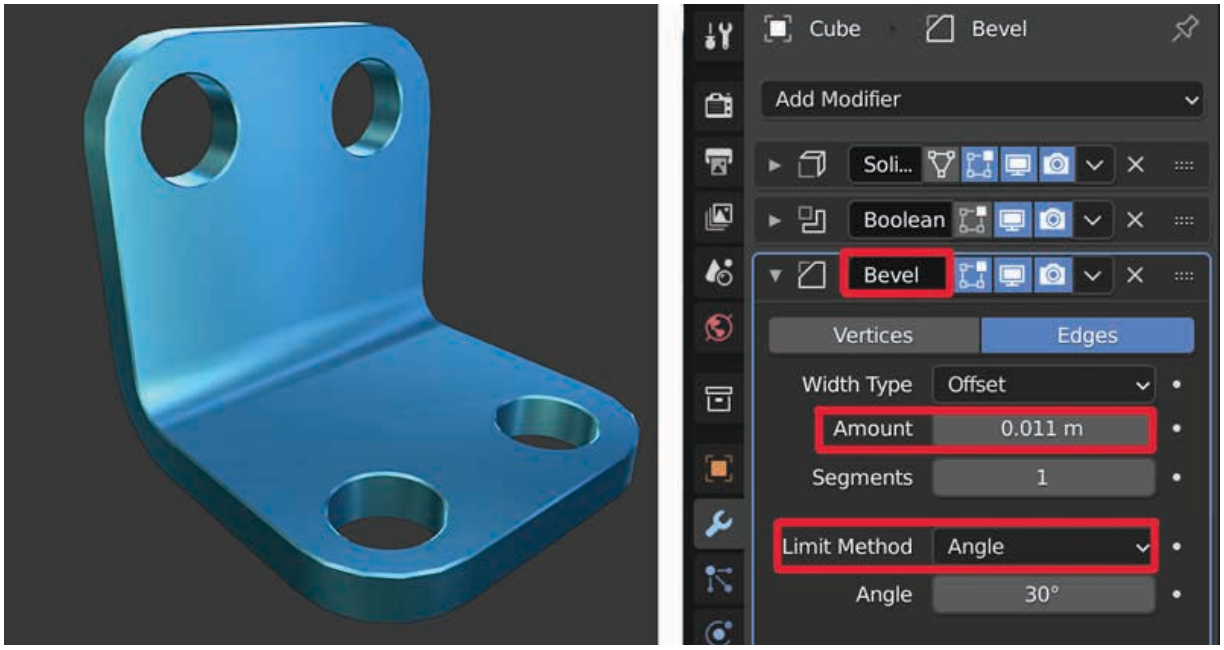


Instead of turning off visibility, you can set the **Display As** type of the cylinders to **Wire**. This way you can see through them, but you can still see where they are in case you want to move them. This won't be visible in the final render.





As a finishing touch, add a **Bevel Modifier** so the edges will reflect some light, like real metal objects do. The Bevel should be below the other modifiers. Play with the **Amount value** to get the right amount of bevel, and use **Angle** as the Limit Method.

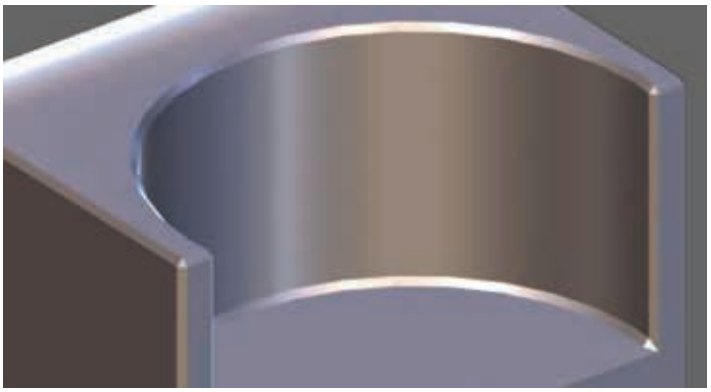


Click here to return to the index.

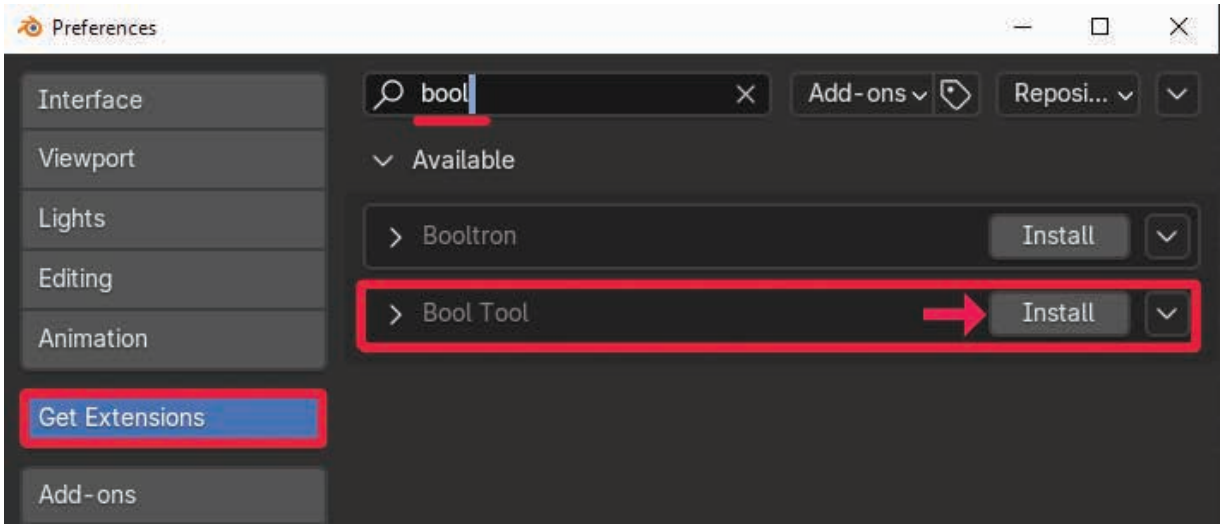
How To Use the Bool Tool Add-on

Bool Tool is an awesome add-on.

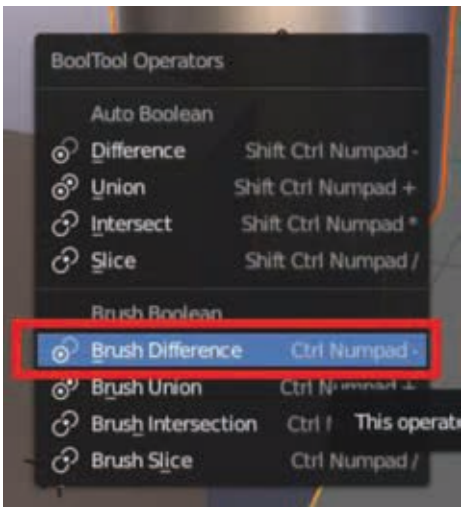
I was honestly afraid of using boolean operations before discovering this. Now it is an essential part of my workflow.



Bool Tool comes by default with Blender. Find it in the Get Extensions tab of the Preferences and install it.



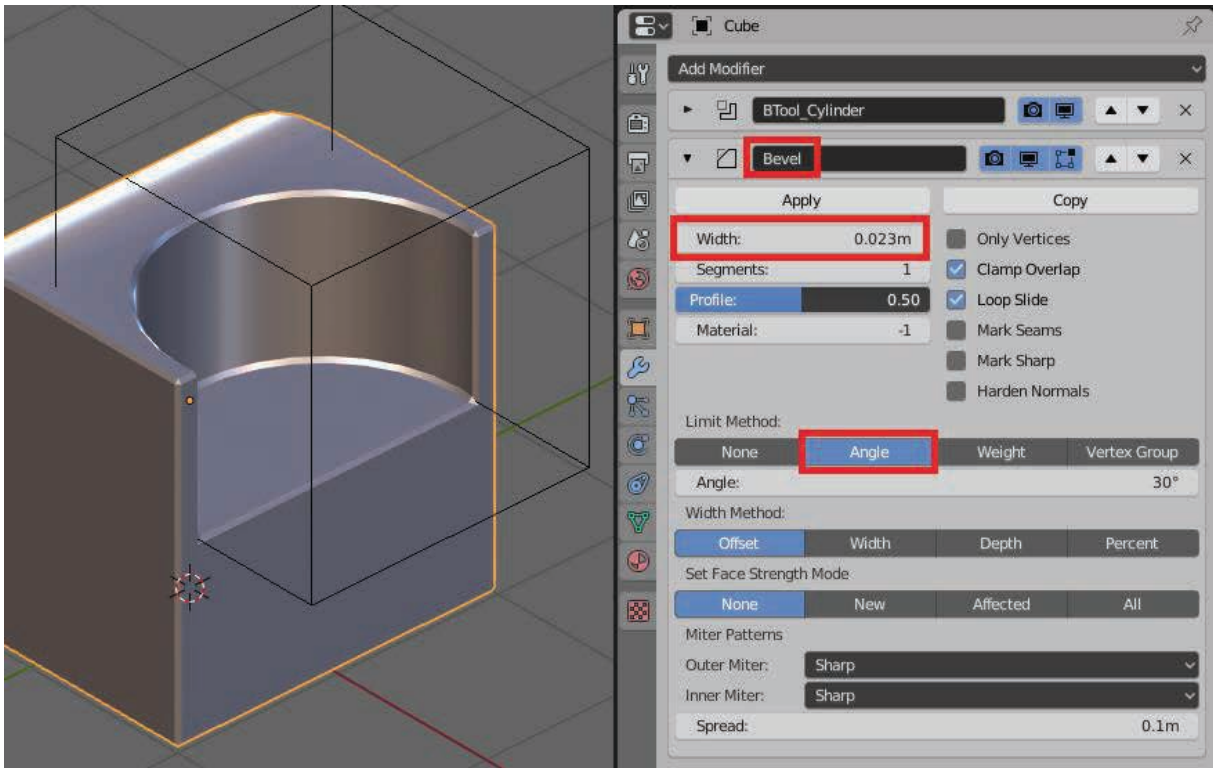
Select your boolean object first (the one that needs to take a "bite" out of the other object) and then Shift+select the object to cut it from. **Press Ctrl+Shift+B** to open the Bool Tool menu. Choose "Brush Difference".



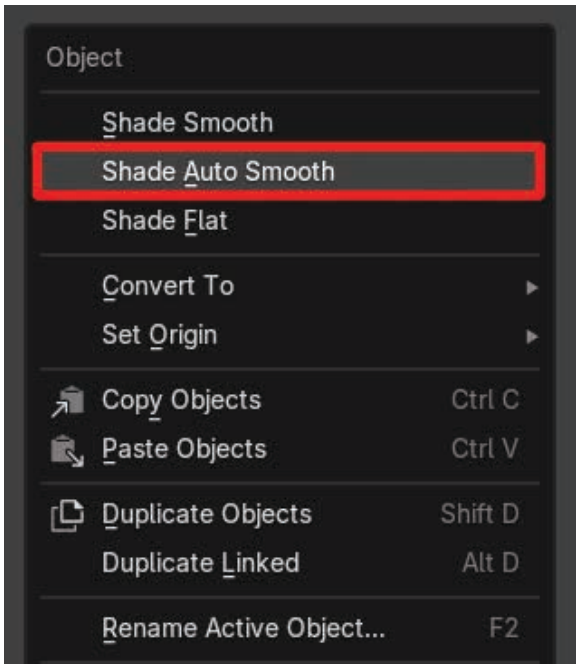
This turns the first object into a boolean **Brush**. So you can still move it around.

Alternatively you can simply **press Ctrl [-]** (minus on the numpad).

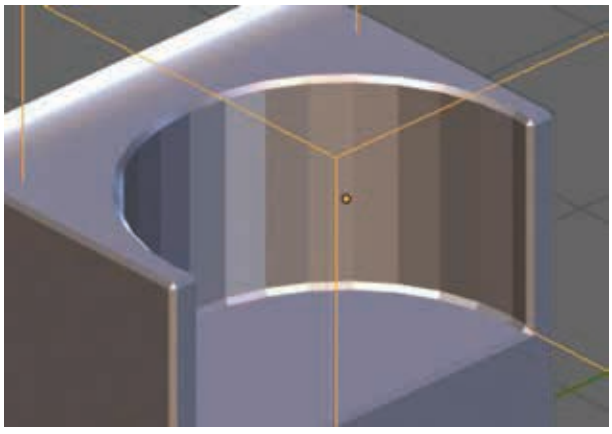
If you want you can add a bevel. Put it at the bottom of the modifier stack. As you can see a **BTool modifier** has been added by the Bool Tool add-on. The **Bevel modifier** is set to “Angle” in this case, as it got the best result.



Note that I have **Shade Auto Smooth** turned on for this object. To do this, **right-click** on the object and choose **Shade Auto Smooth** in Object Mode.



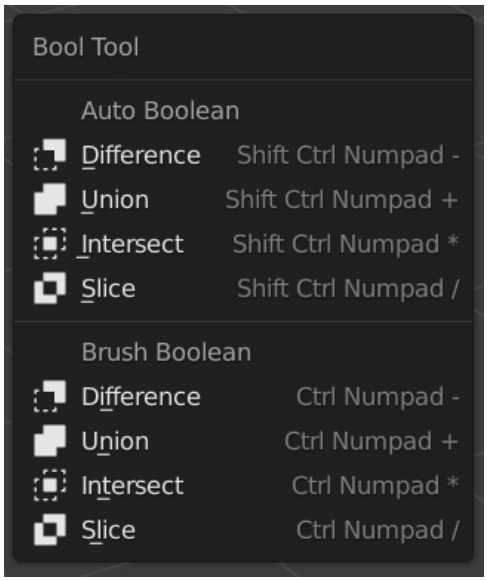
Otherwise the cut of the boolean object will have flat shading like this:



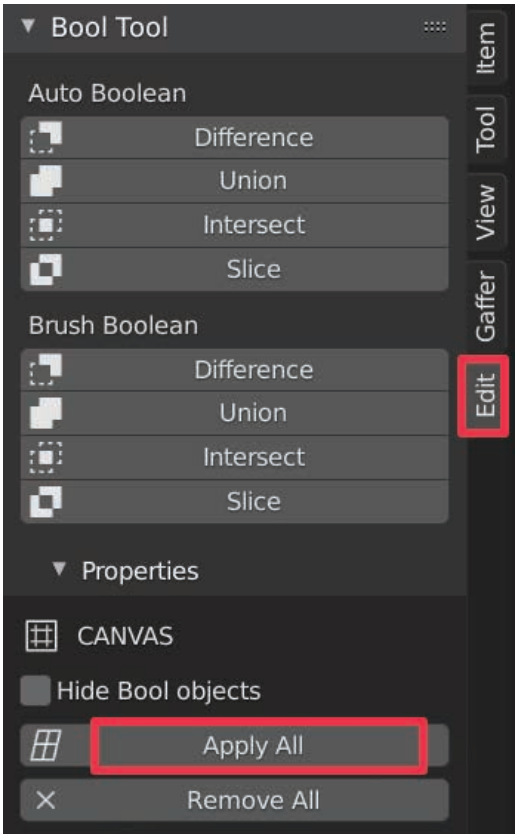
Interesting to know: the object that cuts a hole in another object is called the “Brush”. You can keep this so that your workflow remains non-destructive, but if you want you can apply the Brush in the Bool Tool menu.

Press Ctrl+Shift+B for the full Bool Tool options menu. There are other options to explore as well, not just this “difference” operation.

Alternatively you can open the Edit tab of the Options panel (open this by pressing N) to get these options.



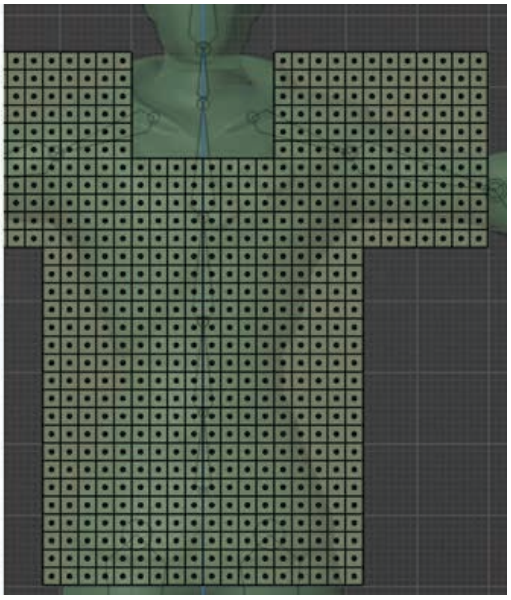
To apply the brush (to make the boolean cut permanent) click on Apply All in the Bool Tool options menu.



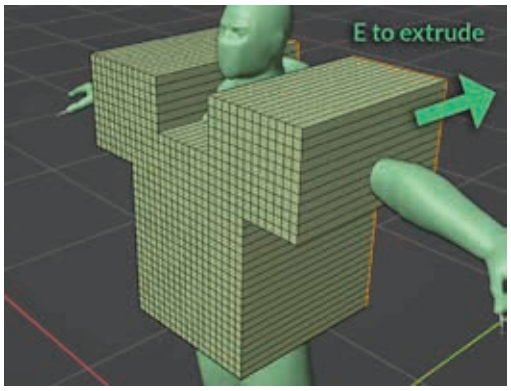
MODELING CLOTHING

Cloth Simulation Sewing basics

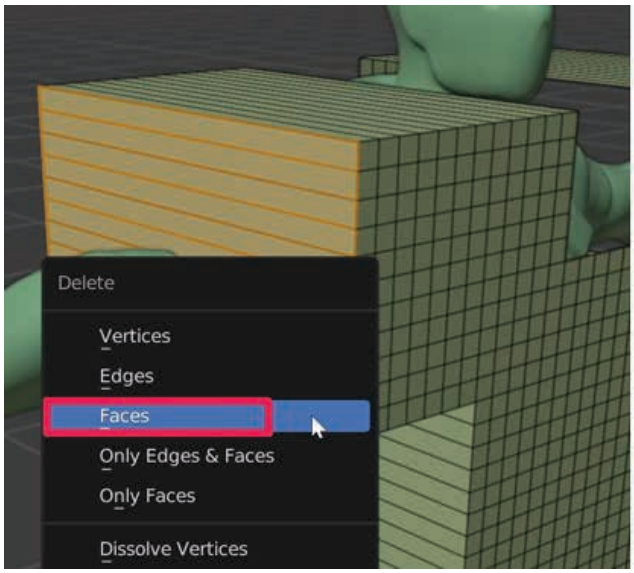
Make a Shirt by subdividing a **Plane** and deleting some **Faces**. You definitely need enough Faces for good Cloth simulation, at least as dense as shown here.

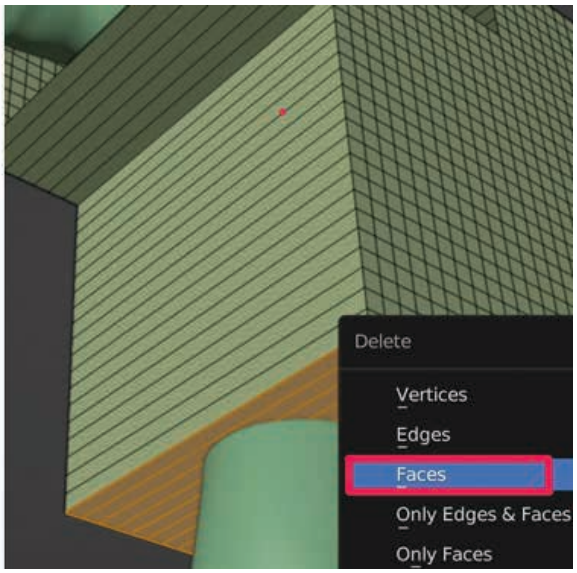


Then extrude it to the other side of the character by selecting it all and **pressing E**.

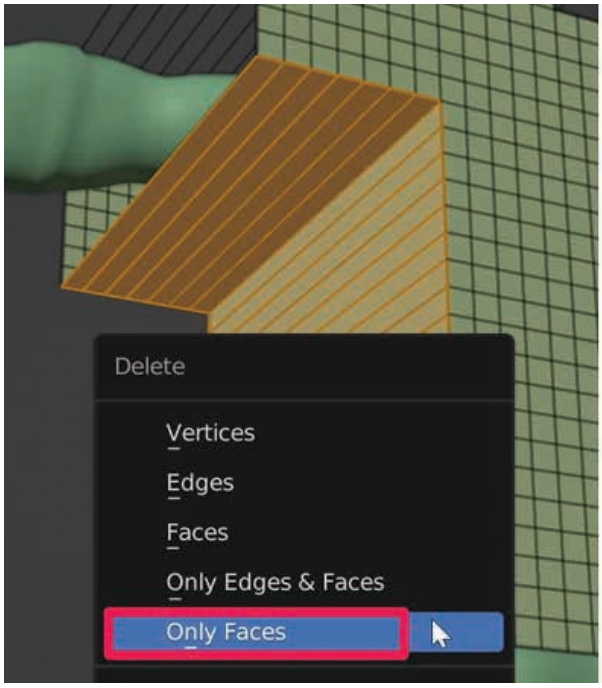


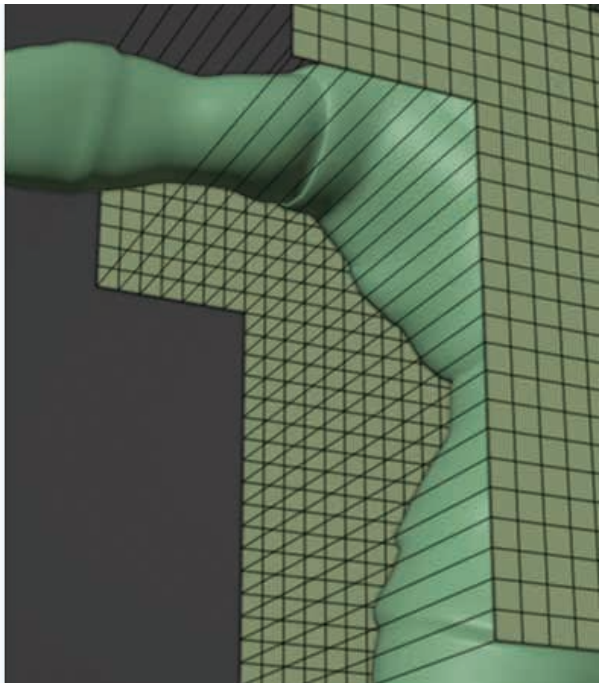
Delete Faces to make holes in logical places: a hole for the neck, for the arms, and at the bottom of the shirt.





Select the remaining faces between front and back and **press X > Only Faces**. This leaves the Edges, which will be use as stitches.

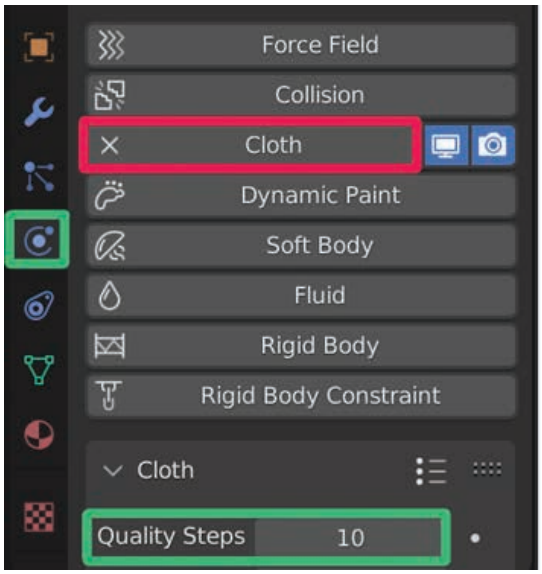




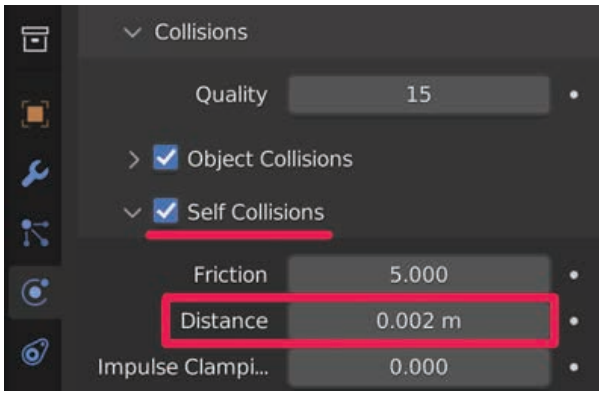
Select all and **press Shift+N** to recalculate normals. Add **Collision** to the character with **Thickness Outer** set very low, like 0.001 to 0.003.

Below you can see the difference: a higher value (like the default of 0.02) adds some space between the Collision object and the cloth.

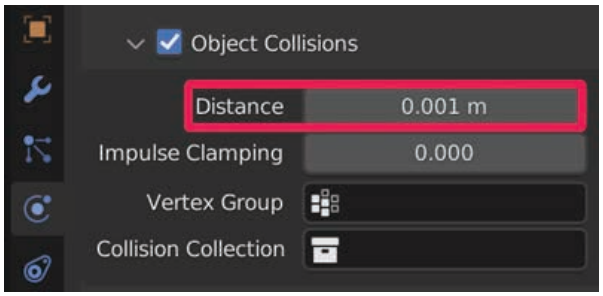
Add a **Cloth simulation** to the shirt and enable **Sewing** and **Self Collisions**. Increase the **Quality Steps** and **Collisions Quality**. The higher these values, the longer it will take to calculate the simulation. But sometimes it's necessary for the simulation to work well enough. The exact best values will take some experimentation for each project.



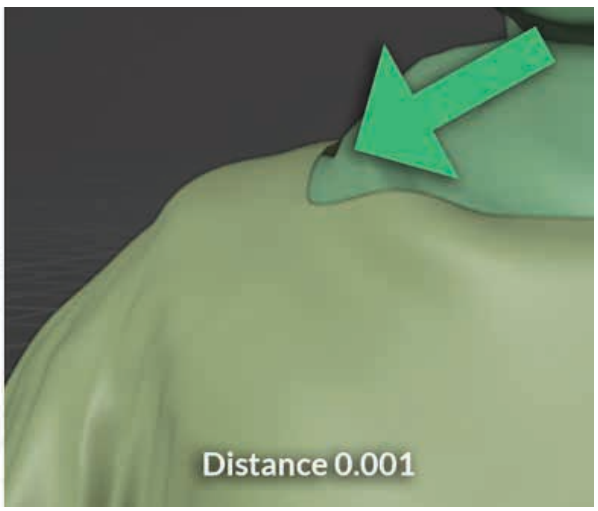
The **Self Collisions Distance** should be low, like 0.001 or 0.002.



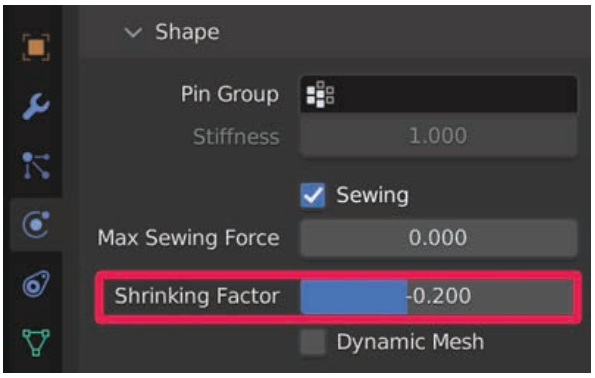
For **Object Collisions** use a small Distance like 0.001 or 0.002.



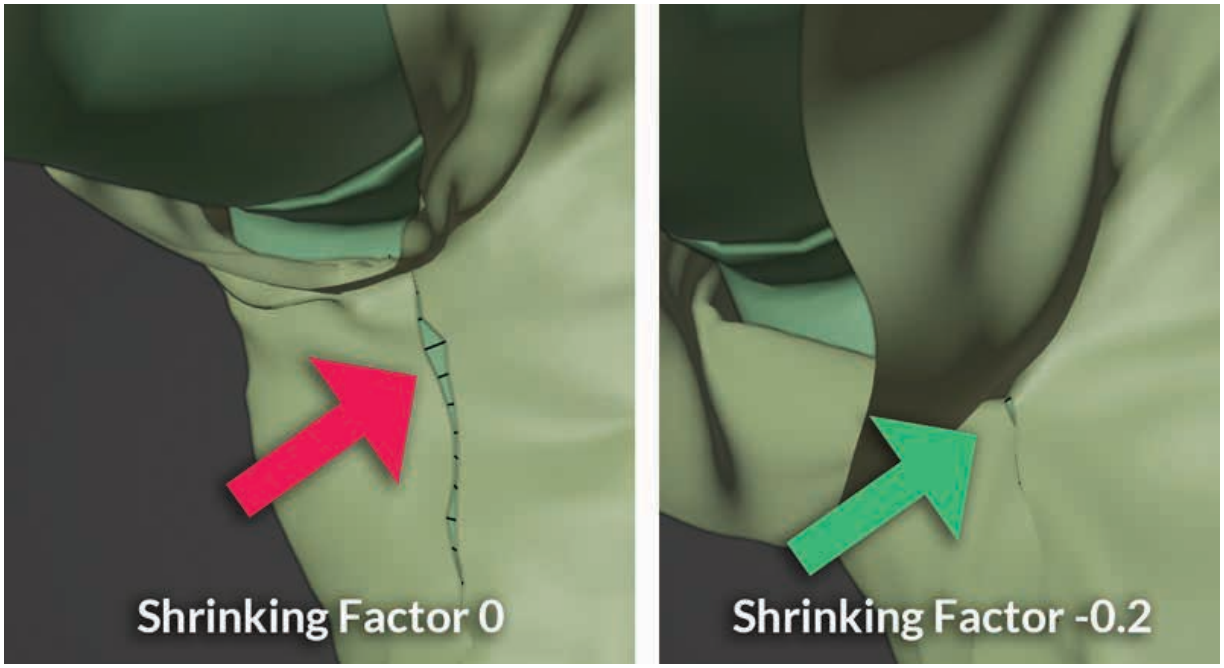
Like with the Thickness Outer setting of the Collision object, you can see that this Object Collisions Distance value increases the space between the character and the cloth.



To fix gaps where parts were sewn together, set **Shrinking Factor** to a small negative number like -0.1 to -0.2.

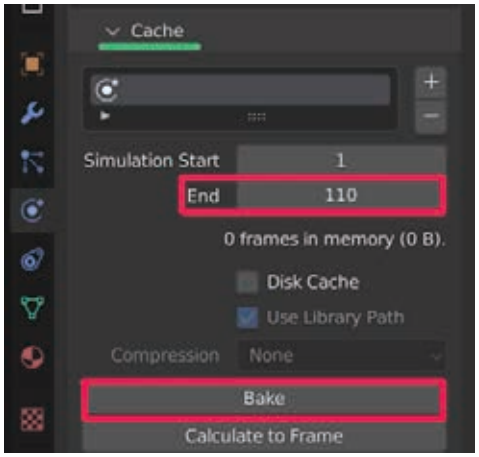


You can see the difference below:



Press the Spacebar or **Bake** the Cache to run the simulation.

Make sure you set a reasonable end frame for the Cache, otherwise you'll be waiting for a long time.

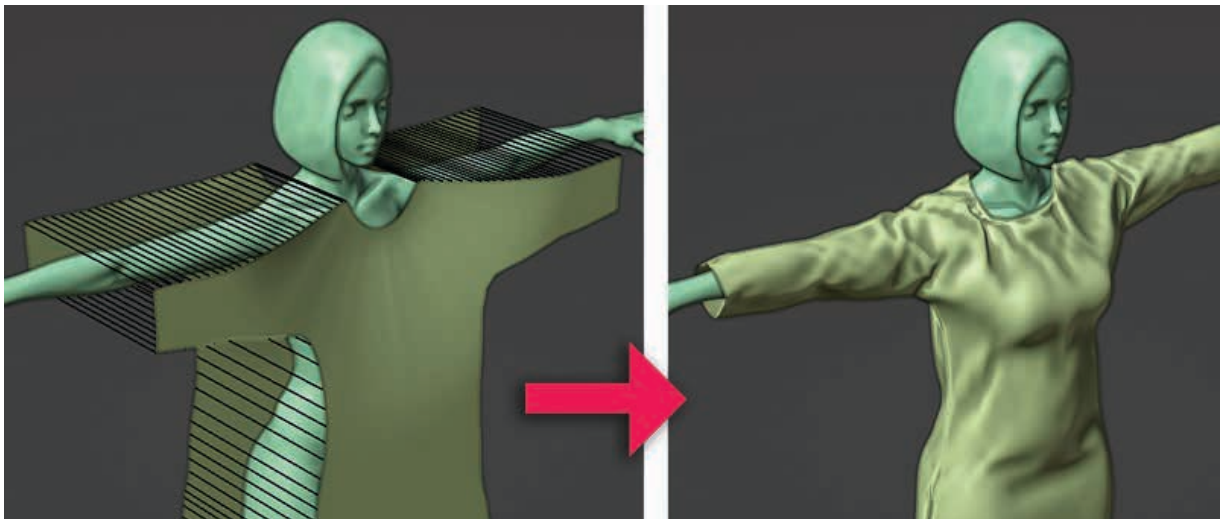


Finally you can add a **Solidify modifier** to add some thickness.

Bonus tip: In some cases, (temporarily) scaling up the objects can help improve the simulation in Blender. Certainly, smaller objects can be tricky to simulate. You can always scale them back down after baking the simulation to keyframes later.

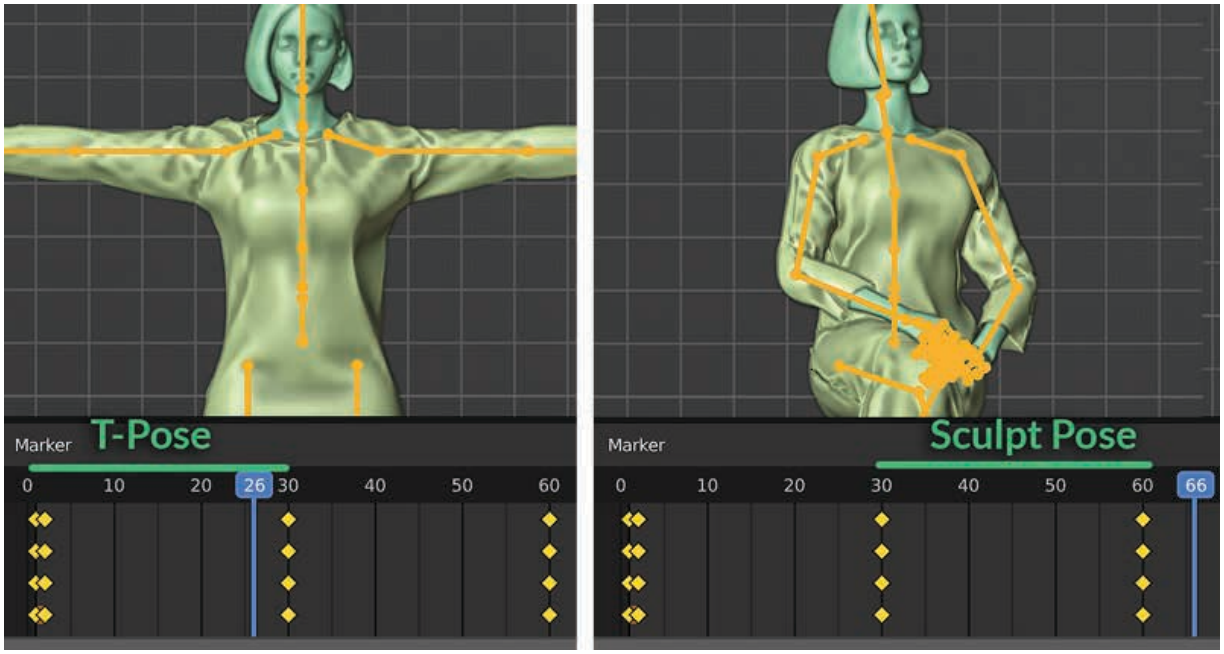
Posed Cloth Simulation for Sculpting

Let's use what we previously learned about the **Sewing** function of **Cloth** simulations for Sculpting.

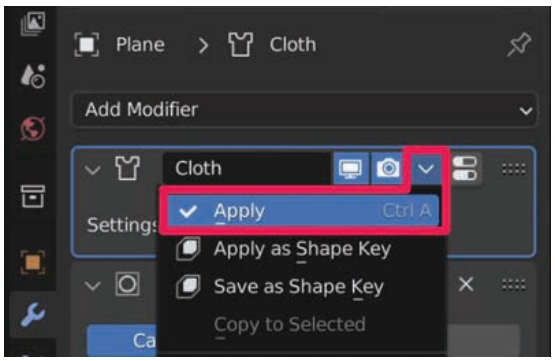


Woman Base Mesh by Rodesqa *on Artstation*

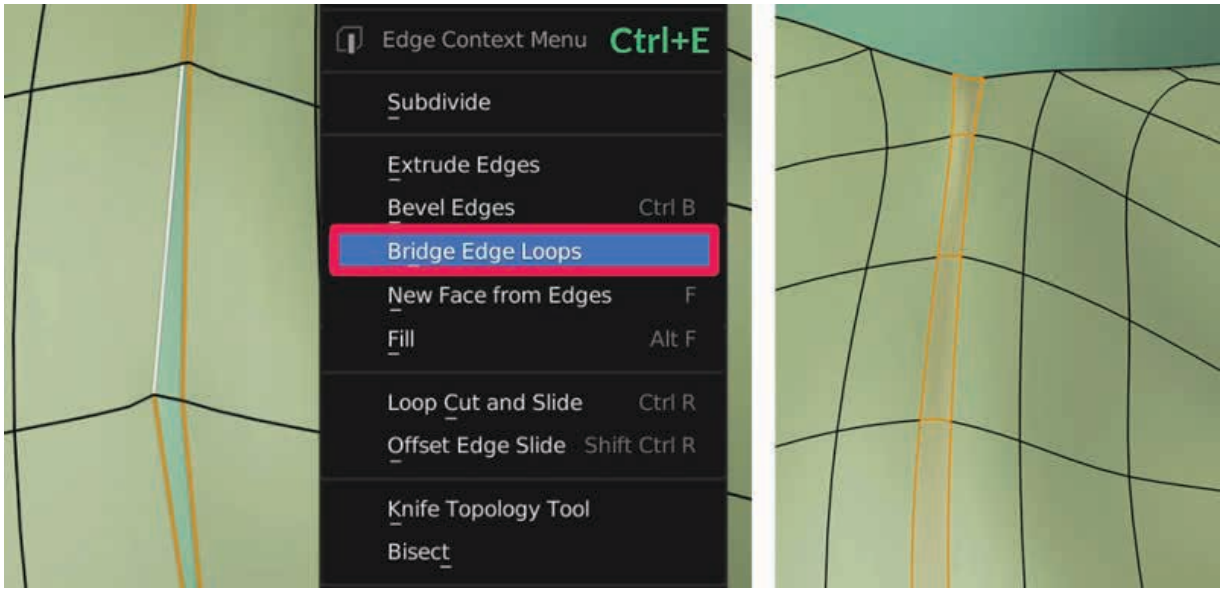
I recommend starting your character in a **T-pose** for a second or two (30 frames let's say) before posing it for your sculpt. That way the Cloth sim has time to settle into the initial clothing shape. Then pose the character the way you want it to be in the sculpt.



Choose a frame where it looks good and **apply** the Cloth modifier.

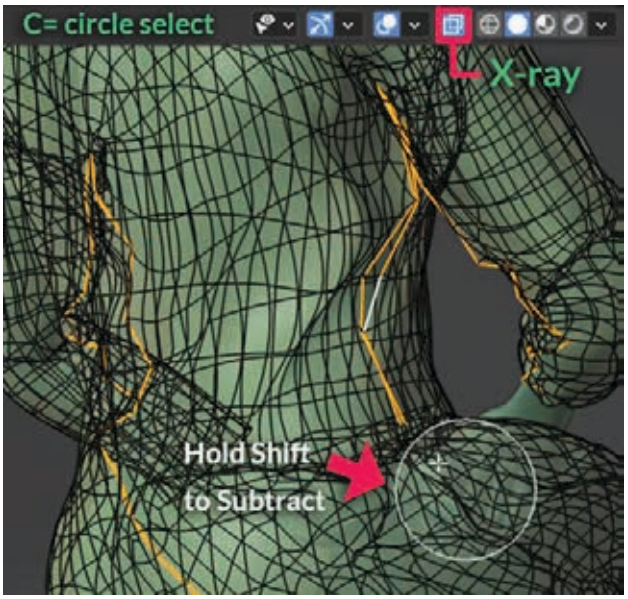


You can close the gaps by **Bridging Edge Loops**.

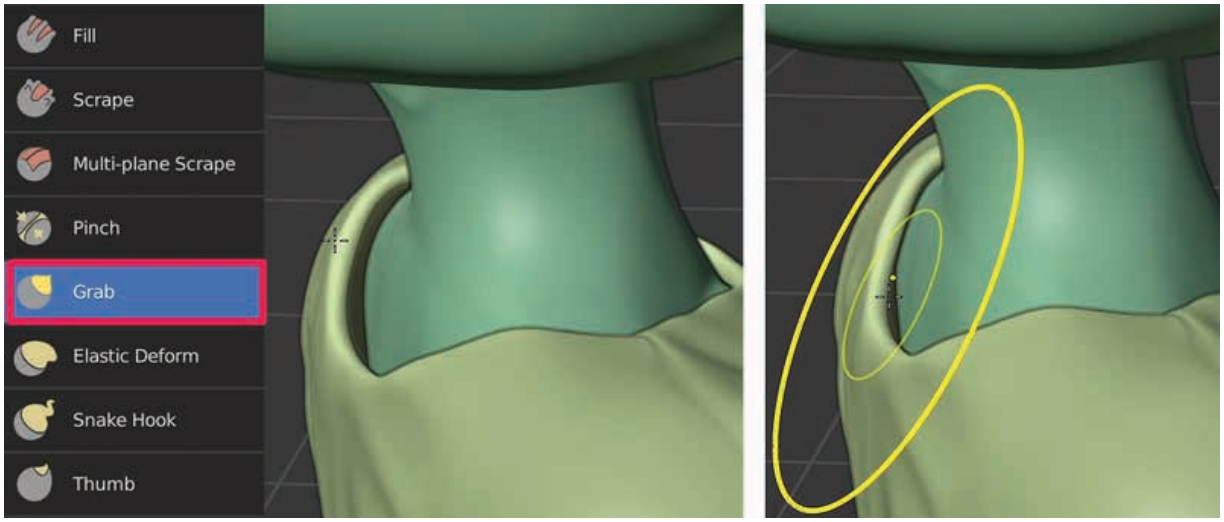


You may have to deselect loops which you can quickly do with **Circle Select (C)** so that you don't bridge too many edge loops at once.

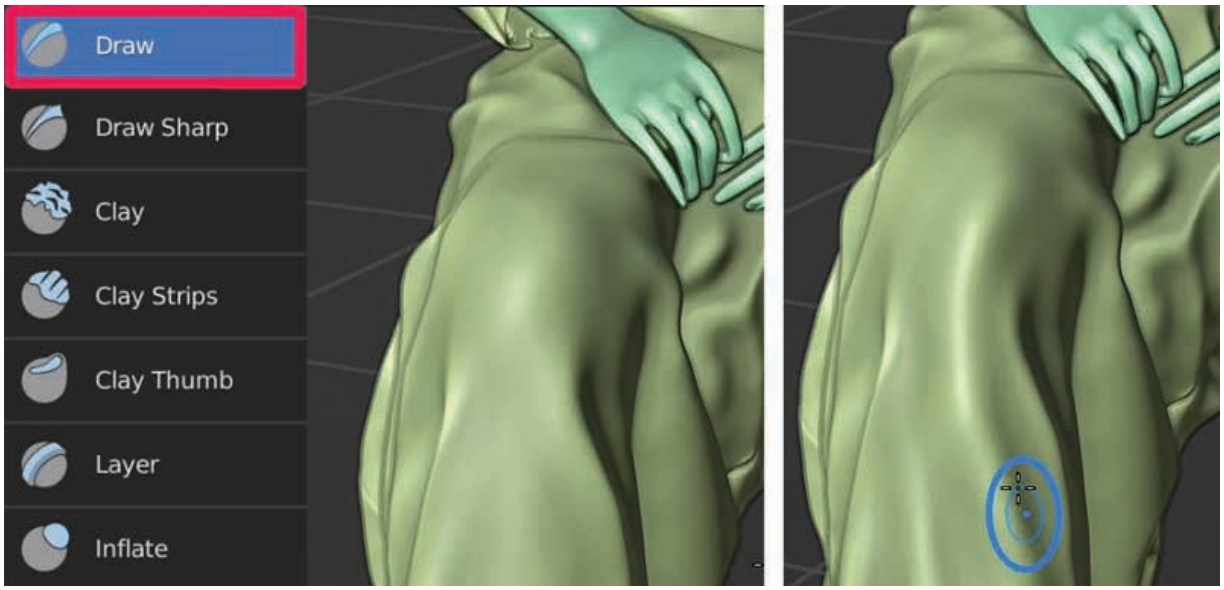
Hold Shift to subtract from the selection, and **right-click** to exit Circle Select mode.



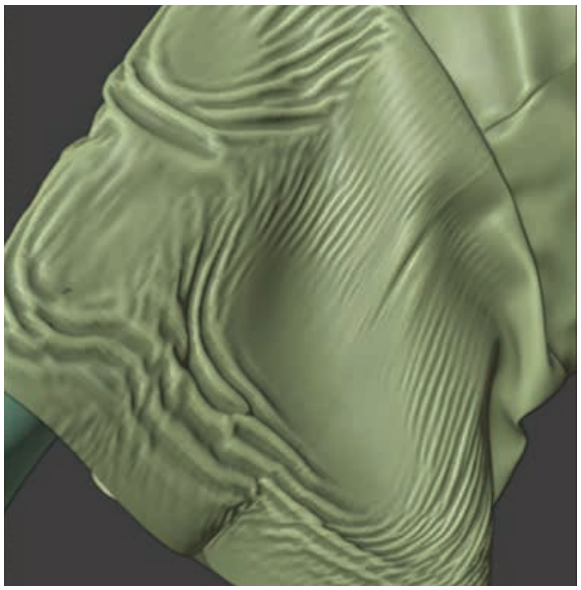
In **Sculpt Mode** use the **Grab brush** to move the cloth closer to the character, in order to close some of the more obvious gaps.



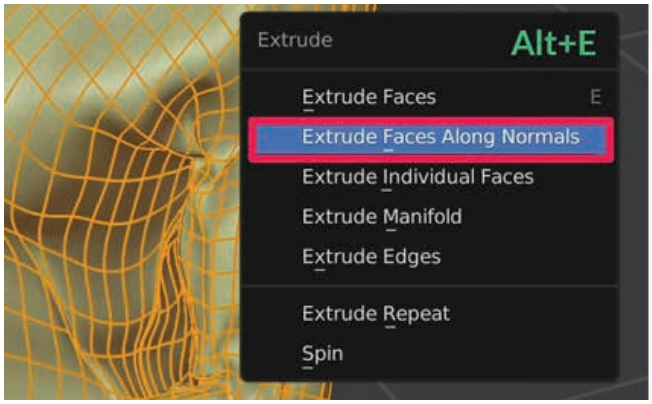
Next use the **Draw brush** in combination with the **Smooth brush (hold Shift)** to add some more folds.

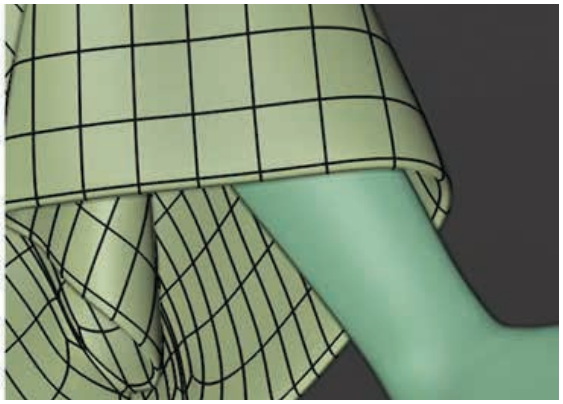


For smaller detailed folds, you can use the various Cloth brush Deformation types, as demonstrated previously.



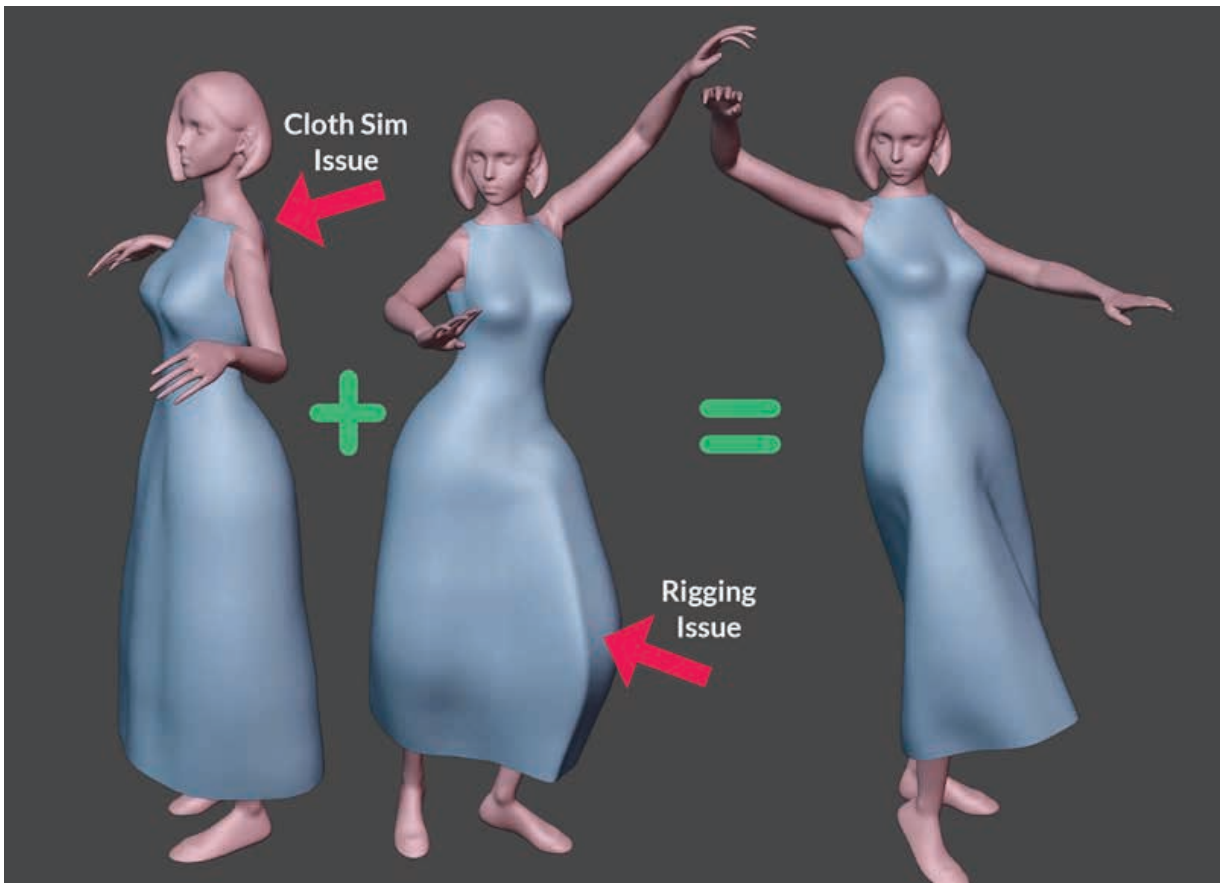
Select all and **press Alt+E** to choose **Extrude Along Normals**. Carefully extrude to add some subtle thickness. Be careful not to have intersecting parts, you may have to choose either extruding inwards or outwards, based on what works best for your mesh.



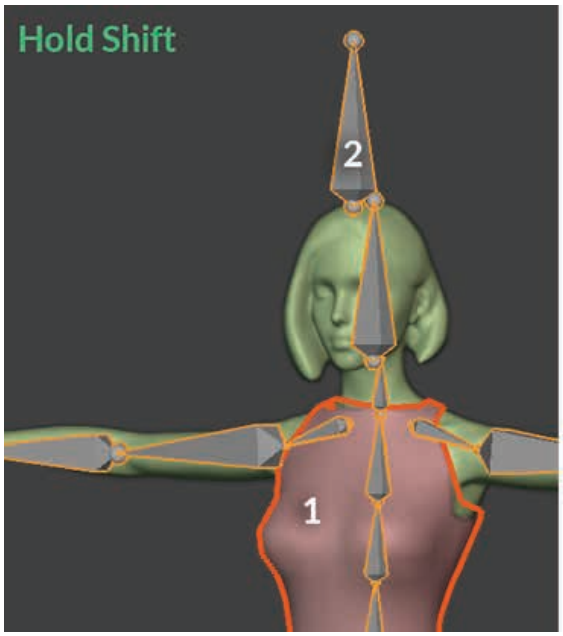


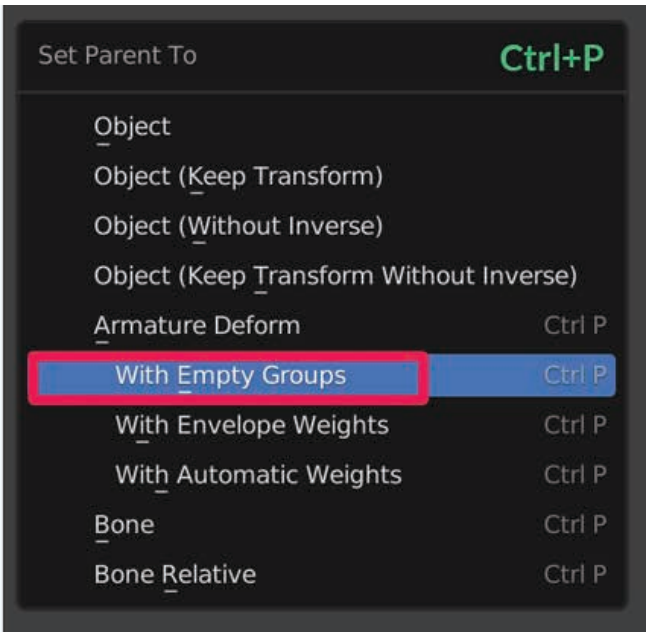
Mixing Armatures and Simulations for Clothing

We can animate a dress on a character using both a Cloth simulation, and rigging. Both have their unique challenges: a Cloth sim may take endless tweaking to get just right, and rigging may not fit for parts that don't conform to the body, such as the bottom part of a dress. Luckily, we can use the best of both!

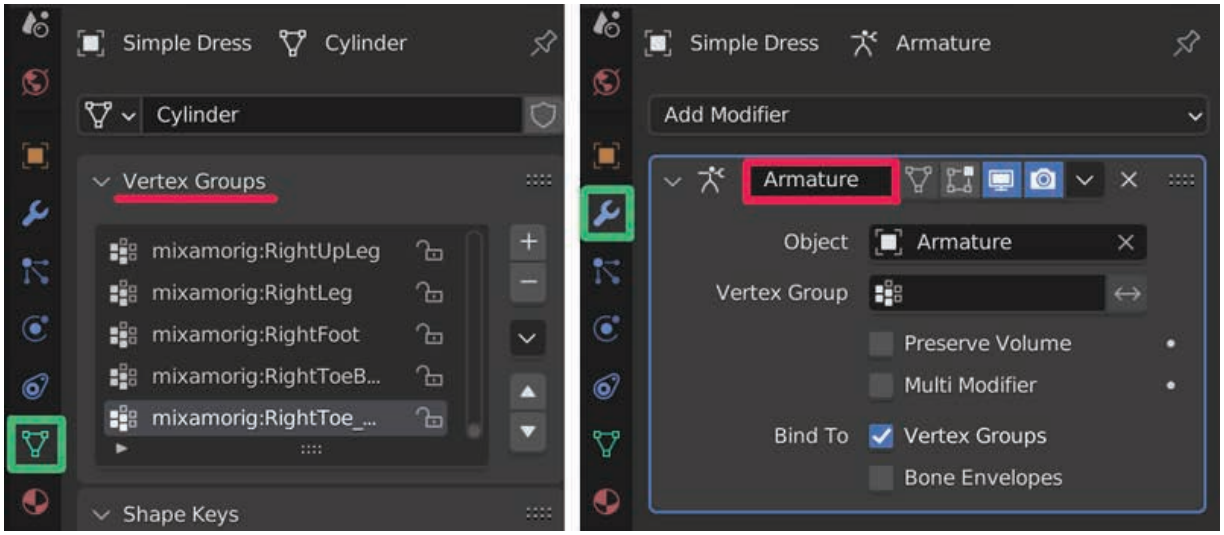


Select the dress and then the Armature while **holding Shift** and then **press Ctrl+P**. Choose **With Empty Groups**.

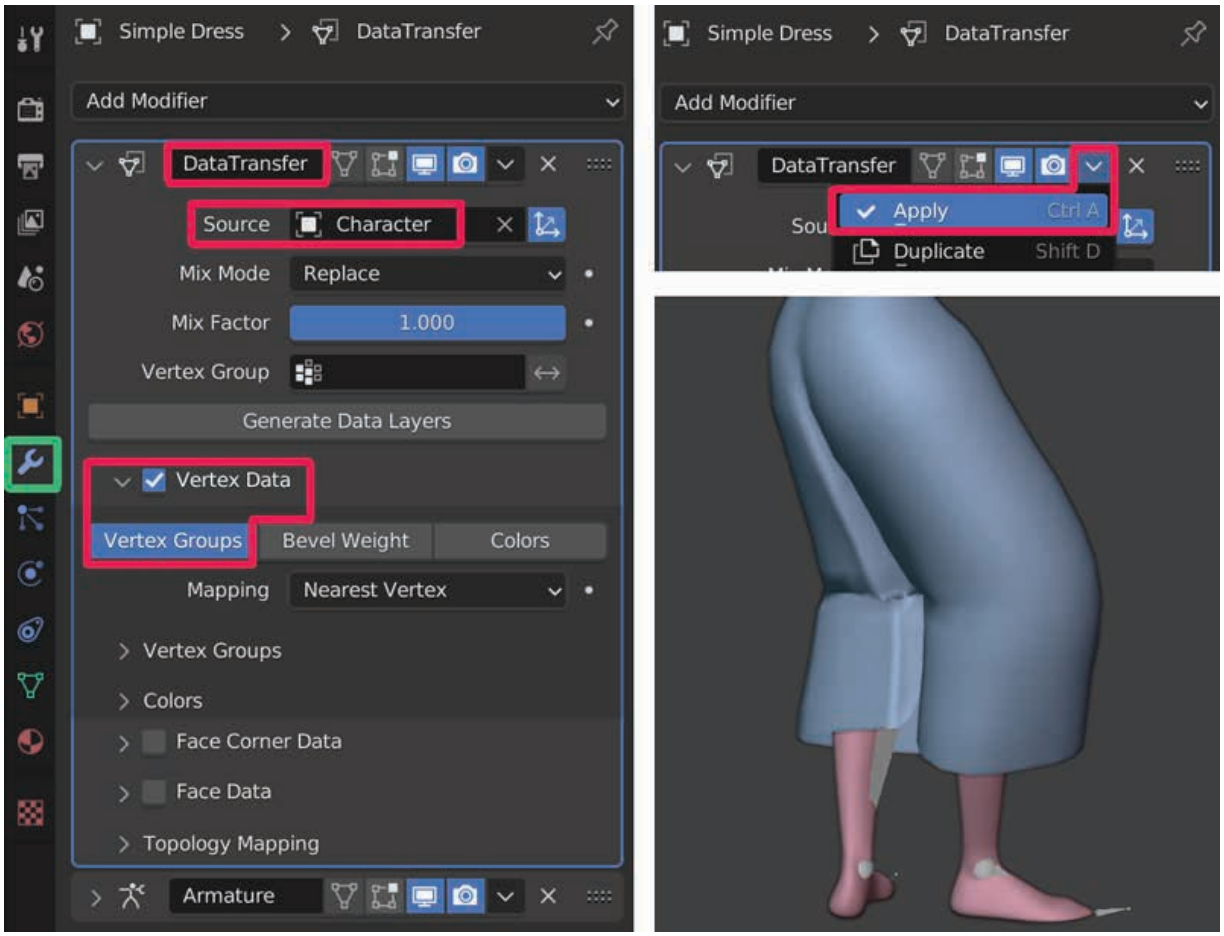




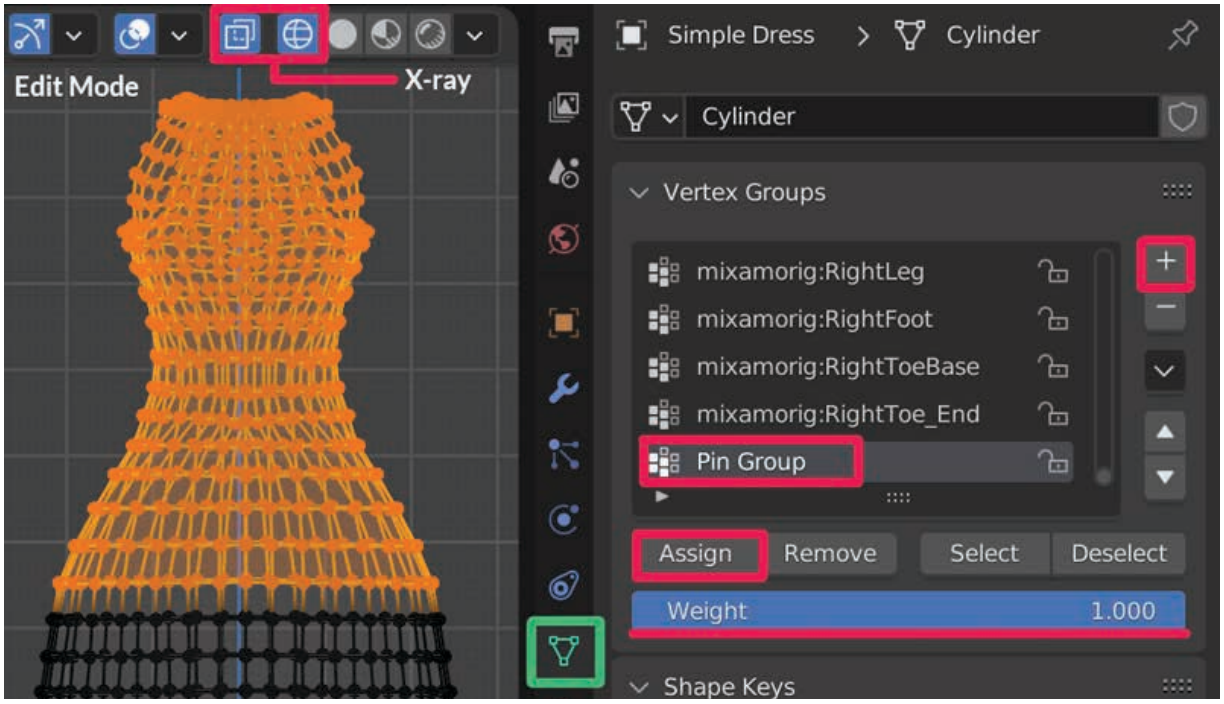
This creates Vertex Groups on the dress that match those on the character, in other words corresponding to the bones of the Armature. It also adds an Armature to the dress.



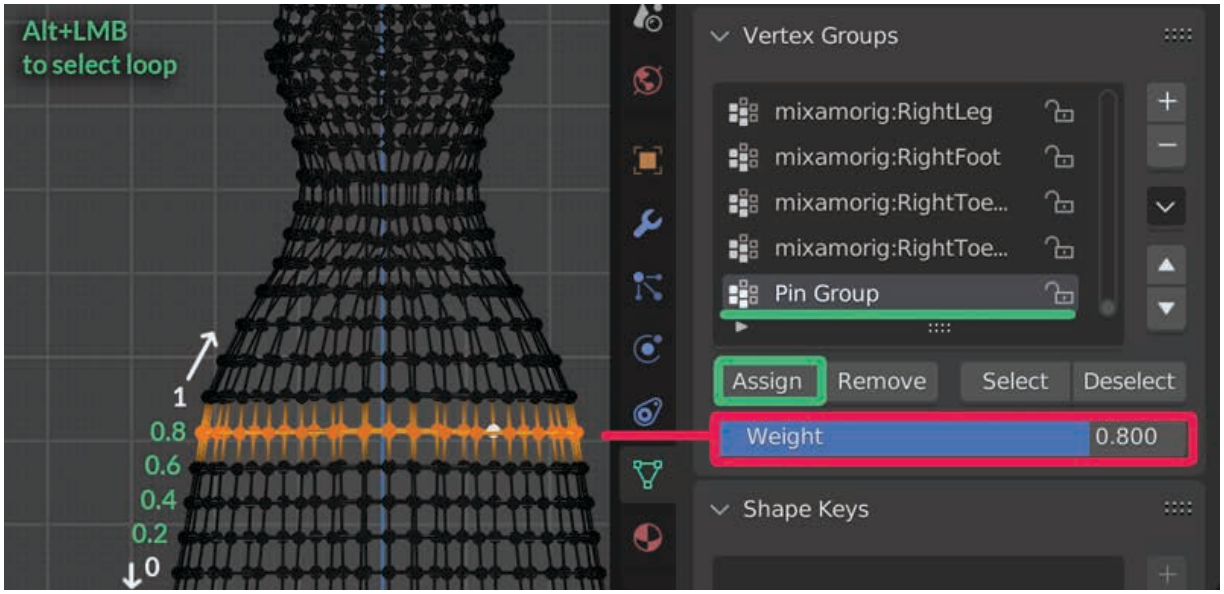
Give the dress a **Data Transfer modifier** and move it above the Armature. Check **Vertex Data** and **Vertex Groups** with the character as the **Source Object**. If you apply this modifier now, the dress follows the character. But the bottom part doesn't look good.



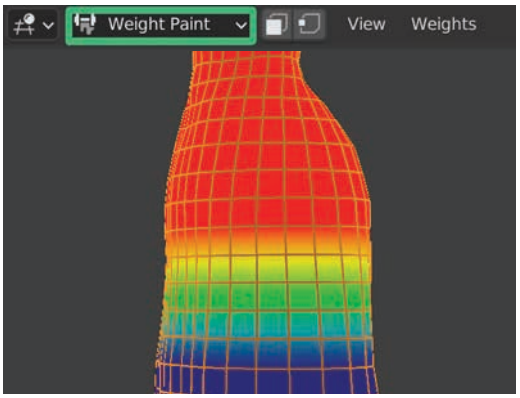
Create a new **Vertex Group** and name it **Pin Group**. Add the top of the dress to it with a value of 1.



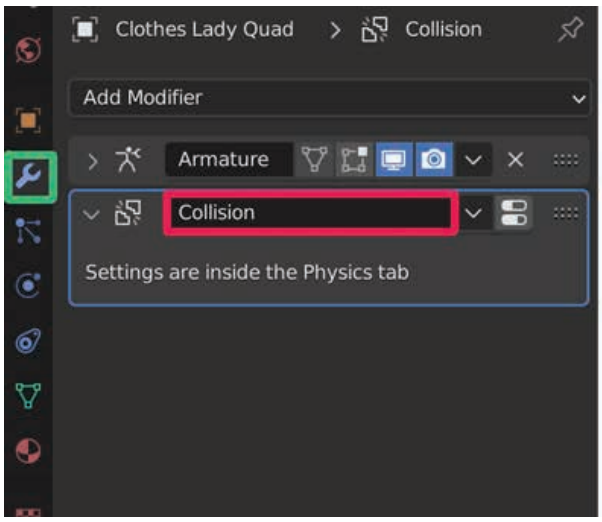
Then select loops underneath that and give them decreasing values one by one: 0.8, then 0.6, then 0.4, and 0.2.

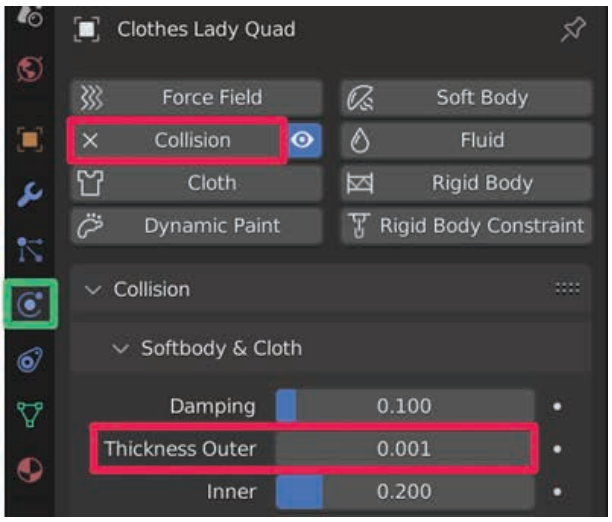


For reference, you can see in **Weight Paint** mode what it looks like.

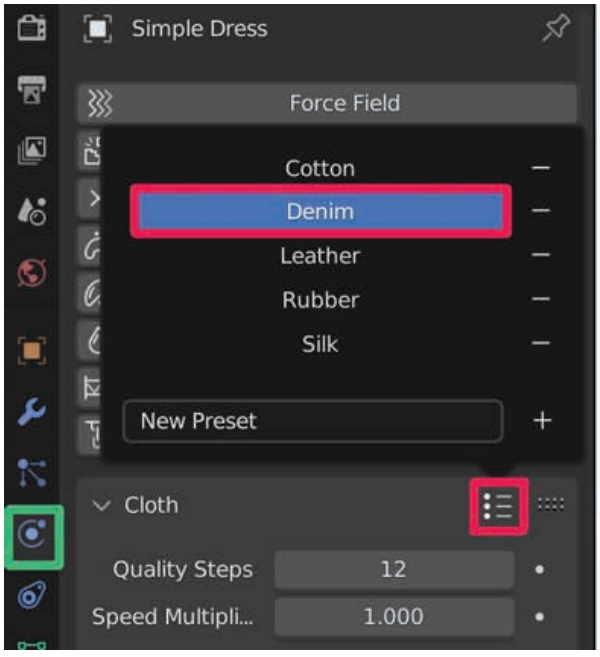


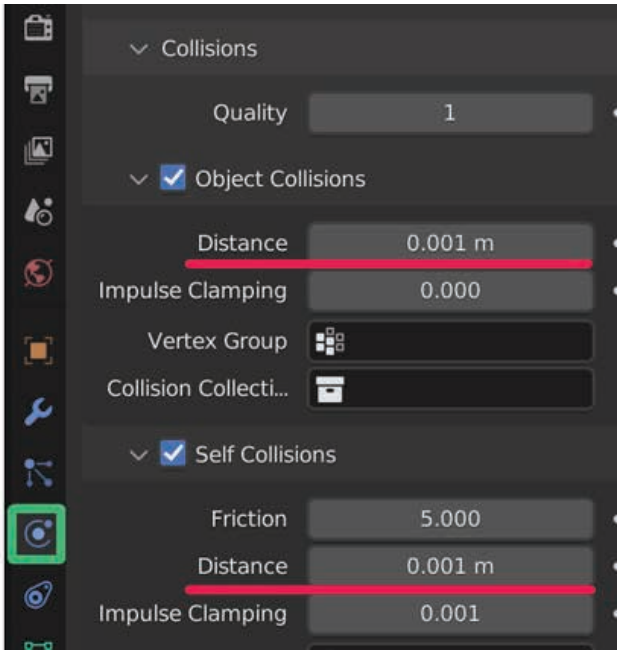
Add **Collision** to the character. Make sure the Armature modifier is above it. For the Collision settings, I only changed the **Thickness Outer** value to 0.001 so there wouldn't be too big a gap between the clothes and the character.



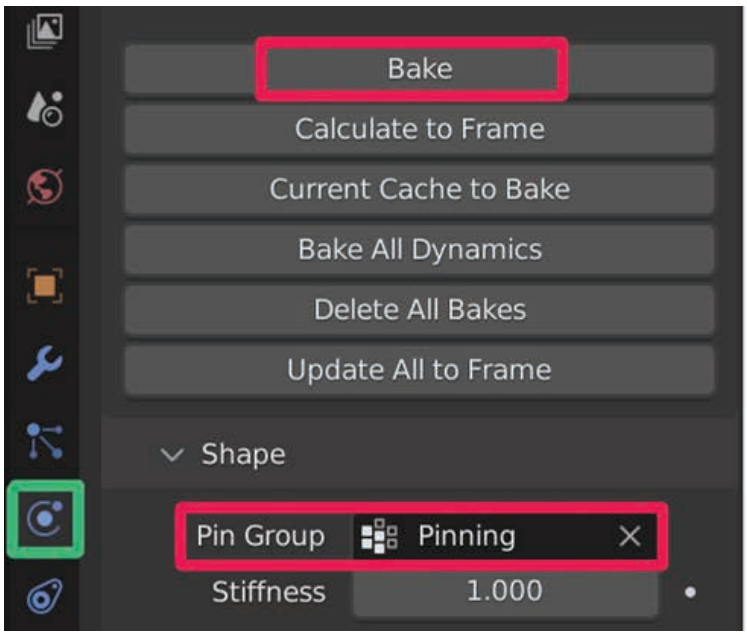


Add a **Cloth sim** to the dress. Enable **Self Collisions**. and for the Cloth sim. I used the Denim preset as it seemed to work best with this particular animation. Like with the Collision Thickness Outer setting, I reduced the object and self collision **Distance** for the Cloth sim to the lowest setting.





Under **Shape > Pin Group** choose the Vertex Group that you created. Bake to test how it works. Now the bottom of the dress is simulated and the top follows the armature.

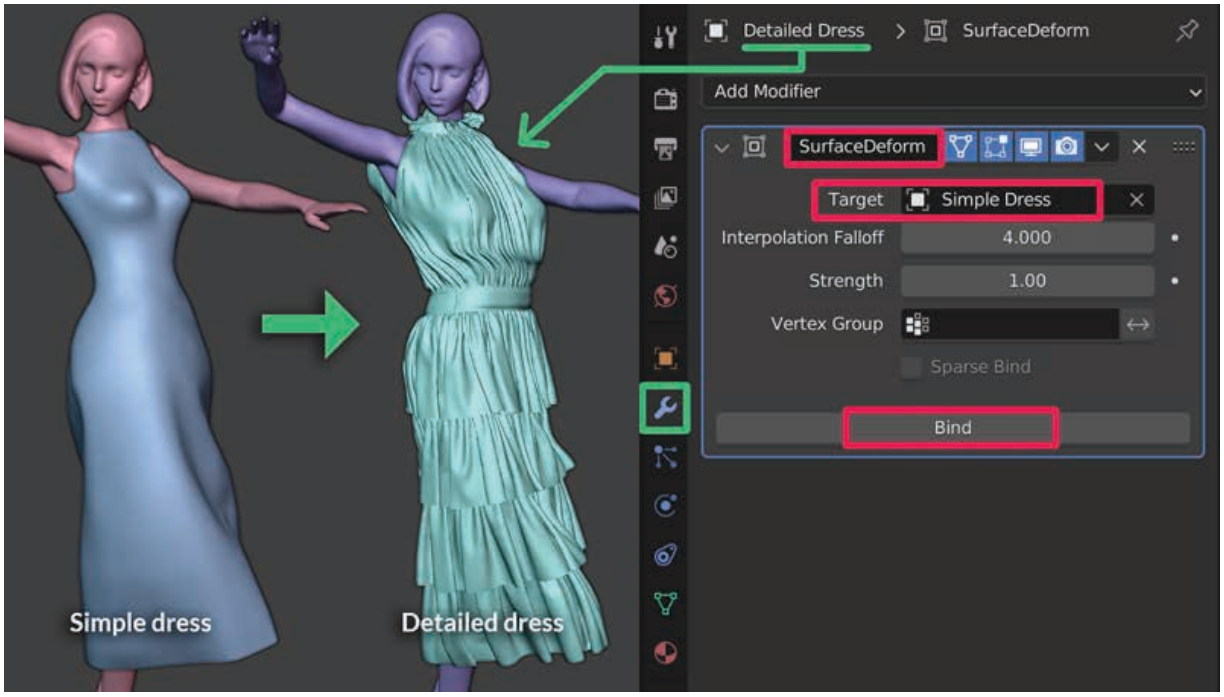


Detailed Clothes simulated Easily

Previously we looked at creating a simple dress and animating it using a blend between Cloth simulation and weight painting for an Armature. We can use this animated simple dress to animate a more detailed dress that would be challenging to animate with a Cloth sim because of the high polycount. Mainly because it would take forever to bake the simulation.

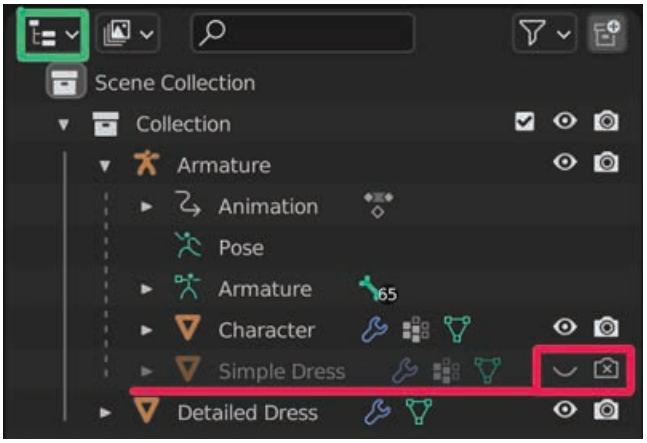
Dress model by Strange Alice on Artstation.
Character Base Mesh by Rodesqa on Artstation.

Add a **Surface Deform modifier** to the detailed dress. As the **Target** choose the simple dress and click on **Bind**.



This binding process might take a little while, after which the high poly dress will follow the low(er) poly one.

Turn off the **view** and **render visibility** of the simple dress.

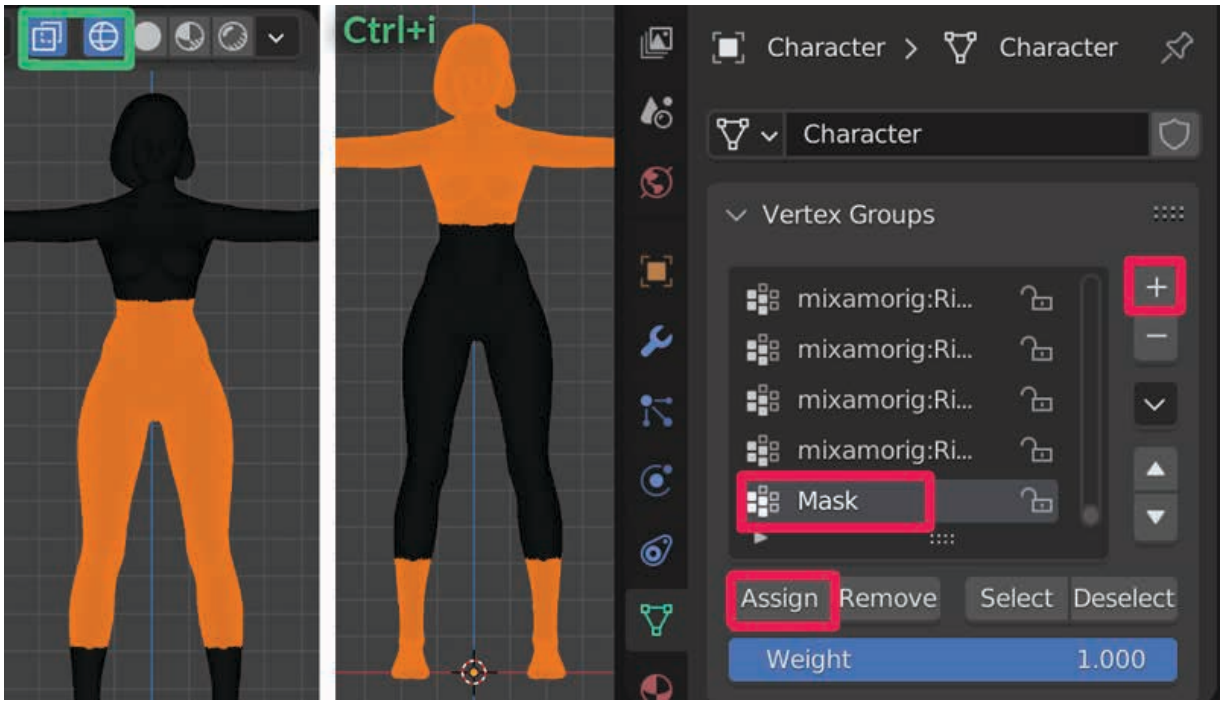


If some parts of the character clip through the dress, you can use a **Mask modifier**.

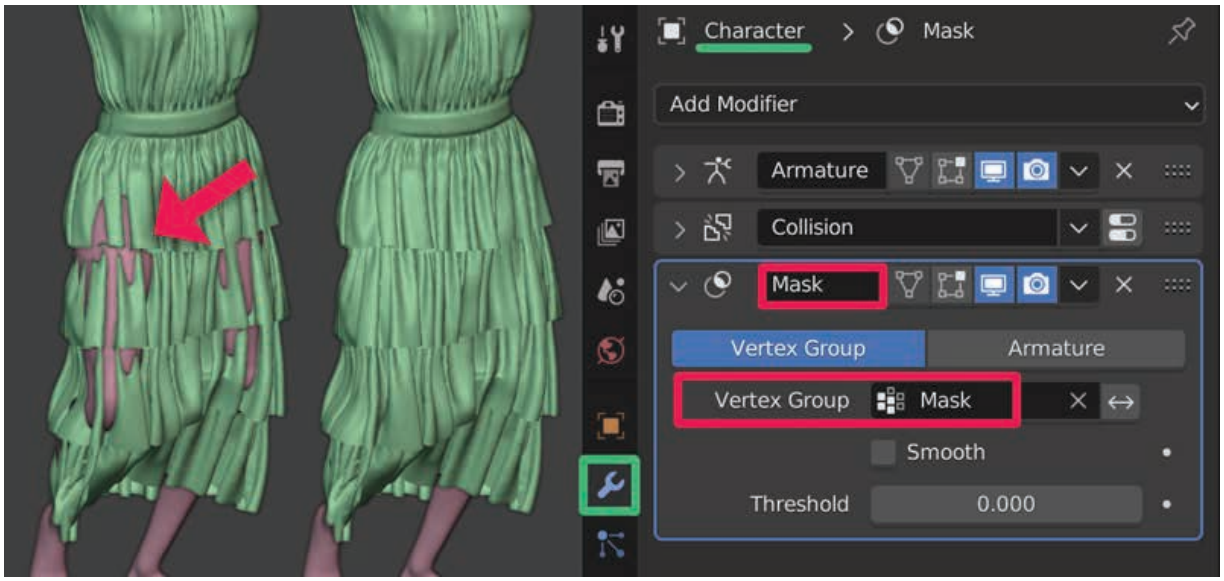
Create a new **Vertex Group** called Mask and select the parts of the character you want to hide in Edit Mode **X-ray** view.

Invert the selection by **pressing Ctrl+i**.

Click **Assign** to add the selection to that Vertex Group.



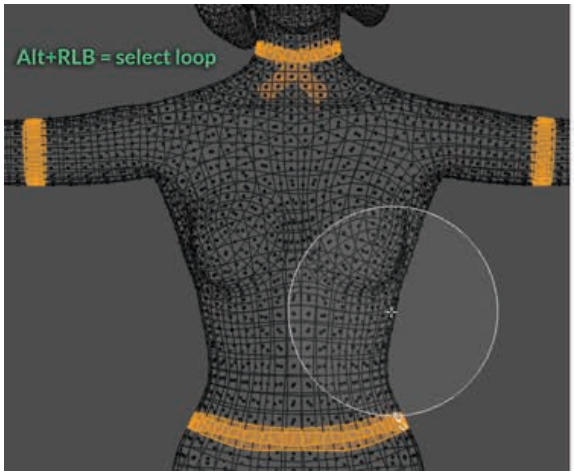
Add that Vertex Group to the Mask modifier.

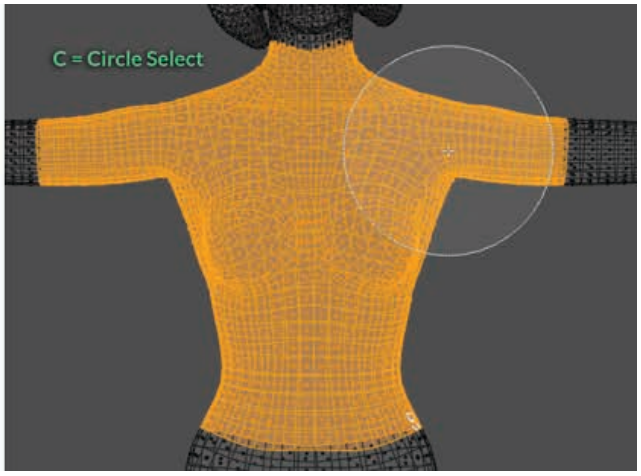


Now you've got a detailed high poly dress animated by a hidden simple dress that's faster and easier to animate.

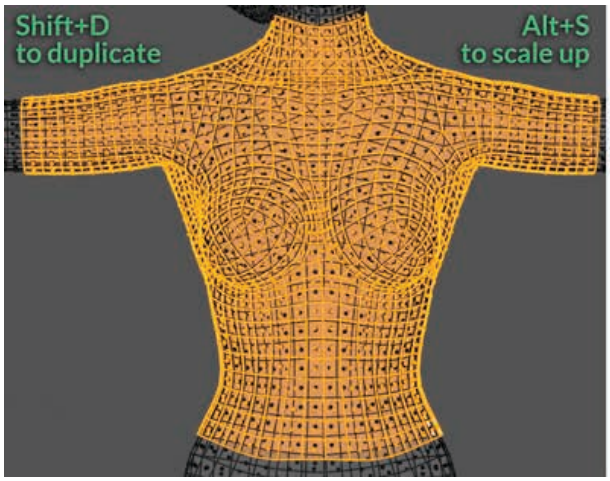
Model a Simple Dress

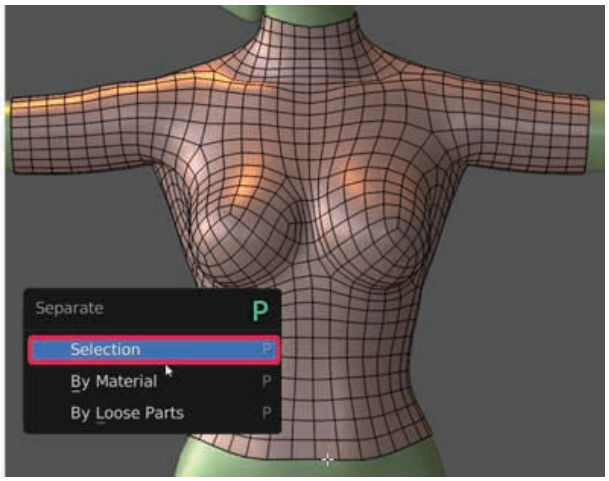
Select the top of the character where the dress should be. You can select loops with **Alt+Left Mouse Button** and then fill in the rest with **Circle Select** by **pressing C**. Exit Circle Select by right-clicking.



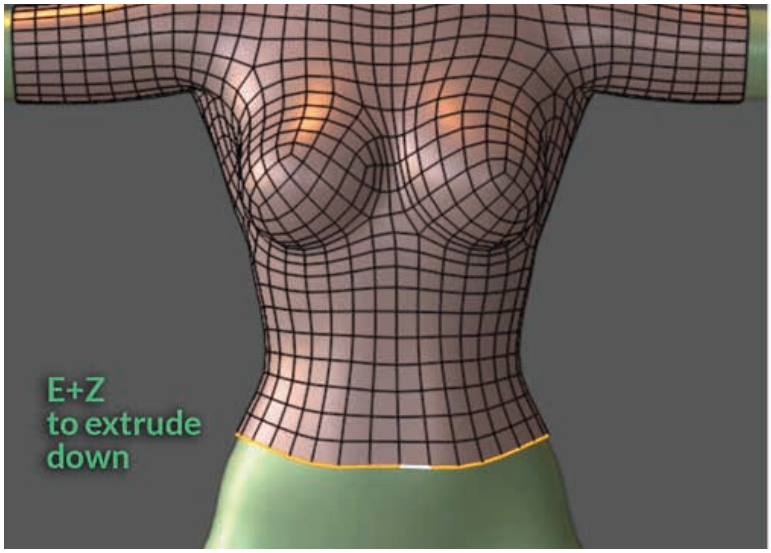


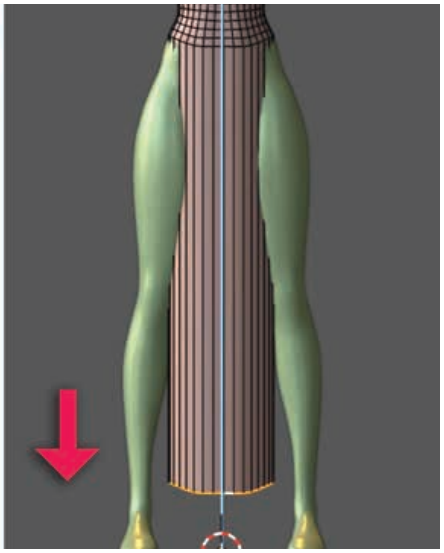
Duplicate it with **Shift+D**. Scale up this geometry along the normals with **Alt+S**. Then **press P** and separate by Selection.



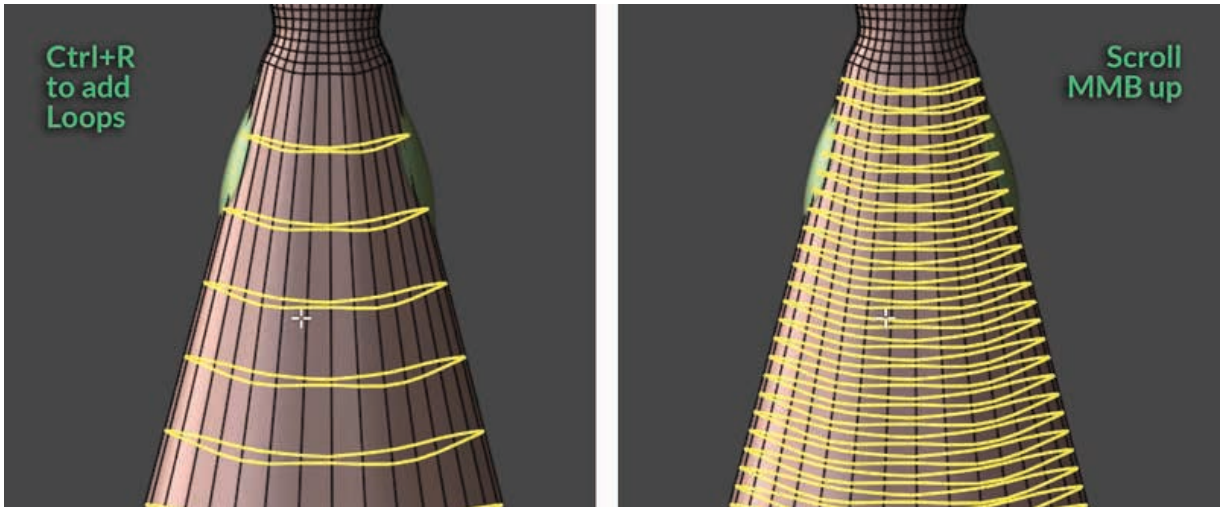


Select the bottom edge loop and extrude it down by **pressing E and Z**.

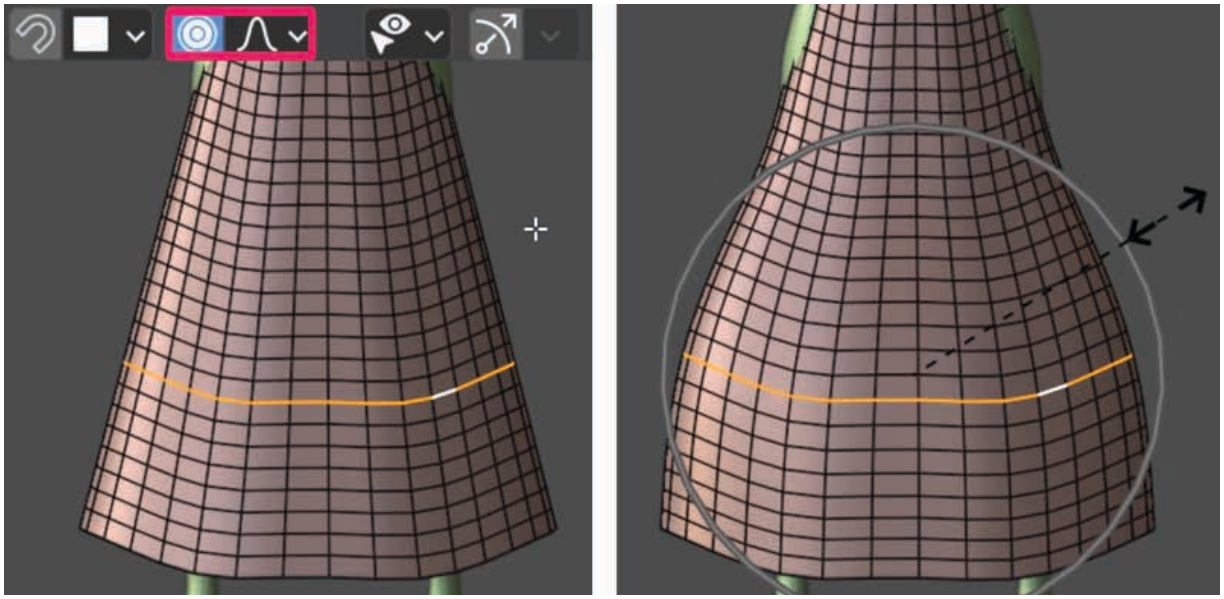




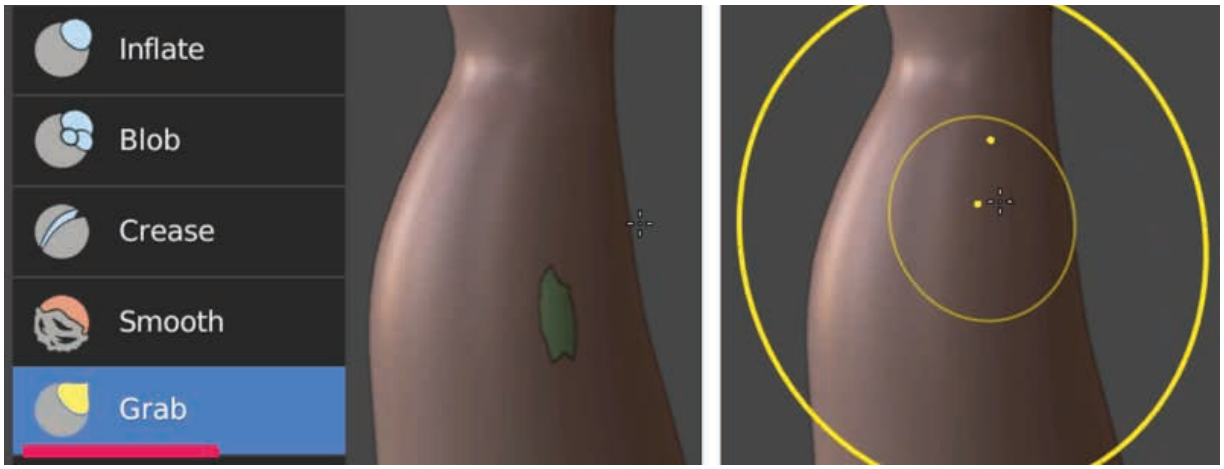
Press Ctrl+R and scroll the middle mouse wheel up to add loop cuts.



Select an edge loop and use **Proportional Editing** (shortcut: o) to scale up the bottom part of the dress as needed.

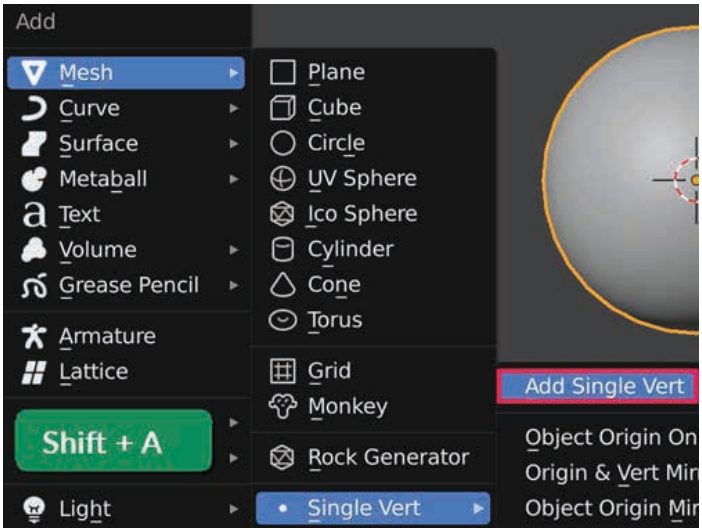


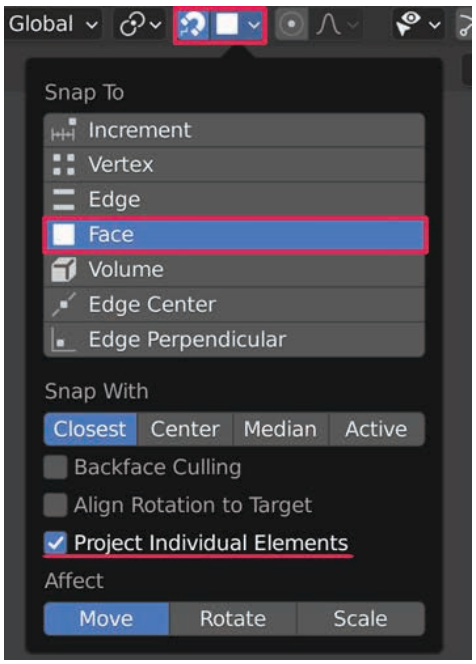
Finally in **Sculpt Mode** use the **Grab brush** and **Smooth brush** (shortcut: Shift) to adjust as needed.



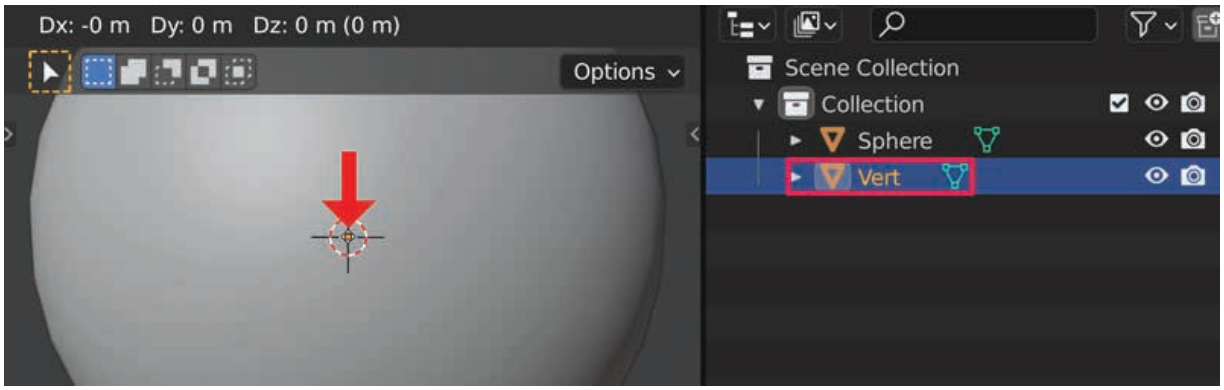
Modeling Stitches

Create a single vertex. To have this option, you need the Extra Objects add-on enabled. Turn on snapping with the settings on the right:

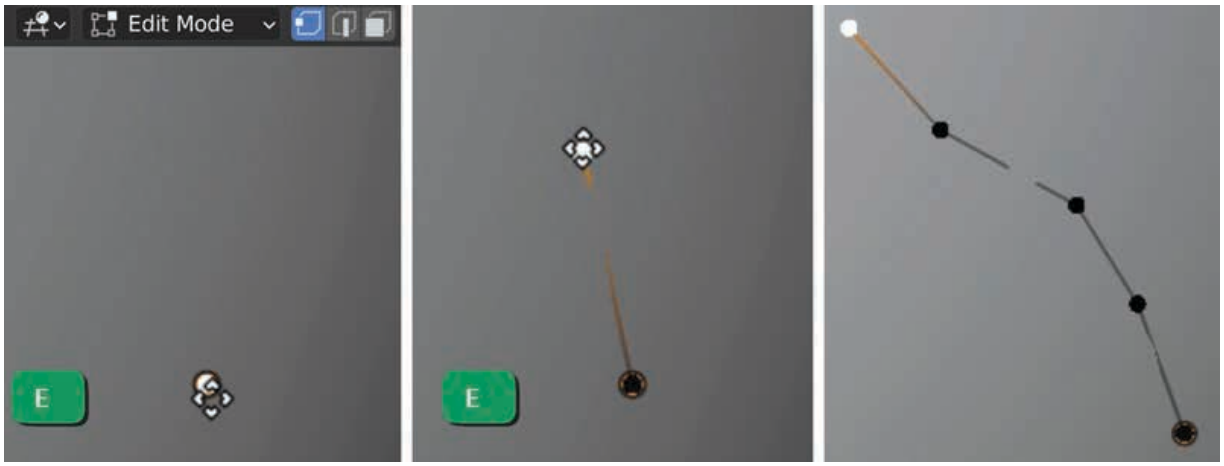




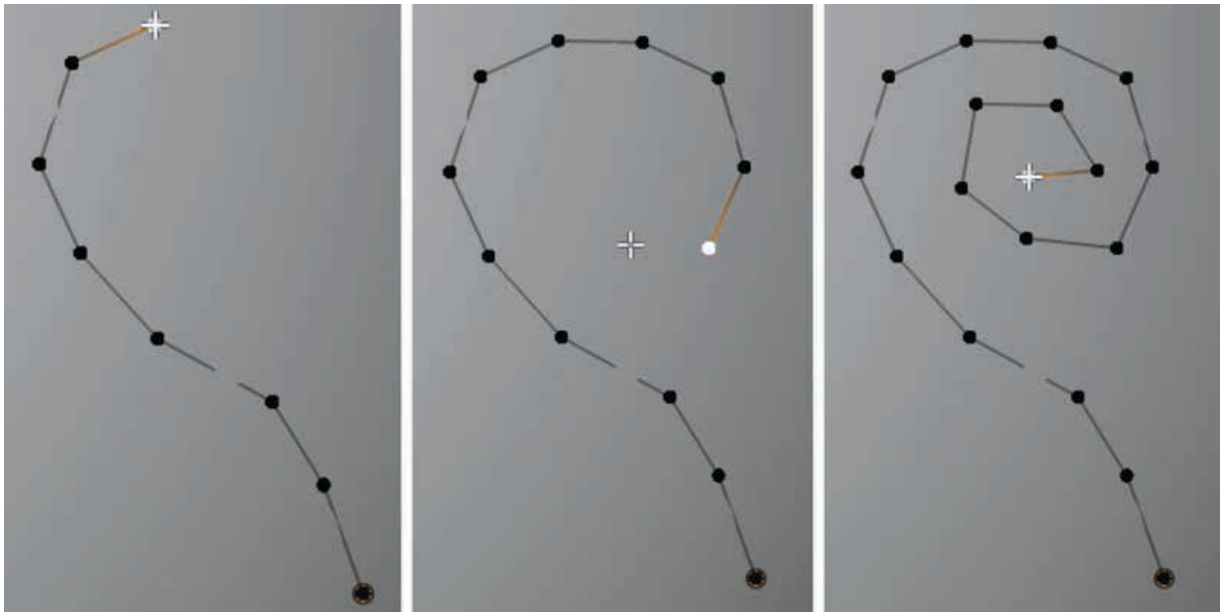
As you can see the Vertex snaps to the surface of the Sphere.



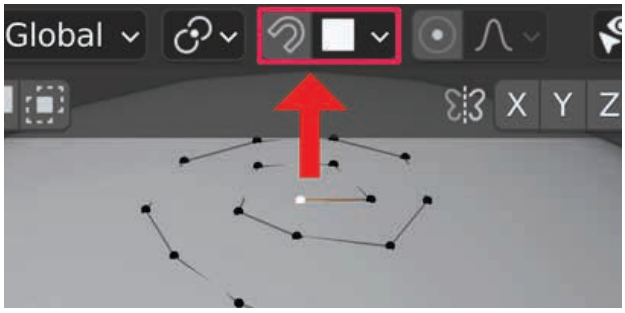
Extrude the vertex in Edit Mode by **pressing E** to create the path of the stitches. Take care to keep an even distance between the vertices.

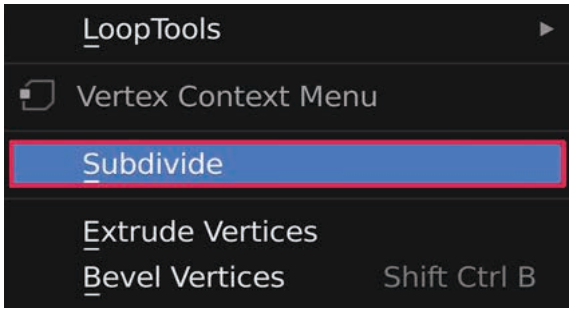


Extrude it in the path that you want the stitches to take, along the surface.

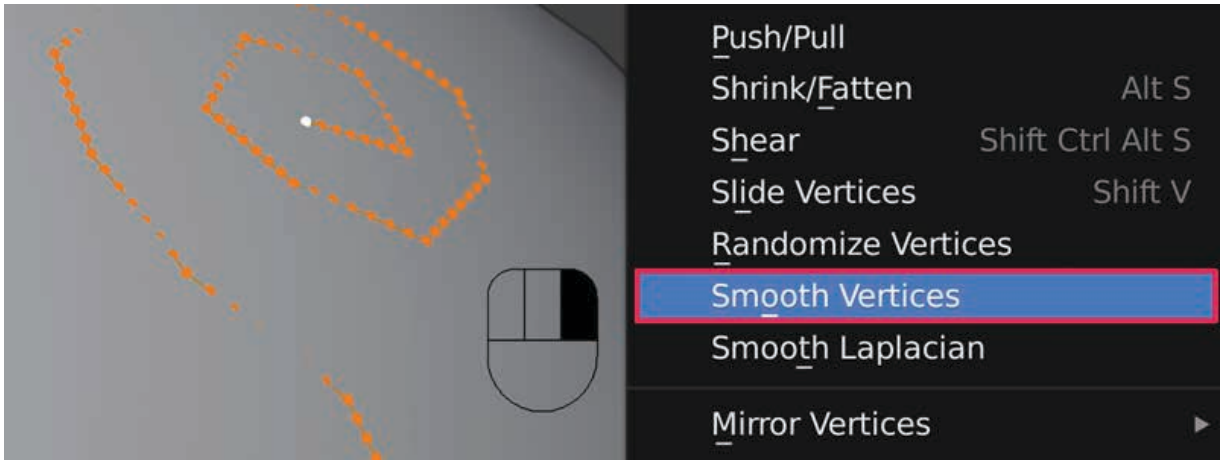


You can turn off Snapping now. Subdivide a couple of times to create more vertices by right-clicking in Edit Mode and choosing Subdivide.

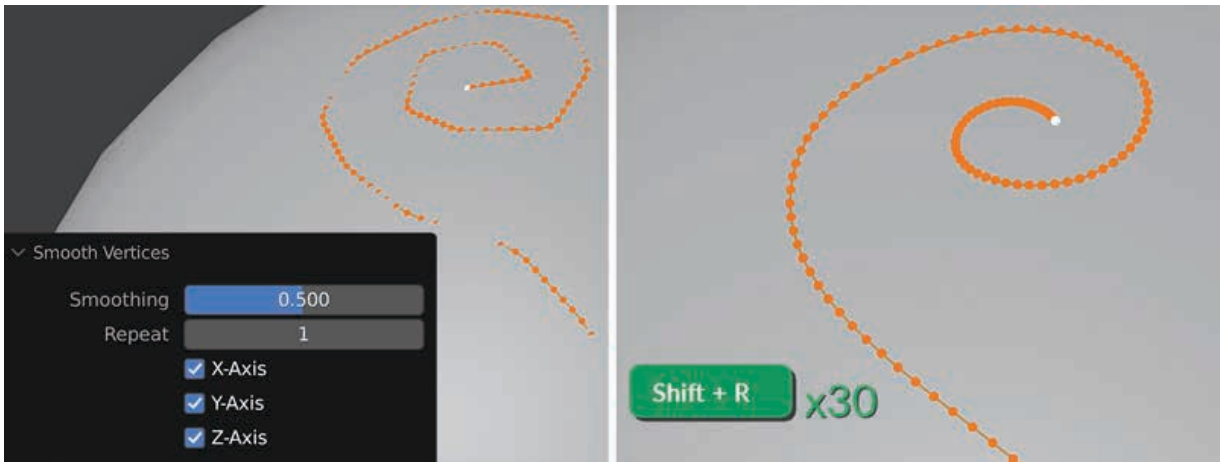




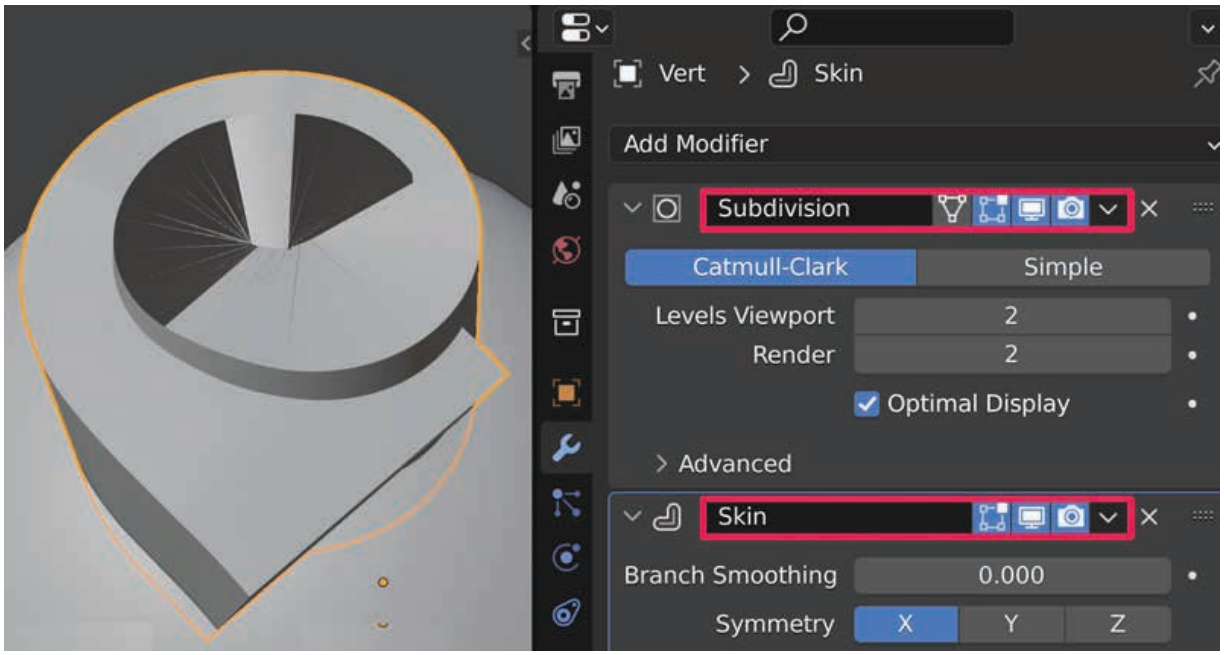
Right-click and choose **Smooth Vertices**.



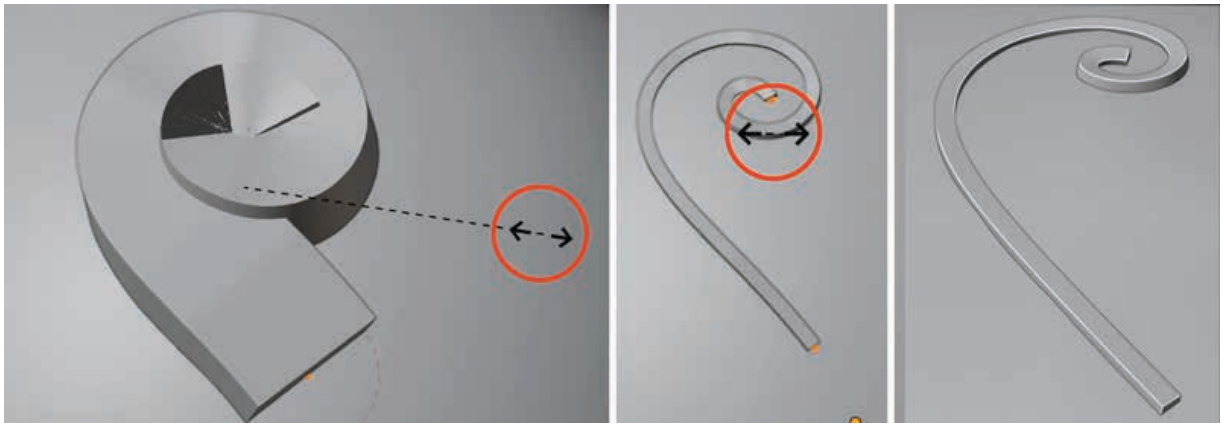
Press Shift+R a couple of times to repeat the **Smooth Vertices** action until it looks smooth.



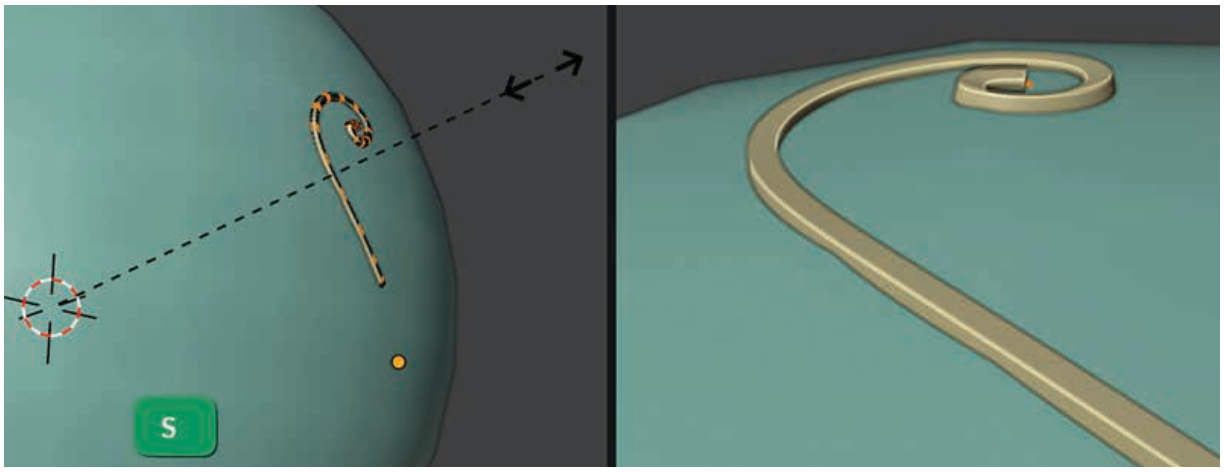
In Object Mode, add a **Subdivision Surface modifier** and then a **Skin modifier** to the stitches.



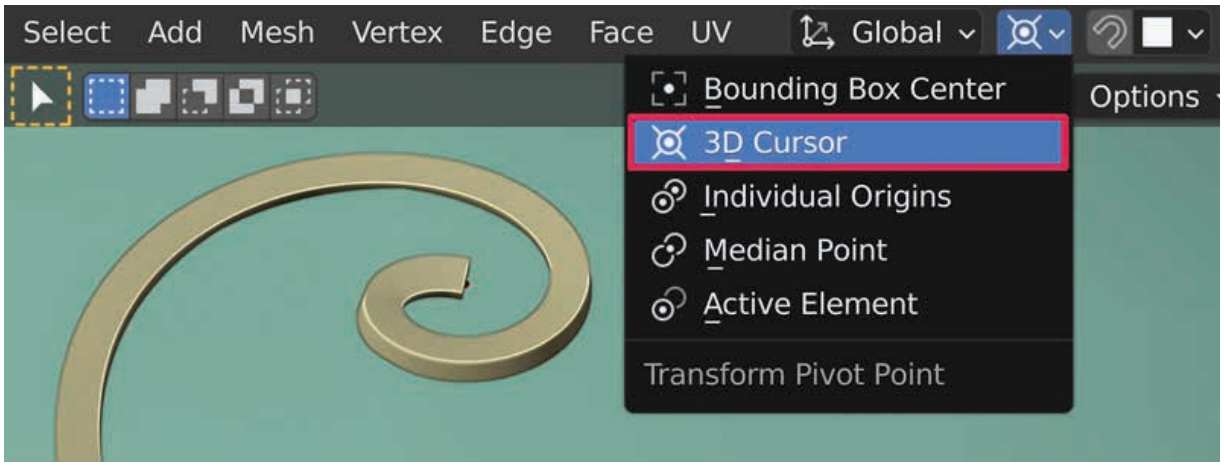
To scale the skin, in Edit Mode **press Ctrl+A** and move the mouse until it's smaller.



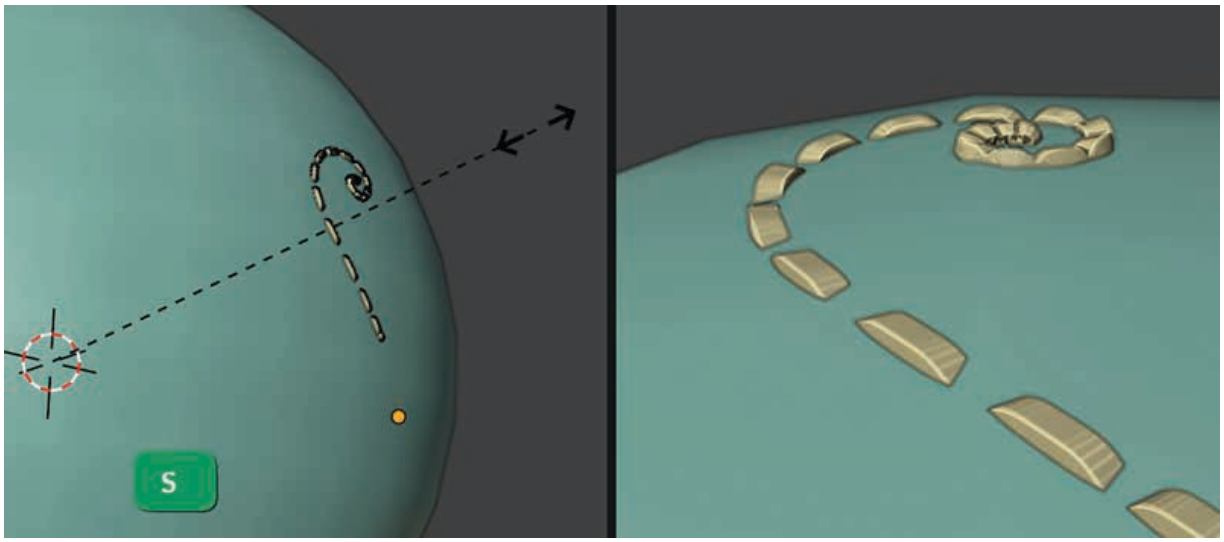
With all vertices selected, go to **Select > Checkered Deselect**. In the Checkered Deselect options choose for example 4 deselected, 2 selected.



Set the **Transform Pivot Point** to **3D cursor** and scale the selected vertices down...



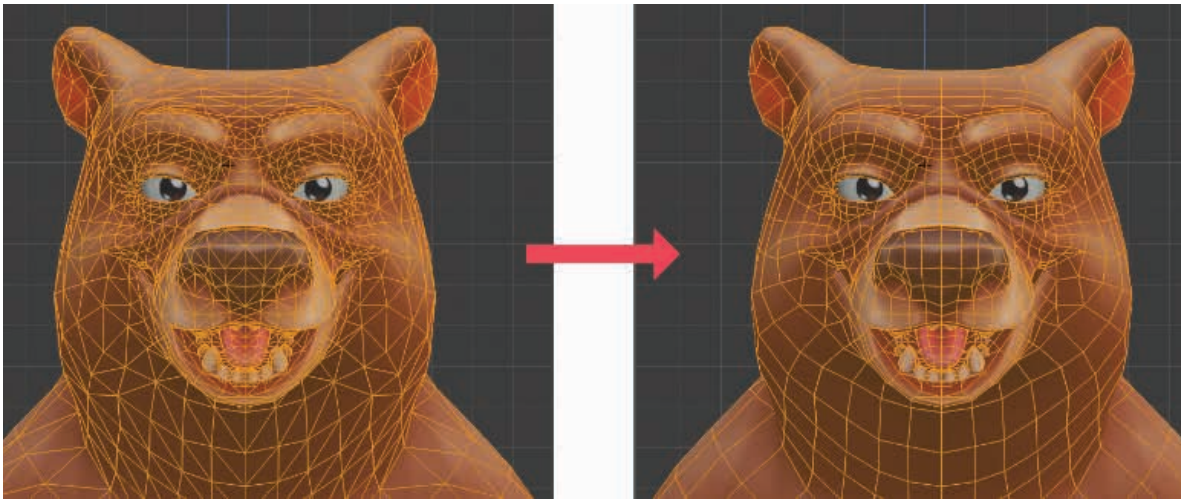
...until they are under the surface of the object.



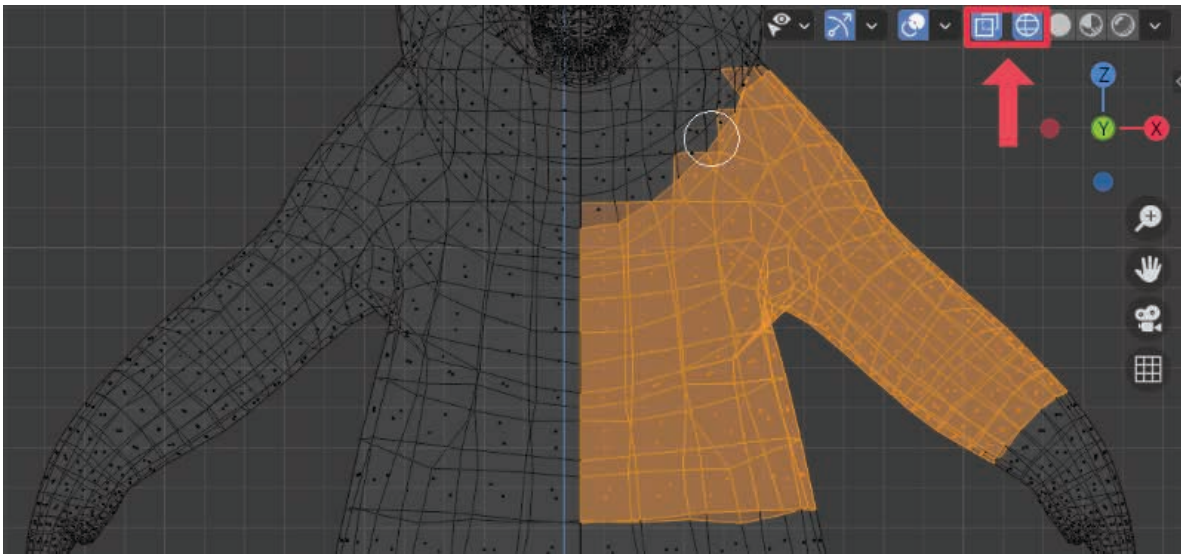
Modeling clothes

Here I've got this nice cartoon bear model created by Sketchfab user Jose Díaz. However I'd like to put some clothes on him to make the character a bit more unique. To do this, we'll use some simple modeling techniques. It's easier than you might think!

First of all lets convert the model topology from triangles to quads. **Press Tab** for Edit Mode, then select all by **pressing A** and turn the triangles into quads by **pressing Alt+J**. Models purchased online are often made up of triangles, but if they were originally modeled with quads, you can usually convert them back.

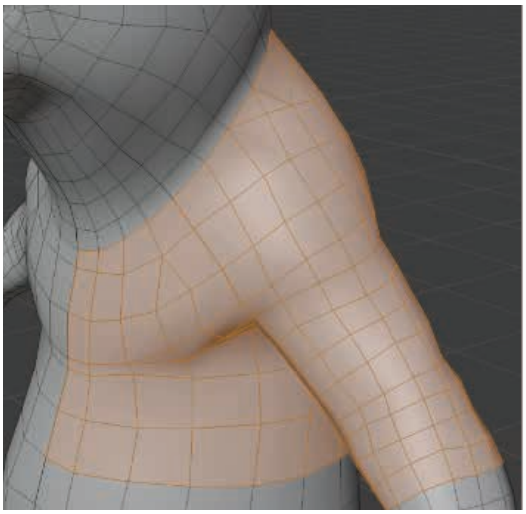


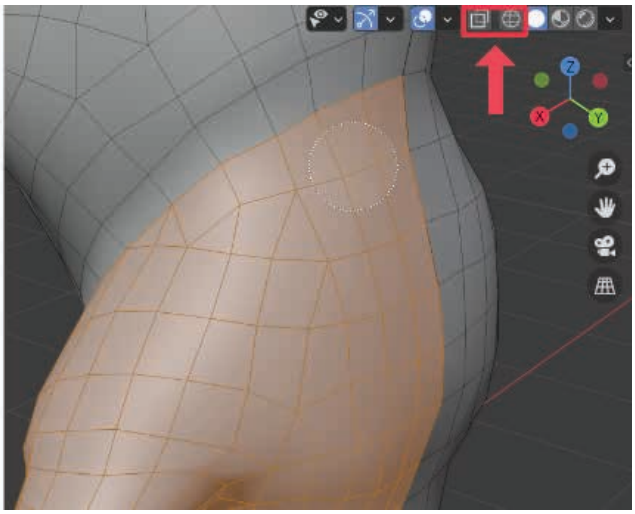
Now it's a bit more easy to view the faces, which is important for the next parts. Toggle on **Wireframe** and **X-ray** mode so you can see (and select) through the model. Then **press C** to activate **Circle select**. This turns your mouse cursor into a circle, which you can expand and contract with the mouse scroll wheel.



Left-click and drag the circle to quickly select faces in the shape of a shirt. You only need to do one side. **Right-click** to exit Circle select mode. Don't worry about getting it 100% perfect in one go.

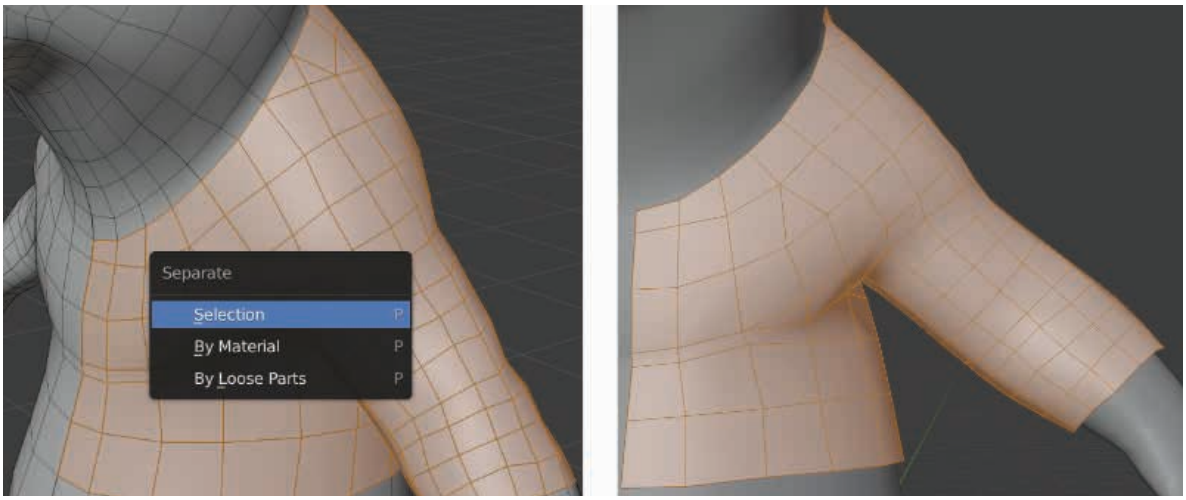
Toggle Wireframe and X-ray mode off to fine-tune the selection. You can "erase" from the Circle selection by **holding Shift**. Once you're done, **right-click to exit** Circle selection mode.



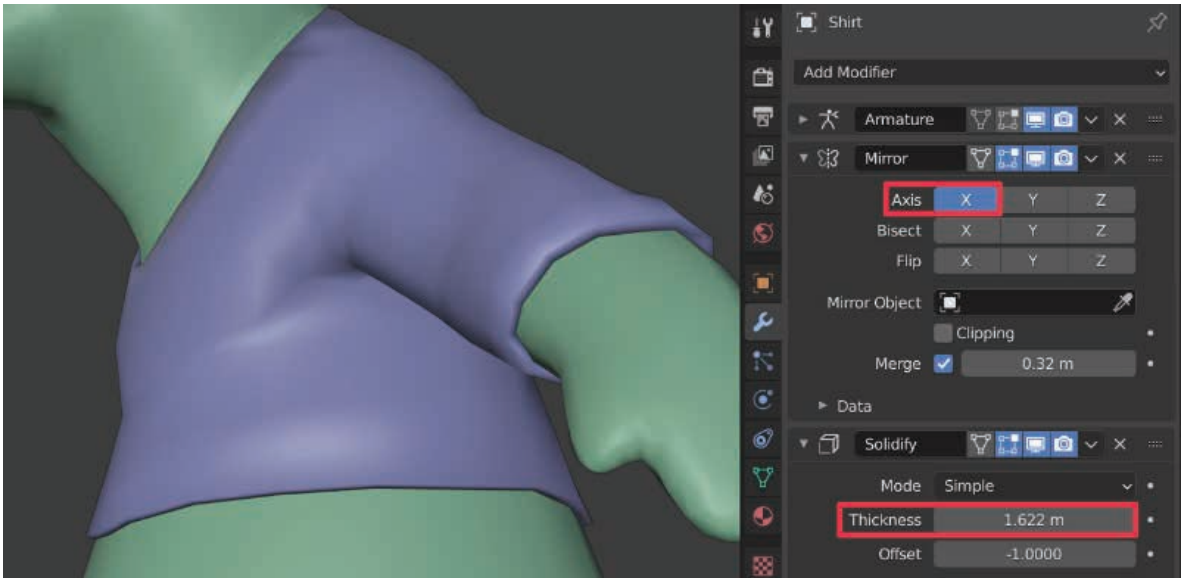


Duplicate the selection by **pressing Shift+D**. In order to separate the mesh from the character, **press P and choose "Selection"**. Exit Edit Mode by **pressing Tab** and select the clothing mesh to make sure we've only got that selected. **Press Tab** again to go back into Edit Mode and **press A** to make sure the entire clothing mesh is selected.

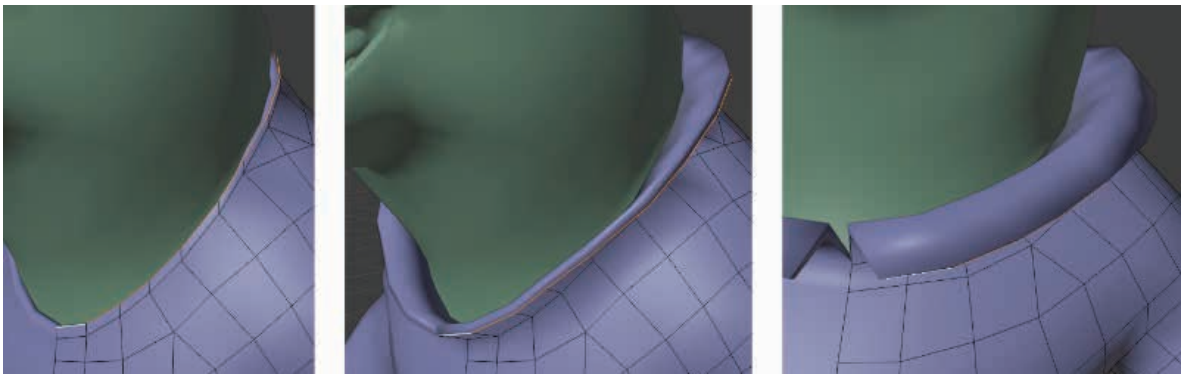
Then, we scale the clothing mesh up just a tad, using the keyboard shortcut **Alt+S**. Just so that it doesn't overlap the character geometry.



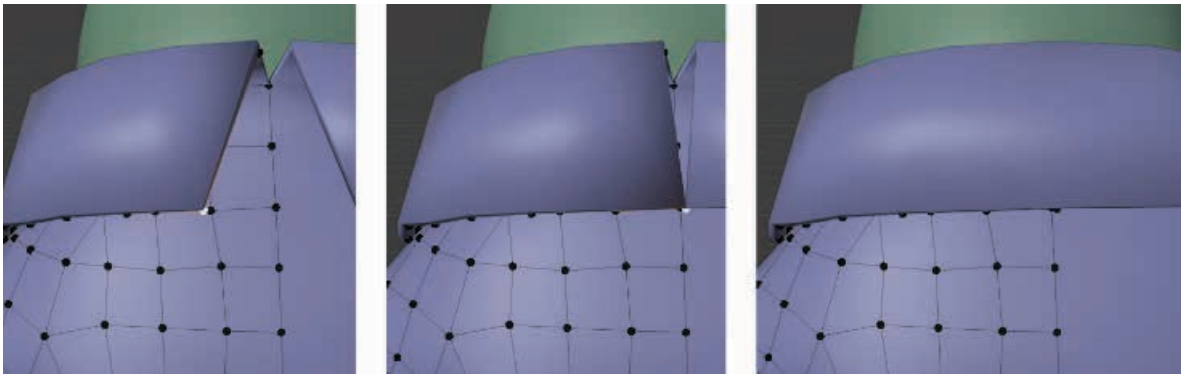
Let's have a look at the modifier tab in Object Mode. As you can see there was already an Armature modifier because I duplicated the mesh from an already rigged character. Add a **Mirror modifier** and set it to the correct axis (X in this case) and a **Solidify modifier** to add some Thickness to the model.



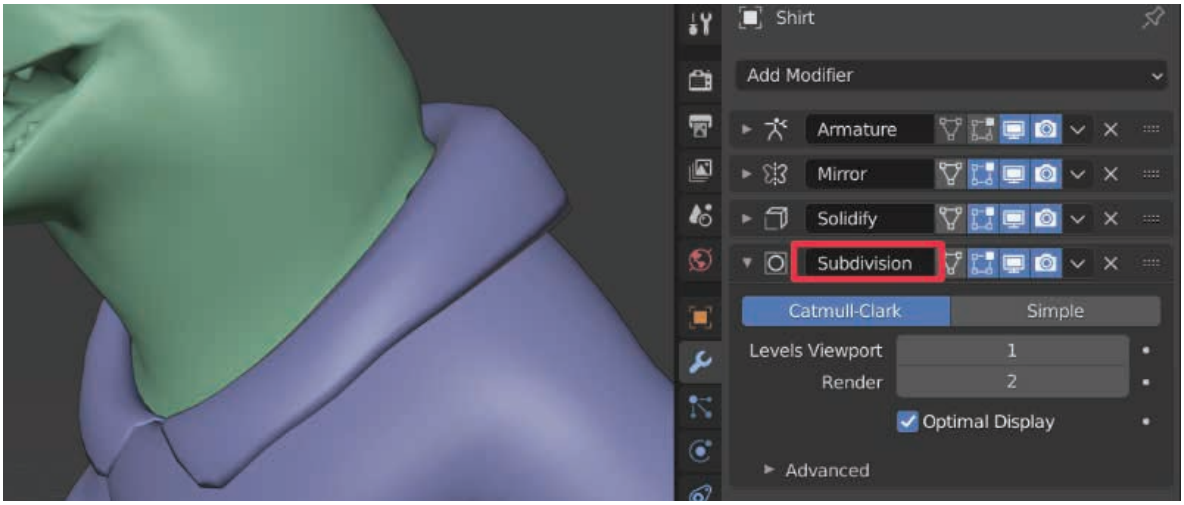
Let's add a collar to the shirt. Select the top row of edges and extrude them up by **pressing E**. Scale the extruded edge up a bit by **pressing S**. Then extrude once more, moving it down and scaling it appropriately. Look at it from different angles to see if you need to rotate or move it some more.



Move any vertices in the back that are too far from the center by **pressing G+X** until they get **merged** by the Mirror modifier.

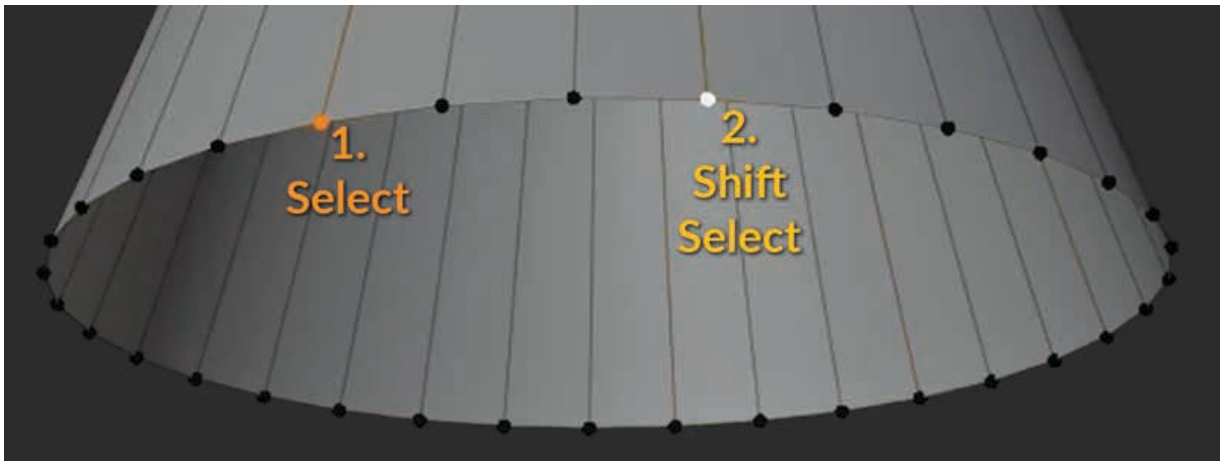


Add a Subdivision modifier to make it all a bit smoother.

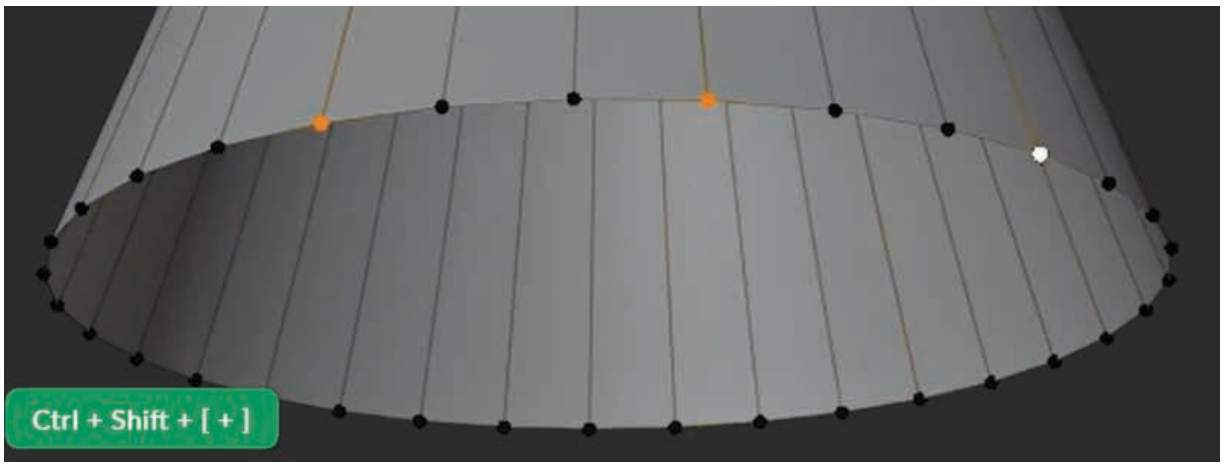


Model a Skirt using Select Next Active

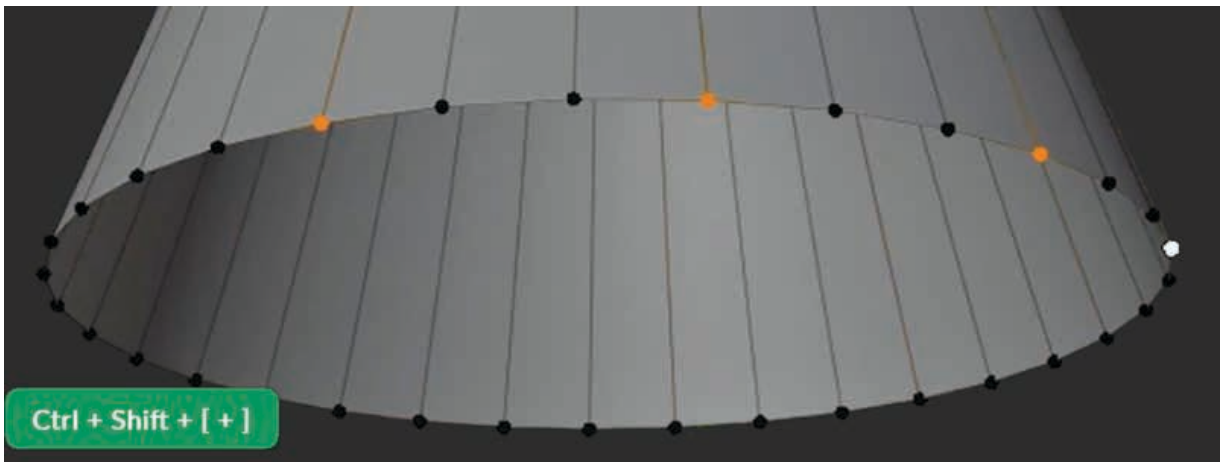
To repeat a selection where you skip a few vertices, edges or faces, select the two initial ones while **holding Shift...**



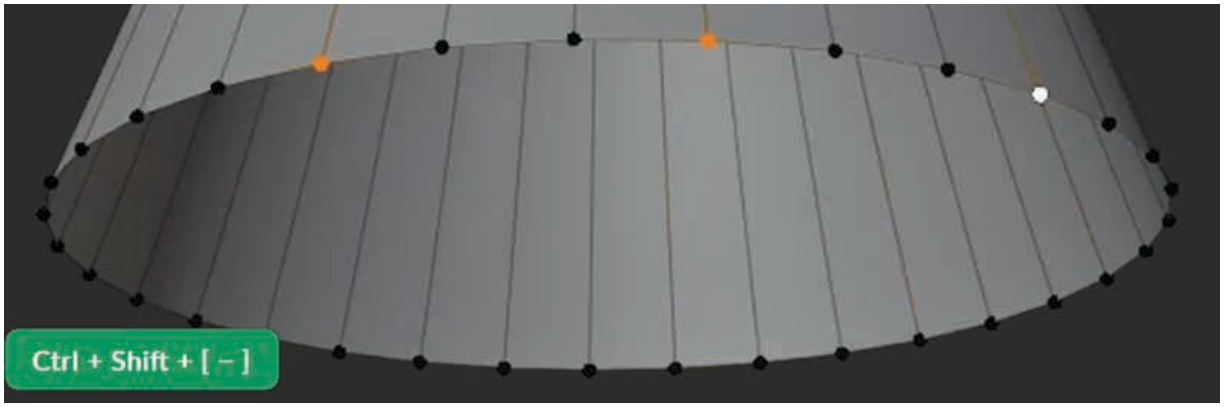
And then **press Shift+Ctrl+[numpad +]**.



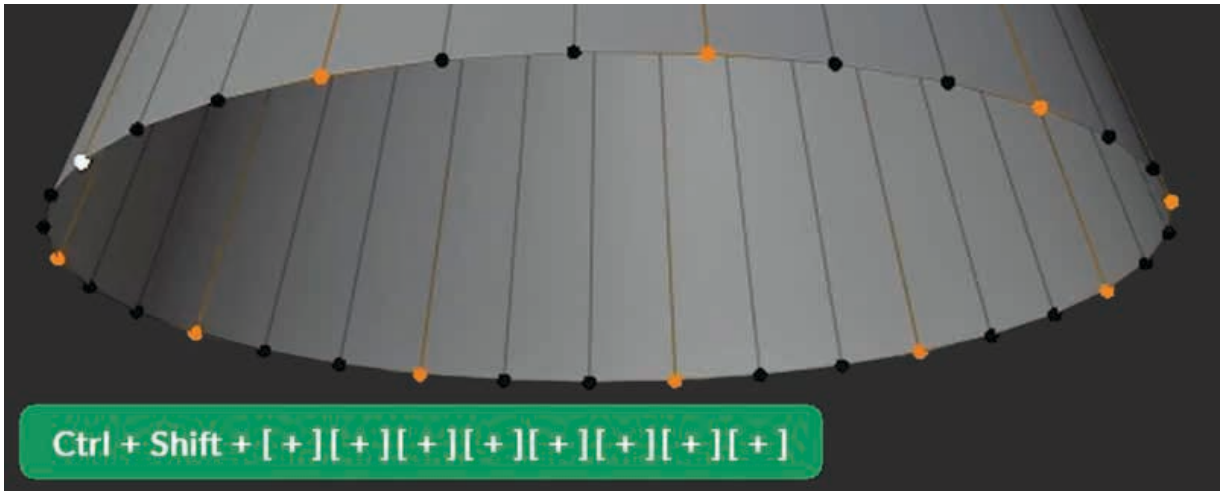
This will select the next vertex edge of face in that selection pattern.



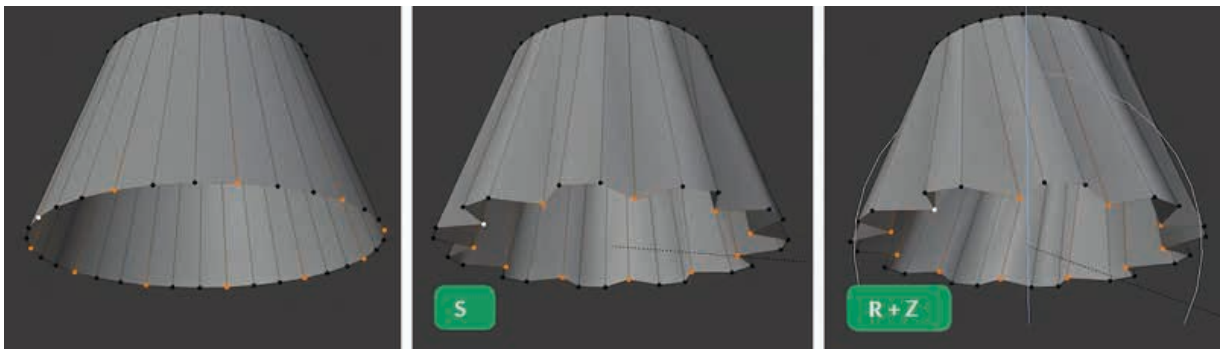
It also works in reverse, by **pressing Ctrl+Shift+[numpad -].**

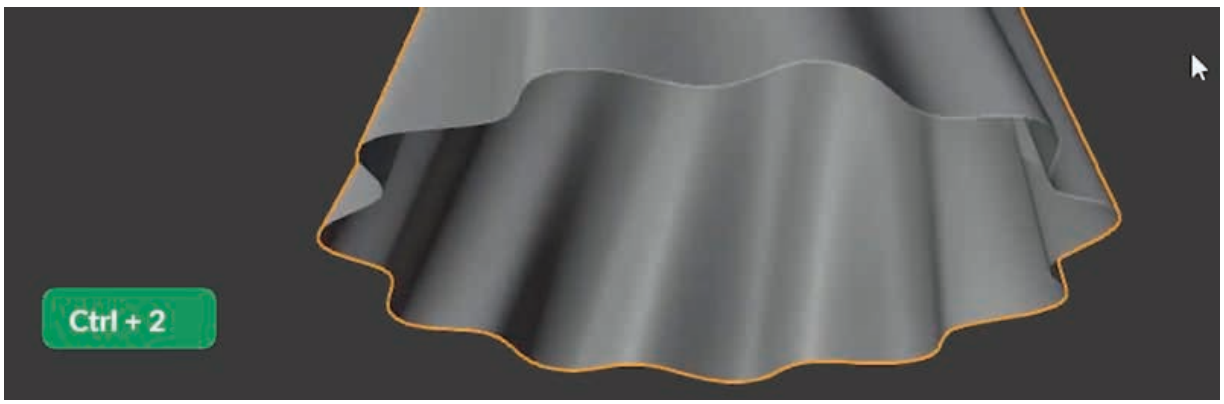


This way, we can select vertices while skipping two vertices each time, for example.



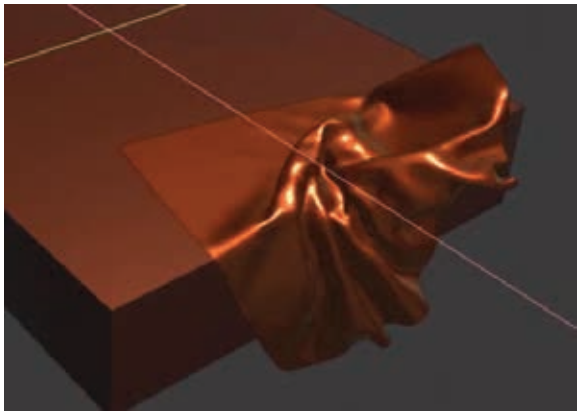
It comes in handy to model a skirt, for example.

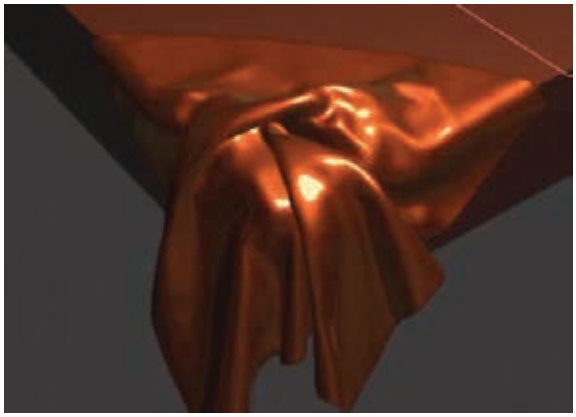




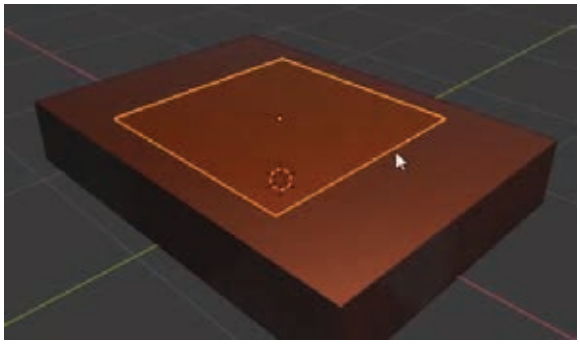
Interactively adjust cloth sims

In this tutorial we'll look at how to set up a cloth object so that you can drag it and drape it over another object.





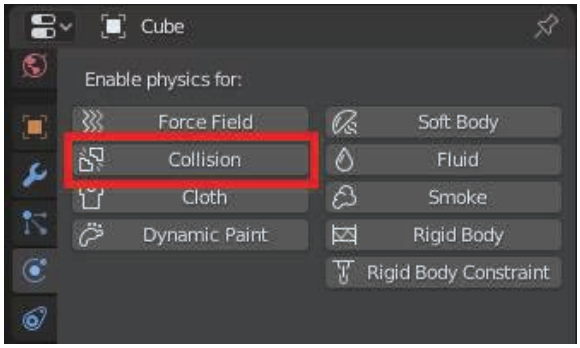
These are a subdivided plane and a box object.



Select the box.

Enable **Collision**.

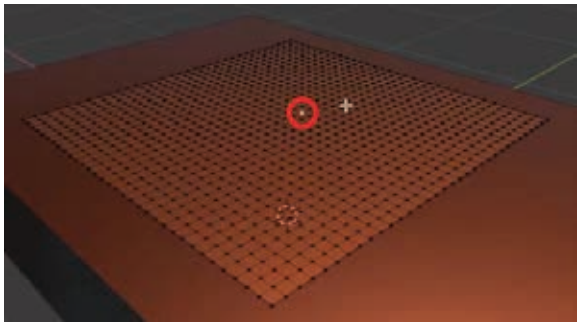
Now, any rigid body or cloth objects will know the box is there, instead of falling through it.



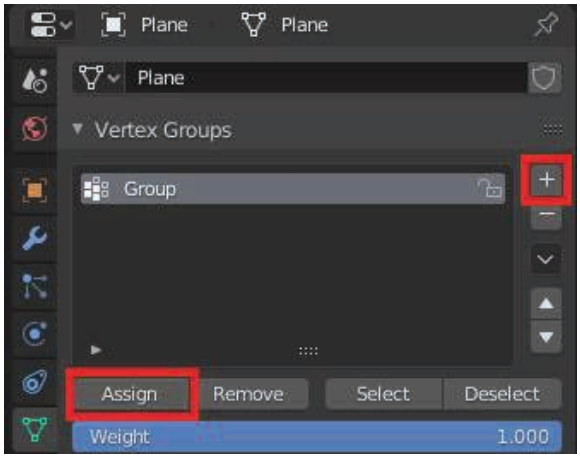
Select the plane.

Press TAB to go into Edit Mode. Subdivide the plane a few times (right-click, choose Subdivide).

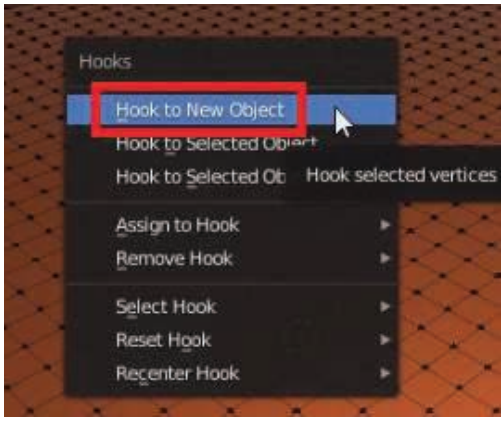
Select a vertex.



Assign the selected vertex to a new **Vertex Group** by clicking the **+ icon** and clicking **Assign**.

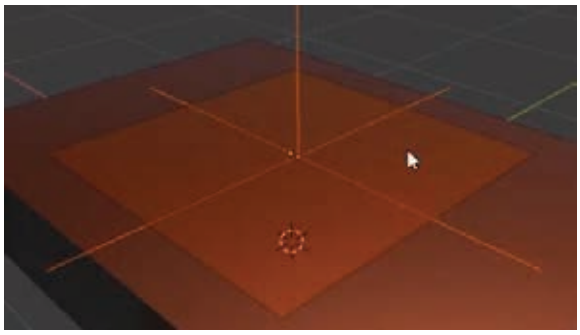


With that vertex still selected, **press Ctrl+H** to add a **Hook** to the vertex.

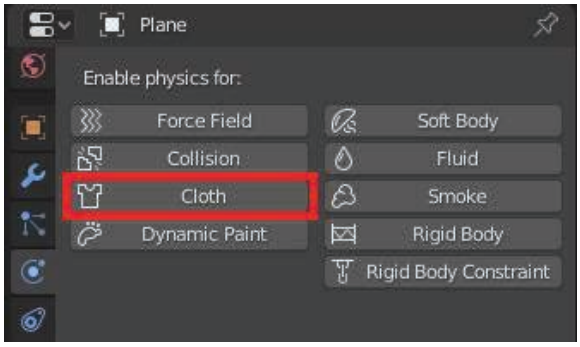


The Hook looks like an Empty.

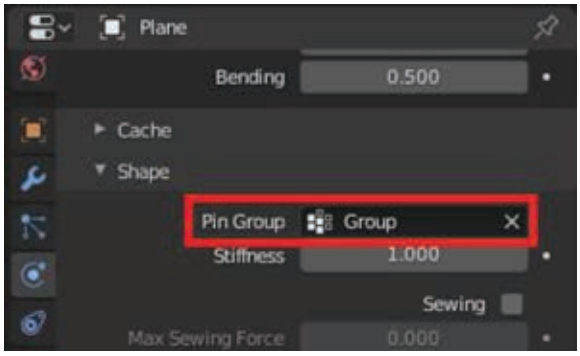
Press TAB to go back to Object Mode.

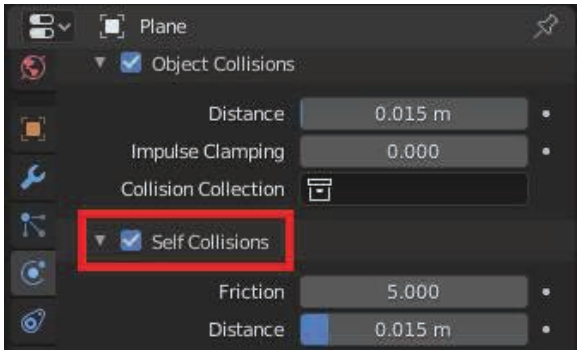


Enable **Cloth** physics for the plane.



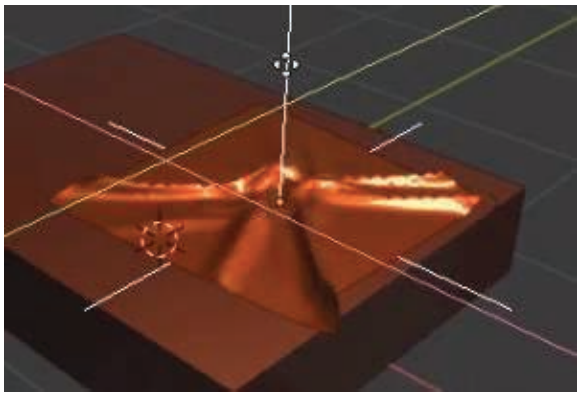
Use **Pinning** (choose the Vertex Group you made earlier) and enable **Self Collisions**.



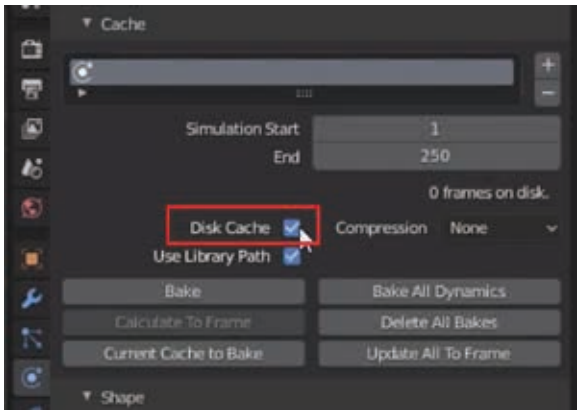


Start the simulation by pressing the **Spacebar**.

You can now move the cloth by moving the Hook.



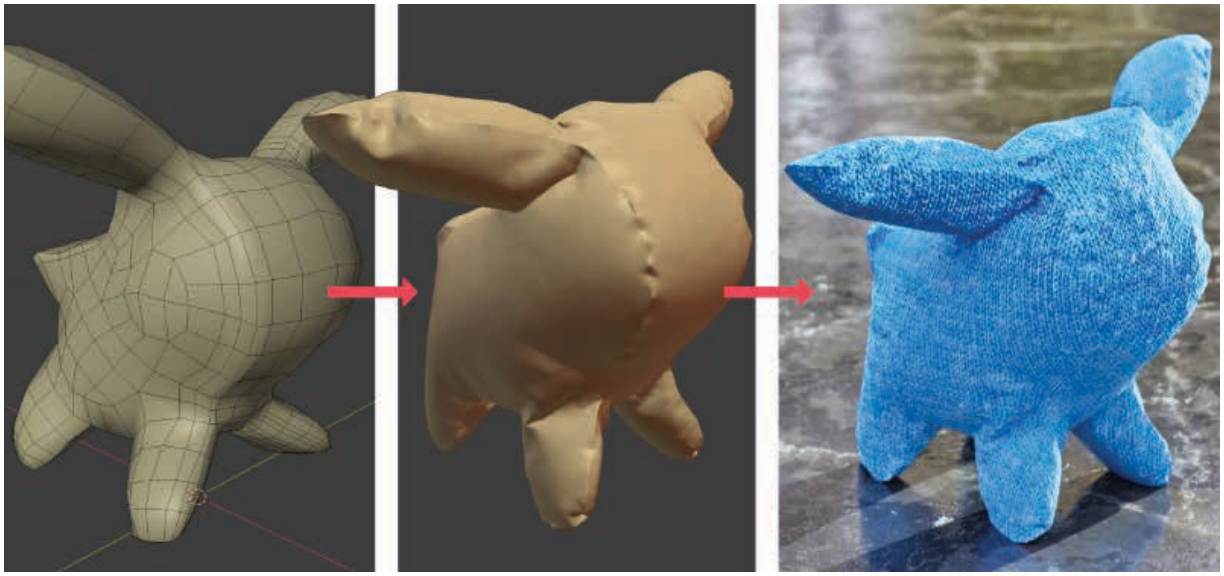
Bonus tip: if you can't delete the physics bake, enable/disable "disk cache", that helps.



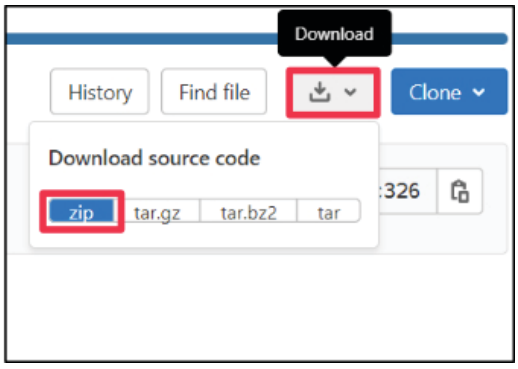
Seams to Sewing Pattern

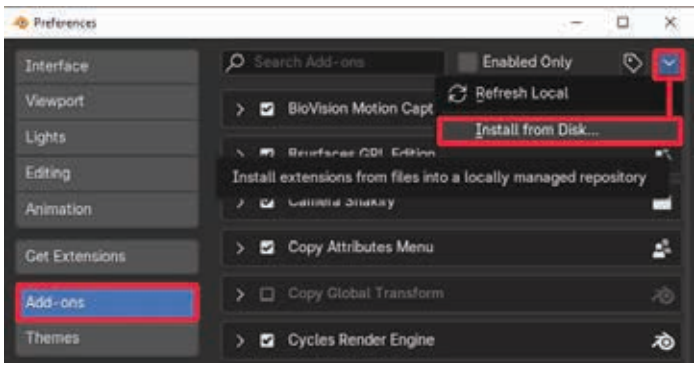
There have been some wonderful improvements to the Cloth simulation in Blender recently, my personal favorite being the **Pressure** option, which inflates objects and gives them a very realistic puffy look.

The free add-on **Seams to Sewing Pattern** makes it even easy to turn your 3D model into a puffy, pillow-like object.



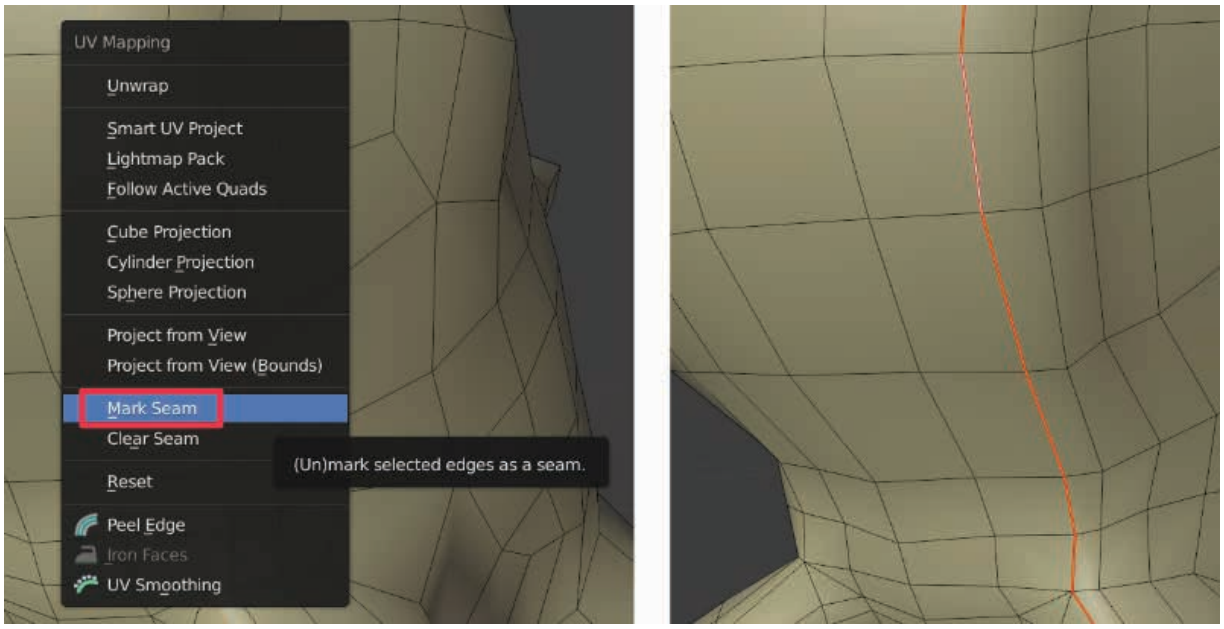
To use it, first you'll need to **download the zip file** from Github and install it from Preferences. You can also check out the creator Thomas Kole's website, a nicely designed page that contains examples on how to use it.





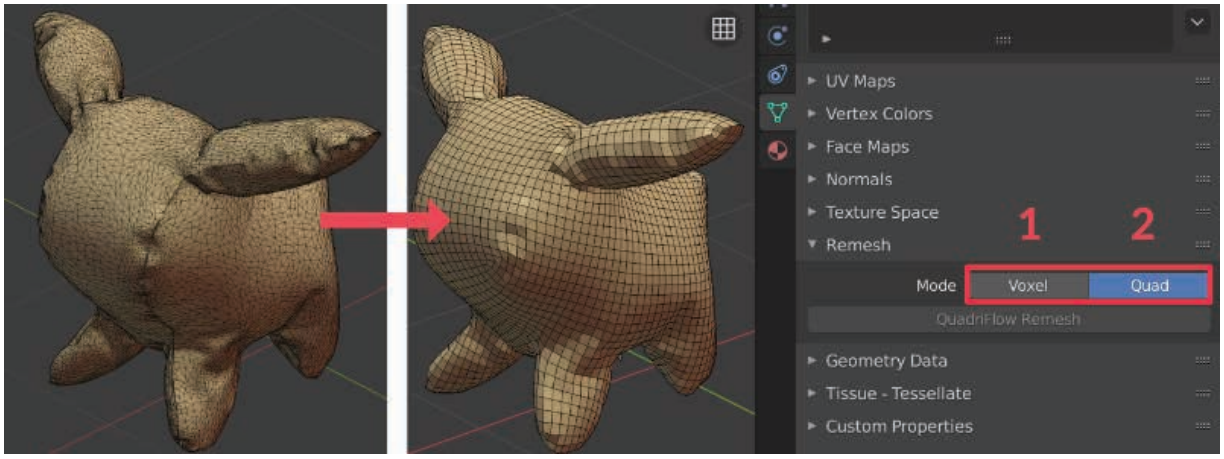
Then, after creating your model, create some seams like you would with UV-unwrapping. This is done the usual way, by selecting edges and **right-clicking**, then choosing **Mark Seam**.

Alternatively, **press U then M** to mark a seam. The seams will be where the mesh will be "cut" as if you are making a real-life object from cloth. Remember that the seams will be visible in the end result.

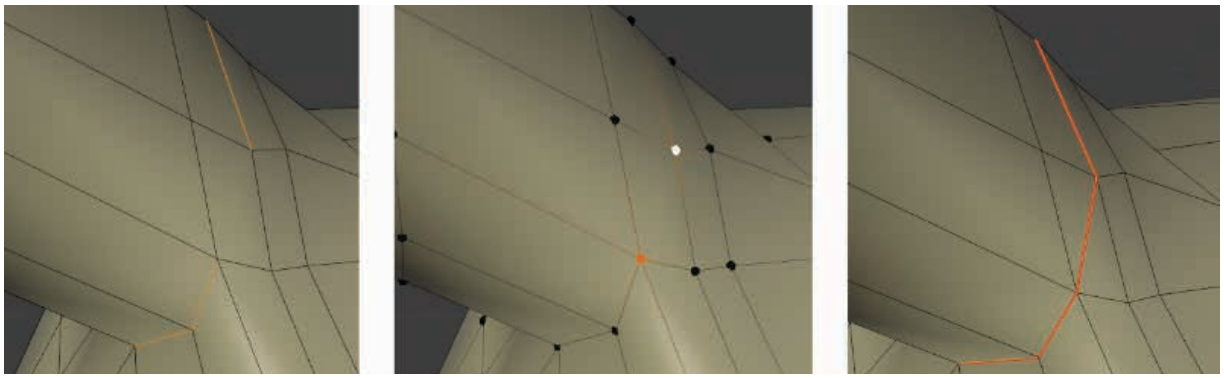


If it's difficult to create good seams because of bad topology, or if you think the topology is pretty bad in general (for example with a 3D scanned object), consider remeshing it first.

You can remesh objects either with the **Remesh modifier** or from the Object Data Properties tab. In my experience doing a Voxel remesh followed by a **Quad remesh** usually gives the best result. A manual retopology is even better if you have the time to do it.

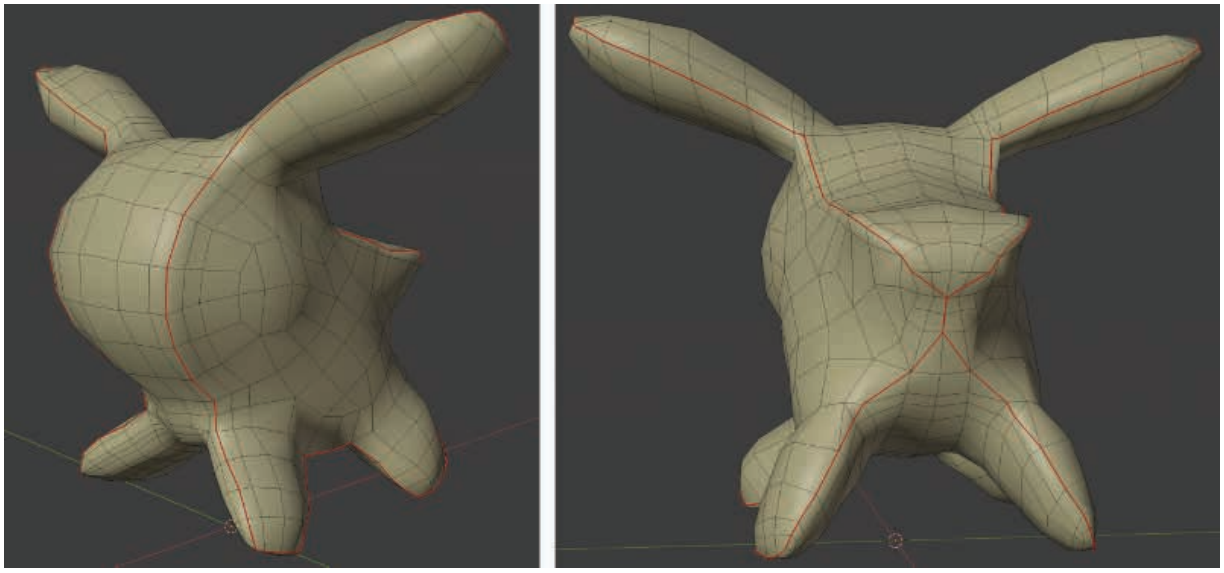


To make a nice loop cut in cases where the mesh doesn't flow well, you can use the **Knife Tool (K)** or select two vertices and connect them by **pressing J**. After that, mark the edges as a seam.



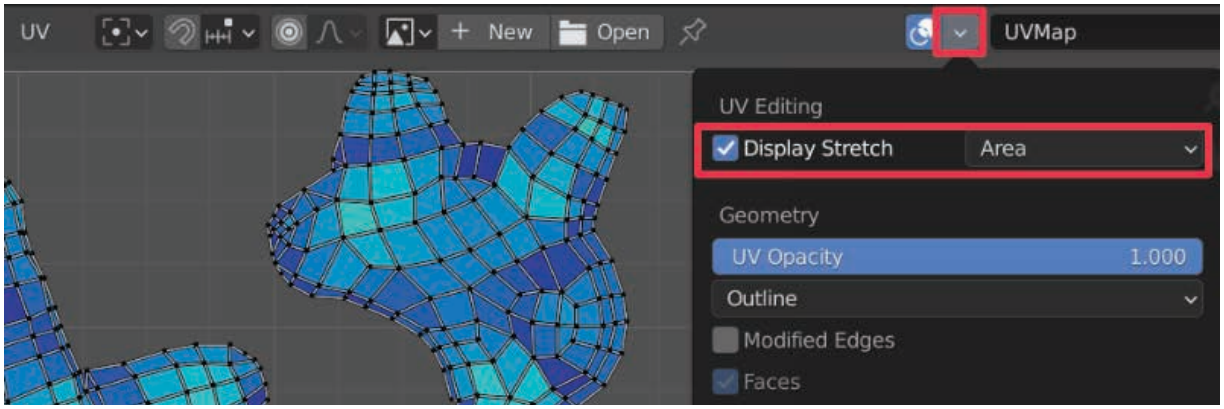
Creating seams for a sewing pattern is similar to creating seams for a UV map, but you don't need to worry too much about hiding the seams.

In fact, you'll see the seams anyway in the final result, so think about placing them in an interesting place. Here you can see how I placed them on my bunny model.



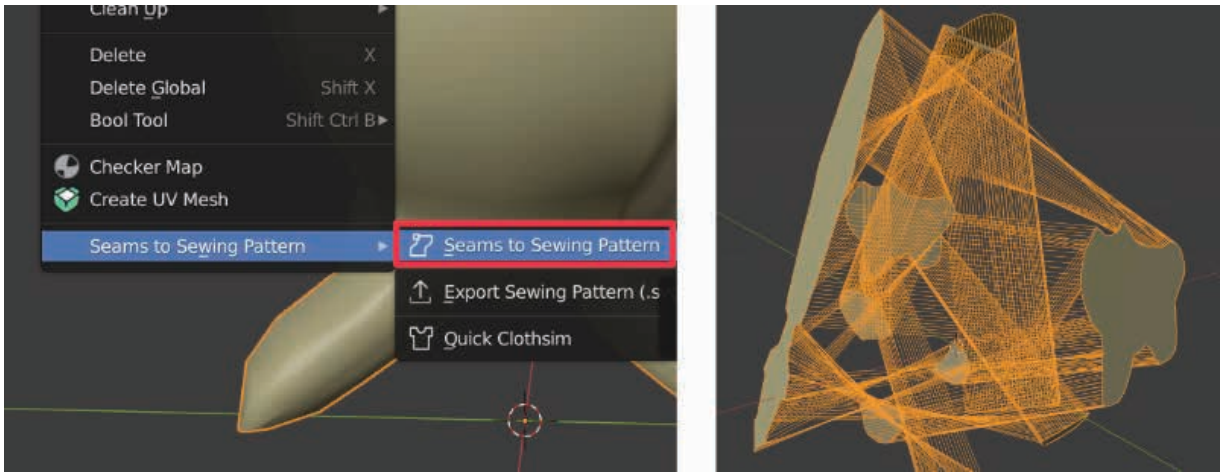
Select all in Edit Mode, and **press U twice** to unwrap the model. You can use the **Display Stretch** function in the UV Editor to see which areas are getting stretched too much in the UV map.

Blue is good, green is still okay but yellow and orange areas are where there aren't enough seams yet. You can improve this by adding more seams and unwrapping again.



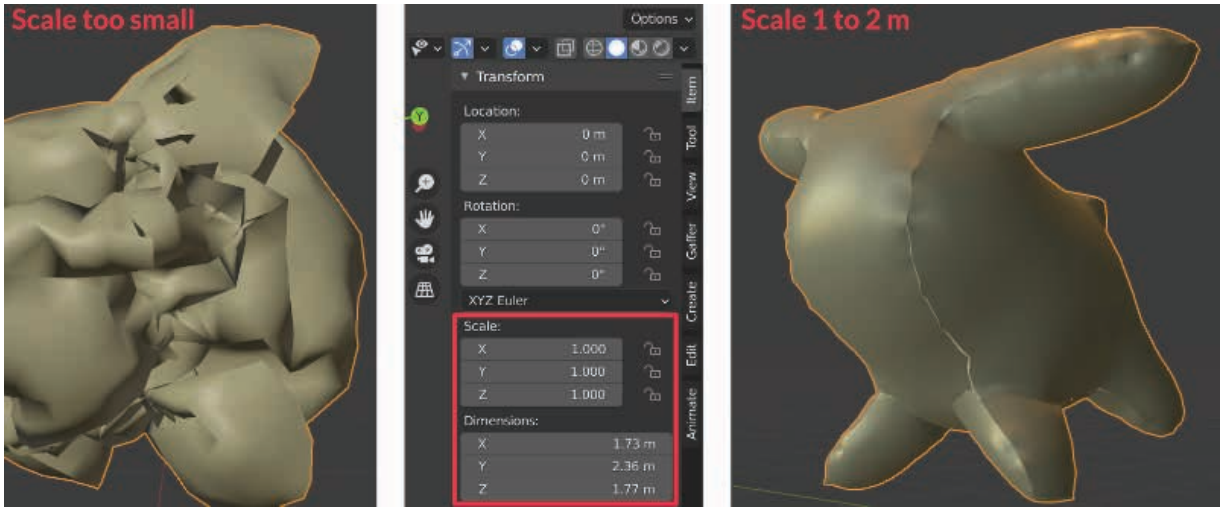
After checking the stretching, I added some more seams, to separate the ears and legs of the bunny completely. Go to Object > **Seams to Sewing Pattern** > Seams to Sewing Pattern and **click OK**.

Now it will look quite strange! The lines you see between the mesh islands are sewing lines.



Go to Object > Seams to Sewing Pattern > **Quick Clothsim**. Then to run the simulation **Press the Spacebar.** Now the object will come together. If your model comes together as a tangled ball of polygons, undo a few steps and then increase the object scale to about 1 or 2 meters by **pressing S** and scaling it up.

That big scale is not realistic for this little bunny, but in my experience the Cloth sim works better that way. Don't forget to **press Ctrl+A** and choose **Apply Scale** after scaling it up. Then run the Seams to Sewing Pattern and Quick Clothsim functions again.



After this step I like to add and apply a Subdiv modifier and remesh the model again.

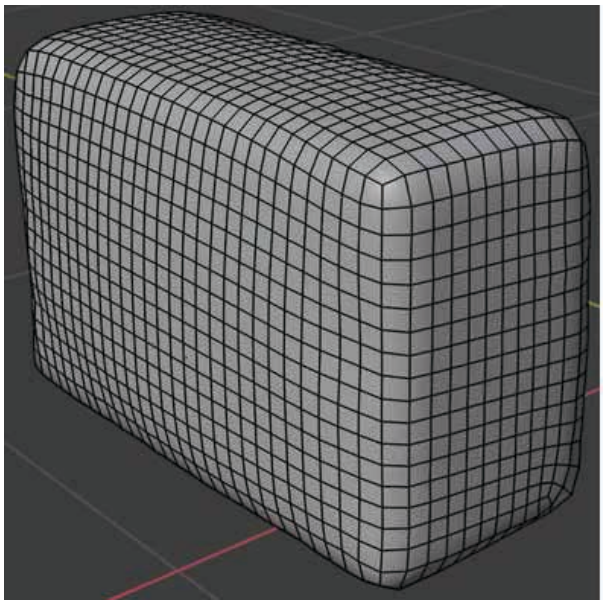
Alternatively, if you want you can export this as a real sewing pattern, by going to Object > Seams to Sewing Pattern > **Export Sewing Pattern to SVG**. After that, you can make real bunnies if you have a sewing machine. Just be careful with your fingers!

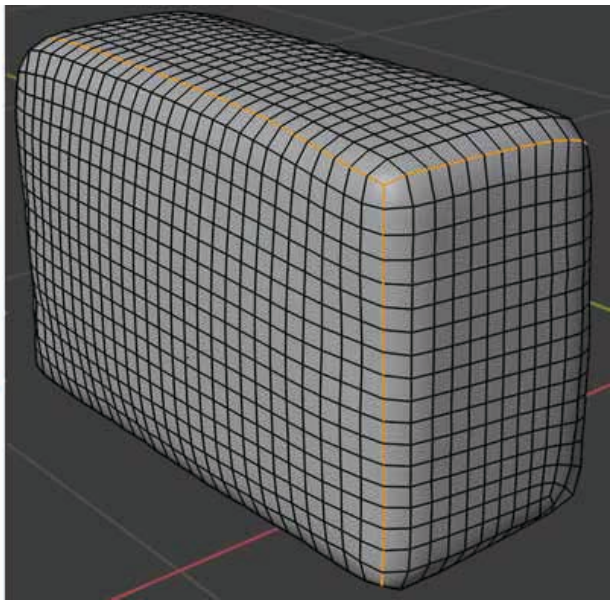
ARCHITECTURE MODELING

Create Decorative Edges on sofas and cushions

When modeling a sofa, at some point you need to create the decorative edges of the cushions and arm rests. These types of decorative edges are called “welded edges”.

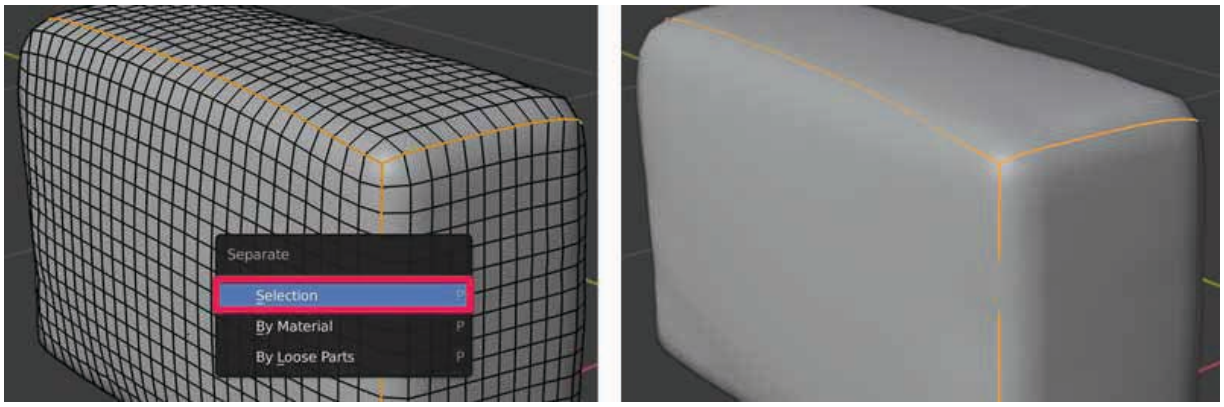
Let’s have a look at an easy way to create these. In case the topology of the sofa or cushions is relatively simple and quad-based, it’s easy: First, select the edges where you want to have these seams in Edit Mode.



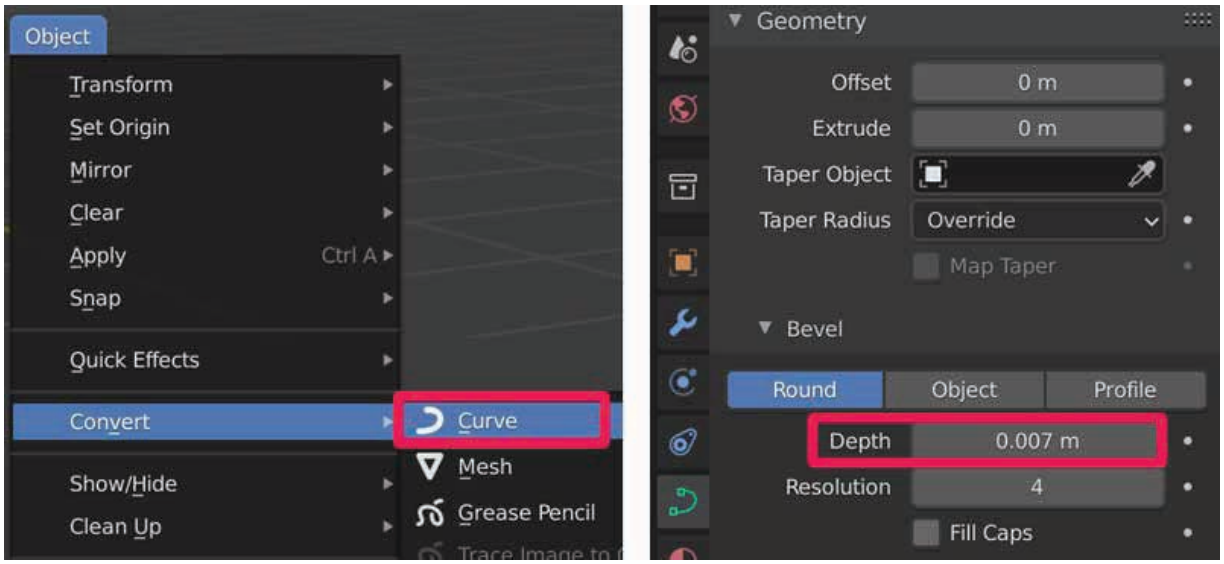


Then, **press Shift+D** to duplicate the selection. **Left-click** to confirm. With the duplicates still selected, **press P** and choose **Separate by Selection**. This turns the duplicated edges into a new object.

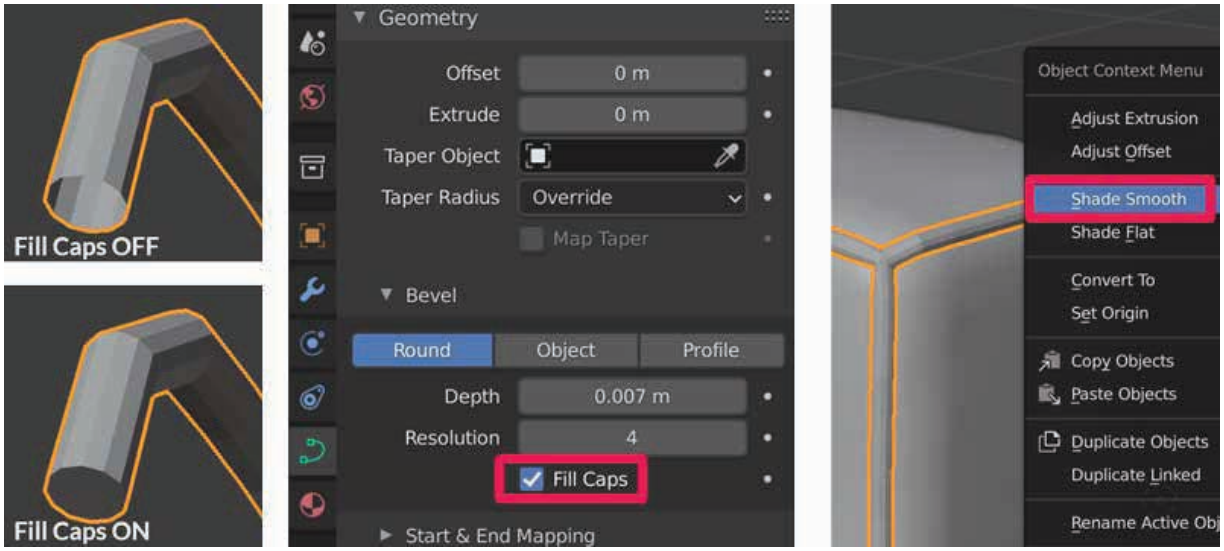
Don’t worry if the new object consists partly of Edges, and partly of Faces. **Press Tab** to exit Edit Mode and select only the duplicated object (if it’s difficult in the 3D viewport, select it in the Outliner).



Go to **Object > Convert > Curve**. This turns the edges into a Bezier Curve object. Now, you can go to the Geometry section of the Curve options, open the Bevel part and add some thickness to the curve by increasing the **Depth** value.



Check the **Fill Caps** option to make sure the ends are closed. In the 3D viewport, right-click on the object and choose **Shade Smooth**.



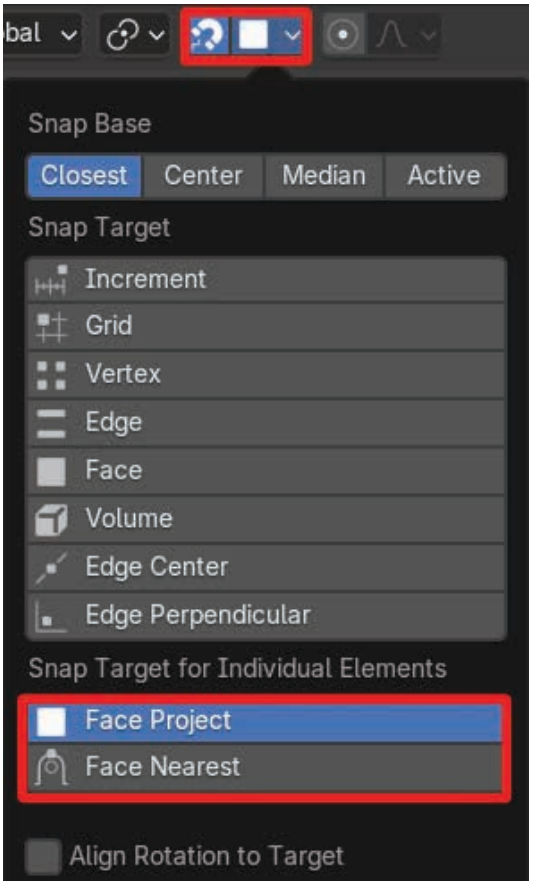
Your sofa or cushion may be very high poly or may have geometry that is difficult to select a clean edge from, like 3D scanned or heavily triangulated models.

This is often the case with furniture models downloaded from the web.

In that case you'll need a slightly different approach.

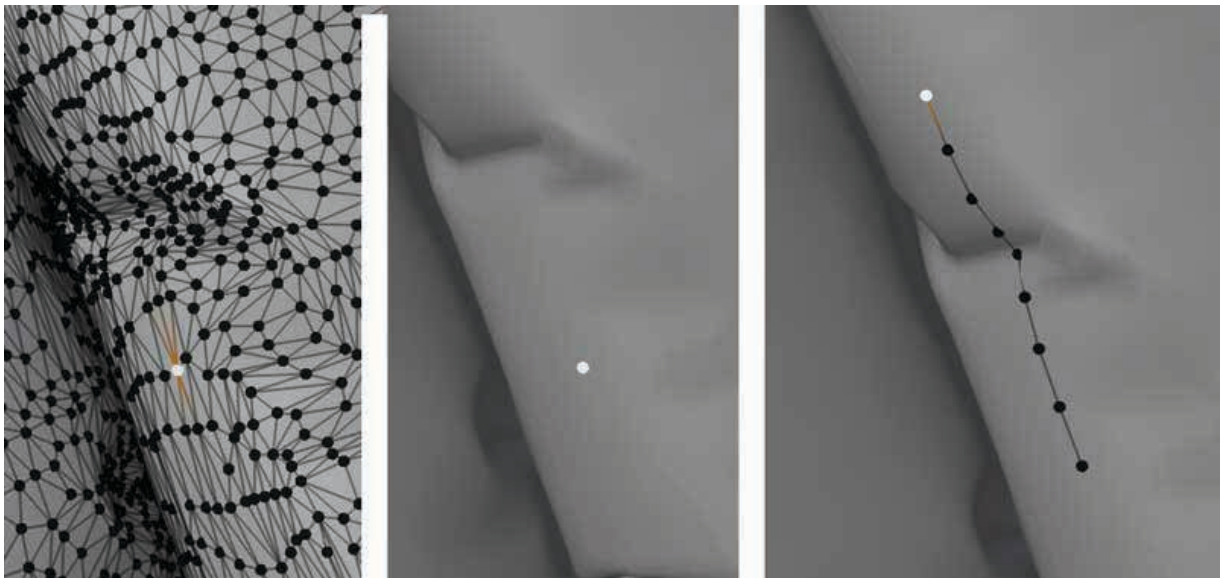
First, turn on **Snapping** and enable either **Face Project** or **Face Nearest**.

Face Nearest is a new option, and can sometimes produce better results (depending on your model).

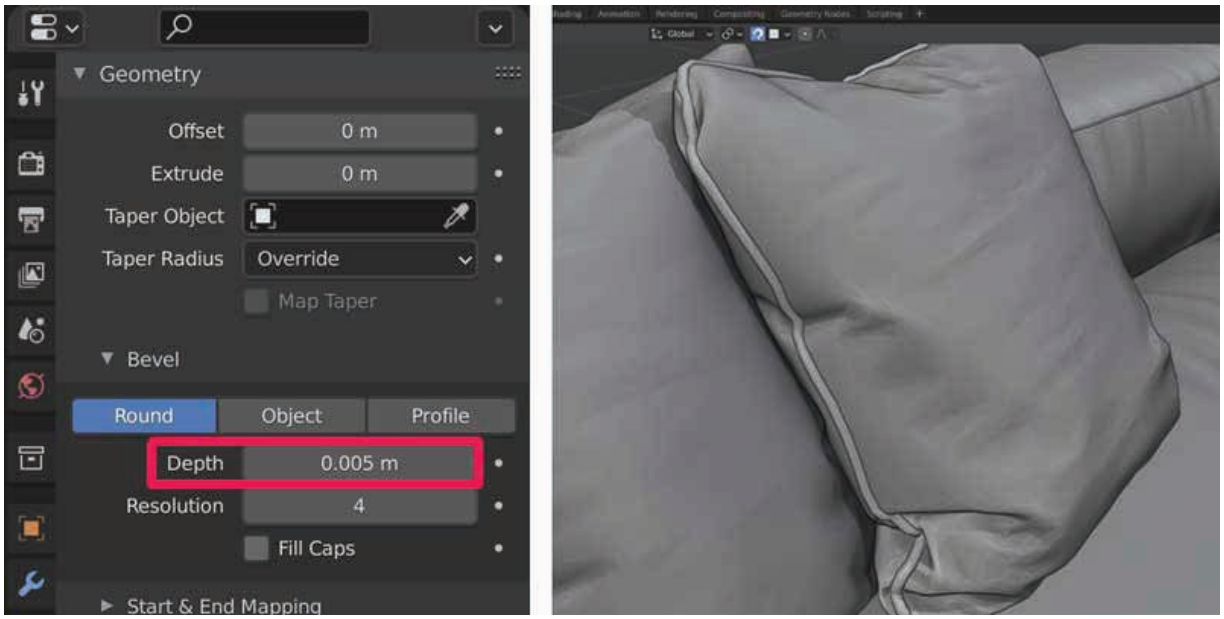


Then, just like how you duplicated some edges in Edit Mode with **Shift+D** and made separated them by **pressing P** and choosing **Selection**, select one vertex and separate it to its own object.

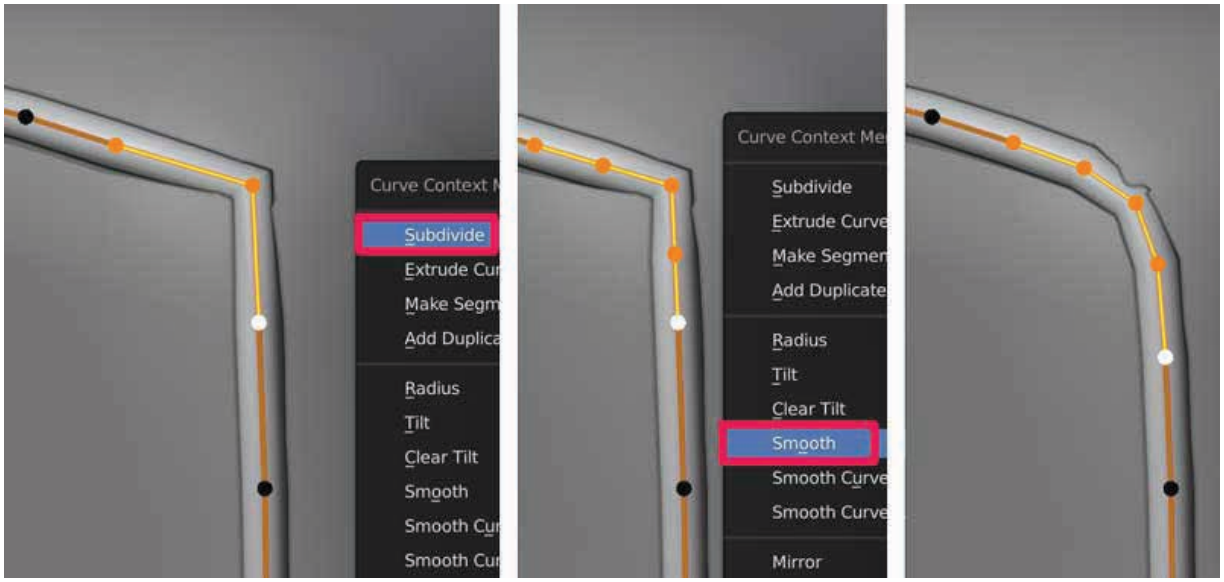
Then, select it in Object Mode and again in Edit Mode select the one vertex and extrude it by **pressing E** and moving the extruded vertex, or just **holding Ctrl and right-clicking** to where you want to extrude.



This way, go around the objects where you want to add a decorative edge. Then, as before, turn it into a Curve by going to **Object > Convert > Curve** and increase the **Depth value** of the Curve geometry to give it some thickness.



Don't forget to **turn off Snapping** once you don't need it anymore. If you want to make some part of it smoother, like the corners, you can select those Vertices in Edit Mode and **right-click, then choose Subdivide**. Then, right-click again and choose **Smooth**. Repeat by **pressing Shift+R** a few times until it looks as smooth as you want.

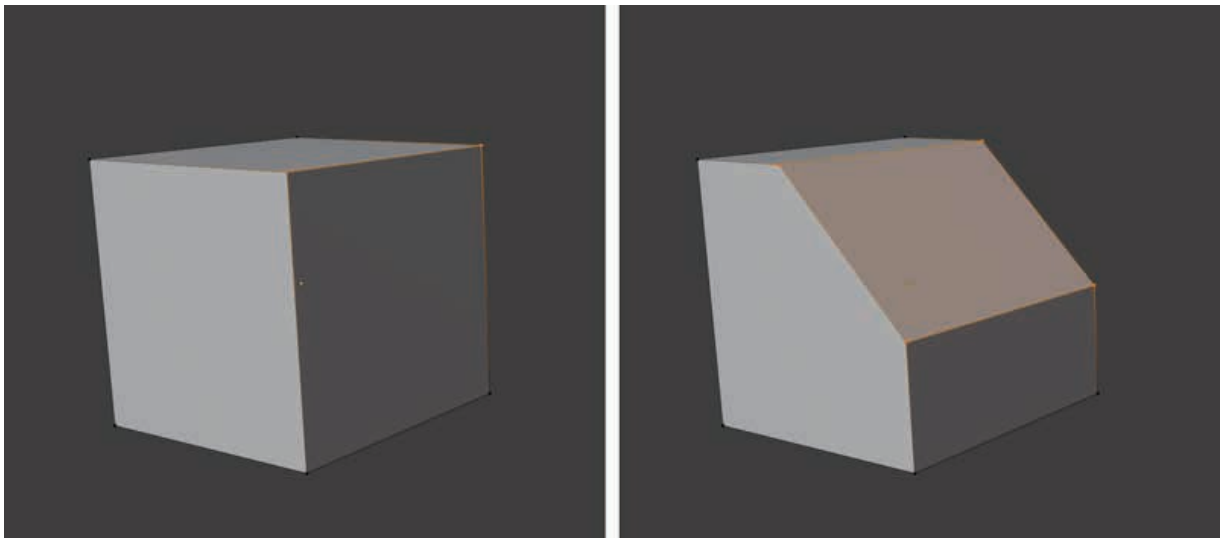


Sofa and stool models by MZiemys on Blendswap.
https://www.blendswap.com/blend/17656

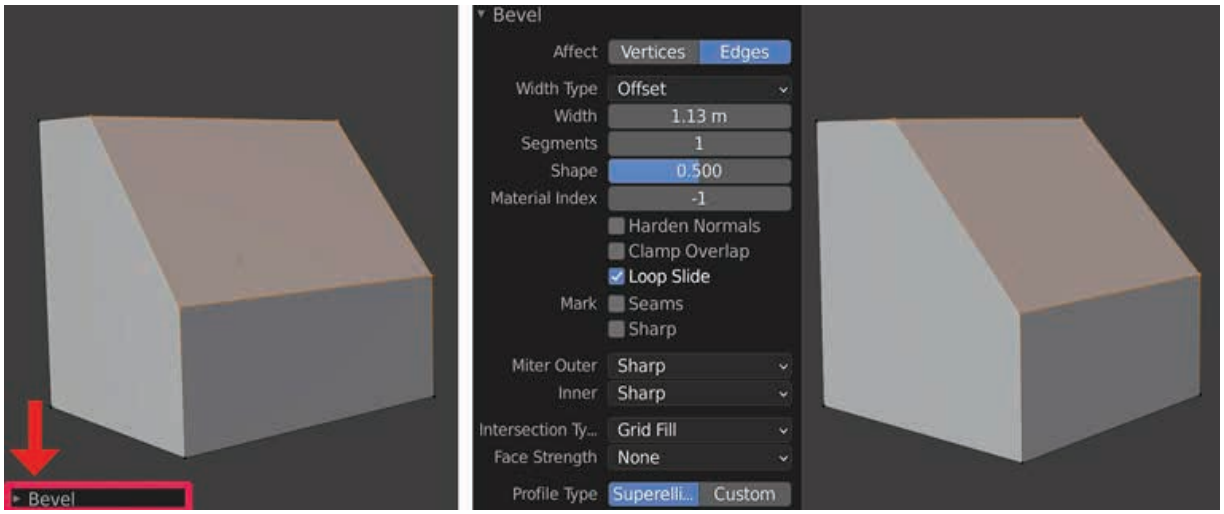
How to make Custom Bevel Profiles

Making stairs and decorative architectural corners is easy now with custom bevel profiles.

Start by beveling an edge using the **Ctrl+B** shortcut.

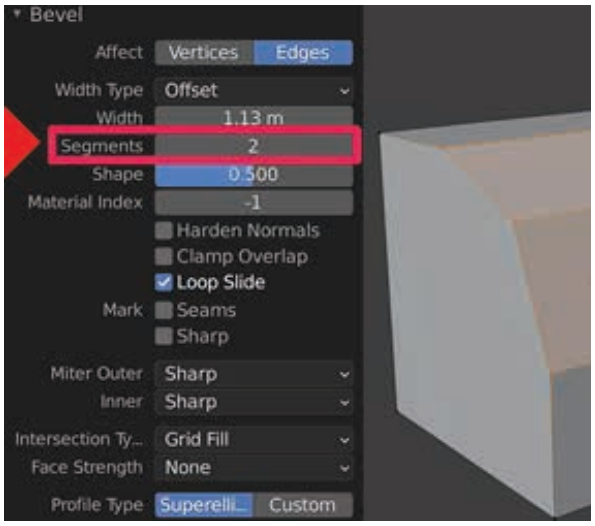


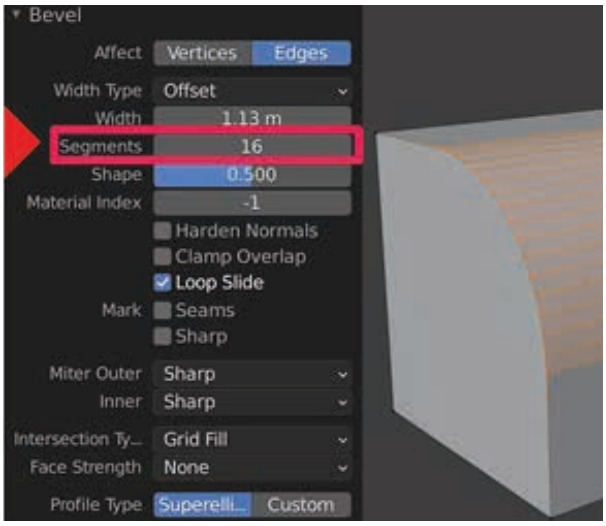
Then open the bevel menu.

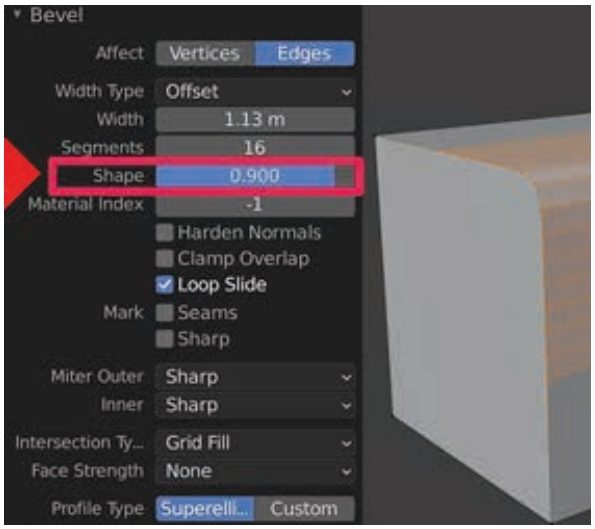


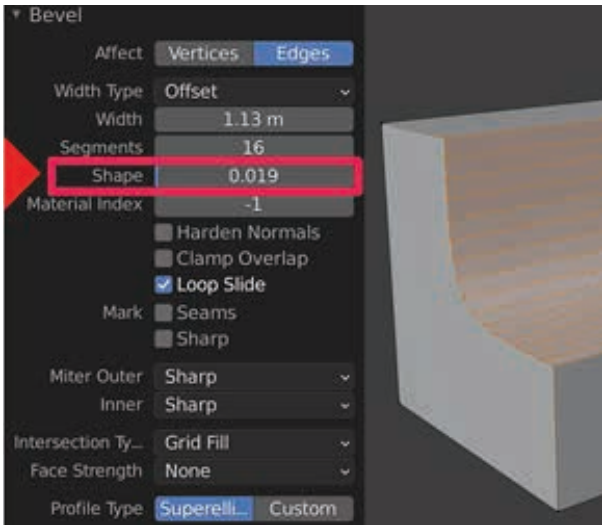
Click here to return to the index.

Increase the **Segments** and **Profile** as needed.

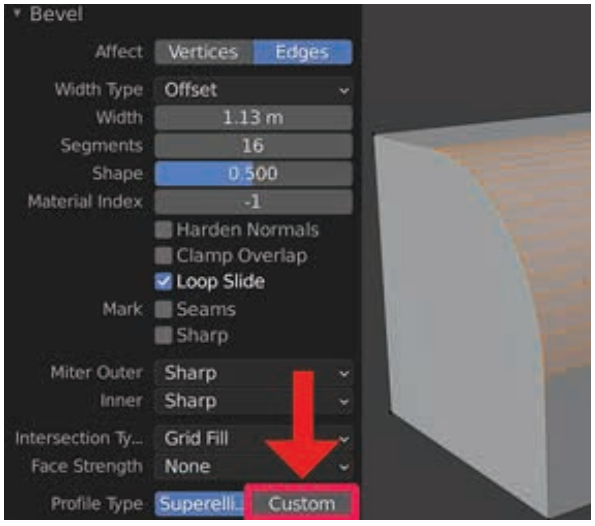


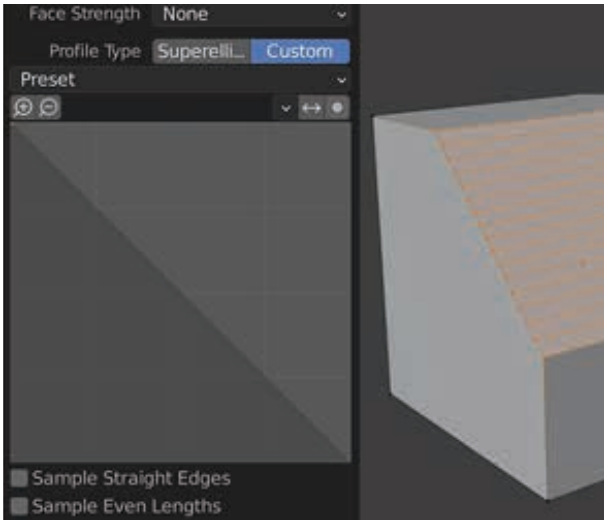




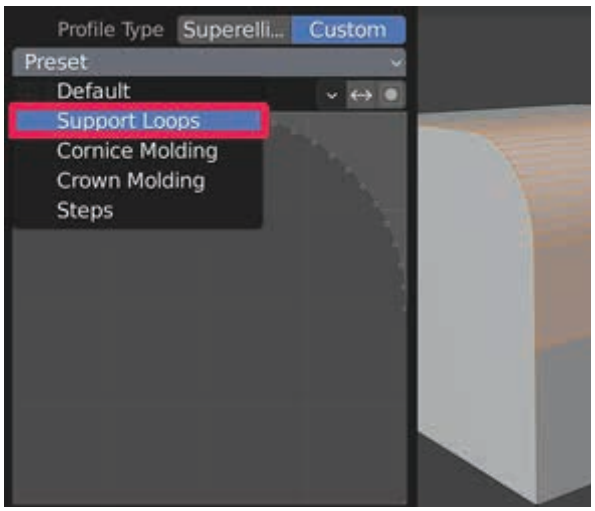


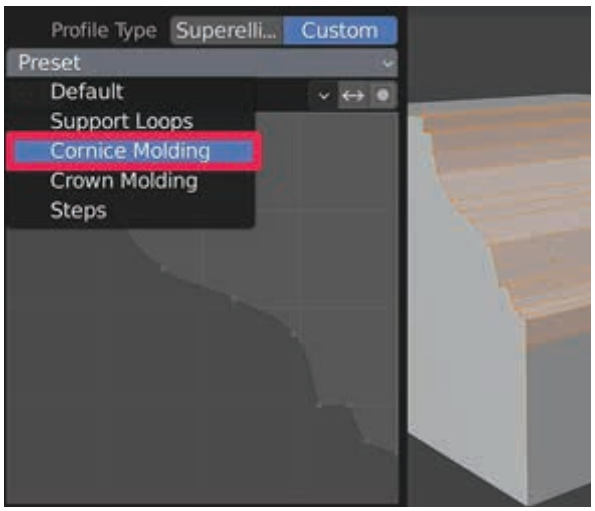
Turn on **Custom Profile**.

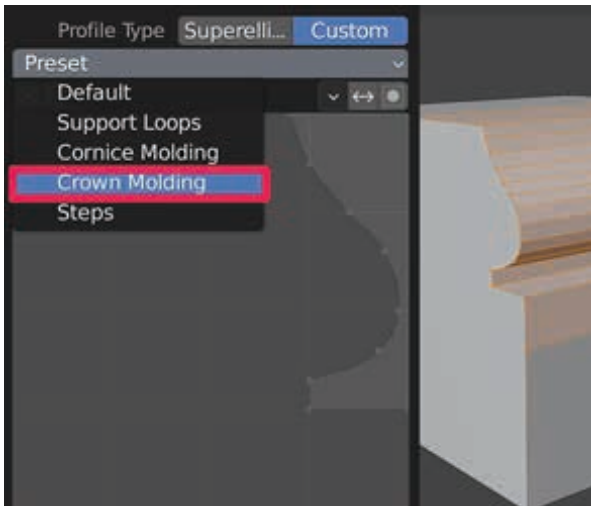


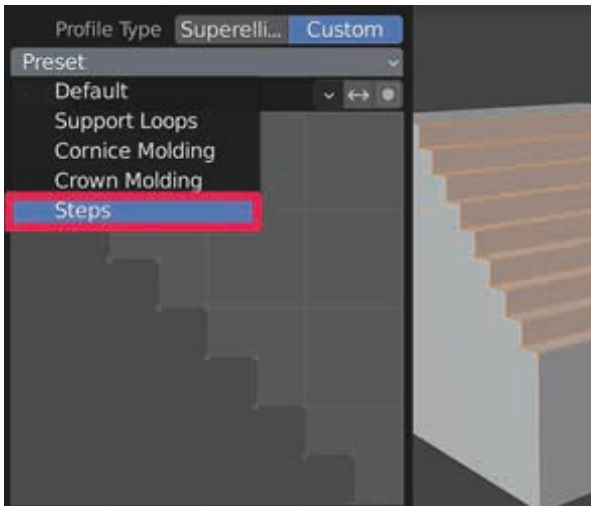


You can now choose several presets. **Stairs and moulding presets** will come in handy for architectural visualisation. There's also a **support loops** option.

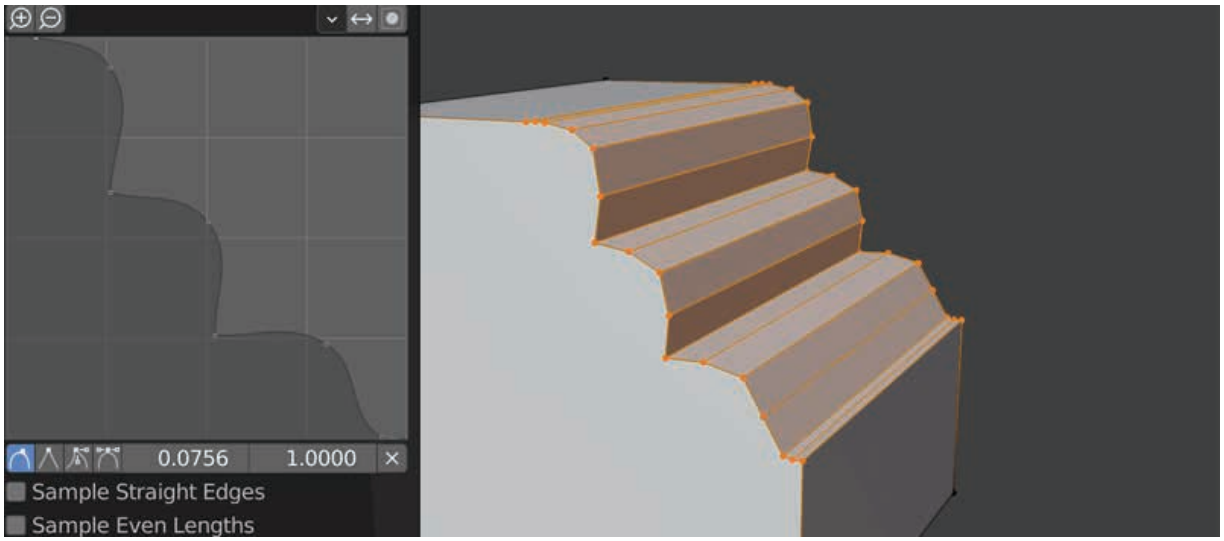








You can even create your own custom profile.



Note that this does not work in Blender builds before 2.82.

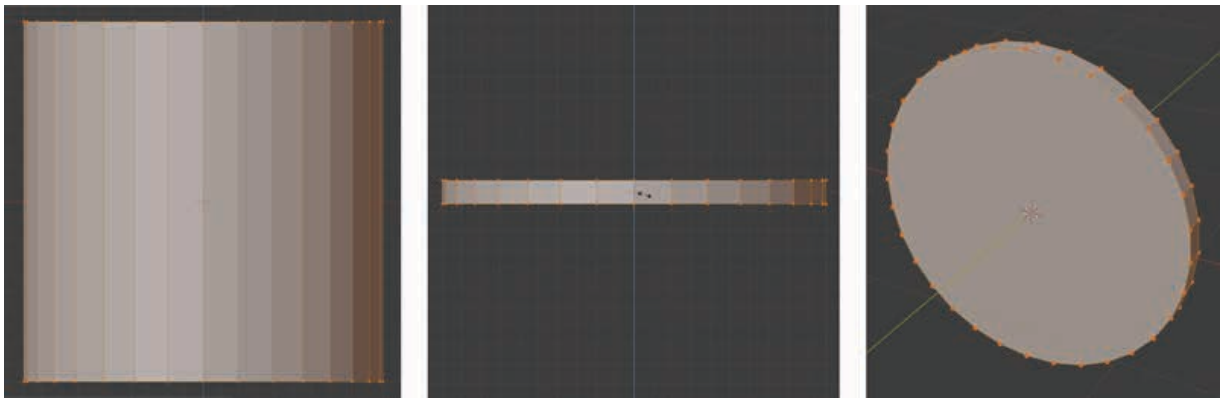
Modeling a designer door handle with Knurl Pattern

A **Knurl Pattern** is a pattern of crossed lines that are machined into the metal of hand grips (like surgical knifes or hand tools), or as in this example: a door handle in the style of home design firm Buster + Punch.

These are very popular in modern architecture. Their function is to give provide better grip, while also being pleasing to the eye.

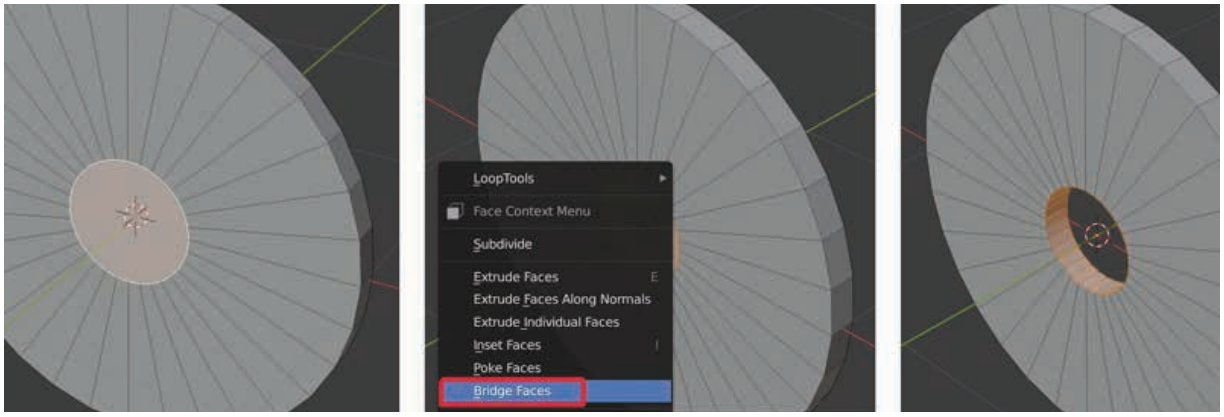
First, delete the default cube. Then **press 1 on the numpad** to go to Front orthographic view. **Press Shift+A** and add a **Cylinder** from the Mesh Menu.

Press Tab to go to Edit Mode and **press S+Z** to scale the cylinder down along the Z-Axis. Now **press R+X+90** to rotate it 90 degrees on the X-axis.



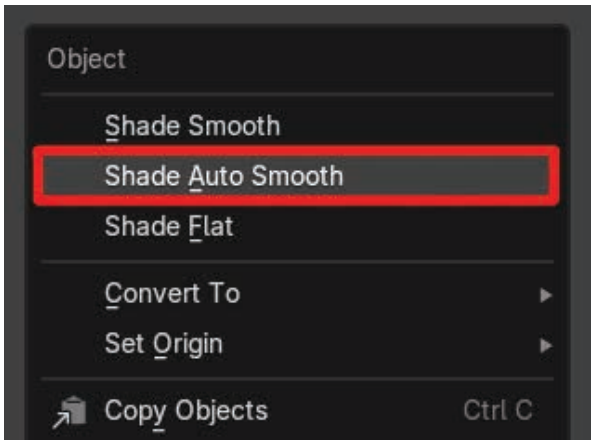
Select the front and back Face and **press i to inset** both, then scale the inset down by moving the mouse.

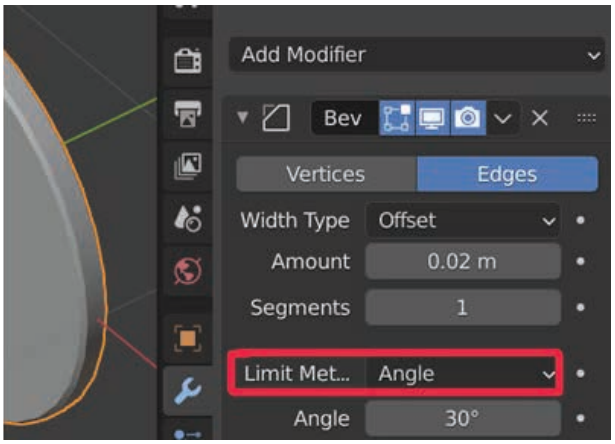
Right-click and choose **Bridge Faces** to create connecting geometry from the front to the back of the cylinder.



In Object Mode **right-click** on the Cylinder and choose **Shade Auto Smooth**.

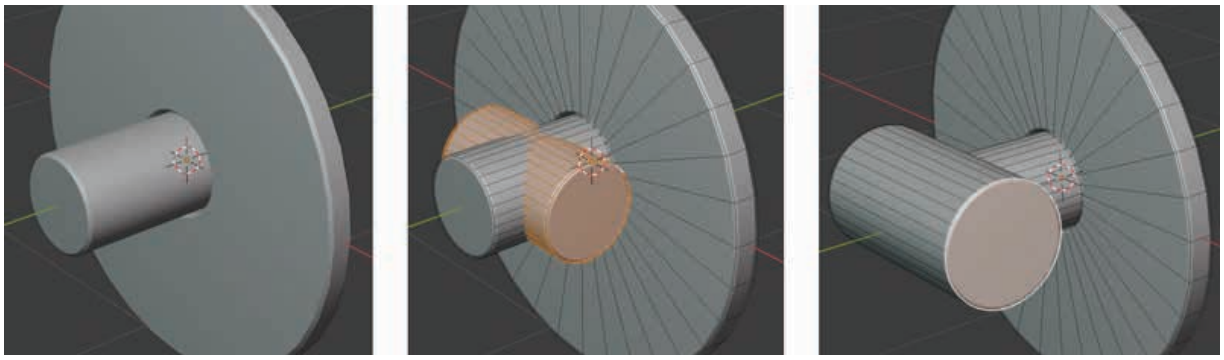
Add a **Bevel modifier**, set Limit Method to **Angle** and adjust the **Amount** until it looks good.



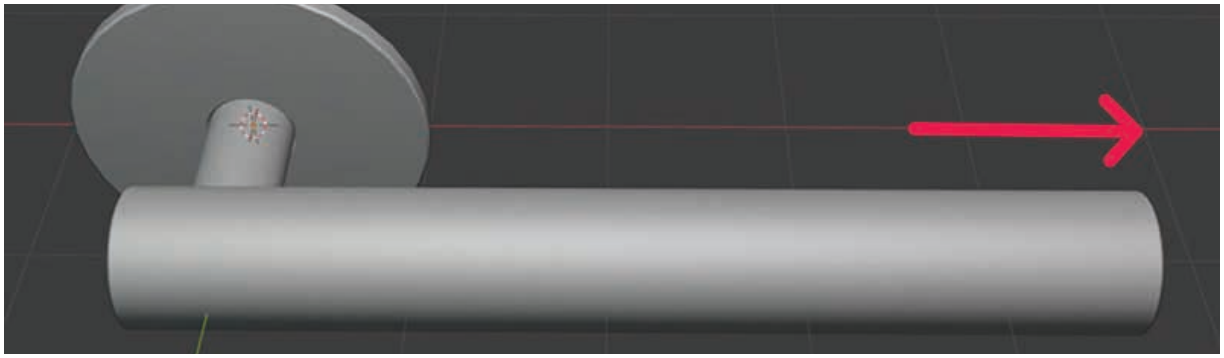


Add another **Cylinder** while in Edit Mode. This way the Auto Smoothing as well as any modifiers are already part of the mesh. Rotate and Scale the cylinder so that it fits in and stick out of the hole in the flat cylinder.

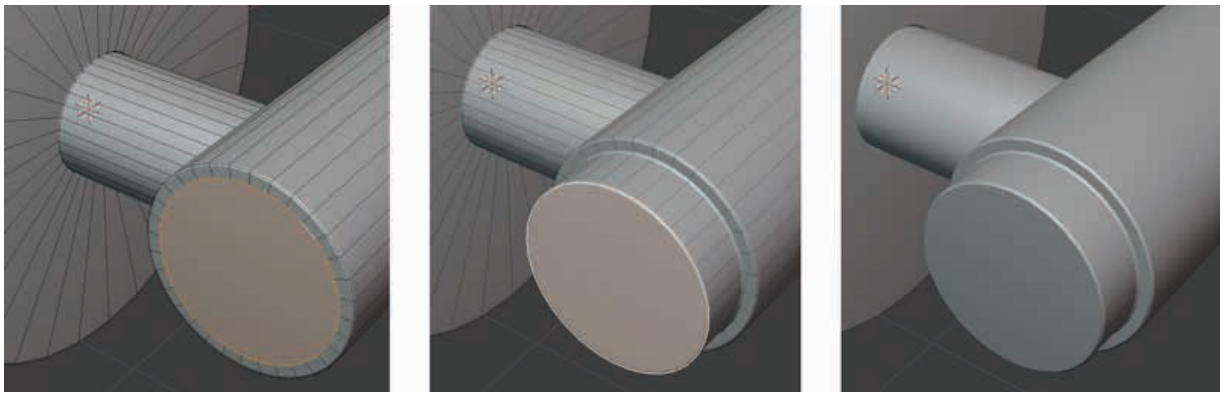
Duplicate it again and rotate it 90 degrees on the Z-axis by **pressing R+Z+90** and **pressing Enter**. Scale it up slightly. Select the round Face on the right.



Extend the cylinder to the right by moving the selected Face on the X-axis by **pressing G and X**.

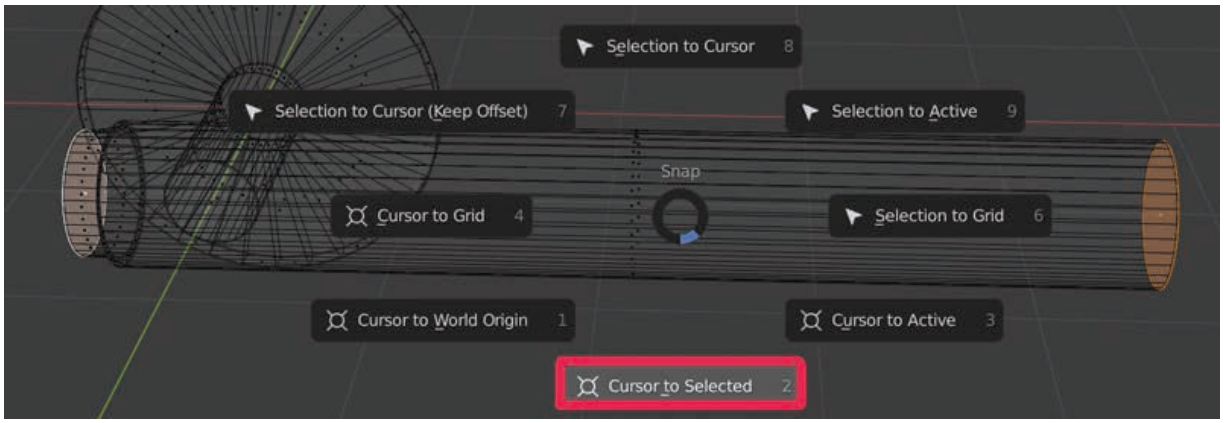


Select the Face on the left side of the handlebar cylinder, and **press i** to inset, scaling the inset down slightly by moving the mouse. Then, **press i again** and then **hold Ctrl while moving the mouse**, in order to Extrude it.



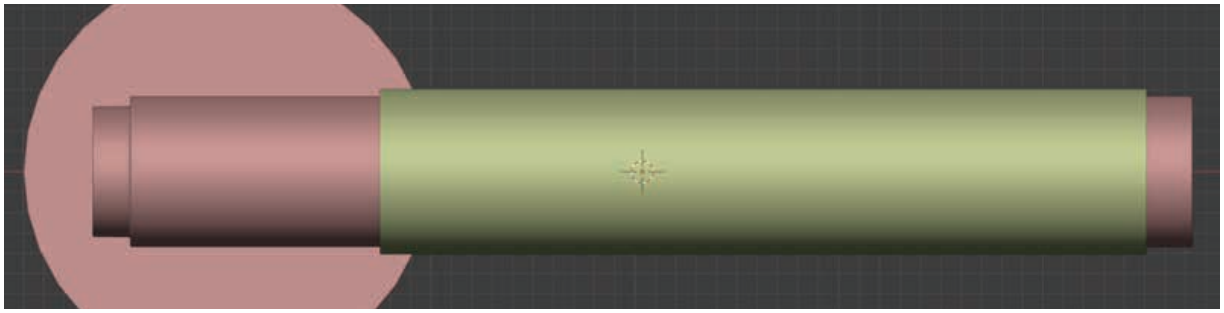
Okay, that's the basic shape of the door handle. But now comes the interesting part, the **knurling**!

Select both ends of the longest cylinder in Edit Mode and **press Shift+S**. Choose **Cursor to Selected**. Now the 3D cursor is in the center of that cylinder.



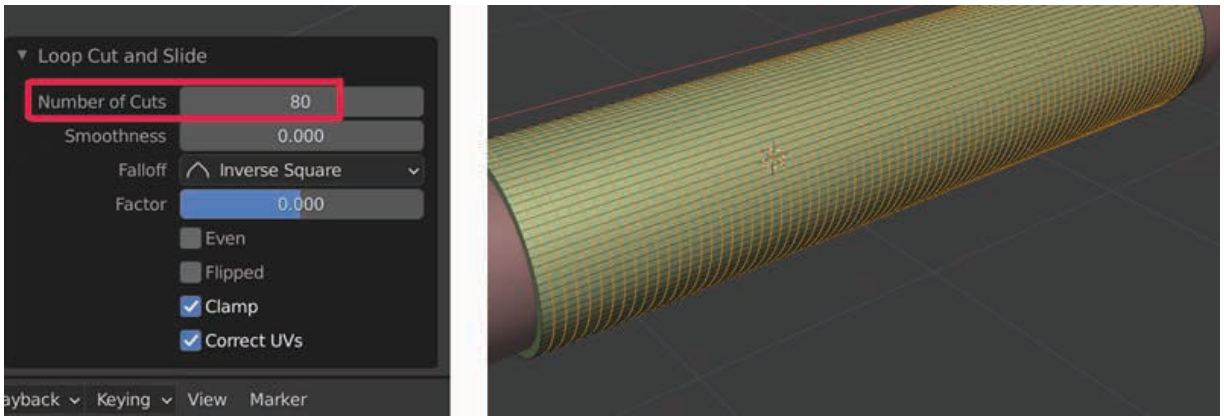
In Object Mode, Add a new Cylinder and give it **80 vertices** instead of the default 32. Rotate it 90° on the Y-axis.

In Edit Mode, scale it down and scale it along the X-axis by **pressing S+X** to elongate it, then move it along the X-axis if necessary by **pressing G+X** so it's in the right place. The new cylinder is pictured in green here.

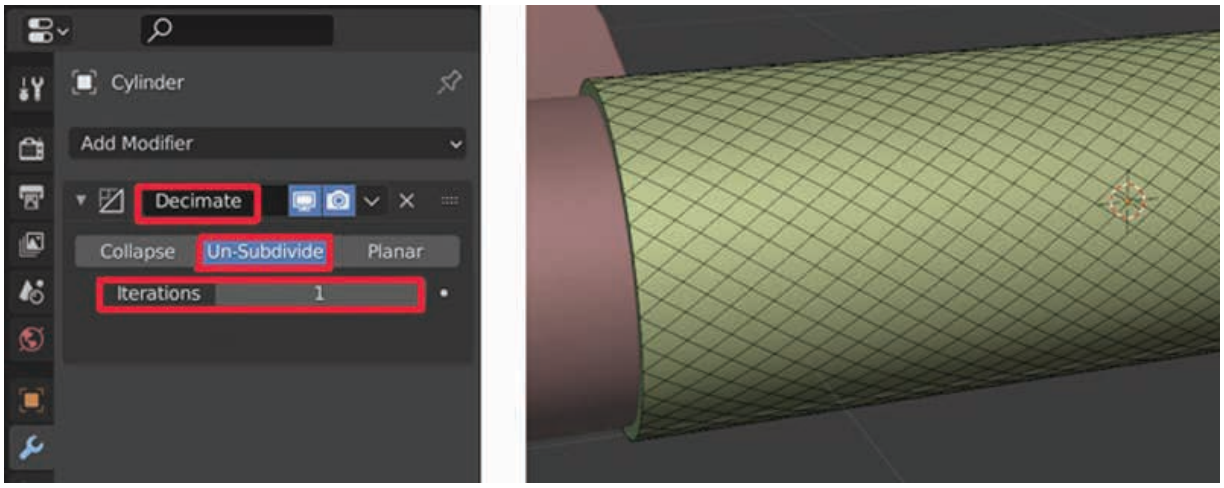


Add edge rings to the cylinder in Edit Mode by **pressing Ctrl+R** and **left-clicking twice**, so that you get the **Loop Cut and Slide menu** in the bottom left corner of the viewport.

Open it and add **80** in the **Number of Cuts**.

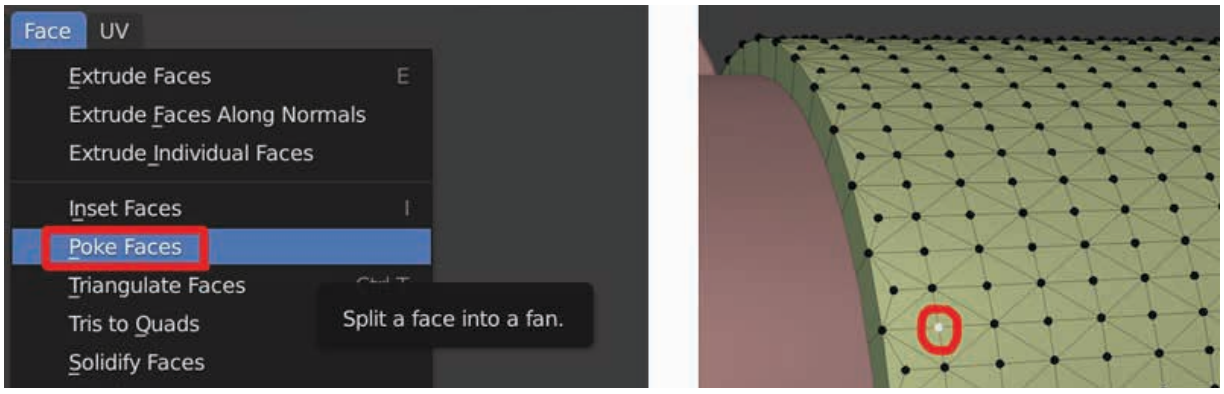


Add a **Decimate modifier** in Object Mode and set it to "**Unsubdivide**", with **1 iteration**. Apply the modifier by hovering the mouse cursor over it and **pressing Ctrl+A**.



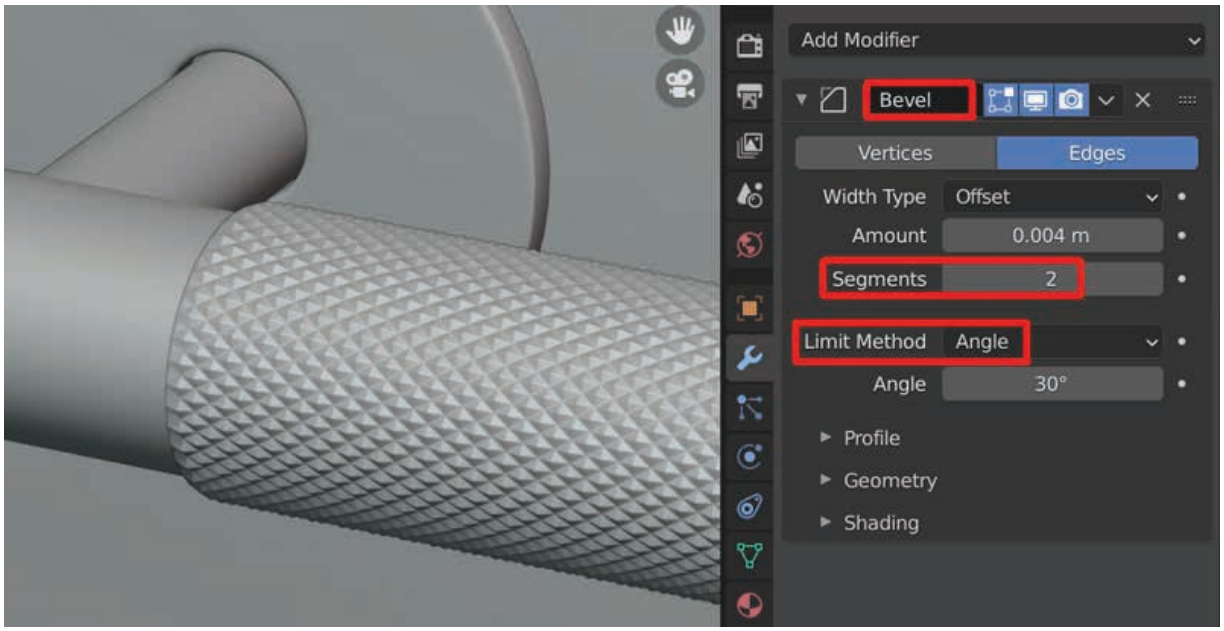
In Edit Mode **press A** to select all the faces, then from the Face drop-down menu choose "**Poke Faces**".

Select one of these vertices like the one circled below, and **press Shift+G**, then choose "**Number of adjacent faces**". This selects all similar vertices.



By **pressing Alt+S** and moving the mouse you can now scale these vertices outwards to create the knurling pattern. Be careful not to scale it too far - you want to keep it subtle.

If you want you can add a **Bevel modifier**, set to Limit Method "**Angle**". I recommend using **2 Segments**, and adjusting the **Amount** value carefully until it looks good.



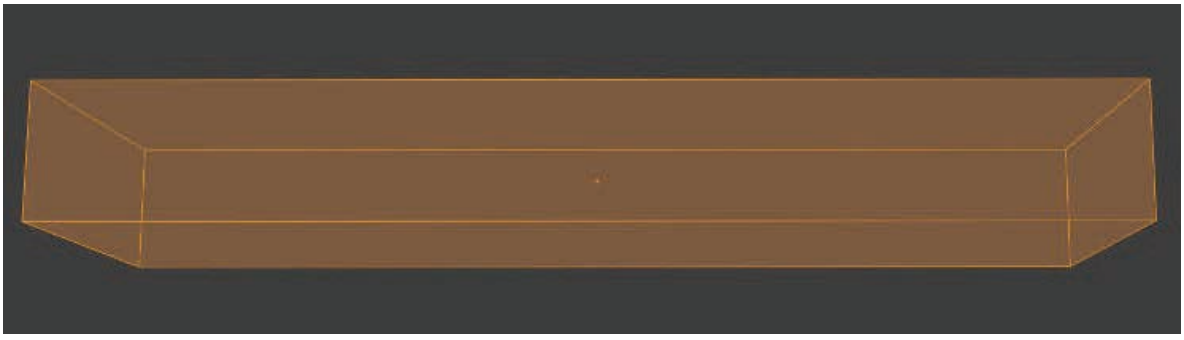
Modeling a window

You're probably already familiar with the Bevel function, which is activated in Edit Mode by **pressing Ctrl+B** or by using the Bevel modifier and which bevels an edge with one or more segments. But did you know that Blender has a custom bevel profile that you can use to make really detailed bevels?

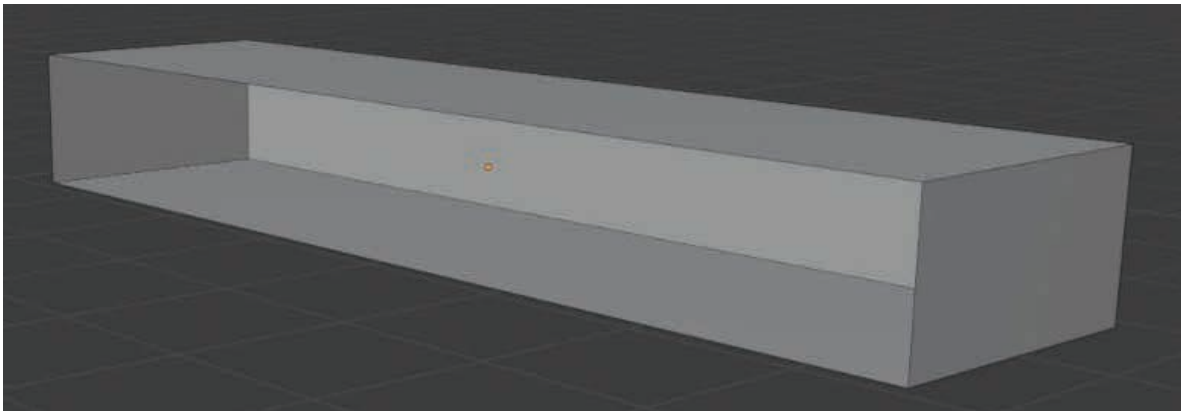
Let's have a look at a typical real-world case: modeling a detailed window for an architectural render. We will be creating this window:

First, we'll make the frames around the window. Select the default cube and in Edit Mode, scale it down to a rectangle as shown in the below image. If you want to follow along with the exact size I used, **press S+Z+[0.12]** to scale it on the Z-axis, **S+X+[0.94]** to scale it on the X-axis and **S+Y+[0.23]** to scale it on the Y-axis.

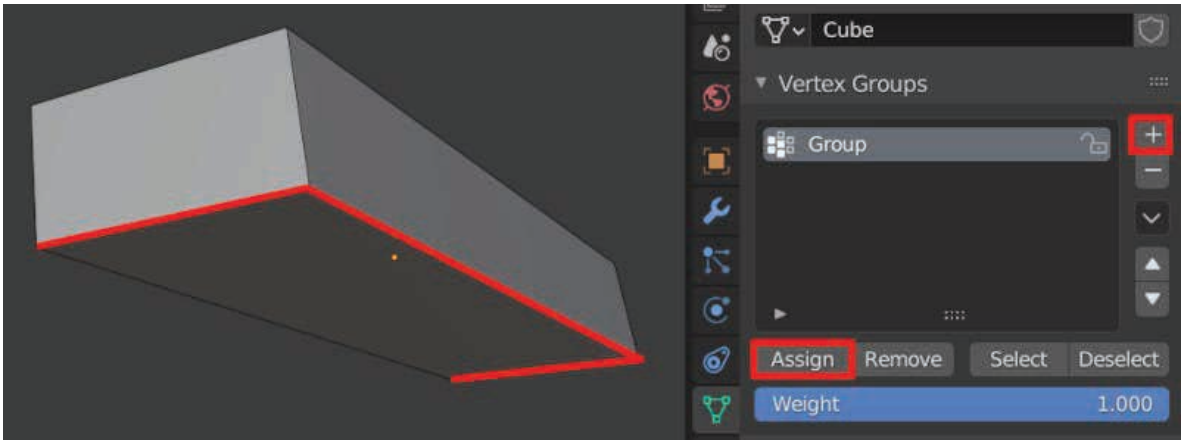
Then to make room for the rest of the window below it, move it up by **pressing G+Z+[1.77]** and finally **G+Y+[-0.16]**. Don't worry - you can still move and scale the window later in case you made a mistake.



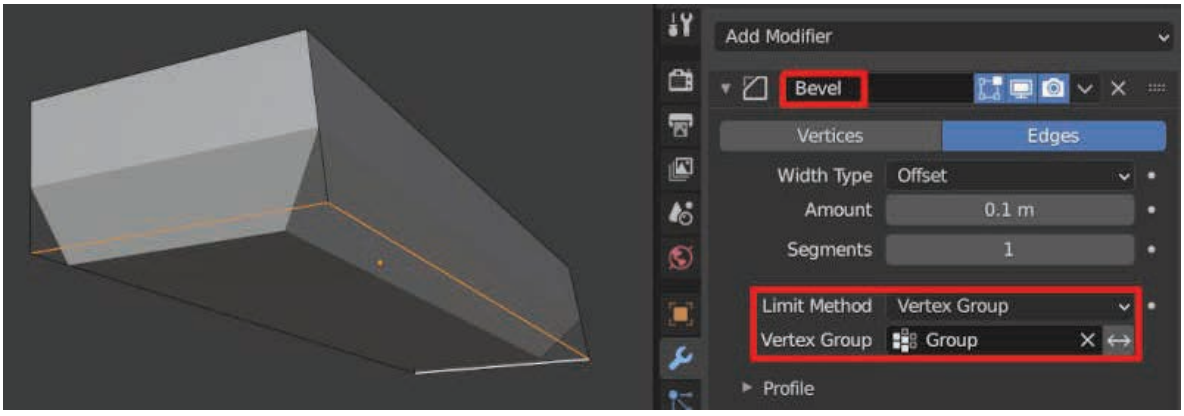
Now, remove the Face at the back of the cube by selecting it in **Face select mode** (keyboard shortcut: 3) and **pressing X**, then choosing **Faces**. Now we've got a stretched cube with a hole on one side.



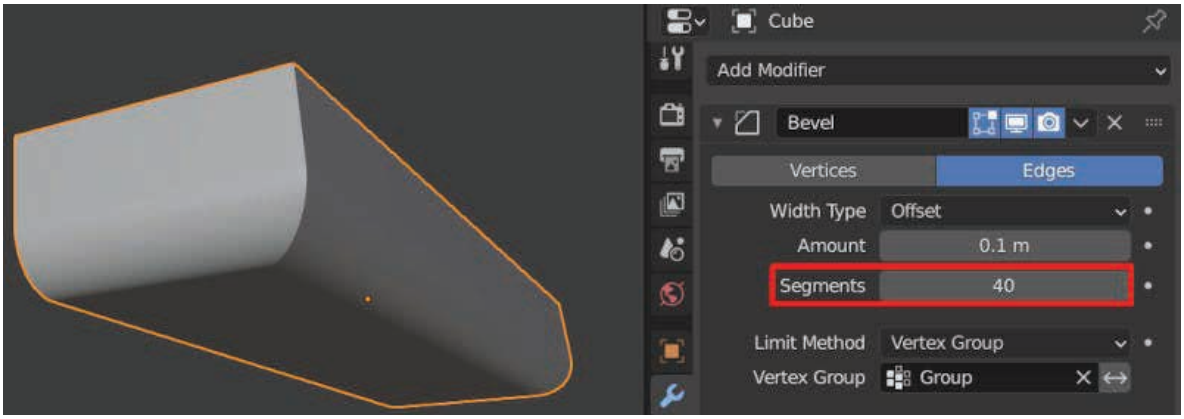
Next, create a new Vertex Group by **clicking on the + icon** in the Vertex Groups panel. Select the three edges at the bottom shown in the image below, and **click on Assign** to add them to the newly created Vertex Group.



Now add a **Bevel modifier** to the cube. As the **Limit Method** choose **Vertex Group** and as the Vertex Group, choose the group you've just created (by default it's called "**Group**", unless you renamed it).

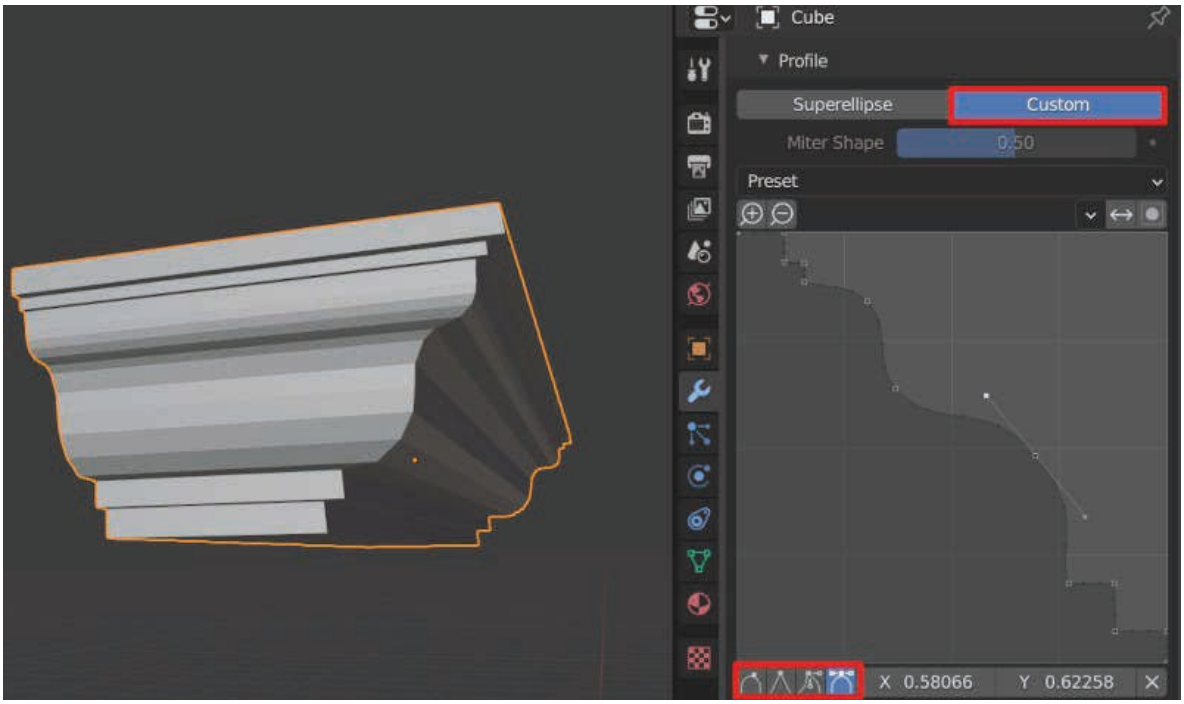


You'll notice that only the edges that you added to the Vertex Group are being beveled. That's exactly what we want. Now, add some more segments otherwise the next step won't work. 40 Segments will do nicely. You'll notice that the bevel becomes smooth and rounded.

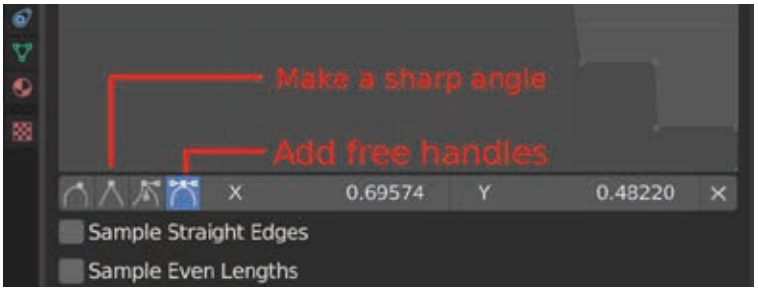


Now let's create a **Custom Profile** to make it really interesting. Set **Profile** to "**Custom**" and you'll get a 45° angle curve by default. By **clicking on the curve** you add points, which are set to Bezier handles by default. If you want to add sharp 90° angles you can **click on the Vector Handle icon** below the curve (the pyramid-shaped icon).

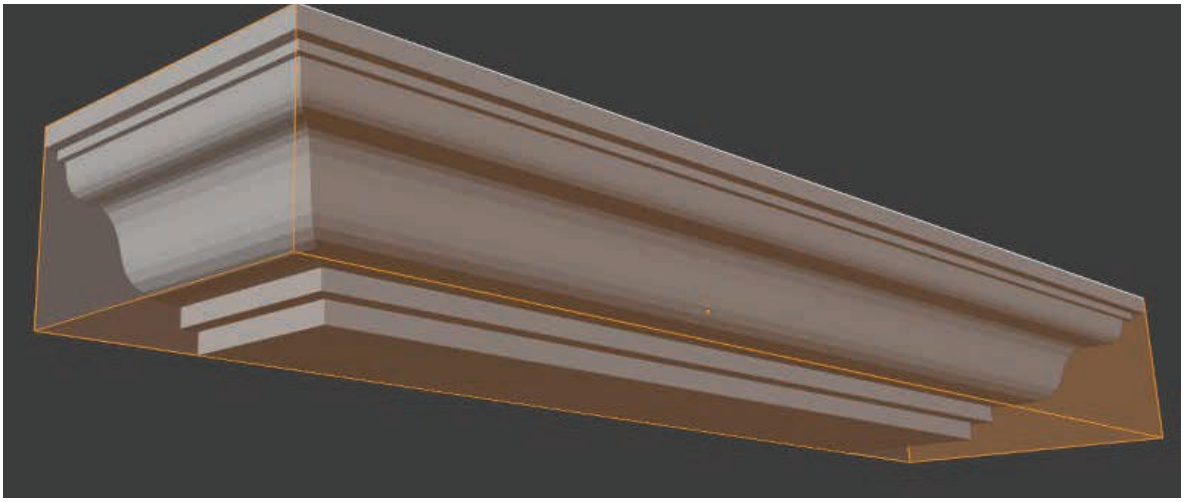
This turns a selected handle into a "sharp" one. You can also add "**free handles**" by clicking on the corresponding handle, as shown below. This can be useful for tweaking the roundness of a handle. Create the curve as shown below (or one of your own design). You can see the model reflect the changes immediately.



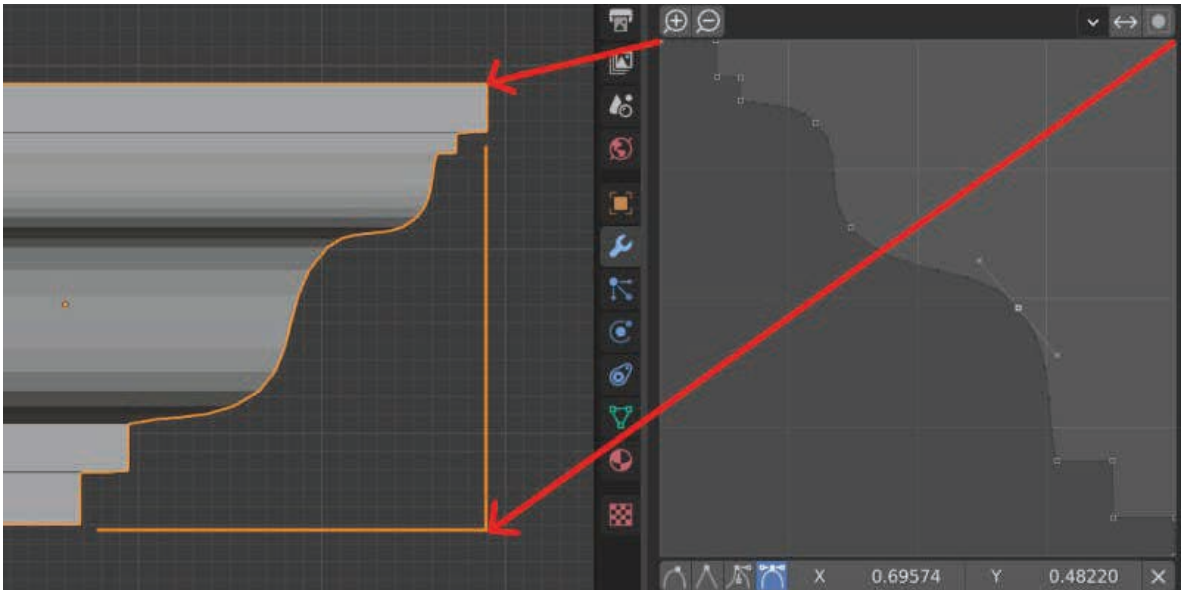
Here's a closer look at the icons for the handles:



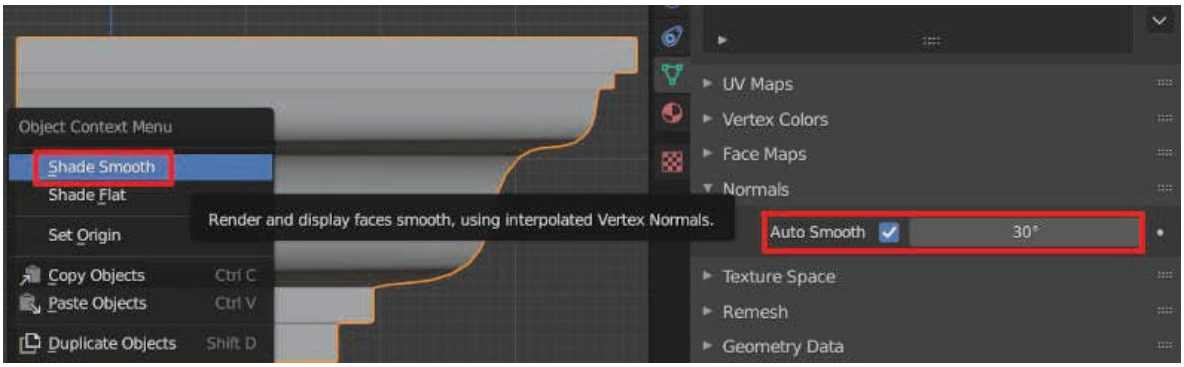
Notice that this is a non-destructive method, so if you go into Edit Mode, the model is still very simple, consisting only of 5 Faces.



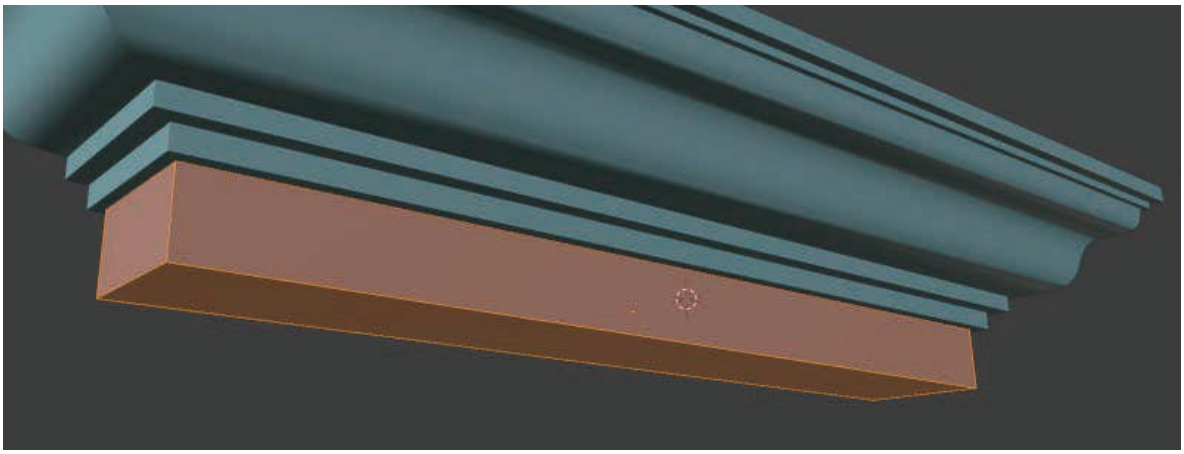
It may be a bit confusing how the curve corresponds to the bevel on the model, because in this particular case we're beveling the edges at the bottom instead of the ones at the top. Below you see how the curve relates to the bevel in this case:



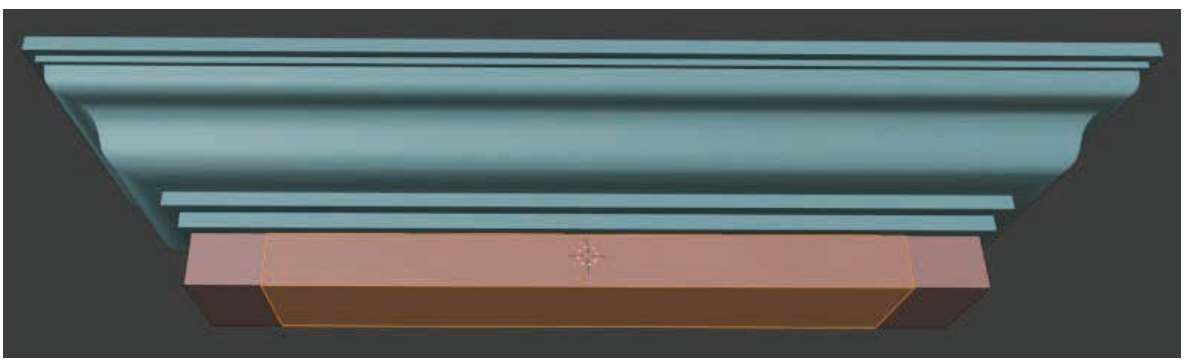
To finish this piece right-click on the object in Object Mode and choose **Shade Smooth**, then set the Normals to **Auto Smooth**. An angle of **30°** works well in most cases. Now it looks smooth in the right places.



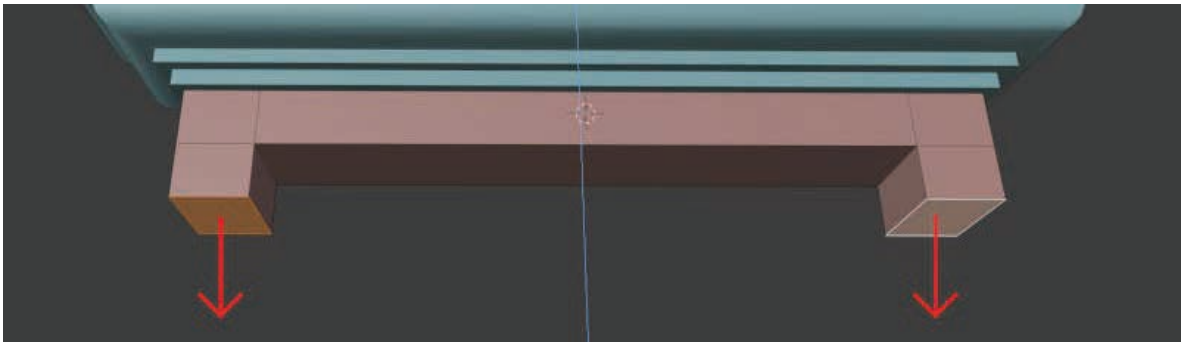
Press Shift+A and add another **Cube**, then scale it down in Edit Mode by **pressing S+[0.05]**. Scale it along the X-axis next by **pressing S+X+14**. Finally scale it along the Y-axis by **pressing S+Y+2**. Move it to the bottom of the beveled object, as shown below.



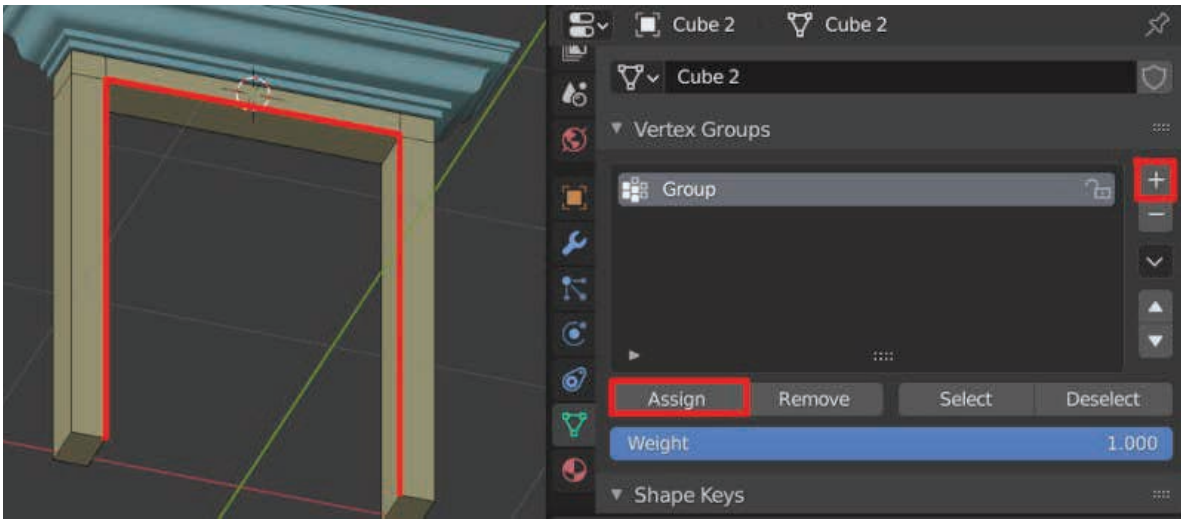
Now add an edge loop in the middle by **pressing Ctrl+R** and then bevel it by **pressing Ctrl+B** until it looks like this:



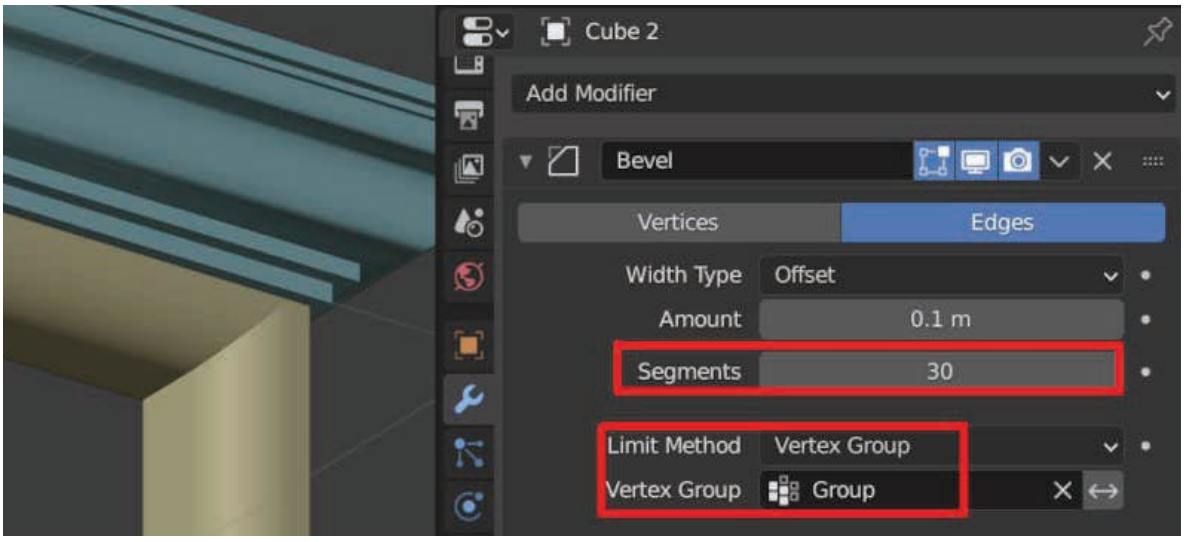
Then extrude the two faces at the bottom left and right down by **pressing E** followed by typing 1.55 on the Numpad, and then pressing Enter to confirm.



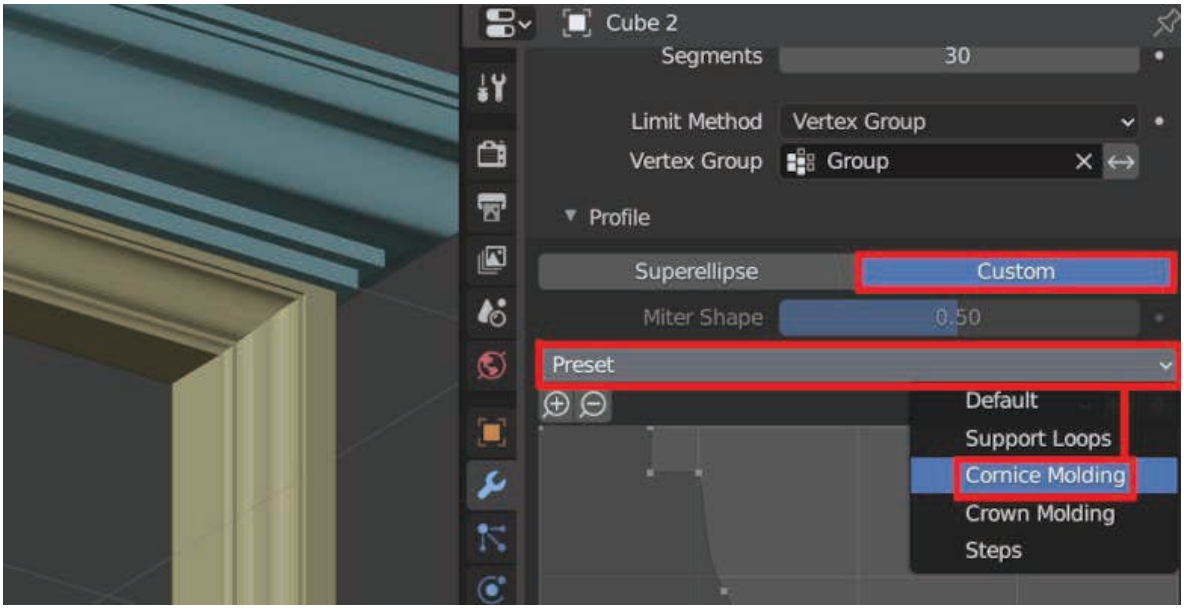
Just like before, create a **Vertex Group** for the edges indicated below.



And as with the previous object, add a Bevel modifier and set **Limit Method** to **Vertex Group**. Choose the group you created. Add about **30 segments**.



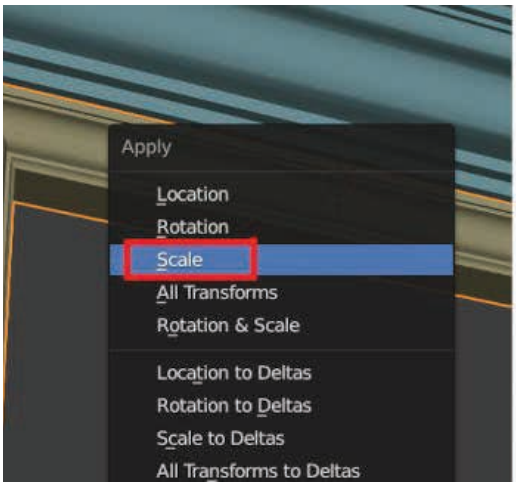
Now, instead of creating our own Custom Bevel Profile, let's have a look at the **Presets** to see if we can use any of those. Set **Profile** to **Custom** as before, but this time choose the Preset "**Cornice Molding**". This is a nice one for windows and for doors as well.

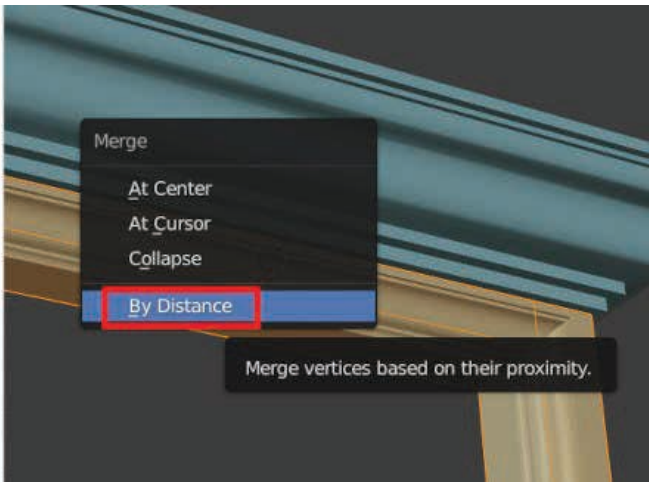


In case the beveling doesn't work, there are two things you should check:

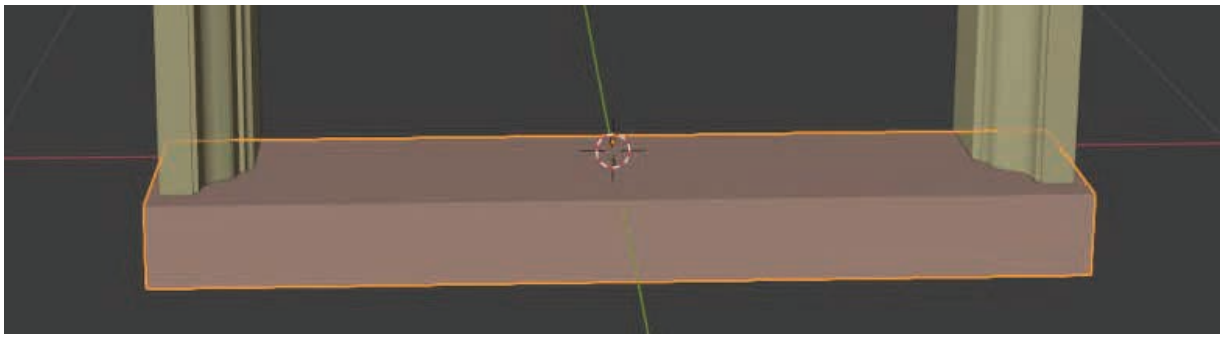
1) have I created double vertices by accident? Select all in Edit Mode and **press M**, then choose **Merge by Distance**.

2) If that doesn't help, in Object Mode check if the Scale is applied by **pressing Ctrl+A** and choosing **Apply Scale**. Normally this should solve any issues with the beveling.



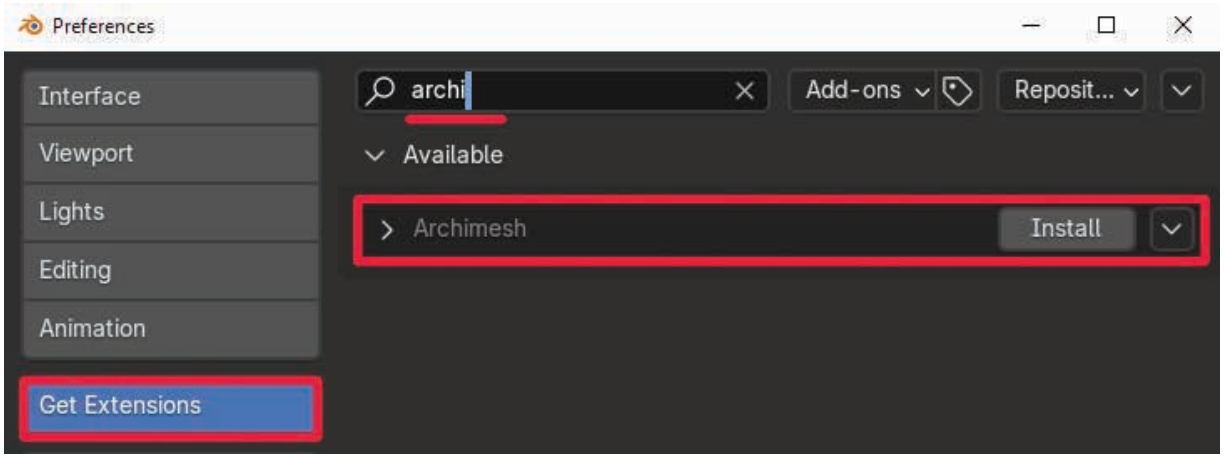


Finally, add another cube and scale it to fit below the window like this:



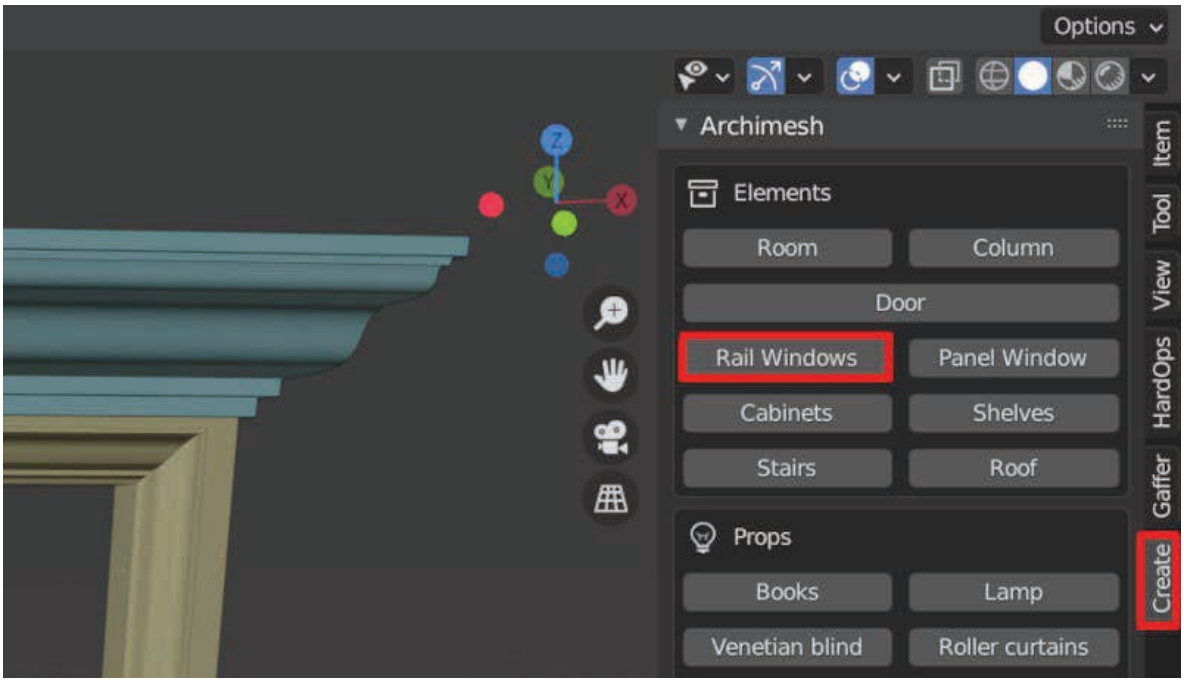
To make the rest of the window we can actually save time and "cheat" a bit by using an add-on that comes with Blender by default. You just have to enable it in Preferences. I'm talking about the "**Archimesh**" add-on.

To enable it, search for "archi" in the Get Extensions tab. Then click on Install.

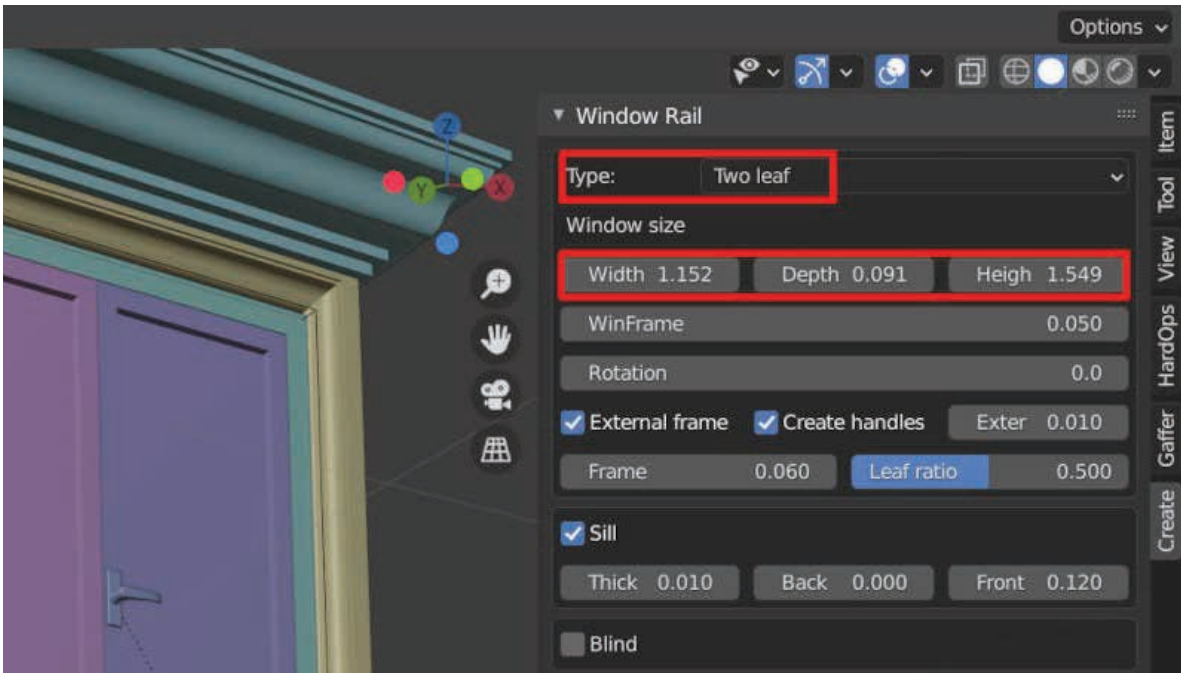


To add a window using Archimesh we have a couple of options: We can add it like any other primitive by **pressing Shift+A** and choosing one of the window types there, or we can press N to open the **Option panel** in the 3D viewport. Here we find a tab called "**Create**". This is the Archimesh add-on tab.

Let's go with that latter method, since we need to use this panel anyway to tweak the settings of our window. Make sure the 3D Cursor is at the world origin and click on "**Rail Windows**".



This creates a default set of Rail Windows, which we will need to tweak a bit. First of all, change the **Type** to **Two Leaf** and change the **Window size** to fit the frame you modeled. Below you can see the scale I arrived at to fit in the frame I created.



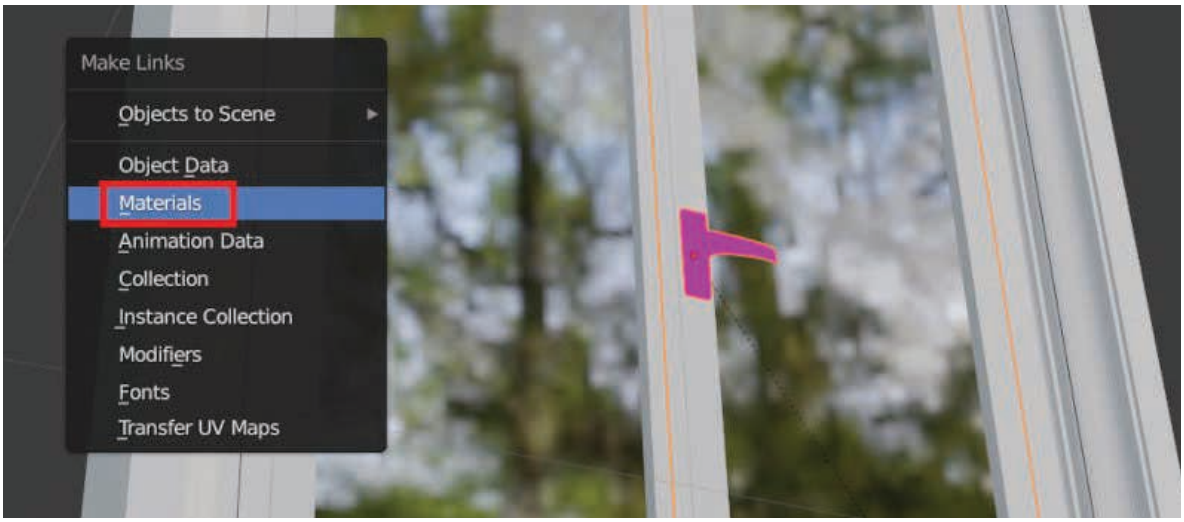
As you can see, the add-on has created the window including the handle and glass panes. Very easy!

You may need to select the objects comprising the beveled frame and move them backwards a bit on the Y-axis to fit better around the window.

If you switch to Material View in the 3D viewport, you may notice that the handle of the window is purple. This indicates a missing texture on that material.

A simple solution is to select the handle and then while **holding Shift** select a part of the window (but not the glass) and **press Ctrl+L**, then choose **Materials**.

This way the handle gets the material of that other object. Or, you can simply remove its material and create a new one.



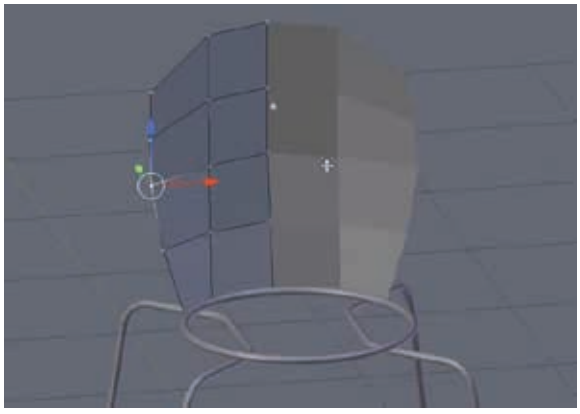
That's it! You've just created a very nice-looking window.

Chair modeling: Wireframe backrest

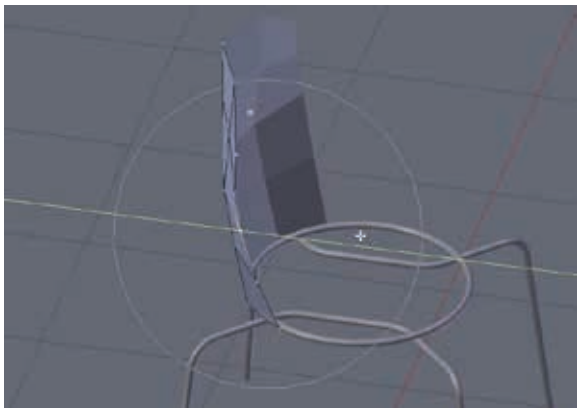
In this tutorial we look at creating this kind of metal mesh behind the backrest of a chair.

If you want to see a longer time-lapse video of me modeling this chair check this Youtube link (Not a tutorial, just a time-lapse recording of the entire modeling process).

For the purpose of this tutorial we'll skip the detailed explanation of how to model the shape of the backrest - suffice it to say, it's a plane with a Mirror modifier (so we only need to model one half), and the vertices are moved by hand (using G) to create the shape.



You can also use Proportional Editing (turn it on/off with shortcut o) to move the vertices.

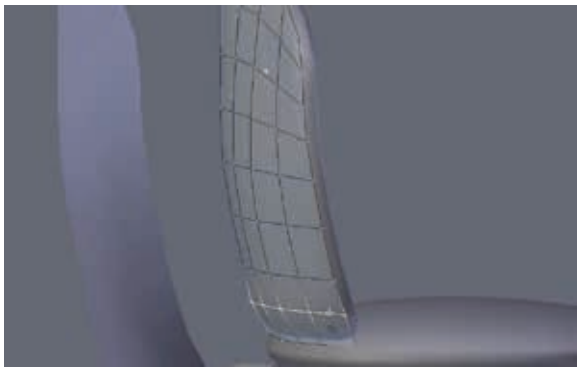


In Object Mode, Duplicate the backrest with **Shift+D** and move it back away from the backrest.

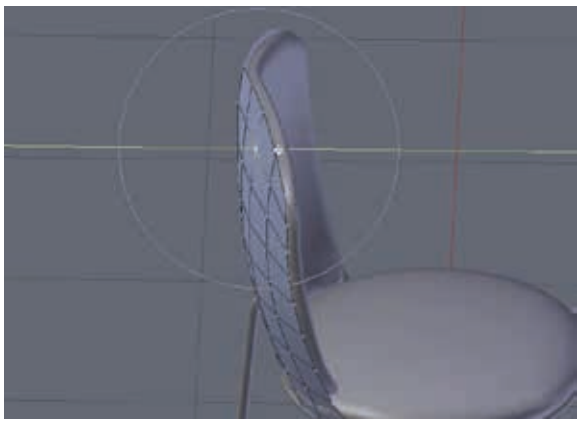
Add a **Solidify modifier** to the backrest to give it some thickness.

You can still tweak it in Edit Mode after adding the modifier.

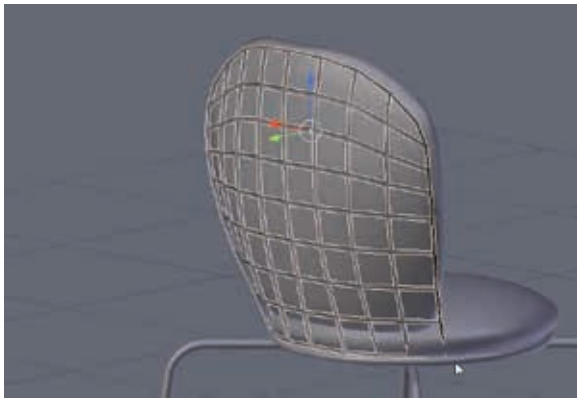
The reason we moved the duplicated back a bit is to be able to still see the original backrest and edit it a little after adding the Solidify modifier.



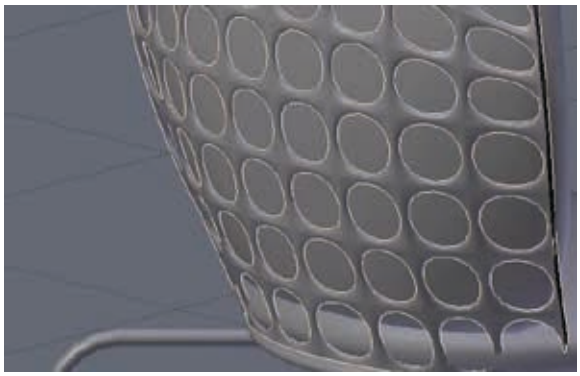
Once we are happy with the backrest shape, move the duplicate back and Edit it where necessary to fit the solidified backrest.



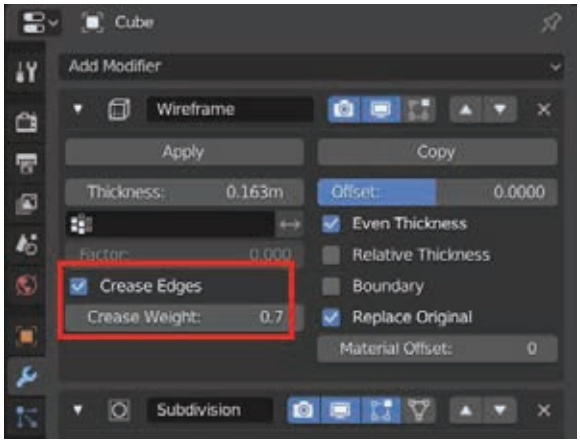
Add a **Wireframe modifier** to the duplicated backrest. This creates a kind of metal mesh look, though not smooth yet.



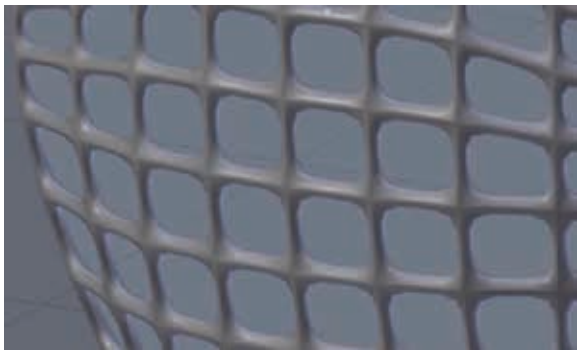
If we add a **Subdivision modifier** (underneath the Wireframe modifier), it will make the result smooth, but also creates these circular holes, which may not be the result we are going for.



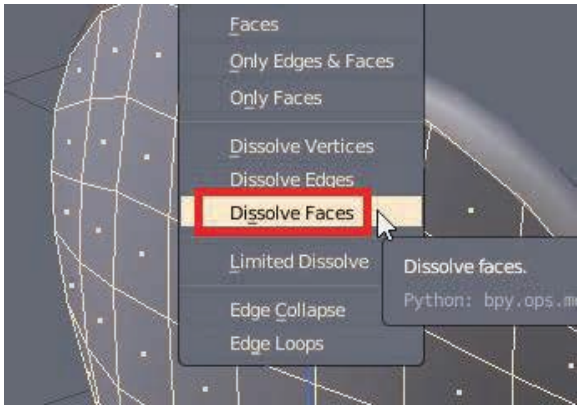
We can reduce this somewhat by turning on **Crease Edges** and experimenting with the value of the **Crease Weight**.



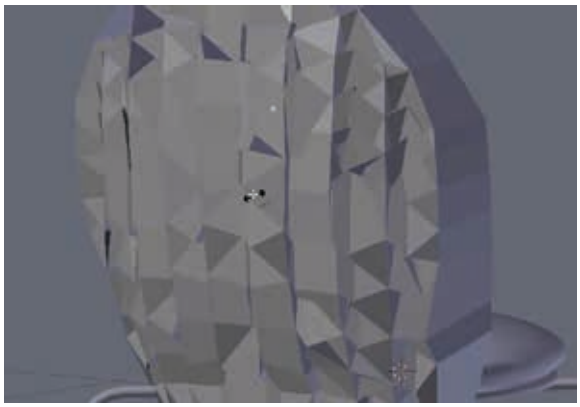
This may still not be exactly what we want, however. But it's good to know you can achieve this kind of look as well.



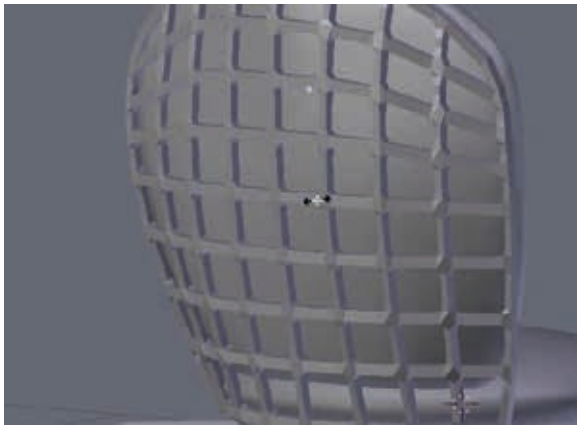
What we can do instead, is remove the Wireframe modifier and then select all faces in Edit Mode with **A**, then **press X** and choose "**Dissolve Faces**".
This removes the faces, but keeps the edges.



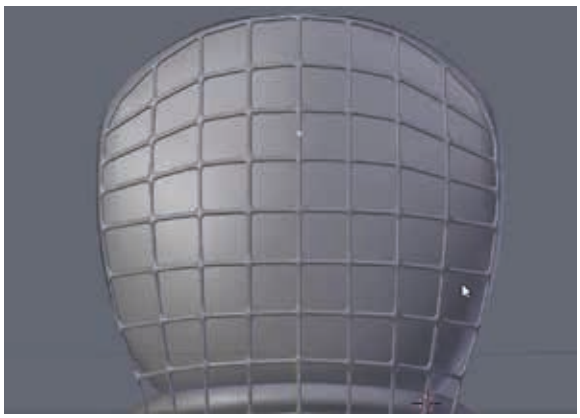
At first it will probably look like a mess, like this.



However if you select all the vertices by **pressing A** and then scale them by **pressing Ctrl+A** and dragging the mouse, you can make the mesh thinner.

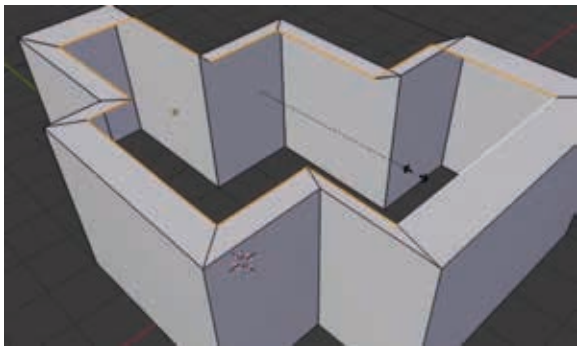


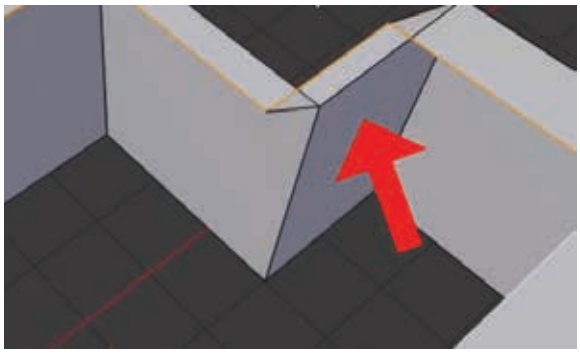
With a **Subdivision modifier** on it, now it looks good!



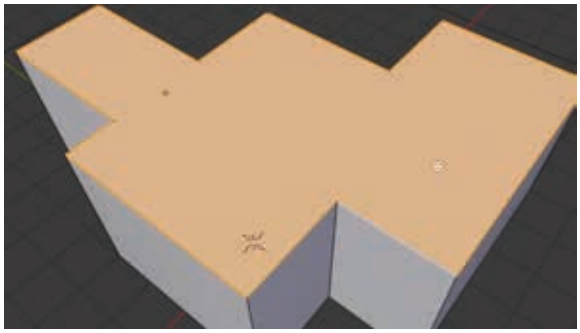
Extruding complex shapes

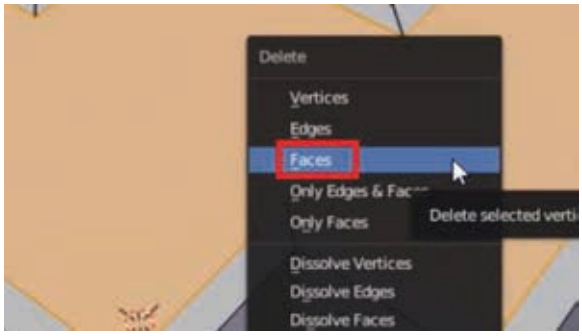
Extruding and scaling these edges creates some problems...



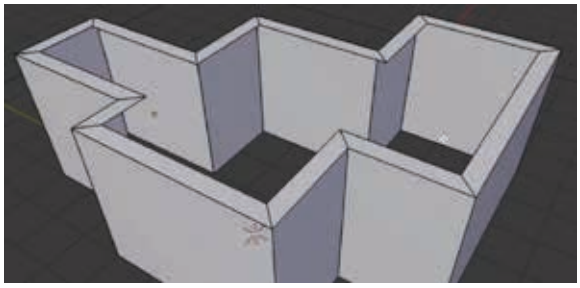


Instead, fill (**F**) and then inset (**i**) and delete the resulting N-gon (**X**).

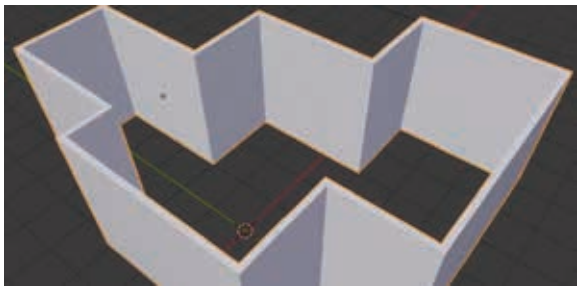


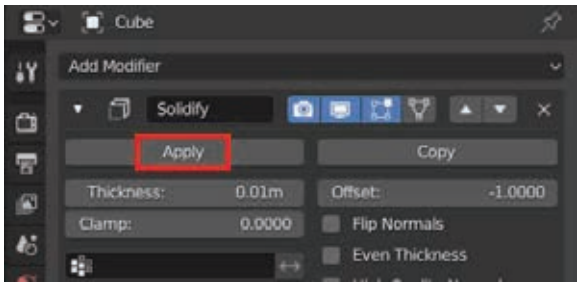


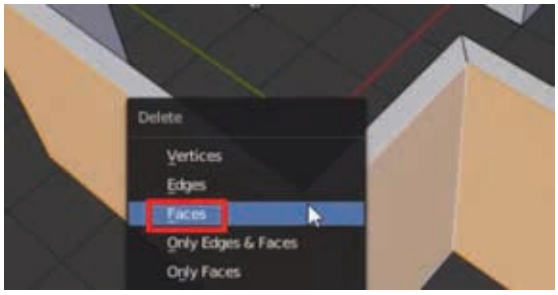
Now you've got perfectly inwards extruded edges.

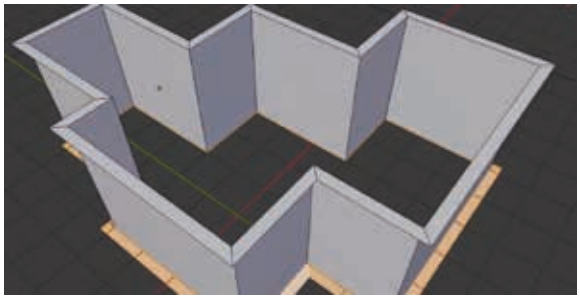


For the other direction, add a Solidify modifier, **Apply** it and delete unnecessary faces.

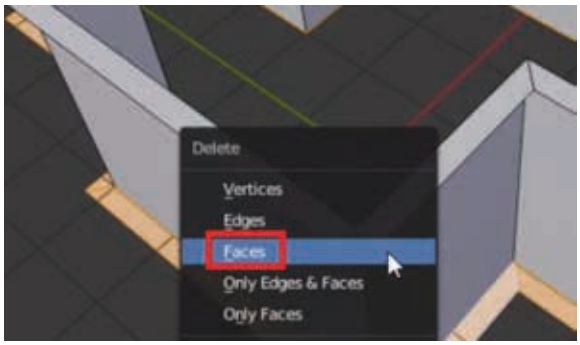






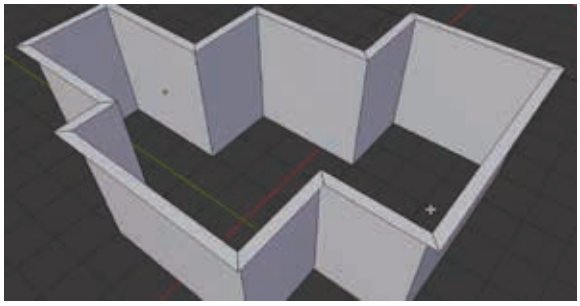


Next, select the remaining bottom edge of faces and delete it by **pressing X**.



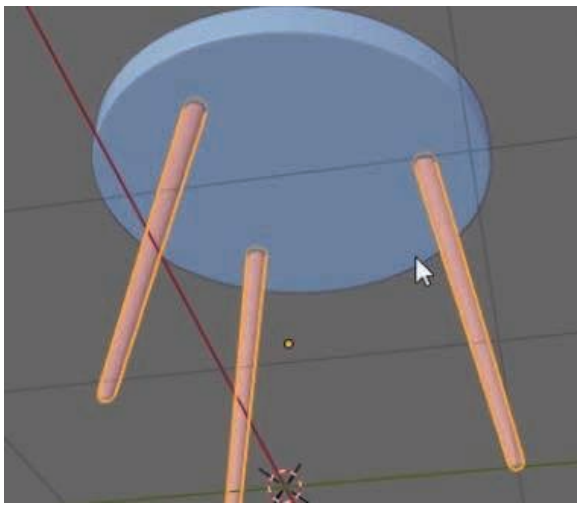
This is how you create inner and outer extrusions of edges from a complex shape.

This can be very useful in Architectural modeling for example.

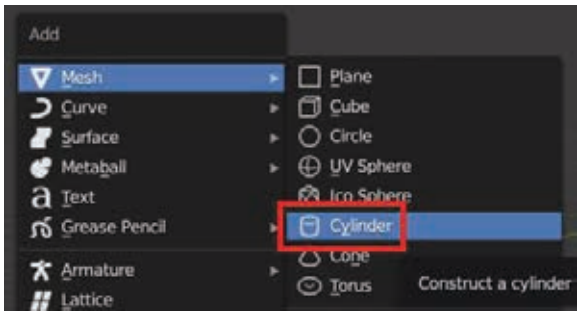


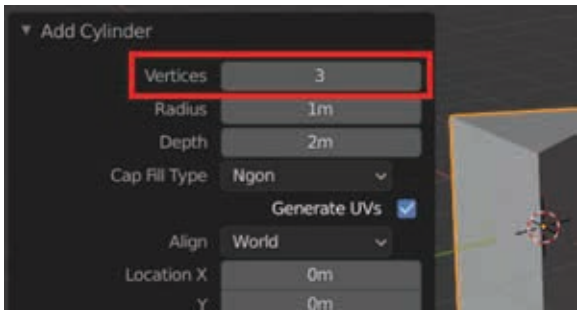
Align three furniture legs

In this tutorial we'll look at creating the legs for a stool or table with only 3 legs, and how to space them perfectly apart.



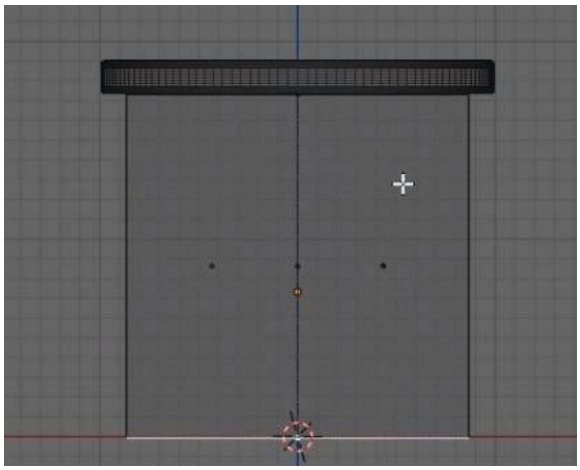
Create a cylinder and set the vertex count to 3.



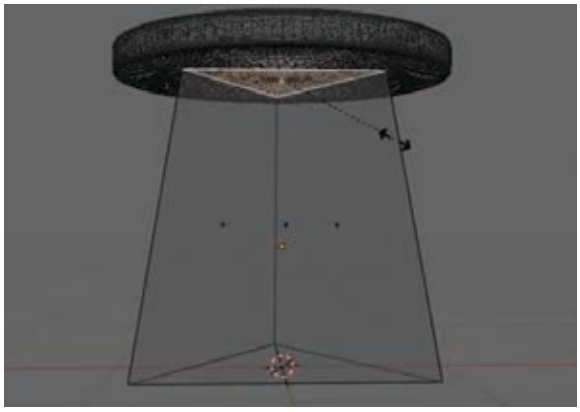


Let's assume you've already got the table top (a flattened cylinder with some beveled edges in this case).

Move and scale the new cylinder we'll use for the legs until the edges are aligned the way the legs should end up.

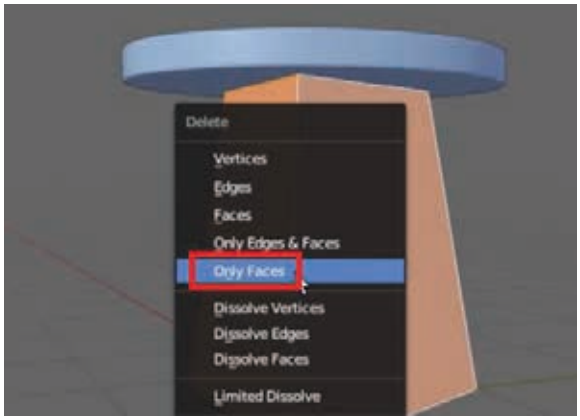


Scale down the top face so the cylinder becomes narrower towards the top.

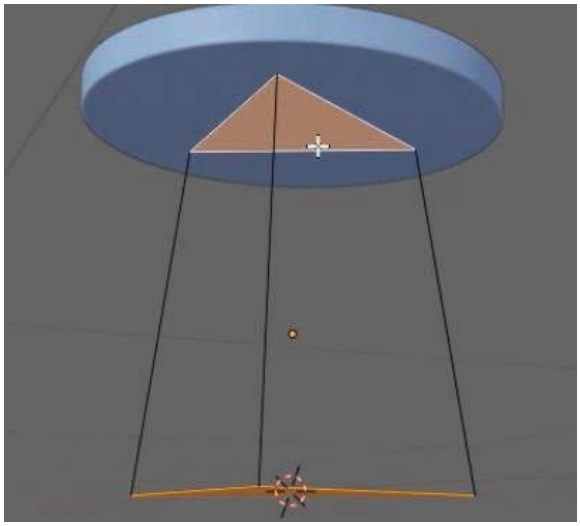


Select the side faces and **press X**, choose **Only Faces**.

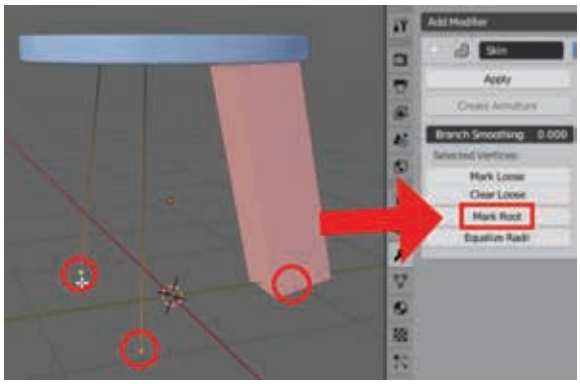
This will remove the faces but leave the edges.

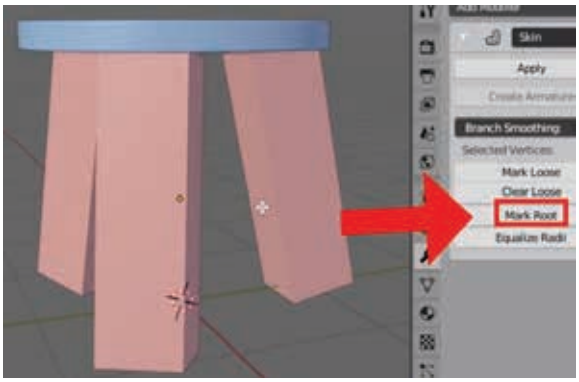


Select the top and bottom faces and **press X**, choose **Faces**, to delete those.

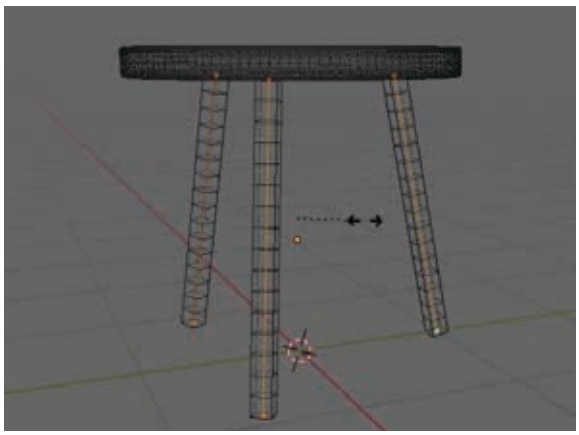


Add a **Skin modifier**and select the 3 bottom vertices, click **Mark Root**. This makes sure all three edges get thickness.

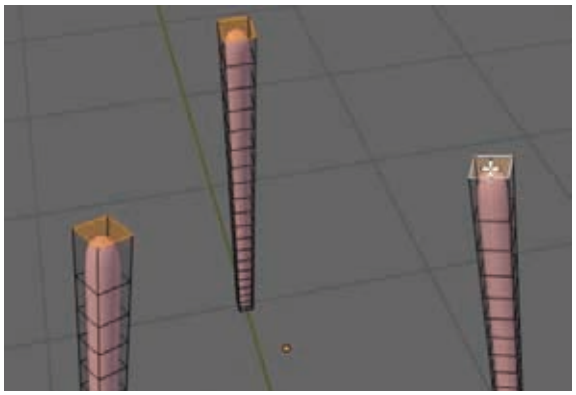


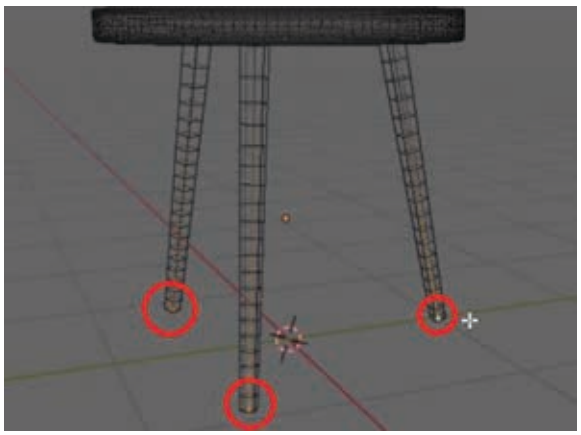


Select all the vertices of the legs with **A** and with **Ctrl+A** scale them down to get the thickness you want. Then select the bottom three vertices and scale them, to get a tapered look.

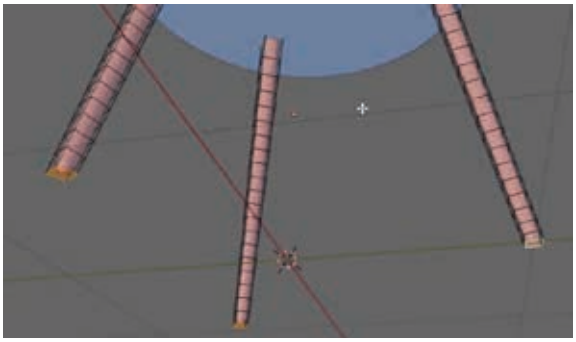


Add a **Subdivision surface modifier**. Select and delete the top faces.





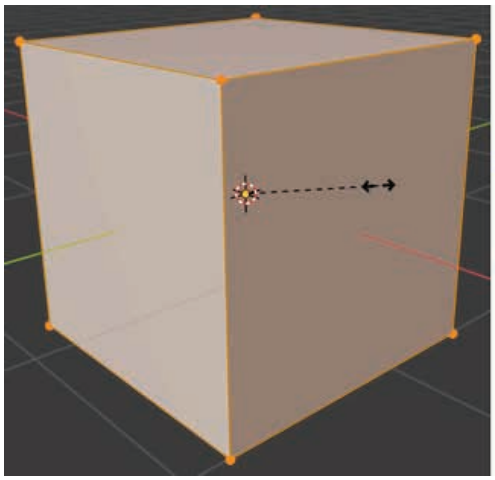
Select the bottom faces and **press S+Z+0** to make them flat (scaled to zero on the Z-axis).

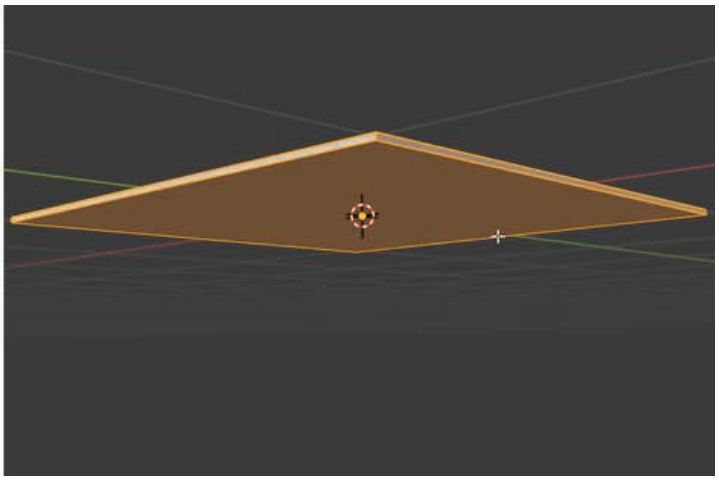


Easy Pillow using the Cloth modifier

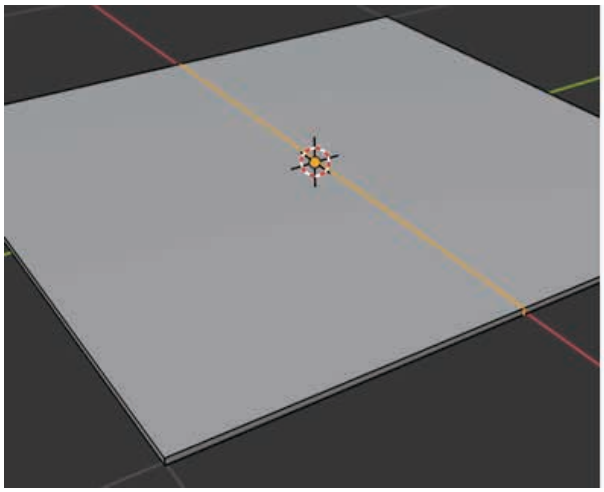
Using the **Cloth modifier** and the **Cloth Brush**, it's easy to create a pillow in just a couple of minutes.

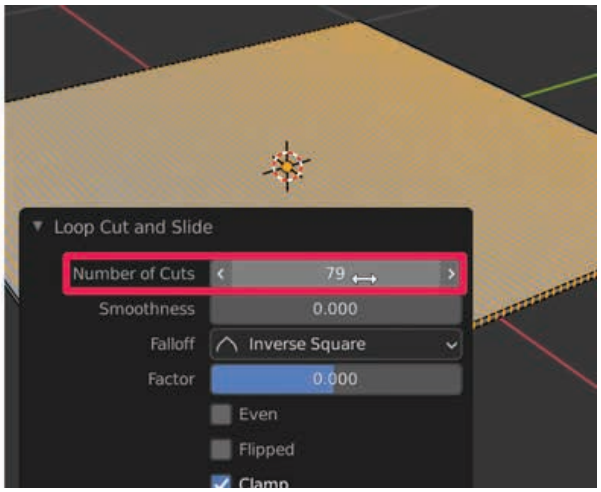
In **Edit Mode**, scale the Cube down by **pressing S+Z+0,01 on the numpad**. This scales the cube down to a very thin slab. Do not do this in Object Mode, as that will affect the scale.





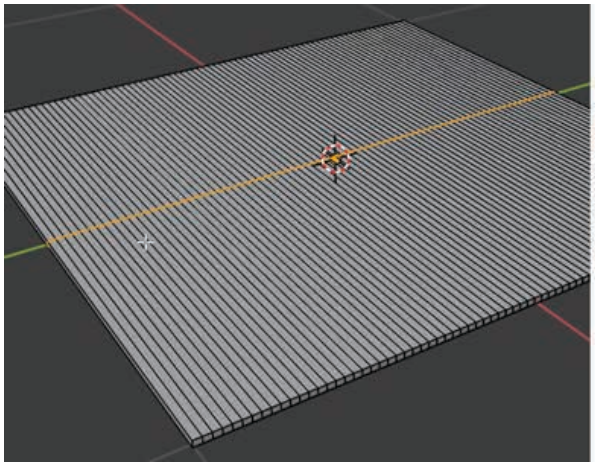
Add Edge Loops with **Ctrl+R** and increase the number of cuts. If you accidentally click somewhere and lose the option panel for the edge loops, **press F9** to bring it back.

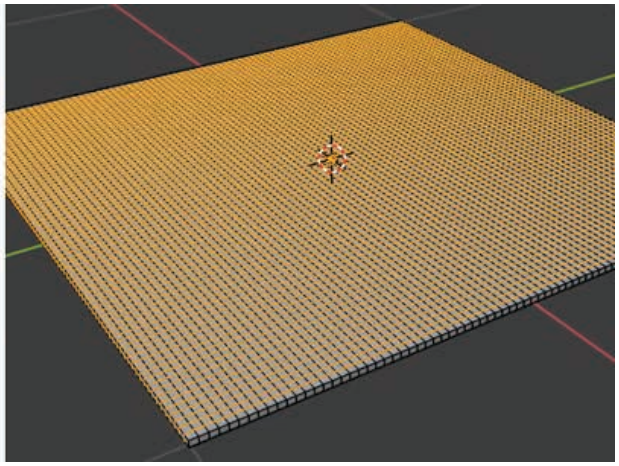




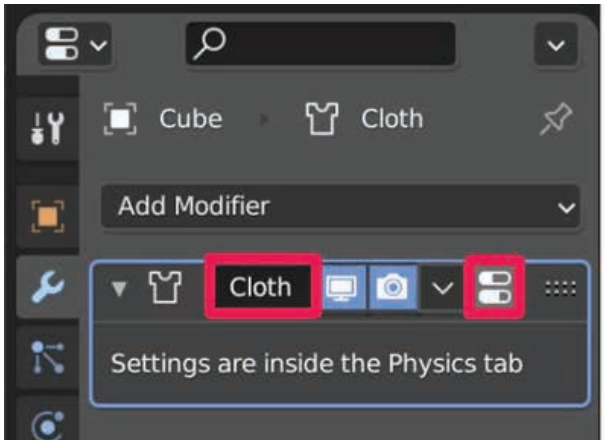
Click here to return to the index.

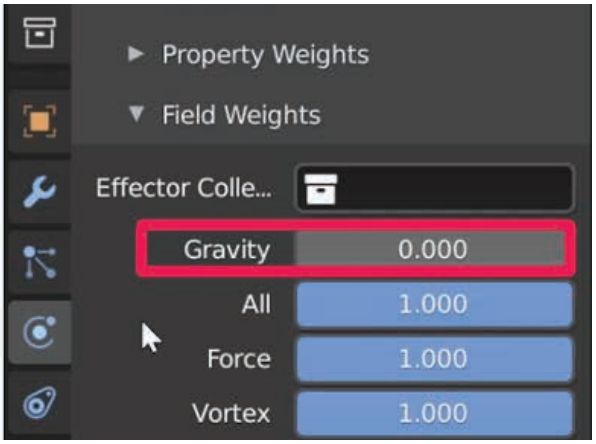
Repeat for the other direction so that the Cube is evenly subdivided.



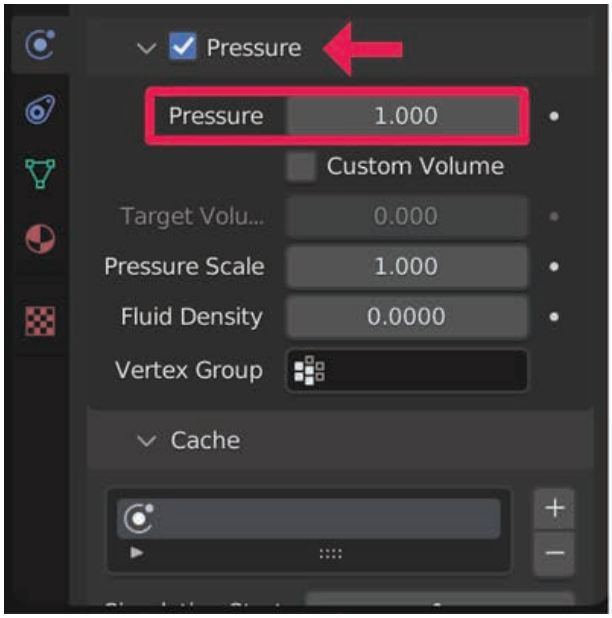


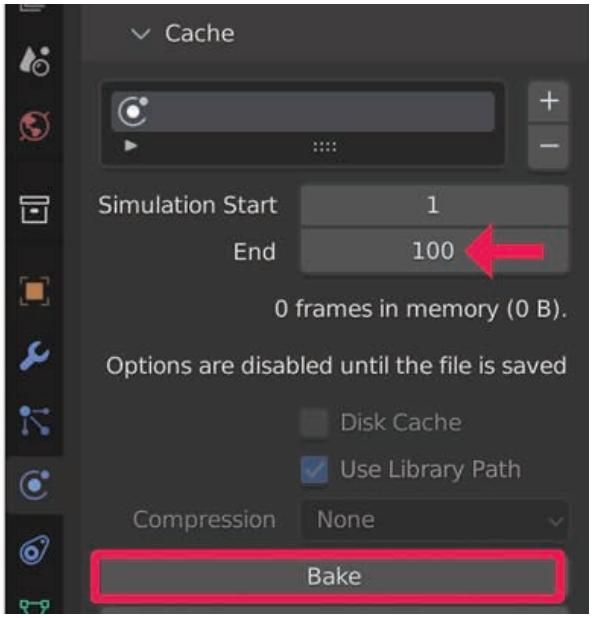
Add a **Cloth modifier** and go to the properties. Under **Field Weights** set **Gravity** to zero. This will ensure that the pillow doesn't fall down once we run the simulation.



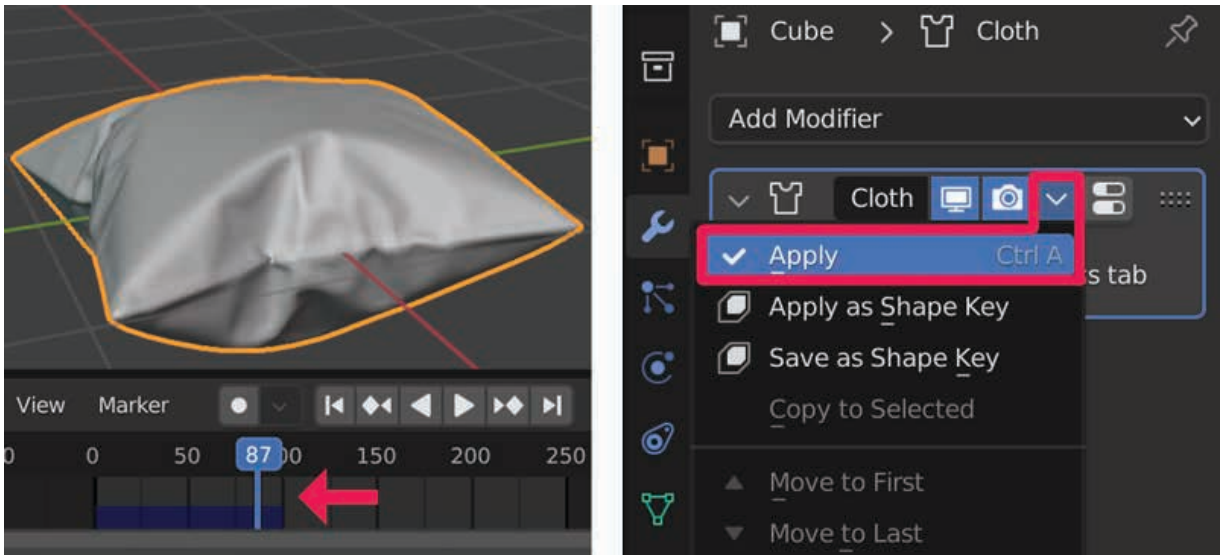


Enable **Pressure** and set it to **1**. Press the **Spacebar** to play the simulation. Alternatively you can also click on **Bake** in the Cache settings and wait for it to finish. To save time, you don't need to bake more than 100 frames to get a nice pillow.

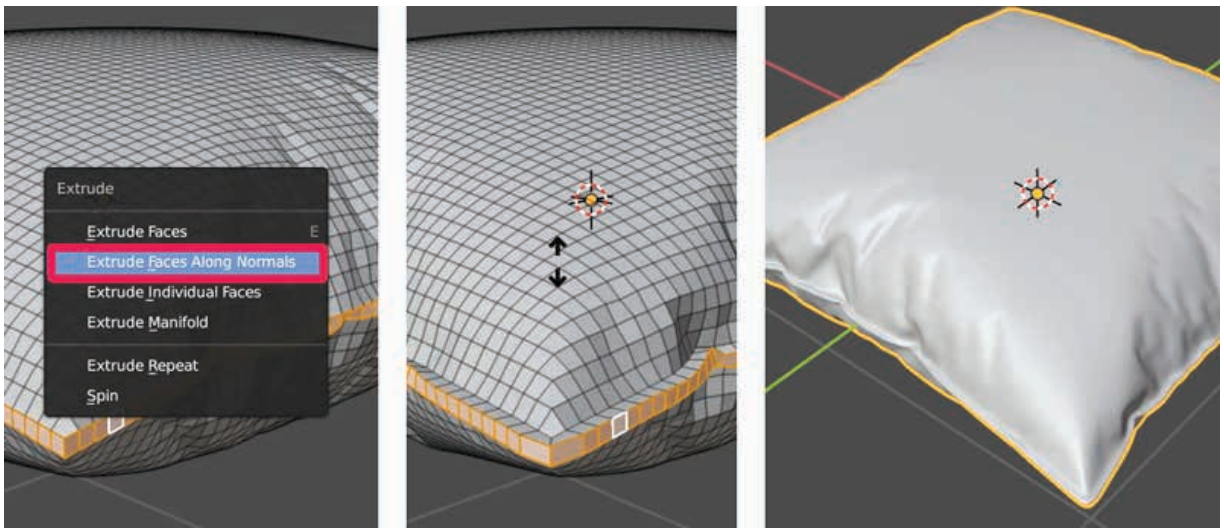




Find a frame on the **Timeline** where the pillow looks good, select the pillow and apply the Cloth modifier by choosing **Apply** from the drop-down in the modifier. Alternatively you can **press Ctrl+A** in the 3D viewport and choose **Visual Geometry to Mesh** to apply the modifier.

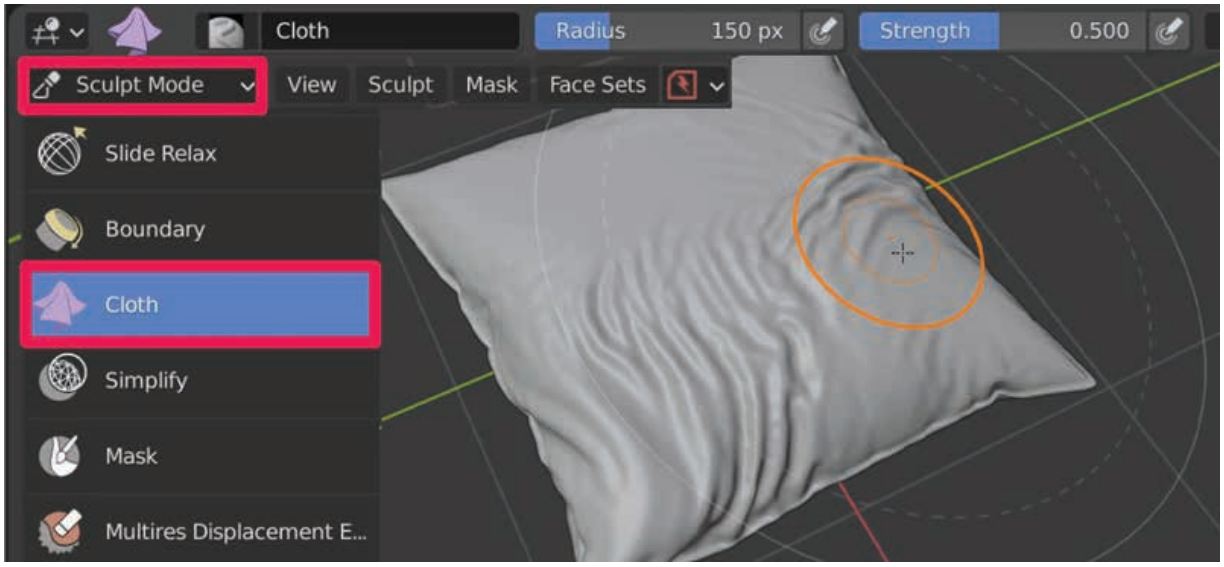


In **Edit Mode**, **press 3** for Face Selection Mode. **Hold Alt and click** on a Face in the middle to select the row of Faces around the pillow. **Press Alt+E** and choose **Extrude Along Normals**, then **press 0.02 on the numpad**. This way we create a subtle decorative edge. In Object Mode, add a **Subdiv modifier** to make it look smooth and soft. You may want to right-click and choose **Shade Smooth** as well.



Using the **Cloth Brush** in Sculpt mode, you can add some more wrinkles. Note that if you scaled the cube down in Object Mode initially, you'll have to apply the Scale by **pressing Ctrl+A** and choosing **Scale**, otherwise the Cloth Brush won't work properly.

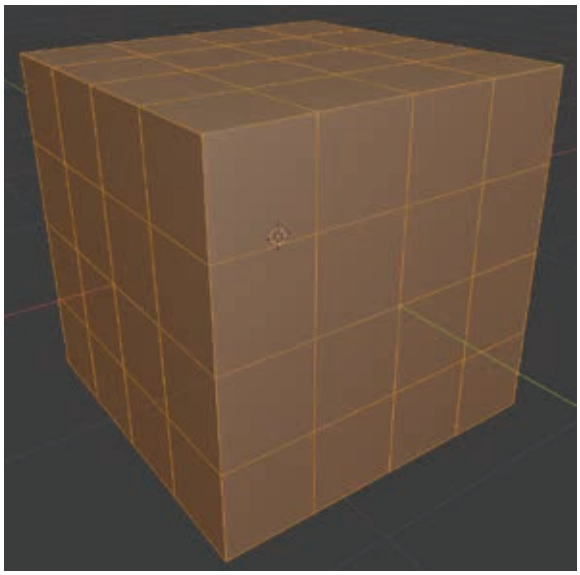
The **Grab brush** is another useful brush to use here, to reshape the pillow.



How to make diamond tufting patterns

In this tip we'll make the most comfortable-looking default cube ever.

Select the cube and **press TAB** to go into Edit Mode.

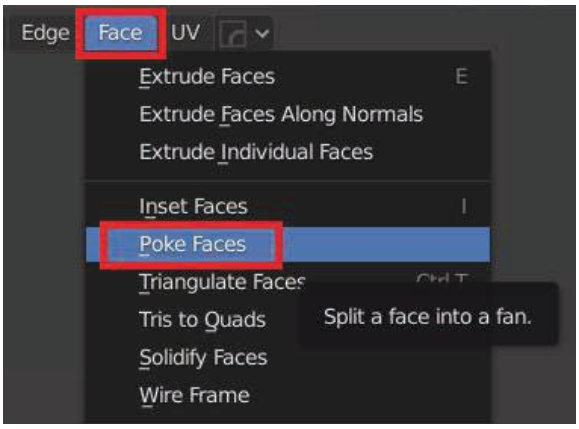


Press A to make sure all the Faces are selected.

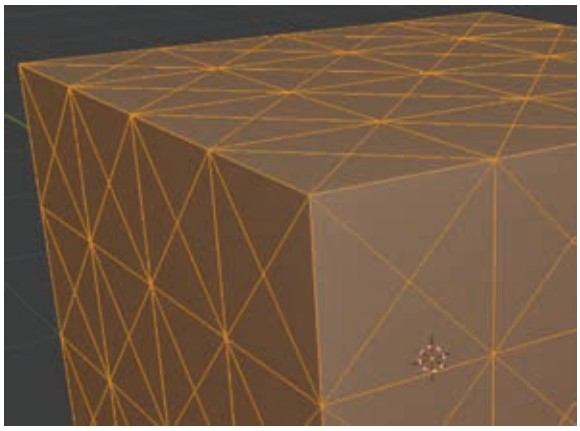
Right-click and choose **Subdivide**.

Repeat that by **pressing Shift+R** (this shortcut always repeats the last action). Now you've subdivided the cube twice.

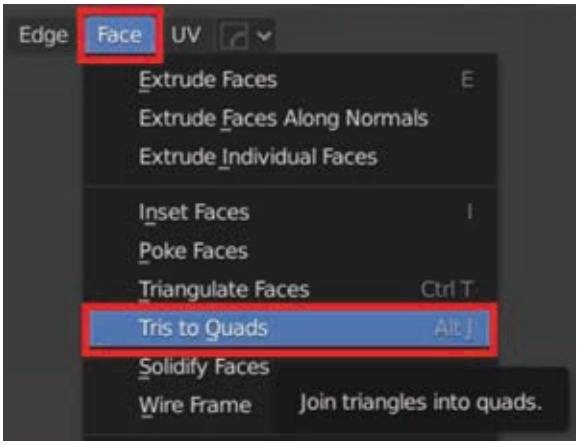
Go to the Face menu and choose **Poke Faces**.

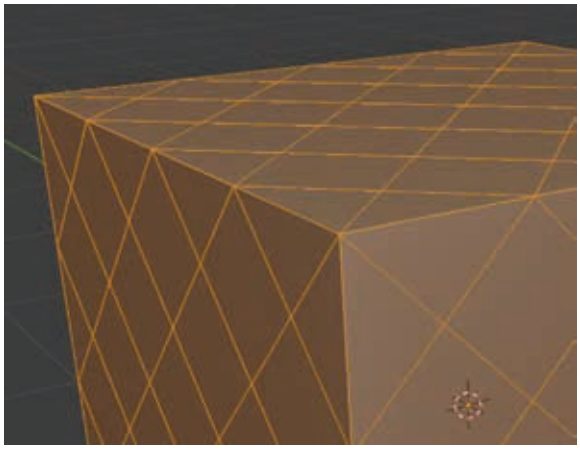


It will now look like in this picture:

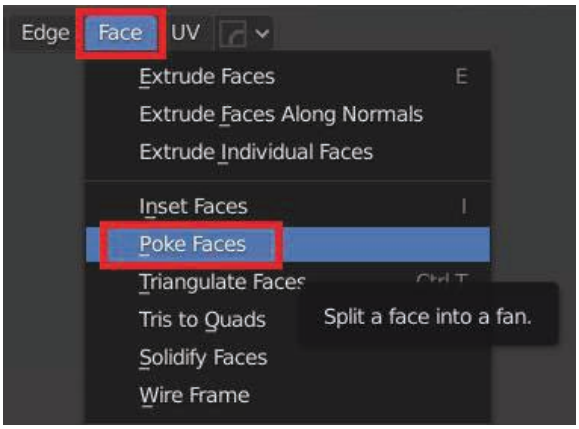


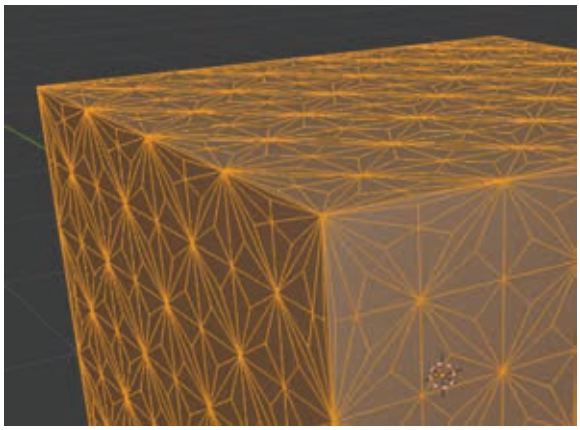
After that, go to the Face menu again and choose **Tris to Quads** (or **press Alt+J**). It will now look like the picture below right:



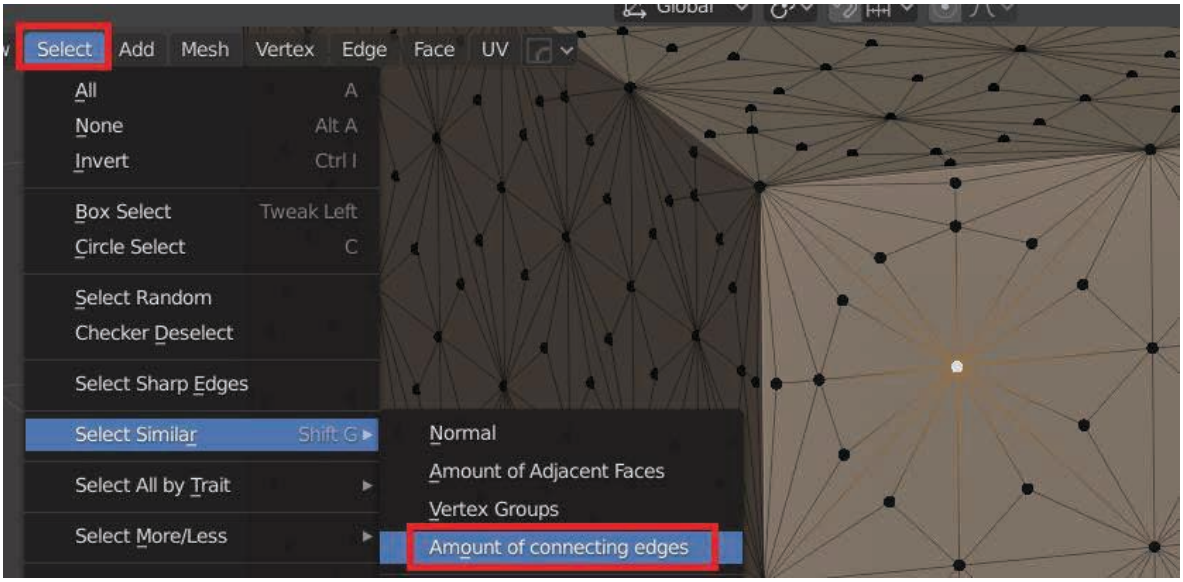


Go to the Faces menu again and choose **Poke Faces** and **press Shift+R** to repeat. It will look like this:

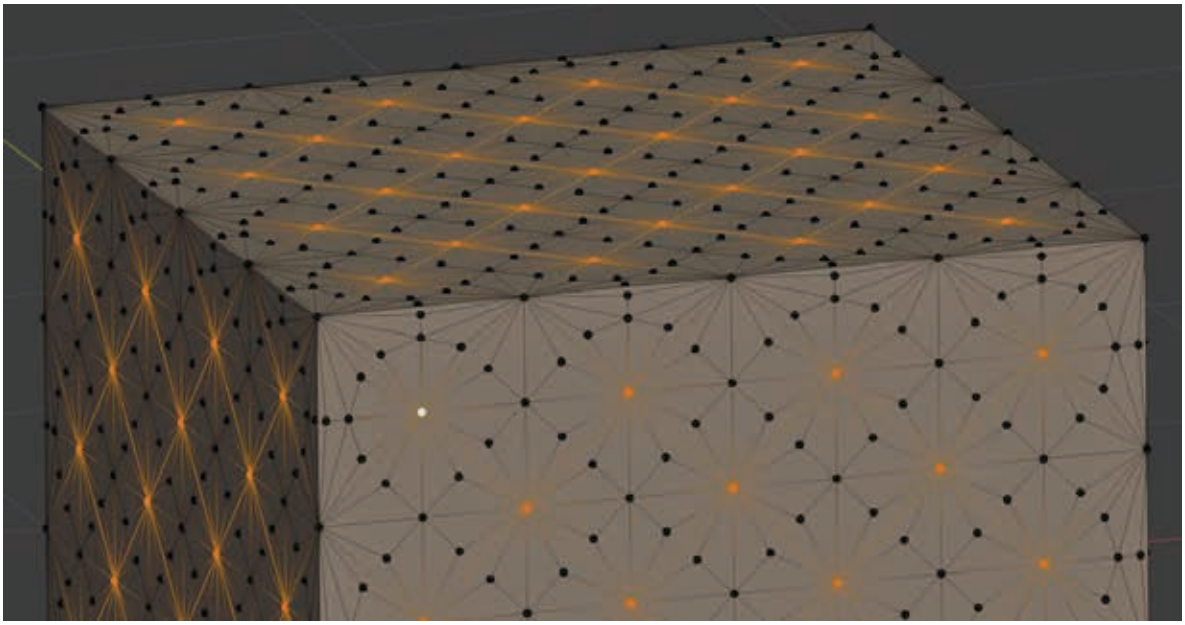




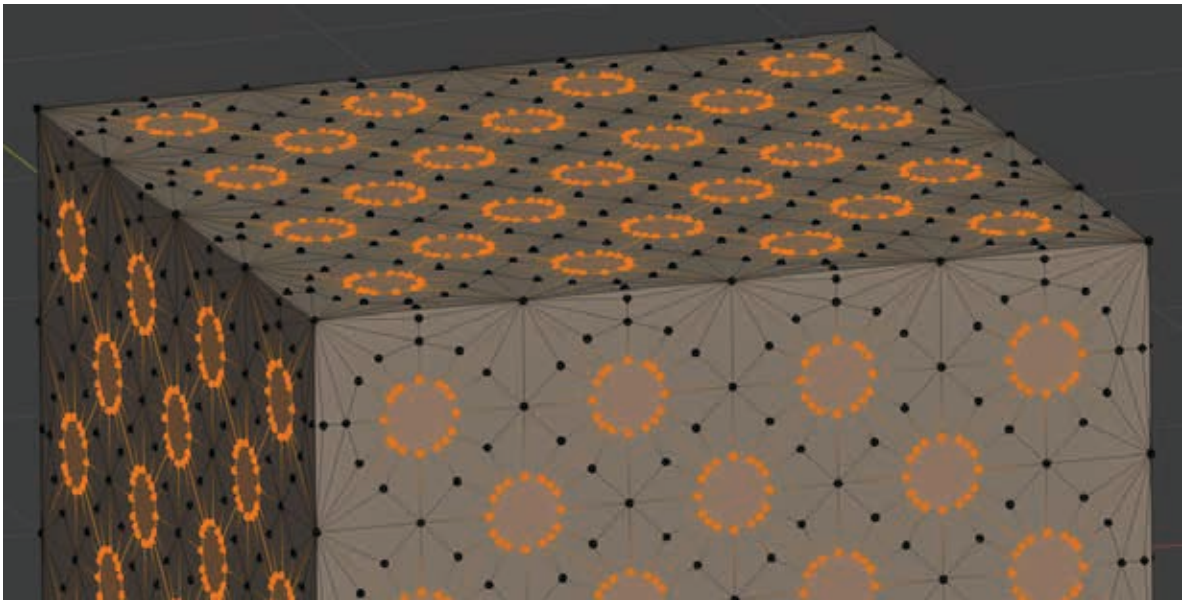
Press 1 on the keyboard (not on the numpad) for Vertex Selection Mode and select one of these vertices that are at the centre of 16 edges, then go to Select > Select Similar > **Amount of Connecting edges**.



The selection should now look like this on all sides of the cube:

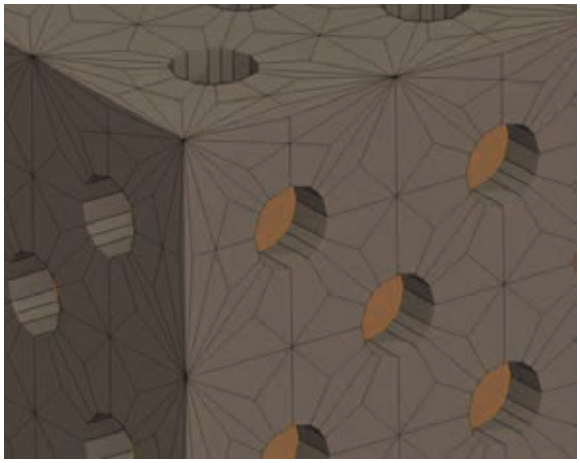


Press **Shift+Ctrl+B** to bevel the selected vertices. Drag the mouse while **holding Shift** to have more fine control over the beveling action.

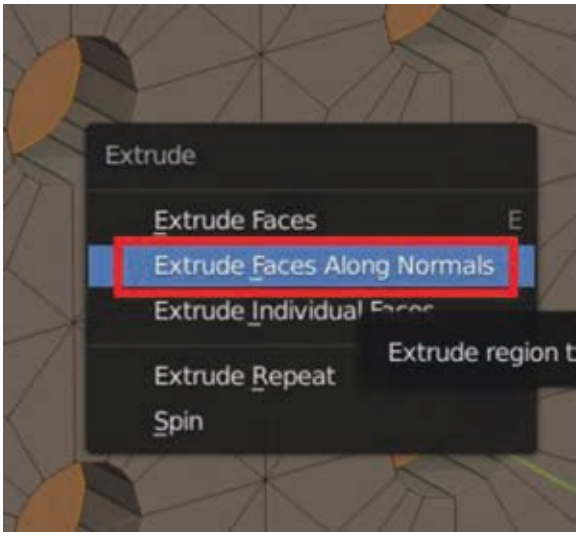


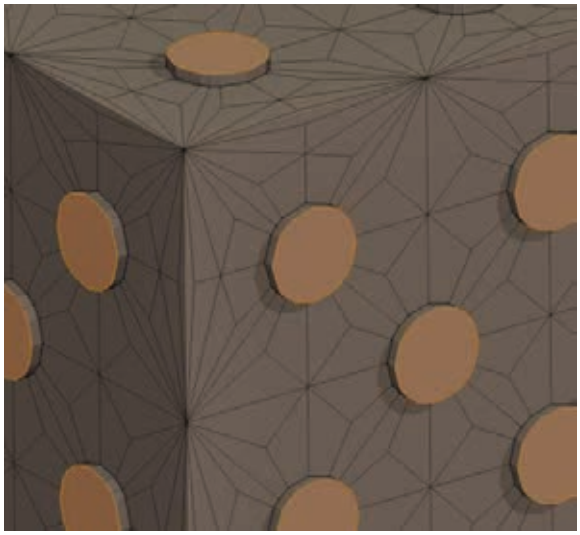
When the vertices have been beveled to around the size shown here, **press 3** on the keyboard (not on the numpad) to switch to **Face Selection Mode**.

press Alt+E to extrude and choose "**Extrude Along Face Normals**". Using this method of extrusion, you can move the faces inwards. Don't deselect the faces yet.

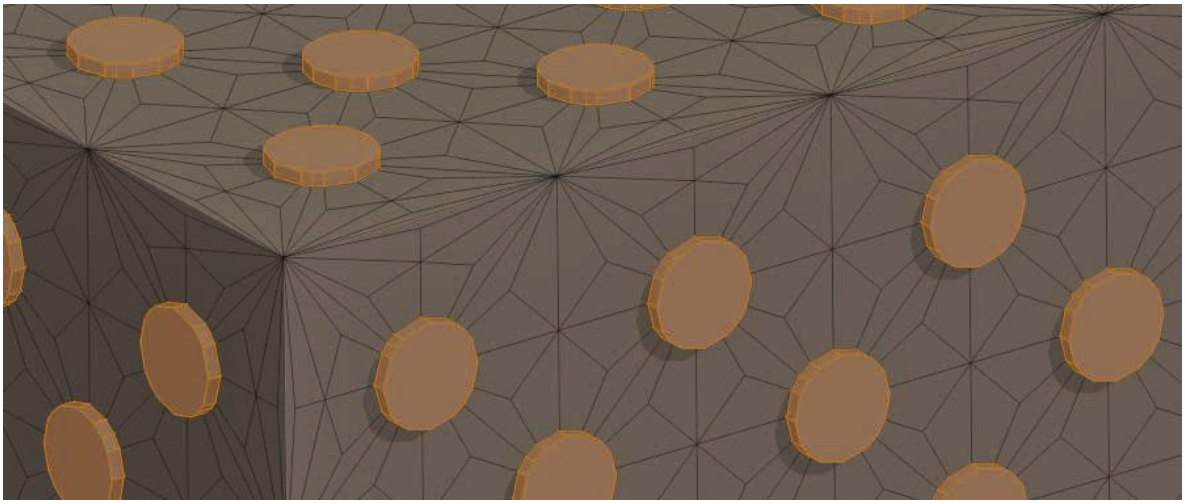


Repeat the extrusion by **pressing Alt+E** and choosing "**Extrude Along Face Normals**" but this time extrude them to slightly outside of the cube. These will become the "buttons" on our cushion-like default cube.

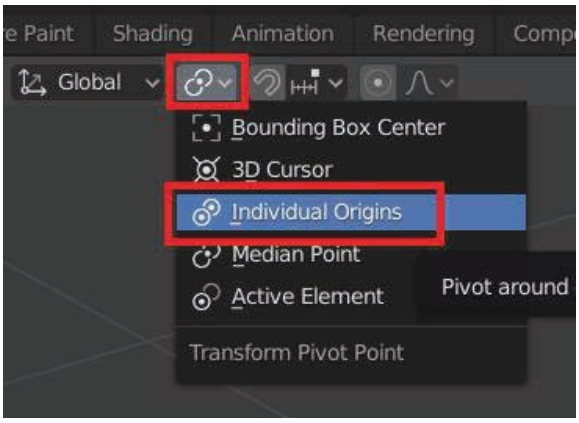




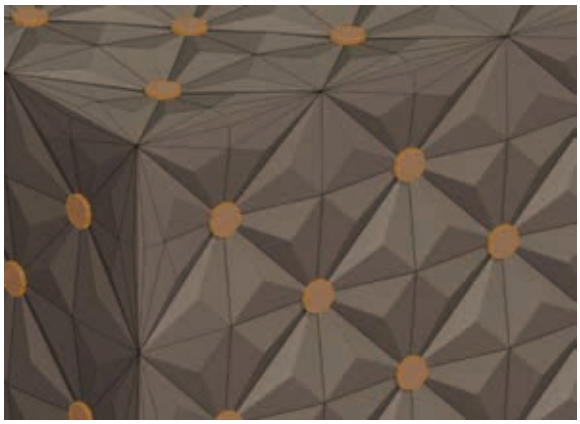
Press Ctrl-[+] on the numpad twice to grow the selection.



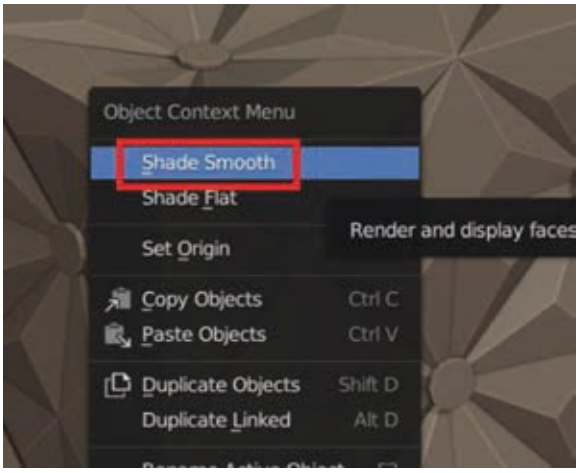
Set Transform Pivot Points to **Individual Origins.**

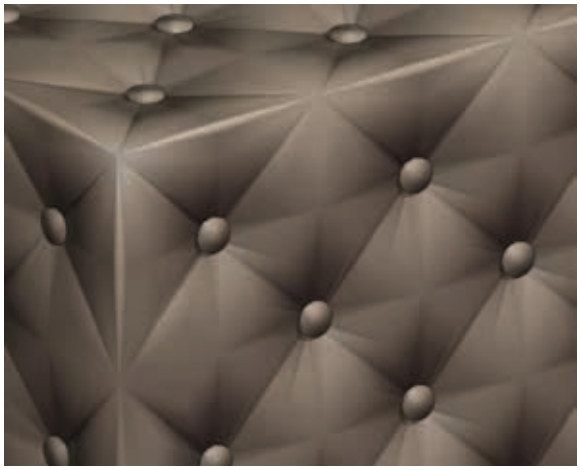


Press S and scale the selection inwards.

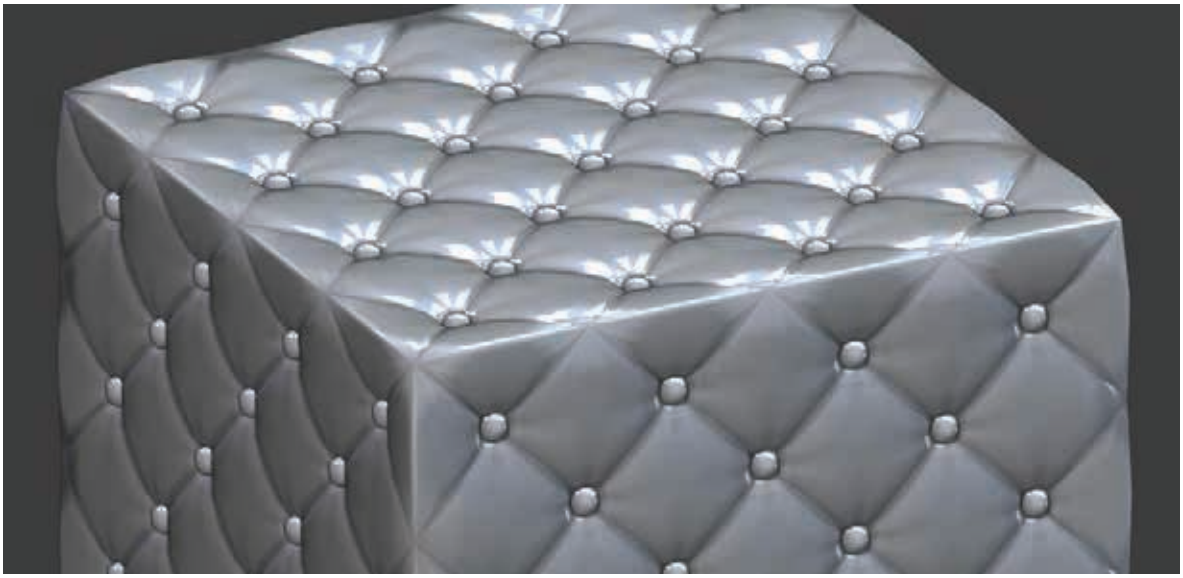


Press TAB to go back to Object Mode. **Right-click** and choose **Shade Smooth**.





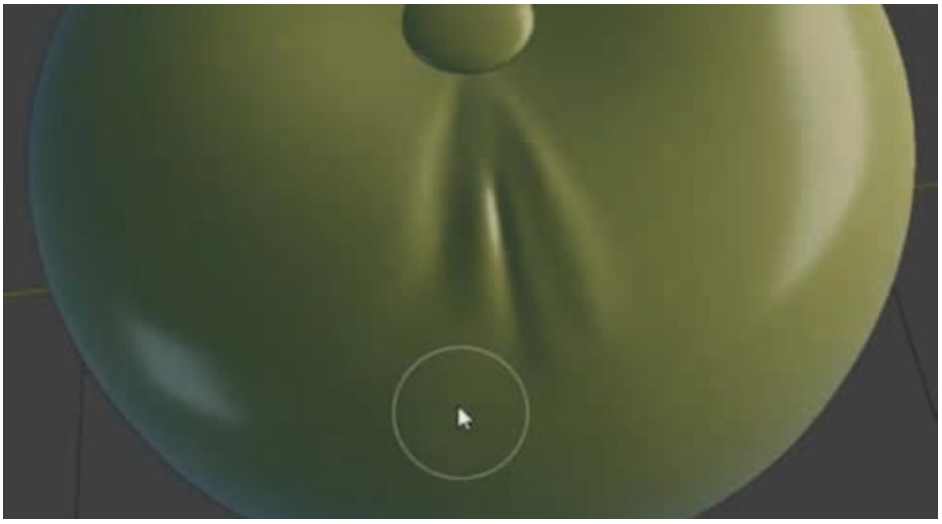
Now it looks nice and smooth (also thanks to a MatCap material).



If you want you can add a Subdivision Surface modifier to make it even smoother by **pressing Ctrl+1**. This can be a good base for sculpting even more detail. You could also try adding some wrinkles with the new Cloth Brush.

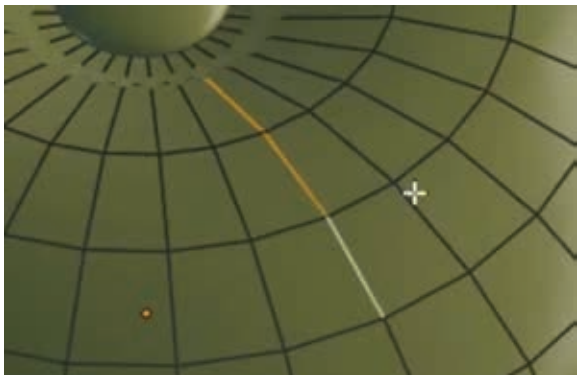
Modeling folds in cloth

Although sculpting can be a very effective way of creating folds, it is actually far from easy, and can be difficult to control. If you want to be more precise, you can model the folds by hand. It's not difficult if you know the trick.



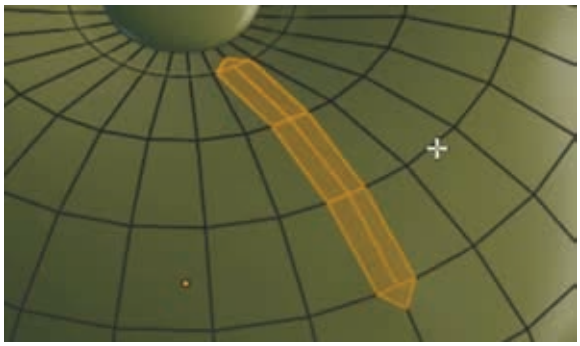
This is a UV Sphere which has a Subdivision modifier on it to make it look smooth.

Press TAB to go into Edit Mode.

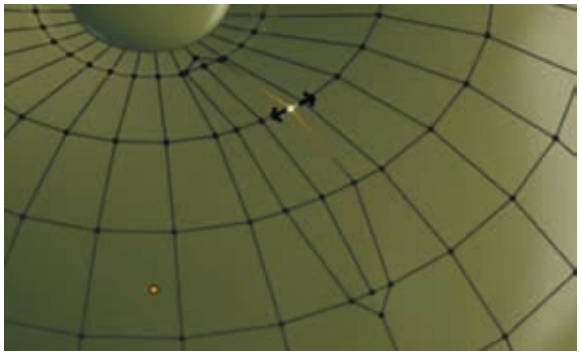


Select some edges and bevel them by **pressing Ctrl+B**.

While beveling, scroll the middle mouse wheel up to create an extra loop in the middle.

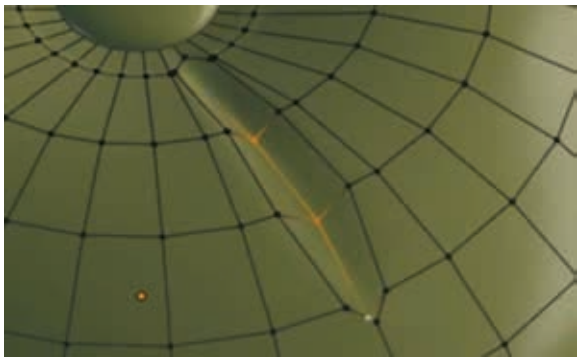


You may want to slide some vertices (by **pressing G twice**) along edges to make for a more natural edge flow.

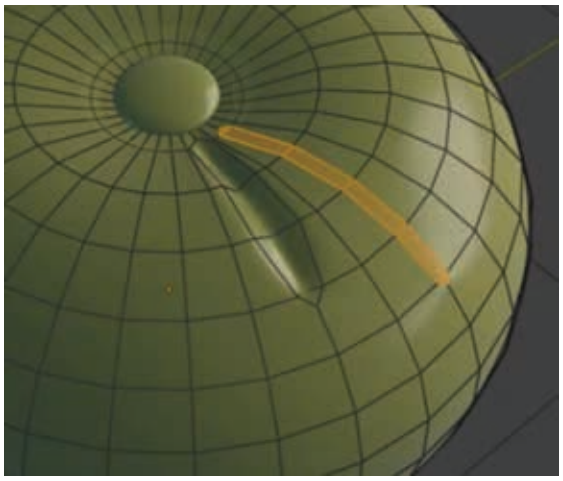


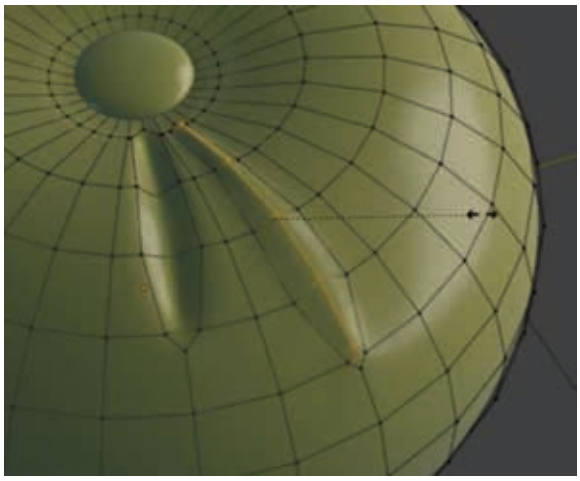
Instead of sliding the individual vertices you can also select two opposing ones and **press S** to scale them away from each other.

Select the middle edges and **press Alt+S** and move the mouse up to scale them inwards along the normal direction.

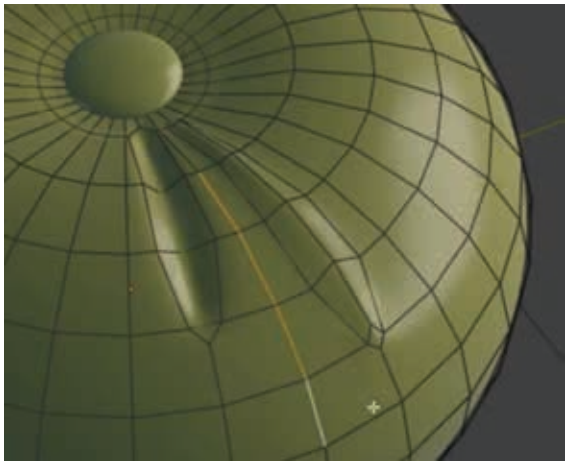


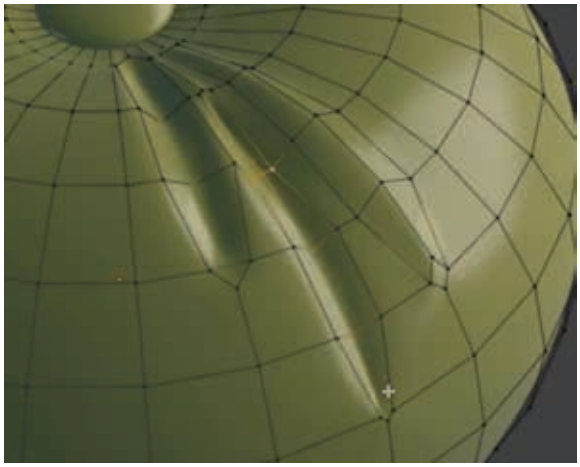
We repeat these steps a few times. Make some folds that go into the object, and some that protrude.



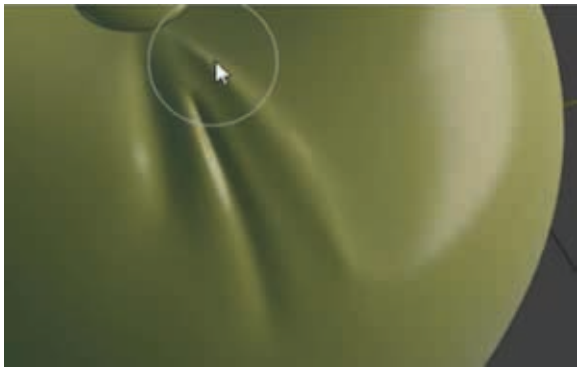


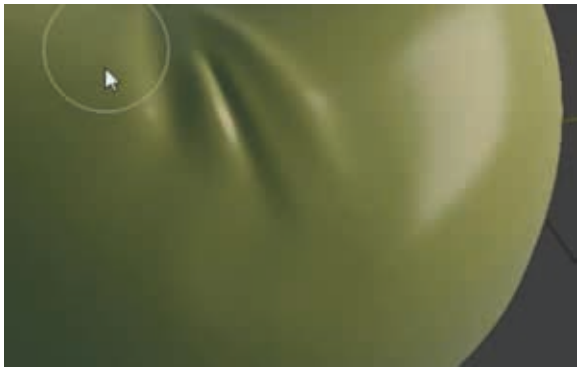
You can use your imagination or look at reference of cloth with folds. Using reference is always a good idea when going for realism.





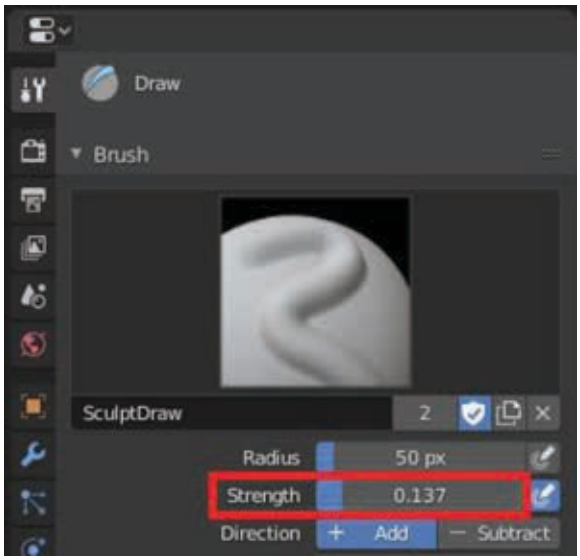
In the end, you can use the **Smooth Brush** in Sculpting mode to make it look a bit more smooth. Make sure the strength of the brush is not set too high. You want a subtle effect.





To get the Smooth Brush, **hold Shift** while sculpting. Or select the **Smooth brush** from the sculpting brushes.

You can set the **Strength** lower to have more control over the smoothing.

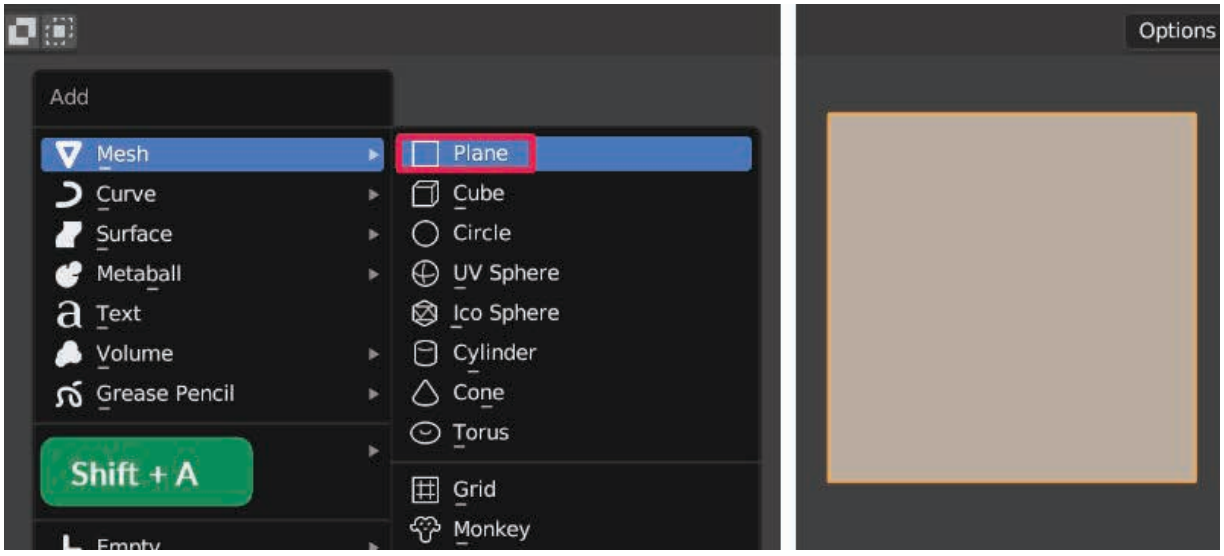


How To Make A Curtain In One Minute

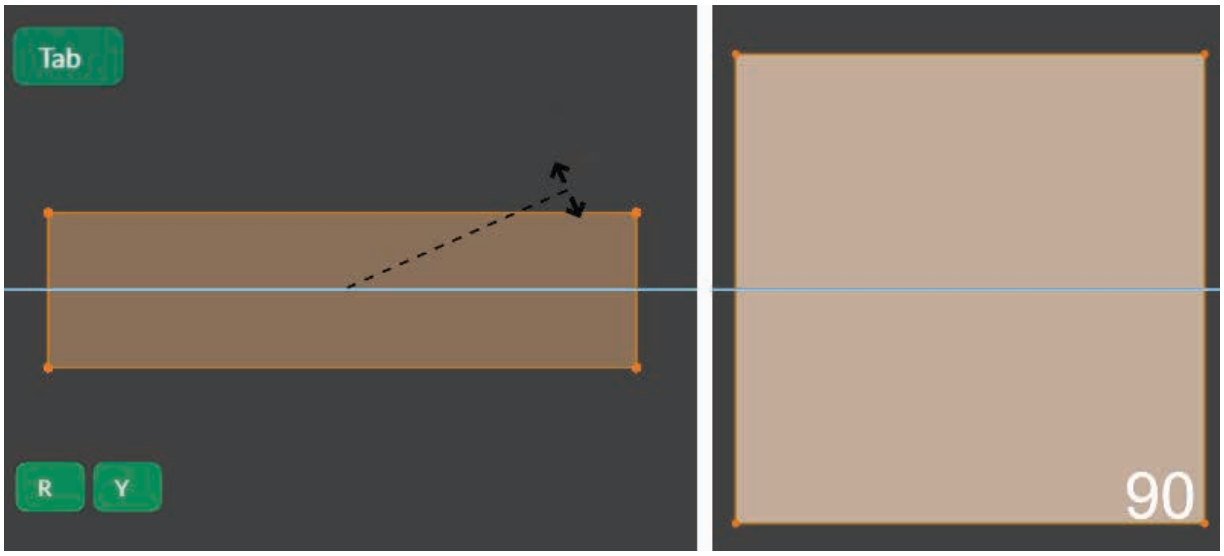
Cloth simulations are not only fascinating to look at, they can help us with modeling shapes that would otherwise be quite challenging.

Let's make this curtain using a cloth simulation:

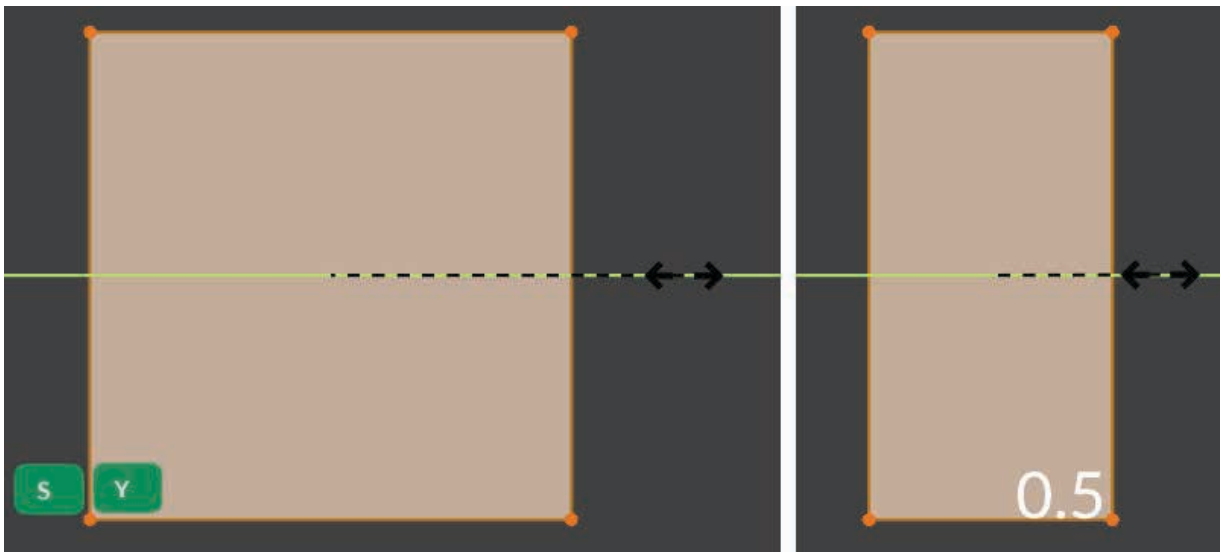
Add a plane with **Shift+A** > Mesh > Plane.



Press TAB to go into Edit Mode. Rotate the plane on the Y-axis 90° by typing **R+Y+90 <Enter>**.

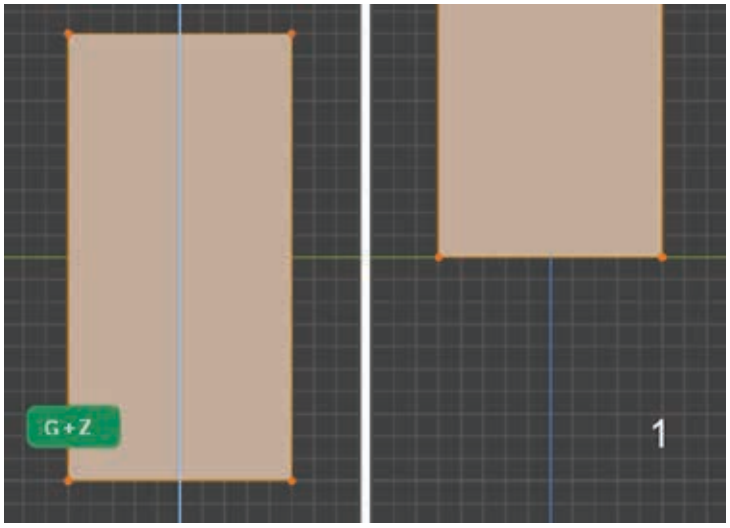


Then scale it on the Y-axis by -50% by typing **S+Y+[-0.5] <Enter>**.



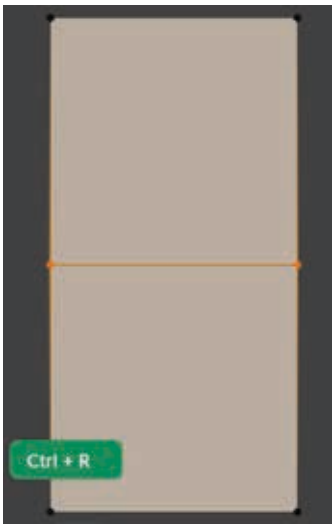
Move it up one unit by typing **G+Z+1 <Enter>**.

Moving it above the grid floor like this, will make it easier to see what we are doing.

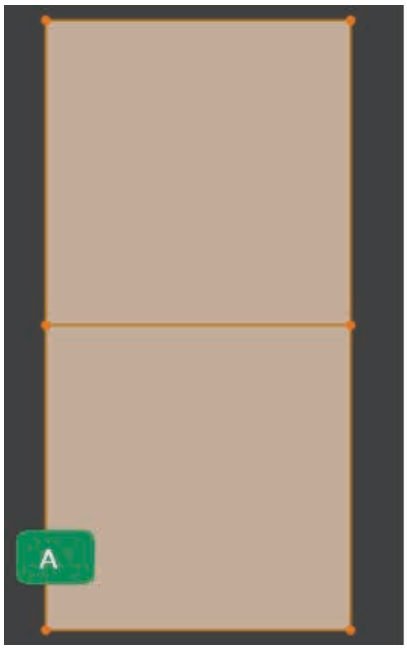


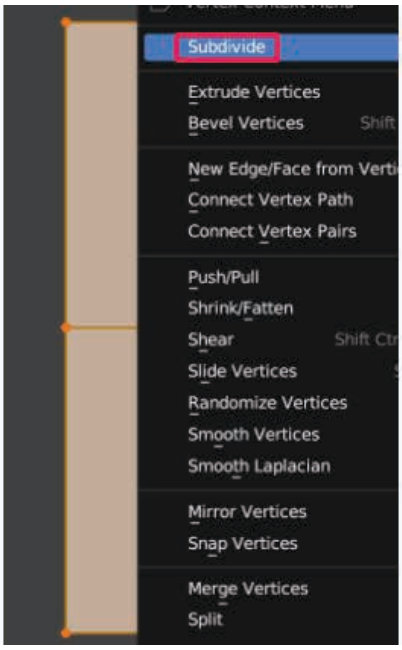
Add an edge loop in the middle by **pressing Ctrl+R**.

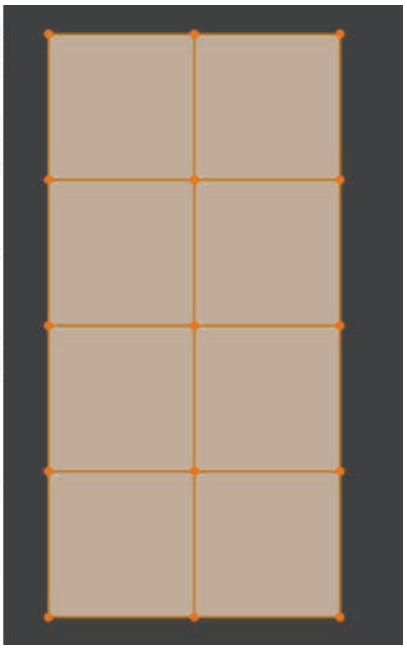
Left-click to confirm, then **right-click** to make sure it doesn't move from its initial position.



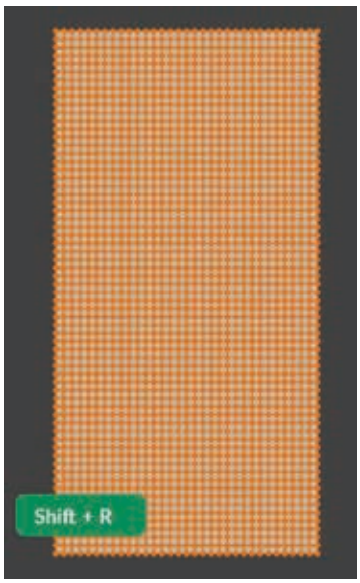
Subdivide the mesh a few times by selecting all (**press A**) then **right-click** and choose **Subdivide**.



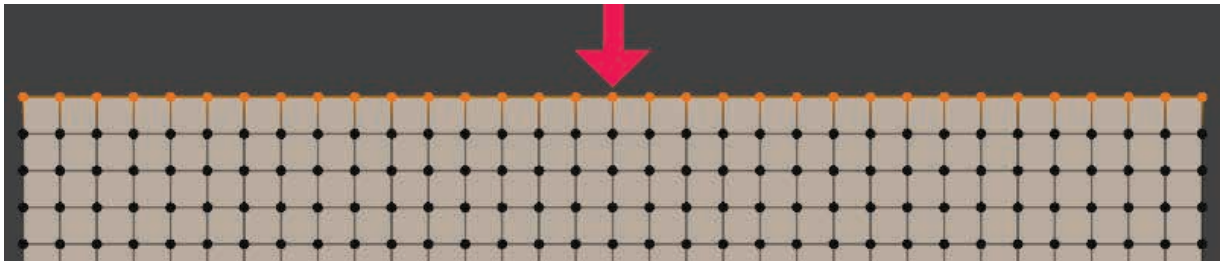




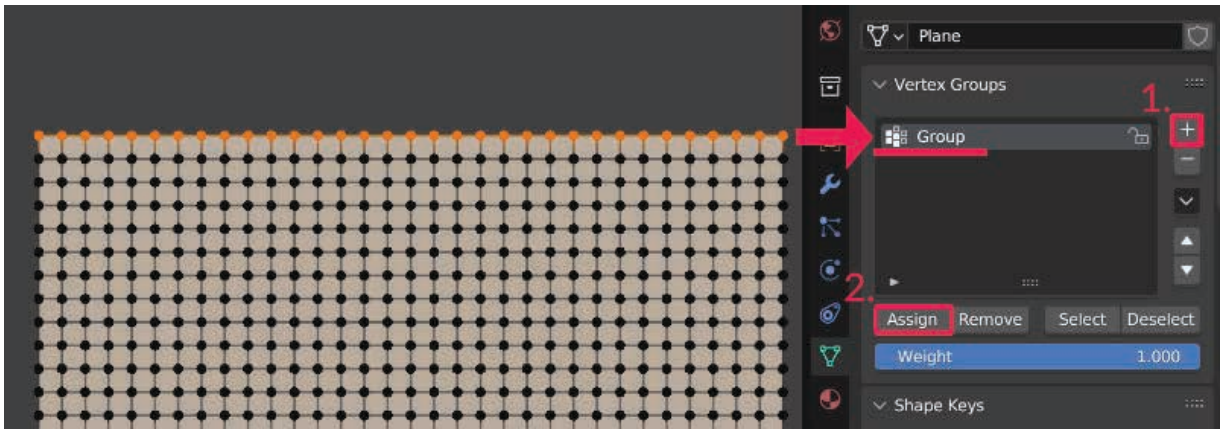
Press Shift+R 4 times to repeat the subdivide step 4 times.



Select the top edge row by **ALT-clicking** on it.



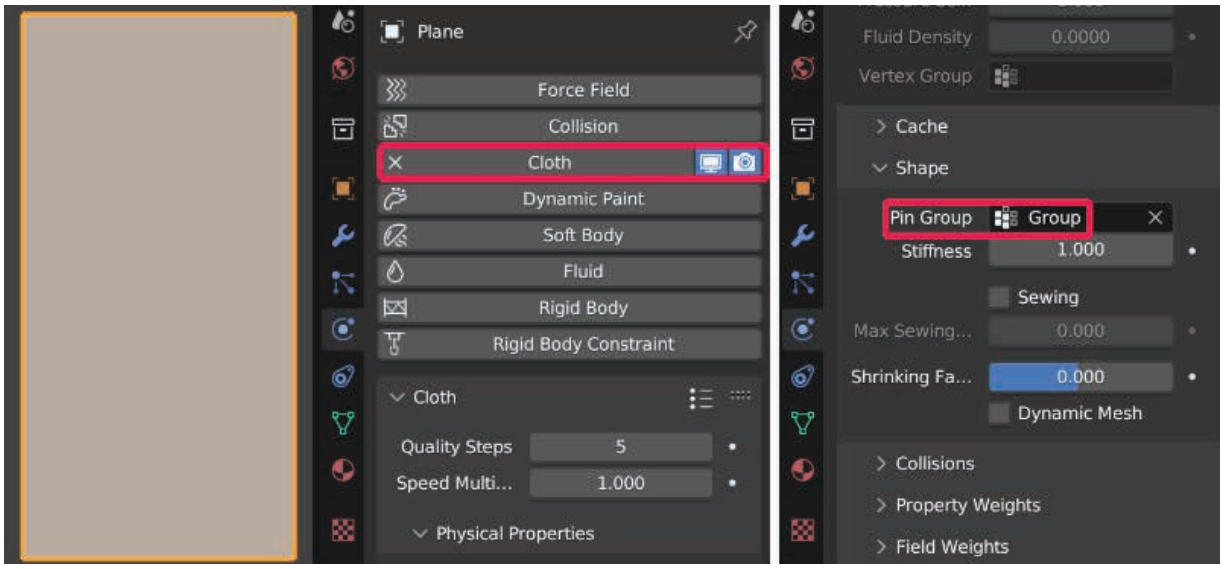
Create a new **Vertex Group** by clicking on the **+ icon**. Click **Assign** to add the selection to the group. This group will be used as a "Pin Group" for the Cloth Simulation.



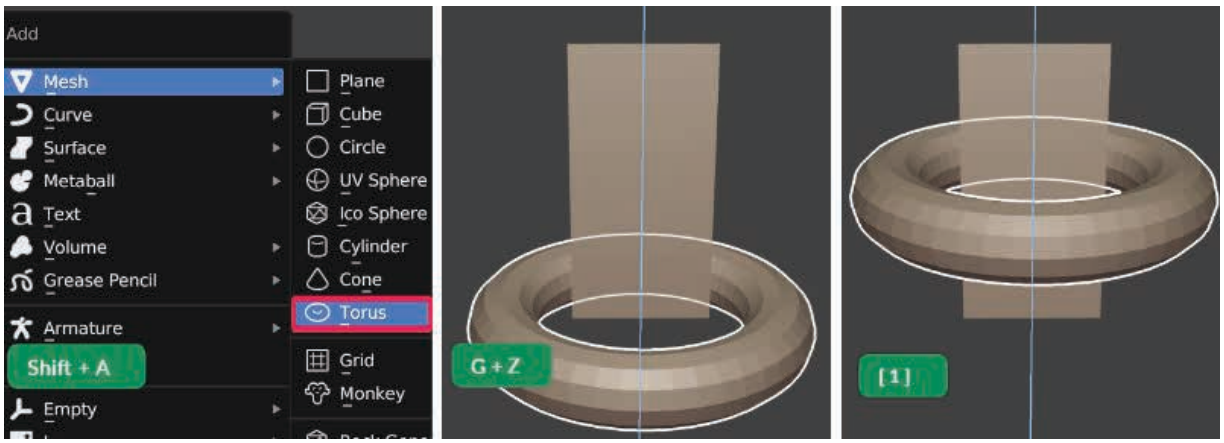
That means that those edges won't be affected by gravity. Otherwise the whole thing will fall down once the simulation starts.

Press TAB to go back to **Object Mode**, and enable **Cloth** physics for the Plane.

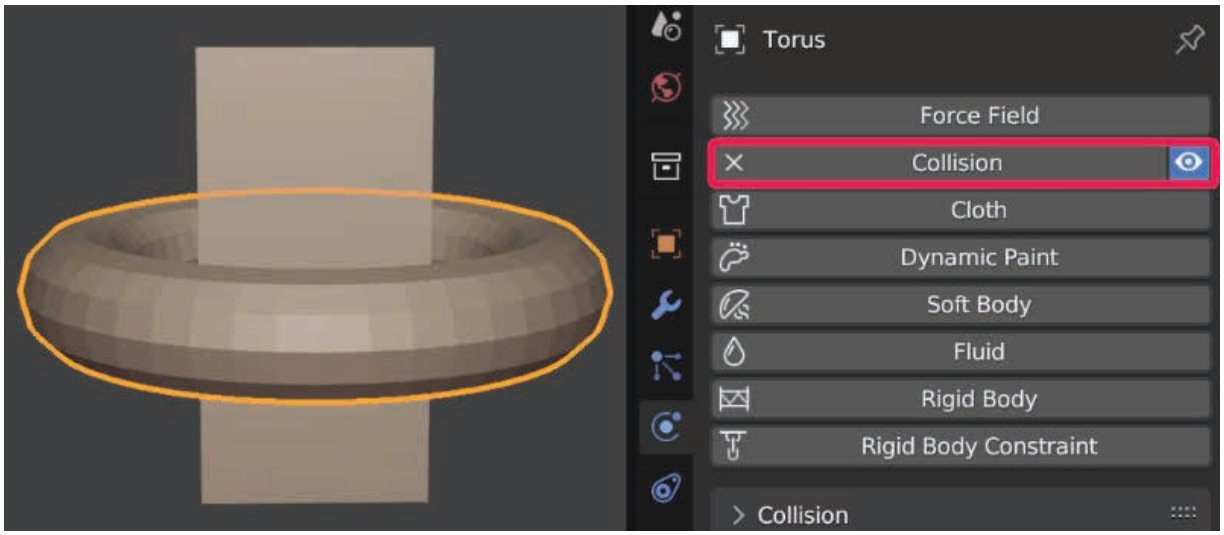
Add the **Pin Group** under **Shape** in the Cloth settings.



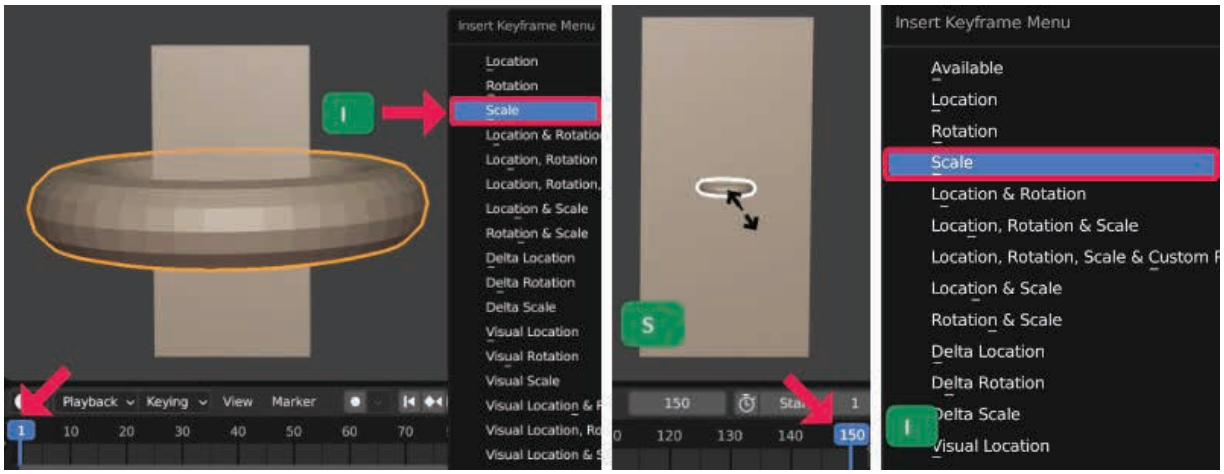
Press Shift+A and choose Mesh > **Torus** to add a torus around the plane. Move it up along the Z-axis to about the middle of the plane. (**Press G+Z+1** to move it up one unit).



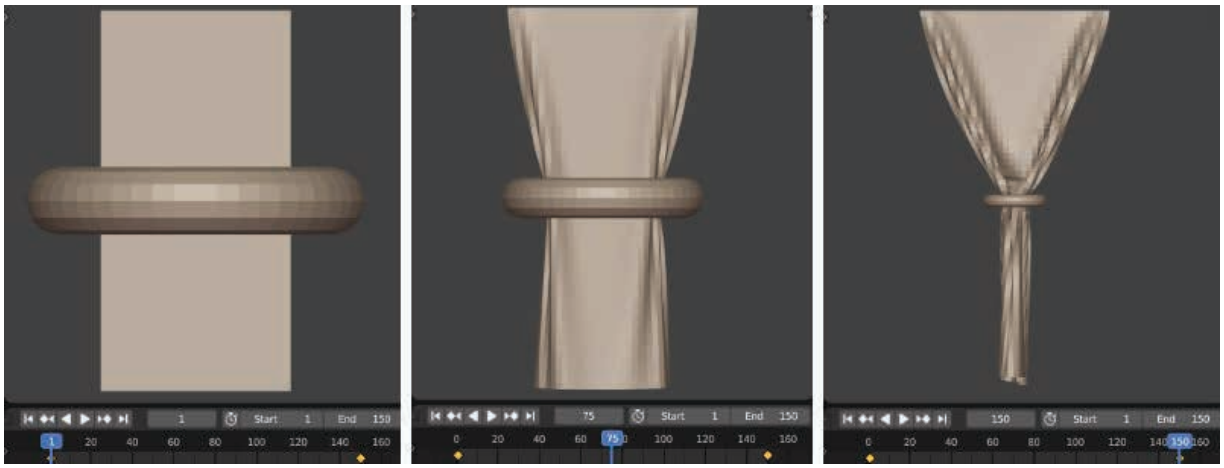
Enable **Collision** for the torus.



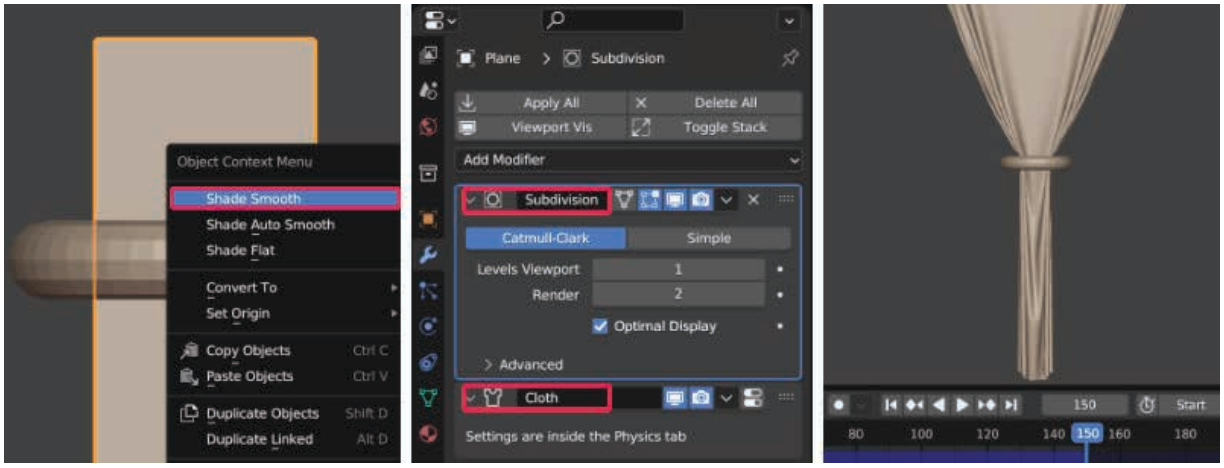
Set a **Scale** keyframe for the torus at frame 1 by **pressing i** and choosing **Scale**. Go to a later frame like 150 and scale the torus down, then set another Scale keyframe.



Press Shift and left arrow to go back to frame 1 on the timeline. **Press Spacebar** to play the simulation. After a while it will give you something like this result.

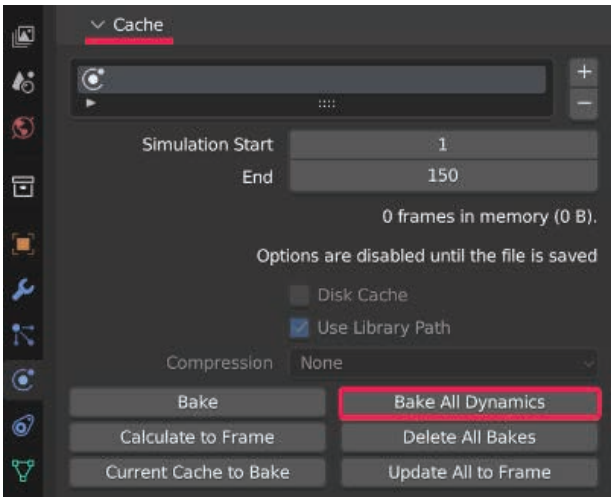


In case you want a more detailed and smooth result, you can right-click and **Shade Smooth** and add a **Subdivision modifier**. Place it above the Cloth modifier.



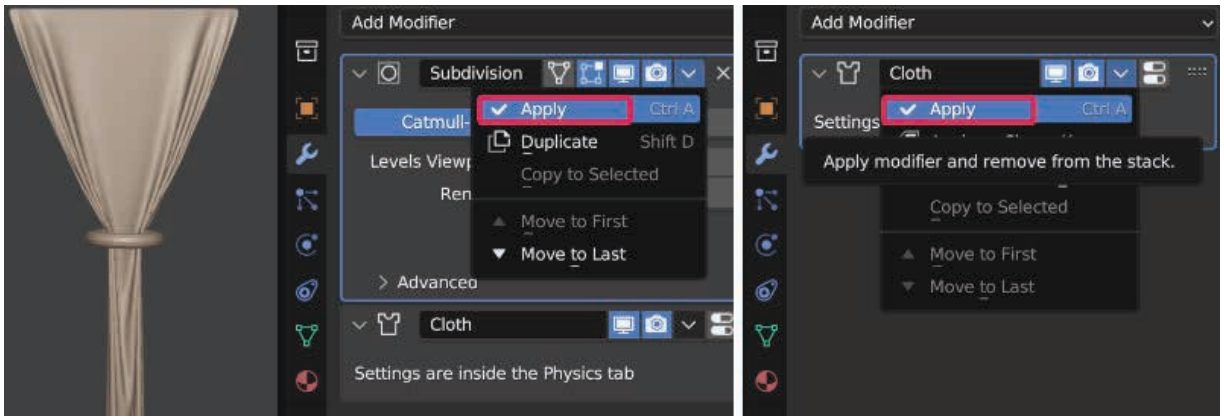
If your computer is having trouble calculating the physics simulation in real-time, try using **Bake All Dynamics** in the cloth settings, that way it is calculated in the background.

It will take some time, but it is less likely to crash.

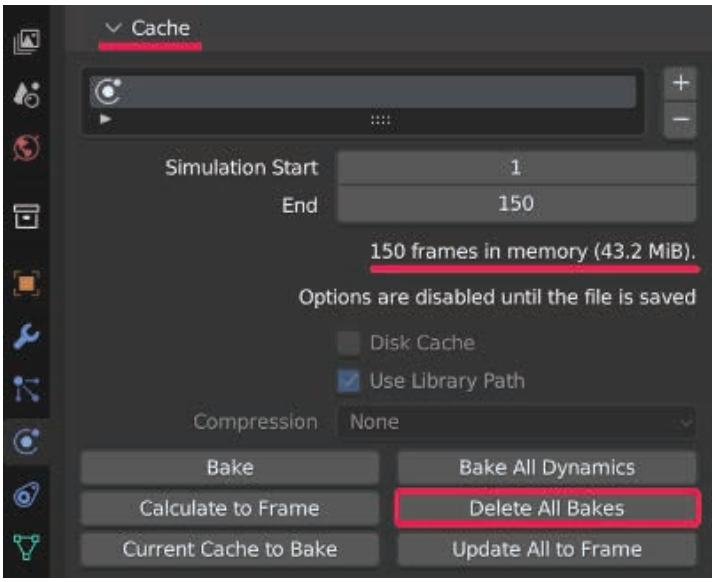


You will see how many frames have been calculated. Once ready, play the simulation by pressing the **Spacebar**.

When you are satisfied with the result of the simulation, click **Apply** in the Cloth modifier. From there on, it's just a regular mesh that you can fine-tune with polygonal modeling and / or sculpting.

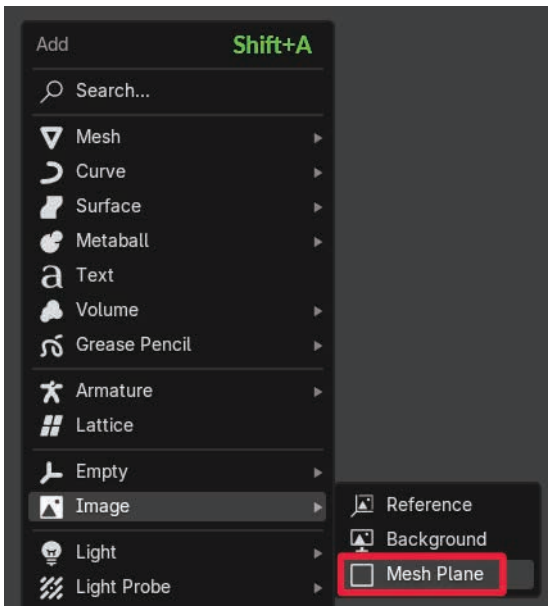


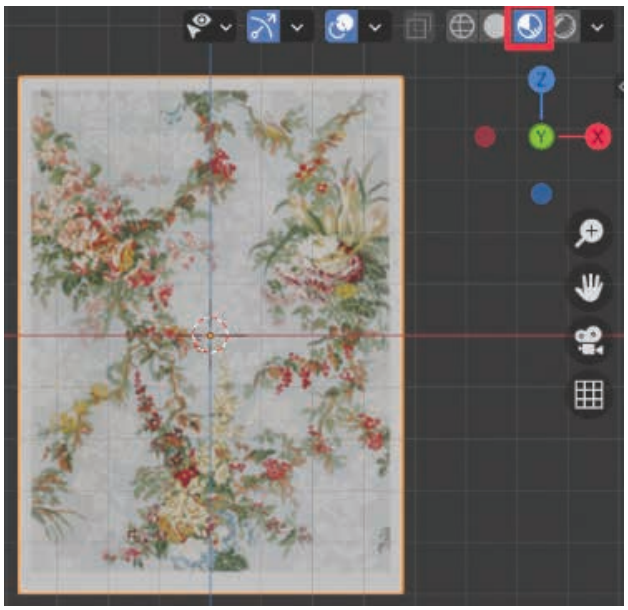
Just remember to click on "**Delete All Bakes**" in case you change something, and want to calculate the physics again.



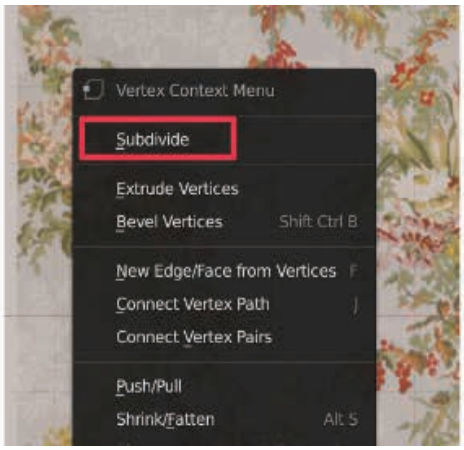
Make a curtain using Shape Keys and a Cloth simulation

Create a **Plane** or import a curtain texture (like this one from Rawpixel) as a Mesh Plane. **Press Shift+A and go to Image > Mesh Plane** to load your image as a mesh. You may need to rotate it 90° on the X-axis by **pressing R+X+90**. Enable **Material view** to see the texture.

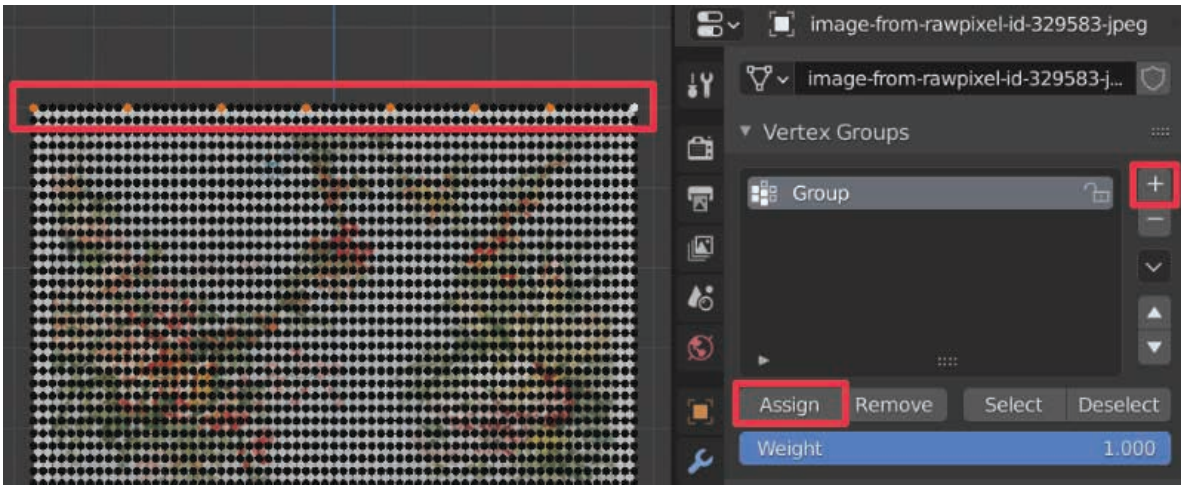




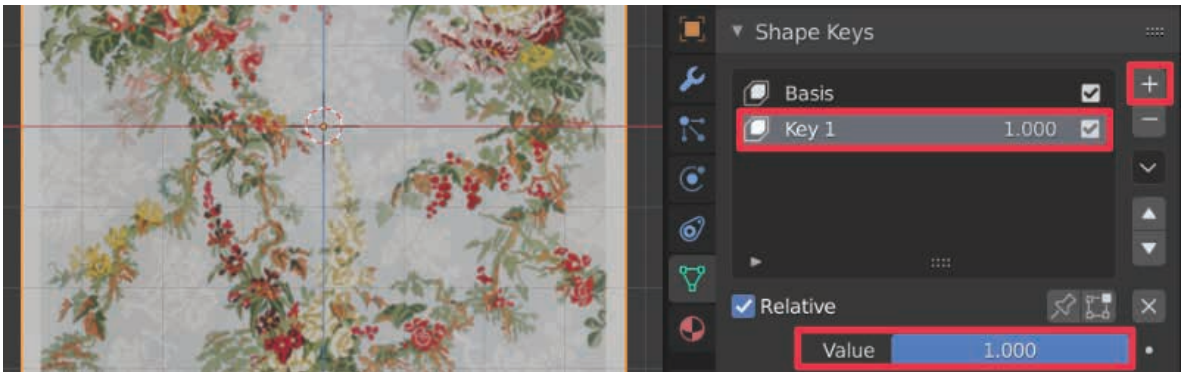
In Object Mode, right-click and choose **Shade Smooth**. In Edit Mode, right-click and choose **Subdivide**. Press **Shift+R** a few times to Subdivide more. You'll need those subdivisions for the cloth simulation.



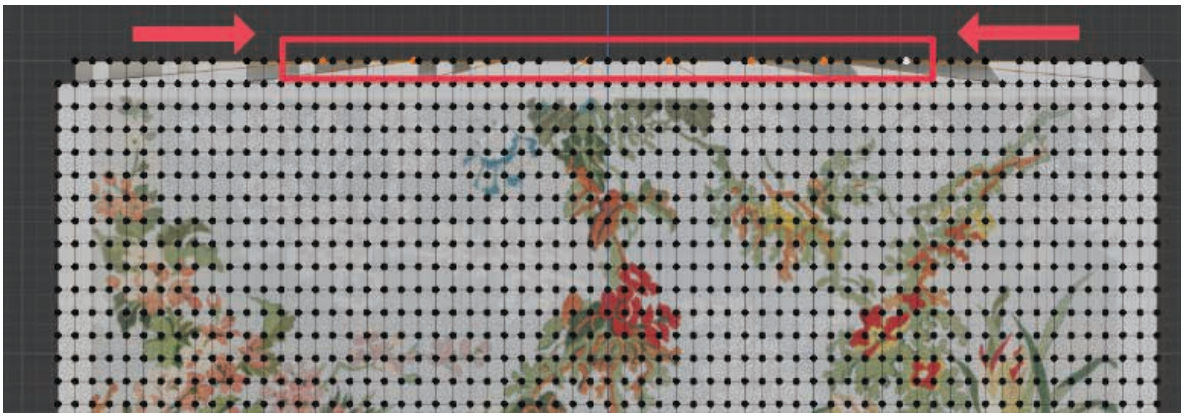
Select the top row of Vertices (or **one Vertex for every 10 or so**) and add them to a Vertex Group by **clicking on the + icon** and clicking on **Assign**.



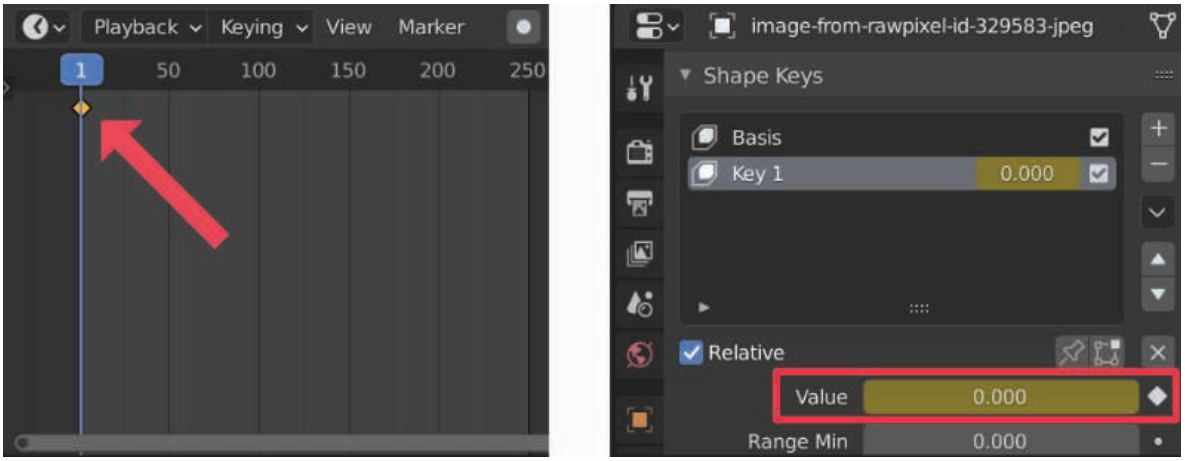
In Object Mode create a **Basis** shape key by **clicking on the + icon** and create another shape key by **clicking on the + icon** once more. Set the second Shape Key Value to **1**.



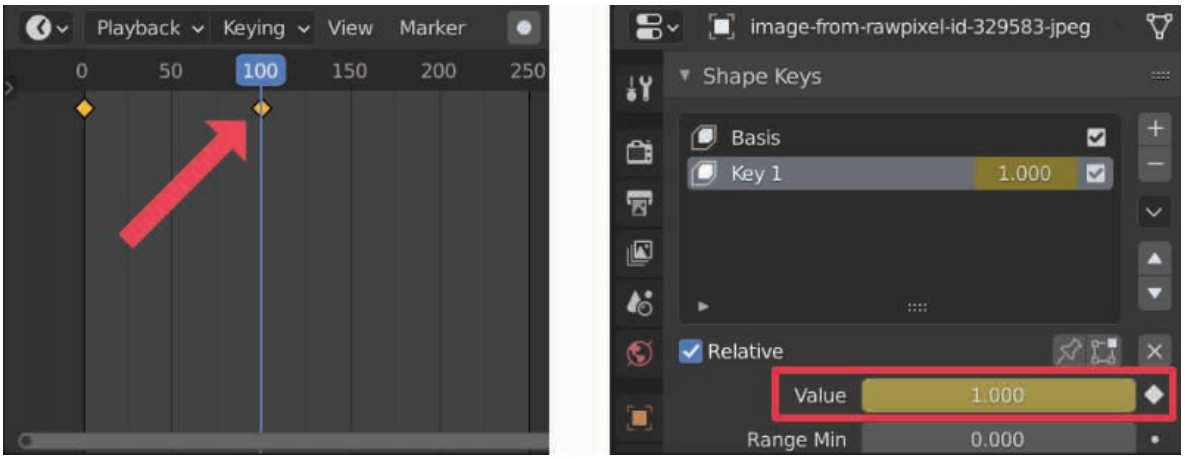
With that Key 1 selected, in Edit Mode scale the top vertices down horizontally by about 50% by **pressing S+X** and moving the mouse. If you only selected one Vertex every other ten or so Vertices, this will look kind of glitchy, but don't worry about it.



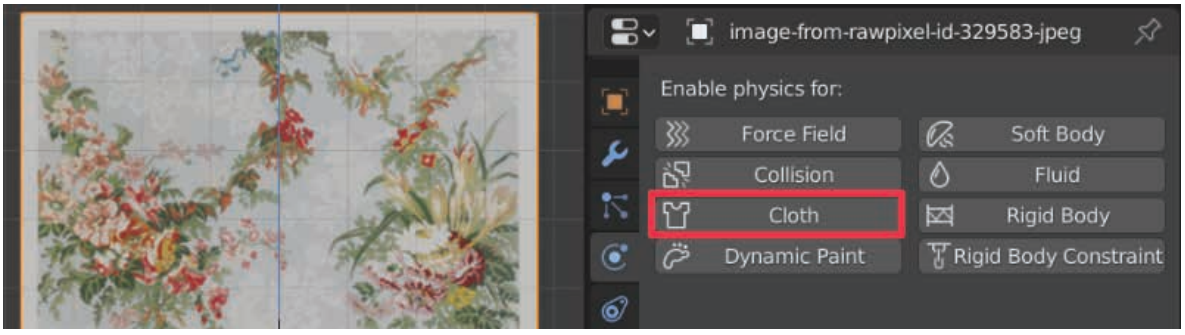
Now set a keyframe on the first frame of the Timeline for Shape Key value 0 by **pressing i** while hovering the mouse cursor over the value field or **clicking on the dot** next to the field.



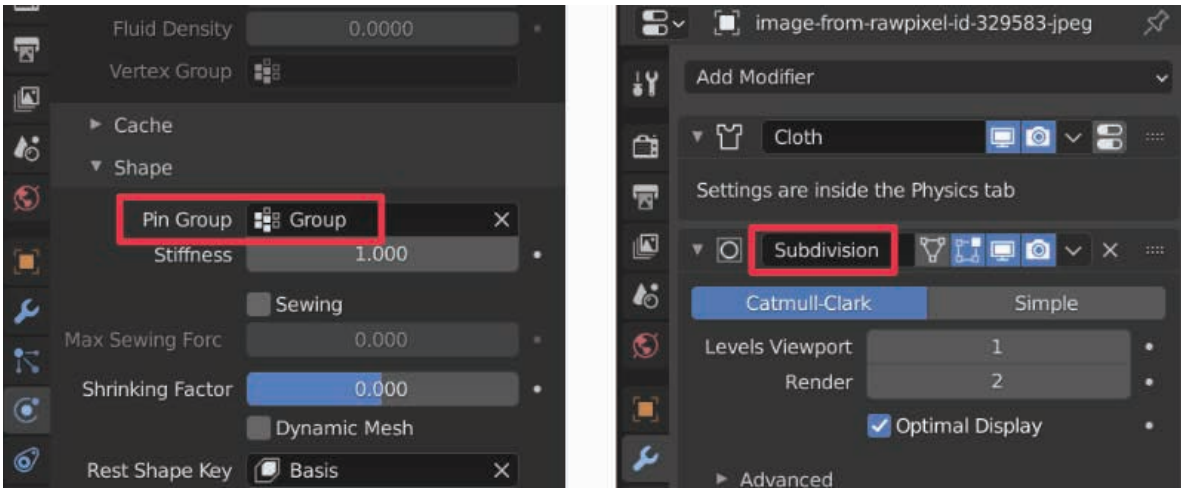
Set a key on a later frame of the Timeline for value **1**.



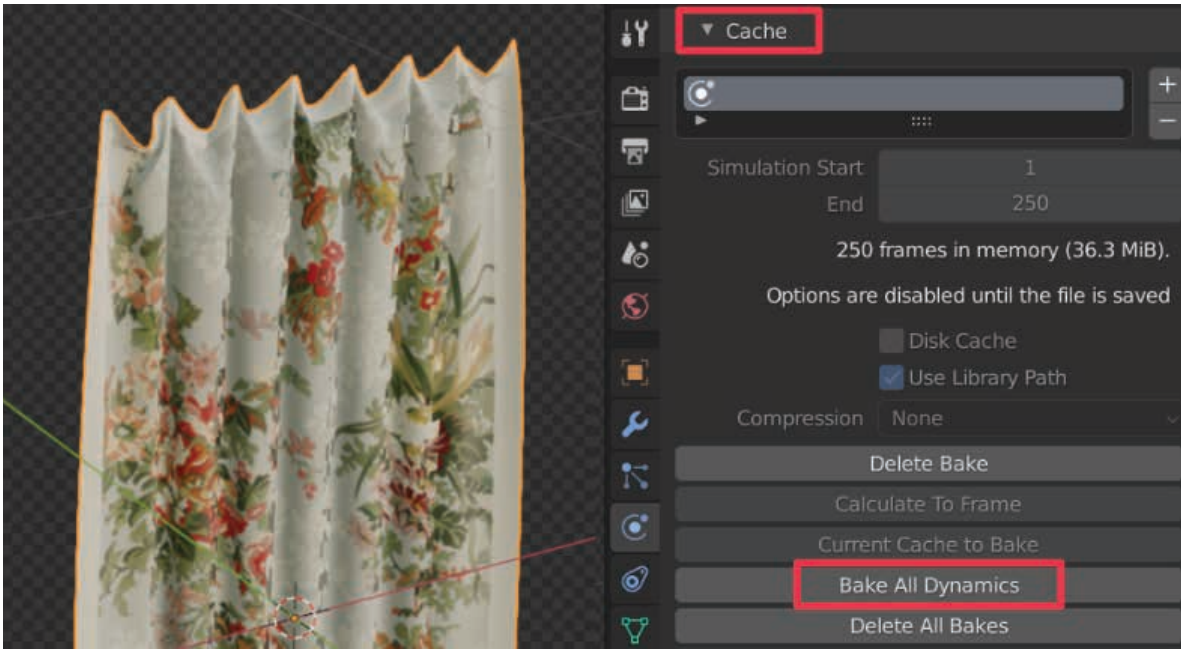
Now add a **Cloth** sim.



Under **Shape**, add the Vertex Group to the **Pin Group**. This will make sure the Cloth doesn't fall down from gravity once the simulation is played back. Add a **Subdivision modifier** under the Cloth modifier to smooth things out.

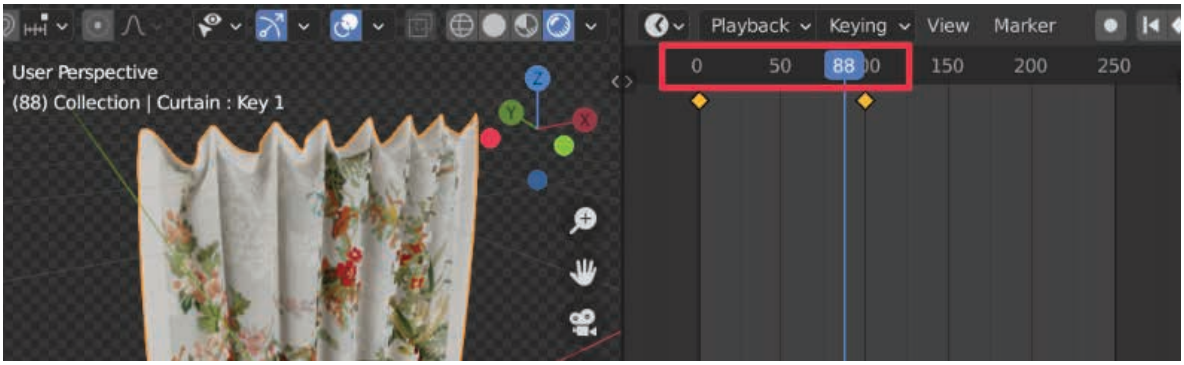


Now you can play the simulation by **pressing the Spacebar** (make sure you start from frame 1), but I recommend you Bake the simulation first by clicking on **Bake All Dynamics**. After the baking is finished, the simulation will play back more smoothly.

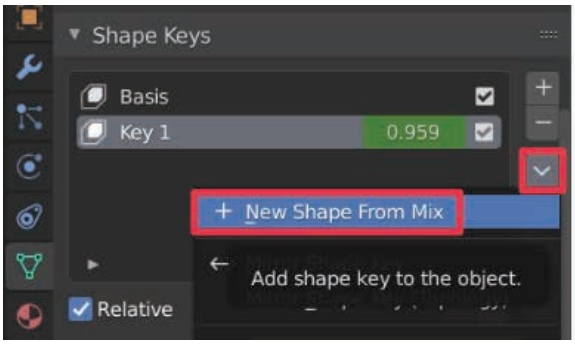


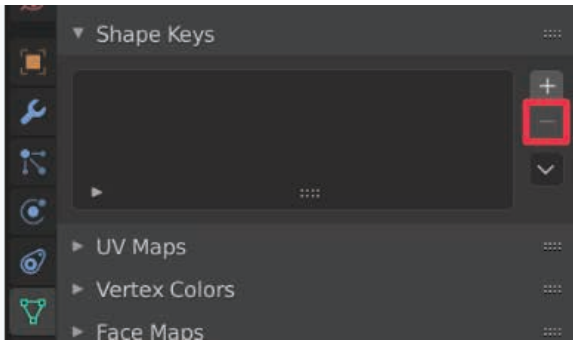
Apply Shape keys and Cloth sim

So, you've created a curtain with the cloth simulation and now you want to turn it into a regular polygonal mesh. First, pick the frame of the simulation you like best.

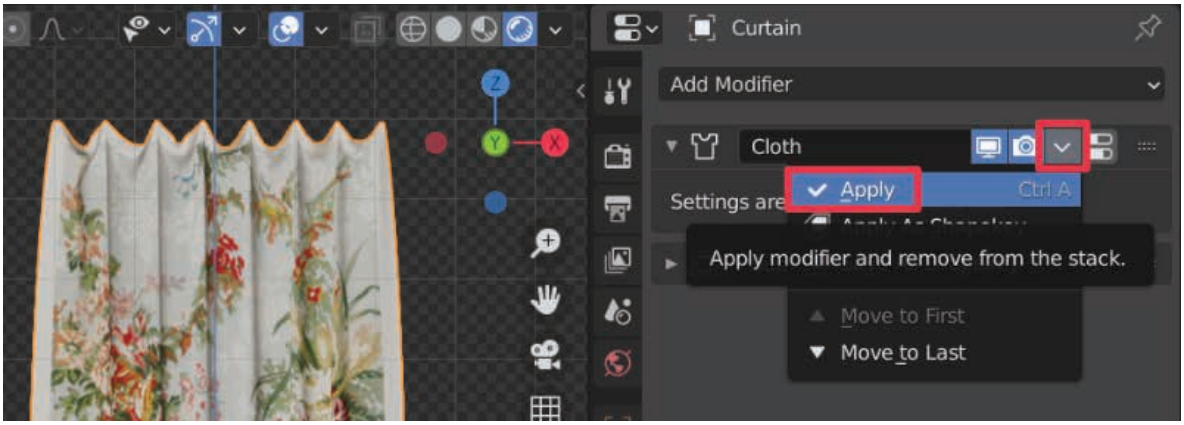


Create a **New Shape from Mix**, then delete the old Shape Keys (Basis and Key 1) and then finally delete the Mix (named Key 2 by default) by **pressing on the - icon**.

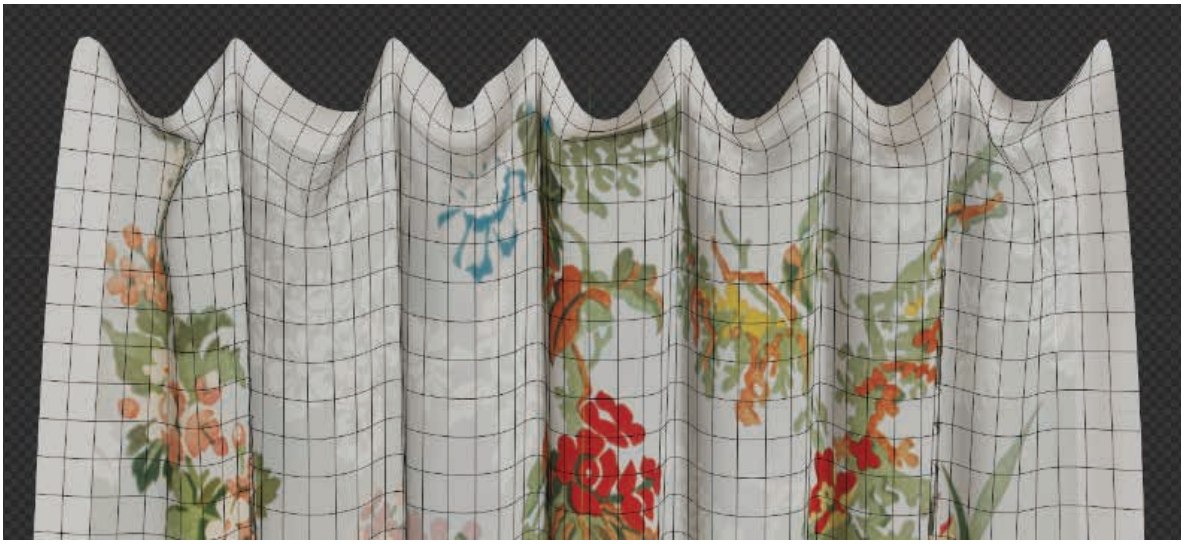




Finally apply the Cloth modifier, either by selecting **Apply** from its drop-down menu or by hovering the mouse cursor over the modifier and **pressing Ctrl+A**.

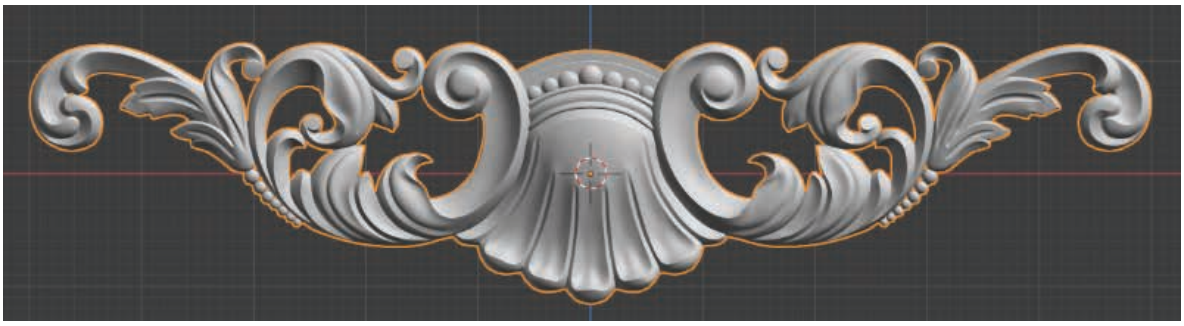


And there you go, a regular old-fashioned polygonal mesh. Just like your grandmother used to make! If you want you can add more detail by using the **Cloth Brush** in Sculpt mode.



8 Tips for modeling Decorative Elements

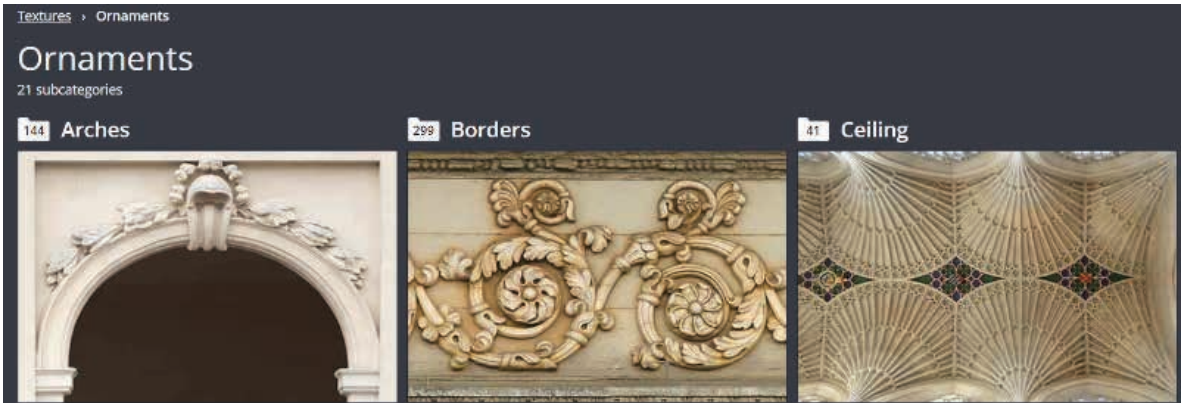
In the next pages we'll look at 8 tips that will help you with the modeling of Baroque (or other styles) of ornamental decorative elements such as this one.



1. Good Visual Reference

First, you'll need decent visual reference to base your model on. A great resource is the website Photobash.co, where you can find all kinds of photographic reference. If you type in "ornaments" you'll find a bunch of photo packs. They are not free, but worth every eurocent in my opinion.

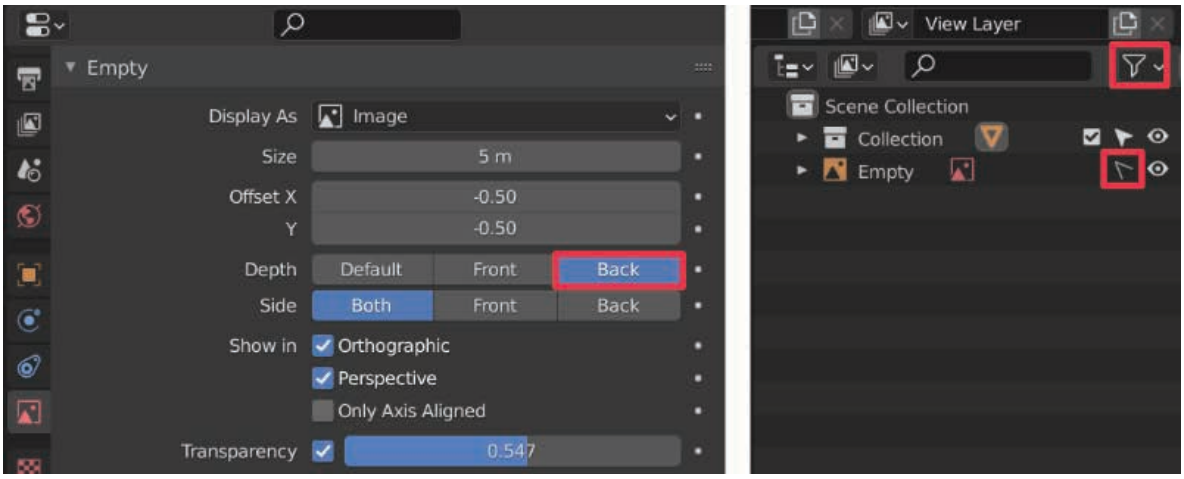
For free alternatives check Textures.com which has a dedicated Ornaments section and Rawpixel, another website which features many decorative visuals, with the added benefit that a lot of them are **CC0** (a type of public domain license).



2. Importing reference images

Importing reference images is easy. First go to Front orthographic view by **pressing 1** on the numpad. Now drag and drop your reference image to the viewport. **Press Alt+G** to make sure its origin is at the world origin. If necessary you can still adjust the position slightly by **pressing G** and moving it.

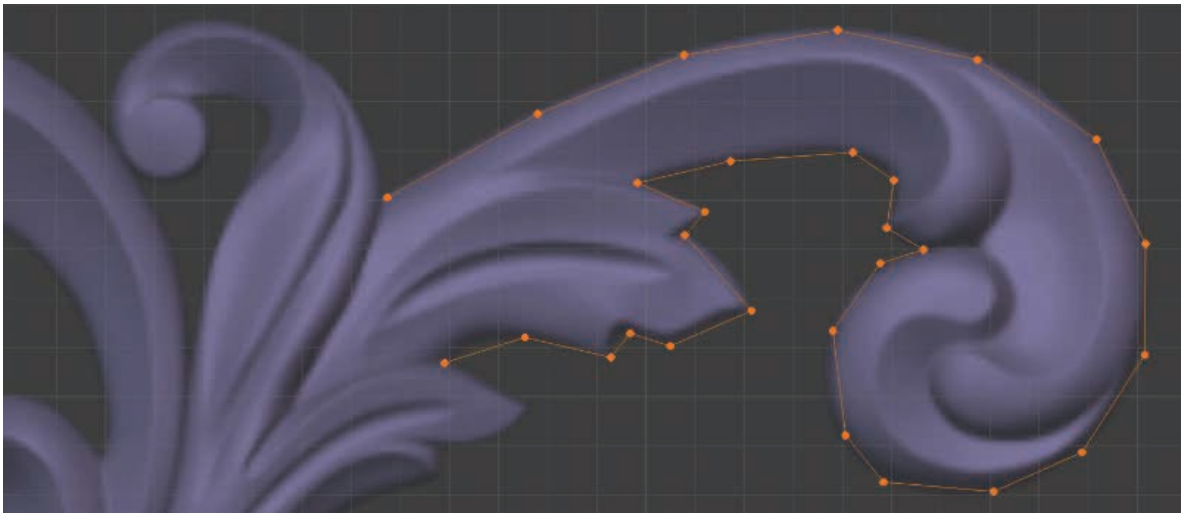
Set Depth to "Back" so it doesn't obscure the 3D model, then turn off selectability for the reference image in the Outliner so you don't accidentally select it while modeling.



3. Start with the boundary

When you start modeling, don't attempt to do the entire object all at once. Instead, model it in small parts. Create a single vertex by merging the default cube to center in Edit Mode by **pressing M** and choosing **At Center**, then extrude it to trace the boundary of the object by **holding Ctrl and right-clicking** to extrude vertices.

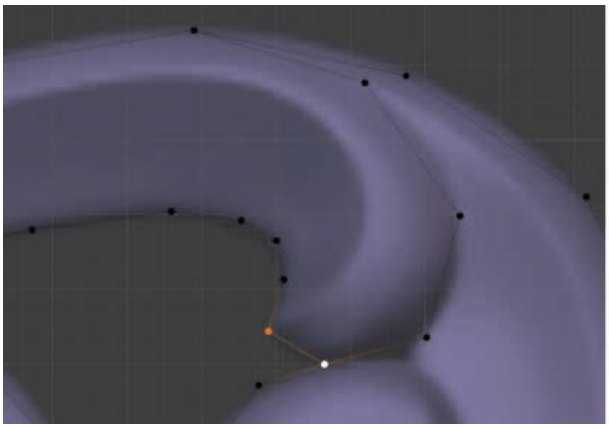
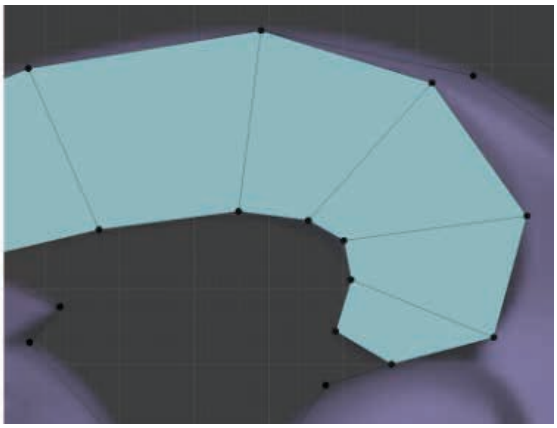
Don't worry about getting the countour to match the reference perfectly while you're extruding, you can fine-tune the position of the vertices later.



4. Work from big to small, from boundary to inside.

After tracing the boundary, start tracing the next step of detail from big to small and from outside to inside.

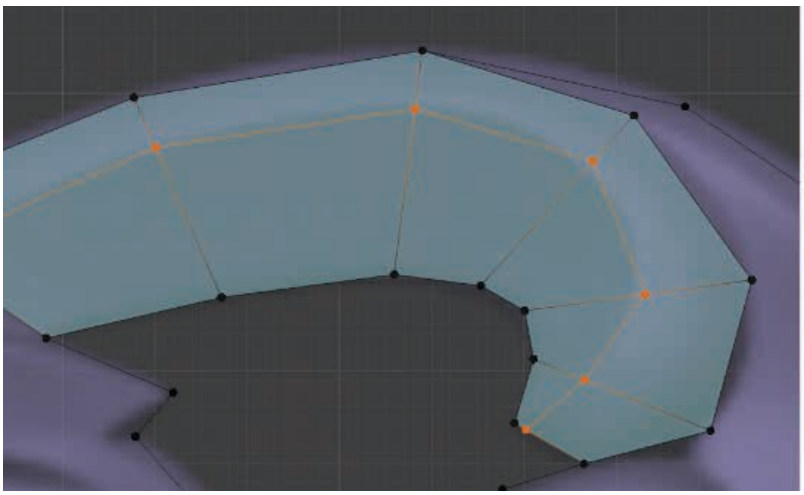
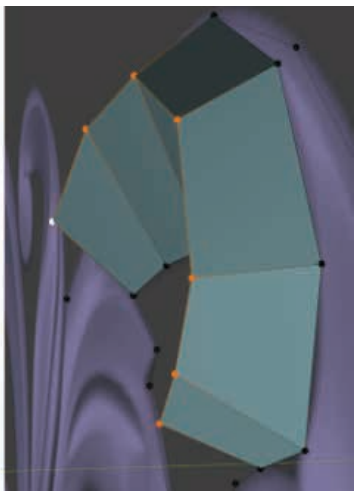
If you enable to **F2 add-on** in Preferences, you can quickly fill the faces by **pressing F** repeatedly after selecting just the first two vertices. You may find that you need to add a vertex here and there, by selecting two vertices and **pressing Ctrl+R** to add one in between.

5. Adding ridges and valleys

To add ridges and valleys to make the model less flat, **press Ctrl+R** to add an edge loop to the faces, then scale it in 3D by **pressing Alt+S**. You can rotate the view in the viewport to check the model from the side.

Slide individual vertices by **pressing G twice**, so they match better the position of the reference image. This is the very basics of modeling ornaments in a nutshell, for the rest it is mostly repeating these same steps.

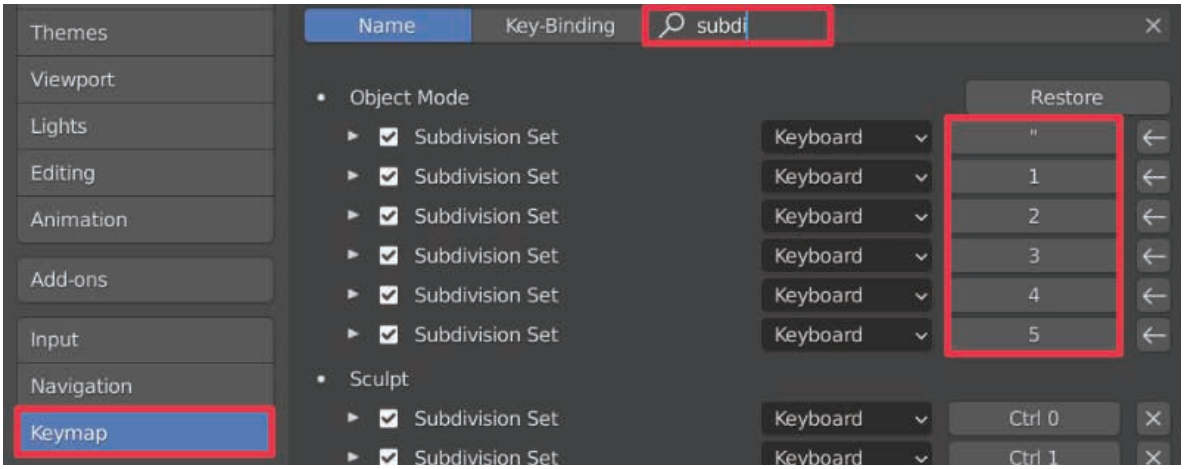



6. Customize your subdivision keymaps

Check your subdivided model regularly in Object Mode, then turn it off again before continuing modeling in Edit Mode. You can do this by p**ressing Ctrl+[a number from 1-5]** to see the model with subdivisions, and **Ctrl+0** to turn off subdivisions. In practice you need very long fingers to perform this last keyboard shortcut.

I prefer to remap these shortcuts in Preferences - under **Keymap** search for "subdivision". Then **click on the fields with the keyboard shortcuts** to enter your own new keyboard shortcut. I've remapped all of these keys so I don't need to even hold Ctrl.

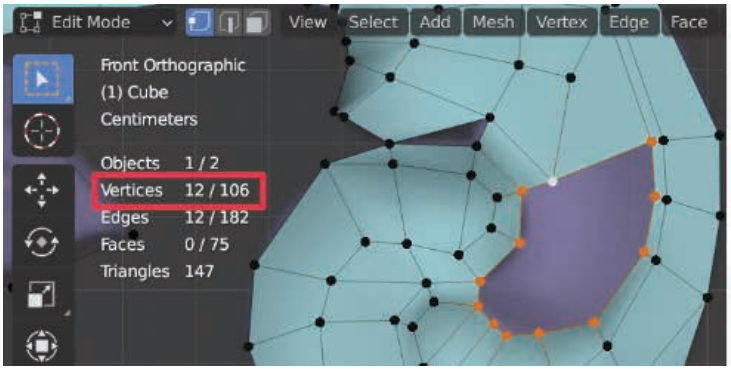
For level 0 I use the key to the left of 1 *(probably ~ tilde/accent grave in your case)*. It's a very quick way to toggle subdivisions on and off and check how your model looks subdivided.

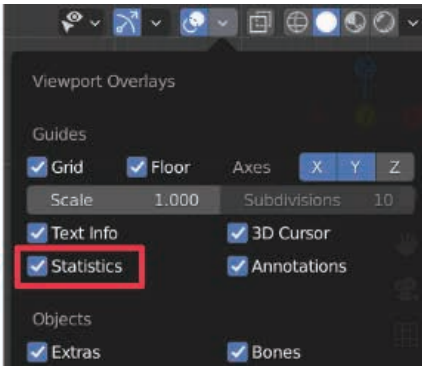


7. Quads and even numbers of vertices

A useful trick to keep in mind when doing quad modeling, is that before you fill an area with several faces, you count the amount of vertices by selecting the inner edge loop of that area. Alt+left-click on a vertex to select the entire inner loop.

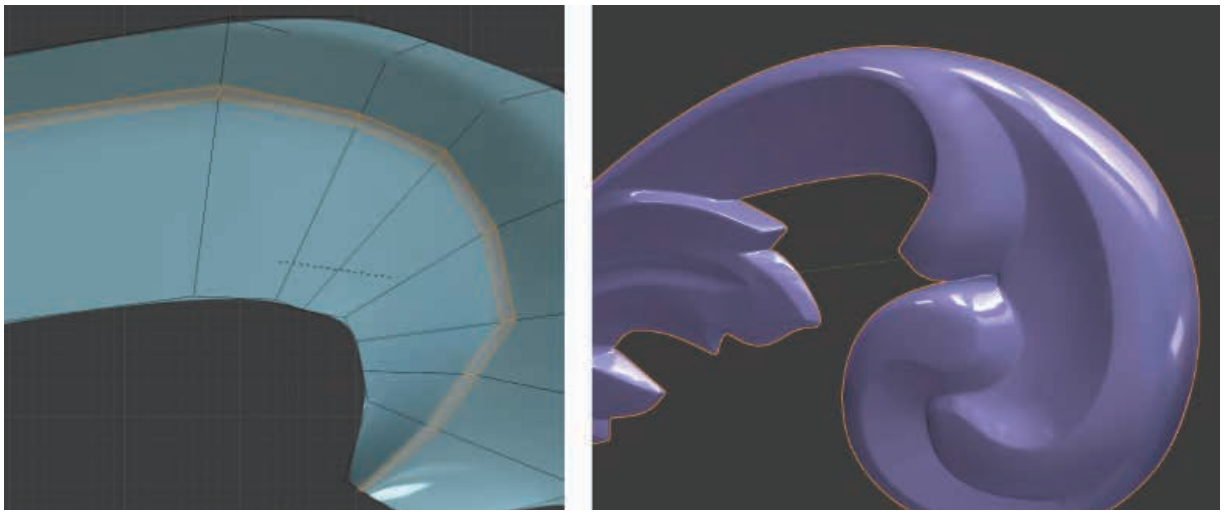
Now in the upper left corner of the viewport you can see the amount of vertices. These should be **an even number**to make your life easier in regards to quad modeling. Since Blender 2.9 you need to enable **Statistics** to see this information. If it's an uneven number, add an edge loop to a face by **pressing Ctrl+R**.





8. Making ridges sharper

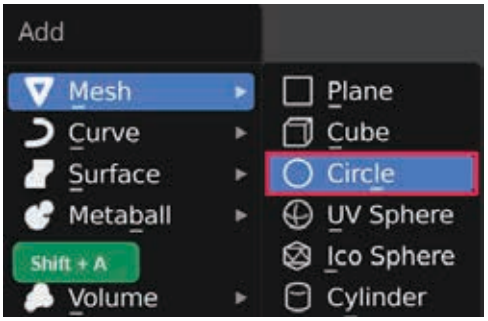
Finally to make the ridges sharper, select those edge loops by **Alt+left+clicking** on them and **pressing Ctrl+B** to bevel. The closer edge loops are together, the sharper that part of the model will be once subdivided.



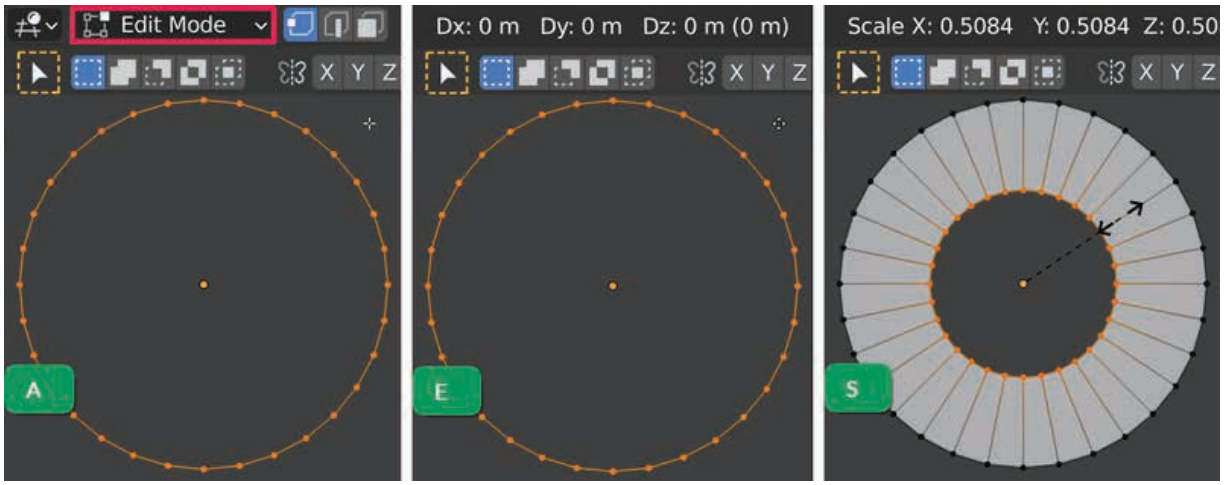
Model a Spiral Staircase

Let's create a spiral staircase.

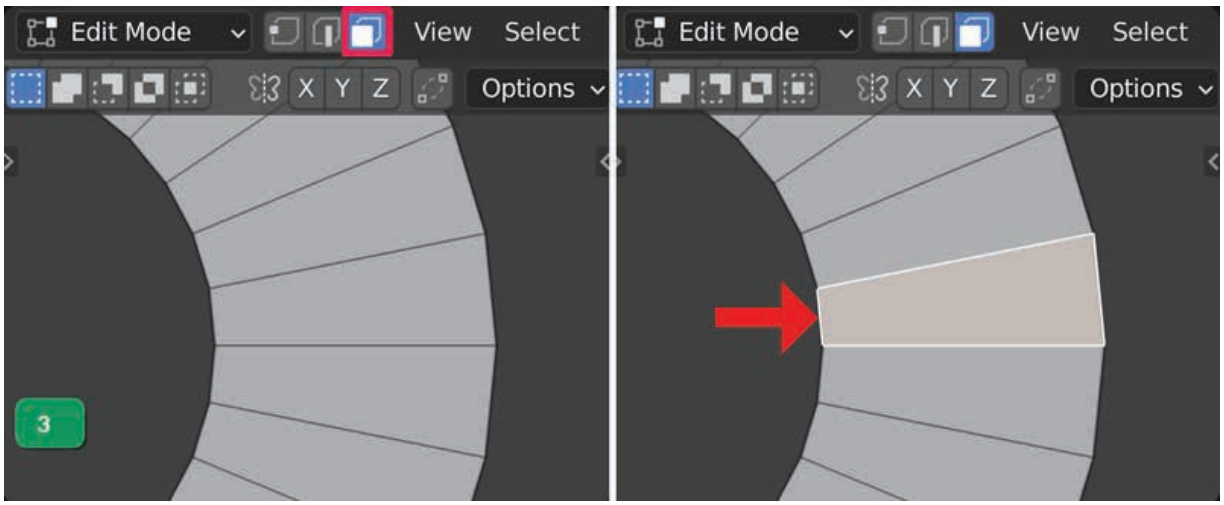
Press Shift A and choose Circle.



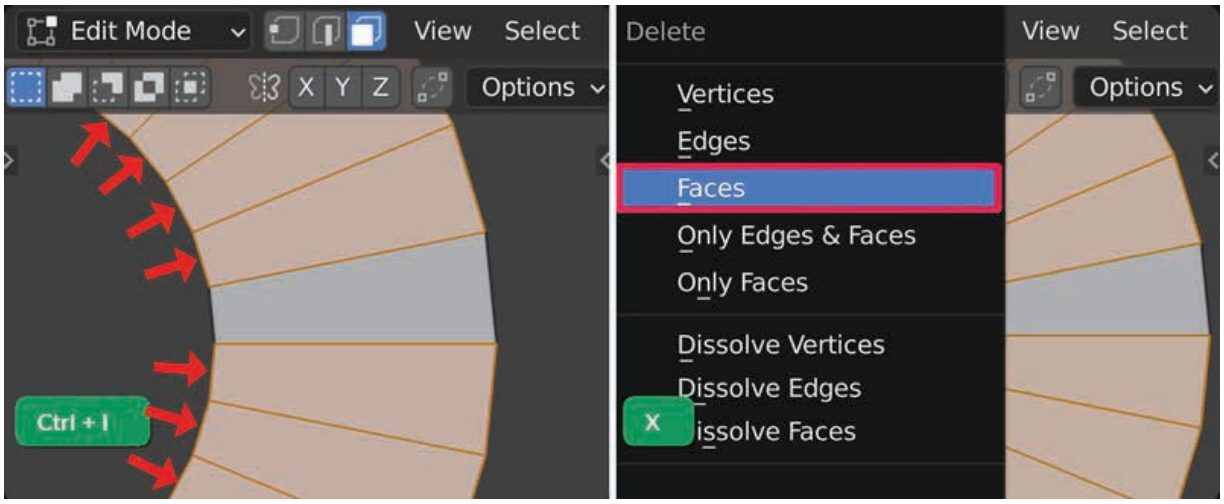
In Edit Mode Select all by **pressing A**. **Press E** to extrude and then **press S** so you can scale down the extruded edges.



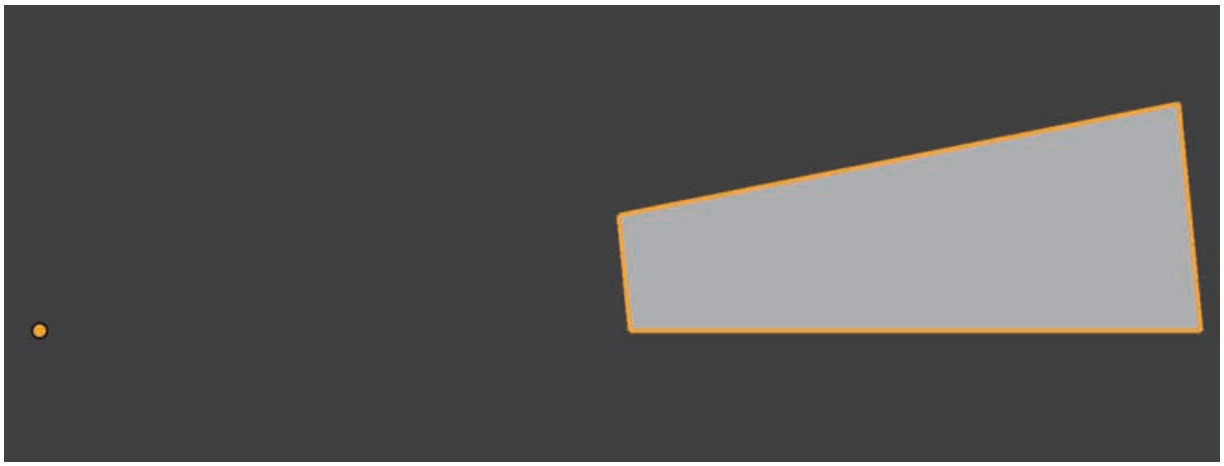
Press 3 to go to **Face** selection mode and select one face.



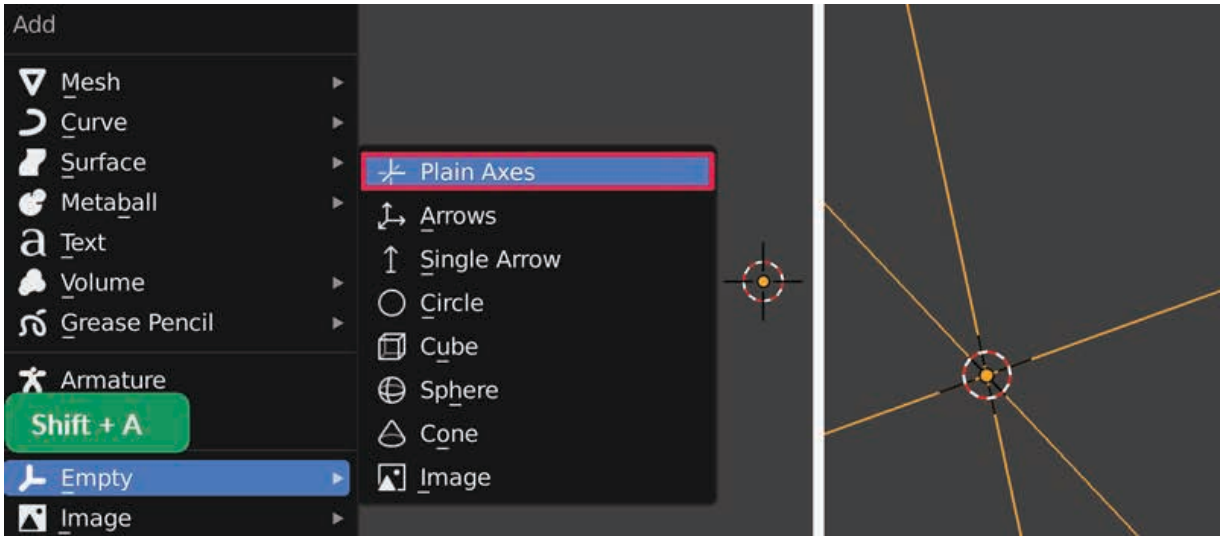
Press Ctrl+i and X then choose **Faces** to delete the other faces.



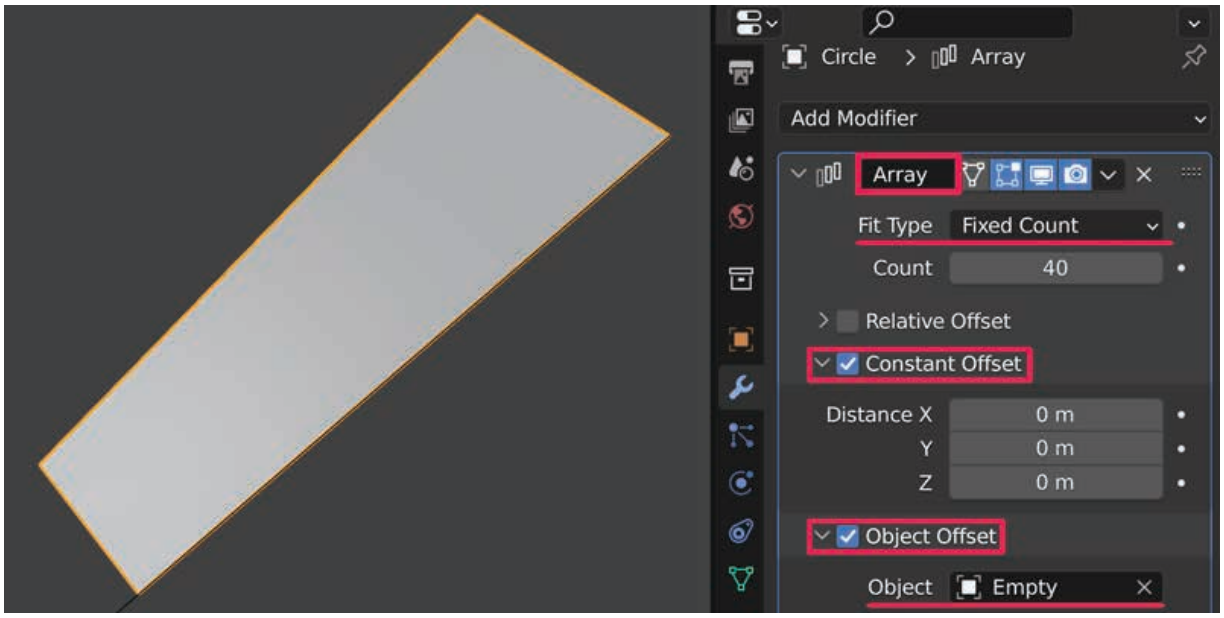
You end up with just the one Face.



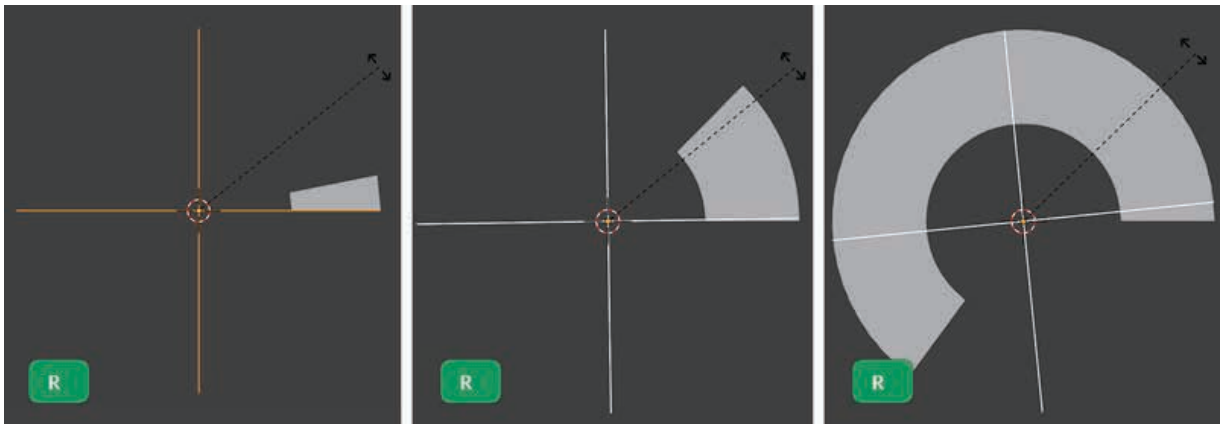
Press Shift+A and add an **Empty.**



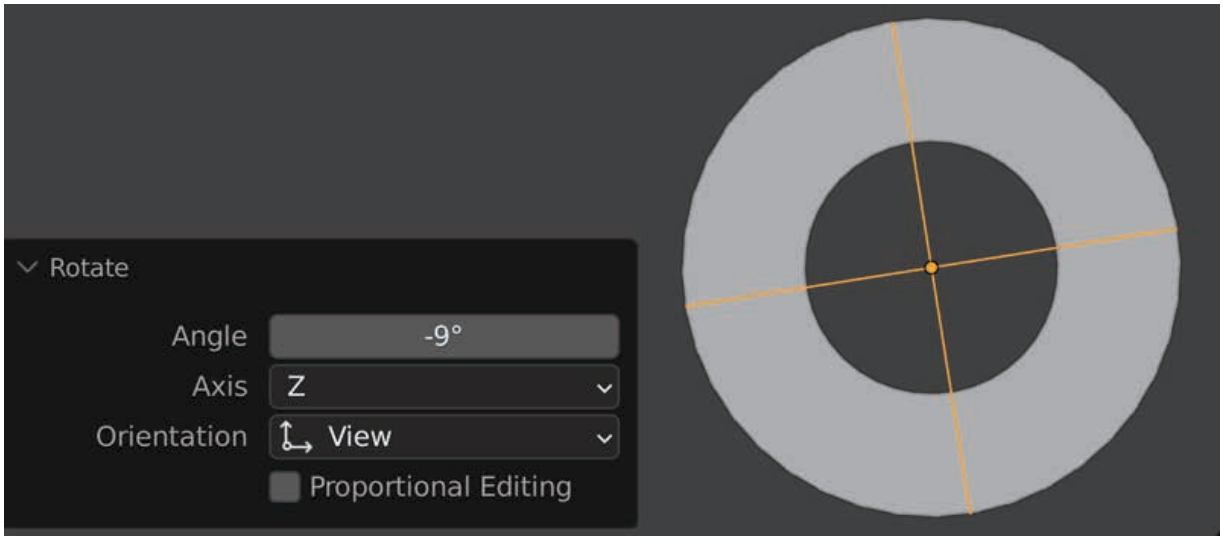
Select the Face again in Object Mode and add an Array modifier. Use these settings for the modifier.



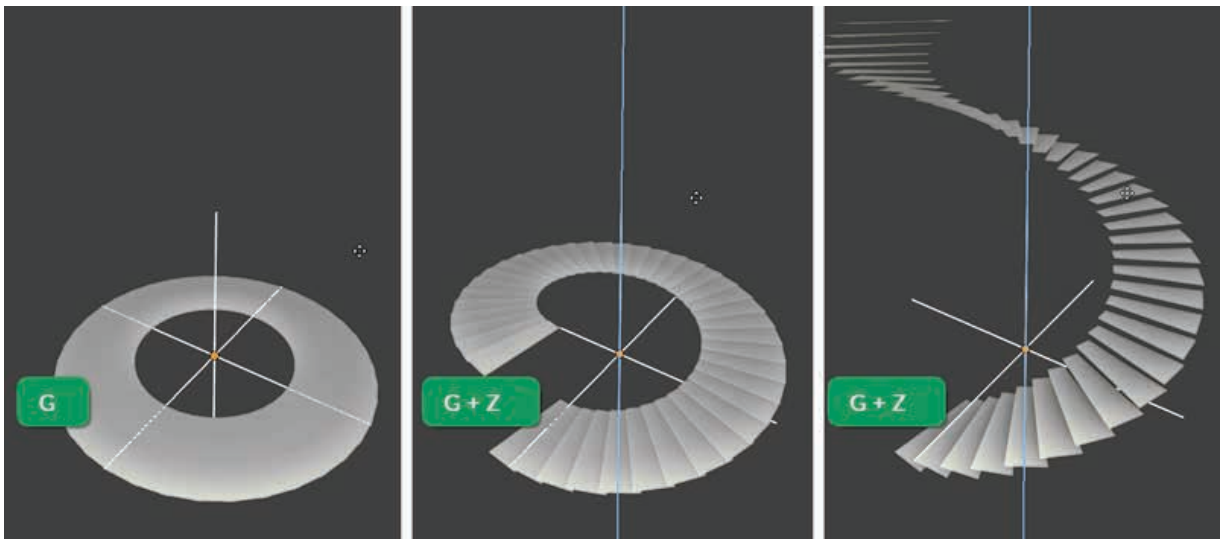
Now when you **rotate the Empty with R...**



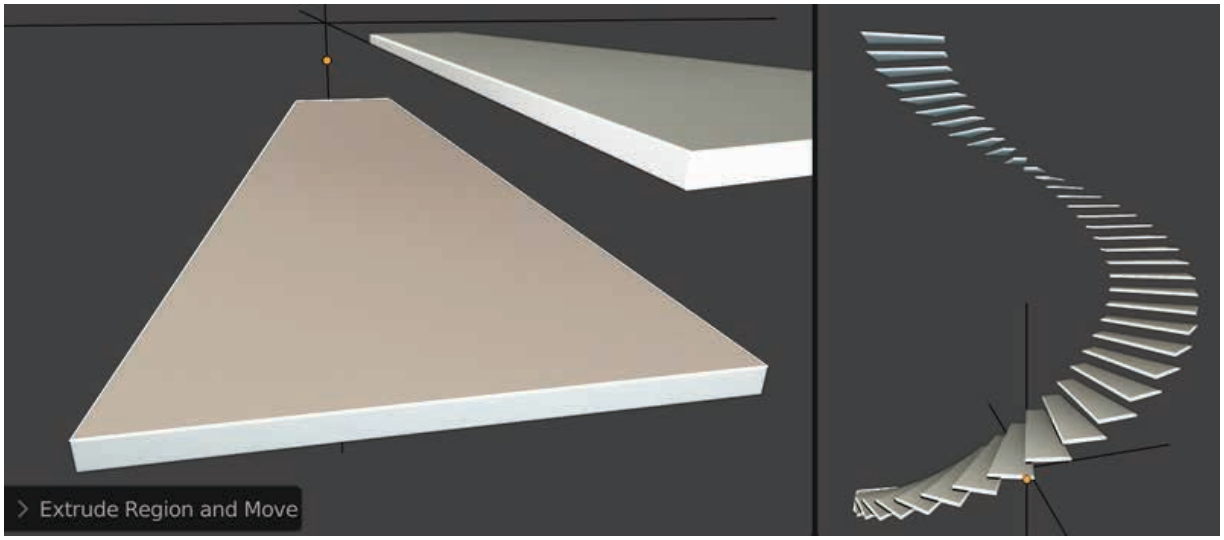
...you create a circular array of the stairs.



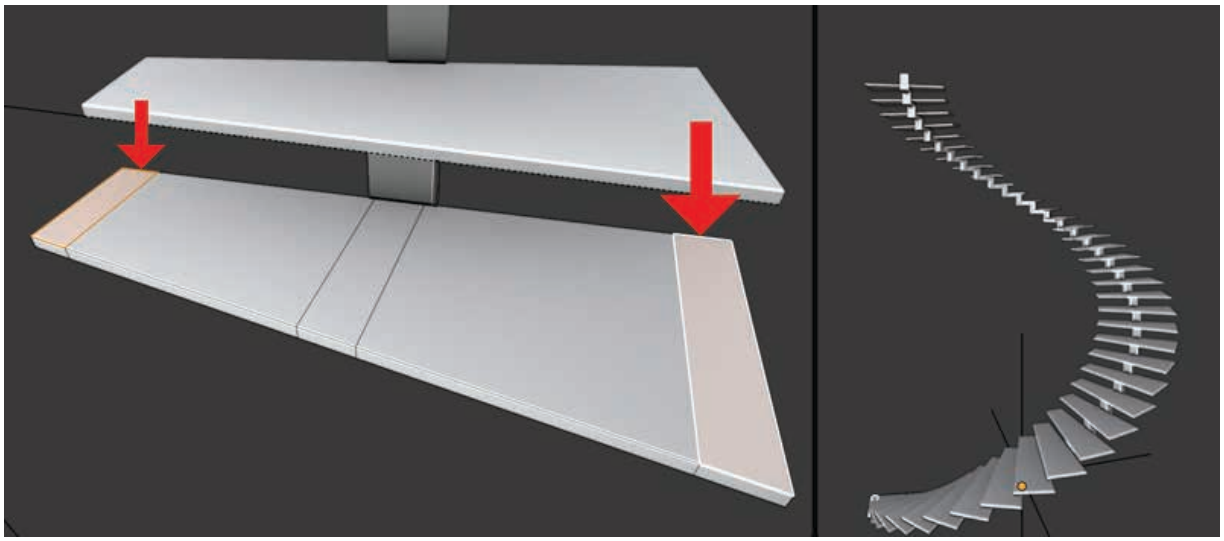
Move it up along the Z-axis by **pressing G and Z** and you'll notice the other steps of the stairs are there.



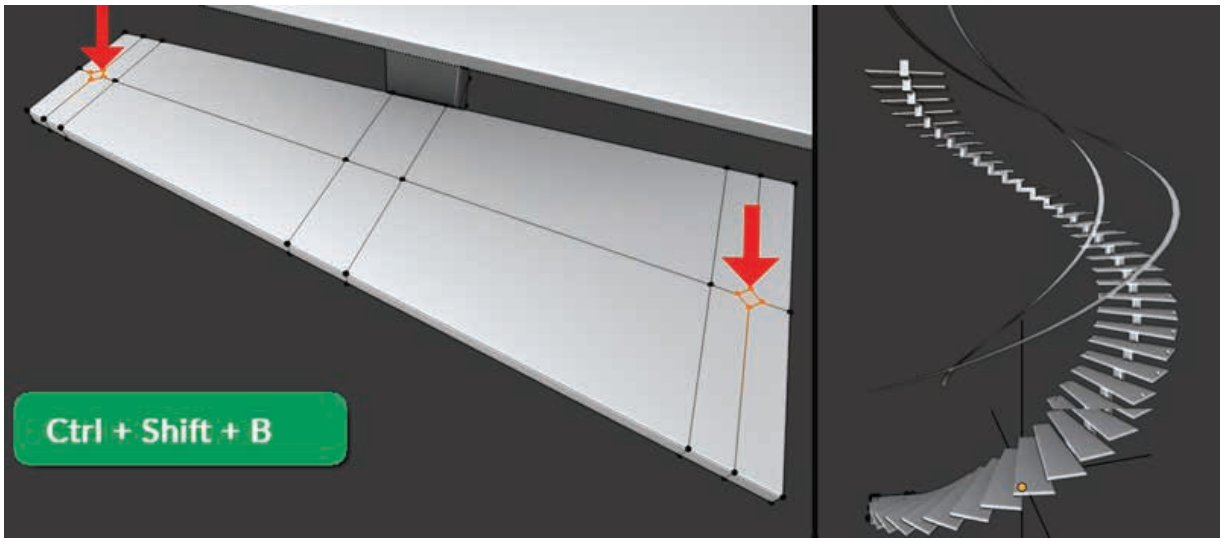
You can still go and edit the original step, all the other steps will be changed as well.



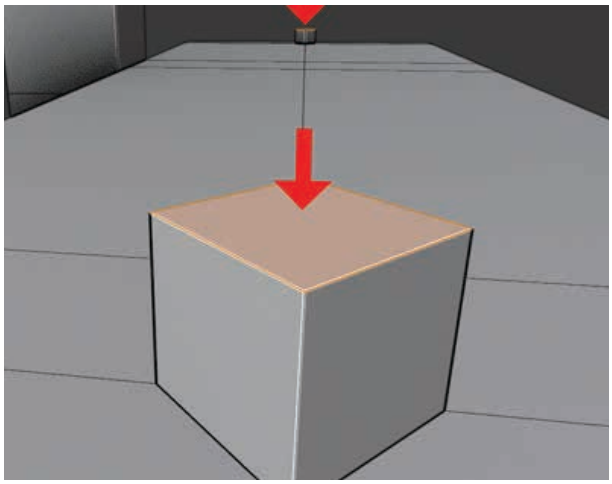
For the arm handle, add an edge loop and then duplicate this face then move it up and move the edges to line up a bit more.

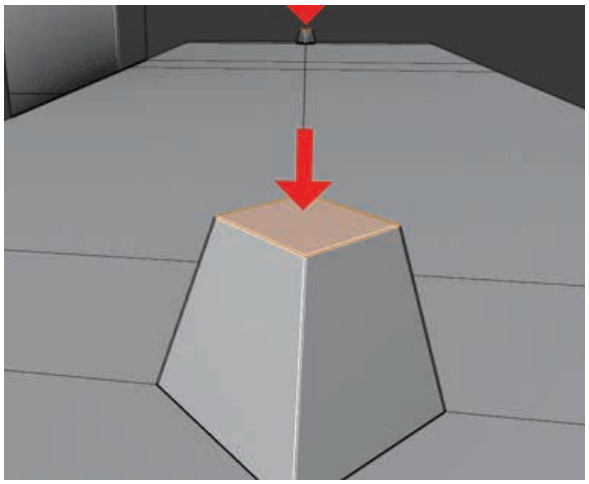


If you add some more Edge Loops with **Ctrl+R**, you can select the vertex where they cross and **press Ctrl+Shift+B** to bevel it.

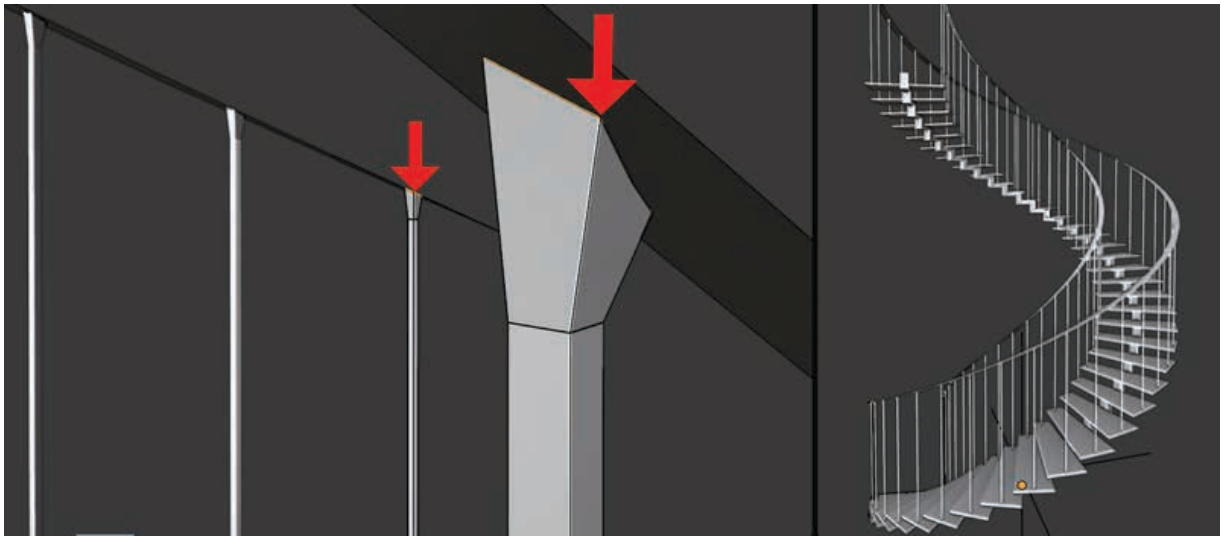


After extruding the beveled vertex up, scale it down, then extrude it and scale it up some more to create vertical beams to support the railing.

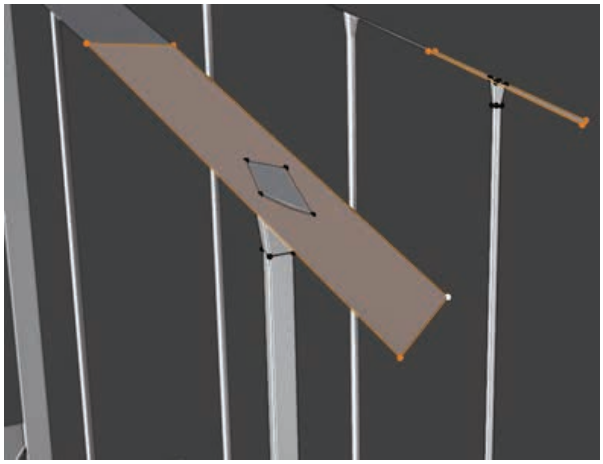


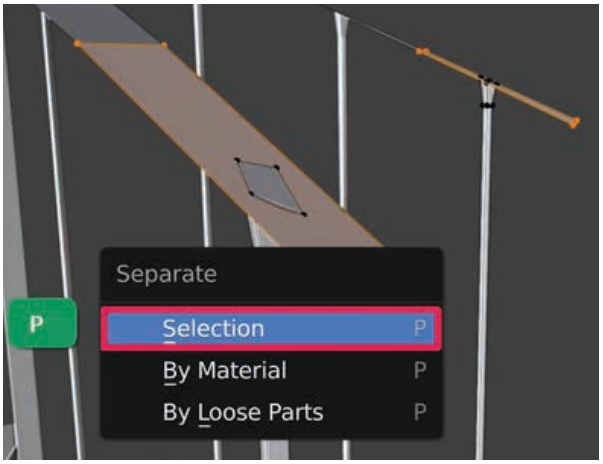


Extrude the beams all the way up to the handrailing.

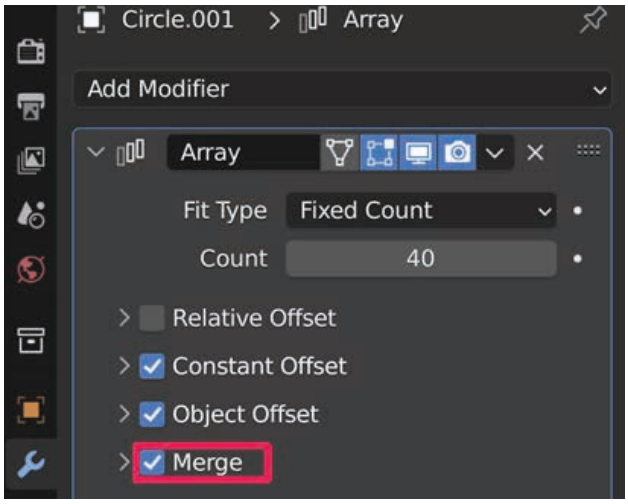


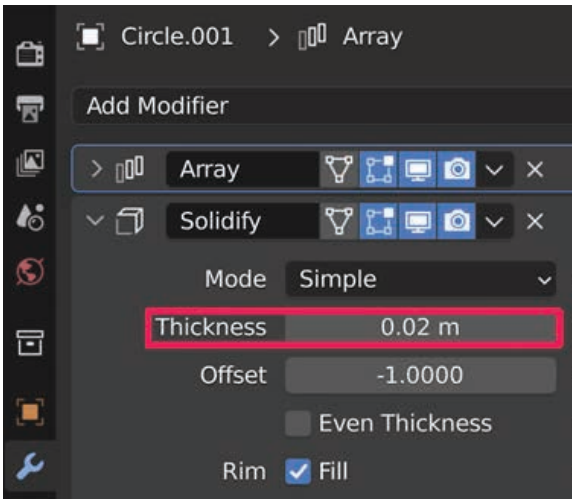
Press P to separate the arm handle as a separate object.



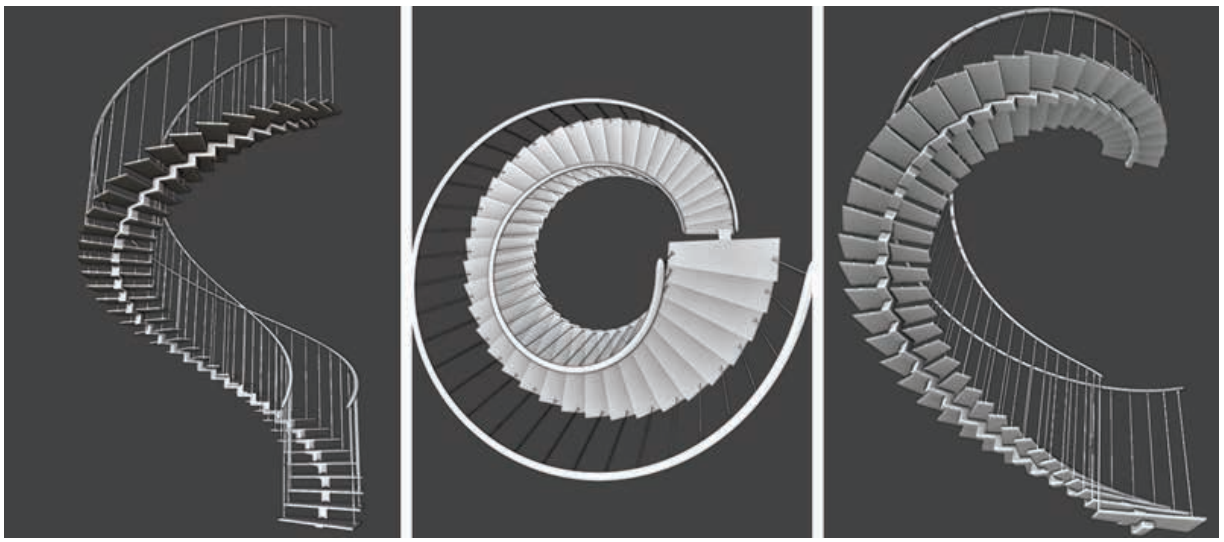


Then turn on **Merge** in its Array modifier and give it as **Subdivision and Solidify modifiers**. Increase the **Thickness** on the Solidify modifier.



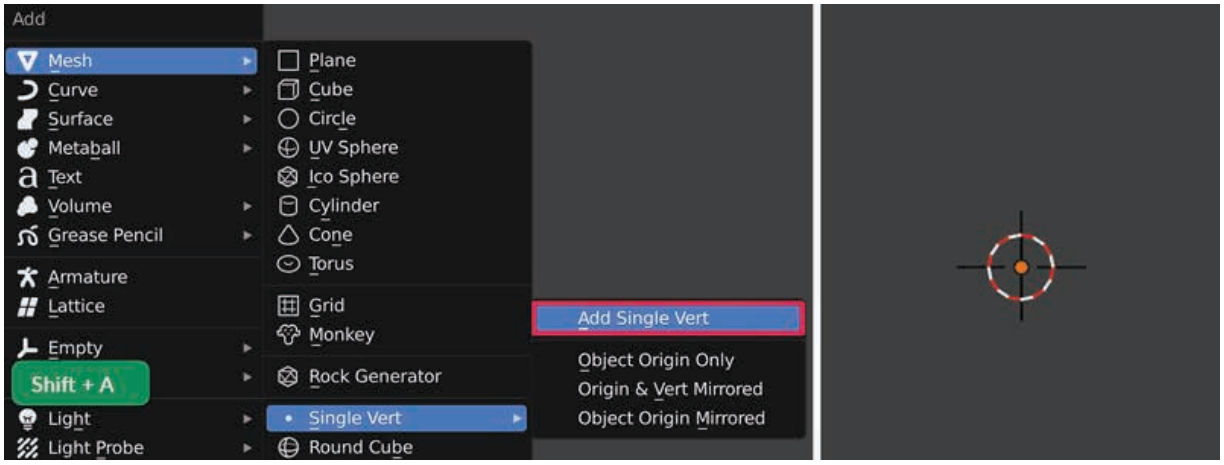


Now the staircase is completed!

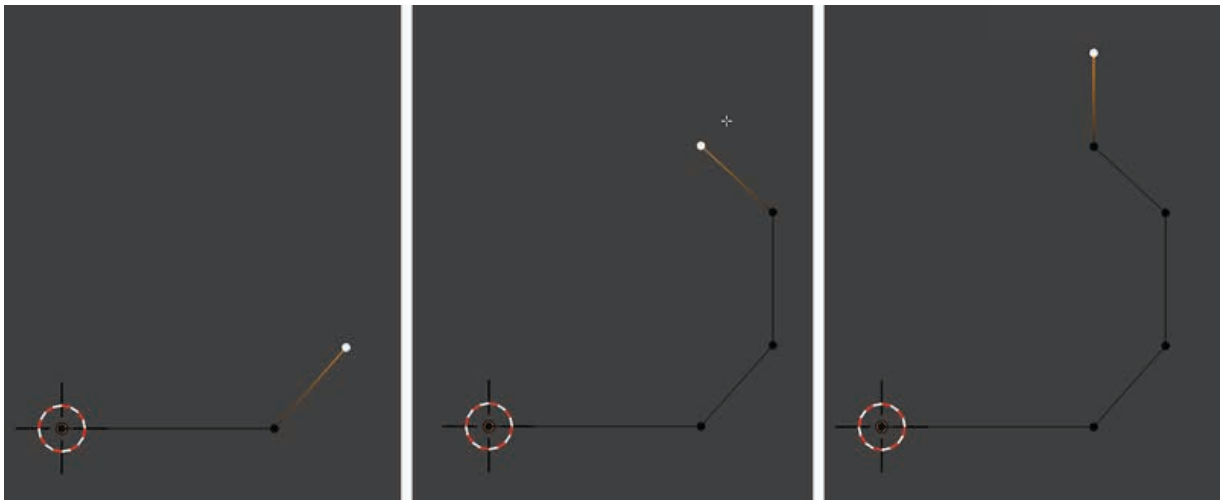


Model a basket using modifiers

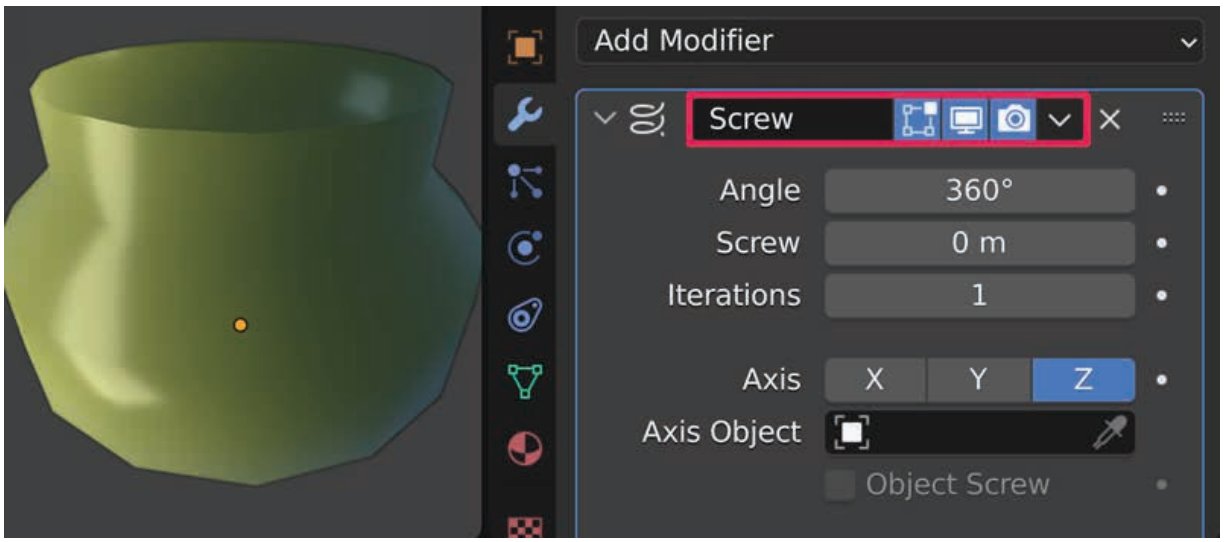
Create a single vertex either by merging vertices of any primitive together or with the Extra Objects add-on. (Enable **Extra Mesh Objects** in the **Get Extensions** tab in Preferences)



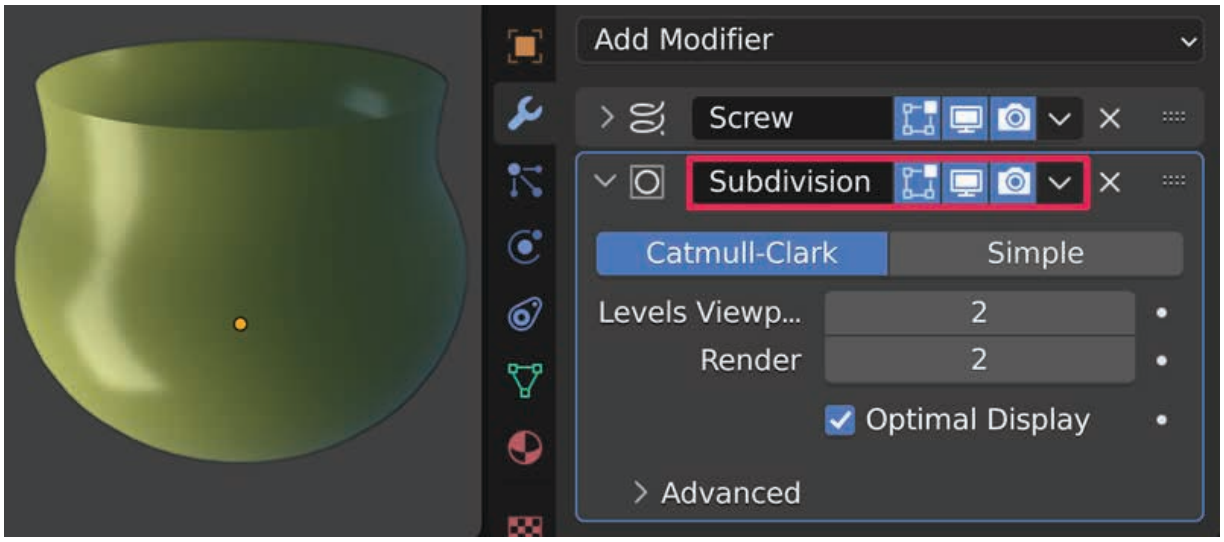
In **Edit Mode** with **Vertex Select** active, extrude the vertex in Side view by **pressing E**, to create the profile of the basket. You may have to **press A** to make sure it's selected.



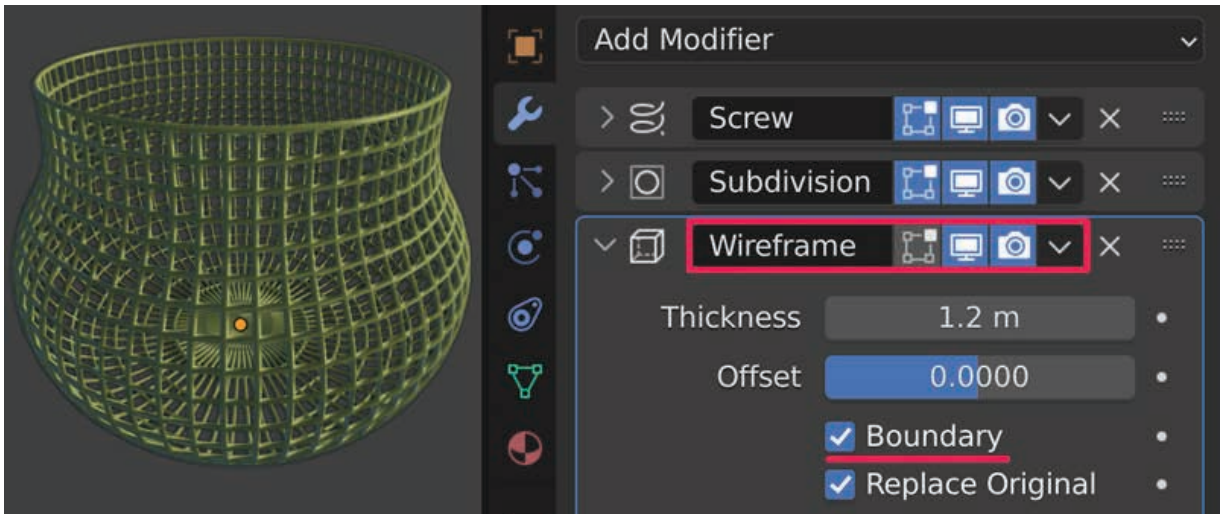
Now add a **Screw modifier** so that it is turned into a round object.



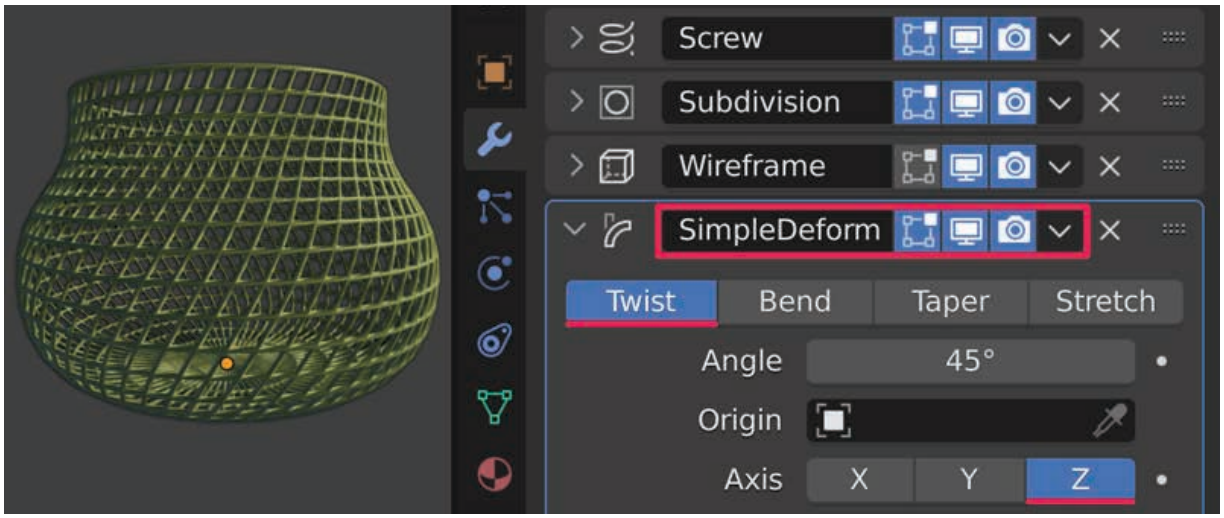
Next, add a **Subdivision Modifier** to create additional geometry.



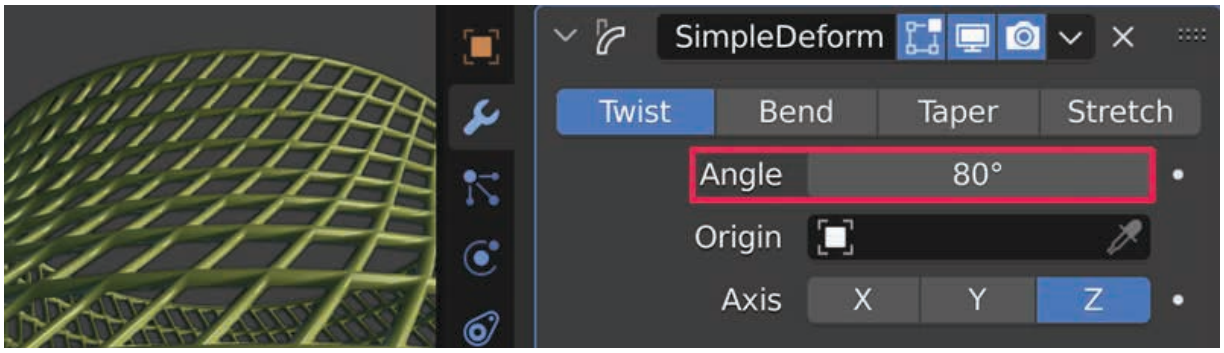
Then add a **Wireframe modifier** to create the Wire mesh effect.



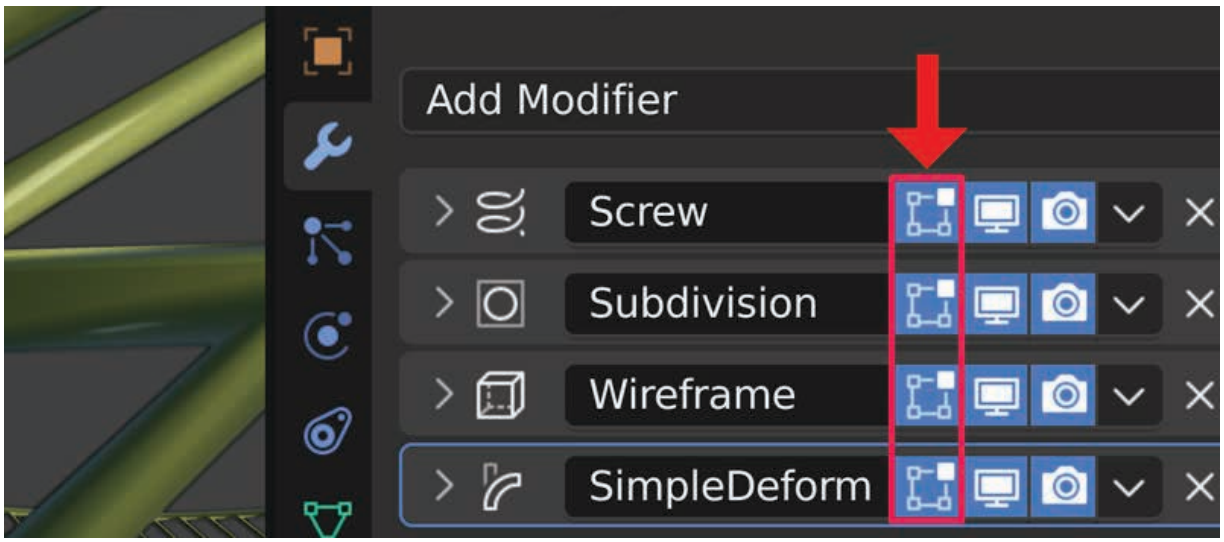
Finally, to create the twisting effect add a **SimpleDeform modifier**, set it to **Twist** and use the **Z-axis**.



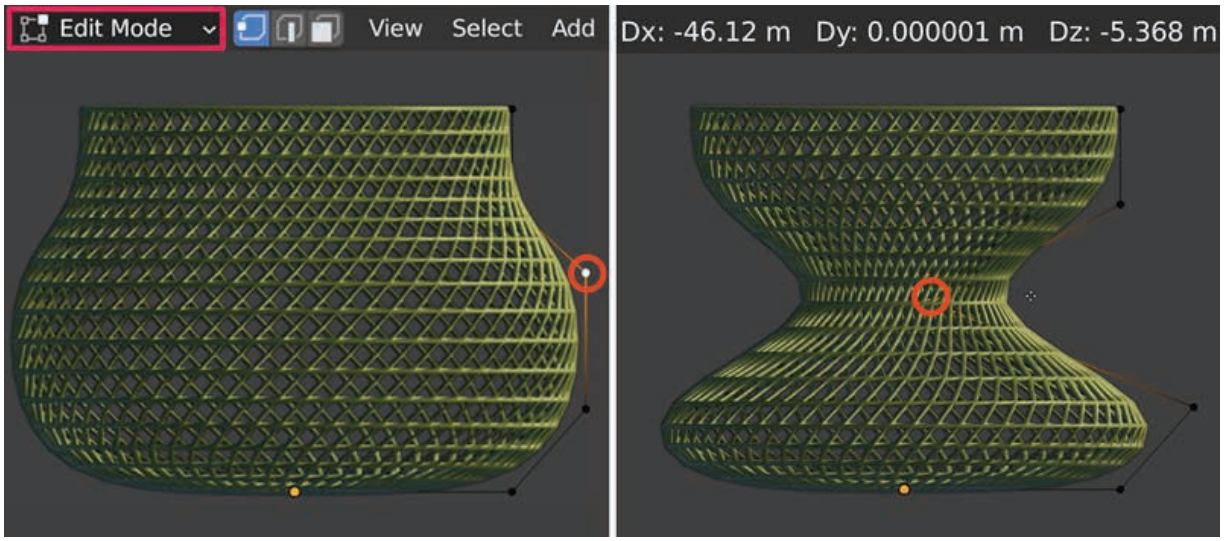
Then increase the **Angle** if you want more of a twist.



If you turn on “Display Modifier in Edit Mode”, you can go into Edit Mode and still move the vertices while seeing the end result.

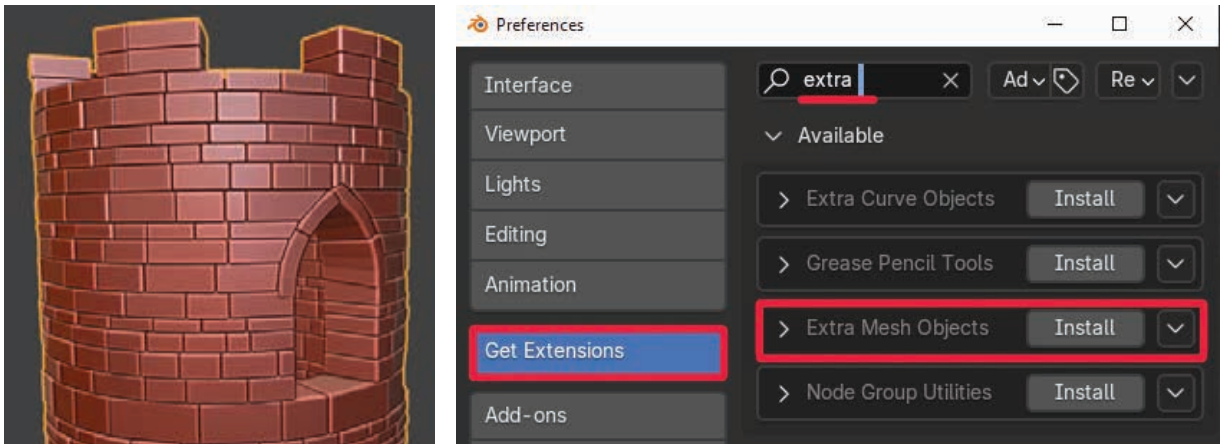


If you turn on “Display Modifier in Edit Mode”, you can go into Edit Mode and still move the vertices while seeing the end result.

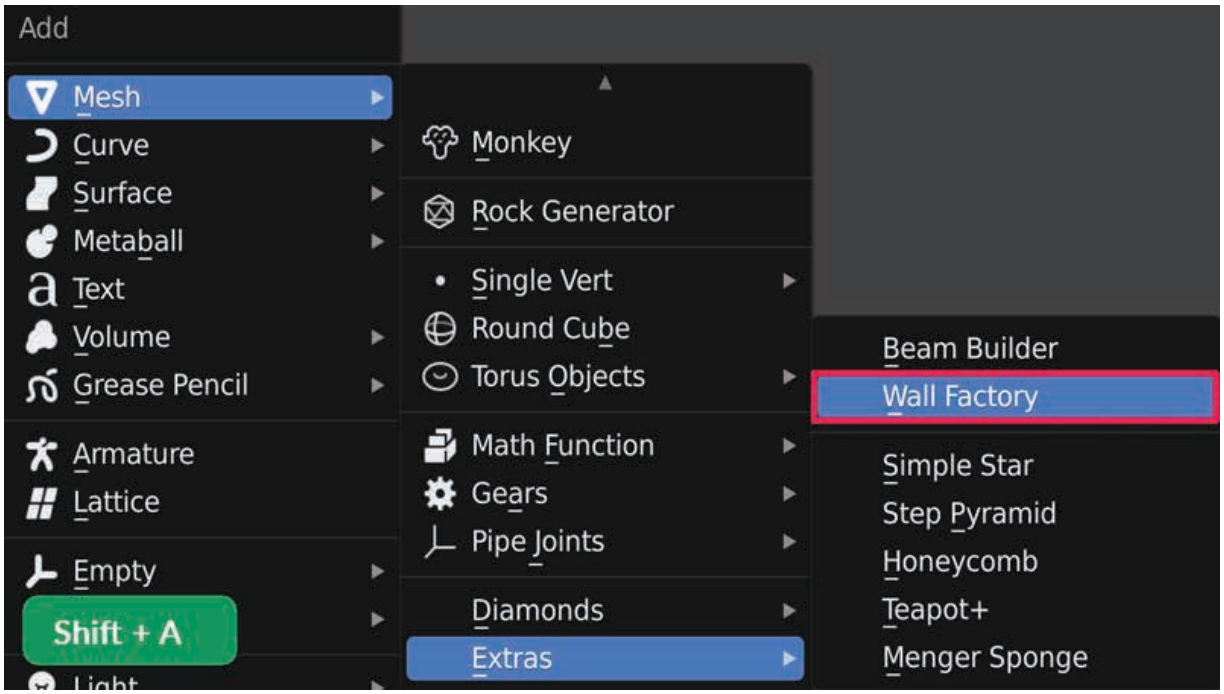


Medieval Tower with Wall Factory

Install the **Extra Mesh Objects add-on t**hat comes with blender. We are going to use it, to create a circular stone tower.

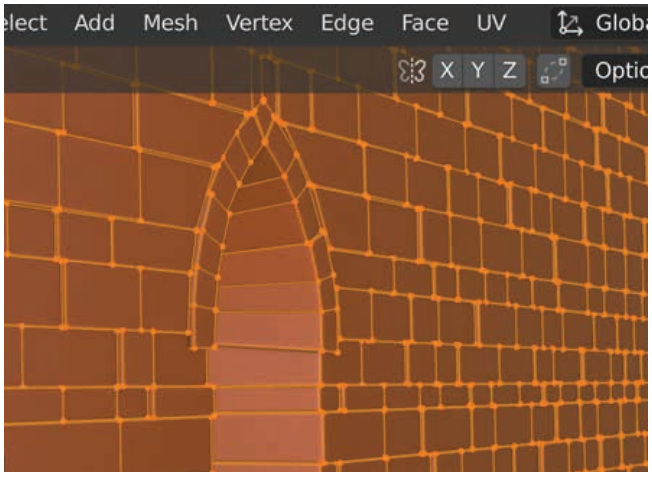


Press Shift+A and go to Mesh > Extras > **Wall Factory**.

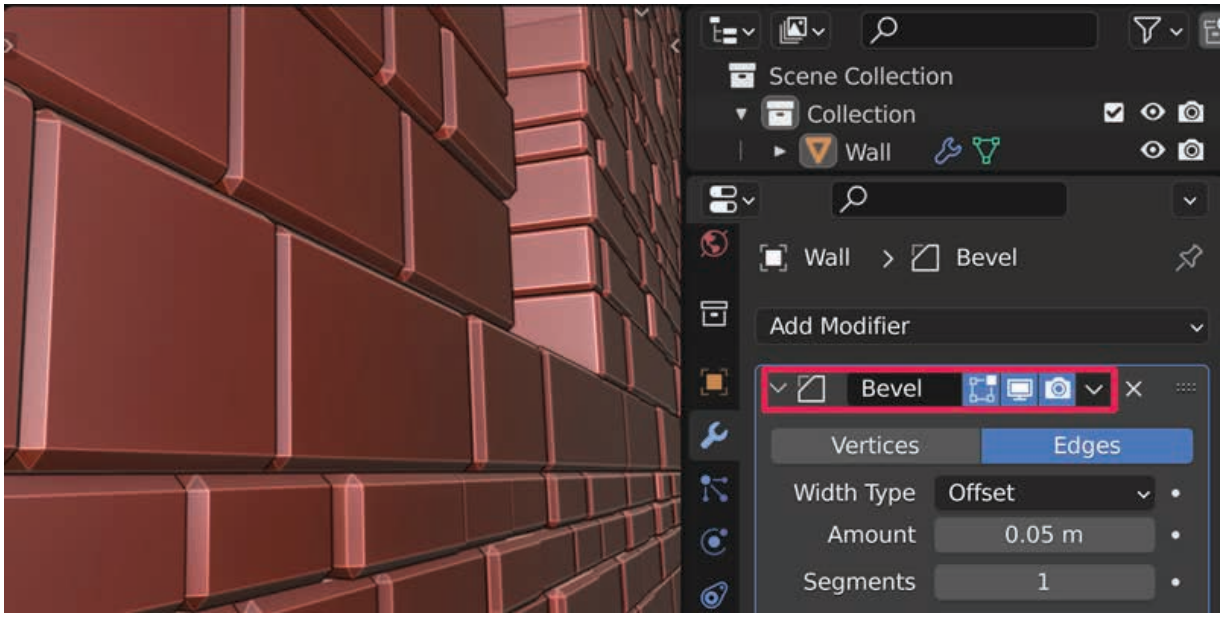


Increase the **End** value to 30 or more.

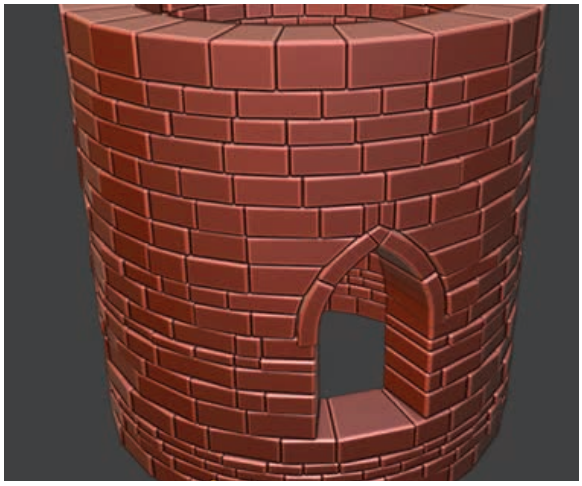
Select all in Edit Mode and scale the stones with **Alt+S** until they almost overlap (as shown below right).

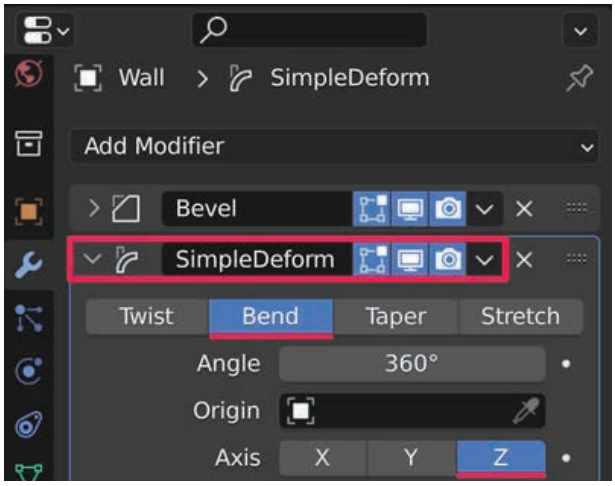


Add a **Bevel modifier**.

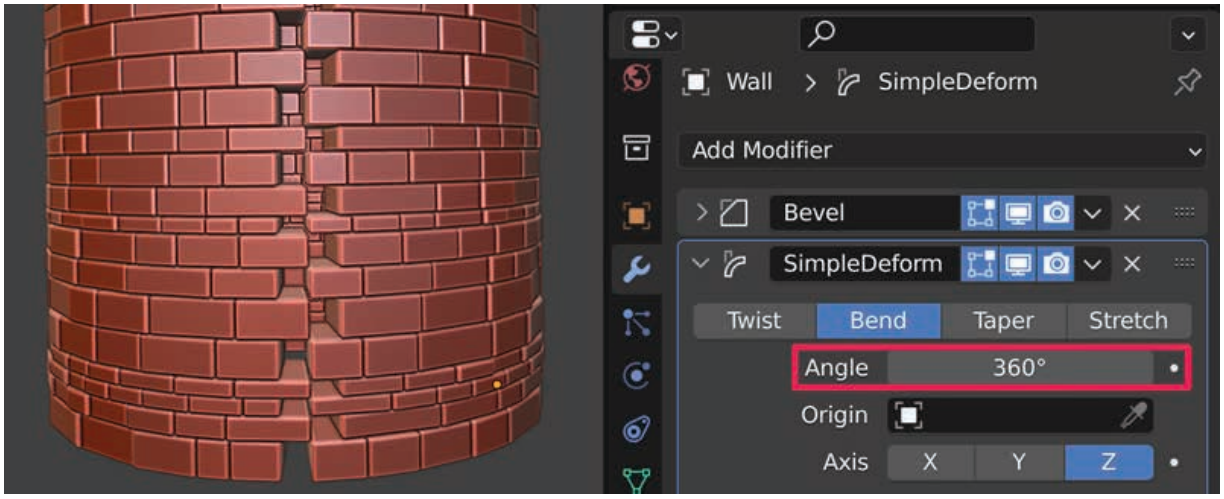


Add a **Simple Deform modifier** and set it to **Bend** 360 degrees along the Z-Axis.

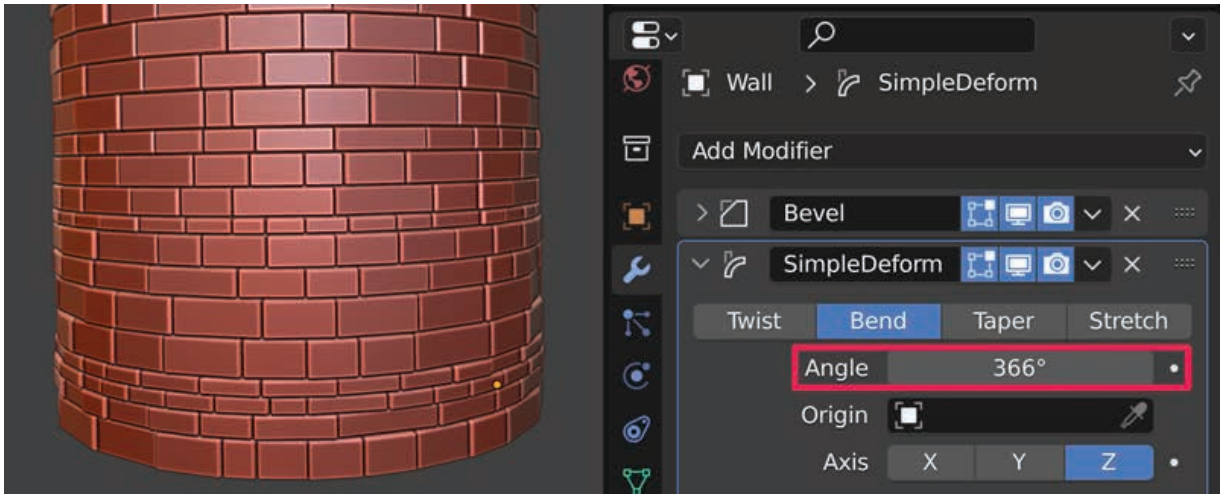




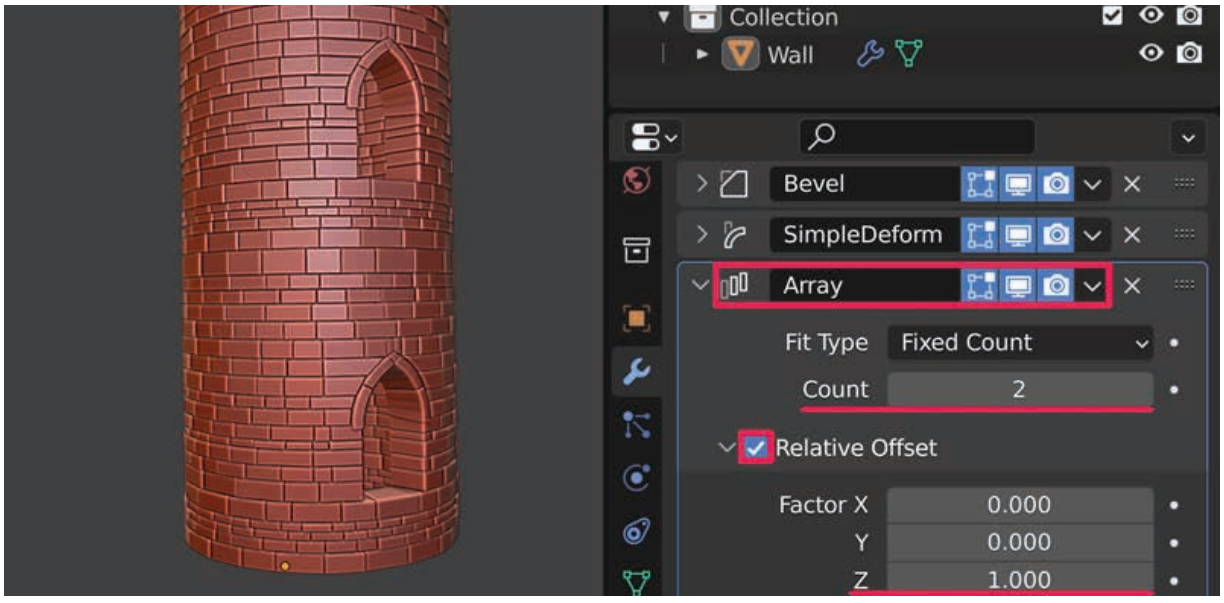
You can type higher values in this field if needed to close the loop.



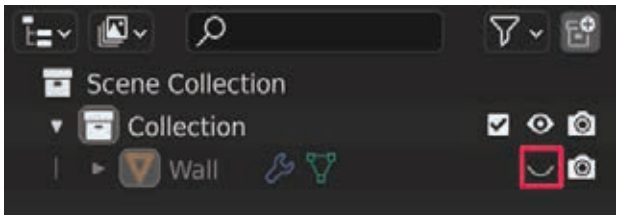
In this case the **Angle** value of 366° closes the tower cylinder.



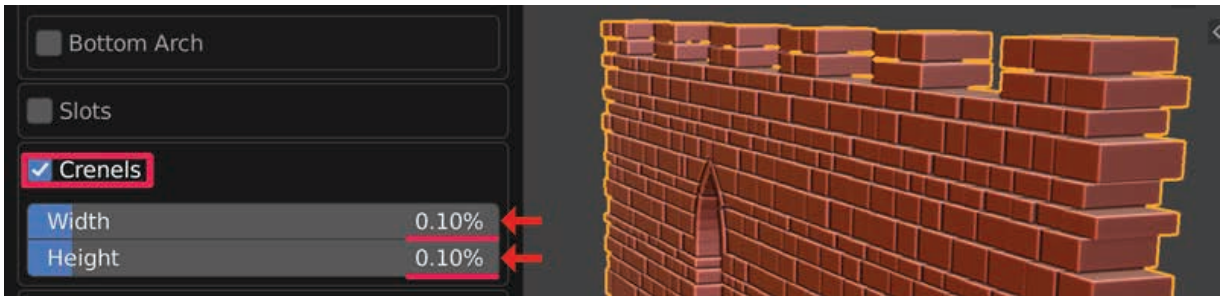
To increase the height of the tower, add an **Array modifier** with relative offset in the Z-Axis.



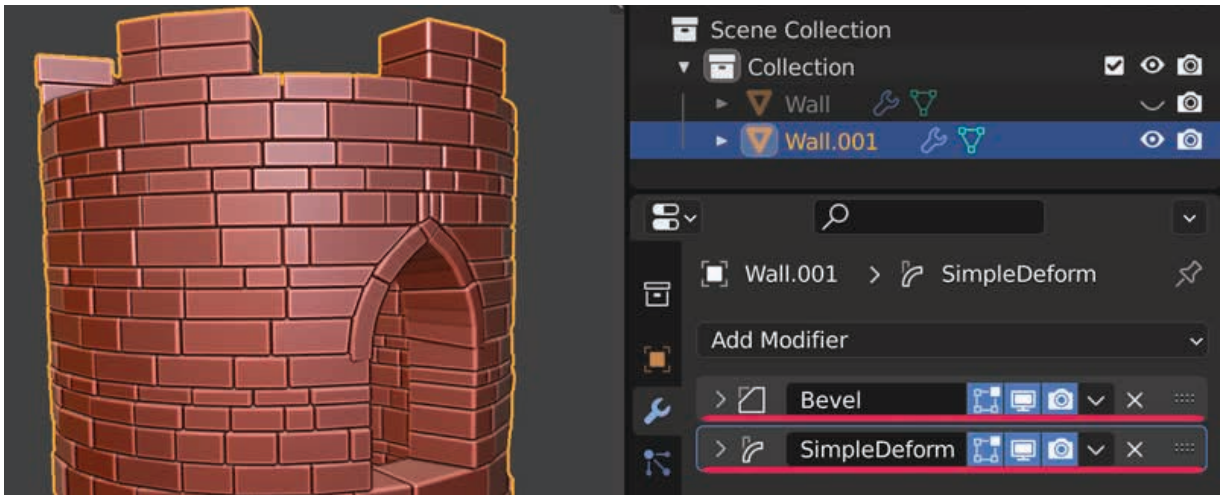
To add some variation on the top, hide the current object and add another wall.



Add some **Crenels**.

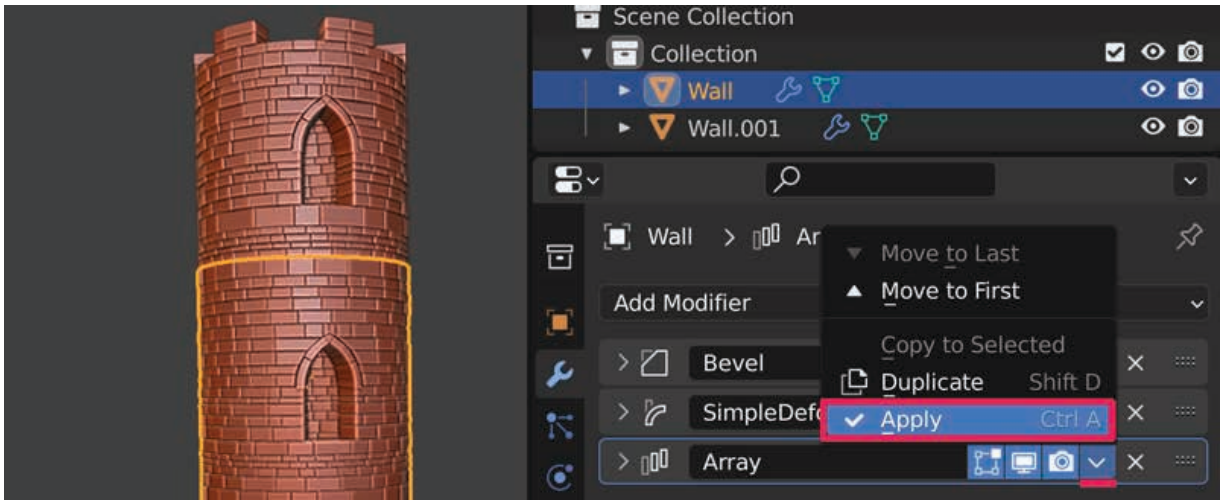


Repeat the previous steps to make this part circular and place it at the top of the tower.

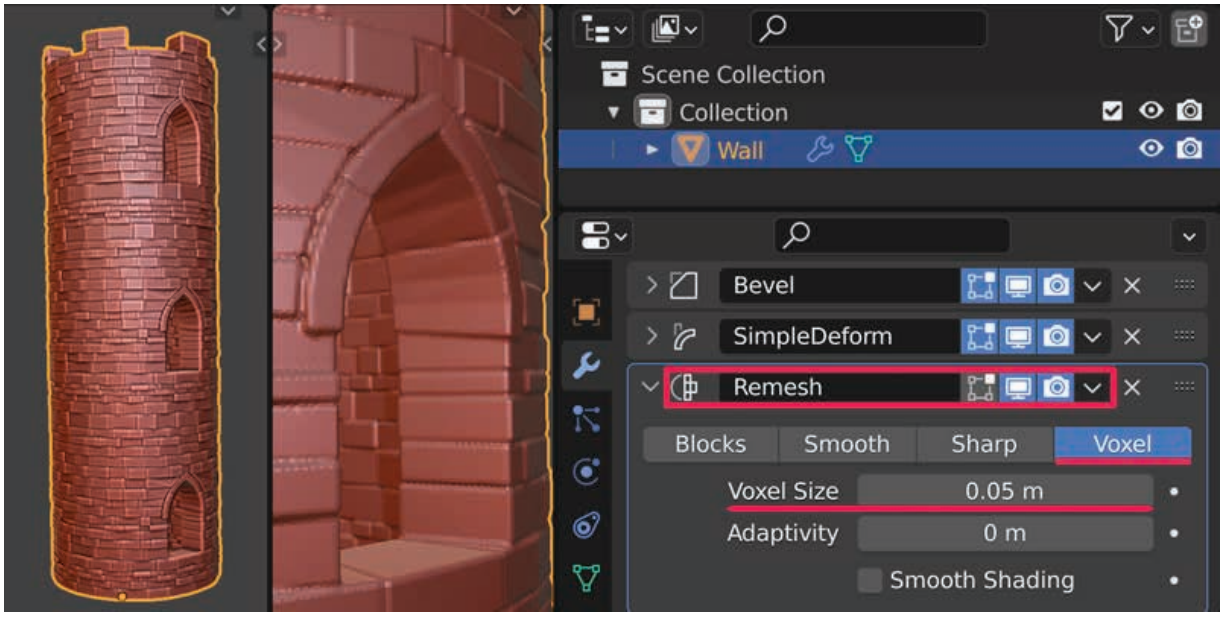


Then unhide the previous wall. Move the new part up along the Z-axis.

Apply the Array modifier and join the top and bottom parts with **Ctrl+J**.

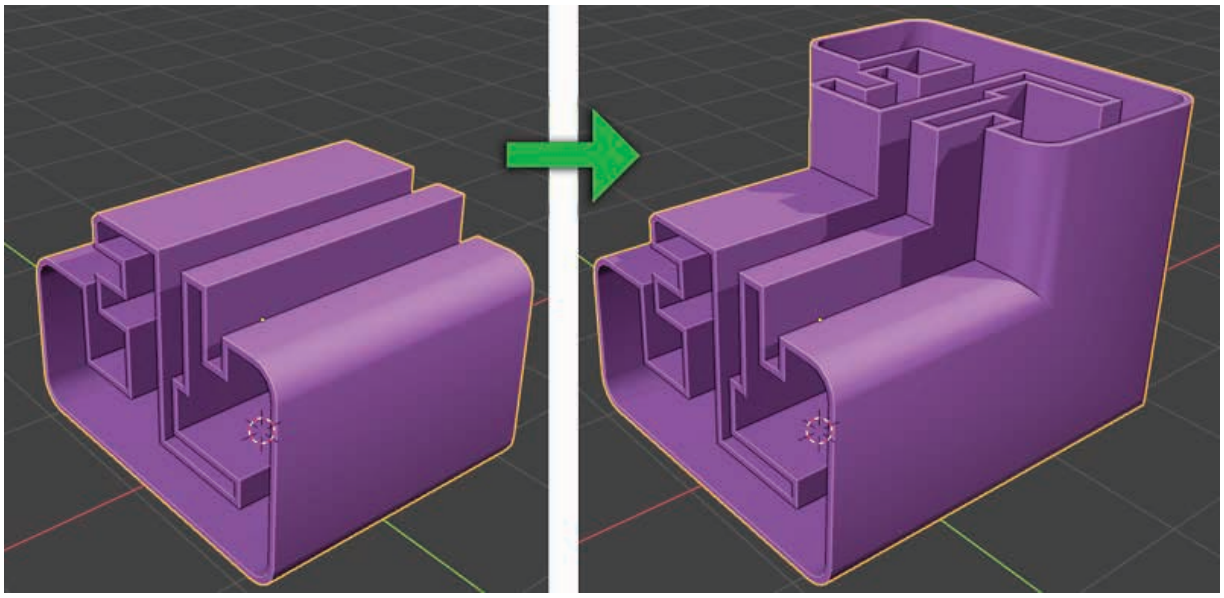


You can use a **Remesh modifier** set to Voxel to make the wall look a bit more organic and realistic.

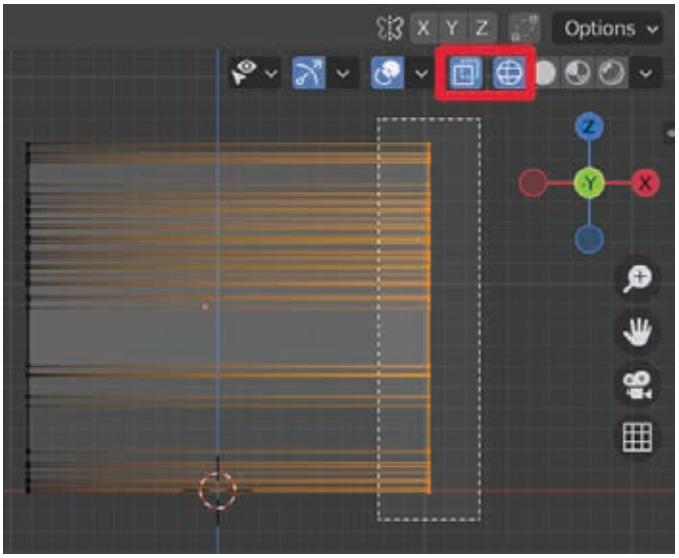


Extrude at 90° angles using Shear

This trick allows you to extrude at a 90° angle, which is essential for making for example window panes and other architectural models.



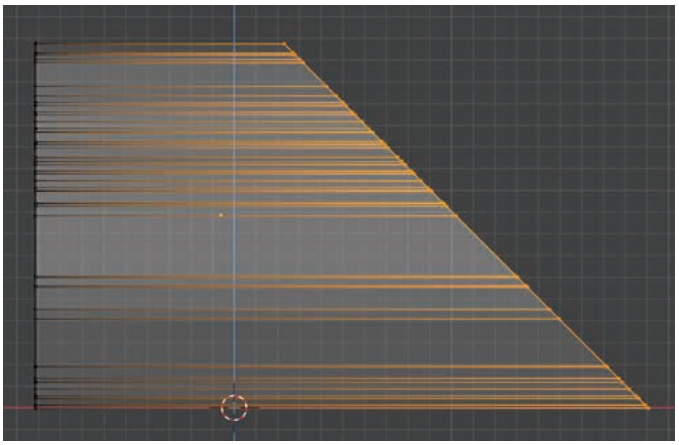
First select the vertices you want to extrude.



Drag a selection box around the vertices you want to select while holding the left-mouse button or **press B f**or “box select” and drag while holding the lef-mouse button.

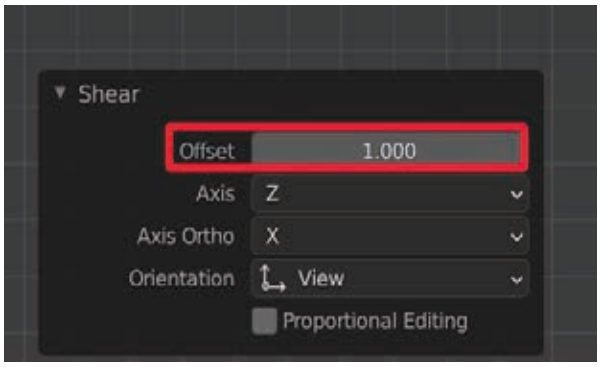
It’s easy to do this in **X-ray mode** so that you can see through the object, and from the **side view** so that you can easily select them.

Press Shift+Ctrl+Alt+S and then let go of the keys and **press 1 on the Numpad**.

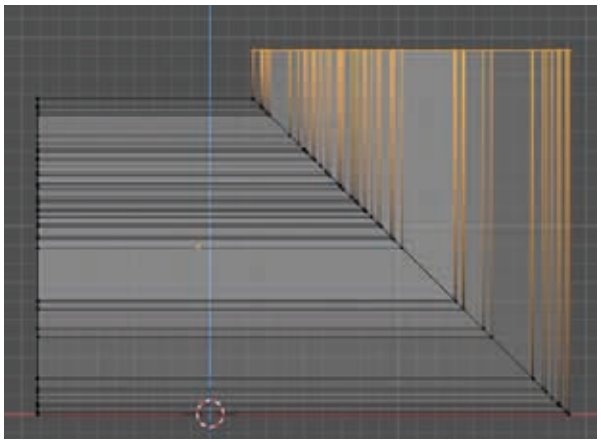


This “shears” the selection with an Offset value of 1, which is the same as a 45° angle.

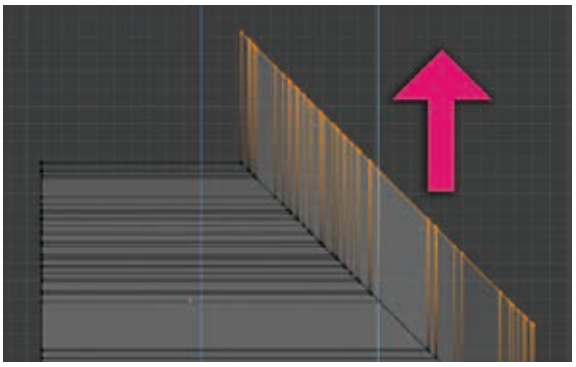
Alternatively you can **right-click and choose Shear,** then left-click to get a menu where you can set the **Offset** value to 1.



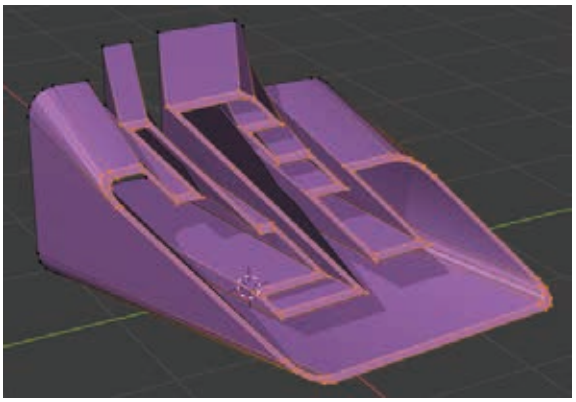
To flatten the extruded vertices, **press S+Z+[0 on the numpad] and [enter]** to scale the selection to zero on the Z-axis.



To extrude the sheared vertices up and get a perfect 90°, **press E and then Z twice** and move the mouse up.



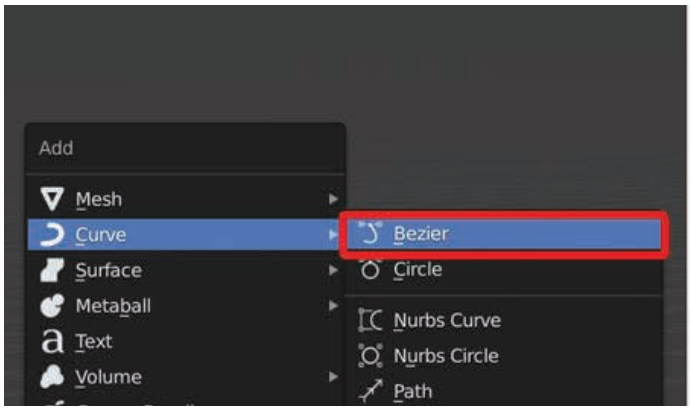
Note that by default, Shear uses **View Orientation**, meaning that you will need to look at the object from the side - otherwise the result will be a mess.

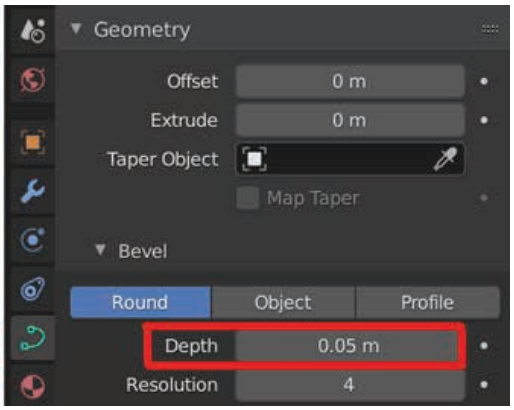


Drawing tapered ornamental details on surfaces

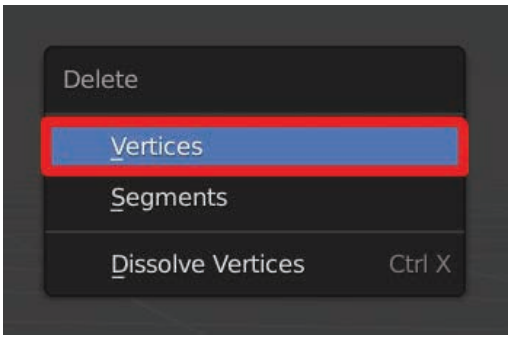
You can draw things like ornamental details and sci-fi cables on surfaces quickly using this technique.

Press Shift+A and create a **Bezier curve**. Give it some material and increase the bevel **Depth**.



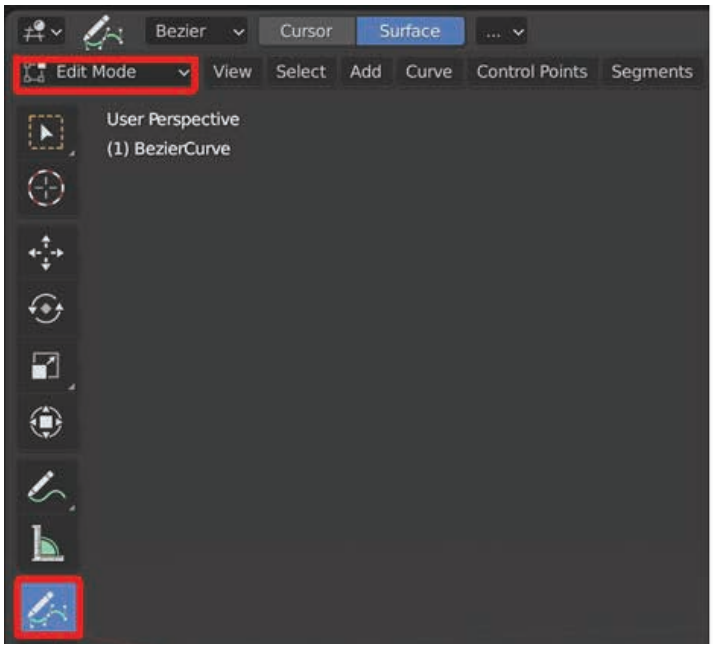


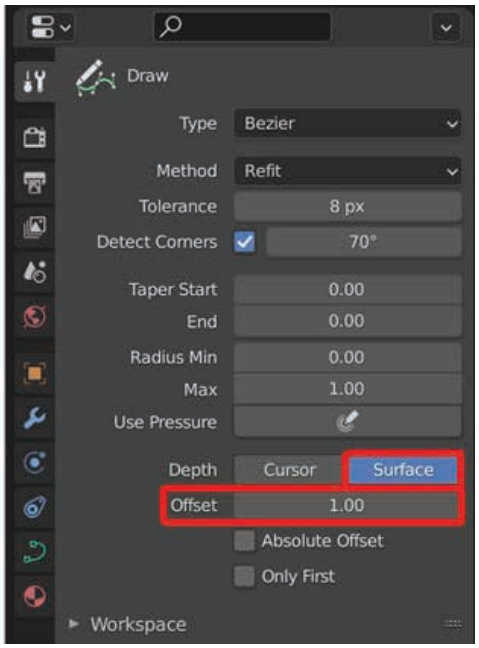
In Edit Mode select all by **pressing A**, then **press X** and choose **Delete Vertices**.



Then, still in Edit Mode, click on the **Draw tool** and set Projection Depth to **Surface**. If you want the Curves to intersect with the surface, leave **Offset at 0**.

For Curves that are on top of a surface and don't intersect with it at all, set the **Offset to 1** when drawing the curves.

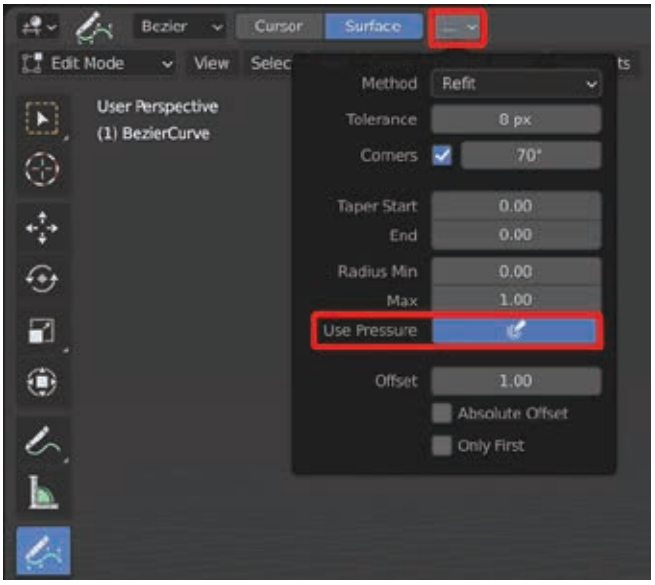




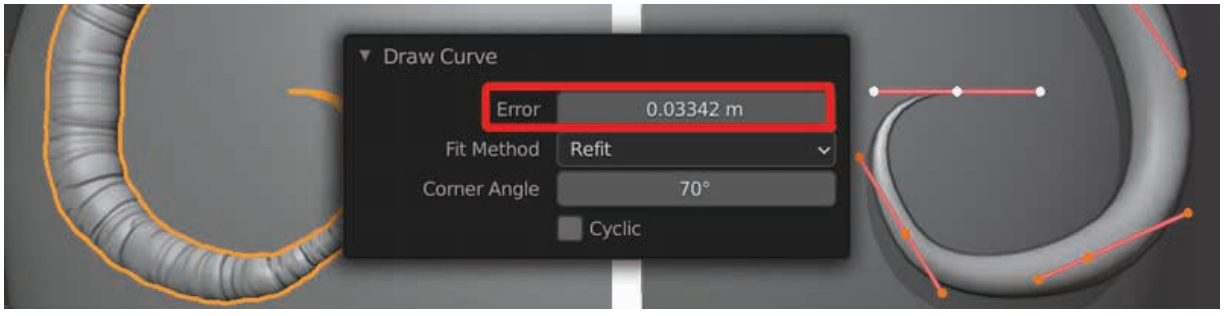
Now you can draw curves on the surface.

Curves come with generated UV's, which is really convenient.

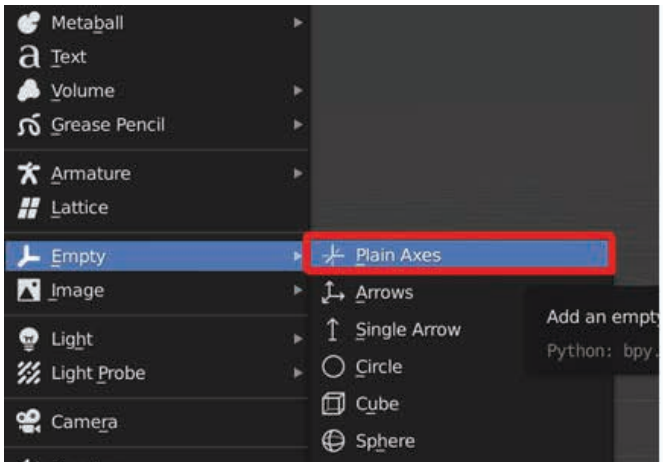
To enable pen pressure, so that you can influence the thickness of the curves by the pressure you apply with your tablet pen, you need to enable **Use Pressure**.

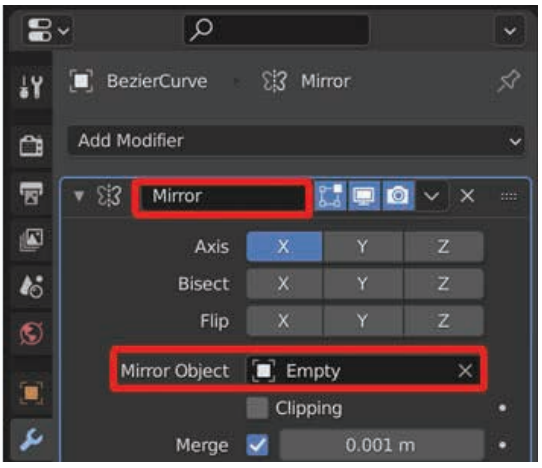


When using the pen pressure, you may find that your curves look like chubby, wrinkly rain worms for some reason. Fortunately you can change them by adjusting the Error value. This is quite sensitive so I recommend holding Shift and left-mouse-button dragging on this value to change it in smaller increments. You can also try a different Fit Method, either Refit or Split, for a different looking curve. You can get this menu by pressing F9 as well.

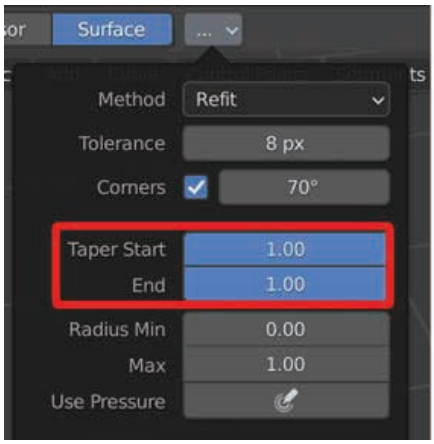


To draw symmetrically mirrored curves, add an **Empty** to the scene (and make sure it's at the World origin). Then add a **Mirror modifier** to the Curve, and add the Empty to the **Mirror Object** field.

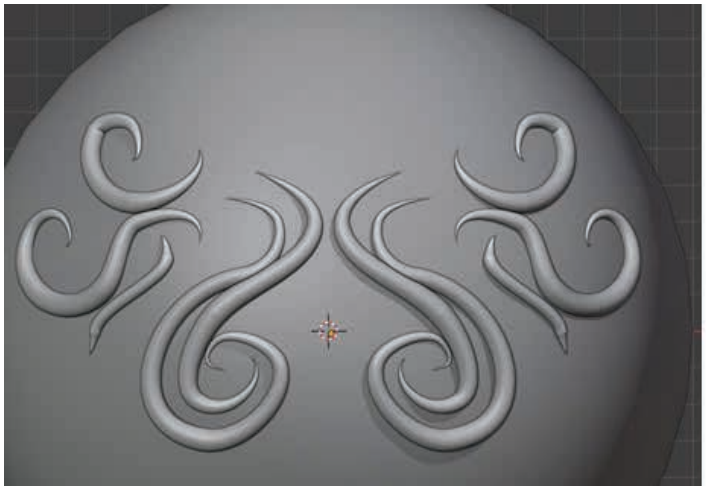


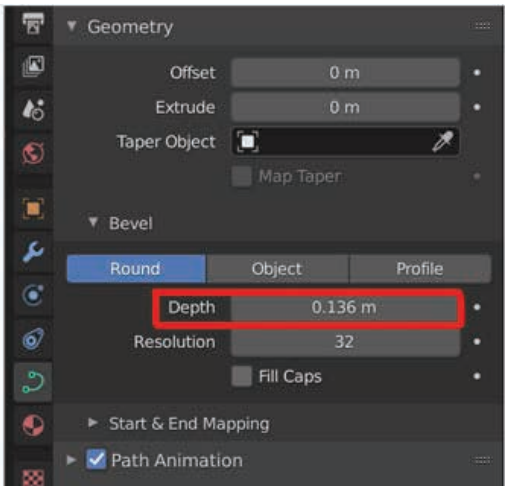


If you don't have a tablet pen, but you still want to have the look of the tapered thickness - you can achieve this with a mouse as well. Just set both **Taper Start and End to 1**.

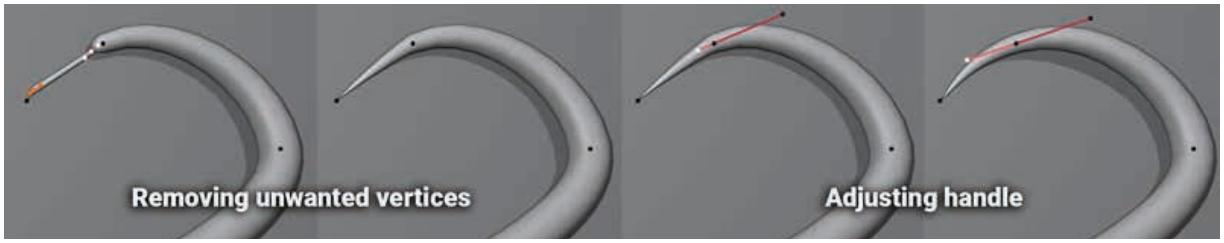


In my experience this will give thinner curves than if you use the pen, but you can still increase the thickness after drawing the curves by increasing the **Depth** value.

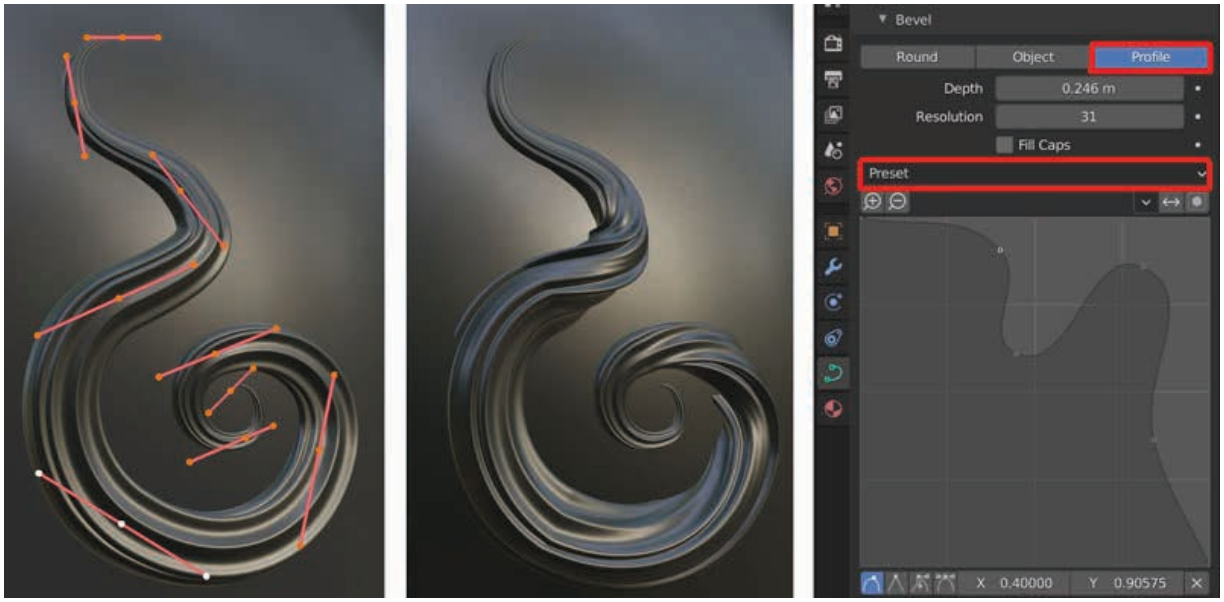




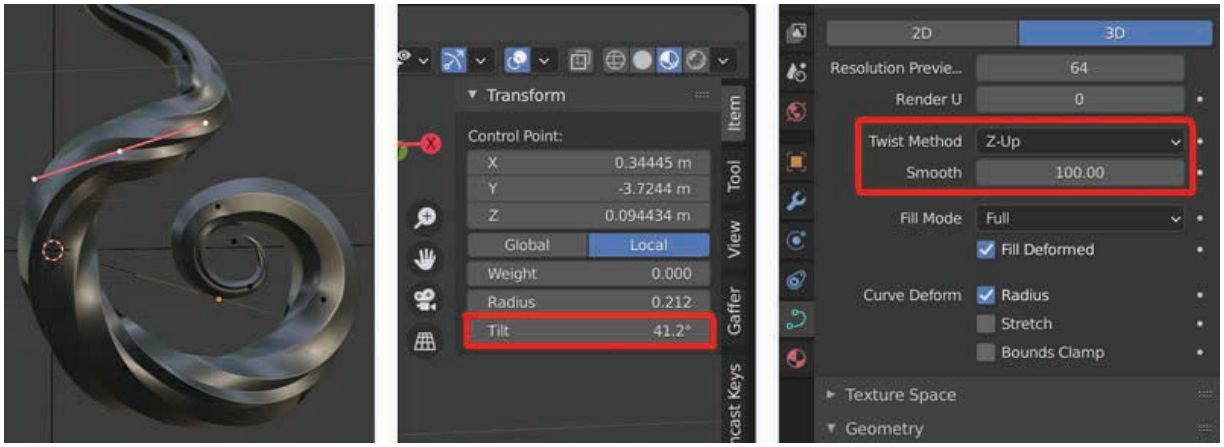
In fact, you can use this method with your tablet pen as well, just turn off **Use Pressure** and set **Taper Start and End to 1**. You'll get perfect tapered curves every time. You can still remove some unwanted vertices by selecting them and **pressing Ctrl+X**, which often solves issues with ugly tapered curves. Then you can select and move some handles to make the curve perfect.



You can also use the **Custom Bevel Profile**, either using one of the presets or creating your own, to add more visual interest. If you want to rotate the curve to accommodate the bevel profile, select one or more vertices and **press Ctrl+T**, then move the mouse.

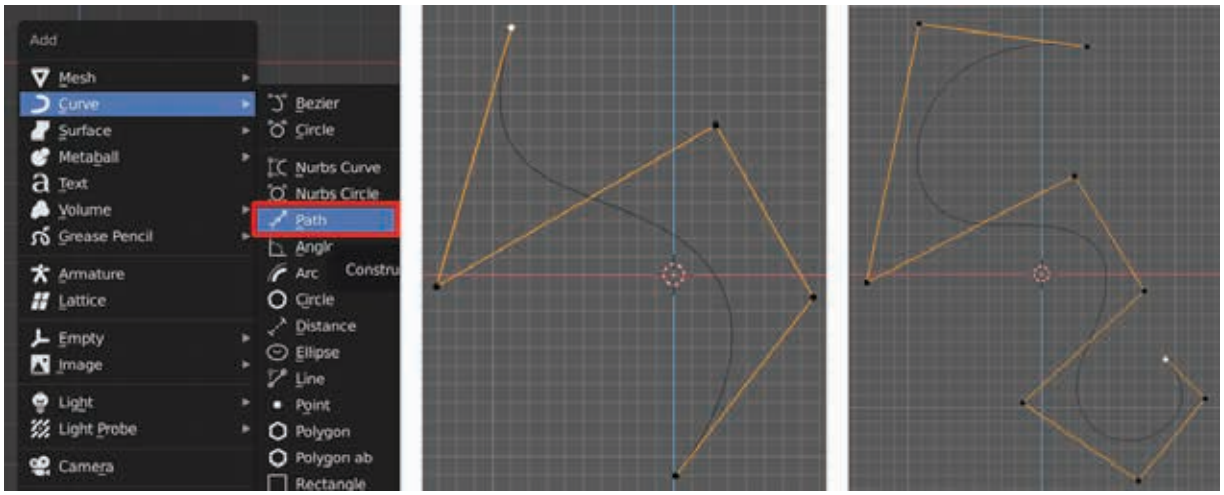


I find that you can create more pleasing twist results by changing **Twist Method to Z-up** and increasing the **Smooth** value, then tweaking the individual vertices some more. You can **press N** to open the Option panel and tweak the **Tilt** value for each vertex until it's perfect.

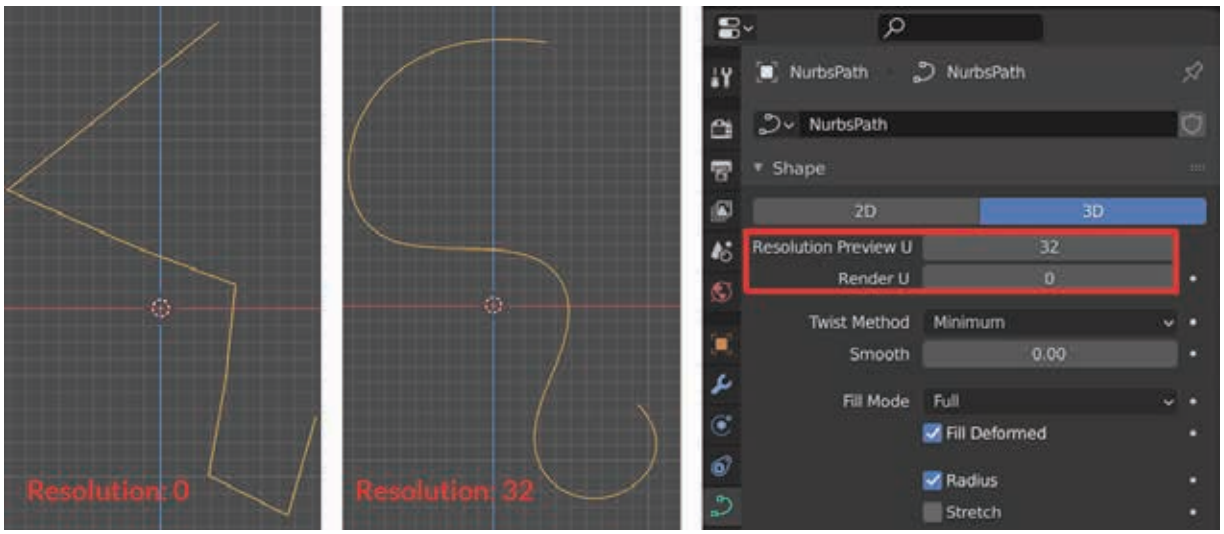


Ornament Modeling with Custom Curve Bevels

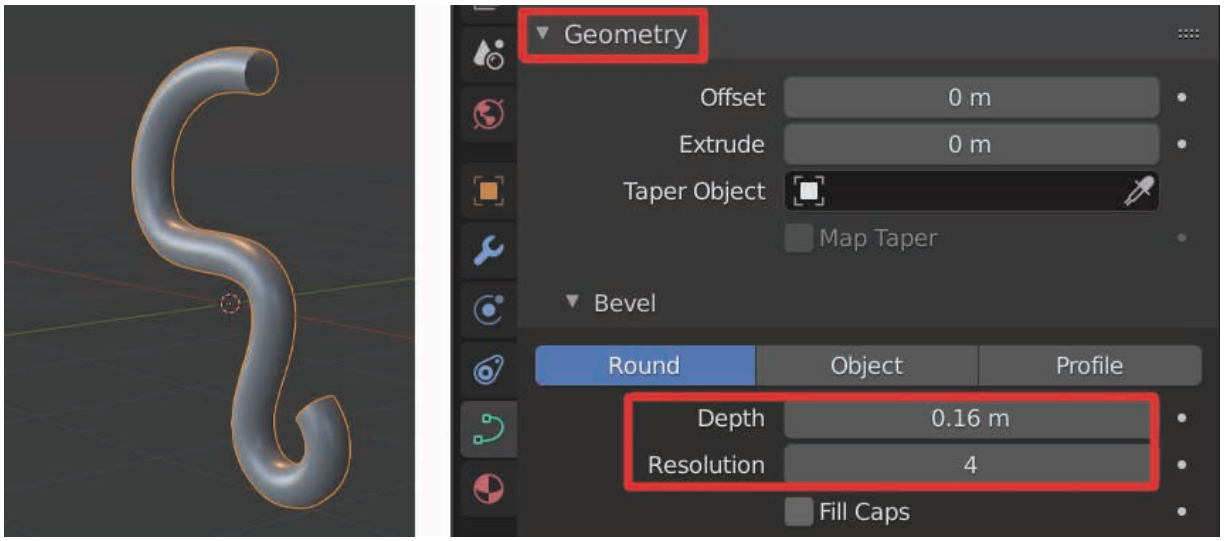
Create and extrude a Path in the shape you want. You can extrude by selecting a vertex and **pressing E**, or by **holding Ctrl and right-clicking**.



The **Resolution U** value controls the smoothness of the Curve. Preview is for the curves in the 3D viewport, Render only applies to the renders. In other words, you can set the Render resolution high and the Preview resolution low, if you need to keep the scene optimized for your computer.



In the **Geometry** section you can find the Bevel Profile settings. Increase Bevel **Depth**. This adds thickness to the Curve. Bevel Resolution controls how detailed the bevel is. The more detail you add to your Profile, the more Bevel Resolution you need.



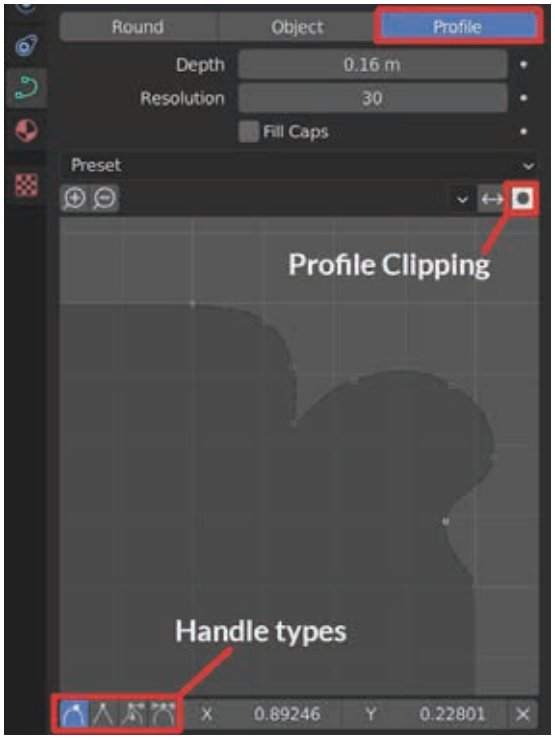
Click on **Profile** to get access to the custom profile curve.

Click and drag on the Profile curve to create handles to add detail to it.

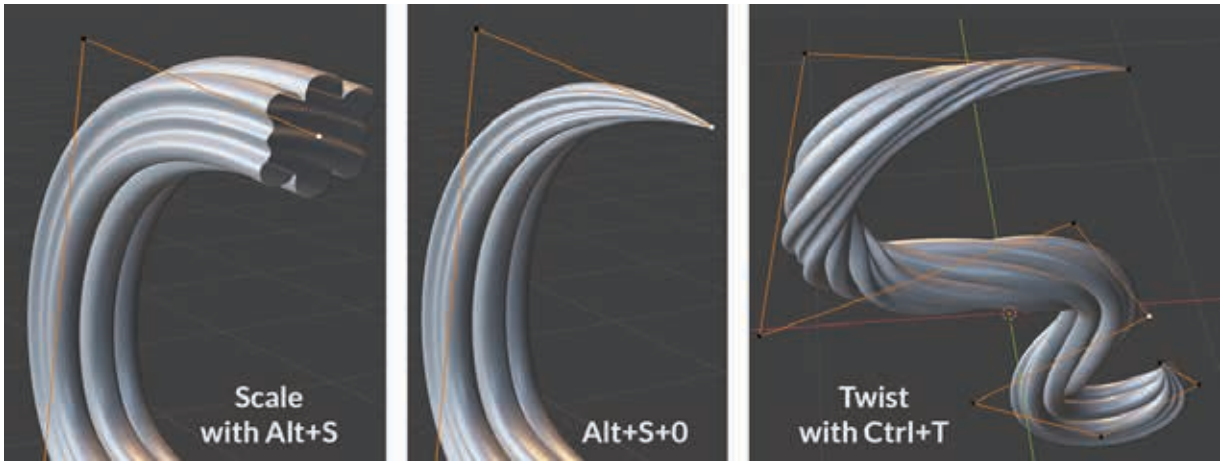
You can change the handle type by clicking on the icons below the curve Profile.

To be able to zoom out of the default workspace of the Profile, click on the Profile Clipping icon.

To change thickness, select a vertex and **press Alt+S.**



To taper the ends to a point, select those vertices and **press Alt+S+0**. Select a vertex and **press Ctrl+T** tç twist the curve, which also looks interesting.



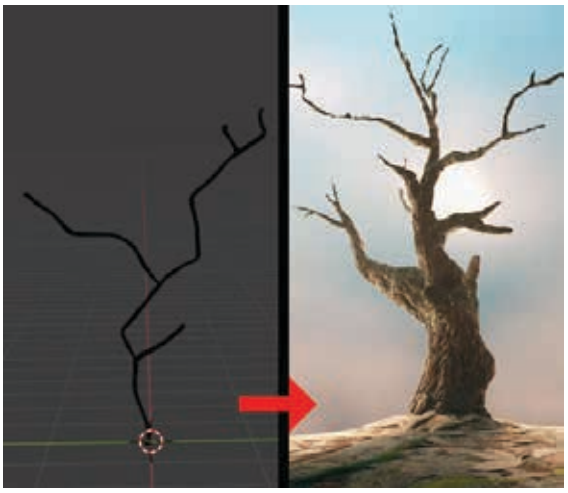
ENVIRONMENTS

Draw in 3D with Grease Pencil and Convert to Mesh

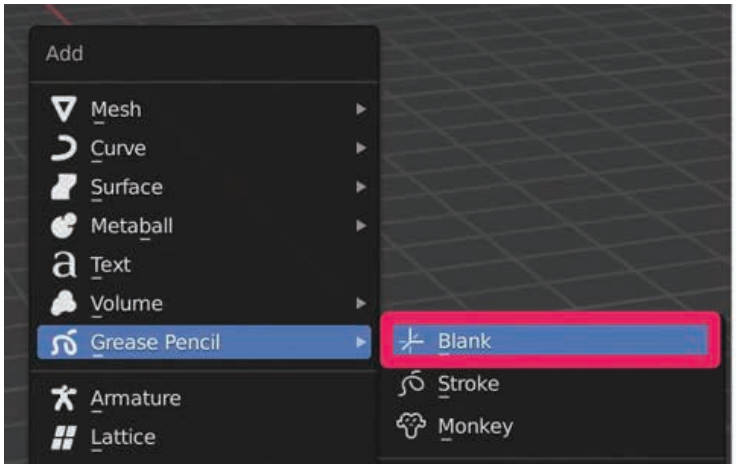
You can use Grease Pencil to sketch concepts in 3D.

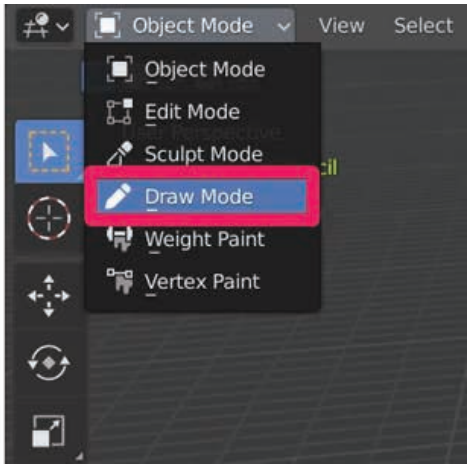
This can be a quick way to crease a base mesh for sculpting organic objects.

For example, a tree trunk.

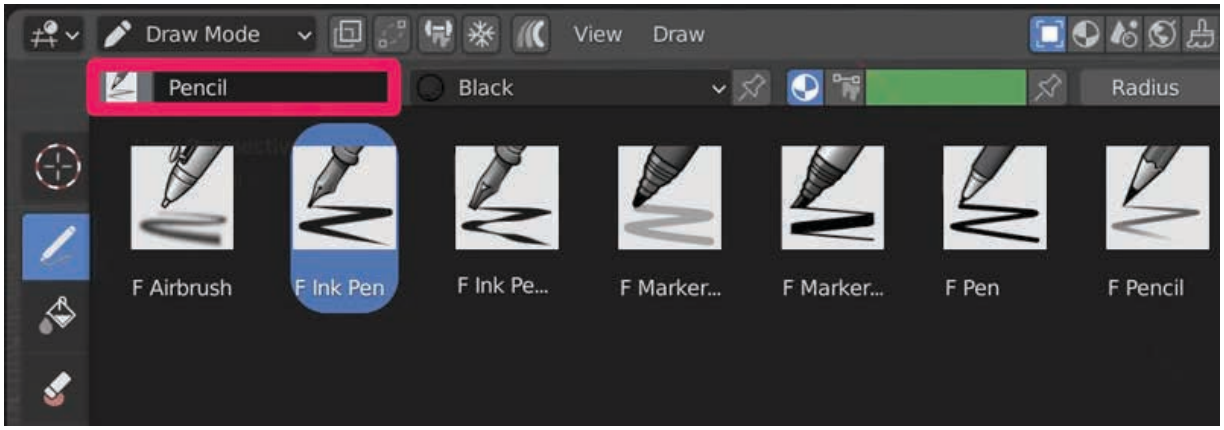


Press Shift+A and add a **Blank** Grease Pencil object. Switch to **Draw Mode**.





If you want you can choose a different pencil. It doesn't make any difference in this case as it will be converted to a Path later, but personally I find it more satisfying to draw interesting-looking strokes.

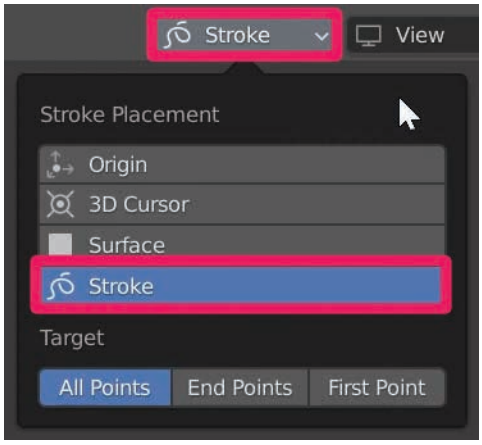


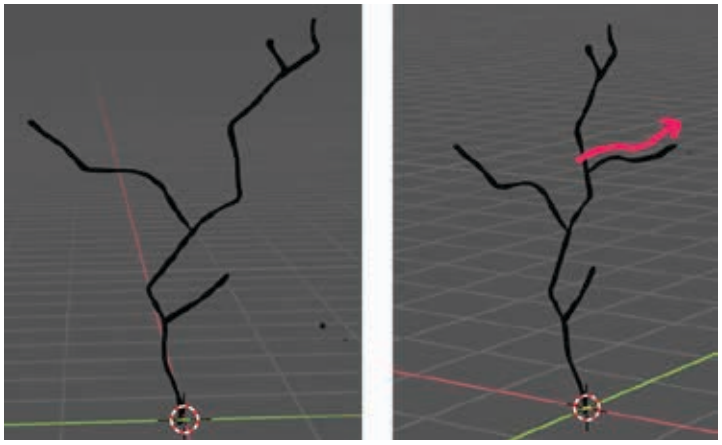
Set **Stroke Placement** to **Stroke**.

That way, if you draw a line it will snap to where any other line is that it crosses from the current view angle.

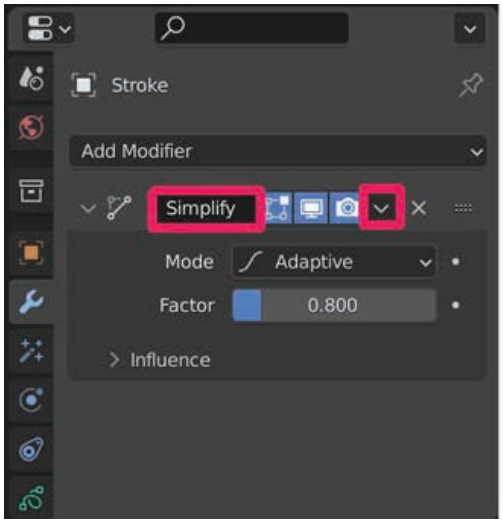
Changing the view angle to accommodate your stroke placement is therefore an important element of this technique.

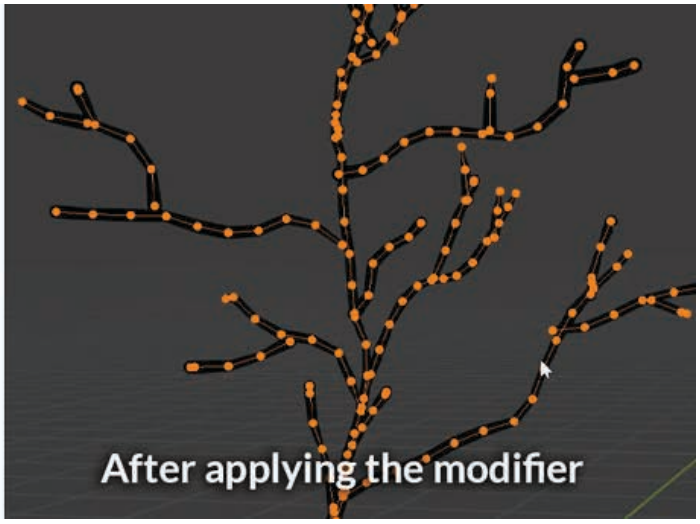
As long as your stroke overlaps a previous stroke, it will be snapped to that stroke in 3D space.





After drawing your Grease Pencil art, you can add a **Simplify modifier** to reduce the complexity. Apply the modifier by choosing **Apply** from its drop-down menu. In **Edit Mode** you can see that there are now significantly less points, making it easier to select and move individual points of the stroke in case you want to do some refinement.

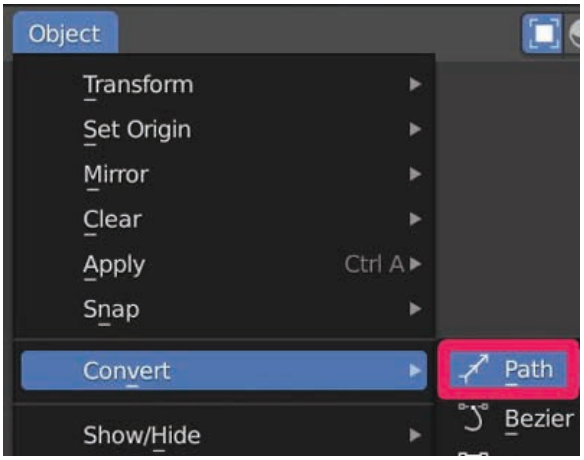




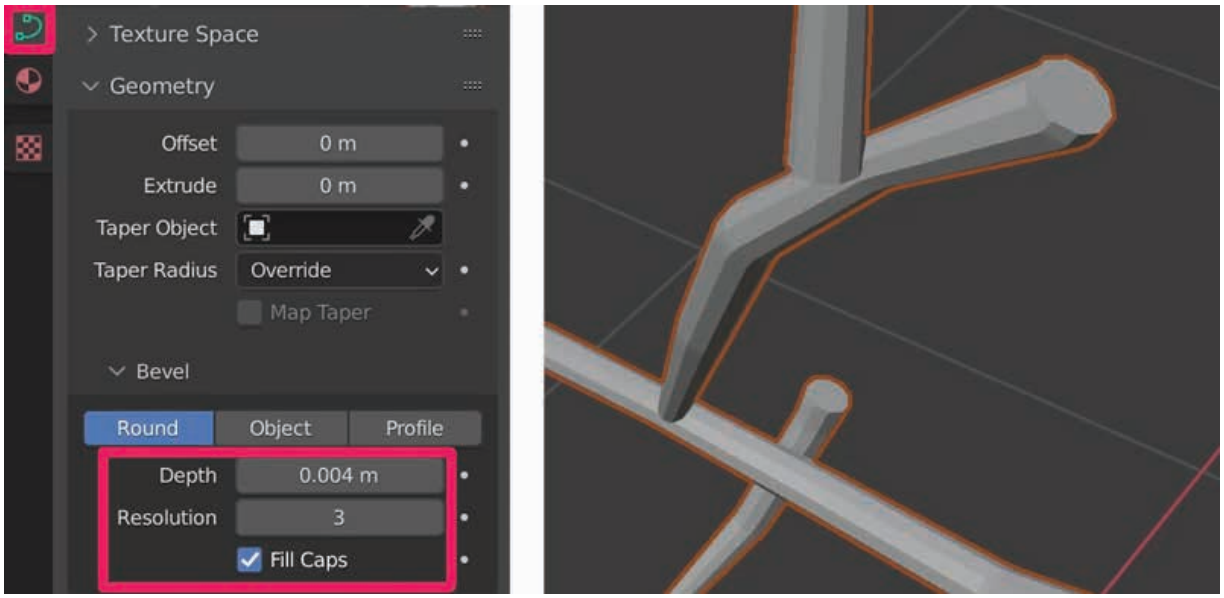
In Object Mode select the Grease Pencil object.

Go to **Object > Convert > Path** to convert it to a Path.

You can't convert directly to a Mesh, but that's okay as the Path offers some options that are very useful.



Select the new Path object in the Outliner. Also in the Outliner, you can hide or just delete the Grease Pencil object. Give the Path **Depth**, increase the **Resolution** and **Fill Caps**.

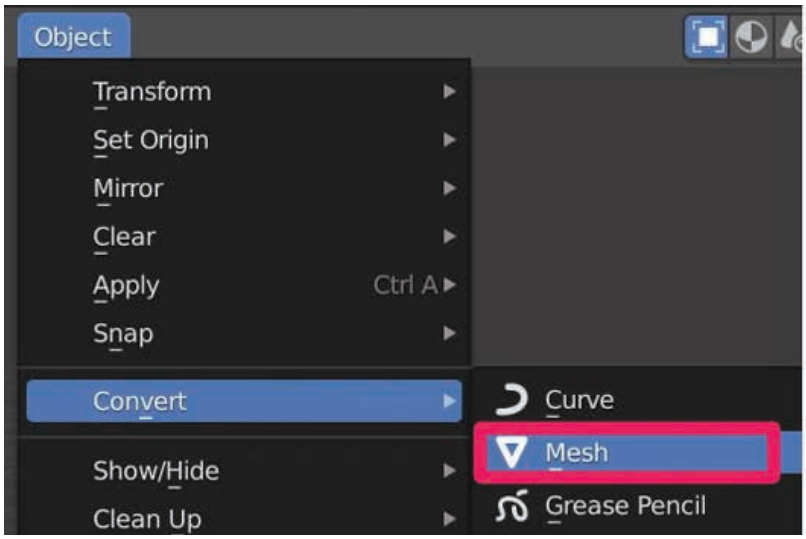


You can select vertices and scale them up with **Alt+S**.

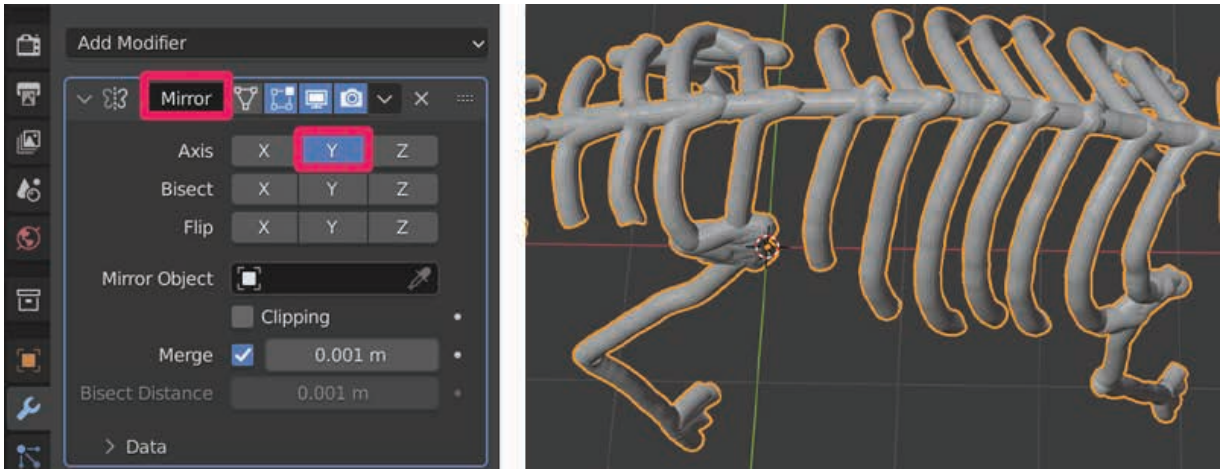
Using Proportional Editing here can be a good idea.

You can toggle it on/off by **pressing o**.

Go to **Object > Convert > Mesh** to convert the Path to a Mesh. Next we will look at refining this using sculpting.

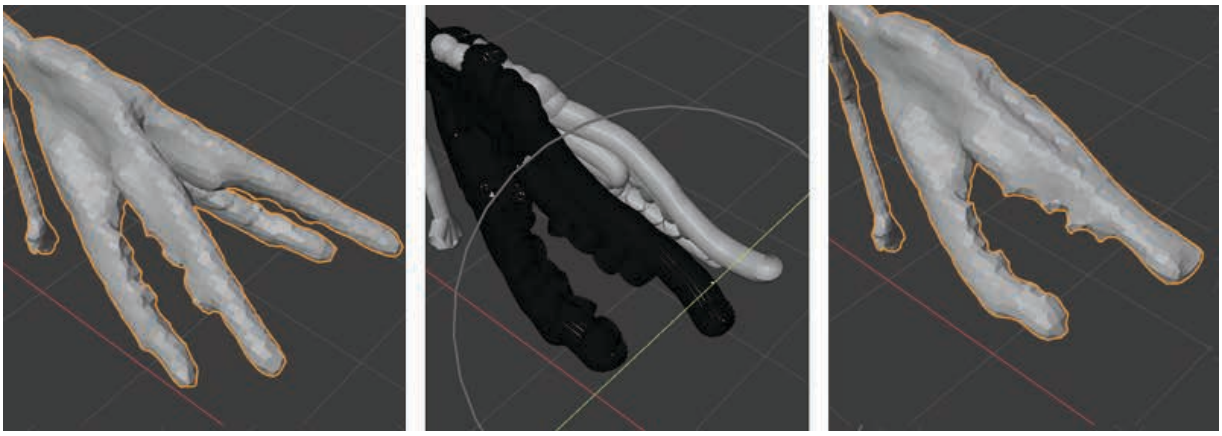


In case you're working on a symmetrical model, add a **Mirror modifier** to the mesh and set it to the right **Axis**. If necessary, you can also check Bisect for that axis. This option removes the other half, so you don't have to go in Edit Mode and remove vertices manually.



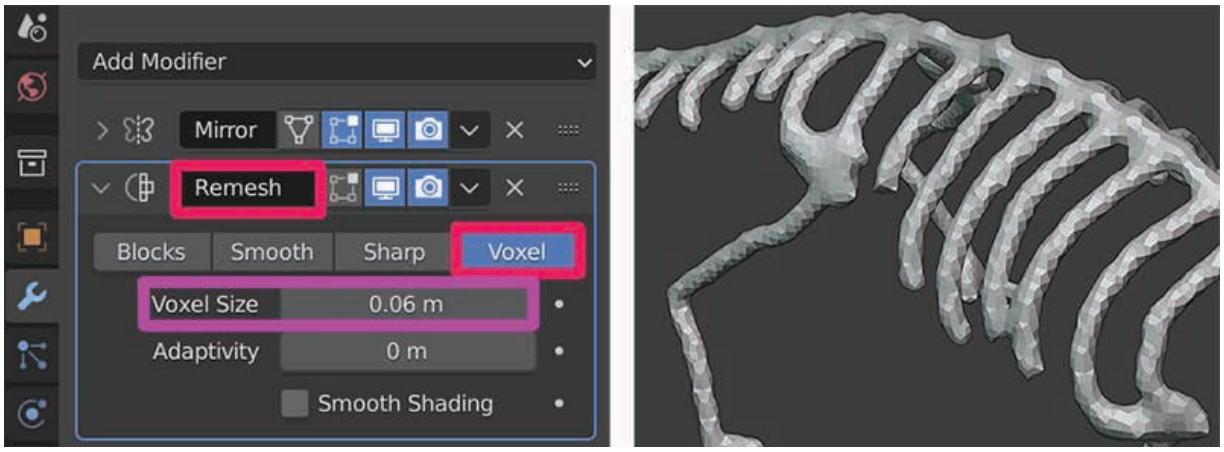
A dinosaur skeleton, taking advantage of the Mirror modifier for symmetry.

Move vertices in Edit Mode on the mirror axis if necessary to make the mirror work better. You can also still do this after the next step, adding the Remesh modifier.

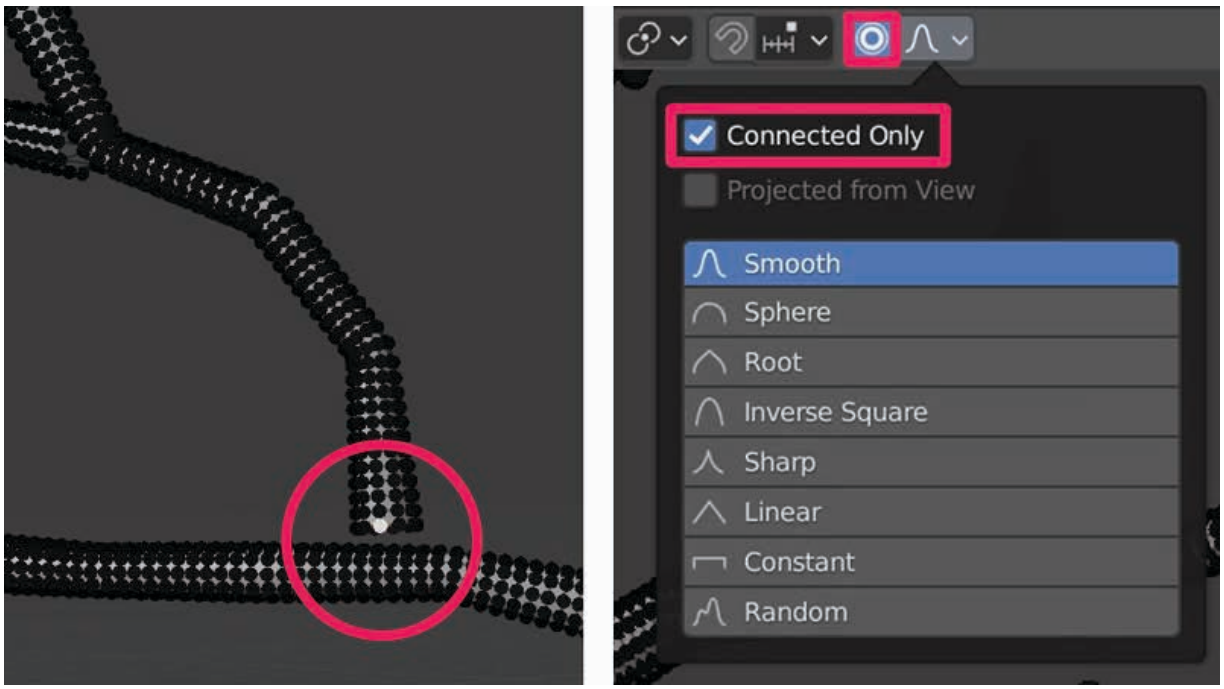


Merging the mirrored head parts by moving vertices

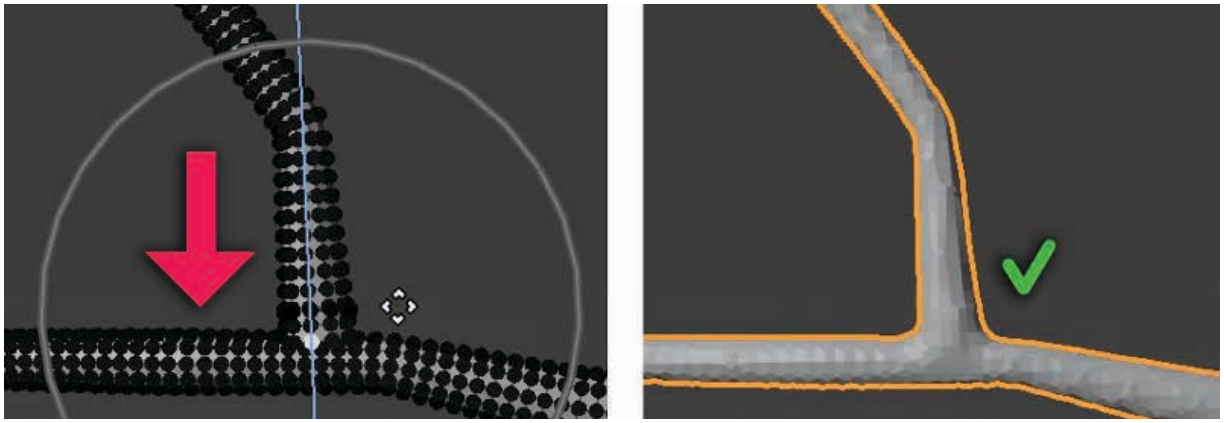
Add a **Remesh modifier**. You may have to decrease **Voxel Size**, but be careful - if you go too small with this value, it will take a long time and your computer might freeze. So just do it incrementally: go from 0.1 to 0.05 first, then try 0.01 and so on until you get a decent amount of detail. More detail is not necessarily better - for a base mesh meant for sculpting, you don't need it.



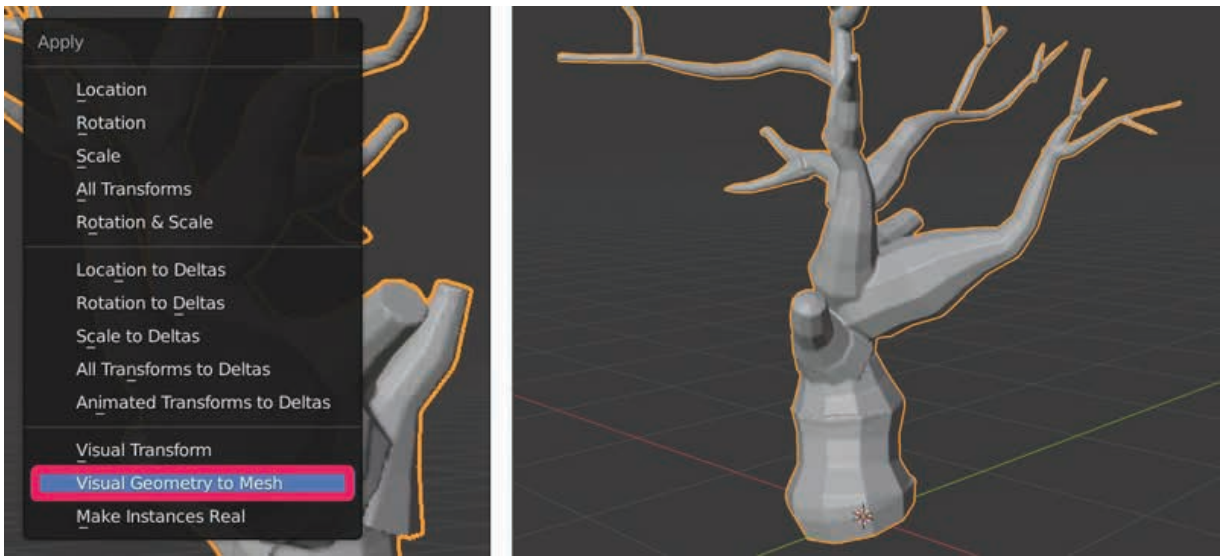
As it's a modifier, you can still enter Edit Mode and move vertices to get a different Remesh result. This is convenient for example to fix places where meshes should connect, but don't. I recommend using **Proportional Editing** with "**Connected Only**" turned on so you can grab and move the loose parts.



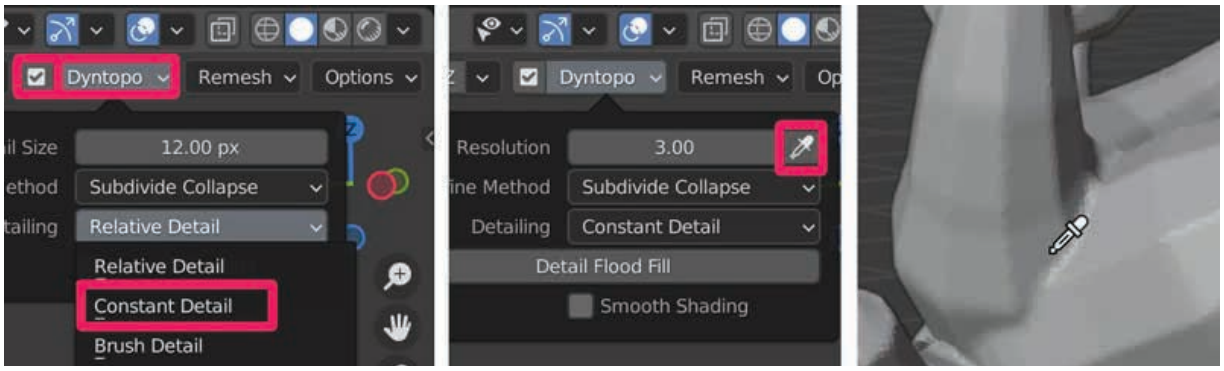
As long as the mesh overlaps in Edit Mode, the Remesh modifier will take care of the rest.



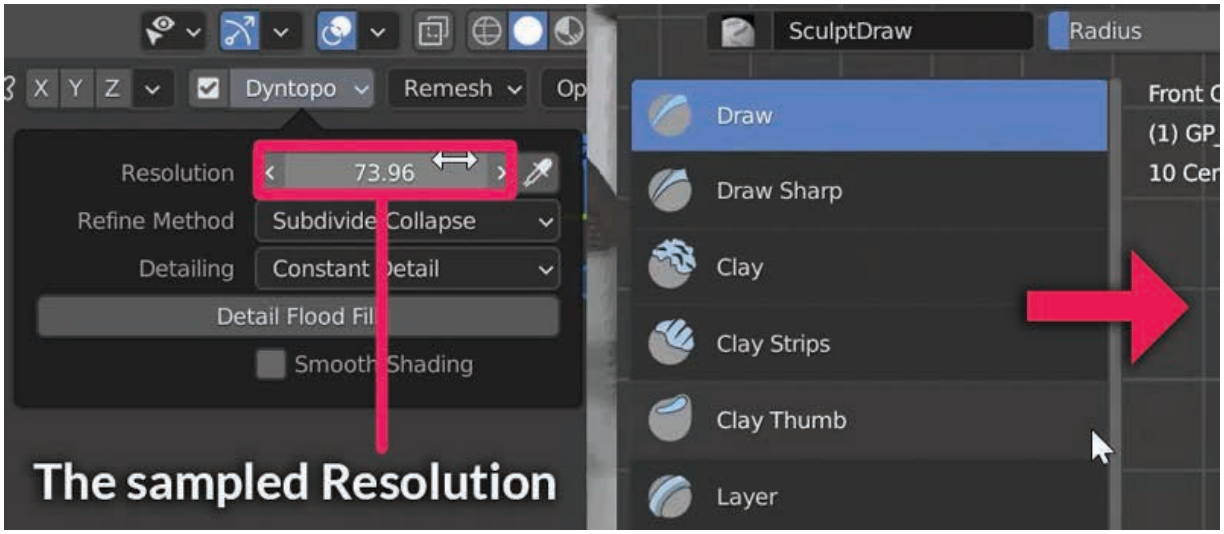
Press Ctrl+A and choose **Visual Geometry to Mesh**. This applies all modifiers at once.



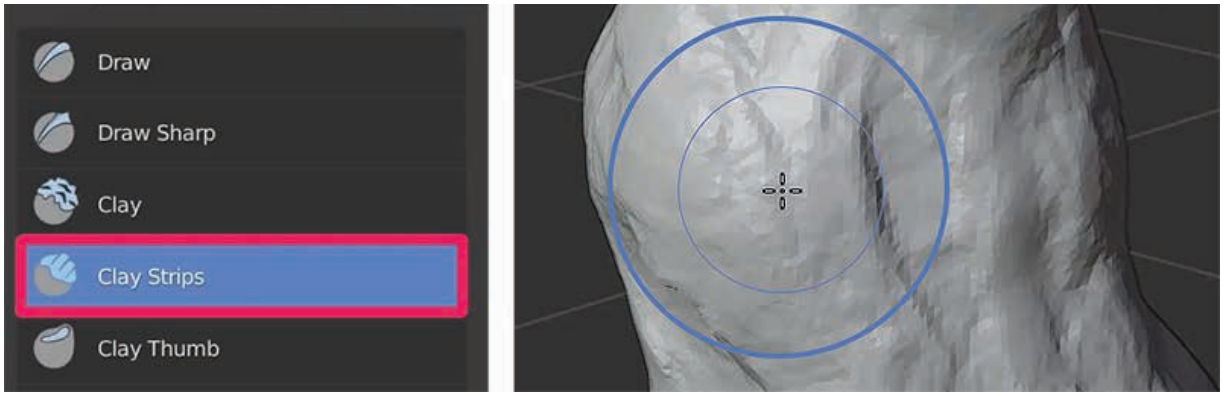
Now you can start to further refine this by sculpting. Go to **Sculpt Mode** and enable **Dyntopo**. Choose **Constant Detail** and sample the current **Resolution** of the Mesh with the eyedropper tool.



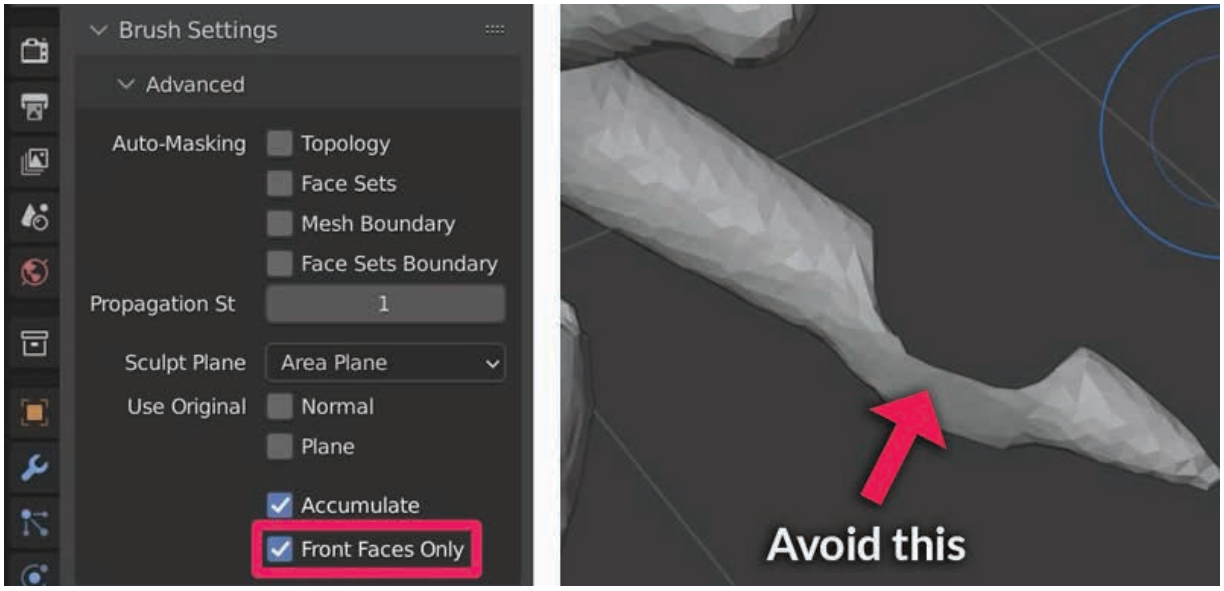
Double the sampled value as a starting value for sculpting. If that gives a brush resolution that's still too low, increase it incrementally until you get what you need. Expand the Tool panel with the sculpt brushes by pulling the edge to the right, to see the names. That way it's easier to find what you need.



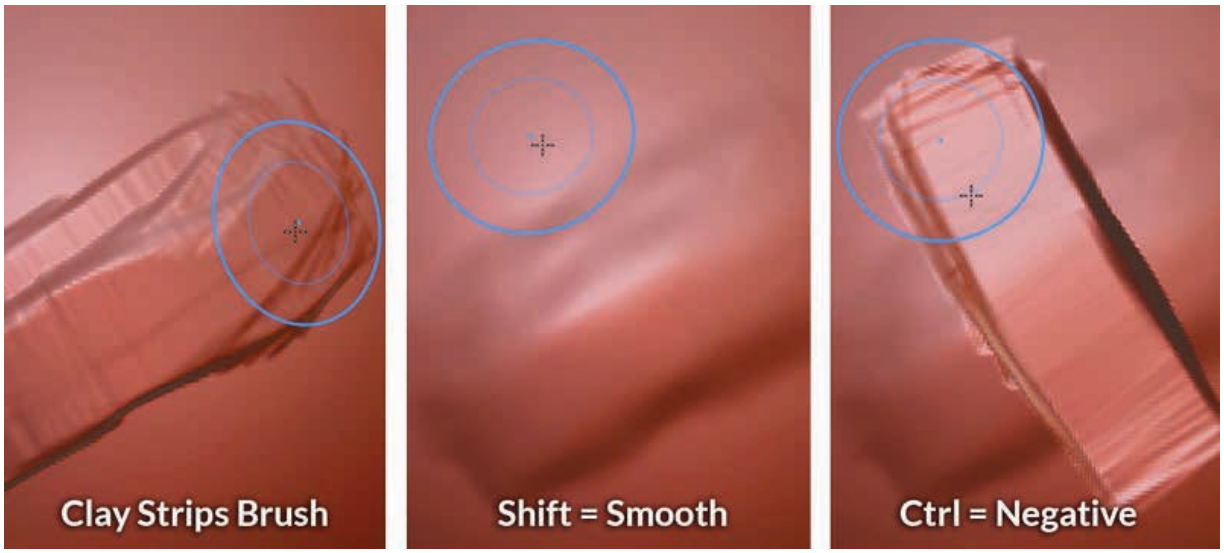
Choose the **Clay Strips brush** for starters. I like this brush as it's good for building up detail.



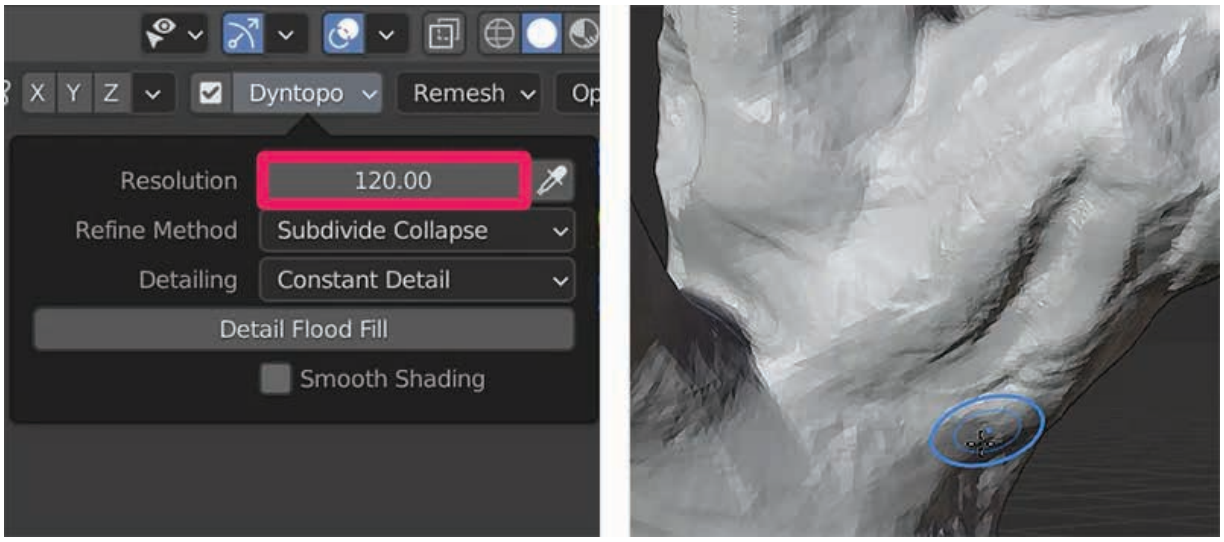
Under **Advanced** in the Brush Settings, enable **Front Faces Only**. This makes sure you don't accidentally mess up the backside of where you're sculpting.



Then, build up detail with the Clay Strips brush and hold Shift whenever you want to smoothen something. Hold Ctrl when you want the brush to go in the negative direction.



Start with a big Radius, then after a while increase the **Resolution** of Dyntopo and continue adding details with a smaller Radius.

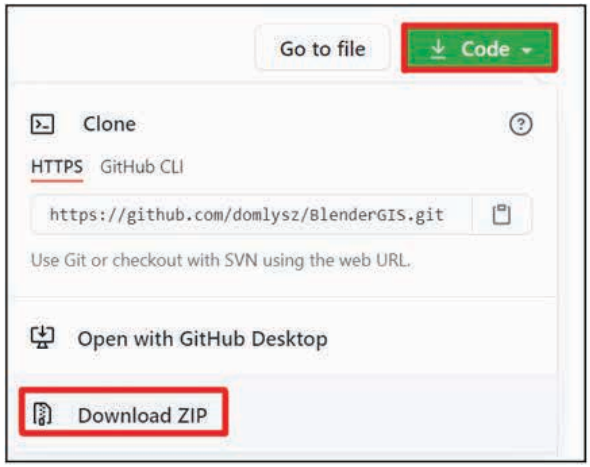


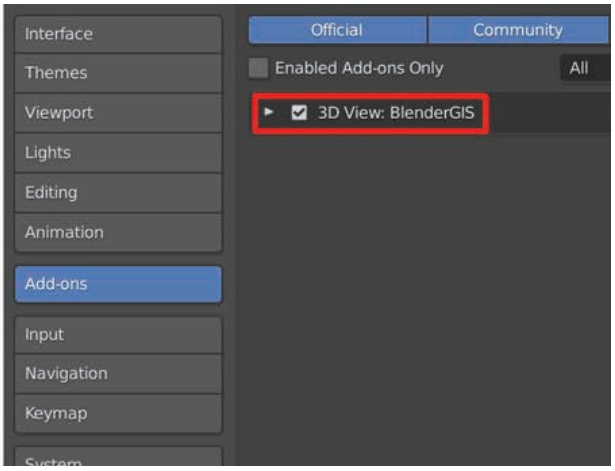
Finally, we end up with a sculpted tree... all from some Grease Pencil strokes! Can you think of other things you can make with this workflow, besides skeletons and trees? I'd love to see what you'll make!

Easily Create Textured Mountains

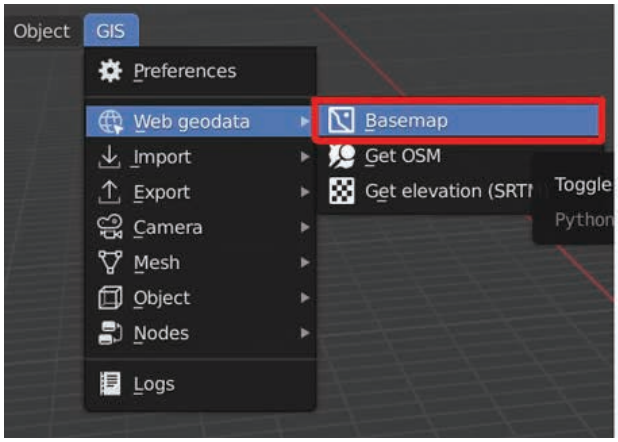
BlenderGIS is a free add-on that lets you import height map data and more. Using this add-on, I was able to quickly make the below render. Note that for the sky with clouds I used a simple photograph from Textures.com.

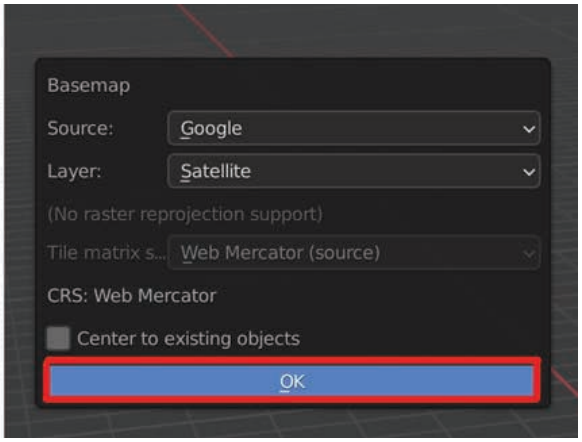
Download the zip file from Github (https://github.com/domlysz/BlenderGIS) and install it in Preferences. It might take a few seconds after you click on the checkbox.



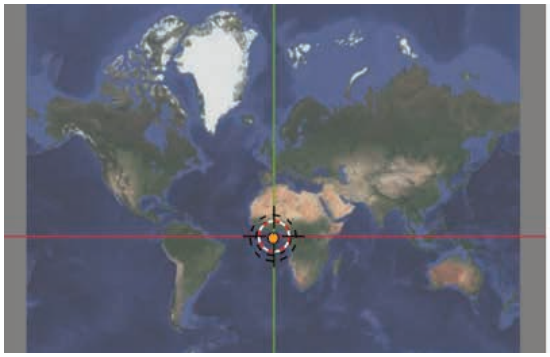


You'll notice that the GIS menu has been added to your 3D Viewport. Open the drop-down menu by **clicking on GIS**, then go to **Web geodata > Basemap**. Here you can choose the source of the terrain data, for now let's use the defaults. **Click OK**.

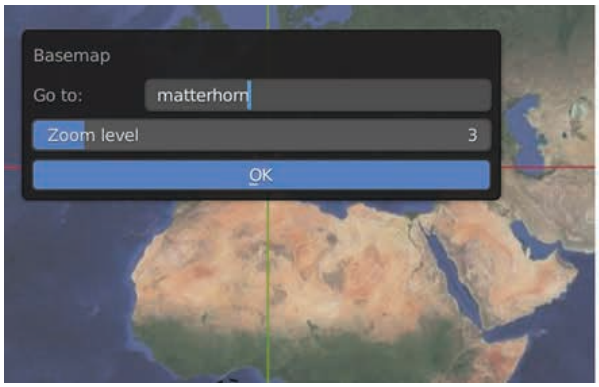




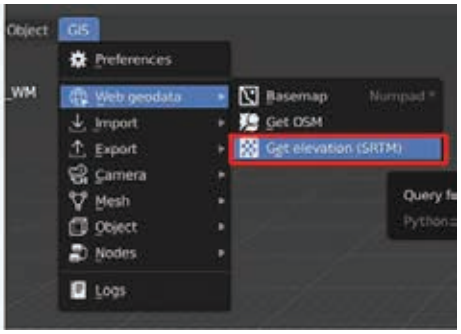
As you can see, you've now downloaded the entire earth into Blender. Your internet bill is going to be huge this month! Just kidding, this is just an overview where we select which part to download. Use the scroll wheel on your mouse or the **+/- buttons** on the numpad to zoom in or out on this map. It takes a few seconds to update the zoomed area to a more detailed version.



I'm a bit of a chocoholic, so the first area that comes to mind is the famous Swiss Matterhorn mountain, as depicted on the package and in the shape of Toblerone chocolates! After zooming in you can select a region by **pressing B**. However I don't know the precise location of this specific mountain, so I'll take advantage of the search function of BlenderGIS by **pressing G** and typing the name of the geographic feature we're looking for.



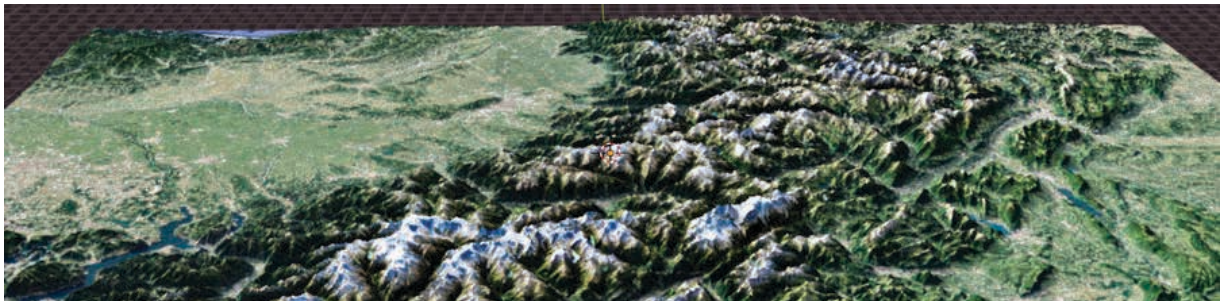
That zooms the map to the area you're looking for. **Pressing E** exports that selection to a plane with an image texture. However, it's still very flat! To get the height data, go to the GIS menu once more and now select **Web Geodata > Get SRTM**. Then **click OK** in the next pop-up window. Now Blender is downloading the height map for that area. This might take a minute or two.



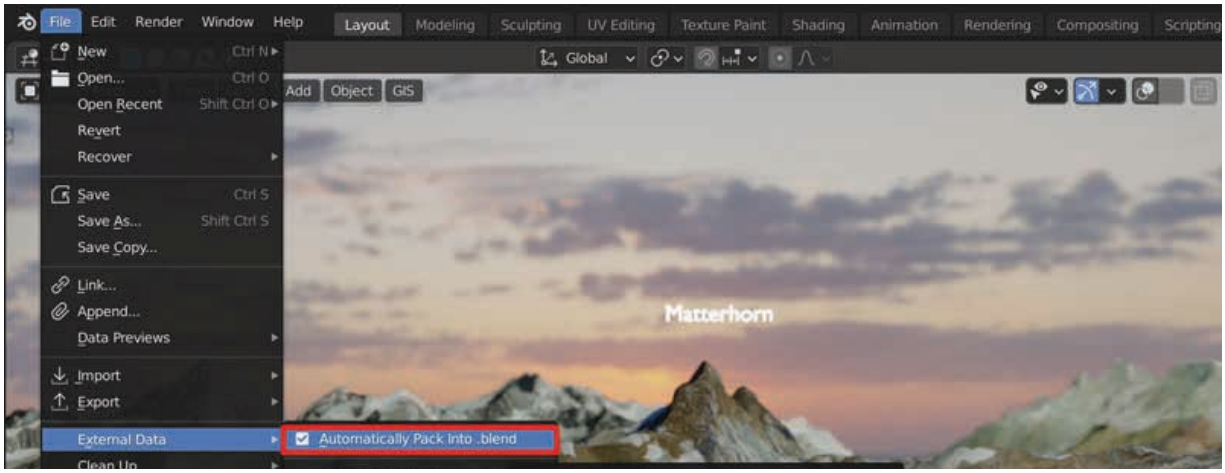
If you look closely, you'll see that there's some subtle height differences now on the plane. But you'll probably want them to be a bit stronger. When you open the Modifiers panel, you'll find a **Subdivision modifier** (to give the Plane enough geometry for displacement) as well as a **Displacement modifier**.

Strength of the displacement is set to 1 by default, but you can increase it so the mountains are higher. The value that works best will depend on your specific case, so just increase it until it looks good. For me, a value of 2.62 looked good.

I find that subdividing the plane in Edit Mode a few times also helps to give the terrain more detail. Here I'm using an HDRI to add some light.

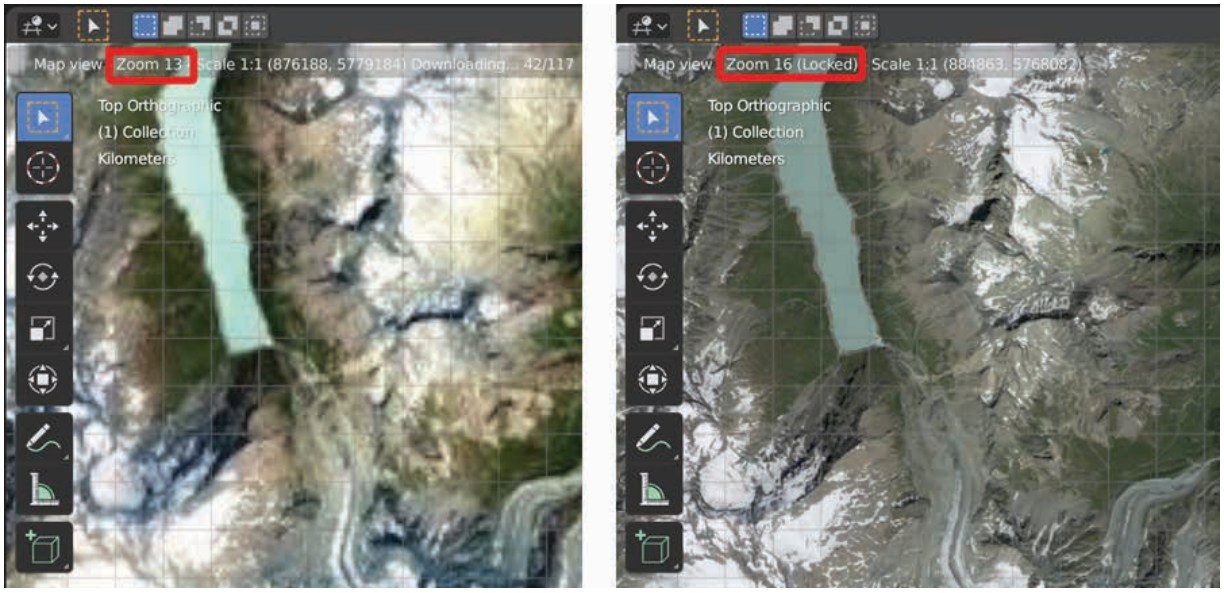


Before you close your Blend file, there's a final important step that you should do, and that's to check "**Automatically Pack into .Blend**". Otherwise, next time you open the file, your mountains will be flat...



Bonus tip: If the resolution of the downloaded region is a bit low, go back to the step where you select the region, **then press L** to "lock" that selected region. After this, you can still zoom in to a higher detail level by scrolling the mouse wheel up (or pressing + on the numpad).

Press E to export that region and then go to **Web Geodata > Get SRTM**. Below you can see the difference between zoom levels 13 and 16, for example.



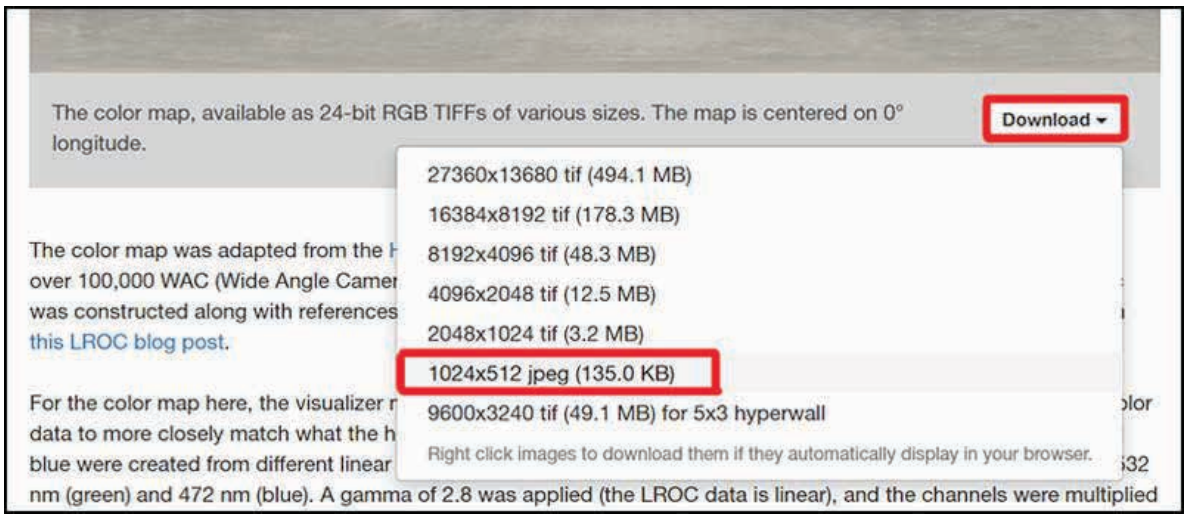
Do keep in mind that the higher the zoom level, the longer it will take to download the data. Also, not every zoom level has satellite photographs available, or they will be taken at different times so part of the mountain may have a slightly different color or sunlight angle.

Create the Earth's Moon

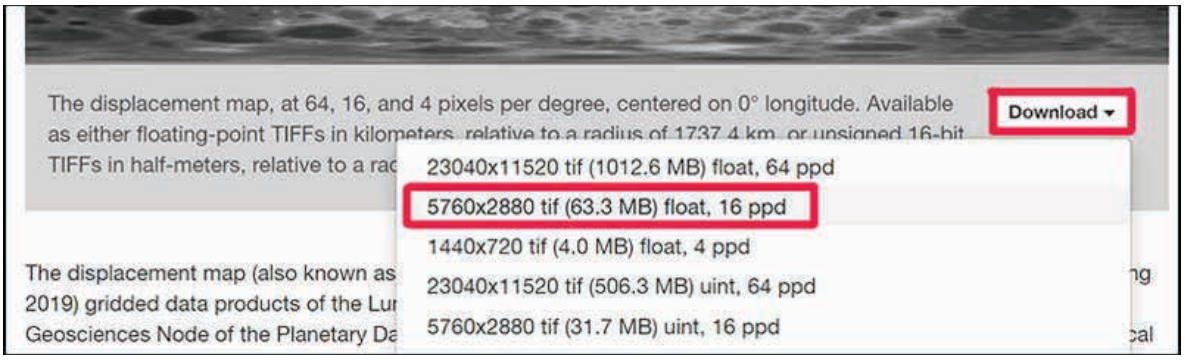
Making a 3D model of the Earth's Moon in Blender is easy thanks to the assets we can find for free on the NASA website. NASA's website is a treasure trove of textures and displacement maps for use in 3D software. They even have 3D models of their spacecraft like the lunar lander.

On this page you can find diffuse and displacement textures for recreating the Earth's Moon in Blender.

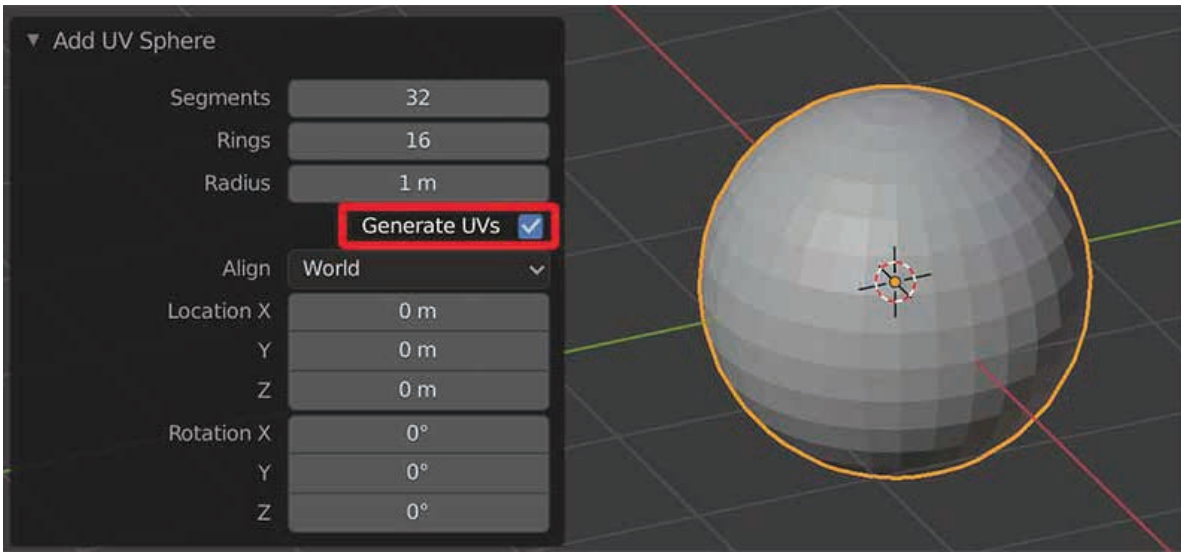
Download the **Diffuse texture** (called "**Color**" on the NASA website). In this example, I used the small JPEG texture for the color, as the resolution is not as important as for the diffuse texture as it is for the displacement. When clicking on the filename, it will load that image in your browser window so you still have to right-click on it and choose "Save Image As...". Rename it something like "**Moon_Diffuse**" so it's clear what it is.



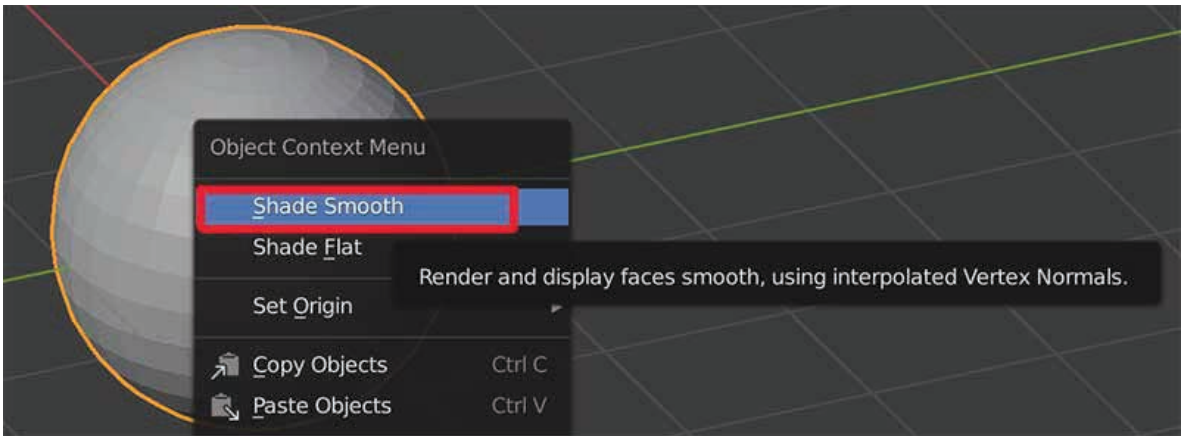
Do the same for the **Displacement map**, which you can find if you scroll down on that same page. Rename it "**Moon_Displacement**" for clarity after downloading it. With displacement maps in general the better the quality (size and bit depth) the better the displacement result will be.



In Blender, delete the Default Cube (leave the Camera and Lamp) and **press Shift+A** and choose **Mesh > UV Sphere**. Since Blender 2.8 the checkbox "**GenerateUV's**" is checked by default. If you are still using Blender 2.79, you need to check this box. This is important so you get the UV maps that you need to add the textures.

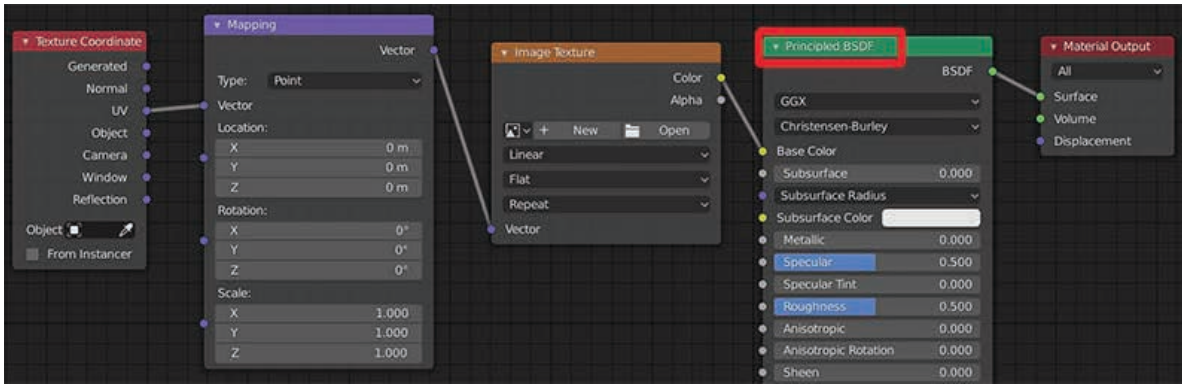


Right-click on the Sphere and choose **Shade Smooth**. Give yourself a pat on the back, as you've just finished modeling a whole moon! Set Viewport Shading to **Rendered** for the next part.

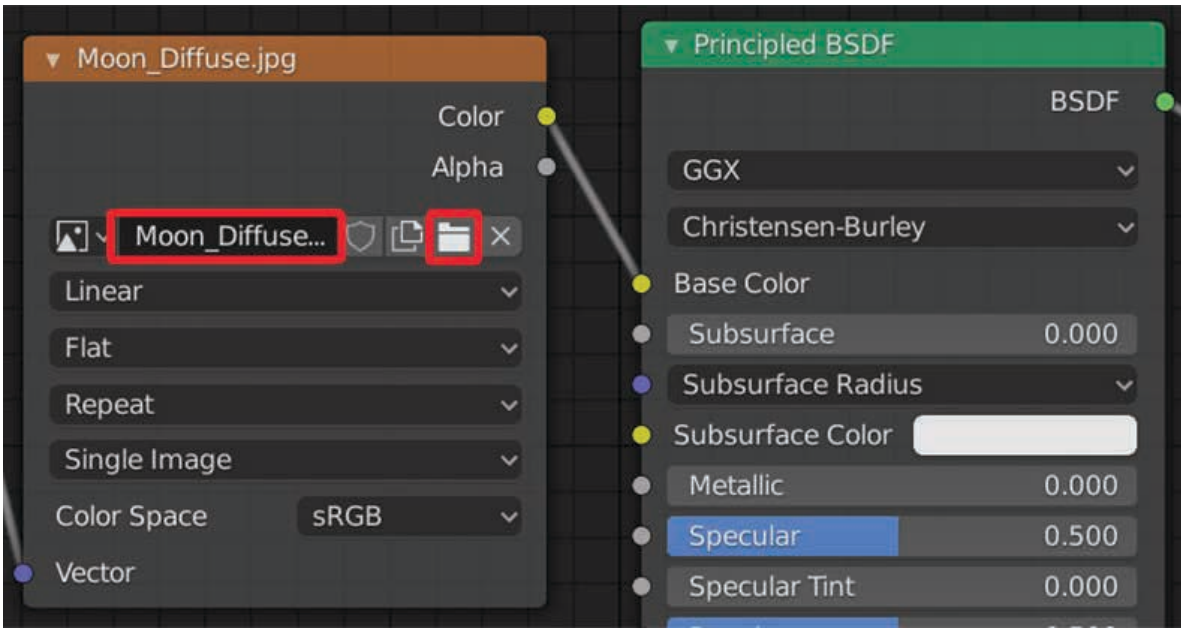


Now open a **Shader Editor window** and make it a bit bigger. **Click on New** to create a material for the Sphere.

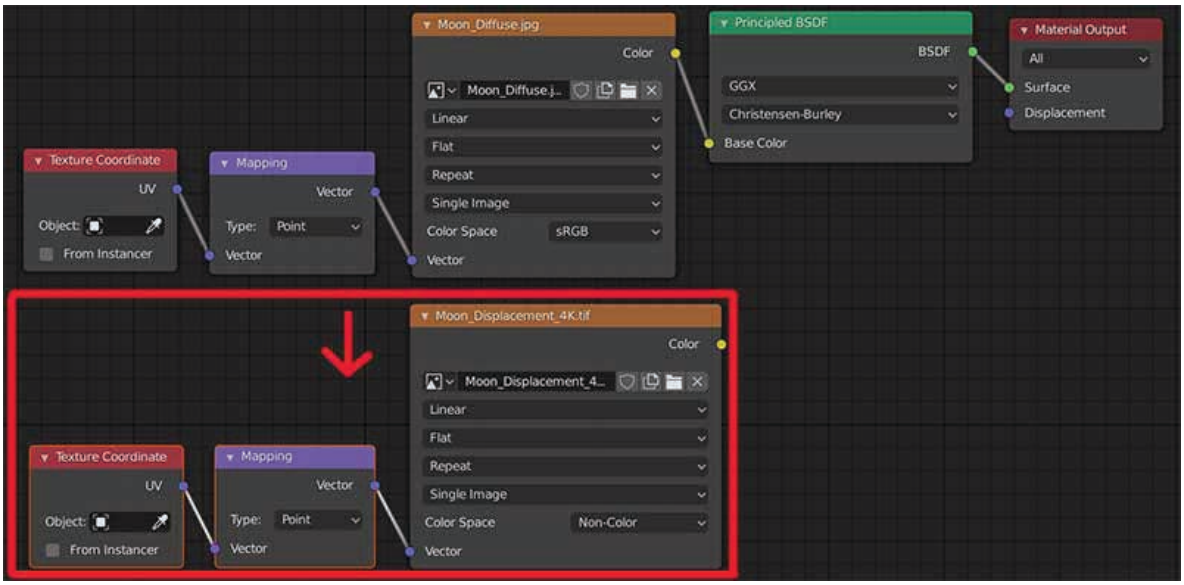
Click on the **Principled BDSF node** to select only that node, then **press Ctrl+T** to add all the image texture nodes (to be able to use this keyboard shortcut you have to have the **Node Wrangler add-on** enabled in Preferences, which comes with Blender by default).



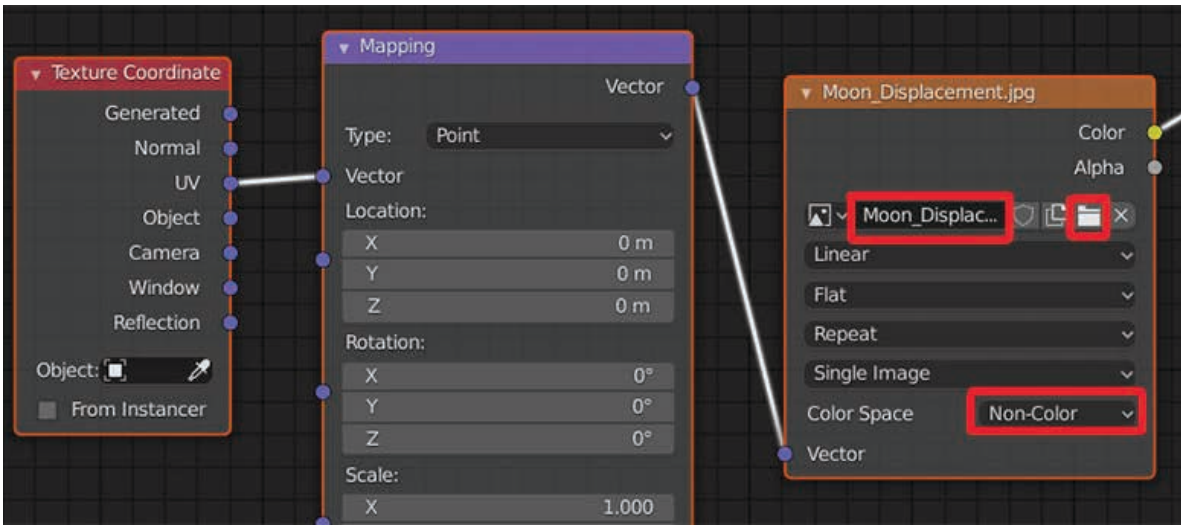
Add the "**Moon_Diffuse**" texture to the **Image texture node**. You'll notice that the Sphere already looks more moon-like! We just need to add some displacement now.



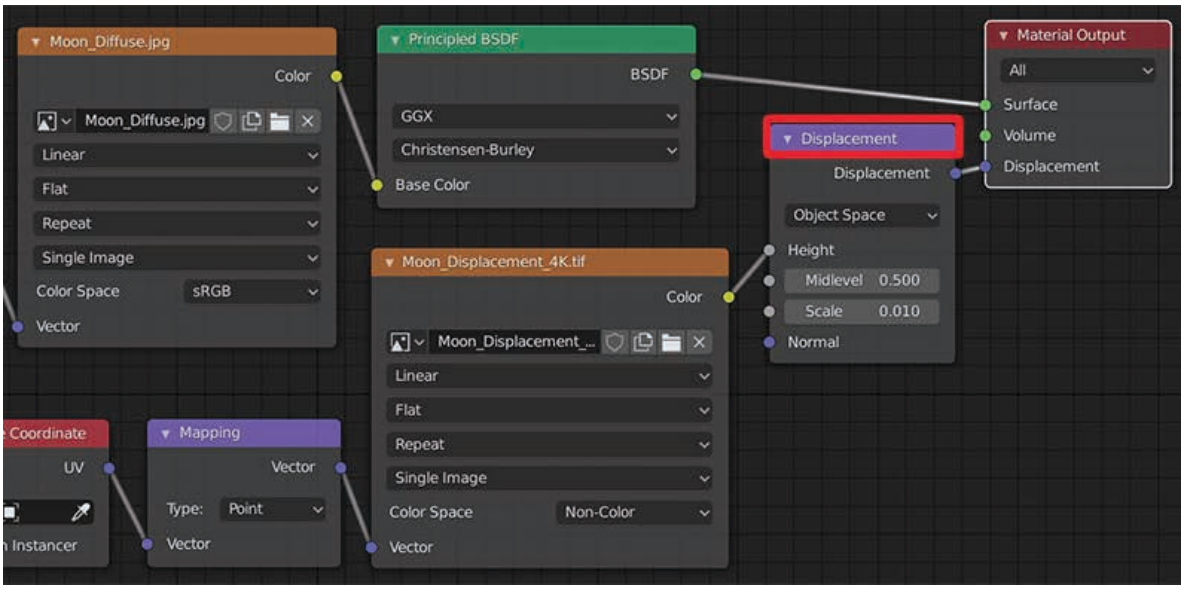
Now, select the three image nodes (Texture Coordinate, Mapping and Image texture) and **press Shift+D** to duplicate them, then move them down.



Change the image texture in the duplicated node to the "**Moon_Displacement**" texture. Change the Color Space to "**Non-Color**".

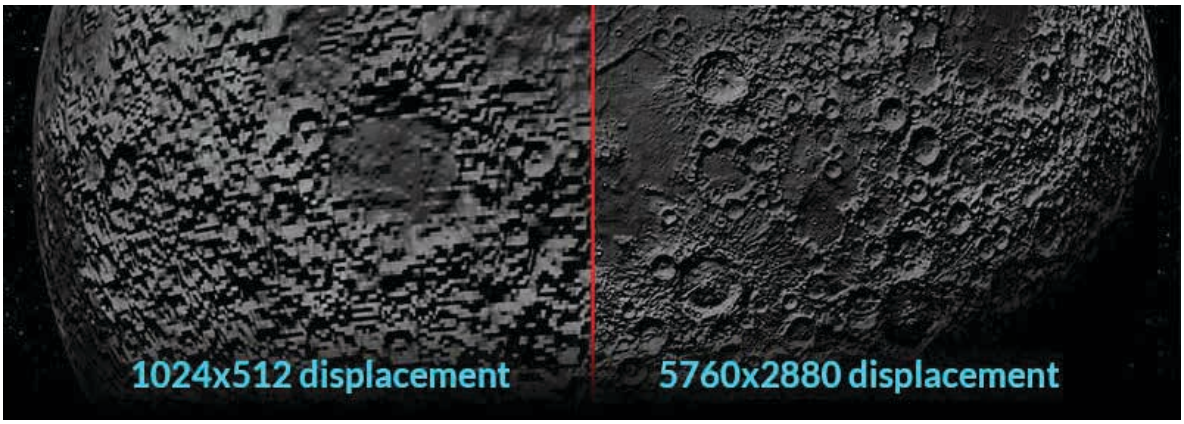


Add a **Displacement node** by **pressing Shift+A** and in the "Search" field searching for "displacement". Connect it as shown below. If it bothers you that the Principled BDSF is in the way, you can minimize it by selecting it and **pressing Ctrl+H**. This hides/unhides all the unused sockets. In fact you can do this for all nodes to tidy things up.

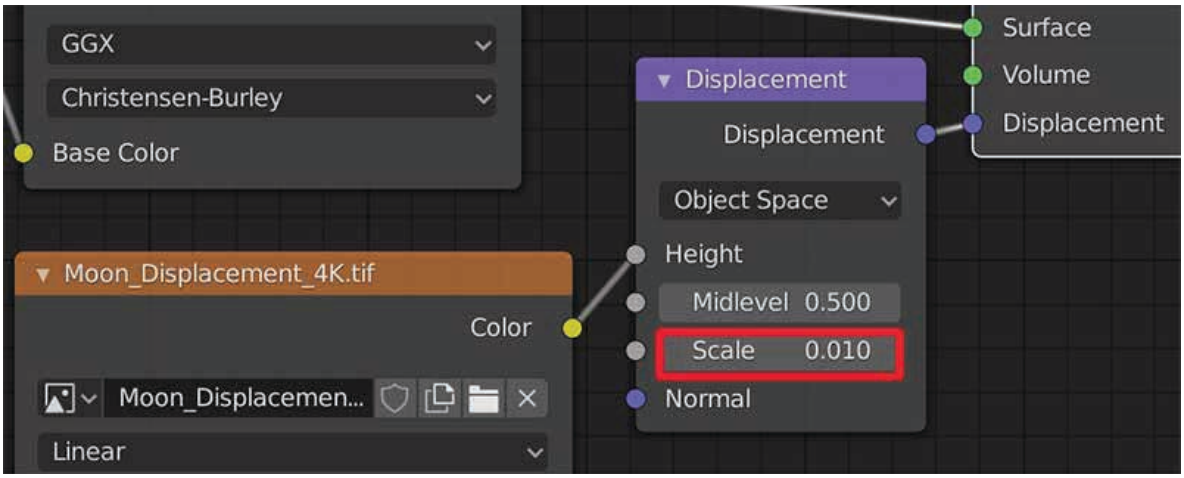


Below it's very clear to see what a difference quality makes with displacement maps. You can see that the smallest 1K JPEG displacement texture is pretty much unusable, with the displacement being very blocky. This is only good if you are showing the moon from very far.

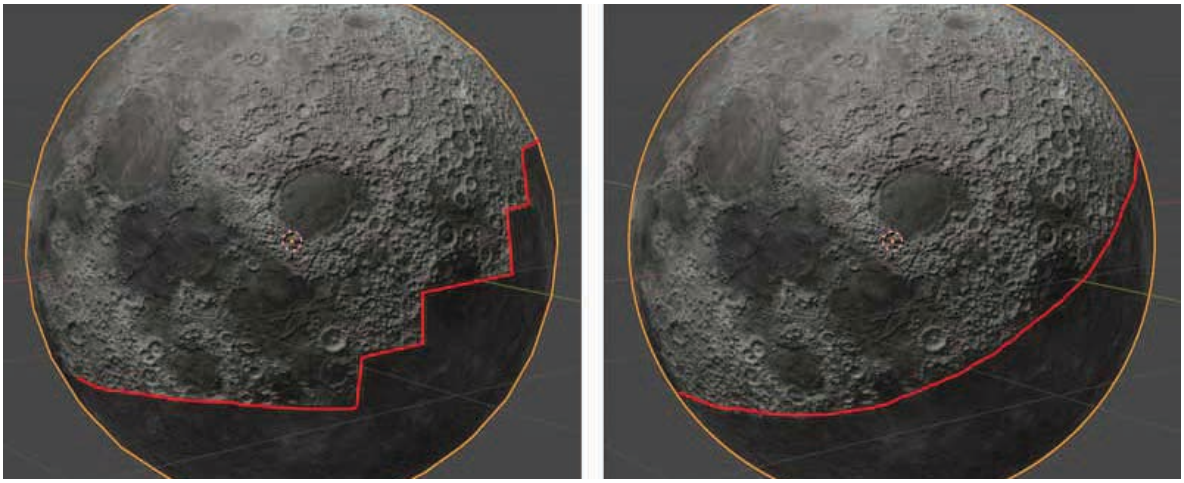
In this case, I will use the second-largest file, a 63.3 MB TIFF file, for the displacement. You can see how much better it looks with this higher quality displacement map.



You will need to lower the **Scale** of the Displacement node to about **0.01** for a more subtle effect.

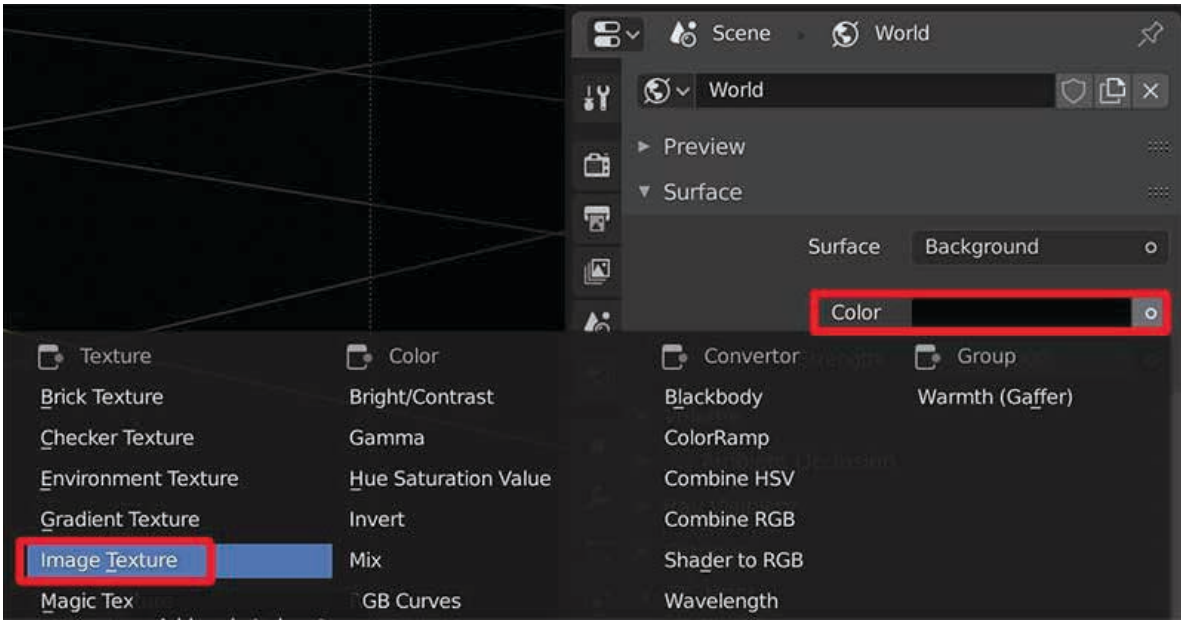


In Cycles you may get what looks like stepping artifacts where the shadow of the moon starts. This can be avoided by adding more geometry. To do this, add a **Subdivision modifier** by **pressing Ctrl+3** (that sets it to 3 subdivisions). This should get rid of the ugly shadow.

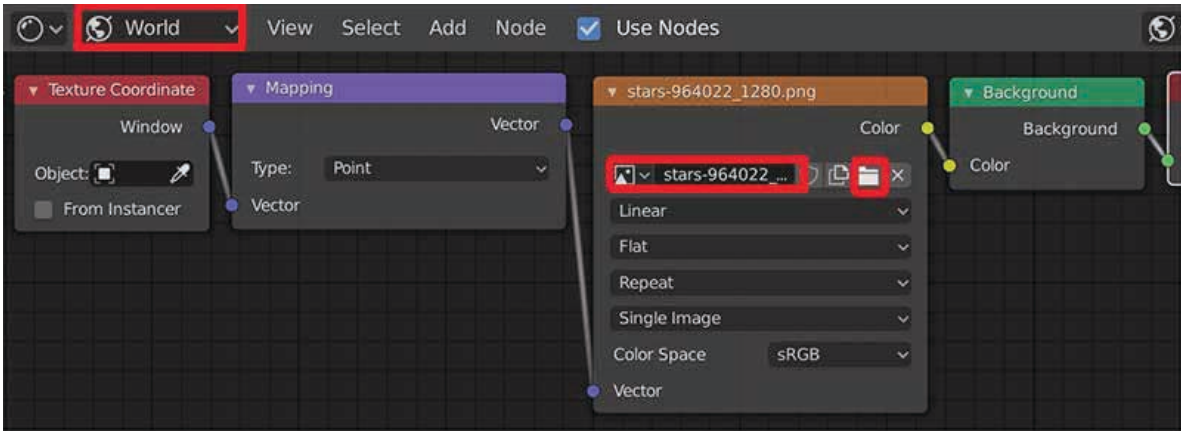


Now that our moon looks good, let's add some stars in the background. You can search the NASA website again using their image search, but I found that it's not easy to find a simple image of stars there, since there are just so many images to wade through. It's easier to use a stock image website like Pixabay where I found this CC0 licenced image.

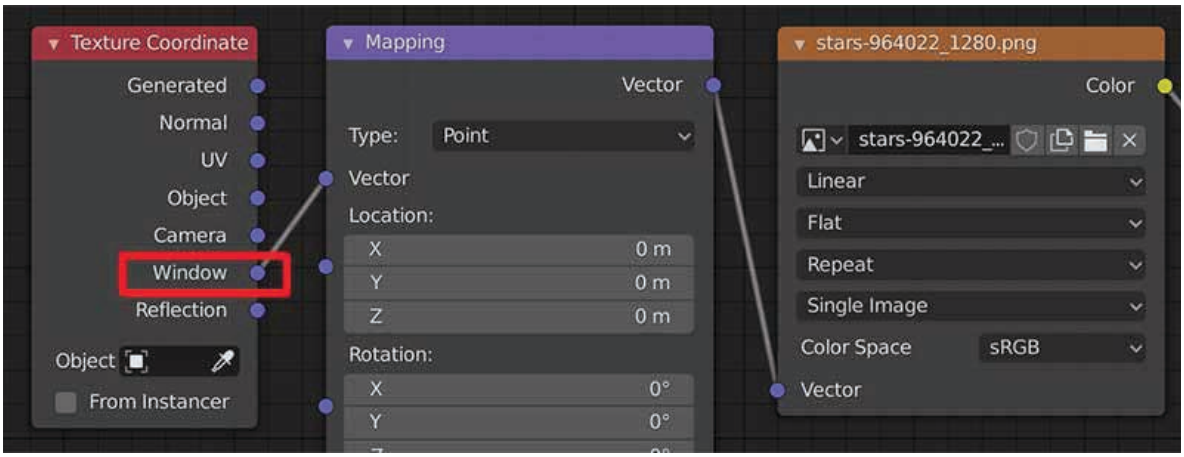
Go to the **World tab** and as the **Color** of the **Background** add an **ImageTexture**. Then choose the picture of stars you downloaded.



In the **Shader Editor window** set it to **World** instead of Object so you can edit the nodes of the environment. Just like with the moon shader, click on the image texture node and **press Ctrl+T** to add the Texture coordinates and Mapping nodes (note that for the sake of the screenshot here, I've minimized the nodes by **pressing Ctrl+H**).

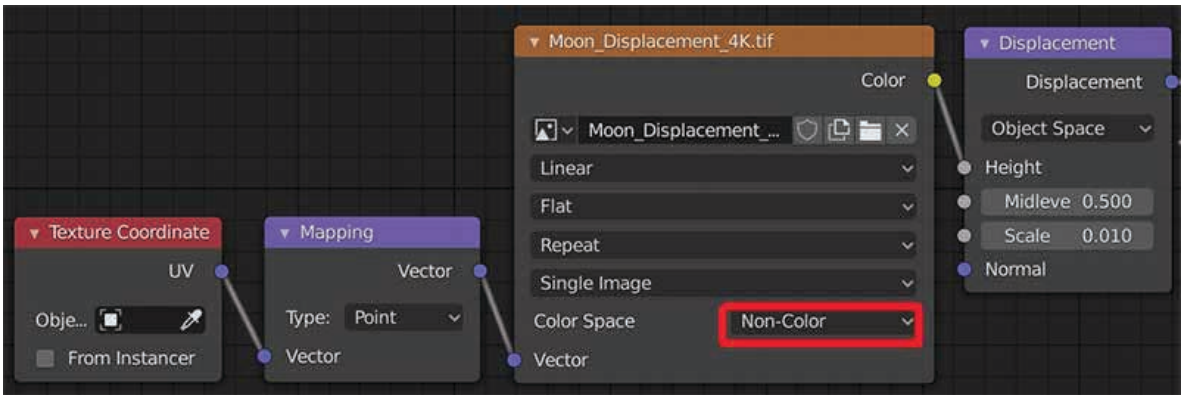


In the **Texture Coordinates node** use the **Window** output. This ensures that the stars image is always shown in the background no matter how we move the camera.

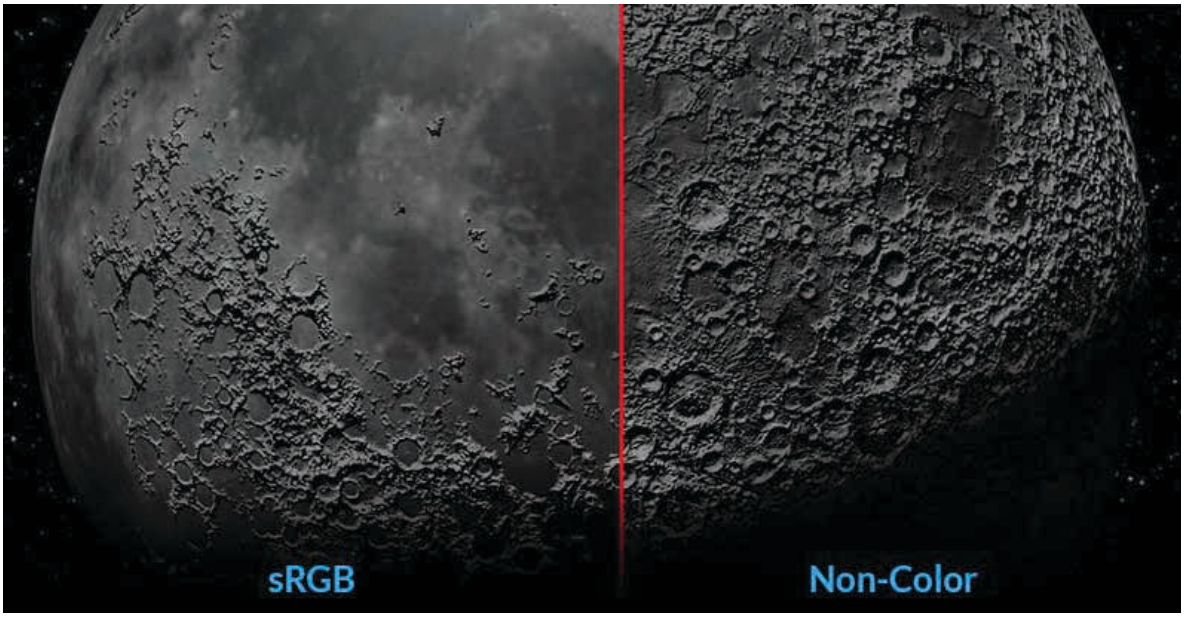


If you want, you can add an extra light with a lower **Power** value on the opposite side of the moon from the original light so the shadow part of the moon is not completely dark. Of course this is not super realistic (as we only have one sun) but some artistic license makes it more interesting to look at.

There you go, that's a simple way to make the Earth's Moon. It pretty much relies on having a good Displacement map. Make sure the Displacement Image Texture is set to Non-Color and not to sRGB.



You can see the difference it makes:



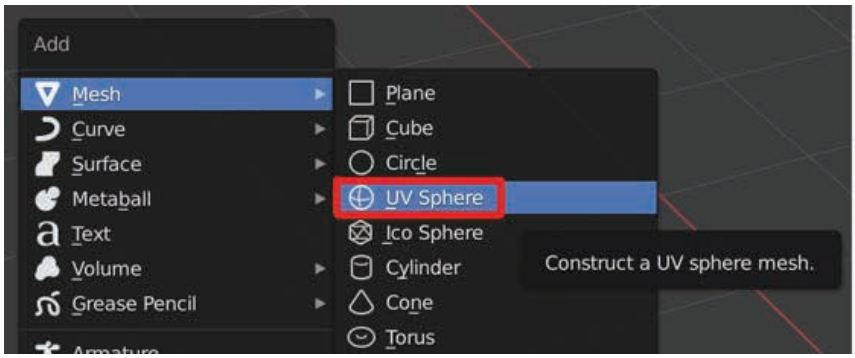
Creating Clouds using Mesh to Volume

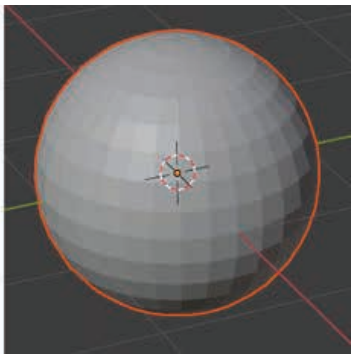
Note that you need at least Blender 2.91 (the stable version of Blender at the time this was written) for this tutorial. This feature was not yet available in Blender 2.8.

The **Mesh to Volume modifier** lets you easily turn any mesh into a cloud. Before we do that, let's look at an easy way to create a mesh that already has an interesting cloud-like shape.

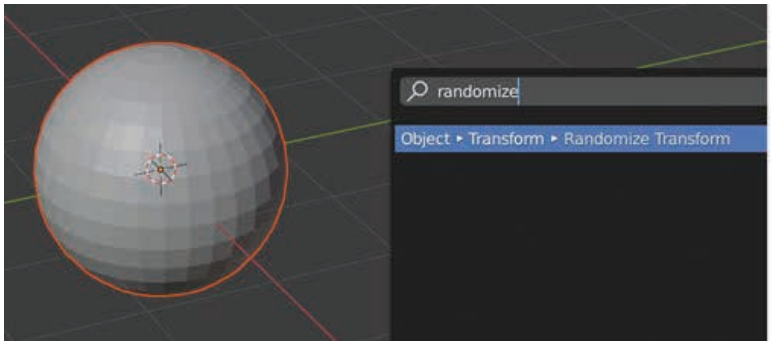
Select and delete everything in the default scene, then add a **UV Sphere**.

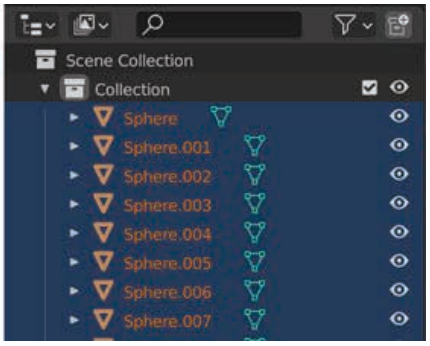
In Object Mode, with the UV Sphere selected, **press Shift+D and Enter** to duplicate it, then to repeat that **press Shift+R** a few times so you've got a bunch of duplicates of that Sphere - 25 should do. Of course they'll all be in the same spot, so it looks like you've only got one.



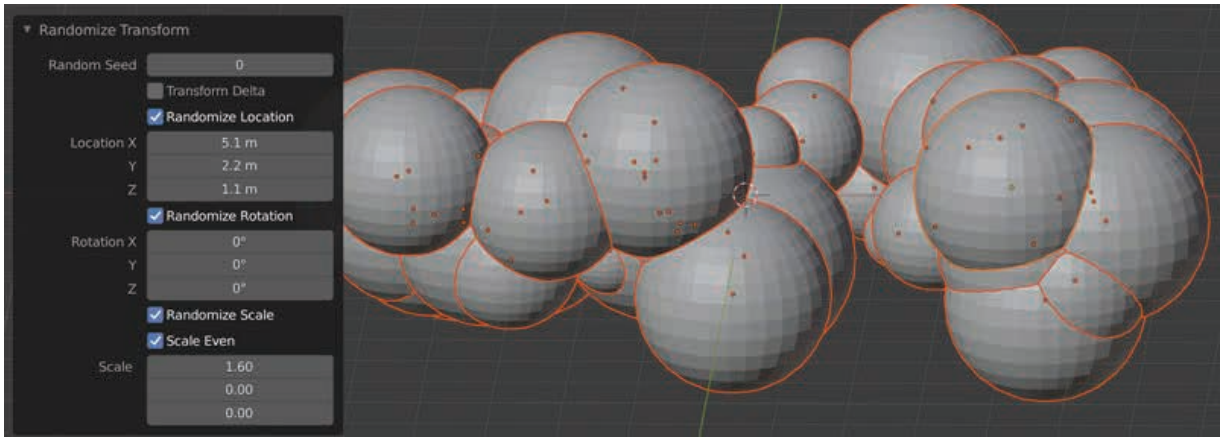


Select them all in the **Outliner** and **press F3**, then search for "**Randomize Transform**". This opens a little menu in the bottom left corner of the viewport. Expand the menu by clicking on the little triangle, so you can see all the options.



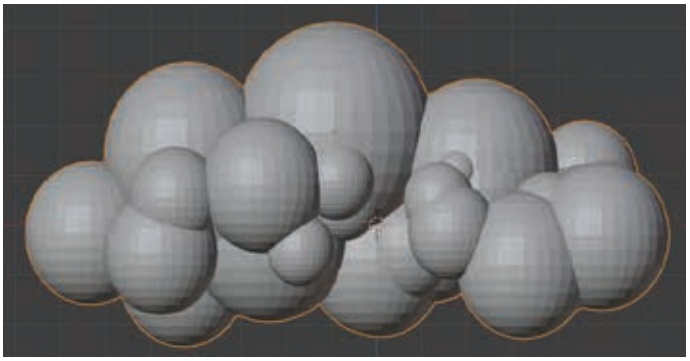


Increase the **Position** and **Scale** values - you'll notice the Spheres moving around to new random locations and to random sizes. You can check some different iterations by clicking on the **Random Seed** value, until you land on something you think looks like a good starting point for a cloud.

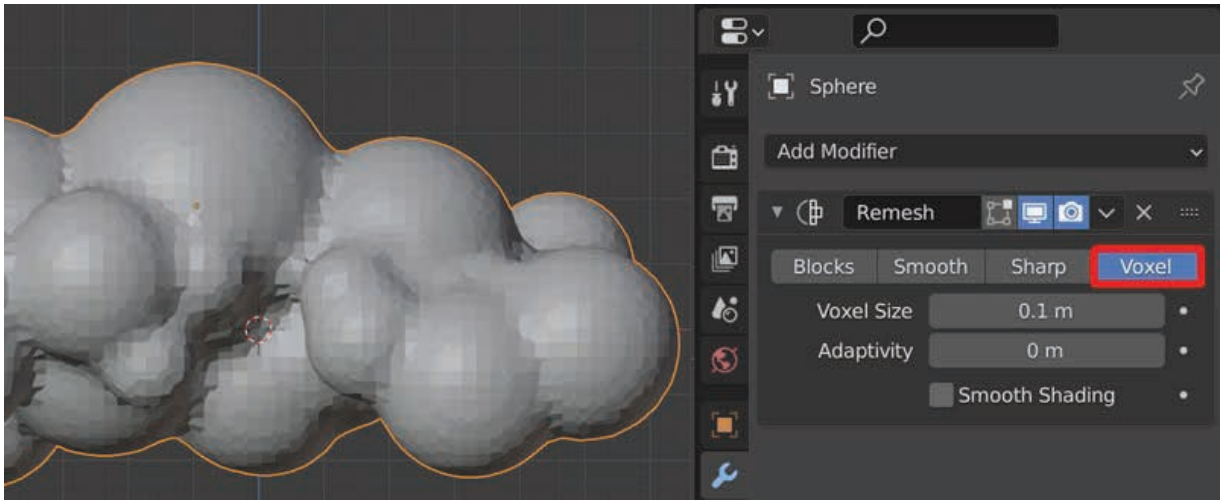


At this point you can also add a couple of additional UV Spheres manually to really make it perfect.

With all these spheres selected, **press Ctrl+J** to join them all into one object.

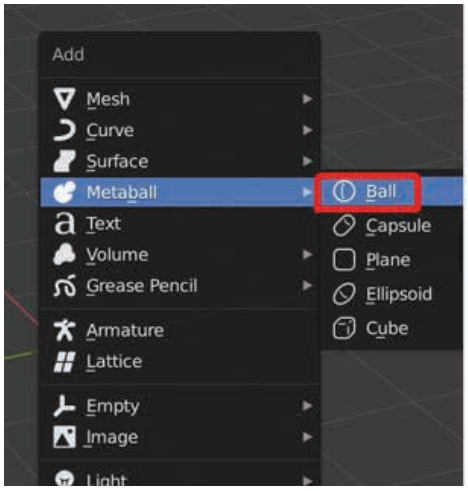


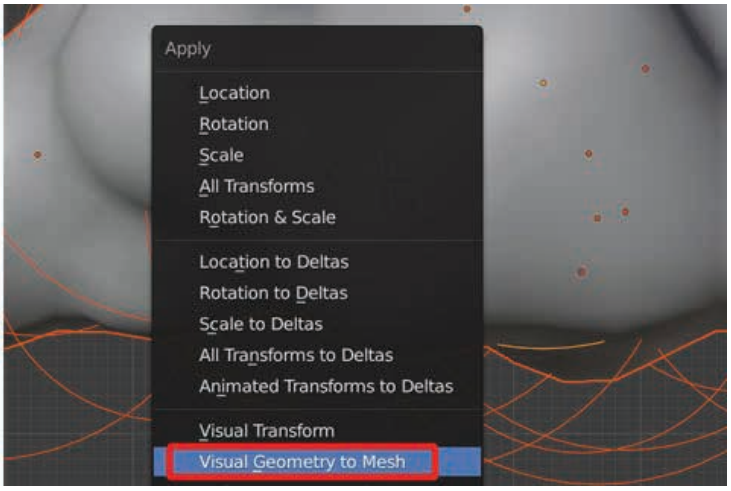
Add a **Remesh modifier** to it, set to **Voxel**. The default values are fine. Apply the modifier by hovering the mouse cursor over it and **pressing Ctrl+A**.



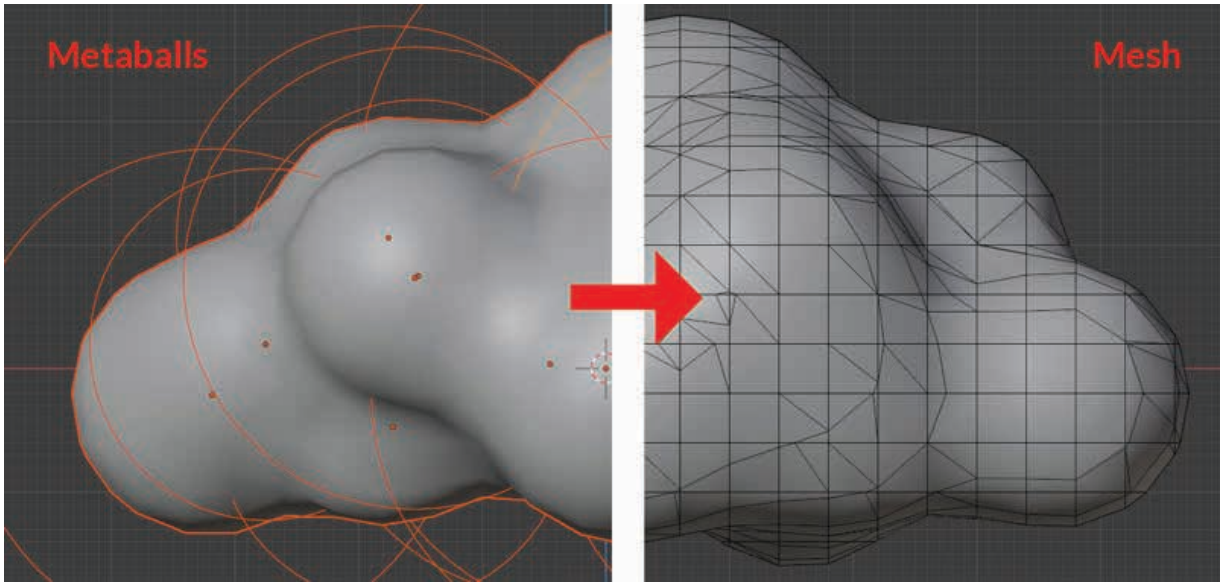
The above method is useful for creating several different clouds as you can just use the **Random Seed** value to quickly create variations. However if you only need one cloud, a fast method is using **Metaballs** and just placing them by hand.

You can add a Metaball by **pressing Shift+A** and choosing **Metaball > Ball**, then duplicate it with **Shift+D** and move the duplicate by **pressing G**. Once you have enough Metaballs, **press Ctrl+A** and choose "**Visual Geometry to Mesh**".

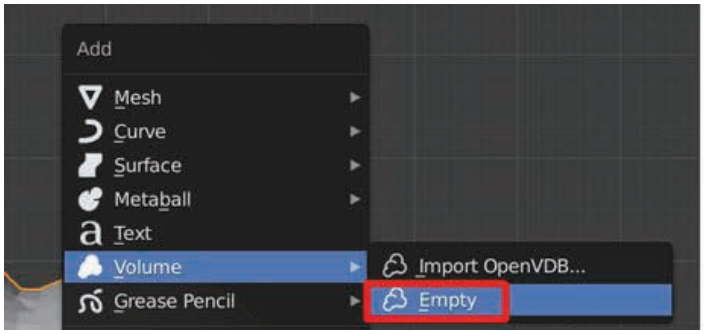


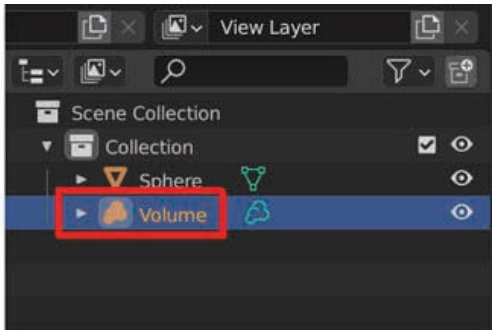


This turns the Metaballs into a normal mesh that you can use with the Mesh to Volume modifier.

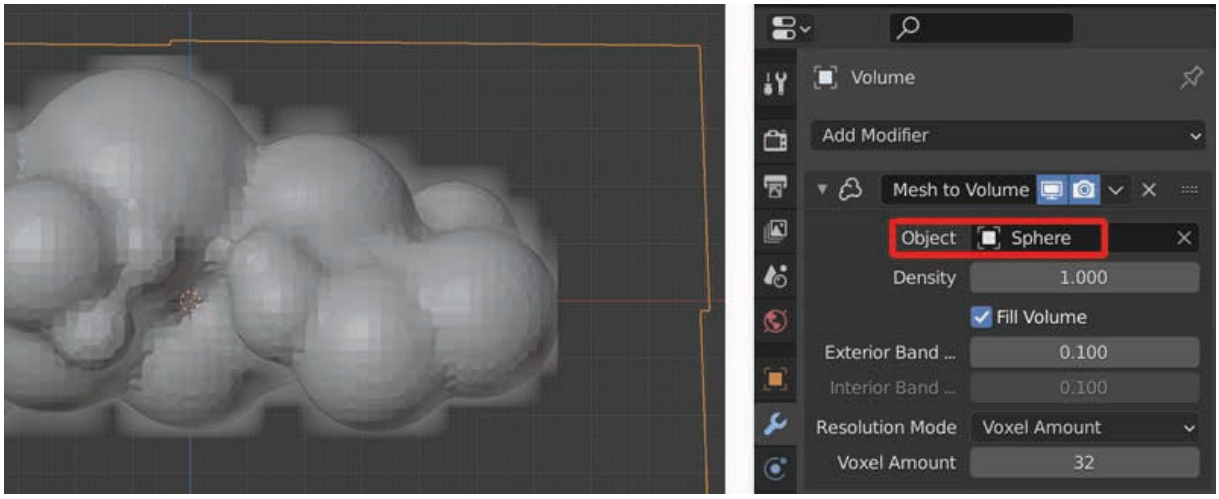


Once you've made your cloud mesh using one of the above methods, **press Shift+A** and add a **Volume > Empty**. You won't see anything new in the Viewport, but you can select the Volume in the Outliner.

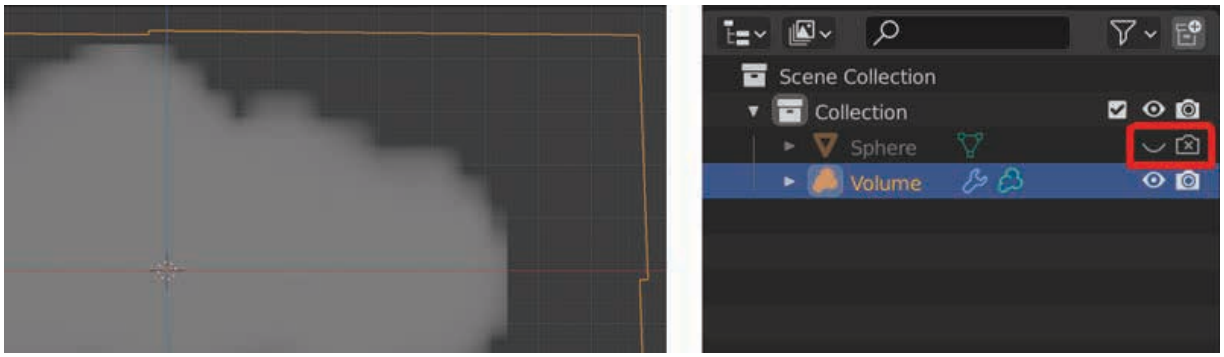




Add the **Mesh to Volume modifier** to it and pick the combined Spheres mesh as the **Object**. Now you can see some cloud-like stuff around the mesh.

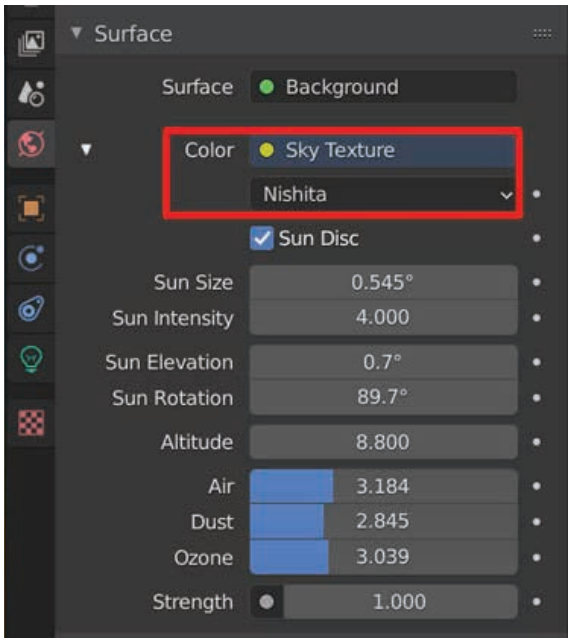


Turn off **viewport visibility** as well as **render visibility** for the mesh in the Outliner, so we can get a better look at the cloud.



Also, let's switch to **Cycles** and add some light before we continue. Now either add an HDRI or use the **Nishita Sky Texture** to add some interesting light. I like the Sky Texture because it has beautiful warm colors and gives me more control over the light.

You can rotate it, change the height of the sun and so on. Make sure you are in **Viewport Render mode**, so you can see the result. I tweaked the Sky Texture settings a bit to create a more dramatic look. You can see the settings I used below.

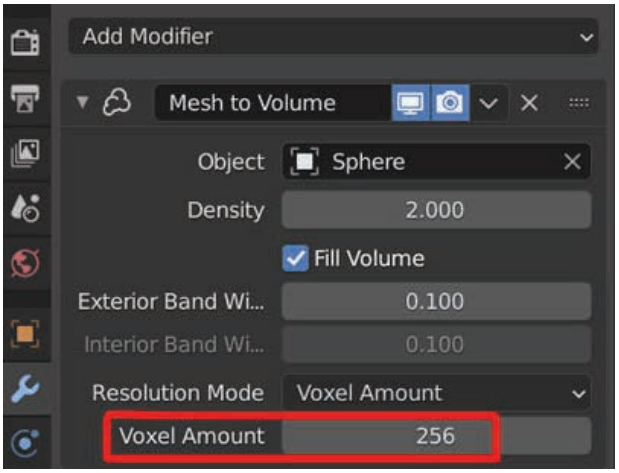


I also added a **Sun lamp** with a blueish tint, to create some contrasting light, as if this was the light bounced off from other clouds. This also makes the cloud less dark.

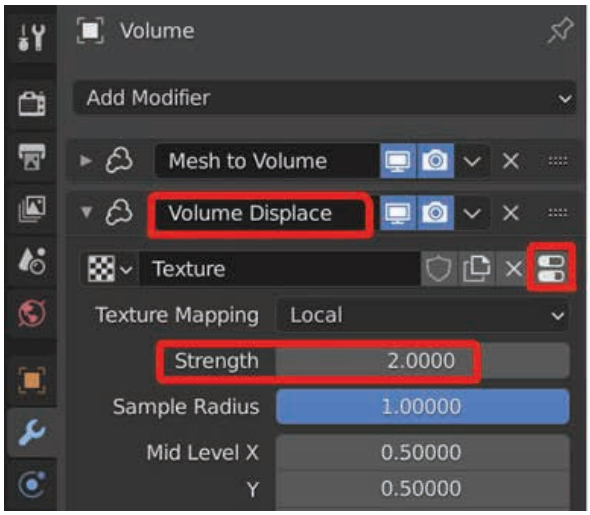
The **Voxel Amount** is the resolution of the cloud. A higher amount is better (and slower, so keep that in mind if you have an older computer).

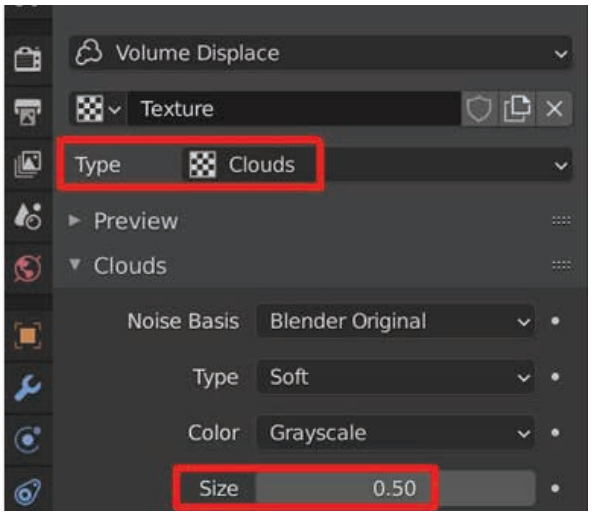
Once you increase the resolution to say 256 however, you'll notice that the cloud is a bit too perfect.

To fix this, add a Volume Displace modifier.

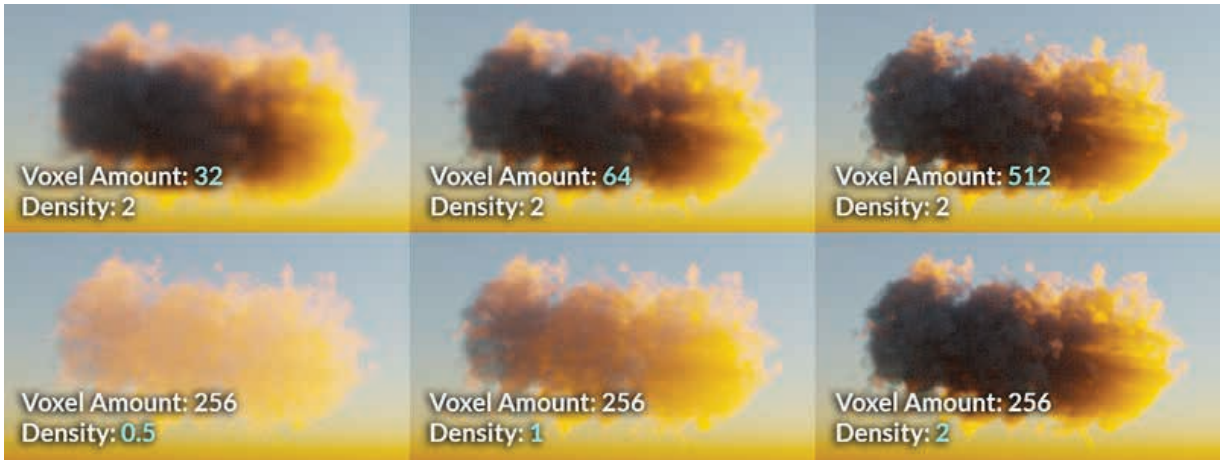


Click on New to add a new texture, and click on the icon next to that to go to the Texture tab. Choose texture Type: Clouds. This is similar to adding a Displacement modifier on a mesh to add some noise. I increased the Cloud Size to 0.5 and the Texture Strength to 2 to make the cloud a bit more "fluffy".

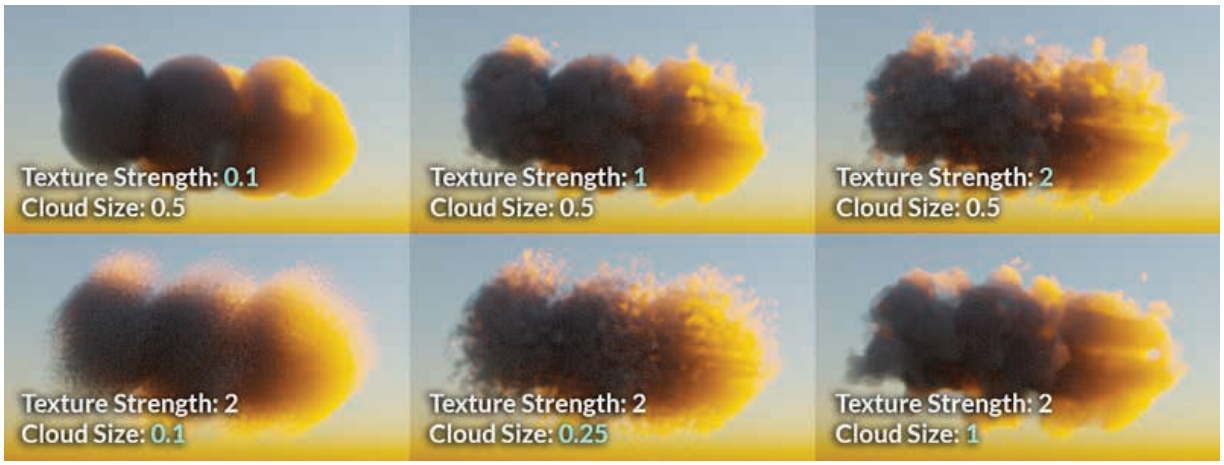




You can increase the **Density** on the Mesh to Volume modifier to get a denser cloud. Below you can see some example of what the Density value and Voxel Amount value do.



You can play with the **Size** of the Cloud texture and with the **Strength** of the Displace modifier to get a different result.



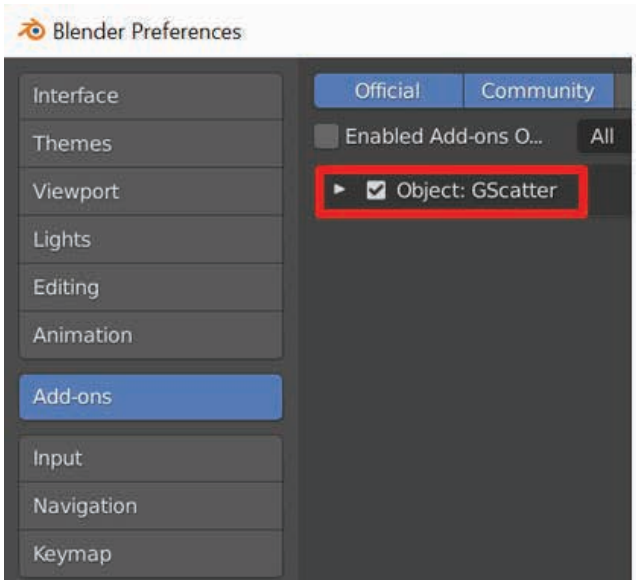
If you run into a Cuda-related error message when rendering this kind of thing on a GPU, try rendering with CPU instead and turning off Optix denoiser if you have it turned on. It will take longer, but at least it will render.

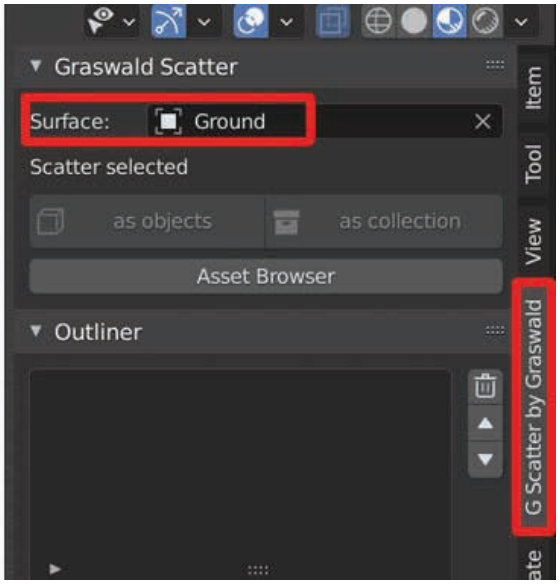
So there we go, we've created a nice, fluffy cloud!

G Scatter - Free Geometry Nodes-based scatter add-on

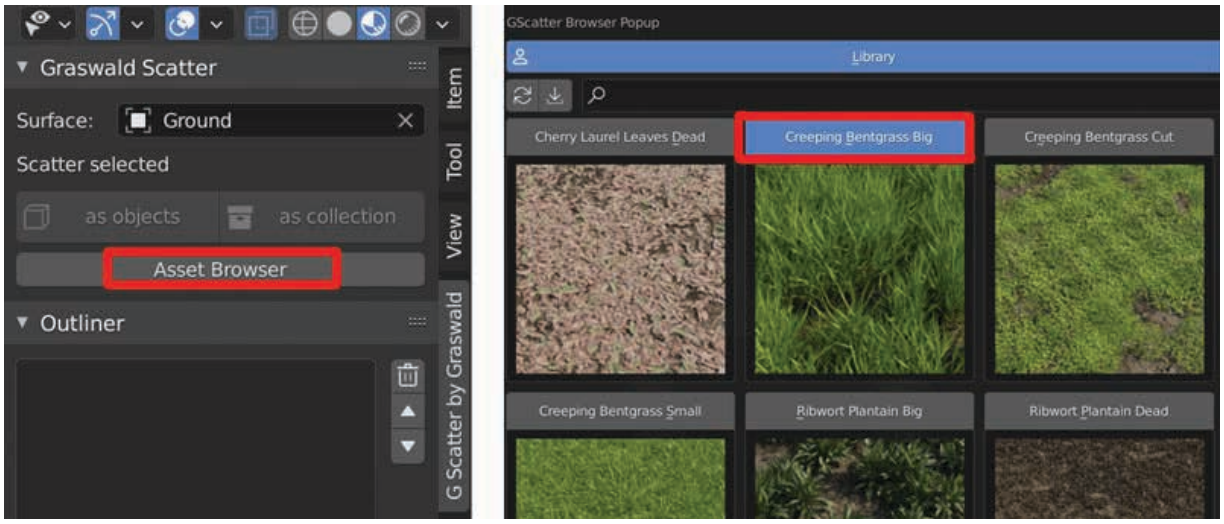
The creators of Graswald have created a free Geometry Nodes based scattering add-on for Blender 2.93. Even though it uses Geometry Nodes under the hood, you don't need to deal with any nodes or math, instead you can just use an artist friendly interface.

To be able to use it, download it from the Grasswald website. After downloading the Zip, you can install it in Preferences. Then you can find the add-on as a tab in the Option panel. First thing to do is pick the surface you want to scatter objects on.



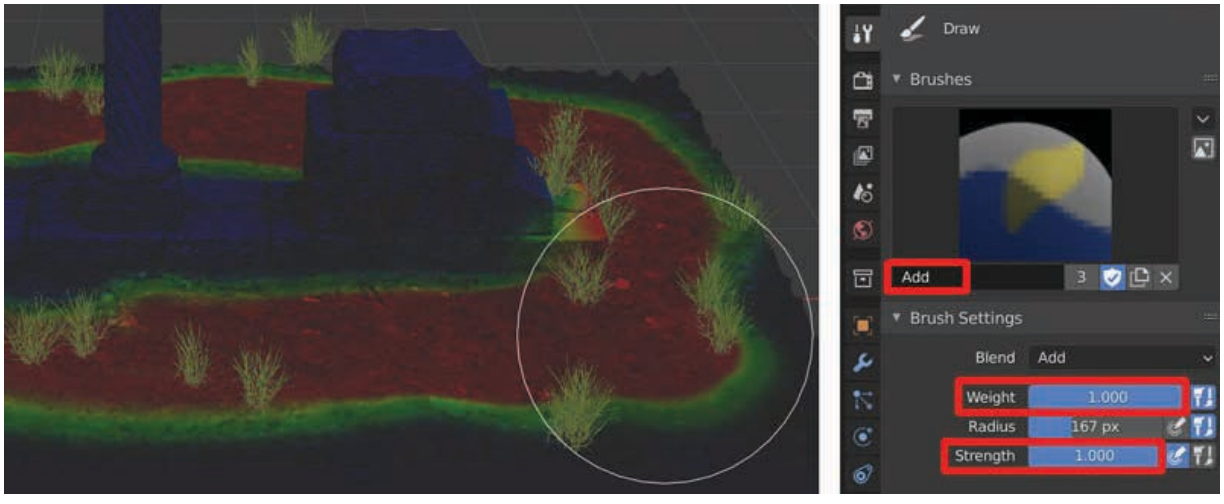


You can use the add-on to scatter any object, but it also comes with some free plants that you can use. **Click on the Asset Browser** to access them. Then, select a plant you'd like to use. **Click on Scatter Selected** and then **click on OK**.



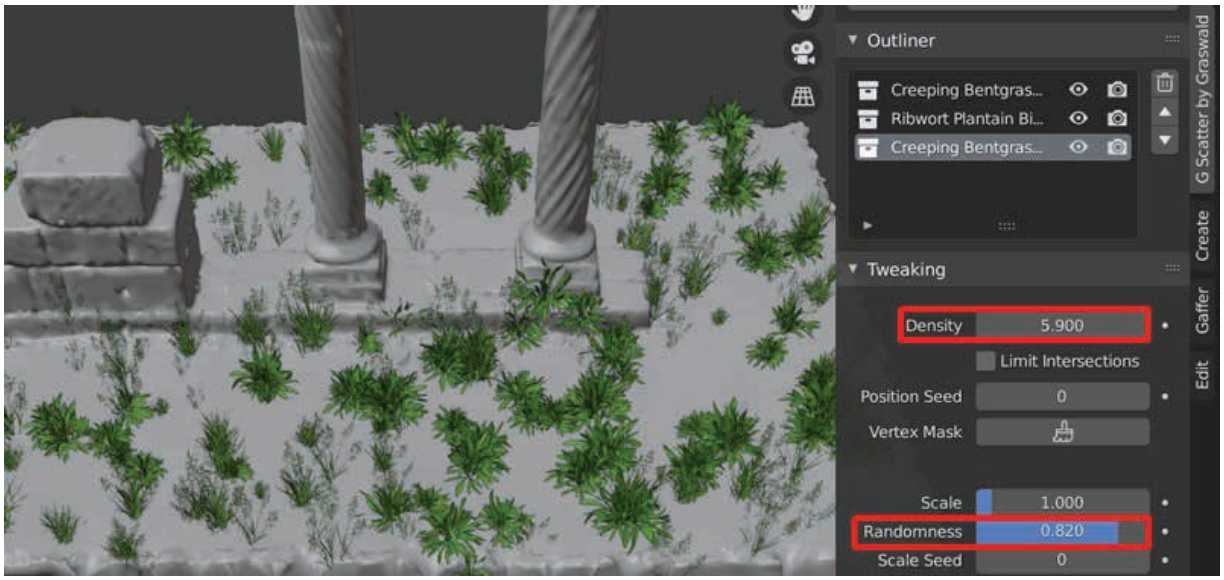
Now everything is scattered on the surface. However in this case I'm using a 3D scan which is all one single object, including the pillars. So the plants are scattered in places where I don't want them. Luckily the add-on has an easy way to paint where you want the plants. Under the Tweaking options, **click on Vertex Mask**.

Now we can simply paint where we want the plants to appear. If it doesn't work, make sure the Draw tool is in Add mode, and Weight and Strength are both set to 1.



Click here to return to the index.

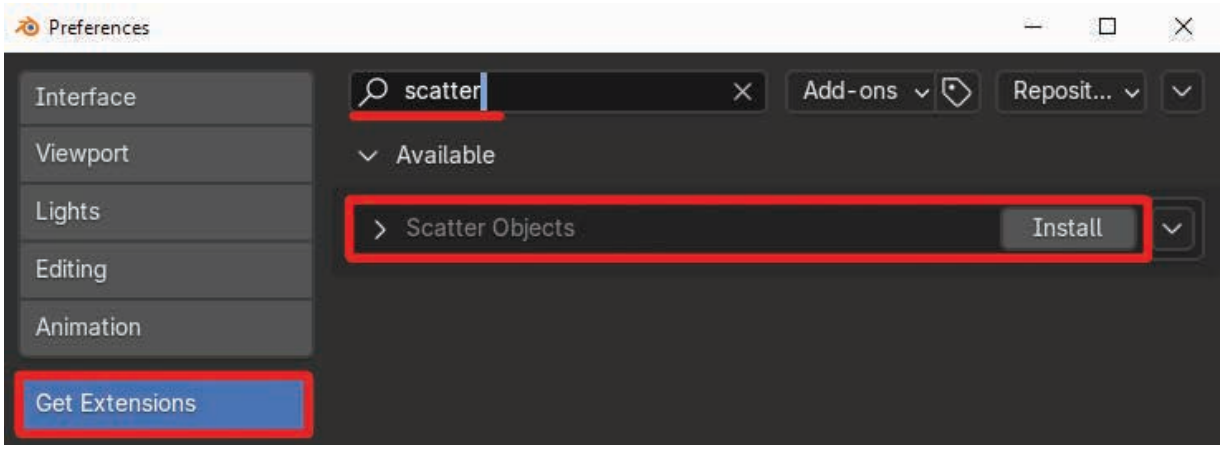
Switch back to **Object Mode** to exit the vertex mask painting process. After this you can increase the **Density** and **Scale Randomness**. And after that, just add more plants!



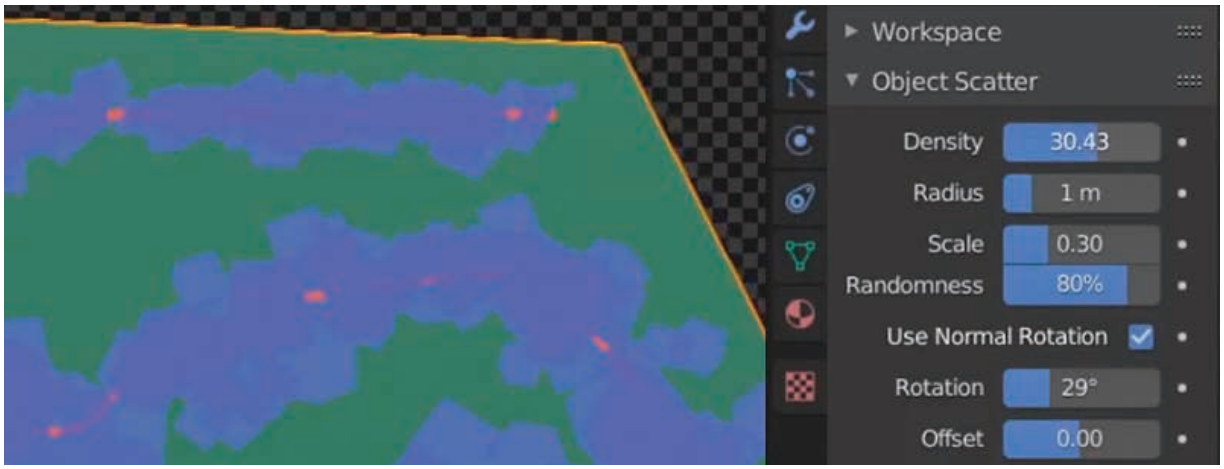
In case something doesn't look right, check these things: Make sure your surface has its Scale applied. If not, select it and **press Ctrl+A**, choose **Scale.** And make sure the Normals are facing the correct way (it should be blue when you display the **Face Orientation**). If not, select all in Edit Mode and **press Shift+N**.

Scatter Add-on

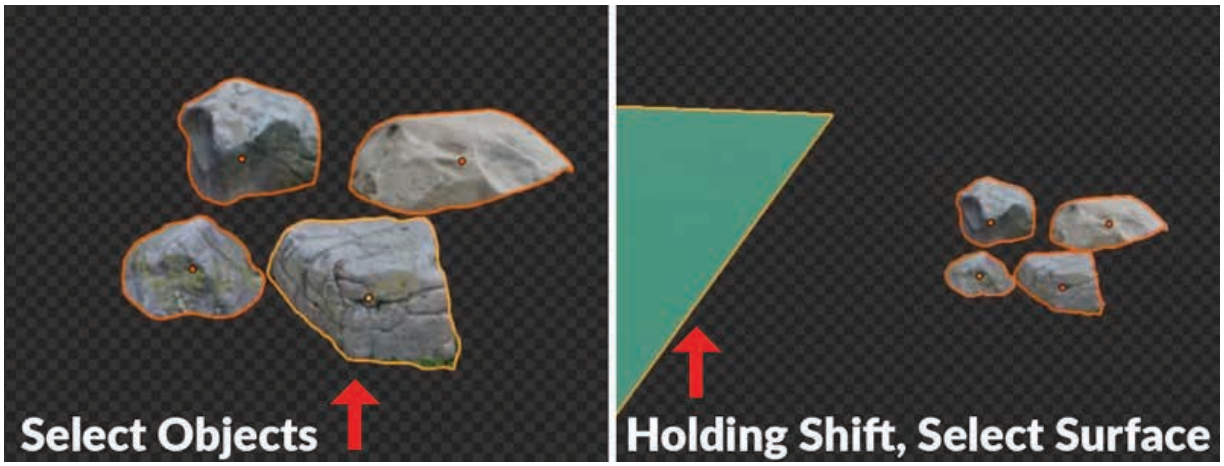
In case you don't want to download an add-on for scattering objects, the **Scatter Objects add-on** is a default add-on in Blender 2.8 that you only need to activate in Preferences to use (in the **Get Extensions** tab). It doesn't use Geometry Nodes like G Scatter does, but it works pretty well all the same!



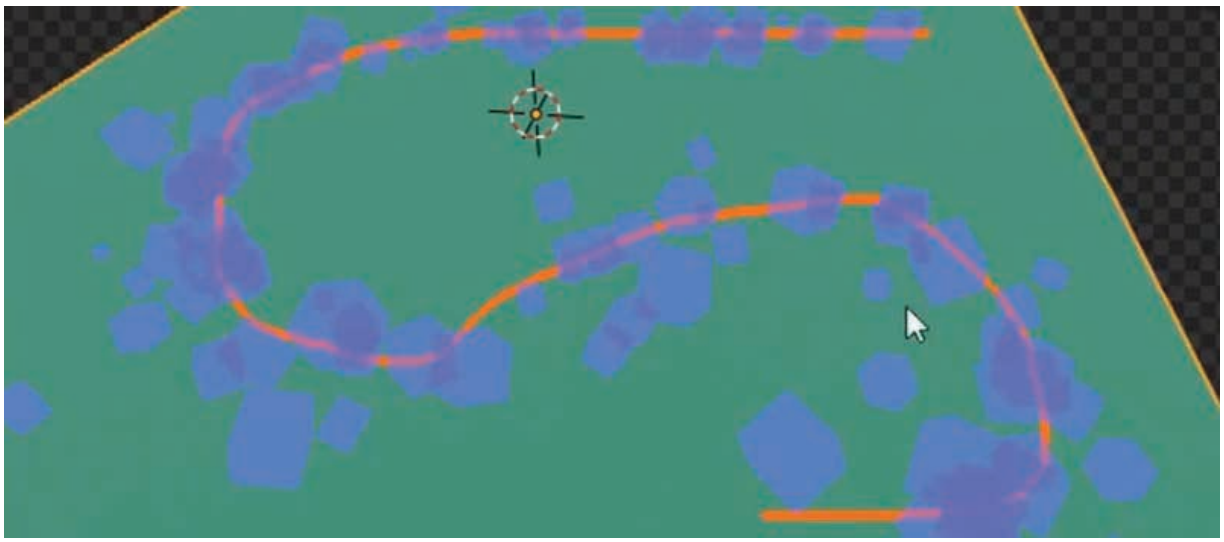
It allows us to quickly scatter things in a more intuitive and easy way than using the particle system.



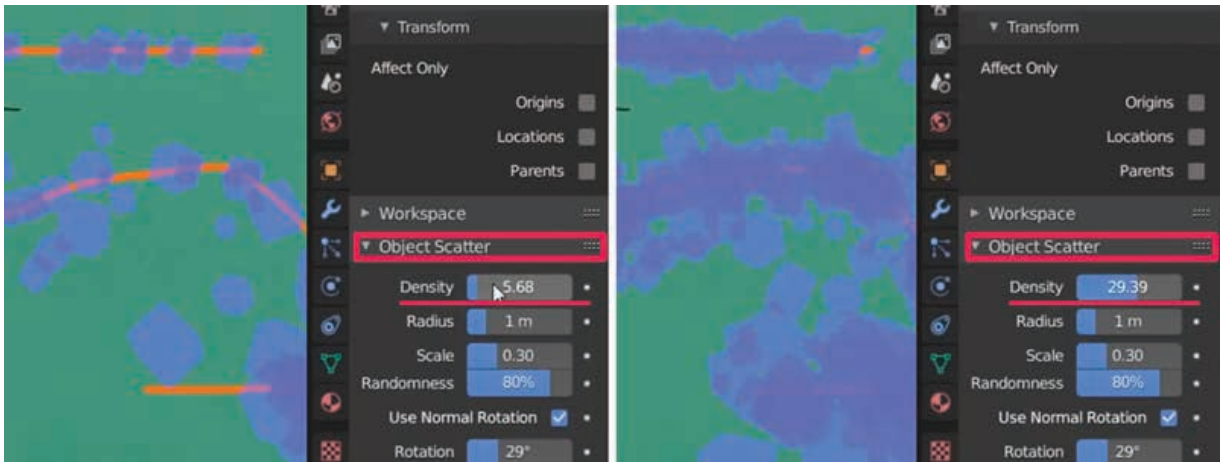
Select the object you want to scatter. Then **holding Shift**, select the surface you want to scatter them on.



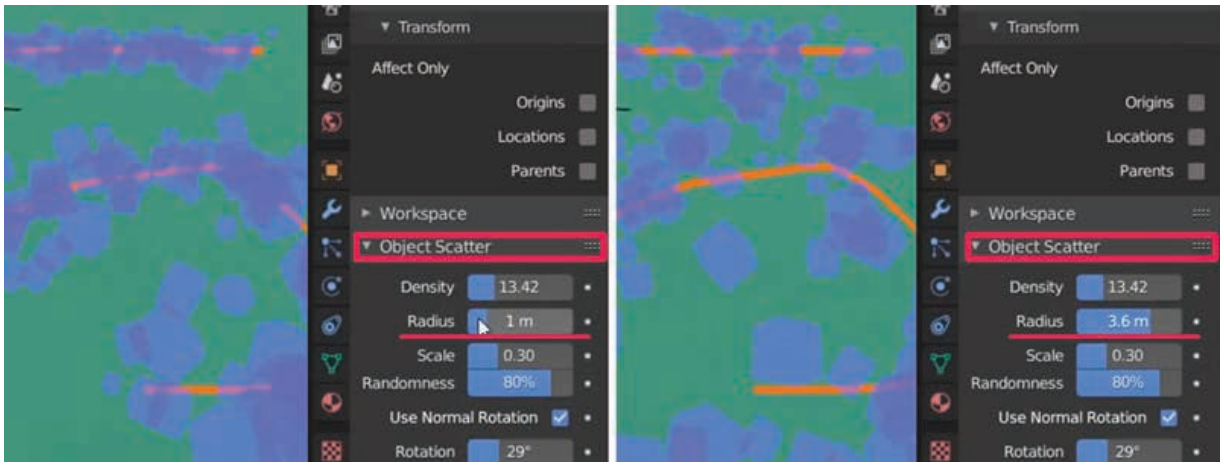
Press F3 for the search menu and search for **Scatter**, then click and "draw" the objects on a surface.



Now what makes this add-on really cool is that you can now change a lot of settings and see visually what you are doing. For example changing the **Density**:

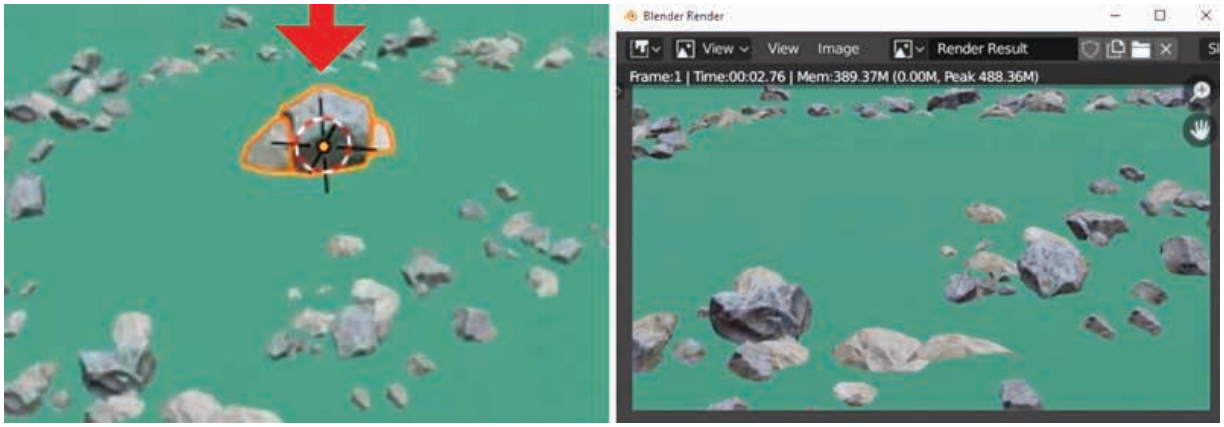


Or the **Radius**:

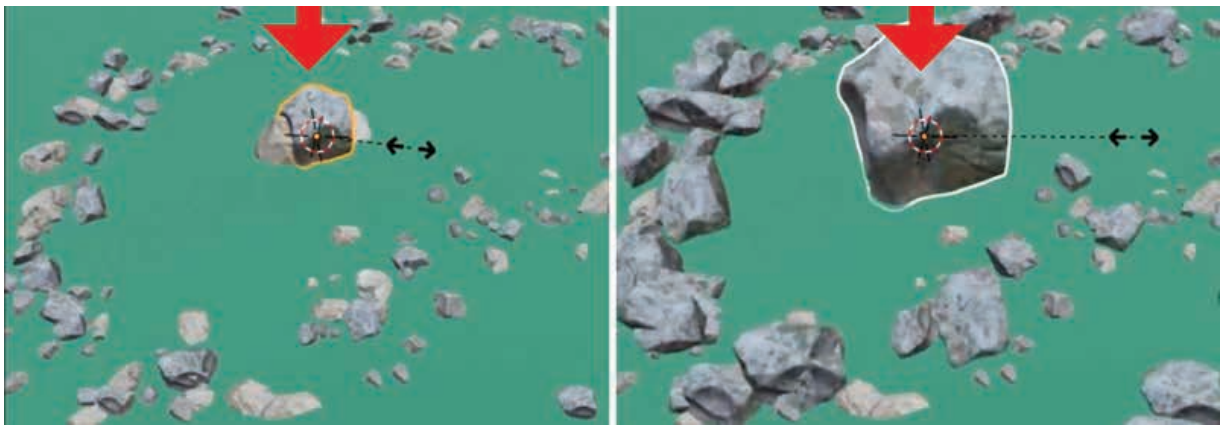


Then **press Enter** to confirm. The blue dots are replaced with the instanced objects. This is really a lot faster and more fun than tweaking Particle Settings options!

The object in the middle is the **Instancer**, which won't show up in the render.

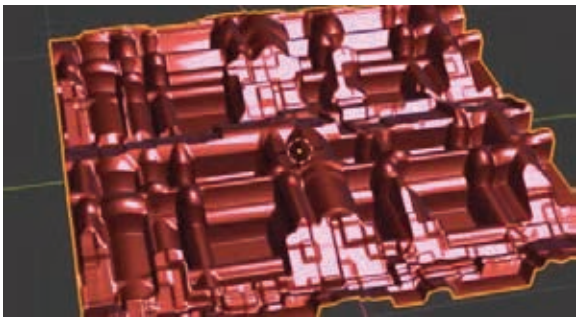


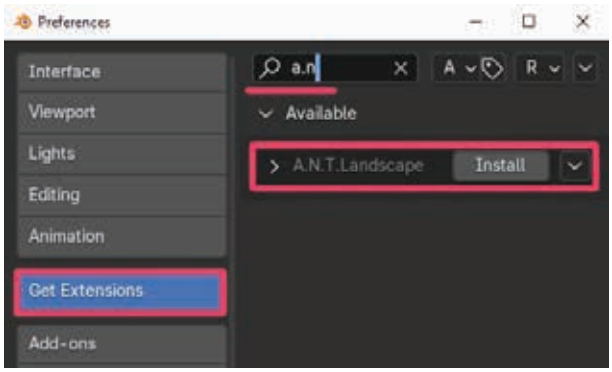
But you can use it to quickly move or scale the instances.



Another Noise Tool (A.N.T. Landscape)

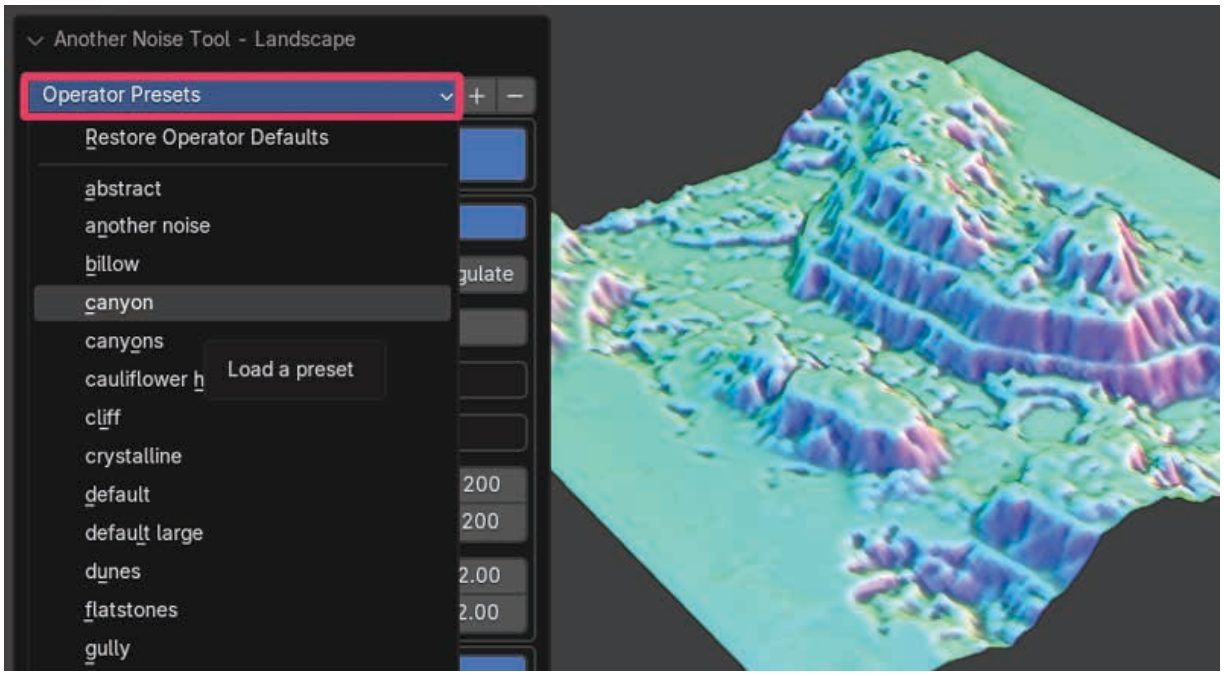
There are many ways to generate "noise" in Blender, and ANT (Another Noise Tool) uses noise to create procedural landscapes and weird geometry.



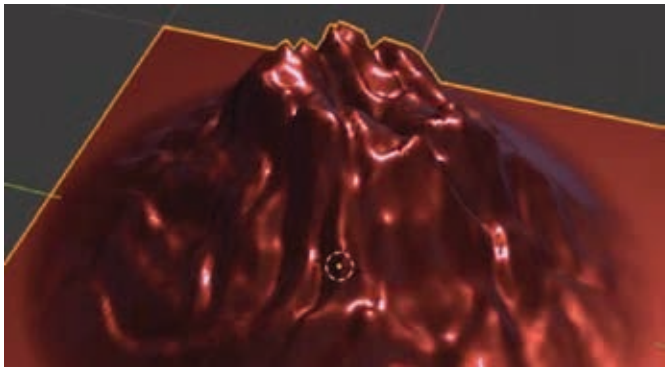


In Preferences activate the **A.N.T. Landscape add-on**. To do this, open the Get Extensions tab and search for "a." to find A.N.T. Landscape, then click on Install. If you search for "Ant" you won't find it.

Go to **Add** > Mesh > Landscape. Or press **Shift+A** and choose Mesh > Landscape. Expand the options panel to will find many presets.



You can increase the size, height, change random seed value and more.



Node-based tree modeling

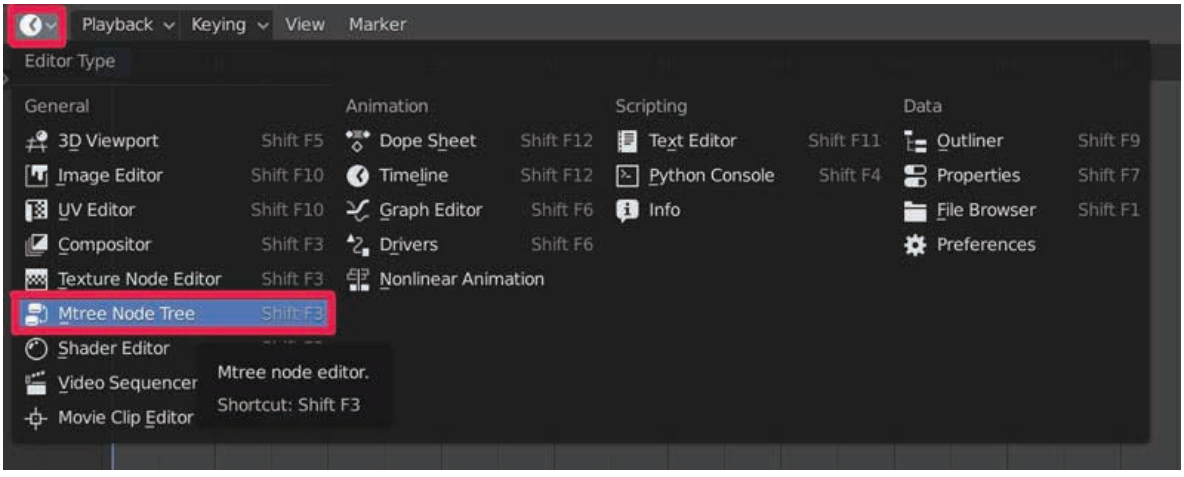
Mtree or **Modular Tree** is a free add-on for Blender by Maxime Herpin. What makes it interesting is that the trees are made by connecting special nodes, like a Trunk node and a Branch node.

This makes it easy to experiment with values in the nodes, while you see the result being updated in the viewport. The resulting tree can become quite realistic - even including roots, a detail often lacking in 3d trees. Conveniently, it comes with some preset materials.

Watch part 1, part 2 and part 3 online.

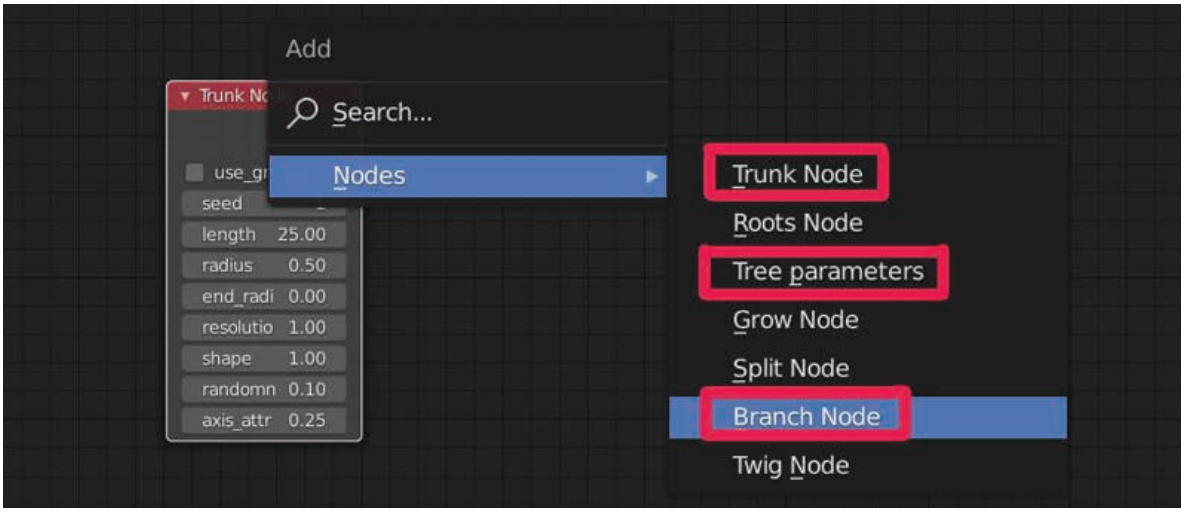
Make sure you get the Blender 2.8 version from Github. Unfortunately at the time of this writing, it doesn't yet work in later versions of Blender. However you can always append the resulting tree in a newer version of Blender.

Mtree has its own Editor type window called **Mtree Node Tree**. Change a window in your viewport which you don't need such as the Timeline, to the Mtree Node Tree window.

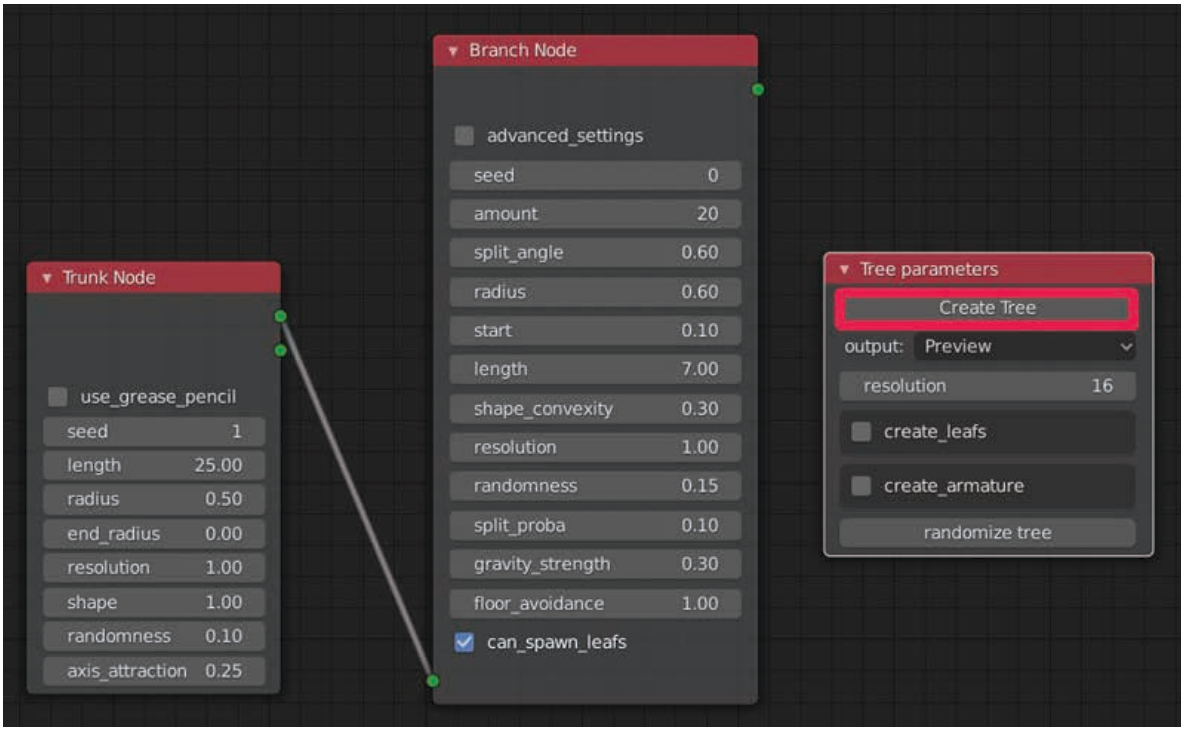


Click on **New.**

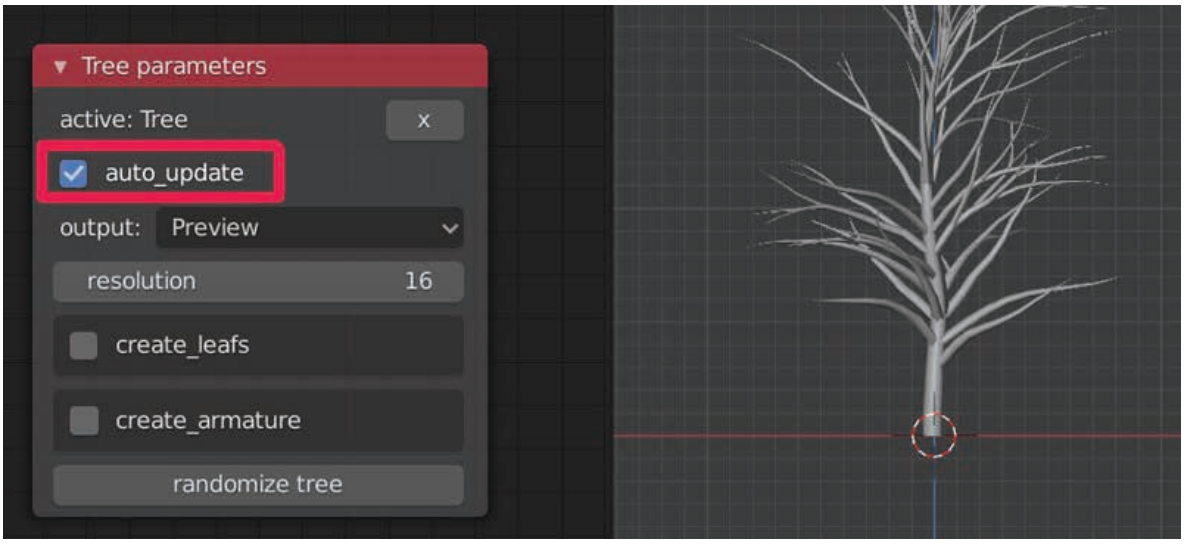
Then **press Shift+A** and add a **Trunk node** as well as a **Branch node** and a **Tree Parameter node.**



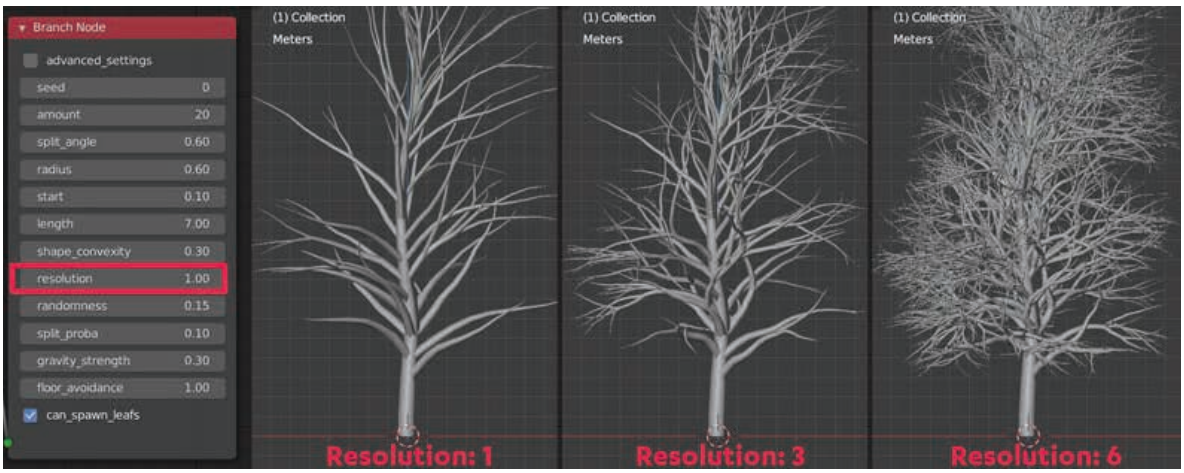
Connect the Trunk to the Branch node and click on **Create Tree**. A default tree shape is now generated in your viewport.



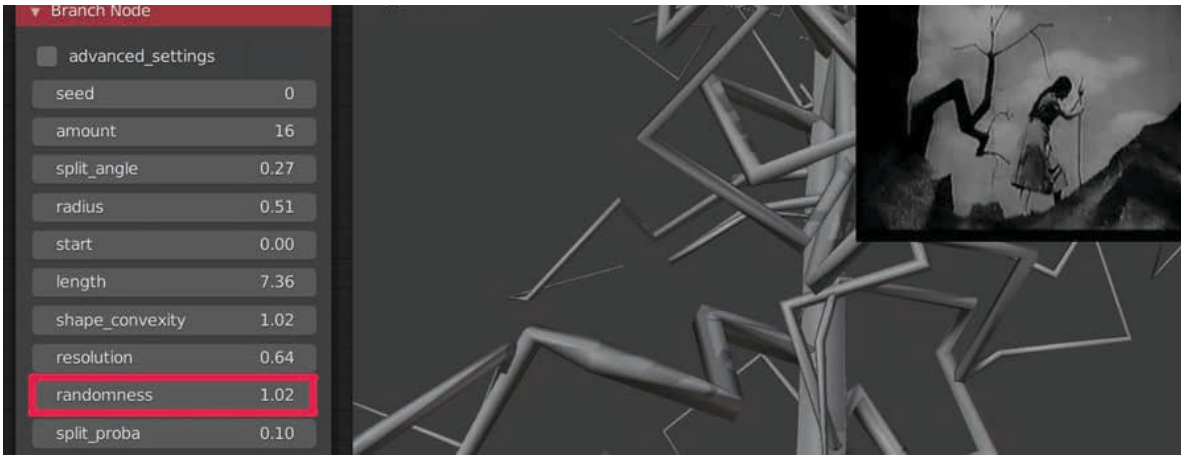
Check the **Auto update** box so the tree will update in the viewport when you change any values. *Note: I've noticed that this box will uncheck by itself from time to time, so if nothing's happening after you change a value, make sure it is still checked.*



Resolution is an important value to add realism to your tree. But be careful to enter a high value here, as Blender may crash. Resolution value 6 is quite high as you can see. I'd say 4 is enough in this case.

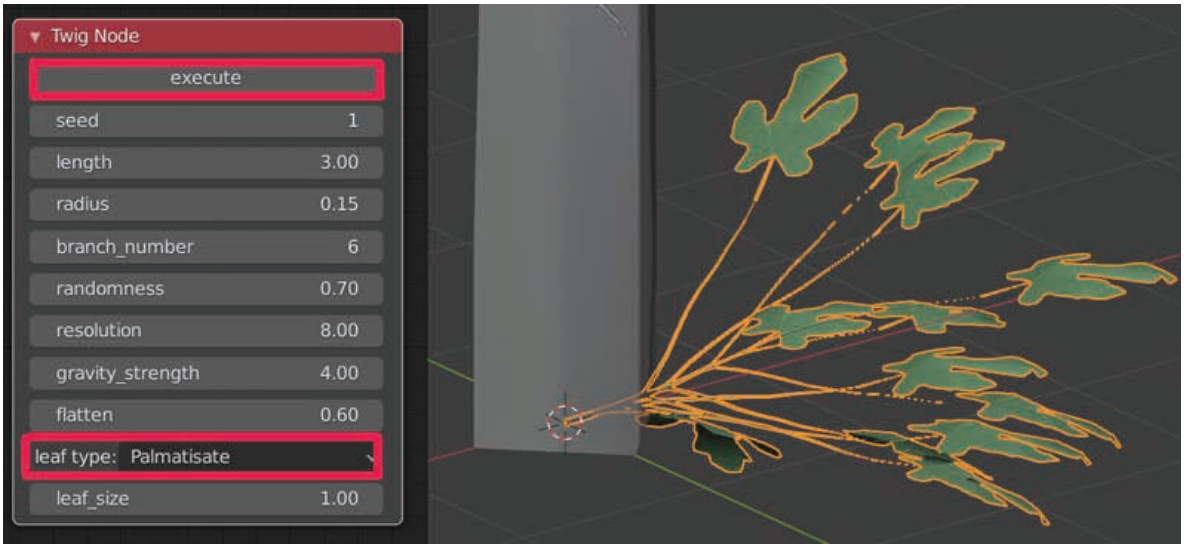


Randomness is for when you are hired to work on a Tim Burton movie or are simply a fan of early German expressionist cinema.

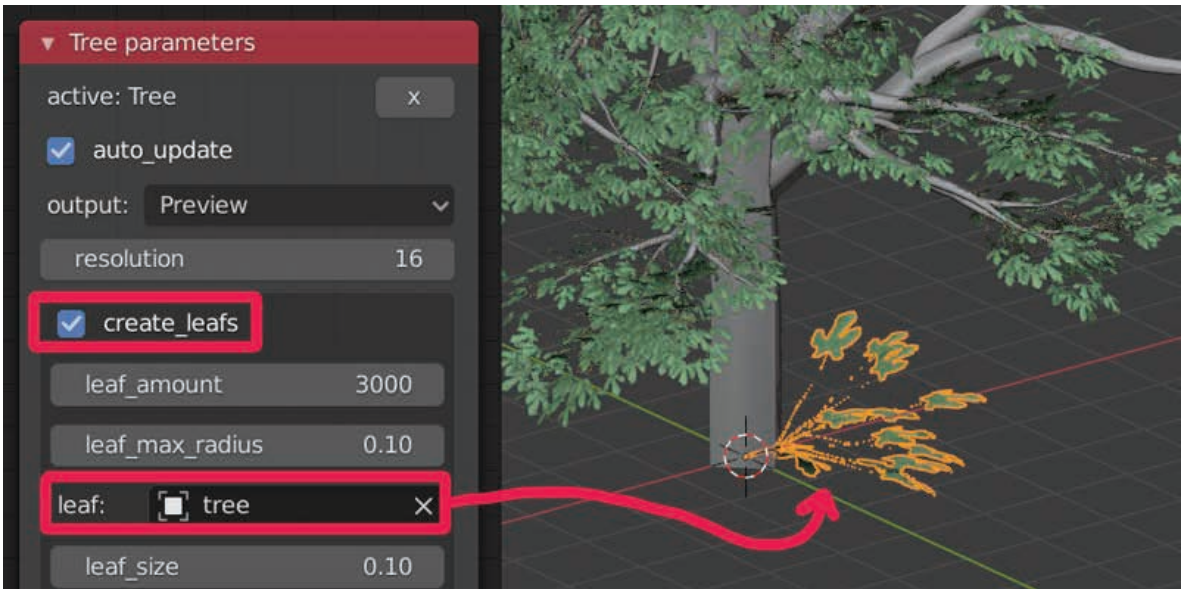


If you ever forgot what the default value was and you want to go back to that, **press Backspace** with the mouse cursor over that value field. This will reset it to the default value.

Add a **Twig node** and **click execute** to create a twig with leaves. The Twig node doesn't need to be connected to other nodes. You can choose a different **leaf type**-just **click execute again** to see the changes. The twig object will be made at the world origin. To use your own leaf model, choose the **Custom leaf type**.



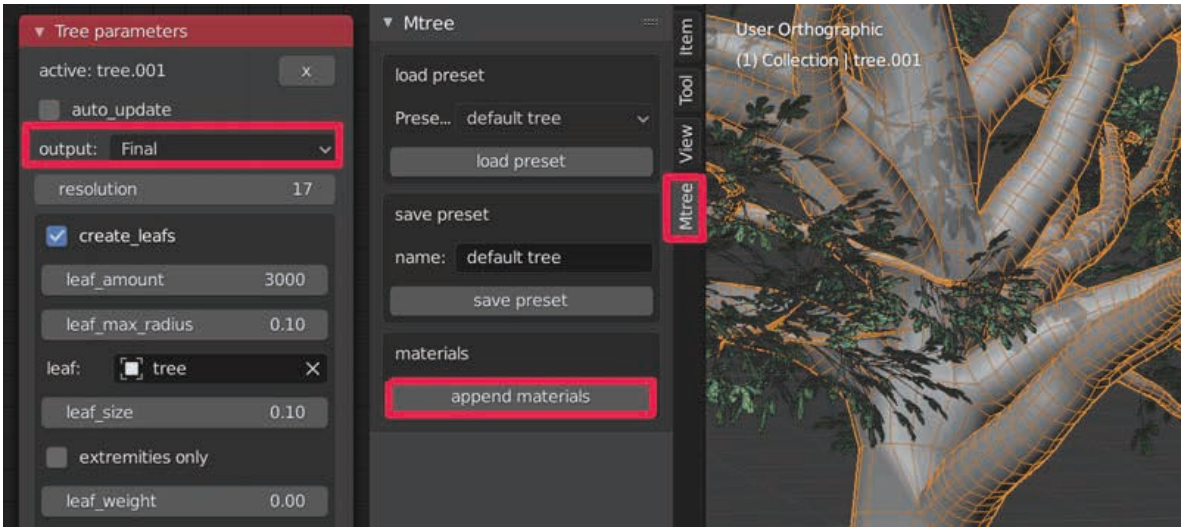
Check **create_leafs** and pick the leafs object in the viewport to add them to your tree.



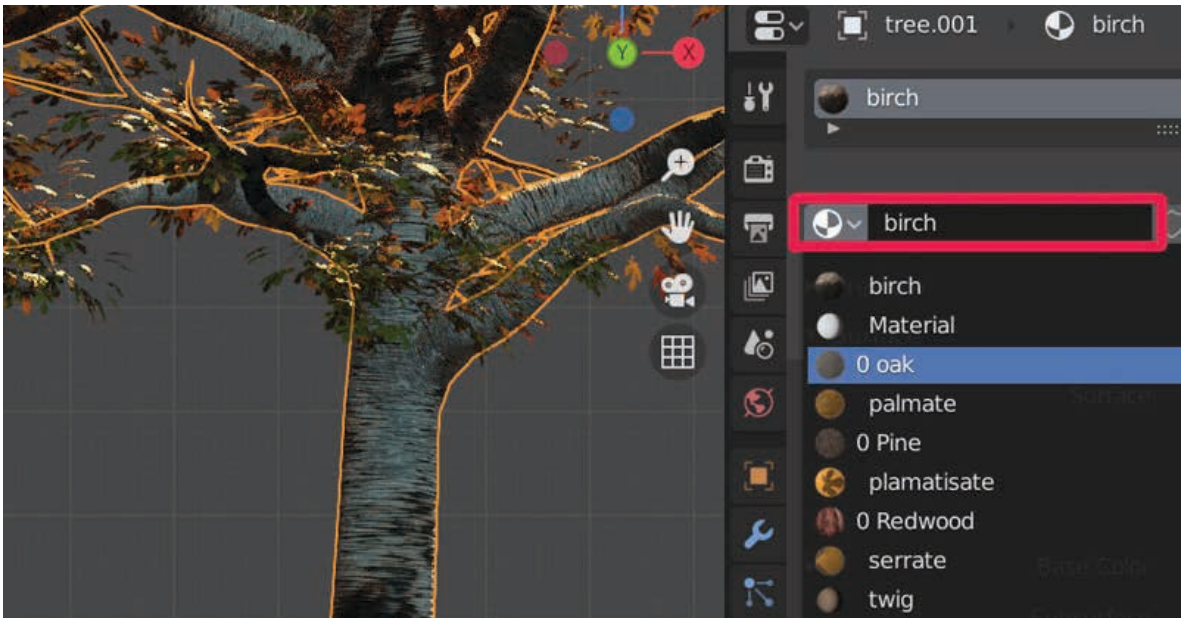
Once that's done you can delete the original twig object.
That's already a nice-looking tree. But it gets better! The good news: it is already UV-unwrapped and comes with textures. Let's see how we can apply those.

First, change the tree output from Preview to **Final**. This creates a better looking mesh, where the branches actually are connected to the trunk.

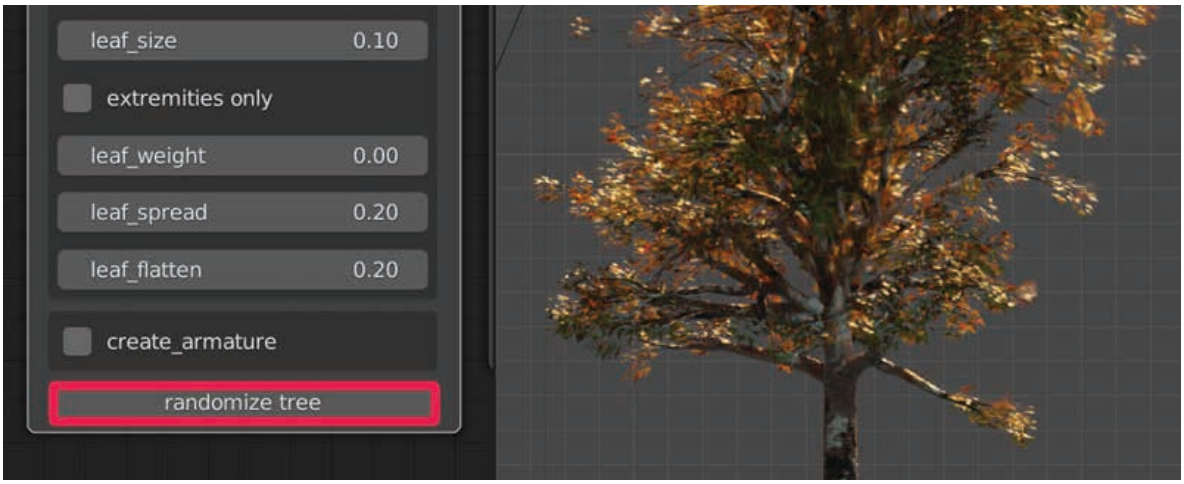
Press N to open the Option panel and with the Tree trunk selected in the viewport, click on **append materials**. This loads some preset materials. It's important that the tree is selected though, otherwise it won't work.



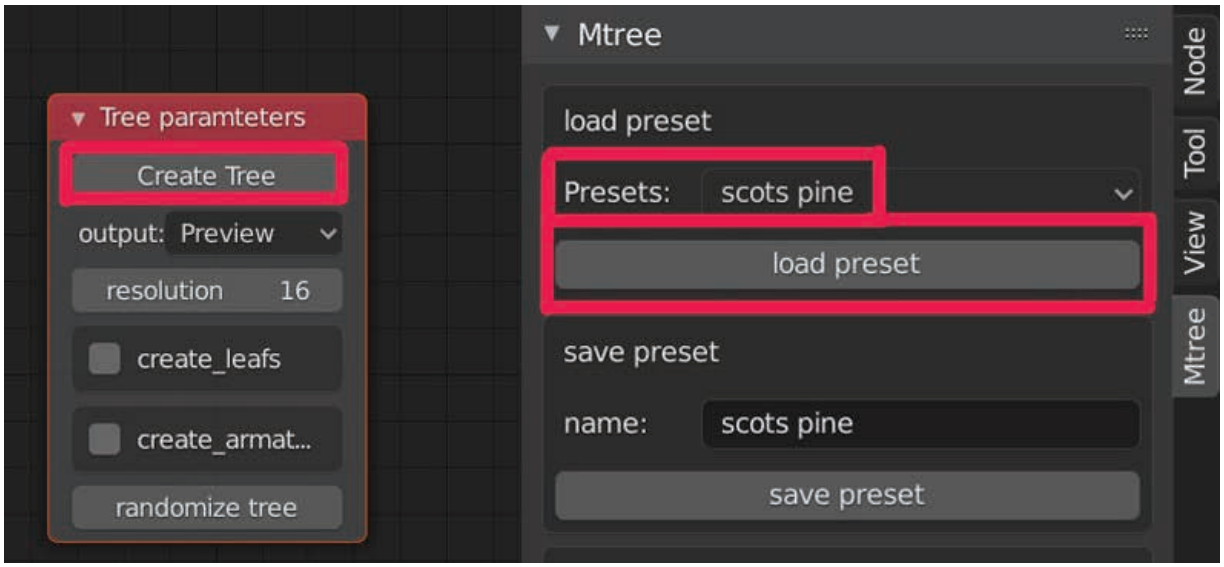
Now you can select some of the preset materials to your trunk. I've noticed that this doesn't work very well in Eevee - if you want to switch a material, go into Cycles render mode first. After that you can go back to Eevee and it will work. You just can't switch materials in Eevee, this is some limitation of the add-on.



That's it! You've made a very nice-looking tree. If you want to see some random variations, click on the **randomize tree** button on the **Tree Parameters node**.



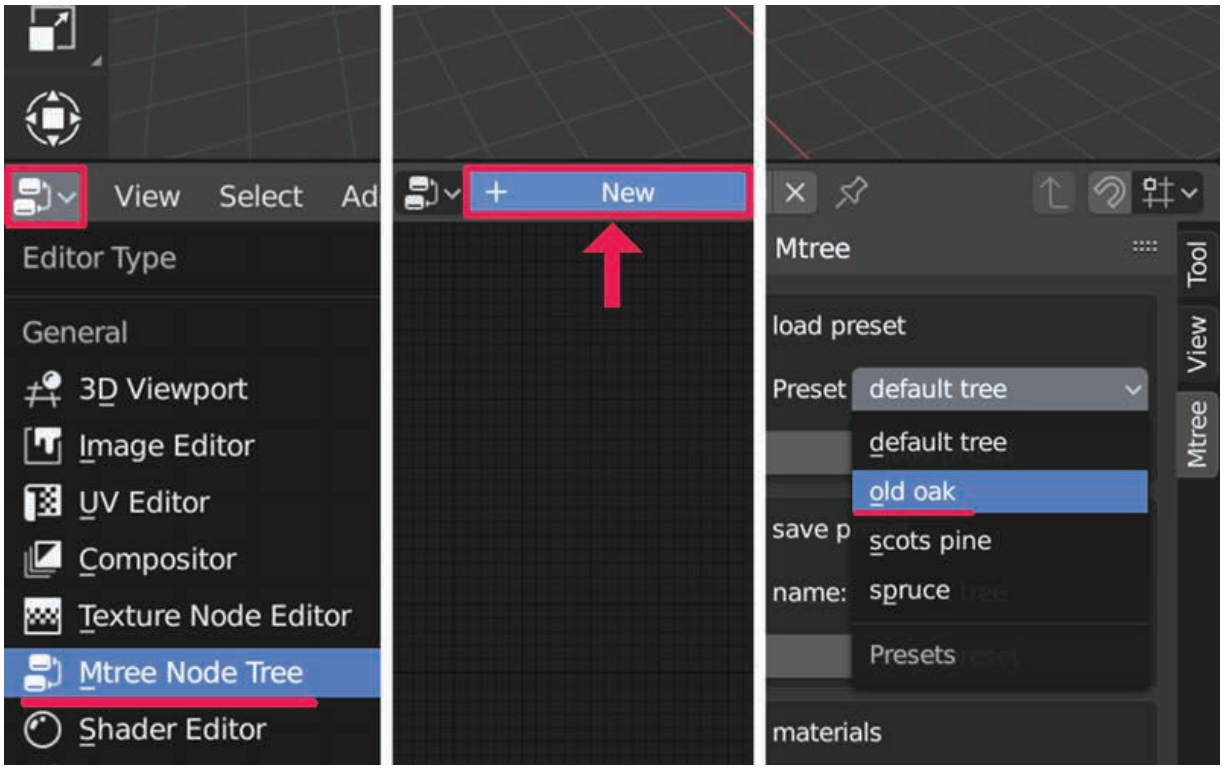
If all this seems like too much work for you, I have good news. You can simply select one of the presets in the Mtree option panel, **click on load preset** and then **click on Create Tree**.



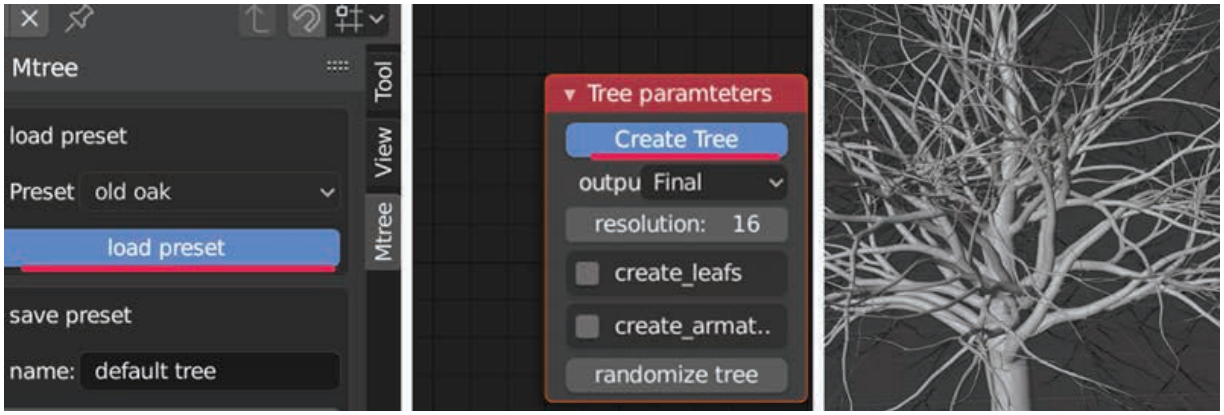
Good luck making your trees!

MTree node-based tree add-on Materials and presets

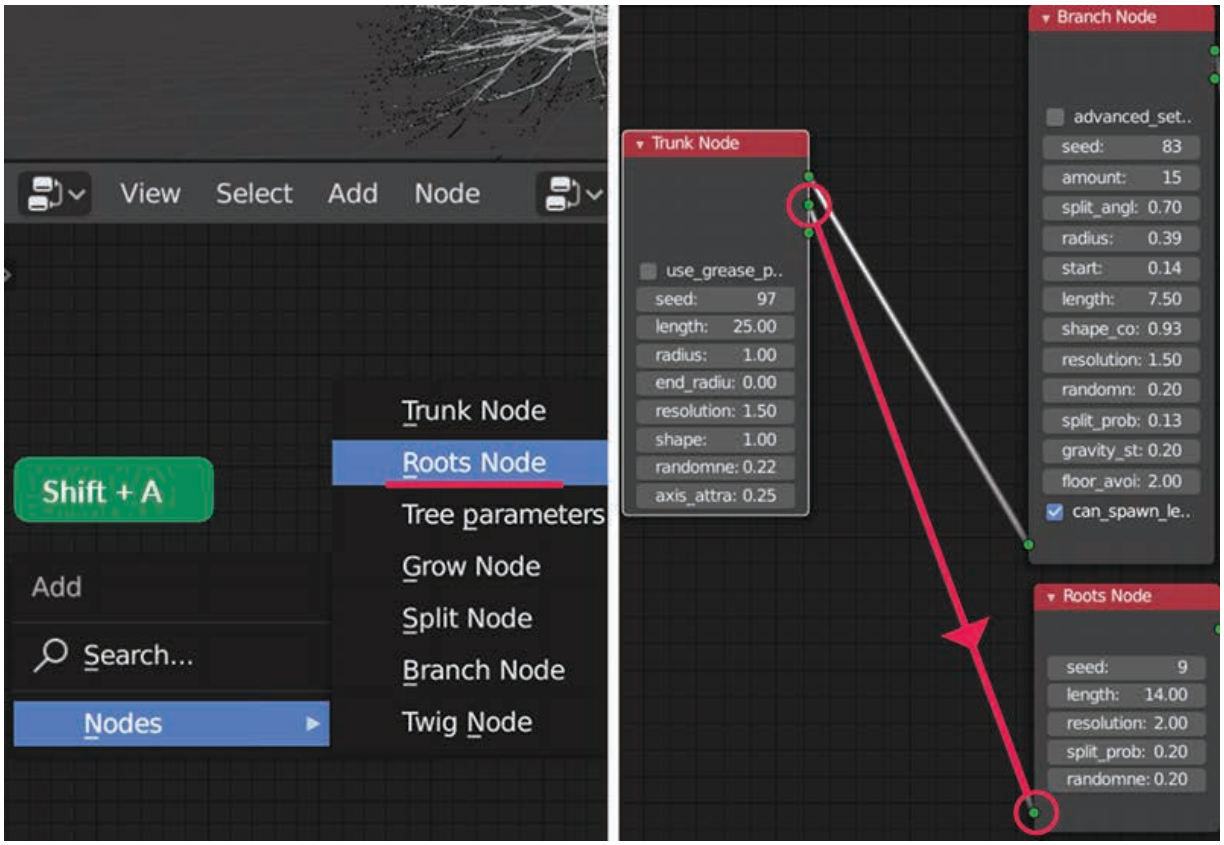
The MTree add-on we looked at previously, also comes with some materials and presets that can save you time. After **clicking on new** in the Mtree Node tree window, you open the Mtree tab and choose a preset.



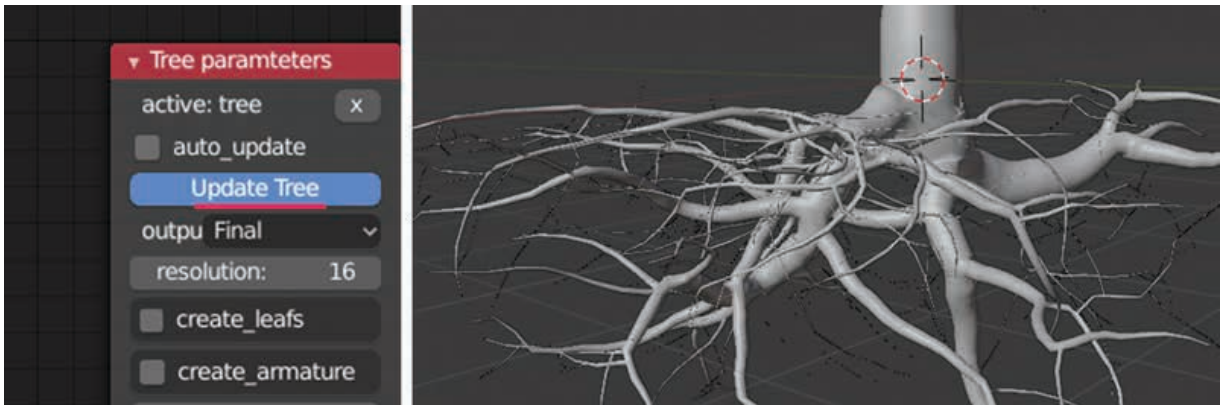
Click on **load preset** and then on **Create Tree** on the **tree parameters node**. This is the old oak preset.



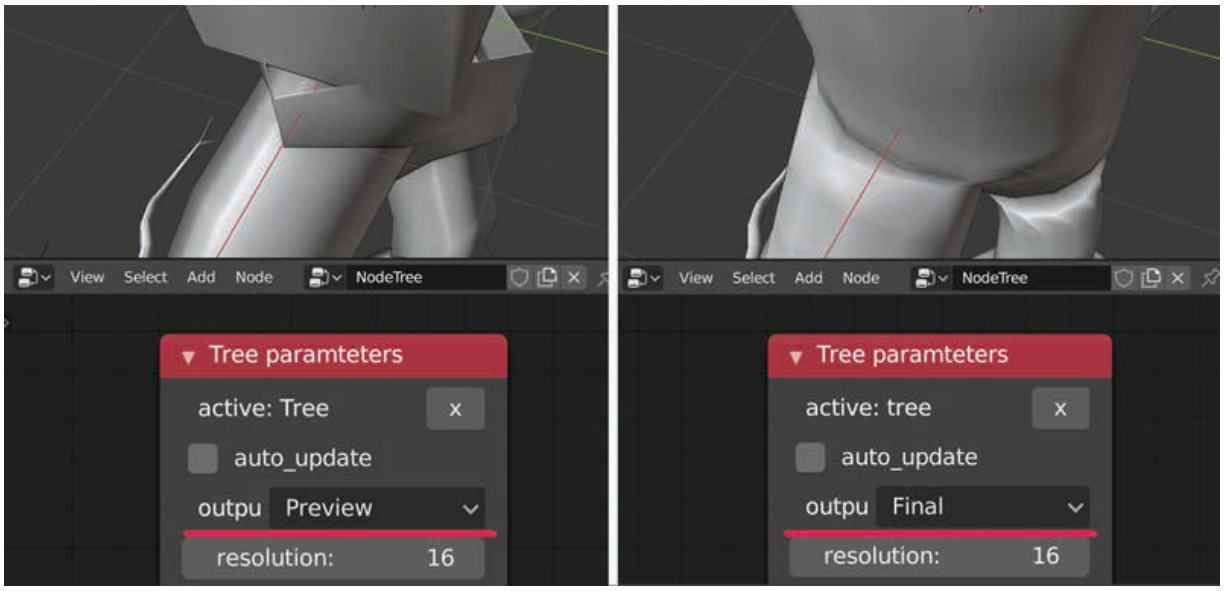
I've added a **roots node** as well to make it more realistic.



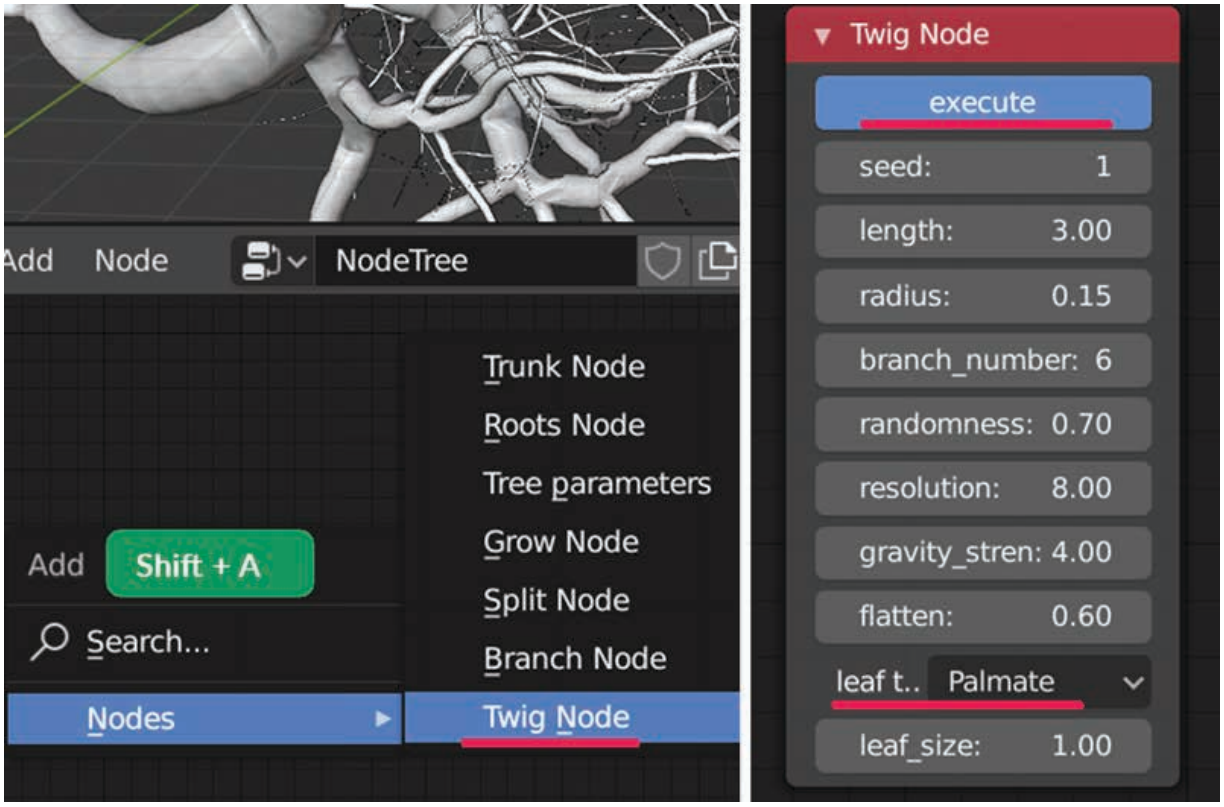
Click on **Update Tree** to see the changes.

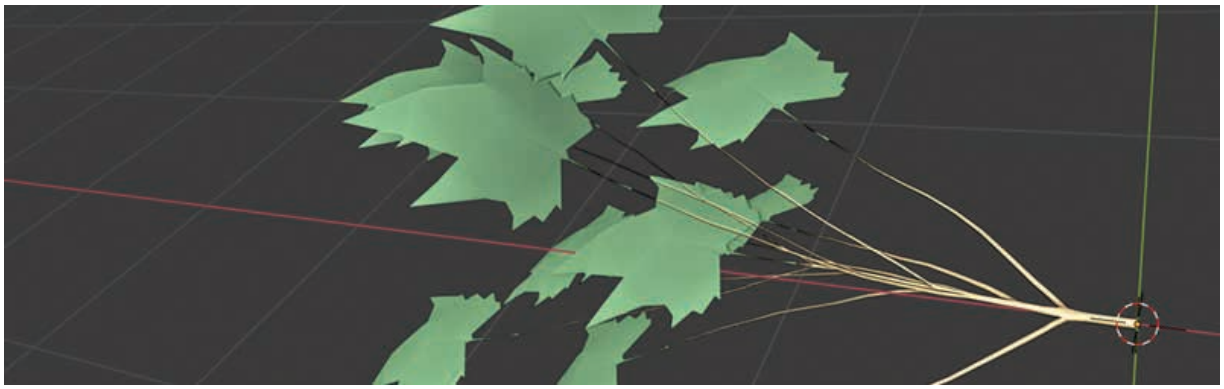


If the branches don't really connect to the trunk you can set output to final.

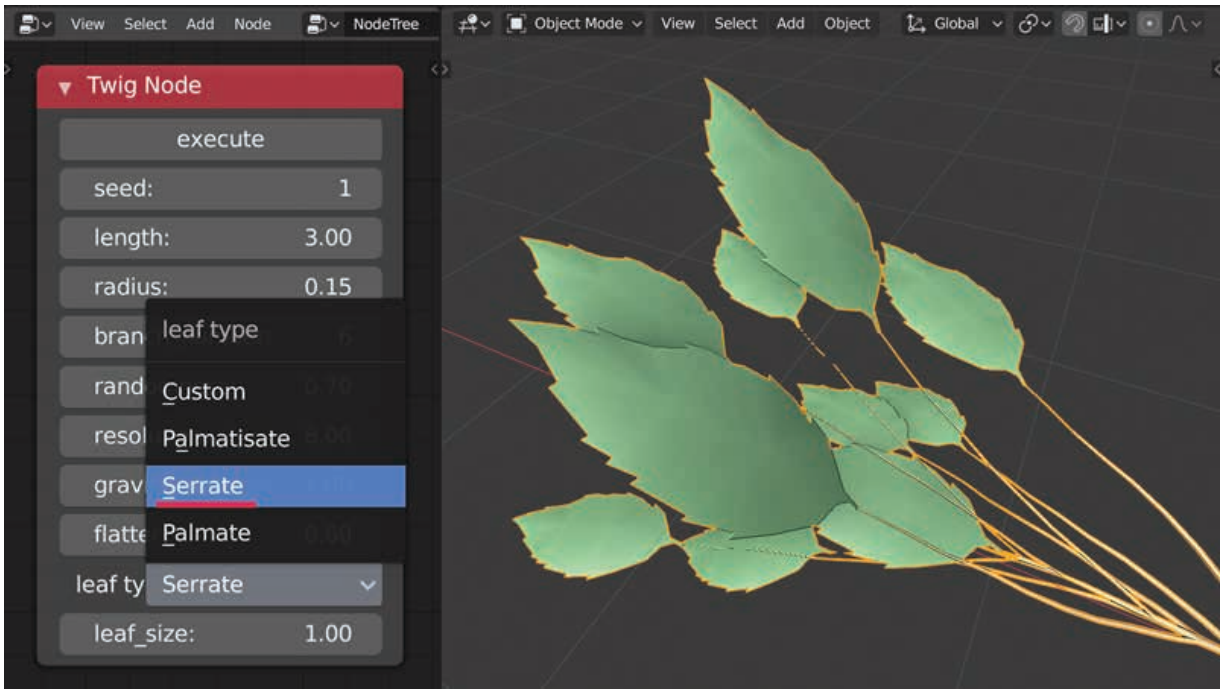


If you create a **Twig Node** and click on **execute**, when you go into material view, you will see that it already has a material. This example is the **Palmate** leaf type.

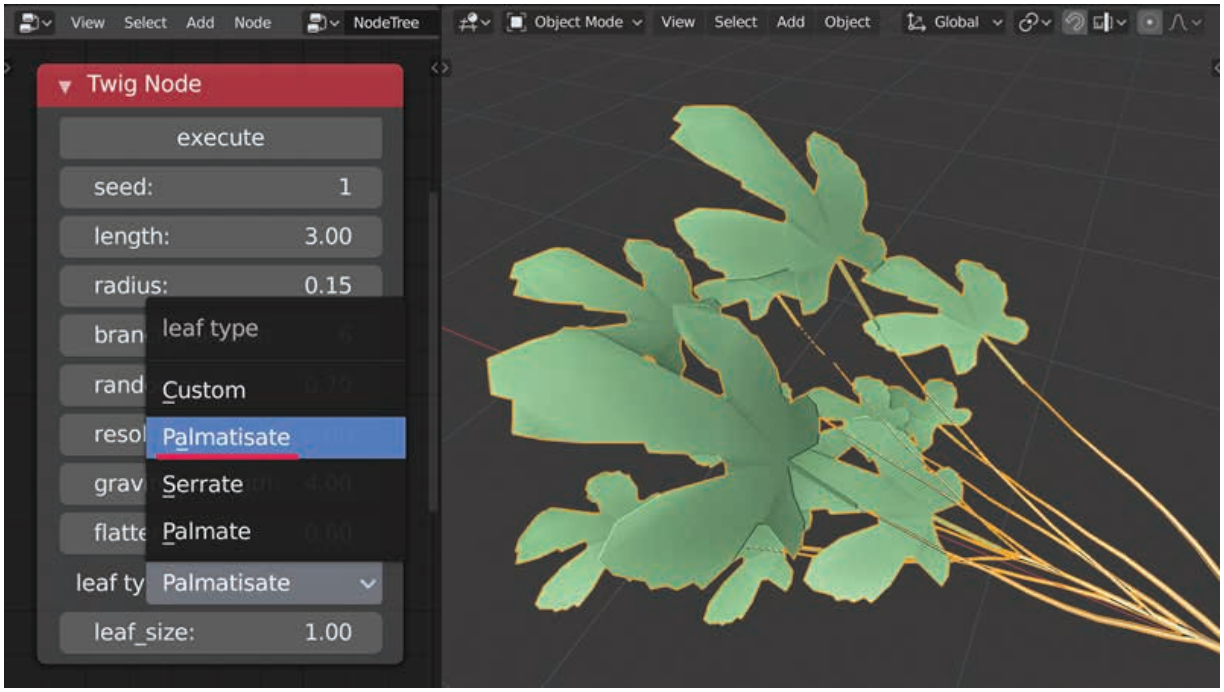




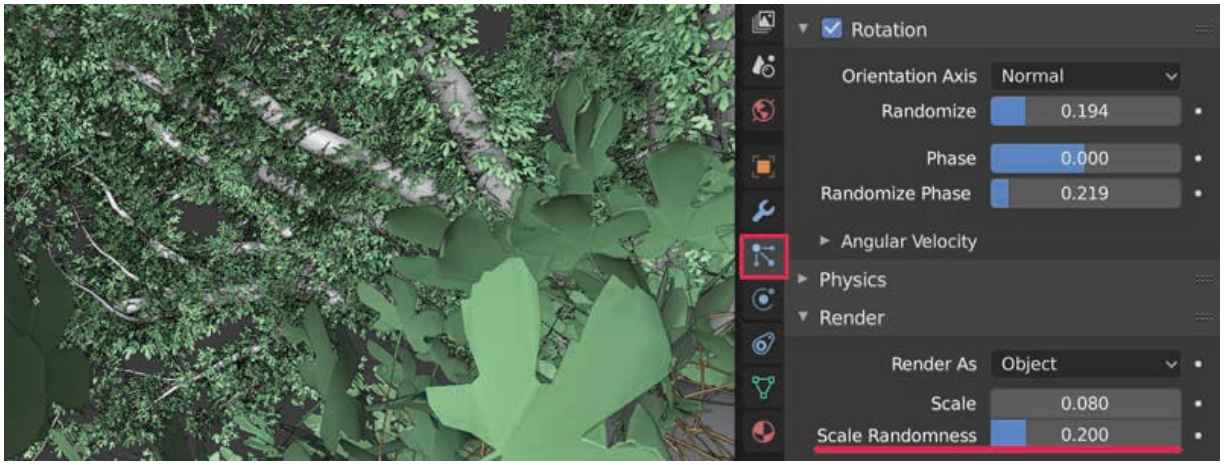
You can choose from 3 preset materials in the node under **leaf type**. This example is the **Serrate** leaf type.



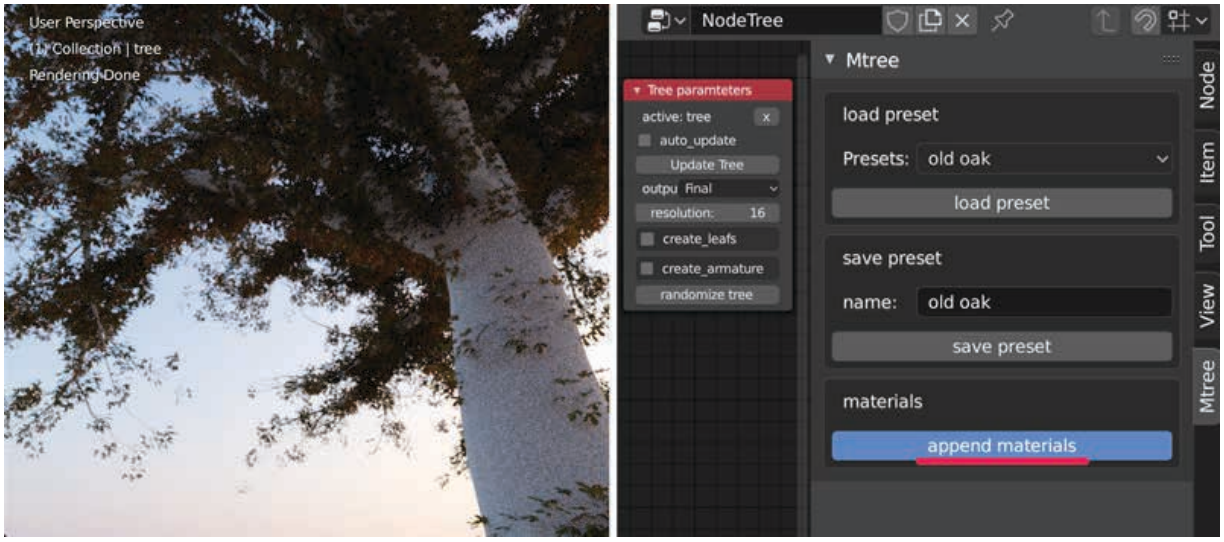
And this example is the **Palmatisate** leaf type.



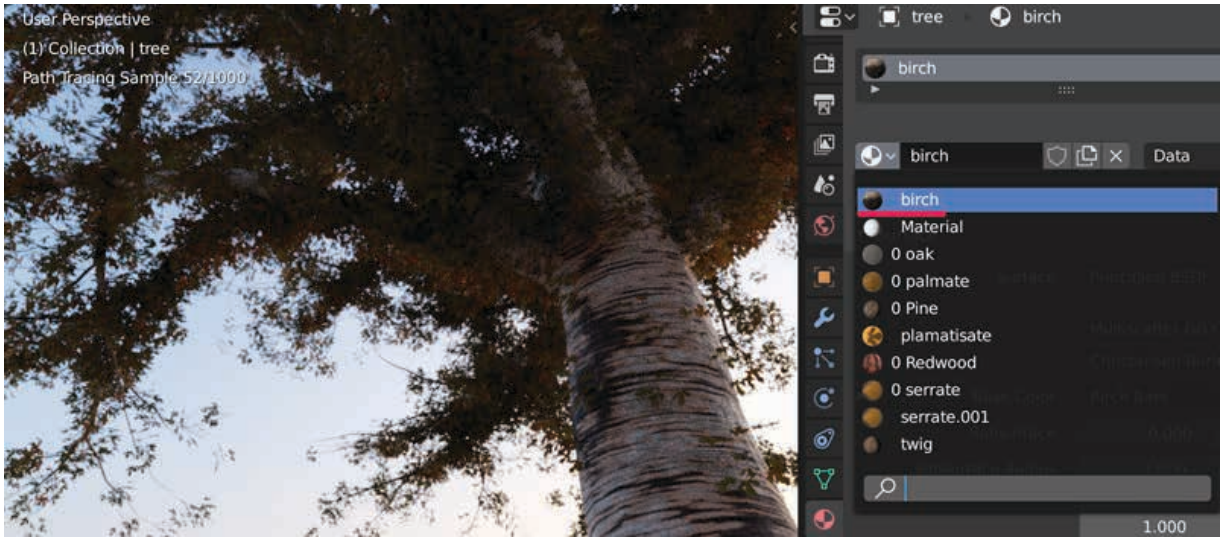
The twigs are distributed with a normal particle system found in the Particle System tab, so you can randomise the scale, for example, by changing the **Scale Randomness**



To add a material to the trunk, click on **append materials** with the trunk selected.



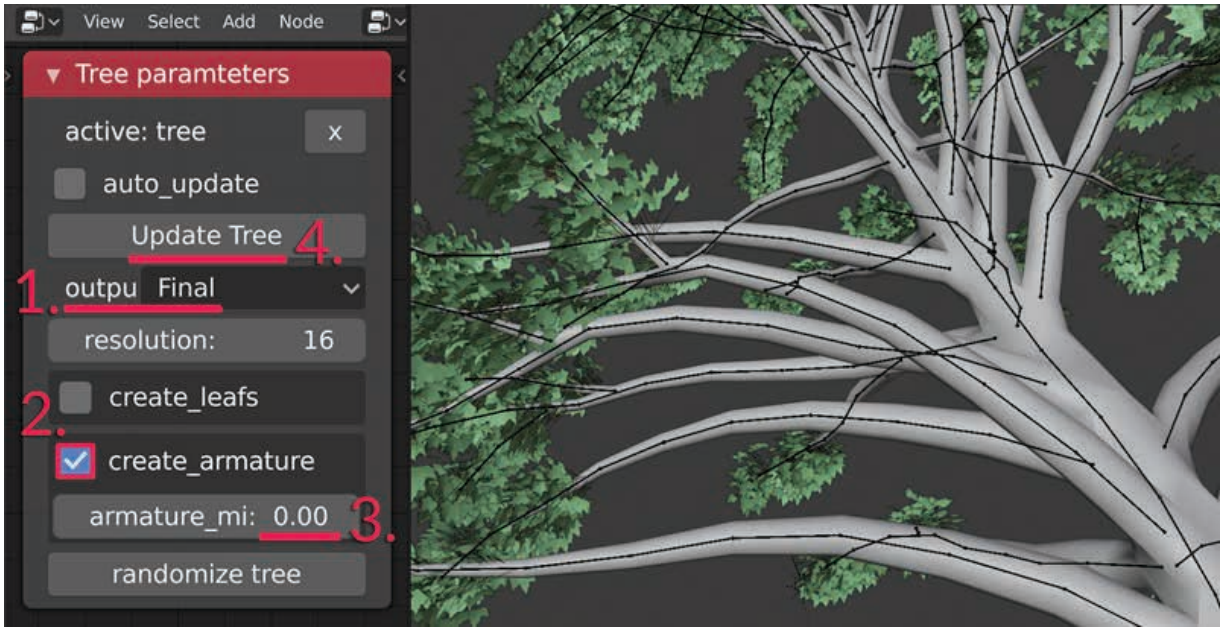
Then in the materials tab you can choose a material from the drop-down.



Easy Tree Wind Animation with free Mtree add-on

Previously we looked at creating a tree with the MTree add-on and using presets and materials that come with the add-on. In this topic we'll animate the tree. Create a tree as shown in the previous 2 topics and turn on **Auto Update**. If you want to experiment with the blend file from this current topic, you can download it here.

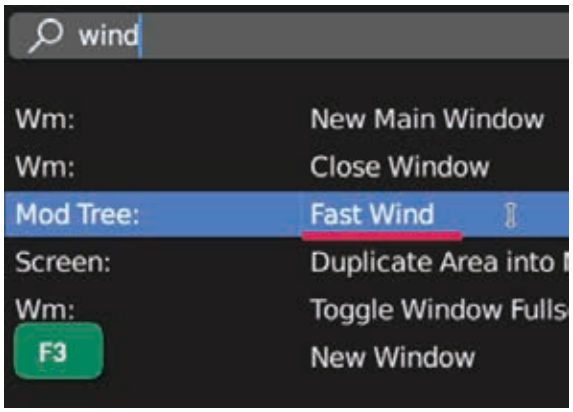
Set it to **Final**, otherwise the next step doesn't work, and check **Create Armature**. Set **Armature_Min_Radius to 0.00**. Click on **Update Tree**.



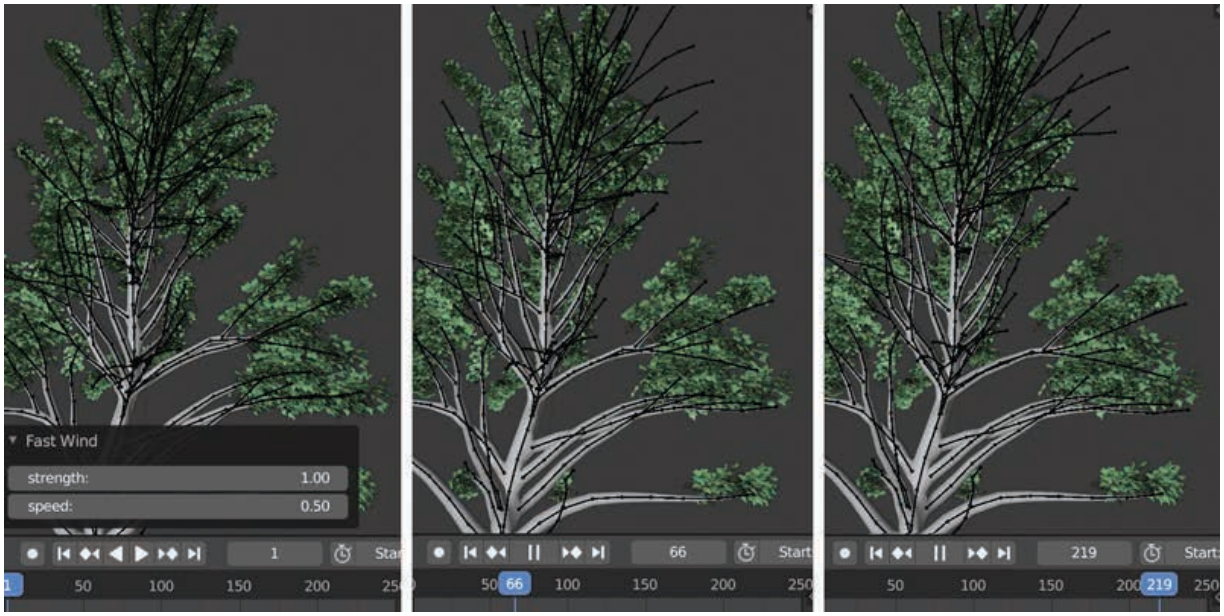
The tree is automatically weight painted and parented to the armature. In **Pose Mode** you can animate the tree branches manually if you want.

However there is a quick way to animate the tree.

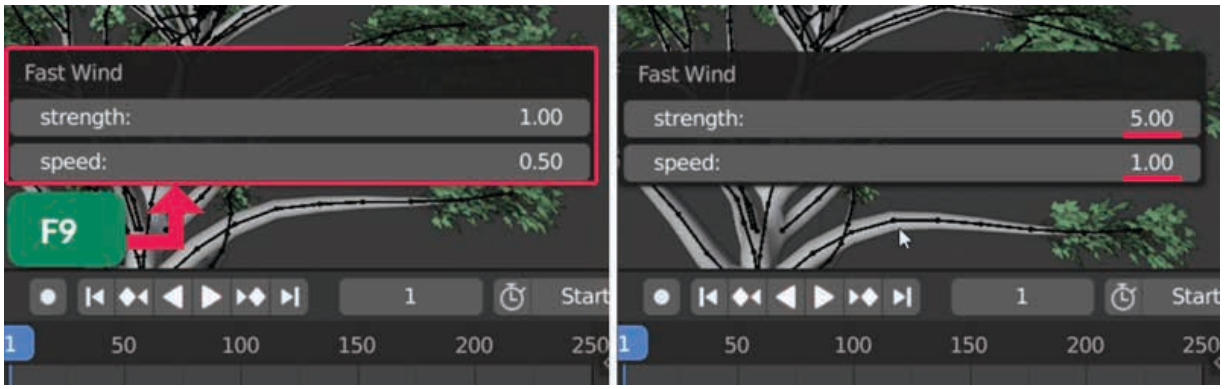
With the armature selected **press F3** and search for "**Fast Wind**".



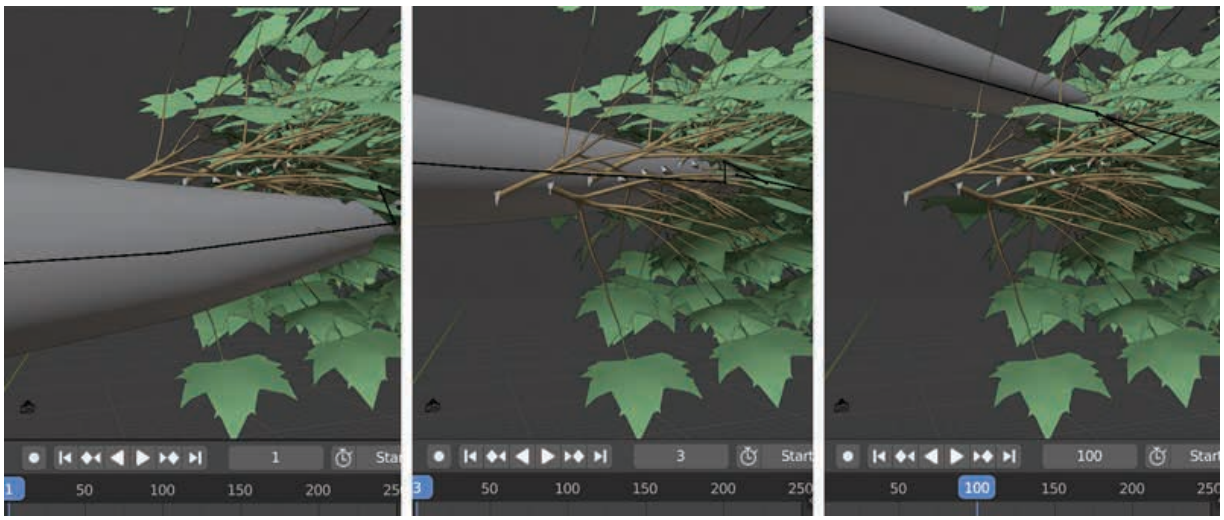
If you **press the Spacebar** to play the timeline, you'll notice that the tree gently moves, as if by the wind.



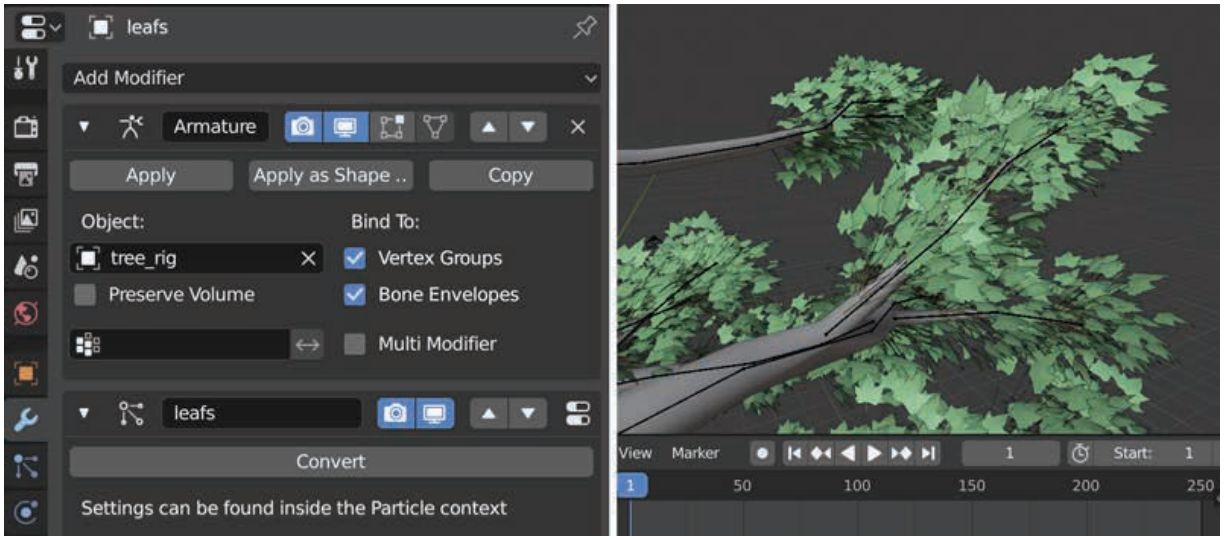
You can open the **Fast Wind options** and increase the **strength** and **speed**.



Notice that the leaves don't move with the tree.



To solve this, add an **Armature Modifier** to the leaves. Place it above the particle system and check **Bone Envelopes**. Then, as the object, choose the **tree_rig**. Now everything works as it should, with the leaves following the tree as it moves.

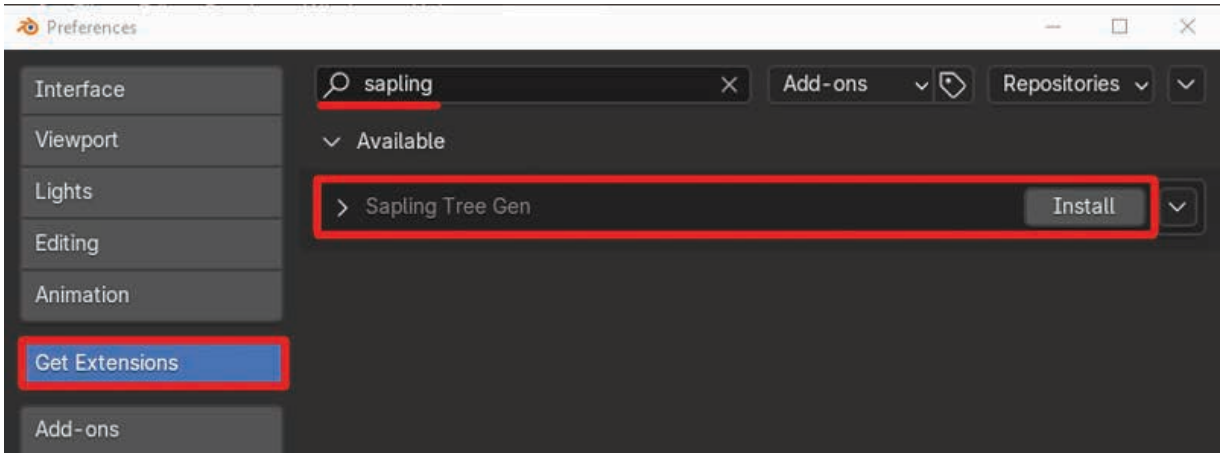


Easy tree and leaves wind animation with Sapling

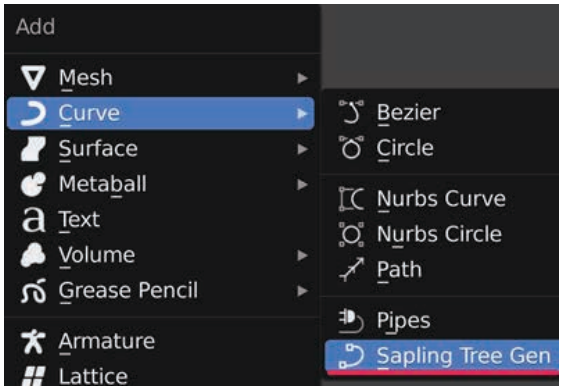
Sapling is an add-on that comes with Blender by default, which is used for quickly making trees. The advantage of Sapling is that it is really very easy to make nice looking tree animation, with very intuitive customizable settings for the wind movement and even for animating the leaves, which makes the whole thing more realistic.

Comparing **Sapling** to **Mtree**, I'd say that the benefit of Mtree is that you can create a really nice looking tree including preset materials. For realistic animation Sapling has the advantage that it adds nice leaf animation, which is important for realism.

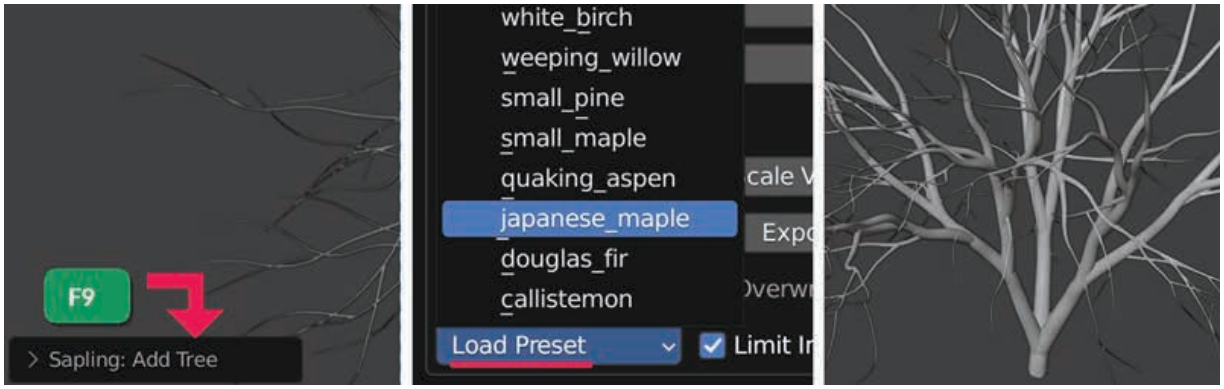
Enable the add-on in Preferences in the **Get Extensions** tab.



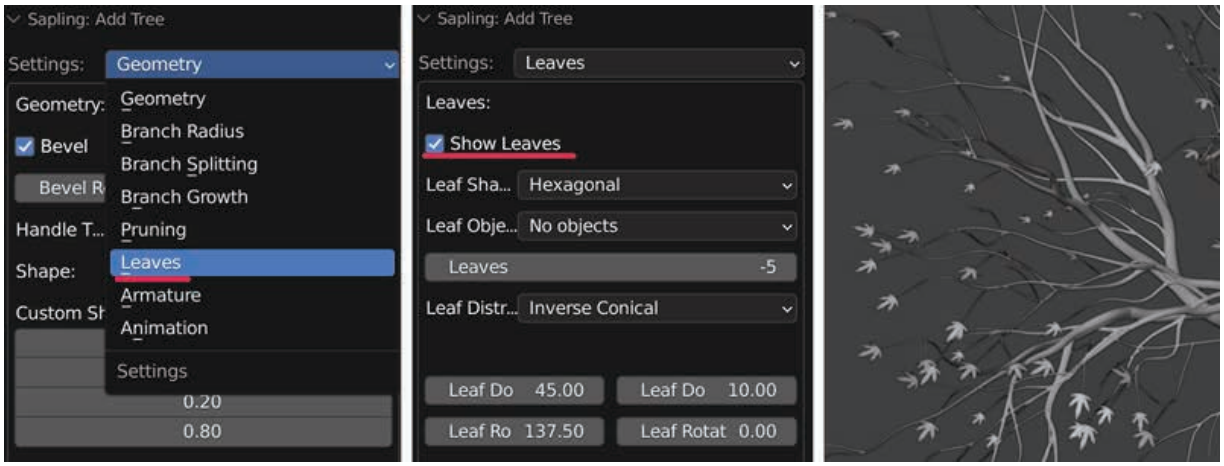
Press Shift+A and choose Curve > **Sapling Tree Gen**.



If you accidentally close the settings you can bring them back by **pressing F9**. Sapling comes with presets and there are a lot of settings that you can customise.

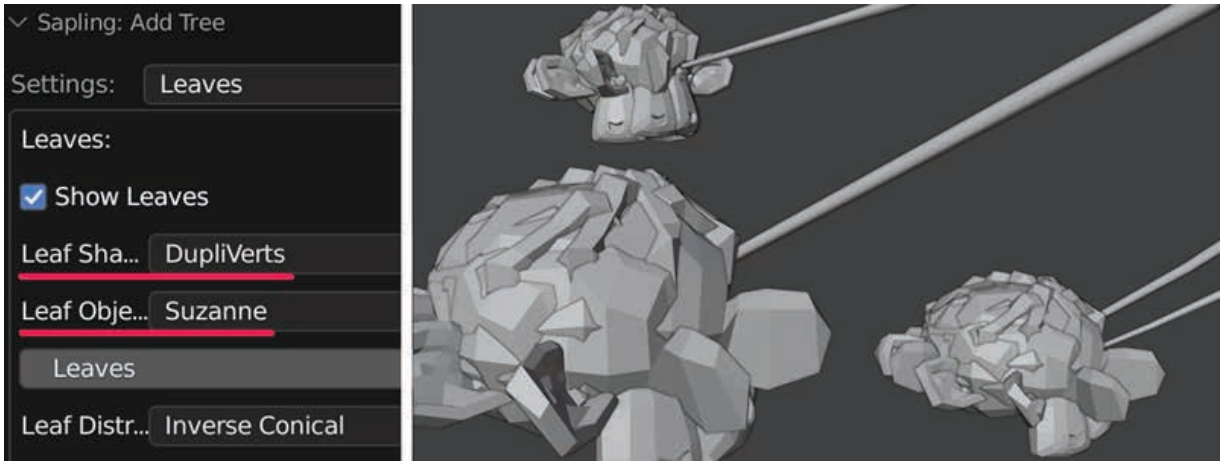


For leaves go to the **Leaves settings** and check **Show Leaves**.



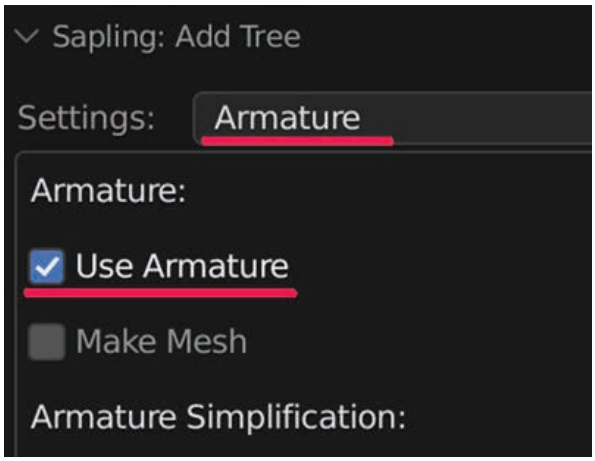
You can choose a preset leaf shape: Hexagonal or Rectangular. Rectangular is more low-poly, but you'll need to use a texture with Alpha transparency which increases render times.

Or you can choose a **Leaf Object** and in combination with the **Dupliverts** Leaf Shape choose a leaf that you modeled yourself.

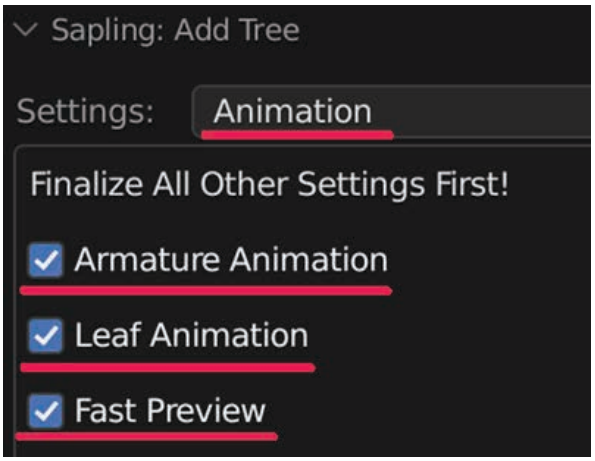


Creating animation with sapling is super easy.

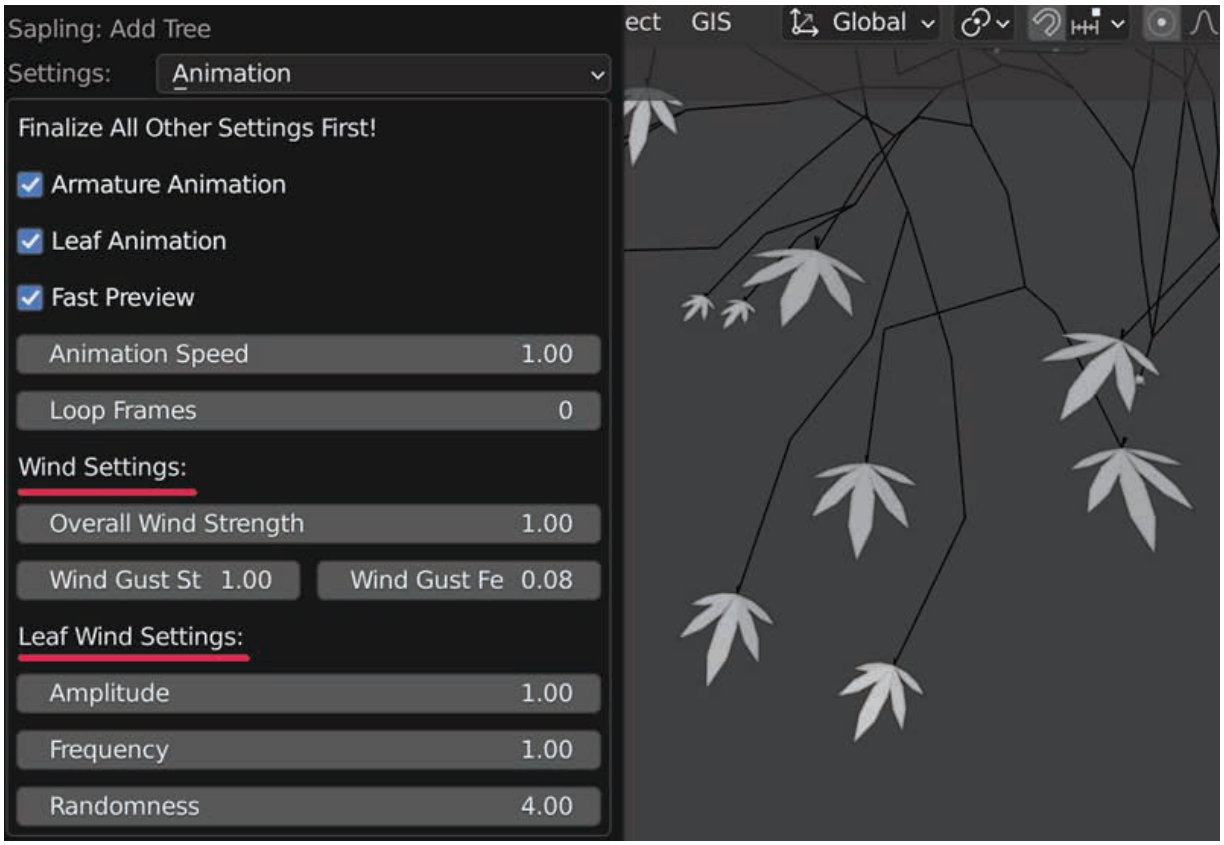
First you go to the armature settings, and check the **Use Armature** box.



Then go to the animation settings and check the **Armature Animation** as well as the **Leaf Animation** boxes. When changing these settings check the **Fast Preview** box.



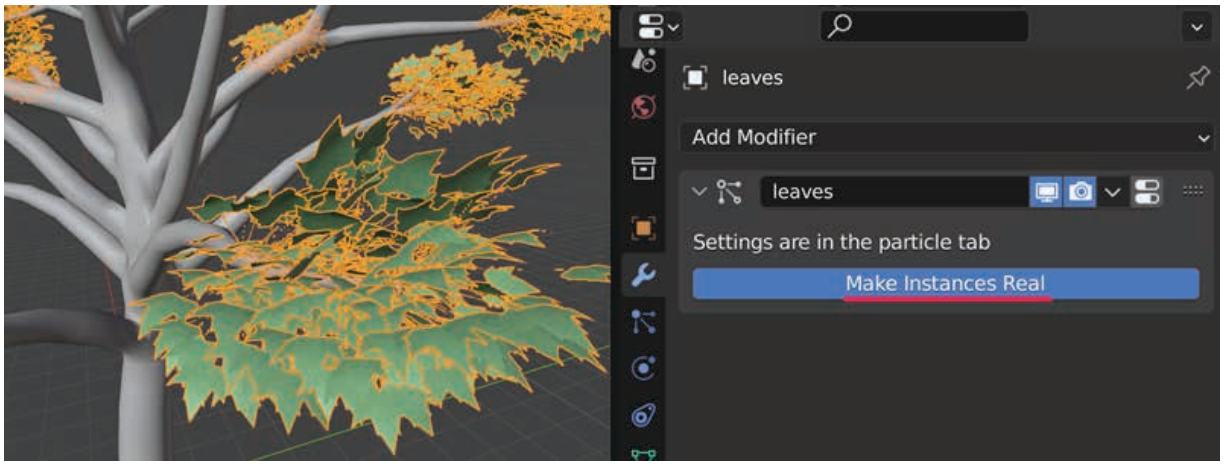
This shows the tree as sticks so it plays more smoothly and you can see better what you are doing when changing the values. You can edit the values for the trunk and the leaves separately.



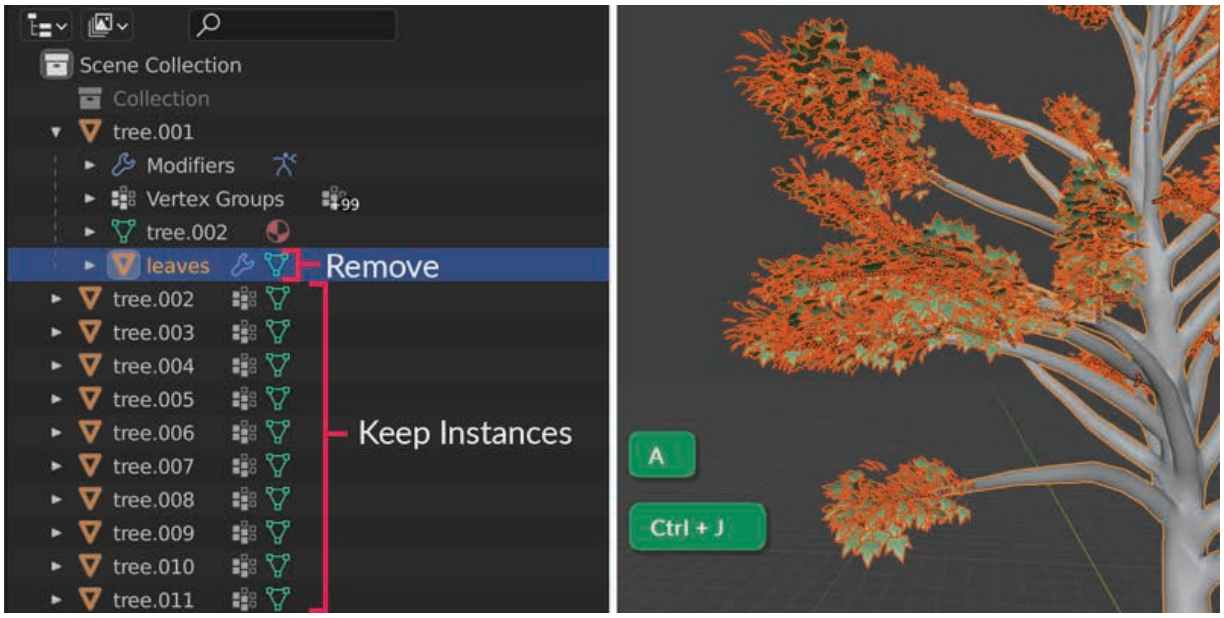
Make a forest

There wouldn't be much point learning all these tree modeling facts, if at some point we weren't going to make a forest, right? So here it is - the obligatory forest tutorial video! Although to make it interesting, I've made it more of a tiny forest planet. You can download the Blend file here.

Convert the particle system to real geometry by clicking on **Make Instances Real**.

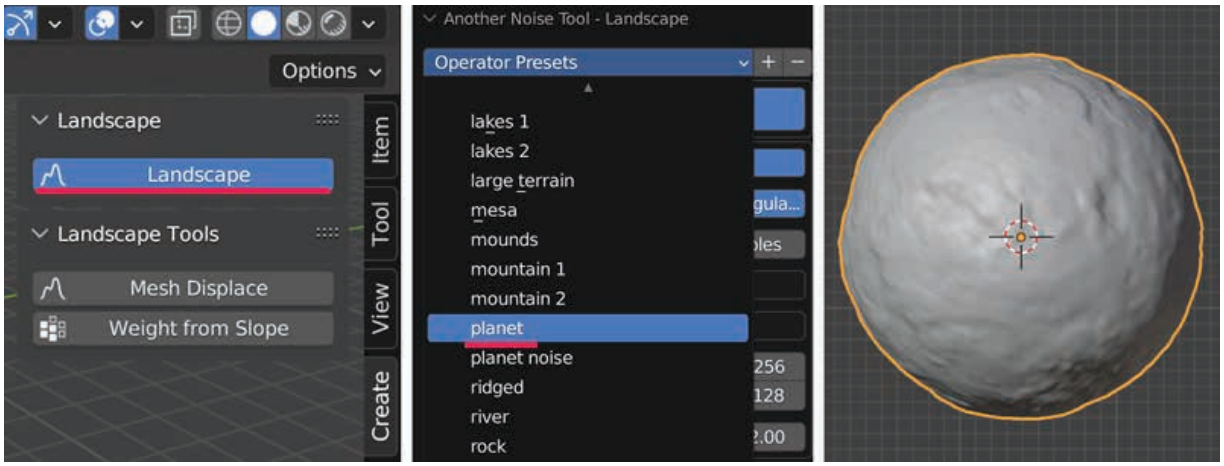


Remove the particle system and join everything together so the tree is one object by selecting all while holding Shift and **pressing Ctrl+J**.

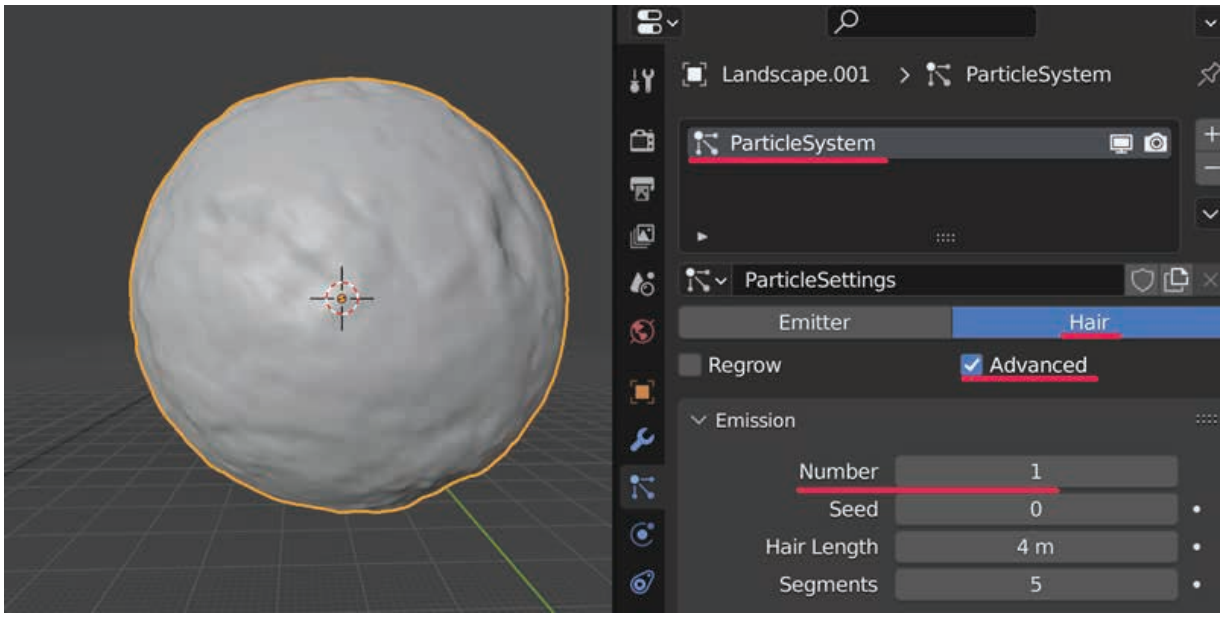


Move the tree to the side on the X-axis by pressing G+X.

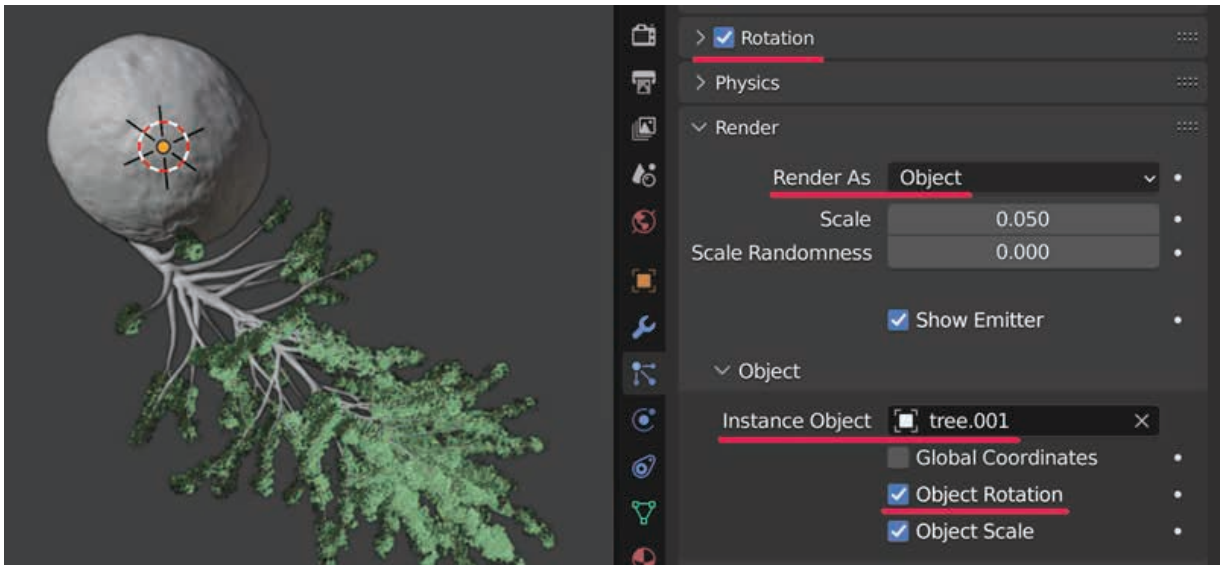
Create a terrain for the trees using the **A.N.T. Landscape add-on**.



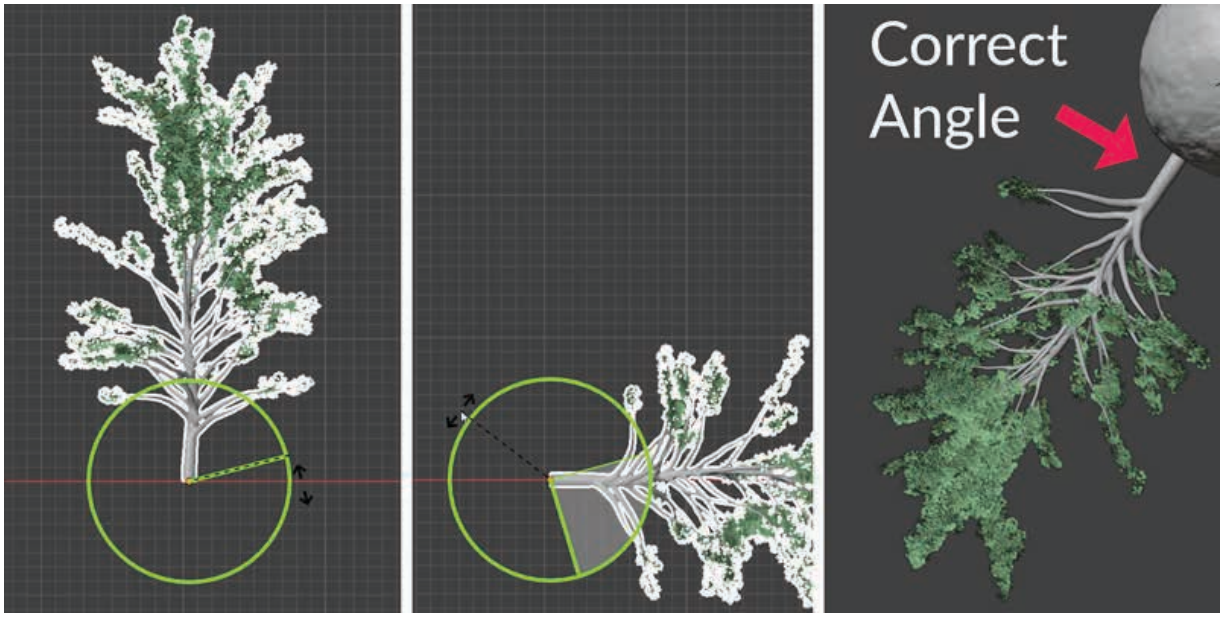
Add a Particle System to it, set it to **Hair** and enable **Advanced**. Set Emission to 1 at first.



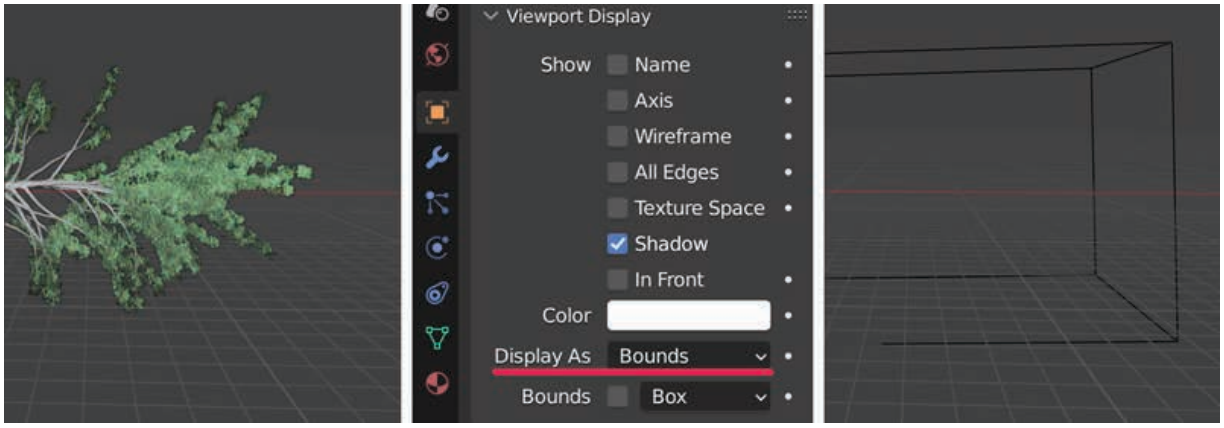
Set Render to **Render as Object** and choose the tree as the **Instance Object**. Check **Object Rotation** as well as **Rotation**.



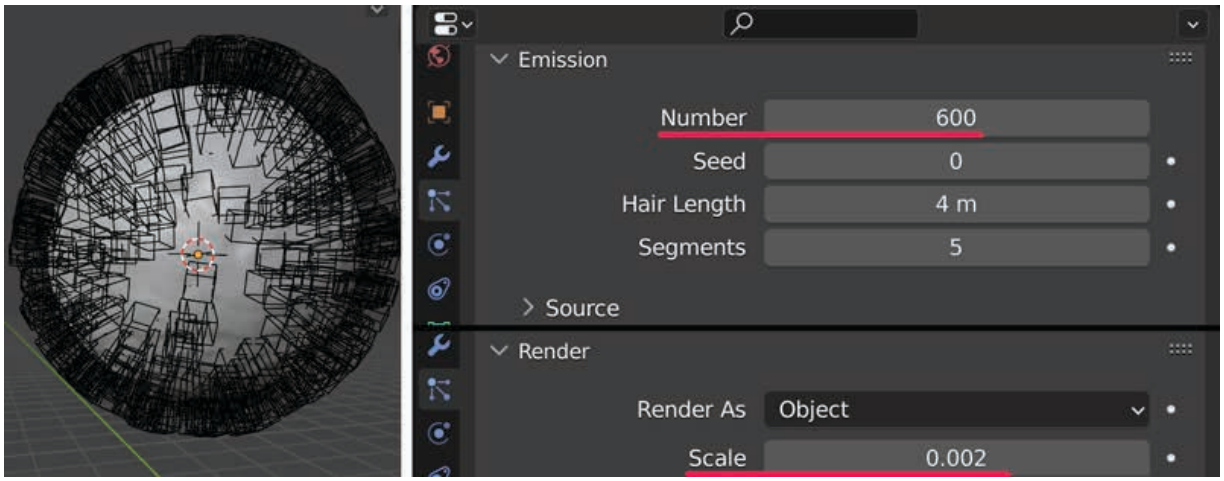
You may need to rotate the original tree, 90 degrees on the Y-Axis to get the correct angle on the emitter.



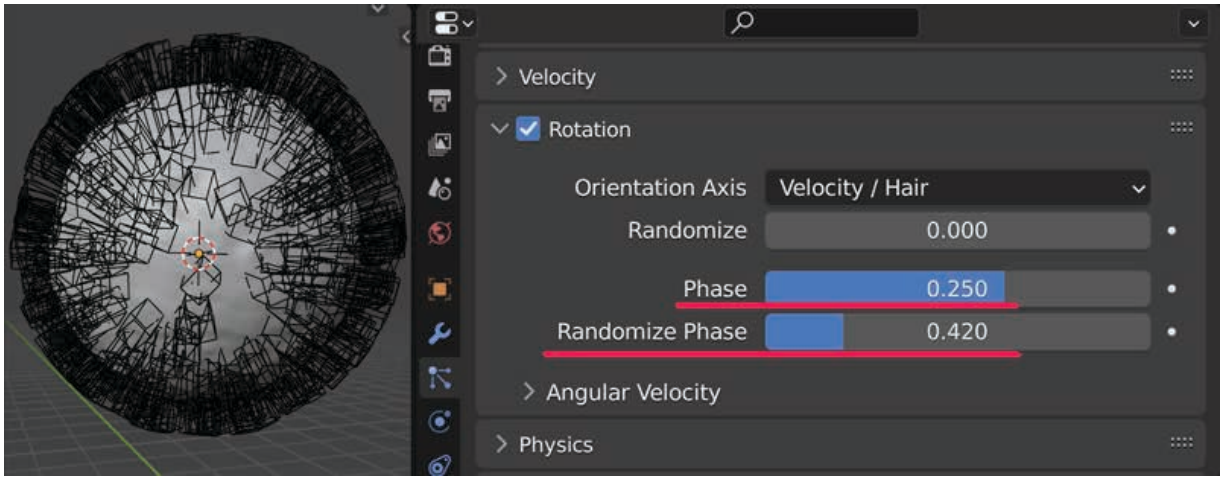
As soon as you see your tree is rotated correctly on the landscape, set viewport display of the original tree to **Bounds**. This will make everything much easier on your computer.



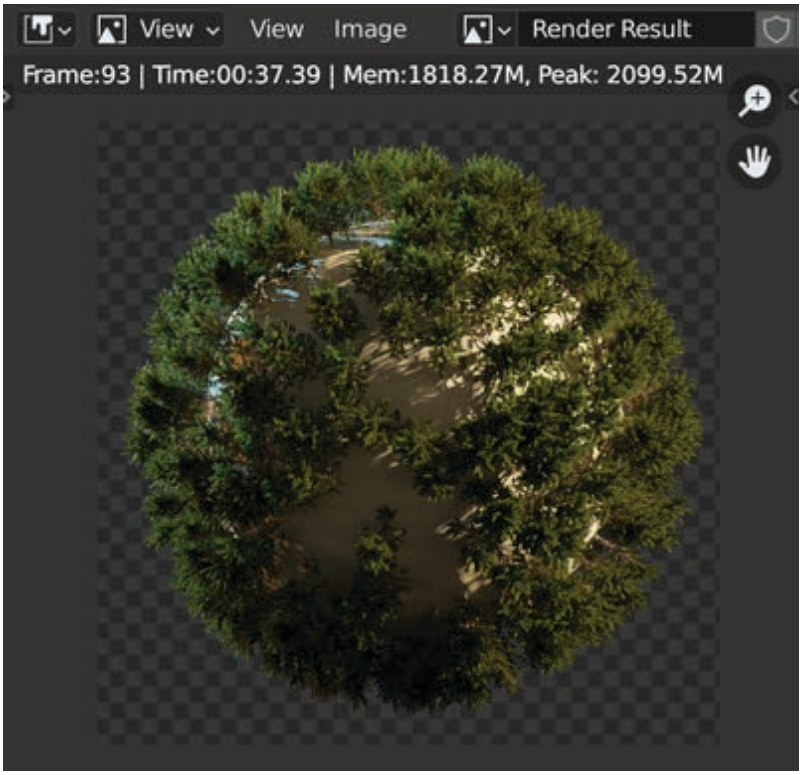
Now increase the **Emission Number**. You can also experiment with the **Scale**.



Increase the **Phase** and **Randomize Phase** values to create some random rotation. Similarly increase the scale randomness.

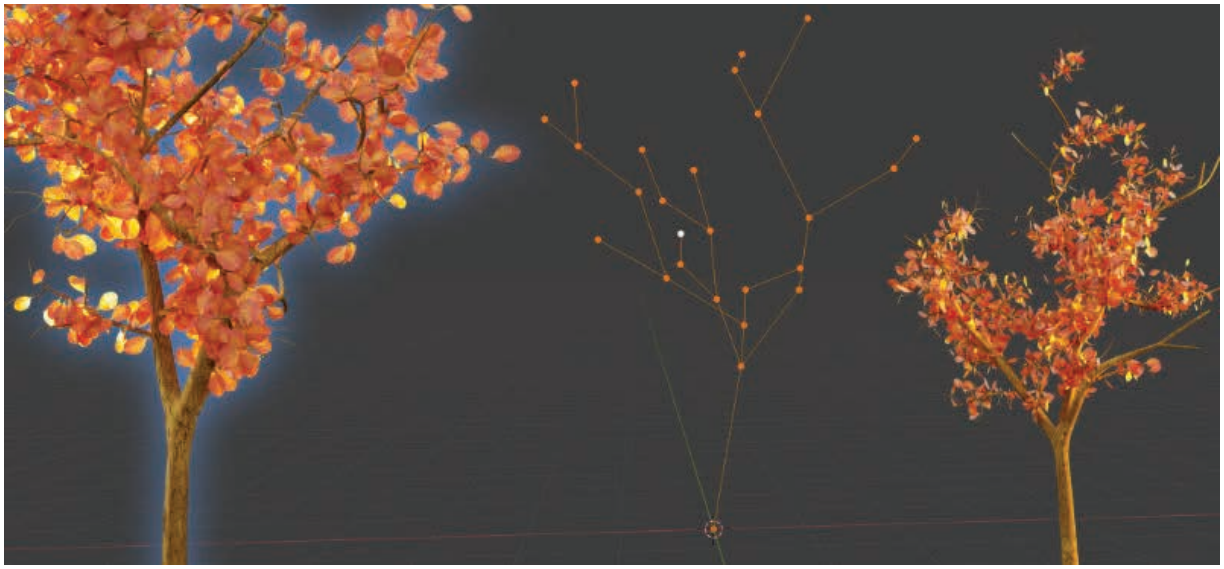


Now if you **press F12** to render, you will get a forest!

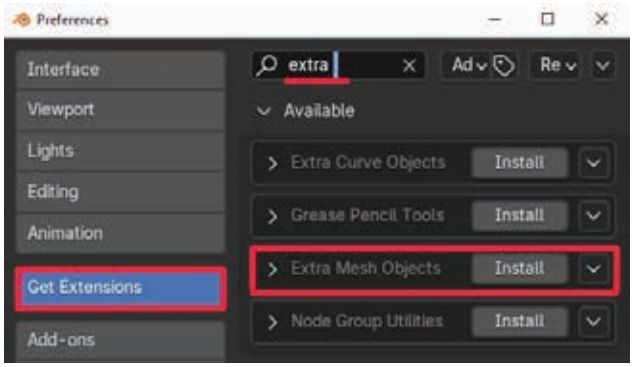


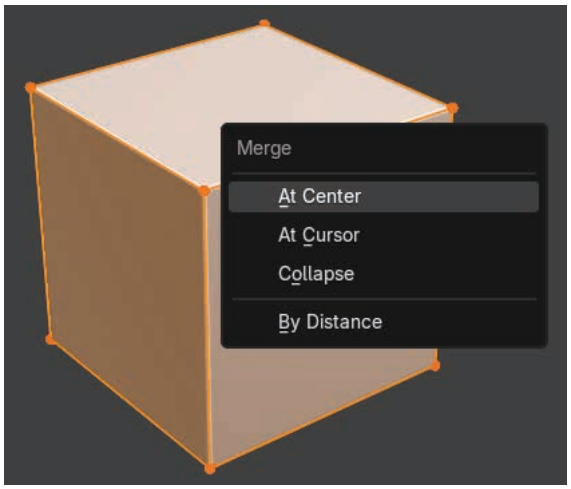
Custom Tree or Plant part 1 - The Trunk

Although you can buy amazing plant and tree libraries online, it's good to know how to make your own. And for architecture visualisation, you may need to use local trees or plants that are not available in any 3D asset store. In the next three tips we'll look at various aspects of creating a plant or trees.

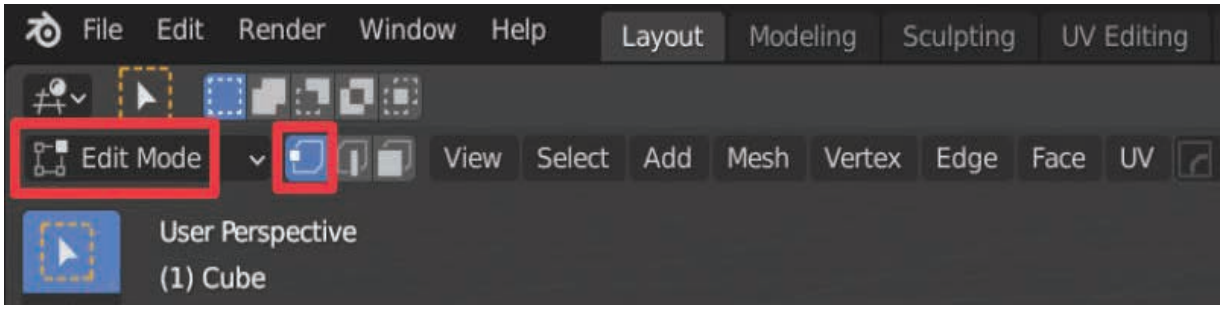


In Edit Mode **press M** and merge the default cube to one vertex. Alternatively, you could activate the **Extra Mesh Objects** add-on, which will add a "Single Vert" option to the Mesh Primitives.

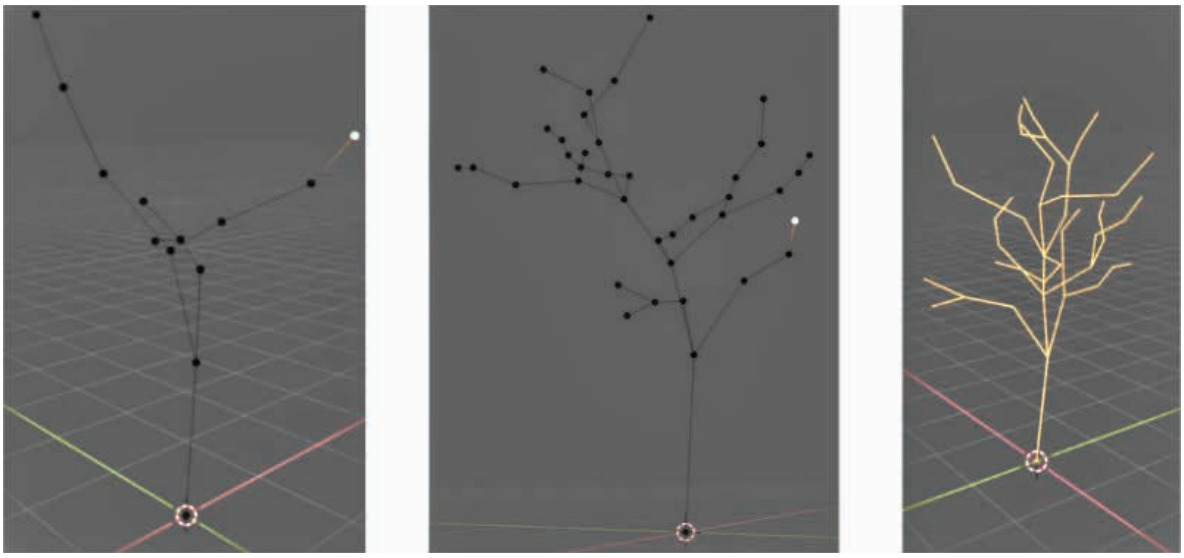




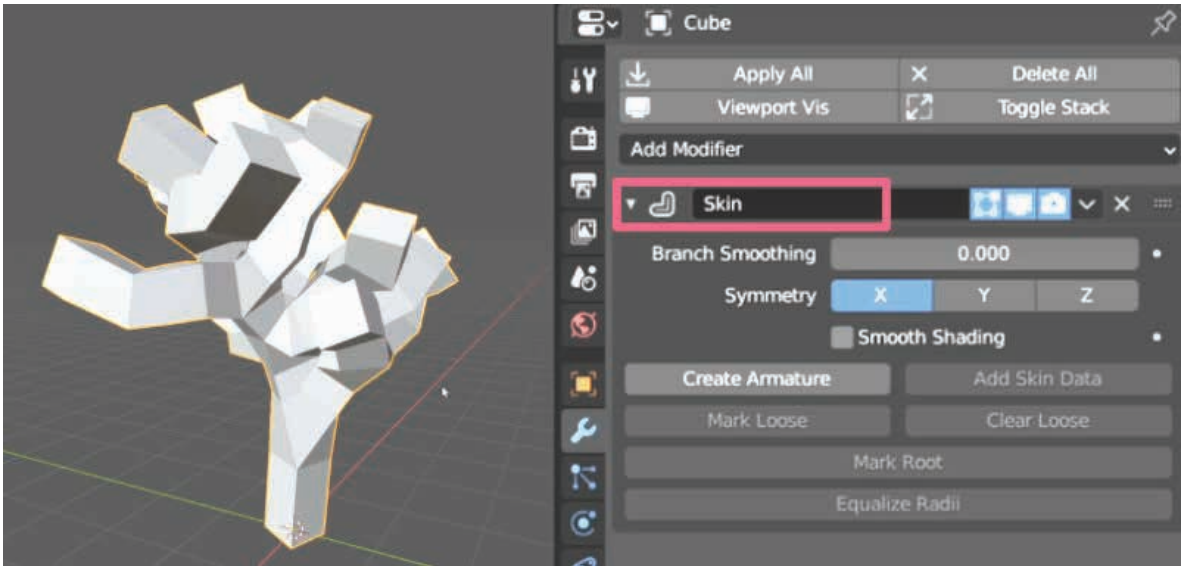
Make sure you're in **Vertex Select Mode** then **press A** so the vertex is definitely selected.



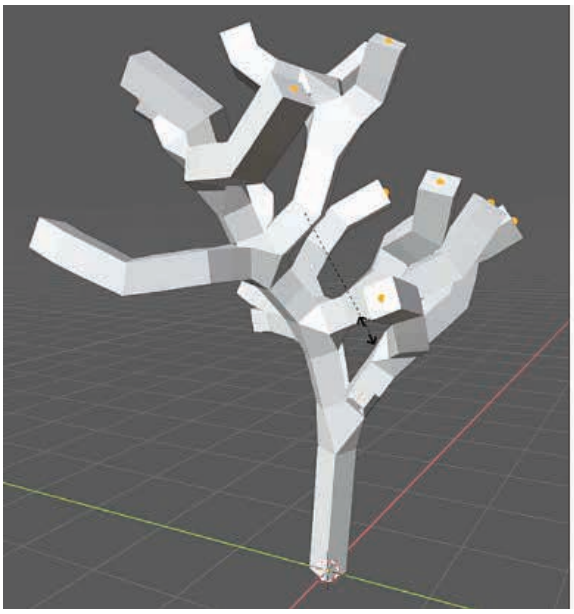
Press E to extrude it and create some branches. Rotate the view occasionally while you do this.

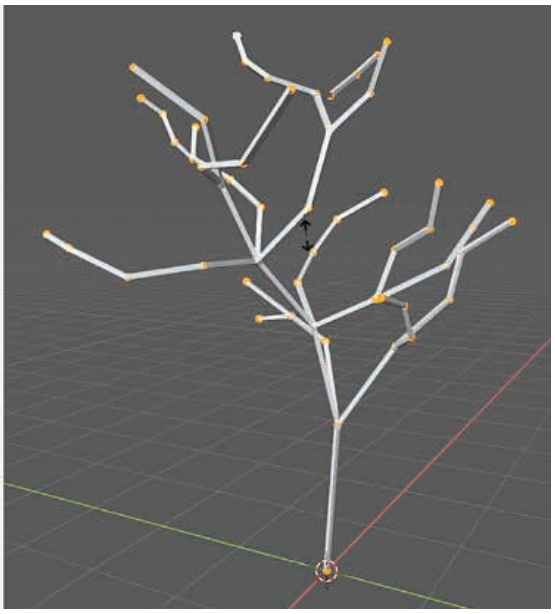


In **Object Mode** add a **Skin modifier**. The tree will probably look weird at first. That's fine.

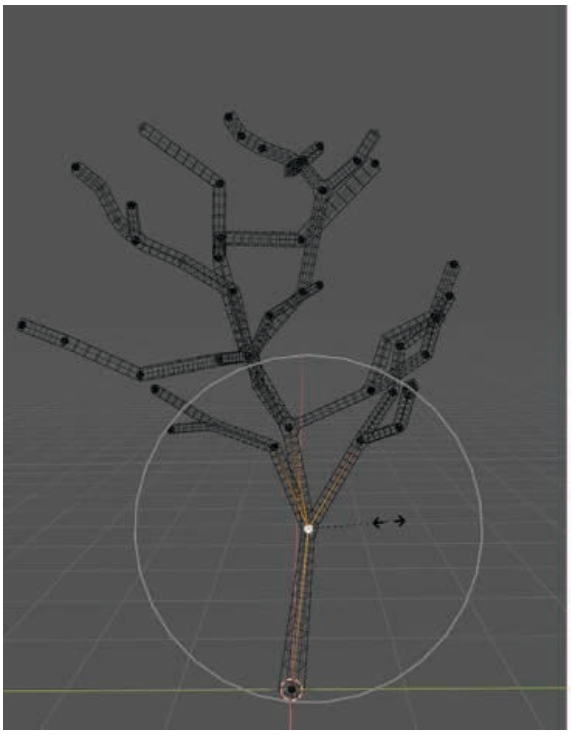


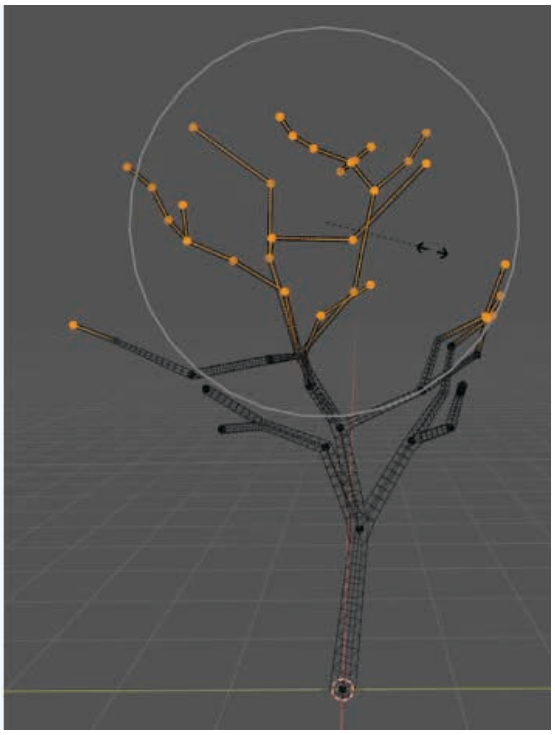
Go back to **Edit Mode** and **press A** to select all vertices, then **press Ctrl+A** and drag the mouse. This changes the **weight** of the vertices, making the trunk more thin overall.



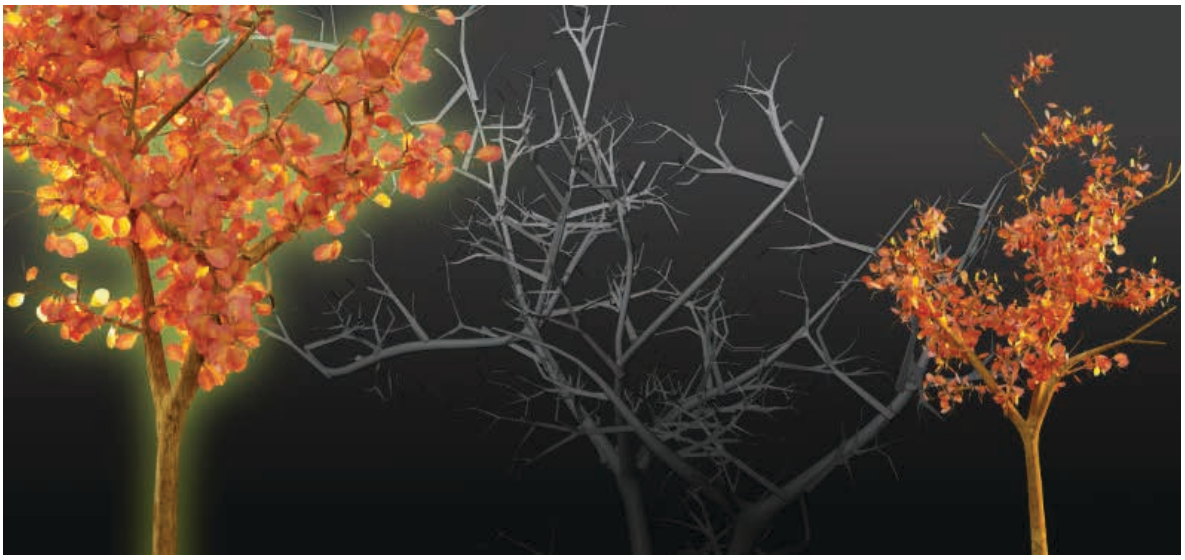


You can also do this for individual vertices and use Proportional Editing (**keyboard shortcut: o**). It's easier to select individual vertices in Wireframe mode. Make sure the base of the trunk is thicker and the ends of the branches are thinner.

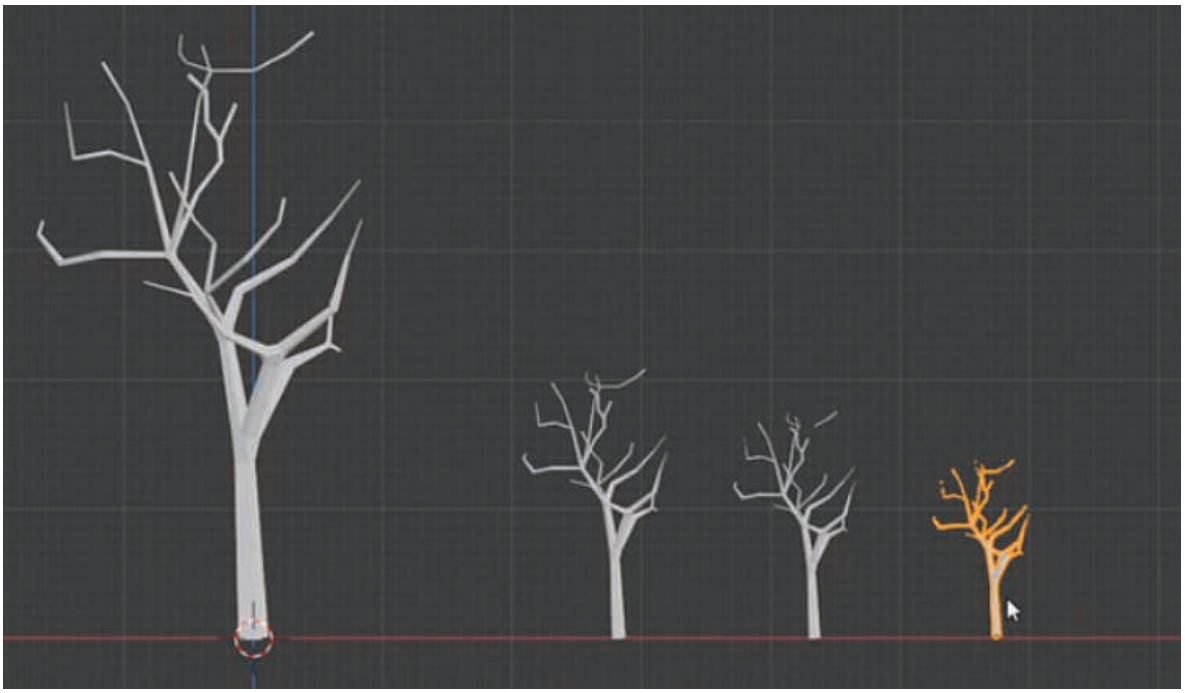




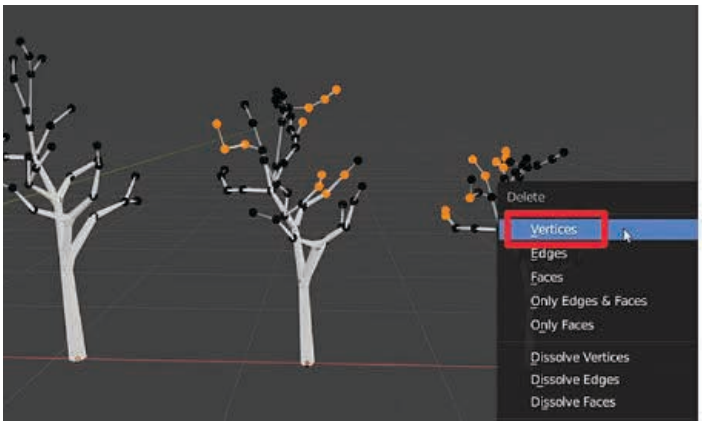
Custom Tree or Plant part 2 - The Twigs

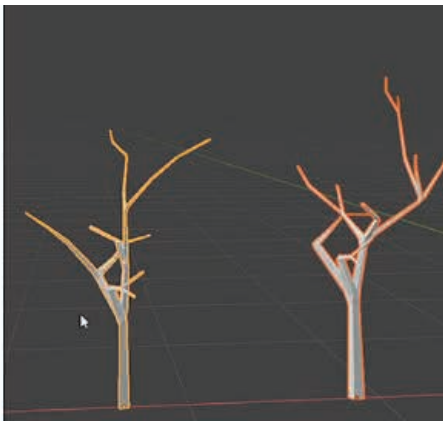


Duplicate and scale the trunk a few times to create some twigs by **pressing Shift+D** to duplicate and **pressing S** to scale.



In **Edit Mode** remove some vertices and extrude new ones to make the twigs unique.



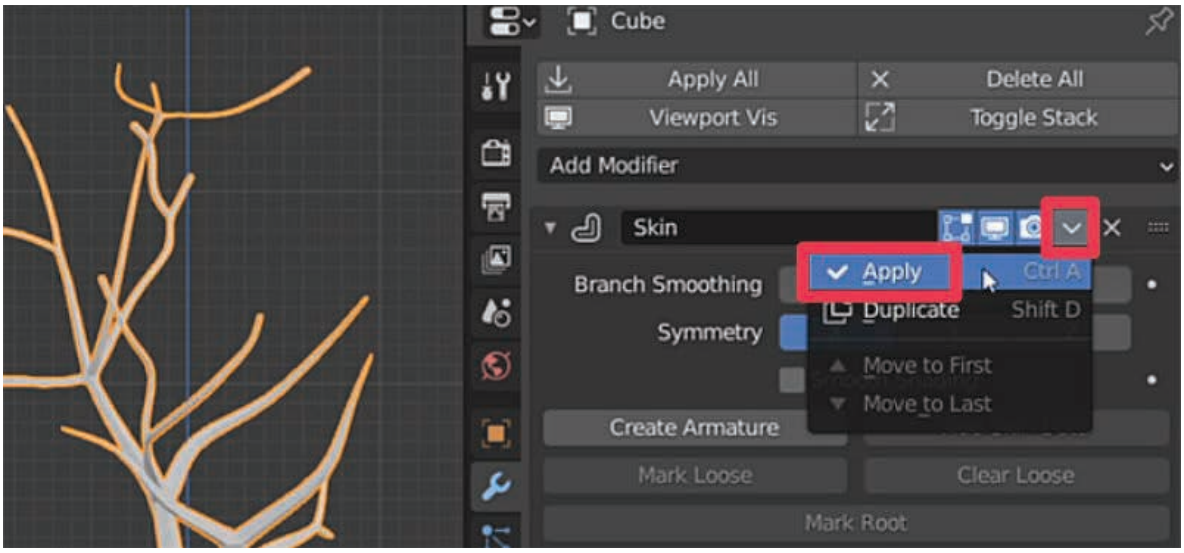


Select the original trunk in Edit Mode, **press A** then **right-click and choose Subdivide**. **Press Shift+R** once to repeat.

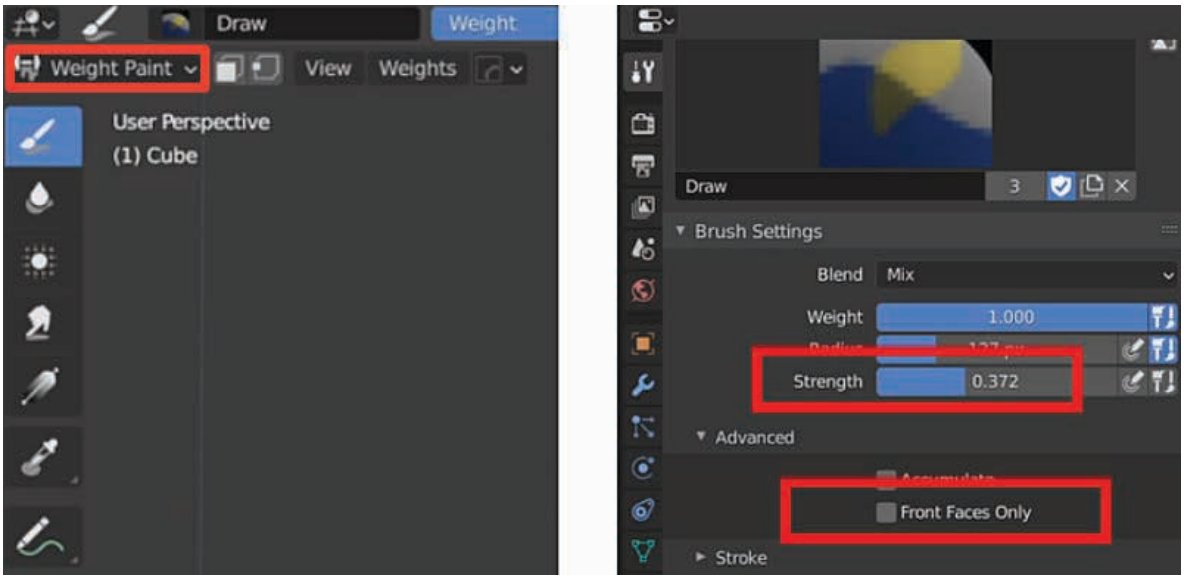
Right-click again and choose **Smooth Vertices**. **Press Shift+R** a few times to repeat. This makes the tree branches a bit more smooth and less angular - of course this is also up to your personal preference, you don't have to do this if you prefer the tree to have sharper angles.

You may need to grab the bottom vertex and move it back down to the ground after smoothening the tree (smoothing vertices tends to make an object smaller).

Apply the Skin modifier. You can do this either by hovering the mouse cursor over the modifier and **pressing Ctrl+A**, or clicking on the drop-down menu icon next to the modifier and choosing “Apply”. This turns the modifier into actual mesh geometry, which is necessary for weight painting.

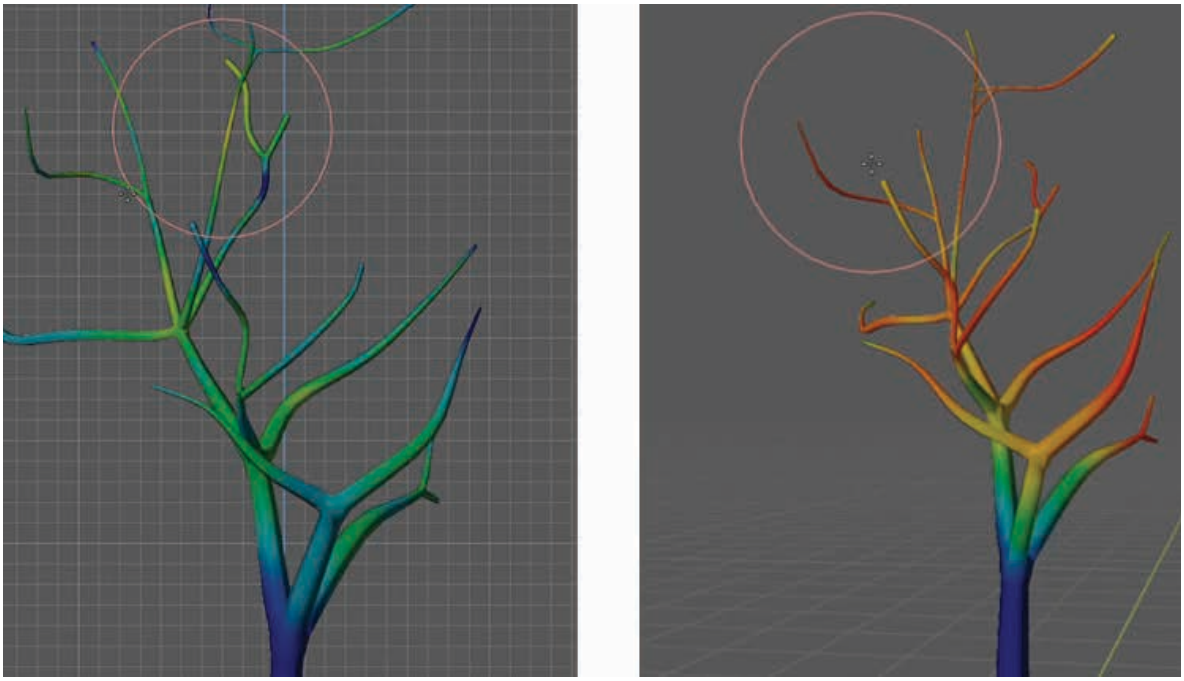


Go to **Weight Paint mode**, lower the **Strength** and uncheck **Front Faces Only**.

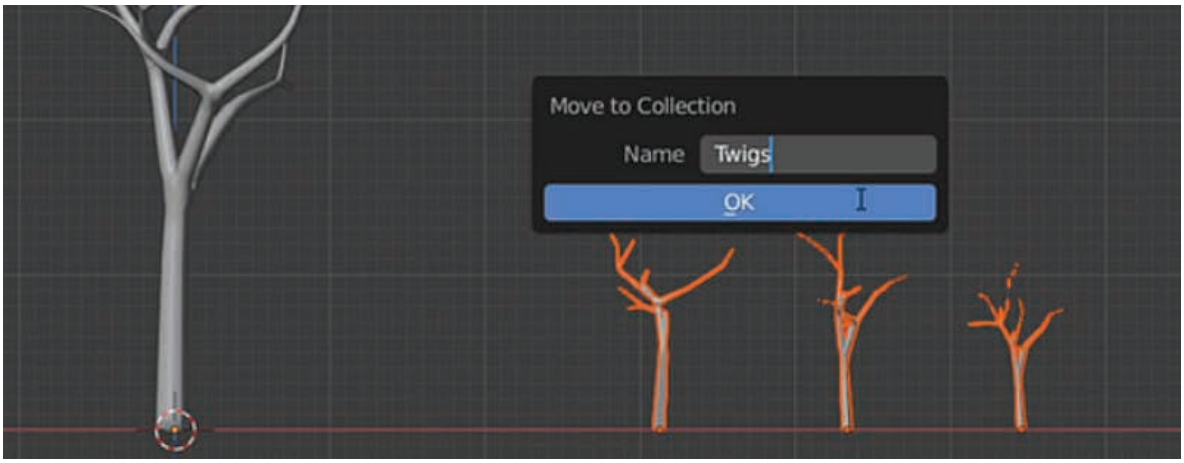


In **Weight Paint Mode**, paint the trunk to indicate where the twigs should be. The more "warm" (orange, red) the color the more twigs there will be. Blue means that there will be no twigs, such as on the base of the trunk for example. You can actually still fine-tune this later after adding a particle system.

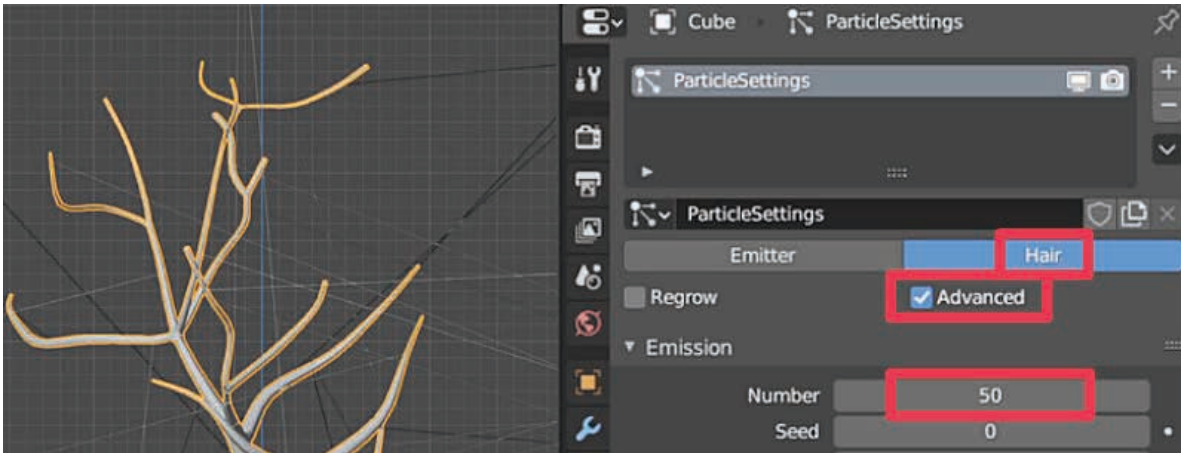
***Tip**: If you can't weight paint because the tree has no thickness as soon as you enter Weight Paint Mode, that's probably because you forgot to "apply" the Skin modifier. So make sure you do that first.*



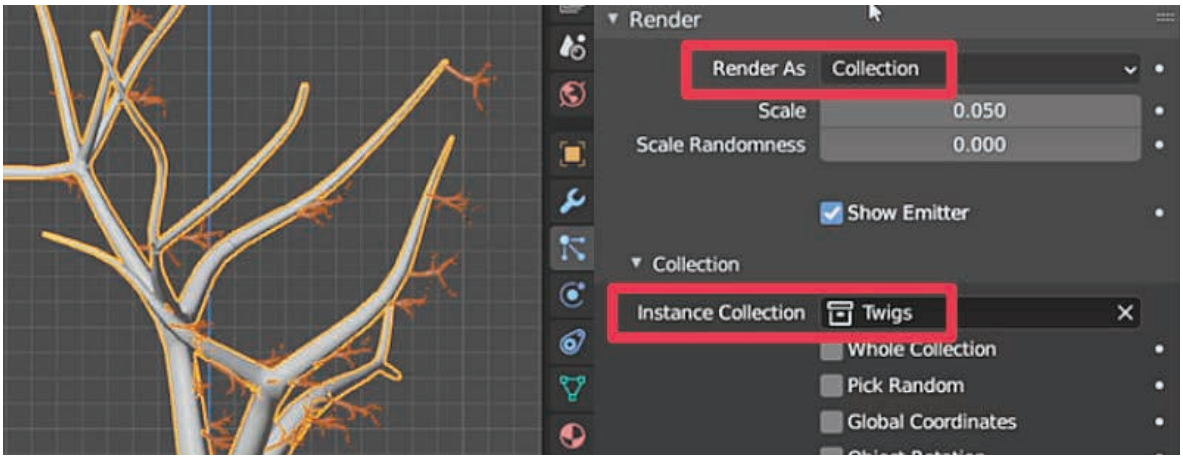
Select the twigs **in Object Mode**, **press M** and add them to a new **Collection**. Give them a clear name like “Twigs” so you know what the collection contains later.



Create an **Advanced Hair Particle System** for the trunk and lower the particle count to something like 50 (you can still change this later to suit your needs).

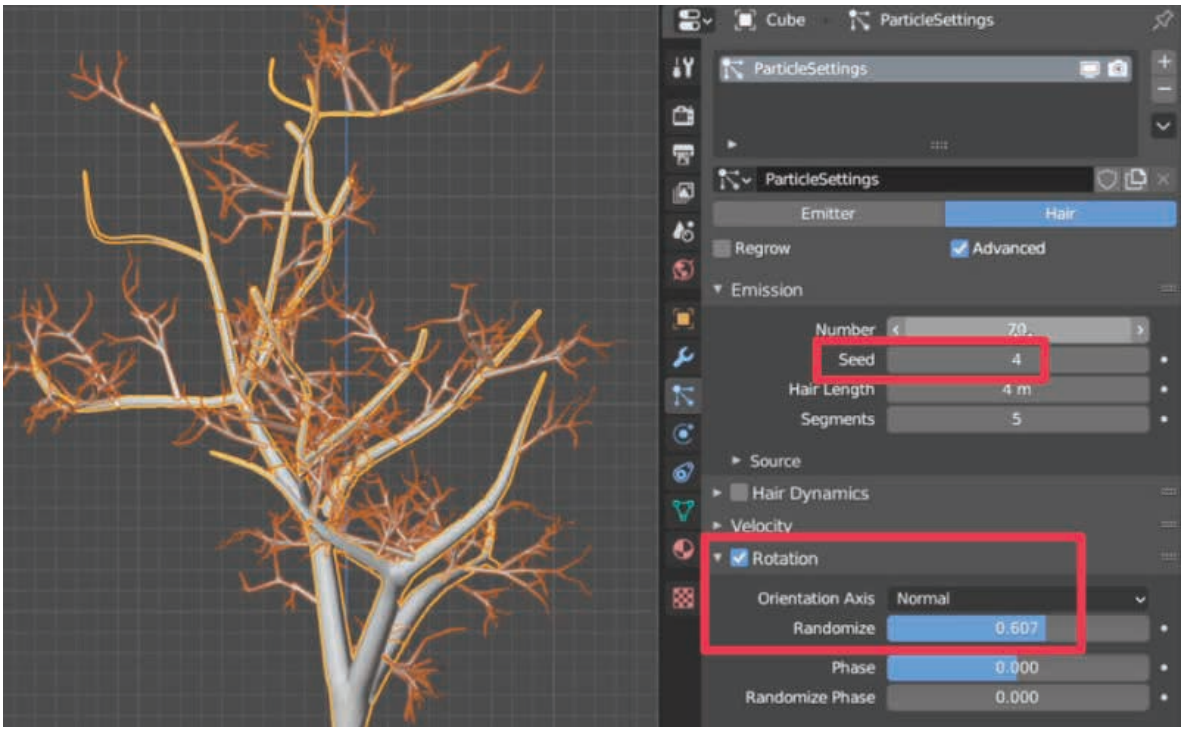


To Render the particles choose the twigs collection.

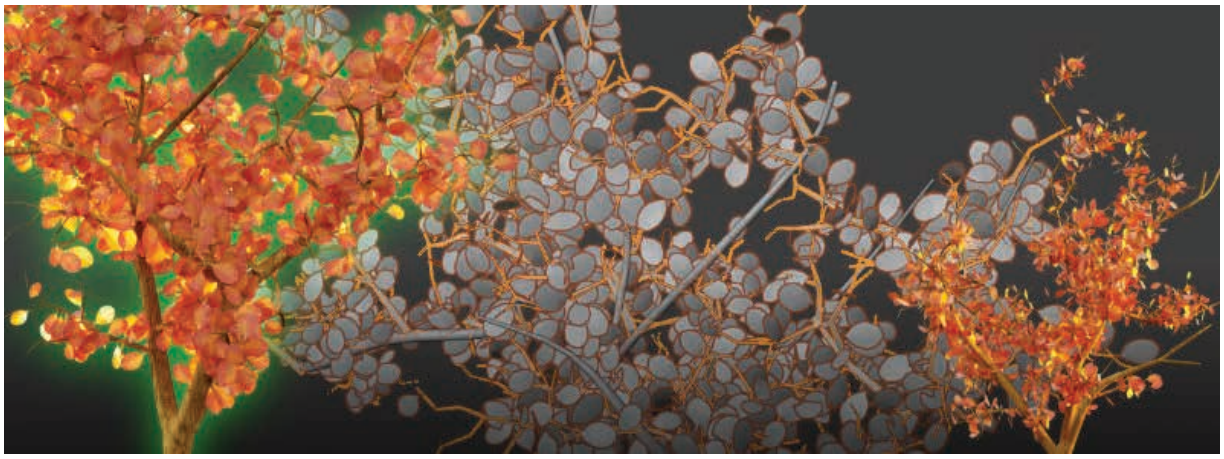


Under **Vertex Group Density** choose the weight painted group. This makes sure the particles (the twigs) are only shown in the areas that were weight painted to be orange or red.

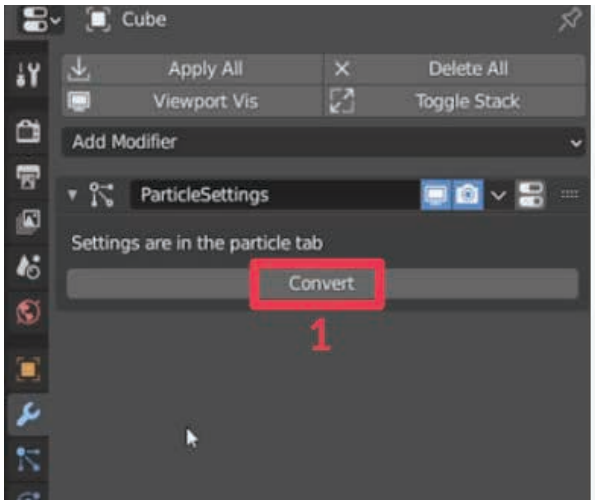
Randomize the rotation and set the Orientation Axis to **Normal**. Change the Emission **Seed** value until you get something that you like. You can also increase the Number of particles if you feel the tree needs it. And you can enter Weight Paint Mode and change the weight painting here and there - the tree will update in real-time so you can see what you're doing.

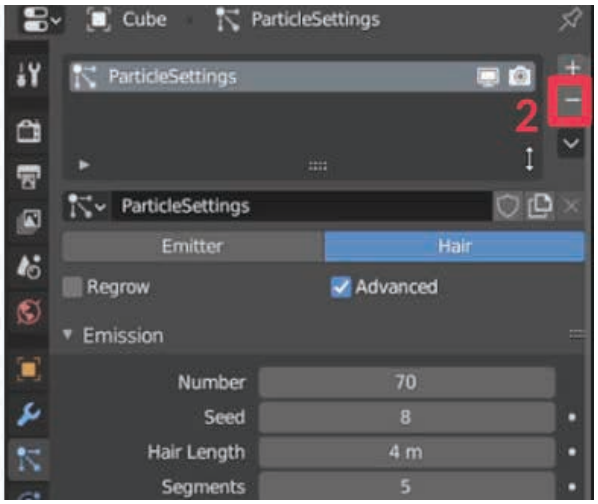


Custom Tree or Plant part 3 - The Leaves

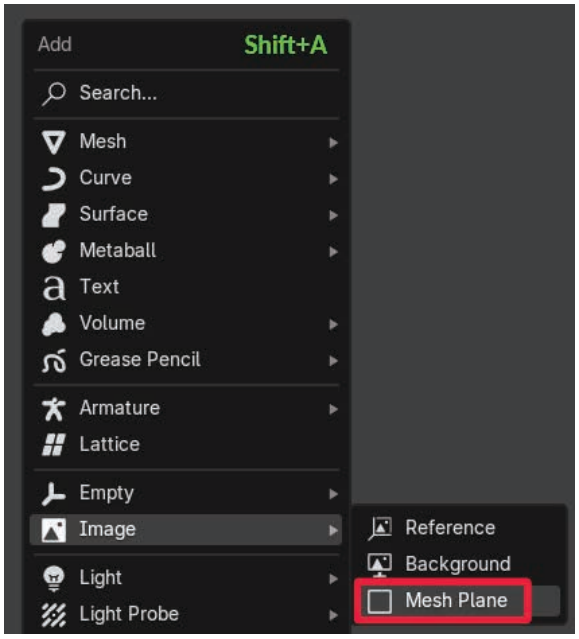


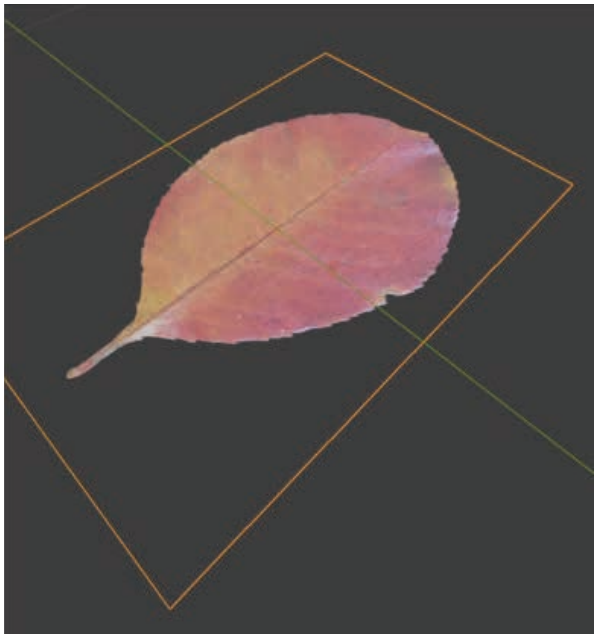
Convert the Particle System and remove it by **clicking on the [-] icon**.





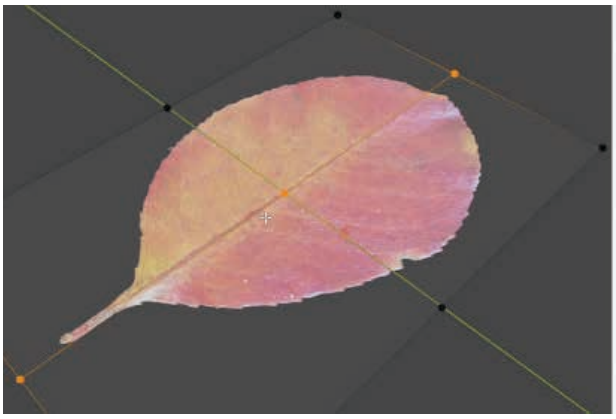
Add leaf image with an alpha channel using Add > Image > **Mesh Plane**. You can find many free leaf textures on websites like Textures.com.

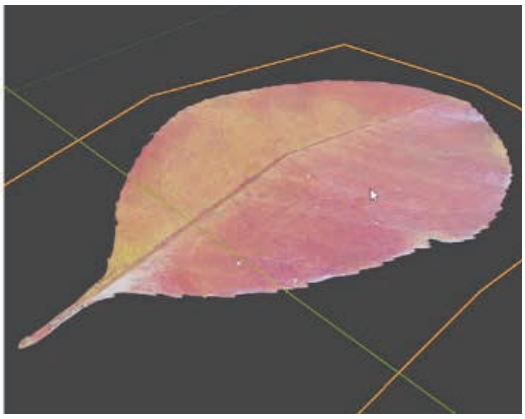




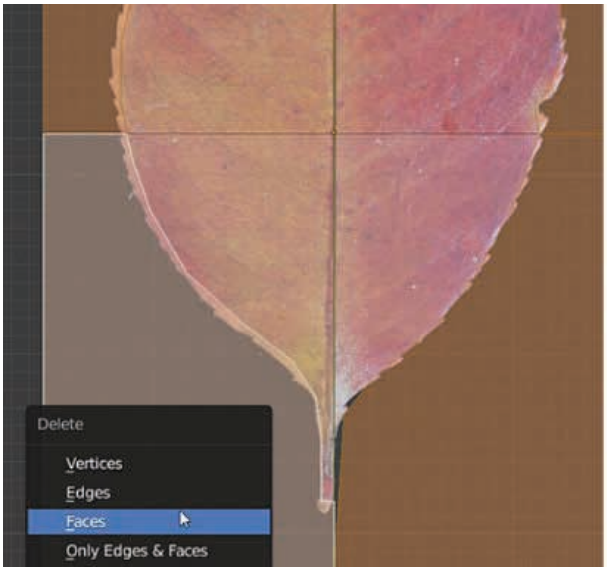
If you can't see the texture, make sure you are in **Material view** (or Rendered view).

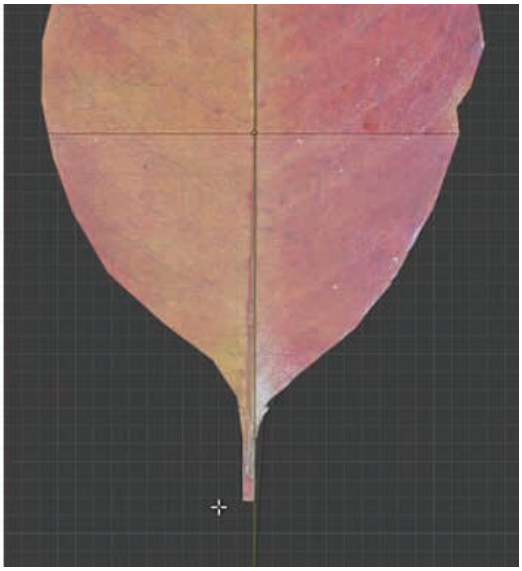
You can add some loop cuts and give the leaf a minimal amount of bending.



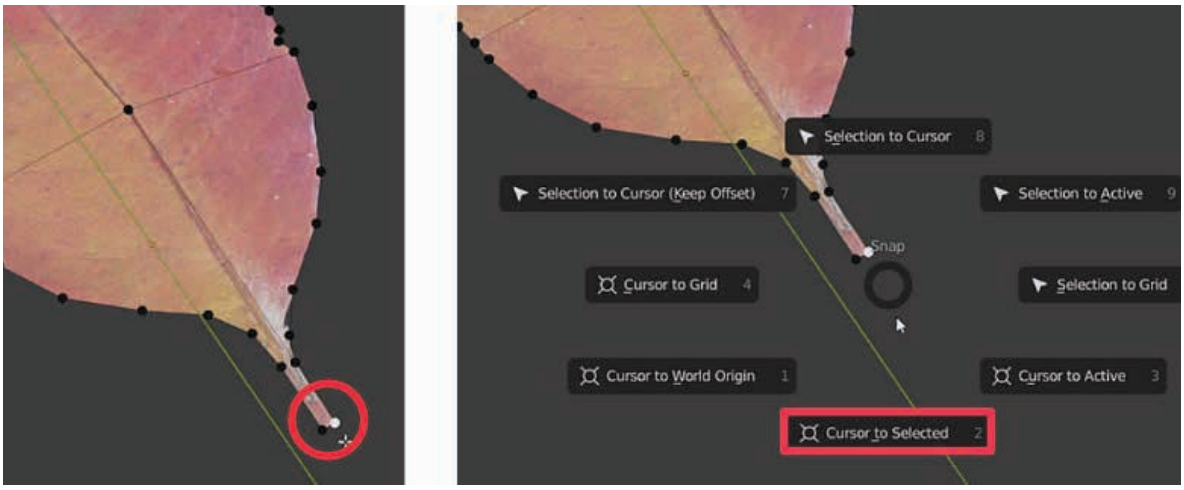


If you don't want to use transparency you can **press K** for the knife tool and cut out the leaf shape.

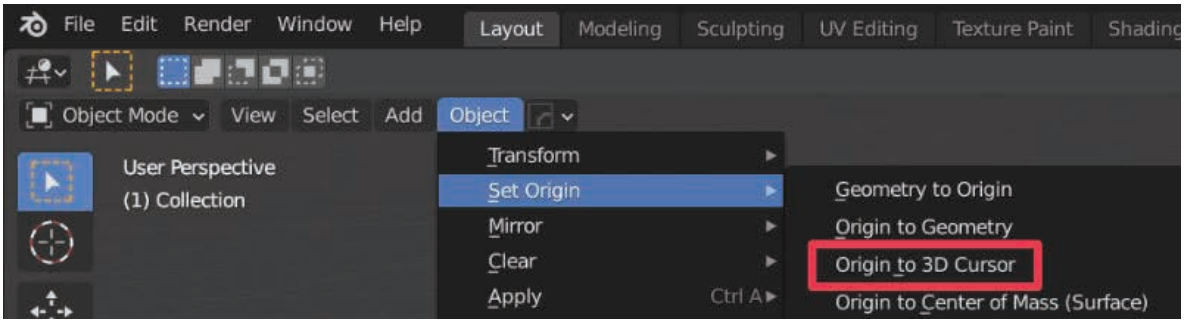




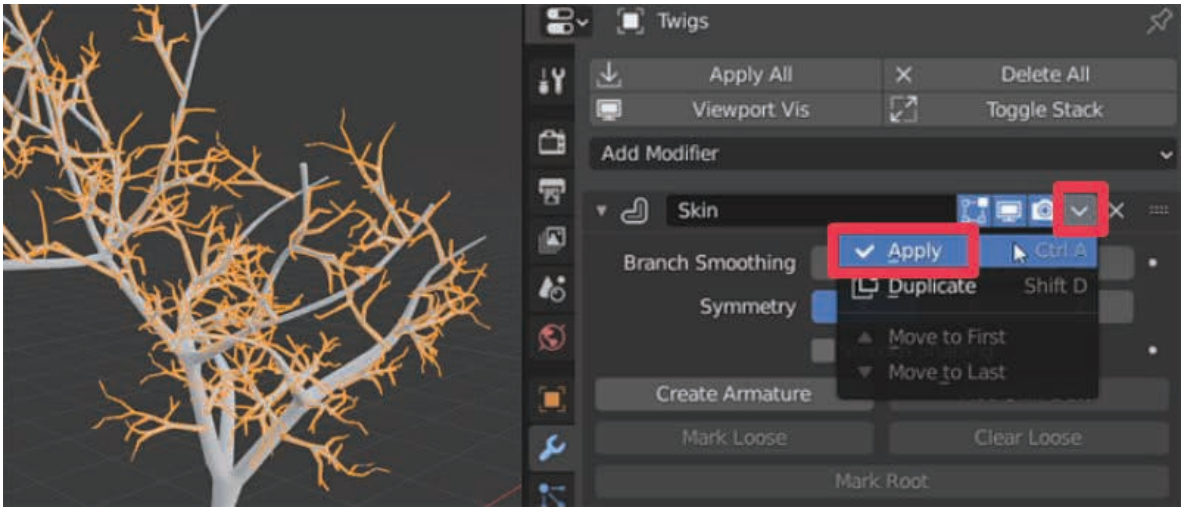
Select the vertex at the base of the leaf and **press Shift+S, choose Cursor to Selected**.



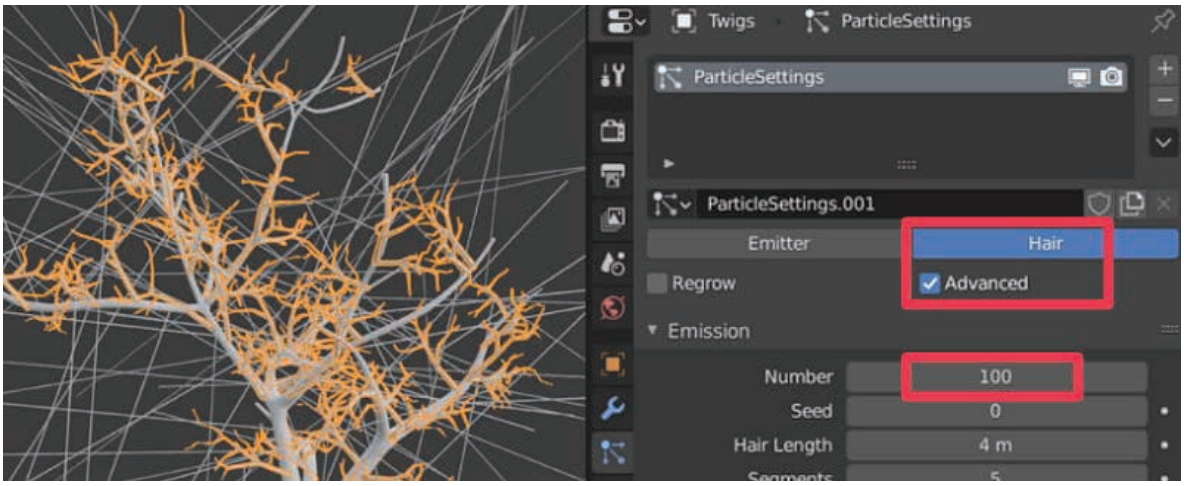
Then in Object Mode go to Object > Origin > **Origin to 3D Cursor**.



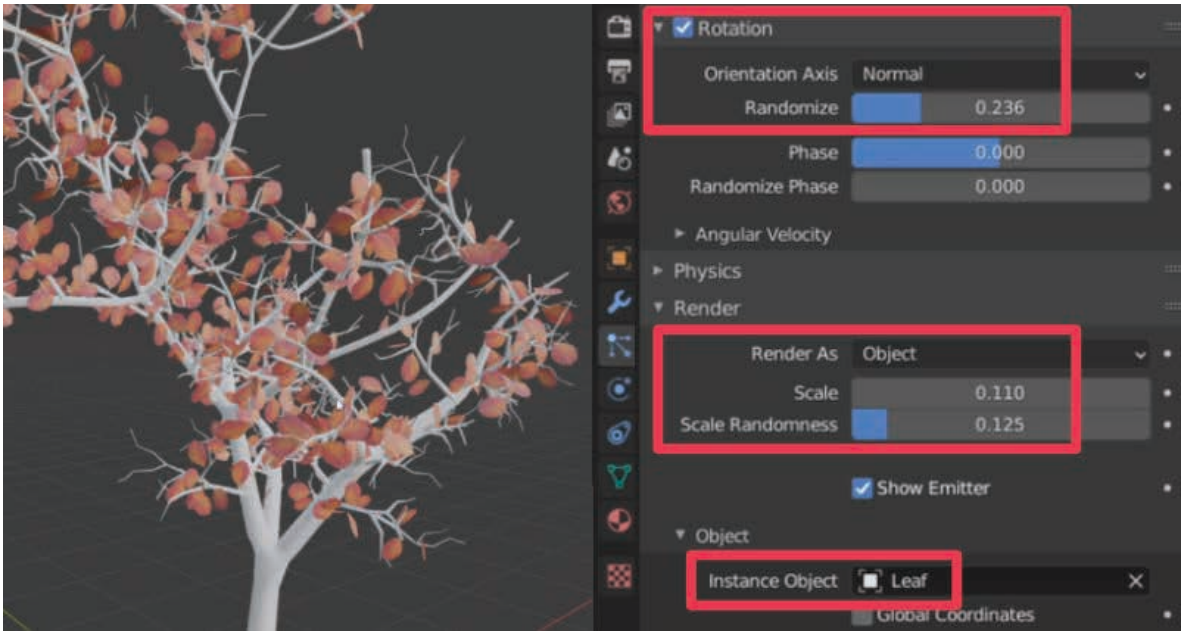
Select all the twigs and **press Ctrl+J** to join them. Then apply the Skin Modifier either by hovering the mouse cursor over the modifier and **pressing Ctrl+A**, or clicking on the drop-down menu icon next to the modifier and choosing "**Apply**".



Add an **Advanced Hair particle system** to the twigs. About 100 particles should do (you can still change the amount later if needed).



Use the leaf object to render the particle and randomize the rotation.



That's about it! Of course the trunk needs to have a material too. For that, I used "box mapping" since the trunk doesn't have a UV map (and it would be a pain to UV-unwrap it manually).

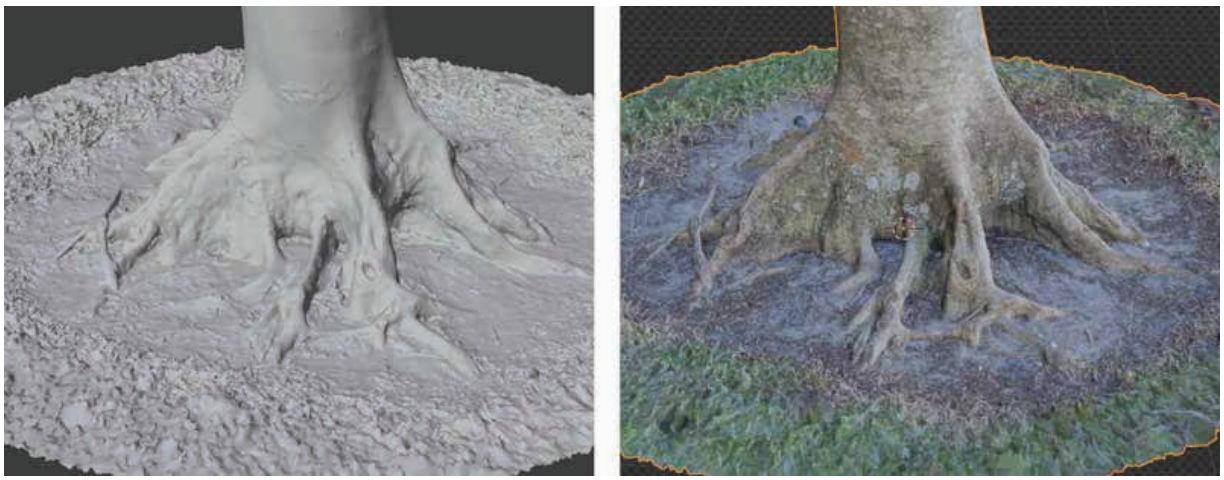
If you want to learn how to do **box mapping** like this I suggest you read the topic "Texturing Without UV-Unwrapping". The leaves were made somewhat translucent as explained in "How to use Leaf textures", and the whole thing was lit using the Sky Texture, which is explained in "Sunset Timelapse with Sky Texture".

Mixing 3D scanned and node-based Trees

Or how to get the best of both worlds: mixing the organic realism of photogrammetry with the benefits of using node-based trees.

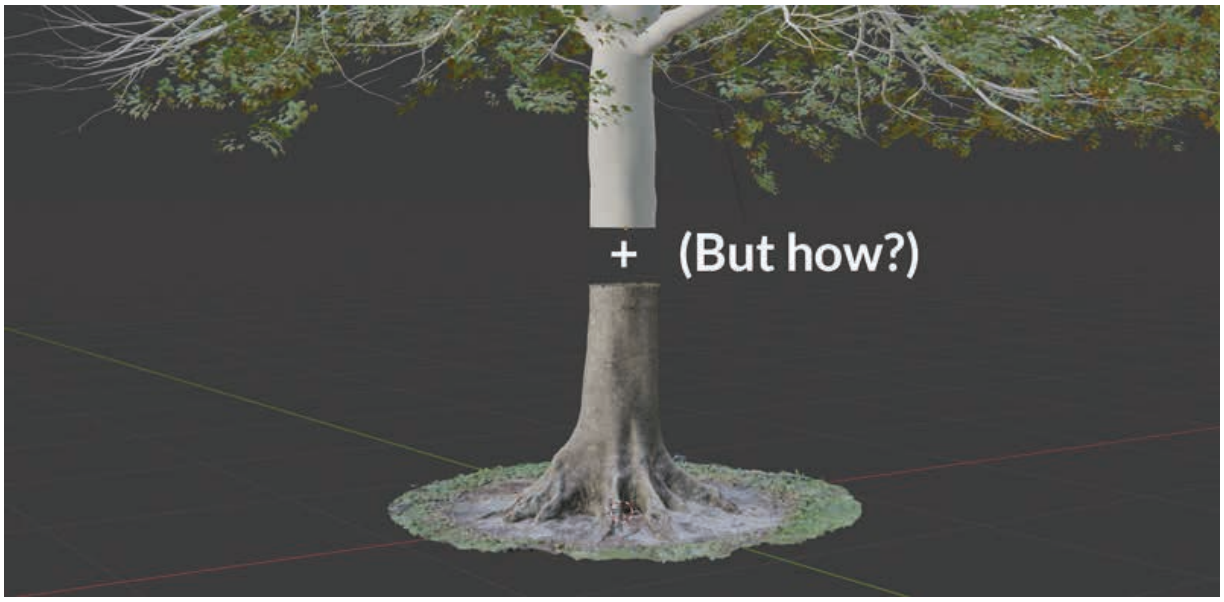
If you've ever tried 3D scan a tree using photogrammetry, you'll know that it's pretty easy to scan the roots and base of the trunk. All you need is a bunch of photographs taken while walking around the tree and software like Meshroom or Reality Capture.

A fellow Youtuber was kind enough to go out and take some pictures of this interesting looking tree for me, which I ran through Reality Capture to get a 3D model.



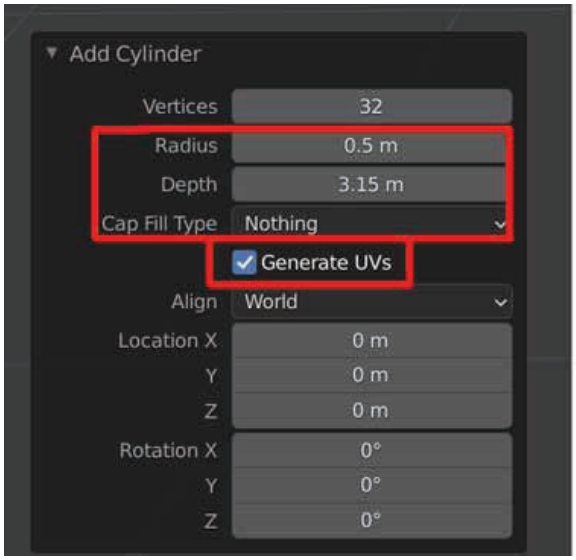
Pretty nice! However the top part of the tree is too difficult to capture with this method. So today we're going to look at using photogrammetry for the bottom of the tree, and a procedural method for the top part of the tree.

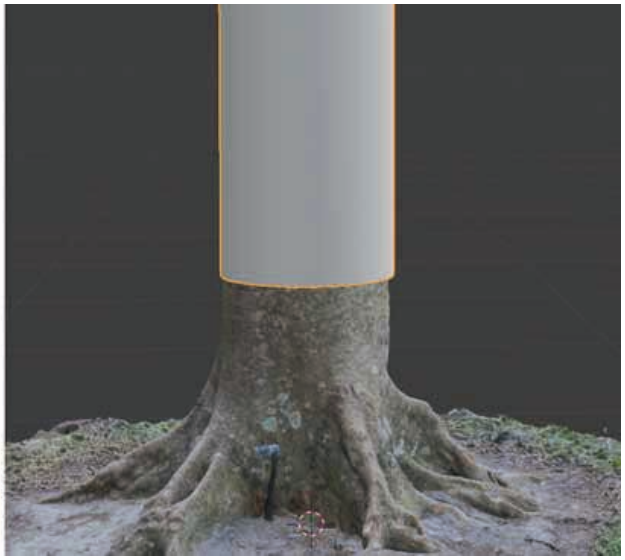
Disclaimer: I would consider this an "advanced" topic, in that I assume you have some fundamental knowledge of Blender. I'm also assuming you know about photogrammetry as well as how to create a tree using either the free MTree add-on or the Sapling add-on that comes with Blender by default.



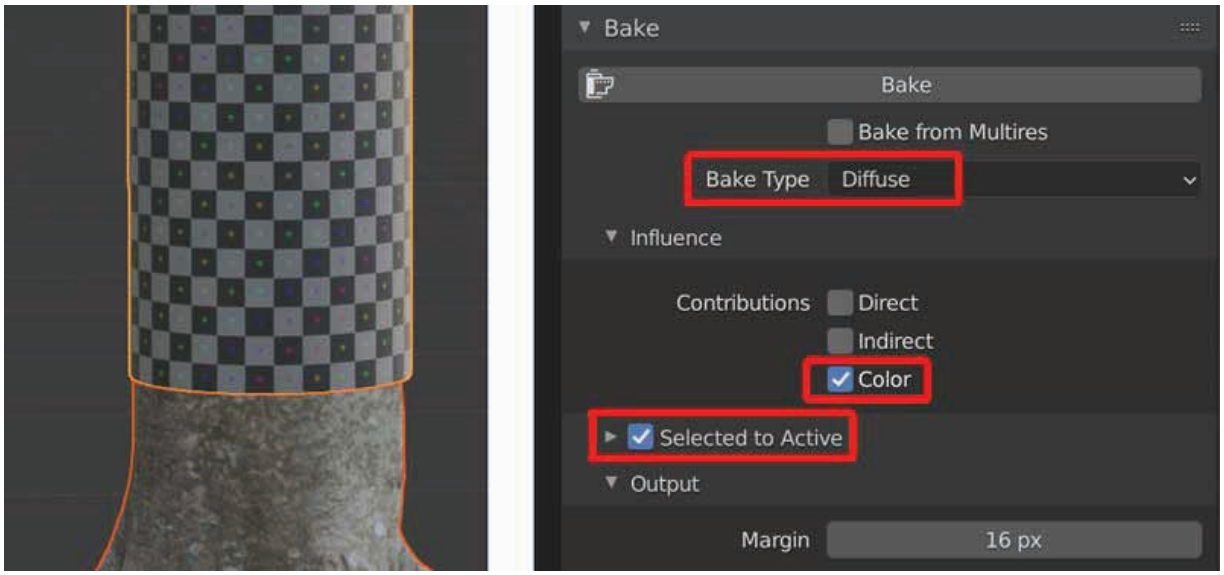
For best results, we need to extract a tileable texture from the 3D scan. **Add a Cylinder** with **Radius 0.5 m** and **Depth 3.15 m**. With those dimensions, we can get a nice square UV unwrap.

Choose **Cap Fill Type: Nothing**. Generate UVs should be checked. **Right-click and choose Shade Smooth** so the texture baking looks good later. Scale and move this cylinder so that it roughly overlaps the 3D scanned trunk.





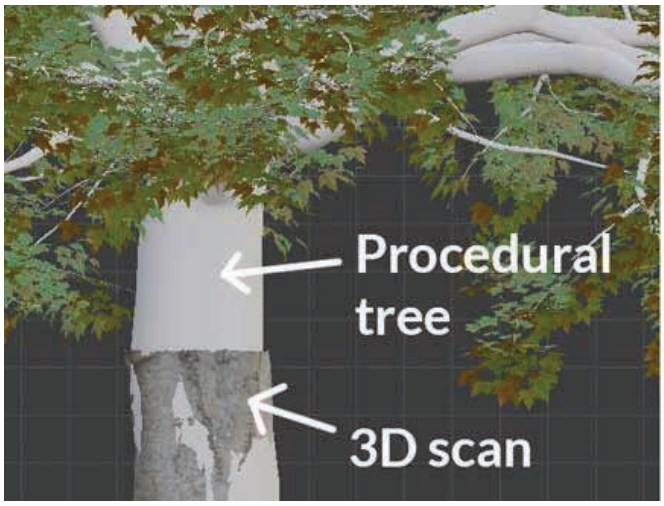
Now we need to bake the 3D scan texture to the new Cylinder. To do that, give the Cylinder a new material and a new texture. Switch to **Cycles** so that you get the Bake options in the Scene tab. As **Bake Type** choose **Diffuse** and set it to **Influence: Color.** Check "**Selected to Active**". Select the 3D scan and then Shift-select the Cylinder so that it is the Active object. **Click on Bake**.

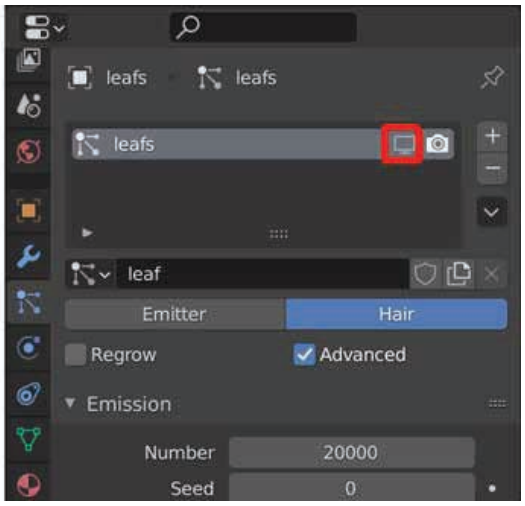


Give it a minute. Once it's done, if you select the Cylinder you'll see that you have a nice square tree bark texture in the UV Editor window.

You could even make this tile seamlessly in Gimp or with the free Anti-Seam addon in Blender if you wanted to, but in a quick Box Mapping test, the difference between seamless and not seamless wasn't noticeable.

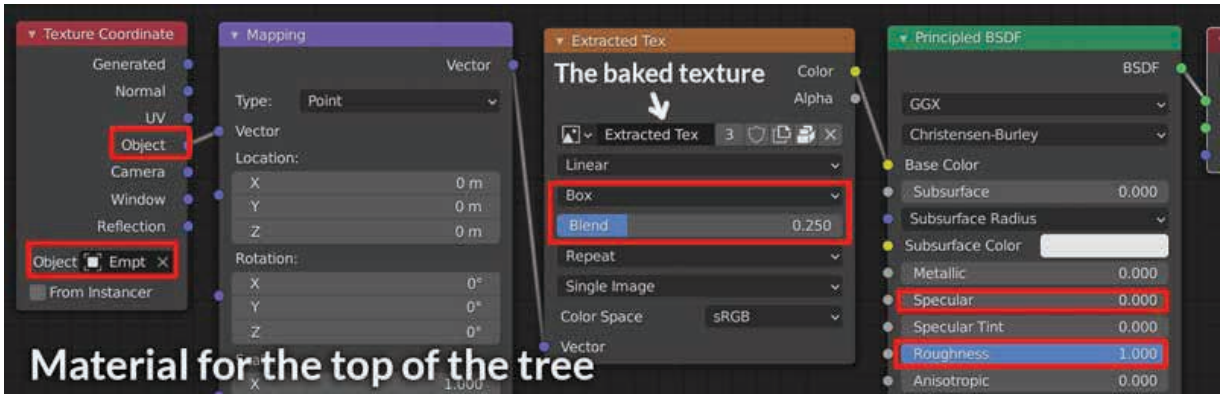
For the top part of the tree, I used the free **MTree add-on** with the default "Old Oak" preset and "Palmate" as the leaf type. I didn't use any of its preset materials for the bark. To make working on the tree a bit easier in the Viewport, I temporarily turned the Particle system for the leaves off.





To use the baked texture for the top of the tree we can use Box Mapping. Create a new material for the top part of the tree and set the Texture Coordinates to **Object.** Then, set projection type to **Box** and increase the **Blend value to 0.25** or so, to remove the visible seams.

Create an Empty and use that Empty as the **Object** in the Texture Coordinates node. Decrease the **Specular** value and increase the **Roughness** so it isn't shiny.



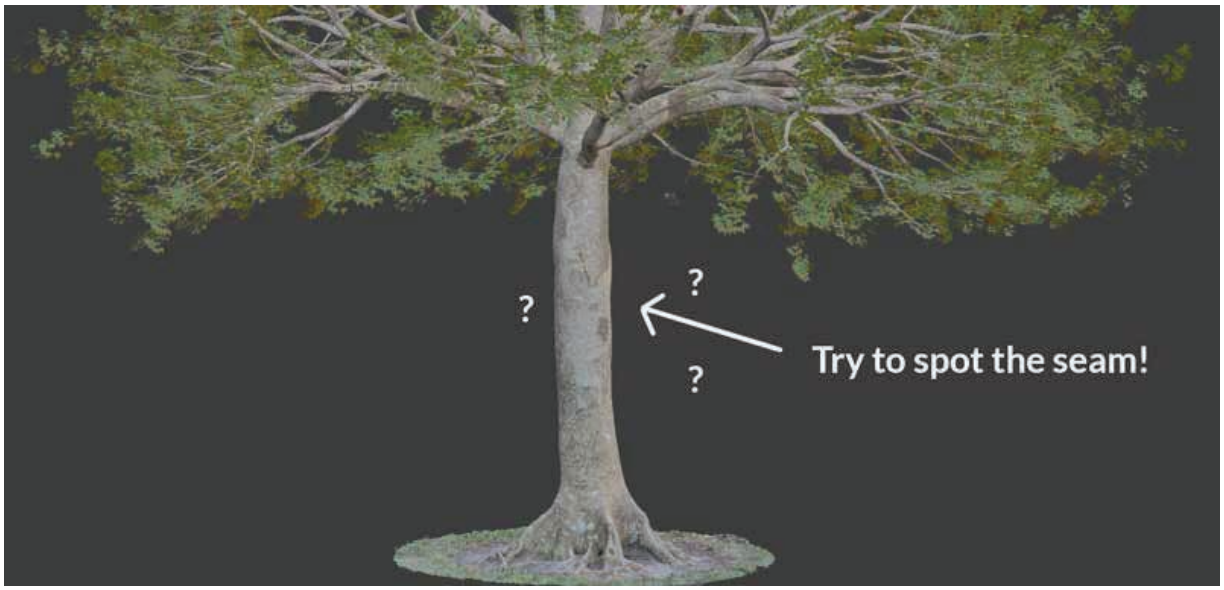
By using Object texture coordinates, we can control the location and scale of the texture mapping with the Empty, and the material will look the same on different objects.

This is essential for the process of visually stitching the top and bottom parts of the tree. You may have to increase or decrease the Scale the Empty until the material looks good and doesn't tile too obviously.

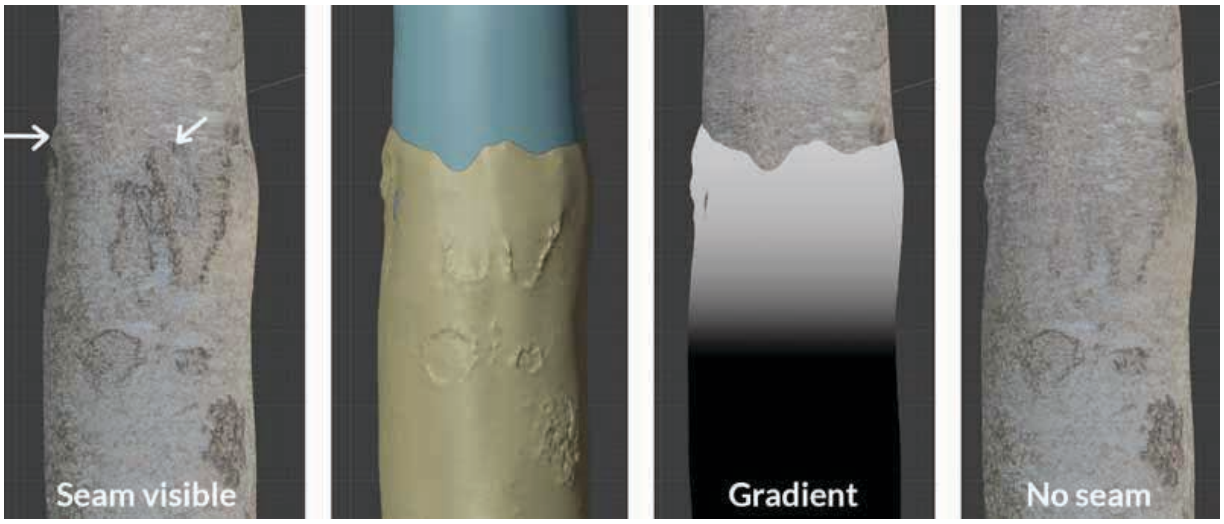
Using Proportional Editing in Edit Mode, move the parts of the overlapping trunks that stick out, into the other trunk so nothing is sticking out.

At this point, it actually already looks pretty good... from a distance! Because of the random, organic nature of the tree bark texture it's hard to see exactly where one tree starts and the other ends.

You can rotate the texture in the Mapping node to see if you can find an even better fit. If you're tired from all the hard work at this point, you have my permission to go take a nap, in the knowledge that you've created a pretty decent looking tree!

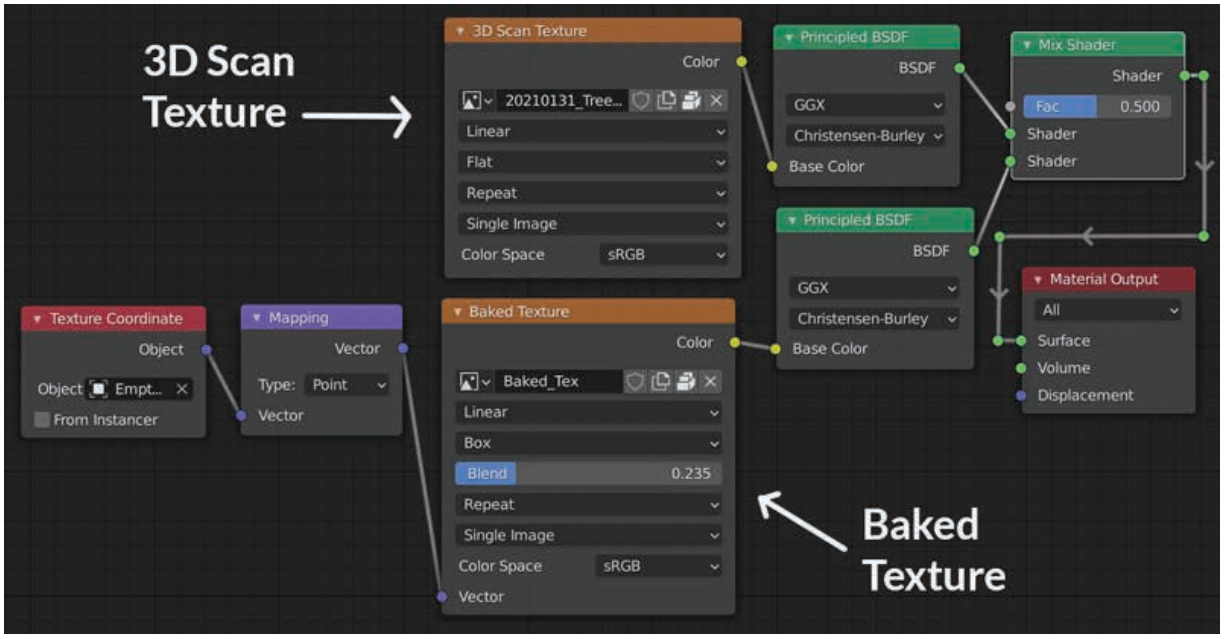


However to make it even better, let's create a smooth gradient from one material to the other on the 3D scan tree model. That way there is really no visible seam at all between the two meshes, even up close.



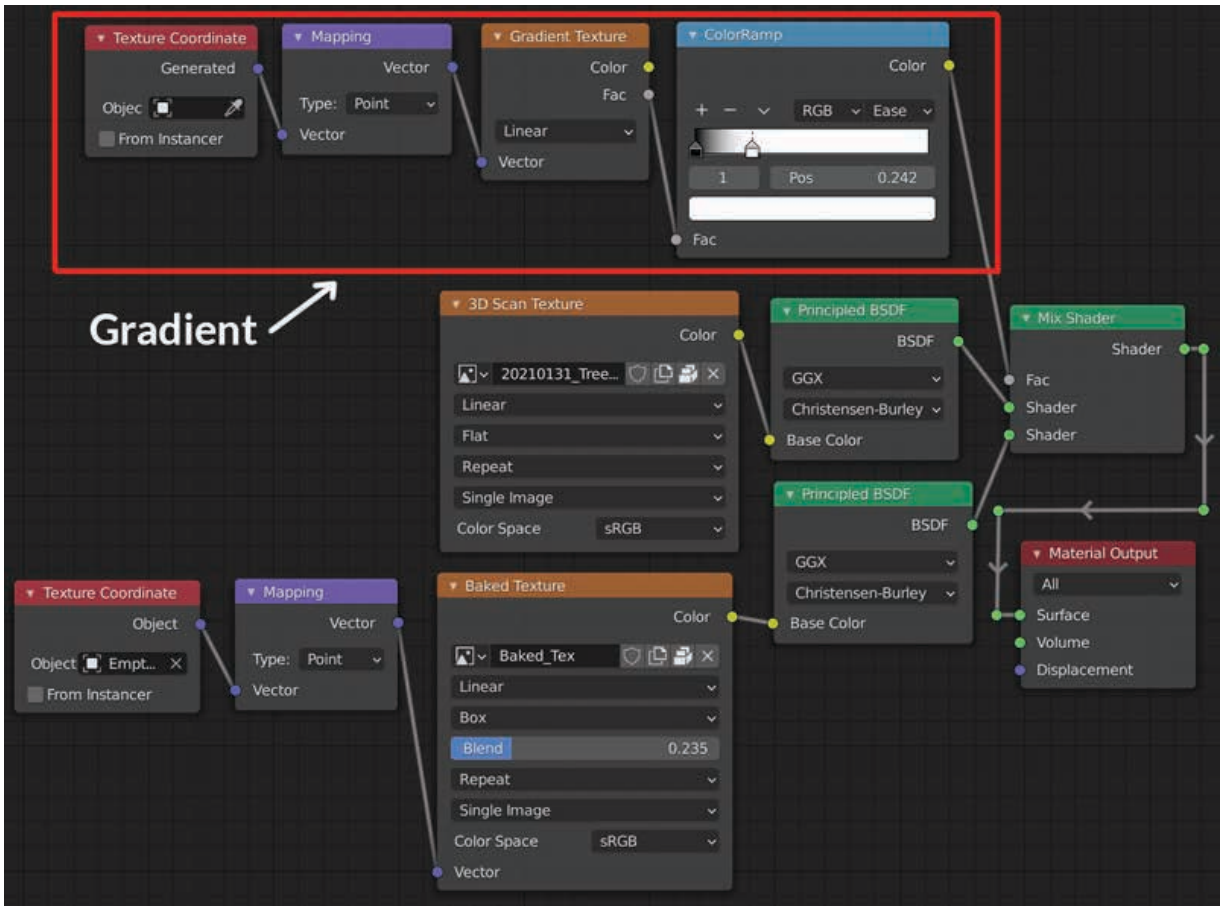
To do this, select the 3D scan and open its material in the Shader Editor. **Add a Mix Shader node** and add nodes for Box Mapping just like you did with the top part of the tree, with the baked texture.

Then plug those into the Mix Shader node. Connect the 3D scan texture to the Mix Shader node as well.

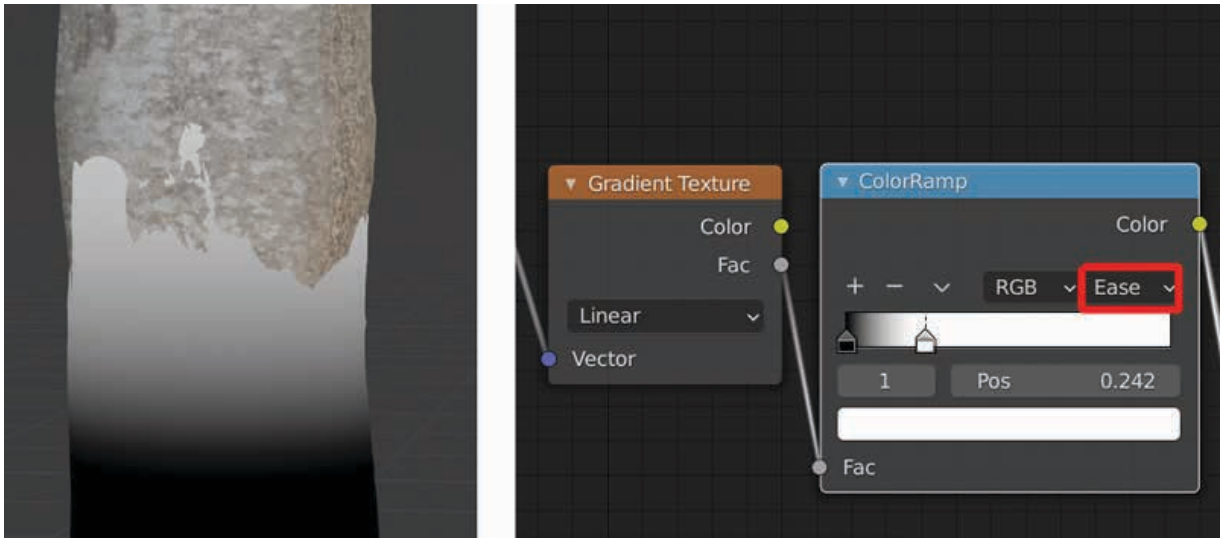


Now you'll see that the Fac value controls which texture you see on the tree. By adding a **Gradient node** and a **Color Ramp node**, as well as **Mapping** and **Texture Coordinate nodes**, we can rotate the gradient 90° and move it to the location where we want the materials to mix.

Using **Node Wrangler**, you can **Shift+Ctrl+click** on the Gradient and Color Ramp nodes to fine-tune them with visual feedback in the 3D viewport.



As you can see, we now have a subtle gradient from the 3D scan material to the Box Mapped material. I found that by setting the ColorRamp node to "**Ease**" I got a smoother gradient. And that's it! You're all done.

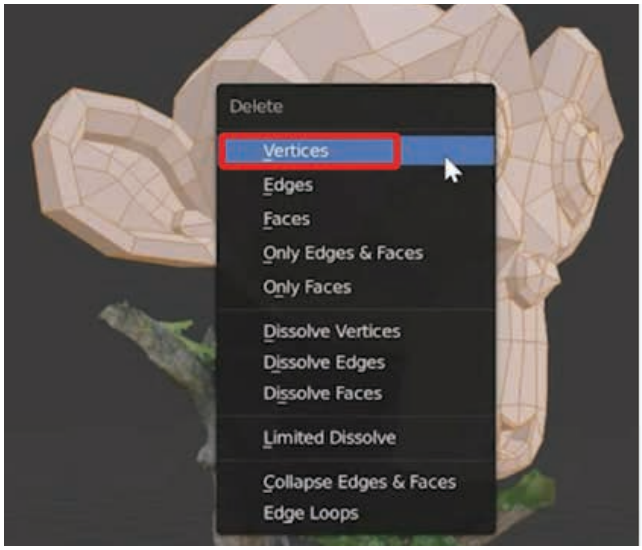


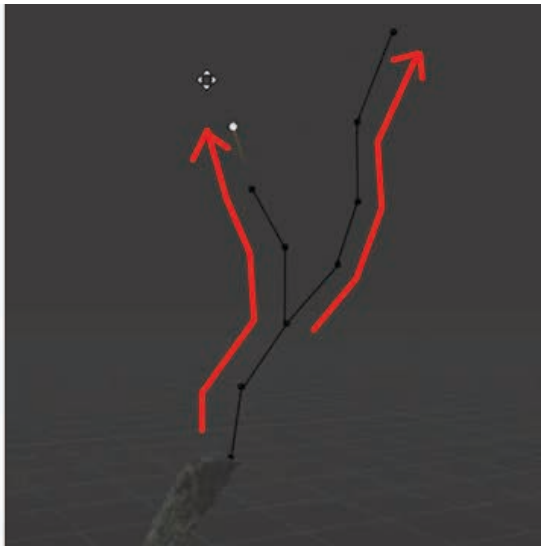
Mixing photogrammetry and hand-made trunks

With what we've learned in the previous topic, we can make even more interesting trees if we model the trunks by hand.

Create a tree trunk using photogrammetry. Add any primitive mesh in Object Mode.

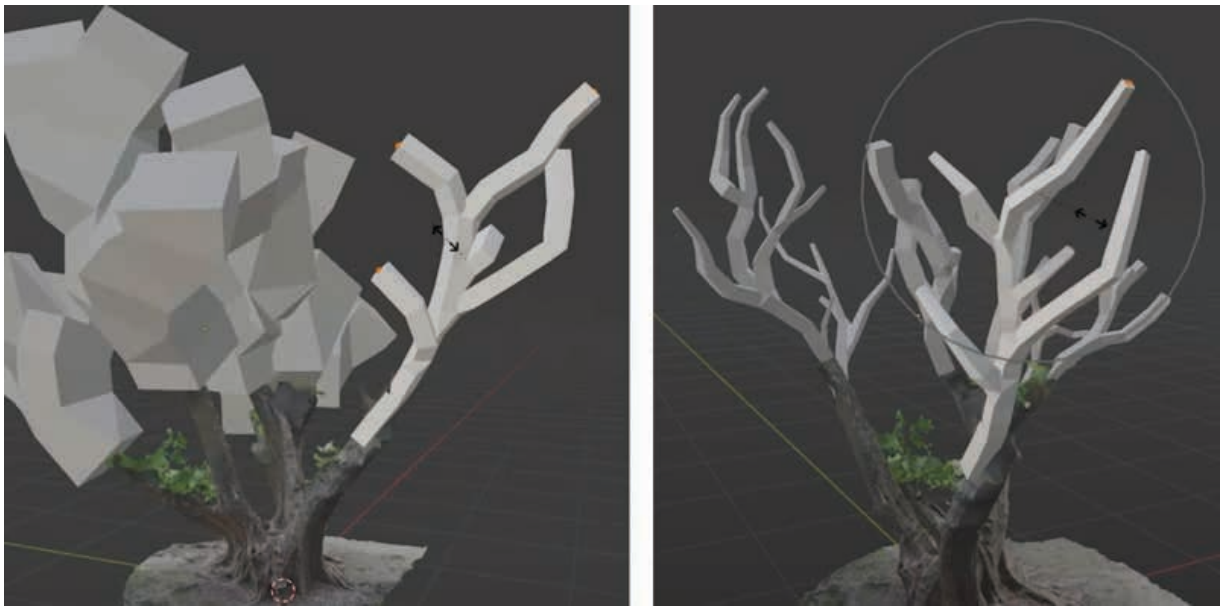
In Edit Mode **press A** to select all, then **X to delete** everything. **Holding Ctrl, right-click** to create vertices where you want to create tree trunks.



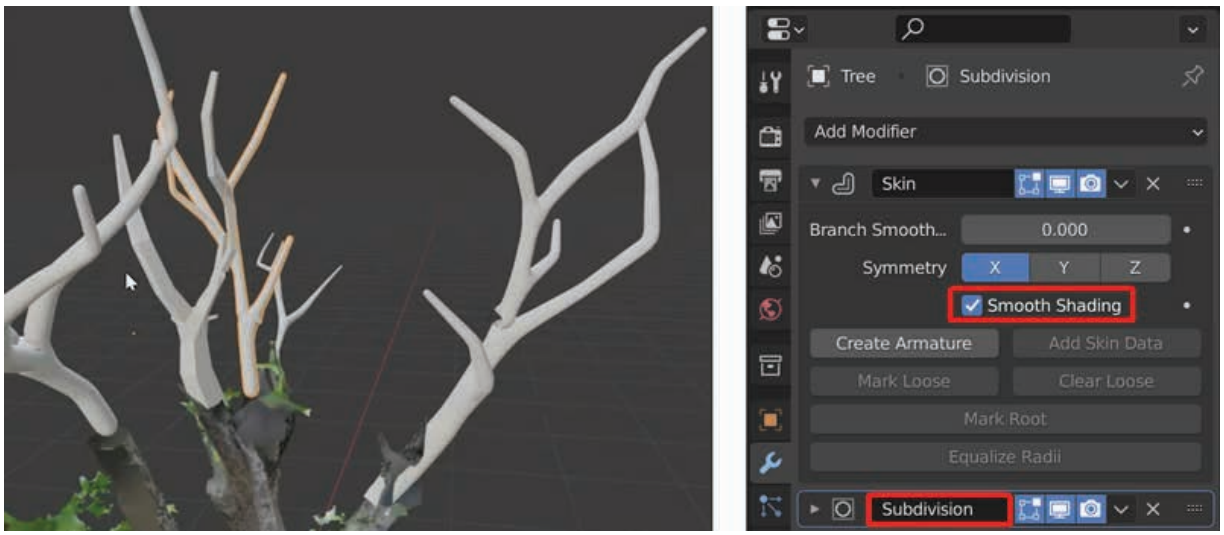


Extrude soms branches by selecting vertices and **pressing E**. By rotating the view each time, you avoid making the tree too flat. Add a **Skin modifier**.

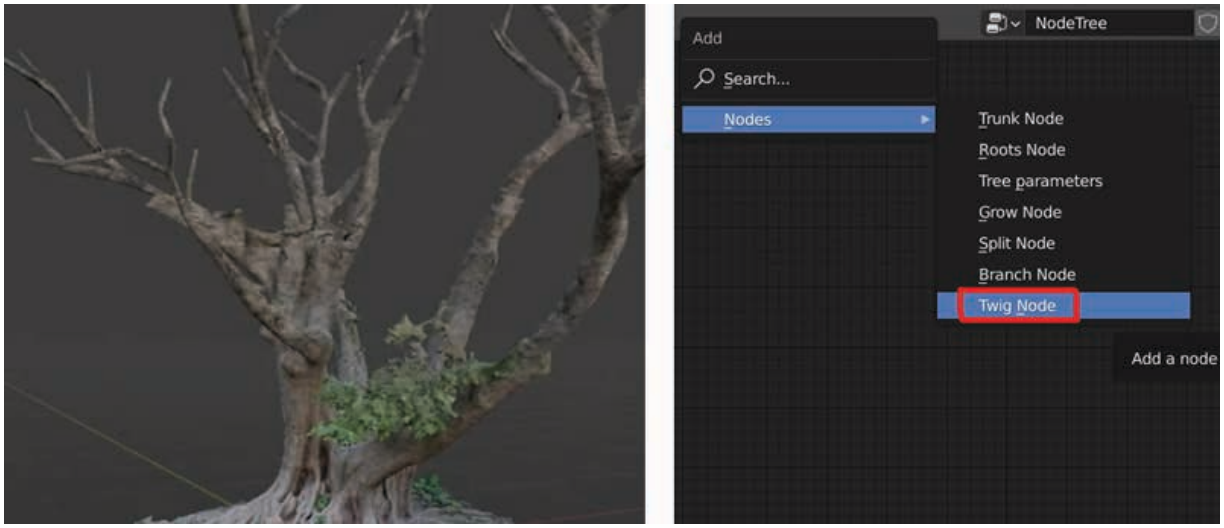
You can change the thickness in Edit Mode by selecting vertices and **pressing Ctrl+A**. You can use **Proportional Editing** here as well. Toggle Proportional Editing on/off by **pressing o**.



Set it to **Smooth Shading** and add a **Subdivision modifier**.



Using the process described in the previous topic about merging photogrammetry trees with procedural trees, add a material and merge the trunks with the base. You can use a Twig node from the MTree add-on to generate a twig with leafs.



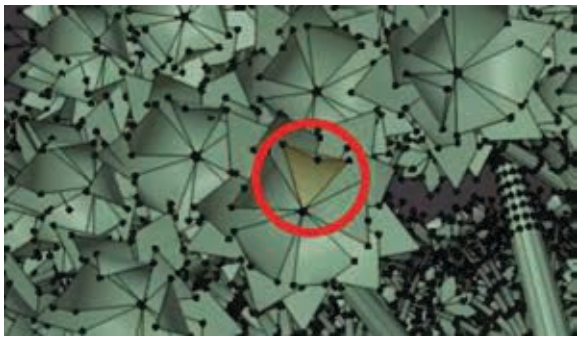
Then you can use that twig in a particle system to add foliage to the tree trunks. Nice!

Selecting lots of tiny things

Sometimes you need to select a lot of similar things like the petals here, to give them their own material.

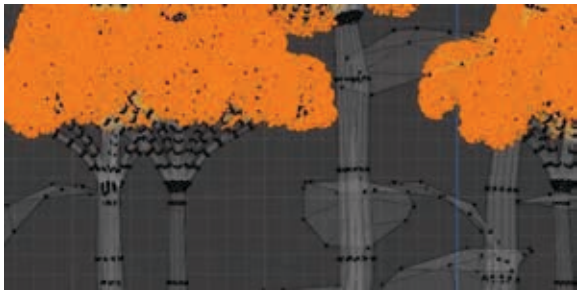
Normally I would use a particle system for all these petals, but when buying models online they are usually just a mesh with thousands of faces.

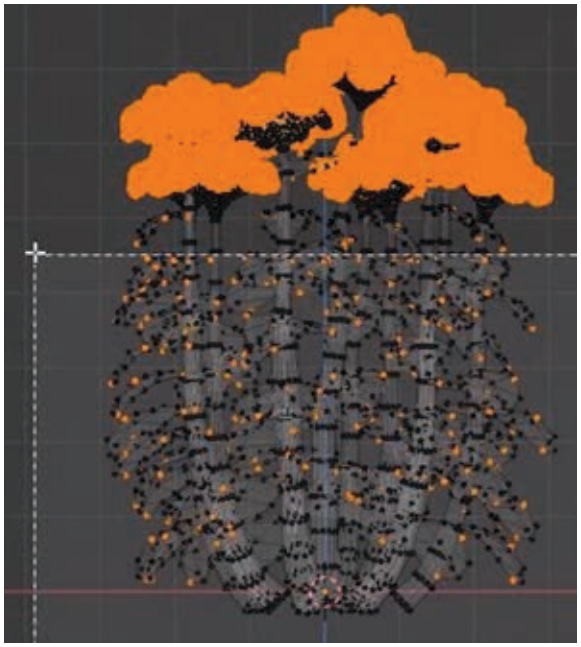
Select the vertices of one petal then go to Select Similar and then **Amount of Connecting Edges**.



In this case it works almost perfectly, we just need to deselect a few things that we don't need.

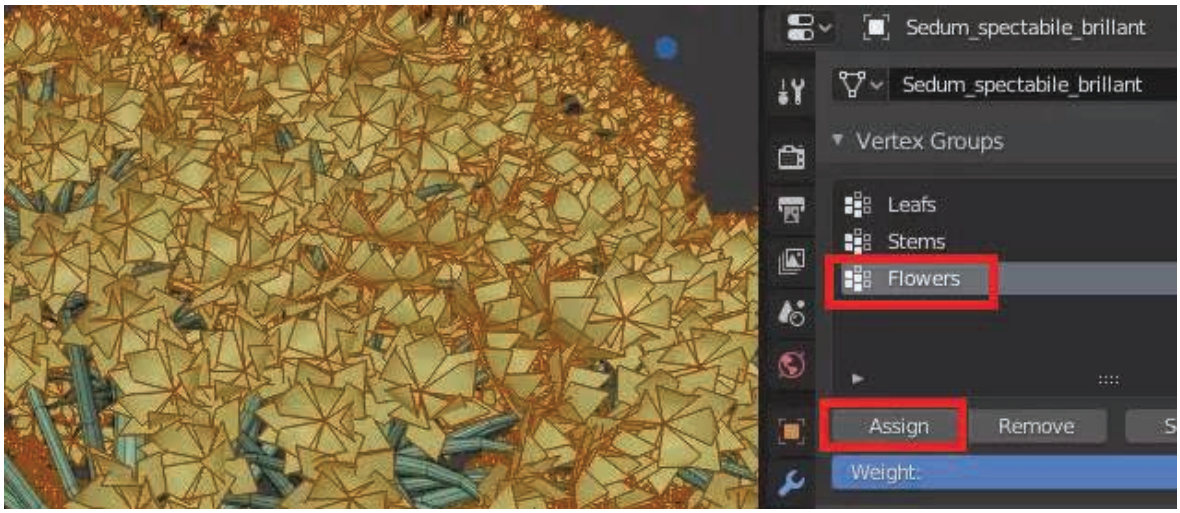
To deselect with a box selection, **press B** for box select, then **hold Alt** while dragging the selection box. After deselecting unwanted vertices, we get this:





Hold Ctrl and click on the **Faces** selection, to convert to a selection of faces instead of vertices.

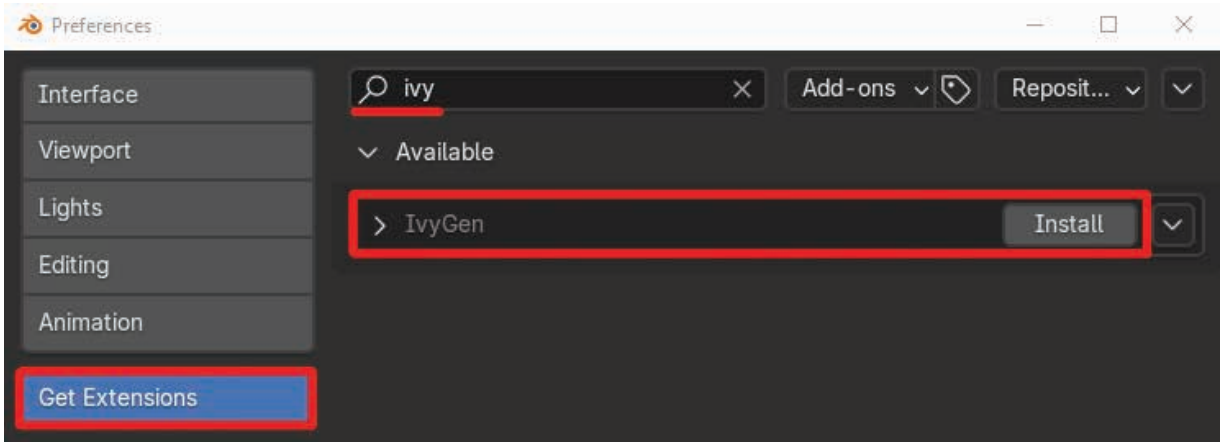
Now assign them to a **Vertex Group** so you don't need to repeat it later. Give it an appropriate name (something like Flowers or Petals in this case) and click on **Assign**.



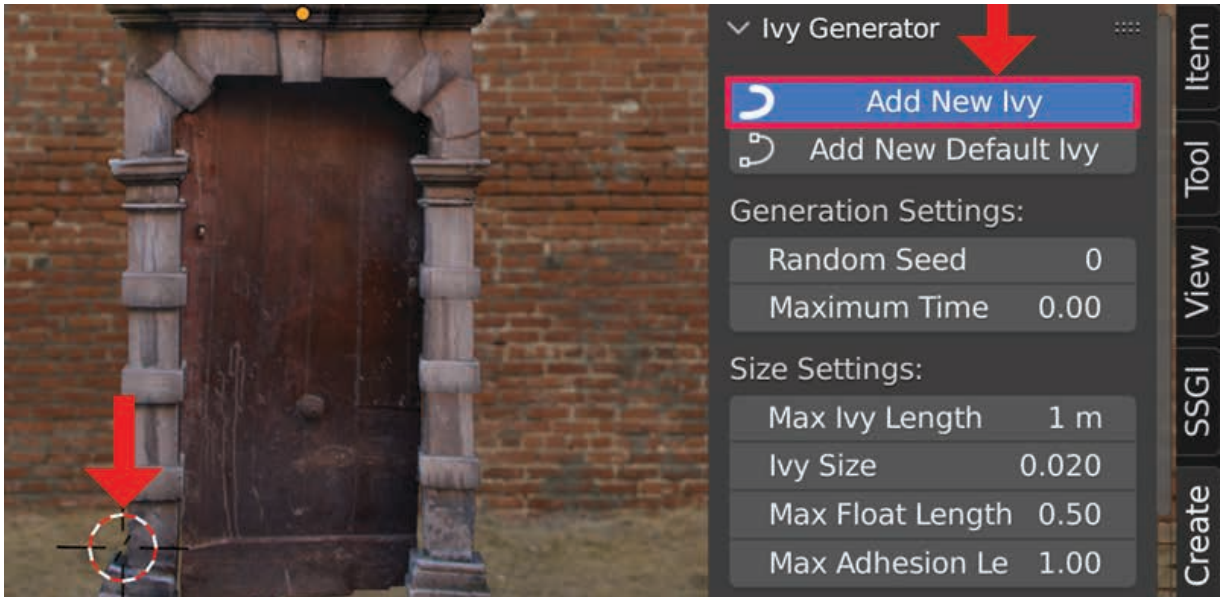
Now any time you need that selection, you select the appropriate Vertex Group and click on **Select**.

Add ivy with IvyGen add-on

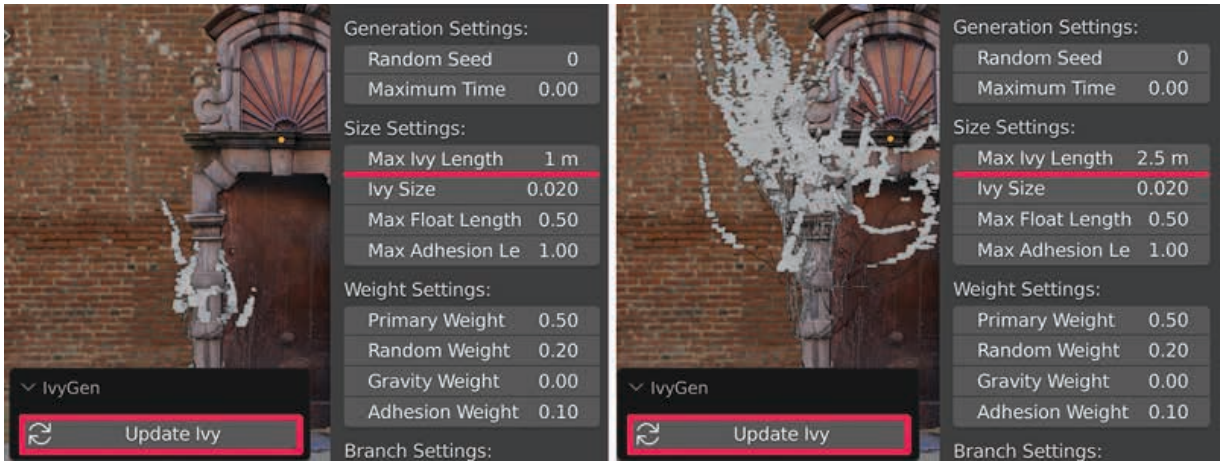
In Preferences, turn on the "**Add Curve: IvyGen**" add-on.



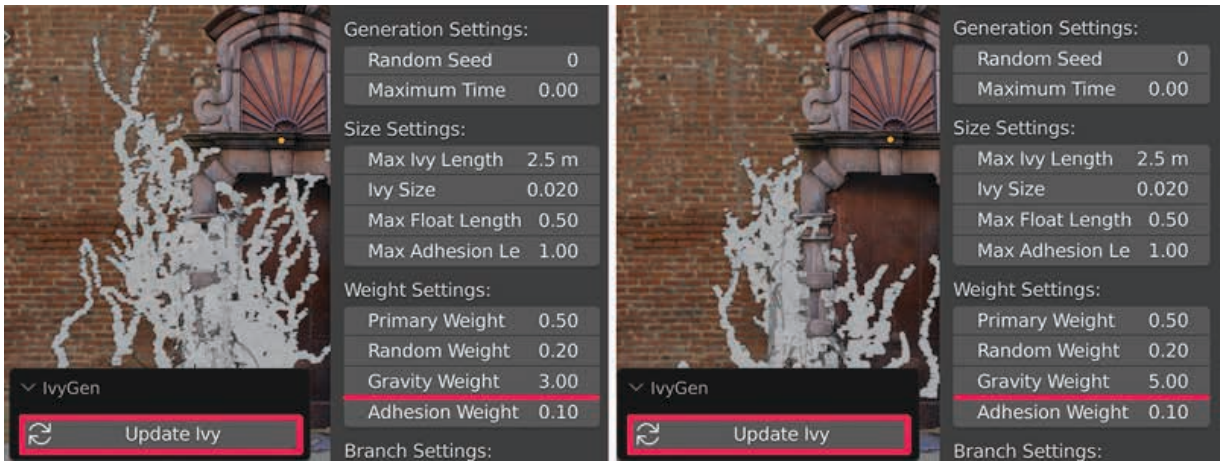
You can download this scene here. To add ivy, simply place the 3D cursor somewhere and click on Add New Ivy.



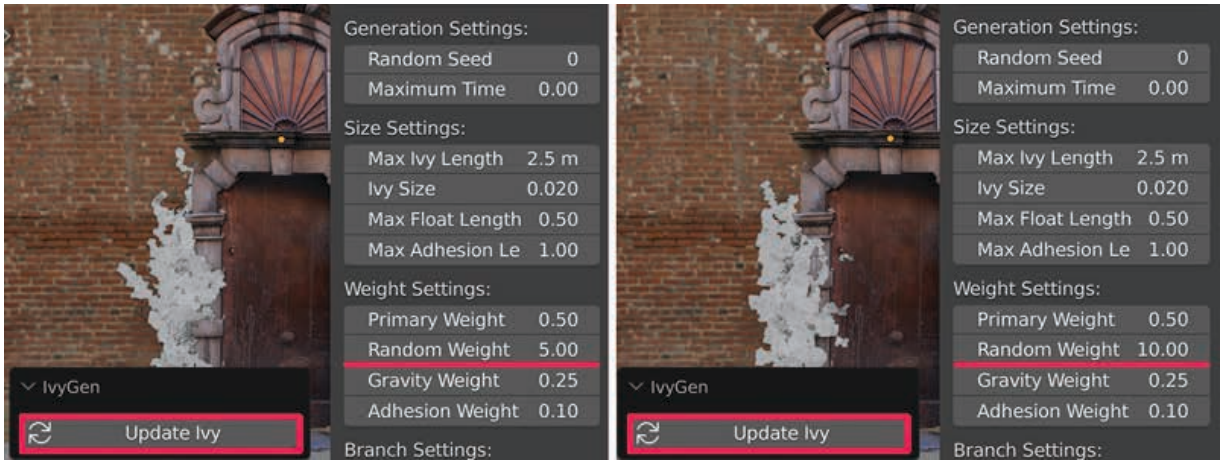
You can experiment with the settings. **Click on Update Ivy** to see the changes. "**Max Ivy Length**" controls how far the ivy grows.



"**Gravity Weight**" pulls the ivy growth down, as if affected by gravity.



"**Random Weight**" makes it more random.

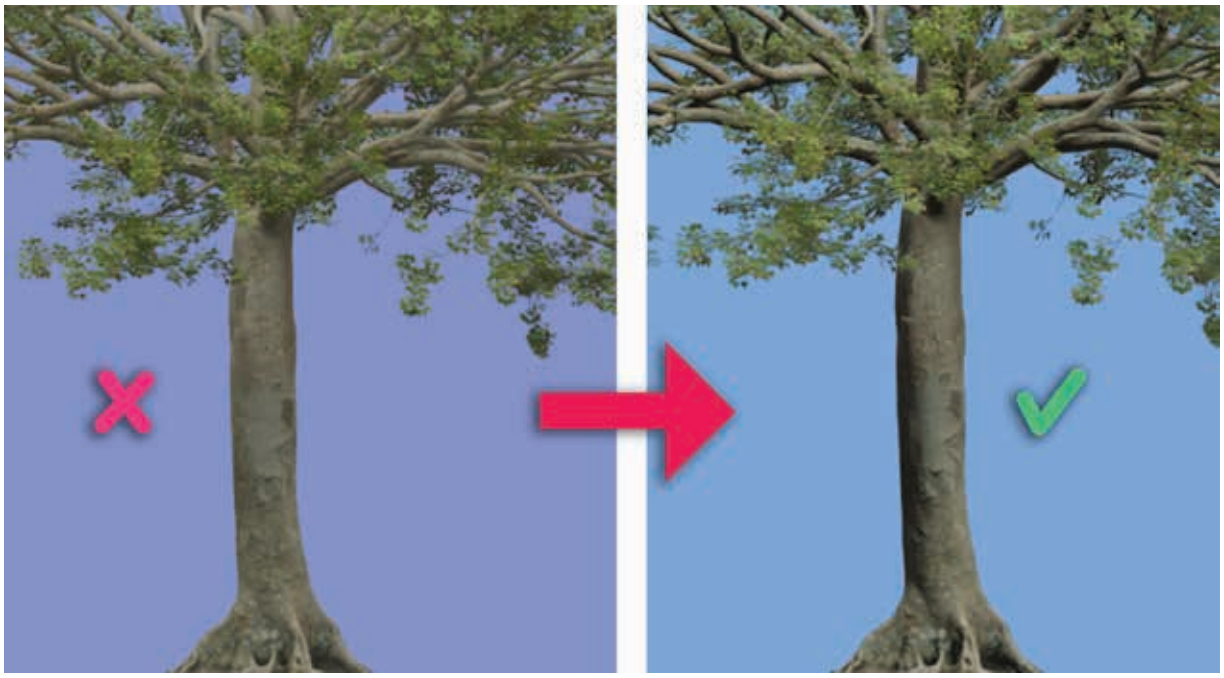


If you click somewhere else, you commit the settings and can't change it anymore. Then you can add materials to the leaf objects and to the stem curves.

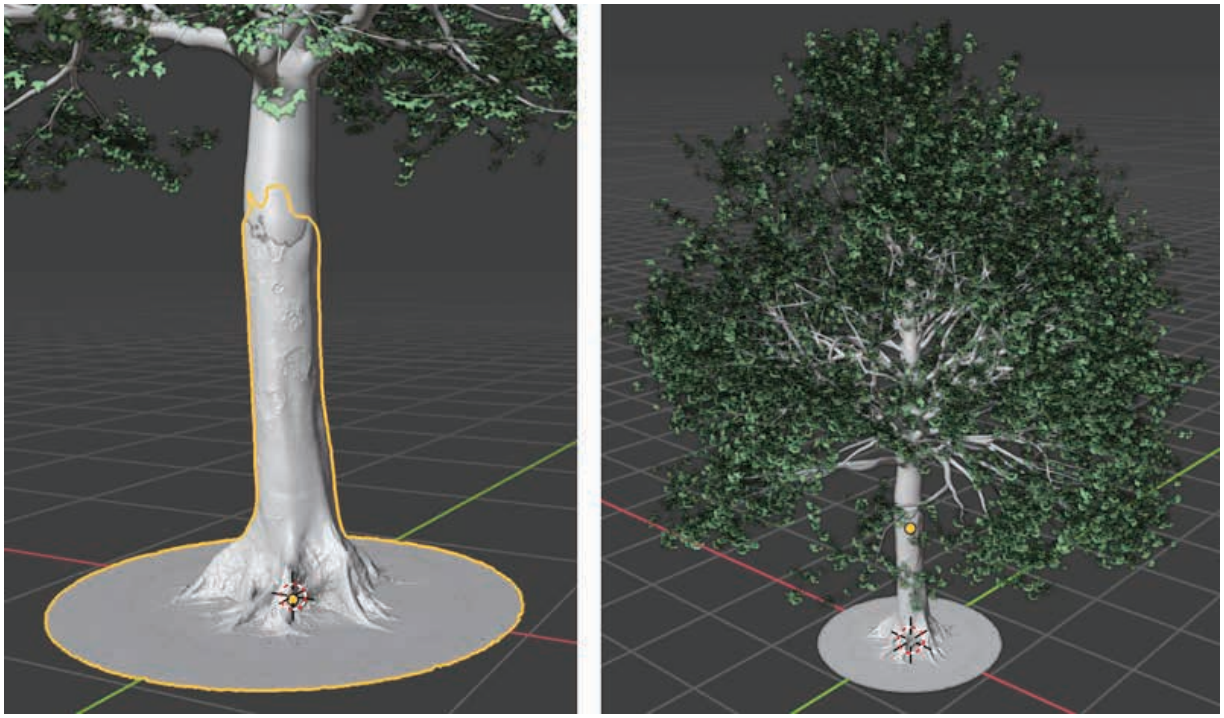
Billboard with Normal Maps

Previously we looked at various methods for making trees, as well as the technique of using "**billboards**" for low-poly trees to create a forest. To achieve this, we used an image of a tree with an alpha texture in on it.

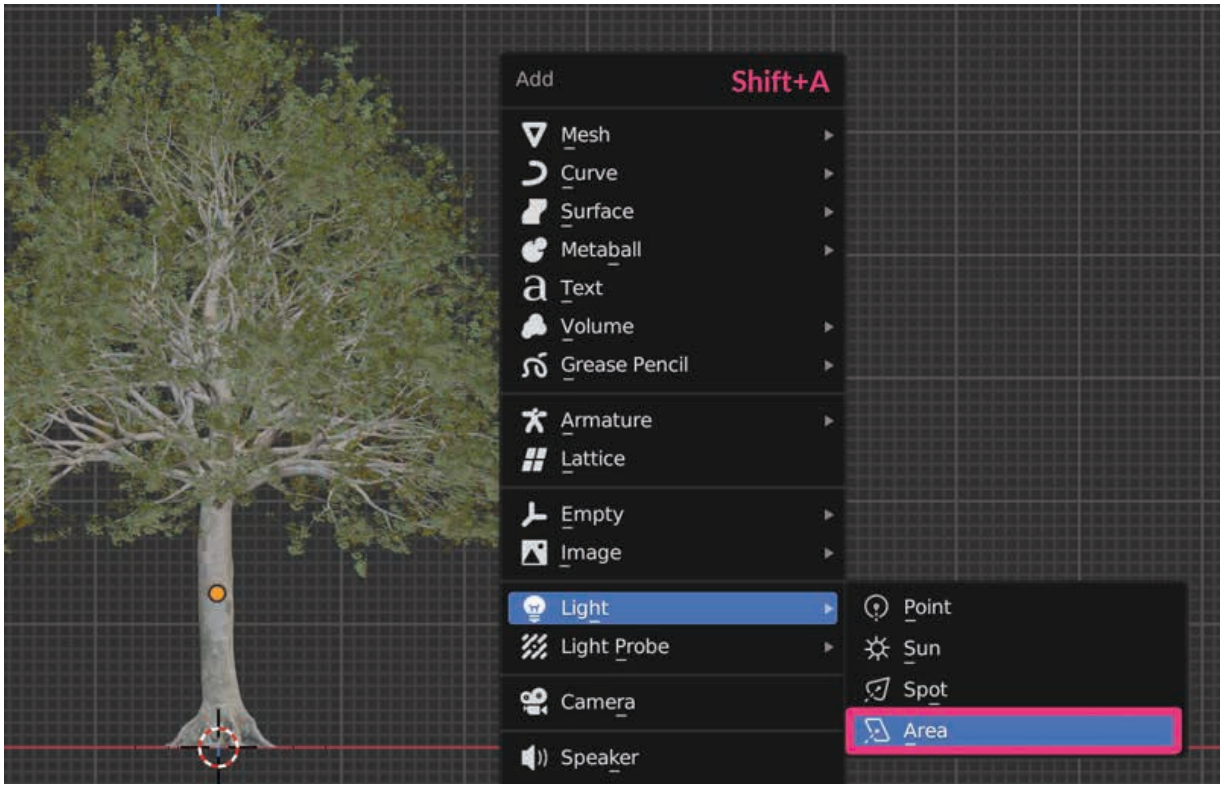
To make the "billboard" tree react better to the light in the scene, we can use **Normal maps**. For this, we'll need to use a 3D tree model as a starting point.



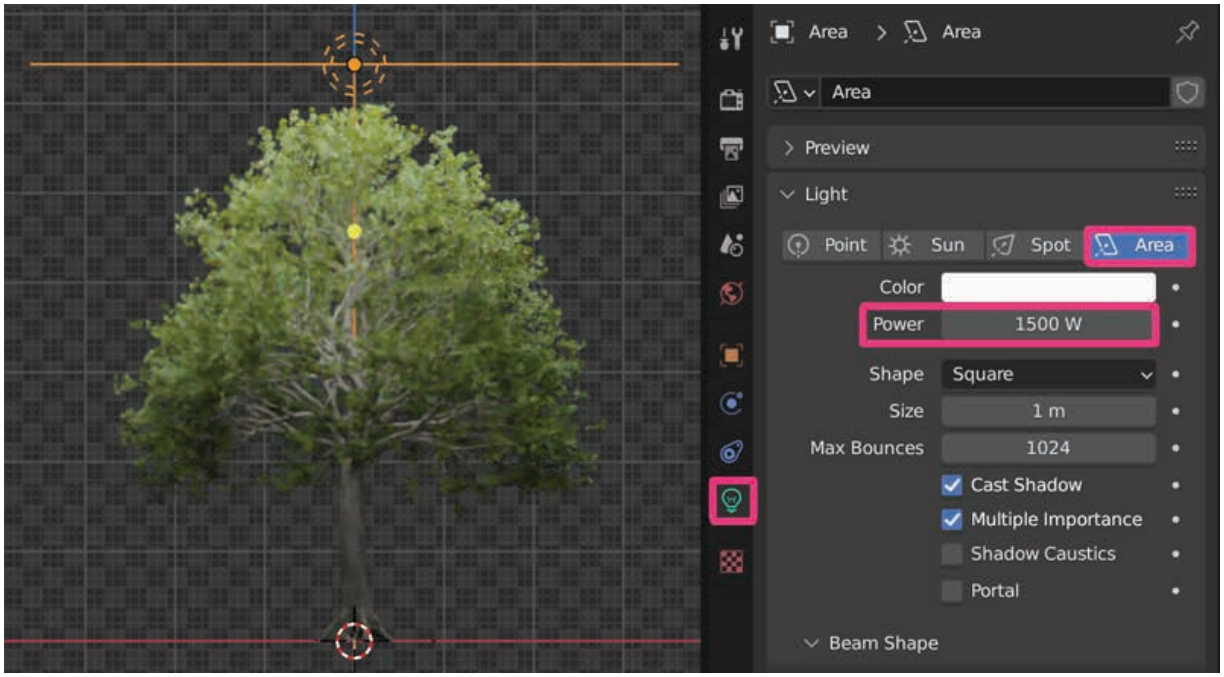
Create a tree using **MTree**, **Sapling**, or by using a 3D scan and manually adding the rest of the trunk and branches. In this example, I used a 3D scanned trunk for the bottom part of the tree and made the top using MTree, as explained in this video.



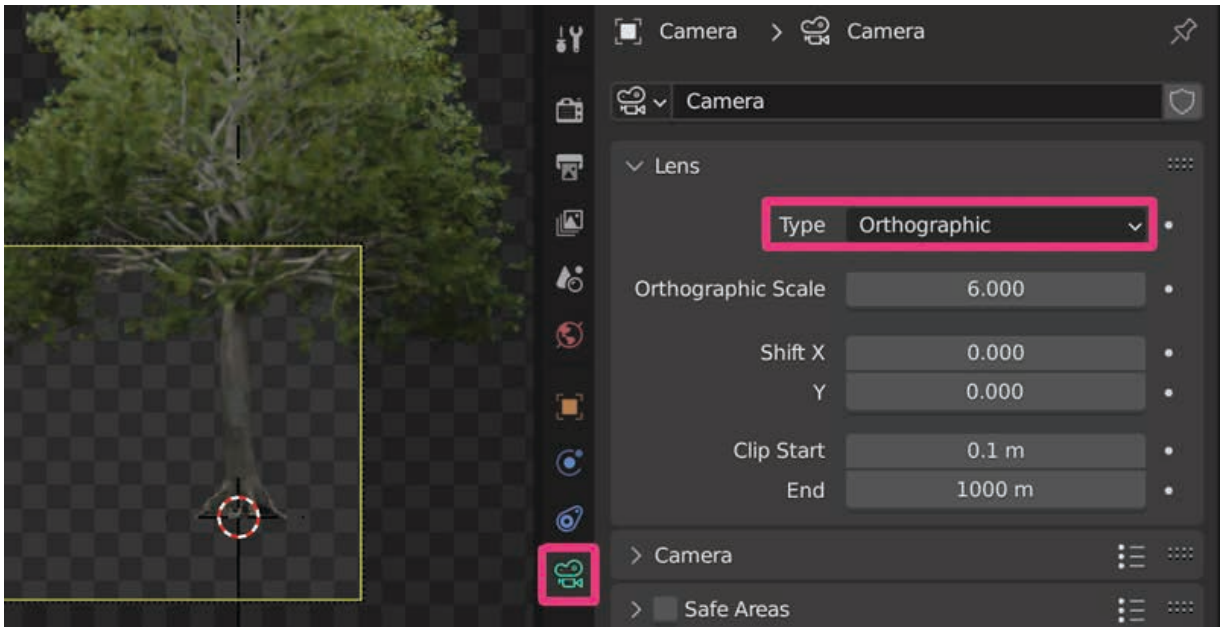
Or, just append a tree someone else made! You can download my tree file here (it's CC0, use it wherever you like). With the tree loaded in the 3D scene, add an **Area light** and move it above the tree.



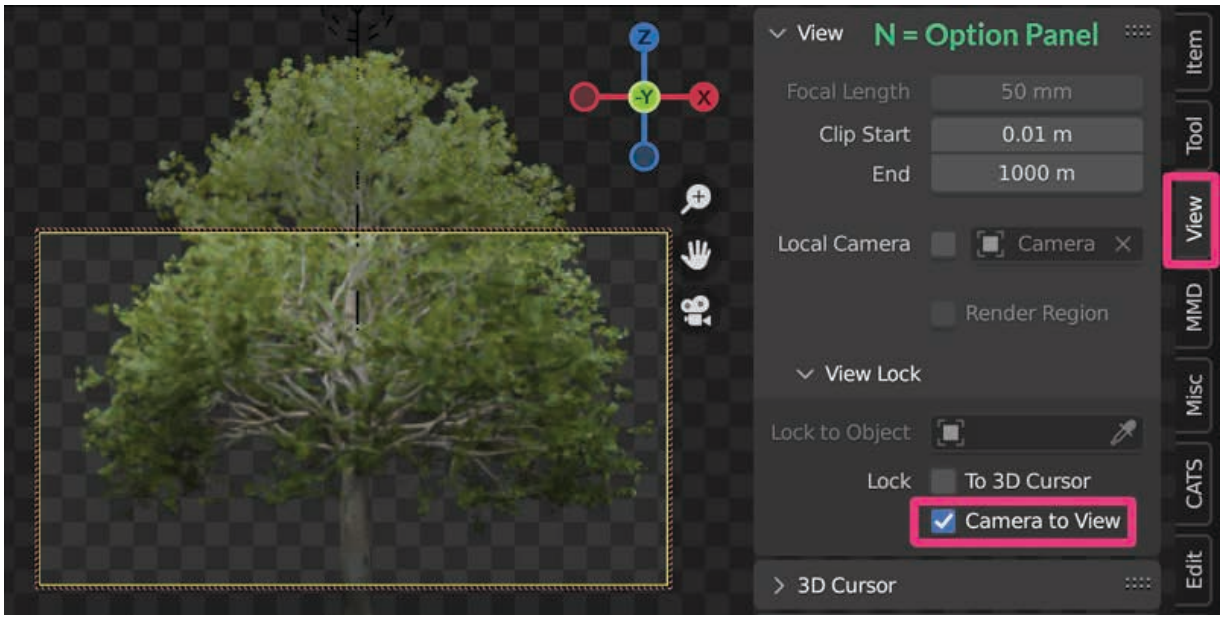
Give the Area light enough **Power**. Scale the Area light up by **pressing S** to make the light more diffuse.



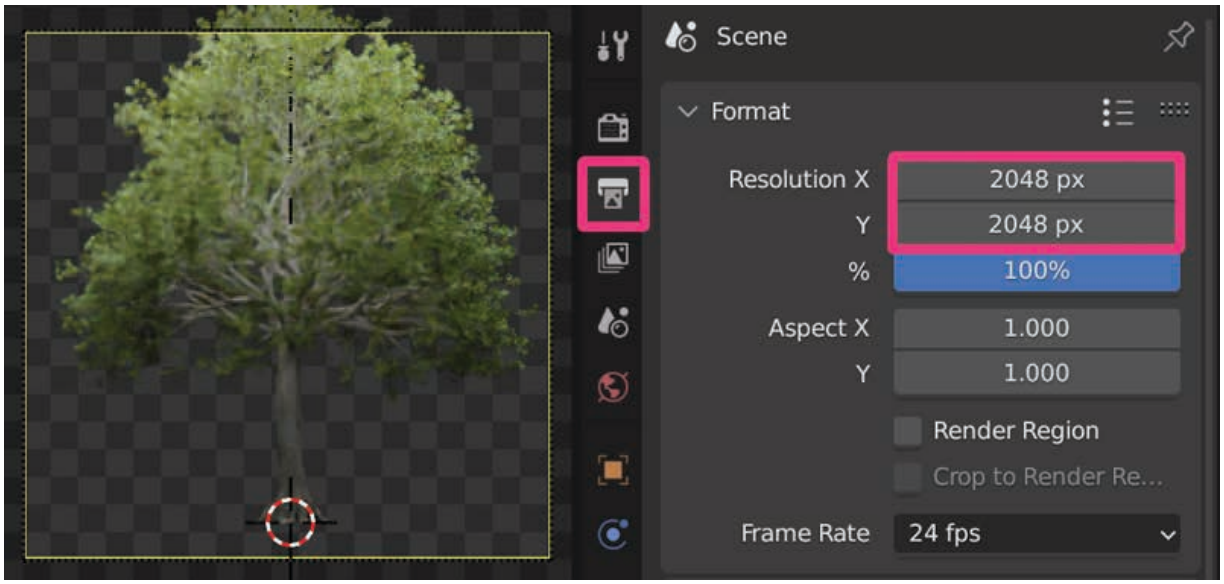
Switch to Front orthographic view by **pressing 1 on the Numpad**. Add a camera (**Shift+A > Camera**) in case there isn't one in the scene yet and set it to Orthographic. You can **Press Ctrl+Alt+[0 on the Numpad]** to match the Camera view to the Viewport view.



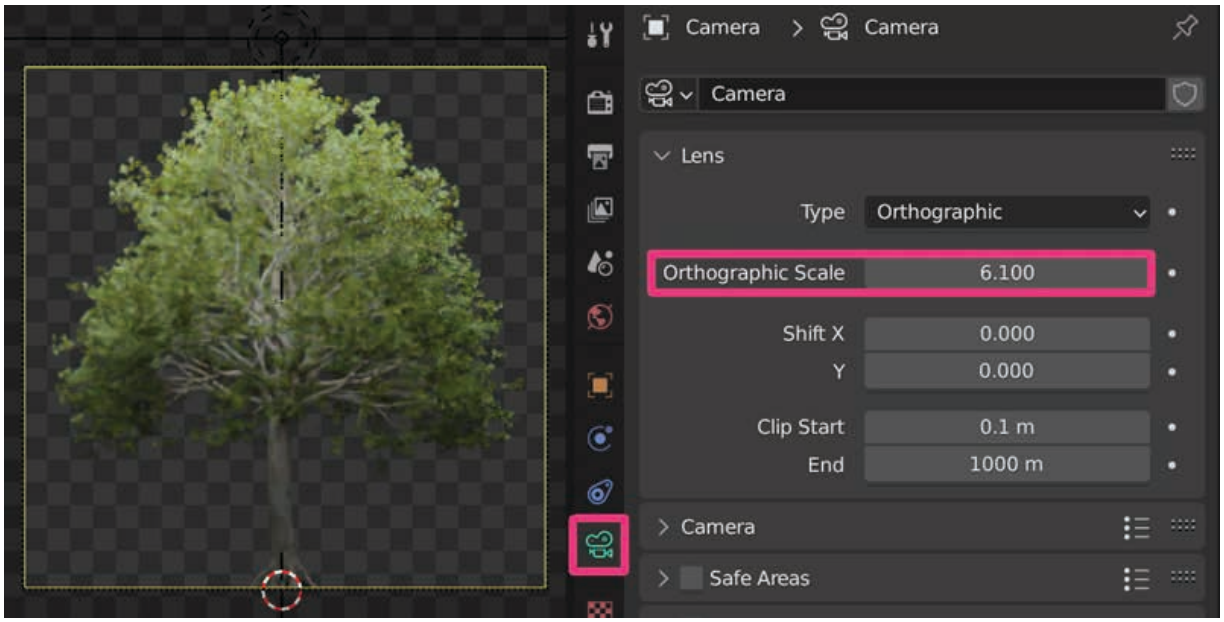
Make sure "**Lock Camera to View**" is enabled and **hold Shift+Middle Mouse Button** to move the camera, to make sure the tree is lined up nicely in the view. It's okay if the tree doesn't completely fit inside the camera window at this point.



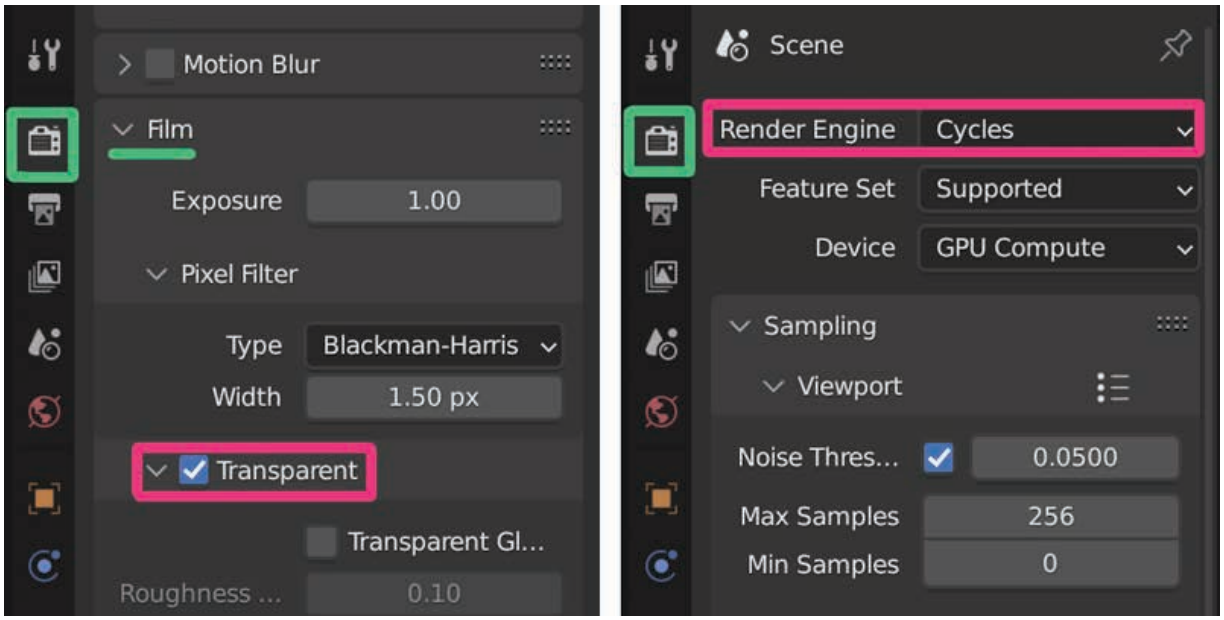
Set the Scene Format to something square, like 2048 x 2048.



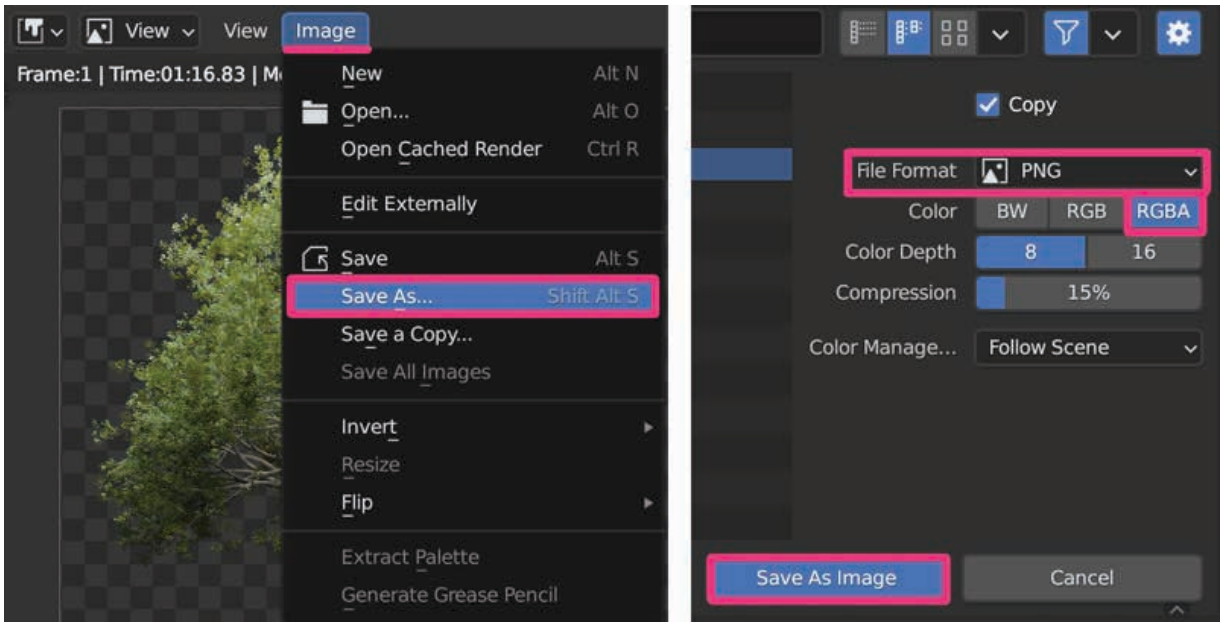
Using the **Orthographic Scale** value of the Camera, you can zoom in or out on the tree, to make it fit the square view of the Camera. The specific value you need will depend on the size of your tree.



Enable **Transparent** and render it in **Cycles**. If possible, use your GPU to speed things up.

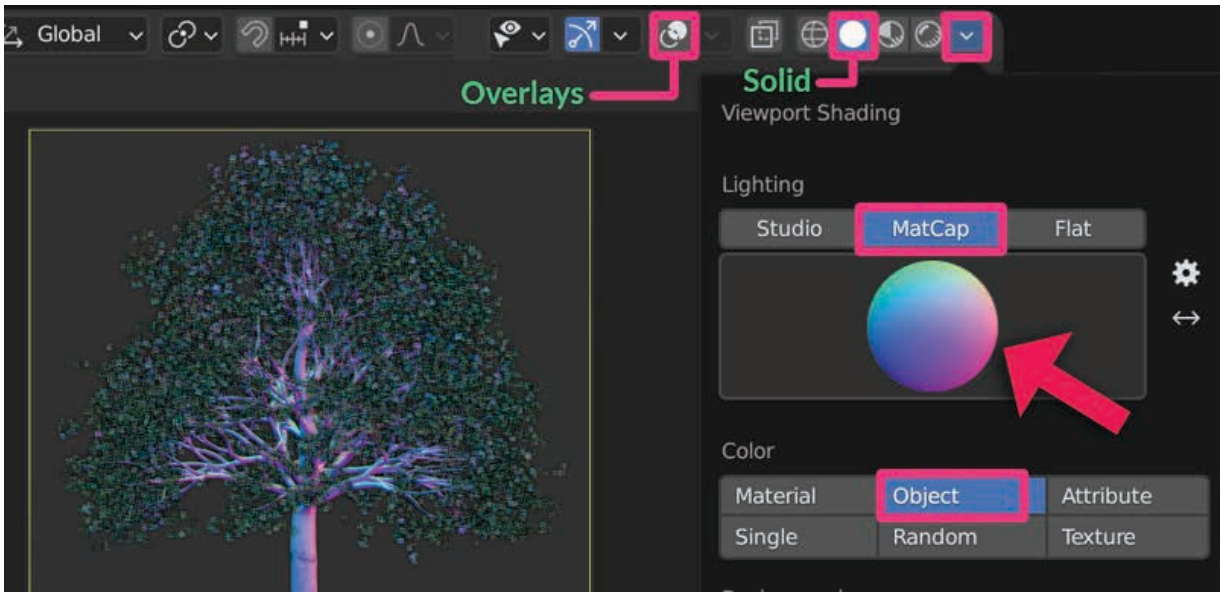


Save it as a PNG image with **RGBA** enabled so that it includes the transparency information in an Alpha channel.

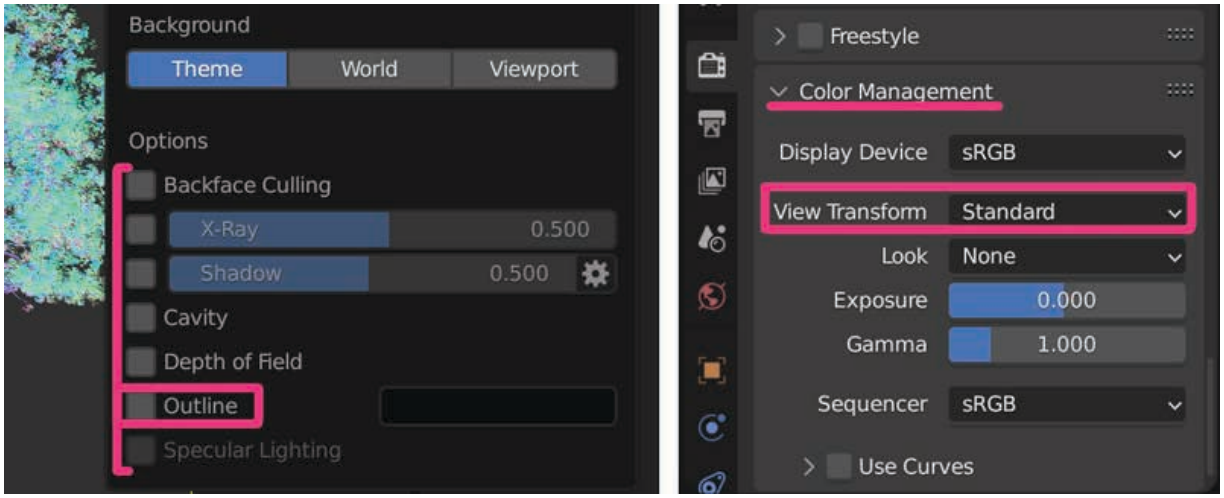


That's the Diffuse map taken care of. Now let's make the Normal map. This will make it possible for the flat image to still give the appearance of being lit from a specific angle.

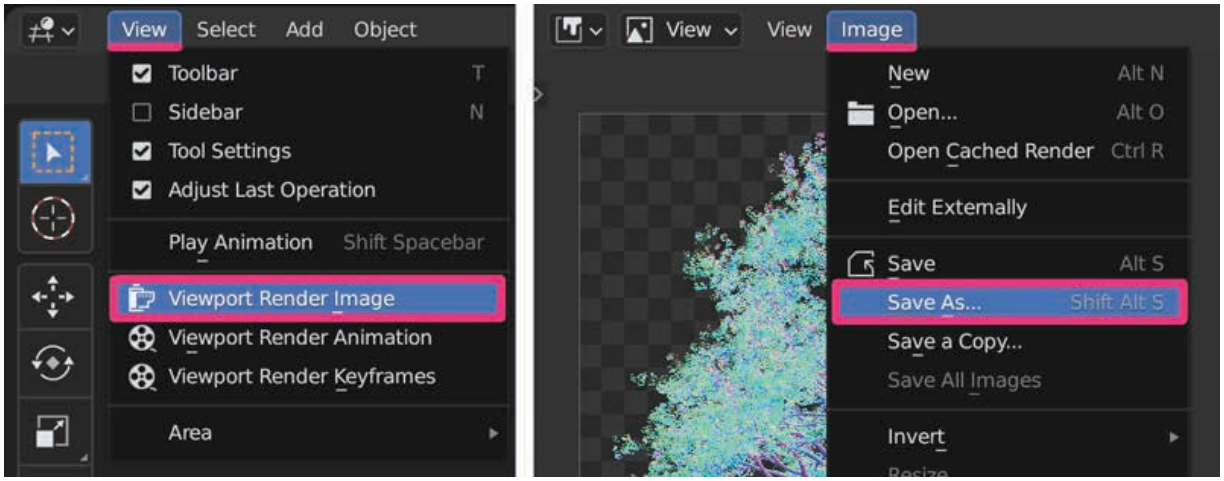
Set Viewport Shading to **Solid** and turn off Viewport **Overlays**. Turn on Matcap shading, set Color to **Object** and choose the "Normal" style Matcap.



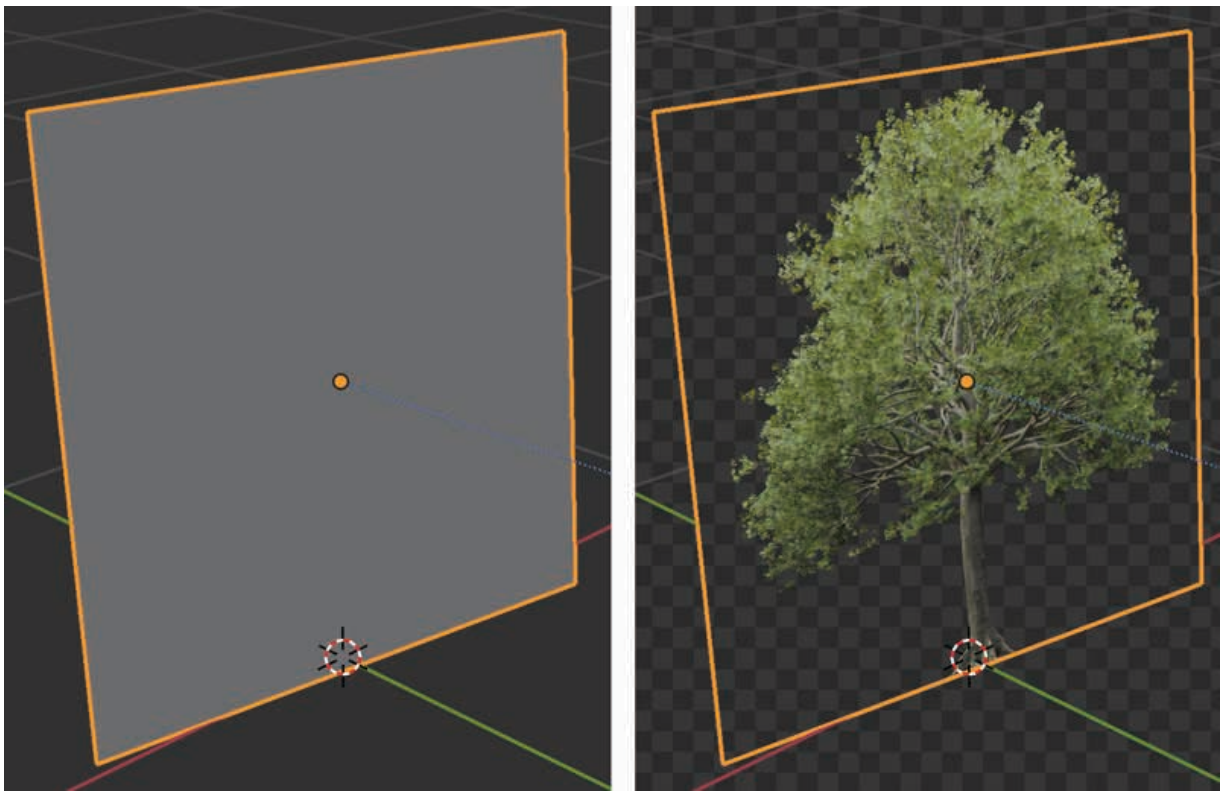
Disable **Outline** and any other Viewport Shading options. Under Color Management, set View Transform to **Standard** with Look "Medium High Contrast" (or "None").



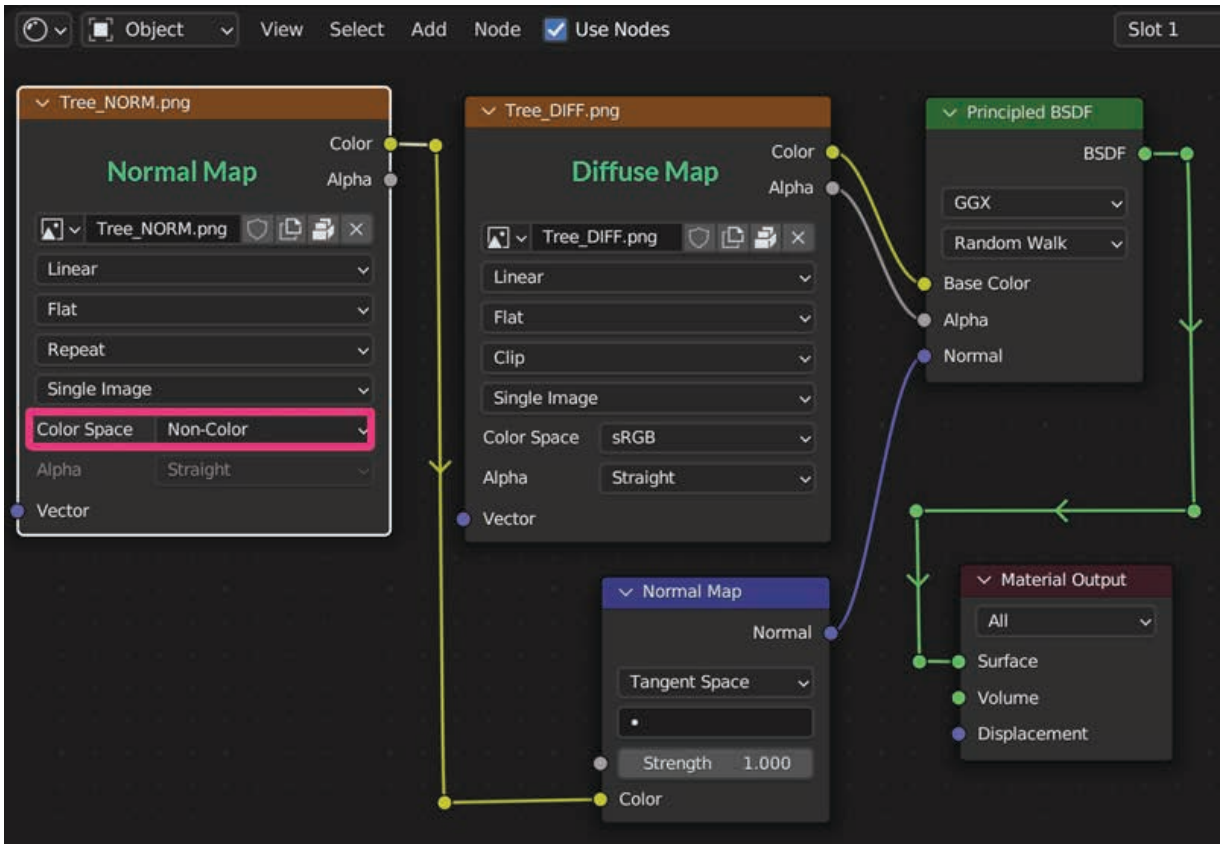
Render a Normal Map with **View > Viewport Render Image** and save it. An 8-bit PNG without Alpha channel is fine in this case.



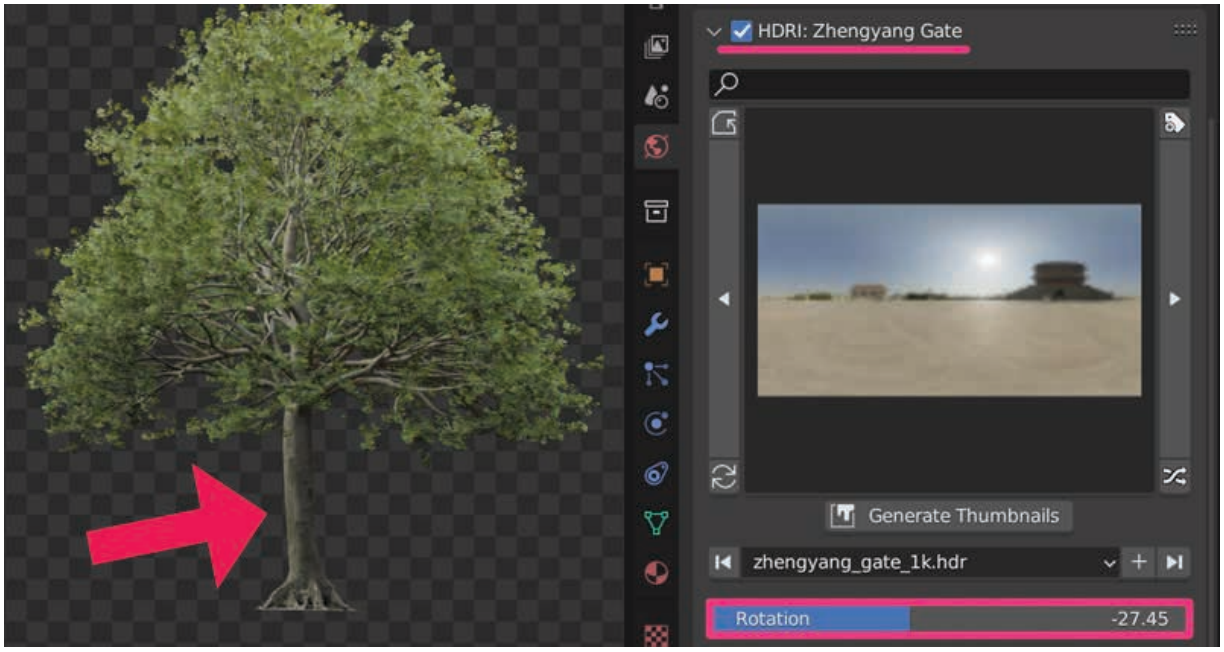
Now let's try out these textures as a "billboard" by loading them on a Plane as a texture set, as explained in the video about using billboards.



Remember to set the Normal Map to "Non-Color".



Now you can see, that when we rotate the HDRI (while is easily done using the Gaffer add-on), the shadows on the tree change to reflect the direction the light is coming from.



Create a mysterious city with Blender and fSpy

I had the opportunity to take some photographs from the Tokyo Metropolitan Building, which are useful for this tutorial. You can download these photos here. Let's turn one of these photos into a dark, mysterious and dystopian 3D landscape and render a short animation.

To do this we'll use **fSpy**, which is both a free stand-alone app and a free Blender add-on. You'll need the app to create the fSpy files, and the add-on to import them in Blender.

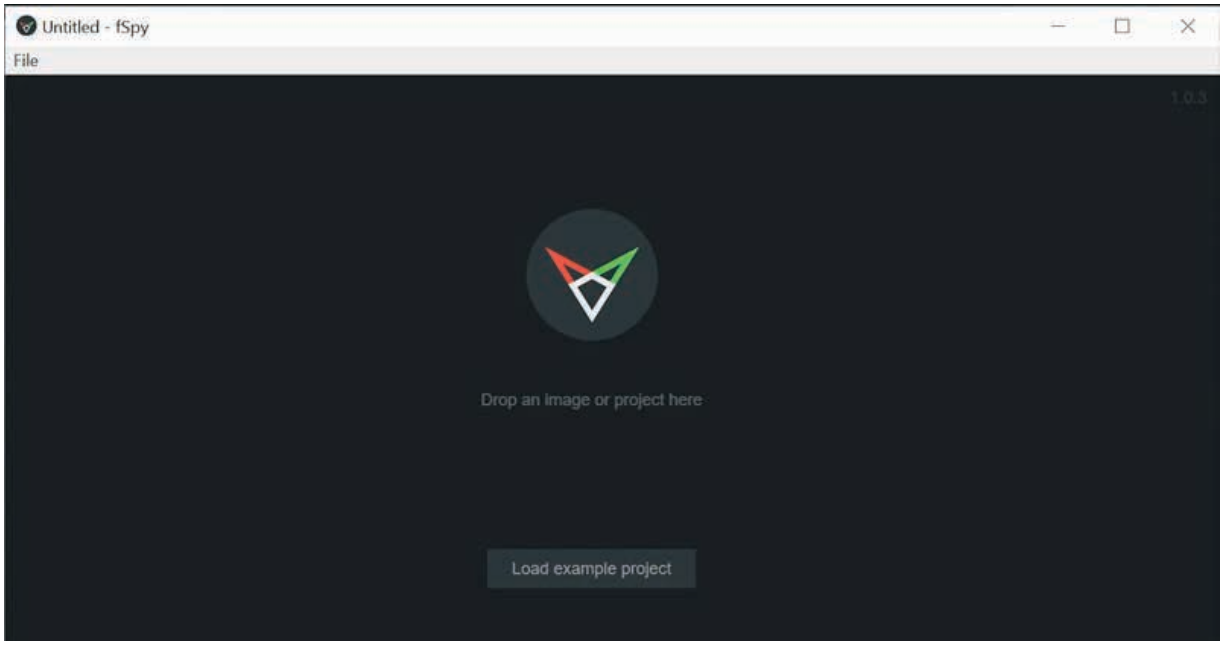
I've noticed from the comments on older tutorials about this topic, that a lot of people download the **wrong version** of fSpy and therefore think it no longer works. So please make sure you use the links for the correct versions, found on the next page.

Download fSpy stand-alone app:
https://github.com/stuffmatic/fSpy/releases/tag/v1.0.3

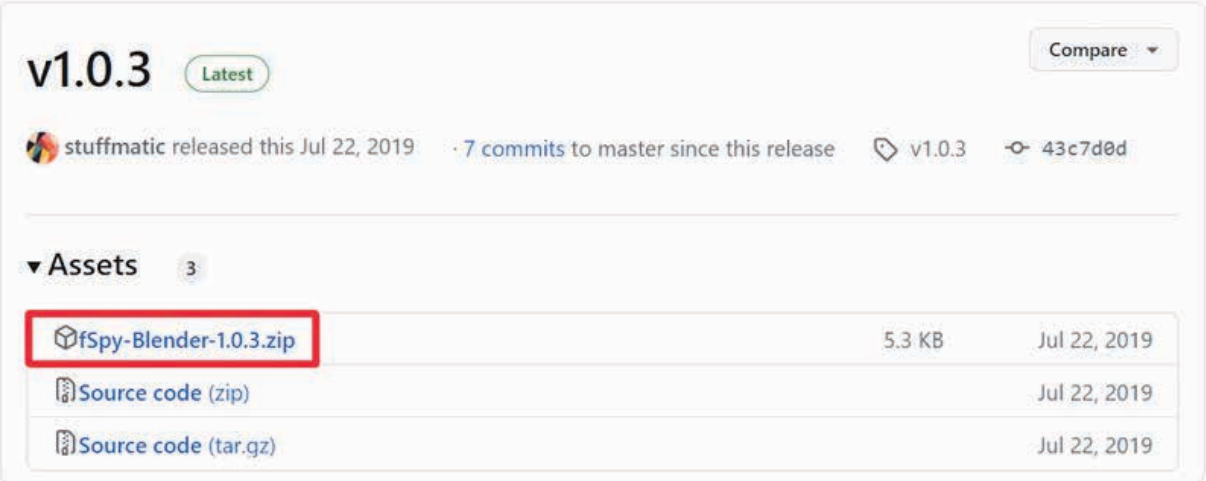
Download fSpy add-on:
https://github.com/stuffmatic/fSpy-Blender/releases/tag/v1.0.3

To get started, download the fspy setup file, **fspy-setup.1.0.3.exe**, from Github. Then click on the downloaded file to install it.

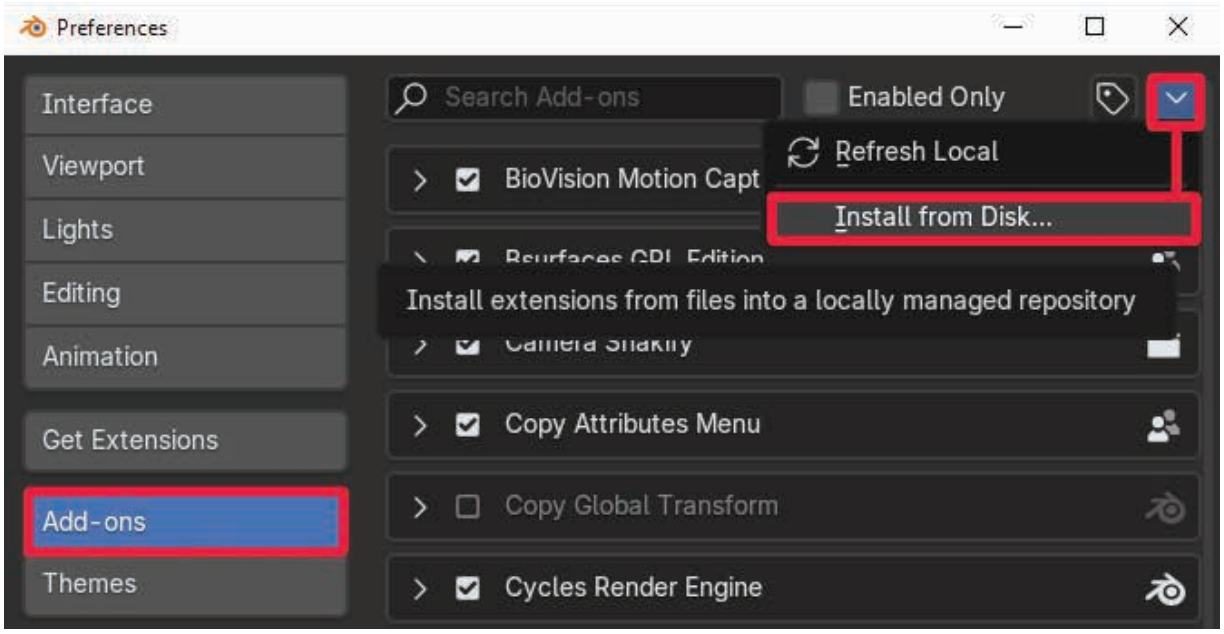
Below you see the interface of the **stand-alone fSpy app**. In this window, you can drag and drop a photo to get started (more in that later). It's a pretty simple app to use.



You'll also need to download the Blender fSpy add-on, version 1.0.3.
Without this add-on you won't be able to import the files made in the fSpy app into Blender.
Download the ZIP-file, but don't unzip it! This goes for most blender add-ons that you download from the web.

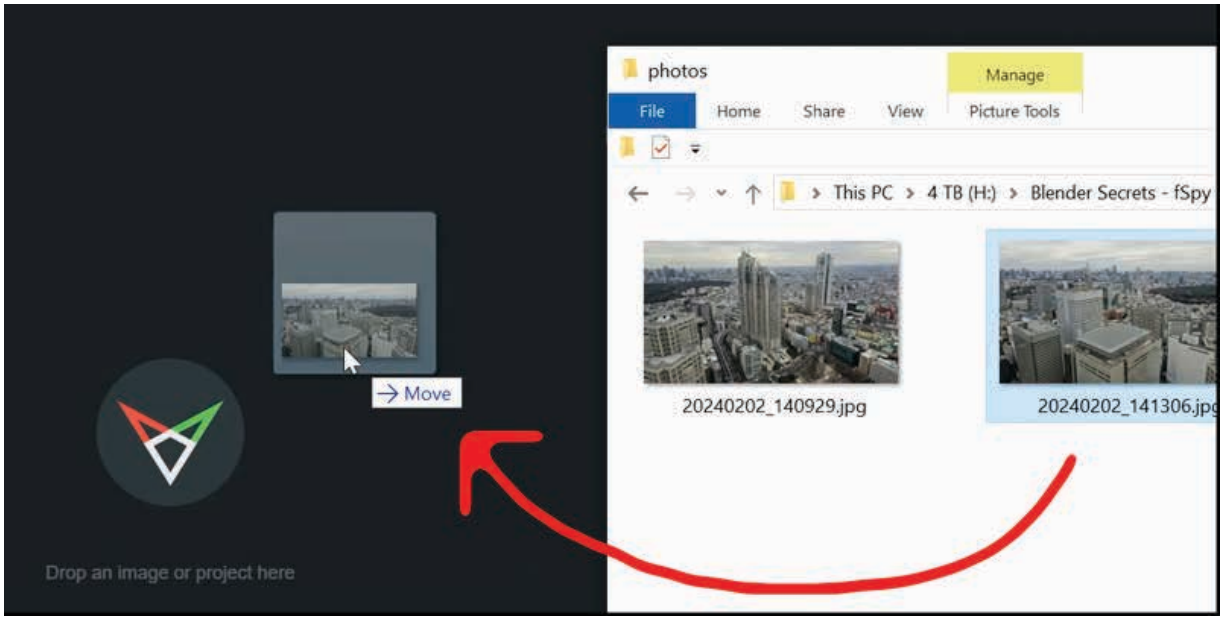


In Blender, go to **Preferences** and install the add-on. Click on **Install** and select the downloaded ZIP file.

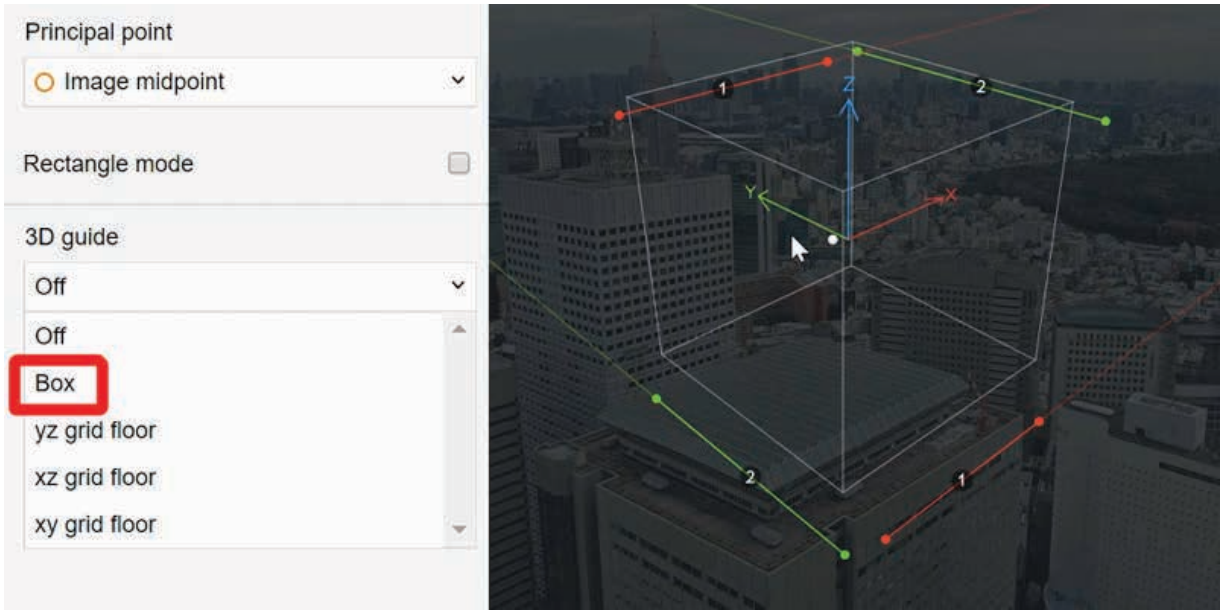


Installing it this way should automatically enable the add-on.

Drag and drop a photograph into the stand-alone fSpy app.



We can enable a **3D guide** in the shape of a **box**. You can move this cube around to check if the perspective lines up with the photo. At first it won't line up, so we'll have to adjust the vanishing points.

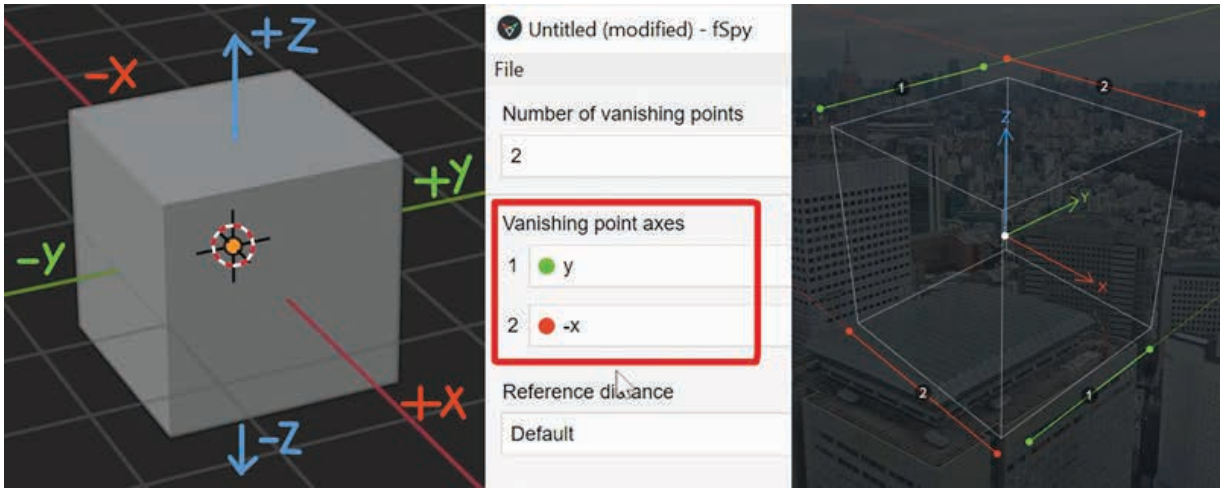


I would like to have the **X, Y and Z-axes** in fSpy point in the same direction as in Blender.

That way we won't end up with some weirdly oriented upside down scene in Blender. For reference, you can see the axis in the Navigation Gizmo in Blender.

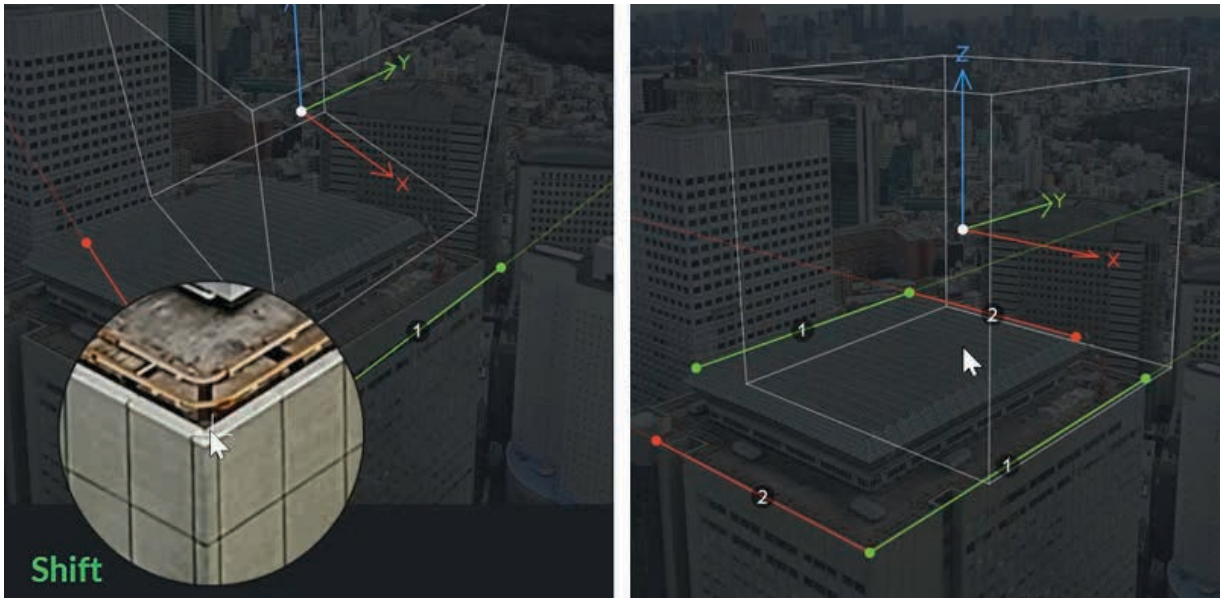
To make the axes in fSpy correspond to those in Blender, we will need to change the **vanishing point axes**. To y and -x in this specific case (this depends on your photo, you'll need to experiment).

As you can see, now they match those in Blender.



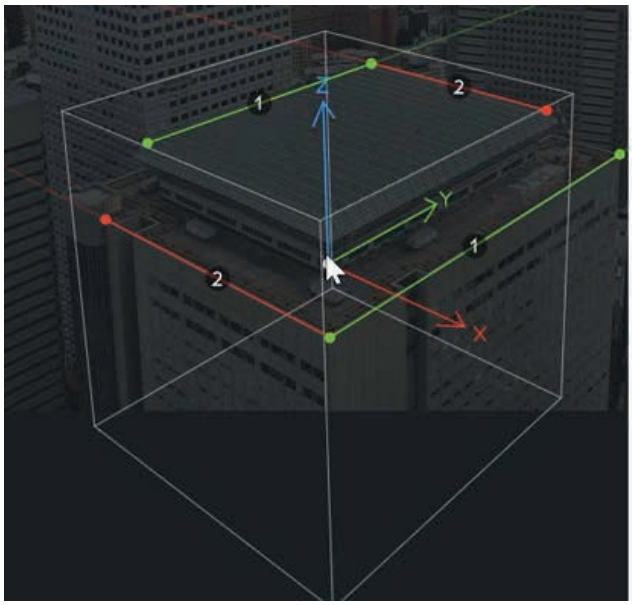
Now we can start aligning the vanishing points, by **clicking on and then dragging their end points**.

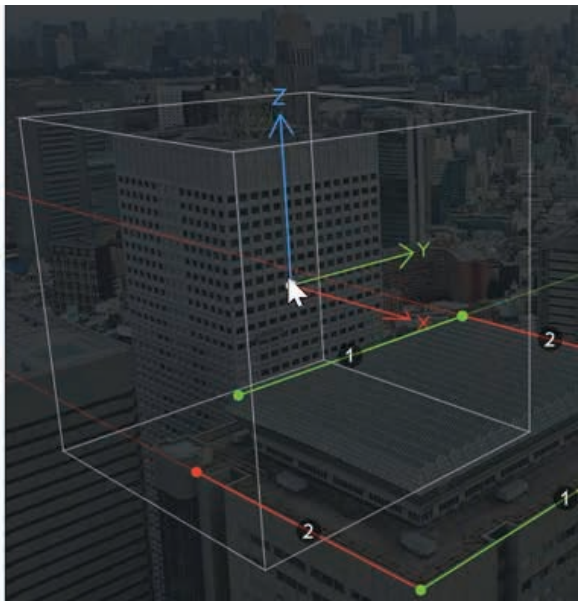
By **holding Shift**, you can zoom in to more precisely place them. It looks a bit like you're looking through a sniper rifle scope...



In this case, I'm using the building in front as reference as it has some clear perspective lines.

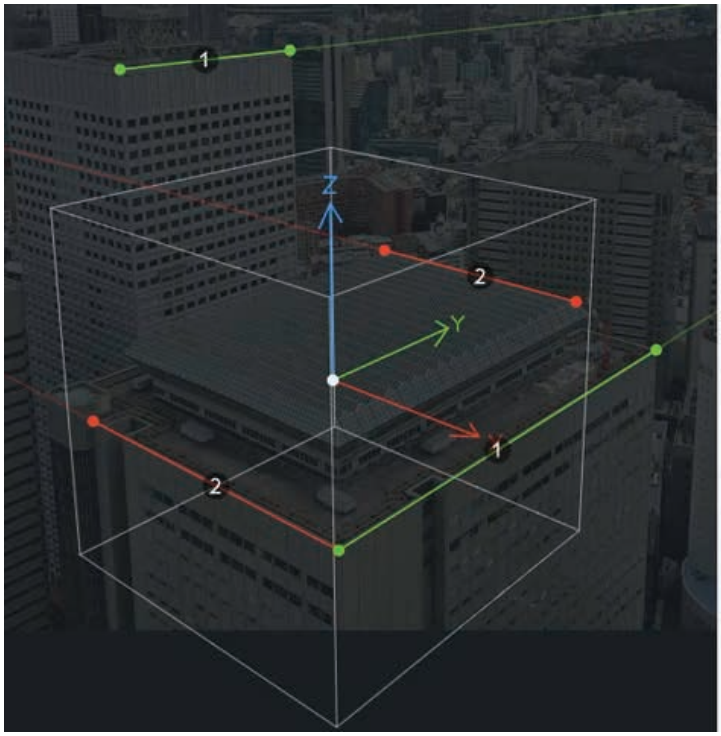
We can test if the perspective of the **3D guide cube** matches the buildings. It seems to be almost perfect. But not quite, as the Z-axis still seems to be a little different than the vertical edge of the building.

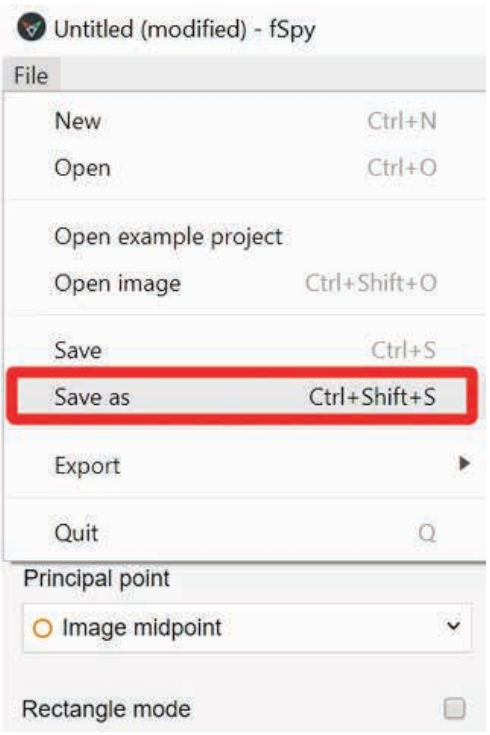




I think we can improve it by using a building further away as reference for this vanishing point. Now it seems to align even better. The next step is to **save this file**.

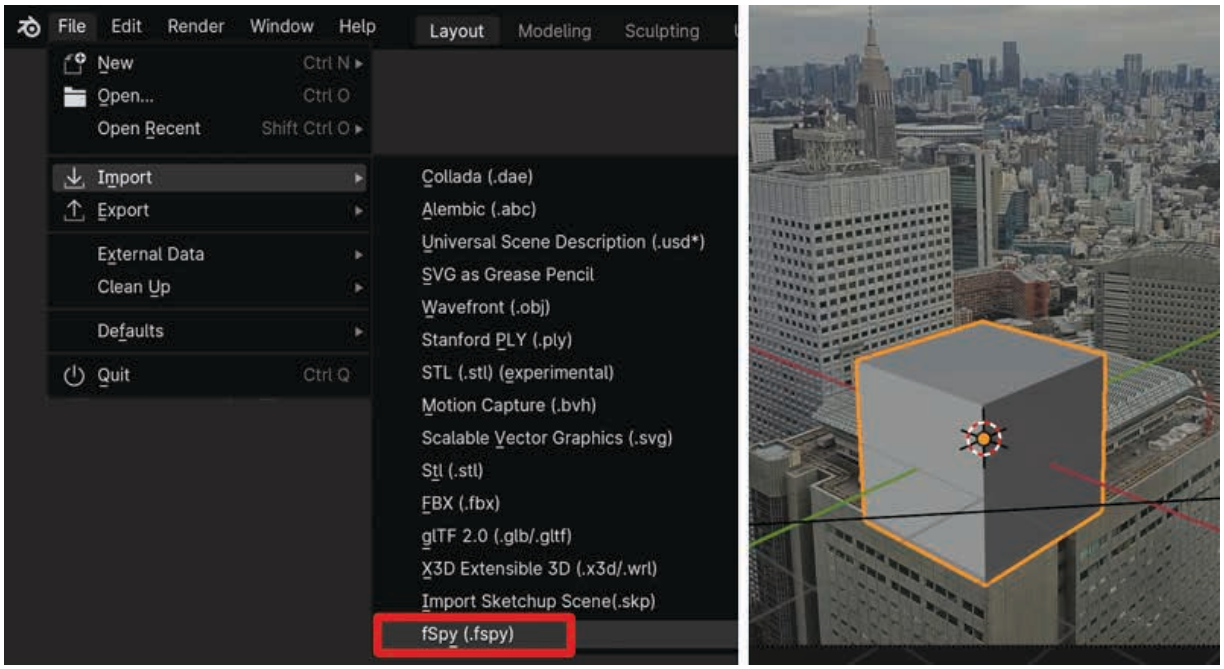
This file can then be imported in Blender.





Because we installed the fSpy add-on, we now have an fSpy import option. I'll import the file I just saved from the stand-alone app by going to **File > Import > fSpy** and choosing the correct file.

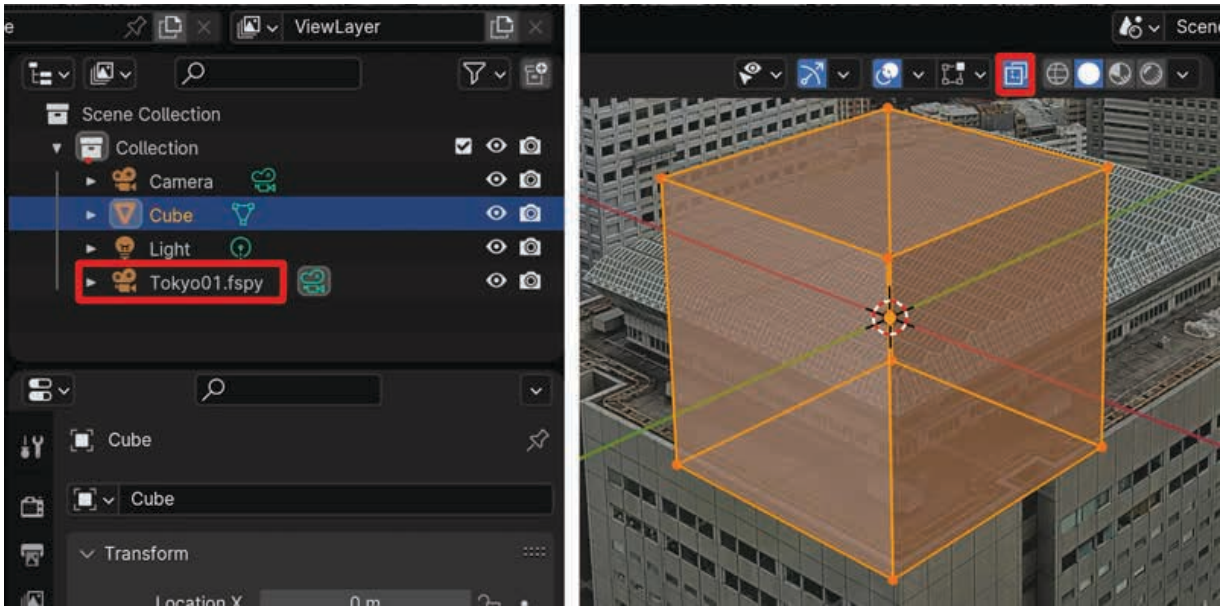
This file contains the photograph as well as a camera. And as you can see, from the Camera view, the Cube aligns to the perspective of the photo. Press 0 on the Numpad to enter Camera view, in case you exit it by accident.



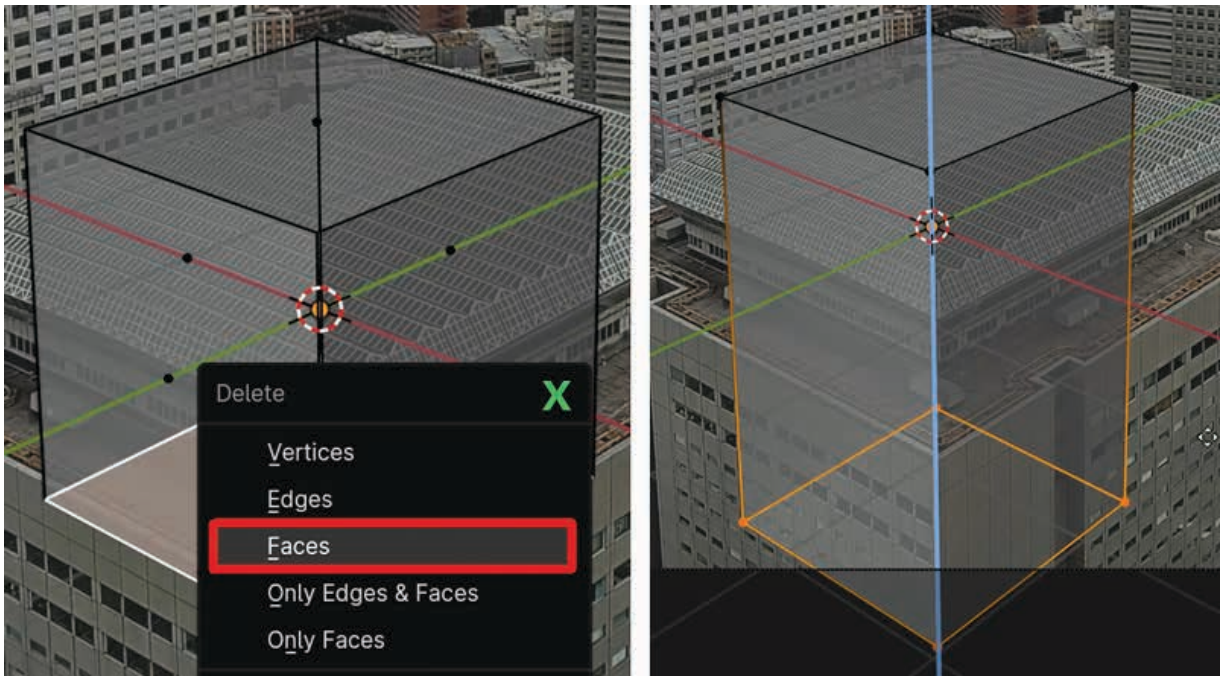
It's a good idea to save this .blend-file at this point.

We still have the original **Camera** from the default Blender scene, which we don't need. So select that one and just delete it.

As you can see the imported Camera has the name of the fSpy project, "Tokyo01.fspy". I'll turn on **X-Ray mode** so I can see the buildings in the photograph through the mesh. This will make modeling easier.

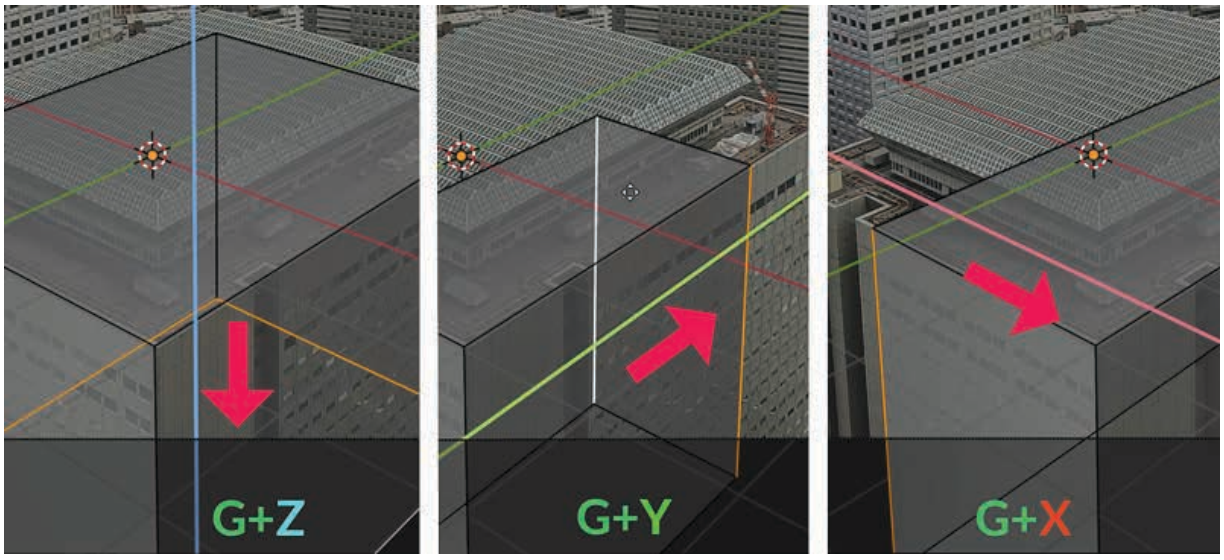


And in **Edit Mode** I'll delete the bottom Face as it won't be visible. Now it's just a matter of moving Vertices, to match the buildings in the photo.



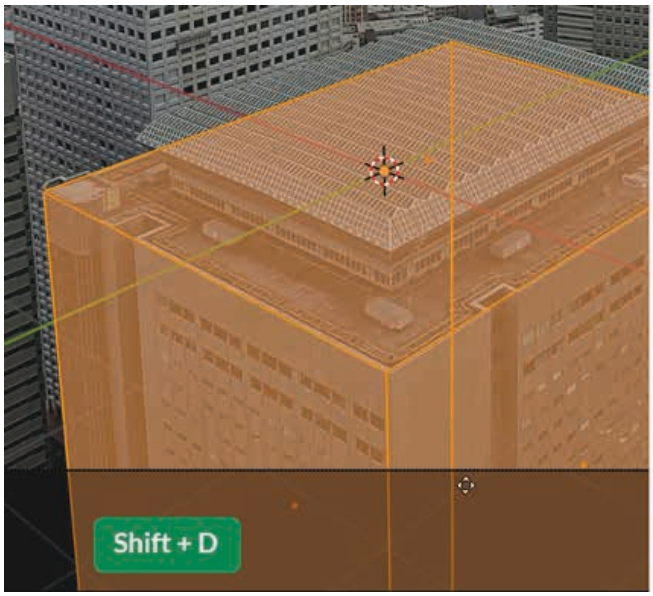
By **holding Alt and clicking**, we can select all these bottom **Edges** and move them down.

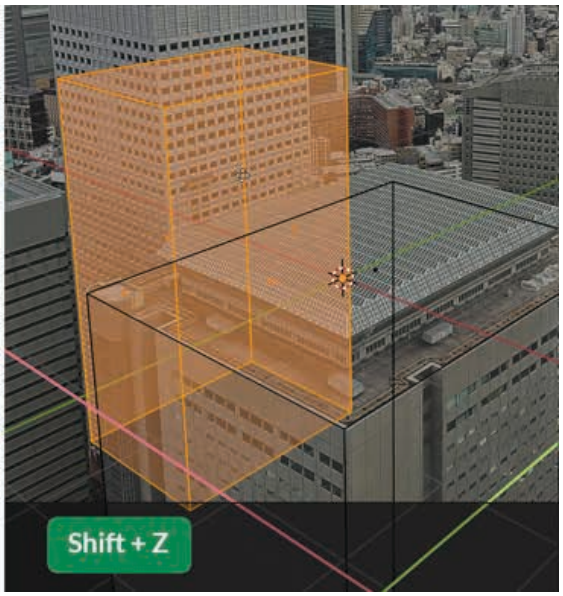
To move things along just one axis, press for example G and Z to move it along the Z-axis. Or G and Y to move along the Y-axis… or G and X to move along the X-axis.



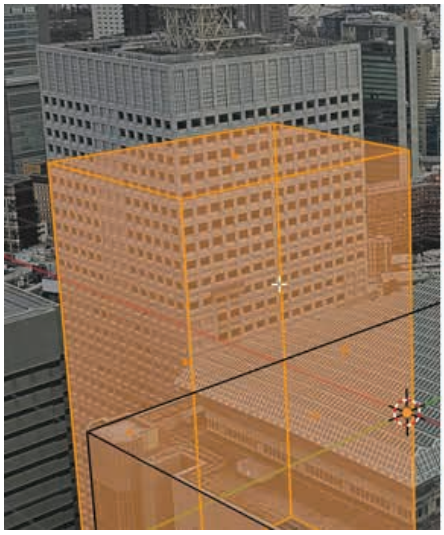
This is the very basic shape of this building, from here on it's just a matter of adding more detail.

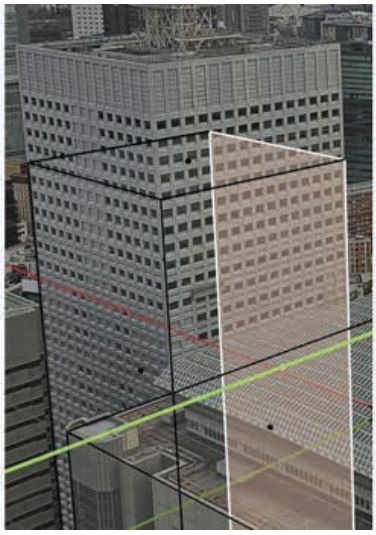
But first let's select all and duplicate with **Shift+D**, for the next building. By pressing G to move and then **Shift+Z**, we exclude the Z-axis so we can move it easily without accidentally moving up or down.

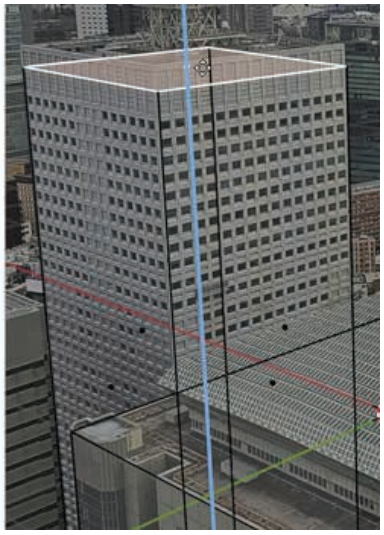




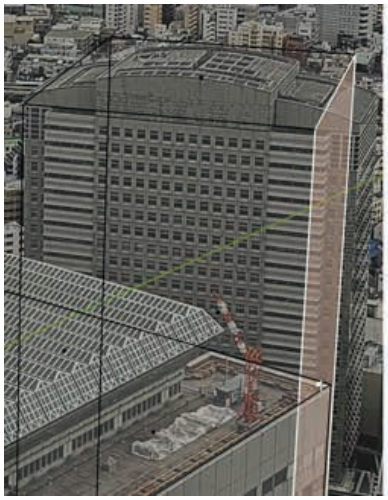
And then it's just a matter of adjusting the shape by moving Faces.

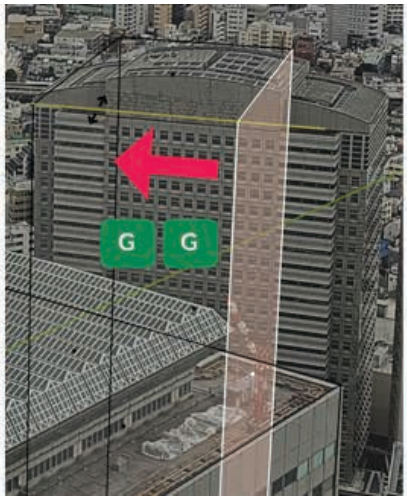


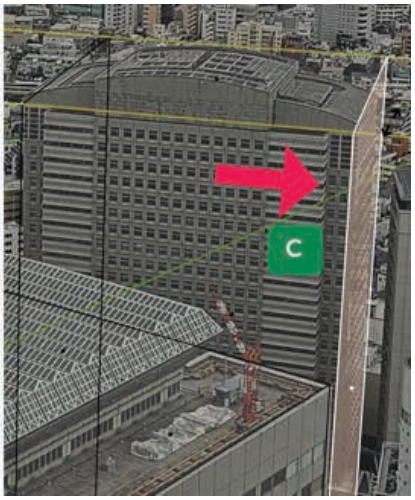




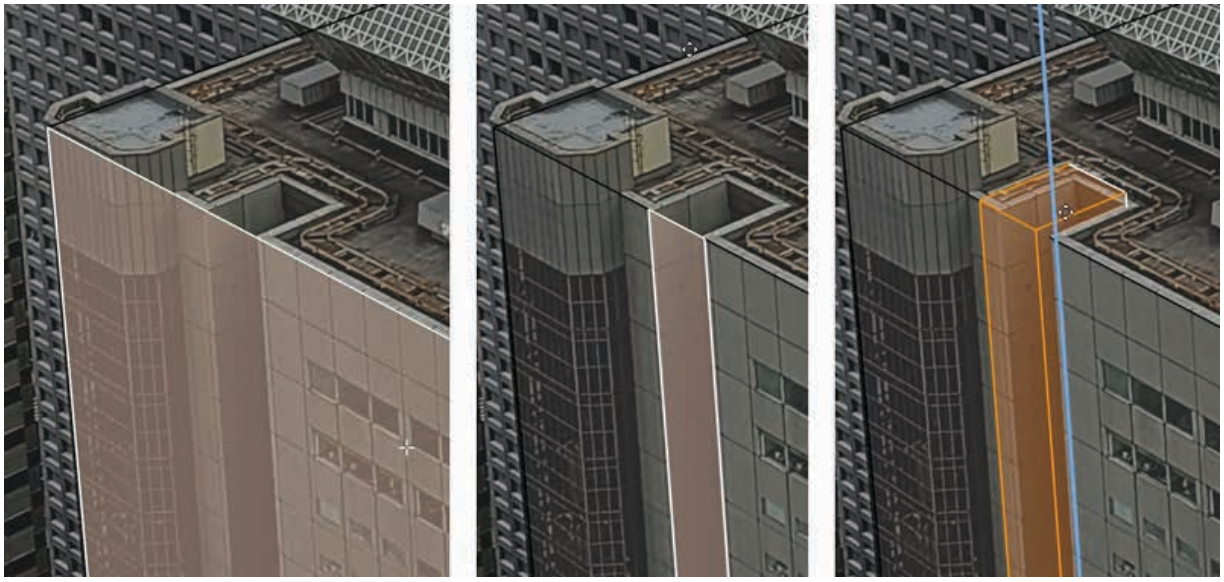
If you **press G twice**, select an Edge to slide along - it will turn yellow - **and then C**, you can slide 'outwards' along that Edge. This can be convenient if you've rotated something in Edit Mode, and need to move along its local orientation.





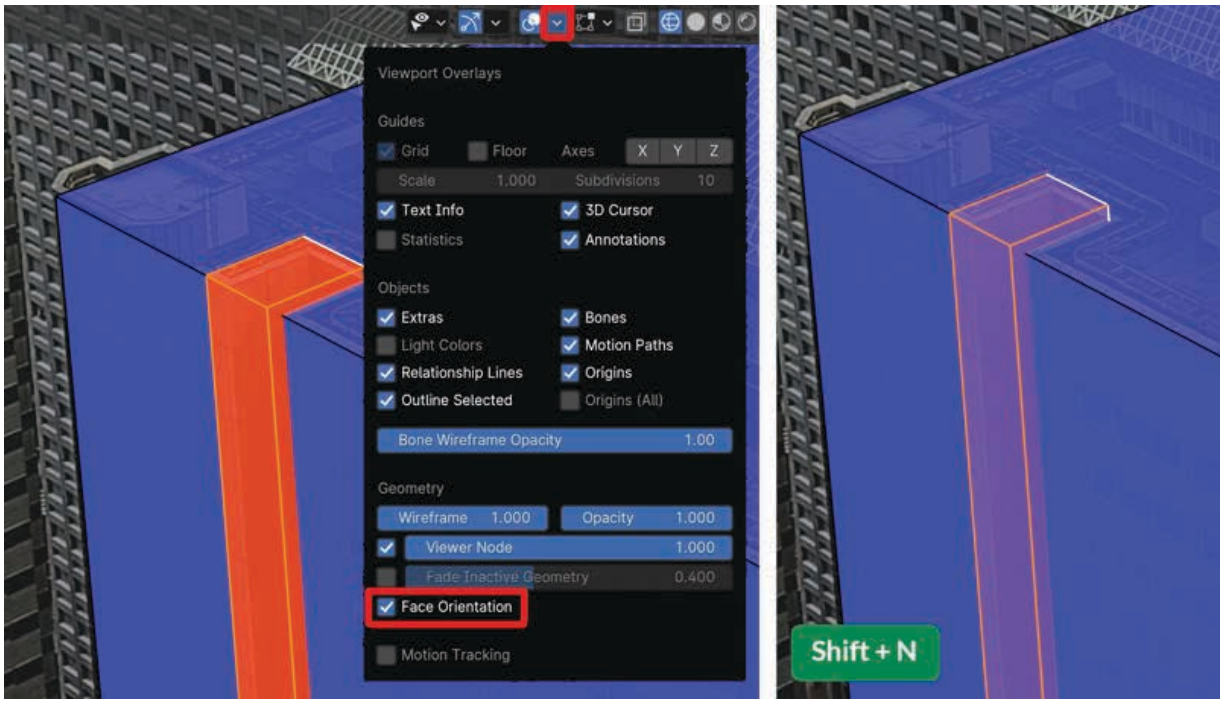


To create these gaps in the side of the building, I'll select and duplicate the **Face** at the side where the gap should be. Then scale it along the X-axis and move it to make it match the width of the gap. Then extrude along the Y-axis with E and Y. Move it slightly so there are no Faces overlapping in the exact same location (that would cause issues later).

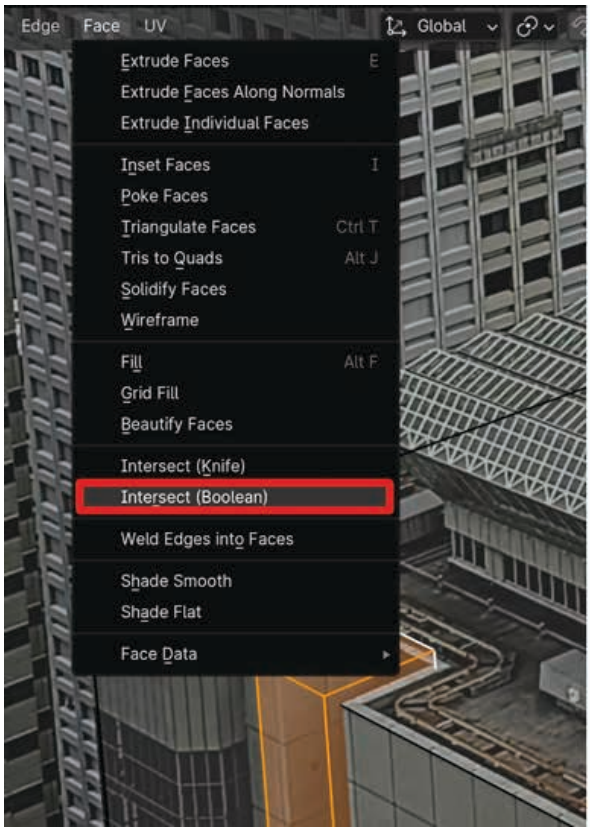


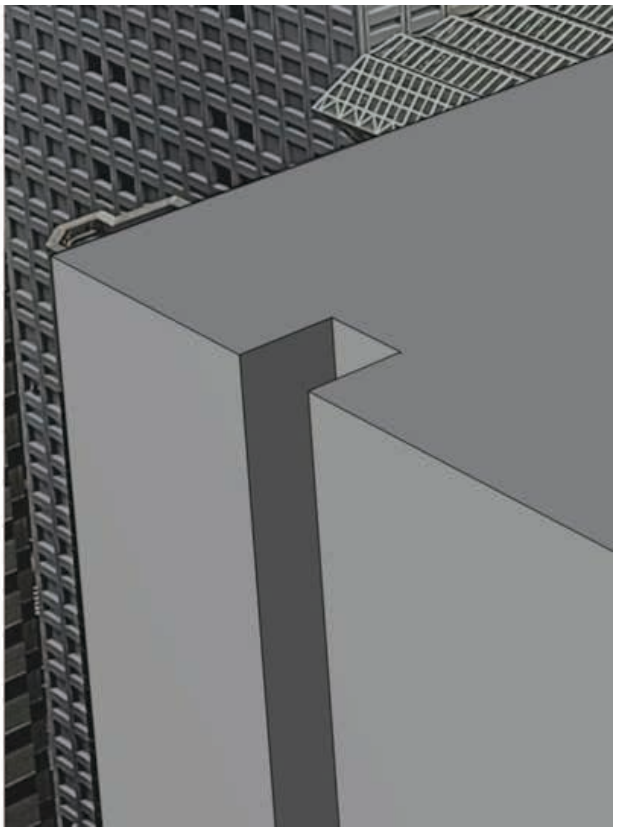
A quick check of the **Face Orientation**, shows that the Faces of this geometry are Flipped (shown in red).

Pressing Shift+N recalculates the normals so they're no longer flipped. I'm planning to do a boolean cut, and flipped Faces can cause trouble.



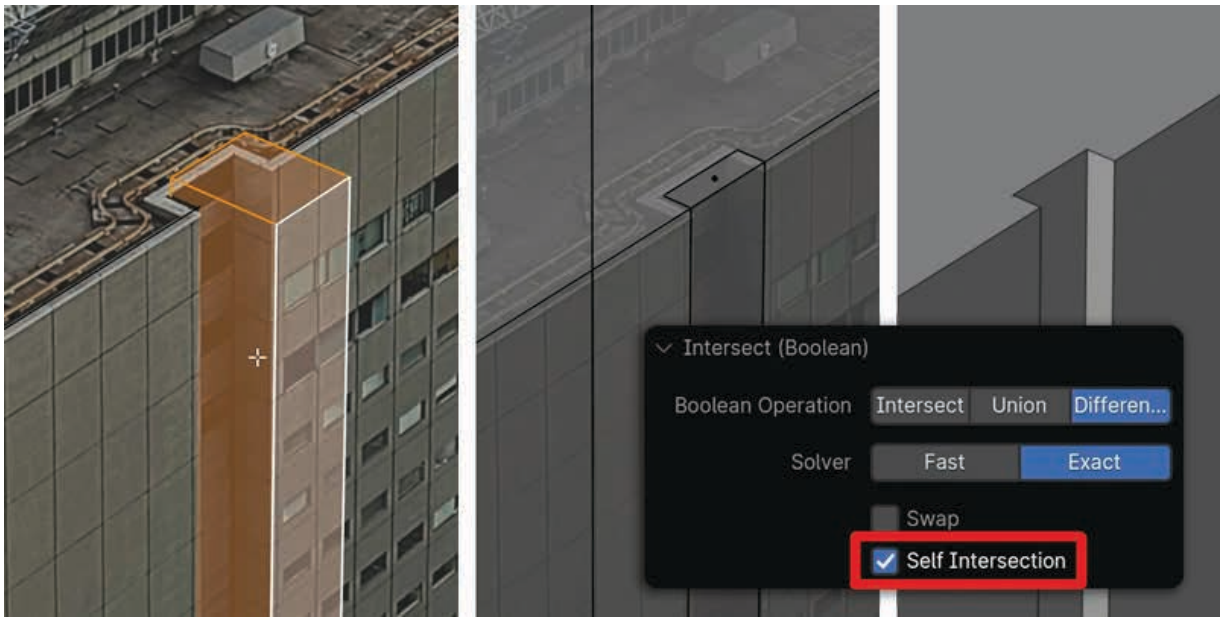
Now **with the extruded Faces selected**, go to **Face > Intersect Boolean.** This cuts a nice chunk out of our building.





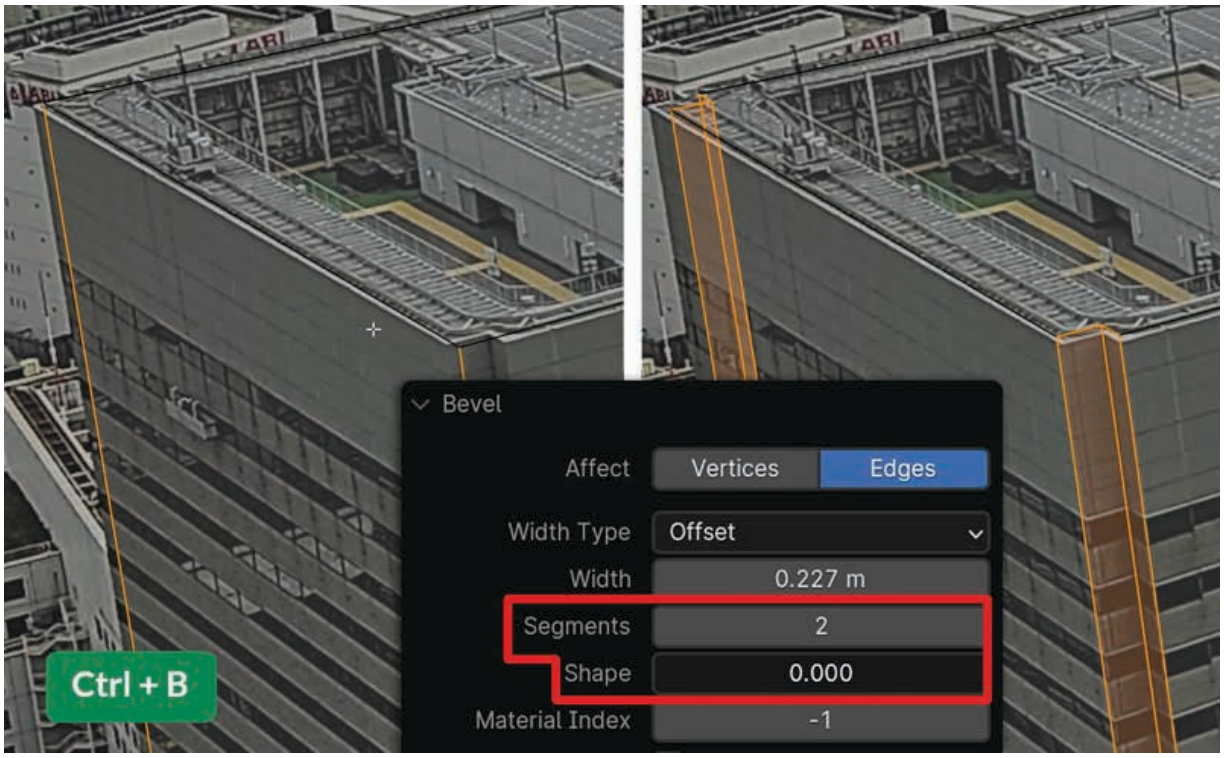
We can repeat the same steps on the other side. The benefit of duplicating a Face for this purpose, is that the Face is already perfectly aligned with the side we want to cut out of.

It seems the boolean cut didn't work on this side (see middle picture). If that happens, you can try a different **Solver** method or checking **Self Intersection,** which solved the problem in this case.

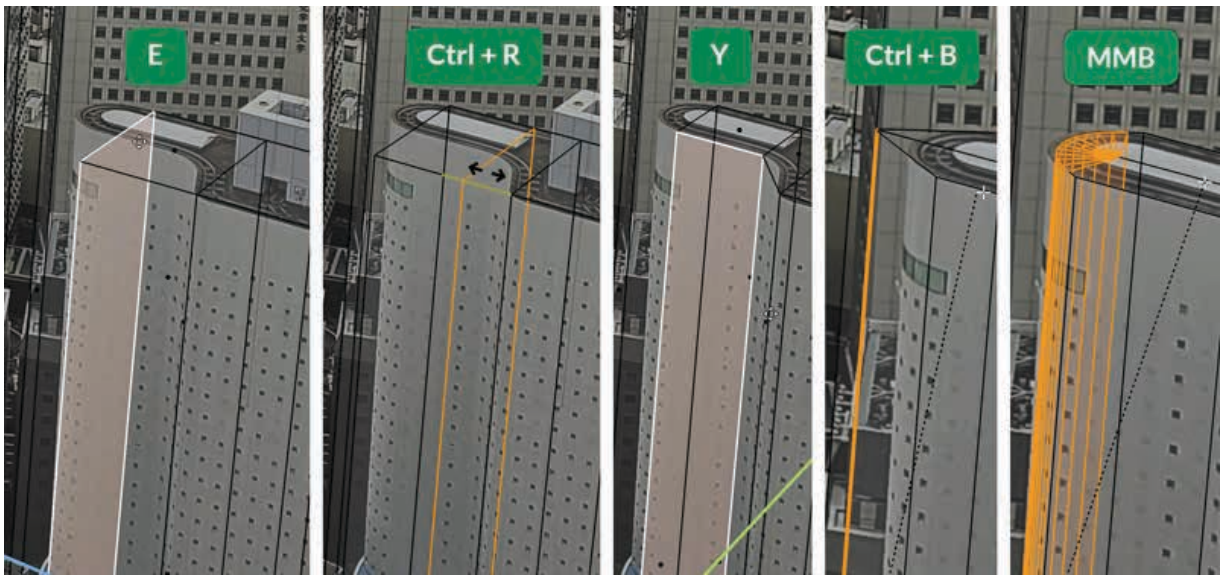


Some minor adjustment, by moving some vertices here and there, may be necessary. Next let's do the corners of this building. **Hold Shift** and select the Edges.

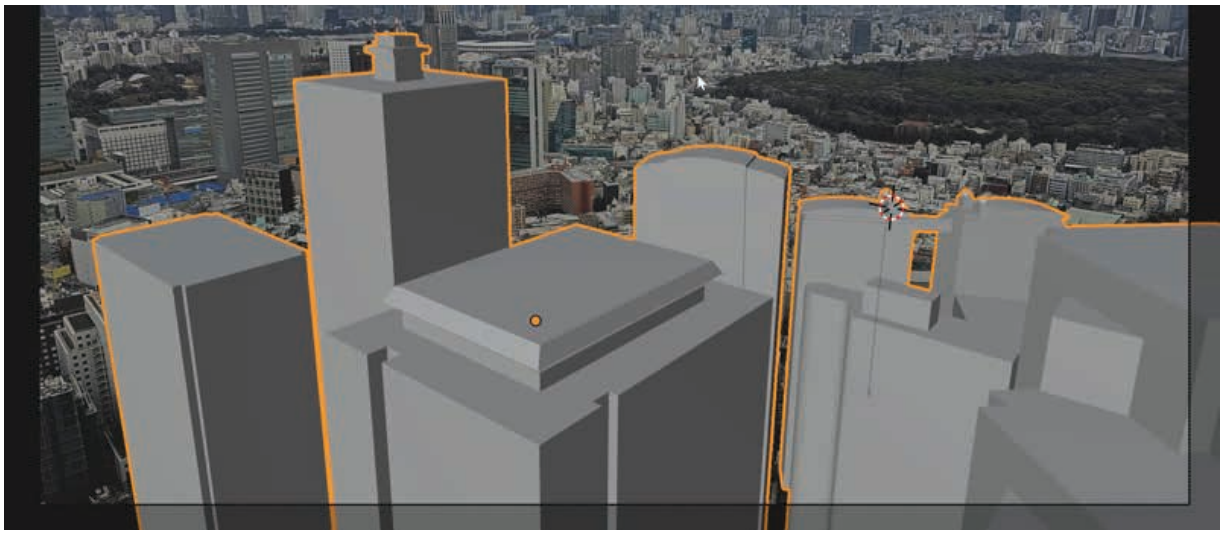
Press Ctrl+B to bevel, and **scroll the mouse wheel up** to add one segment. Changing the **Shape to 0** gives us the exact shape we need.



We can fine-tune it if necessary. I won't explain the rest of the modeling step by step, as it's mostly very basic and repetitive stuff. Just lots of extruding, adding more Edge loops where needed, moving Faces, and beveling here and there to add some rounded parts.

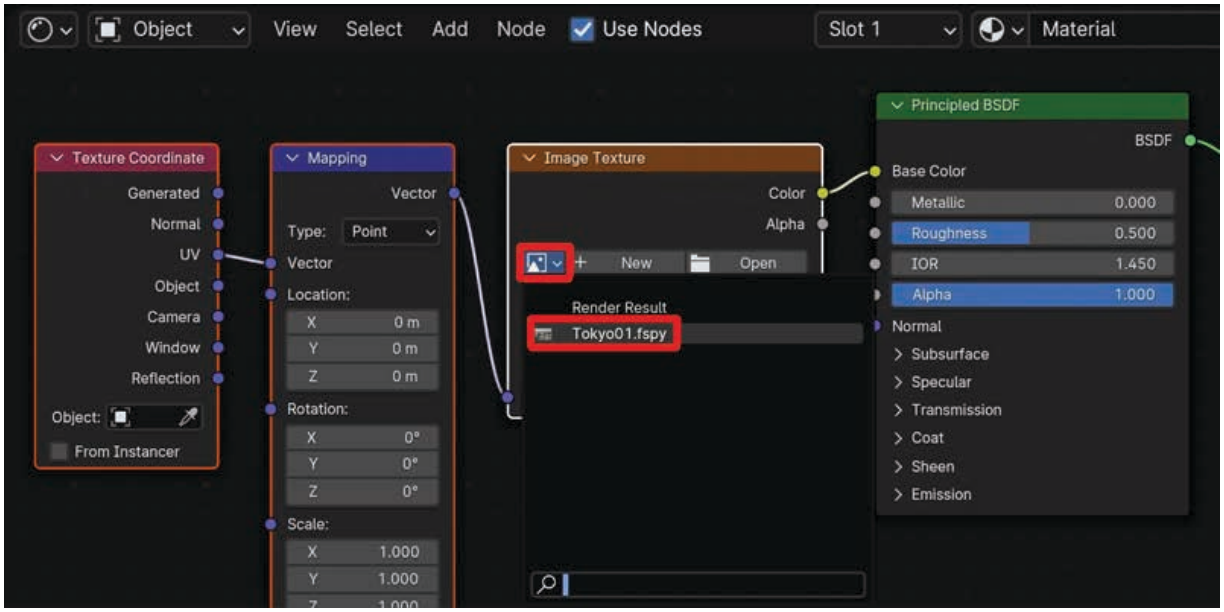


Honestly, a lot of these things are just simple cubes. You can get away with a lot when using projection mapping like this, because the texture has so much information. After some more basic modeling, we have a bunch of simple buildings.

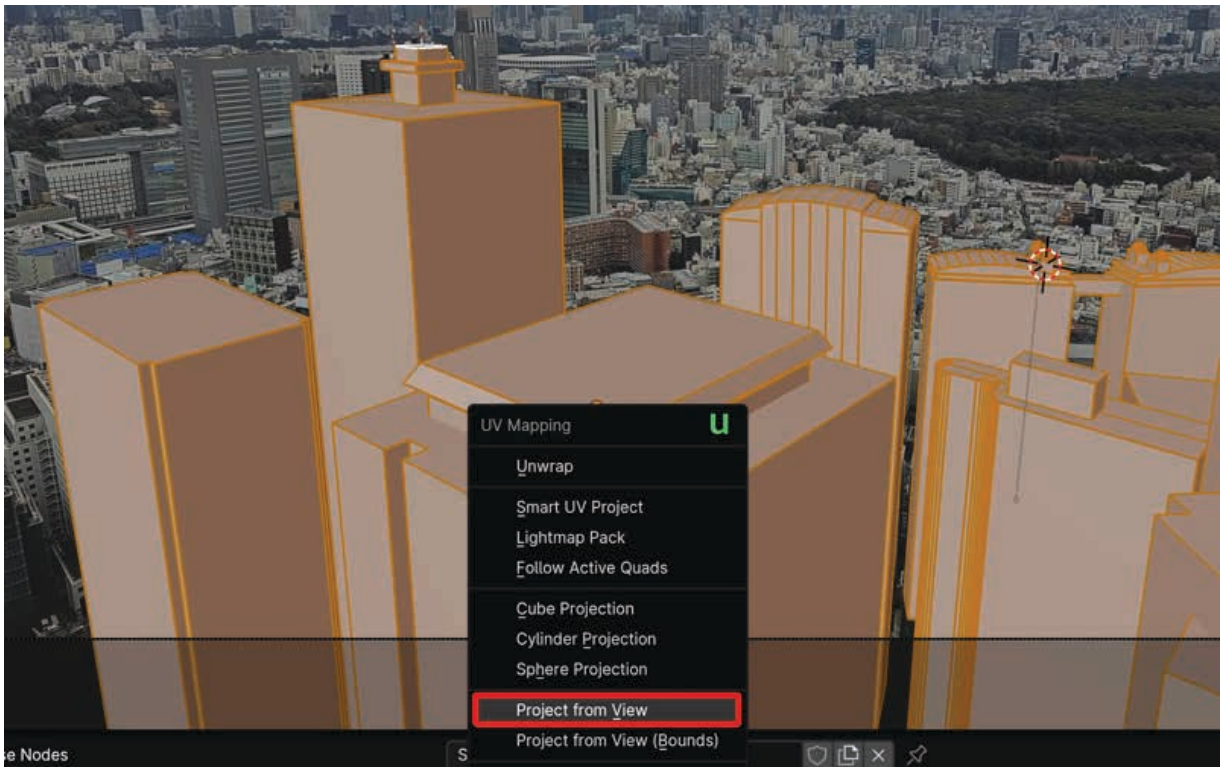


Next we need to add a material. I'll open a **Shader Editor** for that. Select the Principled BSDF and **press Ctrl+T** to add the necessary nodes *(Note that you'll need to enable the add-on Node Wrangler in Preferences to have this option).*

The photo is already loaded with the fSpy file, so we can choose it as the texture.

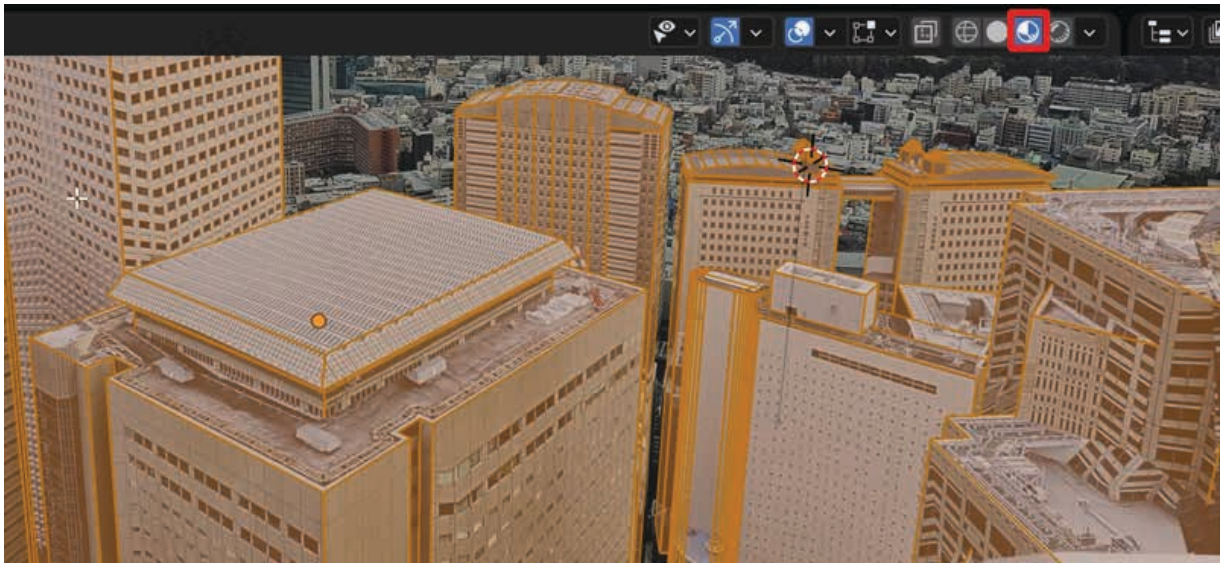


Select all in Edit Mode, **press u and choose Project from View**. This projects the texture from our current view, the Camera view.

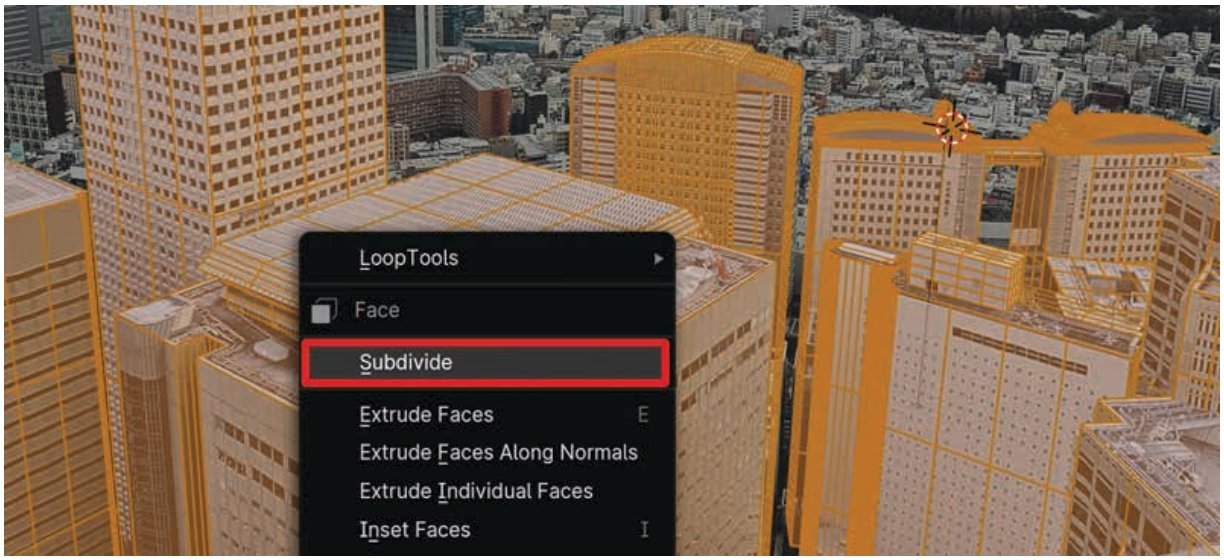


And switch to **Material preview** to see the result.

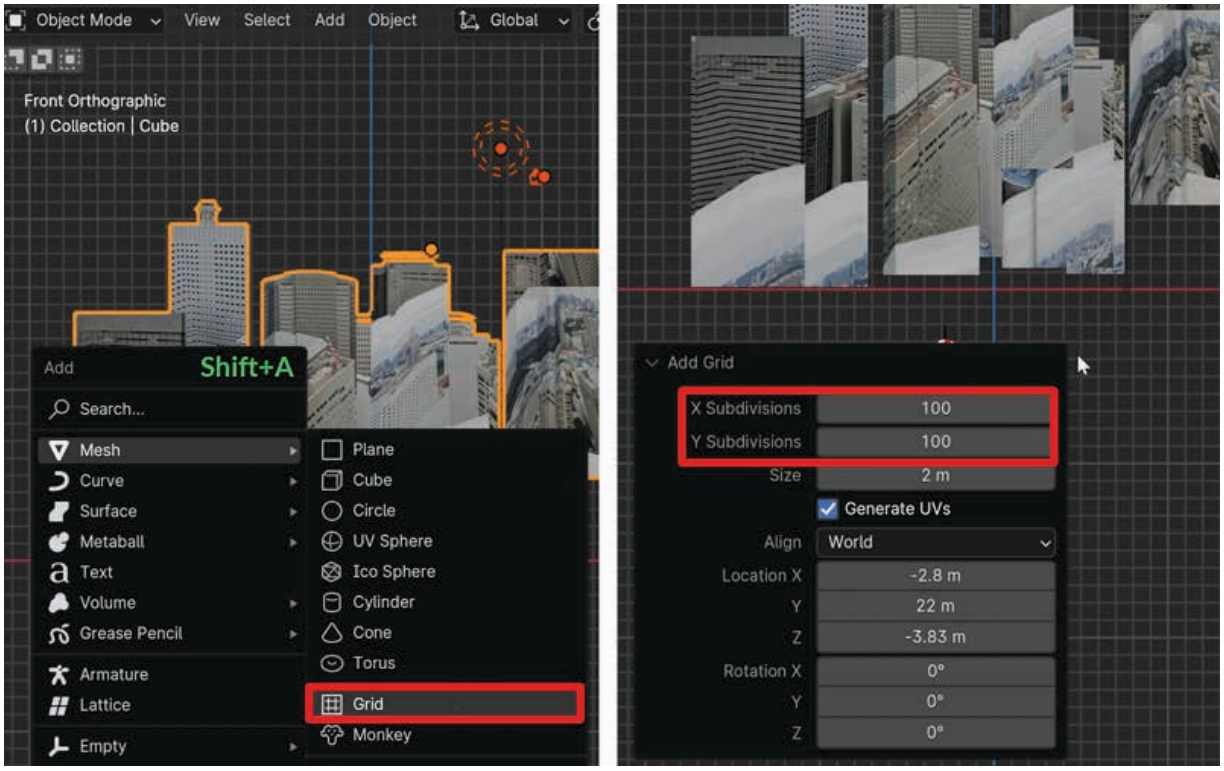
It kind of worked, but as you can see it's all wobbly and distorted. That's because there's not enough geometry to support the texture.



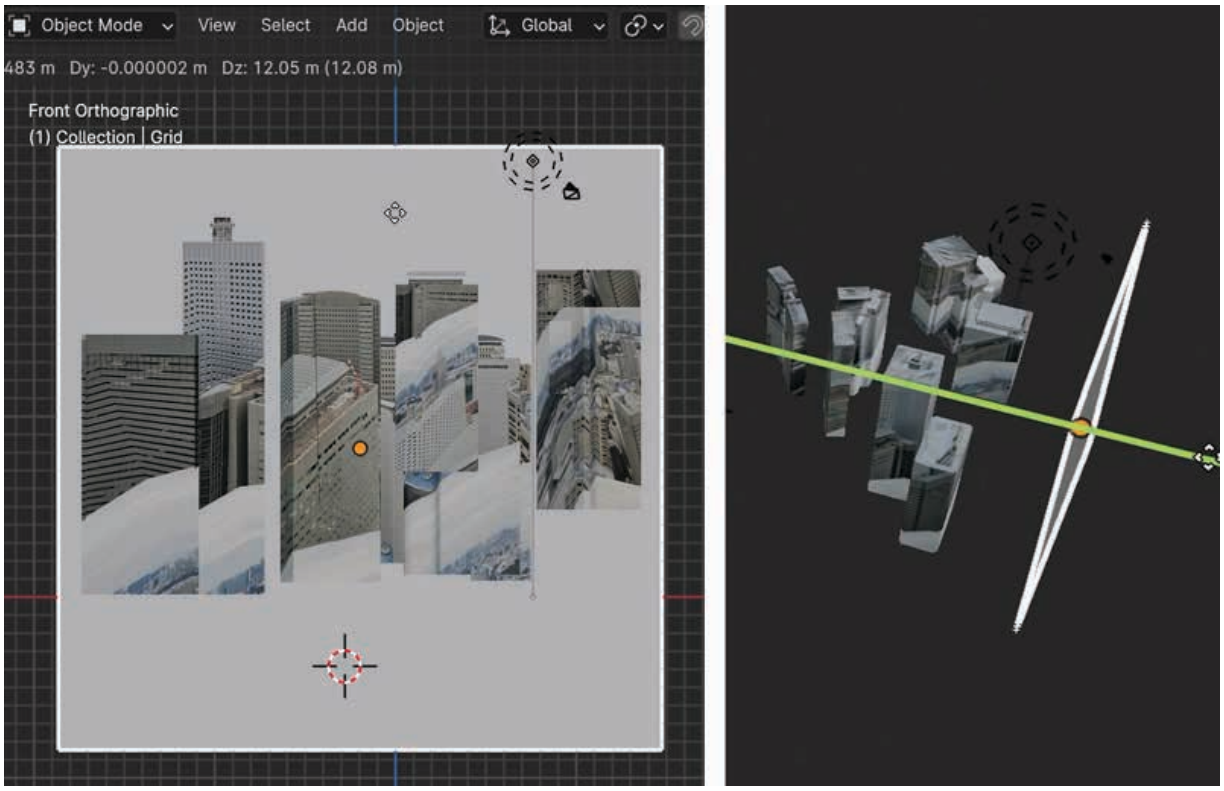
We can make this better by right-clicking and **Subdividing** a bunch of times, although in this case that adds a lot of unnecessary geometry in some places. So let's not do that (in some cases this can be a good solution, though).



Instead, let's "slice" the geometry. To do that, we'll add a **Grid**. And increase its resolution to **100 by 100 Edges**.

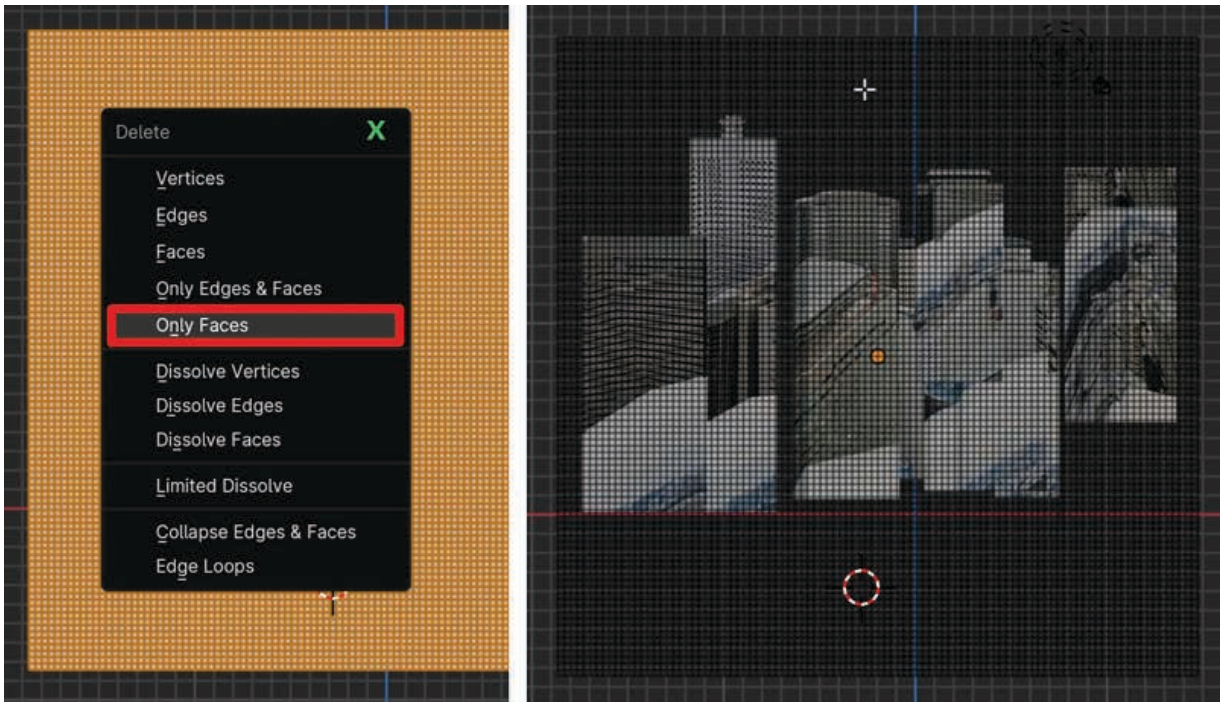


Rotate and scale it so it encompasses the buildings. And move it so it's not overlapping or inside the buildings.

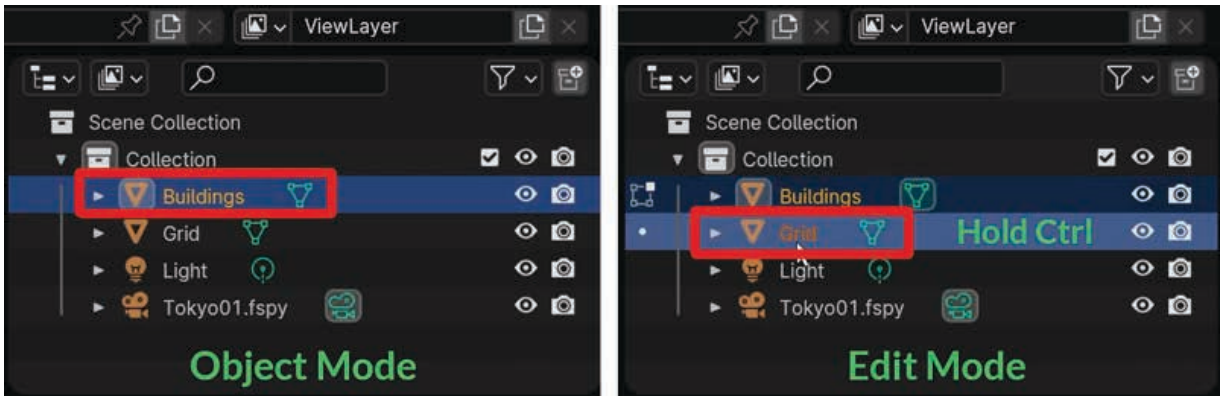


Go to **Front orthographic** view.

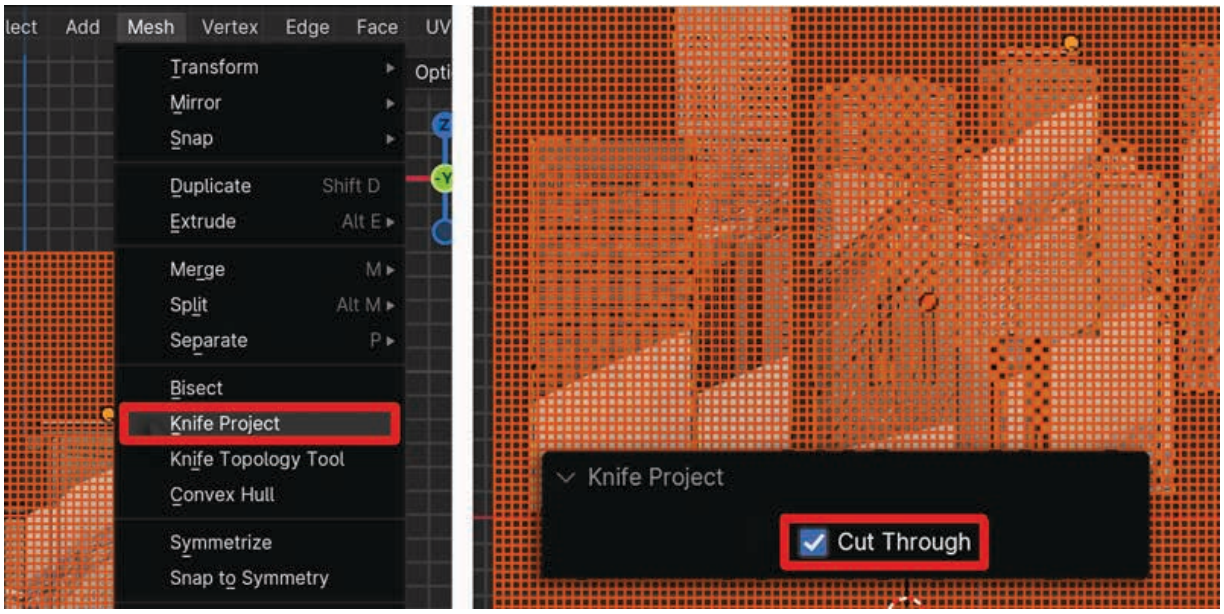
In Edit Mode press A to select all and press X, then choose **Only Faces**. Now we just have this grid of Edges.



Rename the buildings geometry to Buildings, to keep things organised. Now with the buildings selected in Object Mode, switch to Edit Mode. **Hold Ctrl** and in the outliner, select the Grid.

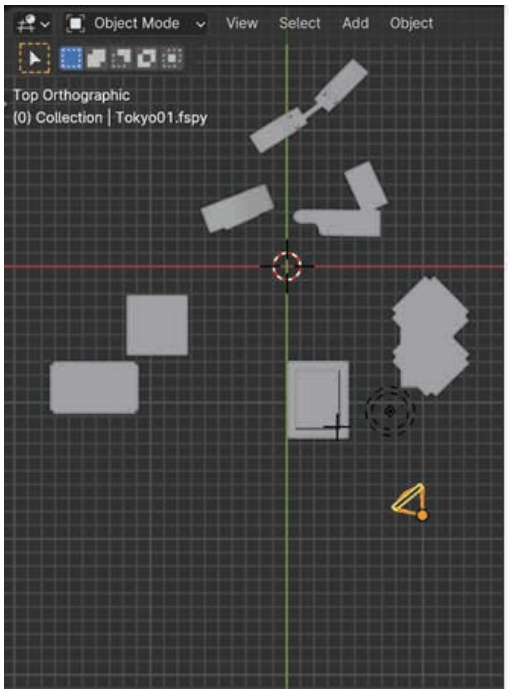


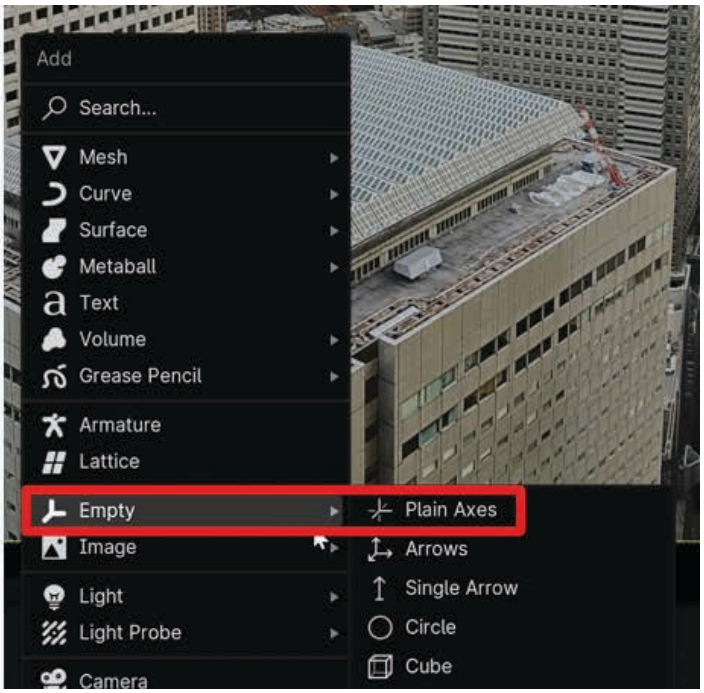
Go to **Mesh > Knife Project**. And then make sure **Cut Through** is enabled. You may have to wait a few seconds after enabling Cut Through, for it to work.



Now the buildings are 'sliced', giving them all enough geometry so that the texture is not distorted.

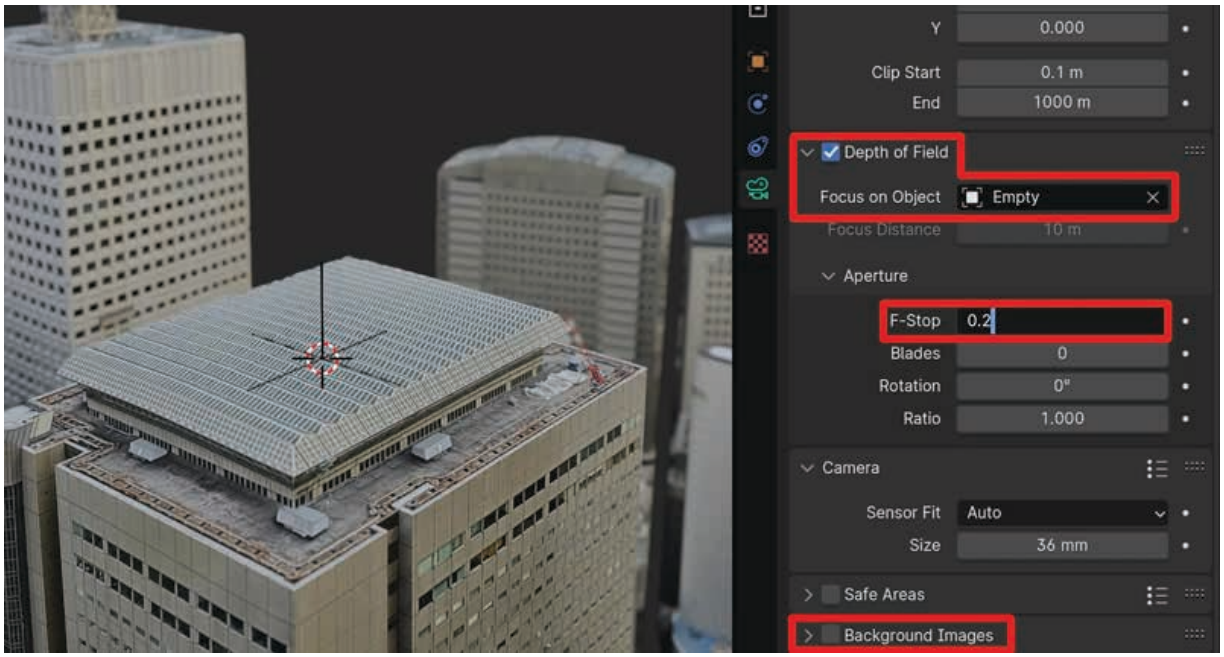
We can hide or delete the Grid now. Now if we carefully **move the Camera**, you see that we get a subtle camera move. And we can add an **Empty** to use as the **Depth of Field target**.



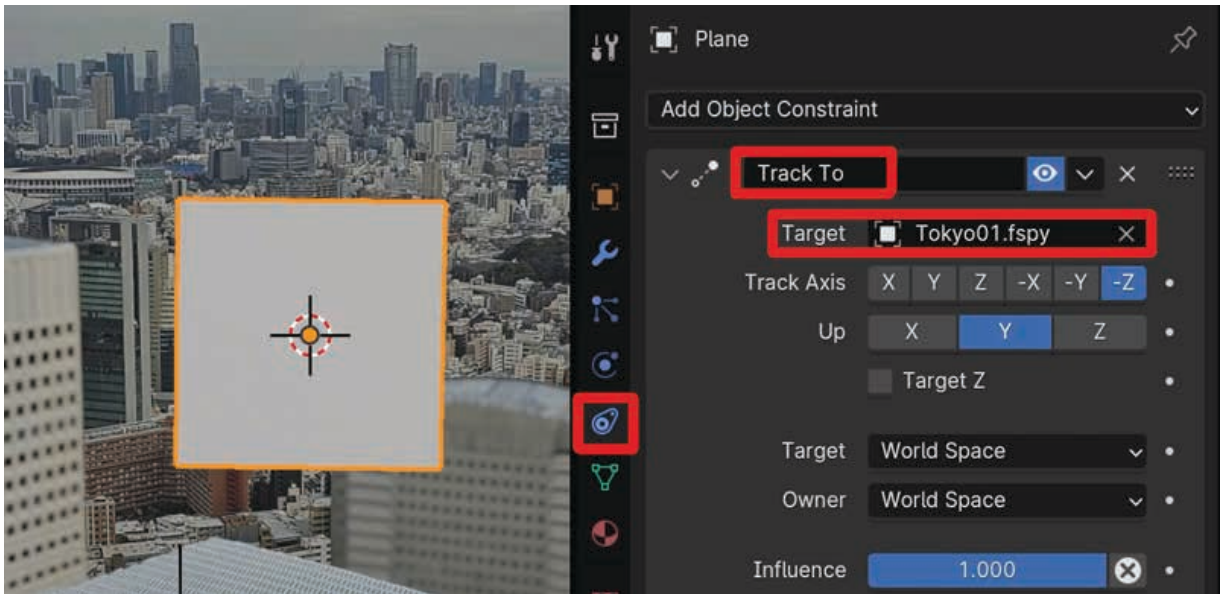


This gives us some unrealistic but nice looking **shallow depth of field**.

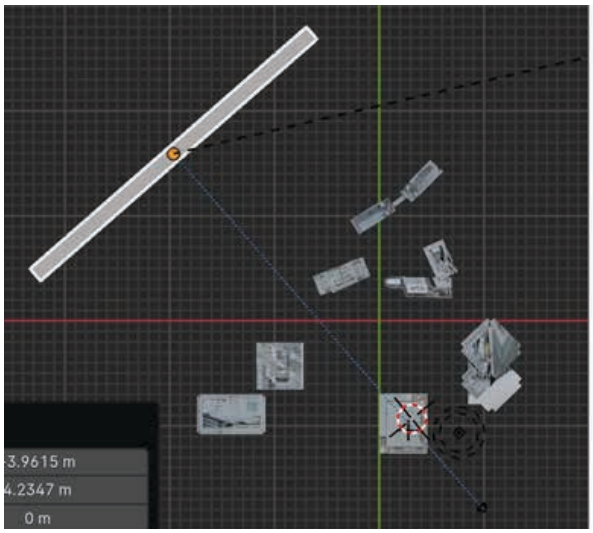
We can disable the **Background Image** now, as we no longer need it for reference. And it's not influenced by the depth of field anyway, which is not realistic.

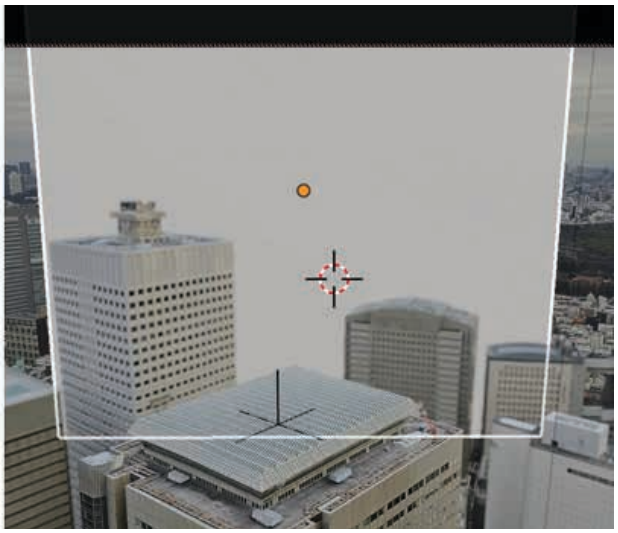


What we can do instead, is add a **Plane** and add the same texture to it.
To make sure it's always facing the Camera, add a **Track To constraint** to it. Choose the Camera as its target.

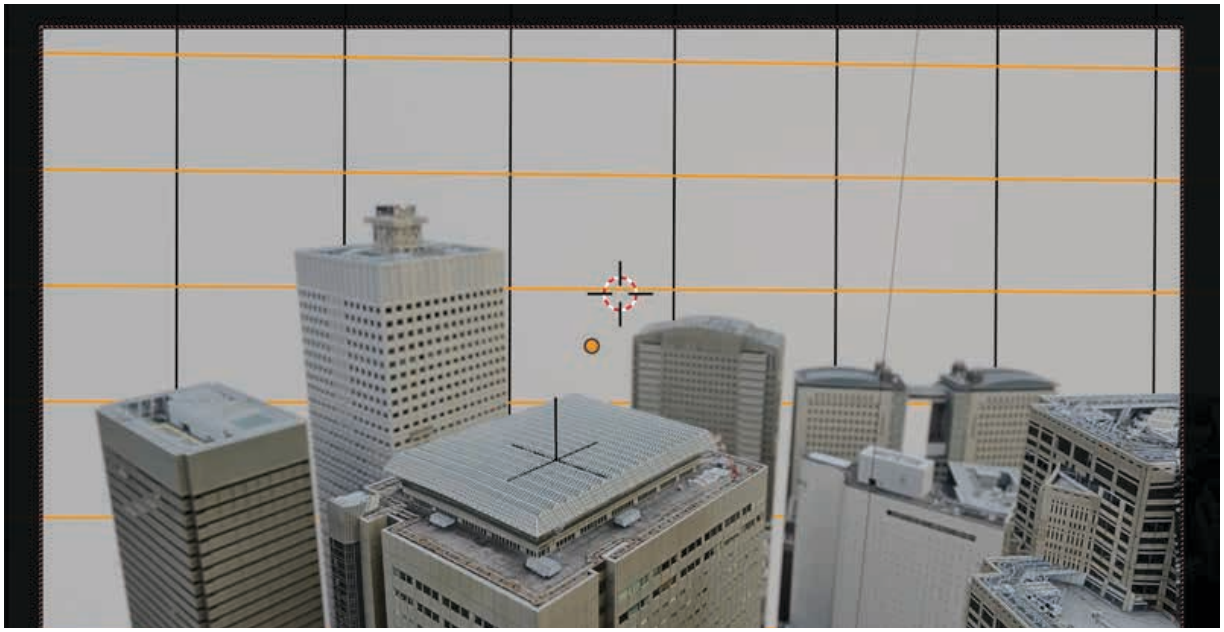


Then move the Plane behind the buildings and scale it up as necessary to fill the frame.

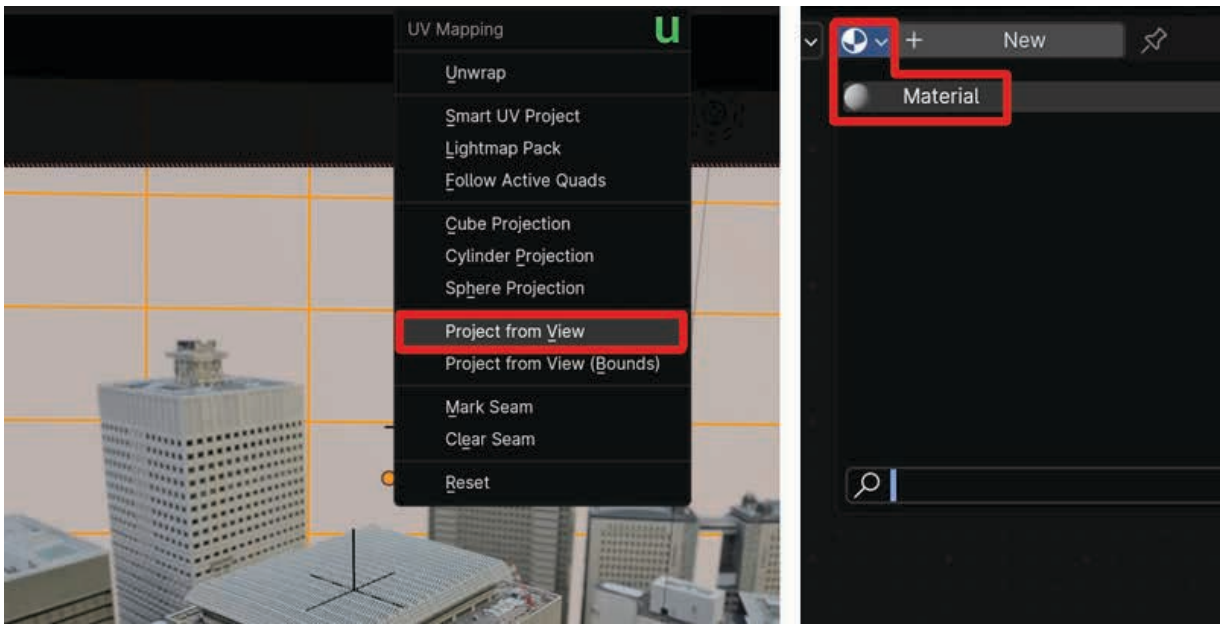




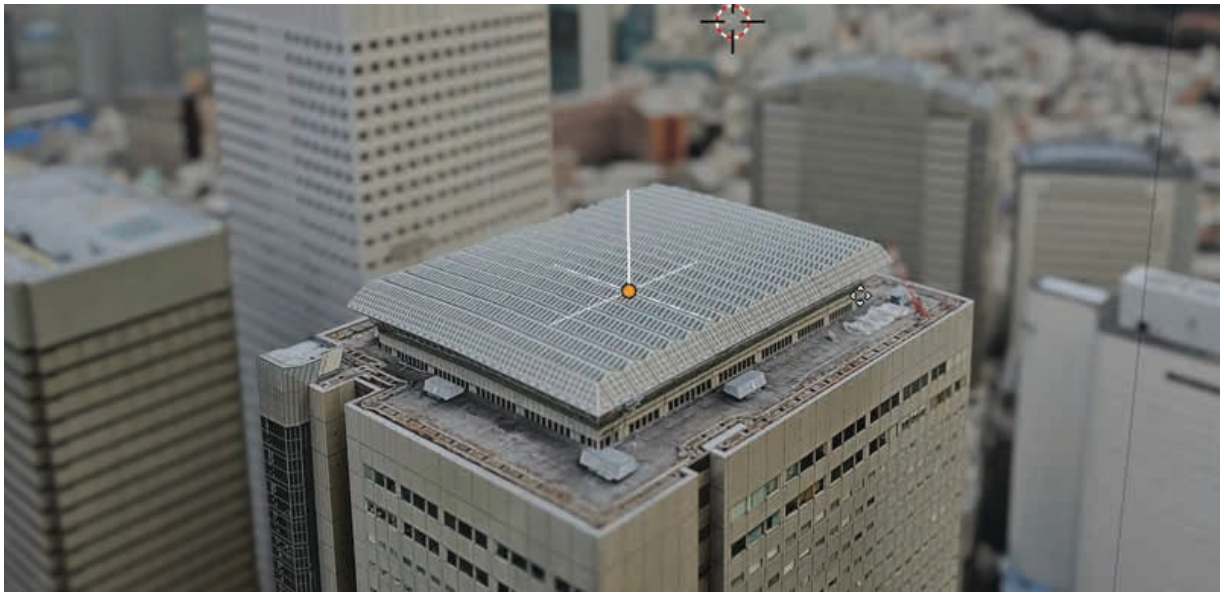
Add some Edge Loops (Ctrl+R) so that the texture won't be distorted.



And just like with the buildings, select all, **press U** and choose **Project from View**. Use the same material as the buildings.

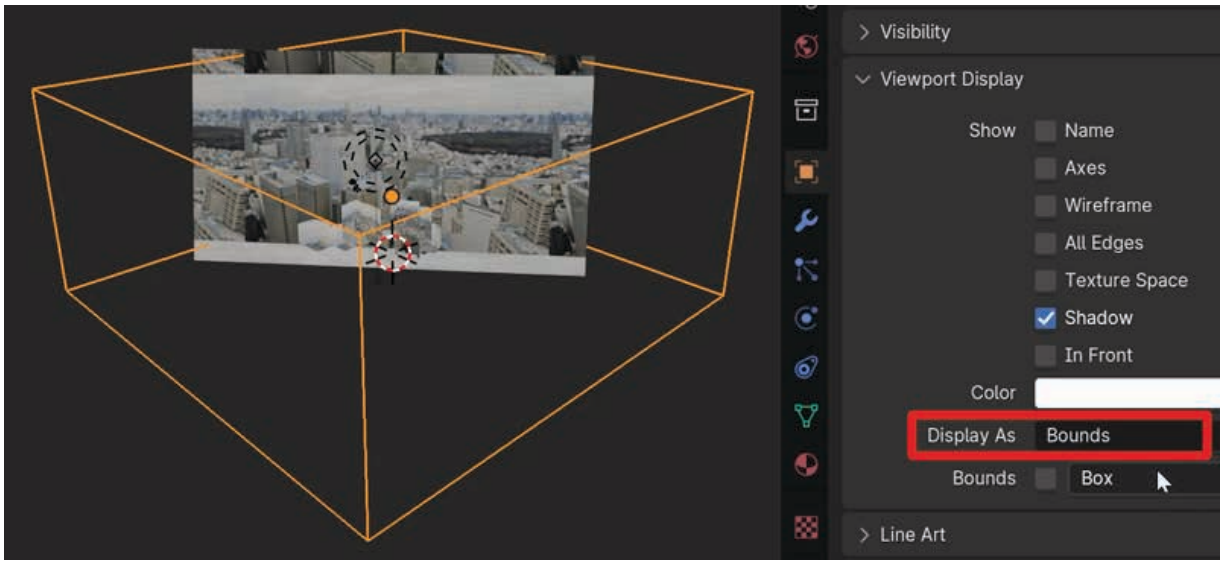


Now the far background is also affected by the camera depth of field. And we can move the Empty to change the focus point.



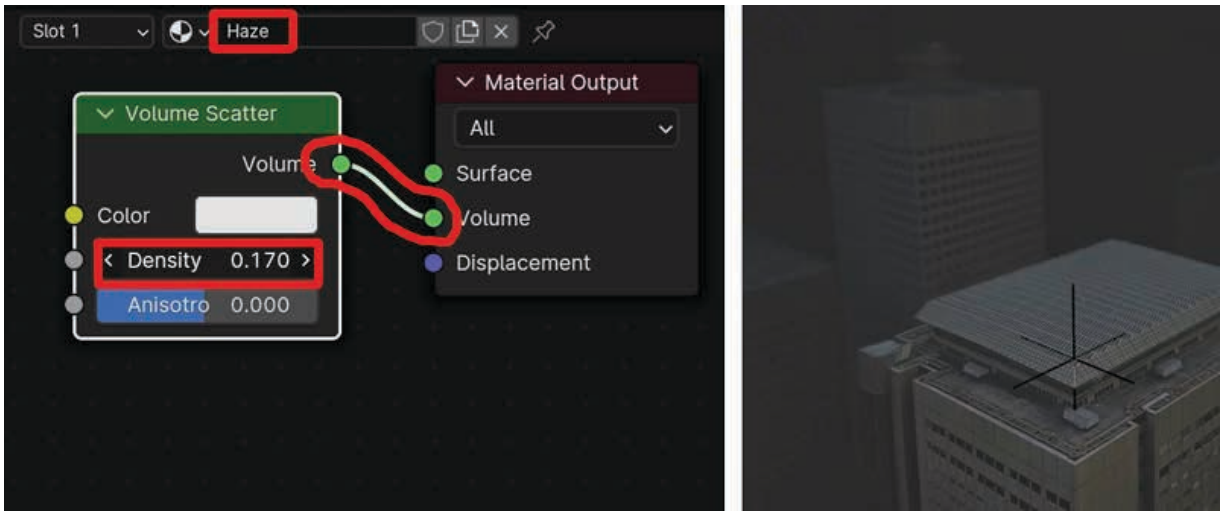
Now I'd like to add some **volumetric haze** to make it look more interesting.

To do that I just add a **Cube** around the scene. Scale it down to make sure it only envelops the scene, it doesn't have to be any bigger. And for ease of use I'll set it to **Bounds** in viewport display.



Give it a material with an appropriate name.

Remove the Principled BSDF and add a **Volume Scatter node** instead. Connect it to the Volume input of the material. The **Density** value controls how thick the haze is. The default value is too high, so you'll need to lower it to your liking.

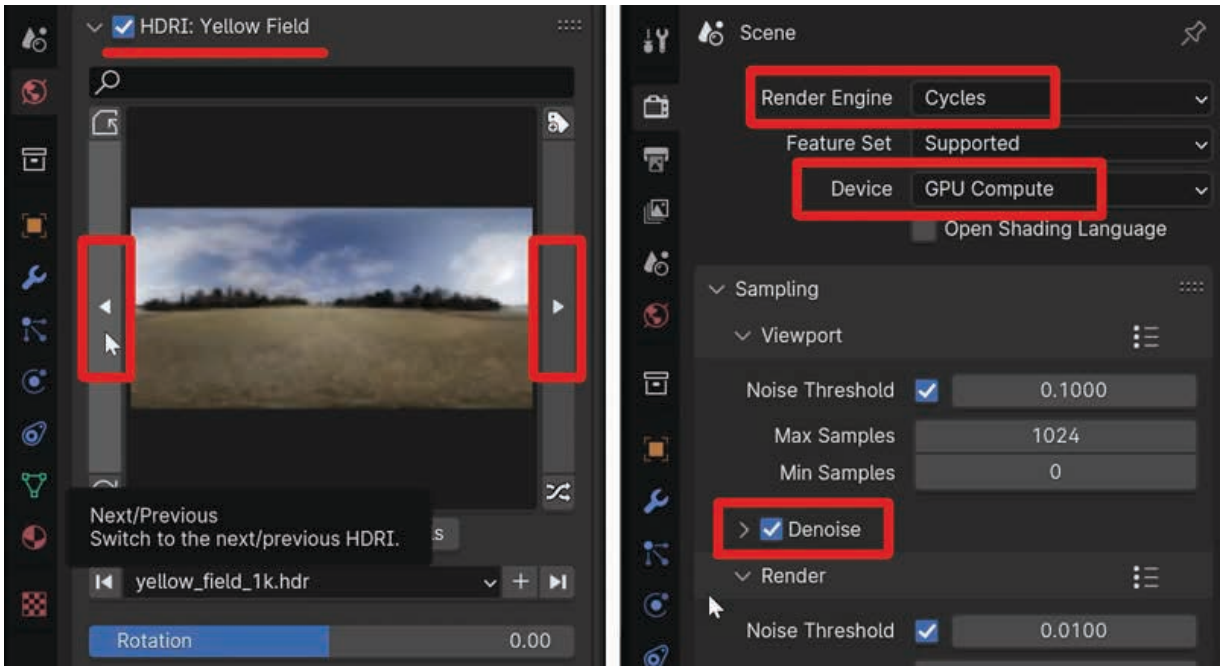


I'll add an HDRI using the Gaffer add-on.

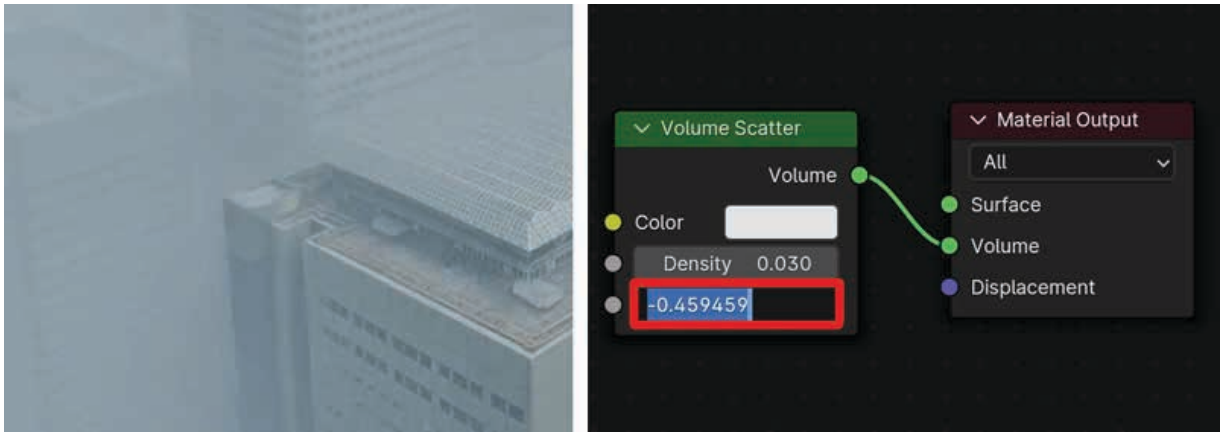
Of course all this will look much better in **Cycles**. And it will be faster if I use my **GPU** and enabled **Denoise**.

You don't have to use an add-on to add an **HDRI**, you can just do it the old-fashioned way.

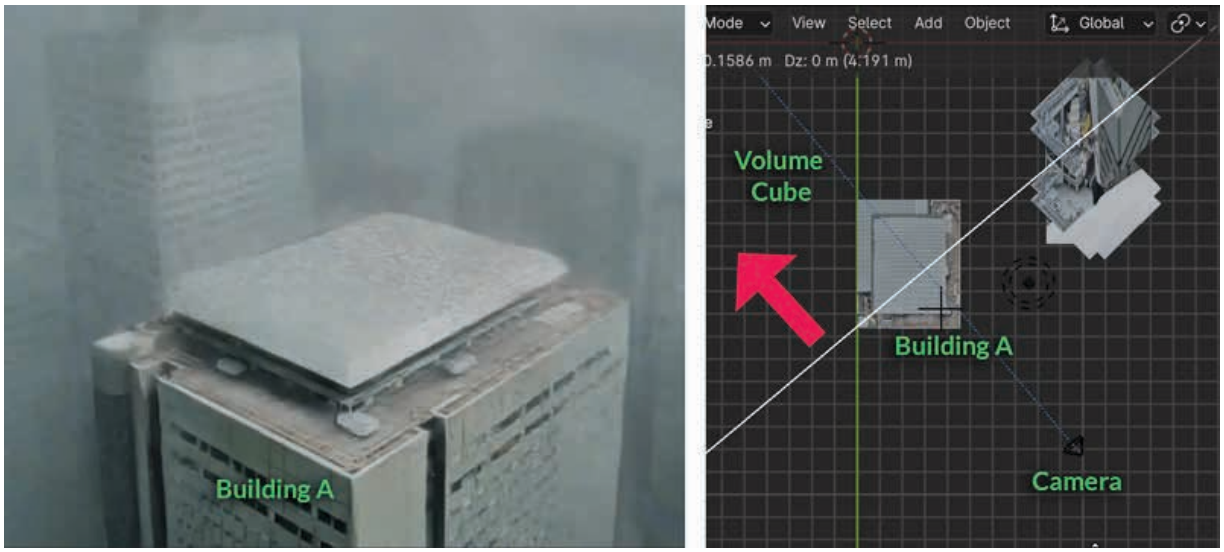
However, I really recommend Gaffer (made by Greg Zaal, of Polyhaven fame). It's a lot of fun to use. There's a free and paid version (so you can try it out - but if you like it, do consider supporting the developer). The many HDRI's of Polyhaven are also free.



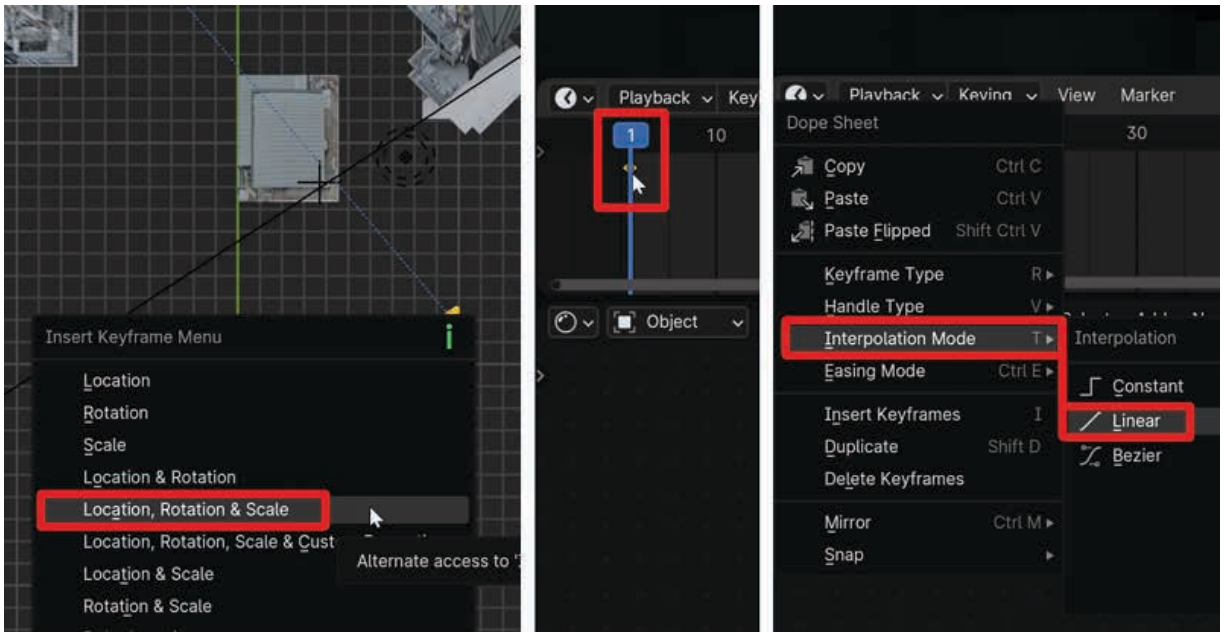
To control the look of the haze, we can use the **Anisotropy** value as well. I think it's a lot of fun to play with these kinds of values, as you quickly get different looks.



However you can also just move the **volume cube**. If you place it just before the Camera like this, the buildings nearest the Camera will have a bit more contrast.



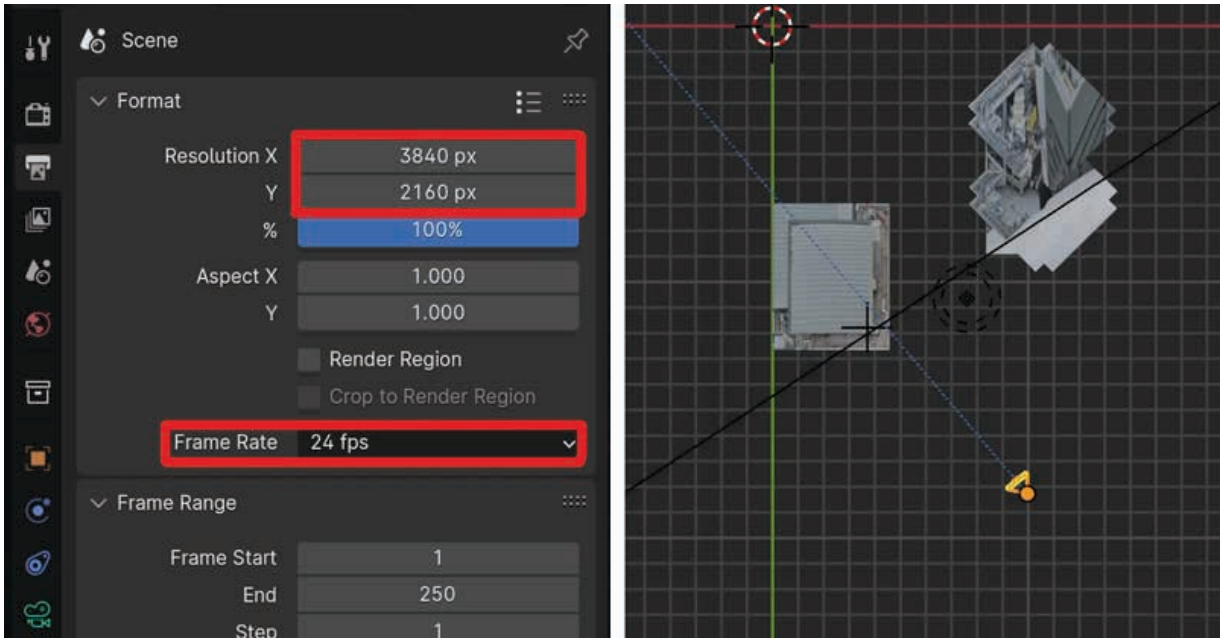
Let's open a **Timeline** so we can animate the Camera. Select the Camera and add a keyframe at frame 1 by **pressing i** and choosing **Location & Rotation** or Location & Rotation & Scale. Right-click on the keyframe and set the key to **Linear interpolation** so the movement is linear instead of easing in.



I'll change the resolution to a more standard 4K format. When using fSpy, it will set the scene to a strange resolution. 4K is **3840 x 2160 pixels** (1920 x 1080 multiplied by 2).

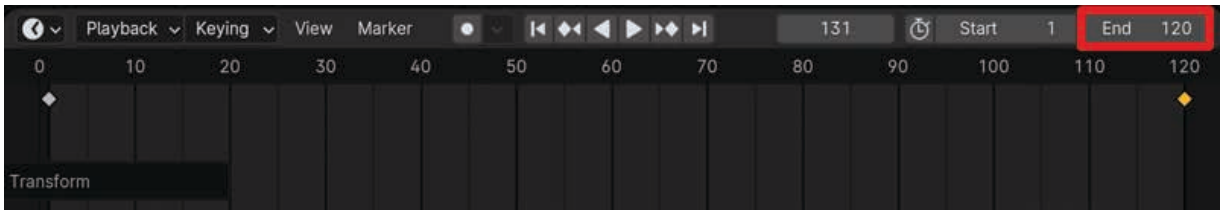
And then we can go to a later frame (frame 120 for example) and move the Camera to a new position by pressing G. By holding Shift, it's easier to make a subtle change in the location of the Camera. I recommend doing this in the **Top Orthographic view**.

You can't go too crazy with the camera position before you start seeing weird stretched parts of the texture, so it's best to keep the camera move subtle. **Set another keyframe**.

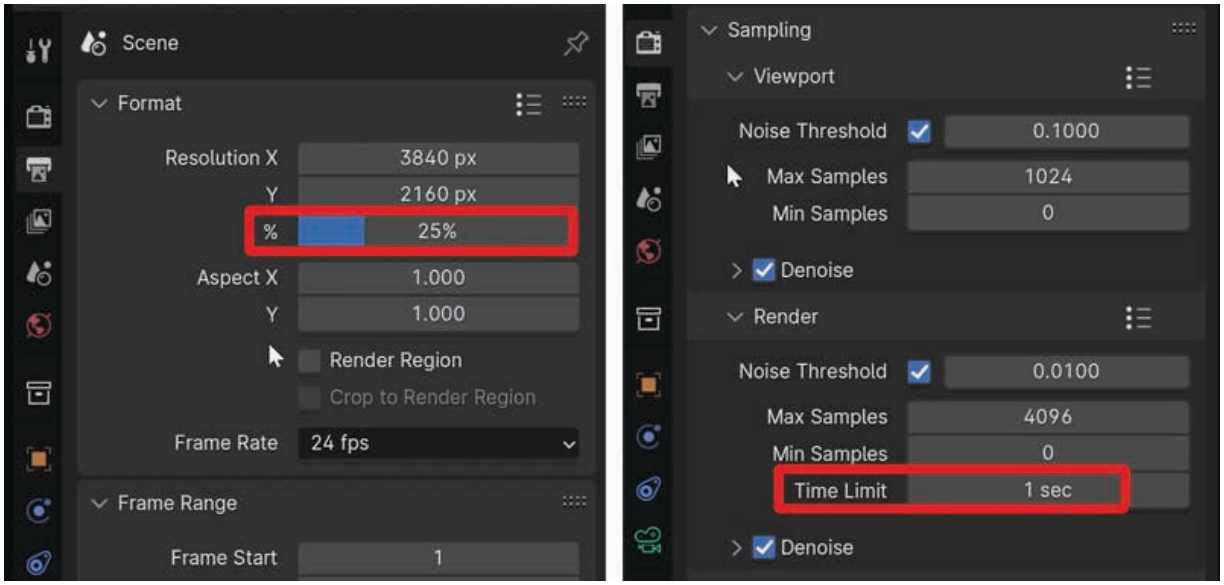


And I'll set that second key to Linear as well (by right-clicking on it and choosing Linear interpolation). If this was shot from a helicopter for example, the movement would be Linear in real life as well.

Let's calculate how many frames we need. I want **5 seconds** of animation with 24 frames per second, so that's **120 frames**. I'll make sure the second keyframe is and frame 120 and set the End to 120.

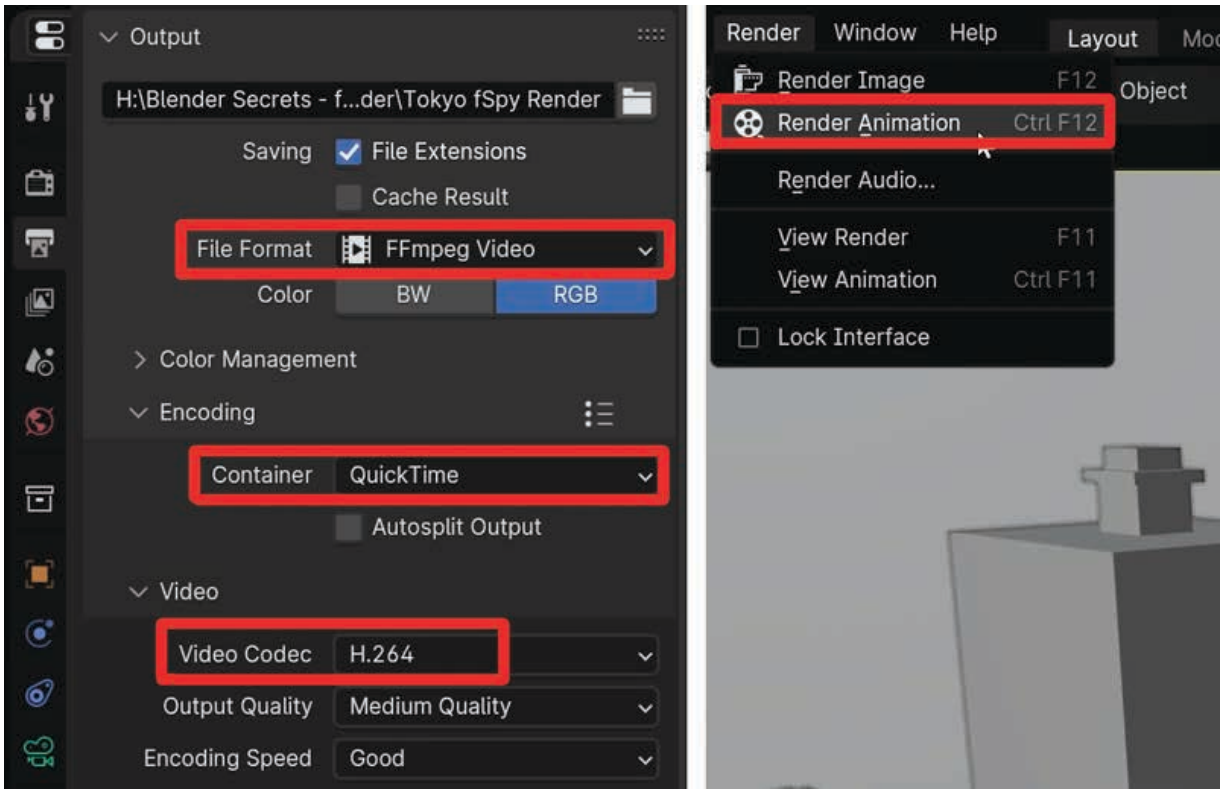


To do a test render, I'll limit the render time to 1 second per frame. And set the resolution to 25%.



As it will render fast, I will choose an **H264** video for the test.

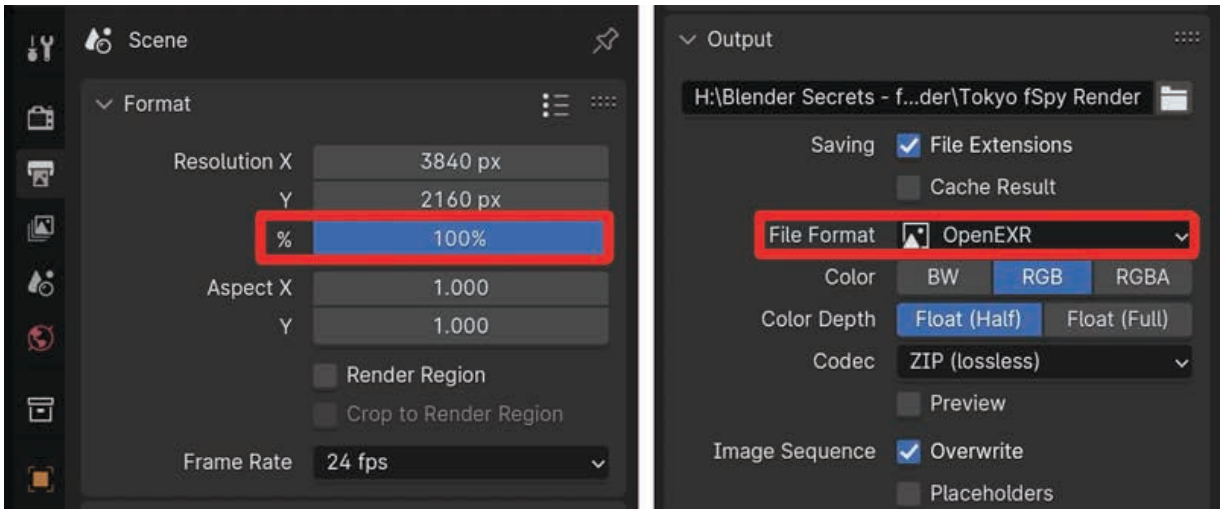
Then **press Ctrl+F12** or choose **Render Animation**. At this low resolution and with only 1 second as the render limit, it goes pretty fast.



And after a few minutes, we have the result of the low quality test render that we can play back. I recommend always rendering a low quality quick render before you commit to rendering high quality frames.

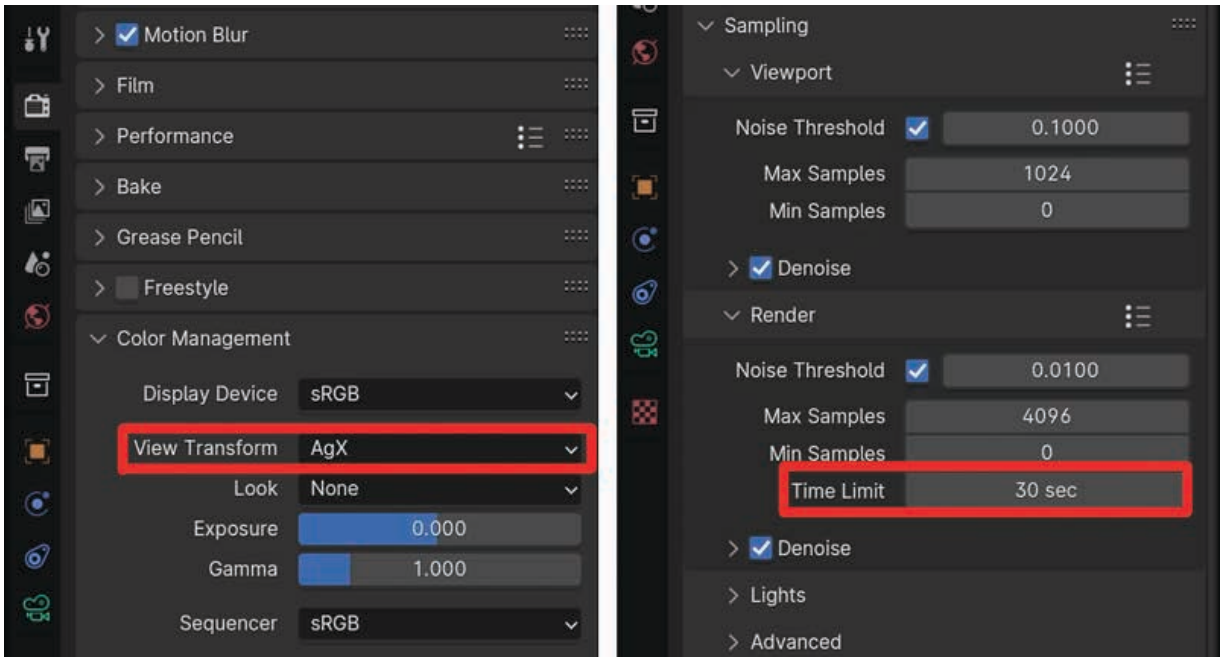
If the result is not to your liking, adjust the keyframes to make it more interesting. Once you've got something that you're happy with, you can set the resolution back to 4K for the final render. And **OpenEXR** for the file format. We'll render an **image sequence**. 16 bit, or half float, is enough to be able to do some nice color grading.

The advantage of rendering an image sequence instead of a video file, is that when the render stops at some point (for example, because of a computer or software crash), you won't lose the frames rendered up to that point. And you can just continue rendering from where it left off.



Just checking that it's set to **AgX**, which will give us a nice Linear image for later color grading. I'll also increase the **Time Limit** for the render. 120 frames times **30 seconds**, divided by 60 seconds gives us 60 minutes of render time.

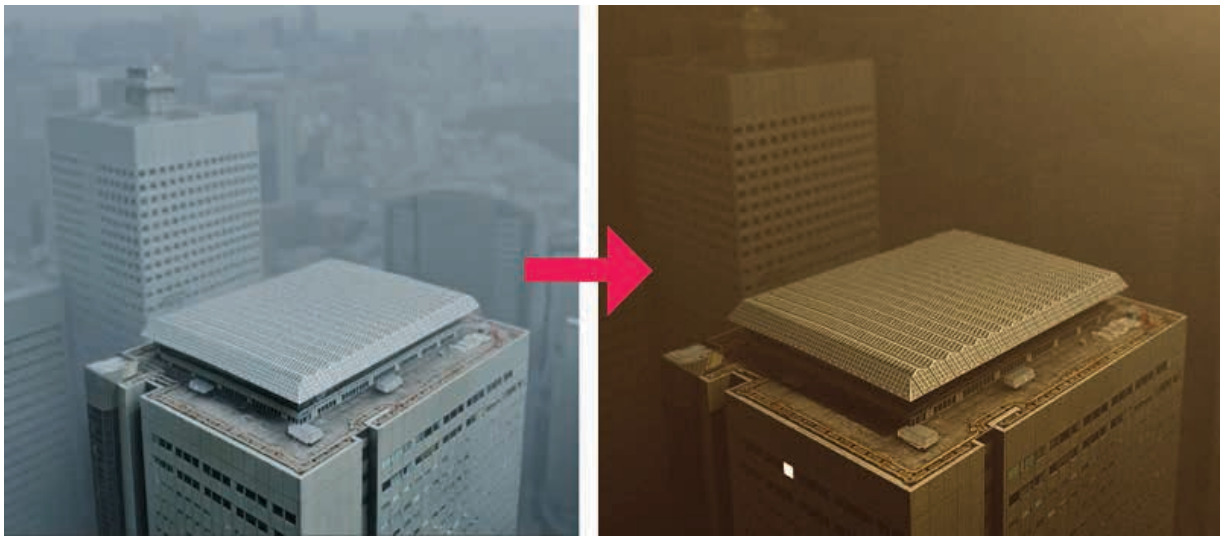
Before rendering, it's good to **save** just in case. In reality it took about 3 hours and 20 minutes to render all the frames, so the Time Limit isn't very accurate.



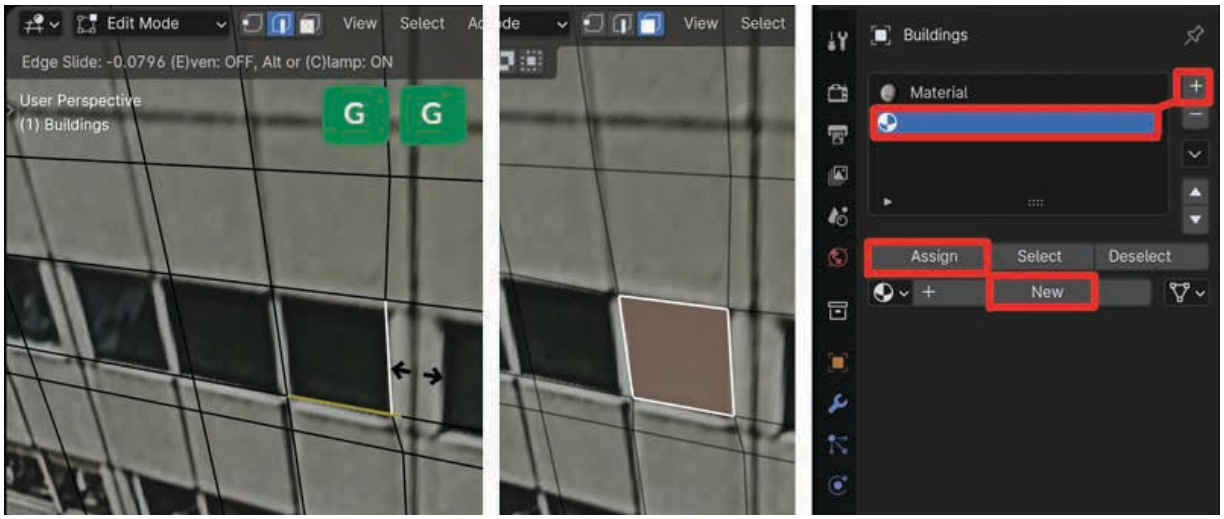
And to make it easier for the computer to render all this, I'll set the viewport to Solid mode.

But actually, before I render I want to make the scene a bit more mysterious.

I think it could be interesting if one of these windows had light inside, as if that's the last person in the whole city living there.



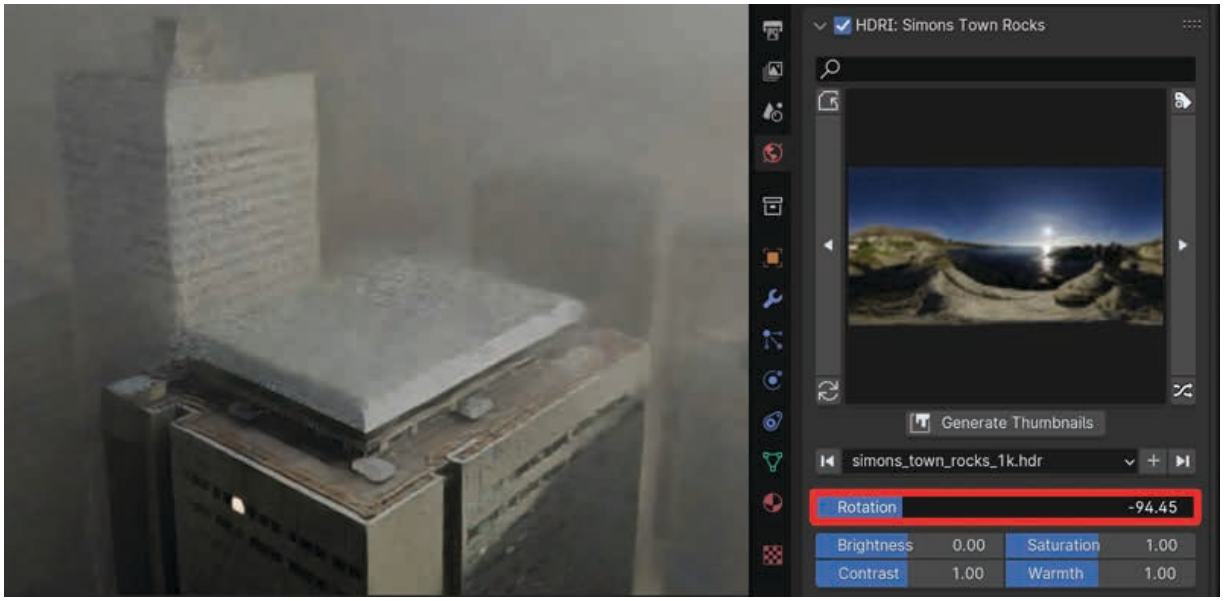
I'll need to add a couple more Edge Loops so I can select just that window. **Press Ctrl+R** to add Edge Loops where needed. If you **press G twice** you can slide Edges without distorting the texture. I'll add a second material to the window. Assign it to the selected Face and give it a name.



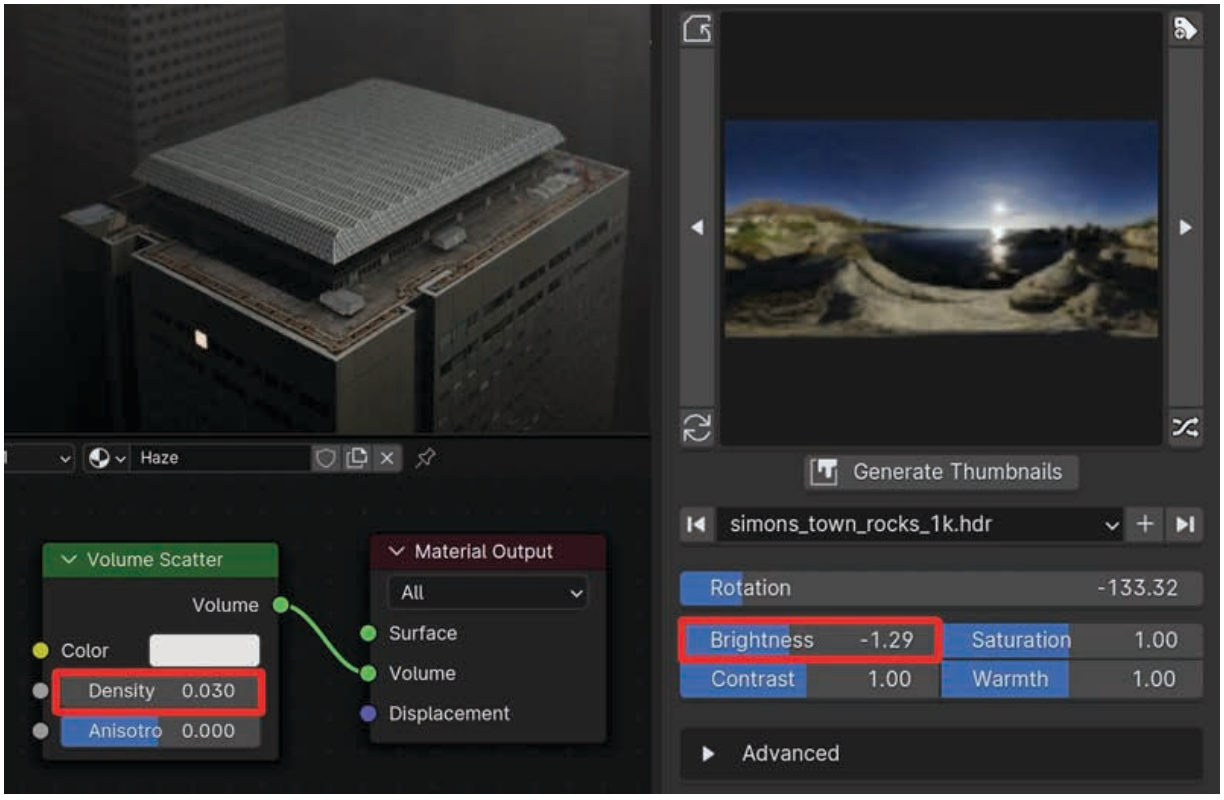
Remove the Principled BSDF and put an **Emission** shader in its place. We can manually give it a color… but to be more realistic, we can add a **Blackbody node** to use a real-life color temperature value. Something like 3200 Kelvin gives a nice warm artificial light color.

You can google these more common blackbody values. Looks like **3400 Kelvin** would be even more appropriate.

I'd like the light to stand out more. We can try to increase the **Strength**… or we can experiment with the lighting, so the window is in the shaded side of the building, which makes it stand out more. We can try different HDRI's, or by **rotating the HDRI** we can also get very different results.



We can make it even darker and more dramatic by increasing the **Density** of the Volume. We can also reduce the **Brightness** of the HDRI. This looks quite interesting and mysterious. Let's save the .blend-file and then render the animation in 4K.



And here is the result of the render, after some quick color grading.

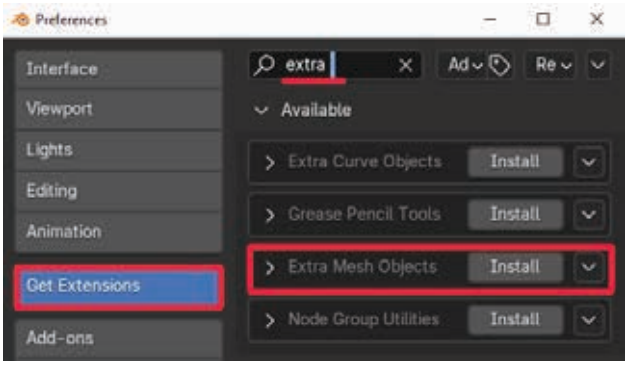
LOW POLY MODELING

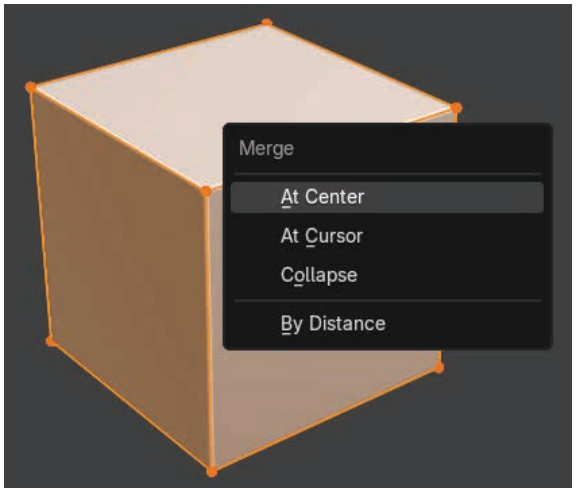
Low-poly tree in one minute

Let's make this cute blocky tree!

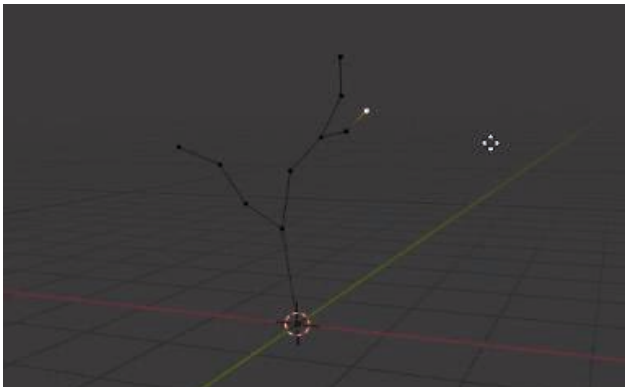
Add a **single vertex** (the Extra Mesh Objects add-on needs to be activated in Preferences for this to work).

Alternatively, add any Mesh primitive and in Edit Mode select all with **A** and then merge it to one vertex by **pressing M**, choose **Merge to Center**.

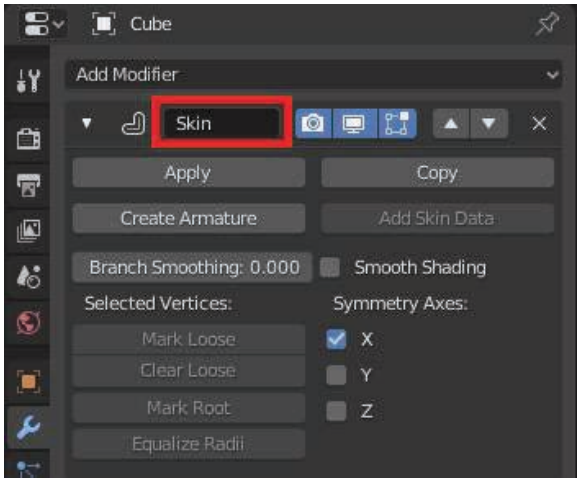




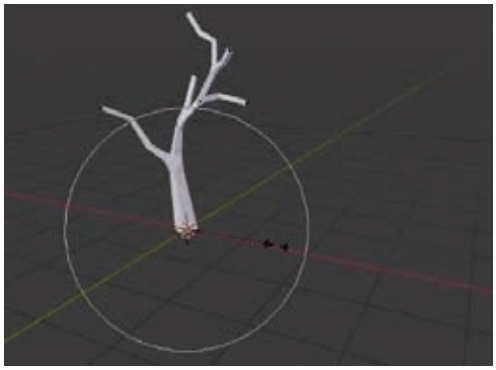
Extract some "branches" in Edit Mode by **pressing E**.



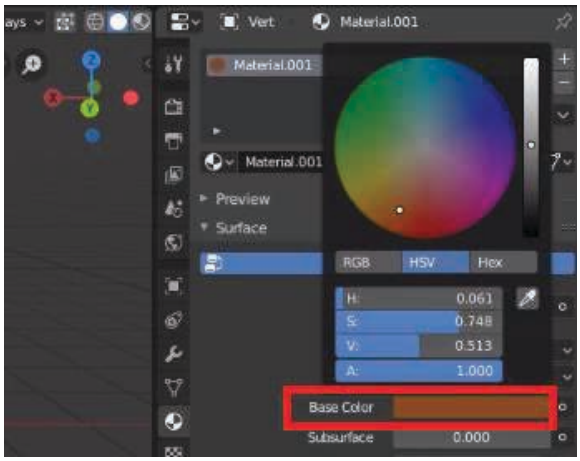
Add a **Skinmodifier**, then in Edit Mode select all vertices by **pressing A** and then scale with **Ctrl+A**.



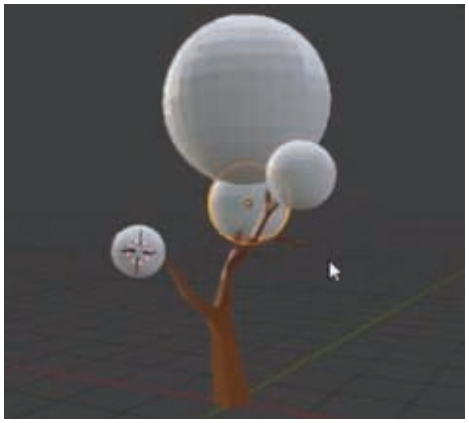
Select the vertex at the base. Using Proportional Editing (**press O** to toggle it on/off) press **Ctrl+A** to scale the base. This way, the base is thicker, with the thickness tapering off as it goes up. This makes it look more organic.



Give it a quick brown material by clicking on New in the material tab and changing the Base Color to brown.
Set the viewport to Rendered View by **pressing Z** and choosing **Rendered**.

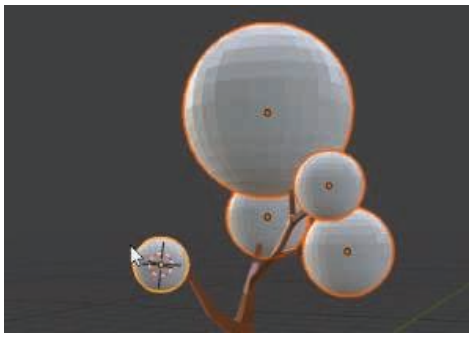


Add some UV Spheres to the branches and scale them with keyboard shortcut **S**.



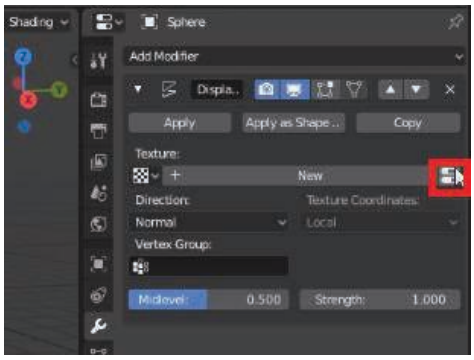
Select the spheres and **press Ctrl+J** to join them together.

Give the joined speres object a simple green color.

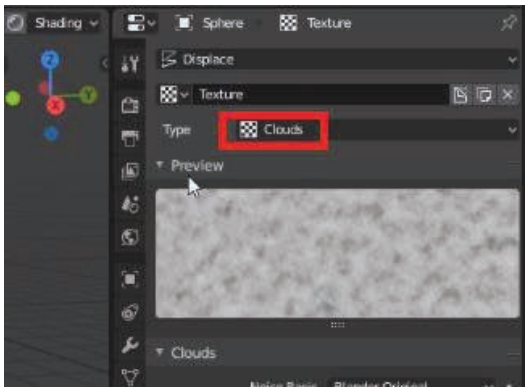


Add a **Displacement modifier** to the joined spheres object.

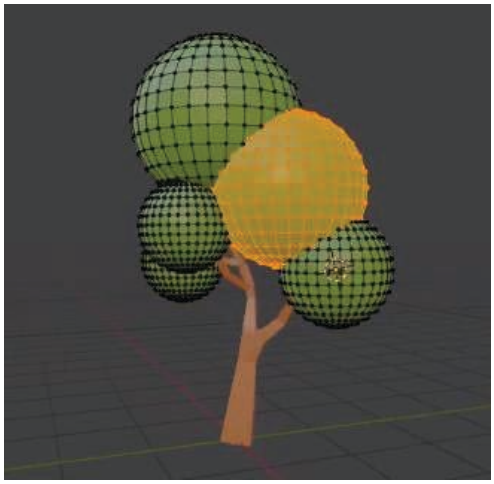
Click on the **Texture icon**.

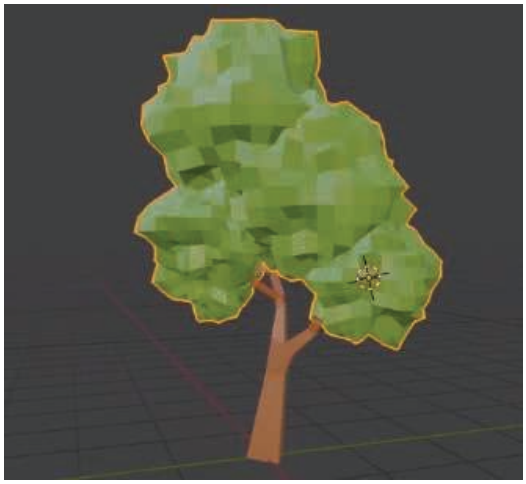


Give the Displacement modifier a **Cloud** texture.

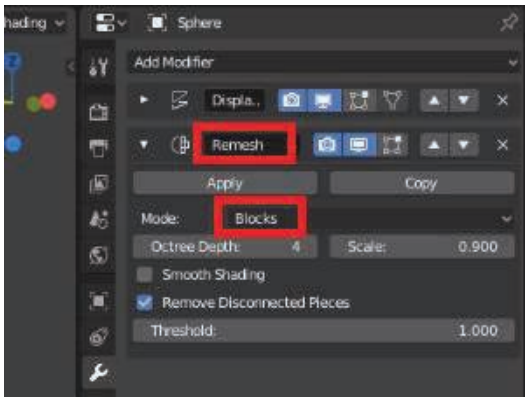


I made another sphere by selecting one vertex of the joined spheres in Edit Mode, selecting the rest with **Ctrl+L** and then duplicating it with **Shift+D**. This is because I felt there were not enough spheres. On the right below you see the result from the Displacement modifier (in Edit Mode you don't see it).





Add a **Remesh modifier** and set it to **Blocks**.

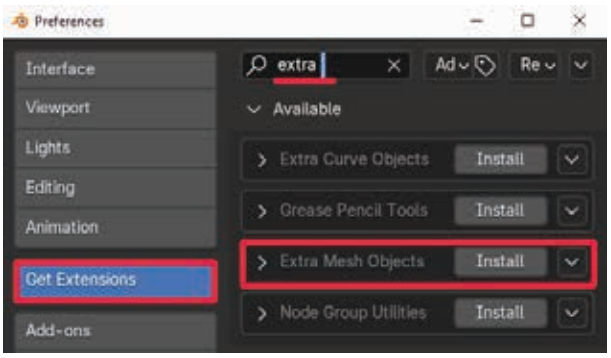


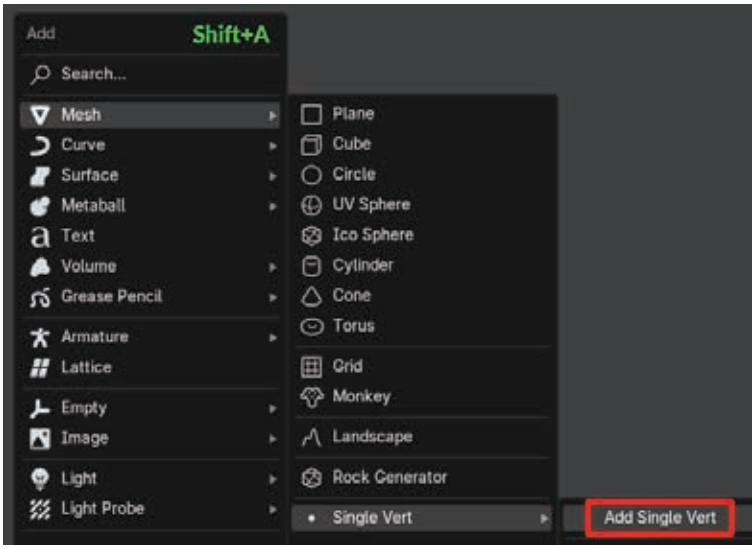
Add some lights and view it in Eevee.

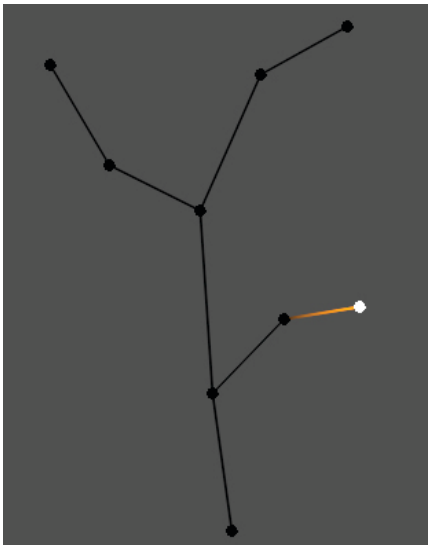
Even lower poly tree

Let's make a tree that has even less polygons than the previous one (which is more of a "pixelated" tree, if we're being honest!).

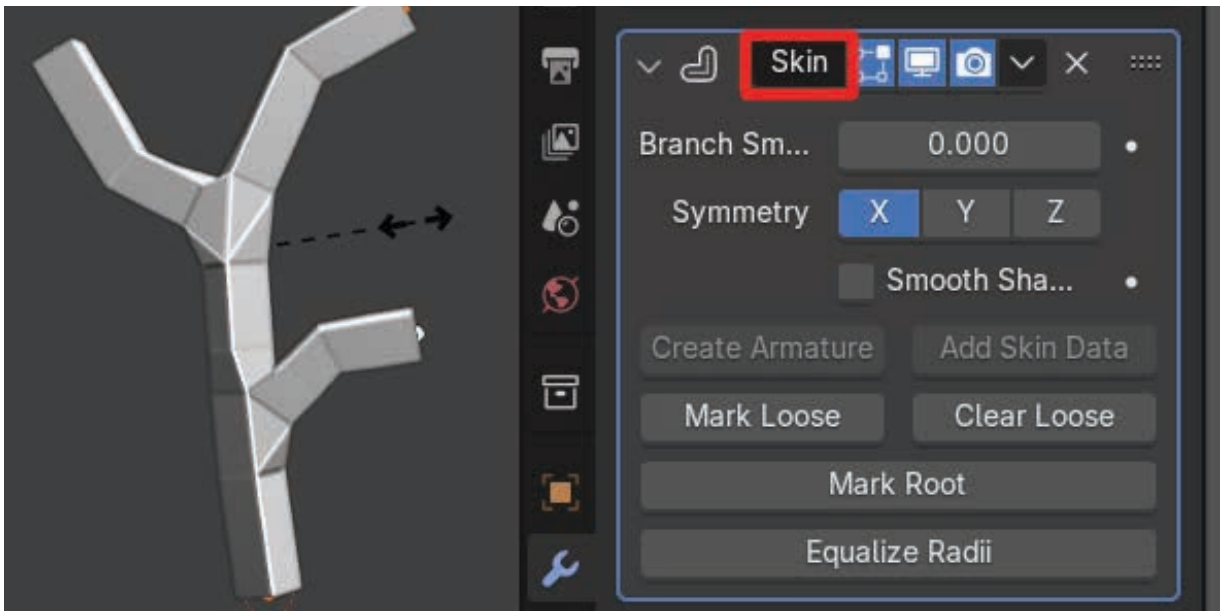
Create the trunk by extruding a vertex. You'll need to activate the **Extra Mesh: Extra Objects add-on** in Preferences to have this option.



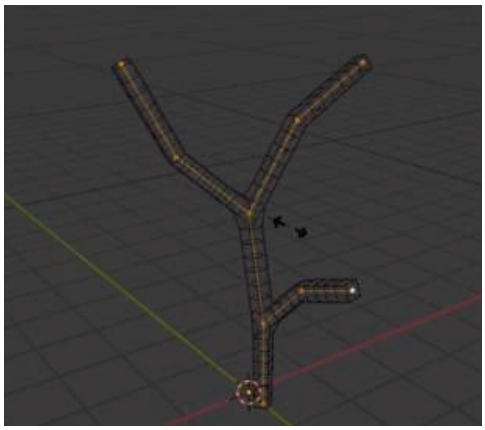


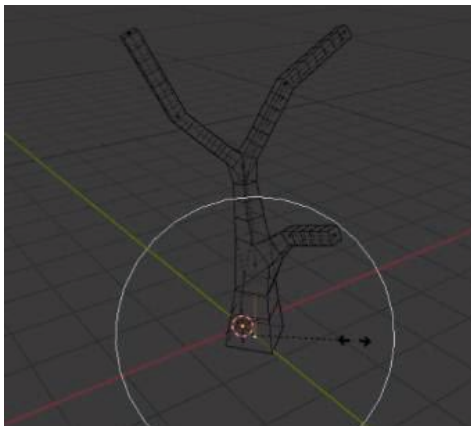


Add a **Skin modifier** for thickness.



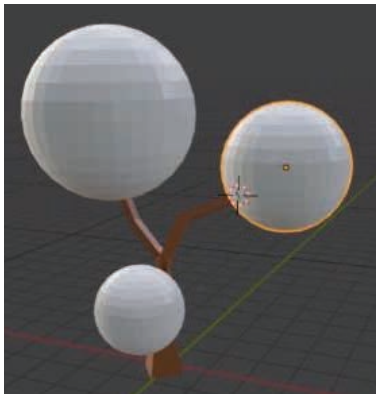
In Edit Mode, press A to select all the Vertices, then press Ctrl+A to change the thickness. Scale the base up with Proportional Editing (toggle on/off by **pressing O**).

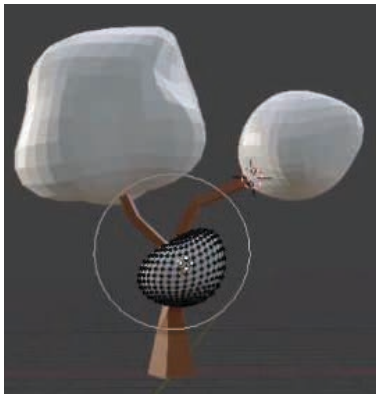


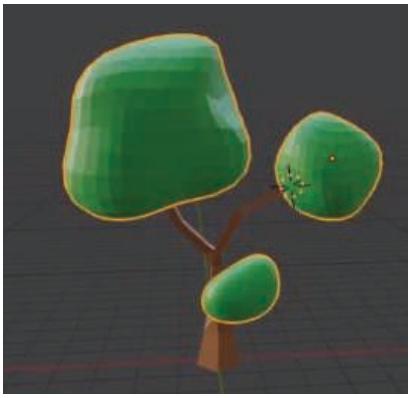


Create some spheres and place and scale them. With proportional editing (**shortcut o**) edit the spheres.

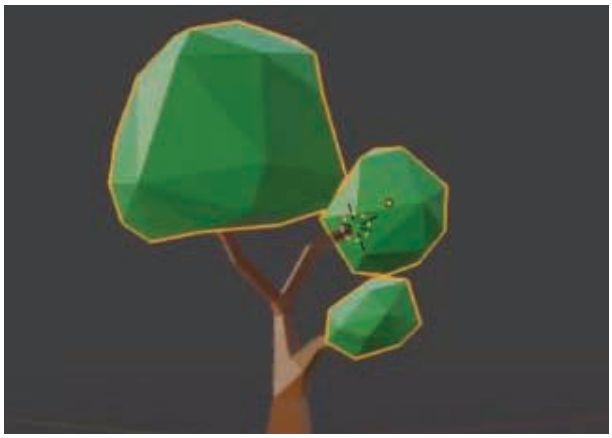
To do this, select one vertex (make sure you are in **Vertex select mode**!) and move it. As long as Proportional editing is turned on, the rest will move as well, according to the influence size (scroll the middle mouse wheel to change the radius of influence). Join the spheres by selecting them all and **pressing Ctrl+J.** Give them a simple green material.

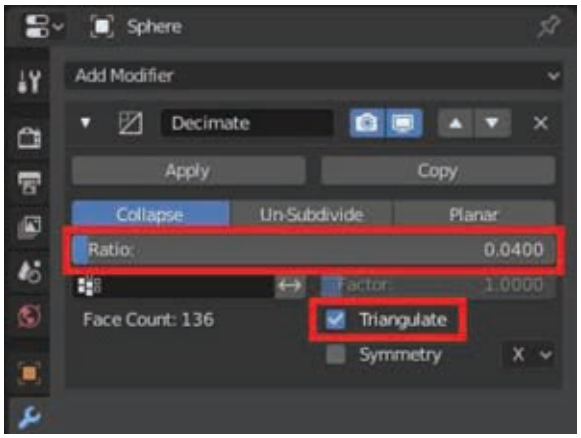






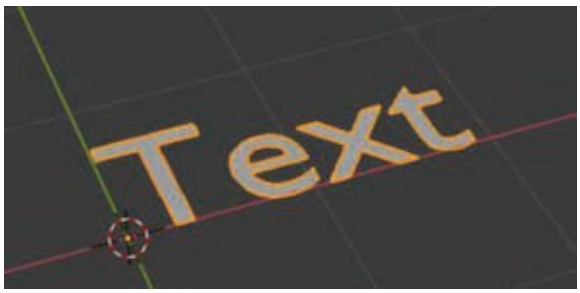
Add a **Decimate modifier**and set it to **Triangulate**. Change the Ratio until you get the look that you want.





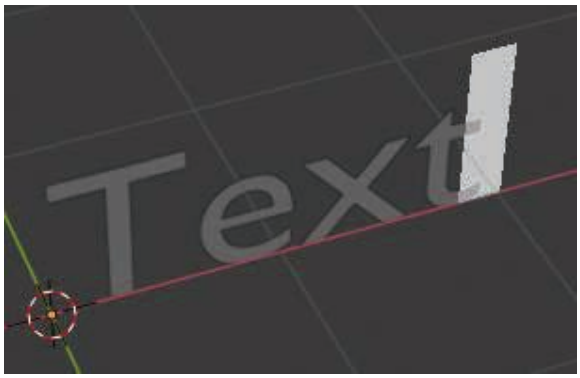
Low poly typography

Press Shift+A, Add Text.



You can rotate it 90° along the X-axis to make it stand upright by **pressing R+X+90**.

Press TAB to edit the text.

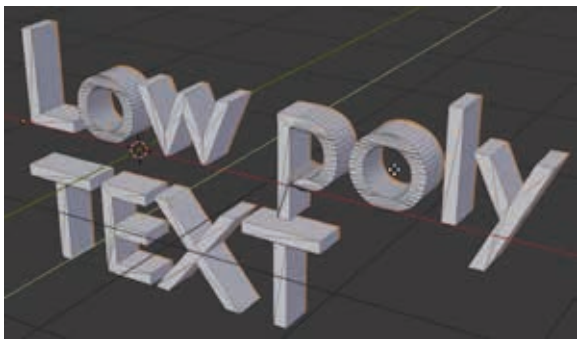


Now you can edit it like normal text. **Press Backspace** to remove characters, then type what you want.

After you are done **press TAB** to go back to Object mode.

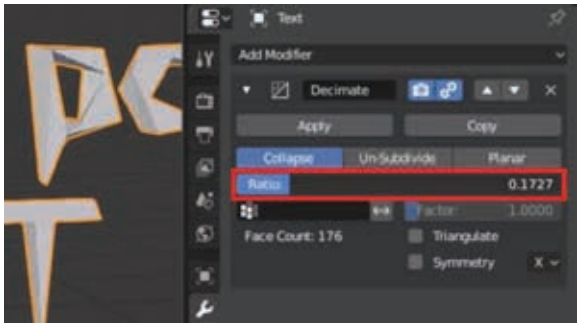
From the Object drop-down menu choose **convert to mesh**.

Now it is no longer a text object but a regular mesh that we can edit.

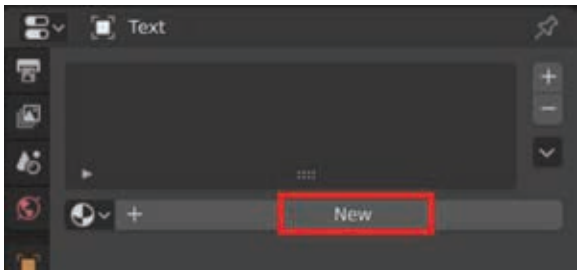


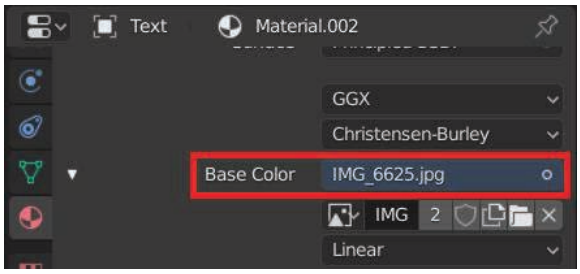
In Edit mode select all faces and extrude them in the Y-axis (**Press A+E+Y**).

Add a **Decimate modifier** and play with the **Ratio** value. The lower the value the more low-poly the text will be. **Apply** the modifier.

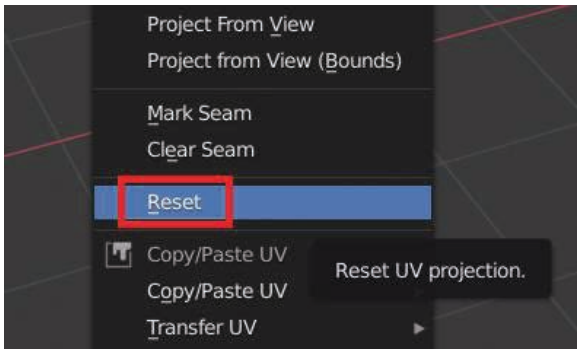


Add a Material to the text and add any image texture as the Base color.

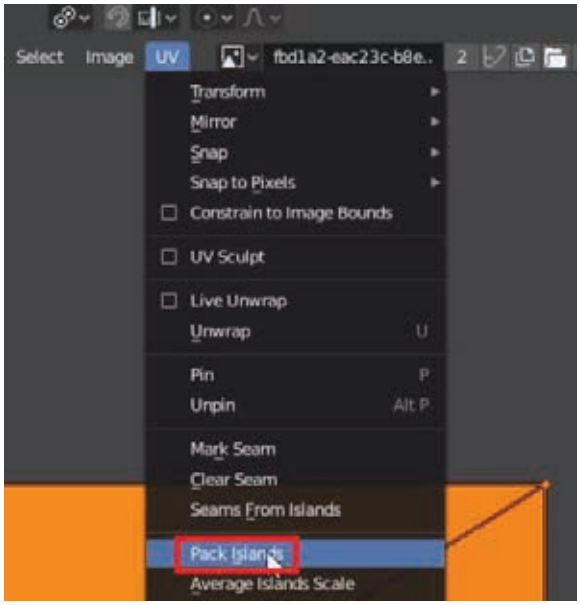




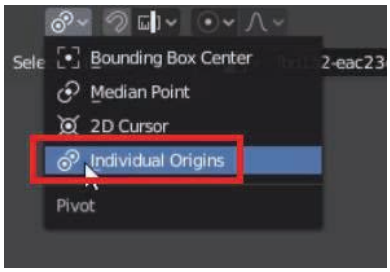
In Edit Mode **press U** to open the UV Unwrapping options, then choose **Reset**.

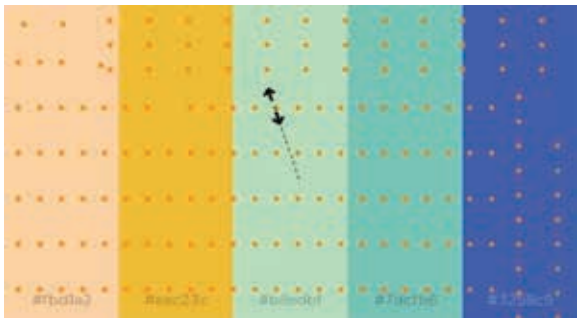


In the UV drop-down menu choose **Pack Islands**.



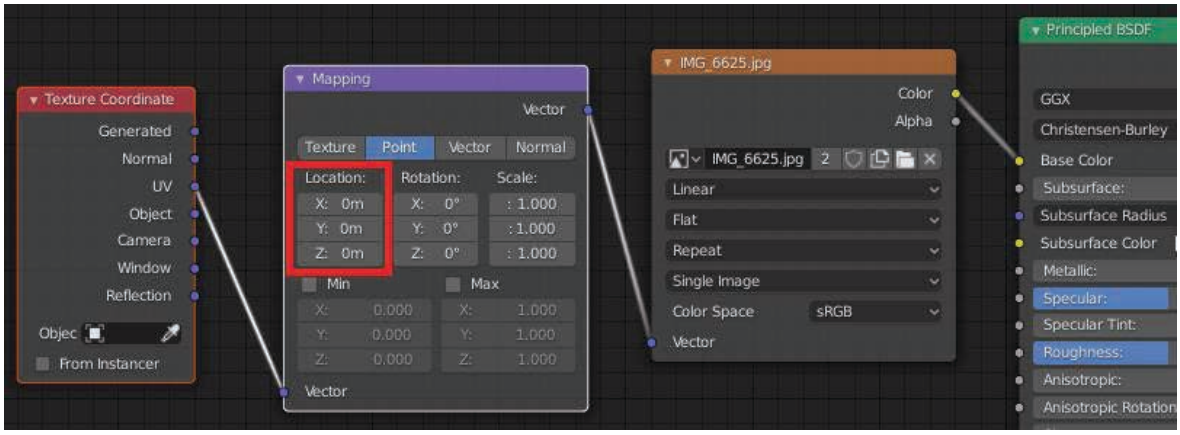
In the UV editor choose **Individual Origins**, scale to 0 by **pressing S+0**. This will reduce all the individual UV islands to points. The UV's will have the color of where they are on the image.





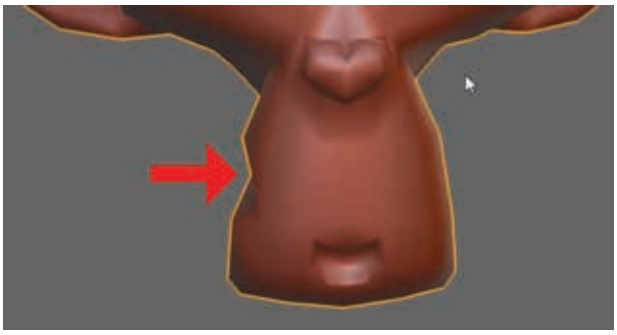
Moving the UV's will change the color distribution.

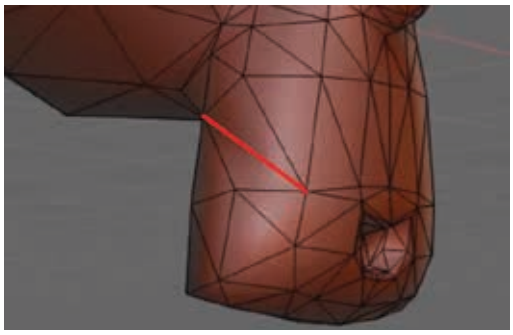
This way you can animate the colors of the text. To record it as keyframes, in the Mapping node in the Shader Editor change the **Location** values.



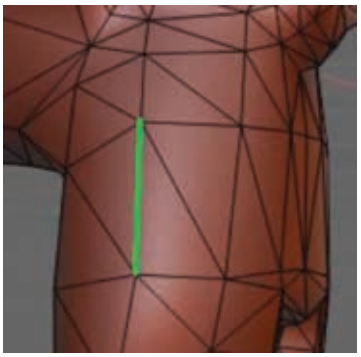
When to rotate edges

When a low-poly model is made of triangles, every edge makes a big difference. Here you can see one edge which is problematic. As you can see all the faces are triangles, they all only have three vertices and three edges. The red edge is the one causing trouble.





You can solve this by rotating it in the Edge menu (**Ctrl+E> Rotate Edge CW**).



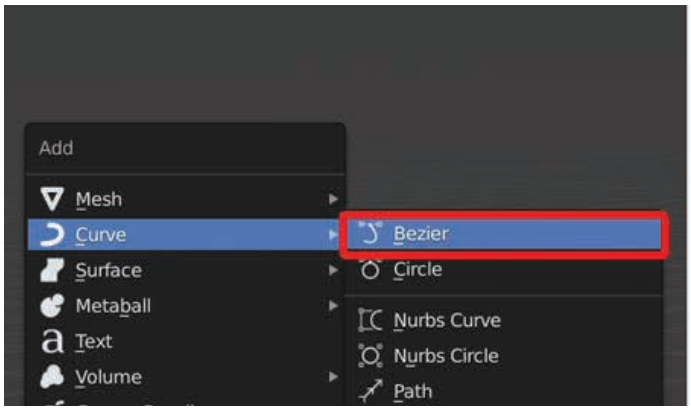
That's better!

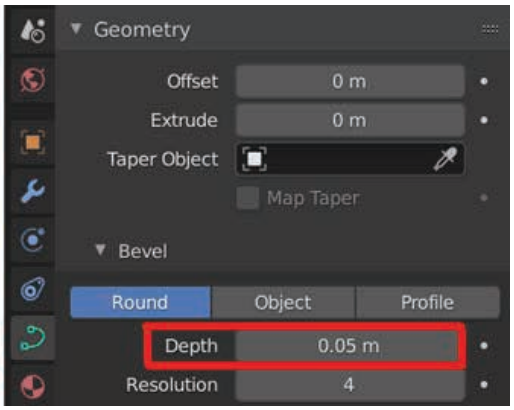
Click here to return to the index.

CABLES, PIPES and ROPES

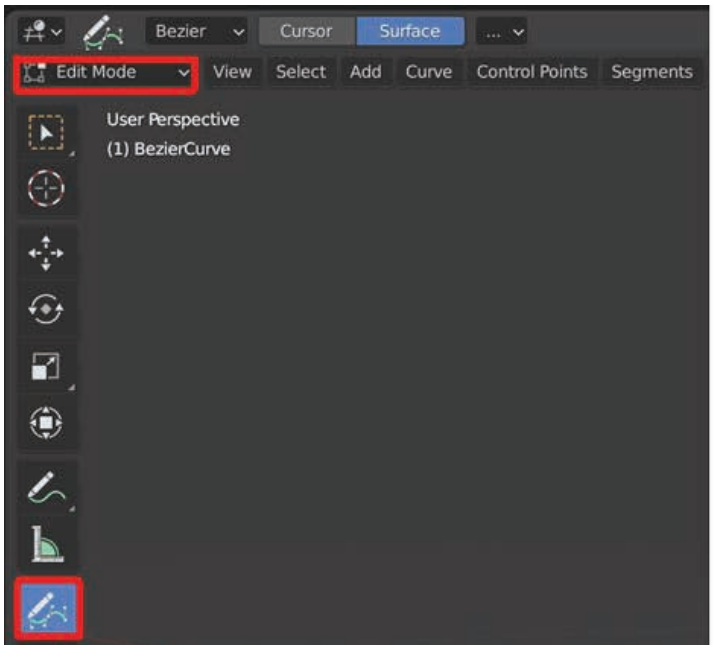
Drawing cables on surfaces

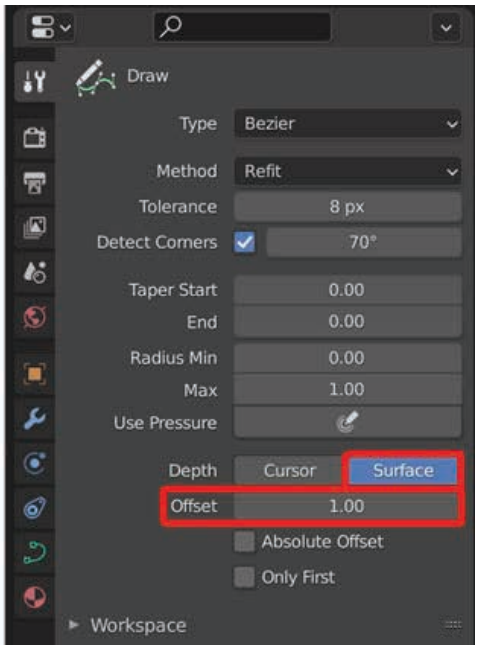
Press Shift+A and create a **Bezier curve**. Give it some material and increase the bevel **Depth**.



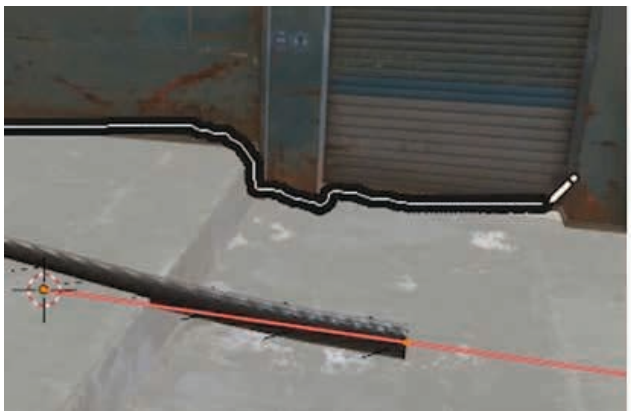


In Edit Mode choose the **Draw tool** and set Projection Depth to **Surface**. For less ground intersecting, set the **Offset to 1** when drawing the curves.





Now you can draw cables to your heart's desire. Curves come with generated UV's, which is really convenient. You can download this scene here if you want to experiment.

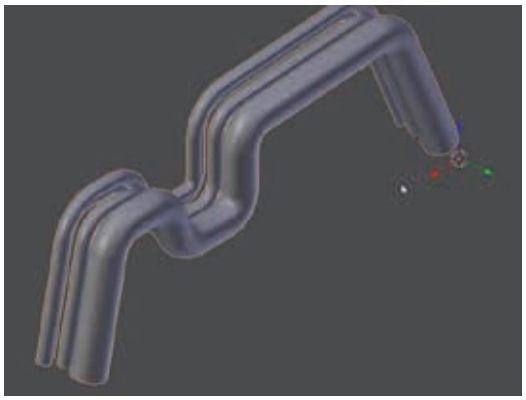


Click here to return to the index.

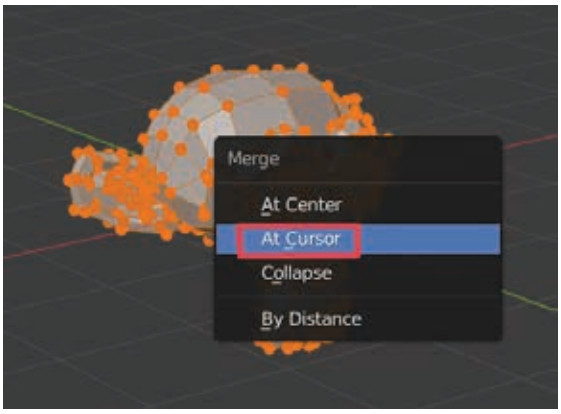
How To Make Pipes

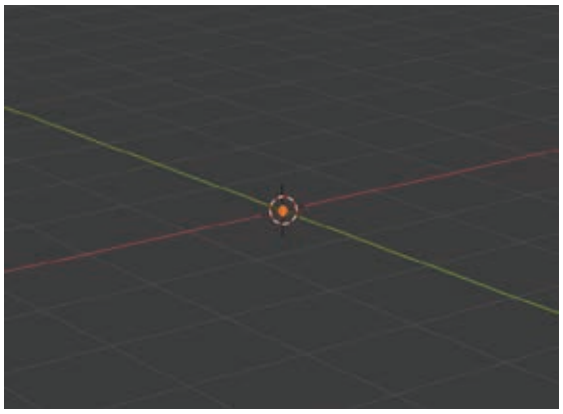
Pipes! Every modeler needs to make them at some point. But how do you make them?

First, you need a single vertex. There are two ways to do this.



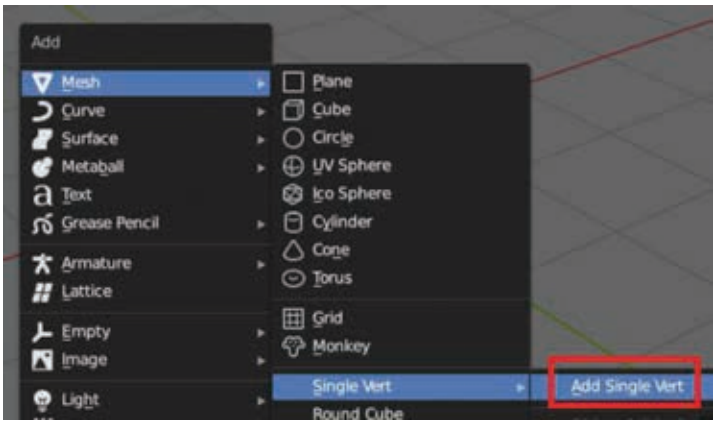
Method 1:
Add a primitive object by **pressing Shift+A** and choosing any mesh, even a monkey - it does not matter. In Edit Mode select all with **A** and then **press M** and from the Merge menu, select **At Cursor**.



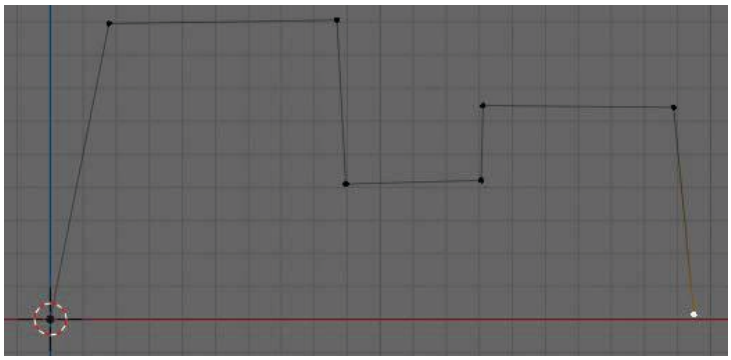


Note that for this option to be available, the Extension "**Extra Mesh Objects**" needs to be installed in the Preferences > Get Extensions.

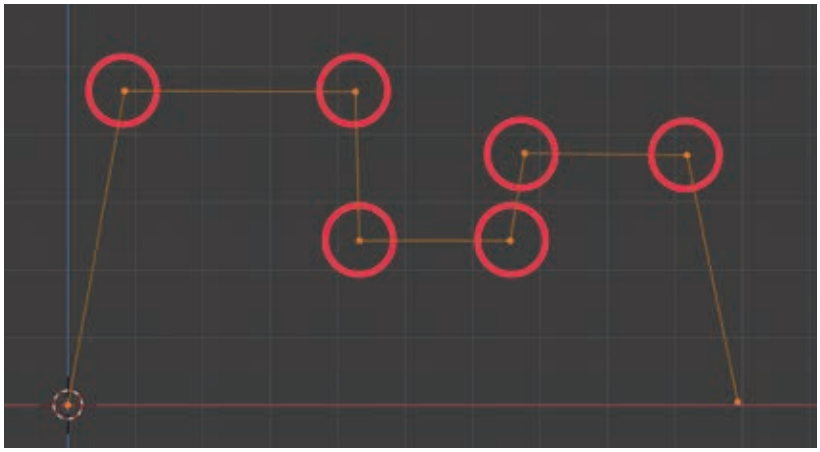
Method2:
Press Shift+A > Mesh > Single Vert > Add Single Vert.



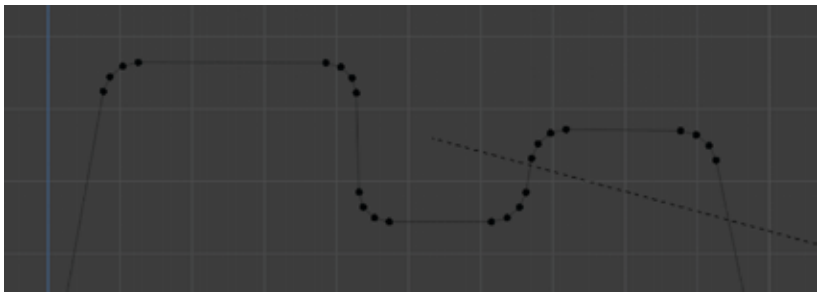
While still in Edit Mode with the vertex selected, **Ctrl + right-click** to extrude it several times to create the general shape of your pipe.



Select the vertices where the pipe needs to become rounded.

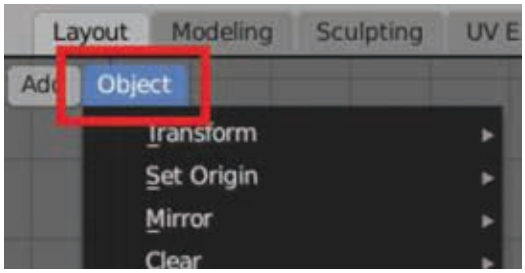


Press Shift+Ctrl+B to bevel the vertices, scroll the middle mouse button to add detail and make them rounder.

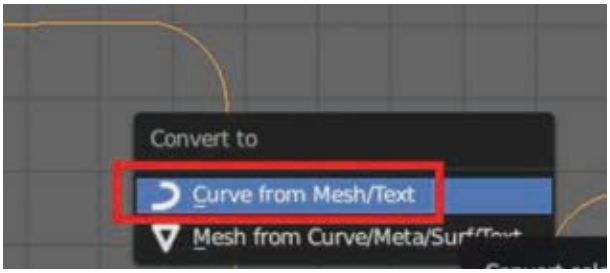


Now we need to turn this Mesh into a Curve. First, go to Object Mode. There are two ways to convert to Curve:

Go to the Object menu and Convert to > Curve from Mesh.

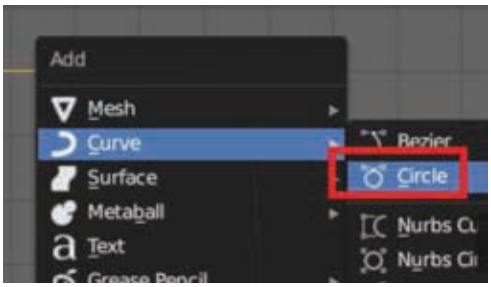


Or **Press F3** to get the search field and search for “convert”. Choose “Curve from Mesh/text”.

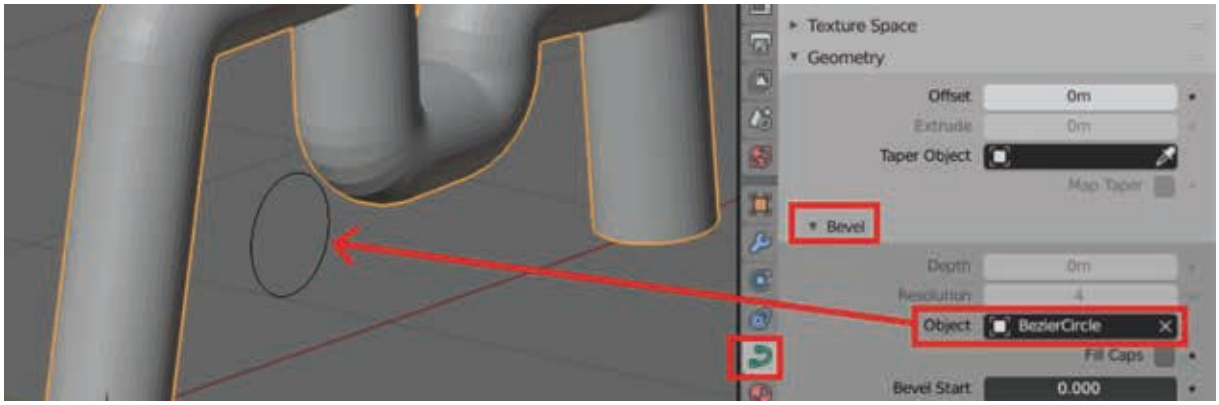


Press Shift+A for the Add menu.

Choose a Curve > Circle. The Location doesn't matter.

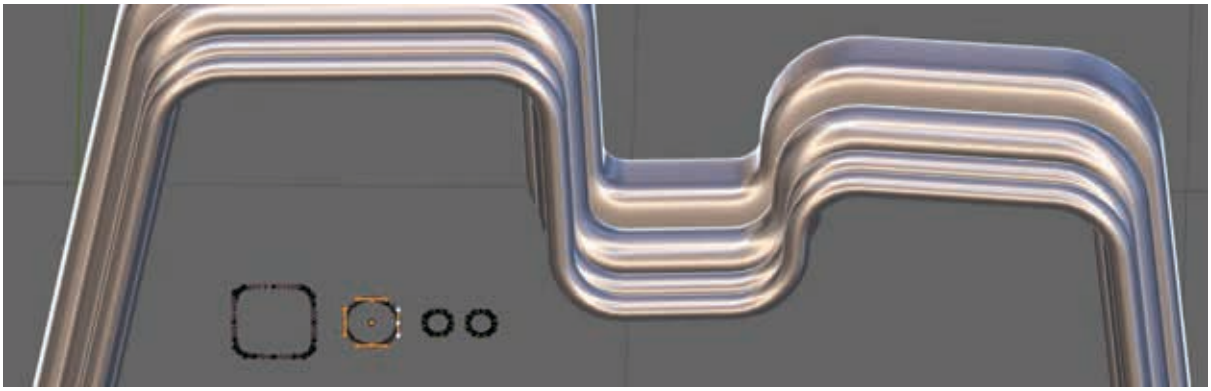


Go to the Curve Tab of the pipe Curve and as **Bevel Object** choose the circle.



Now the Curve takes on the profile of the Circle. Right-click on the pipe and choose **Shade Smooth**. Select the circle and scale it until you get the pipe thickness you want with **s.** Go back to Edit Mode with the Circle selected.

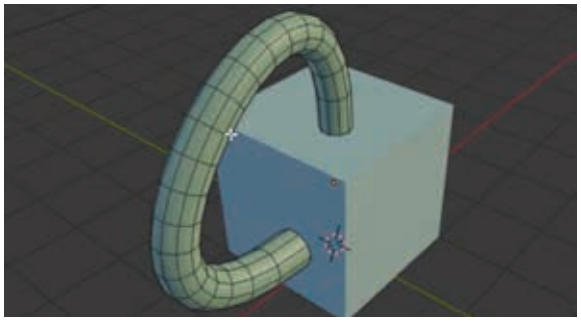
You can duplicate the circles in Edit Mode with **Shift+D** if all the vertices are selected. Move them with the shortcuts G + X or G + Z. The pipe will change shape accordingly. You can even move the handles on the Circle to make the pipe more square, for example.



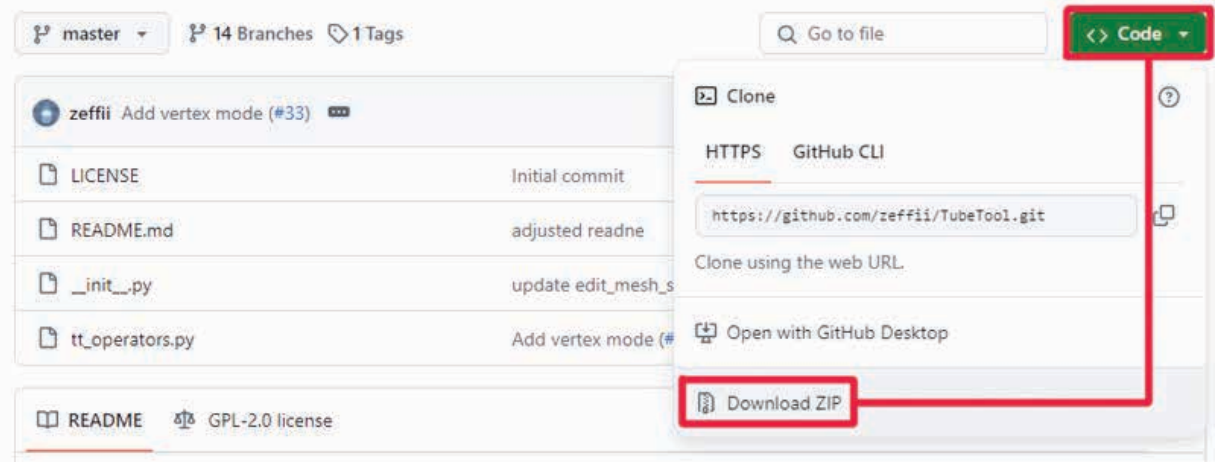
Convert the pipes to a mesh if you want by **pressing F3** and in the search menu again searching for "Convert to". But this time choose "Mesh from Curve". After this you can discard the circles. Have fun making pipes!

Tubetool add-on

Tube tool is an add-on that lets you quickly create tubes by selecting two faces and choosing the Tubetool option.

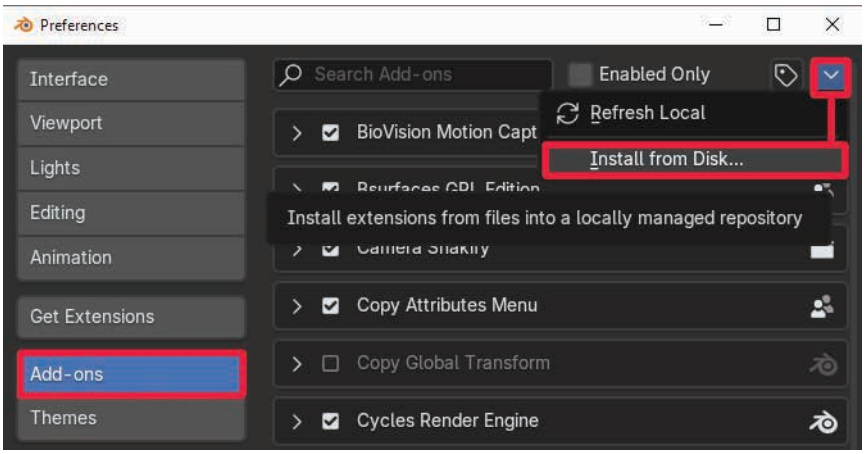


Download and install the TubeTool add-on here. Click on the green **Code** button and choose **Download ZIP.**

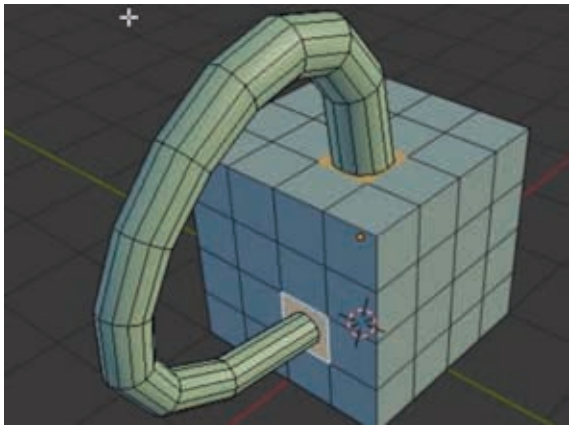


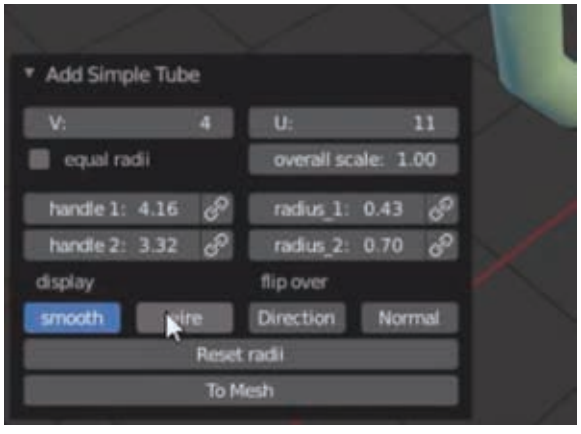
Then in the Add-ons tab in Preferences, click on the drop-down menu and choose **Install from Disk**.

Choose the ZIP file you downloaded.

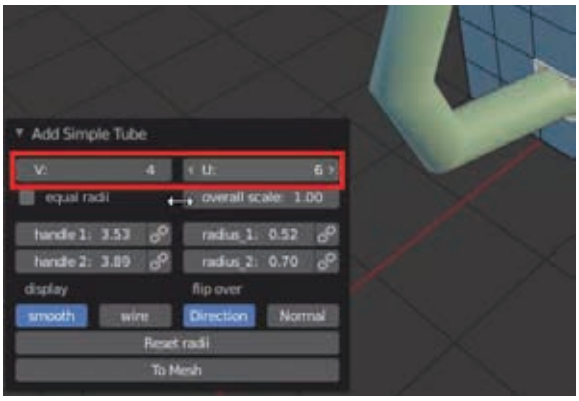


To use it, simply select two faces, right-clickand choose **Add Tubing**.
This creates a tube between the two selected faces. Expand the menu for more options.



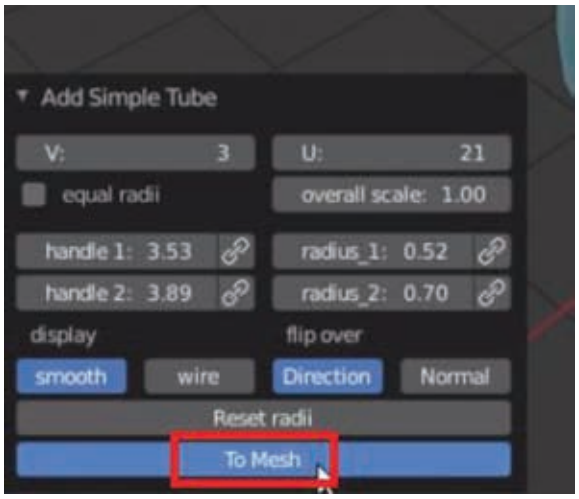


The V and U values control how much detail the tube has. If you've experimented a lot with all the values and want to return to the defaults, click on **Reset radii**.

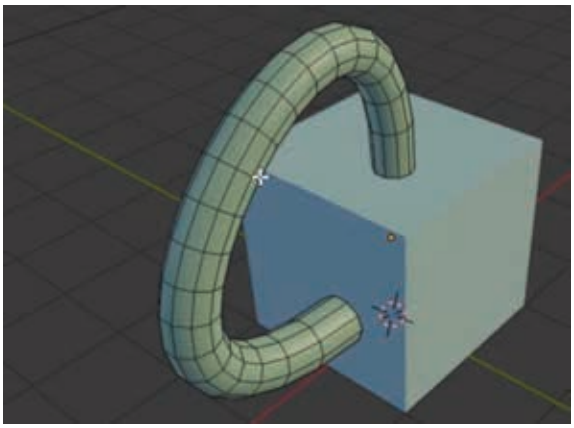


If that button doesn't work (it seems to be unreliable in my experience) **press F3** and search for "convert to". Choose **Mesh**.

Clicking on **To Mesh** converts the tube to a regular Mesh object, as you would expect.



Now you've got a regular tube-shaped mesh.

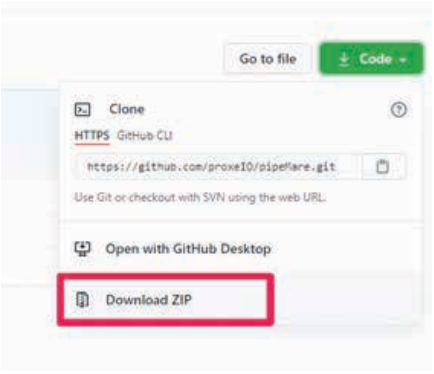


Pipe Nightmare add-on

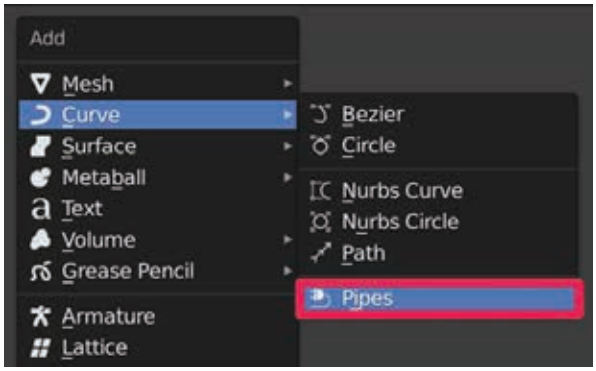
Pipe Nightmare is a free add-on created by **Proxe**, one of the people behind **Hard Ops**. It lets you create crazy amounts of pipes with almost no effort.

First download the Zip file of the add-on from **Github** and install and activate it from Blender Preferences.

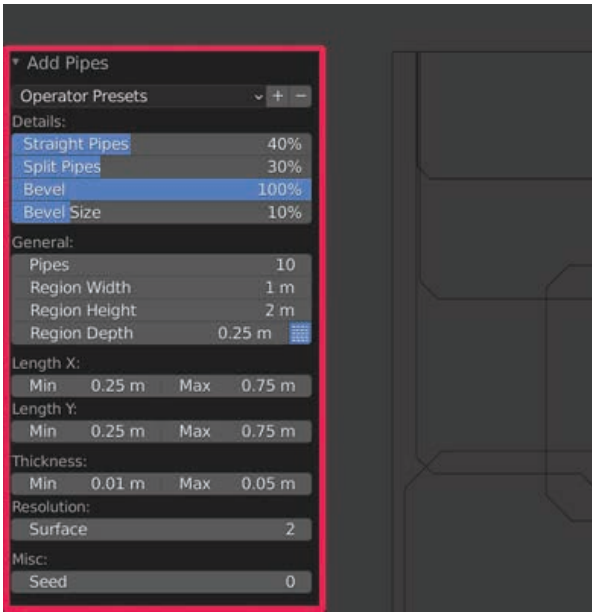
Download link: https://github.com/proxeIO/pipeMare



Then to add pipes, **press Shift A** and go to Curve> Pipes.

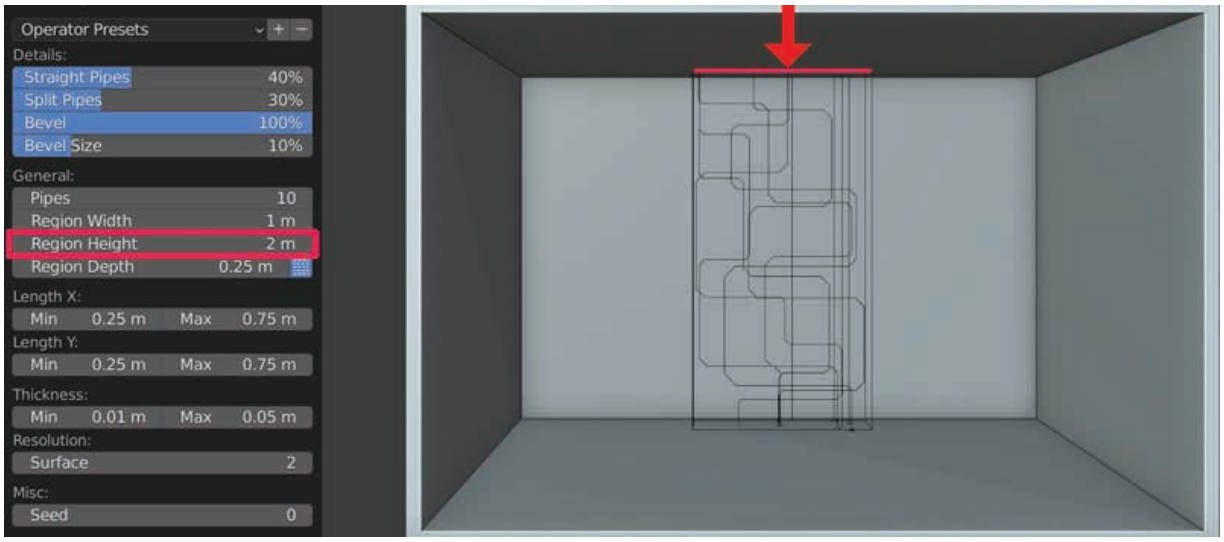


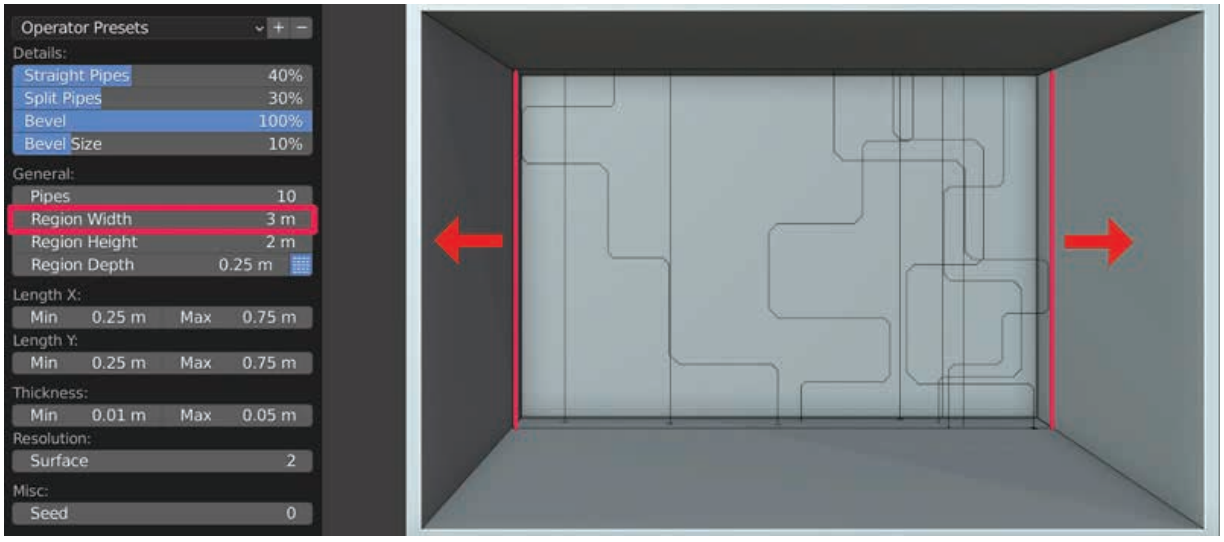
Now you can change a lot of options.

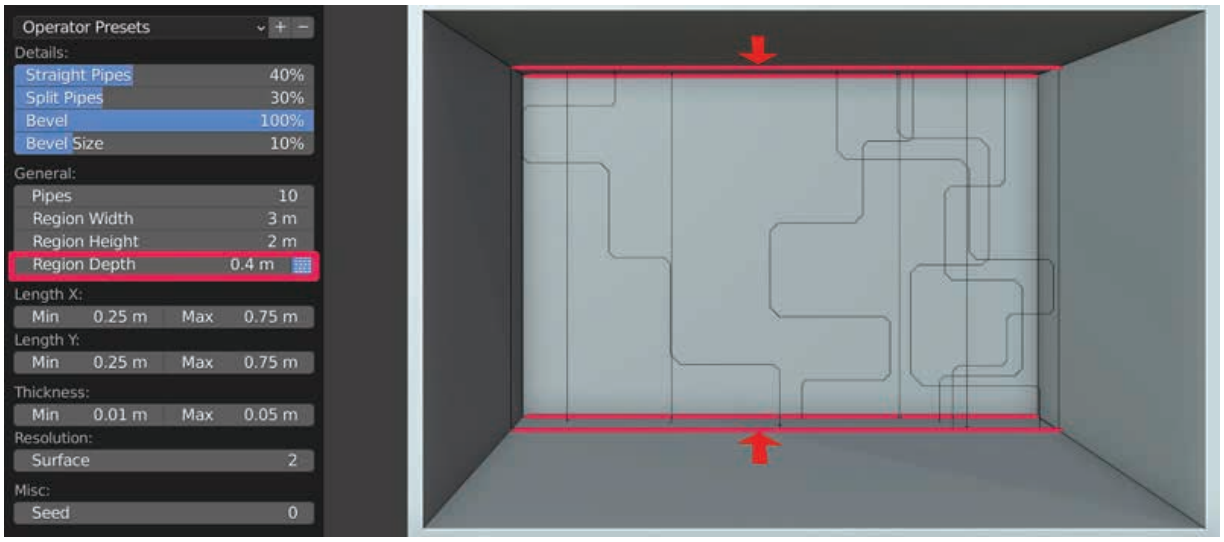


Click here to return to the index.

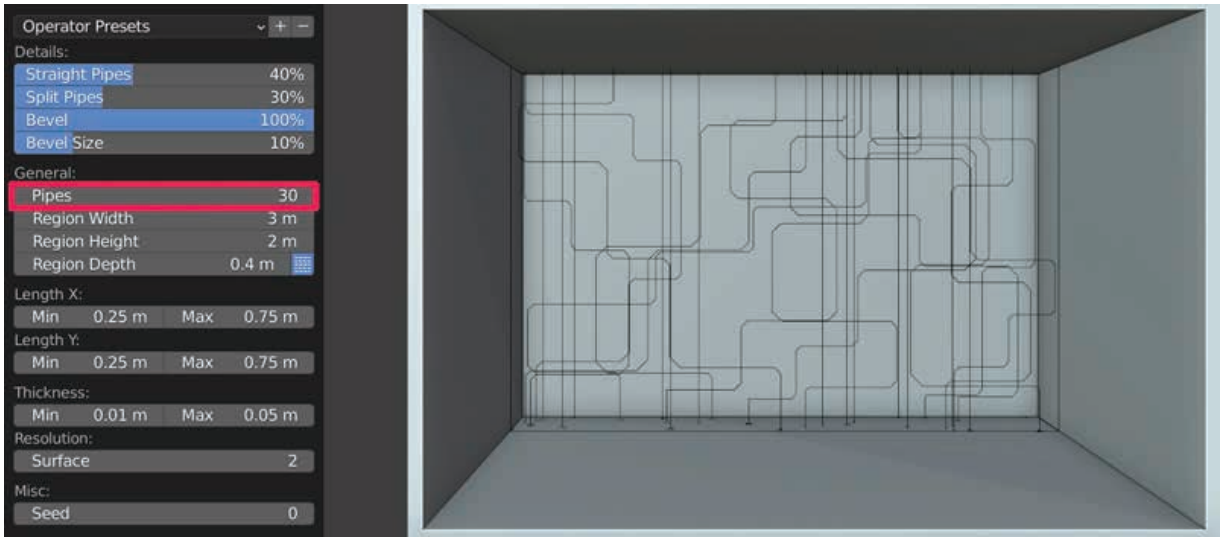
First I would change the region the pipes are created inside of to suit your environment that you want to place them in.







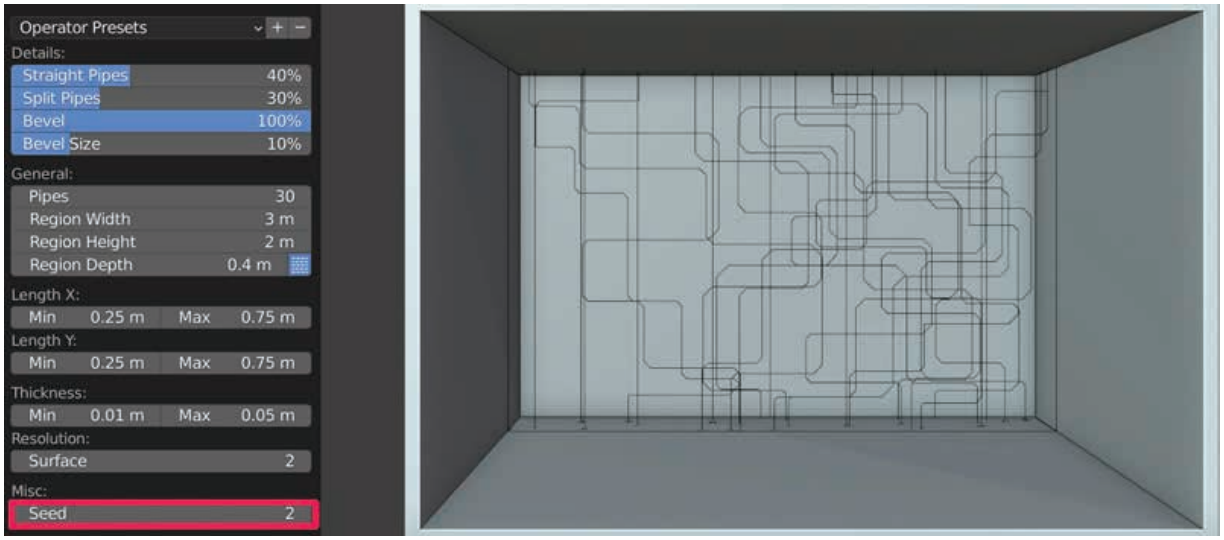
Then you can choose the amount of pipes and whether they are thick pipes or split into separate pipes.



Finally choose whether they are beveled by increasing the **Bevel value** ...

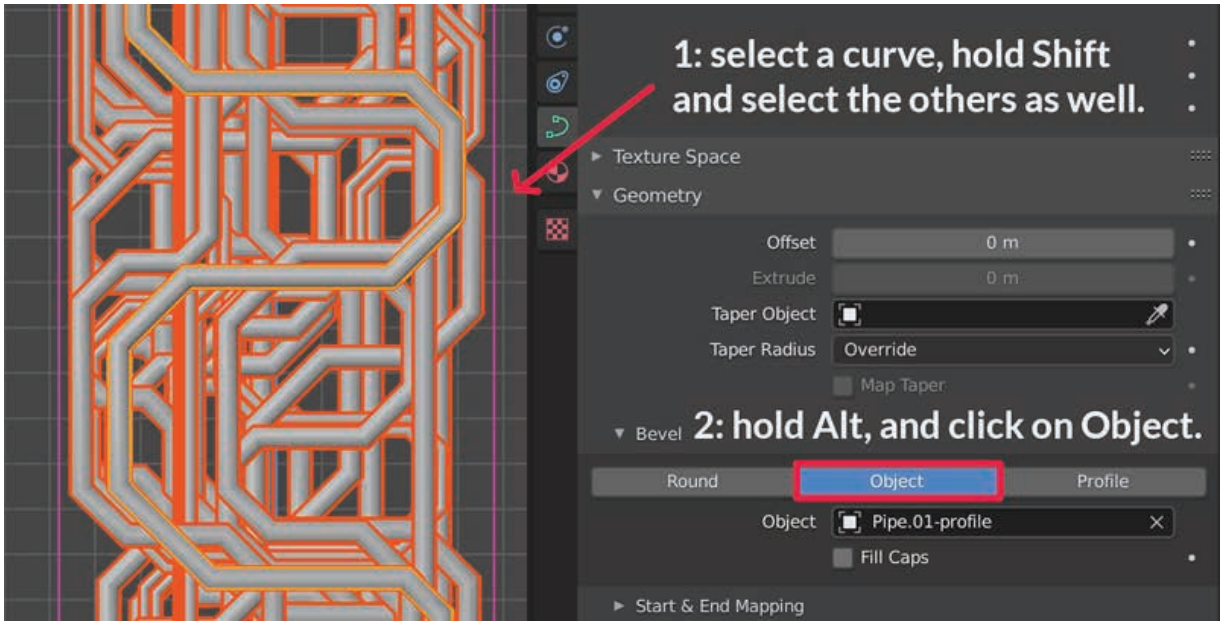
and how much they are beveled with the **Bevel Size** value.

You can also **click on Seed** to get some different random distribution of pipes.

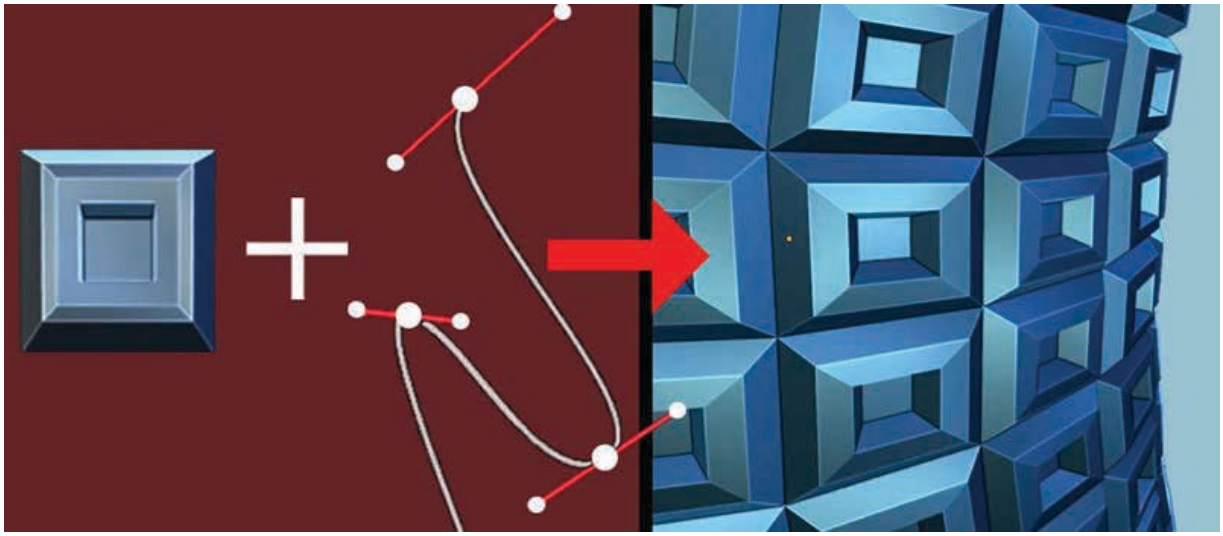


Unfortunately since Blender 2.9 the **bevel thickness** doesn't get displayed immediately anymore like it does in the video about this topic.

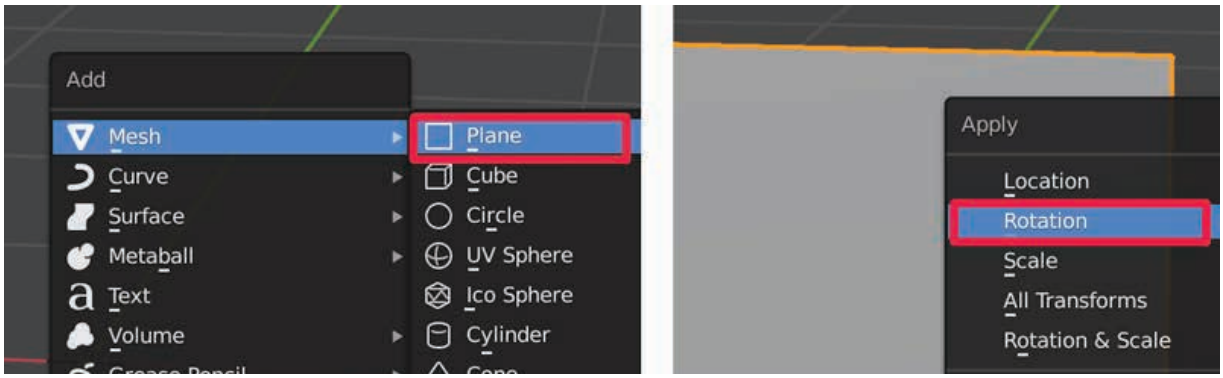
However by selecting a pipe, holding Shift and selecting the other pipes and then while **holding Alt**, setting the **Bevel type to Object**, you get the correct result. *(Note that the trick of holding Alt and selecting things doesn't work if you have "Emulate 3 button mouse" enabled in Preferences).*



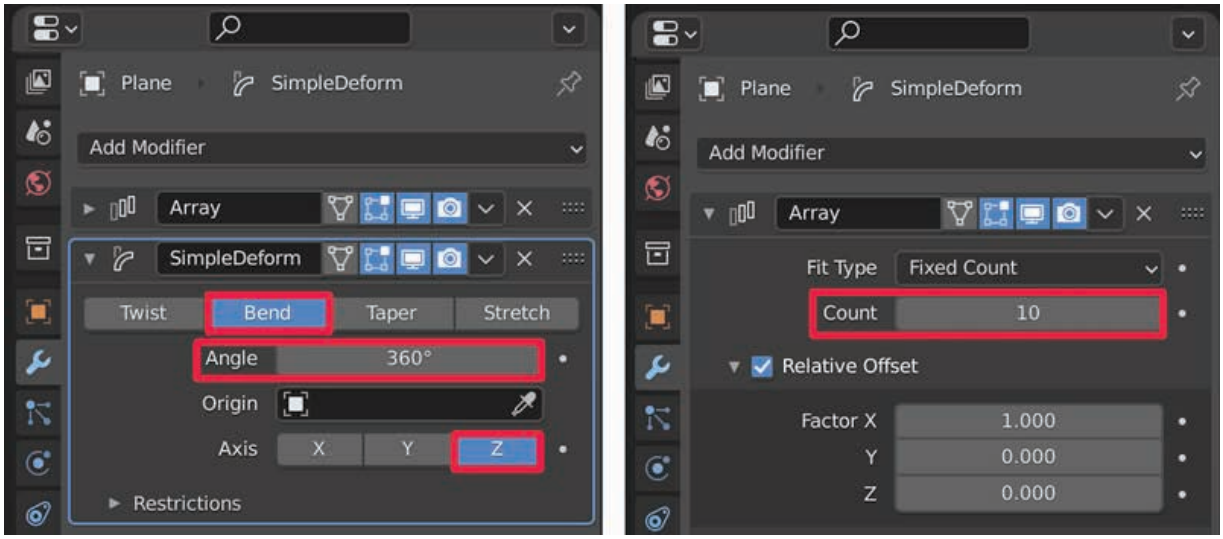
Detailed Cylinder Using Modifiers



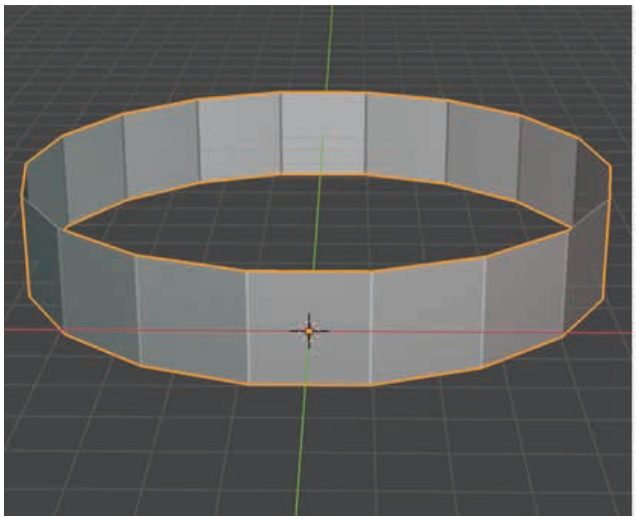
Using modifiers on a Plane, we can create a detailed cylinder. Add a Curve and you can control the shape as well. To start, add a **Plane** and rotate it 90 degrees on the X-axis. **Apply** the Rotation.

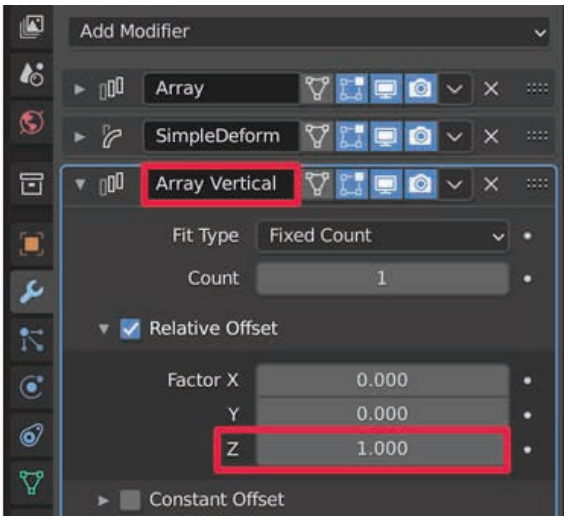


Add an **Array modifier** and a **SimpleDeform modifier**. Set the SimpleDeform to **Bend** at a **360° Angle** on the **Z axis.** Increasing the Array **Count** increases the thickness of the cylinder. Note that the order of the modifiers in the modifier stack is important.

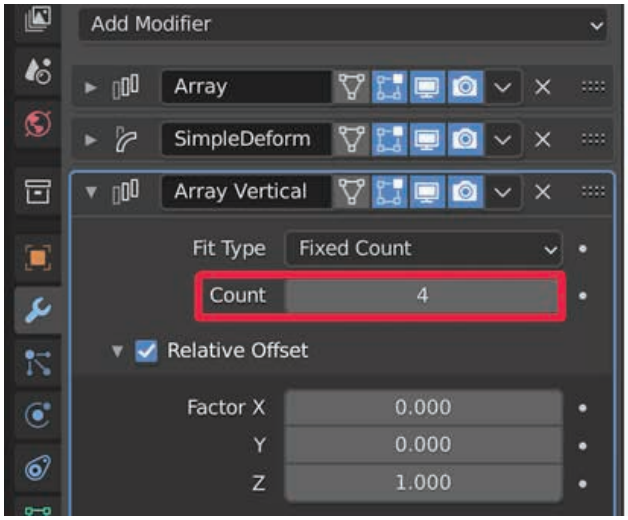


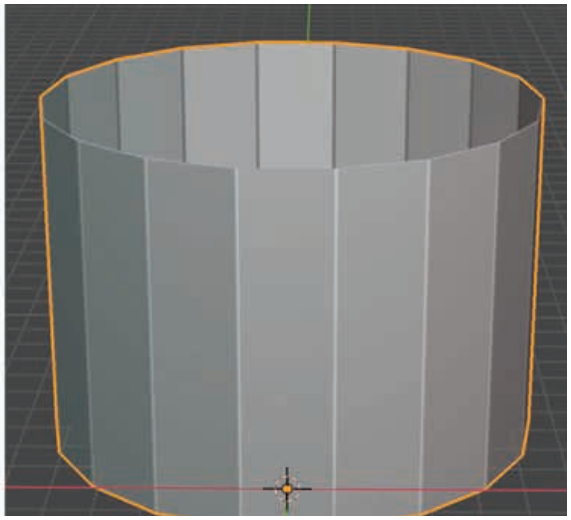
Now we have a cylinder, but it's very short. Add another **Array modifier**. Rename it Array Vertical to avoid later confusion. Set the Offset to **1 on the Z-axis** and 0 on the other axes.



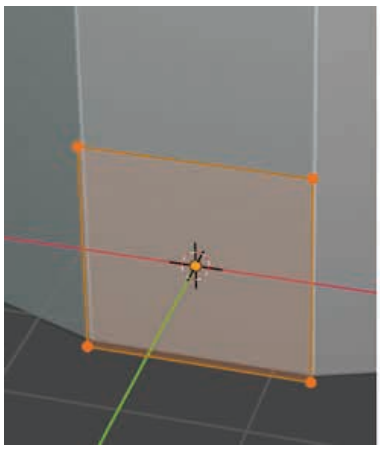


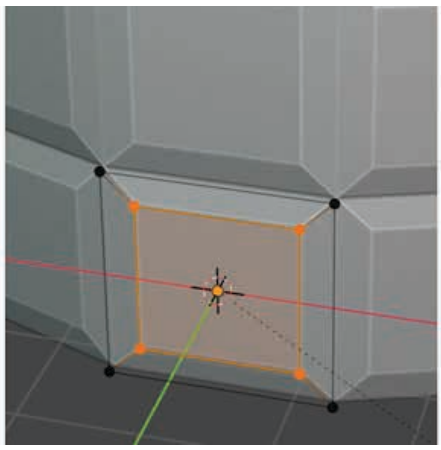
Increase the Count on the vertical Array modifier to increase the height of the cylinder.

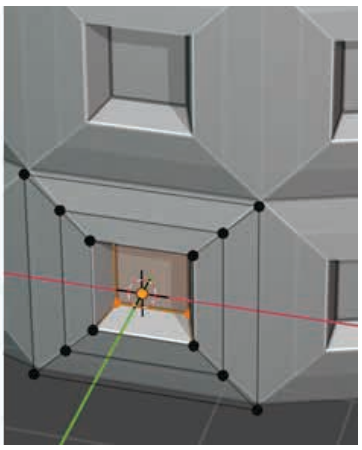




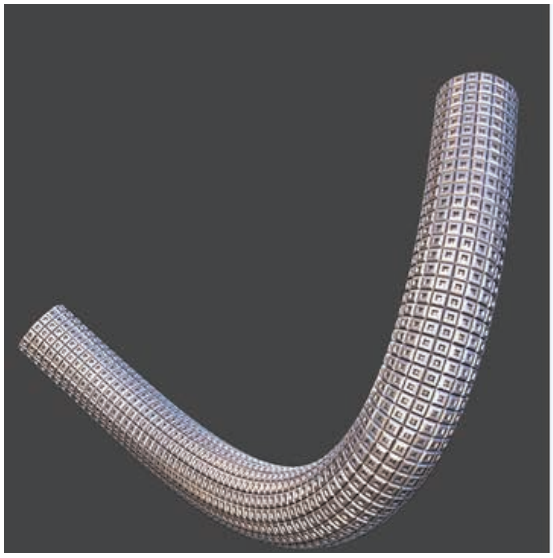
Now you can edit the original Plane to create interesting details. For example by **Poking Faces (Face > Poke Faces)**, or **Insetting (i), Extruding (e)** and so on. Whatever you do to that original Plane will also happen to the rest of the cylinder.

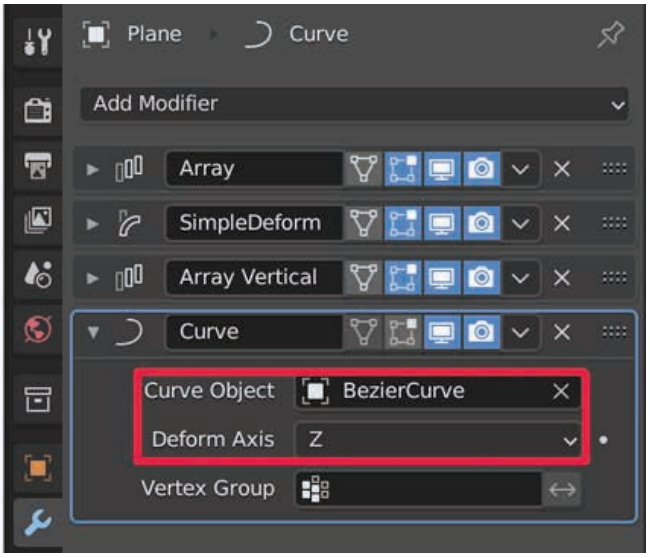




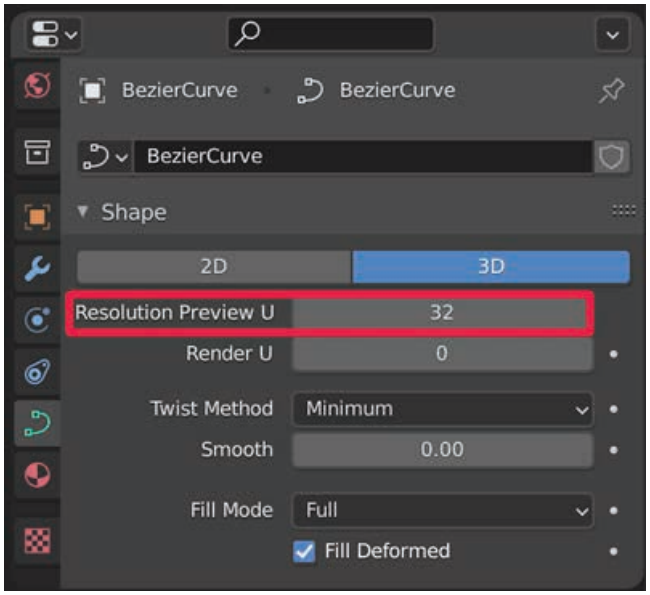


Finally, by adding a **Curve modifier** you can use a **Bezier Curve** to control the cylinder shape. Note that you need to set the Deform Axis to Z. You'll likely need to increase the vertical Array count some more.



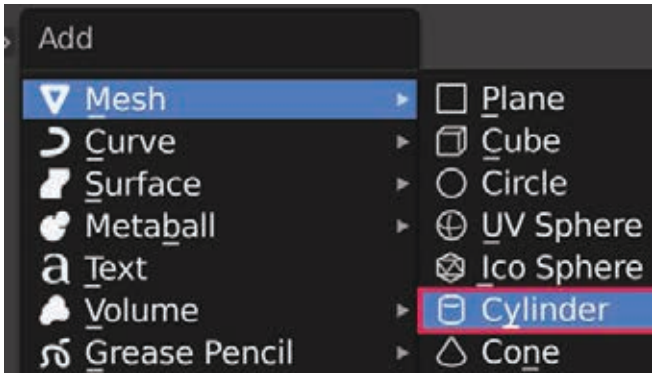


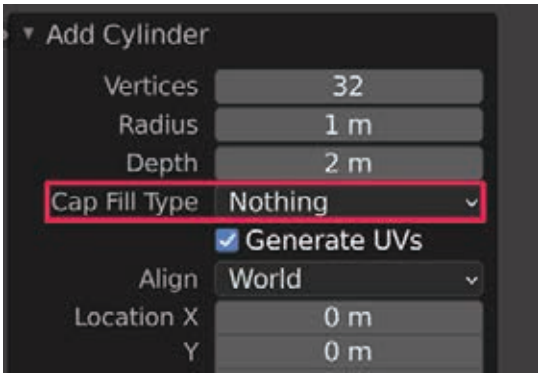
To get a smoother Curve, increase the **Resolution Preview.**



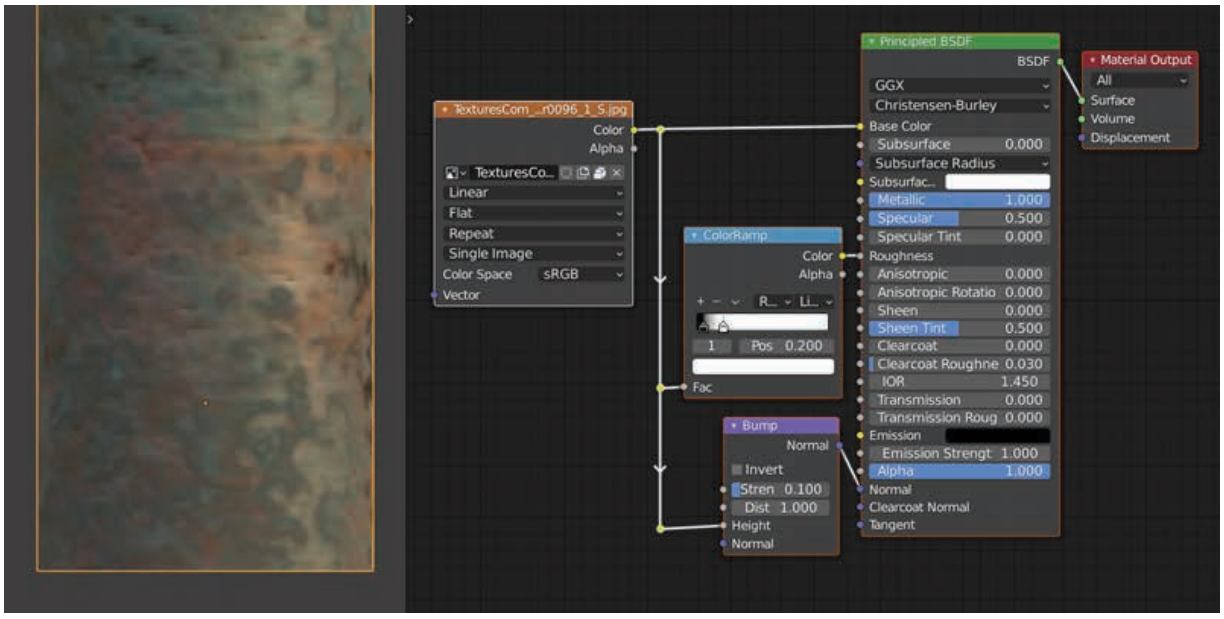
Easy Textured Pipes

Unwrap it and give it some kind of texture.

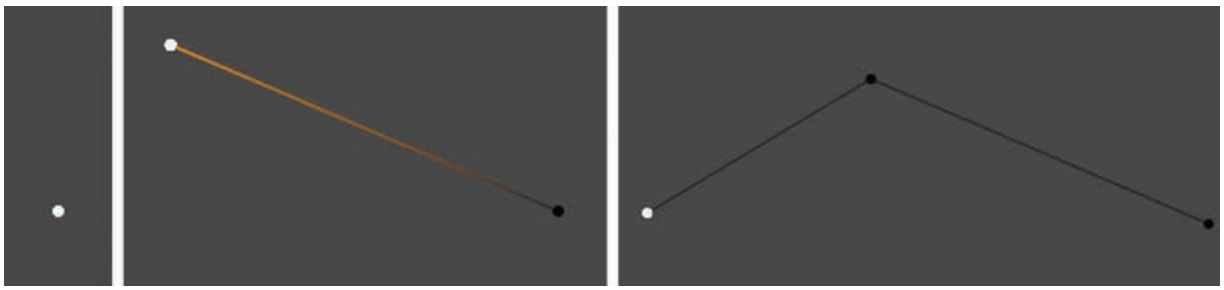




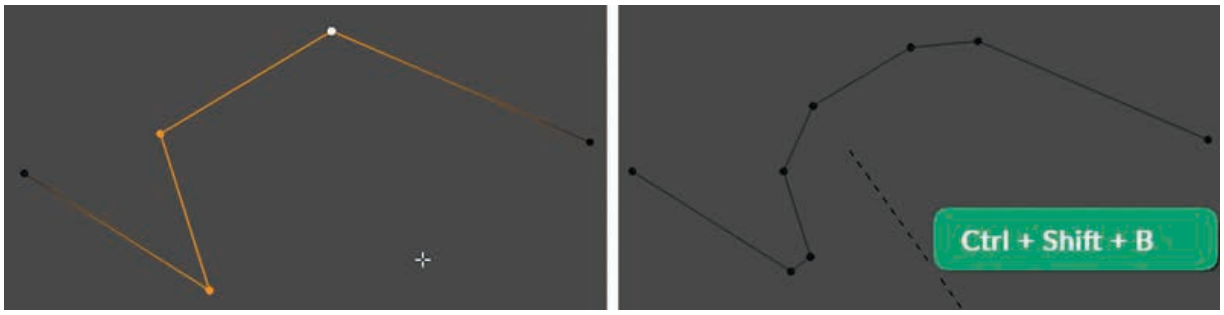
Create a cylinder with no caps. I'm just using some metal texture from Textures.com, and using the image texture for Roughness and Bump as well.



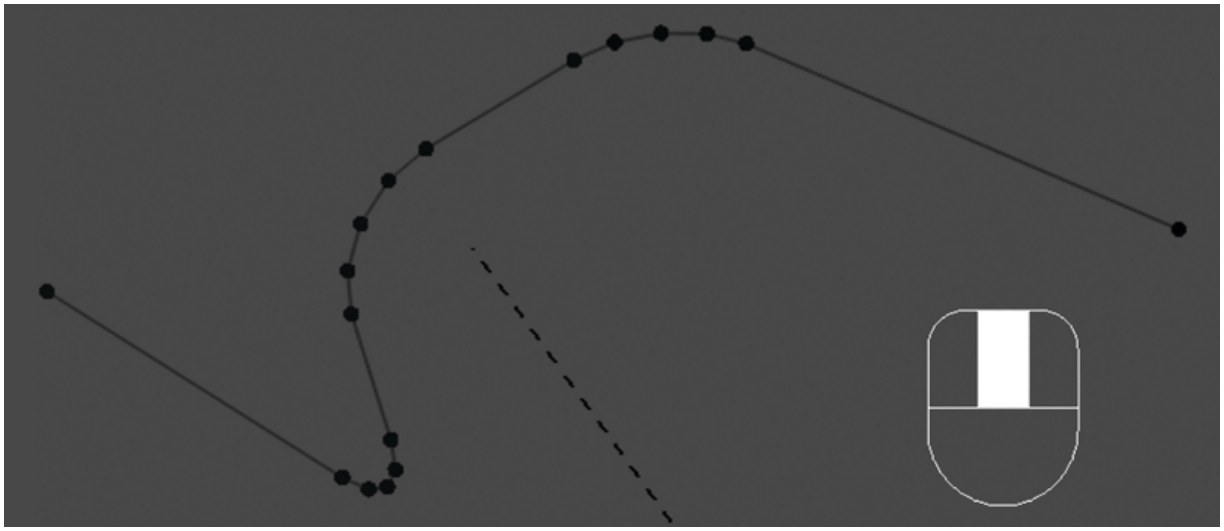
Create a new object, collapse it in Edit Mode to a single vertex by **pressing M > Merge At Center**. Extrude the vertex by **pressing E**.



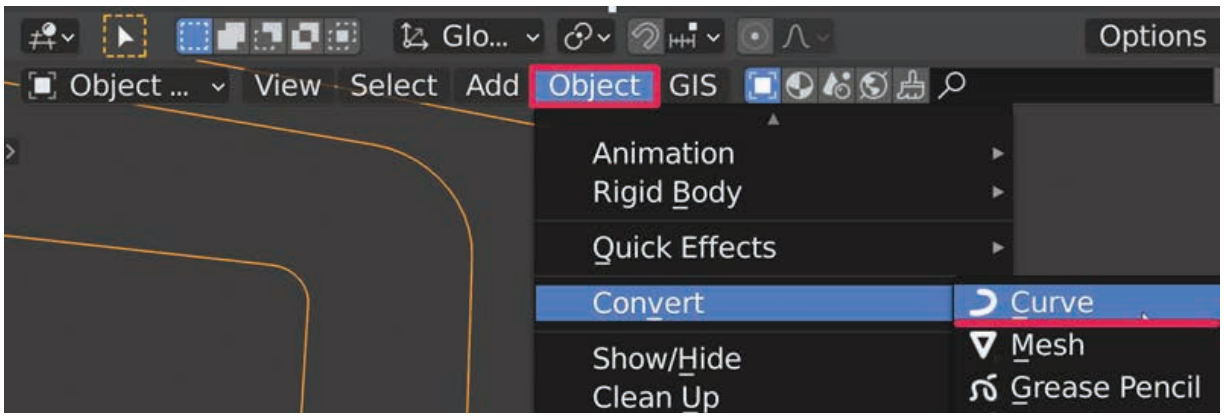
Bevel the vertices with **Shift+Ctrl+B** to make the corners round.



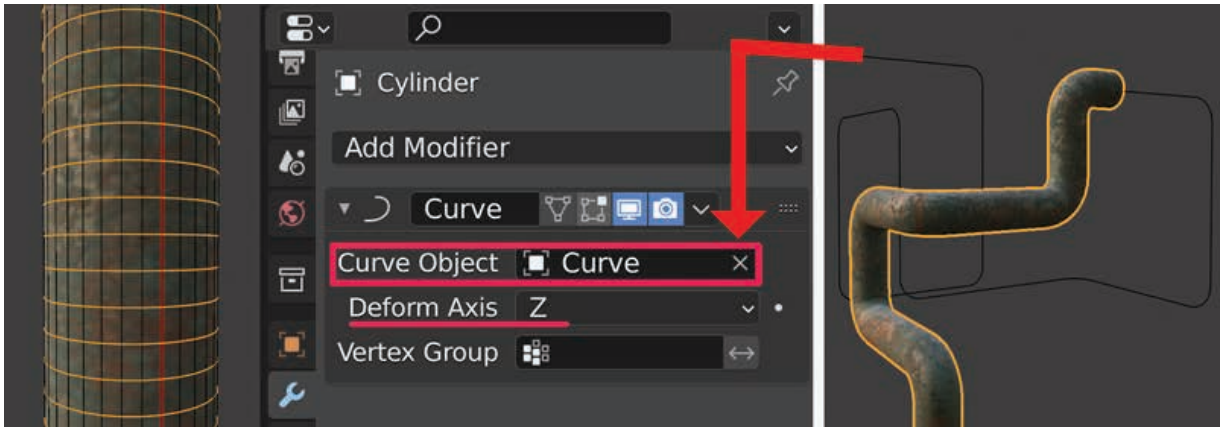
Scroll the middle mouse wheel up to increase the amount of vertices in the bevel.



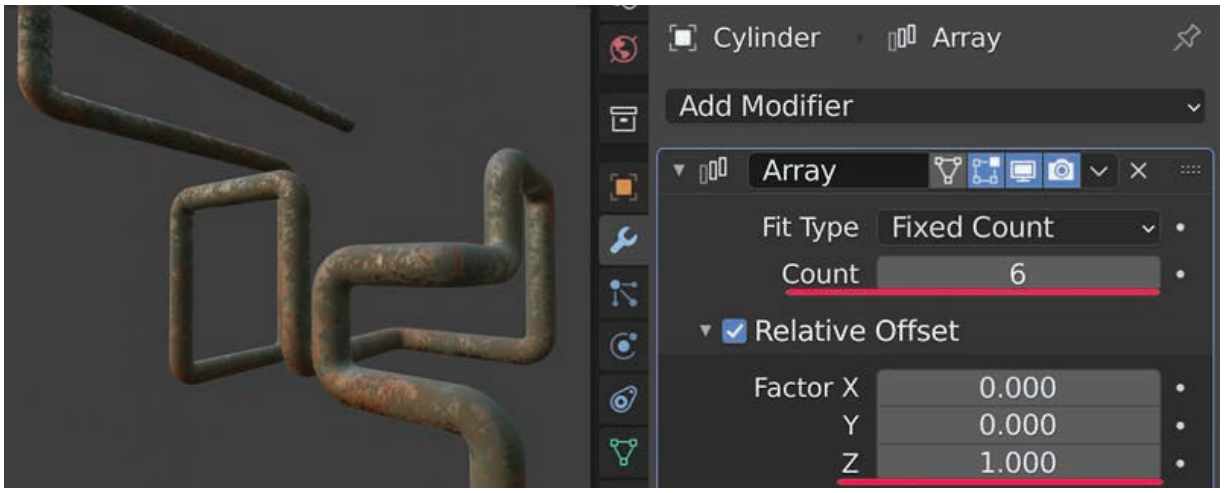
Convert that object into a **Curve**.



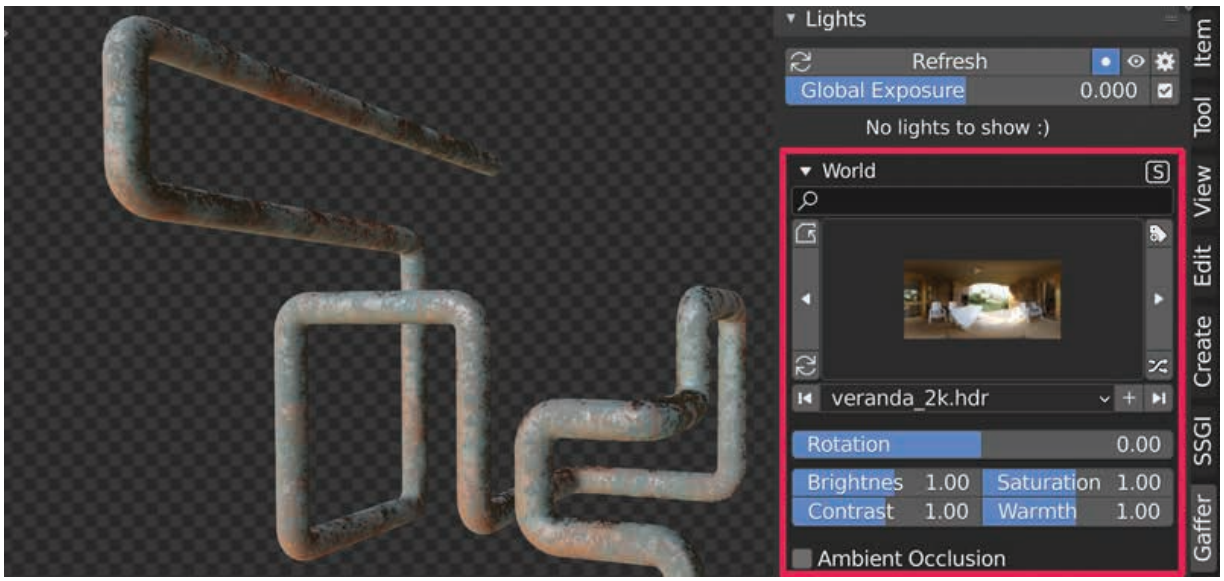
Then add a **Curve Modifier** to the Cylinder. As **Curve Object** pick the Curve. Remember to add some edge loops to your cylinder (with Ctrl+R) or it won't bend at all! Choose the correct **Deform Axis.**



You can add an **Array modifier**, just make sure it's above the Curve modifier. Set it to **Relative Offset** (along the Z-axis in this case) and **Fixed Count**. Increase the **Count** to increase the length of the Cylinder.

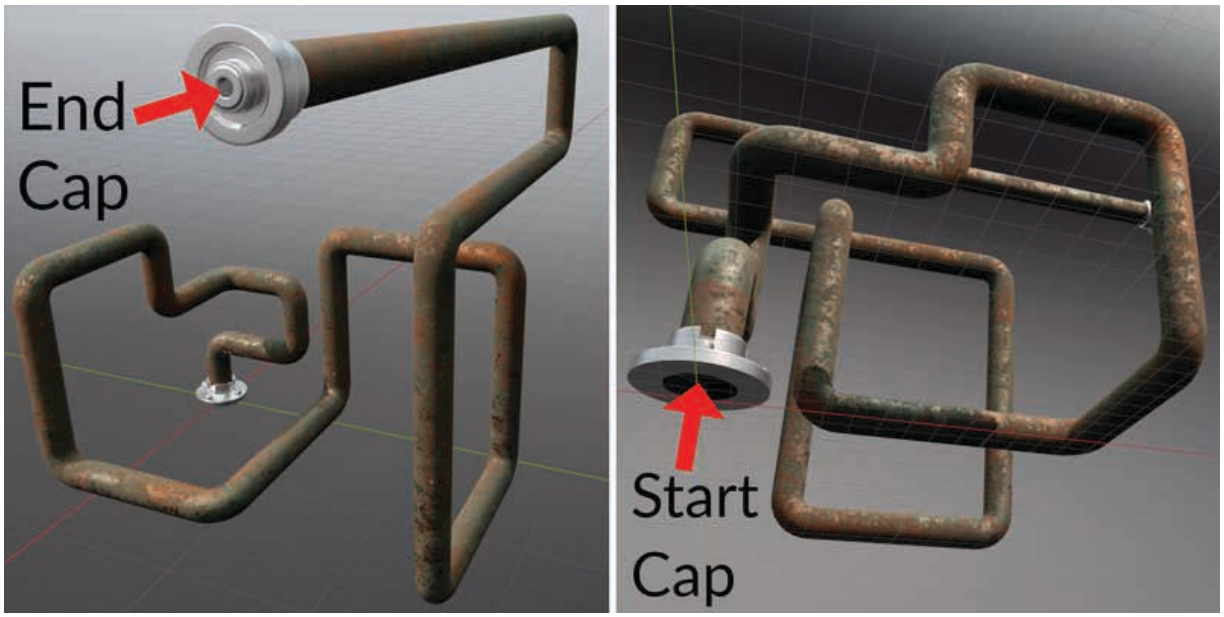


Add an HDRI for some nice lighting. This is the add-on Gaffer by Greg Zaal. I highly recommend this add-on as it lets you download and cycle through HDRI's.

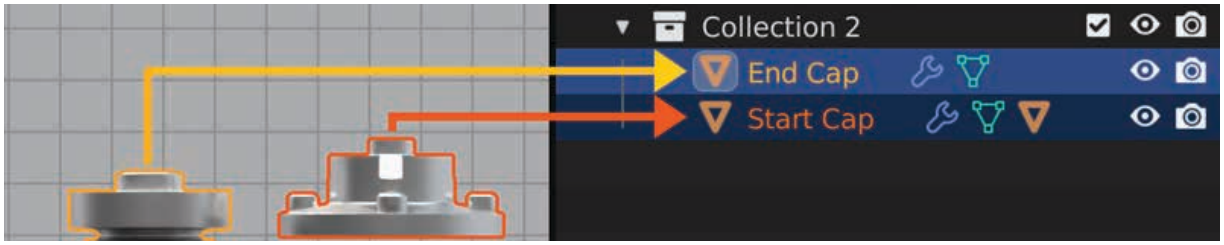


Start and End Caps

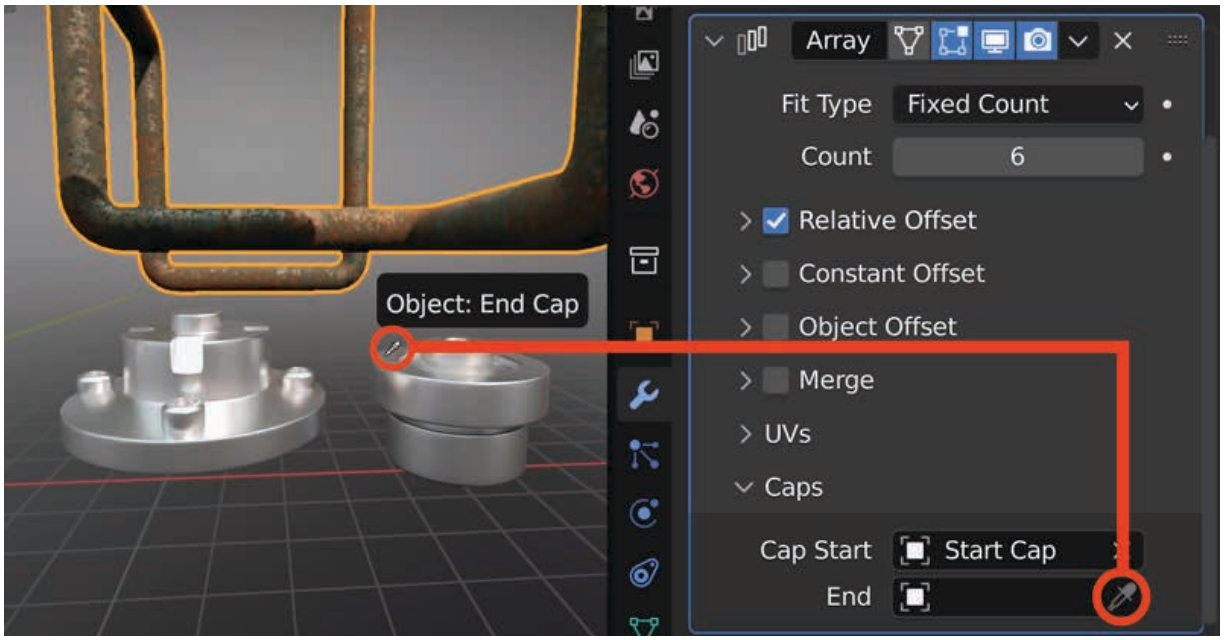
To create some extra detail for your textured cylinder you can add custom Start and End caps.



First model the Start and End Cap objects and give them easy to recognise names.

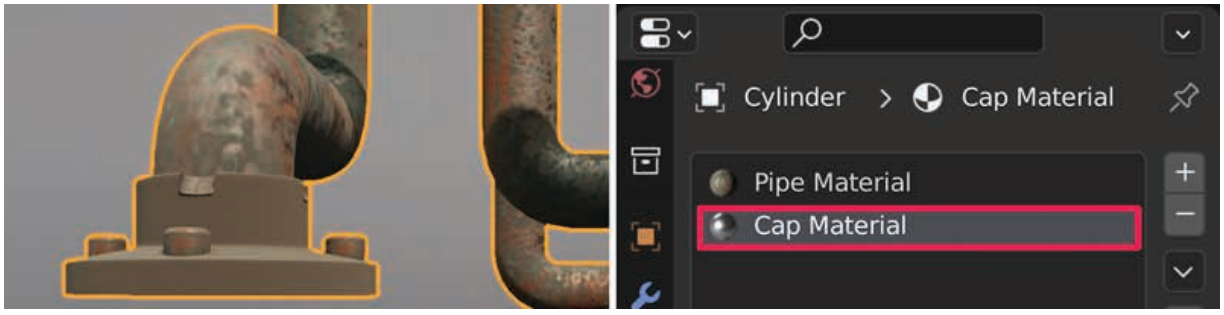


In the Array modifier set the Start and End cap objects that you just modeled.

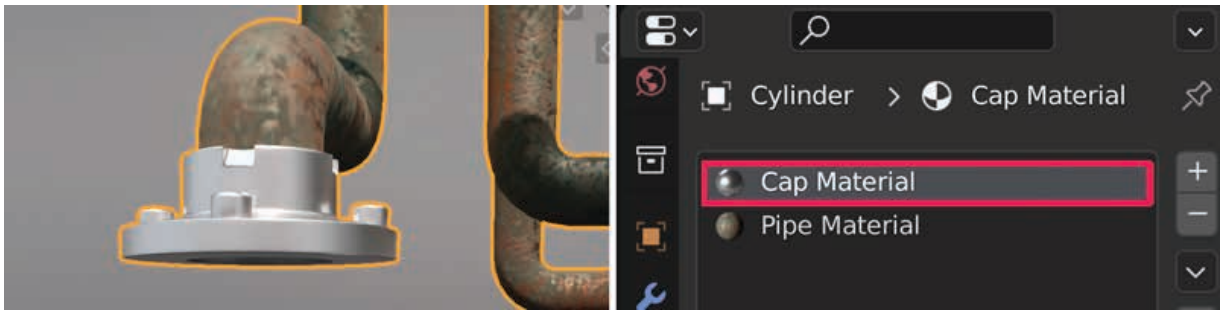


If the caps appear in the wrong place, try going into Edit Mode for that cap object, selecting all vertices and moving them until the cap lines up.

If the caps have a different material from the cylinder, add a material in the material slot and load the material of the caps.

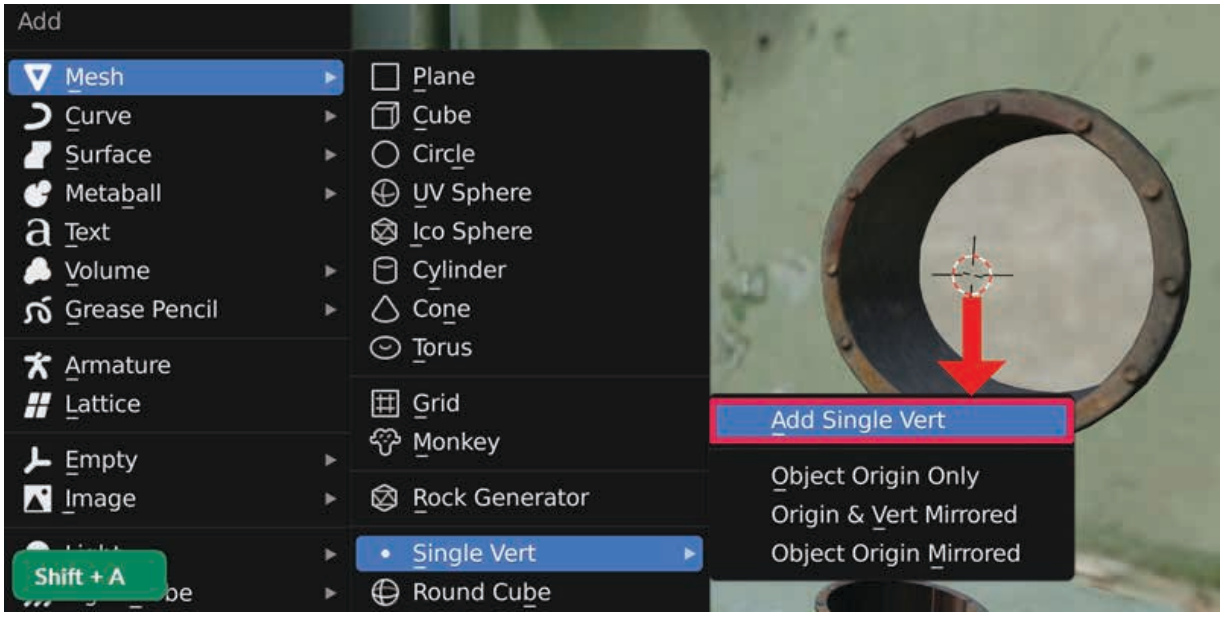


If you don't see the material on your caps, try changing the order of the materials.

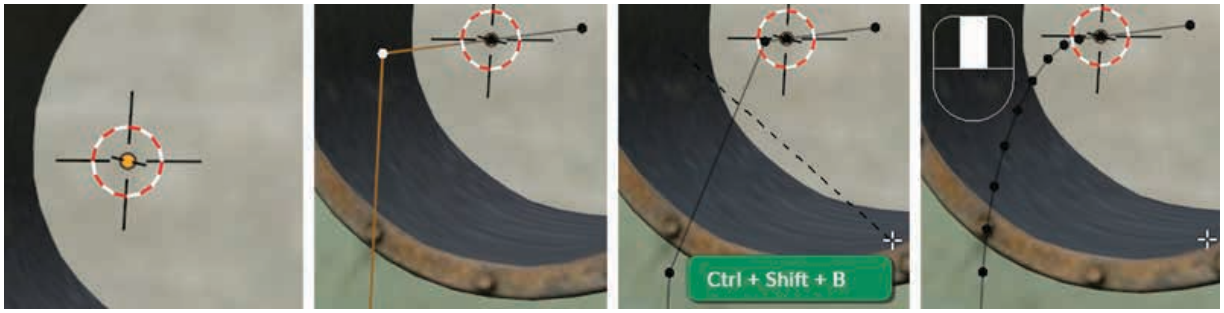


How to add multiple cables at once

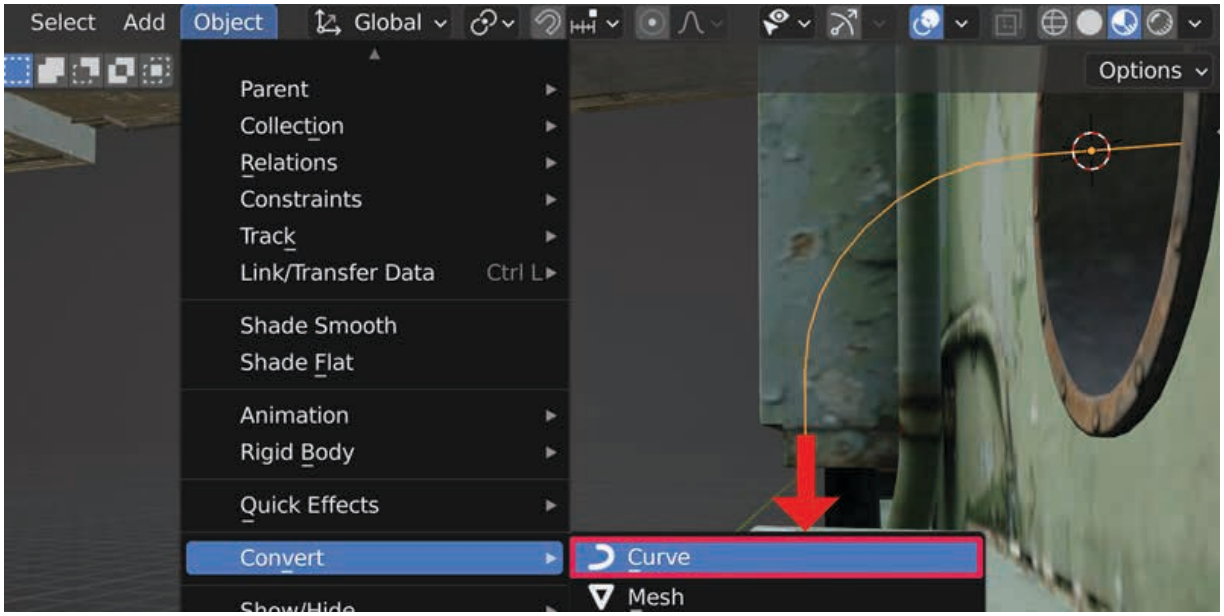
Create a Single **Vertex** and extrude it by **pressing E**. Note that the **Single Vertex** option is available once you've enabled the **Extra Mesh Objects** add-on in Preferences.



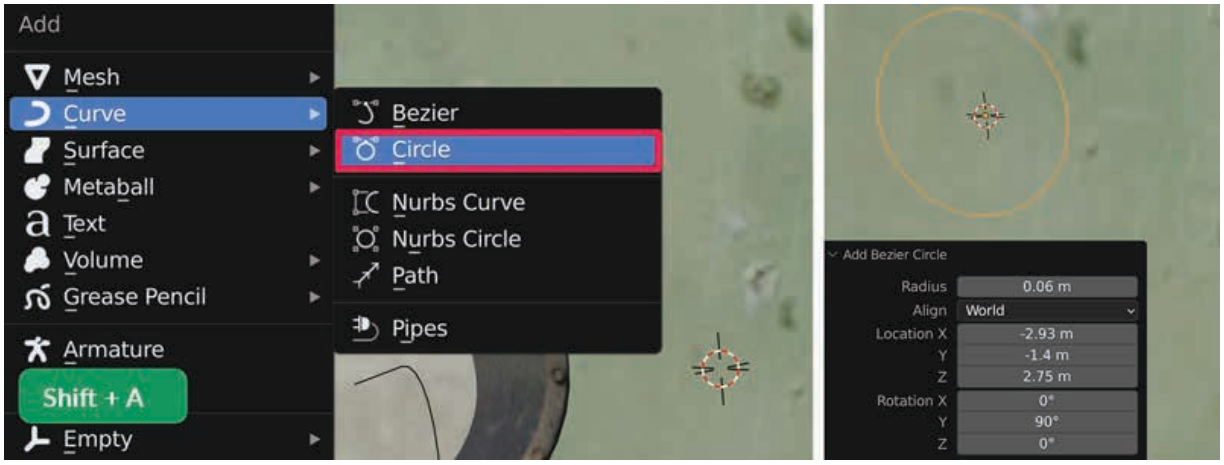
Then, bevel it to create a bend in the middle by **pressing Shift+Ctrl+B**. Scroll the middle mouse wheel up to increase the amount of vertices.



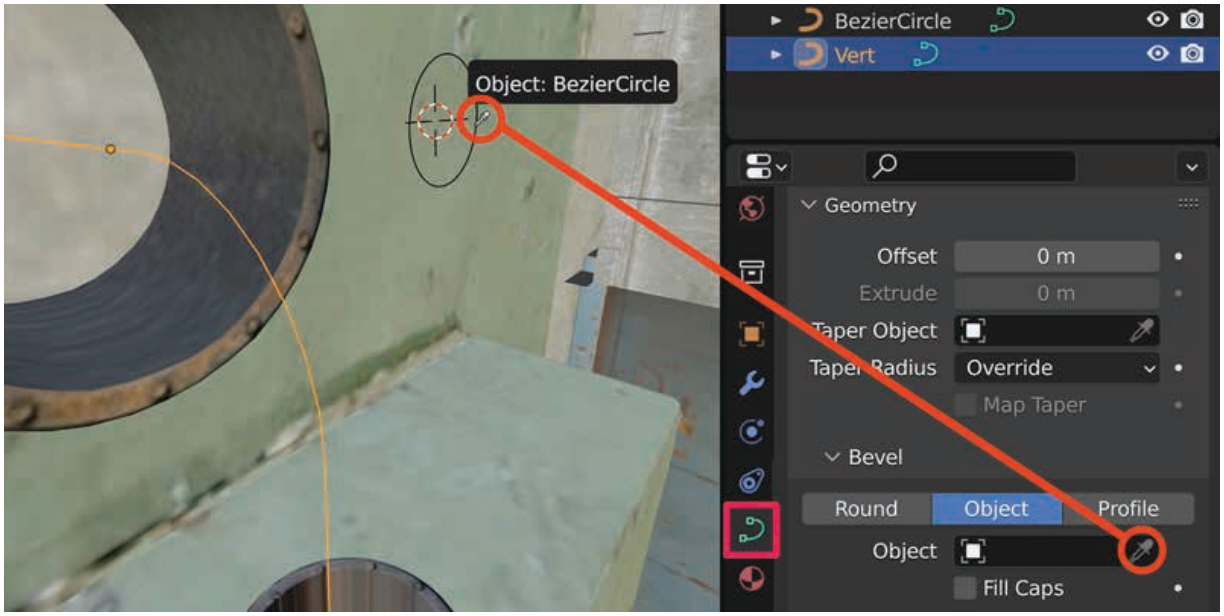
Go to Object > Convert To > Curve from Mesh.



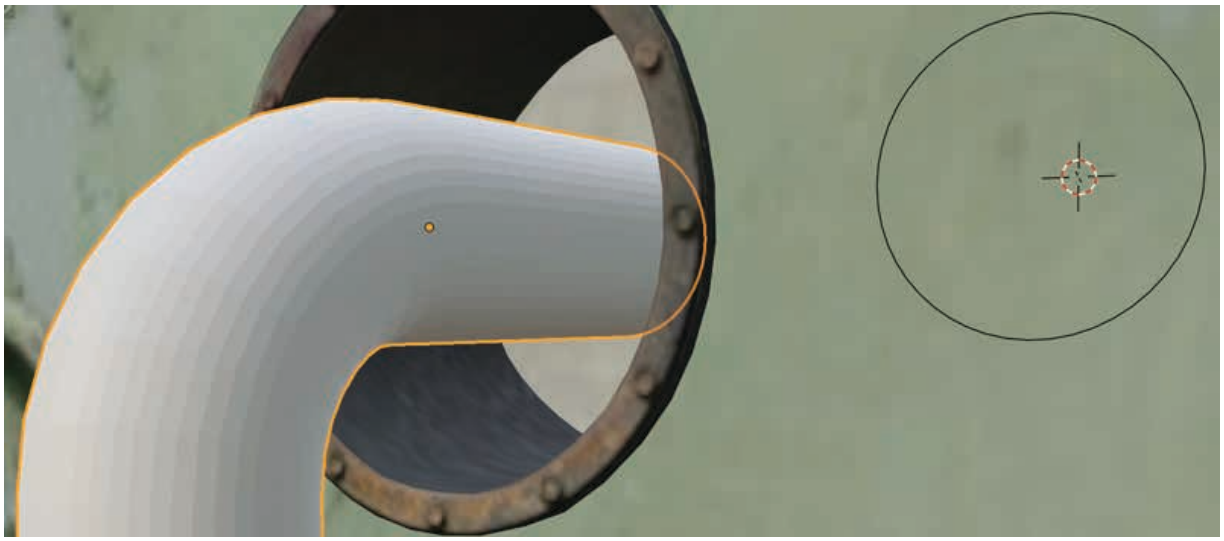
Add a **Curve > Circle** and transform it so that you can see it easily.



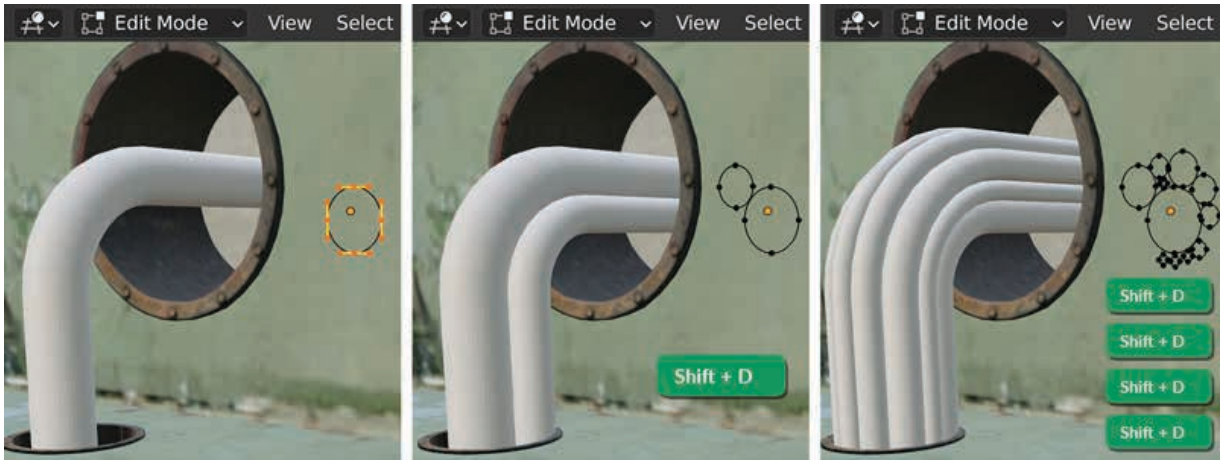
Choose the Circle as the Bevel object for the other cable object.



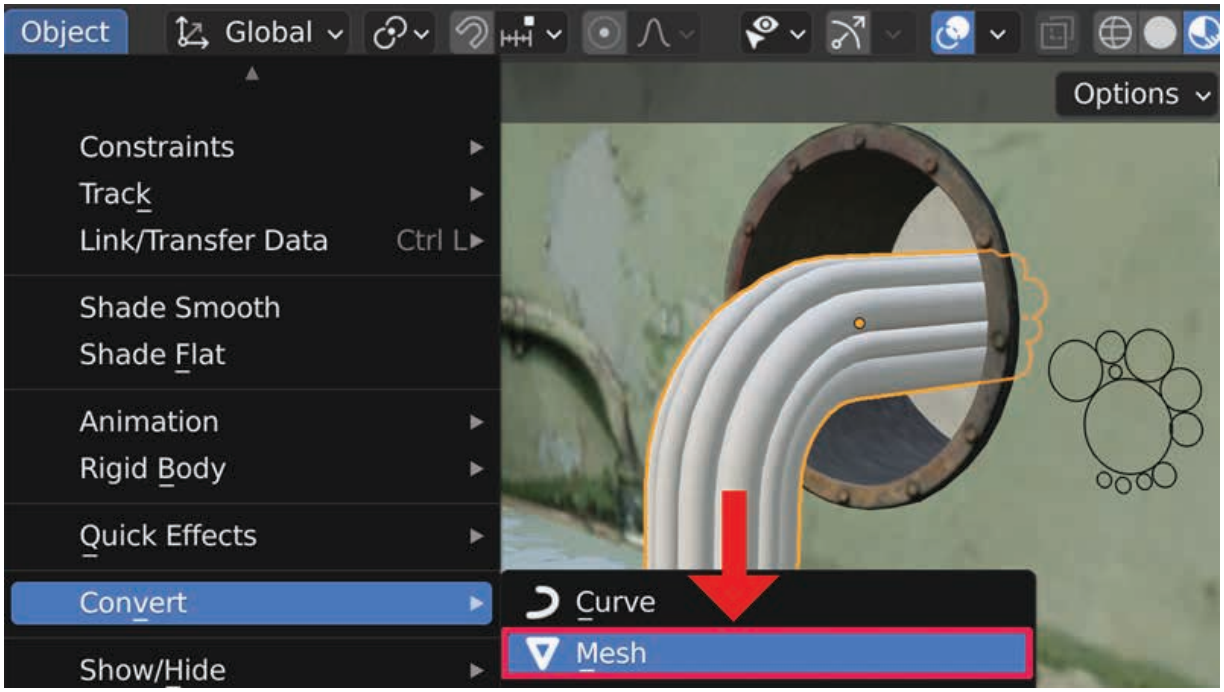
The shape of the Circle now controls the shape of the cables.



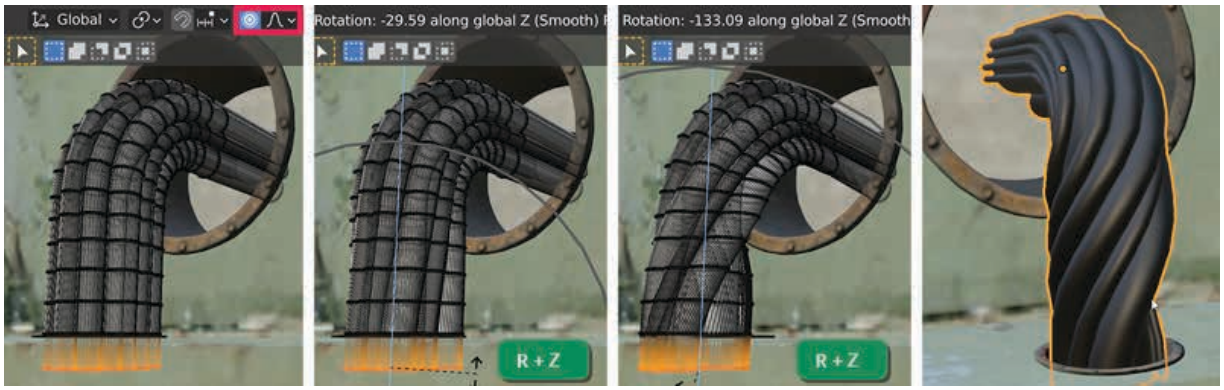
In Edit Mode duplicate and transform the Circle to add more cables.



You can go to **Object > Convert** and convert it back to a **Mesh** object.



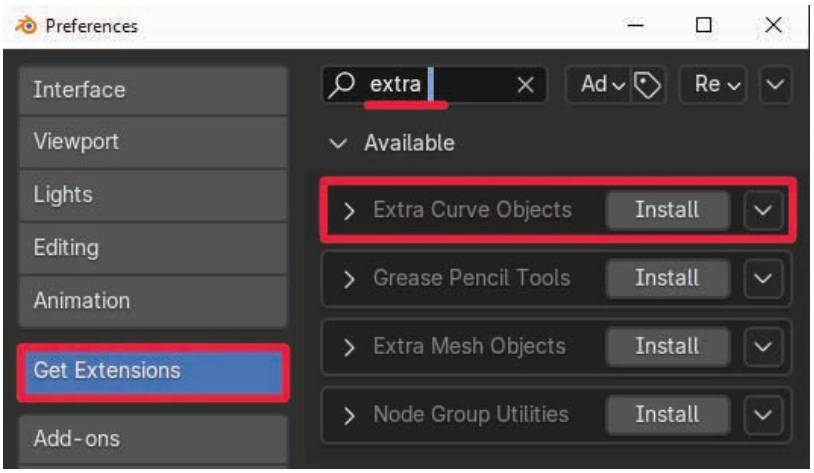
Now you can use Proportional Editing to give it some interesting twists.



Hanging cables (Catenary Curves)

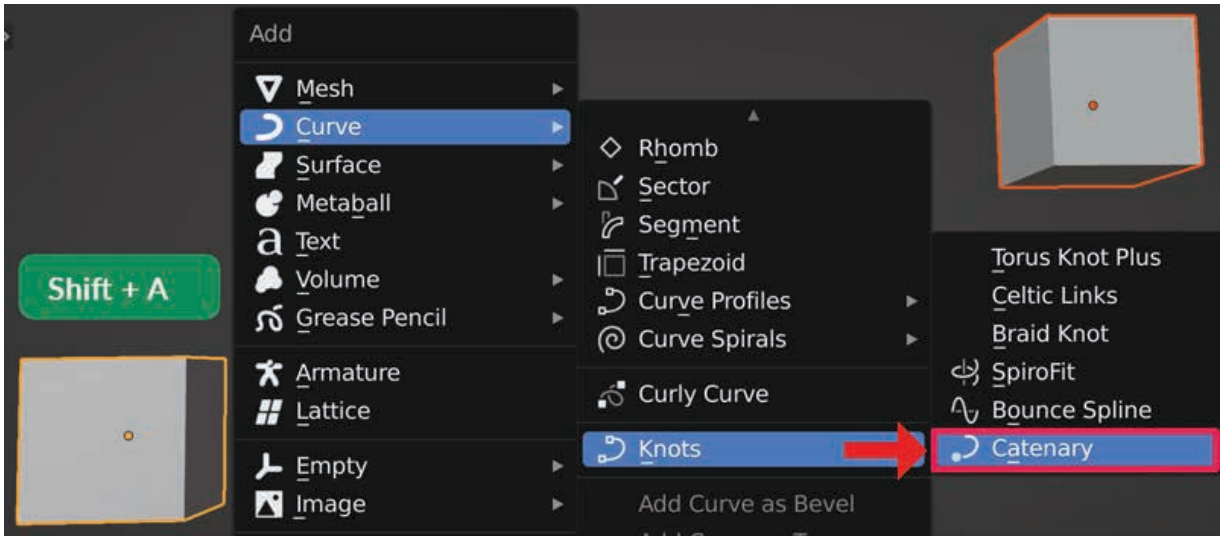
First, install the **Extra Curve Objects** add-on.

You can find it in the **Get Extensions** tab in Preferences.

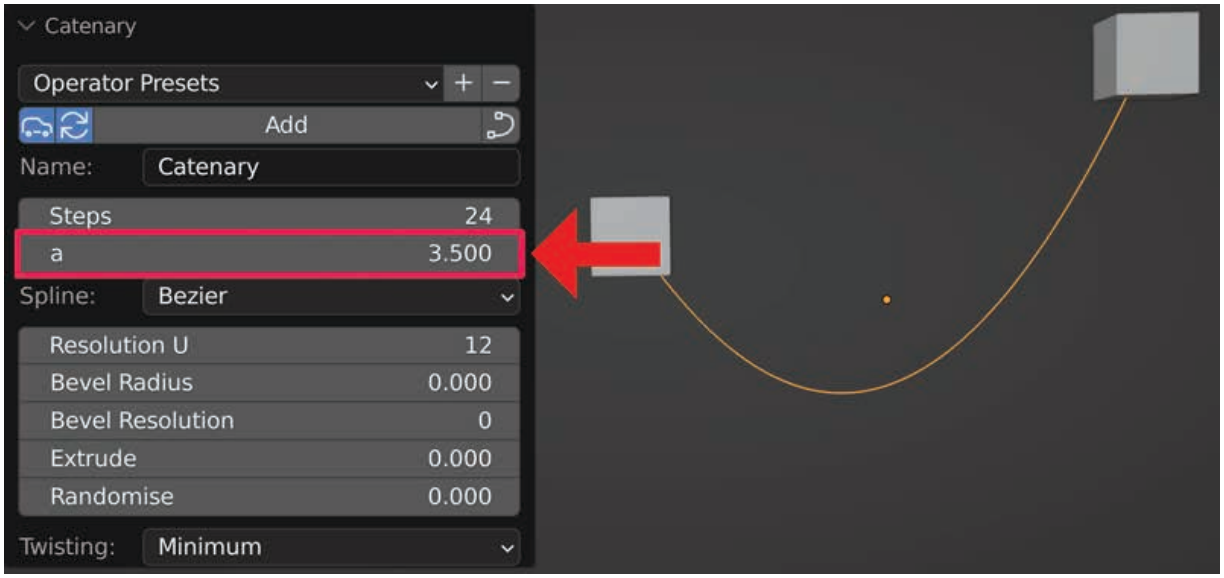


Select two objects between which you want to have hanging cables.

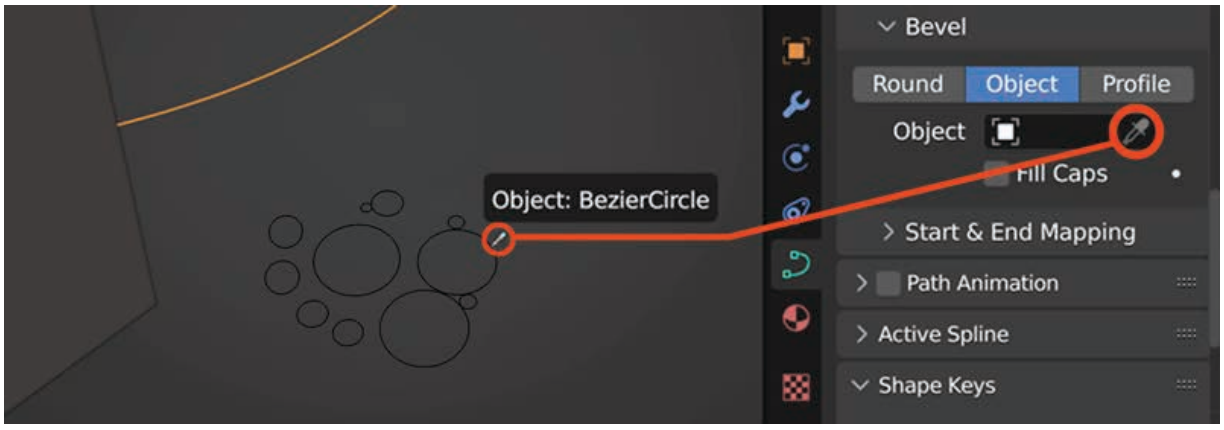
Press Shift+A and choose **Curve > Knots > Catenary.**



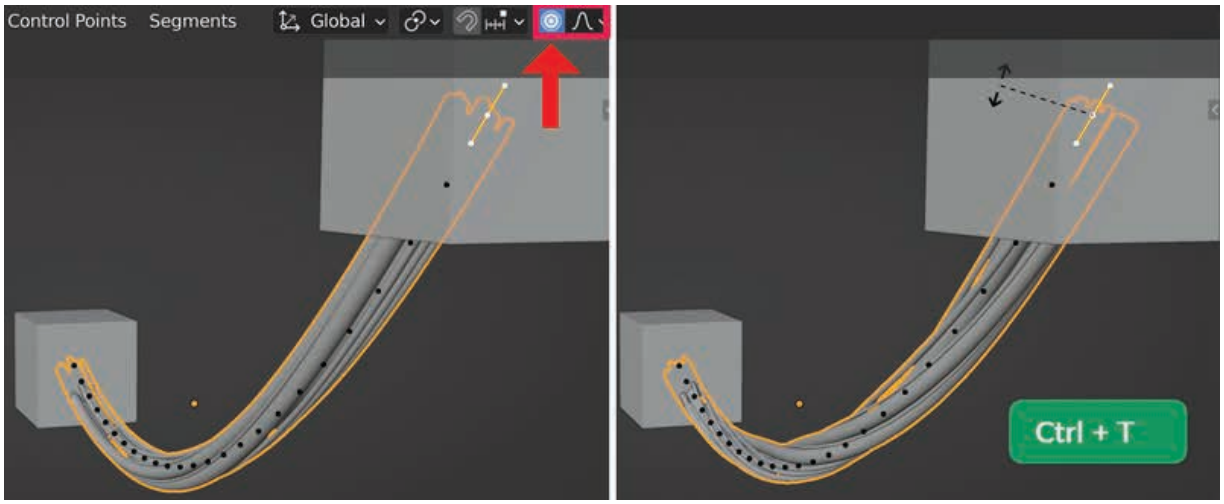
This creates a “hanging” Curve between the two objects. Change the “a” variable to decrease how much the Curve droops.



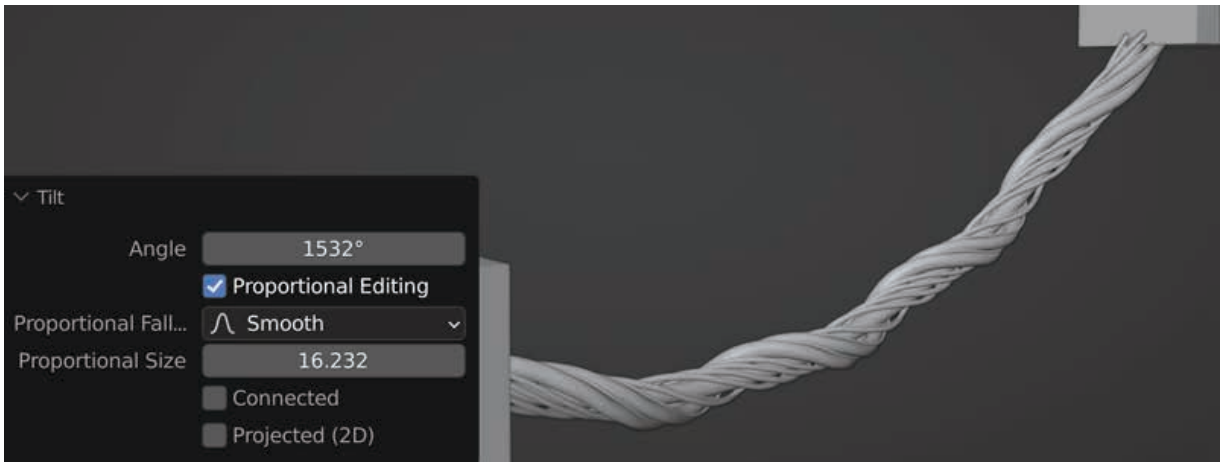
If you want to use a custom Bevel object enter it in the Bevel Object field.



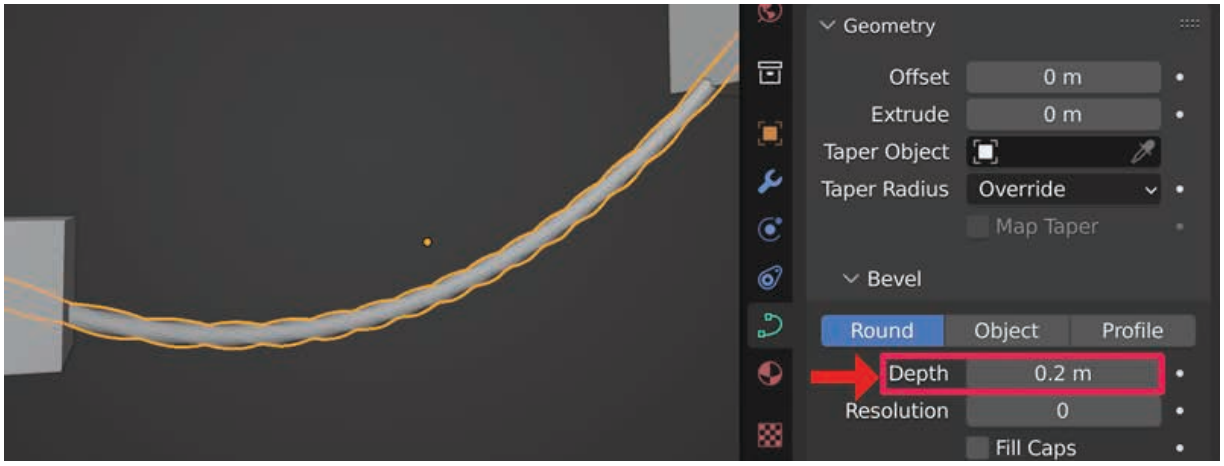
With Proportional Editing enabled, **press Ctrl+T** to tilt the Curve.



You can also control this **Tilt value** from the menu in the lower left corner of the viewport after **pressing Ctrl+T**.



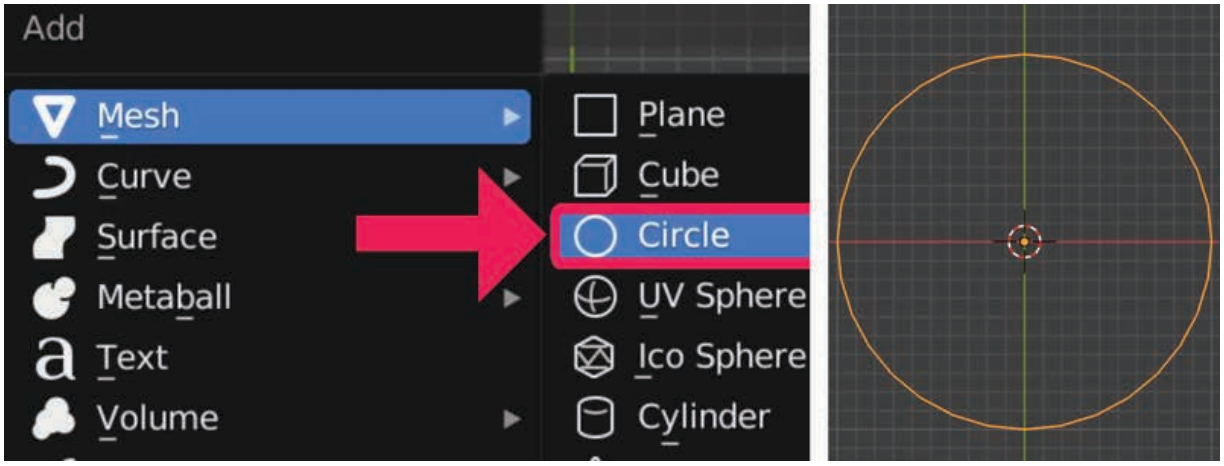
For a simple cable, you don't need any Bevel object, just increase the **Depth** value.



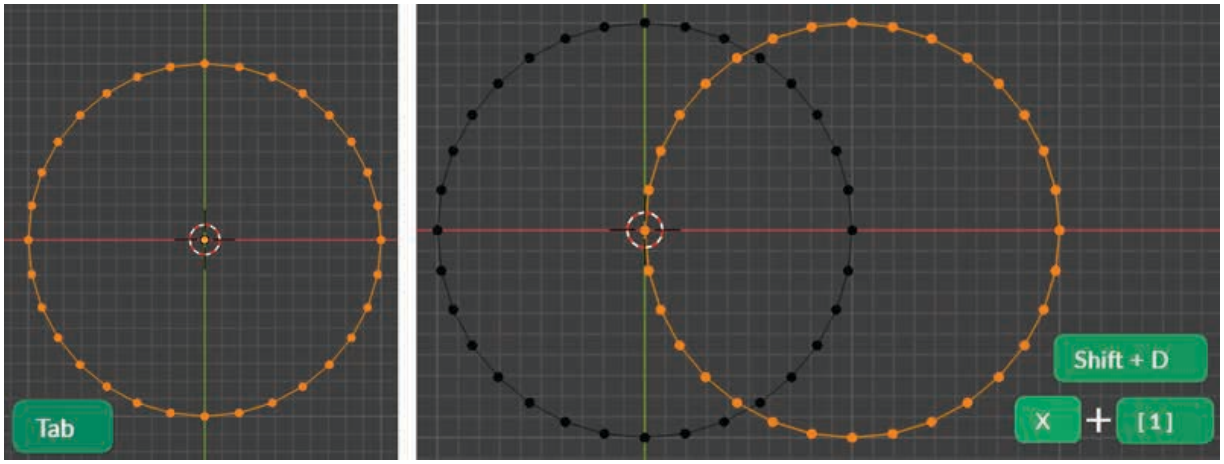
Model a Rope along a Curve

This is a quick way to model a rope, such as the one used in the topic Complex Rope Mesh Physics with the Surface Deform modifier.

Add a Circle with **Shift+A > Mesh > Circle**.

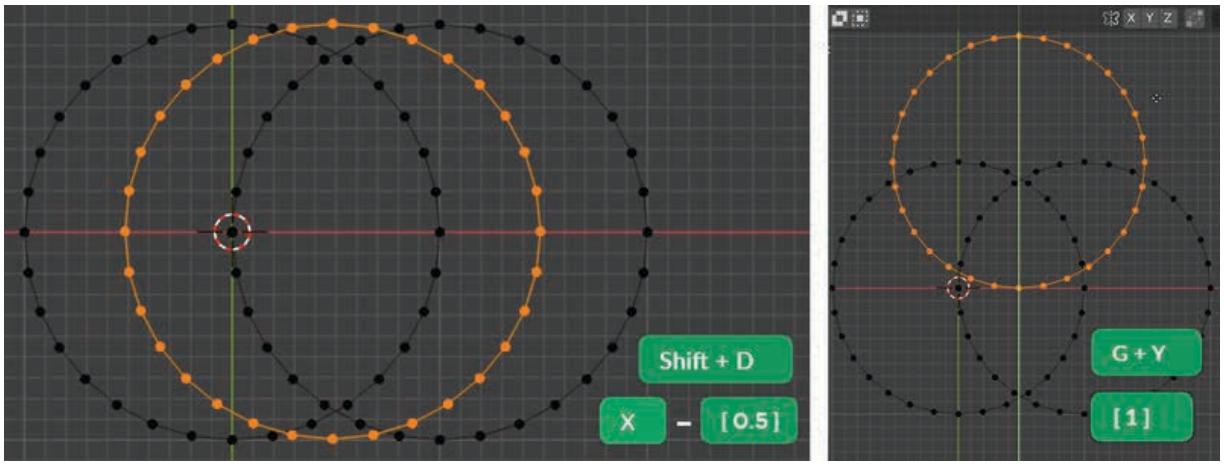


In Edit Mode duplicate it with **Shift+D** and **press X+[1 on the numpad]** to move the duplicate 1 Blender unit to the right on the X-axis.

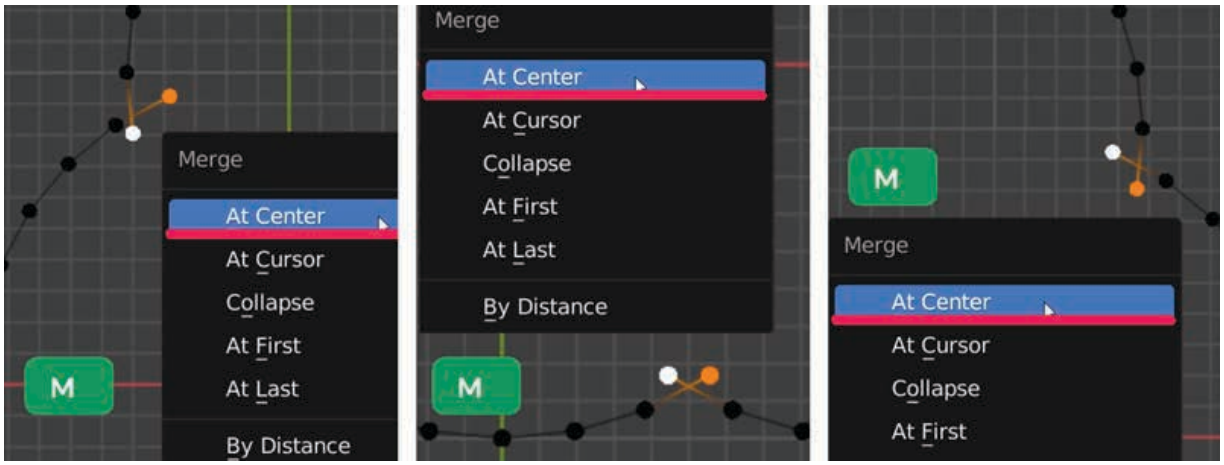


Click here to return to the index.

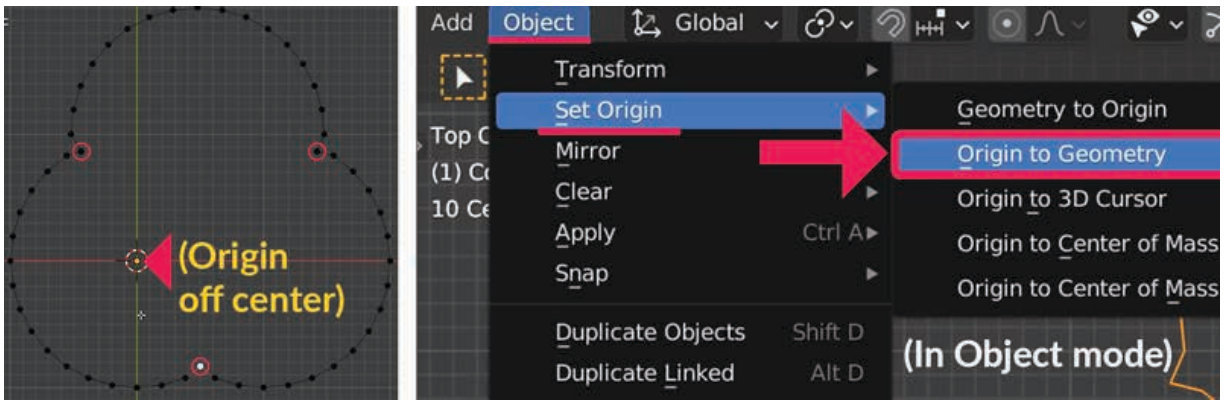
Duplicate the selection again with **Shift+D** and **press X+[-0.5 on the numpad]** to move it half a Blender unit on the X-axis in the negative direction. Then **press G+Y+[1 on the numpad]** to move it up one unit on the Y-axis. Of course you can also just do this by eye, if you want.



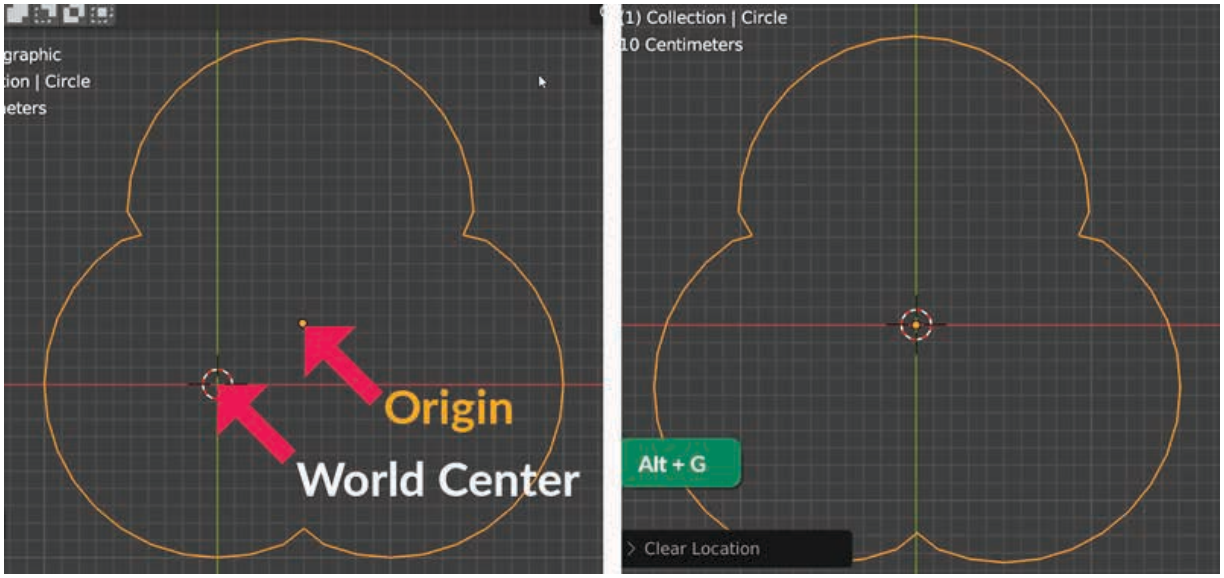
Remove the inner vertices. An easy way to select them is by **pressing C** for Circle Select mode. Then **right-click** to exit that mode. Merge where needed by selecting two vertices and **pressing M > At Center.**



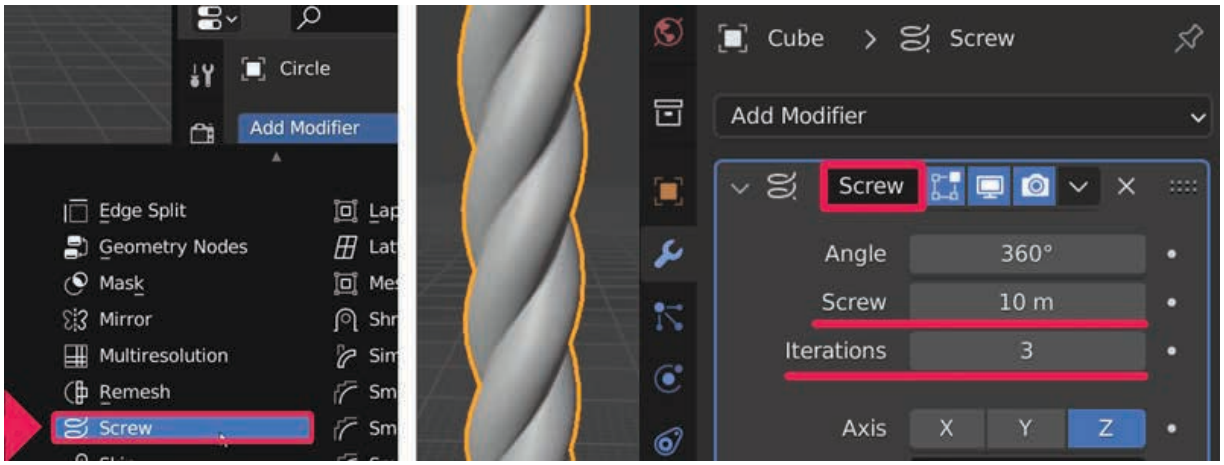
Go to **Object > Set Origin > Origin to Geometry** to place the Origin point in its center.



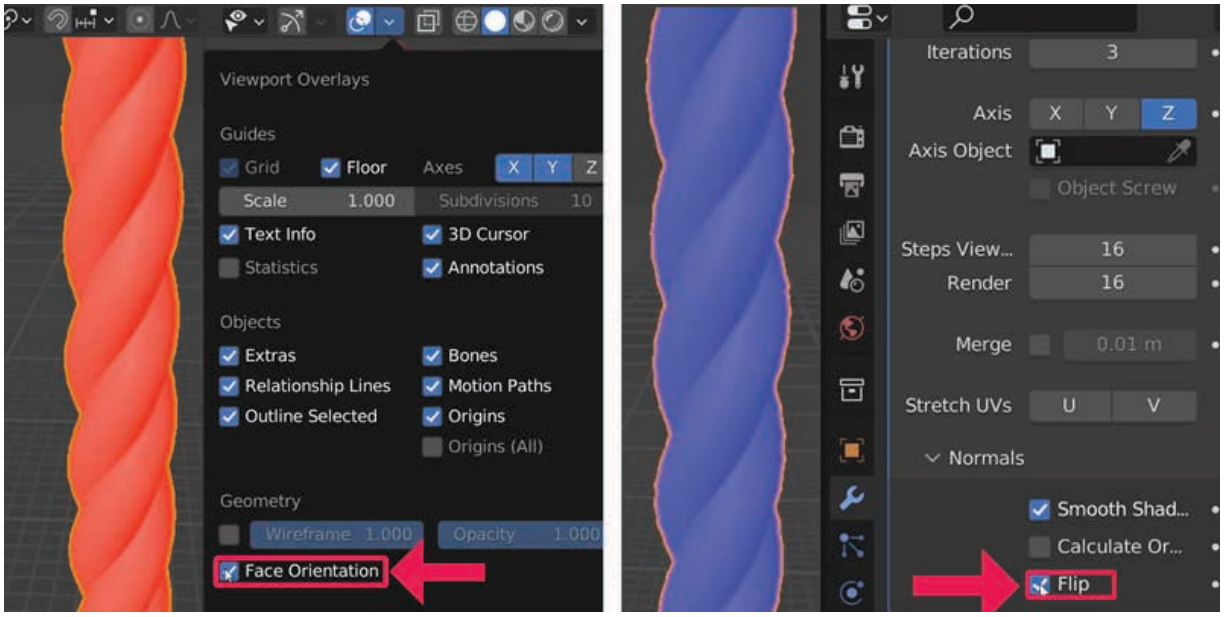
Press Alt+G to reset its location.



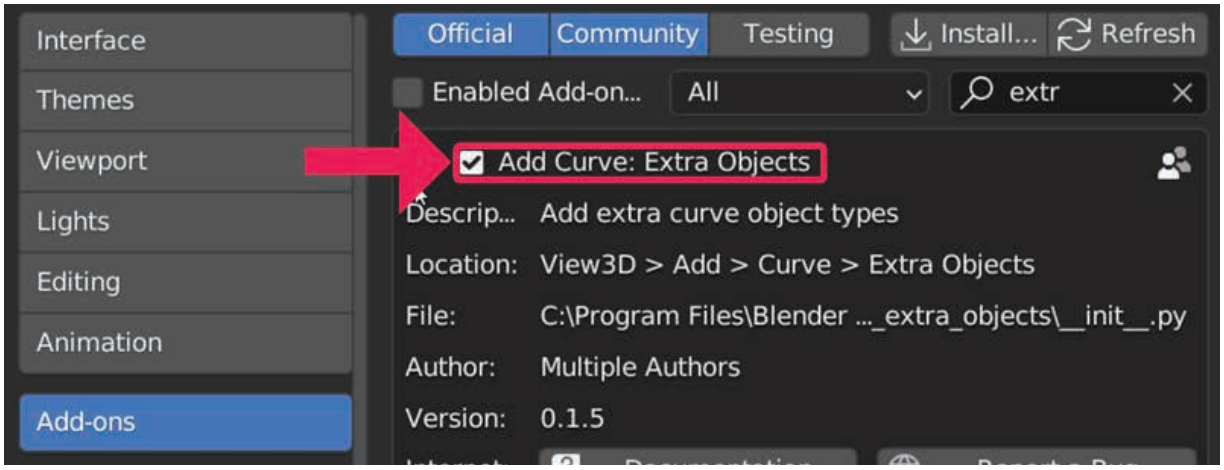
Add a **Screw modifier** and increase the **Screw** value and the **Iterations.**



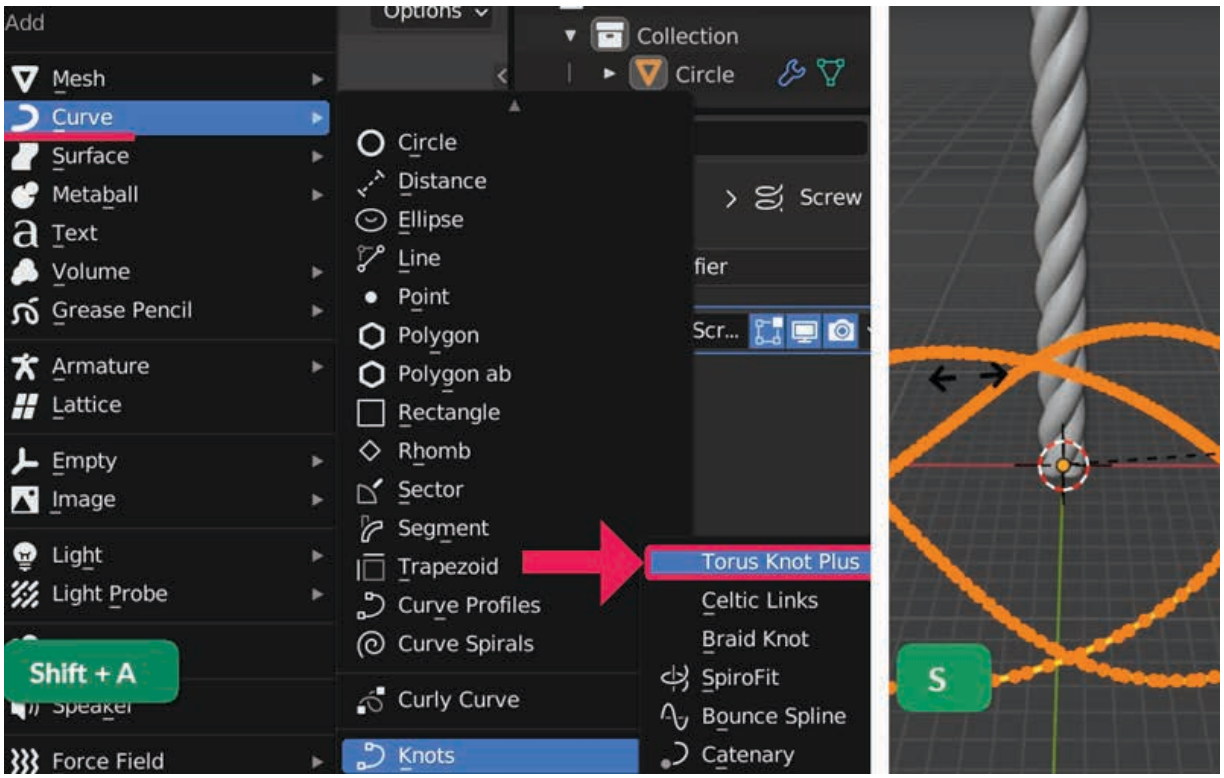
You may need to check **Flip** in the **Normals** options of the modifier. The outside should be blue, not red, when checking **Face Orientation** in **Viewport Overlays**.



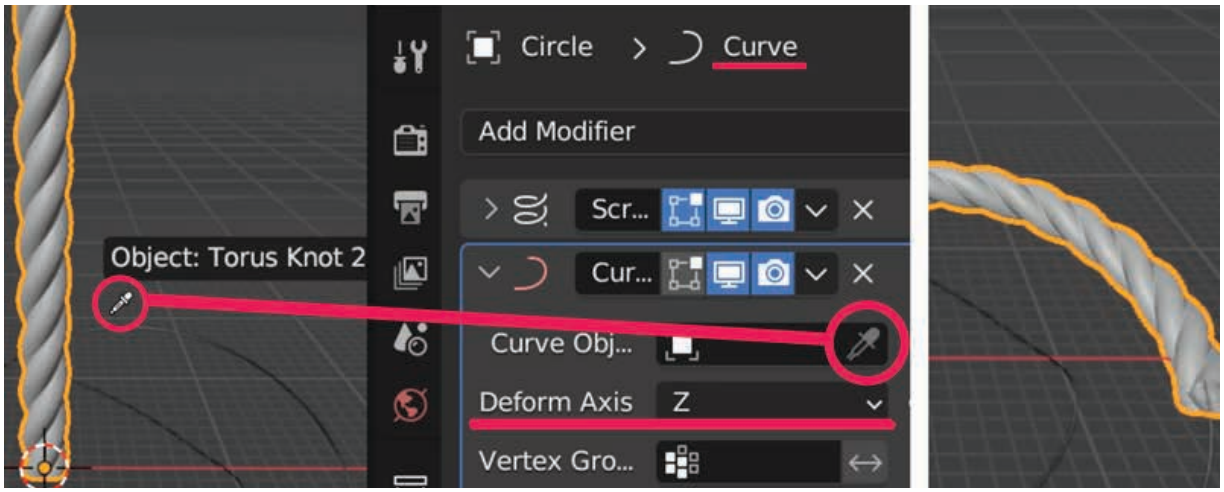
Enable **Add Curve: Extra Objects** in Preferences for more Curve options.



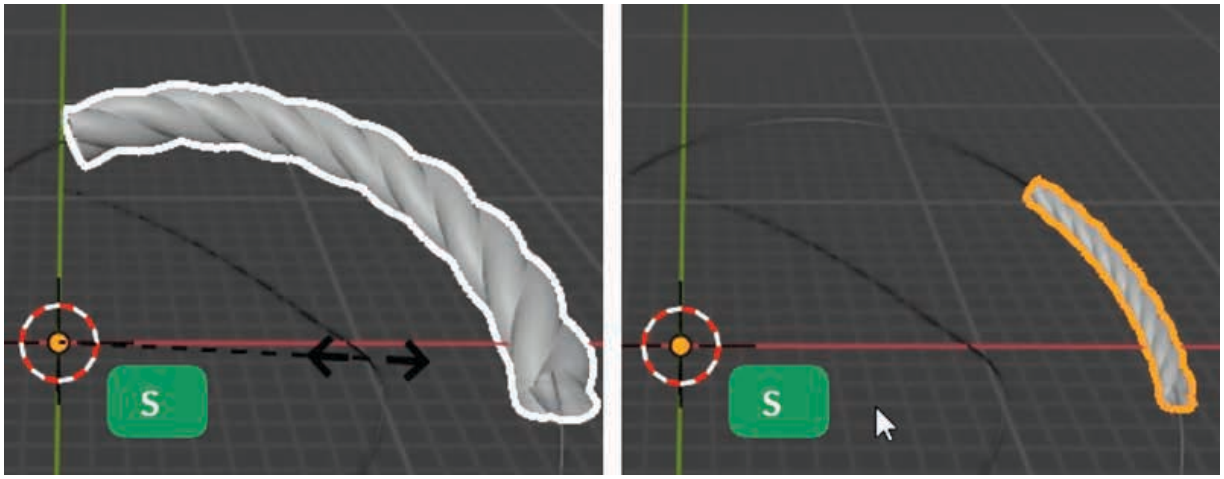
Then, add a Curve and scale it up.



Add a **Curve modifier** to the rope and pick the **Curve Object**. Set **Deform Axis to Z**.

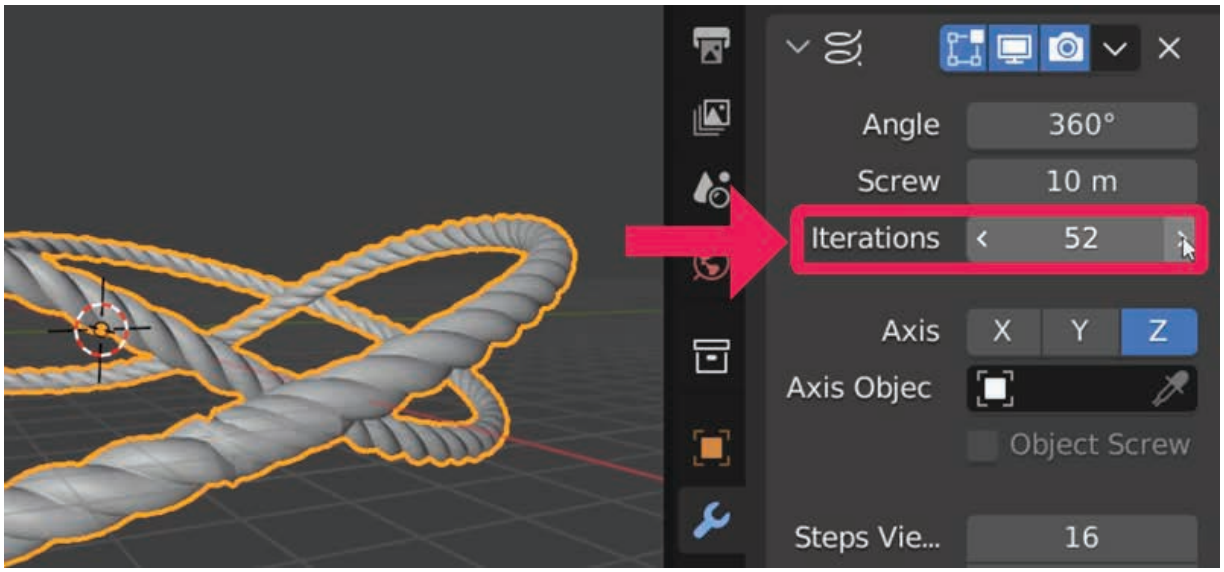


You may have to scale the rope up or down by **pressing S** in Object Mode.



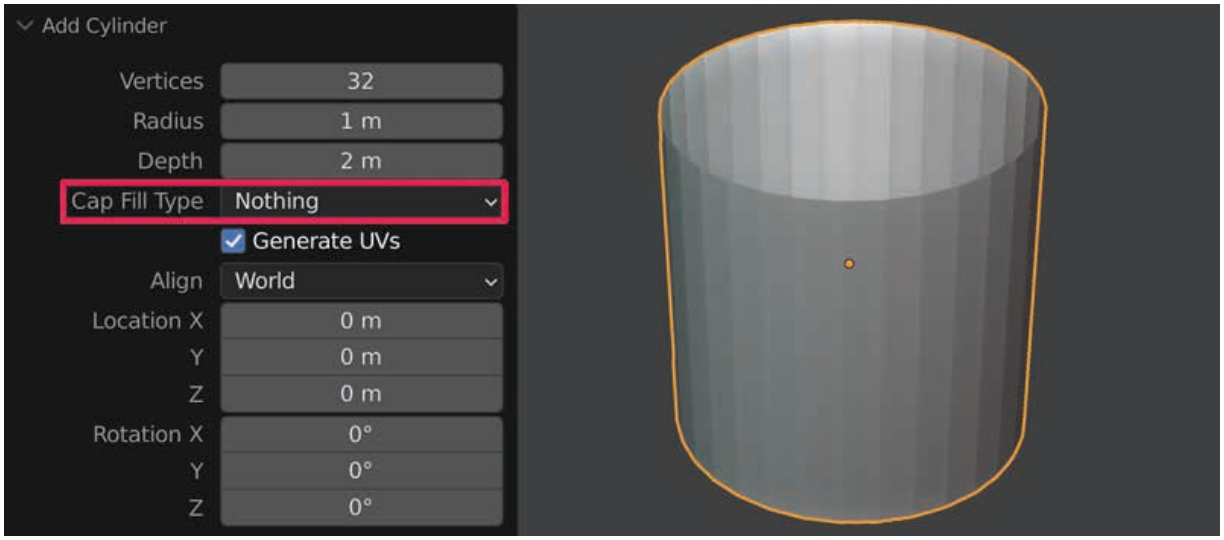
And then move it along the Z-axis by **pressing G+Z**.

You may have to increase or decrease the **Iterations** accordingly.

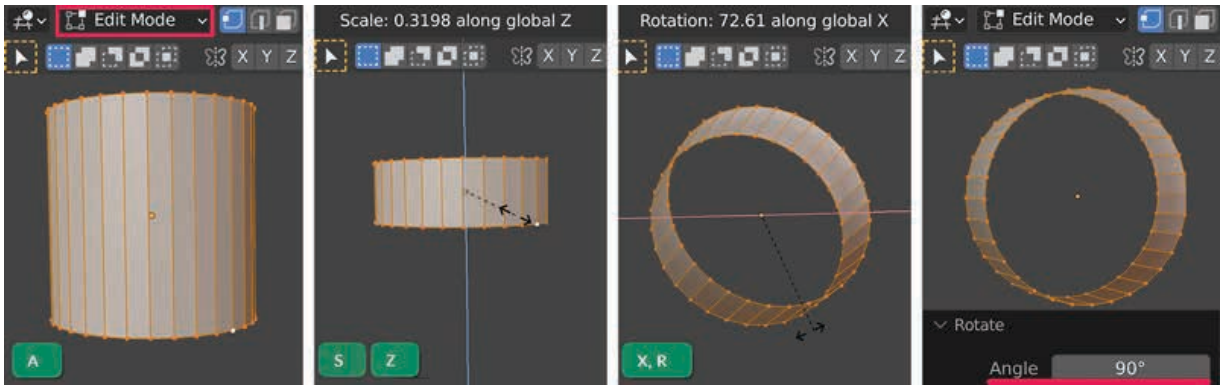


Create Corrugated Pipes

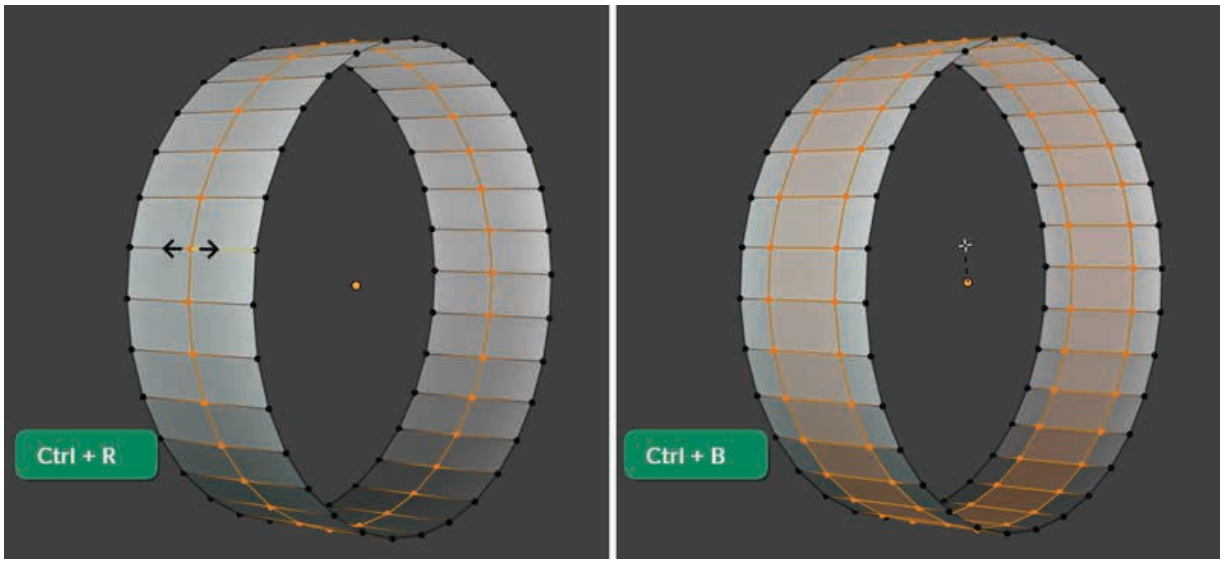
Add a Cylinder by **pressing Shift+A** and choosing Mesh > **Cylinder.** As Cap Fill Type choose “Nothing”.



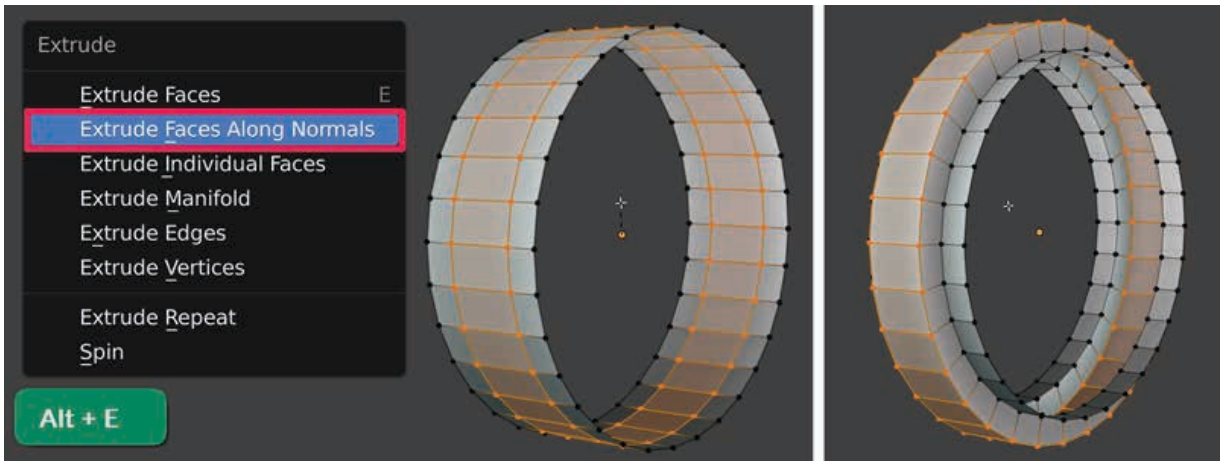
In Edit Mode select all. Scale it along the Z-axis and rotate it.



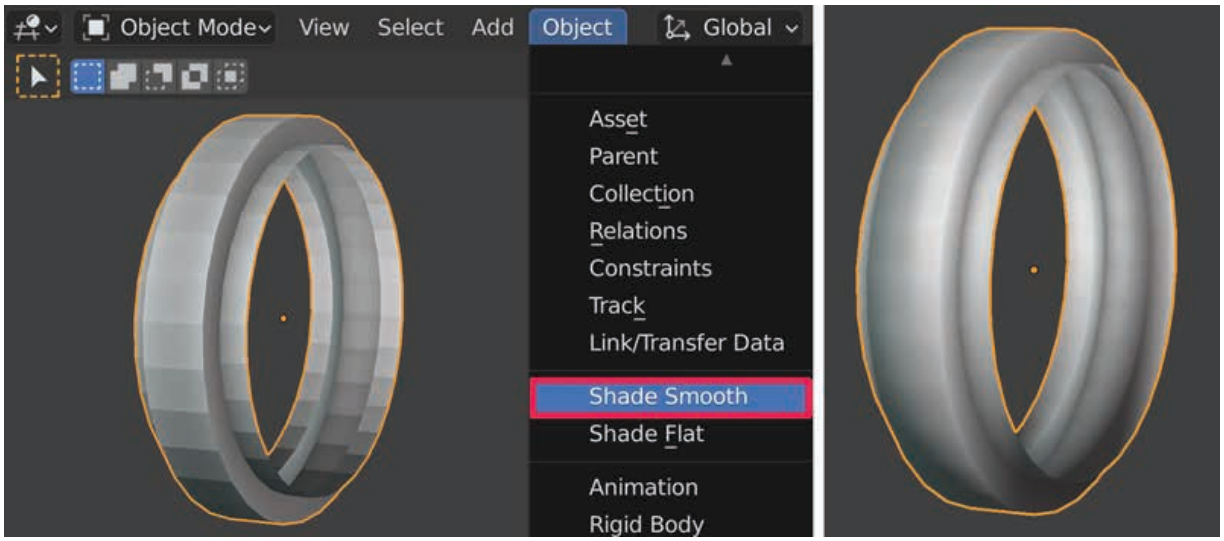
Press Ctrl+R to add an Edge Loop. **Press Ctrl+B** to Bevel the Edge Loop.



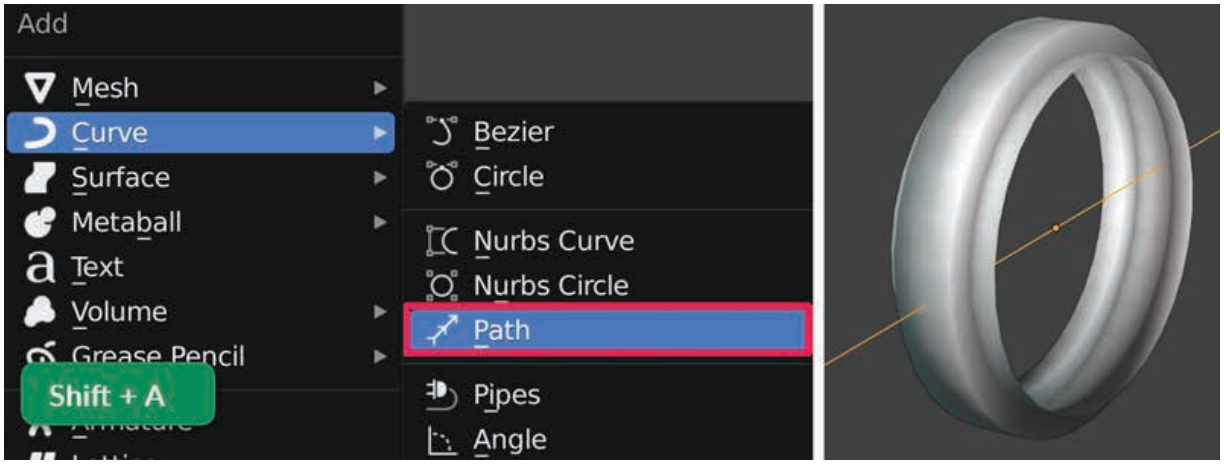
Press Alt+E and choose **Extrude Faces Along Normals**.



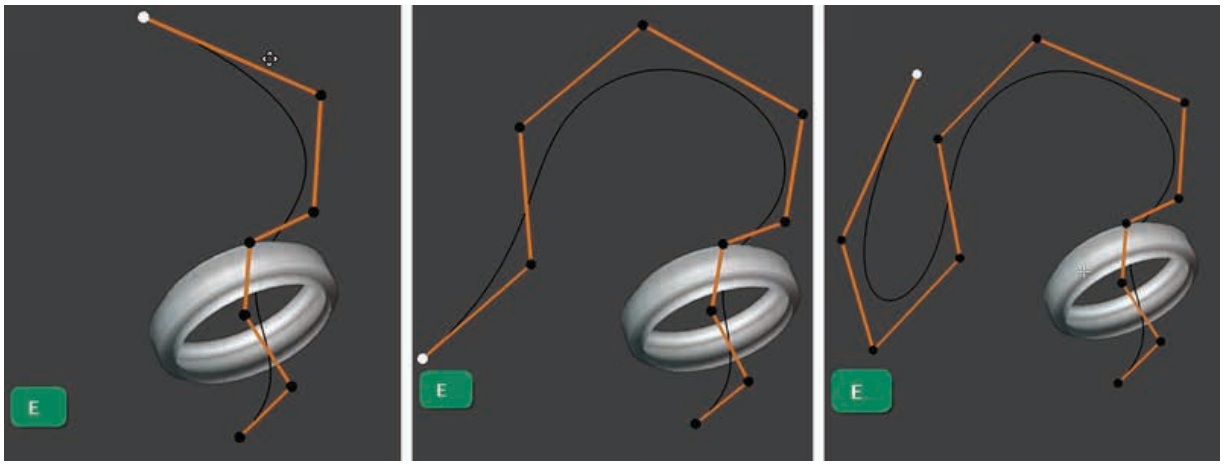
Right-click and set it to **Shade Smooth**. Or choose it from the **Object** menu.



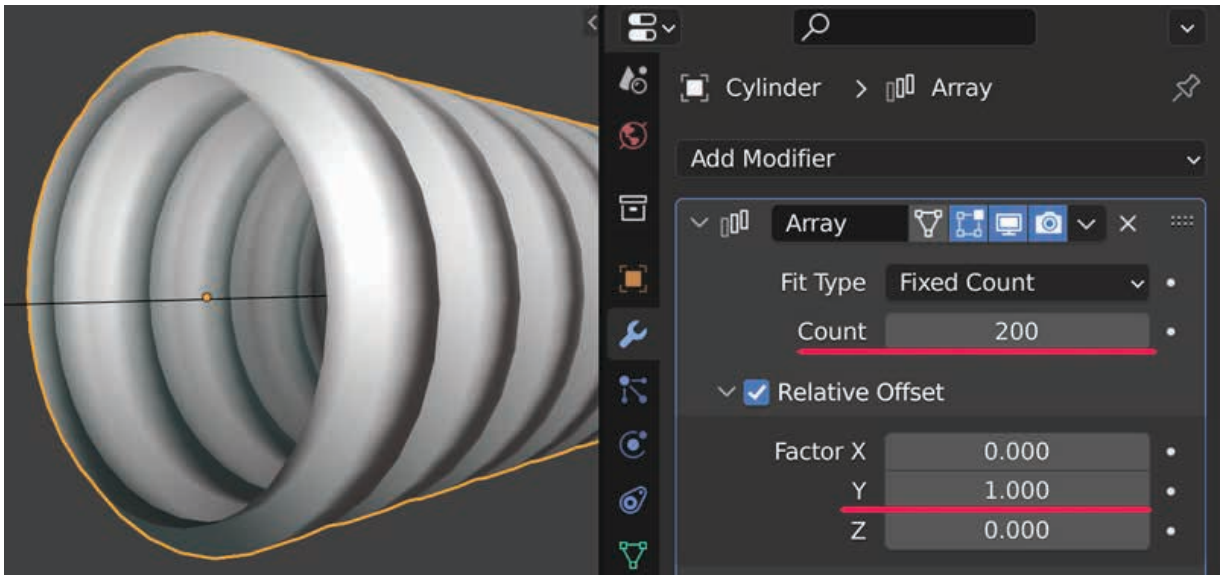
Now add a Curve > **Path**.



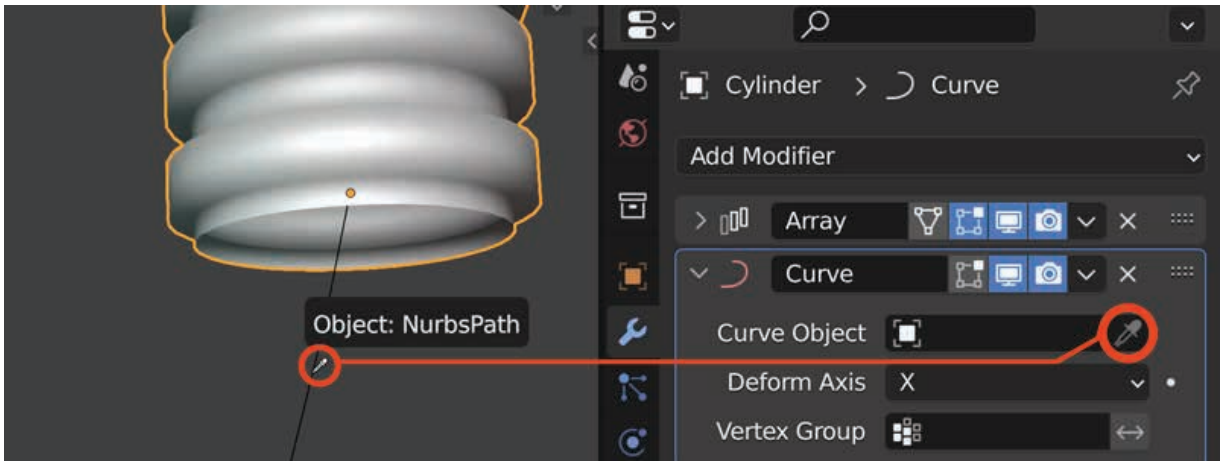
Go into Edit Mode and move the points and extrude more points to make it longer.



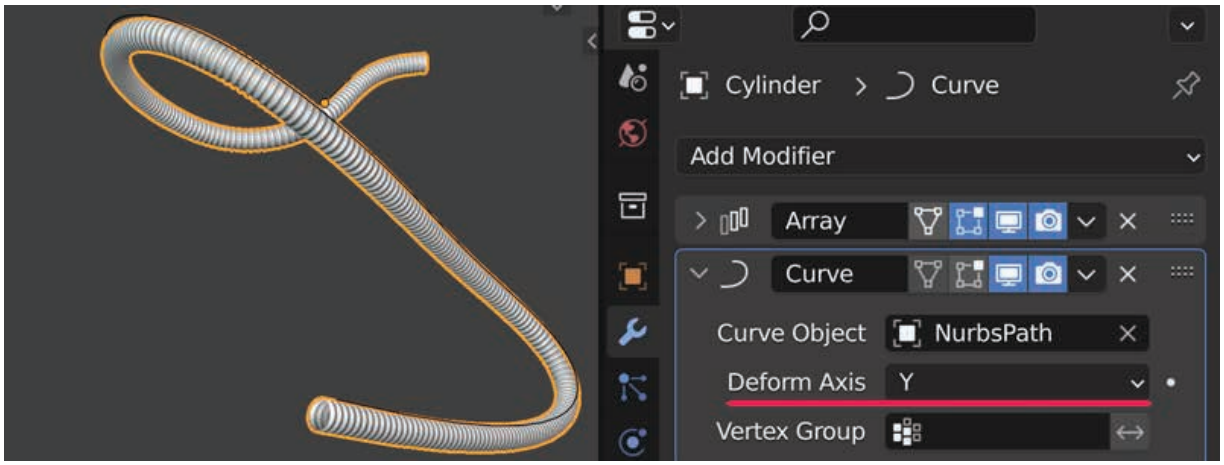
You probably need to scale the cylinder down. Add an Array Modifier to the cylinder Increase the Count.



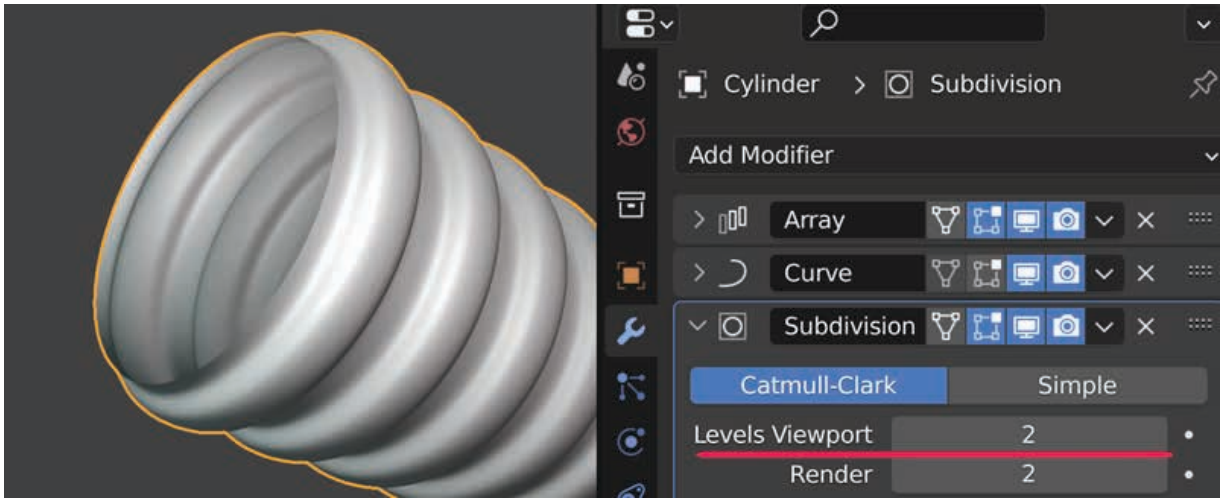
Add a **Curve modifier** to the Cylinder as well and choose the Path as the **Curve Object**.



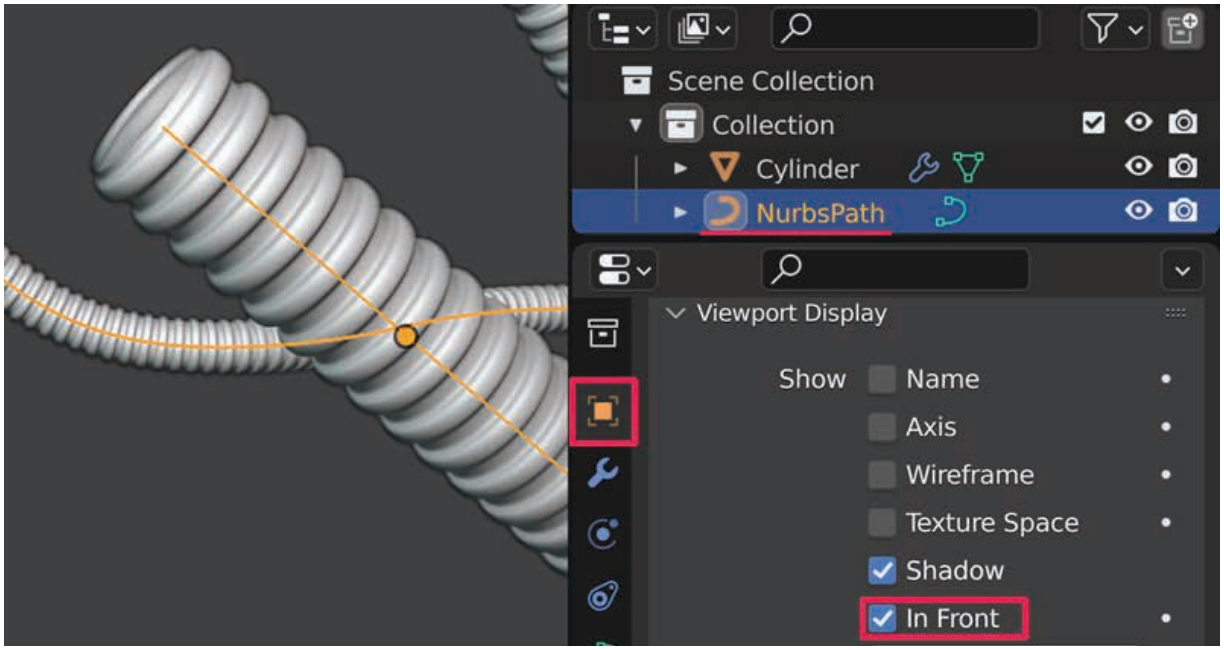
Choose the correct **Deform Axis**. In this case, the Y-axis is the correct one.



You can add a **Subdivision modifier** to make it more rounded.

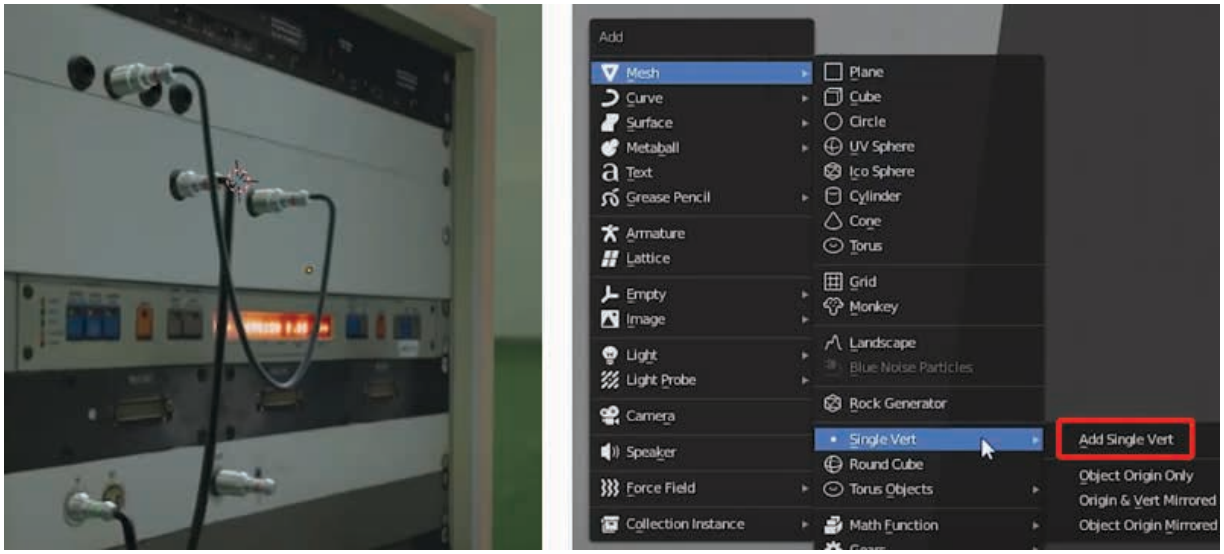


You can set the Curve to **In Front** so it's easier to manipulate the pipe curvature.

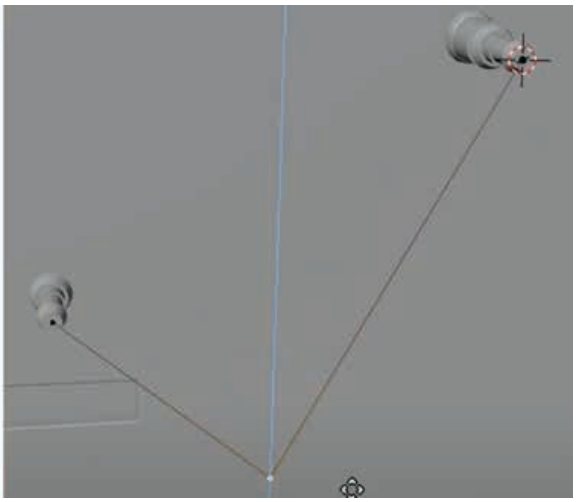


Cable Physics

Create a **single vertex**. You can add this option to the Add menu by installing the **Extra Mesh Objects** add-on in Preferences (in the **Get Extensions** tab).



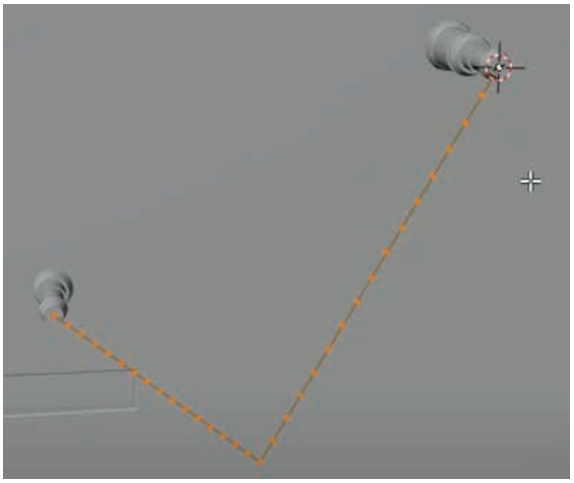
Extrude it by **pressing E** and move the Start and End point to where you want them. Subdivide the vertex by right-clickind and choosing Subdivide, then pull the vertex away to create some more length.



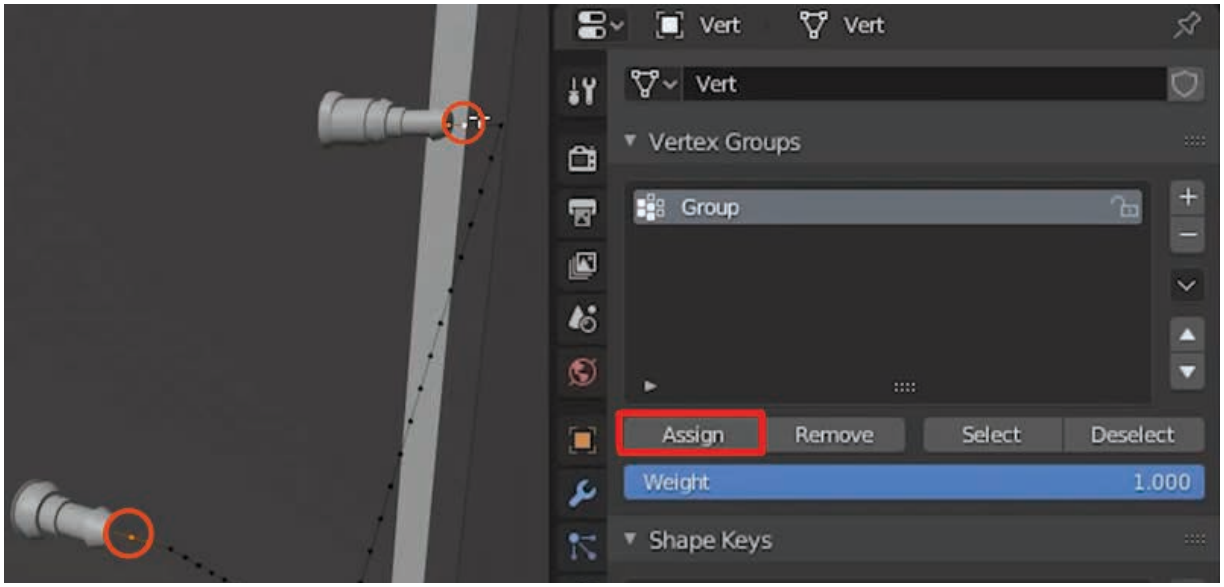
Then subdivide some more.

Instead of right-clicking and choosing “Subdivice” several times, you can do that once and then **press Shift+R** a few times.

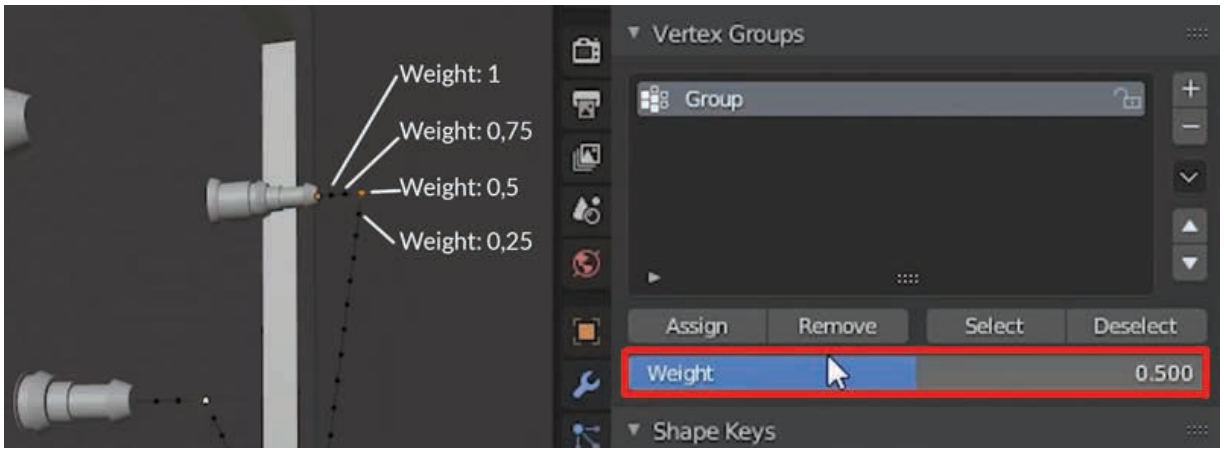
That repeats the last action, in this case subdividing the vertex.



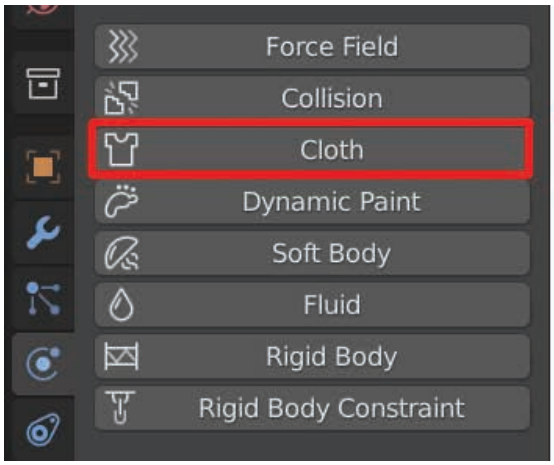
Now add at least the start and end vertex to a **Vertex Group** so we can "pin" them later. Otherwise, everything will just fall down as soon as you start to play the simulation.

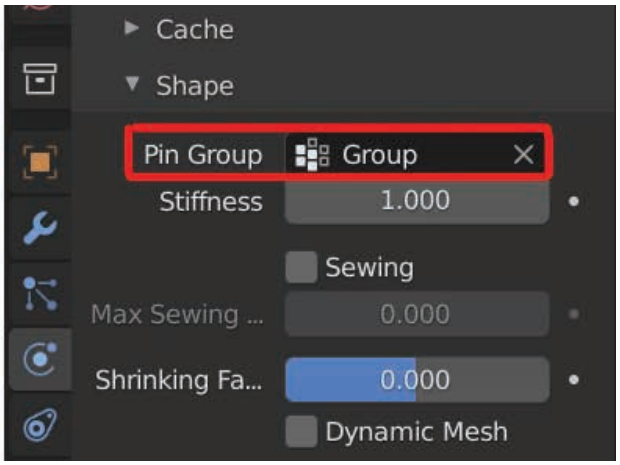


You can also add more vertices to this group with decreasing levels of Weight assigned to them. So they gradually droop down. For example, the first vertices have the Weight of 1, then the ones after that 0,75, then 0,50, then 0,25.

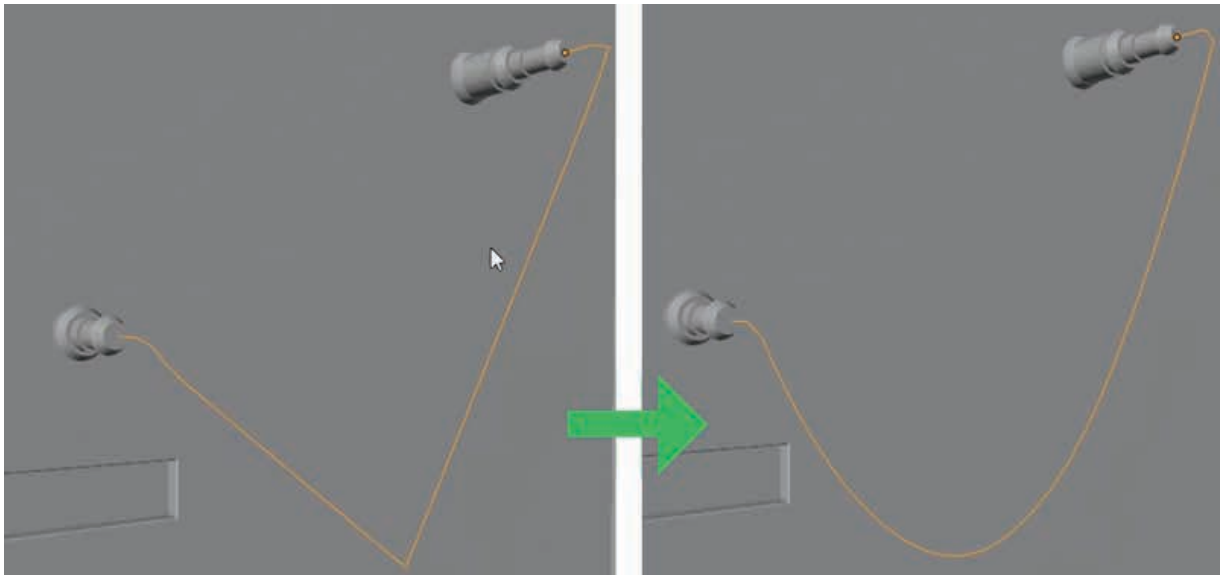


In Object Mode, go to the Physics panel and click on **Cloth**. Then add the Vertex Group to the **Pin** group under **Shape**.

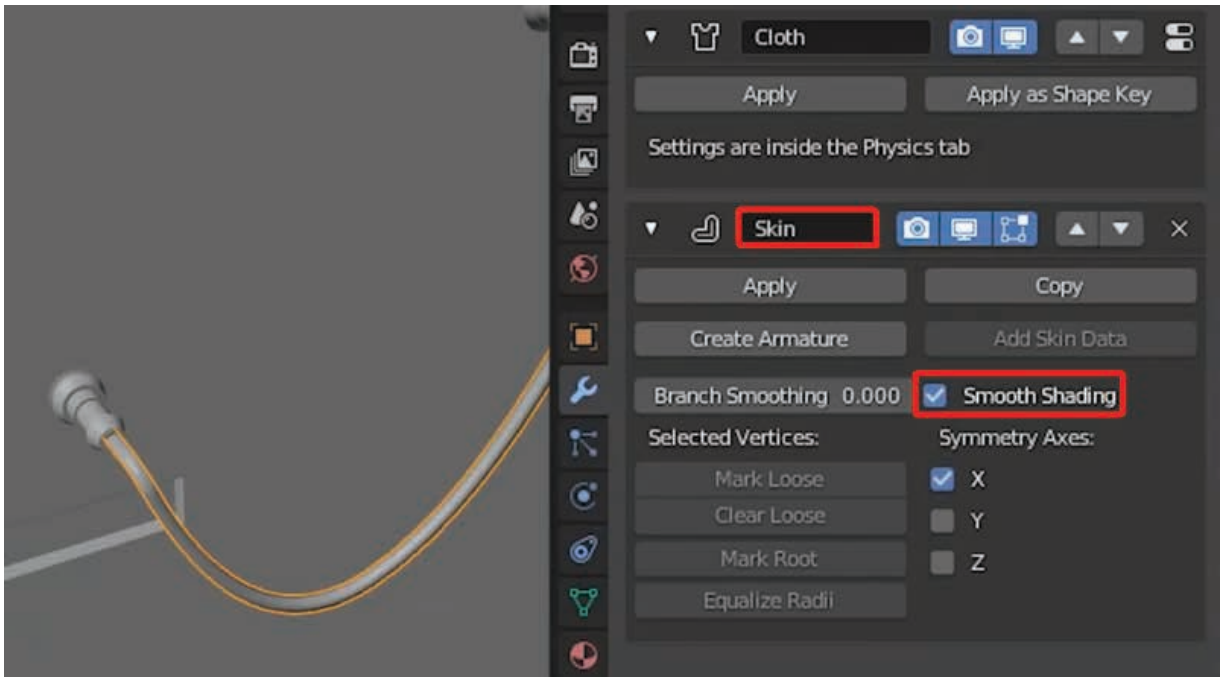




If you **press the Spacebar** in Object Mode, you'll see that the physics work.



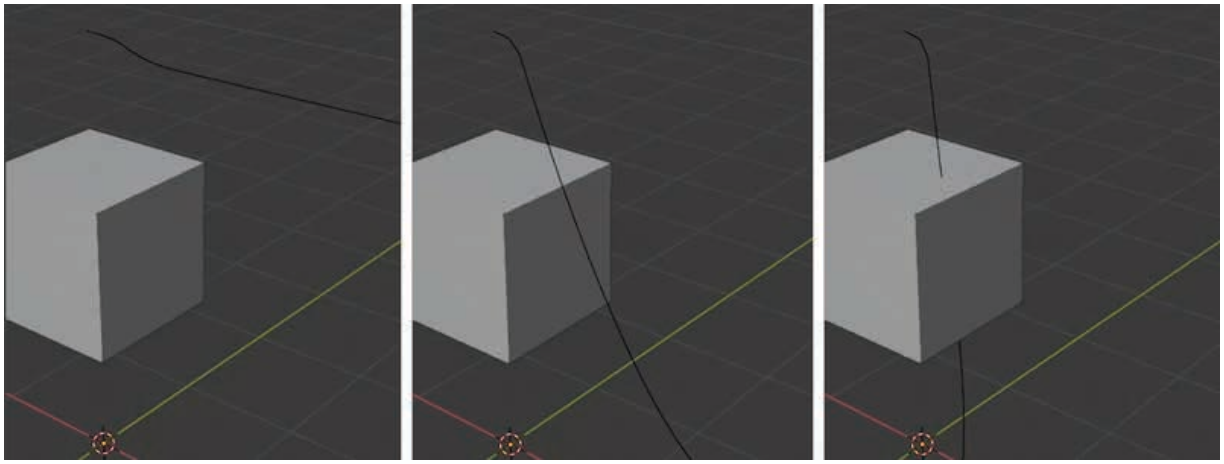
To add thickness to the cable, add a **Skin modifier**. In Edit Mode select all the vertices by **pressing A**, then scale them down with **Ctrl+A** and moving the mouse. Turn on **Smooth Shading**.



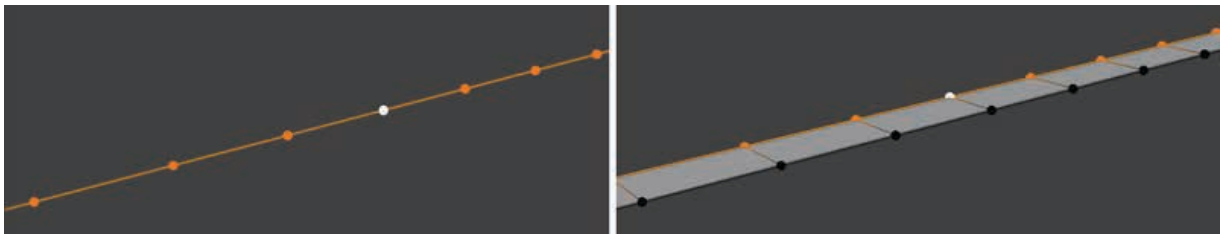
That's all there is to it! If you want to download this file to experiment with it, it's here.

Cables with collision

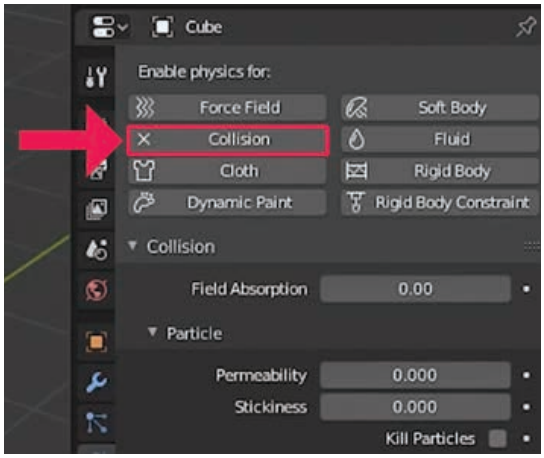
Before we looked at using a Cloth simulation to create a cable. How do you make sure the cable doesn't clip through objects?



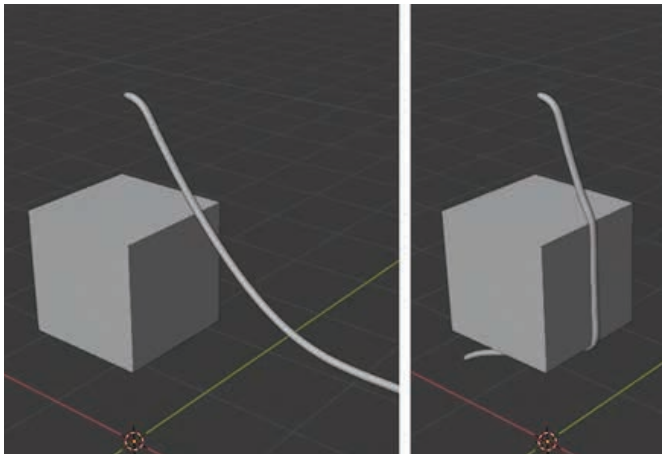
The reason for the clipping is that the cable doesn't have any faces. So we simply need to extrude it to create some faces for the collision.



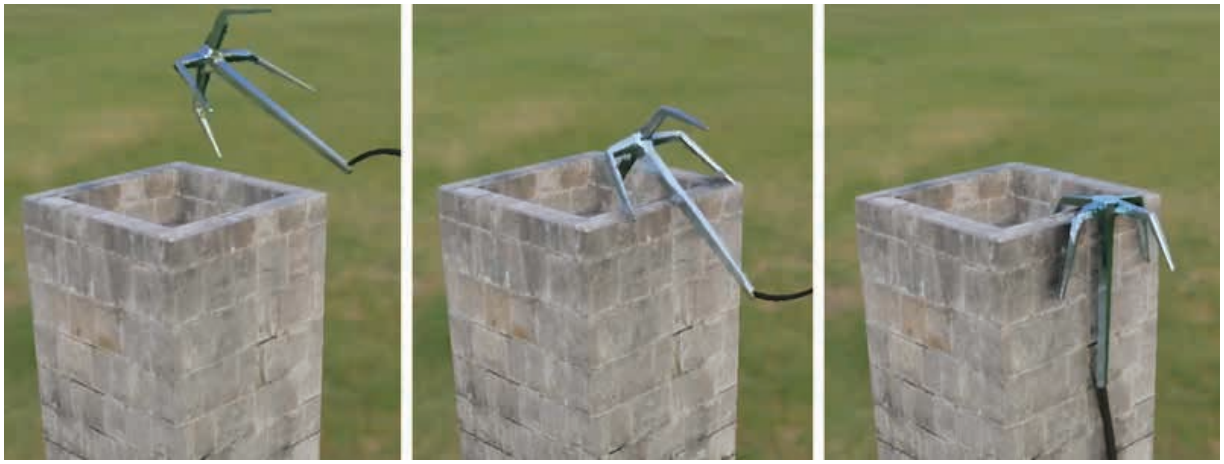
Add **Collision** physics to the Cube and now the Cloth sim interacts with it. That's all you really need to do!



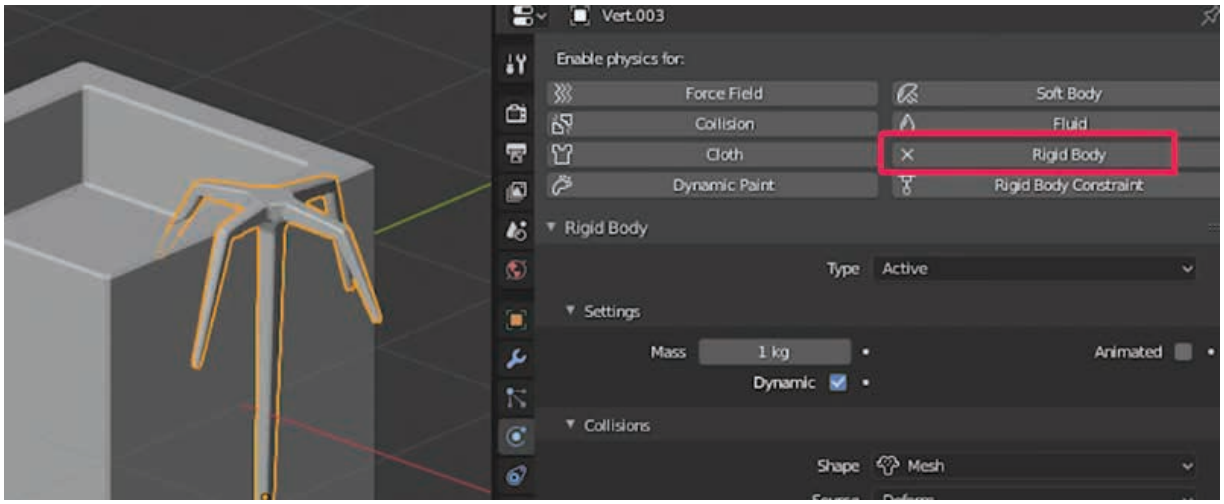
If you want you can add a **Solidify modifier** and a **Subdiv modifier** to the cable to give it some more volume.



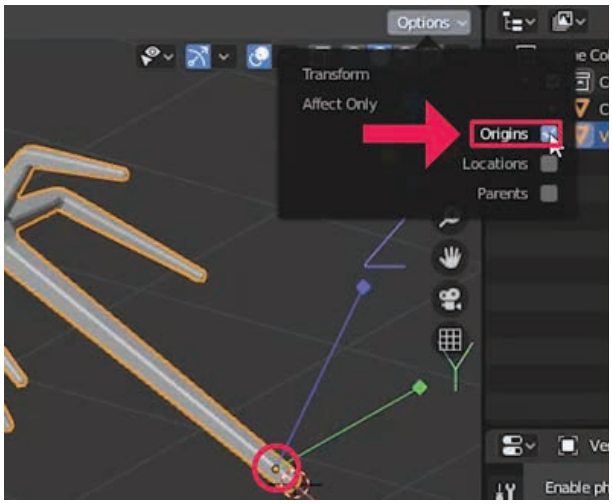
Grappling hook and rope physics

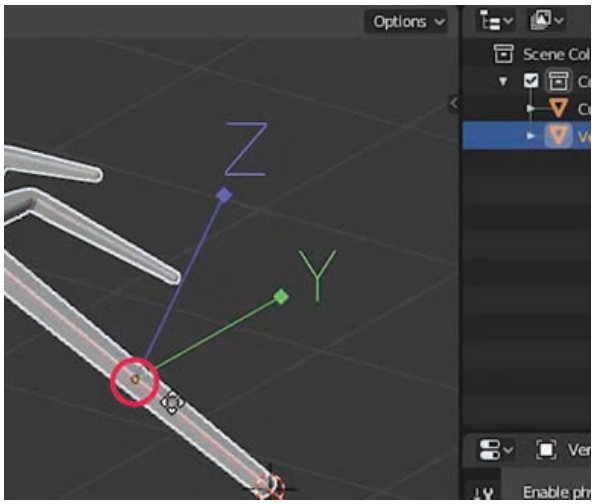


Model a grappling hook and make it an **Active Rigid Body**. Instead of making the model yourself, you can also search online. For example, on Sketchfab you can find this free one.

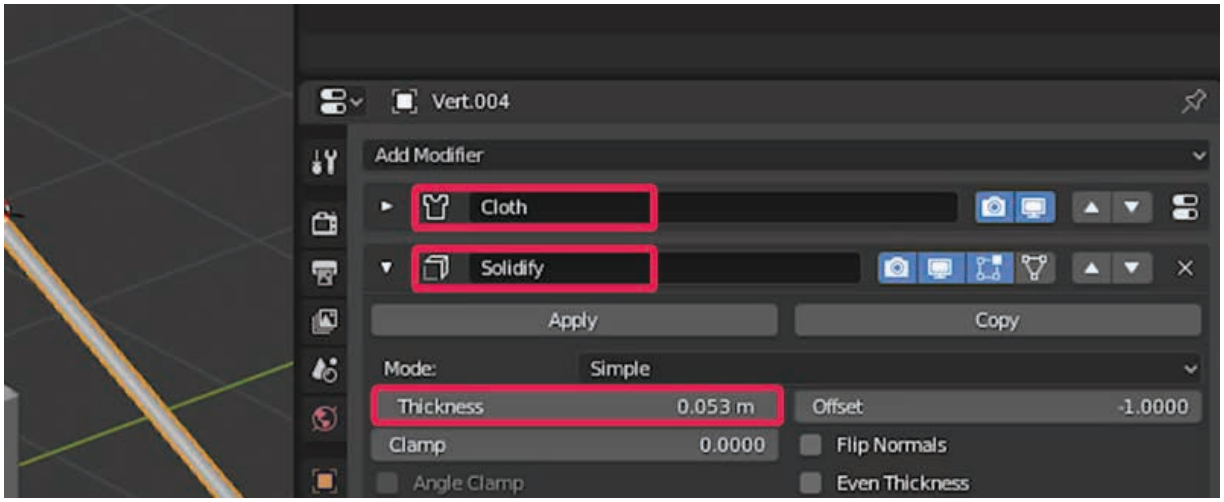


You can control where the center of weight by where you place the origin.

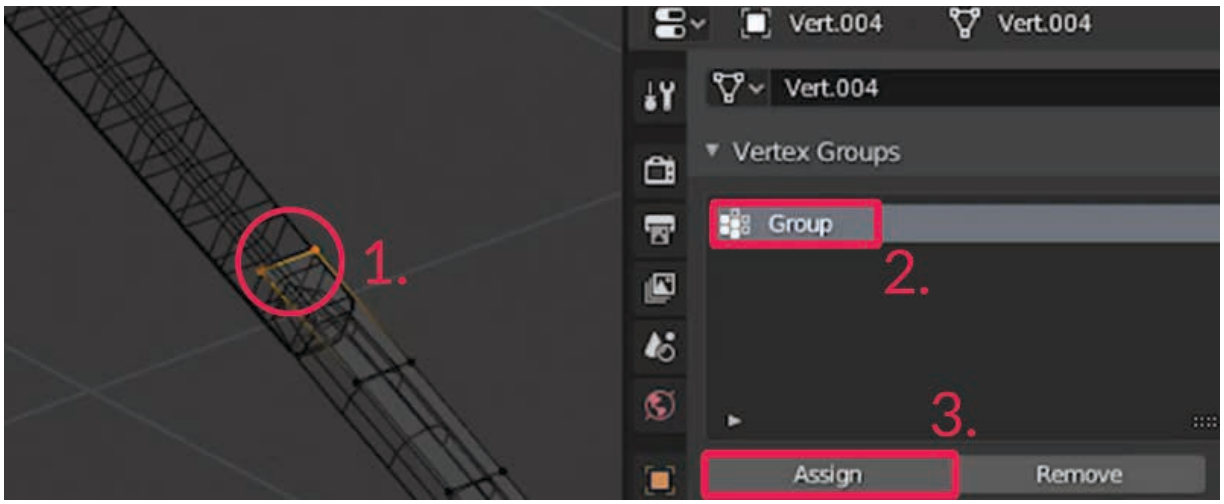




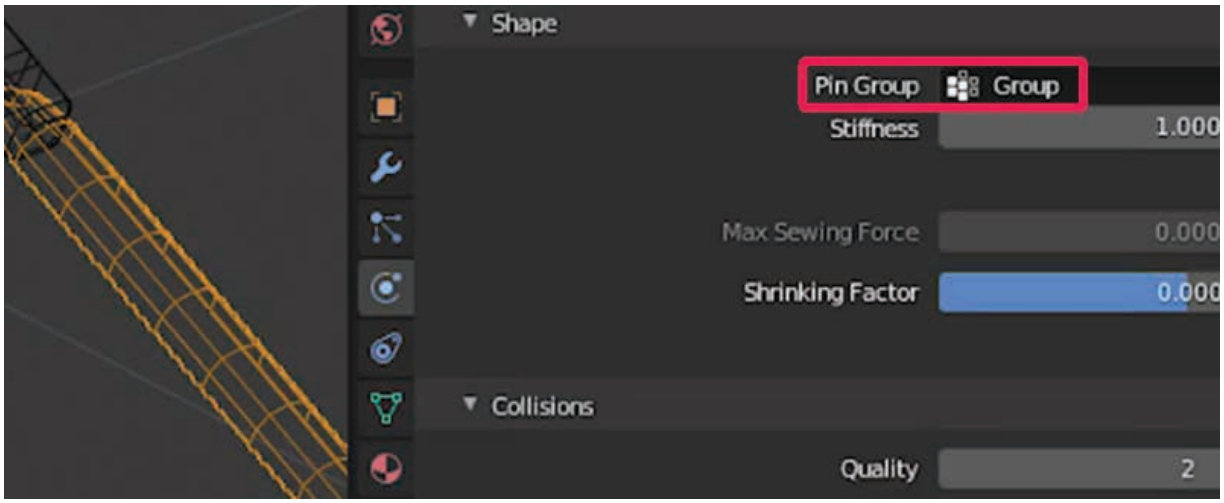
Extrude a subdivided edge and give it **Cloth** and **Solidify modifiers** to make a rope.



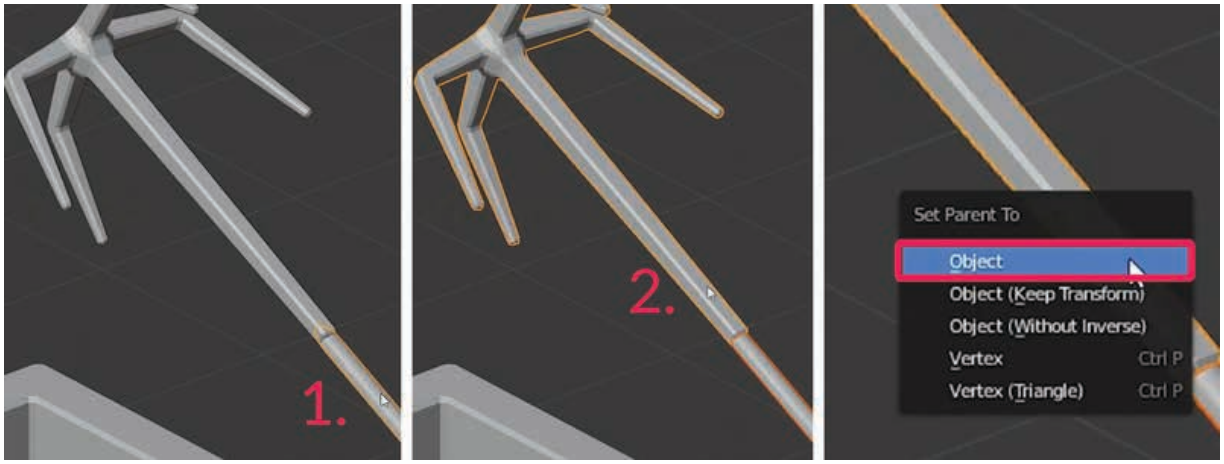
Add the vertices close to the grappling hook to a **Vertex Group**.



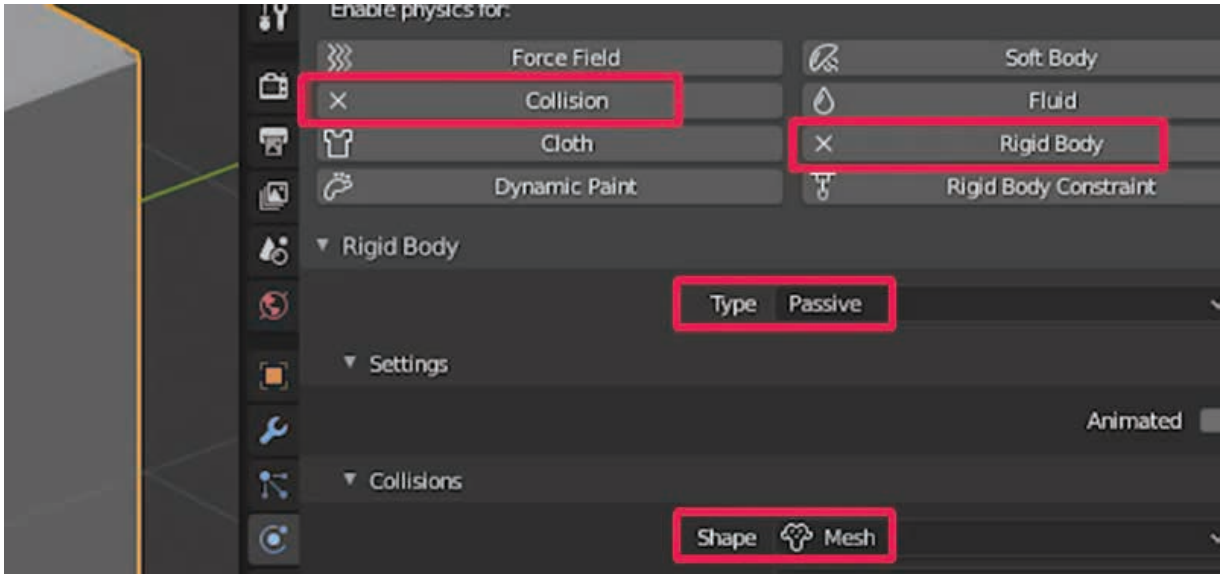
Add the Vertex Group to the Cloth modifier's **Pin Group**.



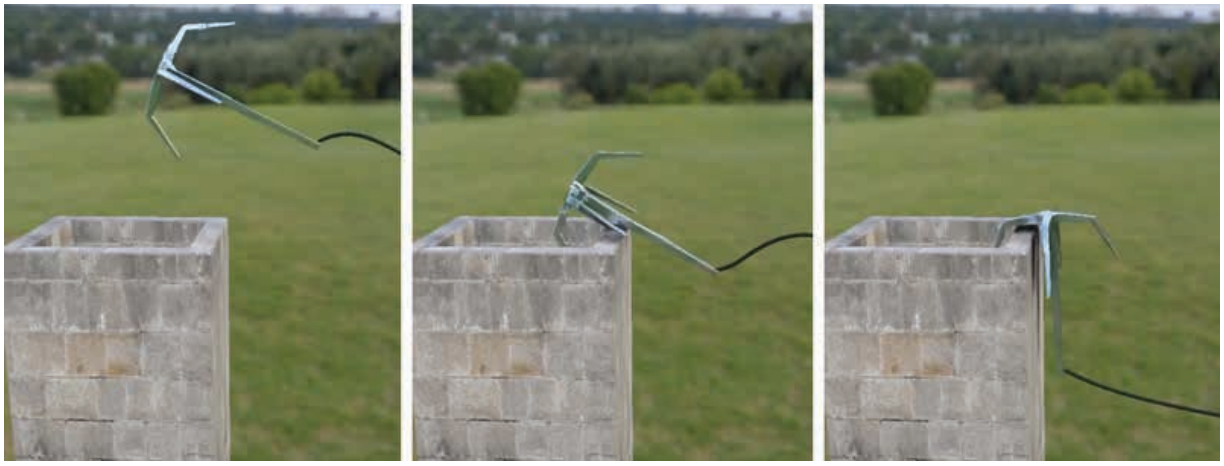
Now parent the rope to the grappling hook with **Ctrl+P**.



Make the cube a **Passive Rigid body** and activate **Collision**.

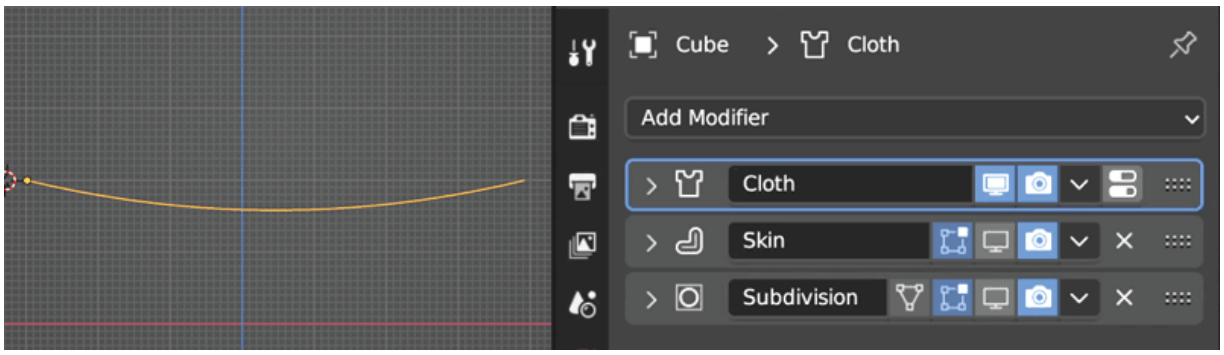


Press the Spacebar to play the simulation.

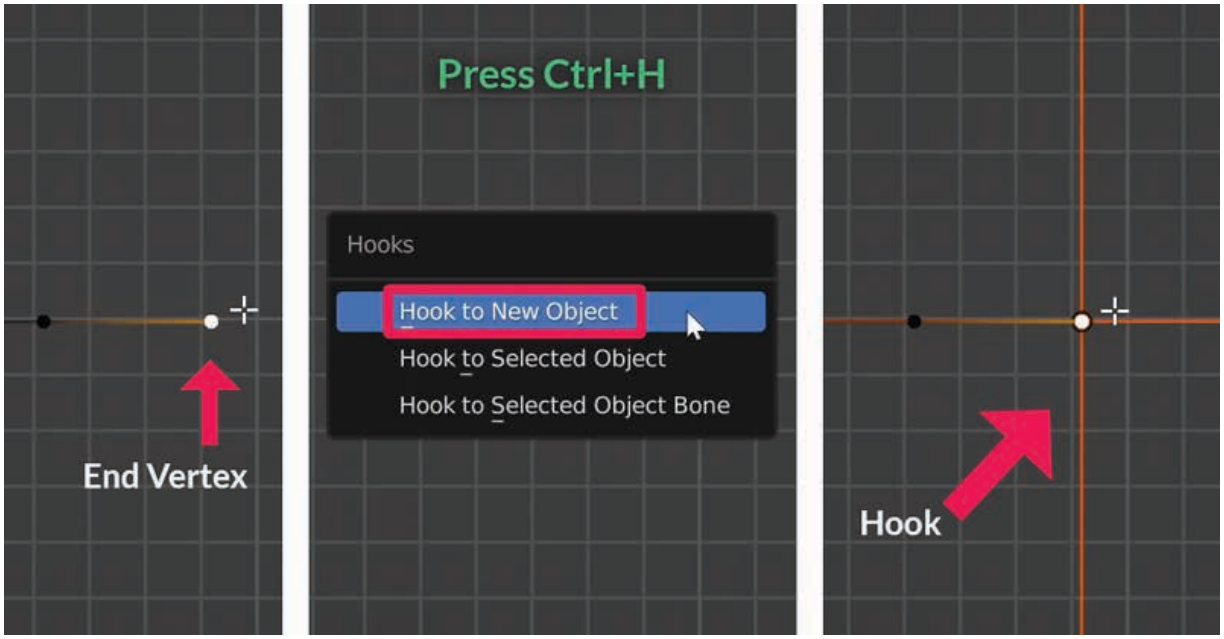


Adding Cables with Physics that are attached to moving objects

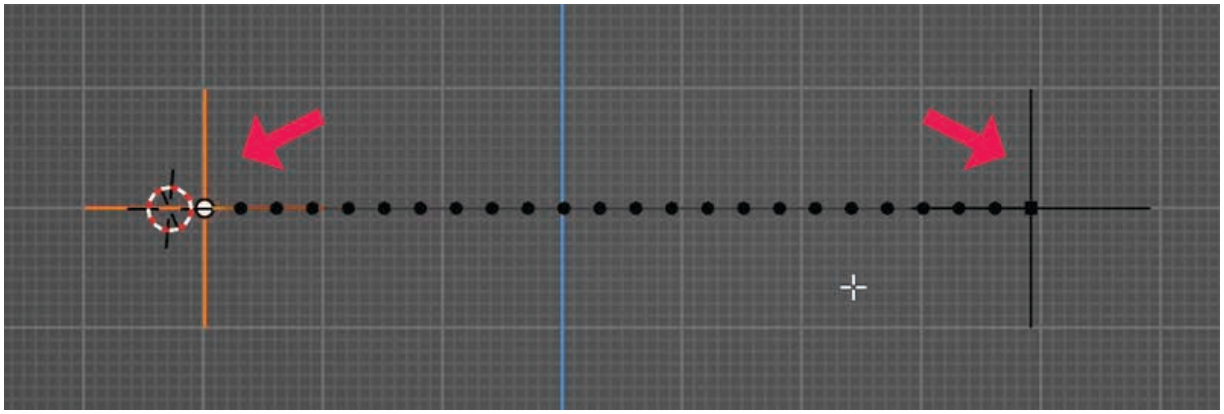
In the previous topics, we learned how to create cables with physics, by using a Cloth sim and a Skin modifier on a subdivided edge that was attached to a static place using a Pinning Group. Now, we'll take this exact method a bit further and add the cable to moving objects.



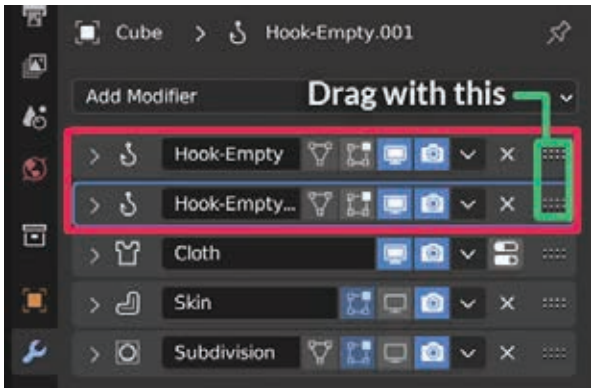
After following the exact same steps as before, including pinning the vertices at either end, we select the vertex on one end of the edge and **press Ctrl+H**, then choose **Hook to New Object**. This "parents" that vertex to a Hook, which is just like an Empty: an object that doesn't get rendered.



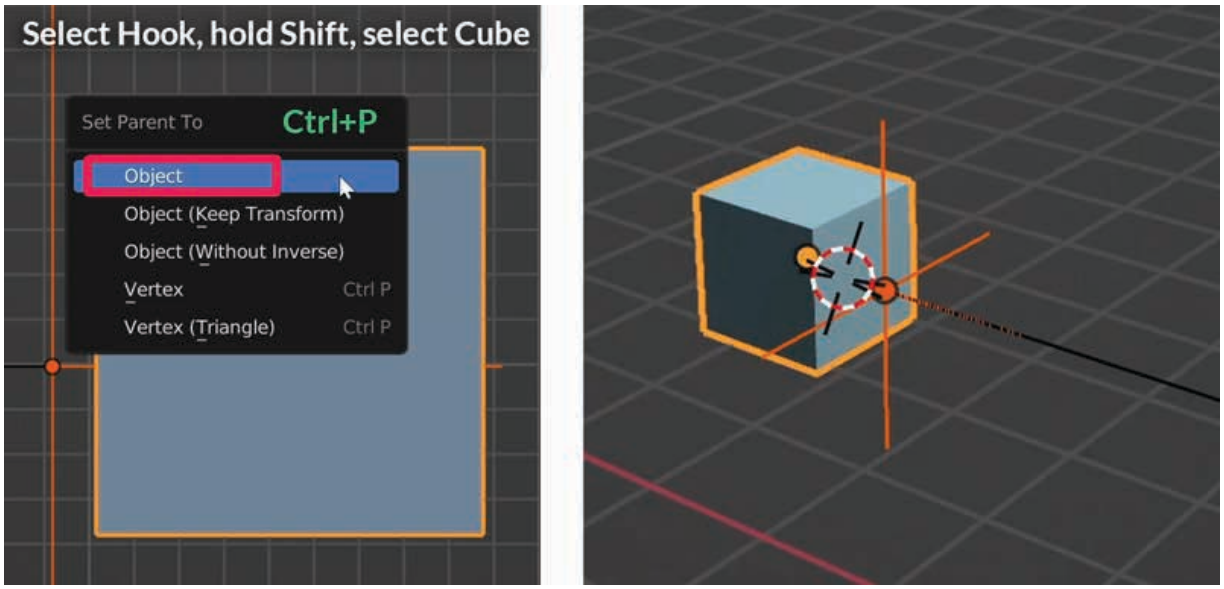
Do the same for the vertex on the other end.



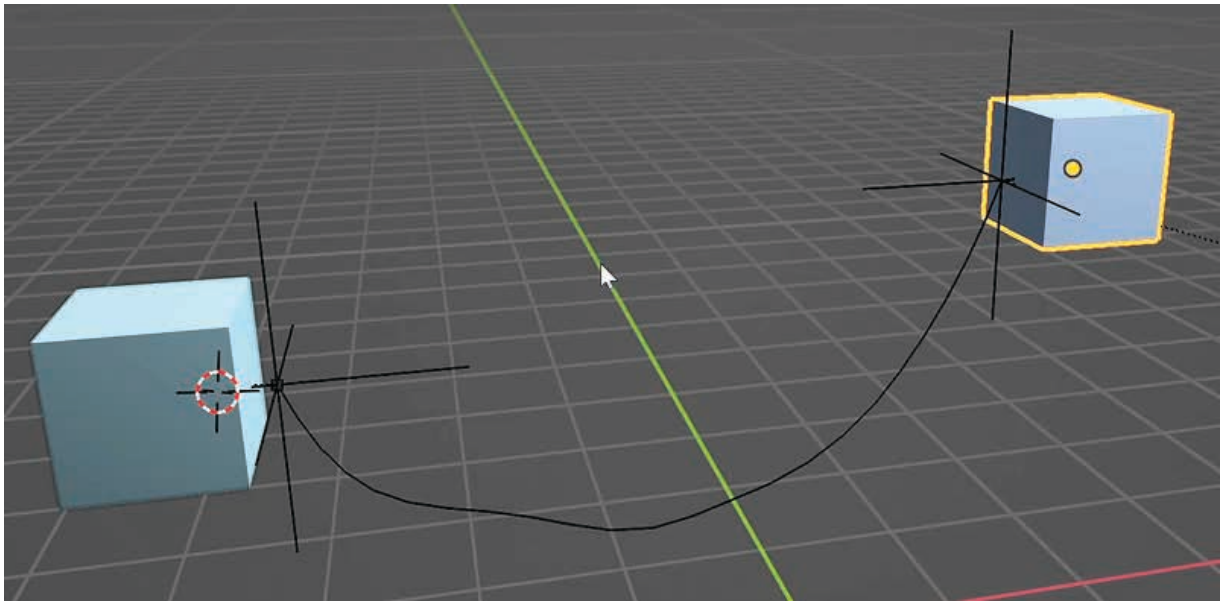
And make sure both Hook modifiers are at the top of the modifier stack of the subdivided edge.



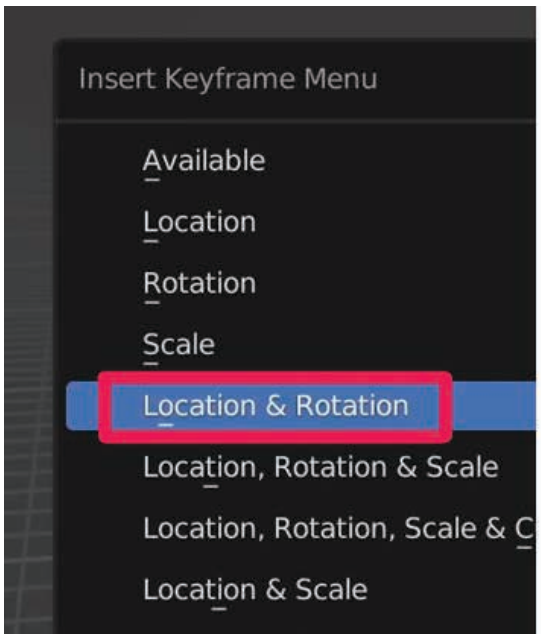
The hooks can then in turn be parented to anything you want with **Ctrl+P > Object**. In this example I'm parenting each Hook to a Cube.

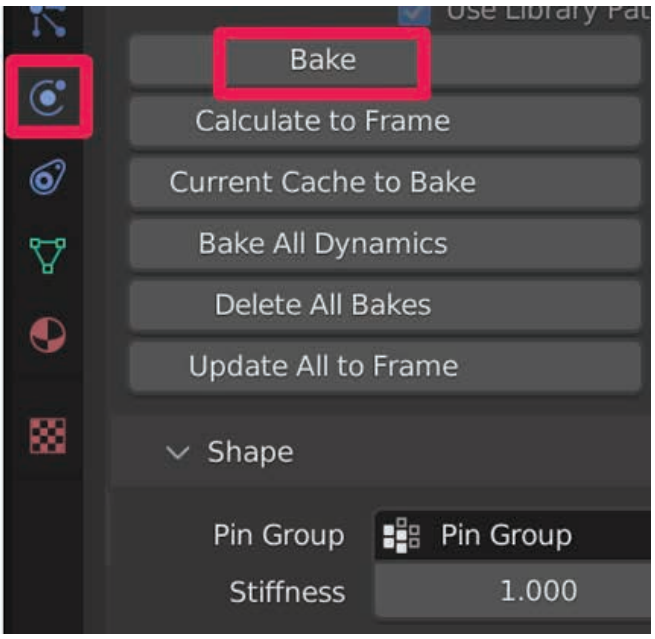


If you play the timeline now by **pressing the Spacebar**, and move the objects in real-time by grabbing them and moving them, it may look like it's not working. Don't despair: you've probably done everything right. But this kind of thing just works better if you "bake" it before playing the animation.



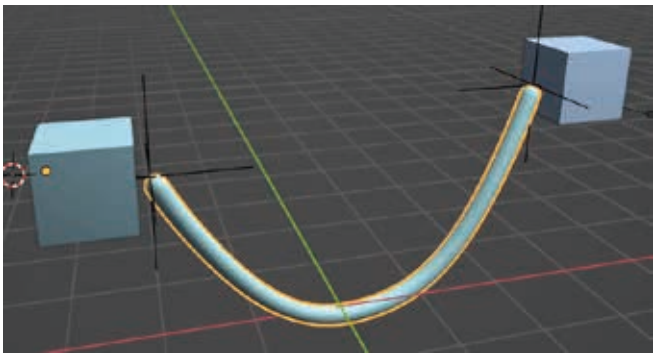
If you keyframe the location of the hooks or the objects they are parented to (by moving them to some location and then **pressing i > Location** and repeating those steps at some different point at the Timeline), and then Bake the simulation, it works just fine.



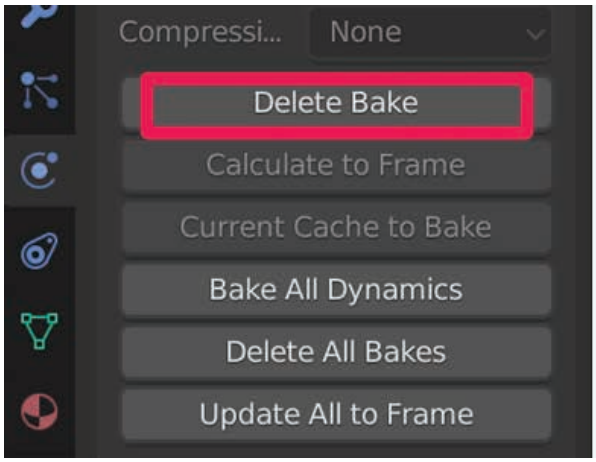


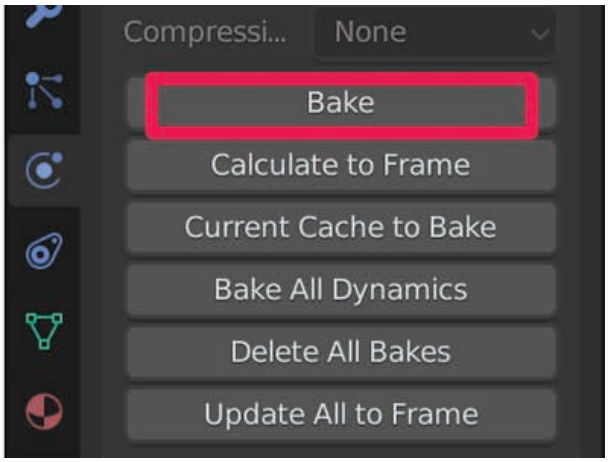
The result, with the Skin modifier and Subdivision modifier turned on.

Having baked the simulation, the cable follows the hooks/cubes perfectly.



Do click **Delete Bake** and **Bake** again every time you've changed something.

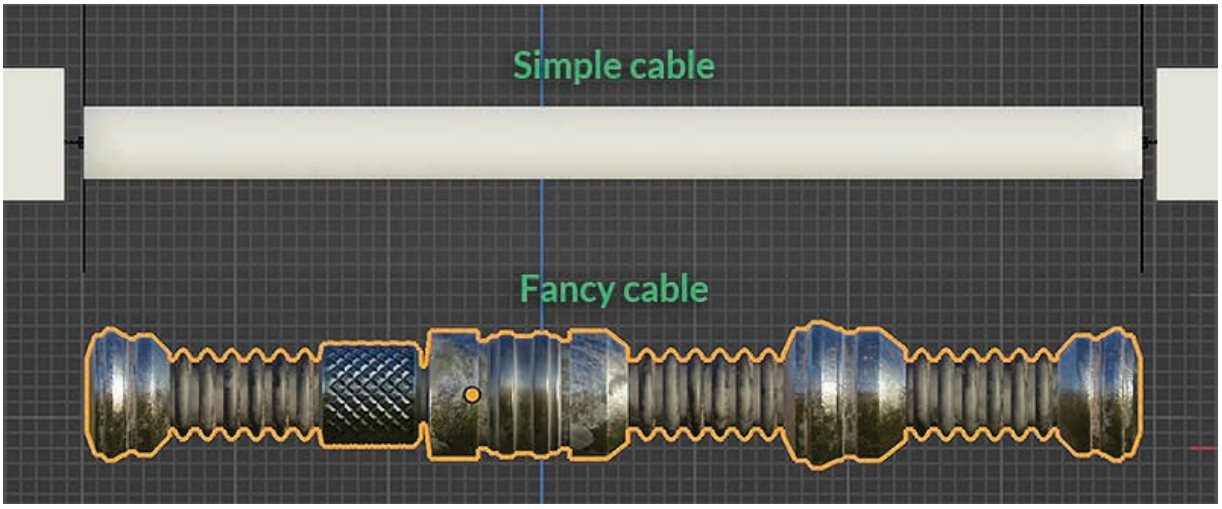




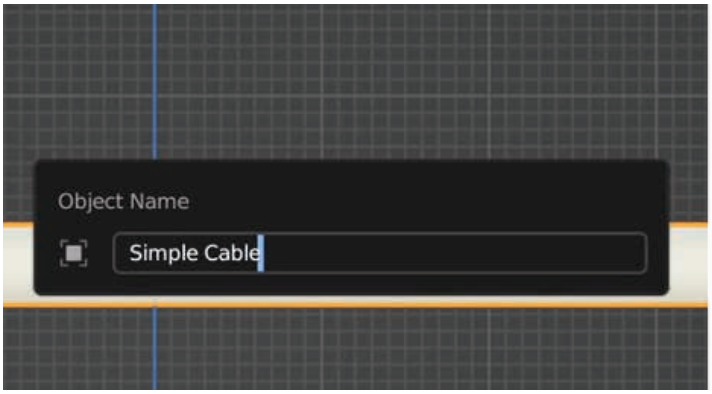
Textured Cable with Physics connected to objects

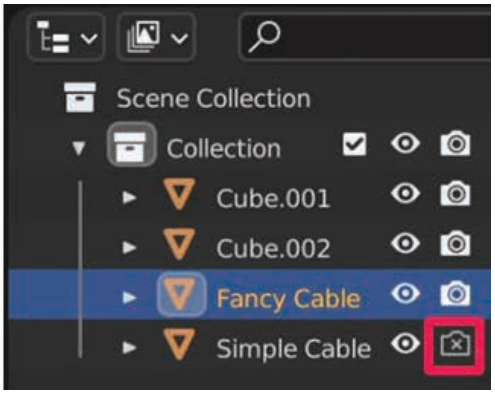
Now that we know everything about creating cables with physics and collision detection so they don't intersect with other geometry, let's make them look a bit more interesting.

Let's bring in a more fancy cable and animate it instead. In this case I've appended a model I've made in a separate Blend file.

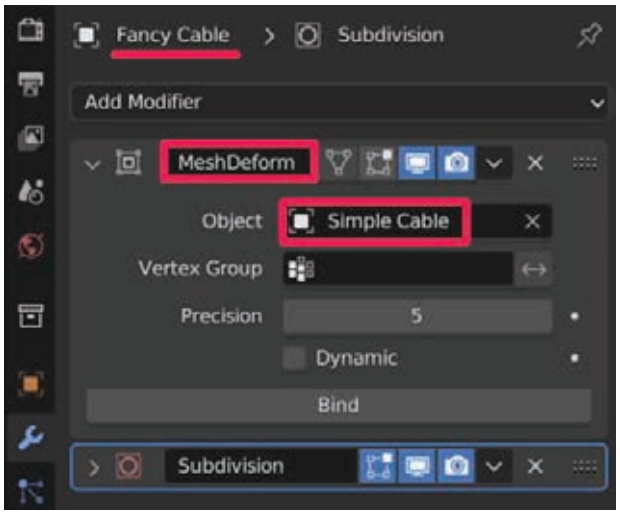


Name the original cable "Simple Cable" and the more detailed one "Fancy cable" or whatever is clear to you. You can rename an object by selecting it and **pressing F2** and typing the name or by double-clicking on a name in the Outliner and typing a new name. **Turn off the Render visibility** of the Simple Cable in the **Outliner**.





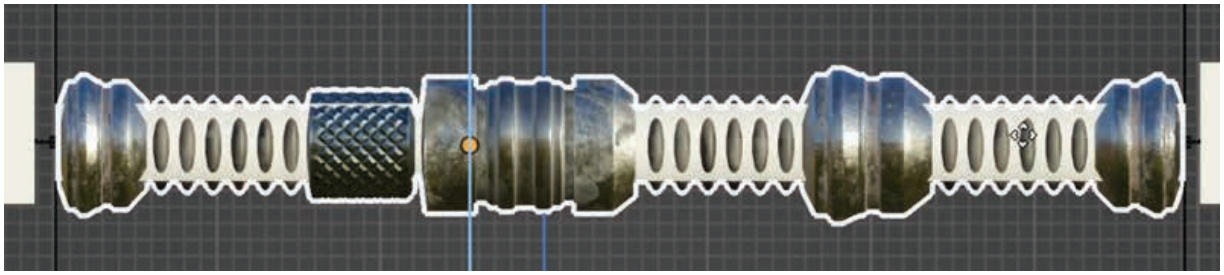
Add a **Mesh Deform modifier** to the fancy cable and choose the Simple Cable as the **Object**.



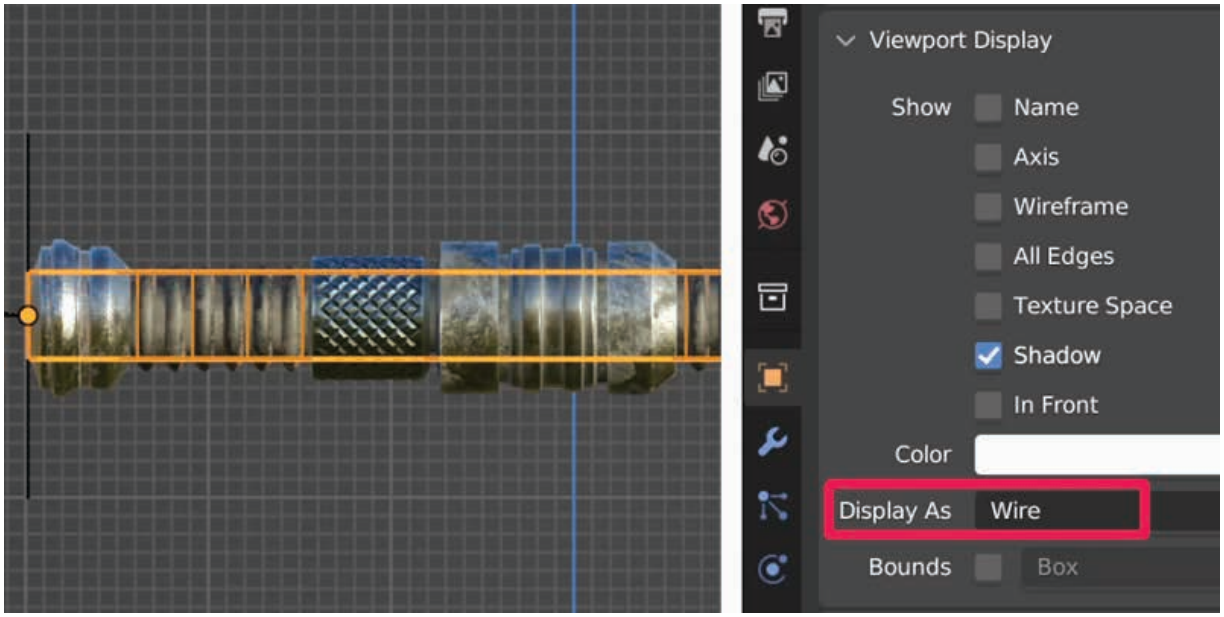
Make sure any **Subdivision modifier** on the fancy cable is underneath the Mesh Deform modifier.

Otherwise it will not work in the render (as the model will have a different amount of vertices when subdivided, and that will confuse the MeshDeform modifier).

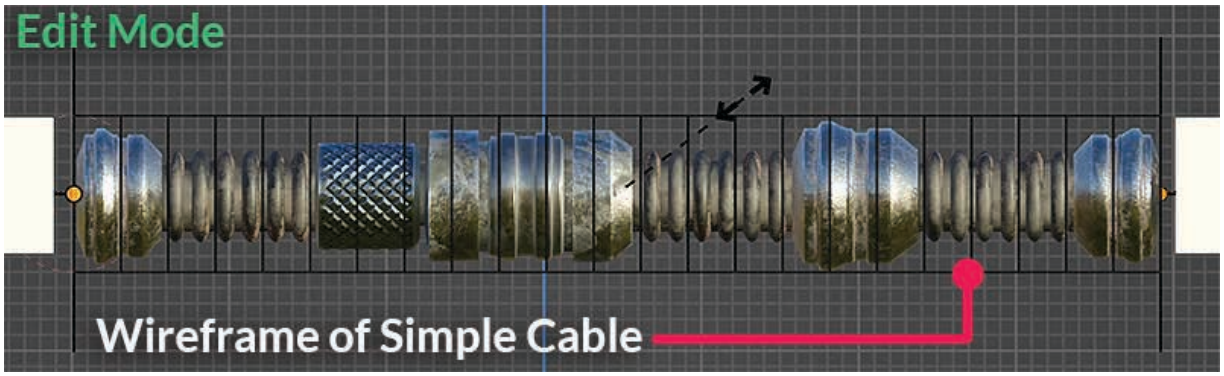
Place it in the same location as the simple cable.



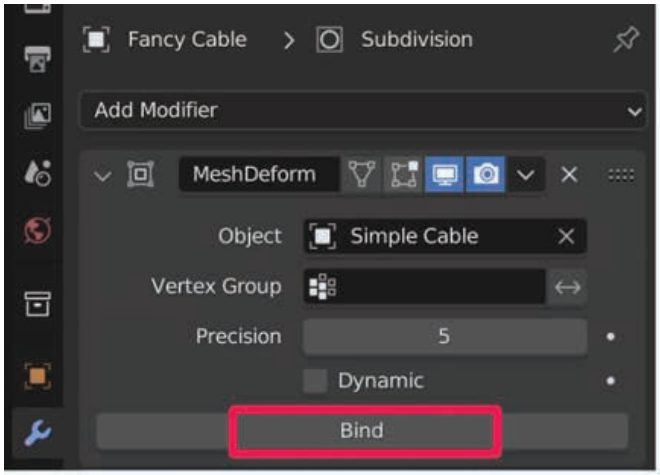
Set the **Visibility** of the Simple cable to **Wire** so you can see the fancy cable inside.



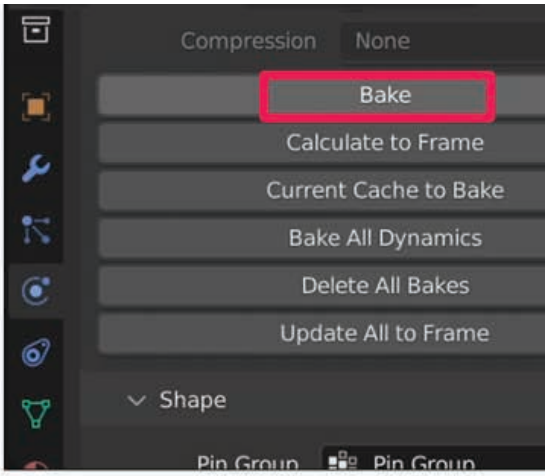
To scale up the Skin of the Simple Cable, in Edit Mode **press A** to select all vertices, then **Ctrl+A** and drag the mouse to scale them up. The Fancy Cable needs to be completely enveloped by the Simple Cable, otherwise it won't work.



Click on Bind and wait for it to calculate. Then **Bake** the simulation.

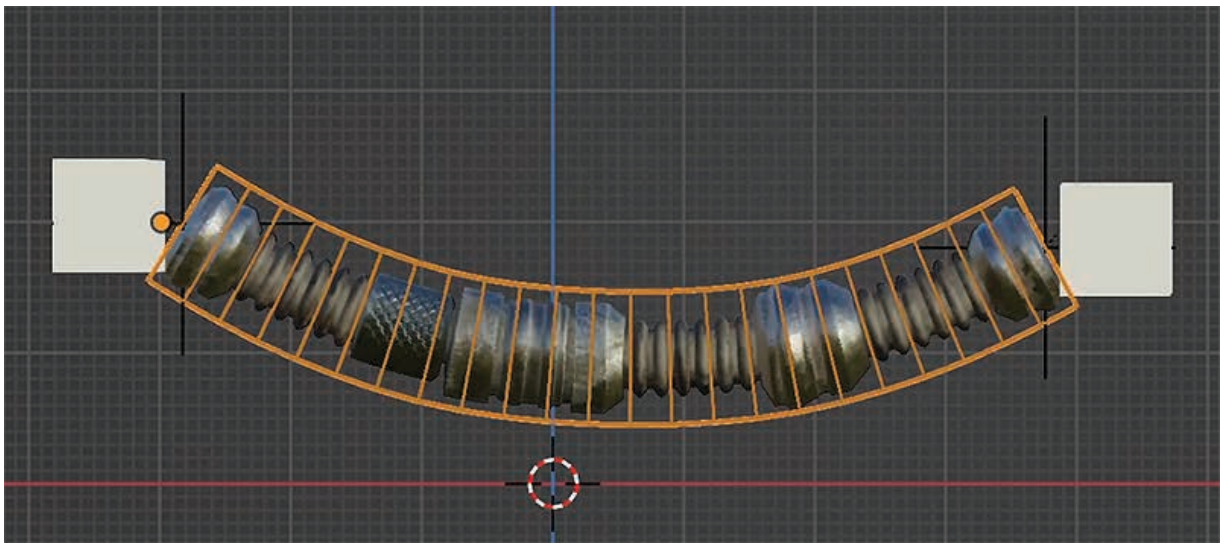


Fancy cable



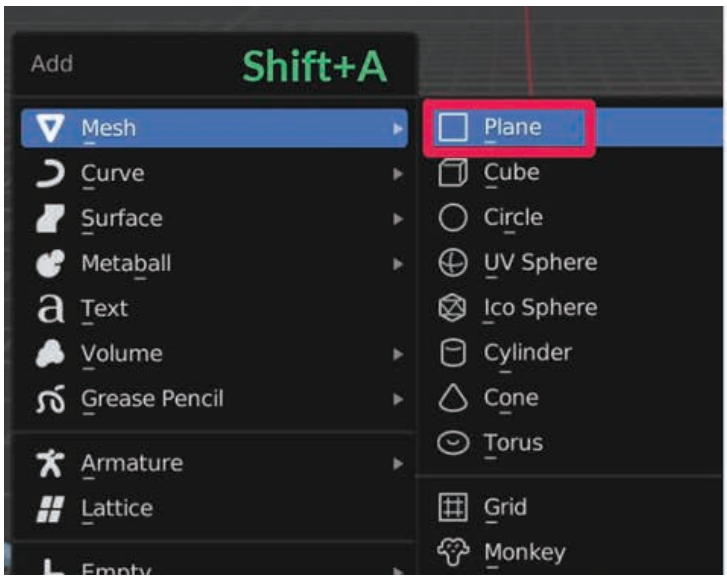
Simple cable

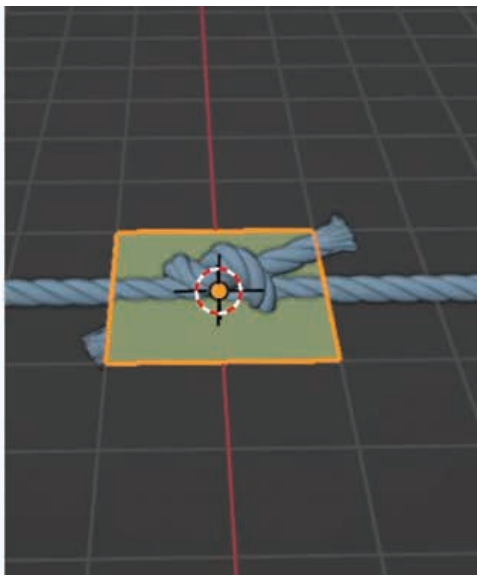
Now when you **press the Spacebar** the Fancy Cable will be deformed by the Simple Cable. The Simple Cable won't show up in the render, since you've disabled its render visibility in the Outliner.



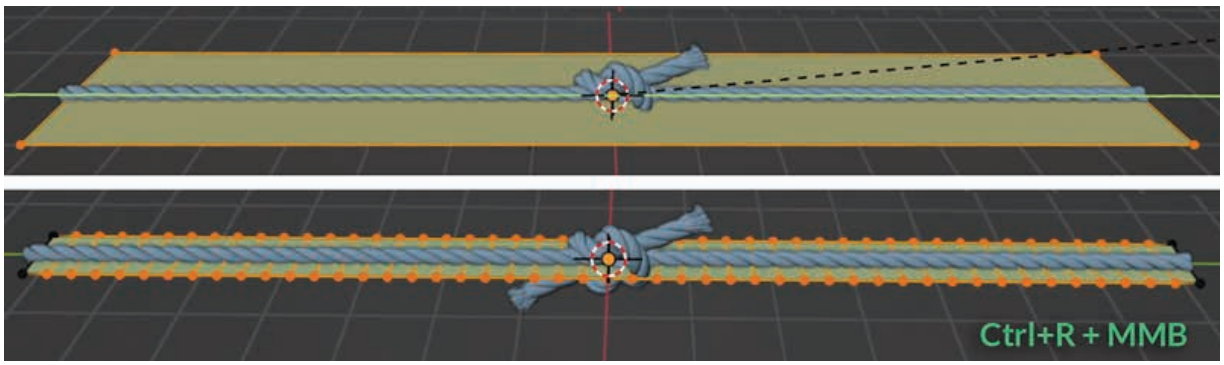
Complex Rope Mesh Physics with the Surface Deform modifier

To enable collisions in a cloth simulation, you need to have some Faces instead of just a subdivided Edge. Add a Plane (**Shift+A > Mesh > Plane**).

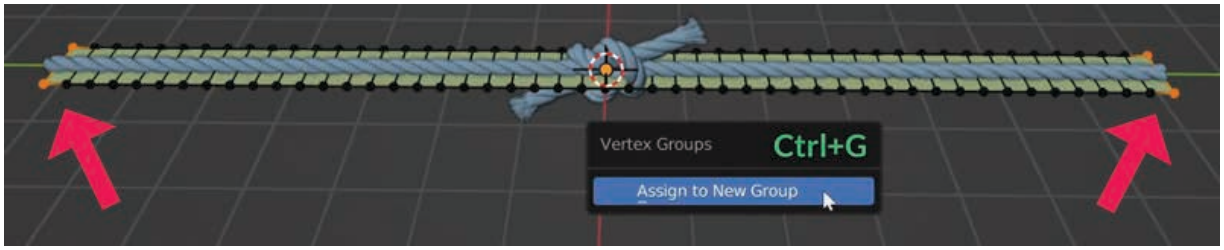




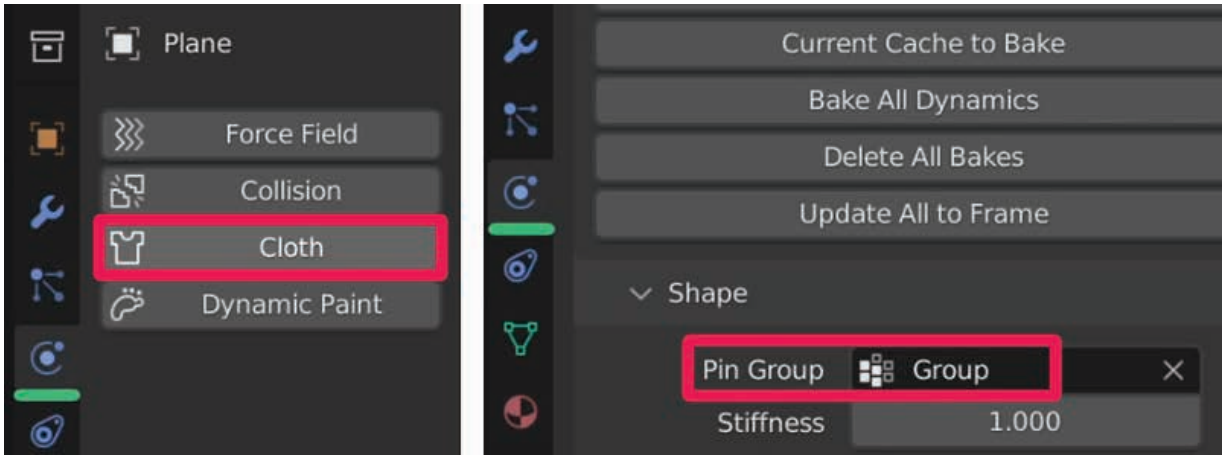
Scale it in Edit Mode to be long and thin and give it some loop cuts with **Ctrl+R** and scrolling the middle mouse wheel.



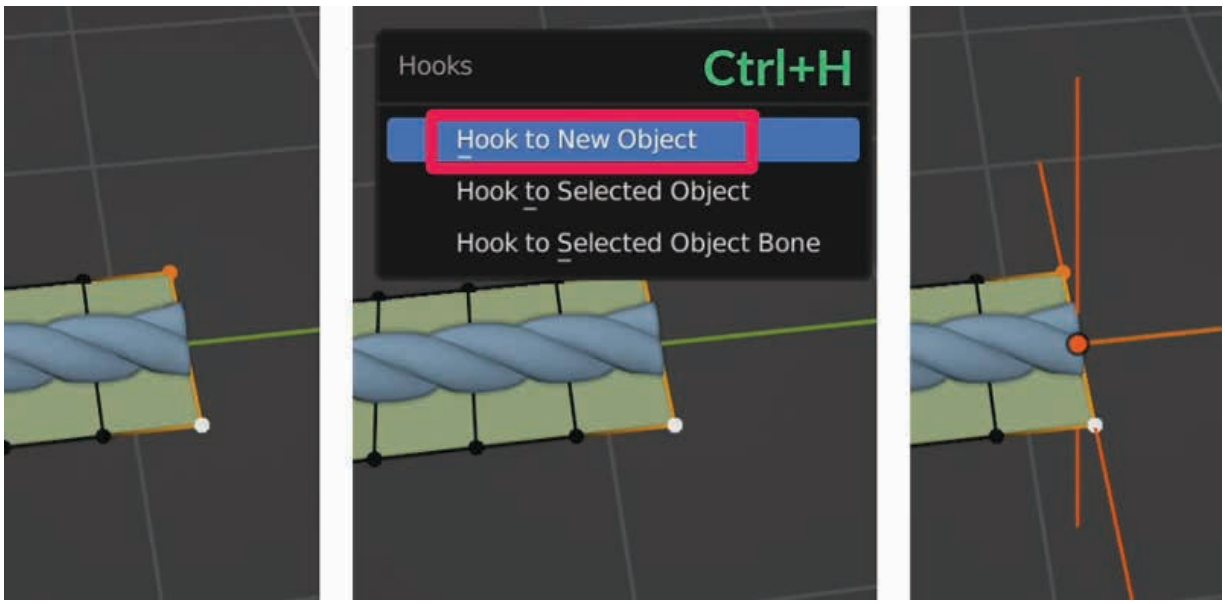
Select some vertices at both ends, **press Ctrl+G** and choose **Assign to New Group**. *Note: the Pin Group can't be larger than the amount of vertices attached to Hooks, otherwise the Hooks won't be able to move the cloth object properly!*



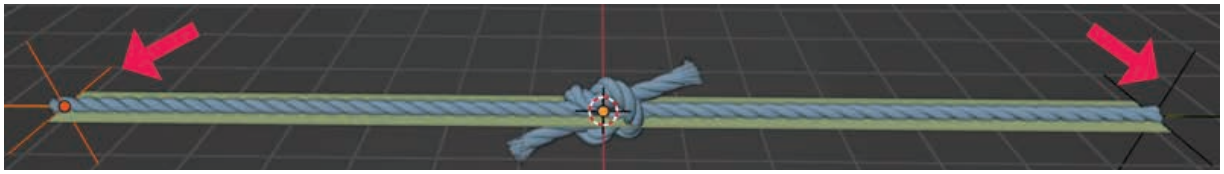
Add a **Cloth** sim and enter the Vertex Group in the **Pin Group** field. This prevents the cable from falling down into infinity.



If you want to be able to animate the position of the ends of the rope, add hooks to both ends by selecting vertices and **pressing Ctrl+H > Hook to New Object**. *Note that this is only necessary if you want to animate the position of the ends of the rope. If you just want it to remain in place, you don't need Hooks - the Pin Group is enough for that.*

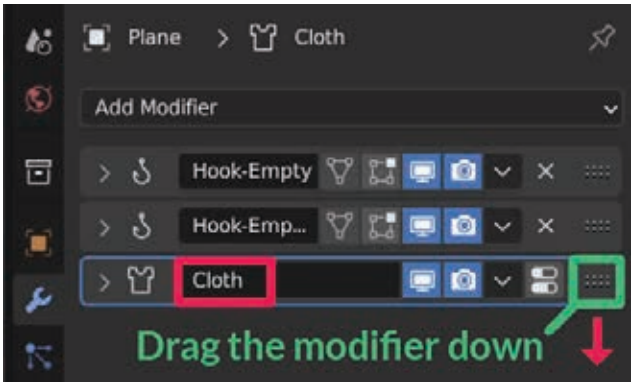


You can have **Hooks** at both ends, or just at one end. In fact, you can have as many or as few Hooks as you want, attached to any vertices, not just the ones at the end of the object. It just depends how you want to animate the rope.

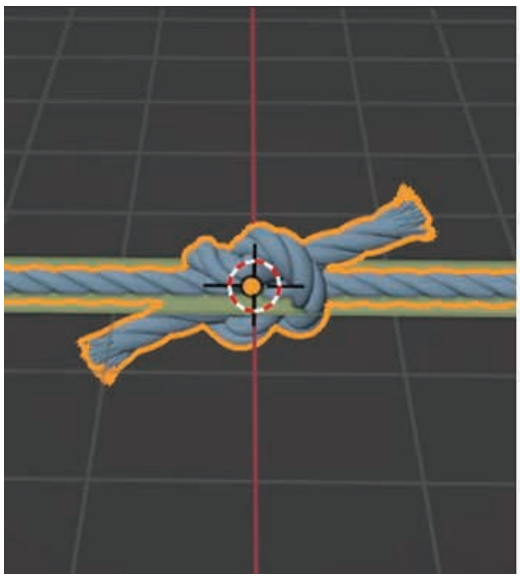


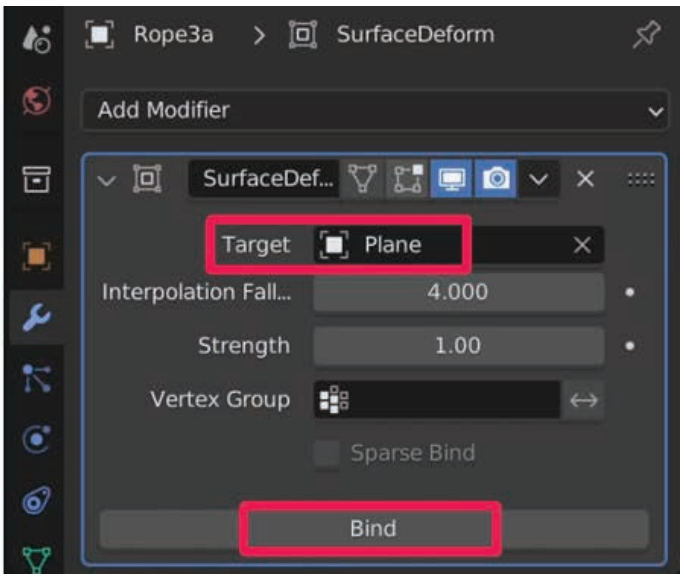
Make sure the **Hook modifiers** are above the **Cloth modifier**.

You can drag the modifier down.

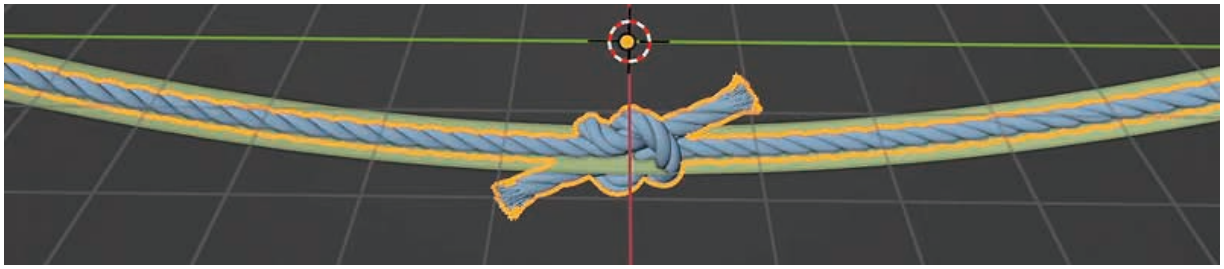


In Object Mode add a **Surface Deform modifier** to the rope mesh. **Bind** it to the cloth object.

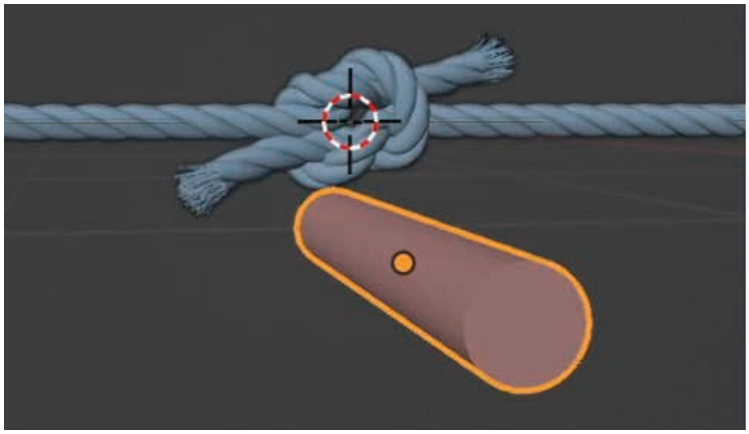


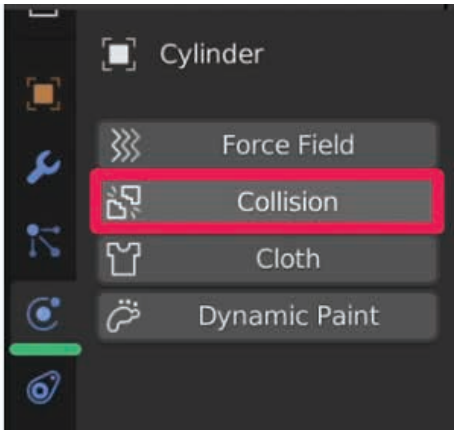


Now when you **press the Spacebar** to test the simulation, the rope follows the cloth object!

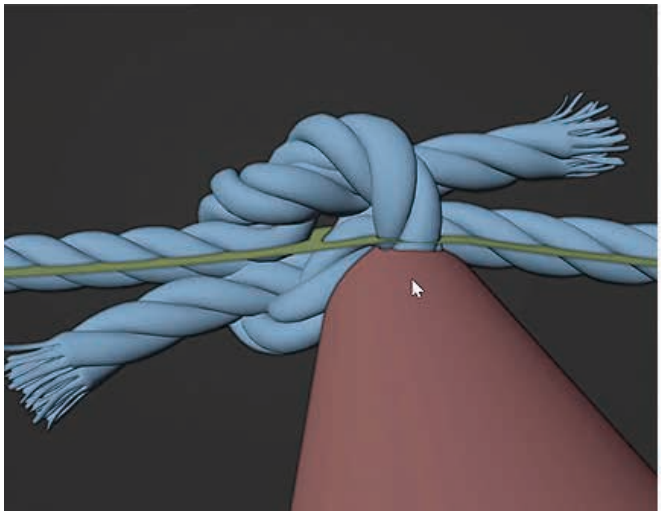


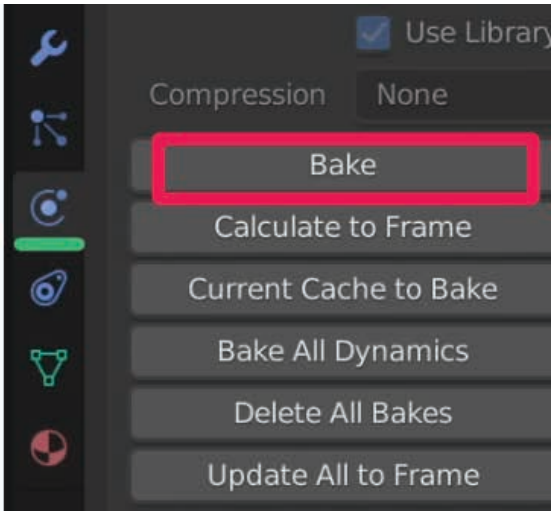
So far so good, let's make it even more interesting with some collision with another object. Add an object, for example a **Cylinder** (Shift+A > Mesh > Cylinder), and **enable Collision** for it.



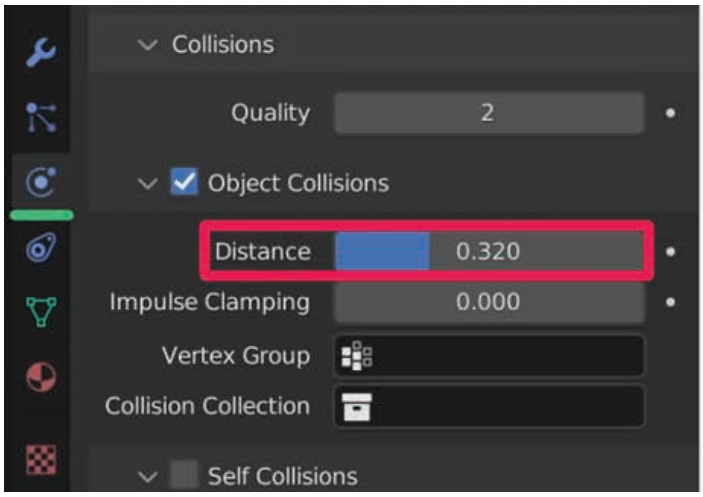


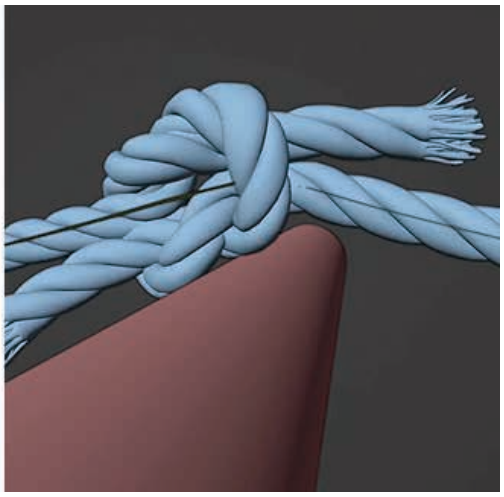
Now **Bake** the Cloth cache. When you **press the Spacebar**, the fancy cable will interact with the collision object.



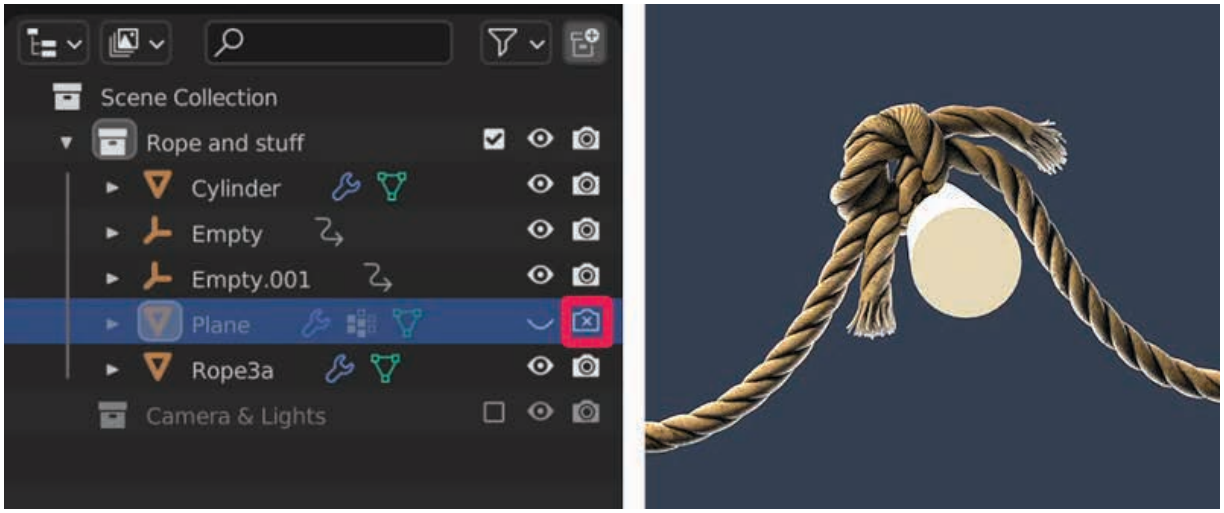


To compensate for clipping/intersecting of the simulation you can increase the Collisions > Object Collisions > **Distance** value. You'll have to **Delete the Bake** before you can change any values of the Cloth simulation. A Distance of 0.32 worked well in this example.





Remember to turn off render visibility of the cloth object so it doesn't show up in your render.

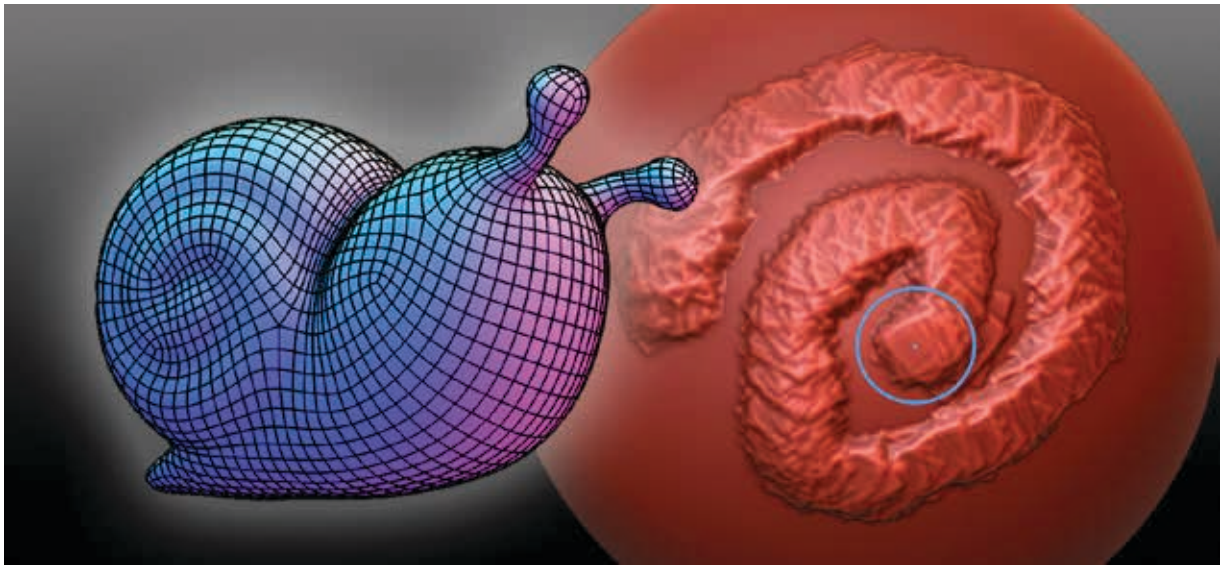


Rope with knots 3D model by "mStuff" on Blendswap (cc0).

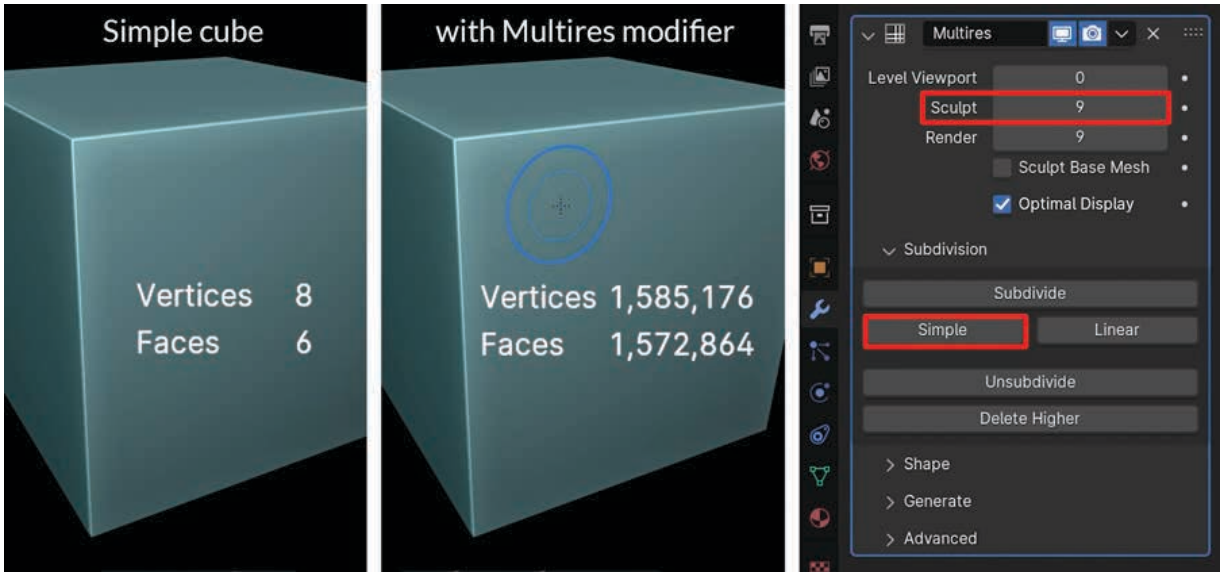
SCULPTING

Improve Sculpting Performance Instantly

If you're worried that you may not have a good enough computer to sculpt with in Blender, here's a simple trick that will instantly improve your sculpting performance.

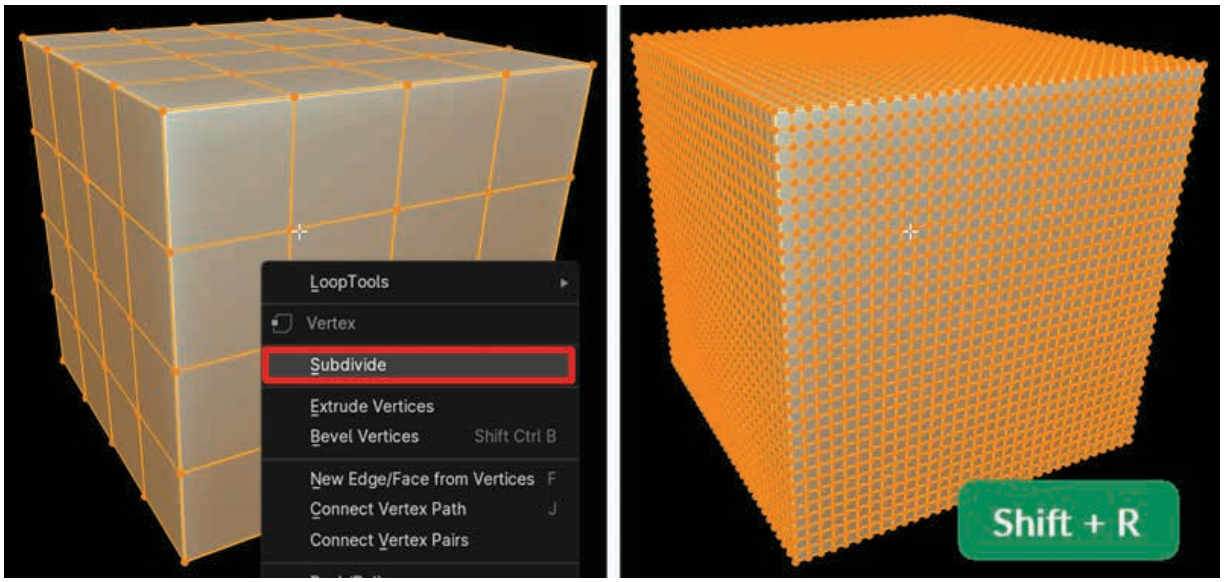


First of all, you have to understand that if you start with too little geometry, and then add a huge amount of levels on the Multires modifier until you have enough to sculpt with, you will get slow performance, even on a good computer.

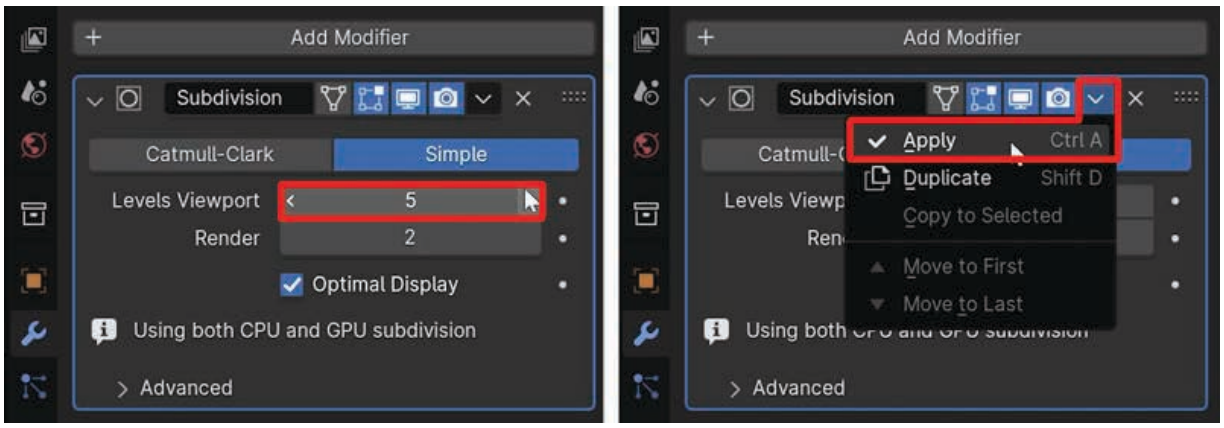


How not to do it: Start with very little geometry, and crank up the modifier to a crazy level…

However, if you subdivide the same object first, either in Edit Mode by selecting all, right-clicking and choosing **Subdivide**, then pressing **Shift+R** a few times to repeat that…



...or by adding a **Subdiv modifier**, increasing the levels and **applying** the modifier.

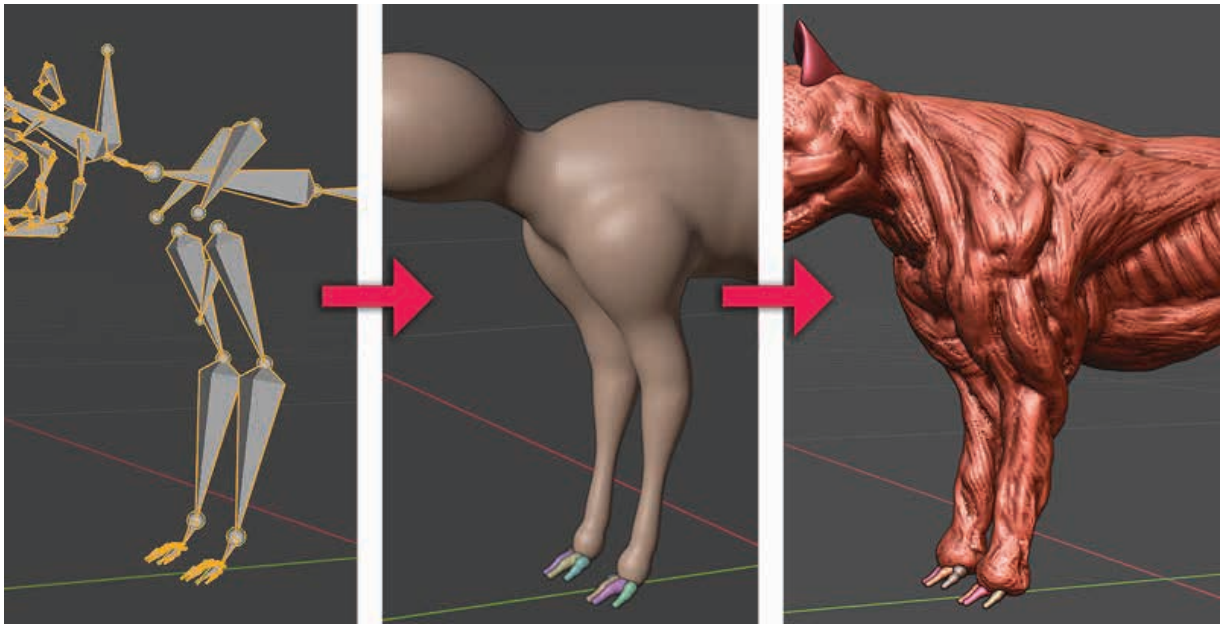


You'll get a much better performance out of the Multires modifier after that.

And you then need a smaller amount of Multires levels to get the same amount of geometry to sculpt with.

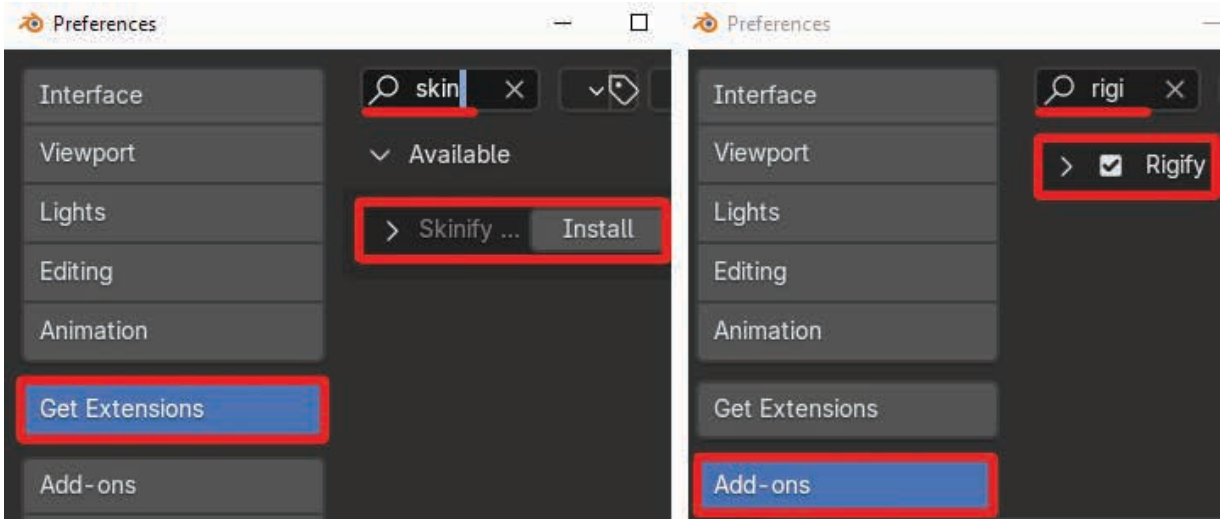
Create a Sculpting base mesh with the Skinify add-on

There are plenty of ways to create a base mesh in Blender for sculpting: The Skin modifier, Metaballs, even simply appending a base mesh someone else already made. One really interesting method that is less often mentioned is the **Skinify modifier** that comes with Blender by default. This creates geometry based on selected bones, with one click.

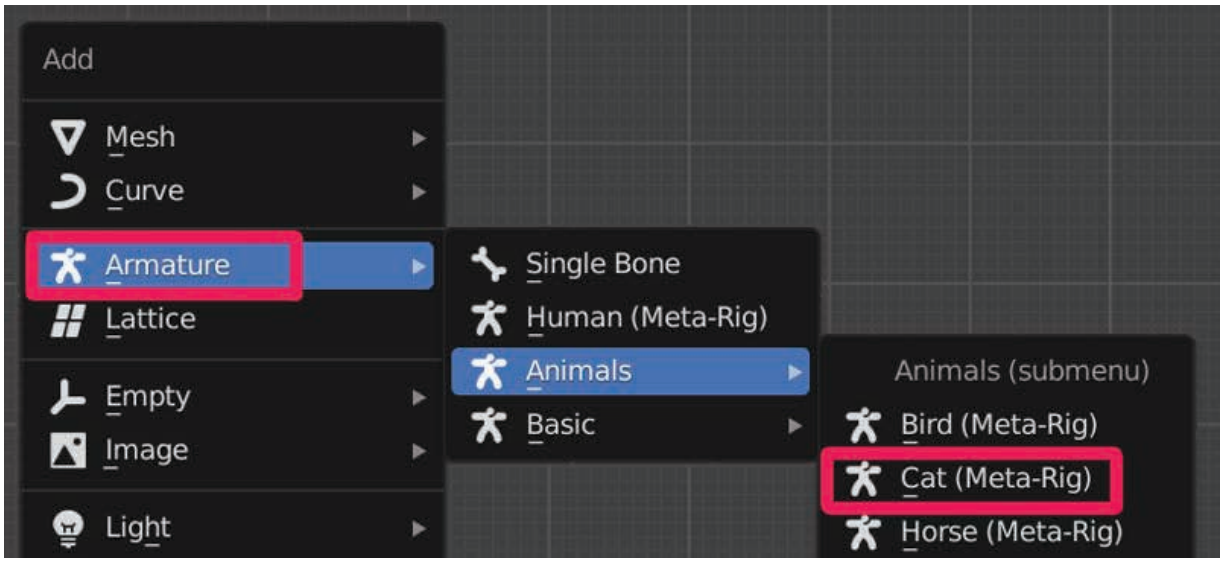


From Armature to smoothed, inflated base mesh and finally sculpted detail.

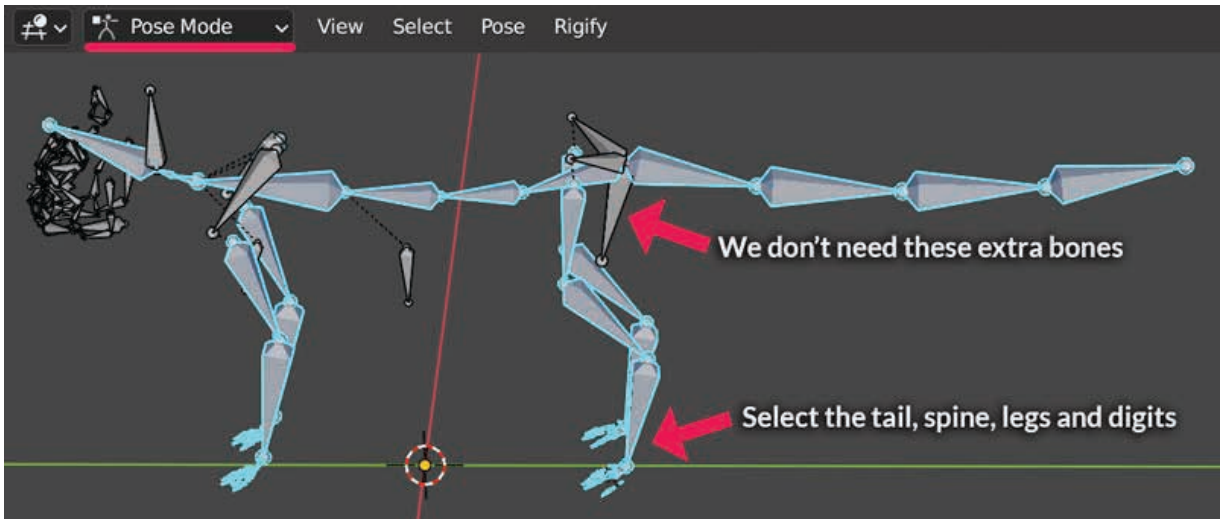
First, enable the **Skinify add-on** in Preferences. You may also want to enable the **Rigify add-on**, so that you can create Armatures easily from the Add menu. One is an Extension, the other an official Add-on.



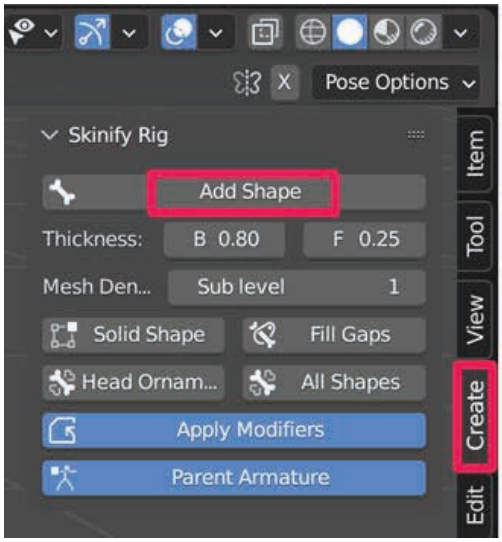
Then, **press Shift+A** and add an **Armature**. You can choose a simple or complex human armature ("Human Meta-Rig"), or some animals like a cat, a shark or a bird. Of course you can also add a Single Bone and create your own Armature from there. In this case, let's make a Cat.

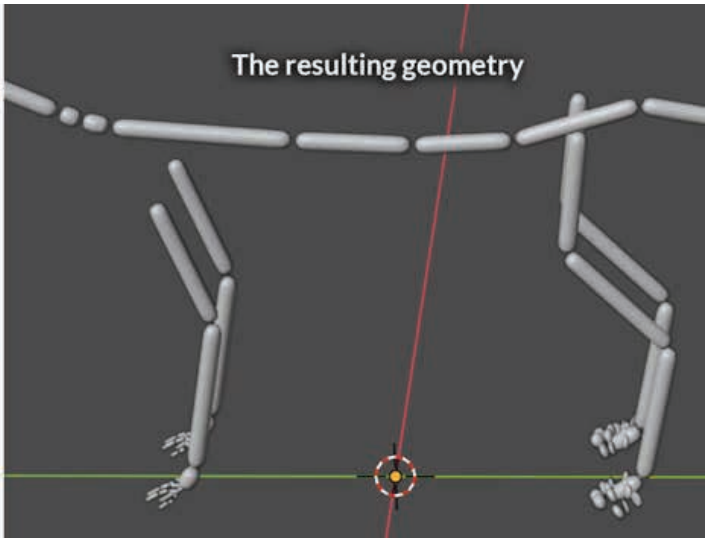


In **Pose Mode**, select the bones where you want to create geometry. Then in the **Create** tab in the Option Panel, Expand the **Skinify Rig** options.

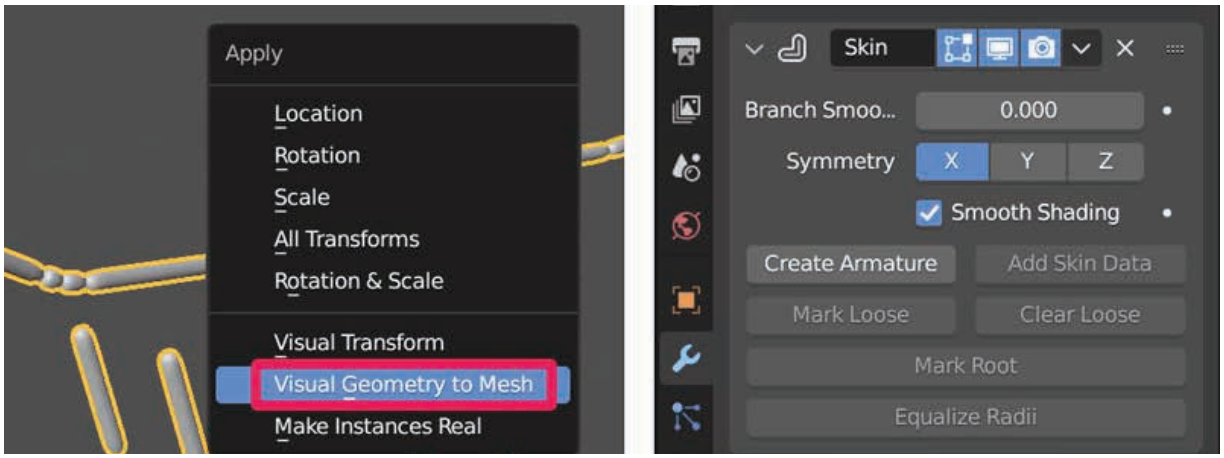


Click on **Add Shape** to generate the geometry. The default values are pretty good. You may get some error messages, but you can ignore these as the add-on does its job anyway. **Press Ctrl+i** to invert the selection from the geometry to the Armature, and **press H** to hide it, so you only see the new geometry.

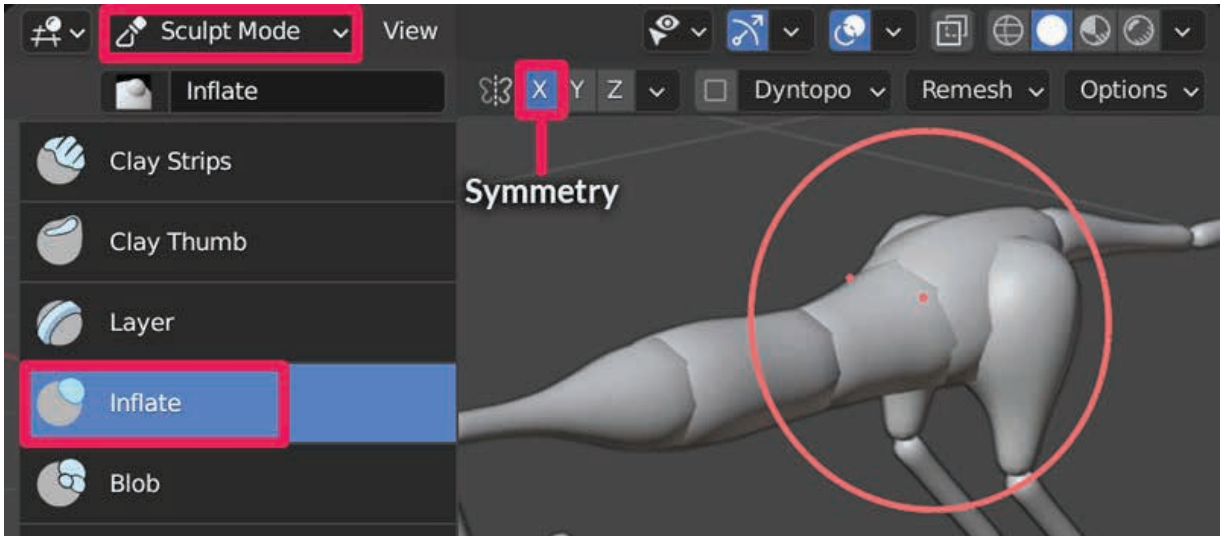




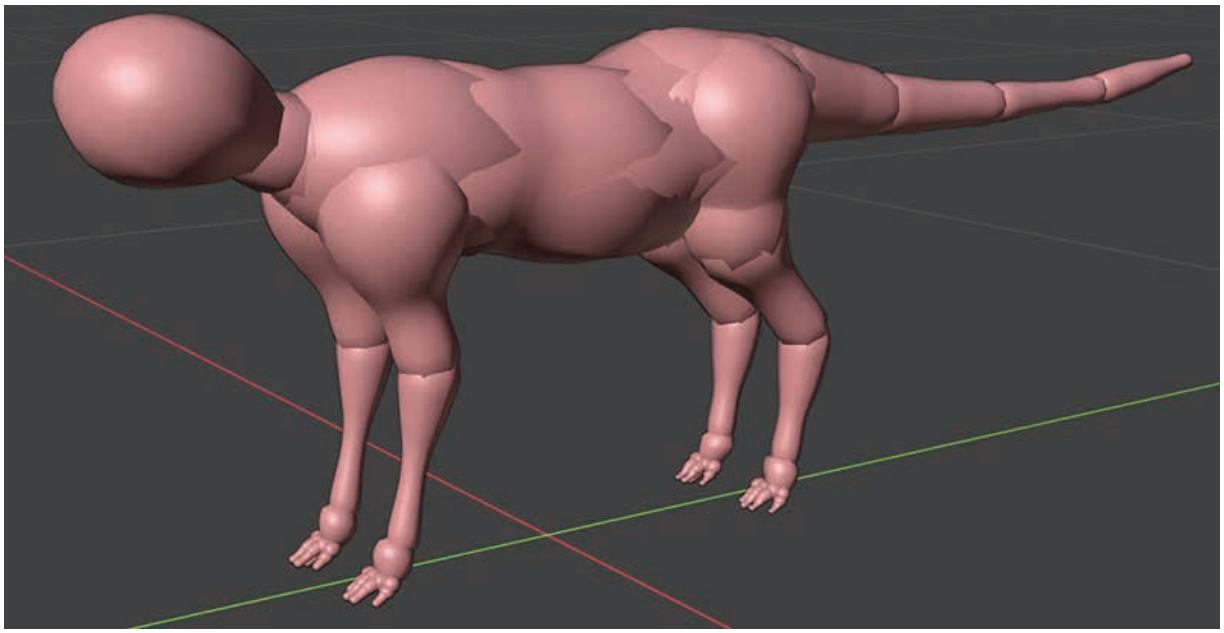
Select the geometry, **press Ctrl+A** and choose **Visual Geometry to Mesh**. This applies the Skin modifier and Subdivision modifier and turns it into actual geometry.



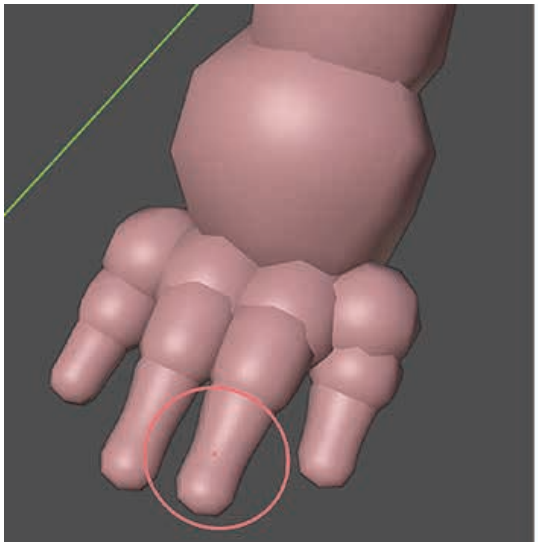
Go to **Sculpt Mode** and select the **Inflate brush**. Enable sculpting **Symmetry** along the X-axis. Now you can inflate the mesh as needed.

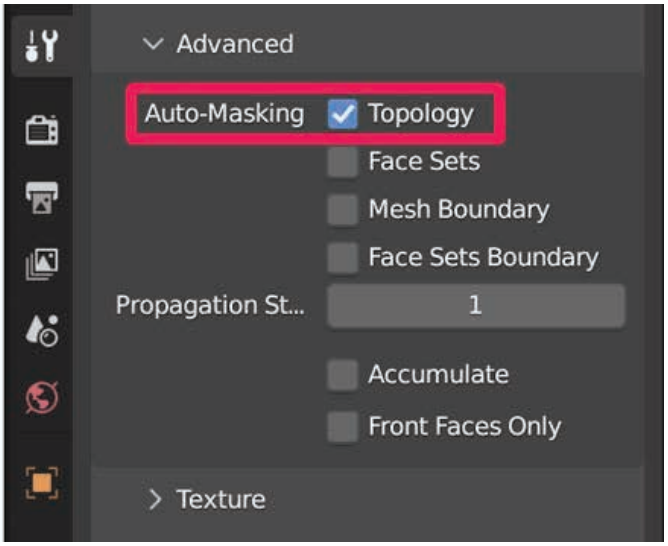


In just 3 minutes or so, we get this result:



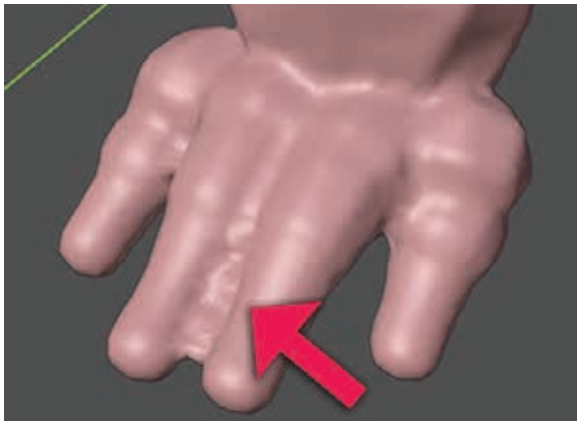
If you only want to affect one separate piece of geometry at a time, enable **Auto Masking by Topology** in the **Advanced** brush settings of the Inflate brush. And as with other brushes, **holding Ctrl** sets the brush to **Subtract** instead of Add, so if you want to "deflate" rather than inflate parts that's how you can do it.



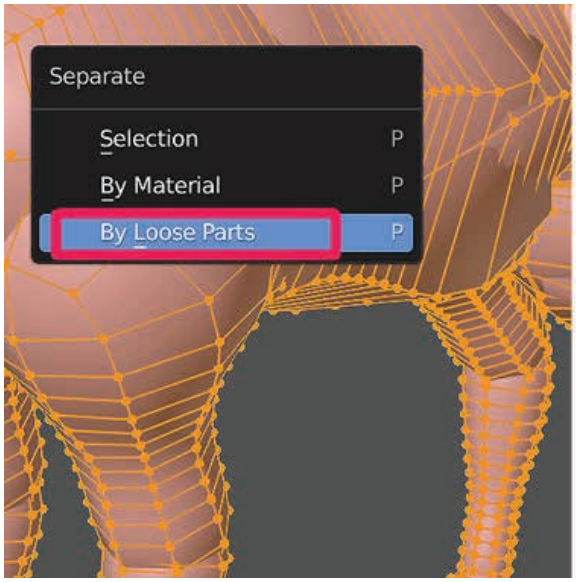


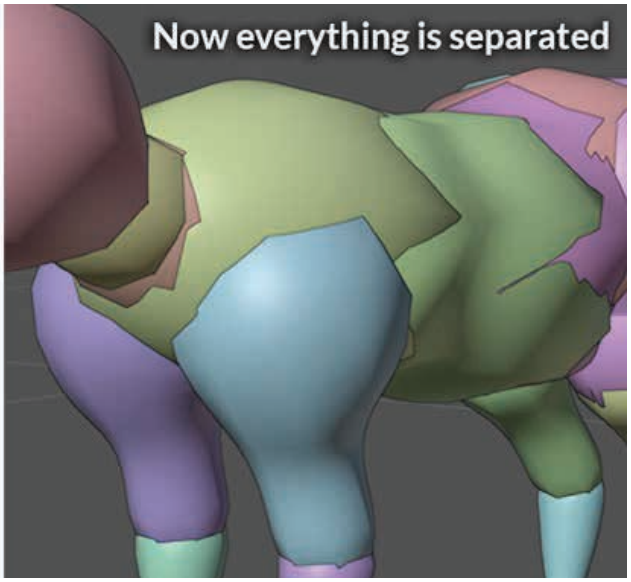
Now we can remesh some parts.

However if we just remesh everything at once, we get the problem that the digits are too close together and get merged in an undesirable way.



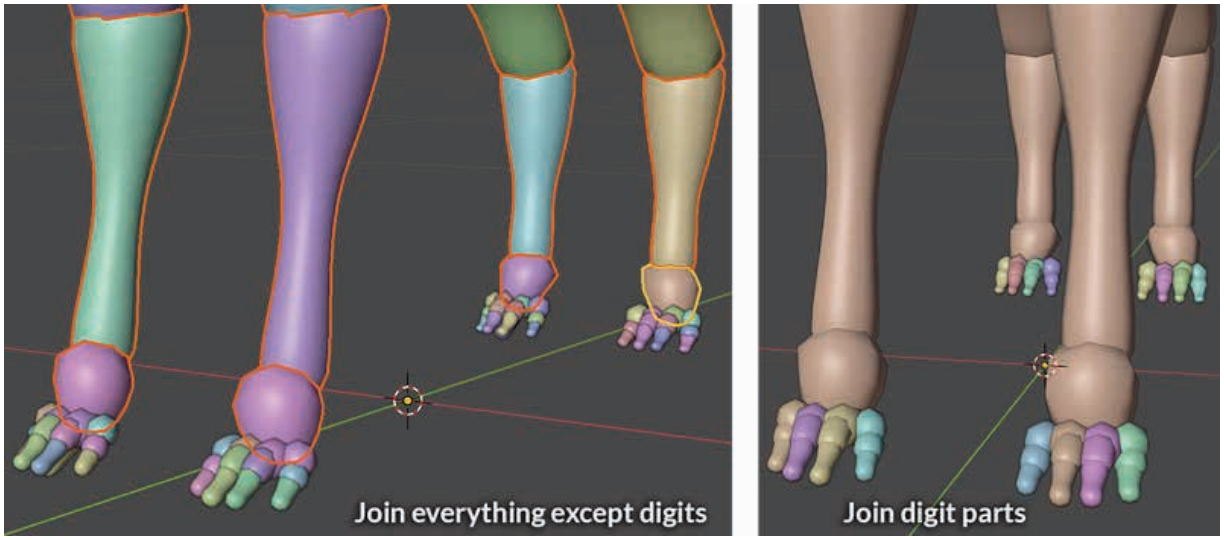
So instead, we will keep those digits separate. But first, we need to separate everything by going to **Edit Mode** and selecting all, then **pressing P** and choosing Separate **by Loose Parts**.



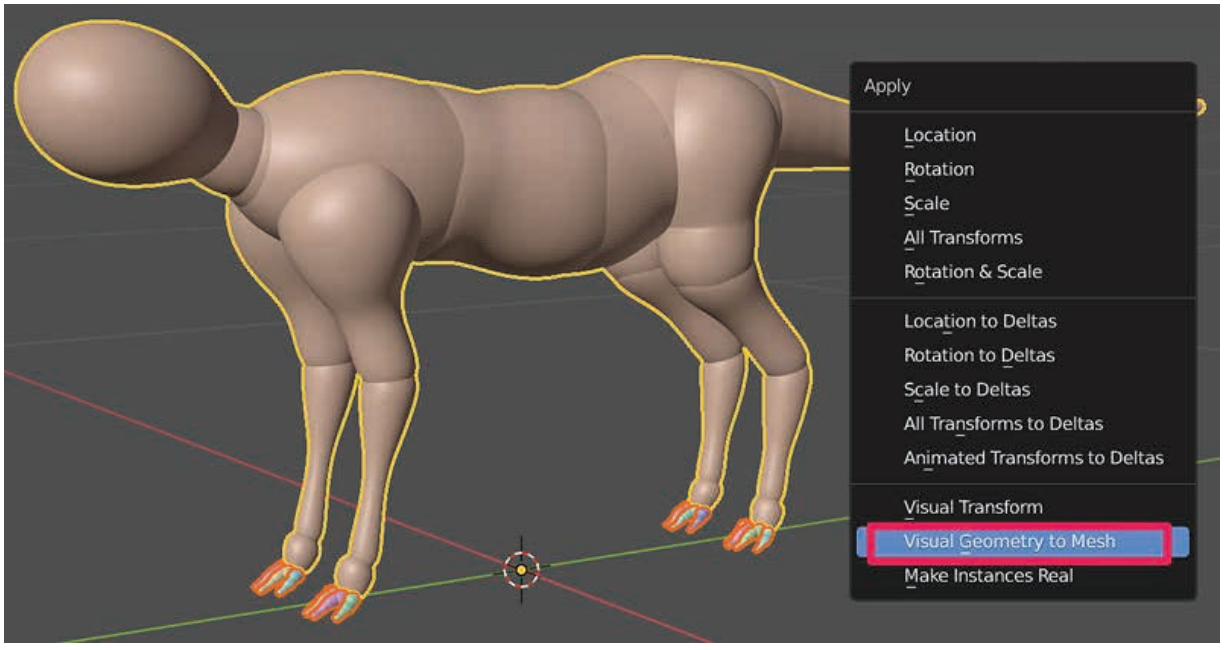


Click here to return to the index.

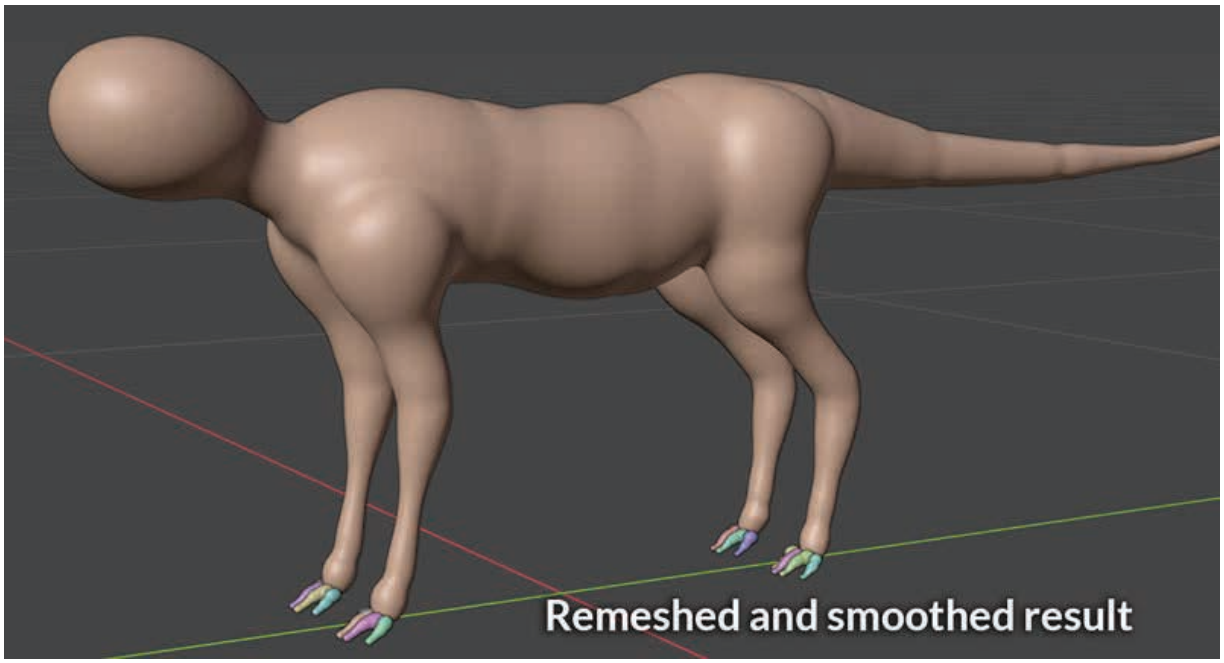
Now you can select parts that should be joined and join them with **Ctrl+J**. First join all the body parts, everything except the digits. Then join the objects that make each digit. So you have one object for each digit, and one object for the rest of the body.



You don't notice it because the meshes are all shaded smooth, but they are in fact quite low poly. So add a Subdivision modifier set to level 3 to each object to make them more smooth before remeshing. The fastest way is to select an object, and **press Ctrl+3**. Do this for each object and **press Ctrl+A**, choose **Visual Geometry to Mesh** (to apply the modifiers).



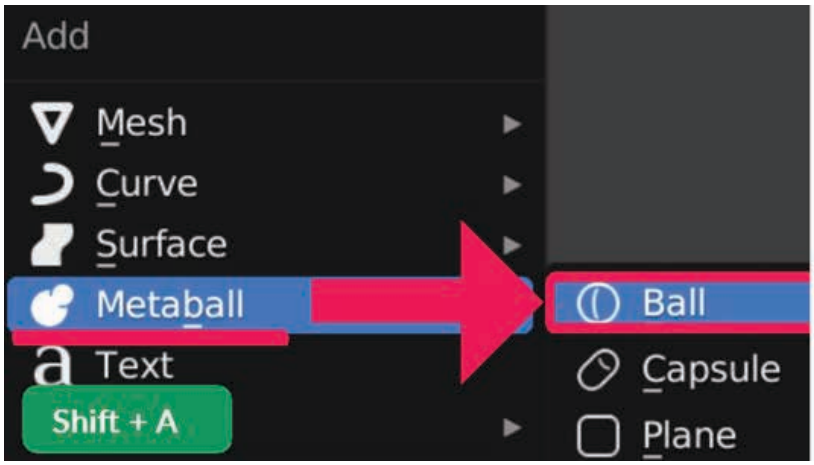
Go back to Sculpt Mode. Select the main body by hovering the mouse over it and **pressing Alt+Q**. Then **press Shift+R** to preview a remesh resolution. Finally **press Ctrl+R** to remesh. Do the same for the digits. You can smooth the remeshed meshes out by **holding Shift** and sculpting over the seams.

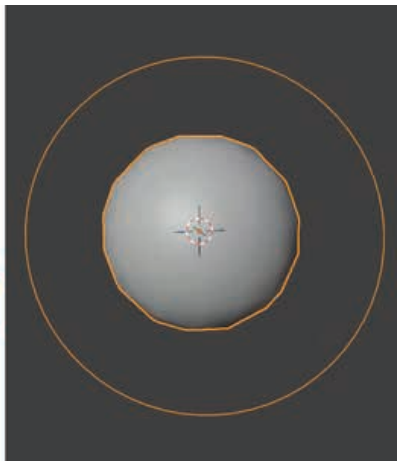


Now you've got a nice base mesh to sculpt a cat, or... some kind of monster, whatever this is:

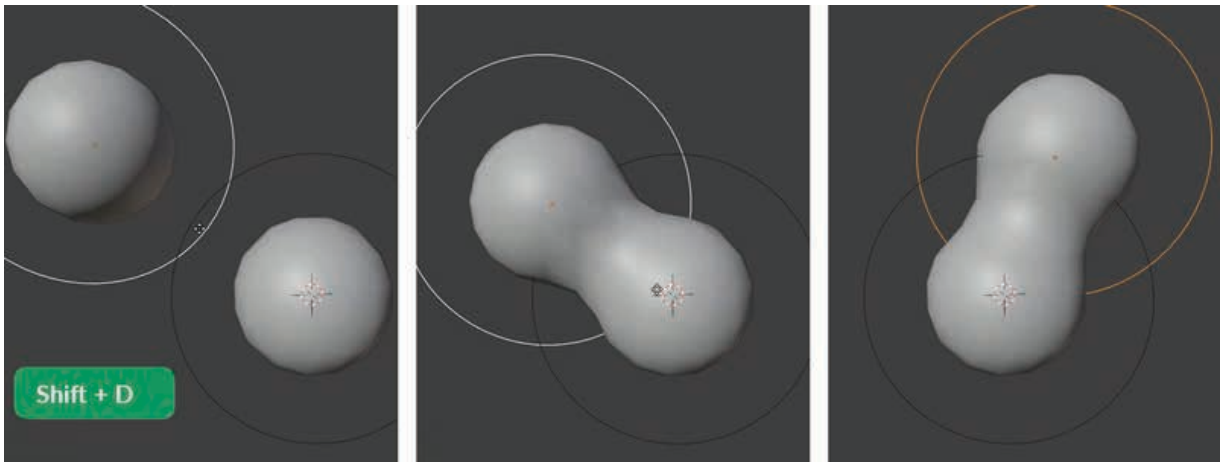
Create a Sculpting base mesh with Metaballs

Metaballs give us another easy way of creating a base mesh for sculpting. In Object Mode **press Shift+A** and select **Metaball > Ball**.

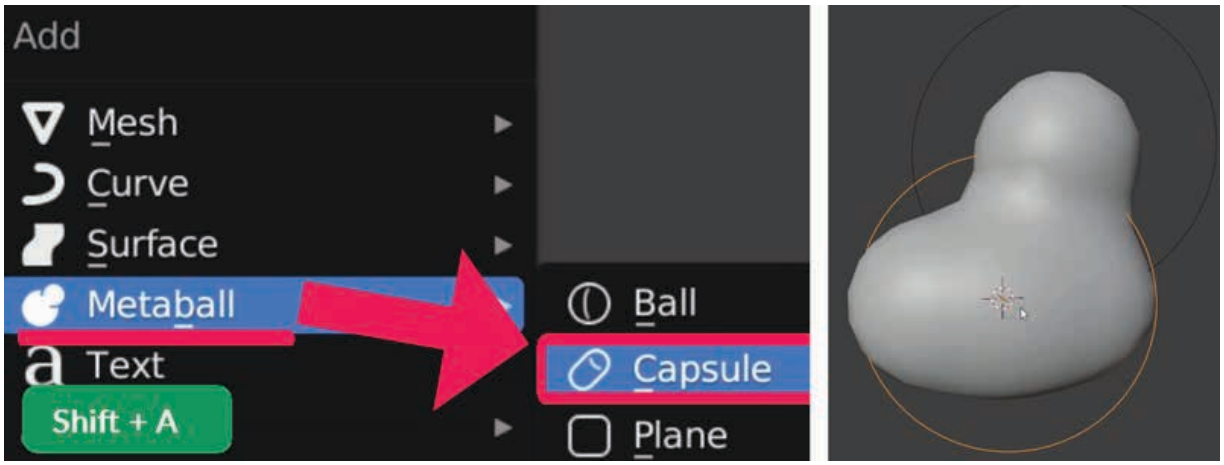




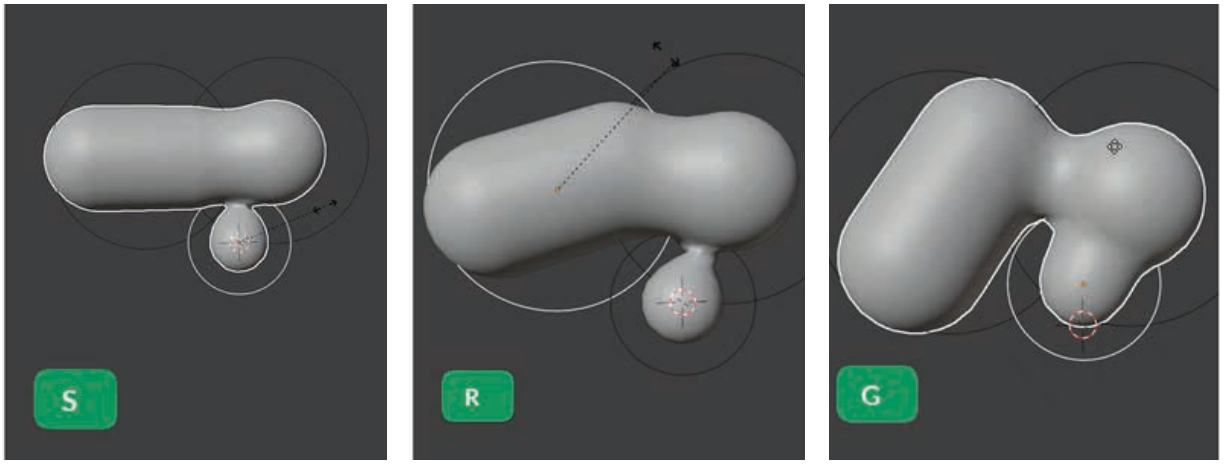
Press Shift+D to duplicate it, then move it to where you want to add it to the overall volume.



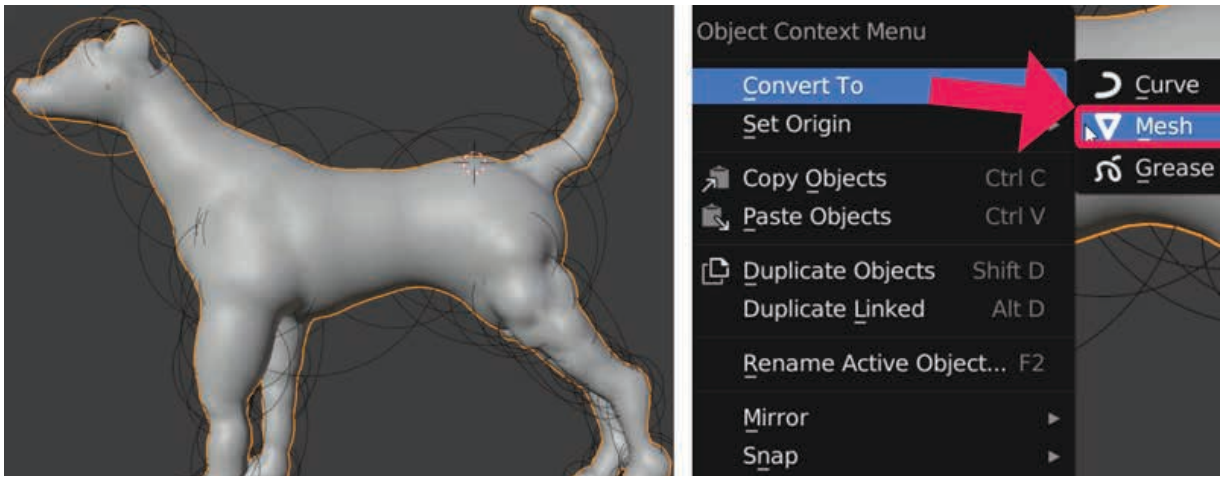
There are also a few other Metaball primitives to choose from.



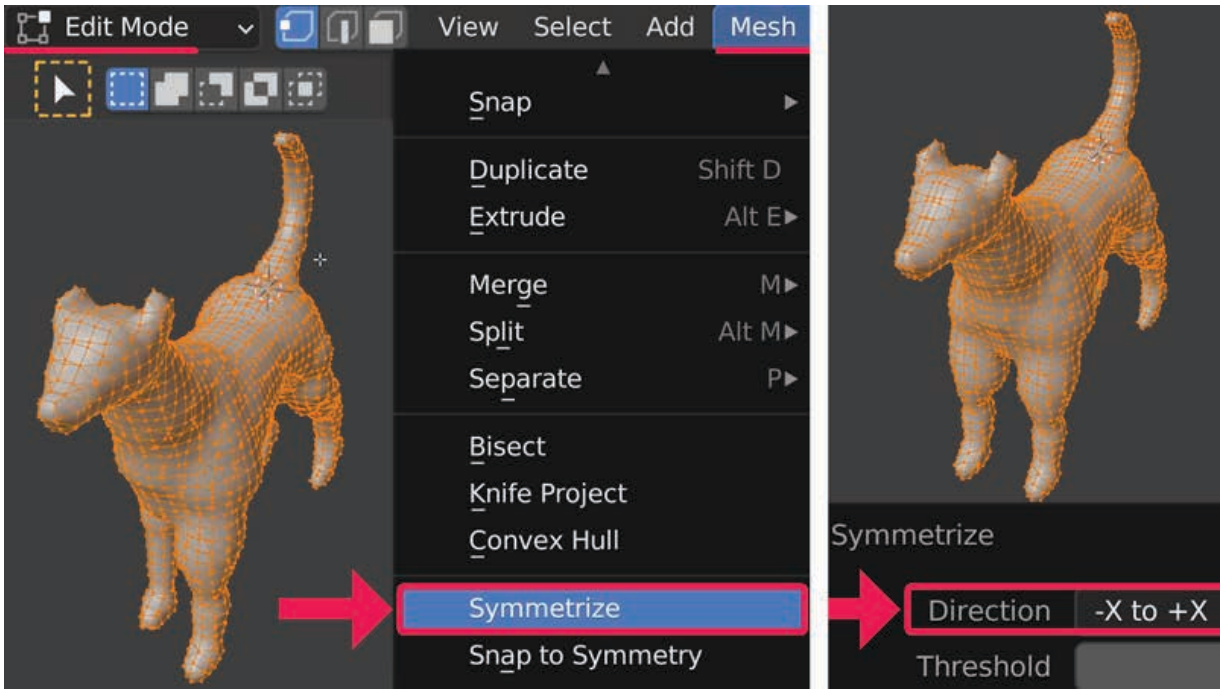
You can **press S** to change their size, **R to rotate** and of course **G to move**.



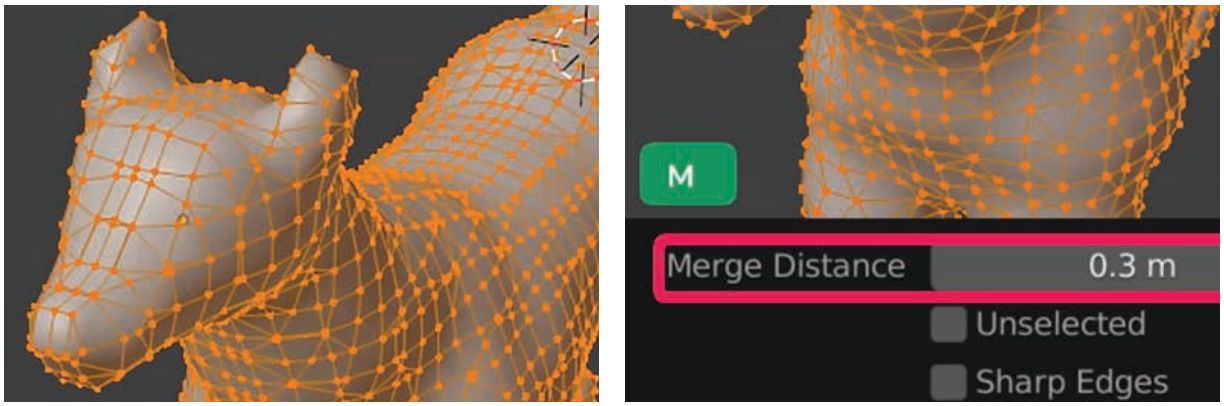
When you're happy with the blockout, click on the main shape instead of the individual circles. Right-click, and choose Convert to Mesh.



In Edit Mode, select all and go to **Mesh > Symmetrize**. You may have to switch the Symmetrize direction.



Then **press M** and choose by **Distance**, so you can clean up overlapping vertices by reducing the **Merge Distance**. Now you have a symmetrical base mesh that you can sculpt on.

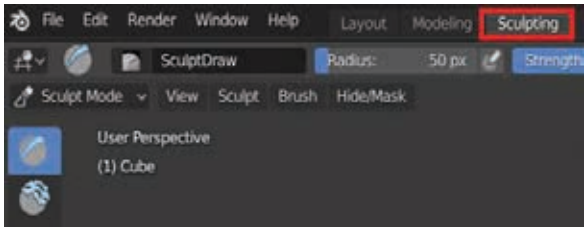


Sculpting Adding Detail

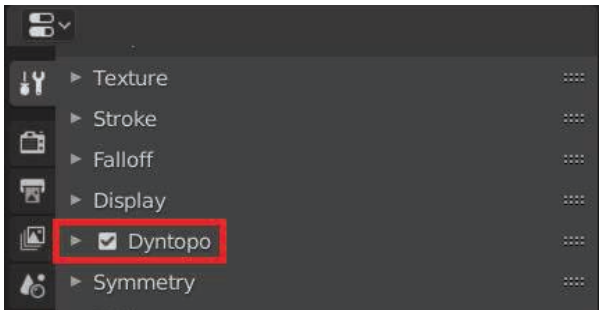
Here we have a model which has fine detail, except in one area where you can see the shapes of the low resolution mesh, as well as some artefacts, that we want to get rid of with sculpting.

But how to do that smoothly?

Click on the Sculpting tab to go to the sculpting layout.

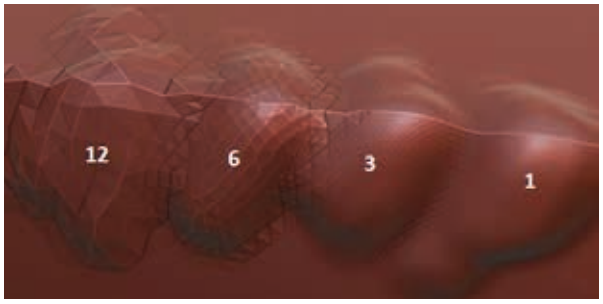


Turn on **Dyntopo** and click **OK**.

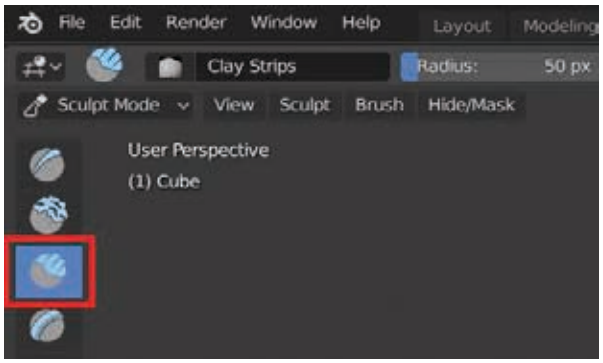


This turns on Dynamic Topology, which adds more resolution to the mesh in the parts where we sculpt.

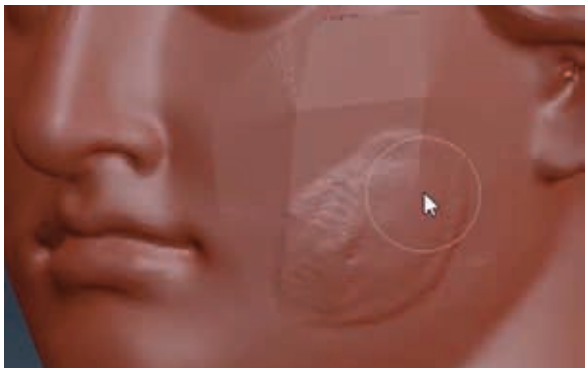
Detail size is set to 12 by default. The smaller the number, the finer the extra detail that Dyntopo creates.

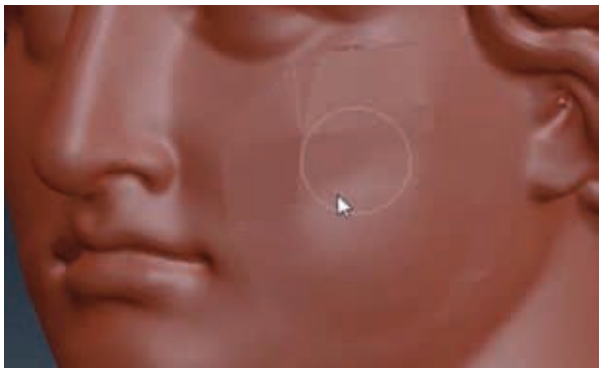


One option is to add some Clay strips (with the **Clay strips brush**, pictured on the right) on top and smooth it out by **holding Shift** while sculpting.



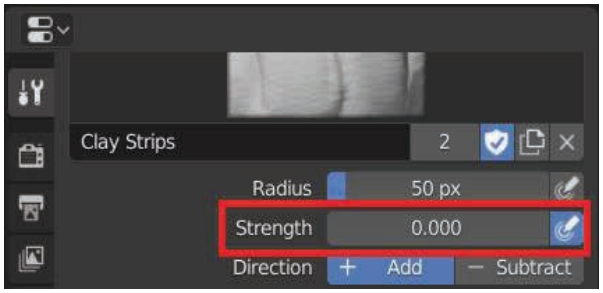
As you can see below, this kind of works but does add a layer on top of the model, which you then have to sculpt out by adding negative clay strips by **holding Ctrl** while sculpting, and then smoothing again.



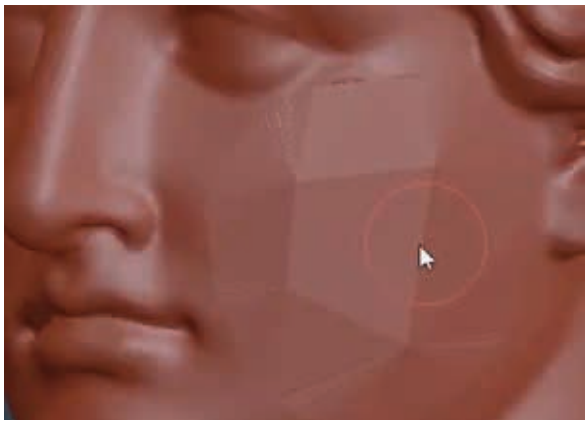


This is not a very easy process.

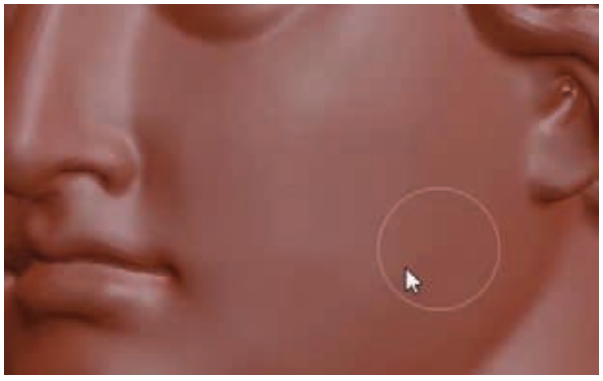
An easier solution is to set **Strength** to zero and sculpt to add detail.



Now we sculpt on the problem area. You don't see any difference because the strength is set to 0. That's what we want. You don't see it, but more detail is in fact added because Dyntopo is still doing its thing.



Now we smooth it out by **holding Shift** while sculpting and - magic! The problem area smooths out perfectly and without adding thickness on top.

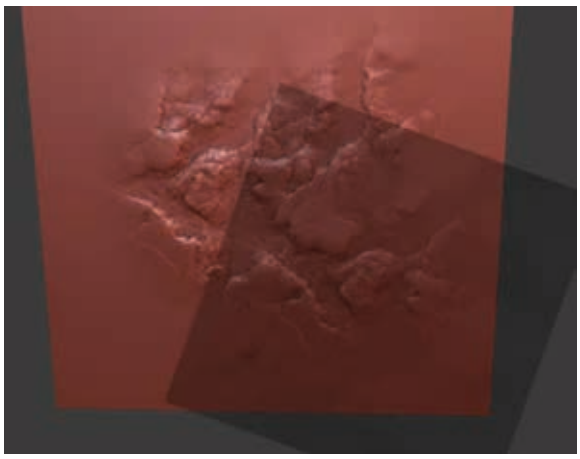


Sculpting with Textures and Stencils

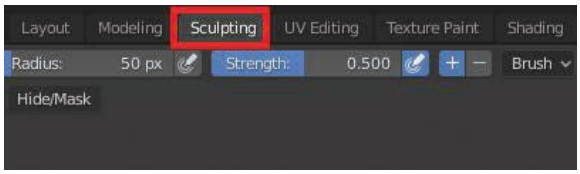
Previously we looked at texture painting using Stencils.

Did you know that in Blender you can also use Stencils to sculpt details?

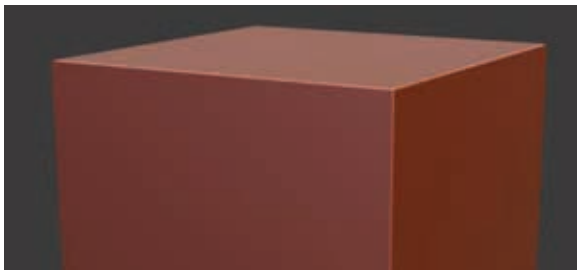
Just like with painting textures using stencils, you can quickly add detail that would be difficult and time consuming to add otherwise.



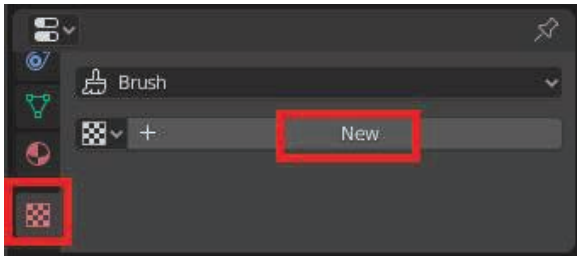
Let's sculpt on our default cube for this example. First, we go to the **Sculpting layout** by clicking on that tab in the viewport.



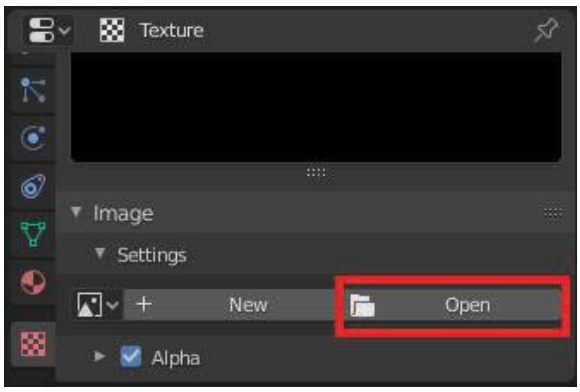
Now our Cube looks like this. That's because Blender has automatically given it a Matcap viewport shader, which is useful for Sculpting.



In the Texture tab, **click on New** to create a new texture.

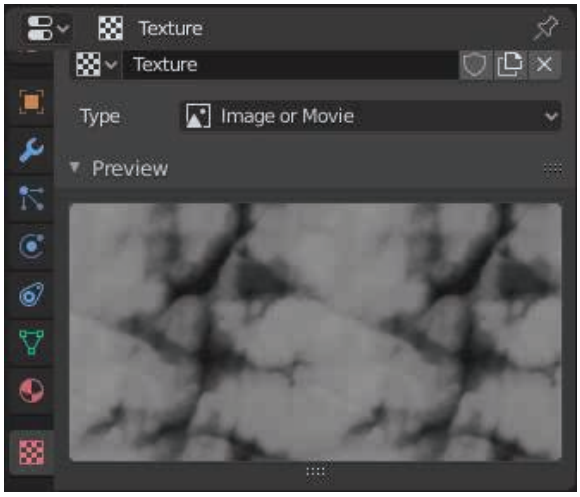


Click on Open and browse to find a texture you want to use for sculpting. Any image will work.

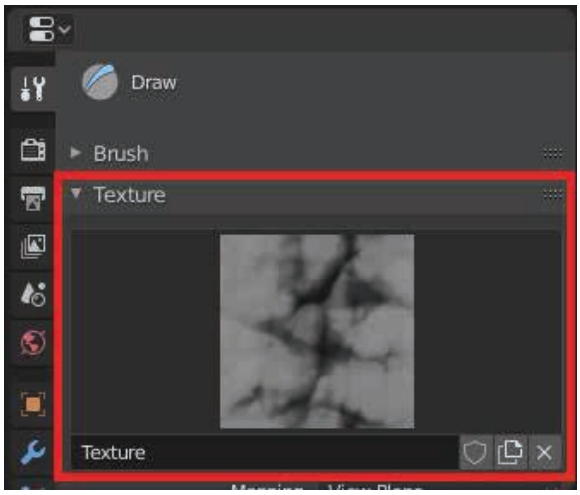


On the website www.textures.com I searched for “rock” and found this free texture set called “Free Volcanic Rock Cliff Material - PBR0188”.

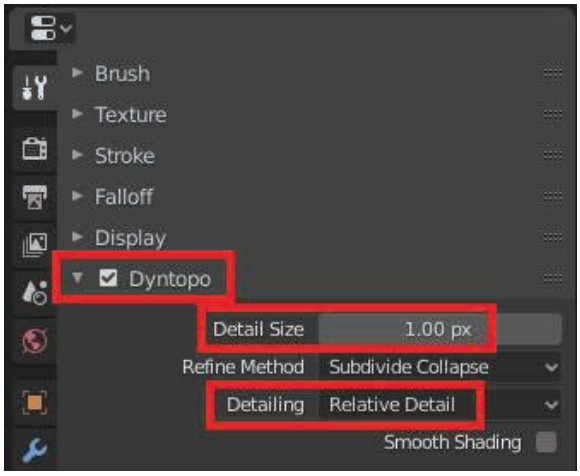
From that set, I downloaded the Height map (a black and white texture that contains height information in the form of black and white values).



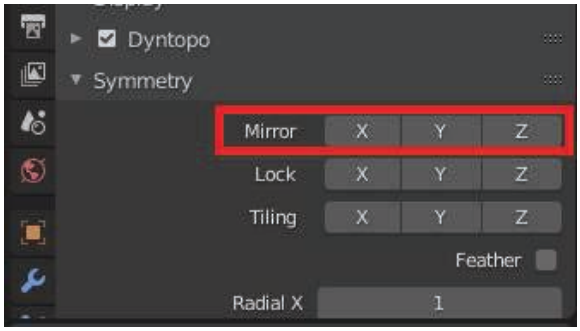
In the **Brush tab** under Texture choose the texture you just loaded.



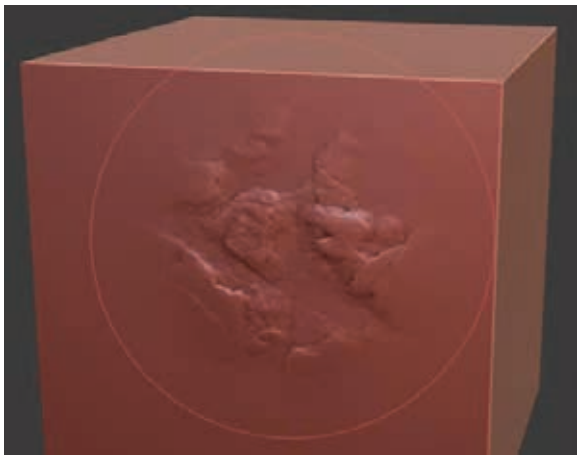
Turn on **Dyntopo** and set **Detail size to 1**, Detailing to **Relative Detail**.



Turn off **Symmetry**.



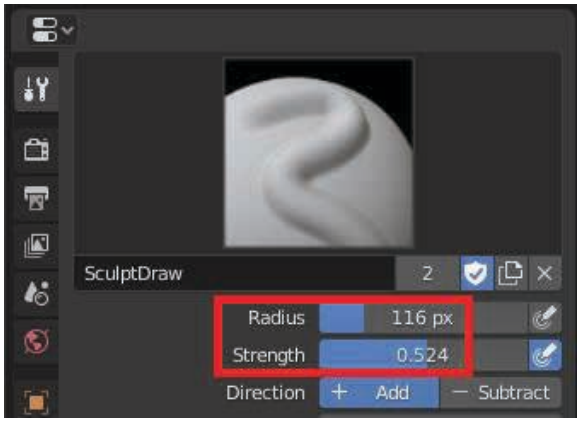
Now if you click somewhere on the Cube, it will create displacement based on the texture.



With **Radius** and **Strength** you can control the size of the brush and the strength of the displacement.

Keyboard shortcuts for these are:

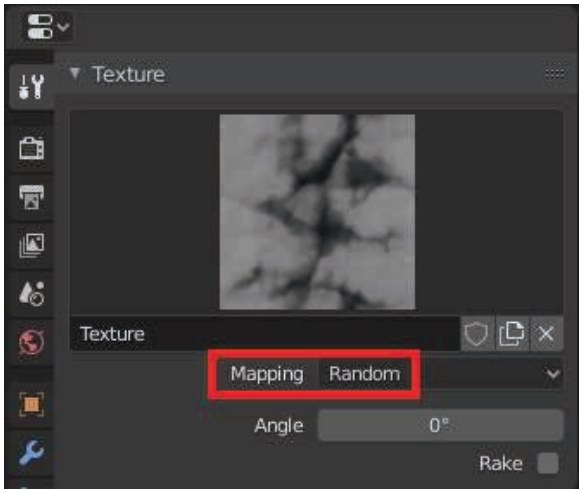
F for Radius
Shift+F for Strength



Click here to return to the index.

So far we're using the Mapping method **View Plane**.

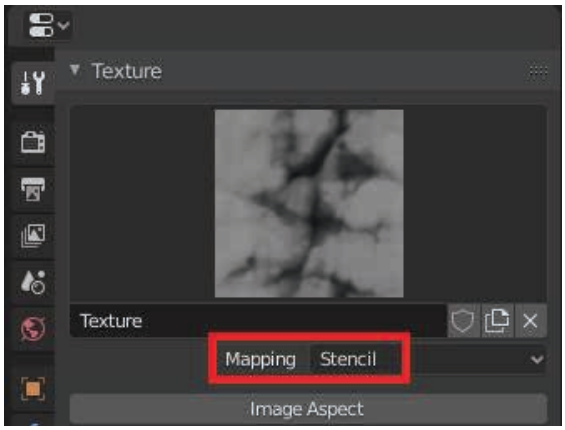
The downside of this is, every time we click on our object, the texture has the same rotation, and we quickly spot the repetition. It would be nice if we could randomize the rotation, to make it more organic. To achieve this, select Mapping method **Random**.



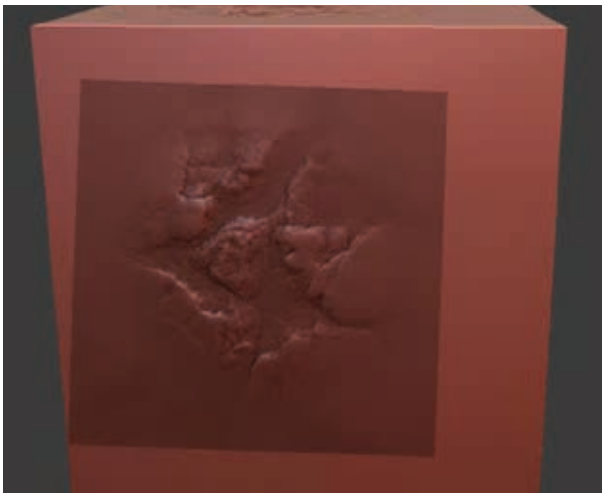
Now each time we click, the texture is rotated randomly and we can start combining it on the surface of the object.

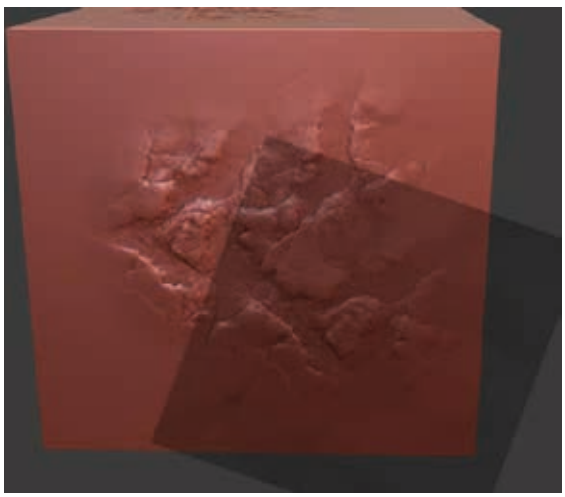
We can also use **Stencil** mapping.

This gives us the most control, as we can see exactly where and how we are going to apply the texture as a displacement.



Hold the right mouse button above the stencil and drag to move it. Shift + right mouse button to scale the stencil. Ctrl + right mouse button to turn the stencil.





Creating a Robot base mesh with Voxel remesh

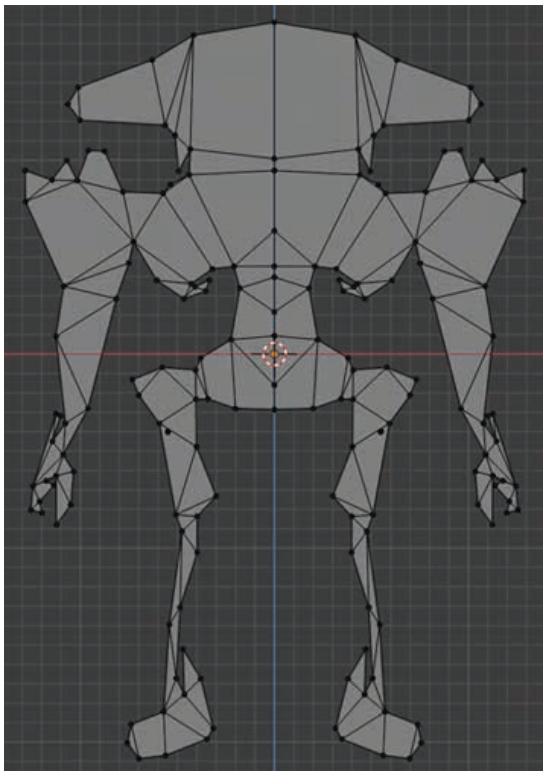
In the **concepting stage** of modeling some kind of mechanical design like a robot, **sculpting** is a really good choice.

It's a fast way to get to a design. But before you can sculpt, you need a base mesh that gives you something to sculpt on.

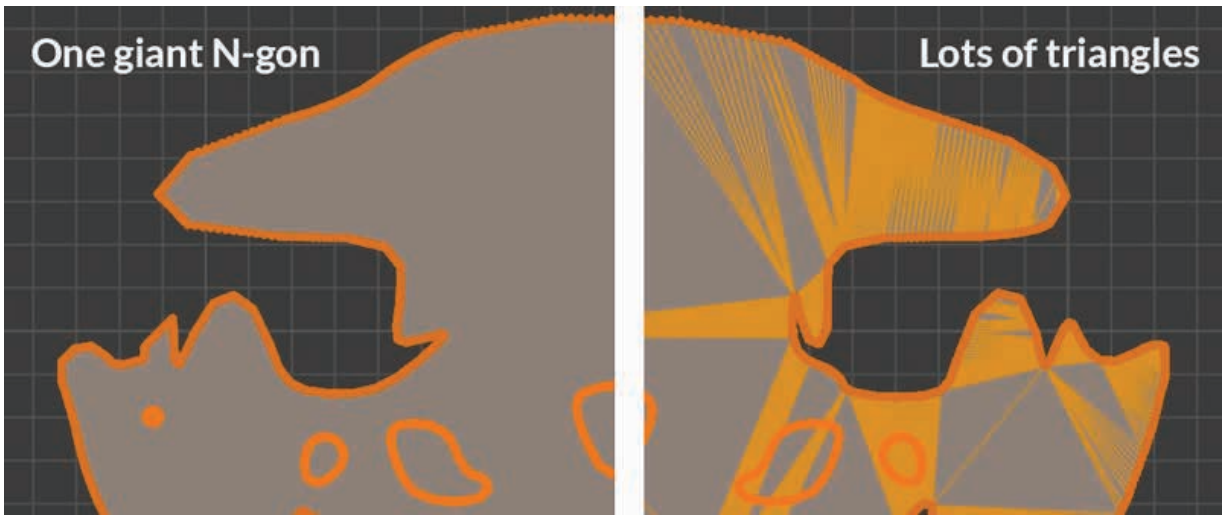
You can make this pretty easily, especially since the introduction of **Voxel remeshing** in the **Remesh modifier**.

First, **drag and drop a thumbnail** that you've created earlier **in the Front viewport**.

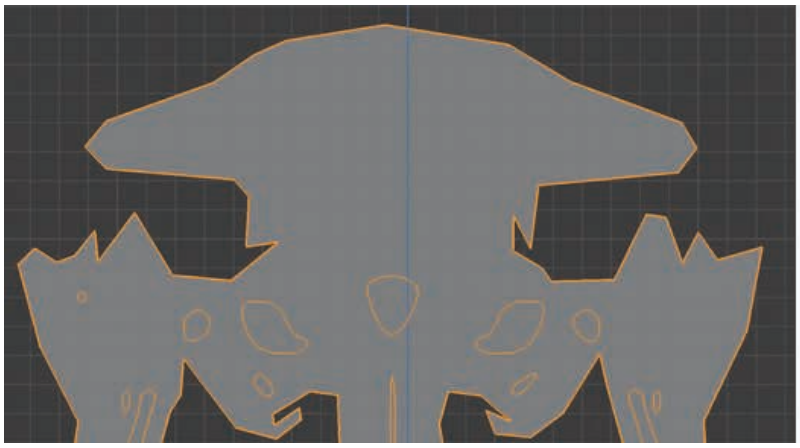
Using the process described in this video, you can convert this drawing to a mesh.

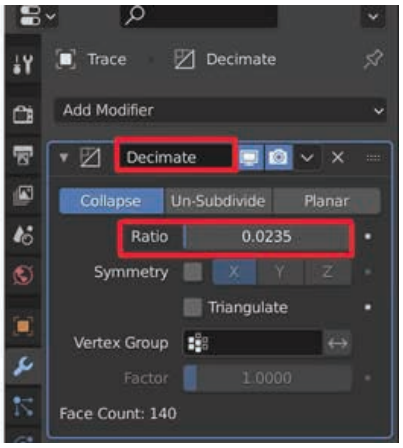


After converting to mesh, I cleaned it up by first **triangulating** the mesh with **Ctrl+T**. We need to triangulate the mesh so we can add a **Decimate modifier**, otherwise it's just one big N-gon. Bear with me here, I know it looks terrible now but it's gonna look awesome - I promise!

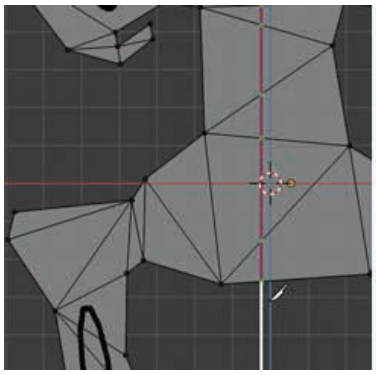


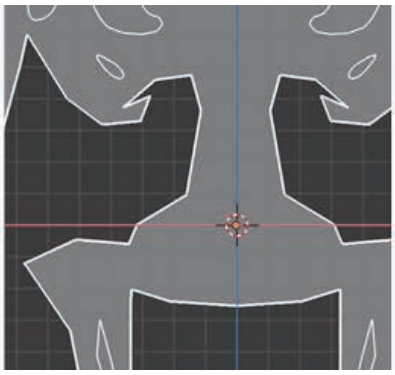
Then by adding a **Decimate modifier** with a very low **Ratio** value we can reduce the amount of vertices and make it look a bit more angular and machine-like, and less organic. **Apply** the Decimate modifier.

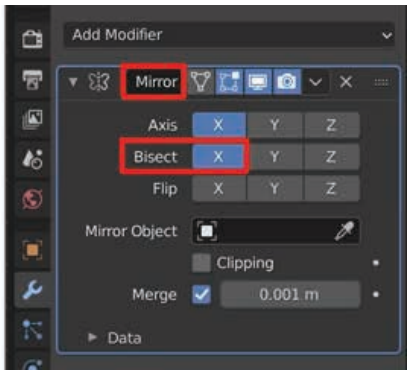




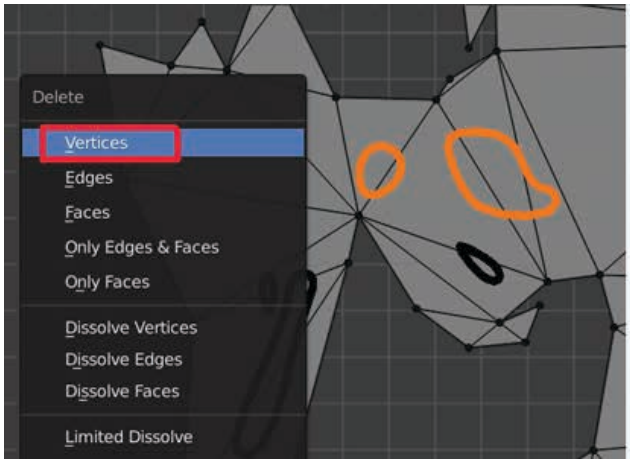
Then I made a vertical cut using the **Knife Tool**, by **pressing K and A** (for Angle) to be able to make a perfect vertical cut. Then I mirrored the mesh using a **Mirror modifier** with **Bisect** checked, so I don't need to remove half of the mesh.

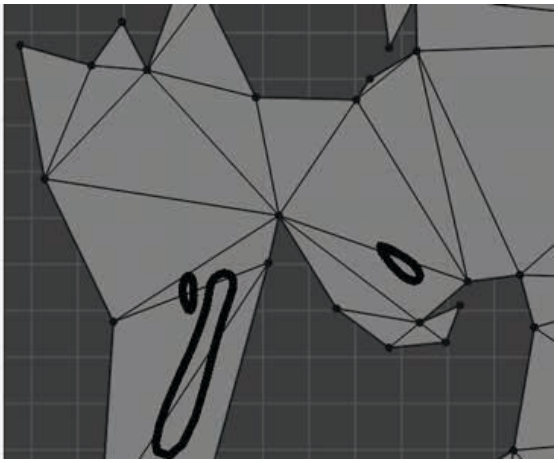




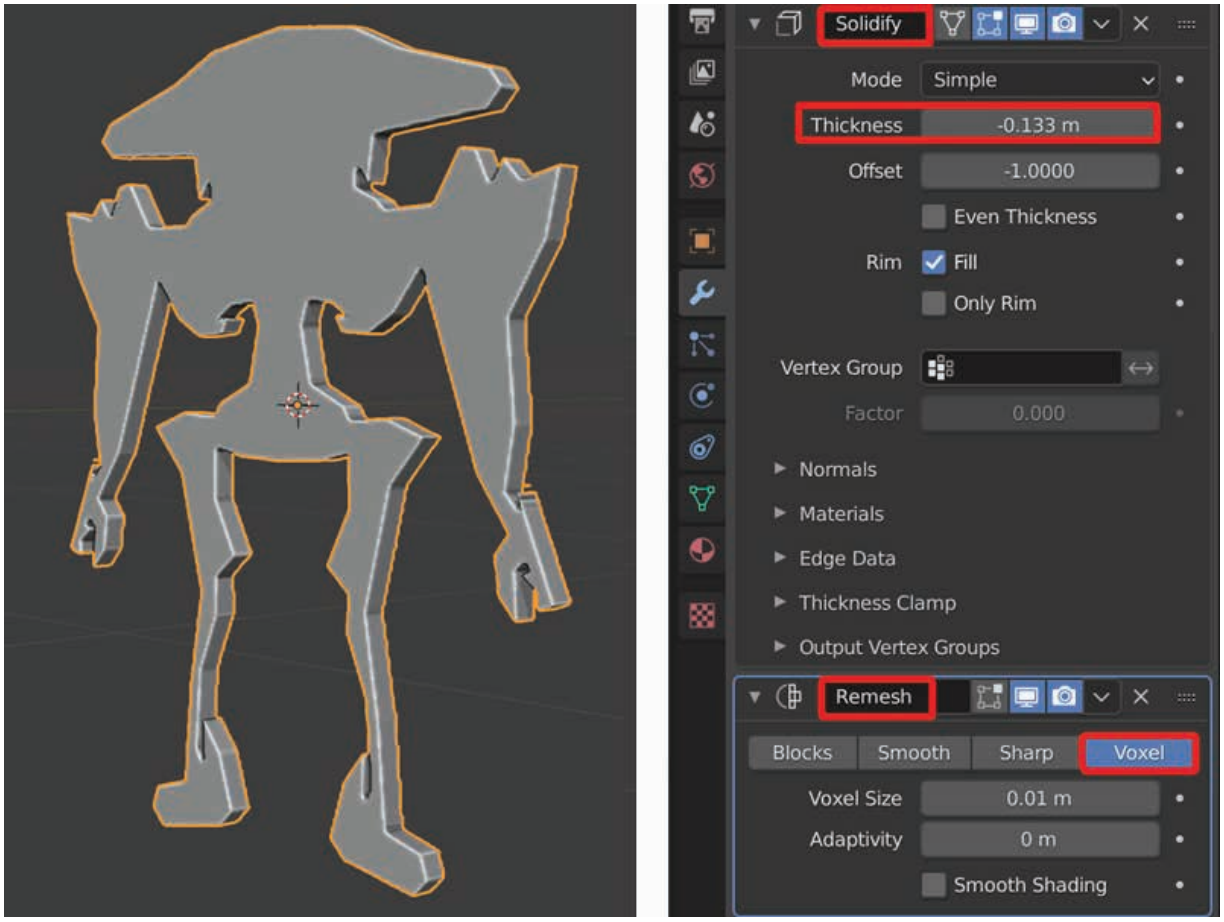


I then applied the Mirror modifier and cleaned up the mesh by removing unnecessary vertices by selecting them and **pressing X**, then choosing **Vertices**.

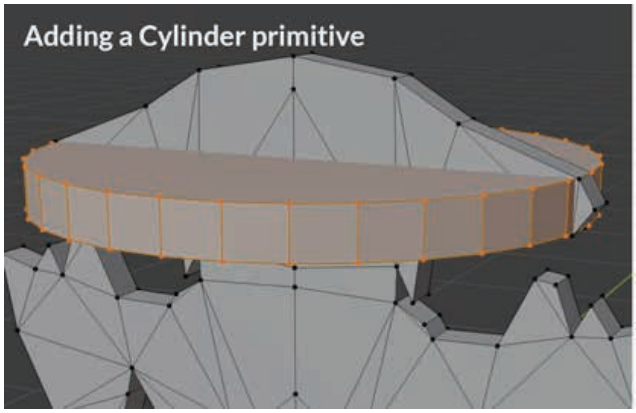


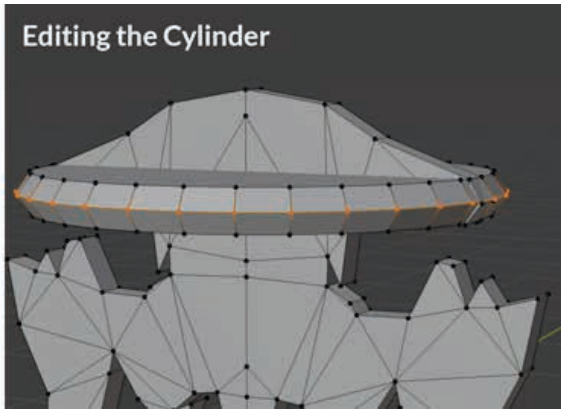


Then by adding a **Solidify modifier** we can give this mesh some thickness, and adding a **Remesh modifier** set to **Voxel** with a low value like 0,01 we remesh it. **Apply** the Solidify modifier but keep the Remesh modifier.

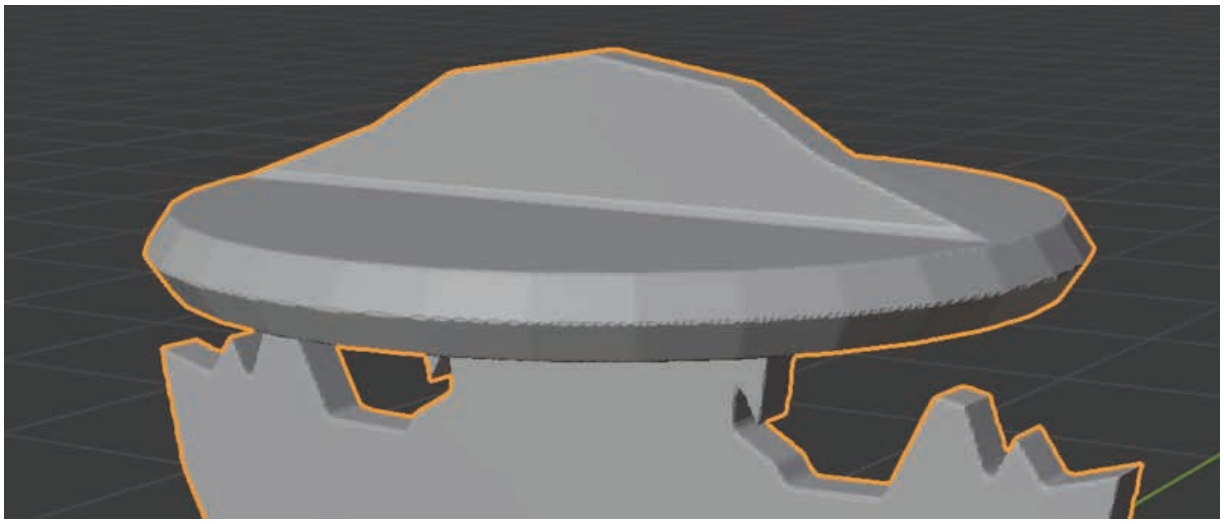


Okay, now for the fun stuff! From now on, when you **press Tab** to go into Edit Mode, you can add primitives with the **Shift+A** Add menu and add them to the mesh, and they'll be merged to it by the Remesh modifier. For example, let's add a **Cylinder**, scale it until it fits the head shape. Of course you can edit these primitives like any other mesh.

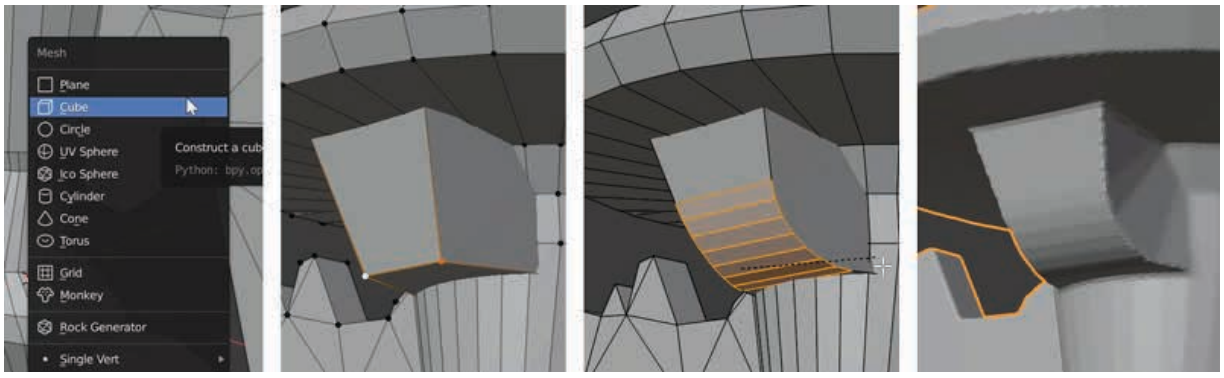




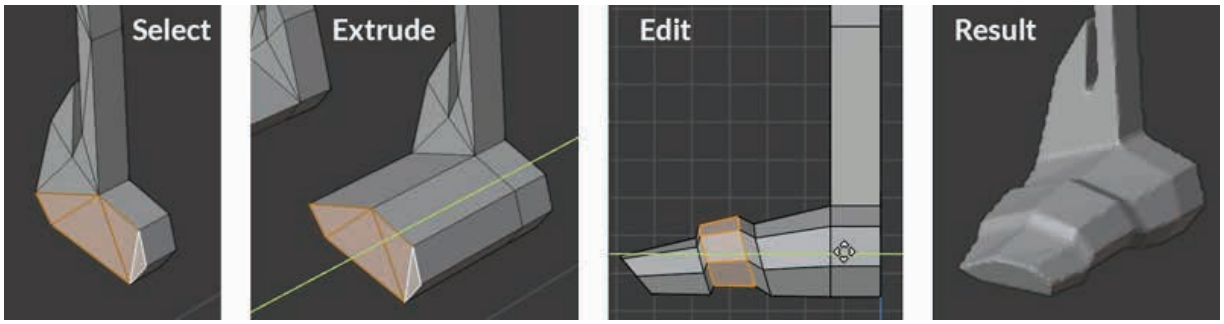
When you **press Tab again** to exit Edit Mode, you'll see that it's fused with the rest of the mesh.



The rest of the process is just more of the same: In Edit Mode, add more primitives and adjust them to match the shape of the outline. Of course, you can just continue using your imagination and adjusting the shape to your liking.



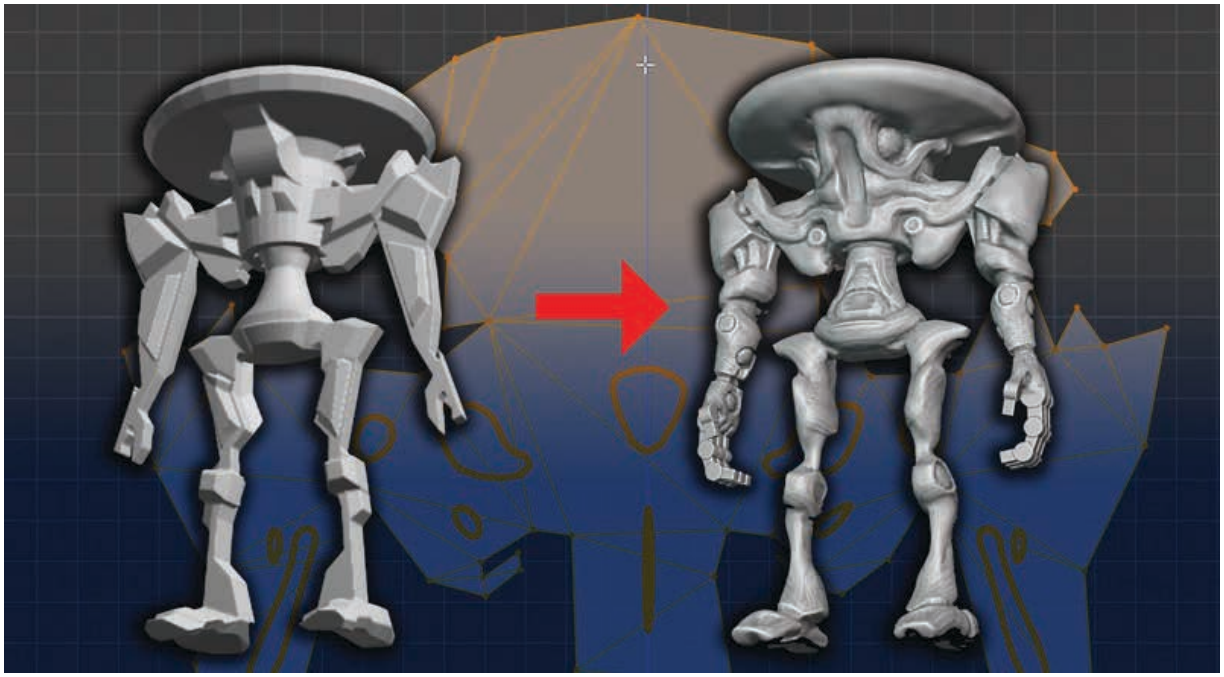
Of course, you can also select parts of the mesh traced from the drawing, and extrude those to create more geometry.



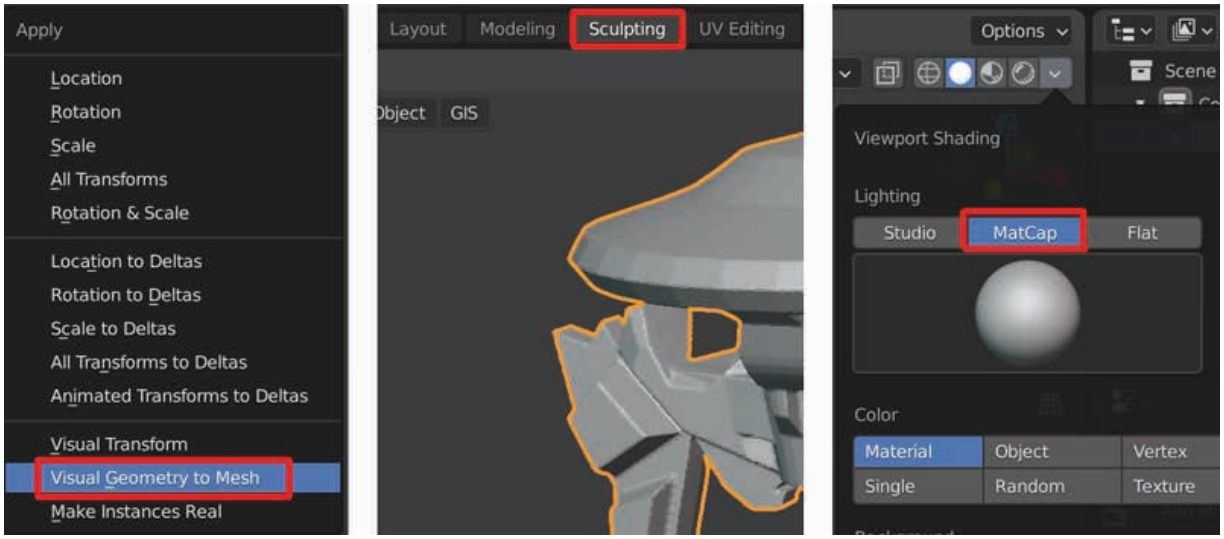
Don't worry about it being perfect: think of it as a starting point for the sculpting process, which we'll look at in the next chapter. Whatever you do is better than just sculpting from a Sphere!

Sculpting a robot concept

In this lesson, we'll look at refining the base mesh using some simple sculpting techniques. The goal is not to make a perfect 3D model, but rather to further think about the design. Sculpting is good for this because it's very quick and gives you a good 3D reference later for the "real" modeling stage. So don't worry about perfection, just let the creativity flow!

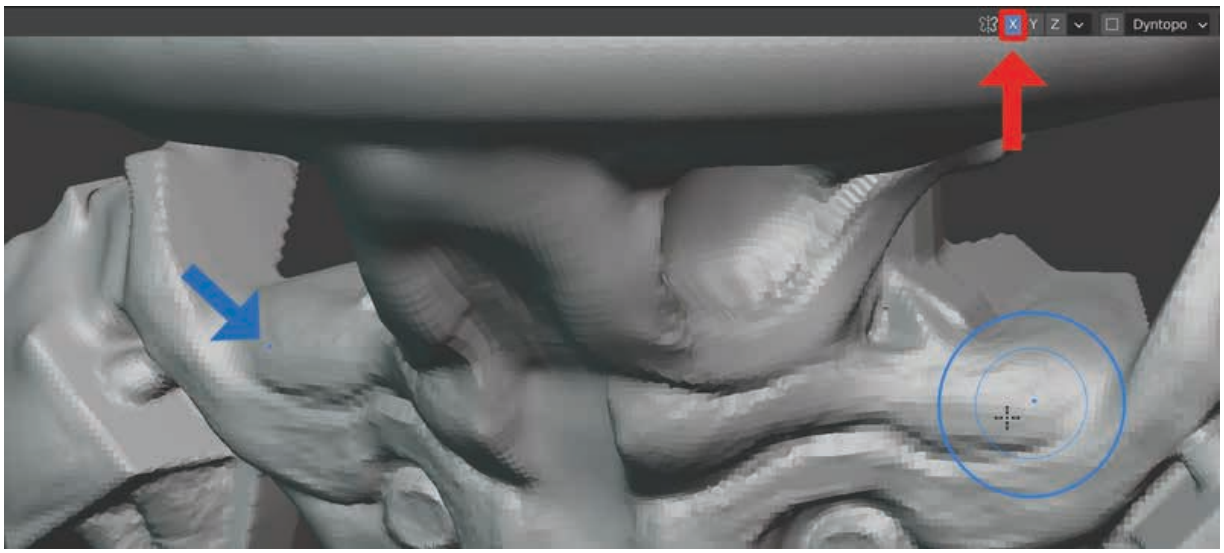


After you've prepared a base mesh as explained in the previous chapter, you can begin to refine the shape by sculpting on it. First, we can apply the modifiers by **pressing Ctrl+A** and choosing **Visual Geometry to Mesh**. Switch to the **Sculpting workspace** and use a **Matcap** so you can see better what you're doing.

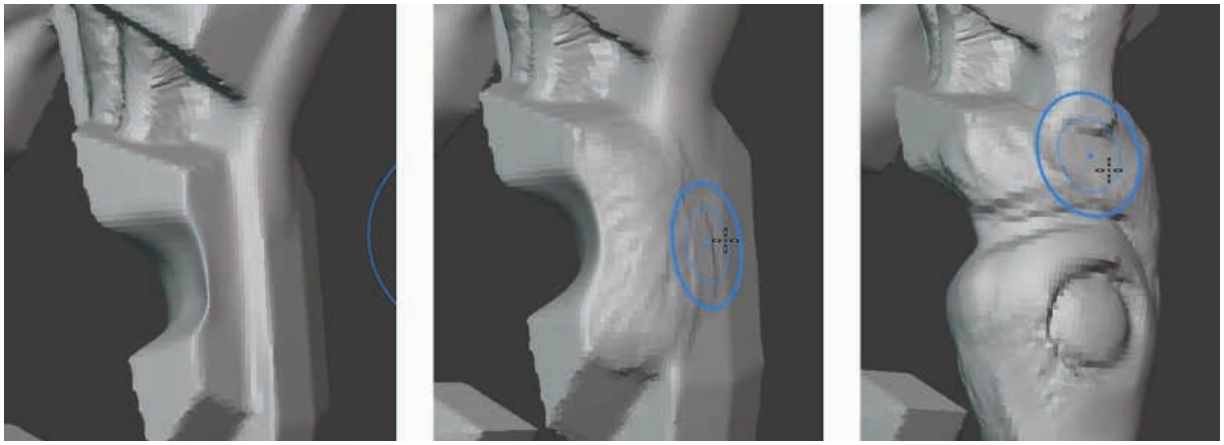


For some tips about using the Wacom pen in Blender, I made this video that you can check.

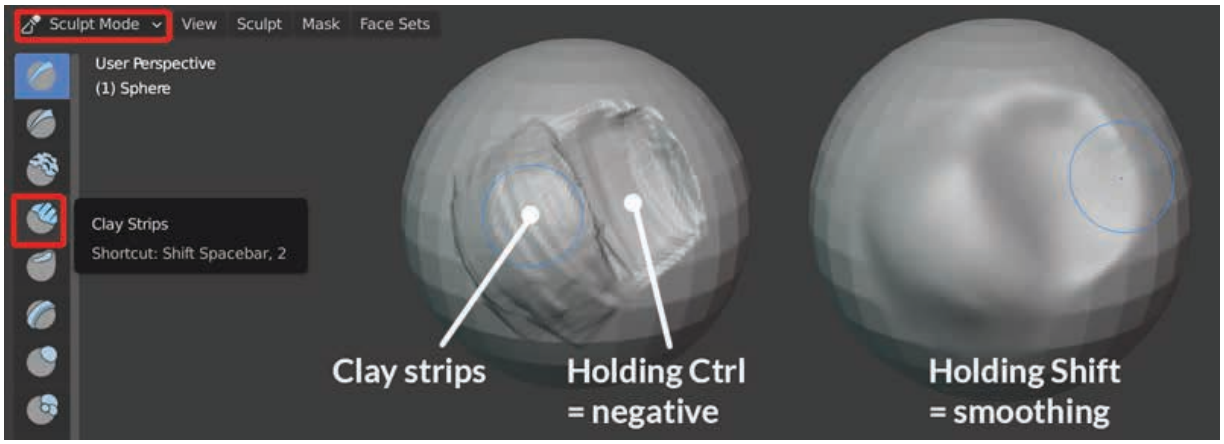
Turn on **X symmetry** so that whatever you do gets mirrored to the other side as well. You can see below that on the other side of the mesh, there's a small blue dot where the brush is being mirrored.



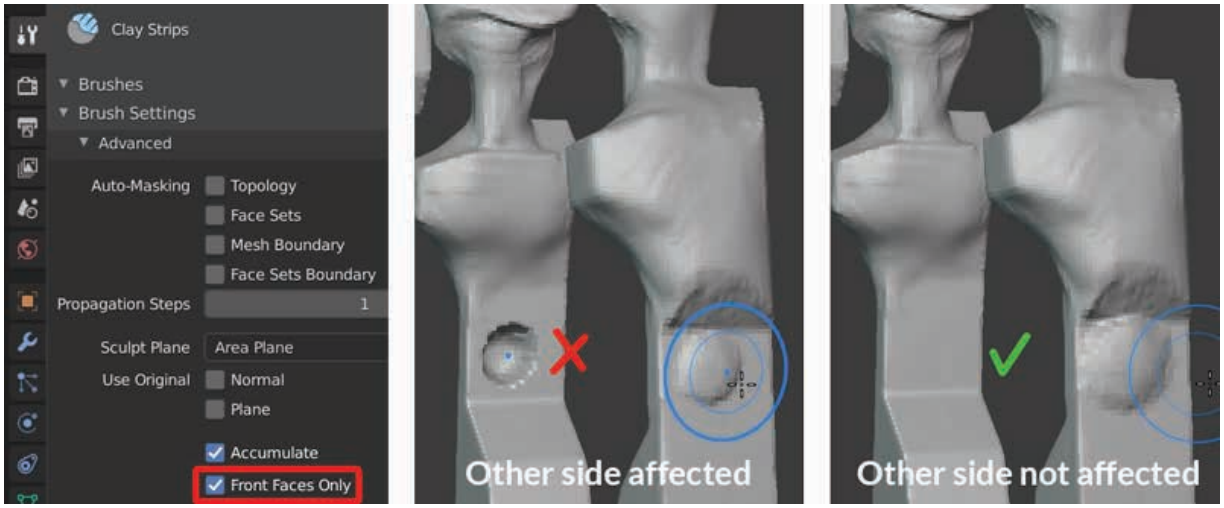
Blender has a lot of sculpting tools, but for this kind of base mesh sculpting, I tend to only use **Clay Strips**, which quickly puts layers of "clay" on your model, very useful to add volume.



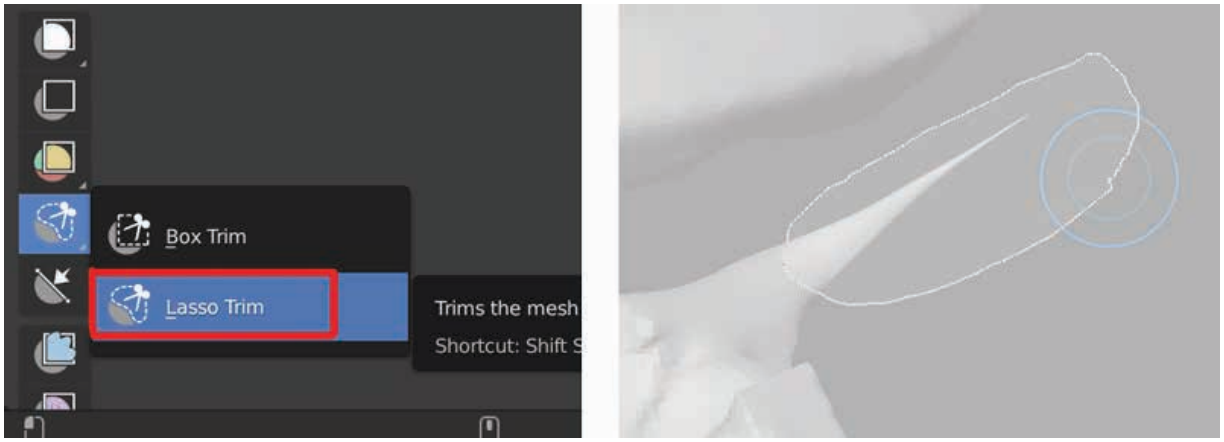
You can **invert** the Clay Strips by **holding Ctrl** and **hold Shift** to activate the **Smooth** brush. So really you only need the one brush!



Of course I'm not saying you can't use all those other brushes, just that this is enough for me for the purpose of furthering the robot concept. To make sure you don't destroy the other side of where you're sculpting, under **Brush Settings** in the **Advanced** options, check **Front Faces Only**. This way you can sculpt also on thinner parts, without the risk of destroying the other side.

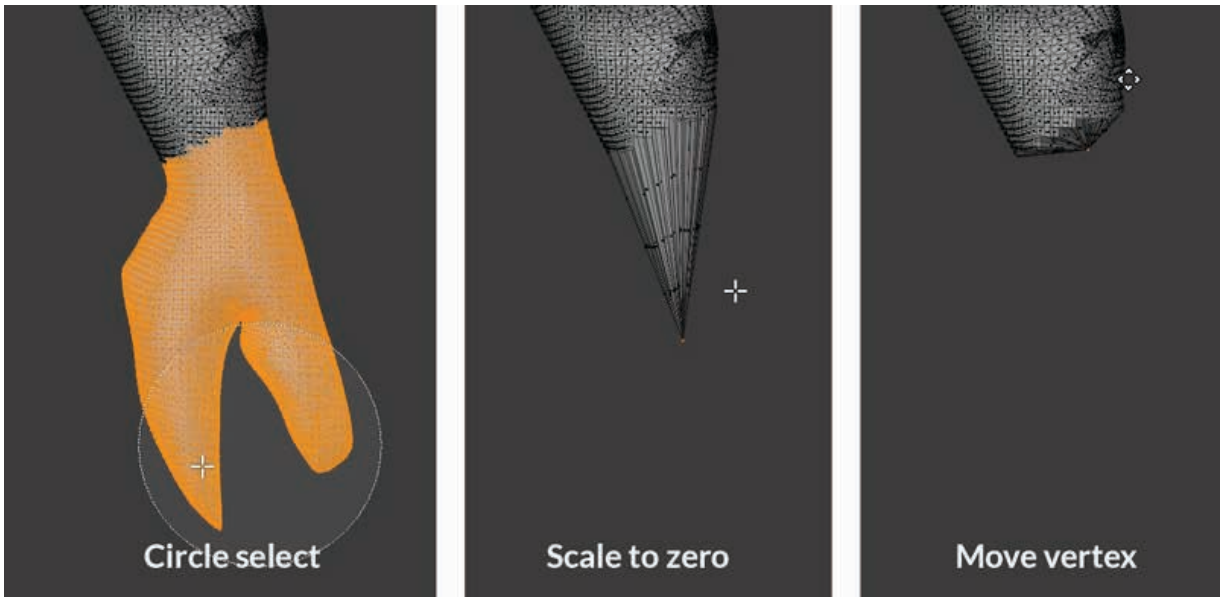


In some cases you may want to trim some parts of your geometry, in that case you could use the **Trim Lasso** tool. However in my experience this is still a rather slow tool, and often will freeze Blender for a long time. If you want to try it, save your project first.

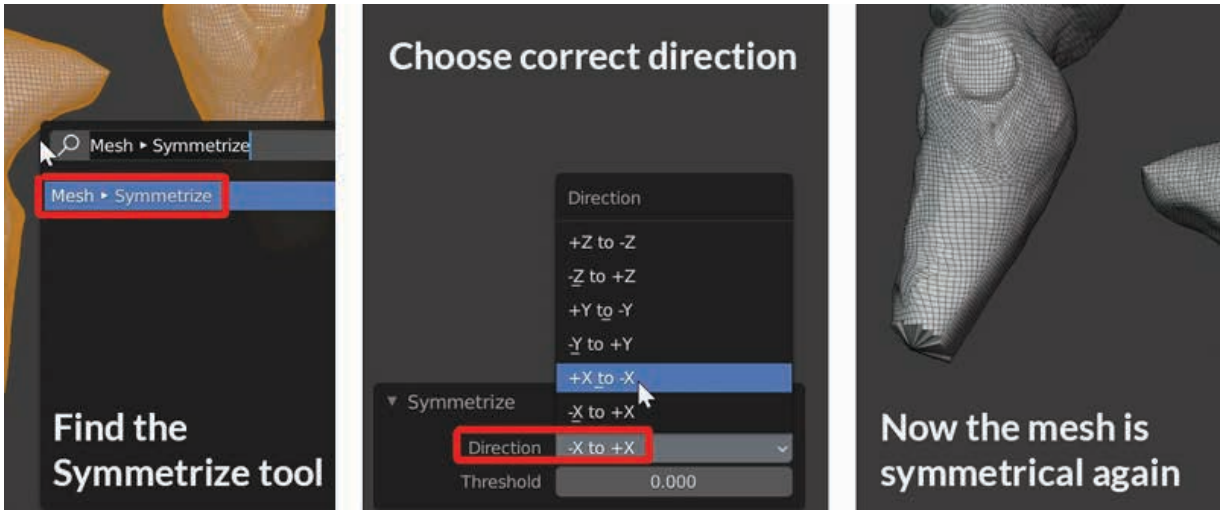


Alternatively, you can switch to **Edit Mode**, Select the faces you want to remove with Circle select by **pressing C** and scaling all the faces to a single point by **pressing S+0+[enter]**.

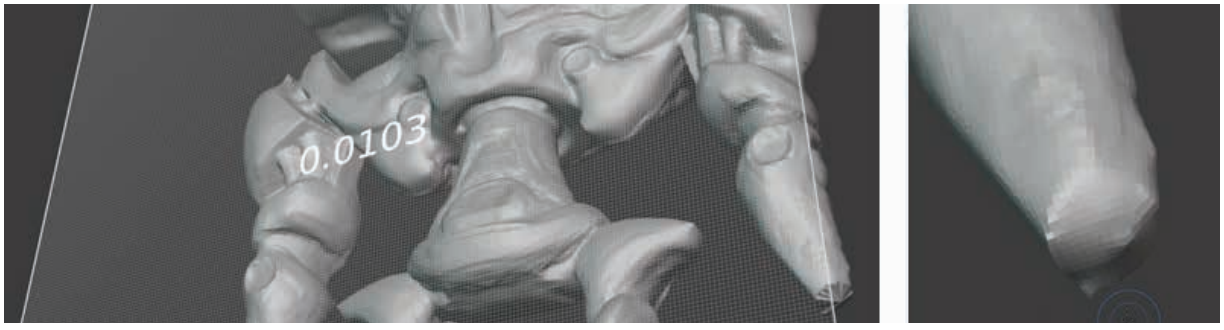
Then you can move the remaining vertex by **pressing G**. I hope the Trim lasso tool will be a bit faster in the future, as it would be more convenient, but at least for now this alternative method works fine.



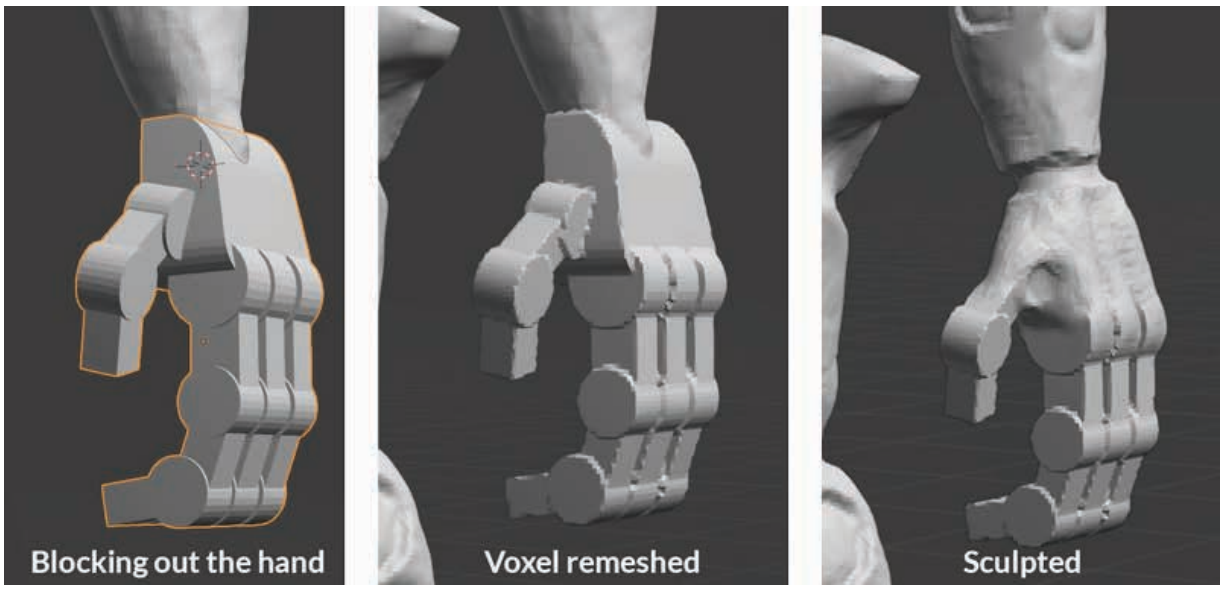
After trimming the parts you wanted to get rid of, select all and **press F3**, then search for **Symmetrize**. You may have to change the direction to "+X to -X", depending on which side of the model you worked on. Now the model should be symmetrical again.



Back in **Sculpt mode**, you can remesh things and smooth out the part where you collapsed the faces. To get a preview of the resolution you want for this, **press Shift+R** and move the mouse - you can see a preview of the size of the voxels. Then **press Ctrl+R** to remesh. Then you can smooth out the trimmed parts.



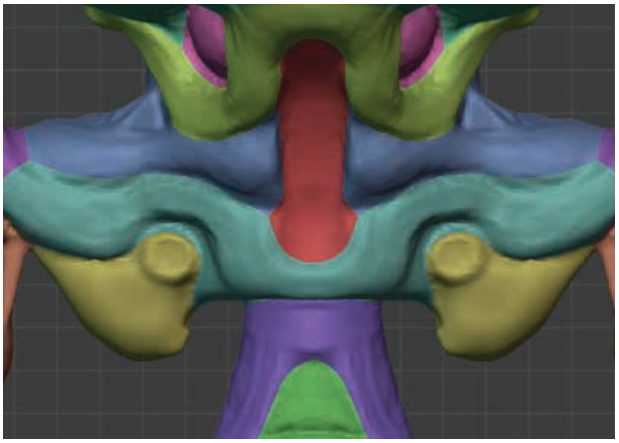
After adding some more geometry for the hands we can merge it with the basemesh by Voxel remeshing once more.



I consider this an essential step in further thinking about the design, kind of like sketching in 3D rather than modeling. It helps tremendously with moving on to the next step, which is the actual modeling of the robot.

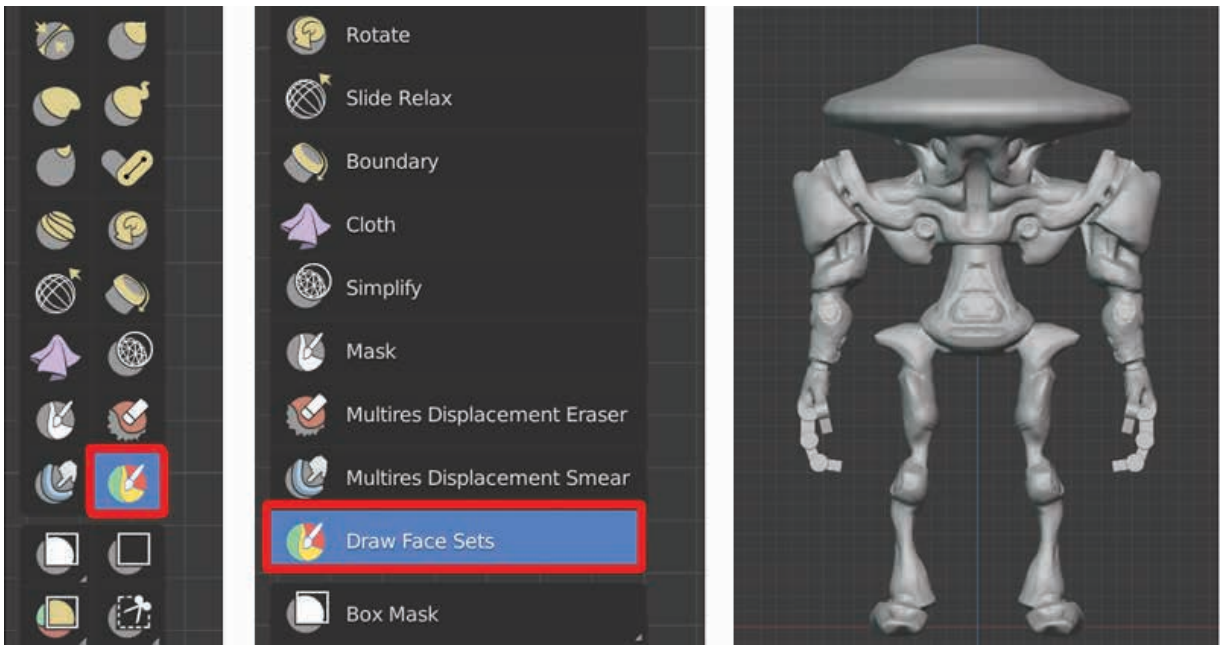
Face Sets

To continue with our robot model it's a good idea to split it into separate parts, so we can focus on one part at a time when we refine the sculpting.

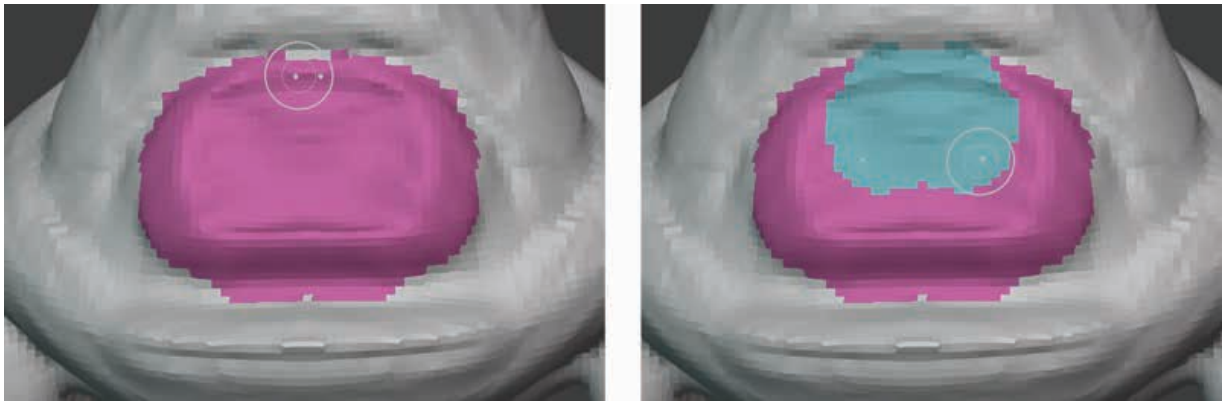


The big benefit of this, is that you'l notice much better and smoother performance when sculpting on a small item with everything else hidden, than if everything was visible.

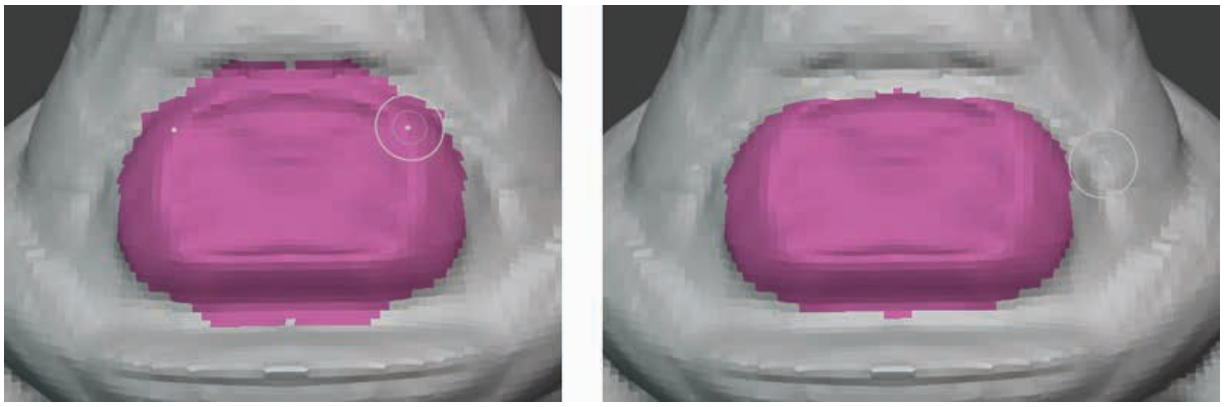
A convenient way split up the model is with **Face Sets**, which is a tool available in **Sculpting mode**. If you don't know what the brush icons in Blender are, you can drag the tool panel to the right to reveal the names of the brushes.



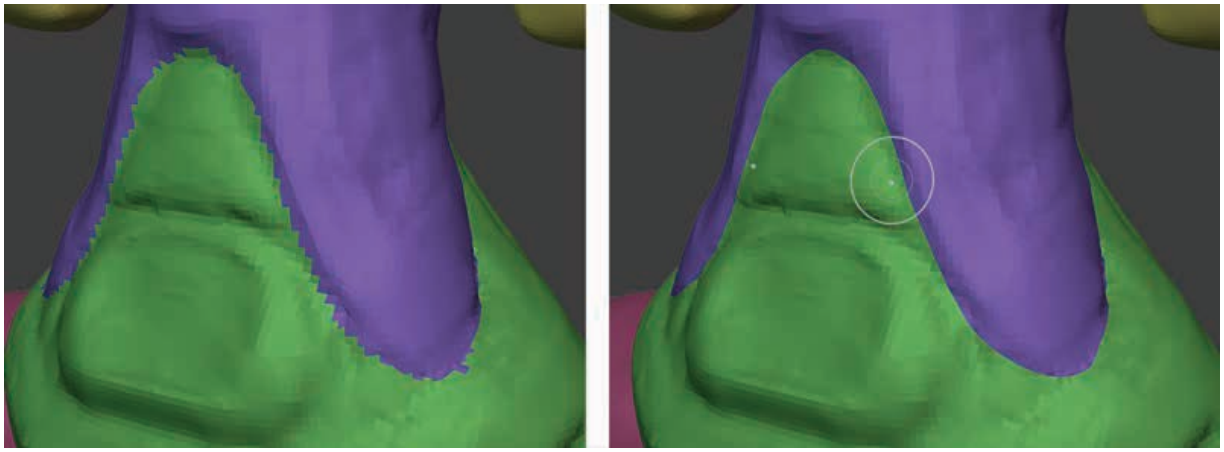
This way it's easy to find and activate the **Face Sets brush**. Now you can just start painting a Face Set. If you stop painting and start again, it will create a new Face Set.



If you don't want that, **hold Ctrl** and start painting from the Face Set that you want to continue painting. **To erase, hold Ctrl outside the Face Set** and paint from there. The grey part where you didn't paint a Face Set yet, is in fact a Face Set as well.

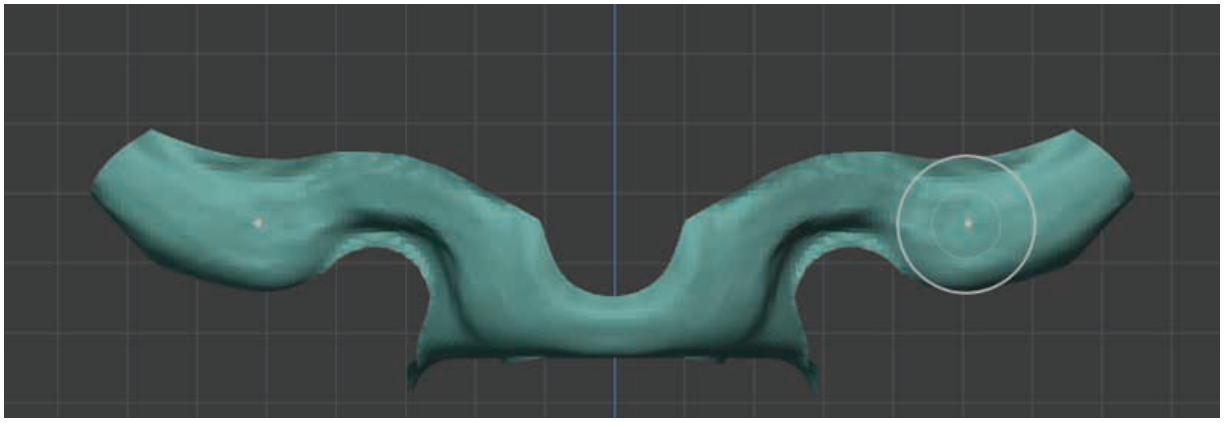


To make the borders of the Face Sets smoother, **hold Shift** and smooth them out. Note that this actually smooths out the edge flow of the geometry underneath, which may not be what you want.

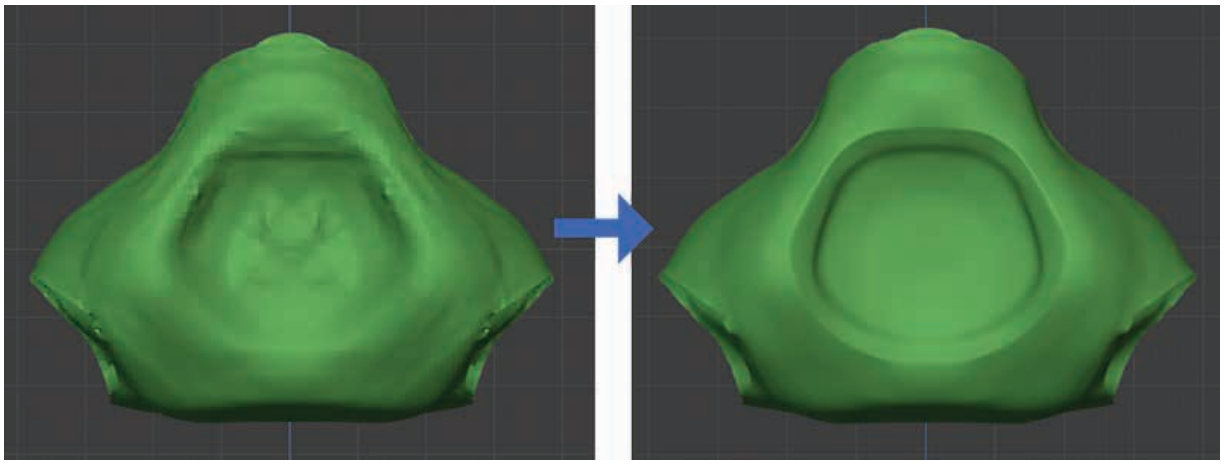


You can also do this for all Face Sets at once after painting all the Face Sets, by choosing the **Mesh Filter** and setting it to "**Relax Face Sets**". Then drag the mouse cursor while holding the left mouse button and it will smooth all the Face Sets.

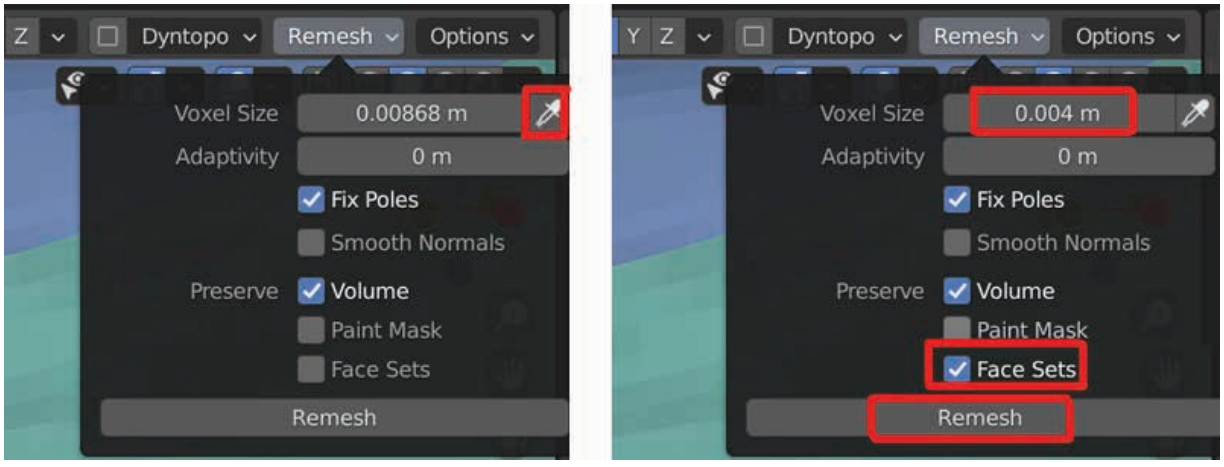
Now hovering the mouse cursor over a Face Set and **pressing H** toggles hiding/unhiding everything but that Face Set.



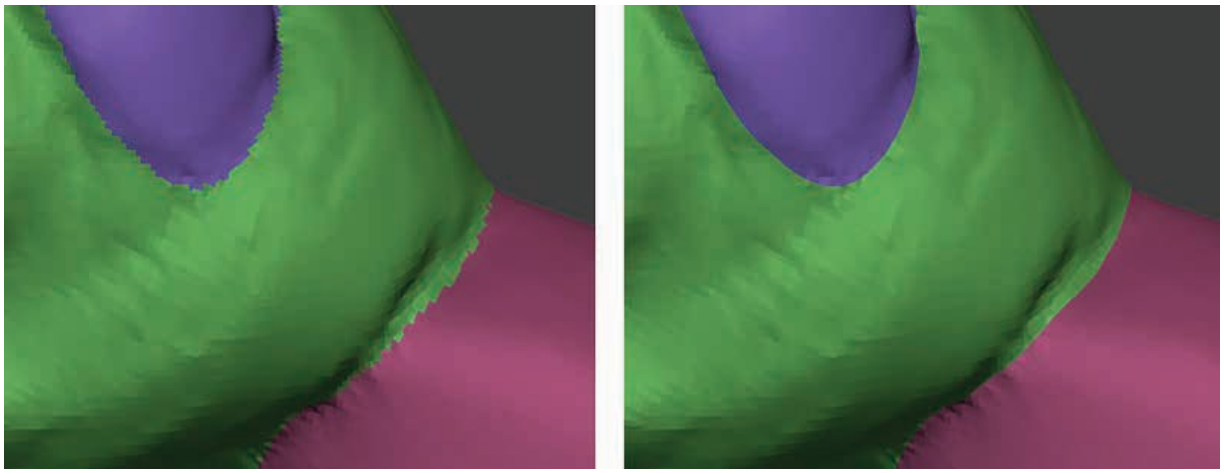
So after this, you can focus on specific parts to refine with either sculpting or manual retopology.



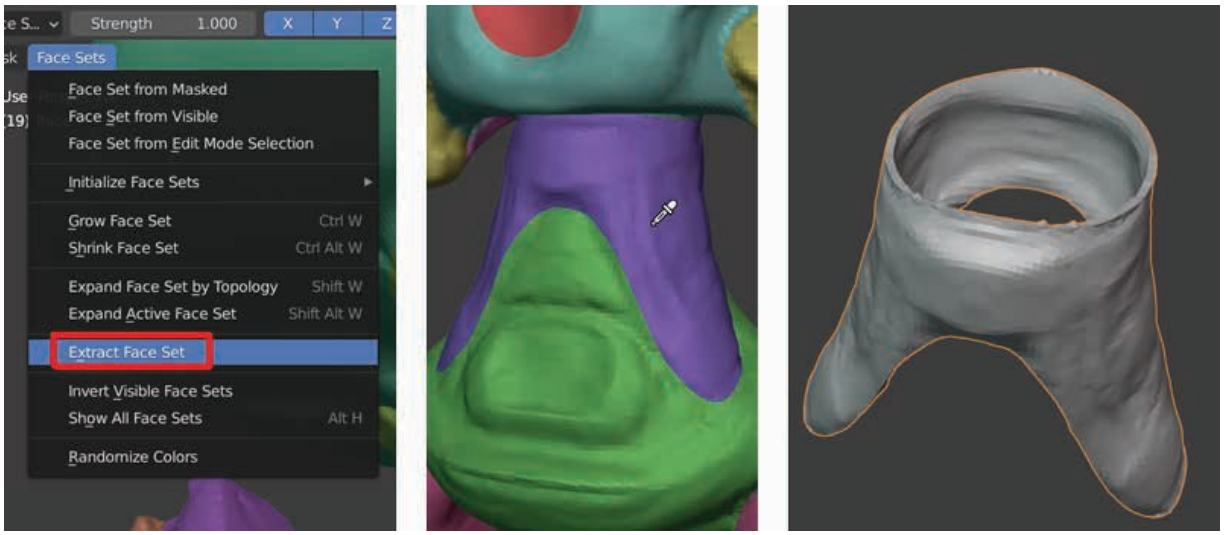
If you want to remesh in order to add more geometry needed for finer sculpting, sample the current voxel size so that you can use a smaller value for the remeshing. Then check "**Preserve Face Sets**" otherwise the remeshing process will remove all your Face Sets. Then **click Remesh.**



After that you'll need to use the Mesh Filter with "Relax Face Sets" again to smooth the Face Sets.

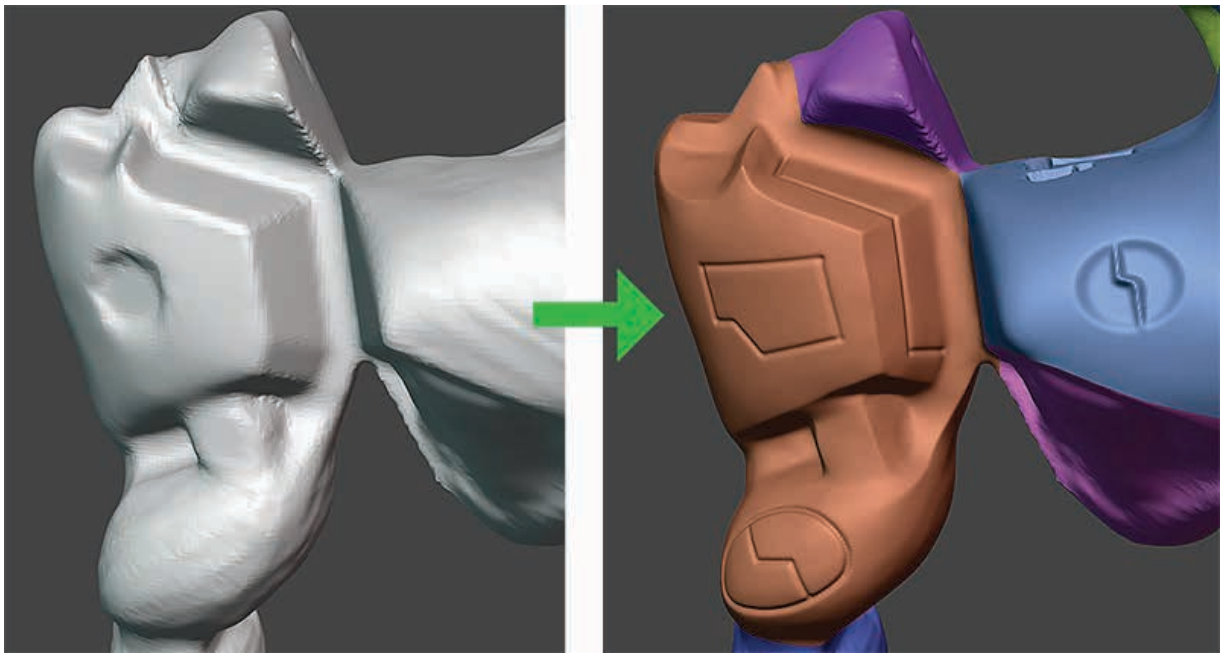


You can also go to **Face Sets > Extract Face Set** and then pick one, which will extract a separate mesh.



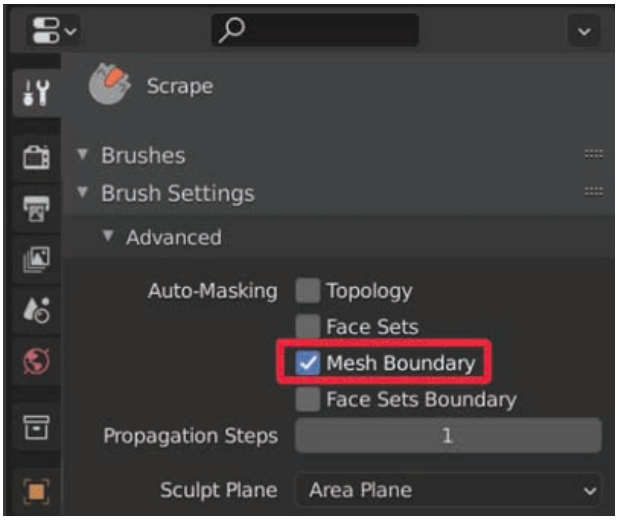
Hard Surface Sculpting: Refining the rough sculpt

Now that we've divided the rough sculpt into several parts we can improve them one by one. Let's look at some other useful sculpt brushes and tools.

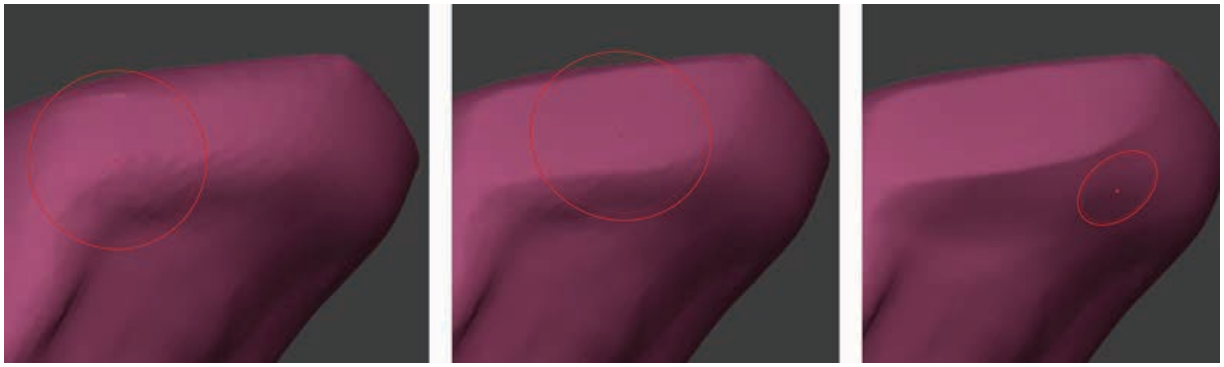


Hover the mouse cursor over the Face Set you want to work on, and **press H to isolate** it. Select the **Scrape Brush**.

With meshes that have an open hole, if you want to protect the edges of the hole turn on **Mesh Boundary** under the **Auto Masking options**. That way when you're sculpting close to the edge of the hole, the edges won't be affected by the brush. Keep in mind, this settings needs to be checked for each brush that you use.

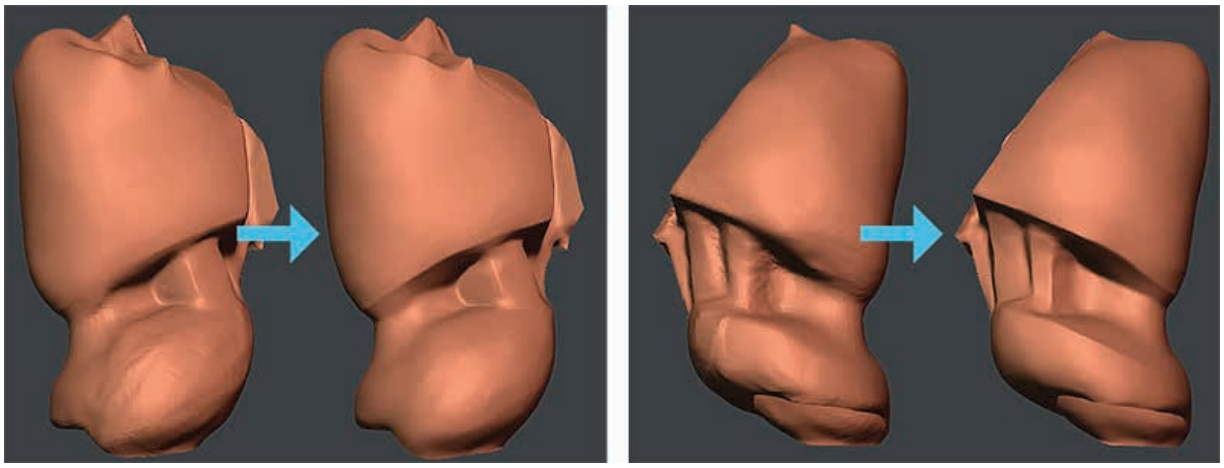


The **Scrape brush** works well to make areas more flat. A good workflow with the Scrape brush is to find an area where you would like to have a sharper edge, and then make it more defined by flattening the areas on both side of where the edge should be.

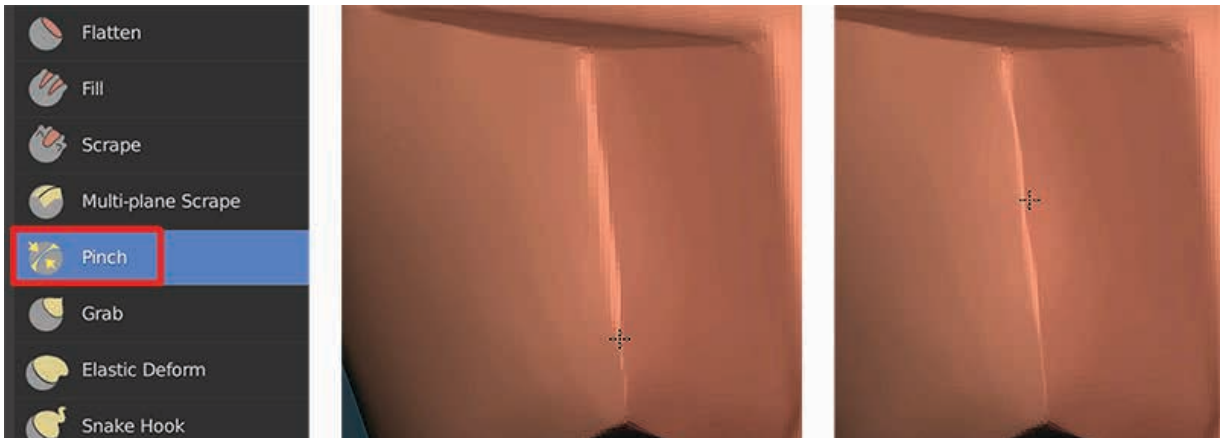


However it's also a great brush for making round surfaces more smooth. By using a **Matcap** and rotating the view you can see surface imperfections more clearly. **Hold Shift** and go over some areas to smooth them out further. **Press F** to adjust the **Radius** of your brush when necessary.

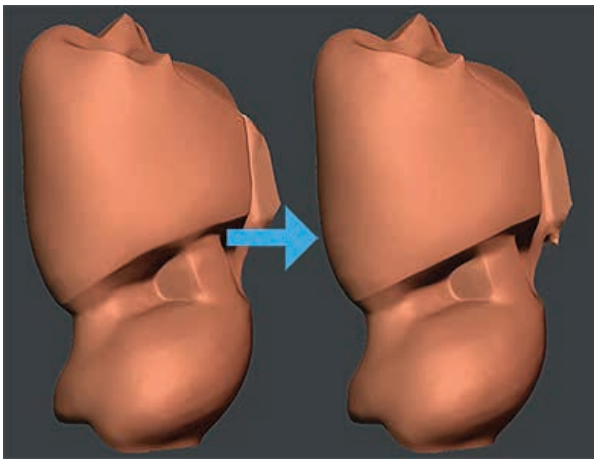
I find that just using the Scrape brush can get you quite far in getting a hard surface look. Below you can see the difference after using the Scrape brush for a while.



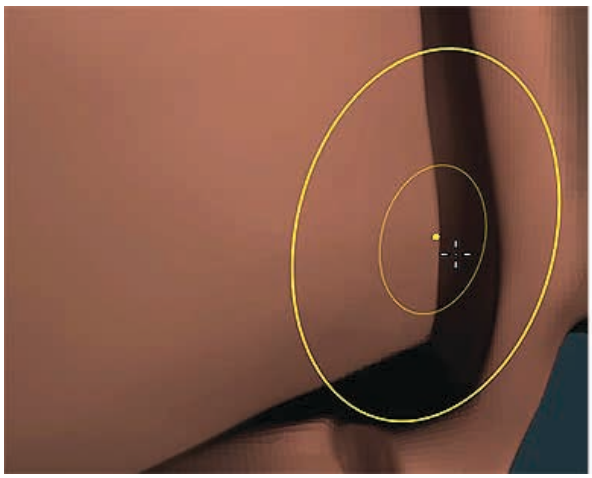
The **Pinch brush** then can create some nice sharp edges. You can experiment with the **Strength** value and the **Falloff** shape, if you like. I find the defaults work well, but you can set the Strength higher and use a sharper Falloff if you like.

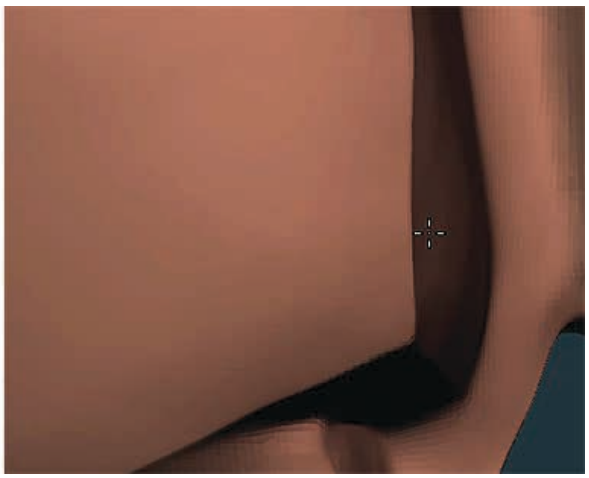


Here is the result after using the Pinch brush on some of the edges, making them a bit sharper and more straight.

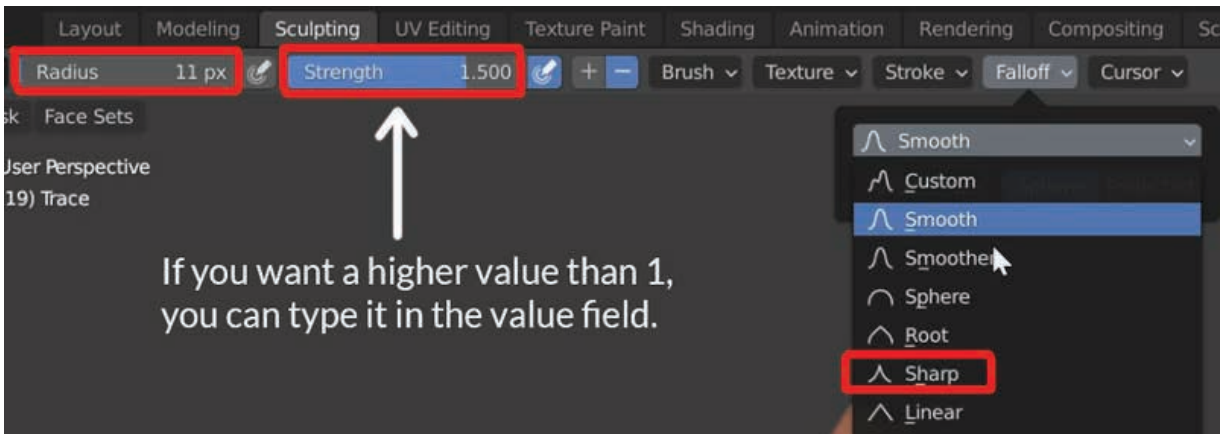


Using the **Grab brush**, you can pull on the mesh to quickly correct the shape where needed.

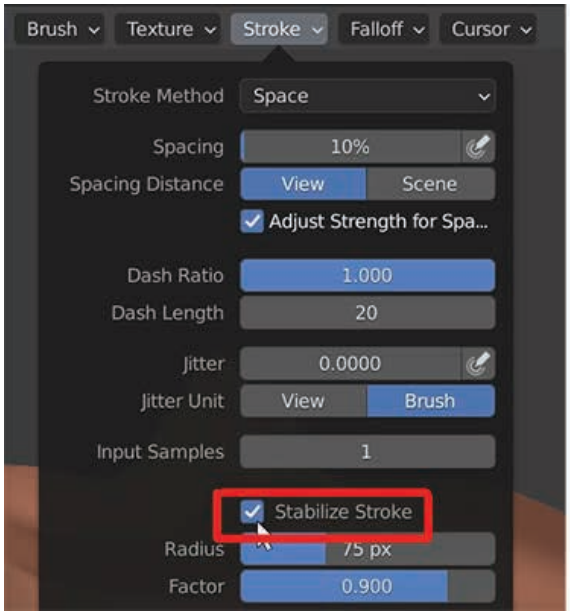


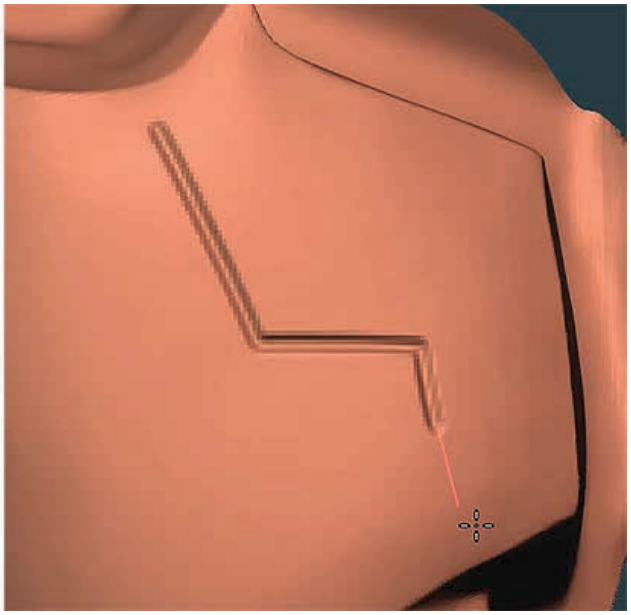


To make panel cuts, you can use the **Crease Brush**, with a small radius. This works especially well if you increase the **Strength** (to 1 or even 1,5) and set **Falloff to Sharp**.

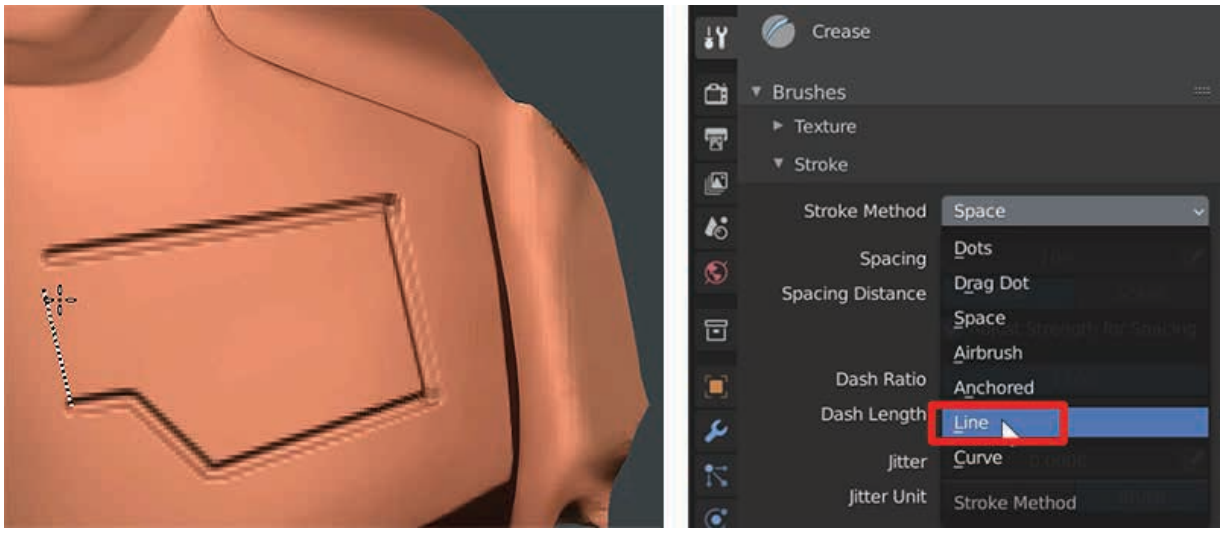


Now you can draw lines, but they will probably not be very straight. If you turn on **Stabilize Stroke**, you can draw more stable lines.





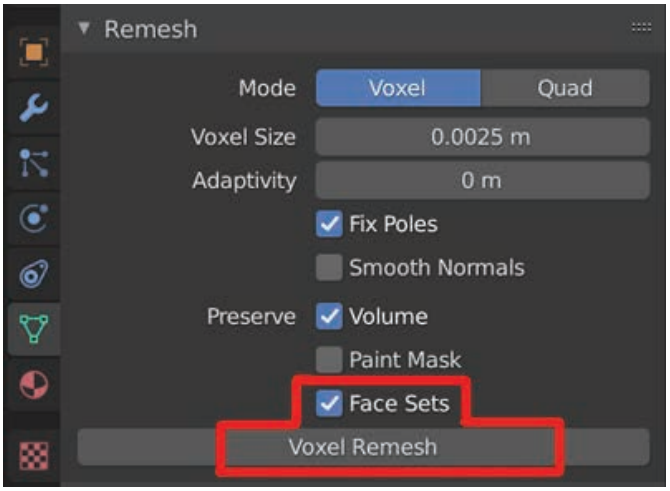
You can also set the **Stroke Method** to **Line**, to get really straight lines.



Of course, if the resolution of the mesh is too low, these panel cuts won't look so good.

So you may have to **Voxel Remesh** the entire model to get some higher resolution.

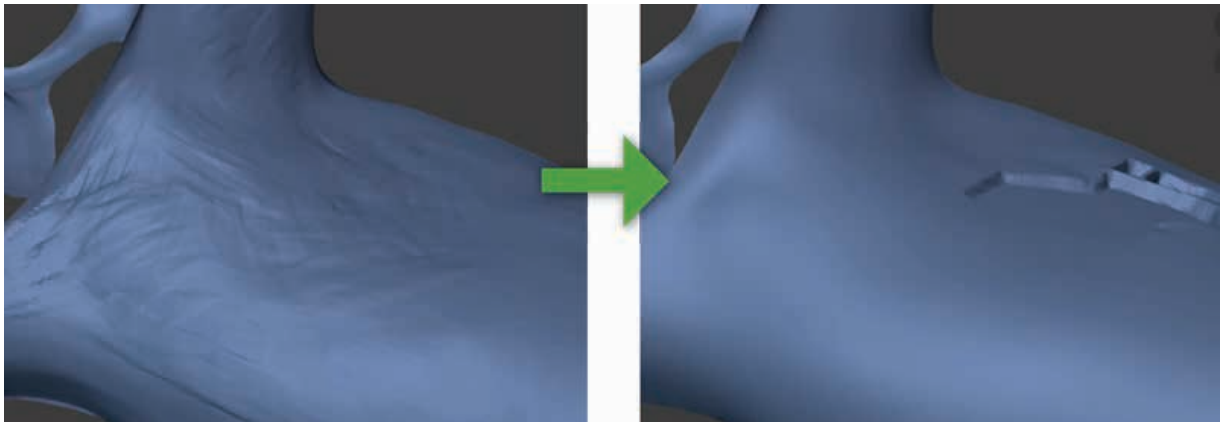
Be sure to check "**Face Set**" to preserve the Face Sets.



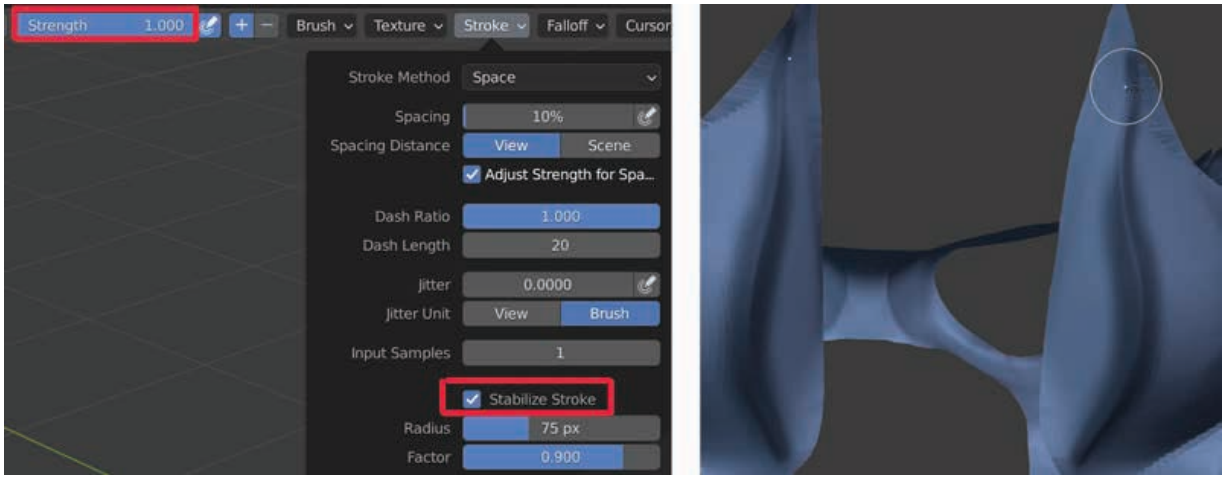
If you find that your computer can not keep up with such a dense mesh, than just skip those very fine details such as the panel lines. After all, we're still in the concepting stage - we can add such fine detail easily in the retopology stage, using height maps or normal maps.

Hard Surface Sculpting: Masking and Mesh Filter

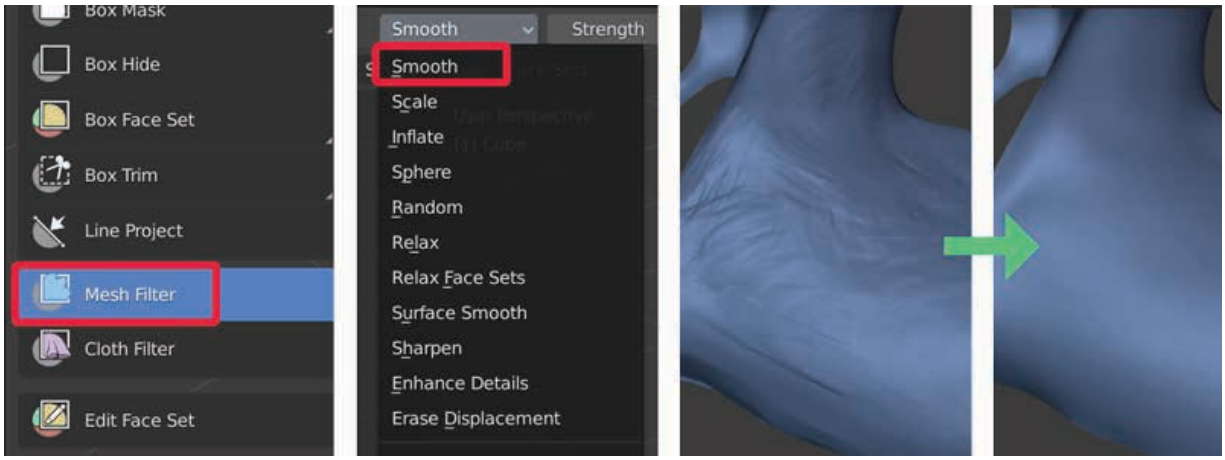
If you don't want to manually go over the entire mesh, smoothing it out bit by bit, you can take advantage of **Masking** and the **Mesh Filter** to smooth the mesh in a smart way.



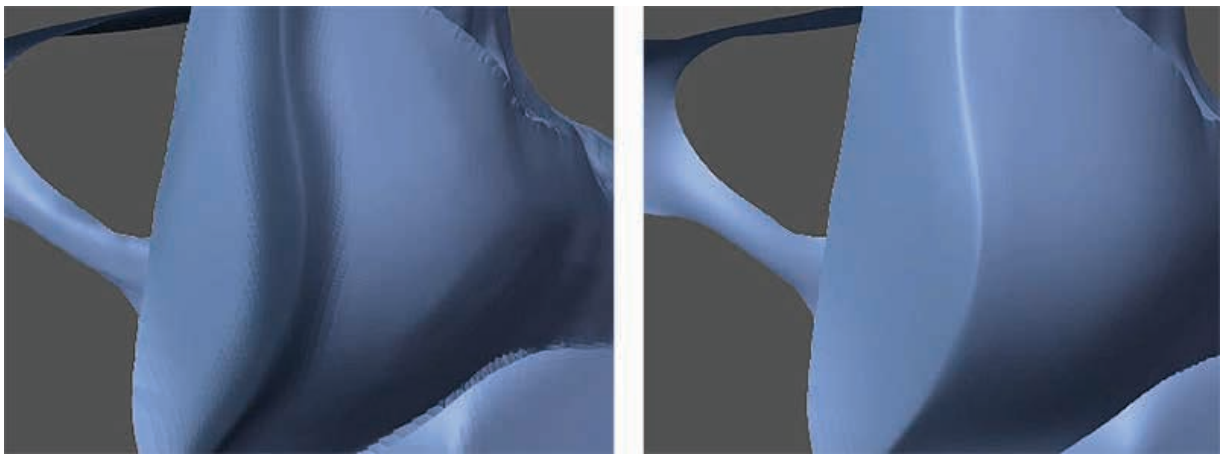
First, mask the areas that you don't want to smooth. You can toggle the mask brush on and off by **pressing M**. To get nice clean masks, set **Strength to 1** and turn on **Stabilize Stroke**. Specifically, you want to mask sharp edges and any other details you don't want to lose.



Then activate the **Mesh Filter** and set it to **Smooth**. Now when you drag the mouse to the right, while holding the left mouse button, it smooths out the unmasked areas.

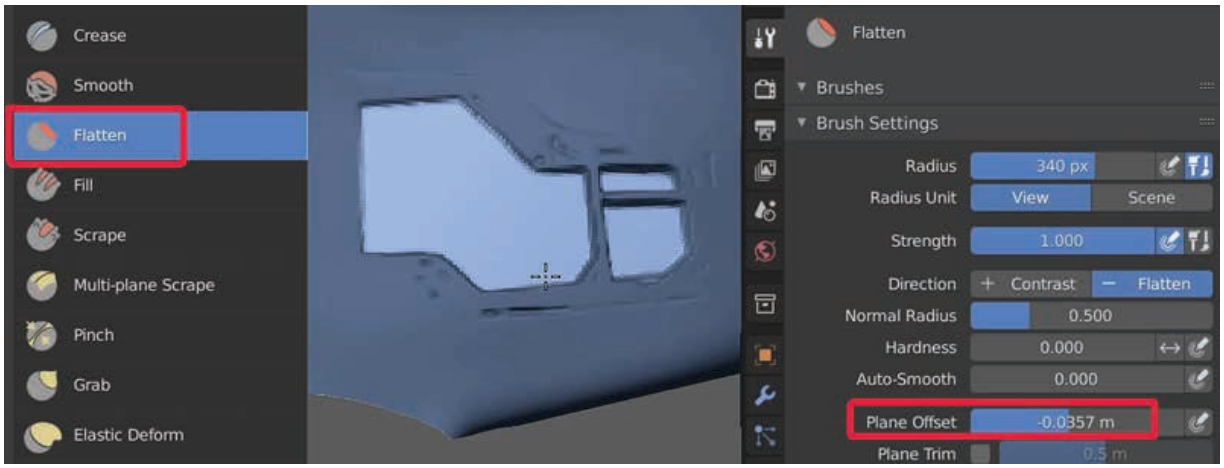


Smooth as much as you want (you can drag the mouse to the right multiple times to add more smoothing), then **press Alt+M** to remove the mask. As you can see, the sharp edges which were masked were not affected by the smoothing.



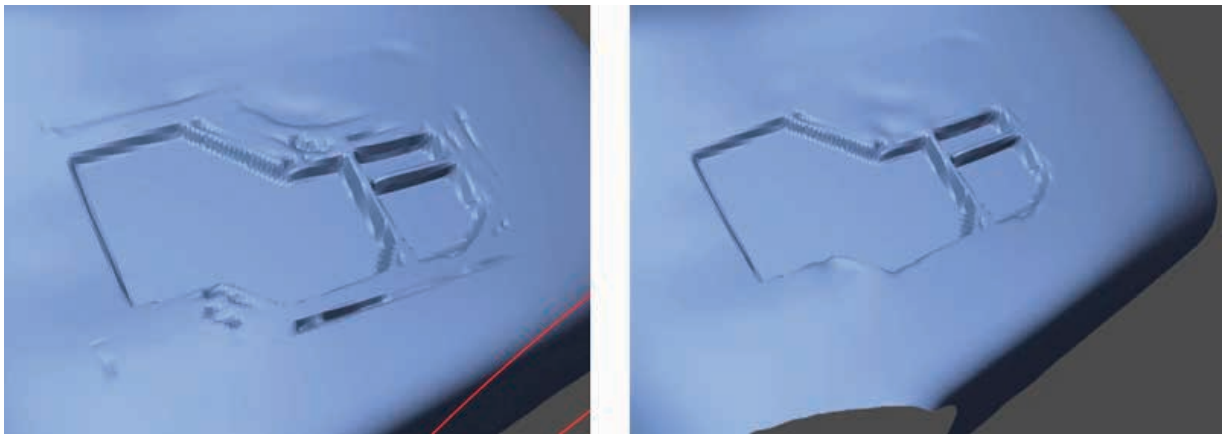
You can also draw some shapes with the mask. **Hold Ctrl** when masking to erase. Then **press Ctrl+i** to invert the mask and use the **Flatten brush** to press down that area.

For this it's best to **set Strength to 1** and make the radius of the brush bigger. If it doesn't get pressed in deep enough, you can adjust the **Plane Offset** value.



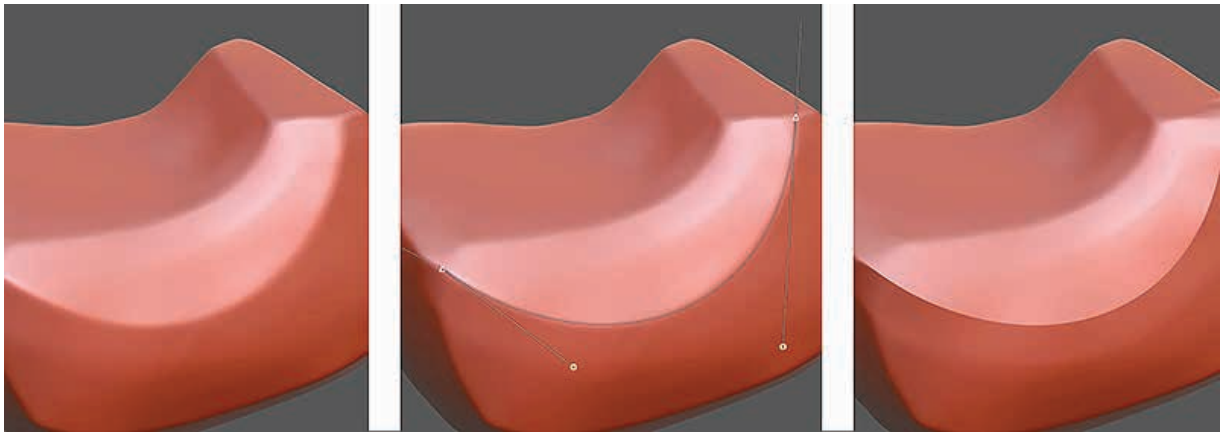
Clear the mask by **pressing Alt+M**.

As you can see it's a bit messy, because when I erased the mask I didn't completely erase it in a very clean way. You can smooth it out a bit more by holding Shift. However, this is all still the "concepting" stage, so it's not a big deal. Think of it as sketching in 3D. Later we will retopologize with a clean mesh.

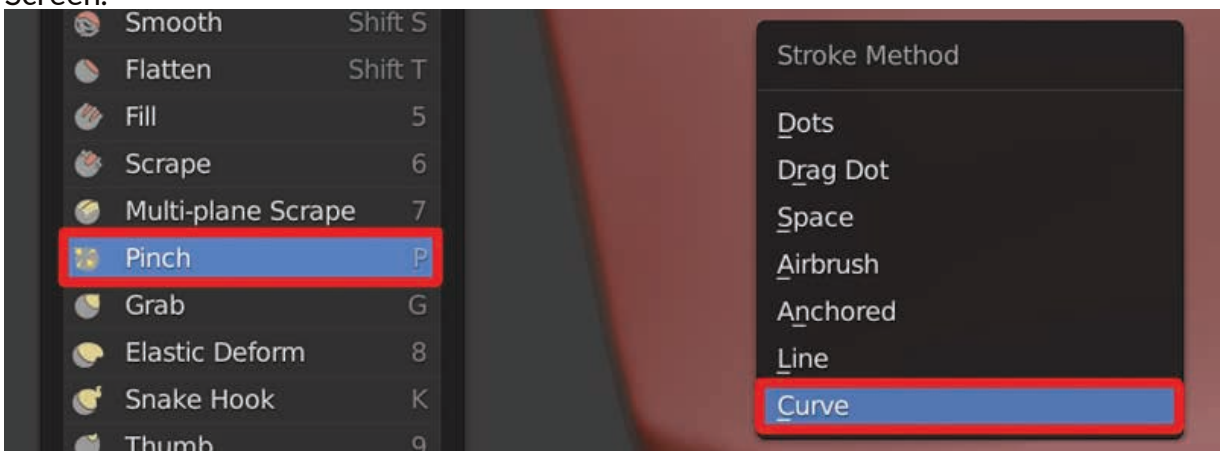


Hard Surface Sculpting: Using Curves

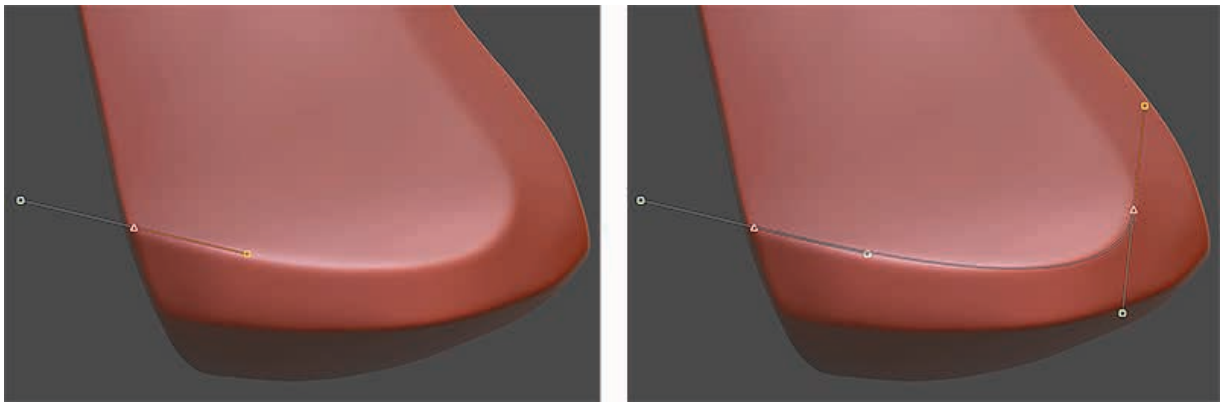
When sculpting, you can use the **Pinch brush** to make nice sharp edges. But it can be tricky to make them perfect, the way machined parts would look. Here you can use **Curves** to your advantage.



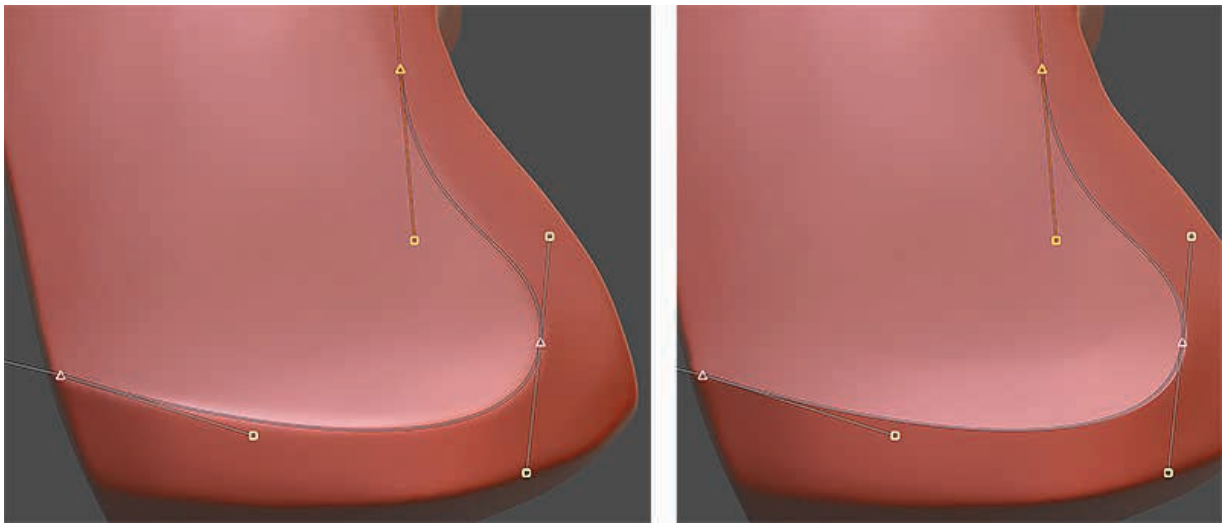
First, select the **Pinch brush**. Then **press Alt+E** to open the **Stroke Method** menu and choose **Curve**. By the way you can **press Shift+Spacebar** to quickly get all the brushes, and then **press P** for example for the Pinch brush. Useful when you're working in Full Screen.



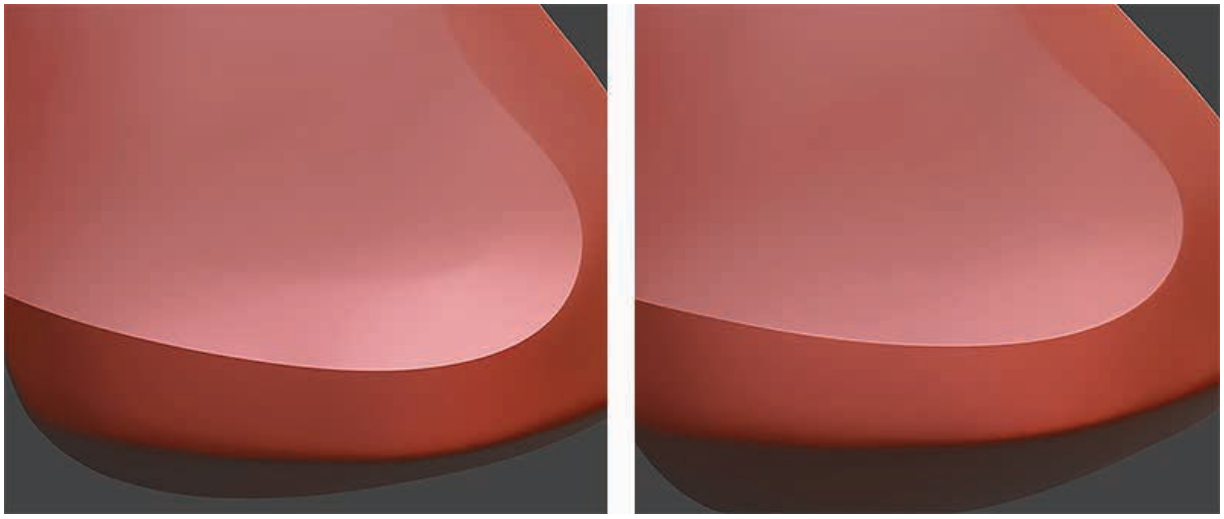
Then **hold Ctrl, right-click and drag** to create the first handle. **Right-click and drag again** to create the second handle.



Now you can fine-tune the curve by manipulating the handles, or add more handles to the curve. When it looks good, **press Enter** to pinch that area.

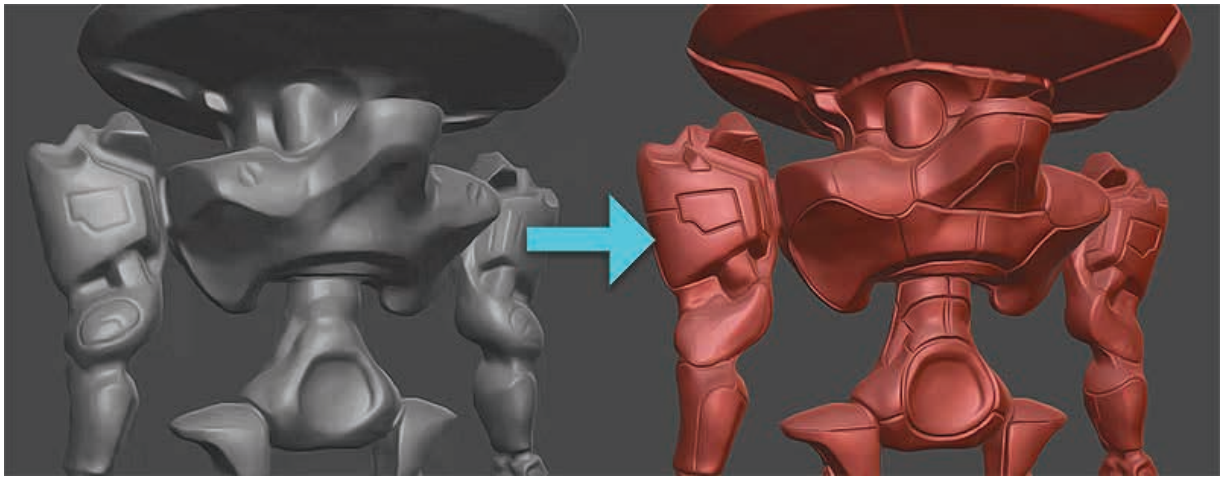


You can **press Enter again** as many times as needed to pinch more. After that, **press A** to select the entire Curve and then **press X** to delete it. You may have to press A twice to select all. Use the **Smooth** and **Scrape** brush if necessary to make the area next to the pinched edge look better.

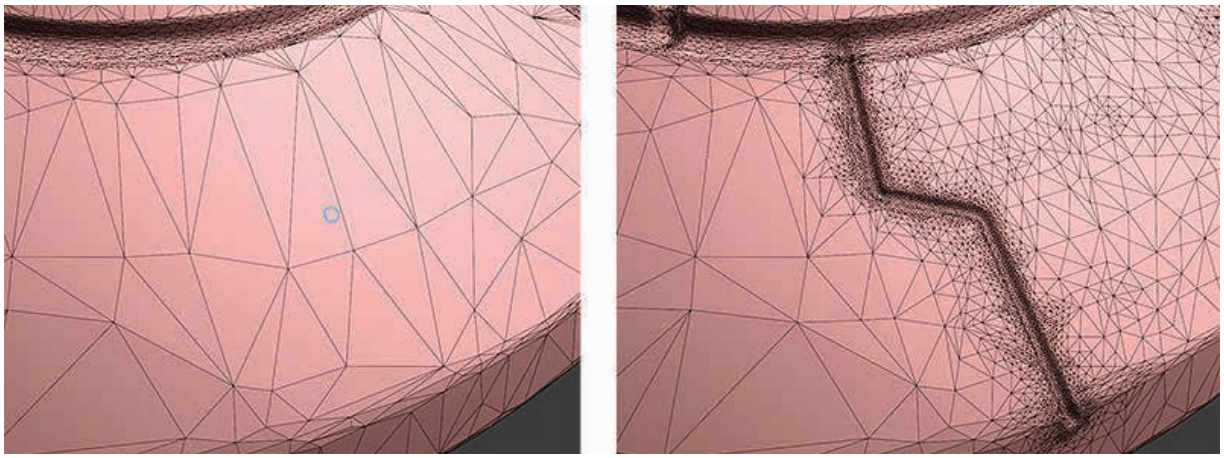


Hard Surface Sculpting: Dyntopo

Another interesting workflow for sculpting is using **Dyntopo**. I will use this to draw some panel lines on the model.



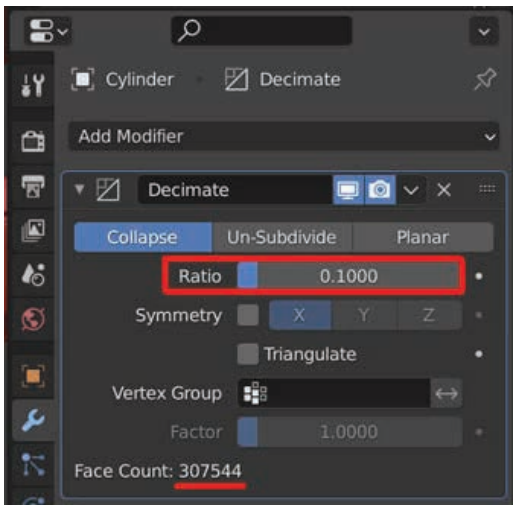
Dyntopo only adds geometry where we place the brush strokes. This way we can get fine creases even when the mesh doesn't have enough geometry. For example, below I use the Crease Brush to draw some panel lines. Notice how extra geometry is added to support the crease.



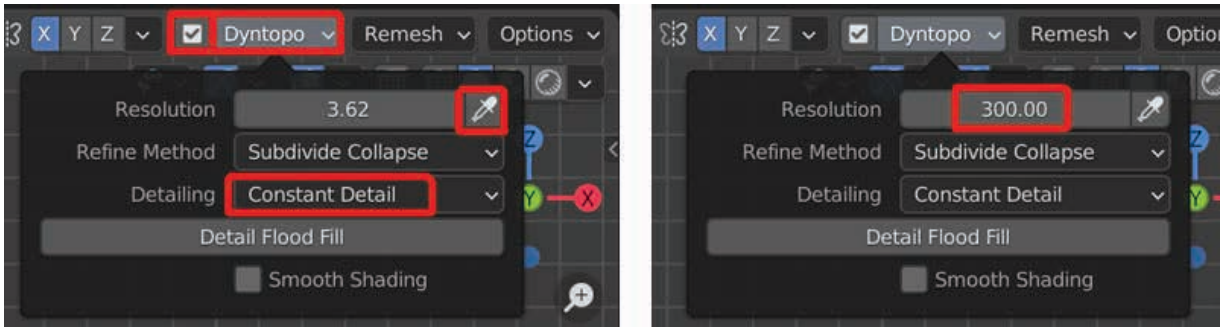
But first, to make life easier: since the robot model was getting very heavy at this point with about 3 million triangles, I added a **Decimate modifier** set to 0,1 (in other words, decimate to 10% of the original Face Count).

I applied the Decimate modifier by **pressing Ctrl+A** while hovering over the modifier to get it back to about 300.000 faces.

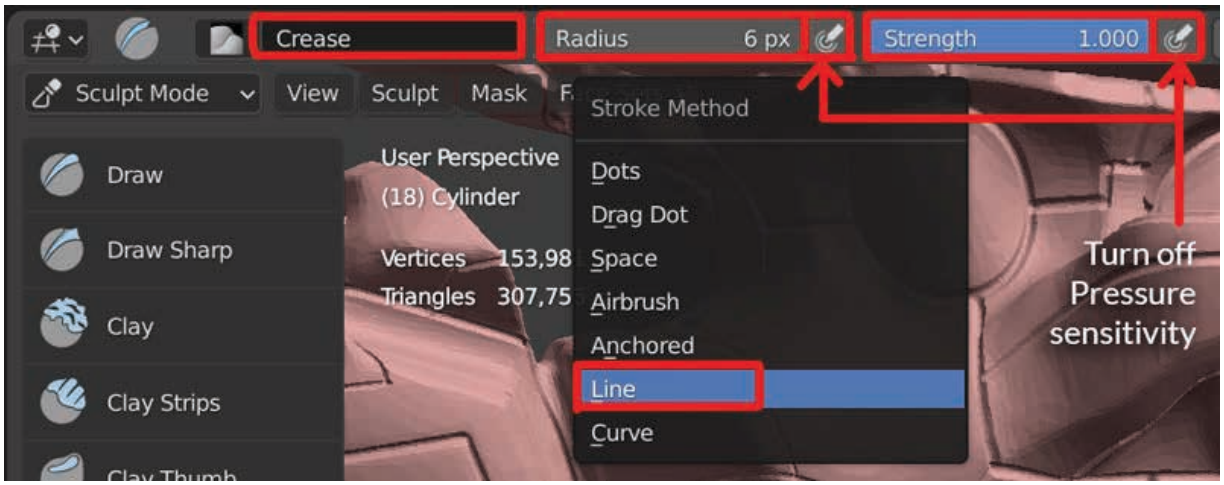
Now Blender is much more responsive again.



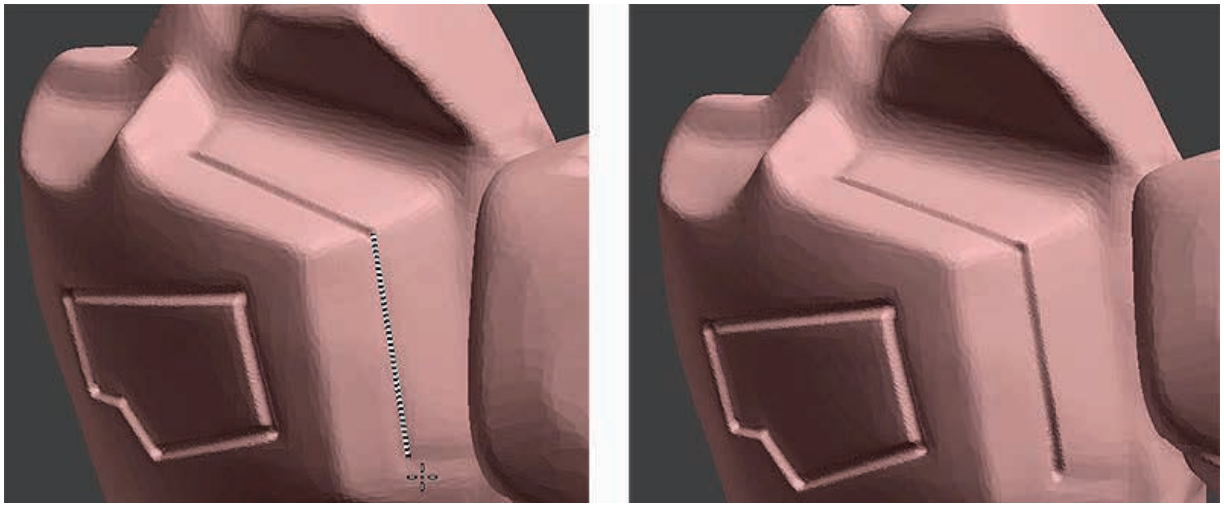
Enable Dyntopo and set it to **Constant Detail**. To know what the current resolution of the mesh is, you can use the eyedropper to sample it. However in this case since we used Decimate, it's certainly going to be extremely low. So increase the Dyntopo resolution to get enough resolution for your brush.



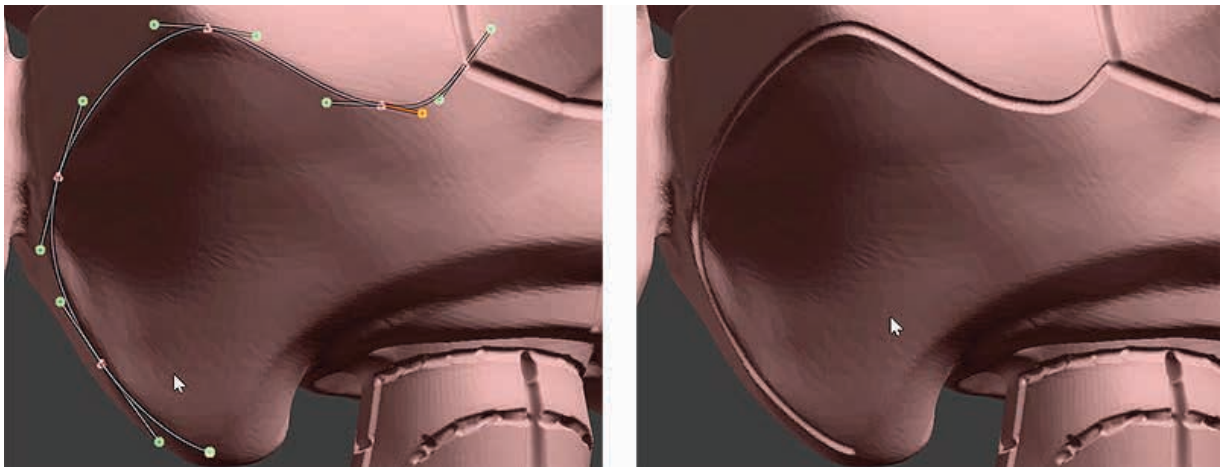
You can experiment a bit to get the amount of resolution that you need. Then to draw panel lines, I like to use mostly **Line** stroke method with the **Crease** brush. **Press Alt+E** to for the Stroke method menu. Set the **Radius** suitably small and the **Strength** to 1. Turn off Pressure sensitivity for both values.



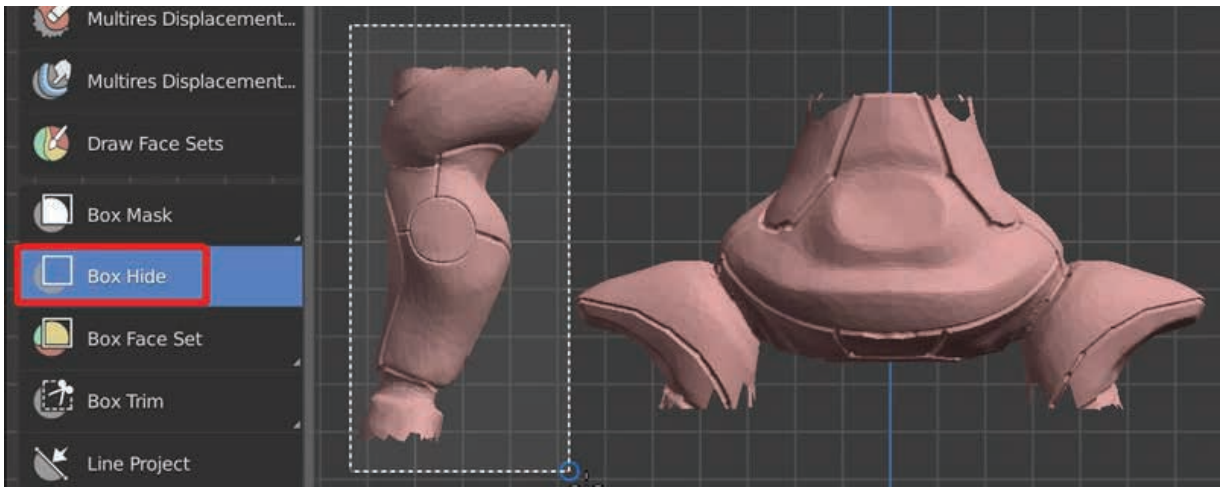
Now you can cut perfect straight panel cuts into your model. To do this, **left-click, drag the mouse to draw the line, and let go.**



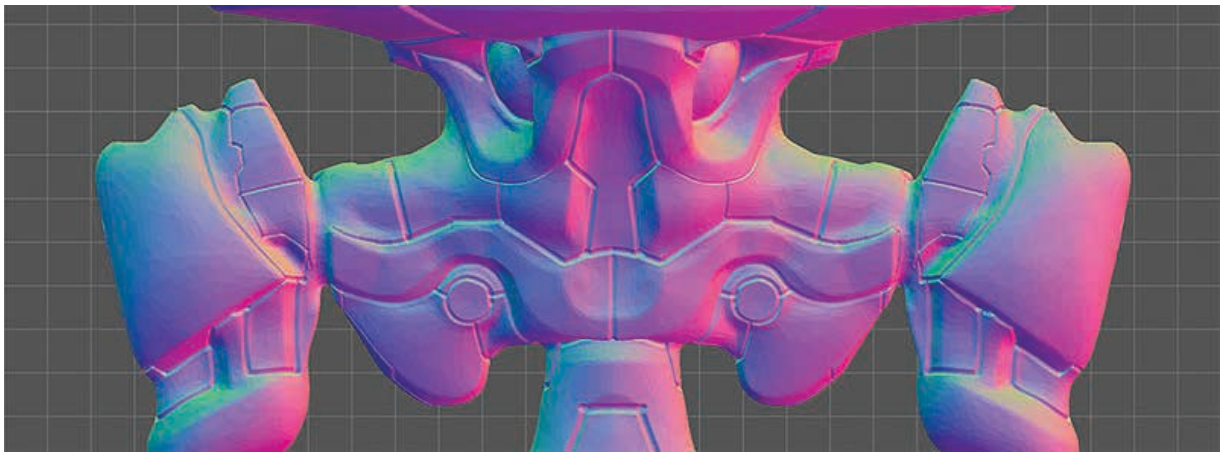
For curved or circular panel cuts you can use the **Curve Stroke Method** (as explained previously).



If your computer is struggling to keep up with all this geometry despite it being decimated, you can quickly hide the parts you're not working on using **Box Hide** to make it easier for your computer. To unhide, **hold Ctrl and drag a box across the hidden geometry** to unhide it (with the Box Hide tool active).



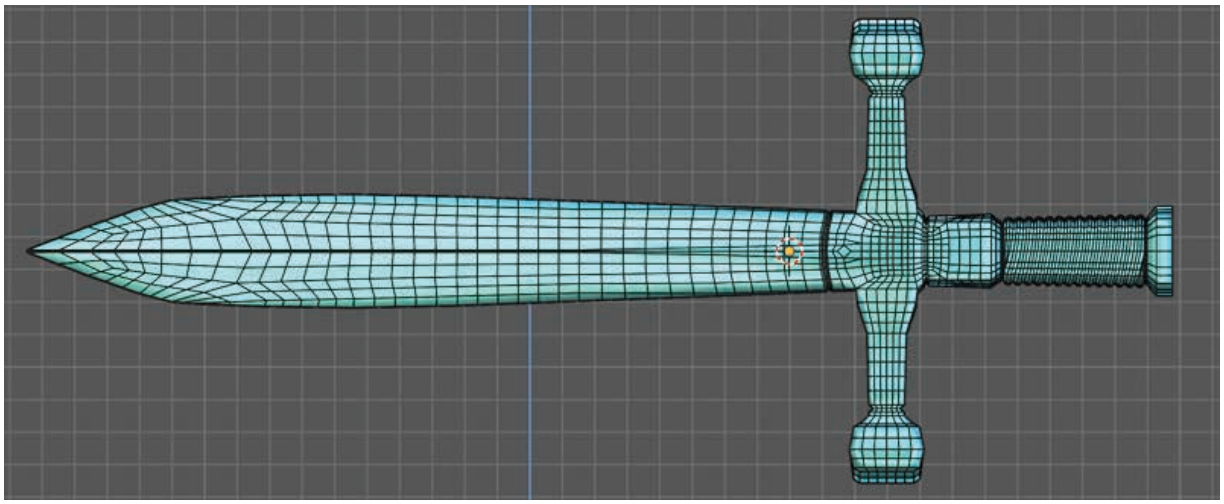
Besides making the model look more interesting and detailed, dividing it into panels will make retopology more easy and fun. Don't worry about the cuts not looking perfect. That's where the retopology stage comes in, which we will look at next!



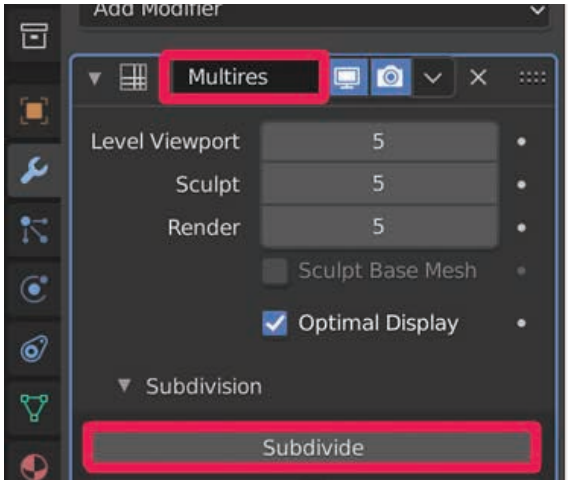
Sculpting Sword Details Using Alpha Textures

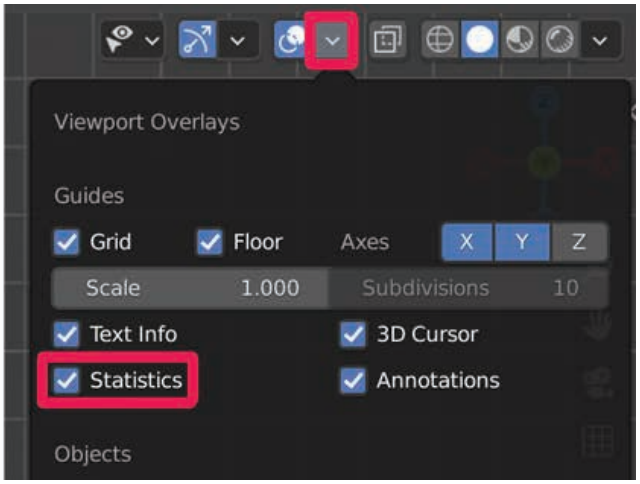
Alpha textures are black and white images that can either be used as masks or height maps, to easily sculpt details onto your model. In this case, I'm using some alphas I made on a simple fantasy sword. You can download the alphas here if you want.

For sculpting details, the object needs to have enough geometry. Make sure your edge loops are more or less evenly distributed. If necessary, add some more loops in areas that need them. Then add a **Multires modifier** and click on **Subdivide** a few times until you have enough geometry.

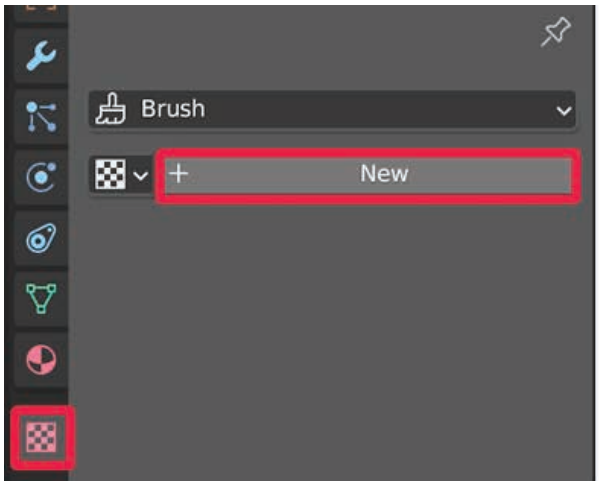


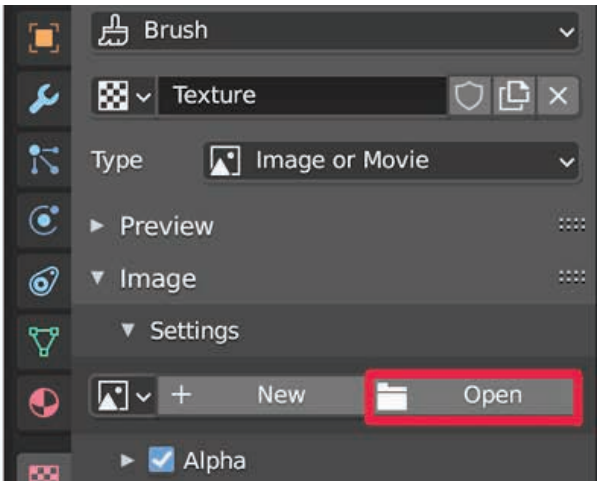
You'll want to make sure you have enough **Subdivision** levels on your **Multires** modifier. In the case of my sword model, it has about 12 million faces. You can turn on **Statistics** in Overlays to see the Face count.





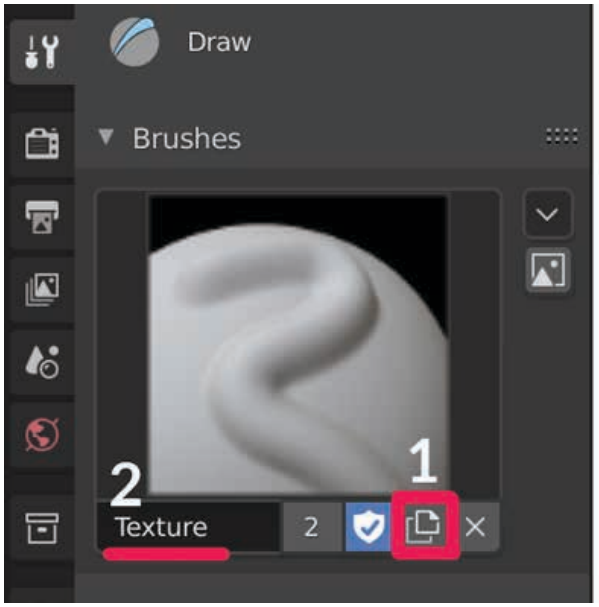
Go to **Sculpt Mode** and in the Texture tab, click on **New Texture**. Load a texture that you've either created yourself or found online.

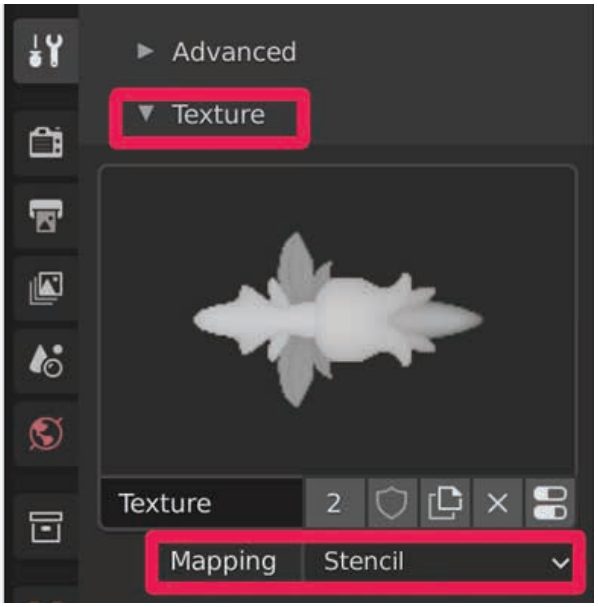




We'll create a new brush for use with this texture by clicking on the **Add Brush** icon *(nr. 1 in the picture below)*. Rename it "Texture" *(nr. 2 in the picture below)*. Under Brush Settings > Texture, set **Mapping** to **Stencil**.

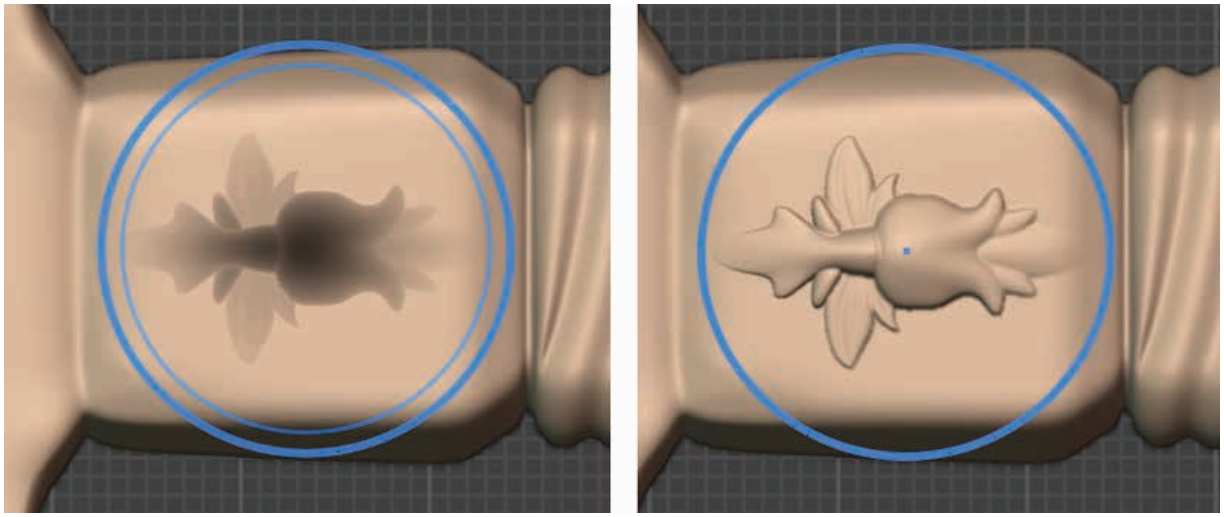
Stencil gives us a preview of the texture that we can move, rotate and scale.





You'll notice that a grey preview of the texture (the stencil) has appeared in the 3D viewport.

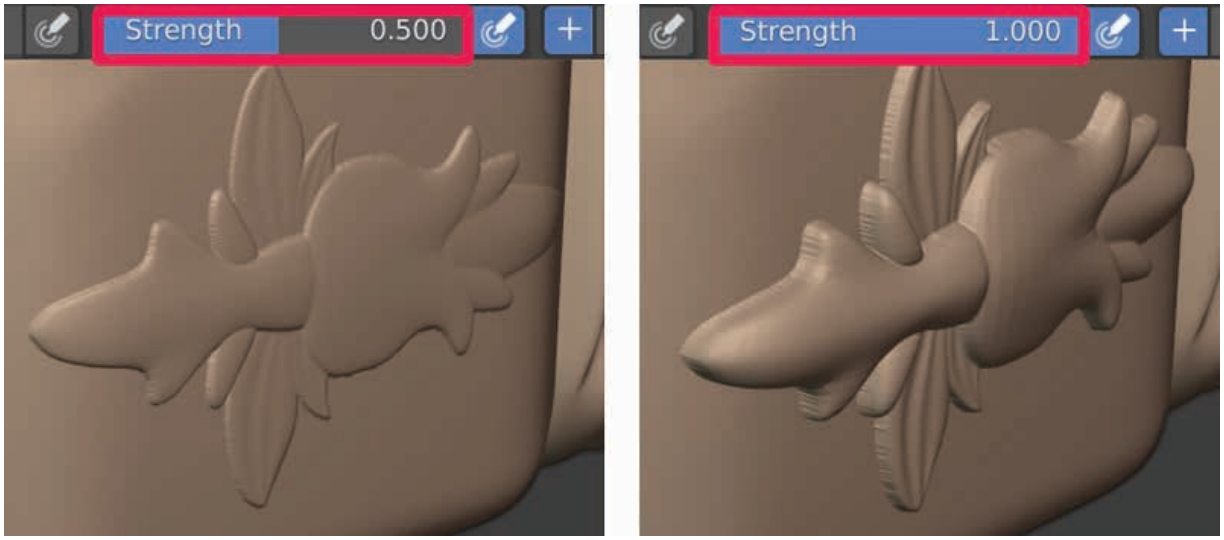
You can move the stencil by holding the **RMB (Right Mouse Button)** and moving the mouse. Rotate it by **holding Ctrl and the RMB** and moving the mouse. And scale it by **holding Shift and the RMB** and moving the mouse.



Make sure your Radius is bigger than the alpha detail!

Then **left-click once** to add this detail to the surface. You can change the Radius by **pressing F** and moving the mouse if needed.

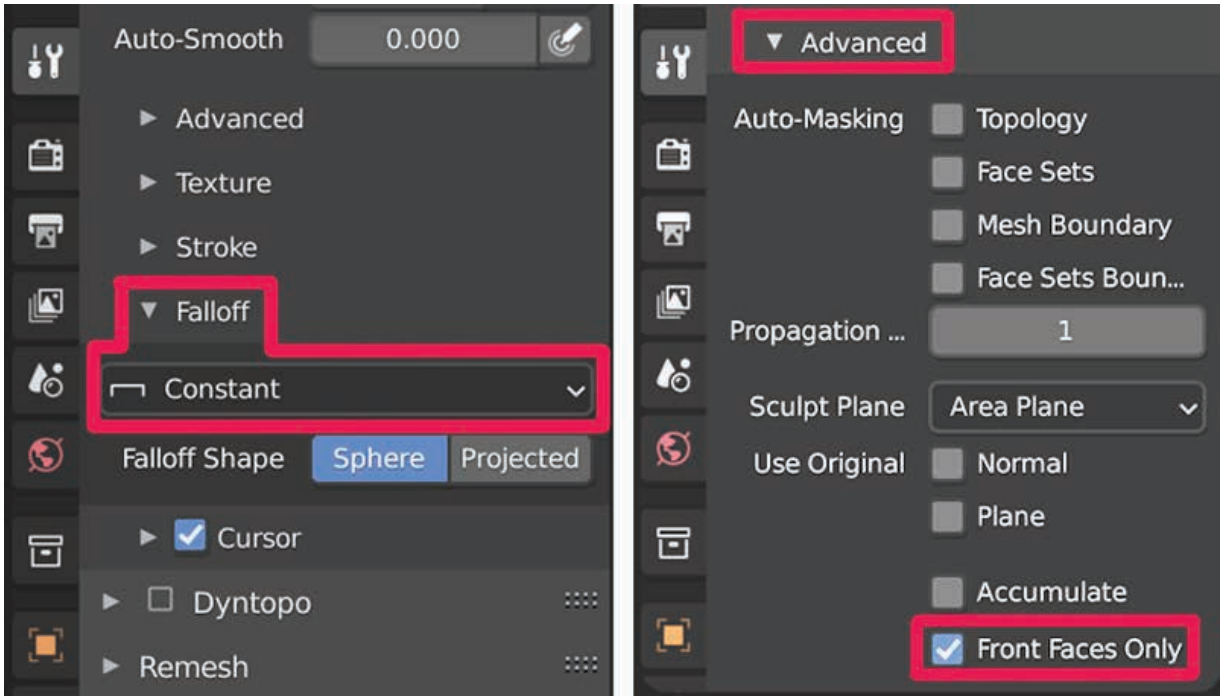
You may have to increase the brush **Strength** to get enough depth. You can always undo and change the Strength, then try again.



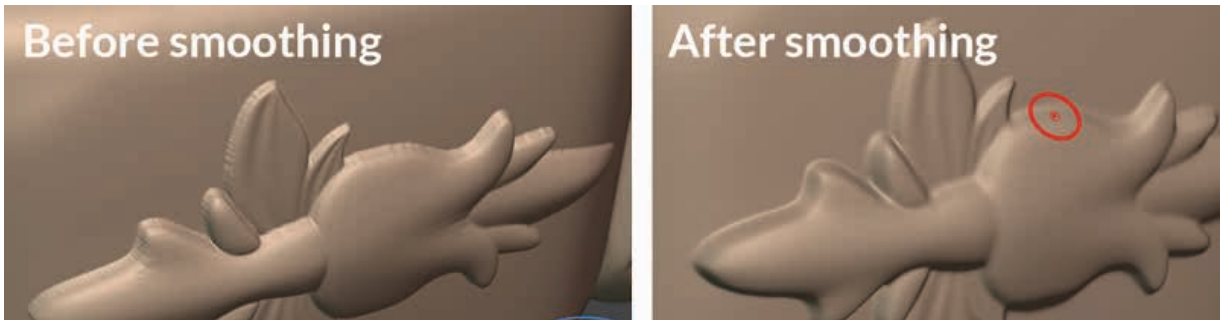
A comparison of Brush Strength 0.5 and 1.0.

With these particular details, I want them not to smooth out towards the edges. To accomplish that it's important to set Brush **Falloff** to **Constant**.

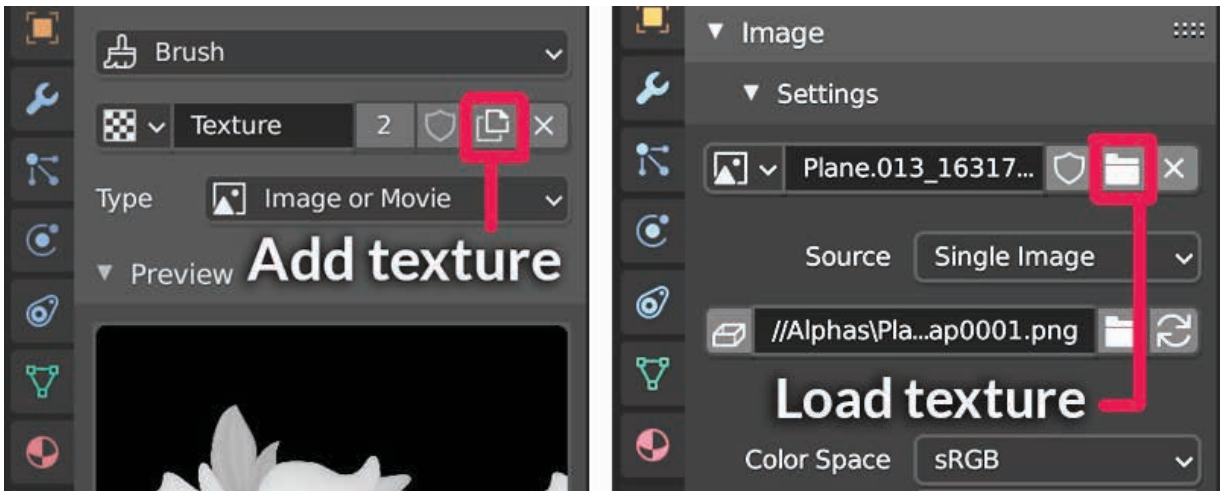
And in order to make sure you don't accidentally affect the other side of the mesh, check **Front Faces Only** under the Advanced brush options.



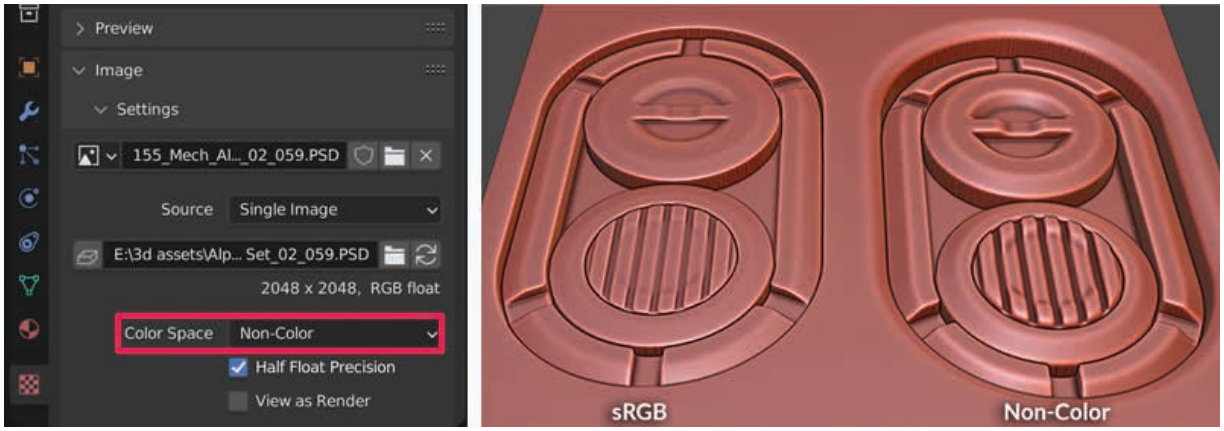
Sometimes the result may need a little bit of smoothing to look nice. Use the **Smooth brush** set to a very low **Strength** value to do this.



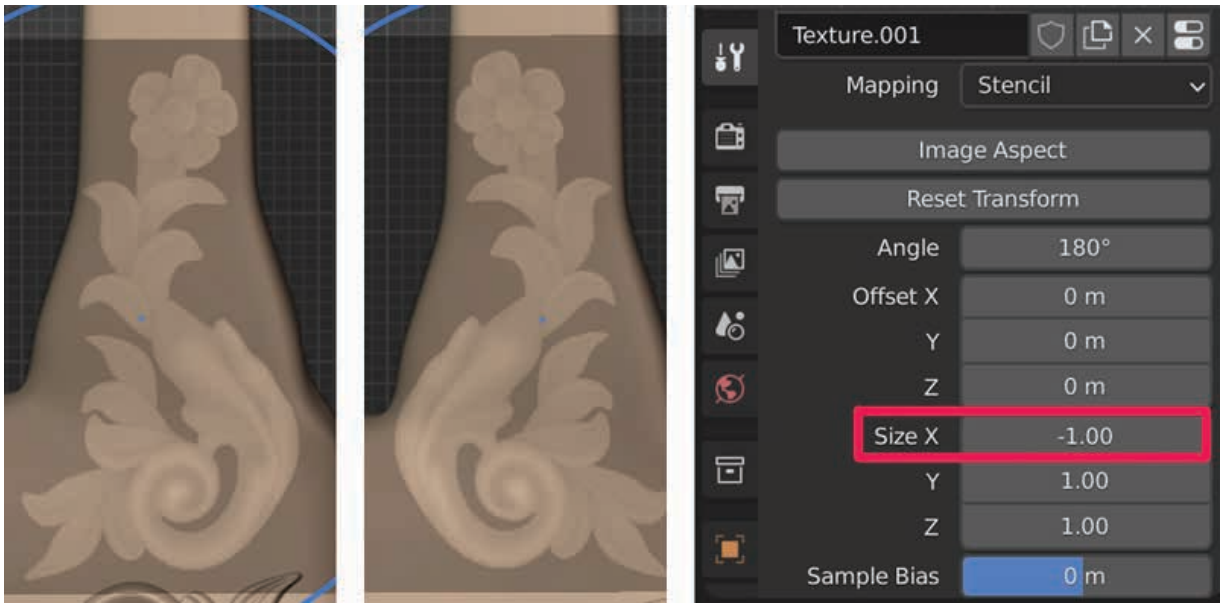
You can add more textures to the brush. In the Texture tab, click on **Add Texture**. Then load the texture you want to use.



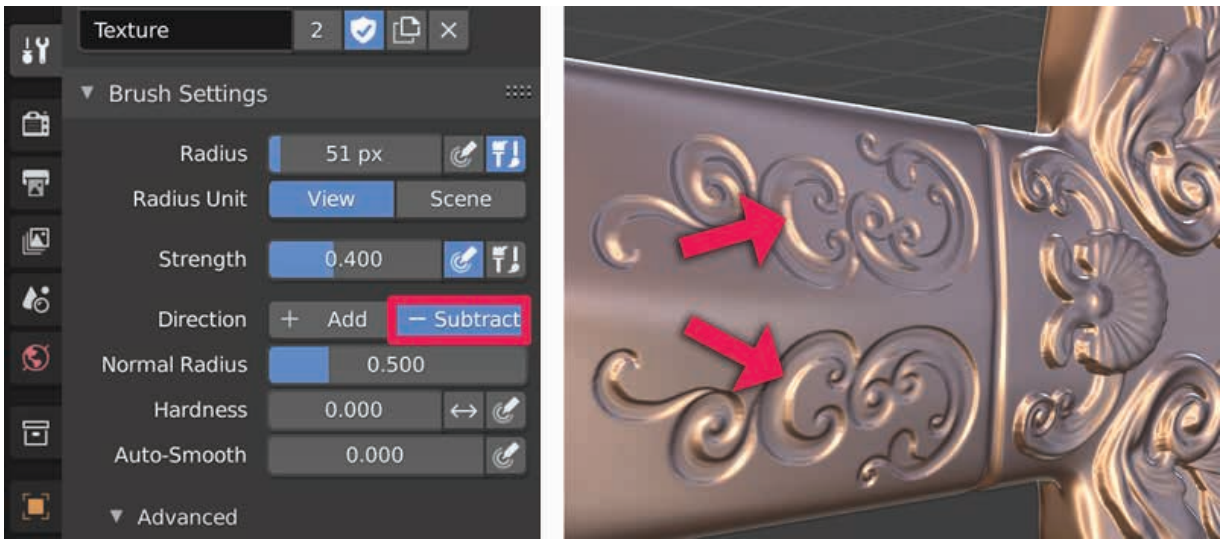
You also want to set the textures to "**Non-Color**" or "**Linear**" instead of "sRGB". This way, you get the most detail out of them, as seen with these mechanical details:



Now you can use the same brush with the new texture. To flip the texture horizontally, simply set the X value for the **size** to **-1**.

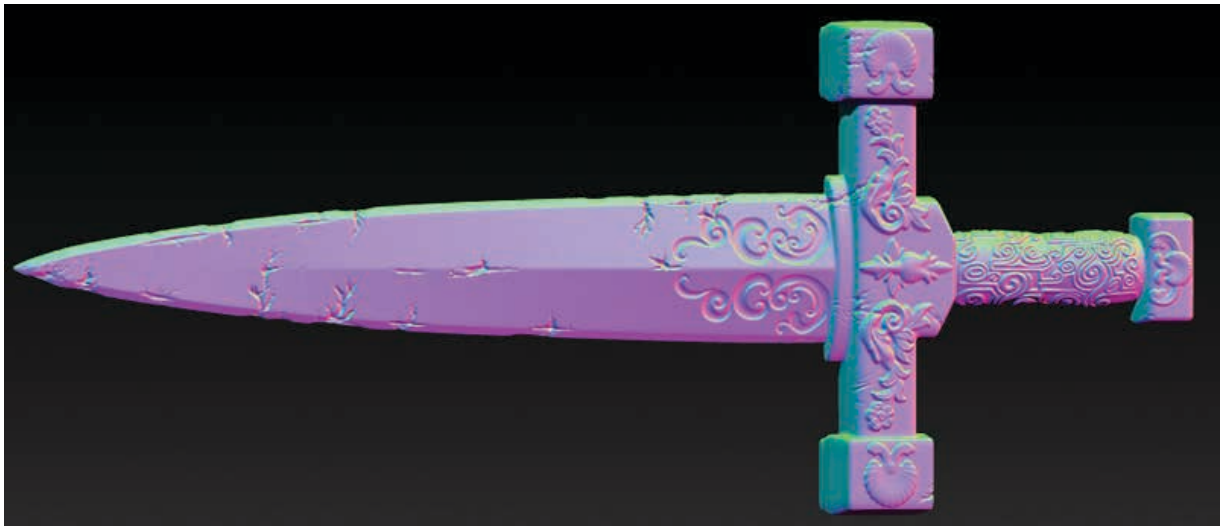


If you don't want to add height but rather want to subtract from the surface, you can do so by setting the Brush **Direction** to **Subtract**.



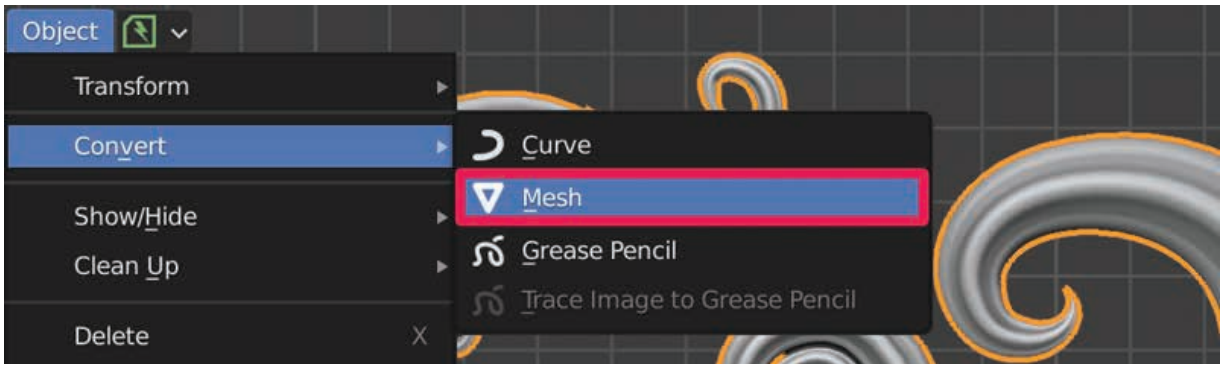
Creating Alphas for Sculpting

Previously we looked at using Alpha textures for adding sculpted details to a model. You can find some nice paid and even free alpha textures online (on Artstation for example, if you search in Marketplace>3D Brushes) but making your own alpha textures for detail sculpting is easy! Especially since there's a good free add-on available that makes it basically a one-click action.

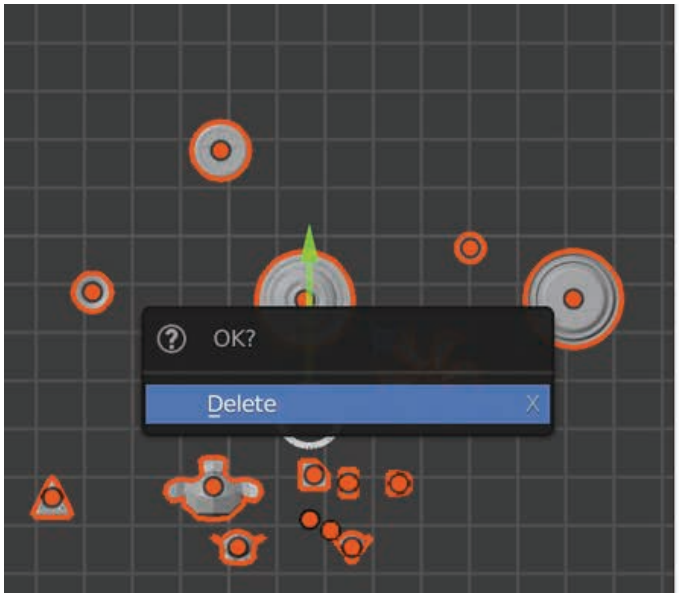


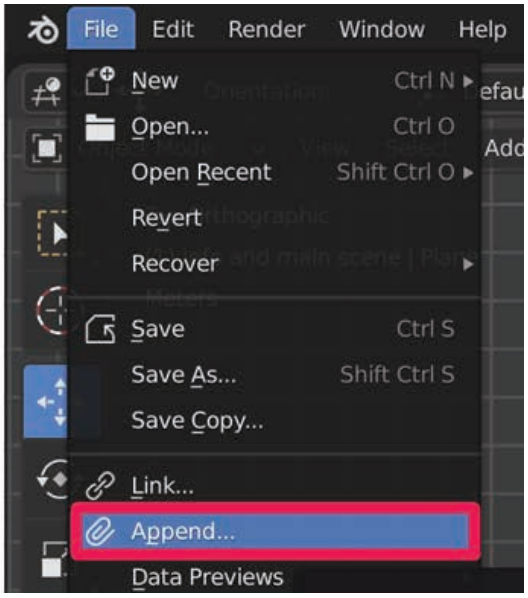
First, create some 3D details using one of the methods explained in previous videos about modeling Ornaments ("8 Tips for modeling Decorative Elements" and "Drawing tapered ornamental details on surfaces").

If you use curves, do convert them to mesh, otherwise the next steps won't work. To do this go to **Object > Convert > Mesh** or **press F3** and search for "**Convert > Mesh**".

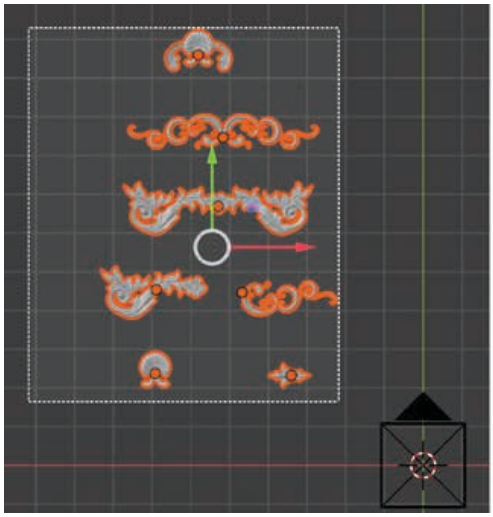


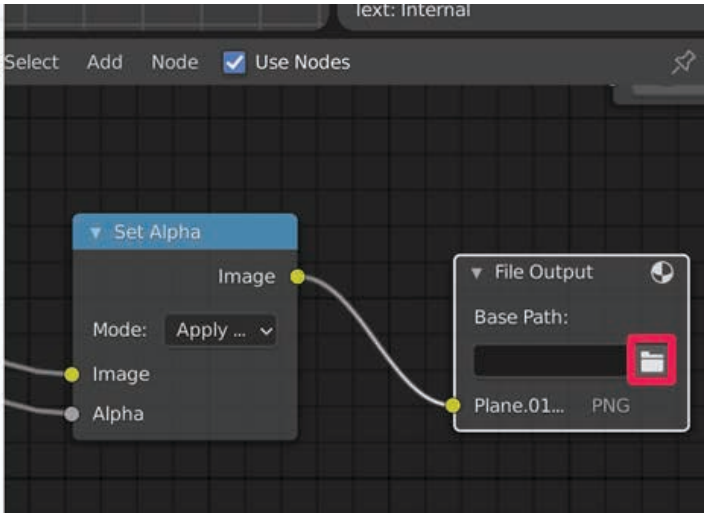
Download the free Blend file shared on Artstation by Twitter user @quiperr here: https://www.artstation.com/marketplace/p/xKO3O/height-map-and-alpha-map-creator-from-meshes You can just delete the test objects that come with the Blend file. Then append your own objects.



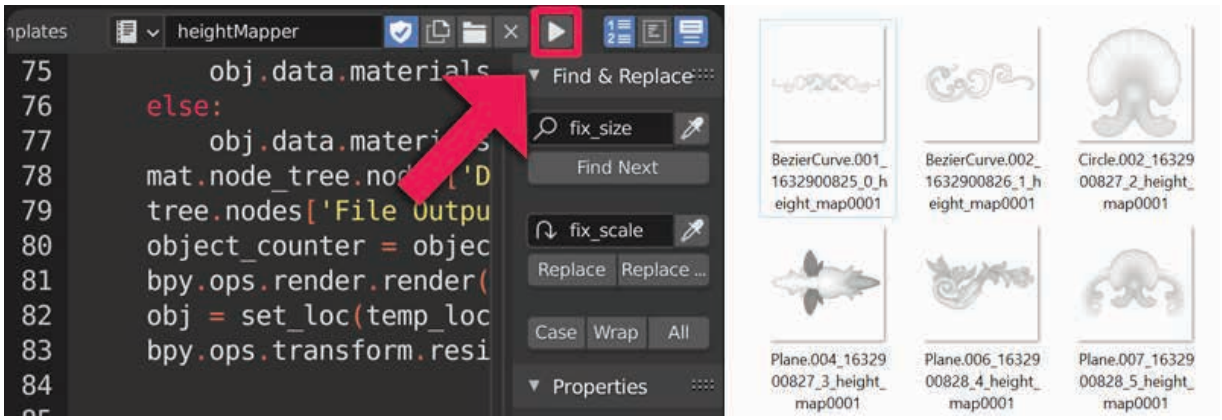


First, make sure they're placed outside of the camera range. Other than that, their location or scale doesn't matter (as long as they aren't too close together). Change the destination path in the **File Output node** and select the objects you want to bake to alphas in the Outliner.





Then, just **run the script** by clicking on the button indicated in the screenshot below. Your alpha textures will appear in the folder you specified. It's kind of fun to see them appear one by one!

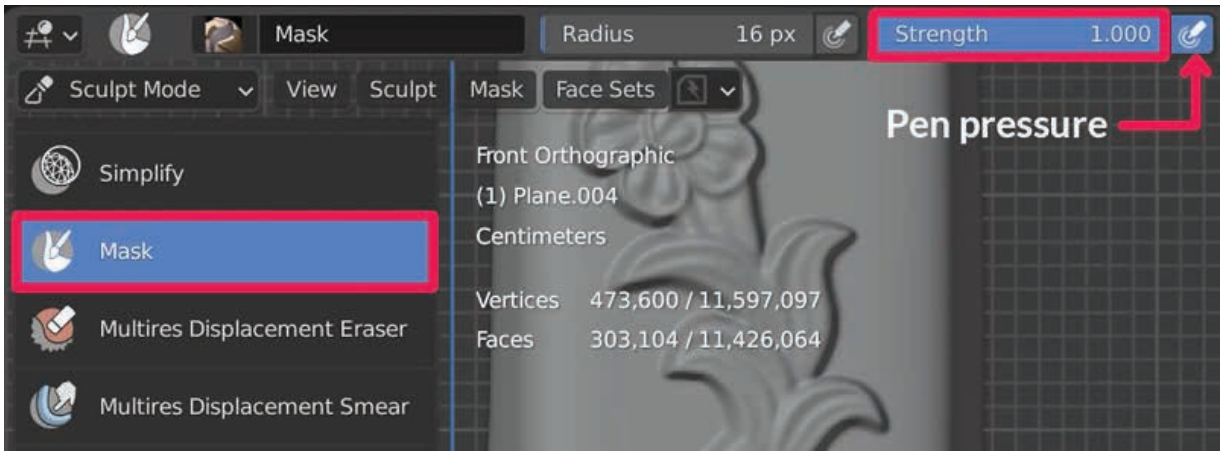


Now you can use those Alpha textures to sculpt details on your objects, as explained in the previous topic.

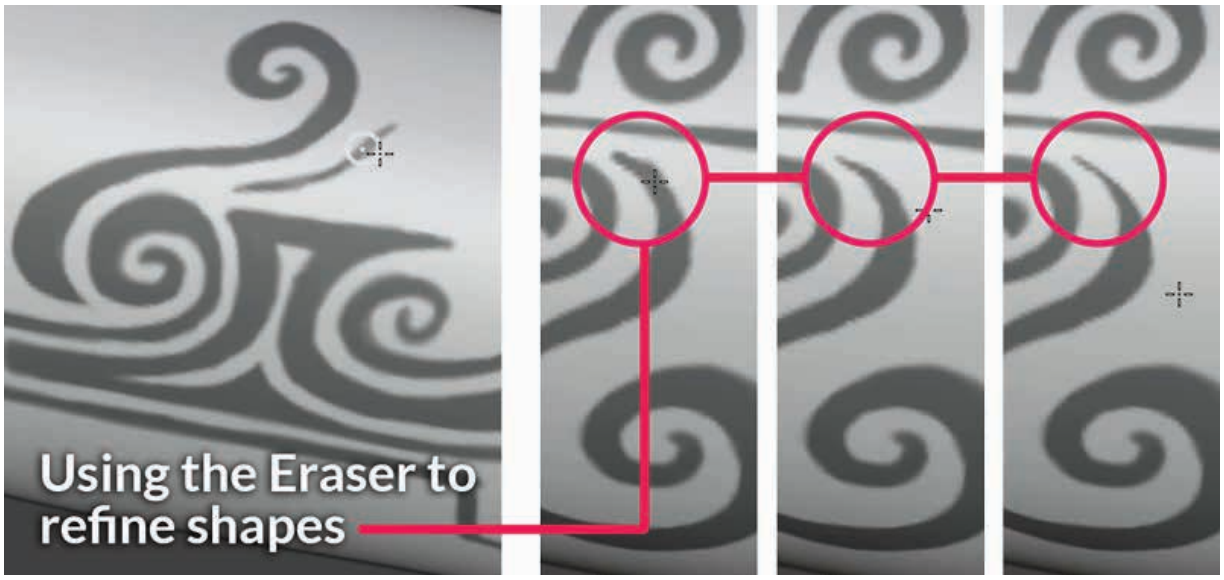
Sculpting details using Masking

Besides using **Alphas**, we can also paint **masks** directly on the model to create details. This is useful for very curved surfaces for example. Having plenty of resolution is important, and here too I recommend the **Multires modifier.**

Select the **Mask brush**. You can change the **Radius** by **pressing F** and also enable pressure for the Radius, so you can control it with the pen pressure. I like to set the **Strength to 1**.

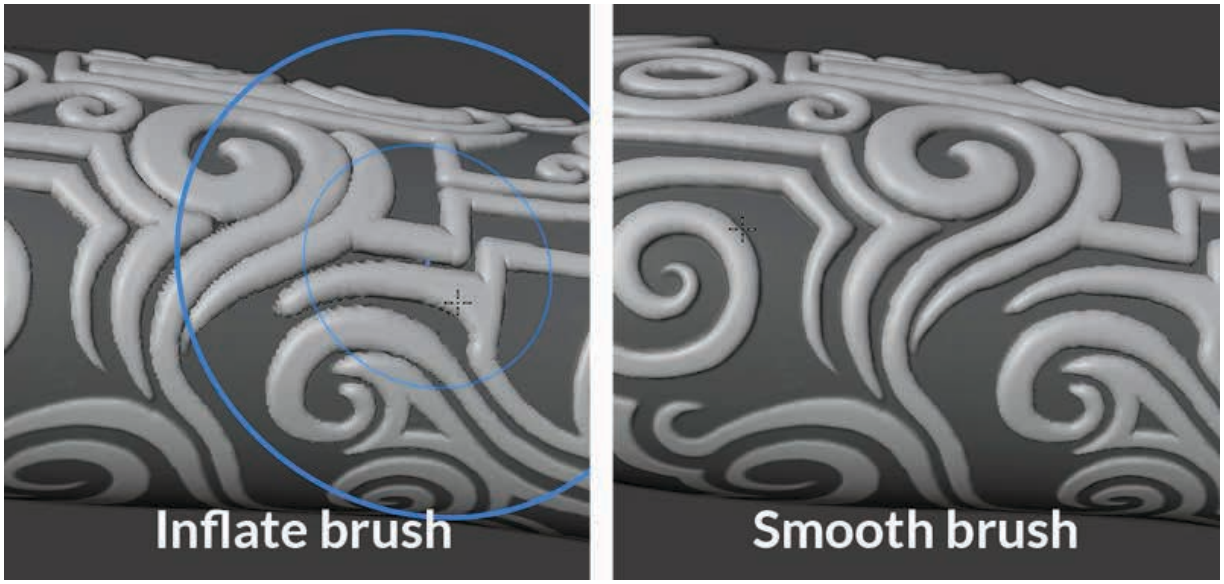


Draw some details with the Mask brush. **Holding Ctrl** turns the pen into an eraser.



Pressing Ctrl+i inverts the selection, so now the mask you drew is the only thing not masked. This makes it possible to inflate this selection.

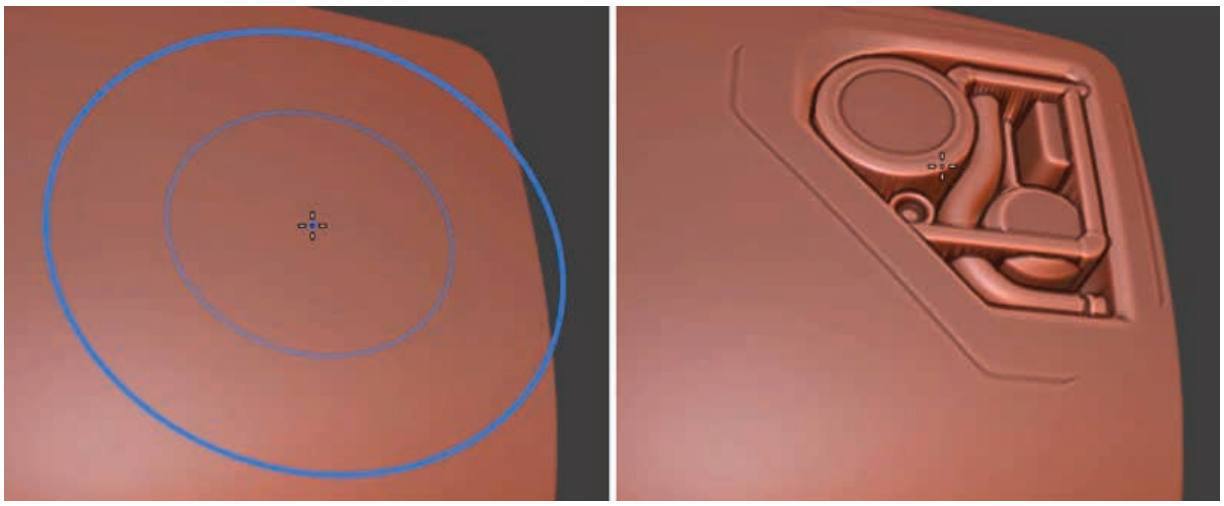
Using the **Inflate brush**, you can now gently raise the unmasked parts. You could also use different brushes like the **Draw brush**, but the Inflate brush gives it a nice falloff. After that, you can smooth it out while **holding Shift**.



Pressing Alt+M clears the mask. A bit of additional **smoothing** here and there after all this is a good idea.

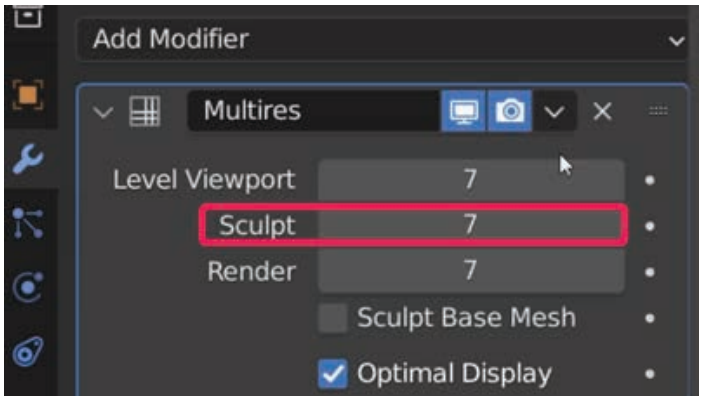
Placing Hard Surface Details in Sculpt Mode

Alpha textures are the fastest way to add details to a model in Sculpt mode. They are used a lot for hard surface sculpting as well. You can find a lot of free alphas and buy big packs of them on Gumroad and Artstation. Usually they are sold as Zbrush alphas, but really they are software agnostic and will work just as well in Blender.

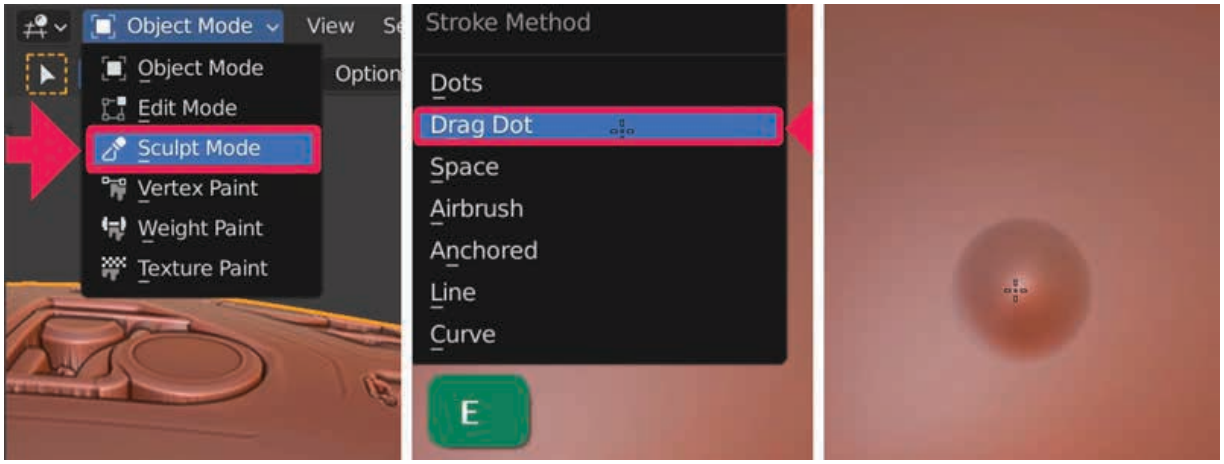


Previously we had a look at using these to make a sword, and we looked at how to create your own alphas.

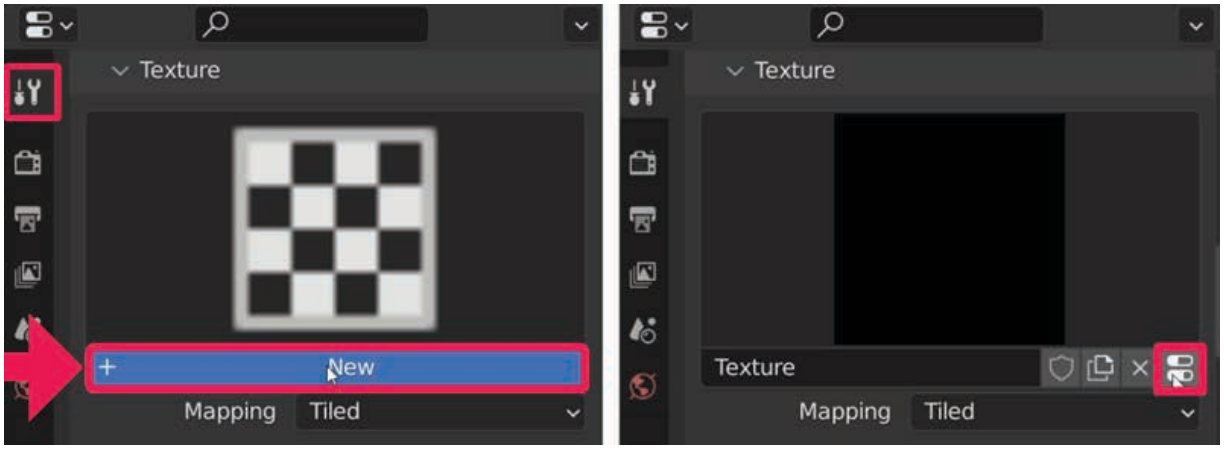
Make sure your model is subdivided enough for sculpting, for example by using a **Multires modifier**.



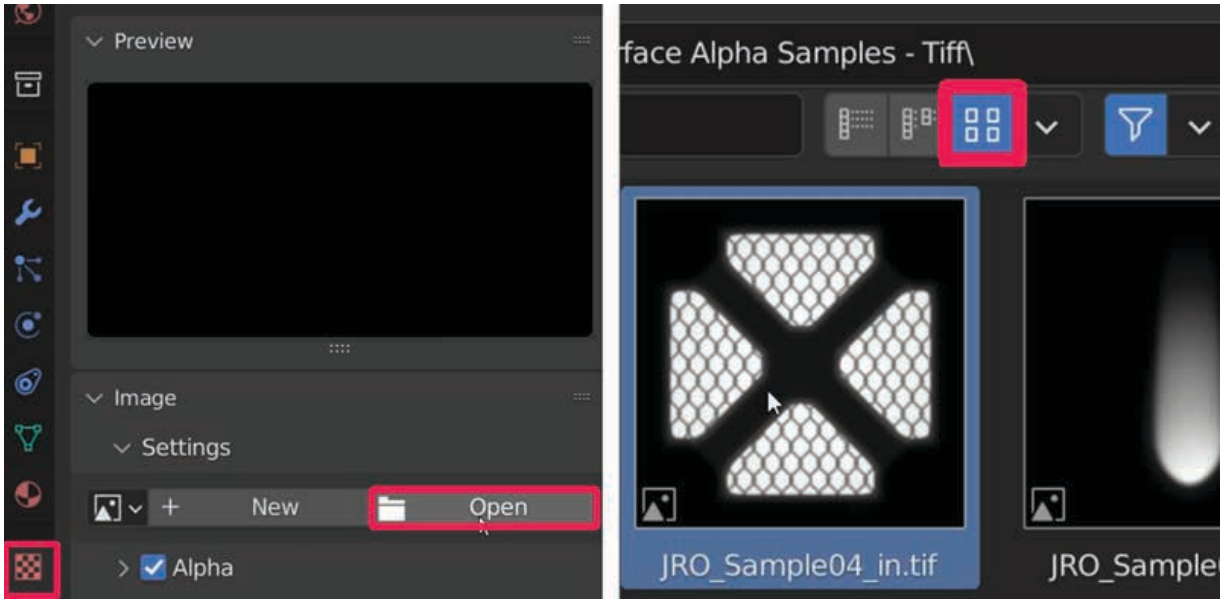
Go to Sculpt Mode. **Press E** for Stroke menu, choose **Drag Dot**. Now you can drag dots across the surface.



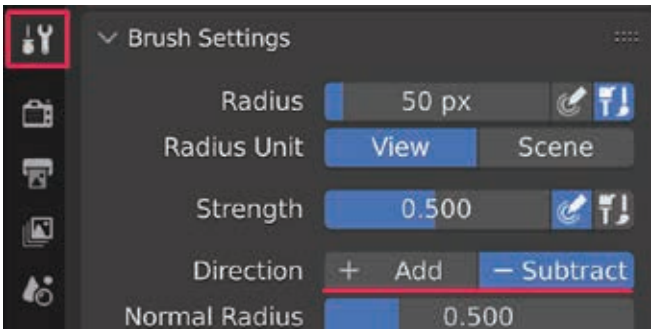
For this example I'm using these free alphas from JROTools. Create a new Brush texture and load one of the Alpha textures.



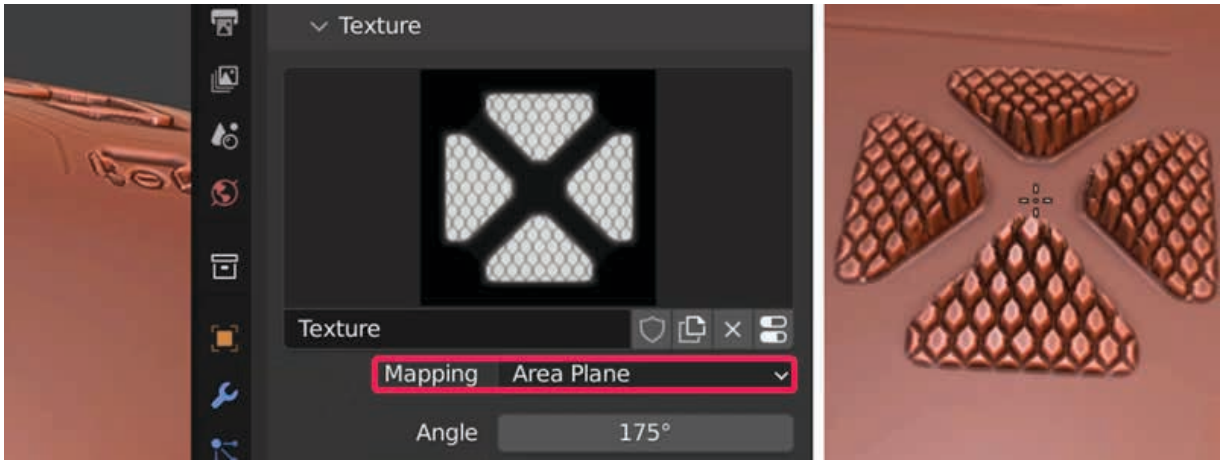
You can set the Display Mode to Thumbnails and pick one you like.



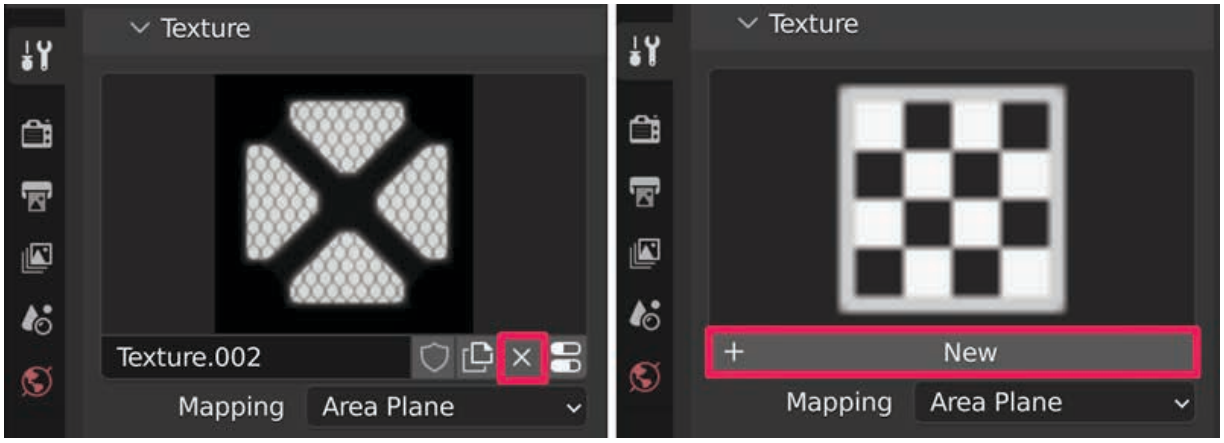
Depending on the alpha texture, you may have to choose either **Add** or **Subtract** as the Direction.



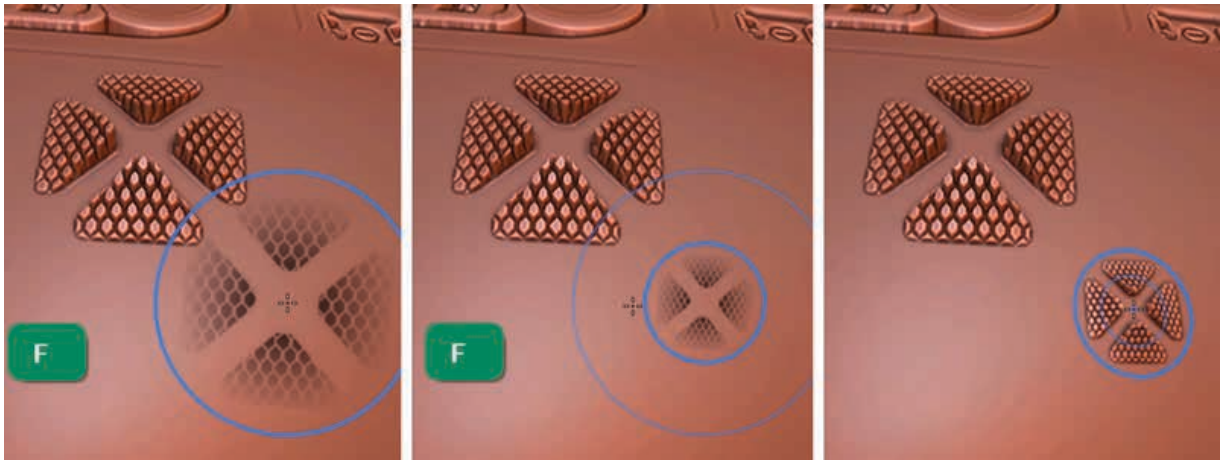
Set mapping to Area Plane so you can drag the alpha to where you want it on the surface, even in a perspective view. Drag the detail and place it where you want.



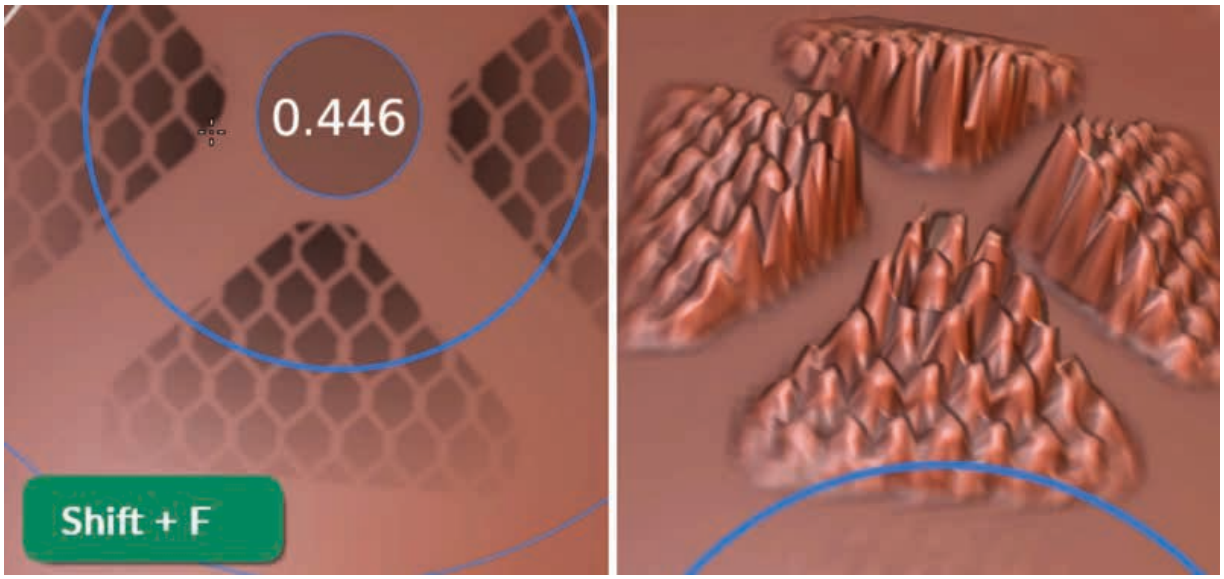
Click on **New** texture if you want to add additional alpha textures.



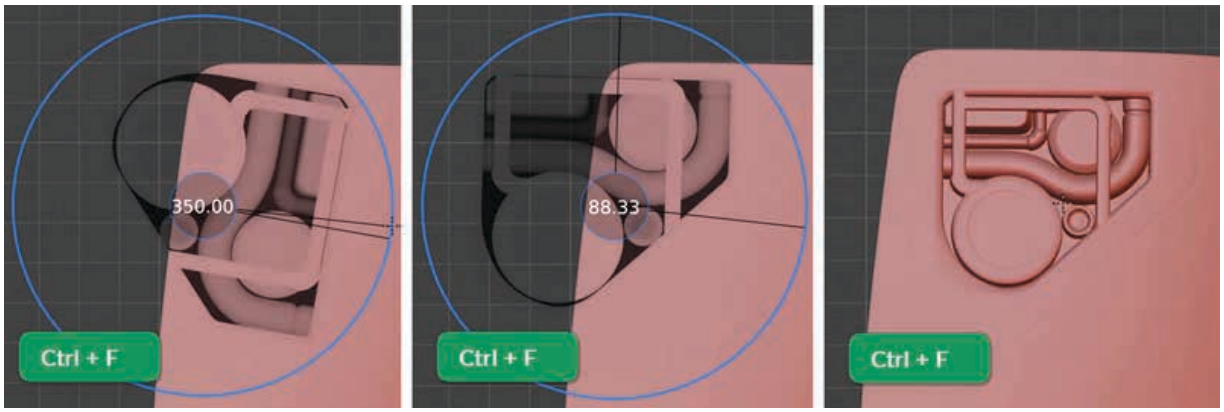
Press F to change the **Radius**.



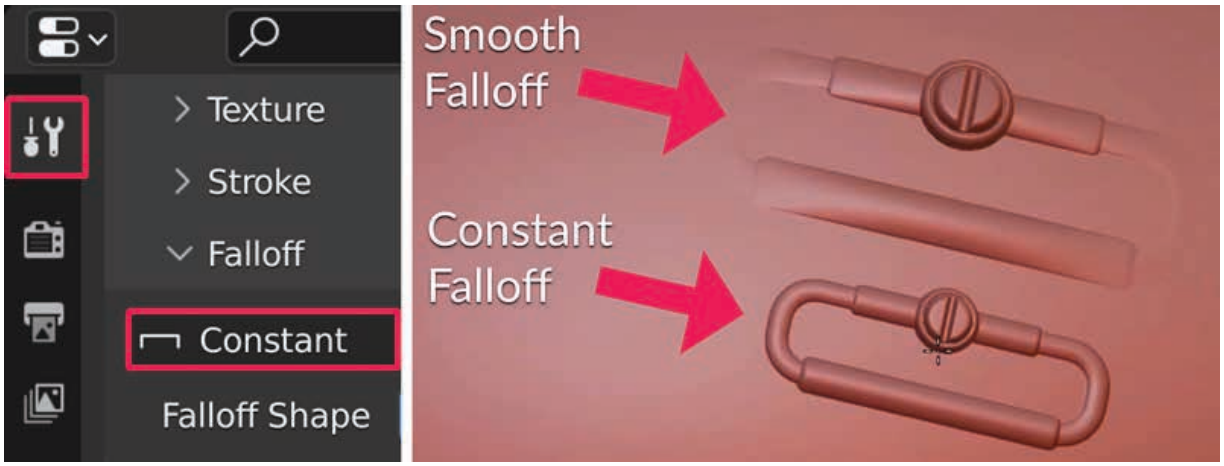
And press **Shift+F** to change the **Strength**.



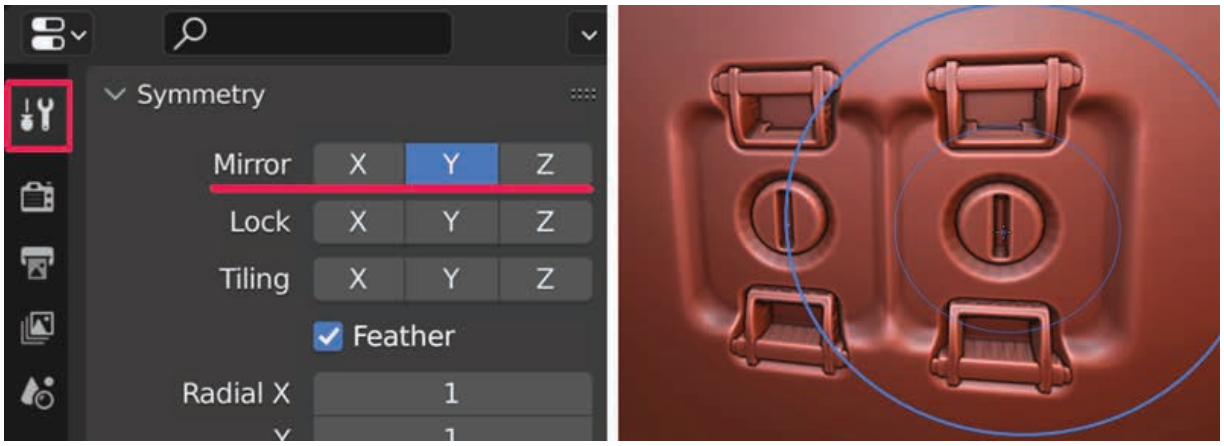
If you want to rotate the texture to a different angle, **press Ctrl+F** before drawing it on the surface.



You may want to use **Constant Falloff** to make sure the entire texture is used when the Strength value is lower.

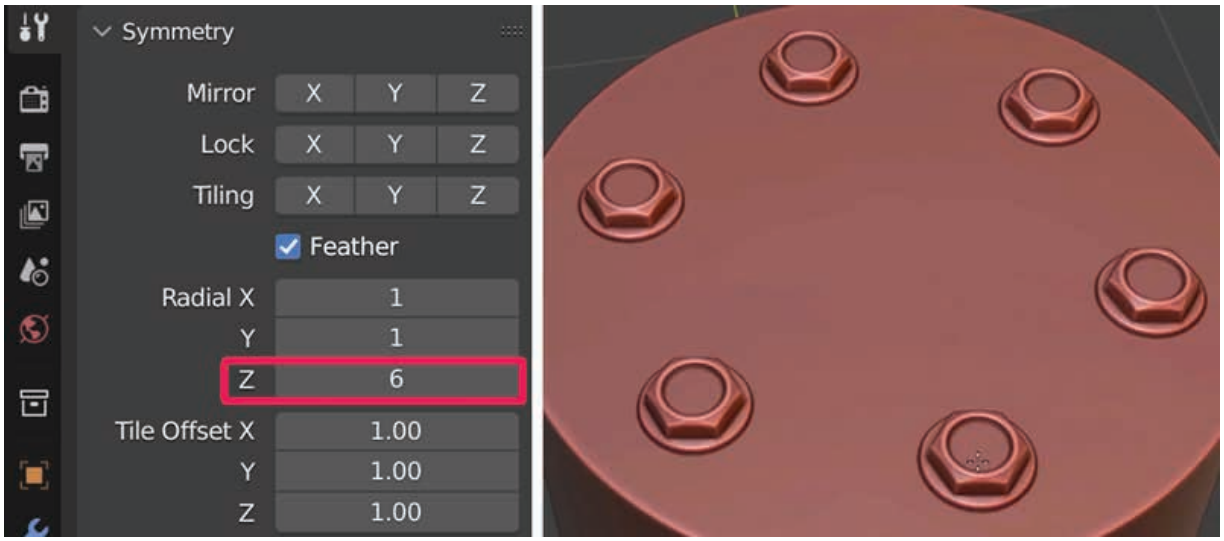


You can also turn on **Symmetry** along an axis of your choosing to mirror the brush.



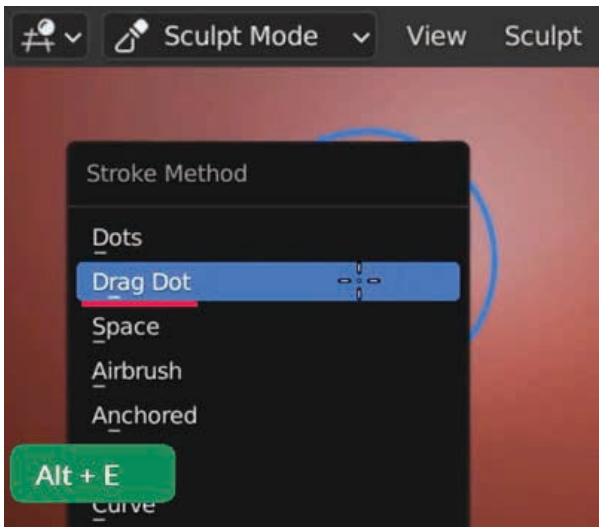
Radial Symmetry in Sculpt Mode

You can drag alpha textures using the Drag dot stroke method one by one (as shown in the previous topic), but you can also take advantage of the **radial array** that is available in Sculpt mode. This is convenient if you need to create a perfect circular array of bolts, for example.

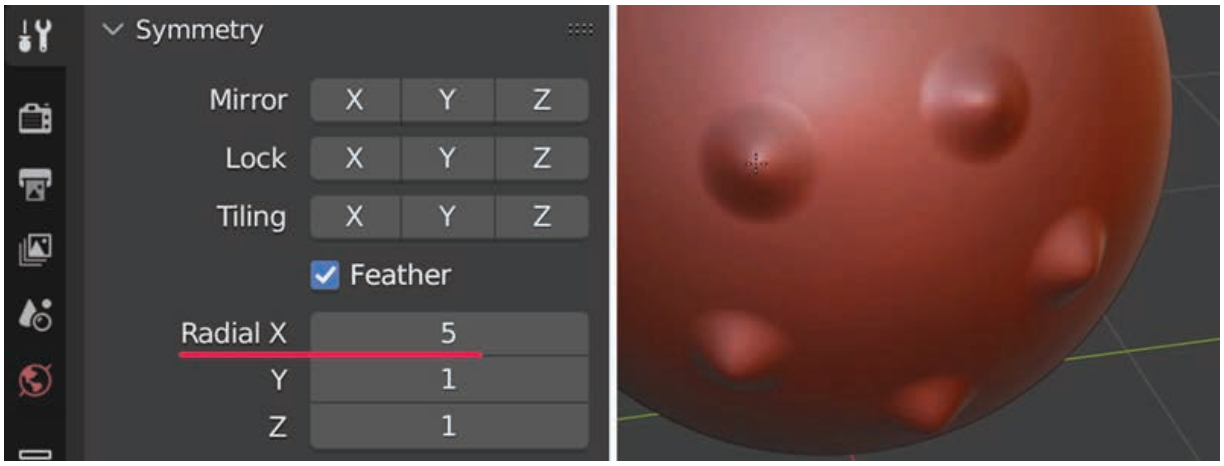


Go to Sculpt Mode.

Press Alt+E for Stroke menu, choose **Drag Dot**.

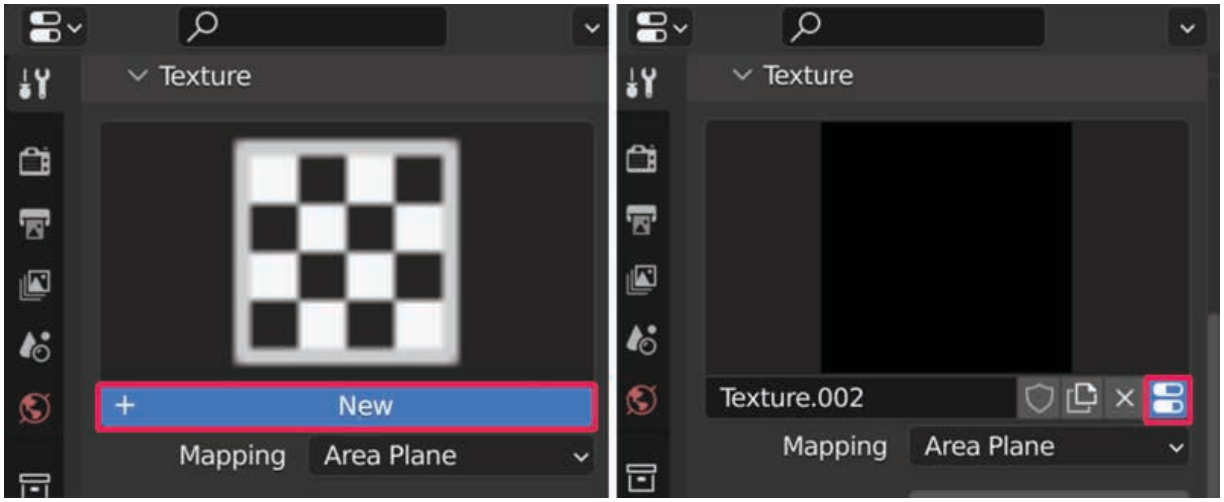


Set a Radial value on the axis you want. Now you can drag dots in a radial pattern.

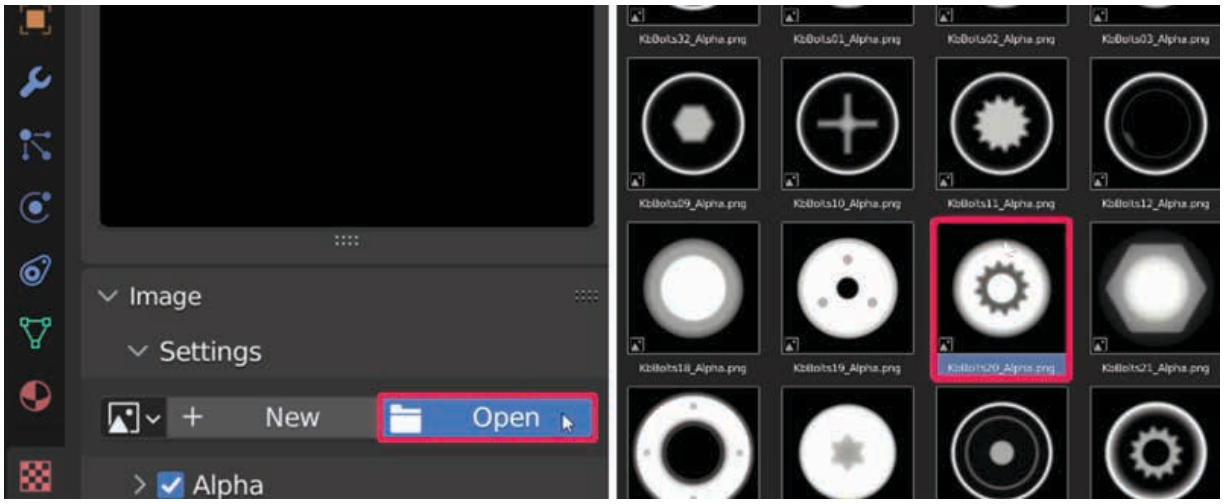


You can choose one or several angles on which to sculpt with a radial array.

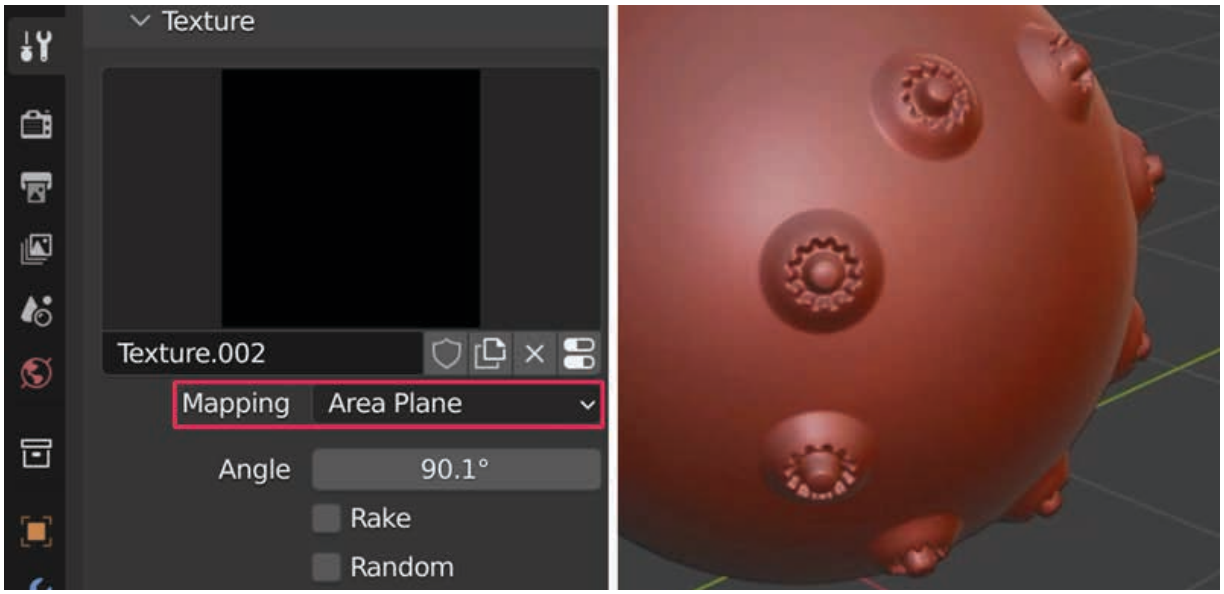
Click on **New** texture and load an **alpha texture**.



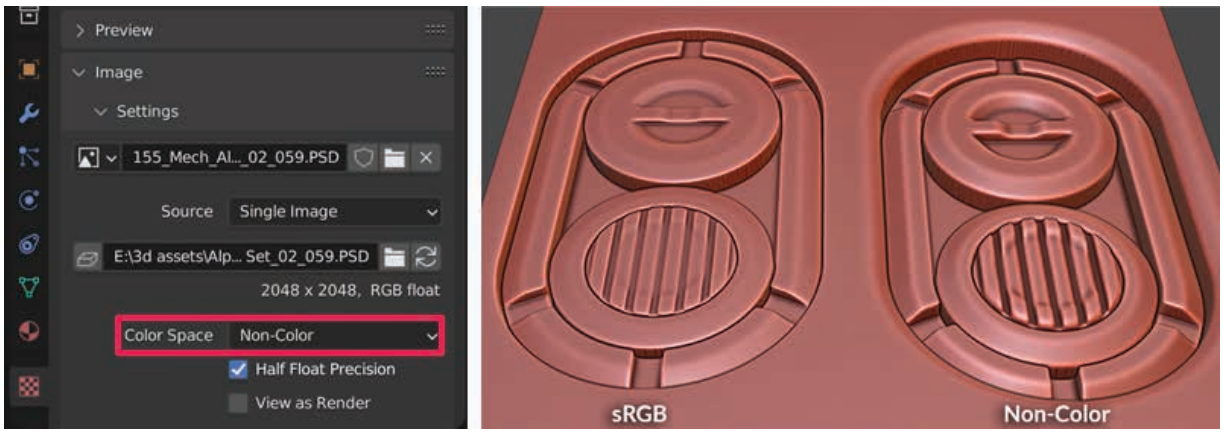
For this example I'm using these free alphas from Bergman3D.



Make sure **Mapping** is set to **Area Plane**. Now you can create a radial array of hard surface details like bolts.

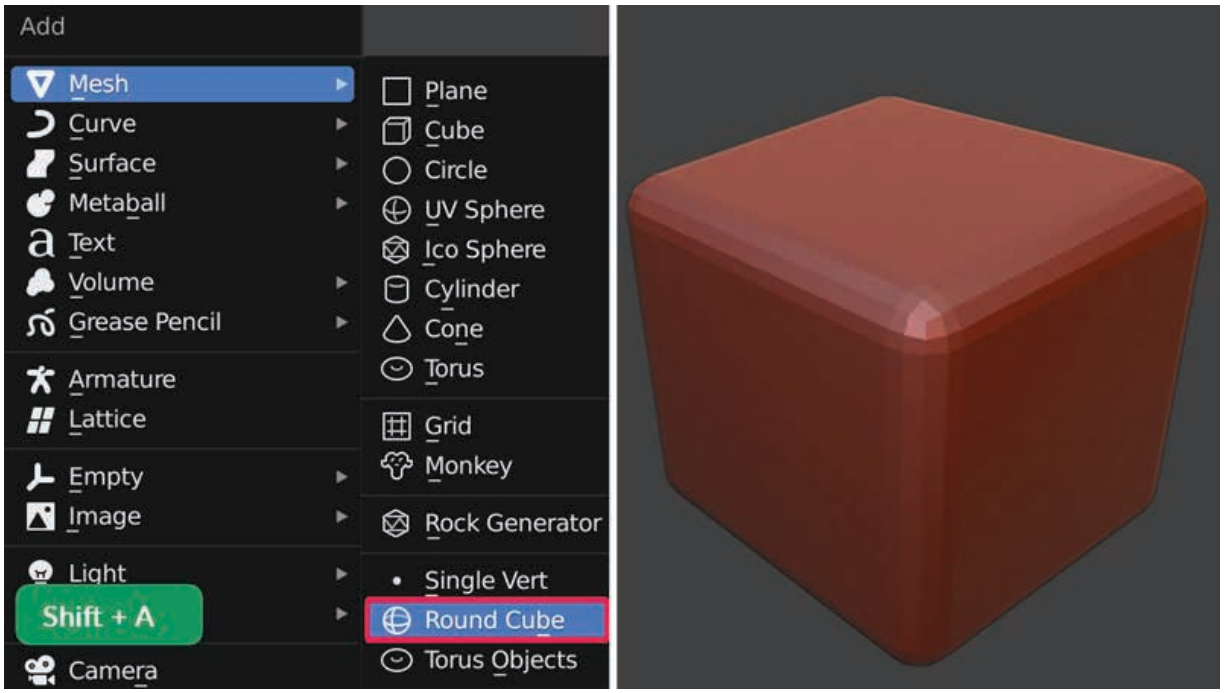


One final detail: You'll want to set the textures to "**Non-Color**" or "**Linear**" instead of "sRGB". This way, you get the most detail out of them.



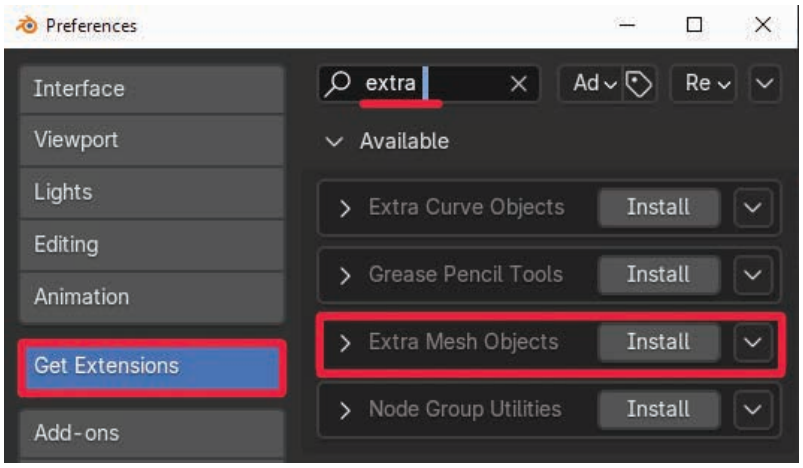
Using the Layer Brush

Let's add a **Round Cube**.

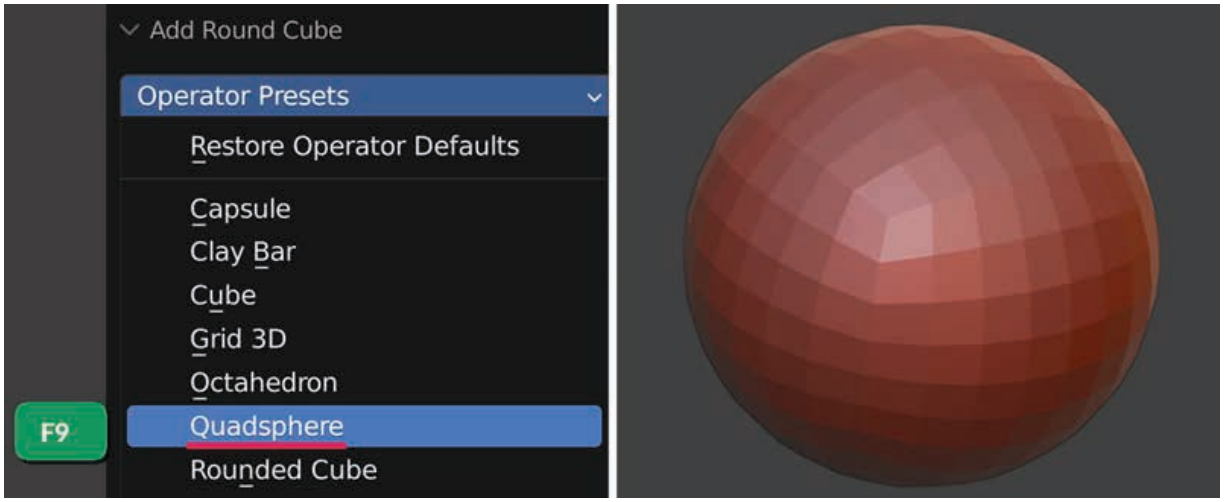


This Round Cube is from the **Extra Mesh Objects add-on** that you can enable in **Preferences**.

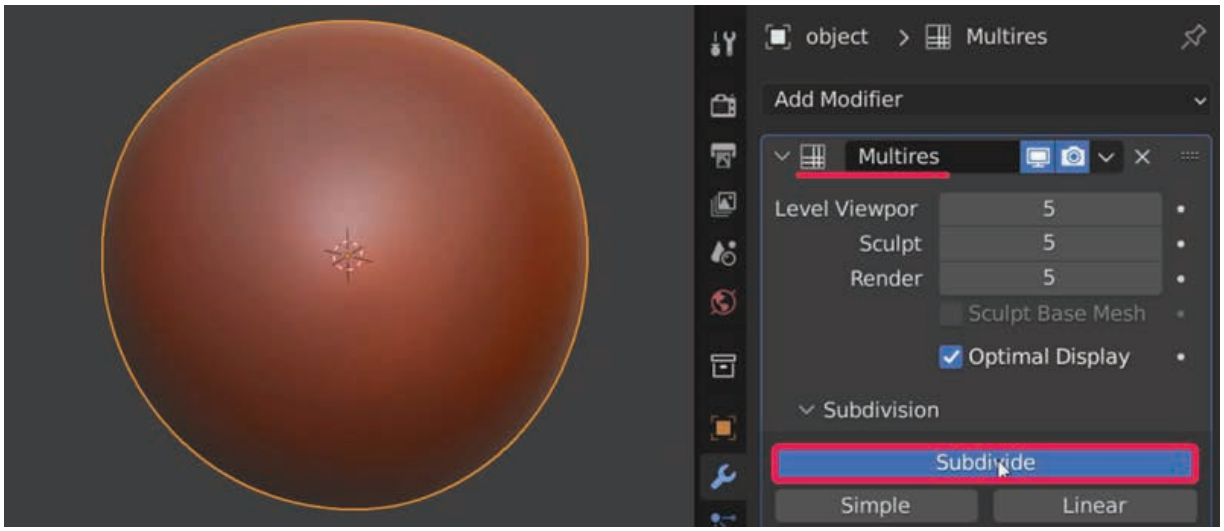
You can find it in the Get Extensions tab.



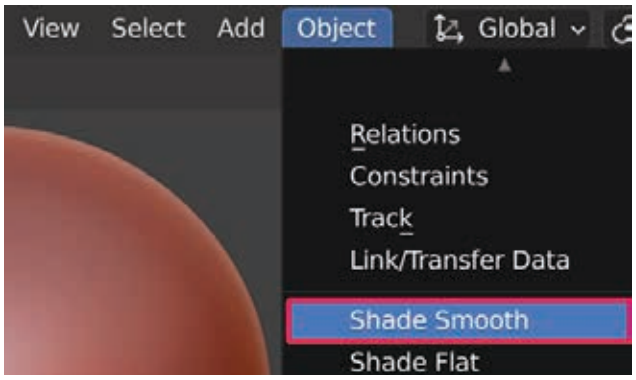
If you accidentally lose this menu, **press F9** and it will pop back up. We need it to select the **Quadsphere** preset.



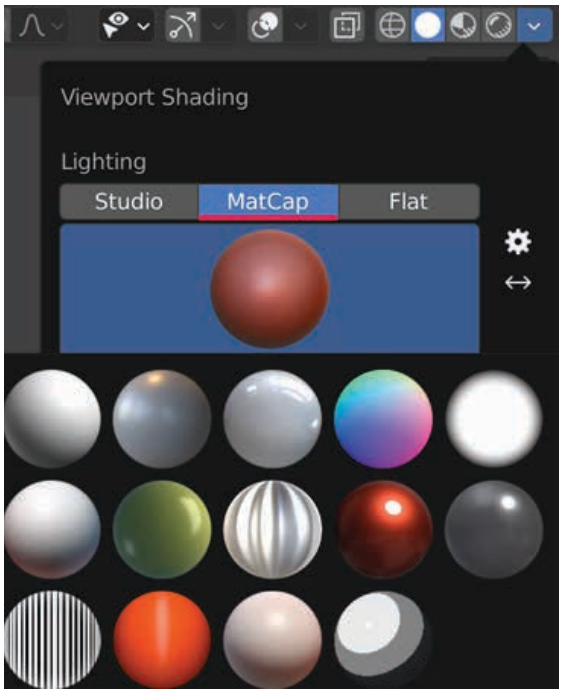
To add some extra subdivisions, which are necessary to sculpt on this sphere, I add a **Multires modifier.** And then increase the Subdivisions by clicking on **Subdivide** a few times.

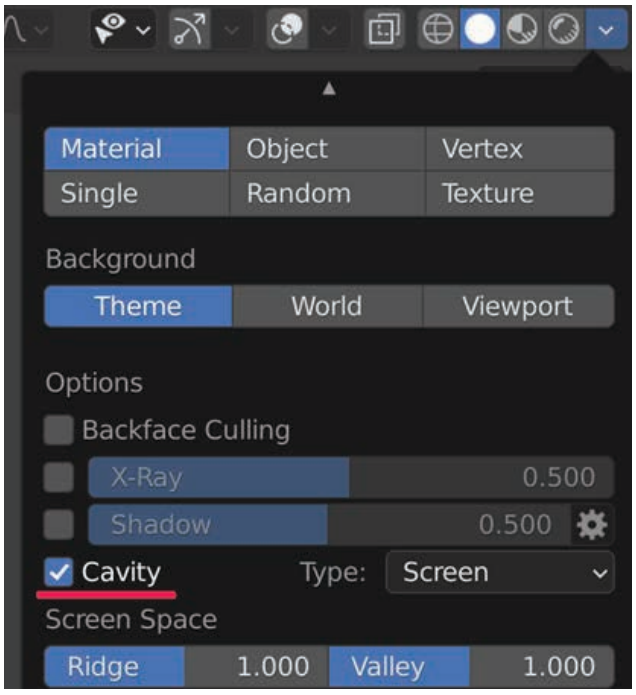


It always helps to **Shade Smooth**.



Let's use a red **Matcap** and turn on **Cavity**, it makes the sculpting look a bit more pronounced. A bit more like ZBrush...

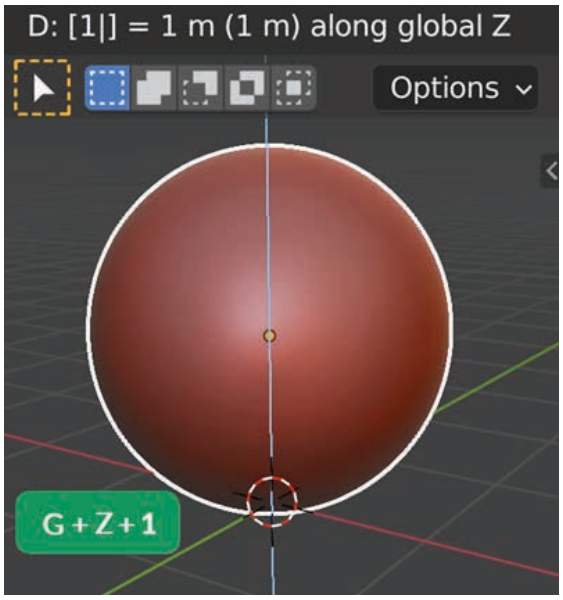




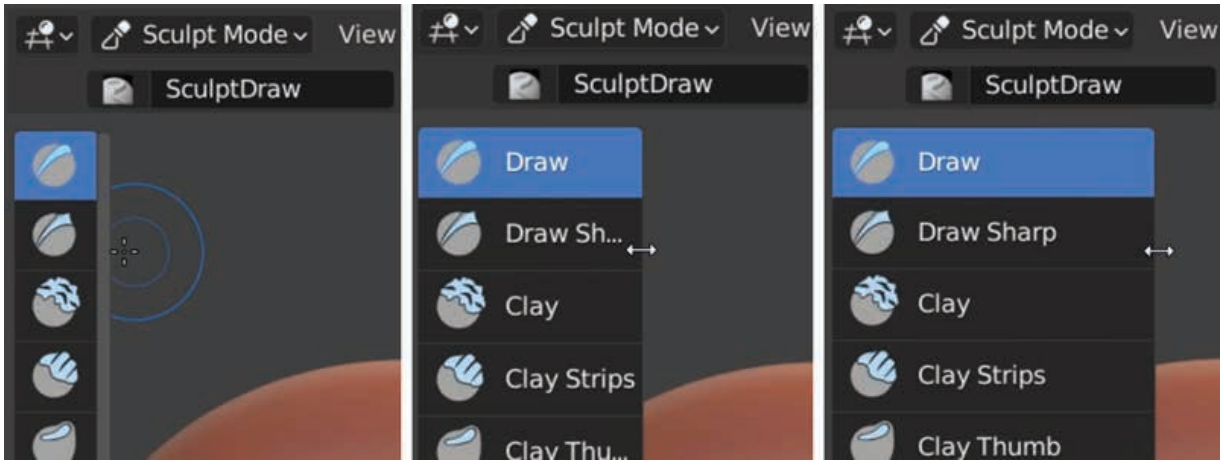
And I move the sphere up one unit by **pressing G+Z+1** on the numpad.

This isn't obligatory, just a personal preference. I don't like the floor grid to be visually in the way.

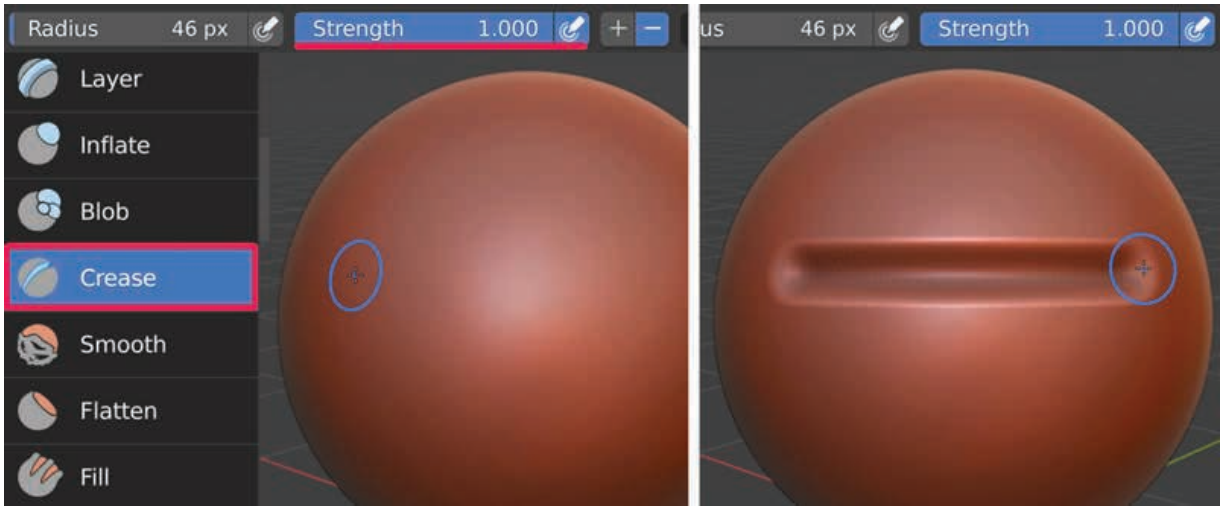
Of course you can also turn those elements off in the Overlay menu.



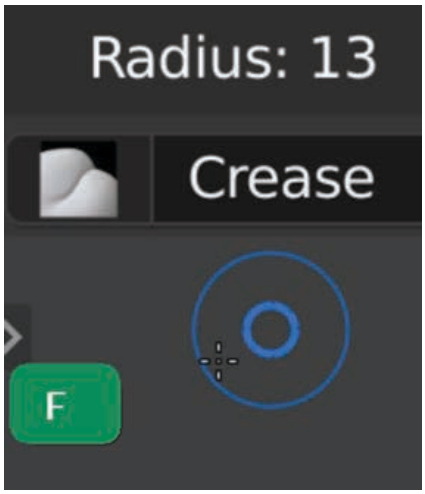
Then in Sculpt mode drag the tool bar to the right so we can see what all these icons are for.

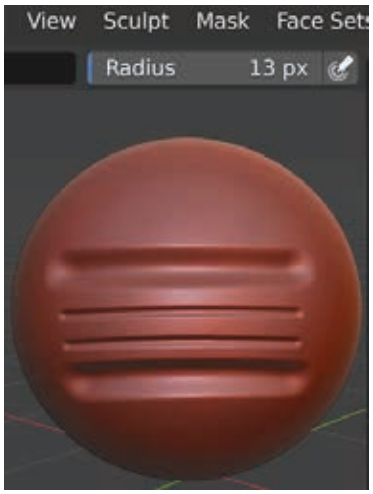


You could use the **Crease brush** with a high **Strength** value to cut some panel lines.

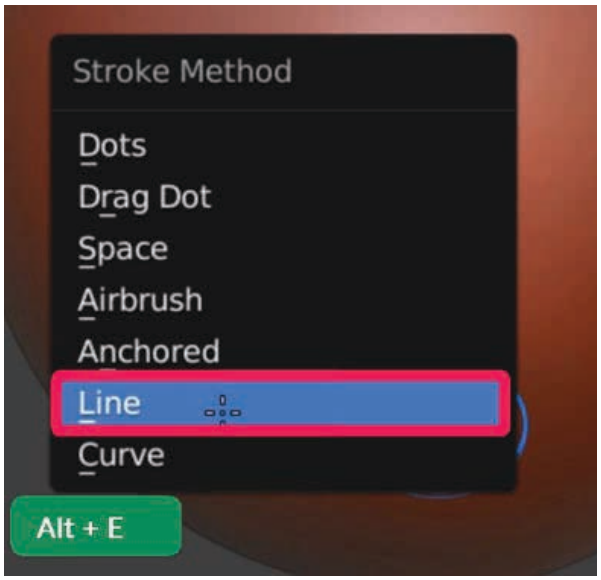


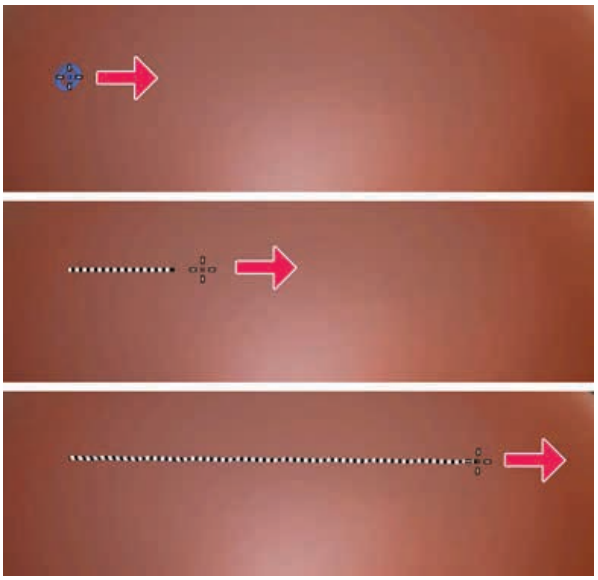
Press F to change the **Radius**, to make it thinner.



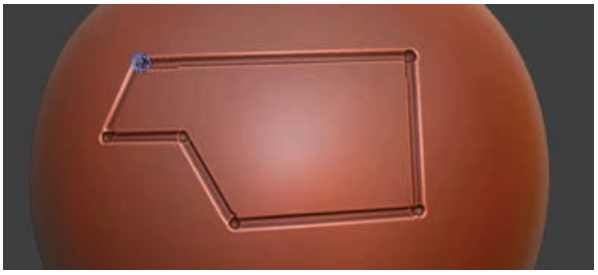


So we can draw lines freehand, but it won't look very good. **Press Alt+E** and choose **Line** from the Stroke Method menu.

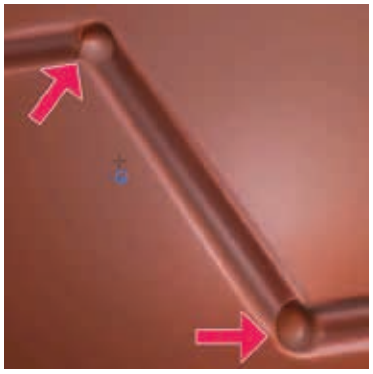




Now we can draw perfectly straight lines at last.

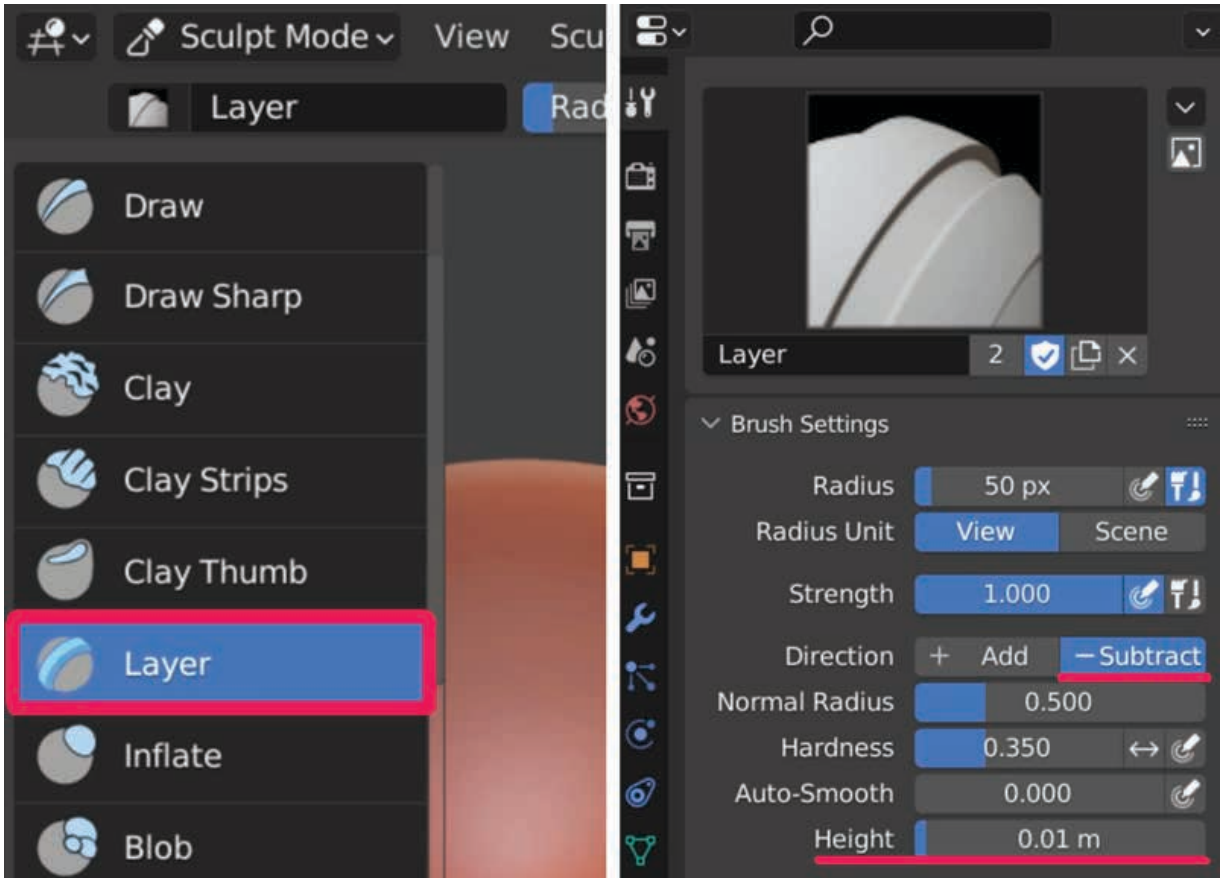


But you can see that where there is any overlap, the crease accumulates in depth. Unless you're just roughly sketching for a concept that will later be retopologized, that's not ideal.

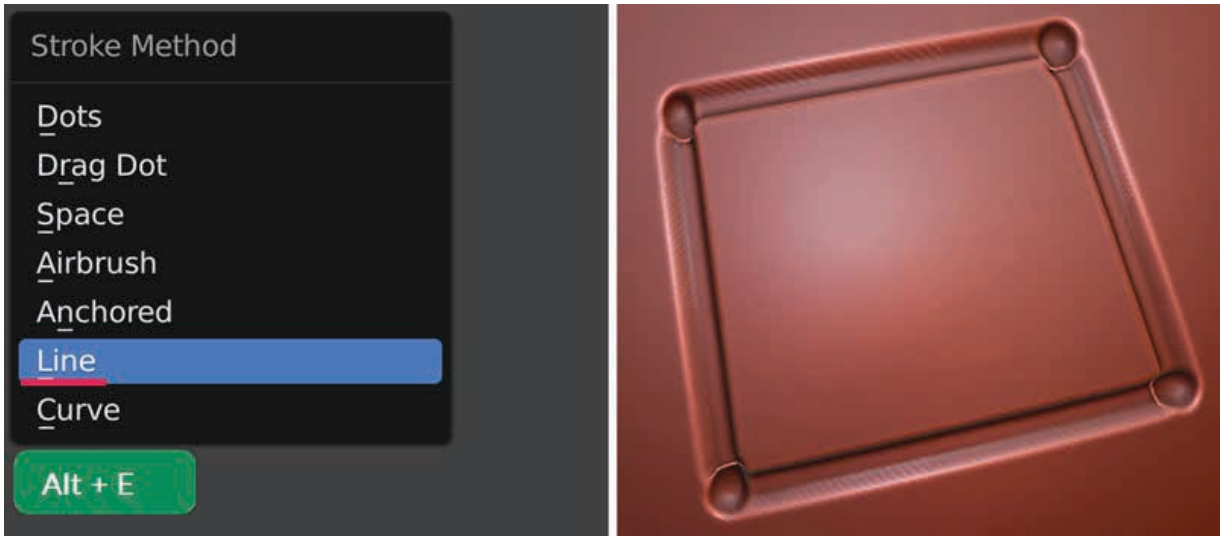


Click here to return to the index.

So instead of the Crease brush let's use the **Layer brush**. This brush is designed to be able to draw strokes that don't create ugly overlapping areas. Set it to **Subtract** so it cuts into the surface. Decrease the **Height.**

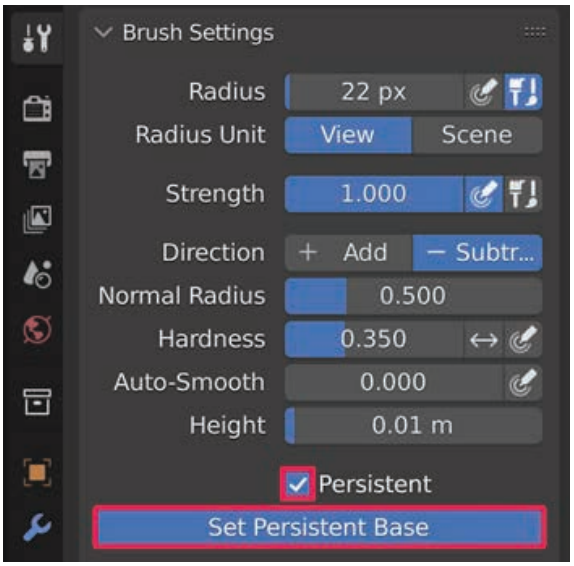


Now we have to **press Alt+E** again for the **Stroke Method menu** and choose **Line** again. As you can see it still accumulates the strokes.



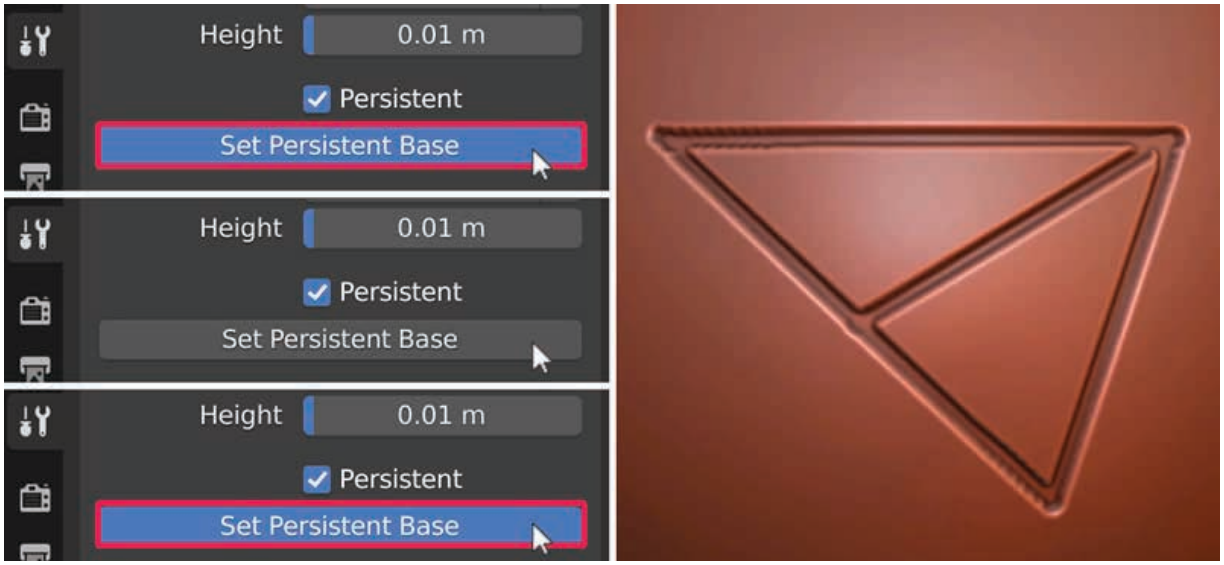
We need to check **Persistent** and click on **Set Persistent Base**.

And you have to do this *before* you draw the strokes.



Now we can draw nice straight panel cuts, and they won't accumulate.

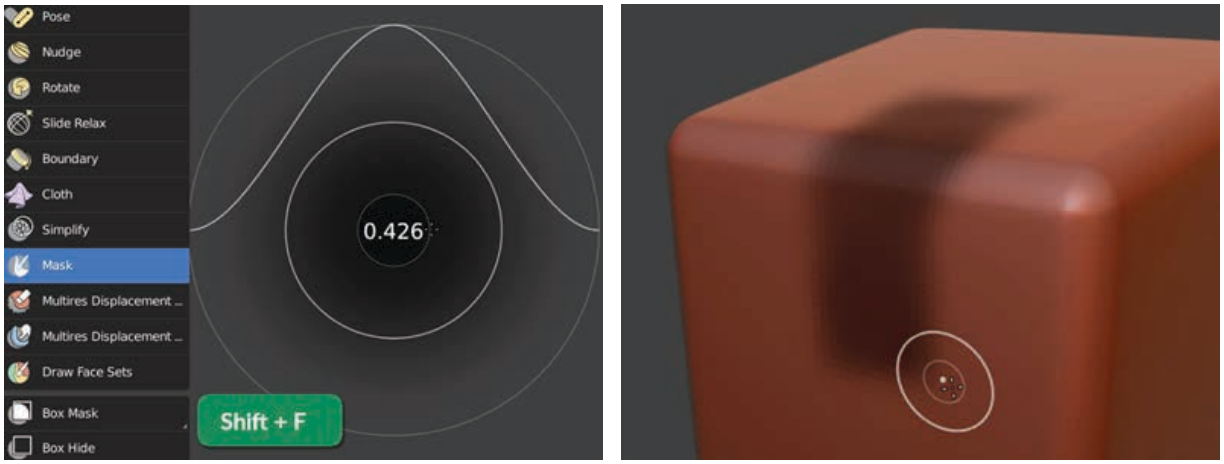
For some reason, sometimes clicking Set Persistent Base once is not enough. You need to **click it twice** for it to work. So if you find that it doesn't work in your case, just hit that Set Persistent Base button a couple of times instead of once.



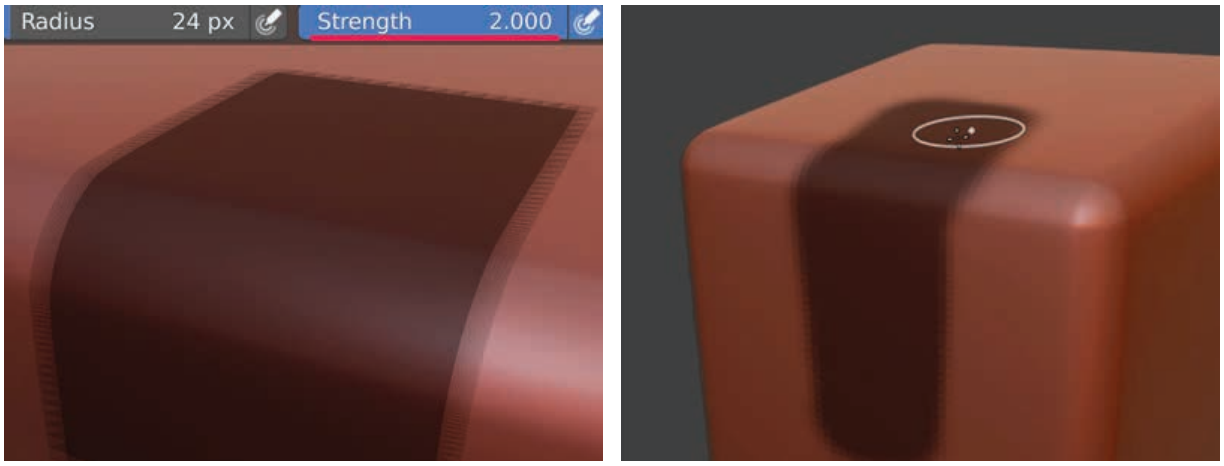
Click here to return to the index.

Extruding in Sculpt Mode

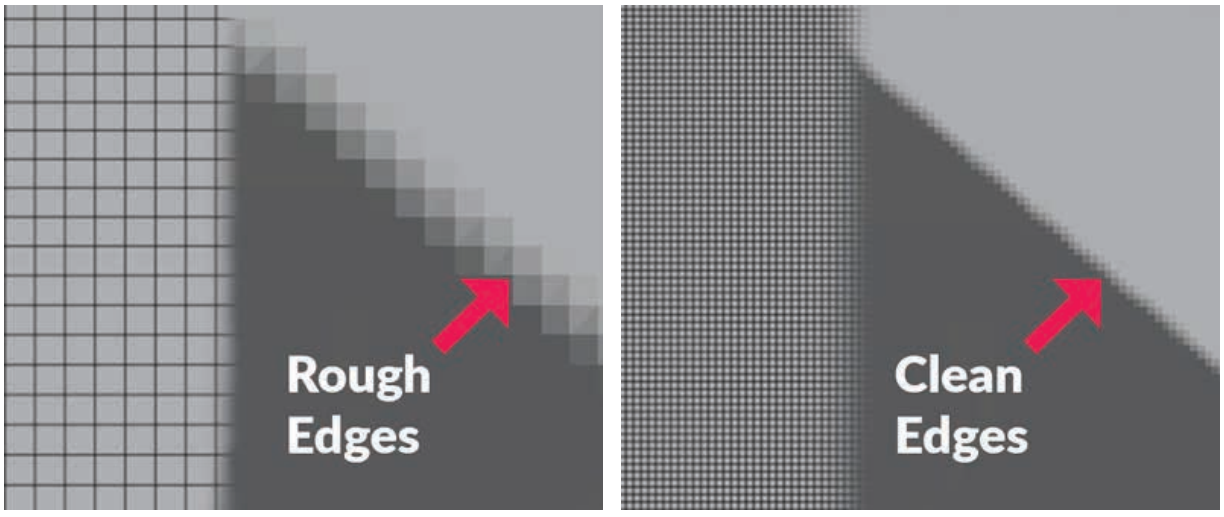
Draw a mask. You can do this manually using the **Mask Brush**.



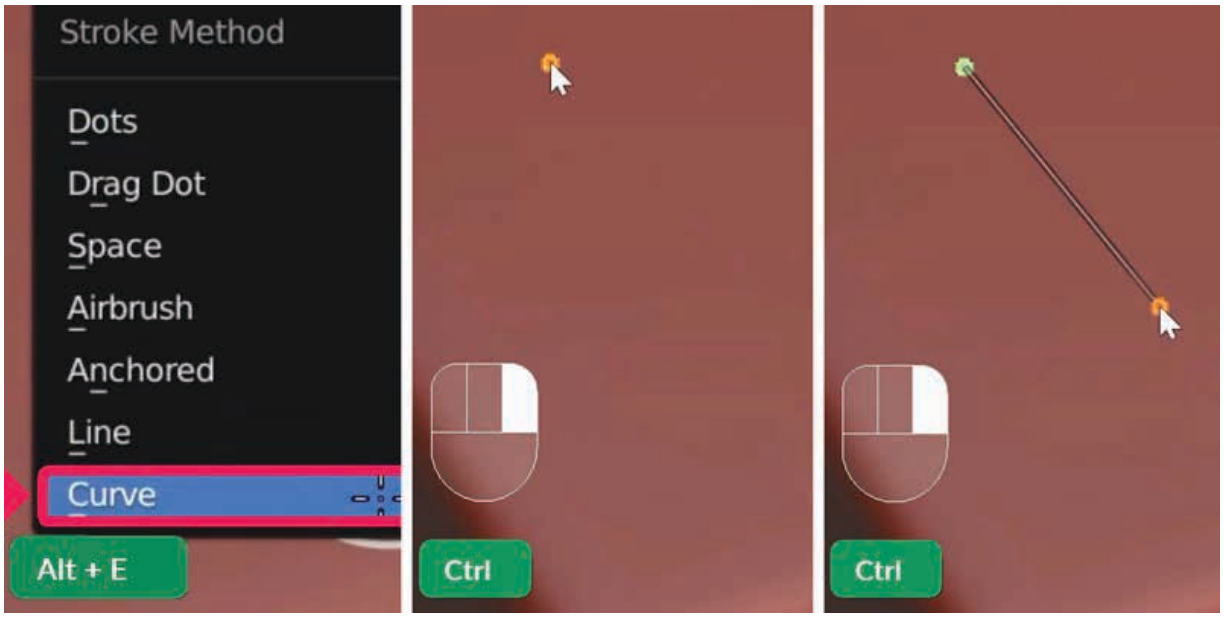
Press **Shift+F** to increase the Strength. If you want you can go as strong as 2, but you'll have to type that value in the Strength field manually. The stronger the value, the crisper the edges of the mask will be.



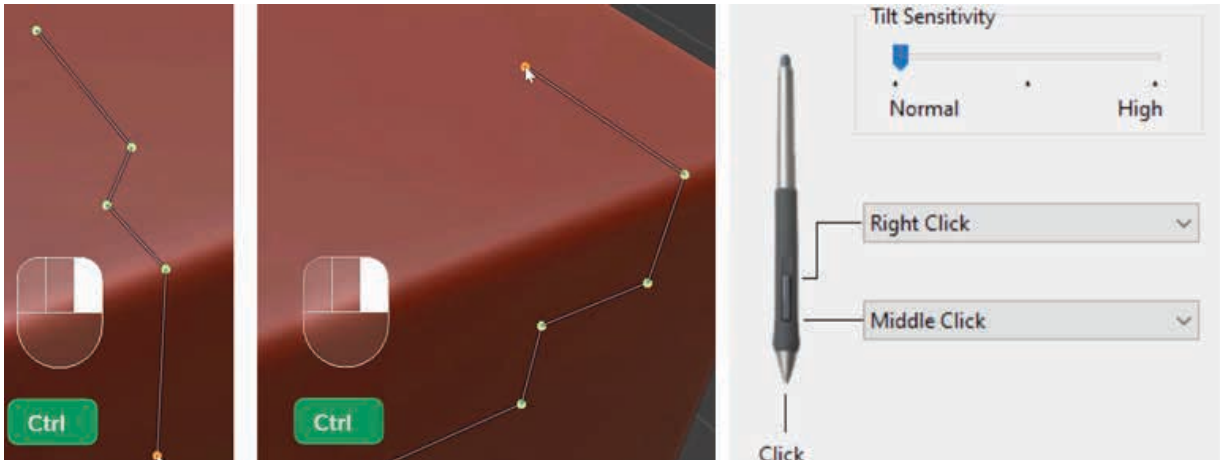
The resolution of your mesh is also an important factor for the mask quality.



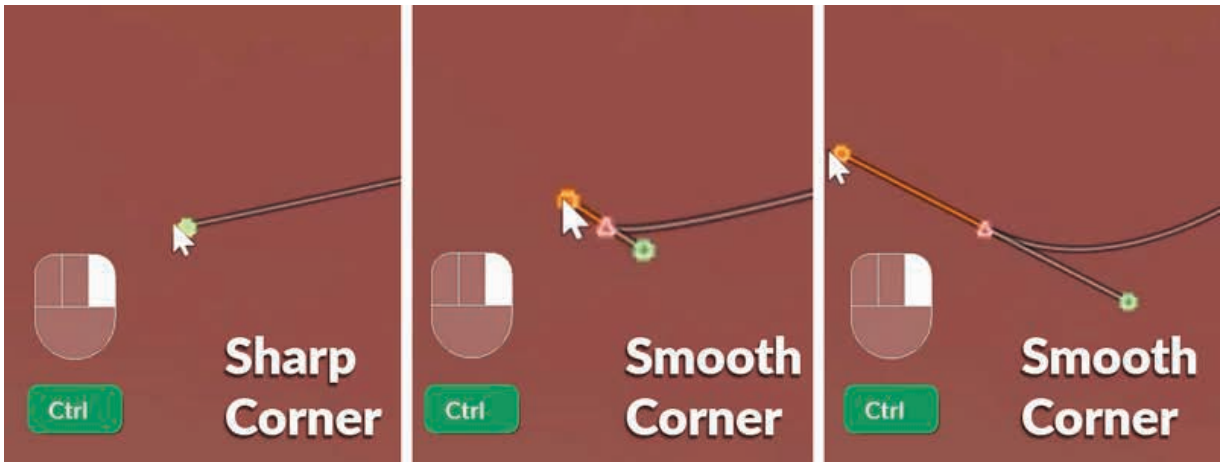
When you **press Alt+E** you can choose a different **Stroke Method**. For example **Curve**.



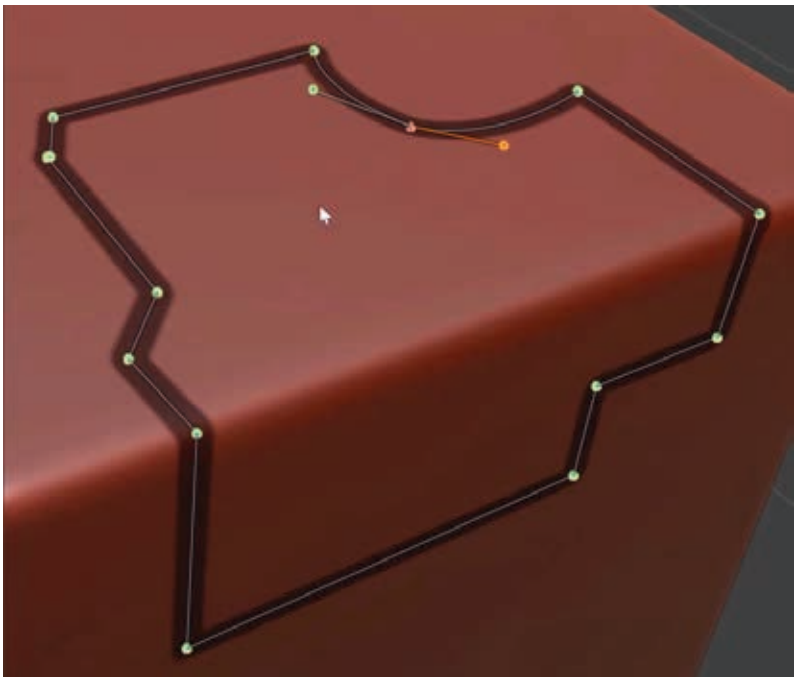
Which you can then use by **holding Ctrl and right-clicking** to draw a curve. Oddly enough, this is easier when using a mouse than when using a tablet pen. But, if you set one of the buttons on the pen to be right-click, you can also do it that way.



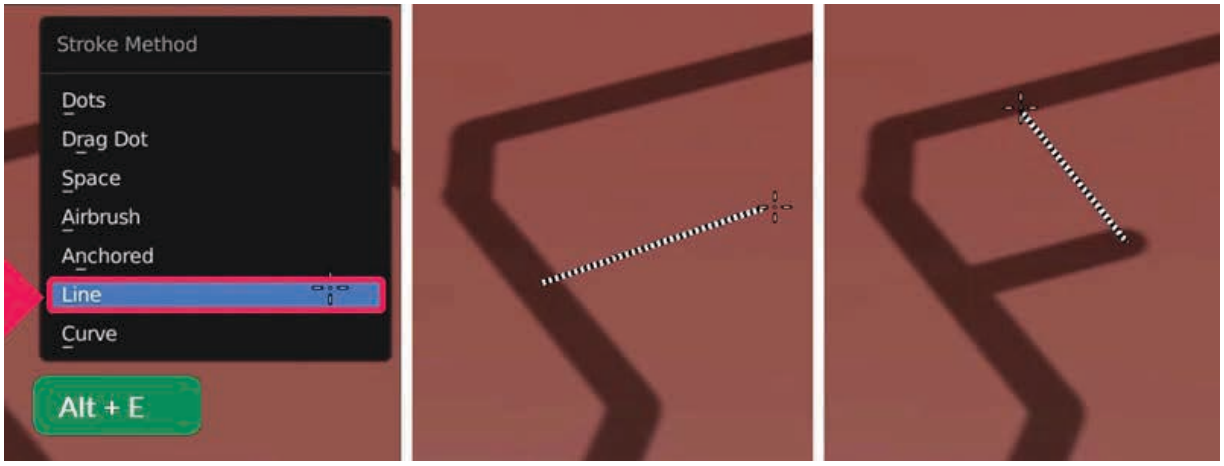
Click for a sharp corner and click-and-drag for a smooth corner with handles.



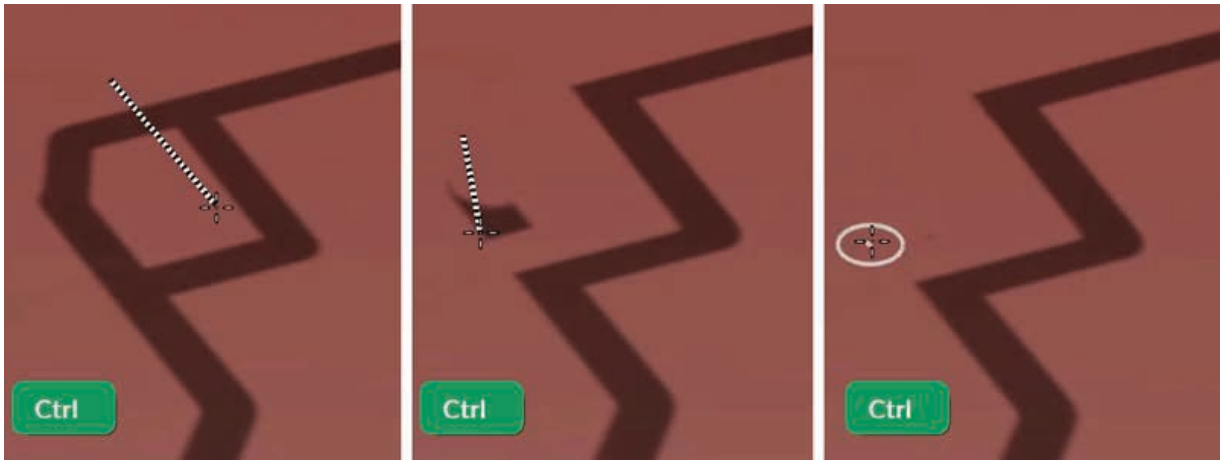
Then while still **holding Ctrl**, **left-click** (or hit **Enter**) to draw the stroke along that curve.



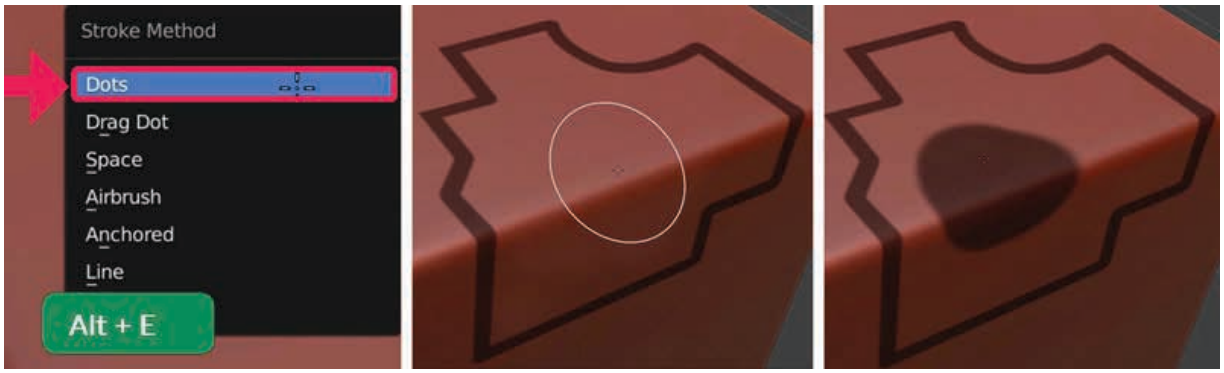
You can use both the **Curve** and the **Line** Stroke Method to draw nice geometric masks.



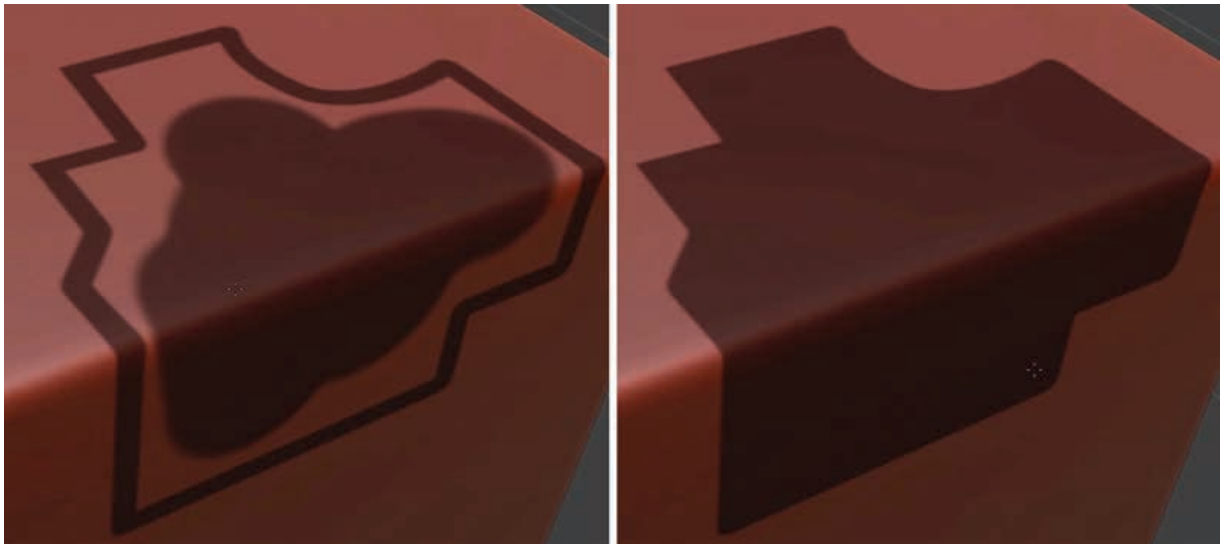
Hold Ctrl if you want to subtract from the mask using **Line** stroke method. Holding Ctrl inverts most brushes.



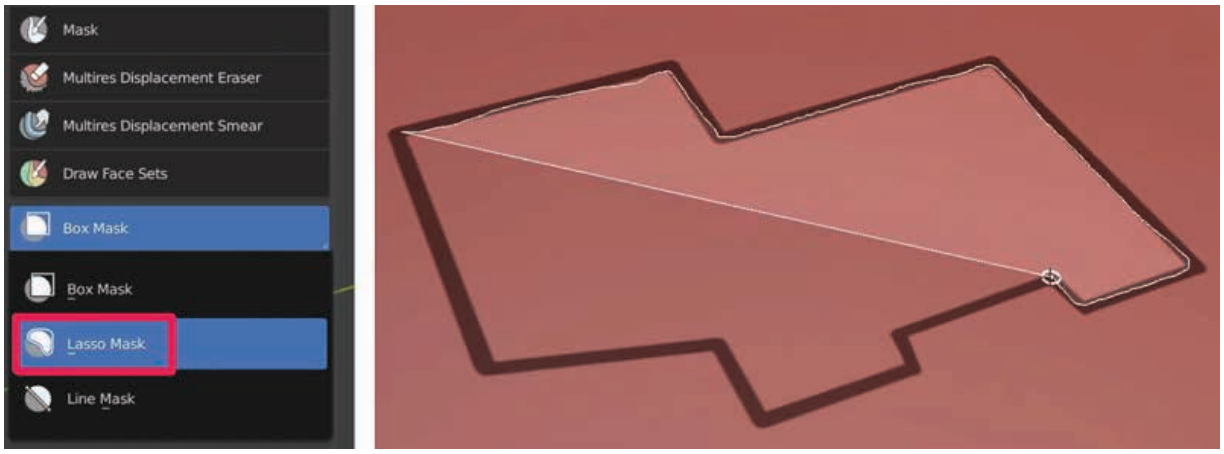
You can then use the **Dots** Stroke Method again to fill the mask. Dots is also the default stroke method if you don't change it.



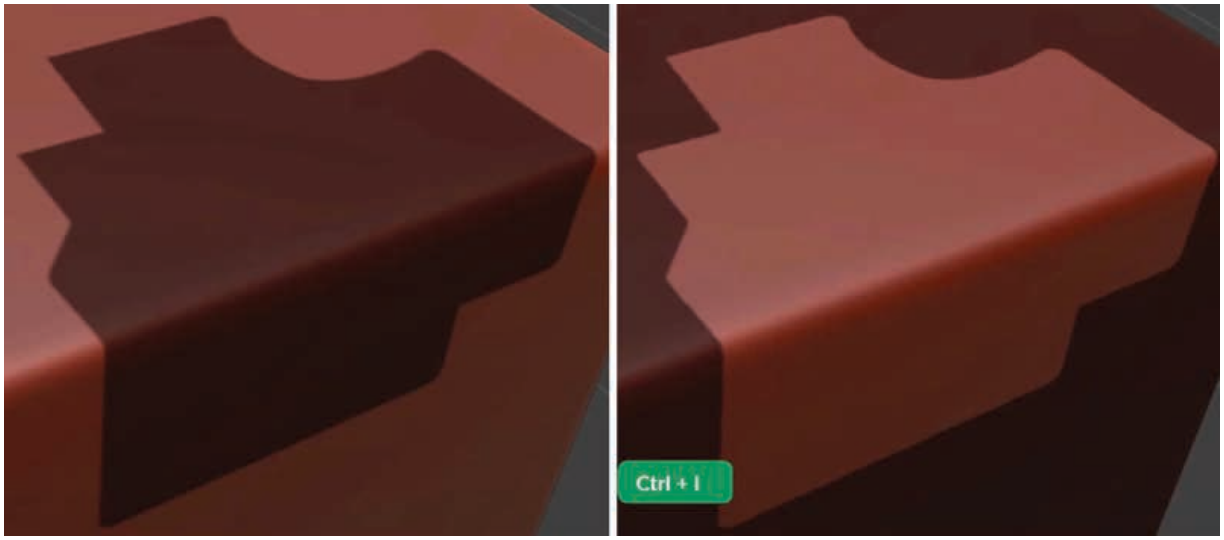
This allows you to quickly fill up the mask. Unfortunately there's no "fill" function yet that would do this even faster. Maybe some day!



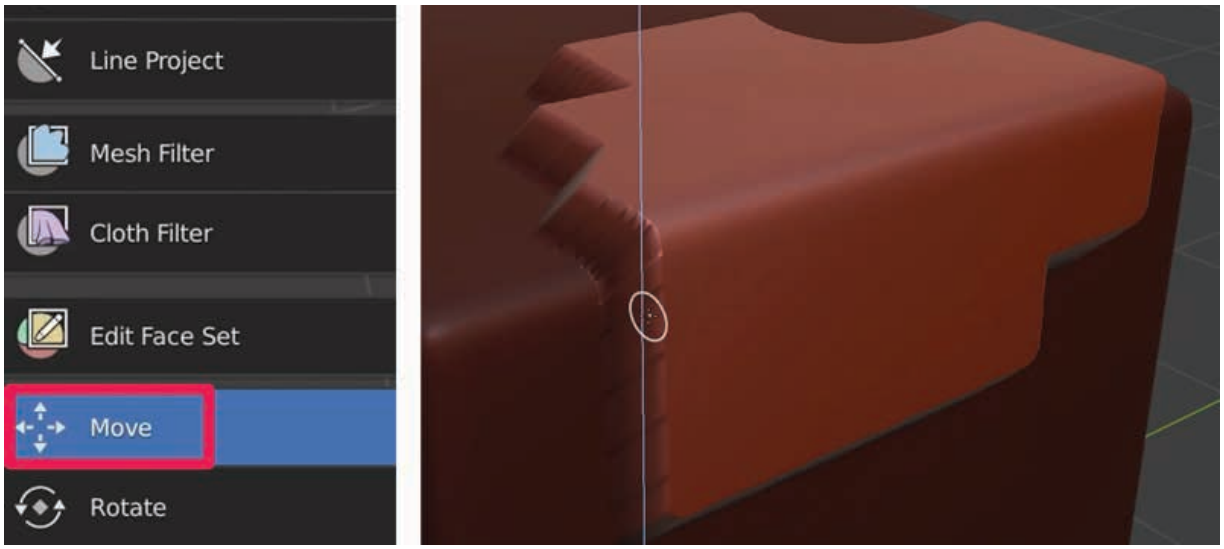
Or, you can use the **Lasso Mask**, which you can find in the Tool panel in the submenu of the Box Mask tool.



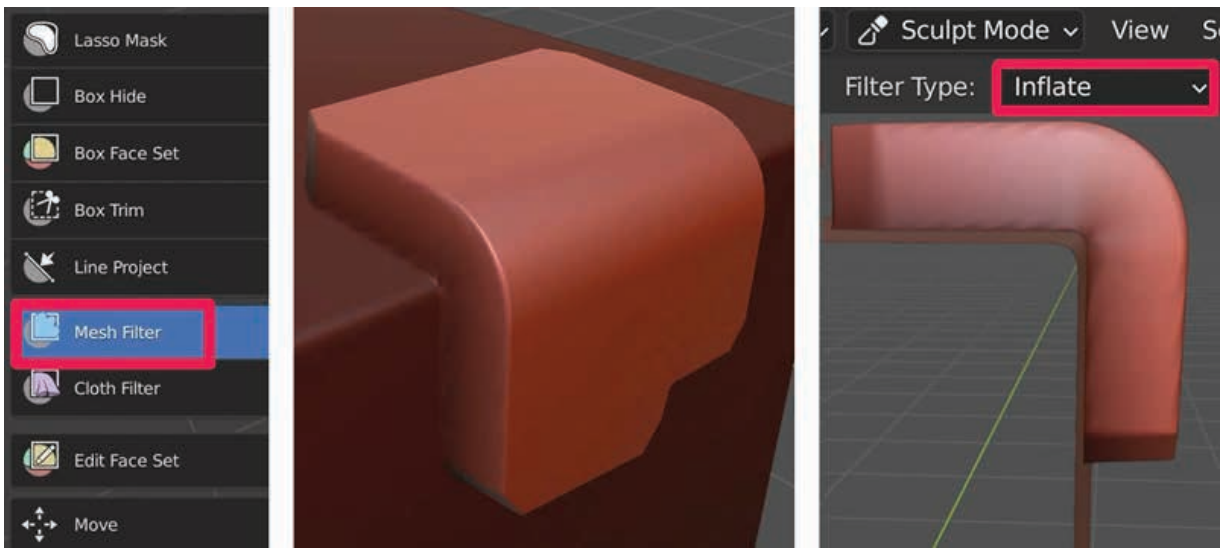
Press Ctrl+i to invert the mask.



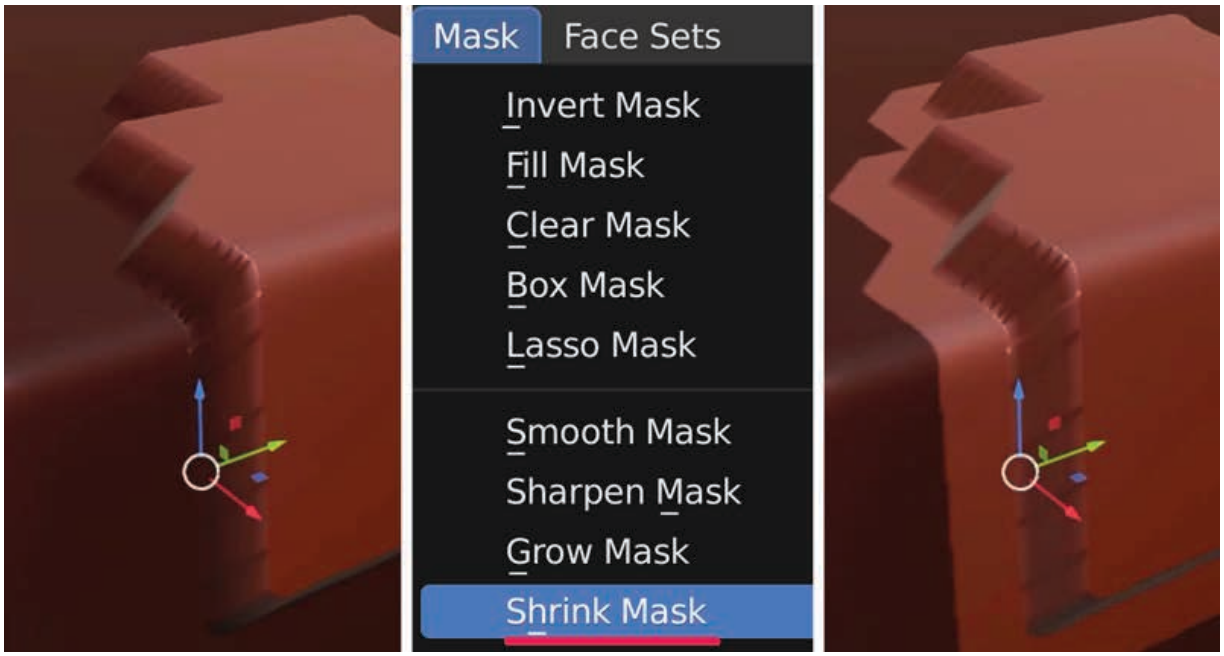
Then you can Move the unmasked part to extrude it using the **Move tool**.



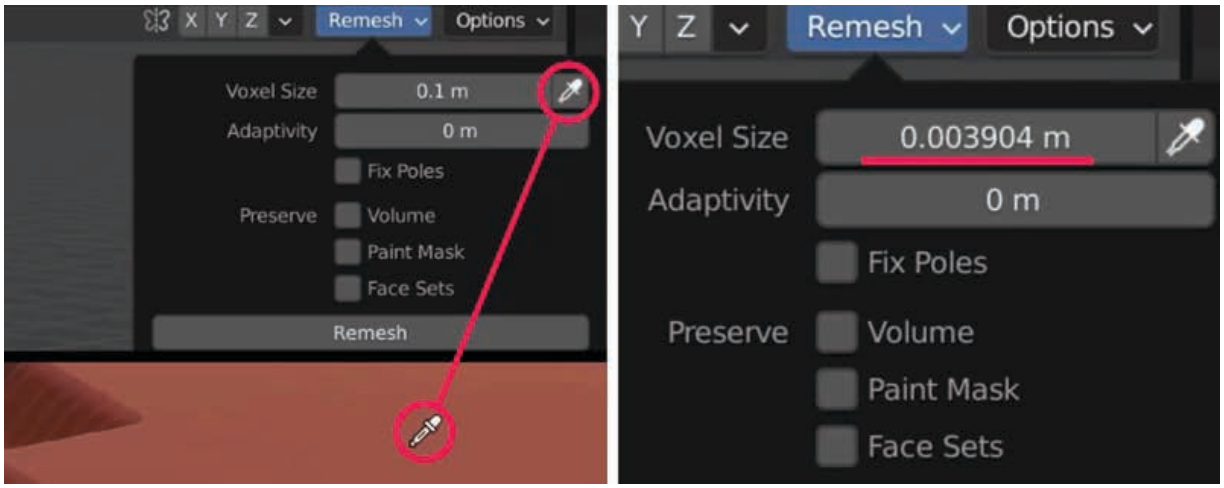
Or, you can also use the **Mesh Filter** set to **Inflate**. This gives a different kind of effect, like extruding along Face Normals.



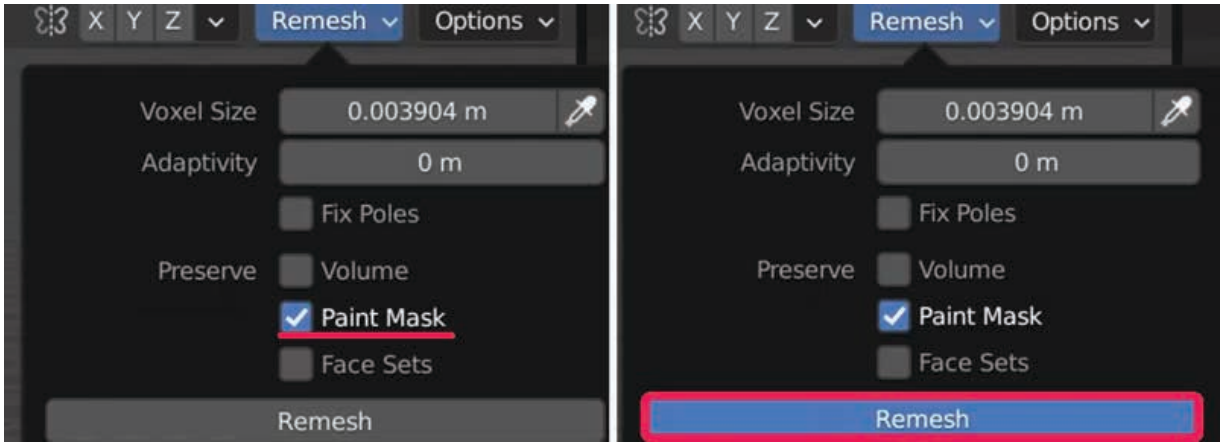
To make the extruded part smoother and remove the artifacts at the sides, first go to **Mask > Shrink Mask**.



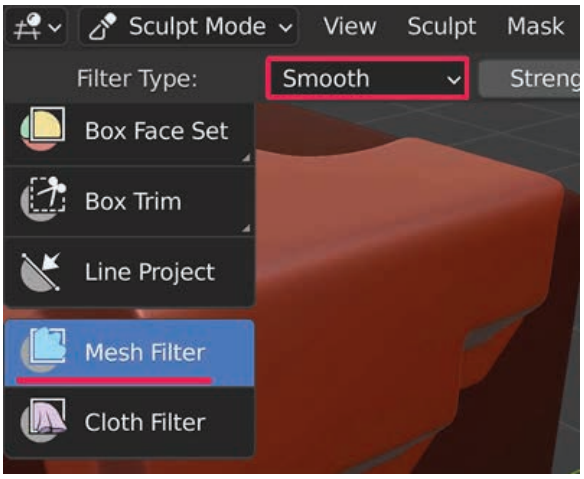
Repeat if necessary. Sample the mesh resolution of the unmasked, unstretched part in the **Remesh** menu.



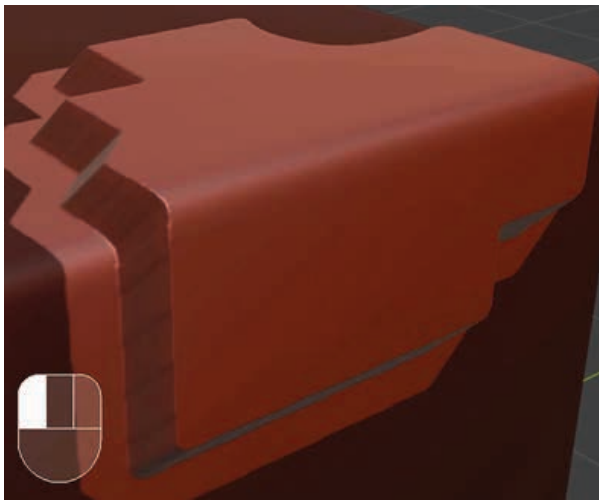
Check the **Paint Mask** option to make sure you can remesh without losing the mask. Then click **Remesh**.

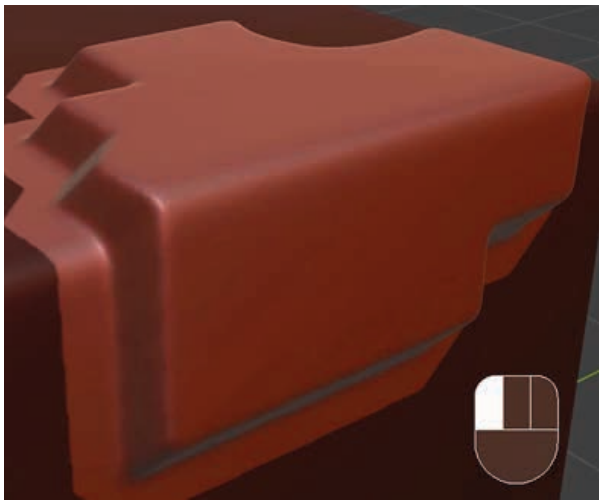


Choose the **Mesh Filter** and set it to **Smooth**.

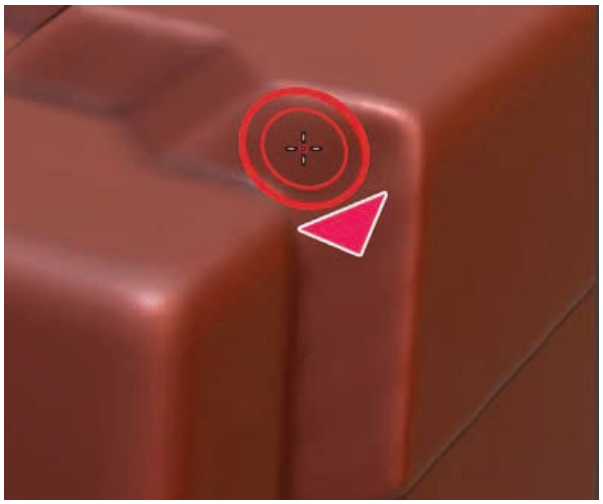


Left-click and drag the mouse cursor to the right for the desired smoothness. Notice that the stretching artefacts on the side of the extruded part get smoothed out.



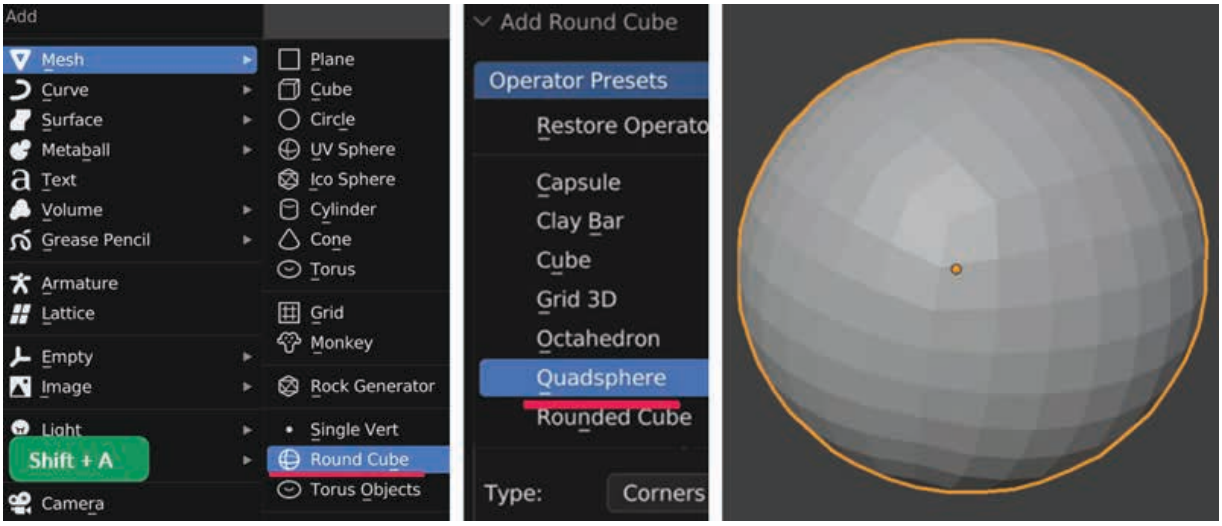


Press Alt+M to clear the mask. You can use the **Smooth brush** for additional smoothing of the sides.

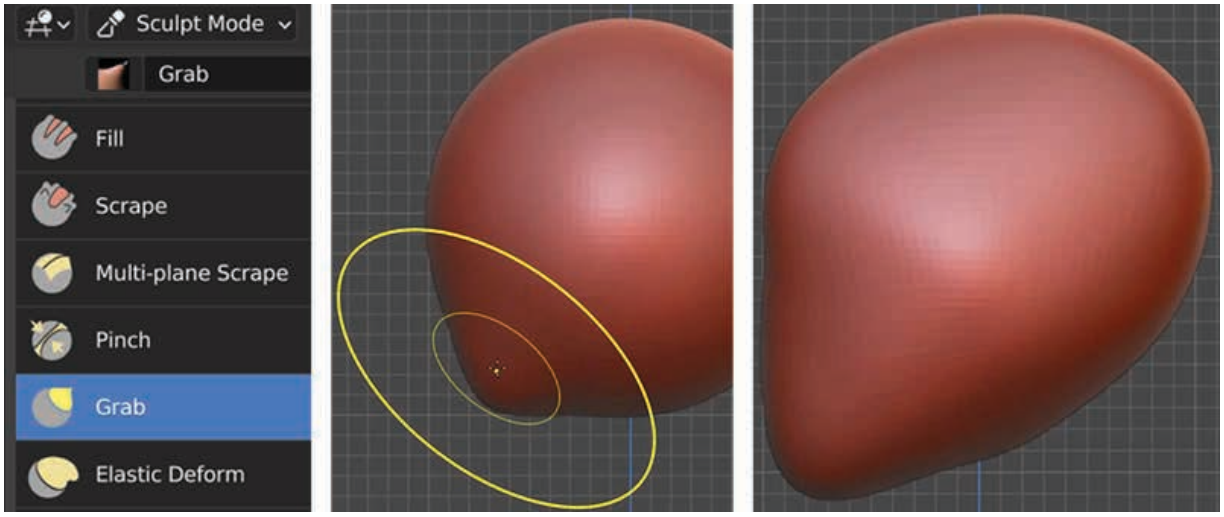


Inflate Mesh Filter

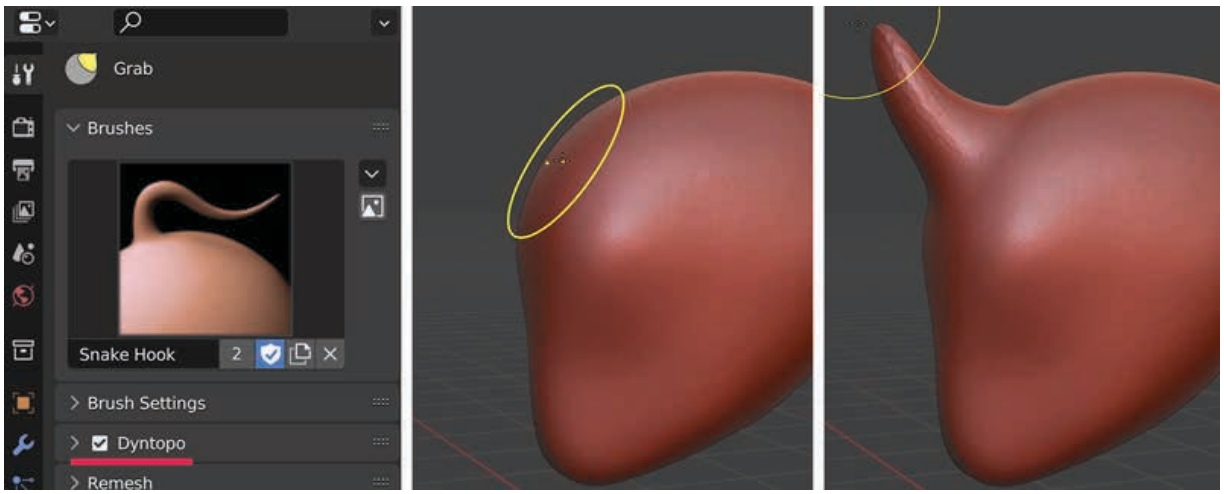
To create the base mesh I add a **Roundcube**, with the **Quadsphere** preset. Note that this option is added by enabling the Extra Objects add-on in Preferences.



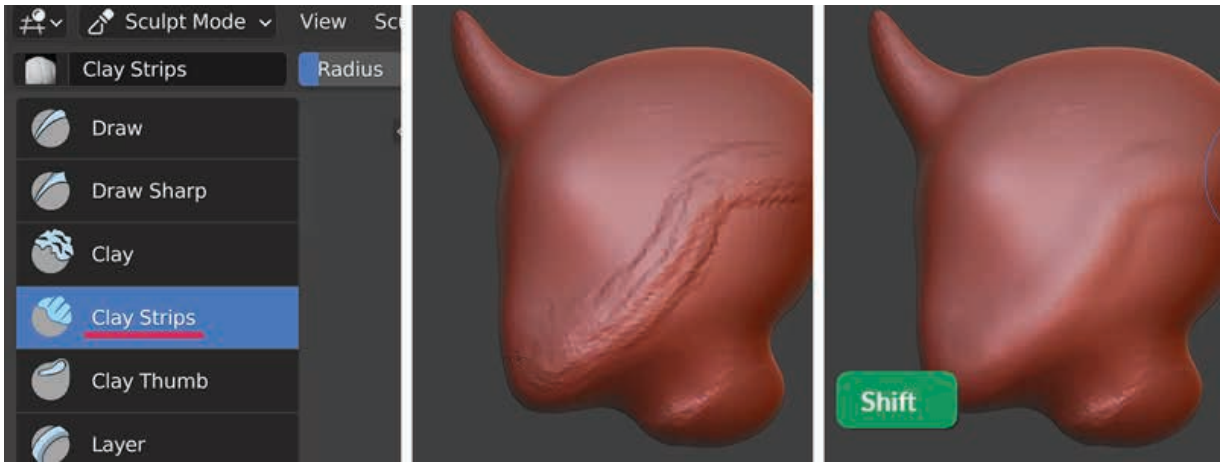
Using the **Grab brush** I push and pull it into shape.



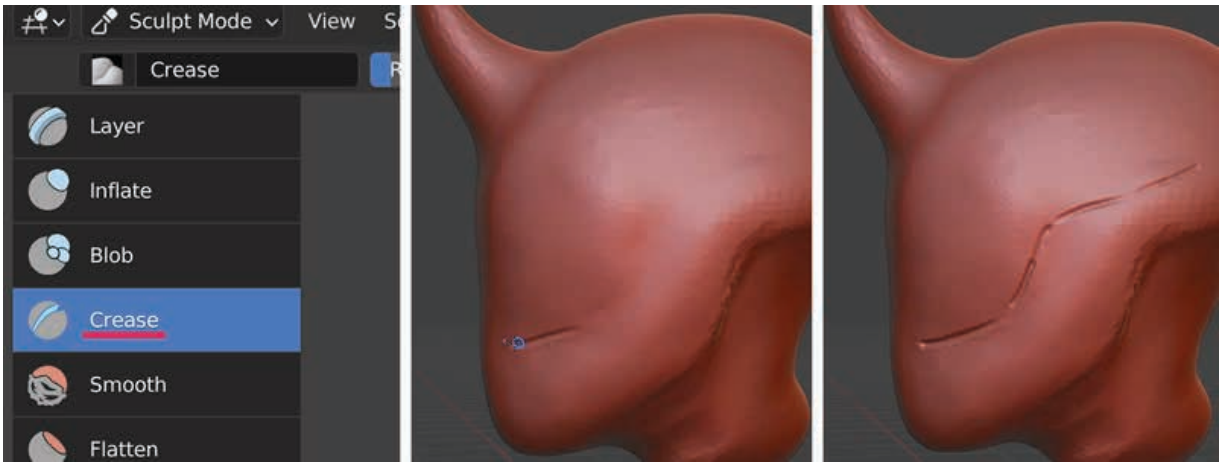
With **Dyntopo** enabled, the **Snake Hook** brush allows me to pull out geometry.



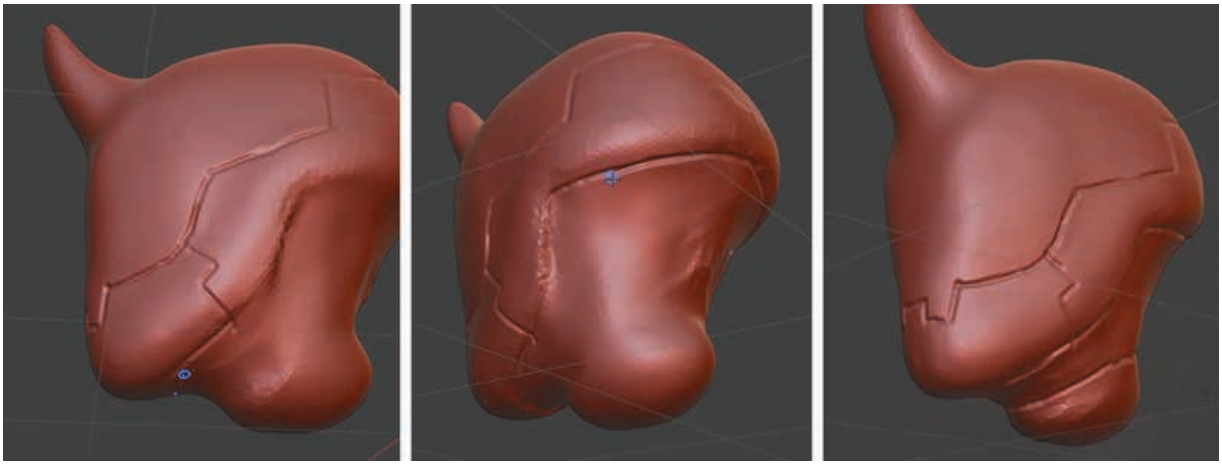
The **Clay Strip brush** is useful for adding and removing volume. Which can then be smoothed out by **holding Shift**.



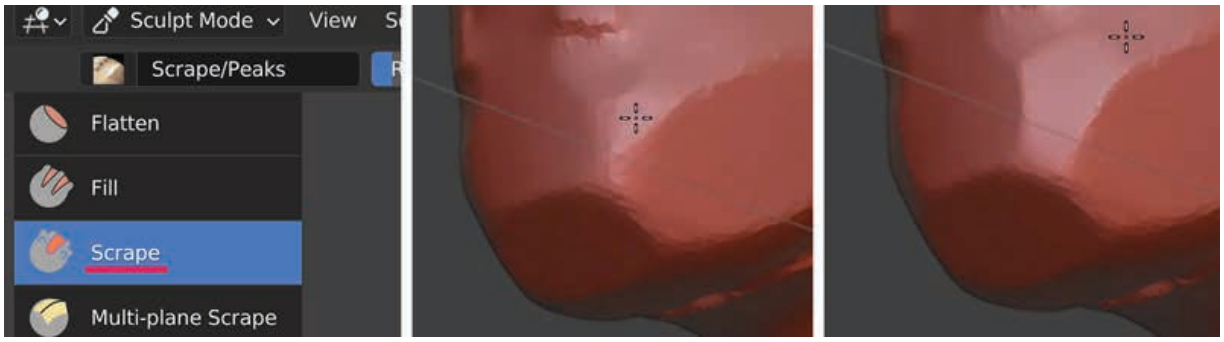
Using the **Crease Brush** you can sketch out some ideas for refining the shape on the geometry.



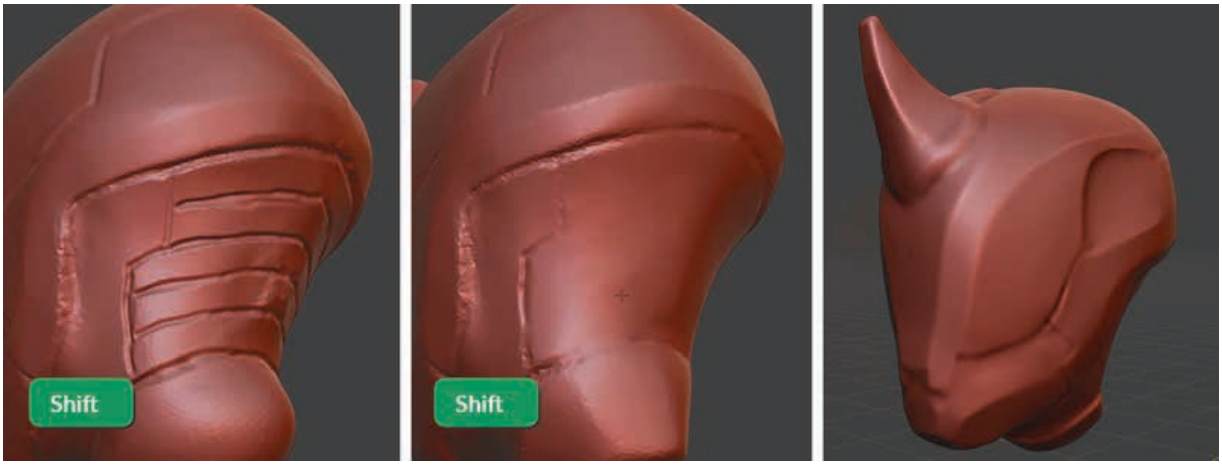
Don't worry about making any of this perfect, the main objective is to sketch out ideas.



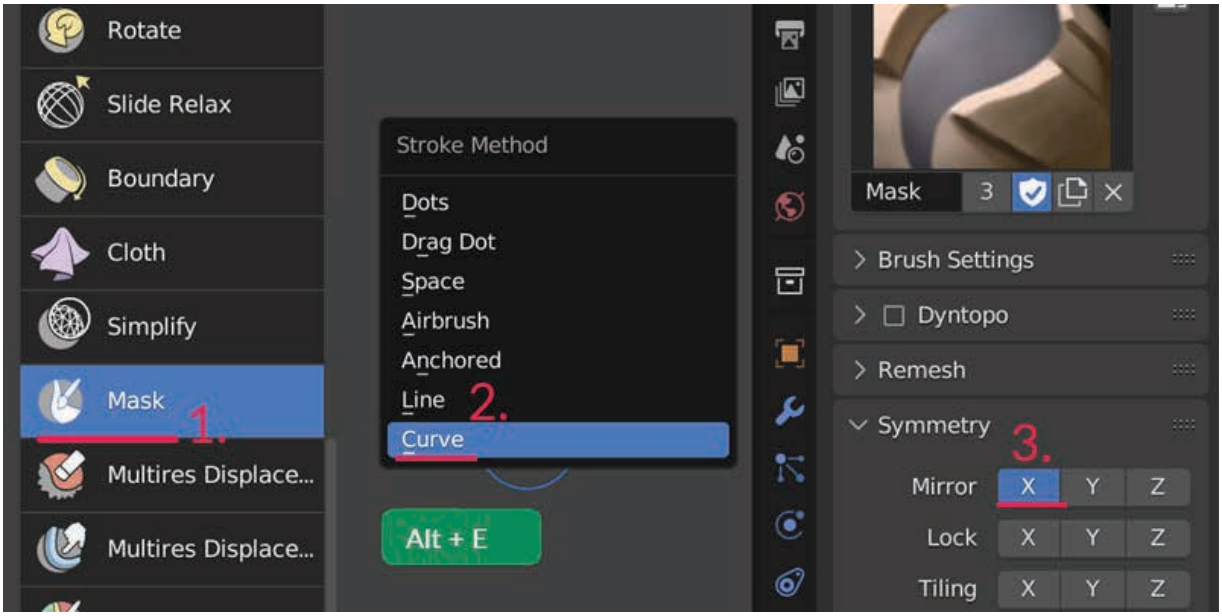
The **Scrape Brush** is good for creating more angular and flattened hard-surface shapes.



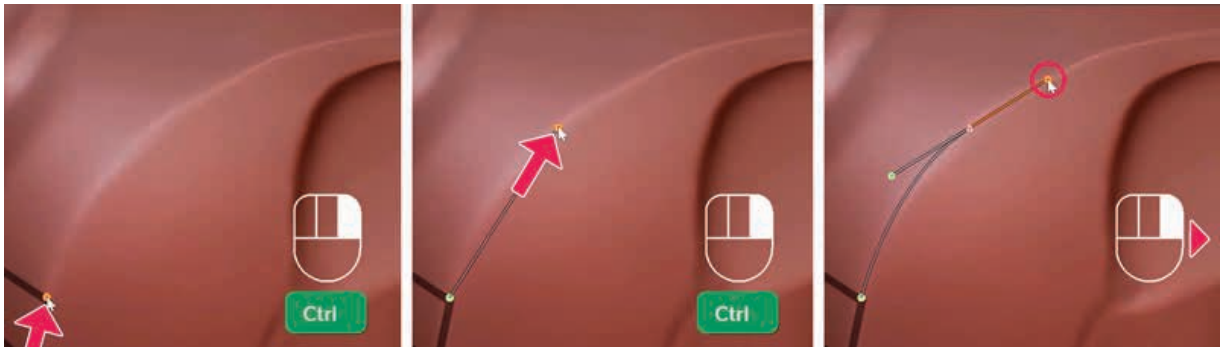
Areas you don't like can just be erased with the **Smooth brush**.



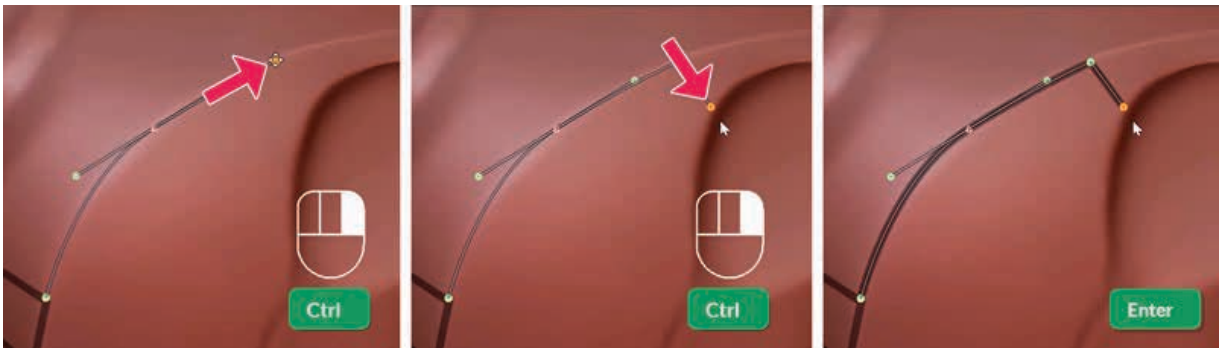
Create some panel cuts using the **Mask Brush**. Using the **Curve stroke method**, we can keep them nice and angular. Make sure you have **Symmetry** enabled so you get both sides.



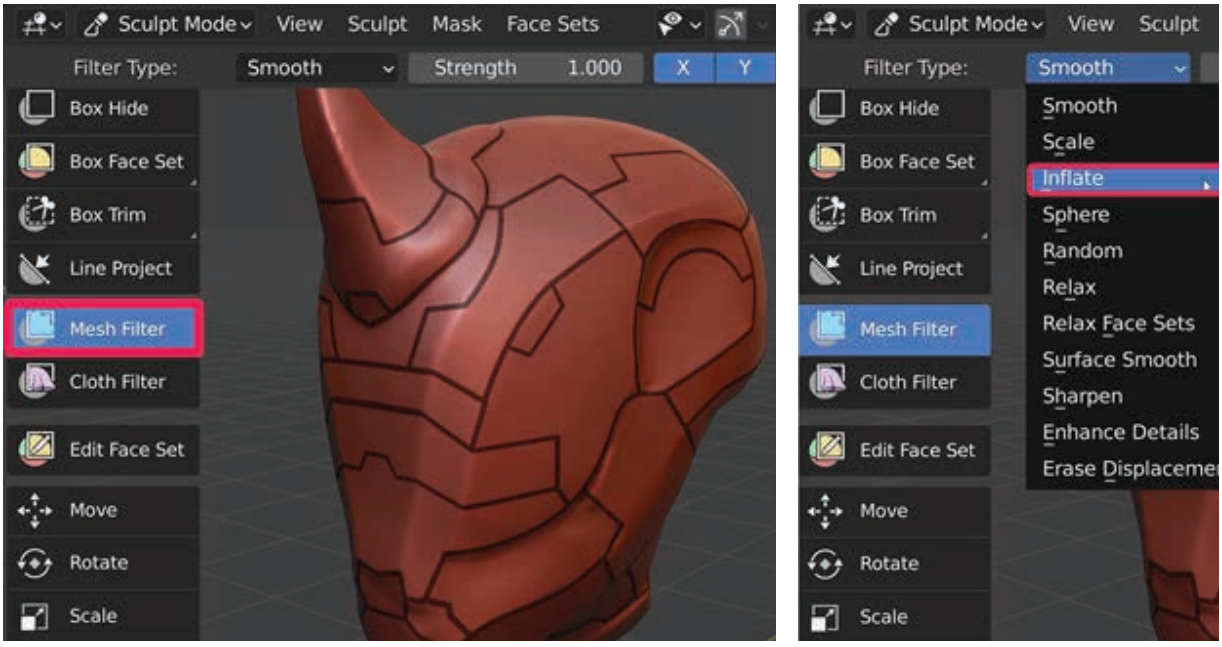
Hold Ctrl and right-click to create a hard corner on the curve, or hold Ctrl and right-click and **drag** to create a bezier handle.



Hold Ctrl and left-click (or press Enter) to create the mask according to the curve.

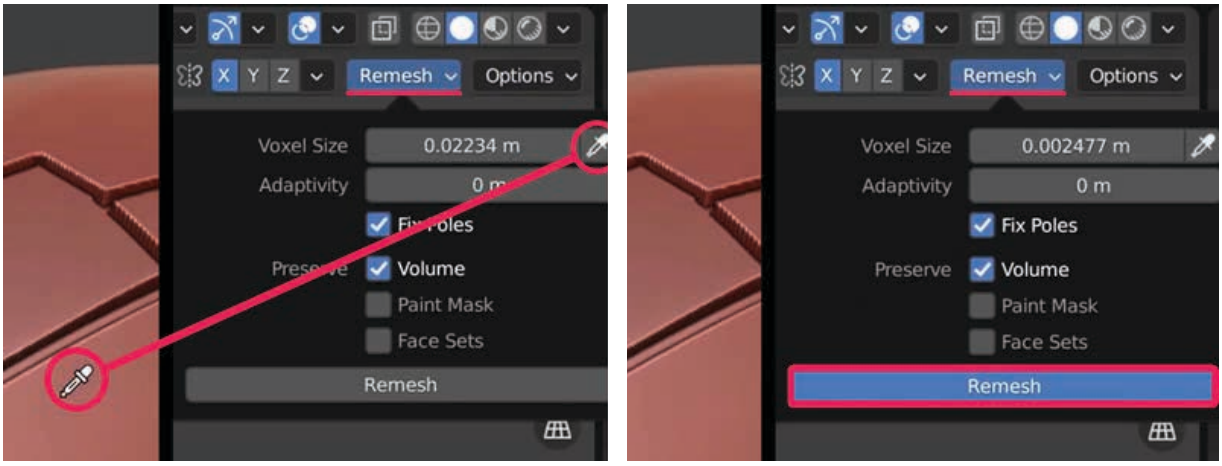


Select the **Mesh Filter** in the Tool panel. Set Filter Type to **Inflate**.

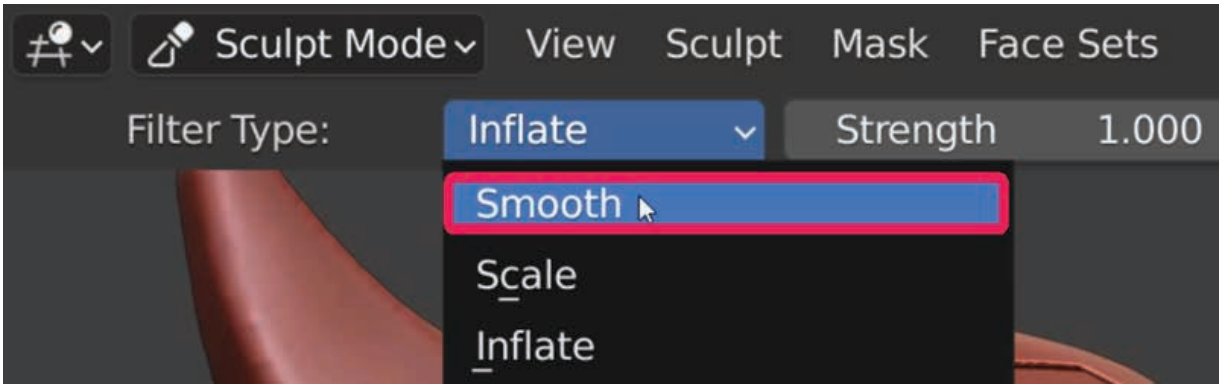


Drag the mouse cursor to the right to inflate. Then **press Alt+M** to clear the mask.

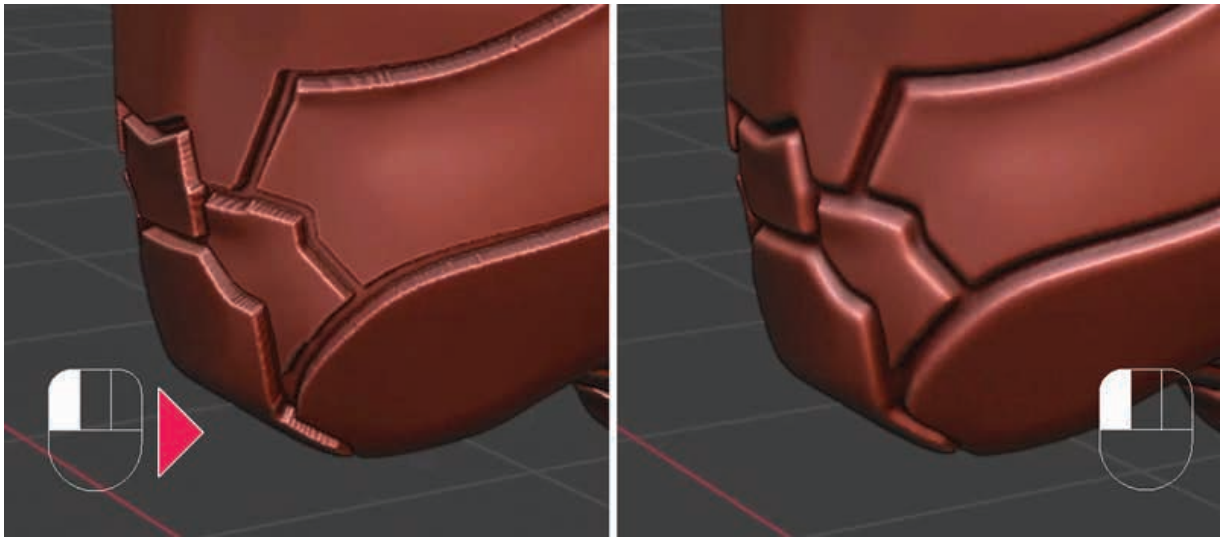
It's a good idea to remesh at this point. So sample the geometry. And then click on remesh, and wait for a moment.



Select the **Mesh Filter** tool. Then, set Filter Type to **Smooth**.

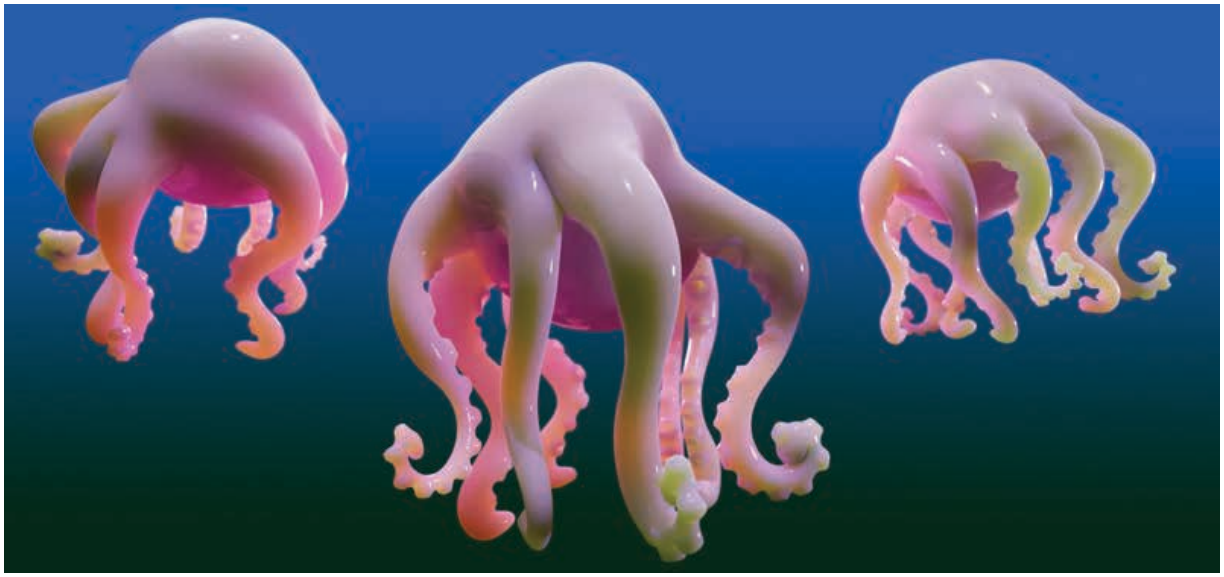


Finally, drag the mouse cursor to the right to smooth the mesh.

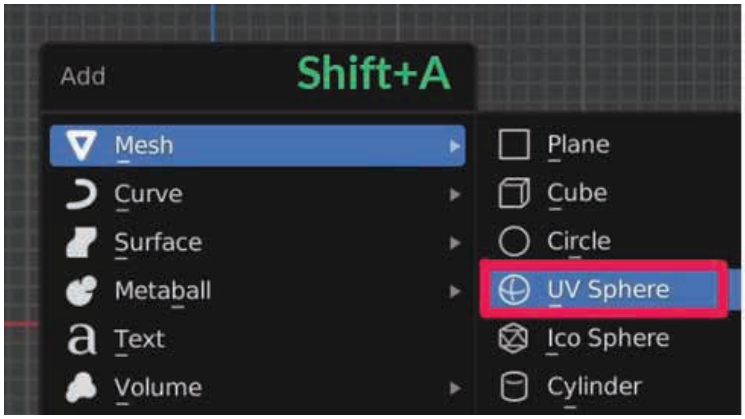


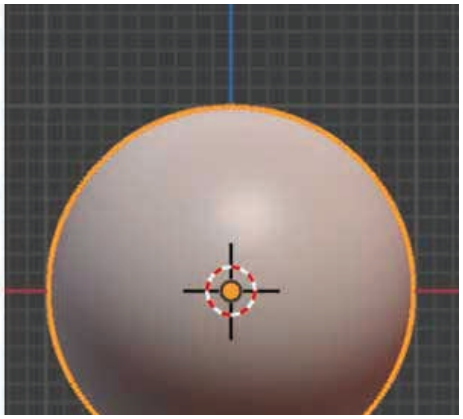
Sculpting Tentacles with Snake Hook and Curve Stroke Method

Don't have a Graphic tablet to sculpt with? No problem - using this method you can sculpt things like these tentacles with just your mouse. It takes advantage of the Curve Stroke Method.

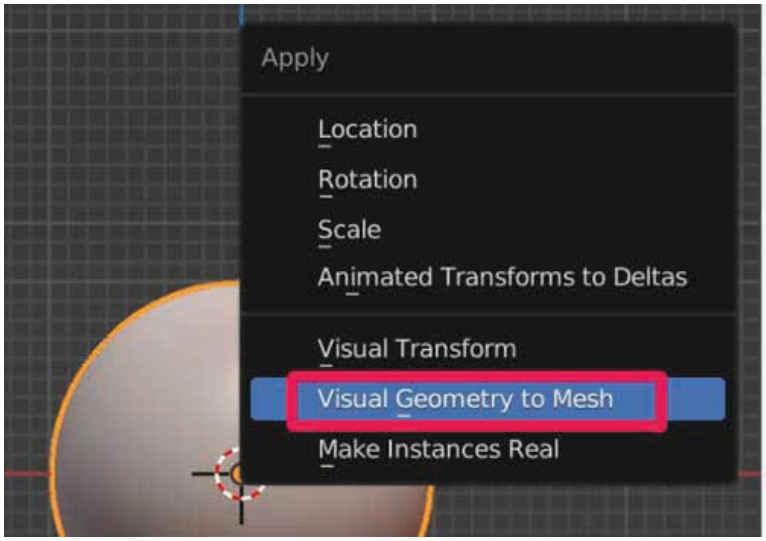


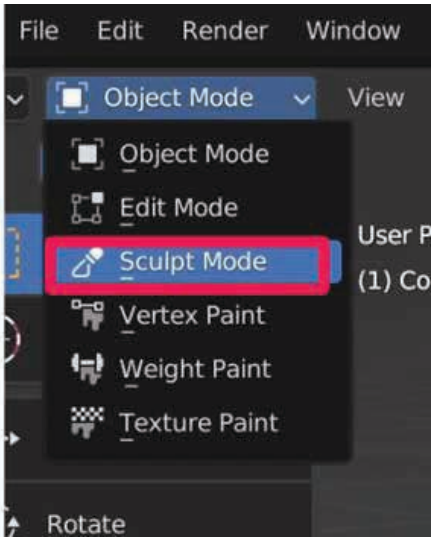
Add a **UV Sphere** and **press Ctrl+2** to add a subdivision modifier.





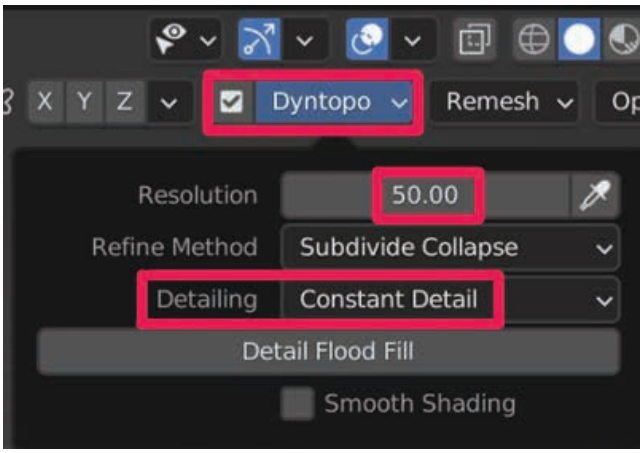
Press Ctrl+A and choose **Visual Geometry to Mesh** to apply the modifier. Switch to **Sculpt Mode**.



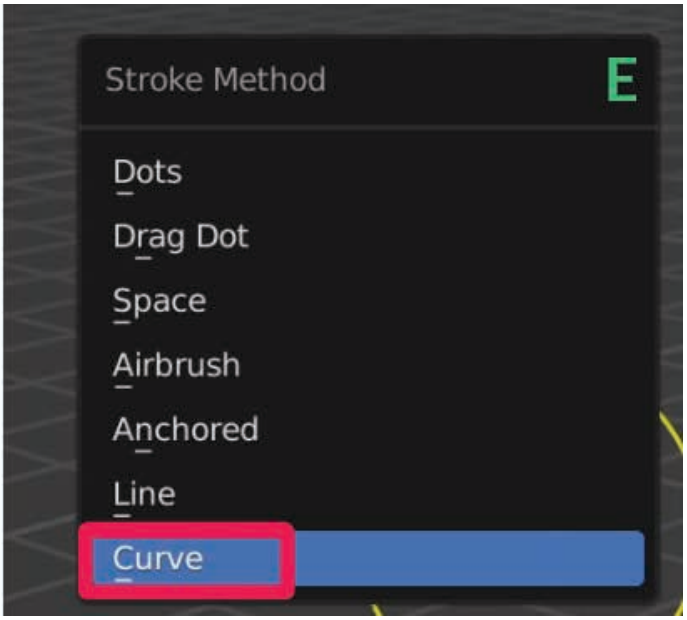


Enable **Dyntopo**. Set it to **Constant Detail** and increase the value.

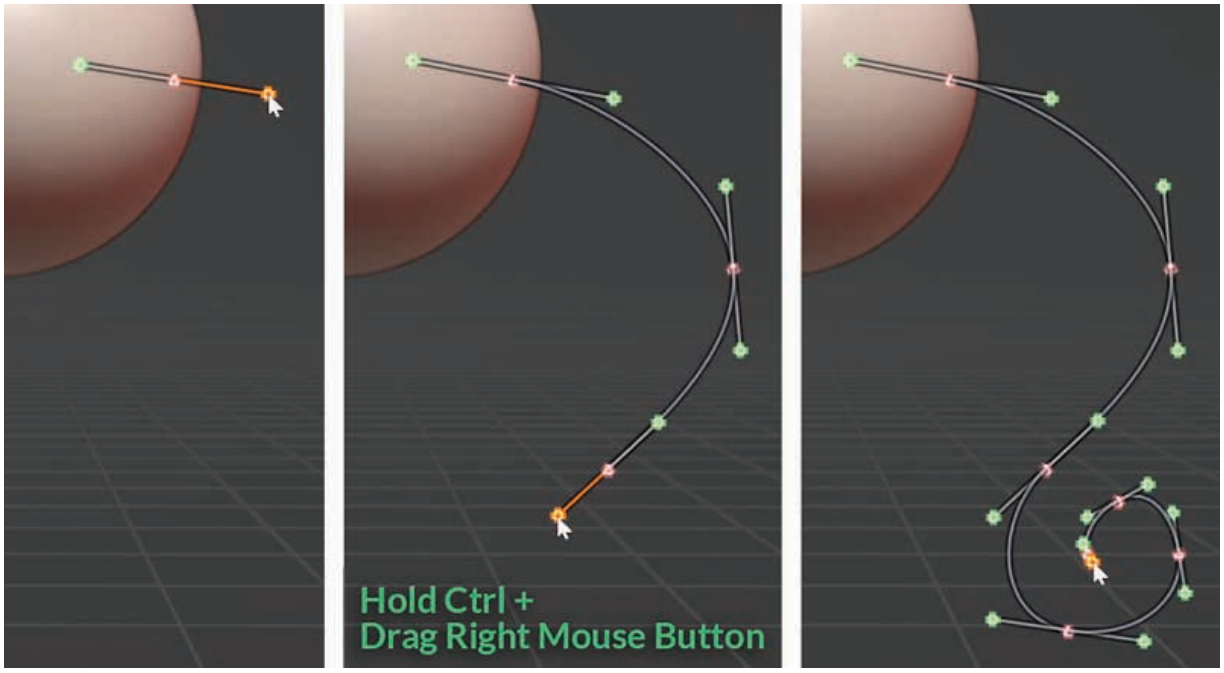
I recommend at least 20, but personally I prefer to go even a little higher like 50, as it makes the masking later more detailed.



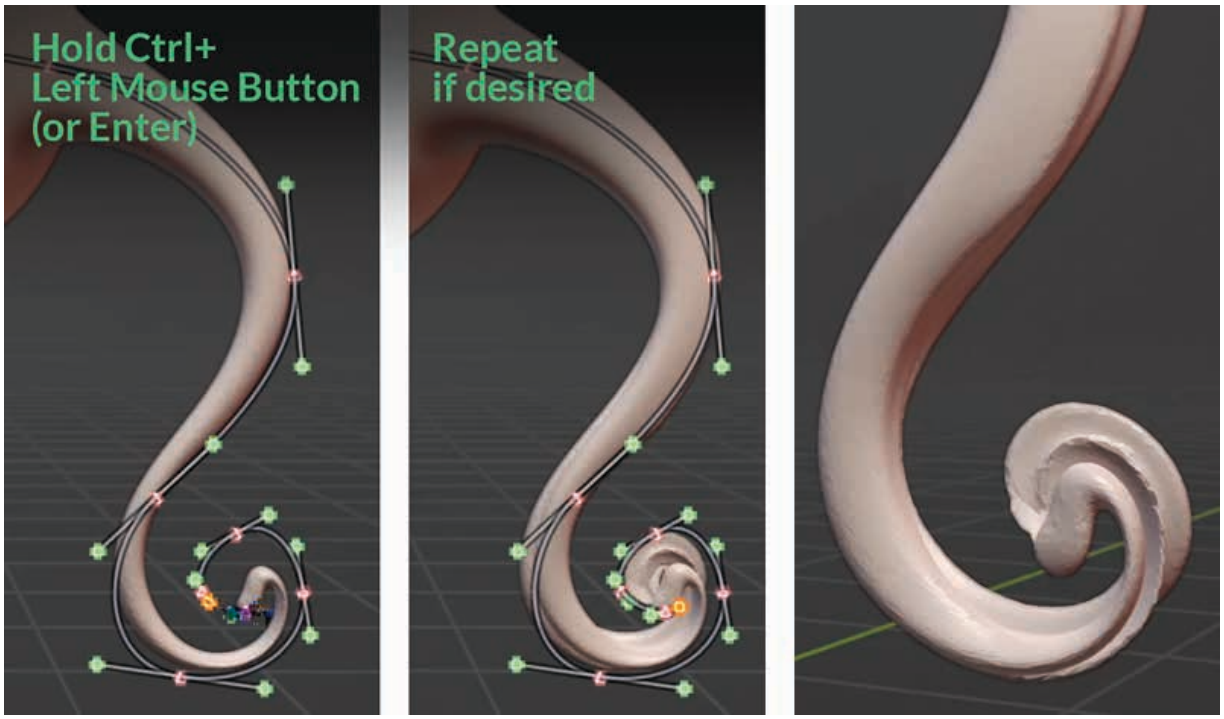
Choose the **Snake Hook** brush. You can **press F** to adjust the Radius. **Press Alt+E** and choose **Curve** Stroke method.



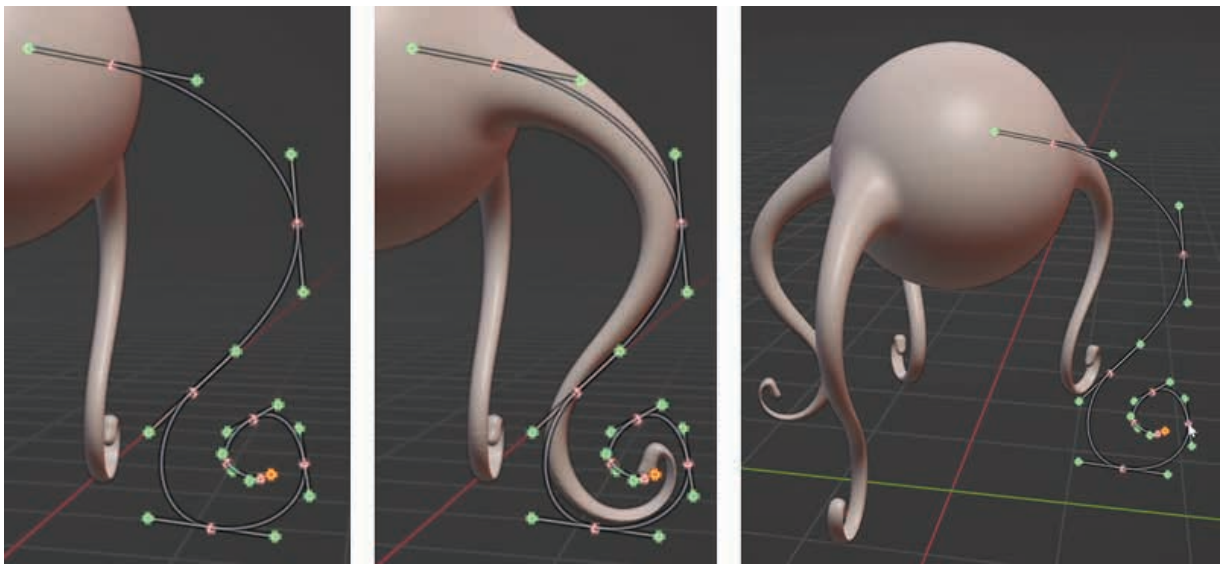
Hold Ctrl, and Right-mouse click and drag to draw the Curve. Repeat to create a longer curve.



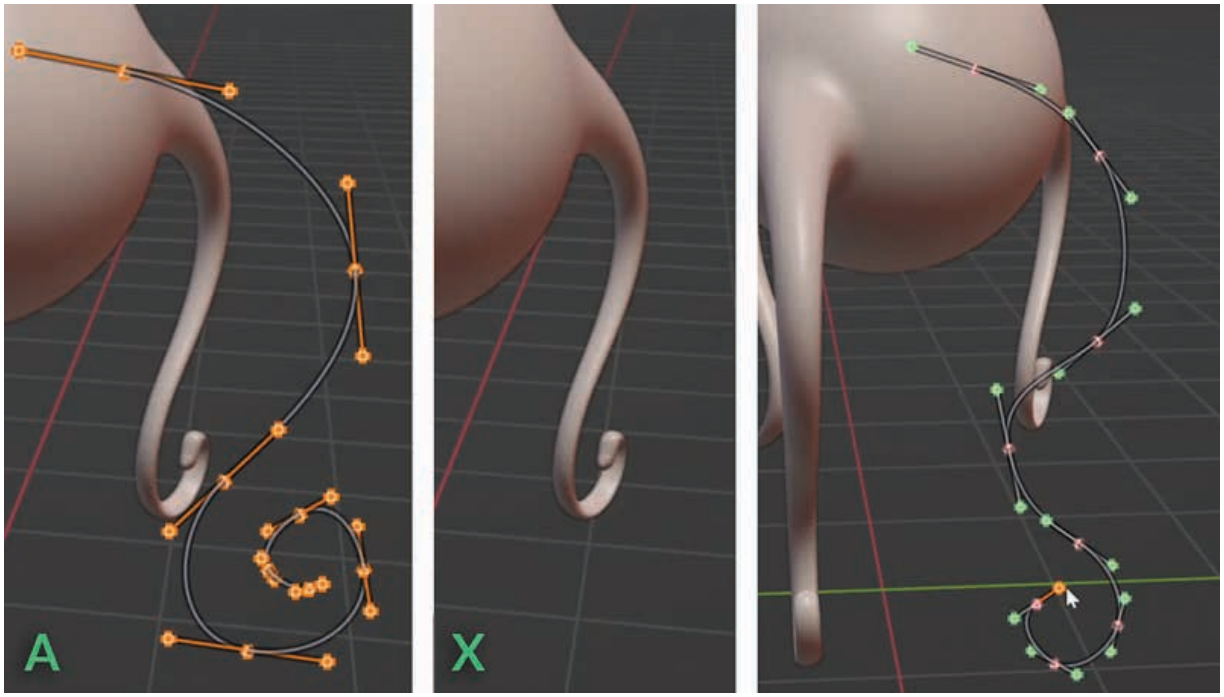
Hold Ctrl and left-click to create the tentacle. If you repeatedly left-click, you increase the intensity. But, repeating the snake hook in the same place can lead to weird geometry (as shown below on the right). However, this can be resolved later by inflating and smoothing those areas, as long as Dyntopo is enabled.



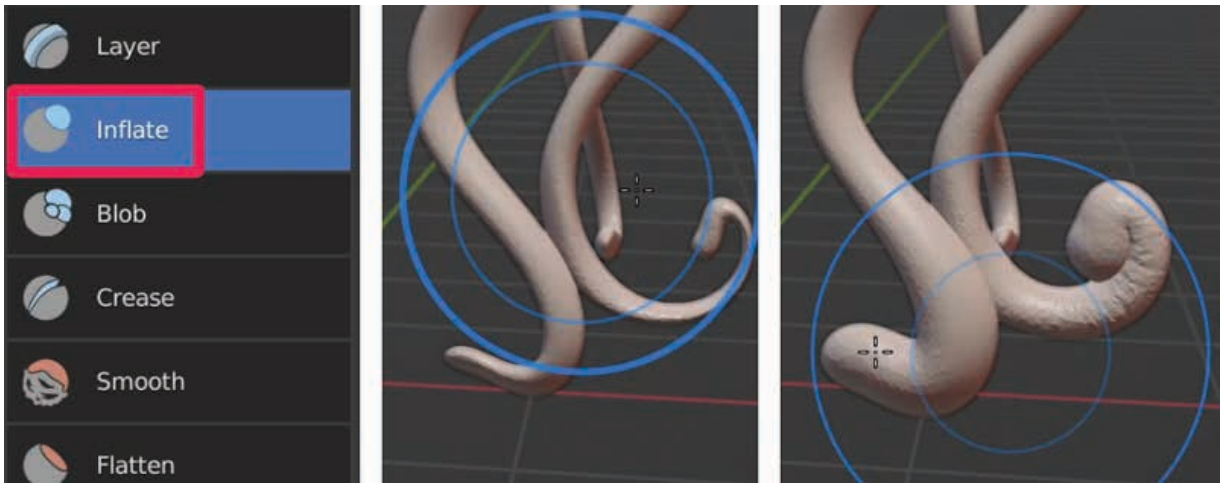
You can reuse the Curve by rotating the view. **Hold Ctrl and left-click** to use the Curve to make another tentacle. It needs to overlap the sphere at least a little bit to work.



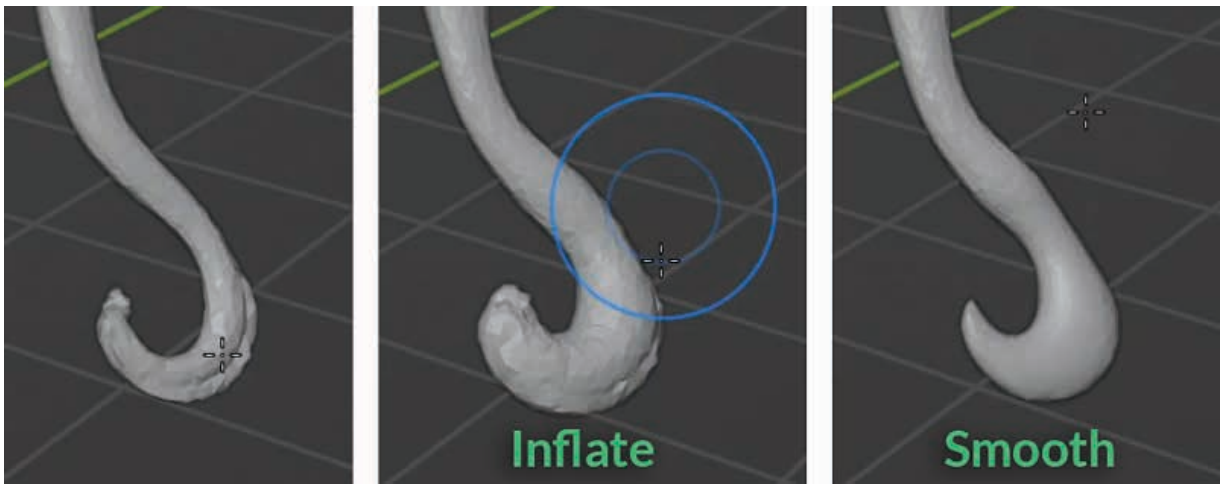
And you can still adjust the curve handles as needed. Select a handle and **press G** to move, **R to rotate**. To delete the Curve, **press A** (you may have to press it twice) to select all handles (they will turn orange) and **press X**. Then you can create a new curve.



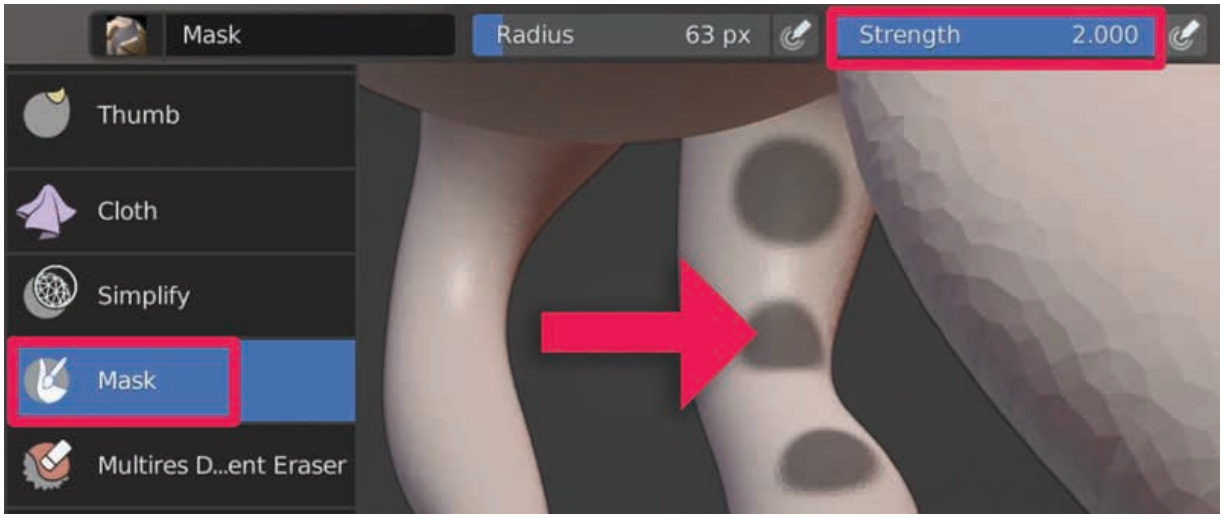
Use the **Inflate Brush** and then the **Smooth Brush** to finetune the tentacles.



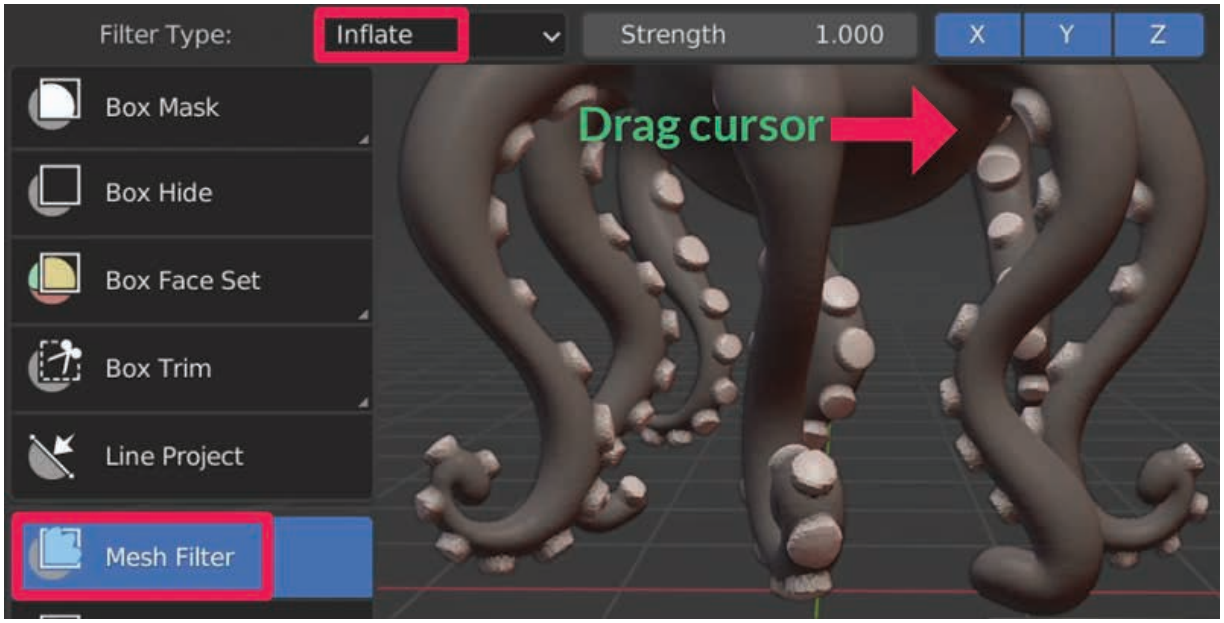
Because **Dyntopo** is still enabled, inflating and smoothing also takes care of weird topology, by repeating the inflating and then smoothing, a few times if necessary.



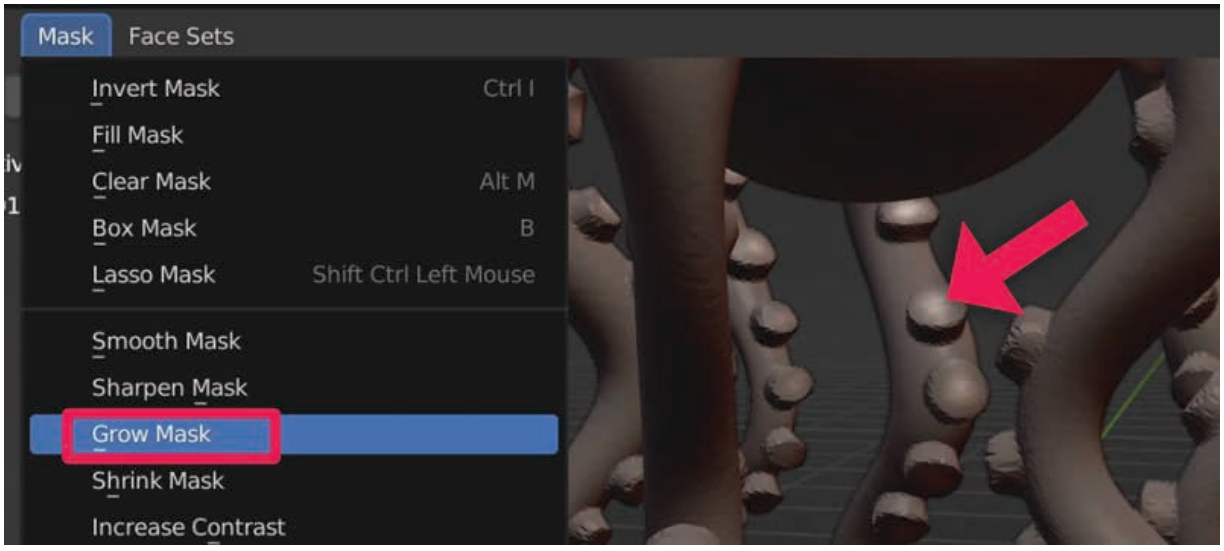
With the **Mask brush** set to a high **Strength** (1 or even 2) draw some dots where you want suction cups.



Invert the mask with **Alt+i** and using the **Mesh Filter** set to **Inflate**, drag the mouse to the right.



Now Grow the mask (**Mask > Grow Mask**) to make the unmasked part smaller. If growing the mask makes the unmasked part disappear altogether, go to **Mask > Smooth Mask** first.



Now you can drag the opposite direction with the **Mesh Filter** set to **Inflate**, to inflate in the negative direction. Clear the mask with **Alt+M**.

If you want, use the Mesh filter set to **Smooth** to smooth everything out.

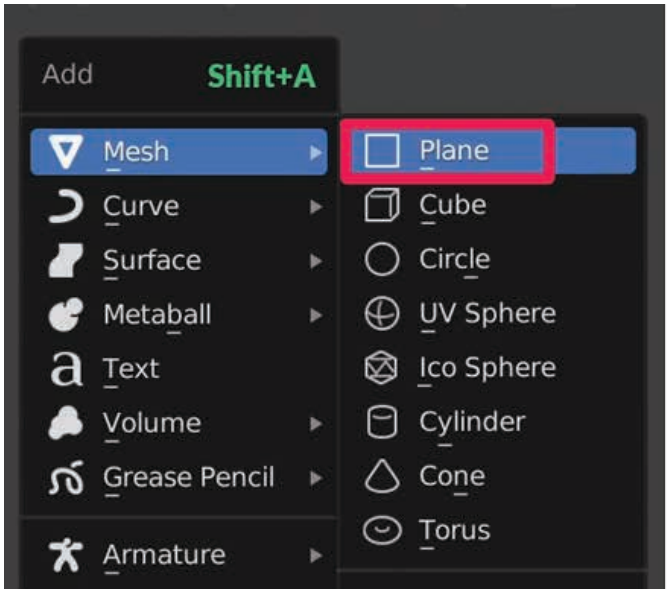
In Object Mode, enable **Shade Smooth**.

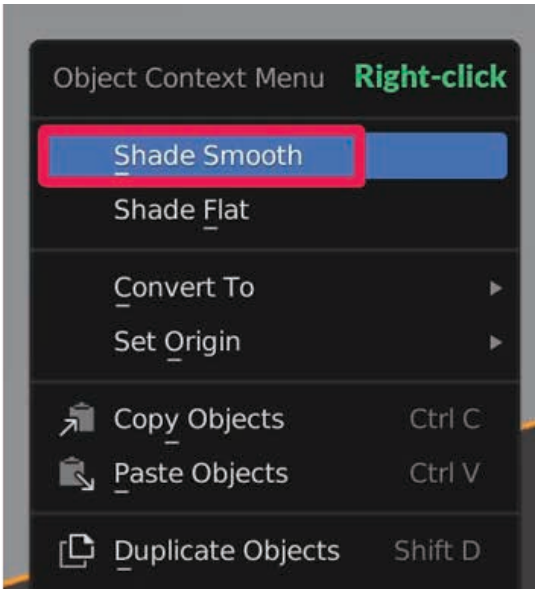
Seamless Tiling Sculpting

Using seamlessly tiling displacement maps is great for easily adding details to properly UV-unwrapped models.

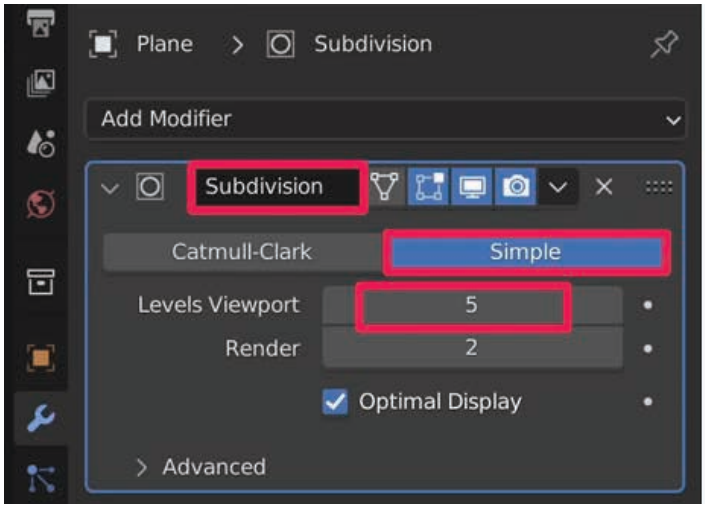
But how do you create such a displacement map using sculpting?

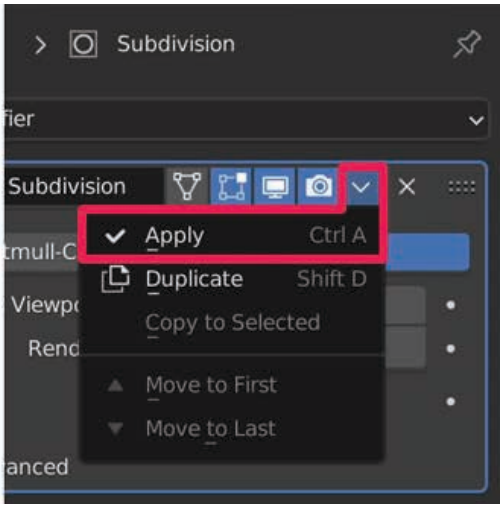
In Object Mode **press Shift+A** and add a **Plane**, right-click on it and set it to **Shade Smooth**.



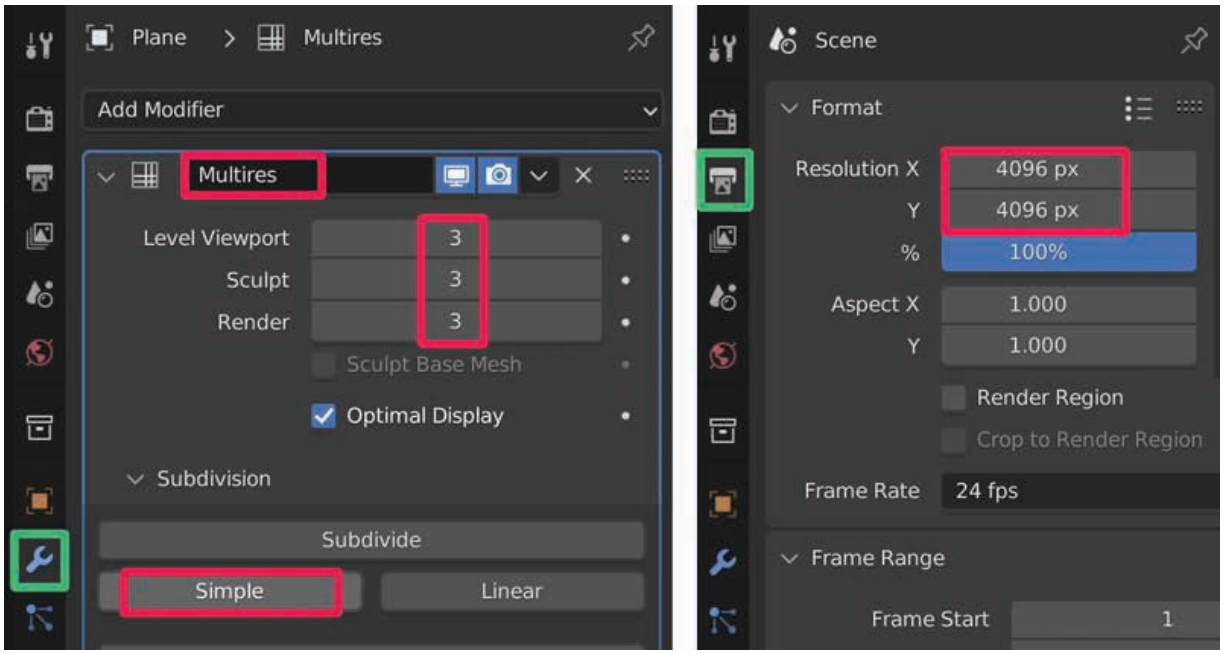


Add a **Subdiv modifier** with 5 levels of **Simple** subdivision and **apply** it. The reason we do this step, is that it improves the performance of the Multires modifier that we add next.

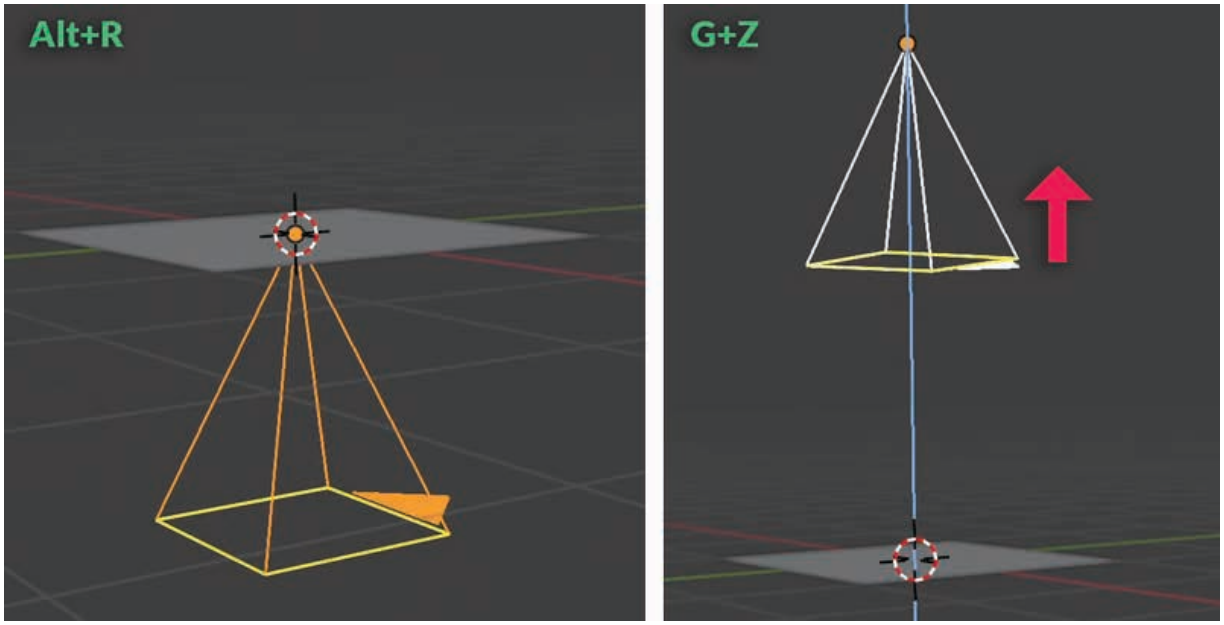




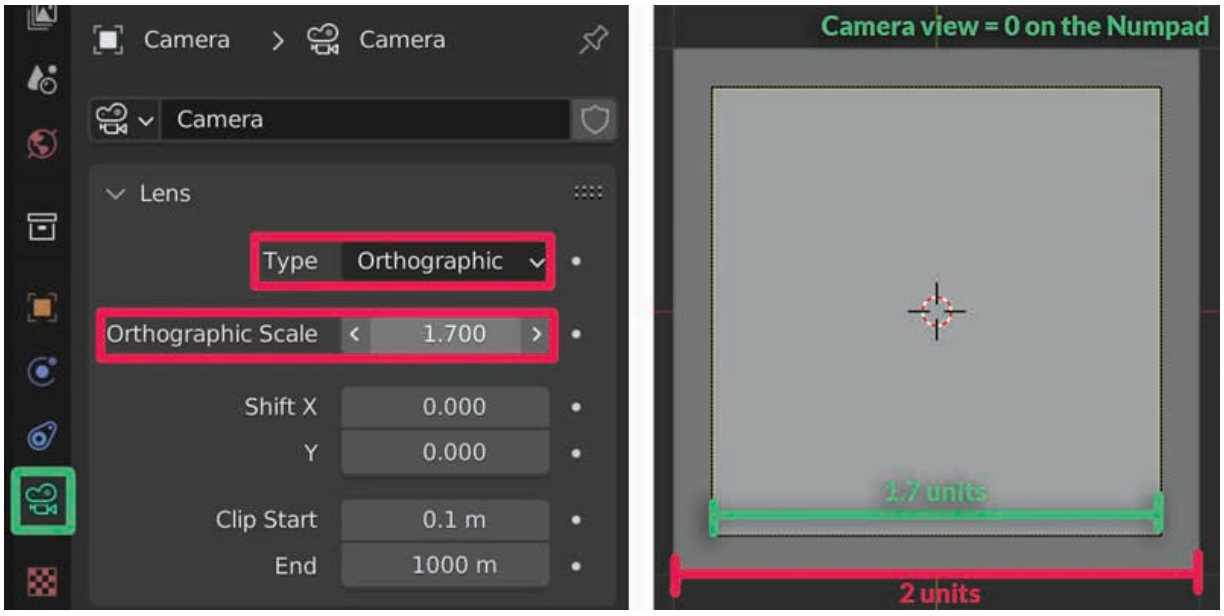
Add a **Multires modifier** with 3 or 4 levels of **Simple** subdivision by **clicking on the Simple button** 3 or 4 times (depending on what your computer can comfortably handle). You don't need to apply this modifier. Set the **Scene Resolution** to a square format.



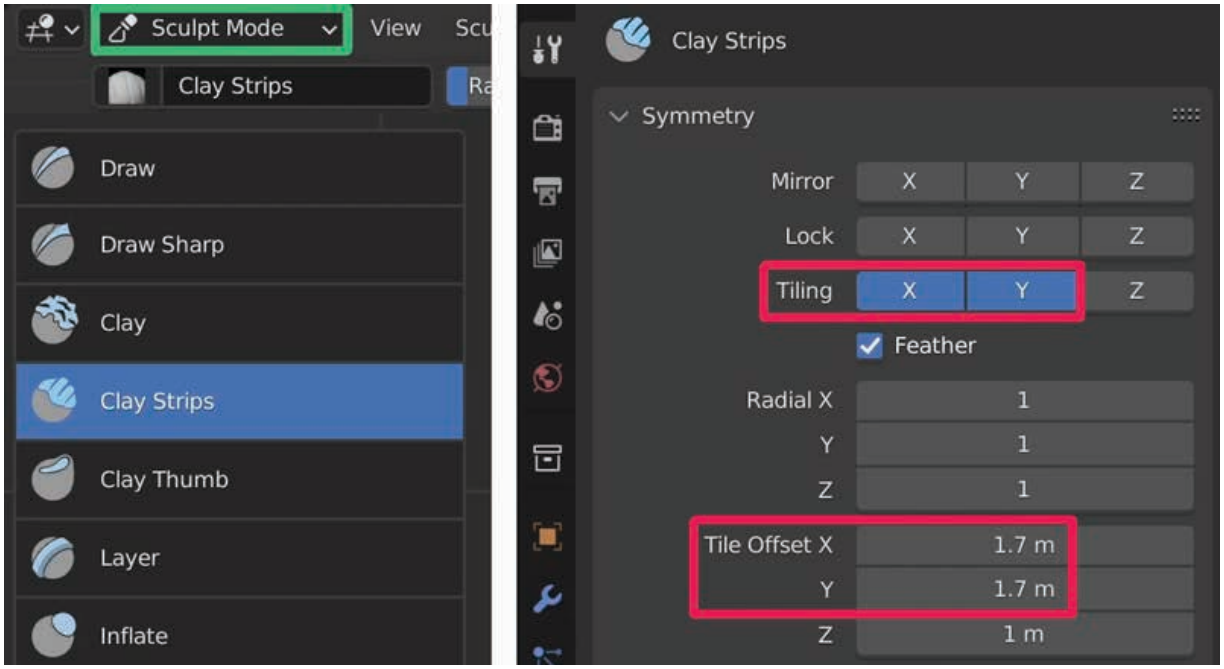
Clear the rotation of the camera by selecting it and **pressing Alt+R**. Move it up the Z-axis by **pressing G+Z** and moving the mouse.



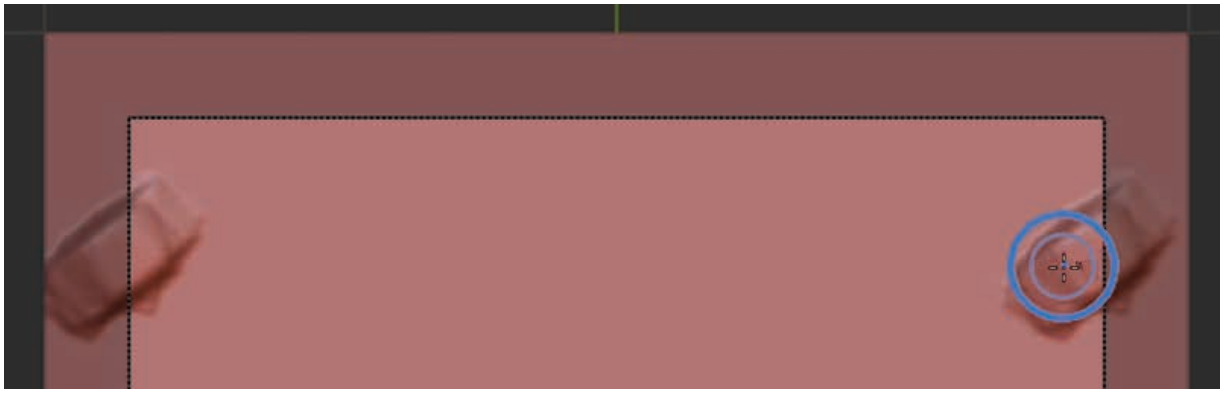
Set it to **Orthographic**, with an **Orthographic Scale of 1.7**. Now when you **press 0 on the Numpad** to see the Camera View, you can see that it's a square format and that the camera view is slightly smaller than the surface area of the Plane (the Plane is 2 units wide, the camera view only 1,7 units wide).



In **Sculpt Mode**, in the **Brush Symmetry** options **enable X and Y Tiling**. The **Tile Offset** value should be the same as the Orthographic Scale, 1.7 in this case.

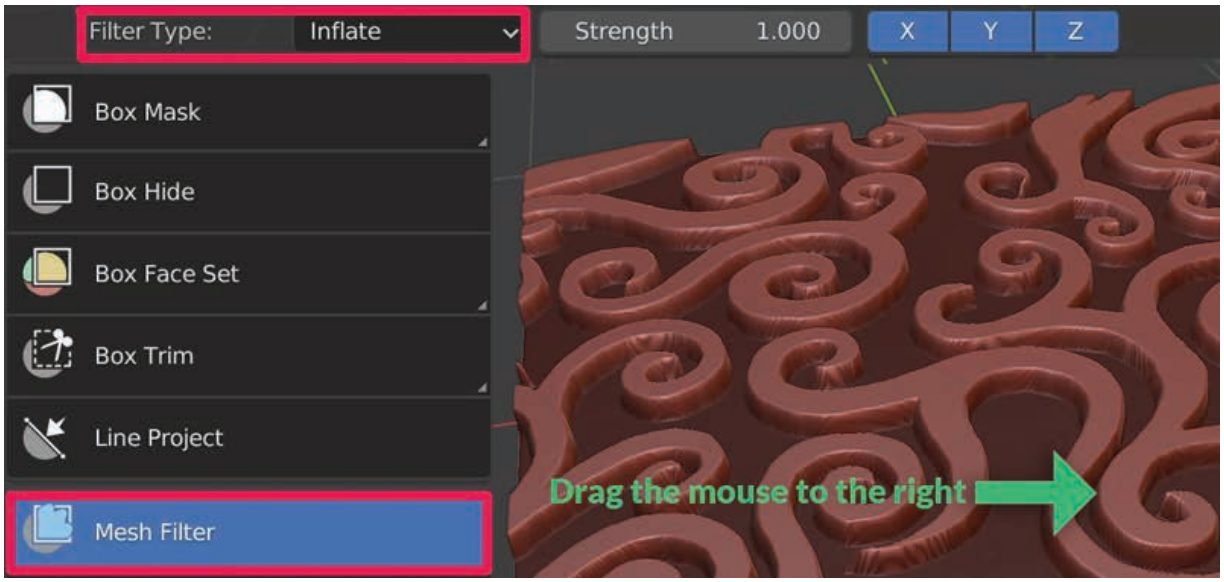


This offset makes it easier to sculpt over the seams area. Now when you go over the border on one side, the brush stroke reappears on the opposite side.

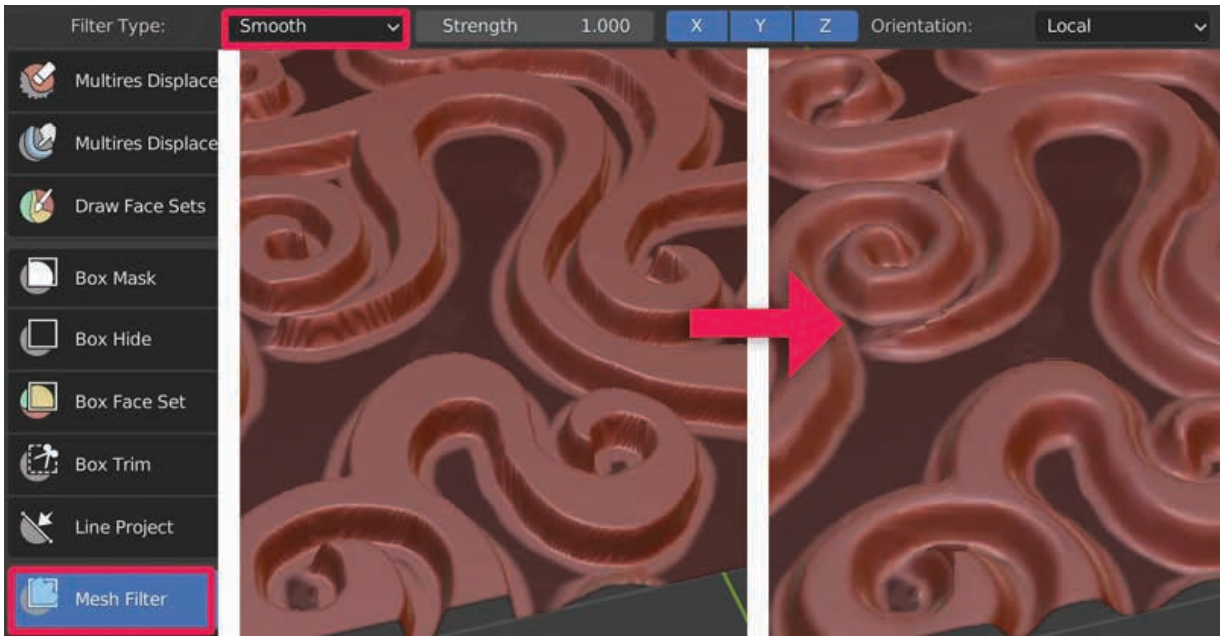


You can use any sculpting techniques you like. In this case, I drew some ornamental designs using the **Mask brush**.

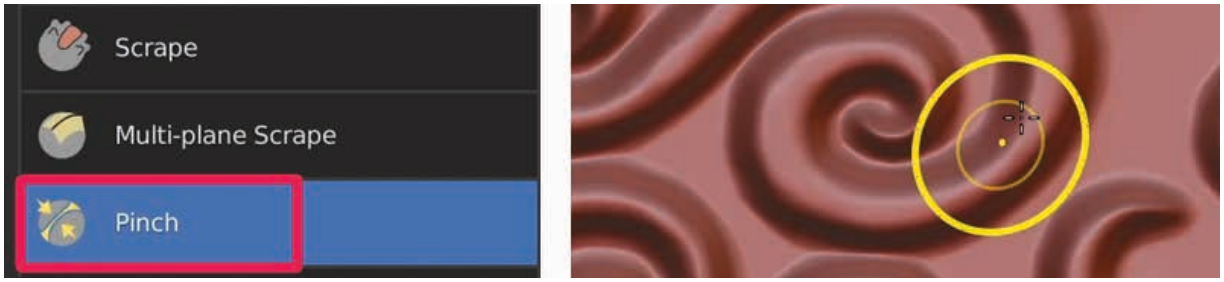
After inverting the mask with **Ctrl+i**, I then extruded the unmasked region using the **Mesh Filter** set to **Inflate**. Drag the mouse to the right to use the Mesh Filter.



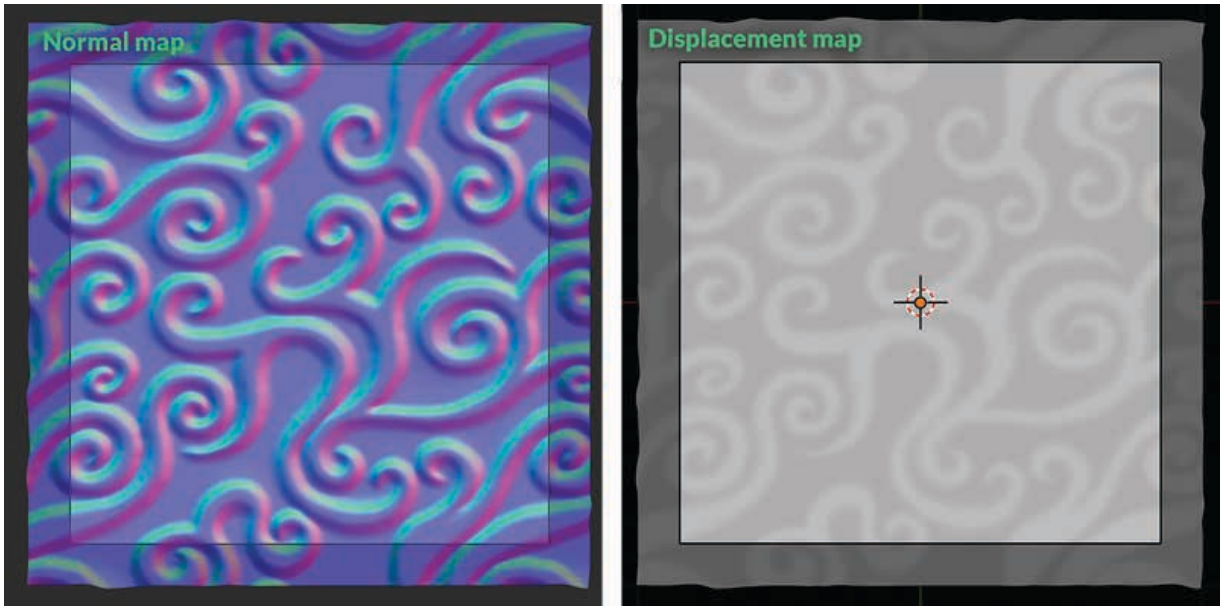
After shrinking the mask (**Mask > Shrink**) to increase the unmasked region, I used the **Mesh Filter** again, but set to **Smooth**. Again, drag the mouse to the right to use the Mesh Filter.



Then as a finishing touch I pinched some of the peaks using the **Pinch Brush**.



Finally you can bake the camera view as a Normal map, using the Matcap method or a Displacement map using a gradient material.

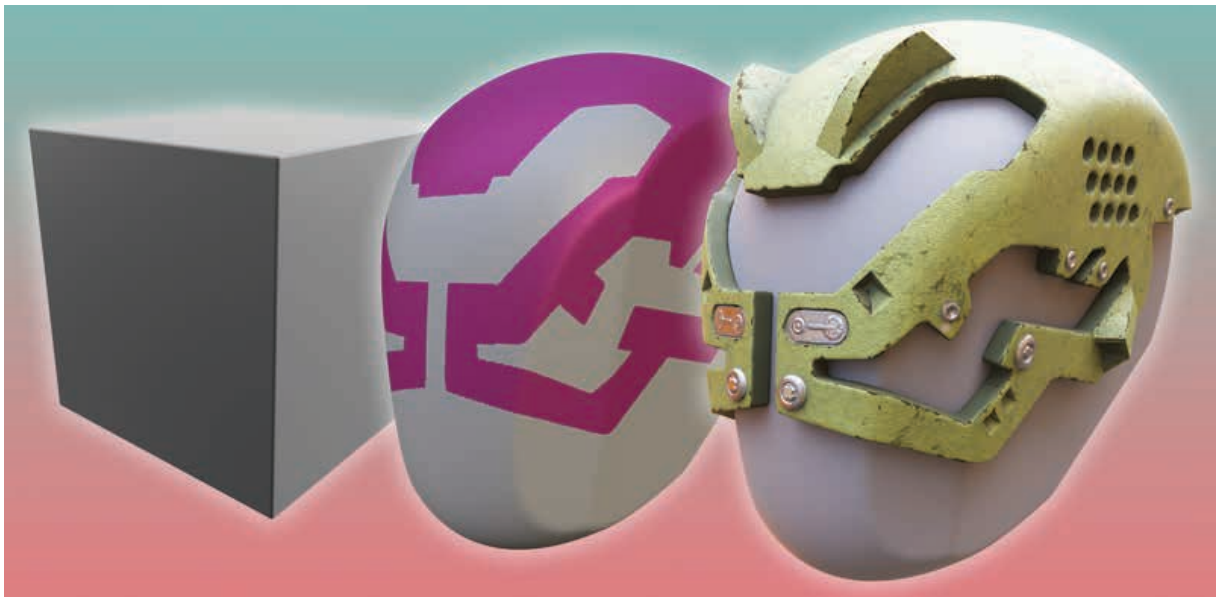


Save it as an uncompressed image (**Targa** for the Normal Map, 32-bits Full Float **OpenEXR** for the Displacement map) and it will tile seamlessly on your UV-unwrapped model.

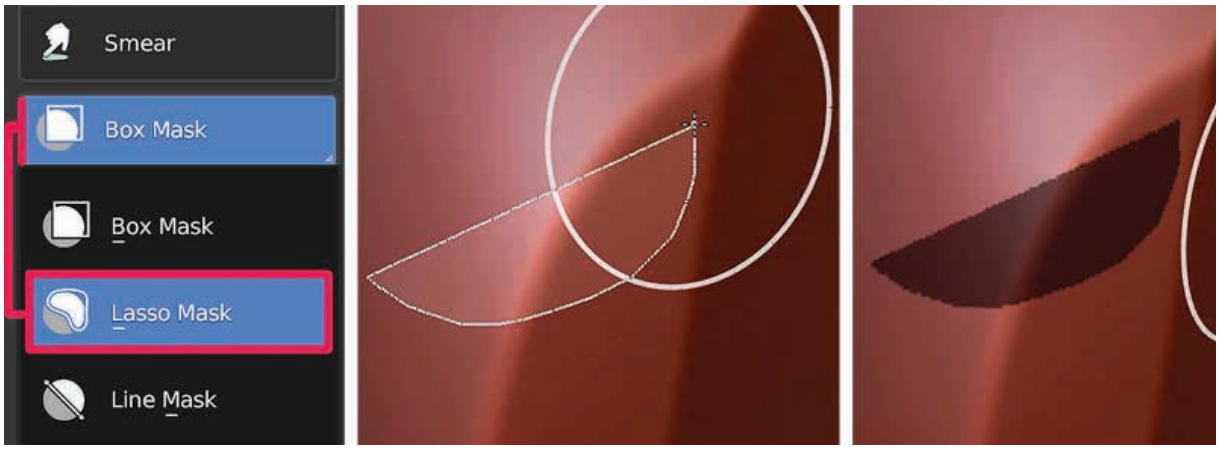
Creating a robot helmet in Sculpt Mode

Using some sculpting tools, we can easily extract geometry to create a helmet. First, we start with a basic head shape that has enough geometry for sculpting.

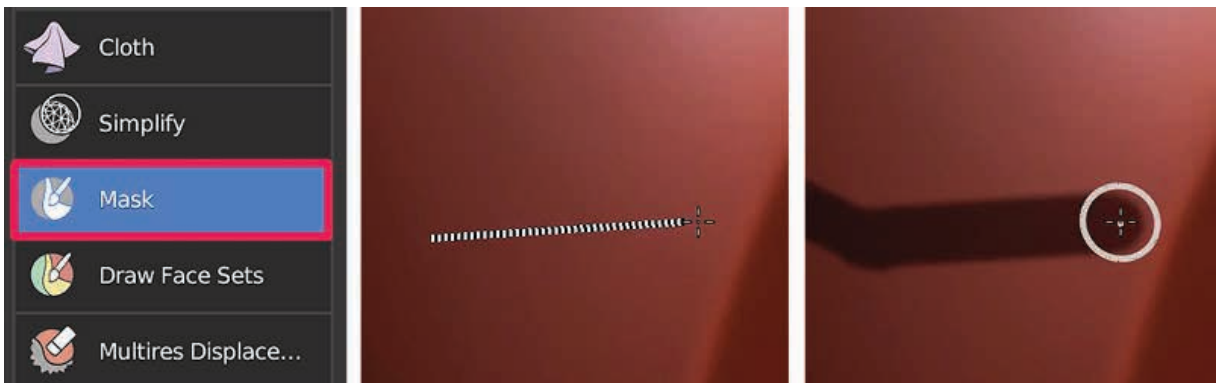
Watch a longer video about this topic here.



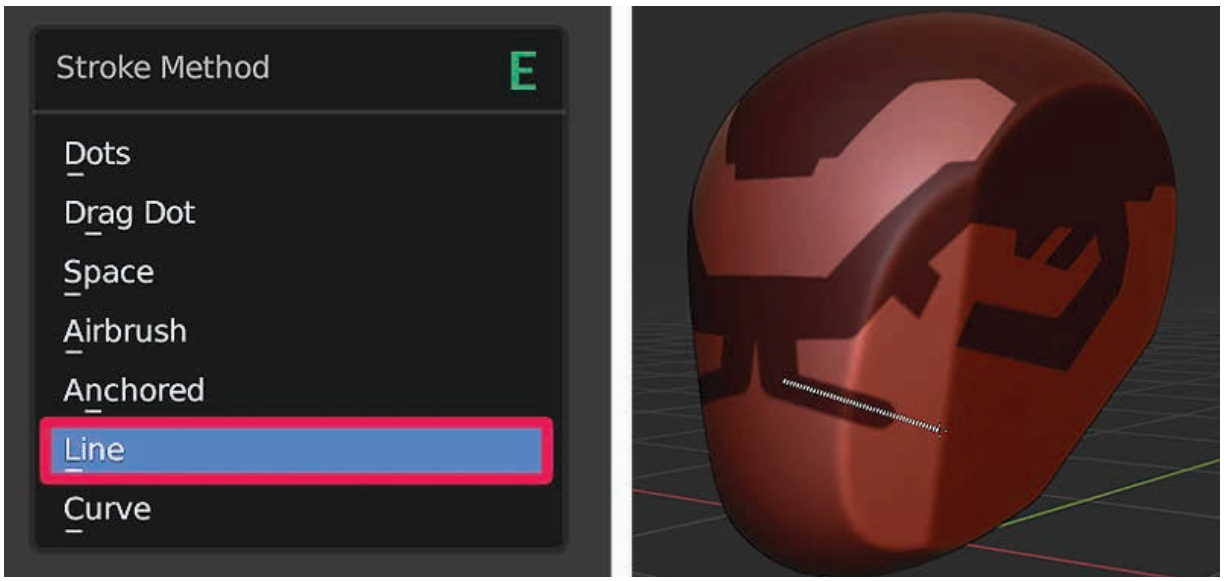
After sculpting a simple and smooth base shape, use the **Lasso Mask** in Sculpt Mode to create some masks.



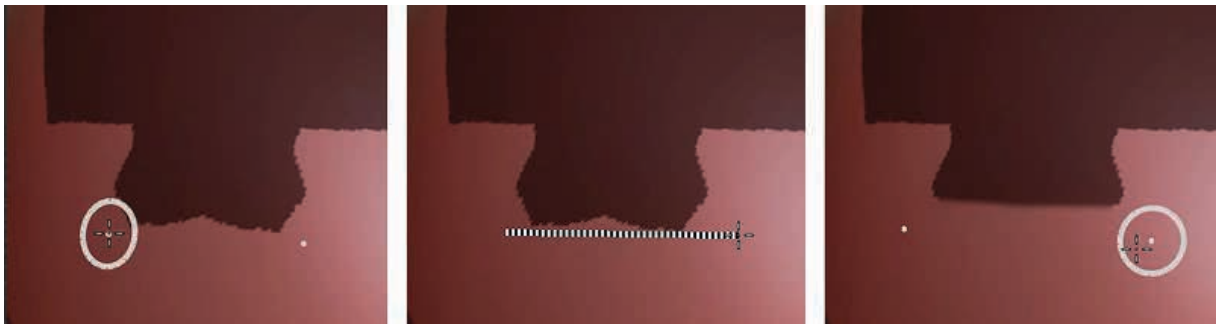
For finetuning, you can use the **Mask tool**. To paint a clean, crisp mask it helps to have a lot of geometry and a high **Strength** of the Mask brush, like 2.



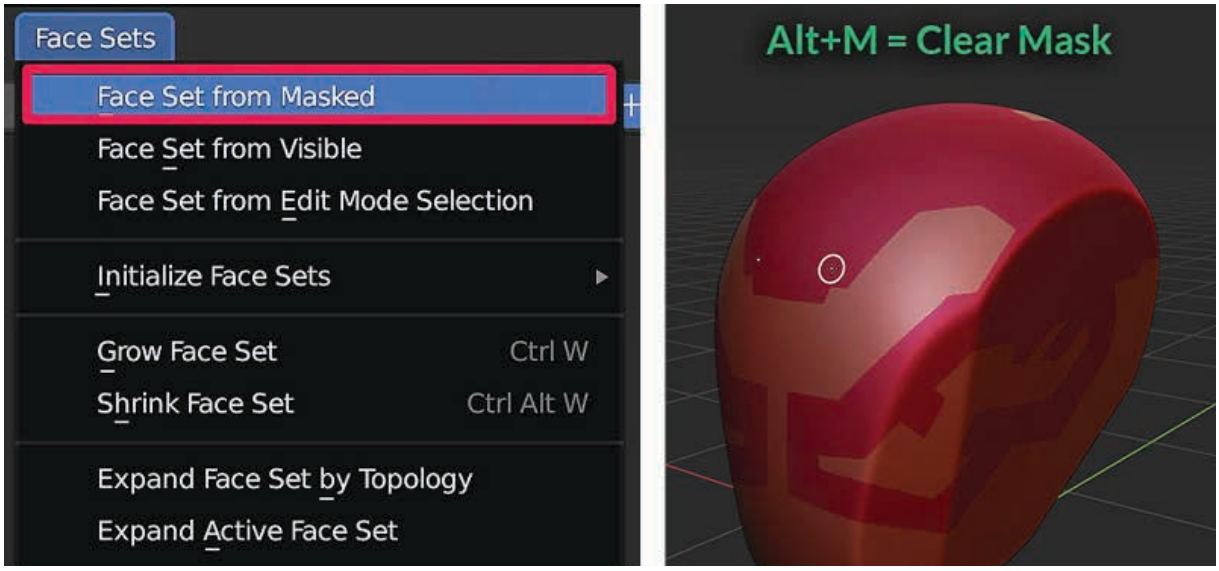
This is set to **Stroke Method "Line"** to create nice straight lines, that are suitable for making a hard surface style mask. You can get the **Stroke Method** menu by **pressing E**. The default Stroke Method is Dots.



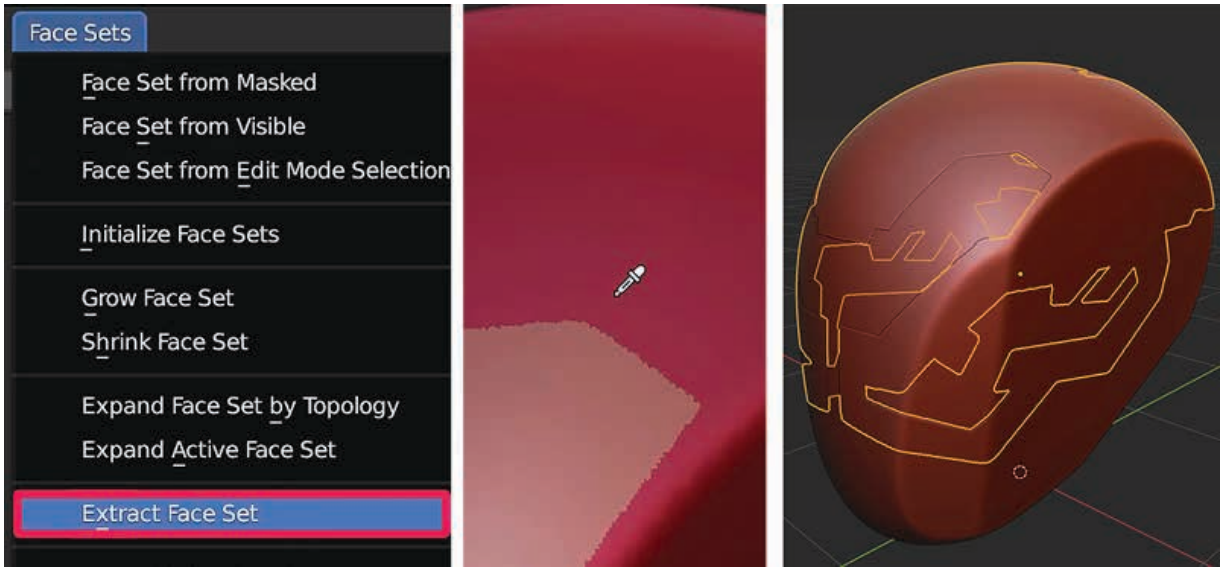
With the **Mask** tool, you can also subtract, by **holding down Ctrl**. This helps to clean up edges of the mask.



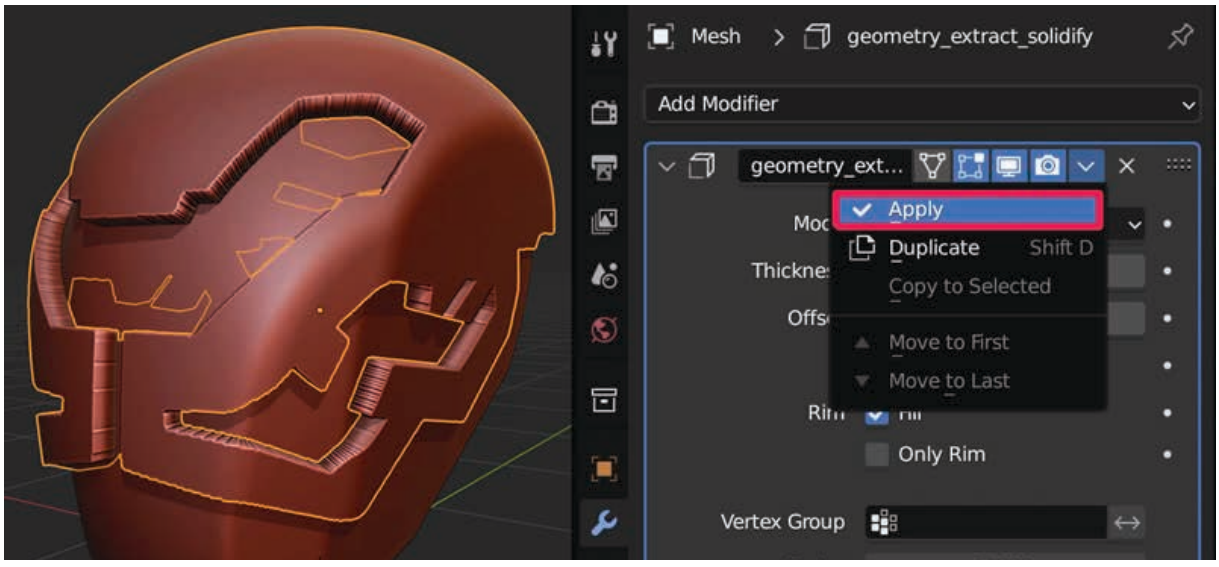
Once you're happy with your mask turn it into a **Face Set** by going to the Face Sets menu and choosing Face Set from Masked. You can also extract Geometry from a Mask, but Face Sets have some benefits, like the ability to use the Mesh Filter "Relax Face Sets".



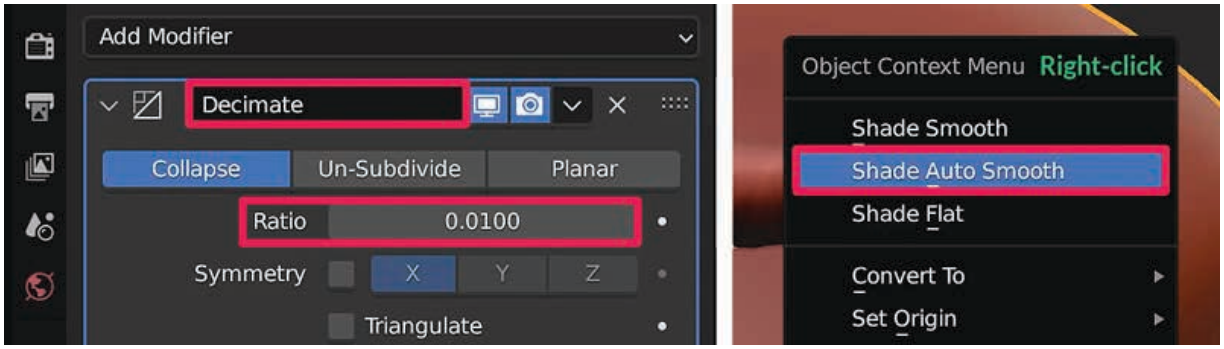
Extract geometry by going to **Face Sets > Extract Face Set**.



This gives us a separate new object, with a **Solidify modifier** on it. We can increase the **Thickness** of the geometry here. After setting the thickness to your liking, **apply** the modifier.



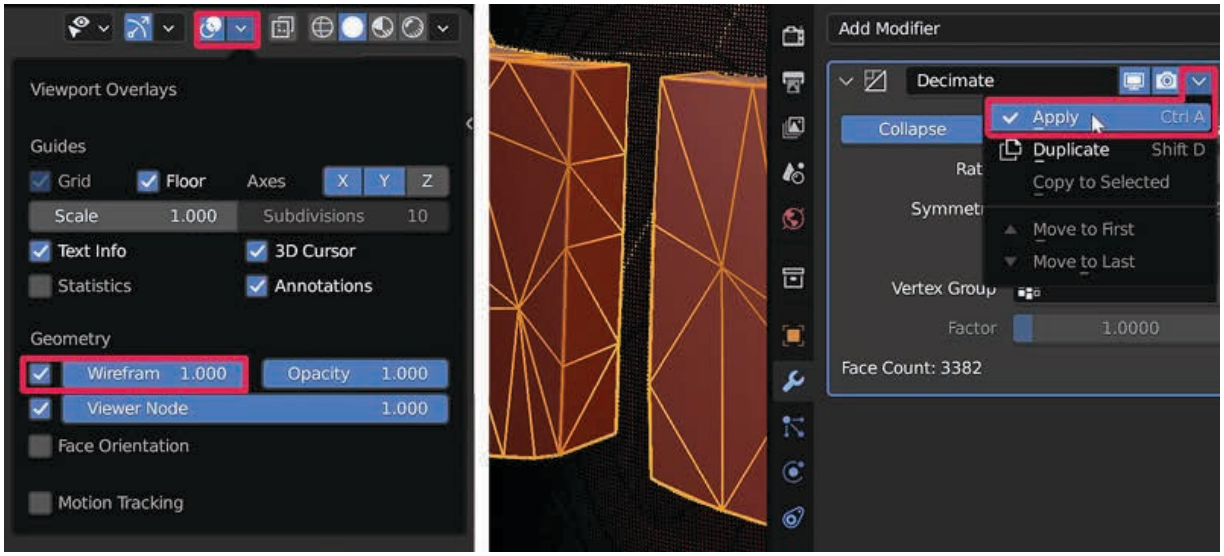
Next add a **Decimate modifier** and find the lowest **Ratio** that's possible without destroying the shape. You can right-click and enable **Shade Auto Smooth** to get a better idea of what the geometry will look like, later.



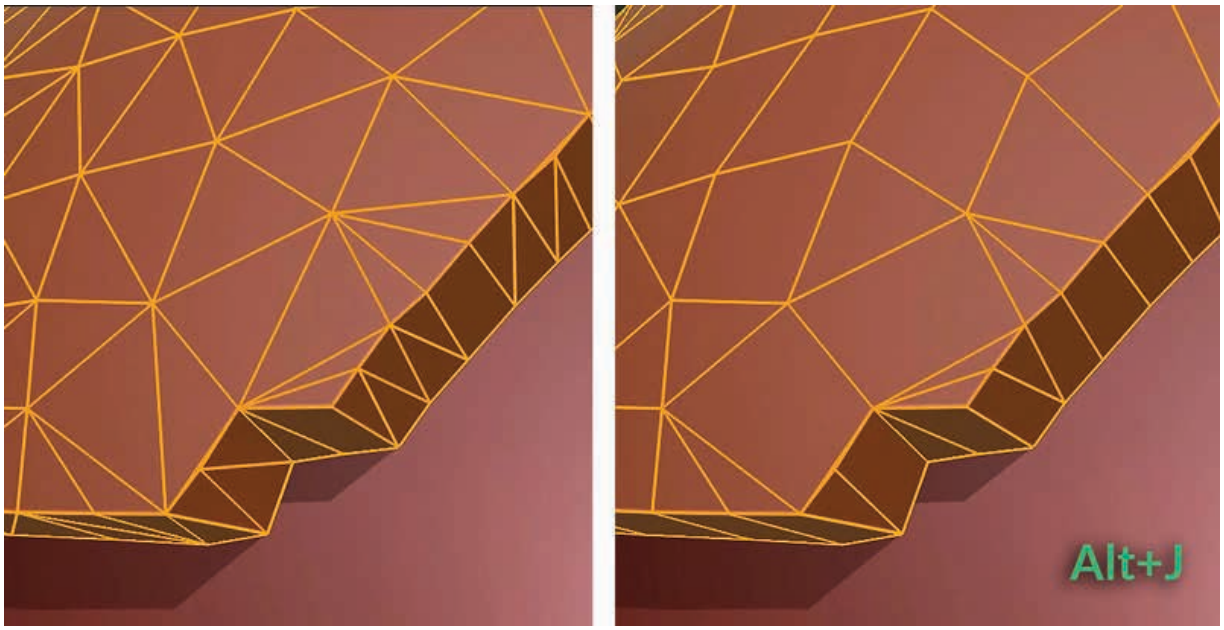
The **Ratio** depends on how much resolution your mesh had, so it requires some experimentation.

The lower you get, the more mechanical it will look due to the low amount of geometry. But go too far, and the shape will change too much.

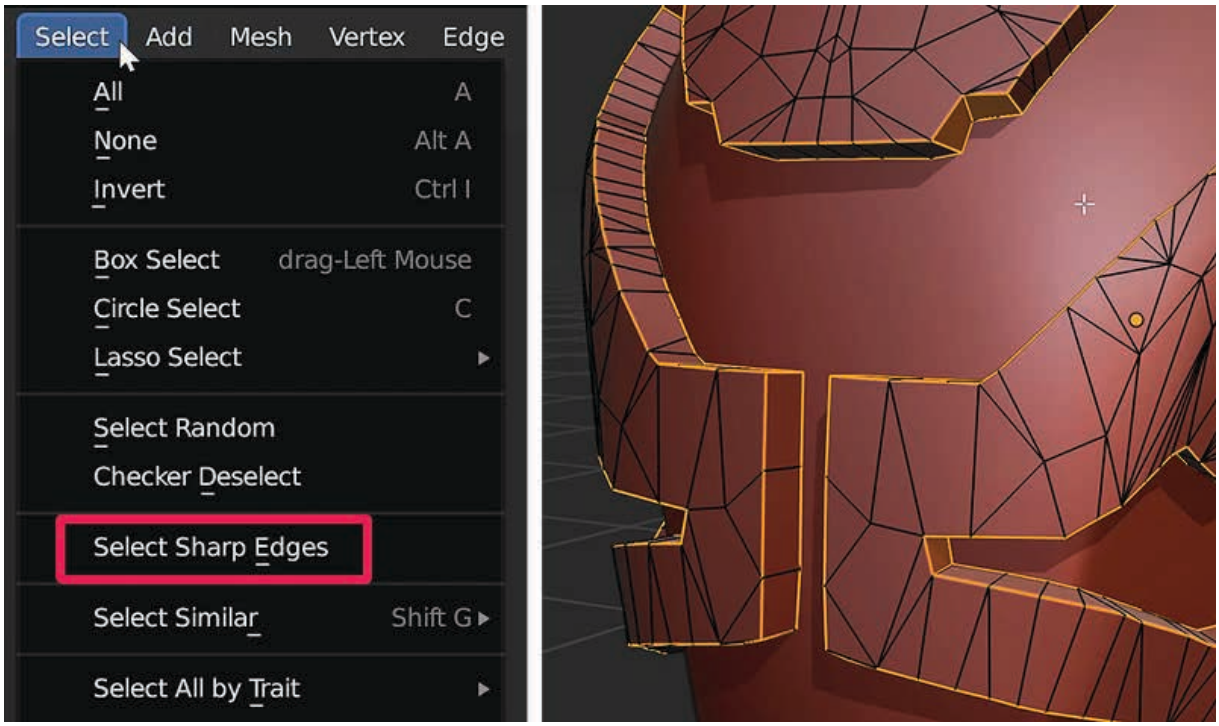
You can enable Wireframe in the Viewport Overlays to see the geometry better. Once you're happy with it, apply the modifier.



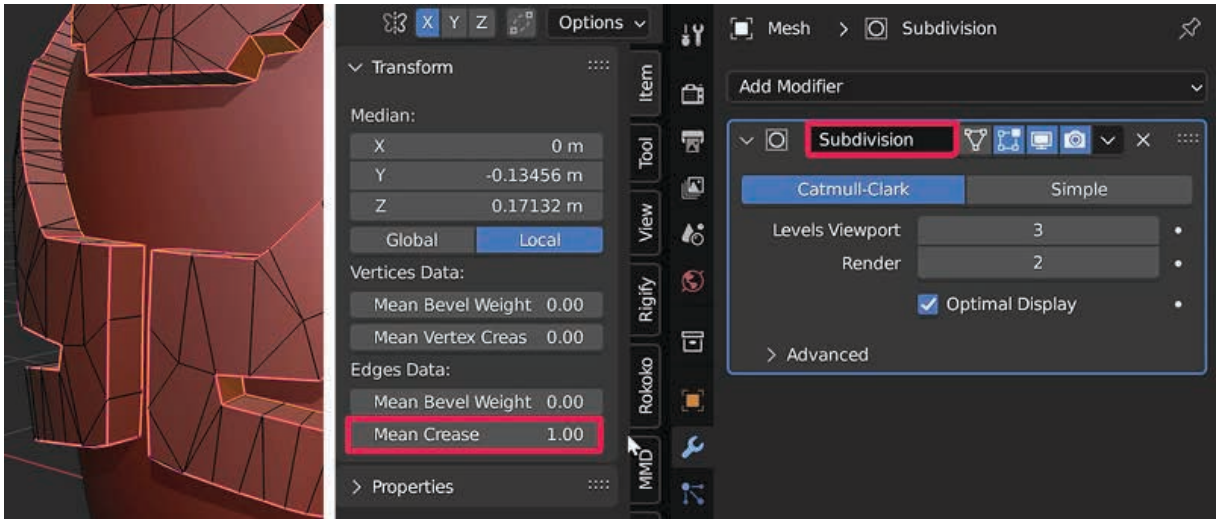
Select all (**A**) and **press Alt+J** to convert the triangles to quads where possible. This reduces the coplexity some more.



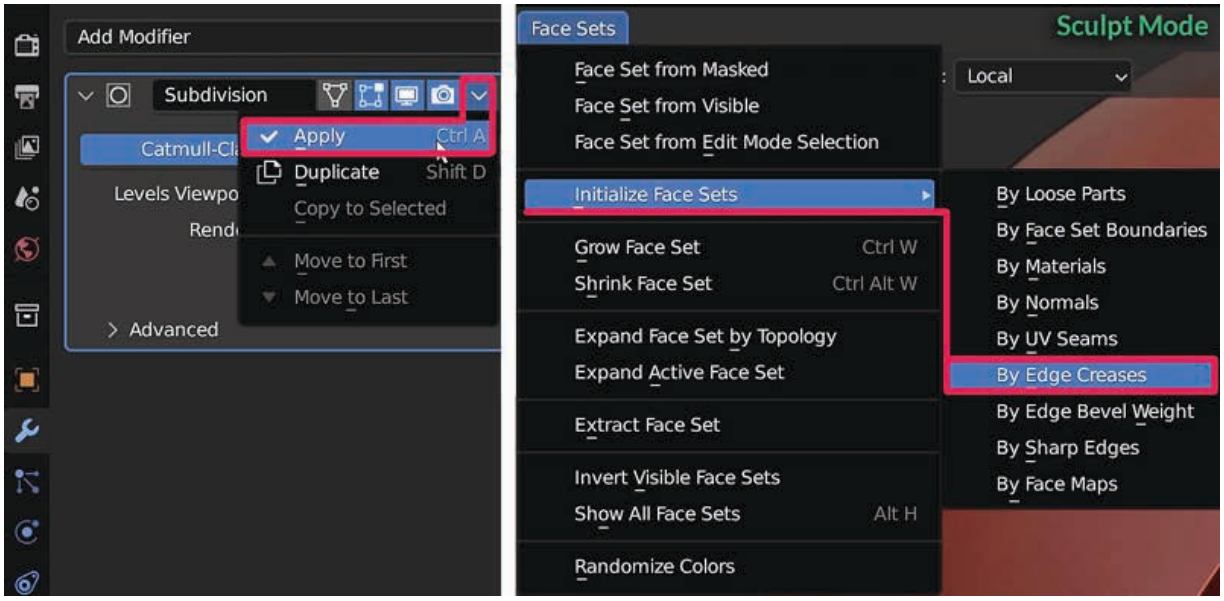
Then deselect all (**Alt+A**) and go to **Select > Select Sharp Edges**. This selects everything that's more than a specific angle (30 degrees by default).



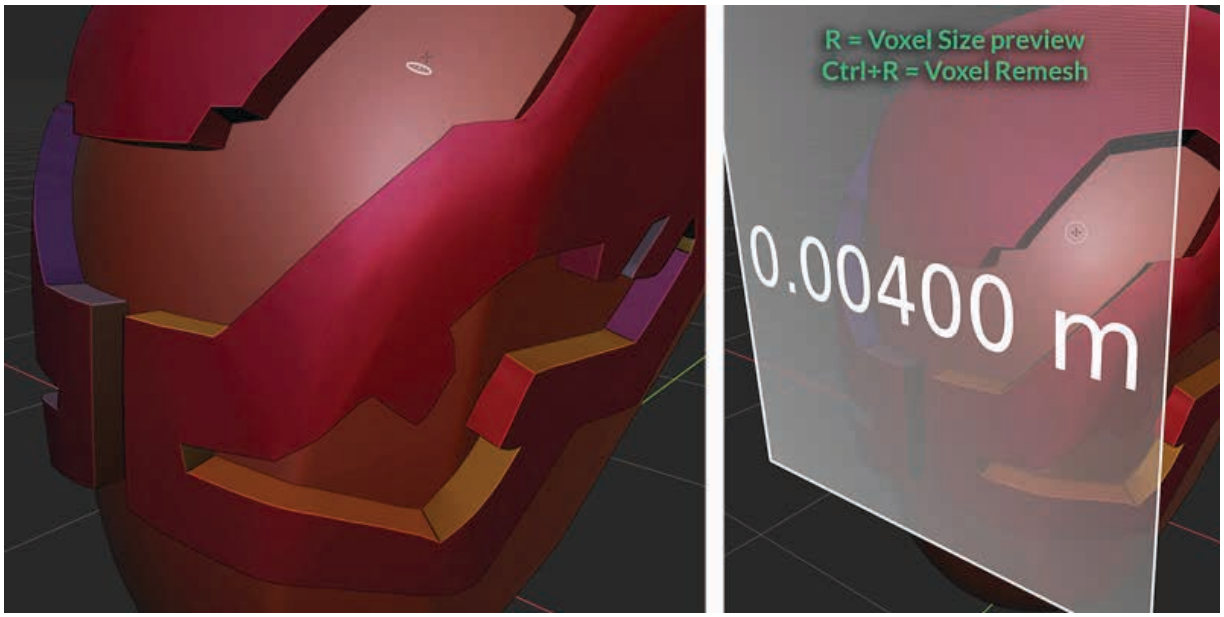
Crease these edges by either going to the Item tab in the Option Panel and setting **Mean Crease** to 1, or by **pressing Shift+E+[1 on the Numpad]**. Add a Subdivision modifier.



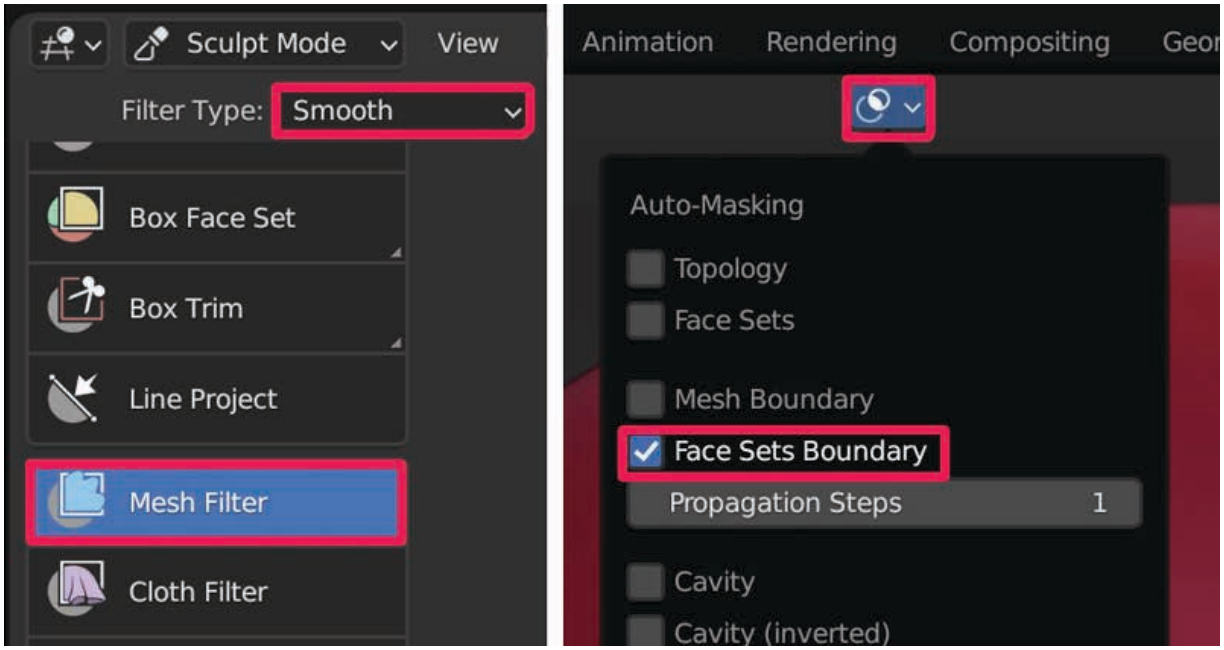
Apply the Subdivision modifier. Now you have more resolution again, but the sharp edges have been protected by the Creasing. In **Sculpt Mode**, go to **Face Sets > Initialize Face Sets > By Edge Creases**.



As you can see, this has created Face Sets bordered by where the Edge Creases were. Do a **Voxel Remesh** to make the geometry more sculpting-ready. **Press R** to choose the Voxel Size, then **Ctrl+R** to remesh.



Now you can use the **Mesh Filter** set to **Smooth** in combination with the **Auto-masking option "Face Sets Boundary"** to smooth everything, while the sharp edges will remain protected. *Note that Auto Masking is available since Blender 3.4.*



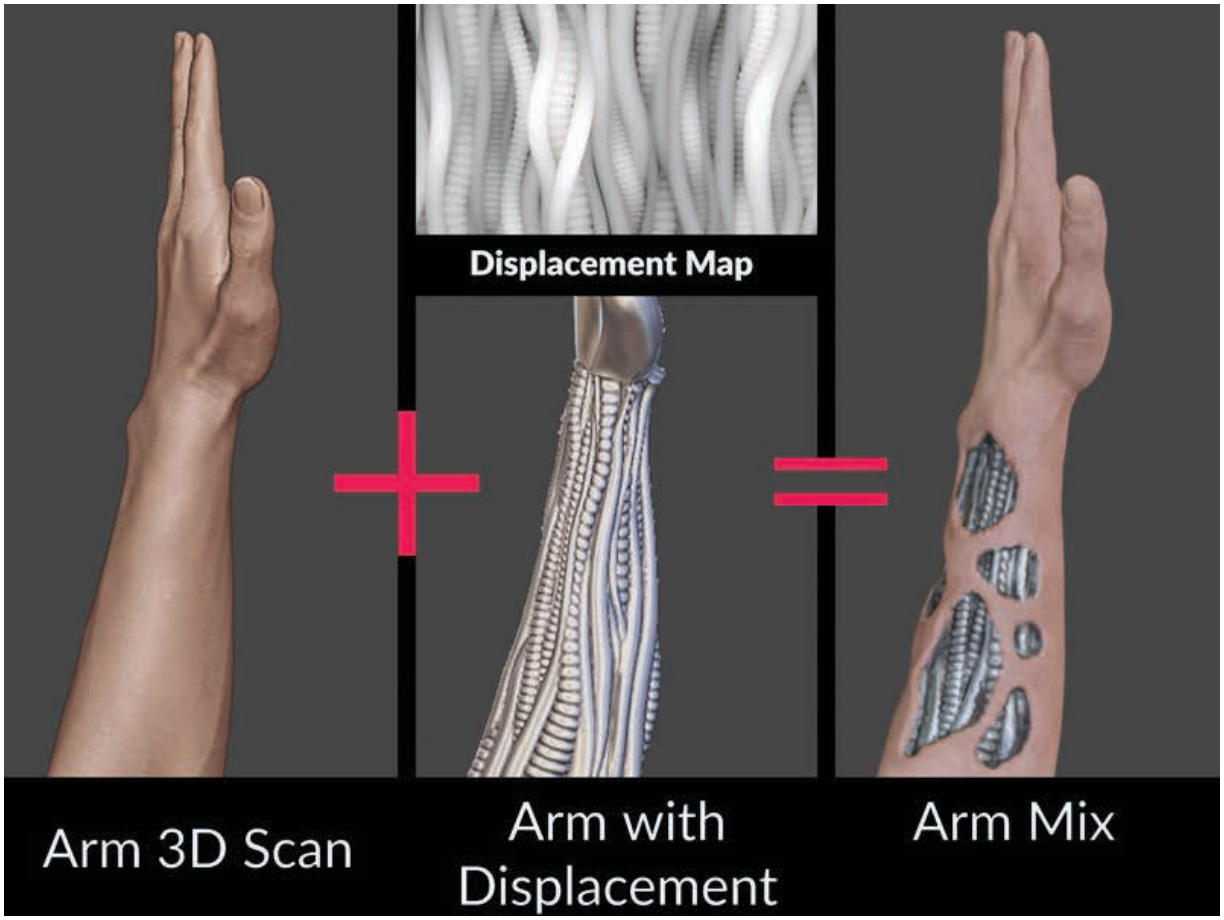
Drag to the right and repeat to smooth more, until it looks smooth enough.

After that, you can for example use some Alpha textures to add more details.

Zbrush-like Morph Target in Blender

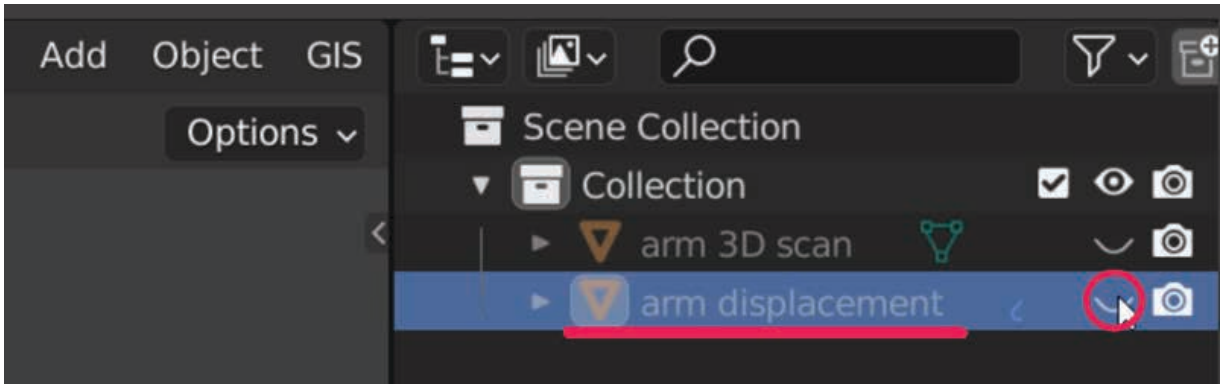
Using this technique, you can mix two different layers of detail together non-destructively.

In this example, we have two different objects that we want to mix. An arm 3D scan and a duplicate which has a tiling displacement map on it. We want to be able to mix them.

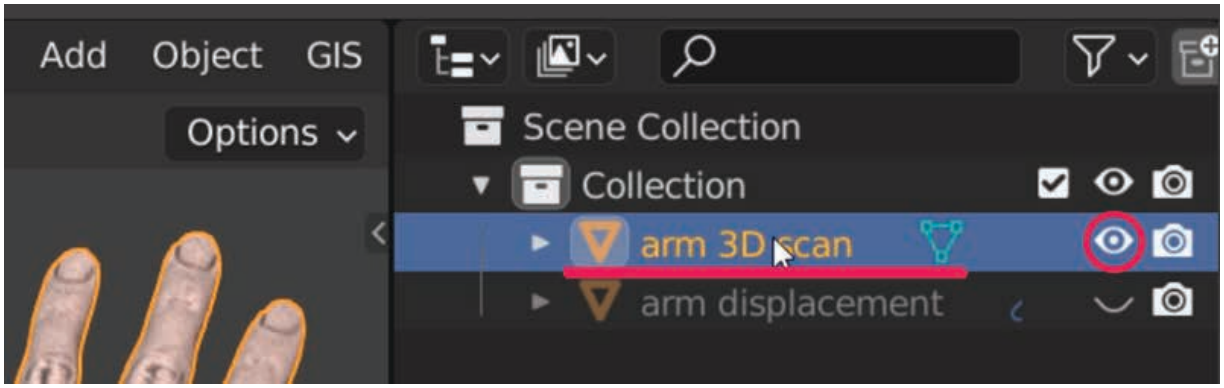


Arm 3d model by Thunk3D scanner on Sketchfab

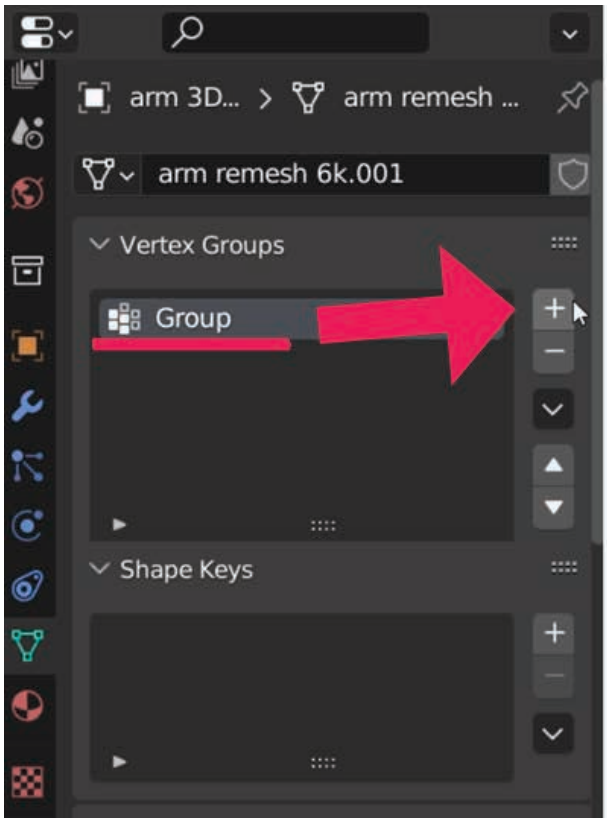
Disable the visibility of the copy with the displacement map.

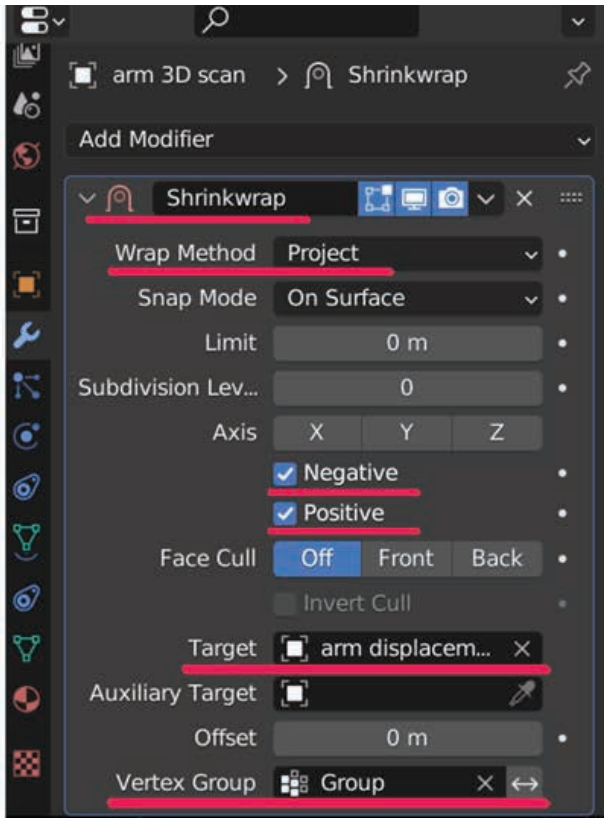


Select the 3D scanned arm.

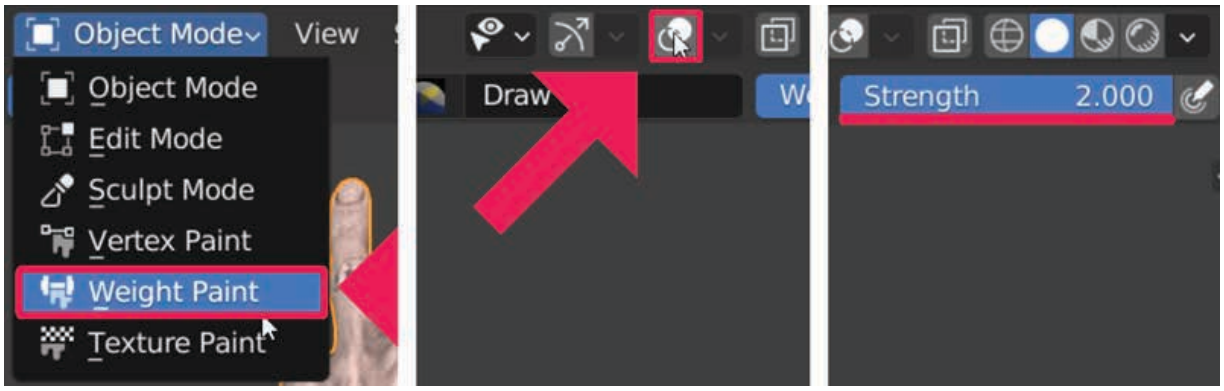


Make a new **Vertex Group** by clicking on the + button. Add a **Shrinkwrap modifier**. Set Wrap Method to **Project** and enable both **Negative** and **Positive**. As the **Target** choose the other mesh (the one with a Displacement map). Add the **Vertex Group.**



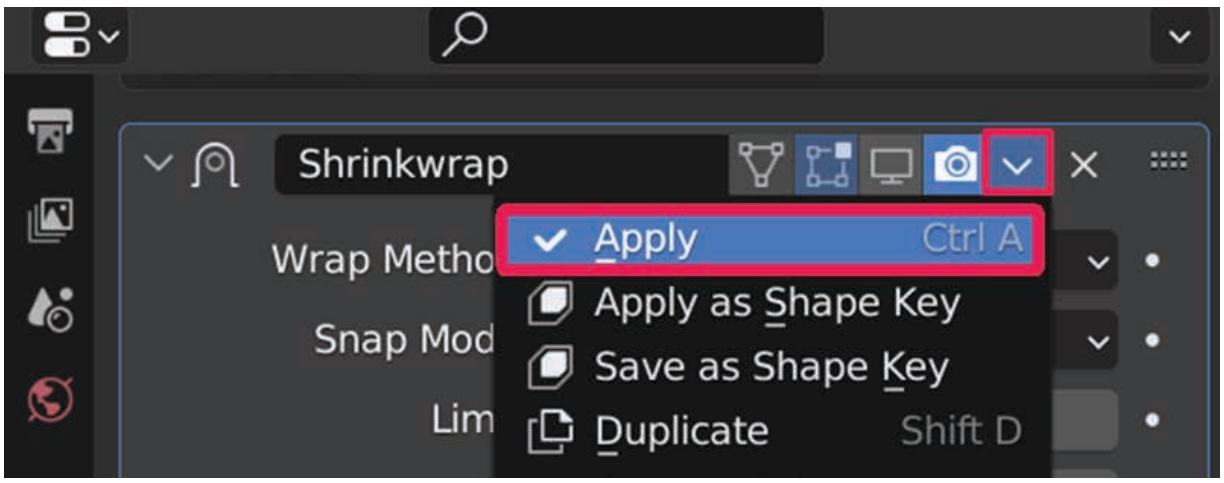


Go to **Weight Paint** Mode and disable **Overlays**. You can increase the **Strength**.



Paint where you want to see the other layer.

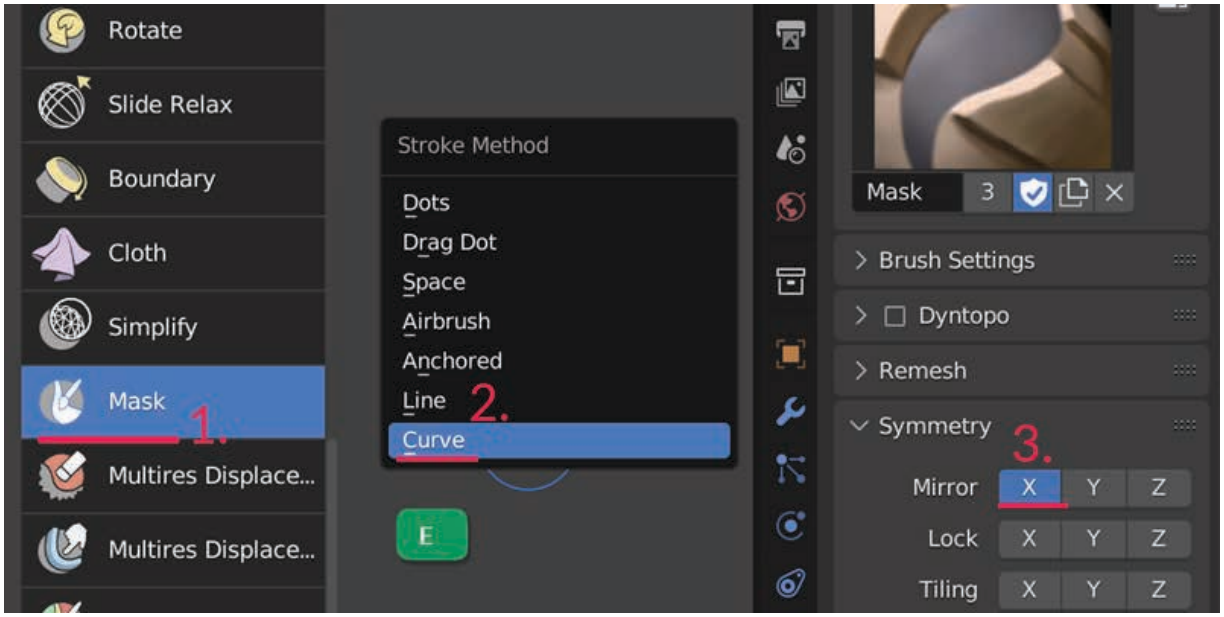
If you want you can **apply** the Shrinkwrap modifier and sculpt on this to fine-tune it.



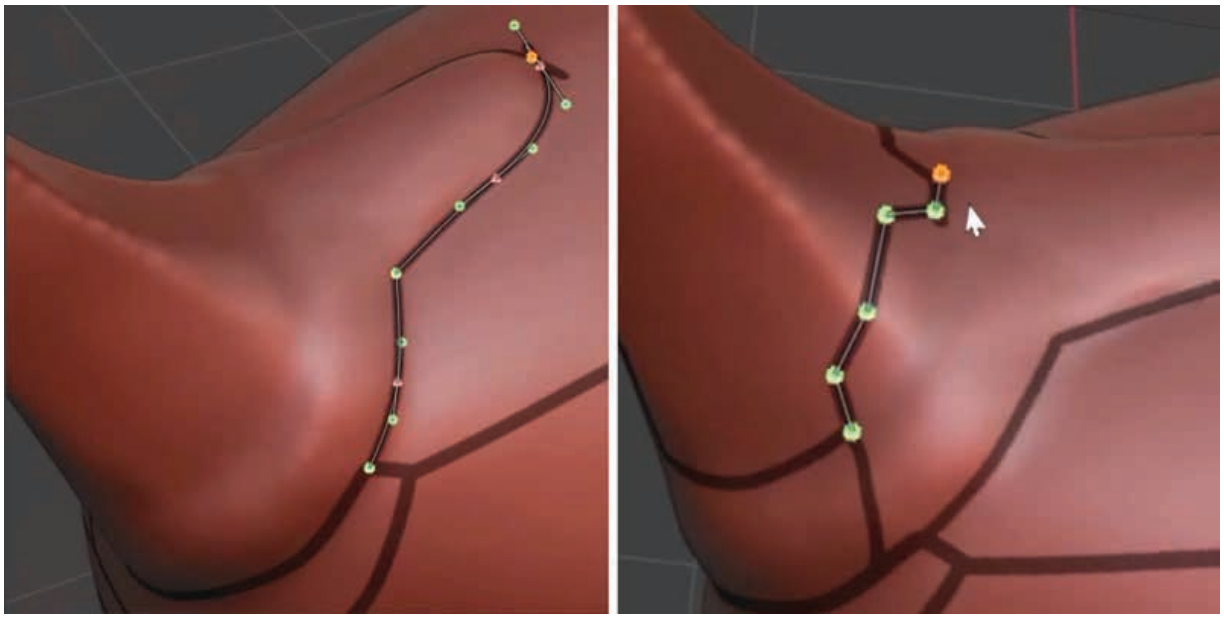
Mask Extract

One way to work with high polygon counts is to split an object into several smaller objects. Once you've created a rough base shape, you can use **Mask Extract** to extract geometry. Note, that Mask Extract doesn't work on objects with a Multires modifier on them.

Draw a **Mask**. Then **Press Alt+E** and choose a more hard surface friendly stroke method, such as Line or **Curve**.



Draw some panel cuts.

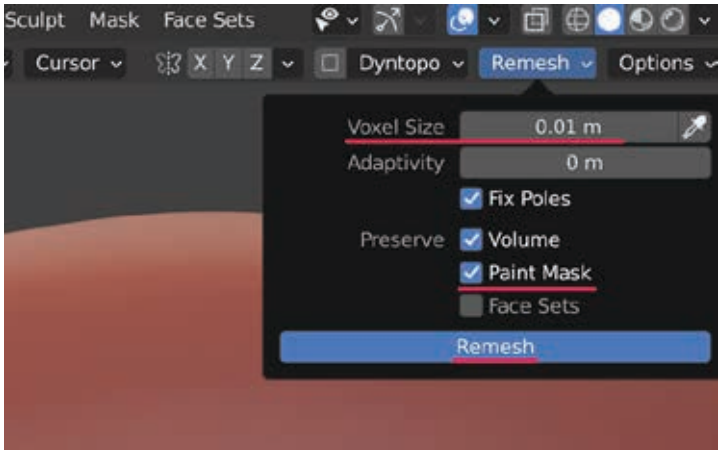


Click here to return to the index.

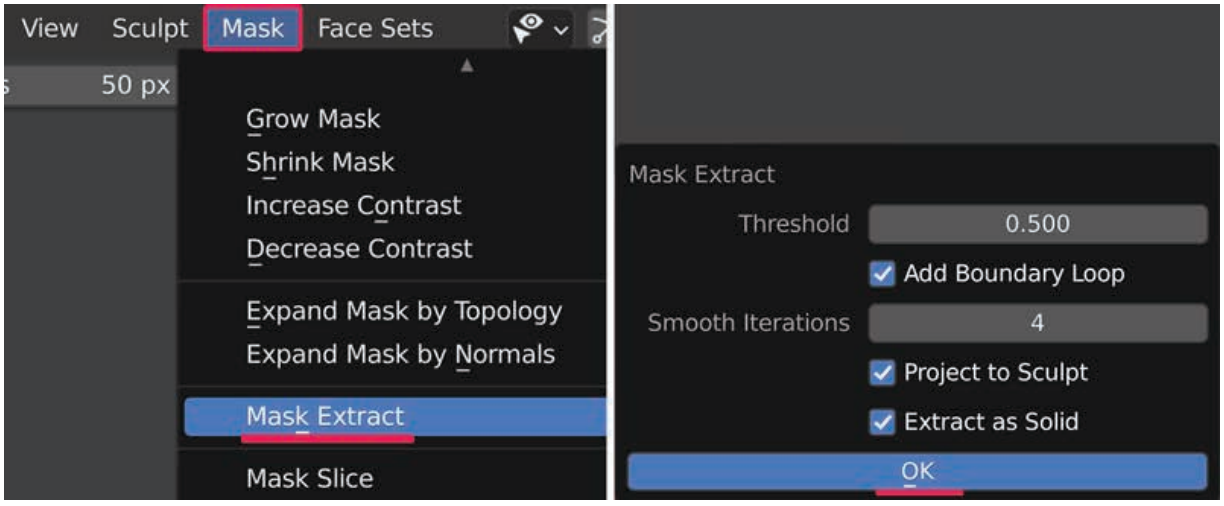
Then invert the mask with **Ctrl+i**.

In this case I want to remesh the object because I don't need that much geometry at this stage and I don't want everything to be slowed down.

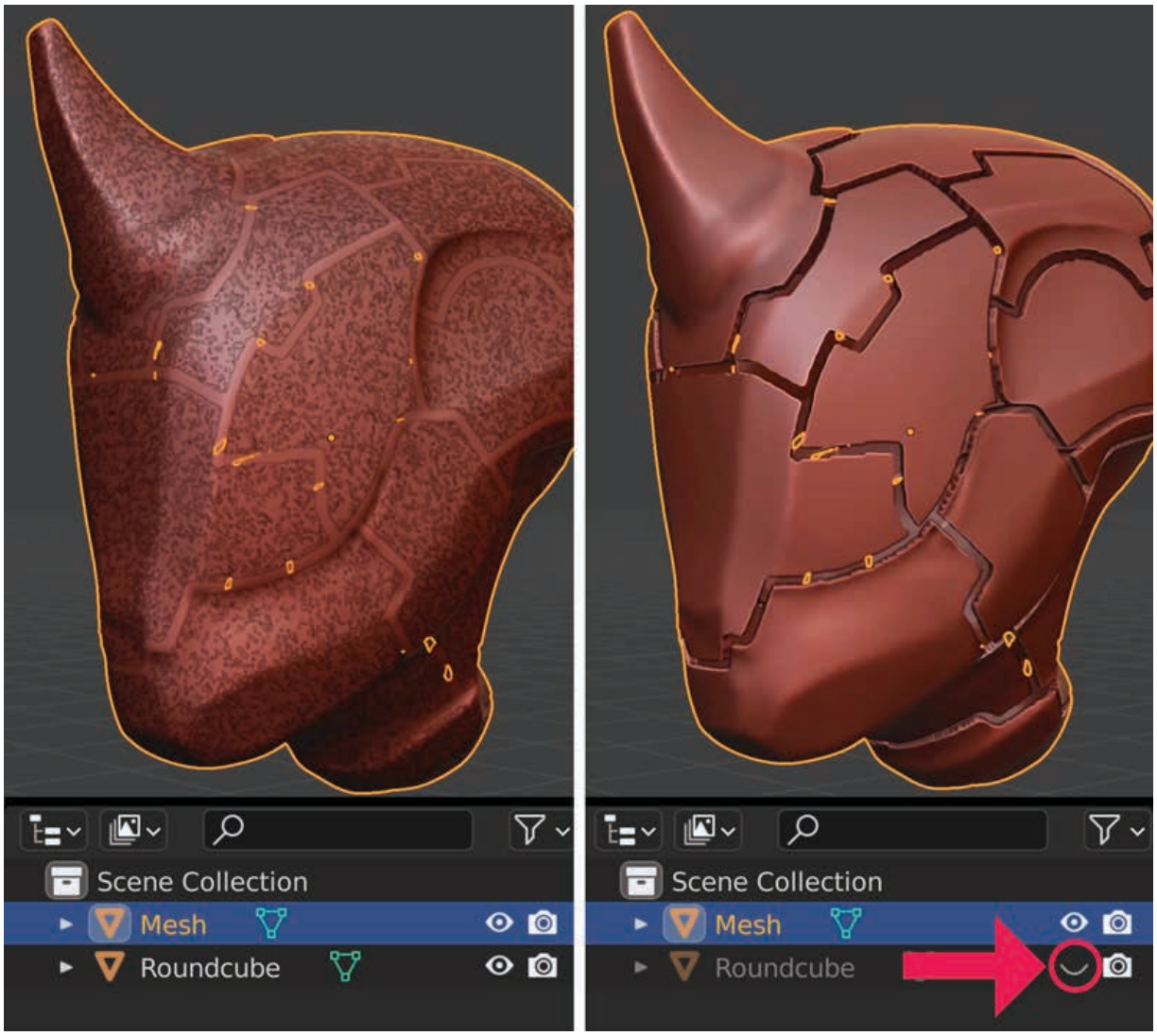
In the **Remesh options**, check **Paint Mask** so you can remesh without losing the mask. Choose a **Voxel Size** and click on **Remesh**.



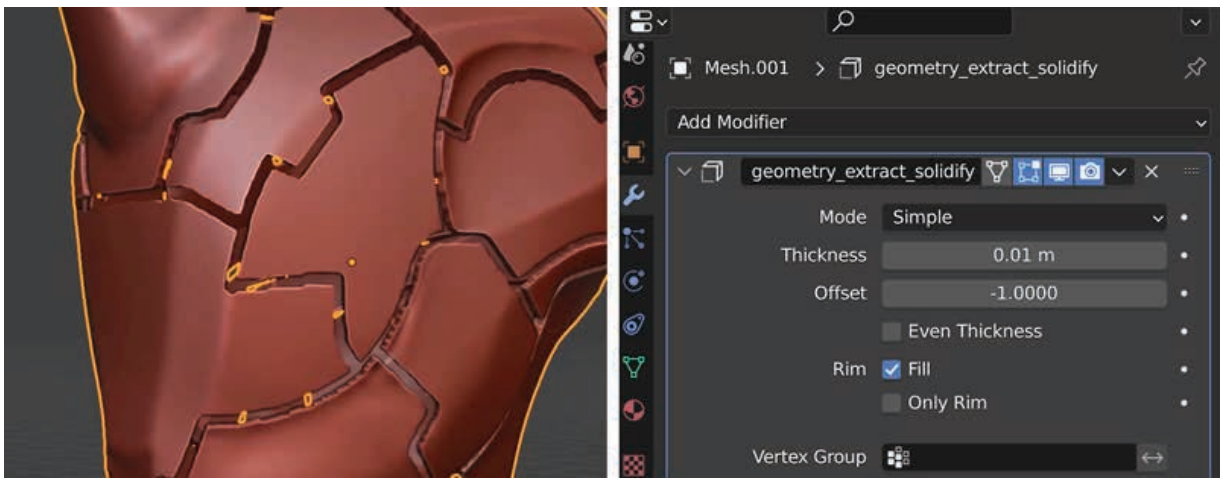
Go to Mask > **Mask Extract**. Click on **OK**.



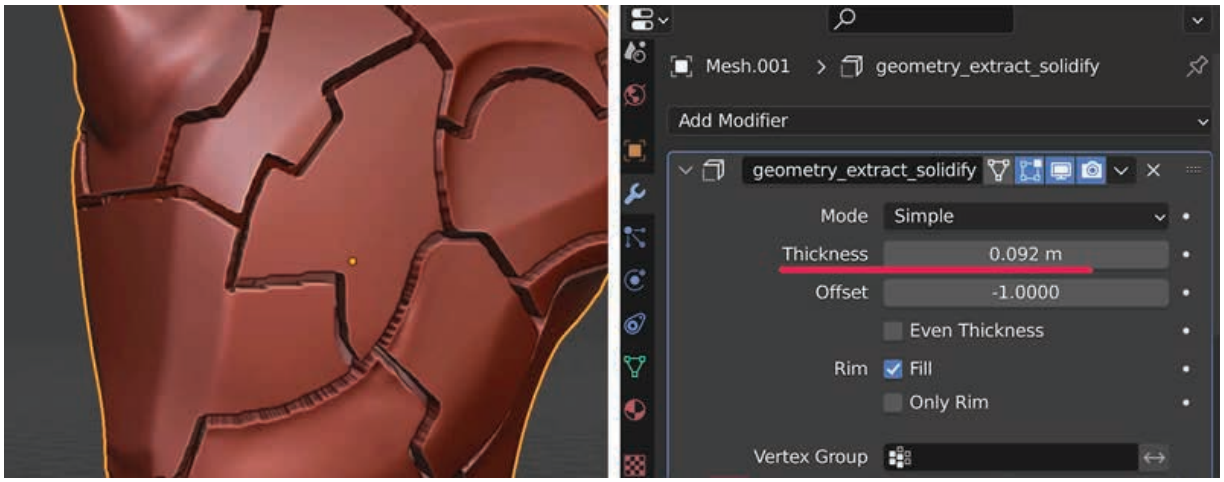
New geometry has been created based on the masked parts. We can hide the original Roundcube in the Outliner.



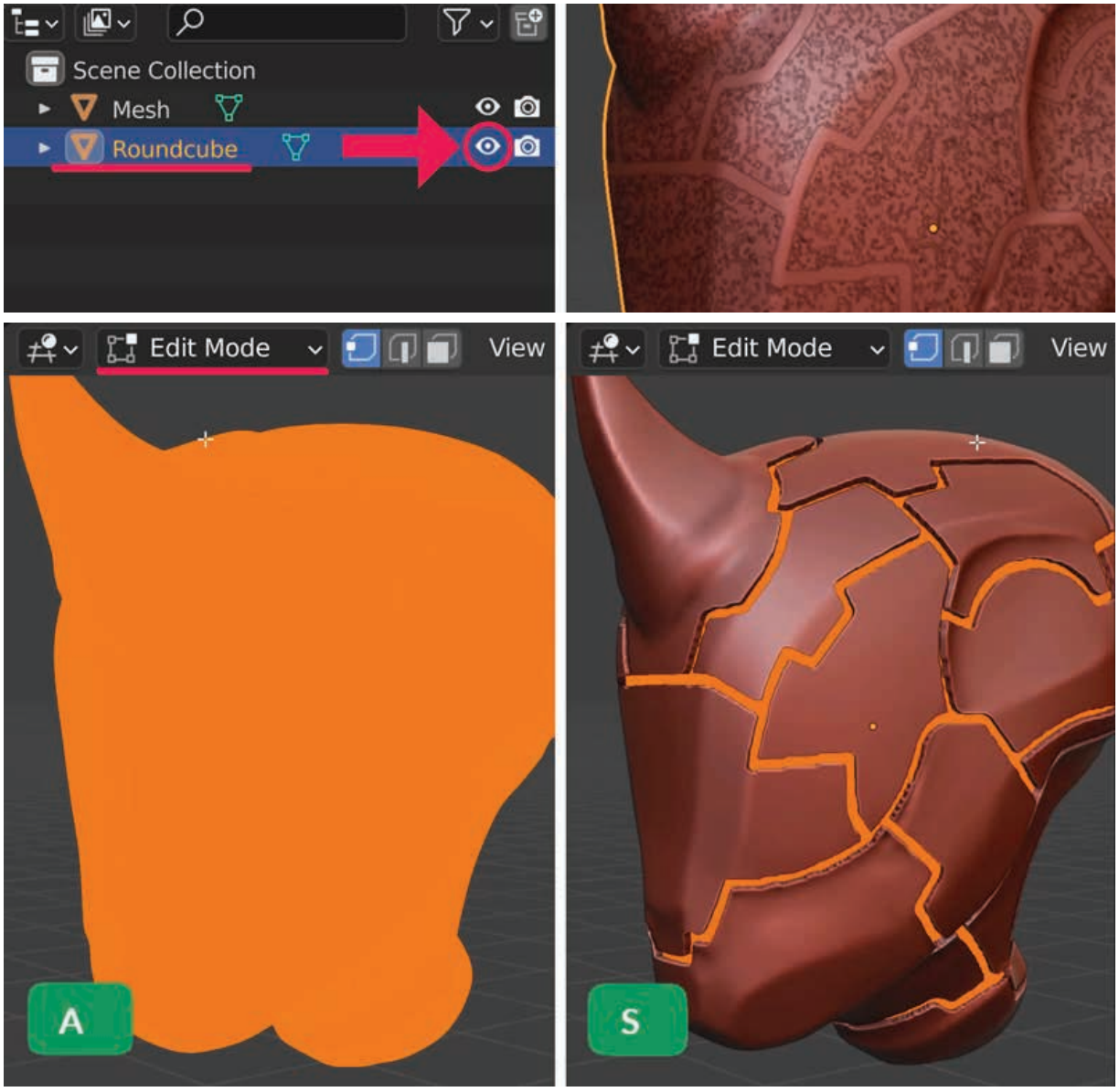
Solidify modifier was added automatically.



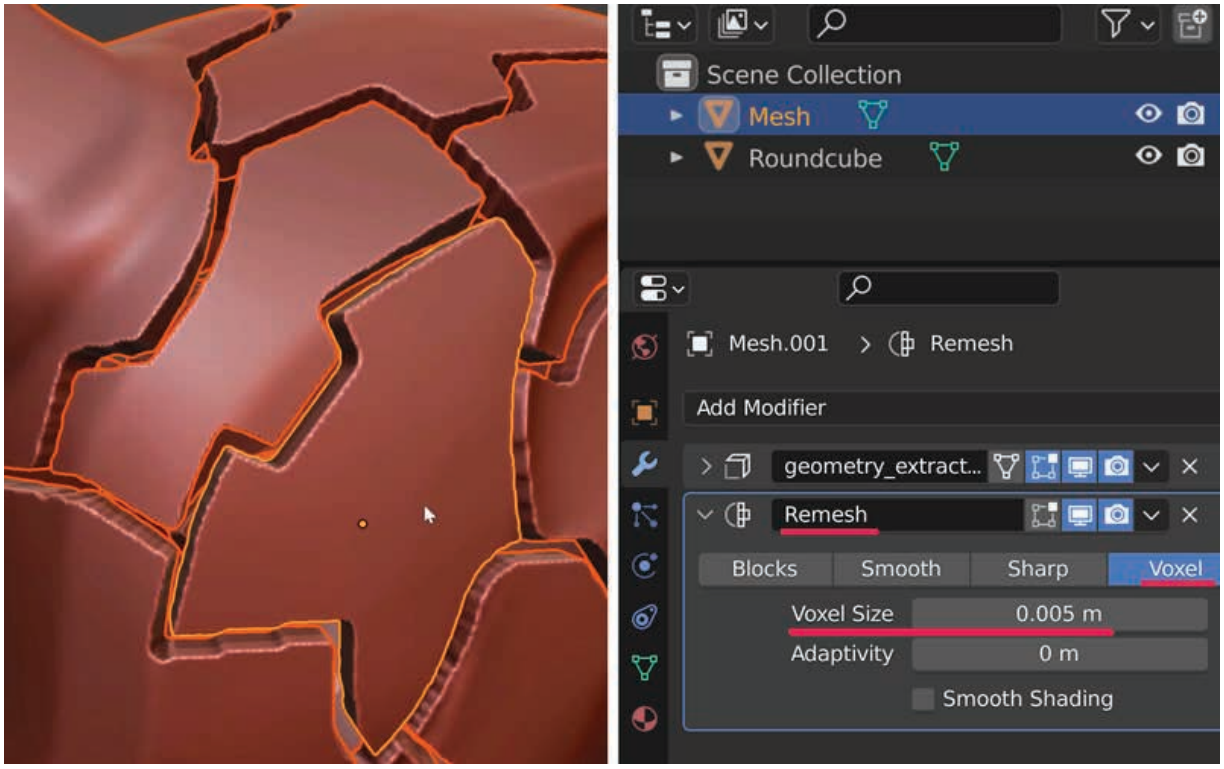
We can increase the **Thickness** of the extracted geometry.



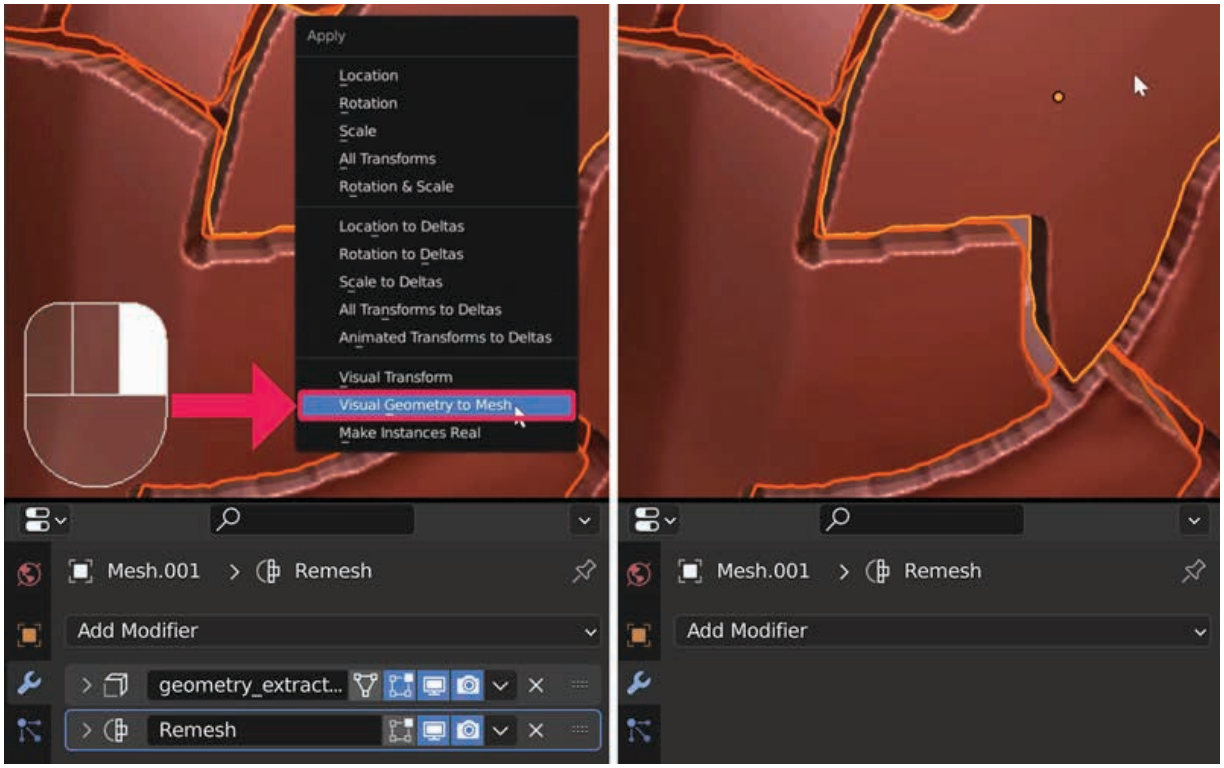
If we unhide the Roundcube, we can select all vertices in Edit Mode and scale them down.



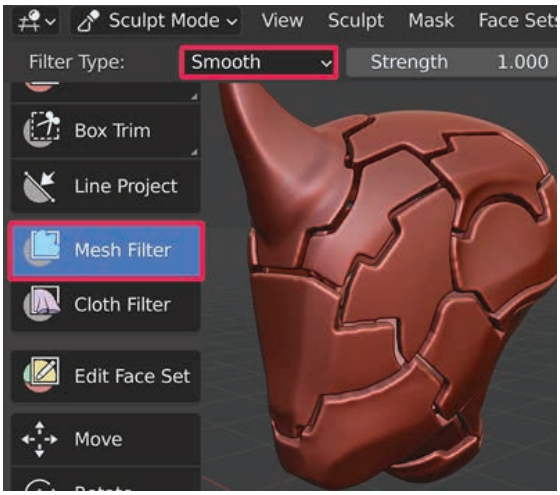
Add a **Remesh modifier** to the new geometry and set a **Voxel Size** that looks okay. It doesn't have to be too high res.

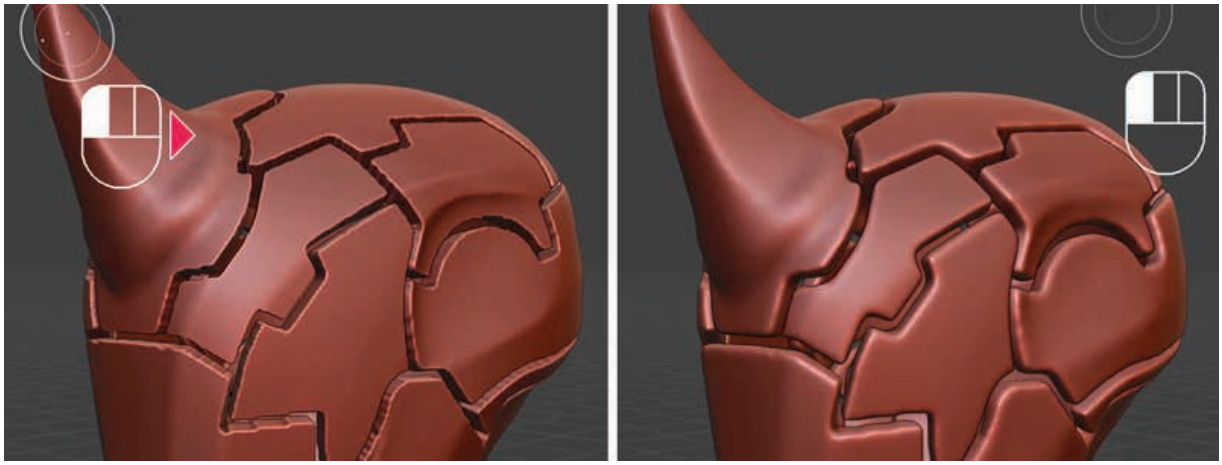


Then right-click and choose **Visual Geometry to Mesh** to apply all modifiers.

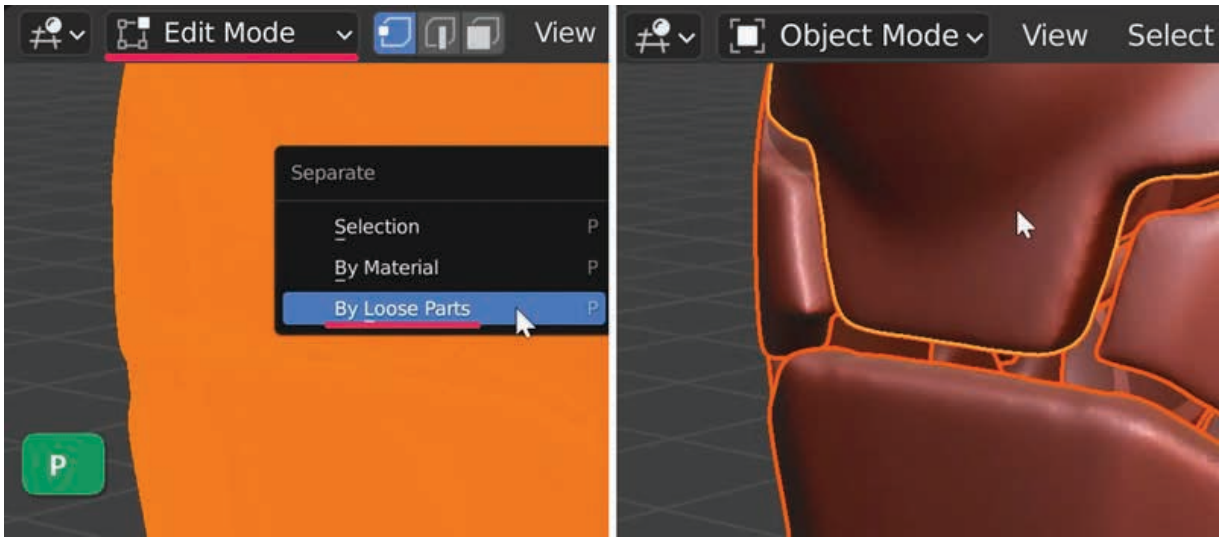


Using the **Mesh Filter** set to **Smooth**, we can smooth out the remeshed geometry.

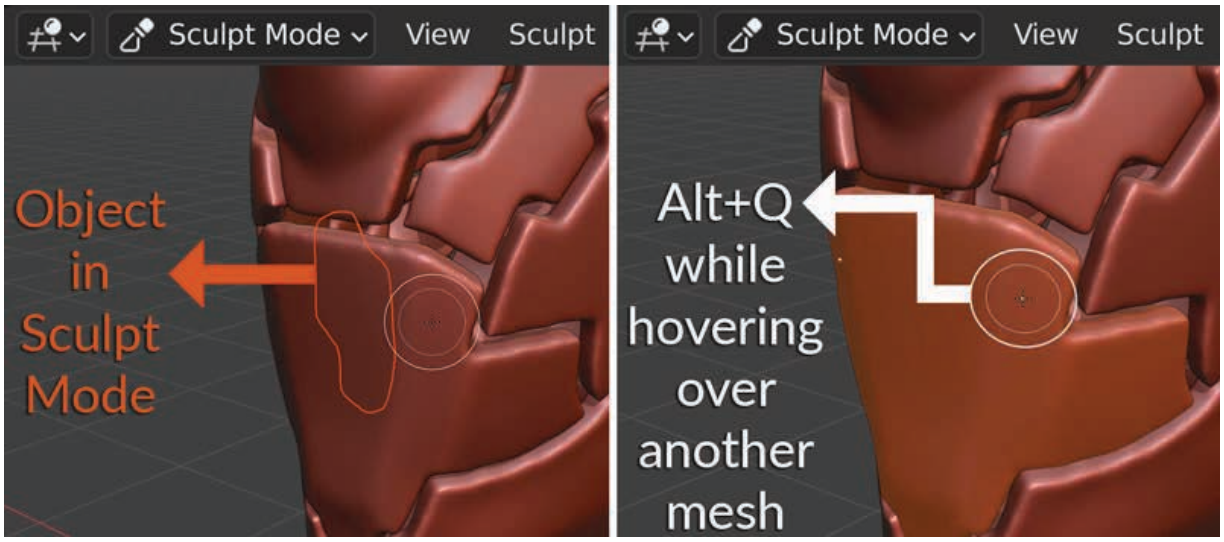




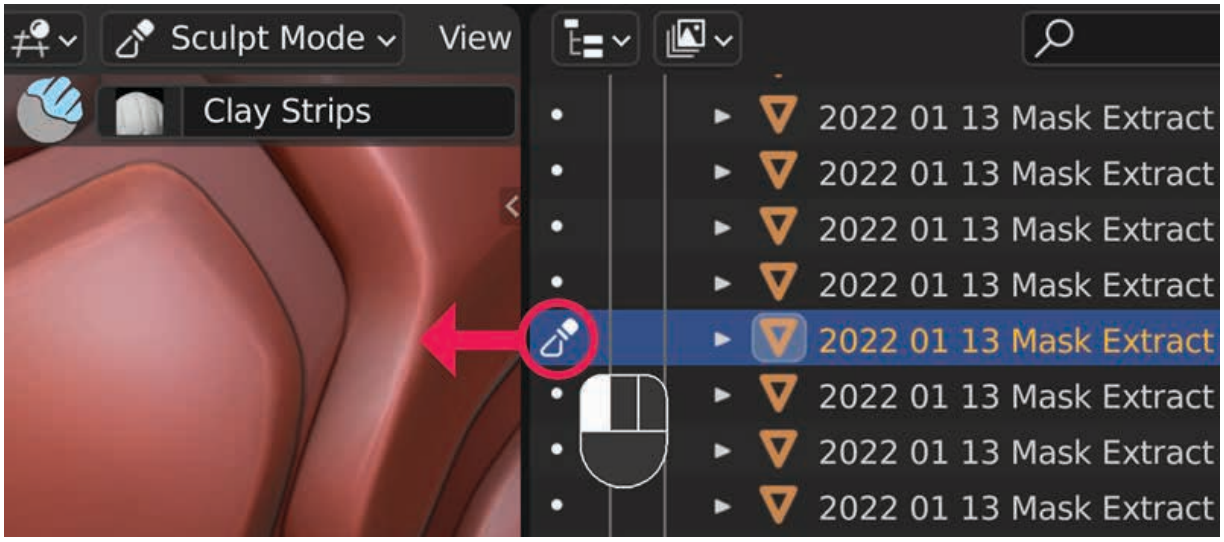
If you want to separate these into individual objects, select all in Edit Mode and **press P**. Then choose **Separate by Loose Parts**.



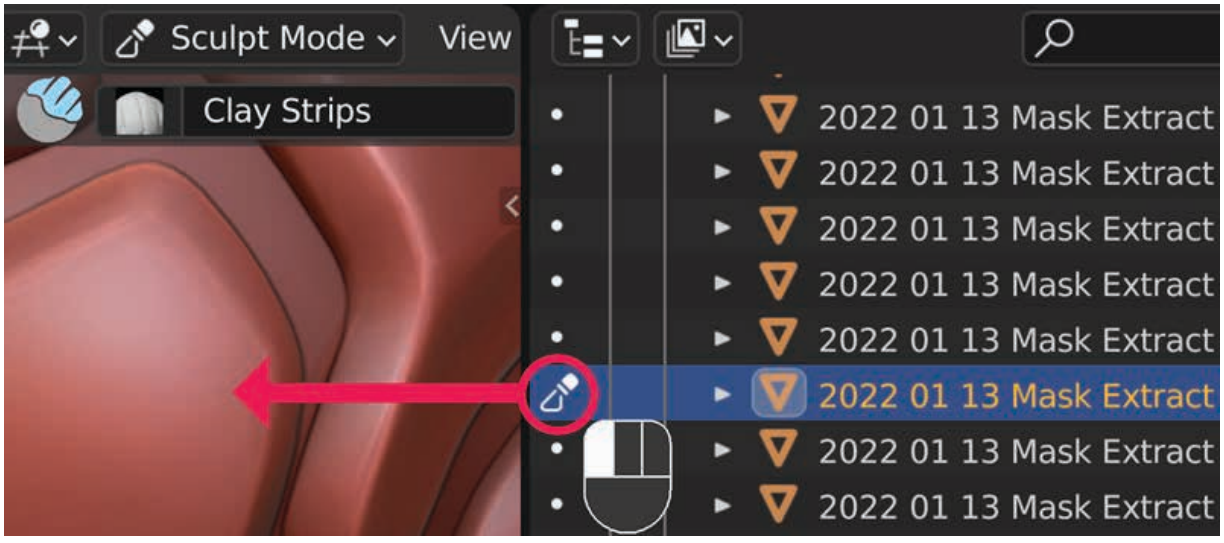
Hover over an object and **press Alt+Q** in **Sculpt Mode** to sculpt on it.



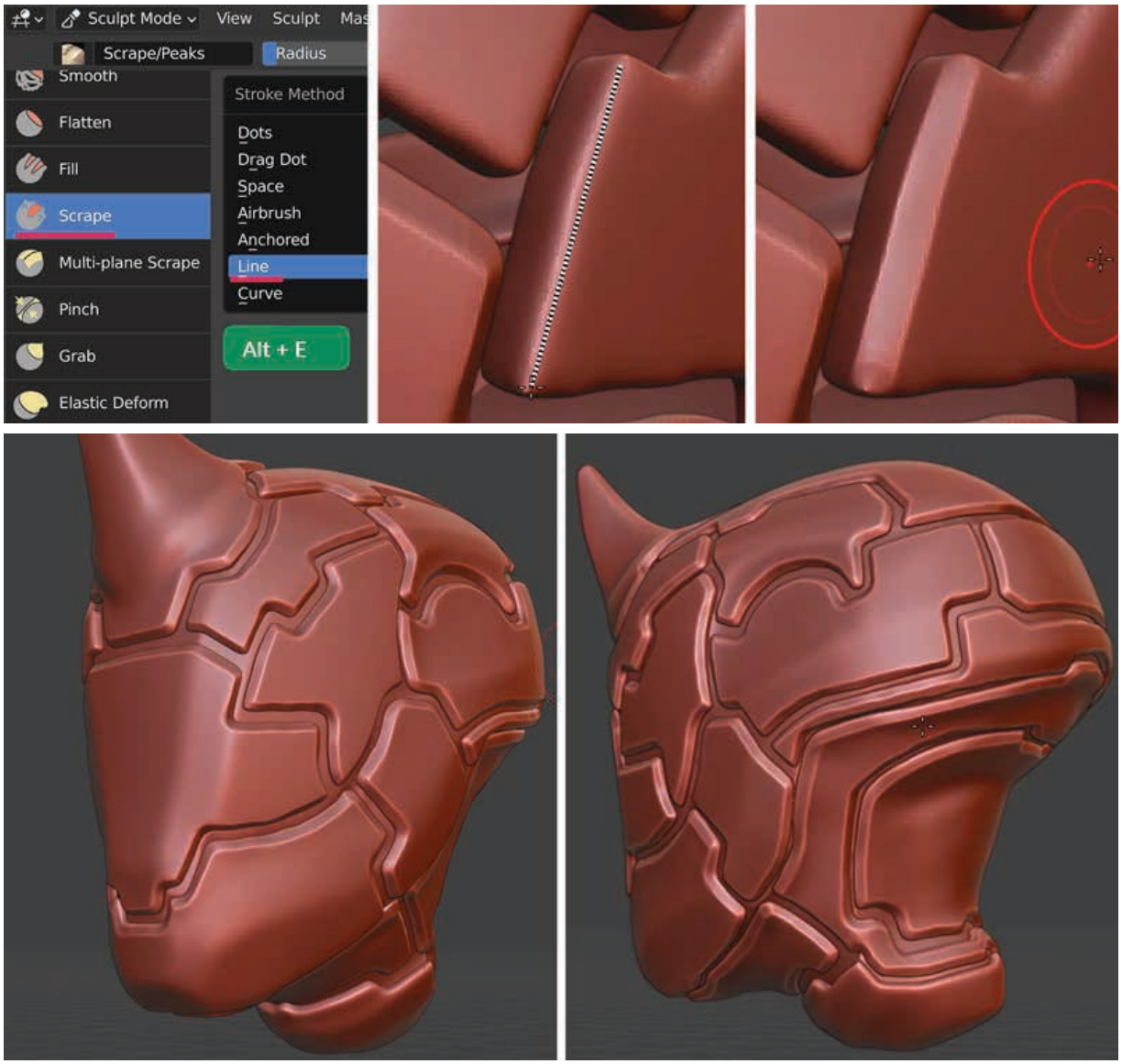
Or click left of the item in the Outliner.



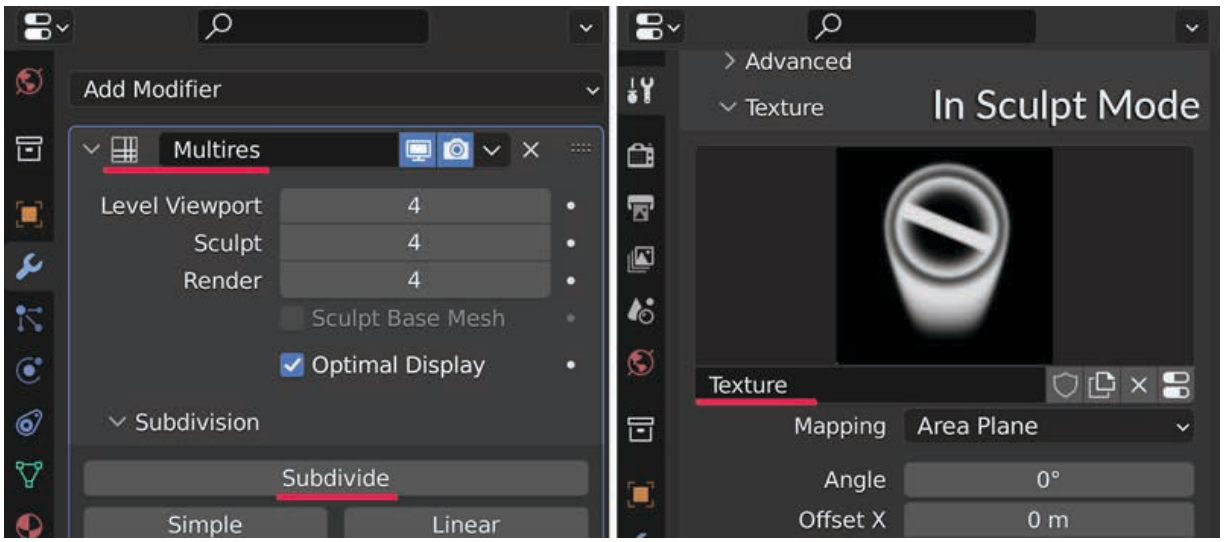
The current object selected for sculpting is then indicated with a brush icon.

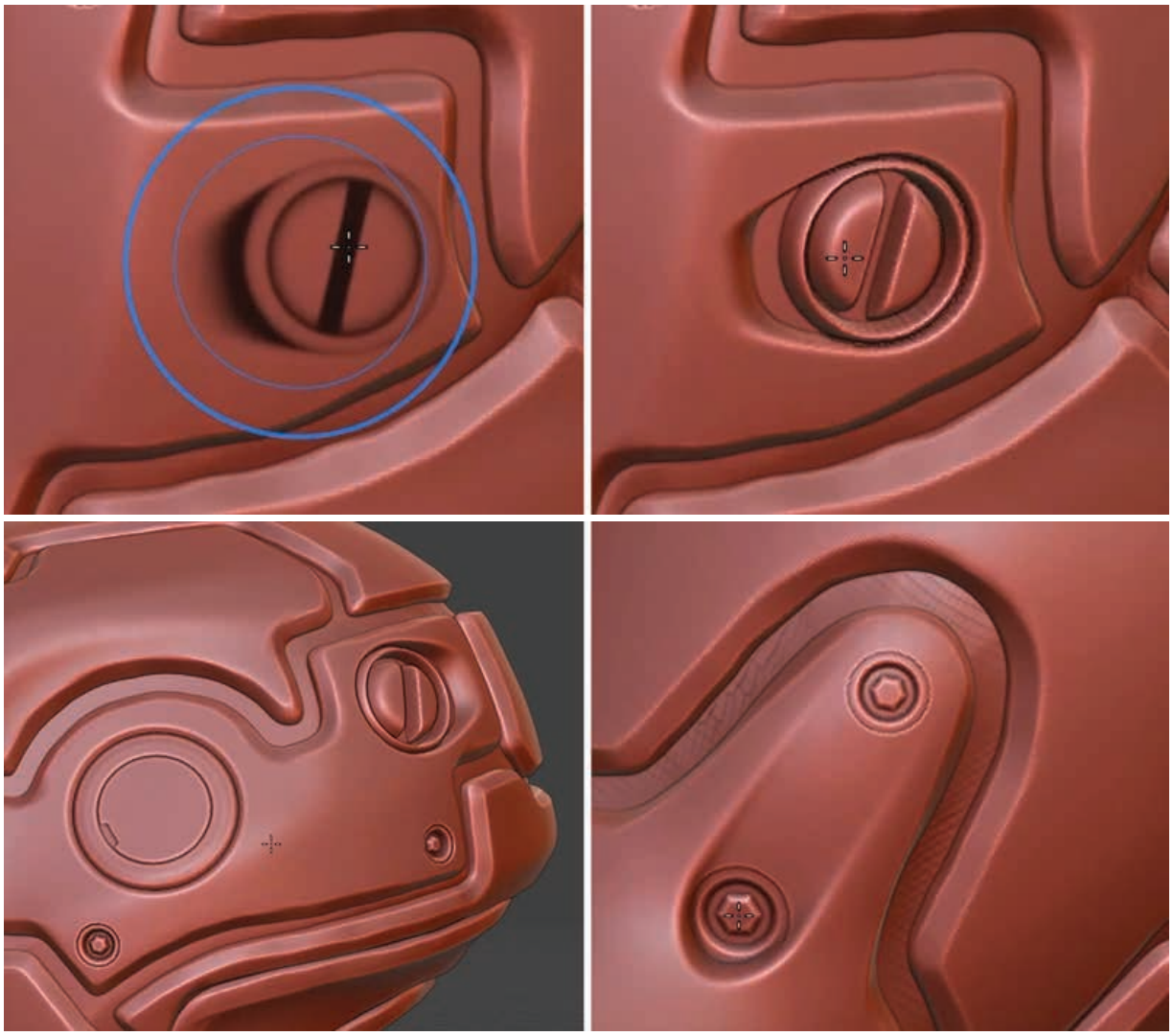


Now we can use the Scrape brush in combination with a Line or Curve stroke method to add some beveling to the edges of these objects.

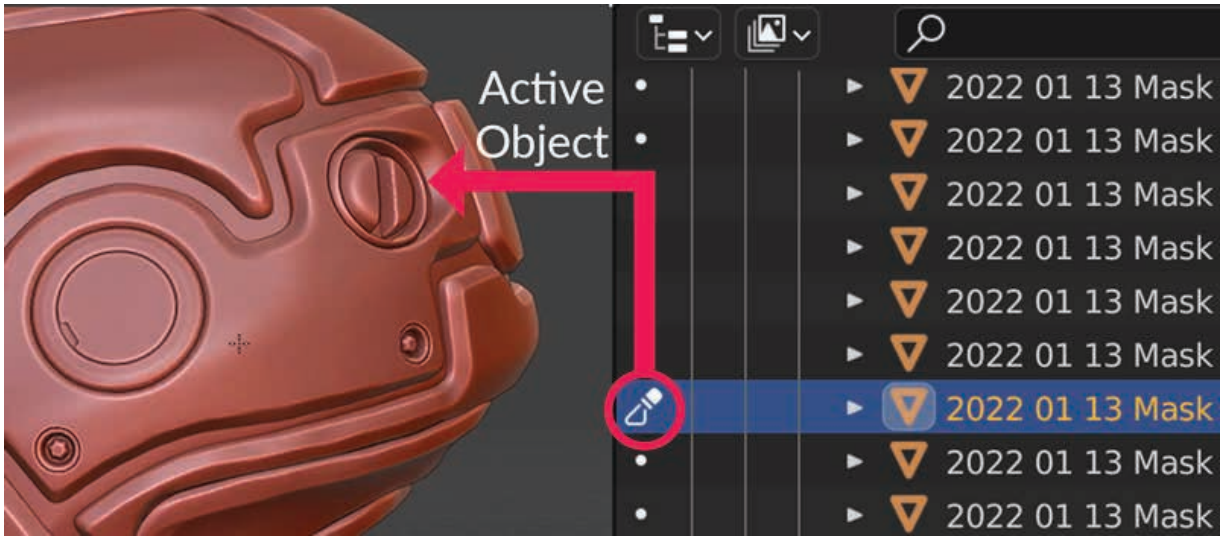


And using a **Multires modifier** to increase the resolution, we can add **Alpha textures** to the individual objects.





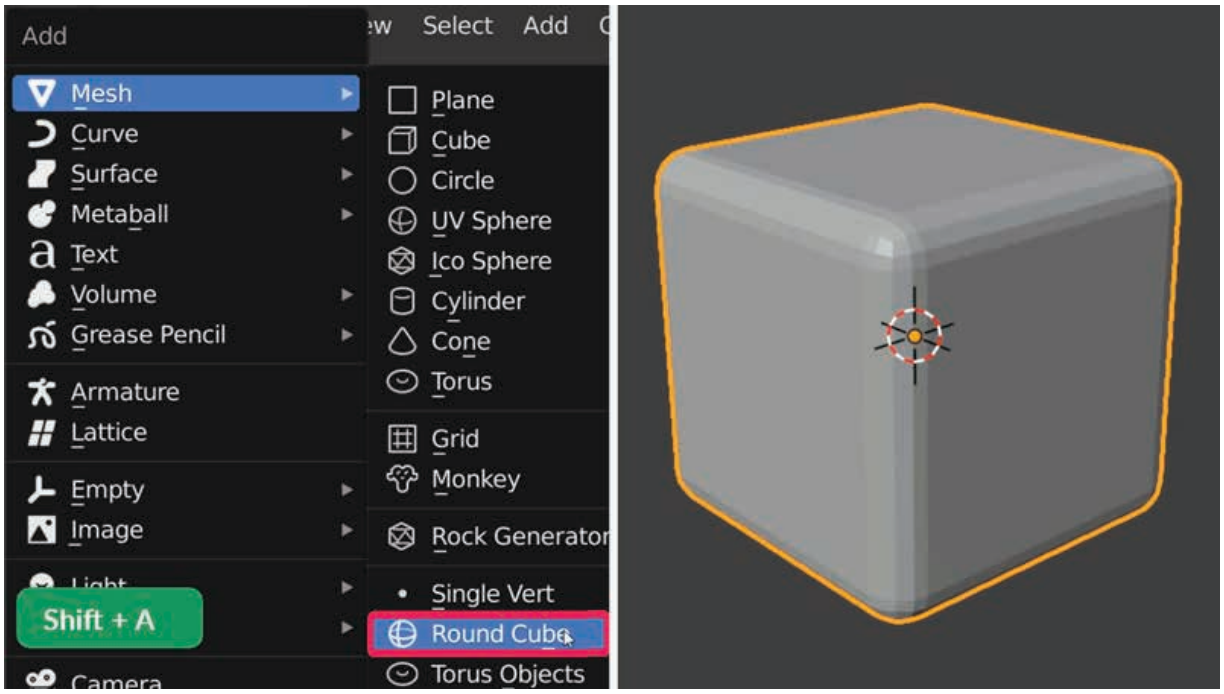
For more info about this, watch the previous Hard Surface Sculpting tips video.
By default only the detail on the currently active object is shown, in order to keep performance optimal.



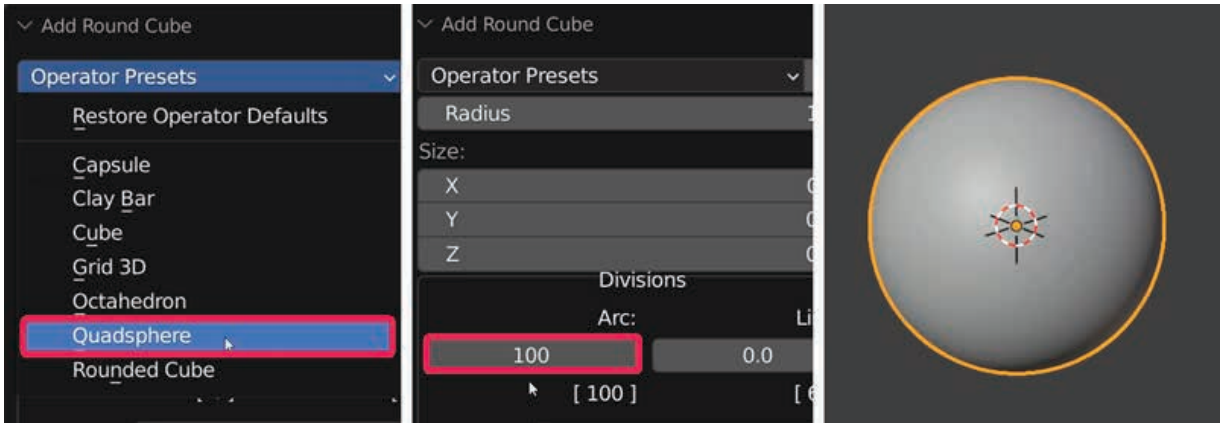
Line Project

Using **Line Project** we can quickly create hard surface shapes.

Add a **Round Cube**.

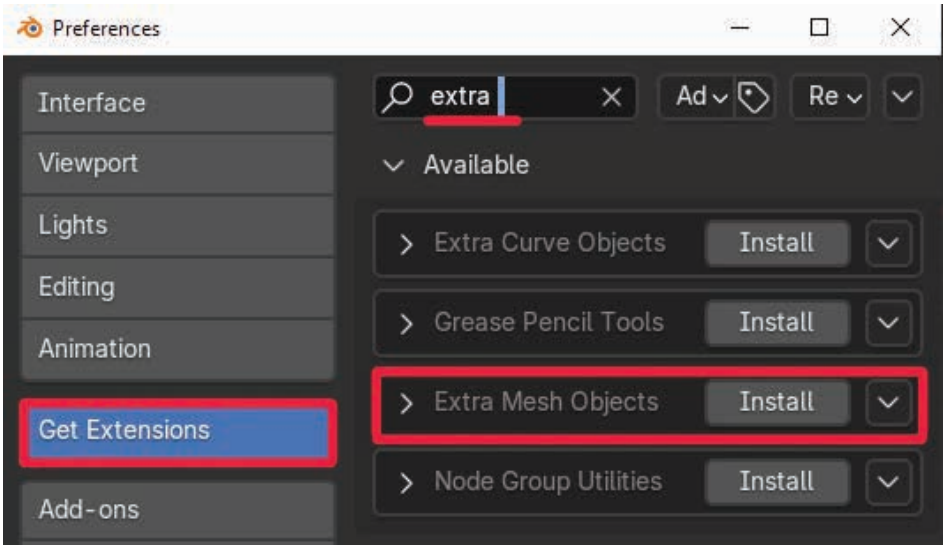


Set it to **Quadsphere** with an **Arc** value of 100 or more.

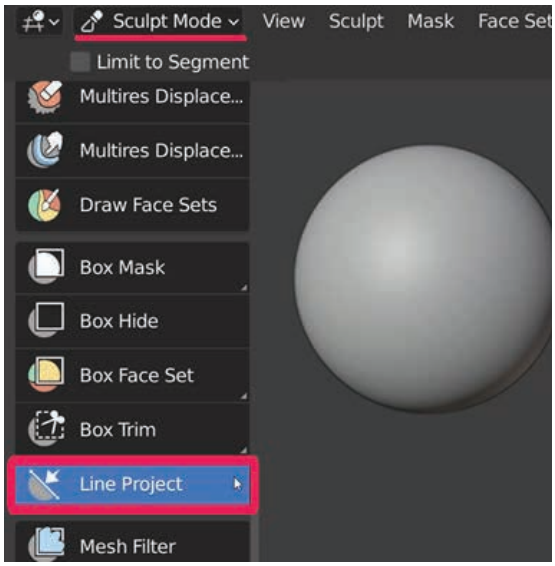


This is available after you enable the **Extra Mesh Objects add-on** in Preferences.

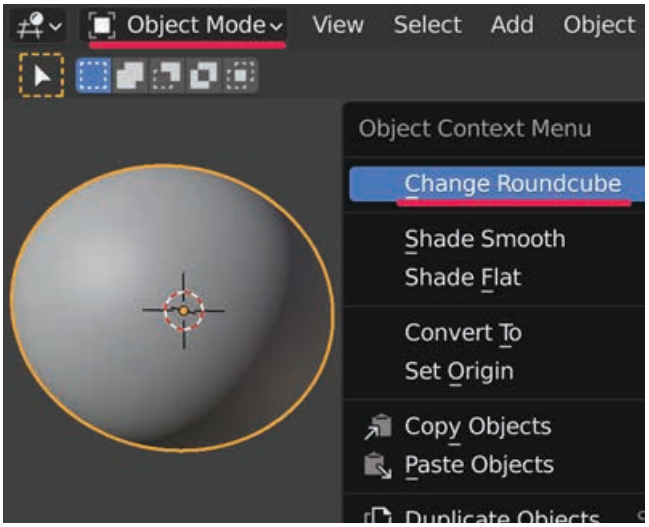
You can find it in the Get Extensions tab.



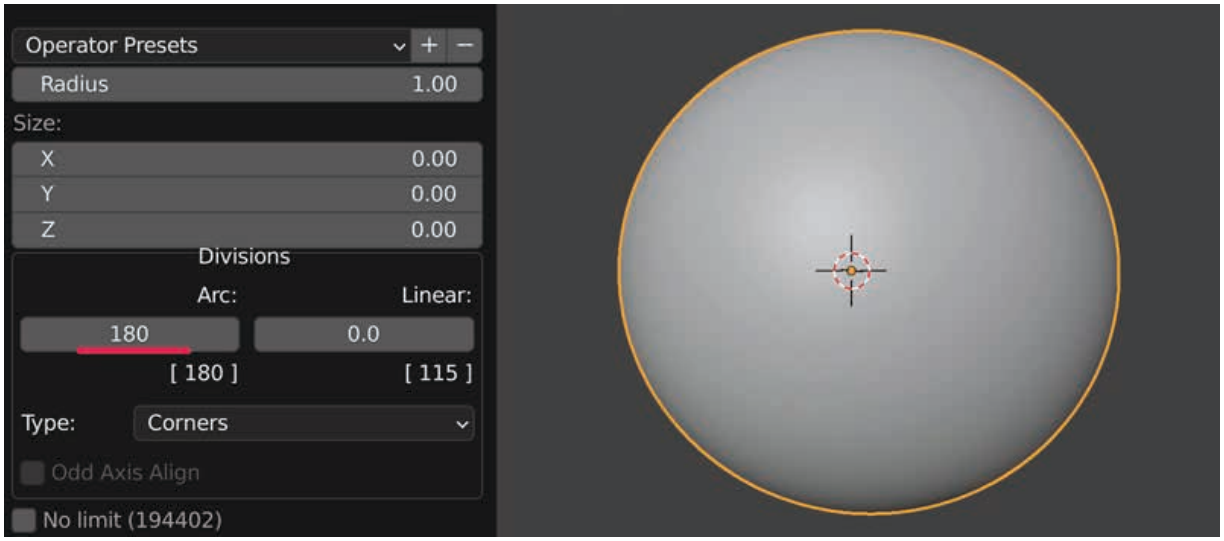
In **Sculpt Mode** select the **Line Project** tool.



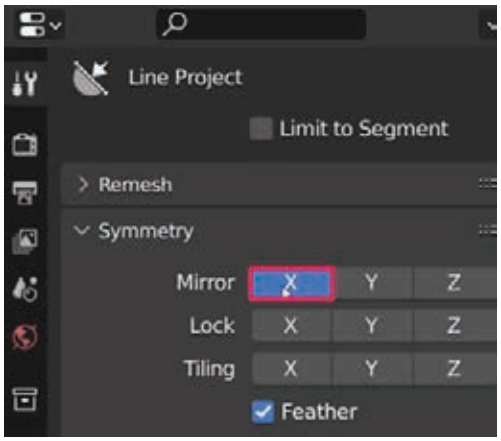
In case your Quad Sphere is too low res, go back to Object Mode. Right-click and choose **Change Roundcube**.



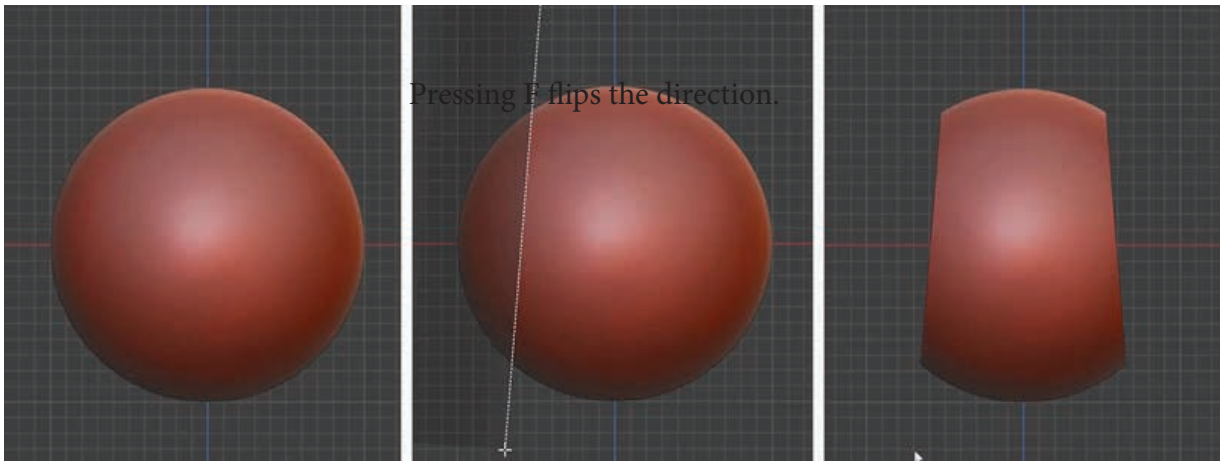
Increase the **Arc** value.



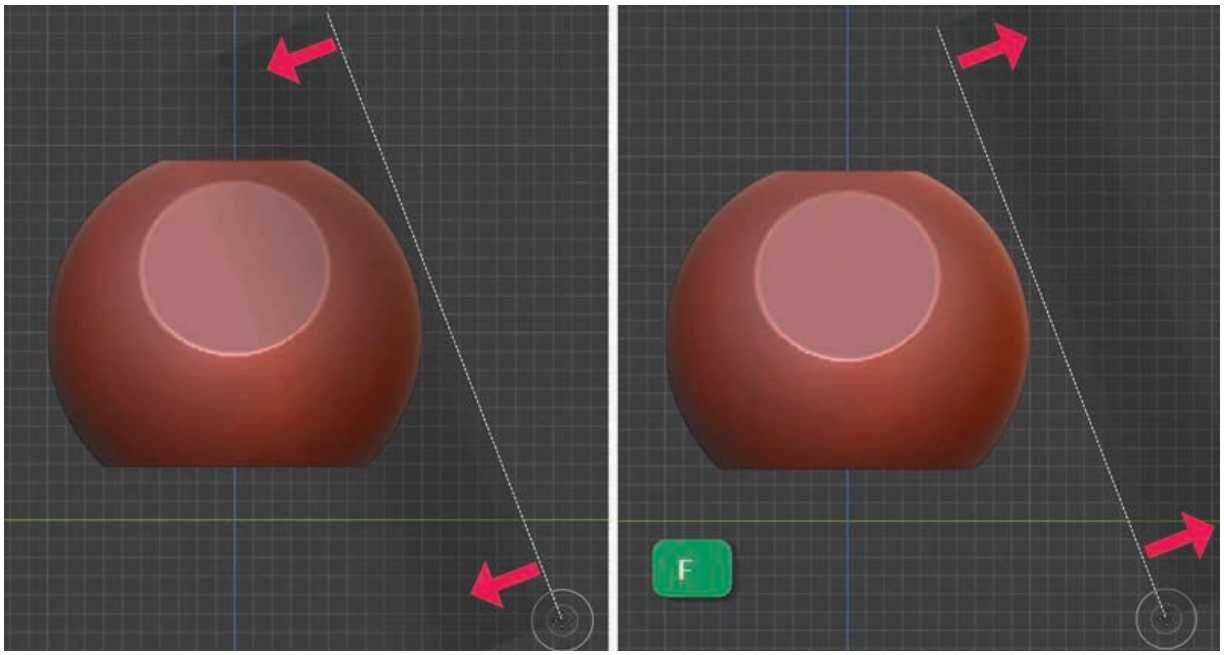
Enable mesh **symmetry.**



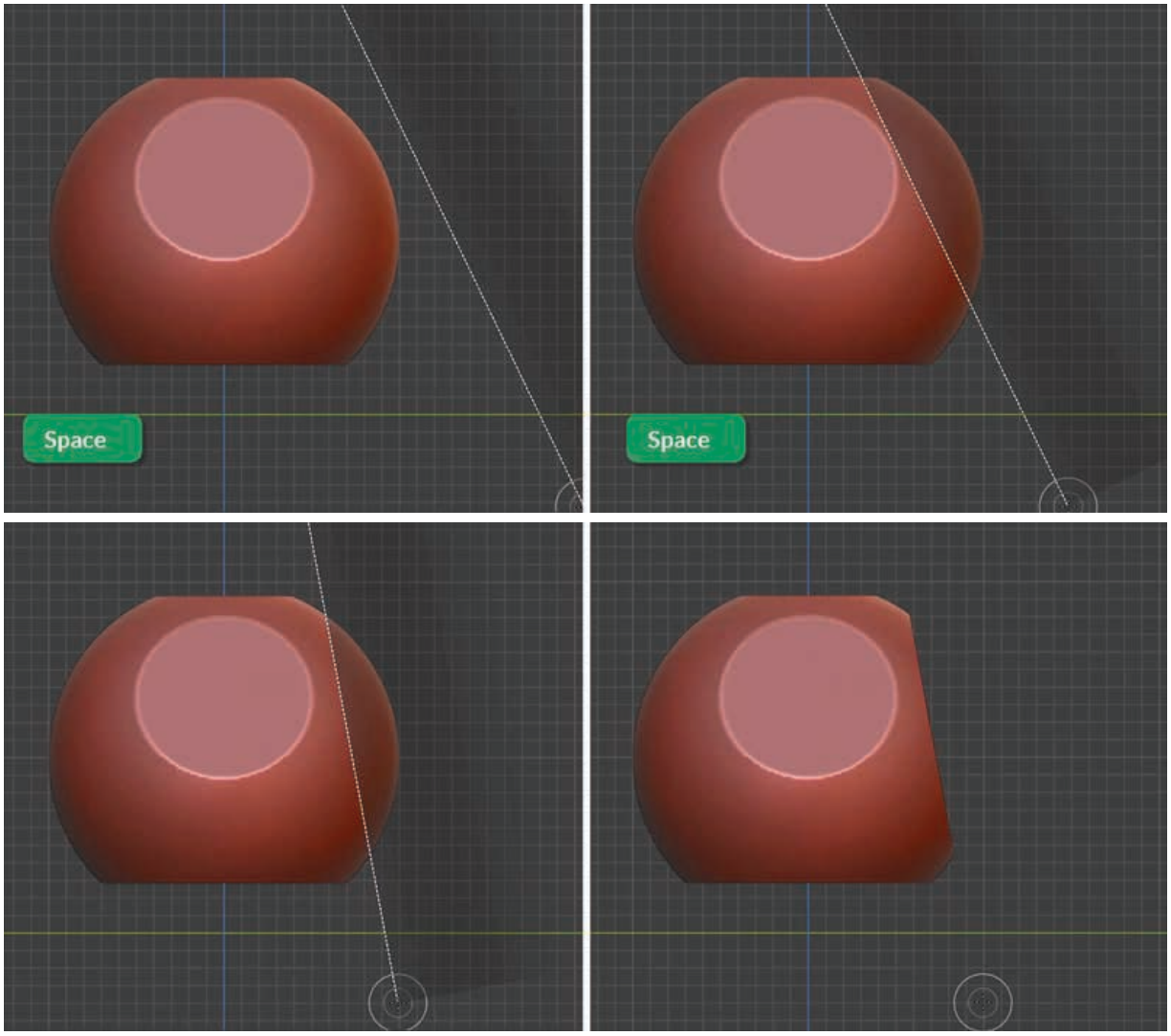
Drag the left mouse button to use the **Line Project** tool.



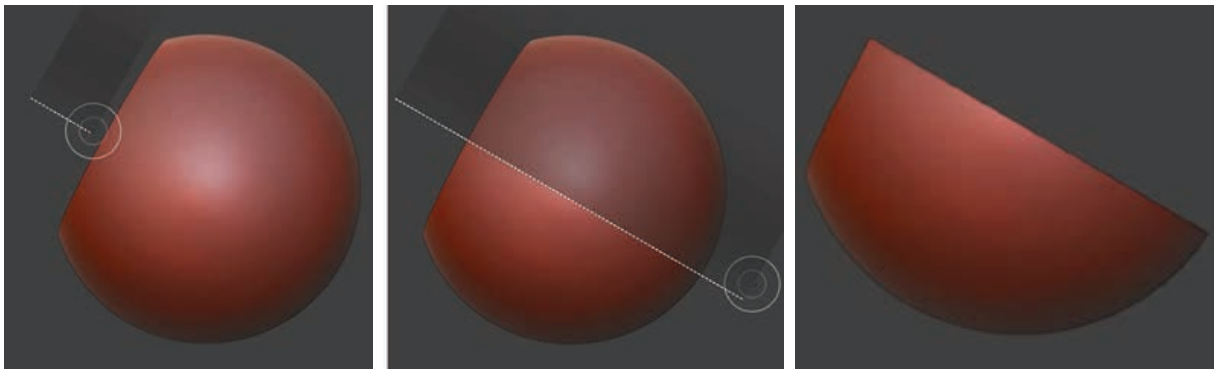
Pressing F flips the direction.



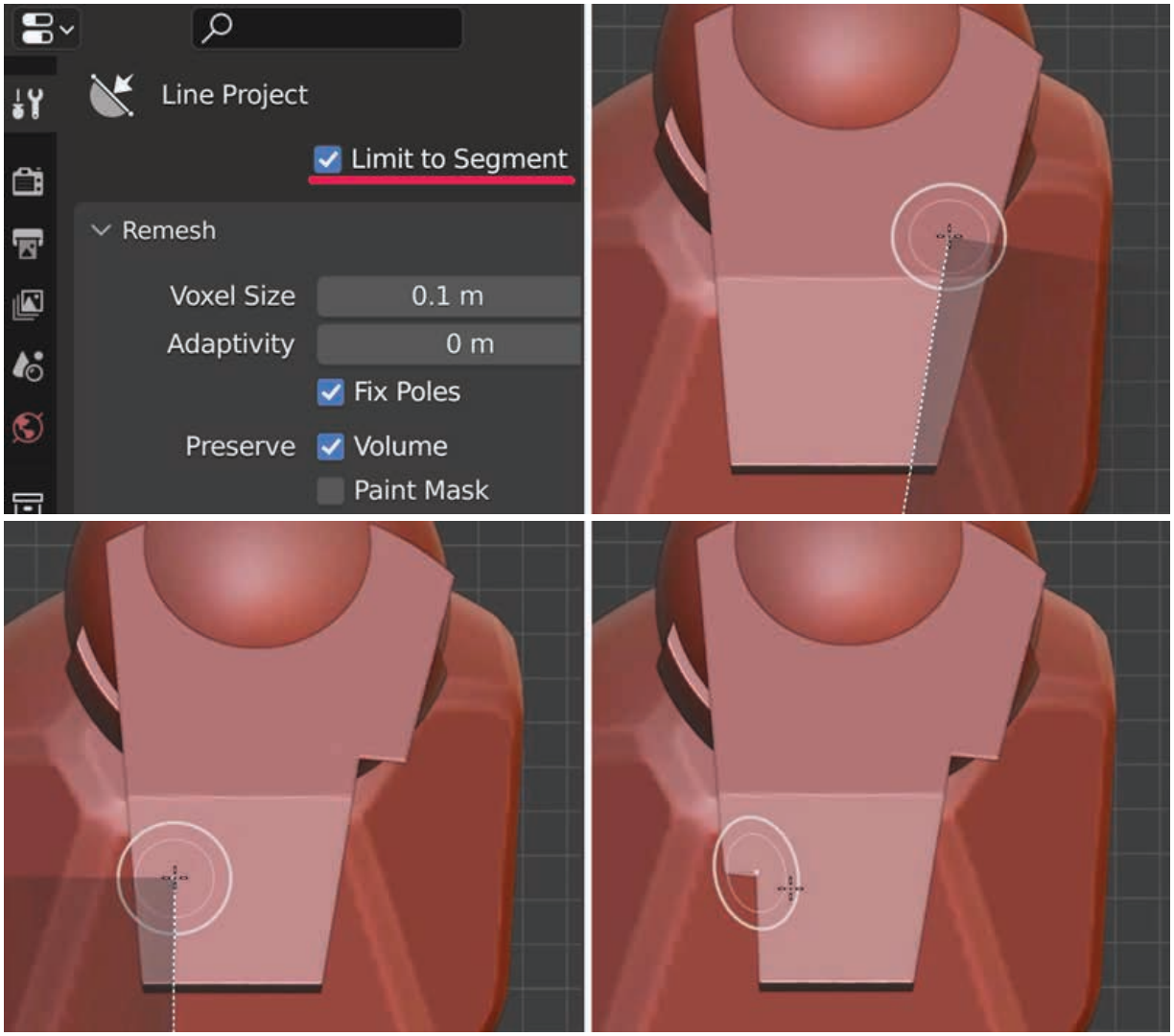
You can **hold the Spacebar** to move the position of the line.



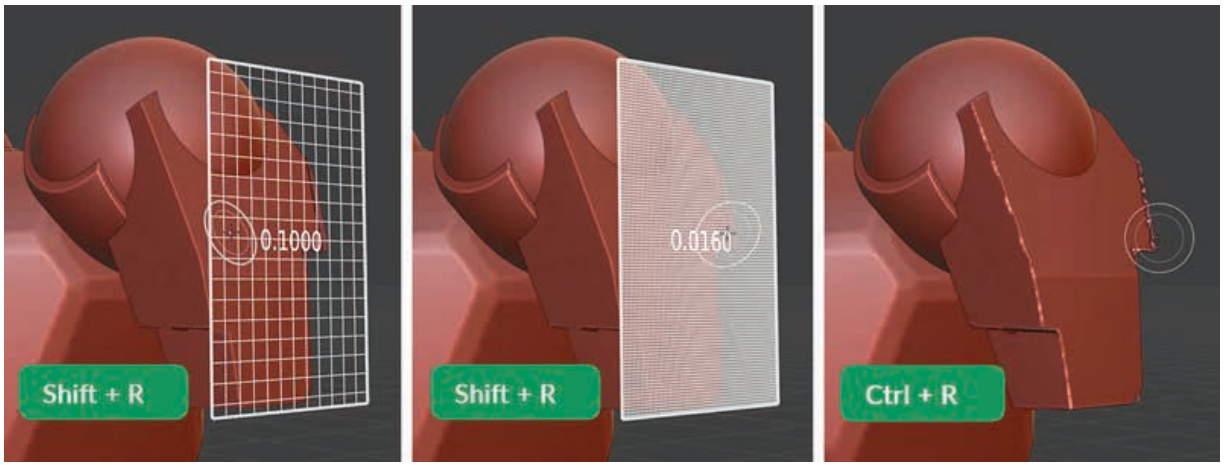
Hold Ctrl to constrain the angle.



Check **Limit to Segment** if you want to limit the effect to where you draw the line.

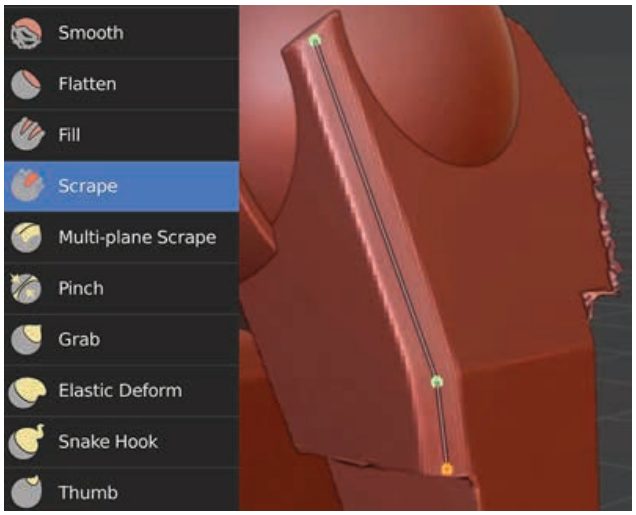


The **Line Project** tool flattens geometry, it doesn't actually do any boolean cutting. So it's a good idea to remesh after you've finished using it. **Press Shift+R** to preview the remesh resolution. Then **press Ctrl+R** to remesh the geometry.

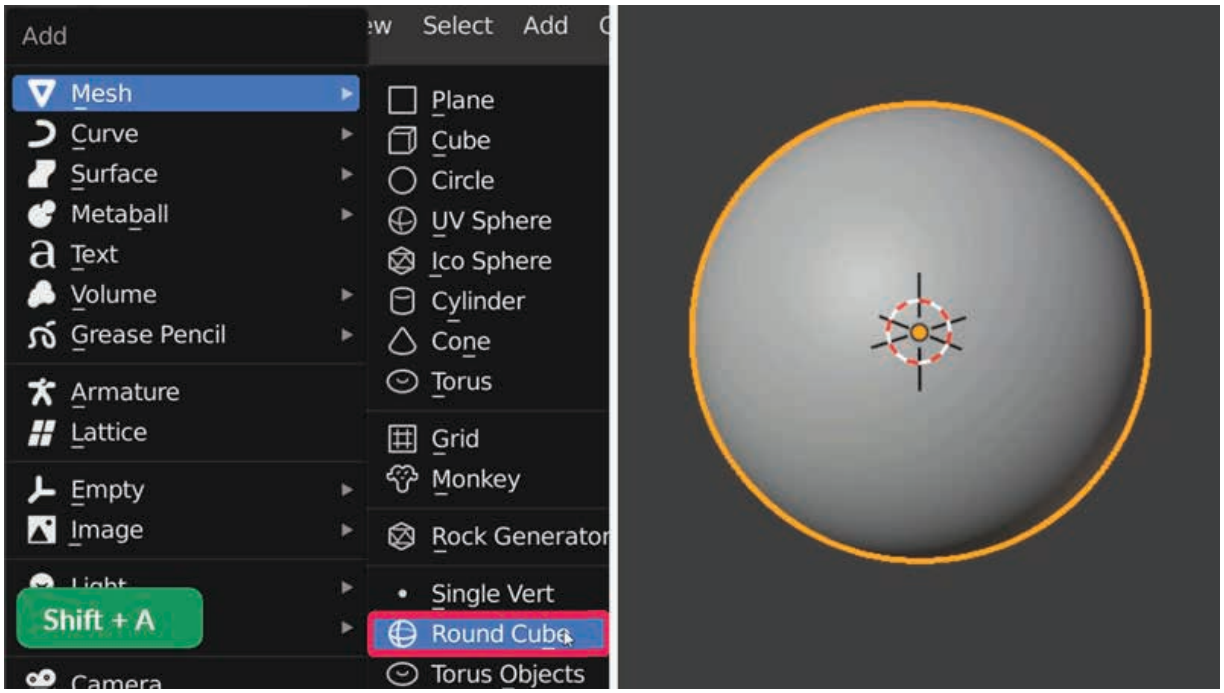


This will make it easier to continue sculpting on it.

For example using the Scrape brush in conjunction with the Line Stroke Method to create a bevel on the edge.

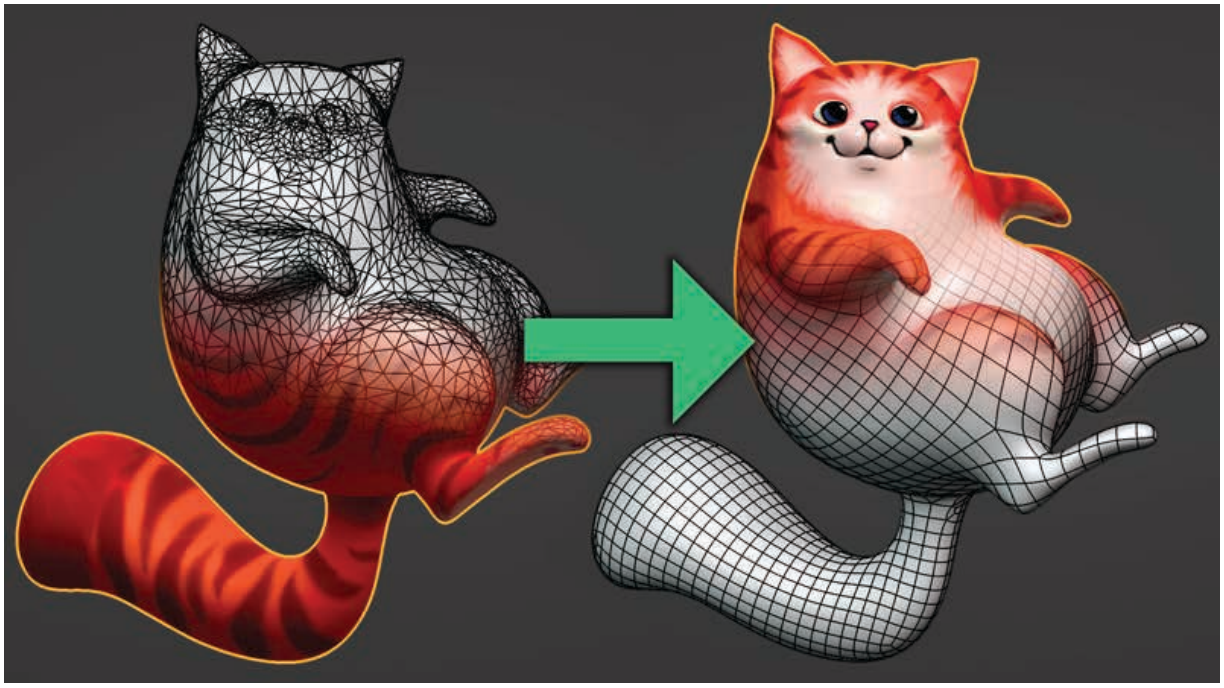


When you add another Round Cube, it will remember the previous settings you used.

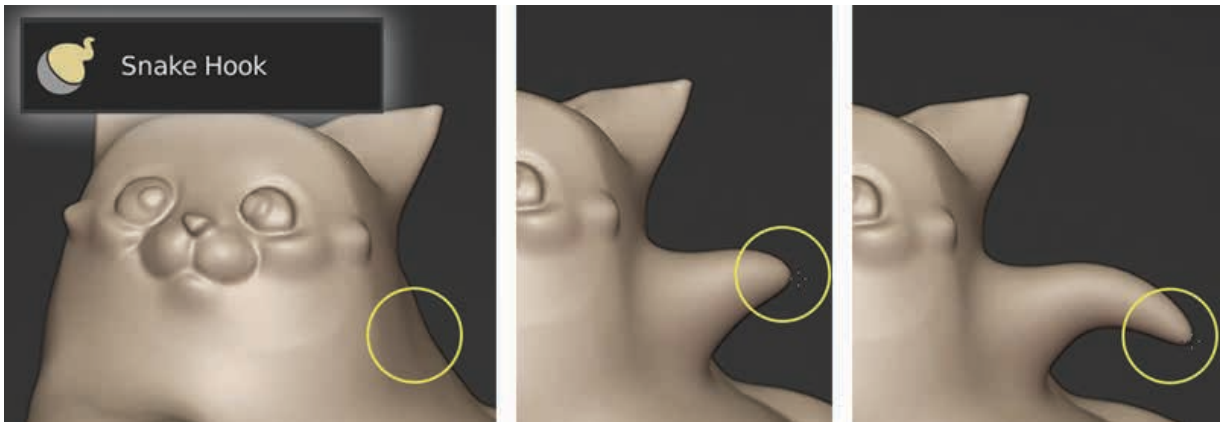


Voxel and Quad Remesh in Sculpt Mode

When sculpting, knowing the keyboard shortcuts for **Voxel and Quad remeshing** is quite useful for smoothly working on your mesh without the performance drops that you might experience with Dyntopo.



To block out shapes, the **Grab** and **Snake Hook** brushes are useful for quickly pulling geometry into a new shape.

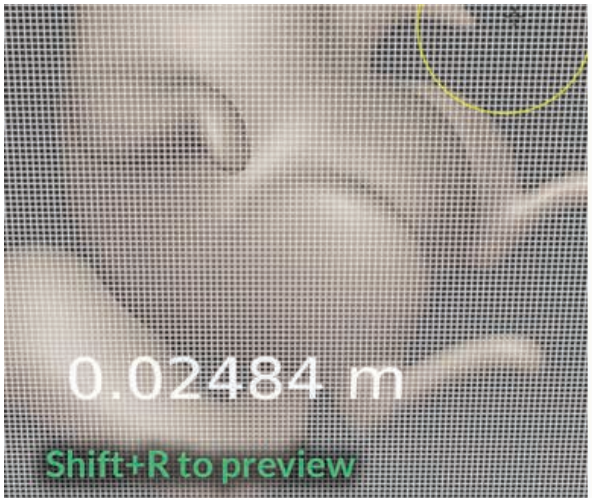


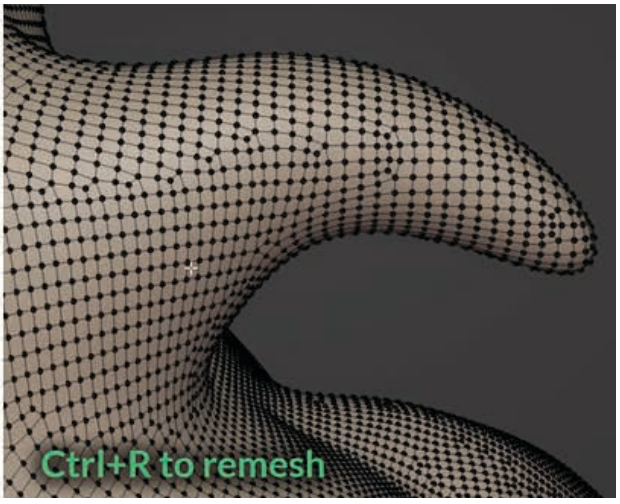
Cat model by Robin Art FX *on Sketchfab*

But when sculpting on the stretched out parts, you can see that there's not enough geometry as it's all stretched out.

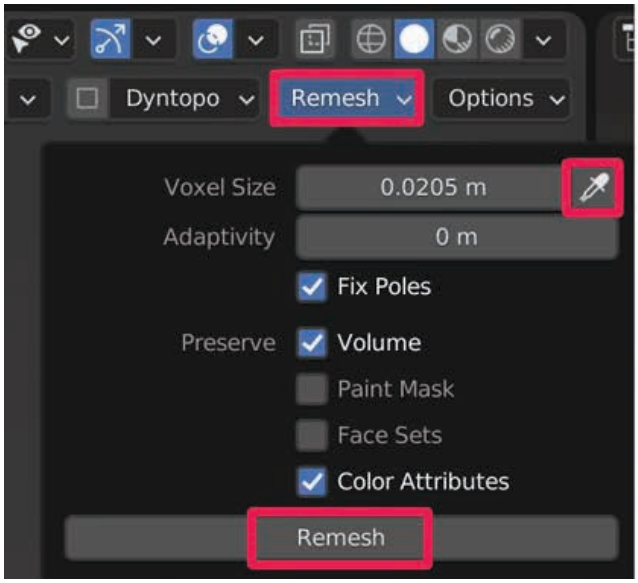
That's why we need to remesh it.

To remesh the resulting stretched geometry, first **press Shift+R** to get a preview of the voxel size. Then, having chosen an appropriate resolution, **press Ctrl+R** to do a Voxel remesh. Now you can sculpt on the stretched out parts with the detail you need.

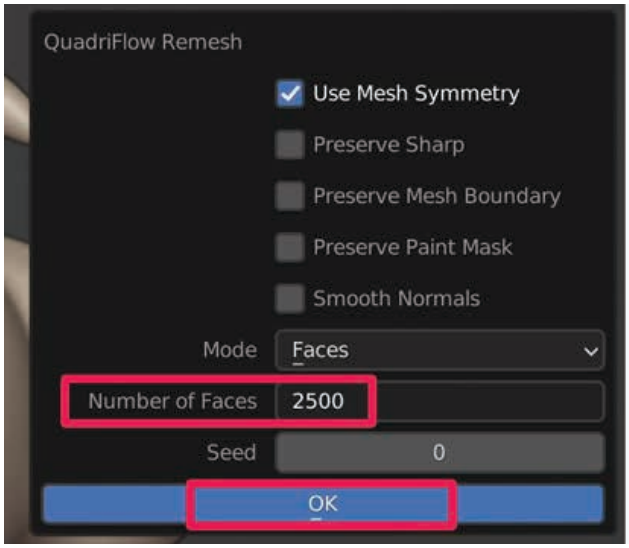


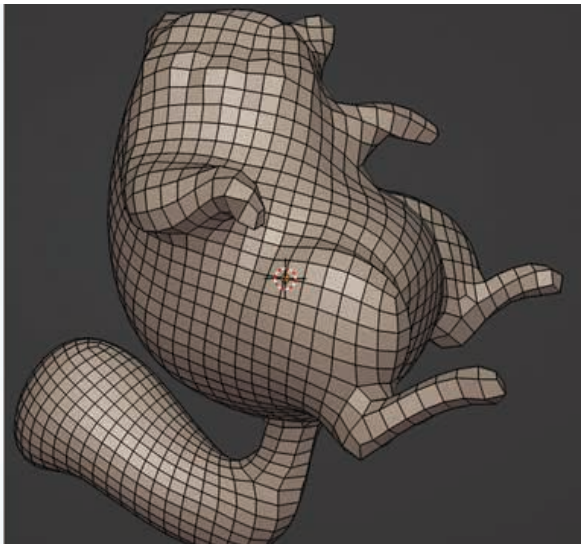


Instead of using the preview function, you can also use the **picker** tool to sample the resolution of a part of the mesh which isn't stretched. Then, you can **click on Remesh** in the same panel or **press Ctrl+R** to remesh at that specific voxel size.

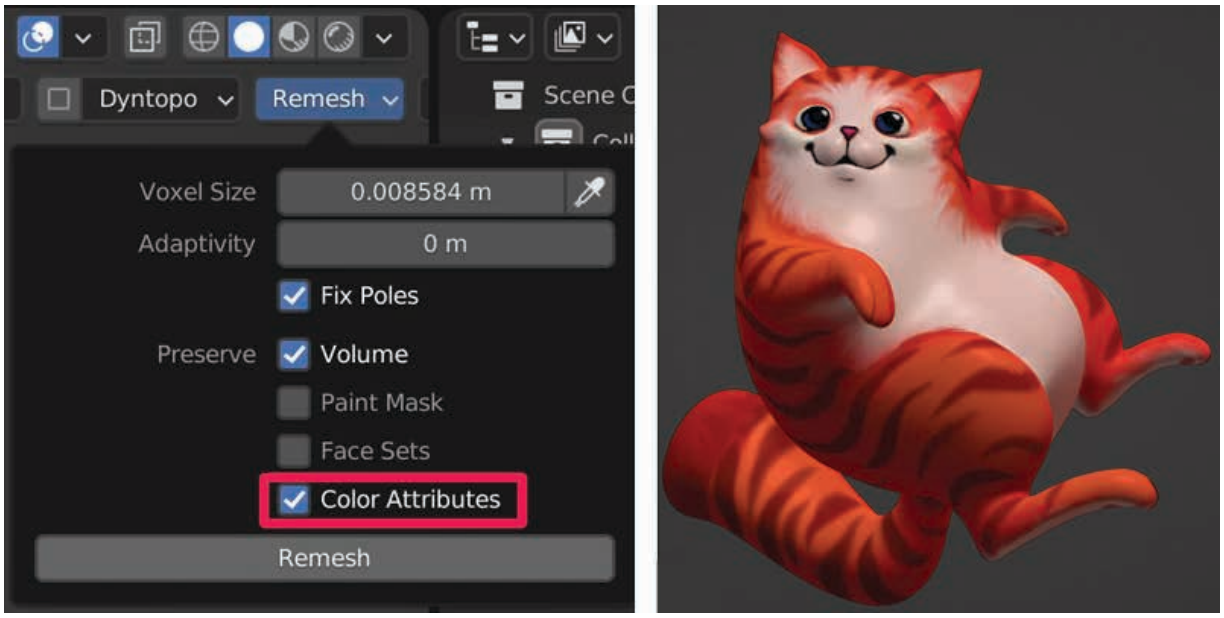


Alternatively, you can **press Ctrl+Alt+R** for **Quadriflow remesh**. This first presents you with some options, such as how many Faces you want the remeshed object to have and to preserve sharp edges.

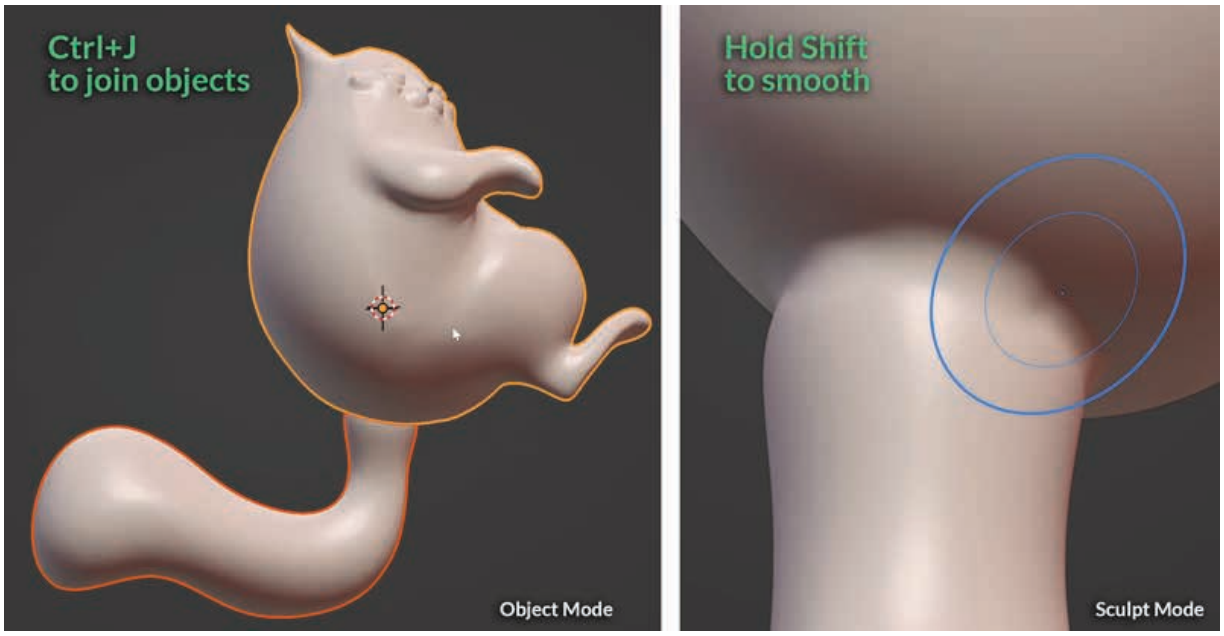




If you already have painted a **Mask, Face Sets or Vertex Colors** during sculpting, you can keep those even after remeshing by checking those options in the Remesh options prior to remeshing. Vertex Colors are named "**Color Attributes**" in the Remesh panel.

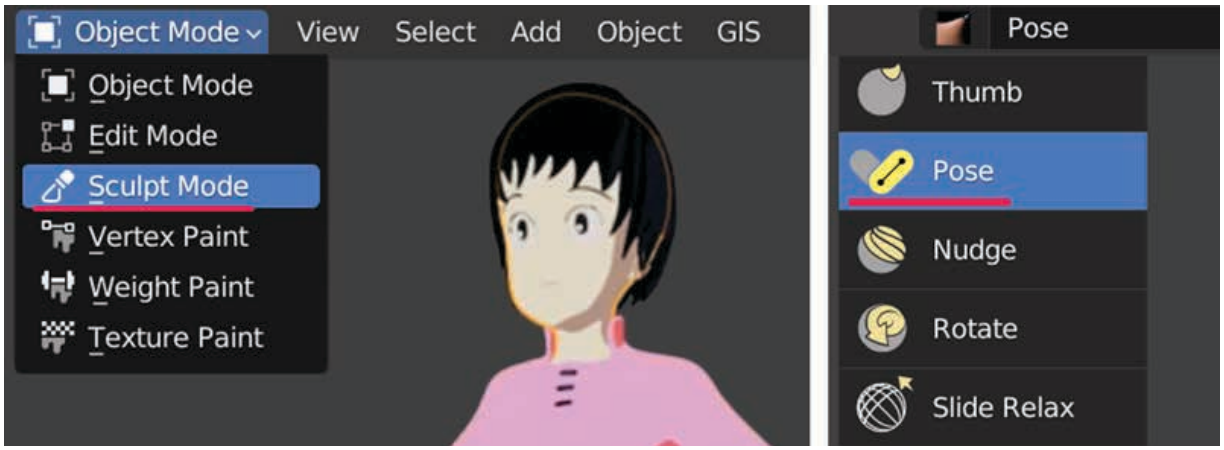


Remeshing can also be used to merge separate objects, after joining them with **Ctrl+J** in Object Mode. After remeshing, they will be merged and you can smooth out the connection by **holding Shift**.

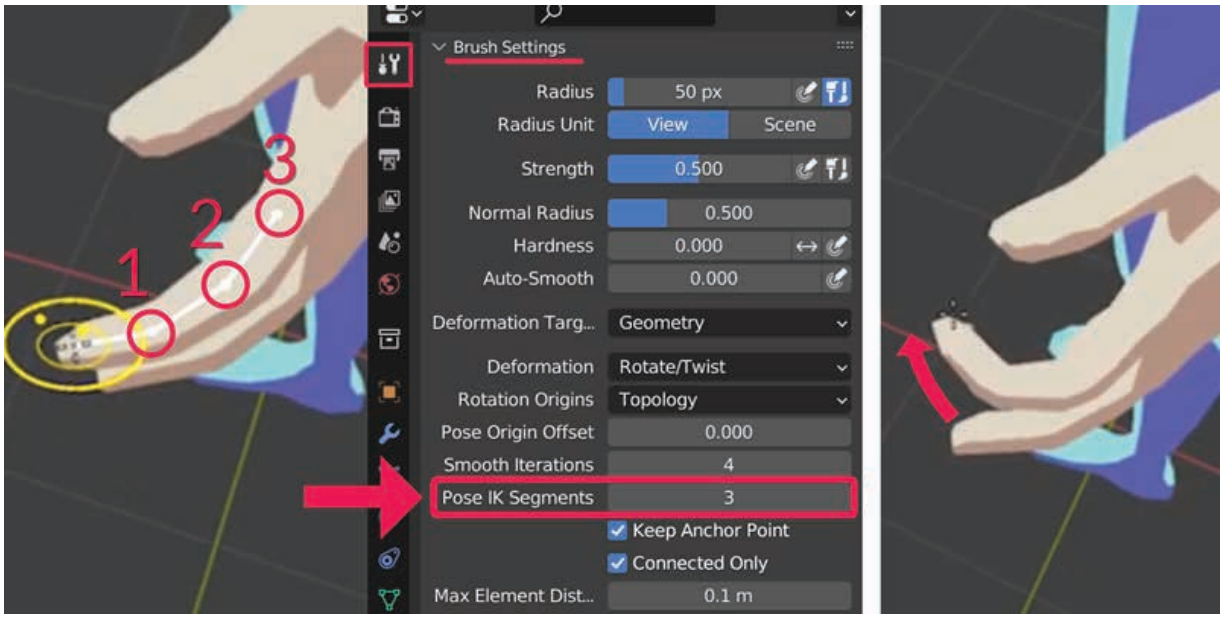


Pose Brush

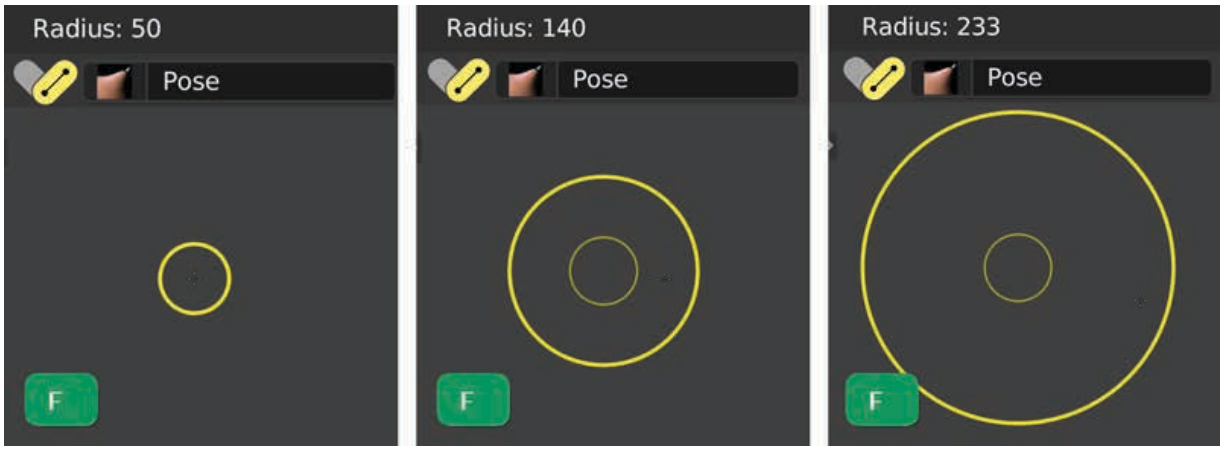
The **Pose Brush** is a handy tool for posing characters when you don't have time to set up an armature, or for when the mesh is too dense for that to work well. Select your character in Object mode and then go to Sculpt mode. Select the Pose Brush.



Depending on what you want to do, you can change the amount of **Pose IK Segments**. For example to pose a finger set the IK segments amount to 3.



Then **press F** to get the correct Radius of your brush.



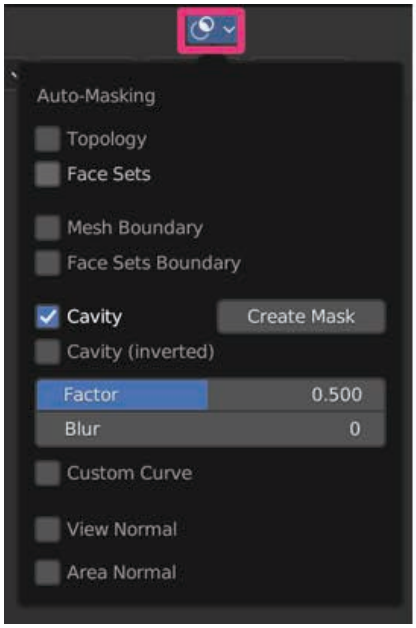
As you can see Blender tries to automatically create some kind of temporary IK bones for the part of the mesh that you move the mouse over. It helps to change the view to a logical angle, in order to get the best result.

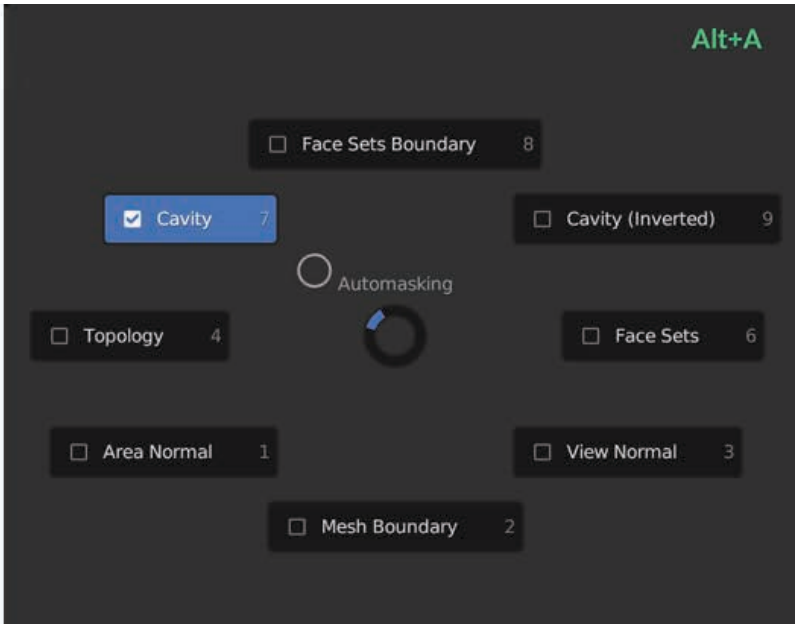
It's not perfect, but if you quickly want to pose a character without having to set up an armature, it works!

Sculpting Auto Masking with Cavity

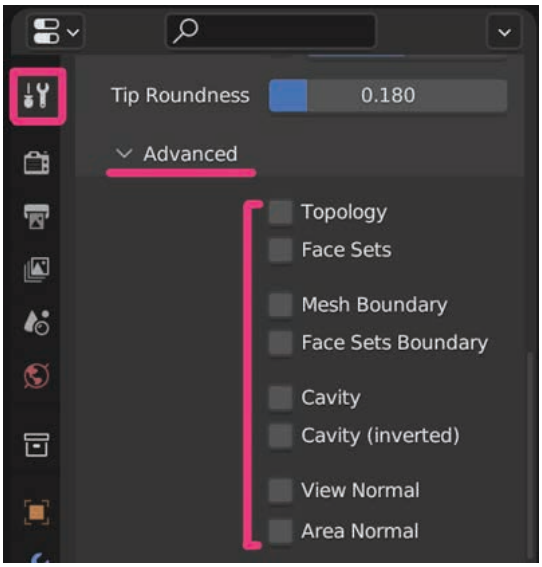
Blender 3.4 introduced an interesting **Auto Masking** feature to the sculpting tool set. This can be useful both for painting in Sculpt Mode and for quickly masking valleys and peaks (called “cavity” or “cavity inverted” in Blender) in your sculpt.

To use it, make sure you’re using Blender 3.4 or later. You can find it from a drop-down menu in **Sculpt Mode**, as well as via the keyboard shortcut **Alt+A** that brings up the auto masking pie menu.

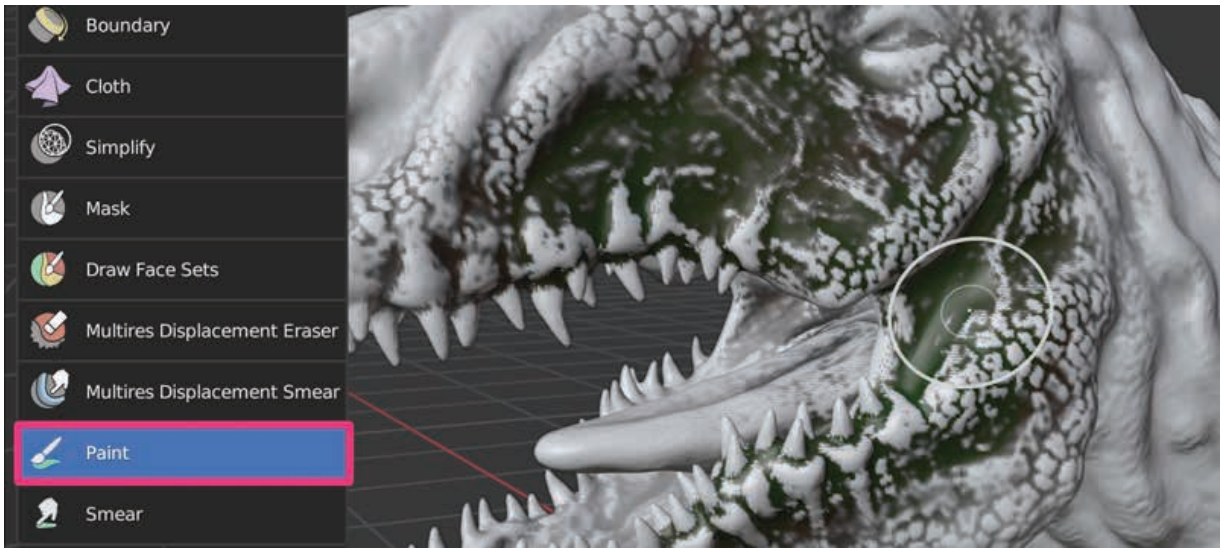




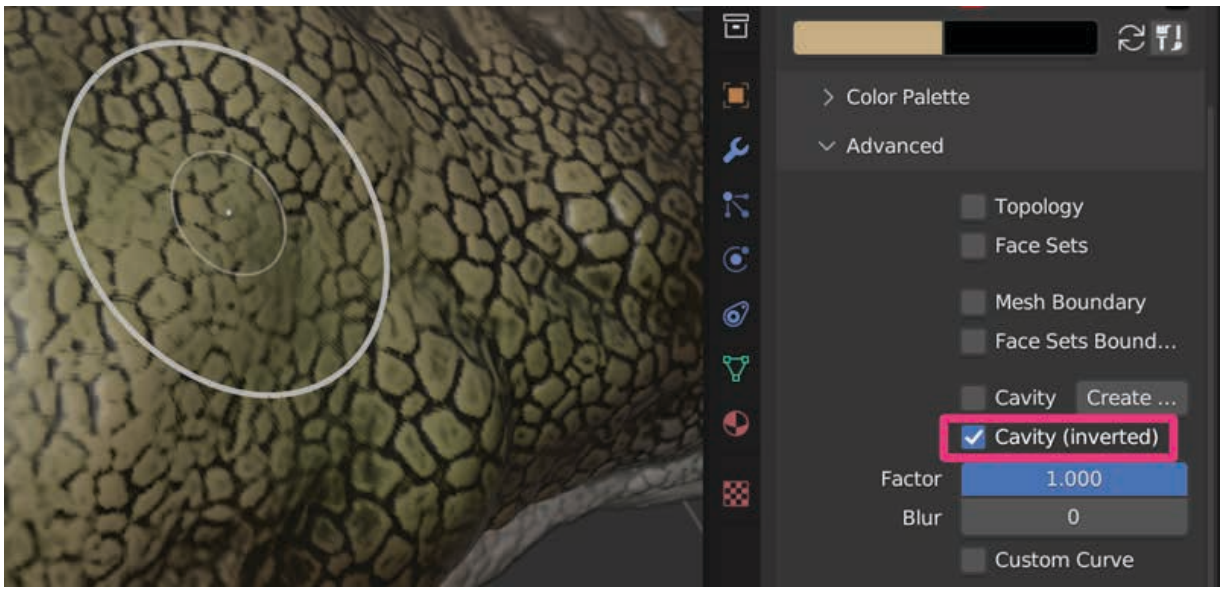
You can override this general setting in the individual Sculpt Brush Settings under **Advanced > Auto Masking**.



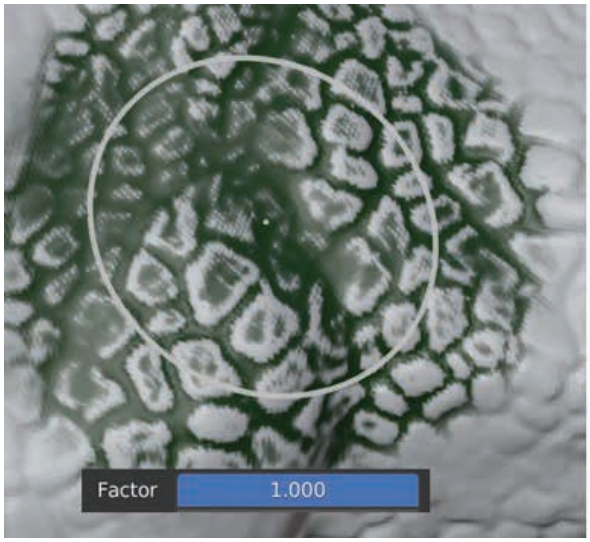
The auto masking feature is useful in combination with the new Sculpt **Paint** tool. For example to paint darker shades in skin creases or pores.

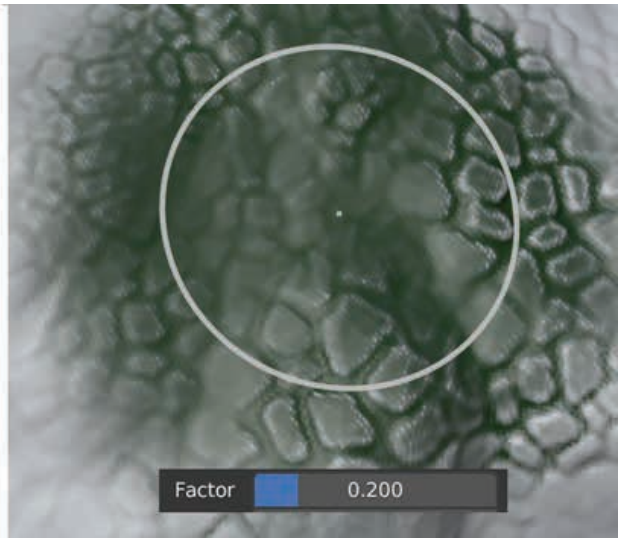


You can invert the effect by checking **Cavity (inverted)**.

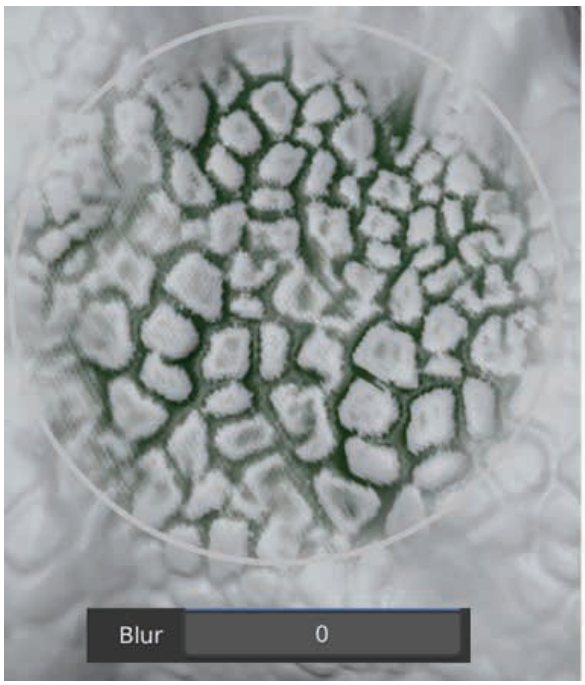


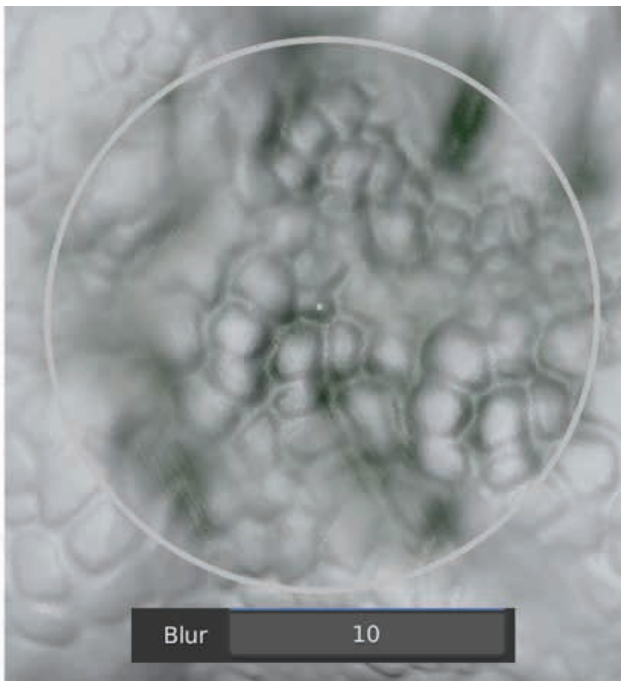
Factor controls how strong the auto masking is.





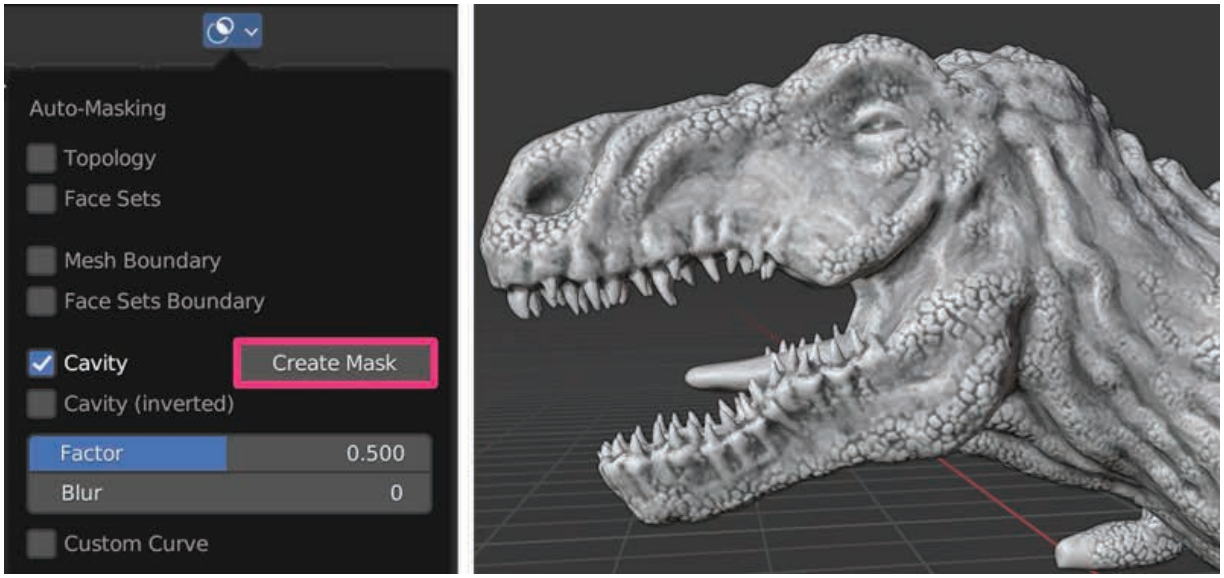
And **Blur** controls how sharp or blurry the mask is.



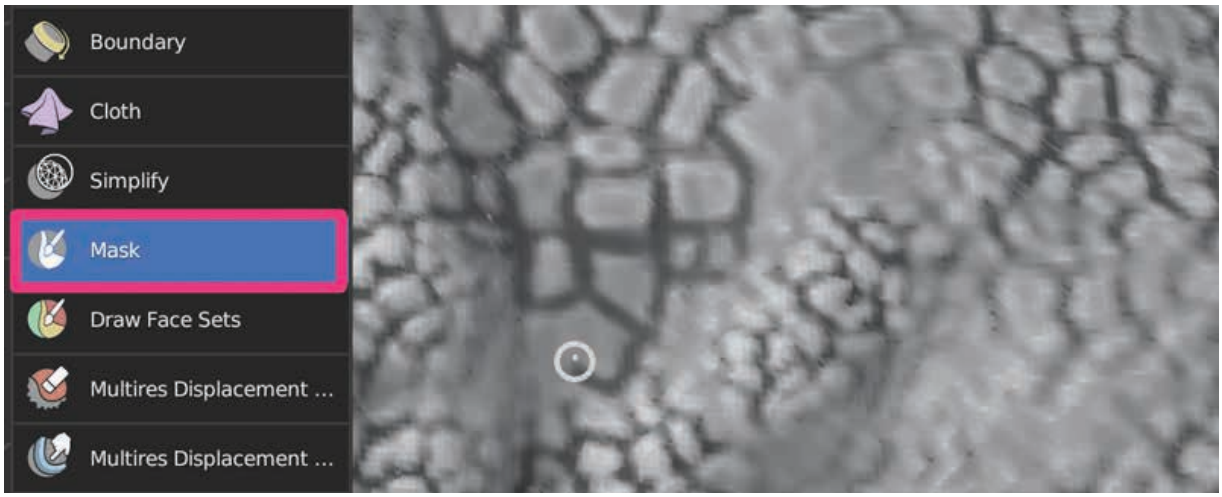


Auto Masking is also useful when you only want to sculpt only in certain areas. For example to add volume only to the non-cavity areas. In this example, I used **Cavity (inverted)** with an **Inflate brush** to make the scales of this dinosaur sculpt more pronounced.

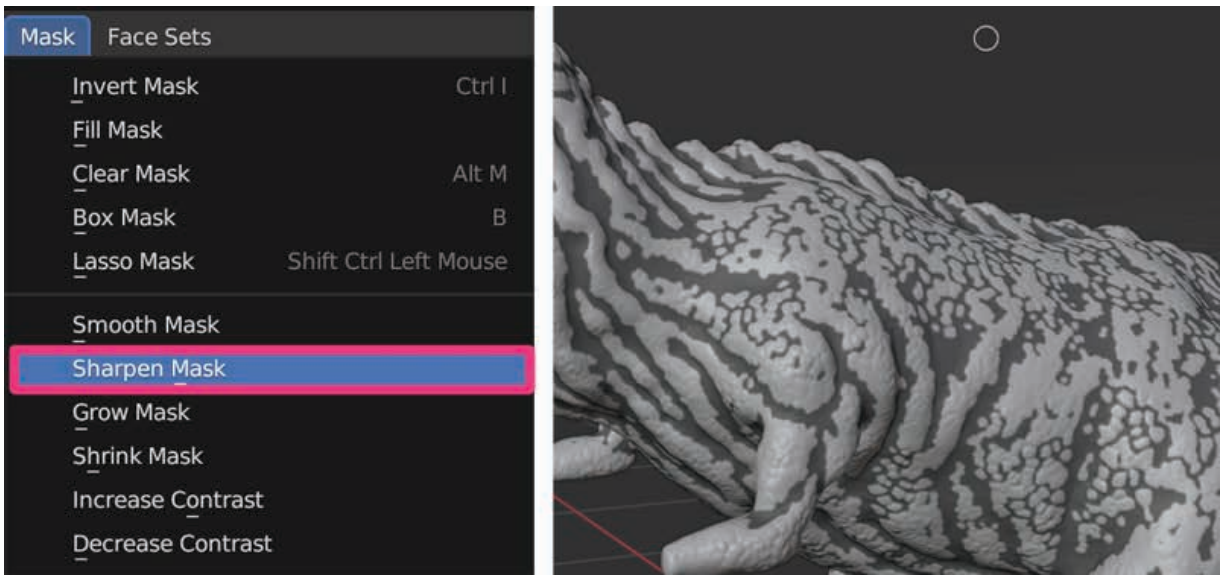
Another interesting use of the auto masking feature, is that when you check the "Cavity" option, a new option appears called **Create Mask**. When you click that you get a mask based on cavities.



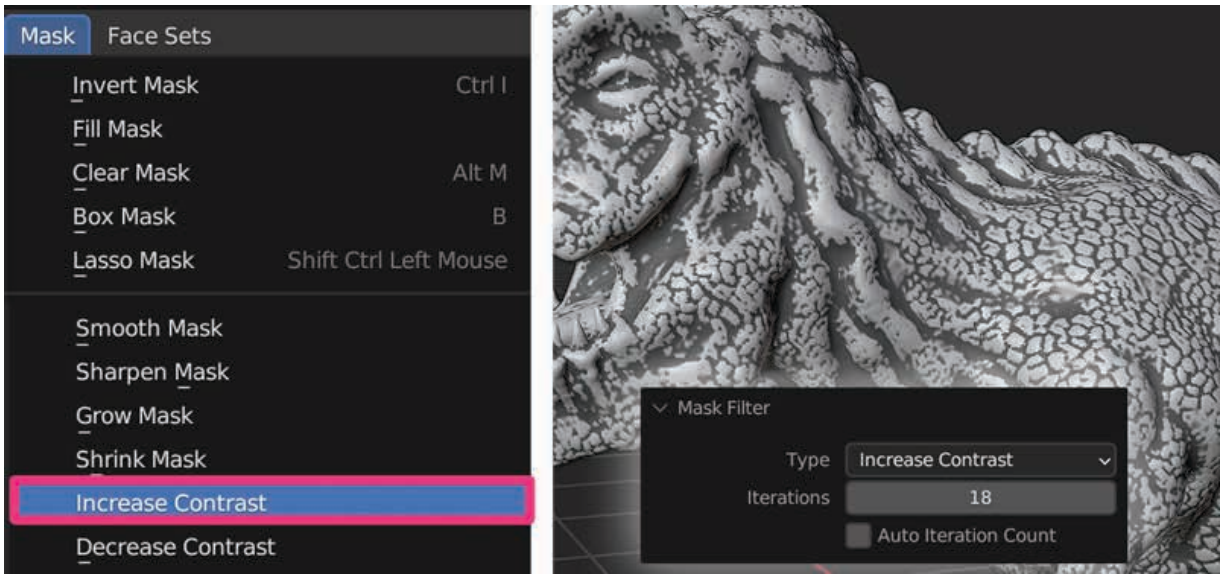
You can further edit this with the Mask tool if needed.



Use the **Sharpen Mask** or Smooth Mask options to get some different interesting results based on this mask.



Or Increase the contrast of the mask to further adjust it as needed.

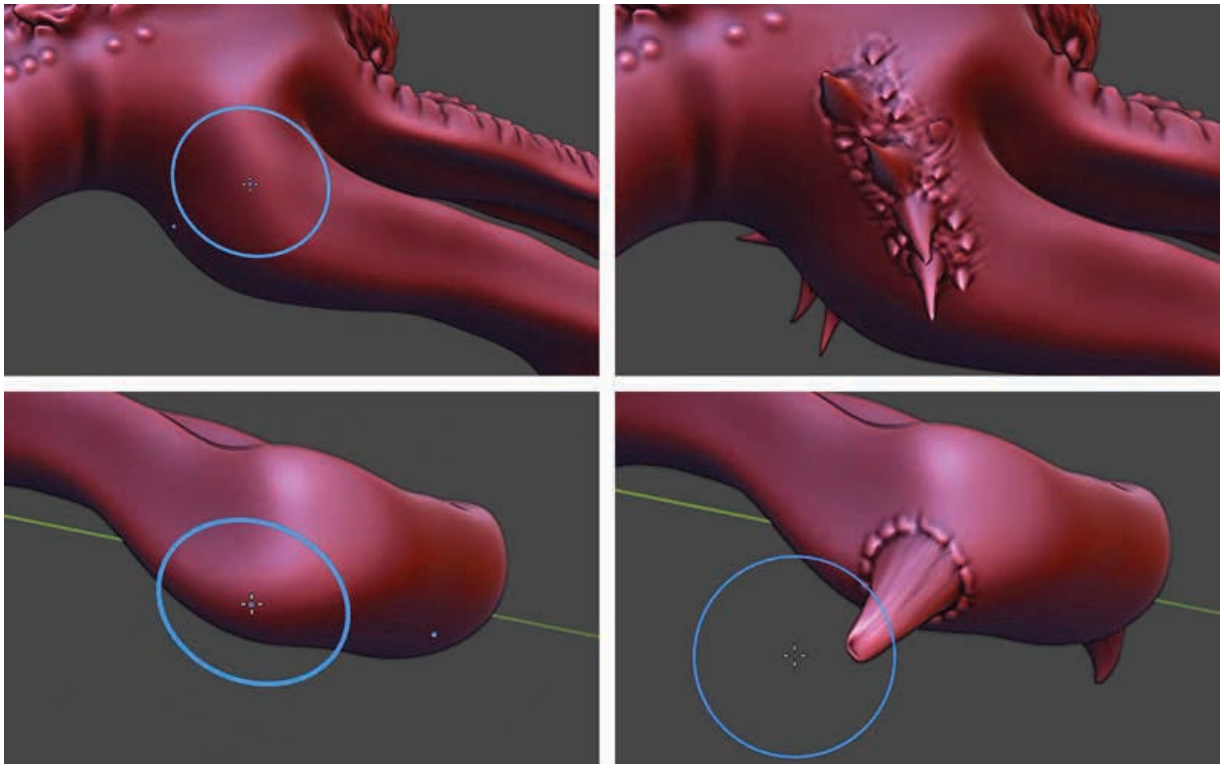


It can be an interesting method to bring out detail in 3D scans as well.

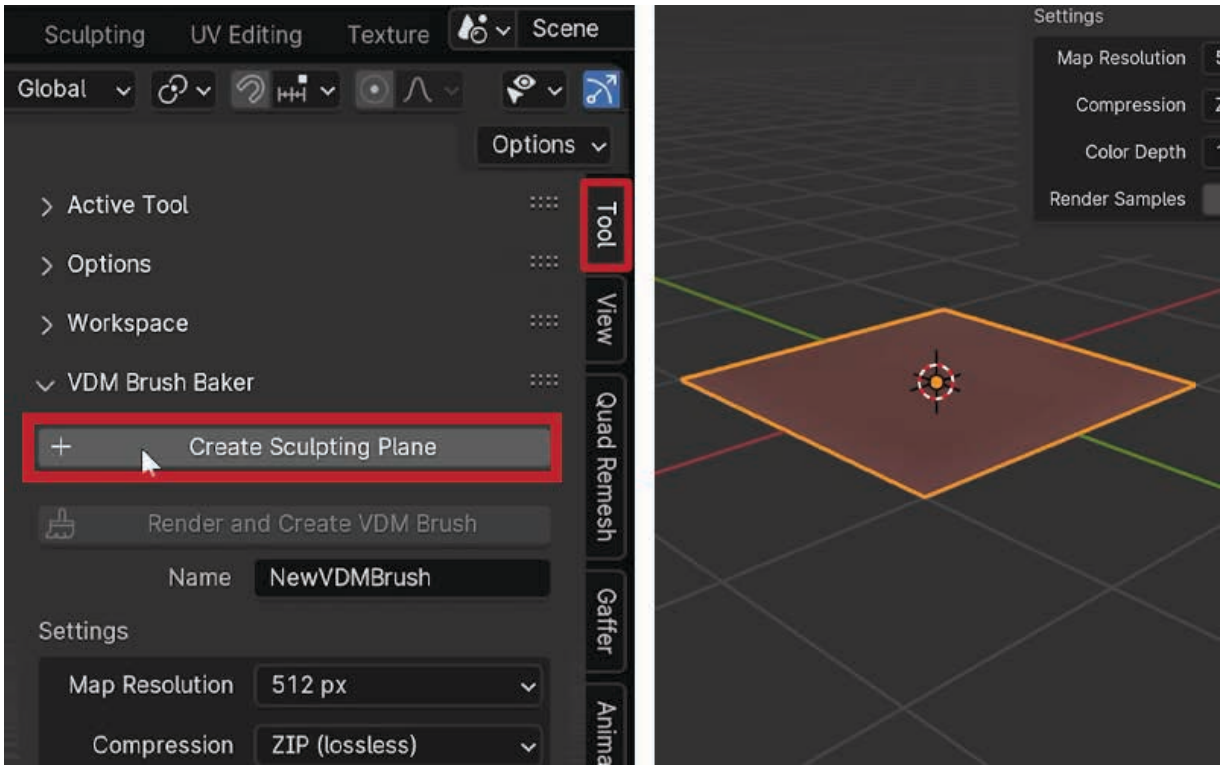
How to create VDM Brushes in Blender 4

A while ago (around Blender 3.6) VDM brushes were introduced in Blender. At that time, you had to install an add-on, and append a special material for baking. But since Blender 4, it's all become very easy. You don't need any more add-ons.

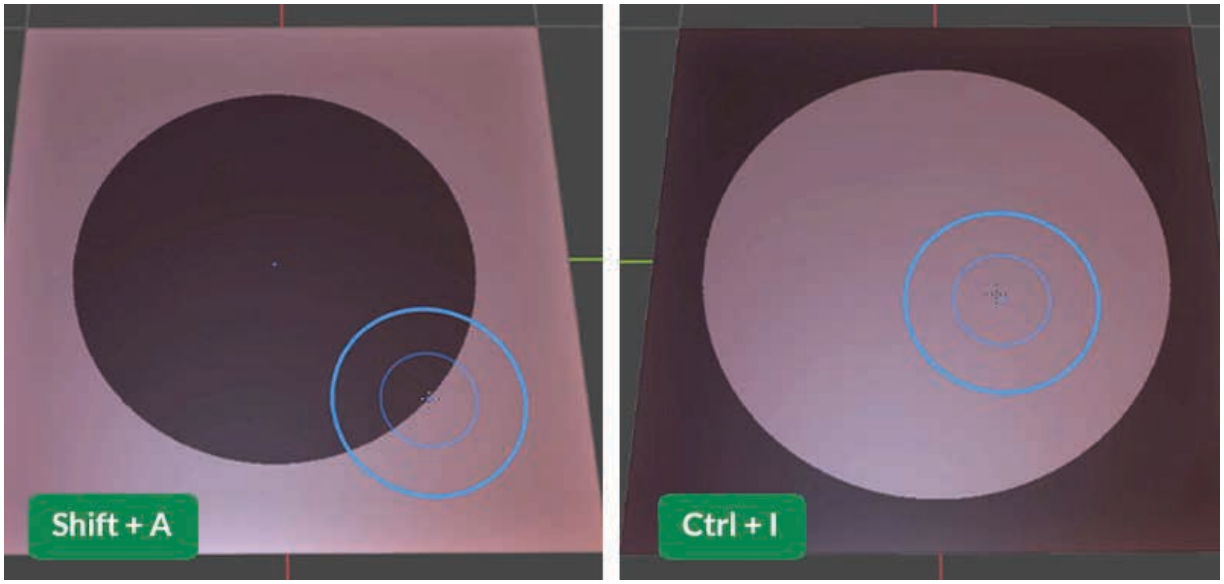
As an example, using VDM brushes by Nicolas Swijngedau it's easy to add some dragon details.



Click on **Create Sculpting Plane** in the VDM Brush Baker option panel. This adds a subdivided Plane with a Multires modifier.

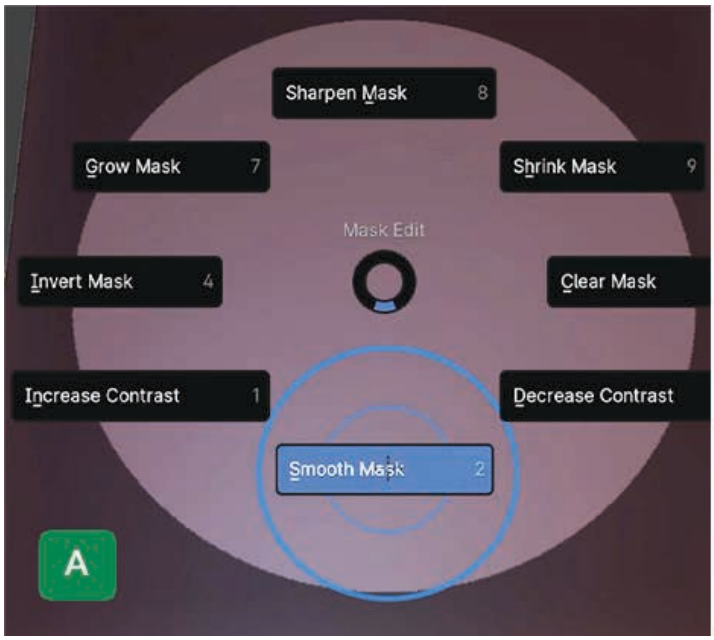


With **Shift+A** you can add a circular mask. You can move it by holding the Spacebar, if necessary. Then invert it with **Ctrl+i**.



Then smooth the mask a few times. You can do this by **pressing A** and moving the cursor down, to select "**Smooth Mask**".

Then **press Shift+R** a few times to repeat the smoothing. This makes sure you don't affect the edges of the Plane during sculpting.

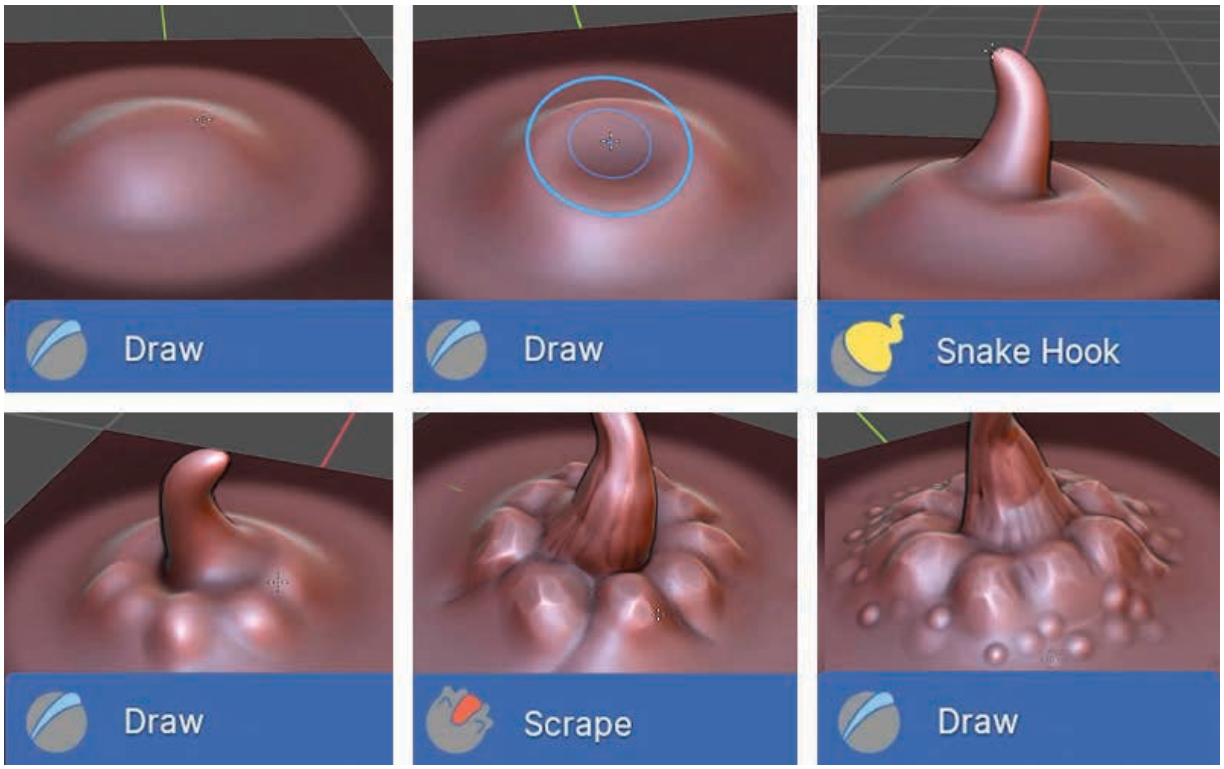


Now sculpt something on the plane.

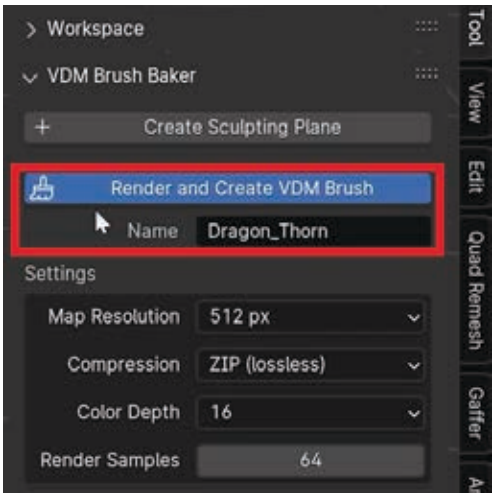
As you can see, I mostly used the **Draw brush** for this dragon part (whatever it is!).

I only used the **Snake Hook** brush to pull out the claw-like part.

Then I also used the **Scrape brush** to make it look a bit more interesting, and to avoid everything being too smooth.



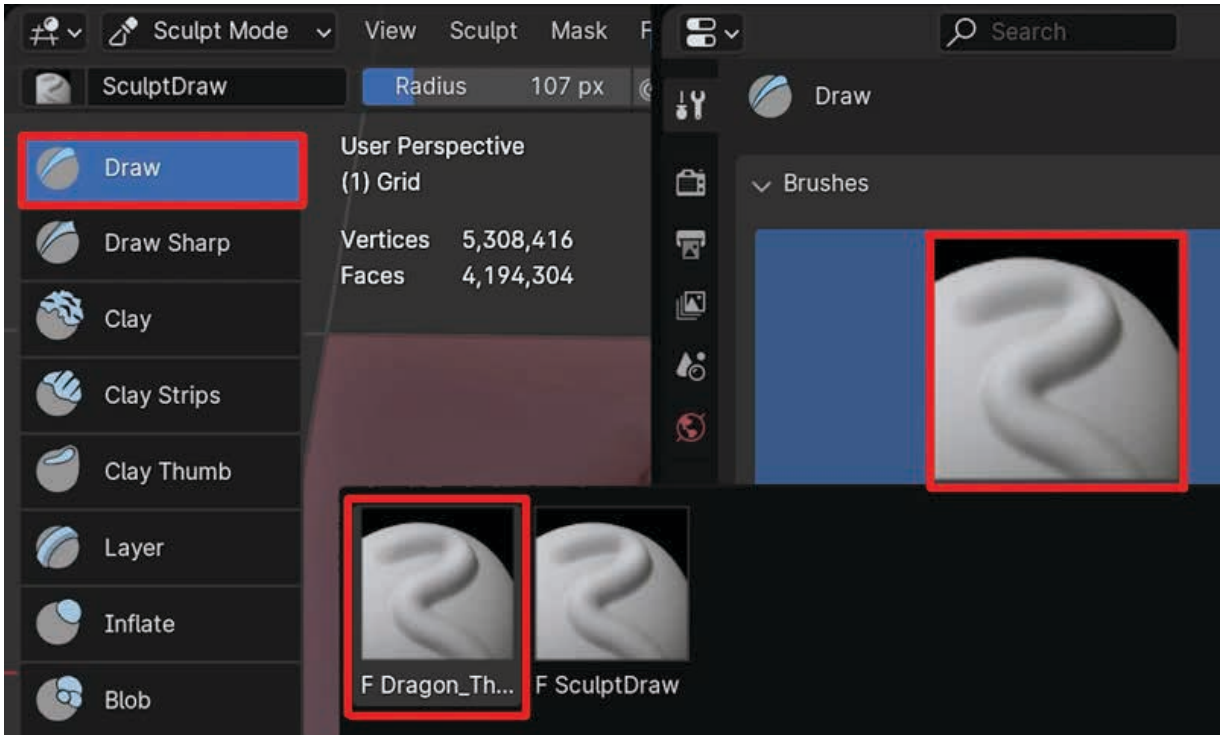
Give your brush a name, and click on **Render and Create VDM Brush**.



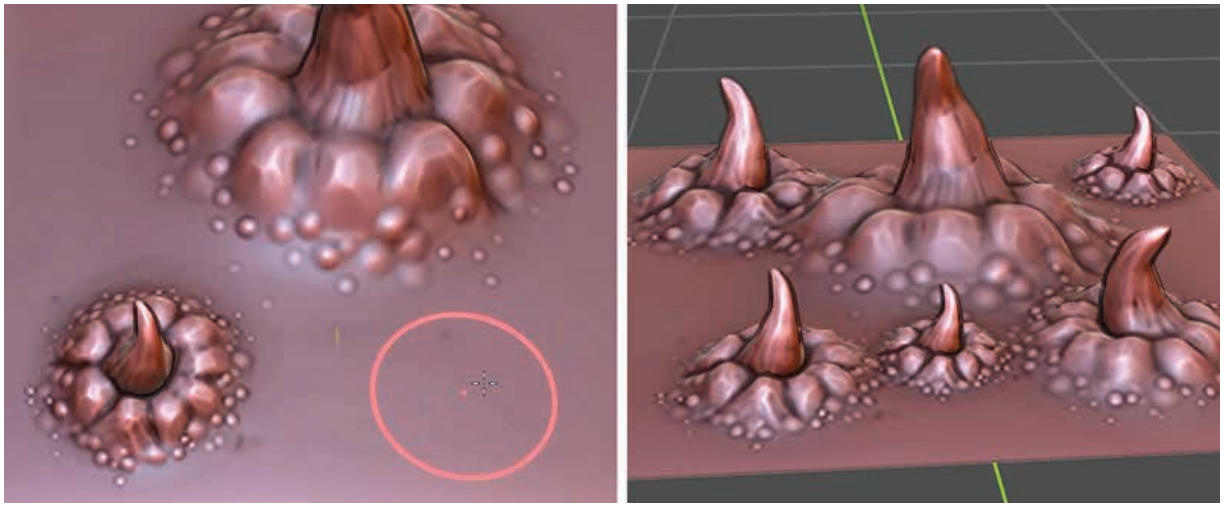
Don't worry about the settings like the resolution. VDM's really need only 512 pixels width and height.

That's one of the reasons they're so small and lightweight.

Now you will find your new brush under the **Draw** Brush icon.

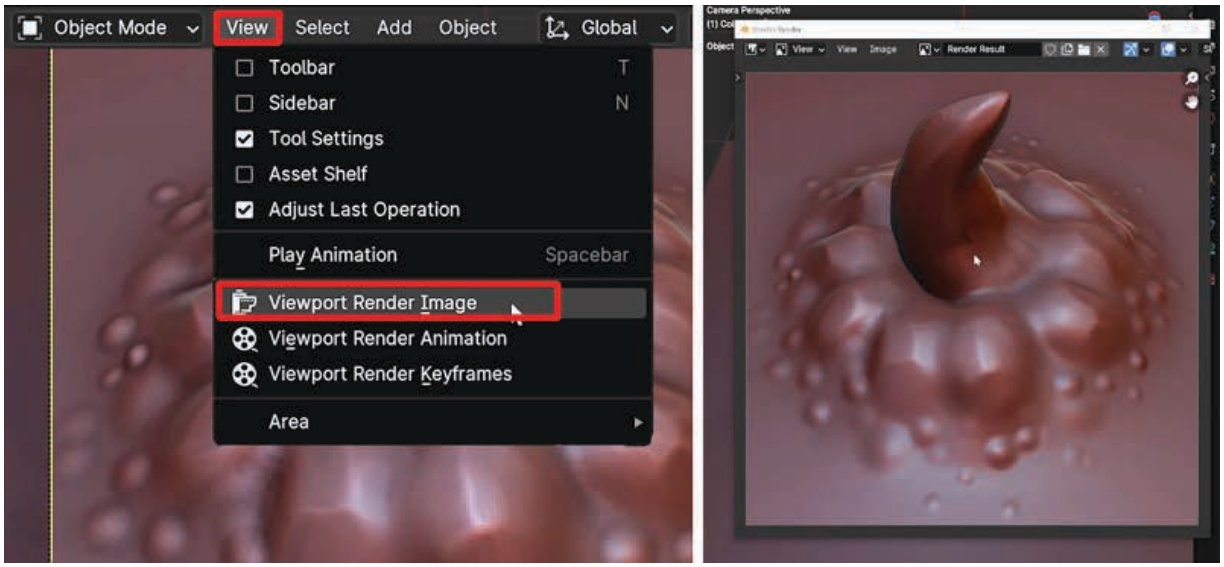


You can test right away to see if it works.



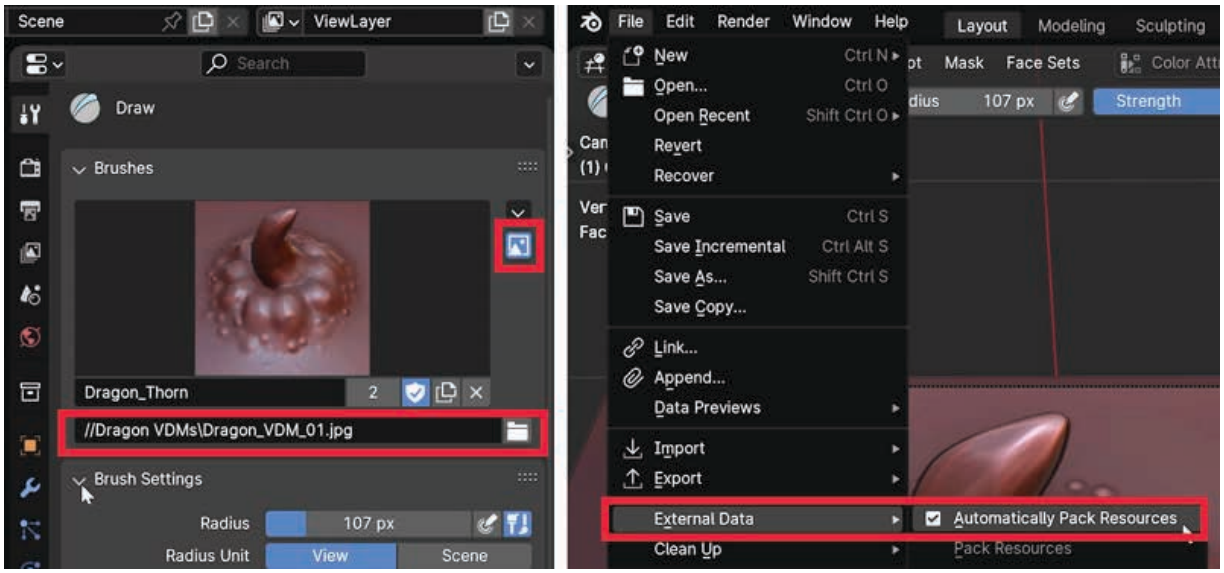
Click here to return to the index.

If you want, you can render an image of your brush, and add it as a custom thumbnail. To do this, you'll need to set up a Camera. I like to just do a **Viewport render**. Save the resulting render as a small JPEG image.

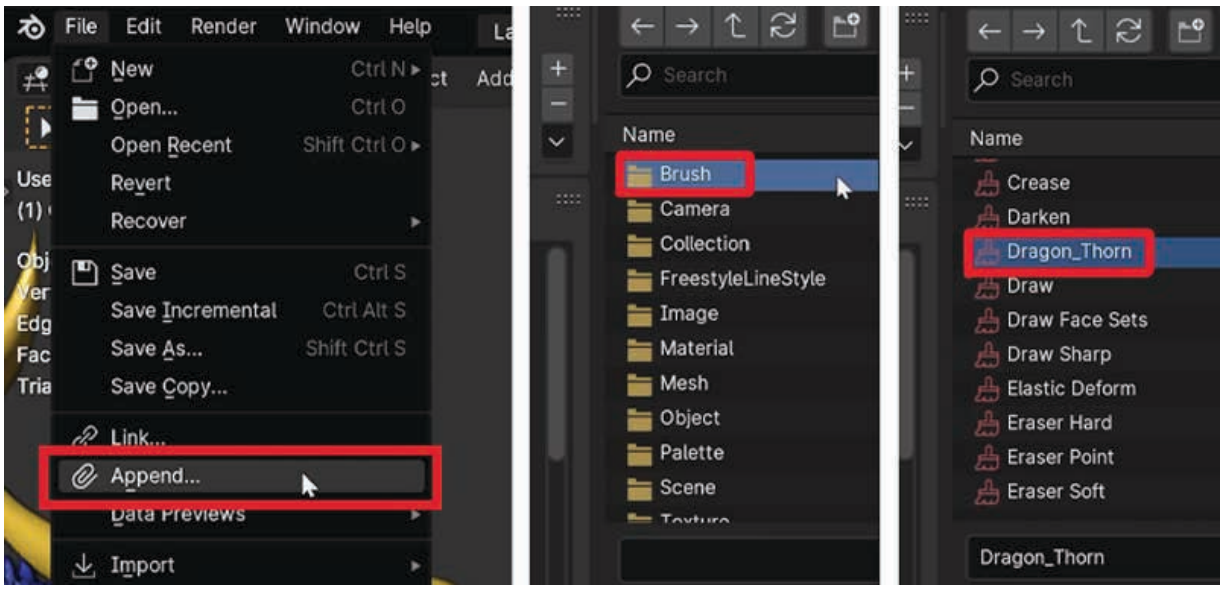


I recommend **JPEG** because I tried with a PNG file, and it didn't seem to display the thumbnail. Even with JPEG, it takes a few seconds before the thumbnail is displayed the first time.

Enable **Automatically Pack Resources** to make sure your VDM texture is saved.



Then save the Blend file somewhere where you can easily find it. Now any time you want to use this brush, you can **append** it from the saved Blend file.



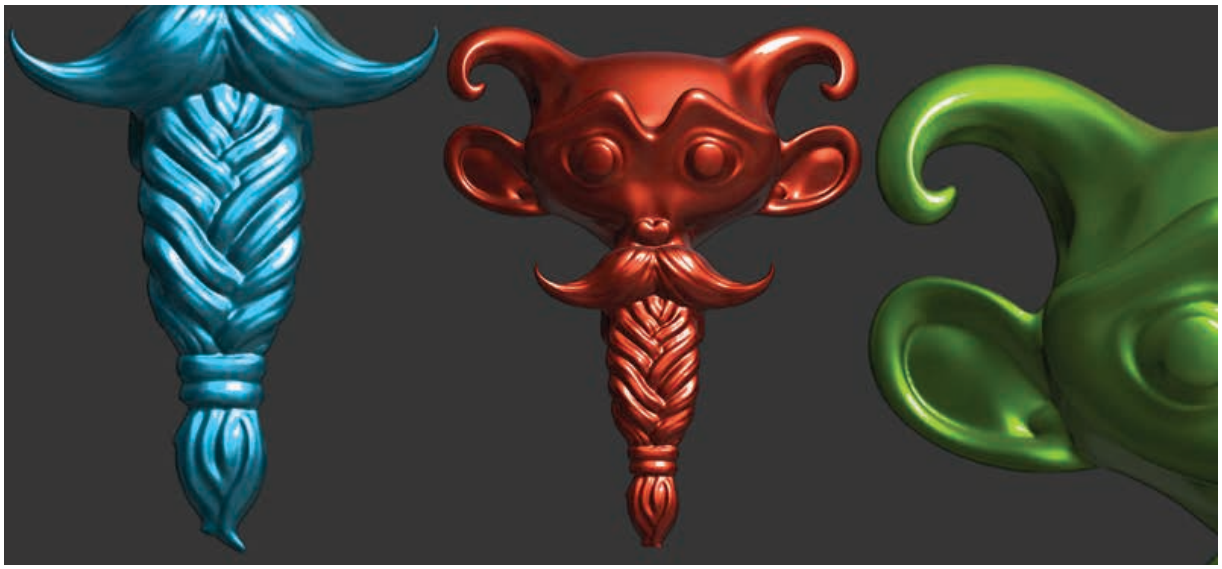
After appending the brush, you will again find it under the Draw Brush icon.

By the way, you can even create VDM brushes from existing models. The process for this is a bit more complex but it's described in detail in my video course. So check that out if you want to learn more about sculpting with VDM's.

Sculpting with Vector Displacement Maps - the VDM brush

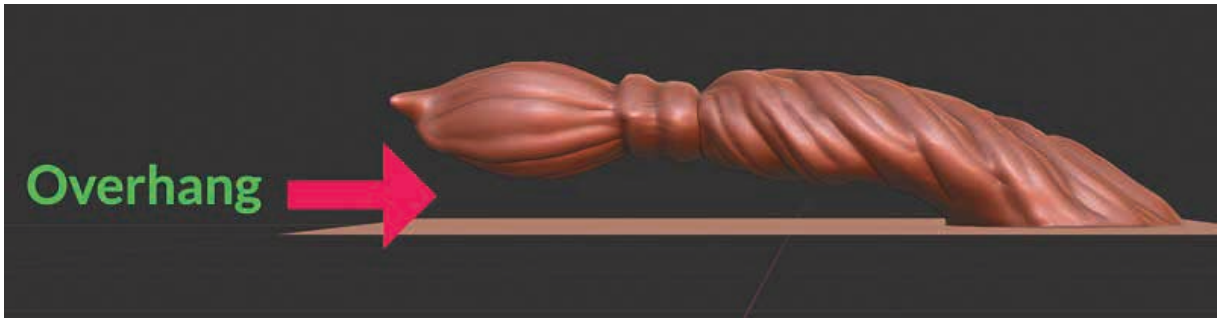
Please note: While everything described here still works, it's now easier thanks to the built in VDM functionality of Blender 4. So read "**How to create VDM Brushes in Blender 4**" for a more up-to-date workflow.

Vector Displacement brushes are great if you want to easily add complex shapes such as for example stylized facial hair or horns, in Sculpt Mode. This feature has long been available in sculpting apps like Zbrush, and has now also been added to Blender since version 3.5.



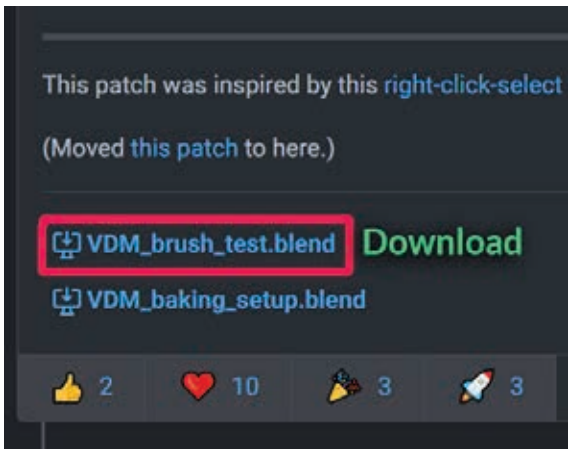
Beard VDM's by Nicolas Swijngedau on Artstation.

Contrary to displacement maps, which displace only in the Normal direction, Vector Displacement Maps (VDM) allow for overhang.



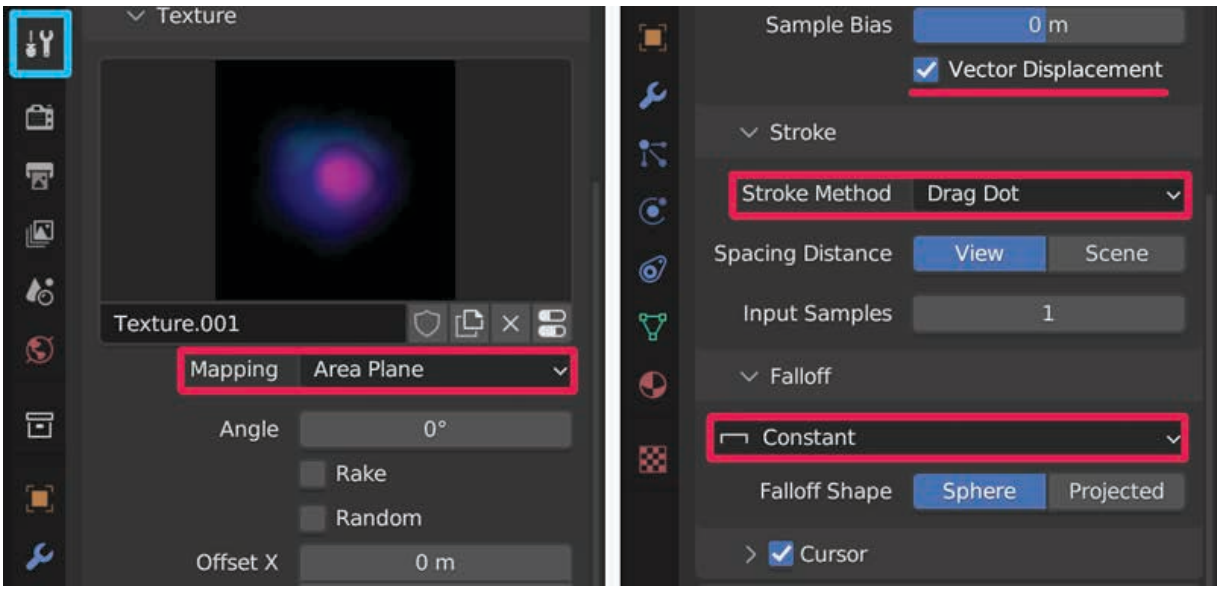
On the Developer page you can find a demo file with a VDM brush to test.

As well as a file for baking your own sculpts to VDMs (which we will look at in the next topic).

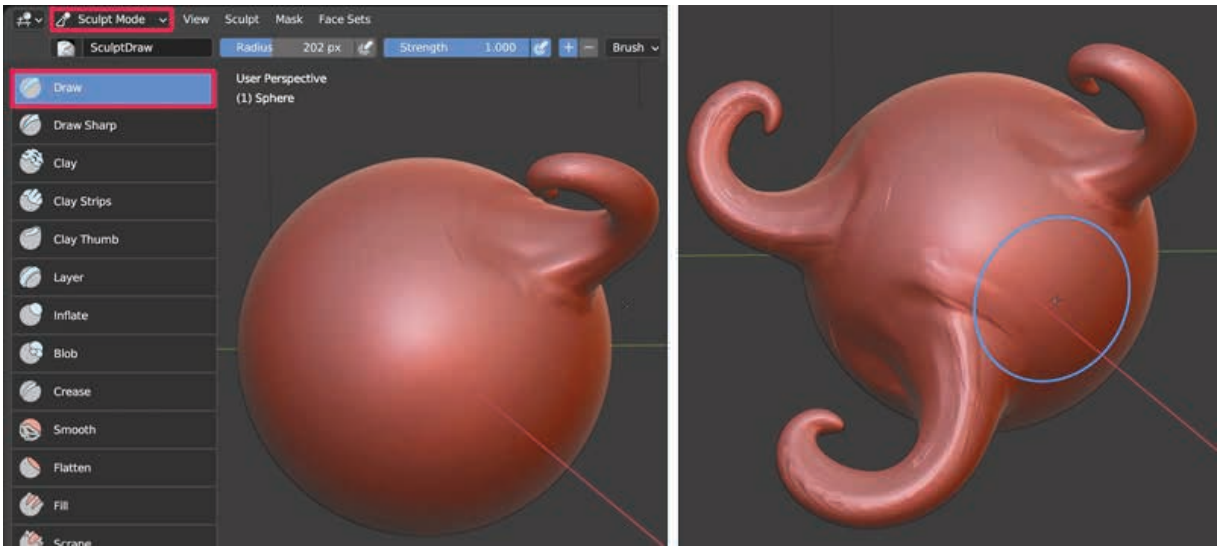


To be able to use the VDM brush correctly, in the Brush properties, under Texture, Mapping needs to be set to **Area Plane**.

And under Stroke, the Stroke Method needs to be set to **Drag Dot**, with Falloff set to **Constant**. The Vector Displacement option is available as an option in the Brush properties under Texture and is enabled by default.



To use the VDM brush, just click and drag on the surface of the object.



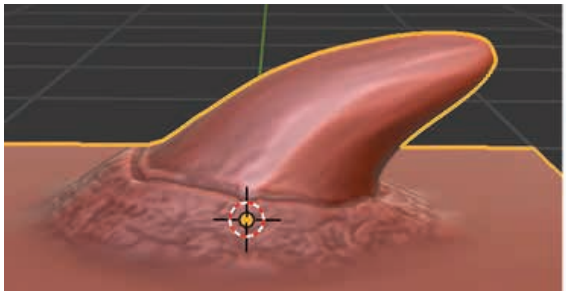
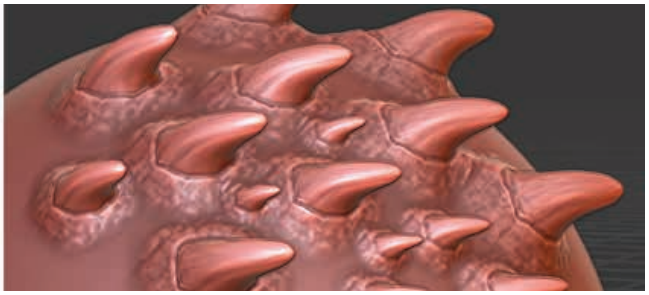
Get the demo files here:
Developer Page

Read more about it here:
Blender.Chat Sculpting Channel
BlenderArtists.org Big Sculpting Thread Part 2
RightClickSelect proposal

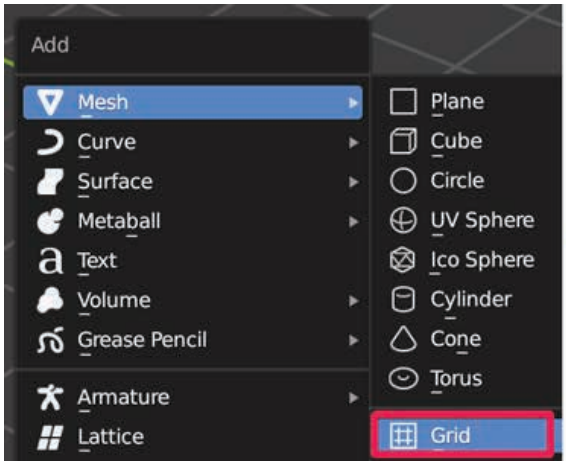
Baking Vector Displacement Maps

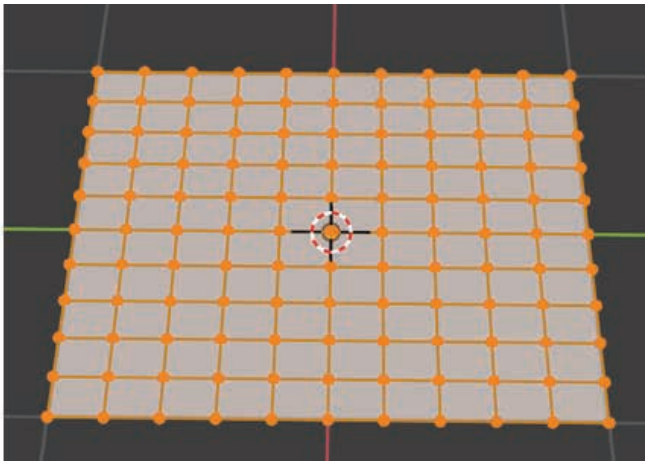
Please note: While everything described here still works, it's now easier thanks to the built in VDM functionality of Blender 4. So read "**How to create VDM Brushes in Blender 4**" for a more up-to-date workflow.

Previously, we had a look at the new Vector Displacement (VDM) Brush functionality in Blender 3.5. Now, we'll explore creating our own VDM brushes.

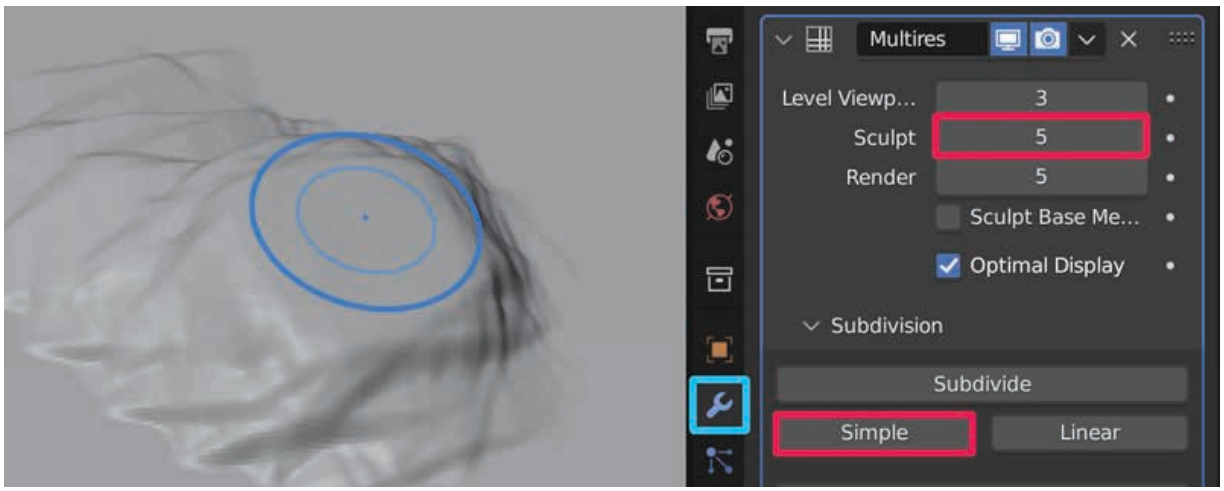



Add a **Grid**, which is a Plane that already has some subdivisions. It's essential to start with a Plane or Grid as they are UV unwrapped by default. You can't UV unwrap a more complex shape after sculpting and bake that to a VDM. It has to be unwrapped from the start.

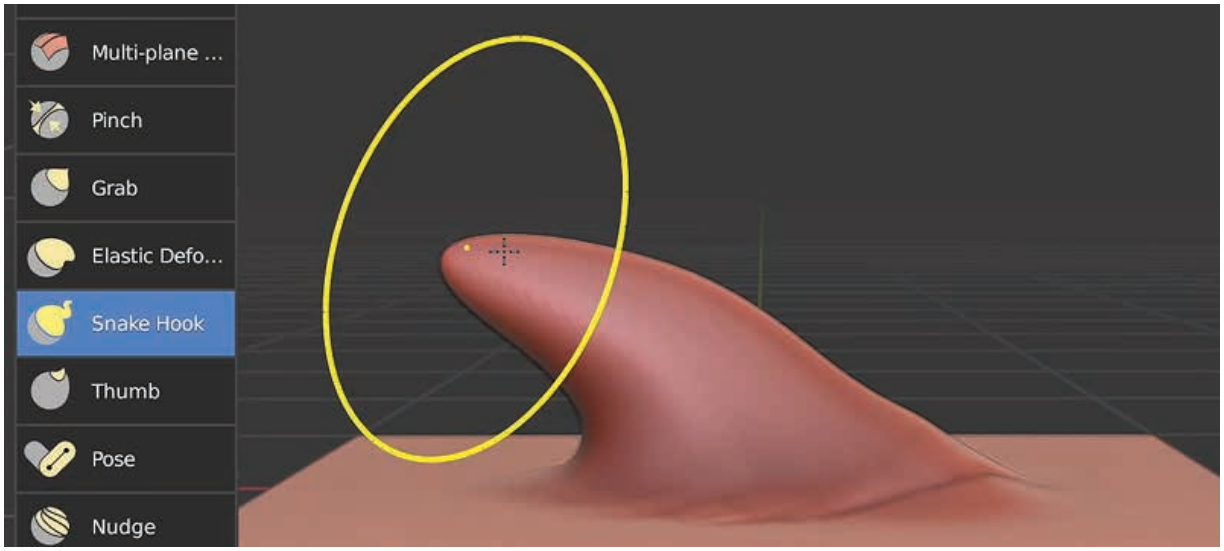




Give it a **Multires modifier** and click on **Simple** a few times to increase the resolution. You can test in Sculpt Mode how many levels you need for a smooth result, in my experience 5 levels is necessary for a nice smooth sculpt.

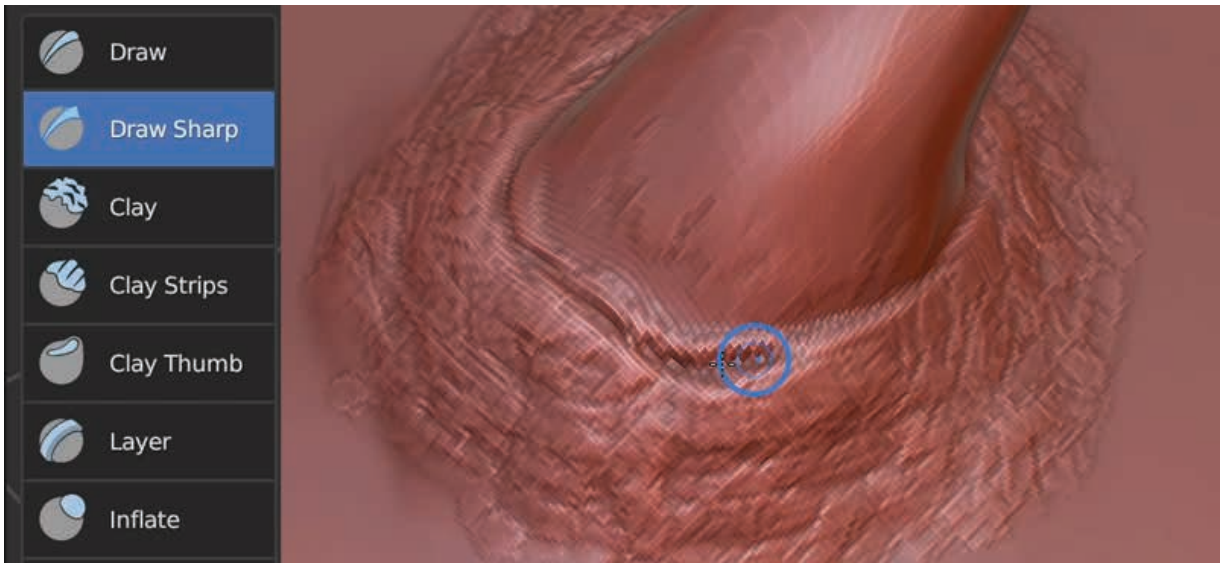


In Sculpt Mode, sculpt some displacement. Be careful not to displace the edges of the Plane. Here I'm using the **Snake Hook** brush to pull out a kind of horn shape.

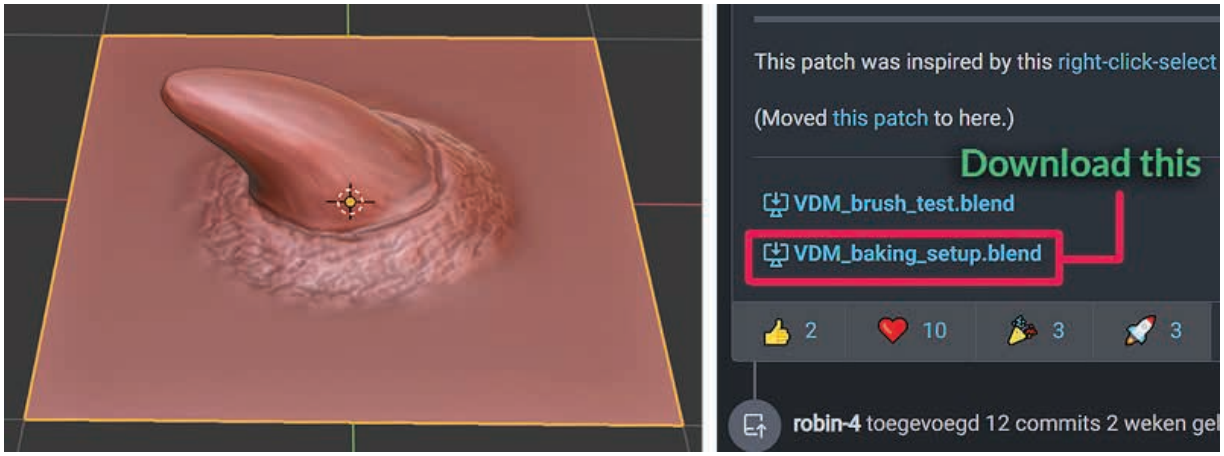


The **Clay Strips** brush is useful for adding some organic volume.

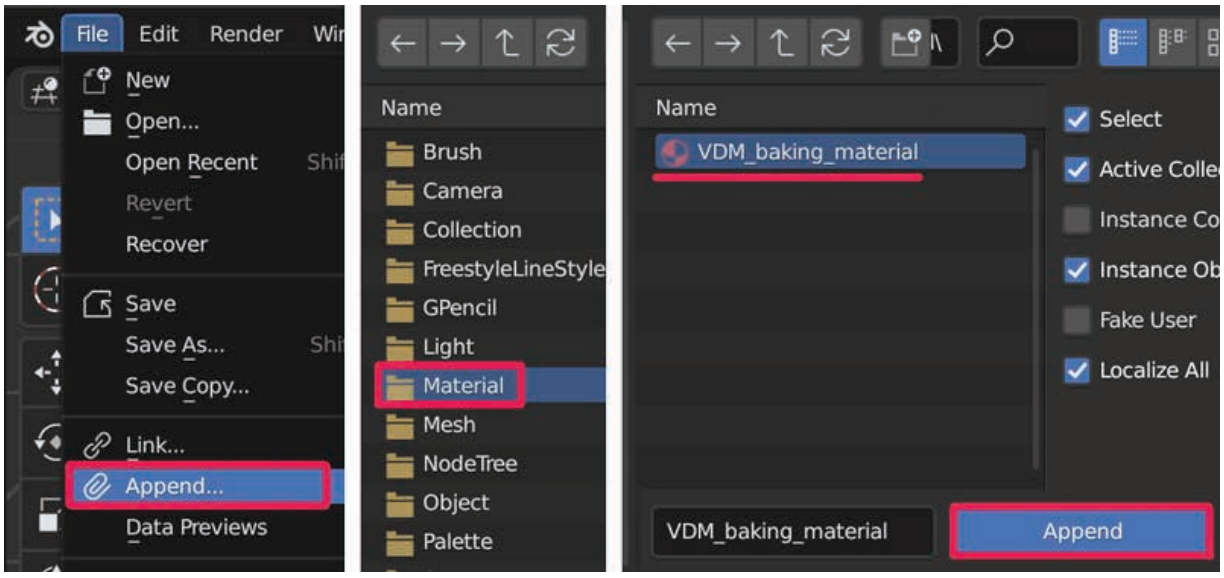
Draw Sharp can be used to create some grooves.



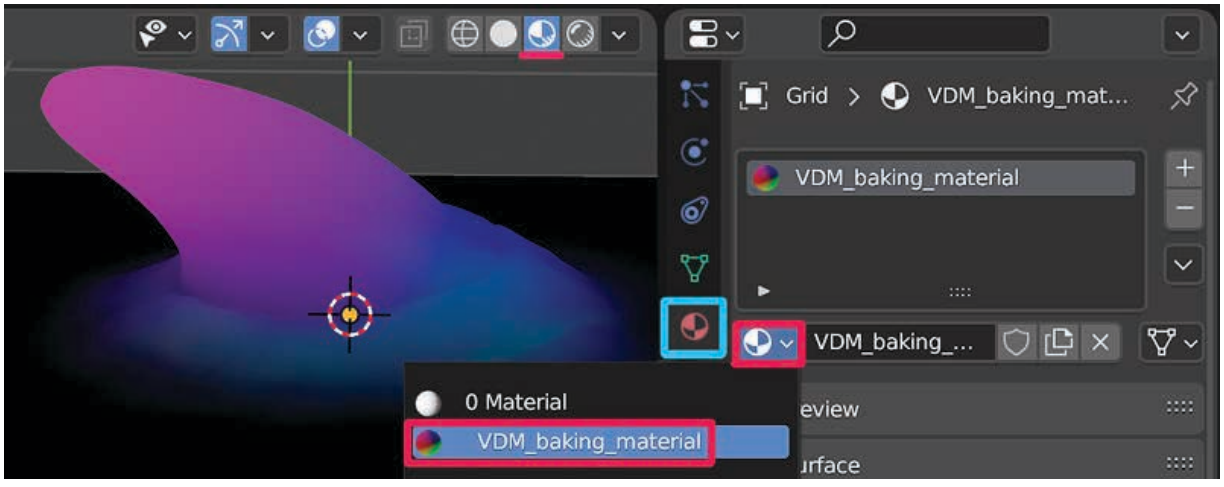
With our sculpting work done, we can bake it. But since Blender doesn't have VDM baking as an option in the Bake panel yet, we have to download the VDM baking setup file from the developer page.



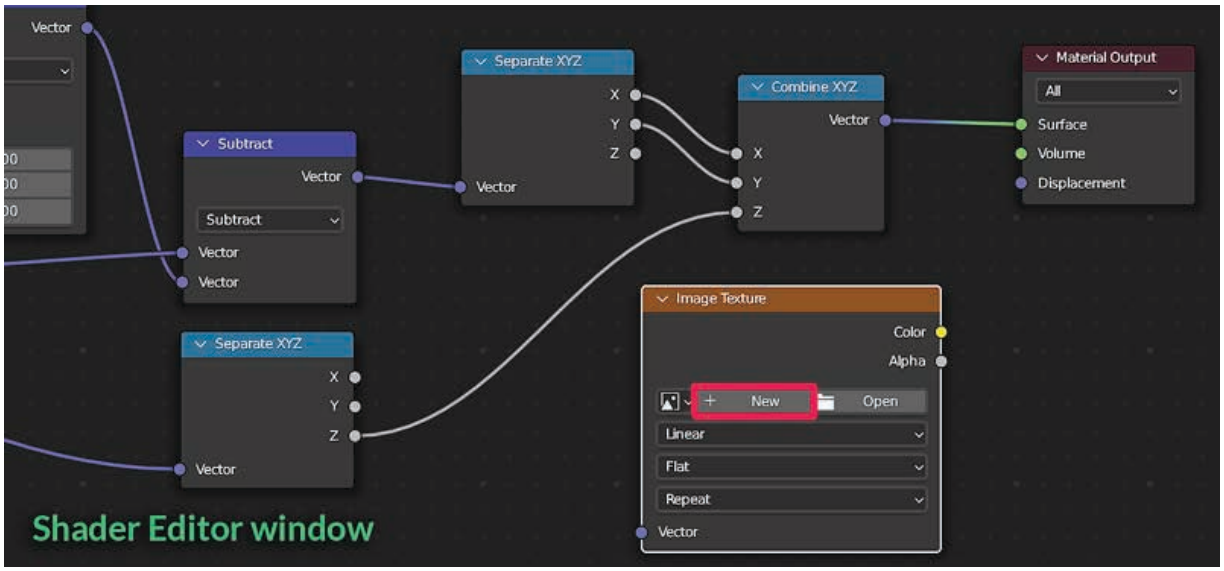
Append the baking material you downloaded: Go to File > Append > VDM_baking_setup.blend and navigate inside the blend file to the Materials, where you can find "**VMD_baking_material**".



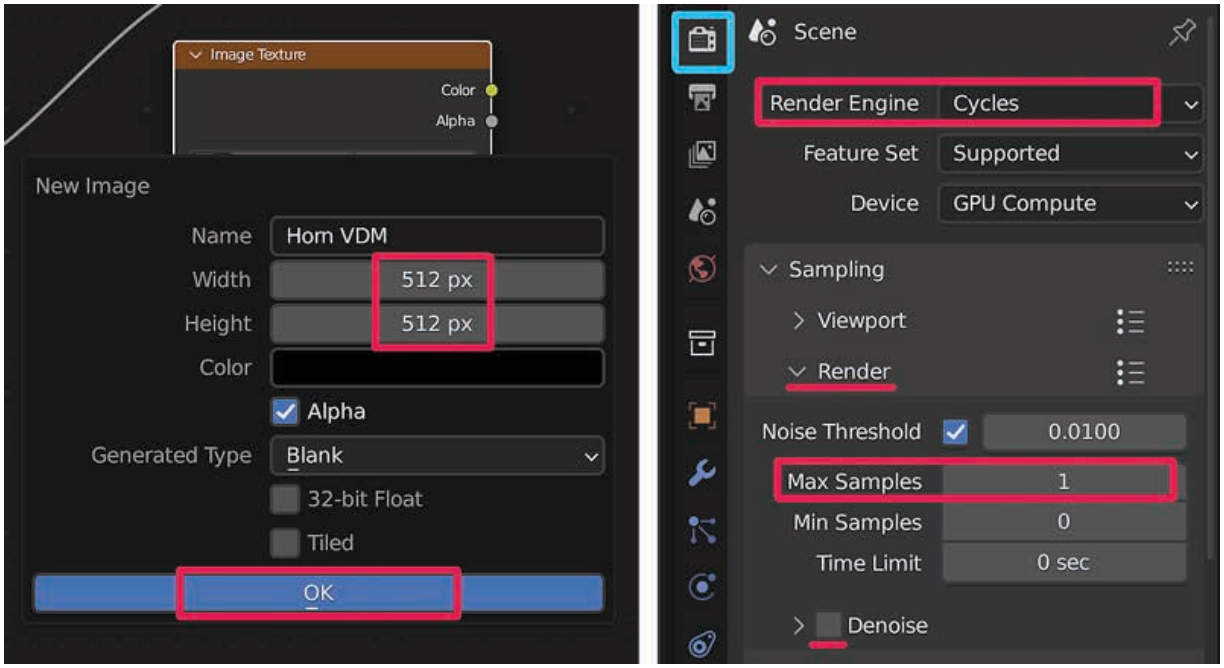
Assign it to your object in the Material Properties panel. You can see what it looks like in the Material preview. It's very colorful.



You can see the nodes for the baking material in the Shader Editor. Create a new image texture by **clicking on New** on the **Image Texture node** that's already there.

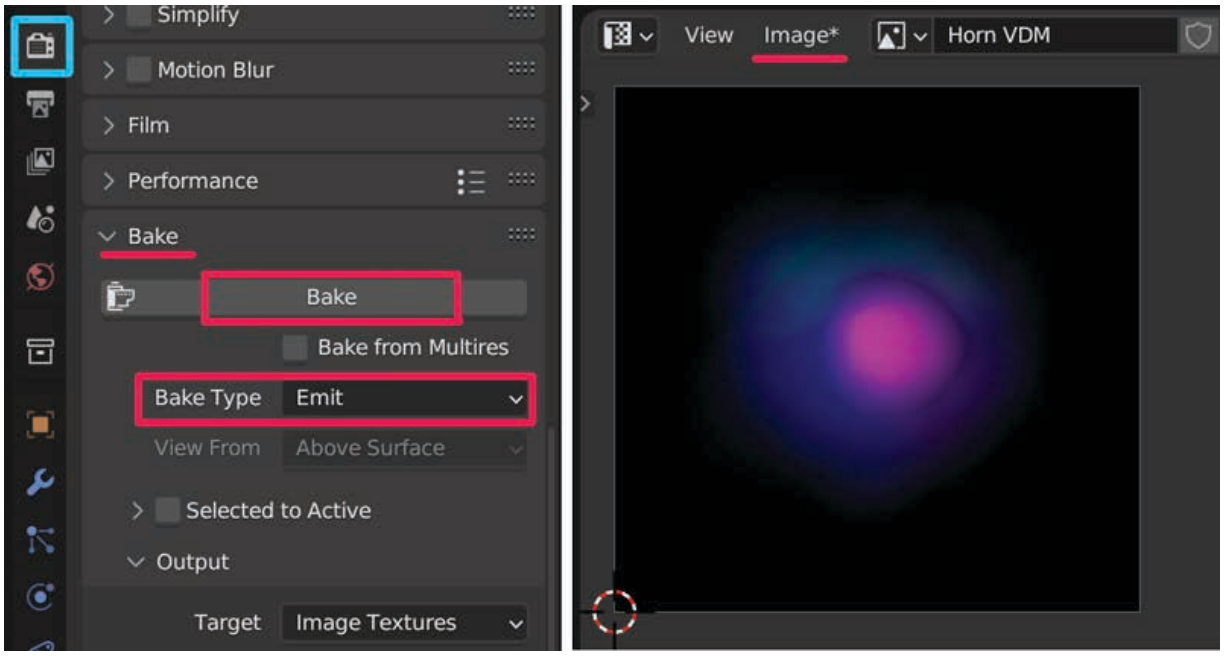


VDM maps don't need a lot of resolution, **512 x 512 i**s sufficient. It's always a good idea to name your texture as well. In Cycles, set Render Samples to 1 and uncheck Denoise, to save time. Otherwise baking takes forever, and 1 sample is enough when using the Emit bake type.

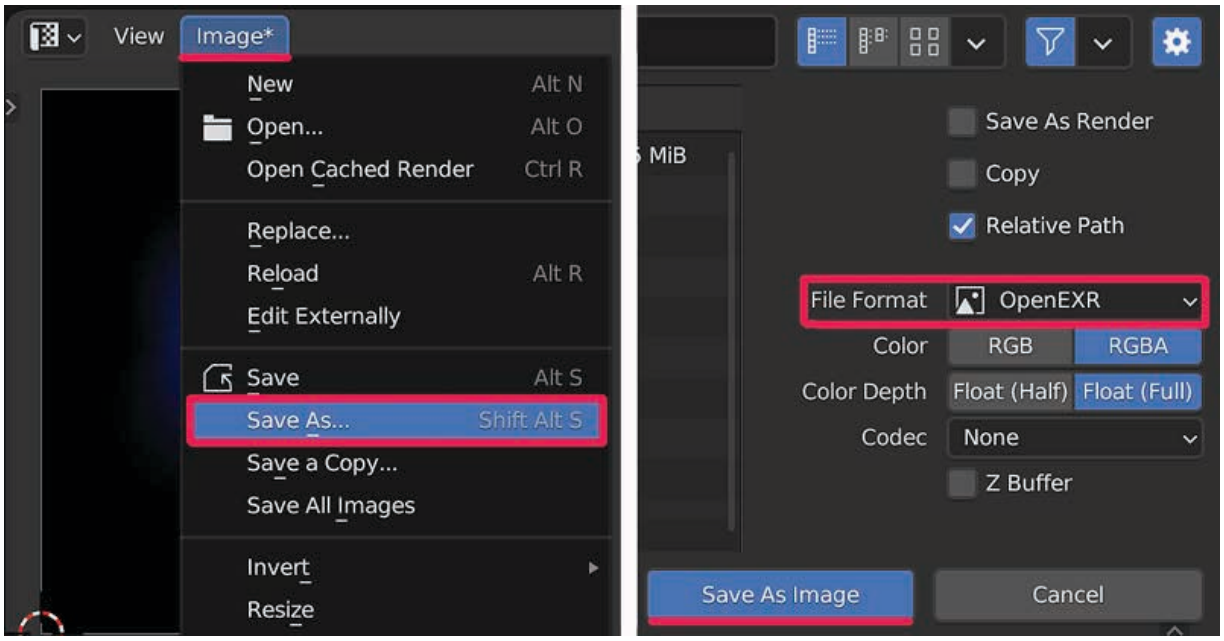


Speaking of which, scroll down to the bake options, and choose **Bake Type: Emit.** Then **click on Bake**. You can see the resulting image appear in the UV Editor window.

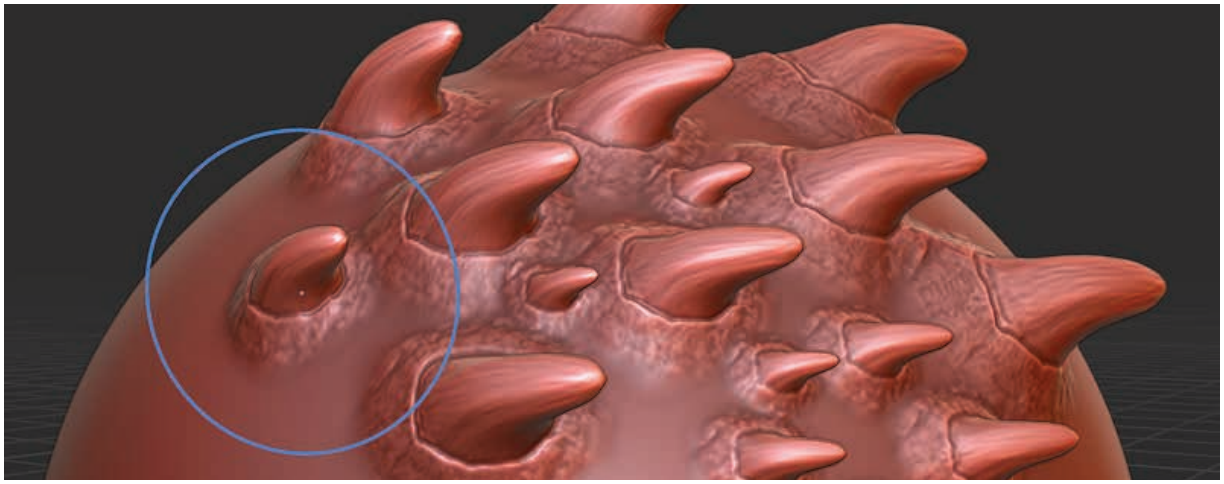
As you can see, VDM maps look surprising, when you're used to Height maps or Normal maps. It looks like something went wrong, because the shape is different to what you'd see from the top viewport. But this is correct.



Save the resulting VDM map as an **OpenEXR** file.

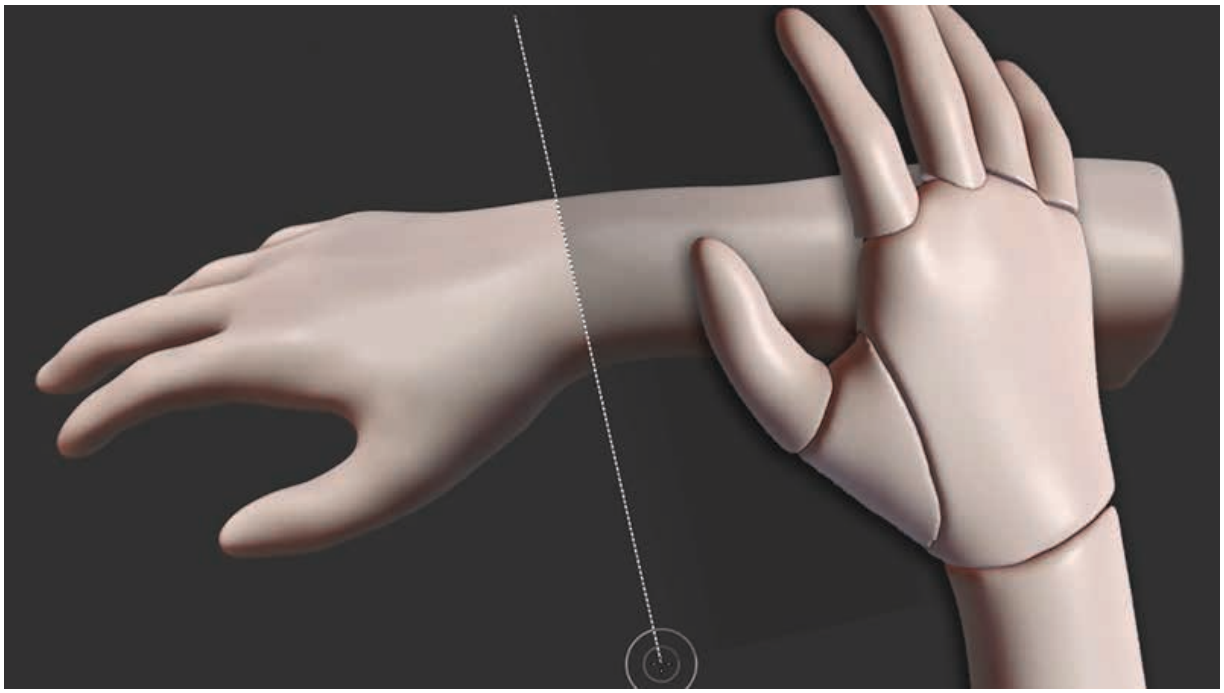


Now you can use it, as shown previously in the video about VDM brushes.



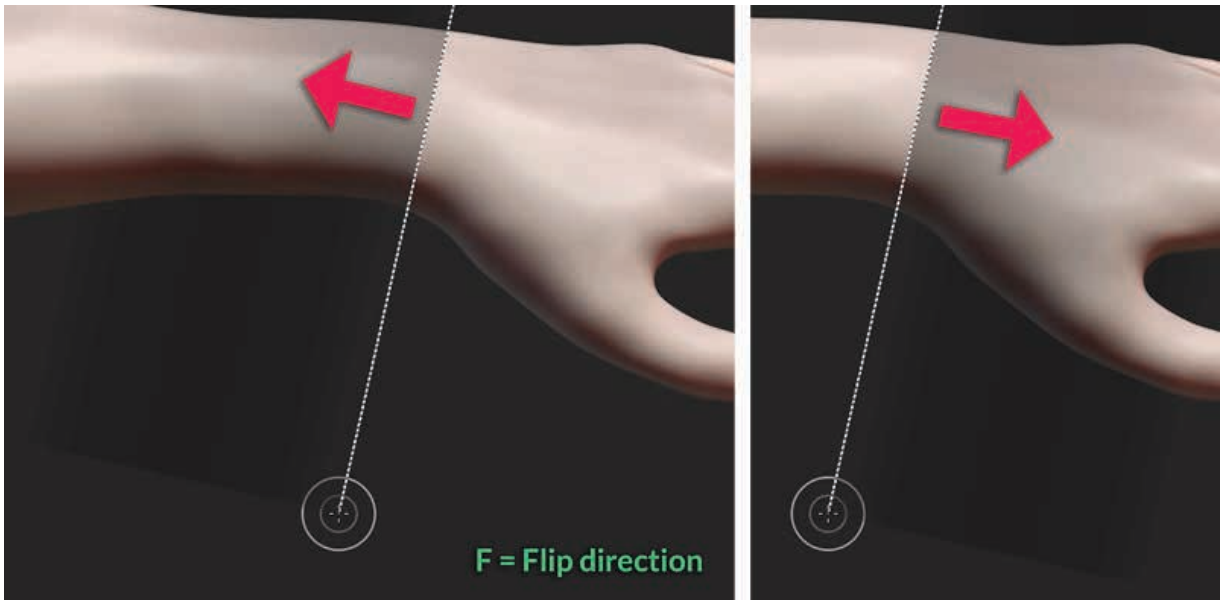
Slicing Objects in Sculpt Mode

To slice an object in Sculpt Mode, so that you can split it into two separate objects, first you need to mask part of it.

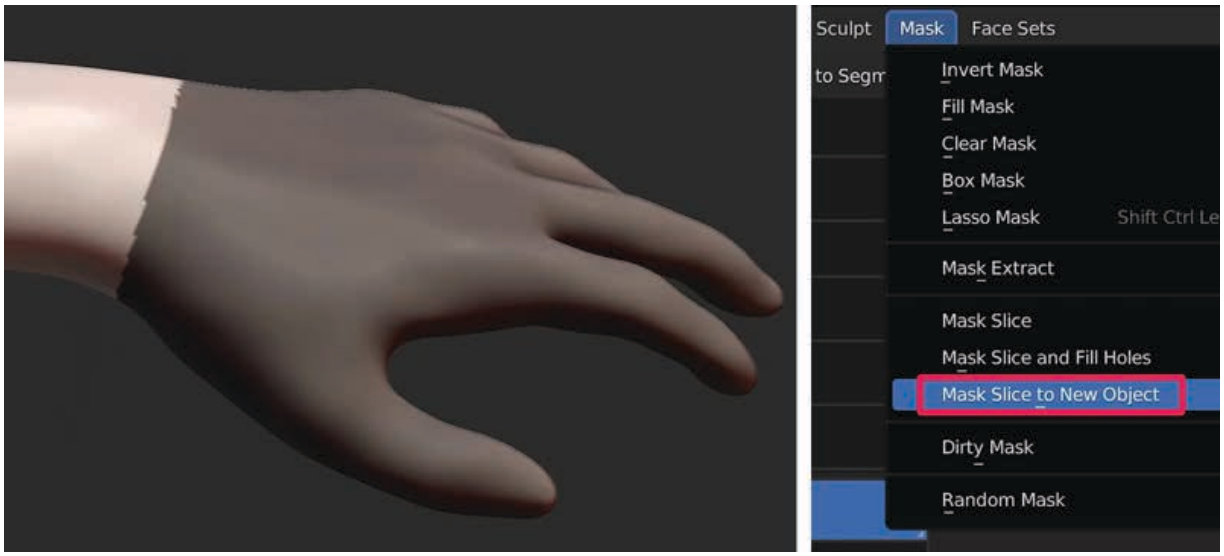


The **Box**, **Lasso** and **Line Mask** tools are useful for quickly drawing masks. With the Line Mask tool hold the LMB, then drag a line.

You can flip the direction by **pressing F** and move it by holding the Spacebar.

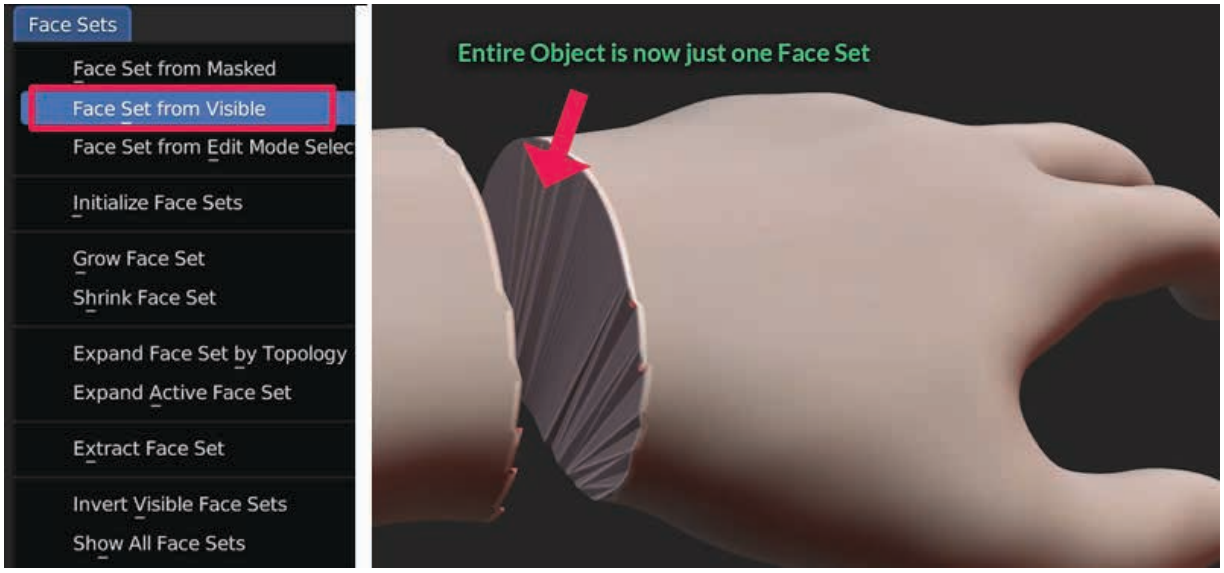


From the Mask menu choose **Mask Slice to New Object**.

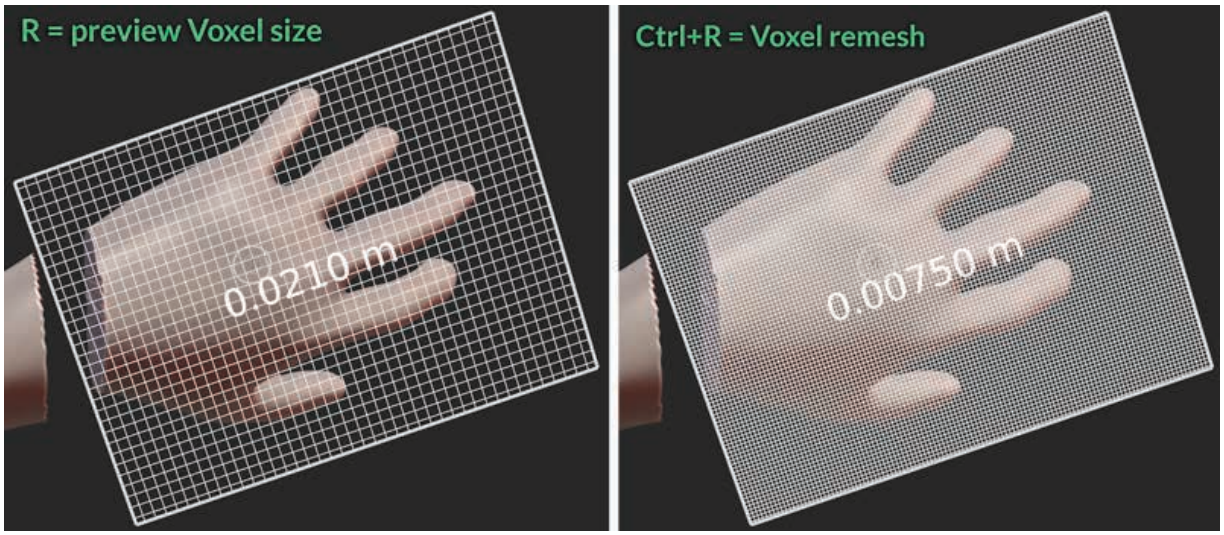


Now you've sliced the object into two objects. In Object Mode, you can move them apart. In Sculpt Mode, you can see that it's also created Face Sets for the filled geometry.

If you don't need those, choose **Face Set from Visible**.

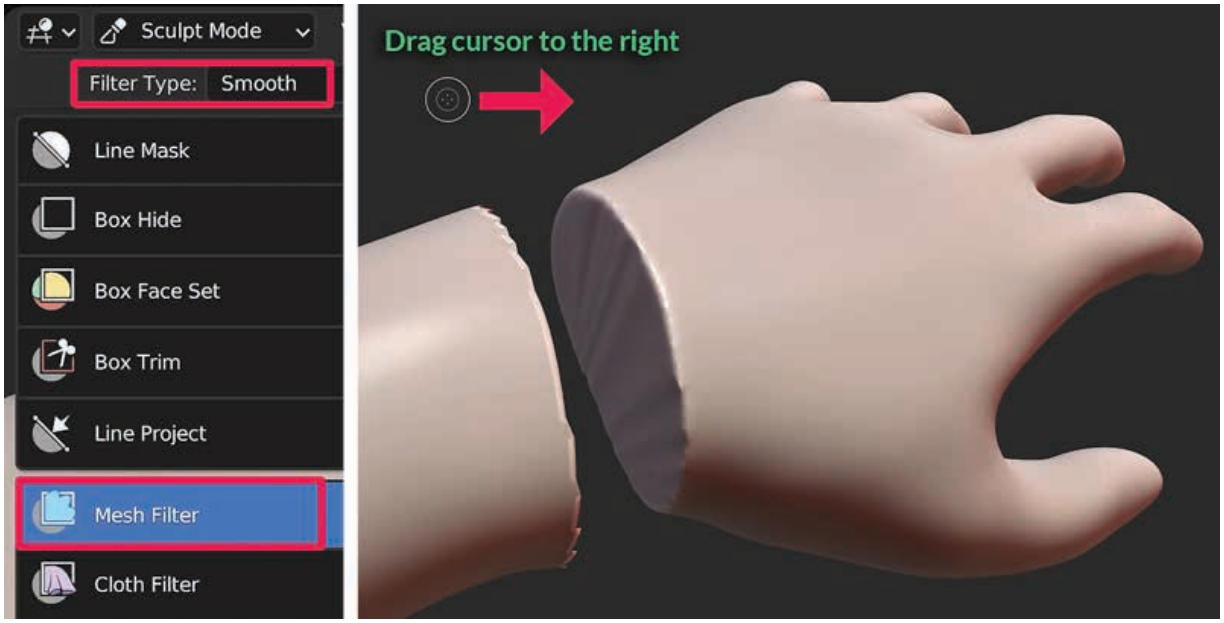


I recommend Voxel remeshing the geometry, with **R to preview (**before Blender 3.5: Shift+R) and **Ctrl+R to confirm**.



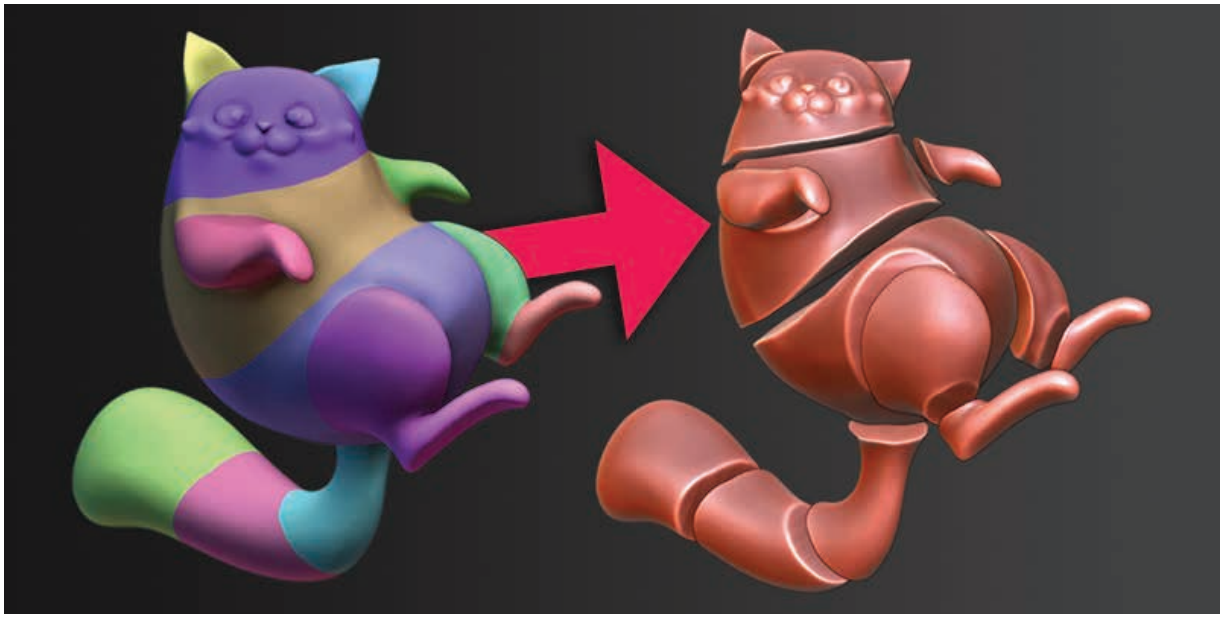
To avoid the kind of issue you see below, make sure you remove any left over mask before drawing a new mask by **pressing Alt+M** in Sculpt Mode.

You can smooth the corners with the **Mesh Filter** set to **Smooth**.

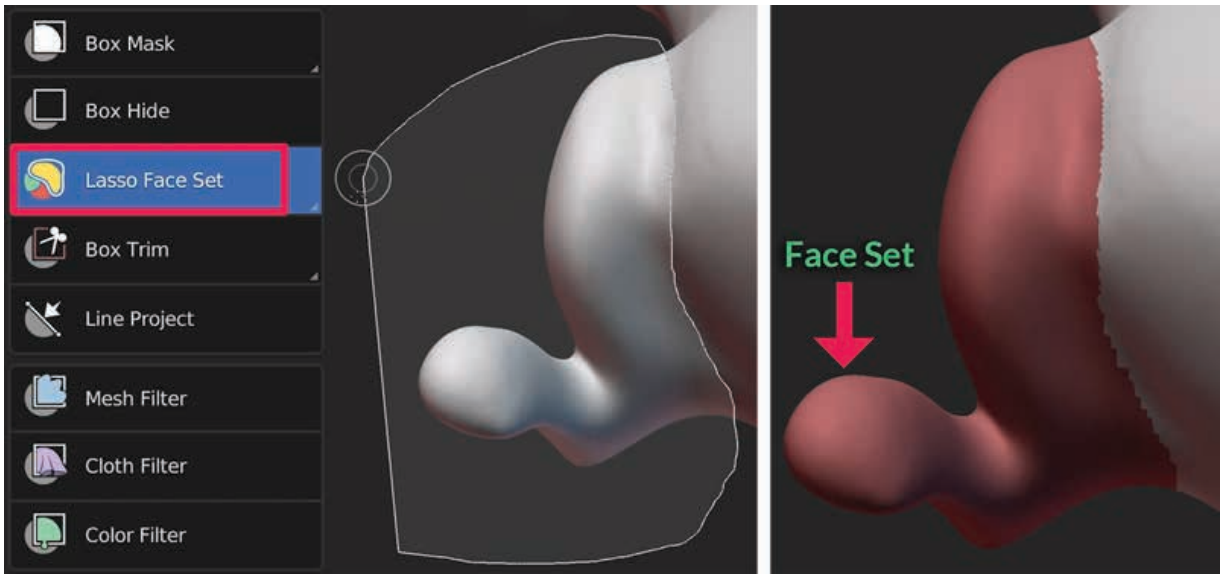


Slicing Objects using Face Sets

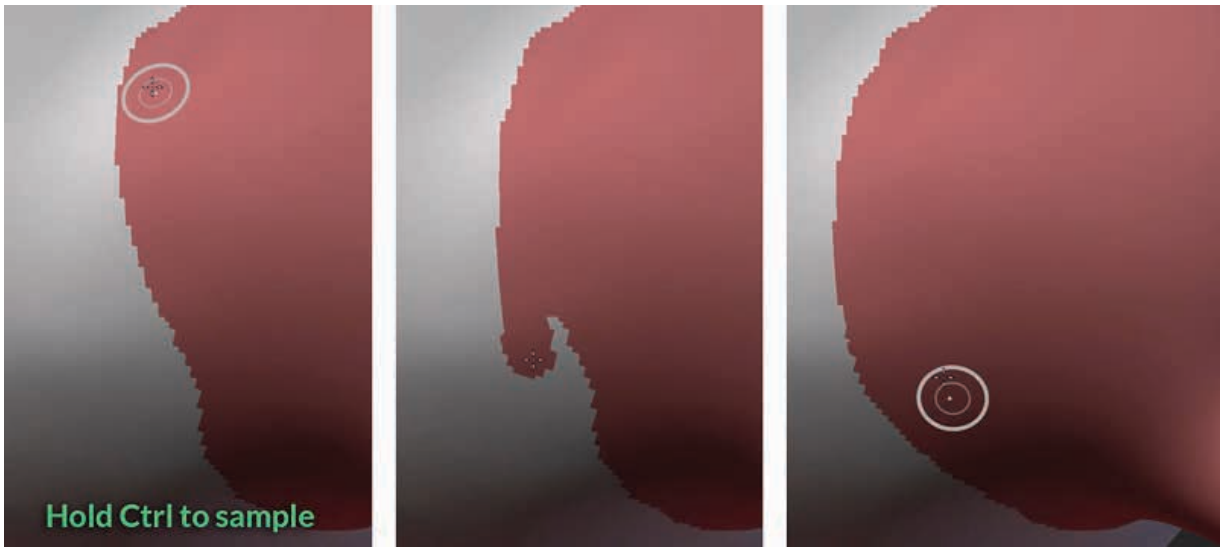
Previously we looked at slicing in Sculpt mode using Masks. If you want a sharper slice, you can use **Face Sets** instead.



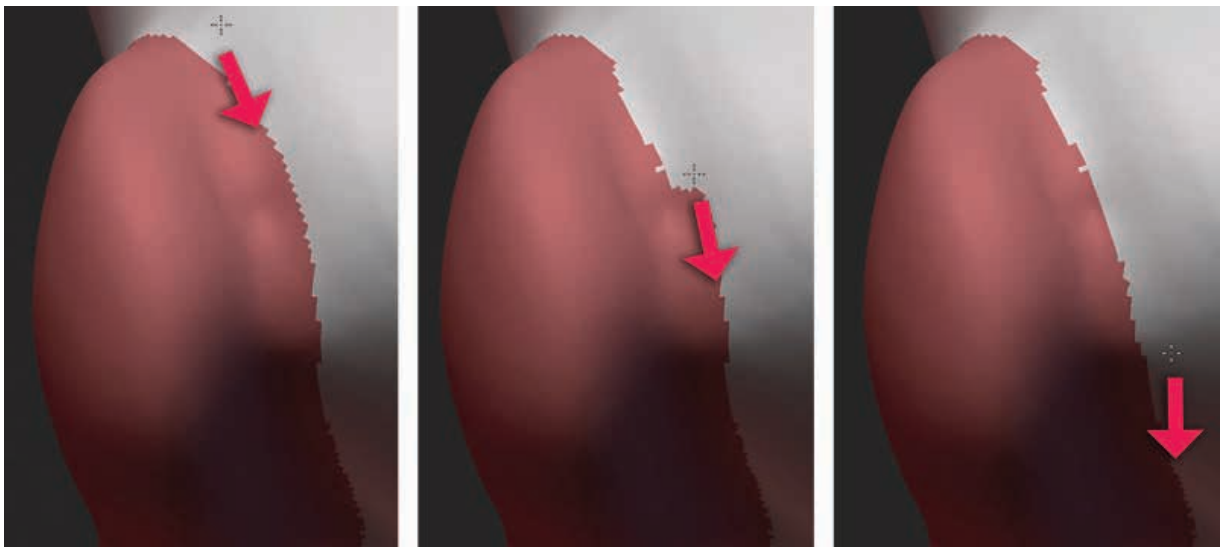
In Sculpt Mode select the **Lasso Face Set** tool and use it to draw a Face Set.



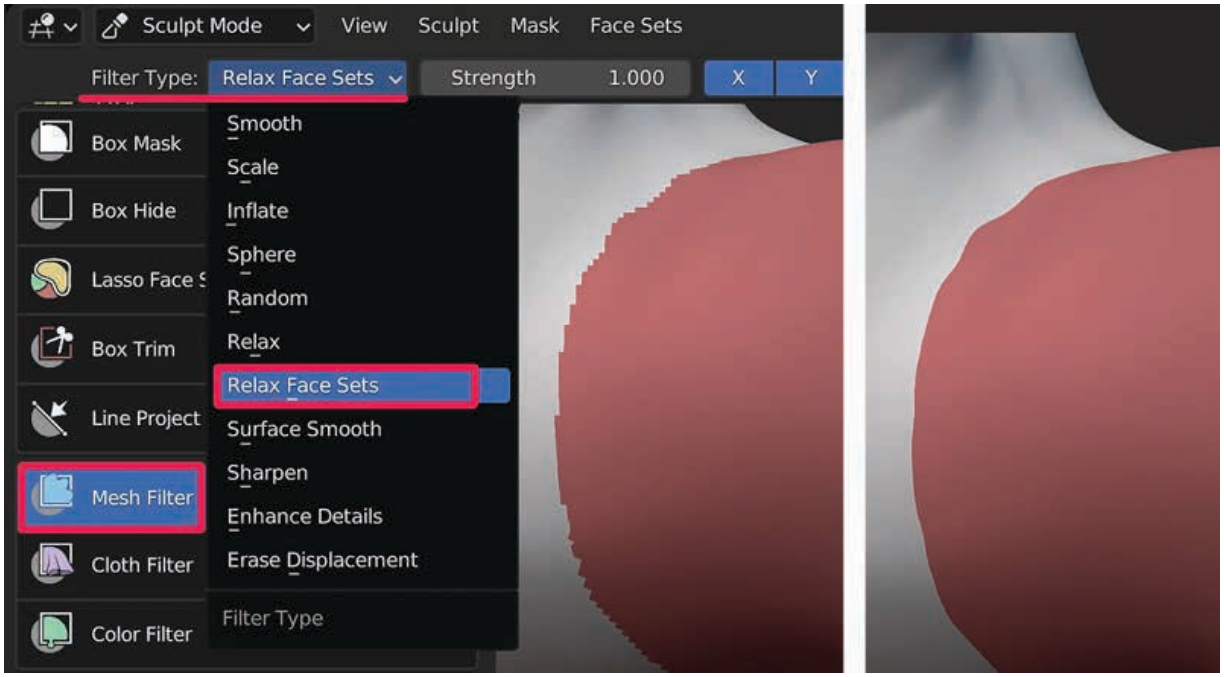
To edit the Face set, use the Draw Face Sets tool and **hold Ctrl** over the Face Set that you want to draw more of.



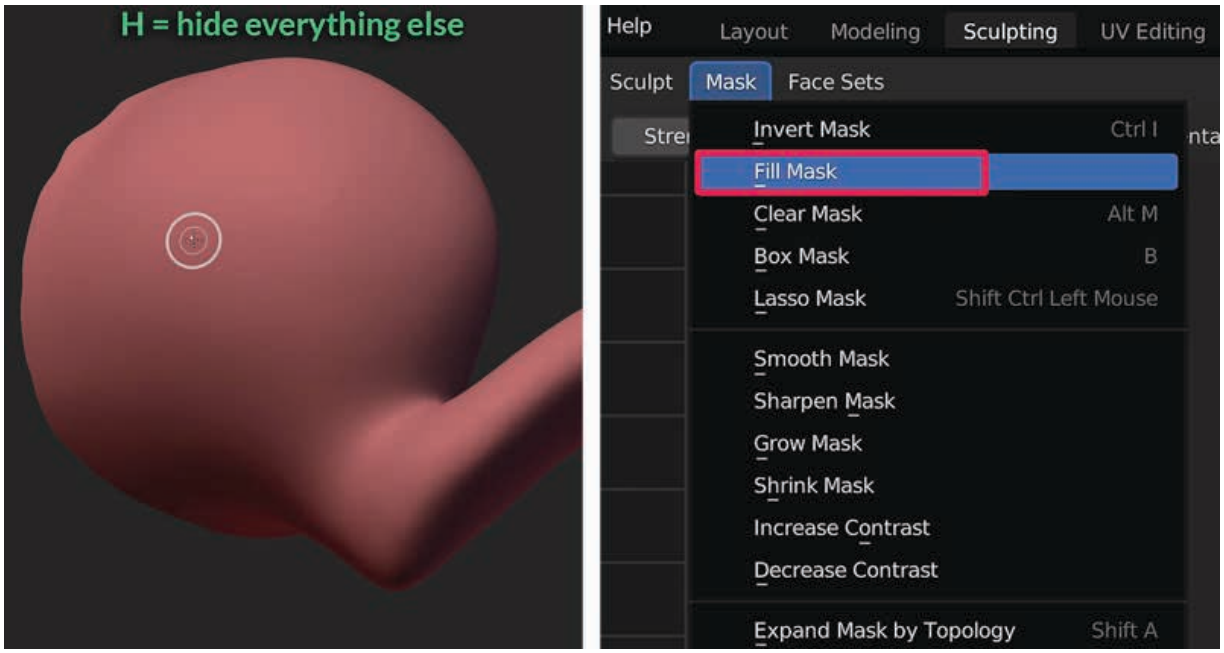
Note that the default grey color on a mesh is also a Face Set, so you can hold Ctrl over that Face set and draw to expand it, as well.



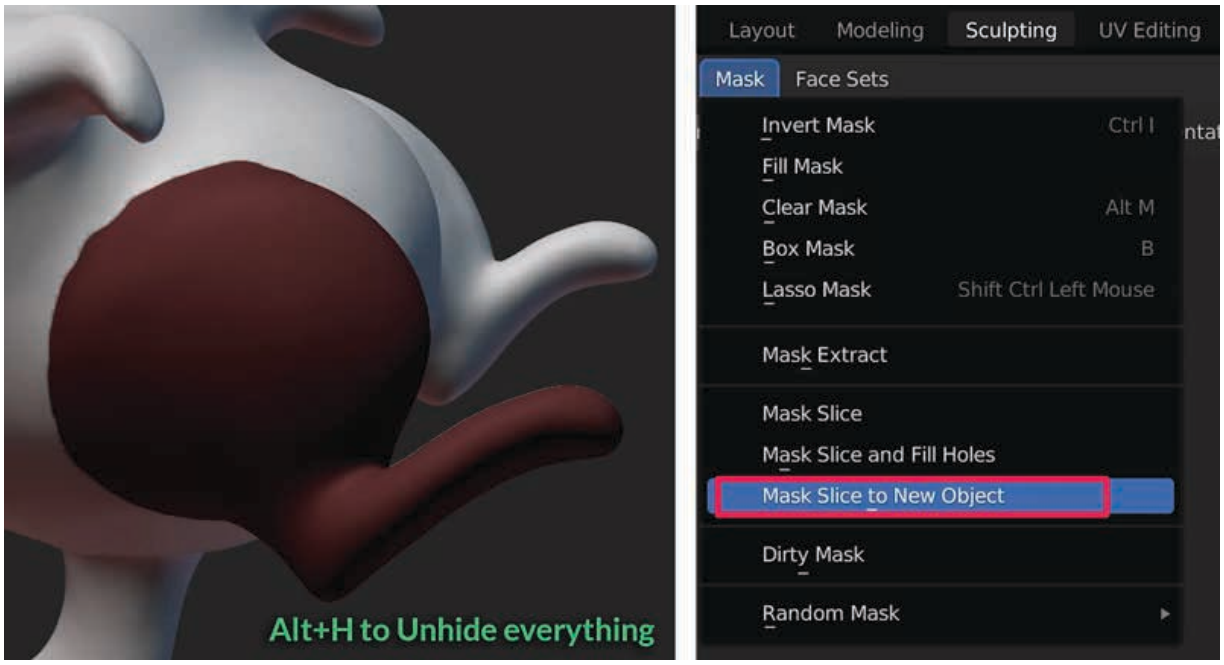
With the **Mesh Filter** set to **Relax Face Sets**, hold the LMB and drag to the right to make the Face Set borders smooth.



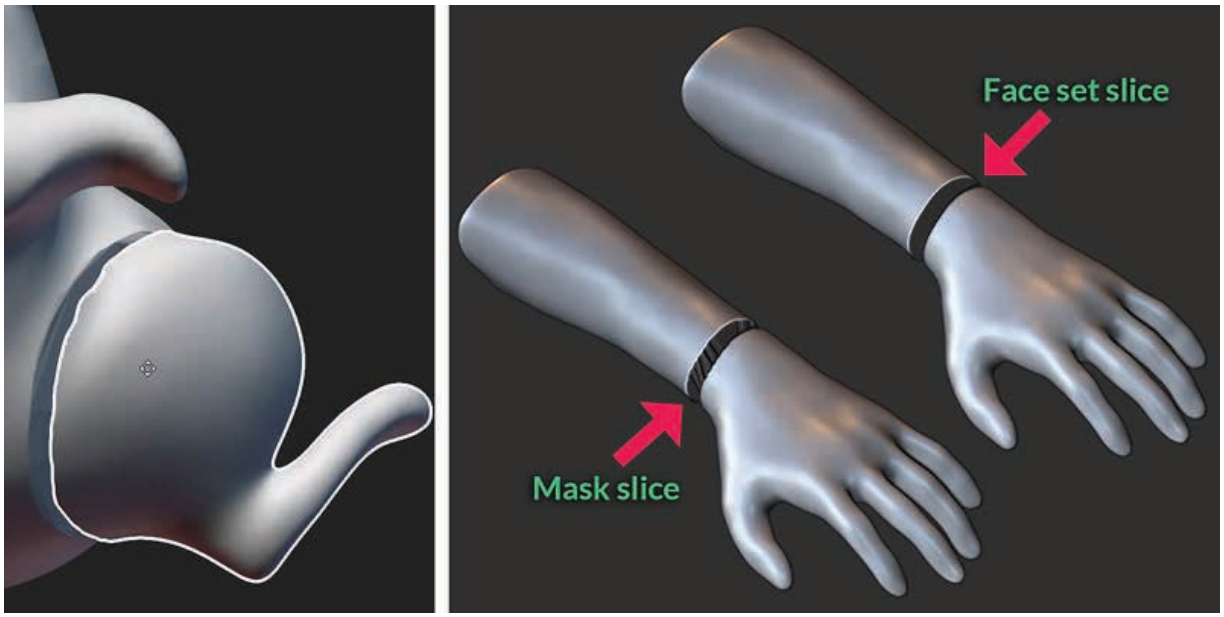
Press H while hovering over either Face Set to hide the other Face Set. Go to Mask > Fill Mask or **press Ctrl+i** to mask the visible geometry.



Press Alt+H to unhide the hidden part. From the Mask menu choose **Mask Slice to New Object**.

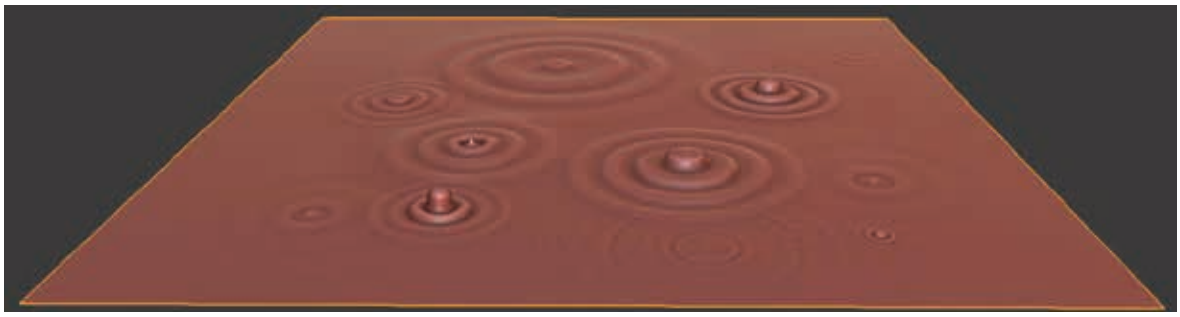


This gives a cleaner slice than if you used a Mask instead of a Face Set.

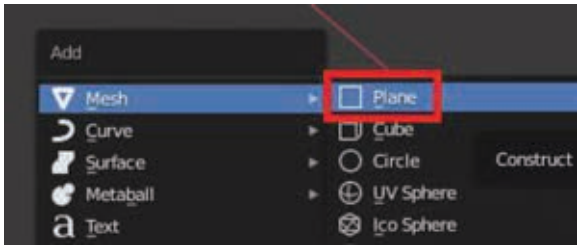


How to make stylized ripples

In this tip, we'll look at making some ripples on a surface, like this:

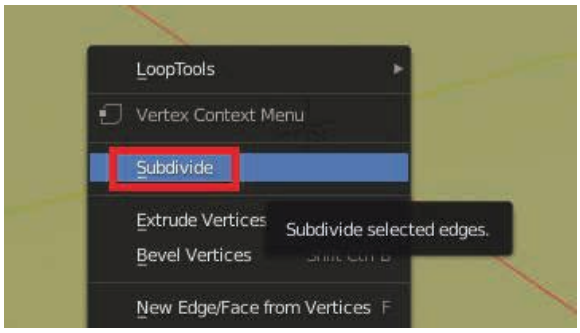


Let's get started by deleting our default cube (select it and **press X**).

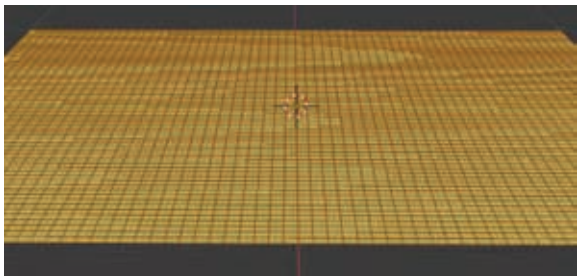


Then add a Plane (**Shift+A**, choose Mesh > **Plane**).

Press TAB to go into Edit Mode, right-click and from the menu choose "**Subdivide**".

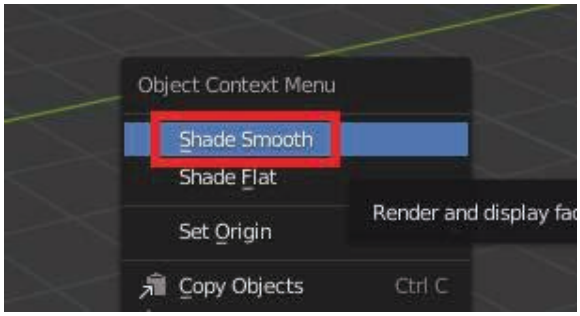


Press Shift+R to repeat that Subdivide action, 5 times. Now the plane has the necessary geometry to be displaced into ripples.

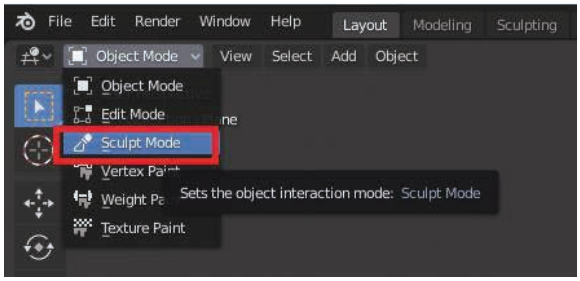


More (7 or 8) subdivisions will give a smoother result, if your computer can handle it.

Press TAB again to go back to Object Mode, right-click and choose **Shade Smooth**. That will create some additional visual smoothness.



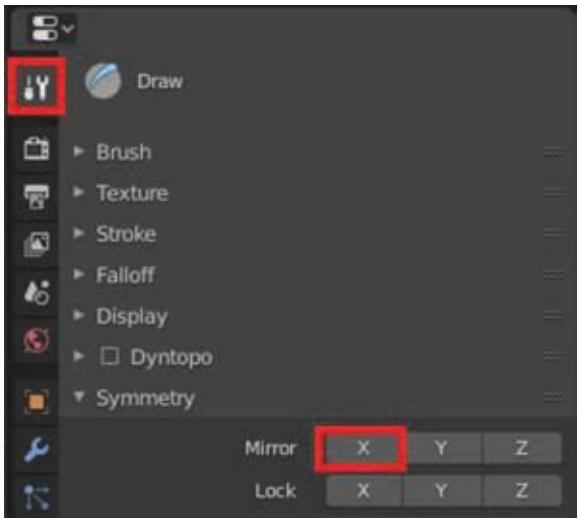
Go to **Sculpt Mode**.



Alternatively you can click on the Sculpting workspace button. This will rearrange the window layout to be optimal for sculpting.

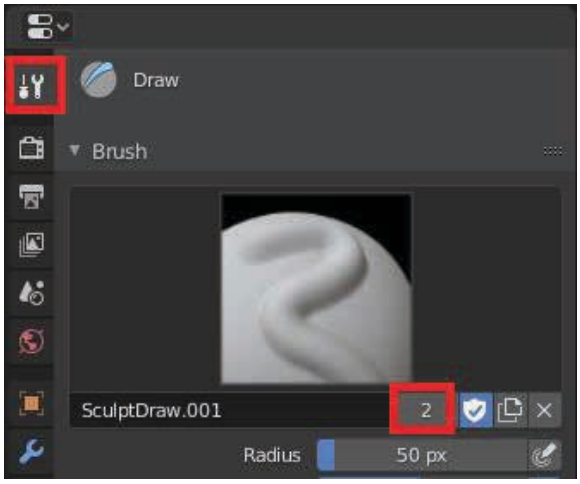
Before you do anything else, turn off "symmetry".

The X, Y and Z buttons should be gray, not blue. By default X symmetry is turned on (the icon is blue). It should look like this:



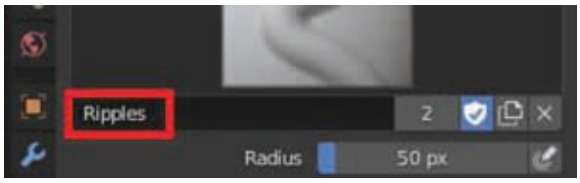
You'll notice that if you drag the cursor across the surface of the plane, you are sculpting on the surface.

Undo that by **pressing Ctrl+Z**, as it was just a test to see the sculpting.



Now the fun starts! Let's make a custom "Ripples" brush.

First, **click on the number next to the Brush**. This makes it an individual brush, in this case called **SculptDraw.001**.



Click on that name and type a new name like "Ripples". Of course this is not essential, but it's nice to stay organized!

If you scroll down you'll find the "Fallout" section.

Open it by clicking on the little arrow next to "**Falloff**". You will see a curve like the image on the right.

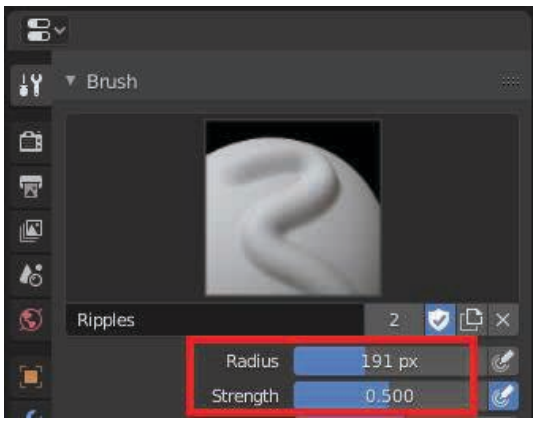
Think of this as half of the profile of your brush. We can click on the falloff curve to add new points, and then move those points. So what we need to do is make the falloff curve look like the side of a ripple.

Try it out - click on the curve and you'll notice that you've added a point. You can now move that point by clicking and dragging it.

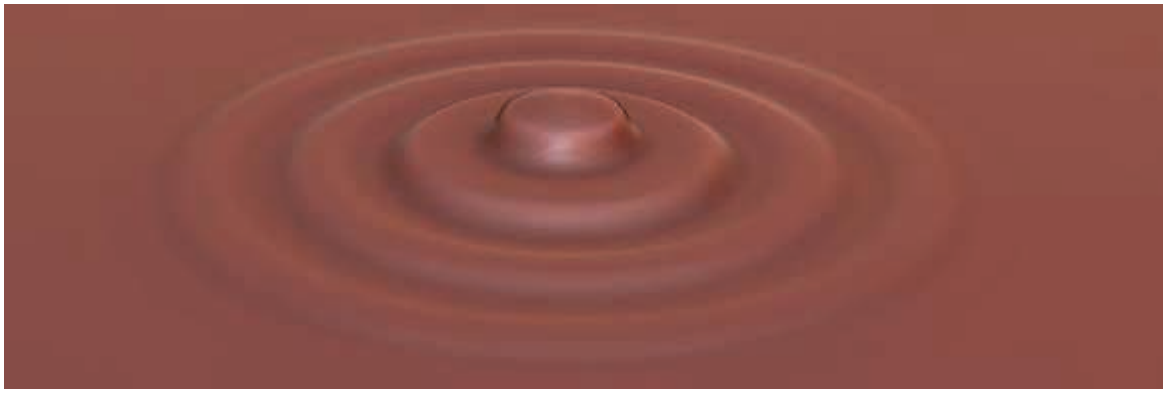
Here I move the new point down.

And in the image on the right you can see that I added some more points, and have dragged them and also the existing points down, to resemble the half side view of a ripple.

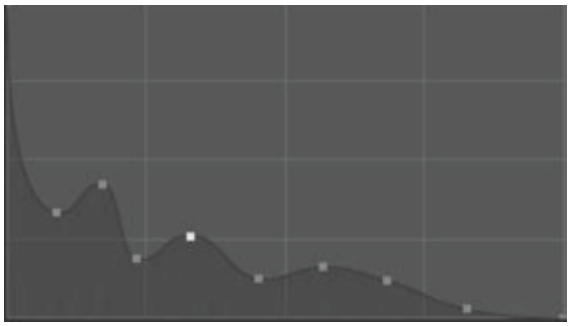
Now if we click anywhere on the plane, the result will be barely visible. We need to change the **Radius** and **Strength** of the brush.

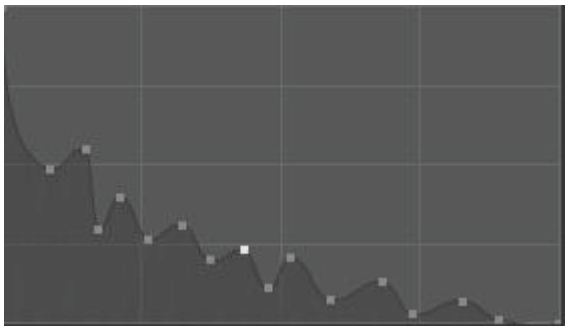


Now if we click on the plane, the ripple will be more visible.

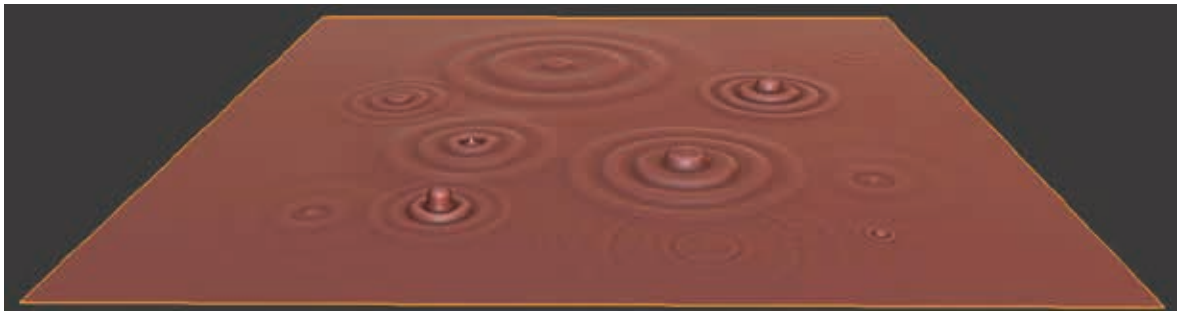


Of course, if you click again it will just produce the exact same ripple again. That's not realistic, so you'll need to both tweak the **Radius** and **Strength** settings each time you click on the plane to create a ripple, and change the falloff curve each time. You can experiment with different values - if it doesn't look good, just undo with **Ctrl+Z** and try different values.

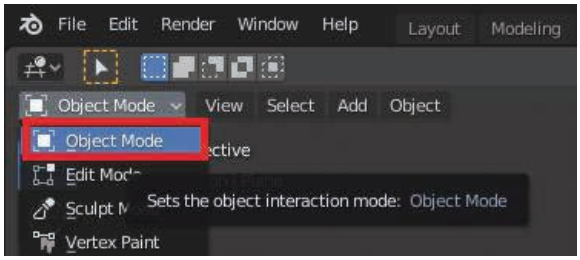




Some different falloff curves
The result:



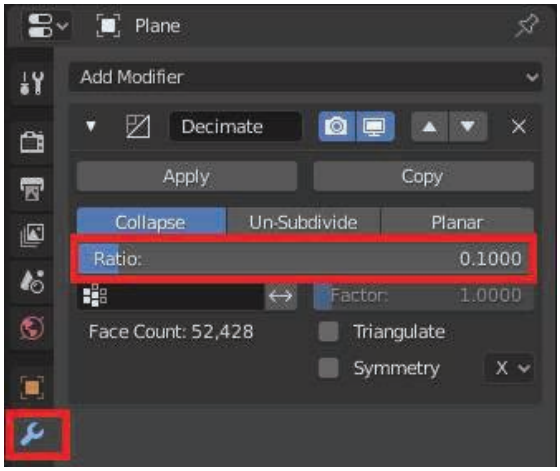
If you are satisfied with it, perhaps you'd like to 3D print it as well!



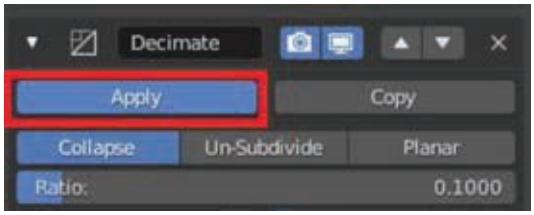
But the model is quite heavy now in terms of the amount of faces. To 3D print it's a good idea to decimate the model first.

To do that, go back to Object Mode and add a **Decimate modifier**to it

A Ratio of **0.1** is usually a good starting point with the Decimate modifier. It may take a minute to process the decimation, so be patient. Don't worry if Blender is unresponsive for a while.

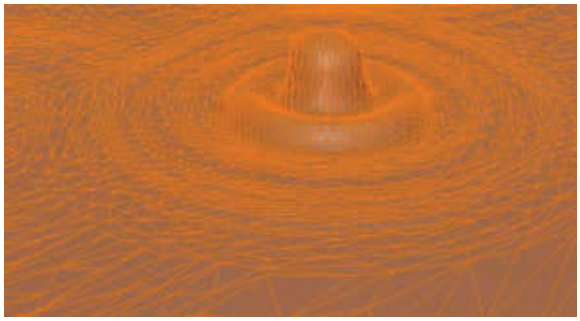


Click on **Apply**. It will probably need a minute to process this.

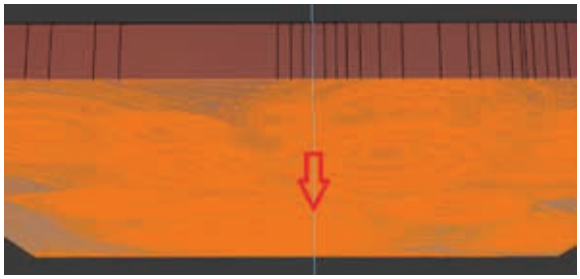


Now if you go into Edit Mode by **pressing TAB**, you'll notice that the geometry is sparse in flat areas, and more dense in highly detailed areas like the ripples.

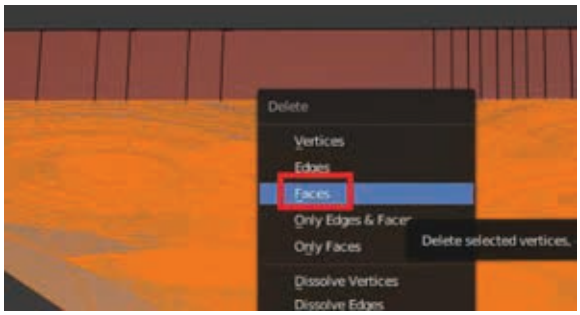
That's good! This drastically reduces the amount of face and the size of your file. To your eye it will still look the same.



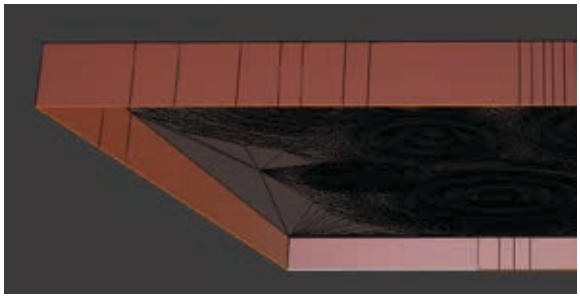
Now to print this we'll need to give it some thickness. Select all the faces (if they aren't selected) by **pressing A**. Then, **press E** (to extrude) **and Z** (to only extrude on the Z-axis). Move the extrusion down.



Then, delete all the selected bottom faces by **pressing X** and choosing **Faces**.

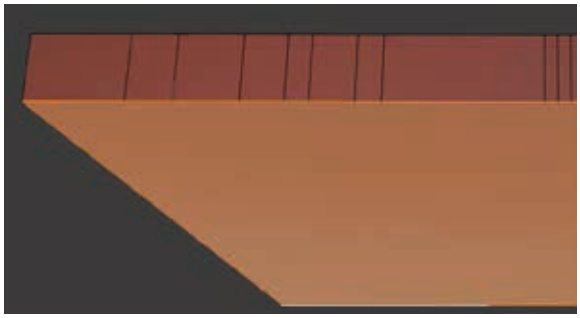


Click on the bottom edge loop and **press F** to fill the hole.

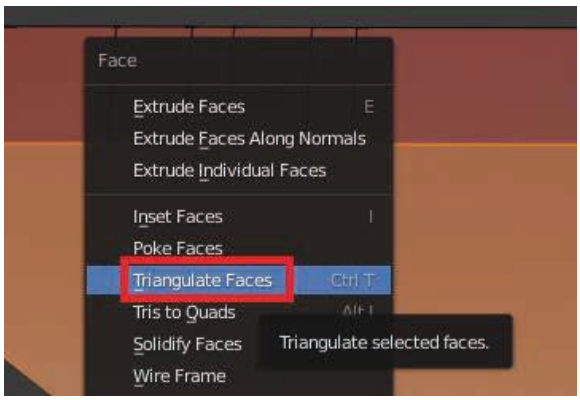


That will fill it, but creates a big N-gon (a face with more than 4 vertices).

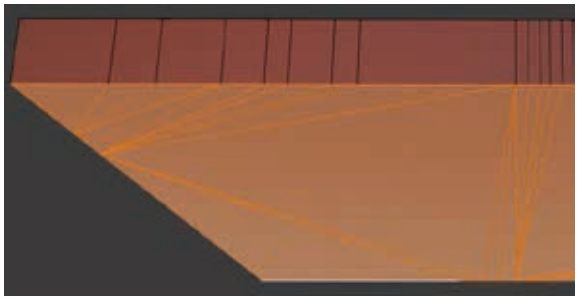
That might cause problems with some 3D printers, so just to be sure we'll fix it.



Press Ctrl+F for the Face menu and choose "**Triangulate Faces**" (or shortcut: Ctrl+T).

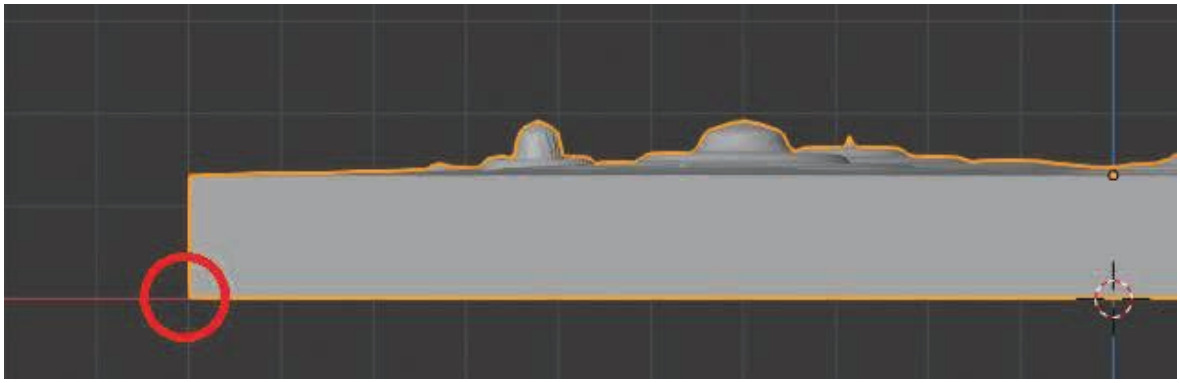


Now the bottom will look like this.

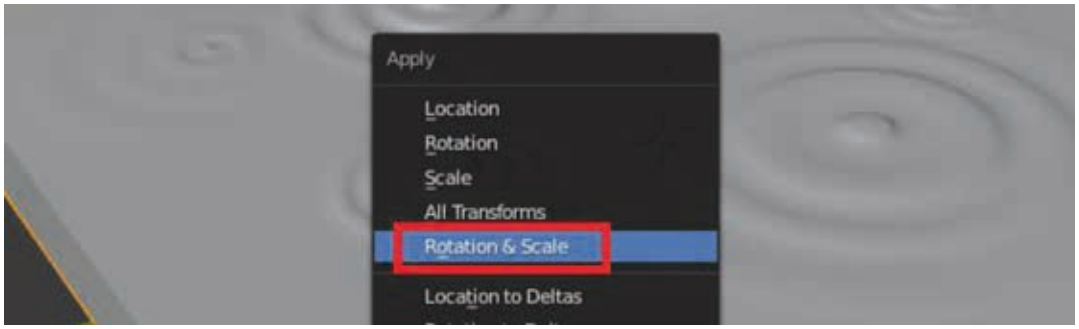


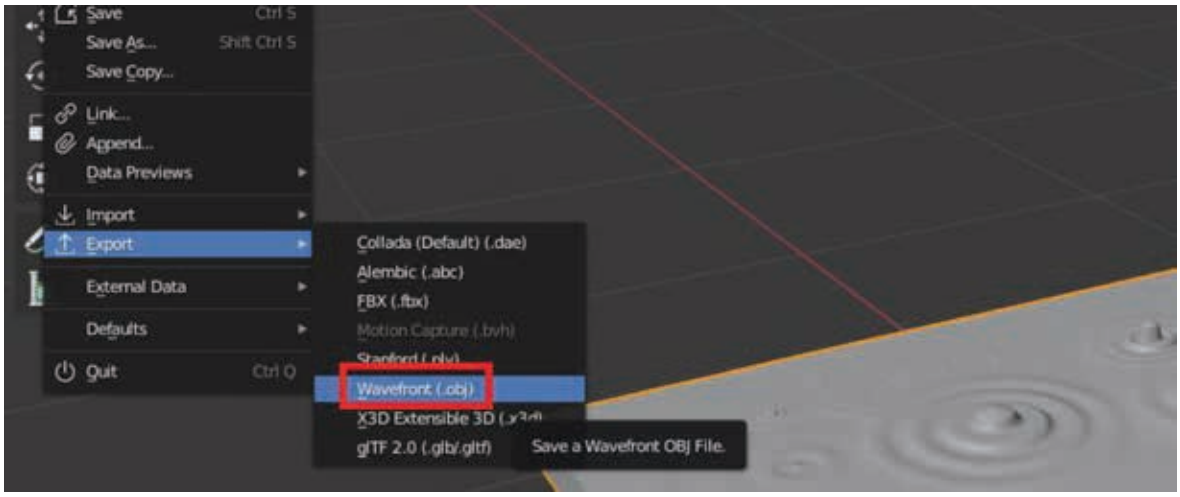
Almost done! Go back to Object Mode with **TAB**, and go back to the Modeling layout. From the side or front view (**3 or 1 on the numpad**), check that the model is flat on the "ground". Most likely, it is not.

Select the object and move it up on the Z-axis by **pressing G, then Z** and moving it up. It should rest on the red line, like this.



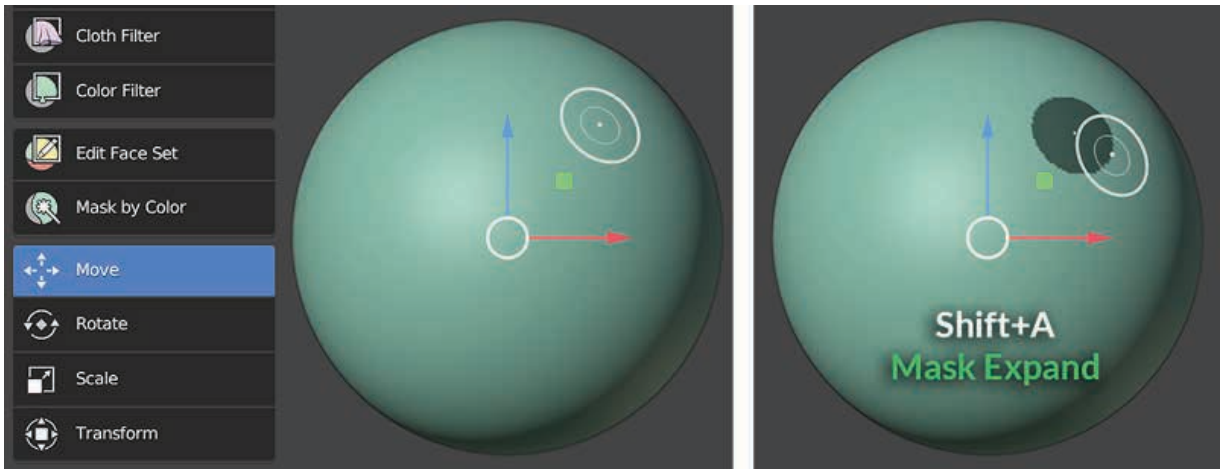
Just to be extra safe, **press Ctrl+A** and choose "Rotation and Scale" so that those values are applied.



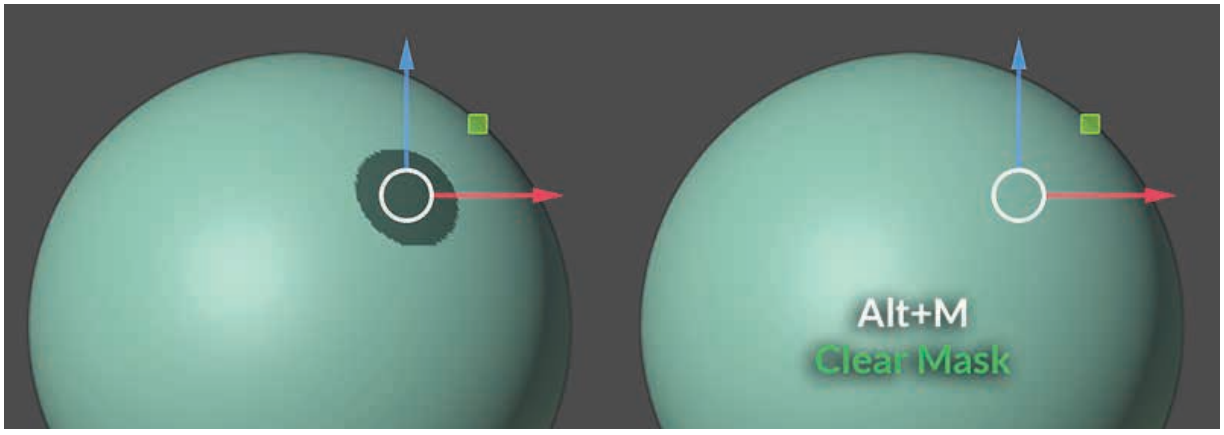

Save
Save As...
Save Copy...
Link...
Append...
Data Previews
Import
Export
External Data
Defaults
Quit
Collada (Default) (.dae)
Alembic (.abc)
FBX (.fbx)
Wavefront (.obj)
glTF 2.0 (.glb/.gltf)
Save a Wavefront OBJ File.

Move the Pivot Point in Sculpt Mode

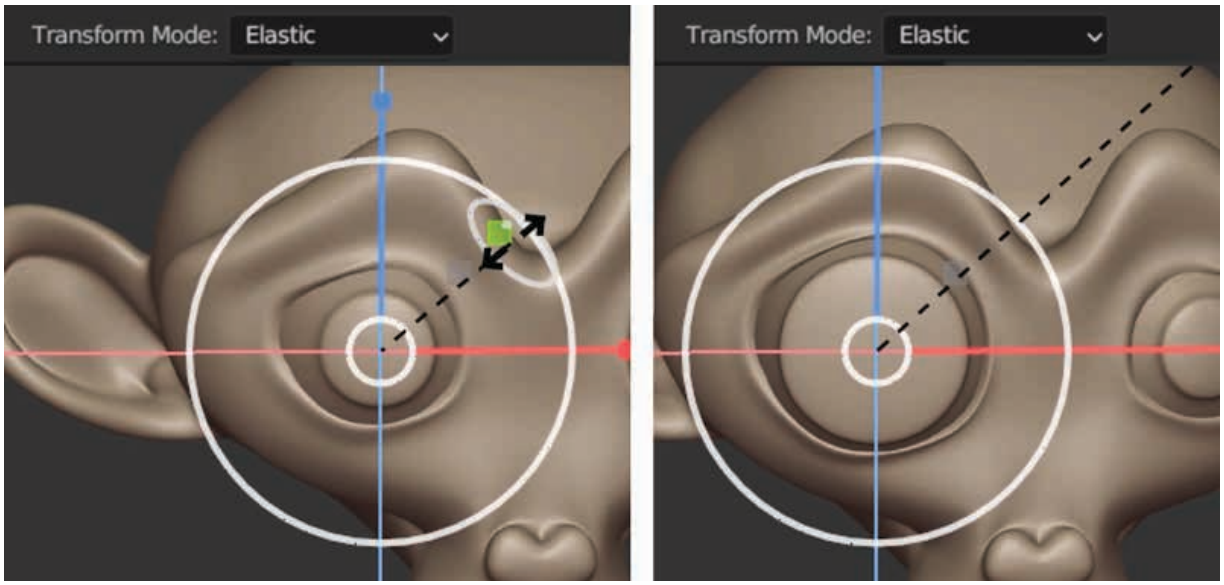
In **Sculpt Mode**, it can be useful to be able to move the **Pivot point**. One way to do this, is by using the **Expand Mask** function, by pressing **Shift+A**. The pivot will automatically be moved to that location.



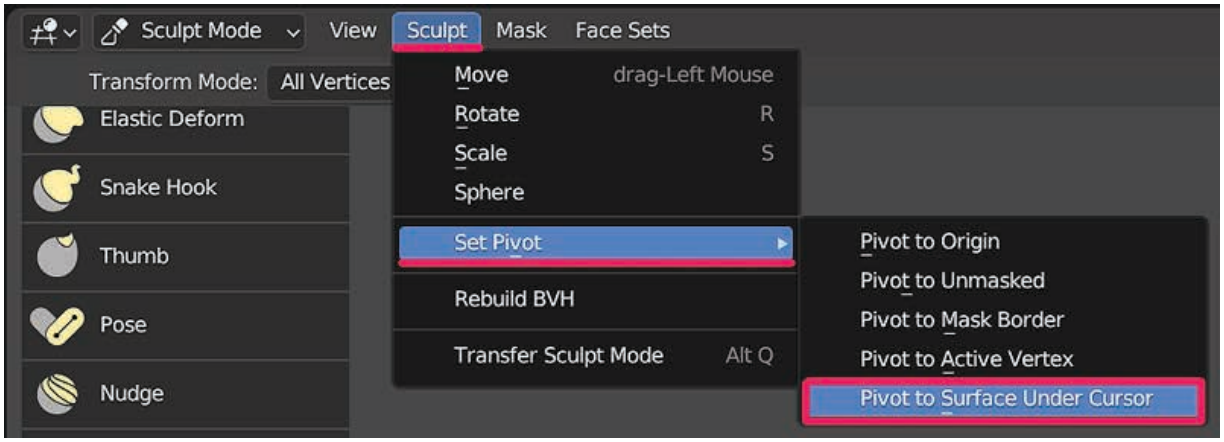
After that, you can just clear the mask with **Alt+M**.



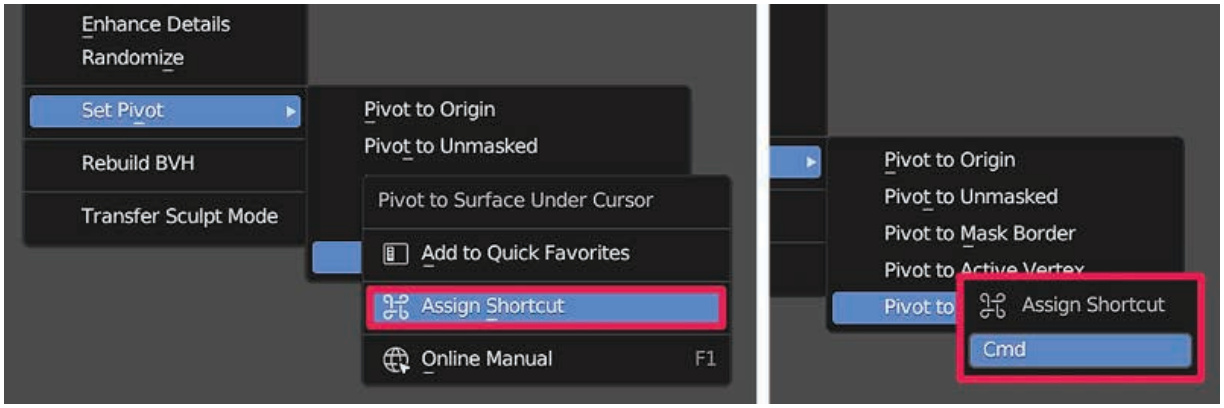
This is useful for example when using the (Sculpt mode tool) **Scale** set to **Elastic**, as a method of scaling up specific parts of a model.



You can also set up a custom keyboard shortcut if you don't want to have to draw a mask every time. Go to **Sculpt > Set Pivot > Pivot to Surface under Cursor**. But don't click on that option!



Instead of clicking on it, **right-click** and choose **Assign Shortcut**. Then press a key of your own choosing.

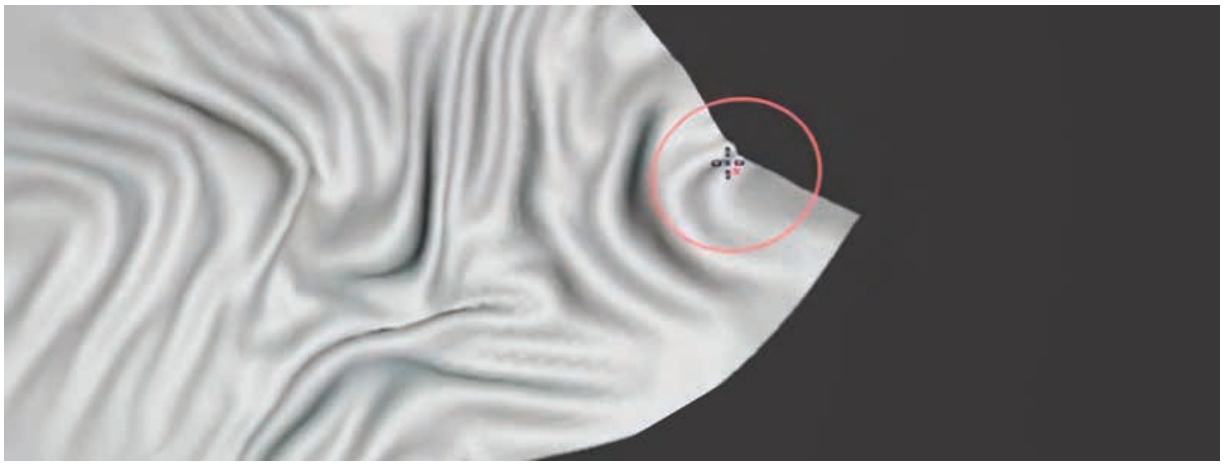


Now, every time you invoke this shortcut, the sculpting pivot point will be set to wherever you are hovering over at that moment.

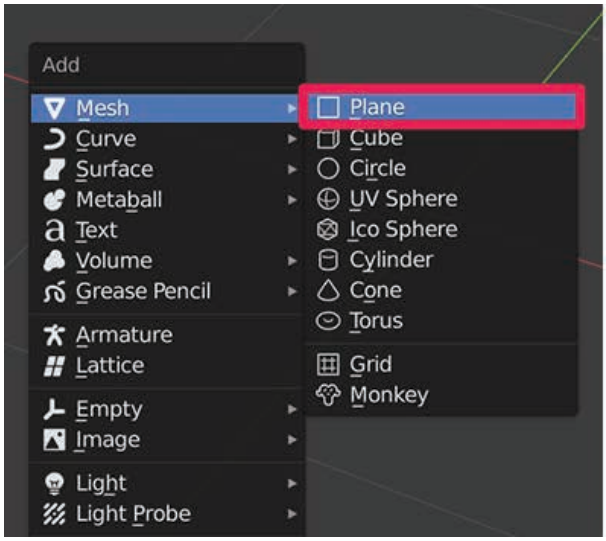
SCULPTING CLOTHING

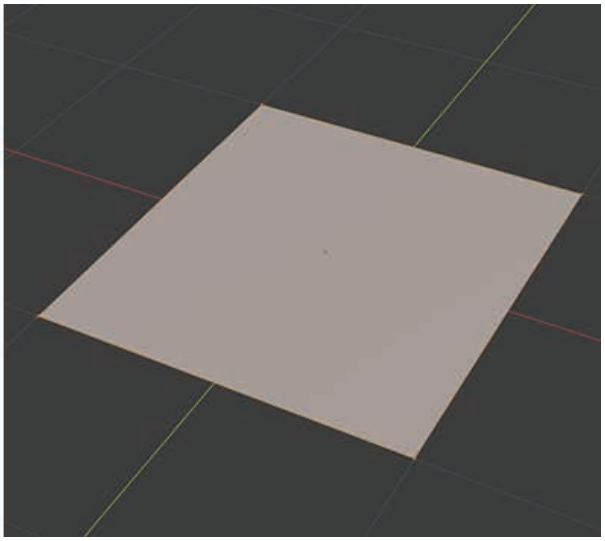
Sculpt clothing folds with the Cloth Brush

The Cloth Brush is an amazing feature for creating realistic cloth folds.

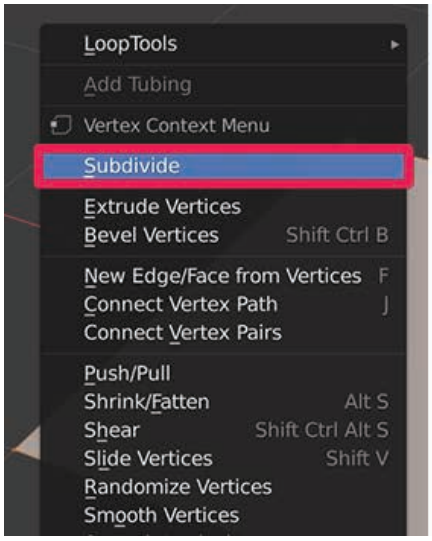


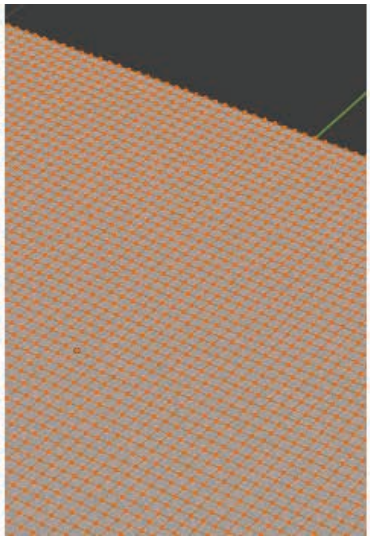
Let's try it out. **Press Shift+A** and choose **Mesh > Plane** to add a Plane.

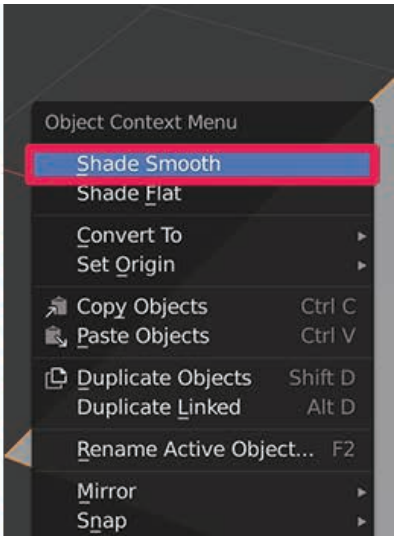




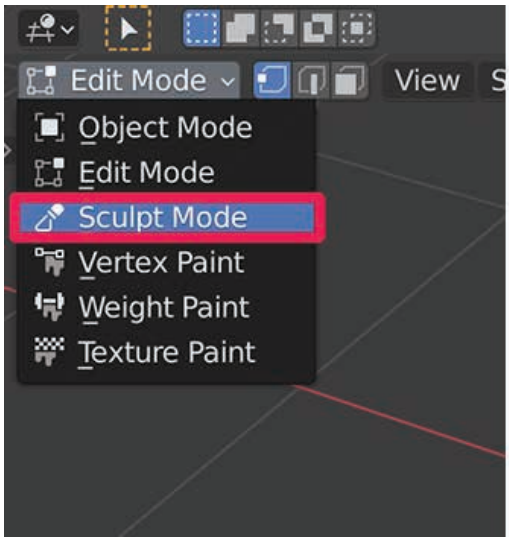
Add some subdivisions, by right-clicking in Edit Mode and choosing Subdivide, then **press Shift+R** a few times to repeat. Right-click and choose **Shade Smooth** in Object Mode.

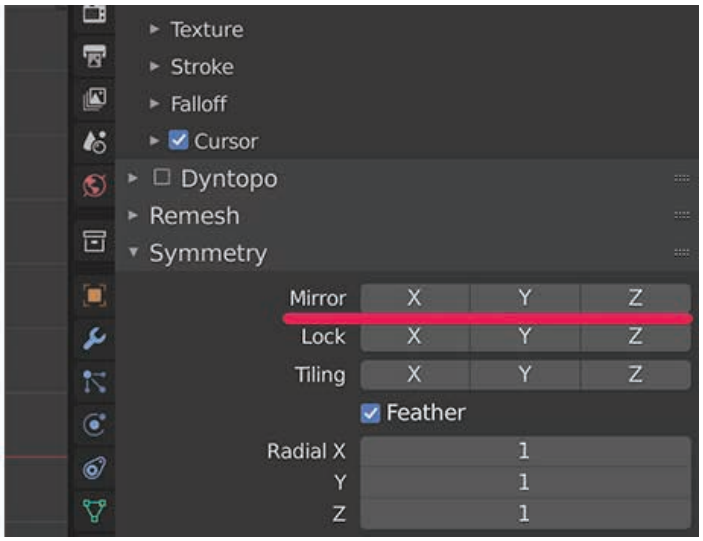




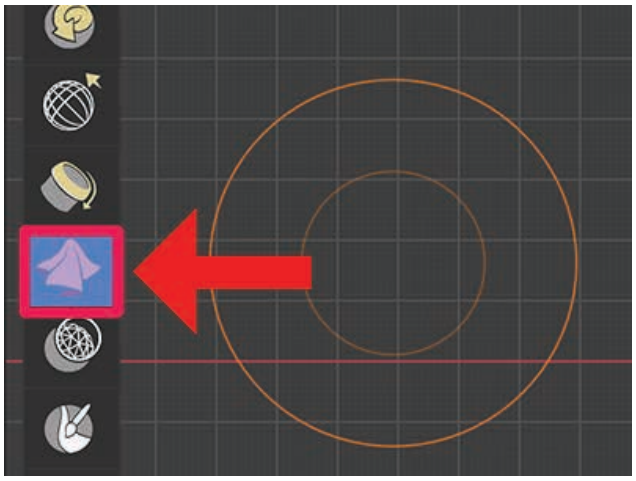


Go to the Sculpting layout and turn off symmetry.

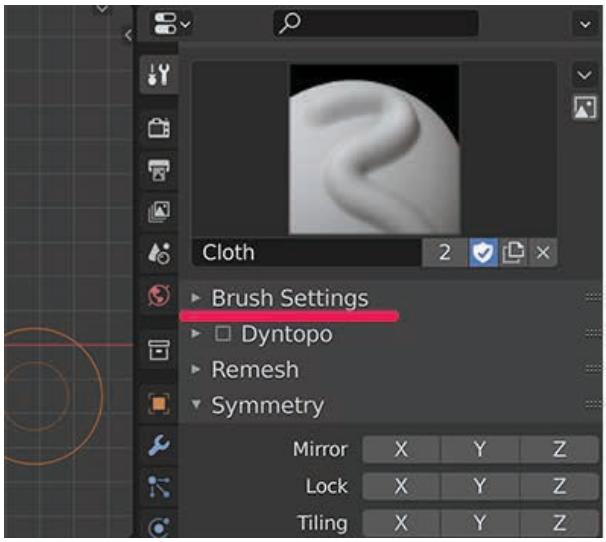


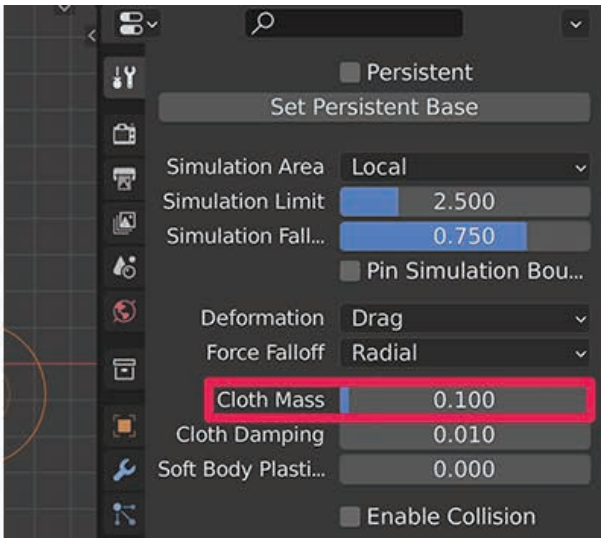


Select the **Cloth brush**.

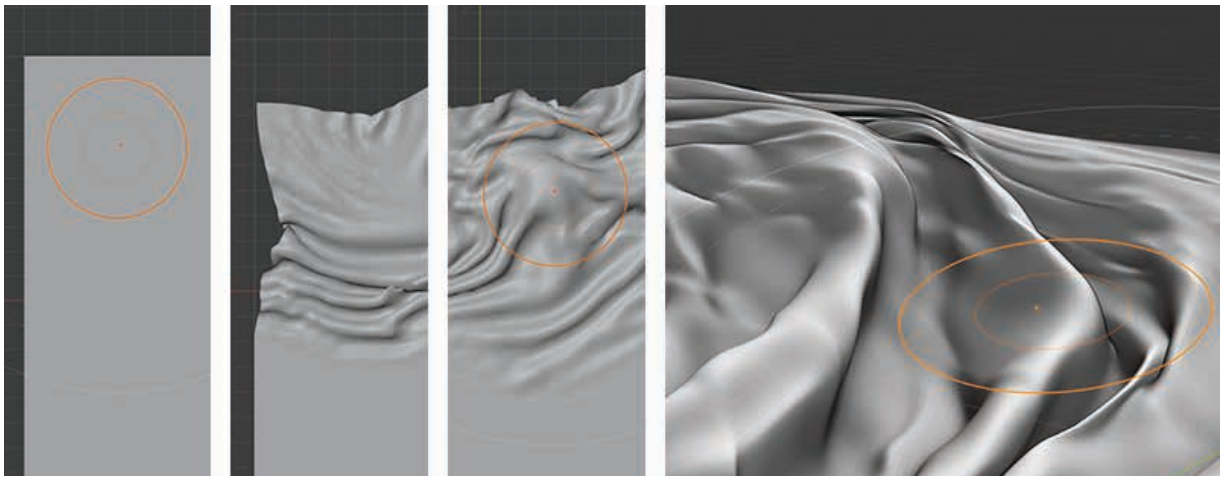


You may need to change the **Cloth Mass** value to something like 0.1.





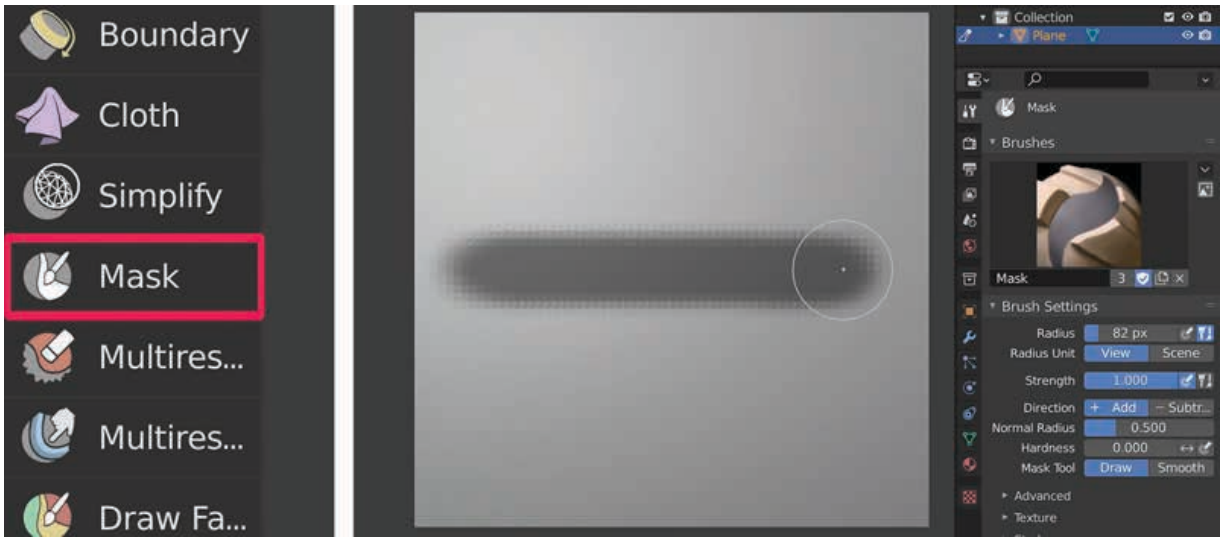
And now you can just drag and see the magic happen.



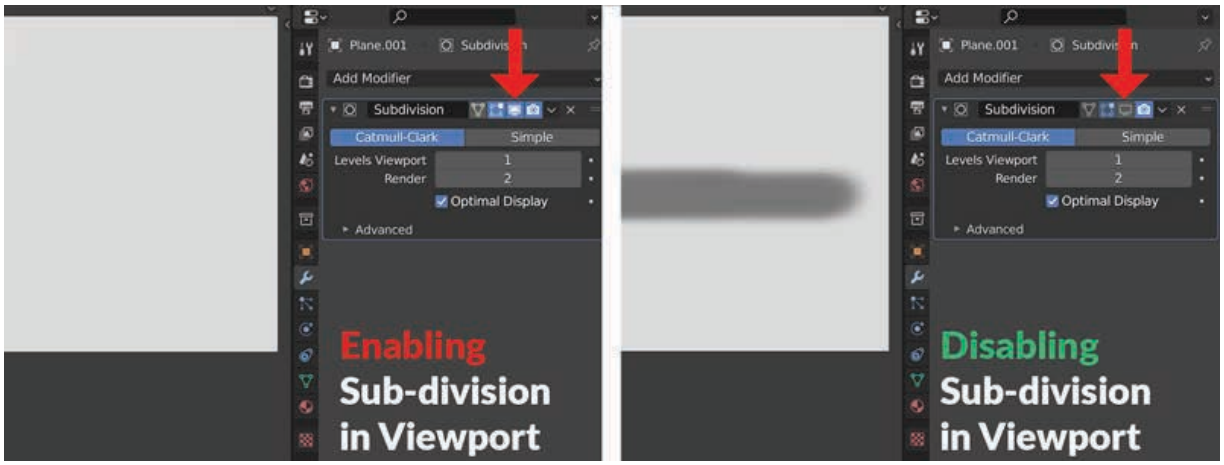
Cloth Brush Tips and Tricks

Previously we looked at the basics of using the Cloth Brush. Now, let's see if we can make the result look even more interesting. First, I recommend adding a **Subdiv modifier** on top.

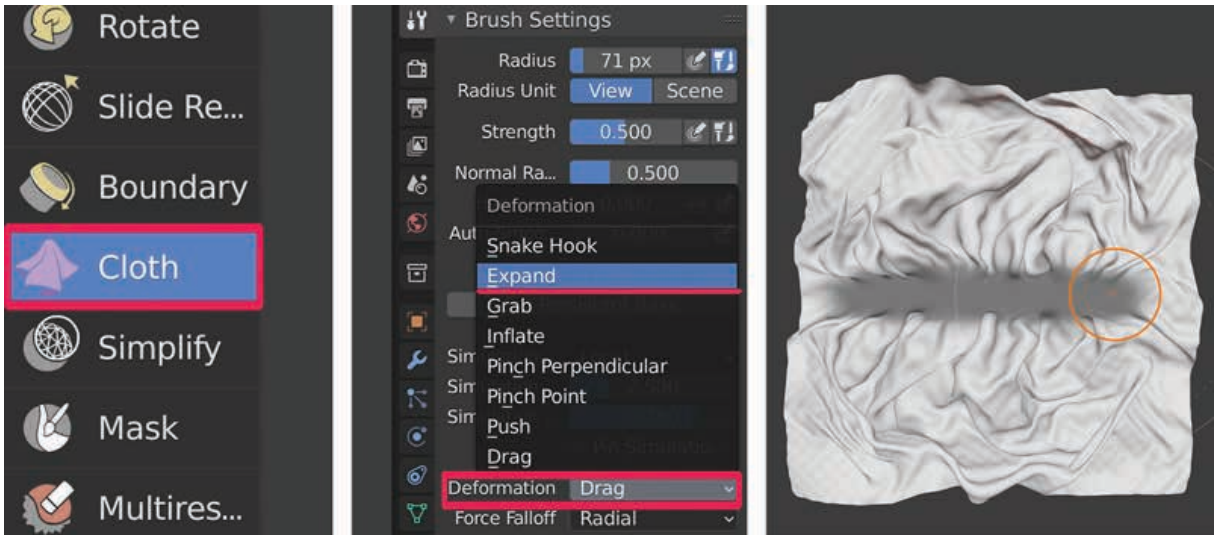
The **Mask brush** creates a Pinned area.



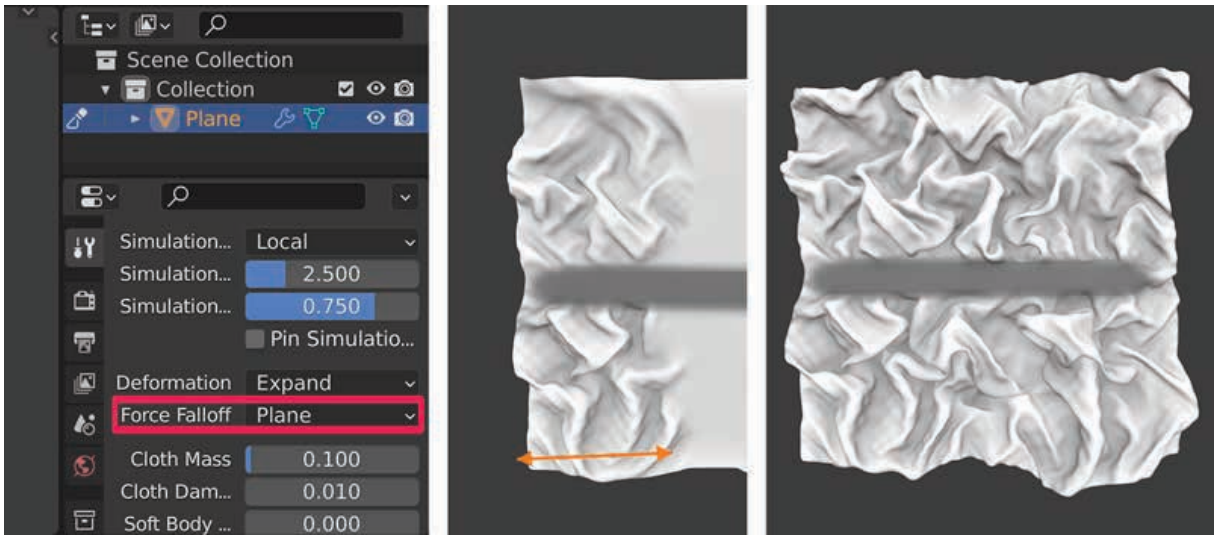
Keep in mind: Modifiers hide the mask. So turn them off for a moment.



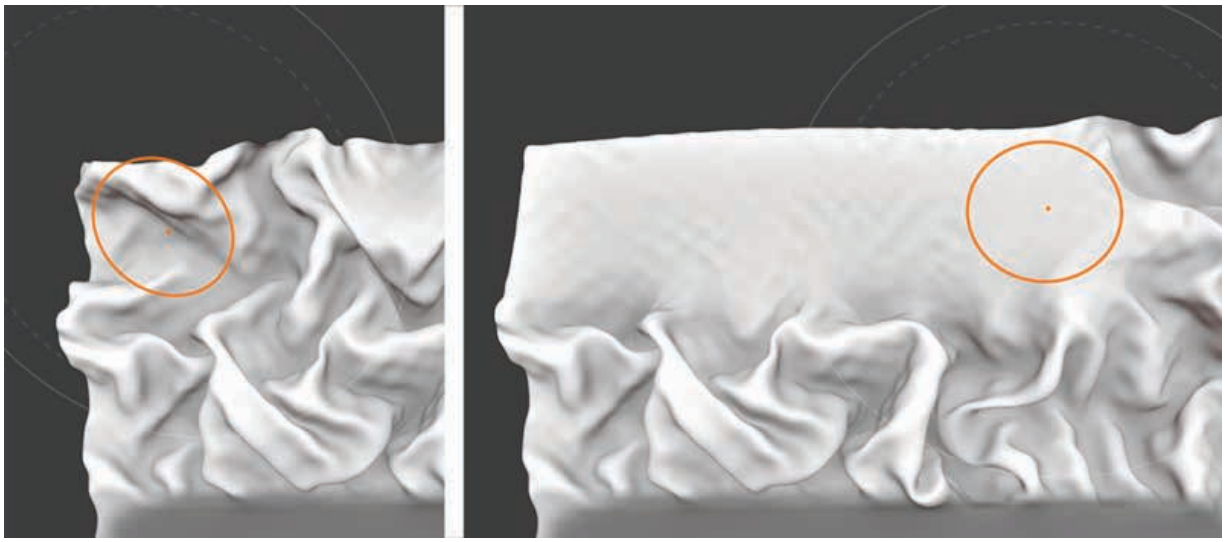
With the **Mask** and **Expand** function of the Cloth brush, you can do stuff like this.



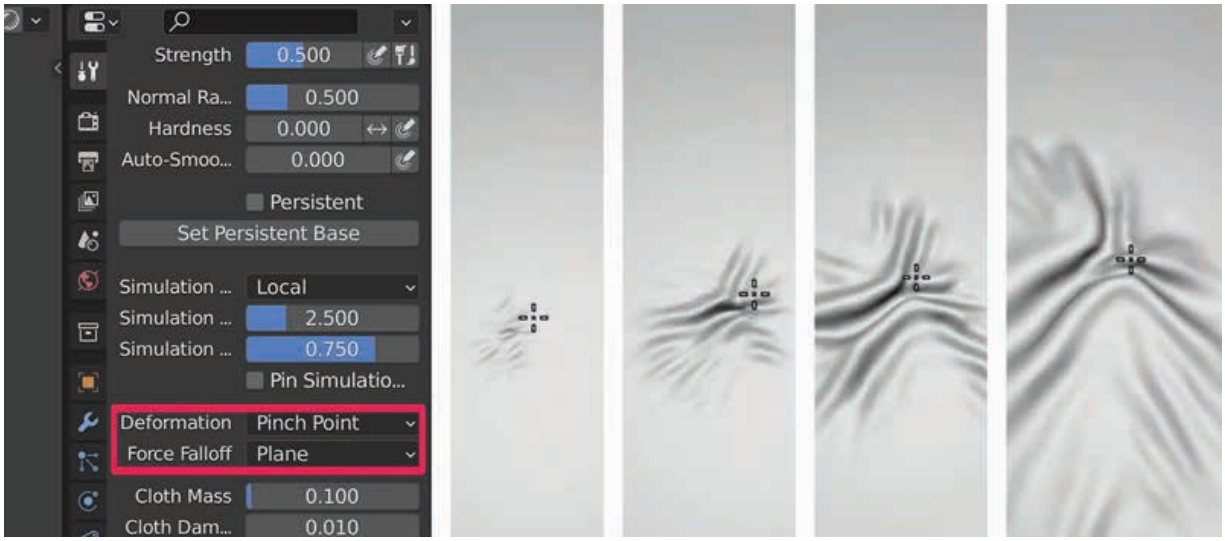
The **Plane Force Falloff** seems to get a better result in this case.



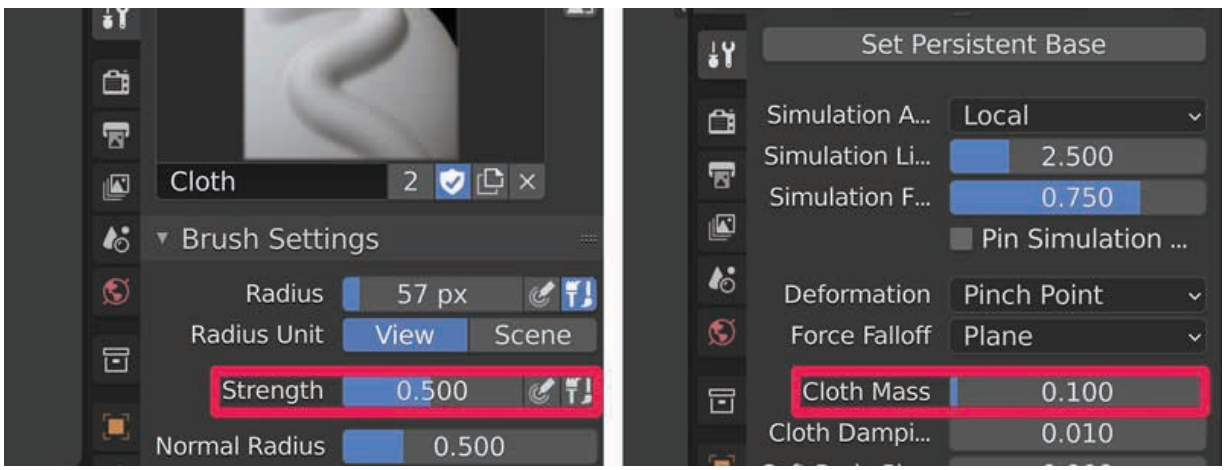
Holding Shift smooths out areas where needed.



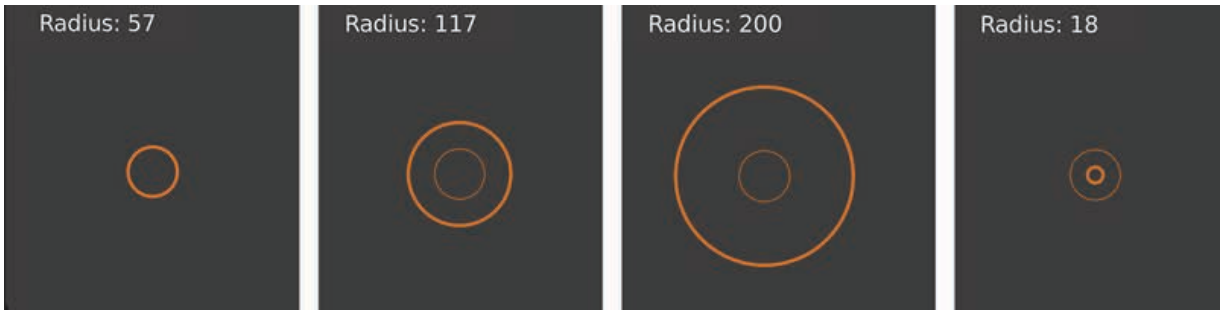
Pinch Point works well together with **Plane Force Falloff** to create this kind of pinched cloth effect.



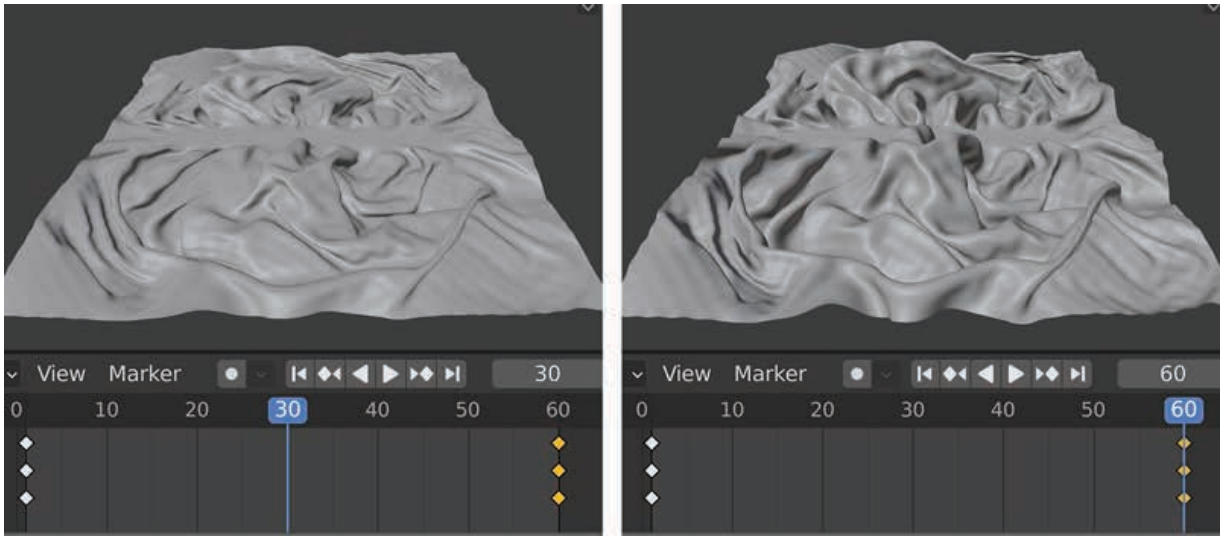
Set the **Cloth Mass** lower or the **Strength** higher. I like to use 0.1 for the Cloth Mass and 1 (*or less - 0.5) for the Strength.



Press F to change the Radius quickly.

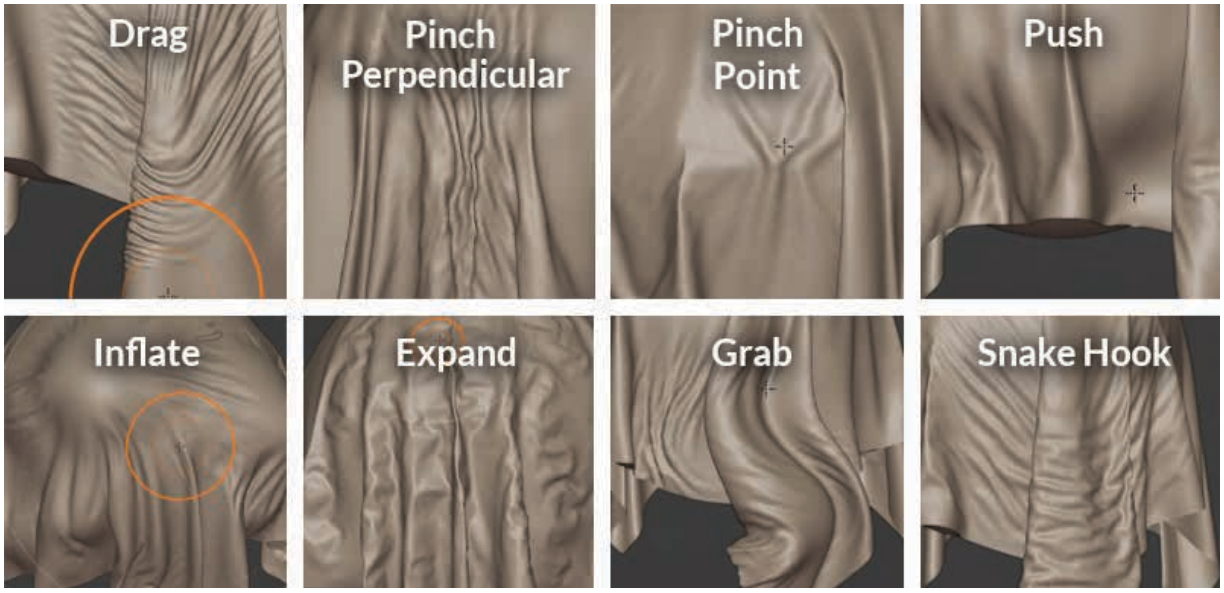


You can also use the Cloth Brush on a **Shape Key**, so you can kind of animate it.

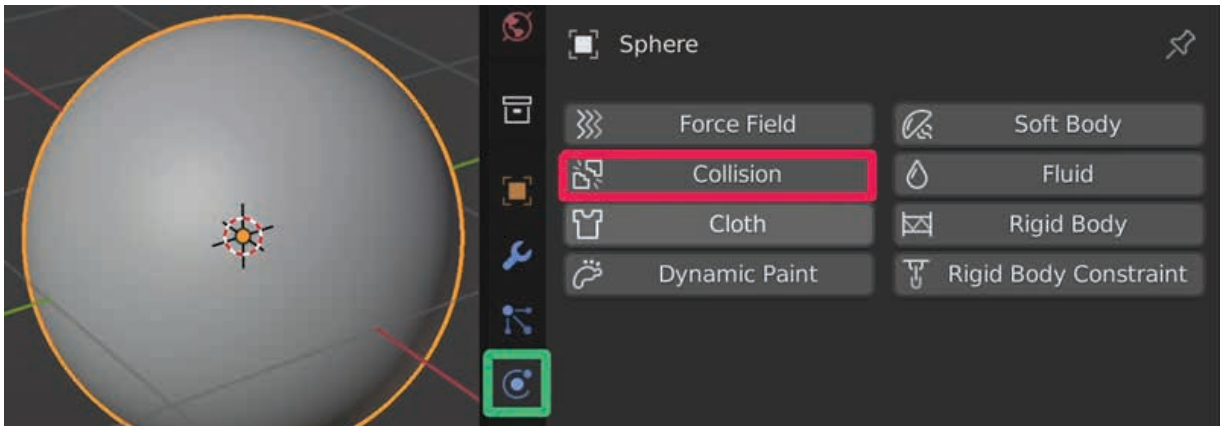


8 Types of Cloth Brush Deformation

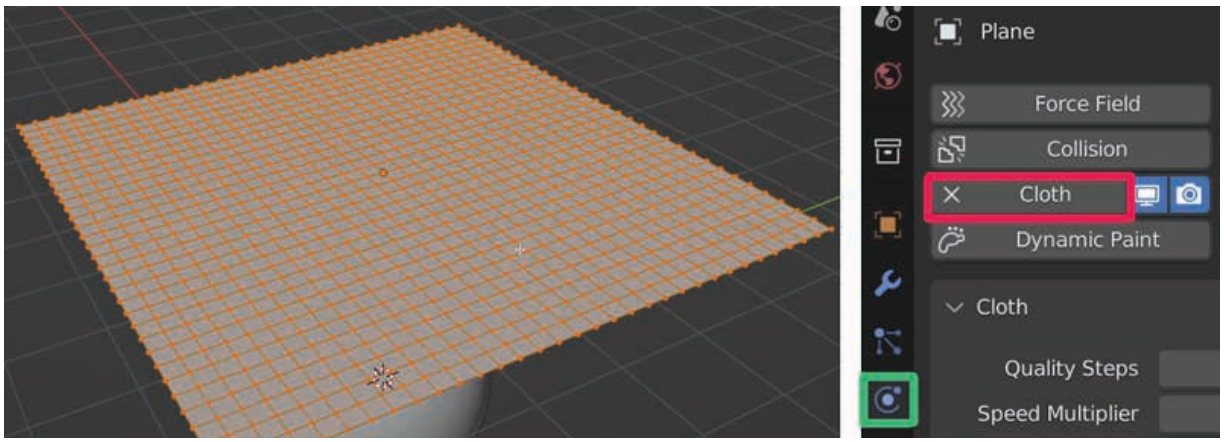
The **Cloth brush** is very useful for adding cloth simulation like details to a mesh. I recommend using it for small, secondary detail on a mesh that already has a real cloth simulation applied to it. Did you know there are 8 different cloth brush **Deformation** types?



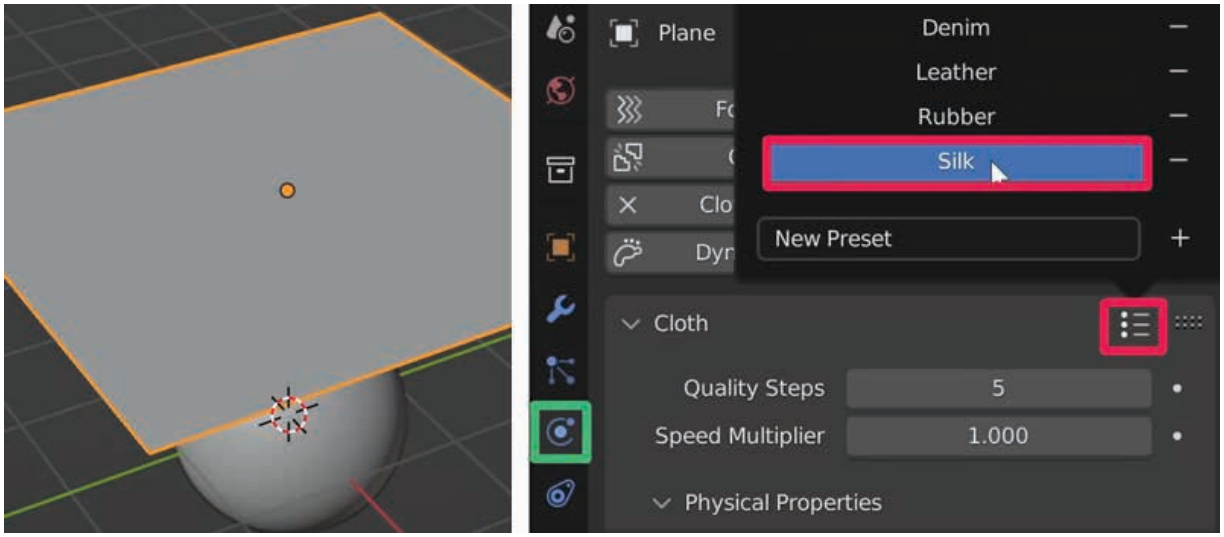
For this example to have some initial cloth folds to sculpt on top of, add a **Sphere** and enable **Collision** for it.



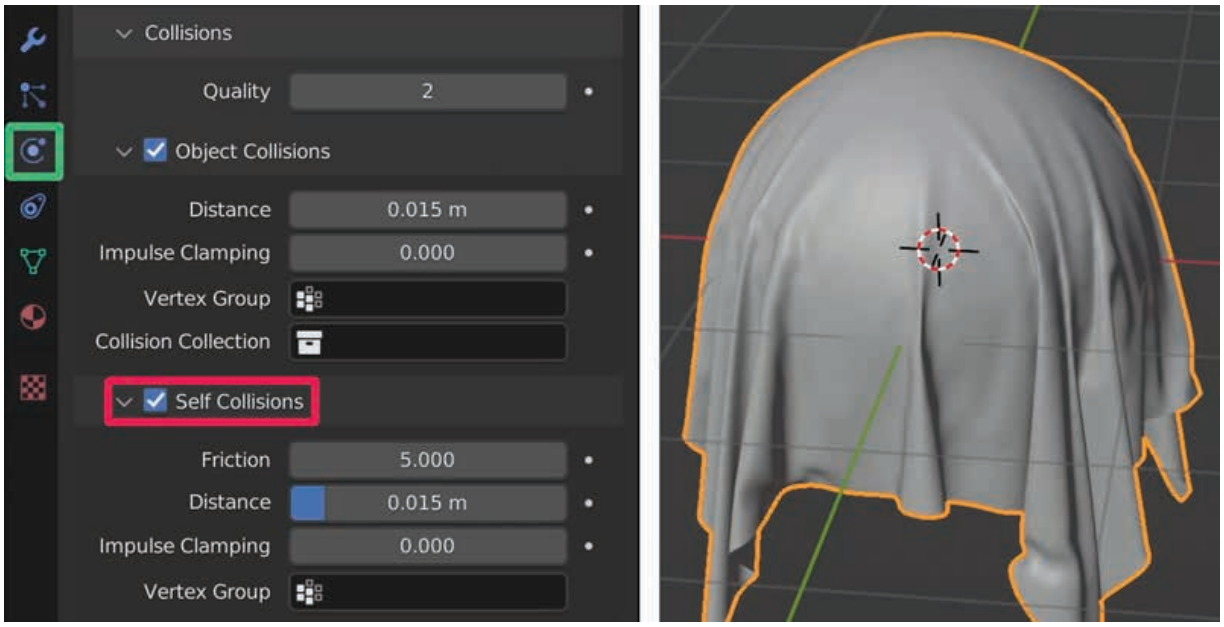
In Object Mode add a **Plane** and subdivide it a few times in Edit Mode so that it has enough geometry for cloth deformation. Add a **Cloth** sim to it.



You can choose different presets for the Cloth sim. In this case, I used **Silk**.

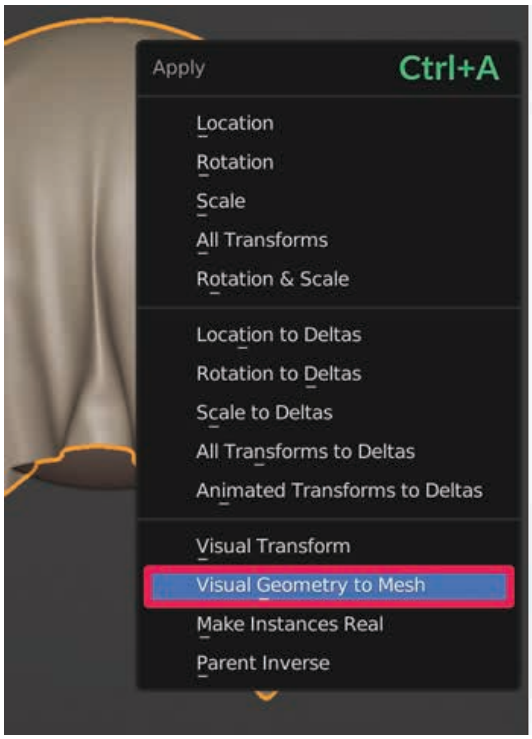


By enabling **Self Collisions** we get better looking results where the geometry doesn't intersect itself. **Press the Spacebar** to play the simulation (or press Bake in the Cache options).



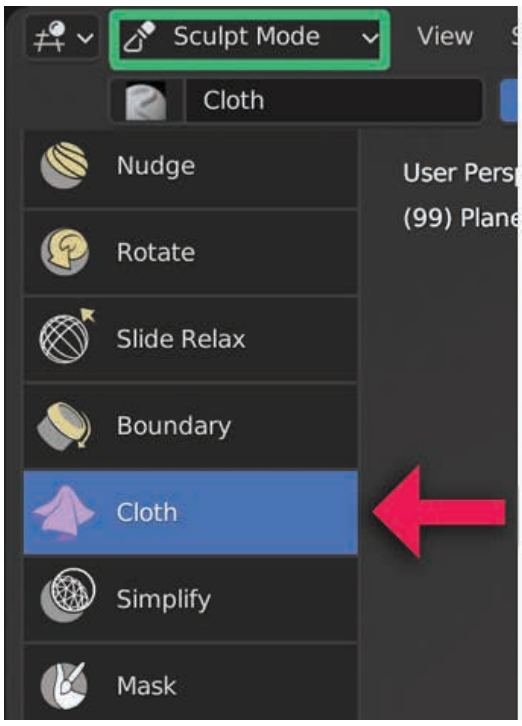
After running the Cloth sim, we can choose a frame on the Timeline where it looks okay and then **apply** the Cloth simulation so it's just a normal mesh we can sculpt on. If you want you can add some extra subdivision with a Subdiv modifier, and apply that as well before sculpting with the Cloth brush.

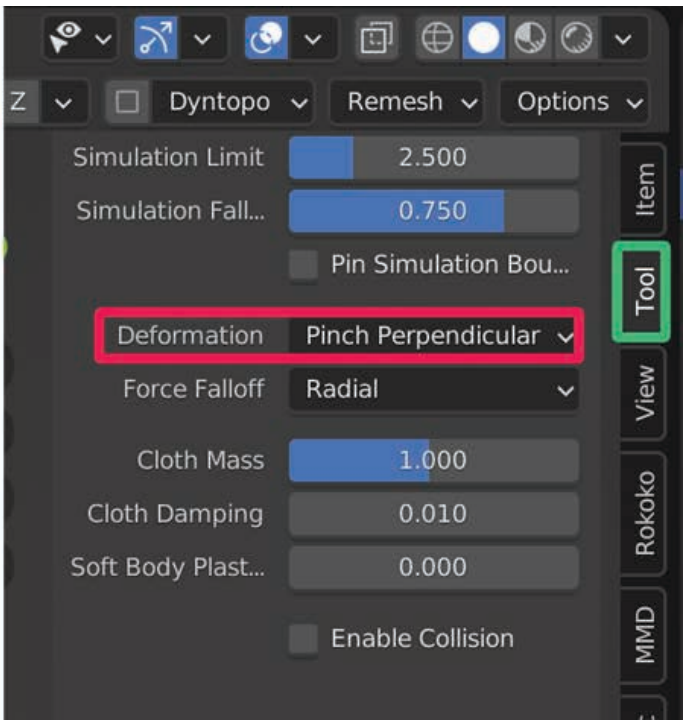
I find that the Cloth brush works best if there's not too many modifiers on the mesh. You can apply all modifiers at once by **pressing Ctrl+A** in Object Mode and choosing **Visual Geometry to Mesh**.



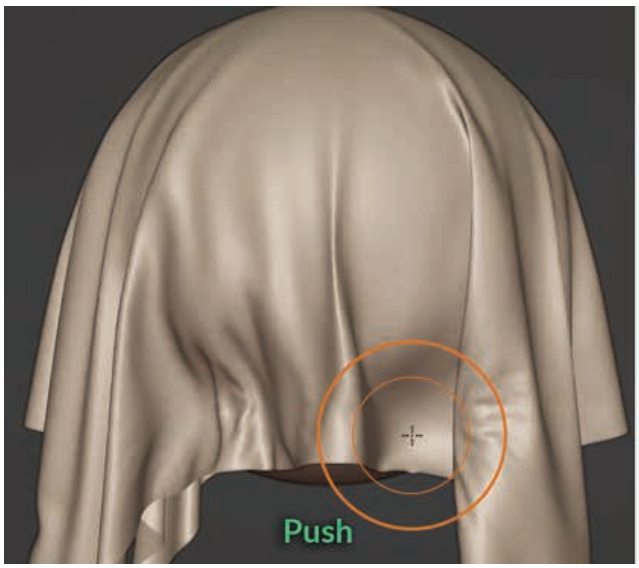
Go to **Sculpt Mode**, and open the **Tool Panel** (press T) and drag it out so you can see the names (I always find all those icons confusing personally). Select the **Cloth Brush**. Then open the **Option Panel** as well (**press N**).

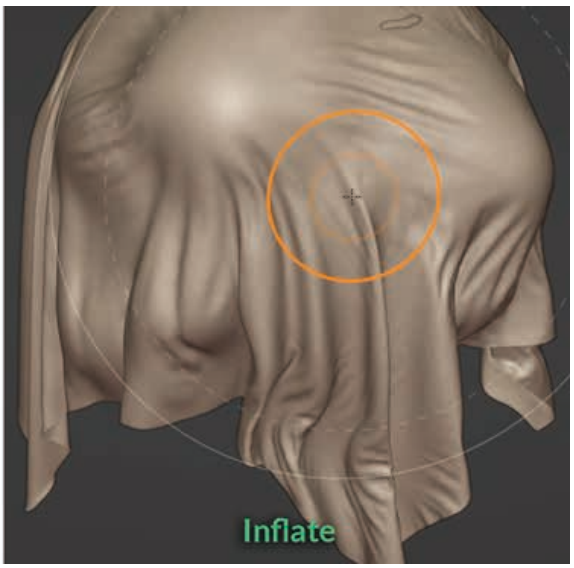
In the **Tool tab** of the Option panel you can see the options of the brush. From the **Deformation** drop-down, you can access 8 types of Cloth Brush. Let's look at what they all do below.



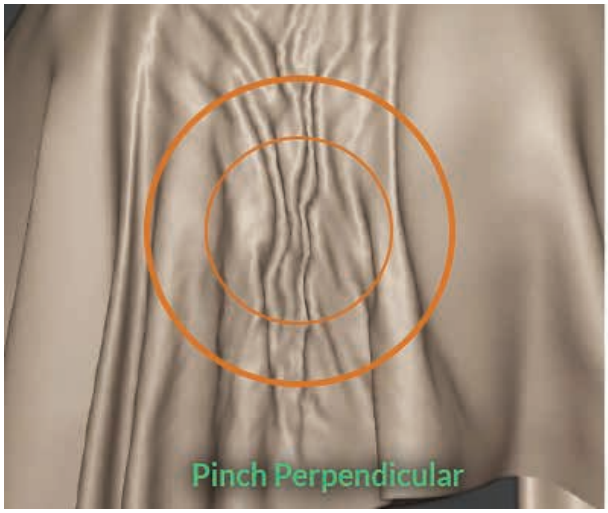


Push will push the cloth down, which can be useful for conforming it more closely to a shape underneath. **Inflate** does almost the opposite, and also adds some nice wrinkly folds around the edges of the inflated area. Both brushes can be inverted by **holding Ctrl**.

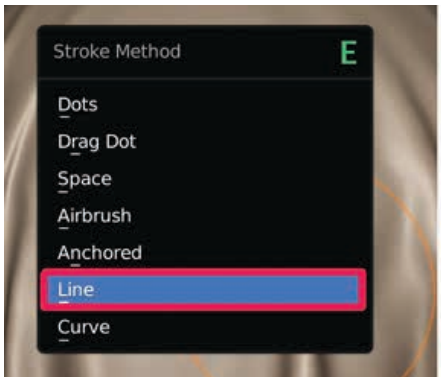




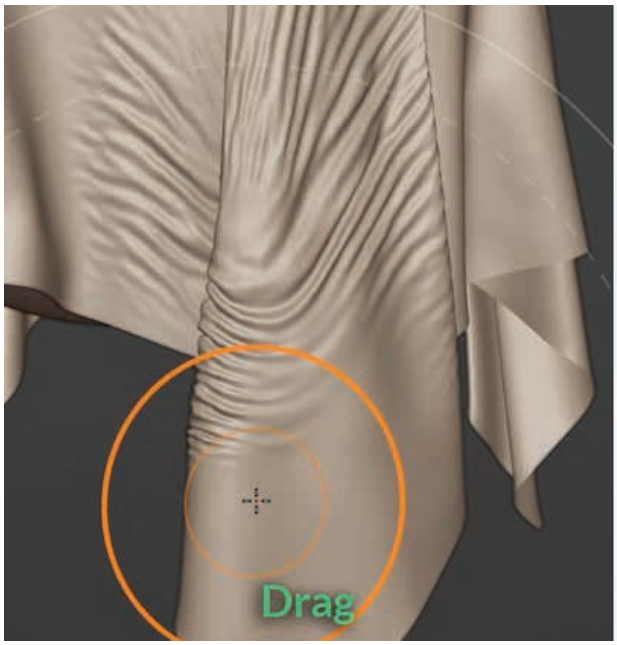
Pinch Perpendicular will pinch cloth towards the area where you brush, and can be used to make folds more pinched and wrinkly. Best used carefully with a low Strength. **Pinch Point** pinches the surrounding geometry to where your brush is.

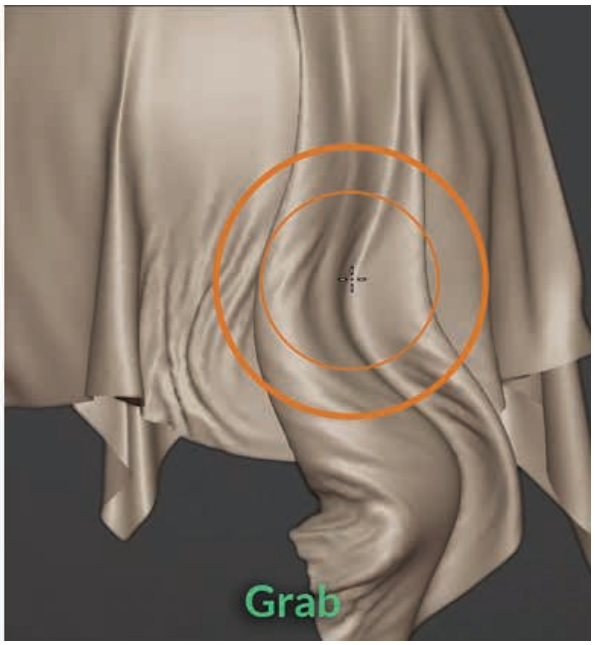


Pinch Perpendicular works best with **low Strength** and the **Line stroke method**. You can change the stroke method by **pressing Alt+E** in Sculpt Mode.



Drag will drag the geometry in the direction of your brush strokes, while creating nice folds. **Grab** does a similar thing, but takes the geometry itself with it as well. Quite similar to the regular Grab brush in Sculpt mode, except that it also adds folds of course.

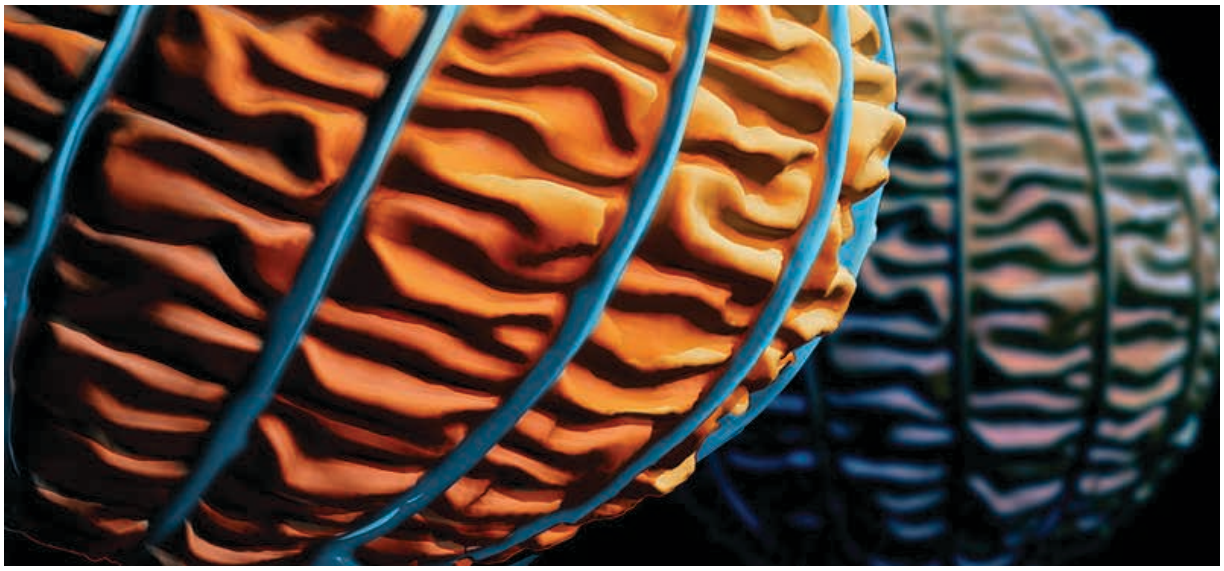




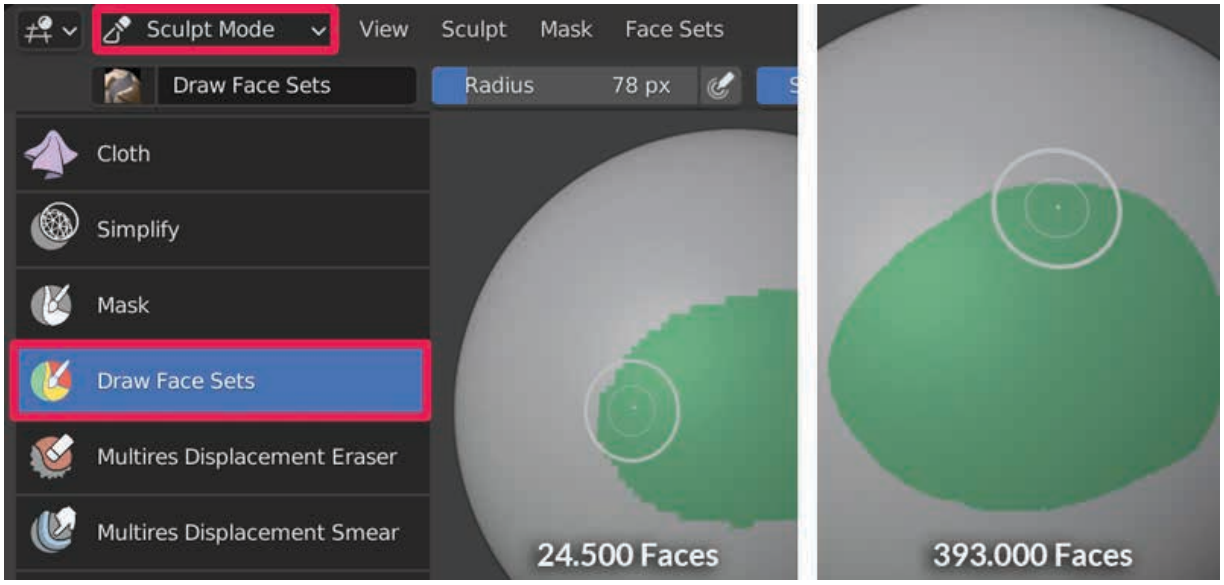
Expand stretches the geometry into random folds and wrinkles where you place brush strokes. It can be really nice to quickly add secondary detail to existing cloth folds. Finally, **Snake Hook** is similar to Drag but simulates the cloth more accurately. However this comes with a noticeably slower performance.

Cloth Sculpting with Masks, Face Sets and the Cloth Filter

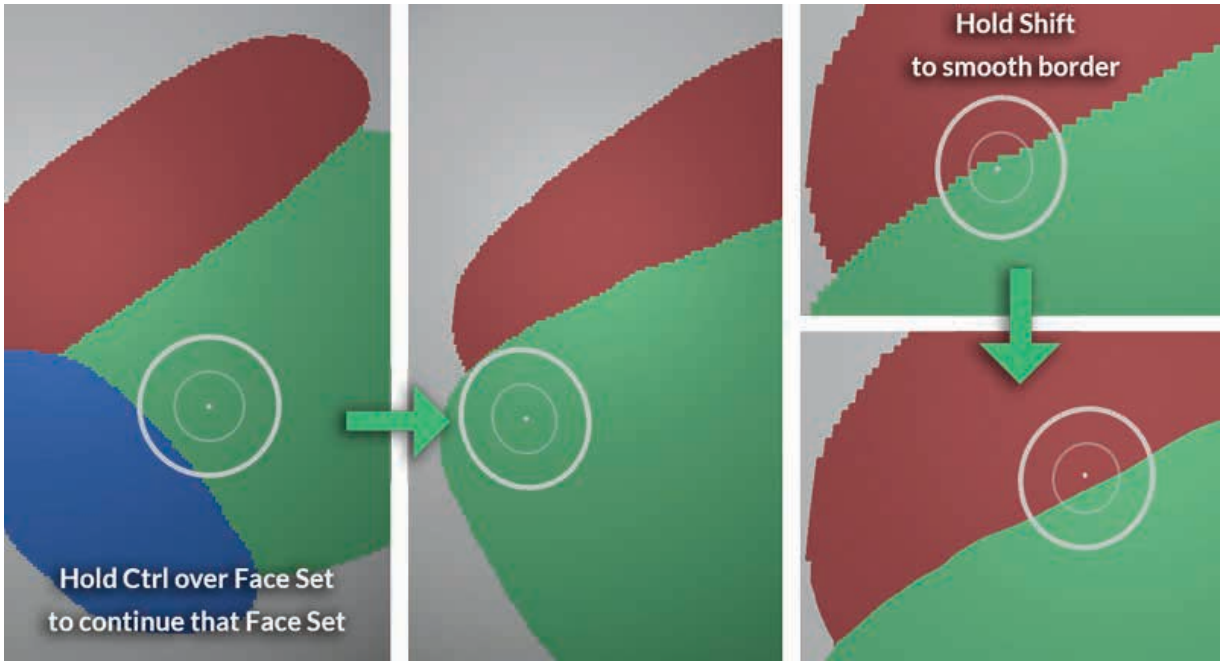
Previously we've looked at the eight different Cloth Brush deformation types. But that's not all you can do to sculpt interesting clothes. Face Sets and masks are useful tools for cloth sculpting.



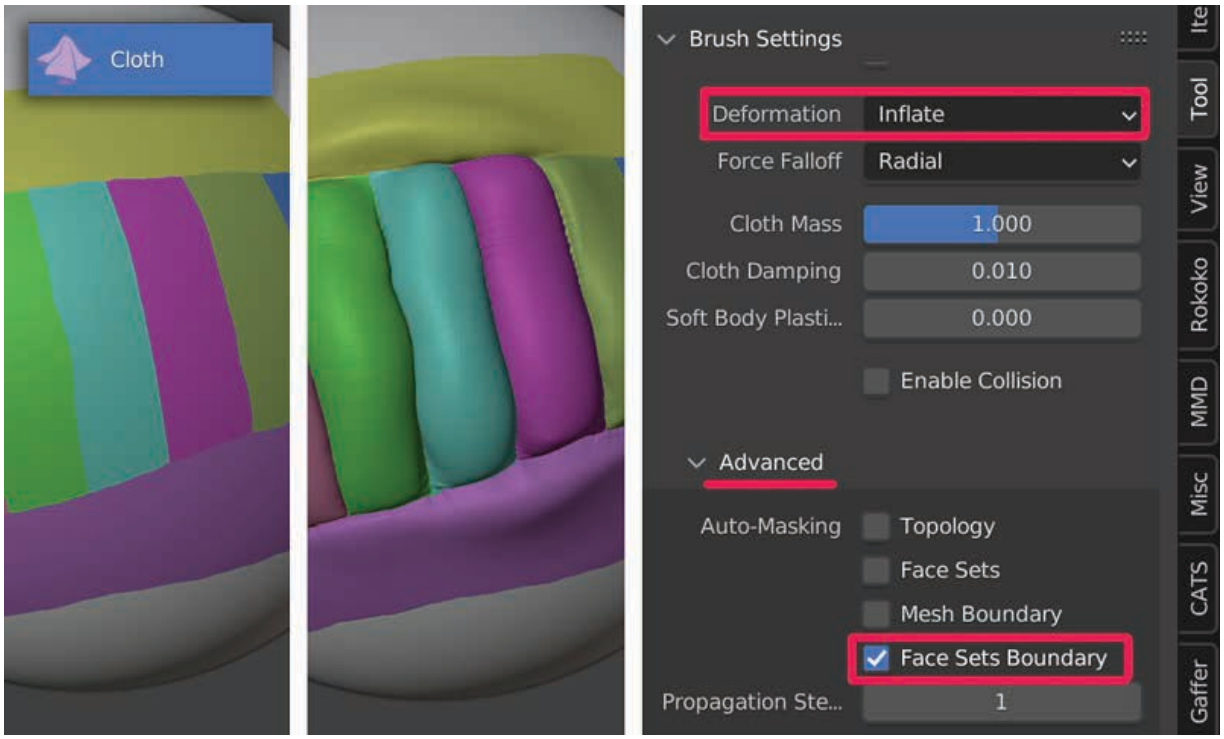
In Sculpt Mode, using the **Draw Face Sets brush** you can paint Face Sets on the model. How detailed these are depends on how many Faces the geometry has.



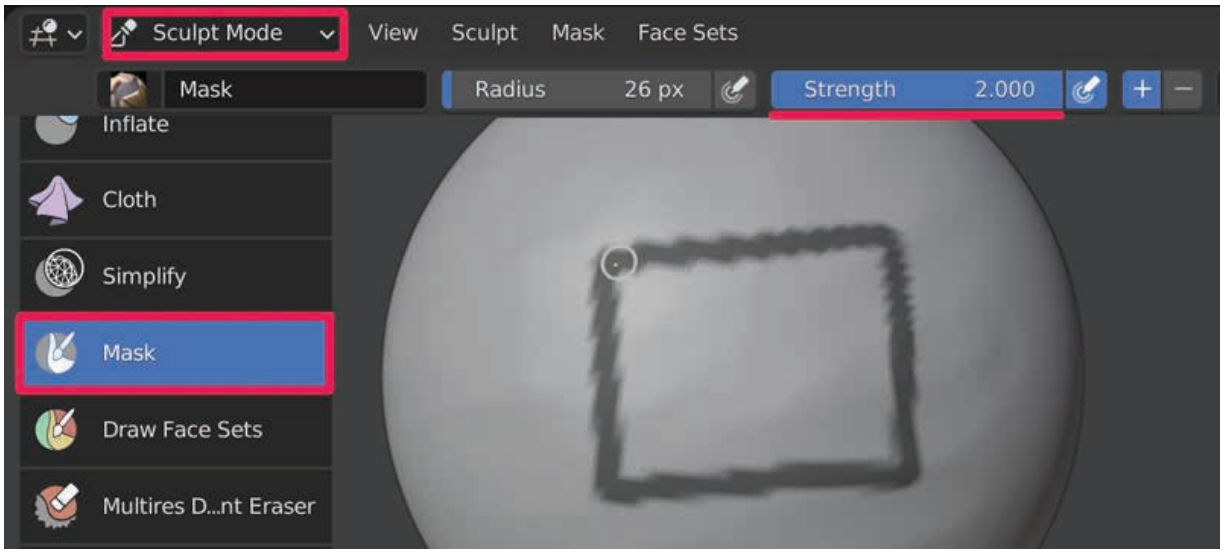
Every time you draw, you'll create a new Face Set, unless you hold Ctrl while mouse hovering over the Face Set you want to add to. Hold Shift to smooth the borders.



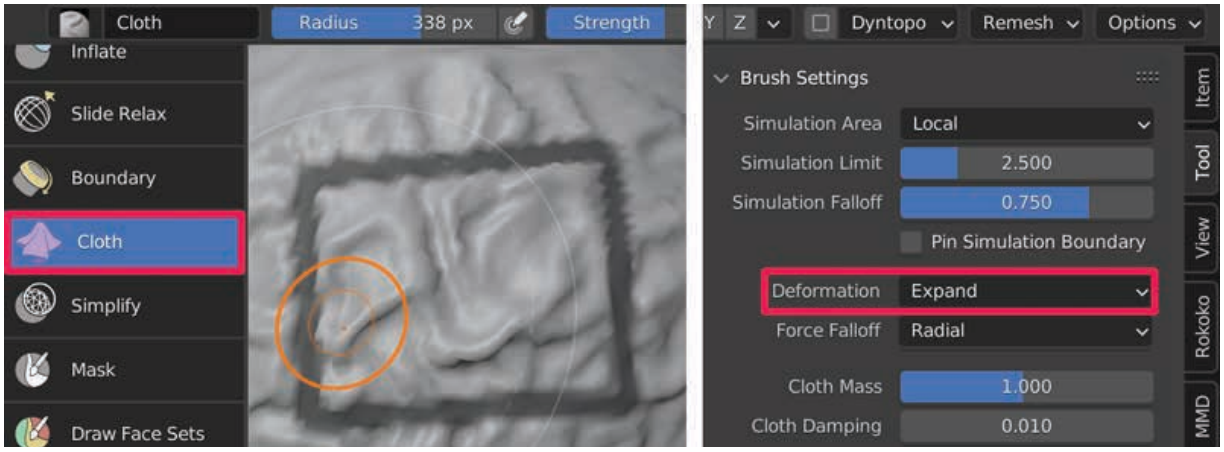
Choose the Cloth brush with a Deformation type, in this case I think Inflate gives an interesting result. In the Option panel (shortcut: o) in the Tool panel under Brush Settings > Advanced, enable Face Sets Boundary Auto-Masking to use the Face Set borders as areas not affected by the Cloth brush when you brush over those areas.



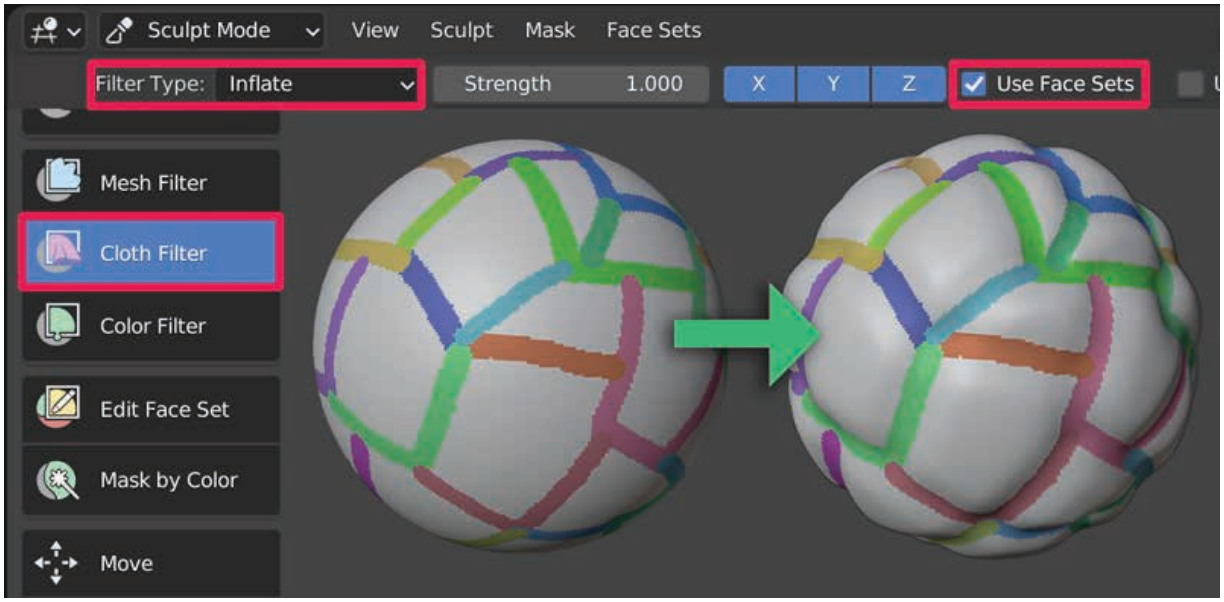
Similarly, you can use the **Mask brush** to paint areas not affected by the Cloth brush.



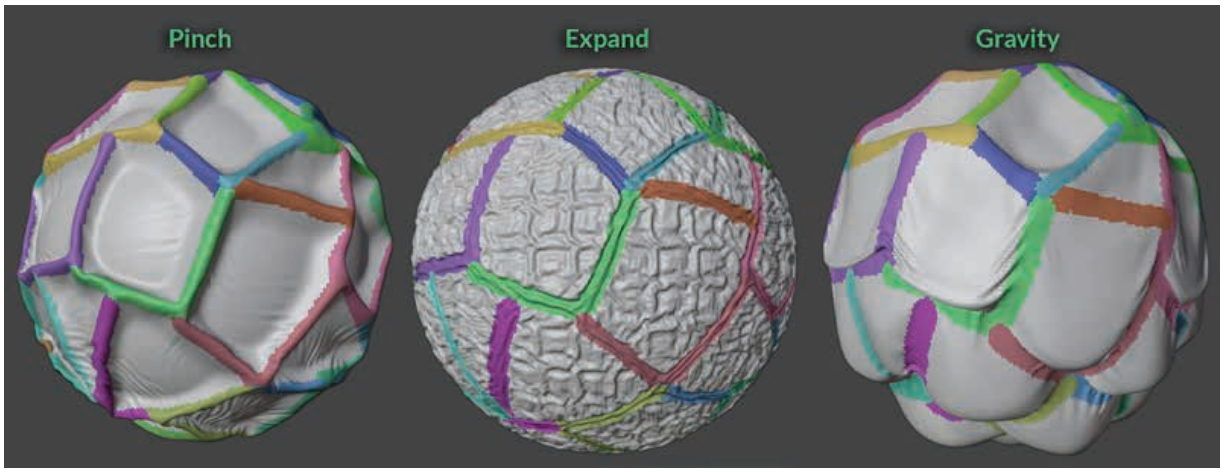
This creates interesting effects for example in combination with the **Expand deformation type**.



Using the **Cloth filter**, we can also add a deformation to the entire mesh at once. Choose a **Filter Type** (for example **Inflate**), check **Use Face Sets** and drag the mouse to the right.



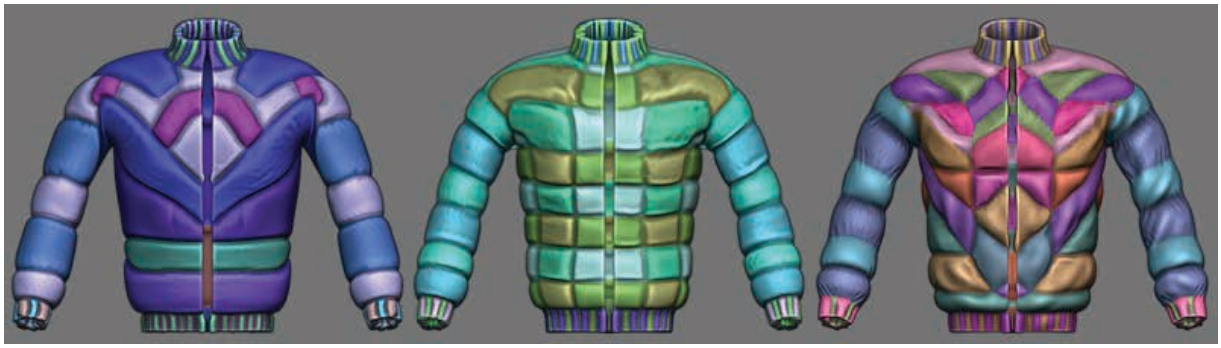
Here is a demonstration of other filter types:



Making a Padded Jacket using the Expand feature

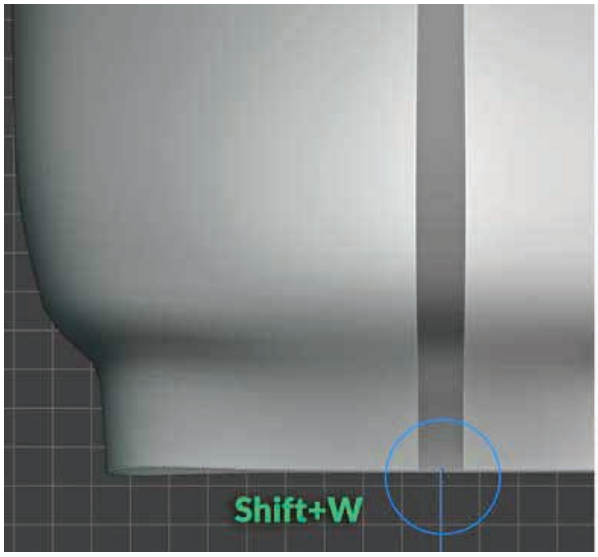
The **Expand** feature in Sculpt Mode lets you quickly create both masks and Face Sets with different Falloff shapes. It has a lot of interesting options, like creating loops and recursions, that make it possible to quickly create complicated shapes.

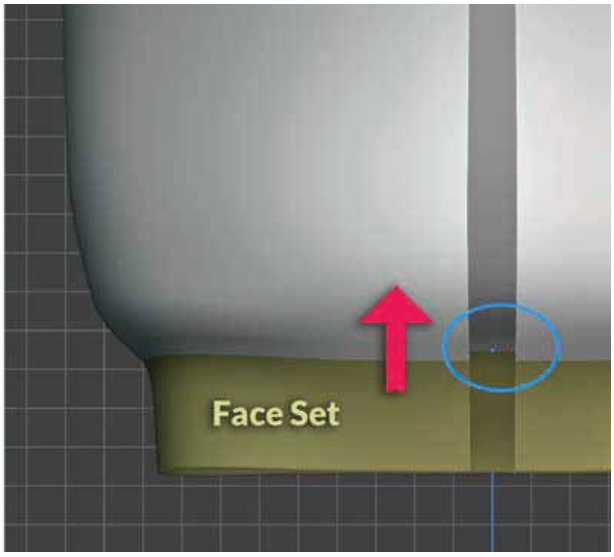
In this example, we'll use these options to create a thick padded jacket (a "down jacket" or "puffer jacket"). I used Blender 3.5 for this topic.



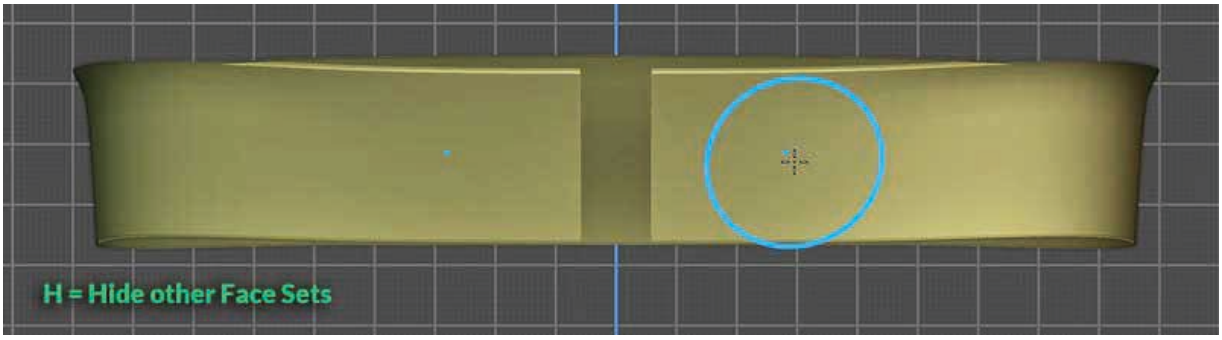
Normally the way Expand works for Face Sets, is that you **press Shift+W** to draw a Face Set (starting from the mouse cursor location). In that case it will expand outwards from that point. But you can also start from just the very border, or just outside of the border, of a mesh. That's how we'll do it in this case.

Press Shift+W just outside the edge of a mesh and drag to draw a Face Set.

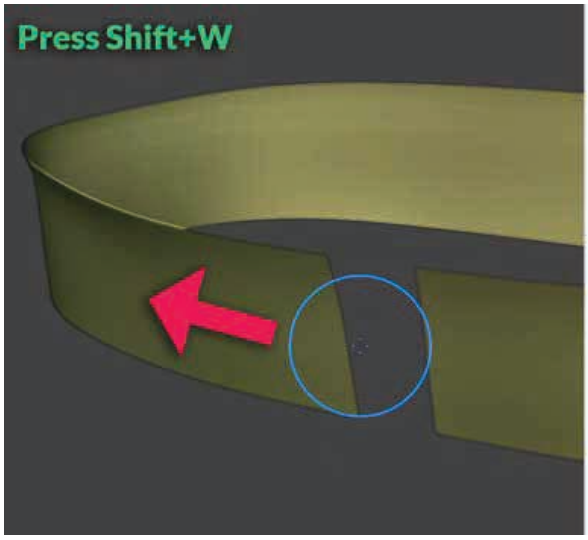


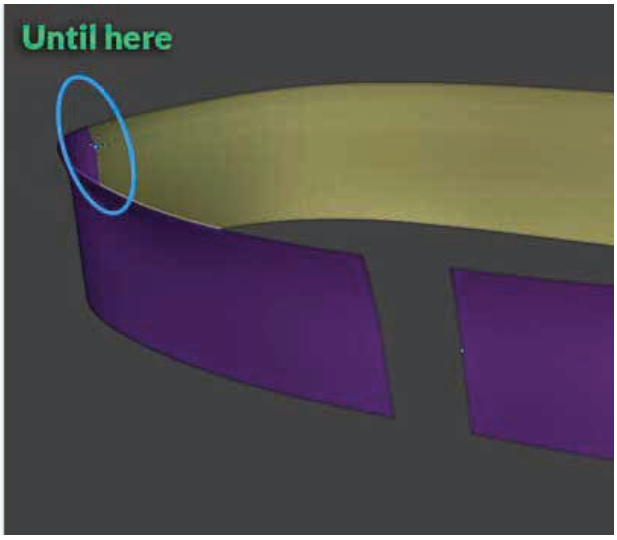


Hover the cursor over the Face Set and **press H** to hide everything else.

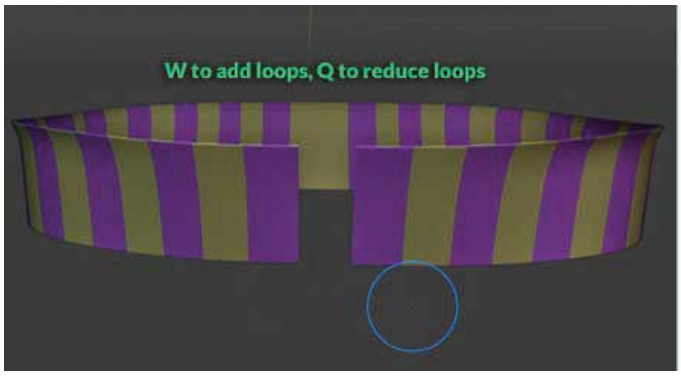


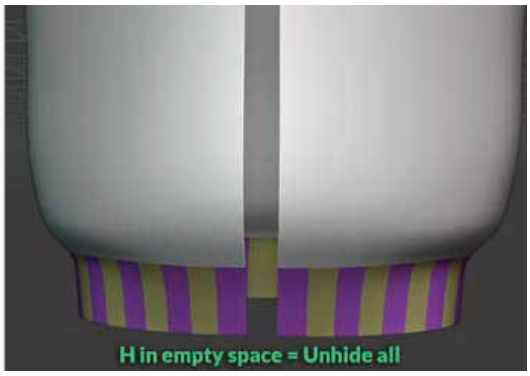
Add another Face Set, this time by **pressing Shift+W** from the left or right side. Drag it to about half way around.



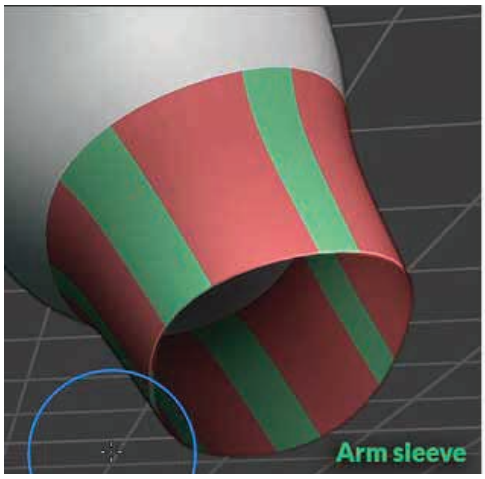


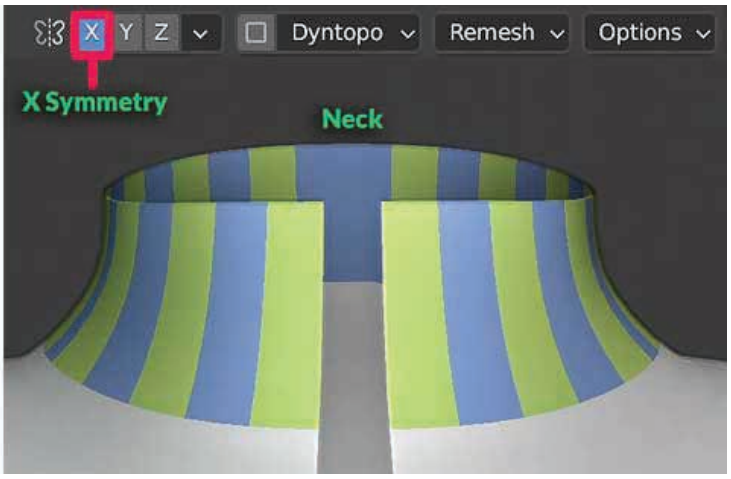
You can **press W or Q** to add or remove **loops**. You'll need to experiment to find the right spot to start with the loops. Undo by pressing Ctrl+Z and just try again, until you get the right distance for nicely spaced out loops. **Press H** in empty space to unhide all.



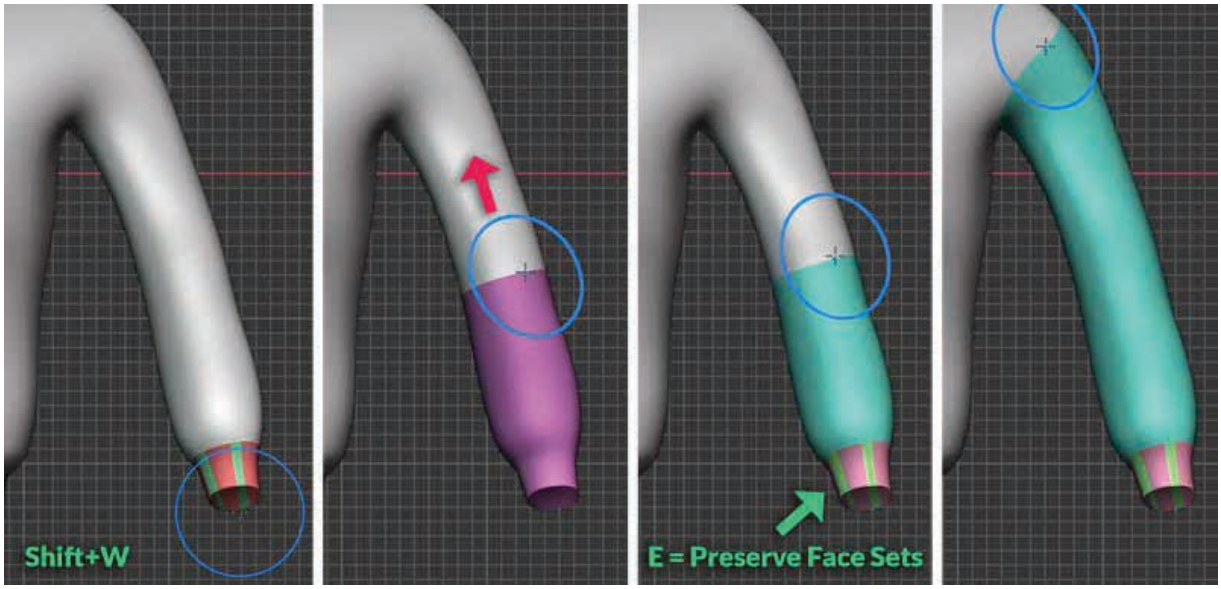


Repeat this process for the arm sleeves - remember to turn on **X Symmetry** so both arm sleeves have the same Face Sets - and the neck.

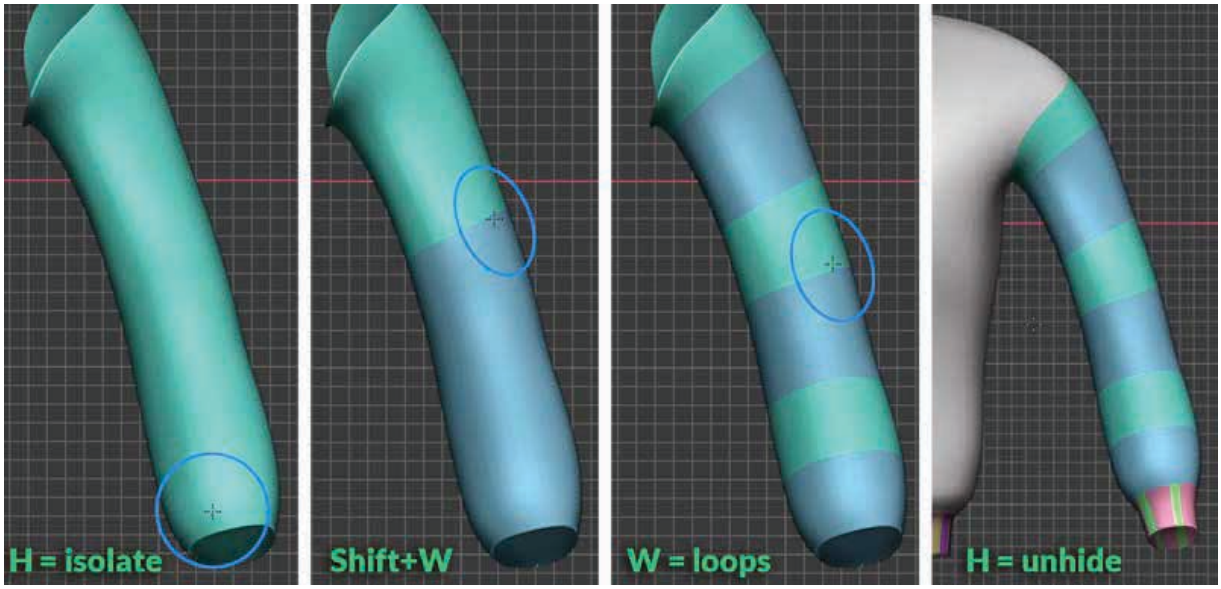




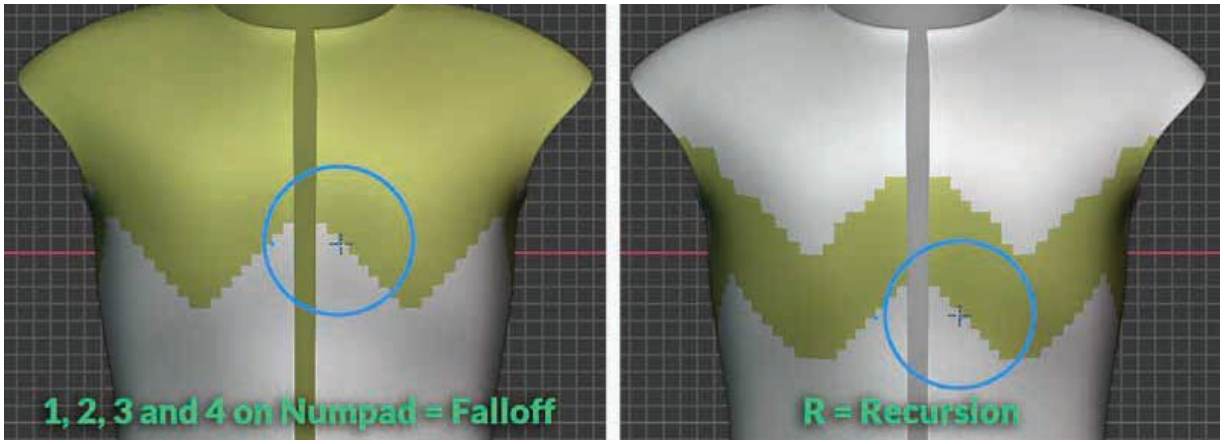
After drawing another Face Set for the entire arm, **Pressing E** will preserve the existing Face Sets, rather than overwriting it.



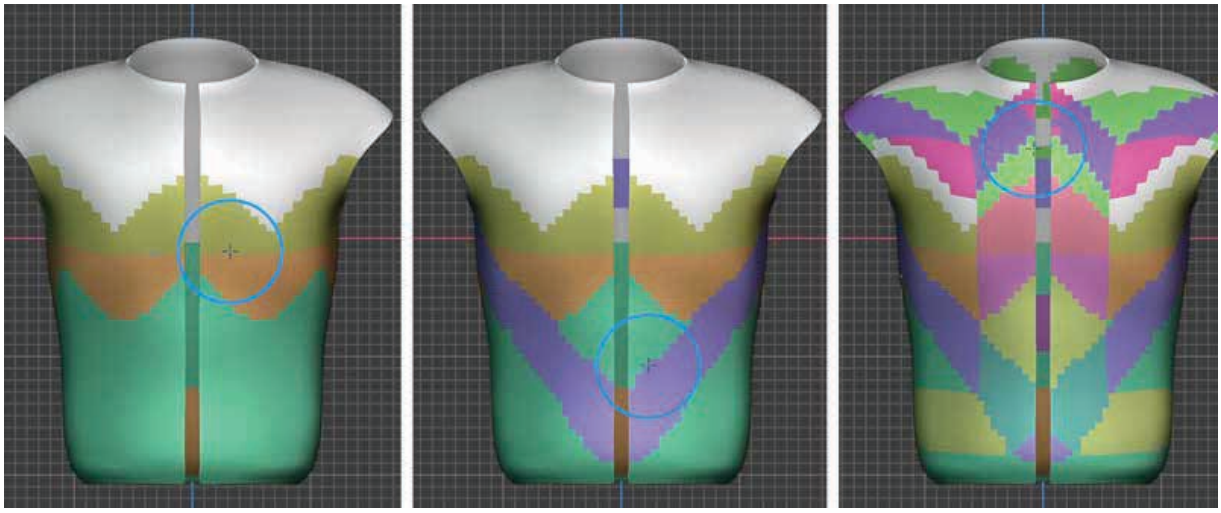
Add some loops to the arm sleeve by isolating it by pressing H while hovering over the arm, then create a new Face Set with Shift+W. Then **press W** a few times to add loops, and H in empty space to unhide everything again.



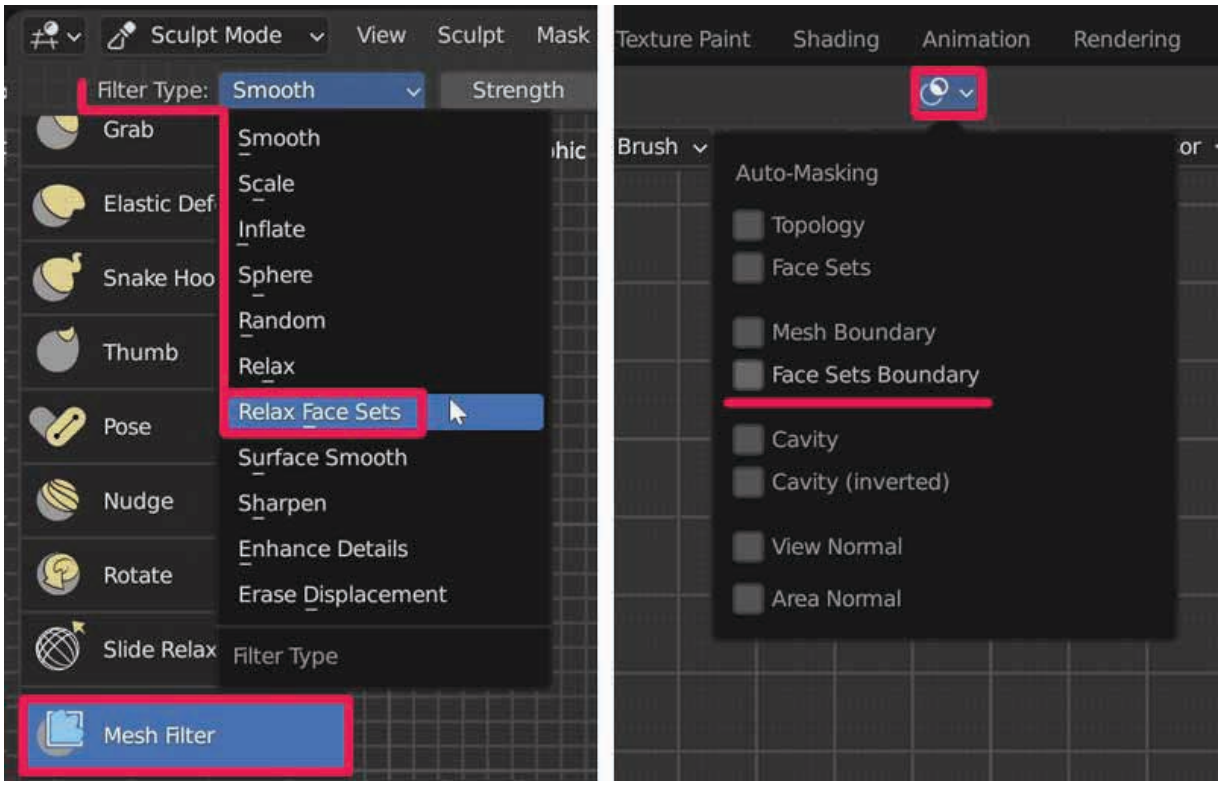
Pressing 1 to 4 on the Numpad gives you different types of **Falloff**. Start a **Recursion** from the border of the Face Set you're drawing, by **pressing R**.



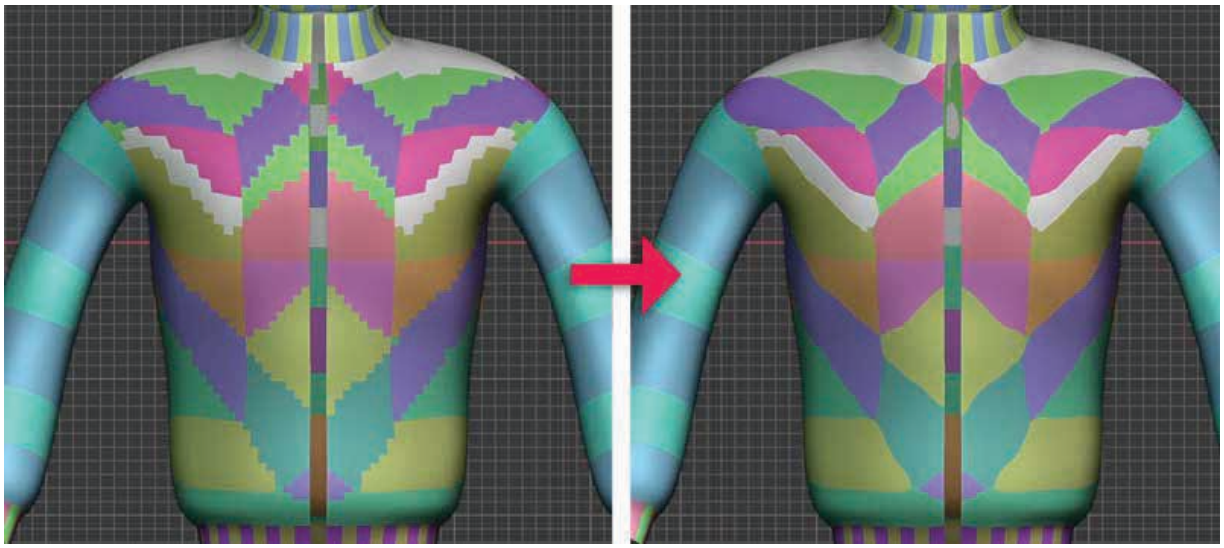
It's fun to try different Falloff types (1, 2, 3, 4 on the Numpad) in combination with Recursion (R). A lot of this is just experimentation, trying things and pressing Ctrl+Z to undo if you don't like it. And **press E** to combine these Face masks rather than overwriting them.



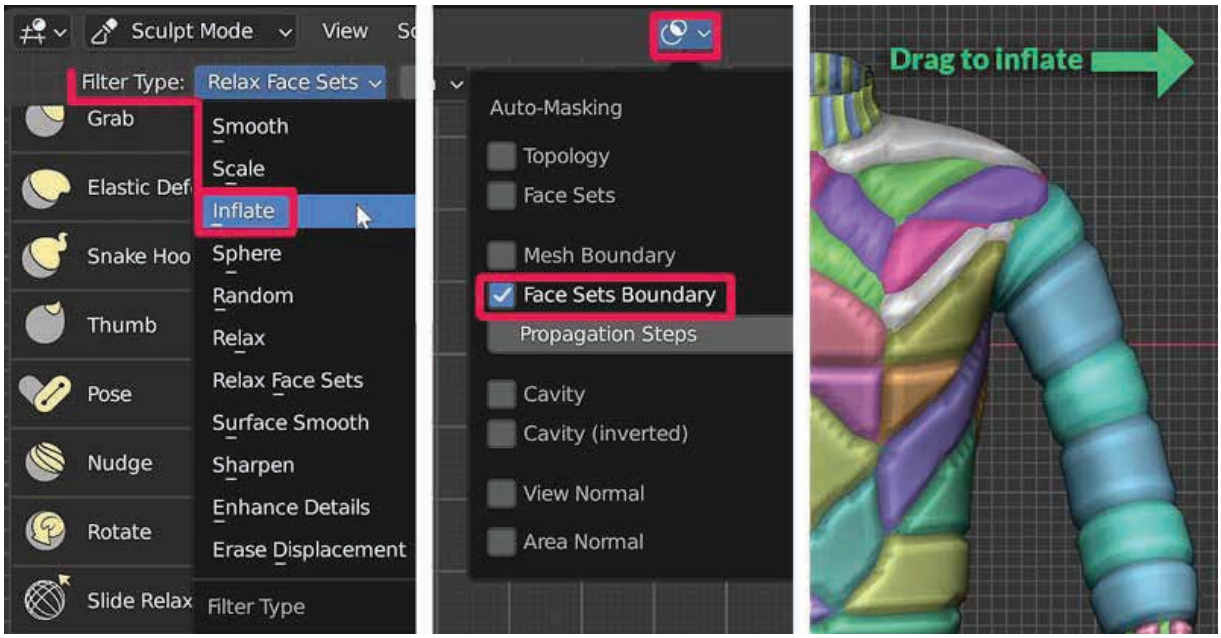
Set the **Mesh Filter** to **Relax Face Sets** and drag to relax the Face Set borders. If nothing happens, make sure the Face Sets Boundary Auto Masking option isn't enabled.



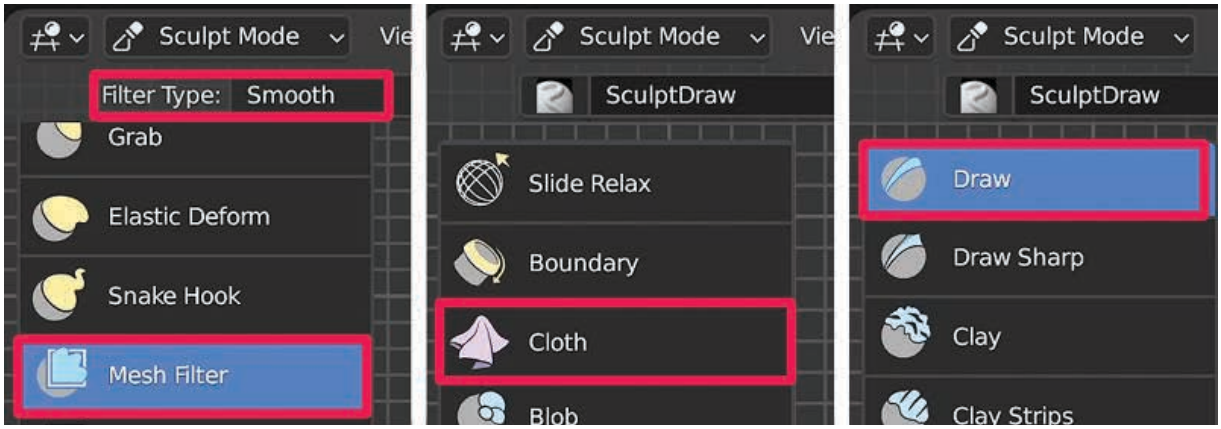
After using the Relax Face Sets Mesh filter, the Face set borders will be smoother.



Now enable **Auto Masking**, set to **Face Sets Boundary**, and use the **Mesh Filter** set to **Inflate**. This Auto Masking feature will make all the individual Face Sets inflate, instead of the entire mesh all at once.

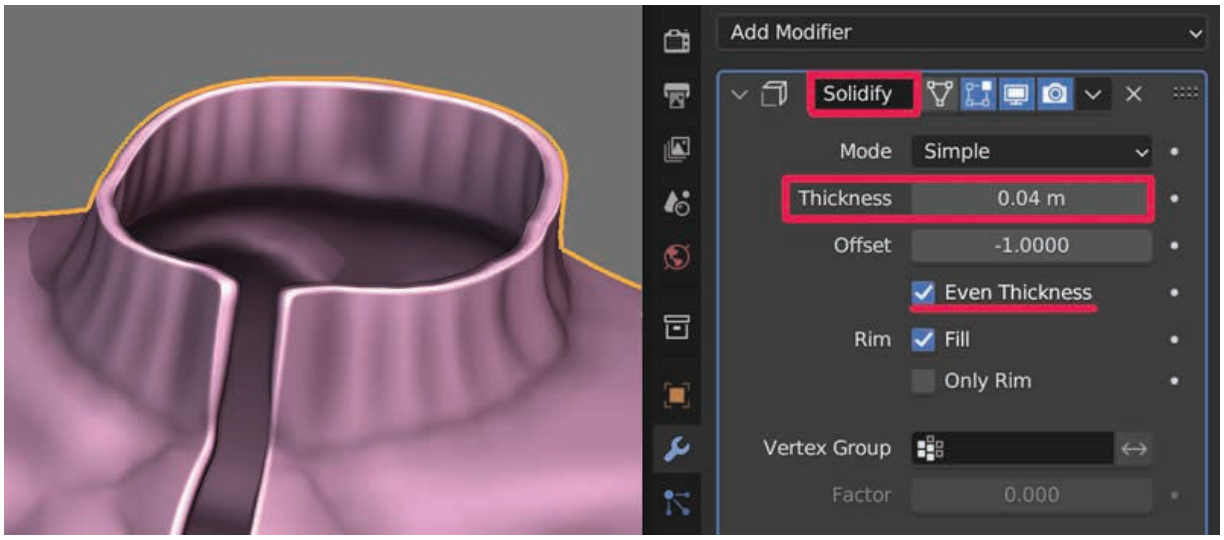


Then use the **Smooth** Mesh Filter as well. Finally improve it with the **Cloth brush**. You can use the various deformation types as explained in this video. The good old **Draw brush** is also a great tool for adding more Cloth-like shapes.



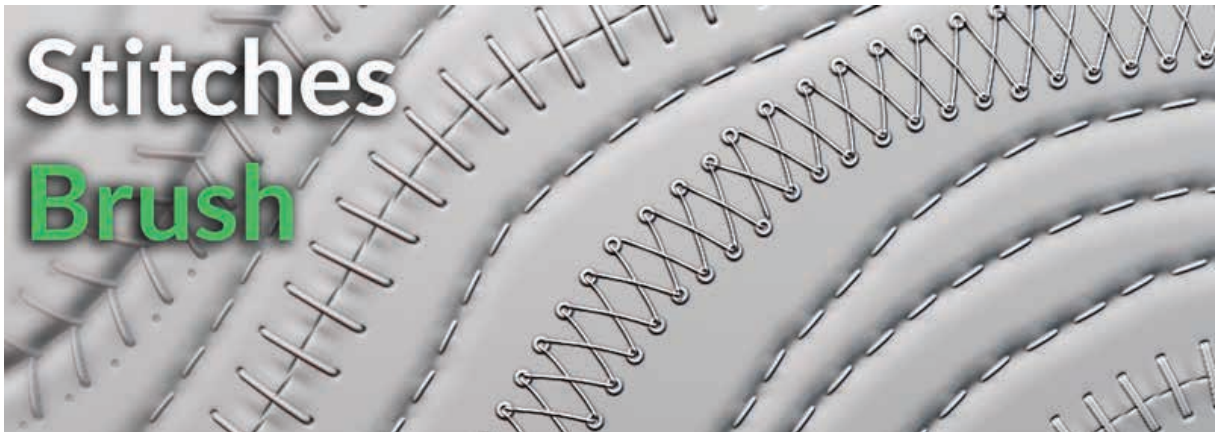
Click here to return to the index.

A Solidify modifier with some subtle Thickness will make the jacket more realistic.

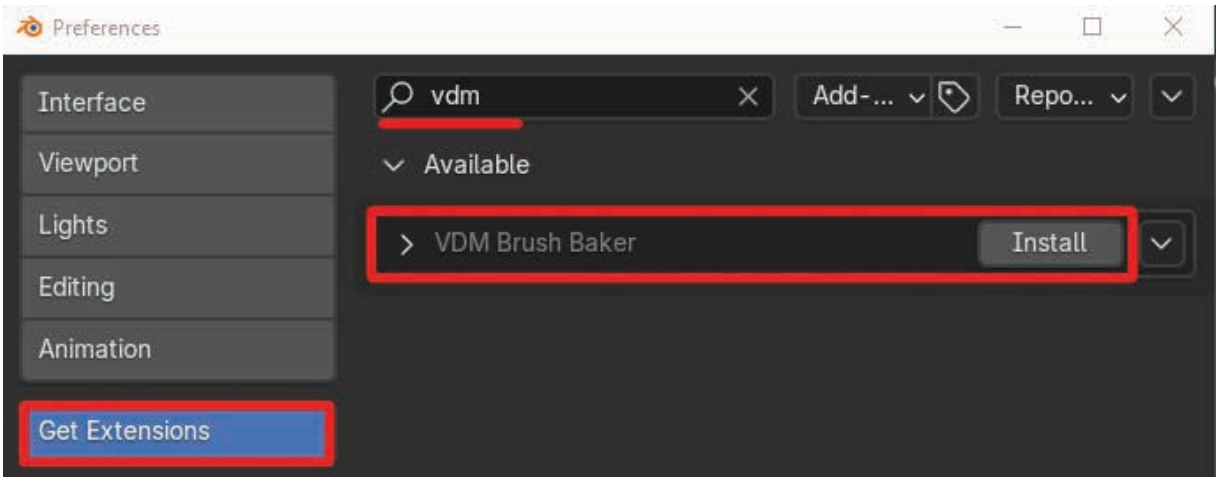


Create a Stitches sculpting brush in Blender

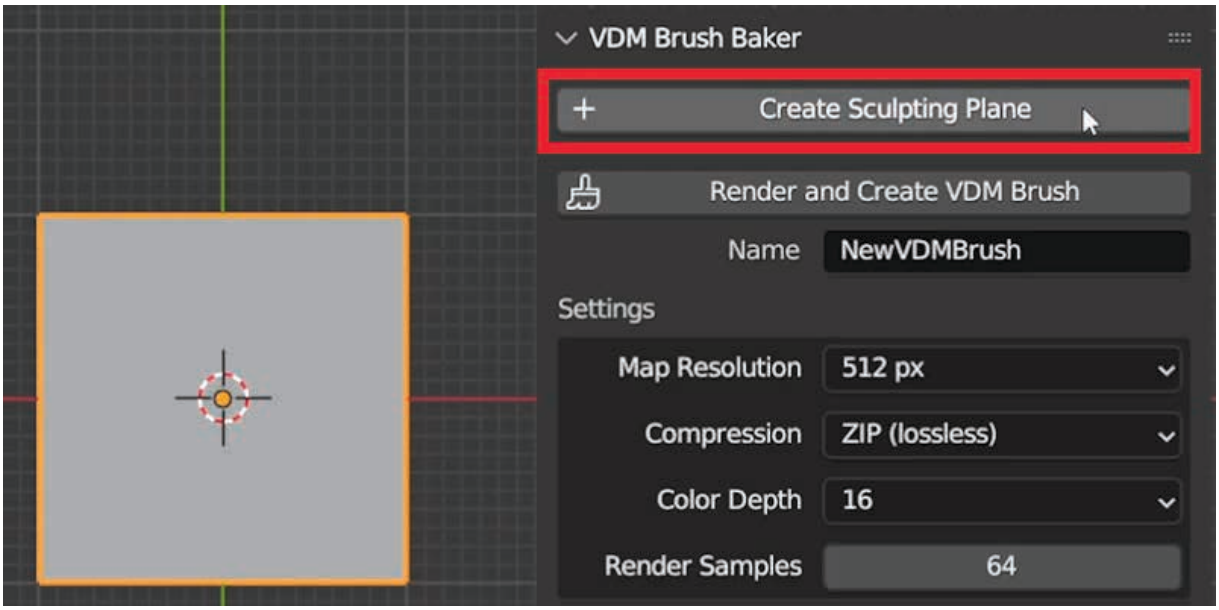
Using the **Space** stroke option in your Brush settings, as well as some other small tweaks, lets you create this kind of repeating brush. This is useful for stitches, and even for zippers, and other repeating patterns.



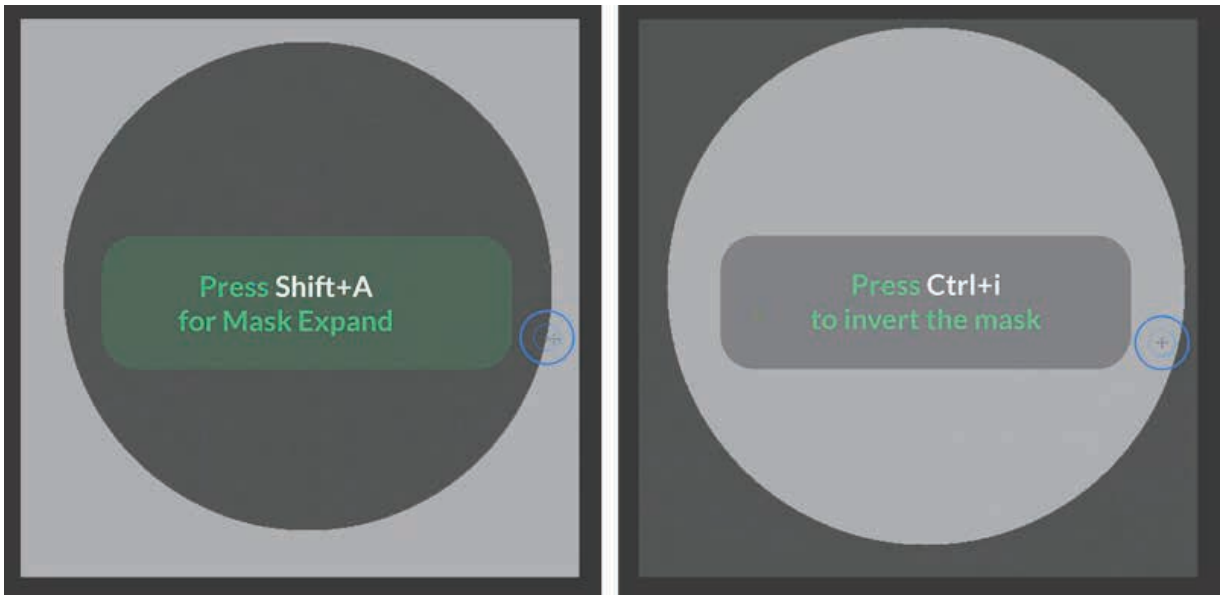
Enable the **VDM Brush Baker add-on** that comes with Blender since **version 3.6**. (Before that, it was an external file that you could download).



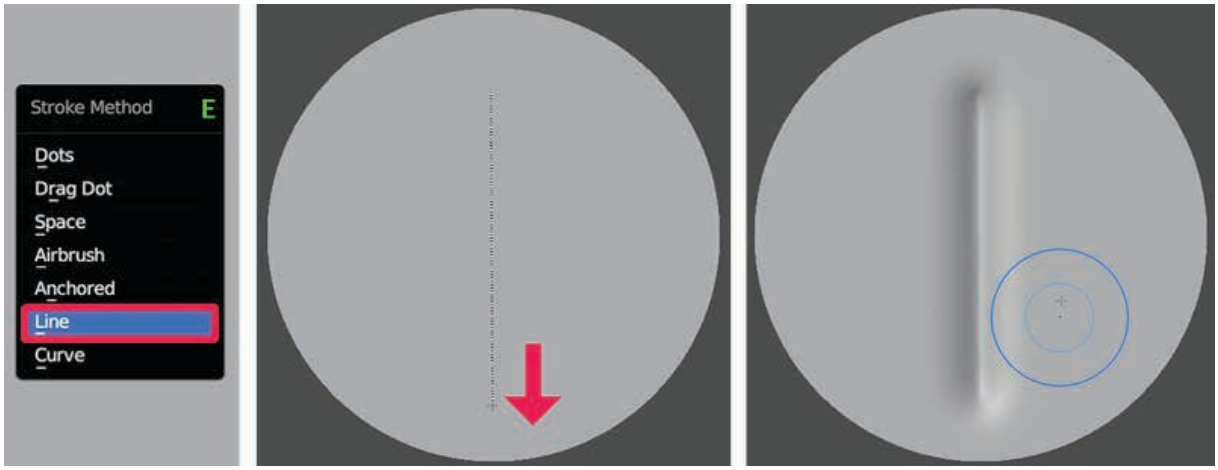
Click on **Create Sculpting Plane**.



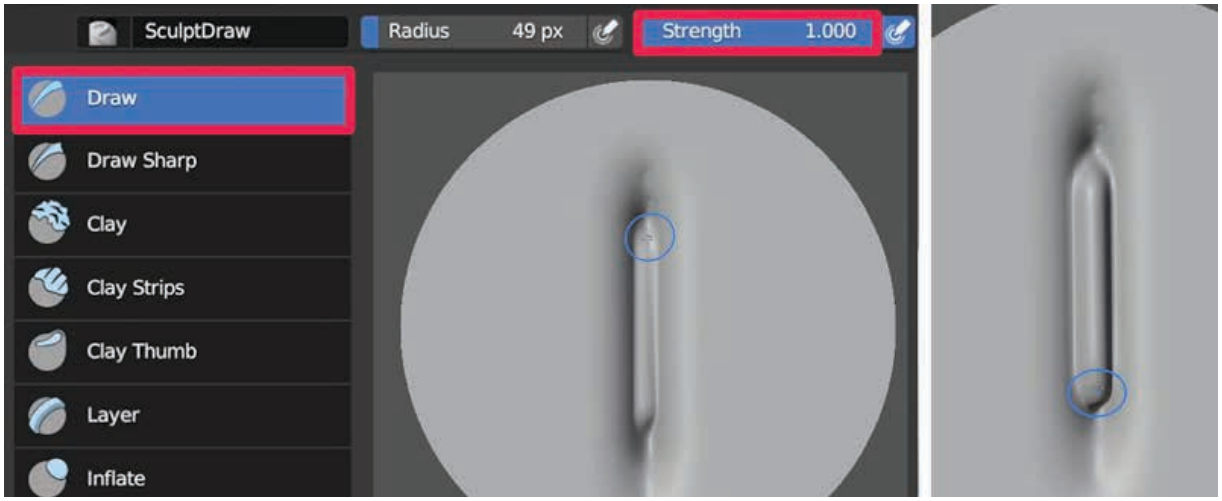
Mask the edges of the Plane, so you don't sculpt there by accident. You can draw a round mask from the center of the Plane by **pressing Shift+A**, using the **Mask Expand** function. Then you can invert that mask by **pressing Ctrl+i**.



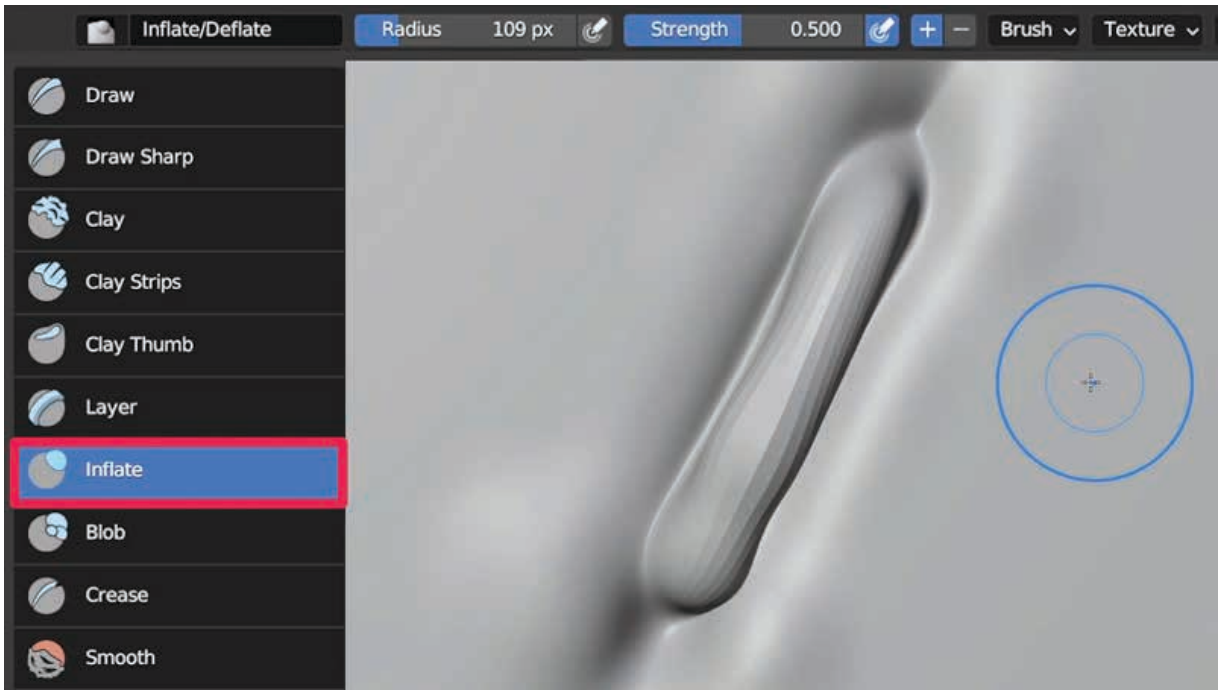
Using the **Draw Sharp Brush**, with the **Line Stroke method**, you can create a crease.



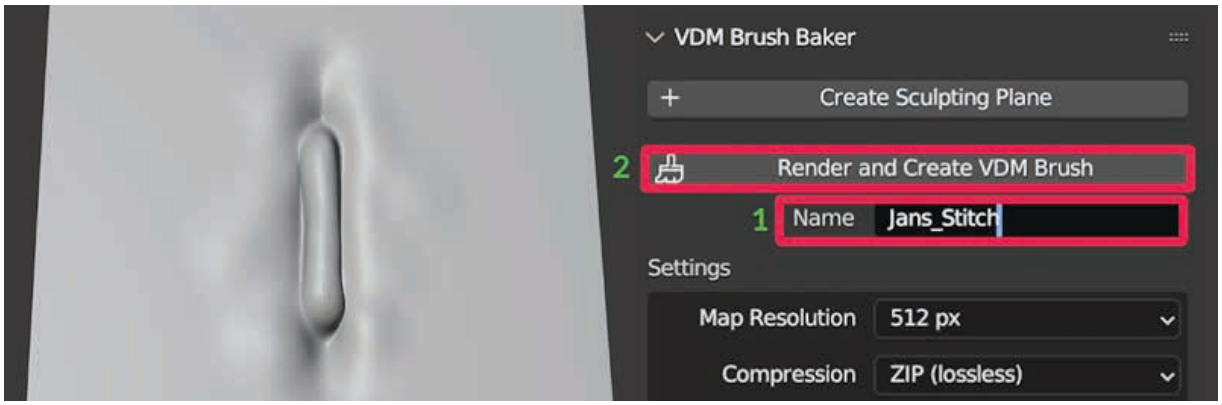
Then use the **Draw Brush** with a higher **Strength**, and also with the Line Stroke Method,



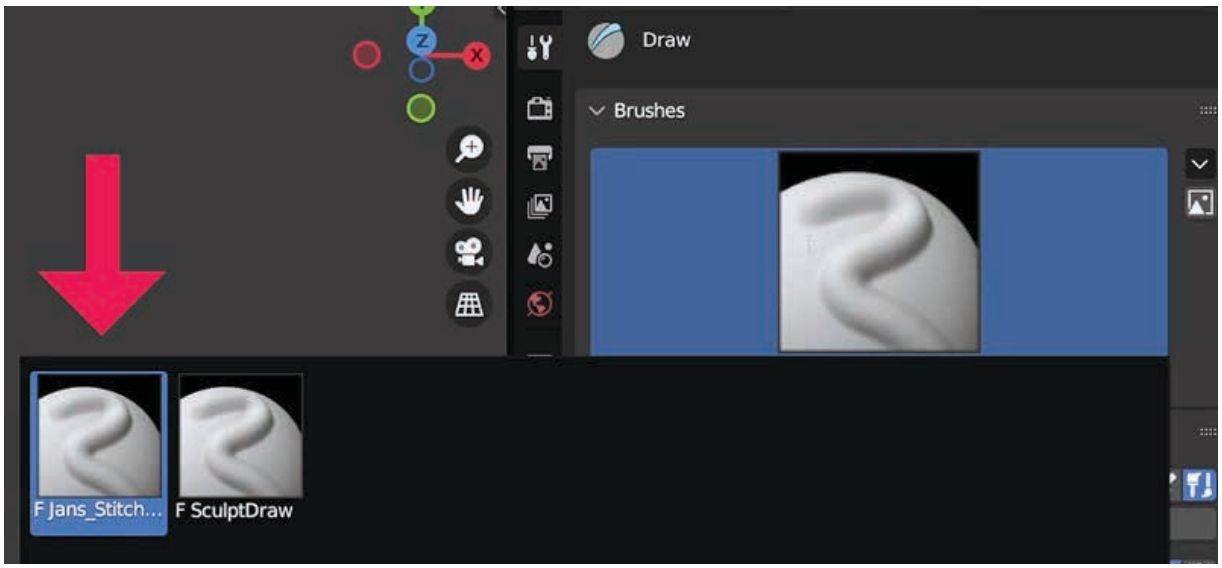
Smooth the result out, and make it less perfect by using the **Inflate brush**.



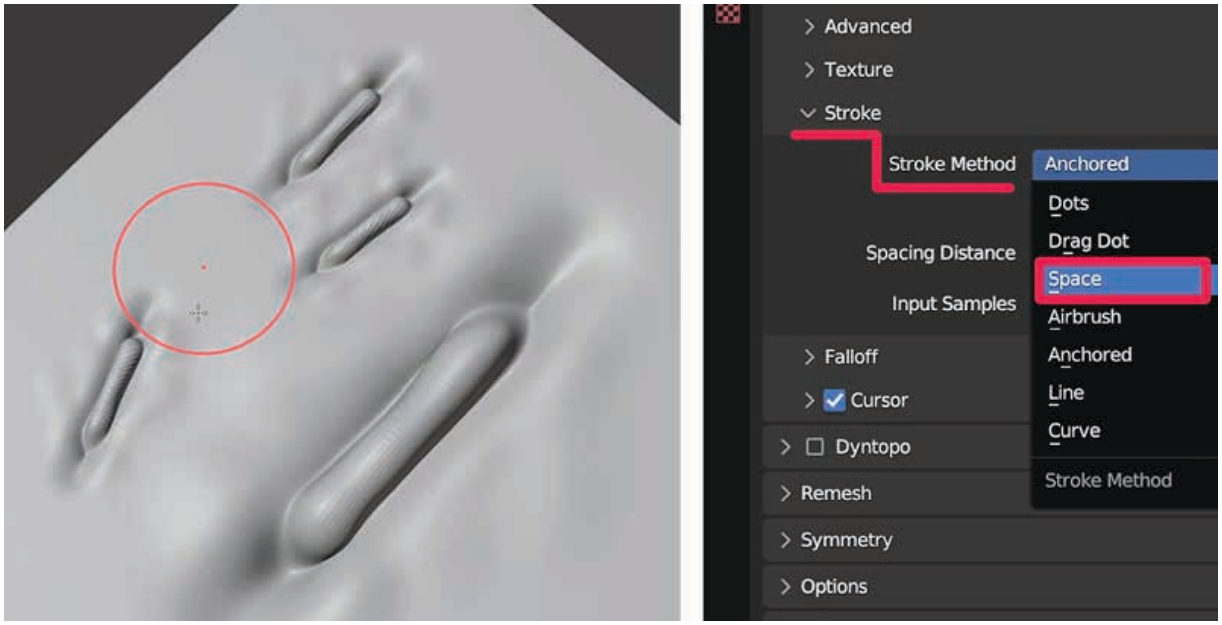
Clear the mask with **Alt+M**. Name it, and click on "**Render and Create VDM Brush**".



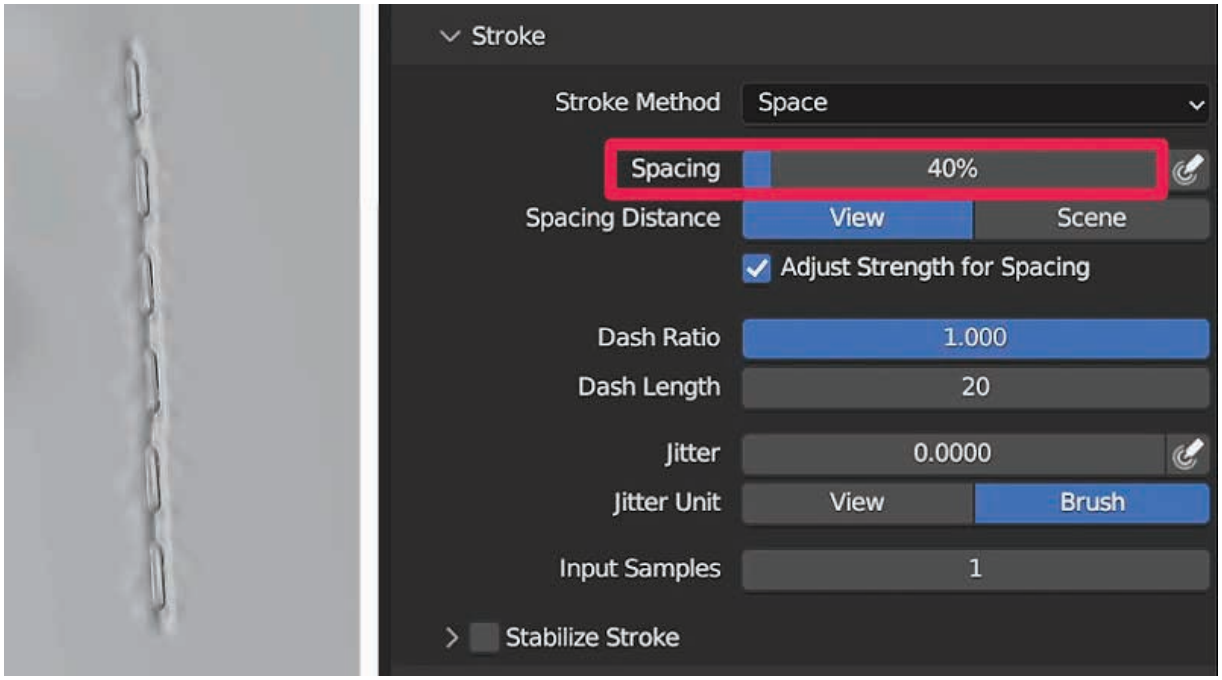
You'll find the new brush as one of the Draw brushes.



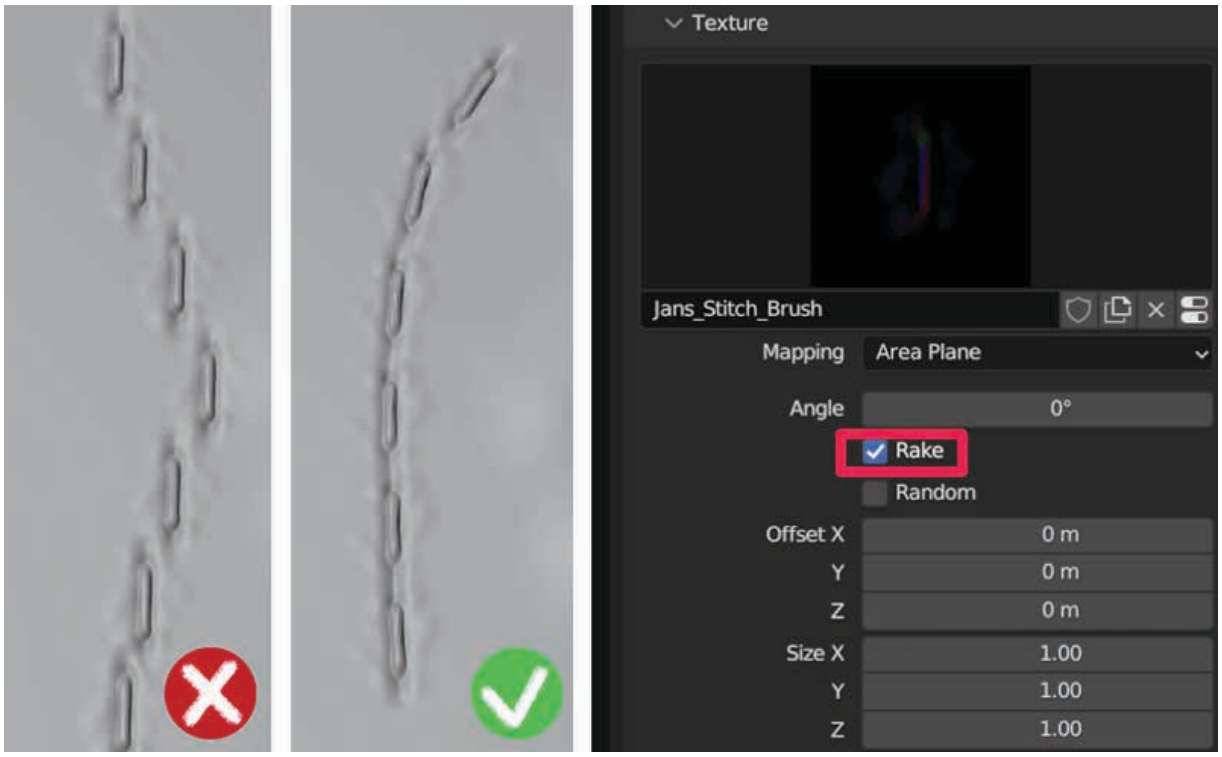
Now you can already use the brush to draw individual stickes. But we want to create a brush that lets us easily draw a whole stitch line. To do this, first set **Stroke Method to Space**.



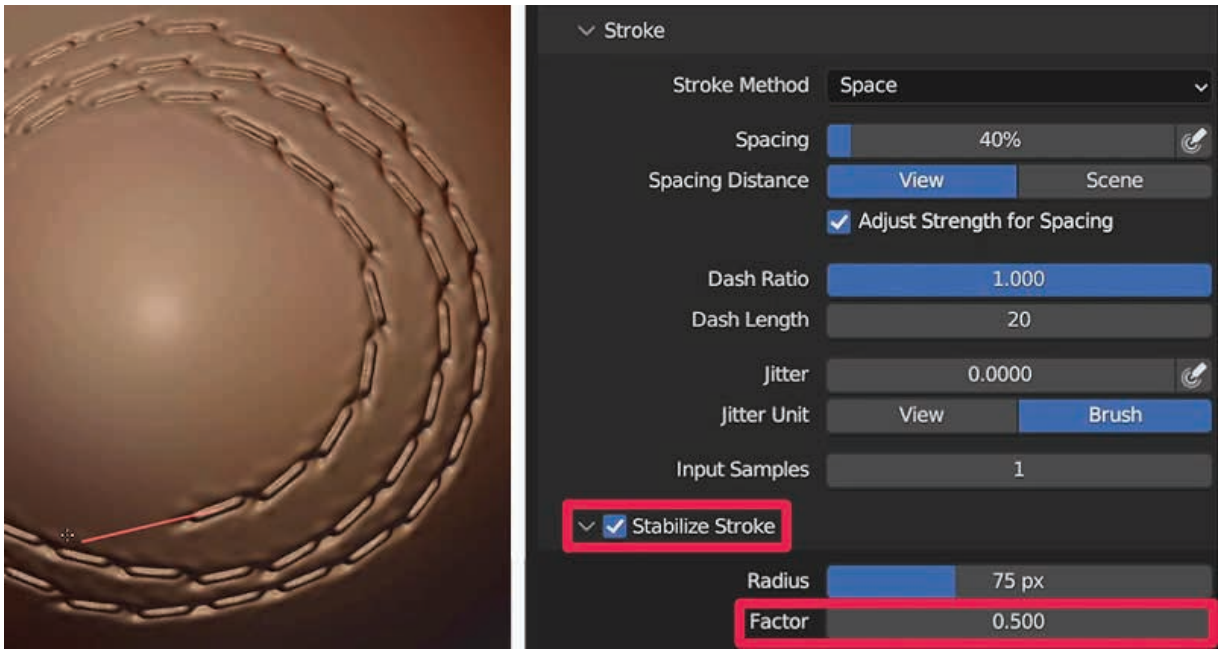
Experiment until you find the right **Spacing** value.



Enable the **Rake** option, so it follows the direction that you're sculpting in.



Finally, enable **Stabilize Stroke**, with the lowest **Factor** value (0.5) for the smoothest results.

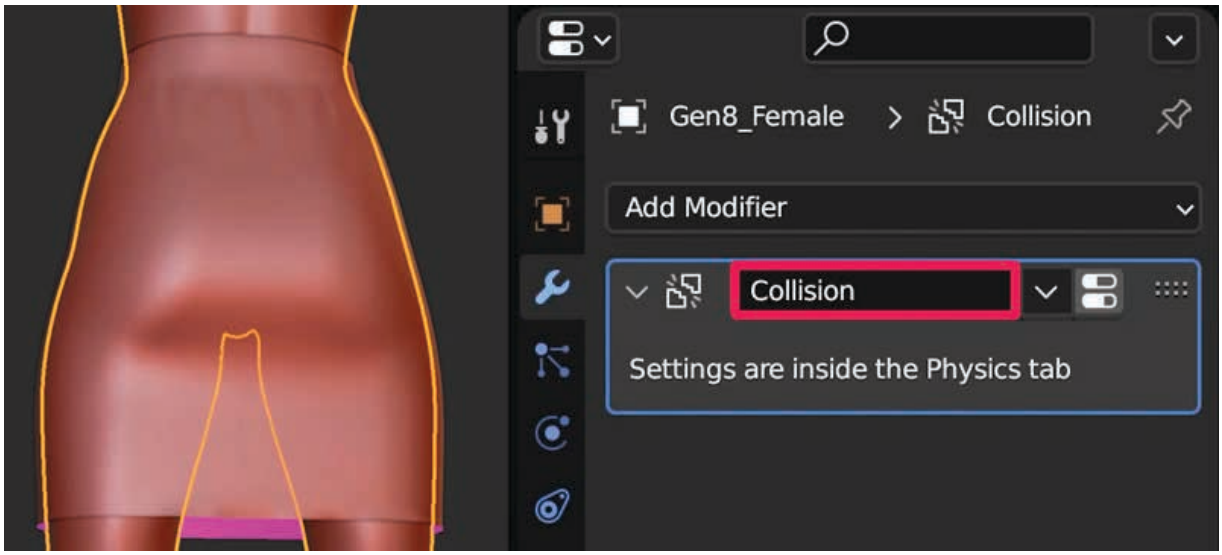


How to create Ruffled Skirts

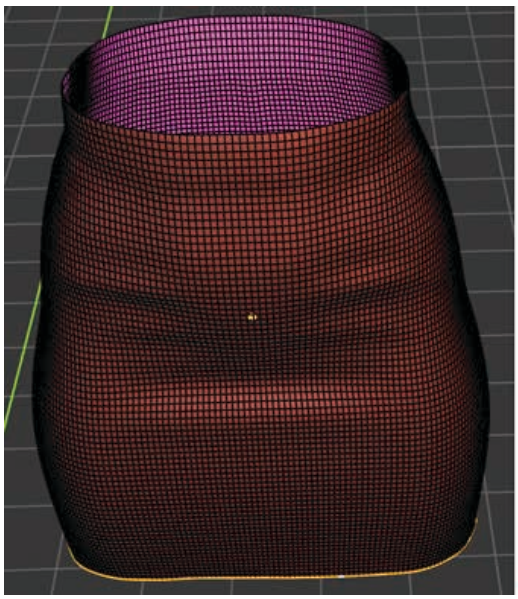
Using the Stitching function of the Cloth modifier, it's easy to create ruffled skirts.

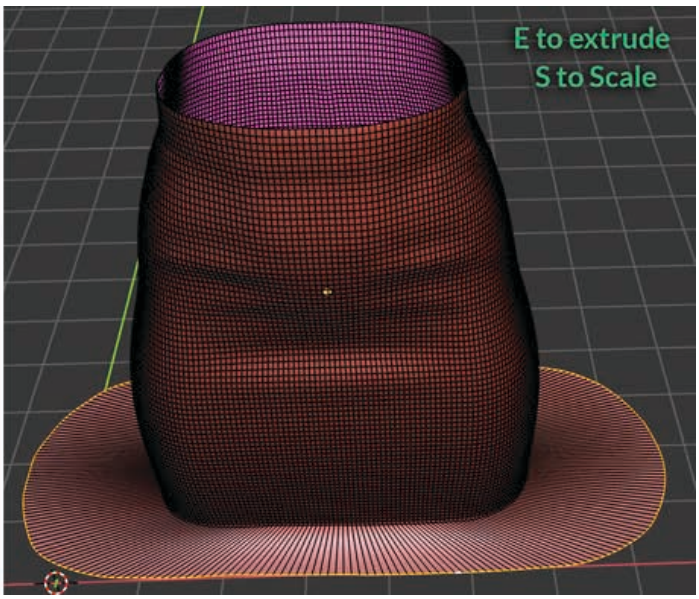
Watch the longer version of this video here.

Model a simple skirt (you can read or watch the previous topics about making a dress) and add Collision to a human base mesh underneath it. You can hide the base mesh, as long as it's got Collision on it.

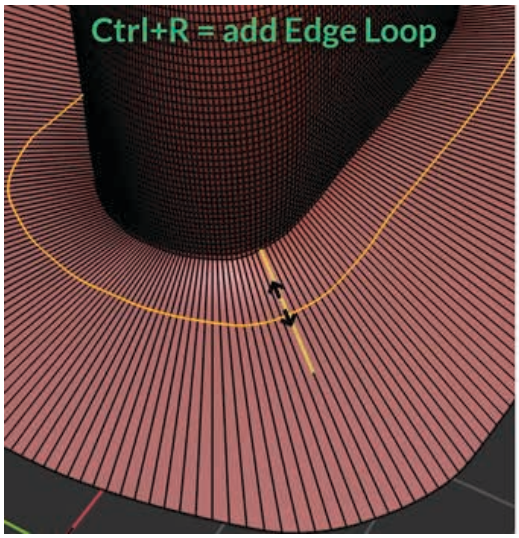


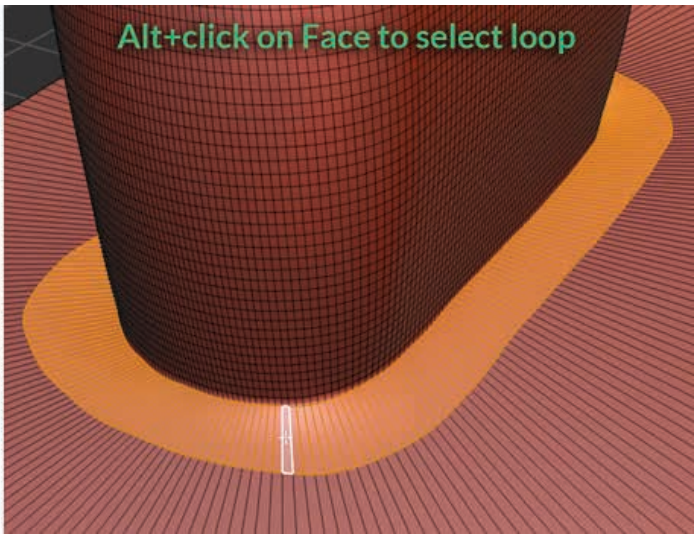
Select the bottom edge loop and **press E** to extrude. Right-click so the extrusion doesn't move, then **press S** to scale it up.



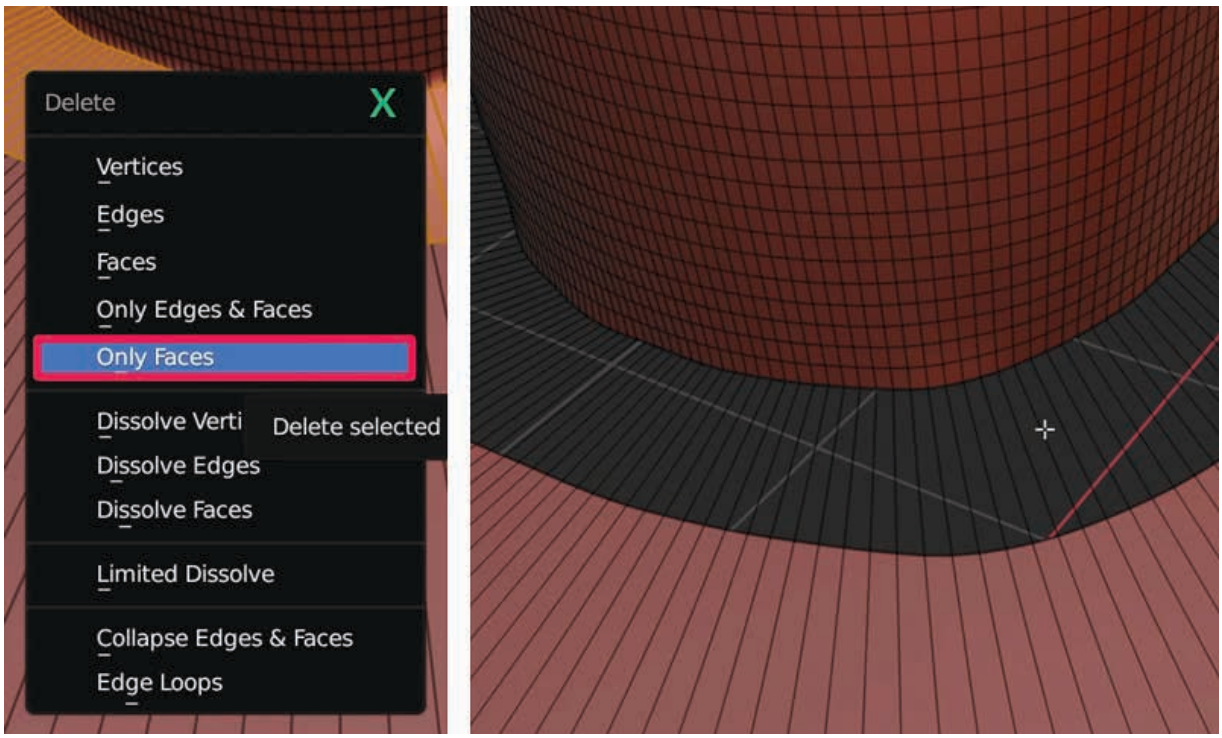


Add an Edge loop with **Ctrl+R**. Then, switch to Face selection mode and **hold Alt** while clicking on a Face to select that loop of Faces.

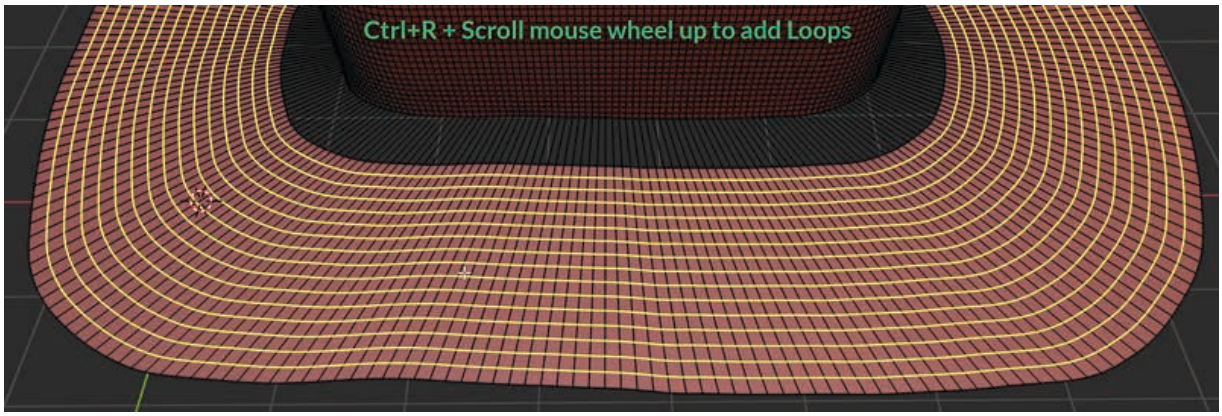




Press X to Delete and choose "**Only Faces**". This deletes the Faces but keeps the Edges in between them. Those Edges will be used for Cloth Sewing.

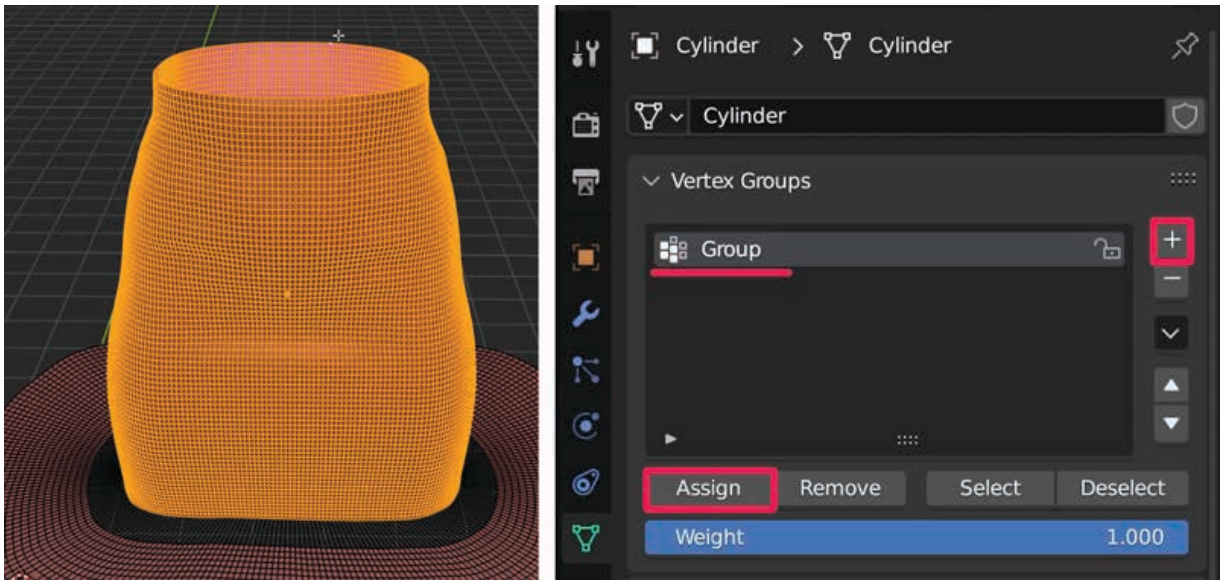


Add more Edge Loops to the extruded piece with **Ctrl+R**. You can increase the amount of loops by scrolling the mouse wheel up. You need this detail for the Cloth simulation to have something to deform.

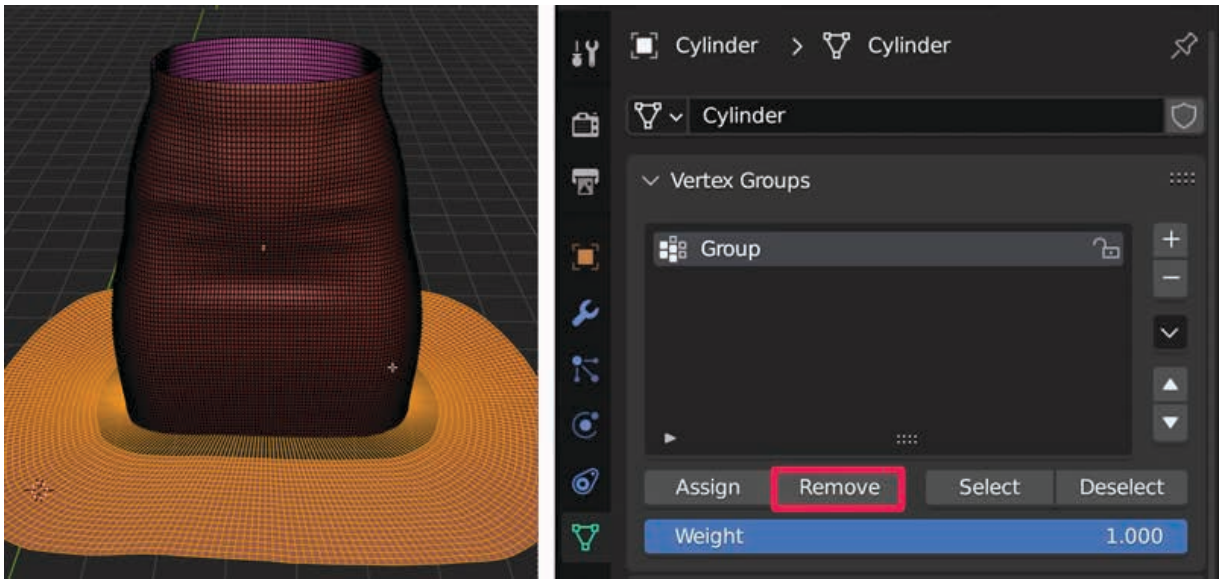


We now need to make sure the extruded part will be included in the Cloth simulation to create the ruffles, but the skirt itself will not be included in the simulation. To do that we'll use a Vertex Group.

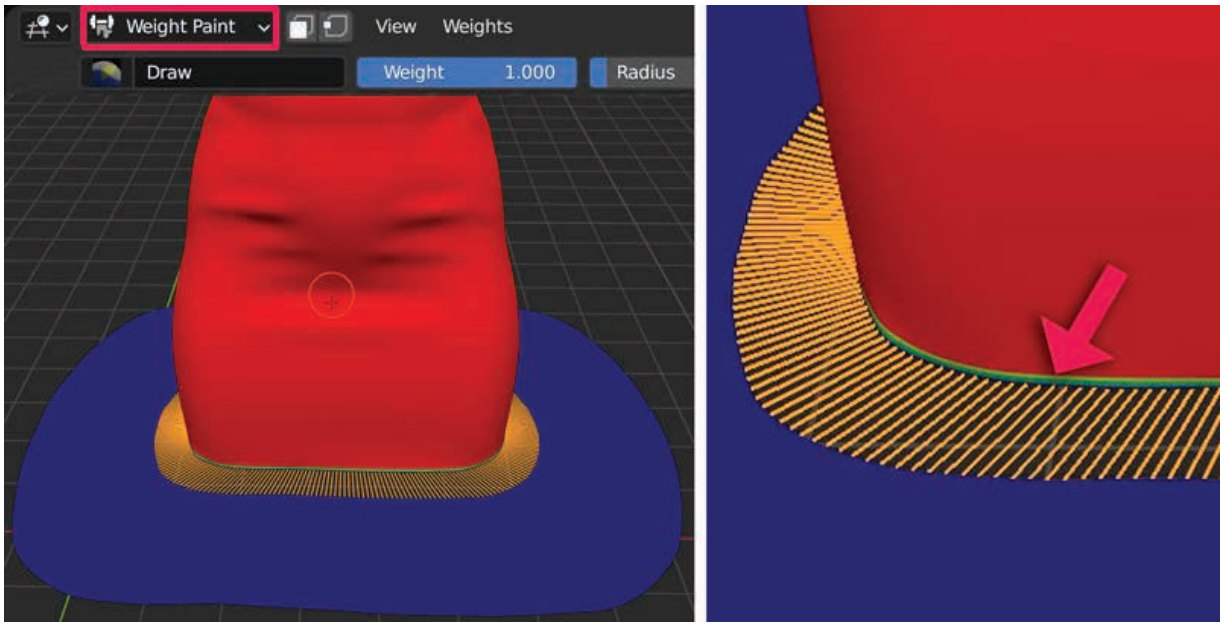
Select the skirt and add it to a Vertex Group. You can do this by selecting the top Edge Loop by Alt-clicking on it, then **pressing Ctrl and [+ on the Numpad]** to grow the selection until you've got everything up to (but not including) the Sewing edges.



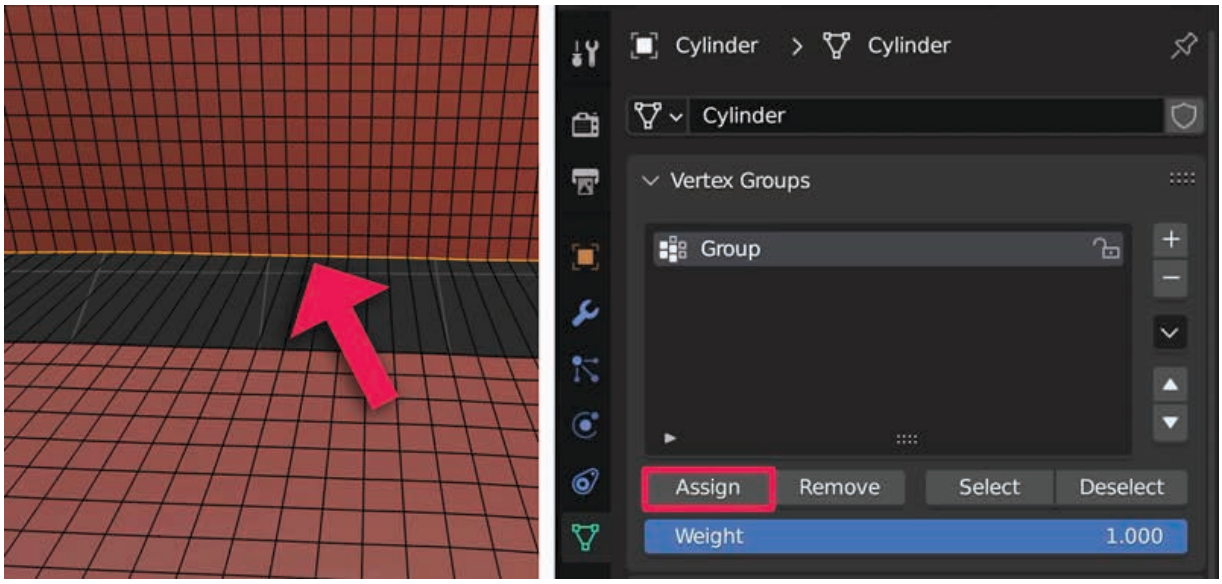
You can invert the selection with **Ctrl+i** and then click on **Remove** to make sure that part is not assigned any weight.



You can check in Weight Paint mode whether the Vertex Group looks good. Everything blue will be included in the Cloth sim, everything red will not. If there's any part that's not either blue or red (green means in between both values), you can easily spot it here.



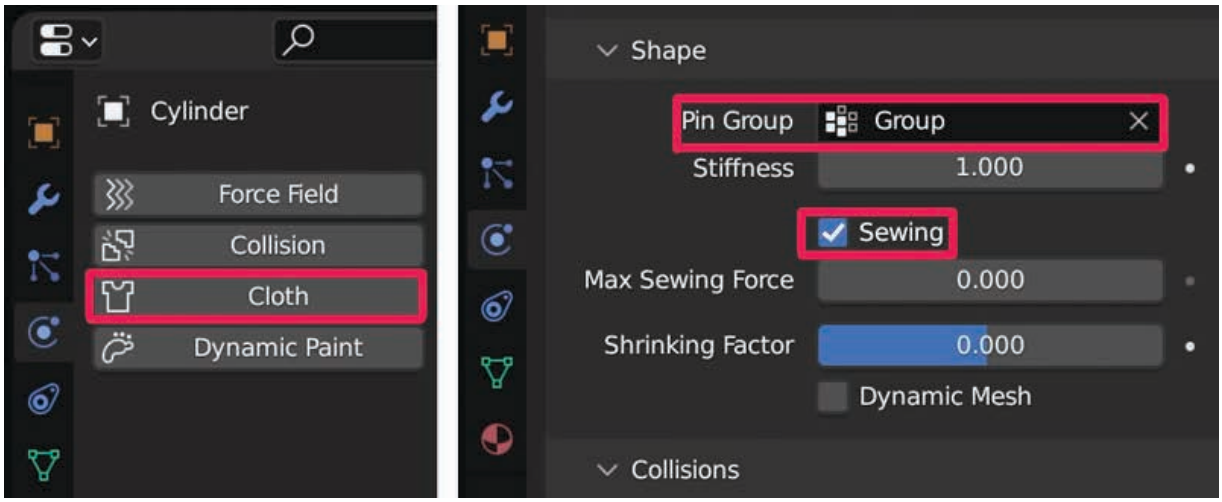
Select any such Edge Loop by **holding Alt and clicking on it**, then click **Assign** in the Vertex Group panel.



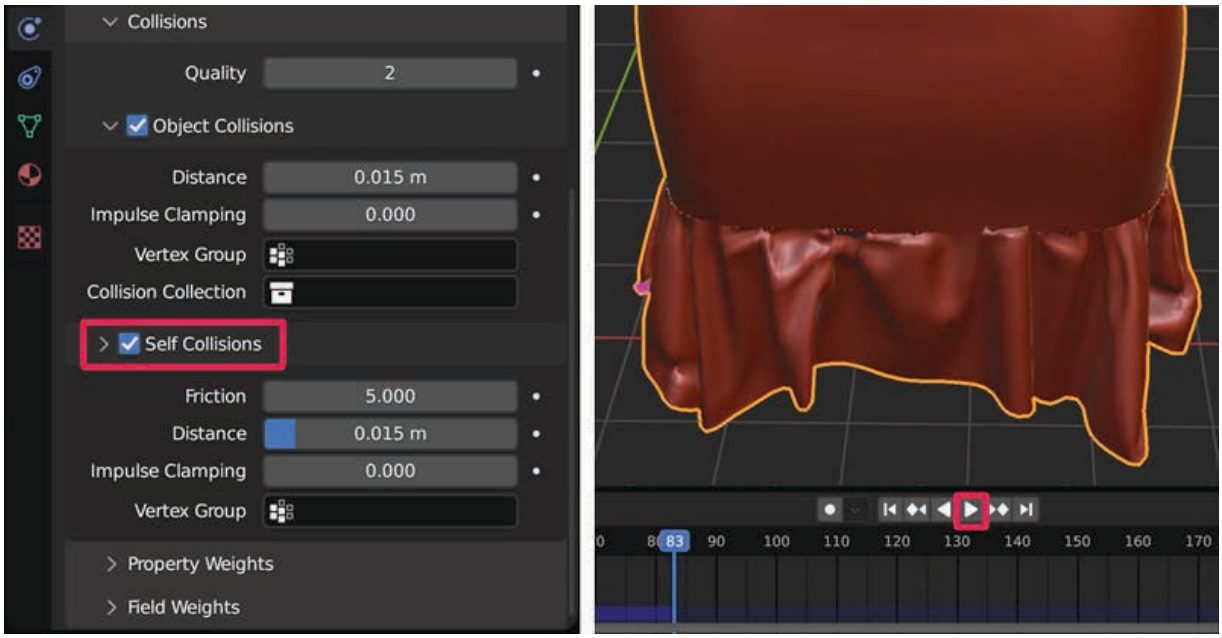
Make sure you're at the start of the Timeline and with the mesh selected click on **Cloth** in the Physics panel to add a Cloth modifier to it.

Add the Vertex Group to it under **Pin Group**. This will "pin" everything with a value of 1 in the Vertex Group so that it can't move.

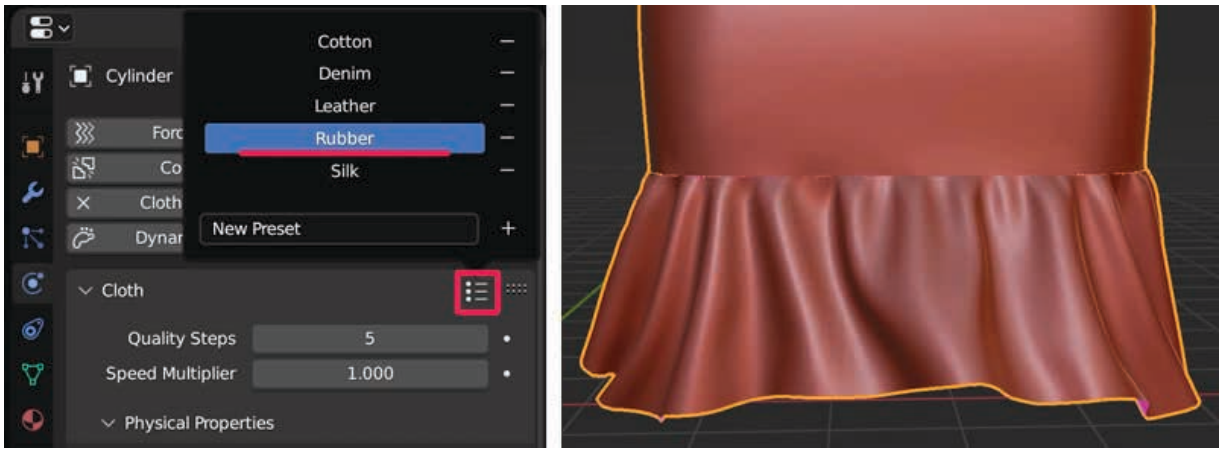
And of course, don't forget to enable the essential **Sewing** function, which is the key to all this.



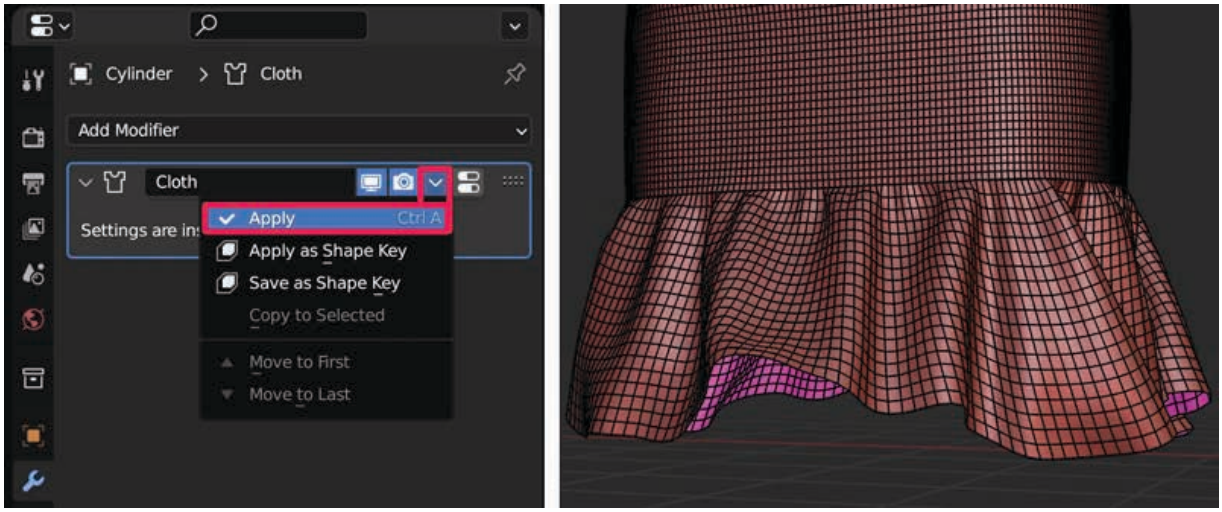
Finally enable **Self Collisions**, which will make the Cloth sim more realistic. Now when you press the Spacebar from the beginning of the Timeline, the Cloth simulation will start. It'll be slow, so be patient. You can pause or stop it when you think it looks good enough.



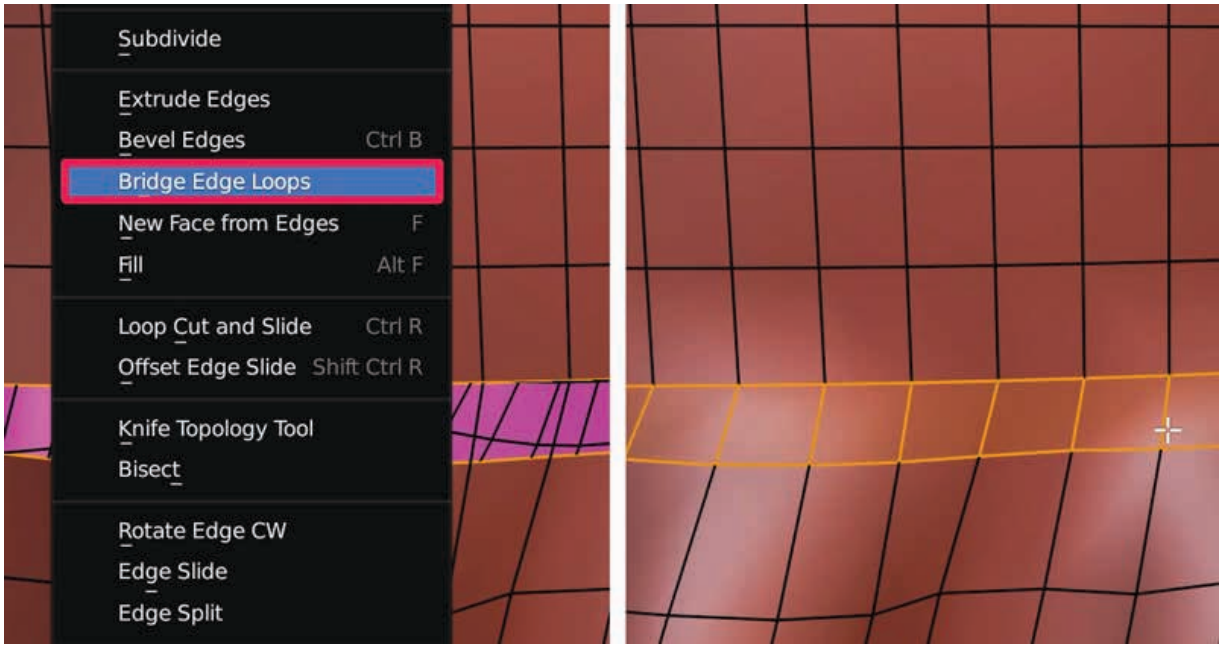
If you're not sure about the result, you can try out the different **presets** the Cloth modifier offers. I liked the result from the **Rubber preset** in this case. You can also try scaling up your mesh, in my experience that often gives a better result in Blender than simulating at real-life scales.



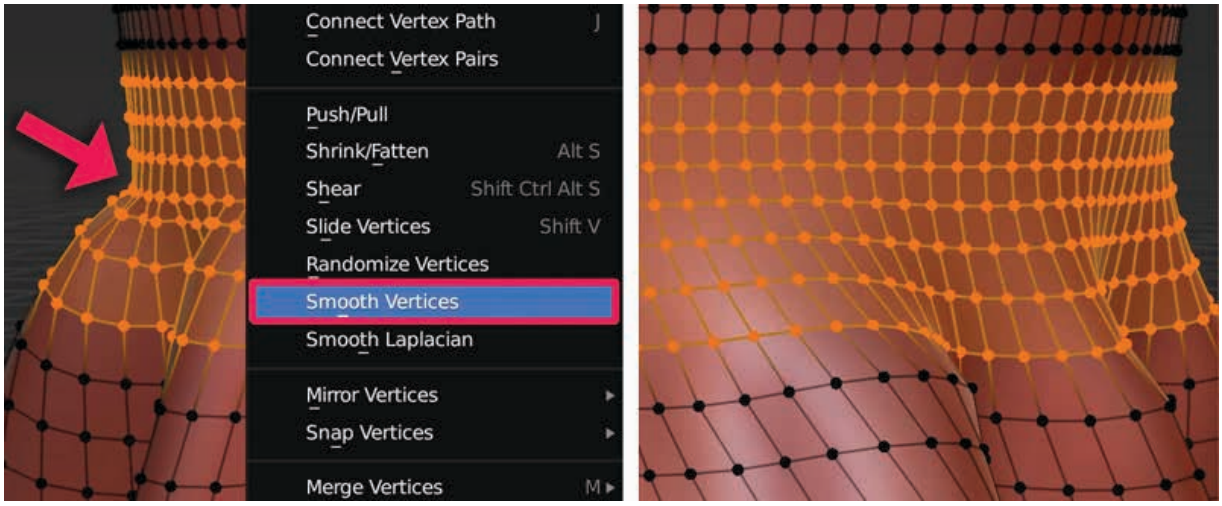
To apply this as a mesh, that you can then sculpt on for example, you can click on the drop-down menu in the Cloth modifier and choose **Apply**.



There will be a small gap where the Sewing did its thing. You can hold Alt and Shift and click on those edge loops to select them, then right-click and choose Bridge Edge Loops to close the gap.



If the transition from the skirt to the ruffles is too abrupt, you can select those vertices and right-click, then choose Smooth Vertices. You can repeat this by **pressing Shift+R** a few times until it looks smooth enough.



You can also use the **Smooth brush** in Sculpt Mode to smooth that area out, of course. And finally add a **Solidify modifier** for some thickness.

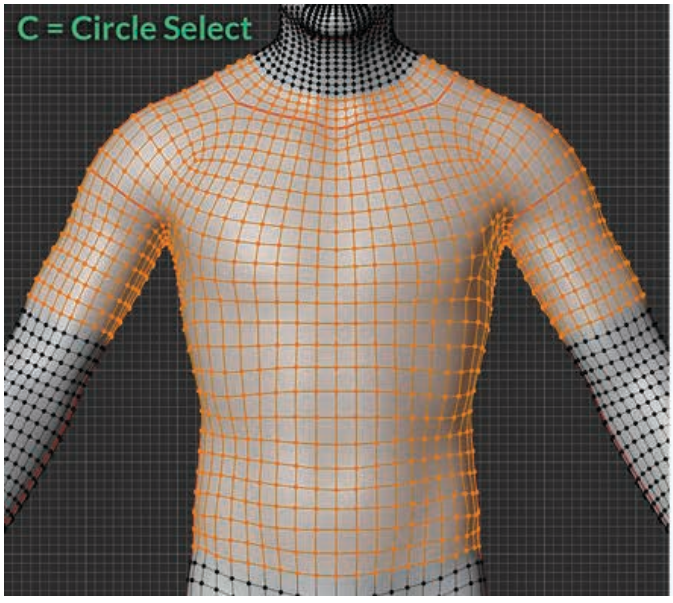
Marvelous Designer-like Cloth pulling / grabbing

In this topic, we'll learn how to use Hooks in combination with a Cloth modifier, to create realistic looking folds that we can then use as a basis for further sculpting. Note that I've created a script for Blender that does a lot of the steps in this tutorial for you by just pressing a button, you can find that here.

You can watch a longer, more detailed video about this topic here.

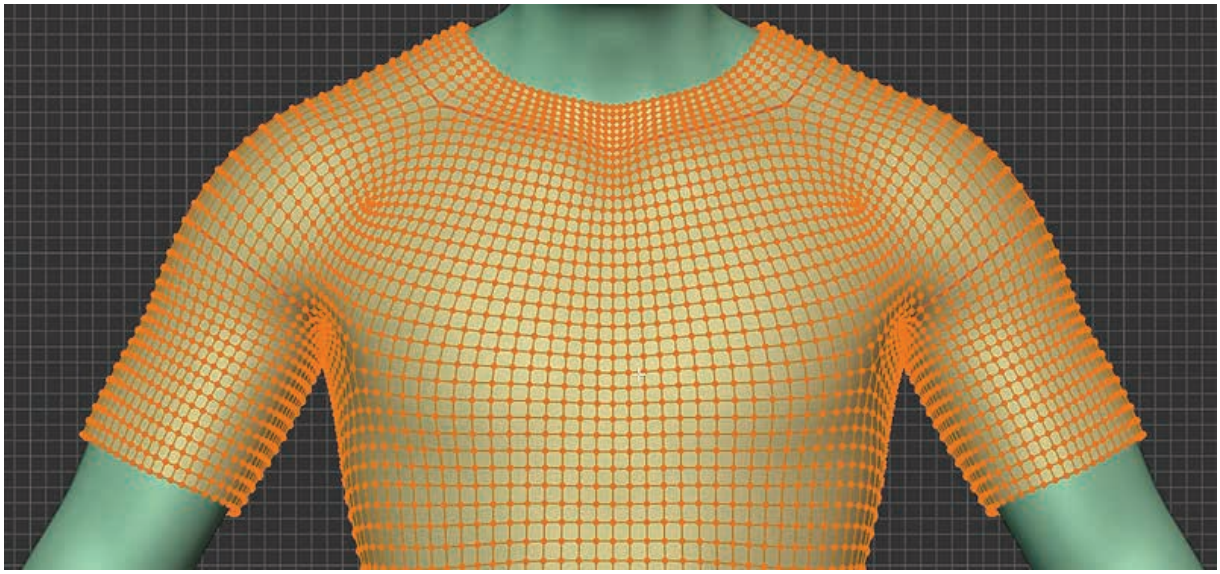
Create a basic shirt by duplicating a selection of your character, separating it by selection and scaling it up along normals, as explained in the topic about making clothes.

In short: select a region in Edit Mode that you'd like to turn into a shirt, using **Circle Select** (which can be activated by pressing C). Then **press P**, and choose **Separate by Selection**. Now you have created a new object from your selection.

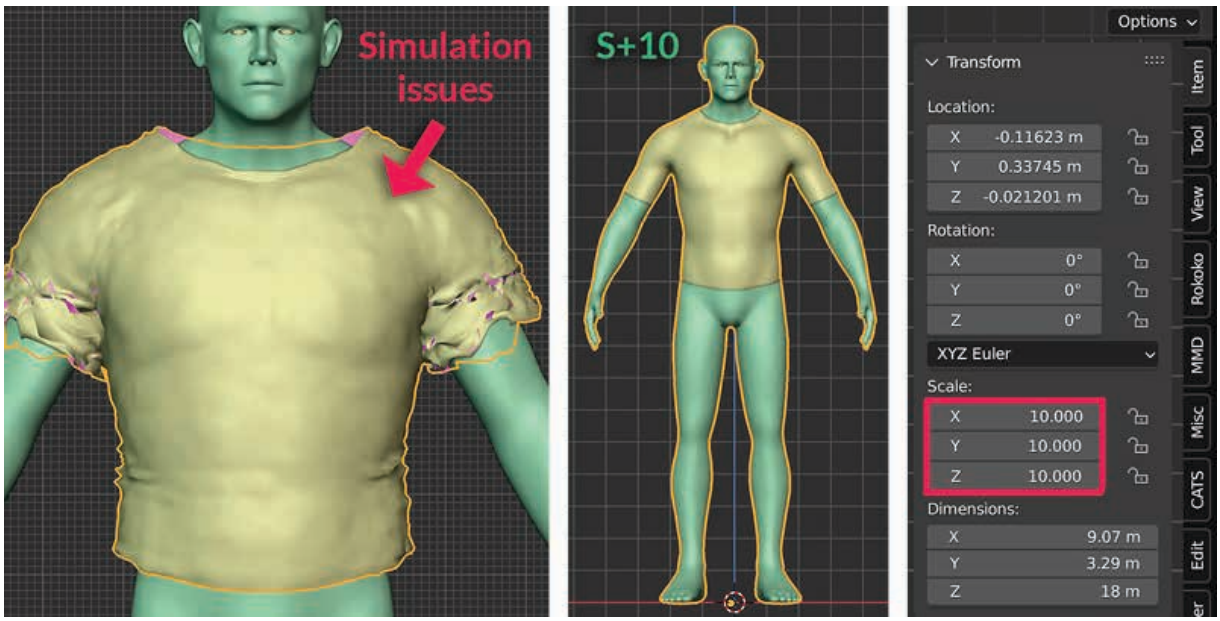


Currently it's in exactly the same location as the body, so scale it up to more clearly see the difference. Do that by selecting all Vertices of the shirt in Edit Mode and **pressing Alt+S**. That scales it evenly along the Normal directions.

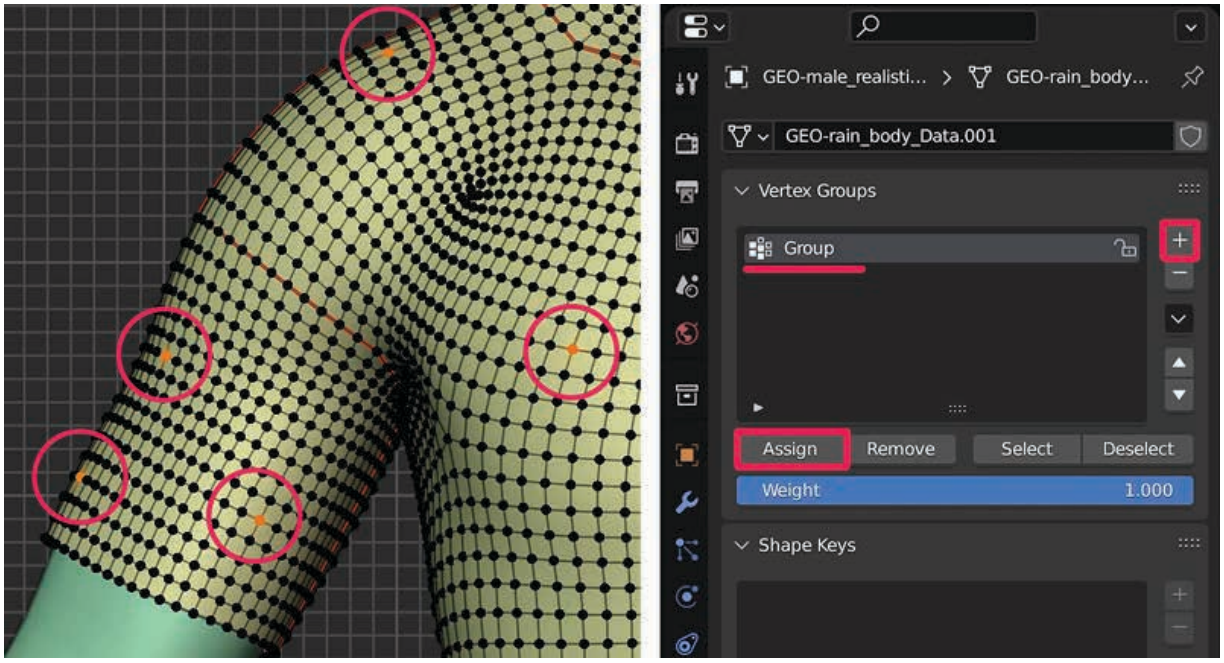
Make sure it has enough vertices for the cloth simulation. If not, right-click and choose Subdivide.



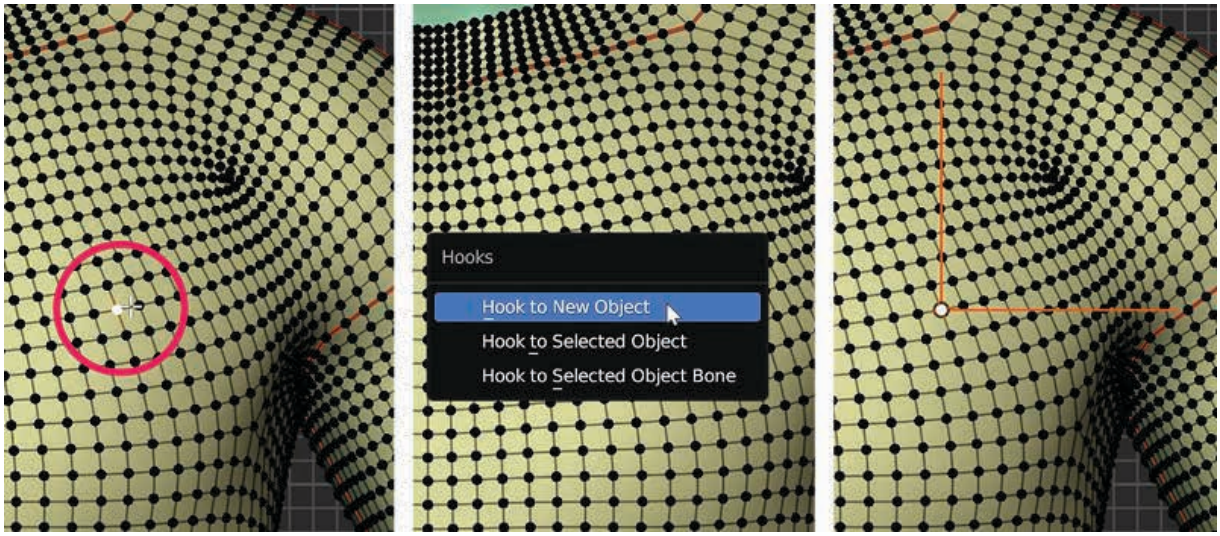
For more reliable cloth simulations, I recommend scaling the objects up. In this case I scaled everything up 10 times. Don't add a Cloth sim yet at this point, the below image is just for illustrating the cloth issues when the scale it too low.



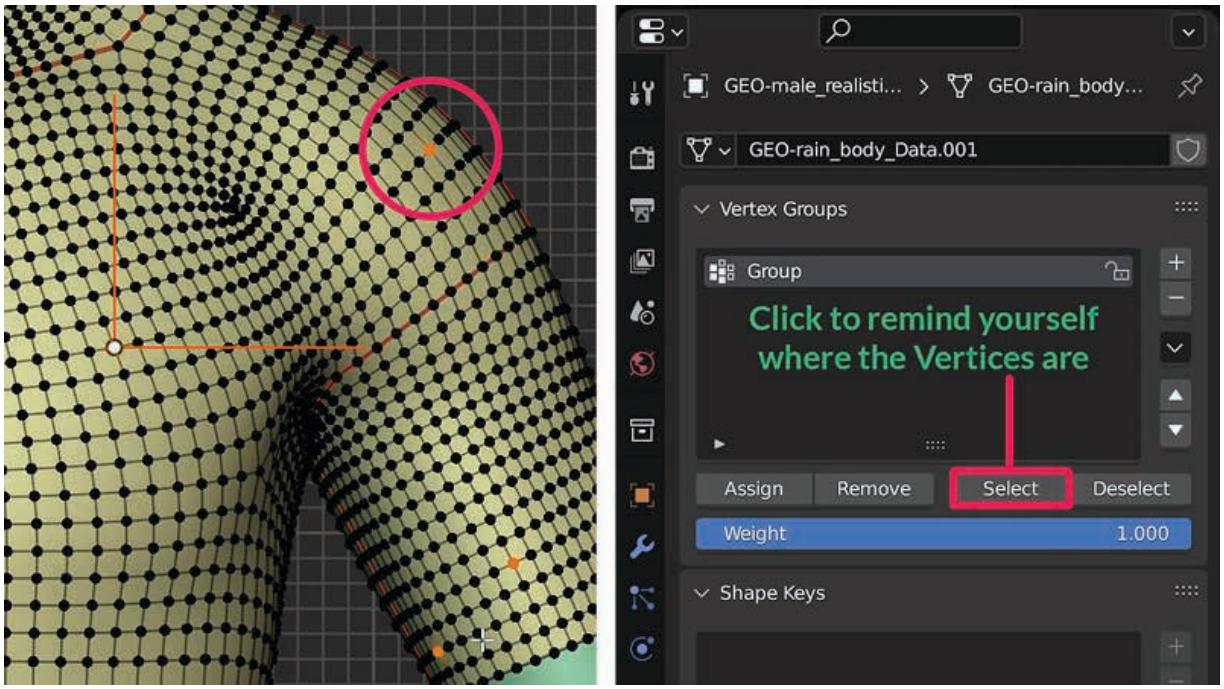
Select 10 to 20 random vertices on the mesh, and add them to a new **Vertex Group** by clicking **Assign**.



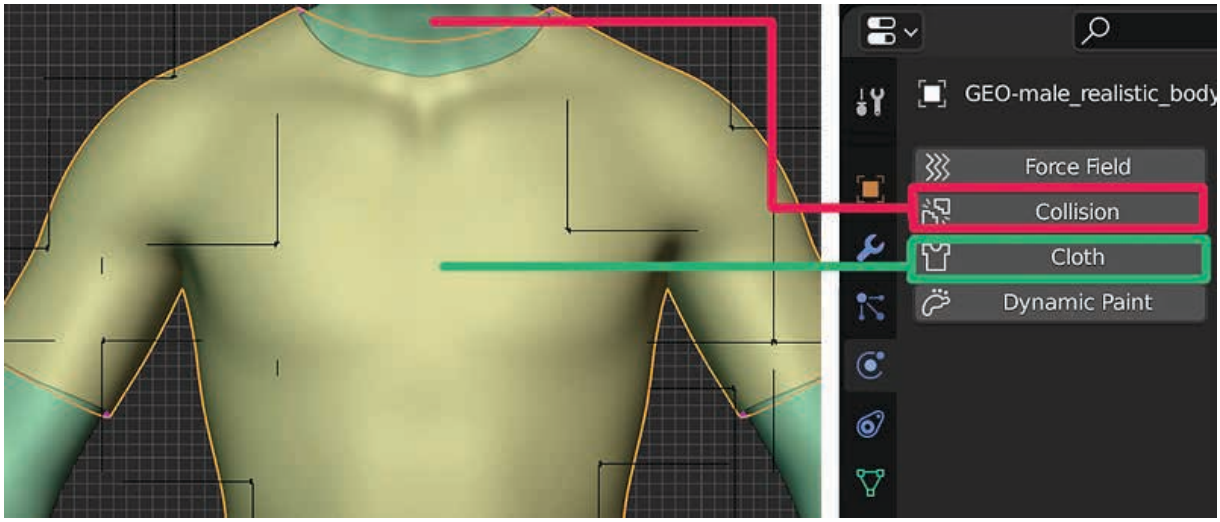
Now select only one of these vertices and assing a Hook to it by **pressing Ctrl+H > Hook to New Object**.



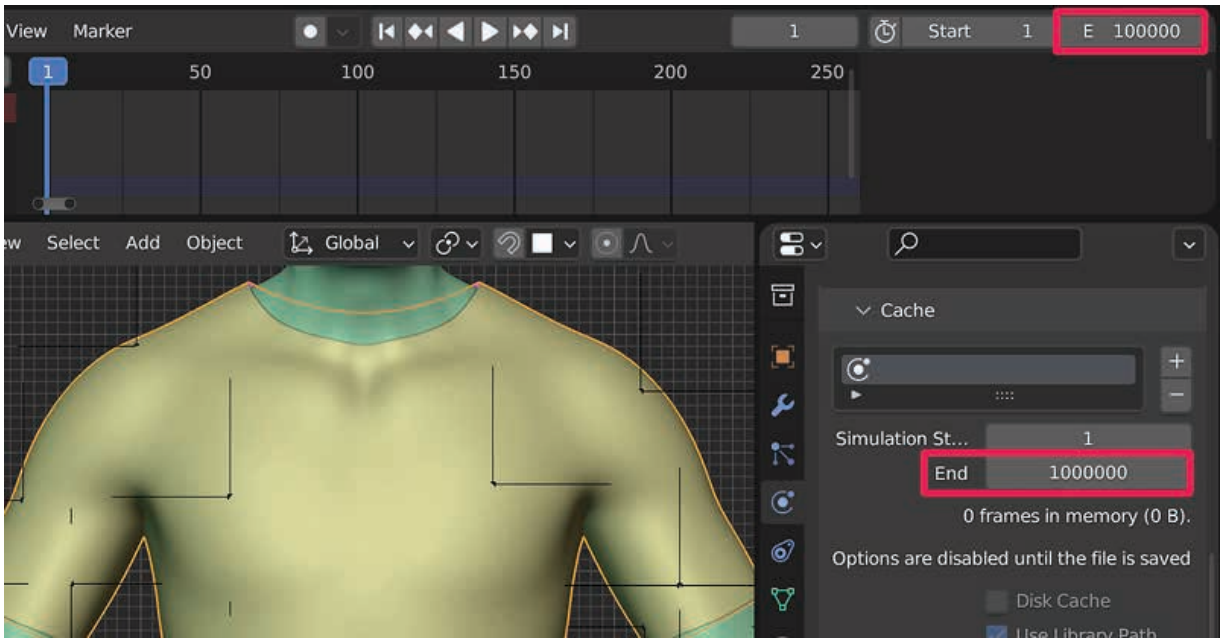
Now repeat this for the other vertices in the Vertex Group. You can remember where they are by clicking on **Select** in the Vertex Group Window. Then select just one of those vertices, and add a Hook to it.



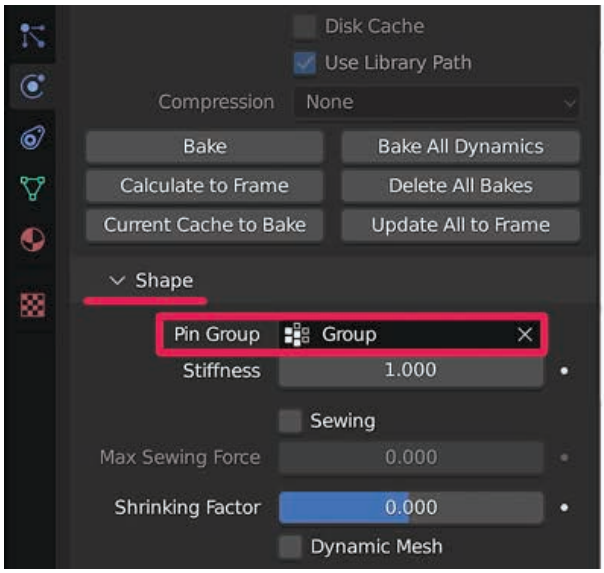
Now add **Collision** to the character mesh and **Cloth simulation** to the clothing.

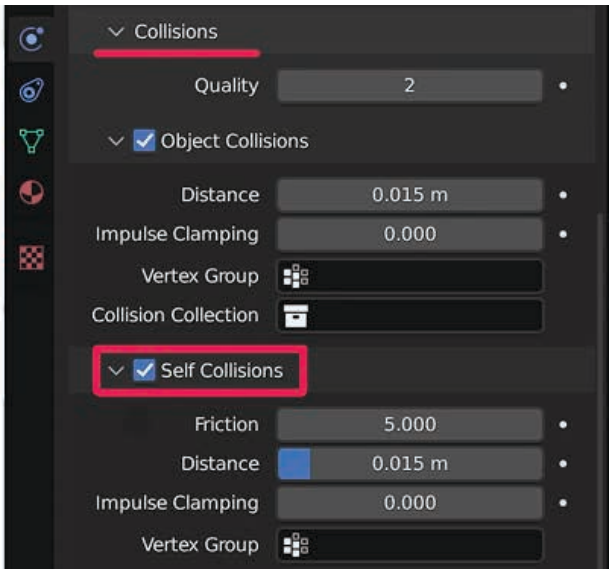


Make sure the simulation can run for long enough, set the **Cache** and **Timeline End** values to some ridicilously high value. That way it won't loop back after 250 frames and you can experiment with the simulation while it's running, without feeling nervous.

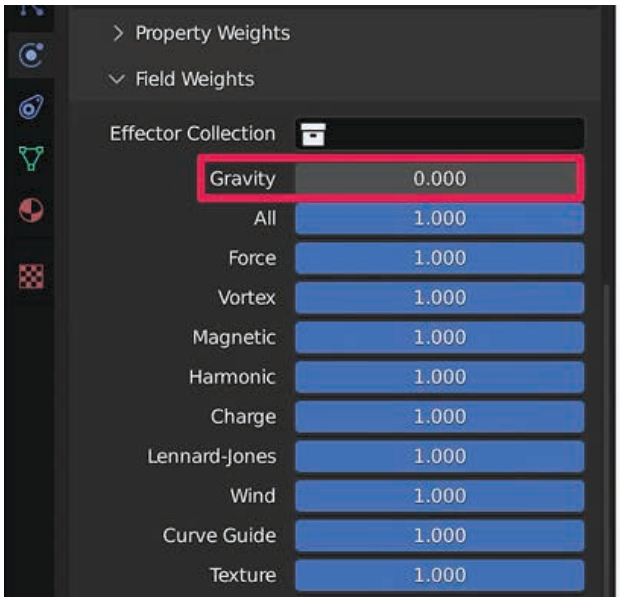


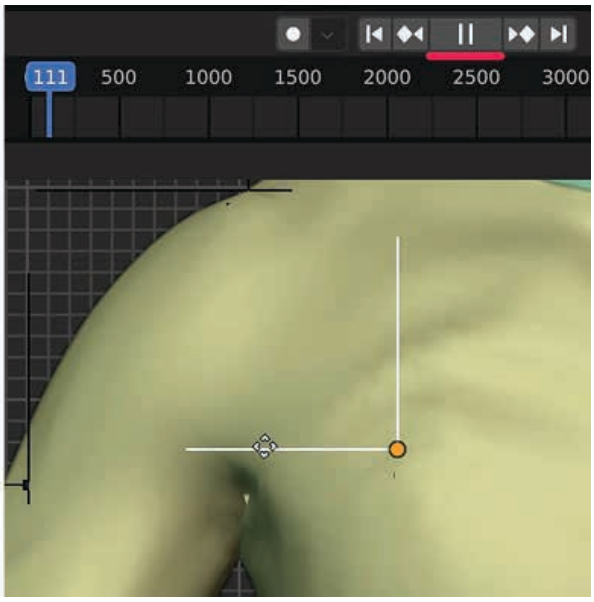
Add the Vertex Group we created earlier to the **Shape > Pin Group**. This will exclude those vertices from the simulation, so we can use them with the Hooks. And enable Self Collisions so that the cloth simulation is more accurate.



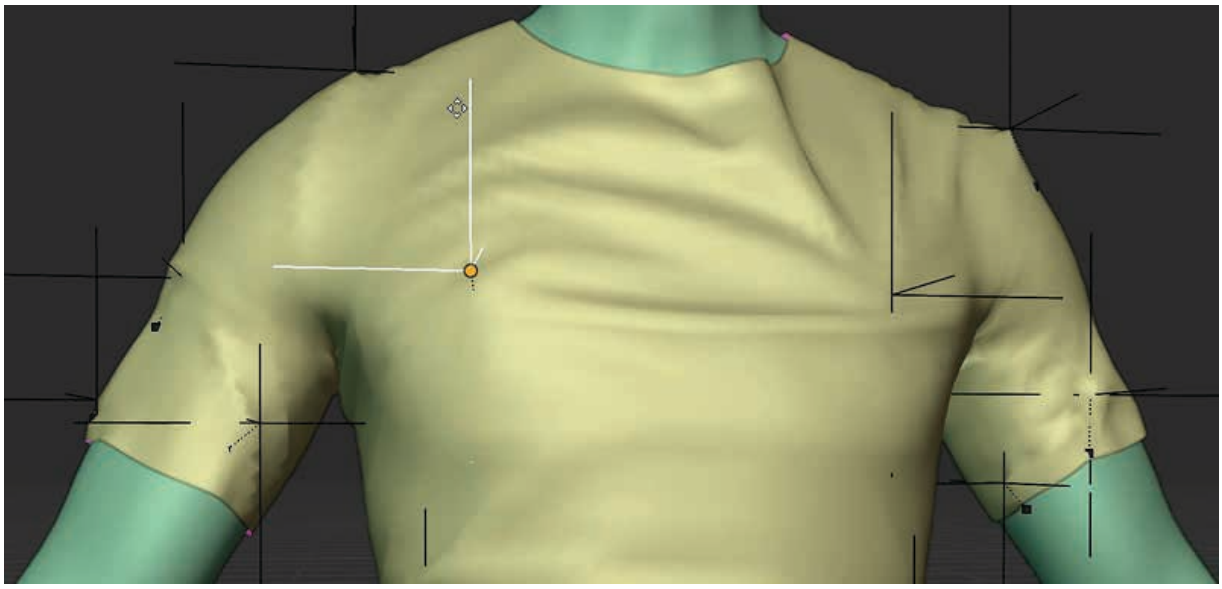


Scroll down in the Cloth settings to reduce **Gravity** to zero. Then start playing the Timeline from the first frame. While it's playing, you can grab the Hooks in Object Mode and move them around. As you can see, the cloth sim will follow, as if you're grabbing the cloth there.

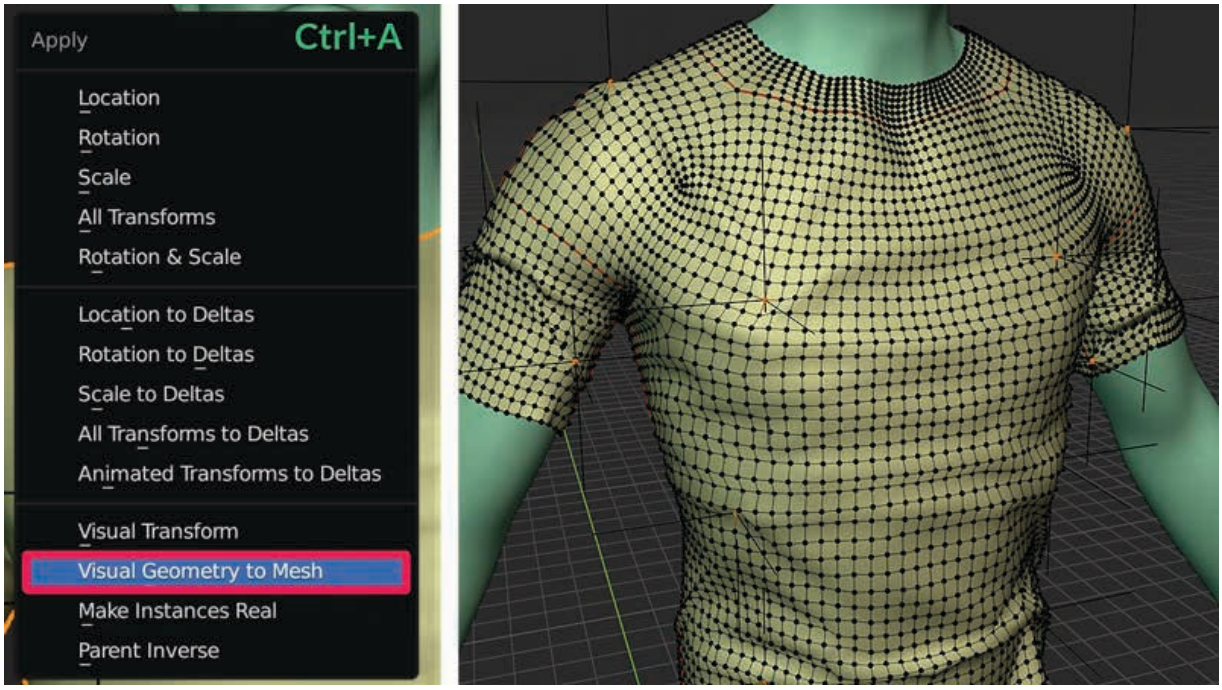




I recommend making only small adjustments, and going around the mesh giving all the Hooks some attention, all the while keeping it subtle. it's better to have no folds than unrealistic folds.

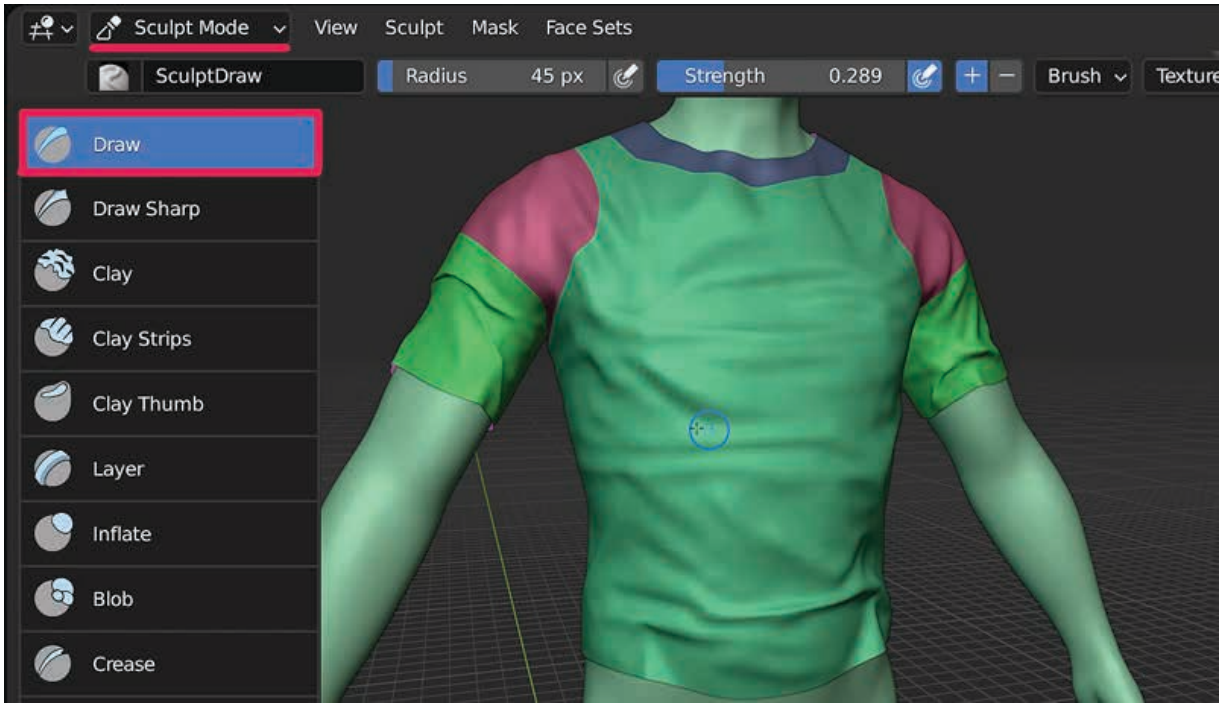


Once you're happy with it, pause playback on the Timeline and **press Ctrl+A, choose Visual Geometry to Mesh**.



Now you can go to **Sculpt Mode**, add a **Multires modifier** and use for example the **Draw brush** to further add some folds. The benefit of this workflow, is that you get realistic folds from the simulation that you can then use as a guideline.

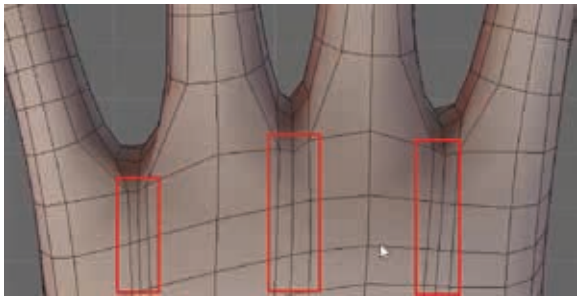
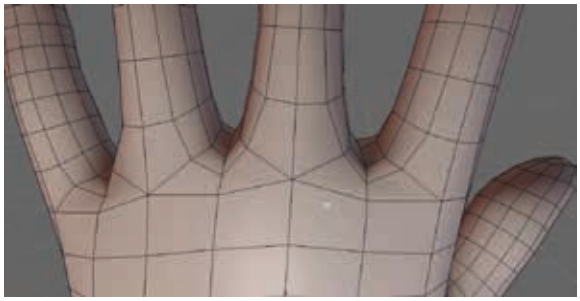
Rather than going from your imagination, which is the kind of thing that can lead to unrealistic folds. Of course, it's also a good idea to use photographic reference.



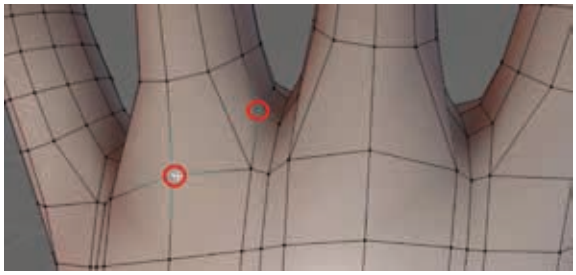
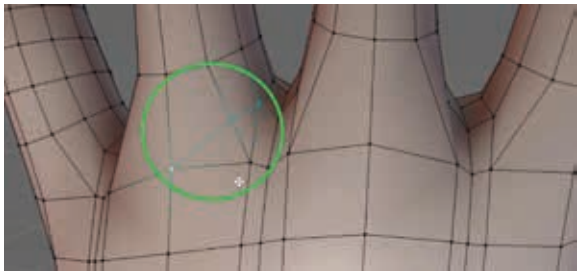
RETOPOLOGY

Solving Edge Flow problems

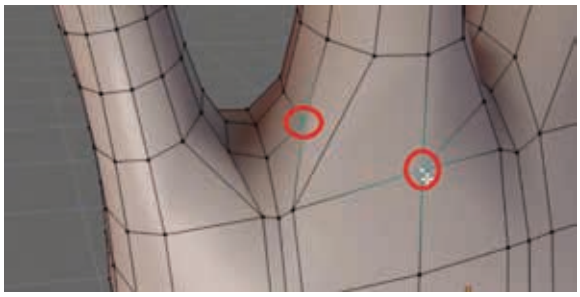
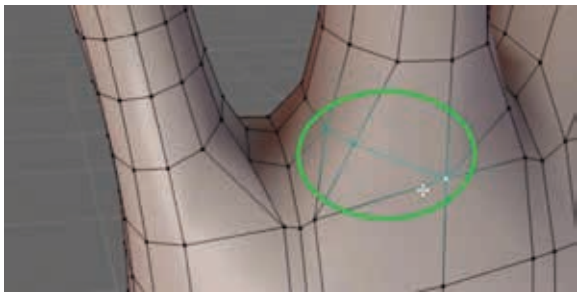
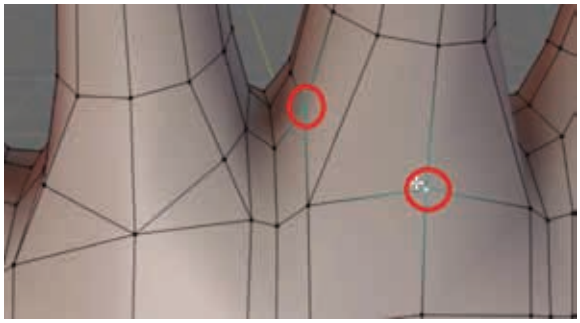
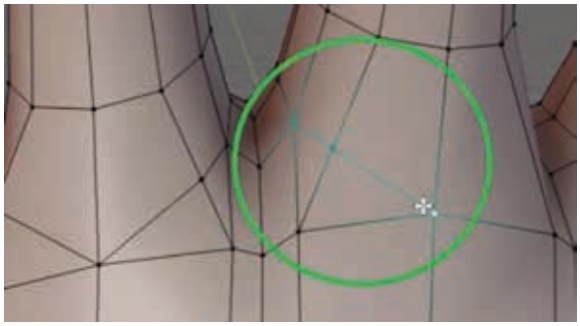
When modeling hands it's easy to get too many edges from the fingers to the hand. Let's see how we can do some retopology and fix things.

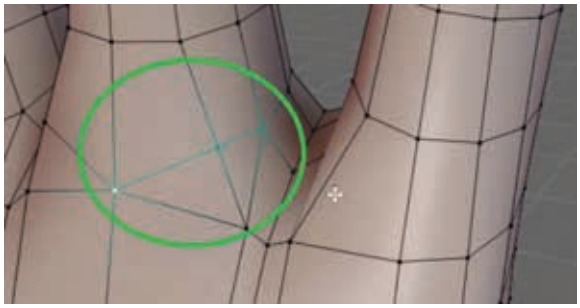
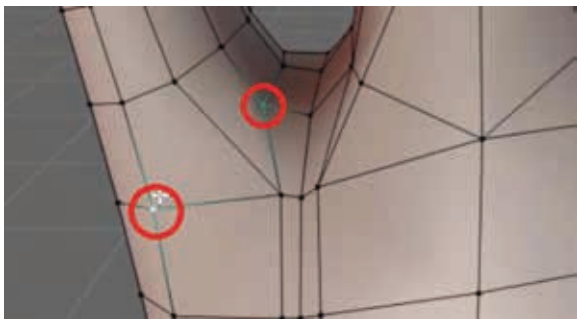



First, connect these vertices by selecting and **pressing J**.

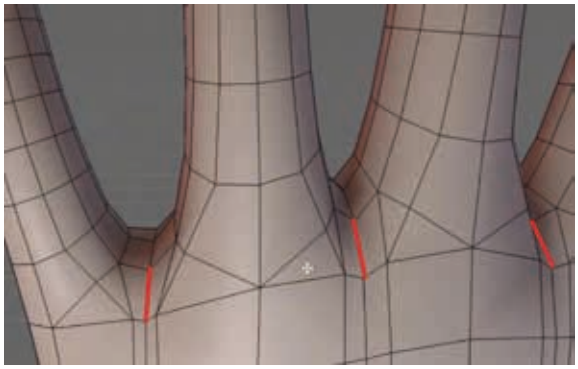
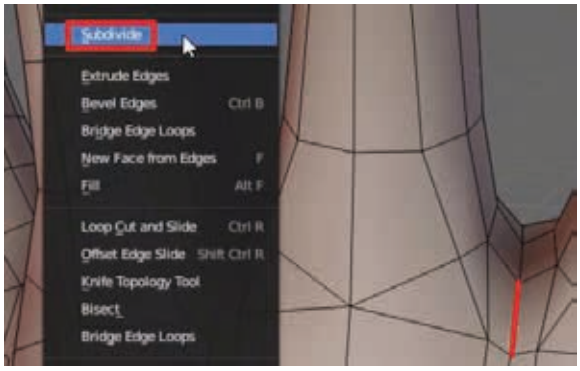



Repeat for all of the similar cases.

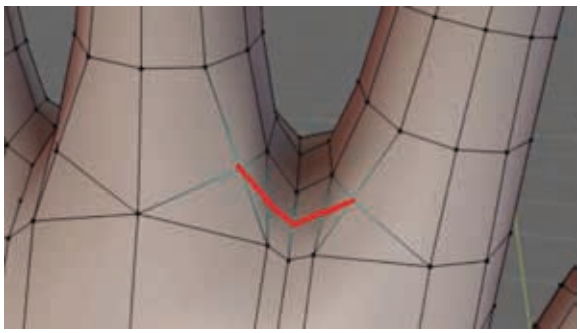





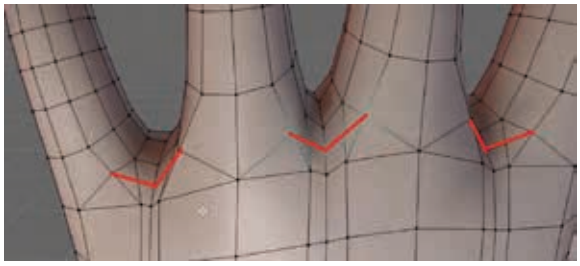
Select these edges and add 1 extra vertex by subdividing them (**Ctrl+E** for Edge menu).

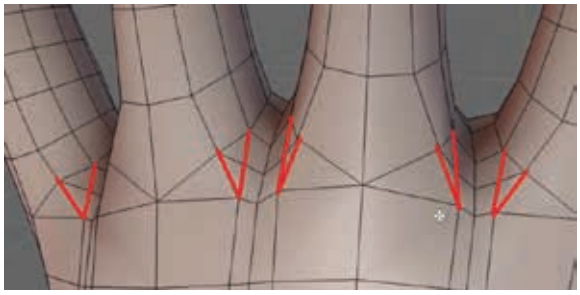
This subdivision creates an extra vertex in the middle of these edges. You can use it to bridge by selecting the vertices shown below and **pressing Ctrl+J**.



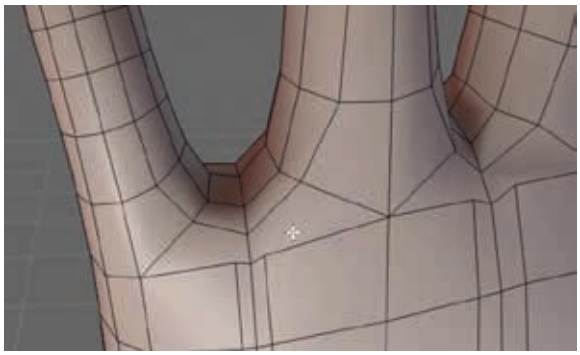
Repeat this between the other fingers as well.



Now we select these edges that are no longer needed.

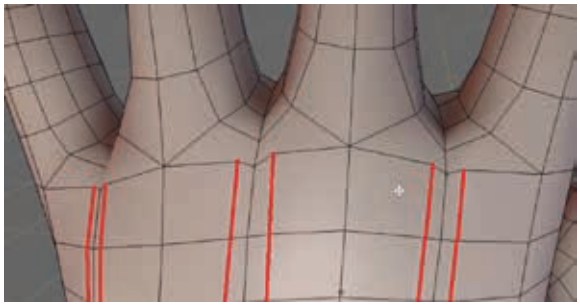


Delete them by **pressing Ctrl+X**.

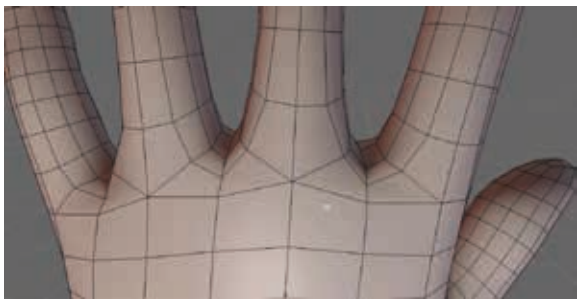


Hold Shift and Double-click on these unnecessary edges to select them and **Ctrl+X** to delete.

Pressing Ctrl+X instead of X to delete skips the menu where you can select what to delete, so it's a bit faster.



Now the finger edges flow back on themselves, keeping the hand polycount low.



Retopology tips - Settings

Before we can start with retopology, there are some settings that will make it easier and let us work faster.

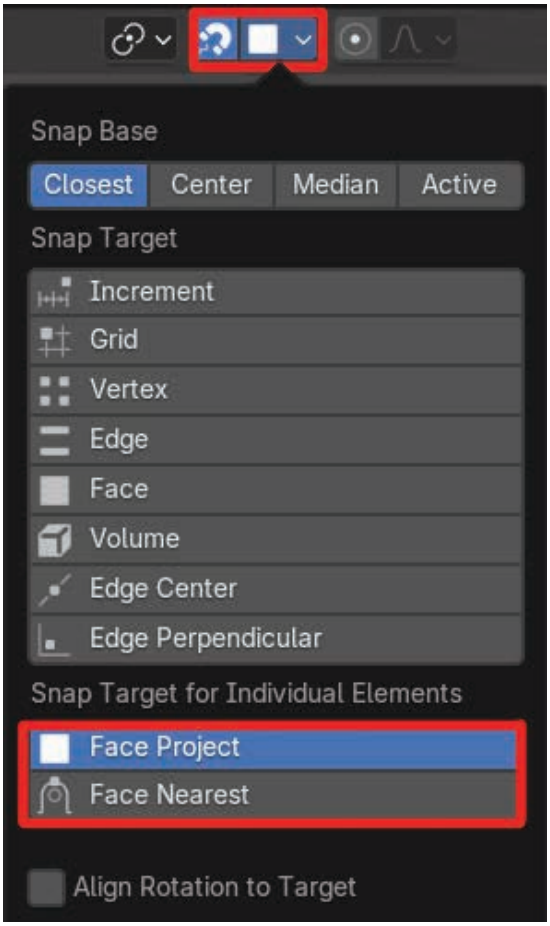
First of all, click on this magnet icon to enable **Snapping**.

Set a Snap Target for Individual Elements: Either **Face Project** or Face Nearest.

Face Project is the best choice in the beginning stages or retopo.

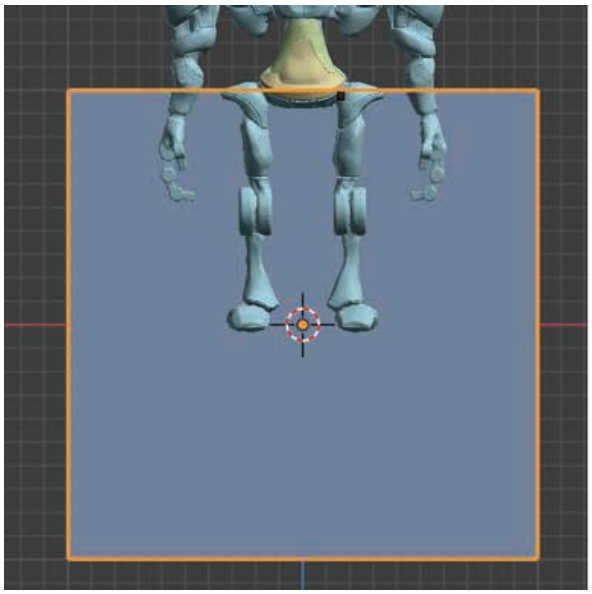
Unless you try to snap things around a round surface. In that case, Face Project might distort the projection of Vertices and you can try Face Nearest for better results.

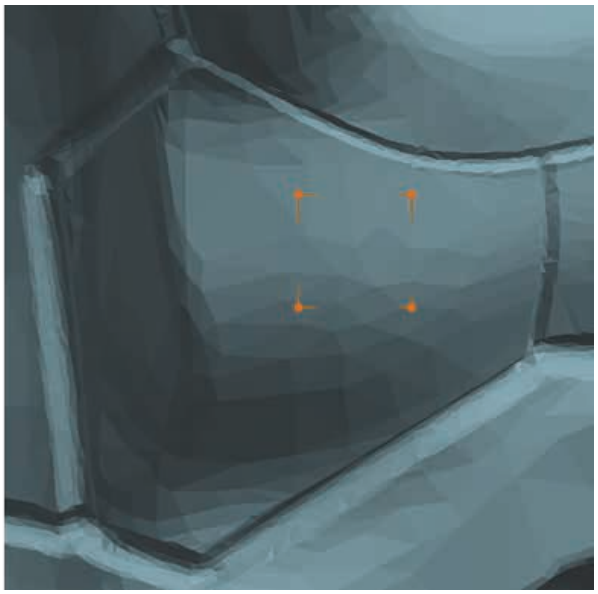
Now, our new geometry will snap to the surface of the sculpted model.



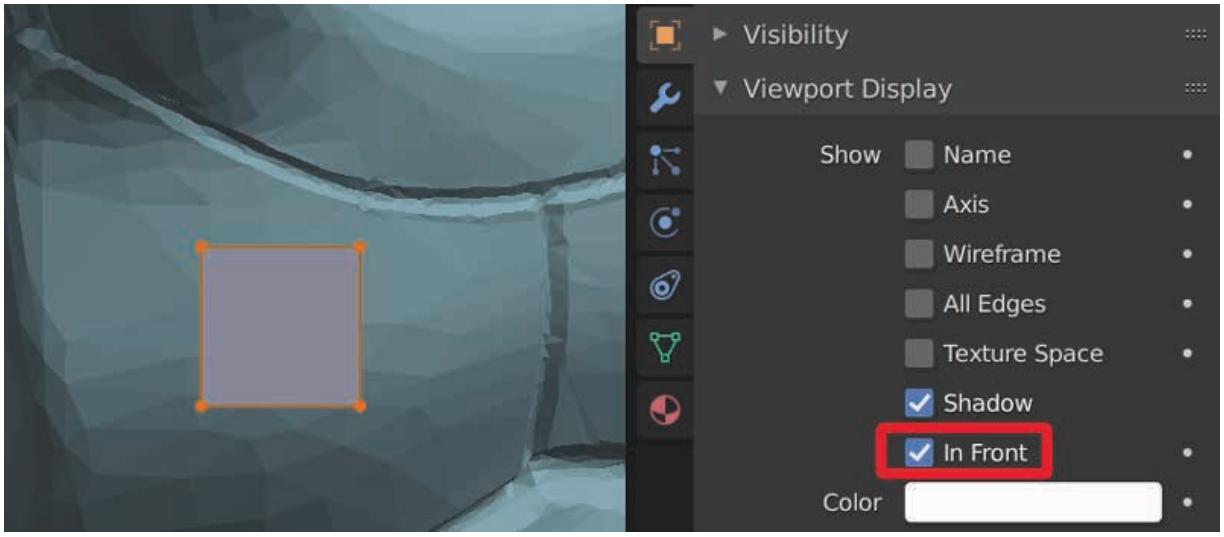
Go to Front view by pressing 1 on the numpad. **Press Shift+A** and add a **Plane**.

Then **Press TAB** to enter Edit Mode and rotate the Plane by 90° by **pressing R+X+90**, and scale the plane down by **pressing S** and moving the mouse cursor towards the plane. Move the plane up to where you want to start retopologizing on the sculpt.



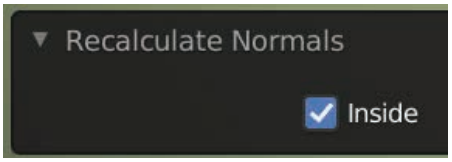


As you can see, it snaps to the surface but it's mostly hidden by the sculpture geometry. Check **In Front** to fix this.



If that doesn't work, **press Shift+N** to flip the normals.

You may have to check **Inside** if nothing happens.

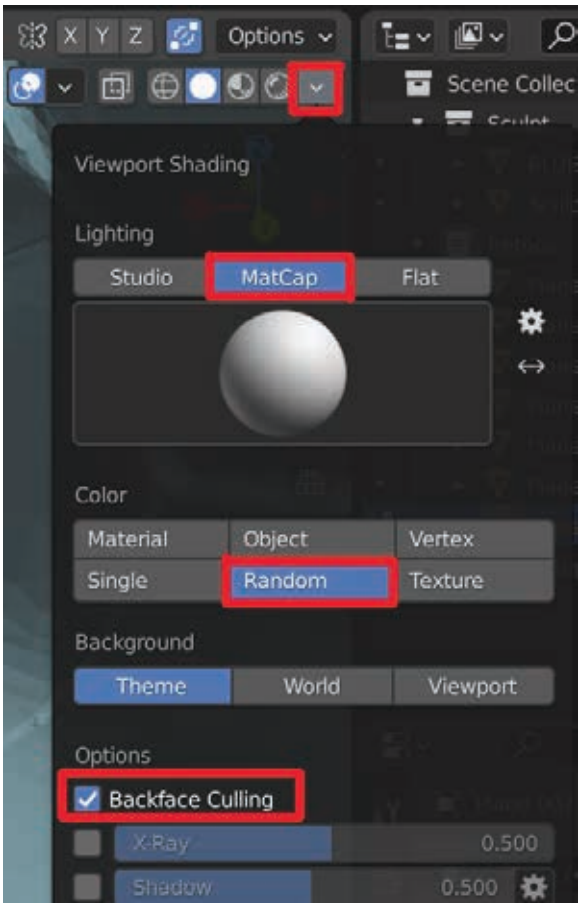


However now when rotate around the sculpt, you can still see the plane from the back.

This will get confusing, so turn on **Backface culling** so you don't see it from the other side.

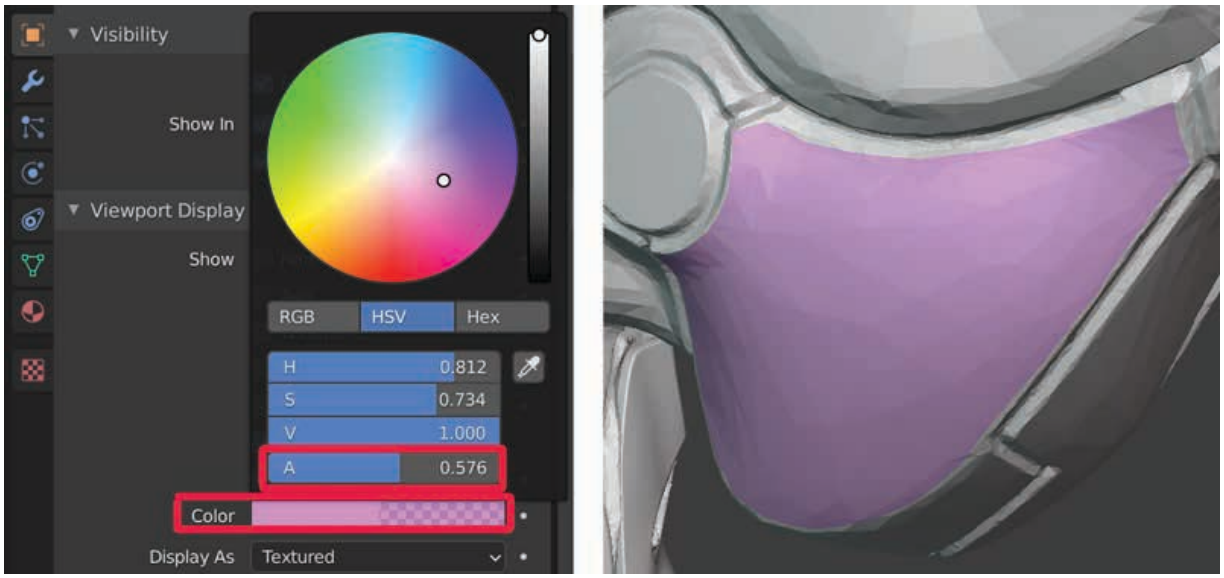
While you're in the Viewport Shading menu, turn on **Matcap** and **Object color**.

This makes it easier to see what you're doing.



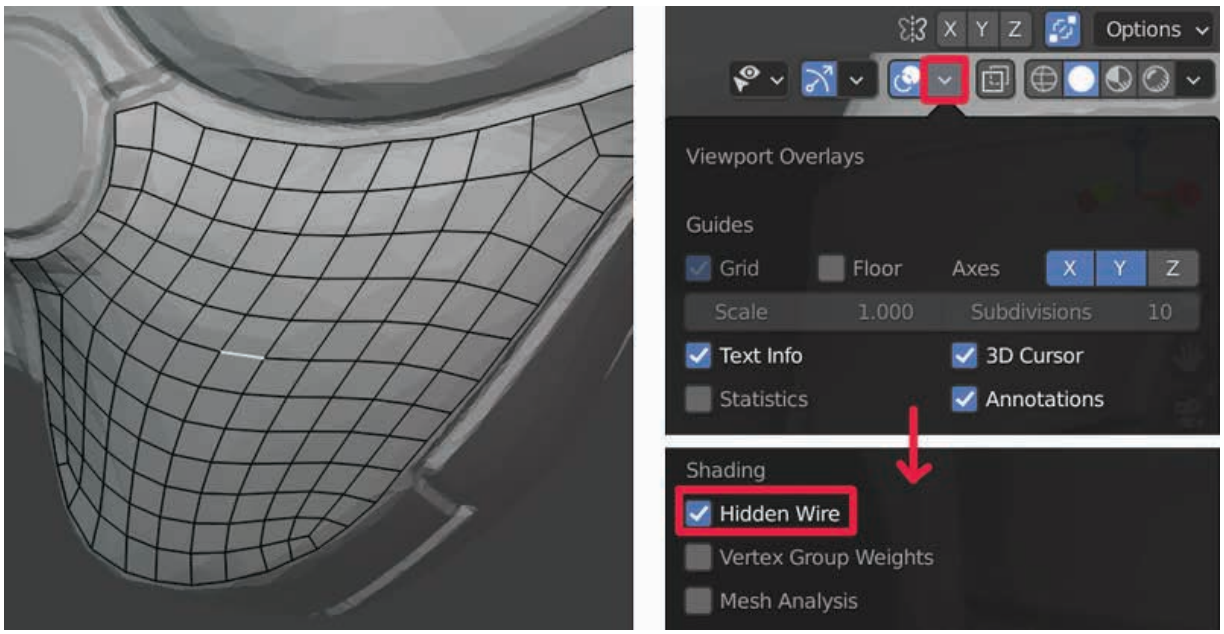
It's tempting to set color to **Random**, but by setting it to **Object** we actually can decide the color ourselves.

And by lowering the **A value (Alpha)** we can actually make the retopo mesh semi-transparent in the viewport, which is quite handy as we can still see the mesh underneath.

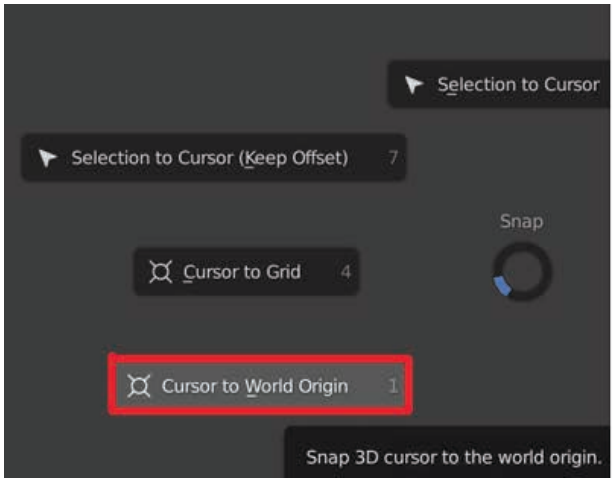


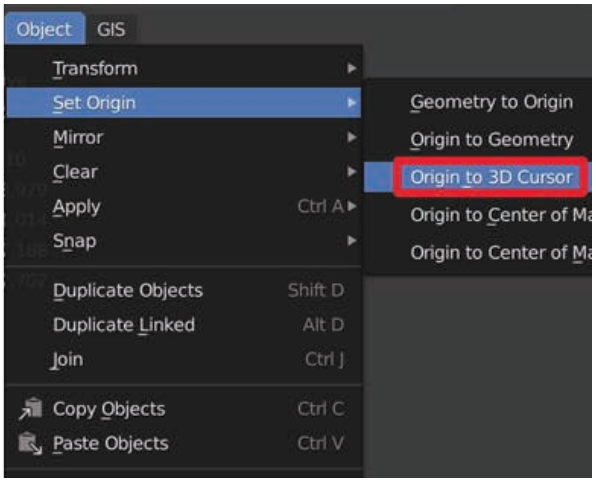
If at any point you'd like to only see the wireframe in Edit Mode, you can enable the **Hidden Wire** option. This makes the Faces invisible, showing only the wireframe of the mesh so you can see "through" it.

You can add this to your **Quick Favorites** if you don't want to have to dig into a menu each time.

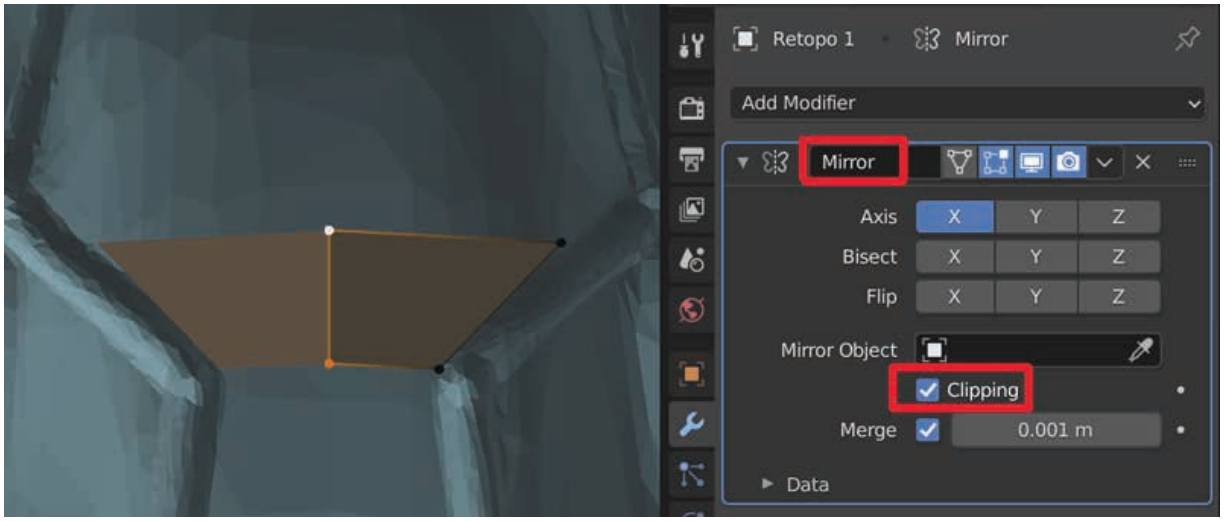


Let's add a Mirror modifier. But first, to make sure our retopology is mirrored, **press Shift+S** and set the 3D Cursor to the world origin and then in Object mode set the Plane's origin to the 3D cursor.



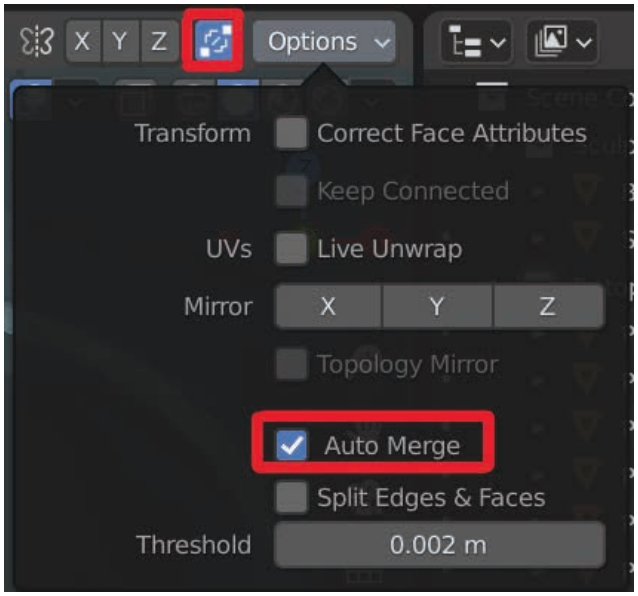


Now add a **Mirror modifier** and check **Clipping**, so it will merge automatically where it meets the other half in the middle.



Finally, turn on **Auto Merge** - this way vertices are merged automatically when they touch.

You can enable this either by clicking on its icon, or by checking the Auto Merge option.

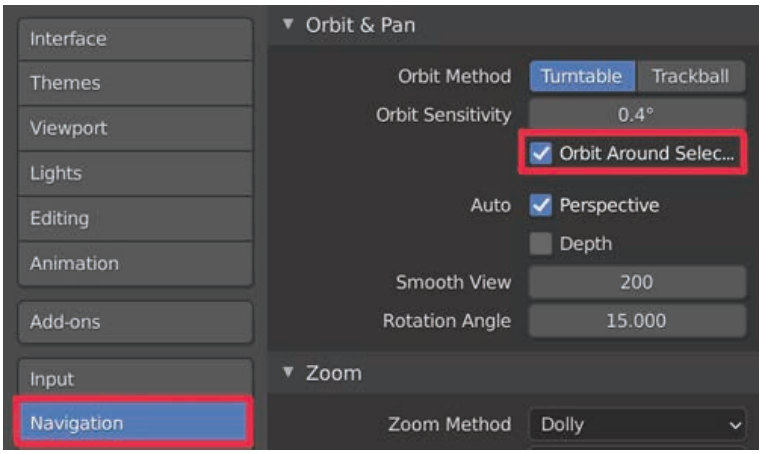


Click here to return to the index.

In Preferences under Navigation, check **Orbit Around Selection**.

This is helpful with Retopology, so you can always be focused on the part you're working on.

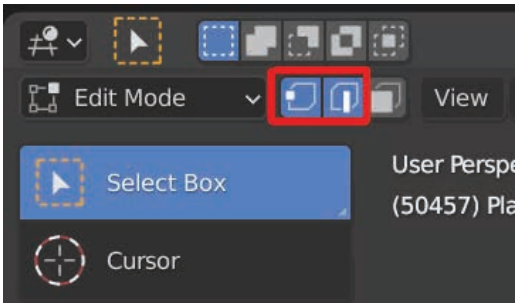
Try it, and see if it works for you.



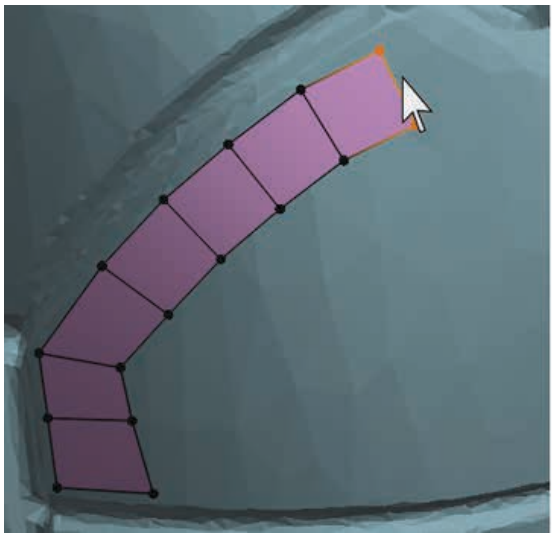
Retopology tips - Creating geometry and the F2 add-on

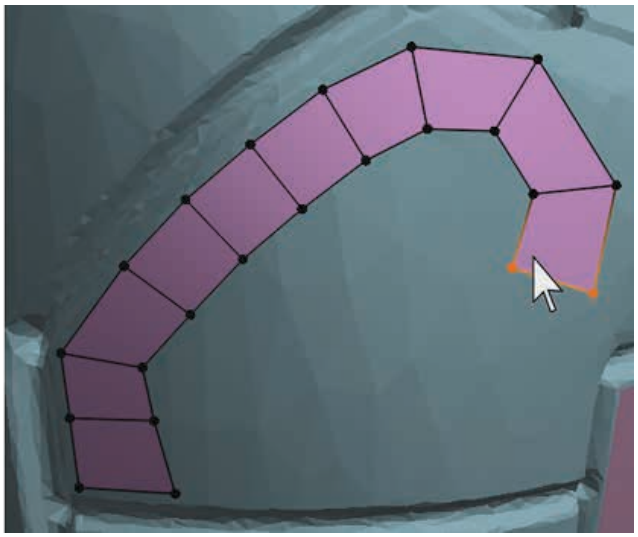
During retopology you often have to select Vertices and Edges. By **pressing 1,2 or 3** you can quickly choose vertex, edge or face select mode.

Hold Shift and click on vertex and edge selection mode so you can select both a single vertex and the two vertices that make an edge.

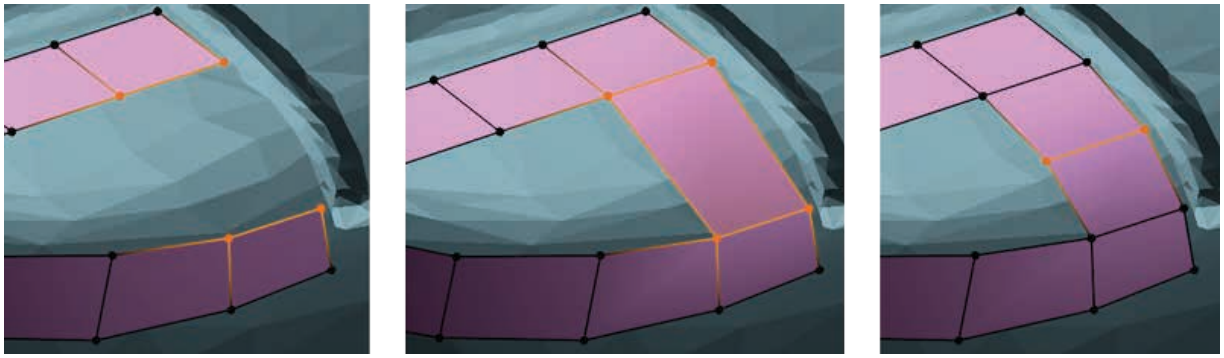


You can extrude by **pressing E**, or even faster by **holding Ctrl and right-clicking**. The benefit of this method, besides that it's very fast, is that it actually rotates the new edges based on where you click.

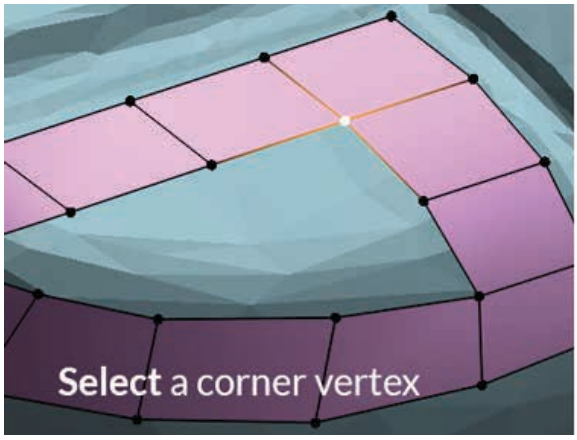


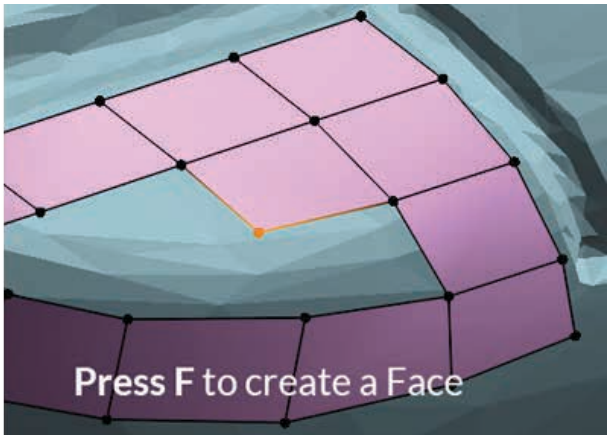


Some more essential retopology keyboard shortcuts are: **F to fill** between two edges and of course **G to move, S to scale and R to rotate**. In the below example, I select two opposing edges and **press F** to fill, then add an edge loop in between with **Ctrl+R** and **press G** to snap it to the surface.



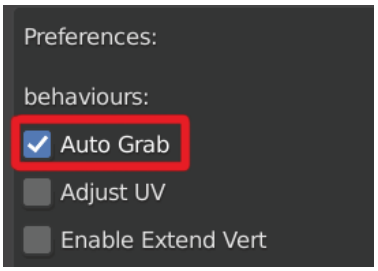
When you enable the **F2 add-on** that comes with Blender in Preferences (in the **Get Extensions** tab), **pressing F** from a corner vertex creates a face.



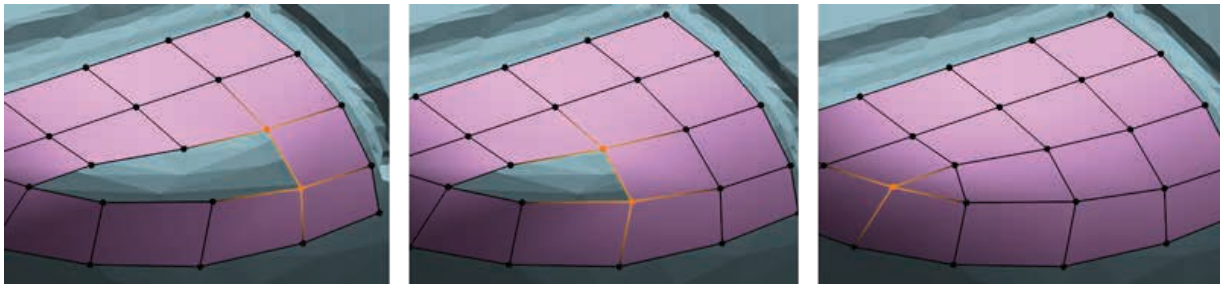


I recommend having the setting **Auto Grab** turned on in the **F2 add-on settings** in Preferences. That way, the corner vertex of the new Face will be selected automatically, so you can immediately move it.

With retopology, reducing unnecessary clicks is smart.



Pressing F starting from an edge between faces adds faces. **Press F again** as many times as you need. In the below example, I pressed F three times to add three faces.

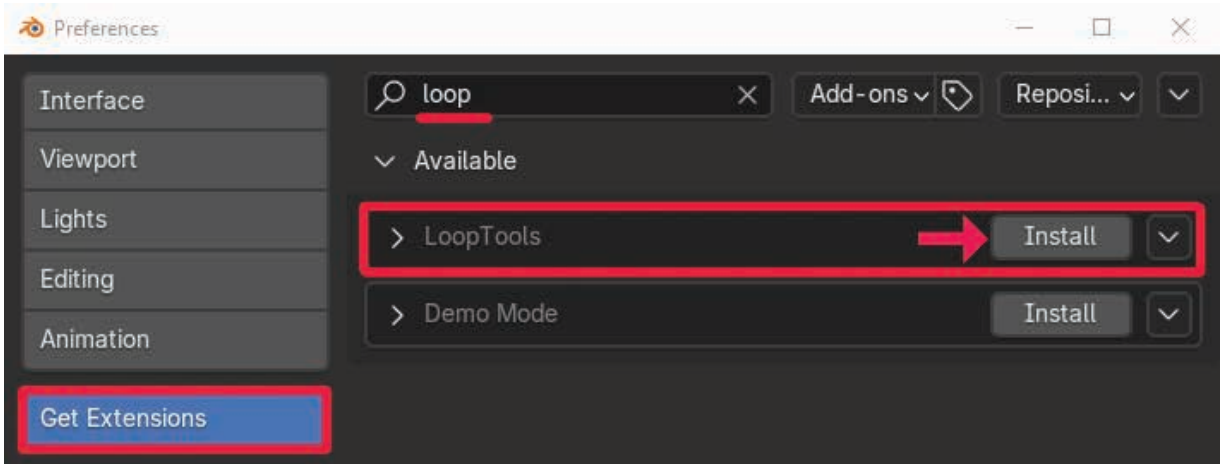


Slide edges or vertices by **pressing G twice**. In combination with Auto Merge this can be very convenient.

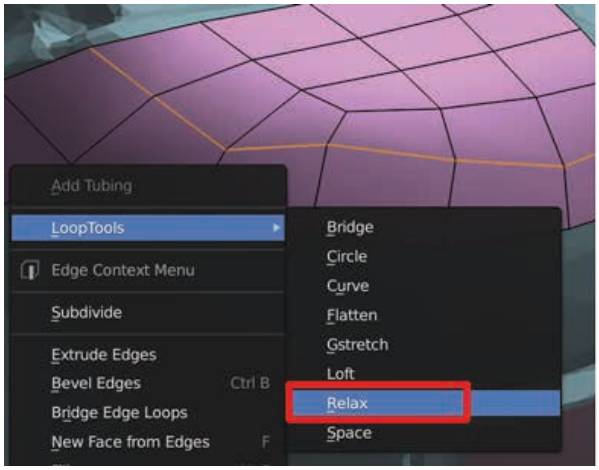
Retopology tips - LoopTools and Zbrush-like Smoothing

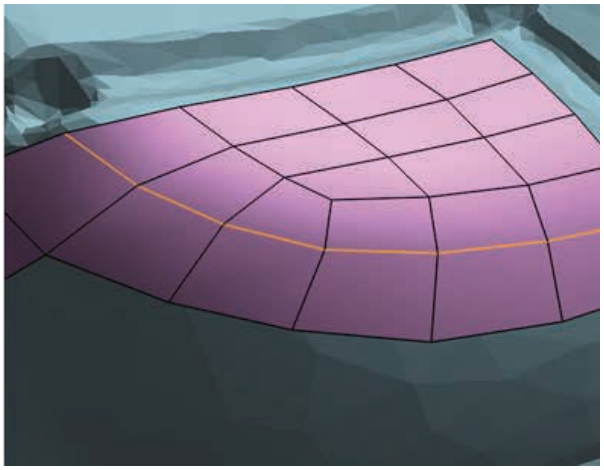
LoopTools is another very helpful add-on for modeling which ships with Blender.

After enabling the LoopTools add-on you can find it by **right-clicking** in Edit Mode.

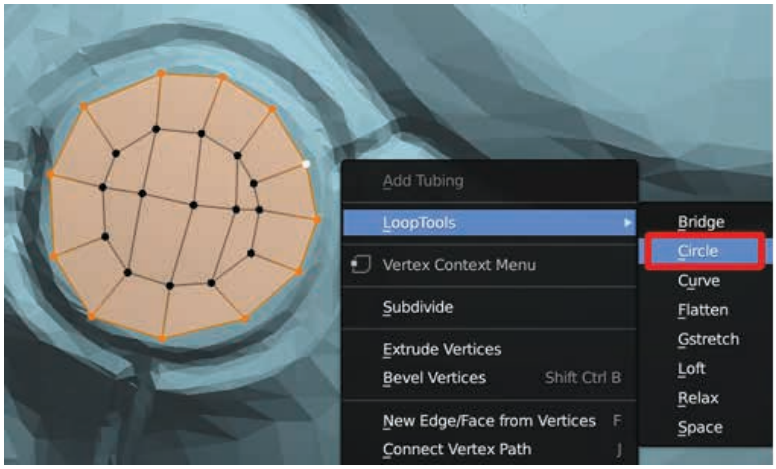


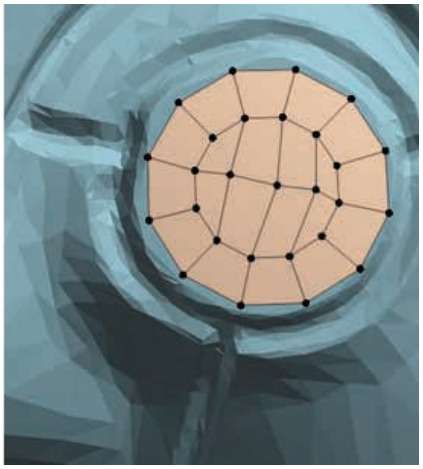
If you have a row of vertices that doesn't have very nice flow, you can improve it by using the **Relax** function. **Press Shift+R** to repeat.



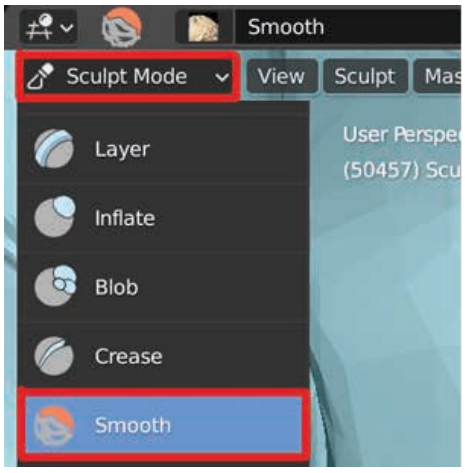


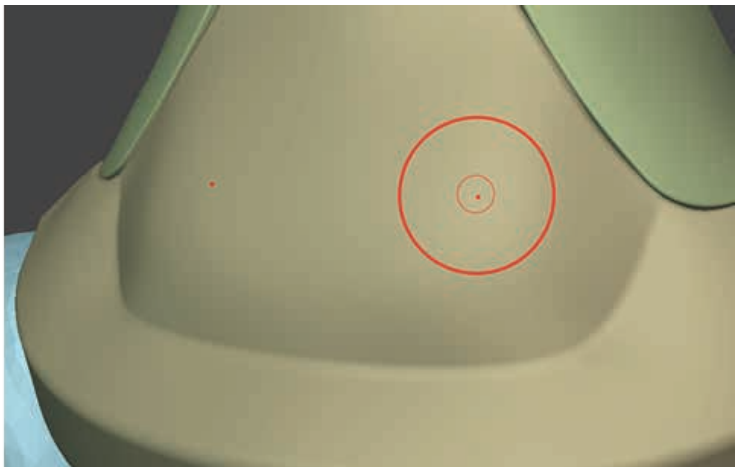
To make a selection perfectly circular, use the **Circle** function. Alternatively you can **press Shift+Alt+S** to make a selection circular (or Spherical, if you have more than a ring of vertices selected).



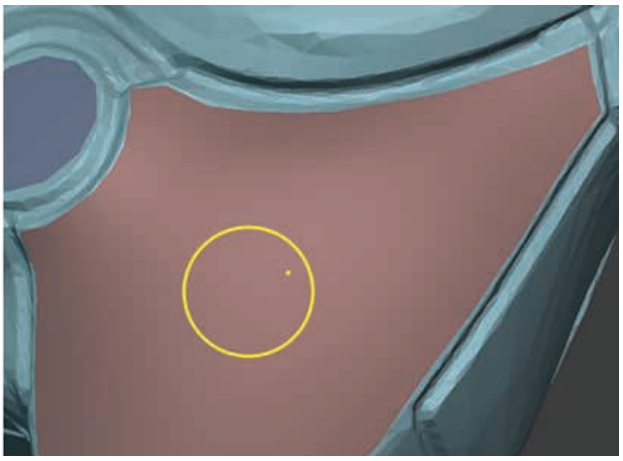


To relax or smooth part of the geometry, turn of Snapping and use the **Smooth brush** in Sculpting mode (either by selecting the actual Smooth brush, or **hold Shift** using any other brush).

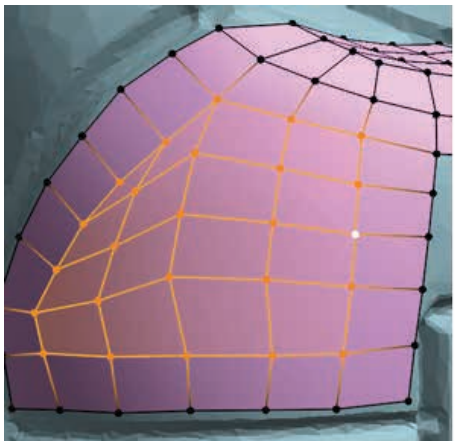


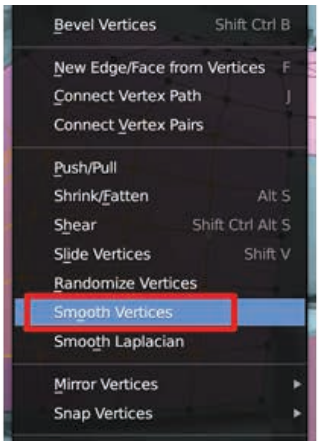


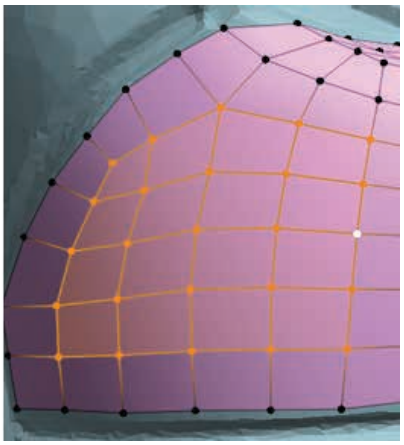
For a Zbrush-like alternative Smoothing brush, use the **Slide Relax brush** and **hold Shift** to get an alternative Smoothing method, that keeps the volume of the geometry intact.



Another way to smooth an area, for instance after you've used Grid Fill, is to use **Smooth Vertices**. **Press Shift+R** a couple of times to repeat the smoothing, until it looks good. **Press G** to snap the selection to the underlying surface.



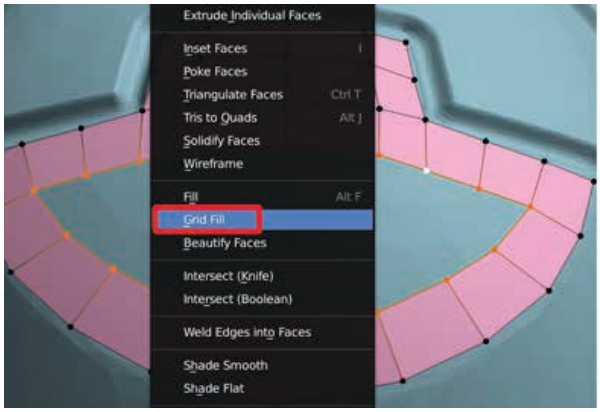


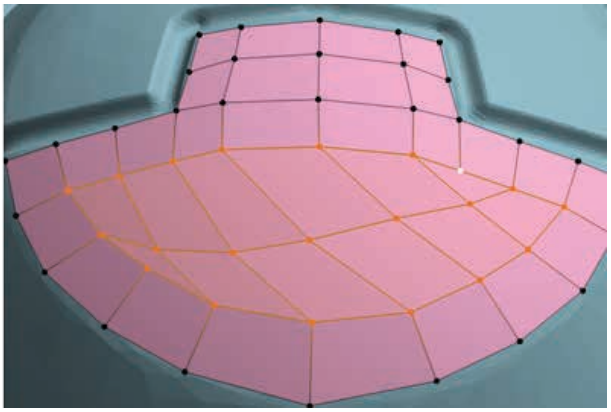


By the way, don't forget to use those matcaps! They're great for finding bumpy areas.

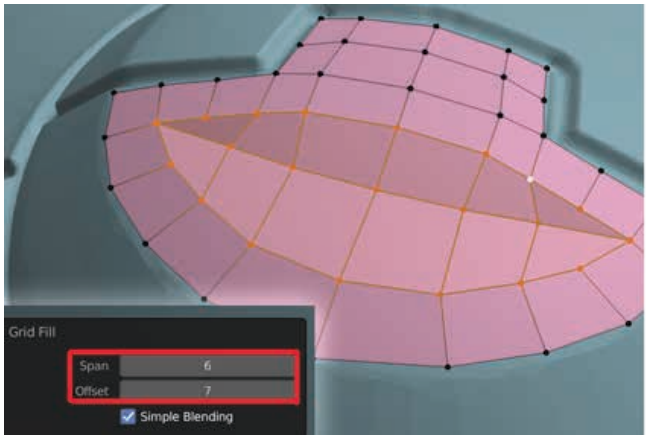
Retopology tips - The magic of Grid Fill

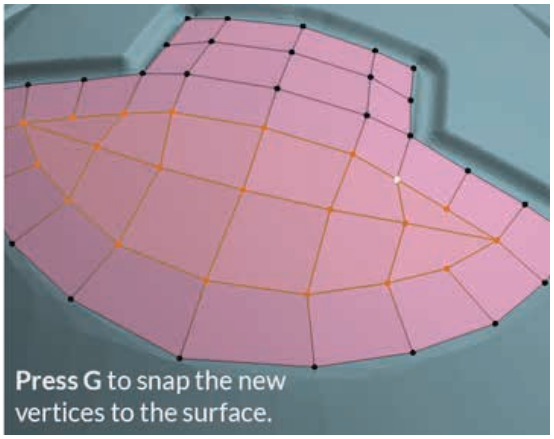
Grid Fill quickly fills the empty space between the loops you've made. **Press Ctrl+F and then G** to activate it or find it in the Face drop-down menu in the 3D Viewport (it's only there in Edit Mode).



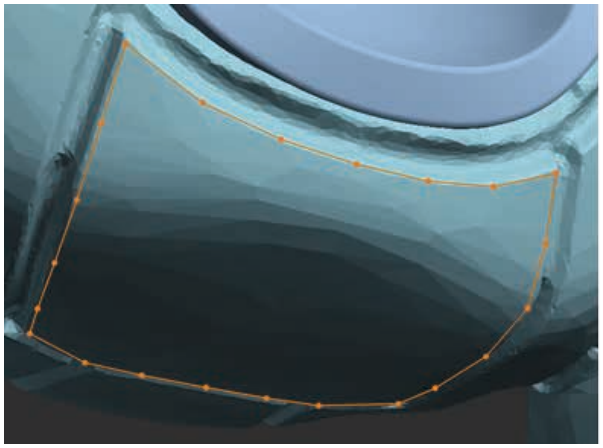


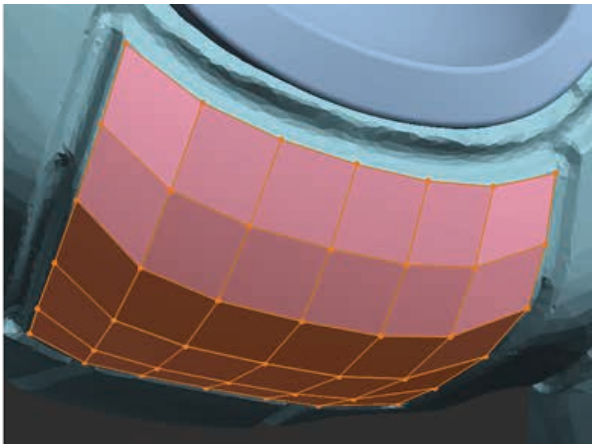
The **Span** value controls the amount of Faces used and **Offset** rotates the geometry. **Press G** to snap the newly added vertices to the surface.





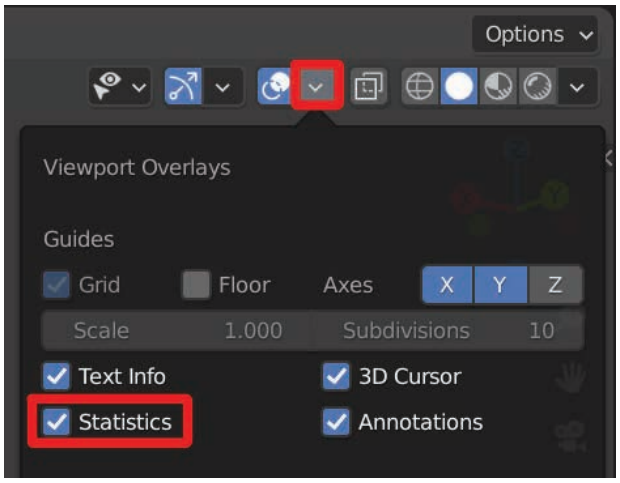
Amazingly, Grid Fill guesses the curvature that the surface of your retopology should have. So as long as you trace the outline by extruding an even amount of vertices, Grid Fill will do the rest (in the case of simple panels like the example below).



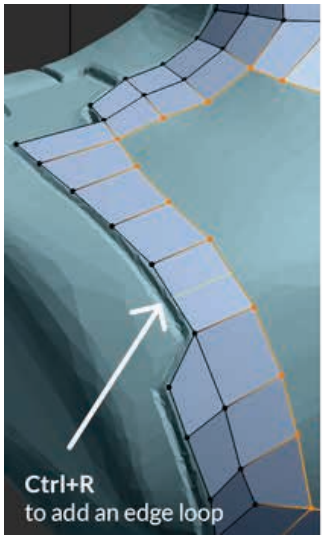


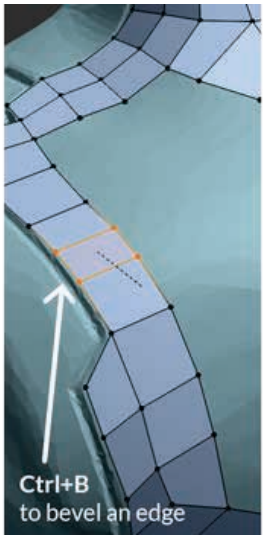
Grid Fill, or indeed any method to fill an area with quads, needs an even amount of vertices and edges to work, otherwise it will give you an error message.

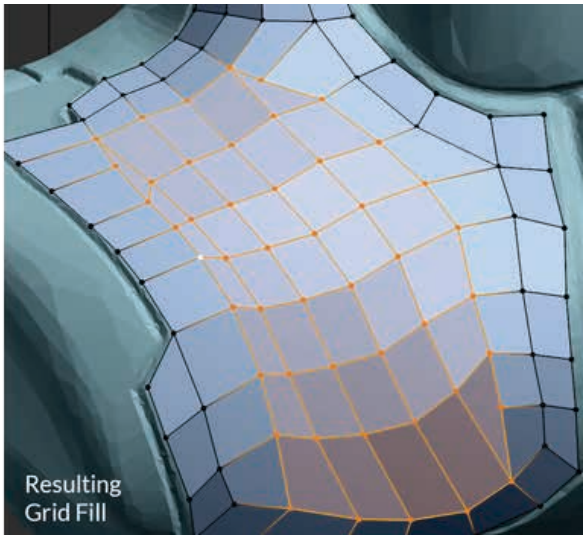
If you have no idea how many vertices are in the selection, turn on **Statistics** in the **Viewport Overlays**.



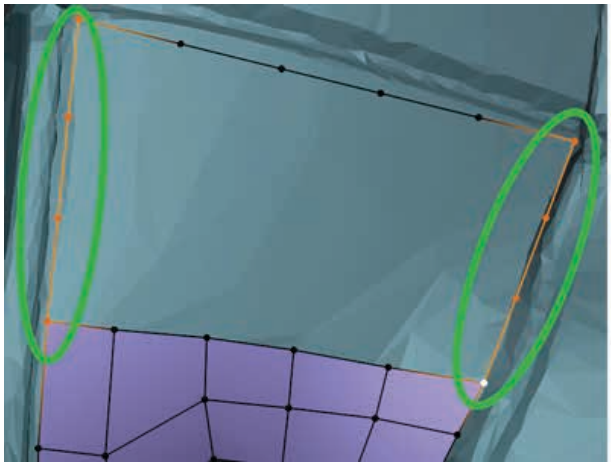
If you have an uneven number, you can add a Loop Cut with **Ctrl+R** or bevel an edge with **Ctrl+B**, then try the Grid Fill again.

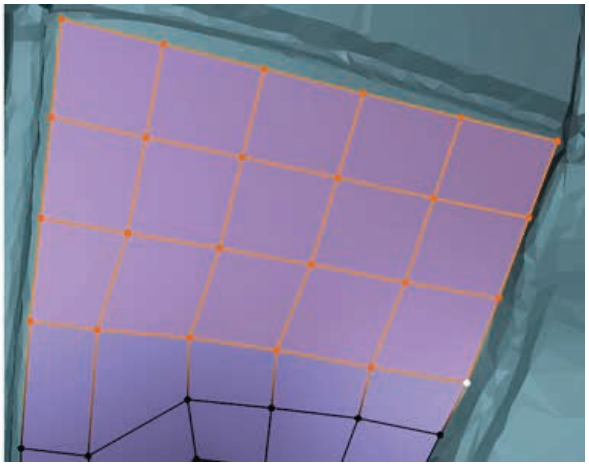






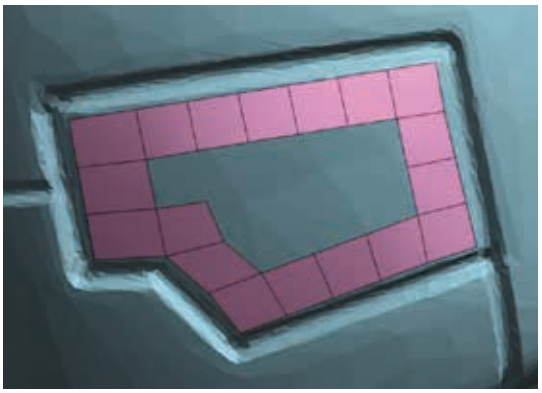
The best case scenario with Grid Fill is when you've got two opposite sides with an equal amount of vertices. In that case, just select the two sides instead of the entire loop. In this case you don't need to experiment with the Span and Offset options, it will just be perfect straight away.



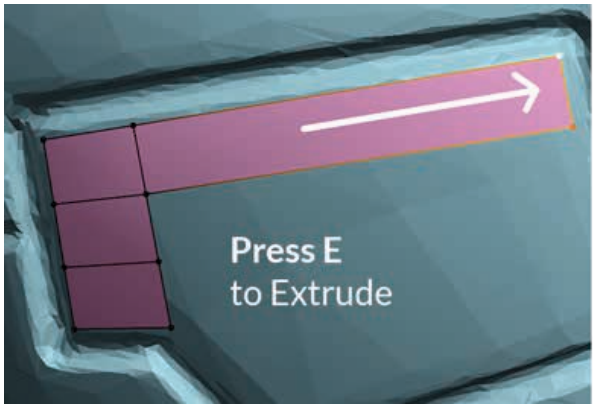


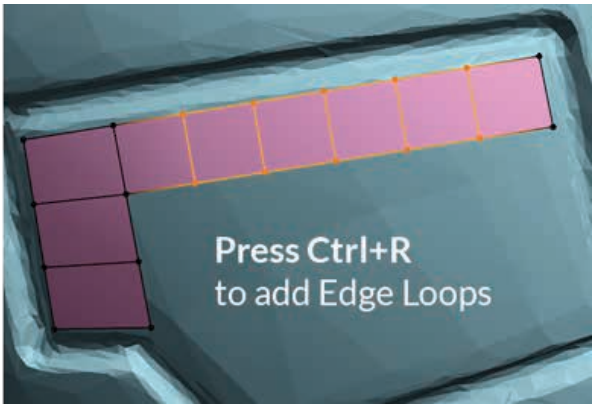
Retopology tips - How to retopo

We finally know all the right settings and tools for retopo... now what? In a nutshell: **Start by creating the important loops**, then fill the gaps. With these robot panels, that mainly means starting with the outline of the panels, but with a character it can be several loops: around the eyes, the mouth and other important features. Then you connect those and fill in the gaps. It may sound boring, but it's actually kind of a fun puzzle! No, really...

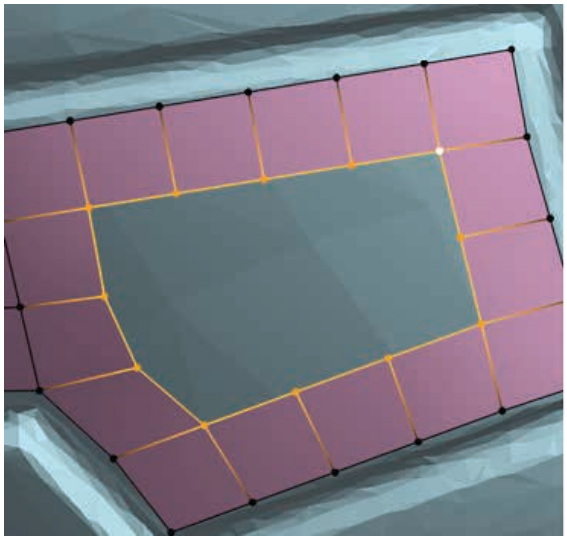


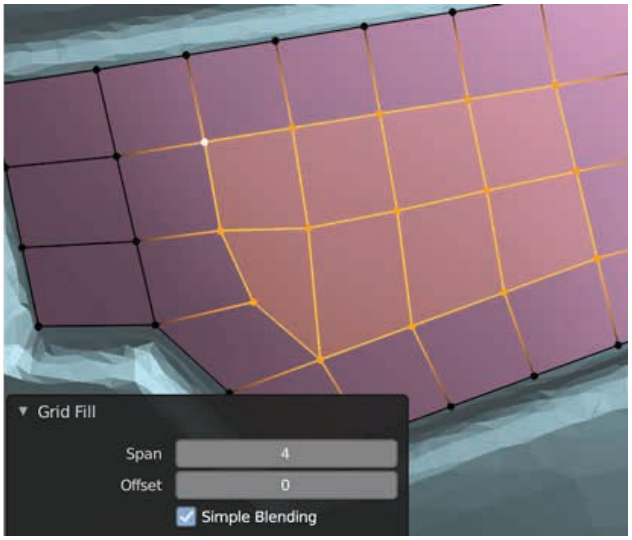
A quick way to create loops is to **extrude (E)** an edge as far as possible, and then add edge loops to it by **pressing Ctrl+R**.



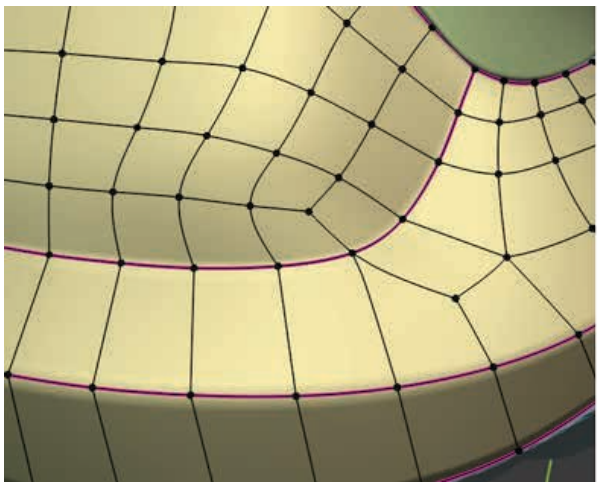


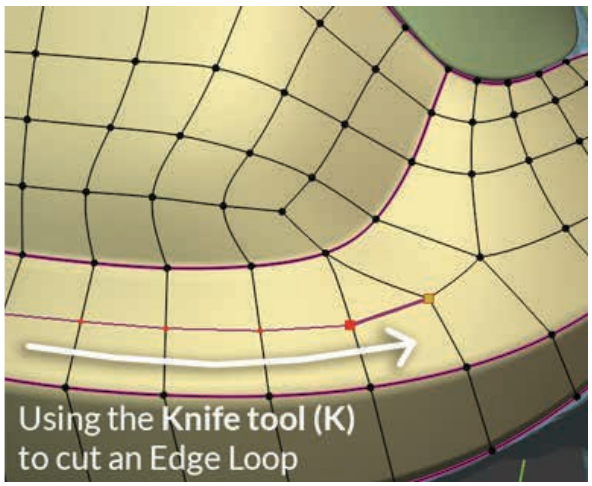
Then, fill up the space inside the loops. In the below example, **Grid Fill** (which is explained in the previous topic) did the job nicely. You may have to **press G** to snap the new vertices to the surface.



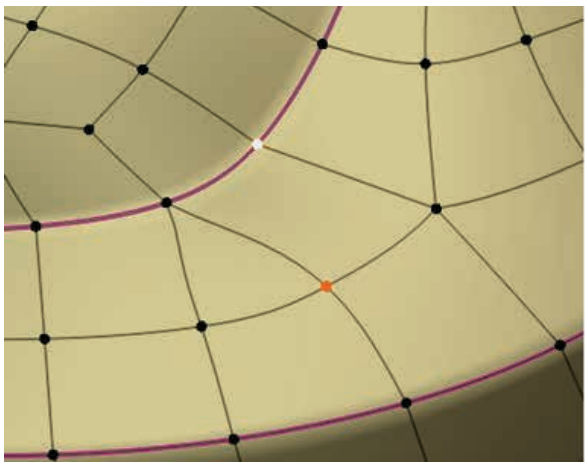


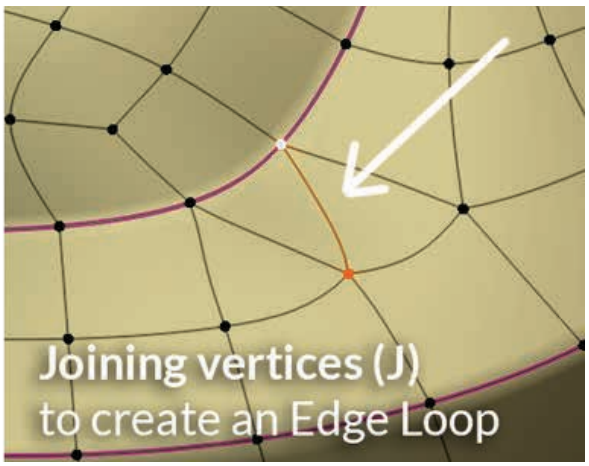
Sometimes, you will need to use the **Knife tool (K)** to cut new edges like in the example below:



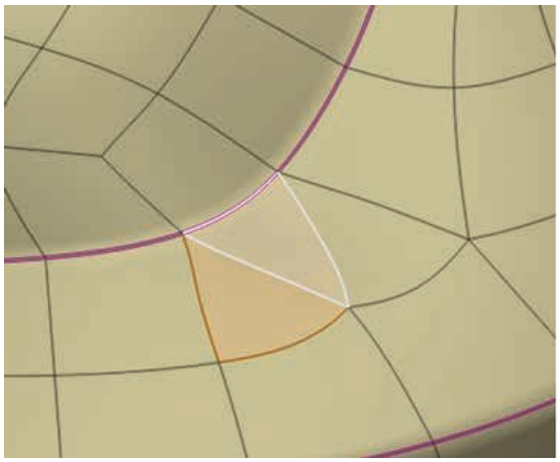


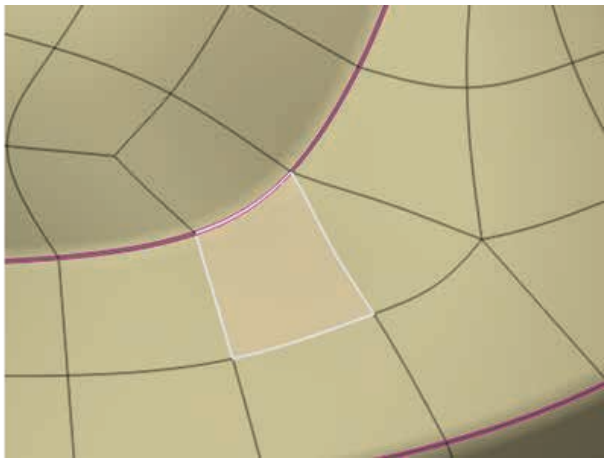
Or the **Join tool (J)**, also to create new edges by connecting vertices.



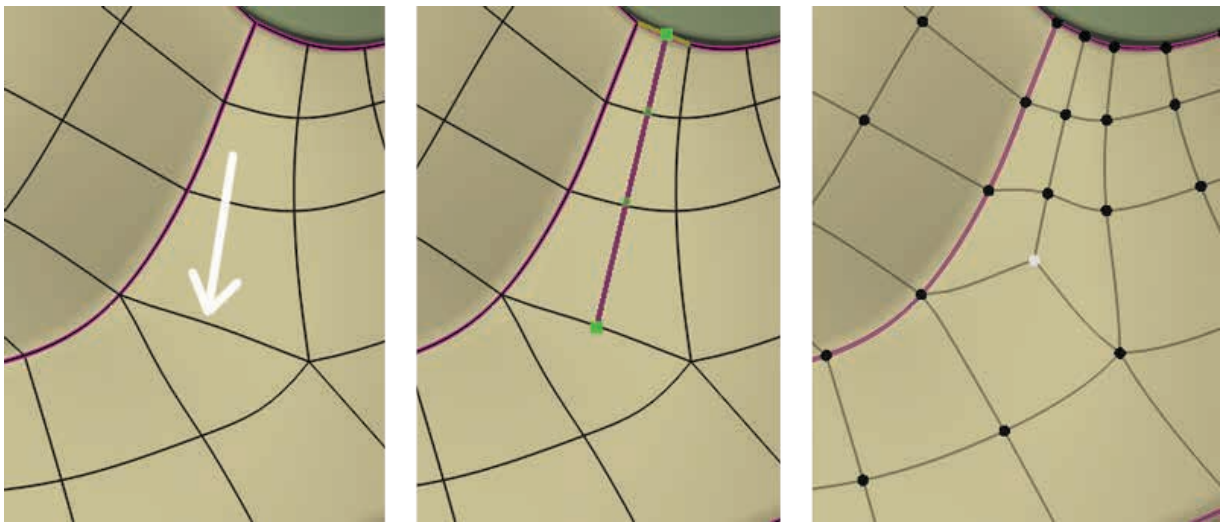


The point of joining those two vertices was to get rid of that triangle that was created by using the Knife Tool. But now I have two triangles instead! To get a perfect quad topology, you sometimes have to create some triangles temporarily. Remember how I said it was like a puzzle? Select to two triangles and **press F** to turn them into a quad. Alternatively, select the diagonal edge and **press Ctrl+X** to dissolve it.





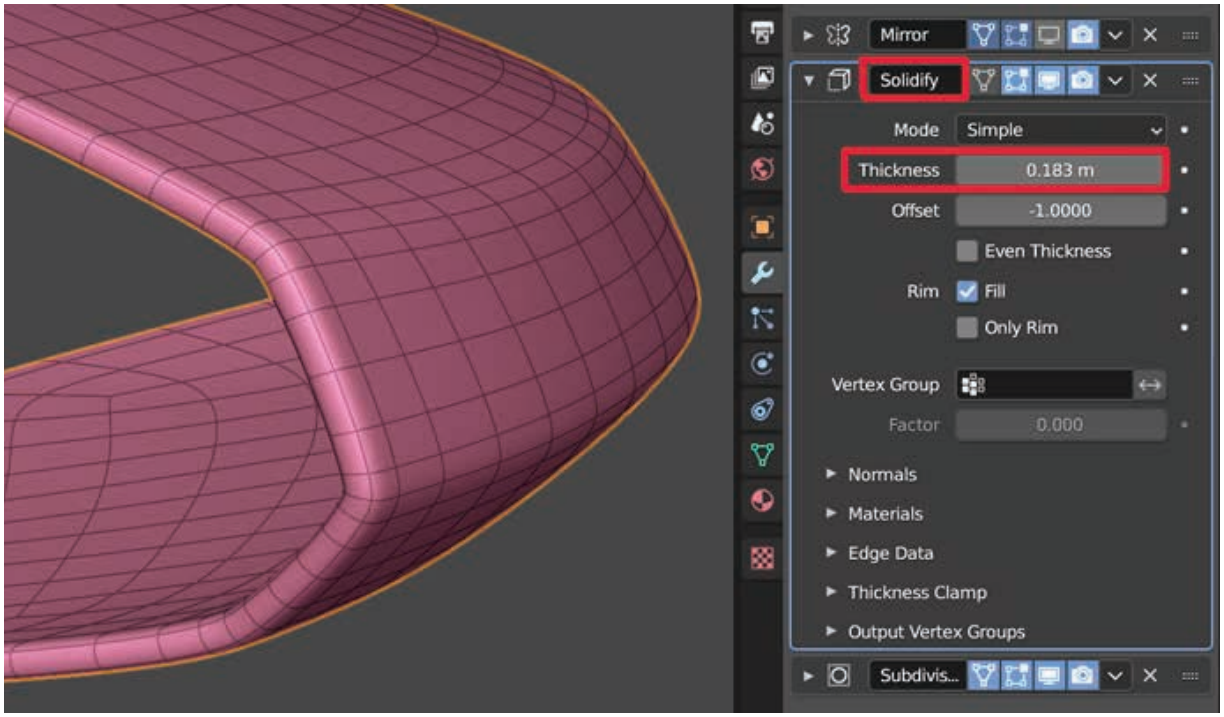
Nice. But… there's still one triangle! This one can be solved by using the Knife Tool again, and moving the vertex up.



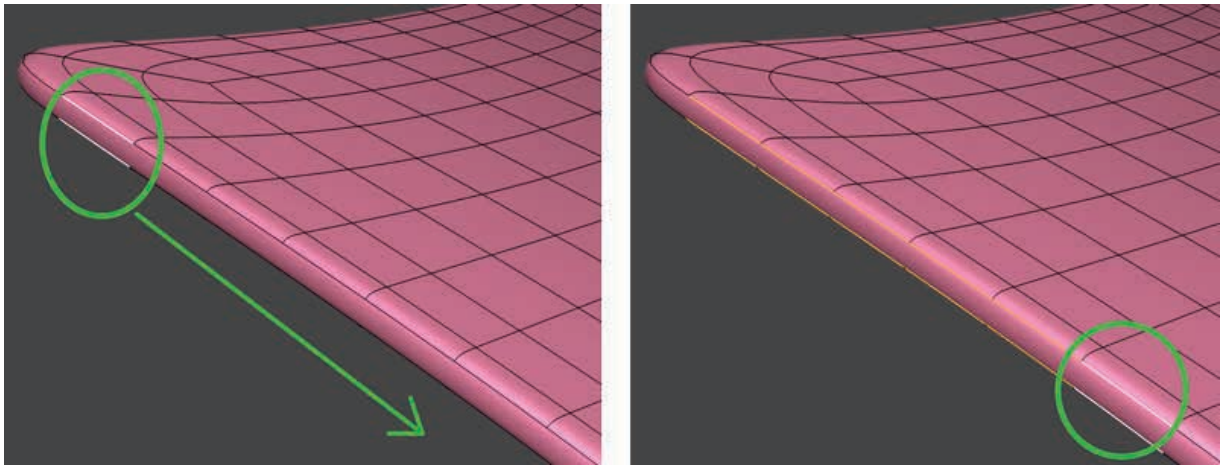
Press Shift+E to **crease** edges, set it to **Shade Smooth** and add a **Subdivision modifier** to see how your geometry is looking.

Bevel and Crease Hard Surface Panels

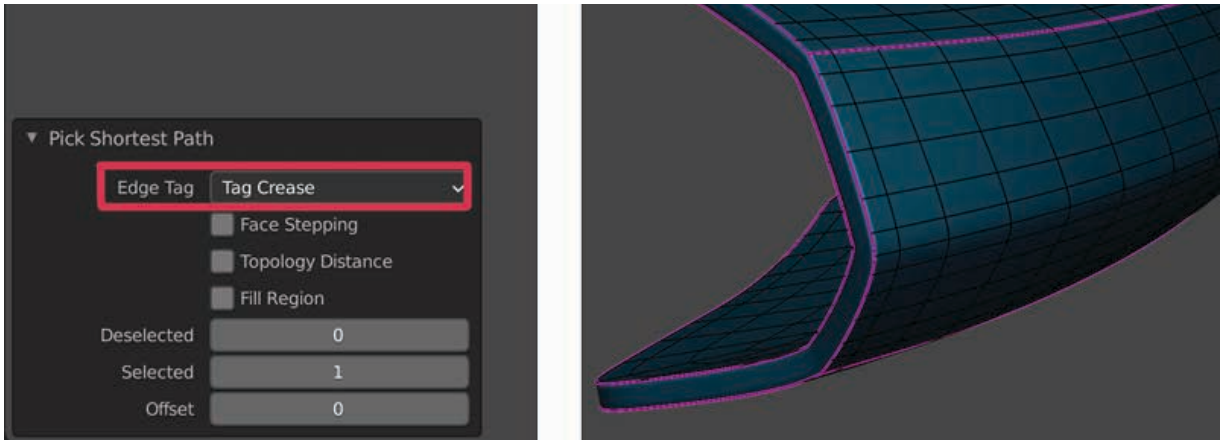
To control the roundness or sharpness of the edges of your panels, add a **Solidify modifier** and adjust the **Thickness**. Add a Subdivision modifier as well.



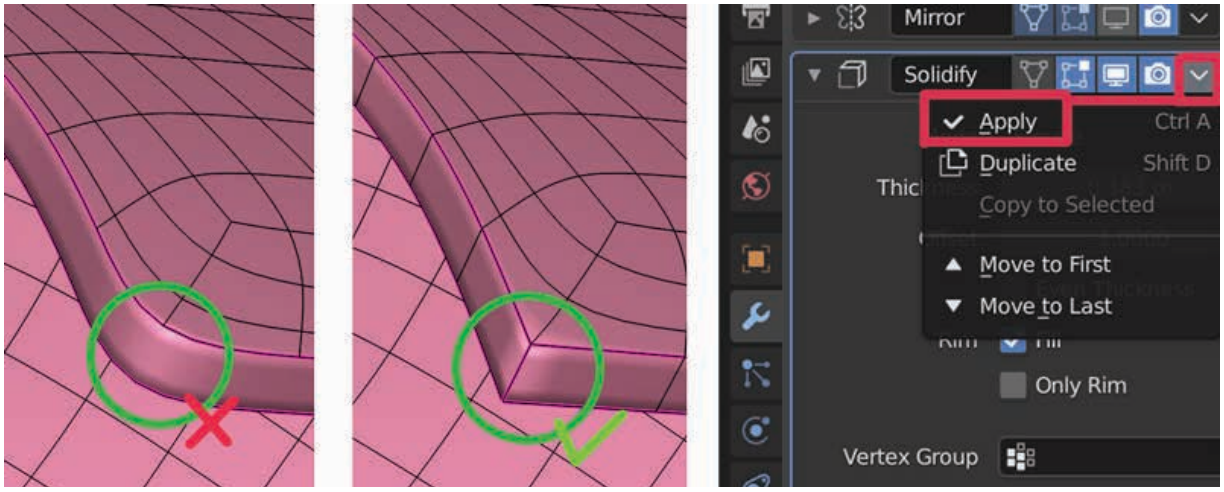
Creasing is a nice non-destructive method of controlling how the edges of the panels look. Select edges by holding Ctrl and selecting two edges by left-clicking at opposing sides. This selects the shortest path in between.



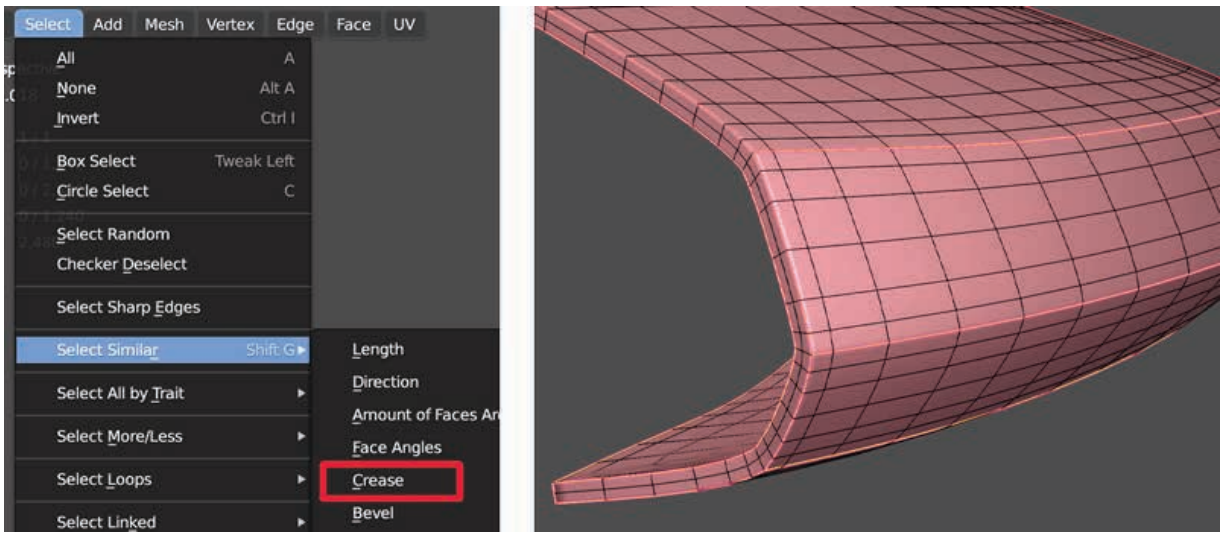
In the pop-up menu, set **Edge Tag** to **Tag Crease**. Now every edge selected with this **Pick Shortest Path** method will automatically be set to Crease. Of course in this case it's pretty easy to just select the edge loops by **holding Alt and left-clicking** on them.



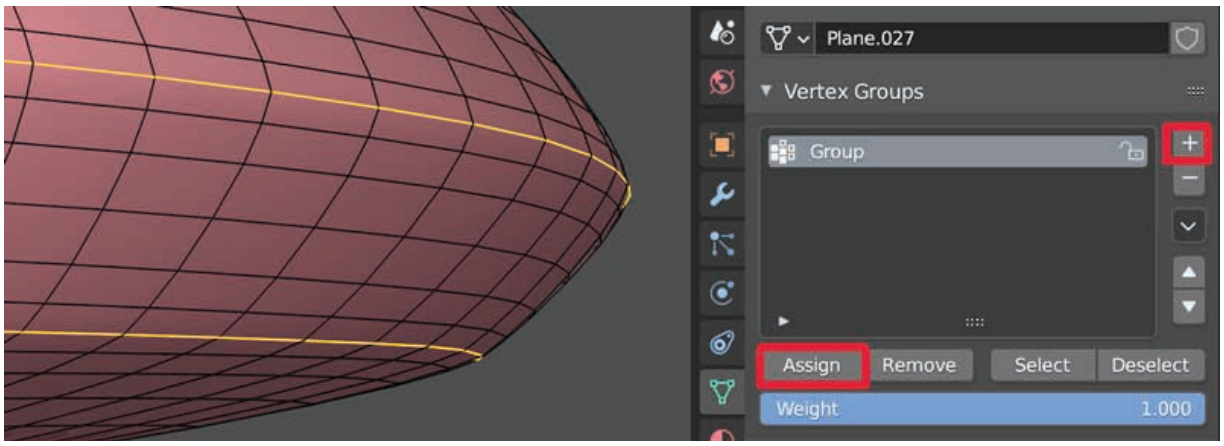
Alternatively you can manually select edges and **press Shift+E** and a value from 0 to 1 to crease those edges, or **press Shift+E** and move the mouse to crease. To be able to crease these corners, we have to **apply the Solidify modifier** in Object Mode.



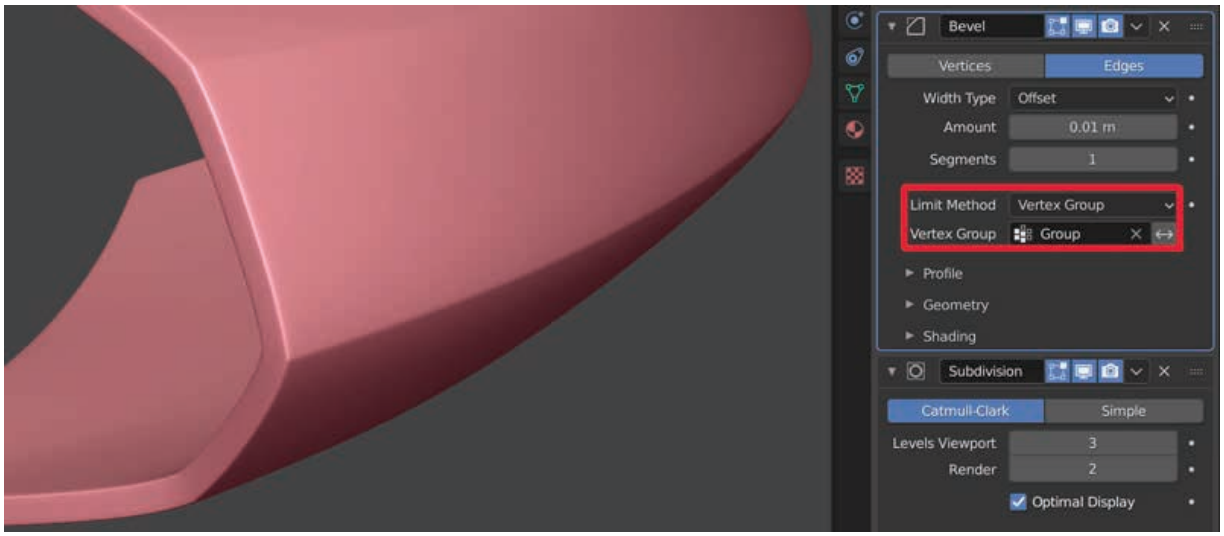
Instead of using Creasing, you can also use a **Bevel modifier**. If you were using Creasing previously, you can easily select all the creased edges by selecting just one creased edge and going to **Select > Select Similar > Crease**.



Then, **press Shift+E+[-1]** to remove the creasing as you will no longer need this when using the Bevel modifier. Since the edges are still selected, you can now easily add them to a **Vertex Group**.

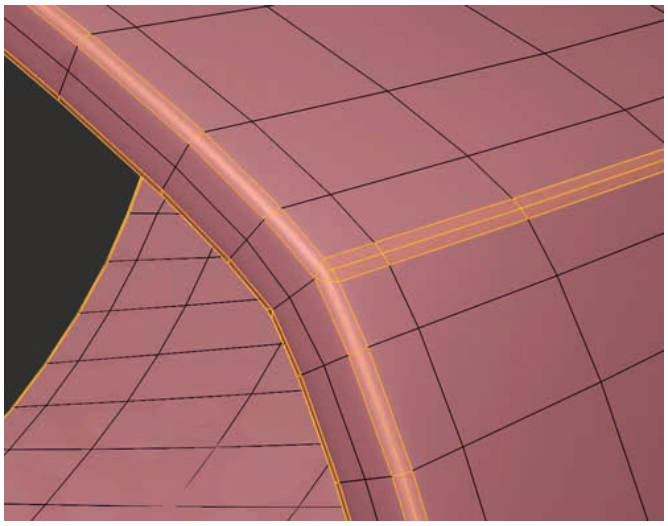


Use this Vertex Group in the **Bevel modifier** to get the most predictable result.



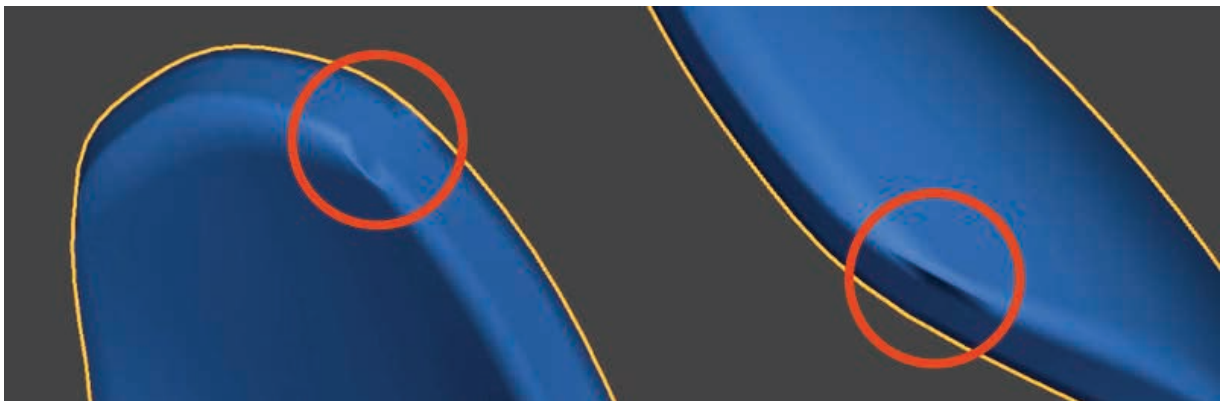
If these non-destructive edge beveling methods are not your cup of tea, you can remove any Bevel modifier or creasing, select the edges you want to bevel and **press Ctrl+B** to bevel them.

Scroll up on the mouse wheel to increase edge loops. With three loops, you get the original one in the middle and a support loop on either side.

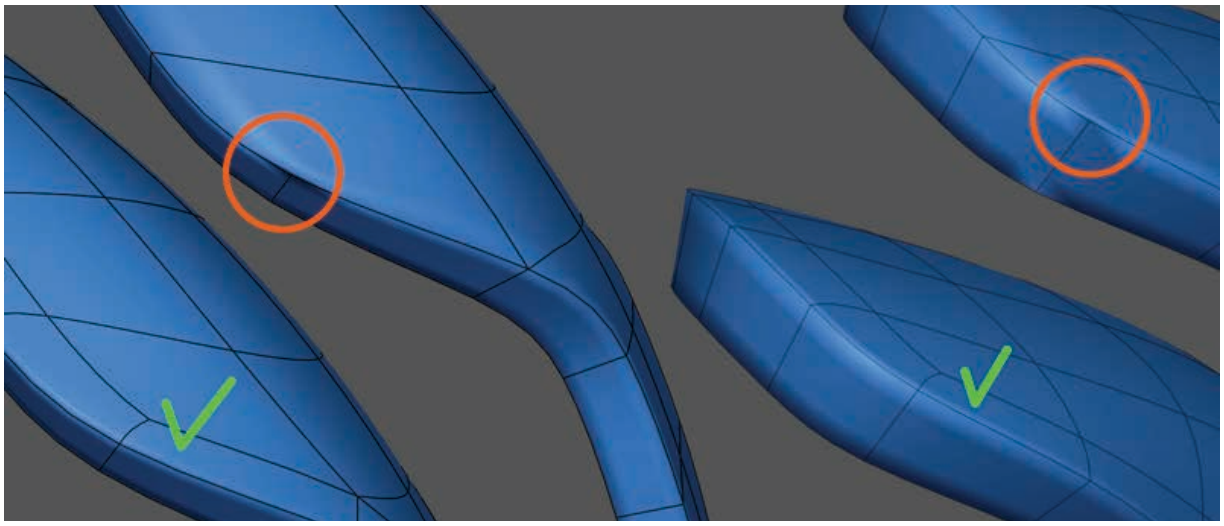


Quick topology fix for ugly edges

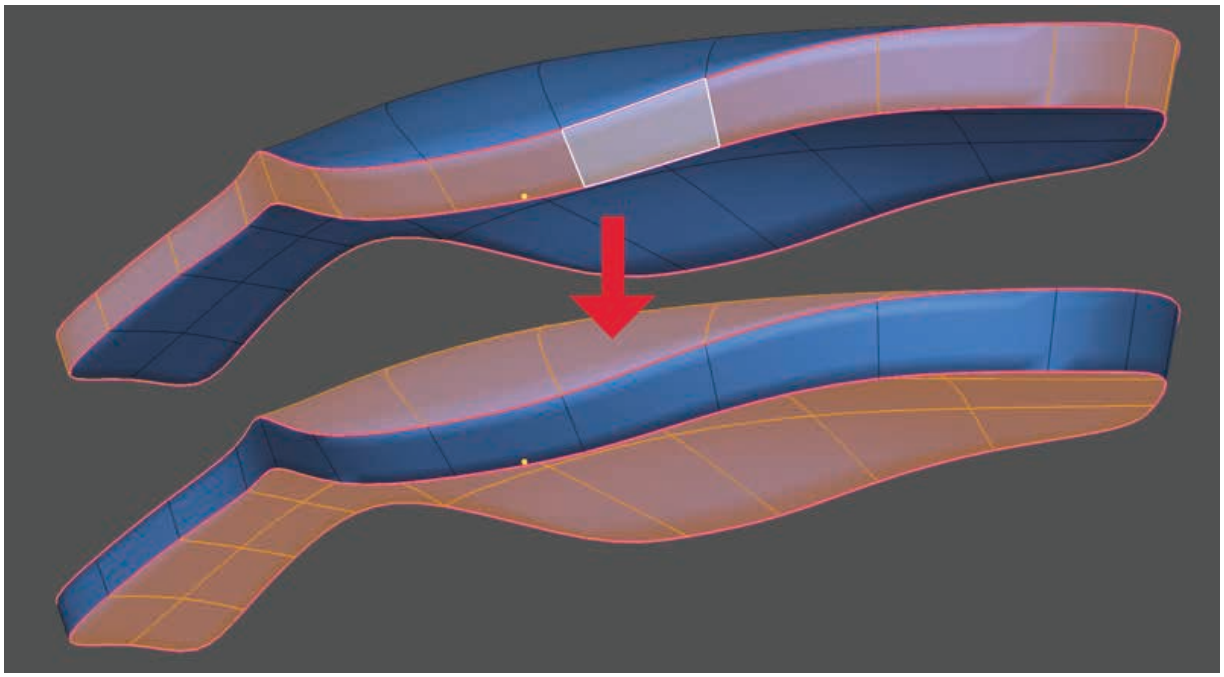
If your panels have some ugly modeling artifacts like this, that's because there's no nice way for the subdivision modifier to create smoothness.



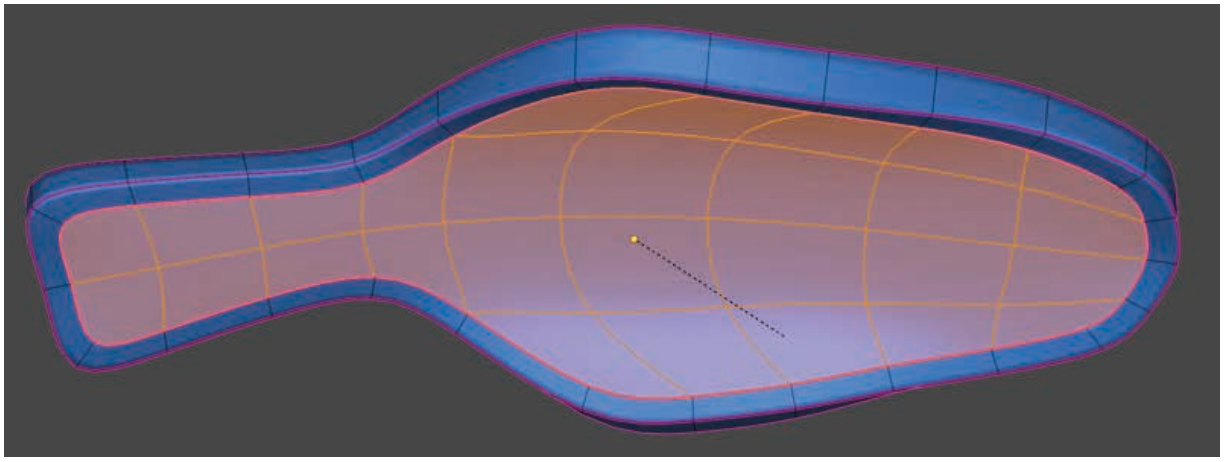
To fix this you need a loop of faces around the edge.



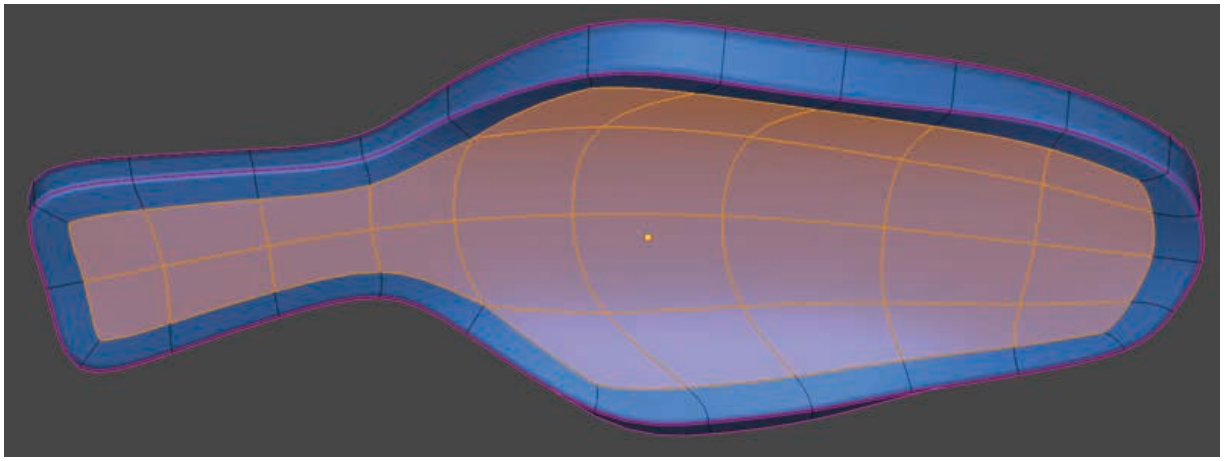
This is very easy to fix, simply **alt-left-click** on one of these faces to select the rim, then **press Ctrl+i** to invert the selection.



Now **press i** and move the mouse to inset, creating a loop of faces around the edge.

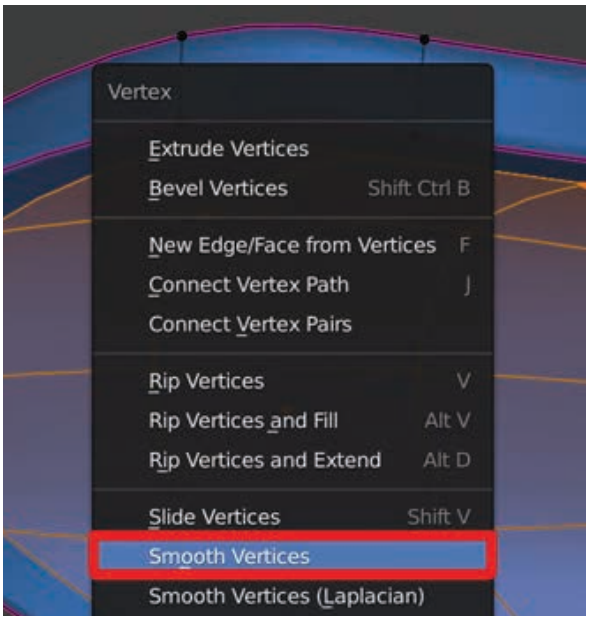


In case you had Creasing on the edges, **press Shift+E+[-1]** to erase it from the inset edges.

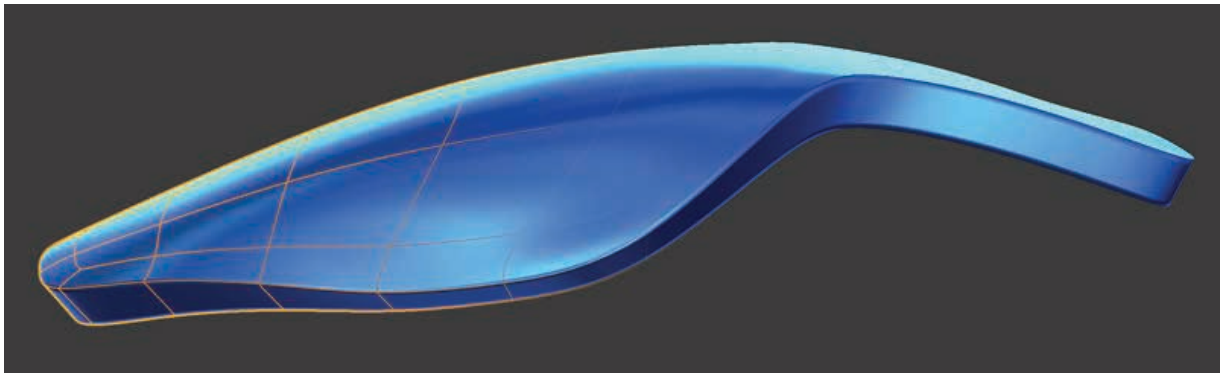


If you want you can smooth the inset vertices. **Press Ctrl+V** for the Vertex menu and choose **Smooth Vertices** (you can repeat this a couple of times by **pressing Shift+R** if necessary).

Be careful with smoothing vertices too much on a curved surface like this, however.

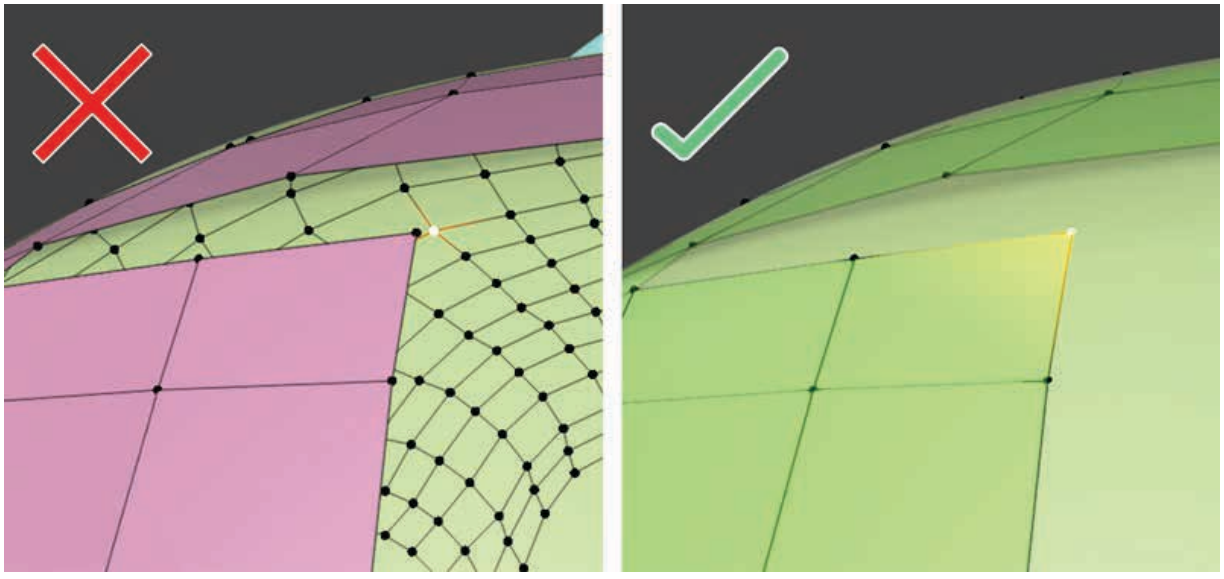


Now the topology artifacts are gone.



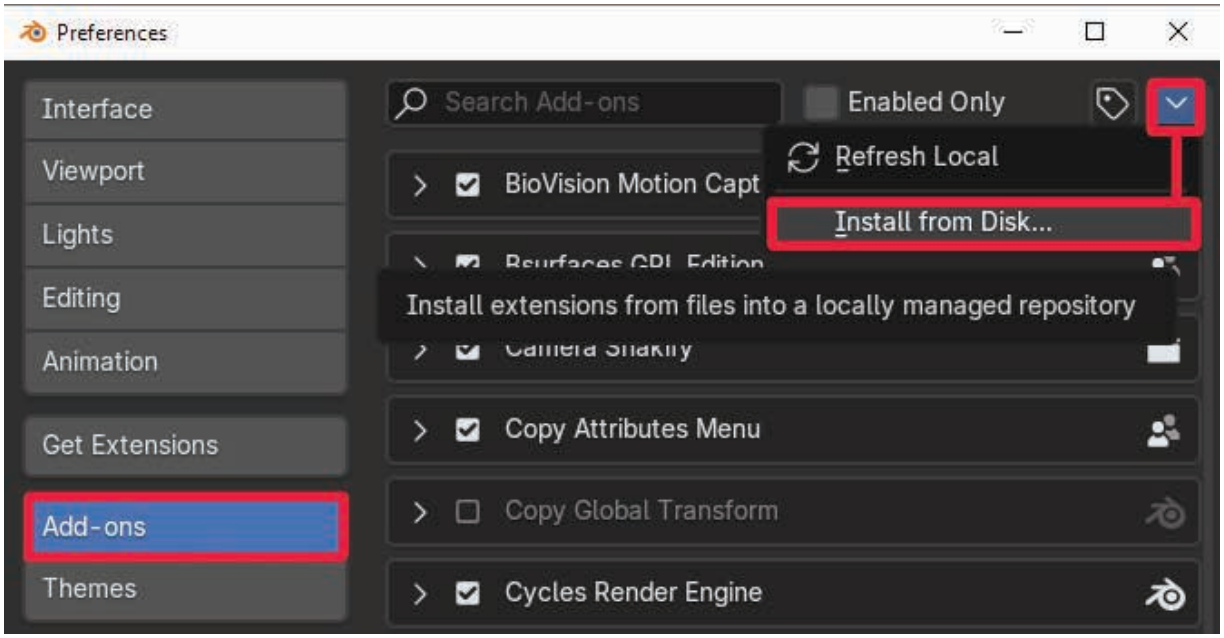
Retopology Visibility Tip: Draw Xray

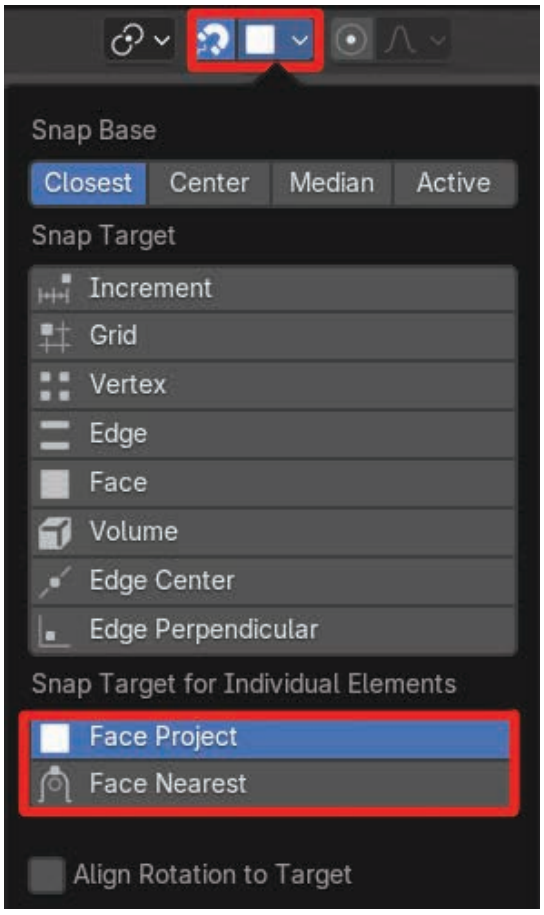
When retopologizing in #Blender, it can be annoying that the vertices and edges on the other side are still visible. Draw X-ray is a free add-on that takes care of this, making retopo life easier. You can download it here.



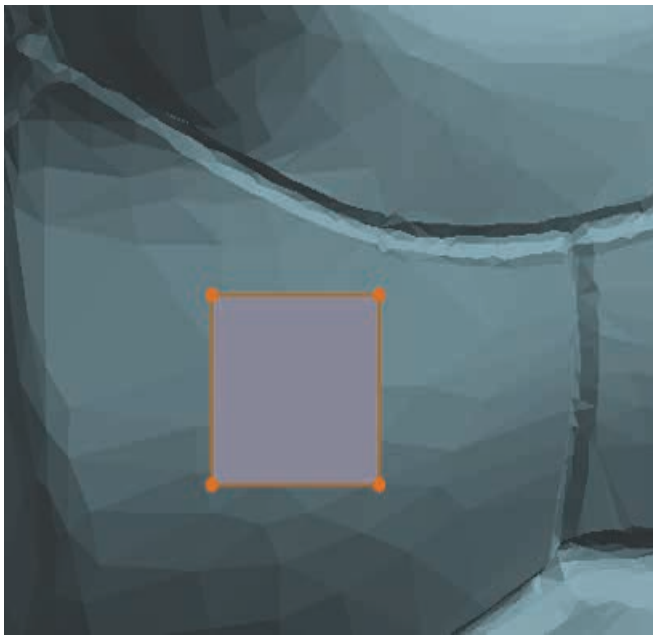
Download the Zip file from Gumroad and **Install from file** in Preferences. You can choose the free version or the paid version.

The paid one offers snapping functionality, however I use the normal snapping tools in Blender (see below) and they are enough for me. But if you use and like this add-on, you can still consider donating some money to the creator.

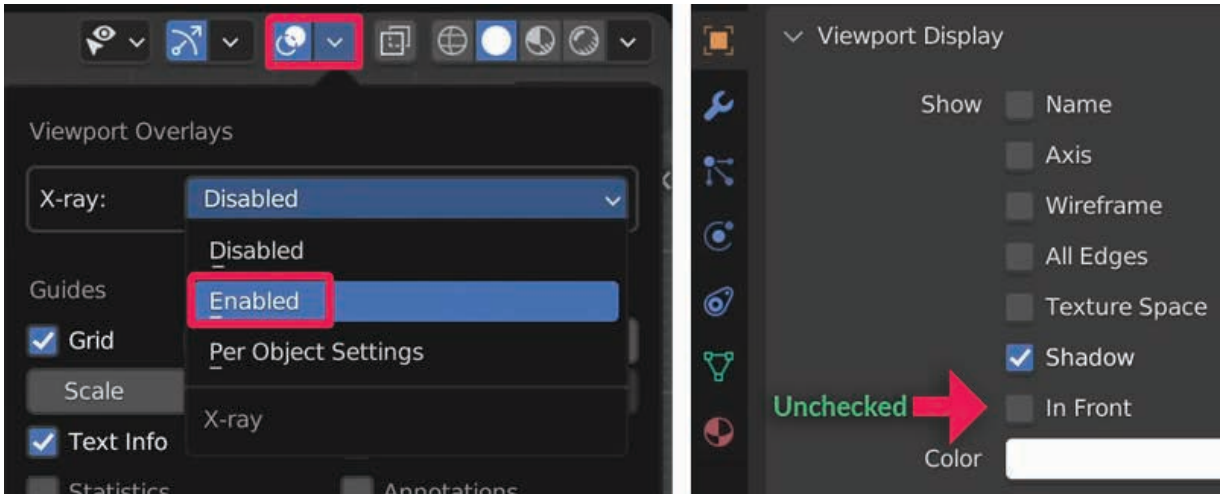




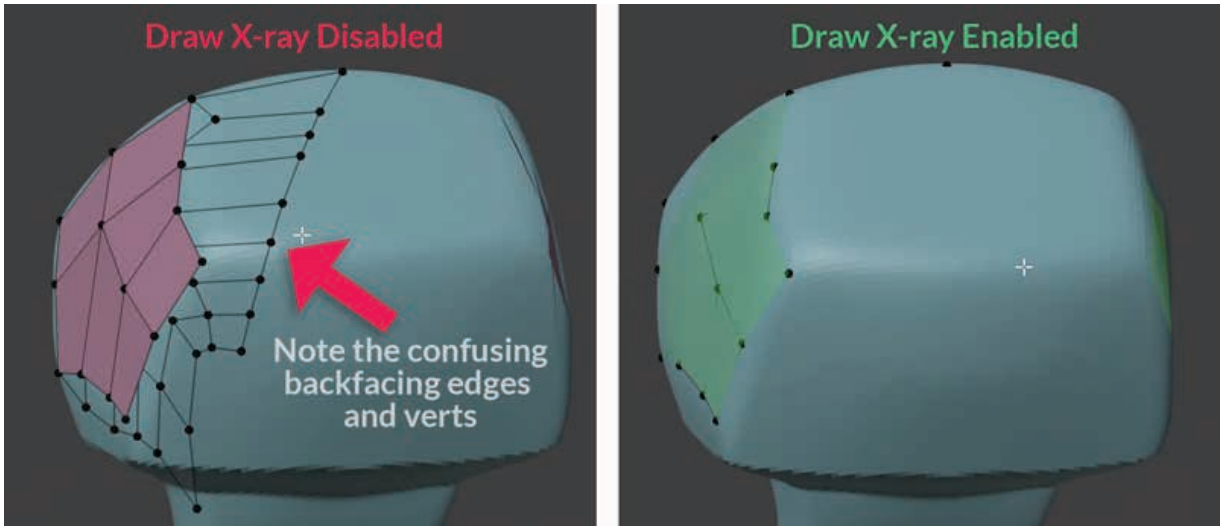
To start your retopo, add a **Plane**, and snap it to the sculpt in Edit Mode using the settings indicated on the left.



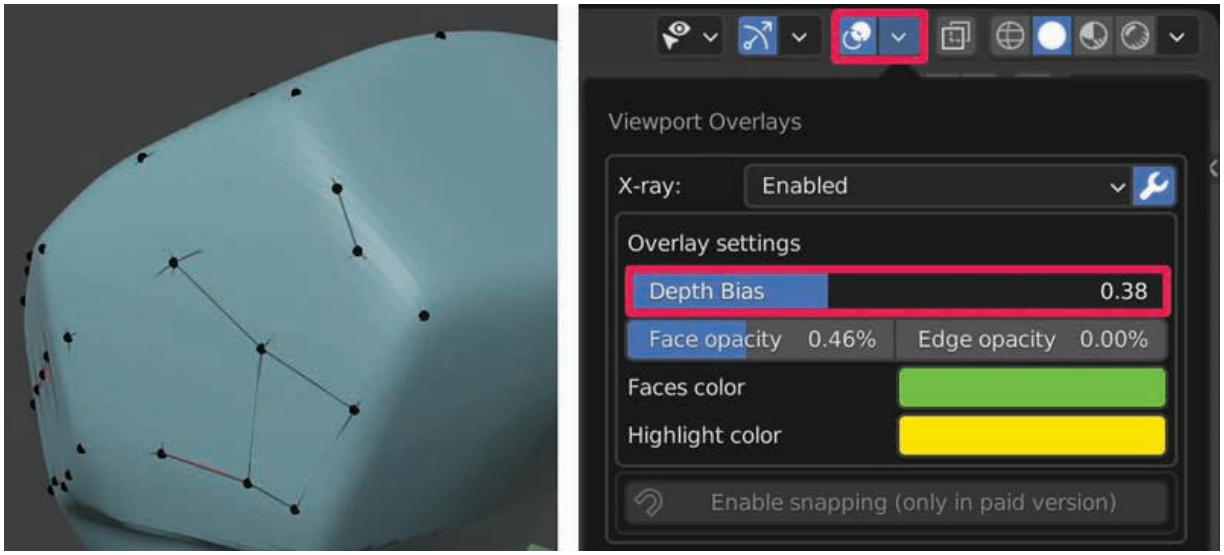
You can find the Draw X-ray options in the **Viewport Overlays** drop-down menu. Make sure it is set to **Enabled**. And make sure under Viewport Display you don't have In Front checked on your retopology mesh, as it will interfere with how this add-on works.



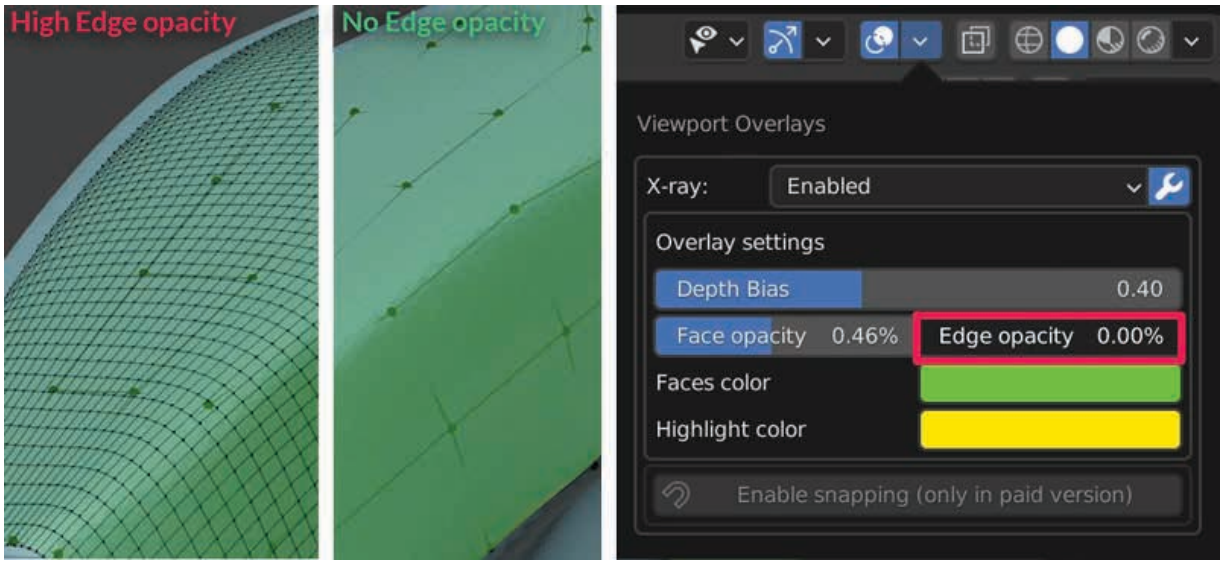
Now you can see the remesh topology on top of your sculpt, without being confused by the edges and vertices on the other side.



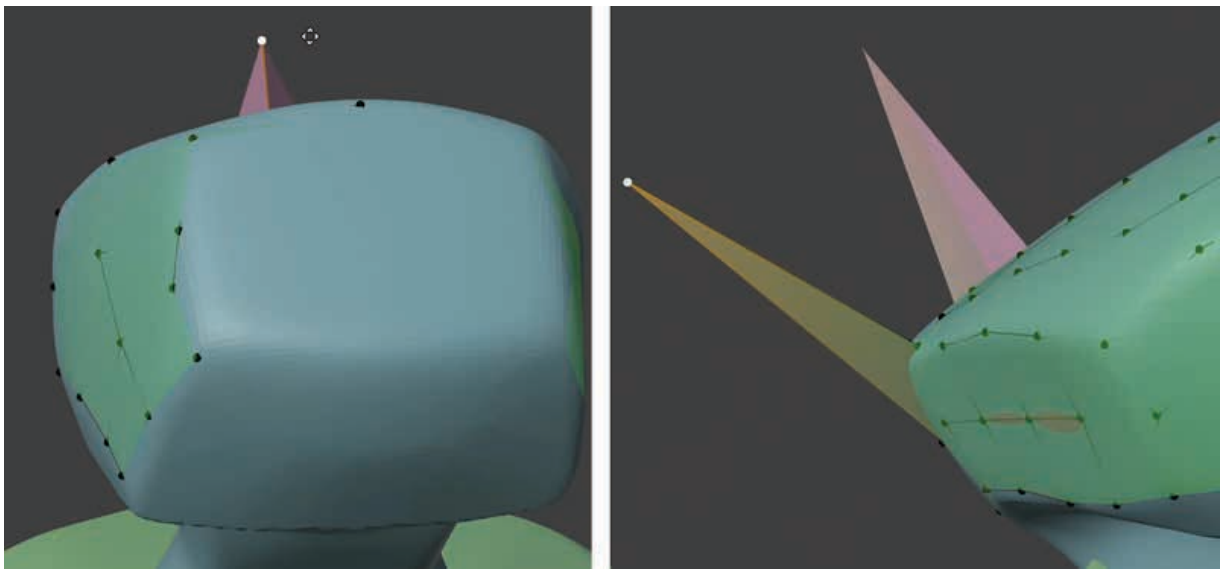
If you don't see the faces, increase the **Depth Bias**.



I prefer to turn **Edge Opacity** all the way down, it looks better once you use a **Subdiv modifier**. Otherwise you will see all the edges the modifier adds, which looks a bit chaotic.



The only thing you have to be careful of, is that you don't accidentally select a vertex on the other side.

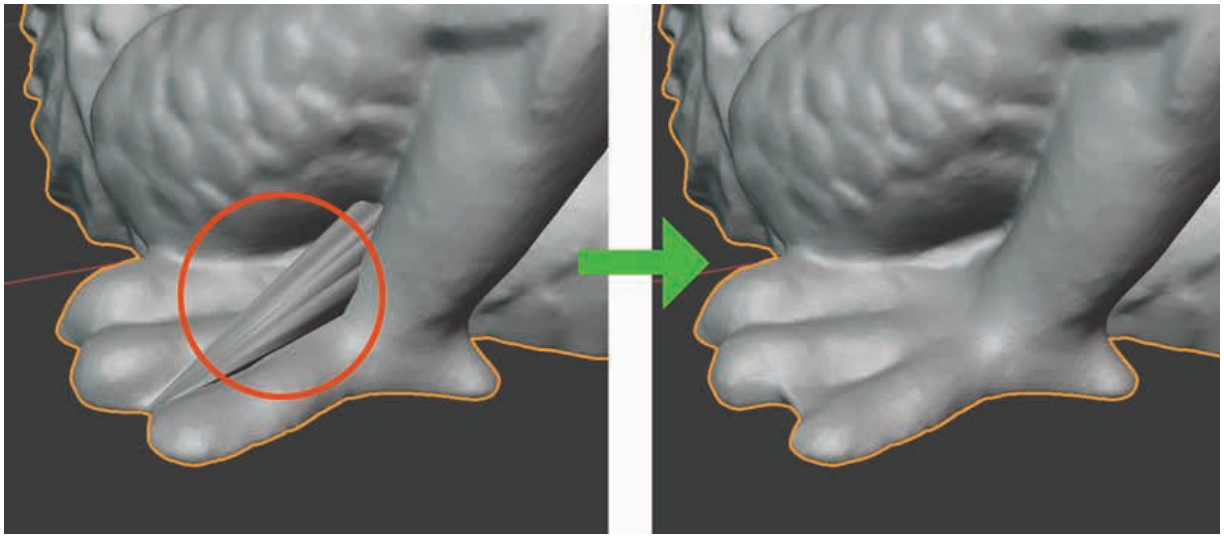


For more retopology tips, check my Retopology playlist on Youtube.

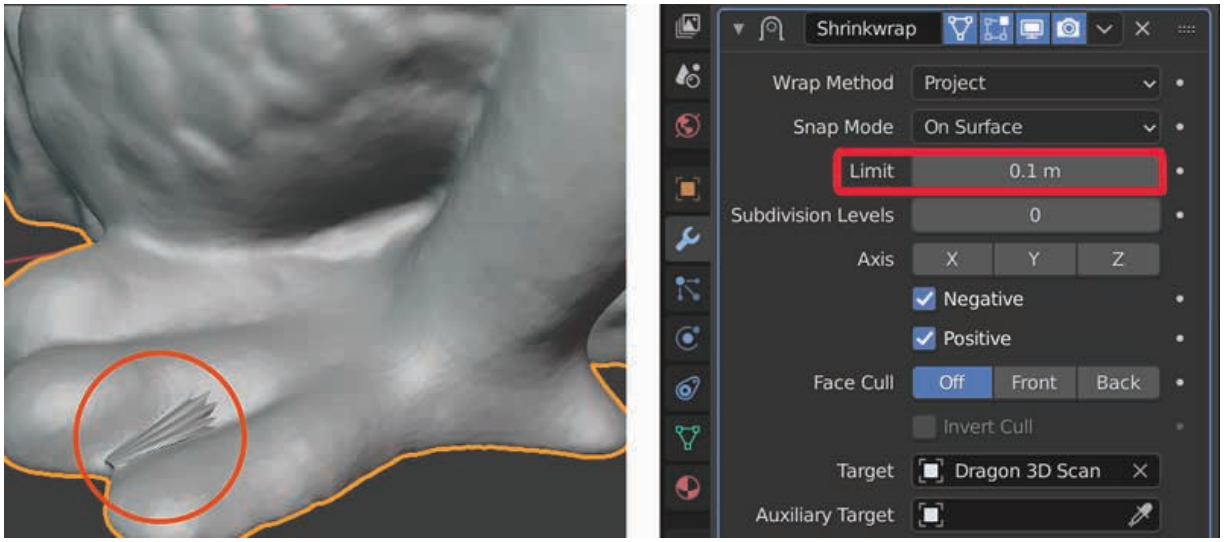
Disclaimer: In Blender 3.1.2 I did notice some crashes when using **Grid Fill** while this add-on was enabled. I can't say for sure it was caused by the add-on, but I've never had crashes using Grid Fill before. So make sure you save a lot, or try it out first before using it for serious work.

Fixing Shrinkwrap Problems

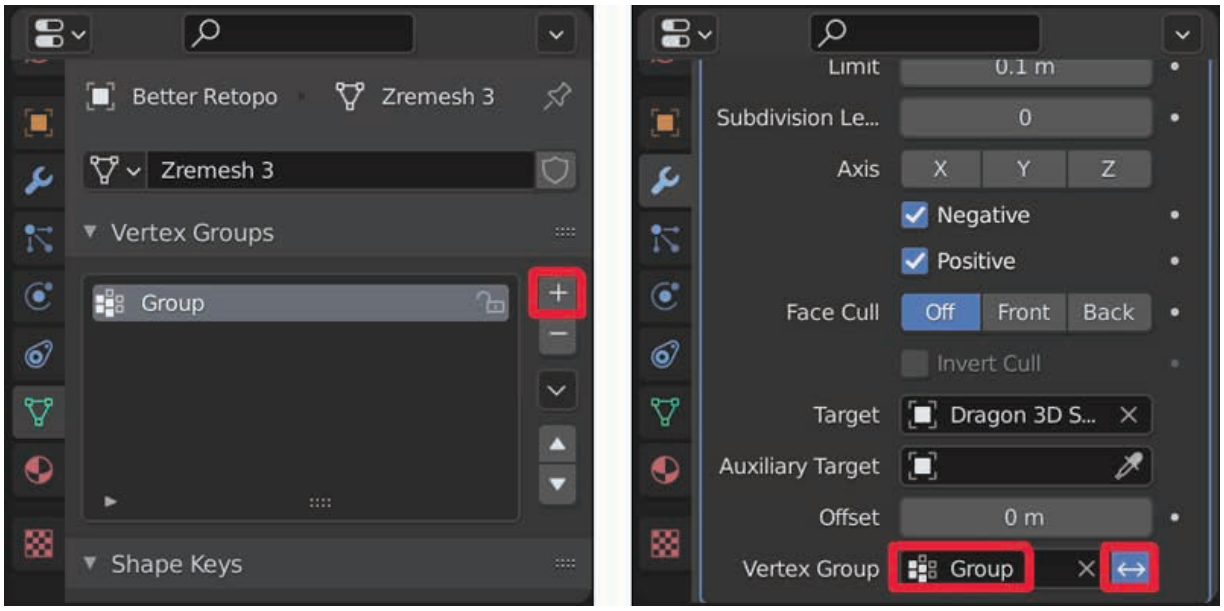
If you're trying to use the **Shrinkwrap modifier** to reproject a high detail model onto a low detail base mesh (as detailed in this video), you might run into this kind of problem as shown below on the left. Let's look at some possible fixes.



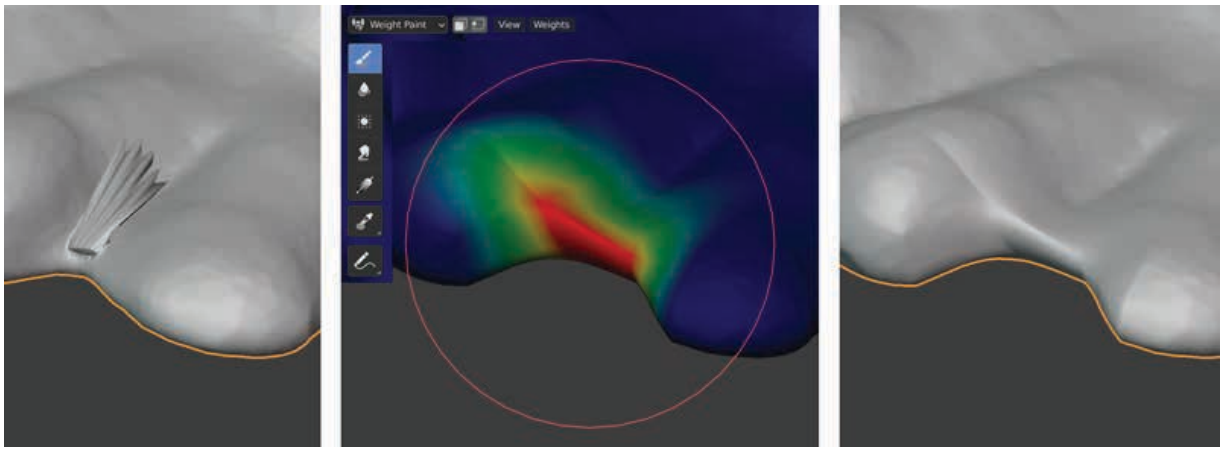
First you can try increasing the **Limit** value. That already takes care of most of the problem, but a small issue still remains.



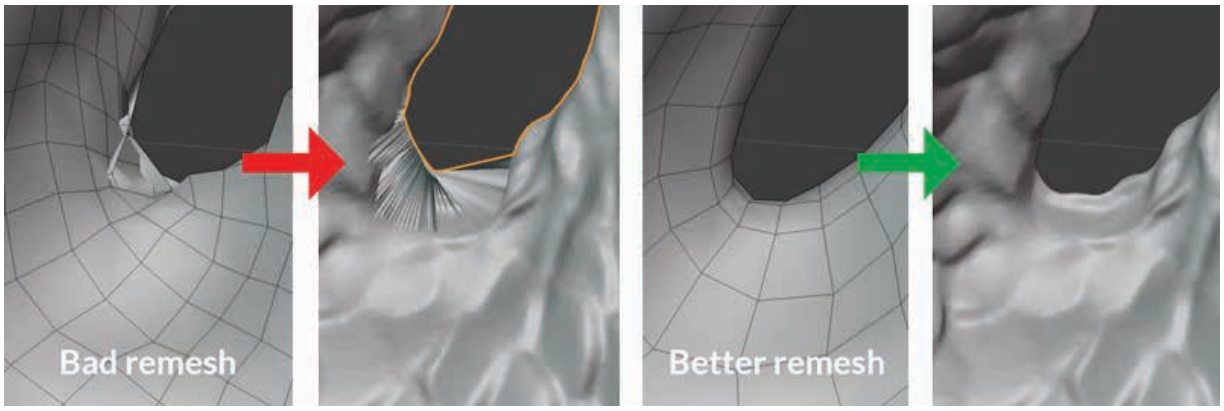
This can be solved by adding those vertices to a Vertex Group and excluding them from the Shrinkwrap modifier. To do this, create a new **Vertex Group**, add it to the Shrinkwrap modifier and **click on the double arrow icon** (this inverts the Vertex Group).



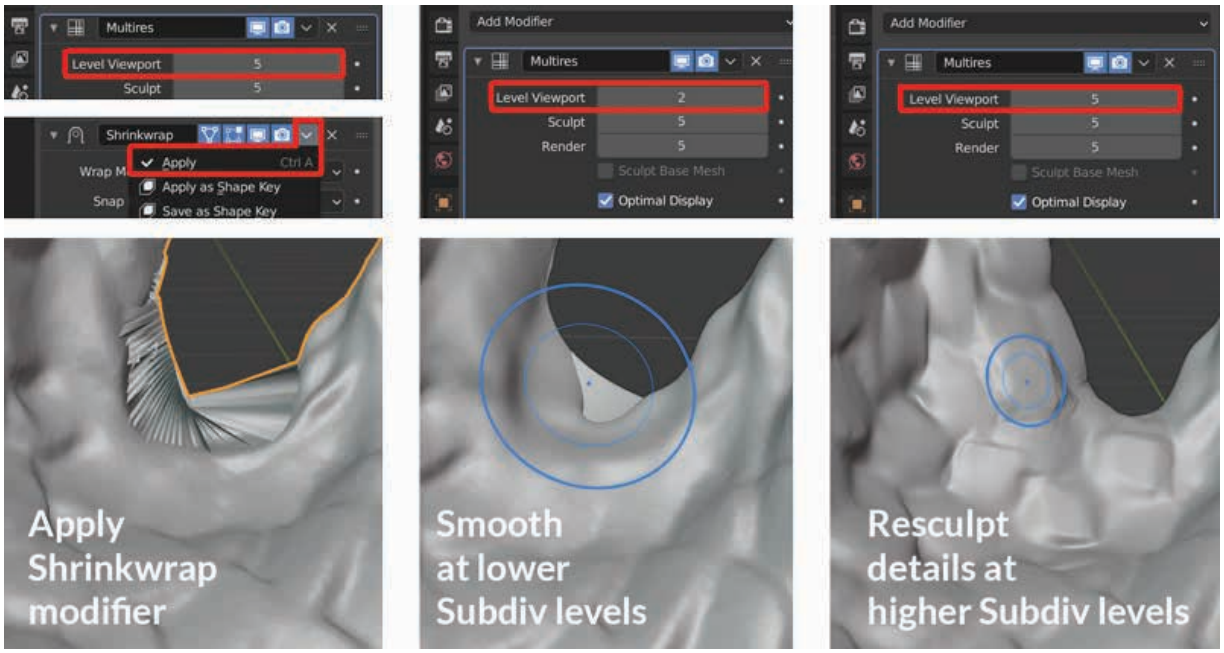
Then go to **Weight Paint** mode and paint the problem away! You may have to lower the Viewport Level in the Multires modifier, otherwise it will go quite slowly if you have a lot of subdivisions.



In some cases the issue is the topology of the base mesh. If you use a remeshing algorithm like the default Quad remesher in Blender, the result is not always ideal. In such a case the safest solution is using a better remesh algorithm (using either the free Instant Meshes app or the paid add-on Dynremesh, for example) or to manually retopo the base mesh.



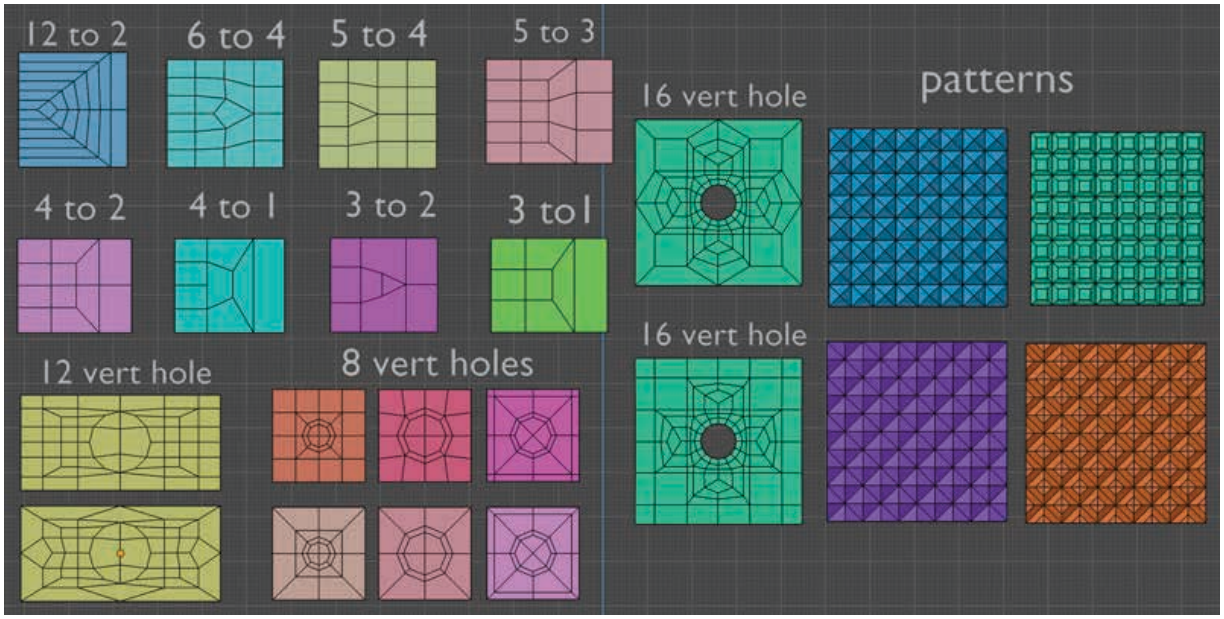
Another method to fix these issues is to first **Apply** the Shrinkwrap modifier (make sure the Multires modifier is in its highest subdivision level) and then in **Sculpting** mode, smooth out the trouble area in a lower viewport level, and manually resculpt the details in the higher viewport level.



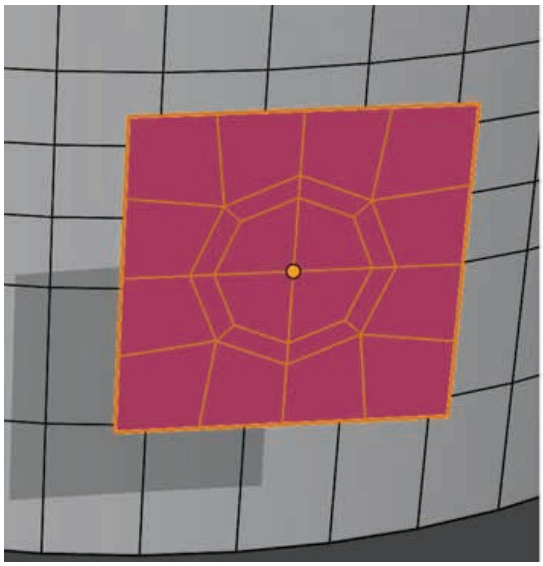
Copy / Paste Geometry for Faster Modeling

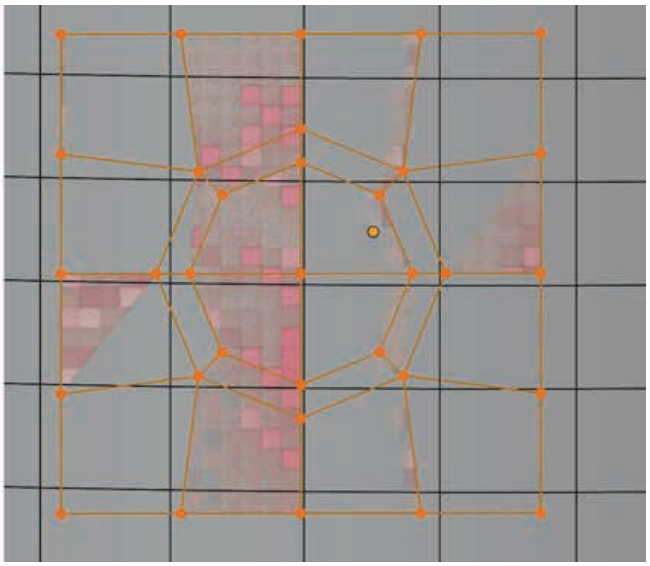
There are some things that you'll find yourself modeling again and again. It can be useful to keep these topologies as assets for re-use.

In case you want to download the **Blend-file** of this topic, you can find it here.

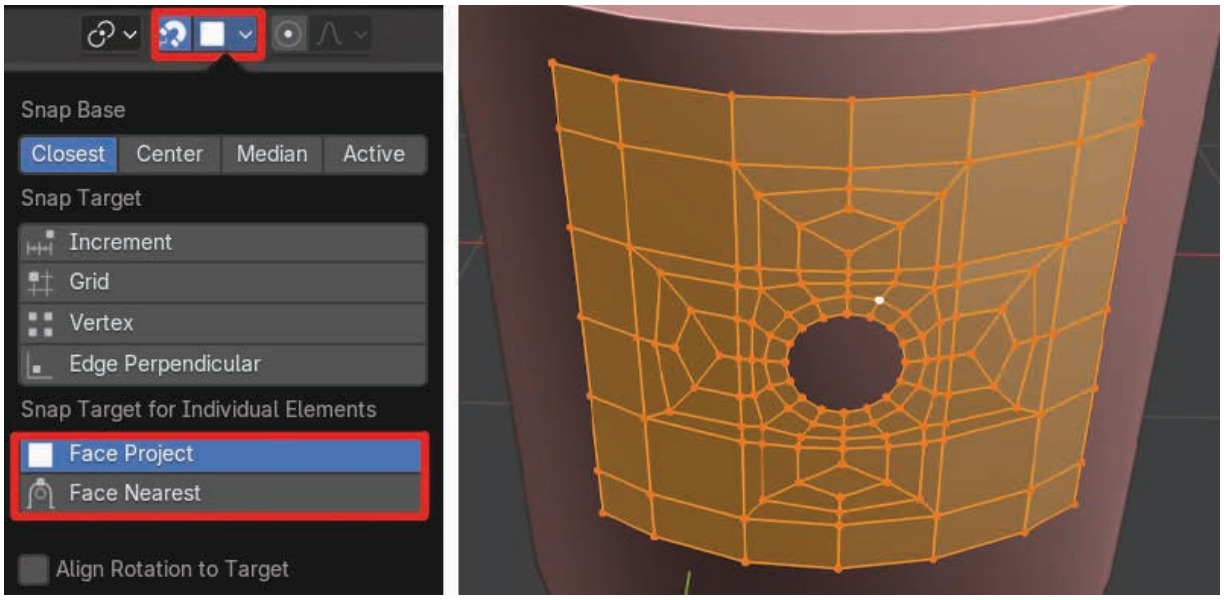


Then just copy / paste them when needed and snap them to your geometry. You can turn on **Snapping** to Surface and move it to where it's needed. Then once you select all in Edit Mode and move it, it will snap to the surface of the other object.

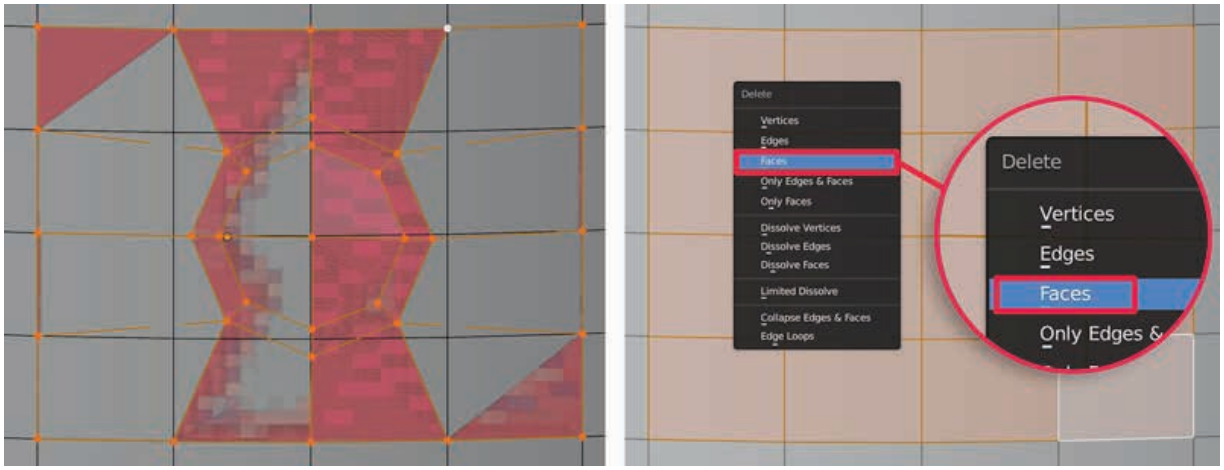




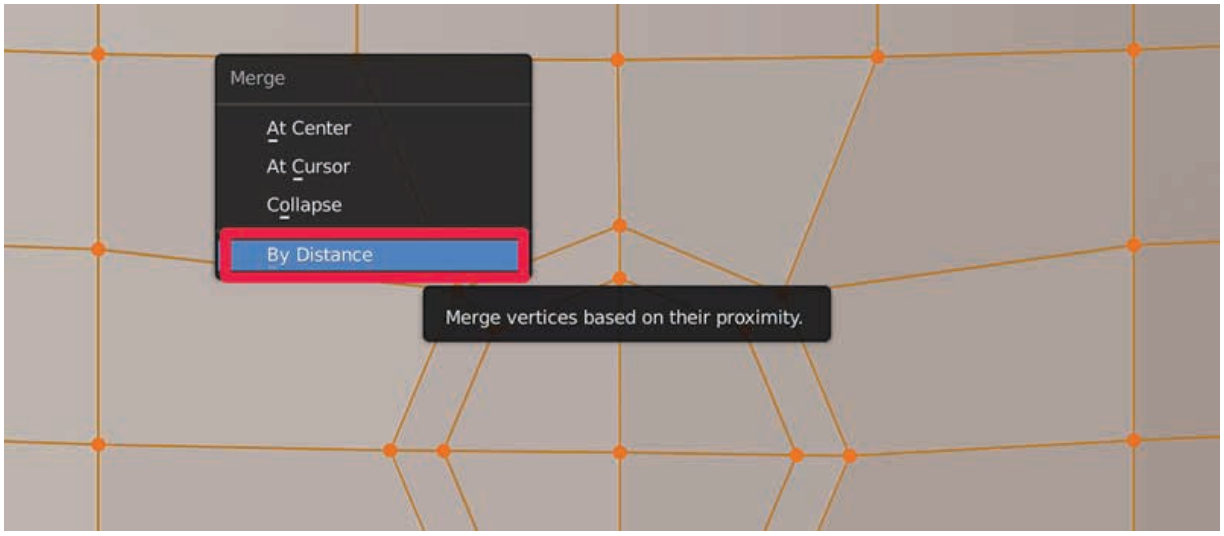
Below are the settings you can use to snap it to the surface. Once snapping is enabled, **pressing G and Enter** or G and Left Mouse Button, will snap all the Vertices to the surface behind it.



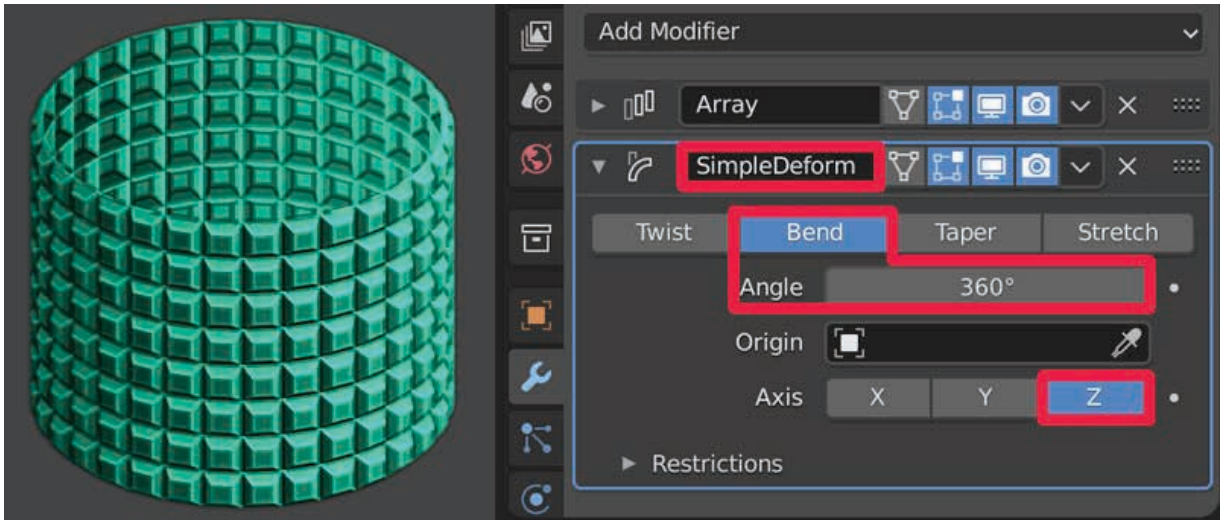
Snap all the boundary vertices so that there's no gaps. Turn off snapping when you're done. Then temporarily hide the pasted geometry by selecting it in Object Mode and **pressing H**. Remove the Faces that the new geometry is replacing in Edit Mode.



Unhide the hidden geometry by **pressing Alt-H** and merge vertices with **Merge by Distance**.



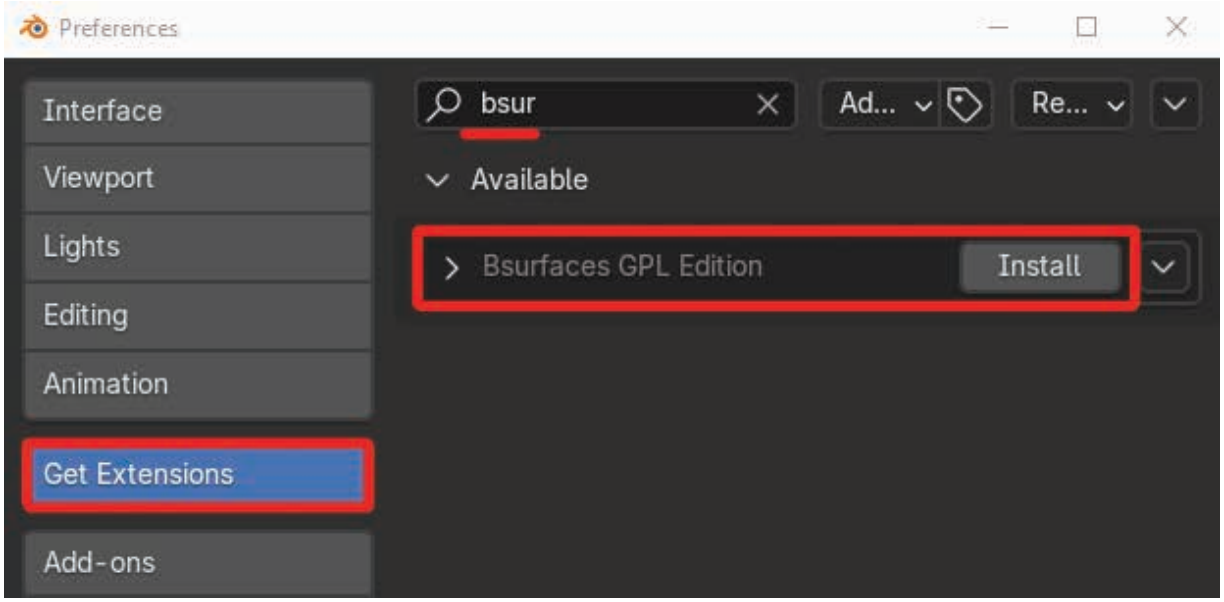
If you want to create circular details you can also just use a **Simple Deform modifier** on the copy/pasted geometry. Set it to **Bend** with an **Angle of 360°** and use the correct axis (Z in the below example). You can optionally add an **Array modifier** as well to increase the size of the cylinder.



You can either **copy / paste** from a different Blender window, **append** it from a different file or from a different scene, or use Blender 3.0's **Asset Manager**.

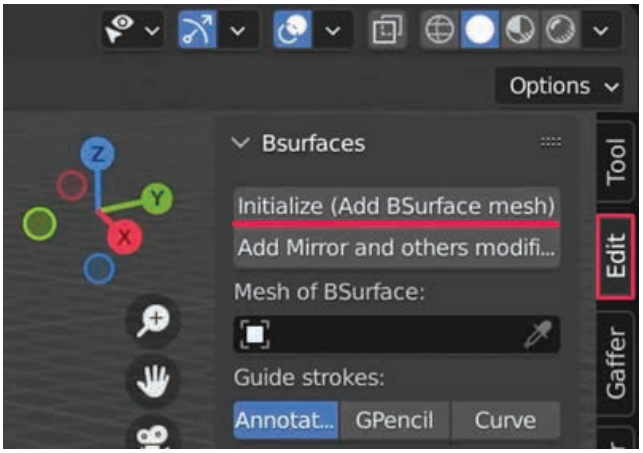
BSurfaces - Drawing Retopology with Annotation

Bsurfaces in an add-on which comes with Blender by default. You just need to install it in Preferences under **Get Extensions**.



After enabling the add-on, you can find it in the **Edit** tab of the Option panel.

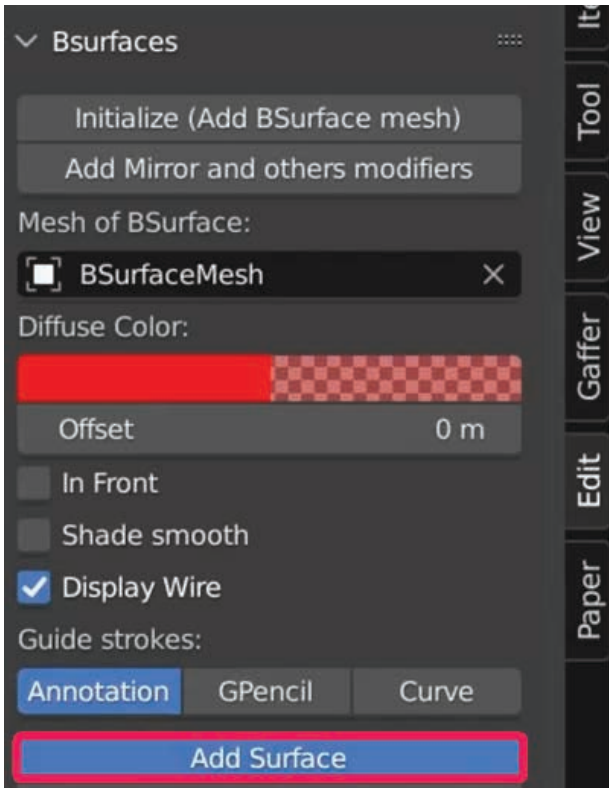
Click on **Initialize** (Add BSurface mesh).



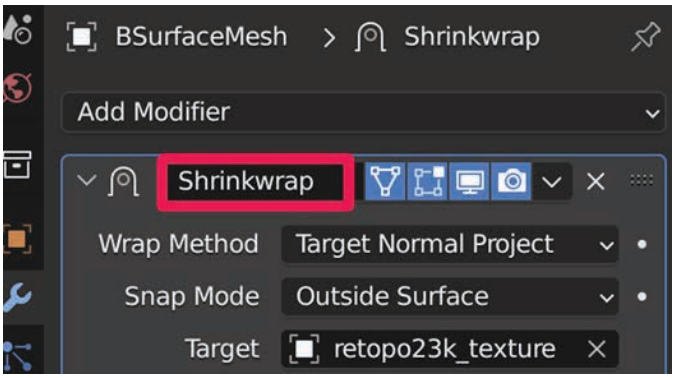
You can now draw some Annotation strokes on the object you want to retopologize.

Click here to return to the index.

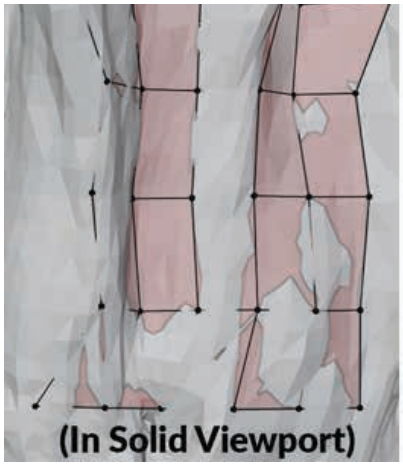
The annotation stroke settings are automatically set to snap to the surface on the object. When you finish drawing the strokes, **click on Add Surface**. Now, BSurfaces has created a mesh based on the Annotation strokes.

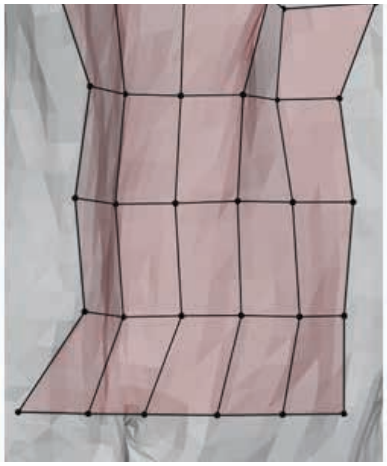


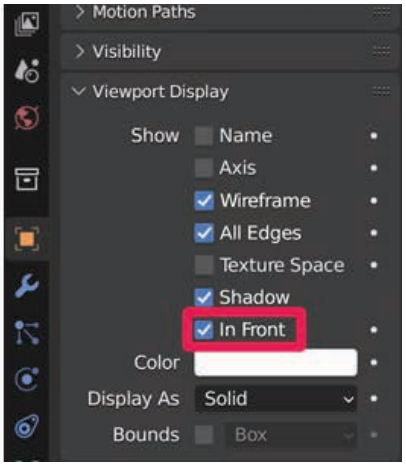
BSurfaces also adds a **Shrinkwrap modifier** to the new geometry it has created.



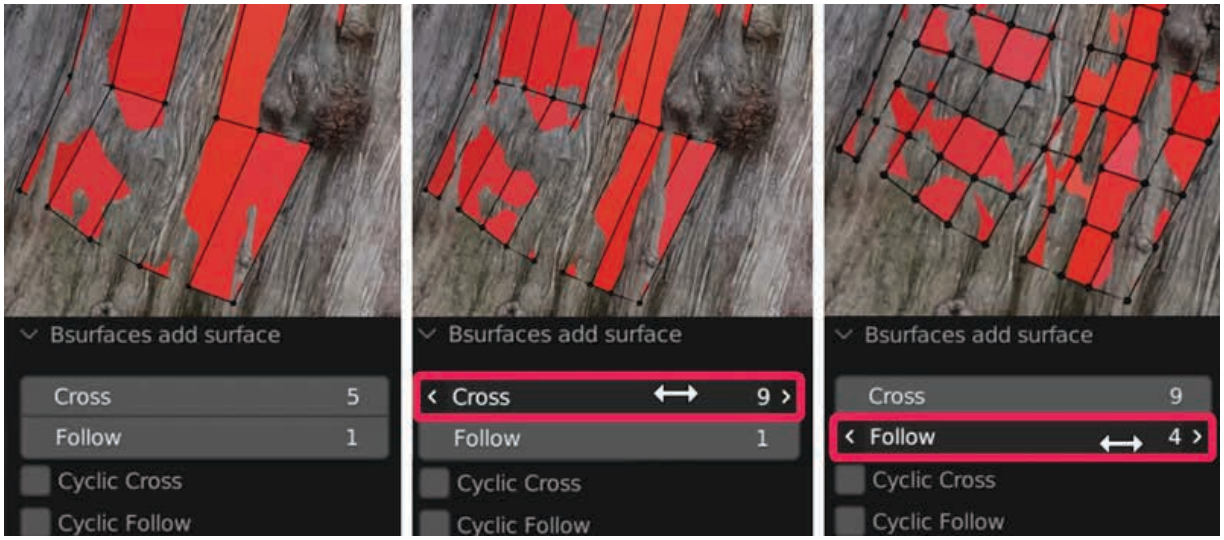
Enable **In Front** to see it better. Note that this only makes a difference in **Solid view.**





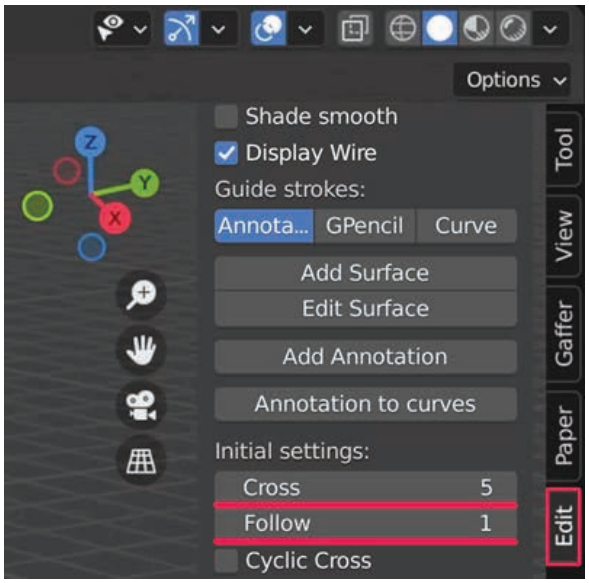


You can still change the amount of loops by changing the **Cross** and **Follow** values.

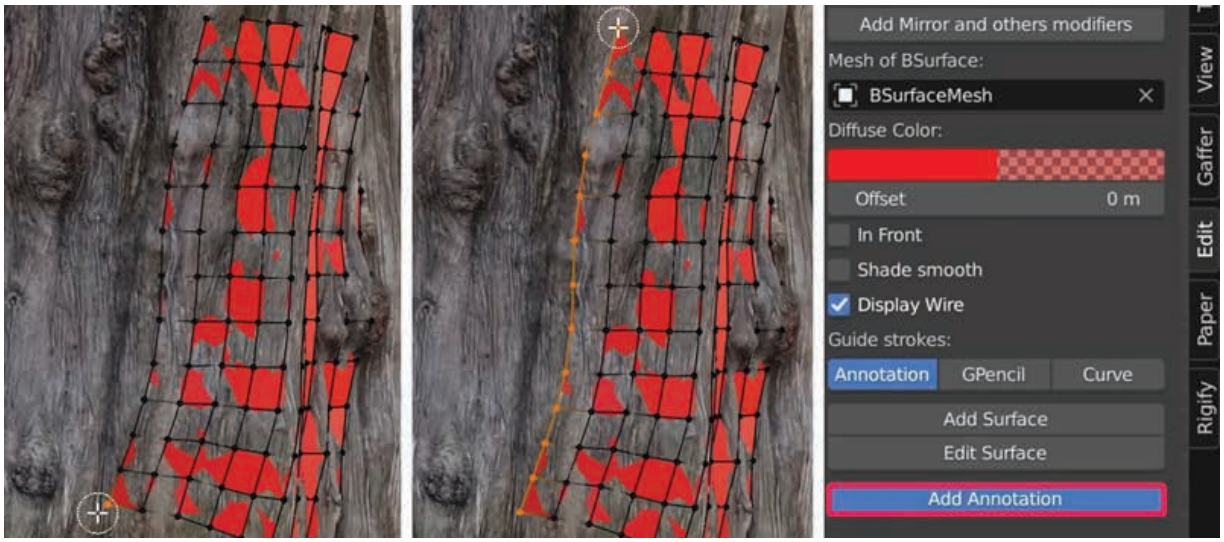


Note that this is not the same as the Cross and Follow values in the **Edit tab**.

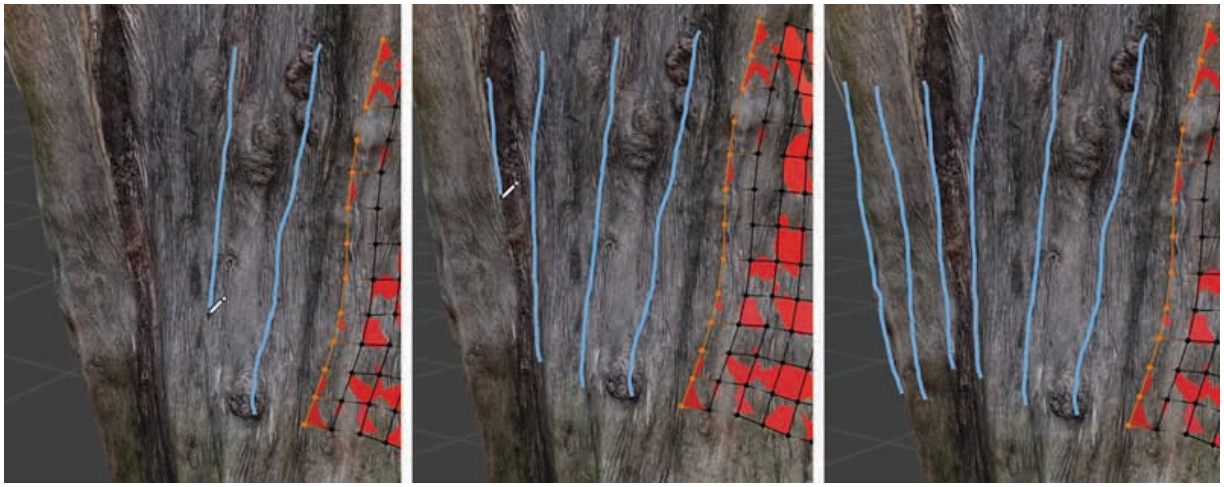
These are just the initial default values that it will use to generate the geometry.



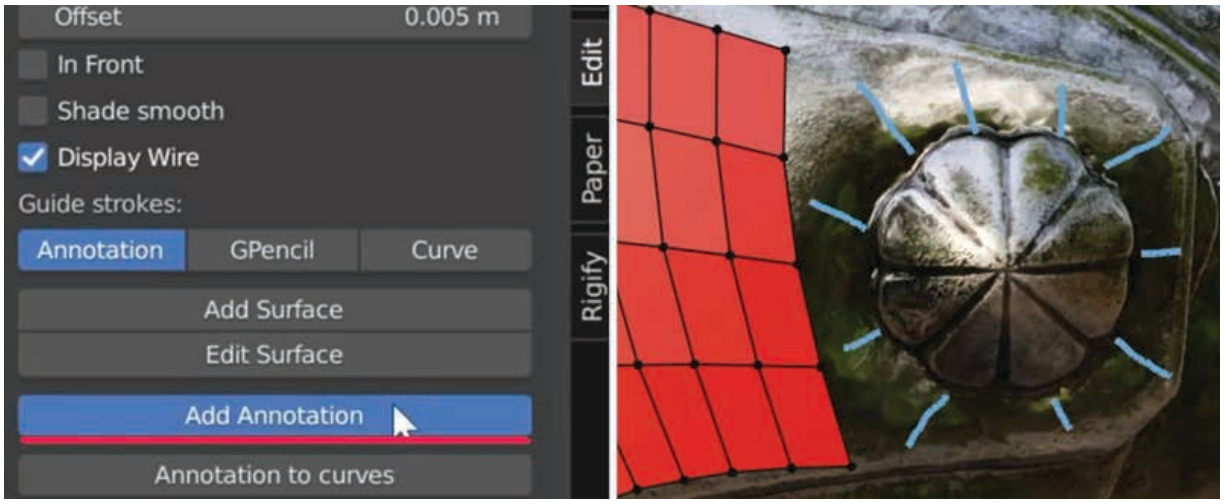
To draw more geometry, click on **Add Annotation**. Then **Press B** and box-select the vertices you want to attach the new geometry to. Or **press C** for circle select, and press Escape once you've made your selection to exit circle select mode.



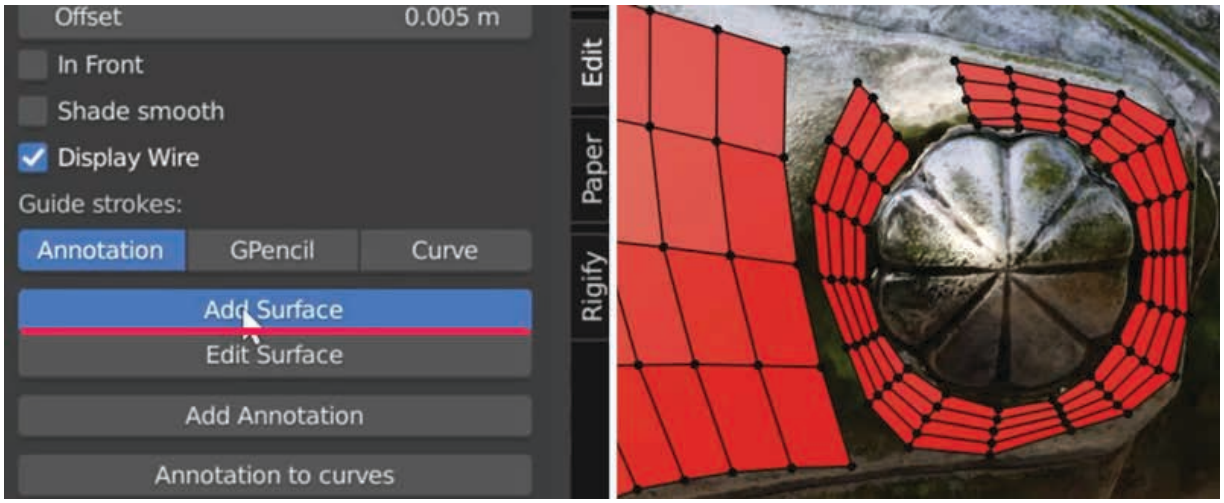
Then draw more Annotate strokes and click on **Add Surface** again.



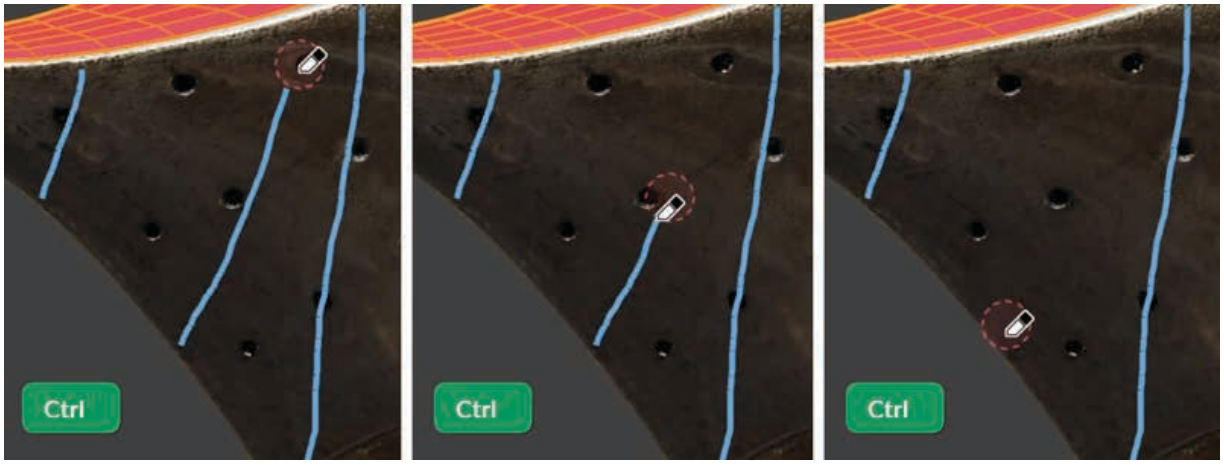
If you don't want new geometry to be connected to anything, don't select any vertices of the geometry before adding more annotation.



Then draw your new annotation strokes and clik on **Add Surface.**



To erase annotation strokes, **hold down Ctrl**.



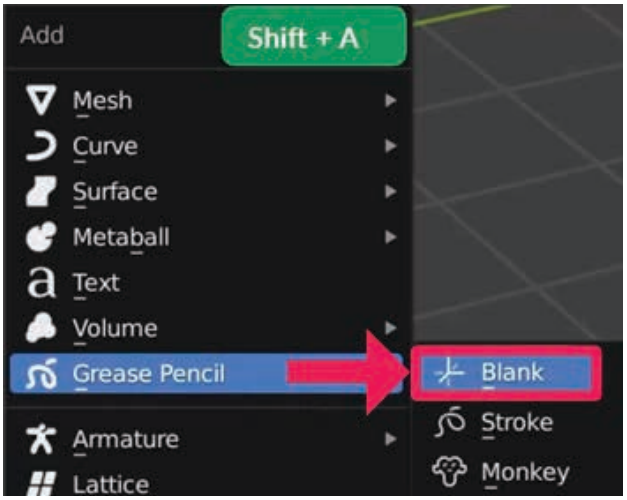
Use BSurfaces with Grease Pencil to sketch geometry

In my opinion Annotation, which we looked at in the previous topic, is the best mode for retopology on top of an existing surface. For example for doing retopo on a 3D scanned or sculpted model. It requires the least amount of clicks and works quite well.

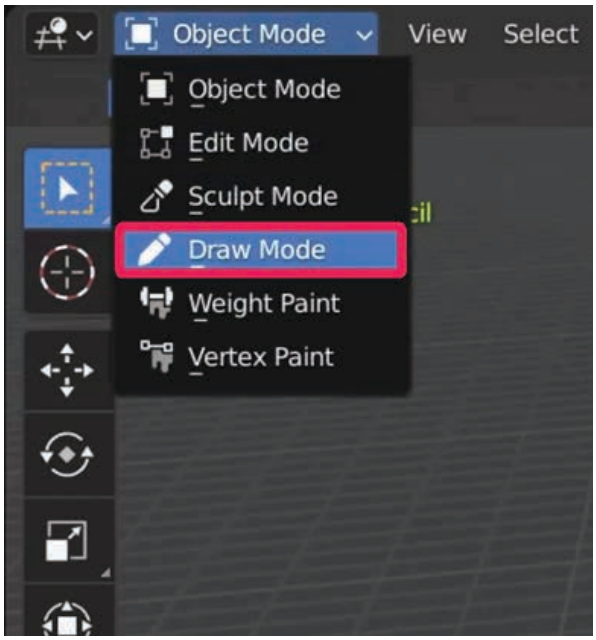
However you can also use Grease Pencil strokes with as Guide strokes for BSurfaces.

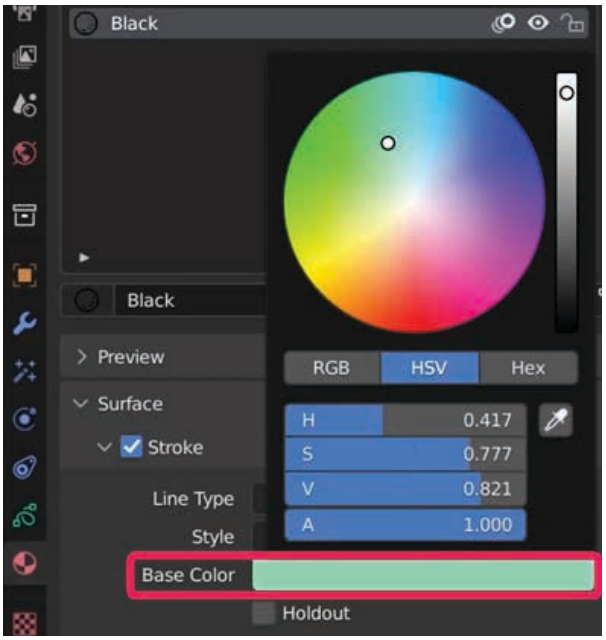
Or you can use it to just create new meshes quickly, instead of using it for retopology.

To do this, add a Blank Grease Pencil object.

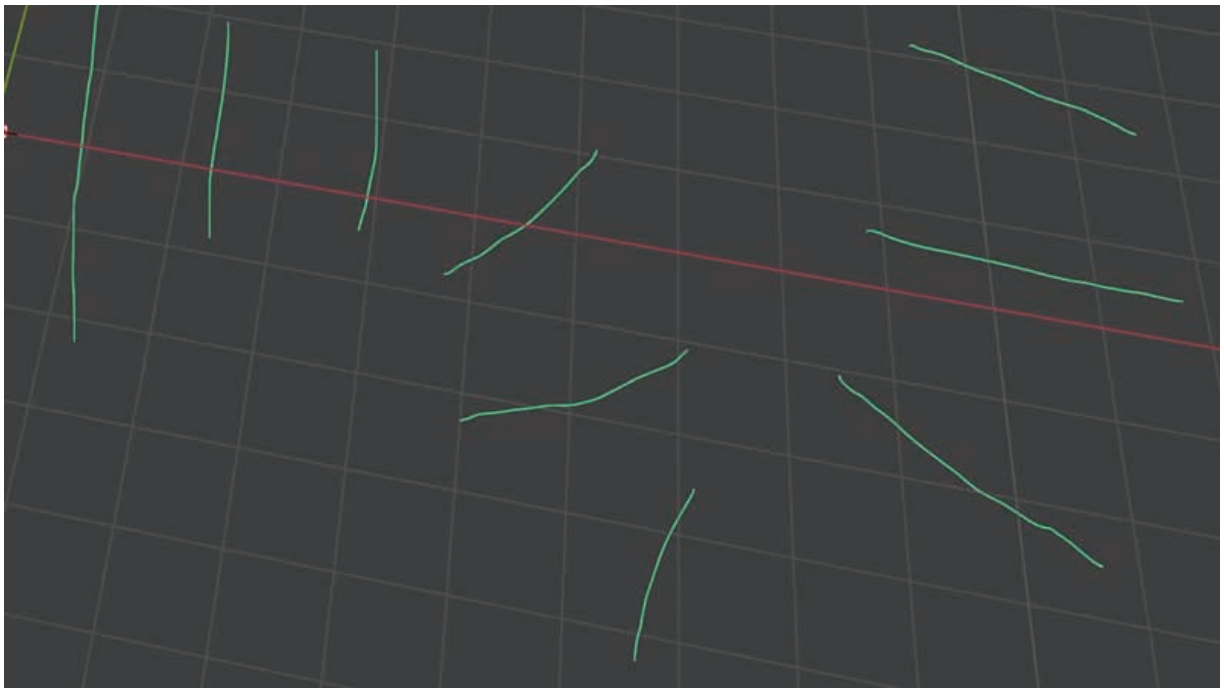


Then switch to **Draw Mode**. You can change the **Base Color** of the Grease Pencil material if you want.

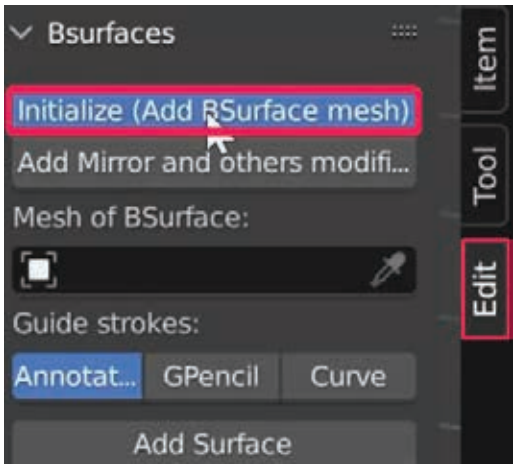




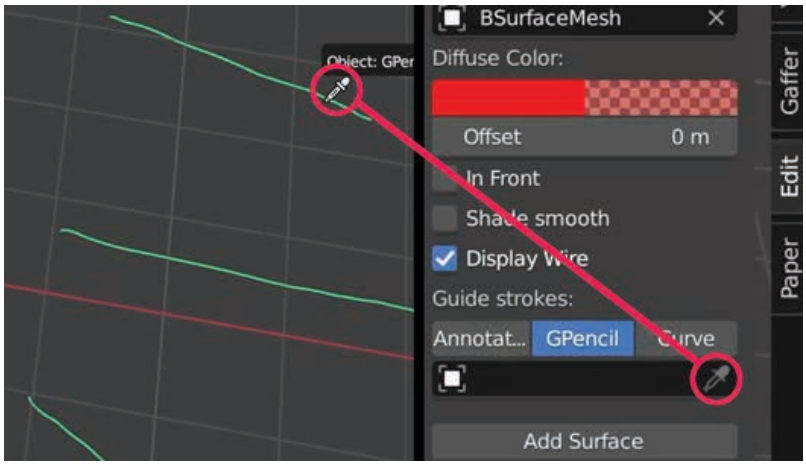
Then draw some **Grease Pencil strokes**.

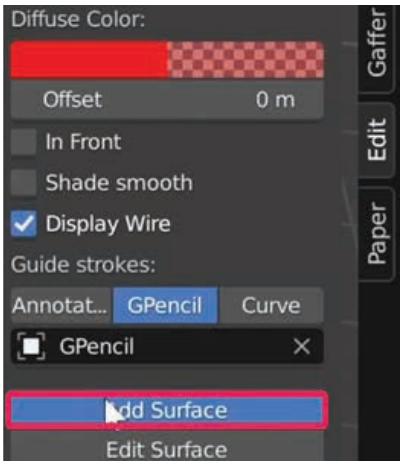


In the BSurfaces panel, **click on Initialize**.

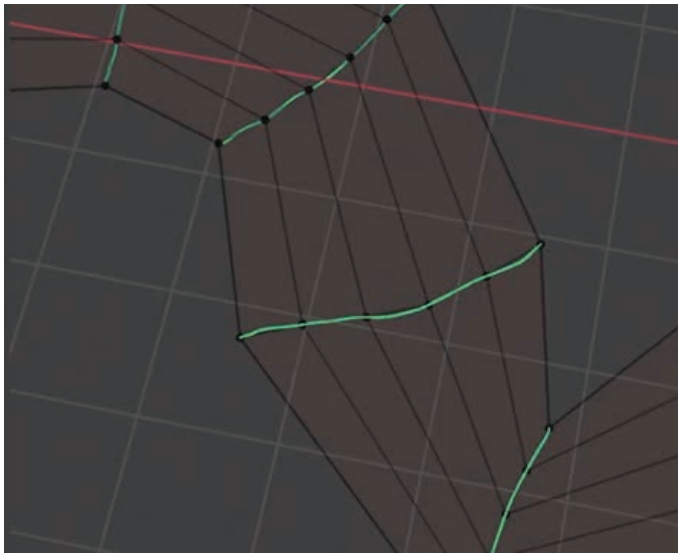


Set **Guide Strokes** to **GPencil** and pick the Grease Pencil object. Click on **Add Surface** to create geometry.

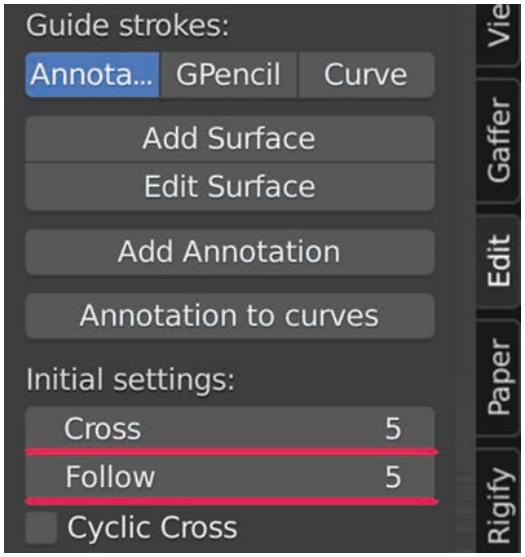




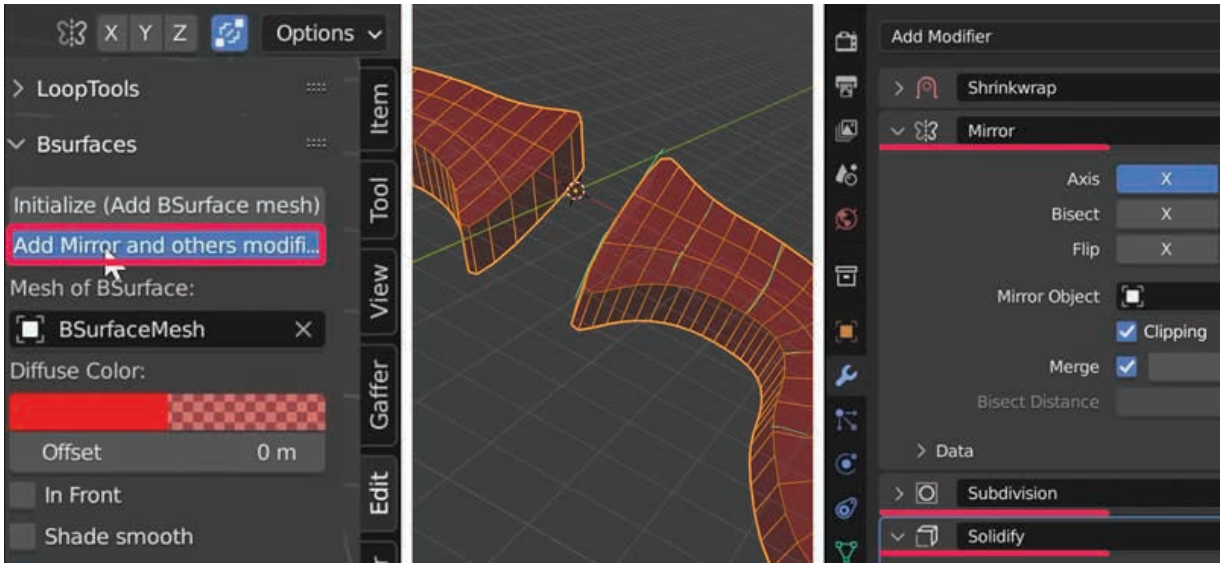
Now you've got geometry based on your strokes.



Adjust the **Cross** and **Follow** values as needed.

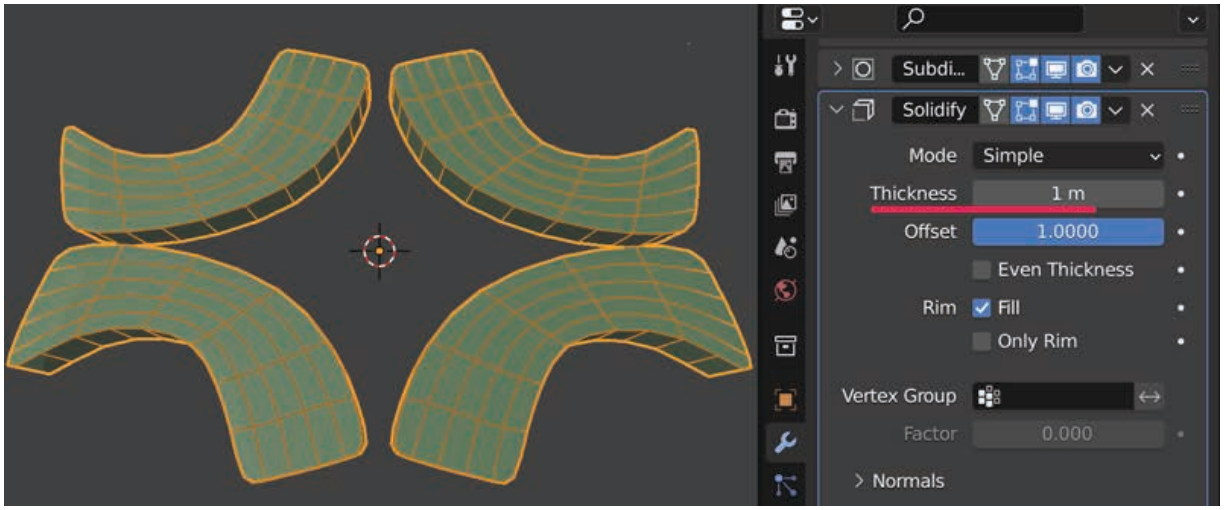


If you click on **Add Mirror and other modifiers**, it will add a Mirror, Solidify and Subdivision modifier.

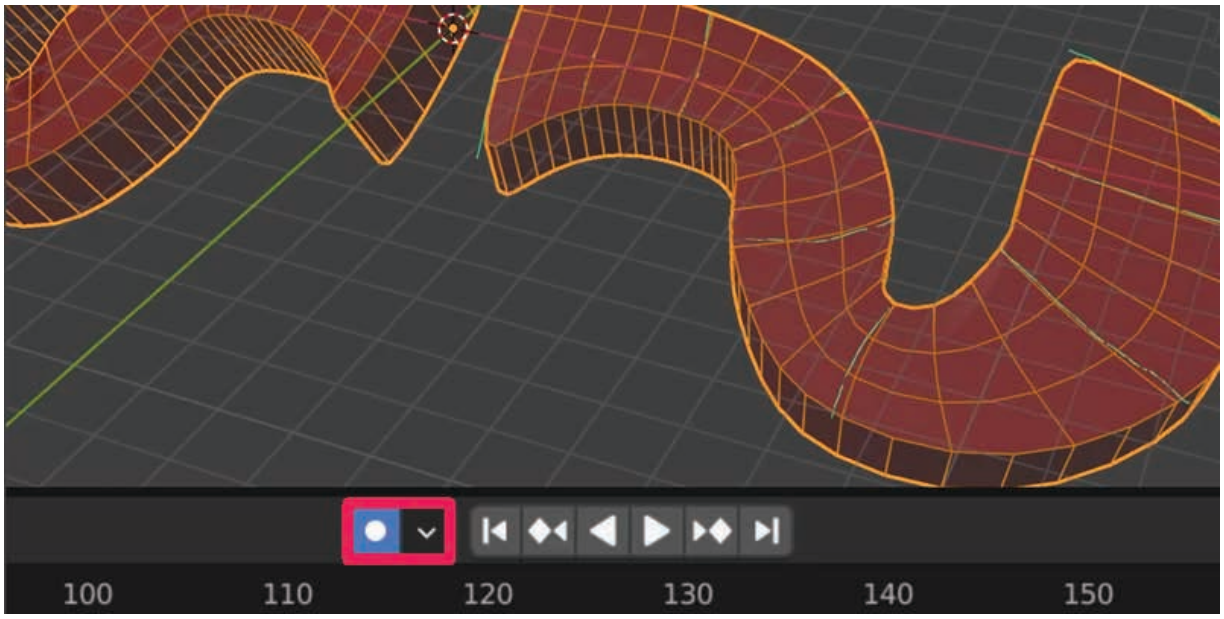


Click here to return to the index.

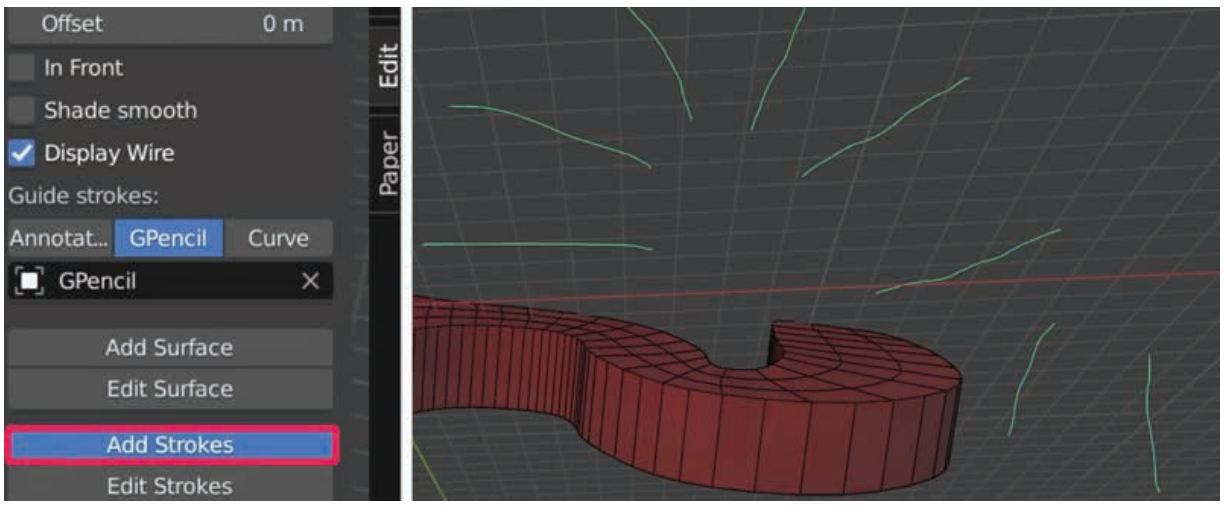
You can then adjust the Mirror axis if needed, and the thickness of the Solidify modifier.



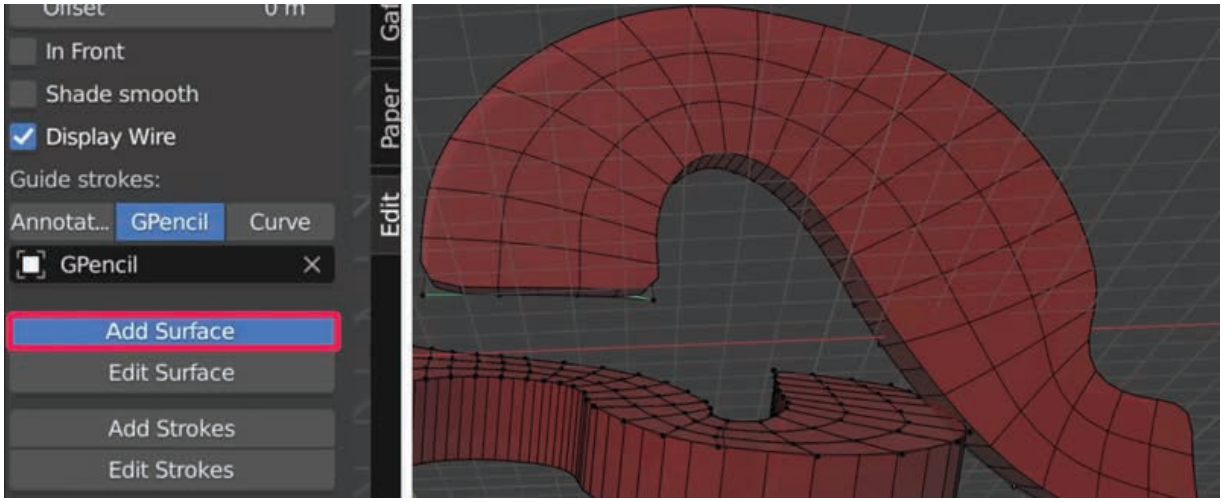
To add more geometry, enable **Auto Keying** so you can add more Grease Pencil strokes.



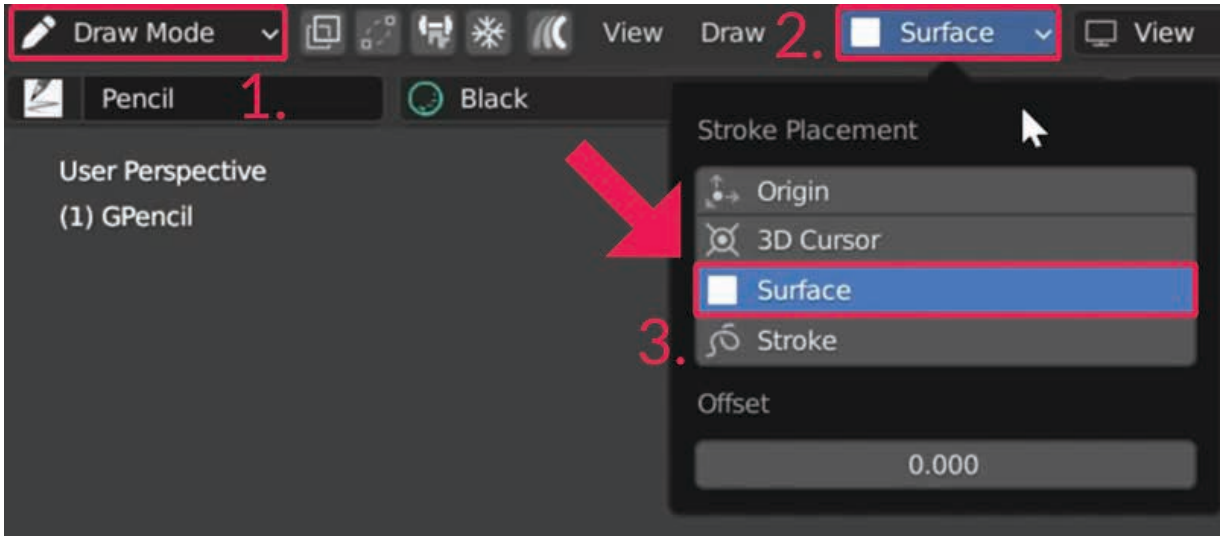
Click on **Add Strokes**.



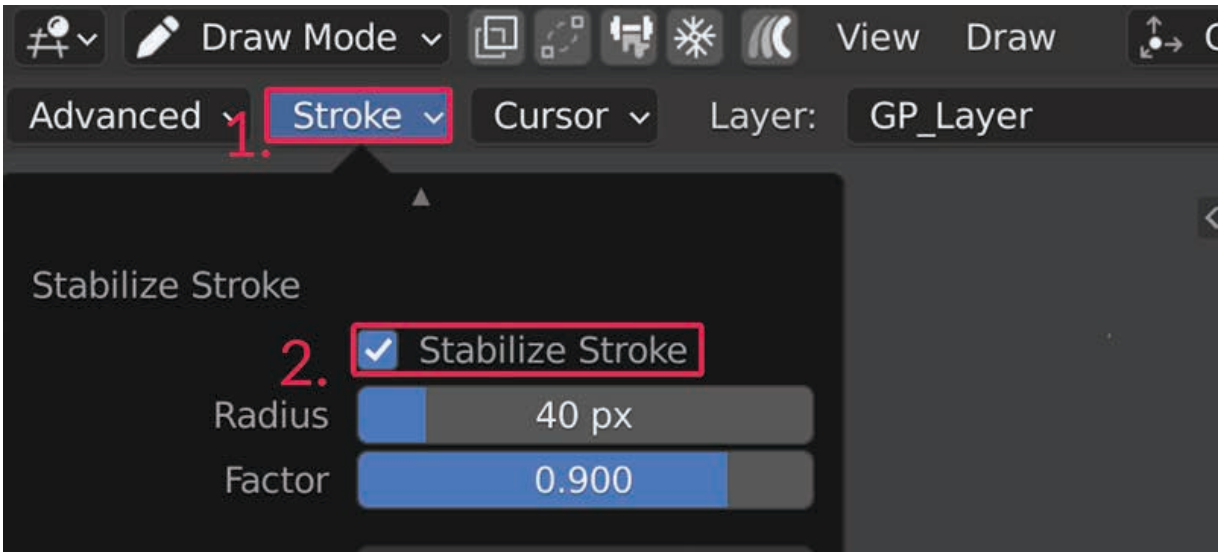
Then click on **Add Surface** again to create the geometry.



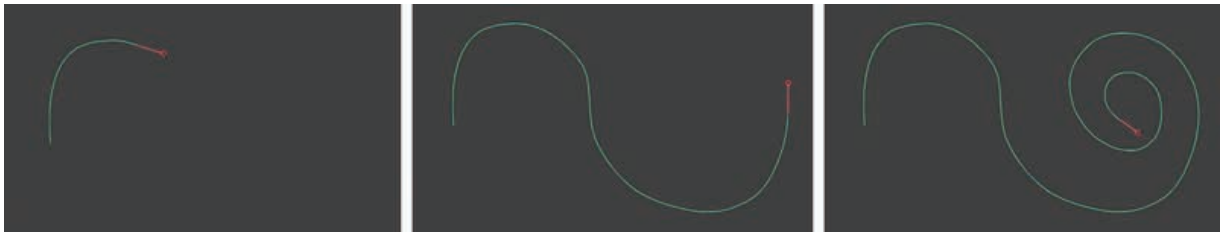
For the purpose of retopology, Annotation and Grease Pencil are very similar. Grease Pencil does have a lot more functionality than Annotation. For example, different types of **Stroke Placement**.



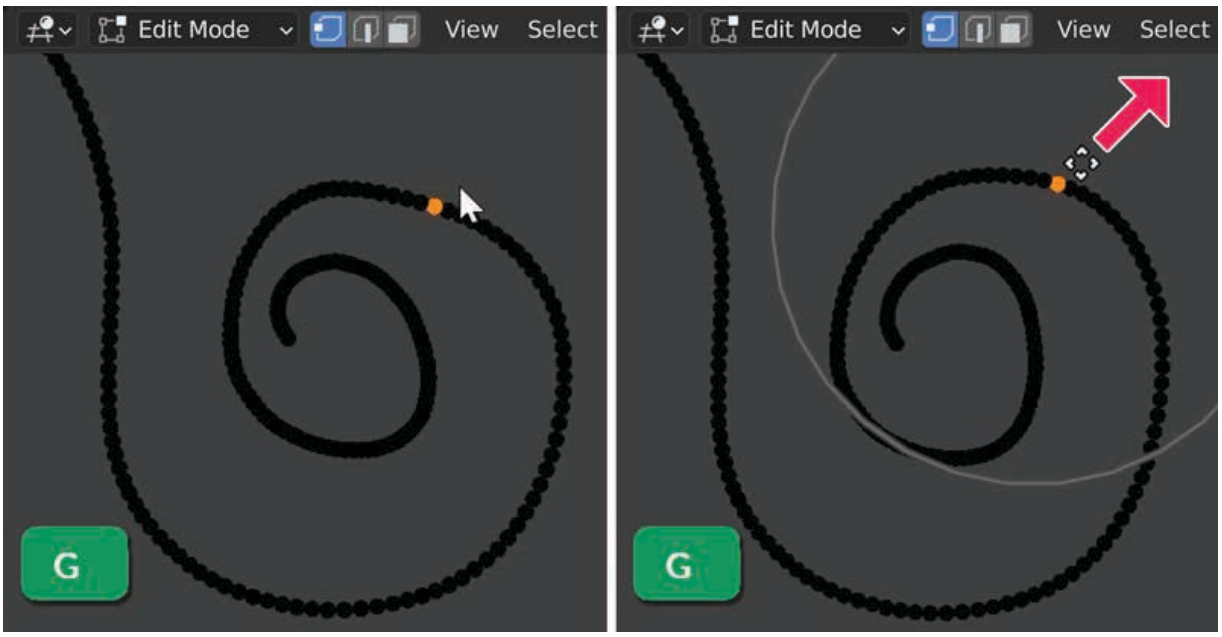
And also **Stroke stabilization**...



...which allows for smooth strokes.

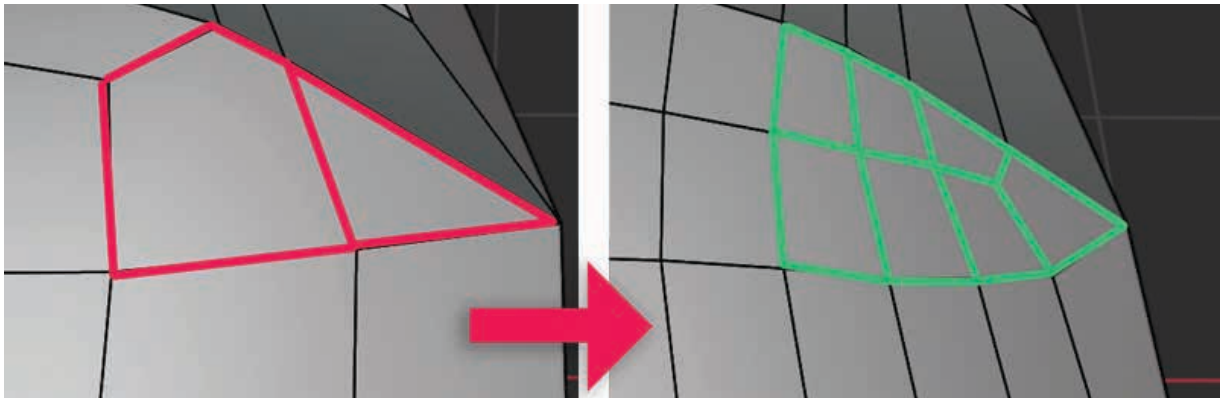


You can also edit Grease Pencil strokes in **Edit Mode**.

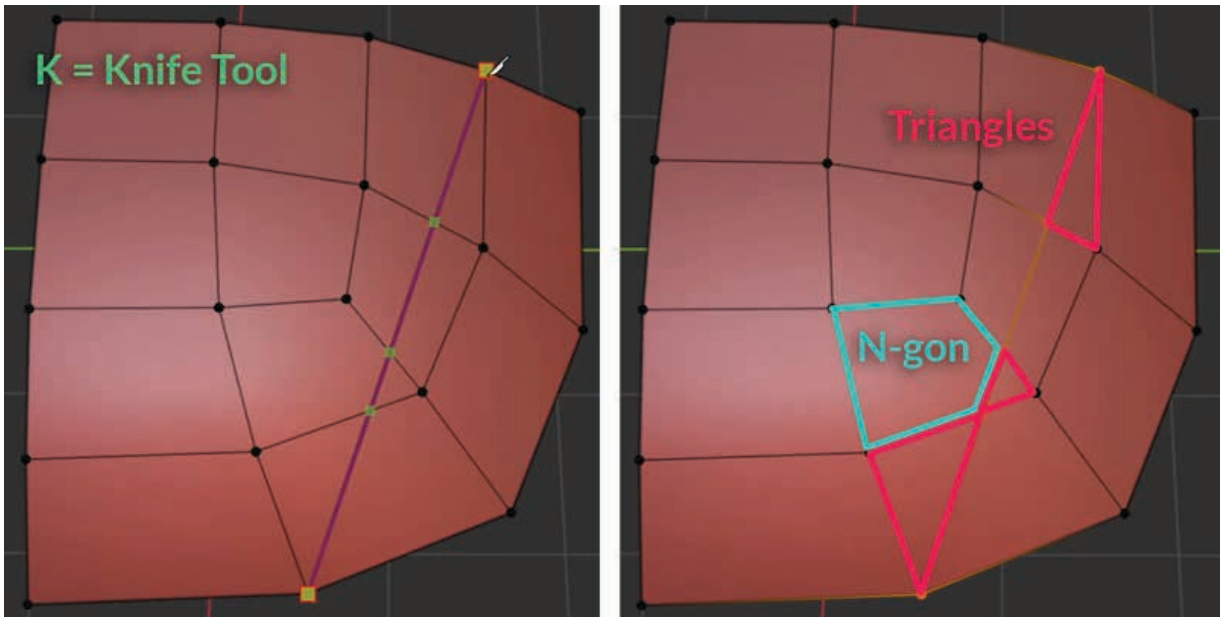


Easy trick to turn N-gons and Triangles into Quads

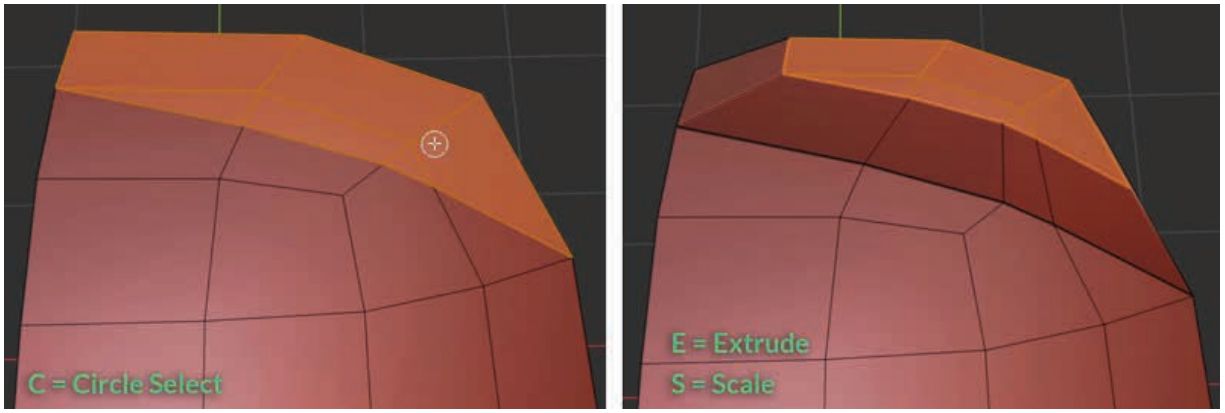
This technique is important to learn for quad-based modelling. It's an easy way to solve N-gons and triangles, at the cost of adding more geometry. So it's best used early in the modelling stage, when the overall geometry count is still low.



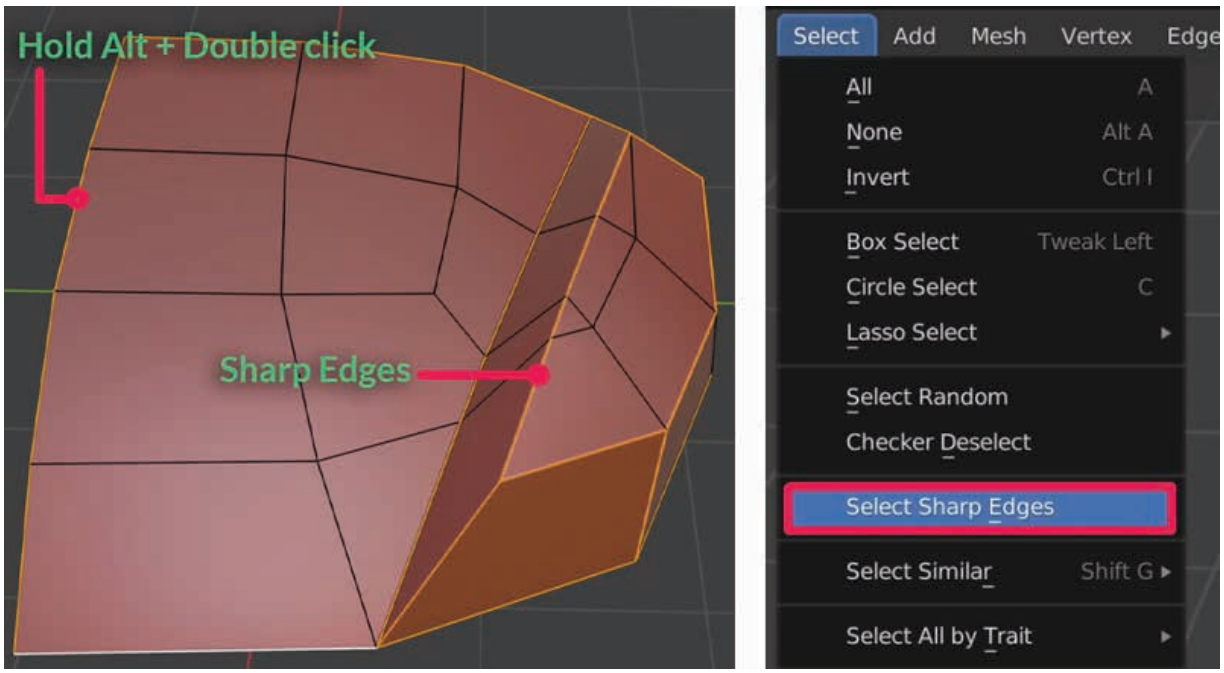
When modelling, you may sometimes create **N-gons** or **triangles**. For example when you use the **Knife Tool**.



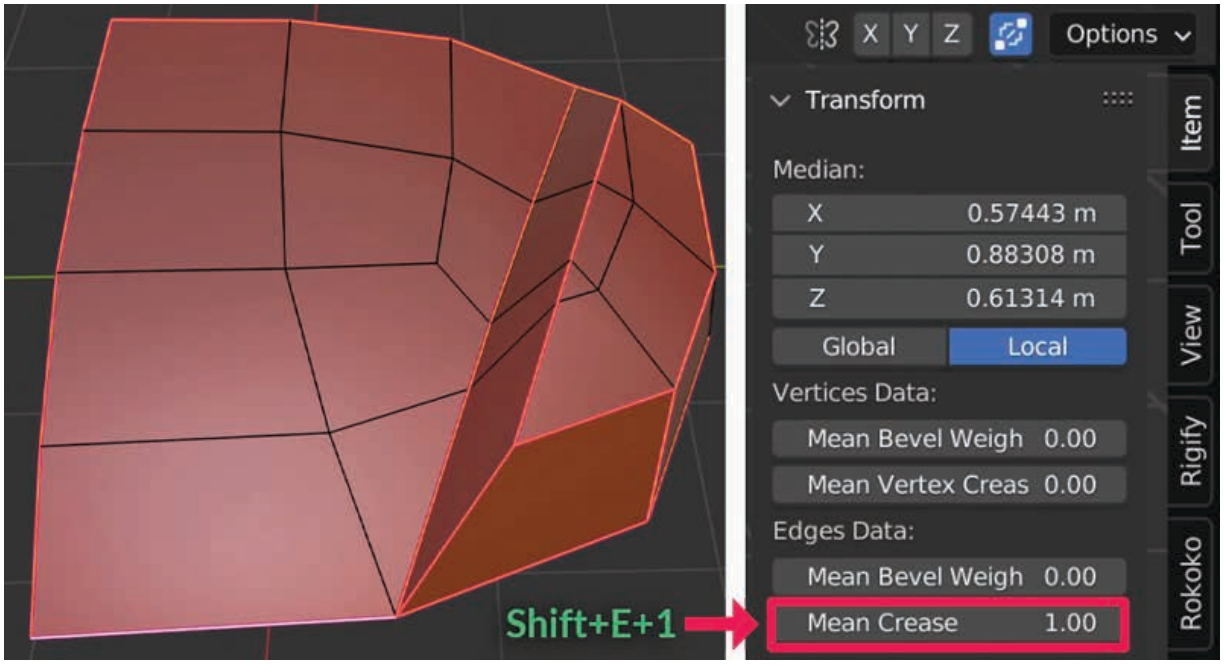
With a flat surface that isn't supposed to deform, N-gons and triangles are not necessarily an issue. Once it starts to become more three dimensional though, for example when you start extruding parts, you generally are better off with Quads.



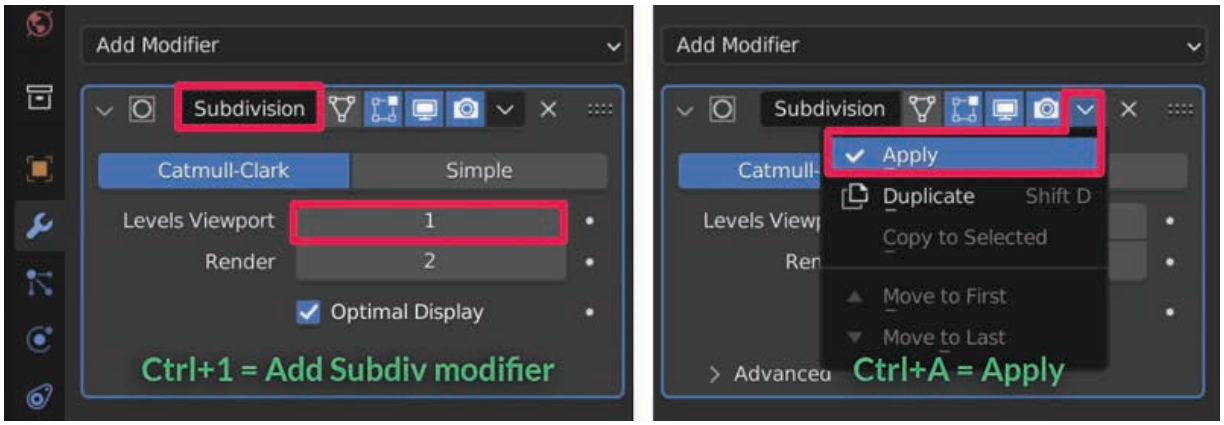
In Edge selection mode, **hold Alt and double-click** on the boundary loop to select it. Then go to Select > **Select Sharp Edges**.



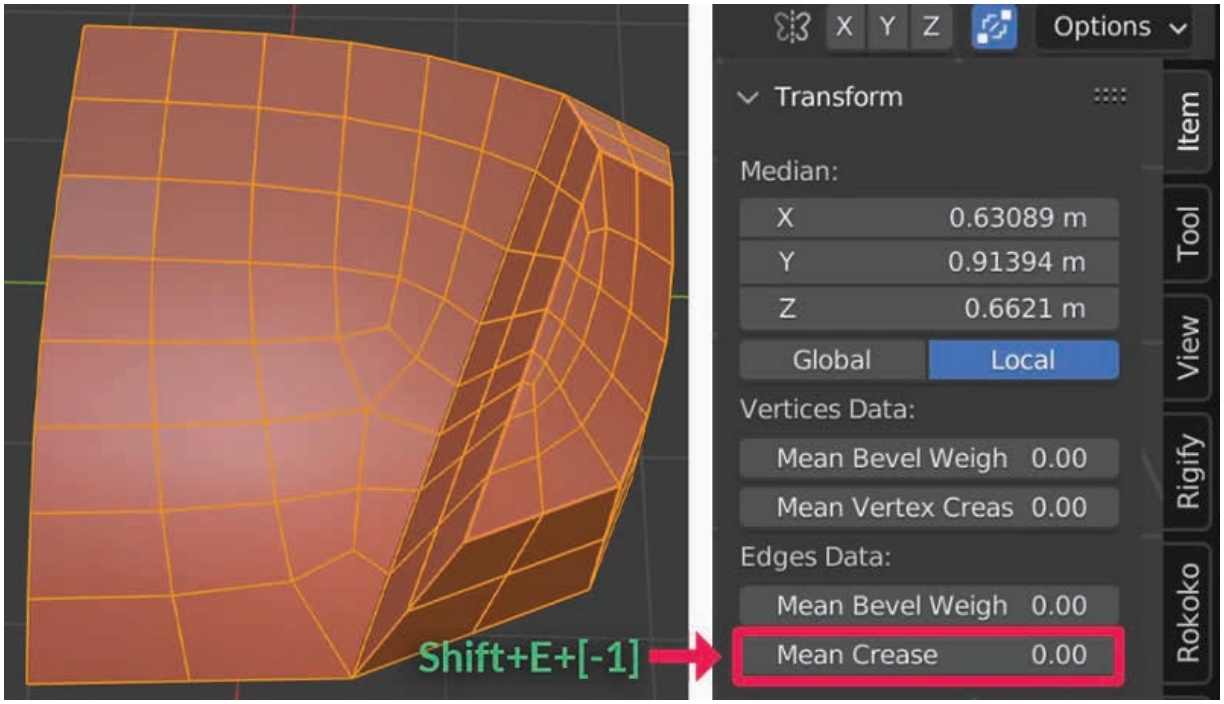
To protect these edges from the Subdiv modifier, **press Shift+E+1 (on the numpad)** to crease them or slide the **Mean Crease** value in the Option Panel to 1.



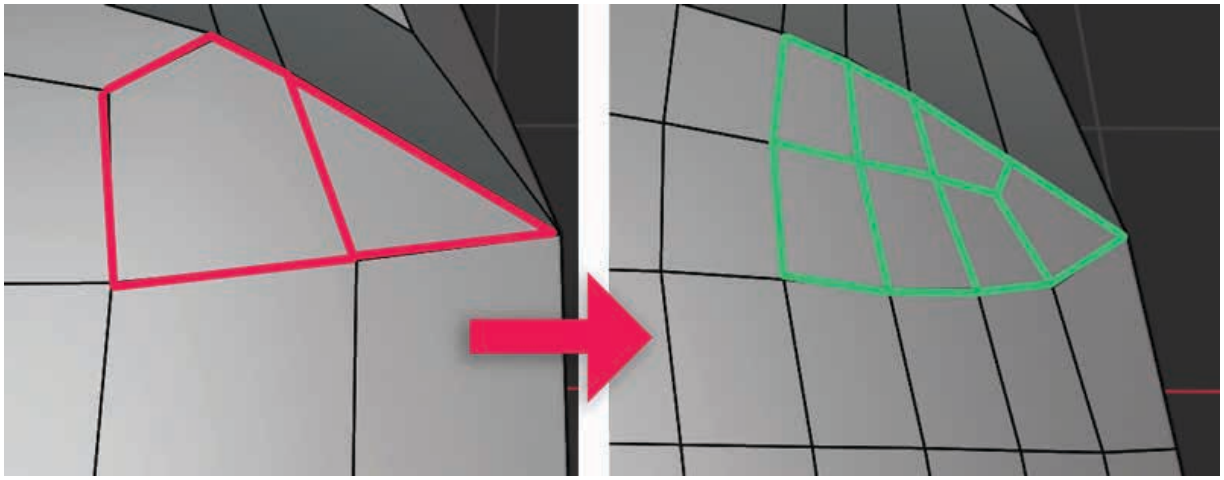
Add a **Subdivision modifier**, and then **apply** it in Object Mode with one level of subdivision.



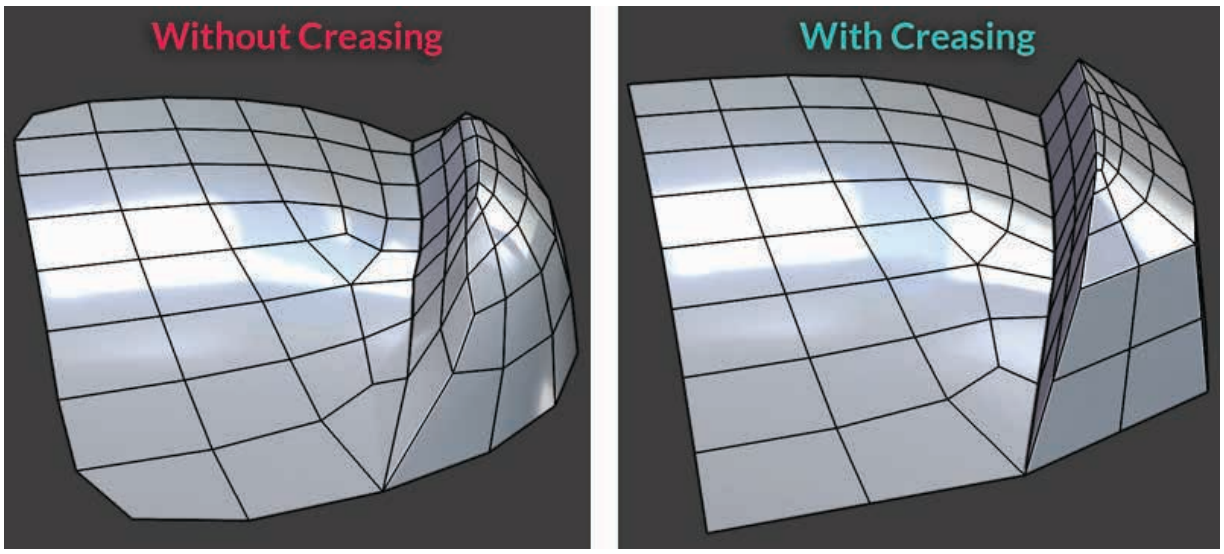
Remove the creasing again by pressing **Shift+E+[-1 on the numpad]** in Edit Mode or by sliding the **Mean Crease** value to 0.



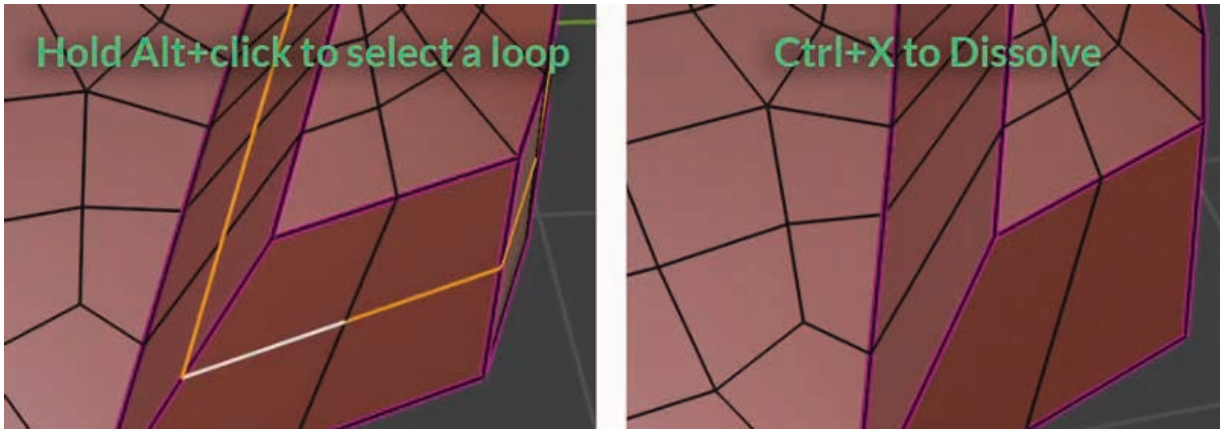
Now the N-gons and triangles are converted to quads and the sharp edges are preserved.



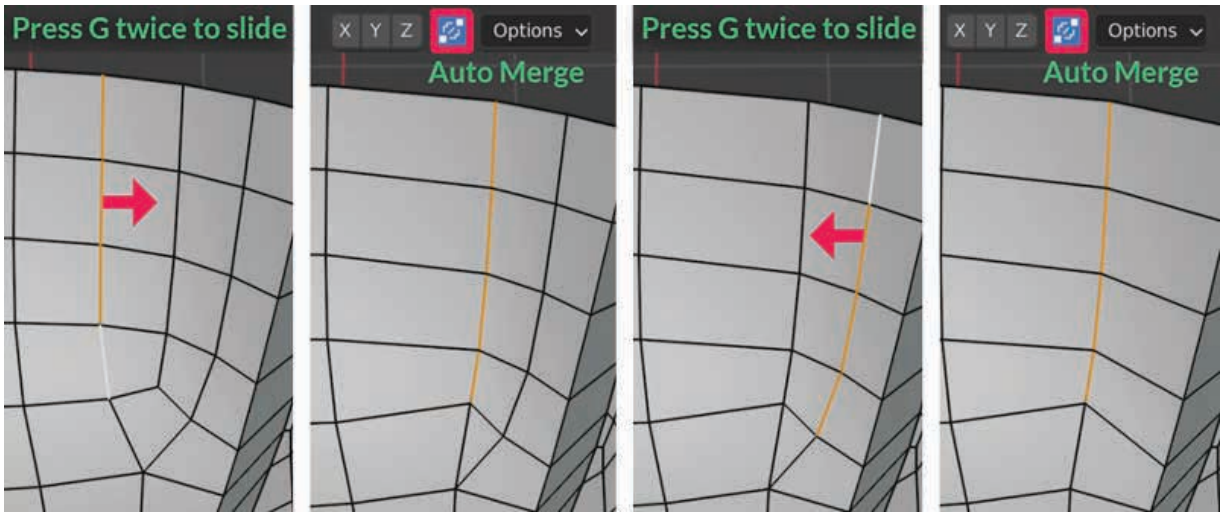
For reference, this is the difference between applying 1 level of subdivisions with, and without creased edges. The creasing really protects the shape.



If you want, you can reduce the amount of new geometry by dissolving some edges. Select an edge loop by **holding Alt** and clicking on it, then **press Ctrl+X** to dissolve it.

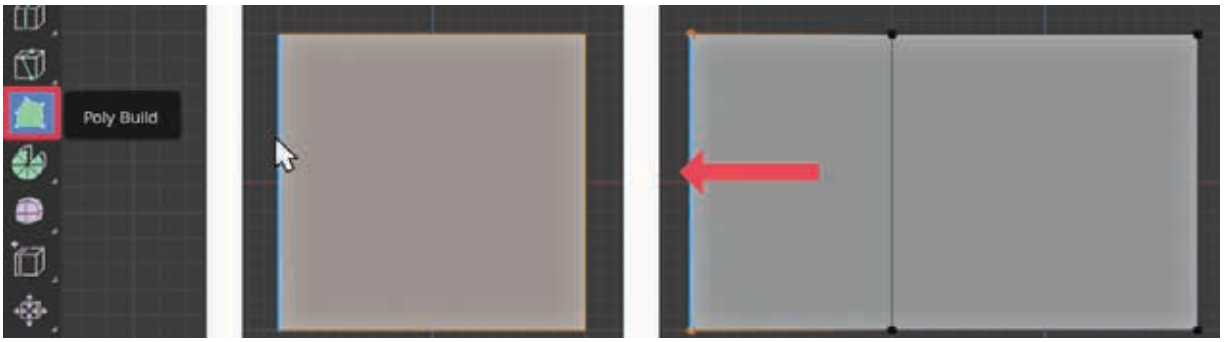


You can also reduce geometry by sliding edges or vertices by **pressing G twice** with **Auto Merge** turned on. Be careful that you don't dissolve geometry that's important for protecting the shape.



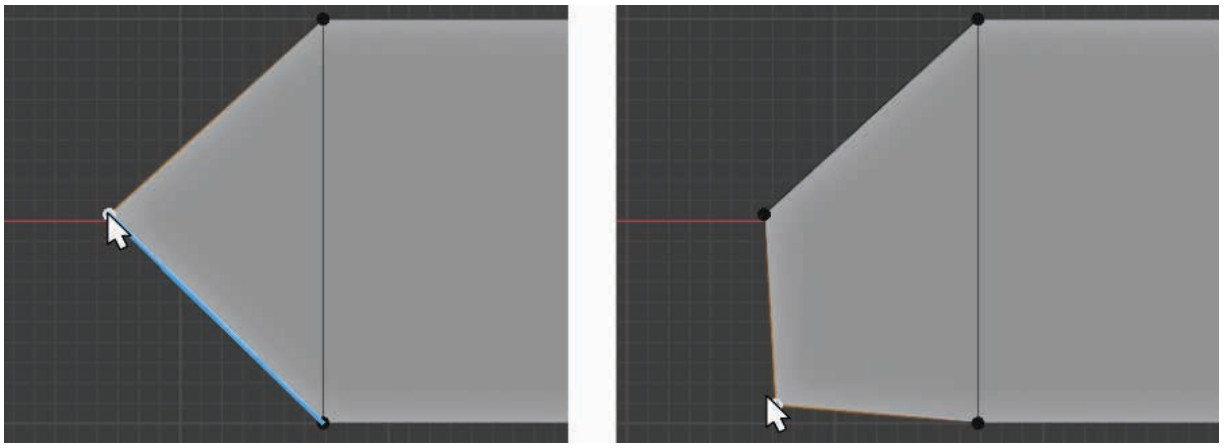
The Poly Build Tool

The Poly Build Tool is a tool that allows us to quickly add faces (polygons) to a model in Edit Mode. To enable it, go to Edit Mode, then click on the Poly Tool icon in the Tool bar. Then, when you hover the mouse cursor over an edge, it will be highlighted in blue. Click on it and drag to create a new face.

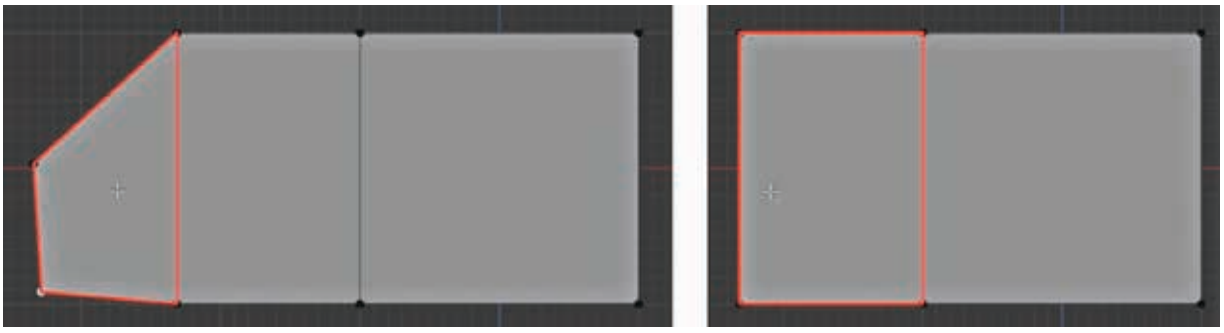


Ctrl+drag to extrude a **triangle**. Keep **holding Ctrl** to enable snapping.

Repeat that on an edge of the triangle to turn it into a face.

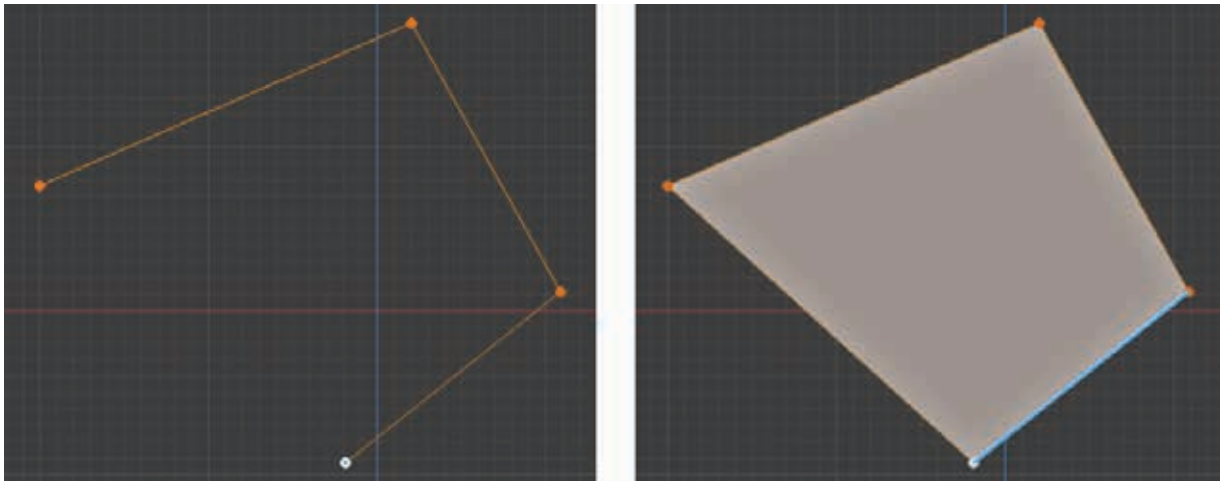


If you **hold Shift** while hovering the mouse over a face, it will get a red border. **Left-click on the face** to remove it.

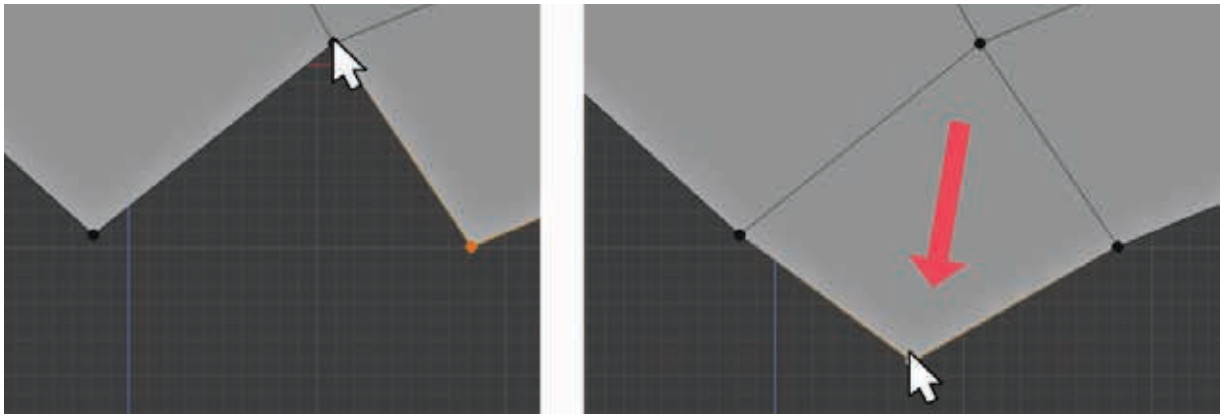


Note that the Poly Build tool works on open edges, so it won't do much on a Cube for example. Unless you **hold Shift and click on a Face** to remove it, after which you can use the tool on the open edges.

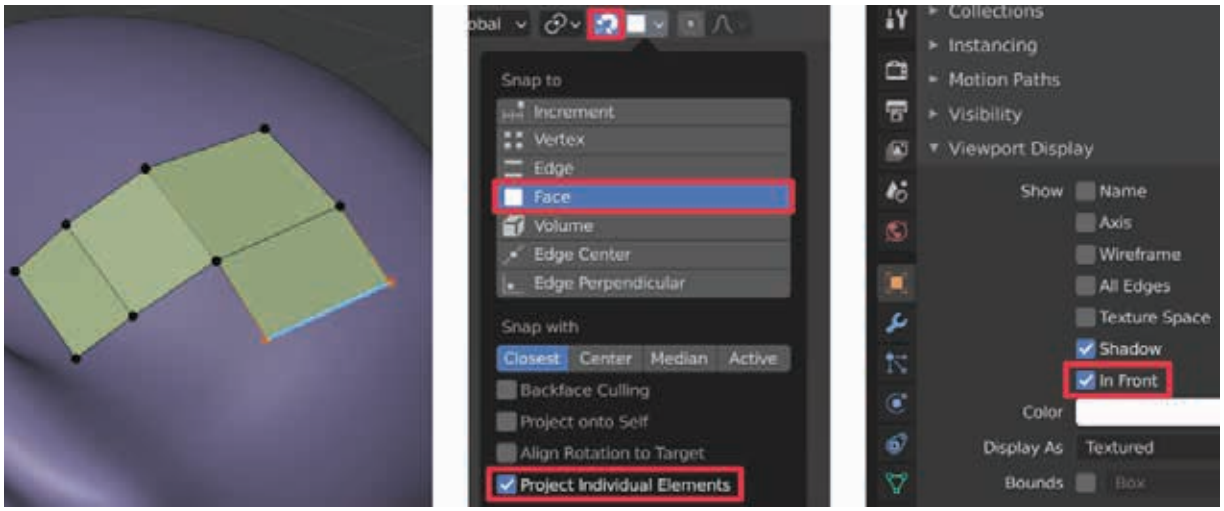
If you remove all the faces, you can create new vertices by **holding Ctrl and left-clicking**. With all the vertices selected **press F** to fill them with a Face.



Hover the mouse over a corner vertex, **hold Ctrl and left-click** on it to drag out a new face from that corner. Let go of Ctrl while moving the vertex unless you want to use snapping.

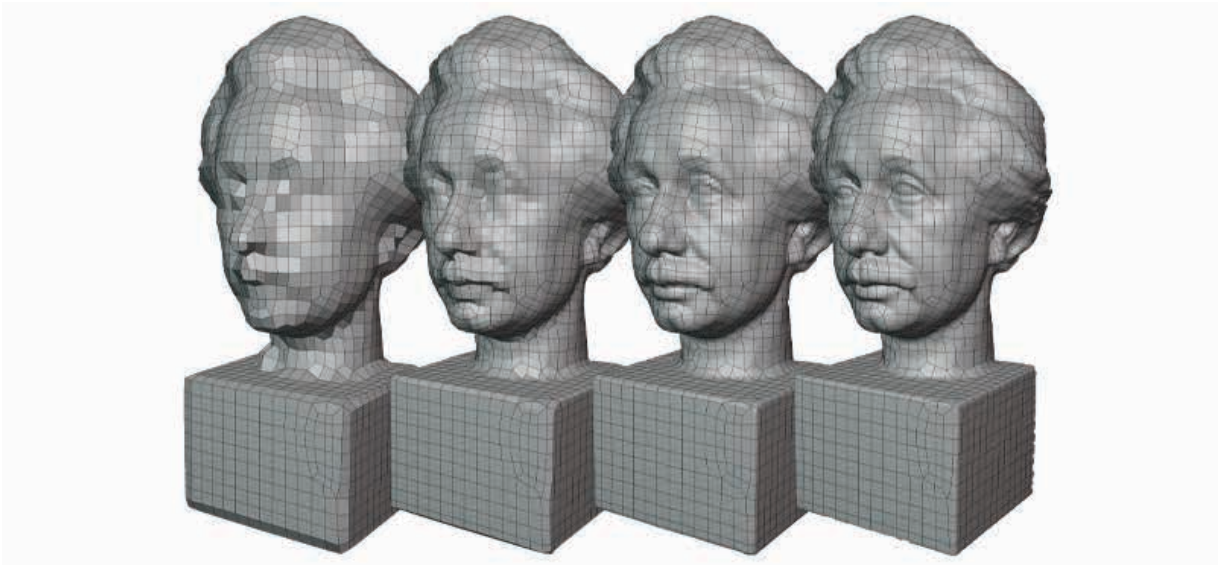


One way to use the Poly Build tool is for retopology. In such a case, turn on **Snapping** to **Face** and check **Project Individual Elements**. Set Viewport Display to **In Front** for the retopology mesh so you can always see it.

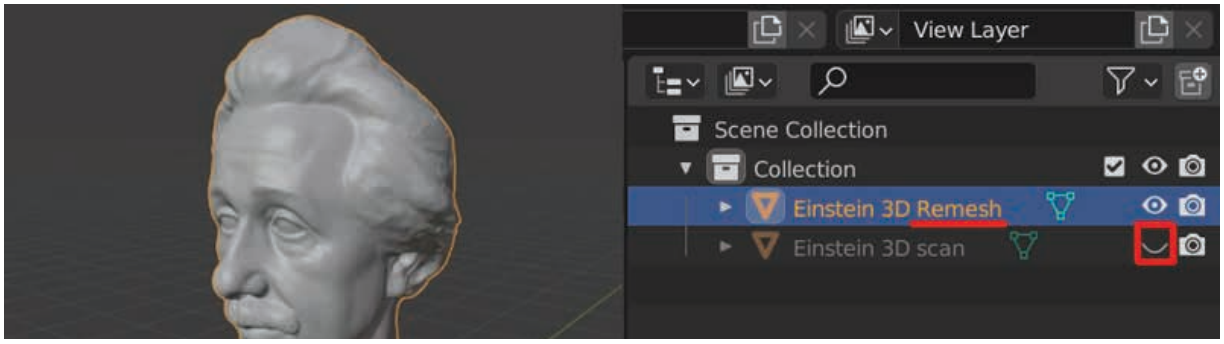


Reconstruct Lower Subdivisions from a 3D scan

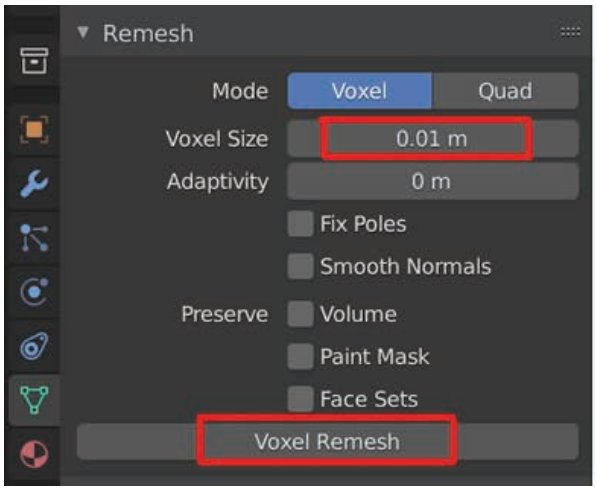
Sometimes you need to reconstruct the lower subdivision level, for example when you have a 3D scan and you want to create a low-poly model with a Displacement map. In such a case, you can reconstruct the lower subdivision levels inside a Multires modifier,

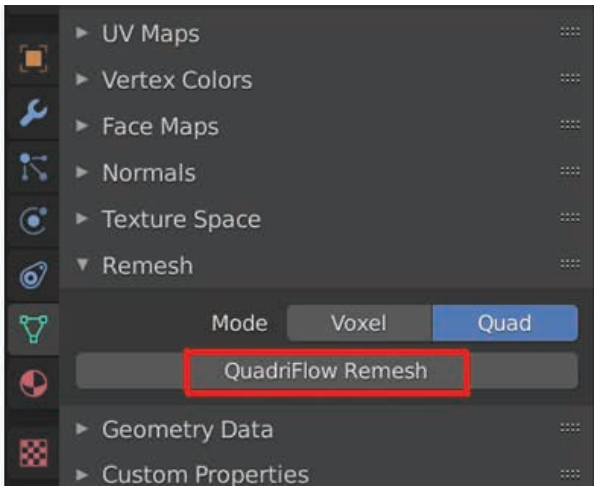


Duplicate your 3D scan and rename it in the Outliner. Hide the original 3D scan.

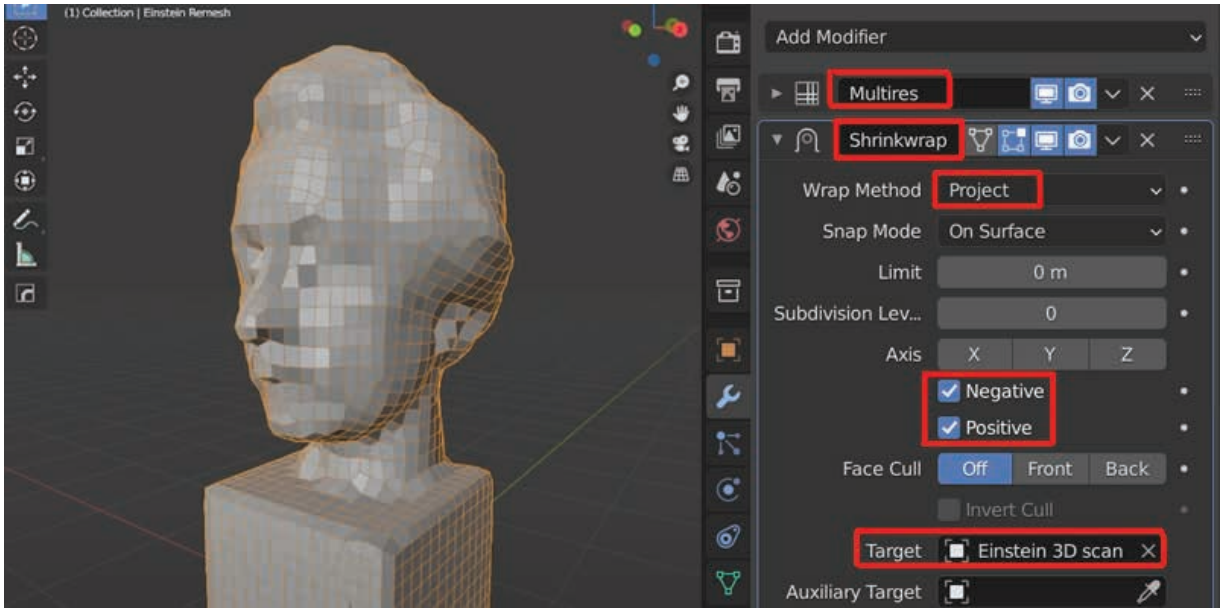


Remesh the duplicate using either the Voxel or QuadriFlow mode. Sometimes, Quad remeshing doesn't work because of problems with the mesh like holes, or double vertices. In such a case, doing a Voxel remesh first tends to solve the issue. Just make sure not to lose details, by setting the value low enough.

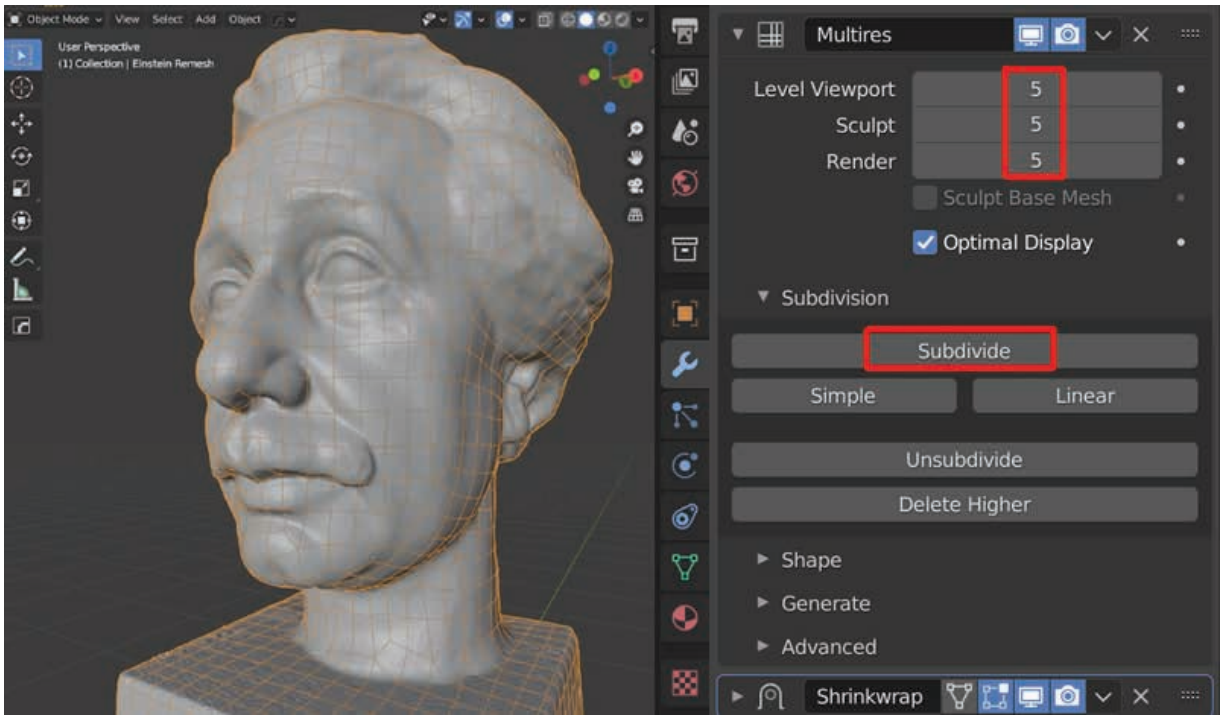




Add a **Multires modifier** and a **Shrinkwrap modifier**. Set the Shrinkwrap modifier to **Project** and check both **Negative** and **Positive**. As the **Target** use the original 3D scan.



Increase the subdivisions on the Multires modifier by clicking on **Subdivide**. Now you have all the reconstructed subdivision levels with Quad topology.



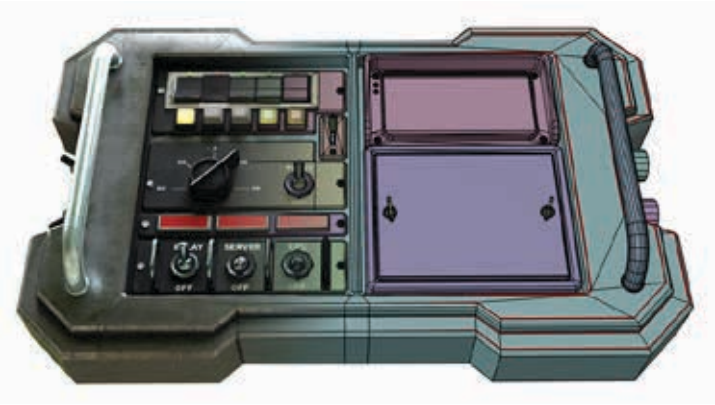
If you want you can now apply the Shrinkwrap modifier (you will still have all the projected detail in the highest subdivision level) and delete the original 3D scan.

UV UNWRAPPING

Tips for Marking Seams

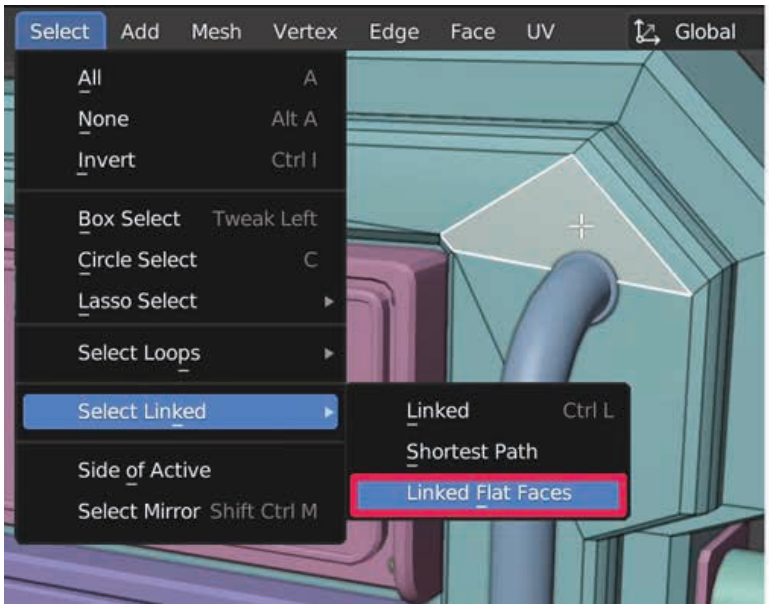
Marking seams for unwrapping can be tedious, especially if you do it manually.

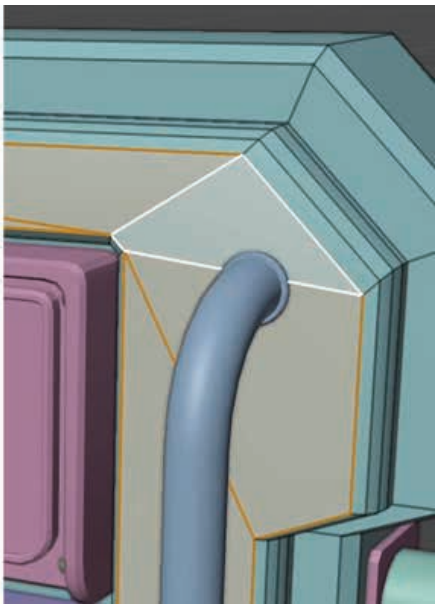
That's why you should use these three tips for selecting edges of complex areas easily.



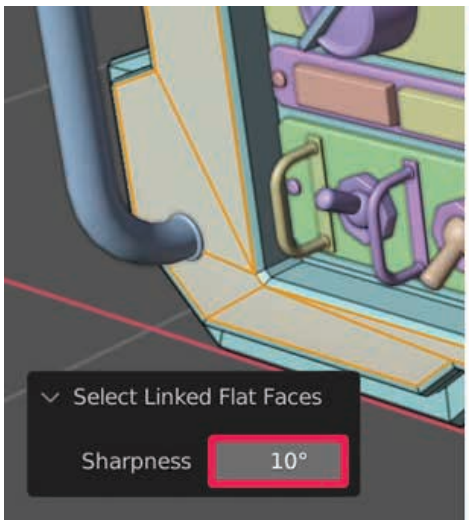
Console 3D model by LuddePudde on Sketchfab

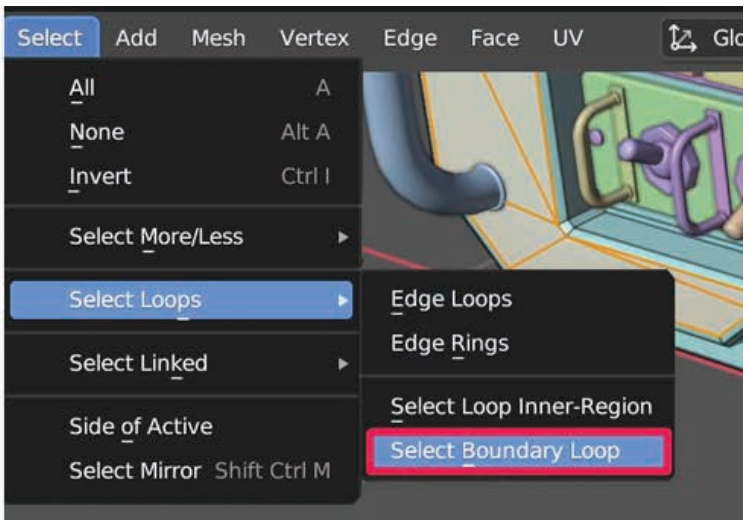
Select one Face on the area you want to mark seams on. Go to **Select > Select Linked** and choose **Linked Flat Faces**.



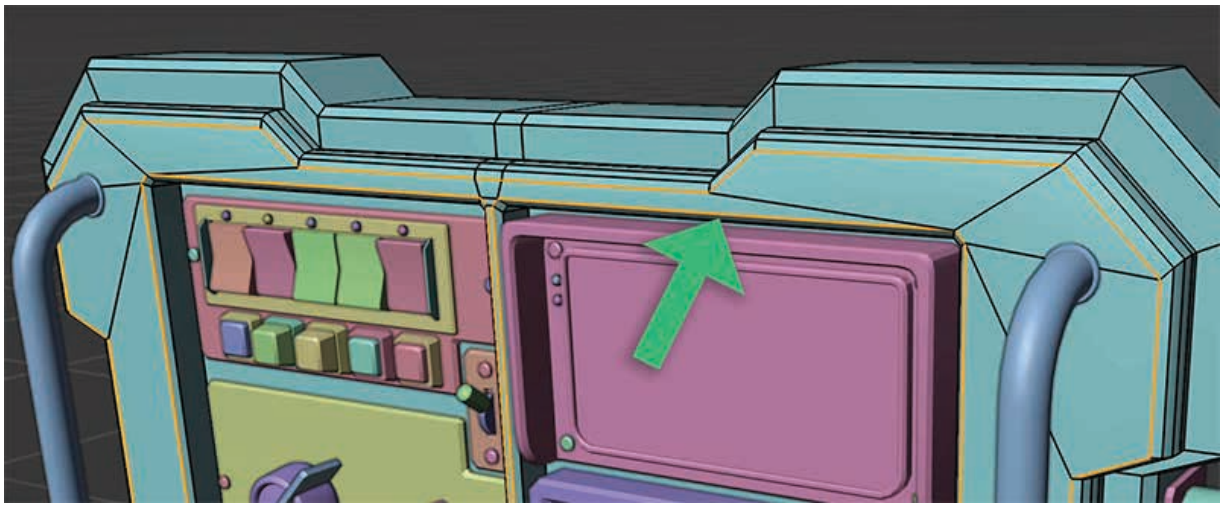


In the Operator Panel you can adjust the angle of Faces to include in the selection. Go to **Select > Select Loops > Select Boundary Loop**.

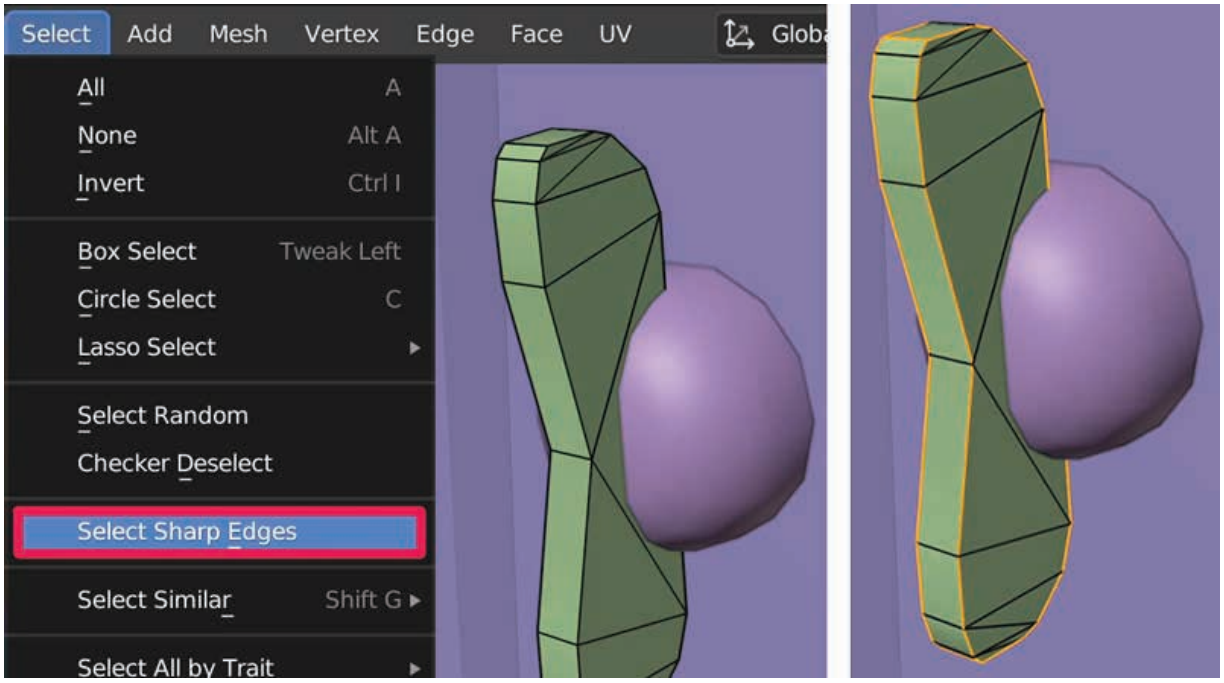




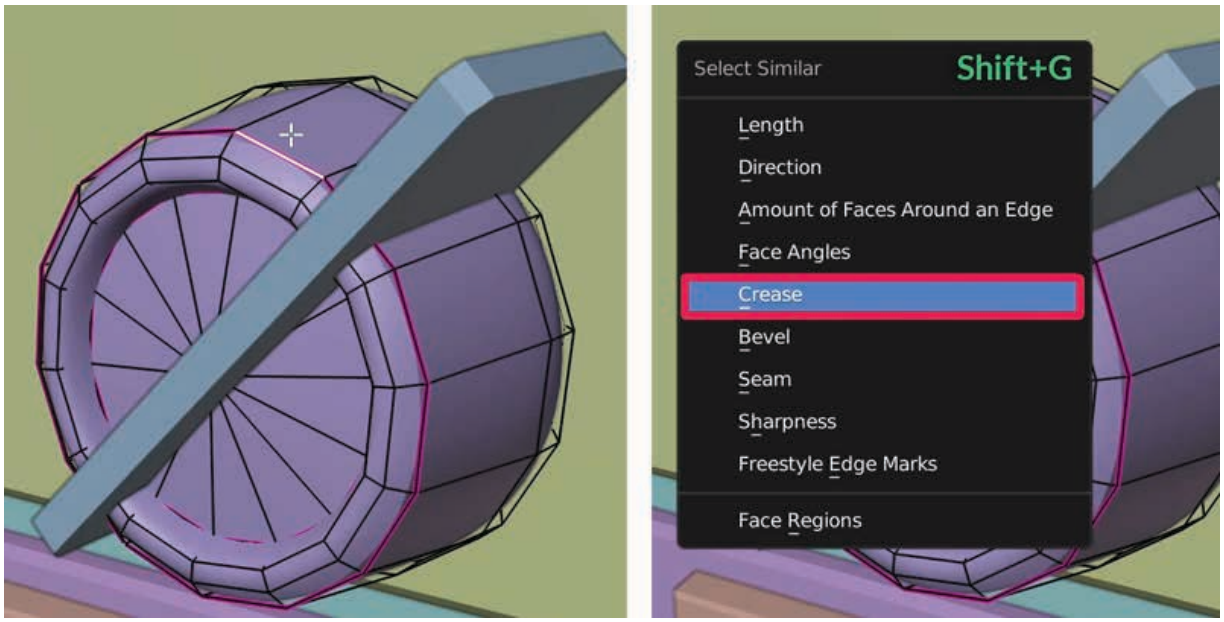
Press Ctrl+E for the Edge menu and choose **Mark Seam**. Now the edges are marked as seams without having to select them all by hand.



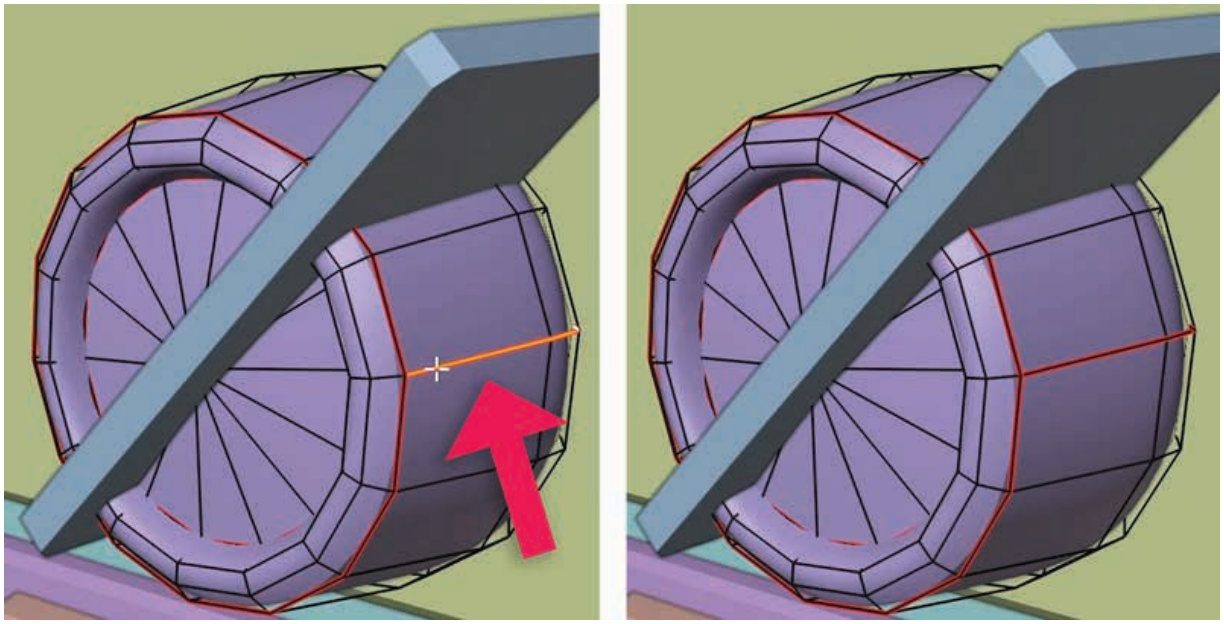
Another useful trick is going to **Select > Select Sharp Edges**. This selects all edges with a specific angle, which you can adjust in the Last Operator Panel (F9). You'll probably need to manually add or remove some edges from the selection. Then **press Ctrl+E** for the Edge menu and choose **Mark Seam**.



If you already have used creasing on some edges, these can also be a good starting point for unwrapping. Select one creased edge, **press Shift+G** and choose **Crease** to select all creased edges. Then **press Ctrl+E** for the Edge menu and choose **Mark Seam**.



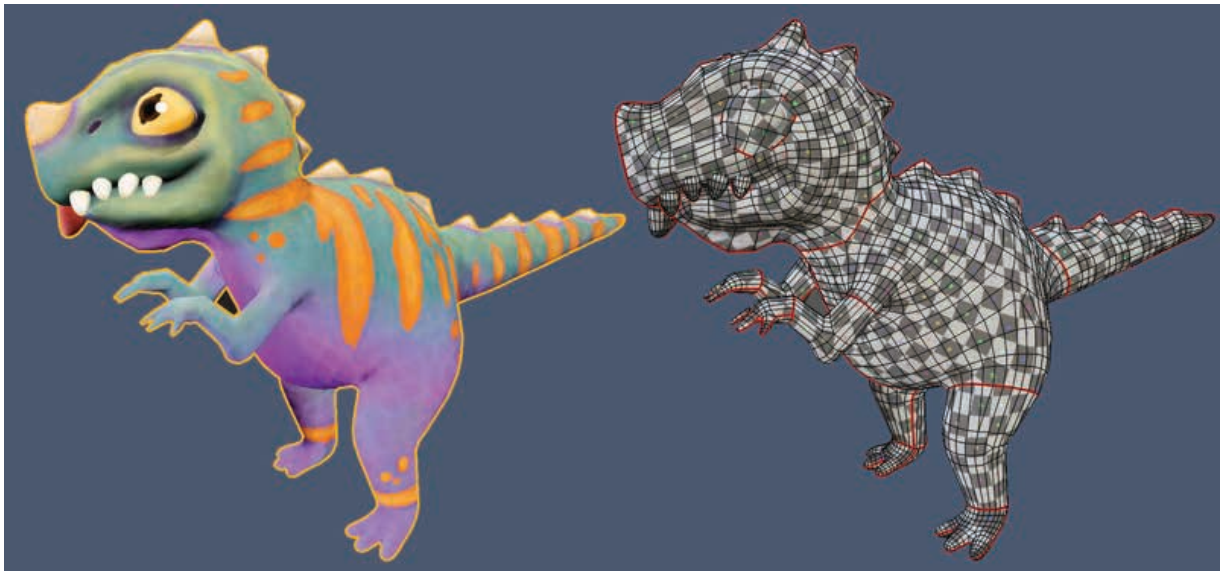
In the latter two cases you usually have to manually select or deselect some edges. Then **press Ctrl+E** for the Edge menu and choose **Mark Seam**. But it's better than having to select everything manually.



Finally select all and **press U > Unwrap**.

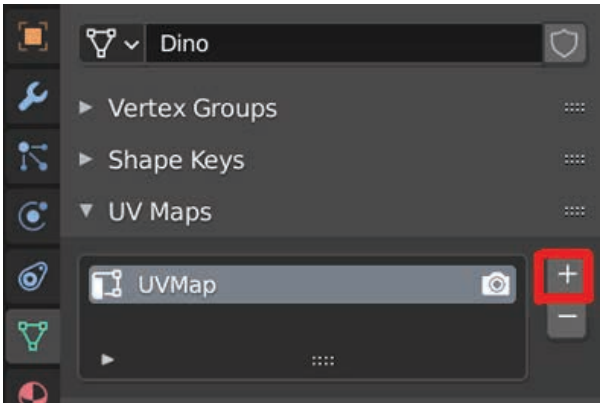
Unwrap like a pro using Live Unwrap

Here are some tips for unwrapping that will speed up your unwrapping workflow, and make it more fun as well.

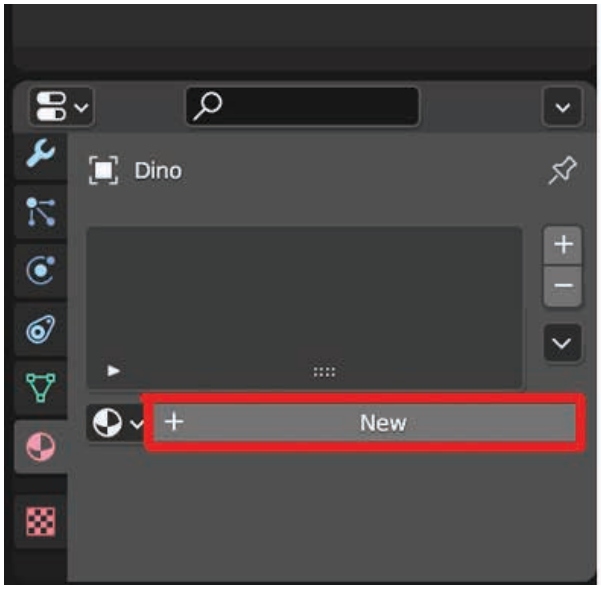


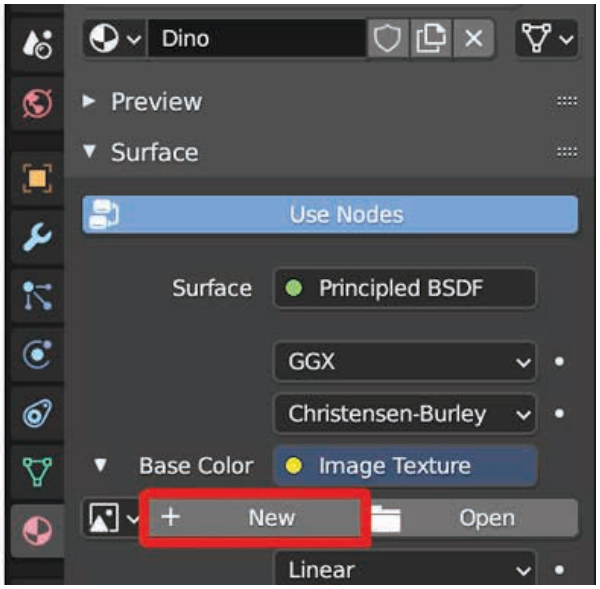
Before we can give a material to the character, it needs a UV Map.

Create one by **clicking on the + icon** in the UV maps panel.



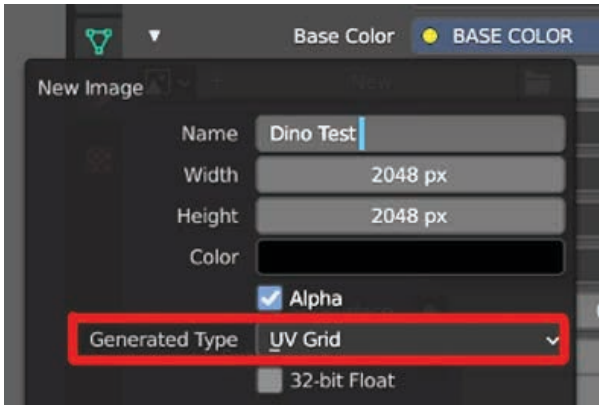
Then give it a material, by clicking on **New**. Then, as the Base color choose **Image Texture**. Click on **New** to create a new Texture. Give your material a name.

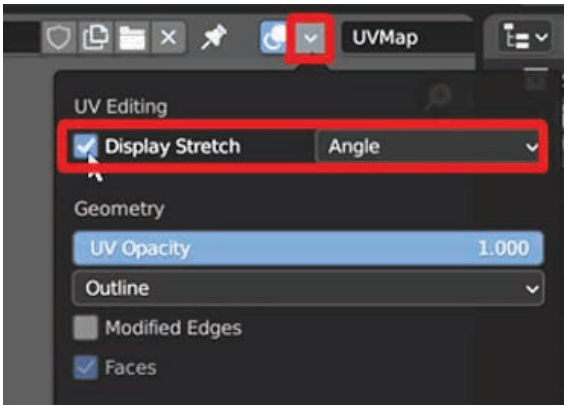




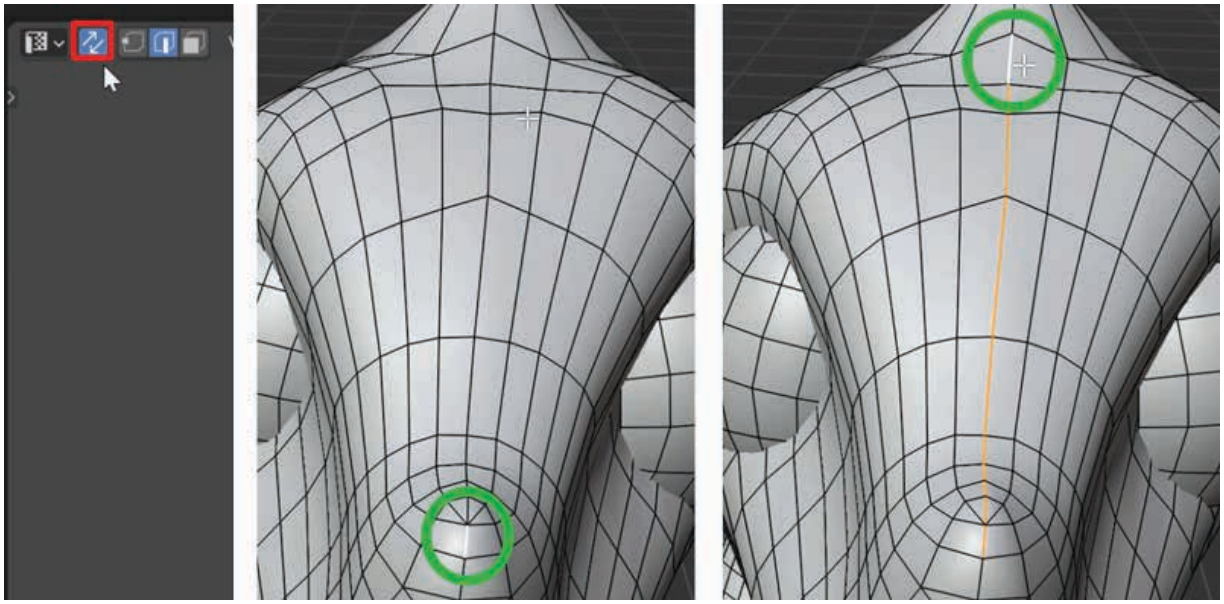
Click here to return to the index.

I make my texture 2K, with **UV Grid** as the **Generated type**. This will make it easy to see where the texture may be stretching on the model once it's unwrapped. I also give it a name here. Another way to check stretching is by checking the "**Display Stretch**" option in the **UV Editor window** *(this option is only available in Edit Mode)*.



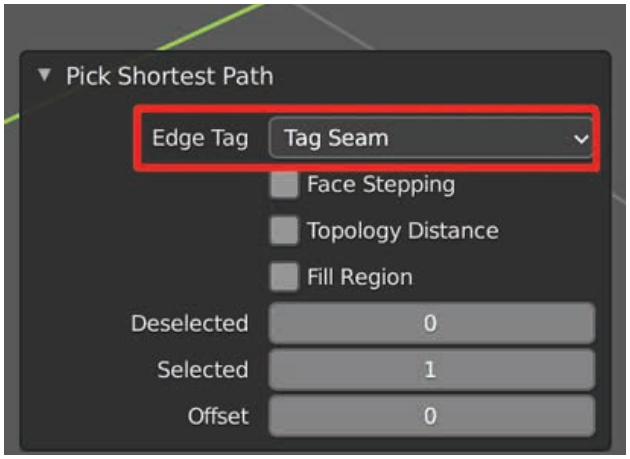


To see better what we're doing, **turn on Sync** in the UV Editor window *(this icon is only available in Edit Mode)*. This way all the UV's are visible, not just when they're selected. Then, select an edge and while **holding Ctrl**, select another edge. As you can see all the edges in between are automatically selected.

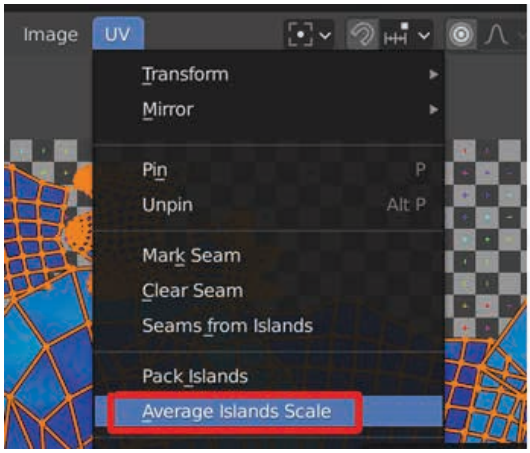


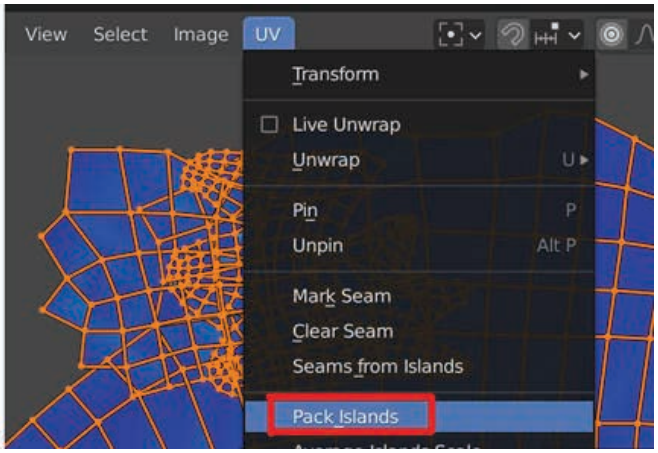
That's not all though. A menu has appeared in the bottom left corner (if not, press F9). Here, you can set **Edge Tag** to **Tag Seam**.

Now, edges selected with this **Pick Shortest Path** method, will automatically be marked as seams. And since Live Unwrap is active, you'll see something has happened in the UV window.

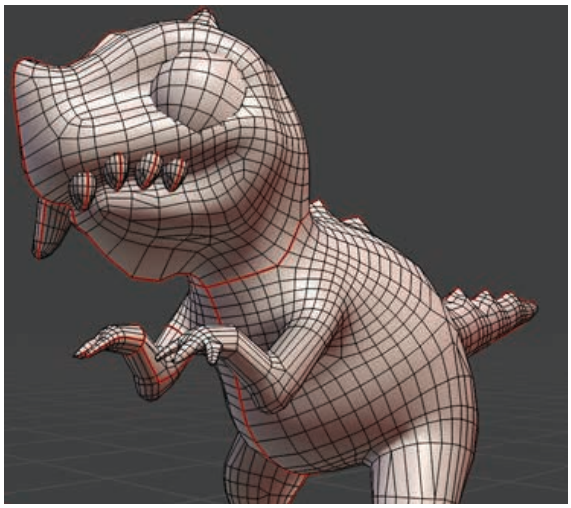


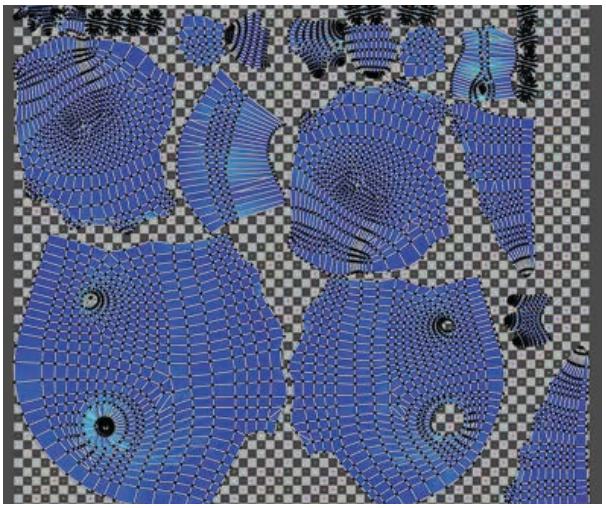
Pretty soon, you'll have marked all the necessary seams. Now, choose **Average Islands Scale** from the UV Editor options, and then **Pack Islands** to create a more efficient layout.



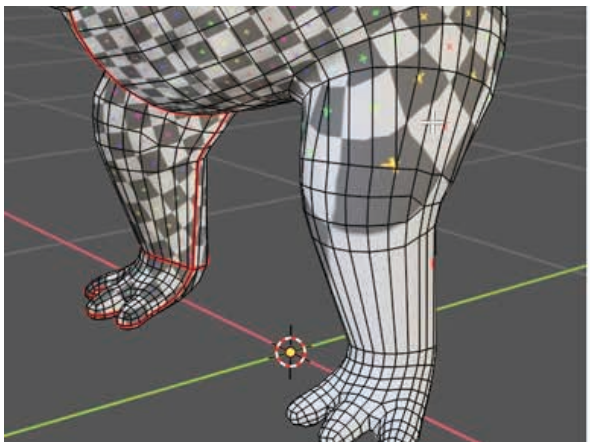


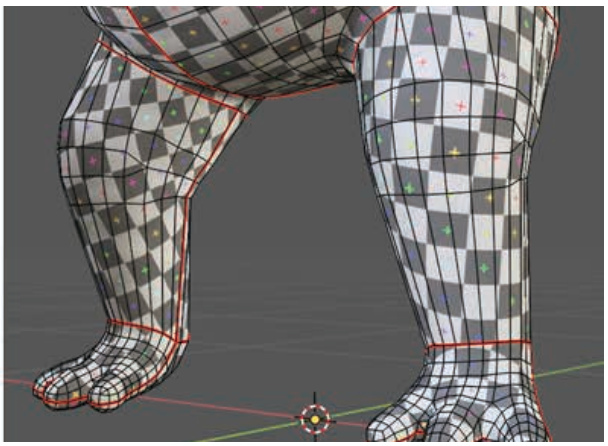
Then, you should get something like this. Now you can start texture painting!





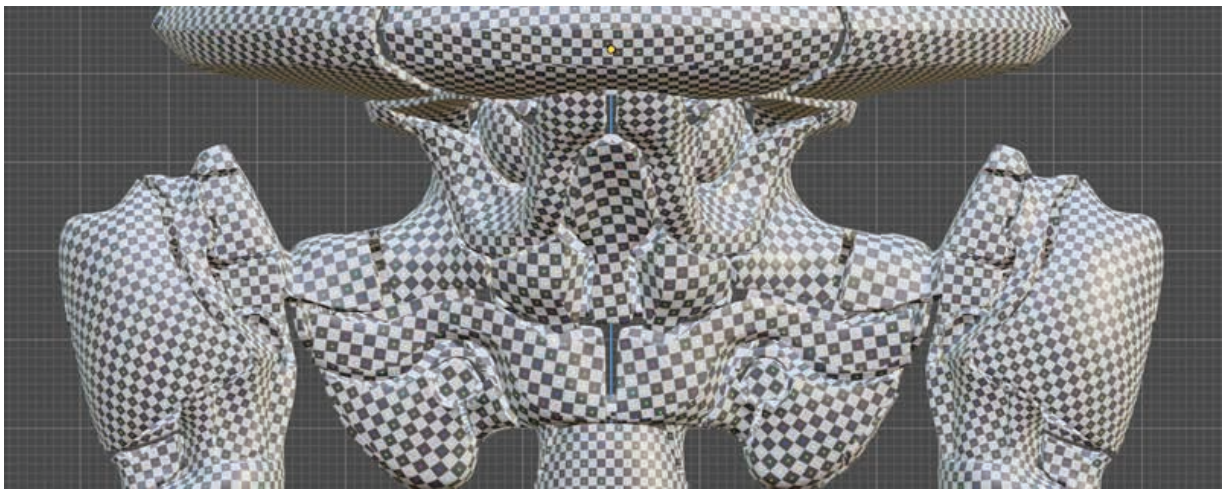
Seems good, but let's look at it in **Material preview**. It looks like there's a lot of stretching on an arm and a leg, because I forgot to mark seams there. This is easily solved by marking more seams using the Pick Shortest Patch method we used so far. As you can see, the texture is more evenly distributed as a result and there's no bad stretching.





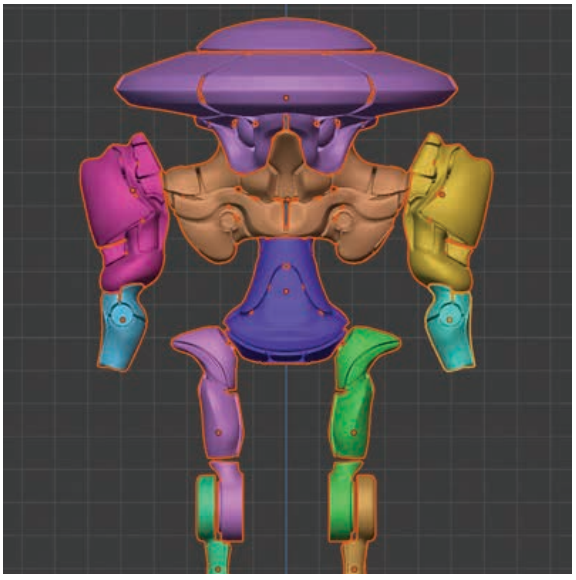
UV Unwrapping hard surface panels

Having modeled some panels for this robot model, we will unwrap it for texturing. How many textures you want to use in a model depends really on how close you want to get with your camera. Some creatures in movies these days have hundreds or even one thousand 4K textures for one model.



So one single 4K texture isn't going to cut it (unless you're making a low-poly asset for a simple game, for example).

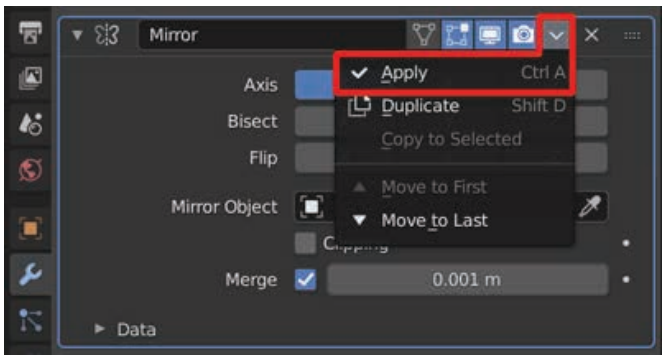
Although I don't want to use 100 textures, let's at least group the panels in distinct body parts like the upper and lower arm, head, torso and so on.



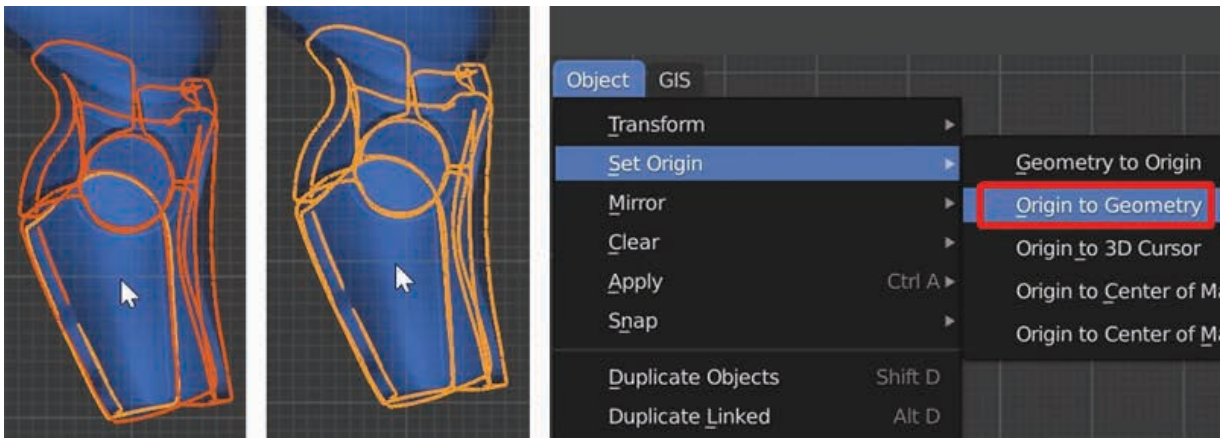
First, make sure all the Mirror modifiers have been applied. Otherwise we can't select for example the left and right arm separately.

To do this make select the mirror modifier and **press Ctrl+A** or select Apply from the drop-down menu.

Be careful, that you don't apply the Subdivision modifier by accident.

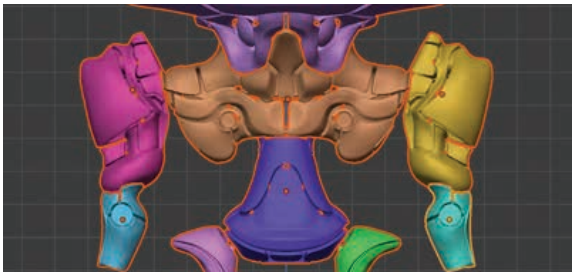


Select the parts you want to group as one object, make sure there's one active object (with an orange outline) and **press Ctrl+J** to join. I like to move the selection with G to see what I've selected. **Right-click to cancel** the transformation. After joining things, hide them by **pressing H** so you it's easier to see what's left. In case you missed the occasional Mirror modifier, you will probably notice it when joining the objects as half of that part will disappear. Unhide everything by **pressing Alt+H**. Then, set the origins to the geometry.

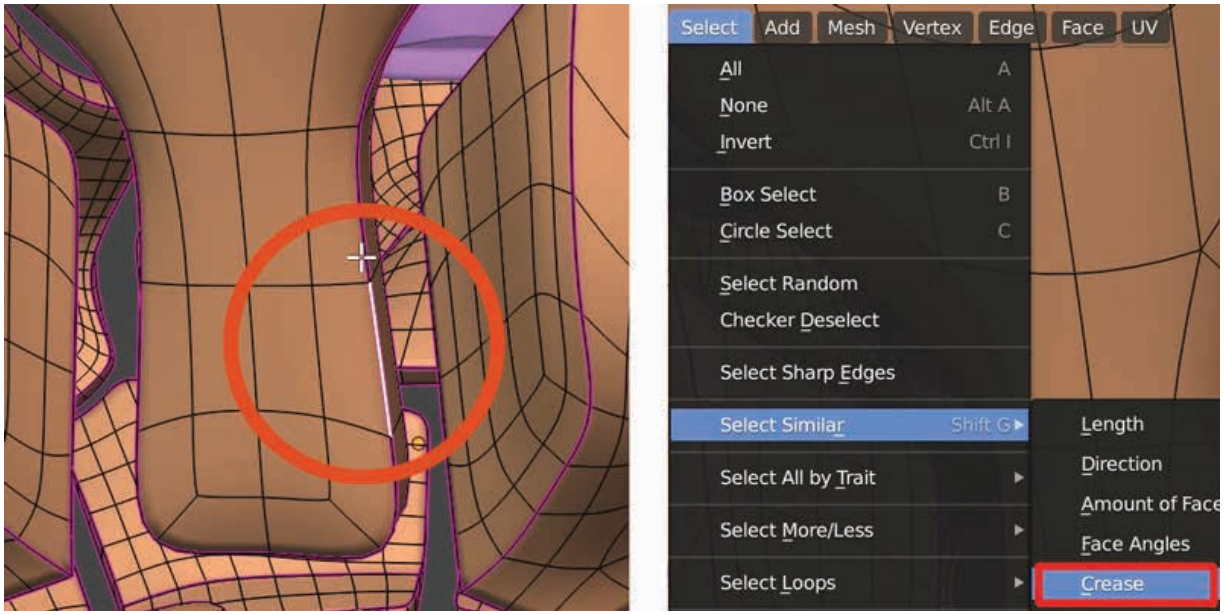


I like to give each group of objects a distinct Viewport Display color, so it's easy to see how the model is divided into textures.

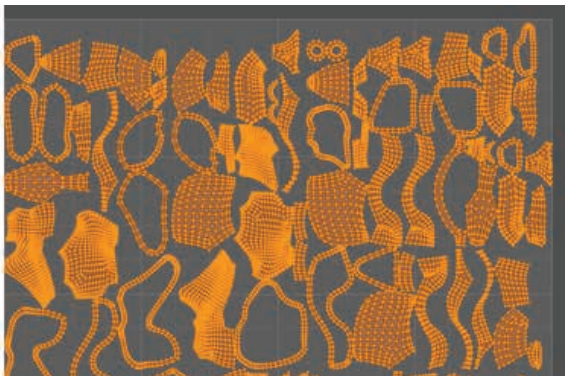
In total we've got 15 joined objects, so we'll use 15 textures, one for each joined object.



Now, it will be a lot of work to UV unwrap these panels, right? Well, good news: you've already done a lot of the work by creasing edges, because we can use those to create the seams for the unwrapping. Select one creased edge in Edit Mode, then go to Select > Select Similar (or **press Shift+G**) and choose "**Crease**". This selects all the creases.



Now go to **Edge > Mark Seam** (or Ctrl+E > Mark Seam). If you select all now by **pressing A** in Edit Mode, and **press U twice**, you'll unwrap the object. In the UV Editor window you can see the resulting UV map. Not bad at all! You can add some additional seams if you want and unwrap again.

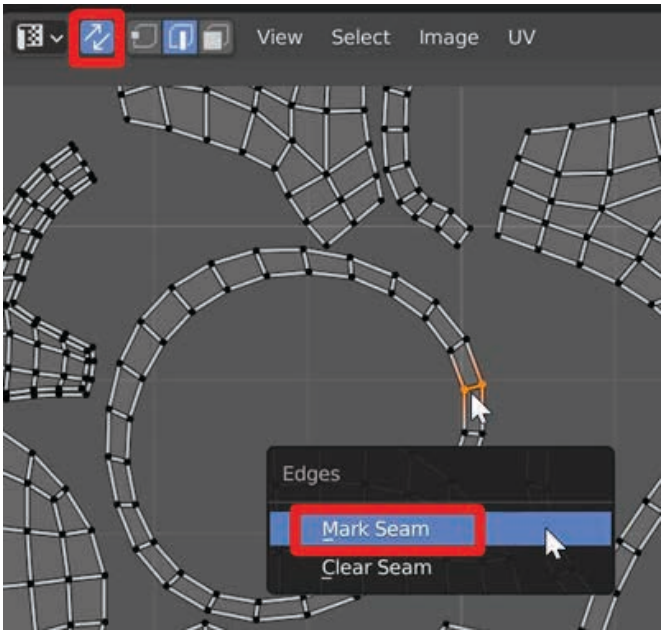


You can also mark seams in the UV Editor window.

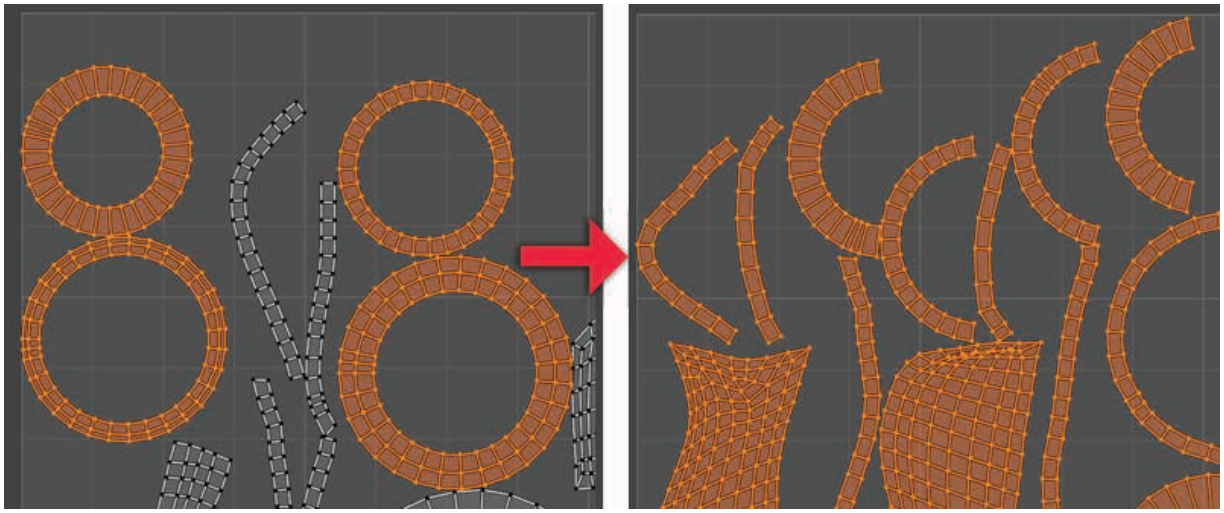
Activate **Sync** by pressing the Sync button (this is only available in Edit Mode).

Now you can see all the UV's even when nothing is selected. Then, select some edges that you want to mark as Seams and **press Ctrl+E** and choose Mark Seam (or go to UV > Mark Seam).

Press U twice to unwrap again.



Although not strictly necessary, I like to split up these circular UV loops in two halves so you can later pack the UV islands more efficiently.



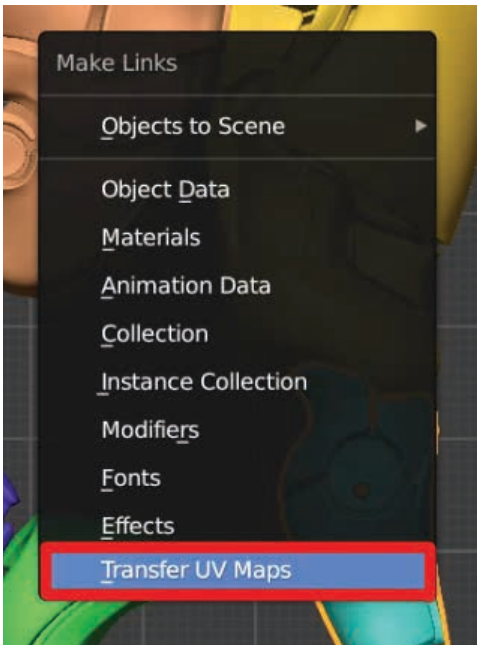
UV Unwrapping Transfer and pack UV's

Just like with modeling, you can save time by mirroring your work.

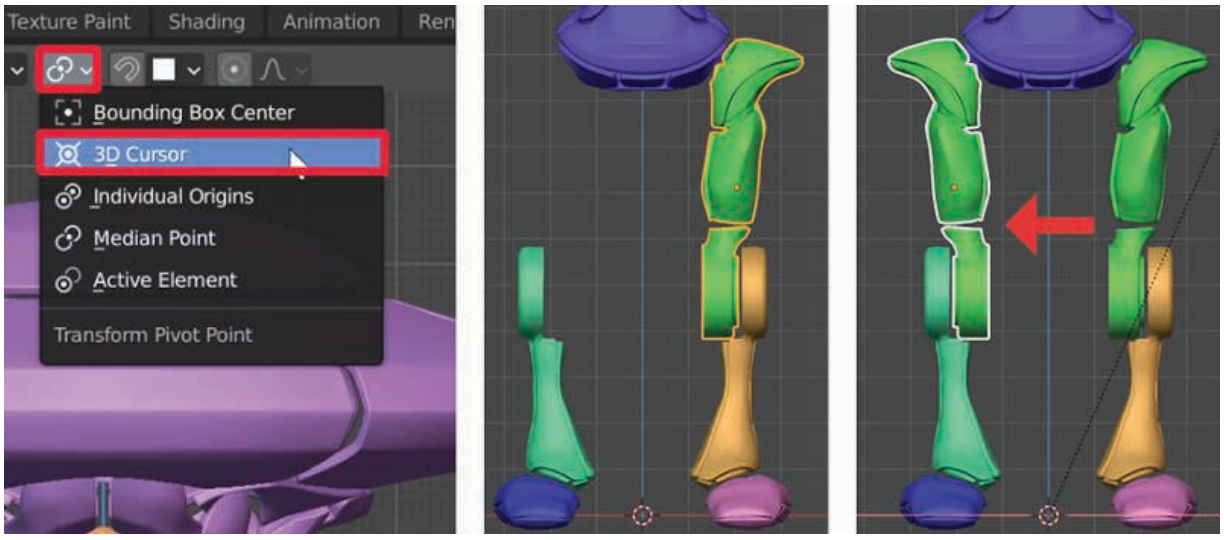
Mirroring a UV map is easy.

Just select the not yet unwrapped object, hold Shift and select the unwrapped object, **press Ctrl+L** and choose **Transfer UV maps**.

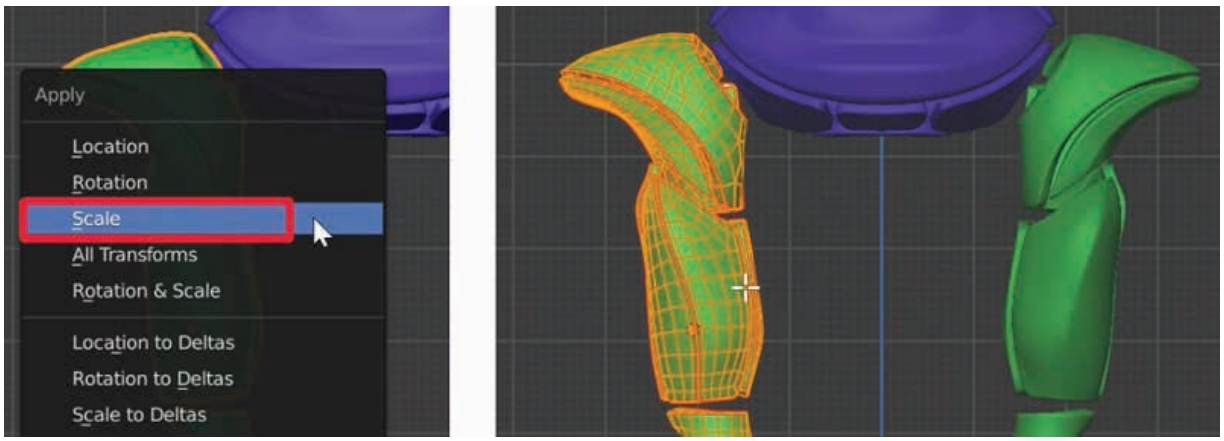
That's all!



Alternatively you can also delete the not yet unwrapped object and duplicate the UV unwrapped one with **Shift+D**, with Transform Pivot Point set to 3D Cursor and the 3D Cursor at the world origin. **Press S+X+[-1]** (-1 on the numpad) to scale the object negatively 1 unit along the X-axis.

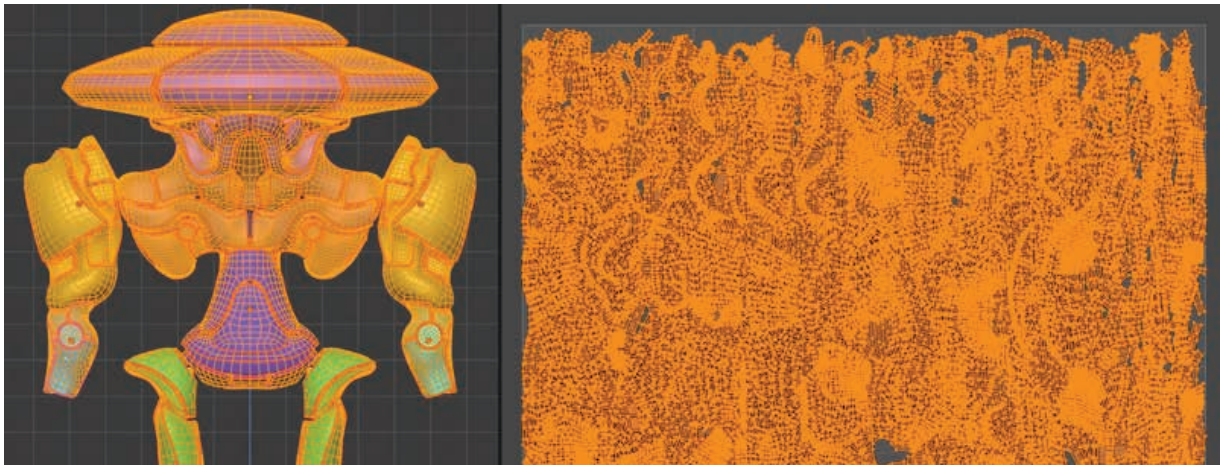


Apply the Scale with **Ctrl+A > Scale** and flip the normals in edit mode by **pressing A** and then **Shift+N** (they're inverted because we inverted the scale).

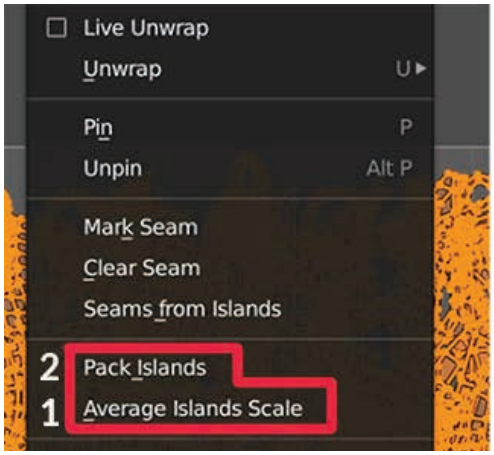


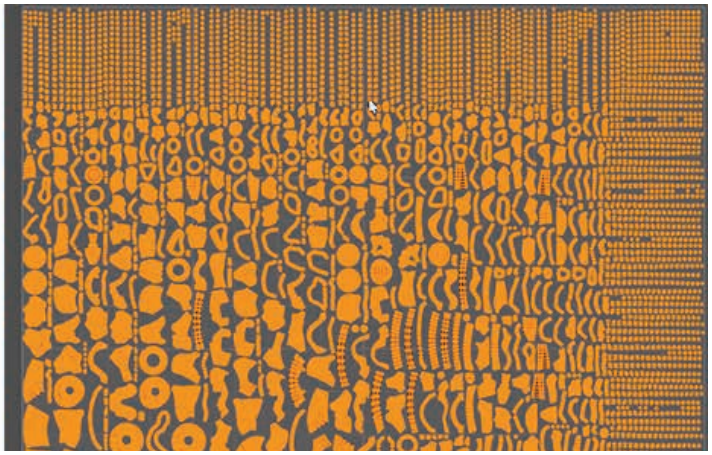
Now all these objects have their own UV's, and we can give them all their own individual texture.

If you don't want that, and just want to give them all one texture, that's also possible. Select all the objects in Object mode, then go to Edit Mode and **press A** to select all. As you can see the UV's are now a big overlapping chaos.

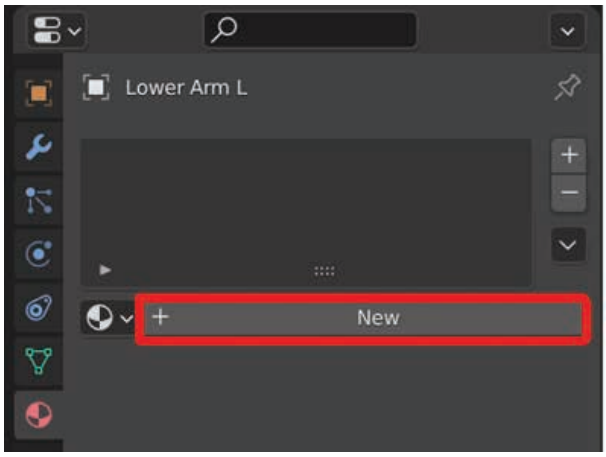


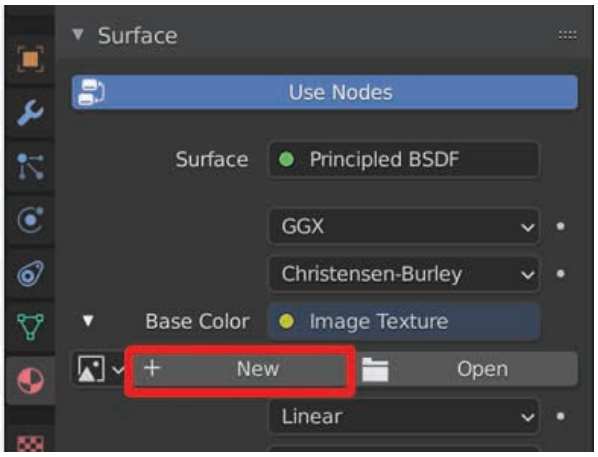
Go to **UV > Average Islands Scale** and then **UV > Pack Islands**. Now all the UV's are packed in one UV tile, and can all share the same texture.



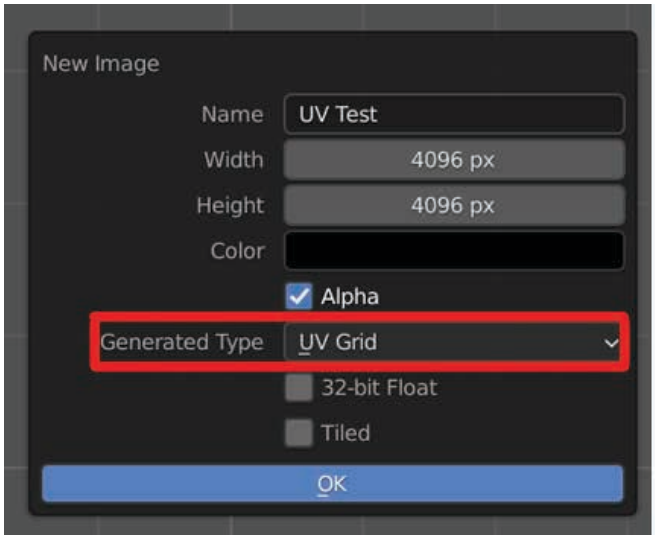


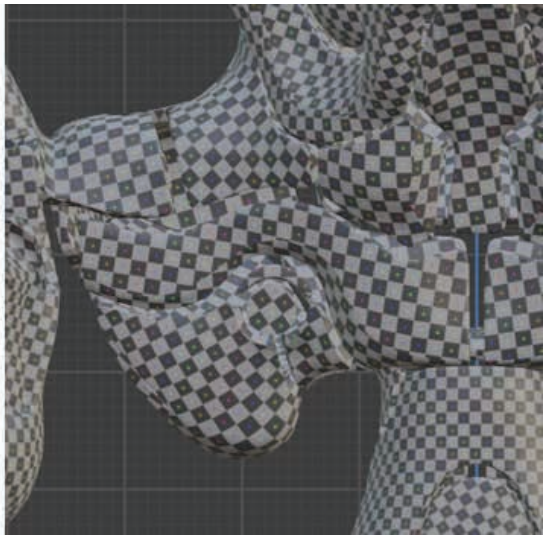
Add a material to the object and a new texture.





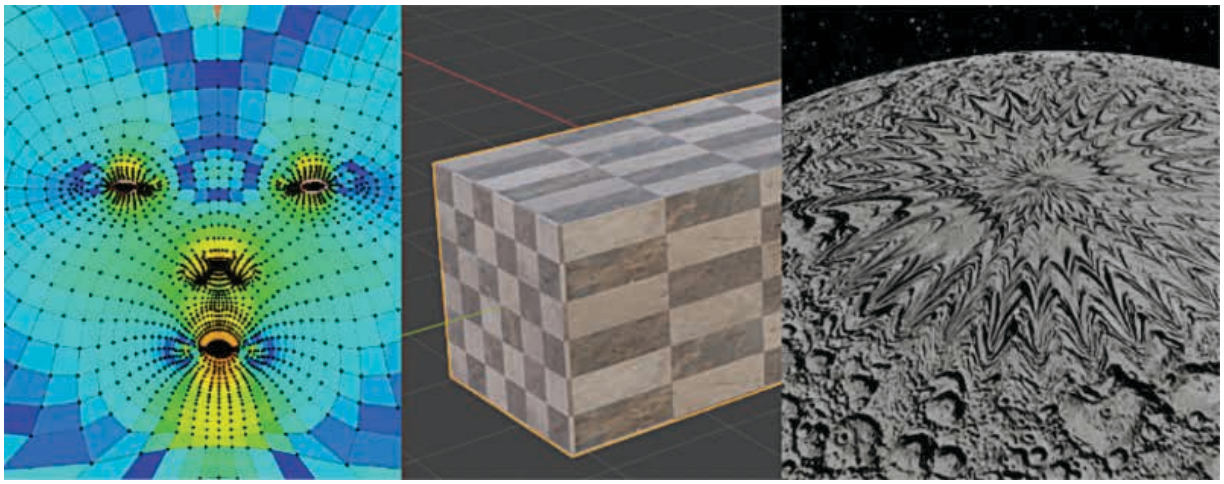
For the new texture use a **UV Grid** as the **Generated Type**. In Material view you can see the texture on the model. The UV grid can be very helpful to see if there's any stretching caused by bad unwrapping, so it will inform you where you may need to add some more Edge Seams.





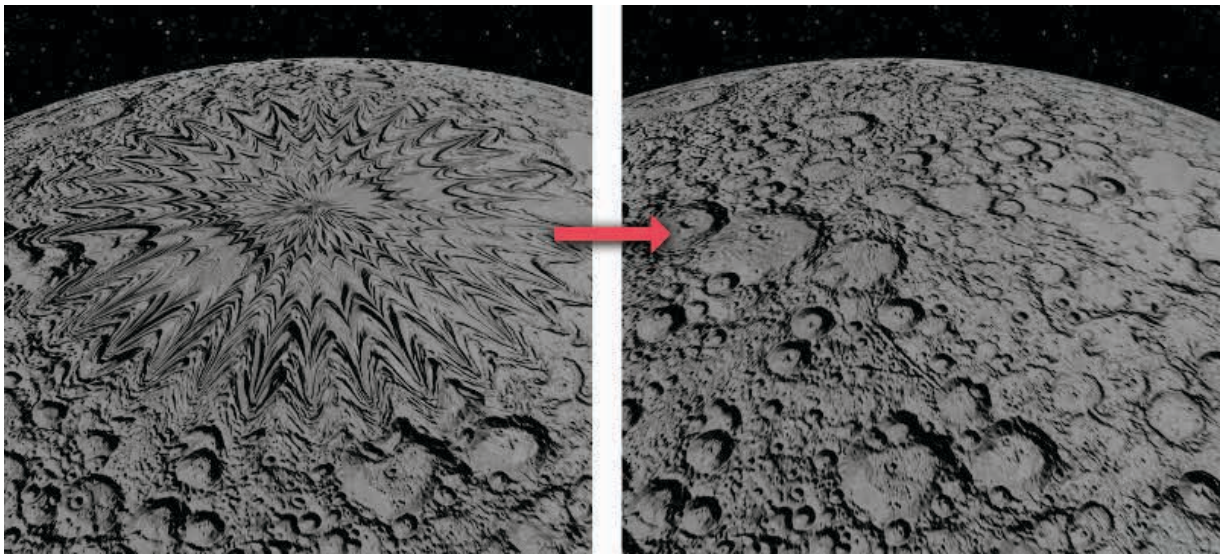
Solving 3 common texture & UV issues

Texturing and UV-unwrapping issues can be very frustrating. Here we look at three real head-scratchers, and how to solve them.



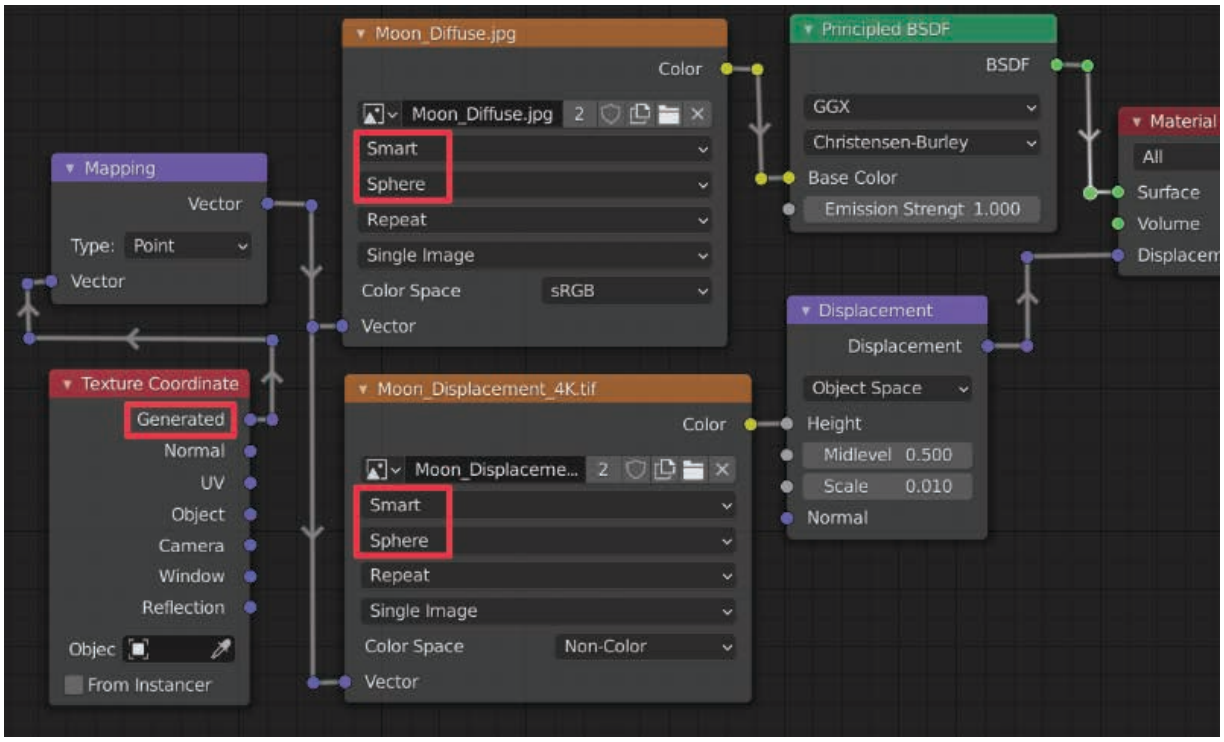
1. Texture pinching on a UV sphere

This has personally been bugging me for months, until I finally found the solution. And it's actually very simple! The issue is that when using an **equirectangular texture** (like the ones NASA gives away for free on their website) on a **UV sphere**, you get this ugly pinching at the poles of the UV Sphere (shown below left). How to get it like the picture below right?



I'm certain that some of you are now grabbing their phones to write me a Twitter DM about how I should use a **Quad Sphere** instead, since they have superior UV layouts. Perhaps a small minority is even going to suggest using an **Icosphere**. Unfortunately to use an equirectangular texture, I really have to use a **UV Sphere.**

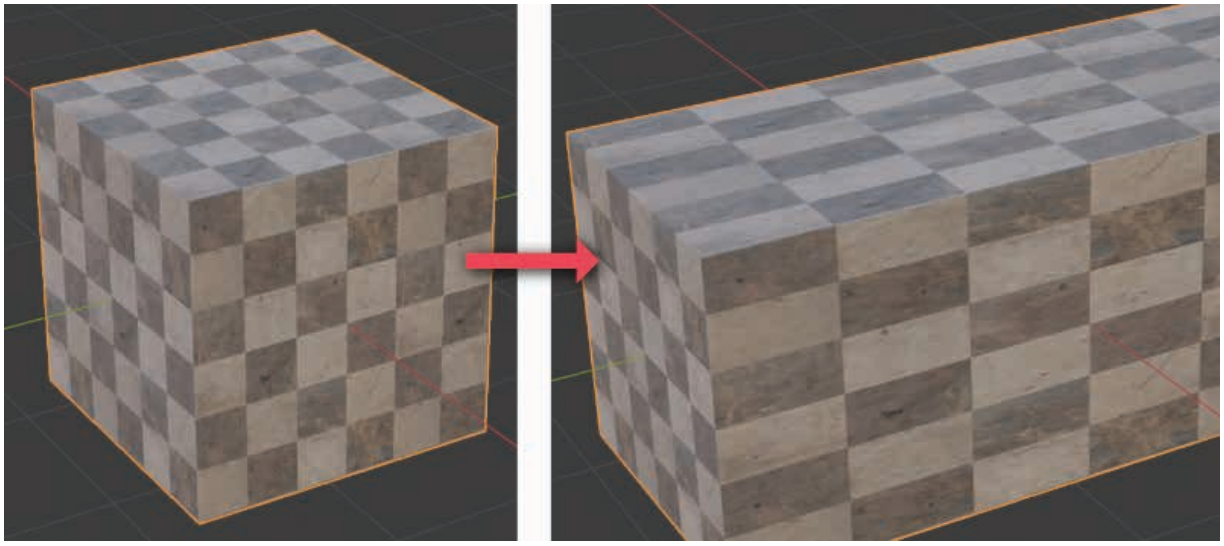
Luckily it's easy! All you need to do to perfectly use a texture on a UV Sphere is to use **Generated** UV coordinates, set the texture interpolation to **Smart** and the projection method to **Sphere**. That's all - now it looks perfect! Remember to do that for all textures *(Roughness, Bump etc)*, not just the Diffuse texture.



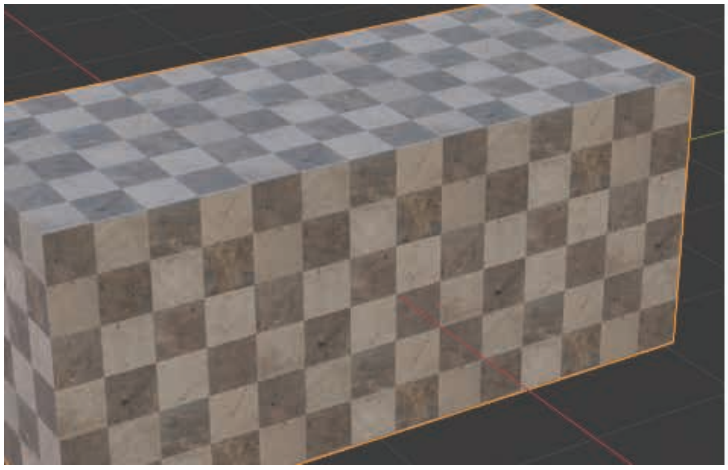
And since I just know you won't be able to sleep tonight unless I tell you how to connect your nodes with those *cool angular connections* like shown above, it's explained in this short video.

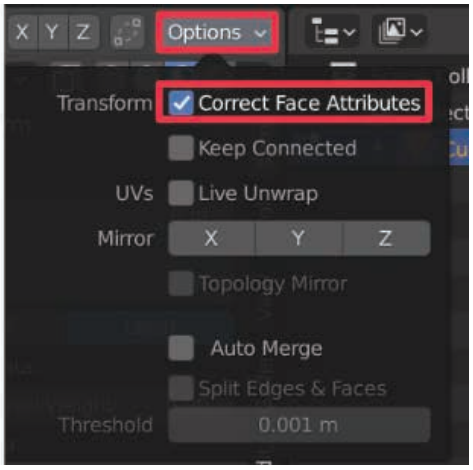
2. Texture gets stretched

Whether you scale your object in Object mode or Edit Mode, textures will appear stretched (or squashed) if you scale an object after UV-unwrapping it. In such a case, you'll have to adjust the UV's in the UV editor window. Or you can take advantage of a relatively new function in Blender.



In Edit Mode, from the Options drop-down menu, check "**Correct Face Attributes**". Now you can edit your model without having to unwrap, as the UV's will automatically update while you scale or extrude parts of your model. It's like magic!

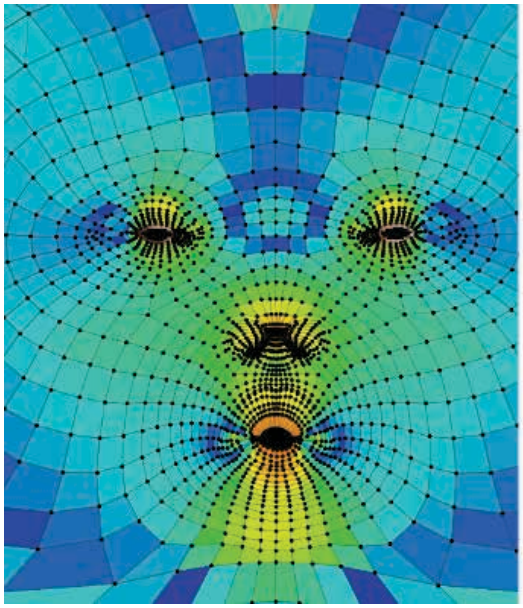


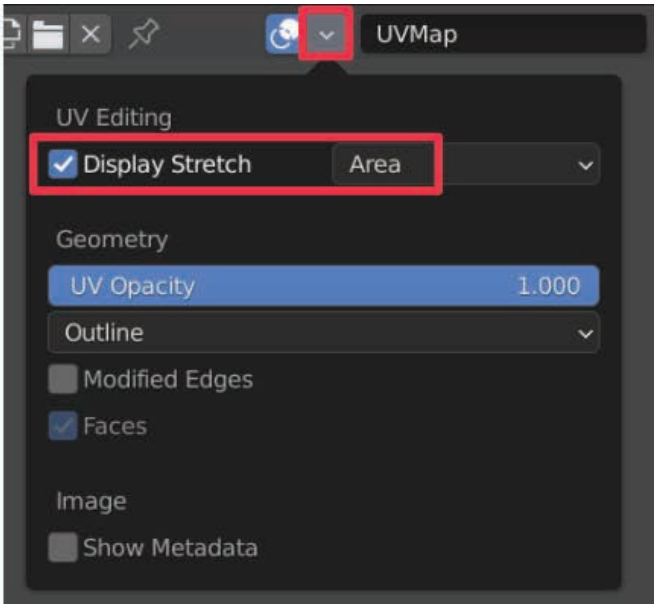


3. Texture resolution too low in certain parts

In some cases, a less ideal UV unwrap may cause parts of your model to have blurry textures, as those parts of the UV map are just too small relative to the rest of the UV map.

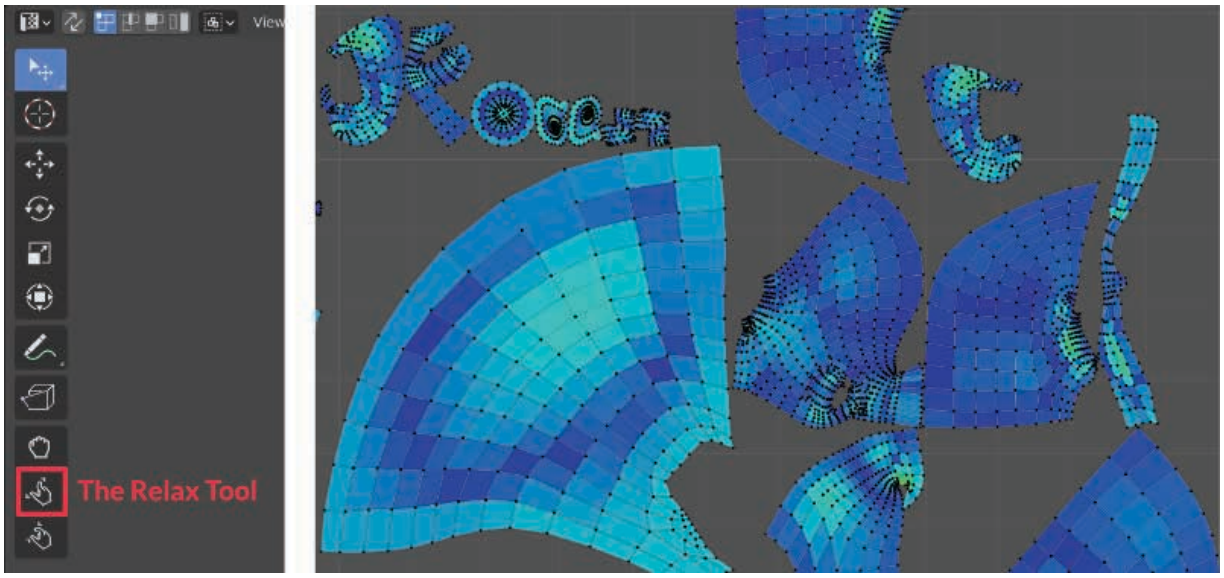
Turn on **Display Stretch (Area)** in the UV editor to see where stretching occurs on your UV map, so you know which areas to give some love.





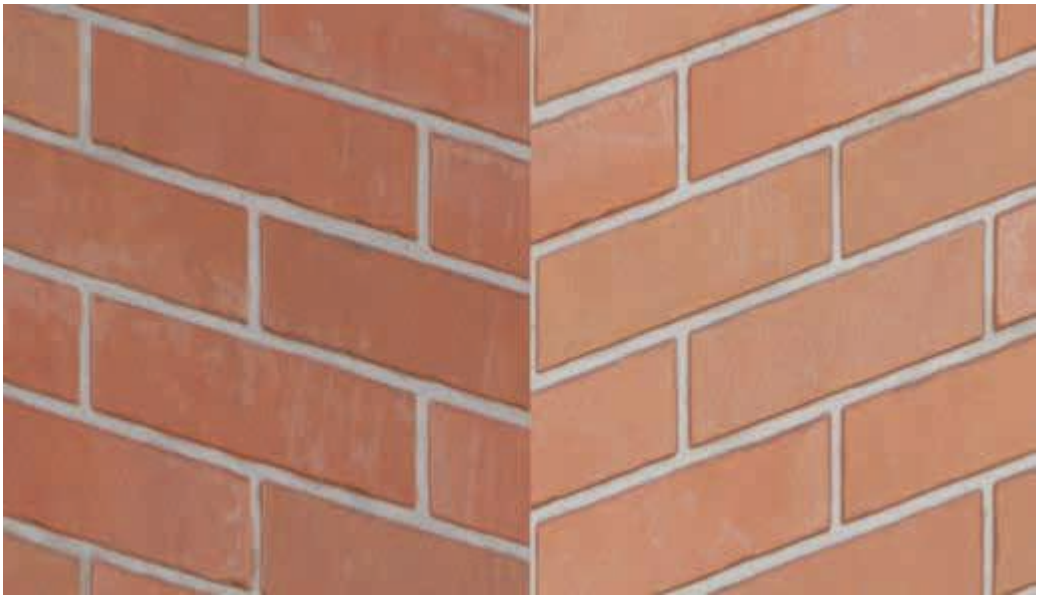
Warm colors (yellow, red) visualize stretched areas, and blue areas are where the UV's are fine. The current location of this checkbox (one that the developers are particularly fond of moving to a different location with each release, to keep tutorial makers on their toes) is a dropdown that is only there in **Edit Mode**. You may need to scroll the header bar of the **UV Editor** to the right to see this drop-down if you're on a small screen.

You can also use **proportional editing** and scale some areas, or use the **Relax Tool** from the Tool panel for a similar effect. In reality there's always a trade-off when trying to make a UV-map "human readable", where some amount of stretching is acceptable, and making it "perfect", but difficult to interpret visually. It's easier to **mark more seams** so the UV map is split up into more islands, which will result in less stretching (pictured below right).

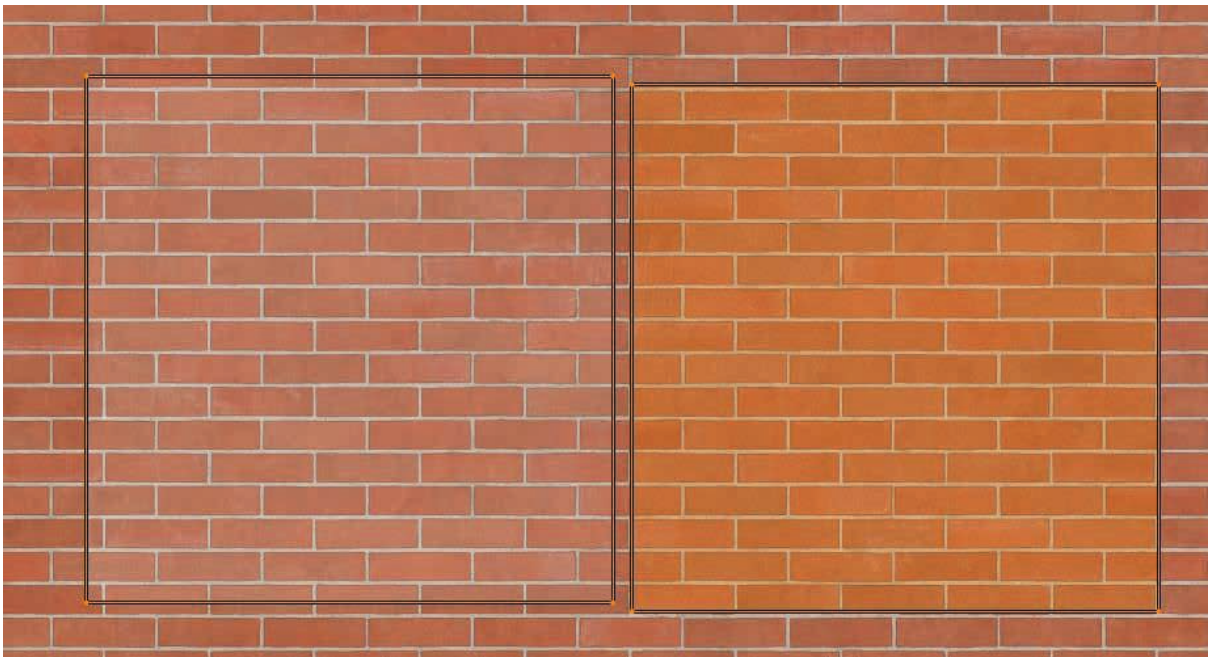


Stitch UV's

Let's say you want to properly align the texture on this wall at the corner where the sides meet.

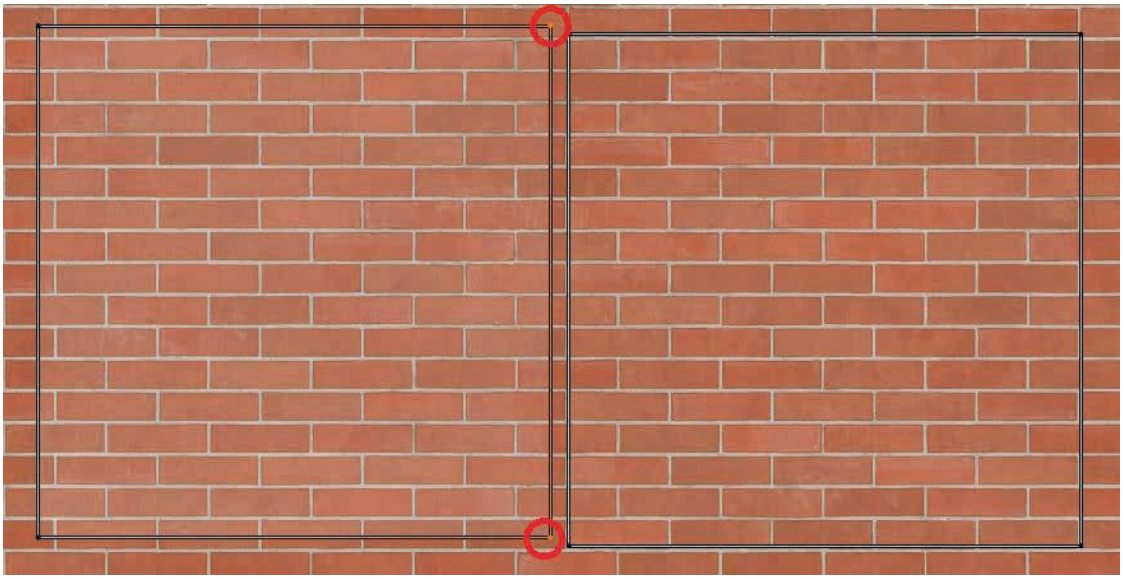


As you can see in the UV editor, the problem is that the UV's are not connected.

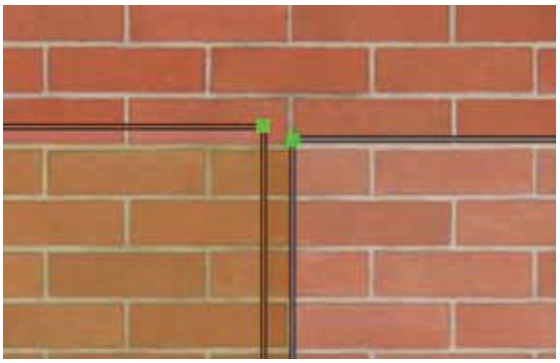


With the faces of the wall selected in Edit Mode, we can see in the UV editor that they don't connect.

Select the vertices of one edge.



Press V. The vertices that are supposed to be connected will become green.

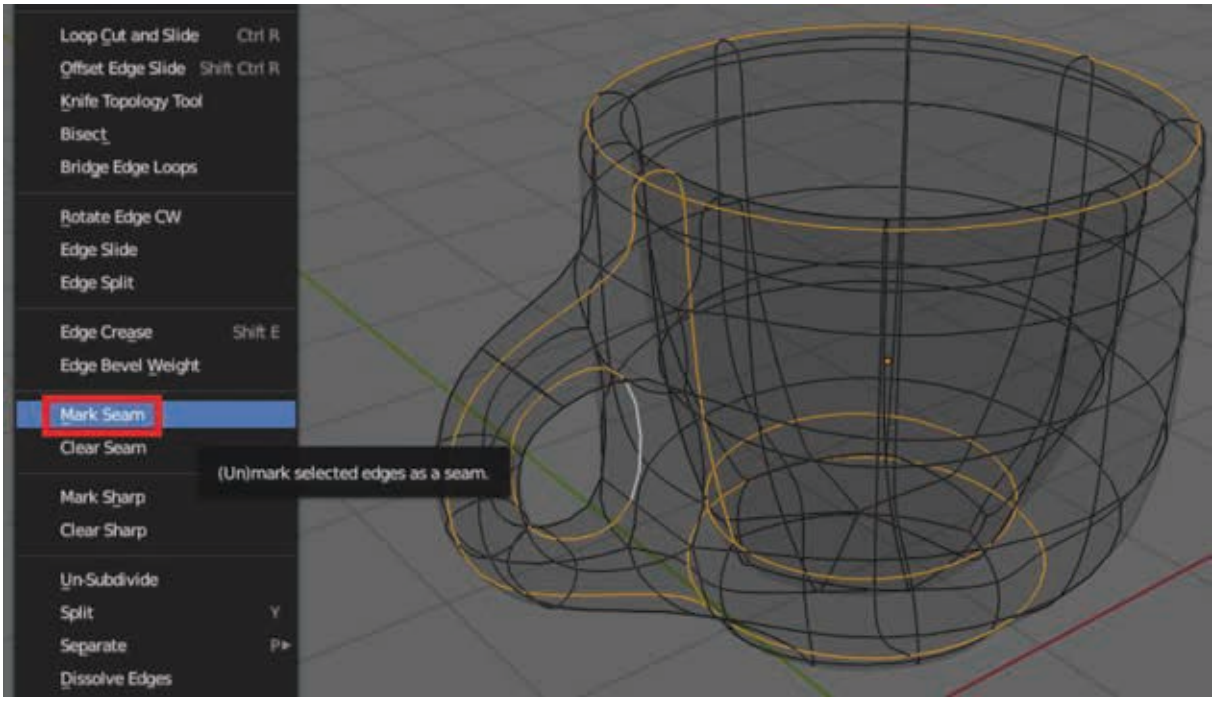


Left Mouse Button-click to confirm (or press Enter). The vertices will snap to each other and the texture will look seamless where the edges meet.

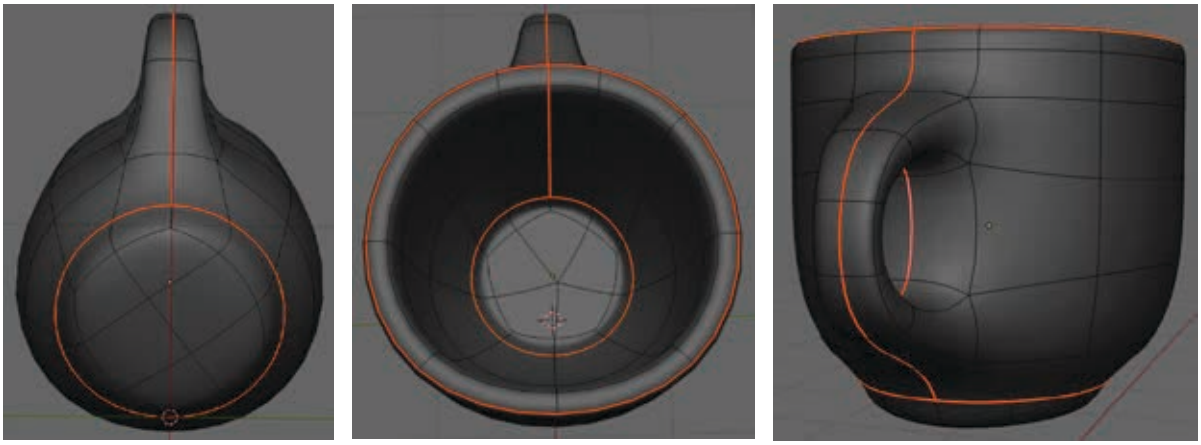
Unwrap a Cup - Follow Active Quads

The following is a demonstration on unwrapping a cylindrical object and straightening out the resulting unwrapped UV-islands. Unwrapping is the process of laying out flat all the parts of a model so that you can place a texture on it.

Select edges to mark as seams and **press Ctrl+E** and choose **"mark seam"** or right-click and choose "mark seam".

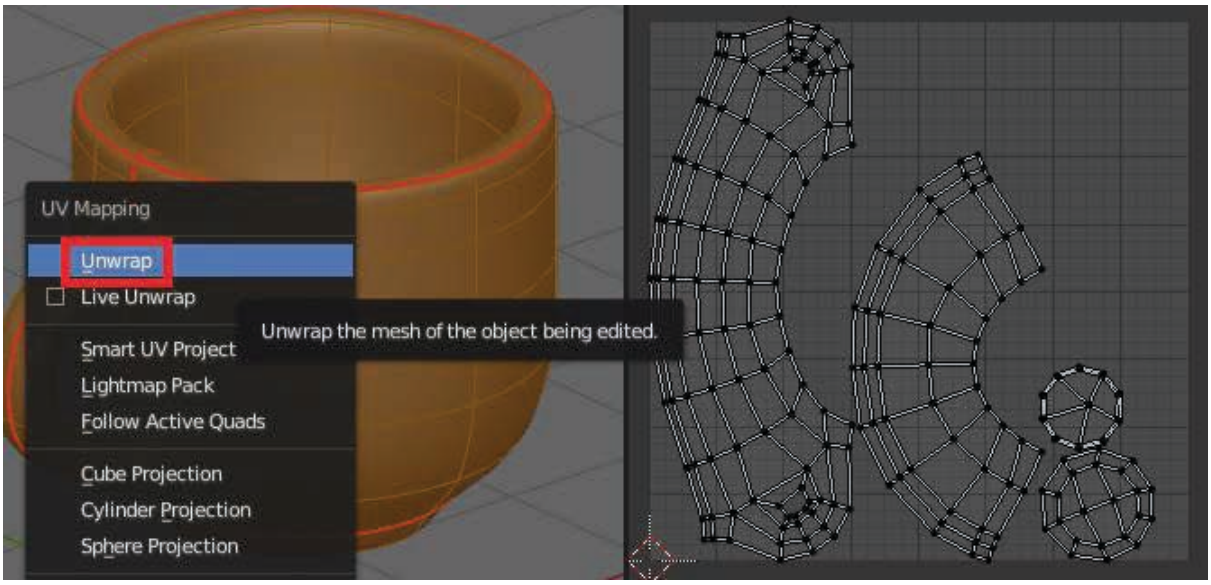


Deciding where to place seams is a matter of experience Try to think about it as if you are cutting an object made of paper in order to flatten it on a table. You don't need to mark all the seams at once, you can mark seams, unwrap the model to see what you get, and continue. You can also select edges, right-click and choose **Clear Seams** if you think you've made a mistake.



Edges marked as Seams are clearly visible as orange lines on this cup.

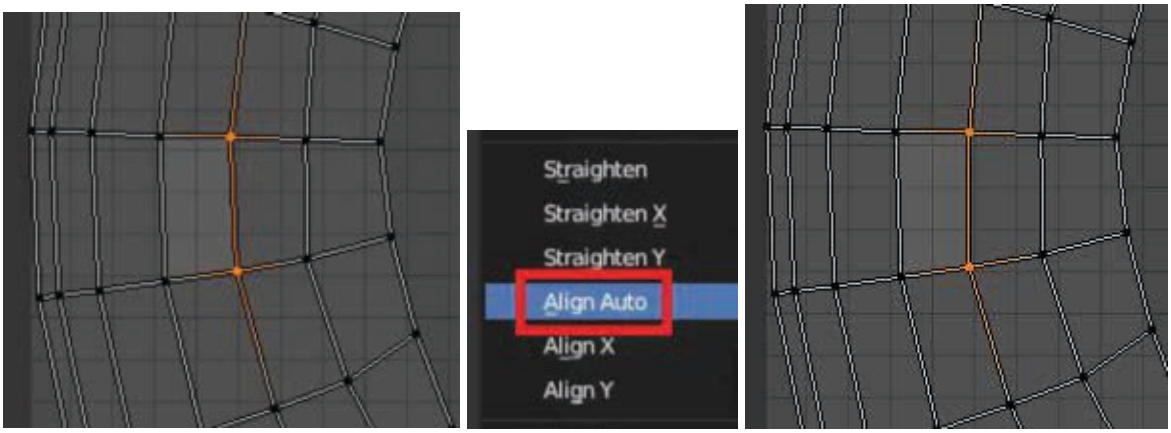
The marked seams are orange. Once you think you've marked enough seams, select all the Faces, **press U** and choose Unwrap.



Left: the UV Mapping menu, right: the unwrapped UV's of the cup.

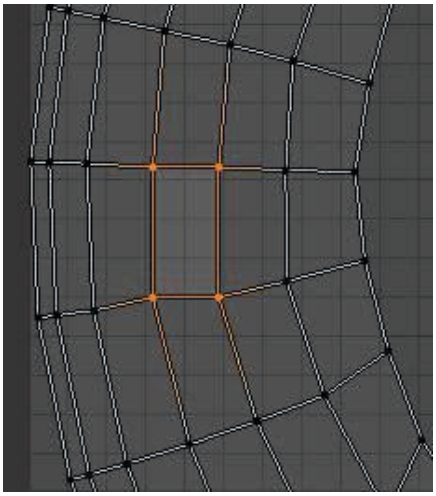
You will notice that the automatic unwrap has done a pretty good job. However, it has distorted some parts, like the cylindrical shapes of the cup which should be a straight rectangle if we want to be able to put a texture on it without distortion. This we can fix manually.

Choose a Face in the part of the cylindrical unwrapped part that already looks fairly straight, then make it even more straight by selecting two Vertices that make one Edge of a Face, **right-click**and choose **Align Auto**.

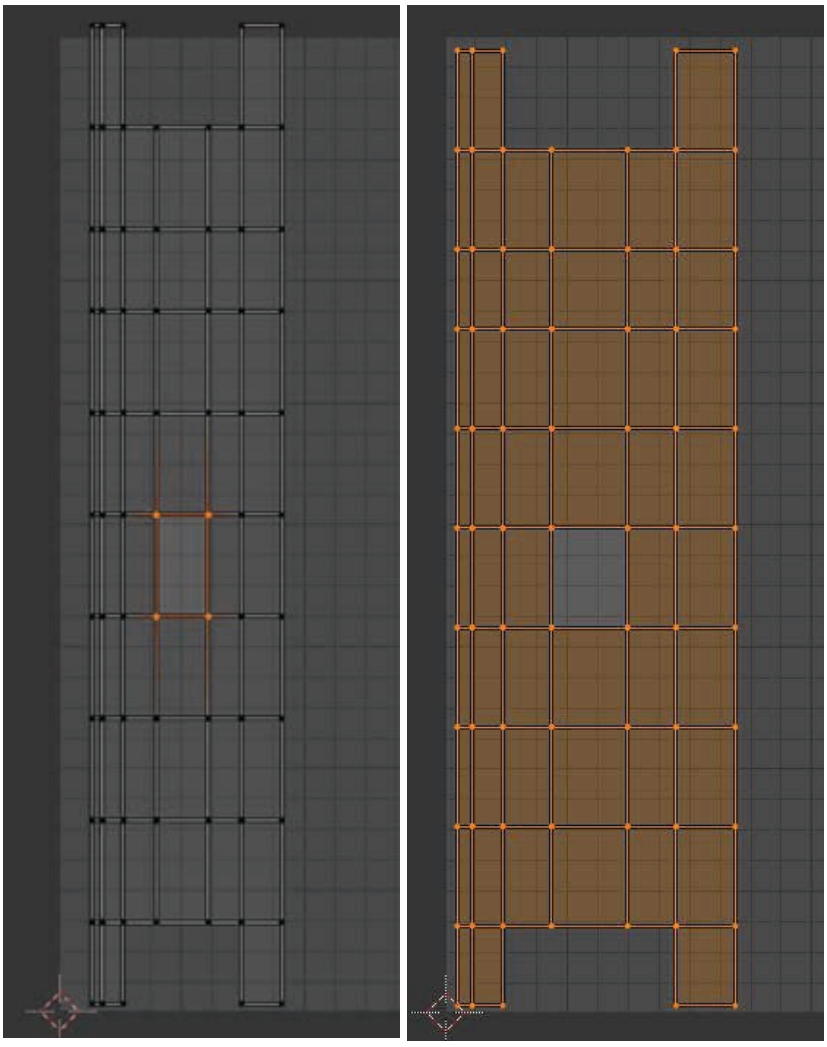


As you can see that Edge has now become perfectly straight.

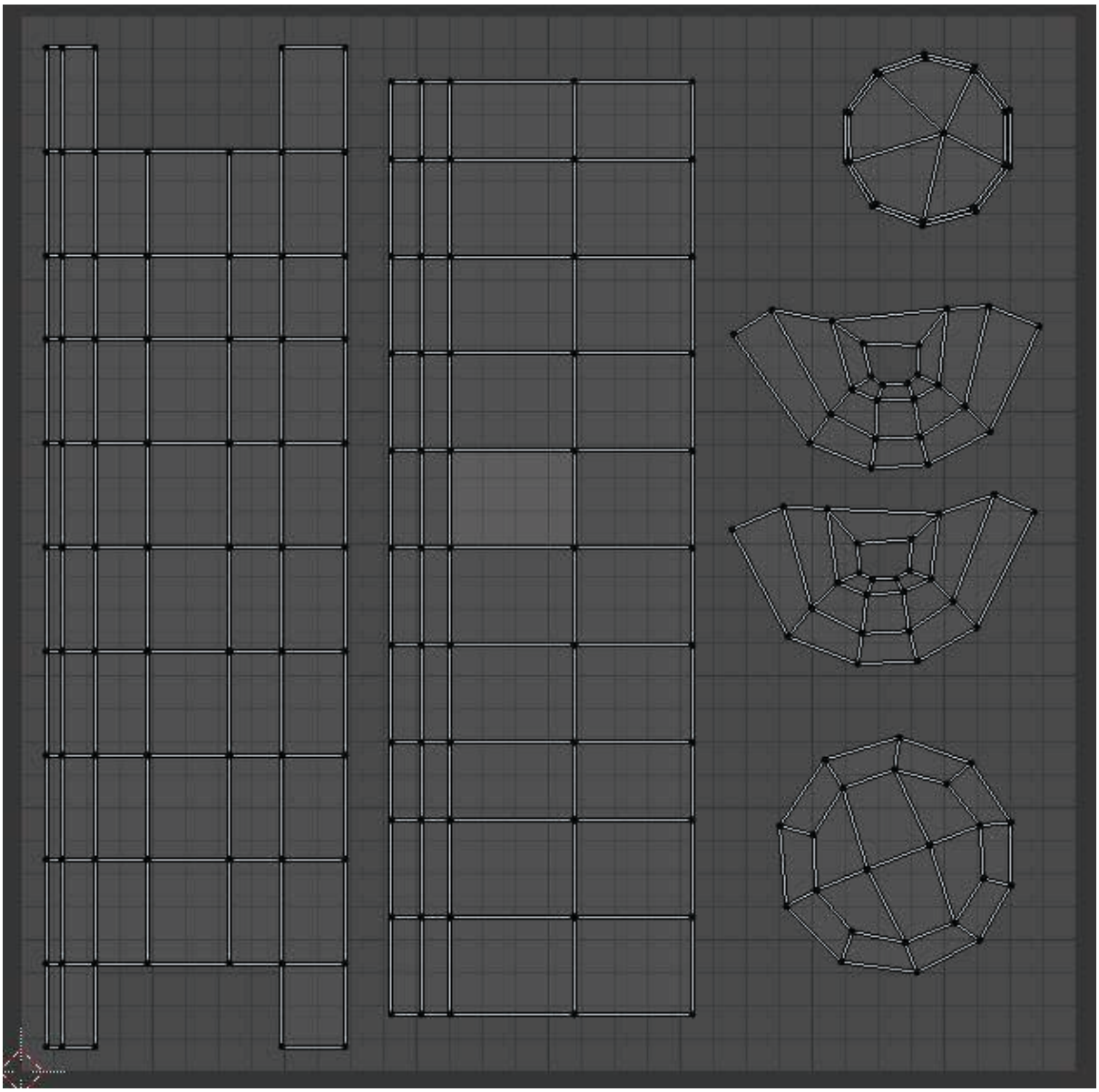
Repeat this step for all the Edges of that Face until the entire Face has 4 straight Edges. With those edges selected, **right-click** and choose "**Follow Active Quads**".



Now you'll find that all the UV's are straightened out (see below left). You may need to rescale them a bit in the Y or X axis (see below right) by pressing **S+Y**or **S+X**. Pressing **Ctrl+L** is a quick way to select all the other vertices of the UV island.



Repeat these steps for all the UV islands that you think would benefit from being straight.

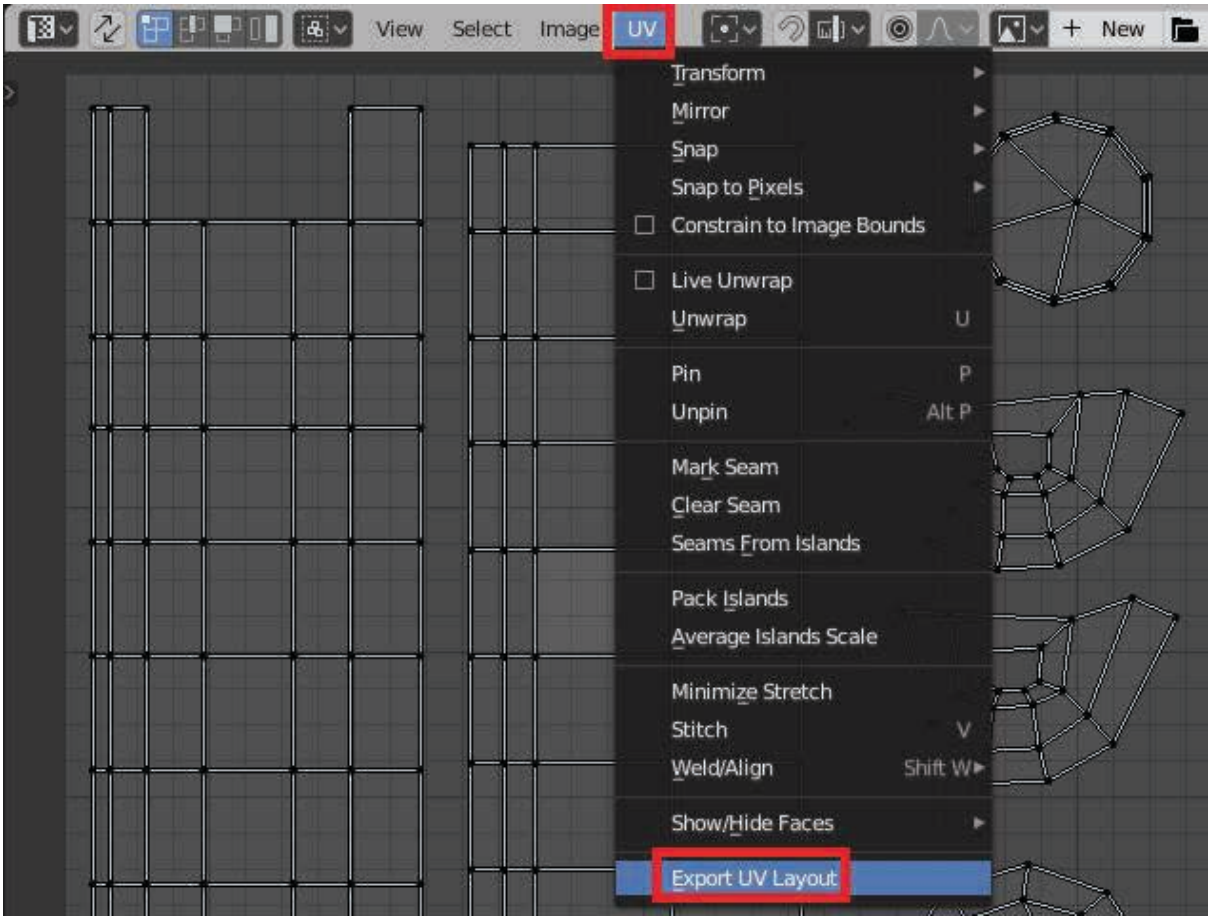


Now any texture you use (or paint) on the cup will not be warped. Notice that I've also moved the other UV islands (the cup handle and bottom) some more space around them by moving them. That way I can paint the texture without having to worry about accidentally painting on the wrong UV island.

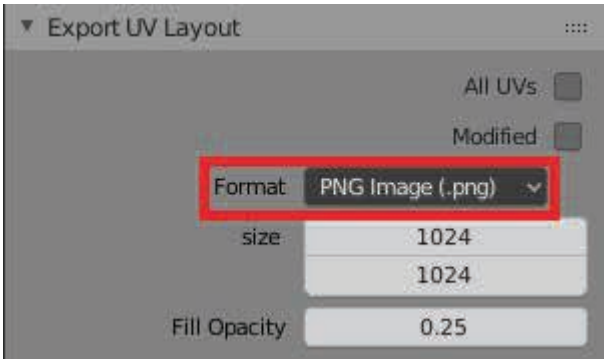
Exporting the UV Layout

Sometimes you want to paint or edit textures in external software like Krita or Gimp. You will need to export the UV layout as a reference for where you are painting.

It's as easy as going to the UV menu and selecting "Export UV Layout".



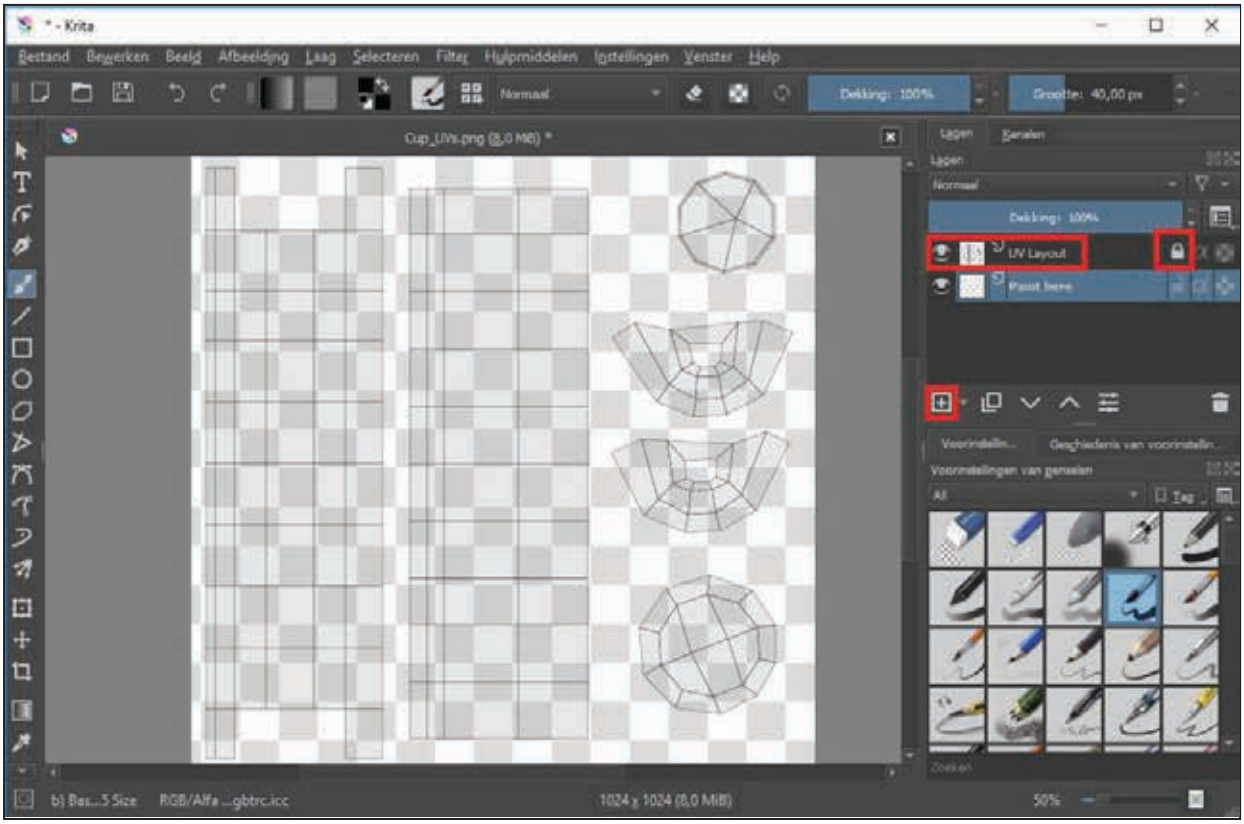
Now you can choose a file destination and file name, as well as file type (I generally choose PNG, which is the default).



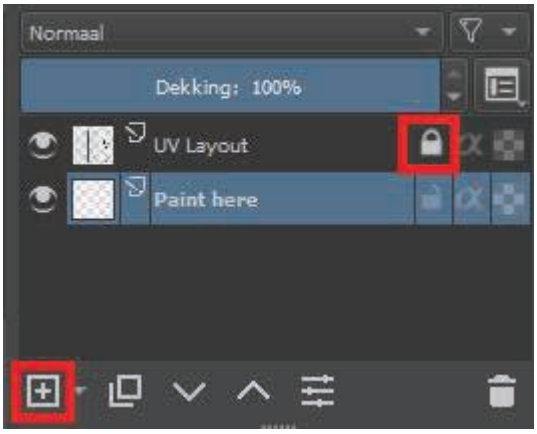
The default settings are perfect.

Now you can open this file in your favorite image editing software. I personally like to use Krita. It's free and really good!

The interface of Krita, with the UV layout on a transparent layer.

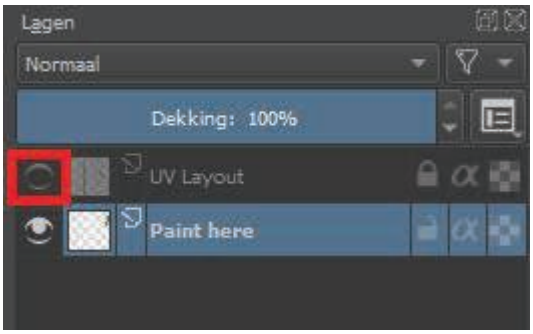


Click on the + icon to create a new layer. Call it "Paint layer" or something easy to identify. Place the UV Layout layer on top, and lock it by clicking on the lock icon so you don't accidentally paint on it. It's only there temporarily for reference so we know where to paint.



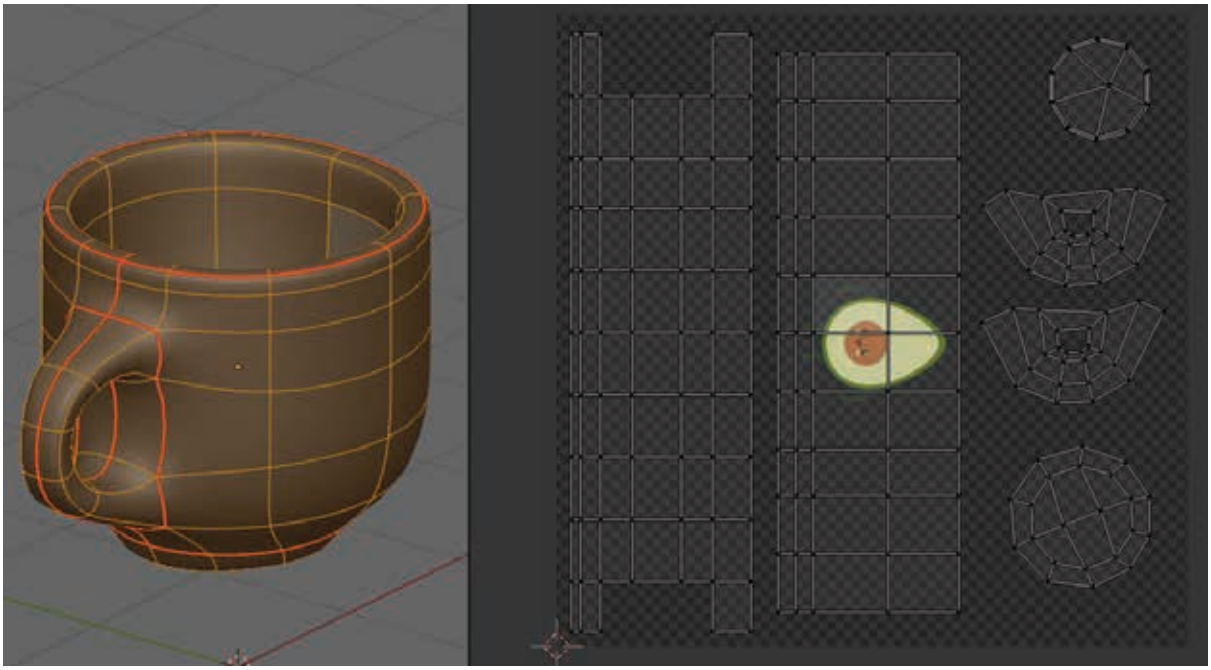
You can change the order of the layers by dragging them up or down.

Paint some texture masterpiece (or place photographic textures) and turn off the UV Layout Layer before saving this texture by clicking on the eye icon next to that layer.



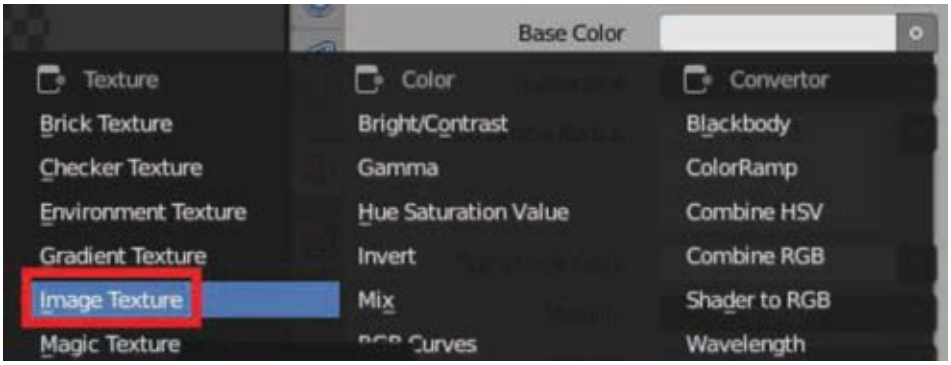
Back in Blender we open the texture file in the UV / Image Editor.

Ok, so far so good, but we don't see the texture on the object yet.

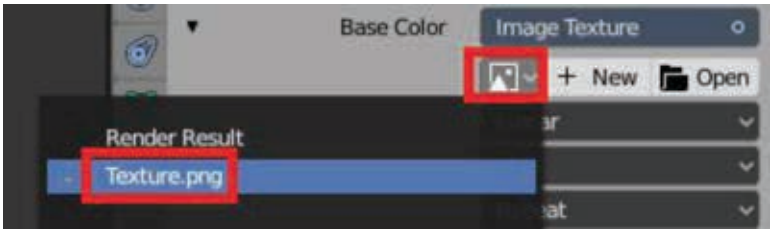


For that we need to make a material for the object. In Object Mode with the cup selected Press "**New**" in the material tab.

Choose "image texture" as color.

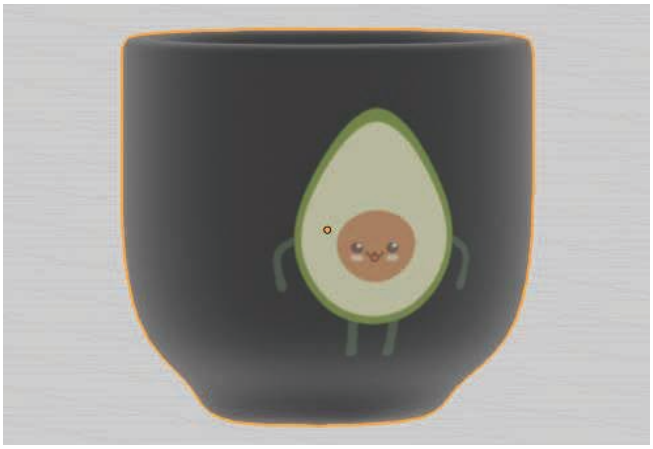


The texture is already loaded in memory so you can choose it here. Otherwise, click "Open".



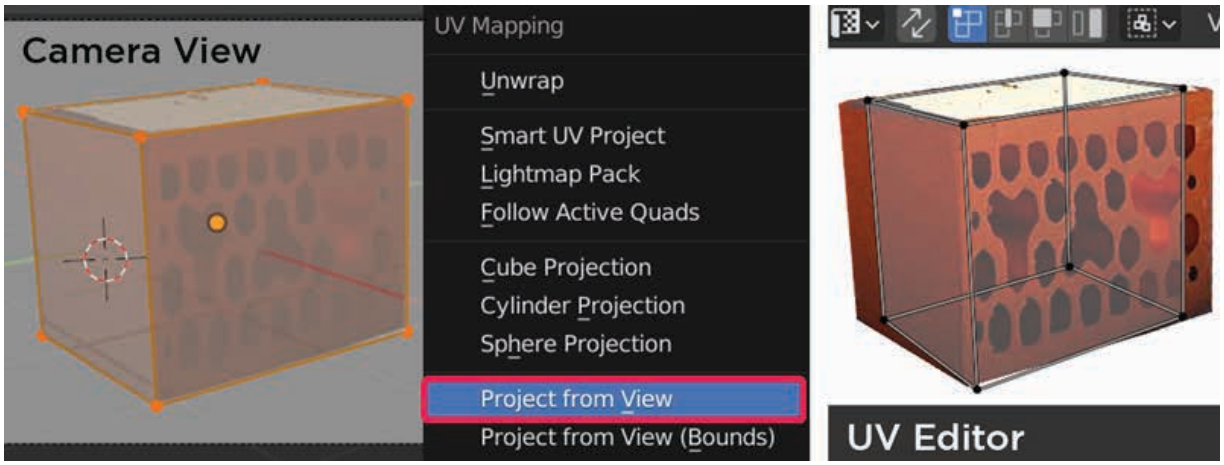
Switch to Rendered Mode (**press Z and move the mouse cursor up**) to see the texture on the model.

Now we see the texture on our model. (Note: I've changed the World color from grey to white)

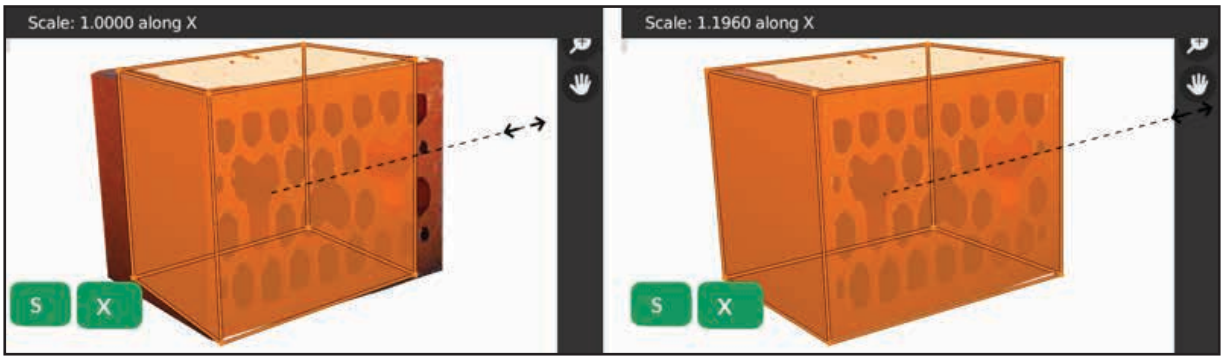


Project from View - avoiding stretched UV's

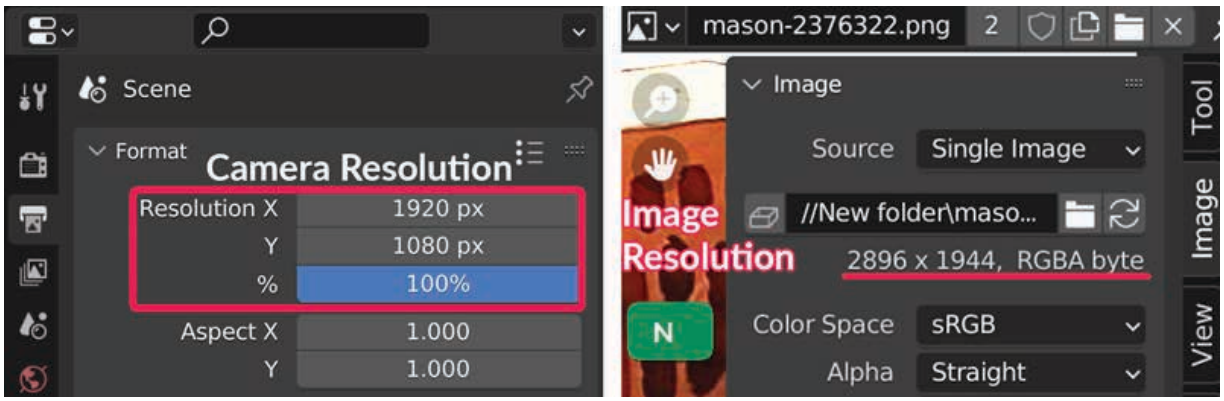
When you're using the Project From View unwrapping method in Blender, it will automatically stretch the UV's to match the texture size. When the image you're using doesn't match the camera aspect ratio, this results in stretched UV's. In this case you have no choice but to resize the UV's on the X and Y axis manually. This video shows a way to avoid this issue. When you're using the Project From View unwrapping method in Blender, it will automatically stretch the UV's to match the texture size.



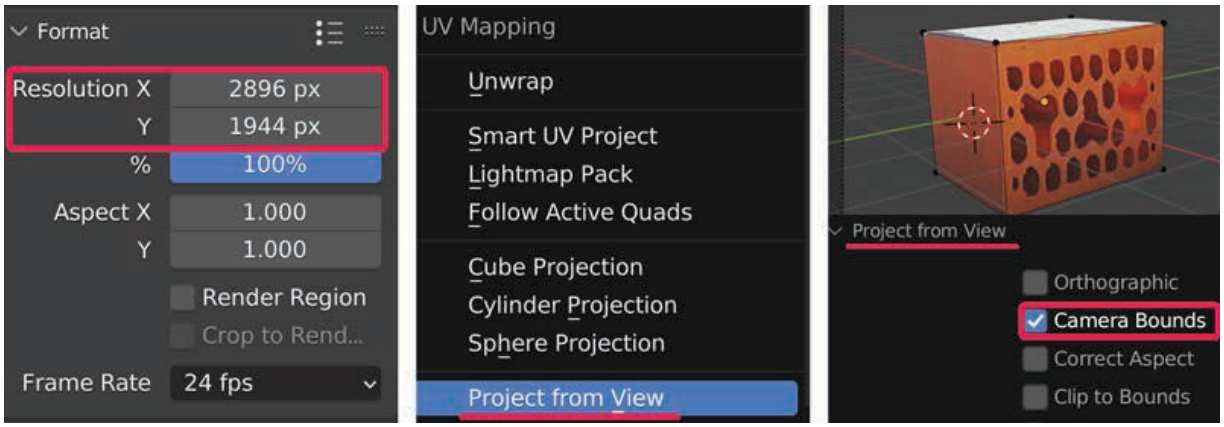
When your image doesn't match the camera aspect ratio, this results in stretched UV's. In this case you have no choice but to resize the UV's on the X and Y axis manually.



One way to avoid this is to **press N** to open the **Option panel** in the UV Editor and look at the Image tab.



Copy the resolution to your Scene Output Properties. Now the scene resolution matches the texture resolution. If you unwrap the object now by **pressing U** in Edit Mode with everything selected and choose "**Project from View**" with the default option of "**Camera Bounds**", it will project it exactly as in the camera view.

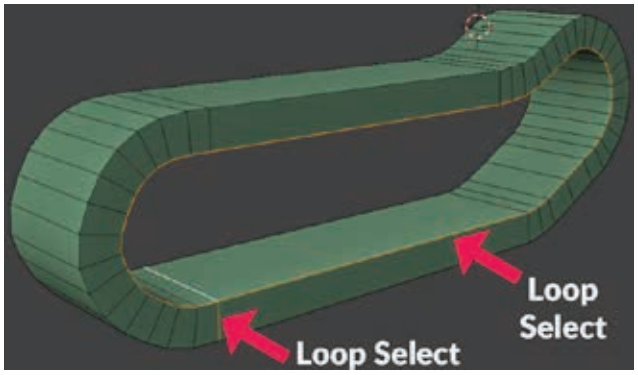


Easy Straight Unwrapping

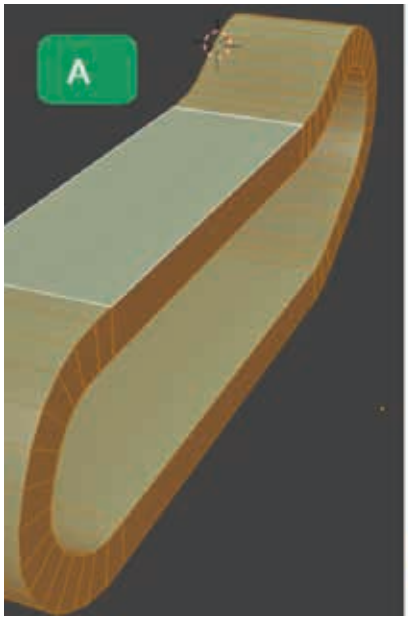
When unwrapping an object like these tank tracks, you usually get distorted UV's instead of perfectly straight ones. This then causes texturing issues.

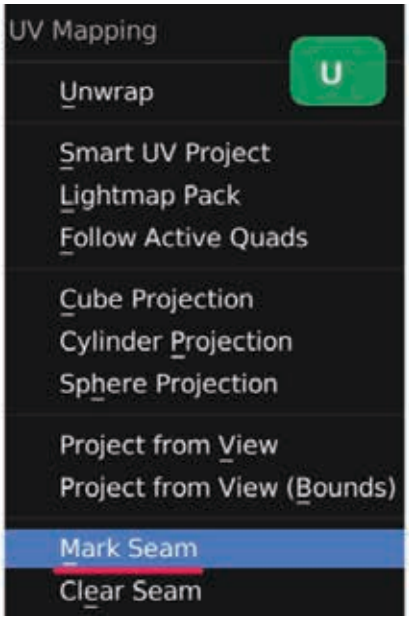
Fortunately there's a simple solution that will give you perfectly straight unwrapped UVs every time!

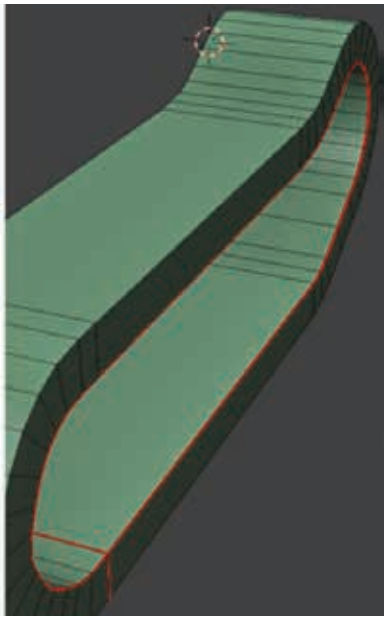
To unwrap something like this tank track, first mark edge loops.



Alt-left click to select entire loops, then **right-click** and **Mark Seams**.

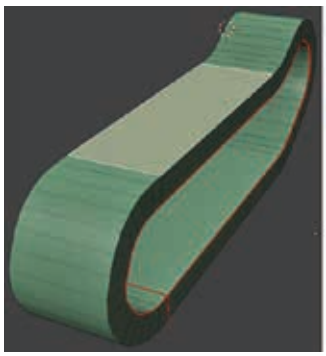


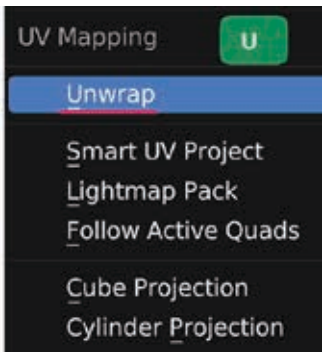




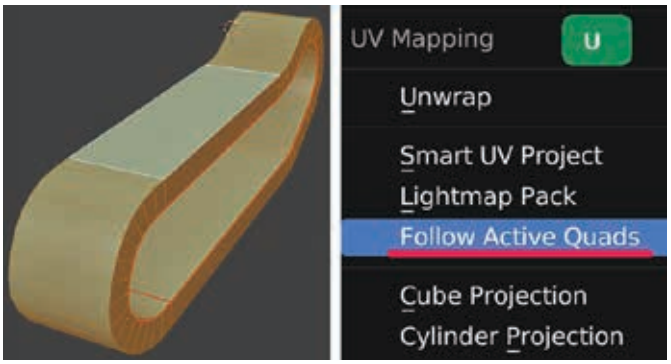
Now if you just unwrap this, you'll see that the unwrap is not perfectly straight.

This will cause problems with texturing, luckily there is a better method. Select one straight face and **press U**, choose **Unwrap**.

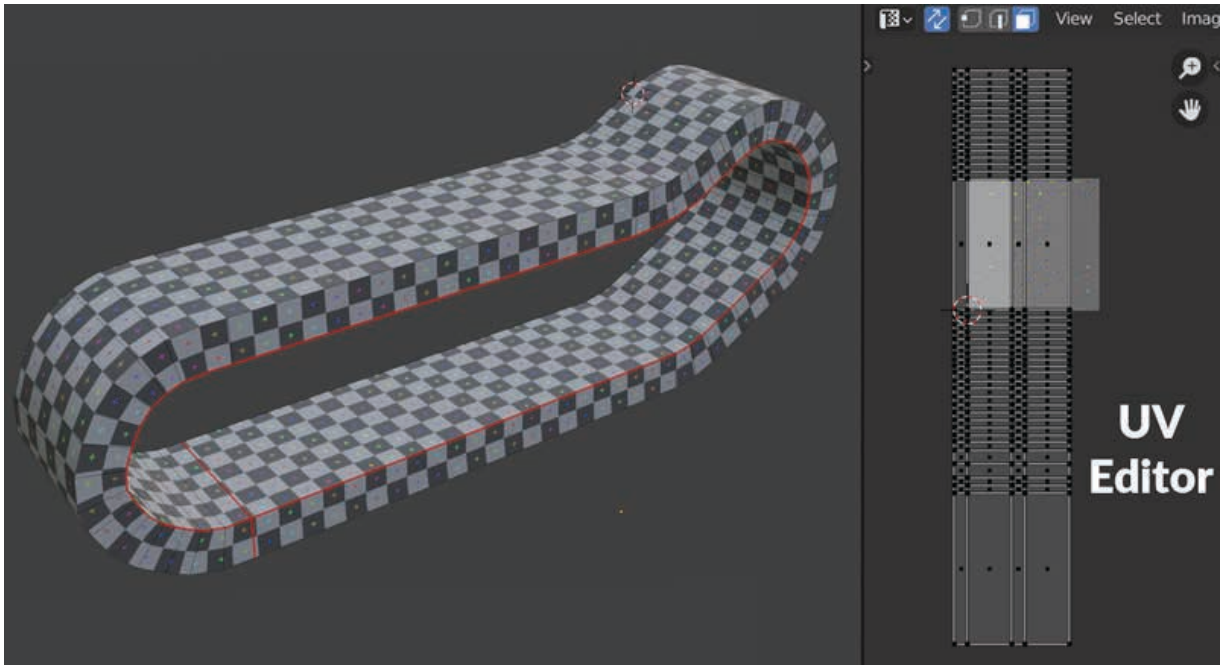




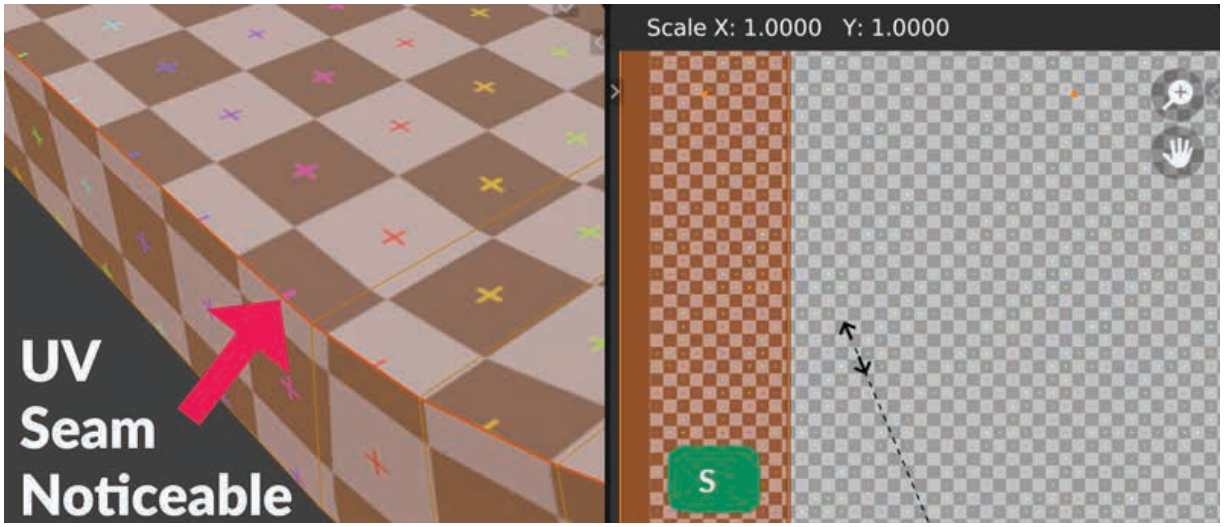
Then select all faces and **press U** again and choose **Follow Active Quads**.



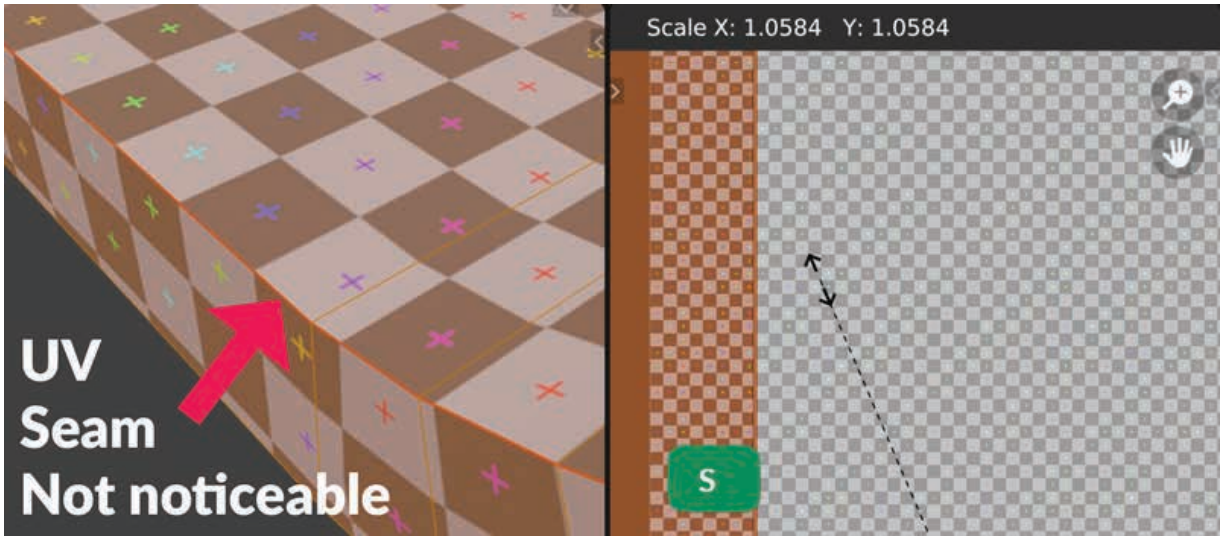
This way, the unwrap is perfectly straight.



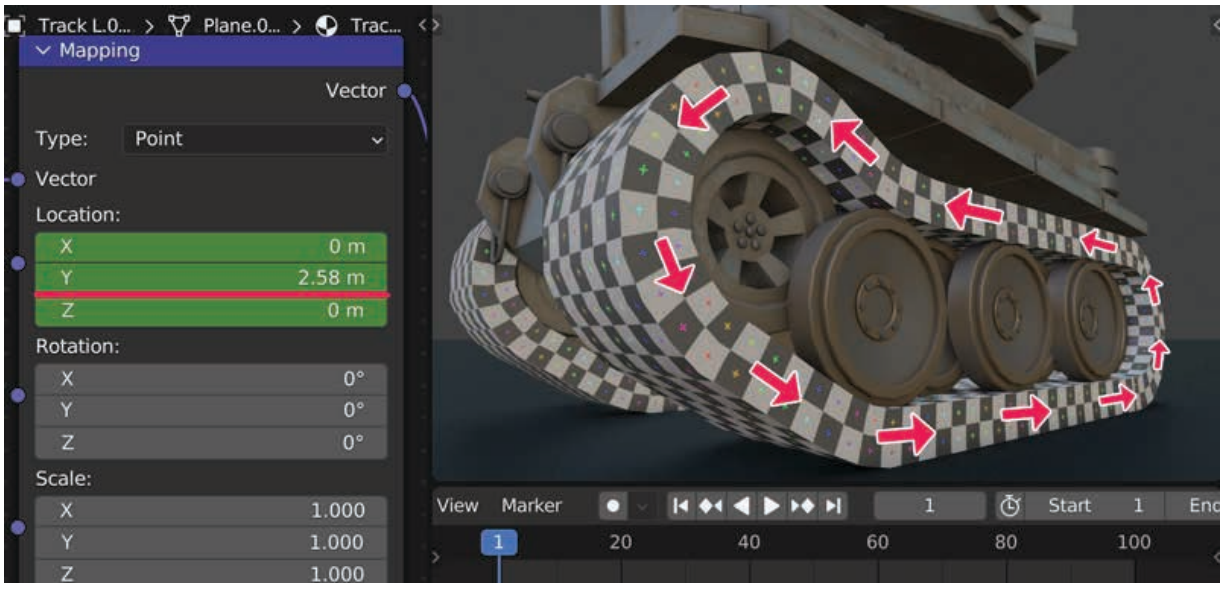
Scale the UV's by **pressing S**...



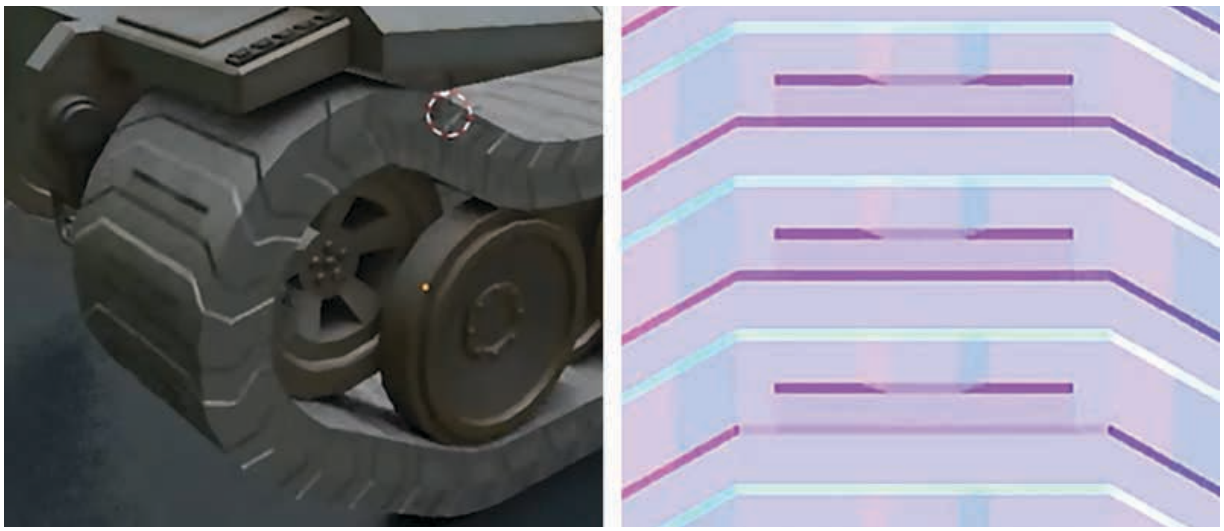
...until the seam is no longer visible.



You can move the texture by changing the **Location** value of the **Mapping node**. To animate it, you can set keyframes or drivers.



This way you could animate a texture including a **Normal map** or Displacement map.

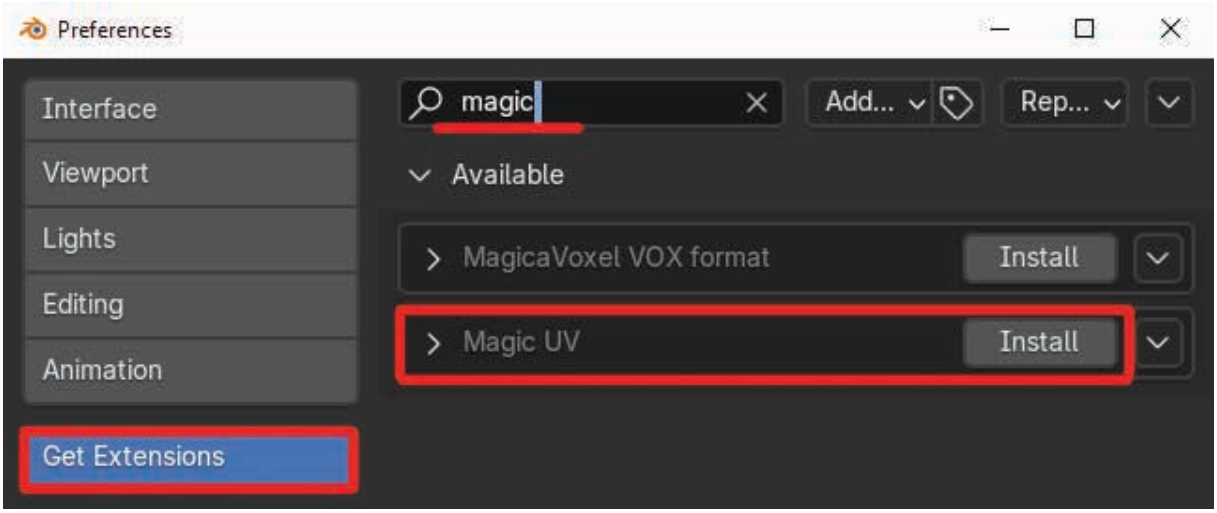


Mirror UV-maps

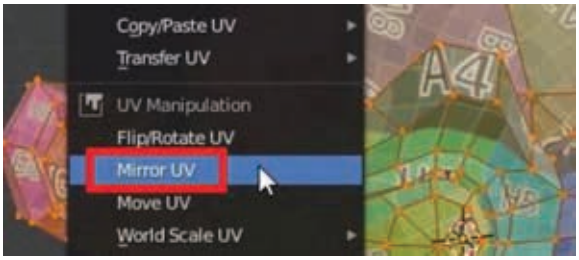
Sometimes it's useful to mirror your UV map, for example when a model is symmetrical. That way you only need to create half the texture. Just like how you only model one half of the model. Below left: without mirrored UV's. Next to it on the right: with mirrored UV's.

For this we need to use the **Magic UV add-on**, which is included with Blender by default.

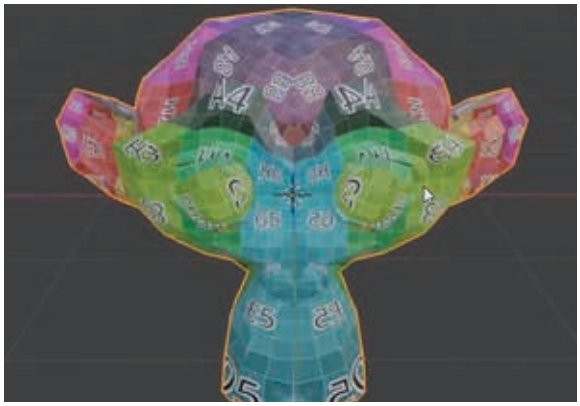
You only need to make sure it installed in Preferences, in the **Get Extensions** tab.



In Edit Mode select the faces of the model and **press U**, then select **Mirror UV**.

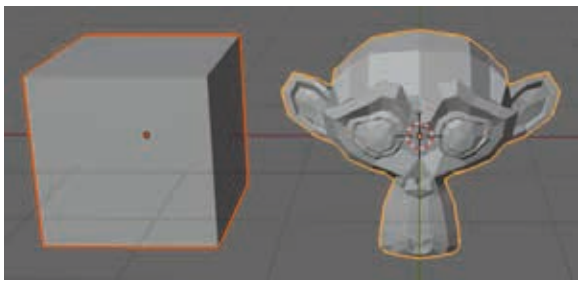


That's all! Magic UV is a very useful add-on for UV unwrapping.

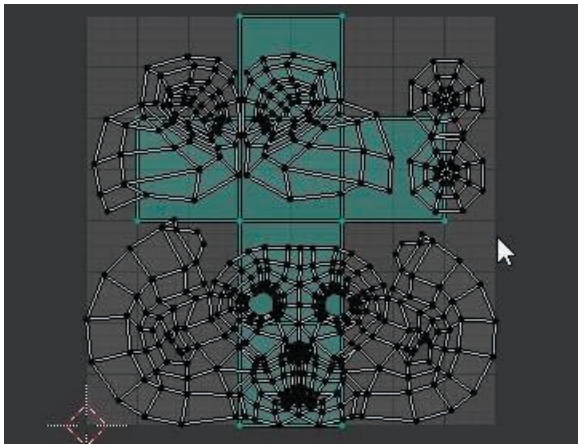


Combine UVs from Multiple Objects

To use only one texture for both of these objects you can combine their UV's.

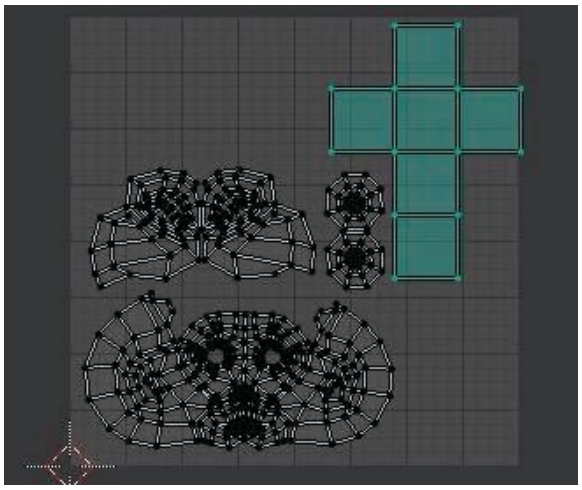


In Blender 2.8 it's easy. Just select both objects and in Edit Mode you can see both UV maps!



Select one vertex, **press Ctrl+L** to select the others, scale with **S**, move with **G** until they are not overlapping.

Now you can use one texture for both objects.



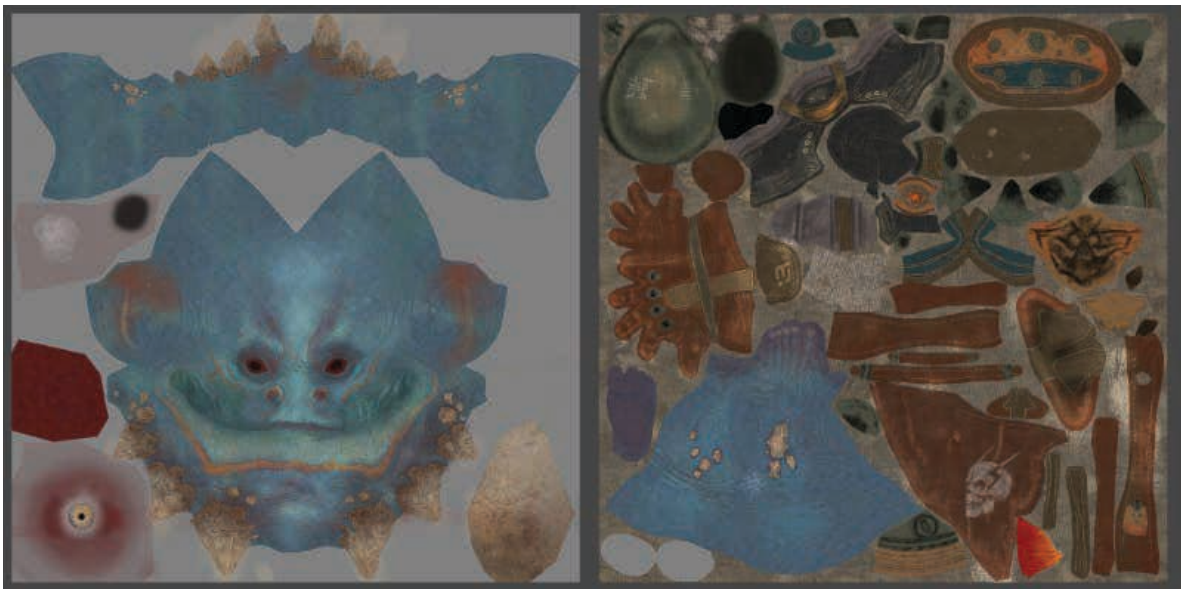
UDIM - Monster-sized textures

You may have heard of **UDIM textures** - but what are those exactly? And why would you use them? In short, it means using many regular-sized textures for an object instead of one ridiculously large one. Before we look at how it works in Blender, let's have a look at why that would be useful.

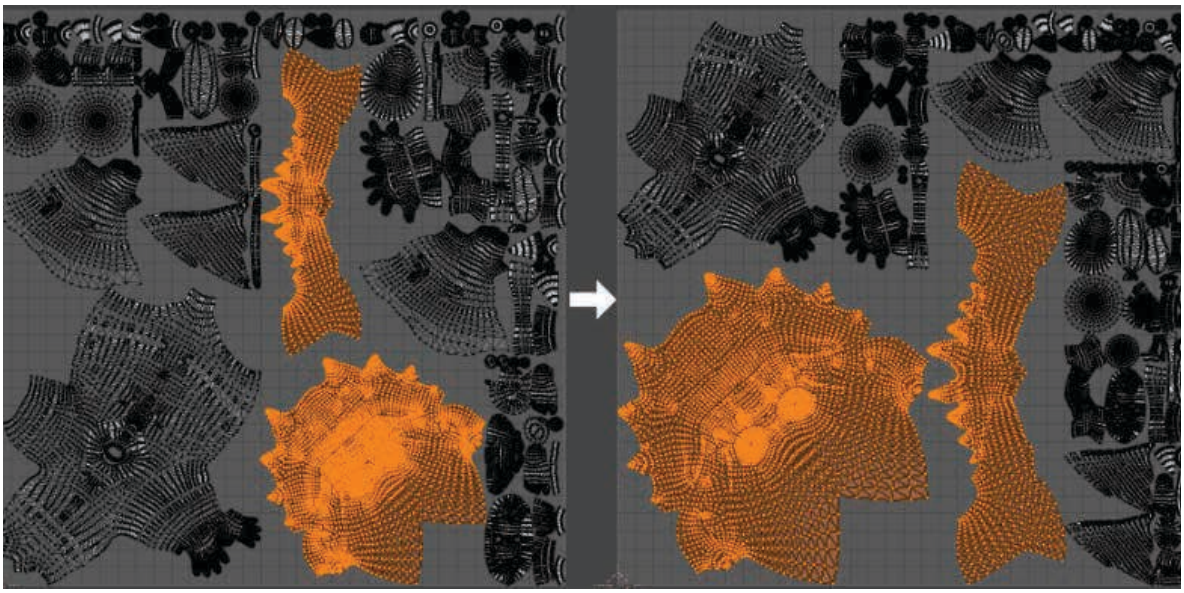
When you're unwrapping your model, parts that take up a larger space in the UV map get more pixels/resolution, as they use more of the image texture space. As a typical example, the face of a character should have more resolution than the rest of the body because people naturally focus more on faces, so this means that in order to fit everything on one texture, the UV's of the rest of the model need to be smaller to accommodate the UV's of the character's face.

For a detailed character, it would be smart to give each of those (head and body) at least their own texture - preferably dividing it up even more, as we'll discuss later.

Below as an example you see how that would work with just two textures for the above character, one for the head (left) and one for the body (right).



Here's how we would do it *before UDIMs*, using just a single texture. Below left for example, the UV's that make up our character's face are selected and then **scaled up (with S)**. Then they are temporarily **Pinned (UV > Pin)** and the other UV islands are **re-distributed (UV > Pack Islands)**. On the right you see the result: the UV's of the character's head now have more resolution.



This has a big disadvantage: the other parts like the UV's of the body, are now smaller. This means that to keep the quality of those other parts reasonable, you then have to compensate by making the overall texture bigger. This is fine until your textures become really large, at which point it's no longer efficient for the render engine and maybe it's not even possible to load the texture.

Above: the computer is struggling...

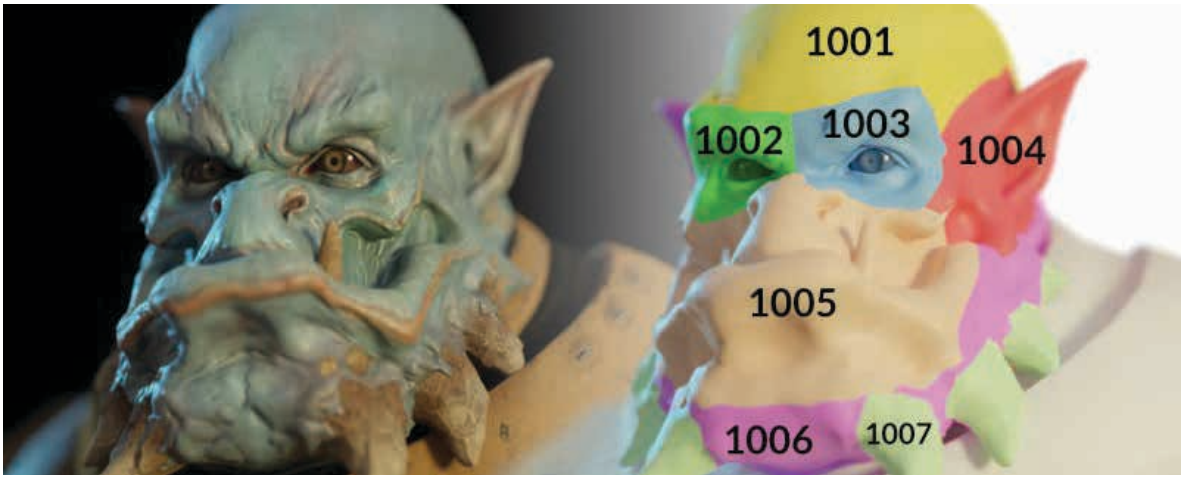
It's also challenging to scale up UV islands after the texture has already been created. This becomes a real problem when you're working for a client or director who decides - after everything is already textured - that they want to have an extreme close-up of the character's eyes. Unfortunately the texture resolution you used isn't high enough to accommodate such a close-up look. And to scale the entire texture up at that point leads to such a high resolution, that the computer just can't handle it. This is where UDIMs will save you.

Above: close-up, the resolution of the texture is not enough.

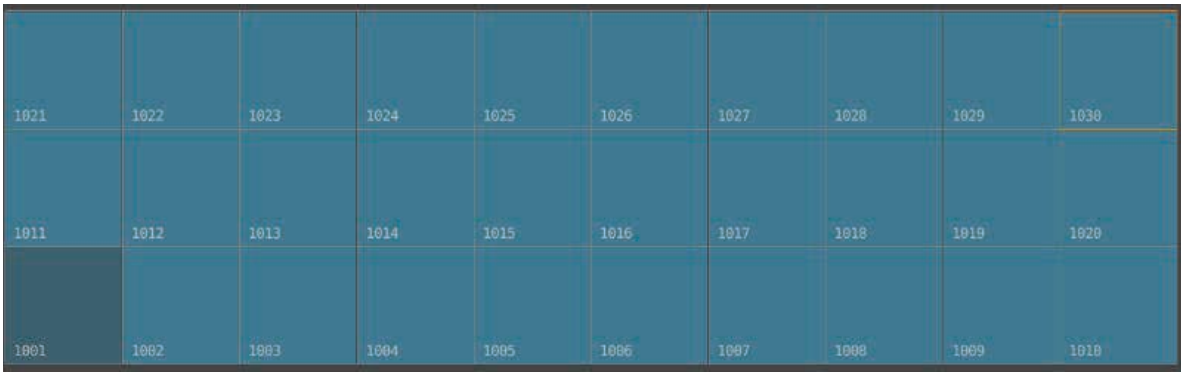
When I'm talking about "really large" textures, let's put this in perspective with some real-life numbers in case you're thinking 8K or 16K textures. In a film production, a monster character like the one pictured above (think **Lord of the Rings** for example, the production for which UDIMs were invented) has not just a resolution of 8K or 16K. They have maybe *15 UDIM textures, each at 8K resolution*. Let that sink in for a moment.

If you do the math, that's a total texture resolution of *120.000 x 120.000 pixels **(120K)***! Obviously, that's going to be hard if not impossible for any computer to load as a single texture, never mind how inconvenient that would be to work with for the artists. It doesn't stop at 120K - some characters for movie visual effects use *hundreds of UDIM texture tiles*.

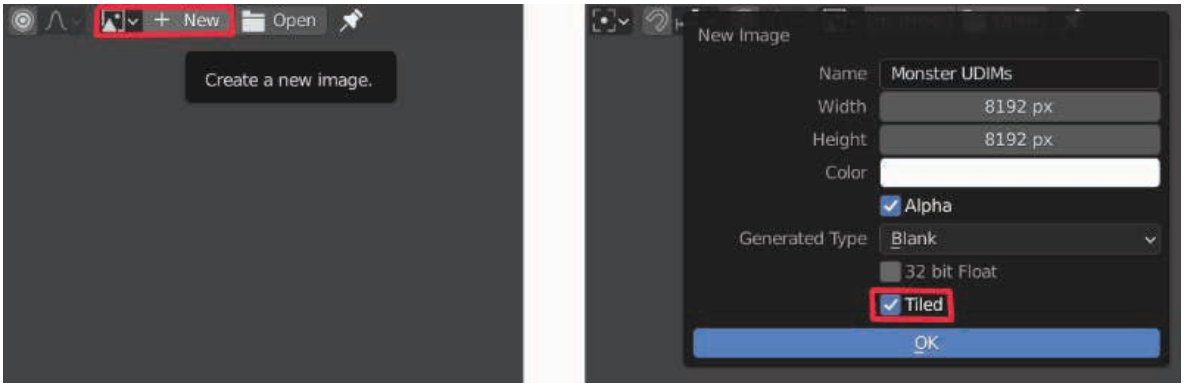
To be clear, we're not talking about different objects on a character having different textures, but one single object having different textures. The image below shows how that might work:



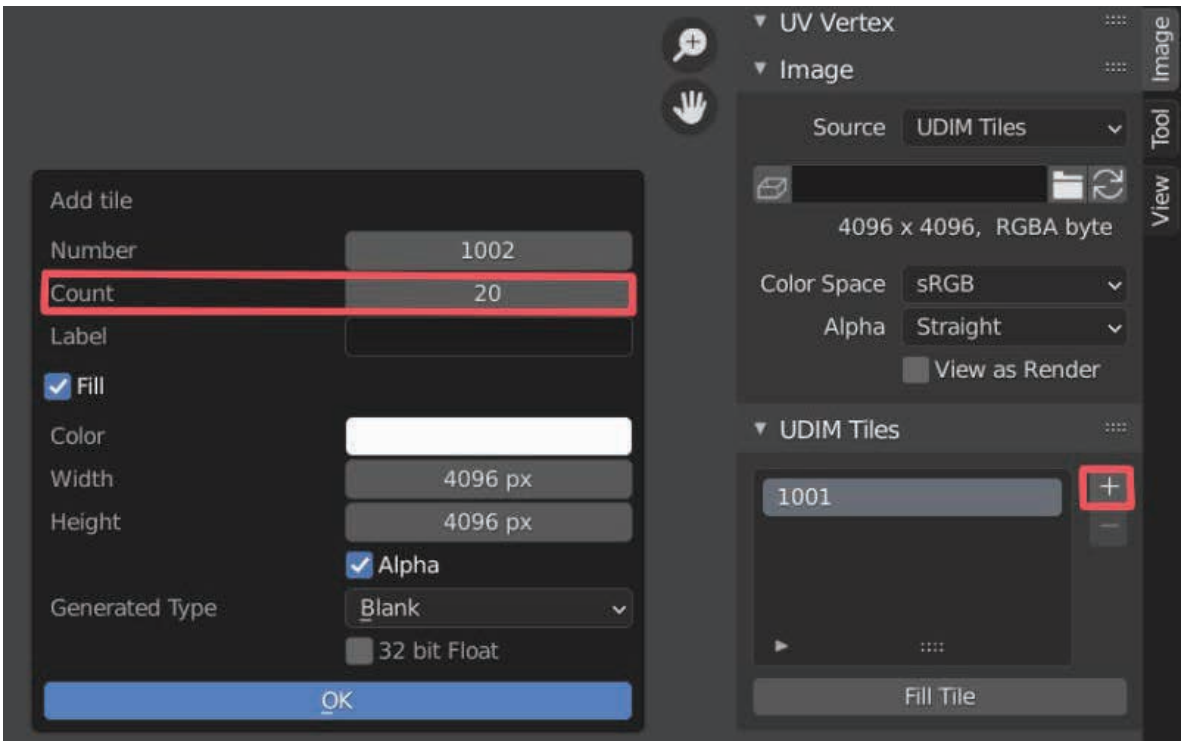
As you can see, the face is divided into parts, each of which will have its own texture tile. The numbering in the above picture is based on the naming convention of UDIMs. The first tile is always "**1001**", the one next to that "1002" and so on. After "1010" the horizontal row ends and a new one starts above "1001", called "1011". Vertically there is virtually no limit.



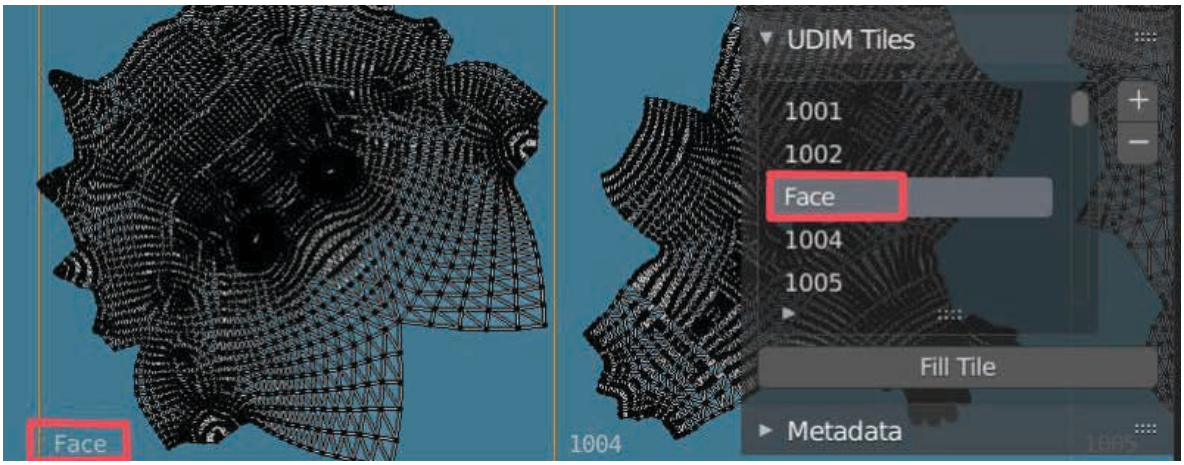
To create a new UDIM texture for your object, go to the UV Editor window and click onNew. Choose your resolution - a square format is preferred. If you want an 8K texture for example but you don't remember the exact resolution for that, you can just time *8 after the default 1024 value and it'll automatically calculate 1024*8 = 8192 for you. It's important that when creating the texture in the UV window, you check "Tiled", otherwise you won't have the UDIM options later.



By default you have only one tile. To add more UDIM tiles, **open the Option Panel of the UV Editor by pressing N** and go to the **Image** tab of the and open the **UDIM Tiles** section. Here you can **click on the + icon** to add another UDIM tile, or in the **Count** field type the amount of extra tiles you wish to add.

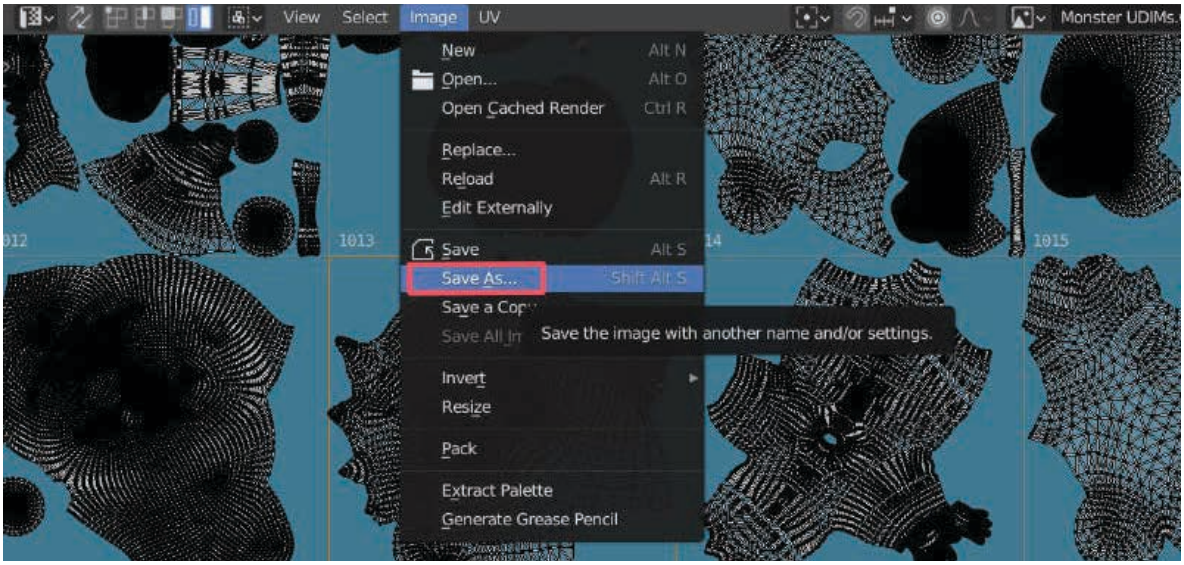


It will automatically assign it a tile number. While creating the tiles you can give them a name in the **Label** field, or you can **double-click on a tile number** in the list afterwards to change its name. This can be convenient later for knowing which UV's are in which tile. The label shows up in the bottom left corner of the tiles.

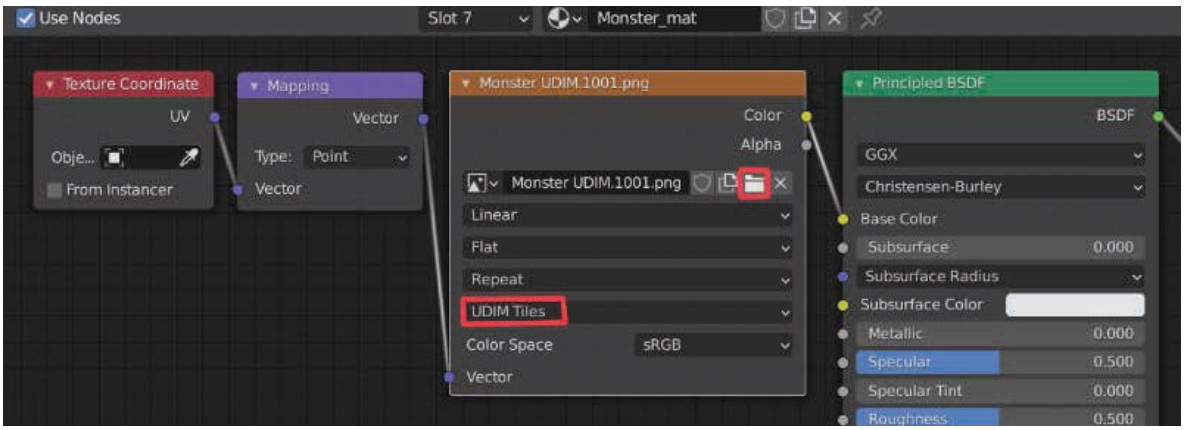


You can set different resolutions for each tile. So, what if you change your mind about where some UV's are placed, and want to move them to another tile? You can move UVs across tiles by **pressing G**, as you are already used to in Blender. Or, perhaps more conveniently, you can **press G+Y+1** for example to move the UV's one tile up, or **press G+X+1** to move them one tile to the right. Keep in mind that when you pack UV islands to fill up a tile, Blender will pack them on tile 1001. So do any UV packing before moving them to a different tile.

To save the textures separately go to **Image > Save As** and save your textures. It will automatically save it as separate textures, one per tile, in the resolution that you set for each individual tile.



To use these UDIM textures in a material works almost exactly like you'd expect from a regular image texture. You load one of the files in the ImageTexturenode, and because of the file name format (the files automatically all end with 1001, 1002 and so on) Blender will know which files belong together as a set of tiles and load the rest of them as well. The Source will be automatically set to "UDIM Tiles".



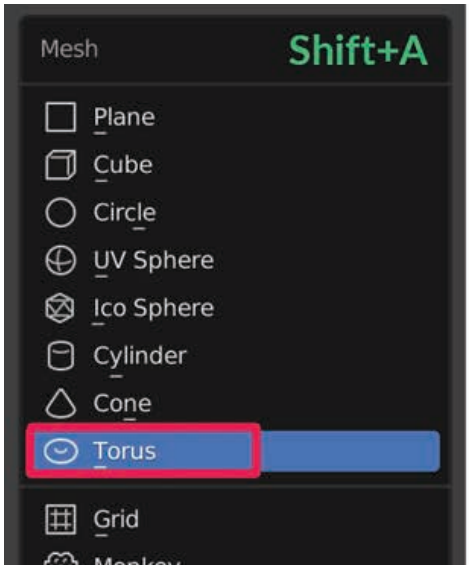
So that's how UDIMs work in Blender, and why you should be using them if your models are highly detailed, or if you just want to have more texturing flexibility in your workflow. Now it's up to you to create some detailed monsters!

UV Unwrapping Cylindrical Objects like a Cyber Donut

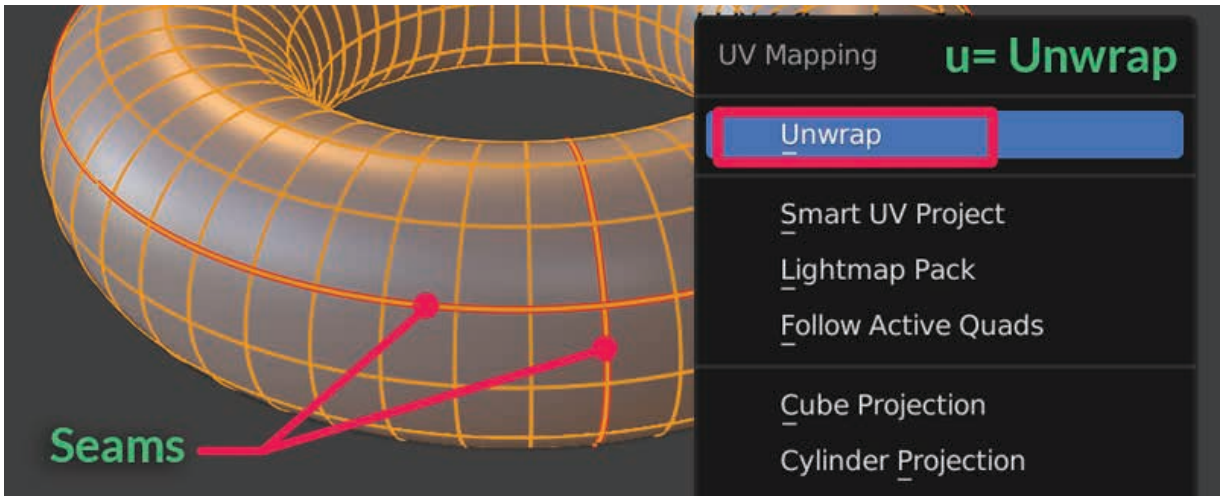
Donuts are all the rage in the Blender community.

Let's assume this will still be the case in the far future, and prepare a "cyber donut" for the occasion.

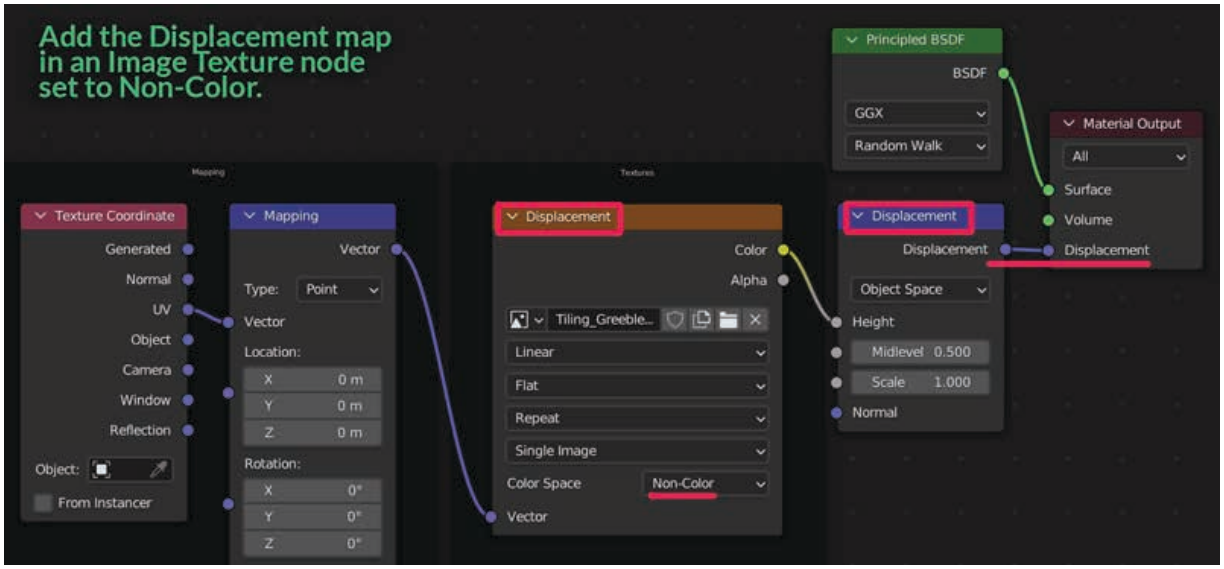
Easy enough, just **press Shift+A** and add a **Torus**. Then, in Edit Mode **hold Alt** and click on a horizontal edge loop to select it, right-click and choose **Mark Seam**.



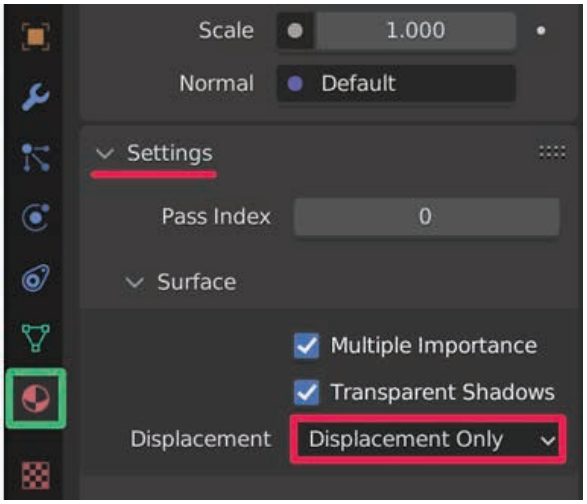
Do the same for a vertical edge loop. Then **press A** to select all Faces and **press U** and choose **Unwrap**.



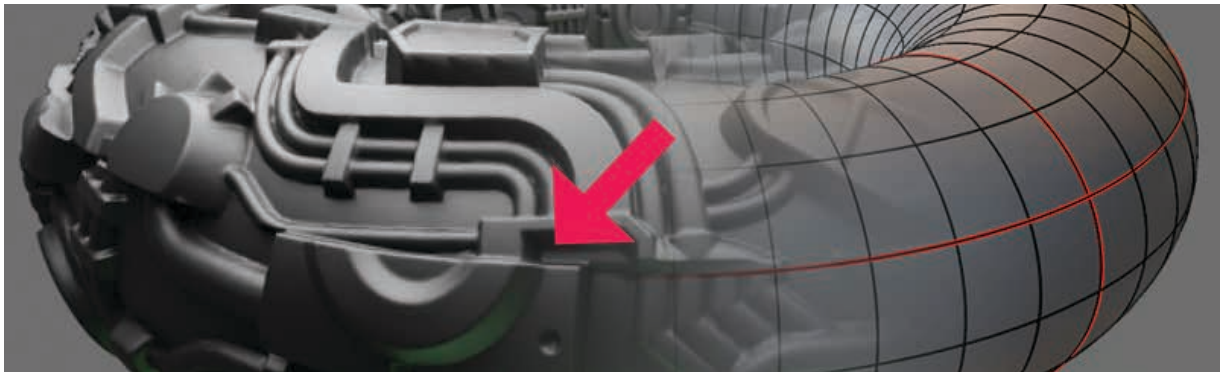
Now that it's unwrapped, we can add a texture. I use a 32-bits Displacement map plugged into a **Displacement Node**.



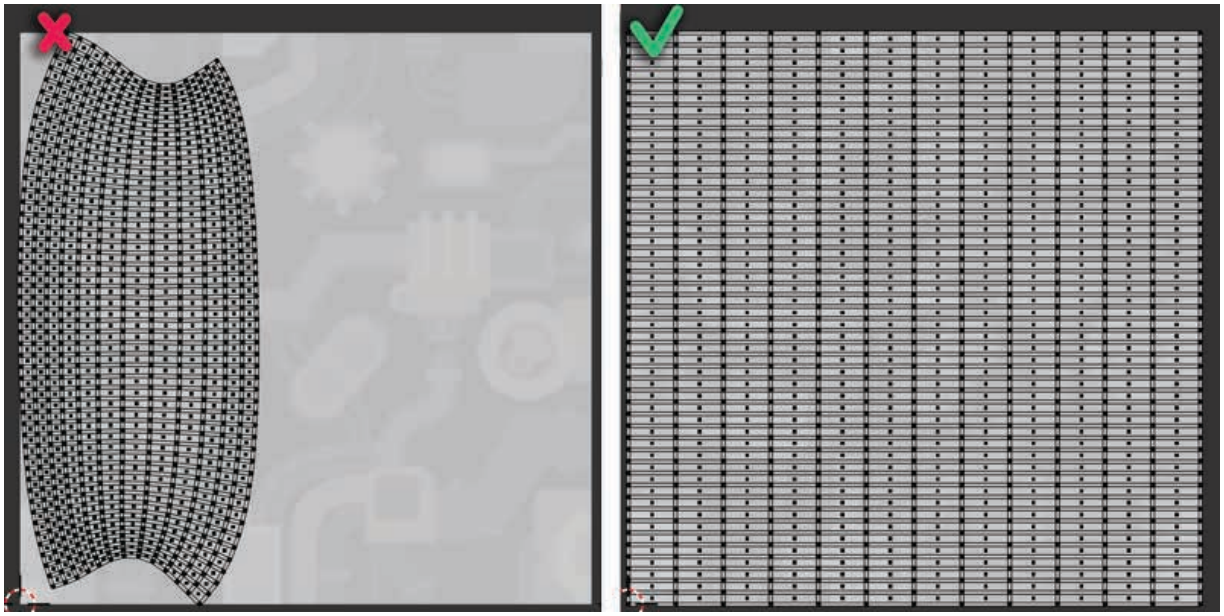
In **Cycles**, in the Material settings set it to **Displacement Only**. Now the displacement effect will show up in Rendered view.



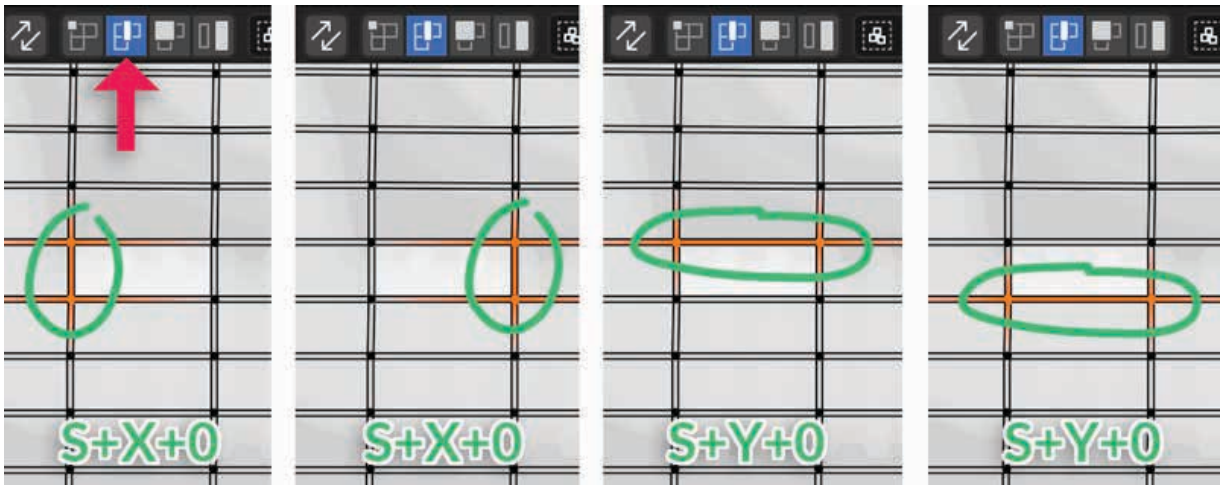
However, although the texture itself tiles perfectly, we can still see an ugly seam where the UV map seam is.



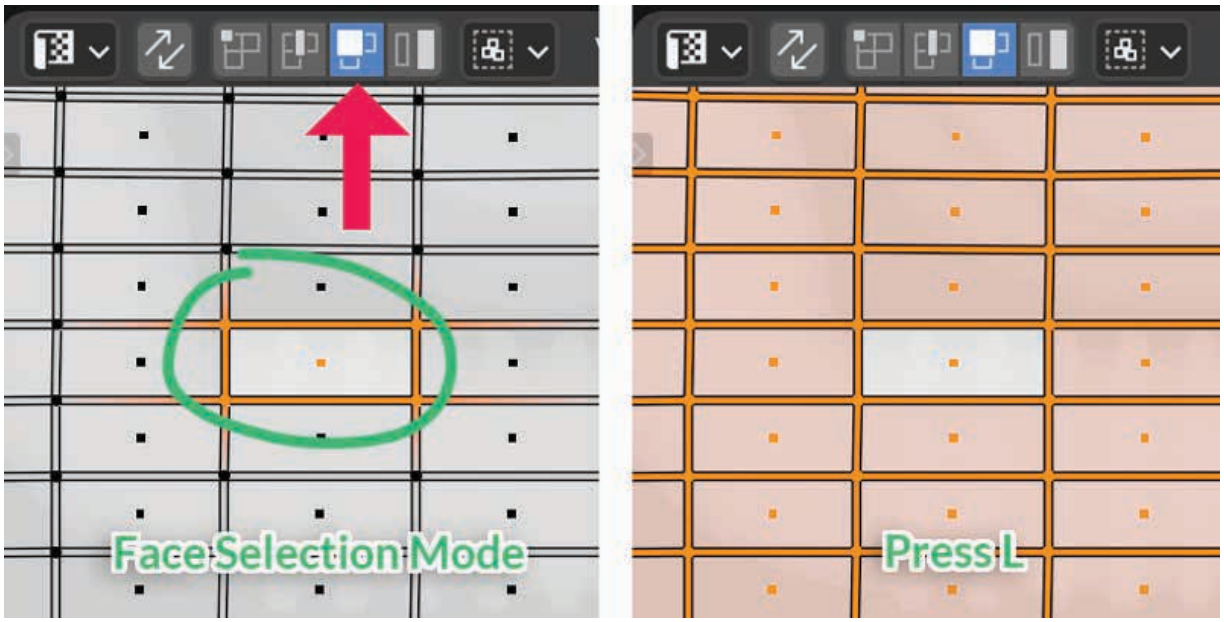
It would be nice if we could take advantage of the seamless tiling inherent to the Displacement map we used, and avoid seeing that seam. The reason we're seeing it, is because the unwrapped UV map looks like below left. And what we really want is for it to look like below right: square and filling the entire UV tile. That way both the horizontal and vertical seam on the torus will not be visible.



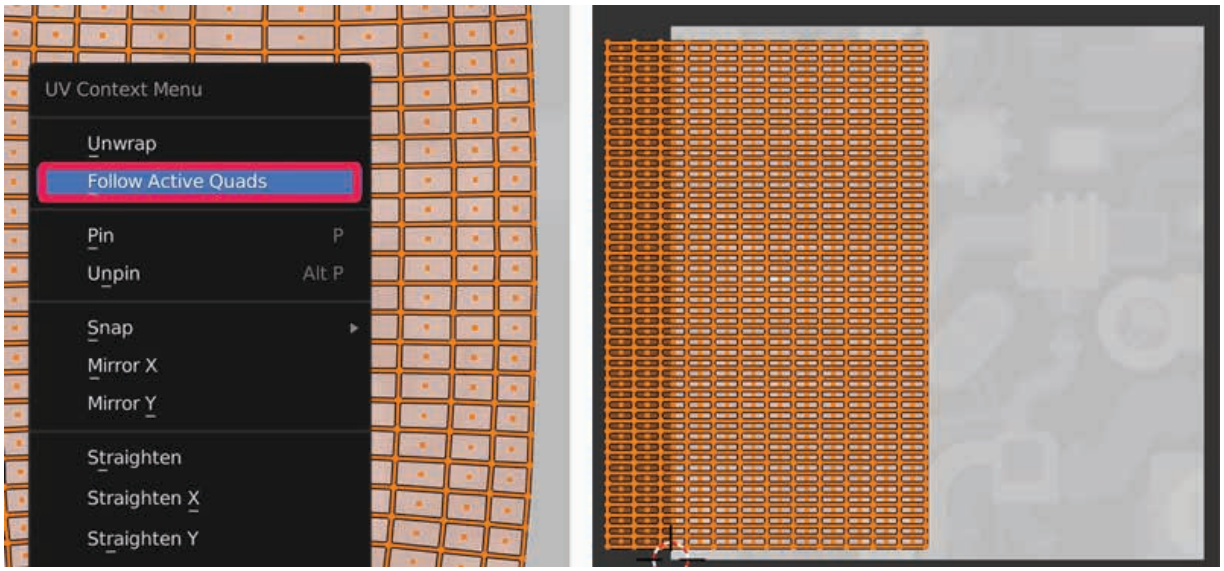
To make all the UV Faces square, choose one Face somewhere in the middle, and make the edges straight by selecting them one at a time, and **pressing S+X+0** on the numpad for the vertical ones, and **S+Y+0** on the numpad for the horizontal ones. This gives you one perfectly square Face. Do this with **Edge Selection mode**.



Next switch to **Face Selection mode**, and select the Face that you made perfectly square by clicking on it. Then, **press L** to select everything else connected to it (the "UV island").

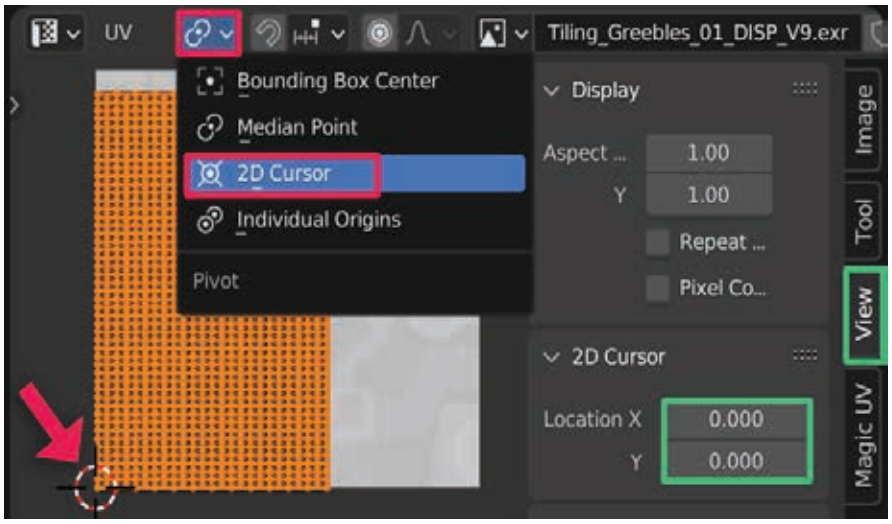


Right-click and choose **Follow Active Quads**. Now all the UV's are laid out as rectangular Faces, which is good. But we need to fill the whole UV tile with them.

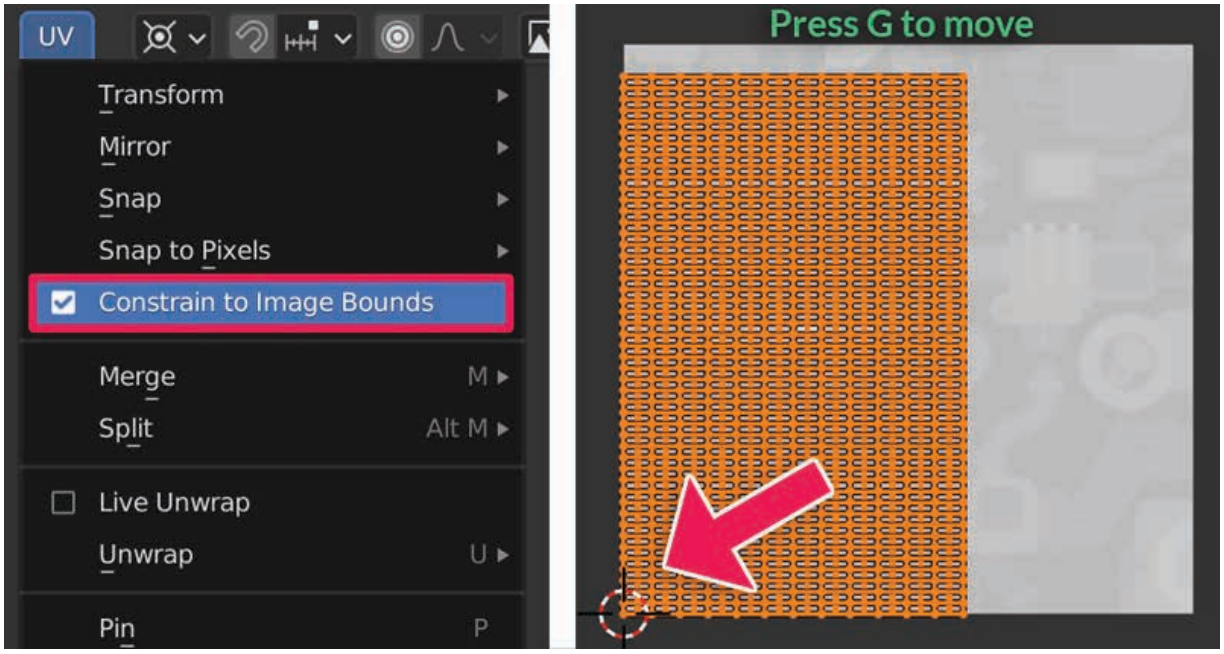


To do this, set **Pivot Point to 2D Cursor**.

If the 2D Cursor is not at the left bottom corner of the UV tile, reset it there by changing the X and Y location values to 0.



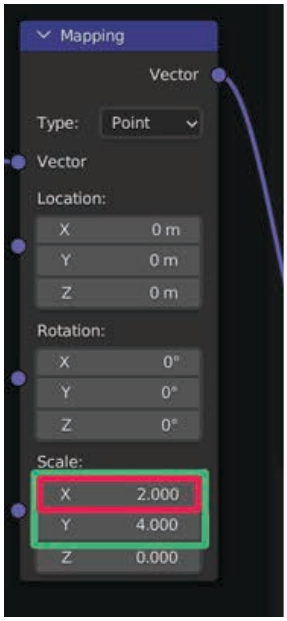
Enable **Constrain to Image Bounds** from the UV menu. Then **press G** to move the UV's, and move them to the bottom left corner. They will snap to the sides because Constrain to Image Bounds is enabled.

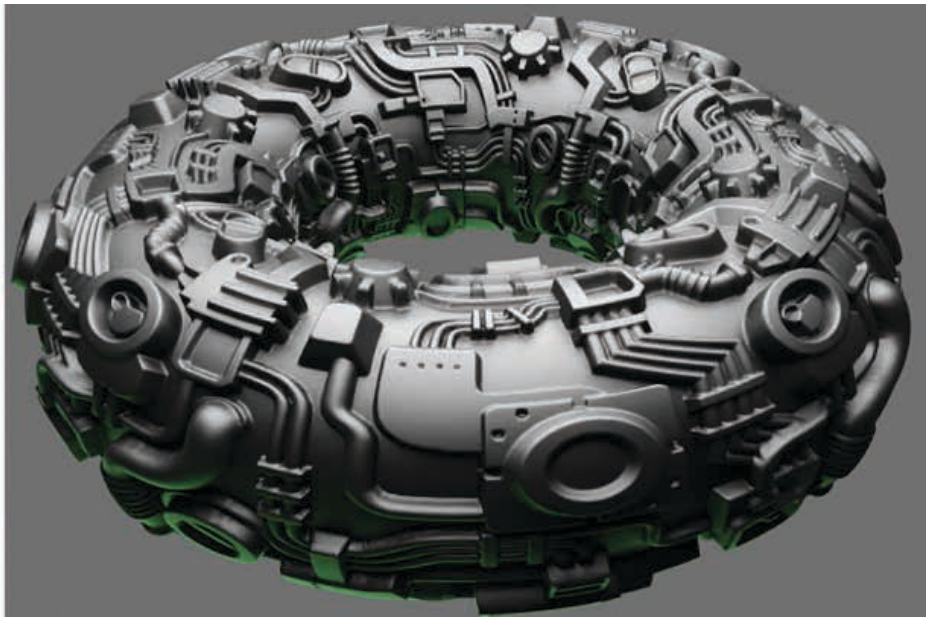


Now simply **press S** to scale up the UV's until the tile is completely filled.

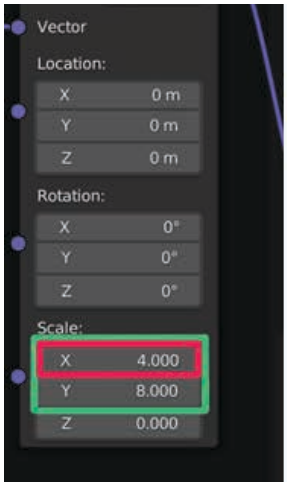
It will snap to the sides, so you can't scale too much. Now we have definitely solved the problem of visible seams (provided that the texture itself tiles seamlessly, of course). But there's a new problem: stretching of the texture, which comes from having scaled the UV's disproportionally in order to fill the tile.

This is easily solved however. We simply change the **X Scale** on the Mapping node from 4 to 2. Looks much better!



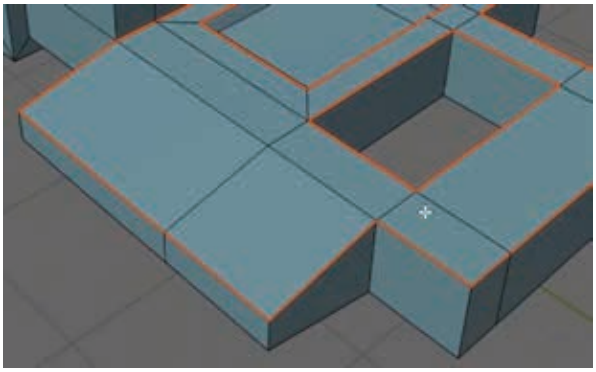


If you want to increase the detail density, you can increase the X and Y values. Just use whole numbers.

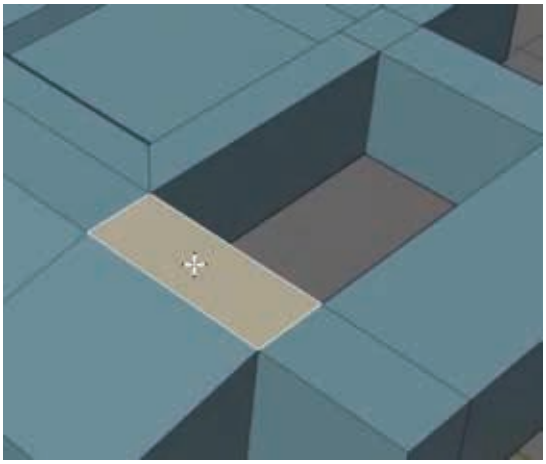


Easily mark seams on complex shapes

In this tutorial we will look at how to quickly select specific edge loops on a complex shape, with the intent of marking them as seams for UV unwrapping.



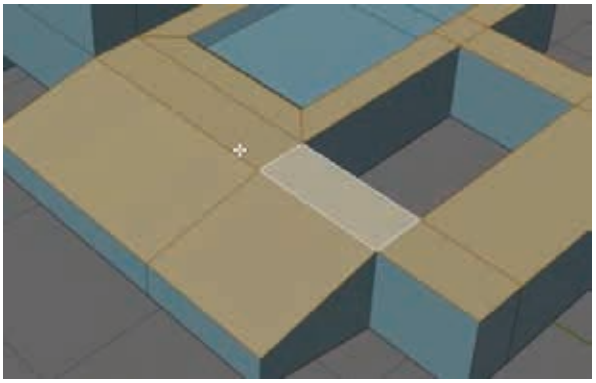
Select one face on the area you want to mark seams on.



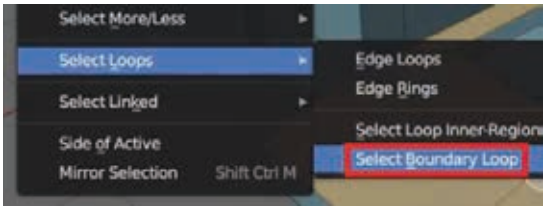
Go to Select > Select Linked and choose **Linked Flat Faces** from the menu.

If necessary you can increase the angle of theselection (**Sharpness**).

Now the faces that are connected to the initially selected face are selected.

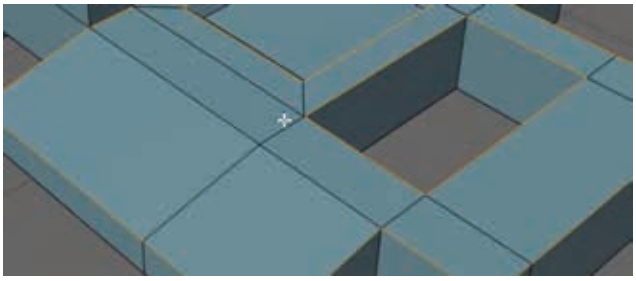


Go to select > Select Loops > **Select Boundary Loop**.

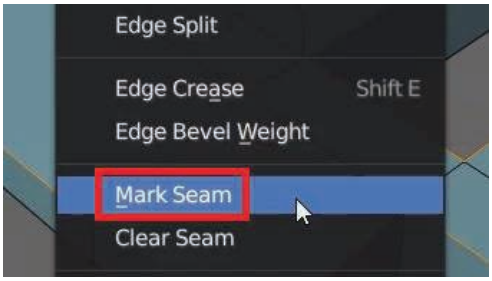


Click here to return to the index.

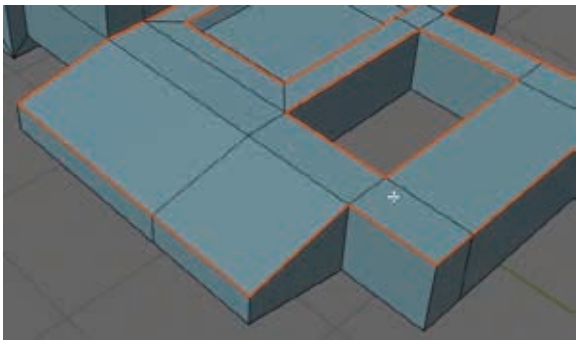
Now the edges at the boundary of the face selection are selected.



Press Ctrl+E to open the Edge menu and choose **Mark Seam**.



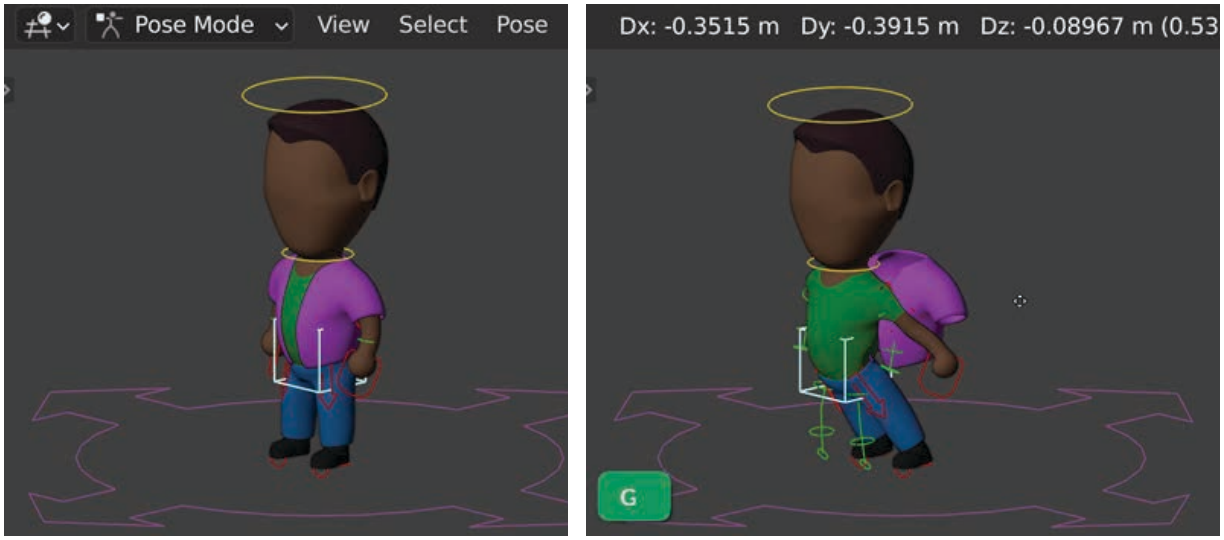
Now the edge is marked without having to select all the edges by hand.



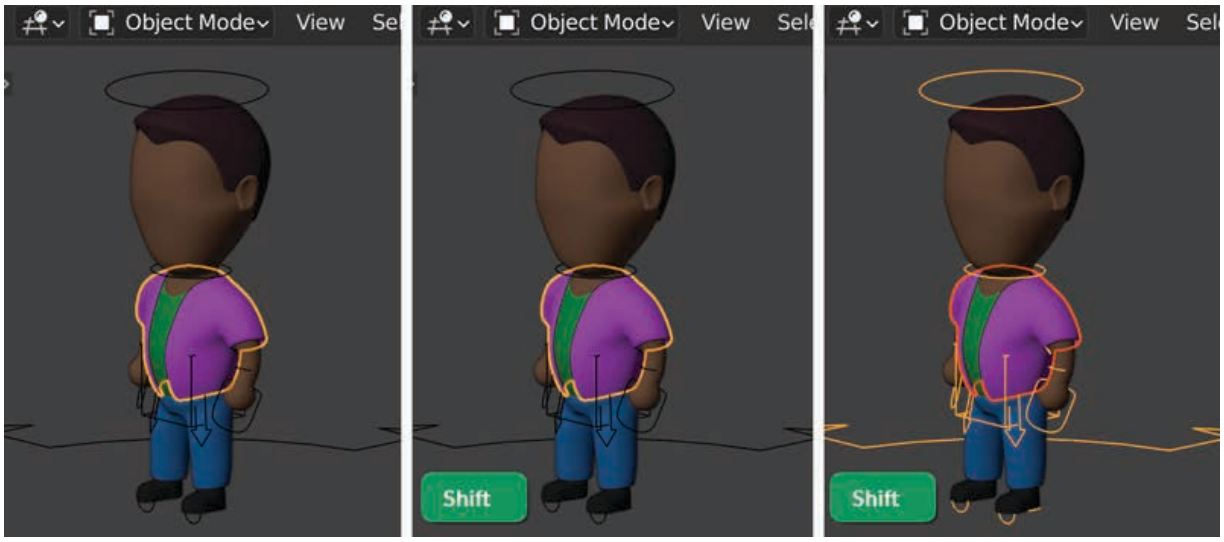
RIGGING

Easy rigging of clothes in 30 seconds

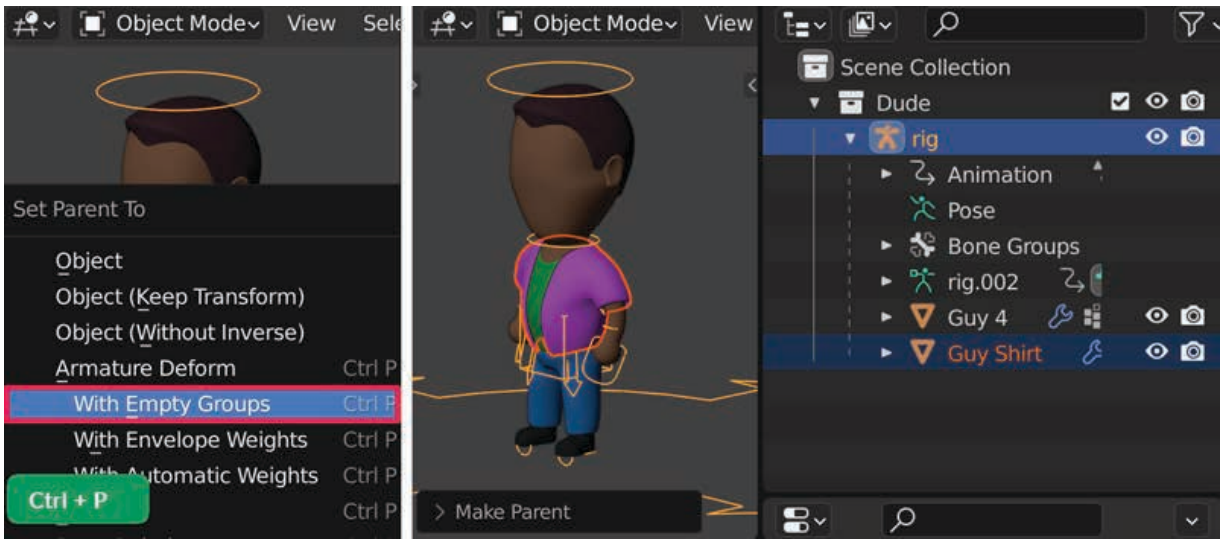
This character is already rigged but the shirt is not.



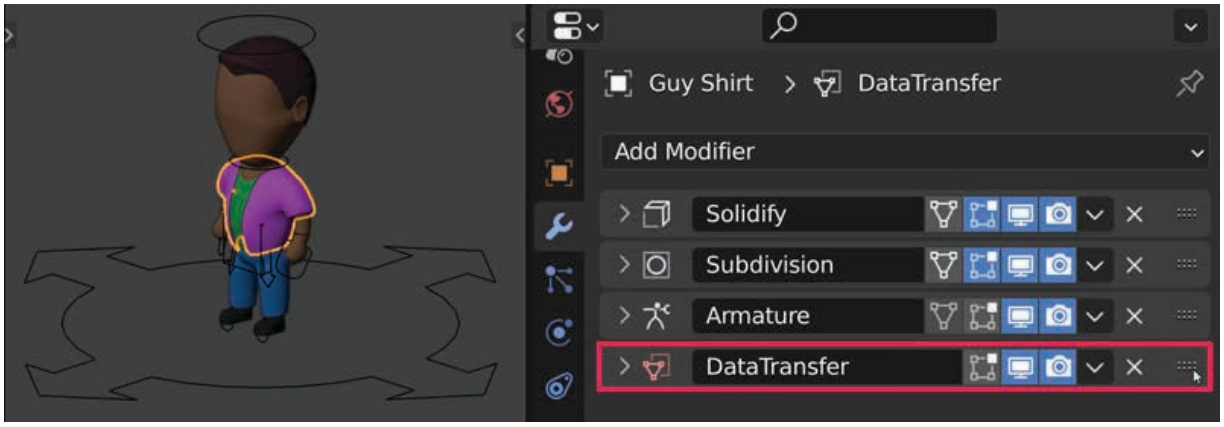
Select the shirt, then **hold Shift** and select the armature.



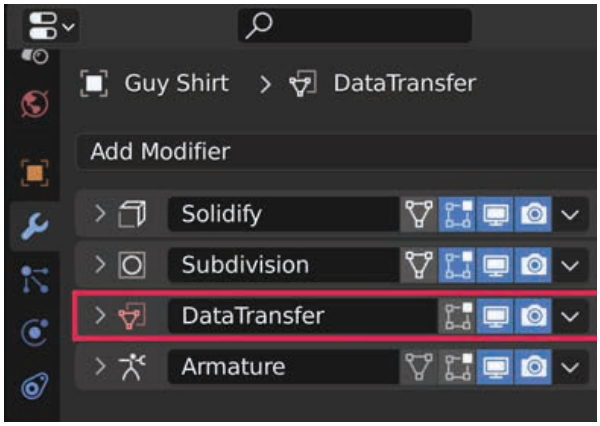
Press Ctrl+P and parent the shirt to the Armature "**With Empty Groups**".



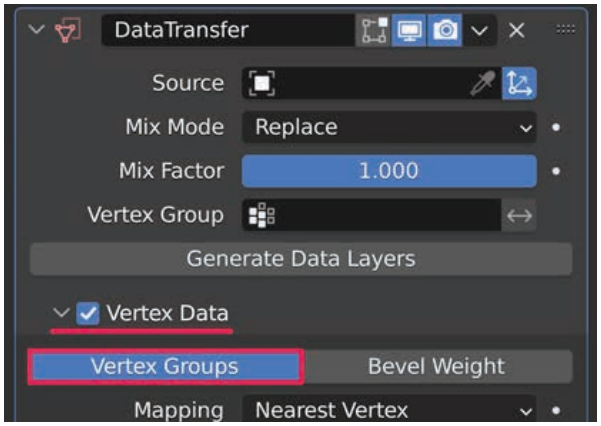
Now add a **Data Transfer modifier** to the shirt.



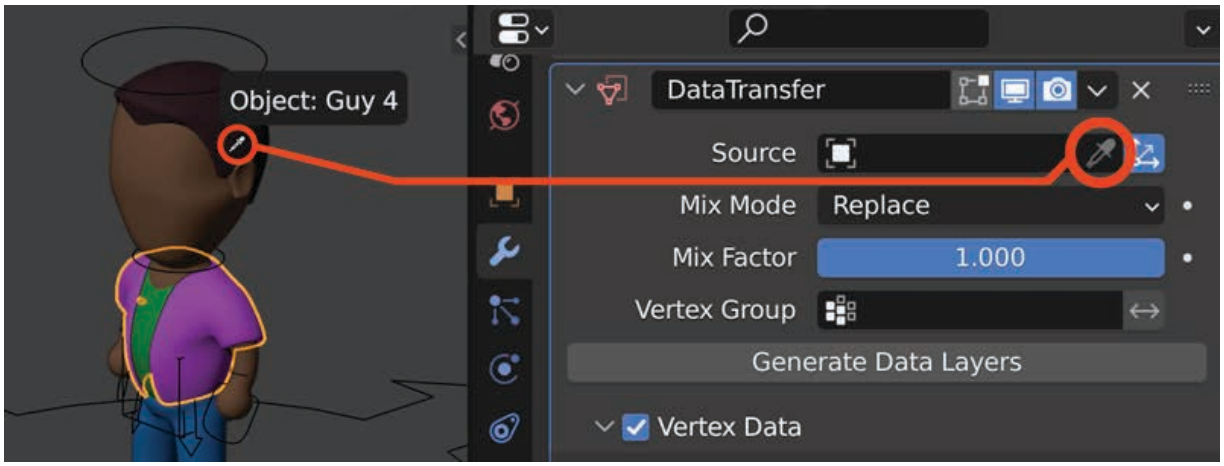
Move the Data Transfer modifier above the Armature modifier.



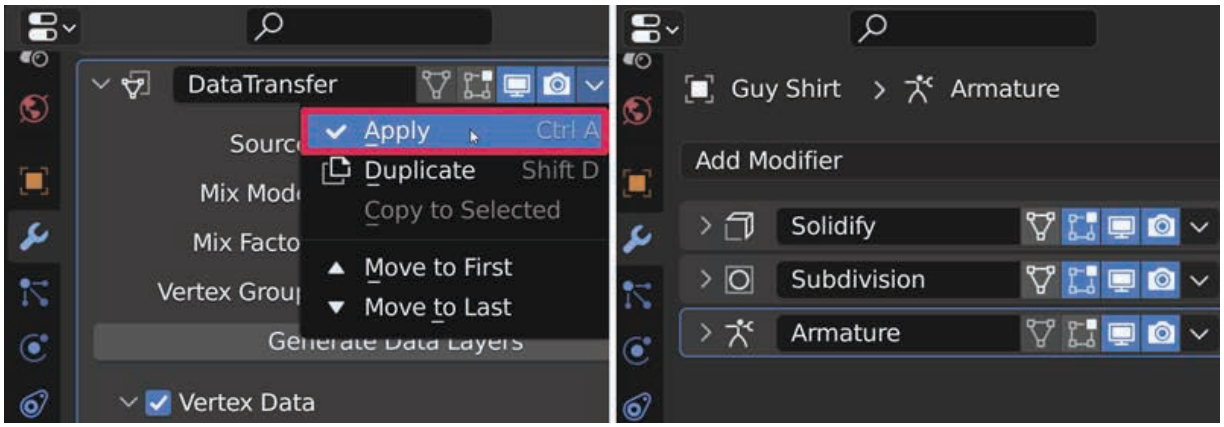
Turn on "**Vertex Data**" and "**Vertex Groups**".



Then as the **Source Object**, pick the character.



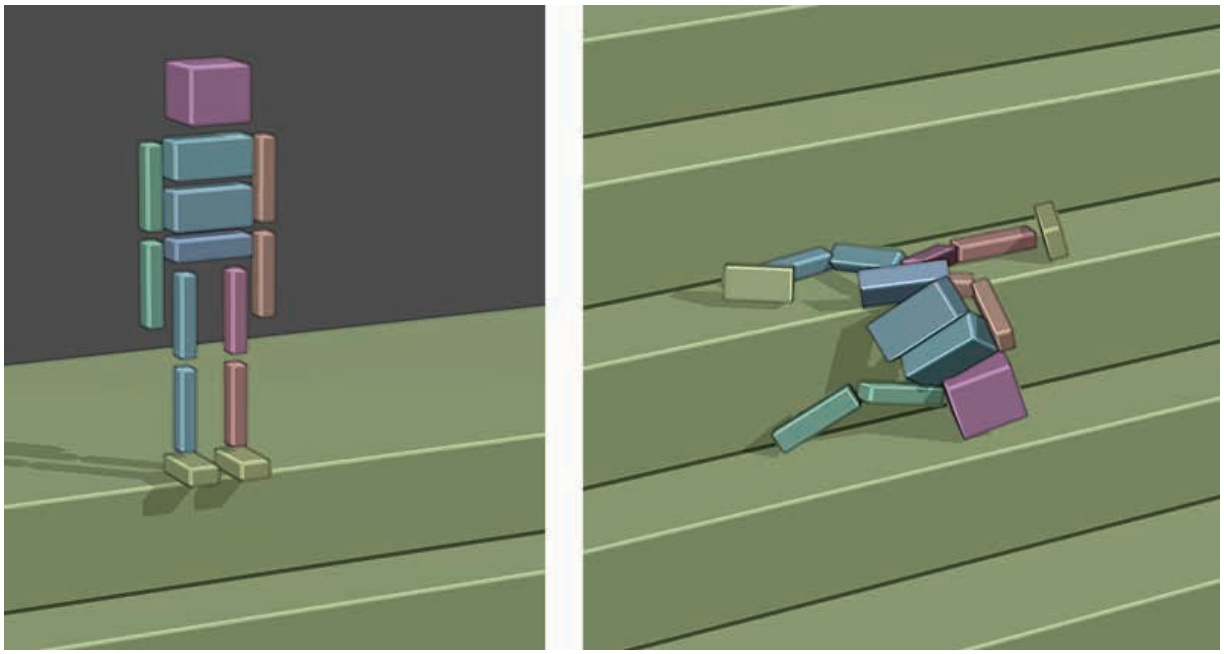
You can now click on **Apply** on the Data Transfer modifier.



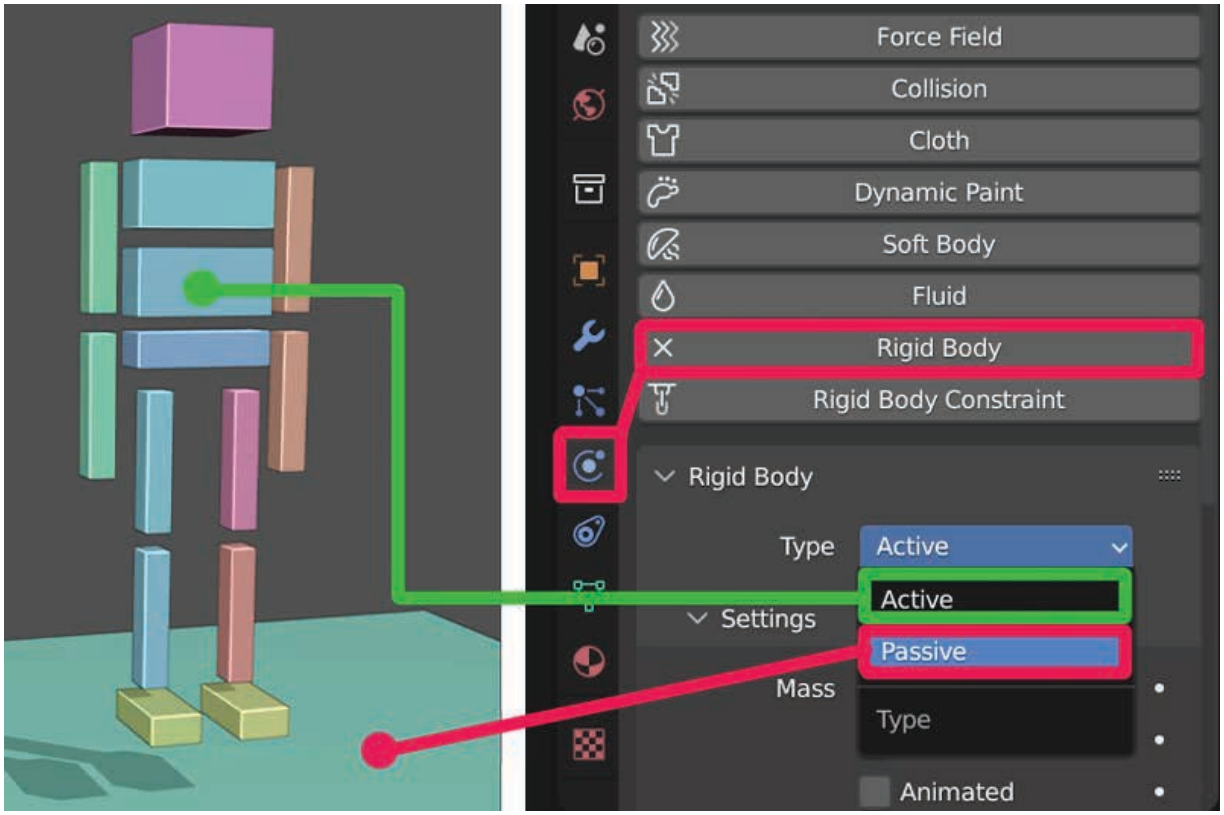
Now the shirt is rigged and you can move it with the armature.

Ragdoll Rigging part 1 - The easy method

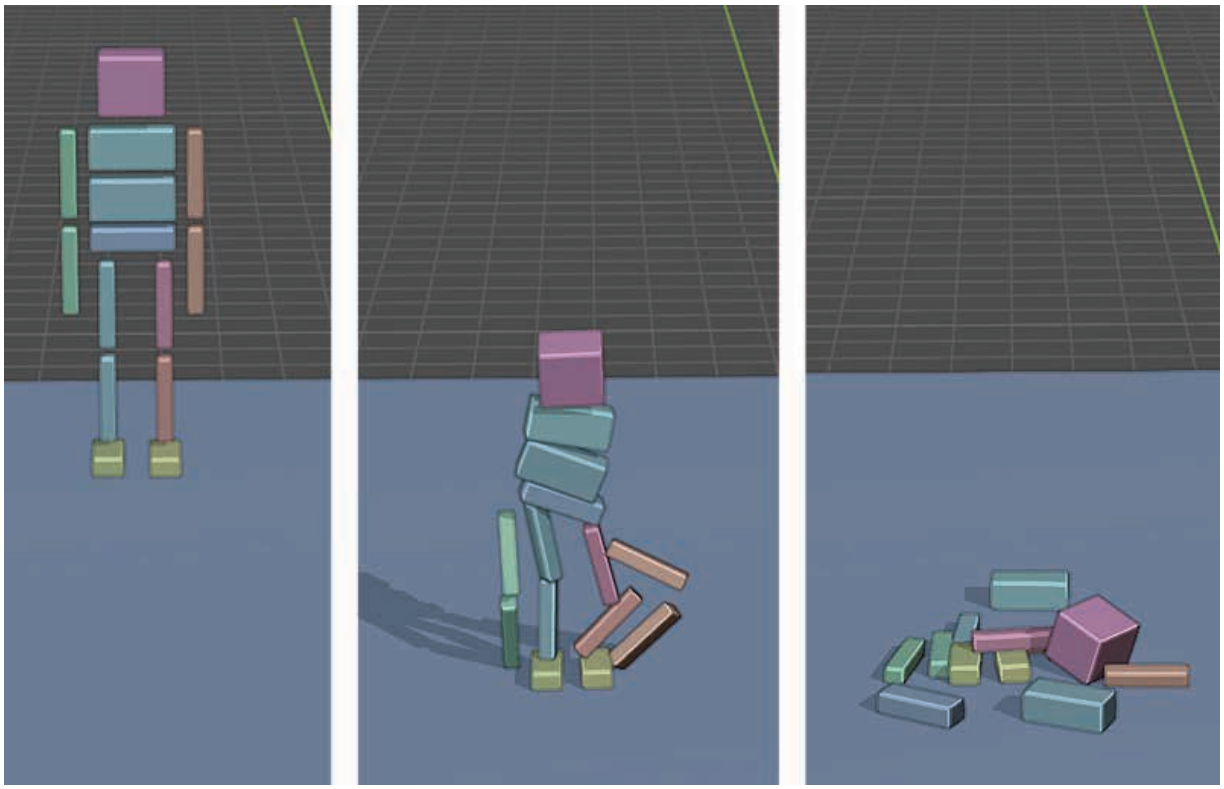
Ragdolls are physics simulation-driven “dolls” that can be connected to an Armature. In other words, you can use them to have your character fall and crash into stuff. It takes a bit of work to set up, but once that’s done, it’s a lot of fun. And first, we look at the “easy way”.



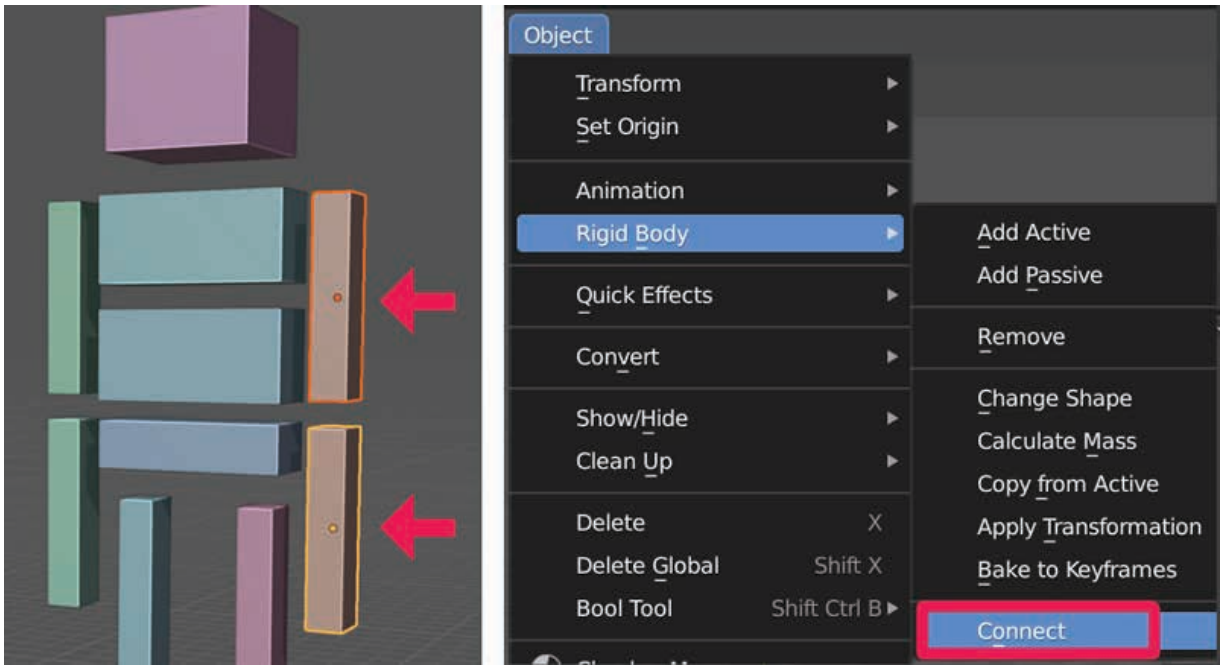
Create a floor plane. Set it to Rigid Body by clicking on **Rigid Body** and set it to **Passive**. Create a ragdoll from a bunch of cubes. Set all the ragdoll cubes to **Rigid Body: Active** (which is the default type of Rigid Body).



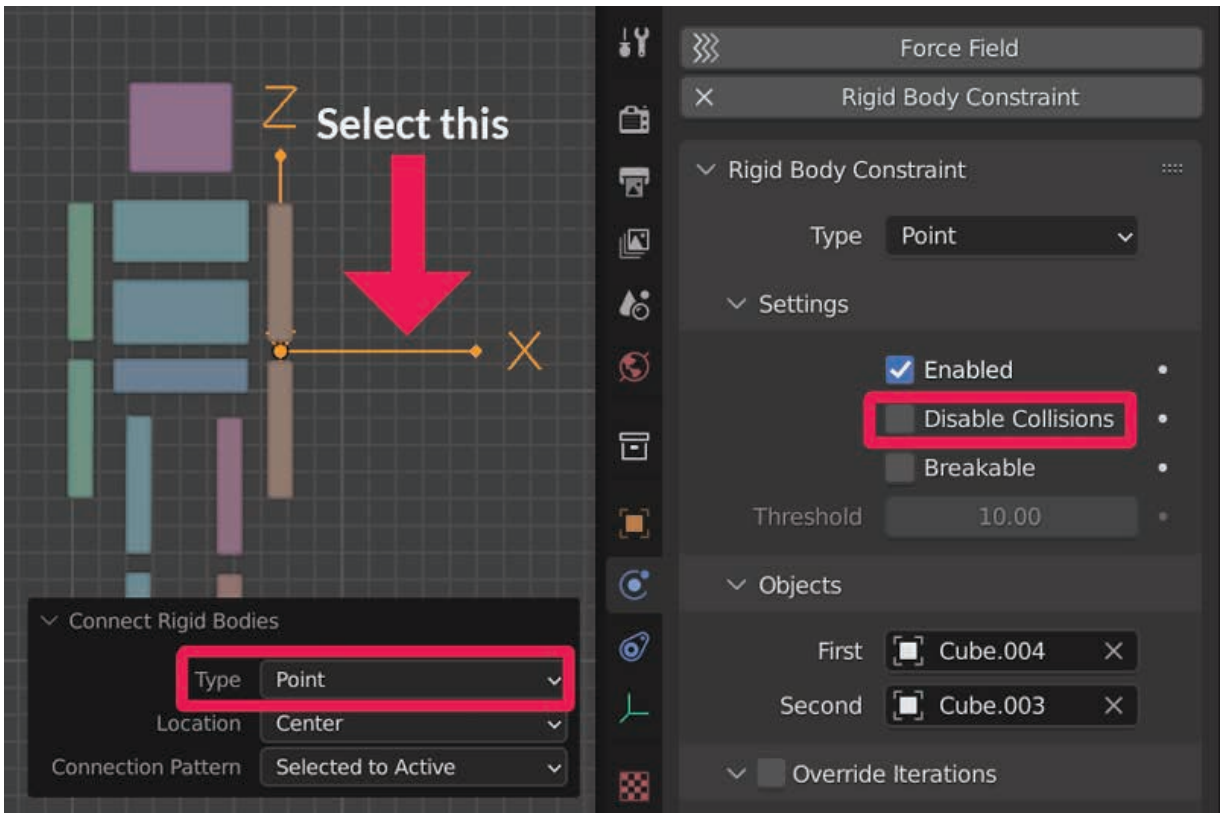
If we play the simulation now by **pressing the Spacebar** the cubes all fall down. So far so good.



Now to connect all the body parts, select two at a time and go to the Object Menu, Rigid Body > **Connect**.

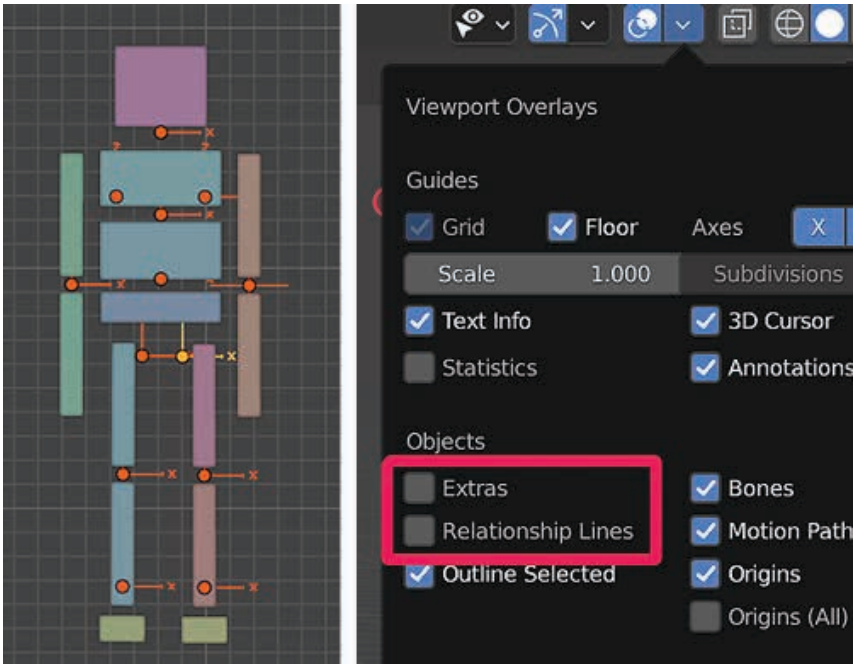


In the Connect menu that now appears in the lower left corner of the viewport, set Type to **Point**. Select the Connection, which looks like an Empty, and **uncheck Disable Collisions**.

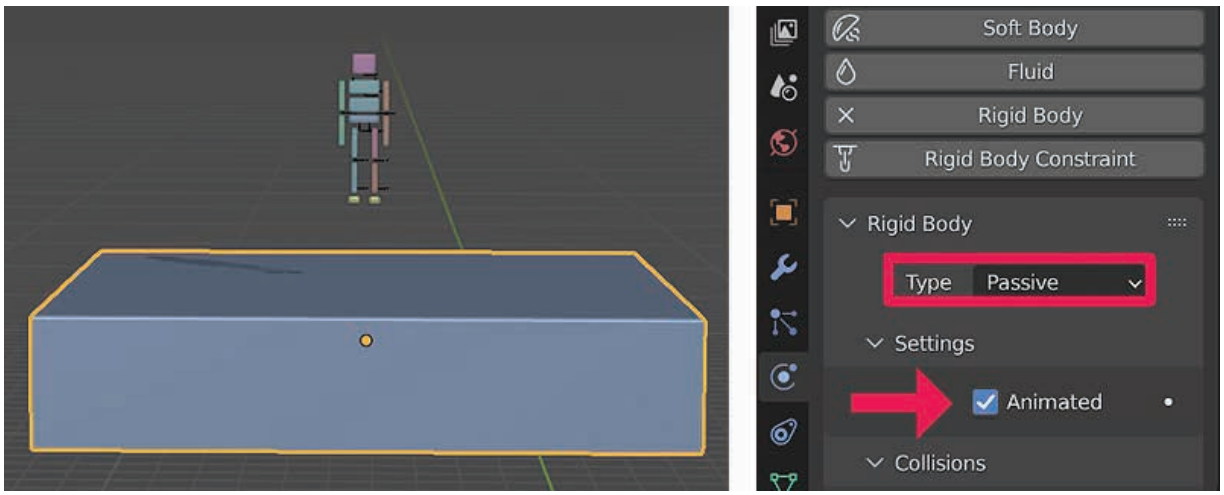


Repeat these steps for all the parts that should be connected to each other. Lower leg should be connected to the upper leg, foot should be connected to the lower leg, and so on.

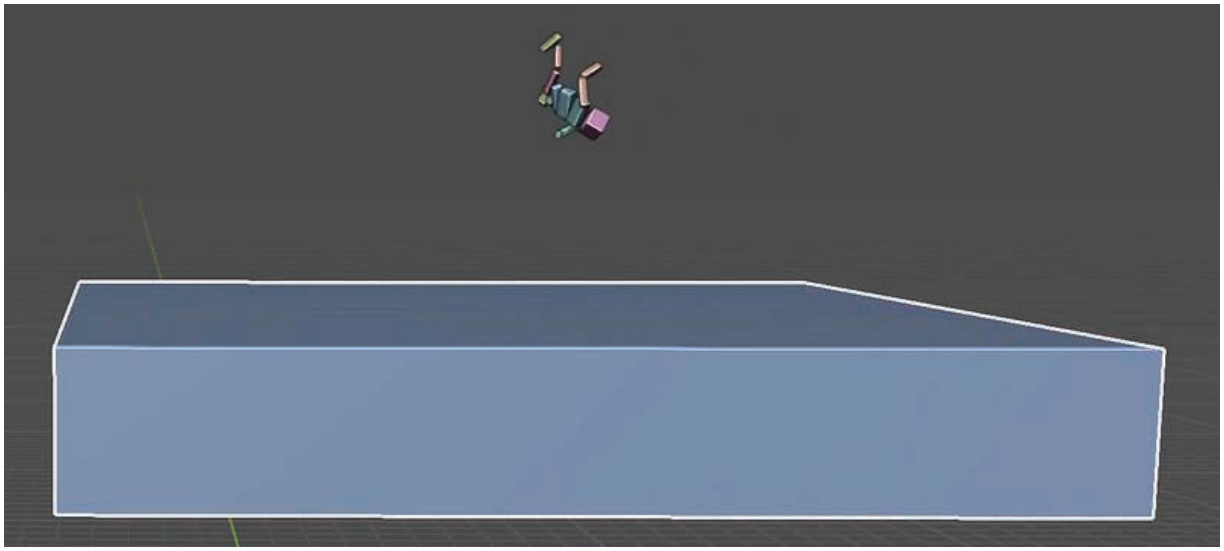
If you want you can hide the **Relationship Lines** and the Constraints (the **Extras** checkbox) under the **Overlays options**.



Play the simulation to check that everything works. To interact with the ragdoll in real time, create new object like a Cube and set it to **Rigid Body - Passive** and check **Animated**.

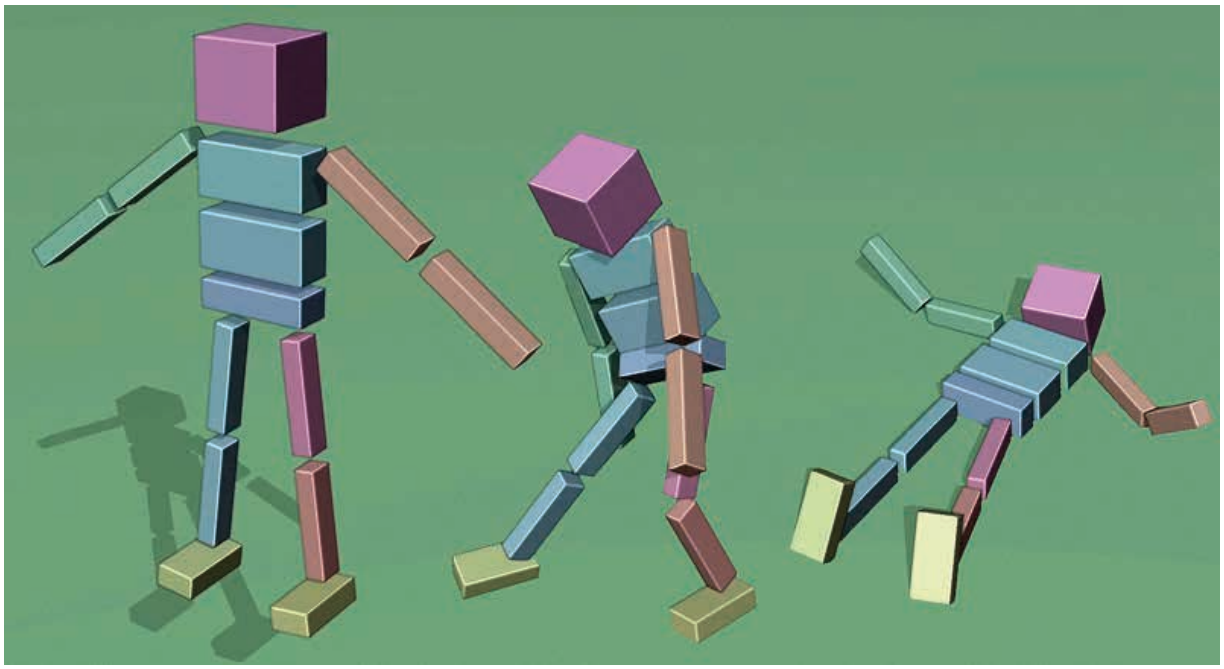


Now if you **press the Spacebar** to start the simulation, you can move the Cube by **pressing G** and have it bounce the ragdoll up. It's like a little mini-game inside of Blender! To make it easier, lower the Gravity value in the World settings.

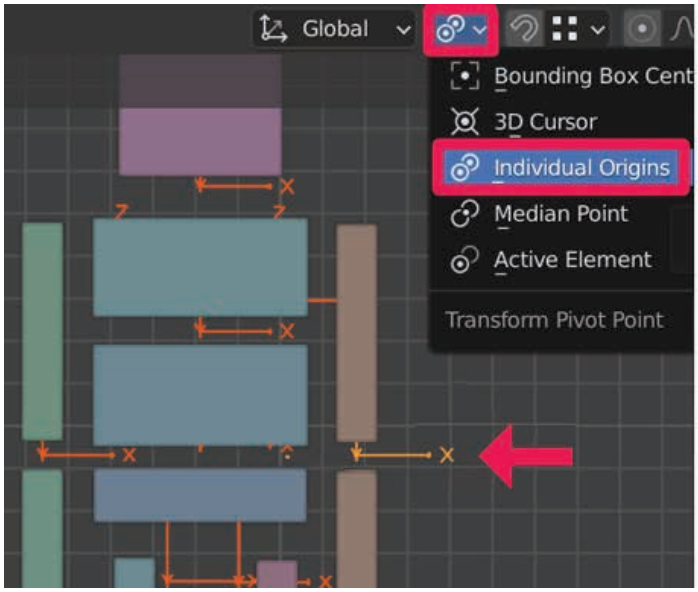


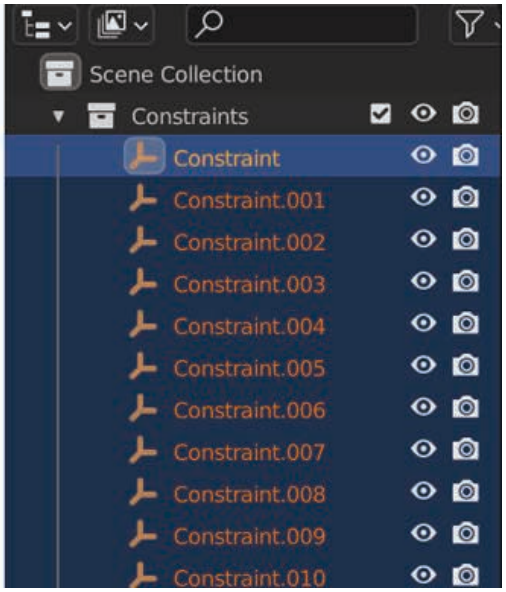
Ragdoll Rigging part 2 - More Realistic Joints

Now that we've had some fun with our very basic and rather floppy ragdoll, it's time to add some realism to it. For one, the arms and legs shouldn't be able to rotate 360°. So we'll add some constraints to the rotation of these objects.

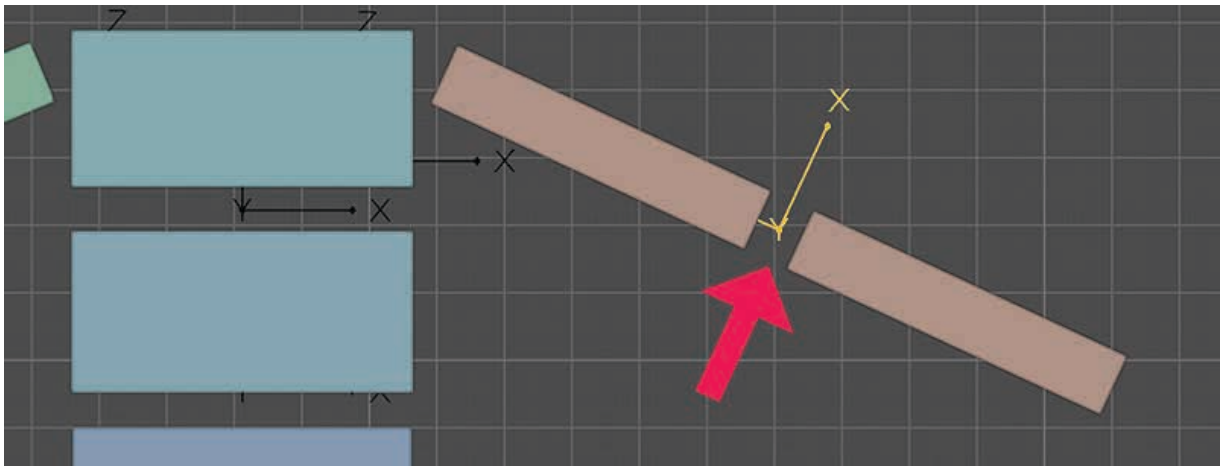


First, scale down the Constrain empties so they are easier to work with. Set **Transform Pivot Point** to **Individual Origins**, select them all in the Outliner by **holding Shift** and selecting the first and last one, then you can scale them all down at once.

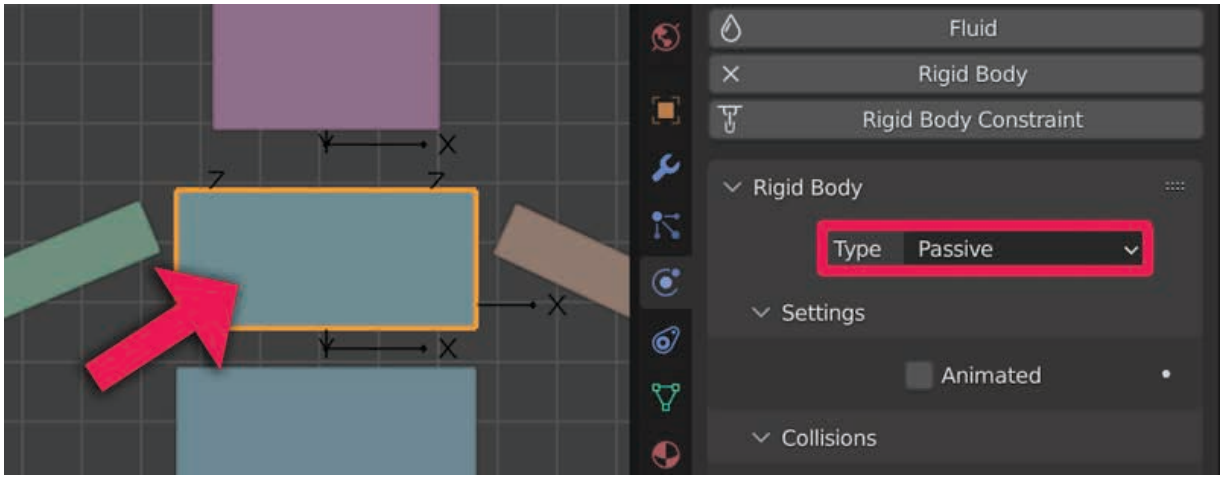




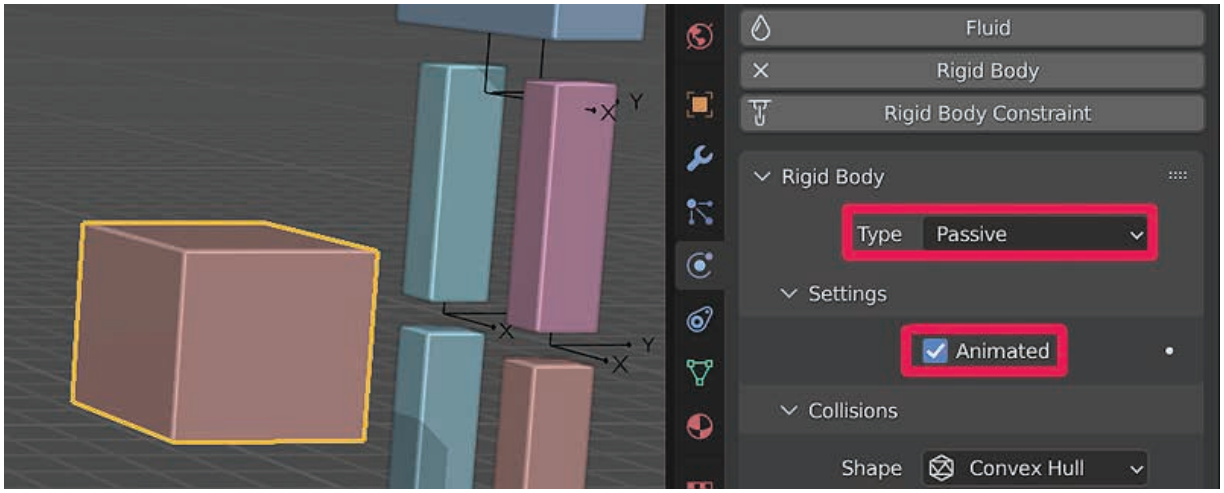
Move / rotate the Constraint Empties to be aligned with the bodyparts, in case you made your ragdoll in a T-pose or A-pose for example.



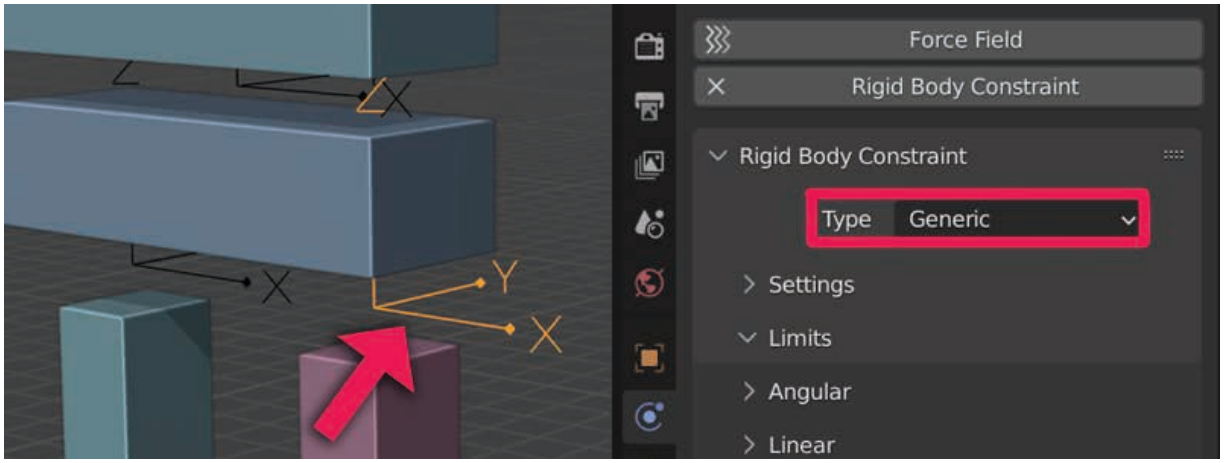
If you set the Rigid Body Type to **Passive** on the upper body cube, it won't fall down and the rest of the boxes will still be attached to it.



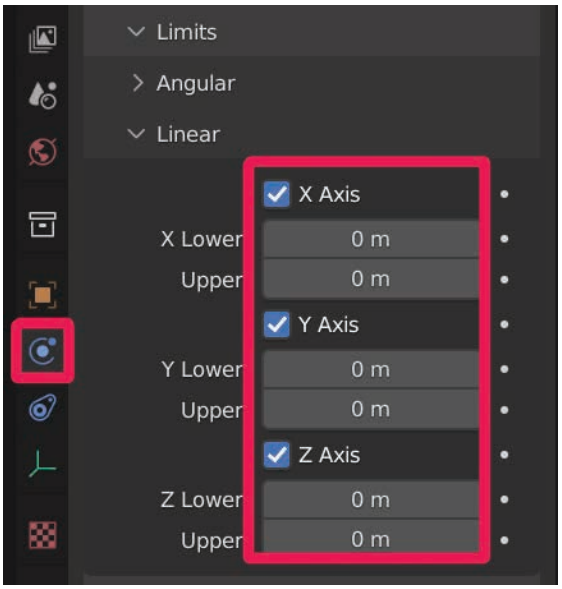
That way we can check that we are limiting the right axis, by creating a **Cube** and making it a **Passive Rigid Body** with **Animated** checked, by playing the simulation and moving this Cube against the body parts, to see how they can bend. *Note: make sure to apply Scale on the Cube, if you've scaled it down. Otherwise the collision detection won't be accurate.*



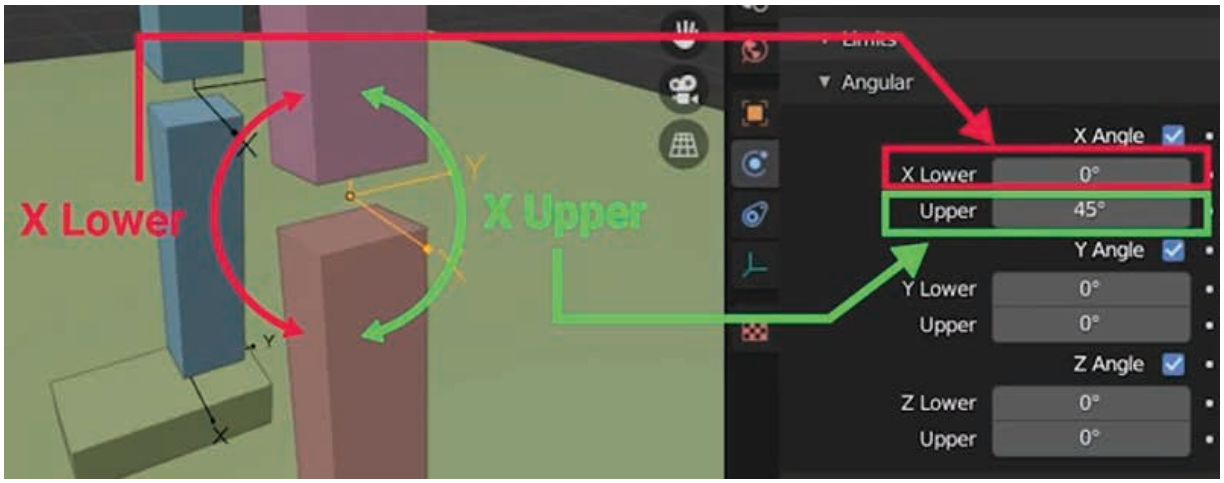
Now that we've "pinned down" the body by it's upper torso cube, and we have a Cube that we can use to test the rotations of the body parts, we can start adding some constraints. Select one of the Constraint empties. Change Body Constraint **Type** to **Generic**.



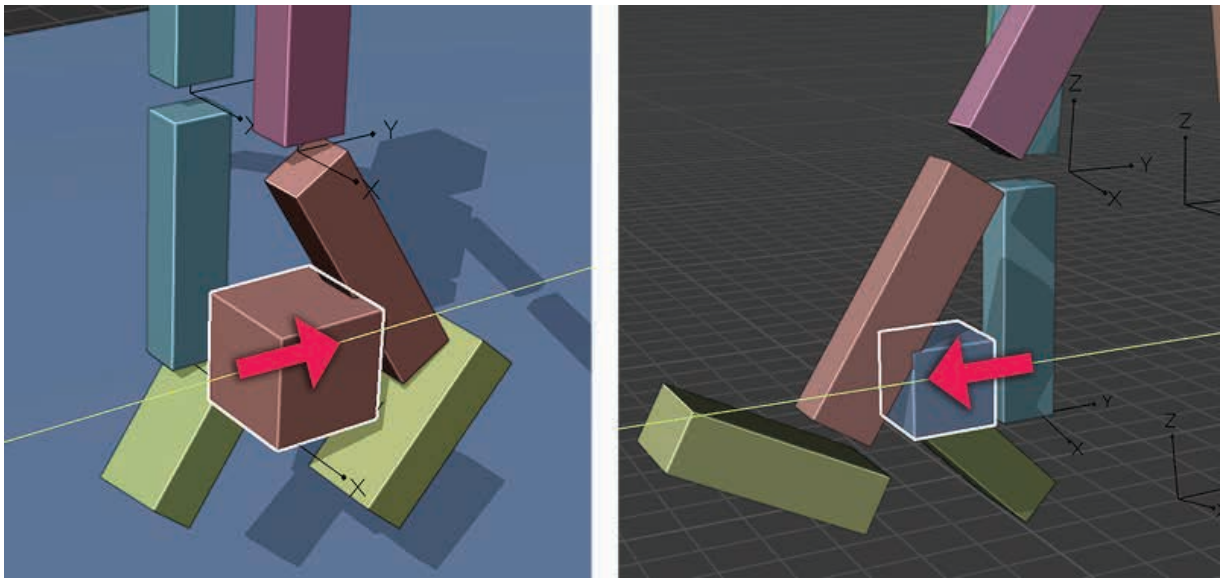
Check the **Linear** checkboxes and set them to **0**.



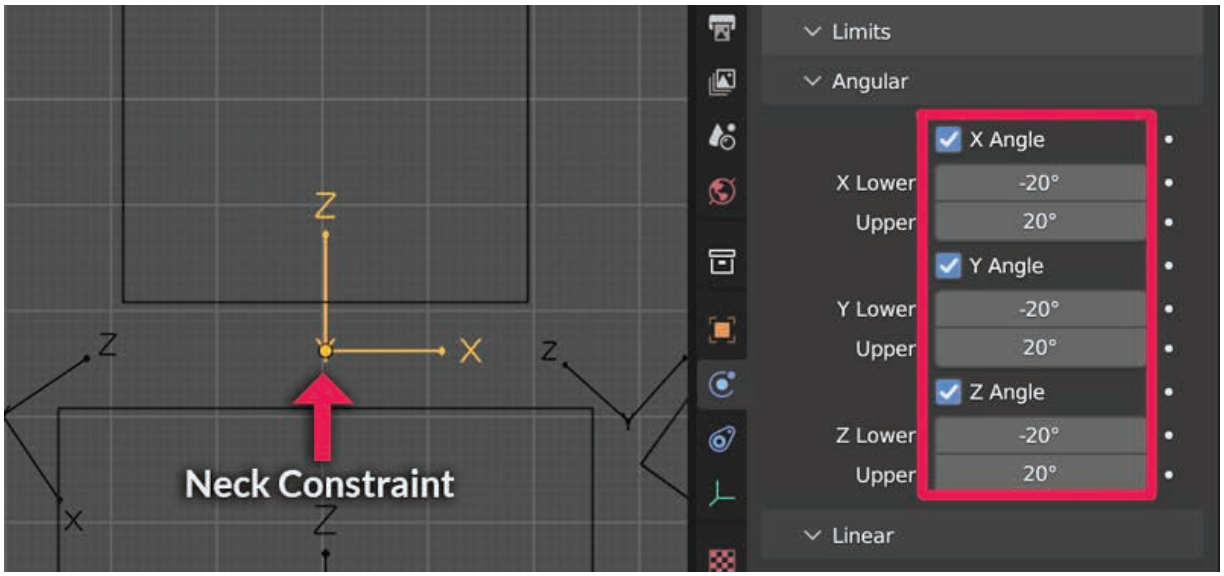
Check the **Angular** boxes as well. In the case of this joint, we only rotate around X-axis. Lower means on the left side of the X-axis. Upper means on the right side of the X-axis.



Now we can test the limbs one by one (with others set to Passive where needed, to keep everything from moving too much). They should only be able to rotate in realistic ways.



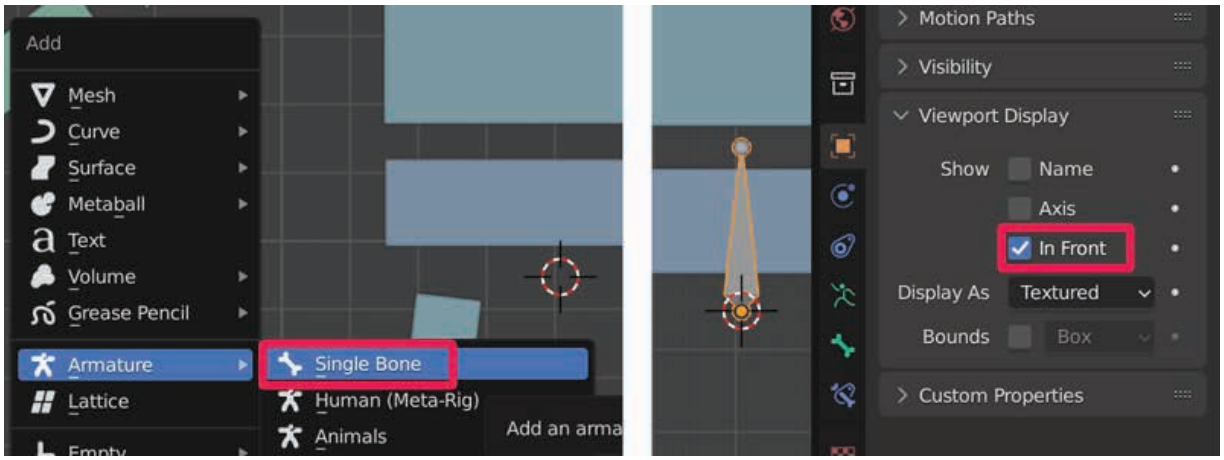
It may take some trial and error to find the right axes for each limb. Of course some joins need more than one axis of rotation, like the neck.



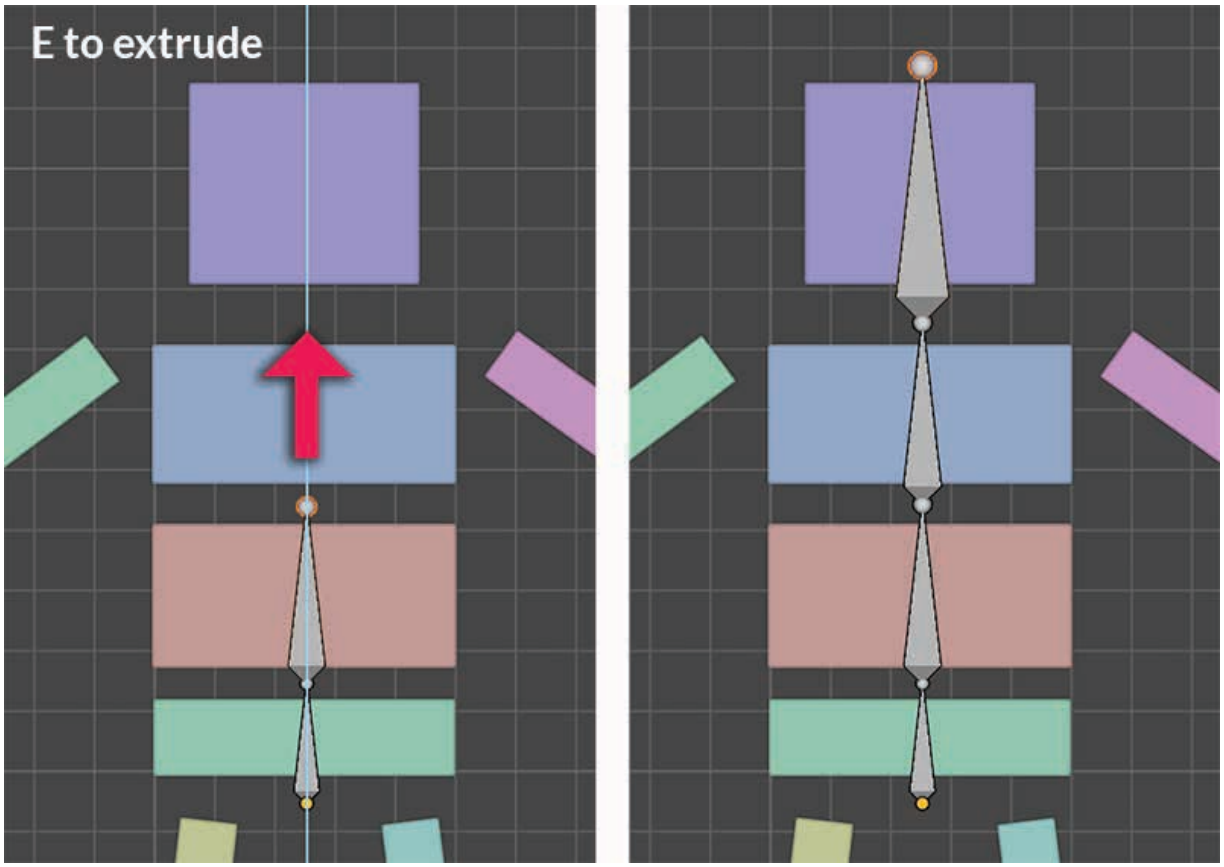
To save you all the trouble of figuring out the right values for each Constraint, you can download the ragdoll Blend file here to skip all this work.

Ragdoll Rigging part 3 - Connecting it to an Armature

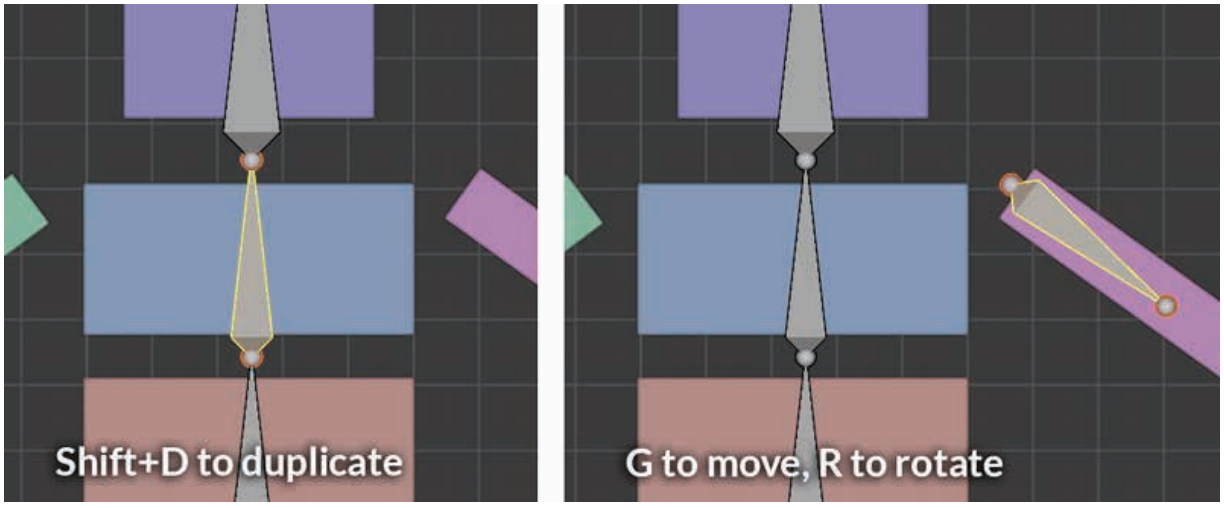
So far we've connected some boxes. These are the "hitboxes" that will interact as Rigid Body objects in simulations later. But to use the ragdoll to control a character, we also need some bones that we can pose. First, **press Shift+A** and add an Armature > **Single Bone**. You can set its **Viewport Display** to **In Front** so it's easier to see. Scale it down.



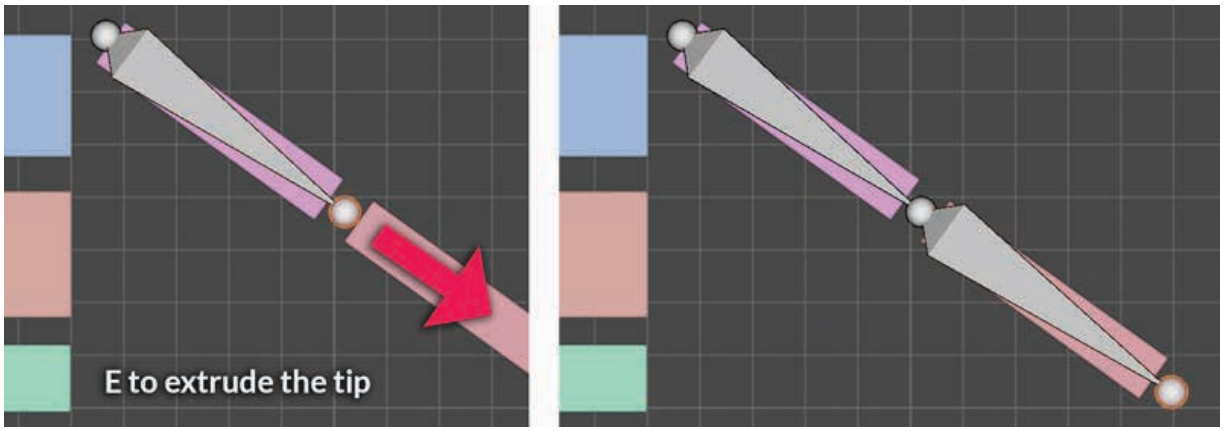
In Edit Mode, select the **Tip of the bone** and **press E** to extrude it. **Press Z** to constrain the extrusion to the Z-axis. Do this a couple more times, until every box of the body and the head has a bone.



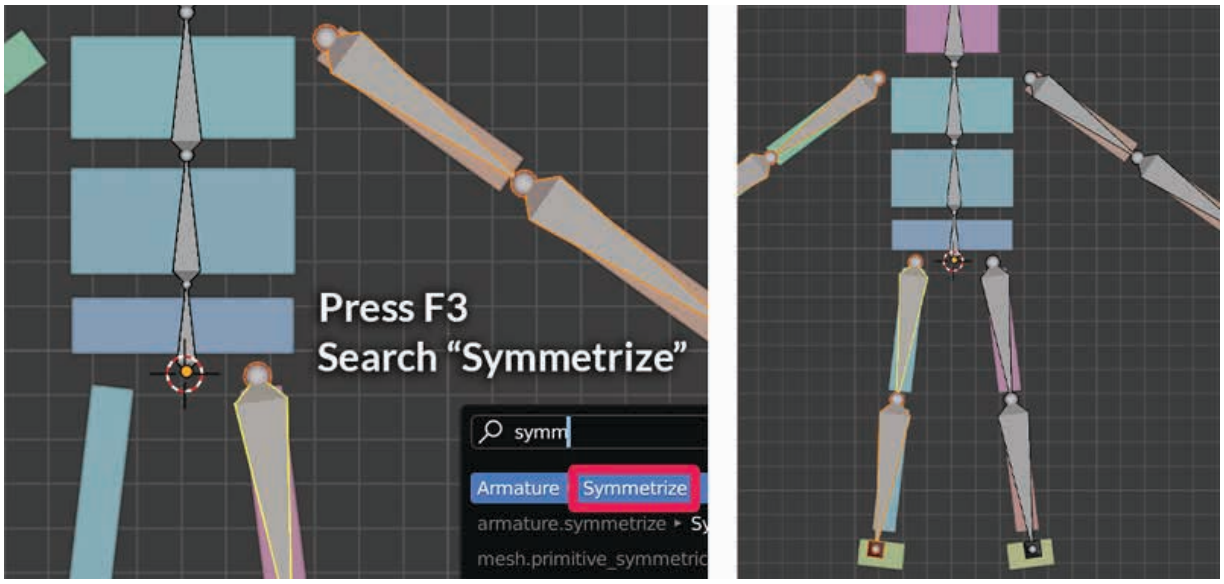
Still in Edit Mode, select a bone and **press Shift+D** to duplicate it. Then move it and rotate it to an arm.



Select the tip of the duplicated bone and move it by **pressing G**, to fit that box. Then extrude the tip by **pressing E**, and make it fit the next box.

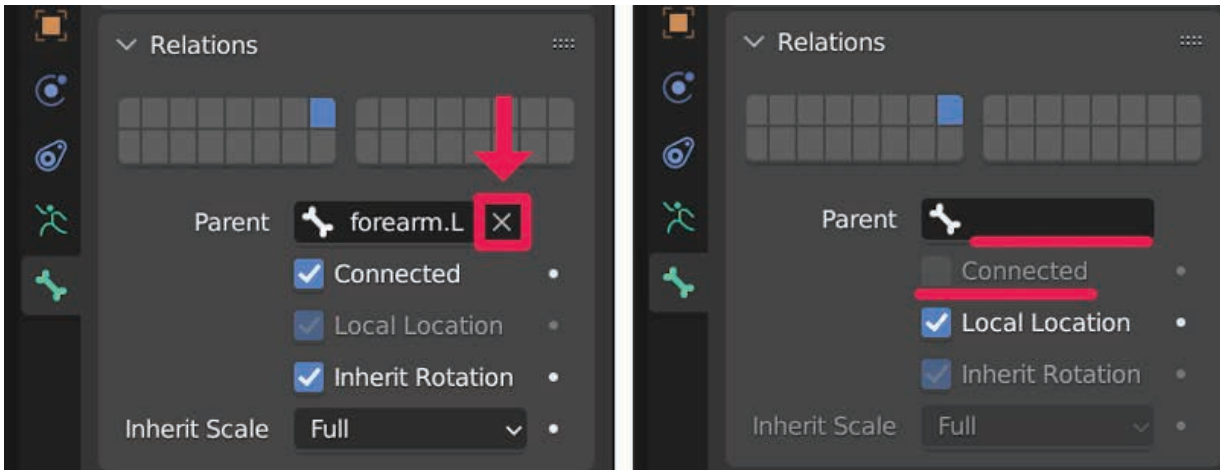


Once you've done this for the arm and leg on one side, you can mirror **selected** bones to the other side by **pressing F3** and searching for "**Symmetrize**". Make sure you've selected the bones you want to mirror or it won't work! This will add the same bones to the other side.

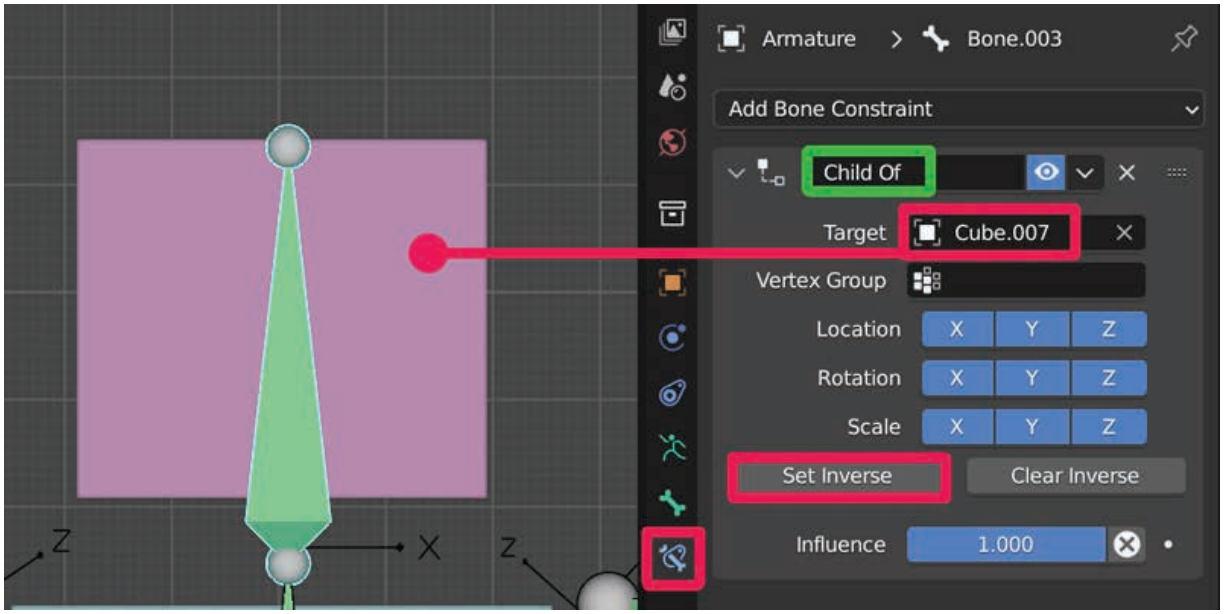


Now we are going to connect our Ragdoll to this Armature. First we need to remove the parent relationship from each bone. Select all the bones in Edit Mode, and **click on the X** next to the Parent field of the bone relationship, **while holding Alt.**

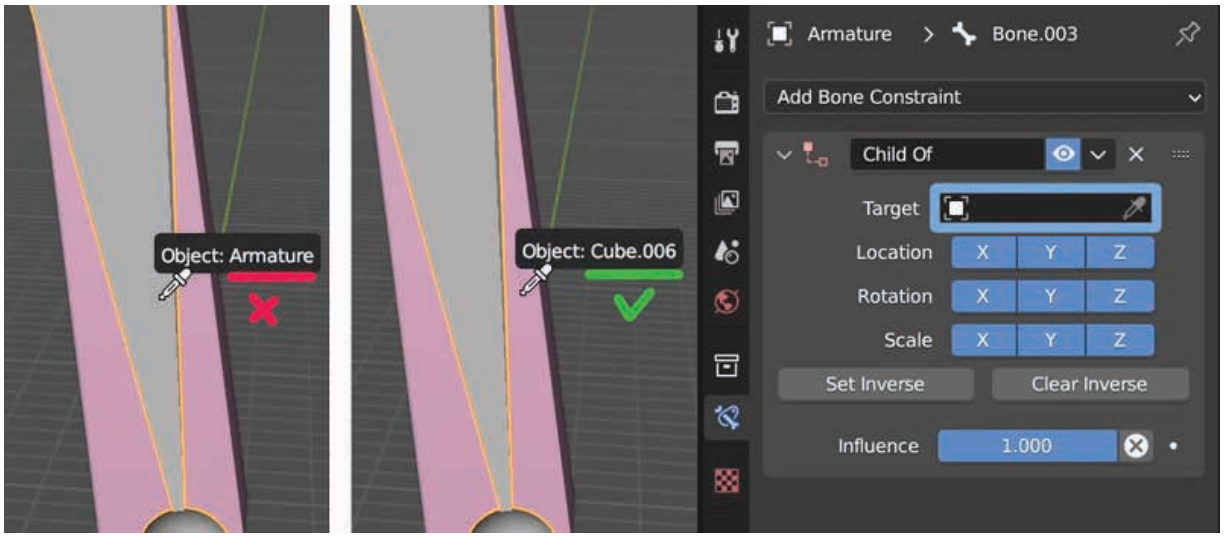
This removes the parenting on all the bones. *Note, that this only works when "Emulate 3 button mouse" is disabled in Preferences. This "holding Alt trick" works for a lot of checkboxes in Blender.*



Next we go to Pose Mode. Select a bone and add a "**Child Of**" bone constraint. Select the cube that goes with that bone as the **Target**. Click on **Set Inverse** if the bone jumps away.



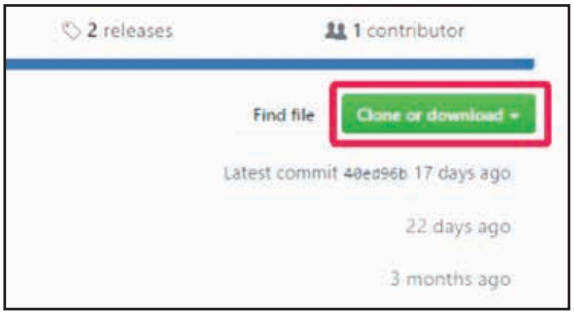
When selecting the Target, be careful that you select the hitbox, and not the Armature. Zooming in helps. Add **Bone Constraints** for all the bones.

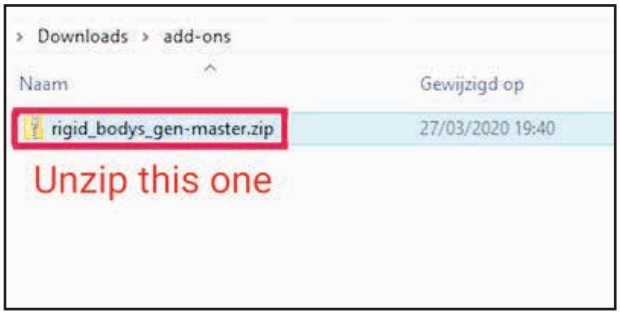


Once this is done, the armature is parented to the ragdoll. When you **press the Spacebar** from the beginning of the Timeline, it will fall down. Download the blend file here.

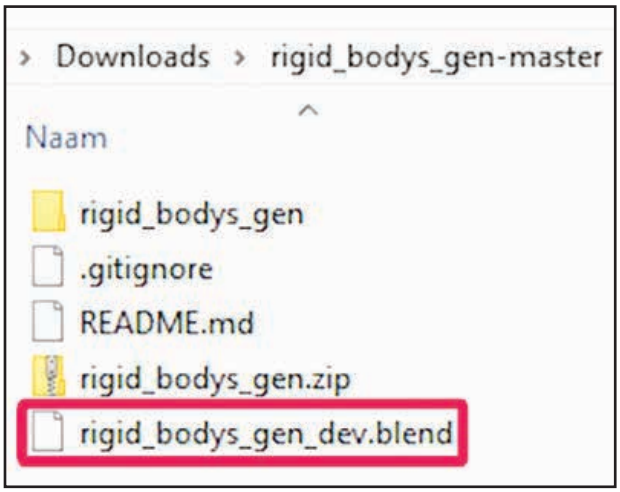
Install and use Rigid Bodys Generator add-on

To install this add-on, download the zip from Github and unzip that one, because there's a zip file inside the zip file.

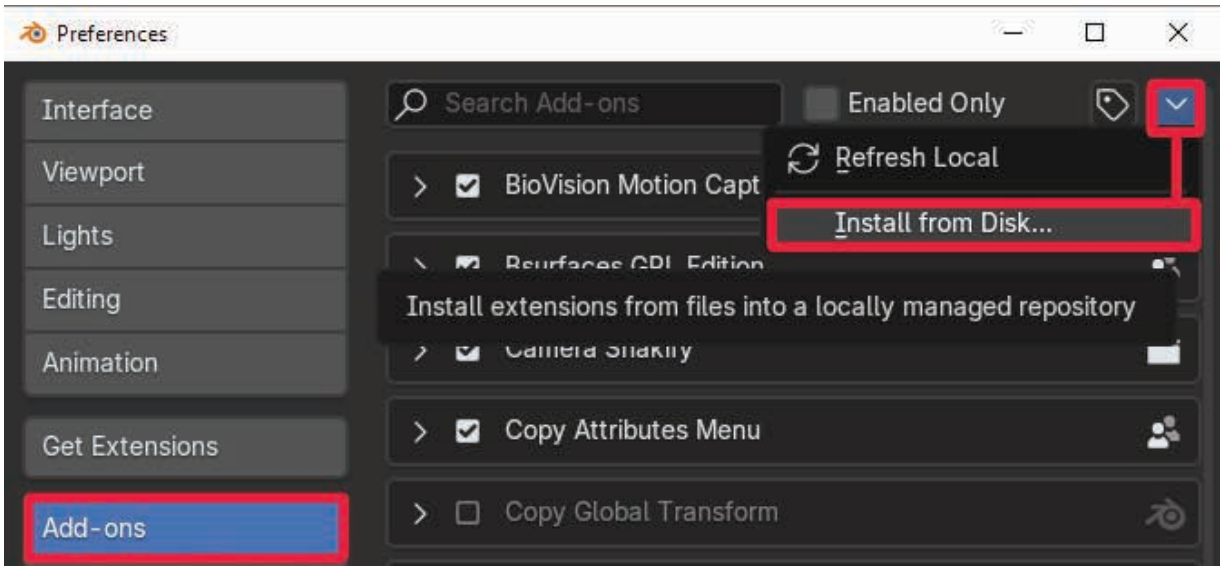




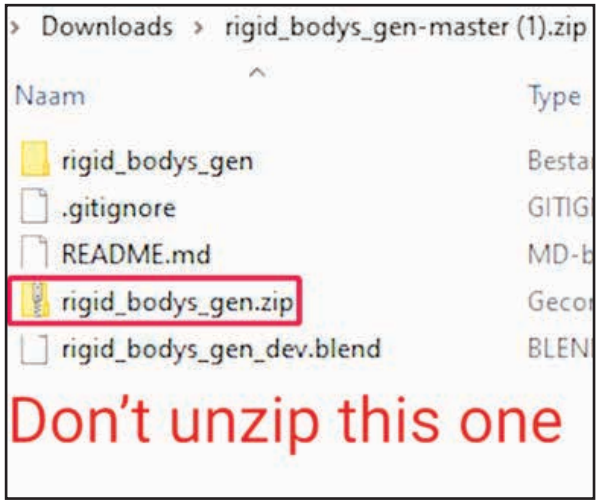
It also comes with this demo Blend file.

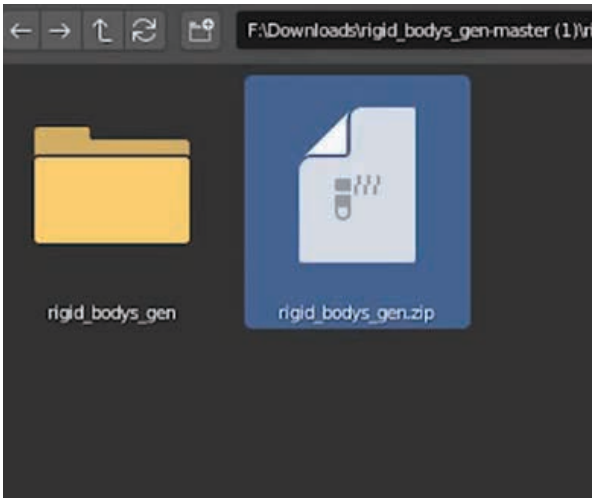


In Preferences, install the (not unzipped!) zip file by clicking on “Install from Disk”.

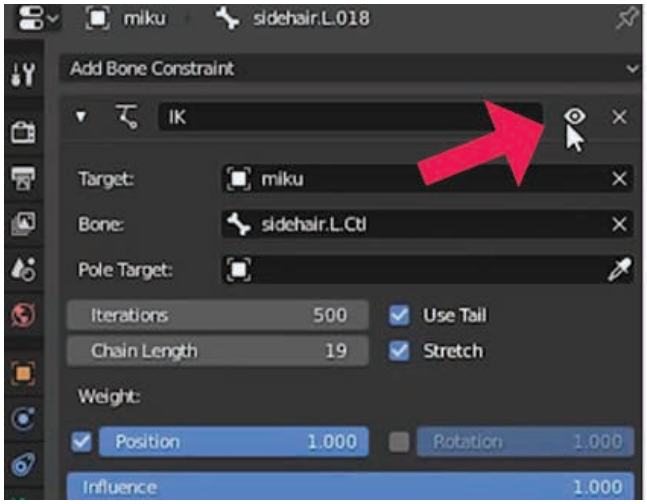


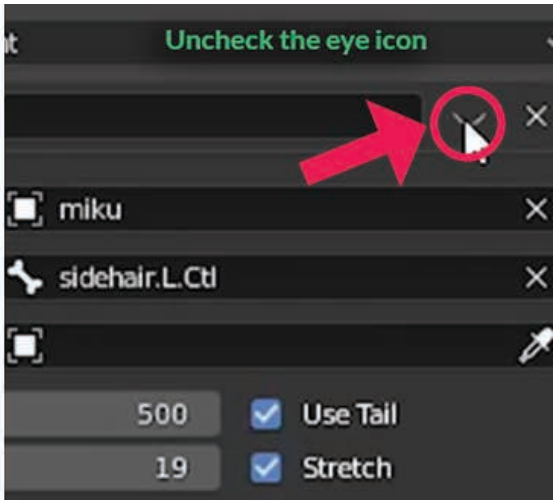
Choose the file **rigid_bodys_gen.zip**, the one that was inside the unzipped download.



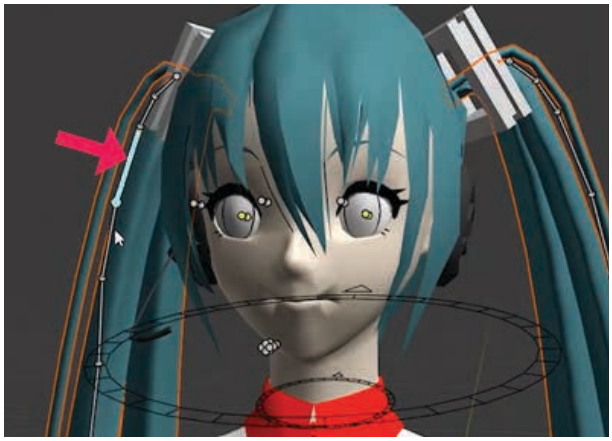


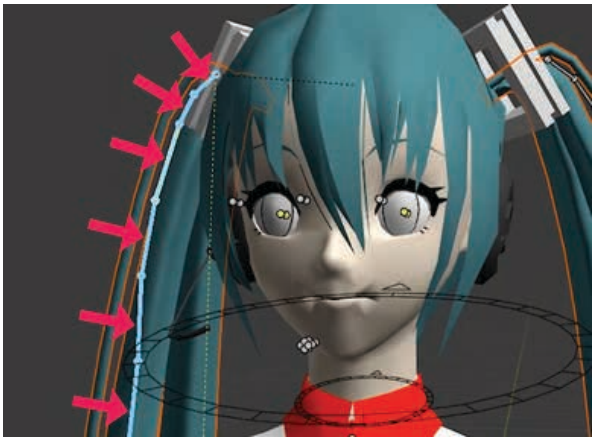
In **Pose mode**, select the bones you want to use and make sure if any of them have an IK bone constraint that you disable that for now.





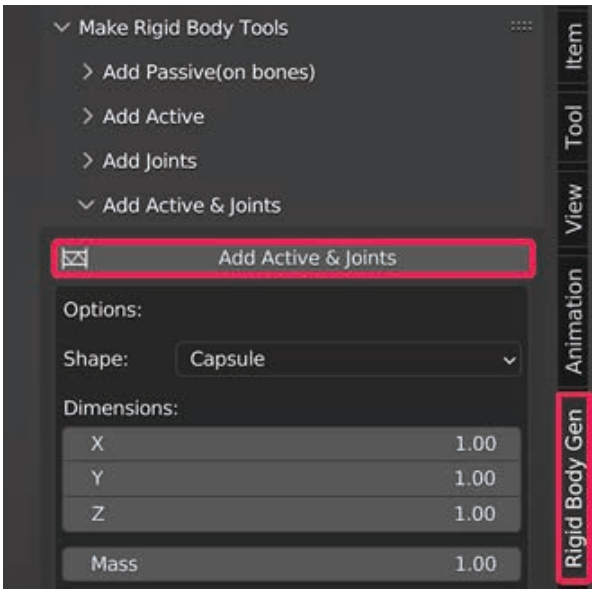
Pro tip - if you want to select all the bones in a chain instead of selecting them one by one, just select one and then **press L**.



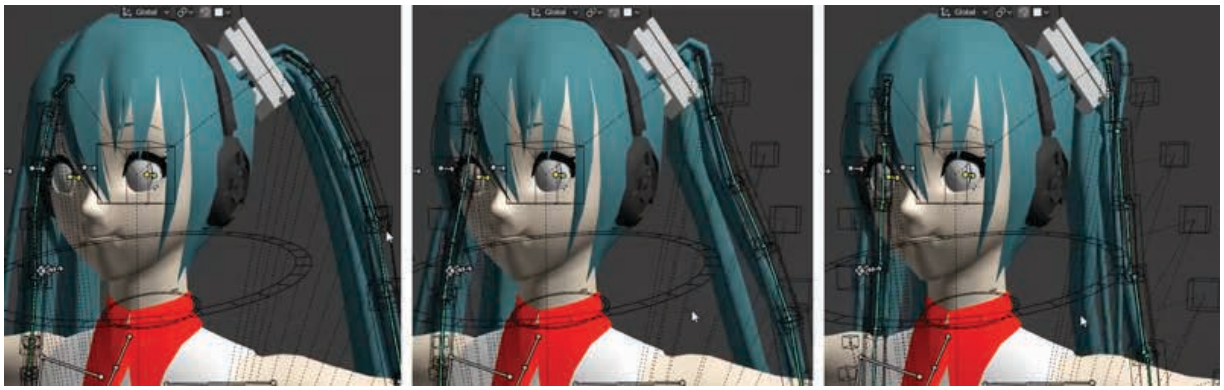


In the Option panel, go to the **Make Rigid Body** tools panel and click on **Add Active & Joints**.

Then wait a little bit as it sets up all the constraints for you.



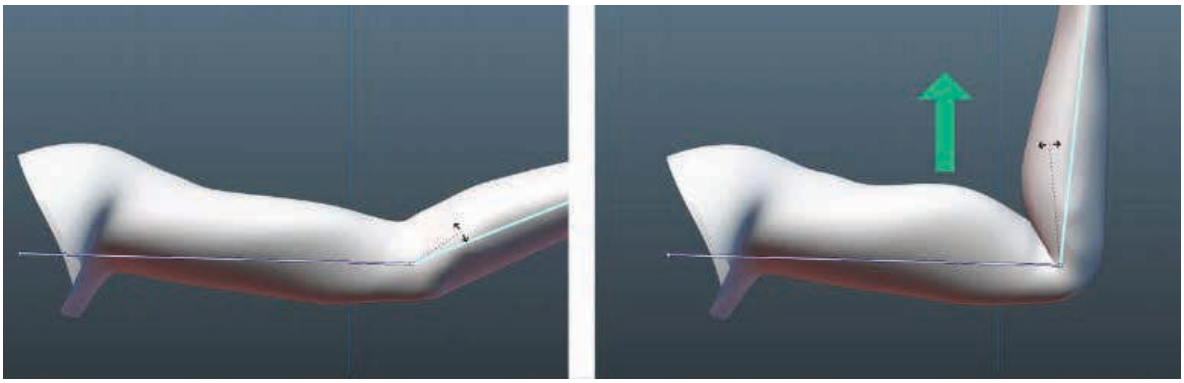
Now if you press Spacebar to play the simulation the bones will react to gravity like rigid bodies.



Automatic Muscle Deformations

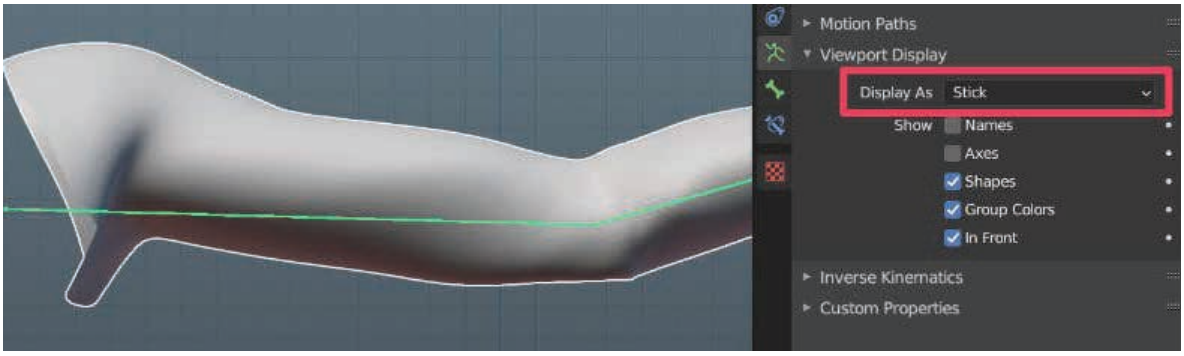
For character animation, it's important to pay attention to how the body deforms when moving, for example because of muscles that contract, which creates kind of "bulges" under the skin (clearly visible with the bicep muscle in your arm, for example).

One simple way to simulate the effect of muscle deformations is by using ShapeKeys. These can then be "automated" by using drivers, so that the deformations are controlled by the movement of the bones.

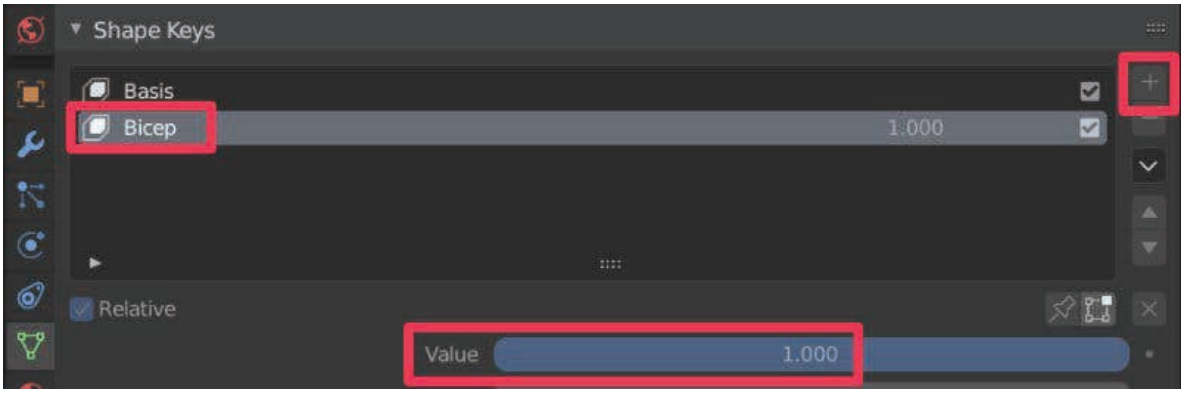


For this example, I'm using this arm model made by Blendswap user Turboyoyo.

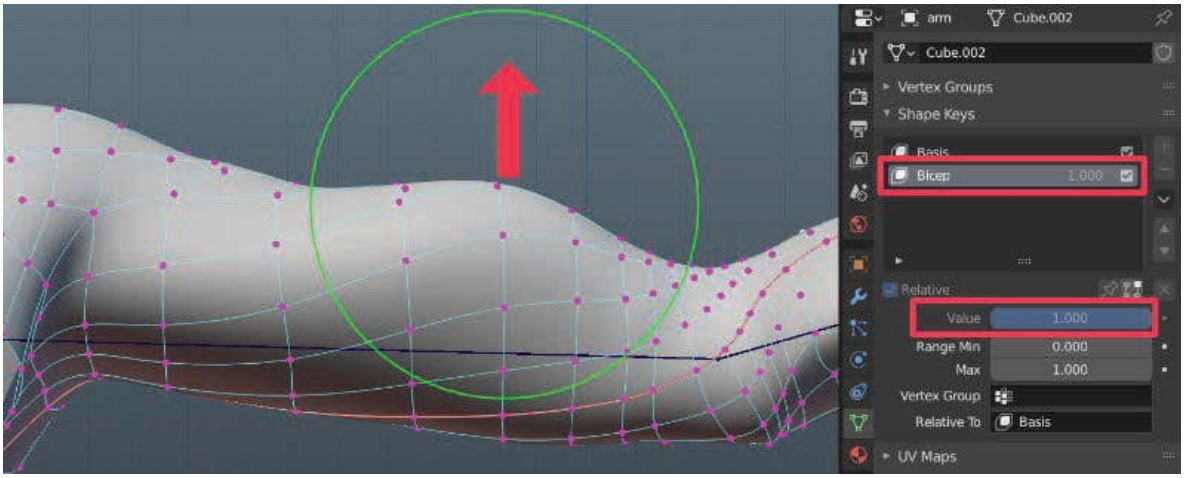
I've set the Armature to display as **sticks**, so it's easier to see the arm geometry. Once you understand the principles of using drivers for muscle deformation, you can use them on different parts of the body, not just the arm. It's a powerful tool for achieving better realism with character animation. And you only need to set it up once, after that it works forever!



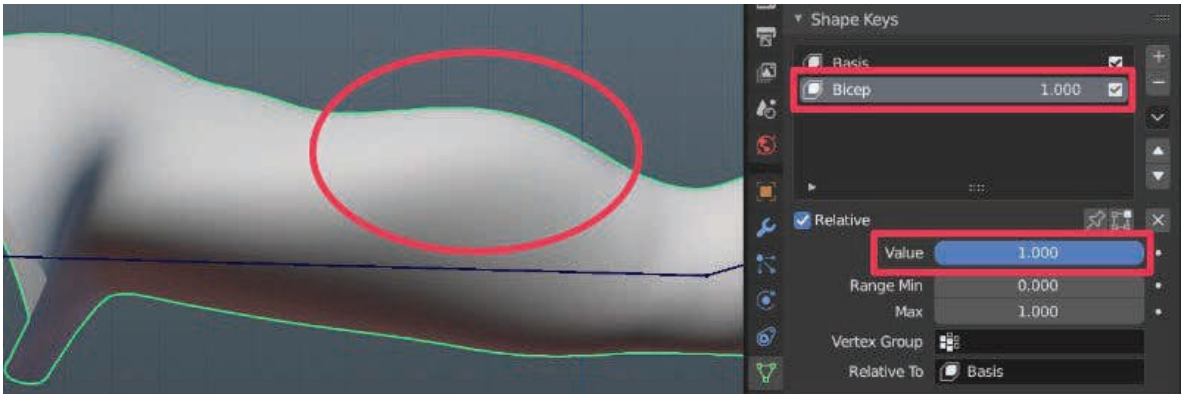
Create a Basis Shape Key and another Shape Key for the deformation by **clicking on the + icon**. This creates a new Shape Key called “Key 1” by default. You can rename it to “Bicep” (in this example) to make it more clear what that Shape Key does.



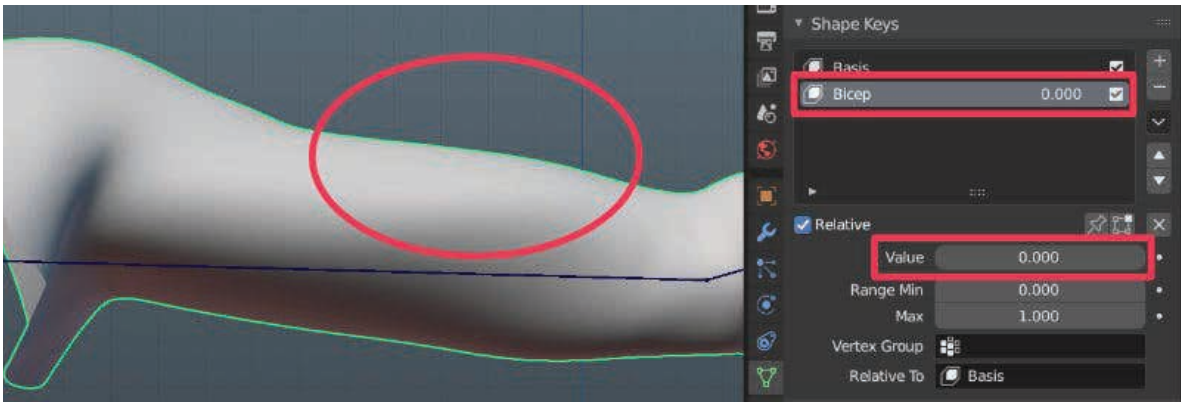
Set the **Value** to 1 and create the deformation of the mesh in Edit Mode. **Turn on Proportional Editing** by pressing o and select the top vertex of the bicep area, then move it up and adjust the proportional falloff circle by scrolling the mouse wheel up or down.



So now when the Value of the Bicep Shape Key is 1, the bicep is inflated.

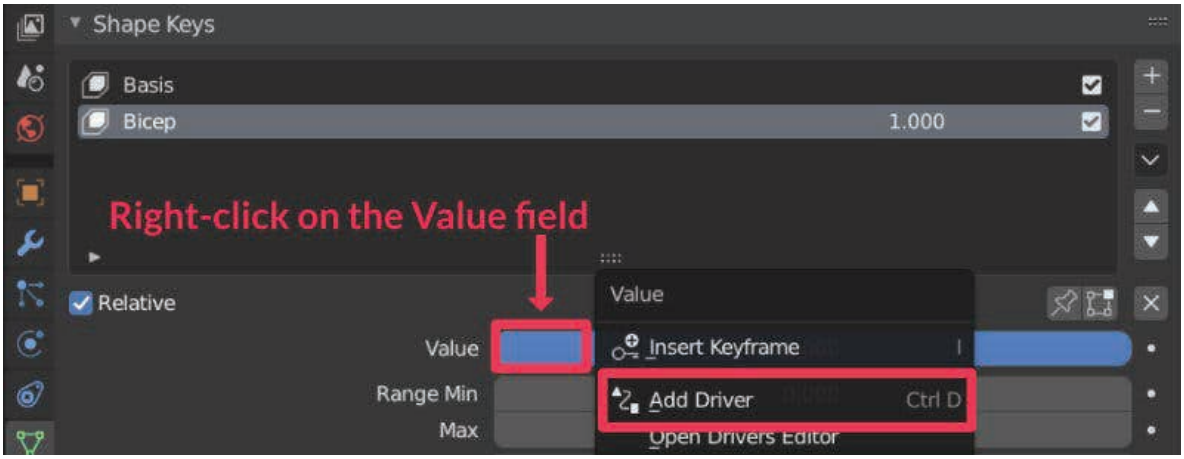


And when you slide it back to 0, the bicep gets smaller.



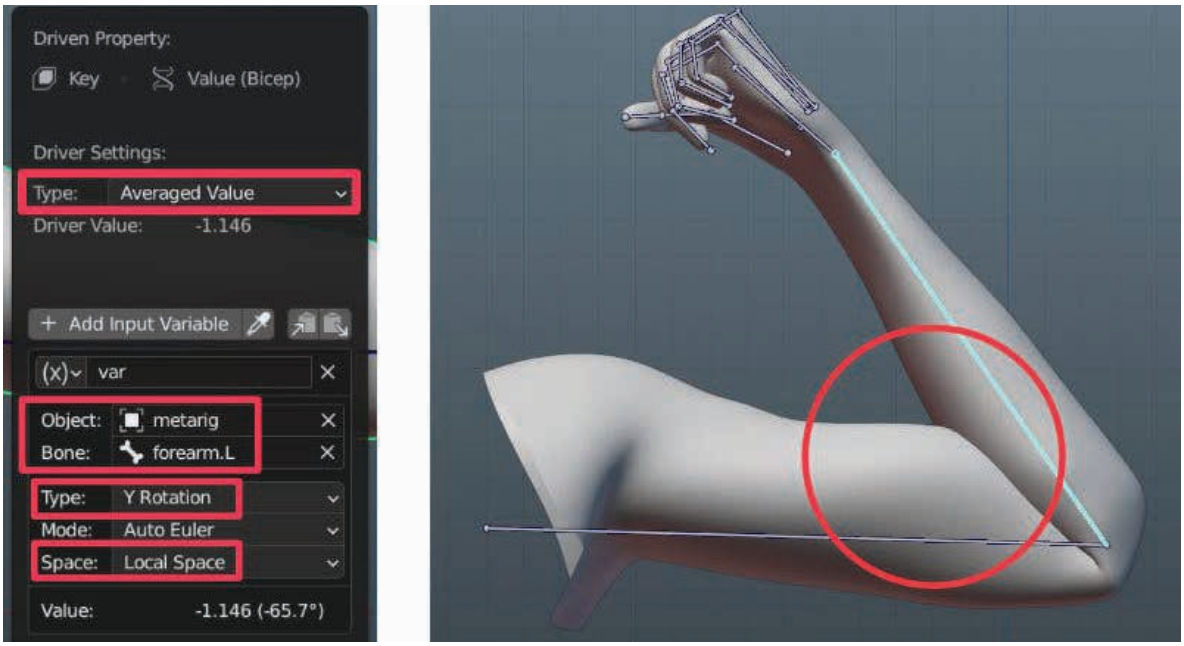
That's good so far, but of course we don't want to have to manually animate that with keyframes any time the character moves an arm. It's better to automate this with a Driver. And it's easier than you might think!

Right-click on the Shape Key **Value** slider and choose **Add Driver**.



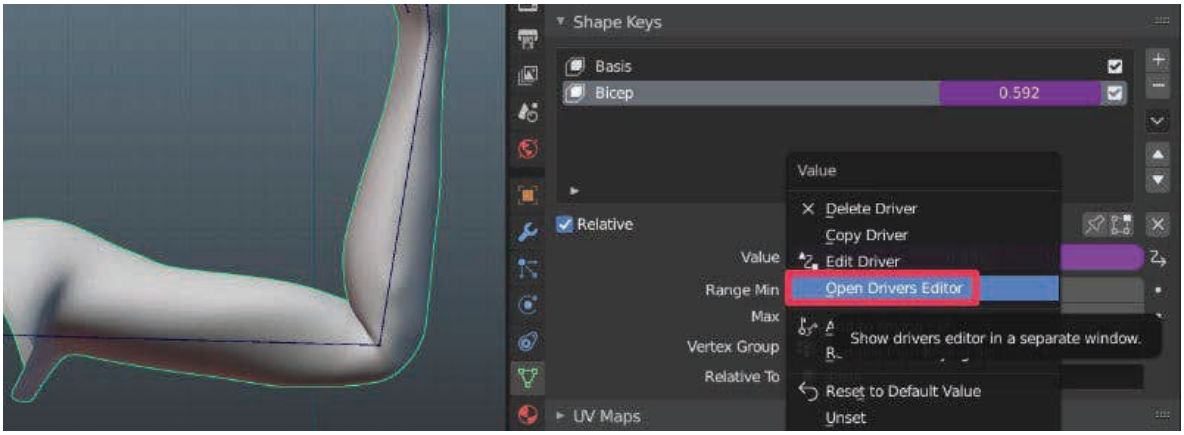
Now a menu pops up that allows us to enter some specific driver settings. Enter these settings:

Type: **Averaged Value**. Object: **metarig** (this is an armature created by the Rigify plug-in so it's named "metarig" by default). Bone: **forearm.L**. Check in advance which axis the bone rotates on for the deformation. **Y Rotation** works in this case, in **Local Space**.



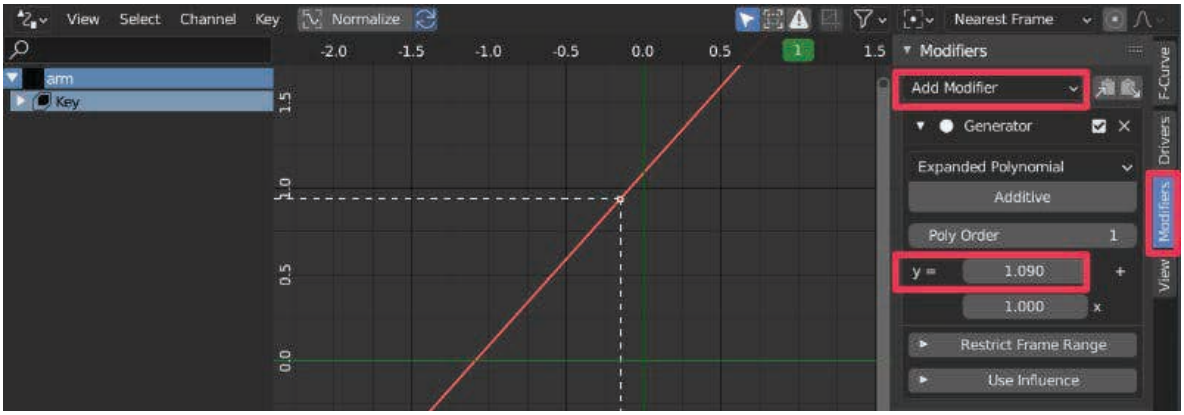
In case the menu disappears because you accidentally clicked somewhere, don't worry - you can get it back by **right-clicking on the Value field** and choosing **Edit Driver**. If you switch to **Pose Mode** and rotate the upper arm, you'll notice that the bicep doesn't get inflated until too late (see circled area in the picture above right). We need to do some fine-tuning to get it to work as we want.

Pose the arm in the pose where you want the bicep to be fully inflated. **Right-click on the Shape Key Value slider** and choose **Open Drivers Editor**.

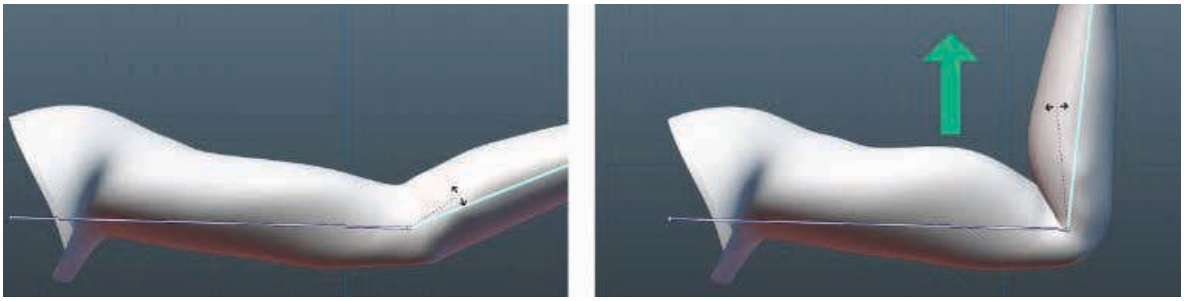


This opens another pop-up window. Go to the **Modifiers** tab of the Drivers Editor and make sure the graph is selected (you can **press A** to select all with the mouse cursor in the graph area). Choose **Generator** from the Add Modifiers drop-down list.

These values control the timing and strength of the Driver. **Holding Shift, drag the Y value** until the bicep in the viewport is fully inflated. The reason you should hold Shift is because it gives finer control over changing the value.



Now close this window. Test the arm again in pose mode. Now the bicep "automatically" gets inflated when the upper arm rotates enough.

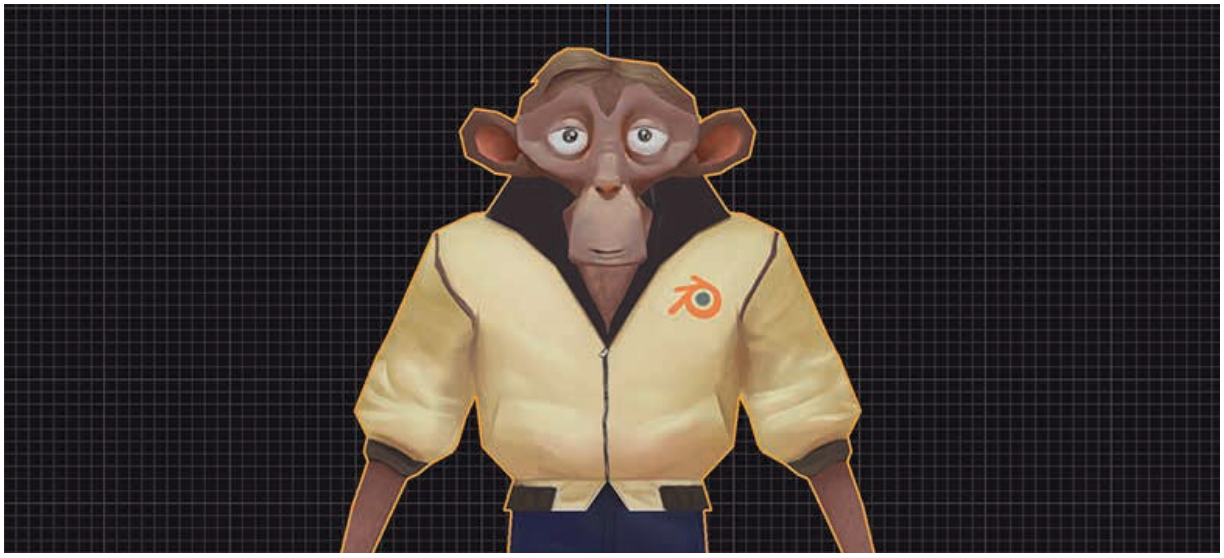


In case it doesn't work like it did in the Drivers Editor, open the Drivers Editor again by right-clicking on the **Value** field of the Shape Key and choosing "**Open Drivers Editor**", then click on **Update Dependencies** in the Drivers tab. Normally this is not necessary, however.

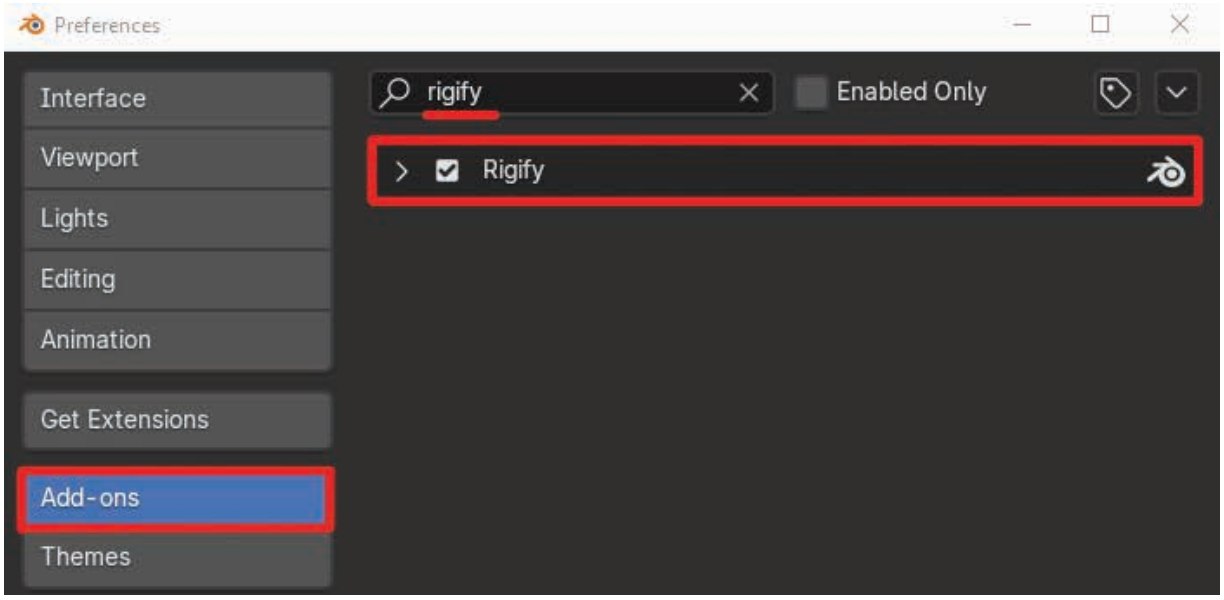
And that's all there is to it! Another area where this comes in handy is for creating subtle adjustments to clothes on a character when they move.

Rigging a simple character with Rigify

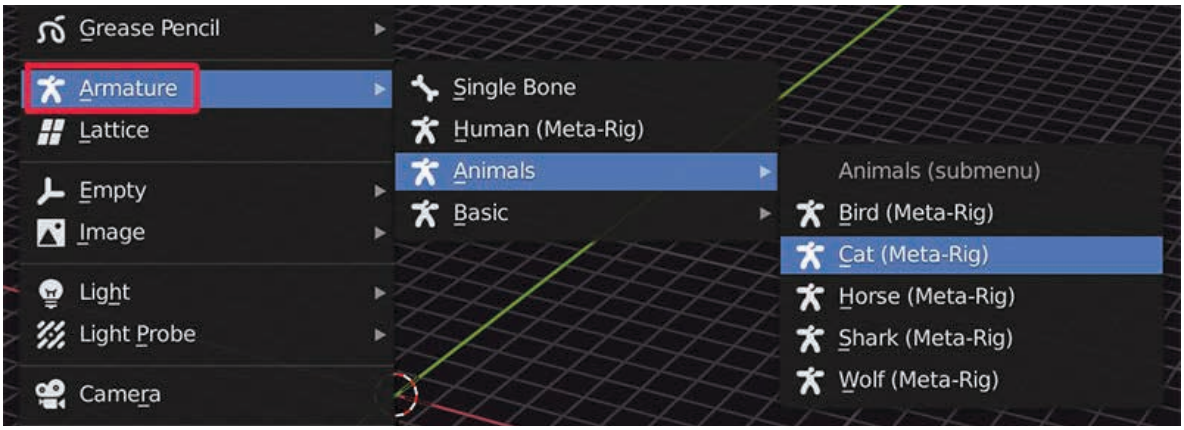
In this tutorial we're going to look at rigging a low-poly character using **Rigify**, the rigging add-on that comes with Blender by default. We'll be using this funny low-poly version of the famous "Suzanne" monkey that I purchased from CGTrader.



First of all we need to enable the Rigify add-on. Go to **Edit > Preferences**. Then click on Add-ons and type "**Rigify**" in the search field until it comes up. **Check the box** next to Rigging: Rigify to enable to add-on.



When you **press Shift+A** and go to **Armature**, you'll see that some extra options have been added. Besides a detailed human rig with fingers and even facial bones, there's a Basic human rig and even animal rigs such as a bird, cat and horse rig.

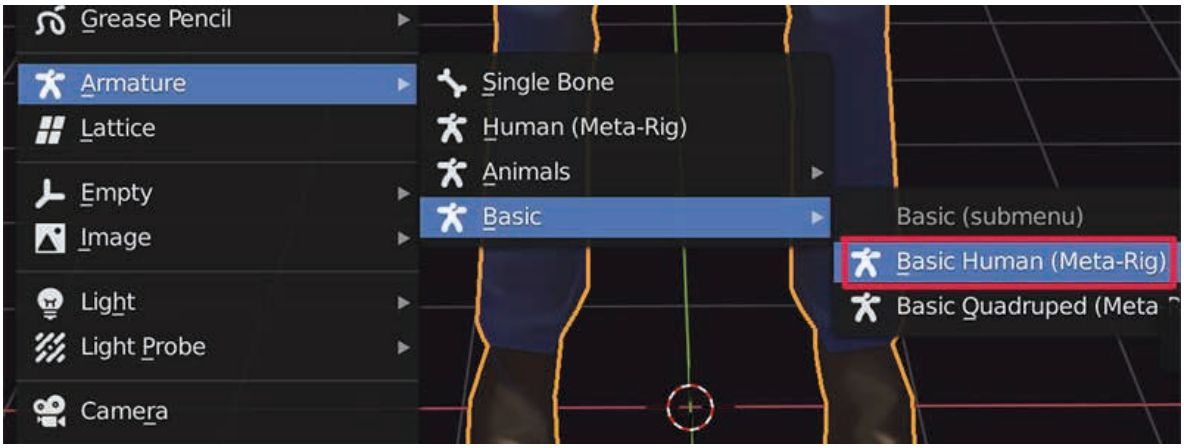


Make sure that your character model has **Scale: 1** and **Rotation: 0°** on all 3 axis. Otherwise you will encounter problems later on. If this isn't the case, in Object Mode with the model selected **press Ctrl+A** and choose **Rotation and Scale** to apply the Rotation and Scale.

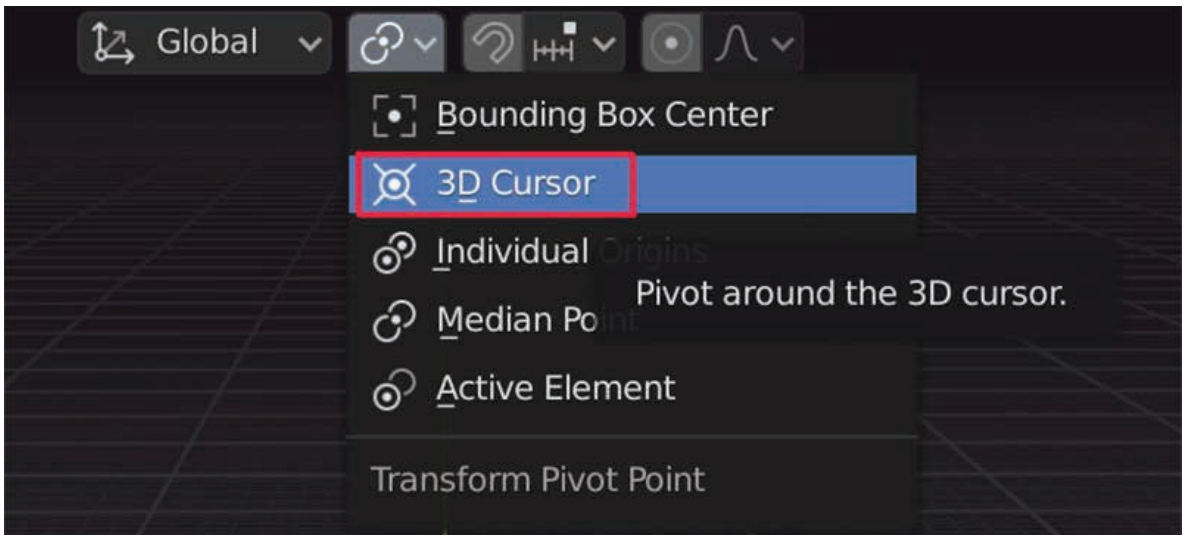
Make sure the **3D Cursor** is at the center of the world by **pressing Shift+S** and choosing **Cursor to World Origin**.

Press Shift+A and choose Armature > Basic > **Basic Human (Meta-Rig)**. This basic rig doesn't have fingers or facial expression bones, making it perfect for a low-poly character like this.

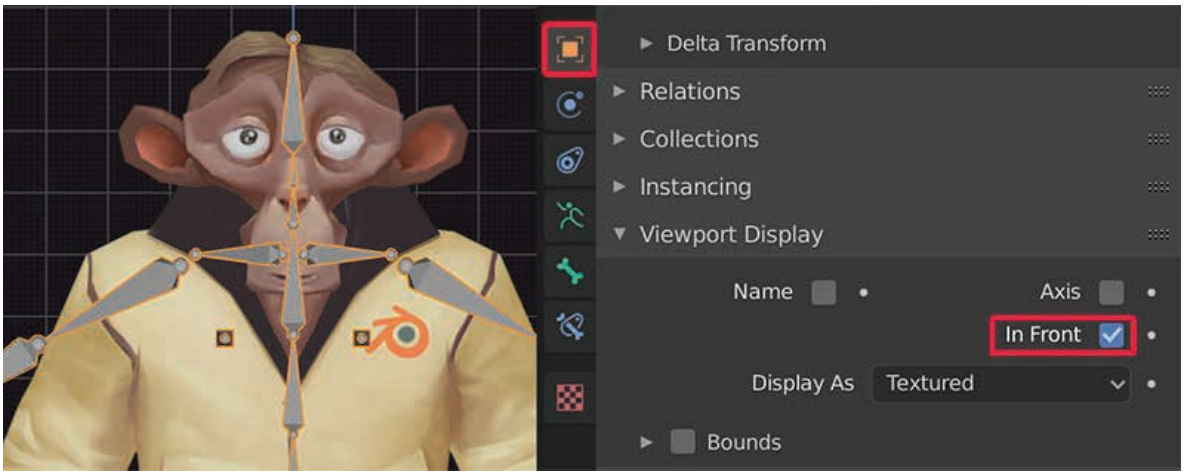
The rig you get is actually called the **MetaRig**, and it is used only temporarily to tell Rigify the proportions of our character so that it can create Custom Control Bones.



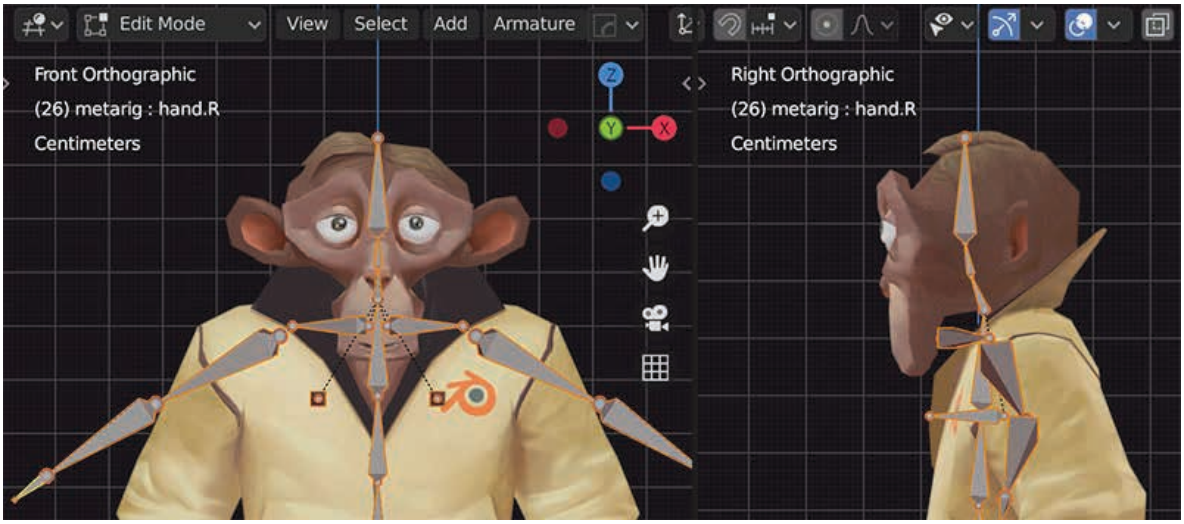
Set Transform Pivot Point to **3D Cursor**.



Scale the MetaRig up or down in Edit Mode to fit your character. Get the scale as close as possible. To see what you are doing more clearly, turn on **In Front** in the Armature options so you can always see the MetaRig even when it's inside the character model.



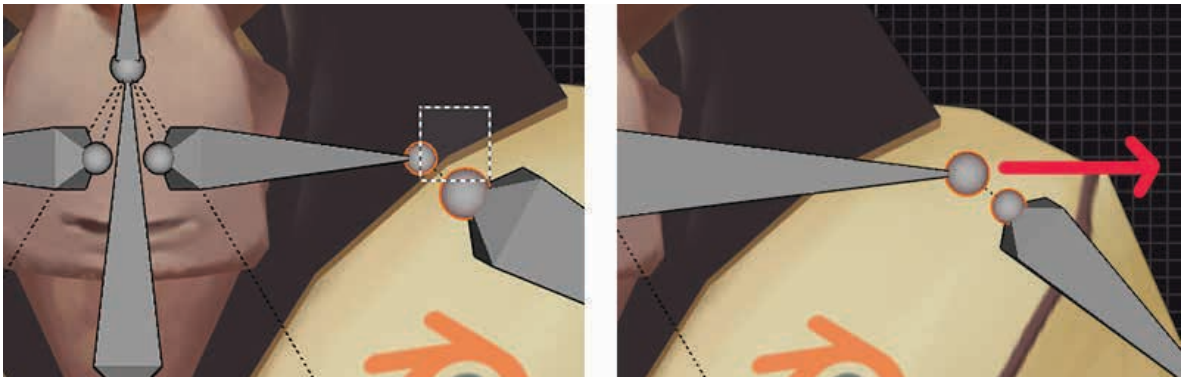
I suggest that you open another 3D viewport and set one to **Front orthographic** view and the other to **Right orthographic** view. This way it's easier to align the bones by looking at it from both angles.



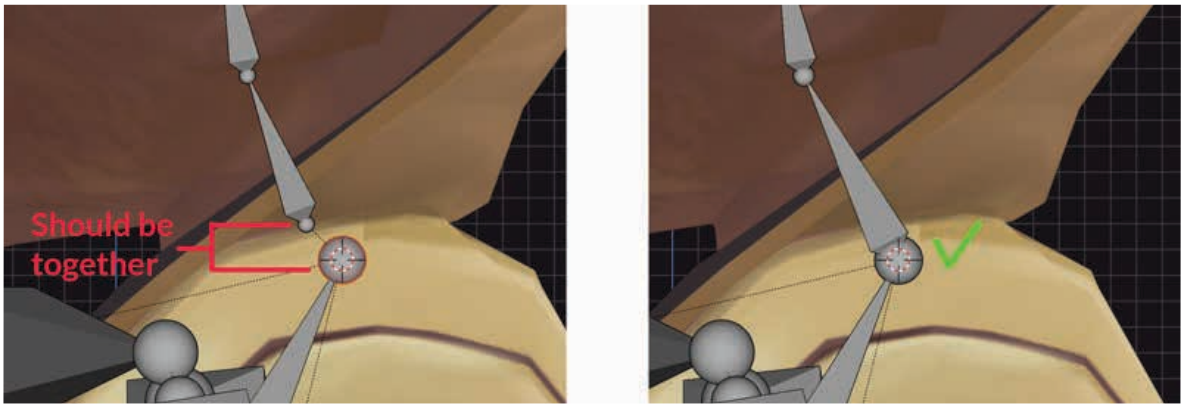
Now you move the bones in Edit Mode on one side of the character in place by selecting and moving the **Tips** and **Roots** (the round parts) of the bones by **left-clicking to select** and **pressing G to move**.

For more precision you can align them to the axis you are moving them on by **pressing G+X** to move on the X-axis and **G+Y** to move on the Y-axis. You can also select the entire bone or a number of bones by dragging a box selection around them.

Be careful when moving bones that are supposed to be connected: **drag a box select** around any **Tip** and **Root** that are supposed to stay together but which can move independently such as the part where the neck and spine connect. Otherwise you might run into trouble later.

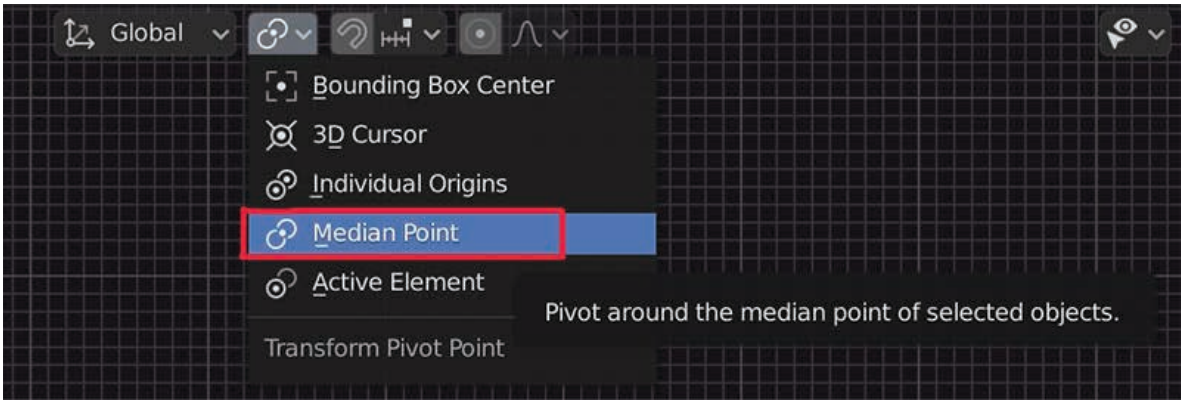


If you ever accidentally move these separately, select the **Tip** of one and **press Shift+S**, choose **Cursor to Selected**. Then select the **Root** of the other and **press Shift+S**, choose **Selection to Cursor**. With the Spine and Neck for example, Rigify will give an error if these bones do not align correctly.



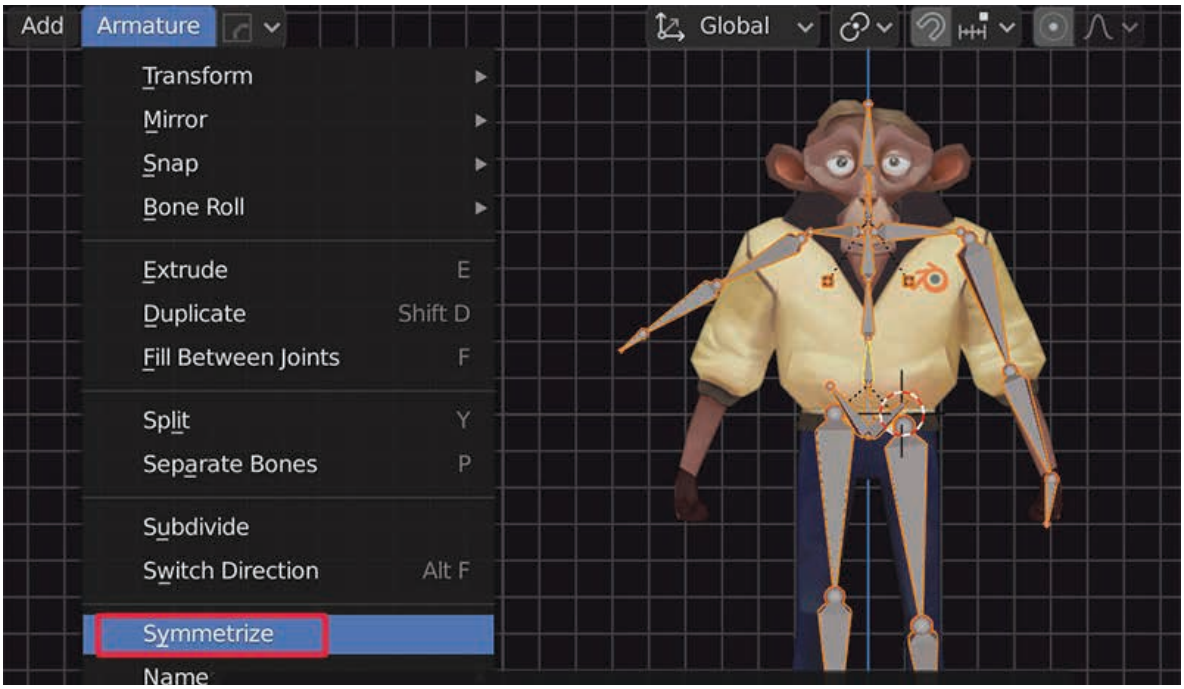
One trick to rotate several bones at once like the arm, is to select the Root of the top arm bone and **press Shift+S**, choose **Cursor to Selected**. Then, assuming you still have **Transform Pivot Point** set to **3D Cursor**, you can select the arm bones and rotate and scale them to match up with the character. You can do this trick for the legs as well.

This gets you about half-way there for the arms and legs. After this you still need to do some manual adjusting to get the position just right (remember to check both the **Front** and **Side** view). Set **Transform Pivot Point** back to **Median Point** when moving individual bone elements.

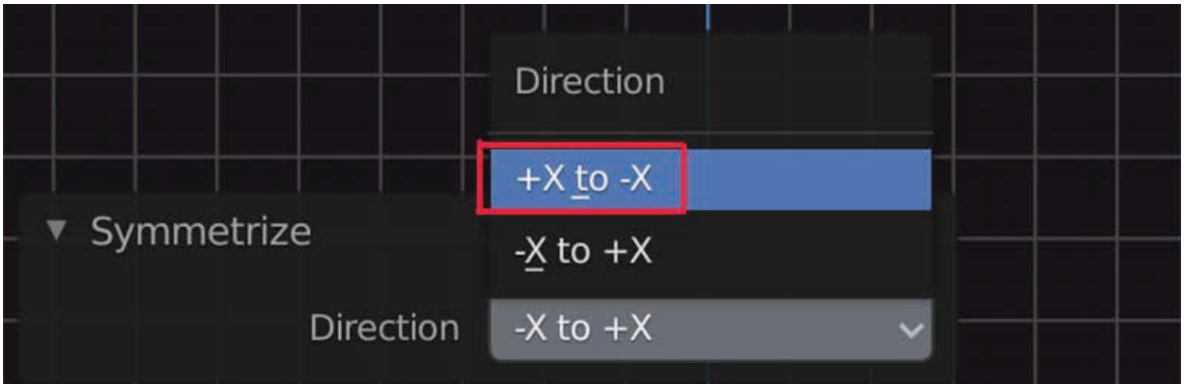


The rigging process with Rigify is actually quite simple, it's all a matter of selecting bones and moving them to fit inside your character model. Rigify takes care of the rest. You may not get it right the first time, but you can redo it until it works well. Each time you learn and get better at it.

With the bones lined up on one side of the MetaRig, select all bones by **pressing A** and go to **Armature > Symmetrize**.



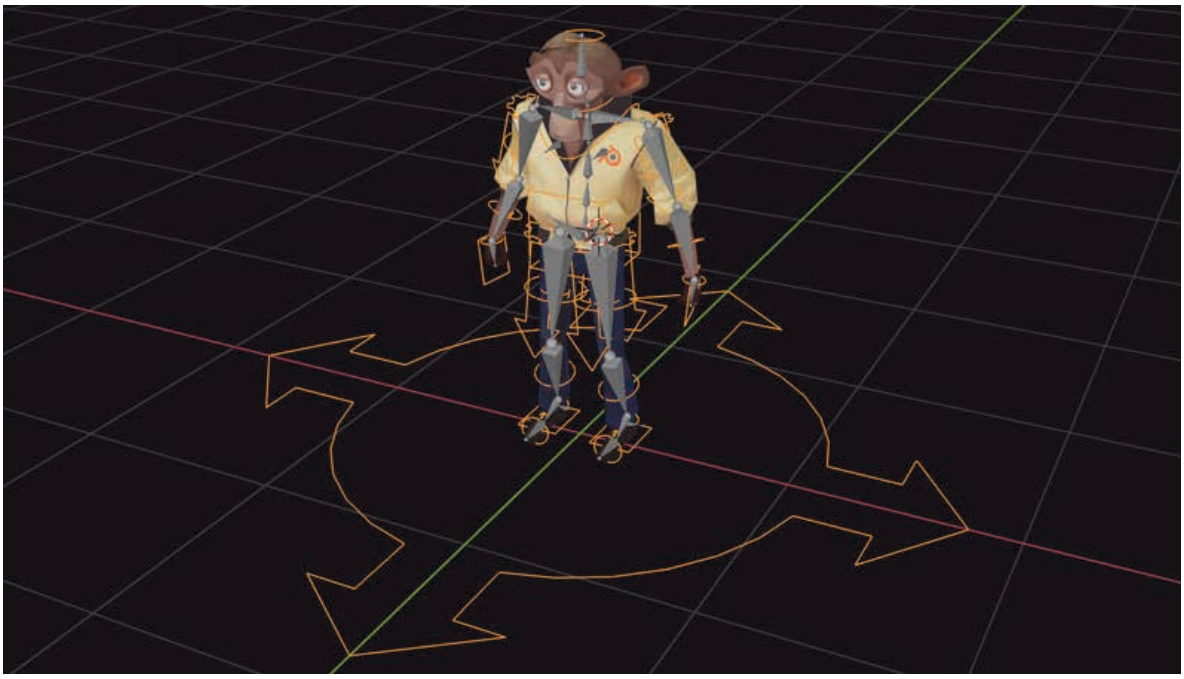
A menu appears in the bottom left of the viewport with an option to change the **Direction** of the Symmetrize action. In this case **+X to -X** is what I need since I worked on the +X side of the model.



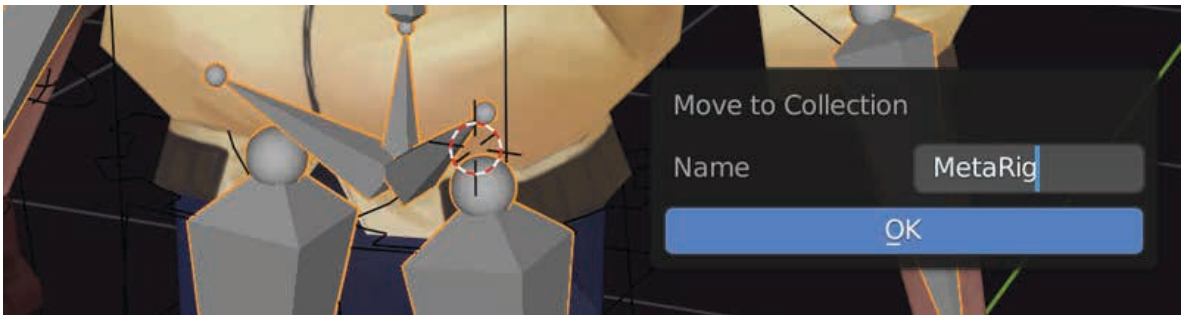
Now the MetaRig is symmetrical.

In the **Armature** tab, scroll down and click on **Generate Rig**. Wait for Rigify to create all the control shapes. Depending on your computer, this might take a couple of seconds or even minutes.

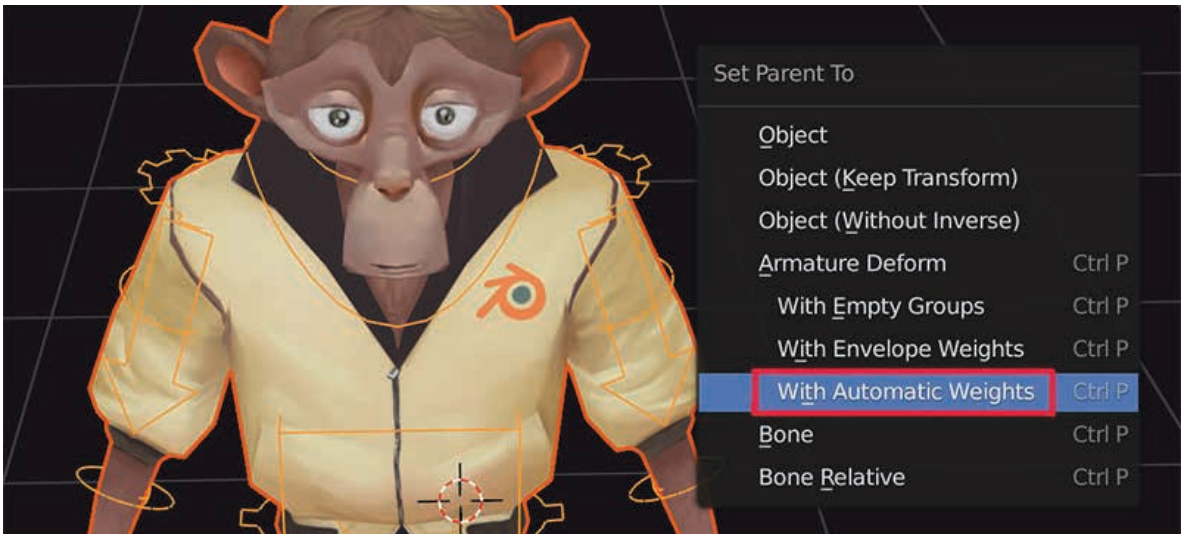
Once it's done you will see that a number of control shapes have been created.



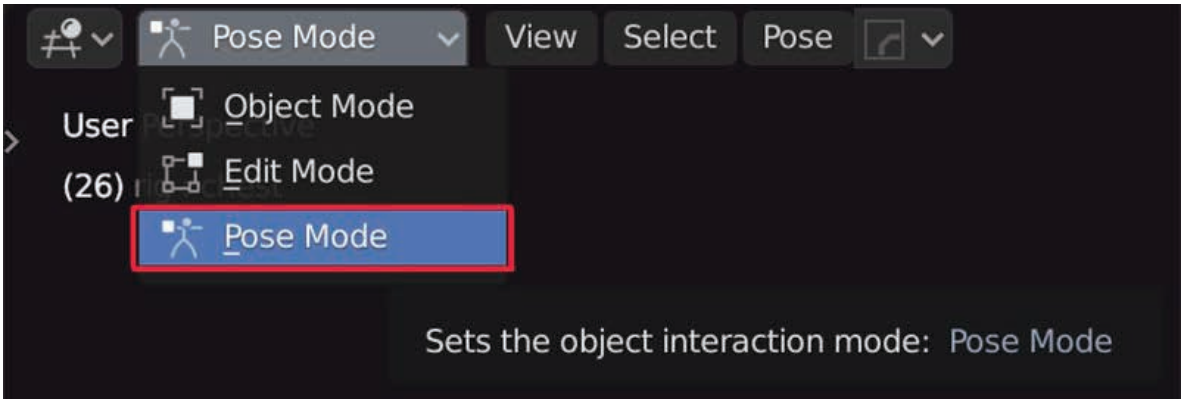
If you're feeling especially confident, you can delete the **MetaRig** now because its job is done. Or you can keep it in a Collection by **pressing m** and clicking on "**New Collection**", calling it "**MetaRig**" for clarity and then turning that Collections visibility off in the **Outliner**. You want to keep the MetaRig around so you can make adjustments later.



In Object Mode select the character and then **while holding Shift** select the control shapes (they will be yellow to indicate that they are the active object). **Press Ctrl+P** for the Parenting menu and choose Set Parent **"With Automatic Weights"**. This binds the character to the armature with an automatic process.



That's it! You've rigged your character. The colorful shapes are the **Custom Control Bones**. You can select them in **Pose Mode** and move and rotate them, to animate your character. To make animation you set keyframes for the poses on the Timeline. Note that you need to select the Control Bones to have Pose Mode as an option.



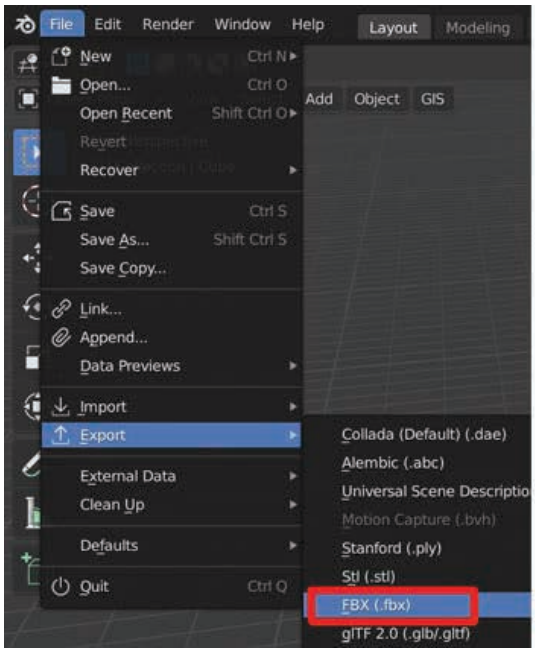
To reset a Custom Control Bone to its default position, **press Alt+G** or **Alt+R**. If you turn on **Auto Keying**, you don't even need to set keyframes manually, just pose the character at different points on the timeline and those poses will be recorded automatically.

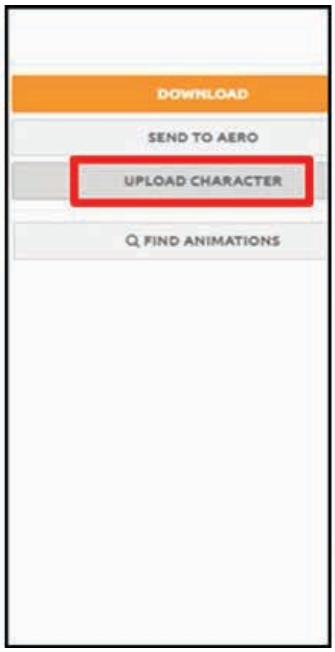
It's always a good idea to test your rig first before jumping into animation, to see if there are some parts that can be improved.

Improvements to rigging can be made by either **weight painting** manually, which can be tricky, or simply by unhiding the MetaRig collection and moving bones to where you think they might be better placed in **Edit Mode** before clicking **Generate Rig** again. In my experience this is the fastest way to improve a rig with Rigify.

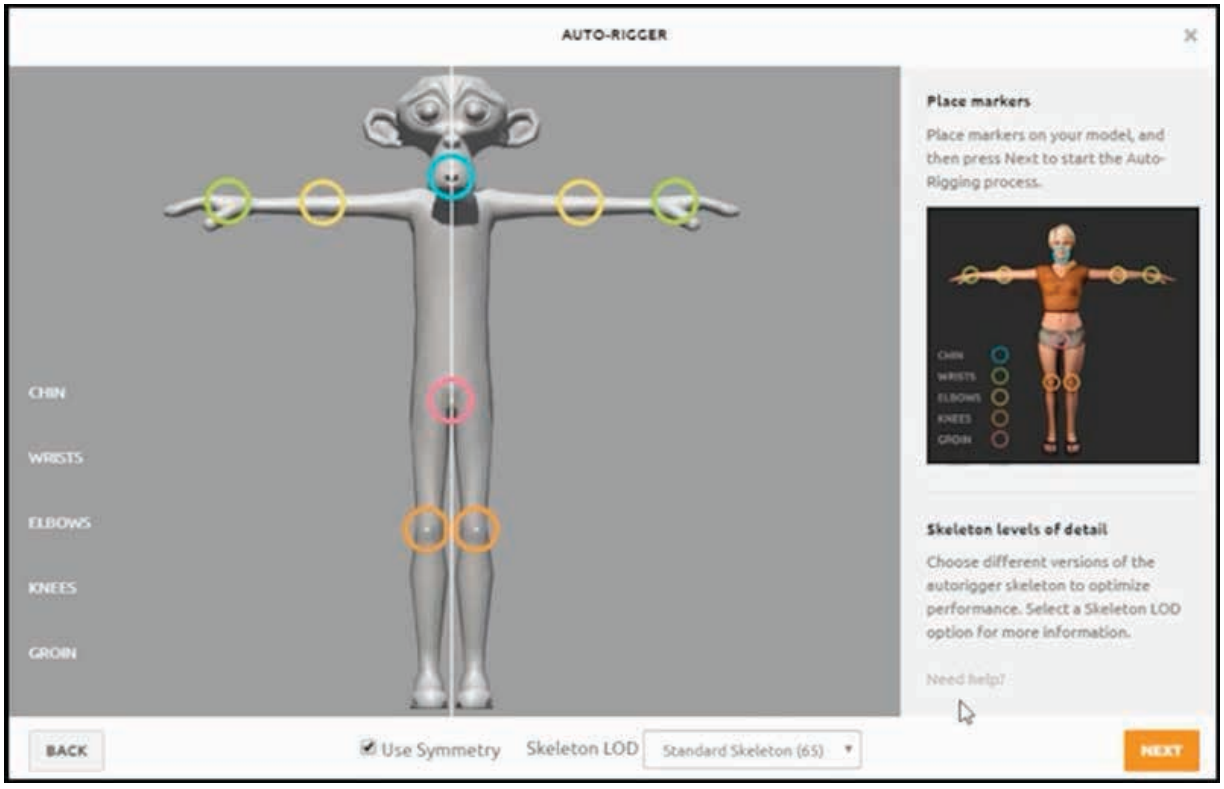
Auto-Rigging with Mixamo

Export your Blender character as an OBJ or **FBX file**. In Mixamo click on **Upload Character** and select your OBJ or FBX file. It will take a moment to load.





Make sure your character is centered and facing the camera. Click **Next**. Now you can place the colored circles as indicated in the picture on the right. When you are done, click on **Next**.

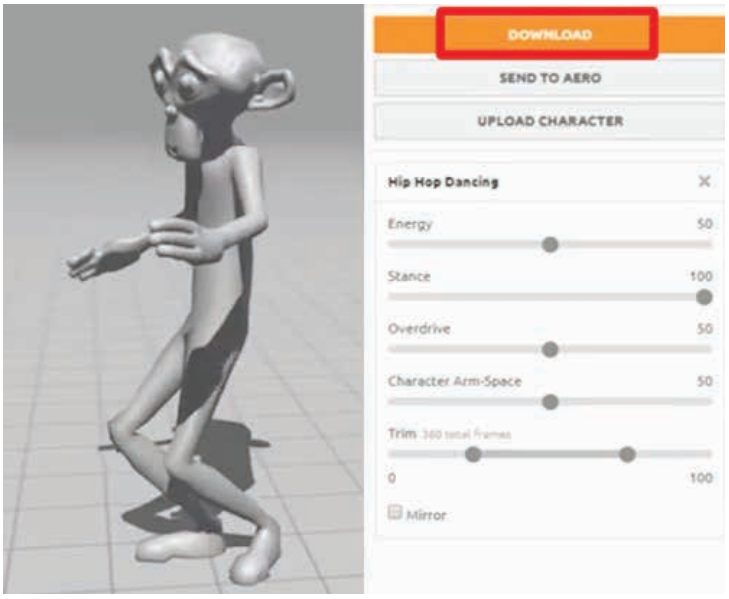


As you can see, it works pretty well.

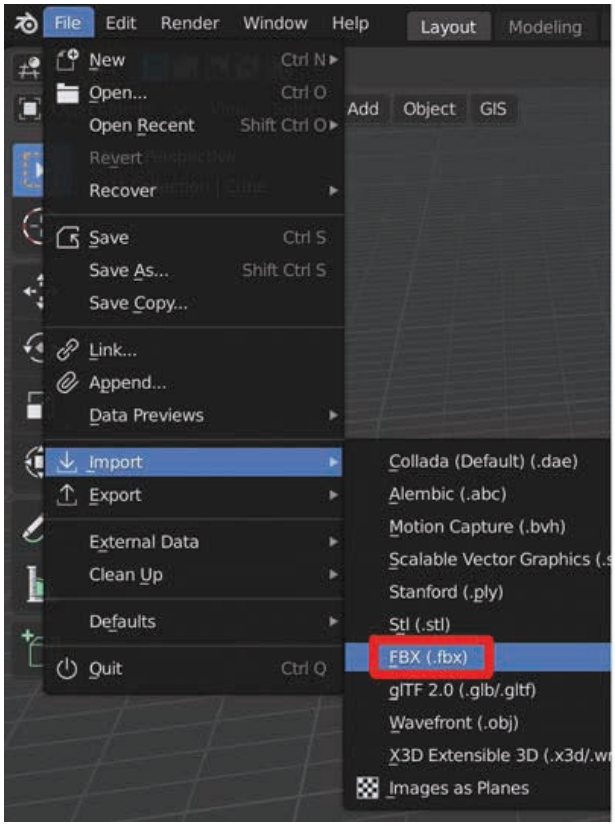
Even the fingers are rigged.

Click **Next** a couple of times.

Choose some animation and download the character.



Import the FBX in Blender. As you can see this is a pretty easy way to create a simple rig for your characters.

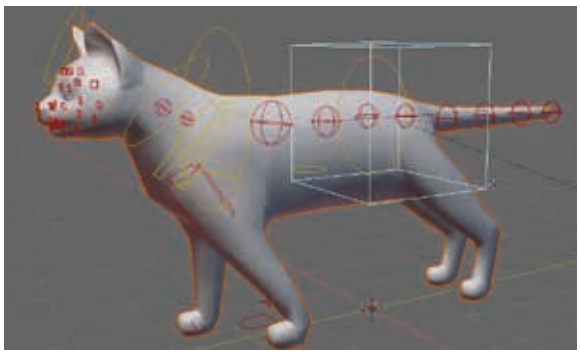


Rigify FK to IK Snap

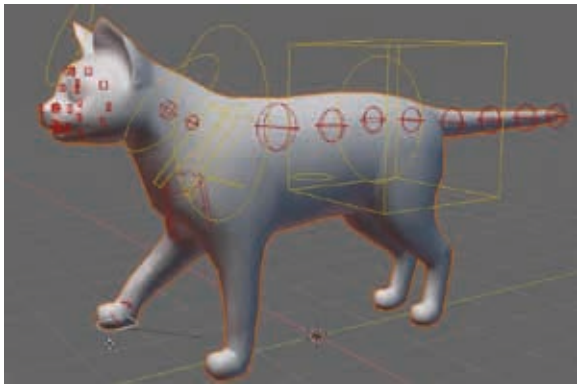
Currently the cat's paws are set to **IK** (Inverse Kinematics) so they don't move.

When we move the main body, the paws stay firmly on the ground (or wherever we have placed them)

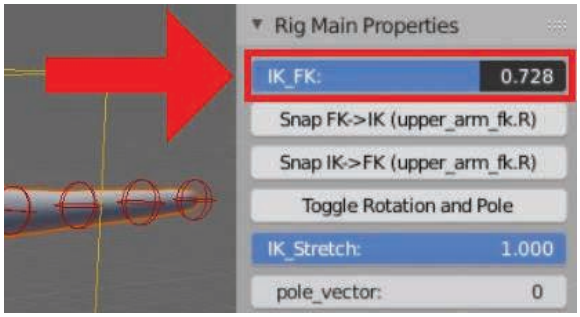
With **IK**, The **parent → child hierarchy** doesn't move forward, in this case for the paws. They are their own boss, and live by their own rules!



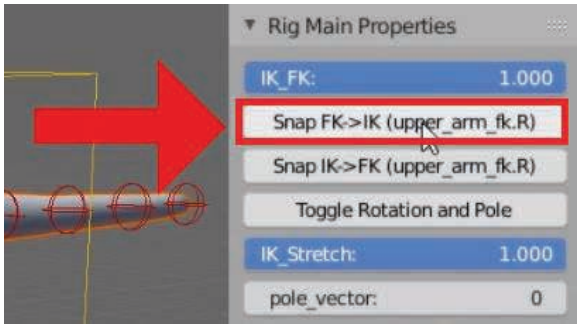
You can move the paws individually. Wherever you place them is where they will stay, even when you move the rest of the body.



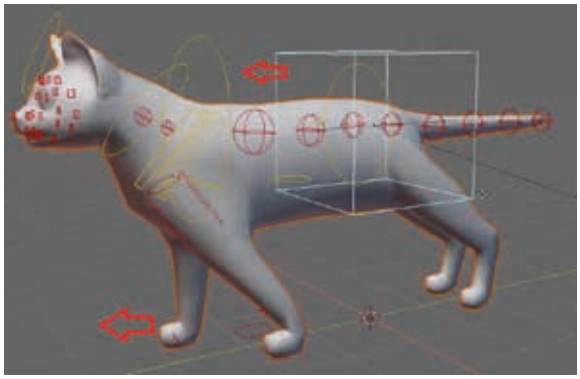
We can change the **IK-FG** slider for the paw to 1, which effectually sets it to **FK** (Forward Kinematics). Now they have to follow the hierarchy.



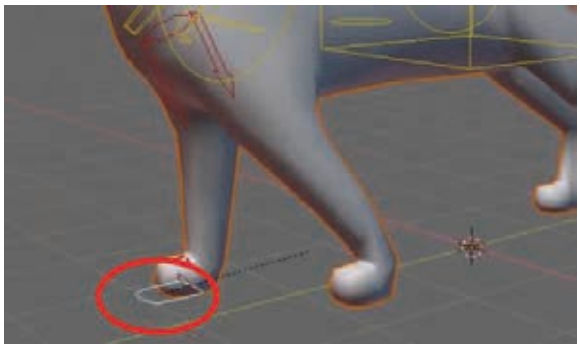
Click Snap **FK-> IK** if necessary to correct the position.

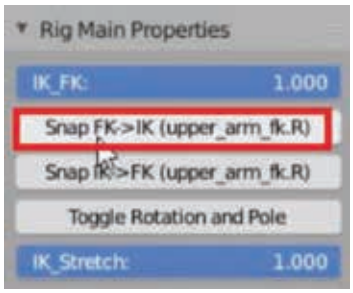


The paw set to FK will follow the bone its parented to. This is Forward Kinematics. The parent-child hierarchy moves forward. So when we move the main body, the paw which is set to FK also moves. The other paws are still set to IK so they remain firmly planted on the floor.



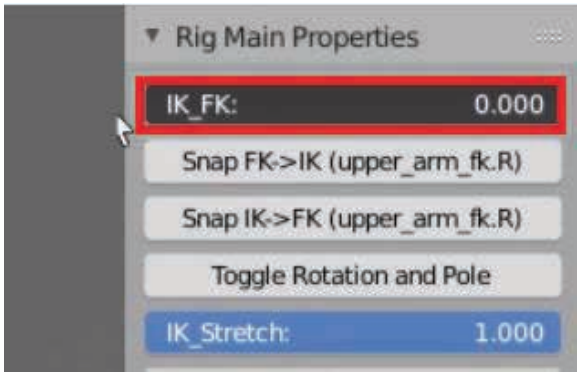
Now, let's suppose you want to place the paw somewhere, and from the moment it touches the ground onwards, switch it back to IK (Inverse Kinematics). Because if you don't do this, it would 'slide' when the rest of the body moves. Move the paw bone (or in this case a Rigify control shape) and then press the Snap FK->IK button.



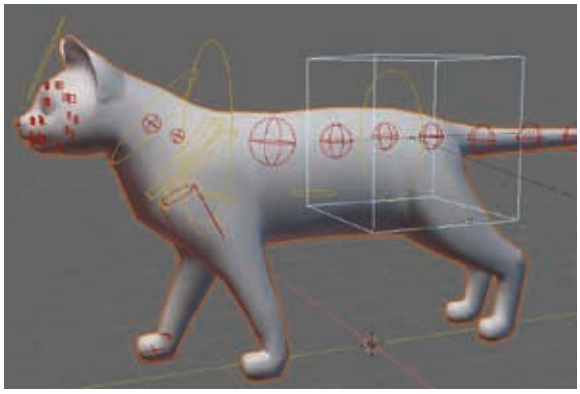


Then turn the paw back to **IK** by setting the **IK_FK** value to 0.

This is how you can change the position of the paw so they don't slide on the floor.



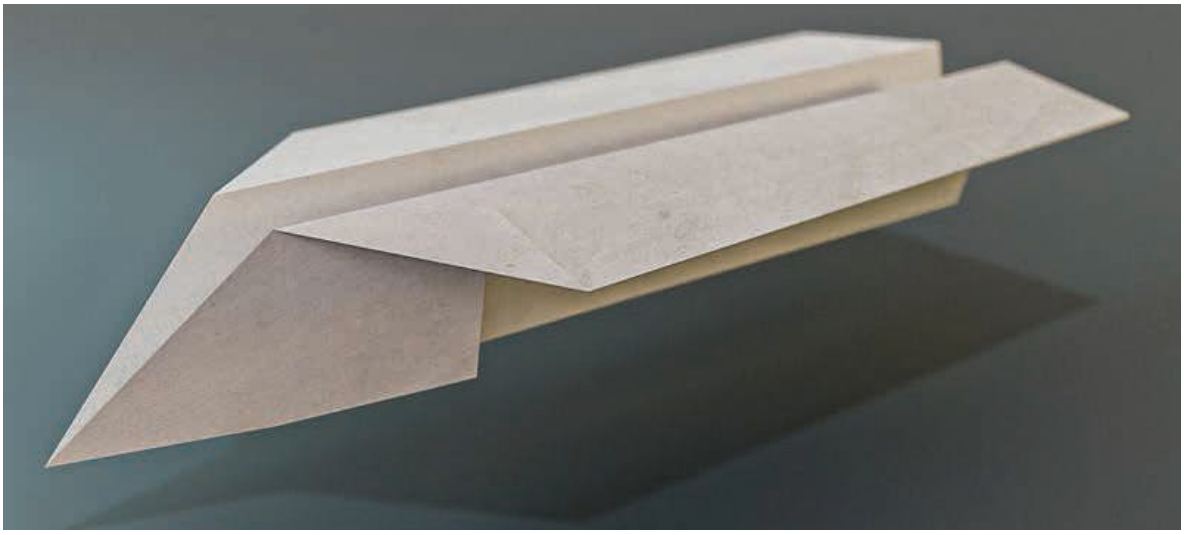
It may seem complicated, but if you understand this key principle of when to switch between IK and FK then you've mastered an important part of character animation, not just for walking (although characters do a lot of that!) but also for example climbing a wall.



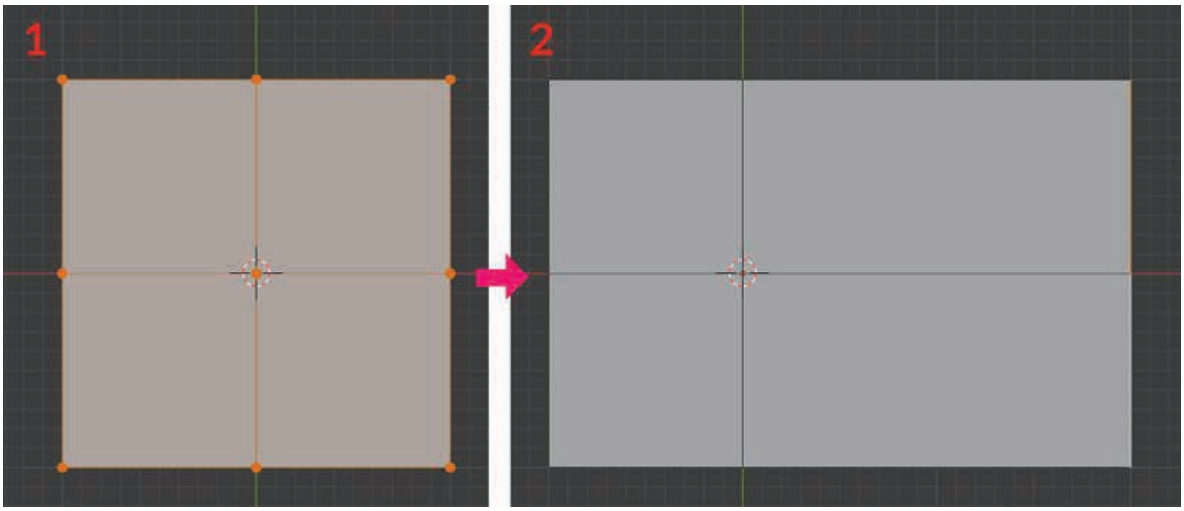
Cat model by Gwinna on Blendswap.com, rig made with the Rigify addon.

Rigging a paper plane

In this tutorial we'll look at creating, rigging and animating the folding of a paper airplane. We'll use a simple design, but once you understand the principles you can apply it to more complex versions and even animate complex origami shapes.

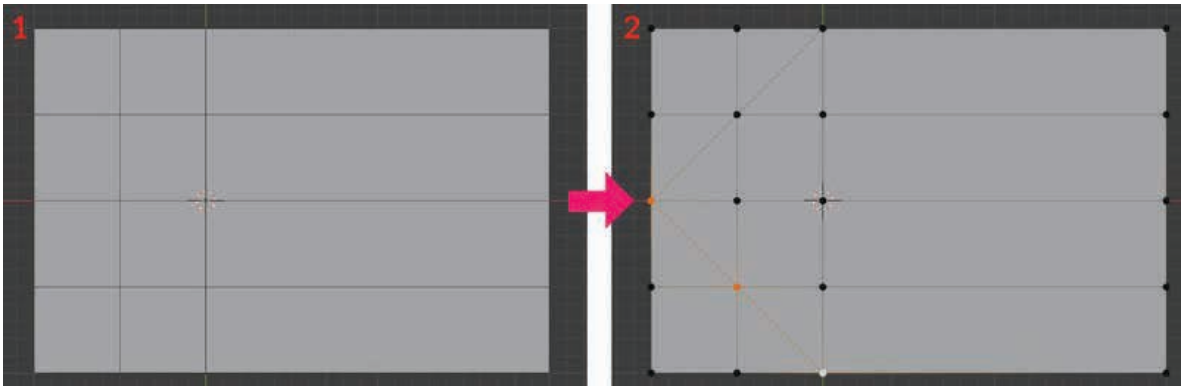


First let's make the paper plane. Delete the default cube, then **press Shift+A** and choose **Mesh > Plane**. Next **Press 7 on the numpad** to see the plane from the Top Orthographic view. In Edit Mode, **right-click and choose Subdivide**. Select the edges on the right and move them 1 unit to the right by **pressing G+X+1**.

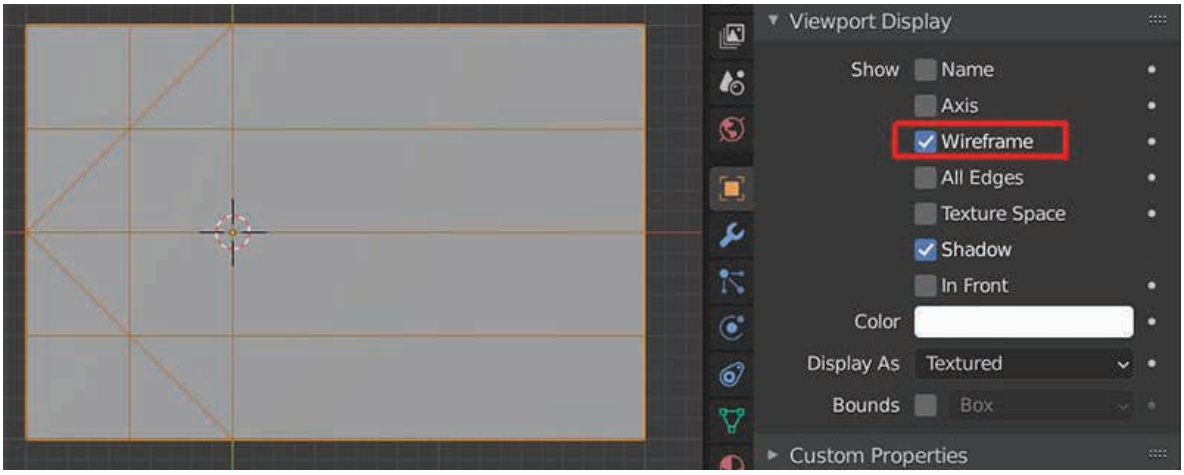


Add two more edge loops on the left by **pressing Ctrl+R**, then **left-click to confirm** and **right-click to exit** the edge loop function without moving the edge loop. Do the same horizontally to create two more edge loops (see picture below left).

Create diagonal edges on the leftmost two squares by selecting vertices and **Pressing J** to create an edge between them (as shown below right). Note that you can't create both diagonal edges at once, rather you have to make one and then the other.



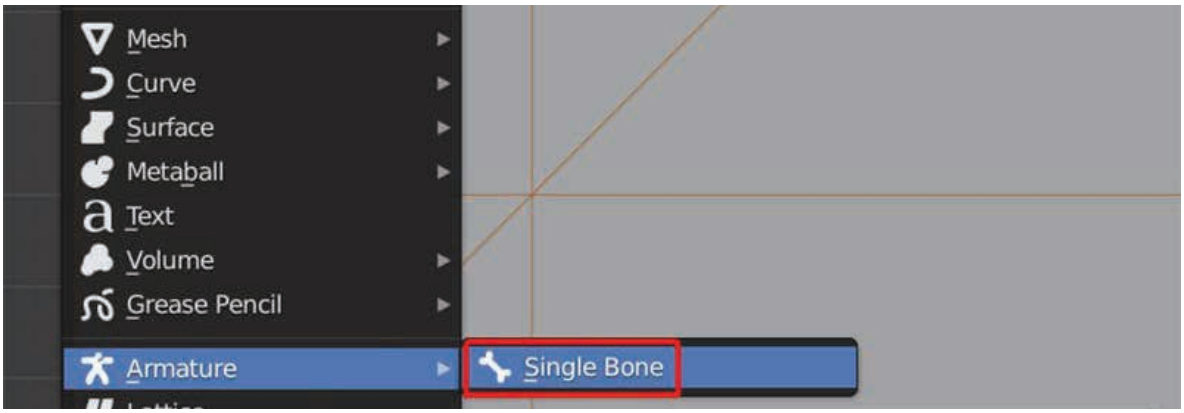
In the Plane Viewport **Display Settings**, turn on **Wireframe**. This way we can see the wireframe even in Object Mode. This can avoid confusion later.



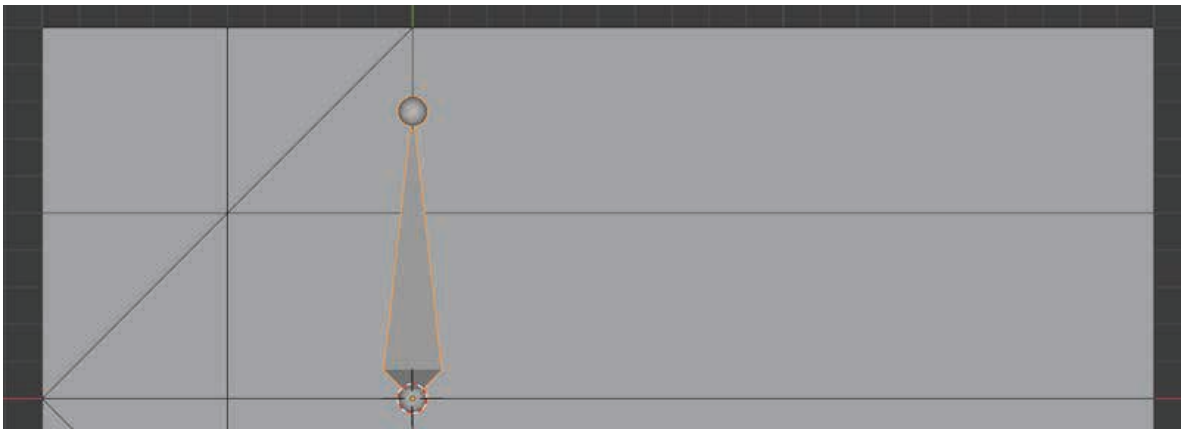
So far so good, now we can move on to the rigging process.

First, enable the **Rigify add-on** in Preferences. This gives us the option of adding a single bone.

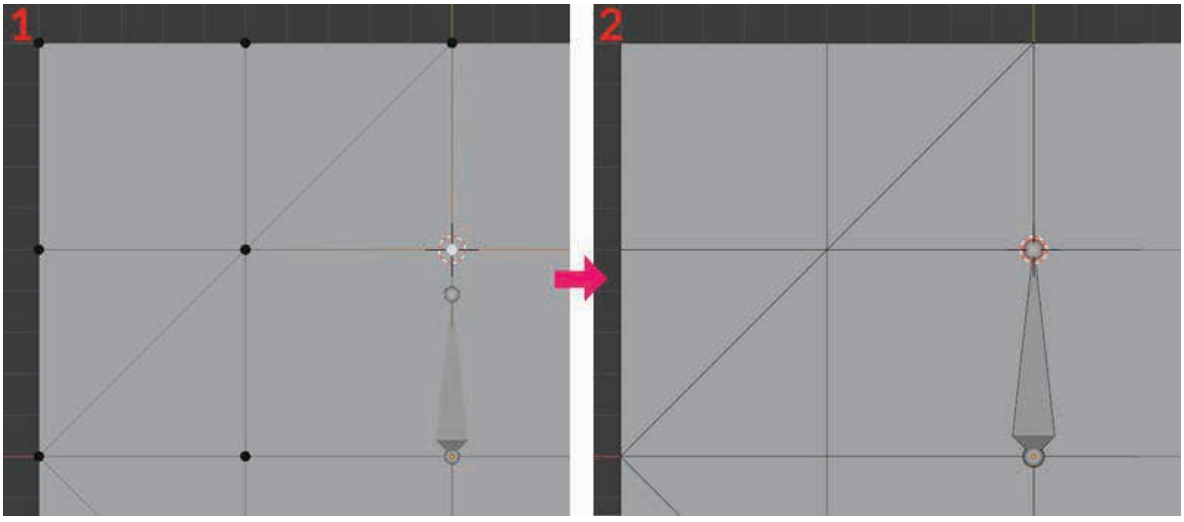
Make sure the **3D cursor** is still at the world origin, and **press Shift+A** and add an **Armature > Single Bone**.



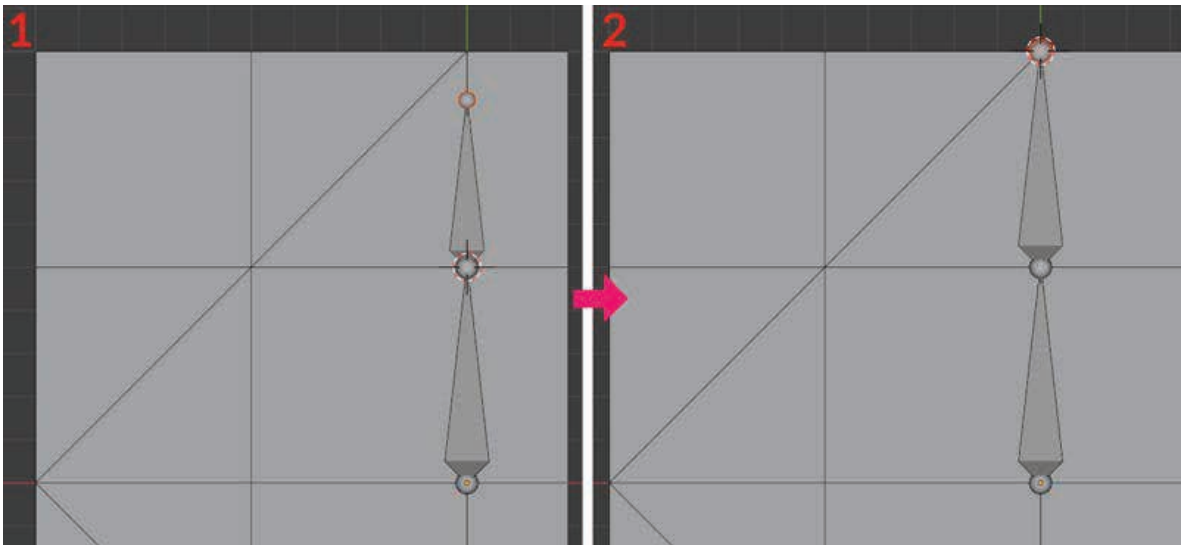
In case you get a giant bone, scale it down first. Then, rotate it -90° on the X-axis by **pressing R+X+[-90]**.



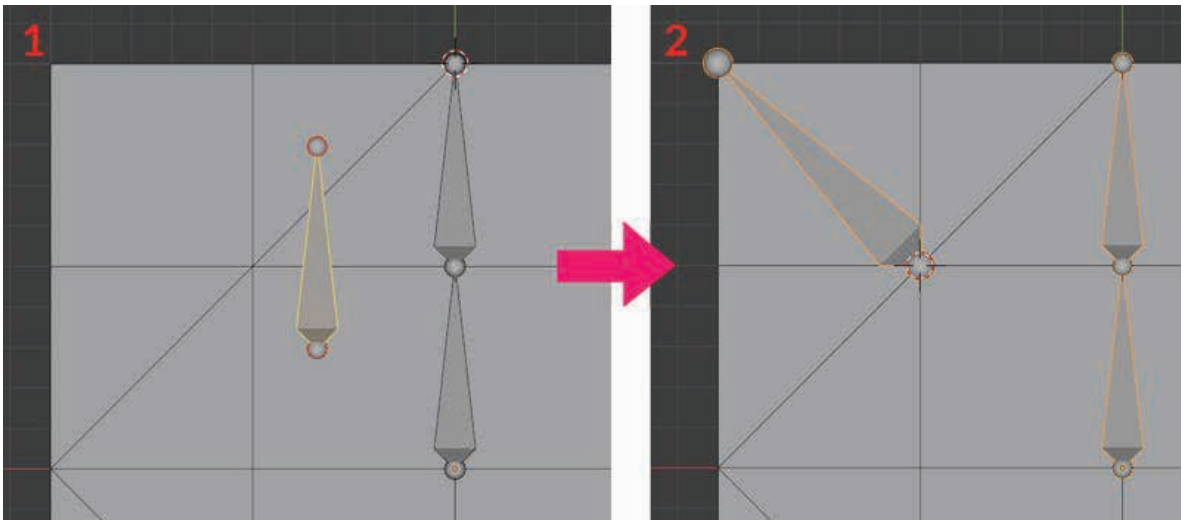
Now, select the vertex shown below and **press Shift+S**, then choose **Cursor to Selected**. This moves the 3D cursor to where that selected Vertex is. Next, select the bone and in Edit Mode select its Tip. **Press Shift+S again** but this time choose **Selection to Cursor**.



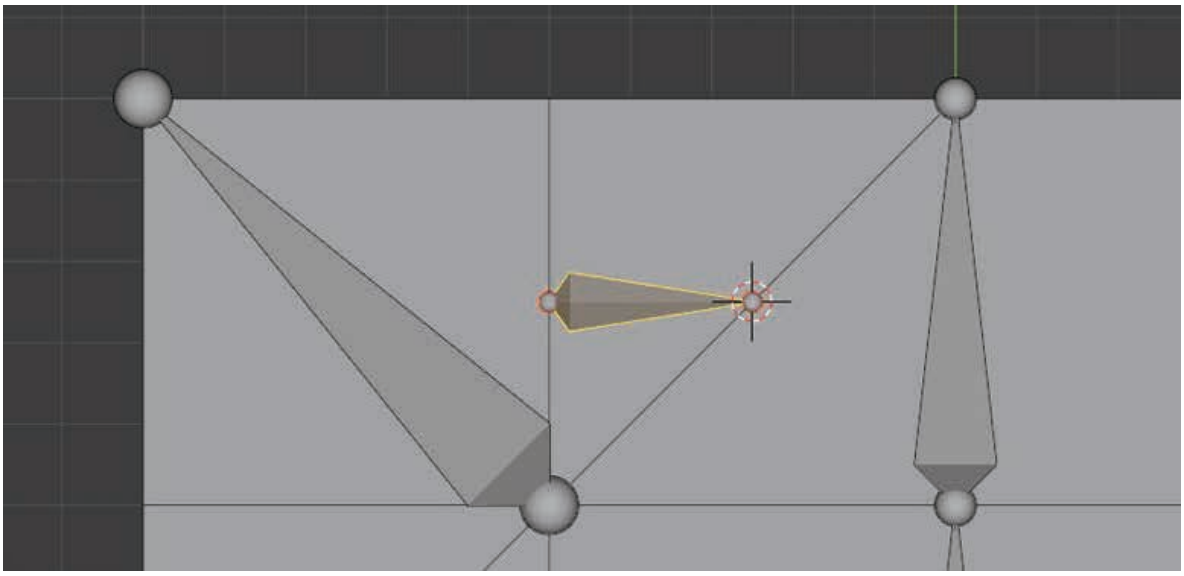
With the Tip of the bone selected, **press E+Y** to extrude a second bone along the Y-axis. Align the Tip with the vertex at the edge of the plane using the same trick as before: select that vertex, **press Shift+S** and choose **Cursor to Selected**, then select the Tip of the bone and **press Shift+S** and choose **Selection to Cursor**.



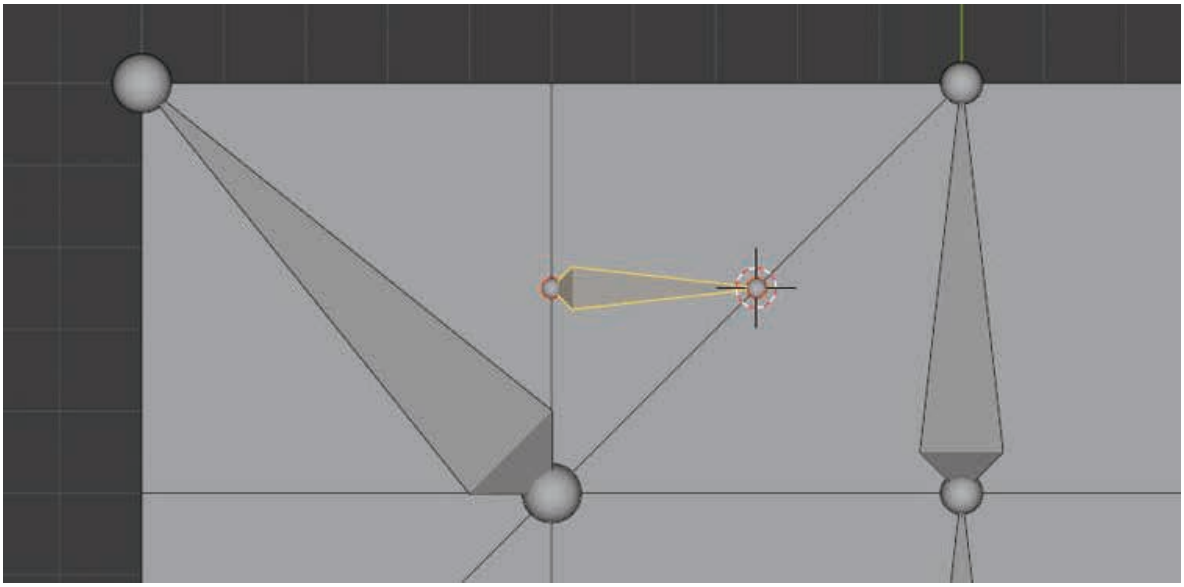
Duplicate one of the bones in Edit Mode with **Shift+D** and then, using the same trick as shown before, align that bone diagonally with the vertices of the top left Face by using the **Shift+S** menu.



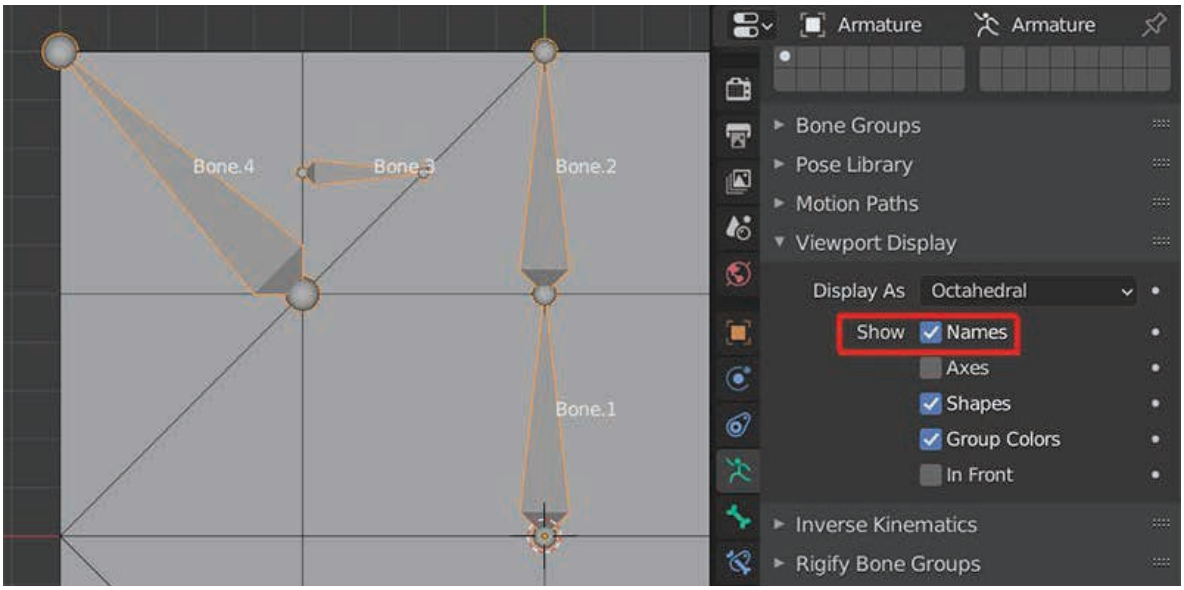
Duplicate another bone, using the same trick as before with Shift+S, align it to the edges as shown below. Instead of selecting vertices, you can select an entire edge and **Press Shift+S**, choose **Cursor to Selected** to move the cursor to the center point of the edge.



You'll notice that sometimes, a bone will be rotated like this one. The rotation of a bone is called "bone roll". We can (and should) correct this. **Press Shift+N** with the bone selected in Edit Mode and choose "View Axis". This should fix the bone roll (as long as you're still in Top Orthographic mode).

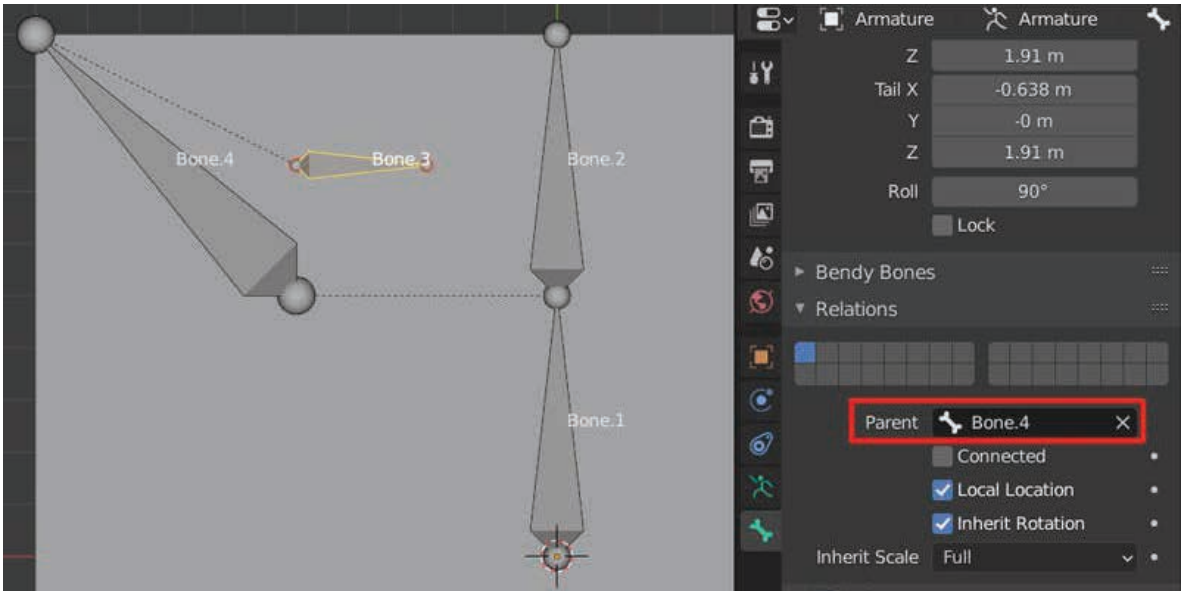


I like to have the Names of the bones visible, and also to rename the bones to a more logical order. I recommend that you do the same. You can rename the bones in the **Bone tab** in Edit Mode.

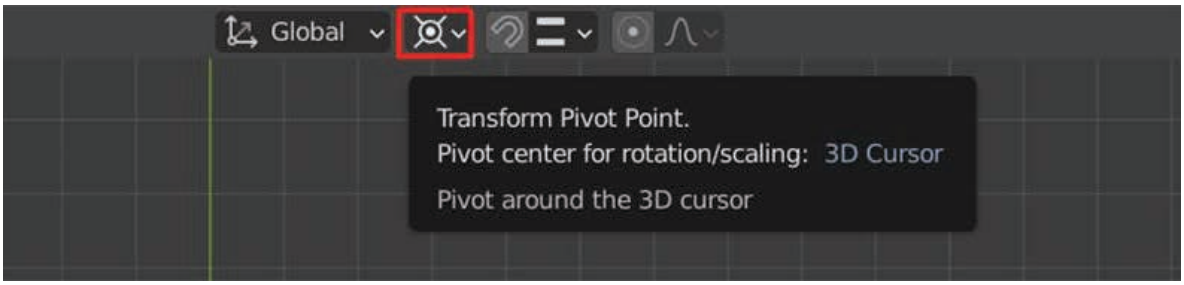


I've turned off the **Wireframe** display for the Plane so you can see the relationship lines of the bones more easily. You'll need to make sure the bones are parented this way: Bone 1 is the parent of Bone 2 and 4. Bone 4 is the parent of Bone 3.

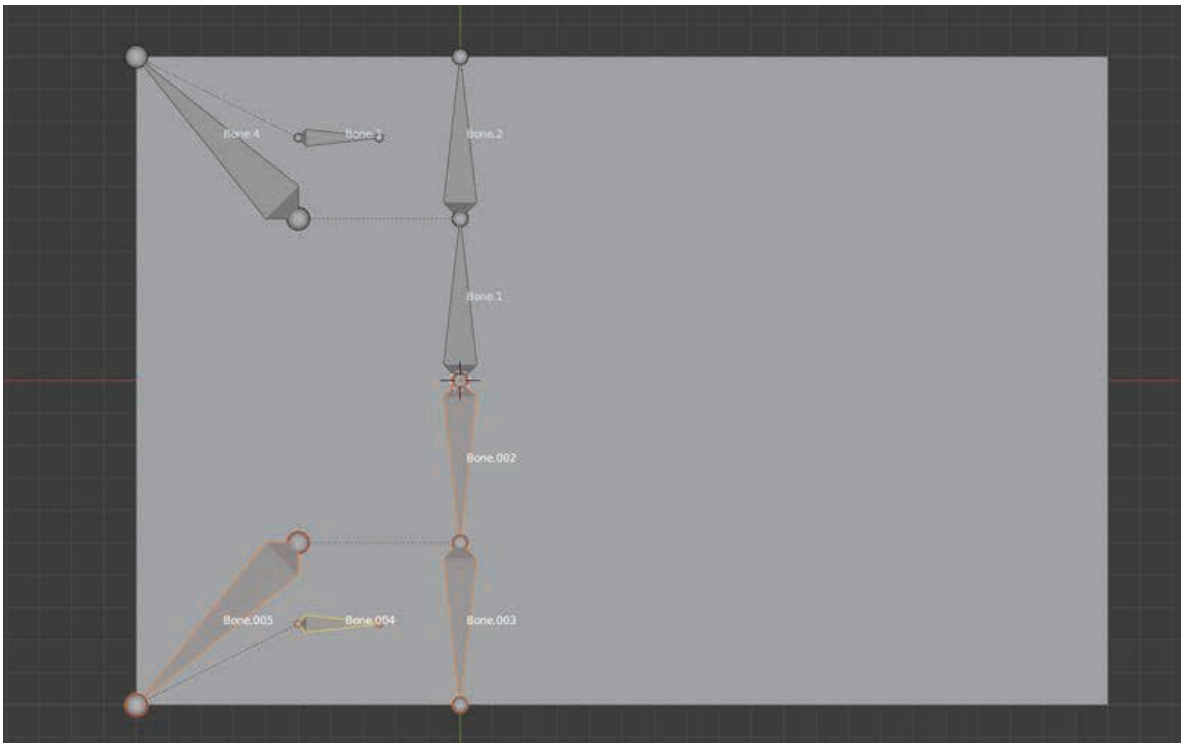
Note that similar to renaming the bones, changing the Parent value also has to be done in **Edit Mode**.



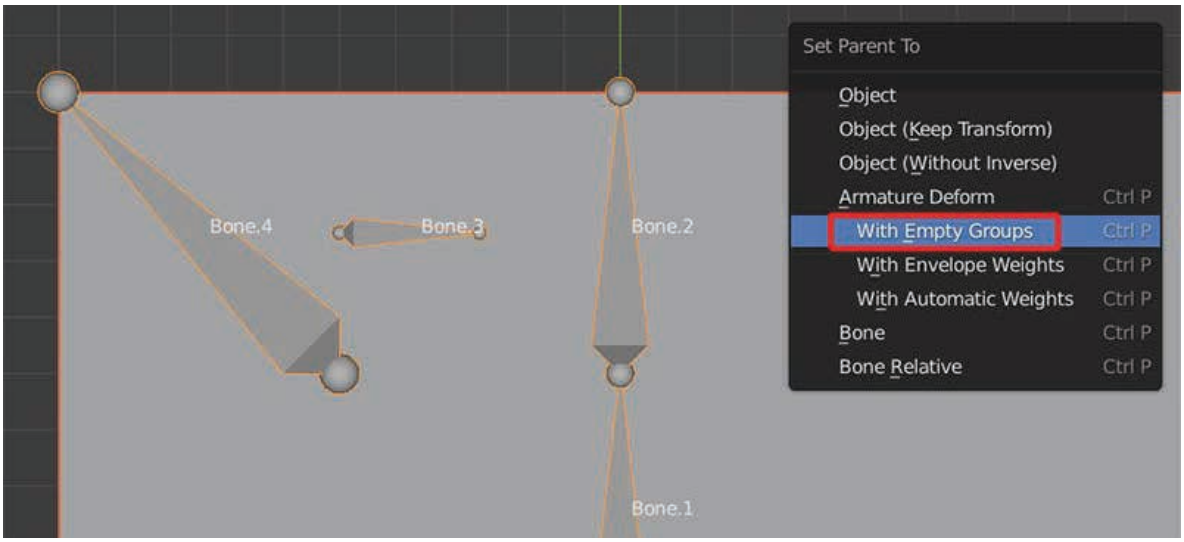
Set **Transform Pivot Point** to **3D Cursor** (and make sure the 3D cursor is at the world origin! If it's not, **press Shift+S** and choose **Cursor to World Origin**.



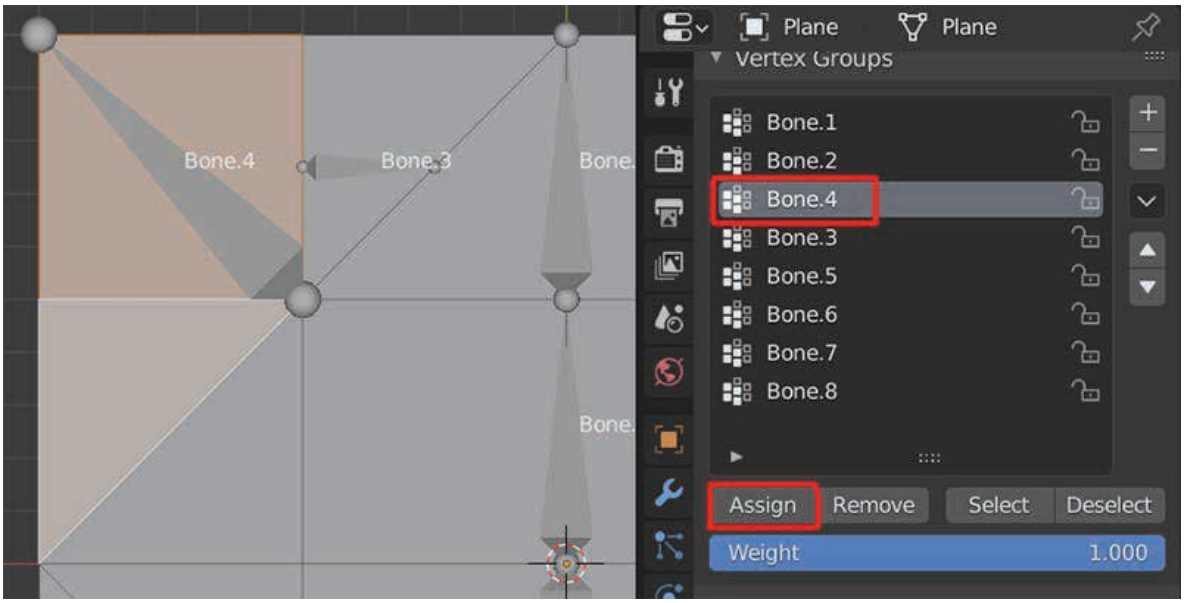
Select all the bones in Edit Mode and **press Shift+D** to duplicate them. Next **Press S+Y+[-1]** to mirror it on the Y-axis. Now we have all the bones we need. Just rename the duplicated bones to keep things organised.



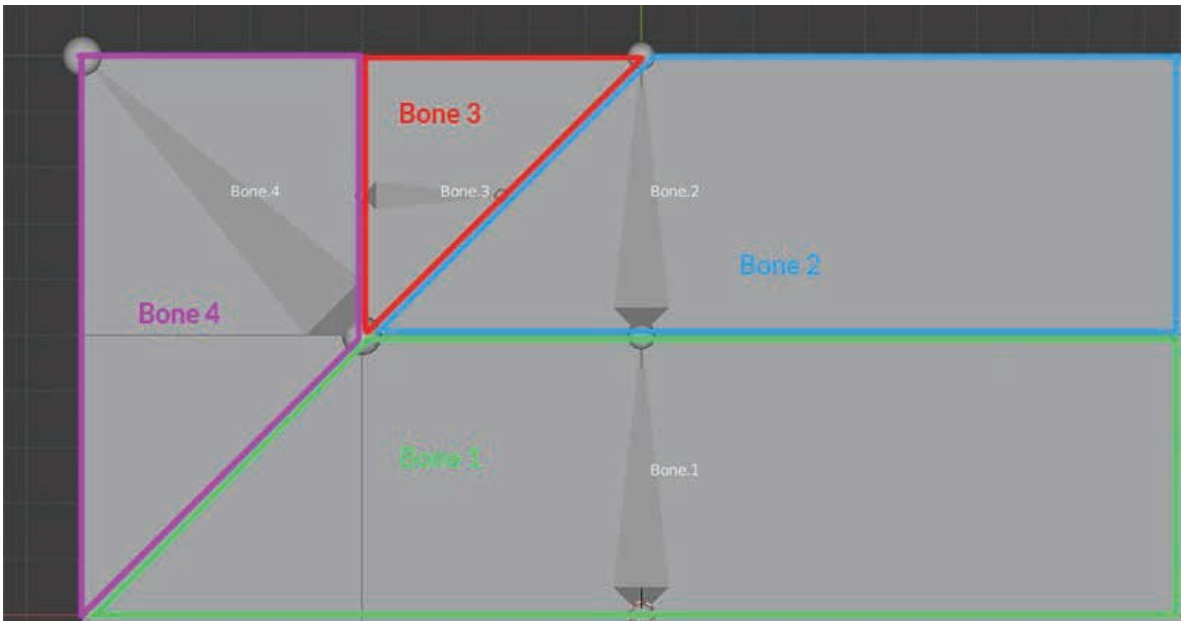
Select the Plane in Object Mode and then, while **holding Shift**, select the Armature. **Press Ctrl+P** to bring up the Parenting menu and choose "**With Empty Groups**". This creates a vertex group on the Plane for each bone, but without actually weight painting it. So if we move the bones in Pose mode, the Plane will not be affected.



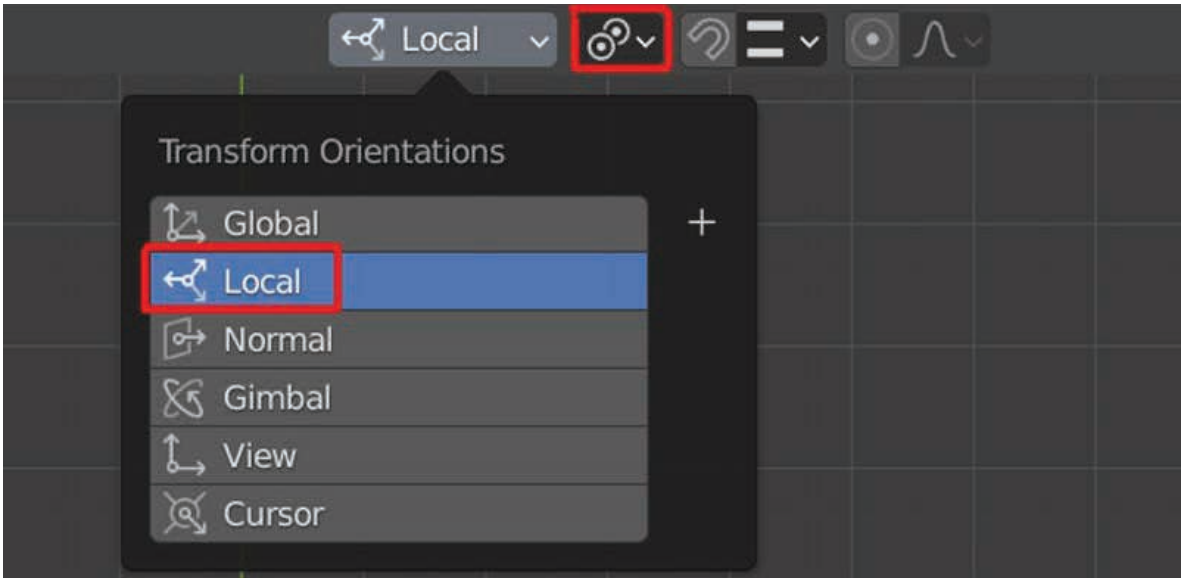
Now, select the Faces that should be assigned to a bone and select that bone Vertex Group. Click **Assign** to manually Weight paint it (with Weight set to 1). Now Bone 4 in Pose Mode will be controlling those Faces.



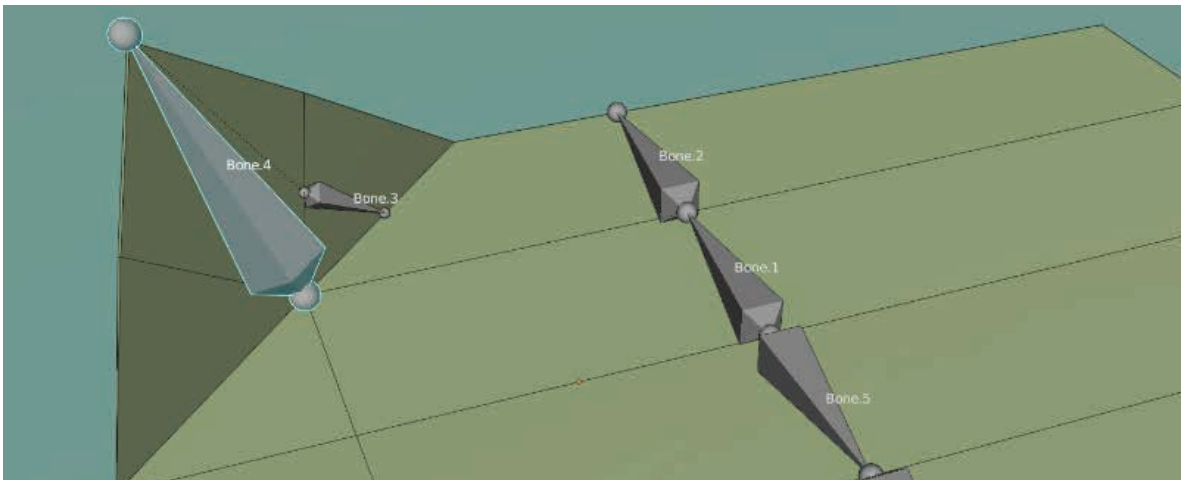
Here's an illustration of which Faces should be assigned to which bones. Following this logic you can also do the other side of the Plane.



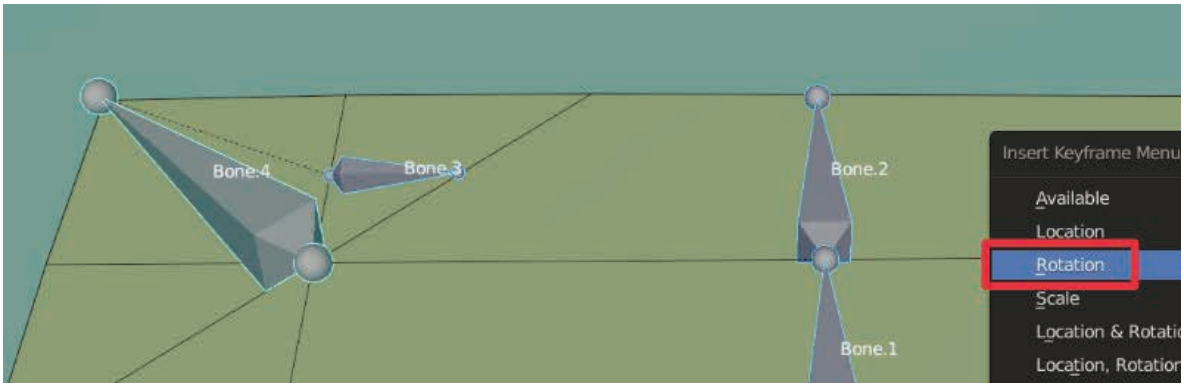
Now that you've Weight Painted the model manually, set **Transform Pivot Point** to **Individual Origins** and set **Transform Orientations** to **Local**.



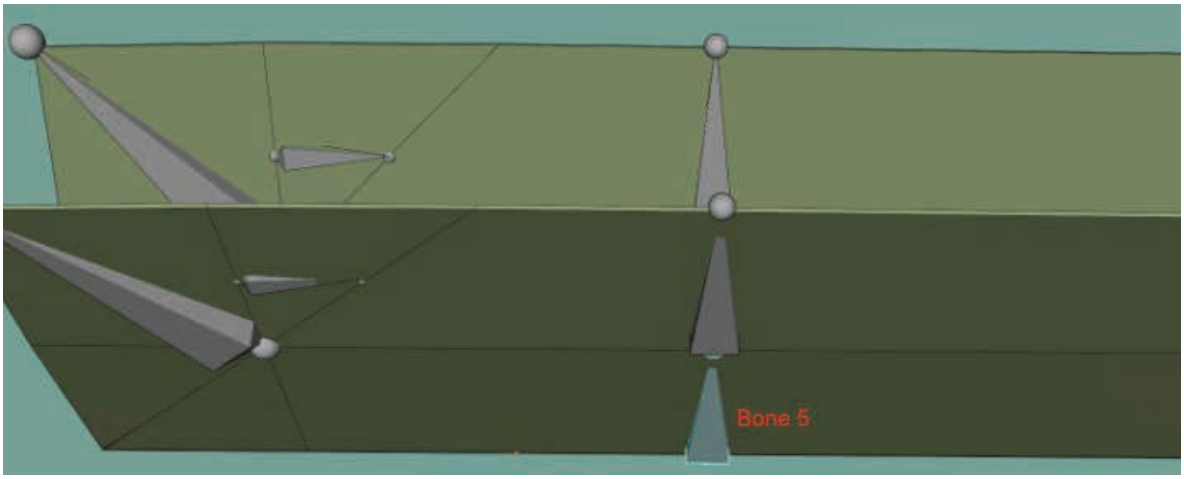
You can rotate the bones in Pose Mode along their **local** X-axis by **pressing R+X**, and the Face that's assigned to it will move along with it. If the rotation is on the wrong axis, just **press X** again. For example:



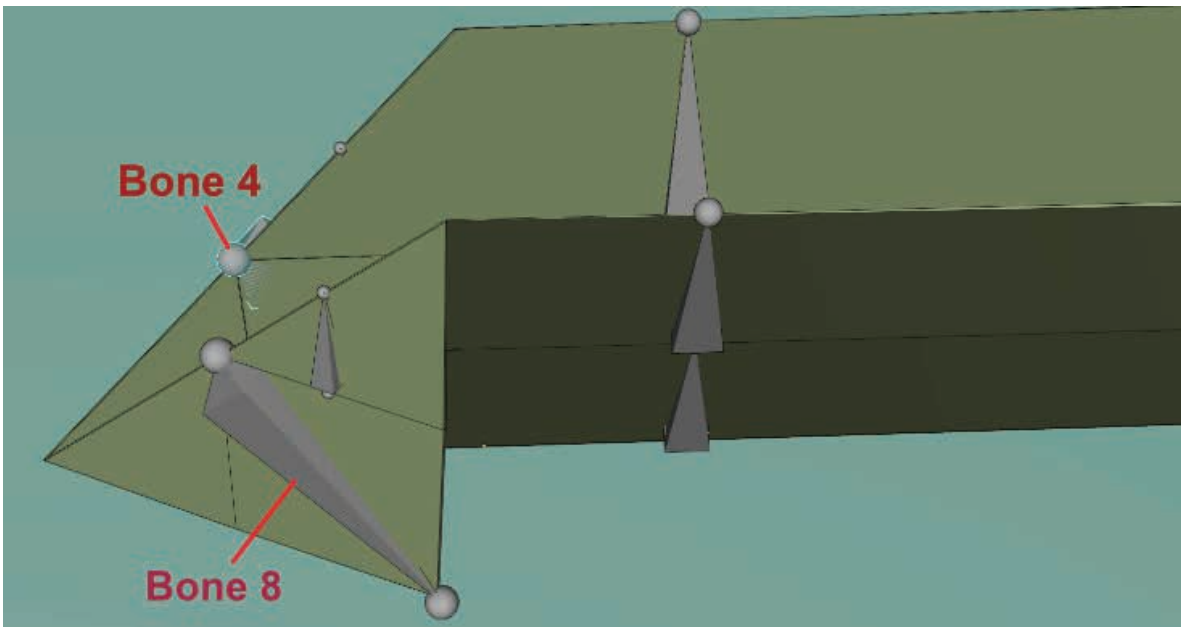
Now you can start animating the folding of the paper plane. At **frame 1**, select all the bones and **press i**, then choose **Rotation** to set a Rotation keyframe. Now there's a keyframe for the flat, unfolded paper.



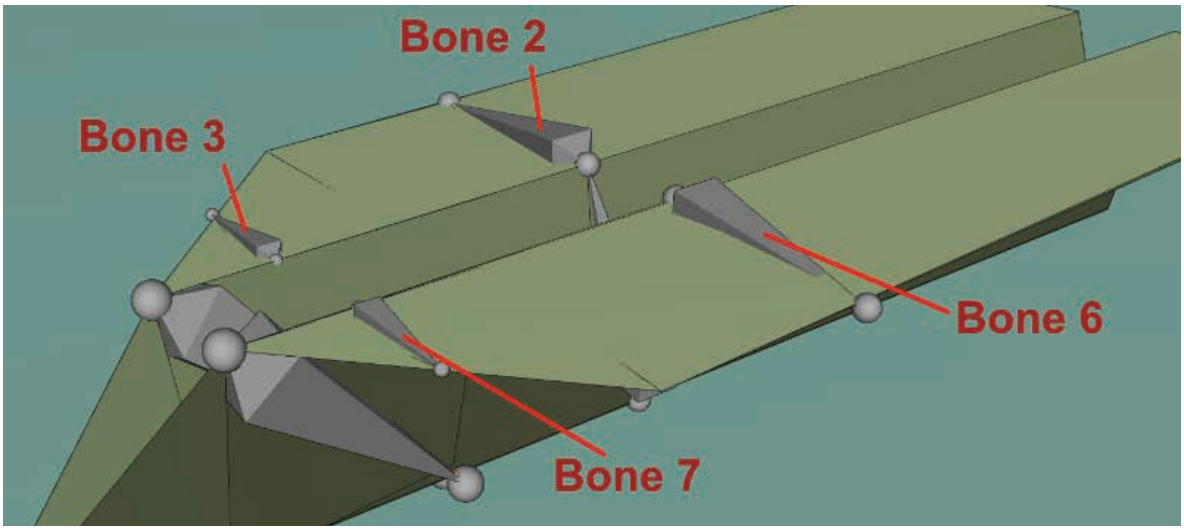
Go forward some frames, to **frame 50** for example (that's 2 seconds). In Pose Mode rotate **bones 1 and 5** individually so the Plane is folded in the middle. Set a keyframe for the **Rotation**.



Go 50 frames further again and rotate **Bones 4 and 8** to fold the diagonal flaps. Set a keyframe for Rotation.

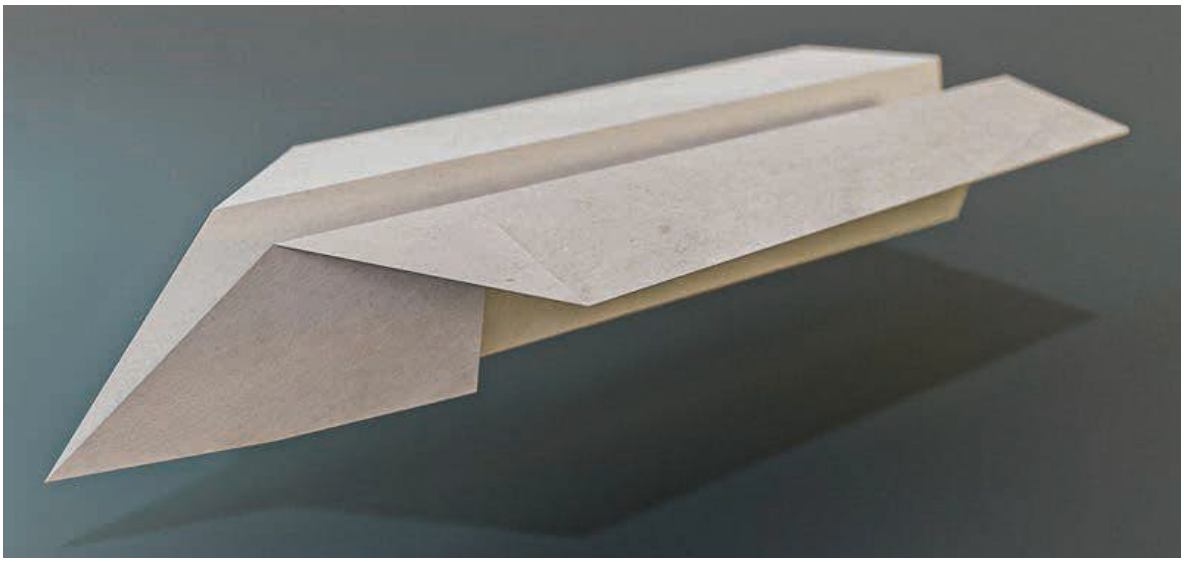


Finally, go 50 frames further and rotate bones **2 and 3**, then rotate bones **6 and 7** and set rotation keyframes.



Now when you play the Timeline from the beginning by pressing the Spacebar, you should see the flat Plane being folded into a paper plane.

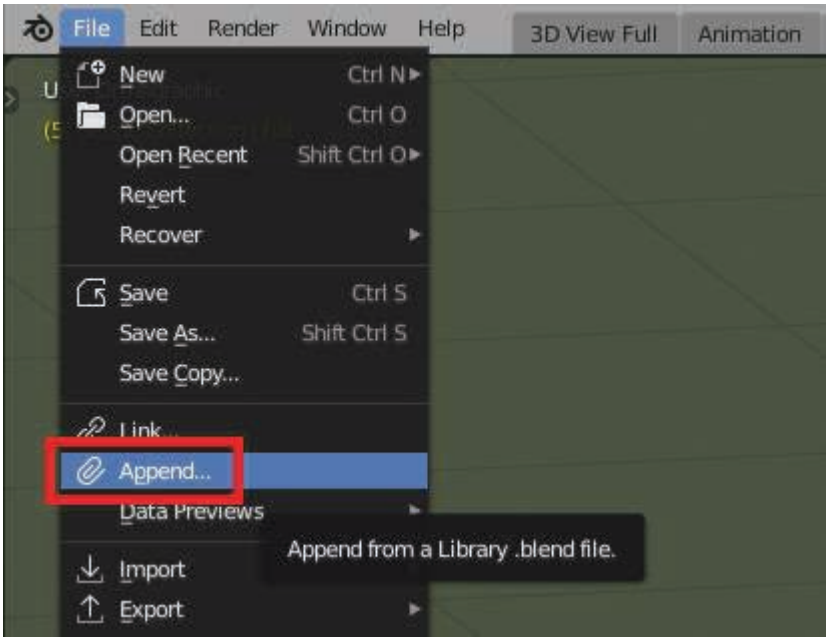
You can add a **Solidify modidier** to make the paper a bit thicker (keep it subtle though) and add a texture. "**Complex Mode**" on the Solidify modifier gave me a better result in this case. I used “PhotoPlain0025” from Textures.com for the material.



Adding Props To Your Character (like a stylish hat)

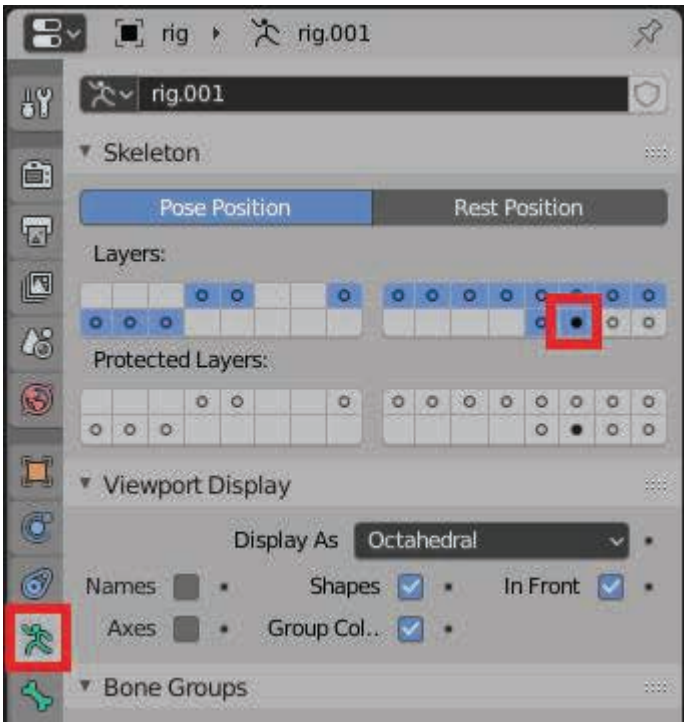
Let's give this little guy a stylish hat.

Model the hat or append it from another file and place it on his head. I found this particular hat on Turbosquid for free.



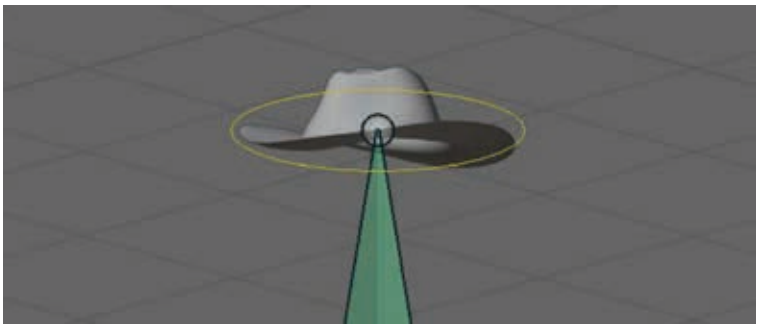
Unhide the armature if it is hidden.

Hide the character by selecting it in Object Mode and **pressing H**. Select the armature and unhide this specific Amature Layer by **holding Shift and clicking on it**.

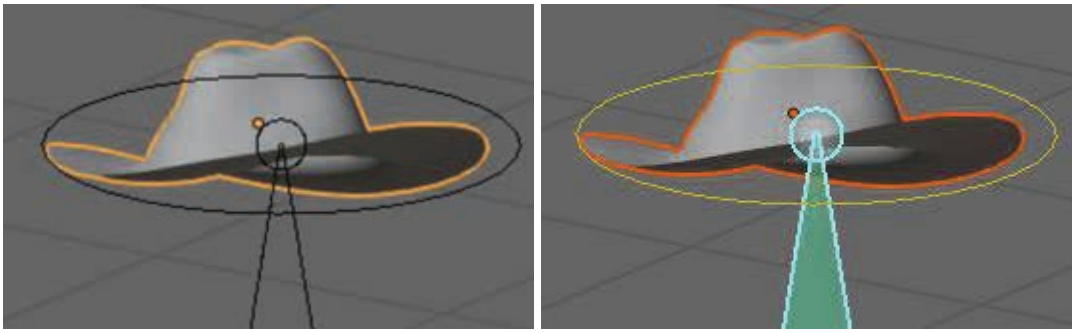


Unhide this Armature Layer.

Go to **Pose Mode** and you'll see some green bones.

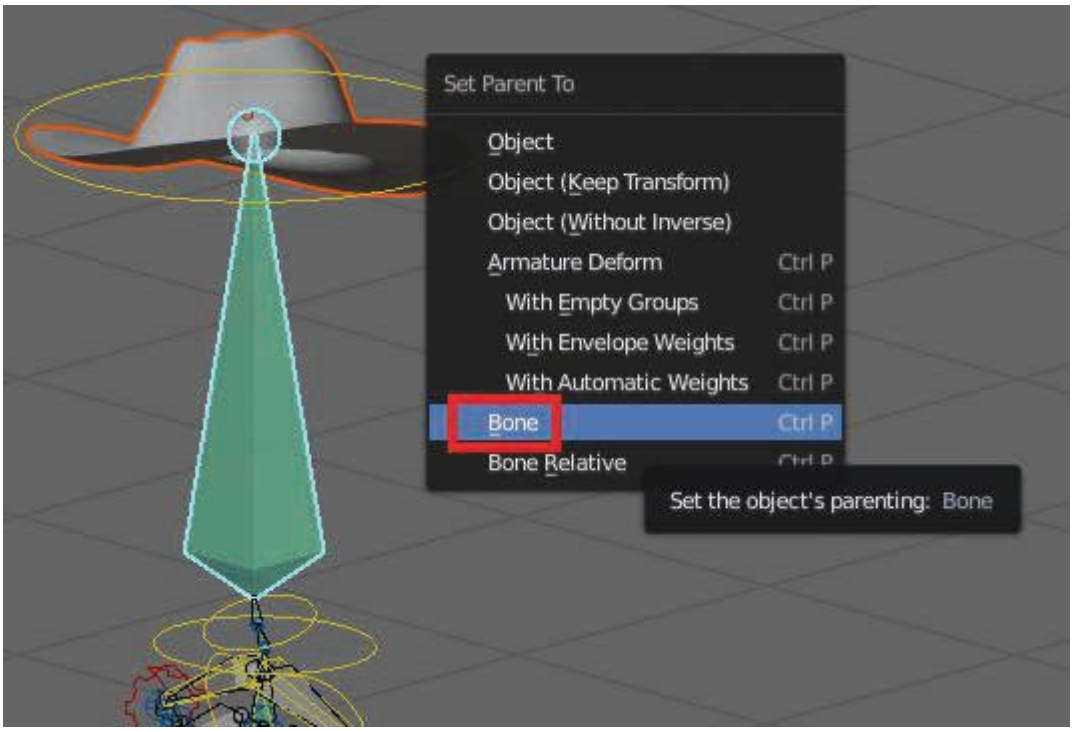


Select the hat, then **hold Shift and select the Head**

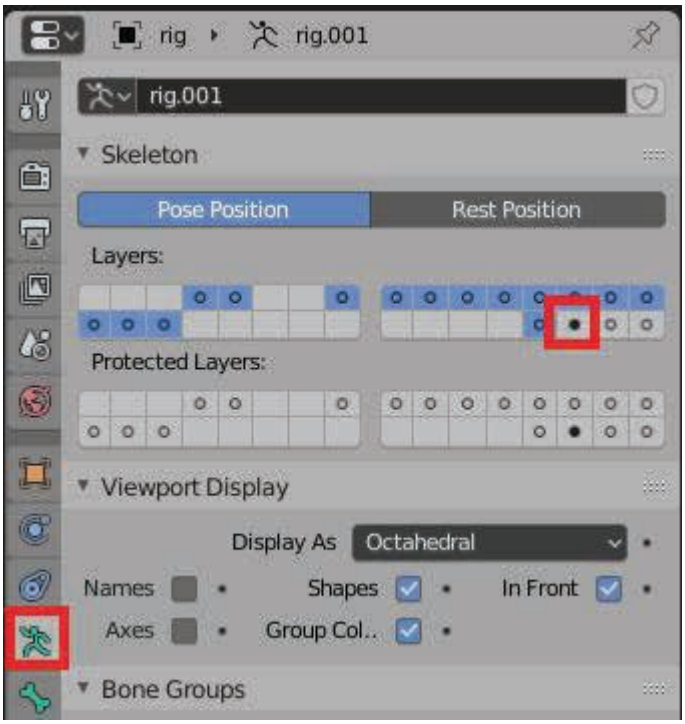


Selecting the hat (left) and then the Head Bone (right) while holding Shift.

Now **Press Ctrl+P** and choose the option “Bone” to parent the hat to the Head Bone.



Hide this Armature Layer again by **holding Shift** and clicking on this specific Armature Layer.



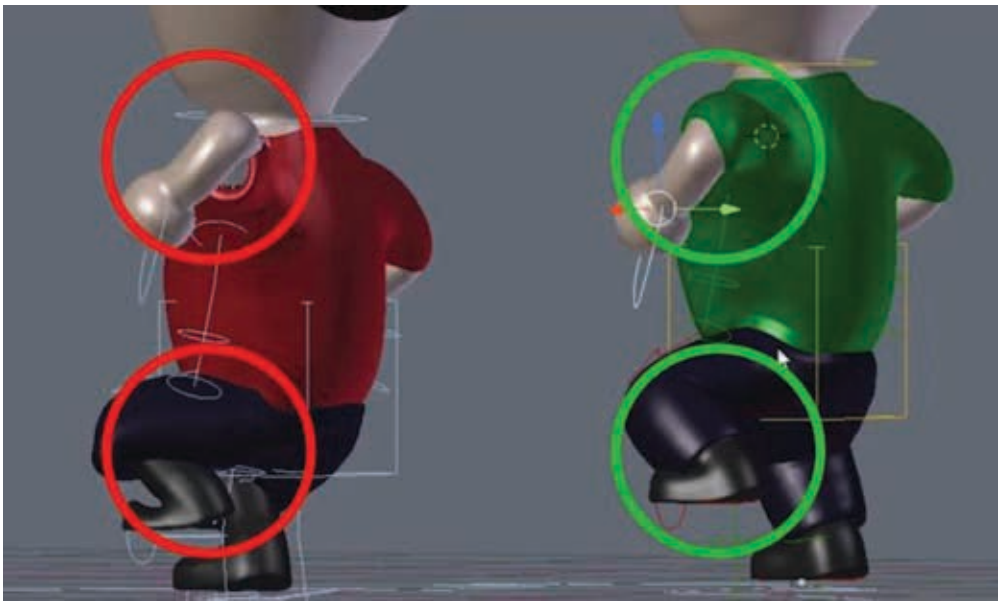
Unhide the character and test if the walk cycle by **pressing Spacebar**. The hat will now follow the Head Bone.

Rigging With Separate Objects Or Not?

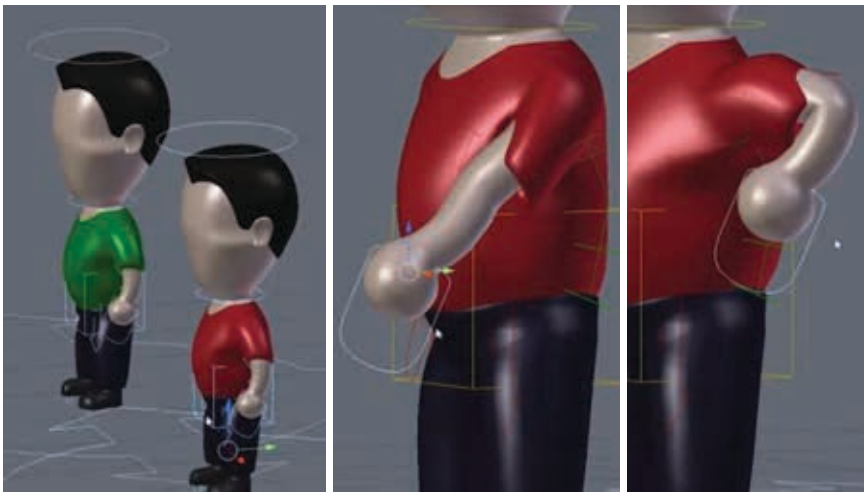
For simple characters, is it really necessary to have the clothing be seperate meshes? Often, it is better to have the whole character be one mesh in such a case. That way, you can easily bind the character to the armature with Automatic Weights with little problems.

The character on the left has clothes as seperate meshes.

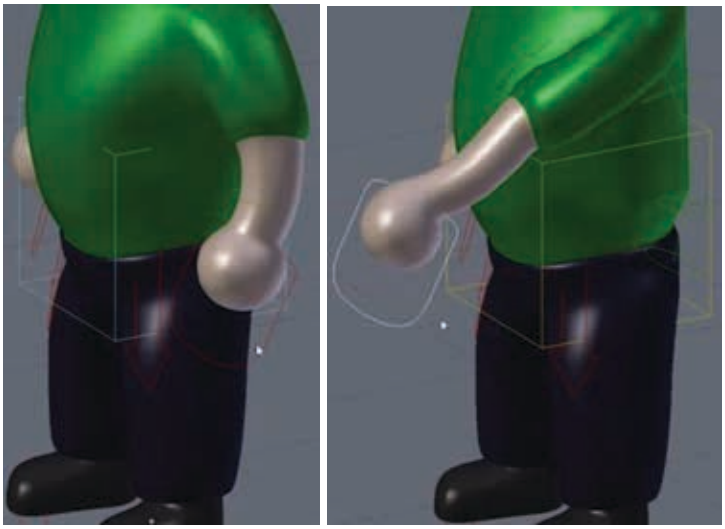
The character on the right is one mesh. Both use the same armature and "**Automatic Weights**". The result on the right clearly can be deformed with less issues.



These two characters have the exact same armature.



The guy in the red shirt consists of separate objects, the clothes are seperate parts. As you can see, positioning the arms and legs creates overlapping and clipping problems.

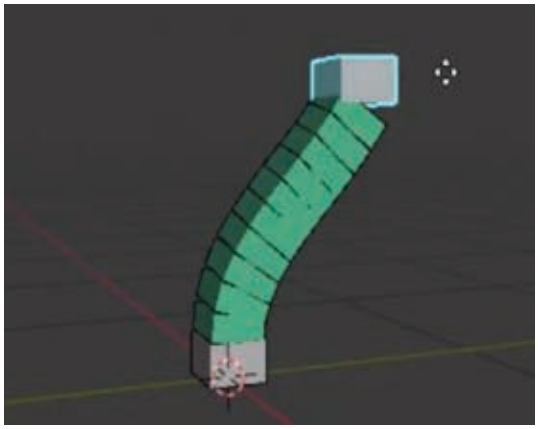


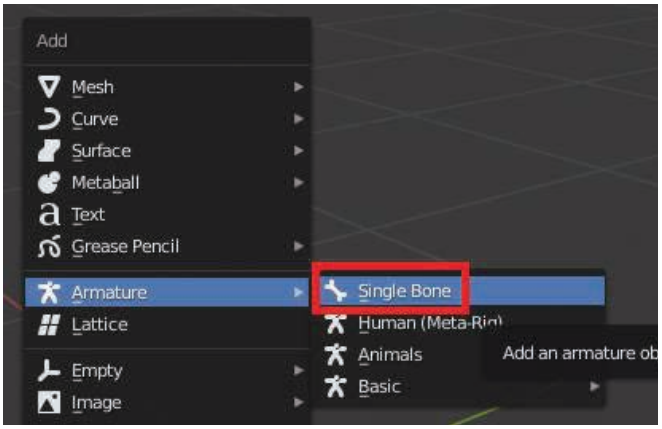
The green guy is one single object. The clothes are connected, even to his “skin”. As you can see, with the same rig, his body can be stretched without overlapping problems.

Again, both characters use the exact same armature and rig and are set up with automatic weights. Can you never have clothes be seperate objects? You sure can! However you will need more bones.

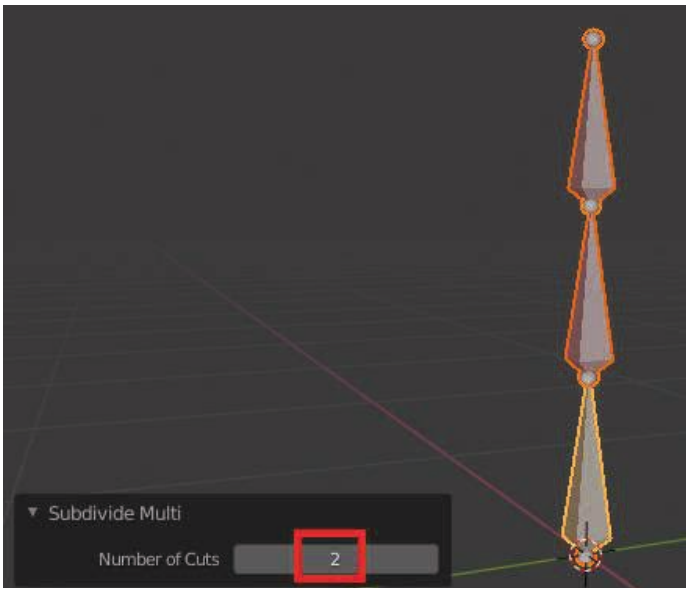
Bendy Bones

Bendy Bones (or B-bones) are a fun alternative to setting up long bone chains with many bones, giving you an easy way to control smooth, bendy curved limbs or facial features. Press **Shift+A** and create a **Single Bone** (assuming Rifigy add-on is enabled).

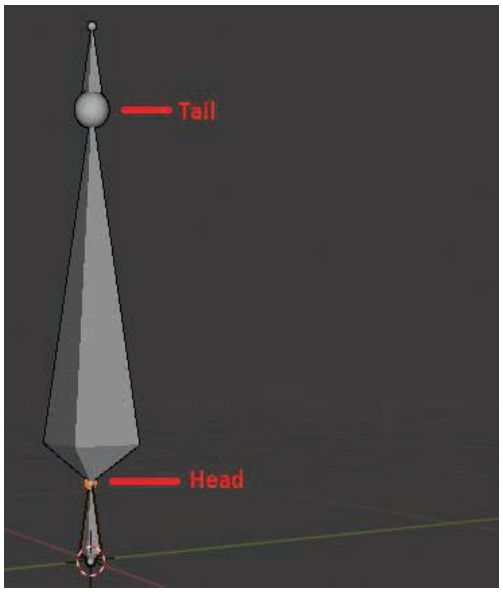




In Edit Mode, select the bone (press A to select all of it), **right-click and choose Subdivide**. Set it to 3 segments (or 2 cuts, in the Subdivide menu that pops up in the viewport).

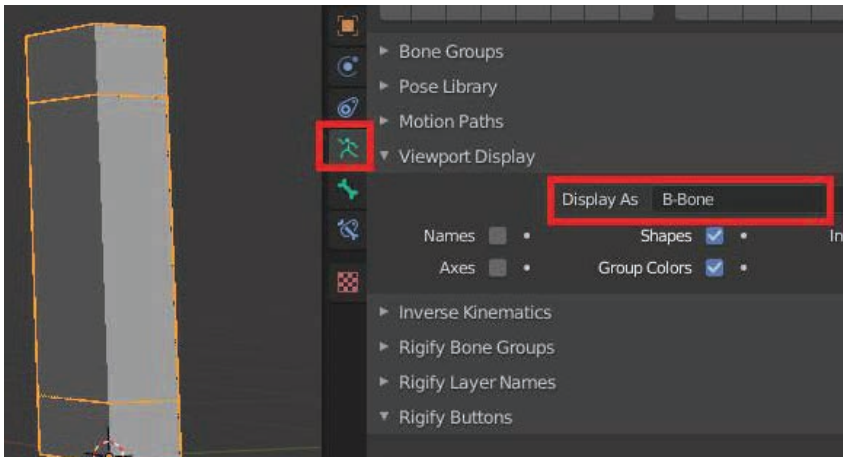


Still in Edit Mode, move the **Tail** and **Head** of the middle bone to make it bigger compared to the other two bones.

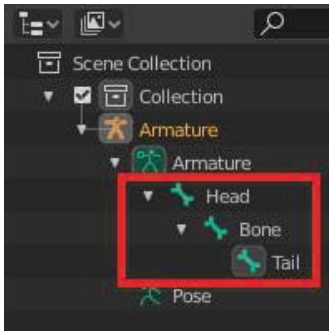


Click here to return to the index.

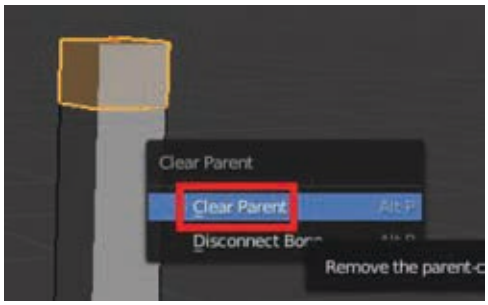
In the **Armature** tab, under Viewport Display, set the display type to **B-bone**.

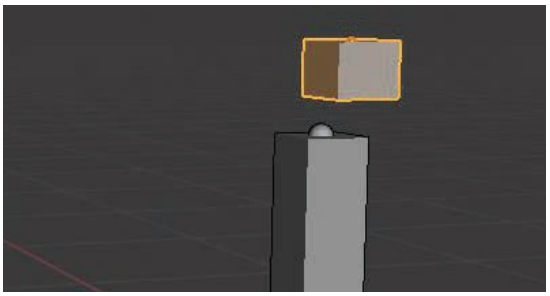


In the Outliner, rename the bones to **Head, Bone and Tail** as shown:

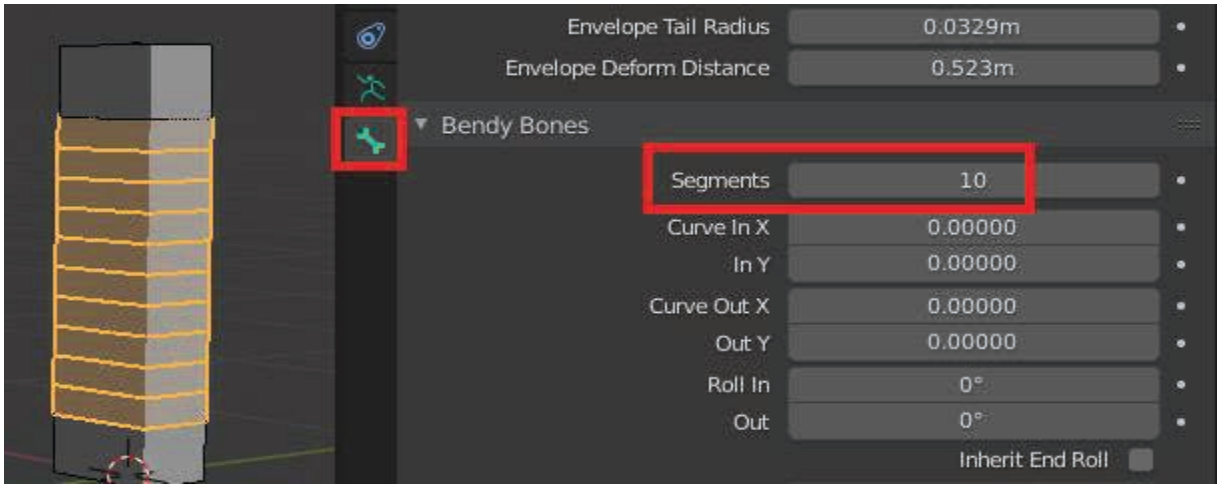


Select the top bone in the viewport in Edit Mode and clear its parent-child relationship by pressing **Alt+P** and choosing **Clear Parent**. You should now be able to move the top bone freely, independently from the rest.

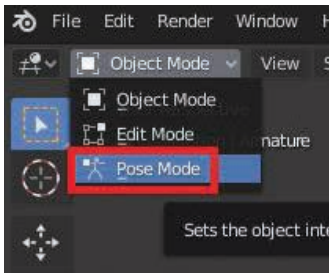




Select the middle bone. Increase the **segments** in the Bendy Bones tab.

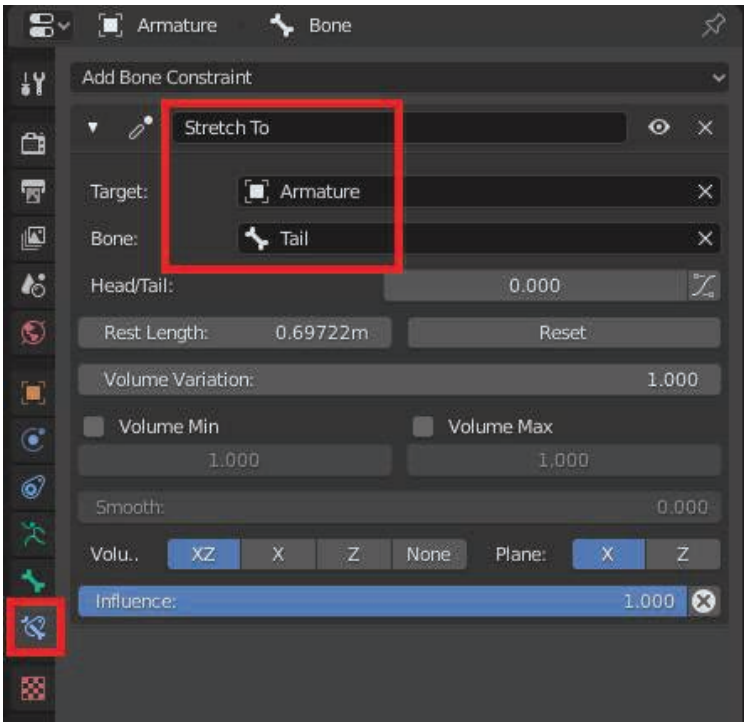


Go to **Pose Mode**.

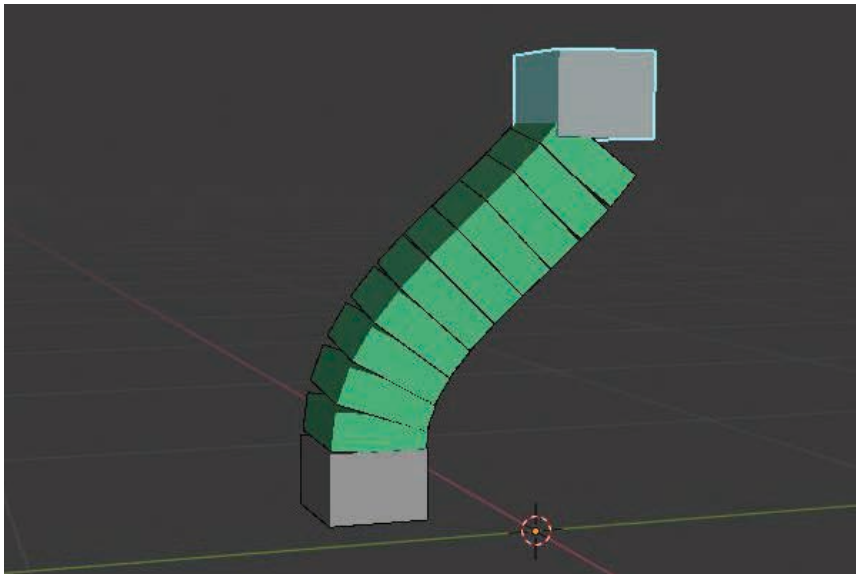


Add a **Stretch To** Bone constraint. Set the Target as **Tail** bone.

Make sure you are in **Bone Constraints**, not Object Constraints! The two icons look similar.

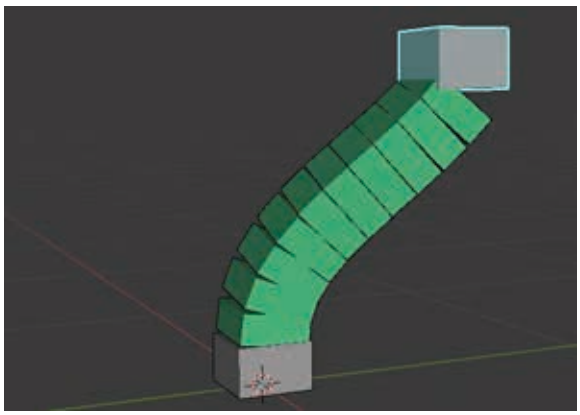


That's it, you should now be able to grab the Tail or Head bone to bend the rest of the bone!



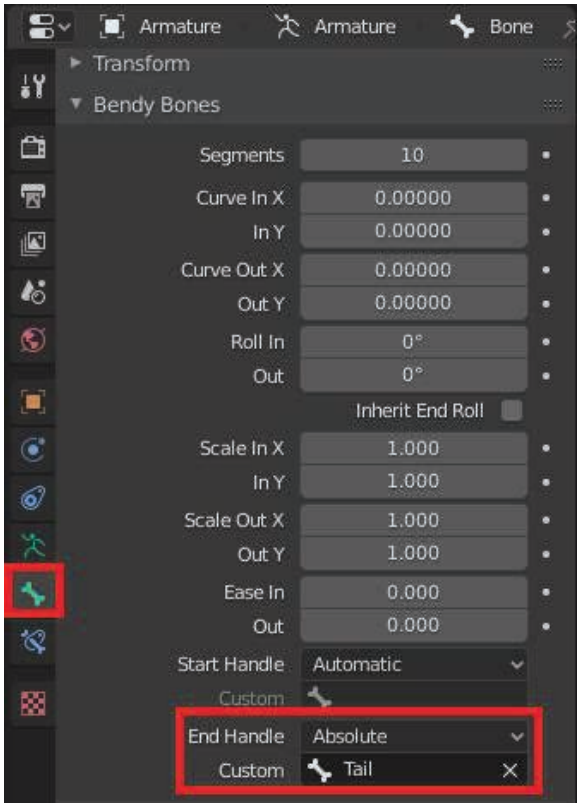
Bendy Bones part 2

This is where we ended up with the previous Bendy Bones tip.



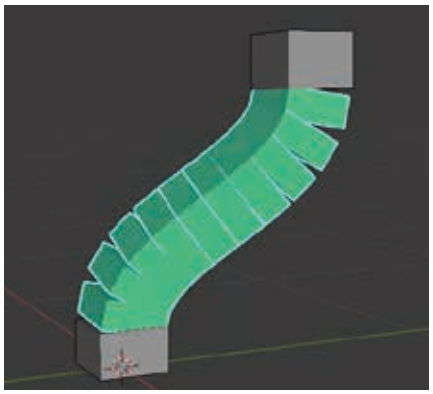
Select the middle (green) segment while still in Pose Mode.

In the Bone tab, under Bendy Bones set **End Handle** to **Absolute** and choose Custom: Tail.



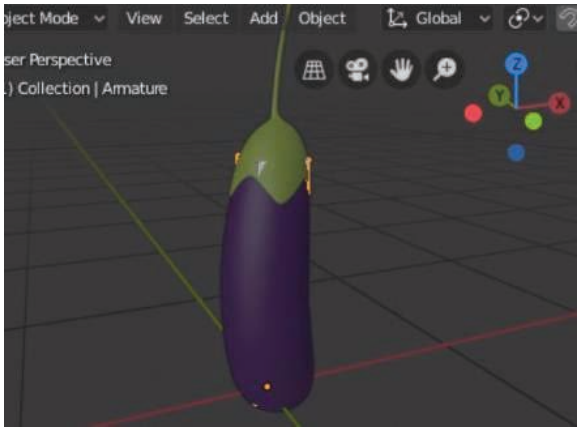
Now rotation is also taken into account. But how do we actually use these bendy

bones to pose and ultimately animate a model?

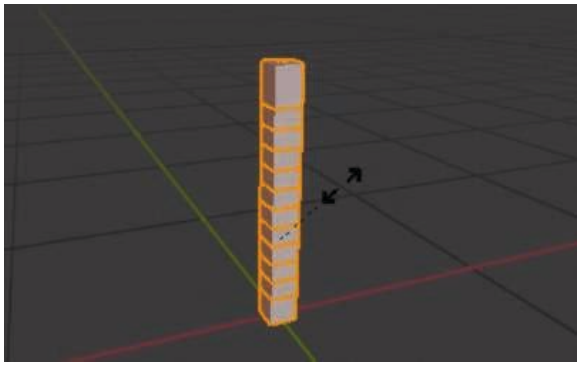


Click here to return to the index.

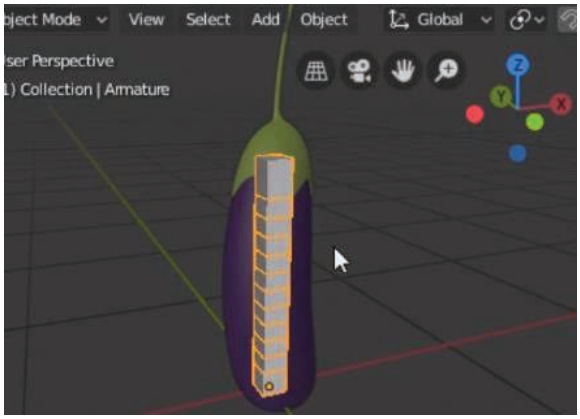
As an example, let's use it to rig this eggplant!



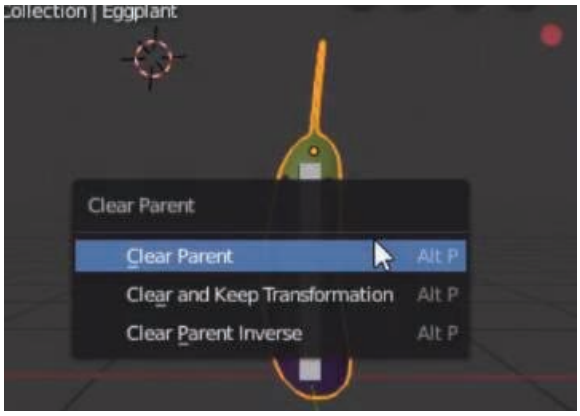
Scale the bendy bone with **Ctrl+Alt+S** if necessary.



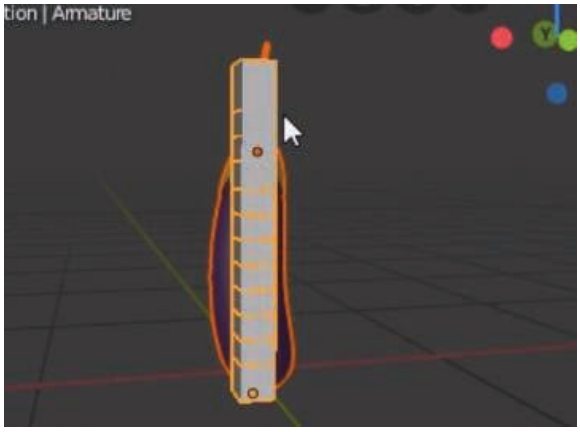
Shift-select object and then the bone, **press Ctrl+P** and choose **Automatic Weights**.



Press Alt+P to clear parent in case you still need to tweak the armature, for example if it's too small like in this example.

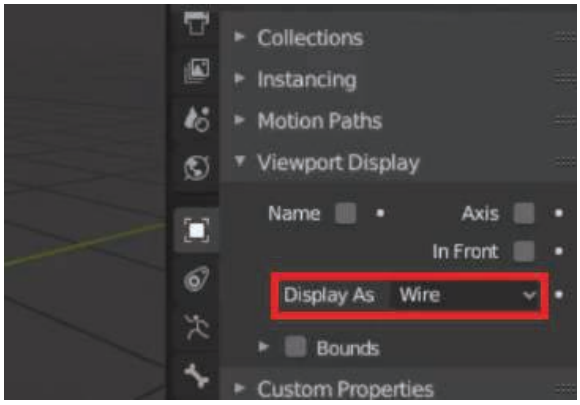


I scale up the bone here to fit the whole eggplant model better.



We can set the bones to **Wireframe** to make them a bit less noticeable.

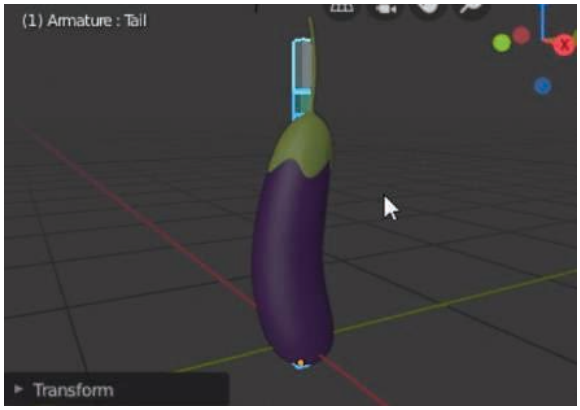
Previously it was set to **In Front** so we could see it better, but we can turn that off now.



You can still make the bone thinner with **Ctrl+Alt+S**.

Shift-select object and then the bone, **press Ctrl+P** and choose **Automatic Weights** (in case you unparented it to do more tweaks).

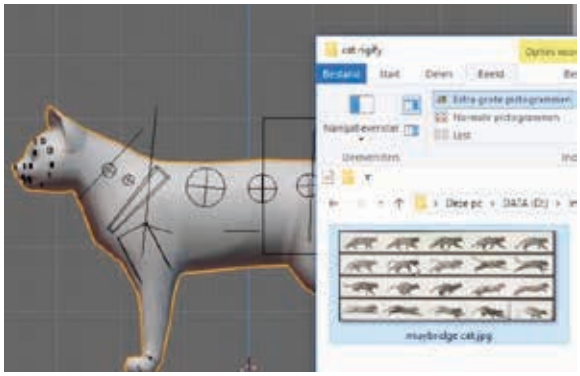
Now you can pose the eggplant and animate it by keyframing poses at different moments in time.



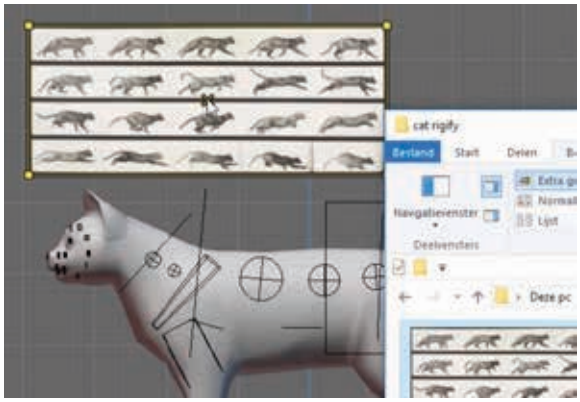
Reference for animating animals

A good source of reference for animal movements is the work of Eadweard Muybridge.

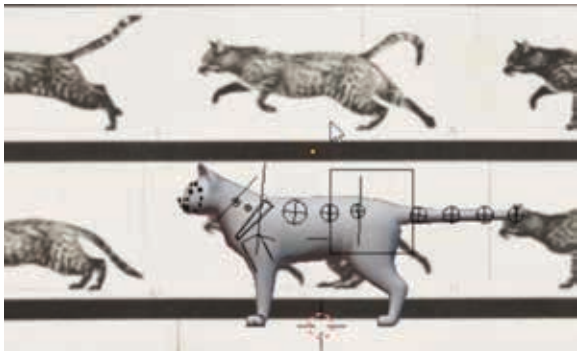
You can find many of his photo series online. Drag and drop the image in Blender.



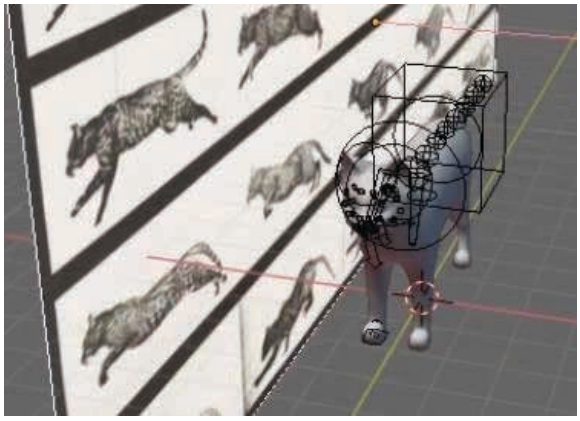
This creates an Empty object with an image on it, which you can move and scale.



To flip the image horizontally **press S+X and -1**.



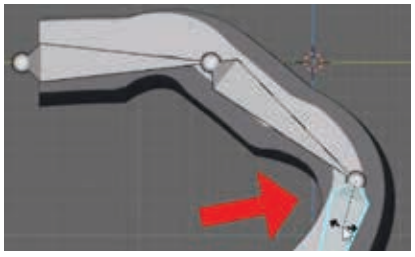
Move it back away from the camera and object so its not in the way when animating.

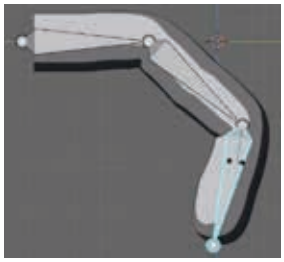


Have fun animating!

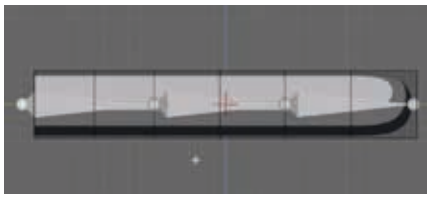
Better rigging with extra edge loops

Let's have a look at proper topology for parts of the object where we need to bend it for animation, like this finger.

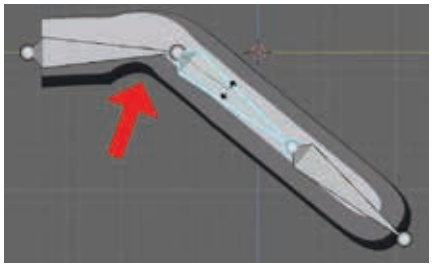


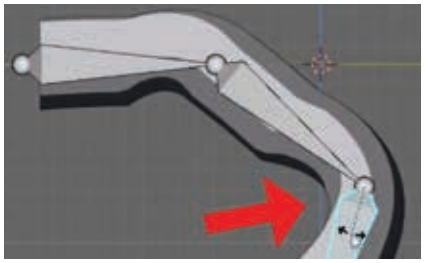


This finger has evenly distributed edge loops. That's good, right?



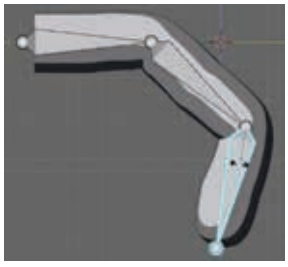
That bending doesn't look so good though.





In fact you really need more edge loops in places where the finger bends. Double-click on an edge loop, **press Ctrl+B** to Bevel it and **scroll the middle mouse wheel** up just a click to add a third loop in the middle.

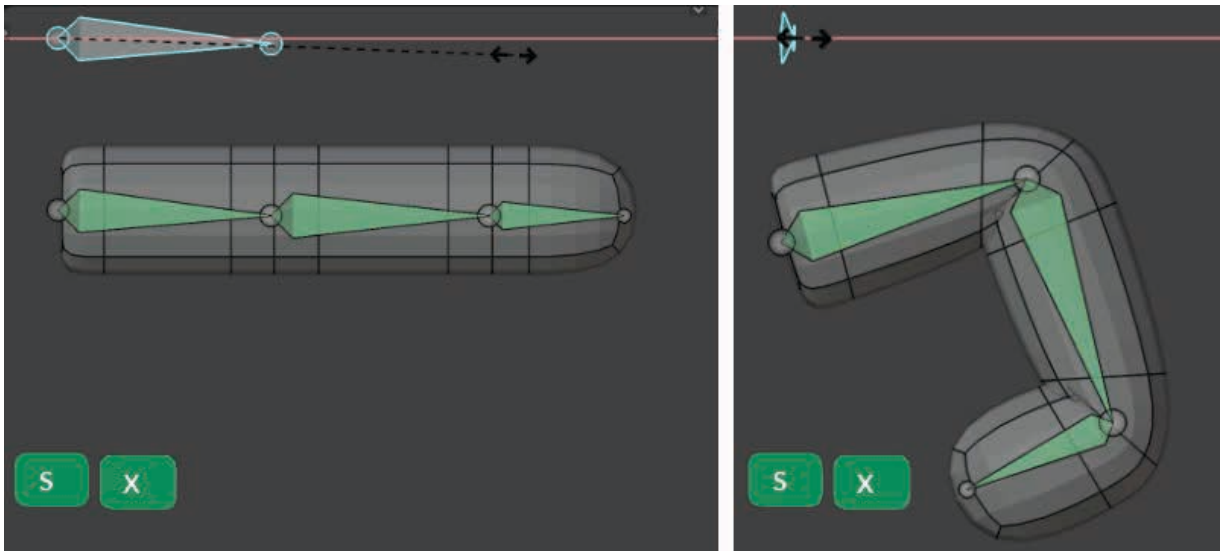
Much better →



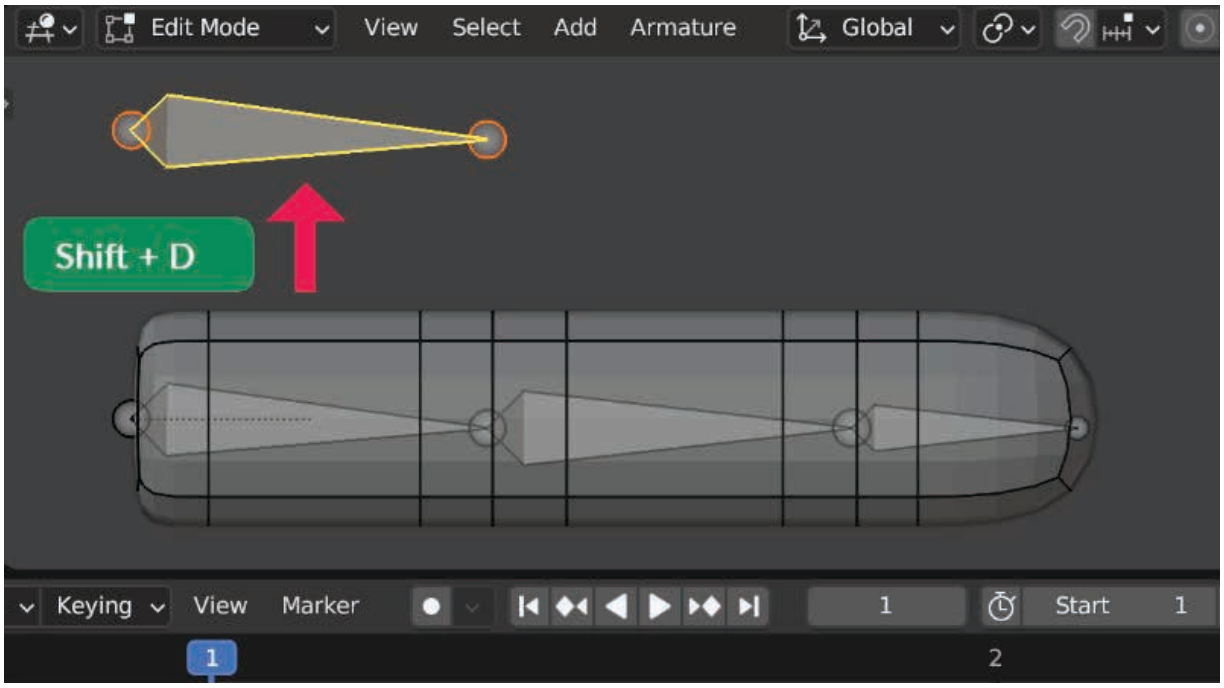
Action Bone constraint

Instead of always posing each individual bone, we can set up control bones that control preset animations of other bones.

This can be convenient when animating fingers and even entire hands. In the example below the single bone at the top controls the three finger bones to make a finger closing gesture.

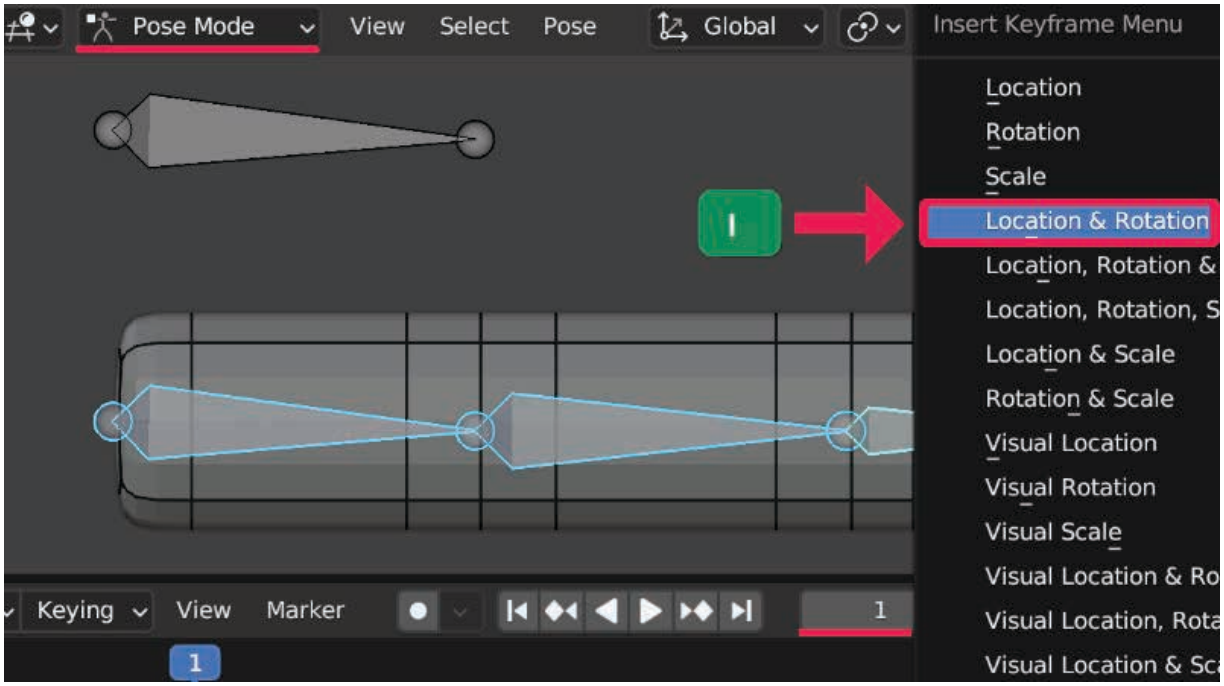


Go to Edit Mode. select and duplicate one bone by **pressing Shift+D** and moving it up. Give it a clear name like **Bone_Controller**.

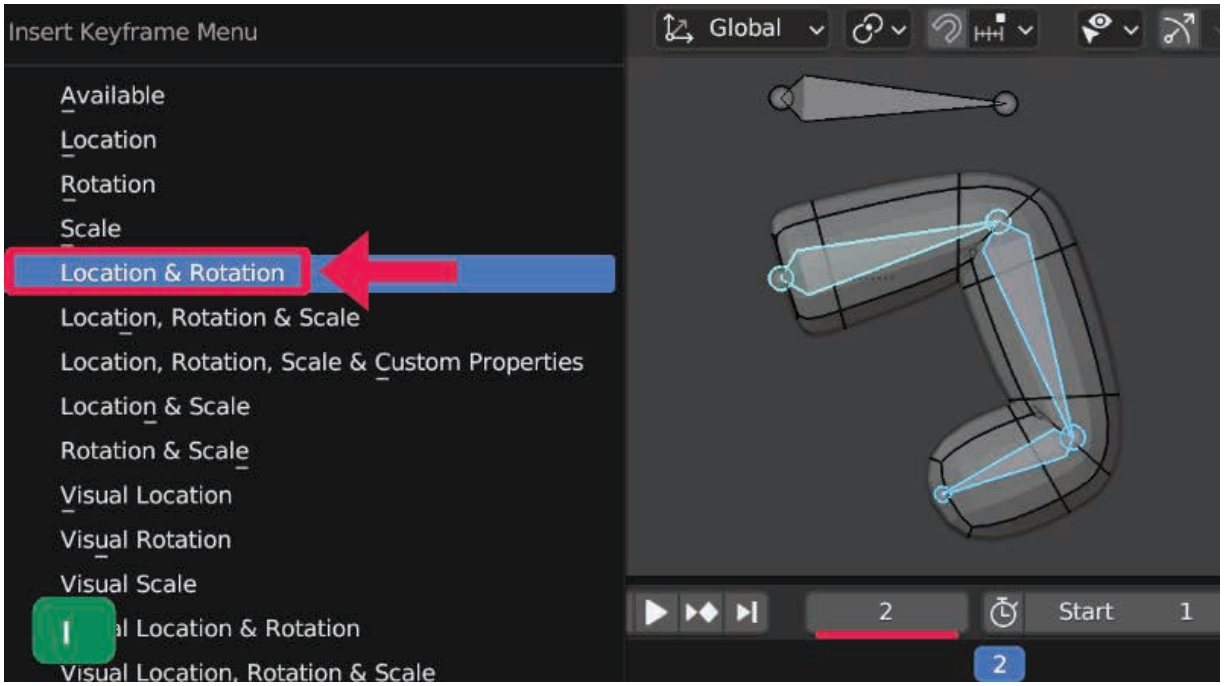


In **Pose Mode** create a keyframe for the first pose of the finger bones.

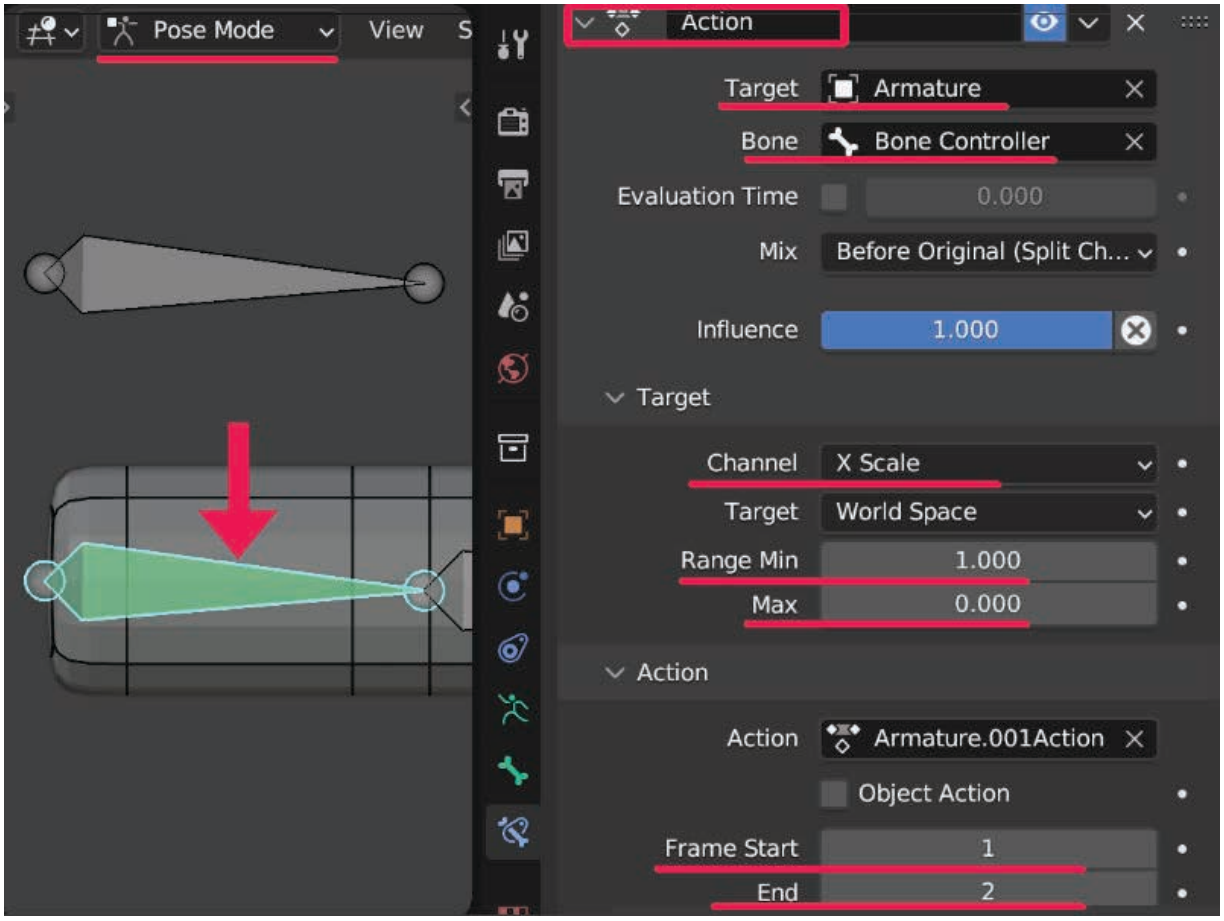
Press i to set the keyframe, choose **Location & Rotation** on **frame 1** of the Timeline.



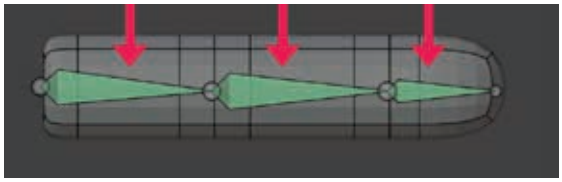
On **frame 2** create the final pose of the finger bones and make a keyframe for that by **pressing i** and choosing **Location & Rotation** again. So now you have keyframes on frame 1 and 2 of the Timeline.



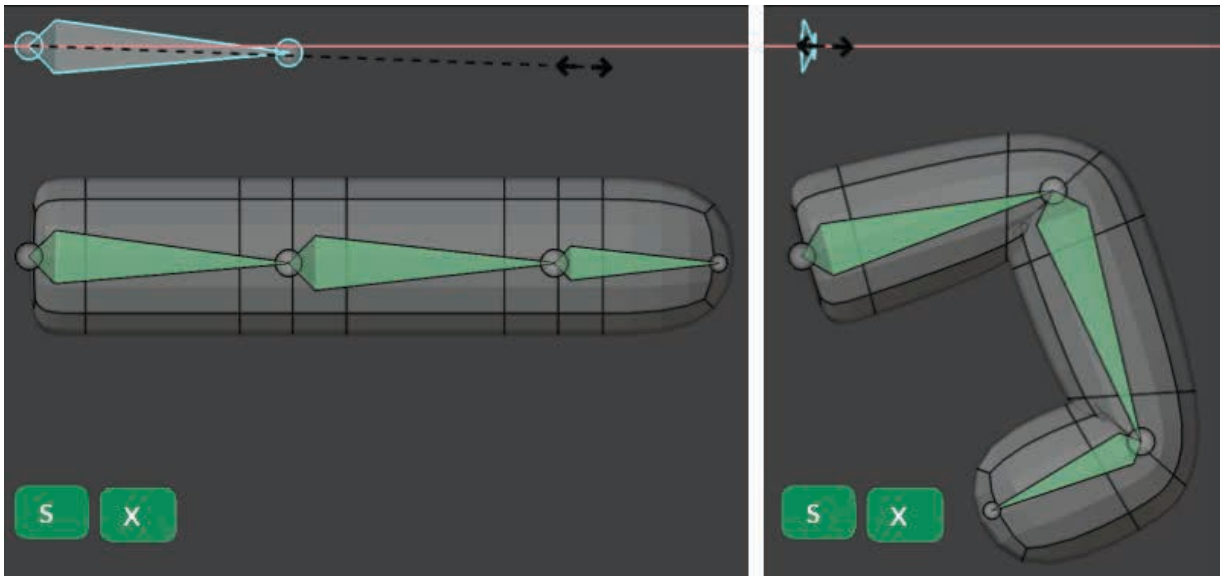
In **Pose Mode** select the first bone of the finger. Add a **Bone constraint: Action** for the first bone of the finger. Use the settings shown below. Now the Controller Bone controls that Action of the finger Bone.



Add the Bone Action Constraints to the other 2 finger bones as well, with the same settings.



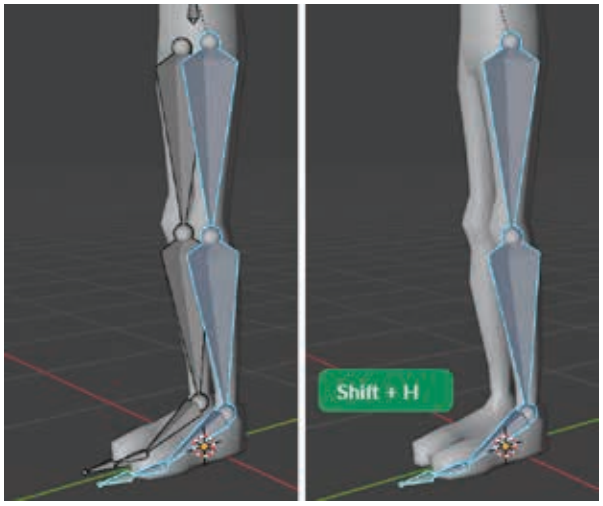
Scaling the Controller_Bone by selecting it and **pressing S** now activates the keyframed poses of the finger.

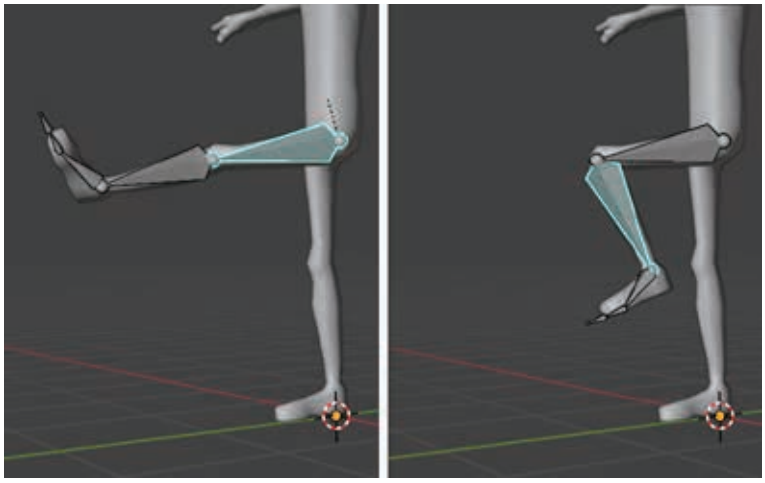


Click here to return to the index.

Auto IK (Inverse Kinematics)

To temporarily isolate the leg bones, select them in **Pose Mode** and **press Shift+H**.

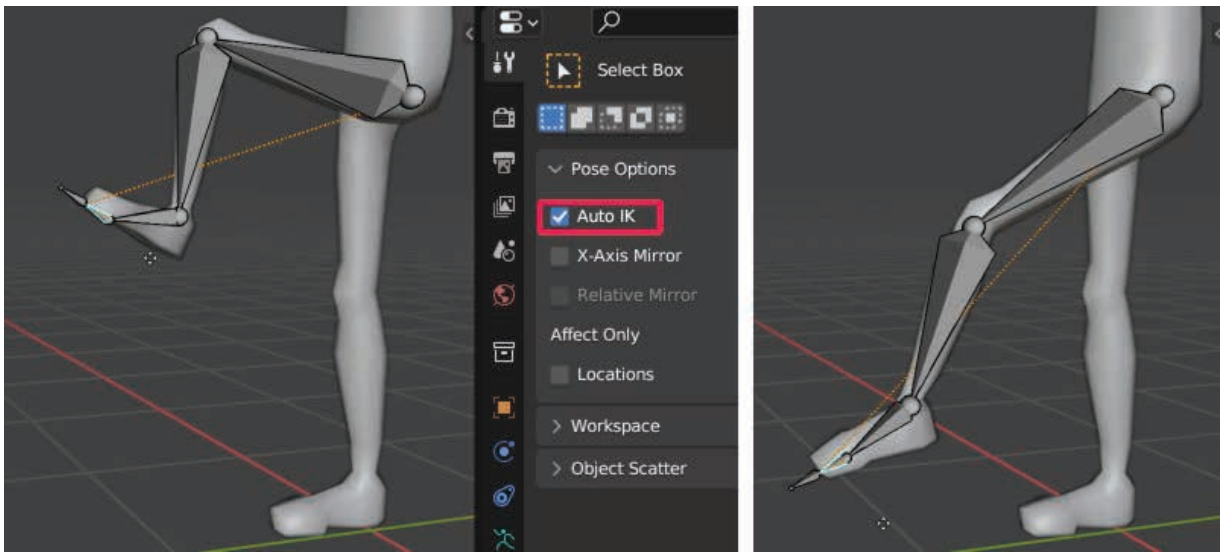




With a manual rig, its set to FK (Forward kinematics) by default.

You can rotate the bones by **pressing R**.

However you can use IK (inverse kinematics) by turning on **Auto IK**. Use Auto IK by moving (**G**) the bones and FK by rotating (**R**) the bones.



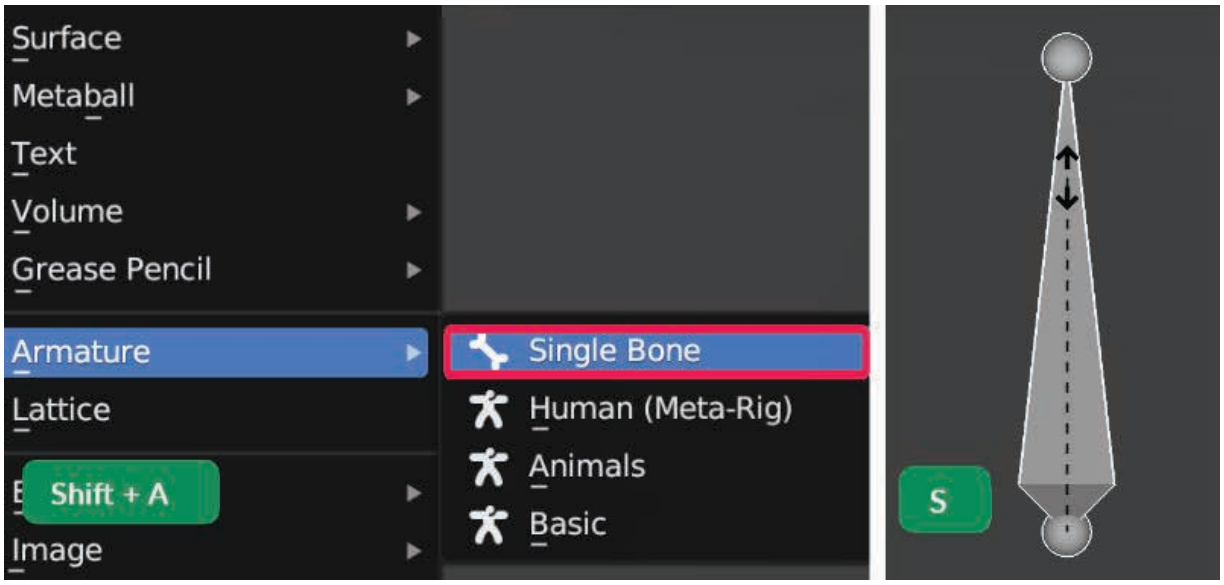
Click here to return to the index.

Manually Rigging Legs

In this tutorial we'll look at the process of rigging a leg step by step.

Of course add-ons like Rigify can make this process easier, but it's good to do this at least once, so you understand the point of using a **Pole target**, for example.

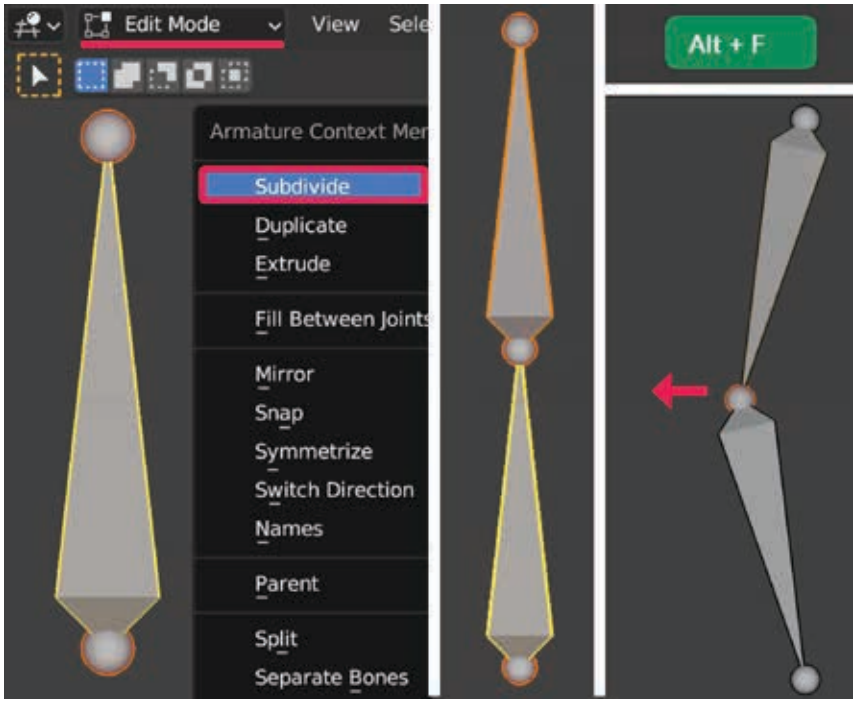
Add a single bone, scale it up.



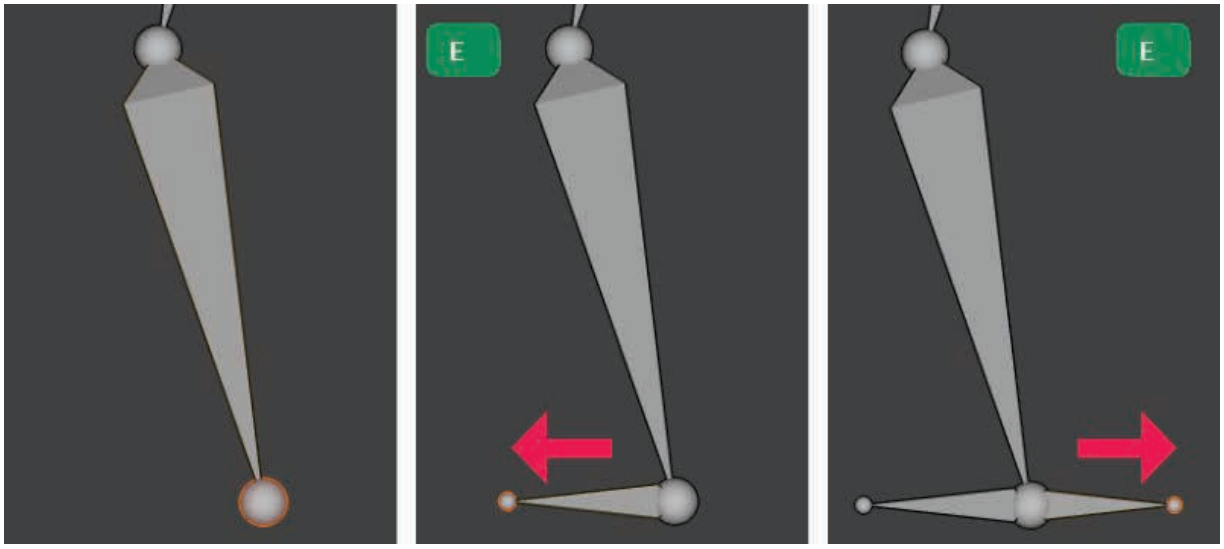
In Edit Mode right-click and subdivide it.

flip the direction of the bones by **pressing Alt+F**.

Move the "knee" so it bends a bit.



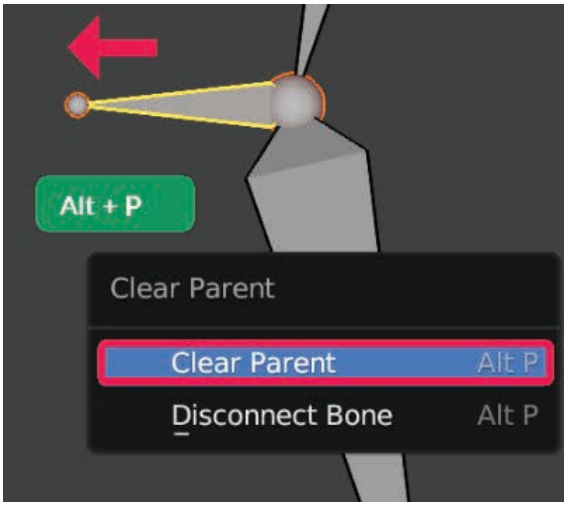
Extrude the "foot" bone. Then extrude another bone in the opposite direction.



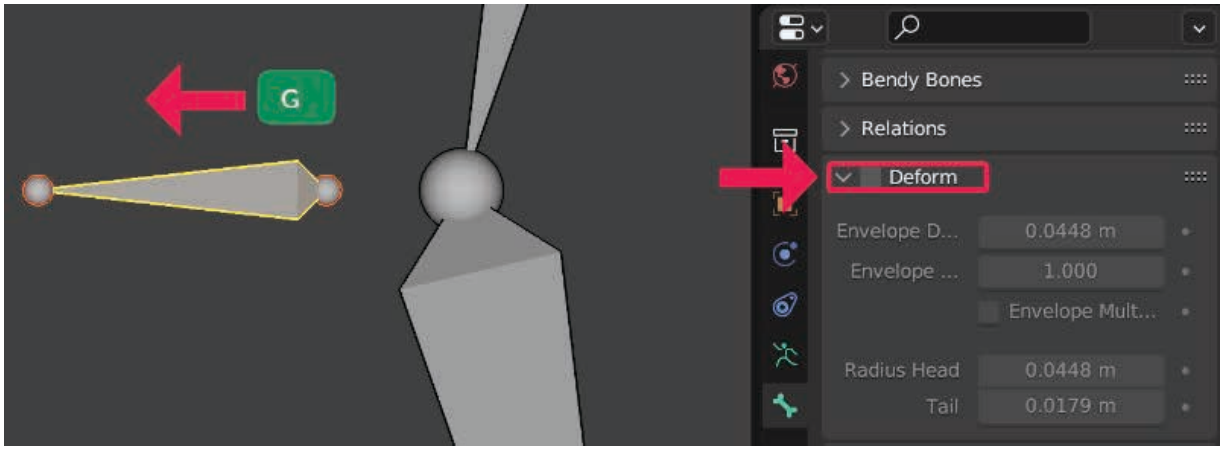
Extrude a bone from the knee position.

Unparent it by pressing **Alt+P** > **Clear Parent**.

Move this bone away from the knee.

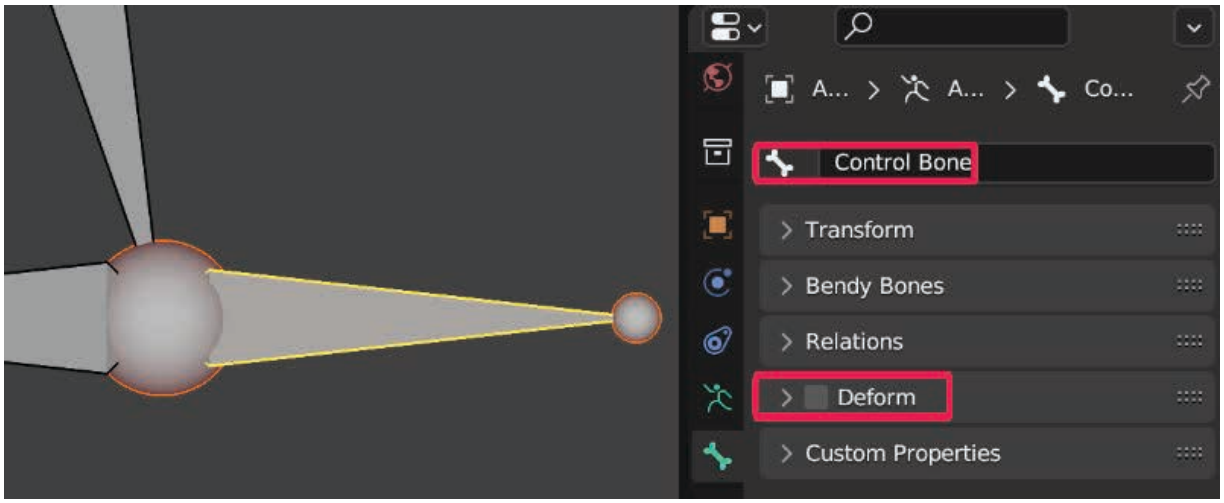


Turn off **Deform**. That means that this bone won't deform any mesh.



Name this bone the "**Pole Target**".

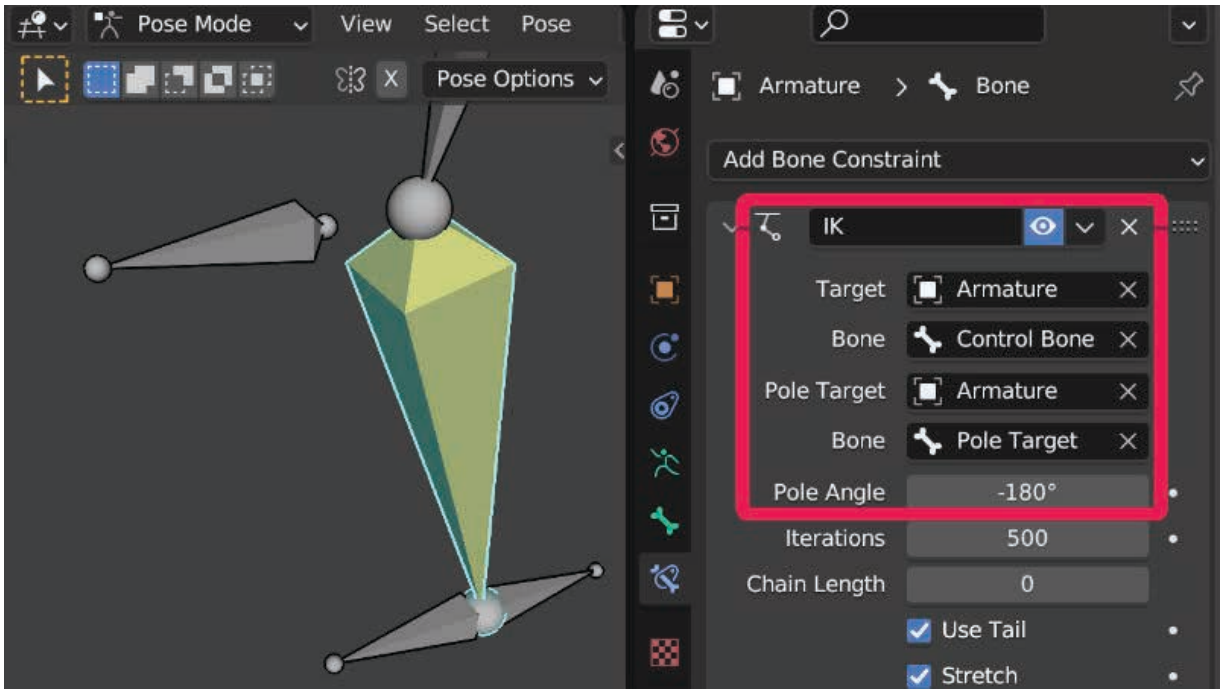
Also turn off Deform for this bone. Name this bone the "**Control Bone**".



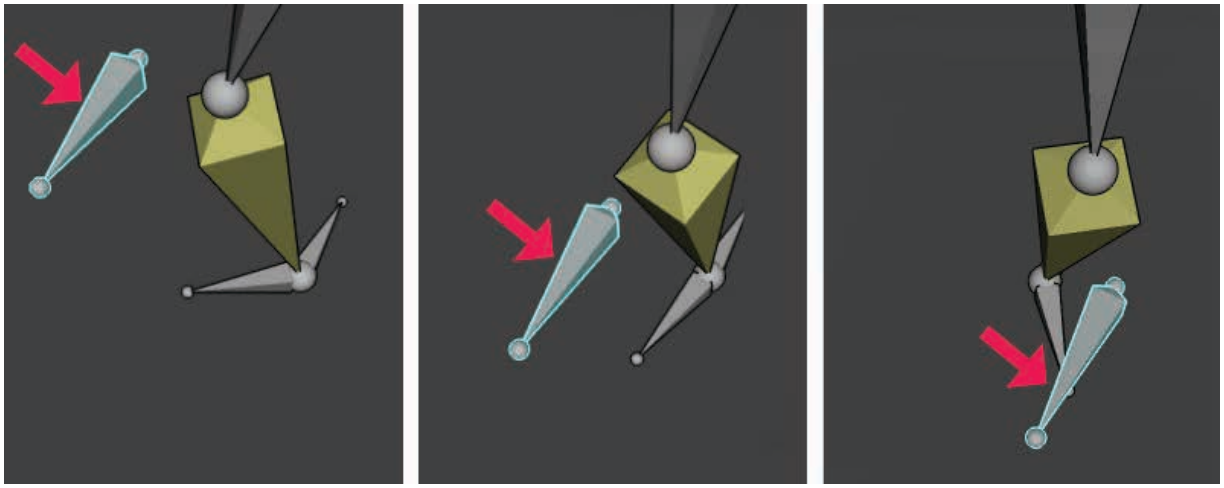
Add a **Bone Constraint: IK** (Inverse Kinematics) to this Bone.

For the **Target** choose the **Armature** and the "**Control Bone**".
For the **Pole Target** choose the **Armature** and the "**Pole Bone**".

You may have to adjust the **Pole Angle** value to rotate the leg correctly.
In this case I had to rotate it -180°.

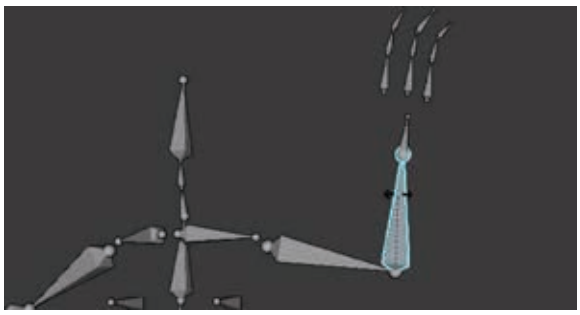


Now you have IK control over the leg and the knees will point to the Pole target.

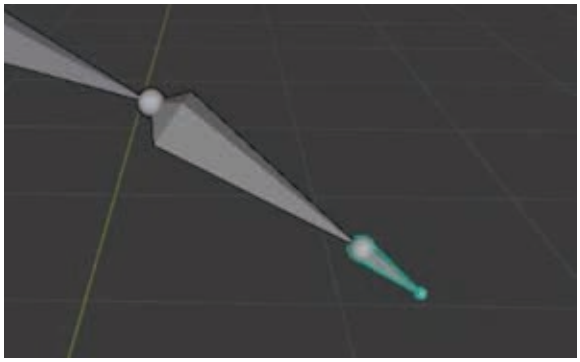


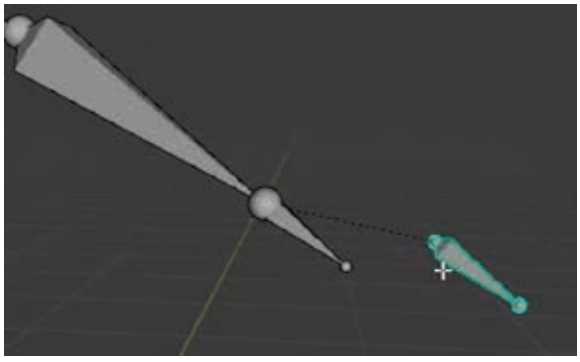
Adding extra bones

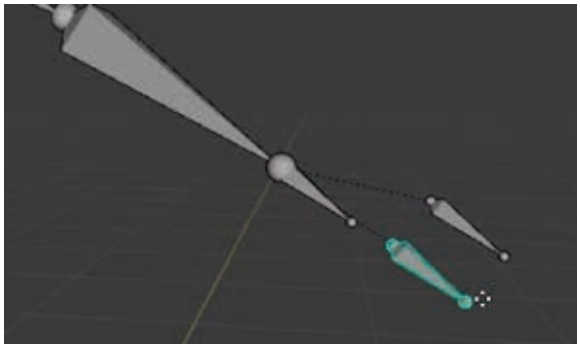
In this tutorial we'll look at how to add more bones to an existing armature.

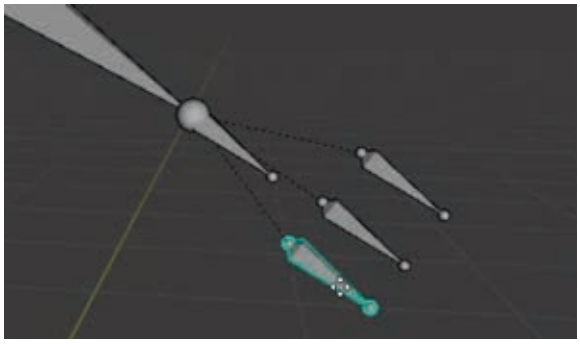


In Edit Mode you can duplicate bones with **Shift+D**.

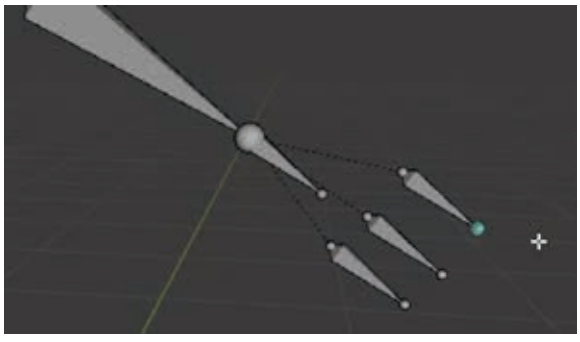


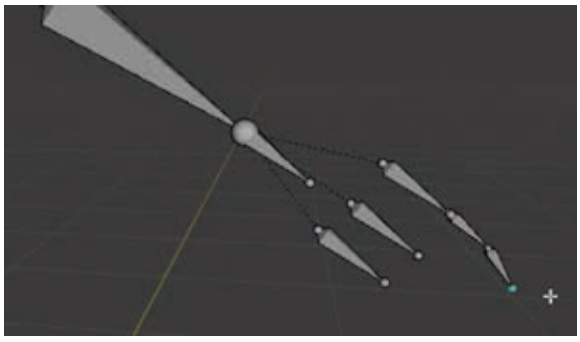




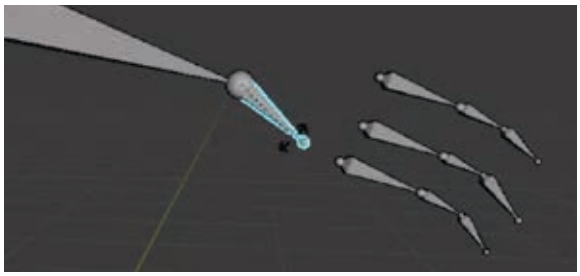


You can also extrude by **pressing E**.



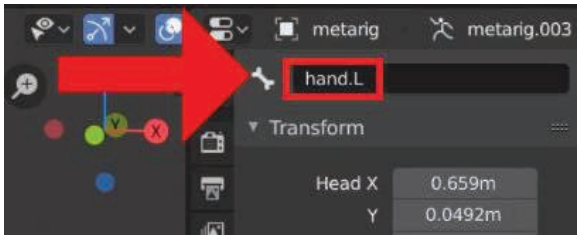


In **Pose Mode**, notice how the hand bone doesn't move the new finger bones.

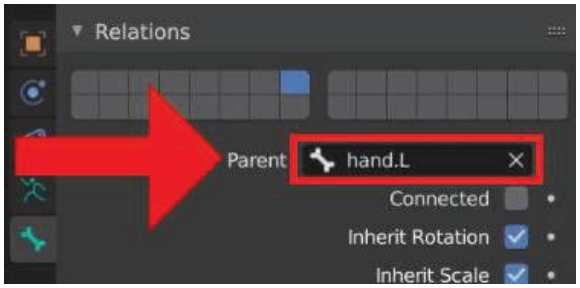


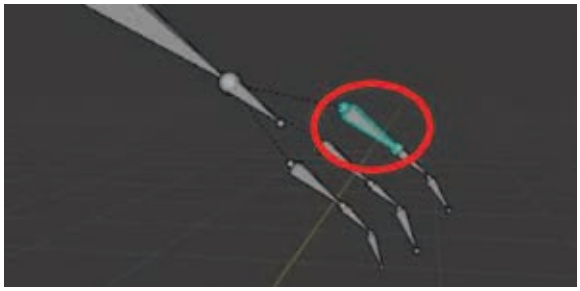
That's because the relations between the bones are currently not set up correctly.

Lets fix it. In Edit Mode hover the cursor over the hand bone Name field (in the Bone tab) and **press Ctrl+C** to copy the name.

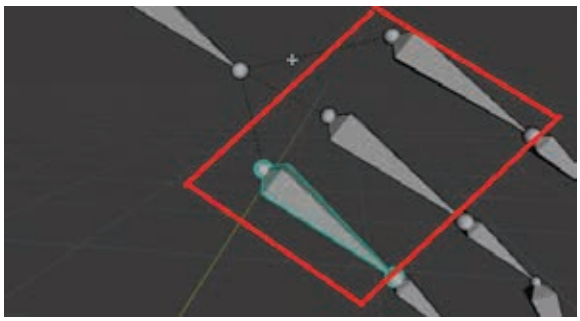


Then hover the cursor over the Parent field of the first fingers, and **press Ctrl+V** (Paste).

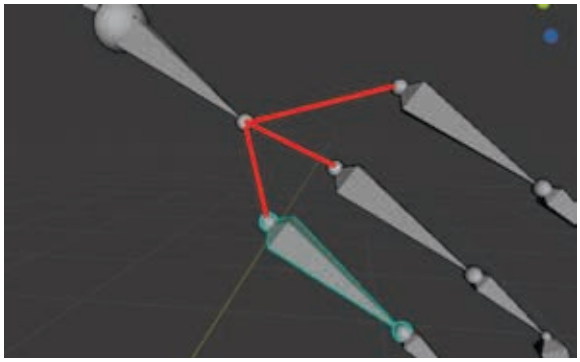


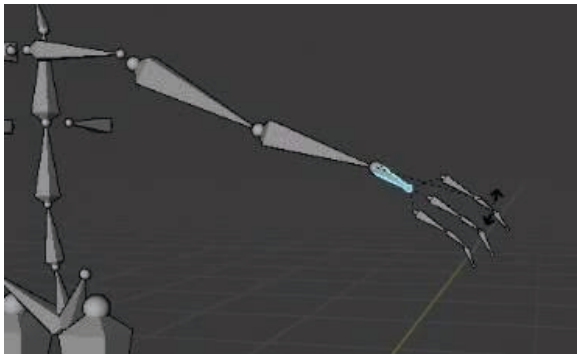


Do this for all three of the first fingers of the hand.



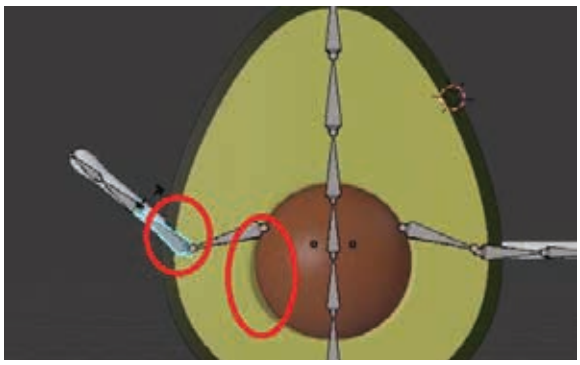
Notice that now the chain of parenting is correct, the hand influences the fingers.



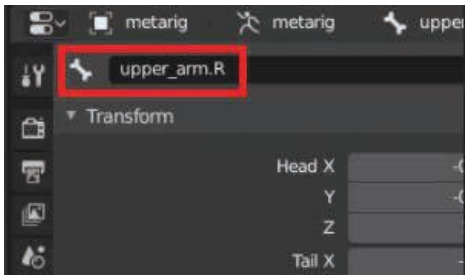


Manually fixing weight paint issues

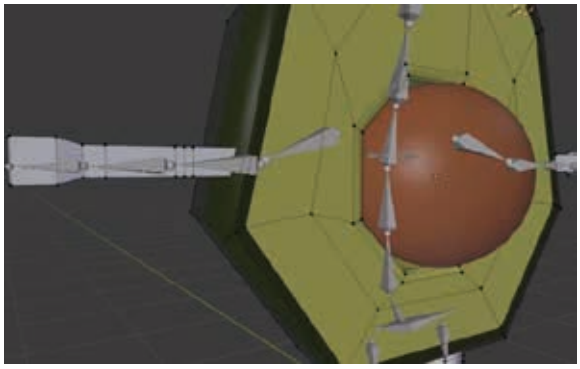
This arm bone is affecting the body, which should not be affected at all.



In **Edit Mode** check the name of that specific bone in the **Bone** tab.

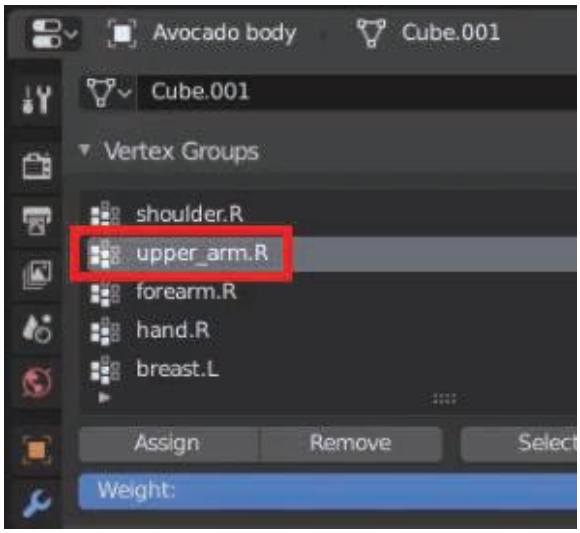


Go to Object mode, select the character mesh and in Edit Mode deselect all by **pressing Alt+A**.

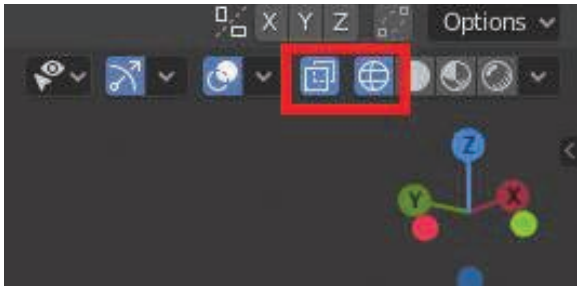


Find the Vertex Group of that bone.

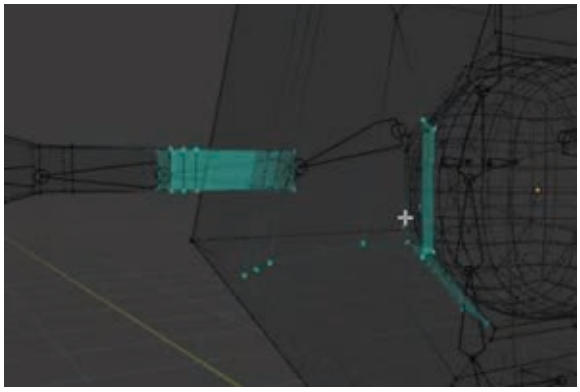
In this case it's called "upper_arm.R". Select the Vertex Group and check it in **Wireframe mode**.



In Edit Mode make sure **Wireframe** and **X-ray** are turned on, so that you can select through the entire model and get the vertices at the back as well.

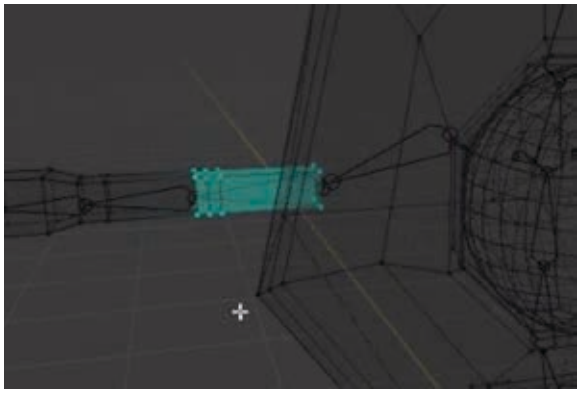


As you can see, though only the arm should be part of the vertex group there are some vertices in the body assigned to it as well.

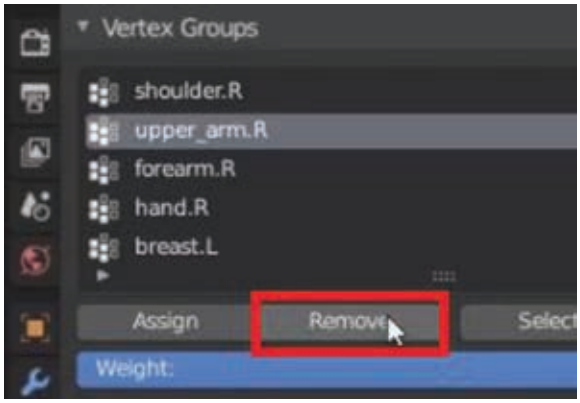


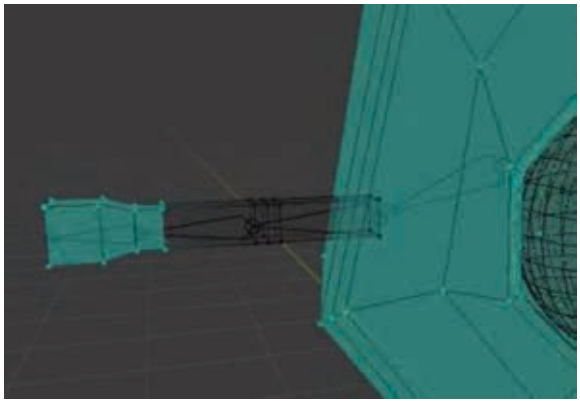
We need to remove these from the vertex group.

Deselect the extra vertices.

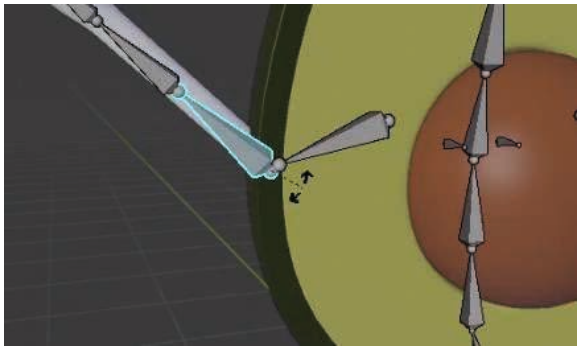


Invert the selection and hit **Remove** to remove them from the bone's vertex group.



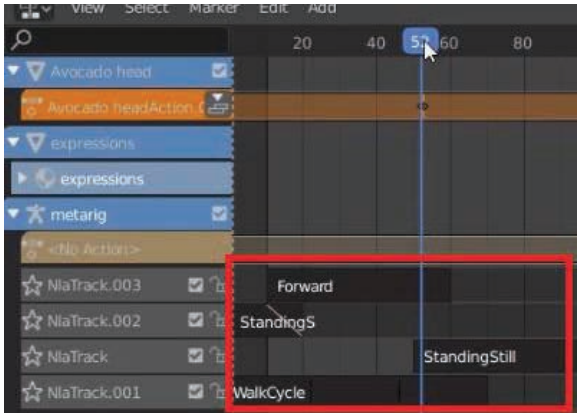


Now the body is not affected by the arm bone.

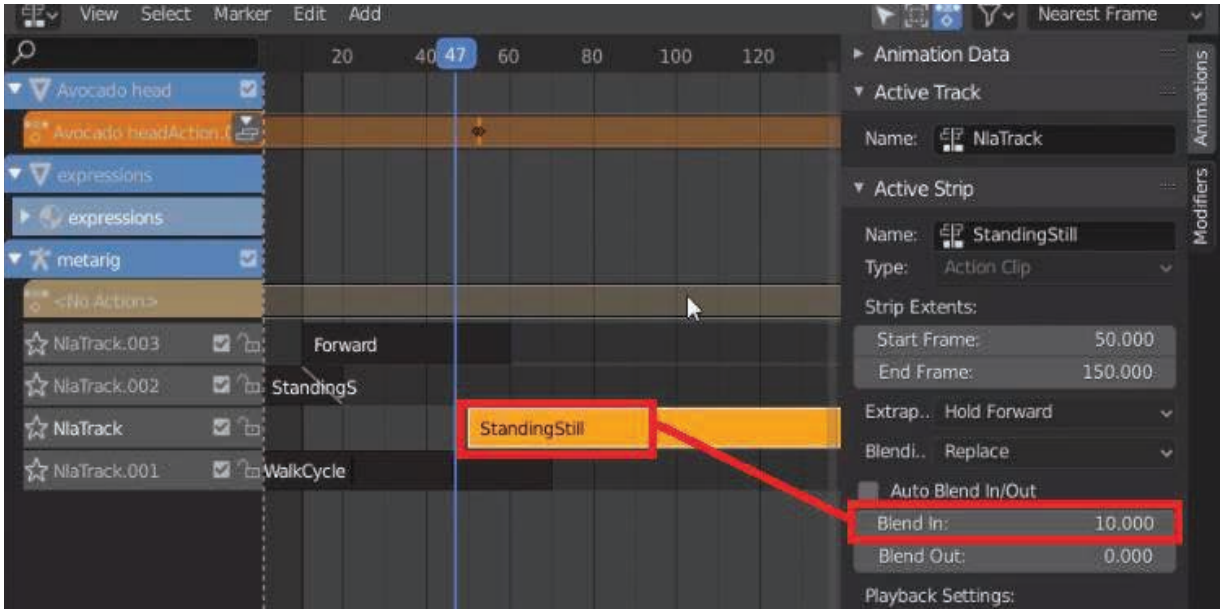


Blending actions

In the **Action Editor**, actions above other actions get priority.



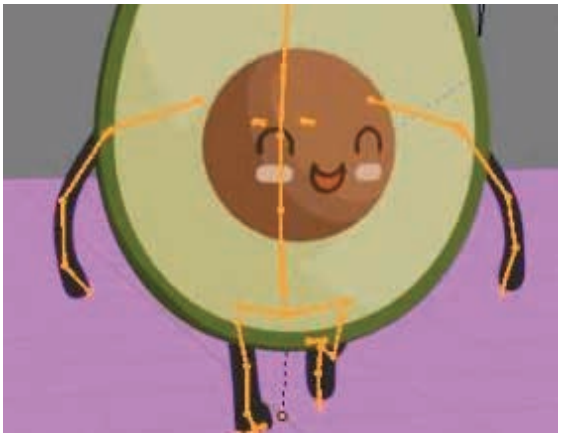
To "blend" between the actions, set a Blend in or Blend out value. For example here I set a Blend value of 10.



The number (10 in this example) is the amount of frames of the blending.

Notice the “fade” line added to the beginning of the action.

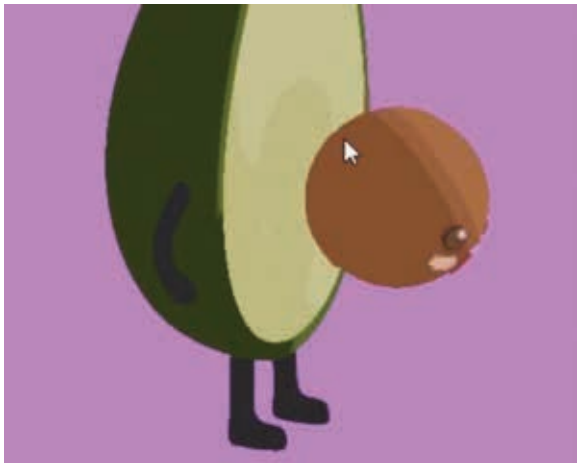
Now the animation will fade from the Walk Cycle action to the Standing Still action, so the character will be walking and then gradually (over 10 frames) come to a standstill.

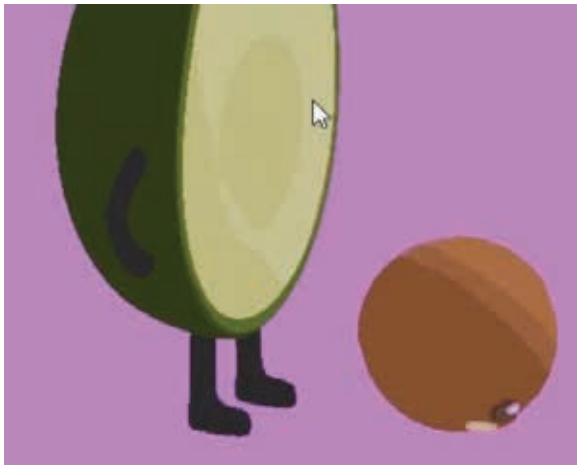


Click here to return to the index.

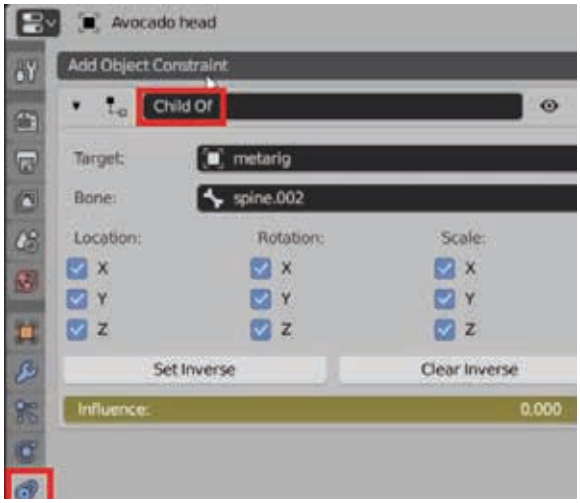
Constraints and Physics

How do you parent an object and then switch to a Rigid Body simulation?





The Avocado's head object has a "**Child of**" Object constraint which parents it to the spine bone.

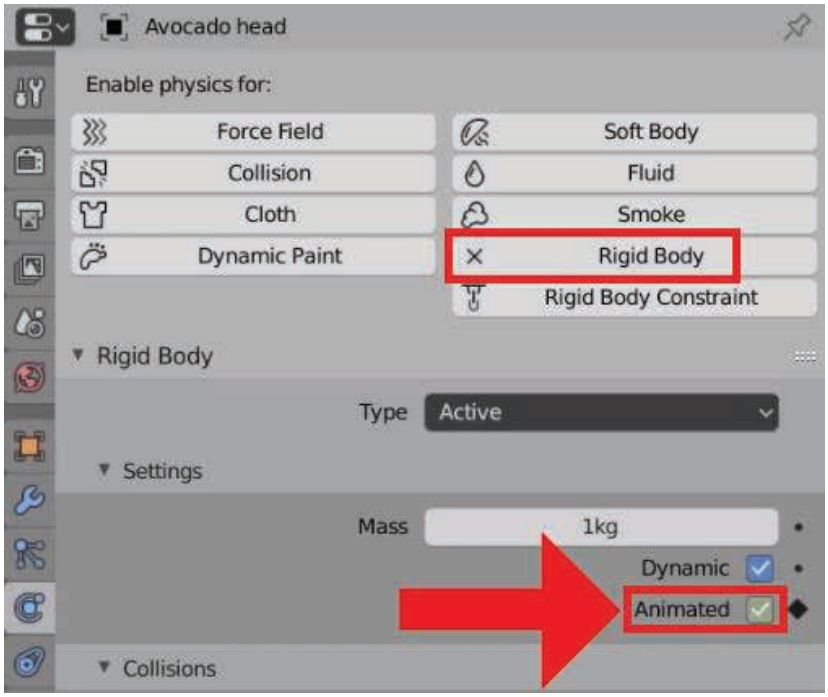


The trick is to change the influence to 0 at the right moment. When the influence of the Child of constraint is at 1, it is 100% parented to the bone. Once the influence is 0, it is 0% parented to the bone and it will fall out of the avocado body, provided there is a rigid body simulation.

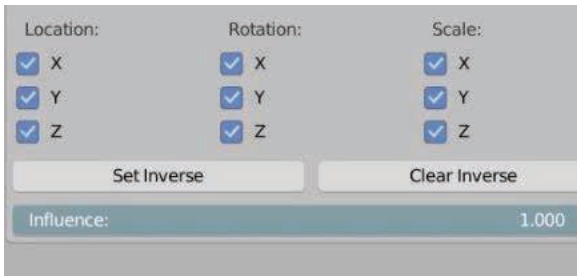
Let's look at it step by step.

First of all, the avocado head is an **Active Rigid body**.

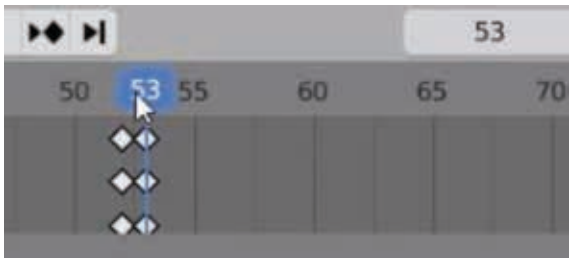
Because it is set to "animated" it doesn't behave like a rigid body. Otherwise it would fall down immediately.

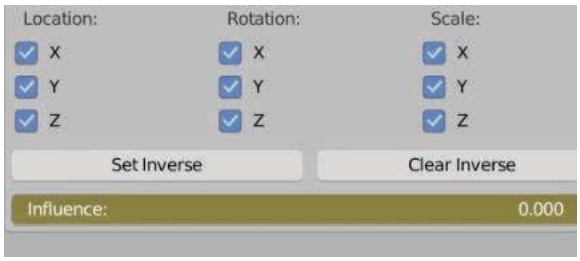


The influence of the **Child of constraint** is keyframed. At frame 52 it is set to **1**.

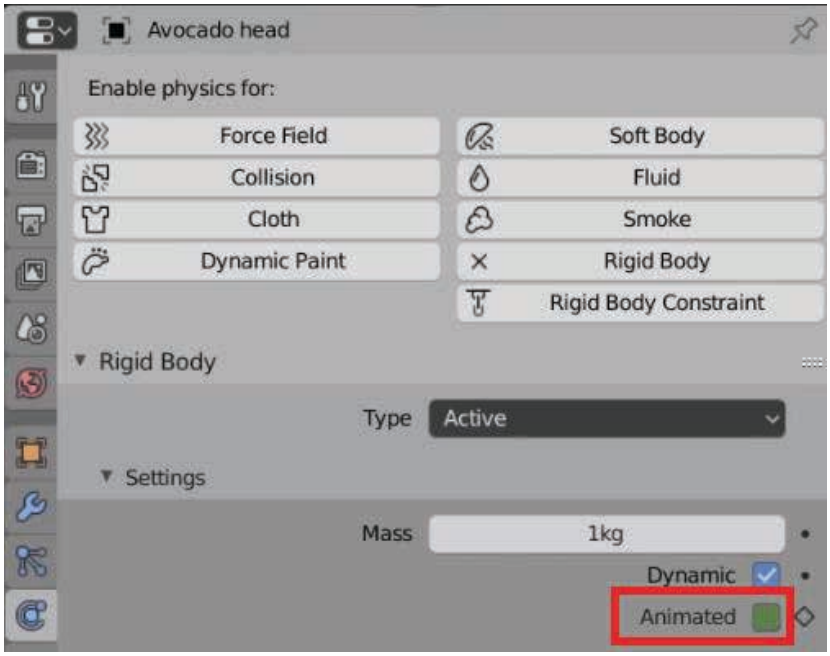


At frame 53 it is set to **0**.





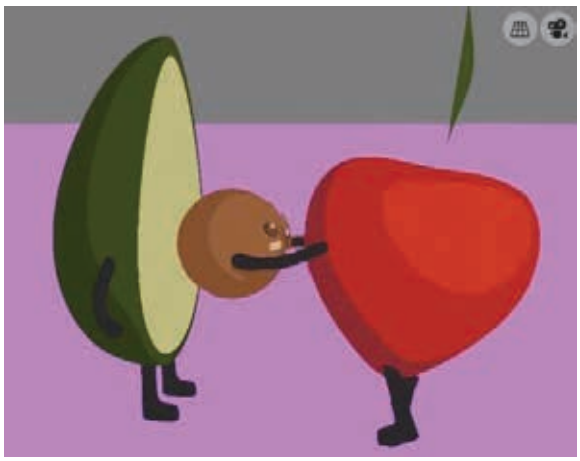
In other words, from frame 53 on the Sphere is no longer parented to anything.
On frame 53, the Rigid Body simulation is keyframed to no longer be "Animated".



Since it is no longer parented and no longer set to Animated as a Rigid Body, it is free to fall down from frame 53 on.

Constraint transfer

How do we transfer a parent relationship from one object to another?



For instance, here an apple picks up the head of the avocado and places it in its body. After that, the head of the avocado should stay parented to its body (or rather, to a bone of its armature).

But while the apple holds the head, the head is parented to a bone in the apple's hand.

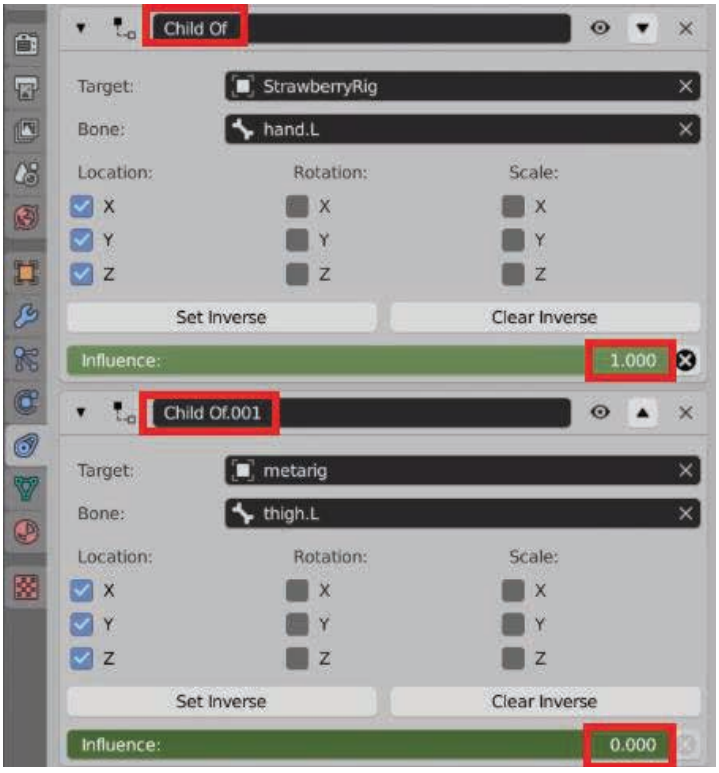
Let's look at this step by step.

Below you see that the head has two **Child Of** object constraints.

The top one parents it to the Strawberry's hand bone ("hand.L"). It is currently set to Influence: 1, so it is completely parented to the hand of the strawberry.

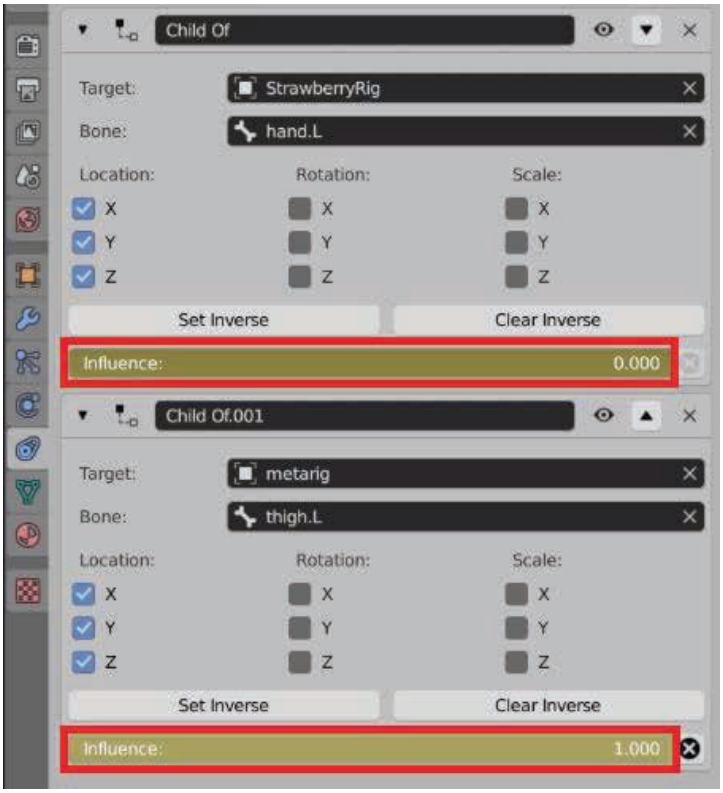
The Child Of constraint below it, parents it to a bone of the avocado ("thigh.L").

However its influence is set to 0, meaning it has no influence and is currently not parented at all.



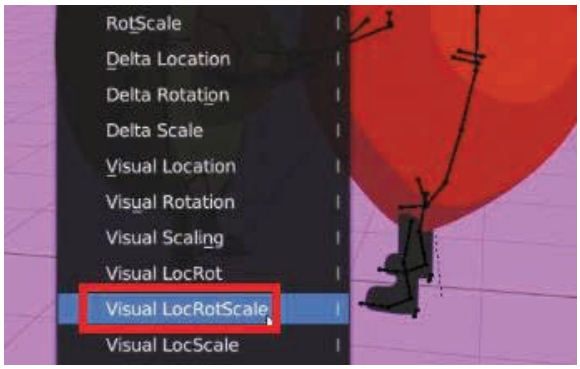
These Influence values are keyframed to change to their opposite values over the course of one frame.

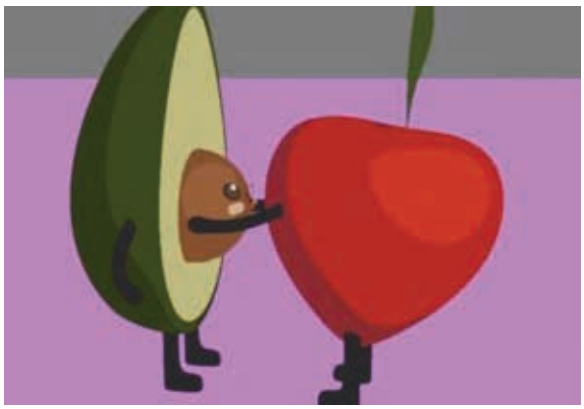
Below you see the values are swapped on the next frame.

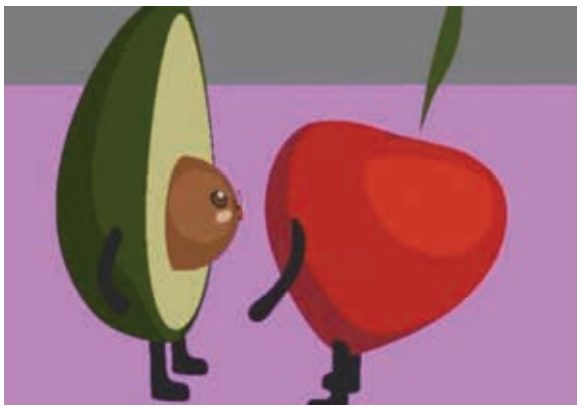


One frame the avocado seed (head) is parented to the strawberry and a frame later it's parented to the avocado.

The final frame when its parented to the strawberry, add a Visual LocRotScale keyframe to the avocado head. This is how you change parenting relationships between objects.

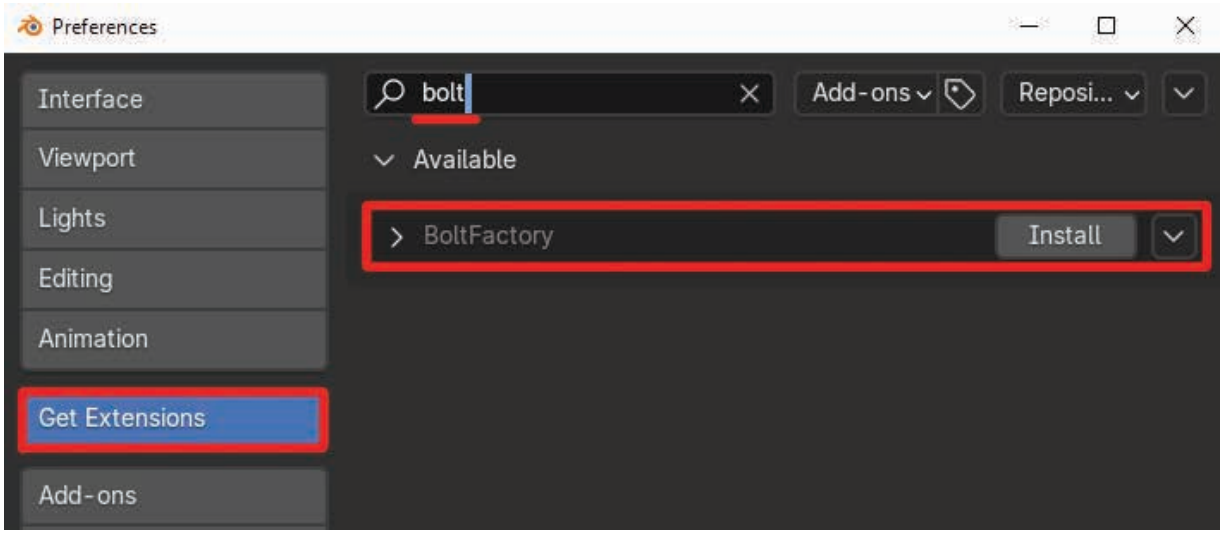




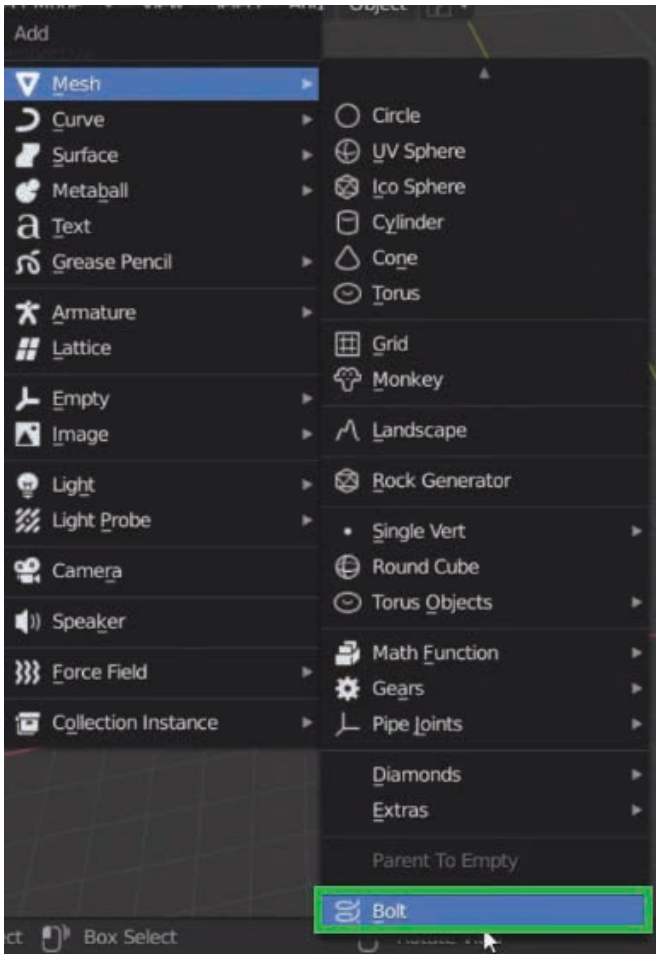


The nuts and bolts of Transformation Constraints

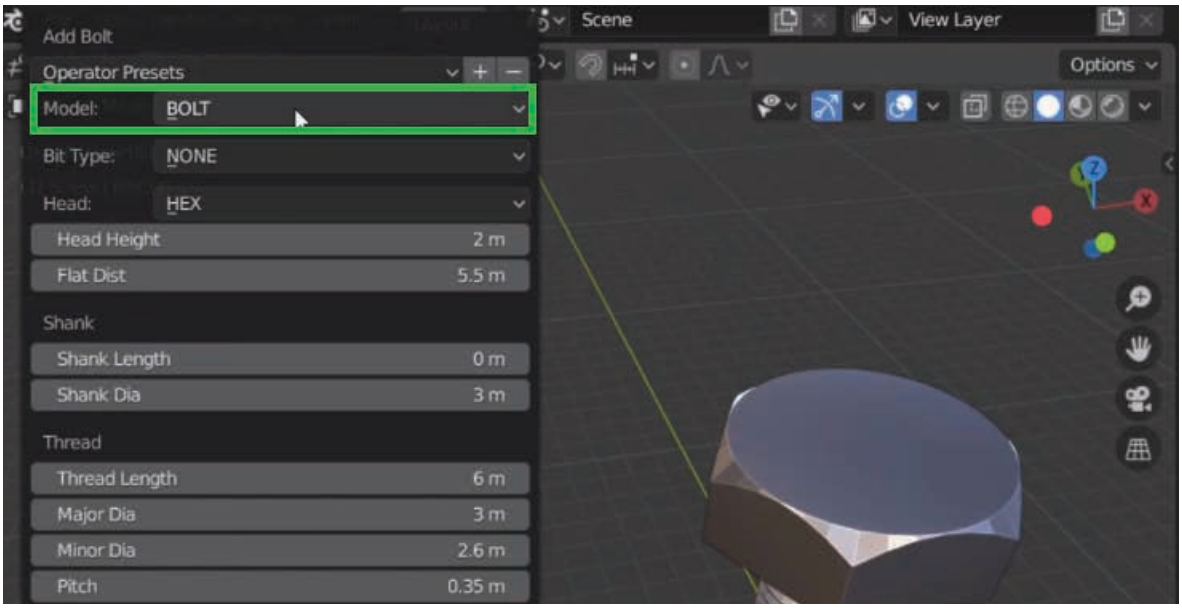
Here's a look at **BoltFactory** (a free add-on that comes with Blender) as well as using the powerful Transformation object constraint to help us create some nuts and bolts animation. First, enable the BoltFactory add-on by installing it in the **Extensions** tab.



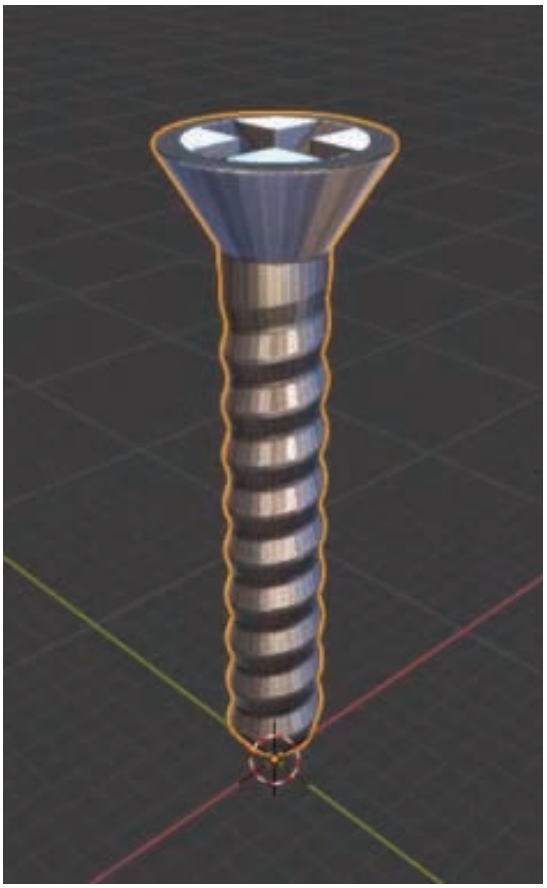
Now you can add nuts and bolts.



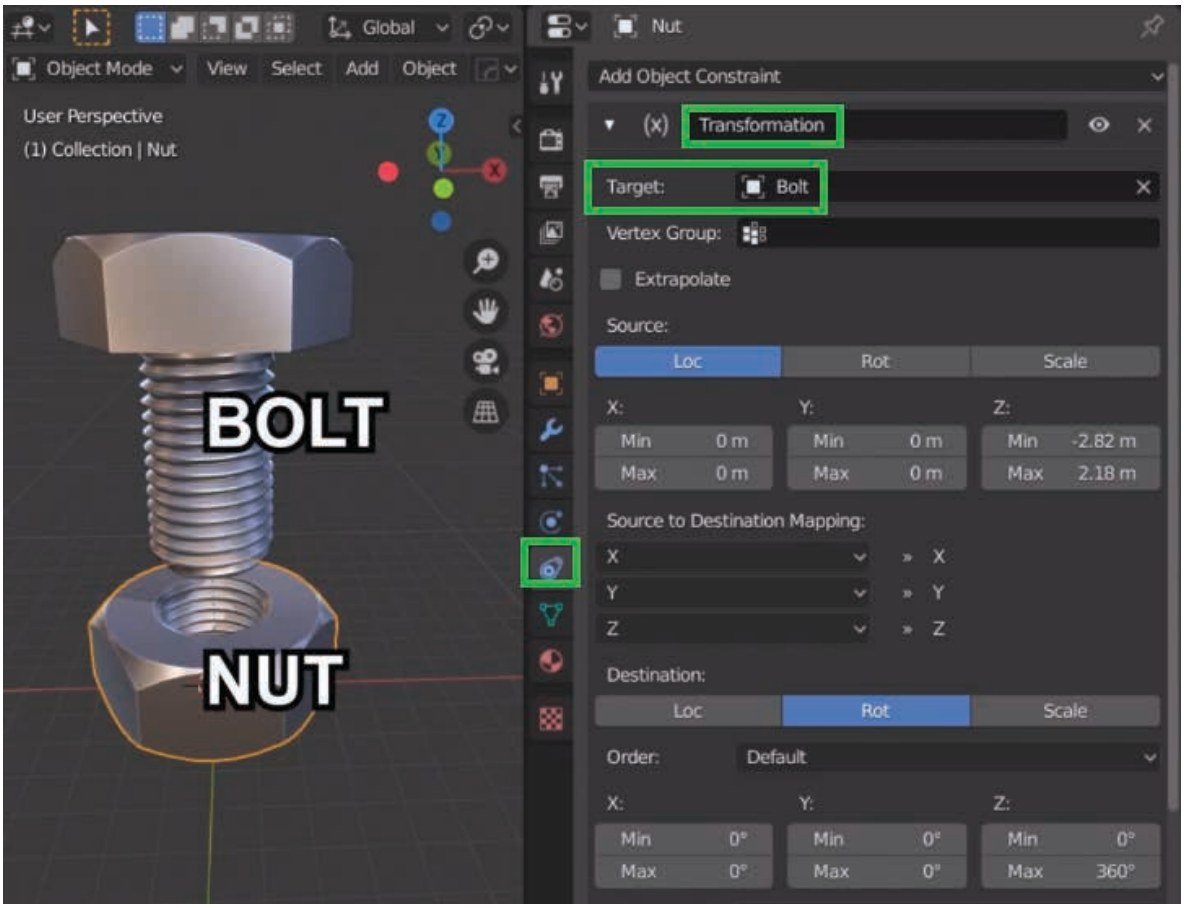
If you accidentally close the settings panel, you can **press F9** to bring it back. You can choose between a **Bolt** or a **Nut**.



To create more of a screw than a bolt set the Bit Type to **Philips**, increase the diameter (**Bit Dia**), set Head to **Counter Sink** and increase the **Pitch** value. You may need to increase the **Thread Length**.

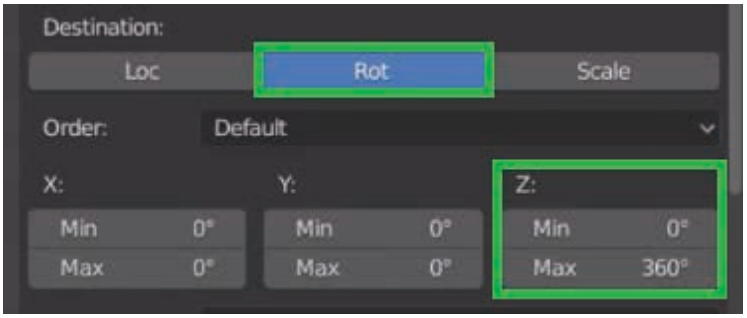


But enough screwing around, let's look at those Constraints. Create a bolt and a nut and add a **Transformation** Object Constraint to the nut. As the **Target** set the Bolt.

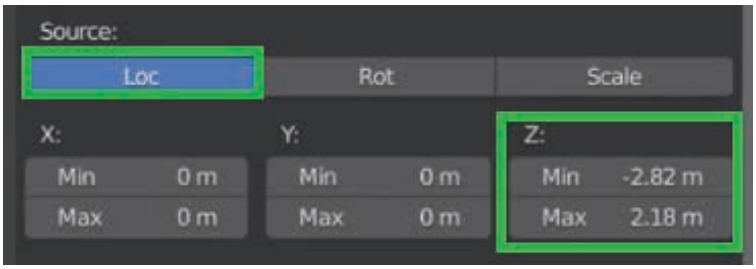


Now we're going to use the Z Location of the source object, the Bolt, to drive the Z Rotation of the Nut.

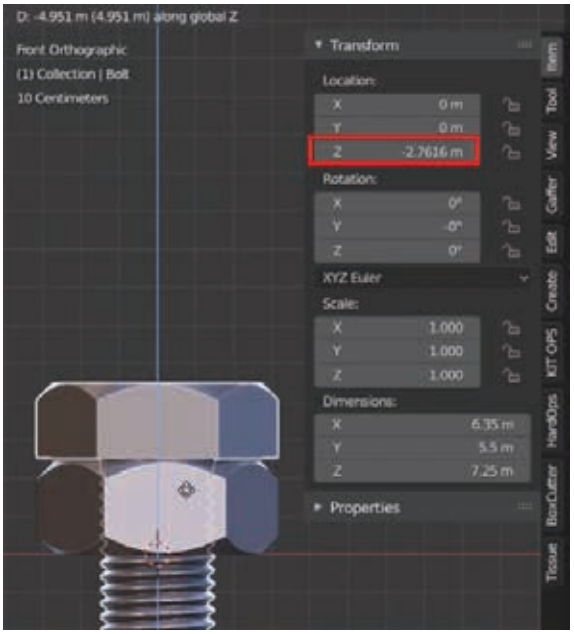
We can set how much rotation the Nut is allowed.

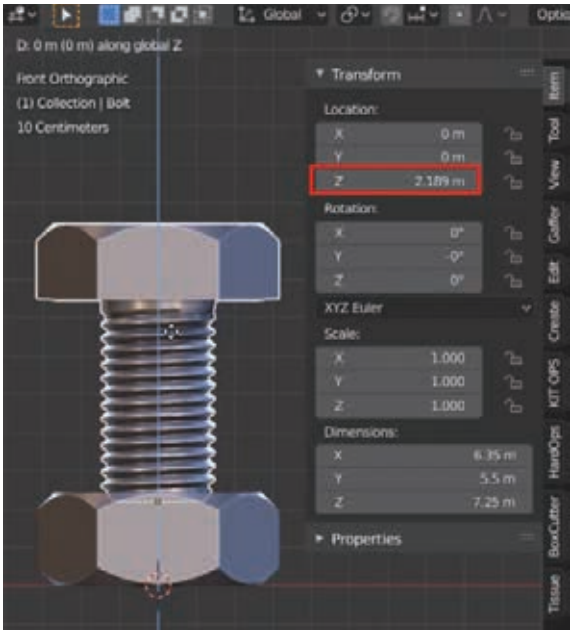


The Z location values of the Bolt determine at what Z location the Nut starts to turn, and when it stops.

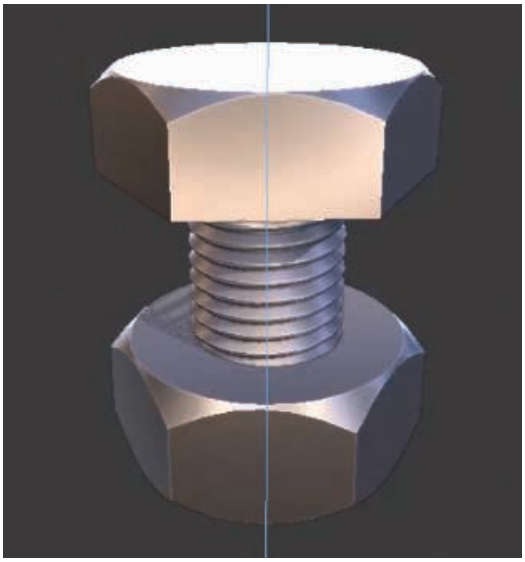


You can copy/paste these minimum and maximum values to the constraint.



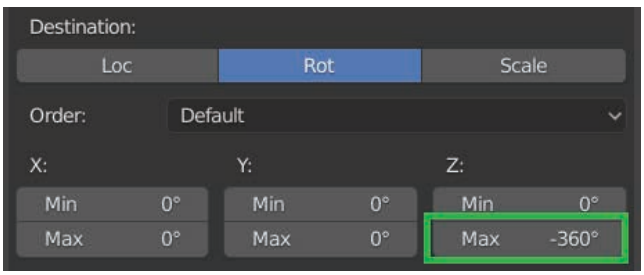


Moving the Bolt now rotates the nut.



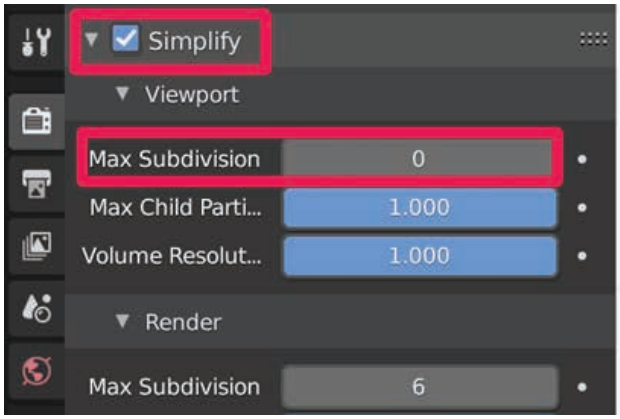
Note: to get the nut to turn in the other direction, add a minus before

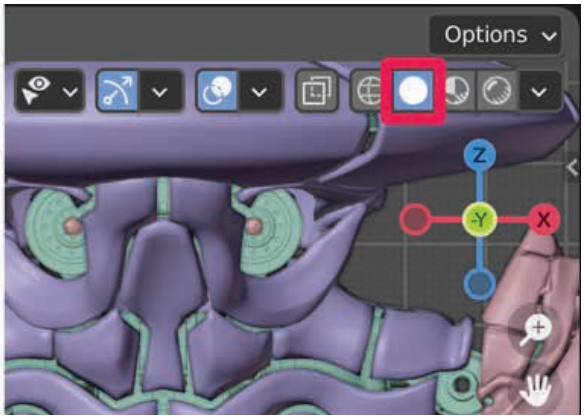
the amount in the Z rotation field:



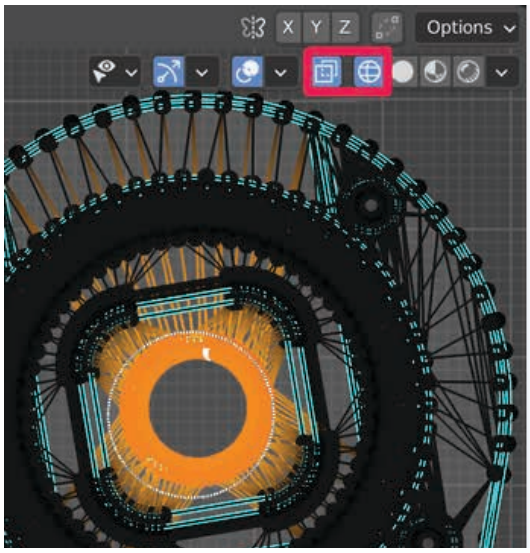
Robot Rigging (Part 1 - The Armature)

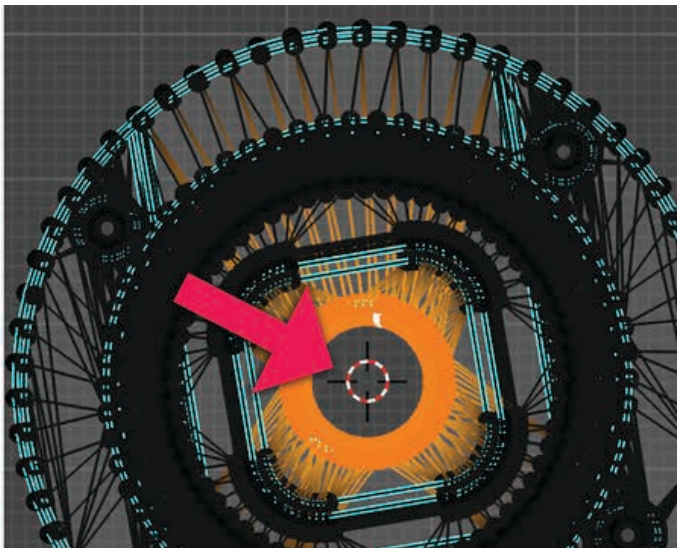
If your robot model is very high poly and/or uses large textures, turn on **Simplify** and set Viewport **Max Subdivision to 0**. To make the model even lighter, use **Solid shading**. This will make working on the rigging more fast and user-friendly.



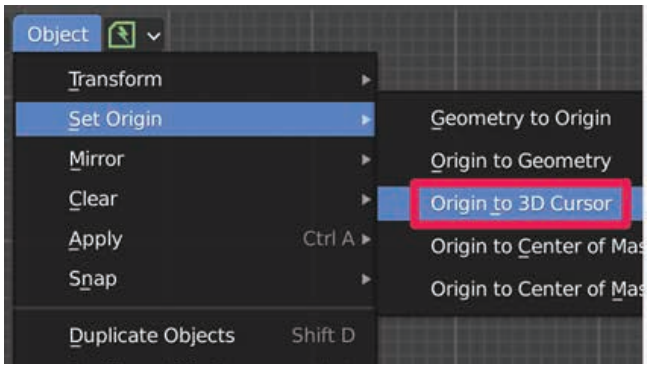


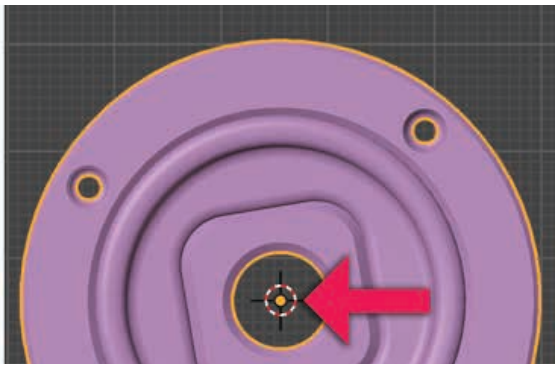
To make sure that all the parts rotate correctly, their pivot points need to be set at the point of rotation. Hide the Outer Shell of the robot. **Press / on the numpad** to isolate an object. Select vertices around the point of rotation in X-Ray mode using Circle Select (C), then **press Shift+S** and choose **Cursor to Selected**. This moves the 3D Cursor to where the Pivot Point should be.



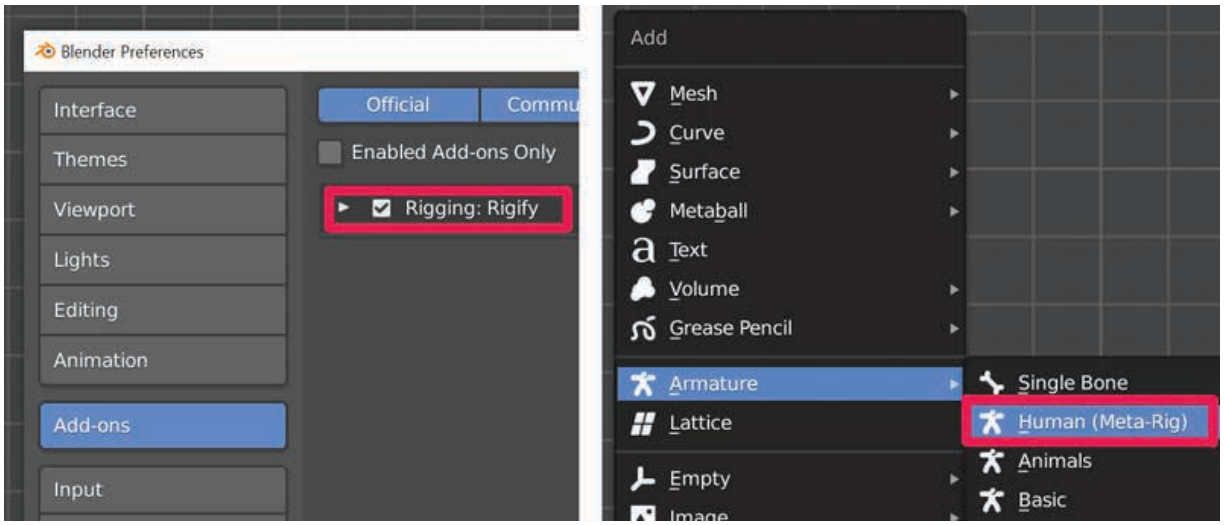


Then in Object Mode go to **Object > Set Origin > Origin to 3D Cursor**. Press / on the numpad again to exit isolation mode. Do this for all the objects.

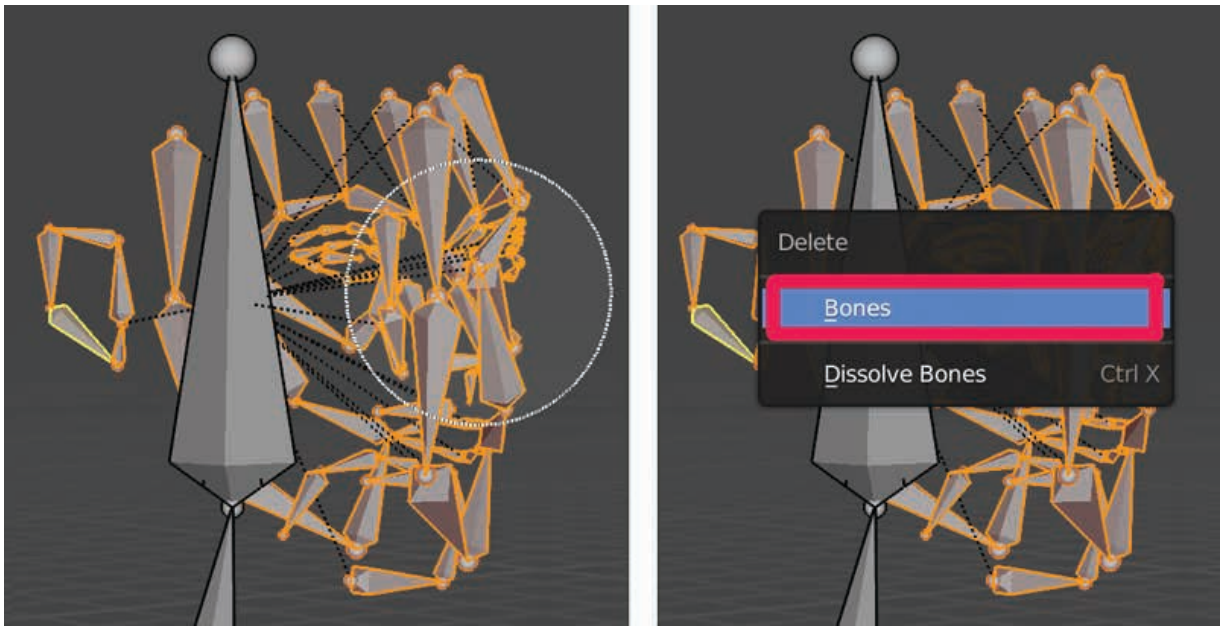




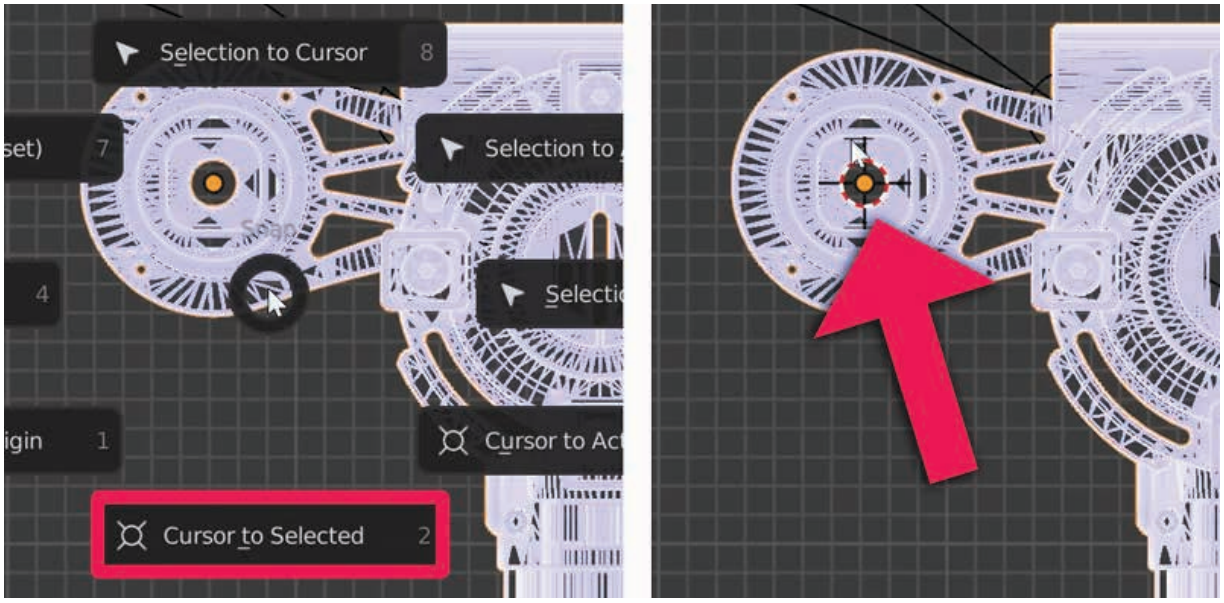
Although you could animate the robot by parenting everything in the right order and just animating rotations, adding a rig with bones though being a little bit more work gives you much more creative control and is definitely worth the extra effort. For the Armature we can use a **Human Meta-Rig**. To get this option, **the Rigify add-on** needs to be enabled in Add-ons. Then add it by **pressing Shift+A > Armature > Human (Meta-Rig)**. Scale it to roughly the size of the robot if necessary.



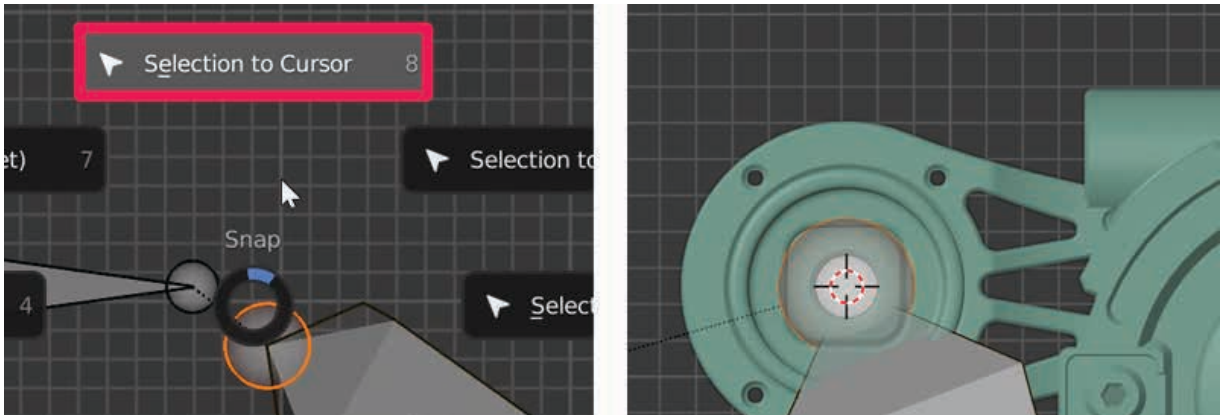
The benefit of the **Human Meta-Rig** is that it has hands and fingers. It also has facial bones which we don't need in this case, so we can delete those. Isolate the Armature by **pressing / on the Numpad** and select the facial bones in **Edit Mode**. Circle select which you can activate by pressing C is useful for this. Right-click to exit Circle select. **Press X** and choose **Bones** to delete the selected bones.



Press / on the Numpad again to exit Isolation mode. Now we need to line the bones up with the model. Fortunately we only need to do this on one side, we can mirror the Armature later. Select a part of the robot. You may want to hide some other objects if that makes it easier to see what you're doing. **Press Shift+S** and choose **Cursor To Selected**. Now the **3D Cursor** is at the same point as the Pivot Point of that object.

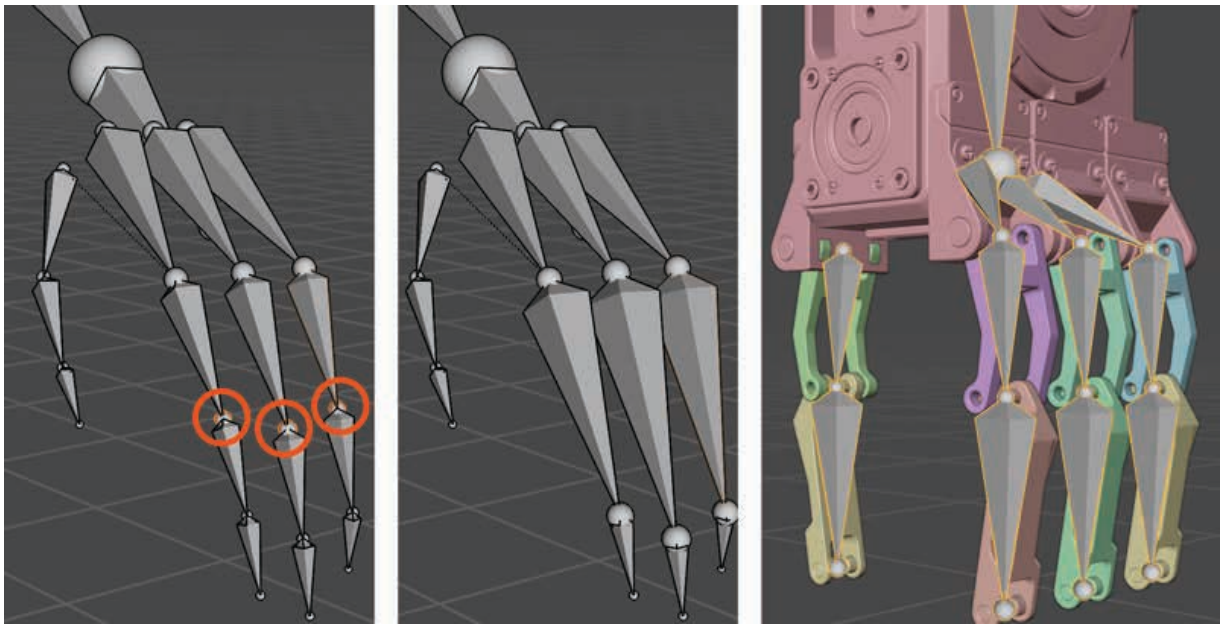


Select the Armature and in Edit Mode select the **root** of the bone (the ball at the top of the bone) that you want to move. **Press Shift+S** and choose **Selection to Cursor**. Now the root of the bone has moved to where the 3D Cursor is, and also where the Pivot Point of that object is.



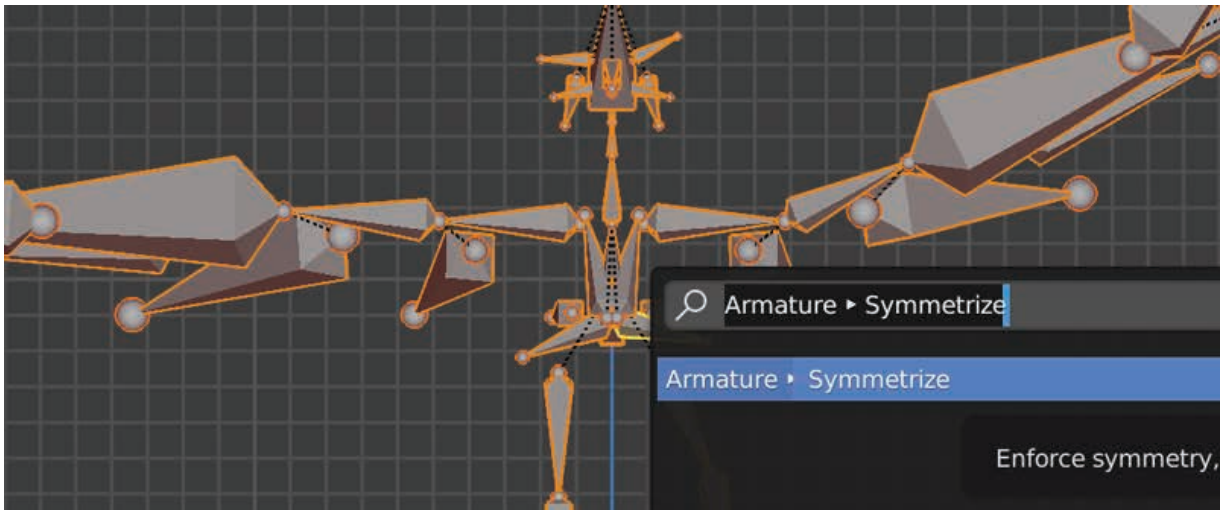
Repeat this for all the bones. You may have to hide parts of the model from time to time, to see what you're doing. Depending on the model, you may have too many bones or segments.

In this case, the fingers only have two segments so I dissolve one of the joints on the armature in each finger by selecting the joint and **pressing Ctrl+X**. I also deleted one of the fingers, since there are only 4 on the model.

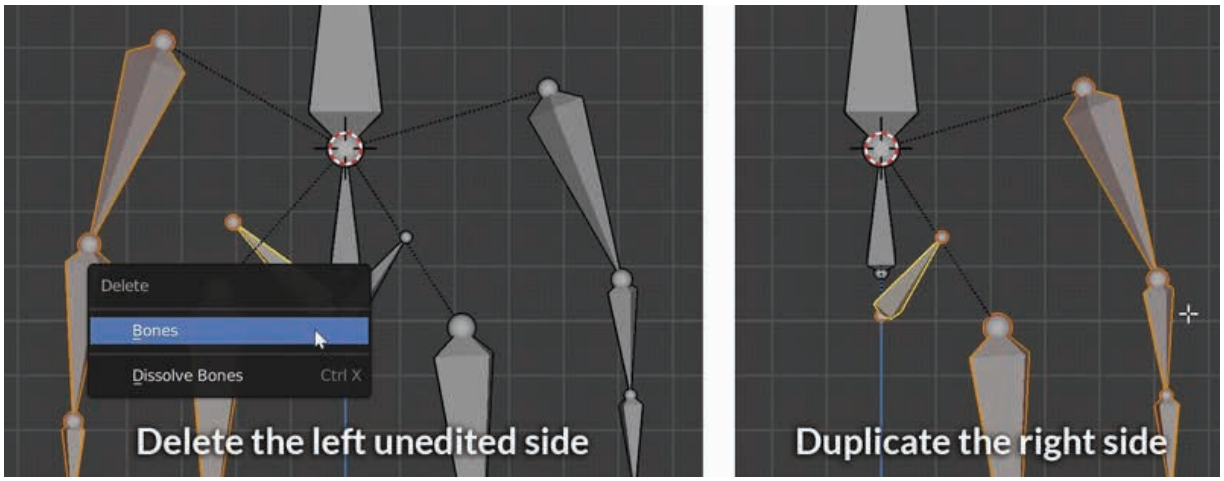


Robot Rigging Part 2 Parenting to Bones

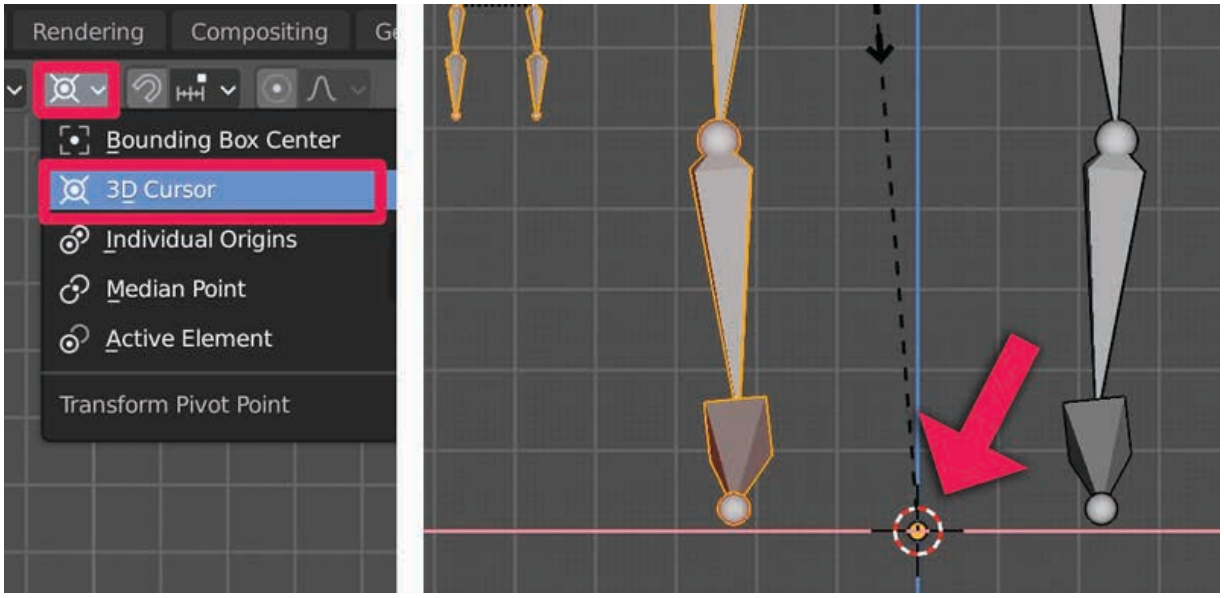
Normally you can **press F3** and search for the **Armature > Symmetrize** function to mirror the changes to the Armature to the other side (shows below with a bird armature example). Unfortunately, since we dissolved and deleted bones, that doesn't work out in this case.



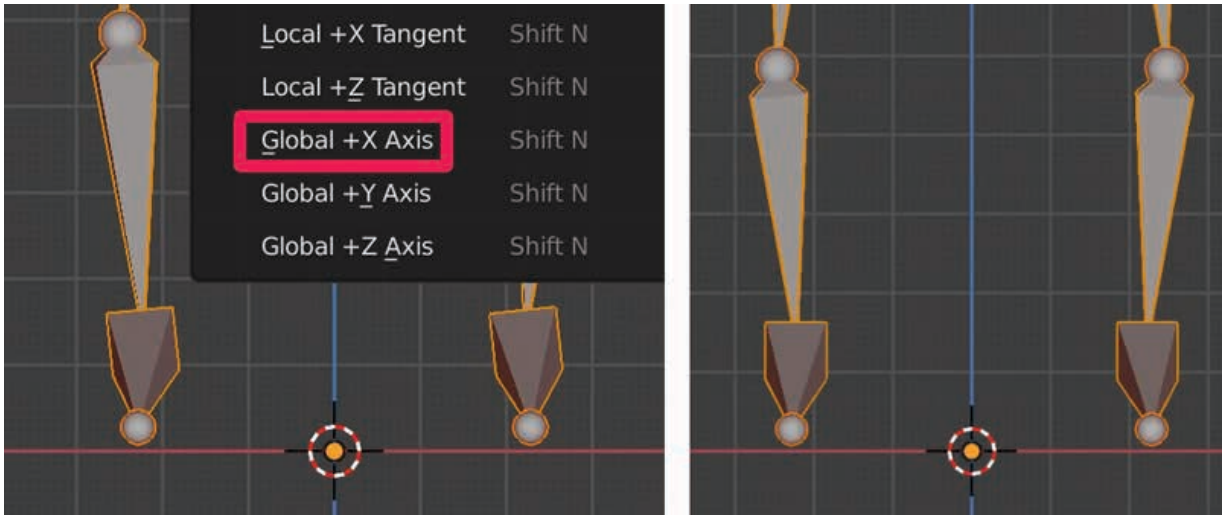
Instead, just select and delete the bones on the other side. Then select the bones you want to mirror, and duplicate them with **Shift+D**.



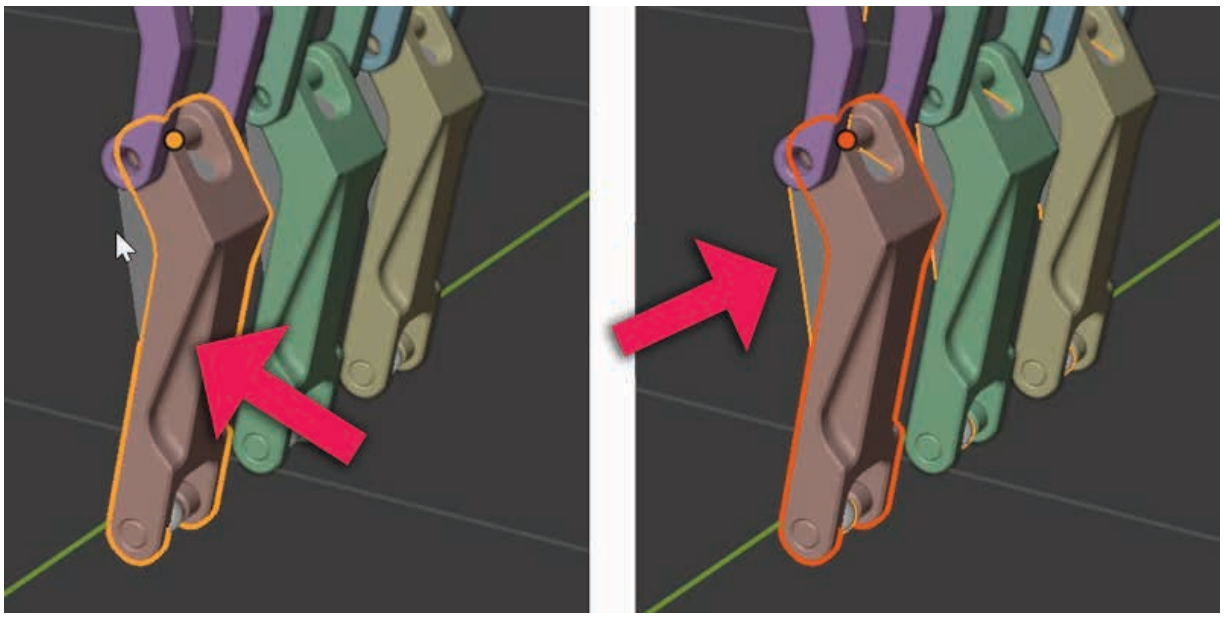
With **Transform Pivot Point** set to **3D Cursor** and the 3D Cursor at the **World Origin**, **press S+X+[-1]** to move the duplicates to the other side.



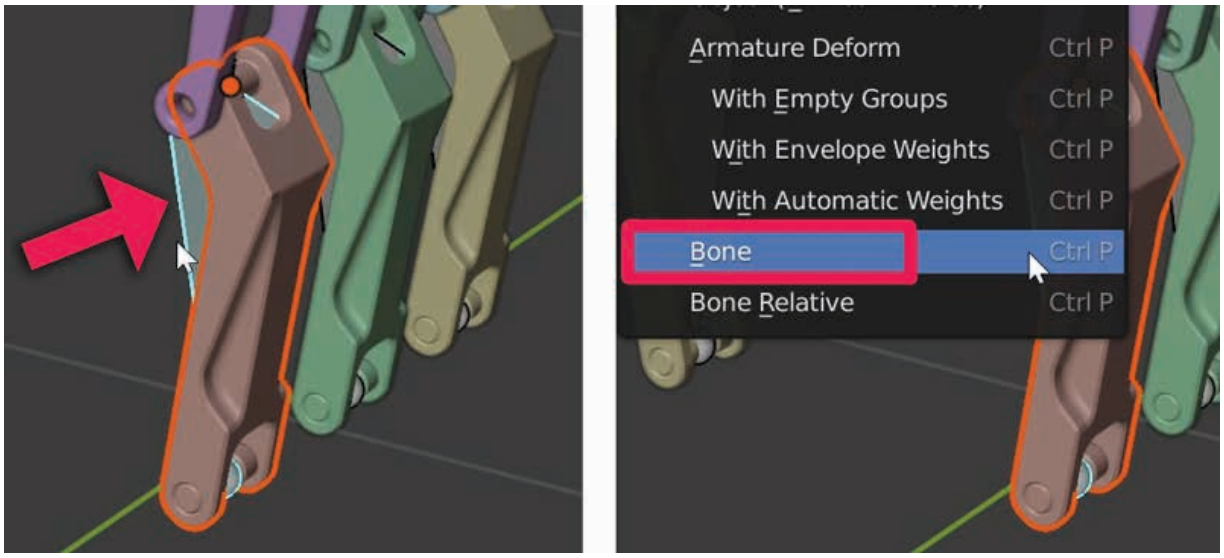
If some bones have a strange rotation, you can select them and **press Shift+N** to get the Bone Roll menu. Then choose **Global +X Axis** to correct the bone roll.



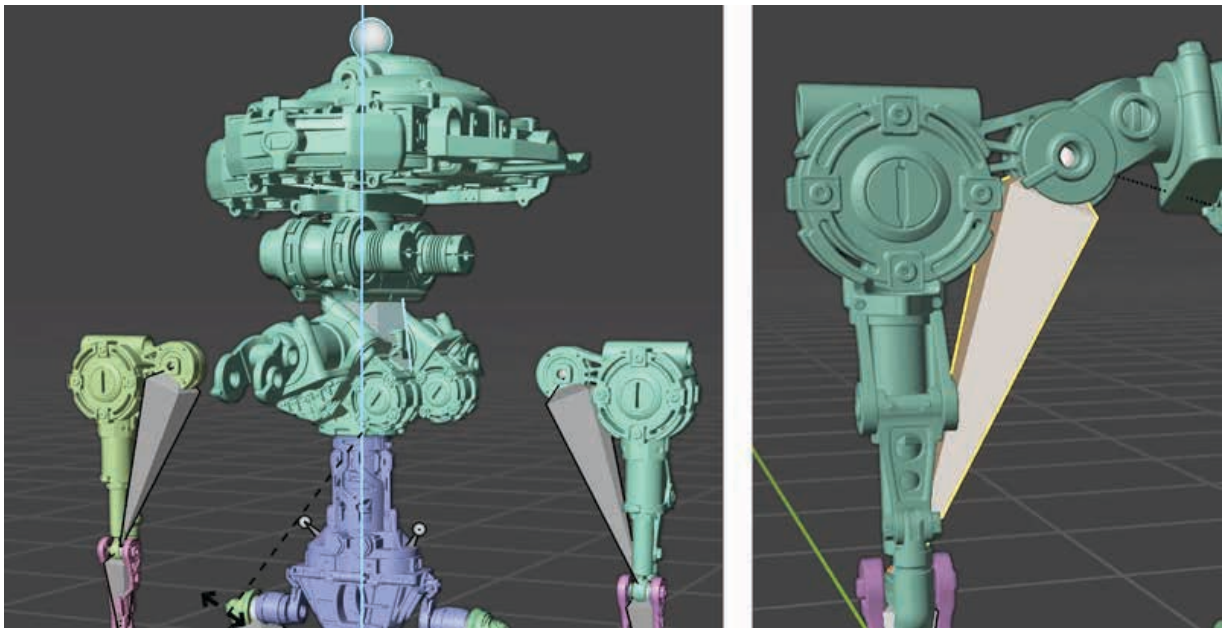
The good news after all that work, is that parenting the model to the Armature is quite easy. There's no weight painting involved at all. We just parent the objects to the bones. **Select the object** you want to parent to a bone, then **hold Shift** and **select the Armature**.



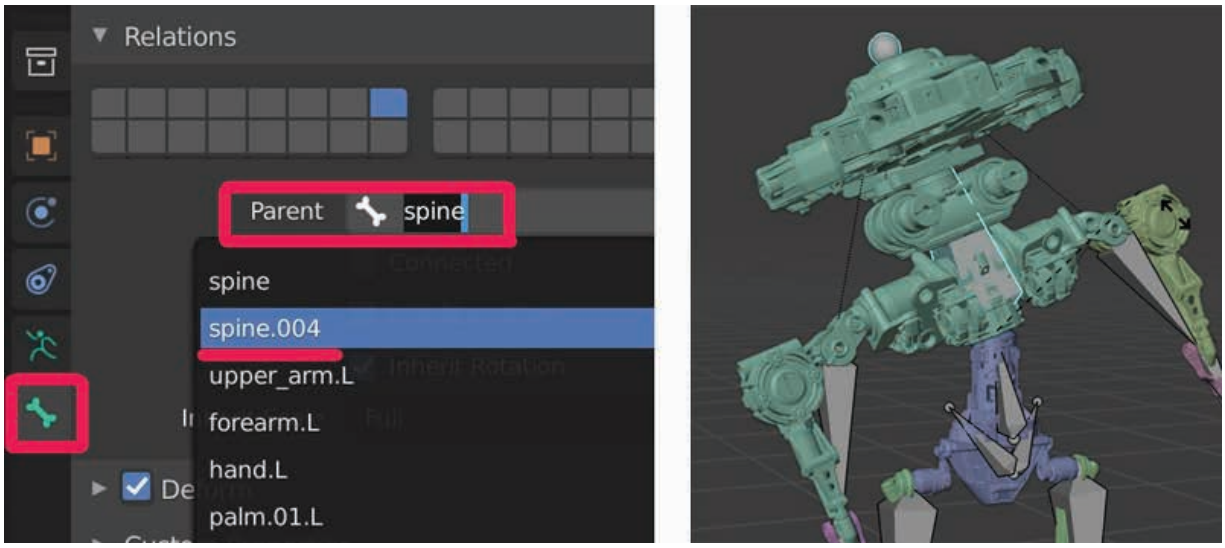
Go to **Pose Mode** and select only the bone you want to parent to by clicking on it. **Press Ctrl+P and choose Bone**. Wiggle the bone a bit to check that the parenting worked. Repeat these steps for all the objects that need to be parented to bones. With object and armature selected, you can **press Ctrl+Tab** to switch between Object Mode and Pose mode quickly.



If you need to change the relationships of some bones, you can do that in Edit Mode under **Relations** in the **Bone tab** in the **Parent** field. In this case for example because I deleted some spine bones, the arms no longer follow when I rotate the one spine bone that is left. To fix this, select the arm bone in Edit Mode.

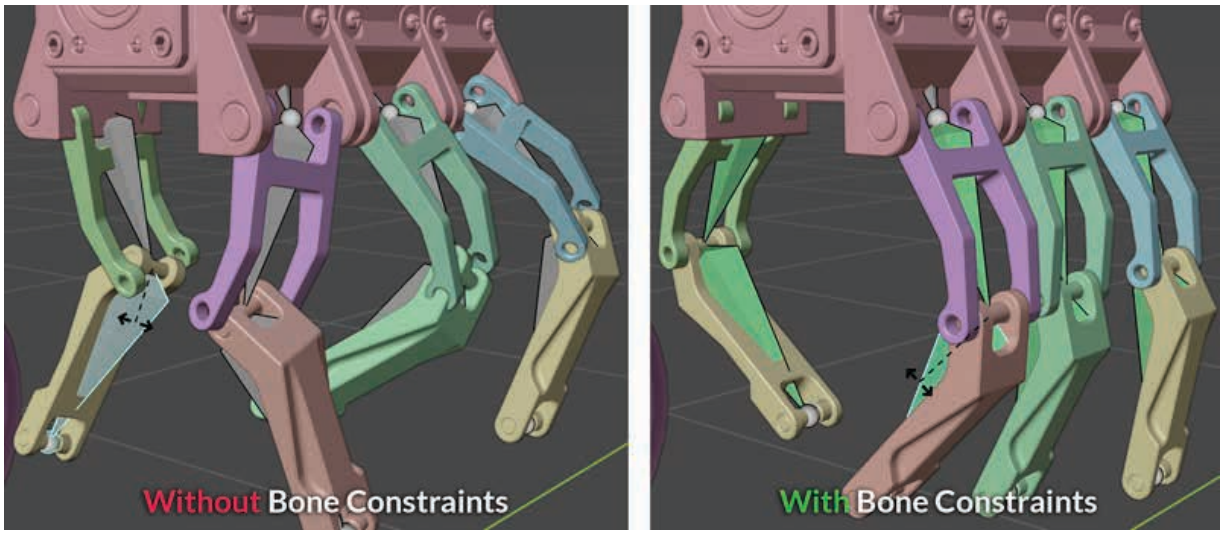


Then, go to the **Bone** tab and under **Relations**, change the bone in the **Parent** field to the one that you want to parent the arm to. In this case, it's spine.004 (it's a good idea to check which one you need first). Repeat for the arm on the other side. Now when you move the spine bone in Pose Mode, the arms follow.

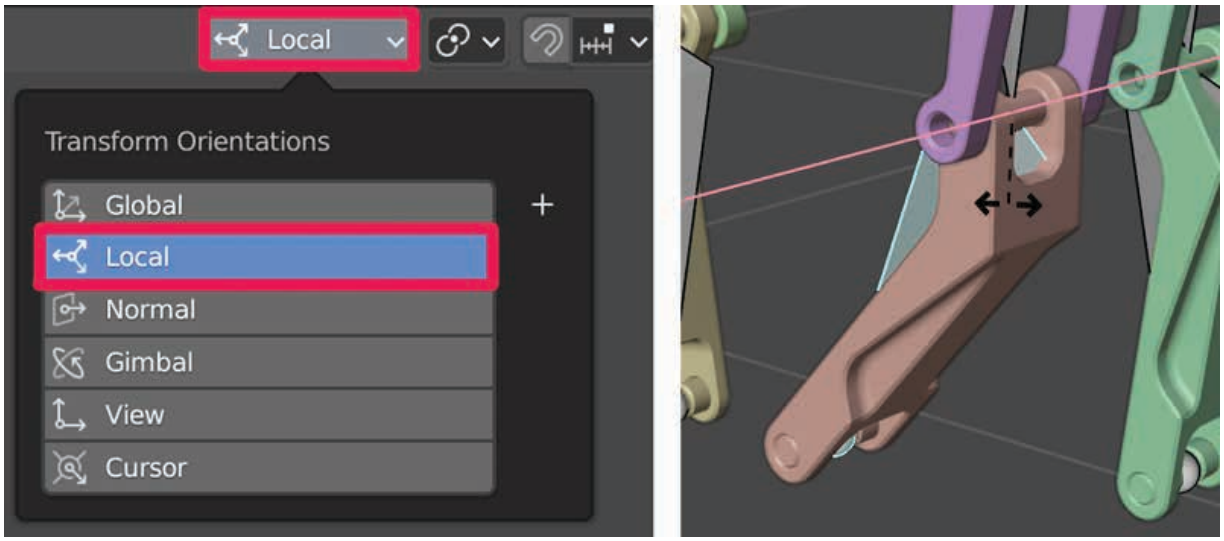


Robot Rigging Part 3 Constraints

Now when you rotate the bones in **Pose Mode**, the objects follow. In this case the robot uses single-axis hinges for the rotation of its parts. In other words, the bone should only be allowed to rotate on one axis. We therefore need to limit the other axes using a **Bone Constraint**.

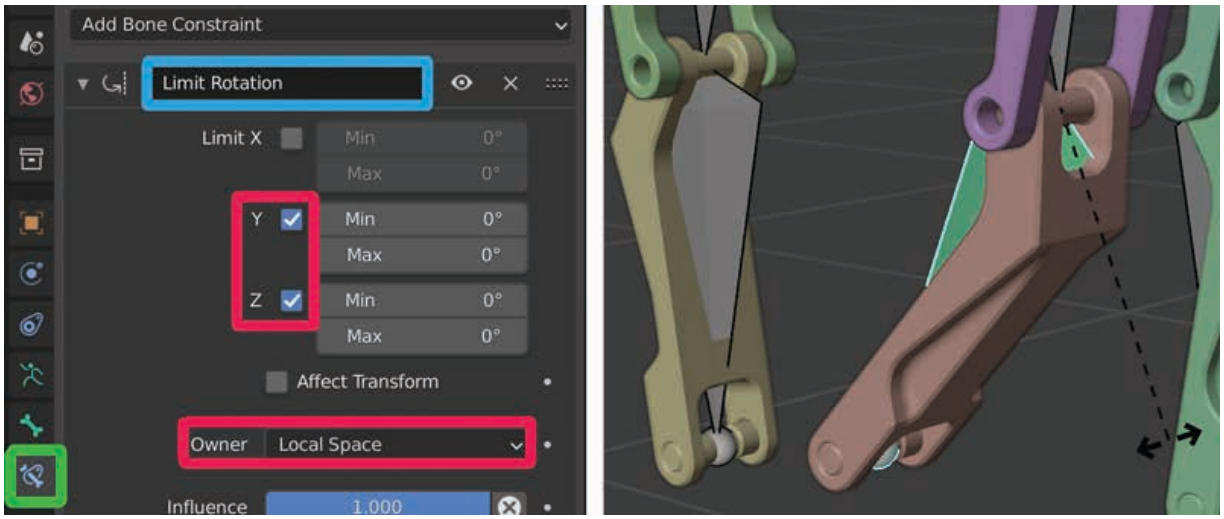


To do this, first check which Axis you really need to constrain. Set **Transform Orientations to Local** in the 3D viewport. Now you can rotate the bone on the X, Y and Z axis in **Pose Mode** to find out which Local Axis you want. In this case it's the **Local X-axis**.

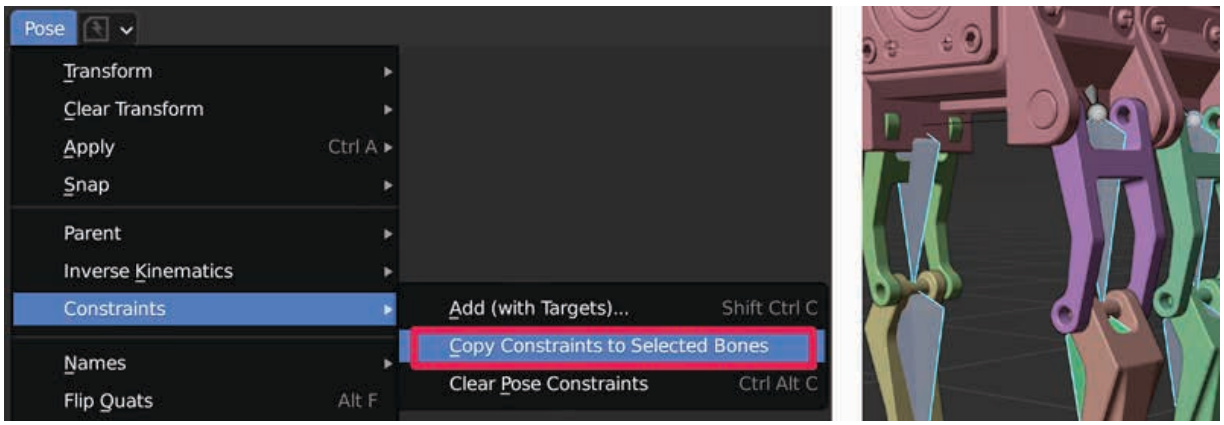


Select the Armature. In Pose Mode, add a **Limit Rotation** Bone Constraint to the bone. *Note: Be careful to use a Bone Constraint and not an Object Constraint. The icons look confusingly similar!*

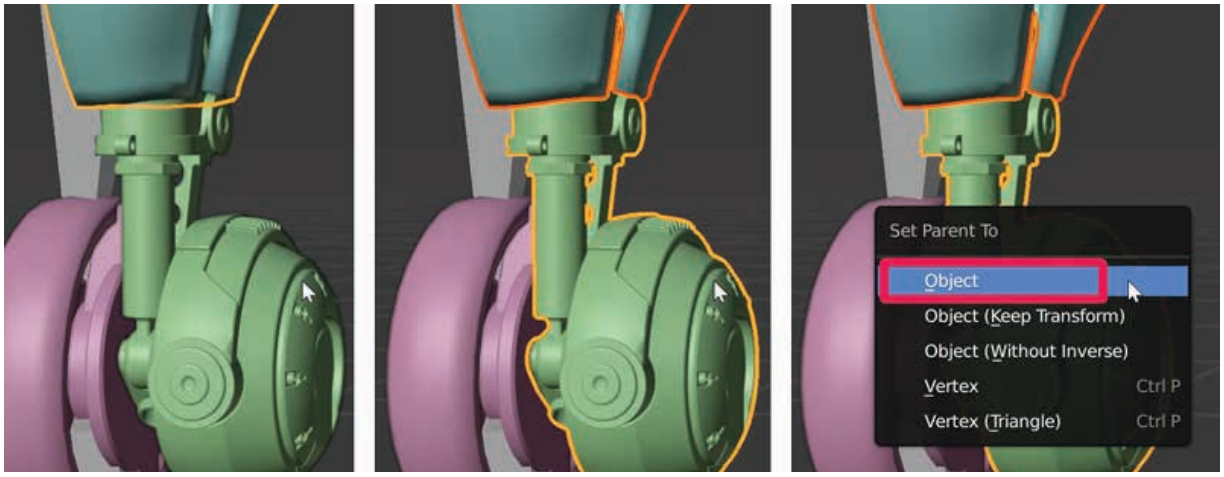
Set **Owner** to **Local Space**. In this case we want the X axis, so check the boxes for Y and Z, so that those axes are completely restrained from rotation. Now if you move the Pose bone, it can only rotate the object along the Local X-axis.



Luckily you don't need to do this for every single bone. While **holding Shift** select the remaining bones that need to only rotate on the Local X-axis. Then while **still holding Shift** select the bone that already has the Constraint. Then go to Pose > Constraints > **Copy Constraints to Selected Bones**.



Now that we've rigged the inner parts of the robot to the Armature, we can unhide the outer shell collection. We don't need to parent these parts to any bones, as it's simpler and faster to just parent them to the inner objects. To do this, select an outer part, **hold Shift** and select an inner part, then **press Ctrl+P** and choose **Object**.

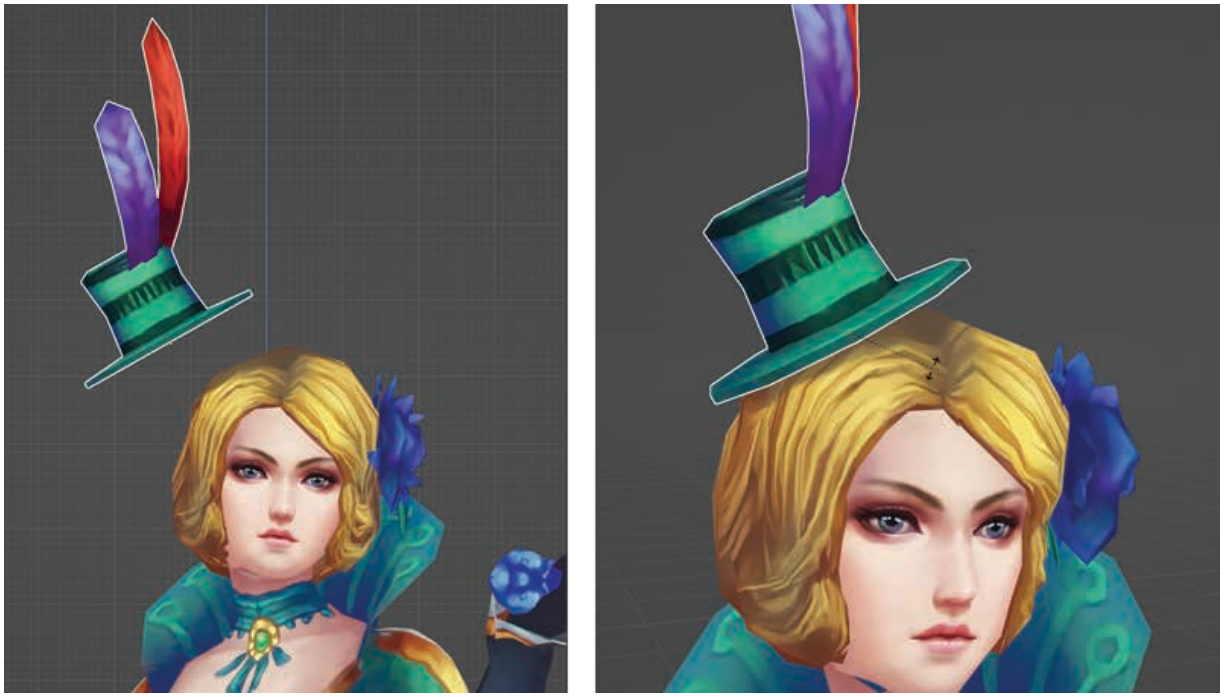


Now you can animate the robot by posing the rig and setting keyframes for the rotation.

Adding Objects to a rigged Character

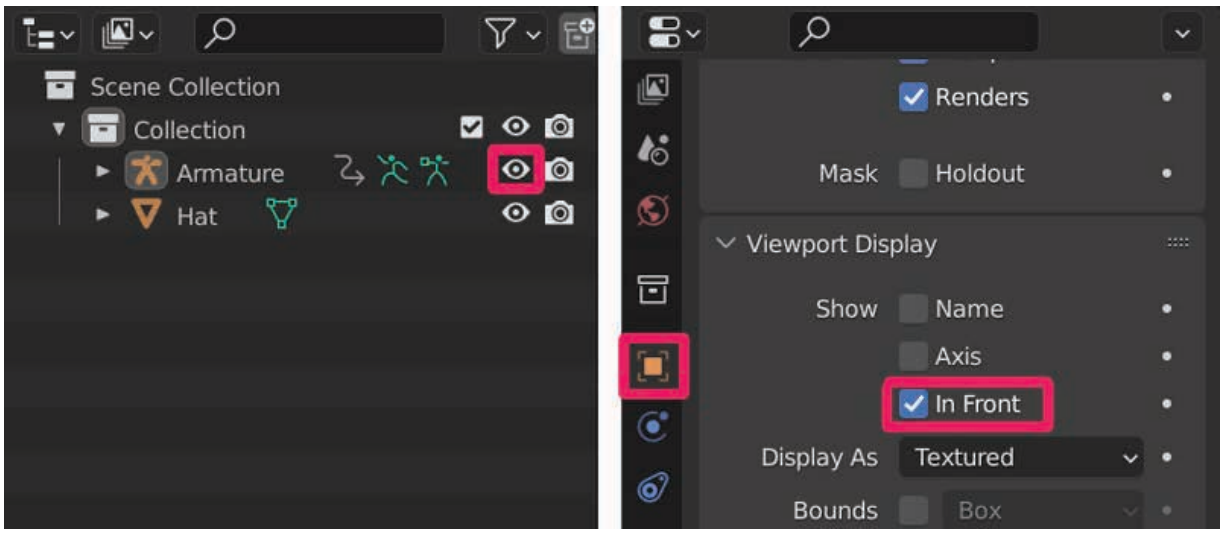
What if you want to add an object - a hat, for example - to a character that's already rigged and perhaps even animated? Luckily, it's quite easy as shows in the next steps.

First, Move and rotate the object to where it should be. **Press G** to move and **press R** to rotate. You can **press R twice** to rotate freely around all axes.

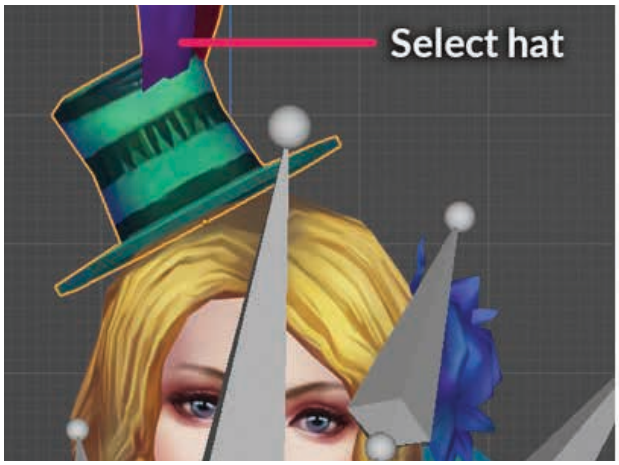


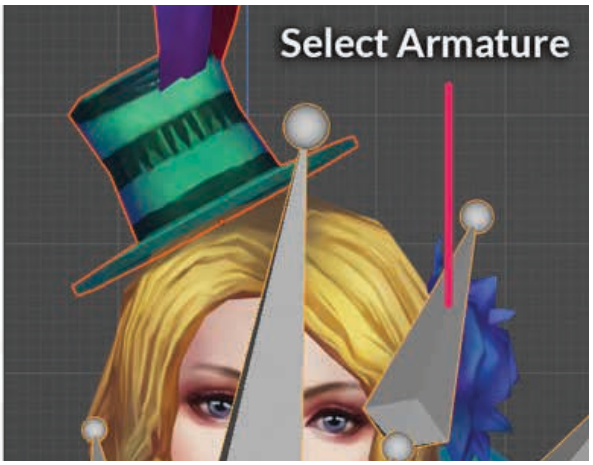
"Nobel Woman Coquette Madam Lady" by Jim Cheng on Sketchfab.
https://sketchfab.com/chengkehan

In case the **Armature** is hidden, unhide it in the Outliner. Set the Armature to **In Front** so it's easier to see the bones.

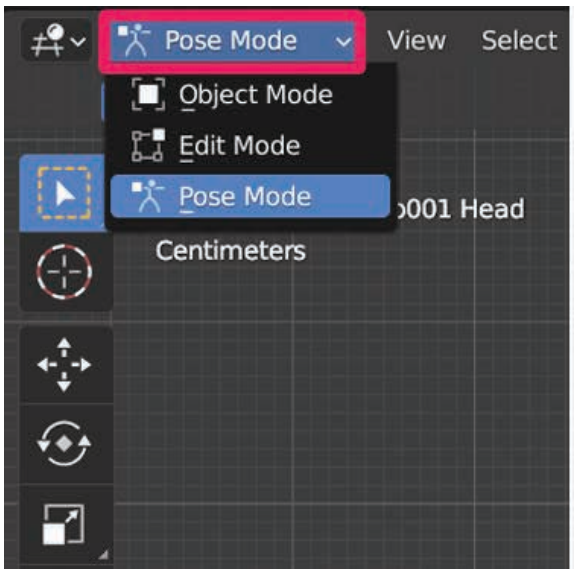


In **Object Mode** Select the object and **while holding Shift**, select the Armature in the viewport.

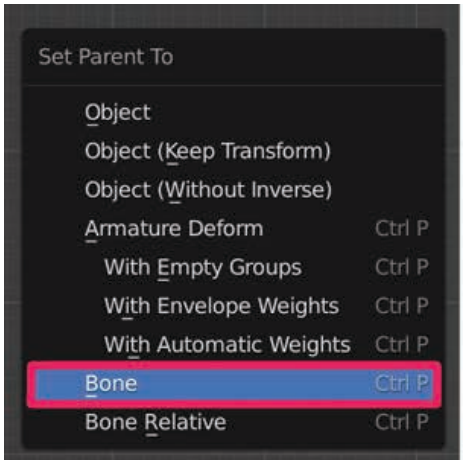


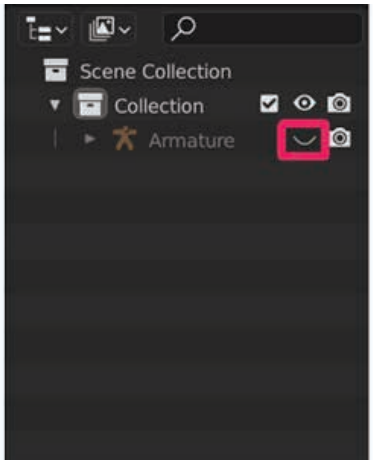


Switch to **Pose Mode** by **pressing Ctrl+TAB** or by selecting Pose Mode from the drop-down menu and select the bone you want to parent the object to. For the hat in this example, the head bone is the logical choice. It will have a blue outline once selected.



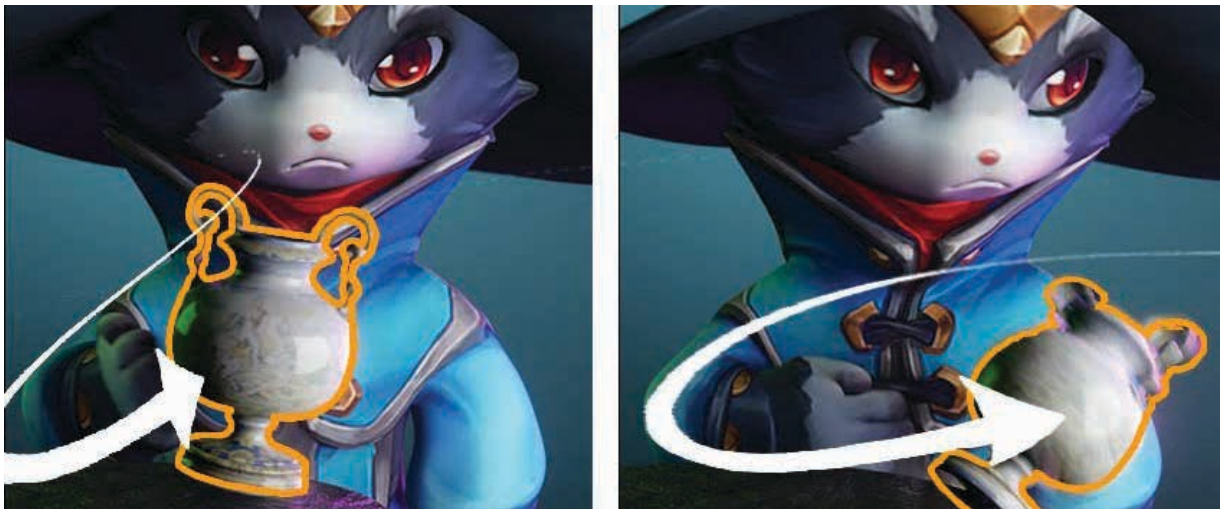
Press Ctr+P and choose **Bone**. **Press Ctrl+Tab** again to exit Pose Mode. Now the object is parented to that bone. You can hide the Armature again in the Outliner.





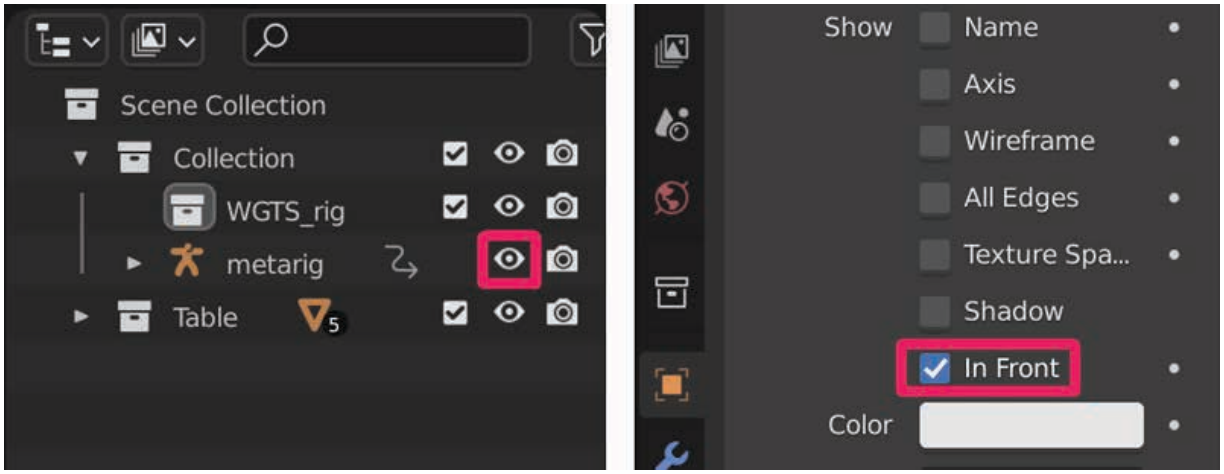
How to have a character interact with Physics Simulations

Here, we look at an easy method for having characters interact with objects using physics simulations.

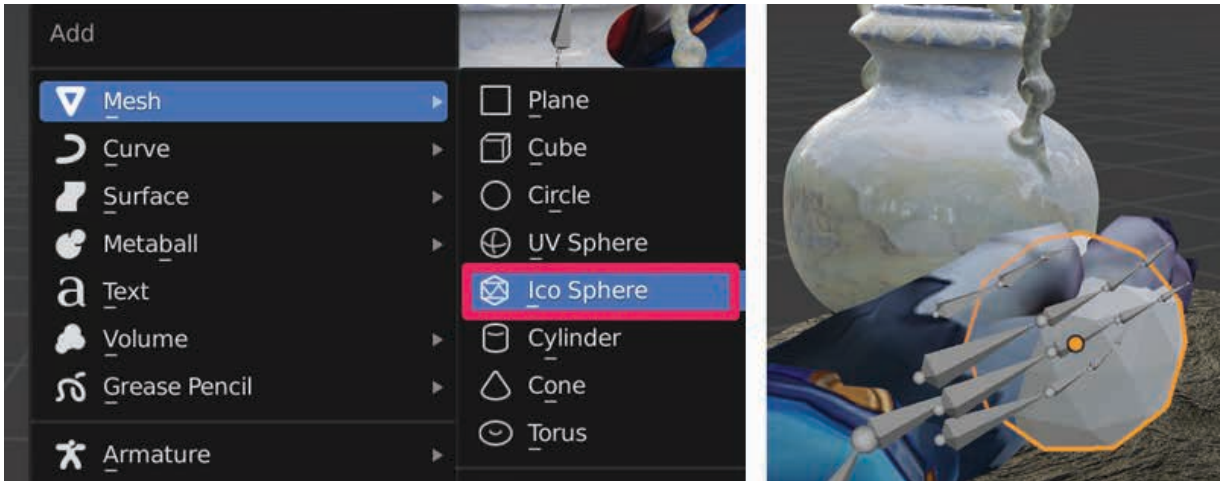


Wizard Cat by Moonstone on Sketchfab.

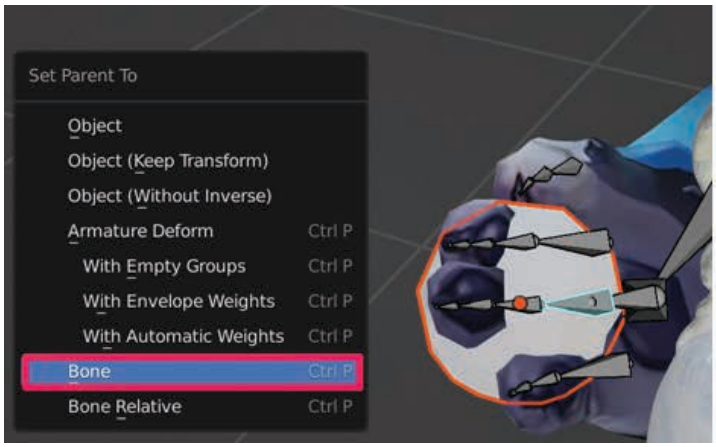
Enable the visibility of the character armature and set it to In Front so you can easily see the bones.

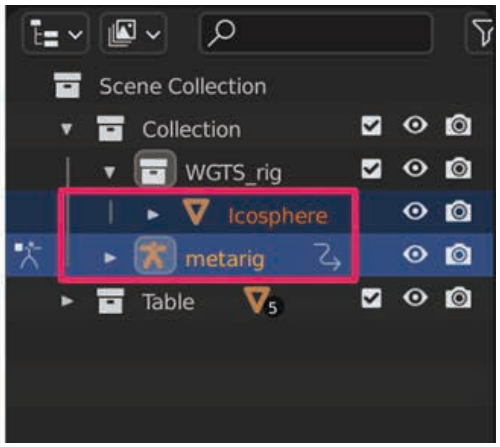


Add a Primitive like an Icosphere by **pressing Shift+A > Mesh > Icosphere** and move / scale it to overlap whichever part of the character interacts with a physics simulation. In this case, the hand of the cat strikes the vase so we'll add the Icosphere to the hand.



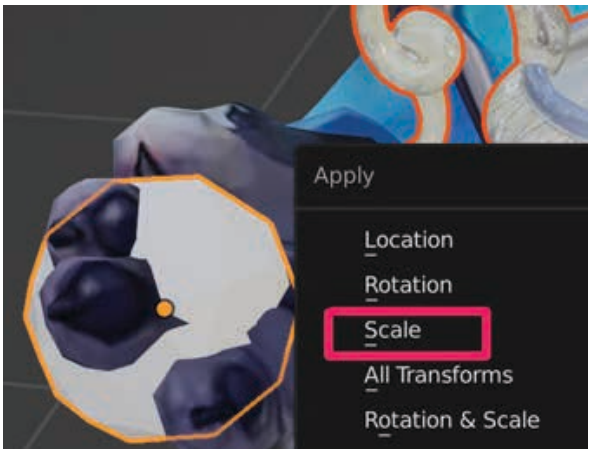
Parent it to the hand bone like explained in the topic "**Adding Objects to a Character**". Repeat for any other body parts that will interact with the simulation. If it's difficult to select the armature, select it in the Outliner instead of the 3D viewport.



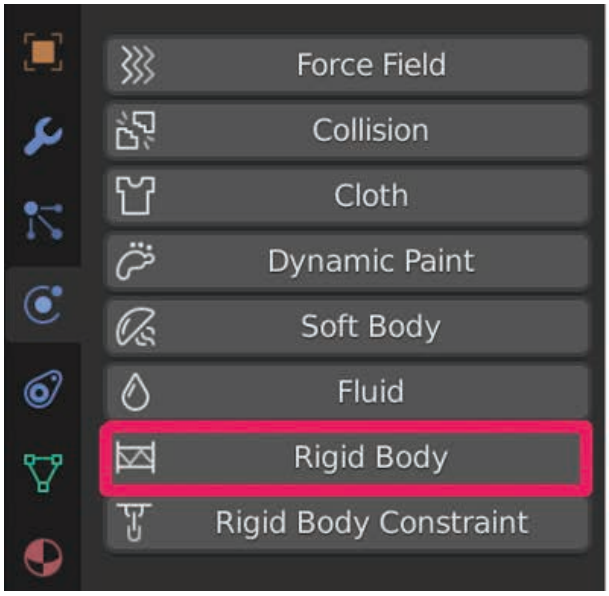


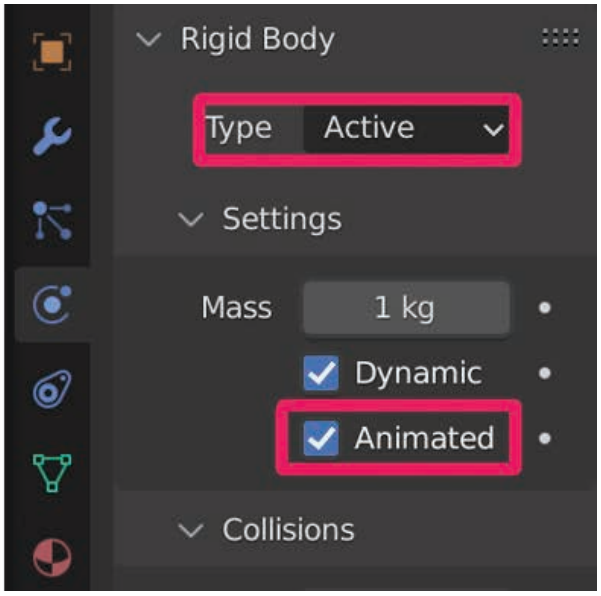
Select the Icosphere(s), **press Ctrl+A** and apply **Scale**.

Do the same for any object that you want to use in the physics simulation, like the vase in this case.

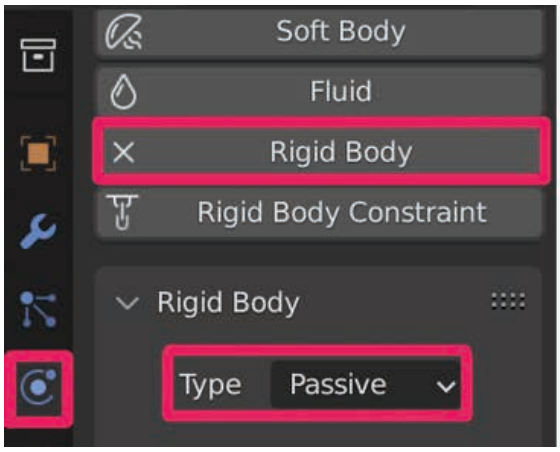


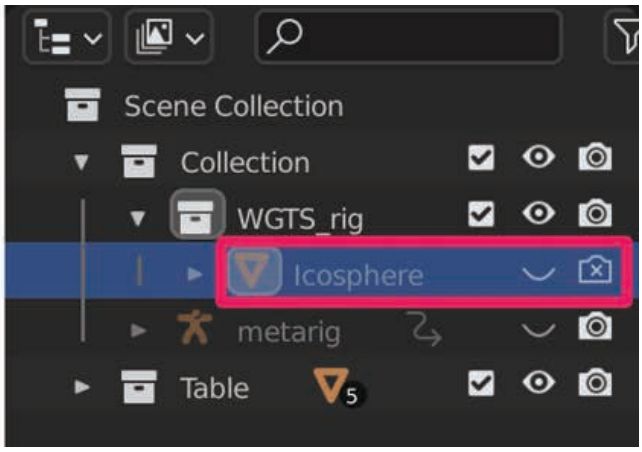
Select any objects you want the character to interact with - the vase in other words - and turn them into **Active** Physics objects by clicking on "Rigid Body". Do the same for the Icosphere(s), but also check "**Animated**" in that case.





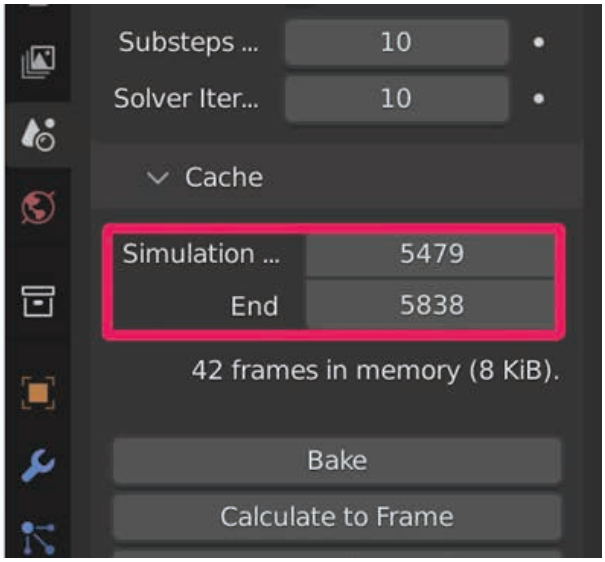
If you have a ground plane, make it a **Passive** Physics object. In this case, that's the little table the vase is on (as well as the floor, but we don't see it so we don't need to worry about it). Uncheck **render visibility** of the Icospheres in the Outliner.





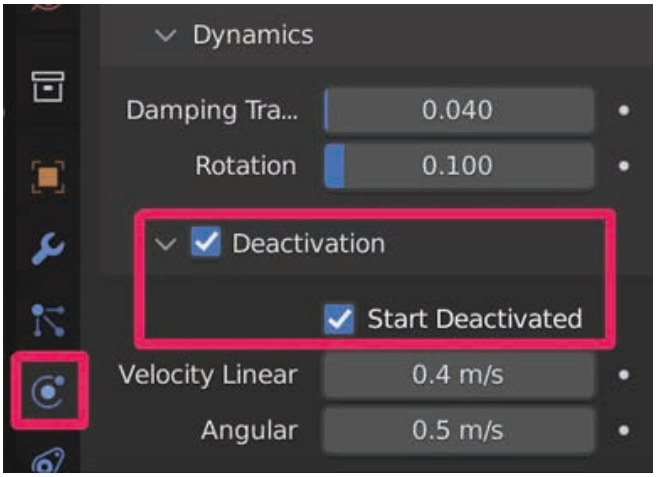
Now when the animation is played, the character will interact with the simulation.

Potential problems:
Note that if like me, you're using motion capture animation, the frame rate between which the action takes place may very well not be the default 1-250 frames. In that case, change the frame range in the **World Rigid Body Cache** settings to match what you have on the timeline.



Also, the vase in this case may enthusiastically launch itself into the air before the cat even slaps it off the table. That's because of the collision detection between it and the table.

This is easy to solve, just check "**Deactivation**" and "**Start Deactivated**". Now it won't move until hit by another Rigid Body.

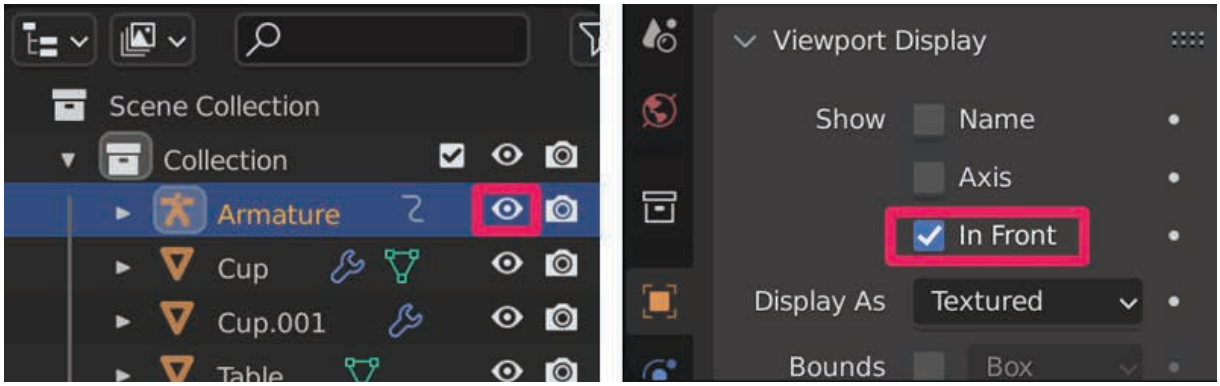


How to have a character interact with an object

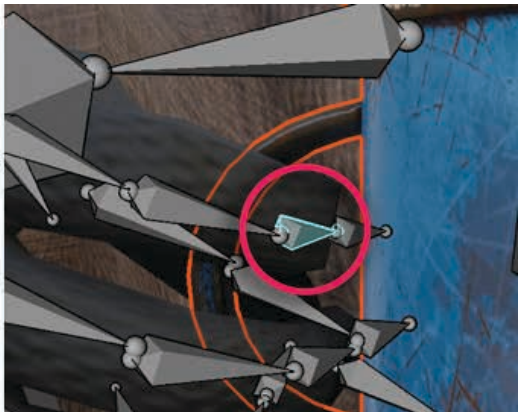
Let's have a look at how to have a character pick up an object, interact with it and then set it back down. In this example we'll have a robot start his day with a cup of oil.

Go to the point on the Timeline from where the object should be attached to the character. Select the object and **press Shift+D t**o duplicate it. **Right-click** to confirm and to make sure that it doesn't move from its position.

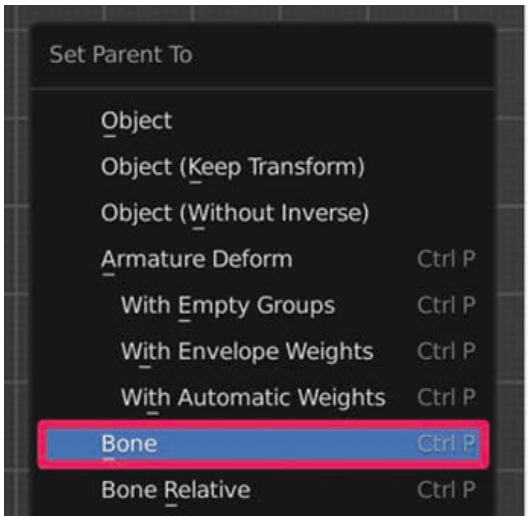
In the Outliner, make sure the **Armature** is not hidden. You can turn on **In Front** to make the bones visible at all times, even inside the character mesh.



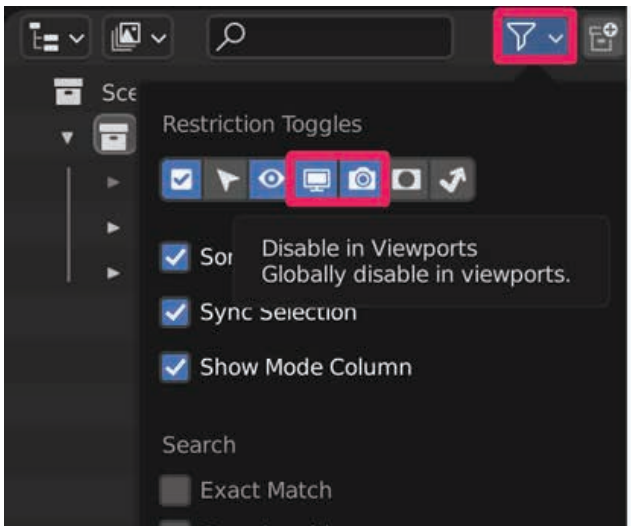
Hold Shift and select both the Cup and the Armature, **press Ctrl+TAB** to enter **Pose mode**. Select the bone you want to parent the object to. In this case, I chose a finger bone.

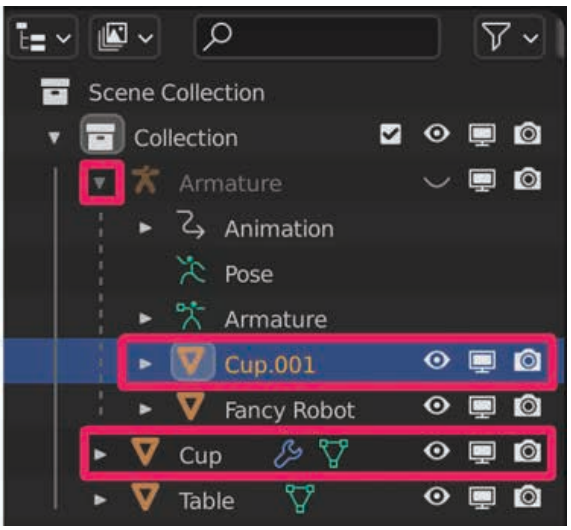


Press Ctrl+P and choose **Bone**.To go back to Object Mode **press Ctrl+Tab** again. Now the duplicate object is parented to the hand while the original object remains on the table. You can turn off the Armature in the Outliner.

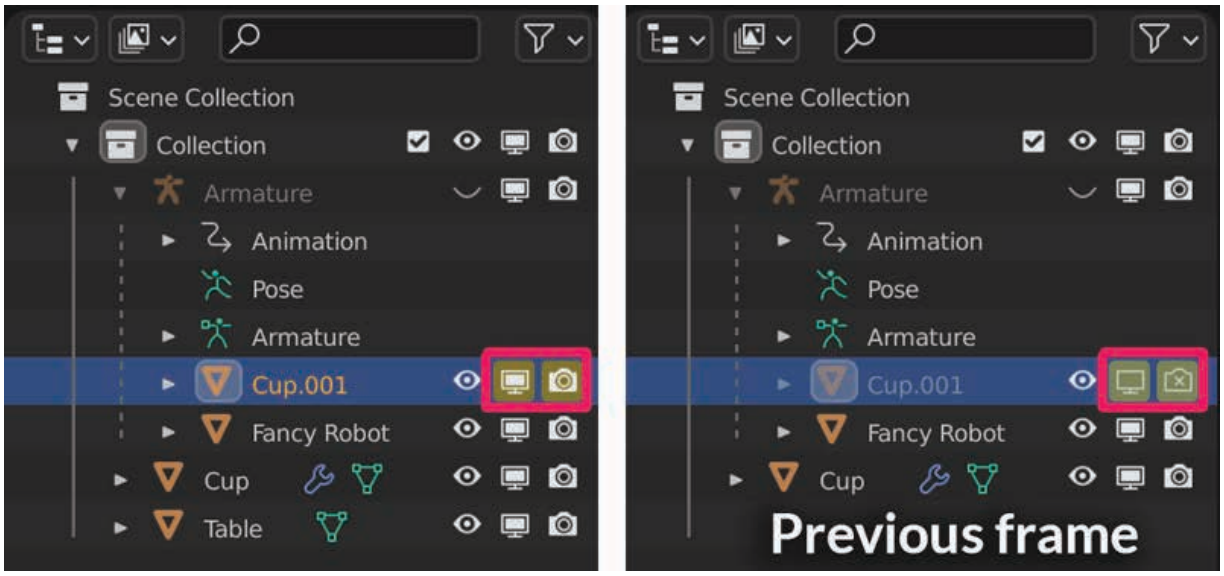


Enable the restriction toggles for **Render visibility** and **Viewport visibility**. Expand the Armature hierarchy to find the parented object. You should be able to see the visibility icons for both the original cup and the parented duplicate (called “Cup.001”).

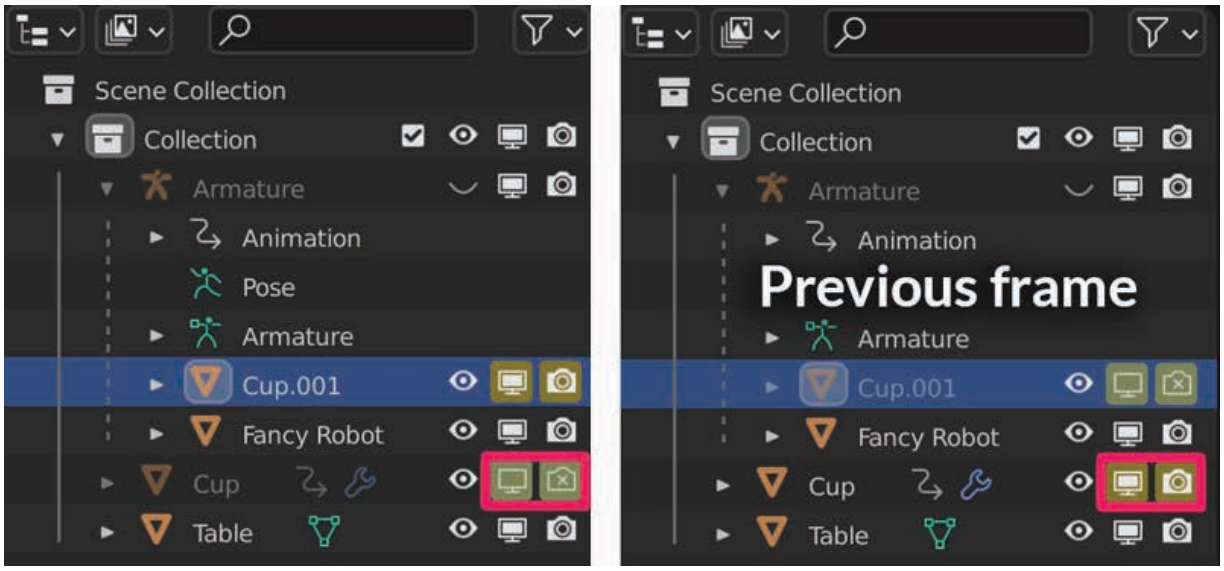




Find the frame where you think the cup should be lifted up from the table and set keyframes for its render and viewport visibility by **hovering over the icon and pressing i.** You want that duplicated, parented cup to become visible in the render at that point. Then **go back a frame**, disable its render and viewport visibility and set another keyframe.



Set the **opposite keyframes** for the original object. In other words, it should be invisible when the other object is visible, and vice versa.



Now the objects visibility is switched from one frame to the next at the right moment, giving the illusion of it being just one object. Do the opposite steps for when the cup is placed back on the table.

Robot with suit by Moon Dong Hwa on Sketchfab
https://sketchfab.com/moondonghwa

Old metal cup by al0sral0 on Sketchfab
https://sketchfab.com/al0sral0

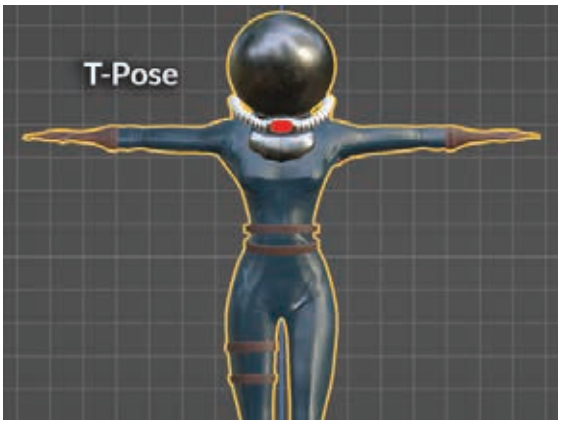
Click here to return to the index.

How to change any pose to T-pose for retargeting

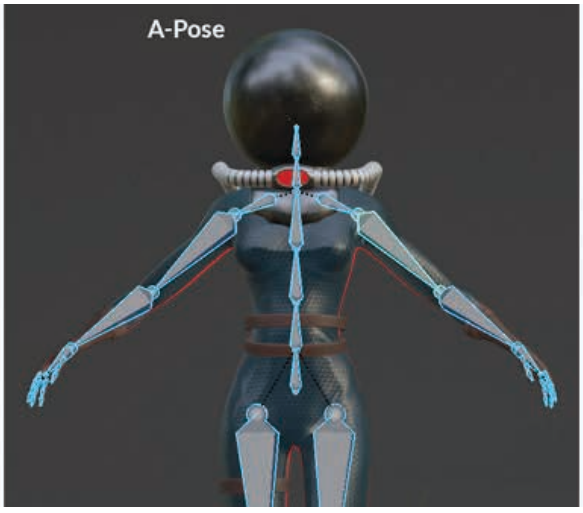
When using a character downloaded from the web, it might be in an **A-pose** or even some other pose.

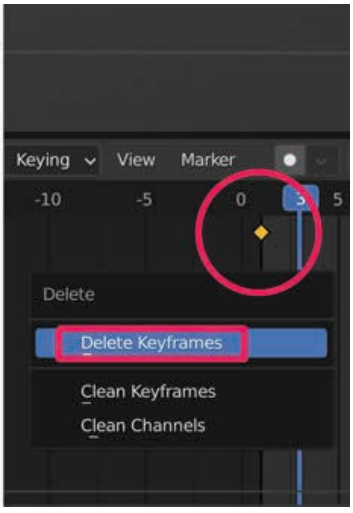
We need to get it into a **T-pose** to be able to retarget it for mocap animation.

Otherwise, the arms will clip through the body as their rotation starting point will be wrong.

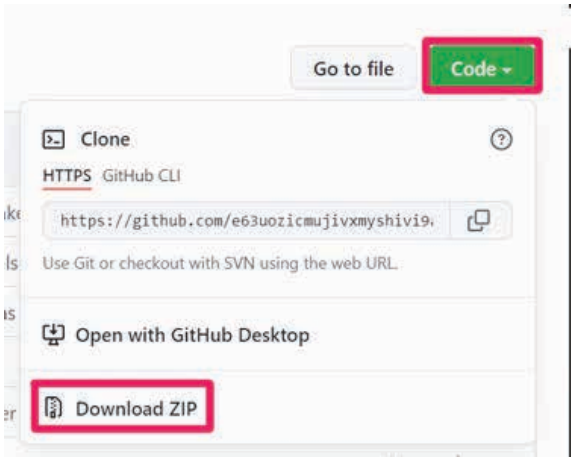


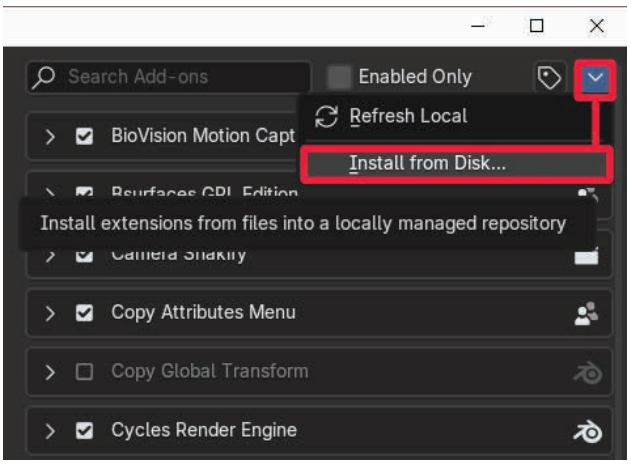
First Select your character's armature and in **Pose mode** select all by **pressing A and then Alt+R** to reset all the bones to their rest pose. Now you can see it's really an A-pose. Delete any keyframes.



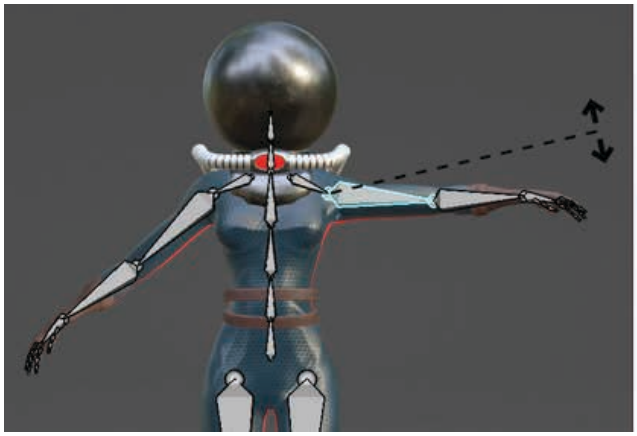


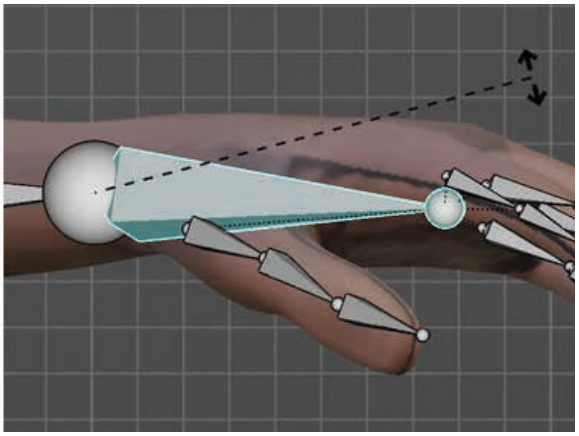
Download the free Cats Blender add-on. In the Add-ons tab in Preferences, click on the drop-down and choose Install from Disk, then find the downloaded ZIP file.
https://github.com/e63uozicmujivxmyshivi9ao/cats-blender-plugin



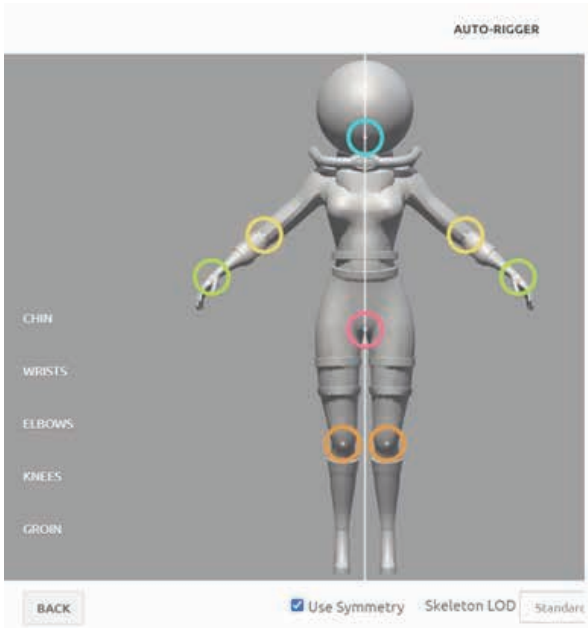


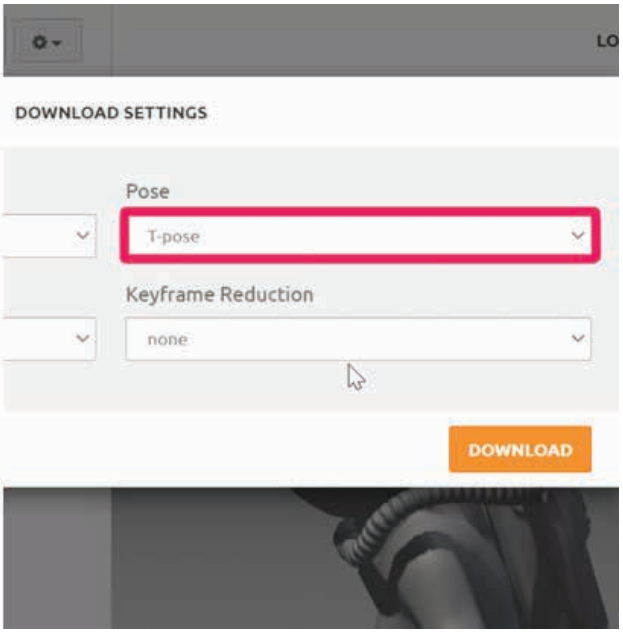
Manually pose the armature into a **T-pose**. Remember to make all the arm bones horizontal and flat, and also the hand and finger bones. Switch to top view sometimes as well to see if everything is straight. You can use a different Armature which is in T-pose as a reference.





Or to skip the manual work, rig it online with Mixamo. This is an easy process, explained earlier in this video Auto-rigging with Mixamo. Remember to delete the existing Armature before exporting it as an FBX for upload to Mixamo though.

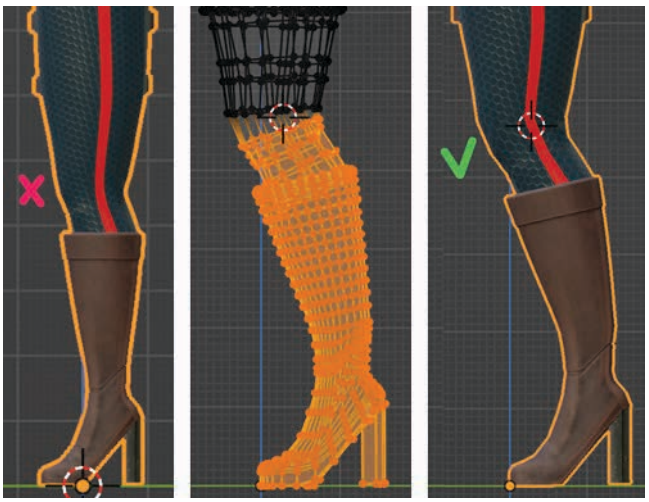




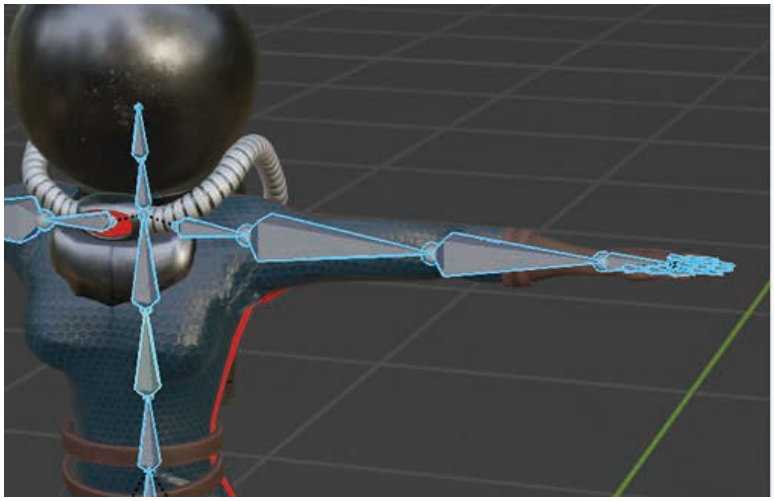
This forces it into a **T-pose**.

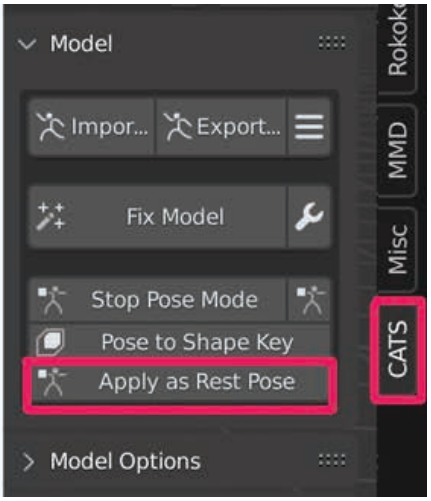
Be sure to delete keyframes after importing the downloaded FBX back into Blender.

In case of trouble with Mixamo: make sure the knees and elbows have a slight bend in them. Otherwise it will not work.



Honestly, Mixamo either works, or will give you an error - in the latter case, I would advice you not to waste any time trying to figure out which of the many problems it can be that causes Mixamo to work and just manually pose the T-pose. It'll only take a couple of minutes anyway. Either way, once done **select all** in Pose Mode. In the **Cats add-on**, click on **Apply as Rest Pose**.

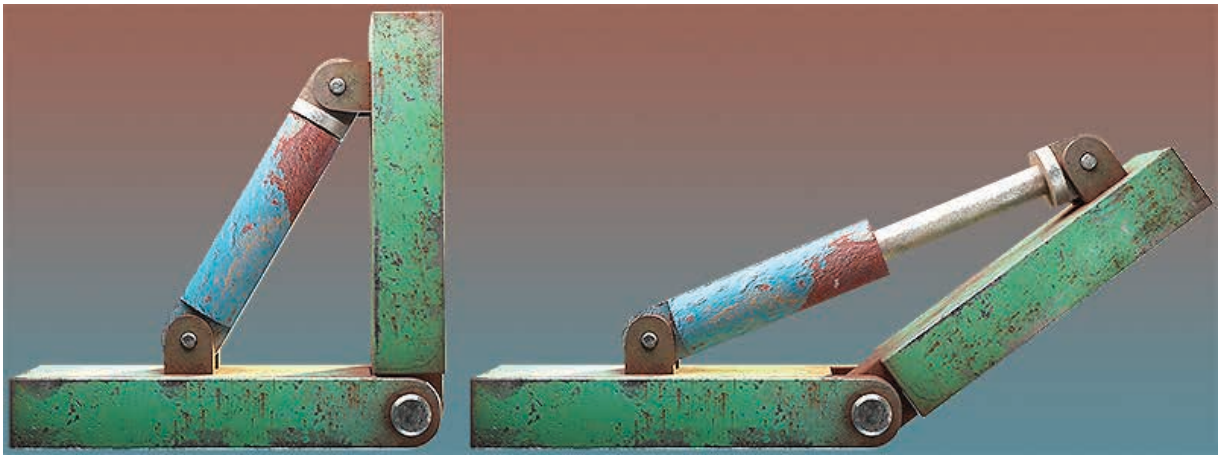




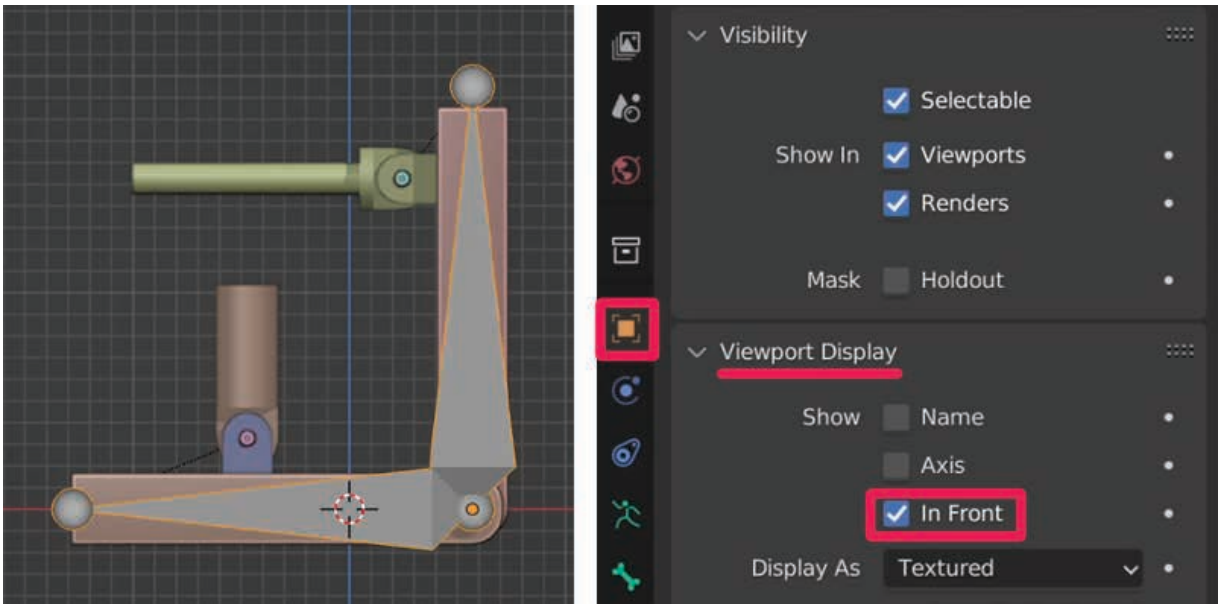
Now use the Rokoko add-on to retarget the animation as previously shown.

Space Girl model by Charles Cloutier on Sketchfab.

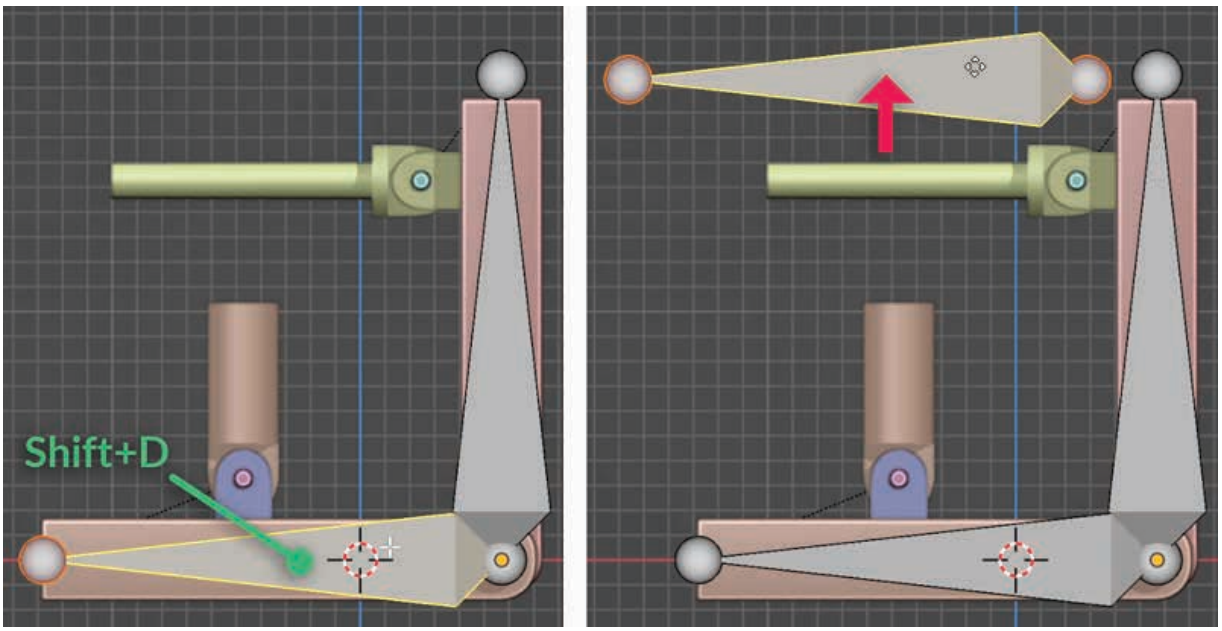
Easy Piston Rigging



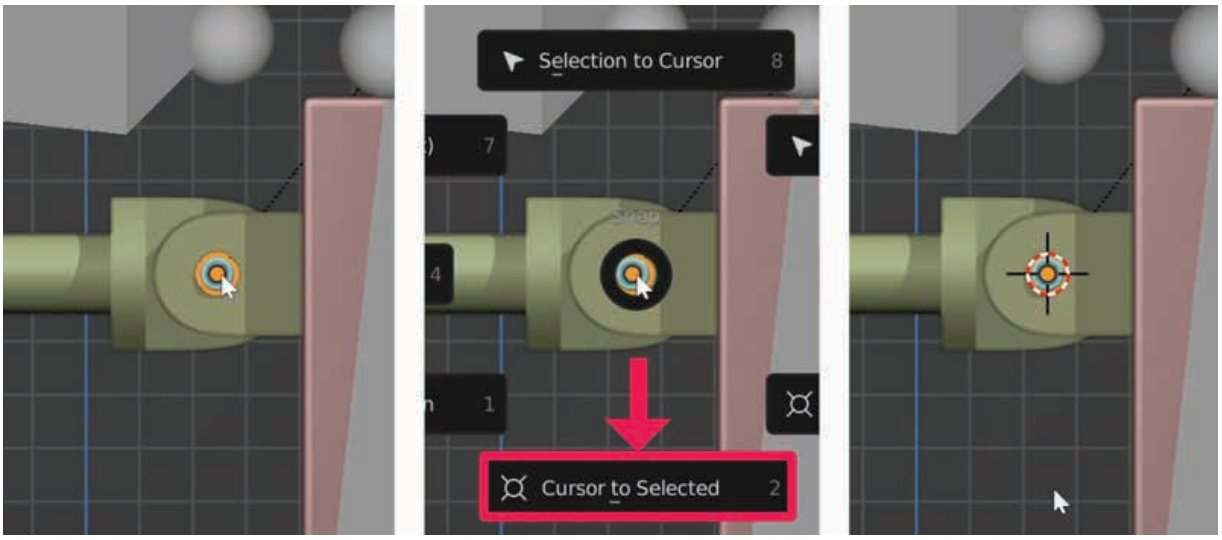
Select the Armature in Object Mode and set it to **In Front**.



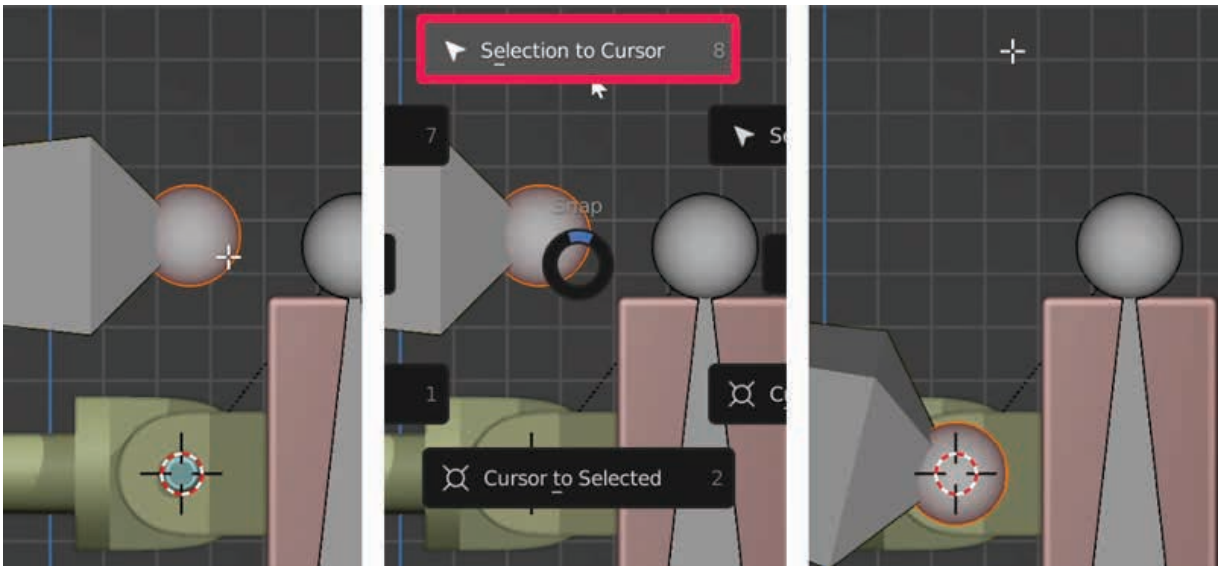
Duplicate a bone with **Shift+D** in Edit Mode.



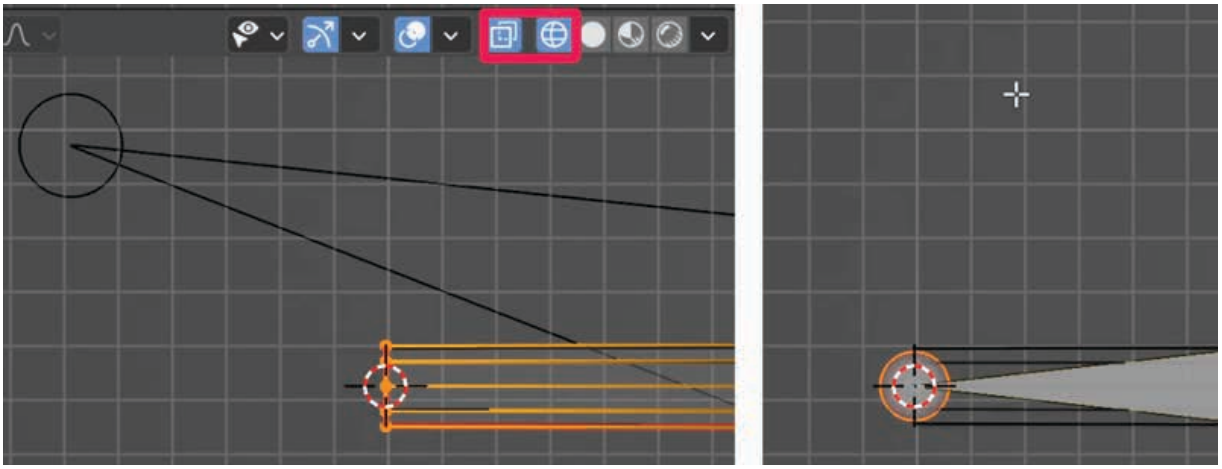
Select a part the Root of the Bone should be aligned to. **Press Shift+S** and choose **Cursor to Selected** to move the 3D cursor there.



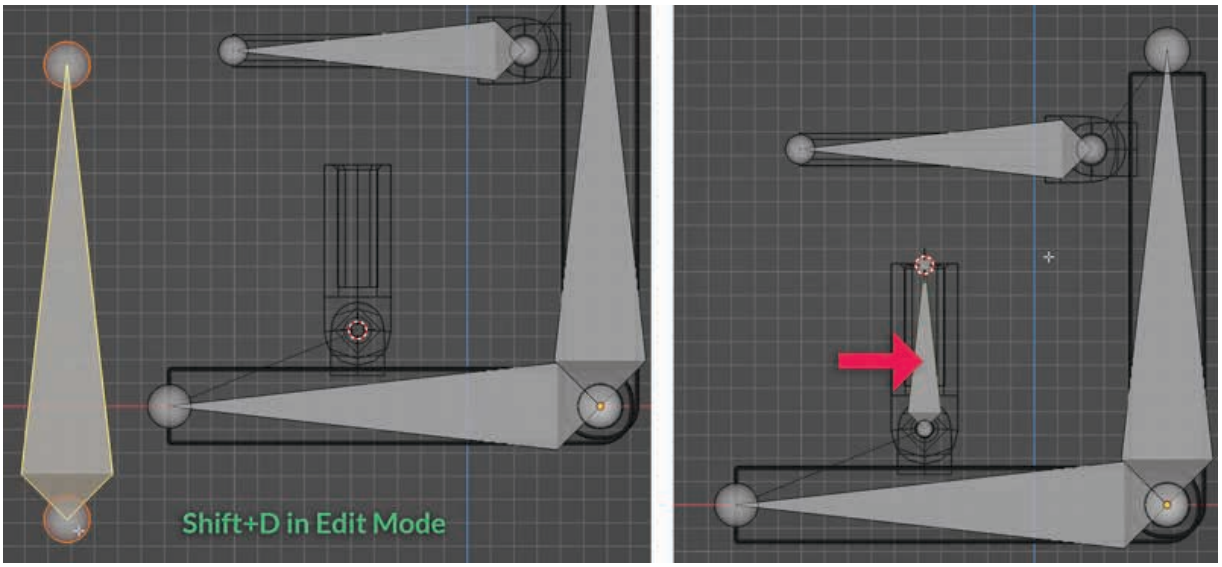
Then select the Root of the Bone in Edit Mode, **press Shift+S** again and choose **Selection to Cursor** to move the Root to the 3D cursor.



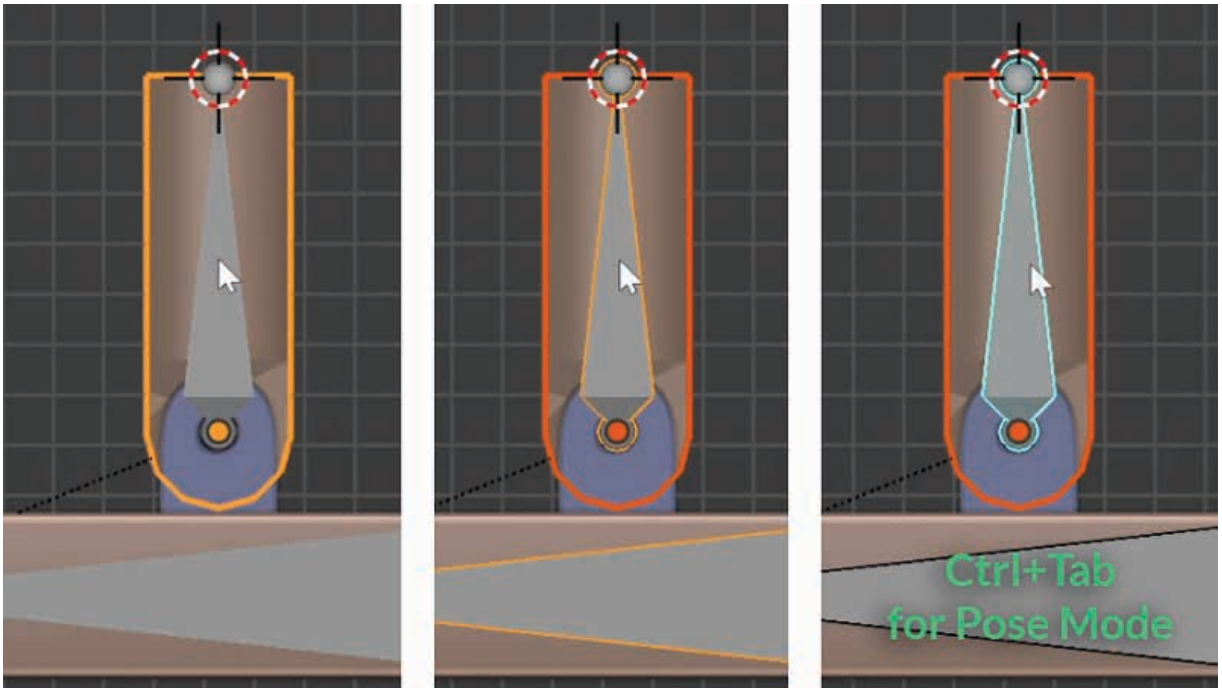
Repeat for the **Tip** of the Bone. Sometimes it's easier to select the right parts in **X-ray** view.



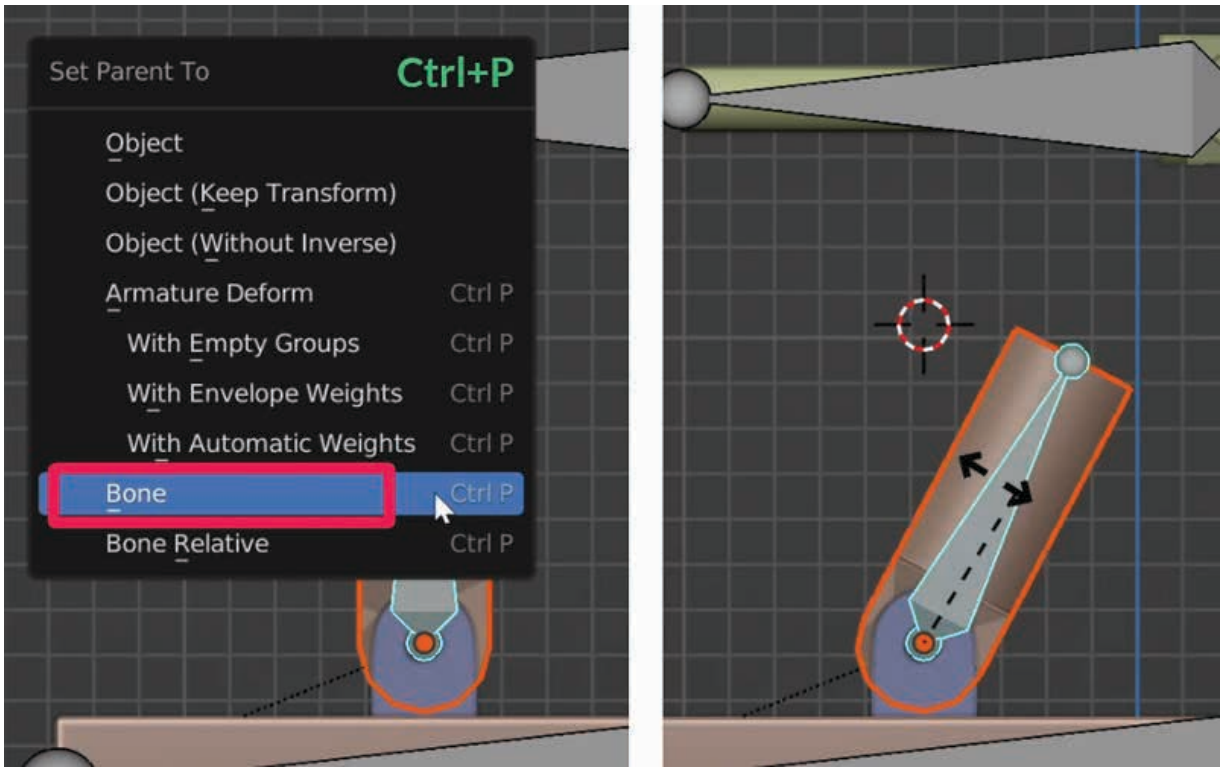
Repeat for the other piston. It doesn't matter which bone you duplicate to create the new bones for the piston.



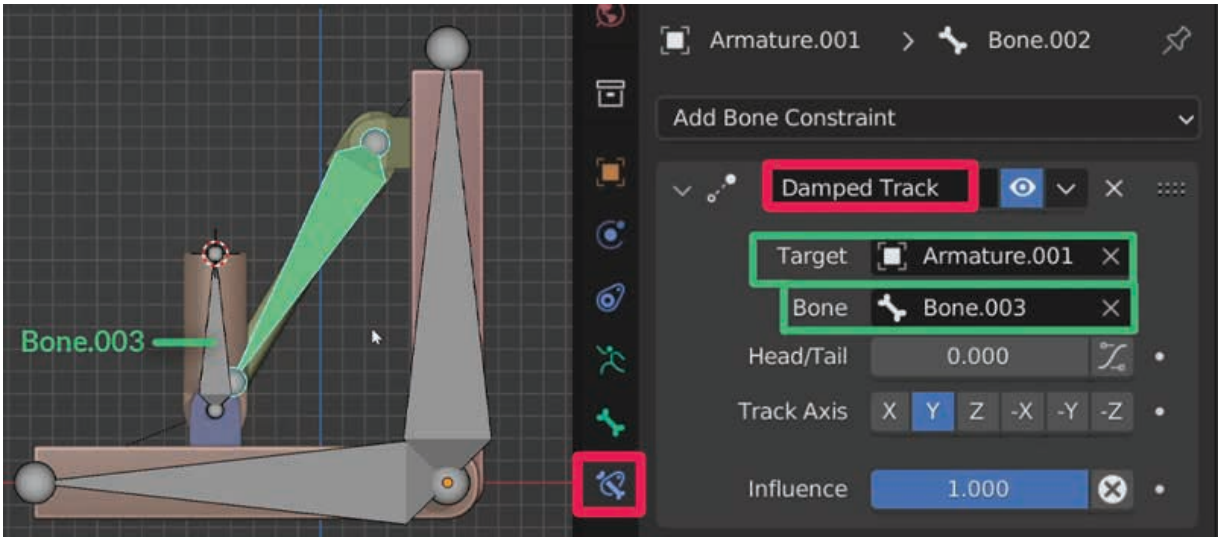
In Object Mode select a piston, **hold Shift** and select the Armature. In Pose mode, select the bone. You can quickly toggle Pose Mode / Object Mode by **pressing Ctrl+Tab**.



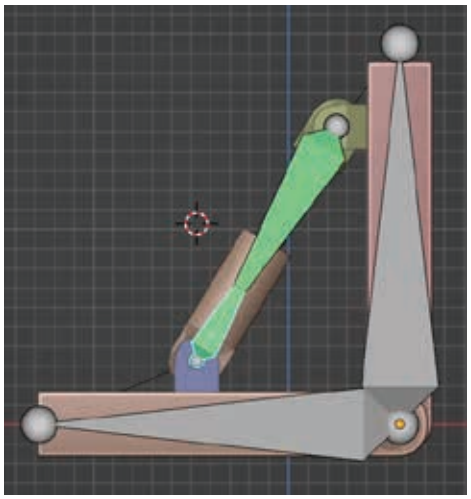
Press Ctrl+P and parent the piston to the bone. Moving the bone in Pose Mode now moves that piston part. Repeat for the other piston.



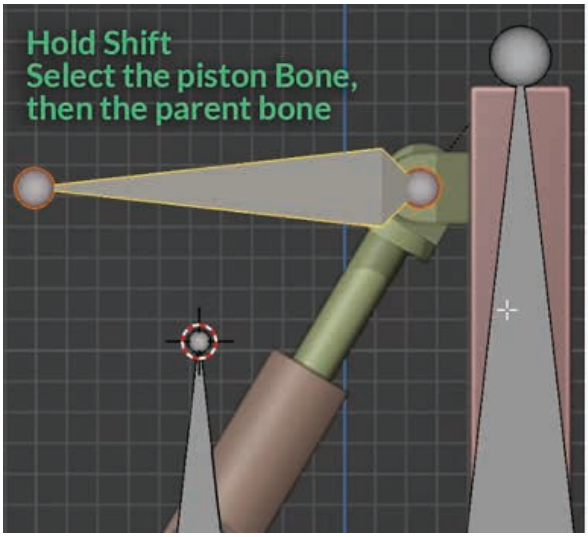
While you're still in Pose Mode, give both bones a **Damped Track bone constraint**. As the **Target** choose the Armature and as the **Bone** choose the bone of the other piston. You can find the bone name by selecting it temporarily and checking the top left corner of the 3D viewport.

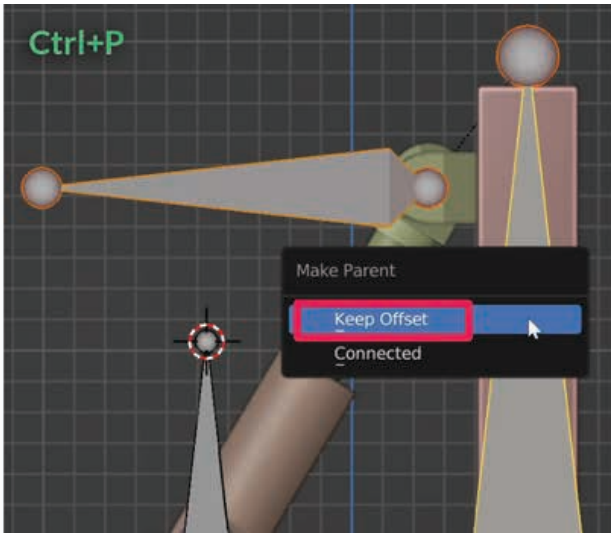


When both piston bones have a Damped Track constraint pointing at the other bone, the pistons will always point at each other.



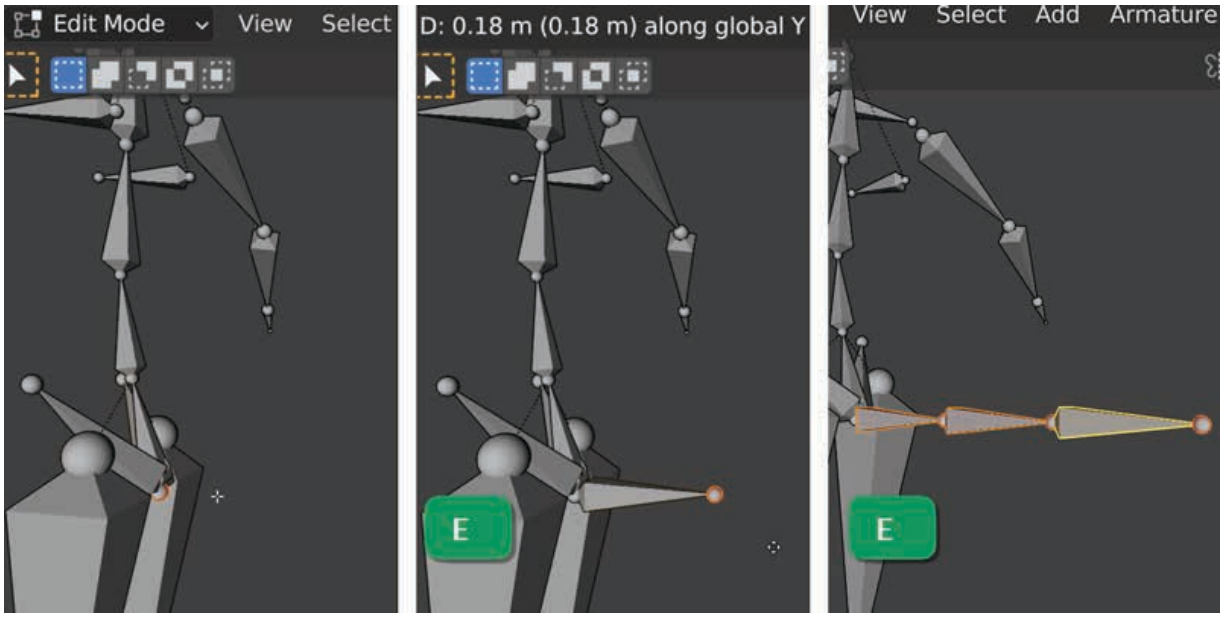
In Edit Mode you can parent the piston bones to any other bones by **pressing Ctrl+P** and choosing **With Offset.**



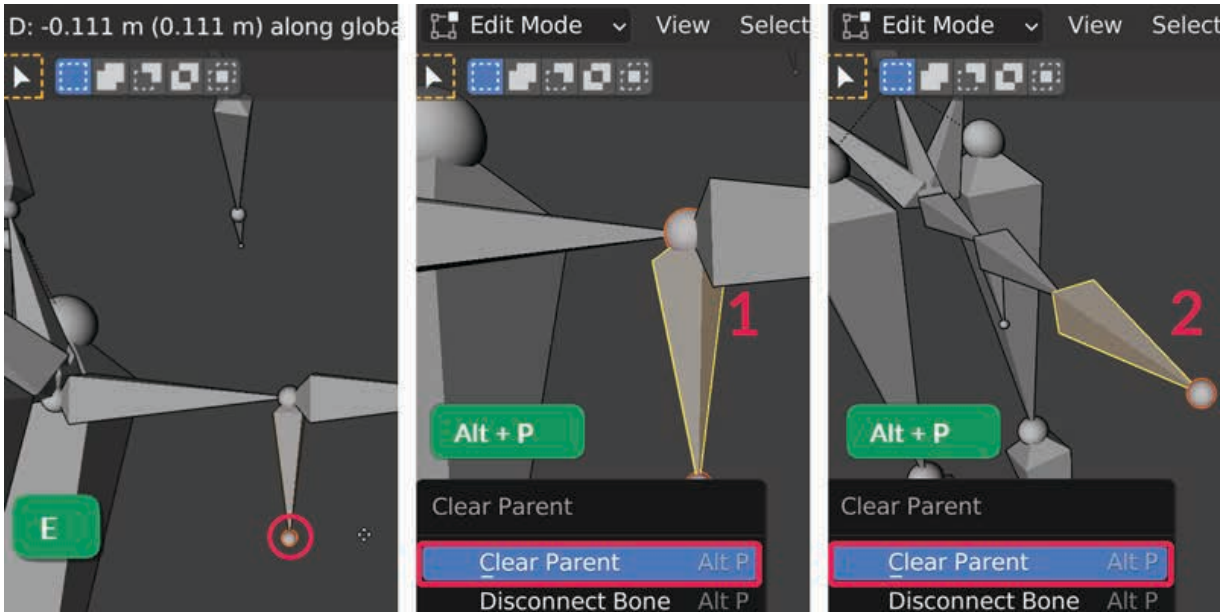


Adding and rigging a tail for your character

Adding a tail to your character is easy if you follow these steps. Download the free Blend file here if you want to see how it's all set up. Extrude three bones from the character by **pressing E**.

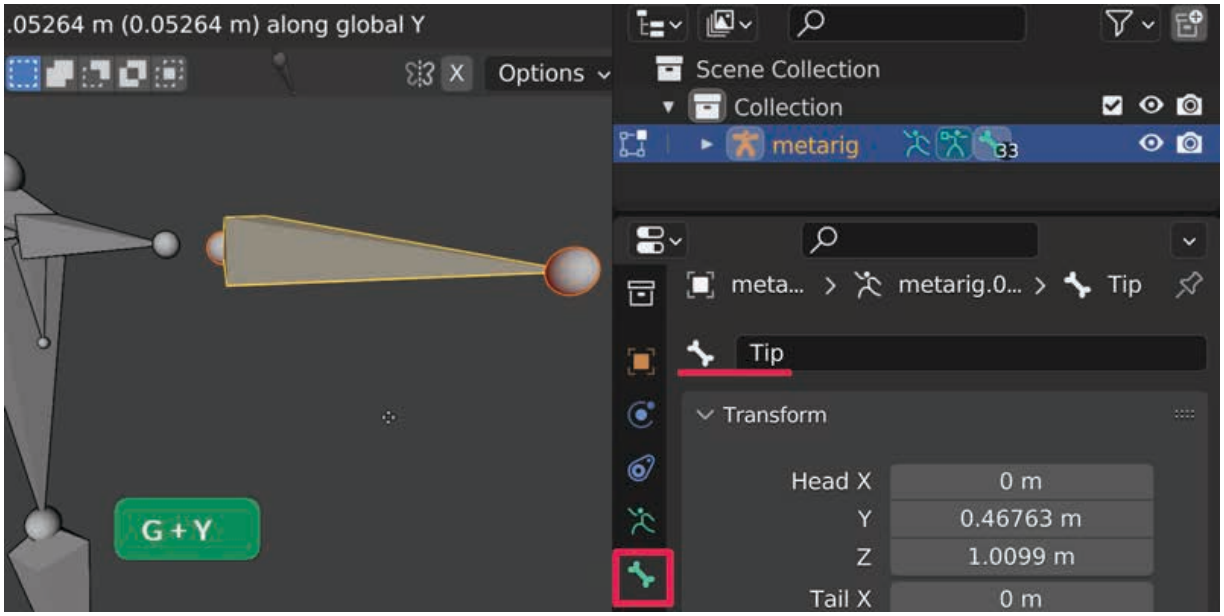


Also extrude a bone down after the first bone. Select these 2 bones and **press Alt+P** to clear the parent relationship one bone at a time.

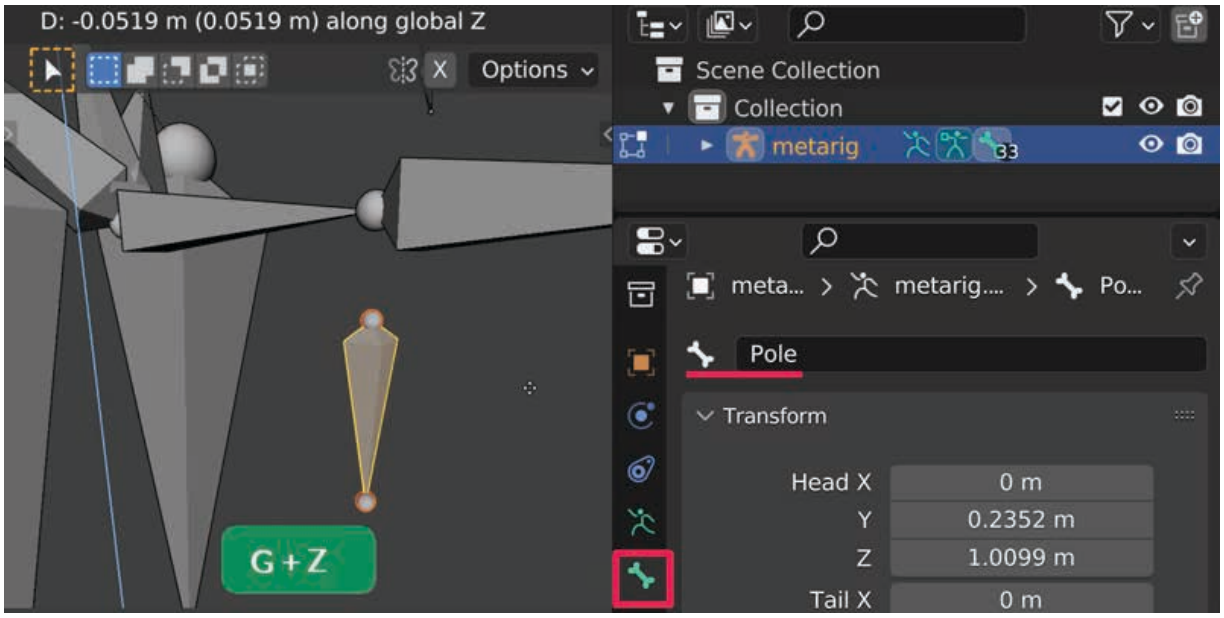


Name the last one tip and the one pointing down pole.

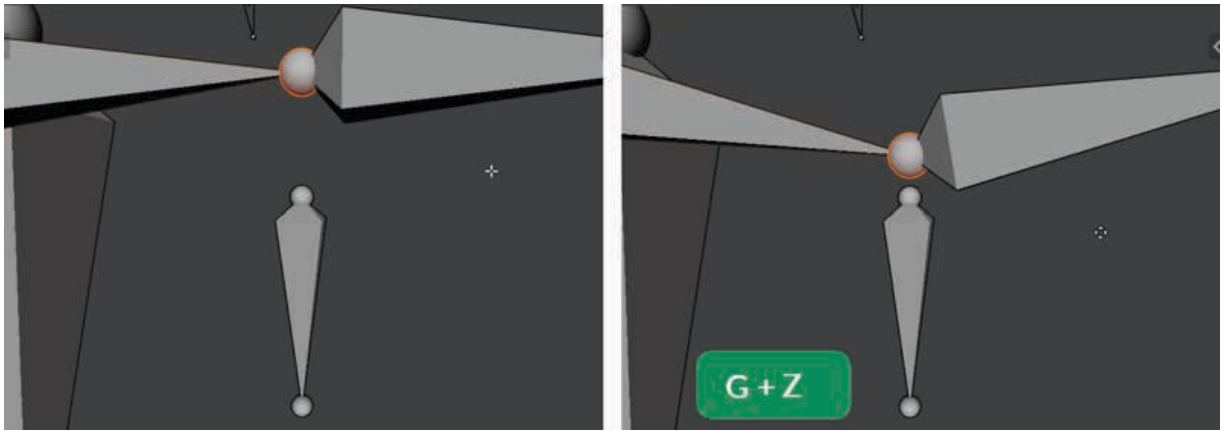
Name the last one “Tip”.



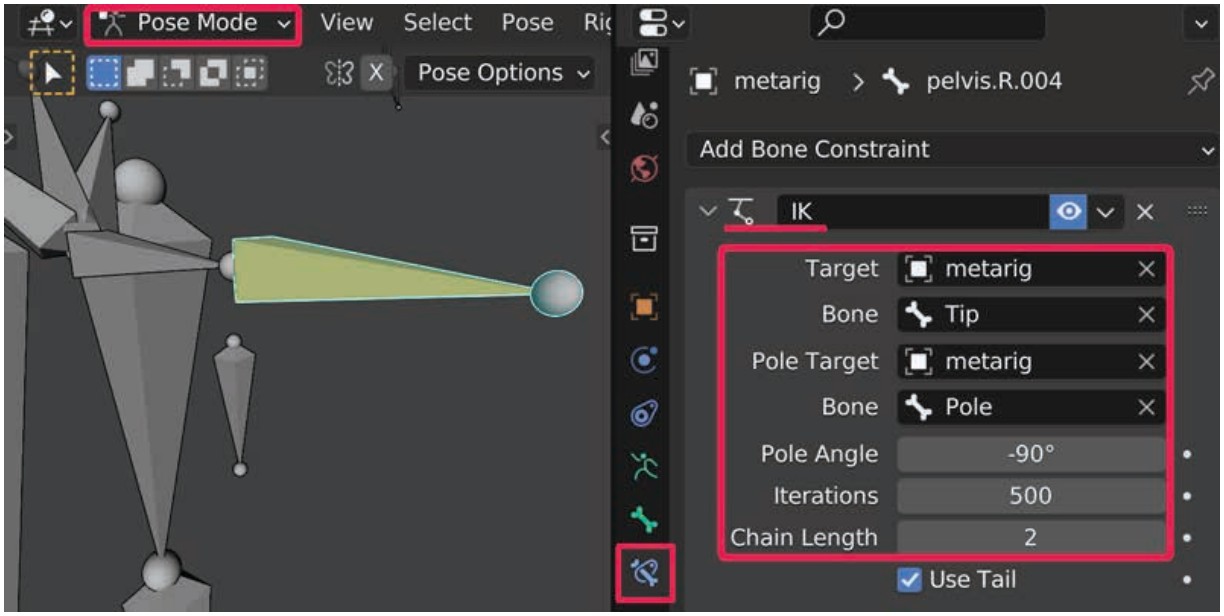
Name the one pointing down “Pole”.



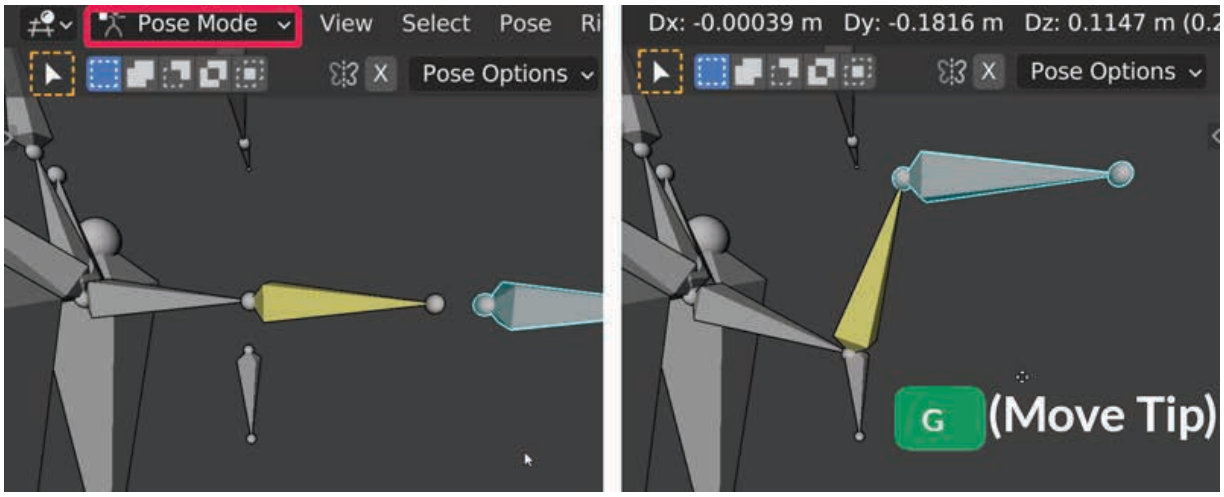
Make sure there is a slight bend in the tail towards the pole.



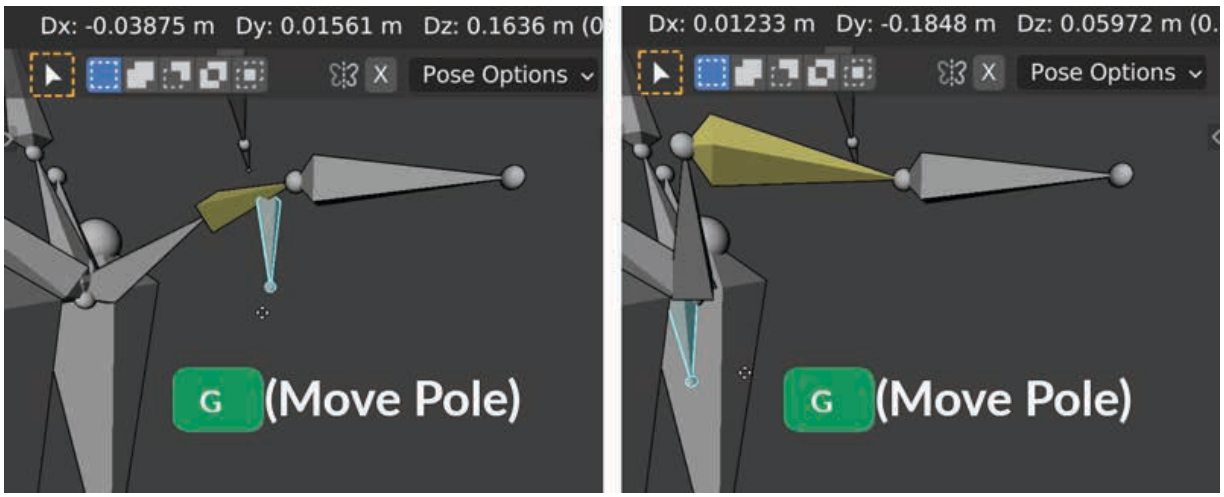
Add an **IK Bone Contraint** to this bone and choose the **Tip Bone** as the **Target**. Increase the **Chain Length** and if necessary change the **Pole Angle** in 90 degree increments until it all points in the right direction.



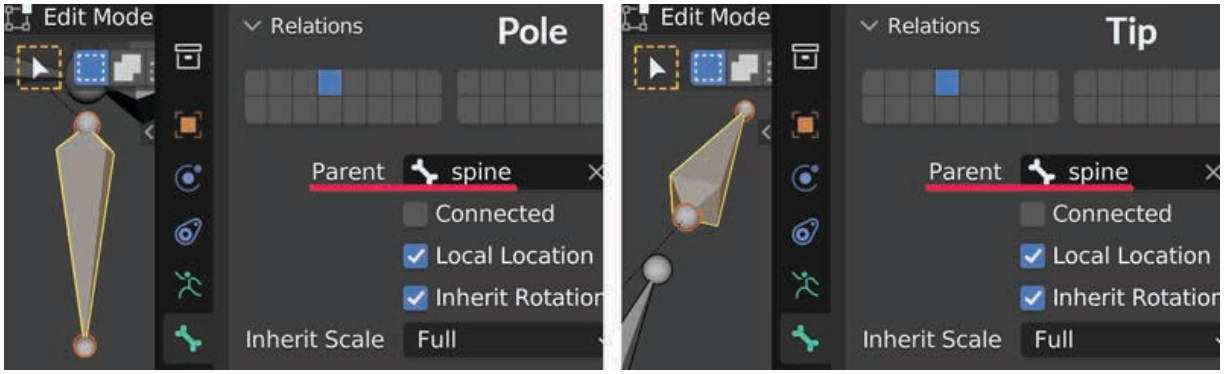
Now you can use the **Tip** bone as well as the **Pole** bone to control the tail.



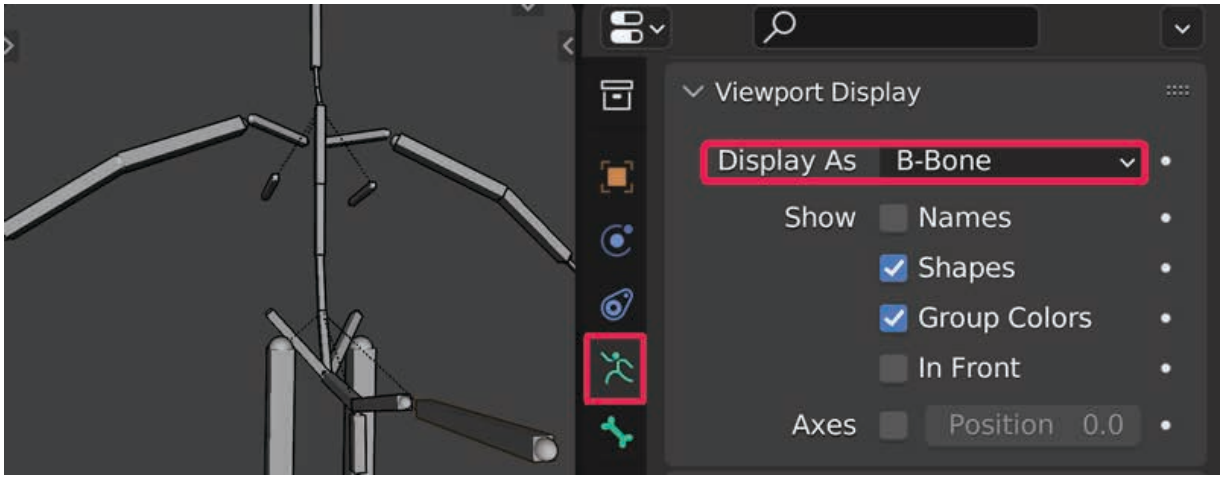
Press G in Pose Mode to move the bones.



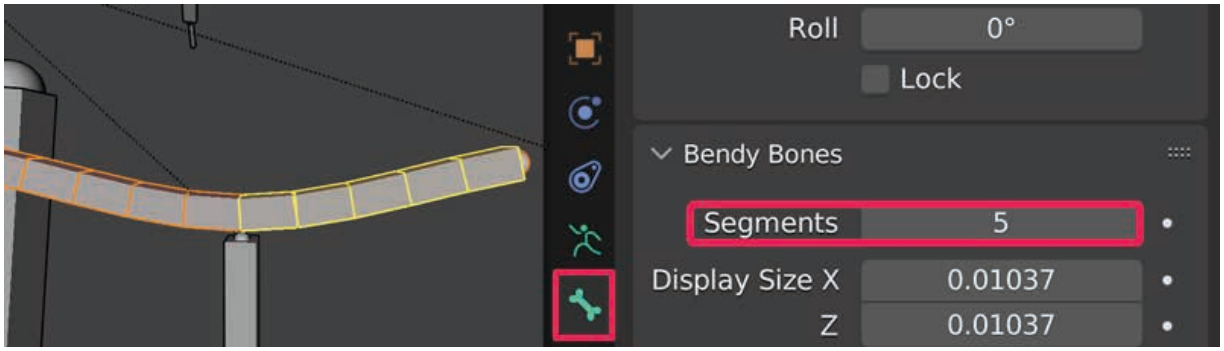
In edit mode parent the tip and pole bones to the spine bone.



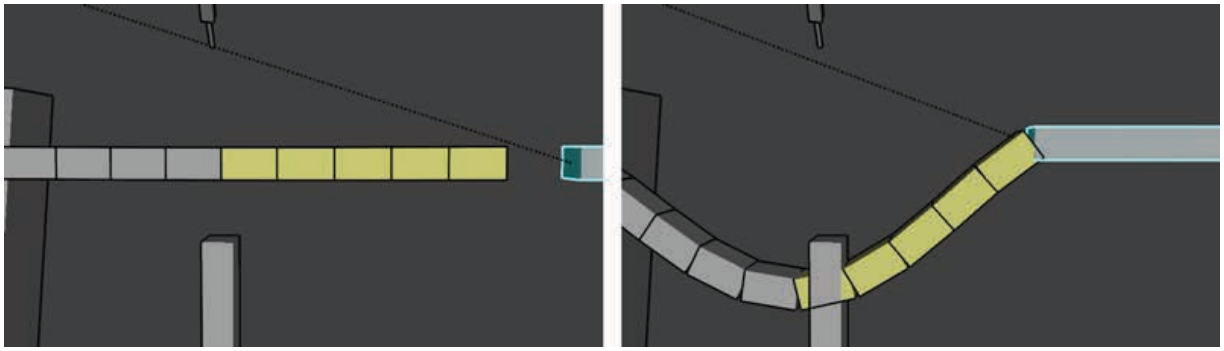
In the Armature panel, set viewport display to **B-Bone**.



Select the bones in Pose Mode and in the bone tab increase the Bendy bones segments.



Thanks to the Bendy Bone segments, you get a nice curved bone chain when moving the Tip bone.



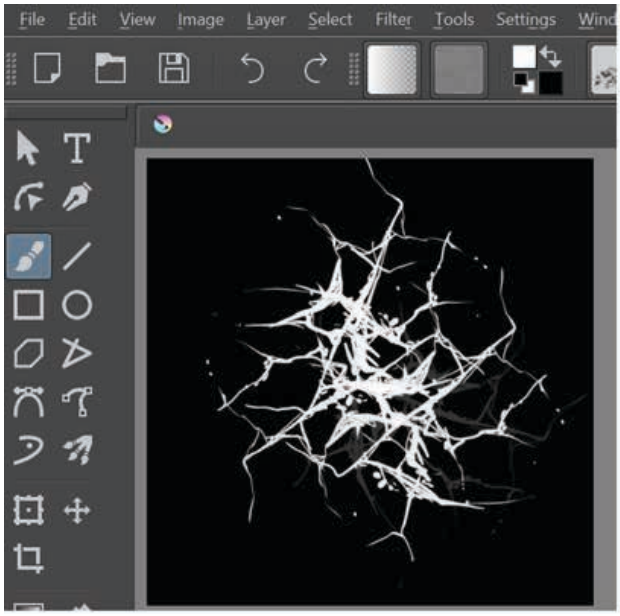
TEXTURING

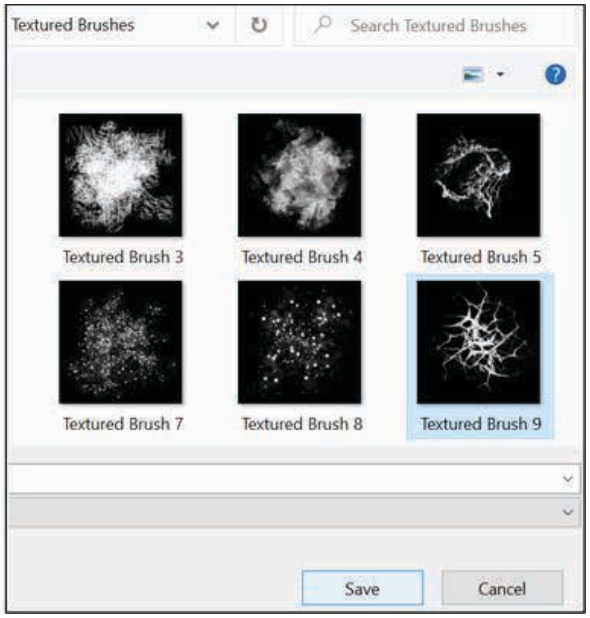
Manually Paint Edge Wear

"Do everything by hand, even on the computer" - Hayao Miyazaki

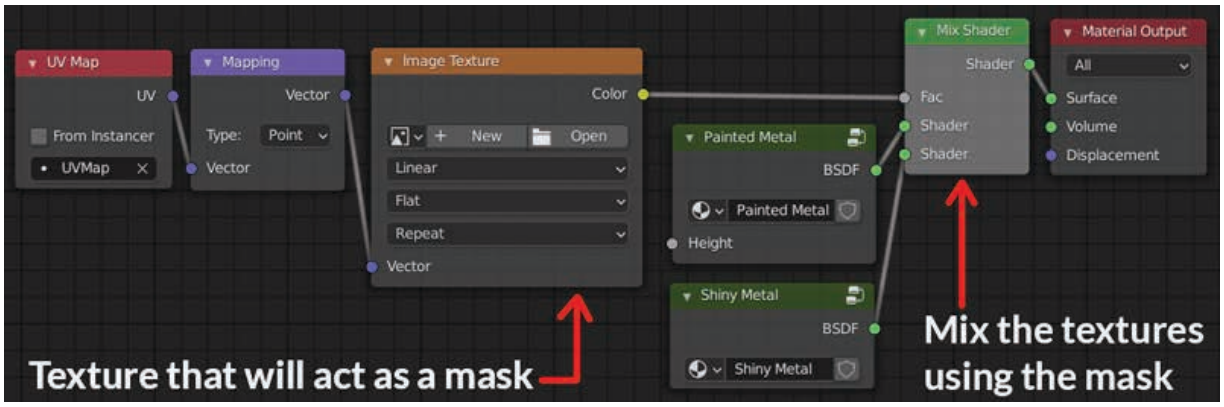
Manually painting a mask for edge wear can be a good alternative to using a procedural method, as it can add a bit more of a hand-made look to your model. But Blender only comes with a simple round brush. Fortunately you can create your own brushes.

Make a custom brush in the painting app of your choice (in this example I used Krita). This custom brush consists of some white on black brush strokes in a square format. Save it as a black and white PNG image.

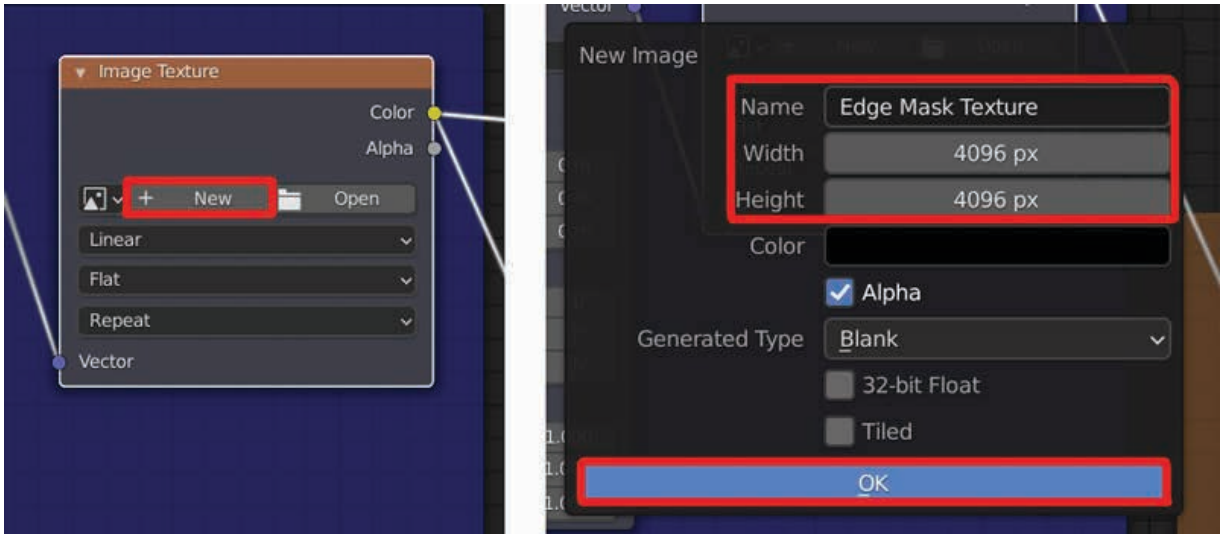




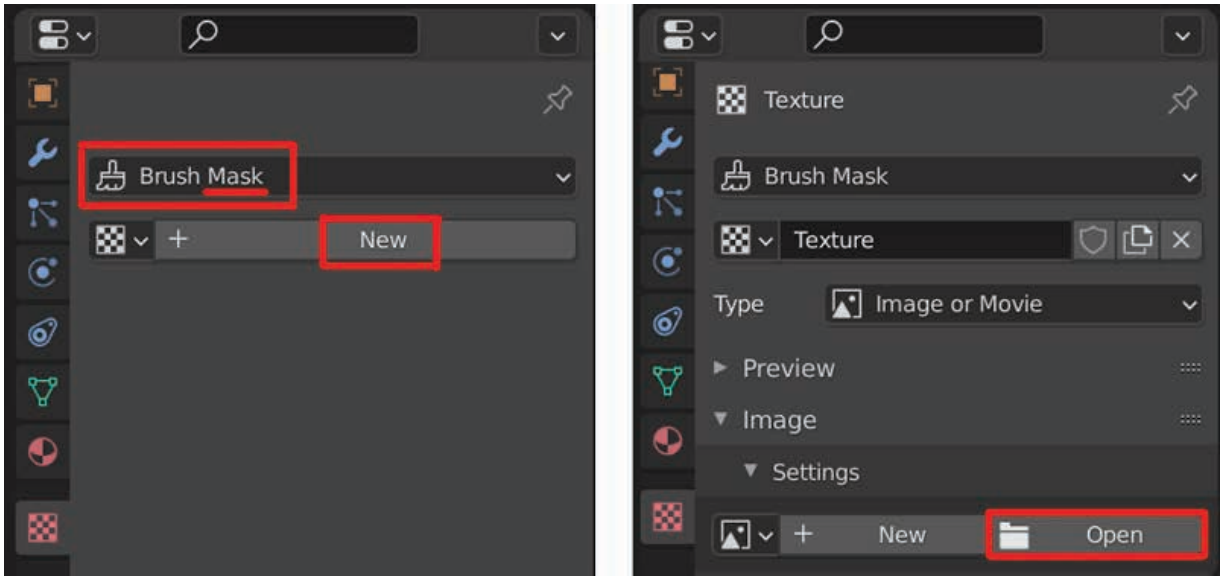
In Blender connect two materials (in this case, shiny bare metal and painted metal) with a **Mix Shader** to blend them. Here I collapsed the two materials to Node Groups to make it visually not too complex. The main thing to pay attention to here is the Mix Shader that uses a texture to mix the two materials.



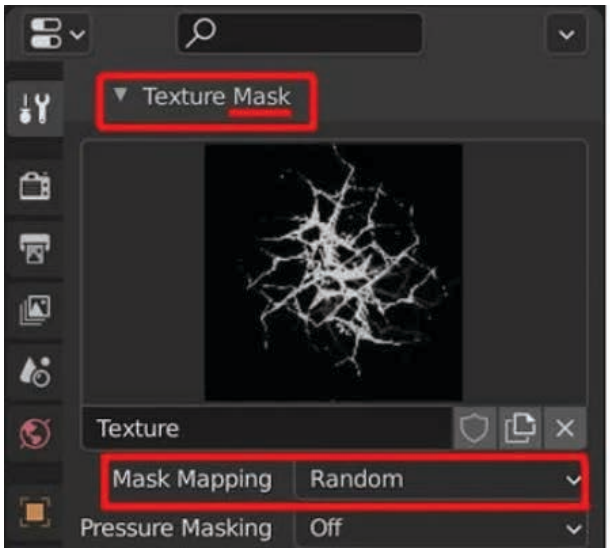
Give that Mix Shader a **new texture**. Make sure the resolution is high enough - you can always decide to reduce the resolution later. I would go with 4096 x 4096. Preferably also give it a clear name, “Edge Mask Texture” makes sense.

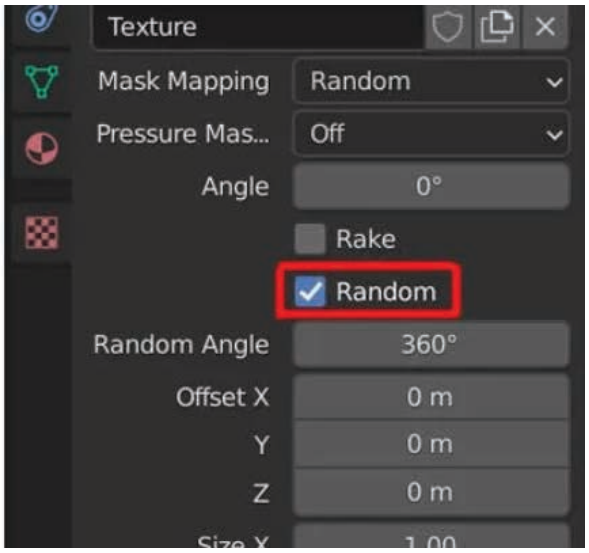


Go to the Texture tab, and from the drop-down menu choose the Brush Mask. Click on New. Then click on Open and load the black and white texture that you made in Krita earlier.



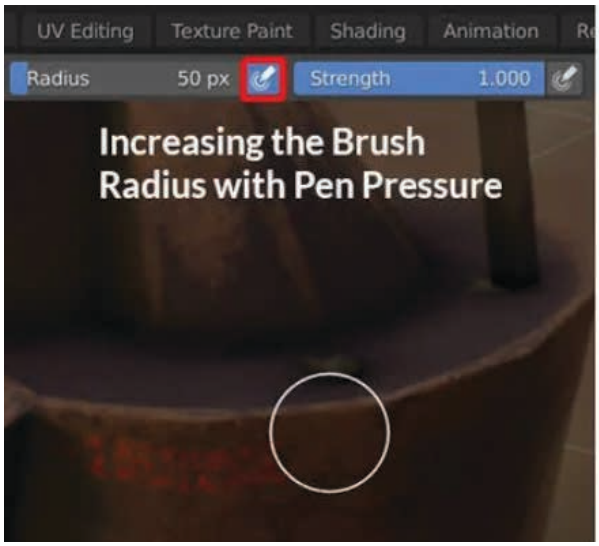
Enable Texture Paint mode. Then in the Active Tool Workspace in **Brush Settings** under **Texture Mask** you should see the brush you created in your painting app. Set **Mask Mapping to Random**. Also check the **Random** option for random rotation.



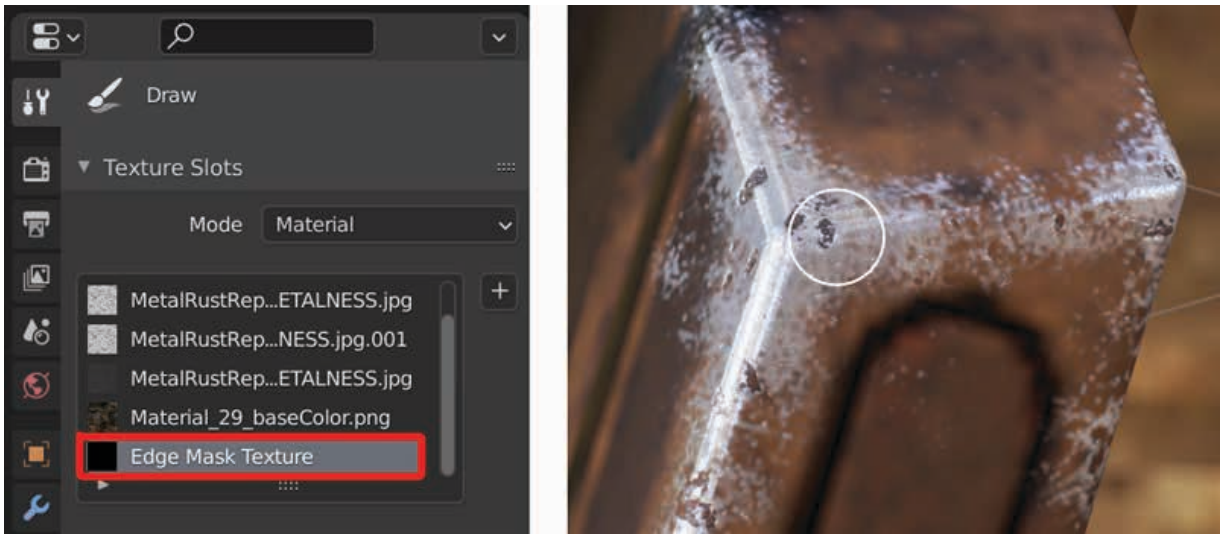


If you use a tablet pen, you can **enable Pressure** for the **Radius**.

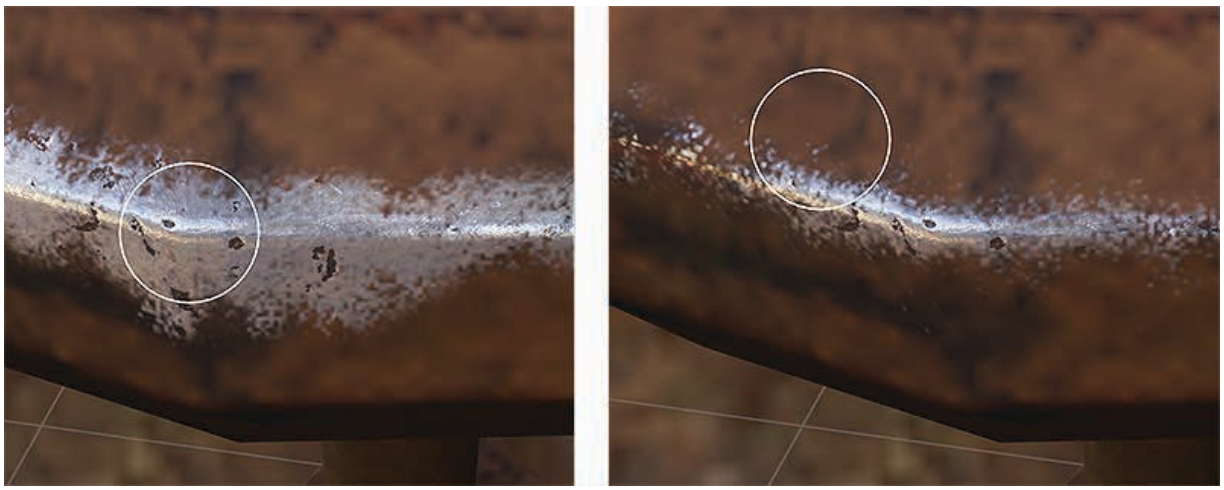
I wouldn't recommend using pen pressure for Strength, instead just leave it at 1. If you don't have a tablet pen (or you're just lazy like me) you can definitely also paint edge wear using your mouse! It doesn't have to be perfect, after all. By **pressing F** you can also quickly adjust the radius of your brush.



Make sure you've got the correct texture (the new "Edge Mask" texture that you created) selected. Now you can use this brush to paint a mask that mixes two other textures in a nice organic way.



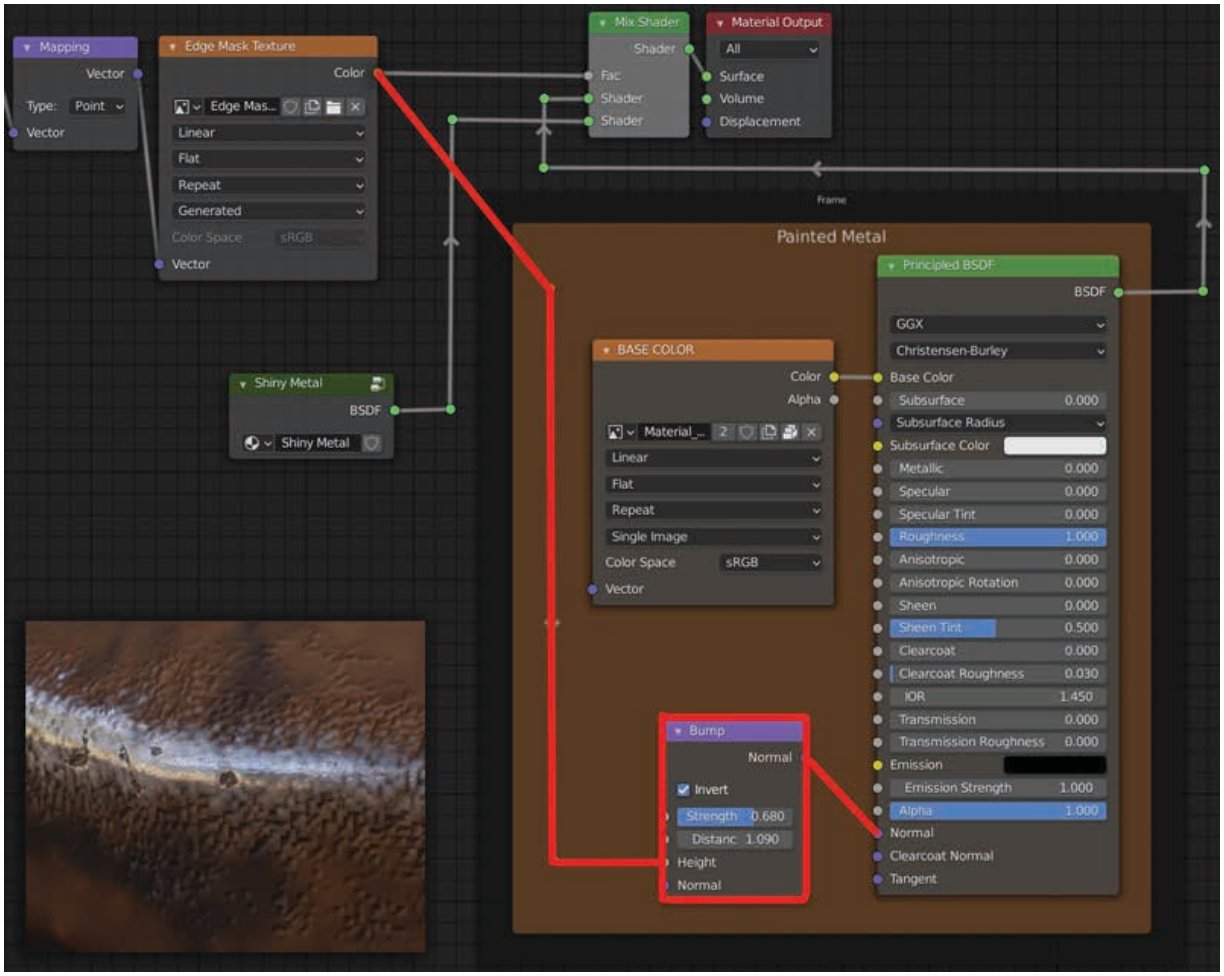
For best results with adding weathering effects like edge wear, I recommend first adding a bit too much of it, and then erasing from that with a textured brush by **holding Ctrl** while painting.



You can use the same mask for bump, to have some subtle difference in height between the paint layer and the exposed metal underneath.

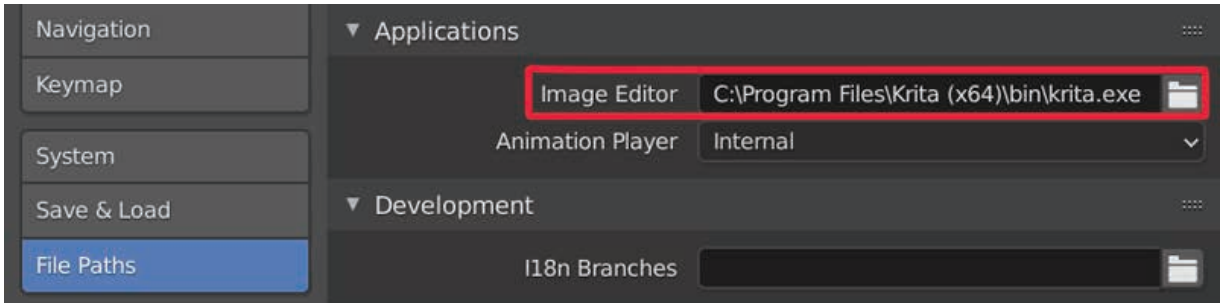
For this, add a **Bump node** and plug the output of the mask in it, then experiment with the values until it looks good. For best results here, you do need the texture to have a high resolution, otherwise you'll see the artifacts of the image compression in your bump detail.

Below: the node set-up to add some bump to the worn edges. You can see from the inset picture that the resolution of the texture is not high enough in this example, resulting in rather pixely bump details.

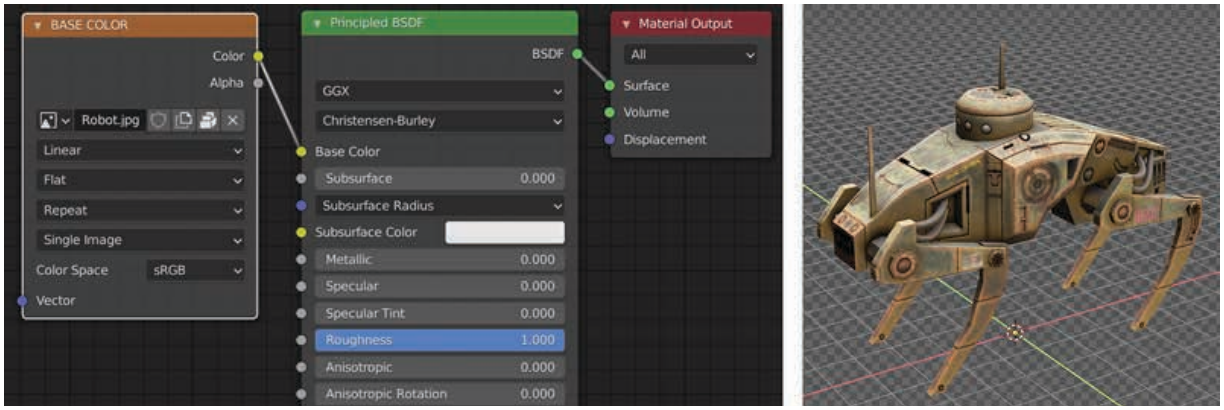


Texture Painting with Quick Edit and Krita

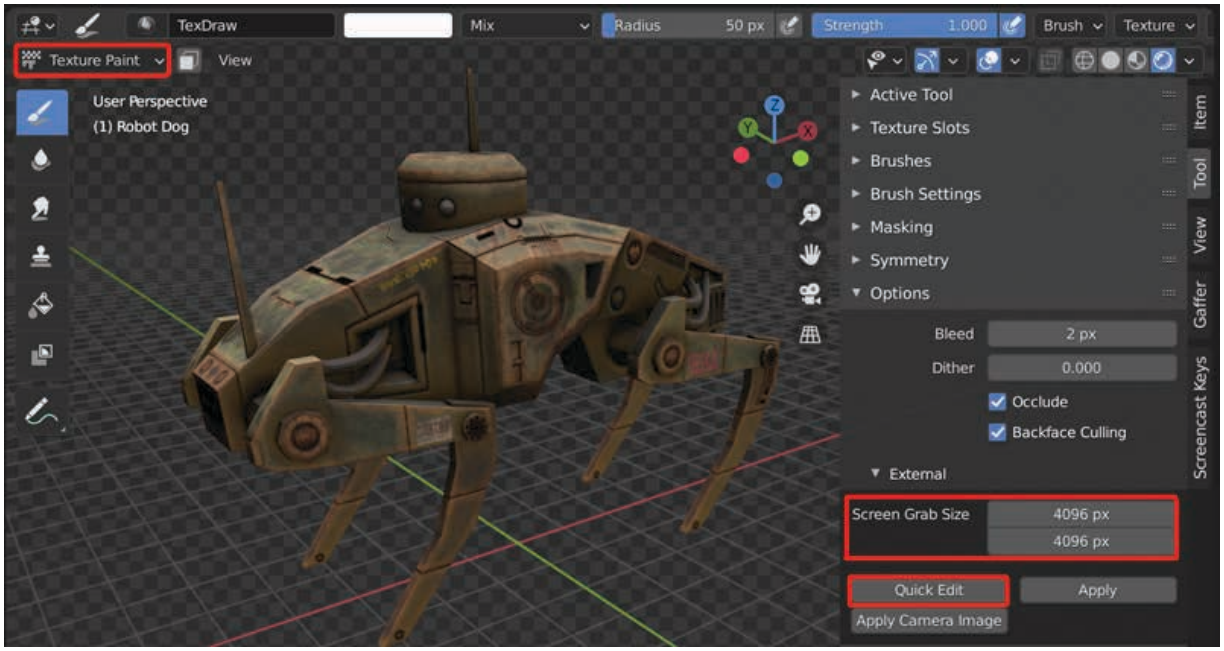
This is an artist-friendly method of manually painting textures, using the not so well known "**Quick Edit**" texturing feature. In Preferences under **File Paths** set the external image editor to Krita (or any other 2D painting app you like, such as Photoshop!)



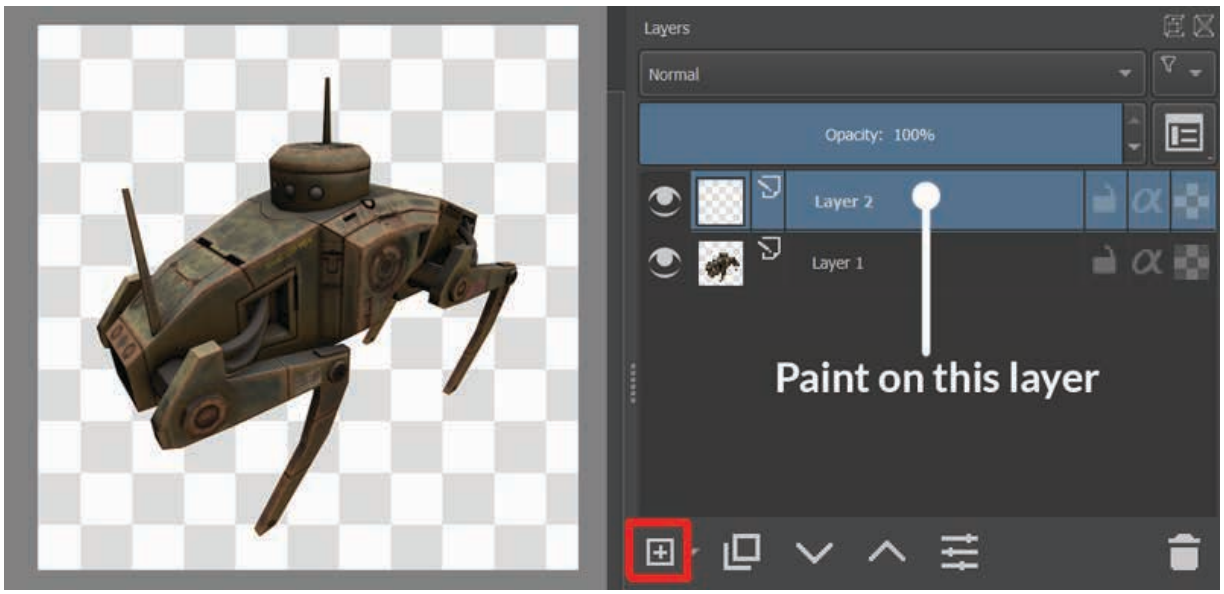
Give the object a material and a texture if it doesn't already have one. This low-poly robot dog by Renafox on Sketchfab already has a simple texture.



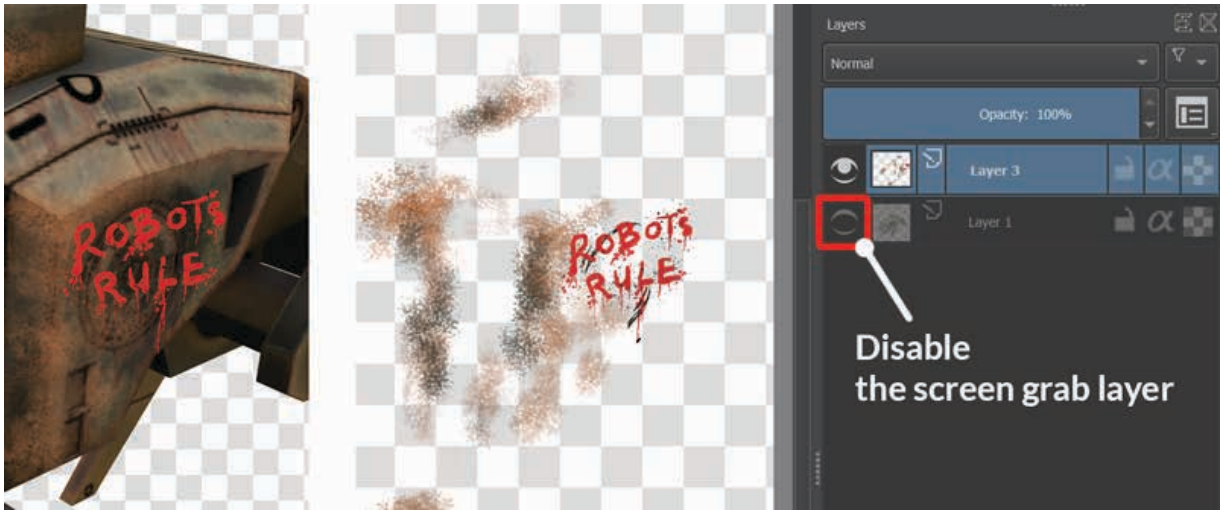
Open the Option Panel by **pressing N**. In Texture Paint mode go to Tool > Options and under Screen Grab Size make sure the resolution is not too low - 4096 x 4096 would be my choice. **click on Quick Edit**.



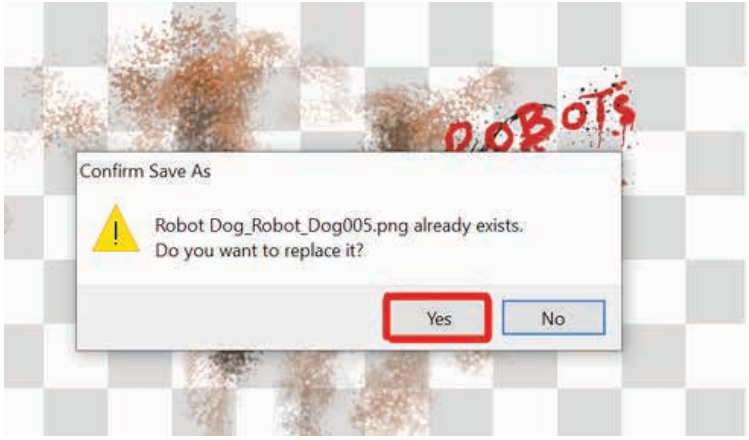
Blender now opens a screen grab of the model from the angle that you were seeing it from in the Blender Viewport, in Krita. Create a **new layer** to paint on.

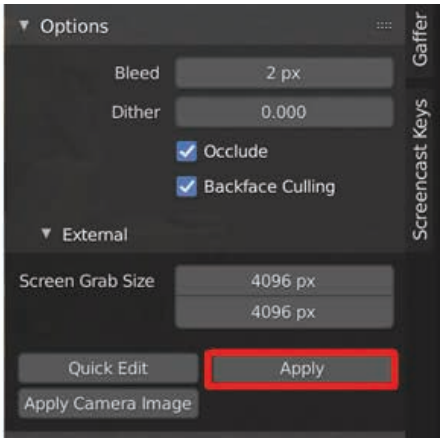


Now you can paint on the new layer to add something to your texture. When you're finished, disable the screen grabbed layer and save the image.

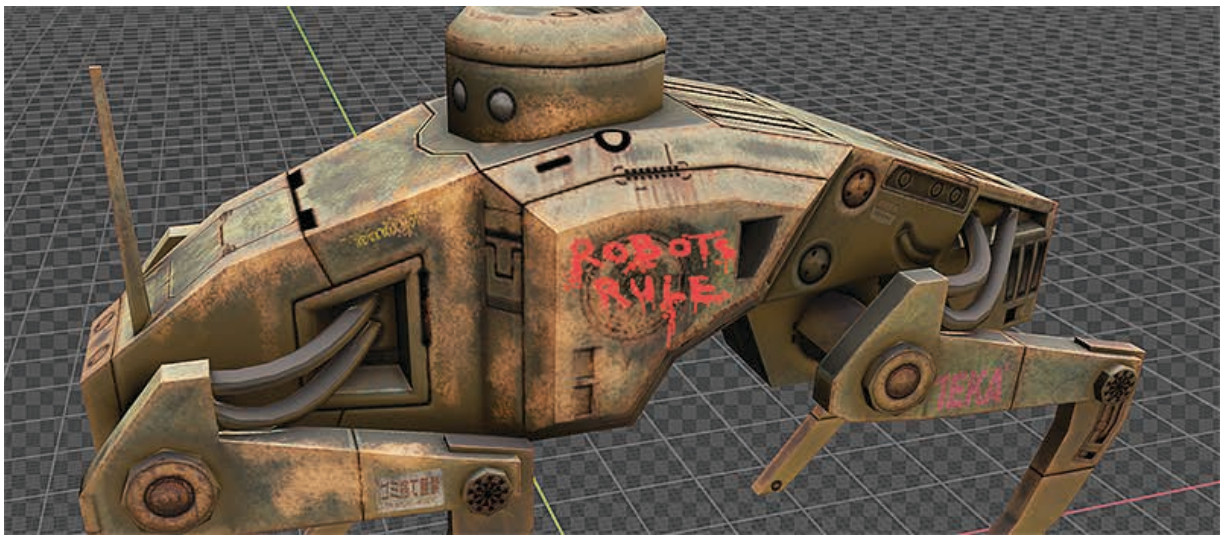


When Krita asks if you want to **replace** the original file, click on **Yes**. Now back in Blender, in the Option panel next to Quick Edit, **click on Apply**.



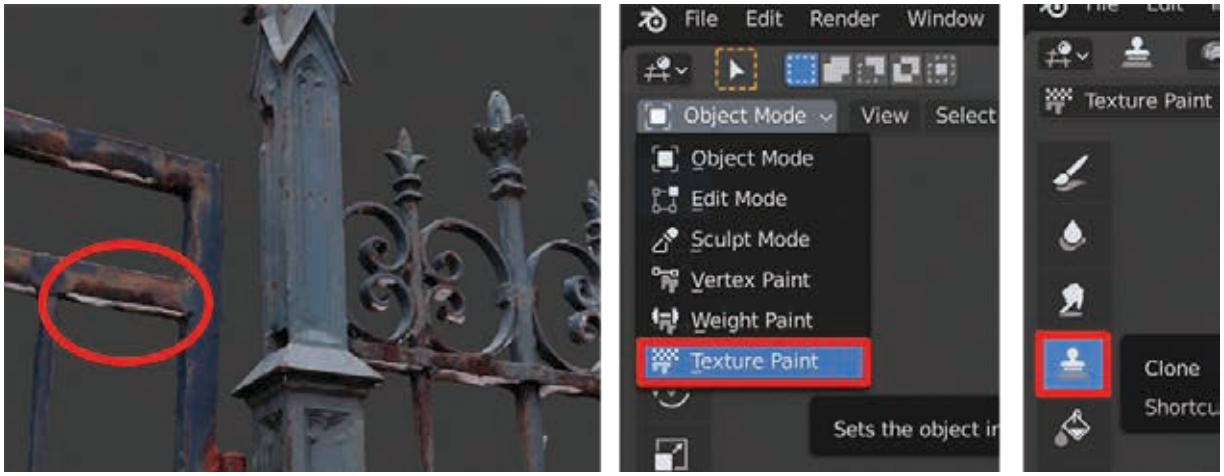


Now the paint you added in Krita is visible on your model in Blender (providing that you're in Material or Rendered preview).



Texture Clone Painting

If you have a textured photogrammetry model like this that needs some small texture fixes, you can use the Clone tool. Go to Texture paint mode and select the **Clone Tool**.

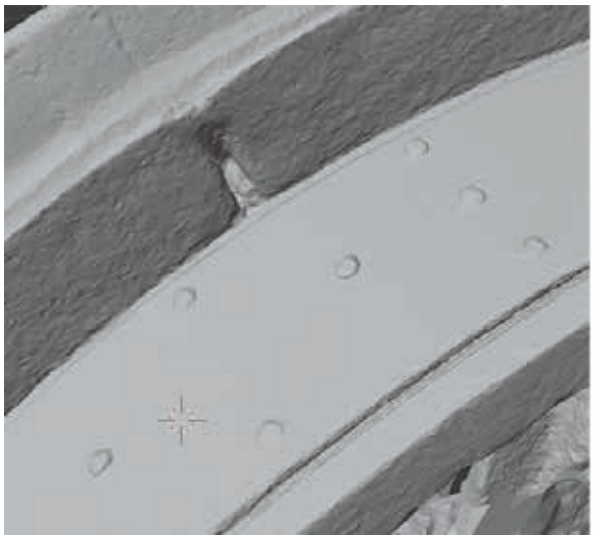


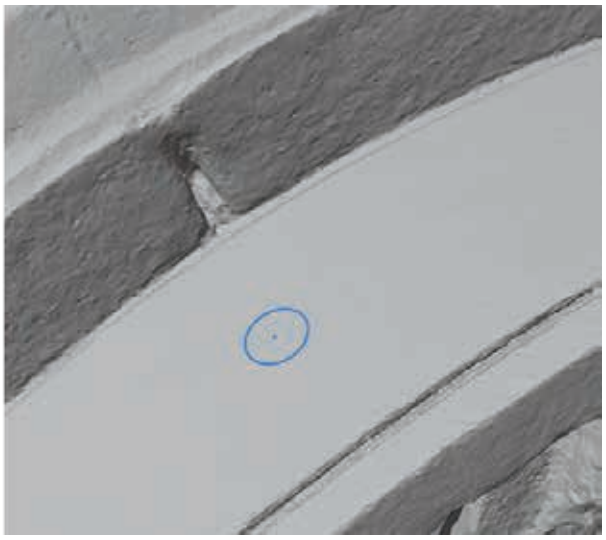
Shift+right-click on the object where you want to sample from, then paint where you want to apply it.

Using this simple technique we can quickly fix the texture issues on the model where the photogrammetry process didn't have enough data.

Keep in mind that this does not change the geometry, so you can still sometimes see a shadow of something you painted away as it also has some height.

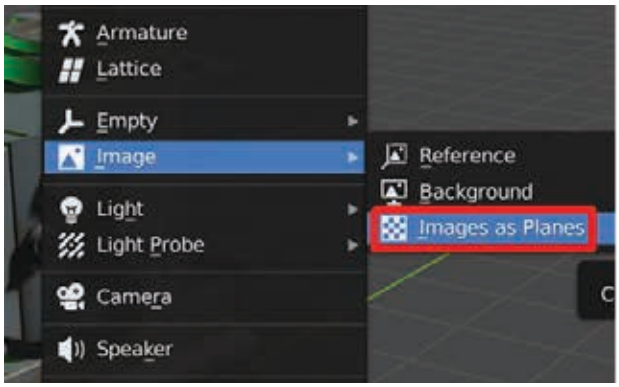
In that case you can switch to Sculpt mode, and **holding Shift** you can smooth that area.

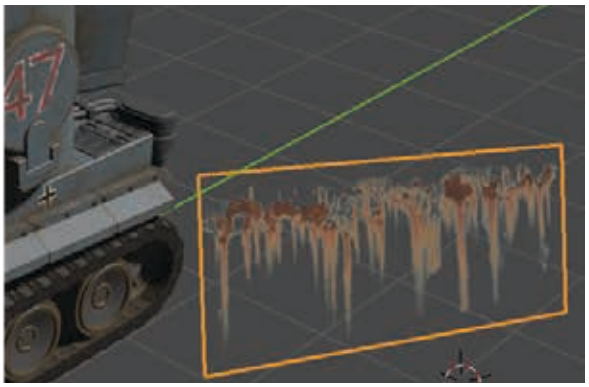




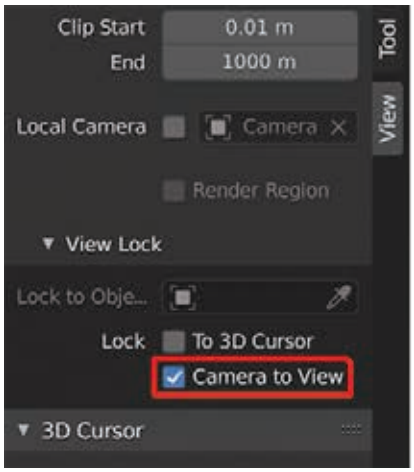
Texture Clone Painting - Clone from one object to another

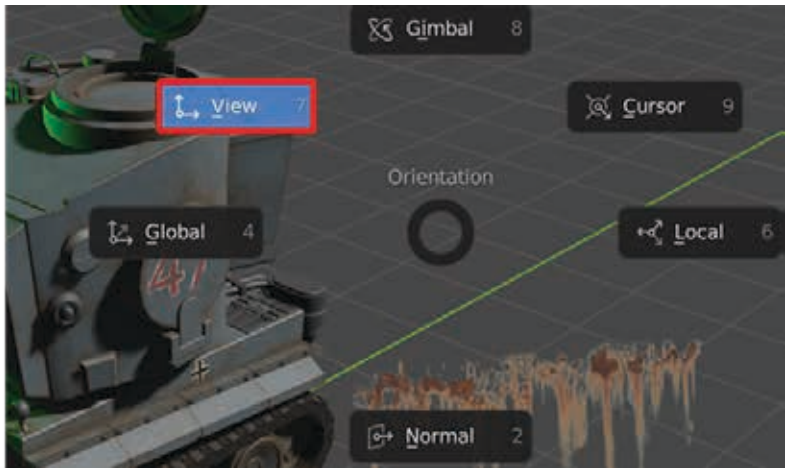
Instead of texture cloning from the same object, you can also clone from an image. Add an image you want to clone from as plane with the Images as Planes add-on. I'm using this rust texture with an alpha (meaning the background is transparent) from Textures.com.



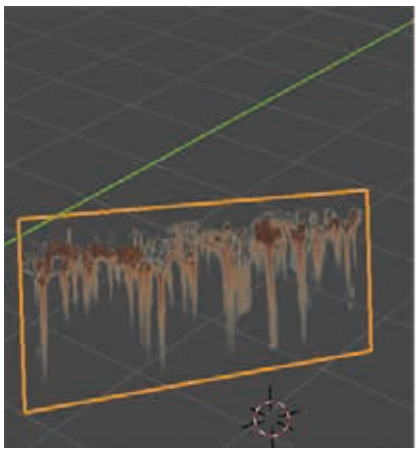


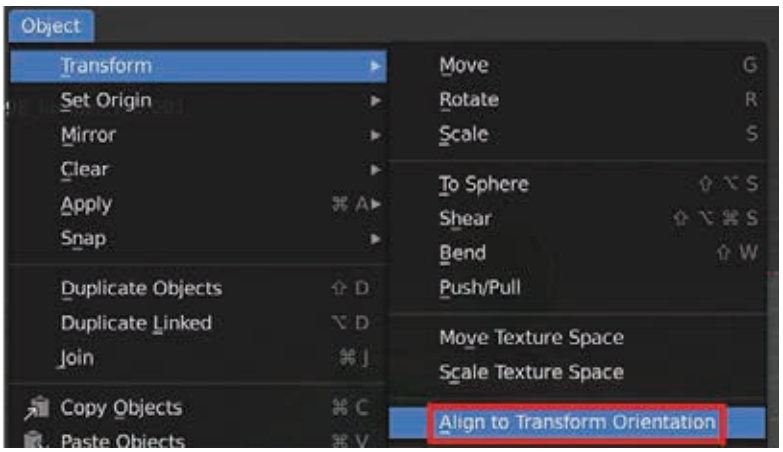
Now, before the fun starts - if you want to keep the plane constrained to the view, check "Lock Camera to View" in the Option panel. This way you can move the camera view as if you would the viewport view. Then **press "," (comma)** and choose **View**.



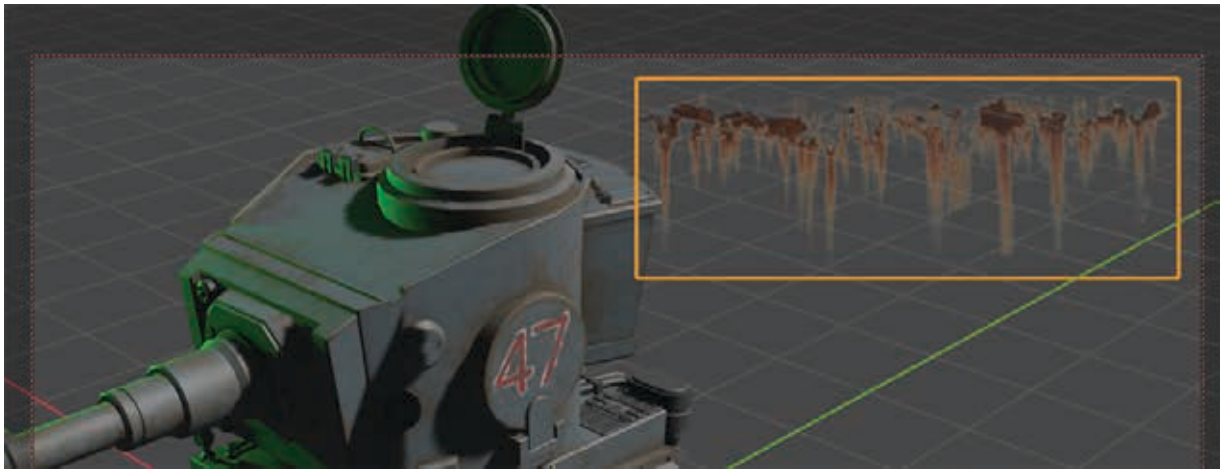


Then select the image plane and go to Object > Transform > **Align to View orientation**.

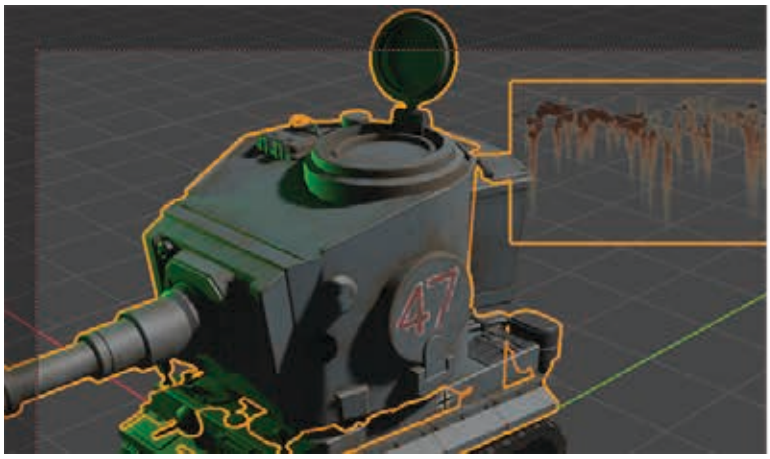


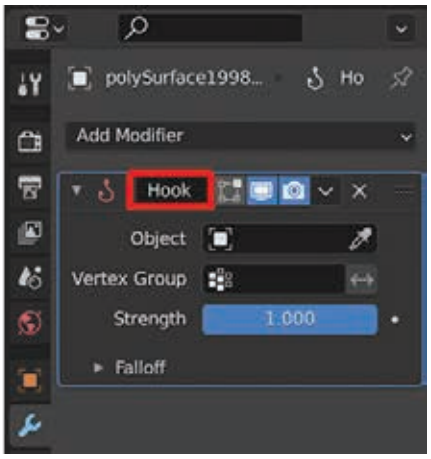


Whew, all that work just to align the image plane to the camera! But it's important for ease of use later. Once you've done these steps a few times, it becomes quick and second nature. Unfortunately we can't clone paint from one object to another, so we'll need to join these objects.

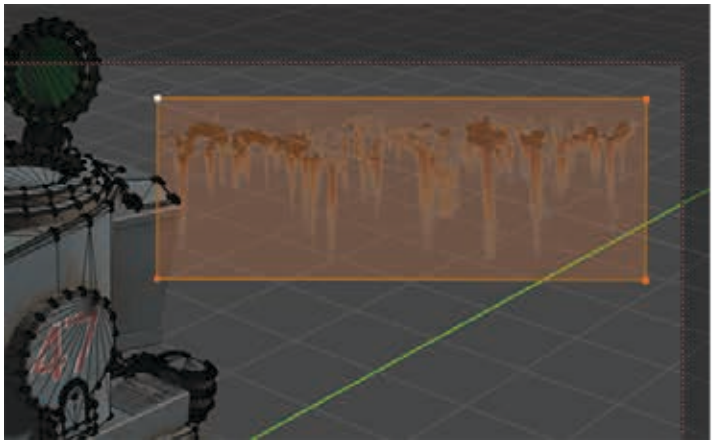


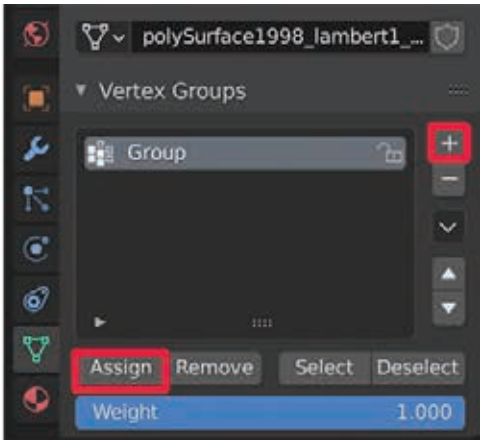
Okay, but now when we move the camera, the plane is no longer aligned. At this point you're probably ready to flip your table over, but don't worry - there's a solution! We can join the objects and still keep the plane aligned to the camera. To solve this, add a **Hook modifier** to the joined objects.



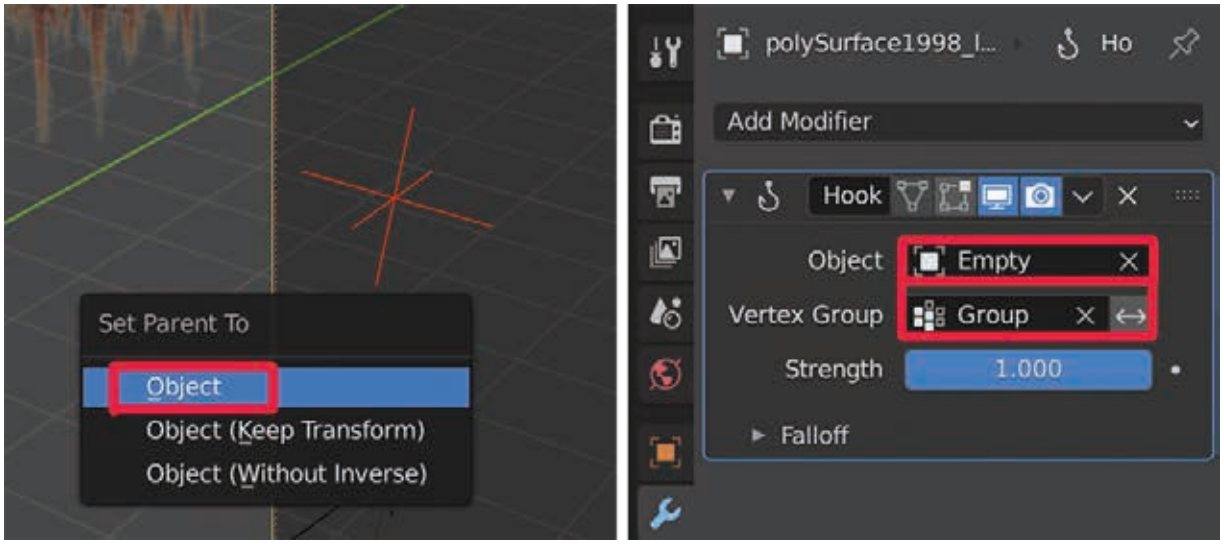


As you can see, the Hook modifier has a couple of empty fields: **Object** and **Vertex Group**. Let's fill them. Assign the vertices of the plane to a vertex group.

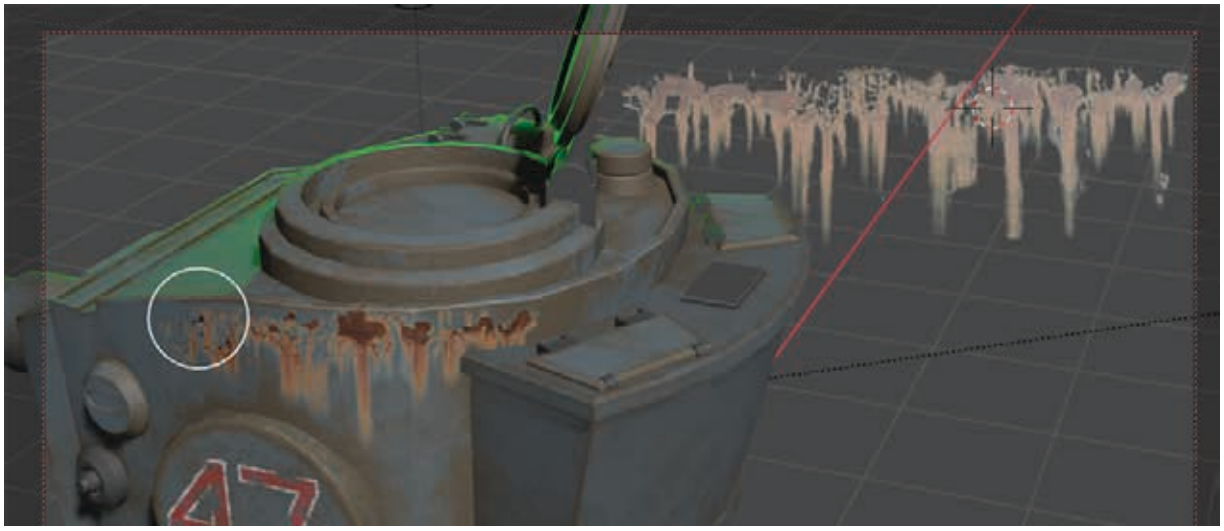




Add an Empty and parent it to the camera by selecting the Empty, **holding Shift** and selecting the Camera and **pressing Ctrl+P > Object**. It doesn't matter where the Empty is, so it's most convenient to keep it outside of the camera frame so it's not in the way. In the Hook modifier choose the **Empty** and the **Vertex Group**.



Now you can move the camera around the object and the plane will stay aligned to the camera so that you can clone paint from it.



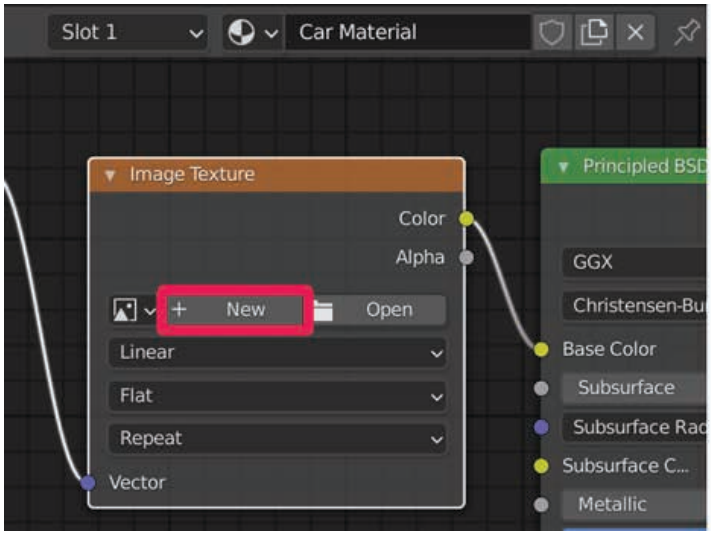
If it bothers you that the image plane is sometimes in the shadow (depending on how you rotate the camera) you could parent a lamp to it. This doesn't affect the way it's painted onto the model. Once you're finished clone painting, you can delete the image plane in Edit Mode.

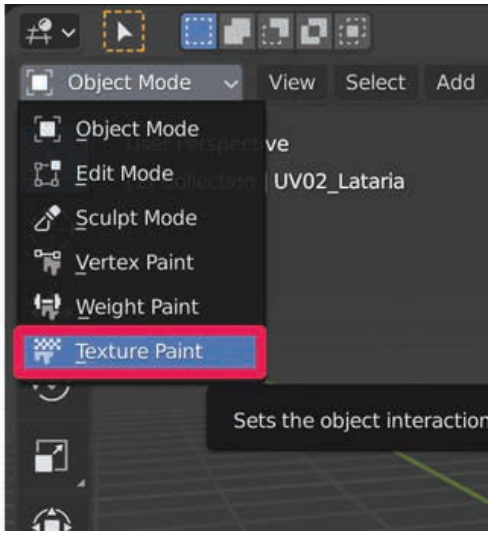
Texture Painting Basics

Texture painting in Blender by default is quite basic, featuring just one brush. But with some tweaking of the settings, you can get really nice results and use complex textured brushes just like in a 2D painting app. First, let's look at the basics.

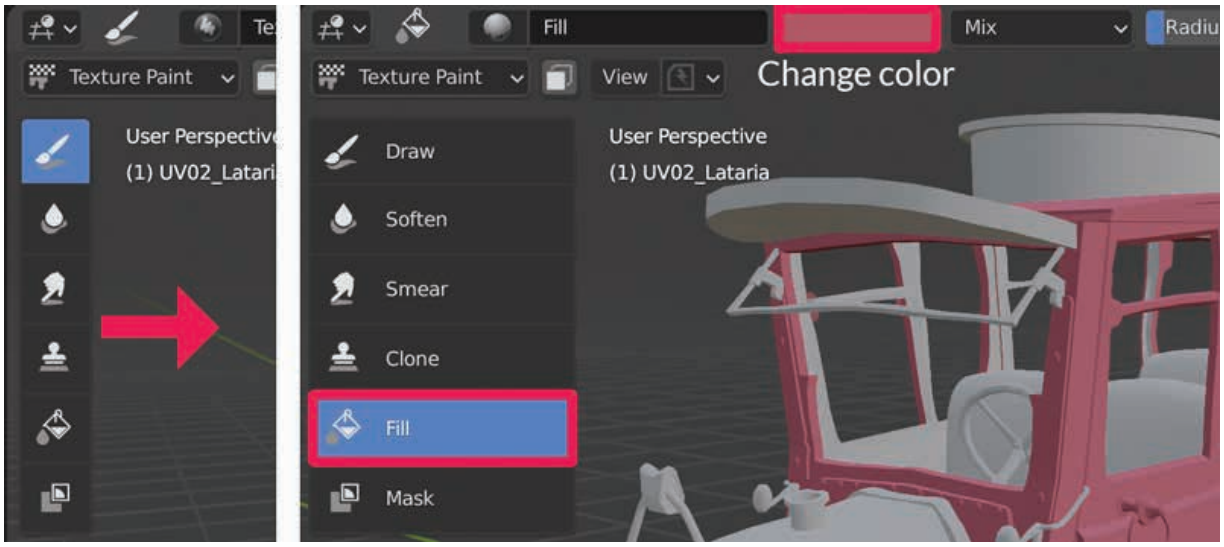
3D Model "French Car" by Arthur Sarah Gonçalves on Sketchfab

To be able to texture paint in the 3D viewport, your model needs to be UV unwrapped and have a material. Add a **texture** to your material. You can quickly add the necessary nodes by **pressing Ctrl+T** if you have the **Node Wrangler** add-on enabled. Then add a new texture by **clicking on New** on the texture node and go to **Texture Paint mode**.

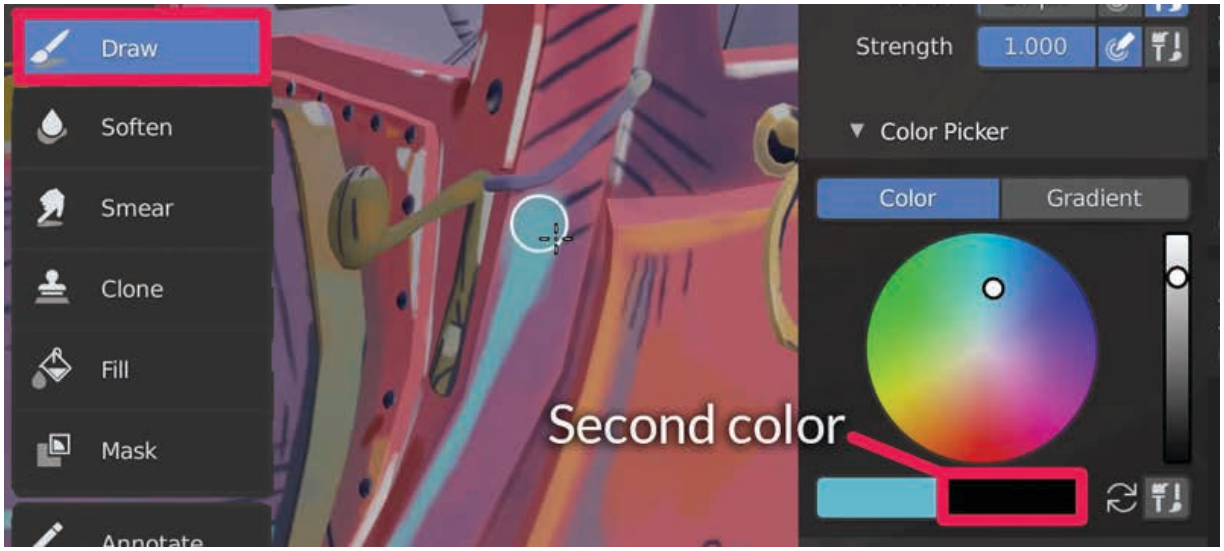




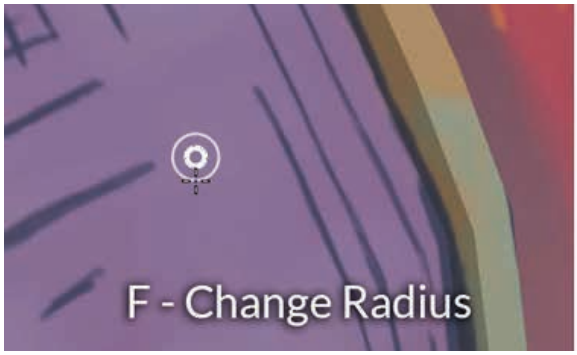
Press T if you don't see the **Tool Panel**. You can drag the Tool panel to the right to see the Tool names. In my experience, you need to grab it slightly to the right of the panel to be able to drag it, instead of at the edge of the panel. With the **Fill Brush** we can change the overall color. The paint color can be changed by clicking on the color field.

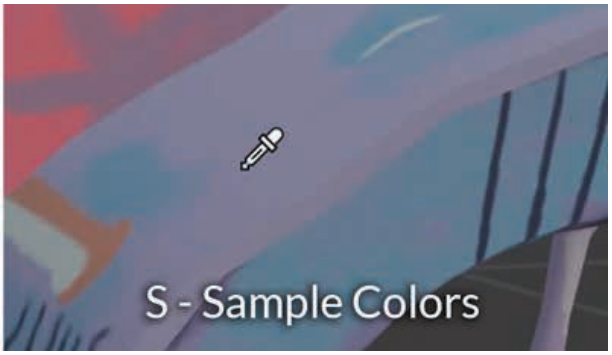


And with the **Draw brush** we can paint strokes. You can also set a second color, and **hold Ctrl** to use that.

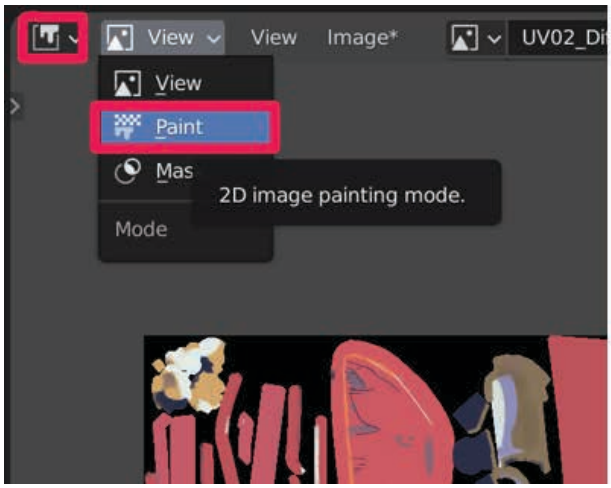


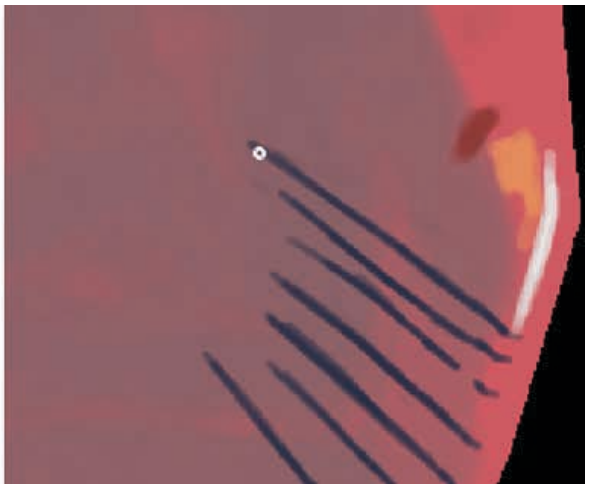
To change the **Radius** of the brush, **press F**. Sample colors from anywhere in the viewport by **pressing S**.





You can also paint directly on the texture in the **Image Editor window** by choosing **Paint Mode**. *Please note that this does not work in the UV Editor window (like it used to, before Blender 2.8). You really have to switch to the Image Editor window, even though they look very similar.*



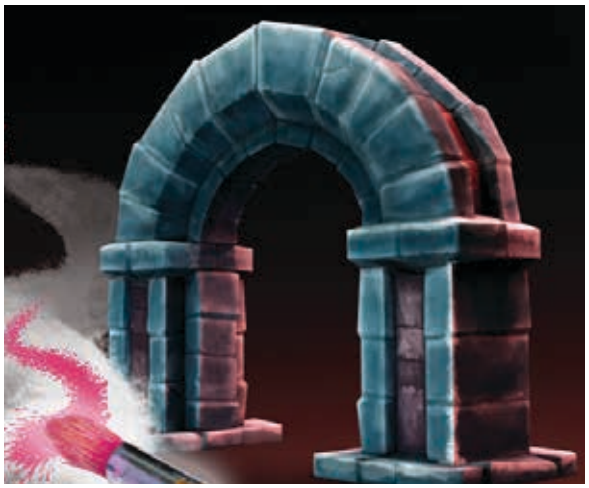


Texture Painting with Custom Brushes

Now that you know the very basics of texture painting in Blender, you'll probably be eager to have more brush options than just the one default brush.

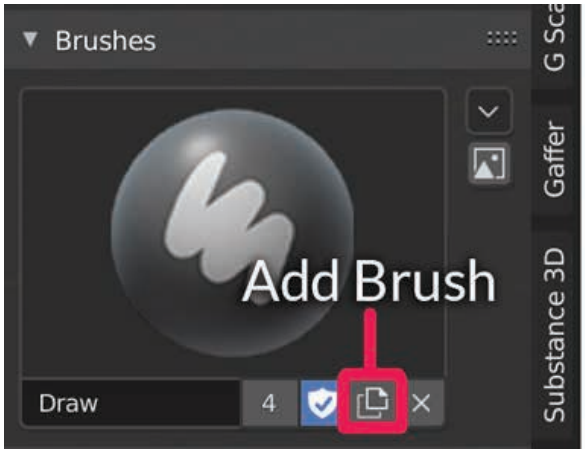
Luckily, changing brush settings to create your own brushes is easy.

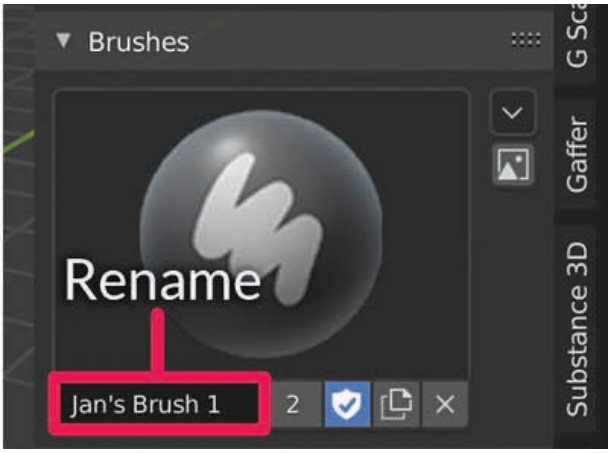
Even more interesting, is that you can use textures to create more organic looking brushes, that rotate randomly.



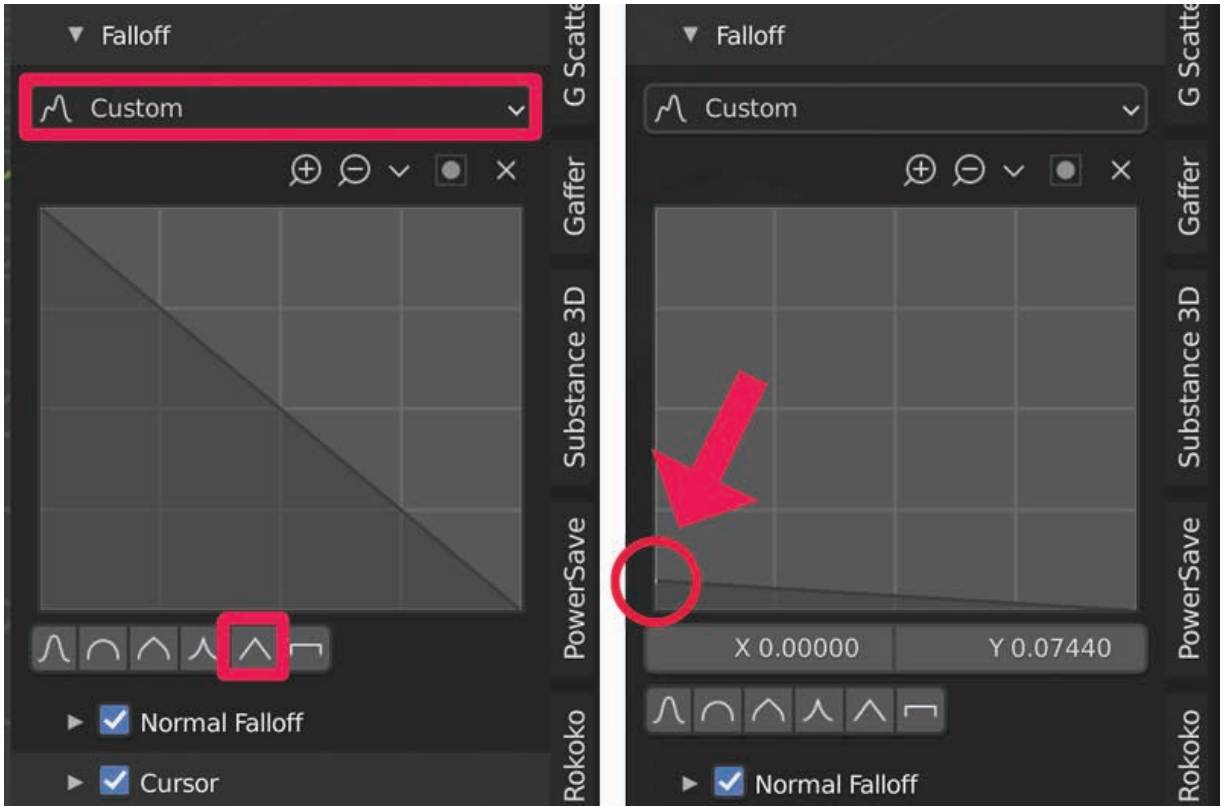
3D model "Low Poly Hand Painted Dungeon Arch" by BitGem on Sketchfab

To create a custom brush, click on the the **Add Brush** icon and change the name.

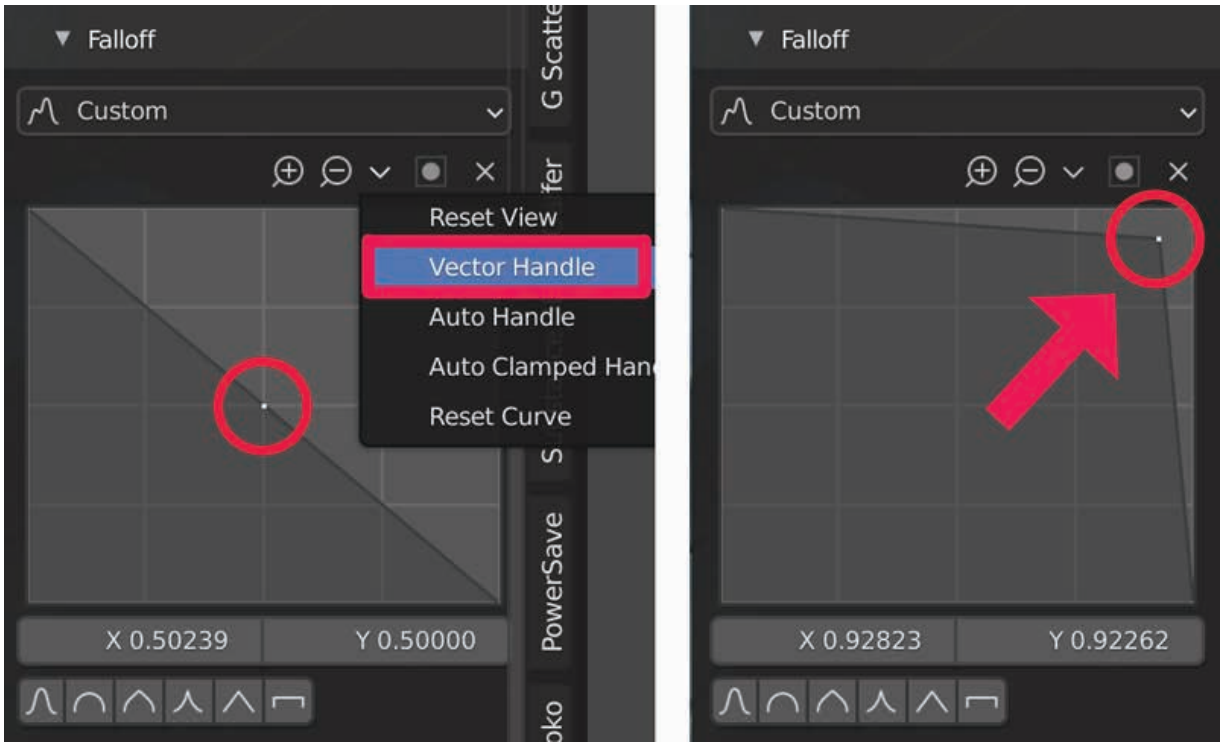




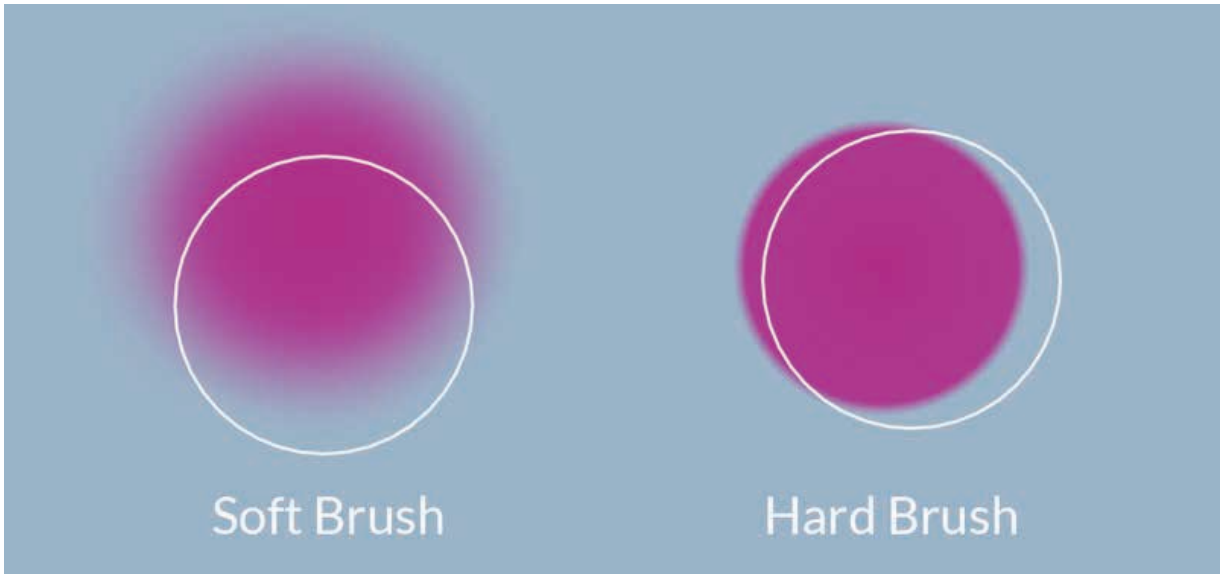
Scroll down in the option panel and under **Falloff**, choose **Custom** and click on the Linear icon (it looks like a pyramid). Drag the leftmost point down to get a very soft brush.



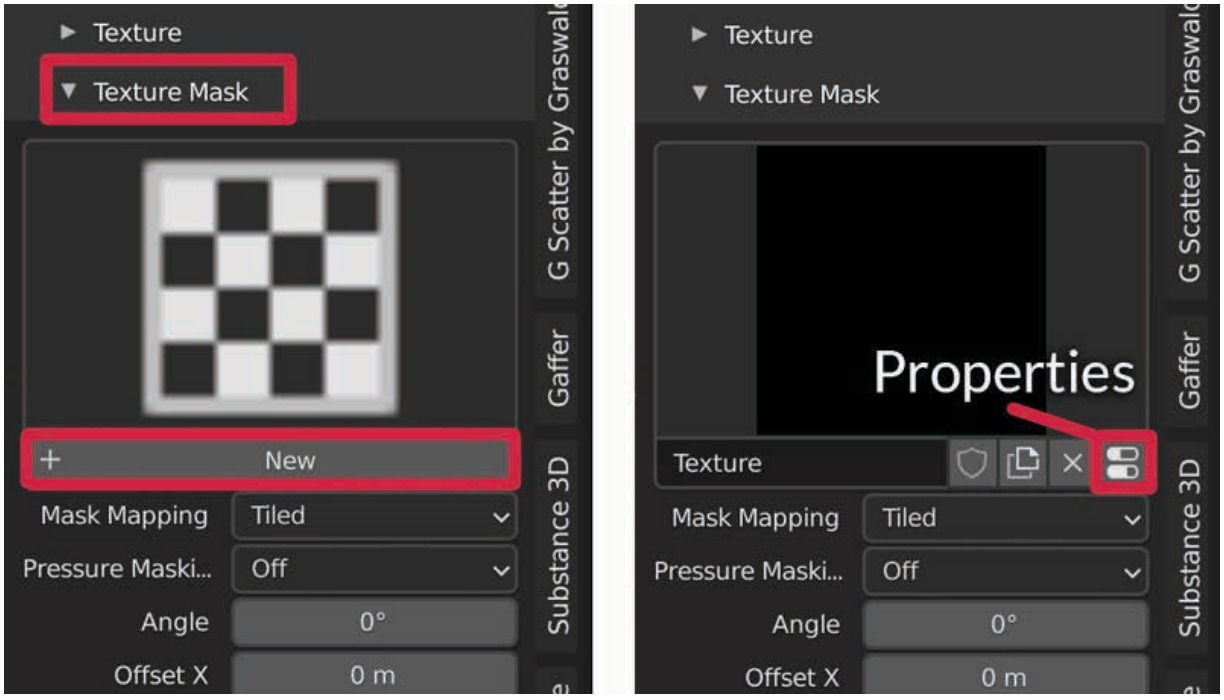
To make a very sharp brush, add a point by clicking on the curve. Then click on the option drop-down and select **Vector Handle**. Now you can drag the point up to get a sharper brush.



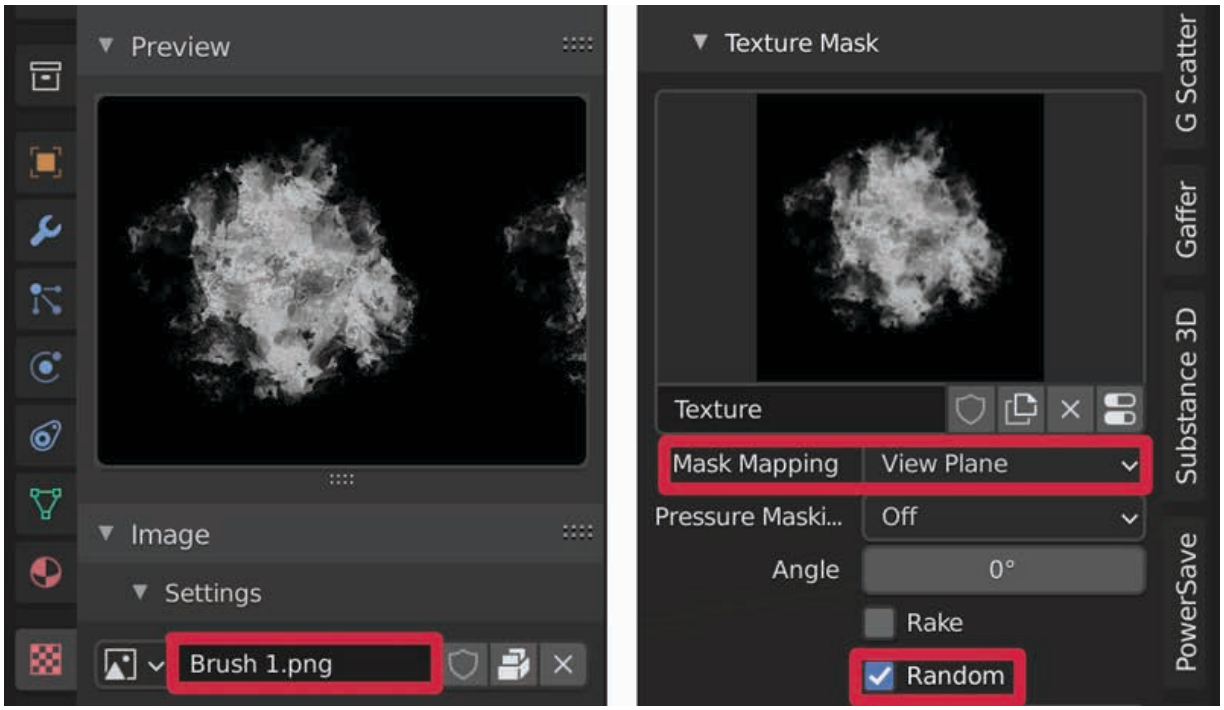
To compare, here's the soft brush (left) and the sharp brush (right):



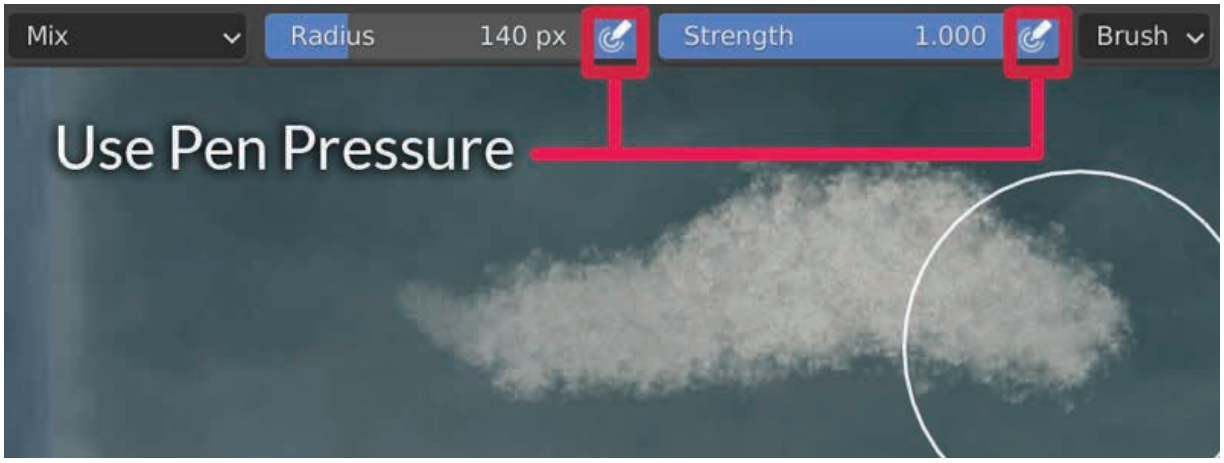
For a textured brush, scroll further down and under **Texture Mask** click on **New**. Click on the Properties icon to go to the Texture properties tab.



Select a black and white brush texture. I've created this one in Krita (an open-source and free 2D painting app). It's simply a black-and-white image saved as a PNG file. Set **Mask Mapping** to **View Plane** and check **Random** so it rotates randomly.



Set **Falloff** to **Constant** and enable **Size Pressure** and **Strength Pressure** if you want to use pen pressure.



Now you get a brush that has a more organic and random feel to it. This works well for these stone textures, for example.

Texture Painting with Gradients

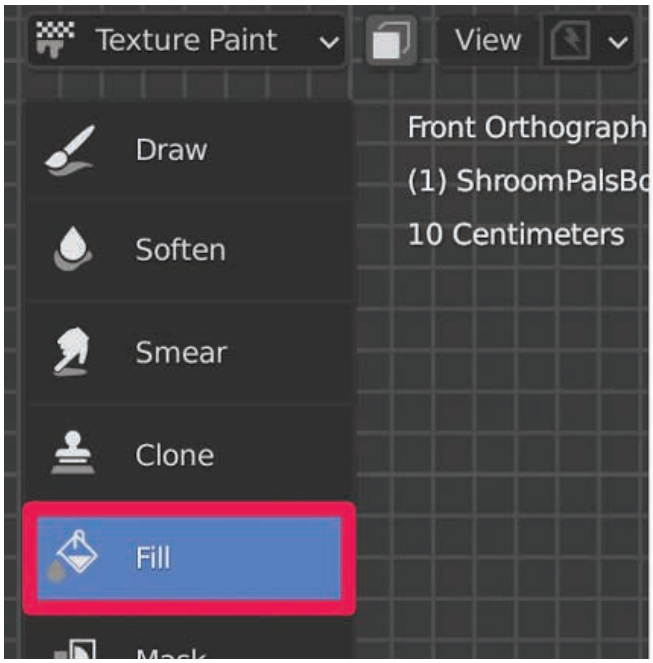
Gradients are an easy way to create a beautiful overall color in your object, that you can then paint additional details on.

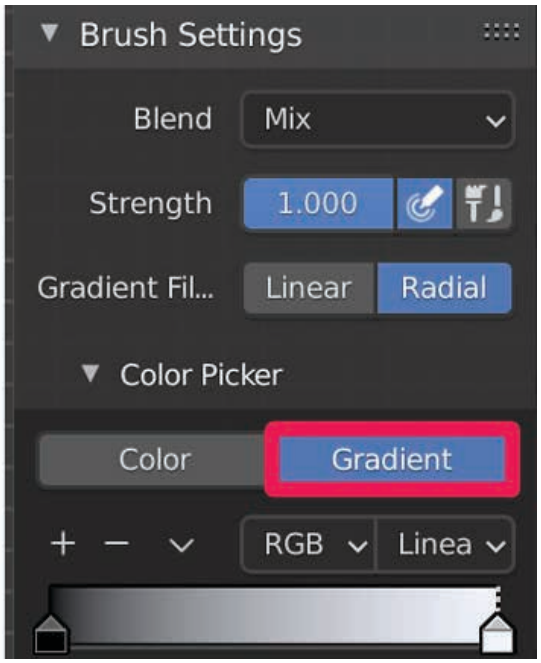
But it can be a bit challenging to find how to use the gradient function if you don't know where to look. Here, we look at two ways you can add gradients to your model.

In Texture Paint mode select the Fill tool. Open the Tool tab in the Option Panel by **pressing N**.

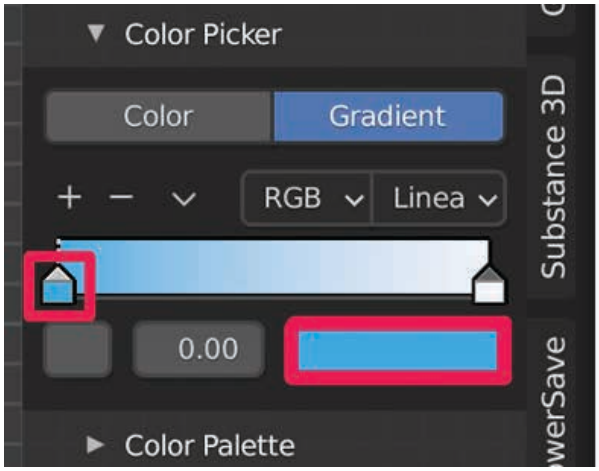
Under Brush Settings set Color Picker to **Gradient.**

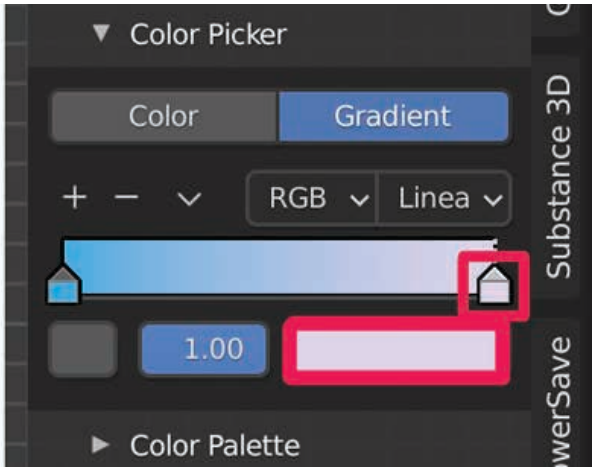
3D Model "Mushy Buddy" by Gorgon Season on Sketchfab.



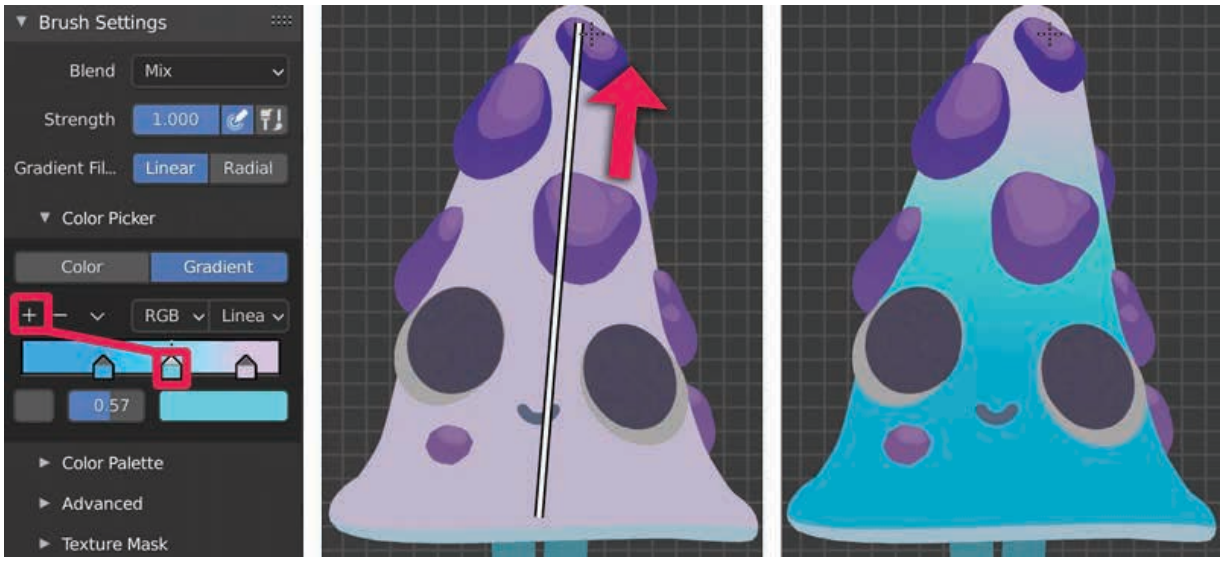


Click on one of the stops. Change the color in the field below it. Then click on the other stop and change its color.

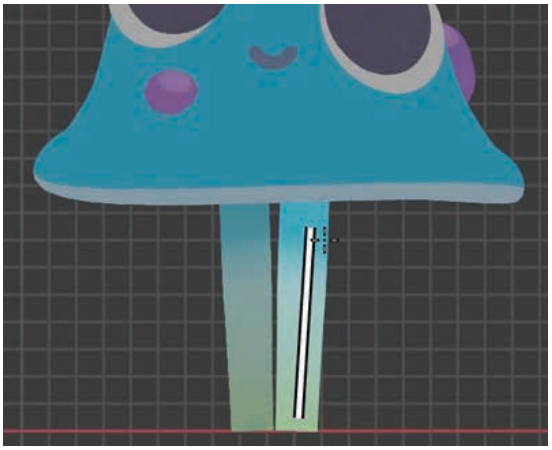




You can add additional stops by clicking on the **+ icon** if you want to add more colors to the gradient. You can also move the stops to different locations on the gradient, to influence how much of one color is visible. **Hold the left mouse button** and drag a stroke across the object to fill the object with the gradient.

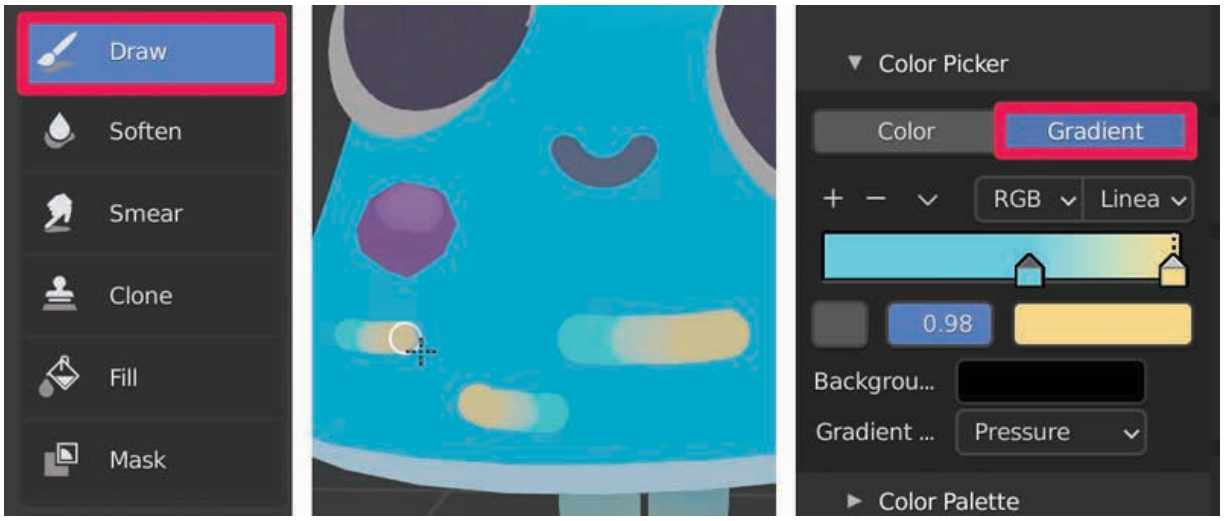


Pro Tip: To Fill a different object with a gradient, you have to switch to Object Mode, select it and then go back to Texture Paint mode.

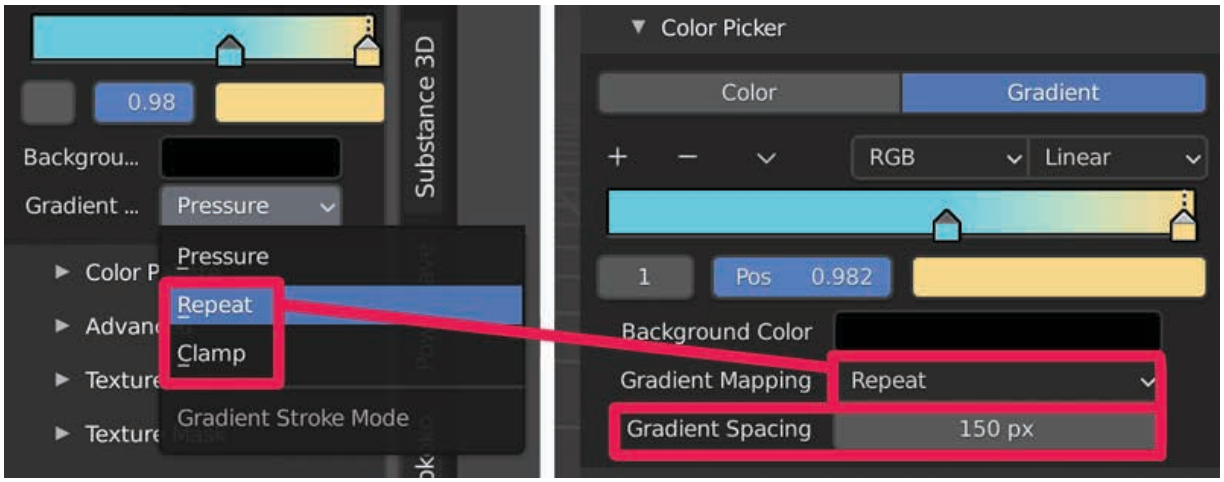


However, from Blender 3.0 on, you can simply hover over the object you want to paint, and **press Alt+Q** to select it without having to exit Texture Paint mode.

You can also paint gradient strokes with the **Draw Tool**. Select the Draw Tool and under Brush Settings click on Gradient. For this to work you need to use **Pen Pressure** so you'll need a graphic tablet pen. Then drag a stroke and use the pressure to control the gradient. It takes a bit of practice.



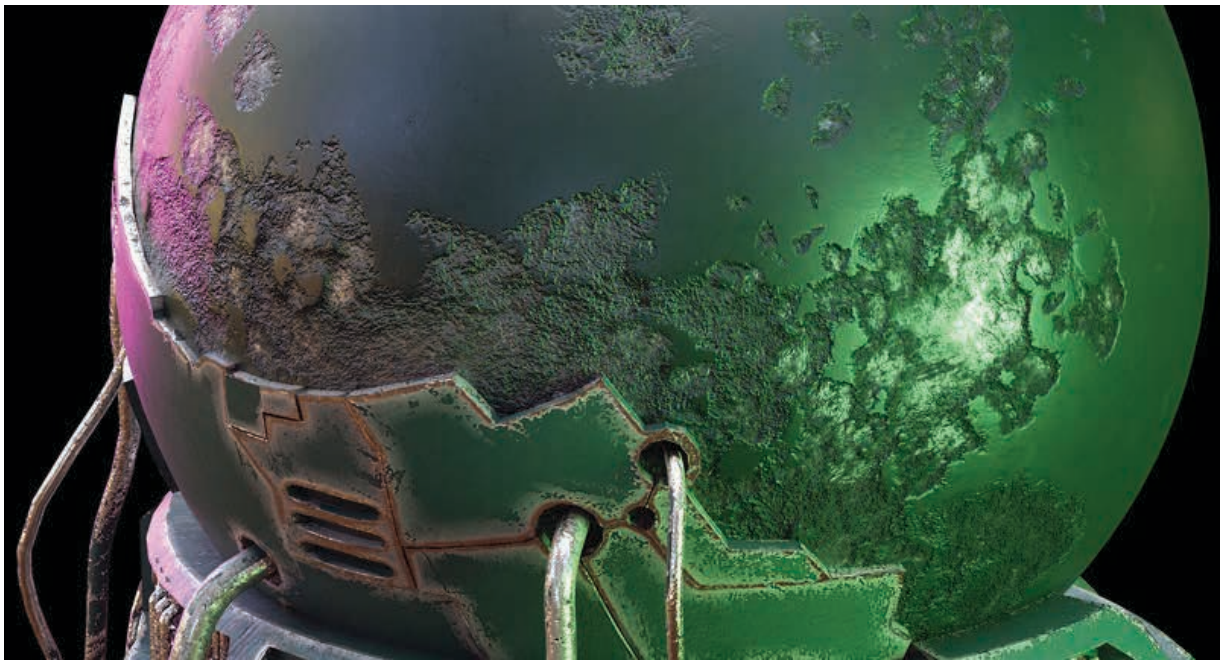
If you don't have a graphic tablet pen and want to use the mouse instead, set it to **Clamp** or **Repeat** and experiment with the **Gradient Spacing** value.



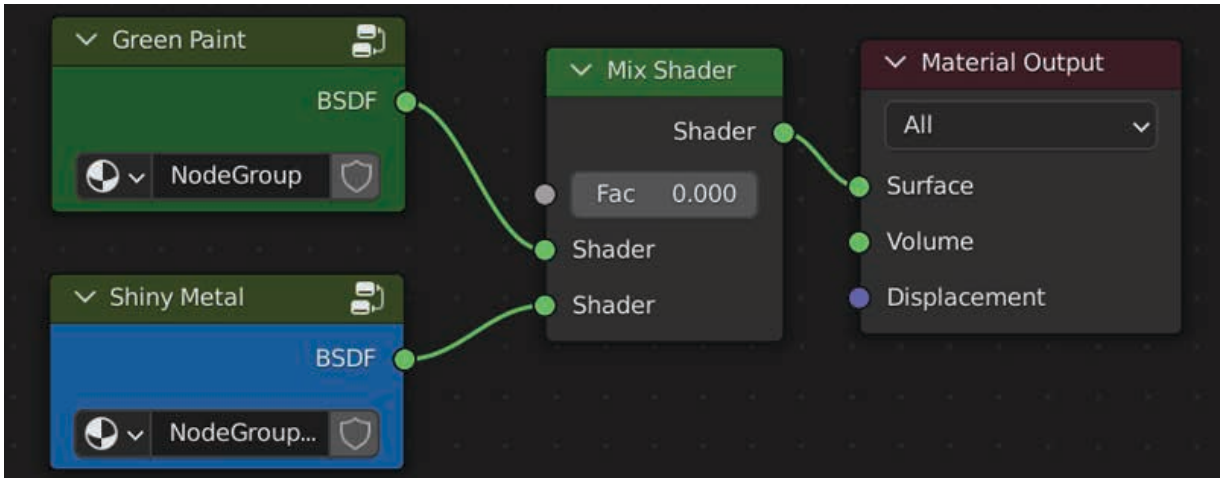
To paint only a part of the mesh, enable **Paint Mask**. Then **hold Shift** and click on the Faces you want to select, to paint only on those.

Adding Bump Height to Texture Masks

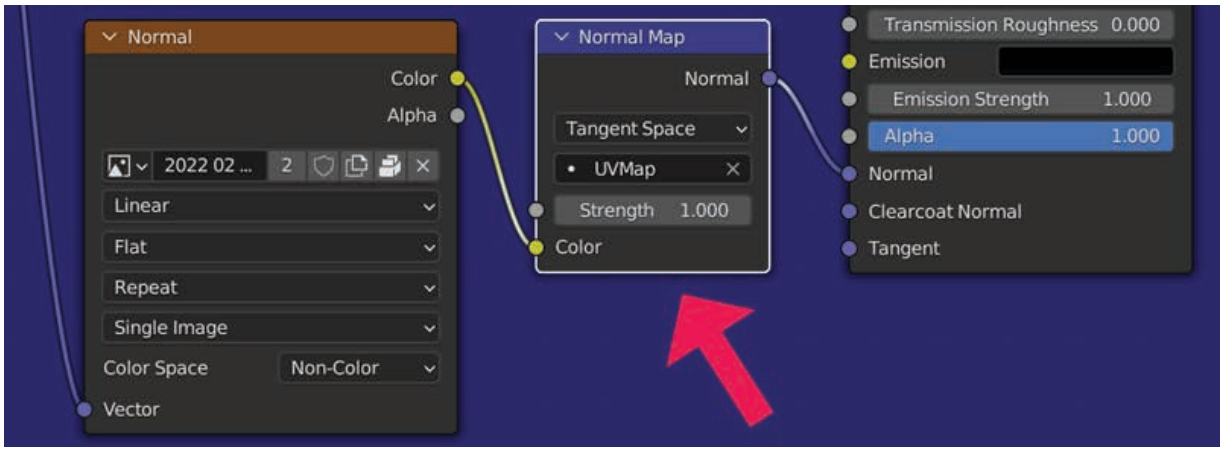
When mixing two materials using a black and white image texture as a mask, adding a height difference can create very interesting effects. For example, the look of a destroyed layer of paint, revealing metal underneath.



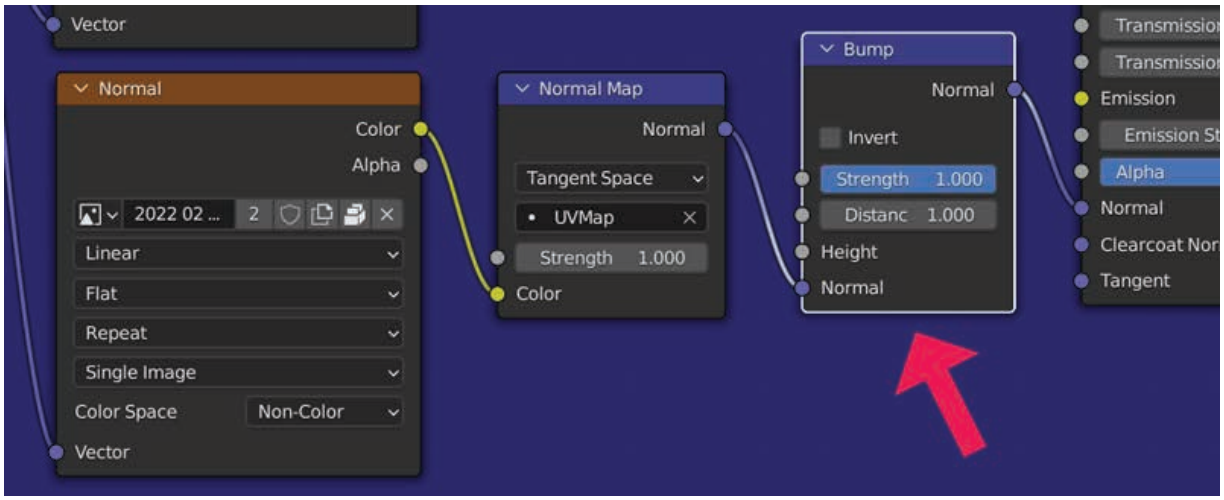
For this, we have two materials (made of PBR texture maps) and a **Mix Shader node** to mix between the two. For now, the Mix Shader node doesn't have an input yet. I've put the materials in Node Groups to make it easier to work with. You can add nodes to a group by selecting them and **pressing Ctrl+G**. More details about Node Groups in this video.



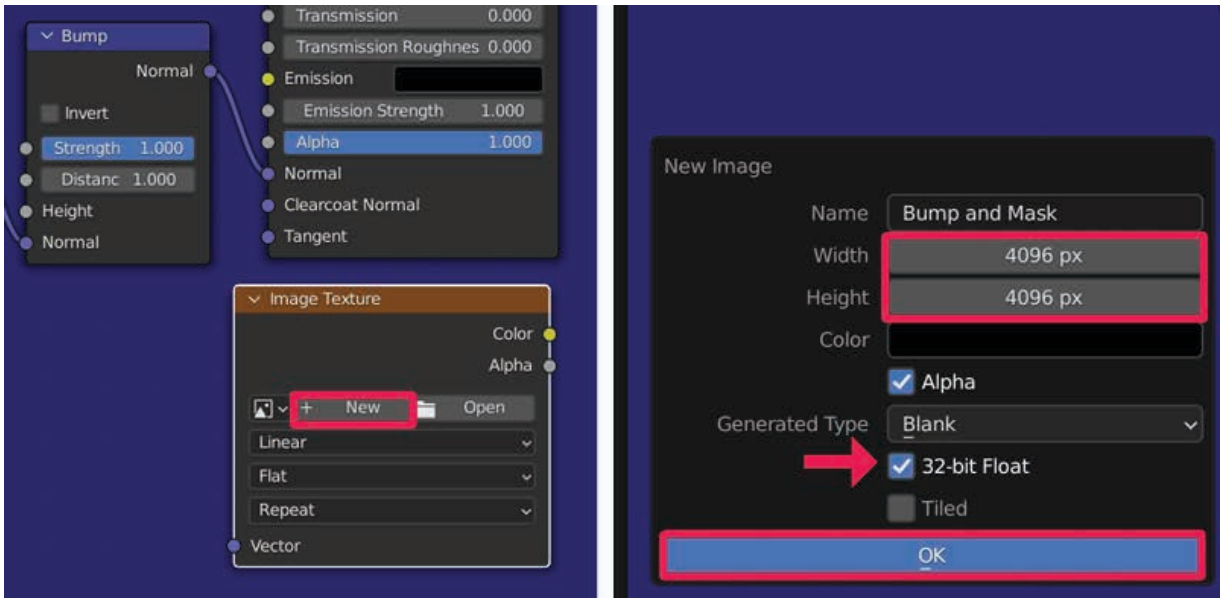
Let's take a closer look at the green paint material. Select its node group and **press TAB** to see what's inside the group. Surprise surprise, It's full of nodes! One of which is a **Normal map**.



Add a **Bump node**. Drag it between the Normal map and the Principled BSDF, and it will connect automatically.

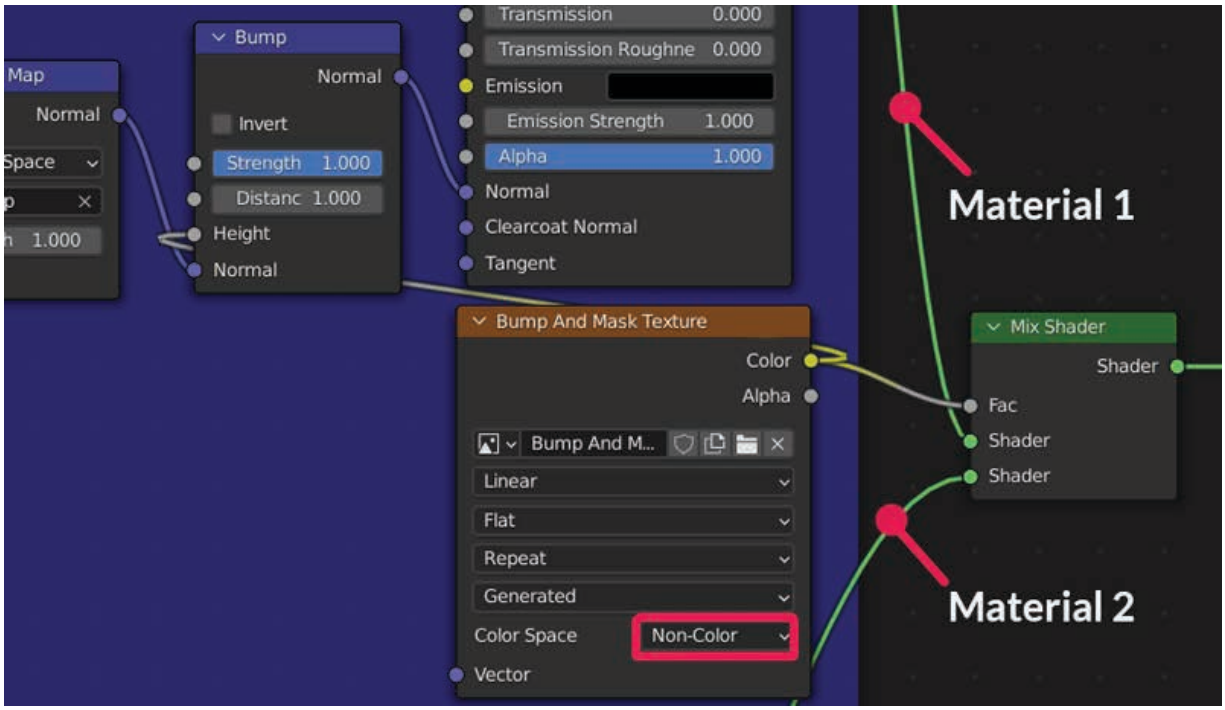


The Bump node needs **a black and white image** (a bump map) as input to know where to add or subtract height. In this case, we'll make a **new texture** that will double as both the Bump map and the Mask that mixes the blue paint material and the exposed metal material. **Add an Image Texture Node**. Click on **New**. Create a 4K image (4096 x 4096) with **32-bit depth**.

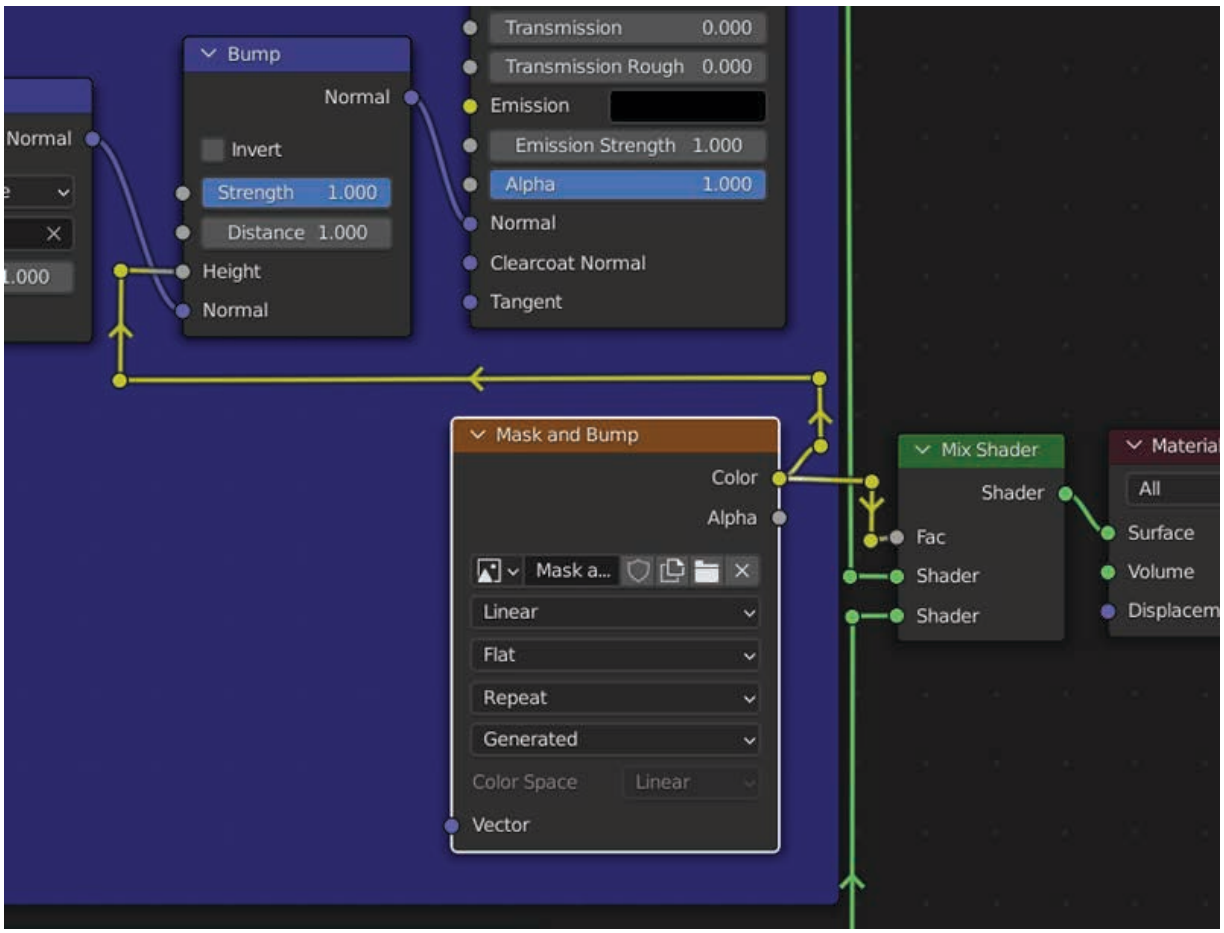


Below you can see why you really want to use a 32-bit map. On the left you can see the stepping artefacts caused by a lower bit depth. On the right, a 32-bit mask. Because the bigger bit depth has more distinct values of grey, you don't notice so much stepping.

Name it something like "Bump and Mask Texture" and click OK, then set it to **Non-Color**. Now, connect its Color output to both the Mix Shader **Fac** input and the Bump node **Height** input.



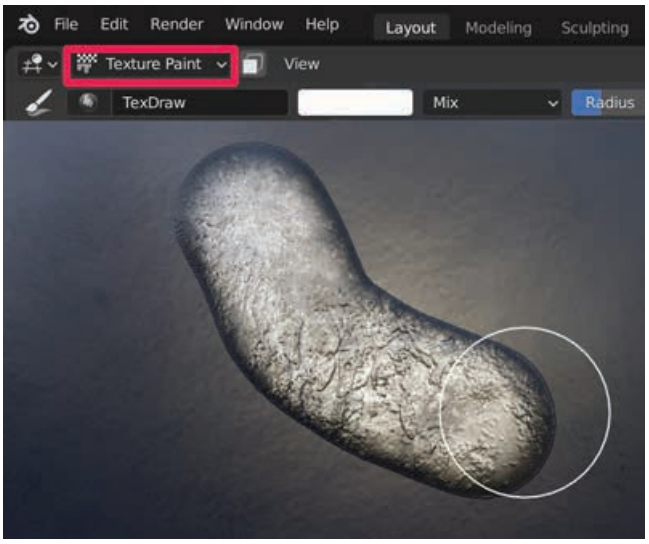
To make it easier to see the connections, you can use some Reroute Nodes. **Hold shift and drag across a node wire** to add such a point, then use **G to move it.** After using some Reroute Notes it looks better:



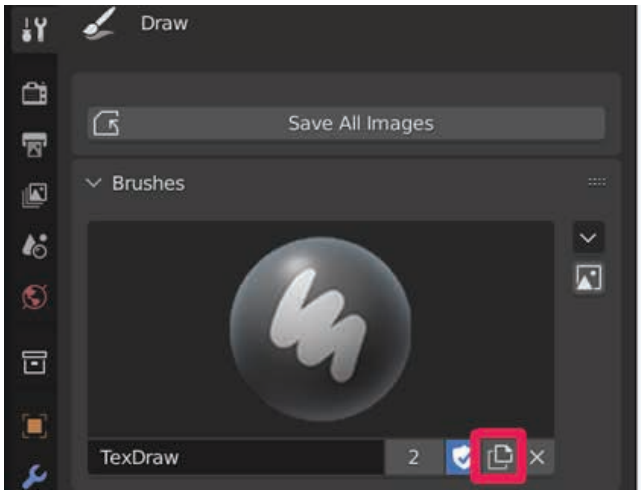
Select the **Bump and Mask Texture image node**, and go to **Texture Paint mode**.

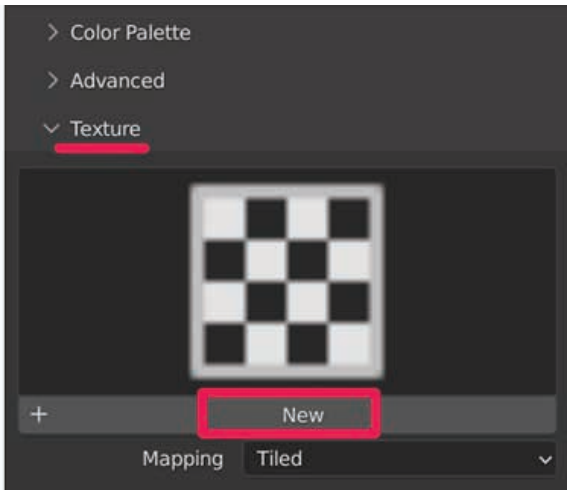
Now if we drag across the surface, we paint white on the mask. This reveals the other material "underneath".

However, with the default brush it doesn't look very good. Undo that stroke with **Ctrl+Z**.

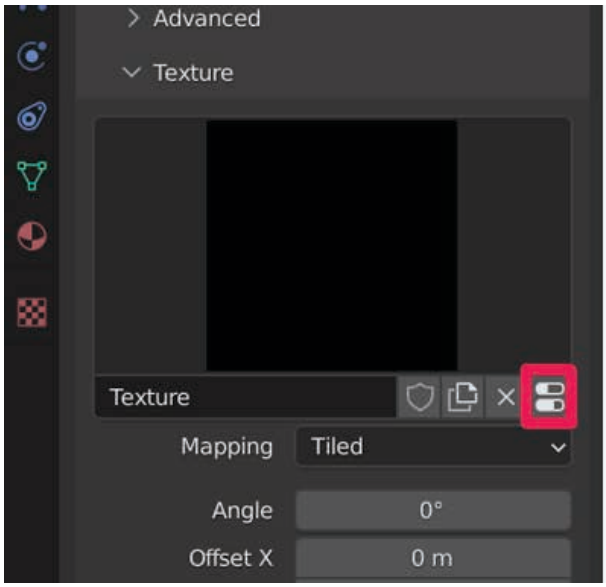


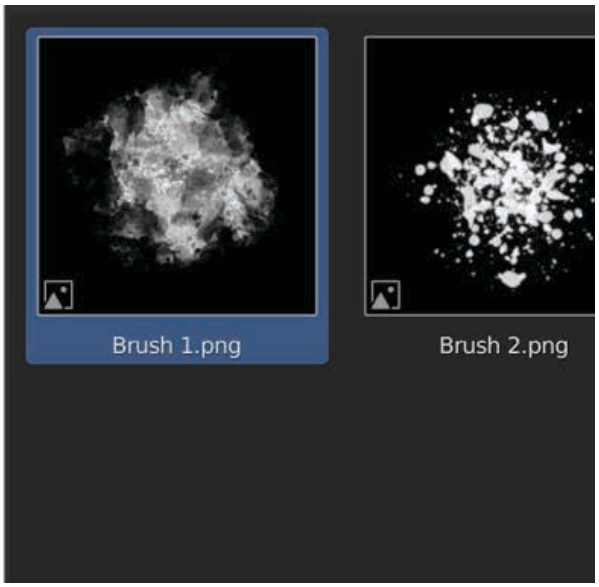
Click on the **Add Brush** icon to duplicate this brush, then give it a new name if you want. Scroll down to the Texture section of the Brush and **click on New**.



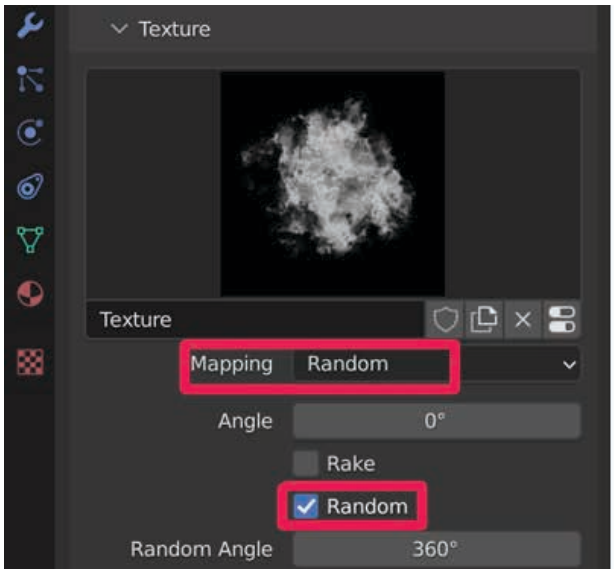


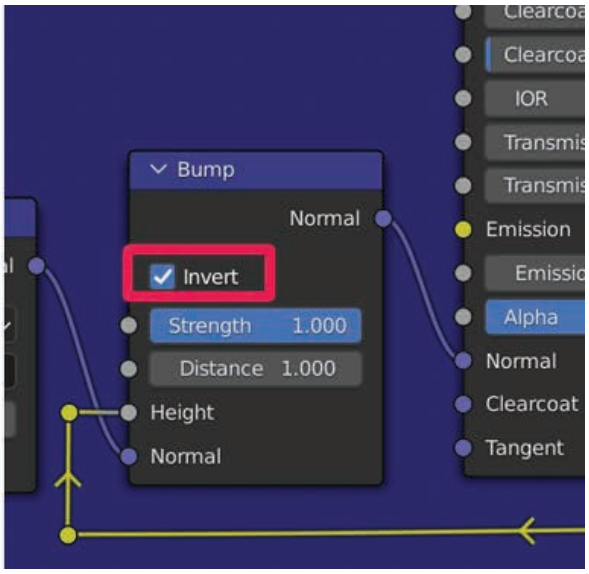
Then click on the texture properties icon. **Click on Open** and choose a black and white brush. I created this one in Krita, by painting a few strokes with one of the default Krita brushes.





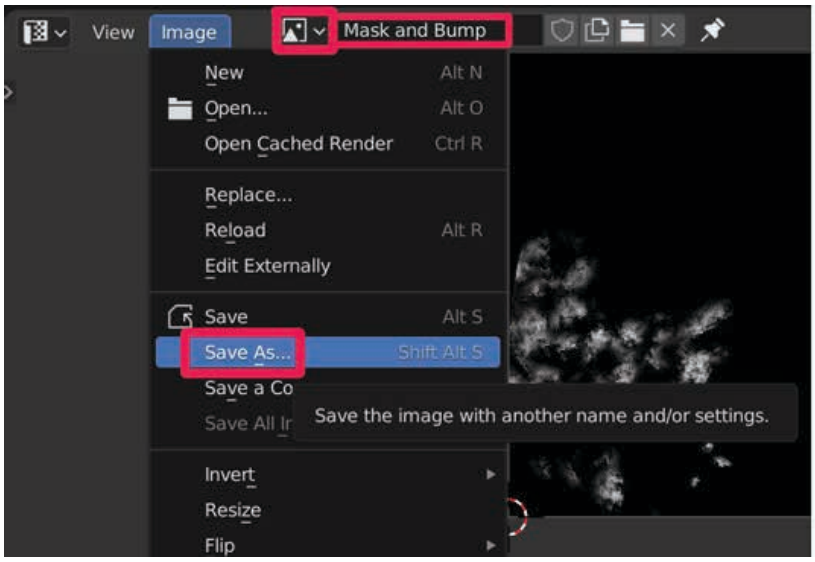
Set **Mapping** and **Angle** to **Random**. This give it a more organic look. Now when we drag across the surface, we get a more interesting masking effect. But, it looks like we still need to invert the effect. Check the **Invert** box on the Bump node. You can also fine-tune the **Strength** and **Distance** values to your liking.

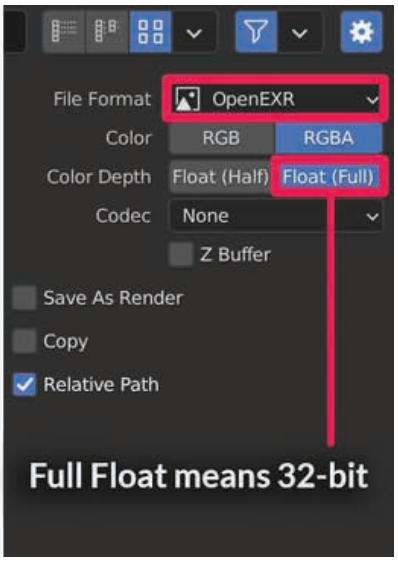




This is the effect we get now. If you want to erase the mask, simply **hold Ctrl** and drag over the place where you want to erase.

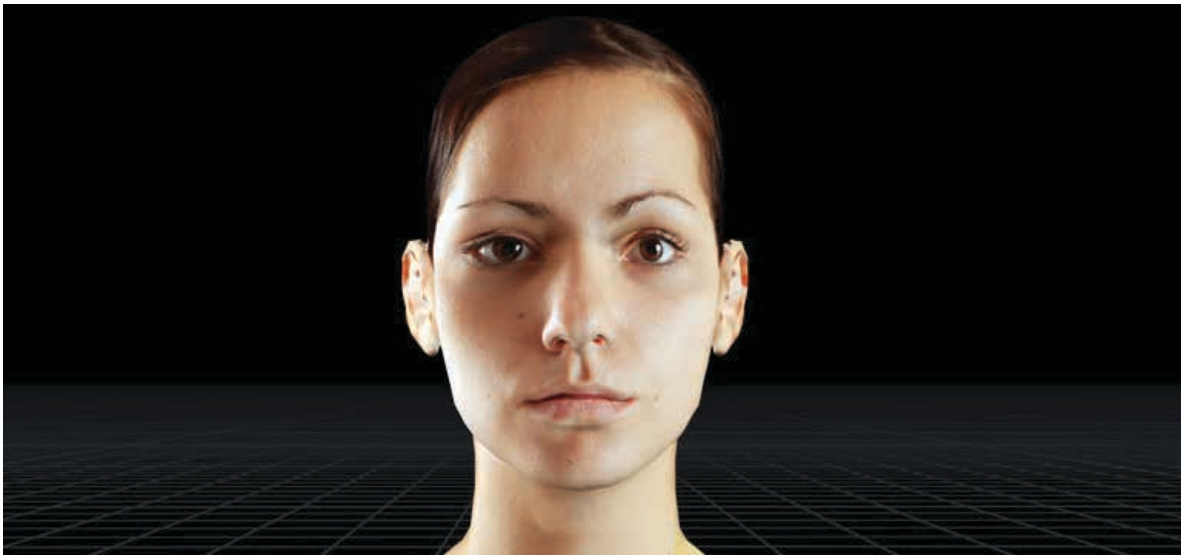
Finally, **don't forget to save your mask/bump image texture!** Otherwise, it will all be gone the next time you open Blender… To save, go to the UV Editor or Image Editor window, select the texture from the drop-down menu next to the name field and save it as a **Full Float EXR** image.



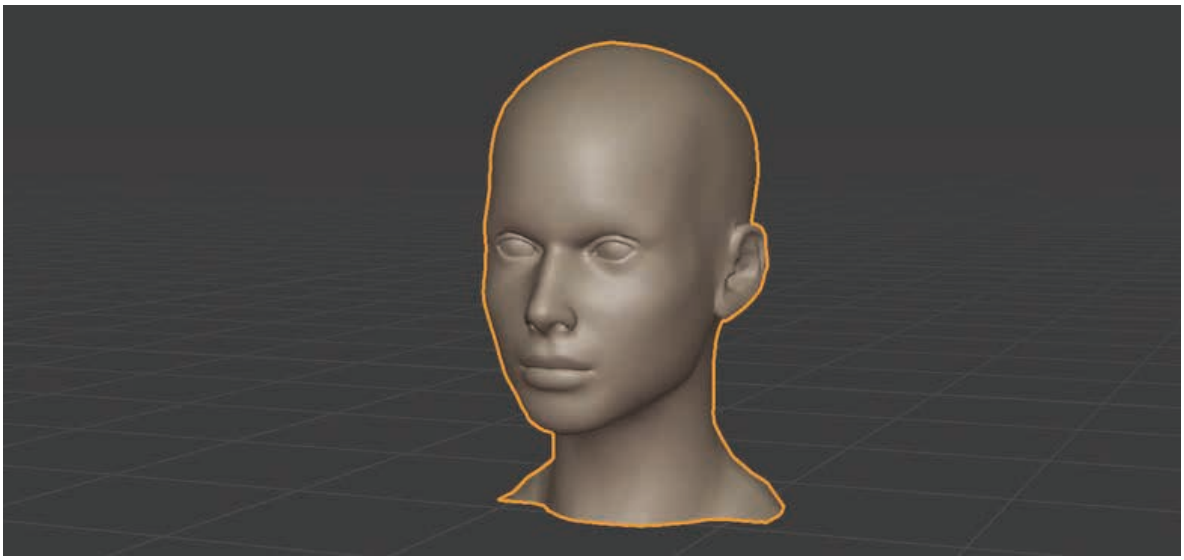


Texturing a head

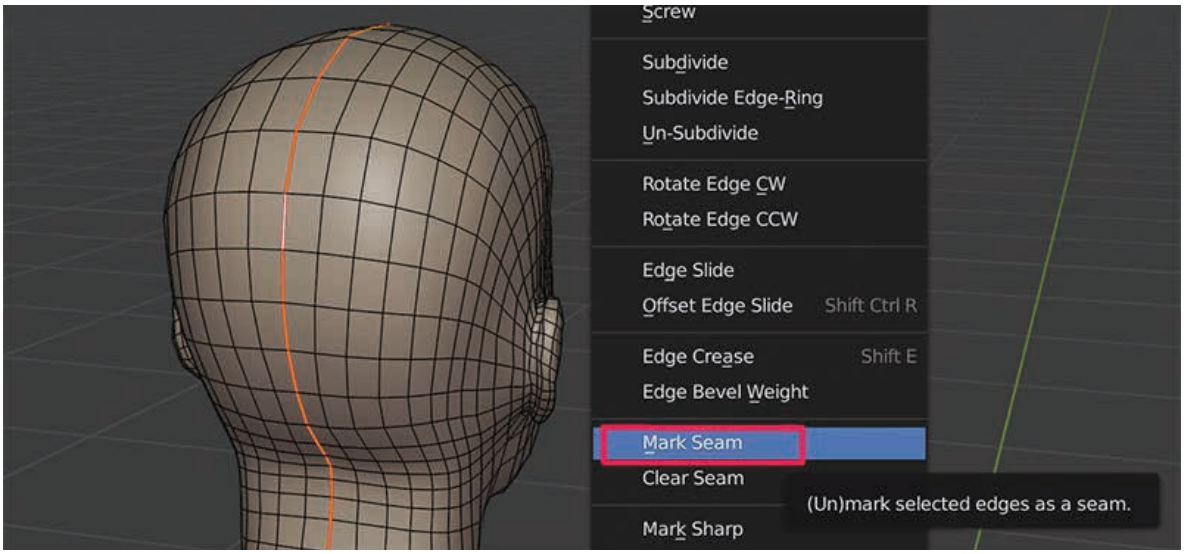
In this tutorial, we're going to look at how to paint a human head using photographs. You can watch part 1, part 2 and part 3 on Youtube as well.



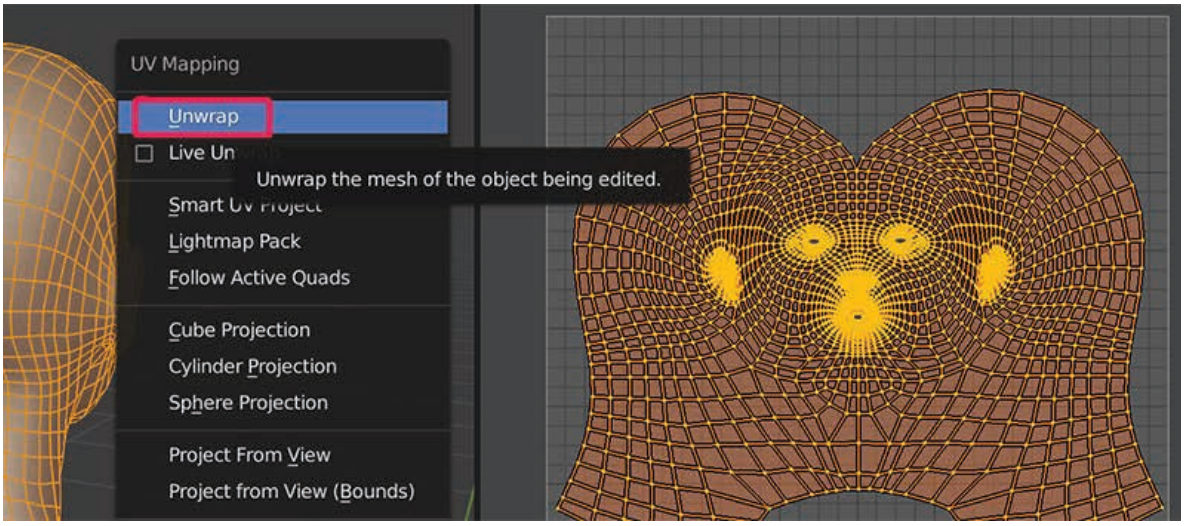
I created the head using the free MB Labs add-on for Blender to create humans. Then (for the purpose of this example) I removed all but the head and neck.



In Edit Mode you can add a seam by selecting edges from across the back of the head all the way to the hairline of the face (this is a good place to put the seam, since normally there will be hair here). **Press Ctrl+E** for the Edges menu and choose "**Mark as Seam**".



With the seam marked, select all in Edit Mode and **press U** to bring up the Unwrap menu. Choose "**Unwrap**" to create what is known as the "butterfly" unwrap of the face. It looks weird, but that's how it's supposed to look.



Give this UV map a clear name like "**UVMap Final**". You will be making more UV maps later, but those will be temporary.

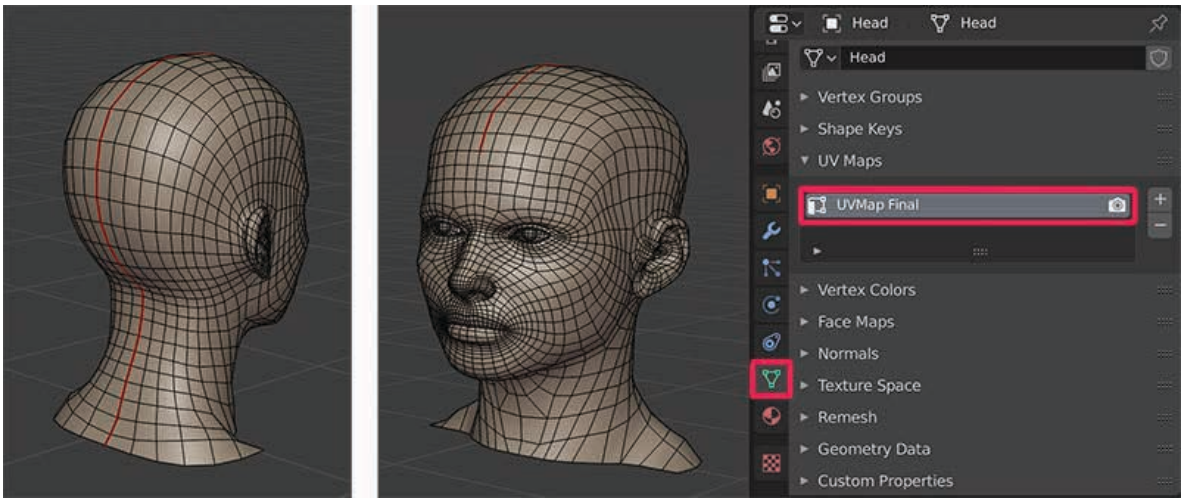
To check stretching generally you want to add some kind of checkered texture to your model first. To avoid stretching you can unwrap in different parts.

Make seams for the ears and neck and unwrap those separately. Select the other faces and go to Front orthographic view, then **press U** and choose **Sphere Projection**.

Go to UV > **Pack Islands** to arrange all the separate parts.

Honestly though, since we are going to use projection painting, the seams are not so important as long as there is no stretching.

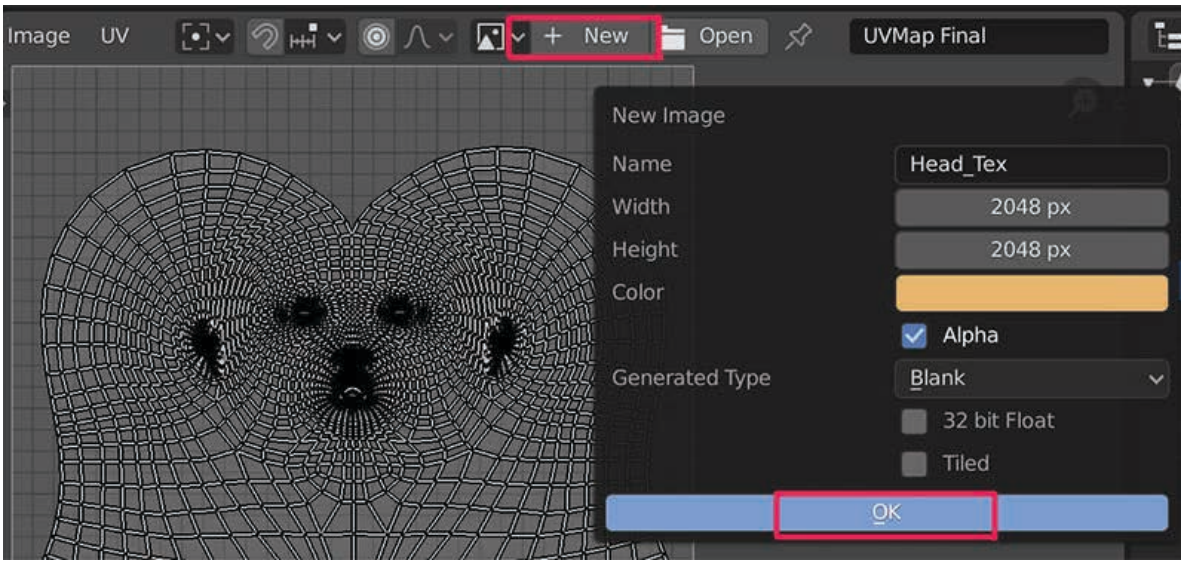
You could even get away with just doing a **Smart UV Project**.



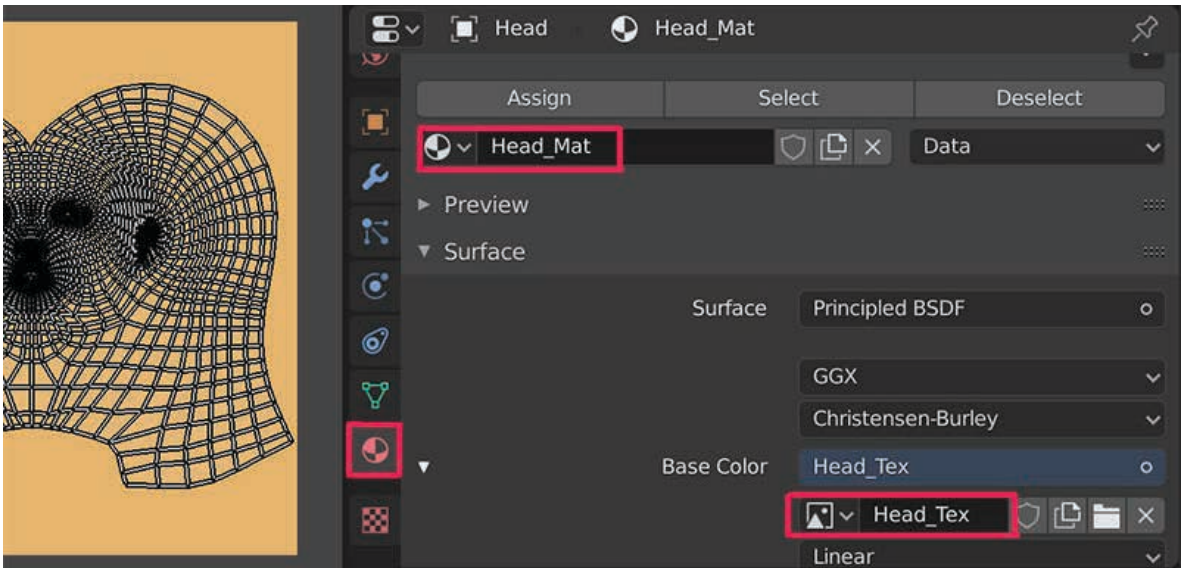
Now we are going to use photographic reference to create the texture. For this, I purchased some photographs from 3d.sk which is a website that sells photographic reference for 3d modeling, specifically for modeling humans. Of course, it's easy to create your own reference images using yourself or a friend as a model.

Now in the **UV Editor window**, click on **New** to create a new texture. Since the reference photos I have are high resolution, I will create a 2K texture.

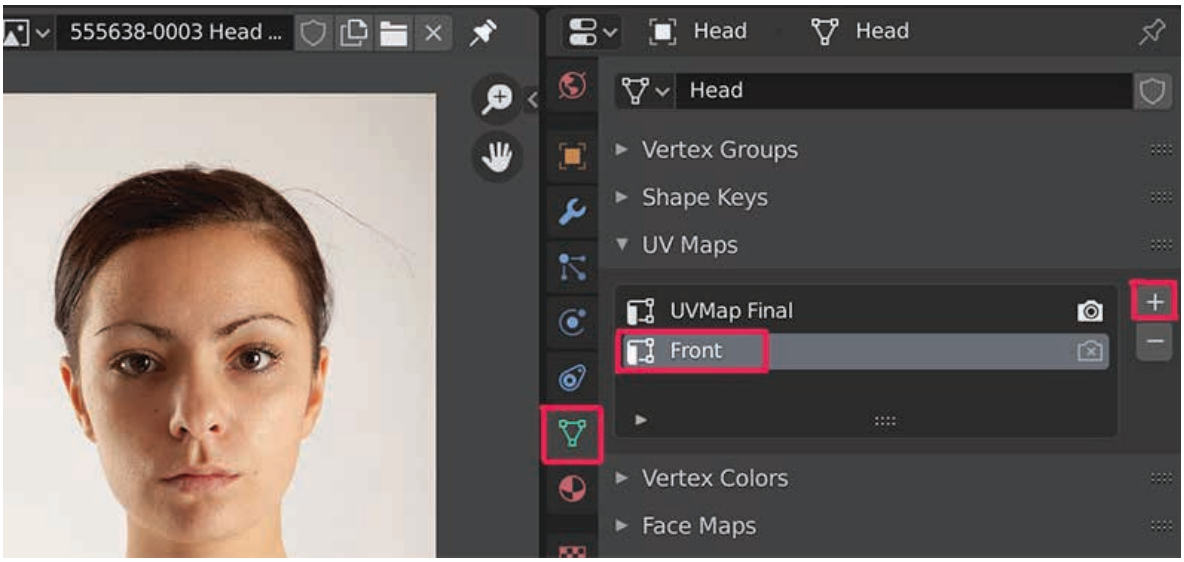
The color doesn't really matter since we'll be painting over it anyway, but I chose a skin tone to get started. Name the texture something clear like "**Head_Tex**".



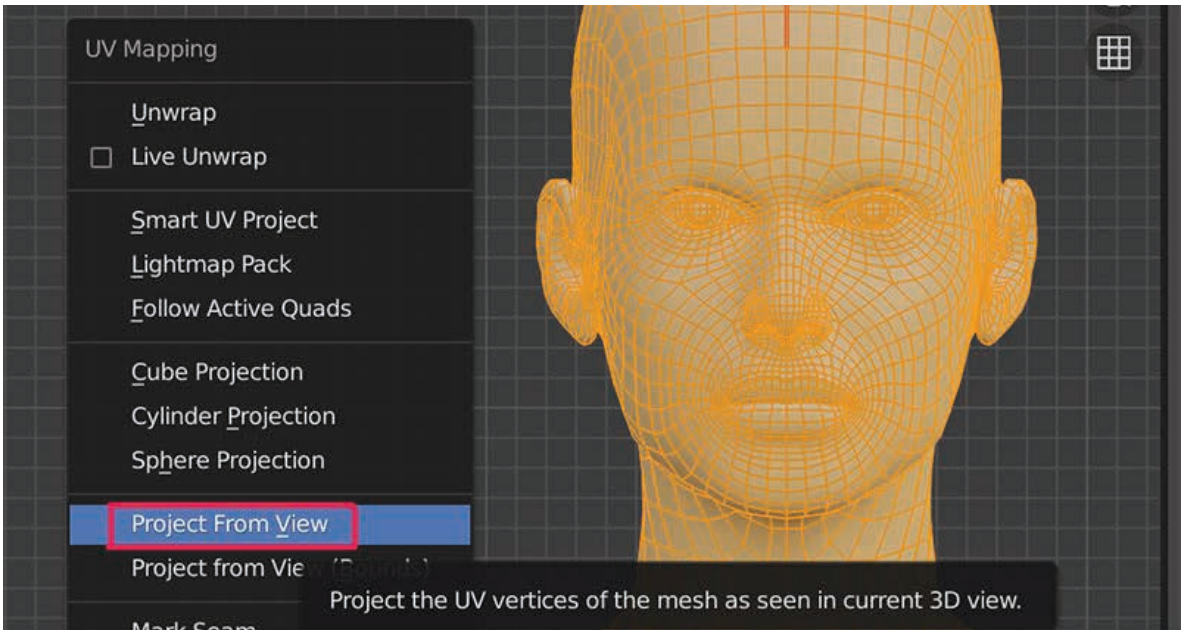
Give the head a new material and as the **Base color** use an **Image texture**, set it to our new "**Head_Tex**" texture.



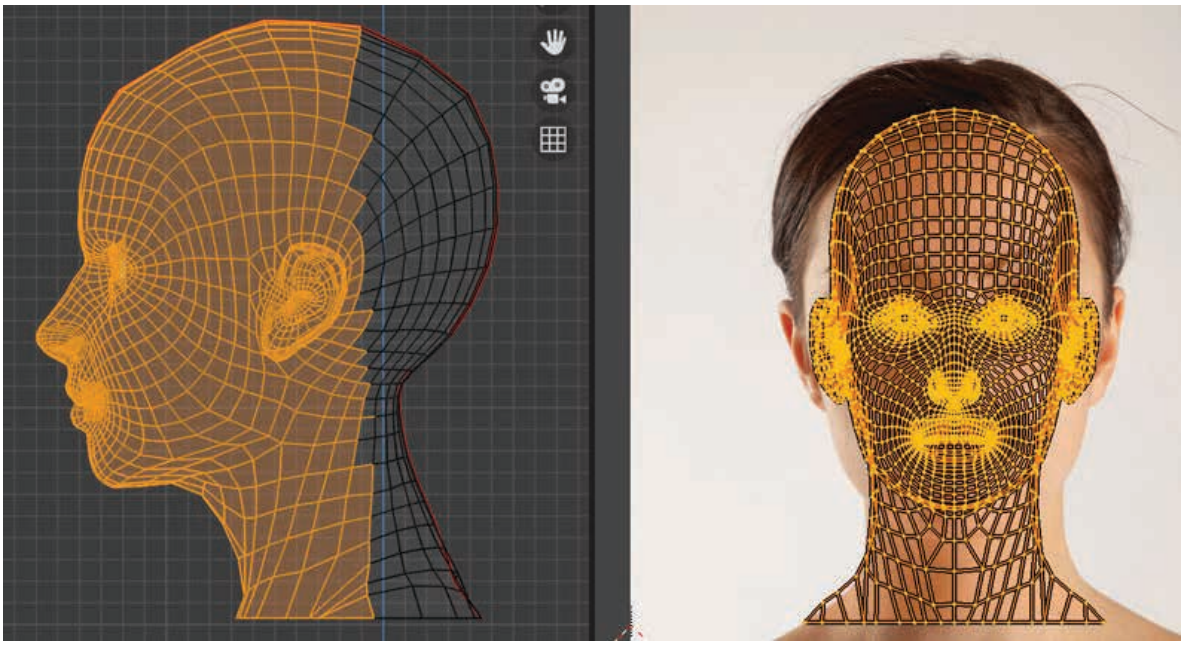
Set your viewport to Material preview by **pressing Z** (**and drag your mouse**) and choosing **Material Preview**. Now you just see the skin tone that we chose as the base color of the head texture. Load the Front image into the UV Image Editor and create a new UV Map by clicking on the **+ icon** next to the UV Maps, and name this new UV map "**Front**".



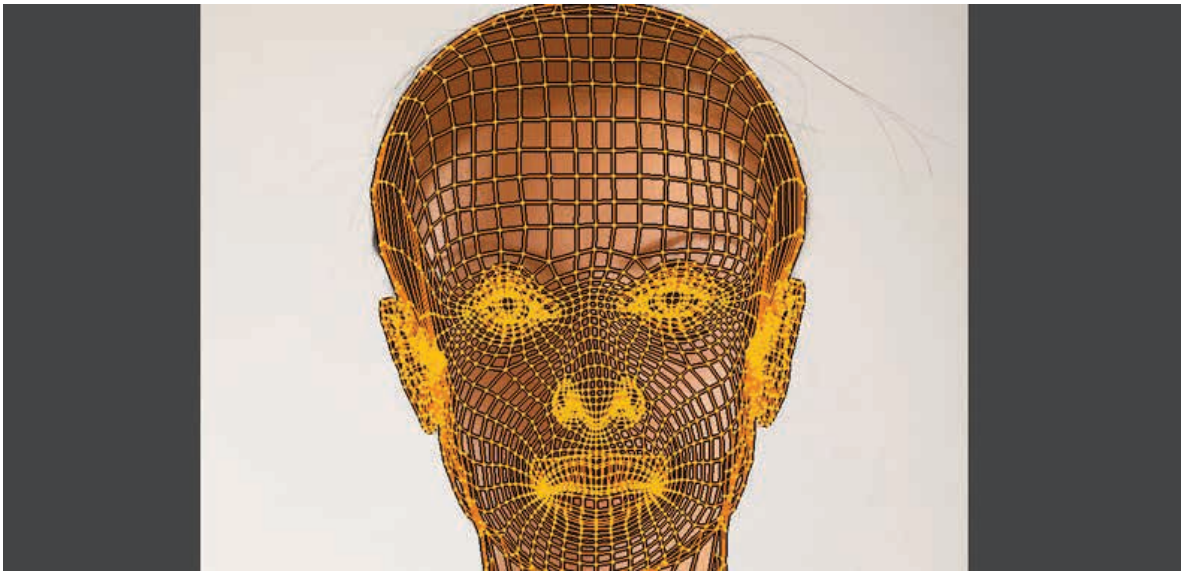
Switch to Front orthographic view in the Viewport by **pressing 1** so you see the head from the front. Then select all in Edit Mode and press U, but this time choose "**Project from View**".



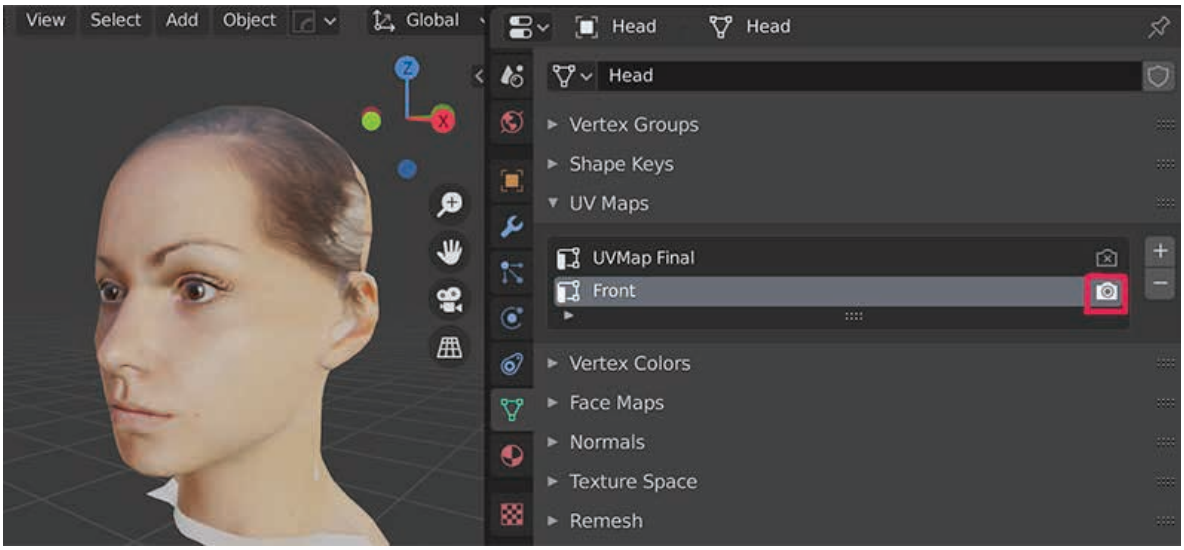
You'll see that most likely, the Unwrapping needs some work to match the photograph. In Edit Mode and X-ray view, select only the front faces so you only see those UV's.



Then, **move(G)**, **scale (S)** and **rotate (R)** the UV's to match as closely as possible the photograph. You can turn on Proportional Editing by **pressing O** and scrolling the mouse wheel to work more quickly.

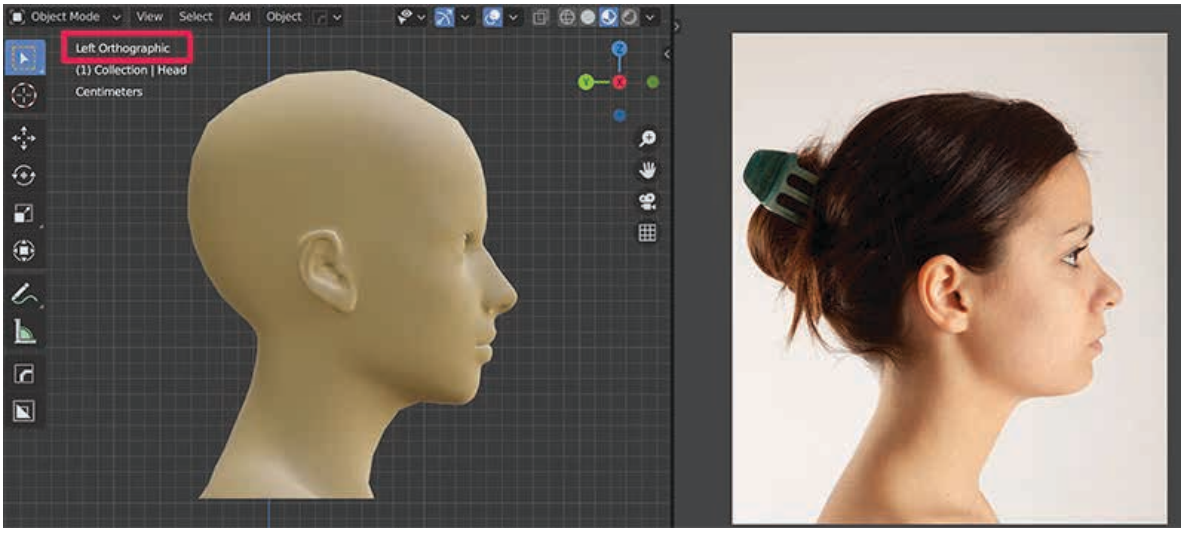


If you want to see what the result looks like so far, change the texture in the material to the Front photo and switch the UV Map to the Front UV Map by clicking on the camera icon next to it.

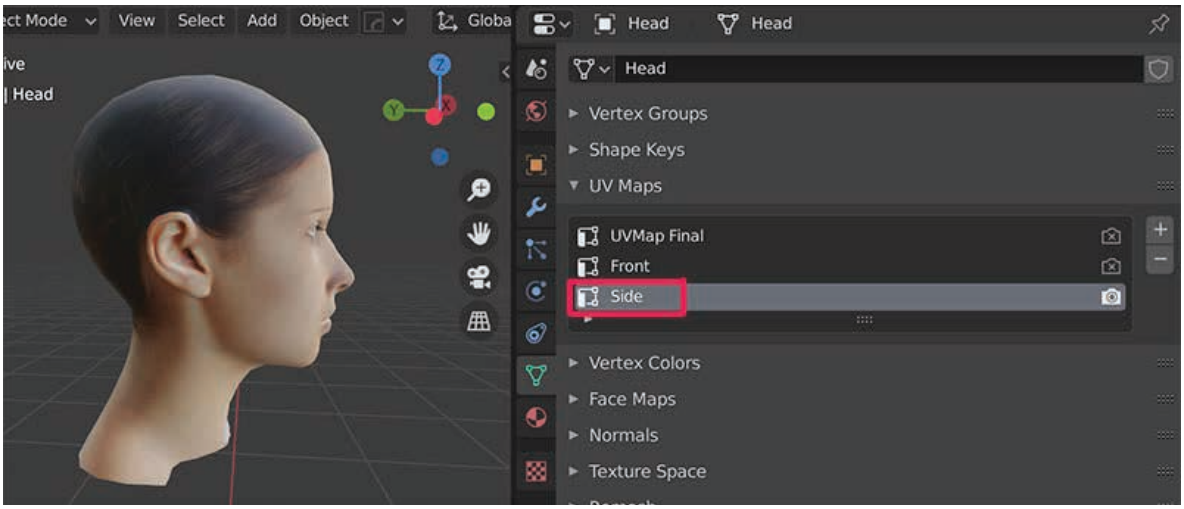


As you can see, even just one photo can already go a long way! However as soon as you look to closely at the side, you can see stretching and just a lack of texture information.

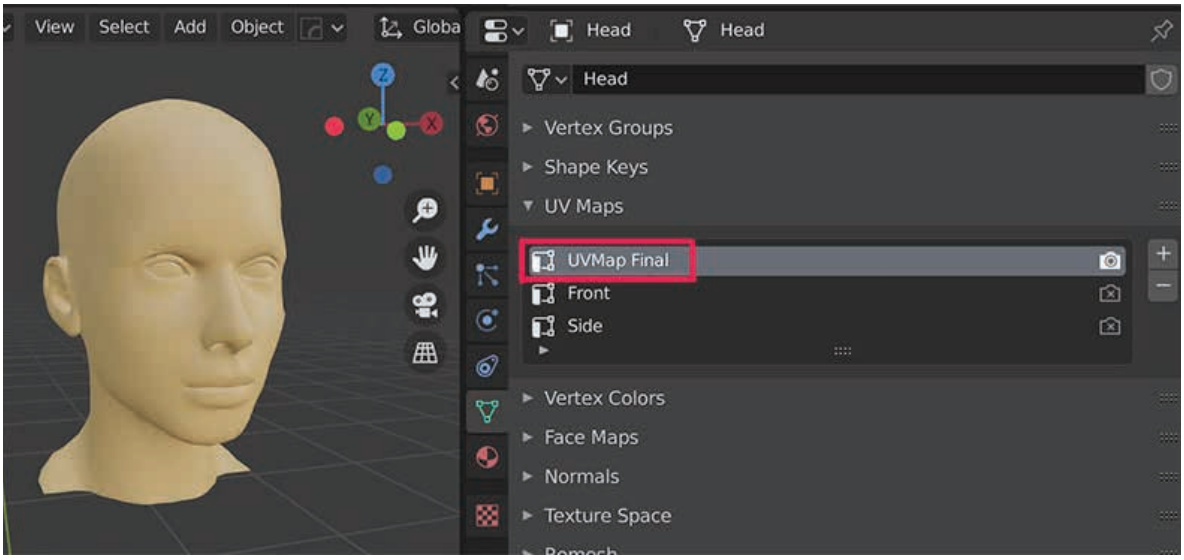
So we need to repeat these steps and create a **Side UV Map**. Load the Side photo in the UV Editor and project from view from the side. In this case we have a good picture from the Left side. To go to the Left orthographic viewport, **press Ctrl+Numpad 3**.



After making the Side UV Map fit the photograph, it should look good from the side. Don't worry if the front looks bad, as we already have that part in the other UV Map.



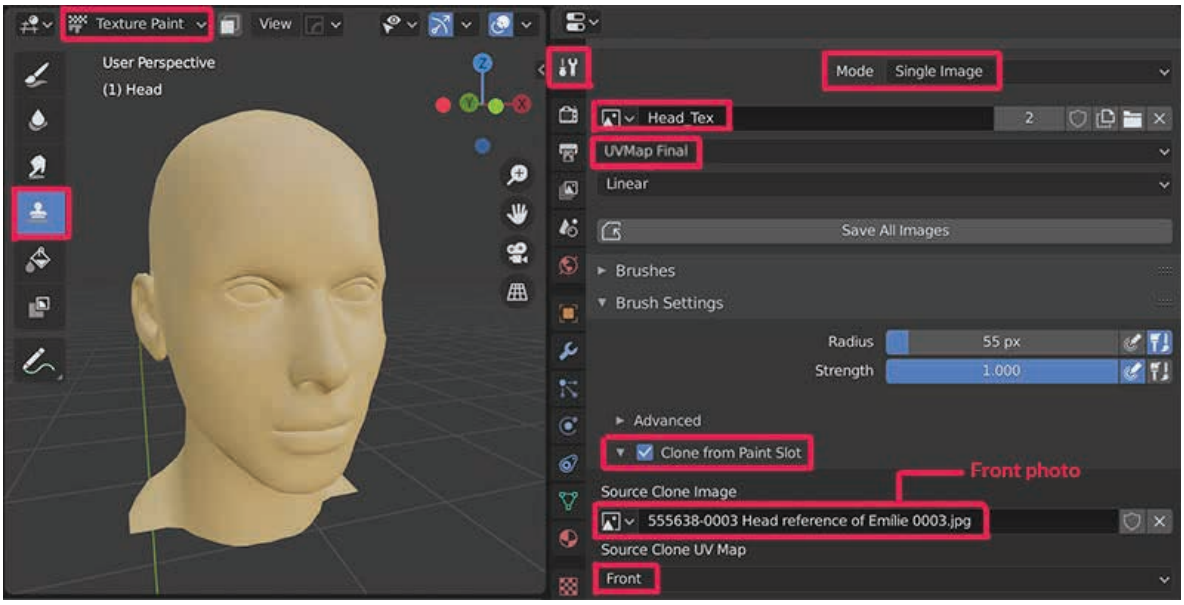
Now comes the really fun part, as we're going to combine these two projected UV Maps into one Final UV map. Make sure the blank head texture is selected as the texture in the material and that the **Final UV map** is selected in the UV Maps window.



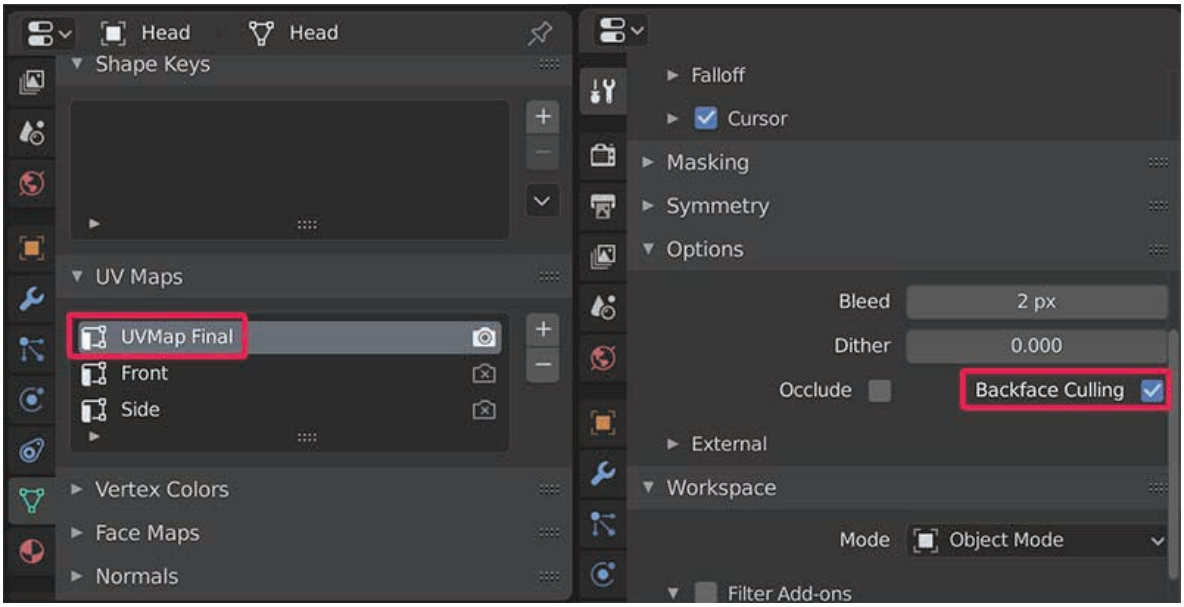
Now go to **Texture Paint mode**. To get started you will need to change a bunch of settings as you can see from the image below.

First, change **Mode** to **Single Image**. Then make sure your correct texture is selected (Head_Tex) which is the blank texture.

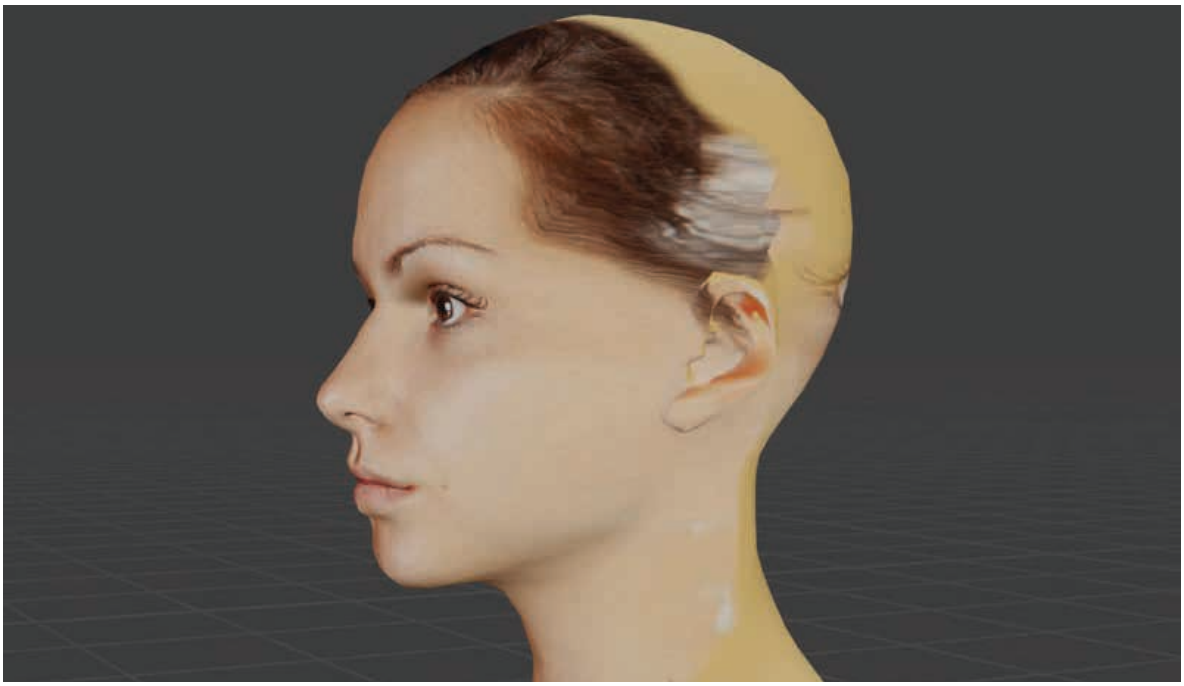
Now check "**Clone from Paint Slot**" and choose the front image, set **Source Clone UV Map** to **Front**. In other words, we will be cloning from the Front UV map and texture, to the Final UV Map.



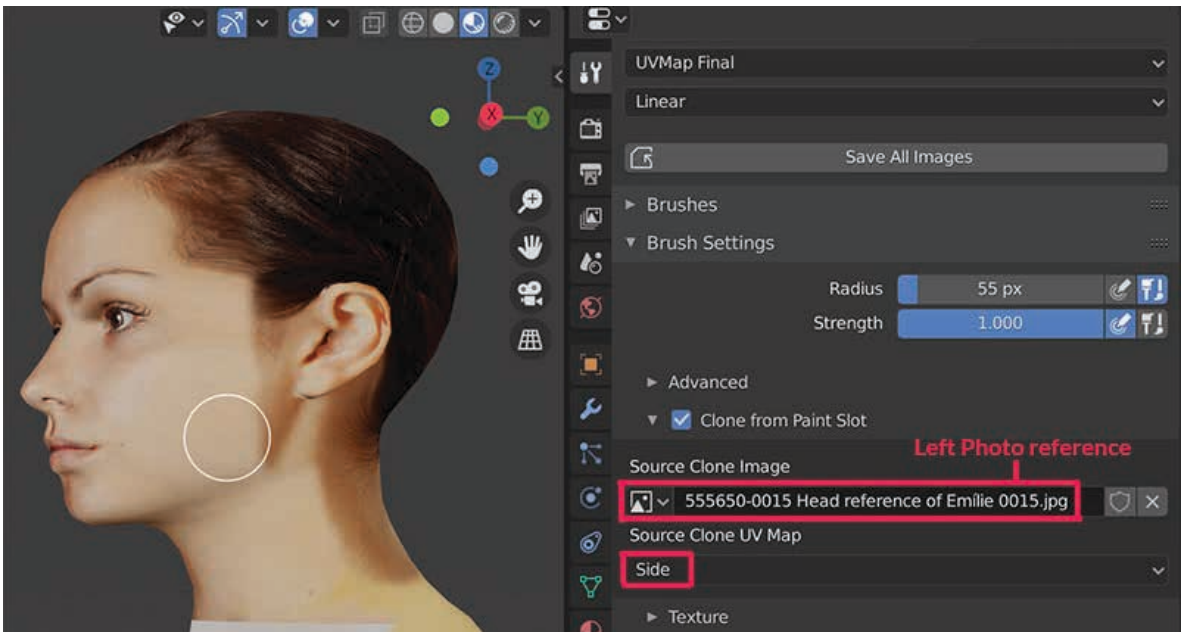
Also, make sure the correct UV Map is selected in the UV Map window, and turn on **Backface Culling** so you don't paint the back of the model while you are working on the front.



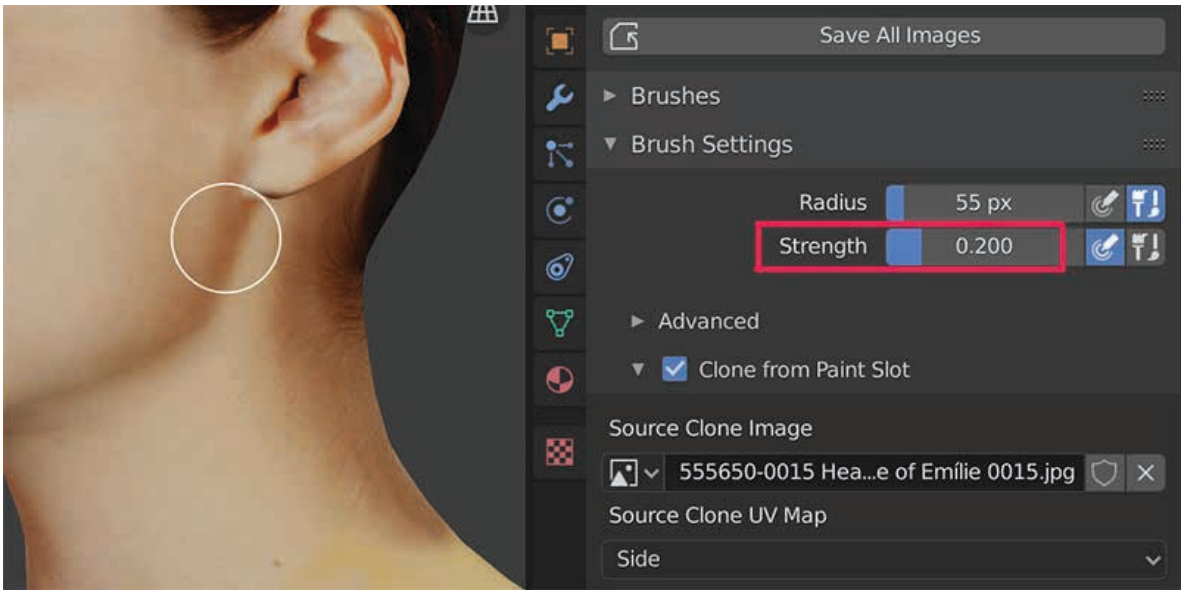
That's a lot of things to set up, but... the result is so much fun: We can now start painting the face onto the model:



Predictably, the Front UV map only gets us so far, which makes sense. So it's time to switch to our **Sideview**. Change the **Source Clone Image** to the Left photo reference and the **Source Clone UV Map** to the Side UV map. Then continue with your Clone brush to fill in the details.



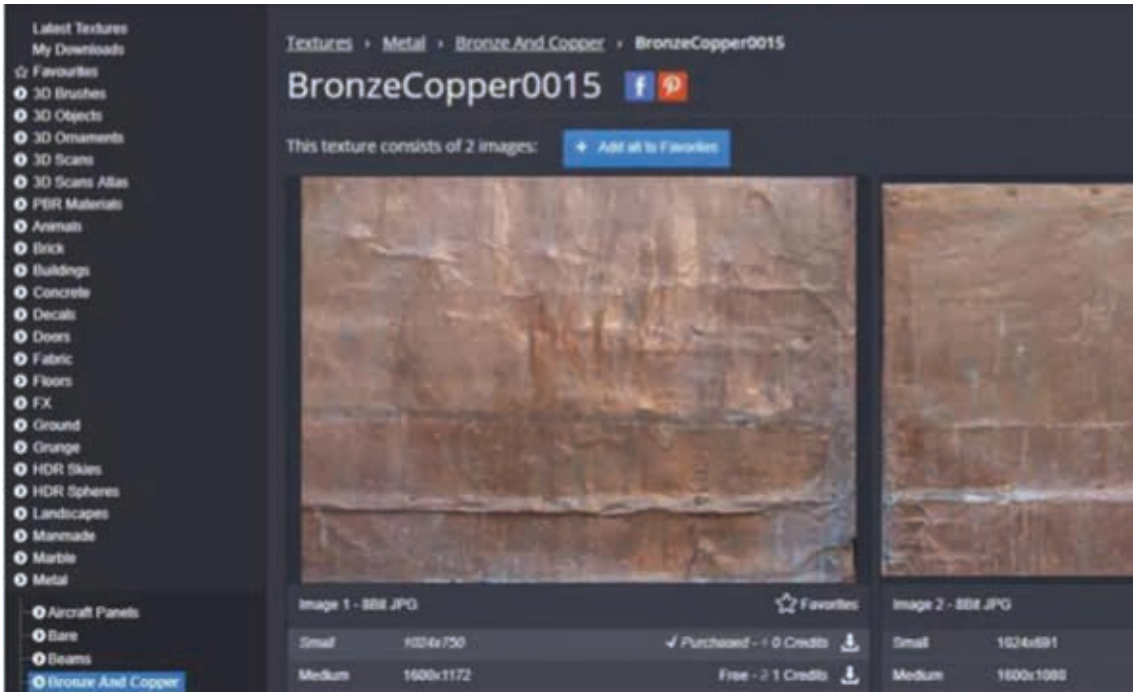
Now we're getting somewhere! However there is an obvious difference in lightness from both photographs, so we need to make that seam a bit more smooth. Lower the Strength value of the Clone Brush and paint over the parts where the two textures overlap, until you get a better blend.



And there you have it, that's the basic workflow for clone painting from projected photo textures. Of course, you can add more angles that you have photographs for to get an even more accurate result.

Make Seamless Textures Easily In Gimp

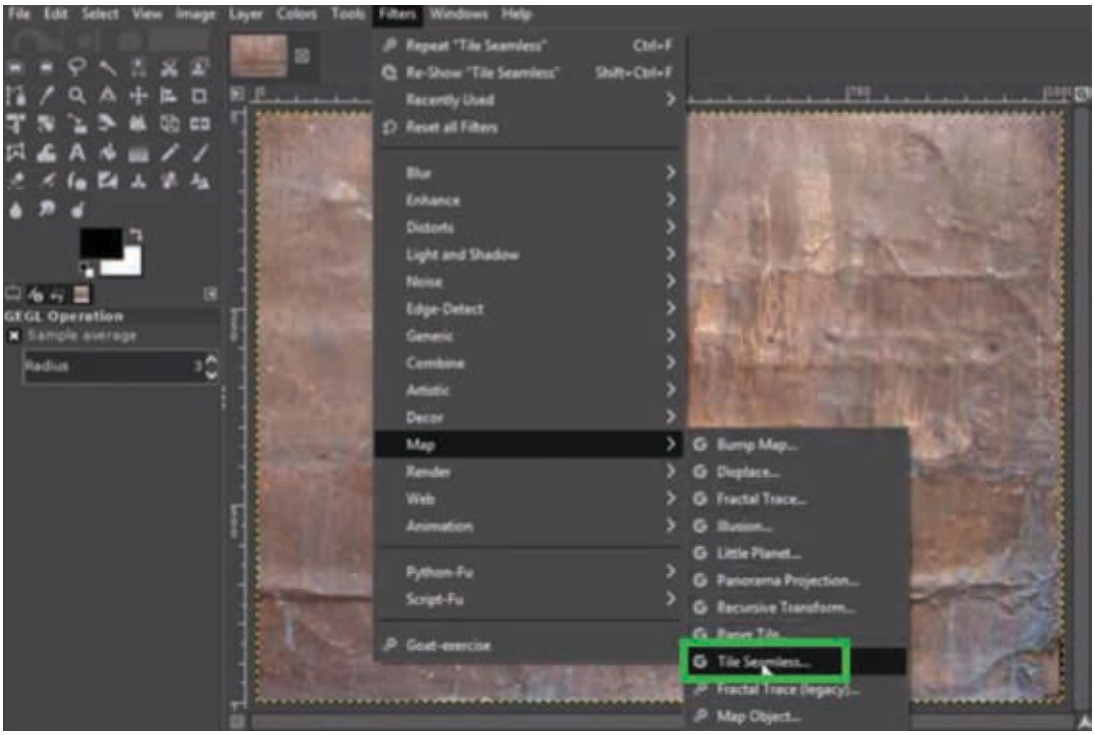
If you find a nice texture which is not seamless, like this photograph on Textures.com:

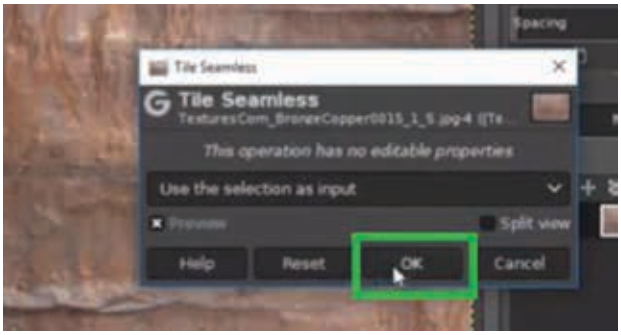


Open it in the free image editing program Gimp:

Click here to return to the index.

You can very easily make it tile seamlessly in Gimp. In Gimp go to Filter > Map > **Tile Seamless** > **OK**



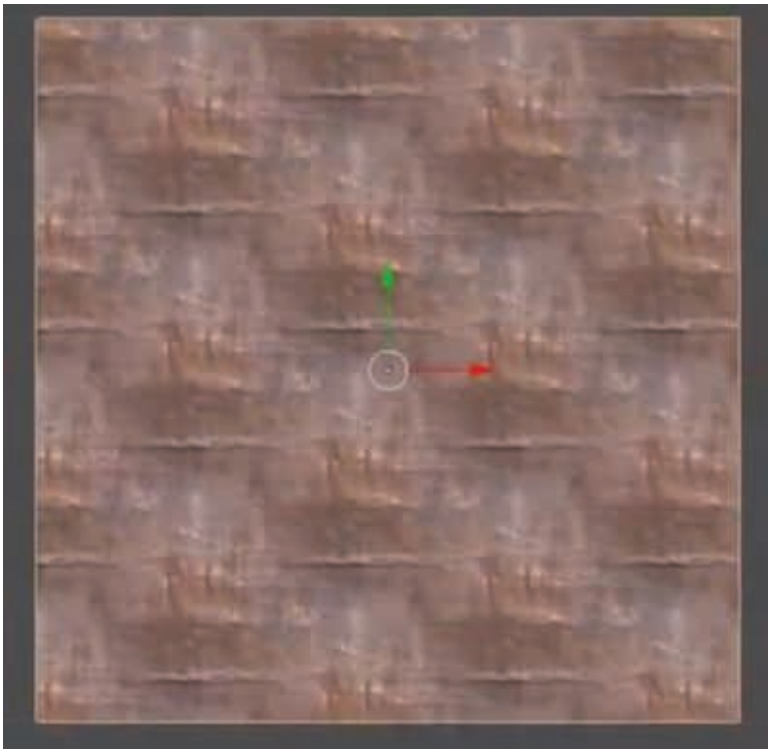


Click on **OK.**

That's all, now you can export it as a new file and use it as a texture in Blender.

Comparison: the original texture tiled 3 times with obvious seams...

...and below the result when it is tiled seamlessly. Of course you can still see that there is repetition, but at least they blend into each other nicely.



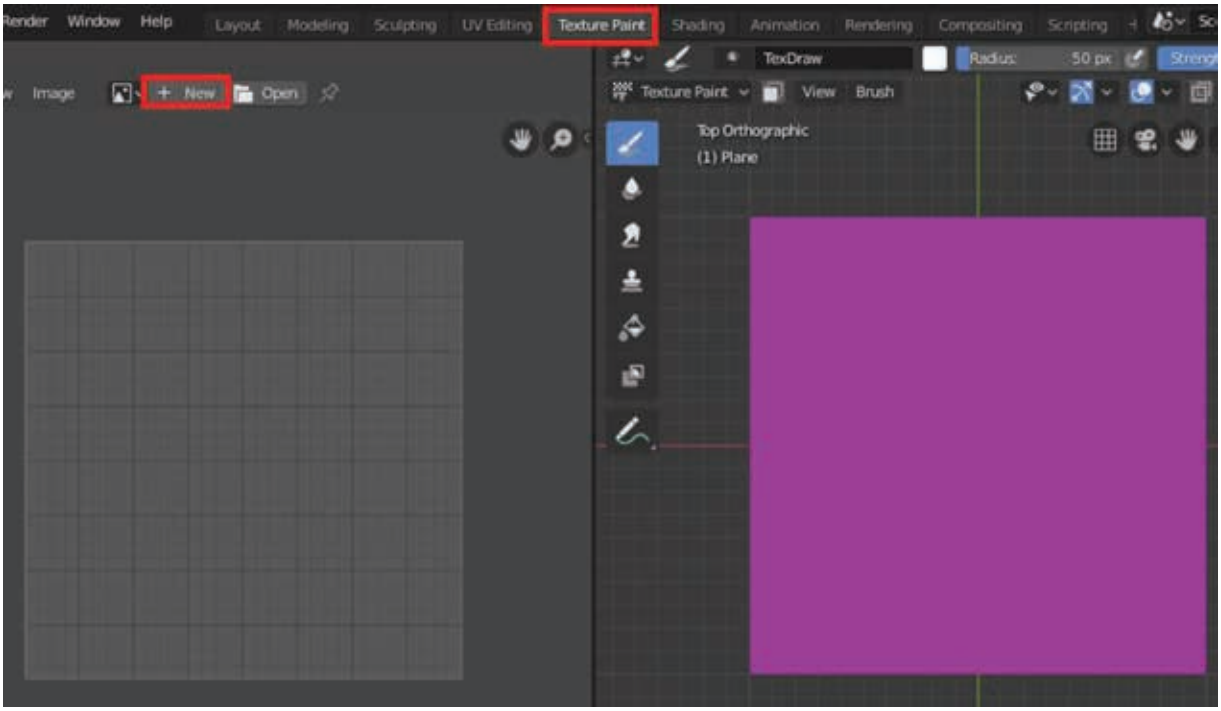
My Painting Workflow In Blender

My painting workflow in Blender is pretty simple. I believe it's important to learn to use some simple tools really well, rather than learn many tools a little bit.

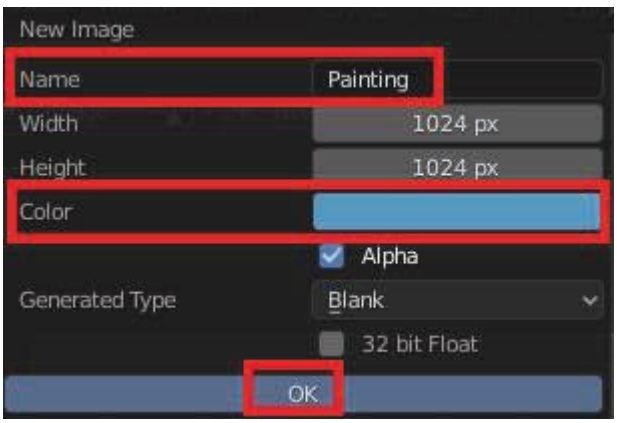
I've deleted the default cube and I have created a simple plane with Generated UV's (an option that you can check when creating any primitive object, which by default is turned on in Blender 2.8).

Use the convenient **Texture Paint** layout by clicking on the corresponding tab.

The plane is purple because there's no texture yet. Click on "**New**".

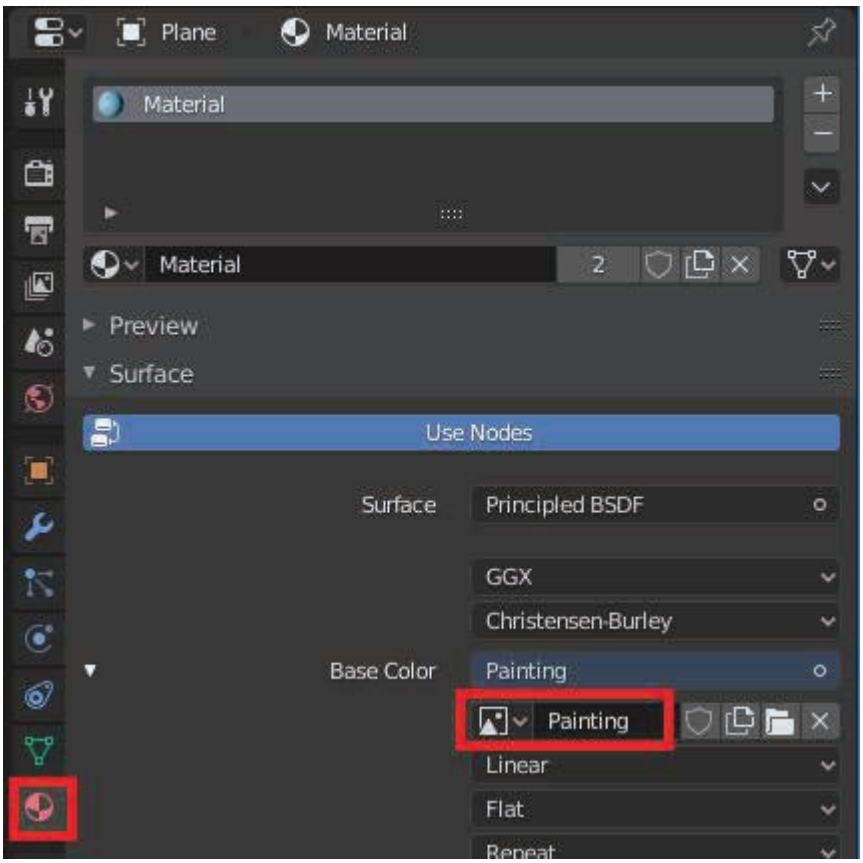


While creating the texture, give it a name and a base color. I choose a blue color in this case.

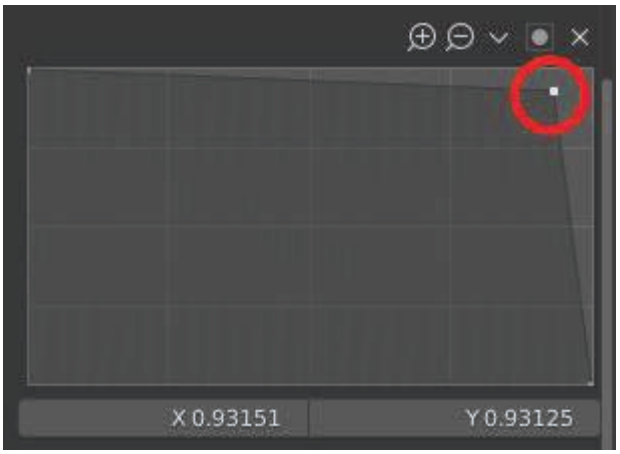


Click **OK**.

Make sure the plane has a material and the material color is set to image texture. Now you can choose the texture which is already loaded in memory. It's called "Painting" in this case.

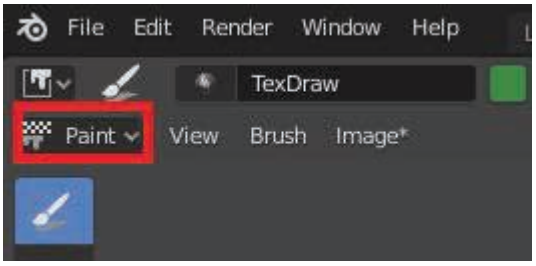


Because I've spent a lot of time painting with a similar brush in Photoshop, I like to use a sharp brush, like I have shown how to make in the previous tip. It uses this Falloff shape (see previous tip for more info):



The strength of the brush I use is always 100%. To blend between values just color pick in between them using keyboard **shortcut S** (for “Sample color”). You can either paint directly onto the plane or onto the texture in the UV window.

If you can not paint directly in the UV window, make sure it is set to “Paint” instead of “View”.

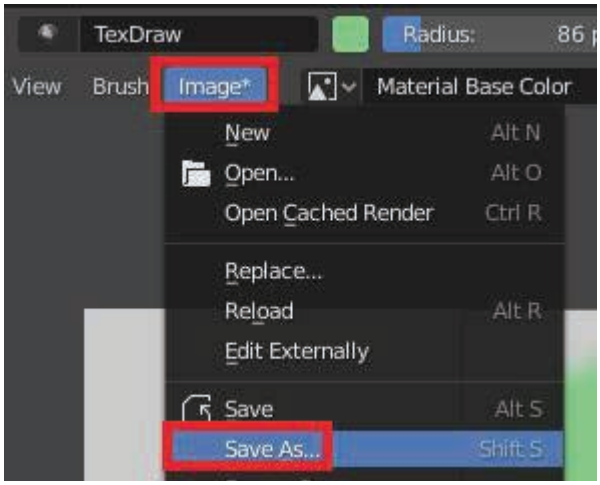


Initially I lay down some basic colors from the Color Picker, after that I color pick with **S**.

To change the brush radius **press F and drag** to increase or decrease the brush size. Important! Blender does NOT save your texture when you save your file.

You need to save it manually as an image file.

In the UV/Image Editor go to Image > **Save As**... Don’t forget…



How to use Leaf textures

Making realistic looking leaves for a tree is a matter of getting the right textures and knowing how to set them up.

Here we'll look at using free textures to create a low-poly leaf to use on a tree model.

You can find a lot of leaf textures on Textures.com. The small version of the textures (512x512) are free!

For leaves, that's really all you need unless you are going for a close-up.

For this example, I'll use this "Parthenocissus" (commonly known as "Creeper") leaf texture set.

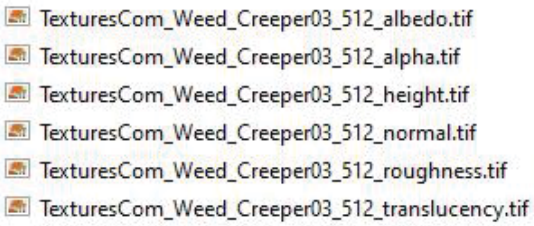
You can find it here:

https://www.textures.com/download/3datlas0177/132892

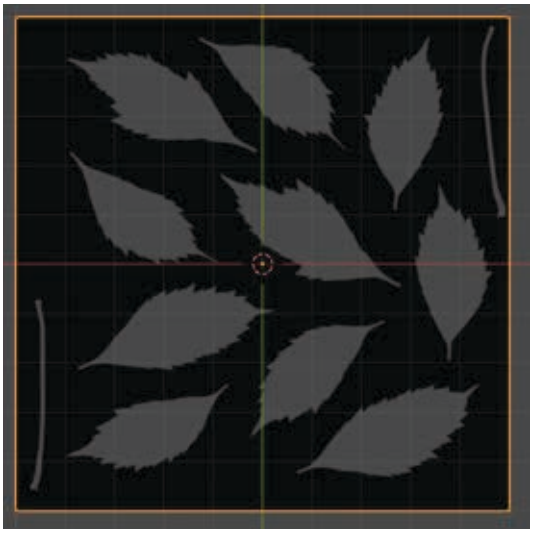
The low resolution maps are free.

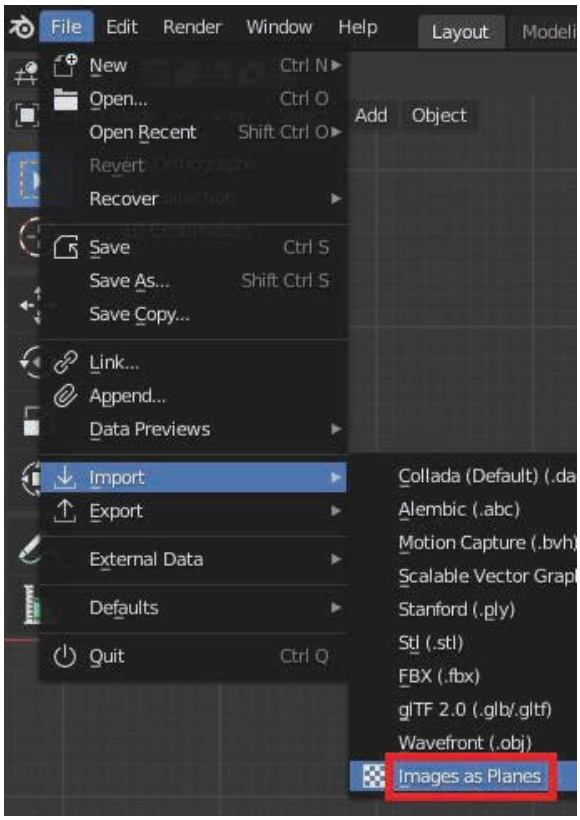
Once you've downloaded them all, you'll have 6 textures.

I normally use only the Albedo, Alpha and Translucency (to save on computer memory usage), but I will explain what they all do.



First, we'll model the shape of the leaf. Using the Import Images as Planes option in the File menu, import one of the textures. I recommend using the Alpha, as it's easier to see the shapes.





Press TAB to go into Edit Mode, then using **Ctrl+R** add loop cuts, and move them to isolate one leaf.

Select the faces around the leaf and delete them by **pressing X** and choosing **Faces**.

You can add some more detail to the leaf. Some shortcuts to remember:

Press Ctrl+R to add loop cuts

Press J to connect two vertices together.

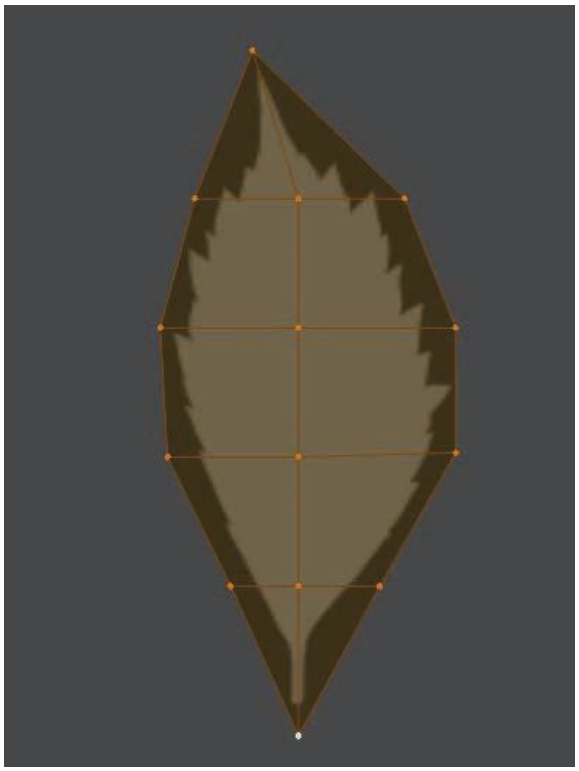
Press G twice to slide a vertex

(without affecting the texture)

Press X to delete.

Press G twice then press C to slide a vertex first inwards (where there is topology) and then outwards (where there is none).

Now we have the basic leaf shape.

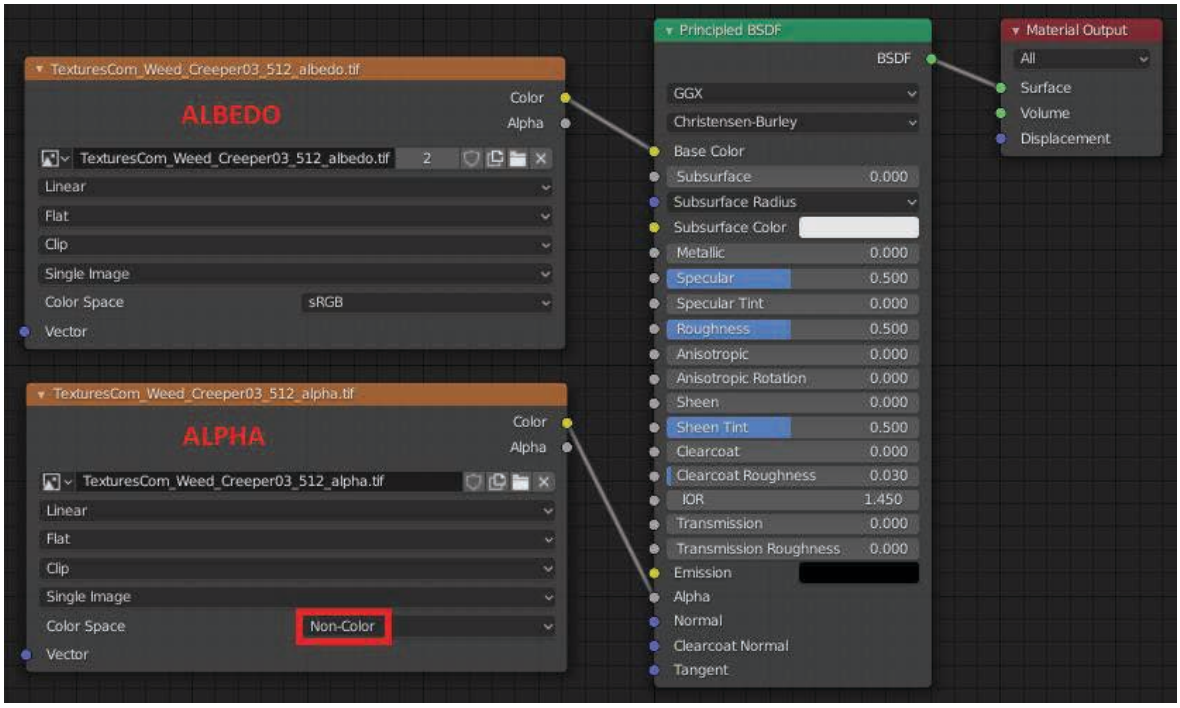


Using **Proportional Editing** we can quickly make the leaf have an organic looking bend, so it is a bit less stiff and flat.

We have the basic shape now.

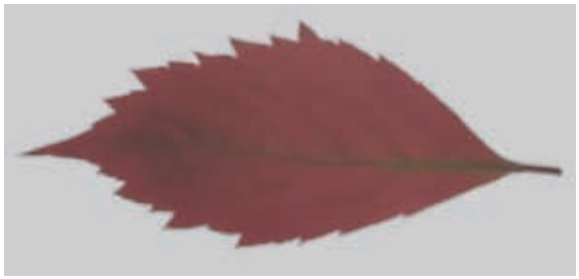
Let's use to the **Albedo** texture. Albedo basically means the color texture, without any baked in shadows.

Below is the node set-up with just the Albedo connected to the Base Color, and the Alpha texture connected to the Alpha input of the Principled BSDF.



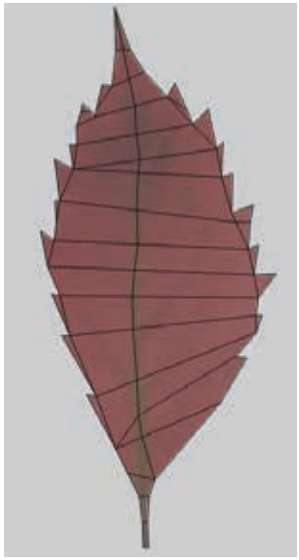
Remember to turn the Alpha (and any other texture that isn't the Albedo) to "Non-Color" instead of sRGB.

Pictured on the right is what we have so far.

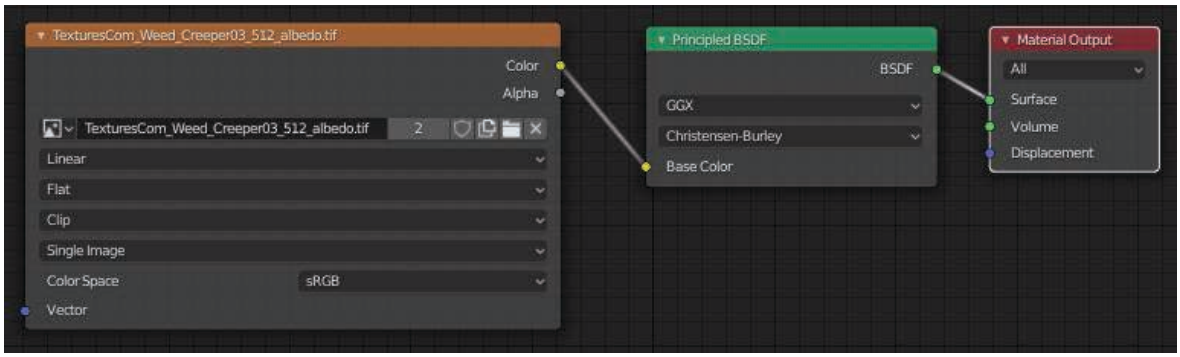


We have an alpha texture, so we don't need to model the detailed shape of the edges of the leaf. We only need to plug in the Alpha texture and the parts of the leaf that need to be transparent will be so.

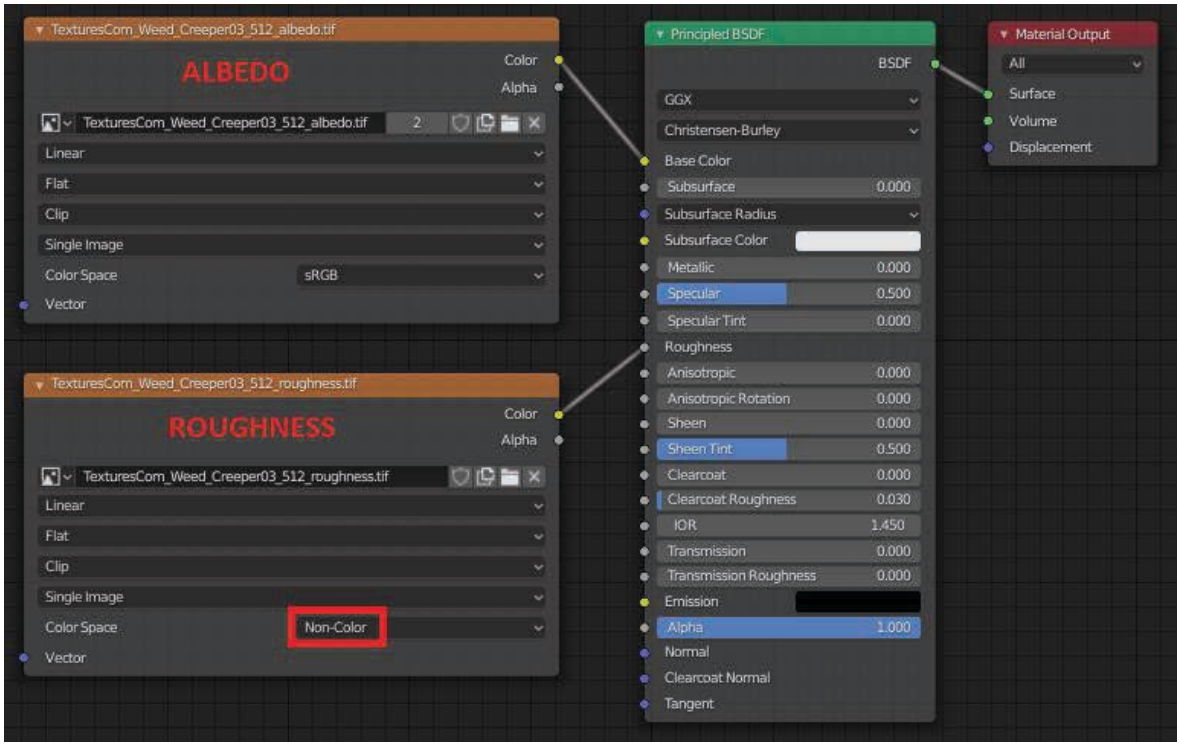
However you may still want to model those edges. Even though that means more polygonal detail, which may seem illogical for something for which you need probably if not hundreds, then thousands of duplicates in your scene. That's because with many objects that have an alpha channel, you'll need to seriously crank up the transparency value of your render. It actually makes more sense (in terms of resource allocation) to go with thousands of instanced leafs without an alpha channel instead.



In other words, with all that detail modeled we don't need our Alpha texture anymore. So now we have this simple node set-up (Principled BSDF temporarily minimized with the **Ctrl+H** shortcut).



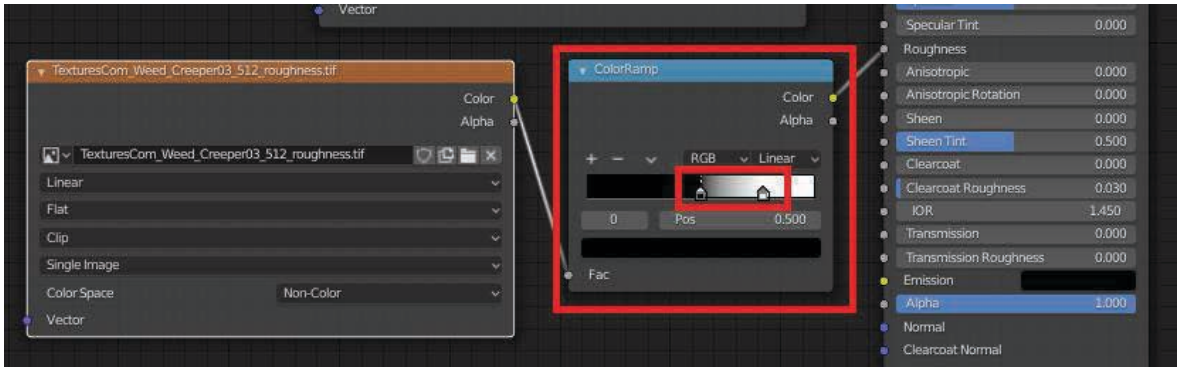
Now we add the Roughness texture. Set it to **Non-Color**.



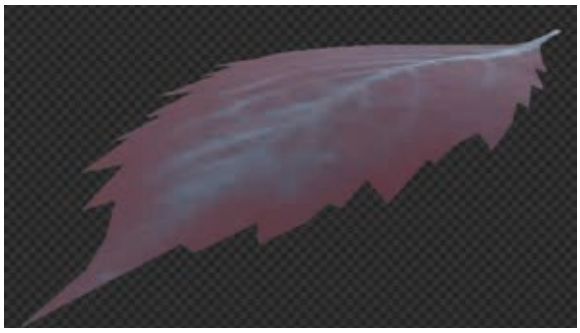
What does this do? Let's look at the comparison without (left) and with (right).

On the left you see that everything is equally shiny. On the left, you see that only certain parts are shiny, and other parts are "rough", based on the Roughness texture.

To control the Roughness further, you can add a **ColorRamp** node in between the Roughness texture and the Principled BSDF nodes. Then, you can increase the contrast of the Roughness texture, making the effect more pronounced.



You can see the effect of the Roughness map combined with the ColorRamp node on the right.



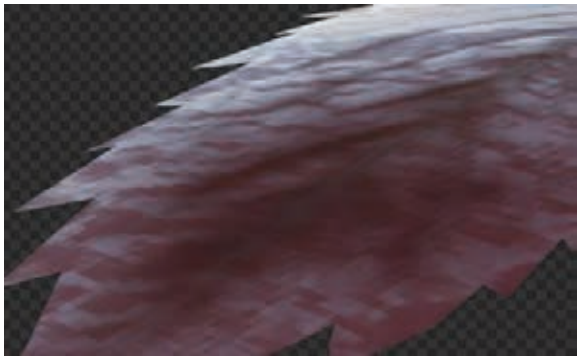
You can also invert the Roughness map by switching the points on the ColorRamp around. Looking at reference pictures, I think in this case this looks more correct.

Adding the Normal texture map gives the leaf some bump, or fake displacement.

Normal maps (like Bump maps) create a kind of displacement effect for which you don't need any additional geometry (such as would be the case with a Displacement map).

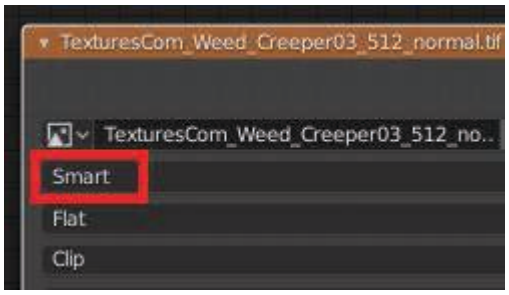
The Strength value will almost certainly be way too high. Turn it down until it looks good.

Using only a small part of an already small texture, the pixels on this Normal map become quite noticeable.



Although this will probably not be an issue if the leaves are seen from afar, you can still make it look better with a simple trick.

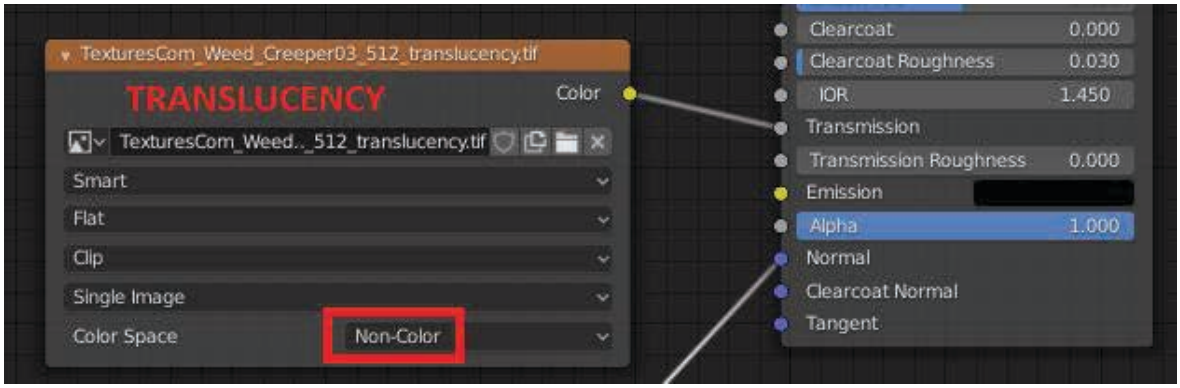
Set Interpolation to **Smart**. Looks better!



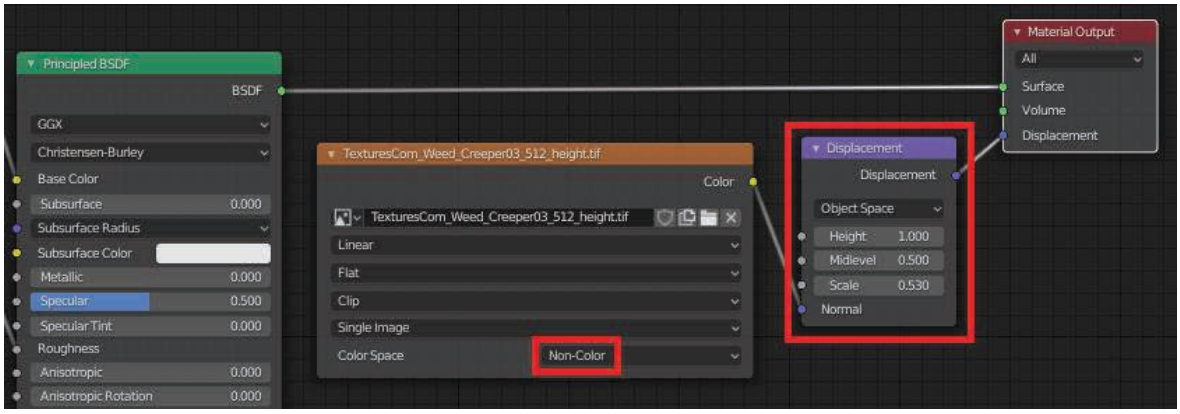
The next map we'll add is the **Translucency** map. This one is important for realism - it lets light pass through the leaf. otherwise, your leaf will block light completely, which will make it look like a fake, plastic leaf (and even that would probably let some light through in reality).

Below you see the difference without translucency (left) and with (right).

Notice how you can see the sun shine through the leaf? It makes it look 100x better! Below you see how to connect the node.



This is good enough for me, but there's still one more texture we didn't use and that's the **Displacement** map. In my opinion not necessary, unless you are going for a close-up of the leaf. Here's how to plug it in (It's called a Height map in this case).

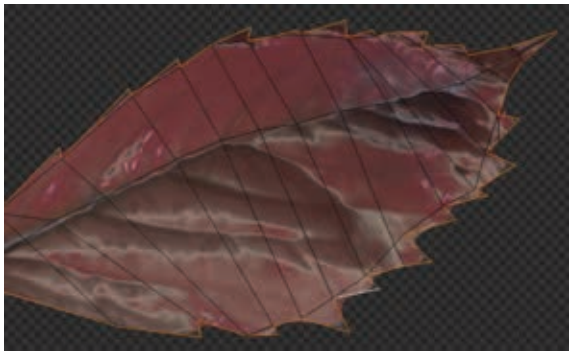


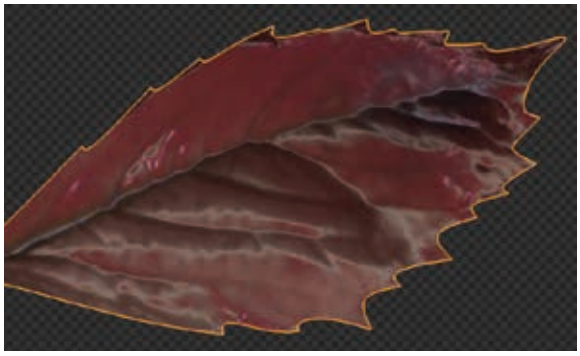
Note that for a Displacement map to work, your leaf probably needs more detail. For this we can add a **Subdivision Surface modifier**.

You'll notice that after adding a Subdivision Surface modifier, the edges of the leaf don't look good anymore. That's because this mesh was modeled without using a Displacement texture in mind (otherwise I'd have used all quads instead of using triangles).

The resolution of the texture is also frankly too low for displacement, remember we are using only a small section of a 512x512 texture.

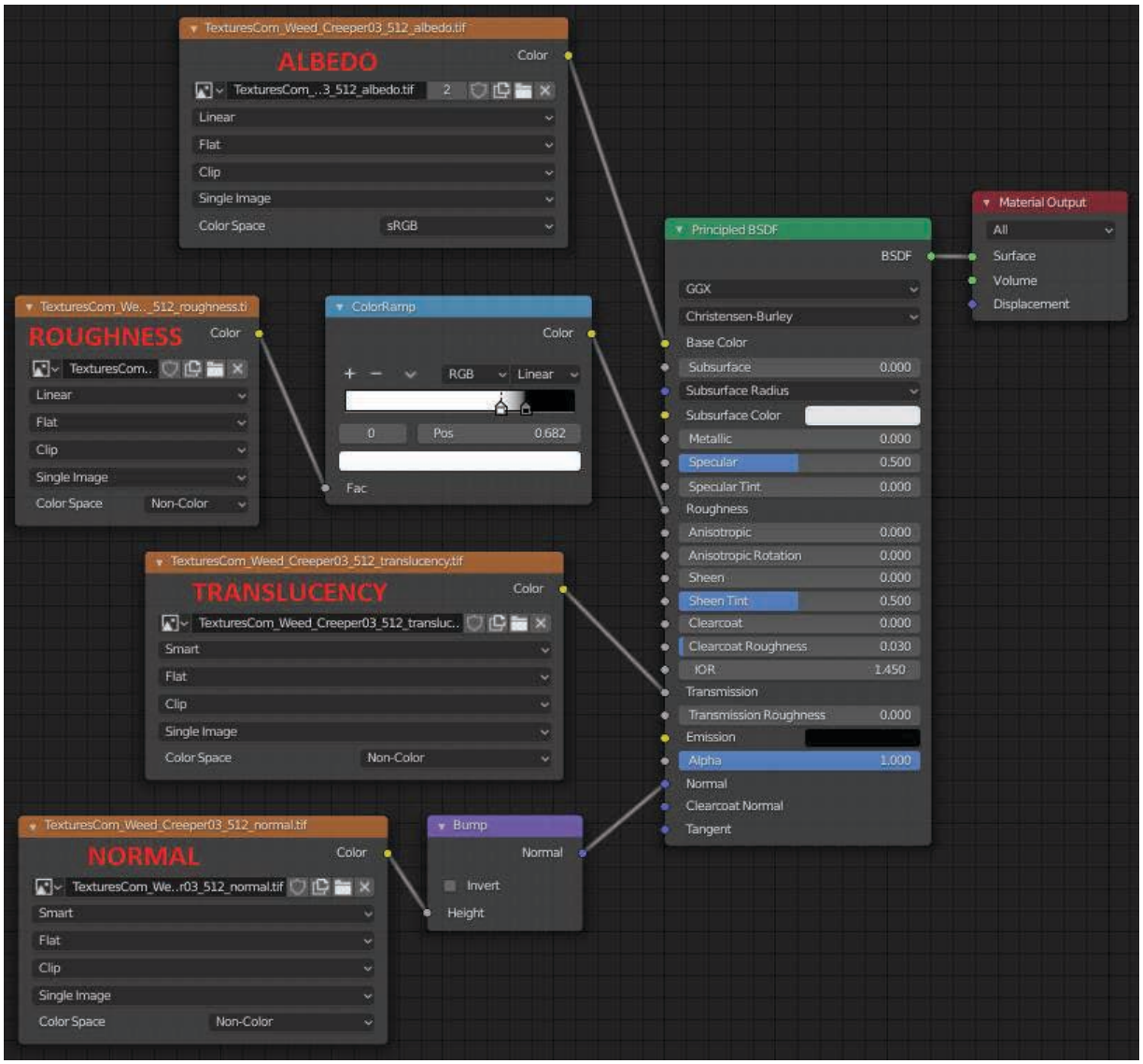
Select the outer edges, then **press Shift+E+1** to crease them.





To be honest, for a close-up of the leaf you may want to more carefully model it with quad topology instead of the quick and messy mesh I'm using here. That way, the Subsurf modifier will do a better job. But for leaves, you'll seldomly need a Displacement map, unless it's for a close-up.

Below is the entire node setup (minus the optional Displacement map).



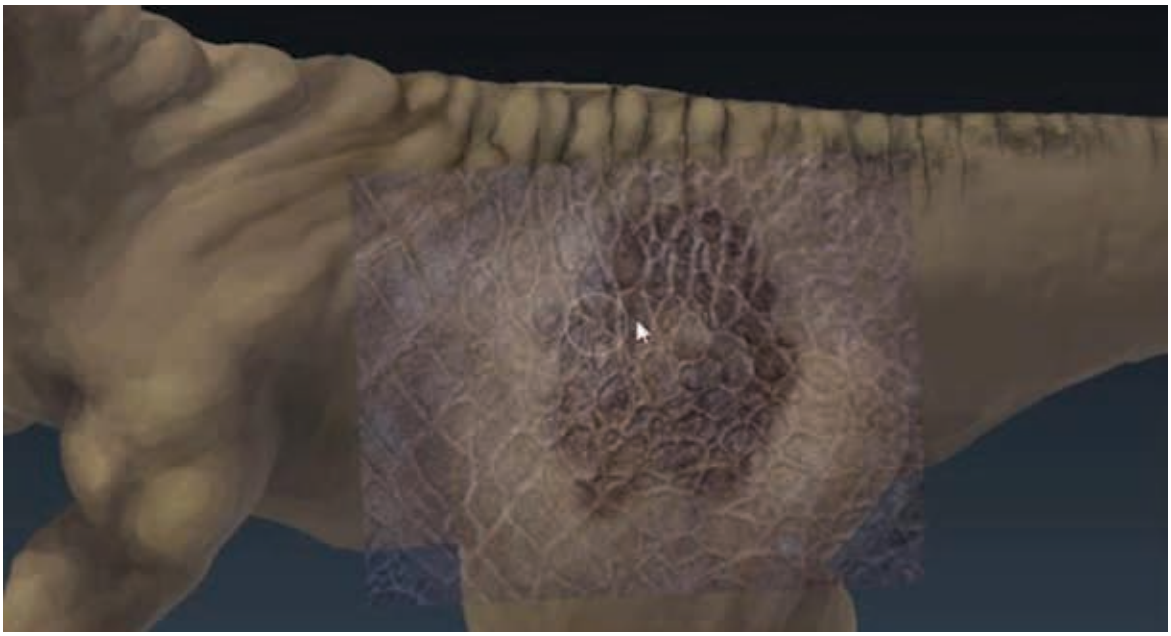
On the right you see an example of a leaf model with a 4K texture.

In this case, the Displacement map works well and the Normal map gives fine bump detail. This is the way to go for a close-up of the leaf.

Texture Painting with Stencils

Did you know that in Blender you can quickly add complex texture details by using an image as a "stencil"? This gives you the kind of details that would take forever to paint by hand in just seconds.

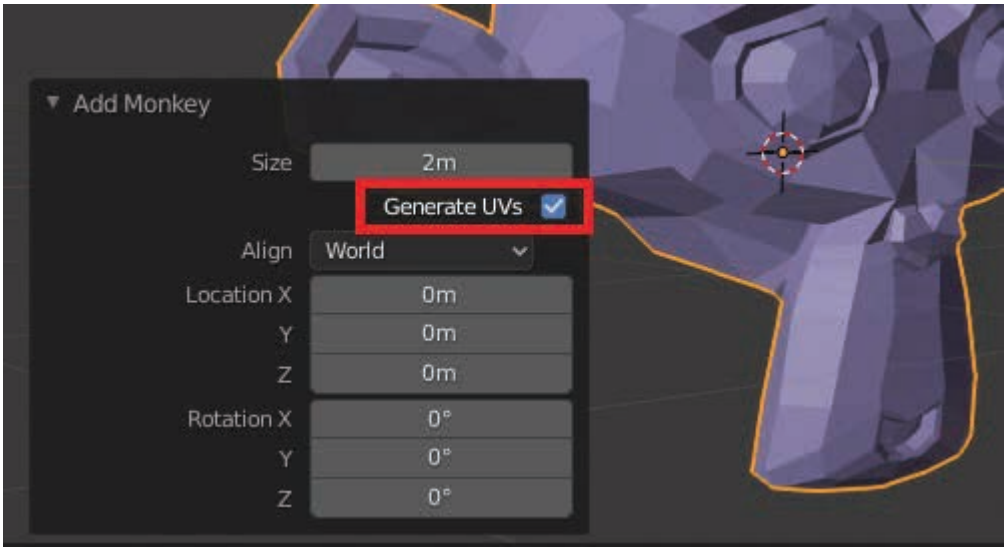
All you need is a model that has is UV-unwrapped as well as at least one image.



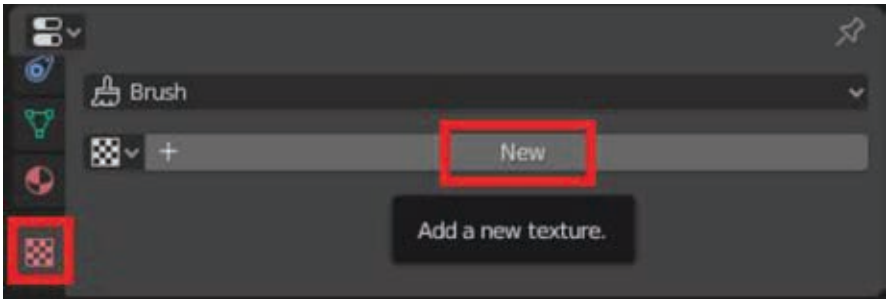
First, collect some interesting images.

This one is from Textures.com. I like to use both Pixabay.com and Textures.com - both are websites that have a lot of images with **CC0 licenses** (in other words, images you are allowed to use without having to attribute the author). Always make sure to check that you have the correct license before using something for commercial purposes.

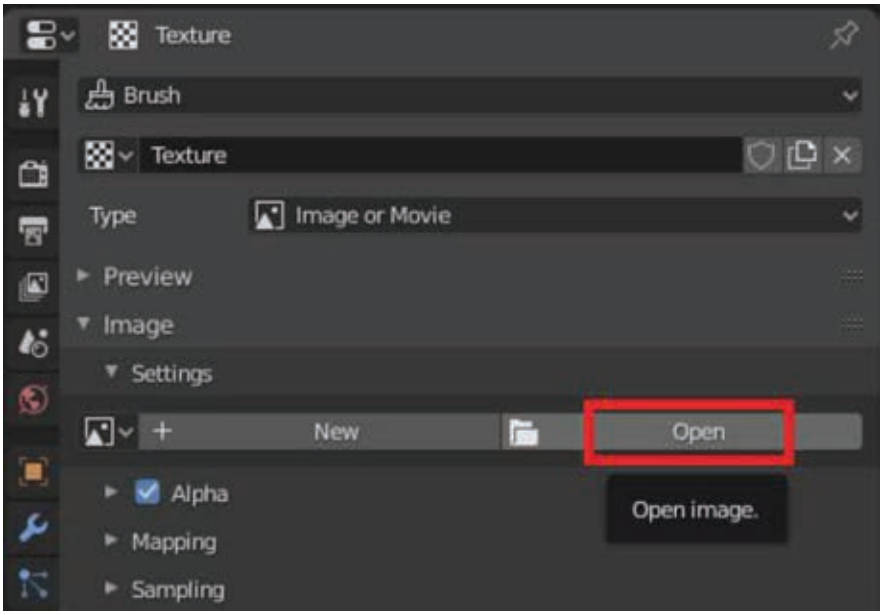
Let's use a monkey for this example. **Press Shift+A** and choose Mesh > **Monkey**. As this is a Primitive object (meaning one of the default objects you can generate), the UV map is generated automatically. In Blender 2.8 this option is checked by default.



Open one of the images in a Texture slot. In the Textures panel, **click on New**.

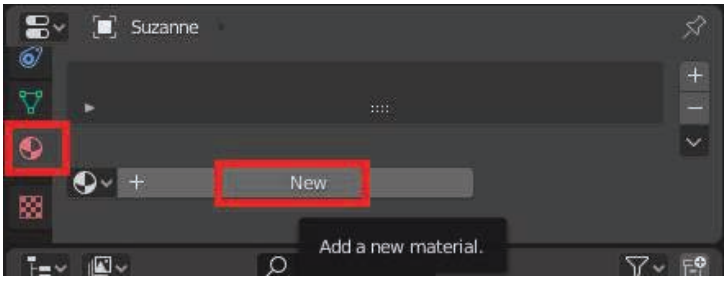


More options will appear. Click on Open and find the image you downloaded.

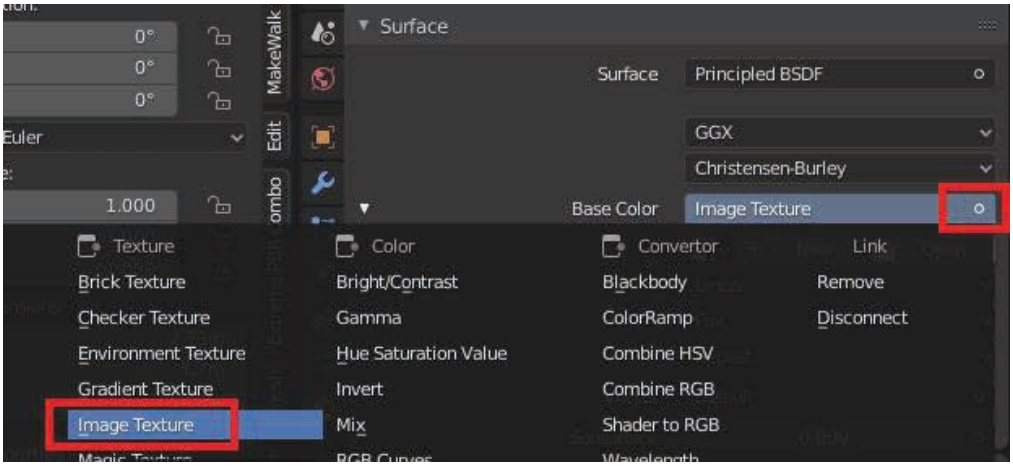


In case your model doesn't have a material with a texture of its own, you'll need to create one as well.

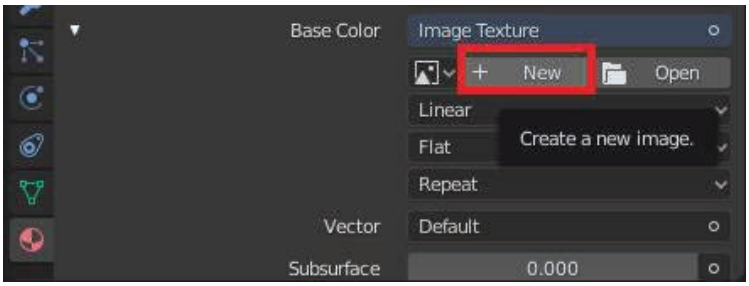
With the monkey selected, **click on New** in the Materials tab.



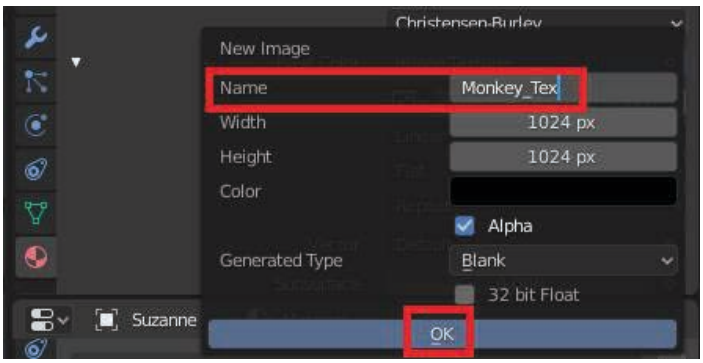
In the Material tab, as Base Color choose **Image Texture**.



Then we'll create a blank texture by pressing New.

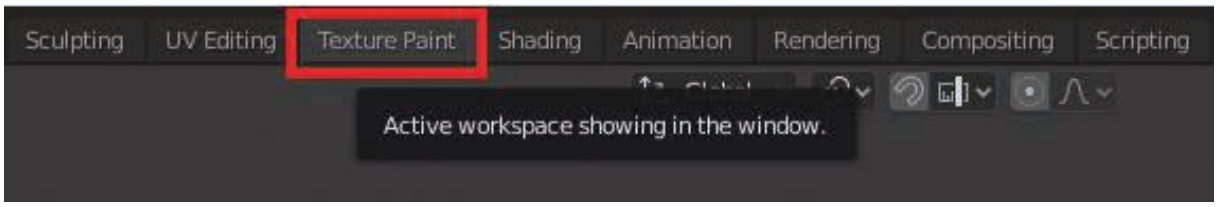


Give it a clear name like Monkey_Tex. If you want you can choose a base color here.

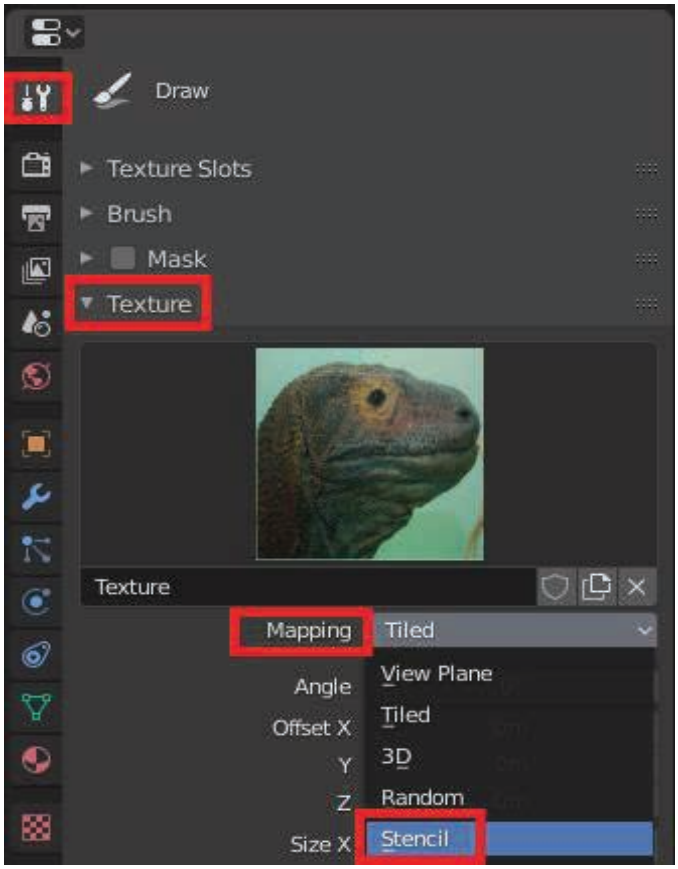


After these steps you can open the image you downloaded in the Paint Tools texture slot and use it to paint onto the blank texture.

Go from Object Mode to Texture Paint mode by selecting it in the top left corner. Alternatively you can click on the new "**Texture Paint**" button in the top row of the UI to get a Texture Painting optimized workspace. This saves some time and possible frustration.



Set the image as "Stencil". The texture will now appear as an overlay in the 3D viewport.

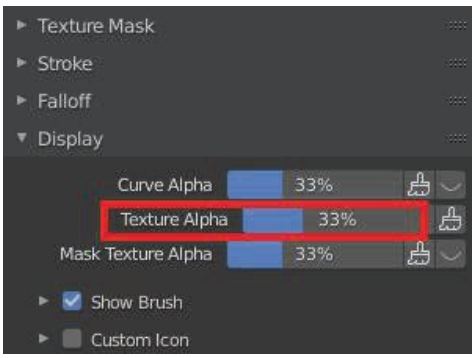


Hold the right mouse button above the stencil and drag to move it.

Shift + right mouse button to scale the stencil.

Ctrl + right mouse button to turn the stencil.

Now paint on top of the stencil to influence the model's diffuse channel. If you want to control the opacity of the stencil, you can do that under the Display settings.



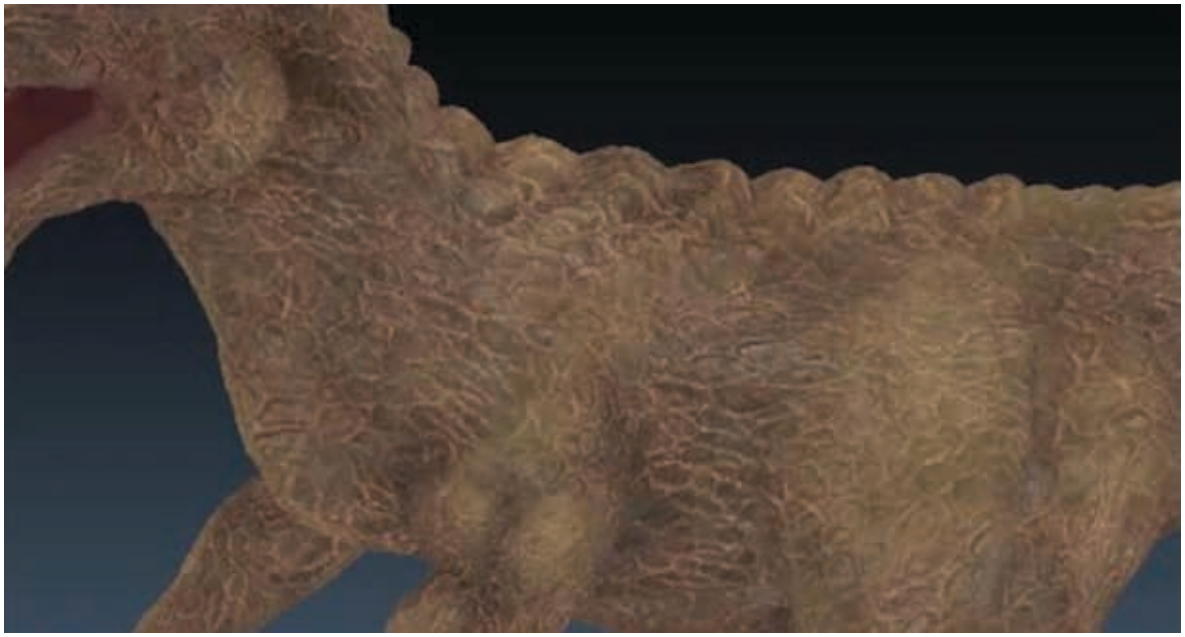
I've noticed that sometimes, the stencil is not transparent even though it should be. In that case, save your work and re-open the file.

Now to stencil paint, you need to scale/rotate/move the stencil in the viewport to match how you want the details to be transferred to your model, and then paint on the stencil.

Here for example I'm painting scales on a dino model. It takes a little while to get used to the controls of the stencil, but after that it's a quick way to add detail!

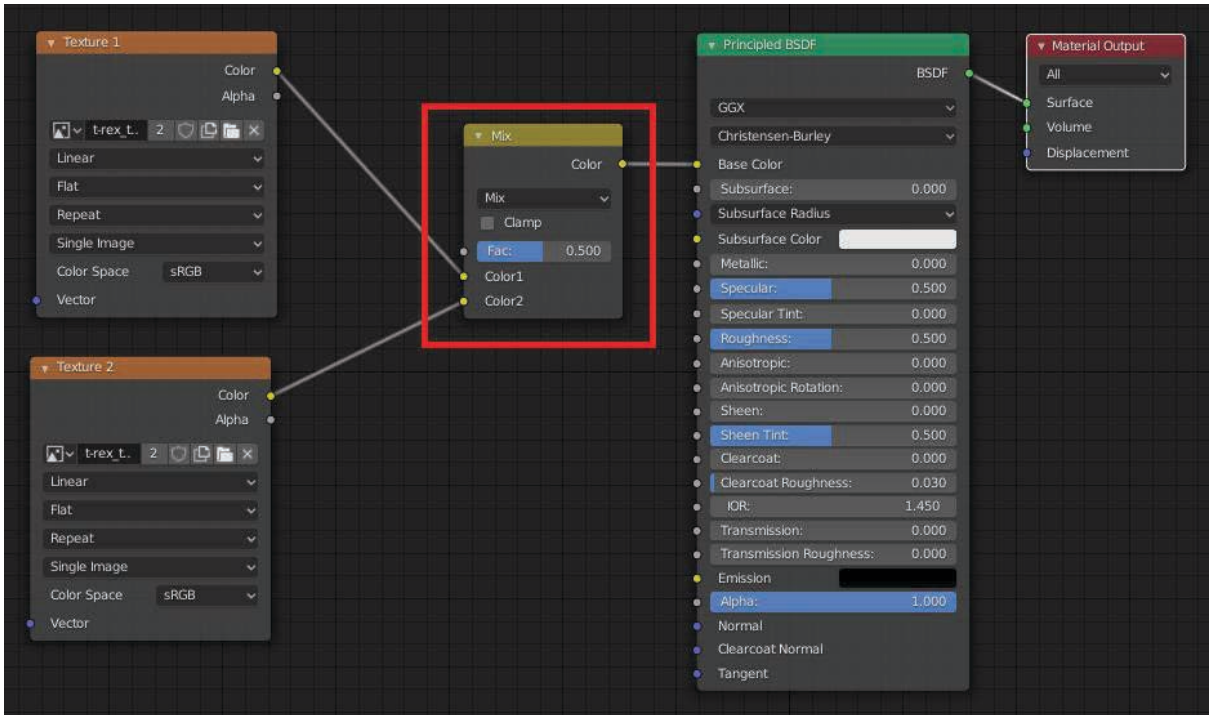
Texture mixing

This is a texture I made using stencils.



I already had a hand-painted texture as well. The two textures (hand painted and the one made with stencil technique) can be combined by using a MixRGB node. Mix them with a value that looks appealing, using the "**Fac**" Slider.

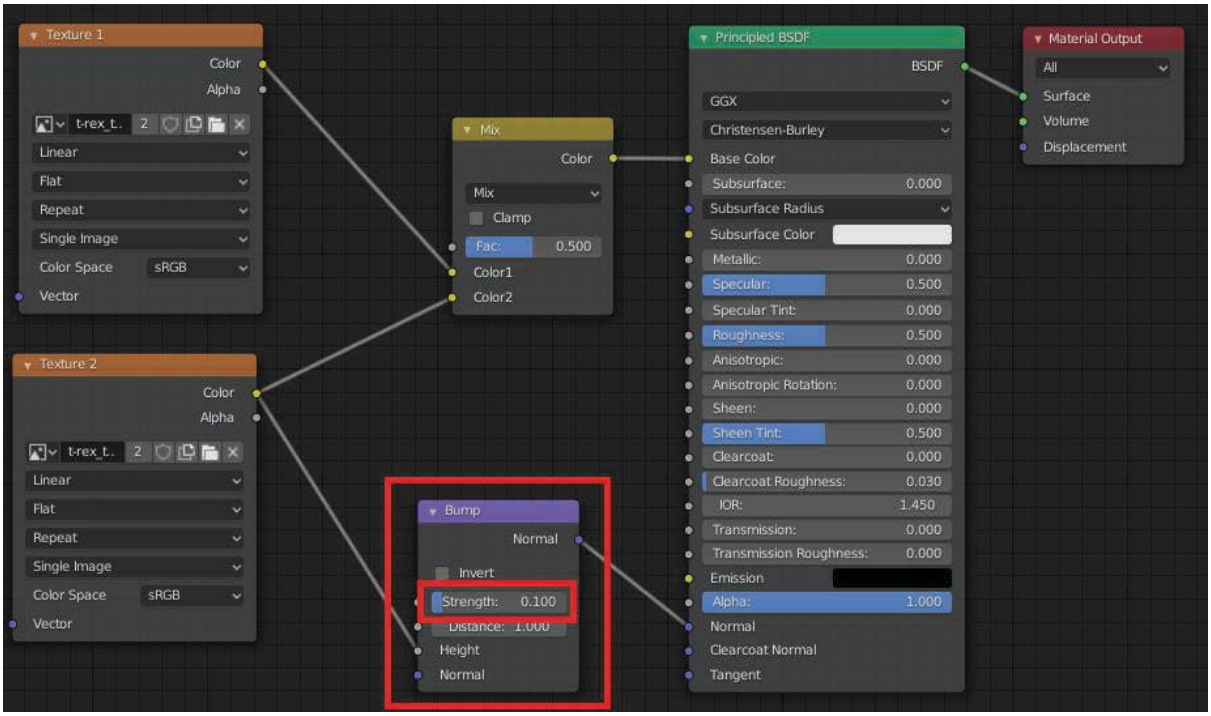
The node setup: left are the two image textures, in the middle the **MixRGB node**.



Make sure you get the right node, as there are two “Mix” nodes. You need MixRGB. To add nodes in the Shader Editor, just **press Shift+A** (just like you add new things anywhere else in Blender) and type the name of the node until it appears below, then choose it.

Sometimes an image texture can also be used as a Bump map. Bump maps (and Normal maps) are an easy way to make a material catch the light in a more realistic way. Everything around you has some “bump” in real life as well. Try to make it a habit to use them. Even just a simple procedural Noise texture used as a Bump map can make a material look more real.

To do this, just **press Shift+A** and find the **Bump node**, then plug it in as shown here.



By default, the **Strength** of the **Bump map** is set to 1 which is way too high 99% of the time. **A value of 0.1** is usually more subtle, but this is something that needs to be determined for each individual case.

Rust Texture that works on any object in Eevee

Let's create a shader that adds rust to this car model. Although this particular model by Sketchfab user Renafox is UV-unwrapped, we'll use **Object Texture Coordinates** for the material, that way we can reuse the material for any other object, even ones that are not unwrapped.

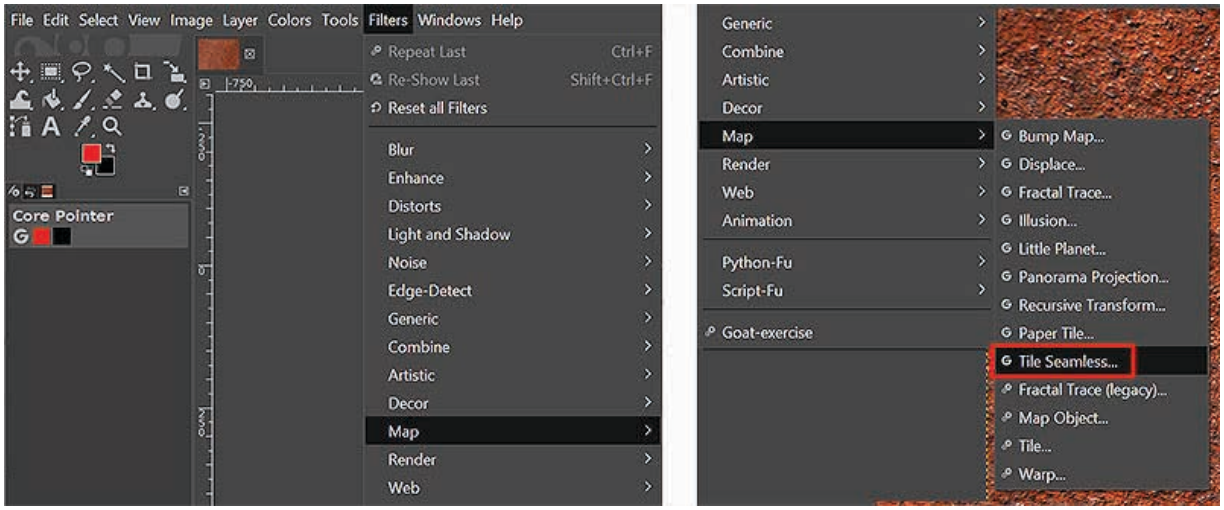
To create a nice shader, it's always good to look at a photograph of a real object with the features that you want to imitate. What do you see when you look at this picture? We can identify these things:

1. Two materials: **Rust** and (weathered) and **Car Paint**
2. Rust becomes visible **at the edges**
3. Rust is more pronounced **at the top**
4. **Streaks** of rust running down the side.

In other words, for this material we'll need to create the car paint as well as rust separately, and then mix them together in a way that makes the rust more visible at the top and at the edges, and with streaks at the sides.

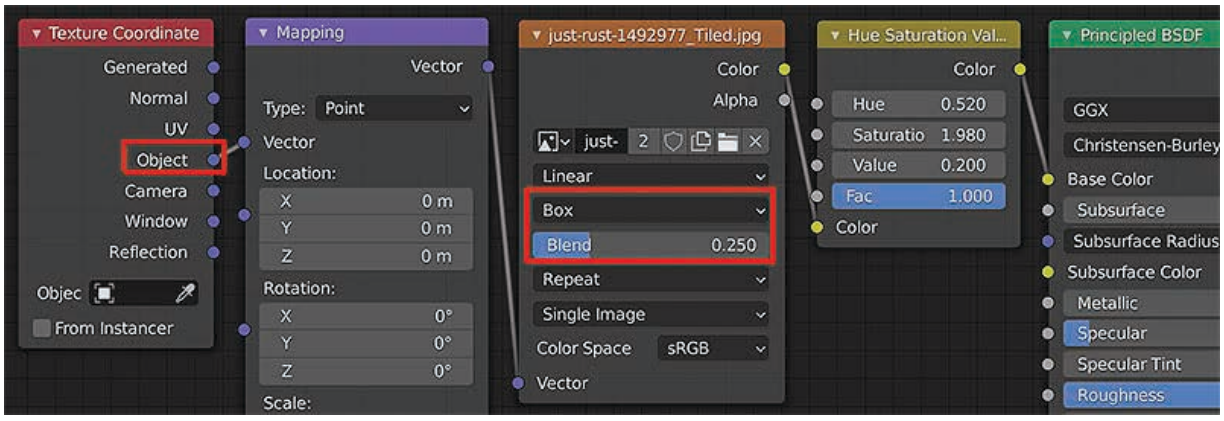
When it comes to something like rust, you can create it procedurally but in my opinion, there's no substitute for photographic textures. I found a free CC0-licensed picture of rust here.

To make sure it tiles seamlessly, open it in Gimp and go to Filter > Map > **Tile Seamless**, then save it again with the affix "**_tiled**" so that you know which one to use in Blender.



Add a **new material** to your object. In the **Shader Editor** window, select the Principled BSDF node and **press Ctrl+T** to add the image texture nodes (this is assuming you've enabled the Node Wrangler add-on).

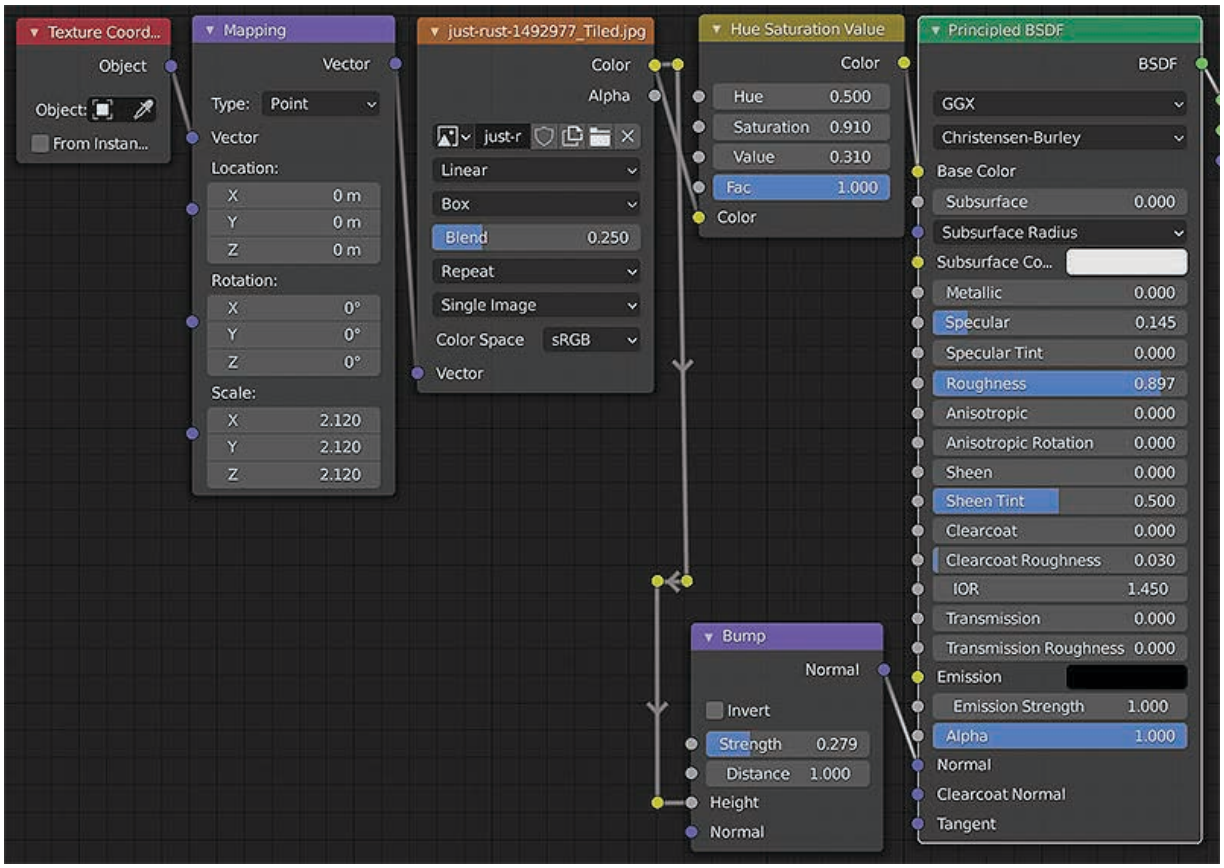
Add the tiled rust texture and set Texture Coordinates to **Object**, Projection Type to **Box** and increase the **Blend value**.



Rust isn't very shiny at all, so reduce the **Specular** value and increase the **Roughness**. If you like, you can add a **Hue/Saturation** node to control how bright and saturated the rust looks.

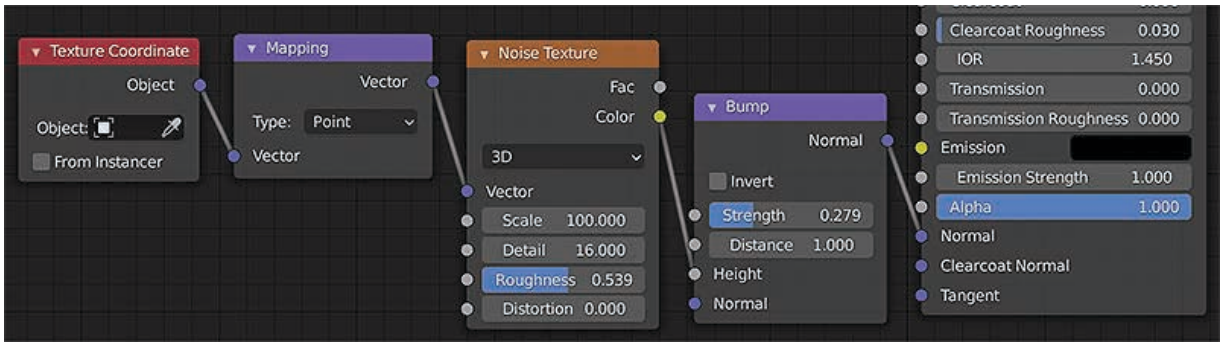
Again, look at your reference image to fine-tune it, don't do it from your imagination of what you think rust is supposed to look like.

If you want you can also just use the image texture to drive a **Bump map**, which breaks up the perfect surface just a tiny bit, adding more realism. Some purists might frown upon using the same image texture for this, but I think it works well!



However if you want a bit more control over the bumpiness, instead of just plugging the image texture into the Bump map, add a **Noise Texture** and give it a Mapping node and Texture Coordinates set to Object as well.

Play with the values until they look right for the image texture.

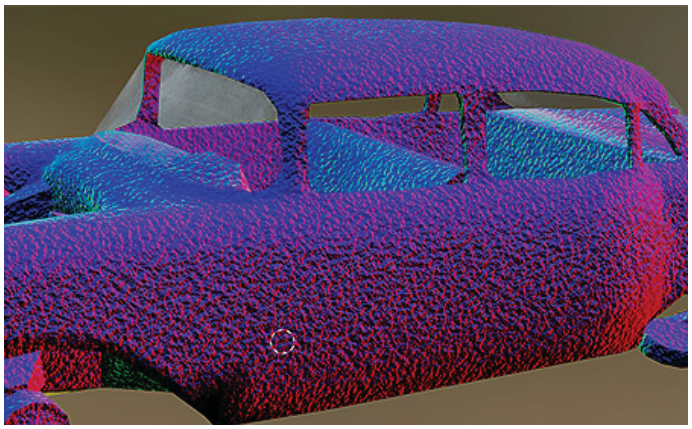


Don't overdo it, you only need a very subtle noise just to break up the smoothness of the surface. If it's difficult to see the result, **mute** the rust texture temporarily by selecting its nodes and **pressing M**.

You can unmute nodes by **pressing M again**.

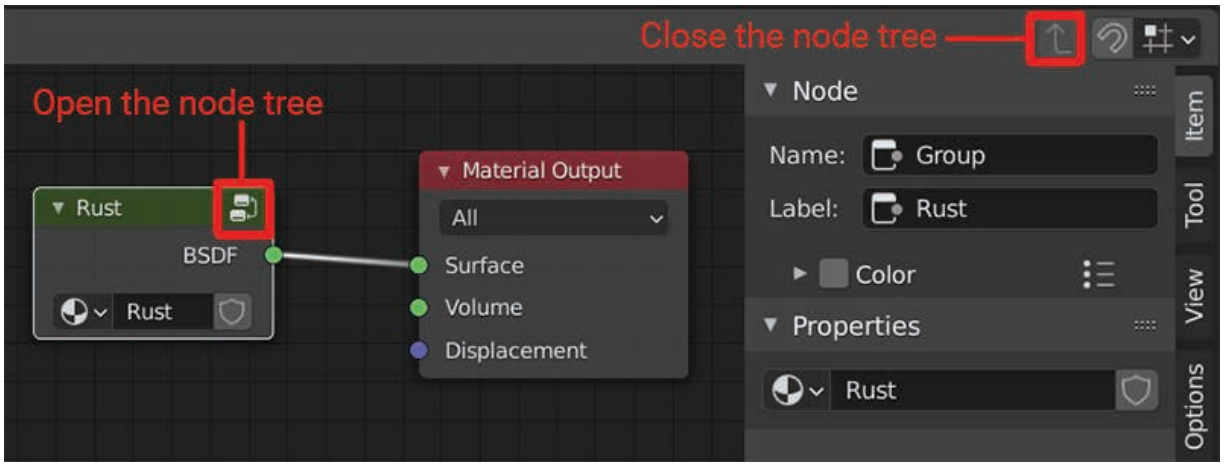
Alternatively, you can select a node like the Bump node, and **press Shift+Ctrl+T** to isolate it and see what it's doing in the viewport.

On the right you see an exagerrated version of the bump.



Great, now we've got our rust. Select all its nodes (everything except the "Material Output" node) and **press Ctrl+G**. This creates a **node group**.

Click on the up arrow icon and your whole node tree will have become one convenient node, which you can then give a name in the Option tab. If you want to edit any of the individual nodes again, click the node tree icon. You can also quicly expand and collapse node groups by selecting them and pressing TAB to toggle.

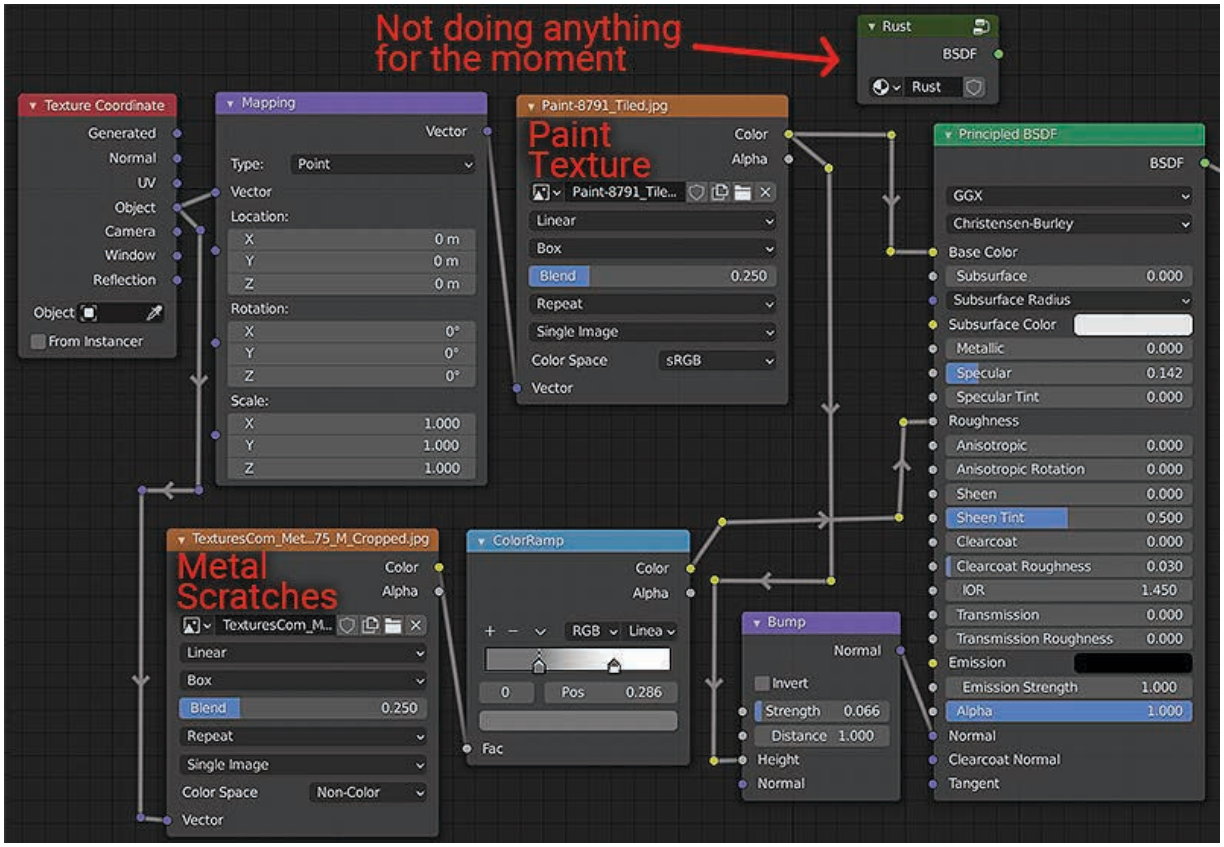


Nice! That sure keeps things organized. Next up is the **car paint material**. I won't repeat all the steps as it's pretty much the same as the rust: Find a suitable picture (I used this one from Textureninja.com) and make it tile seamlessly in Gimp if necessary, and use the same node set-up as before.

In fact you can **copy/paste the nodes** from the Rust node group and connect them to the Material Output node, disconnecting the Rust group. All you need to do is put the new texture in there.

Looks okay, but I'd like some parts to still be a bit shiny, and some to be rough.

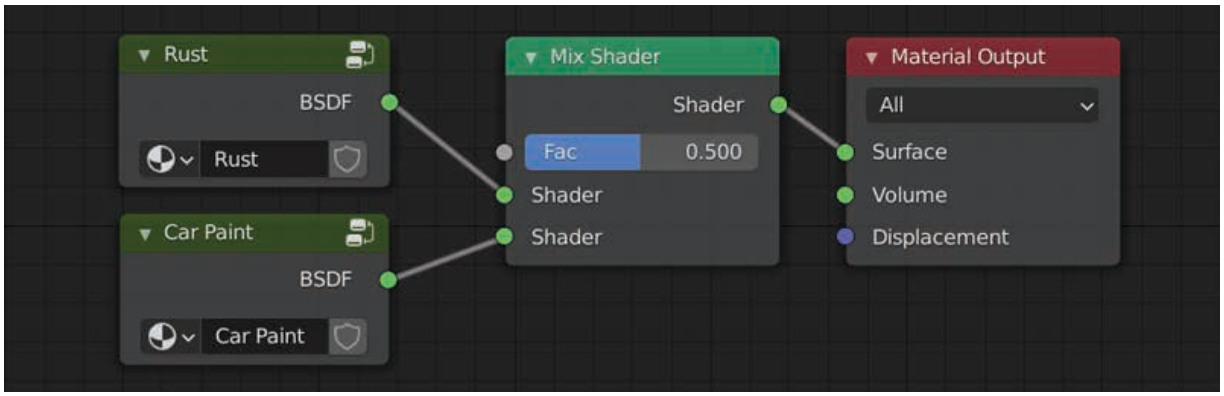
The paint texture doesn't have enough variation for that, so I'll use a different texture (a cropped version of this one from Textures.com) to drive the Roughness.



Using a **ColorRamp node** I can really fine-tune the roughness. White is where Roughness is 100% and black is 0% Roughness.

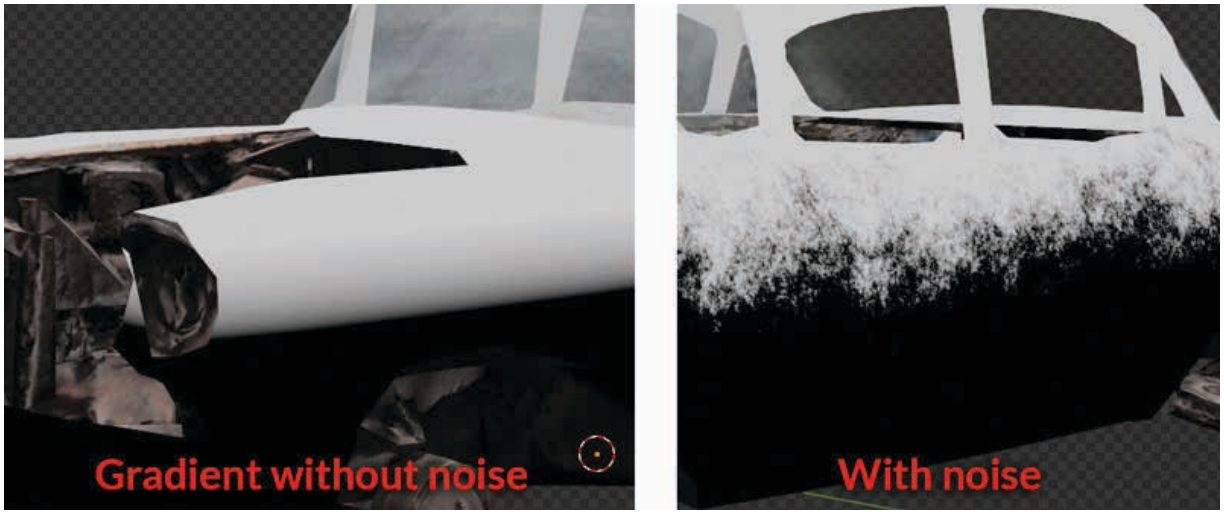
To make even the most shiny parts not too shiny, I changed the black on the ColorRamp to grey.

Now let's add these car paint nodes to a **node group** by selecting them and **pressing Ctrl+G**. As with the rust node group, click on the up arrow icon to close the group. Now you just have these two nodes. **Add a Mix Shader node** in between them.



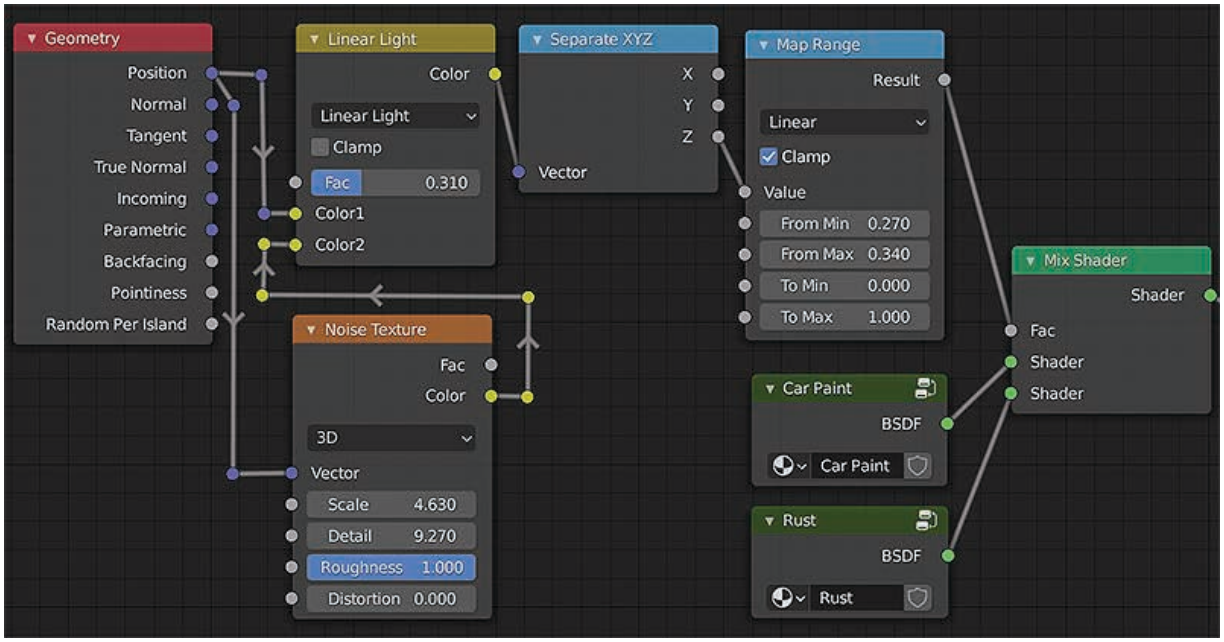
Now we're going to mix these two materials in a more interesting way. To do that we need to create a kind of **black and white mask** that we can plug into the **Fac input** of the **Mix Shader.**

The **Noise Texture node** makes sure the transition between the two materials isn't a smooth gradient.



The **Linear Light node** (actually, a **MixRGB node** set to "Linear Light") gives us a specific way to overlay the materials which works well in this case.

Separate XYZ gives us the ability to choose only one axis, the **Z-axis** in this case, for the gradient to be visible on. **Map Range** is very similar to a ColorRamp node in that it gives us a way to increase the contrast.

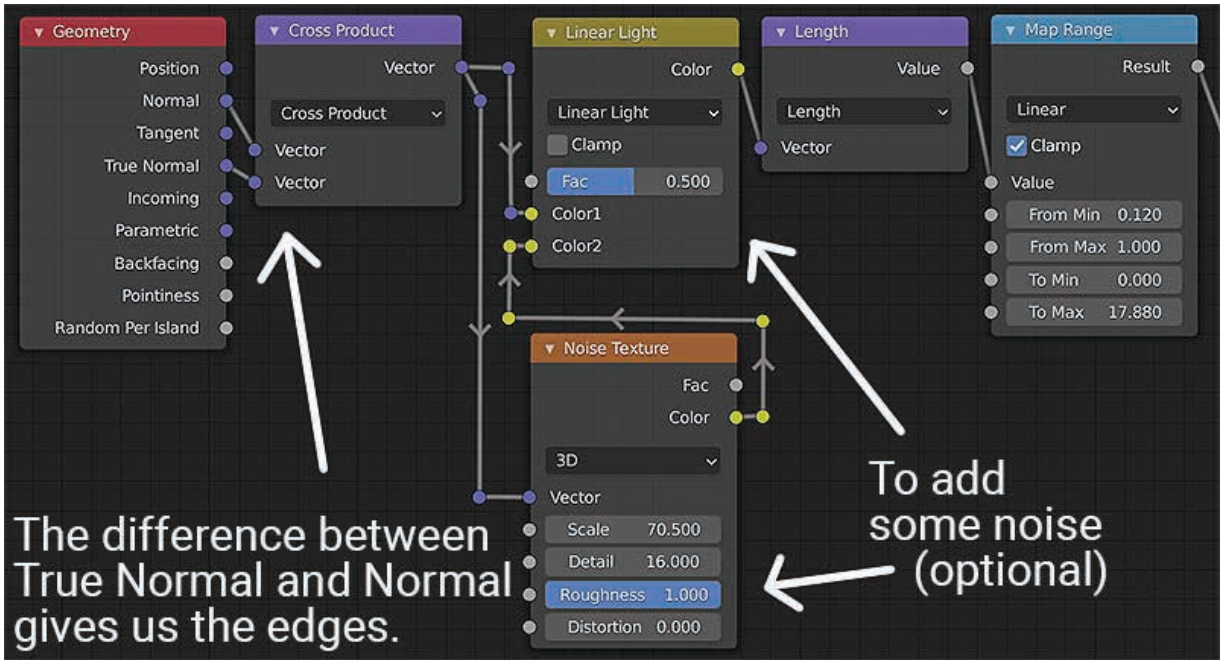


Select these nodes (except the Car Paint and rust node groups and Mix Shader) and add them to a "**Top Mask" node group**.

And below is the result of this mask. You'll probably need to do some value tweaking each time you add this shader to a new object, to get the result you like.

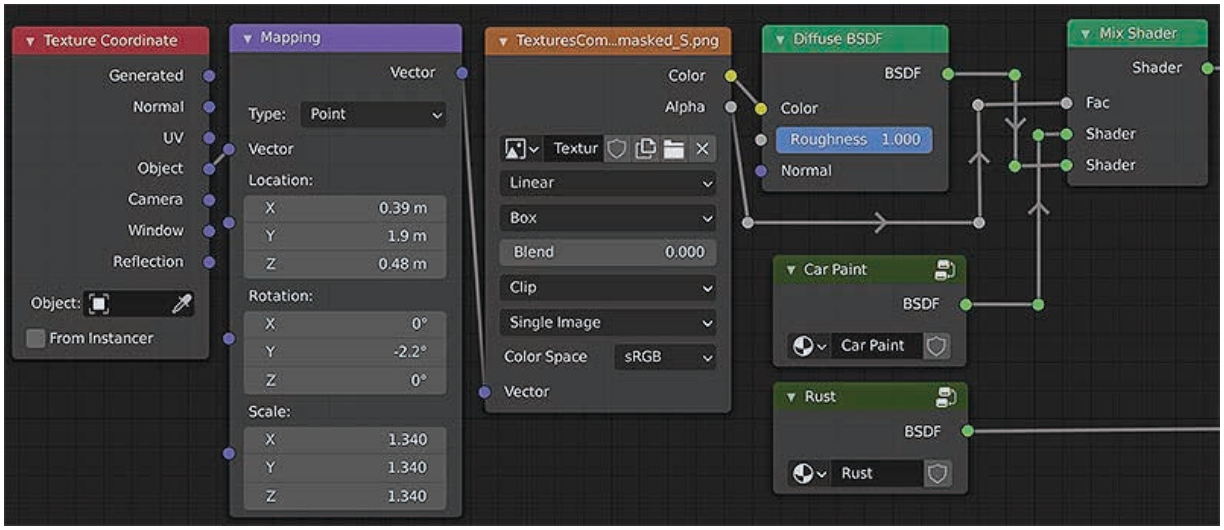
There's another very interesting mask we can create, and that's to add rust to the **edges**. You can easily do this with the "Pointiness" output of the Geometry node, but it only works in Cycles and only in Rendered view. It also works best the more geometry your model has.

So to give you an alternative, here's a version of the edge mask that works entirely in Eevee. The "**Cross Product**" and "**Length**" nodes are both actually **Vector Math nodes**.

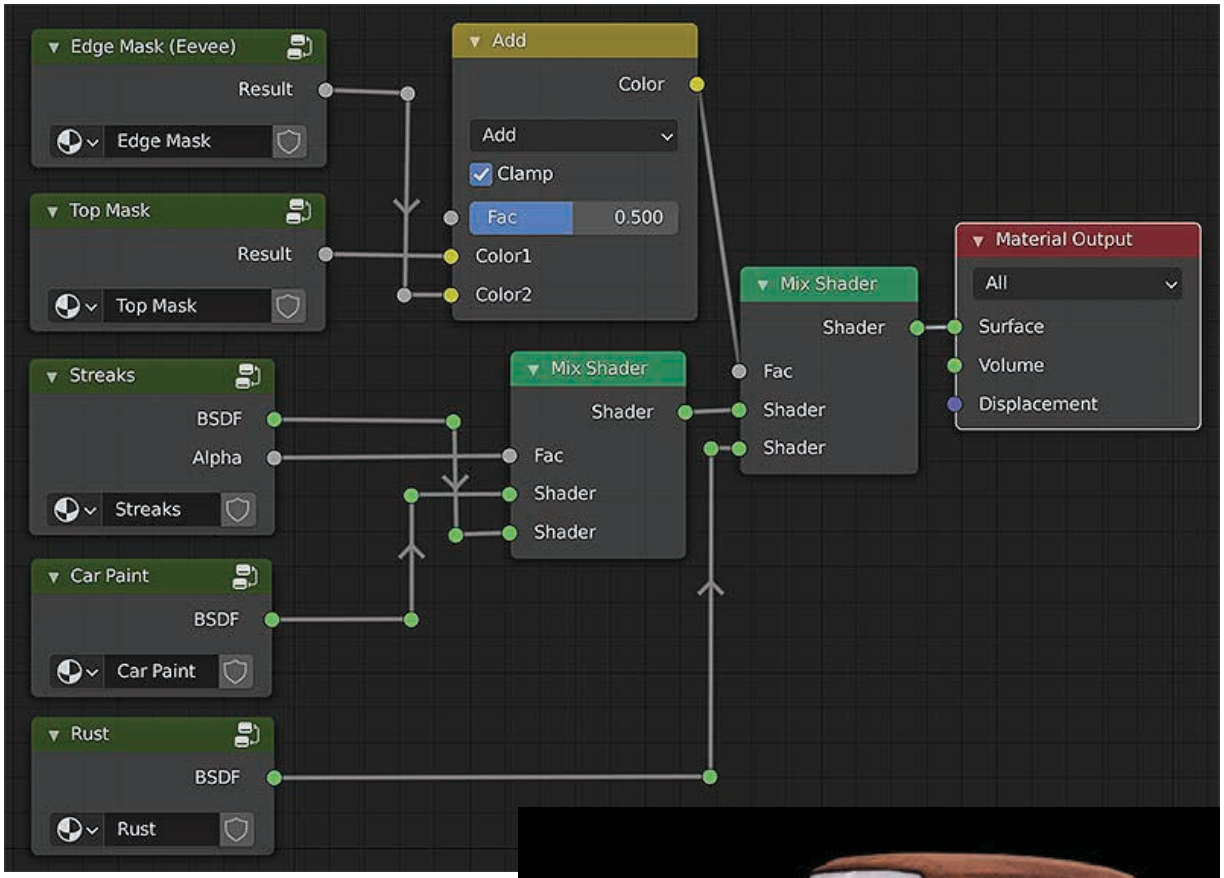


For the leaking on the side I'll use another image texture. Textures.com has an entire "leaking metal" category so it's easy to find a suitable texture. This is the one I used, which conveniently has an **alpha channel** for transparency!

Nothing special here, just another image texture set to **Box Mapping** with a **Diffuse BSDF node** after it (otherwise it'll look weird). You could also use a Principled BSDF node instead.



Here's what the entire node set-up finally looks like:

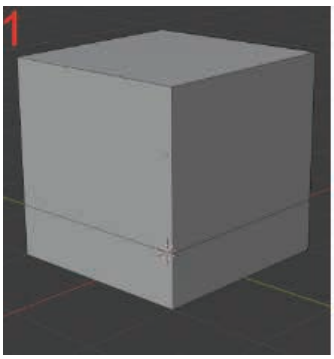


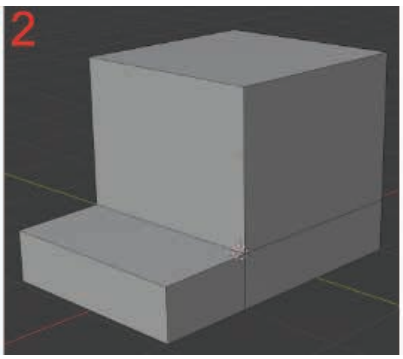
Now the car looks like this. From here on, you can focus on other materials, for example the wheels and inside the engine compartment.

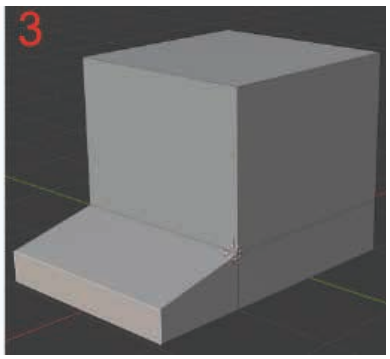
Texturing a cyberpunk computer with photos

In this tutorial we'll look at how to model and add realistic textures to an old "cyberpunk" style computer, using photographs we find online.

The model starts out deliberately simple. You can see the entire process of the initial modeling stage in the three pictures below. The benefit of using photographs during modeling, is that you can still add modeling details that are based on what the textures suggest. ***So to summarize:*** *it's advisable to start with a really simple blocked out model at first, add textures, and then model the details.*



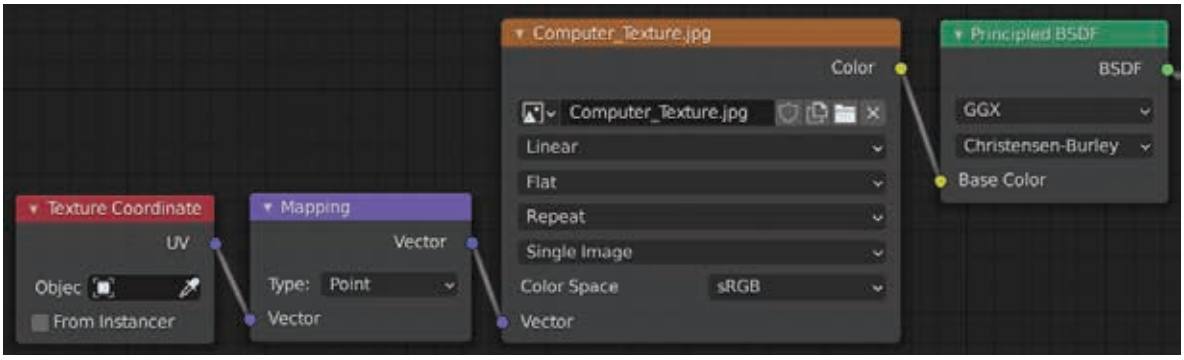




I found some textures on Textures.com that I thought might be partially usable, and made a collage of them in image editing software. I also used that image editing software to create some green computer text that I'll use on the screen. As free image editing software I recommend Krita and Gimp.

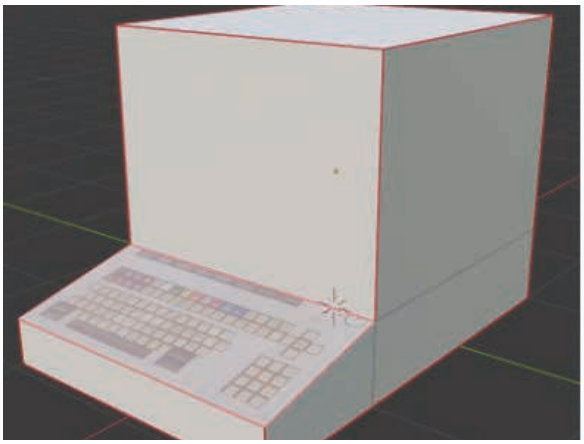
It's possible that I'll only use 20 or 30% of this image, but I can always bake it down to a smaller, more efficient texture later once I've finished texturing the model.

Before we can add a texture to the object, it needs to have a **material**. In this case, I started from the default cube so it already has a material. I like to work on materials in the **Shader Editor** window so I can use nodes. With the **Node Wrangler add-on** enabled in Preferences, I only have to select the **Principled BSDF** and **press Ctrl+T** to add all the image nodes. Add the image in the Image Texture node.

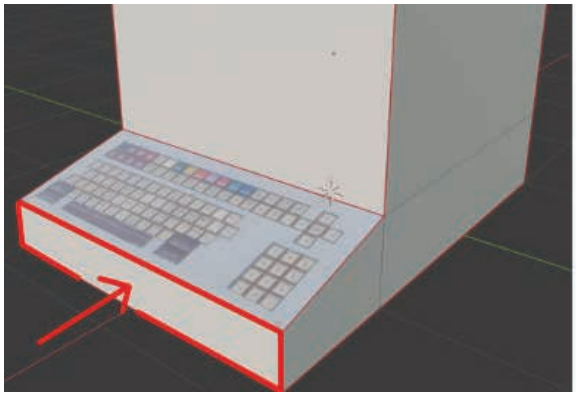


Although you can add seams and unwrap the entire model now, in the case of a very low-poly model like this I find it easier to just select some faces and unwrap them piece by piece.

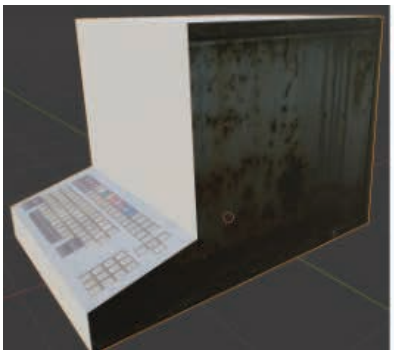
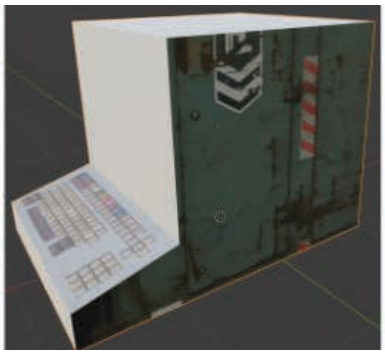
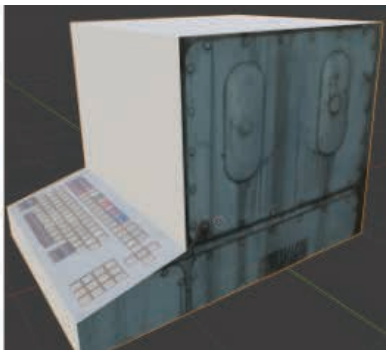
First let's do the keyboard, that seems like an obvious place to start. Select the face that should be the keyboard and **press U**, then choose **Unwrap**. Now you'll see the unwrapped Face in the **UV Editor** window. You can move, scale and rotate it using **G, S and R** as keyboard shortcuts, so that it overlaps the keyboard in the image. **Hold Ctrl** to rotate it in increments.



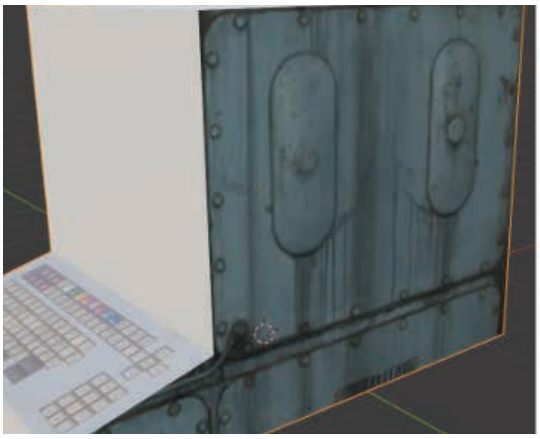
So far so good! However I don't really like that you can see those top rows of darker keys in the image. So I'll make the keyboard part of the model a bit smaller by moving the front face and unwrap the top face of the keyboard again, so it fits the texture better. Keeping the model low-poly at this stage makes it easy to make changes like this.

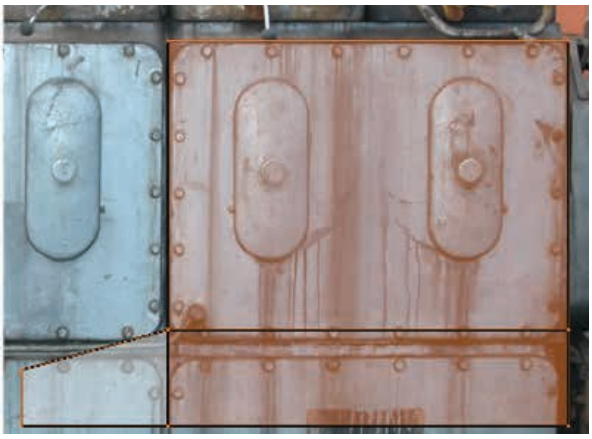


Now the keyboard texture fits well. Let's move on to the next part. This is the fun part for me: unwrapping parts of the model and testing the UV's on different sections of the images I prepared, to see what fits best and hopefully find some lucky accidents. In this case I think the blue metal with bolts works best (image on the right).

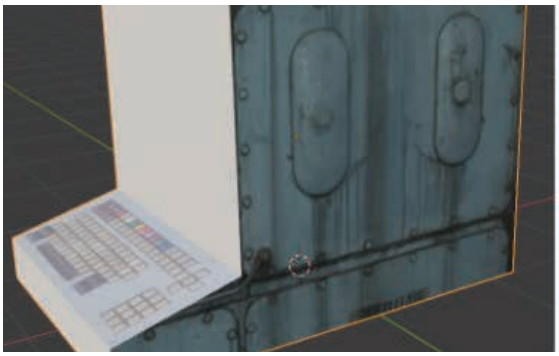




I like this part of the image because of the bolts, which seems to fit nicely on the side of the model.

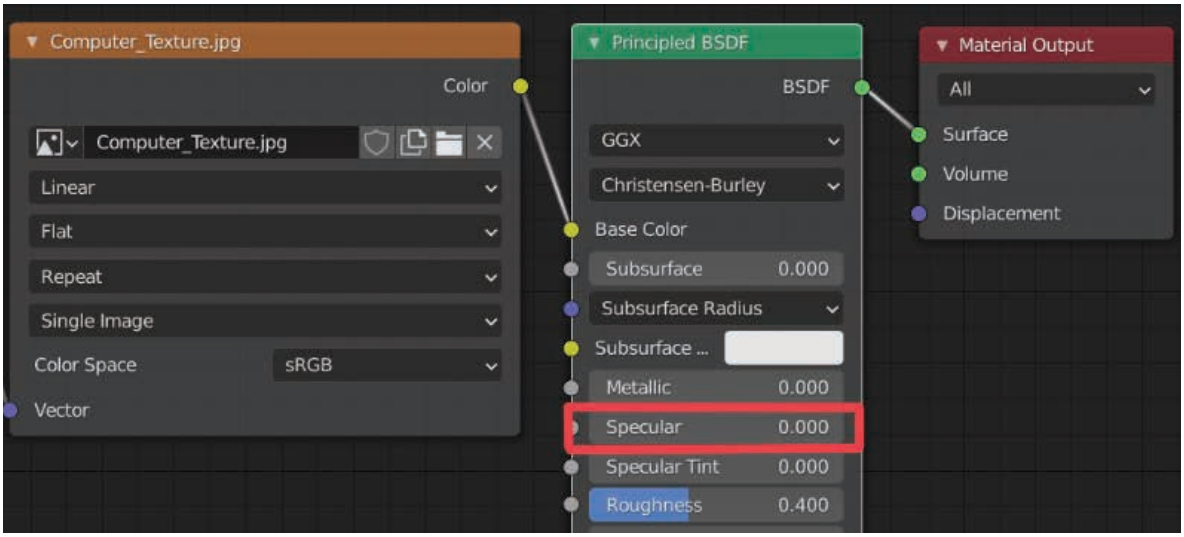




I unwrapped the other side of the computer and used the same part of the image. I had to flip those UV's horizontally by **pressing S+X+[-1]** in order to do this.



Before I move on to the top of the computer, since the specular reflection of the material is making it a bit difficult to see how the texture is working, I set **Specular to 0** in the **Principled BSDF**.

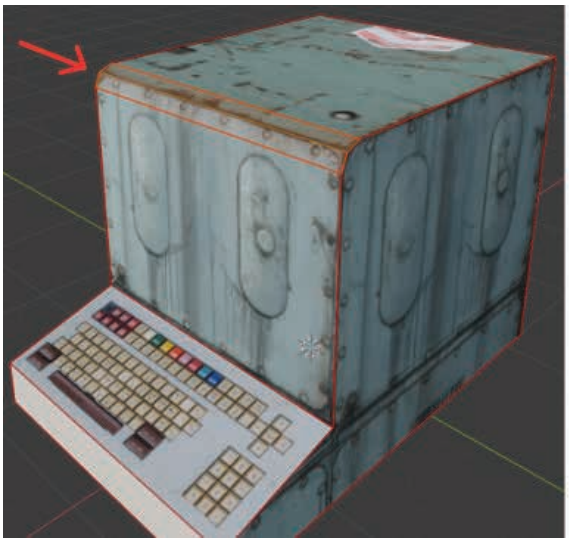


Moving on to the top of the computer, select and unwrap that Face. I found a piece that has a cool stencil with red arrows so I use that.

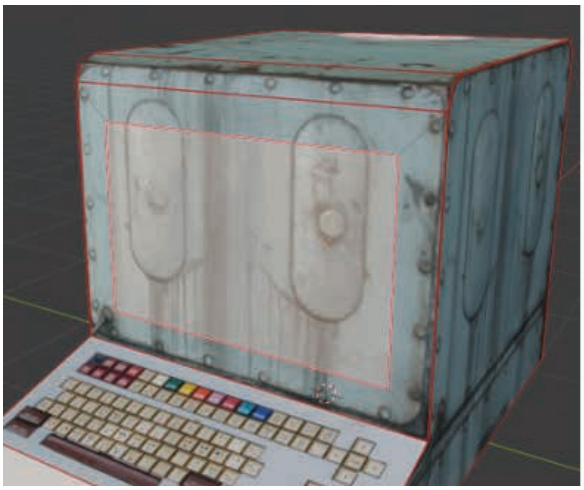
The color is a bit different than the sides, but since we can use the rough rusted edges that are present in the photos as the edges on the model where the different parts of the texture meet, we can get away with it.

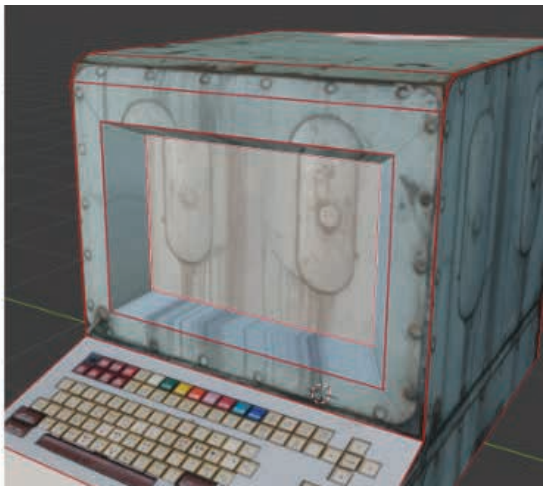
To use the rest of the red arrows stencil on the metal, I unwrap the rear face as well for use with that green metal as well.

Since there's a rounded edge present in the texture, let's try to add that to the model by beveling those edges. Select an edge and **press Ctrl+B** to bevel.

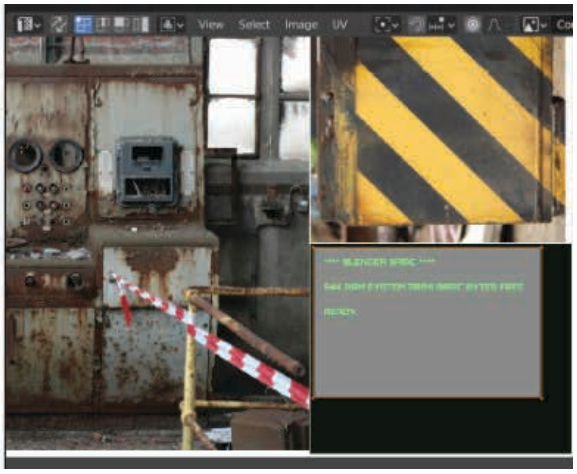


That's looking interesting so far! But it's not a very user friendly computer without a screen. So let's select that front face and **inset it by pressing i**. I then extrude the inset face inwards by **pressing i again and holding Ctrl** (you could also **press e**). Scale it down by **pressing S**. I want to make the screen small, like a vintage computer.

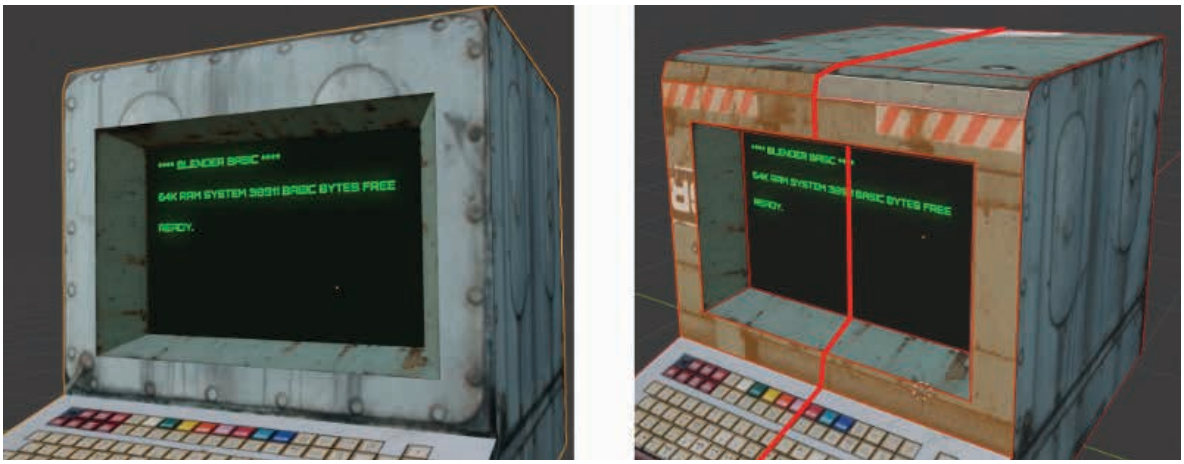




Unwrap that Face that's still selected and move, scale and rotate the UV's so that it overlaps the part of the texture with the screen interface.



After unwrapping the new faces created on the inside and adding some more texture to them, I think the front of the computer doesn't really look good anymore. So I unwrap it again and find some different part of the texture, the green metal, that I like. However it only works for one side of the front, the other looks bad. To solve this, I add another Edge Loop by **pressing Ctrl+R** through the middle of the model.

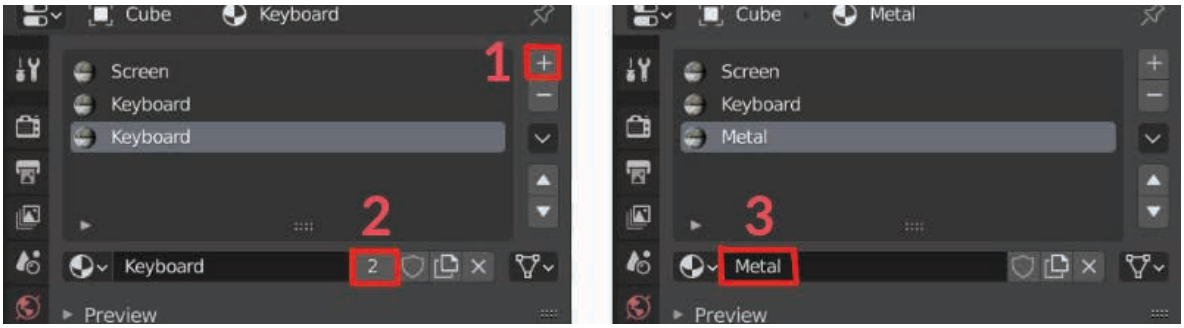


Now I unwrap these faces again and place them on the texture so at least one side looks good. Then I select the UV's of the other side and flip them by **pressing S+X+[-1]**, then move them to overlap the other UV's. Now the texture on the front of the computer is symmetrical and looks nice.

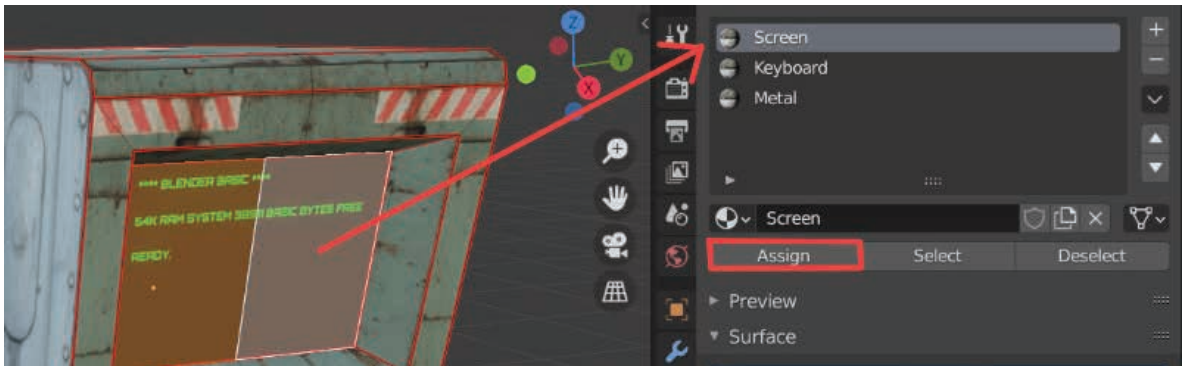
For the remaining faces of the keyboard, I used a part of the keyboard texture without any keys or noticeable detail.

To make the materials look a bit more realistic, I'd like to add some bump and roughness/shininess to it. But first, I'll separate it into 3 materials: The screen, the keyboard and the metal.

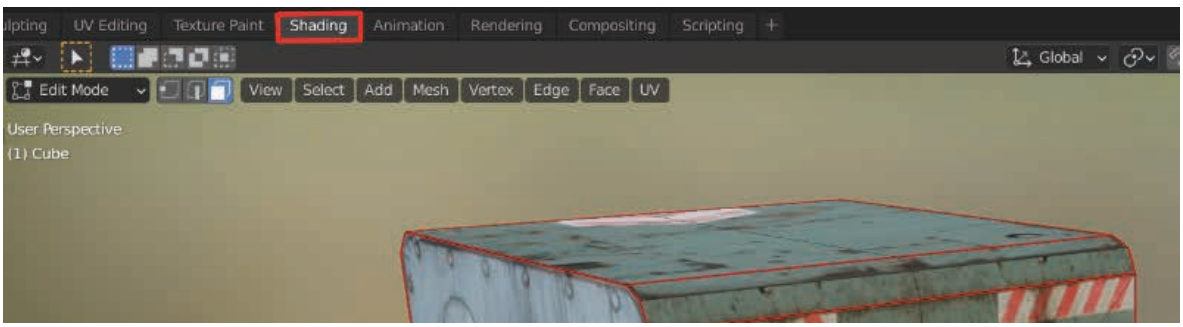
To do this, first add two **new materials** by **clicking on the "+" icon** (1) in the Material slot. Then from the material drop-down menu choose the existing material, **click on the number** next to the name (2) and **change the name** (3). You should name all three materials clearly.



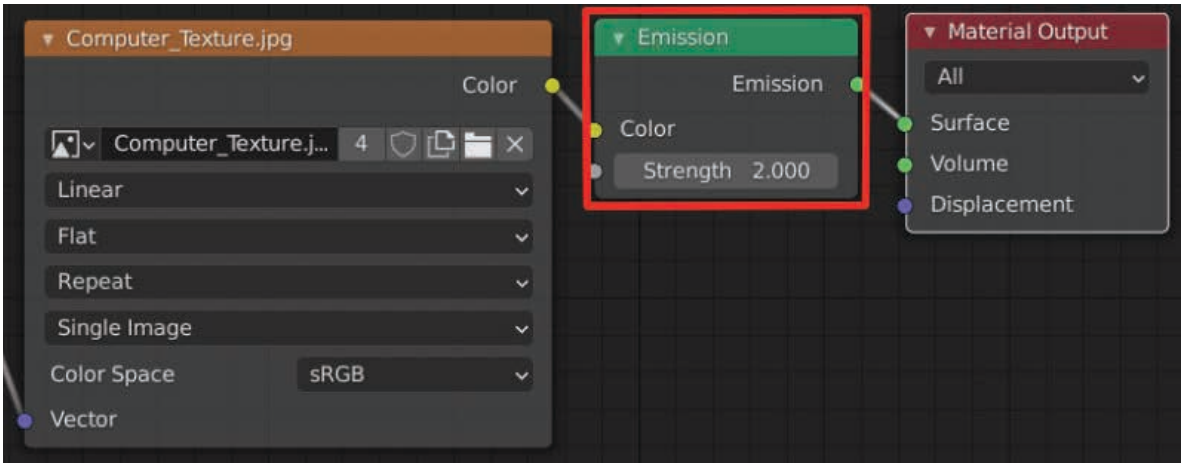
Next, select a material and the corresponding faces on the model, then **click Assign**. Do this for each material. Now each material has the correct faces assigned to it, and you can fine-tune the materials for each part.



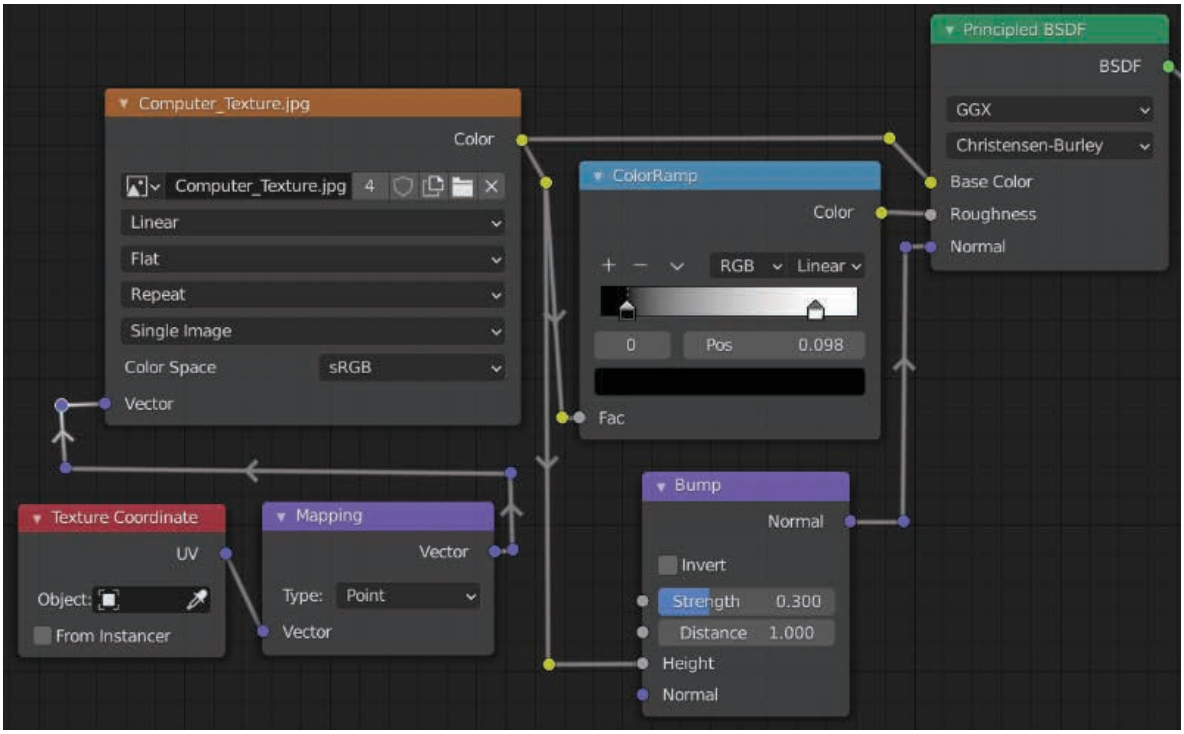
To fine-tune the materials, I'll go to the Shading layout. This way we get some lighting from an HDRI, which is necessary to see how the bump and roughness work later on.



Next I select the **Screen** material and in the **Shader Editor**, remove the Principled BSDF and replace it with an **Emission** node. I increase the **Strength to 2**.



Then to make the **Metal** material more interesting, I added some **Roughness** and **Bump** to it using the same image texture. I added a **ColorRamp** node to control the Roughness better. You could also use a different image texture for the roughness and bump if you want. If your initial texture is a small JPEG that's probably a good idea.



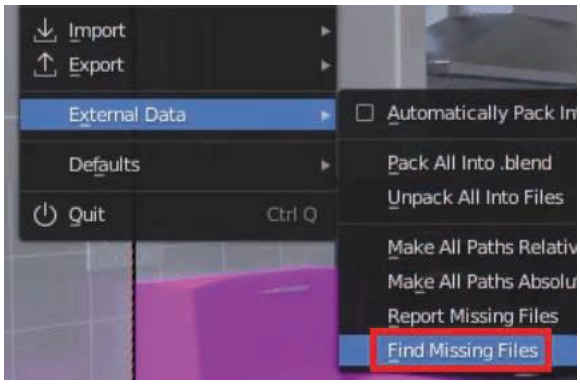
Below you can see the difference without (left) and with roughness/bump (right).

I think the result so far is quite nice for a small amount of work. You can keep adding detail to it of course, for example you could extrude all the keys on the keyboard. But for a low-poly background prop I think it's already good enough. How much detail you want to add depends on how closely it will be seen.

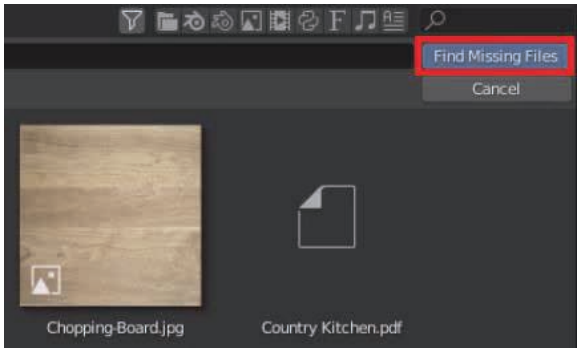
Fixing missing textures

This pink color indicates that Blender can't find the textures for those materials.

Easy to fix: go to File > External Data > **Find Missing Files**.



Browse to the folder which contains the textures and click on **Find Missing Files**.



Now all the materials are using their textures again.

Model by Jay-Artist on Blendswap.com.

One-click texturing with Dream UV

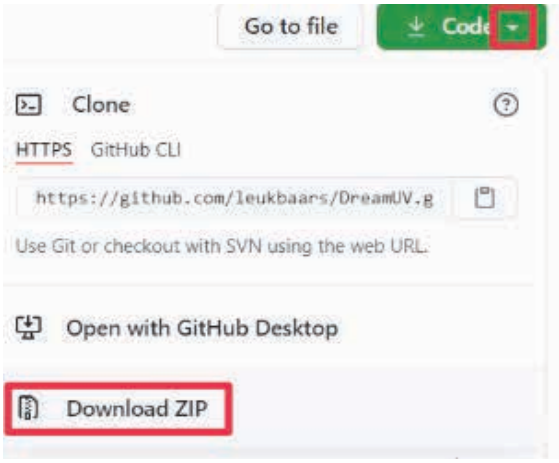
If you're like me, you think UV-unwrapping is a time-consuming uncreative chore that you're always trying to get around somehow. Well, good news! DreamUV is a seemingly magical hack for texturing in - after some intitial setup - literally one click.

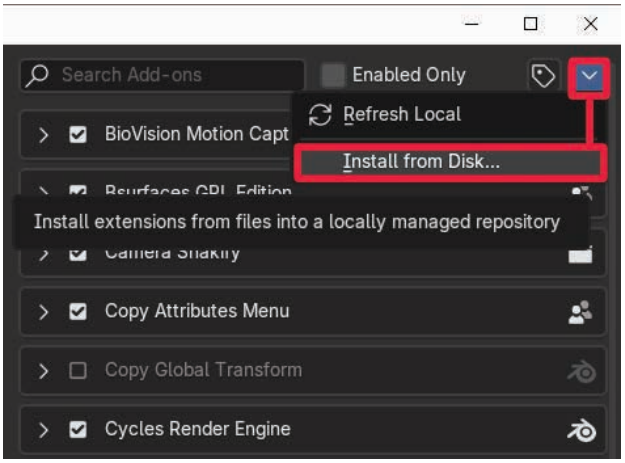
The below model was textured by going into Edit Mode, selecting all faces, and clicking one button. No UV-unwrapping, no marking of edges... is it a dream?

It's not a dream, just a clever and free add-on.

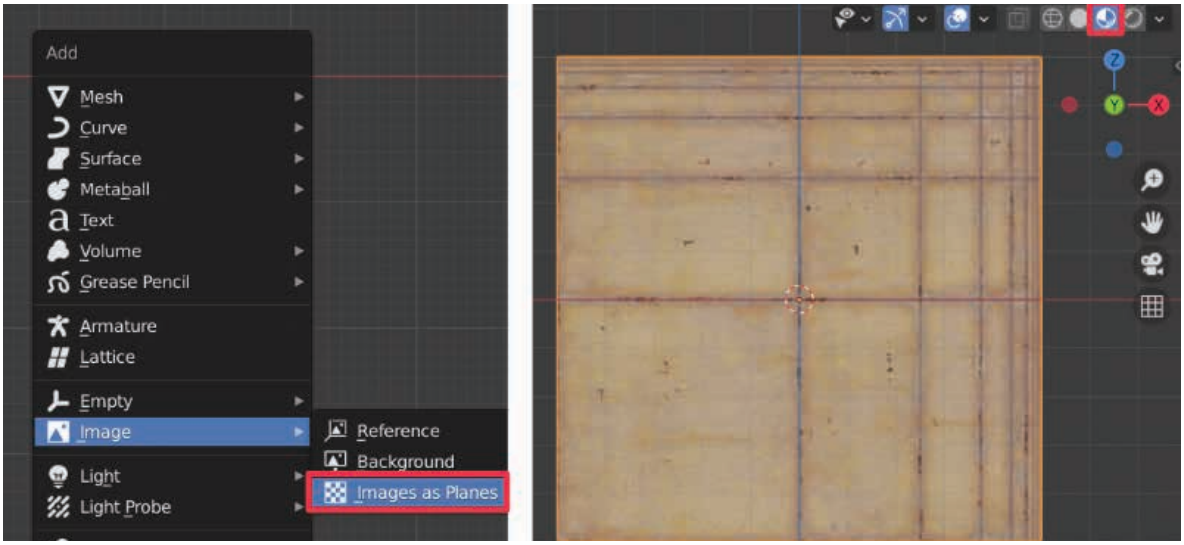
First, you'll need to download the **DreamUV add-on** from this page on Github by clicking drop-down icon on the green icon and choosing "**Download Zip**".

Then from Blender's preferences, choose **Install from Disk** in the Add-on tab. The add-on will be enabled automatically after being installed.



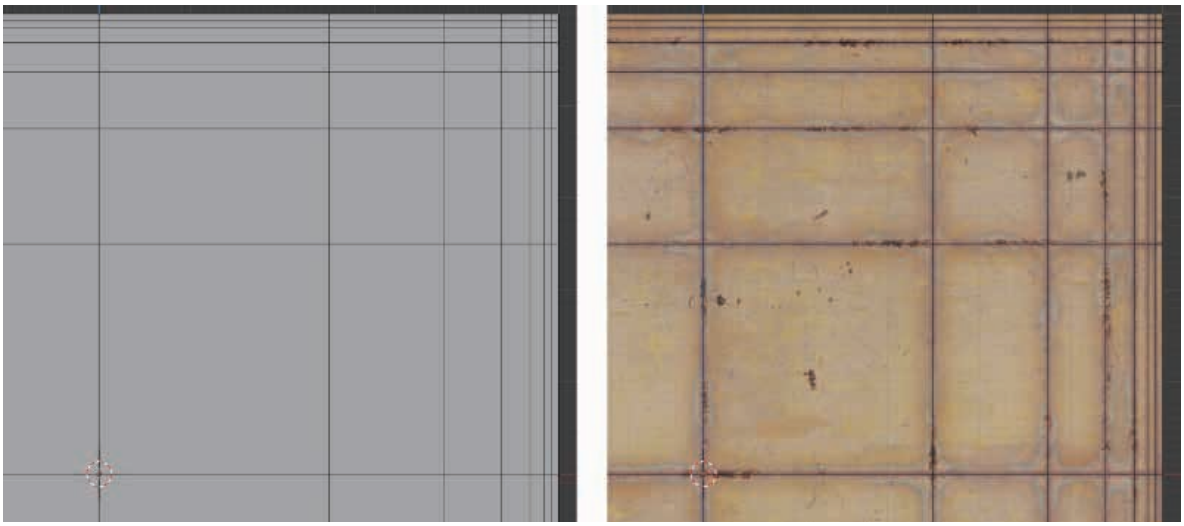


Next you'll need to load a **texture atlas**, which is basically a texture image which has been prepared to have a number of differently shaped and sized visual areas. The author of the add-on, Bram Eulaers, has kindly provided a free sample texture atlas that you can experiment with. Import the image with the **Images as Planes add-on** and enable **Material view**.



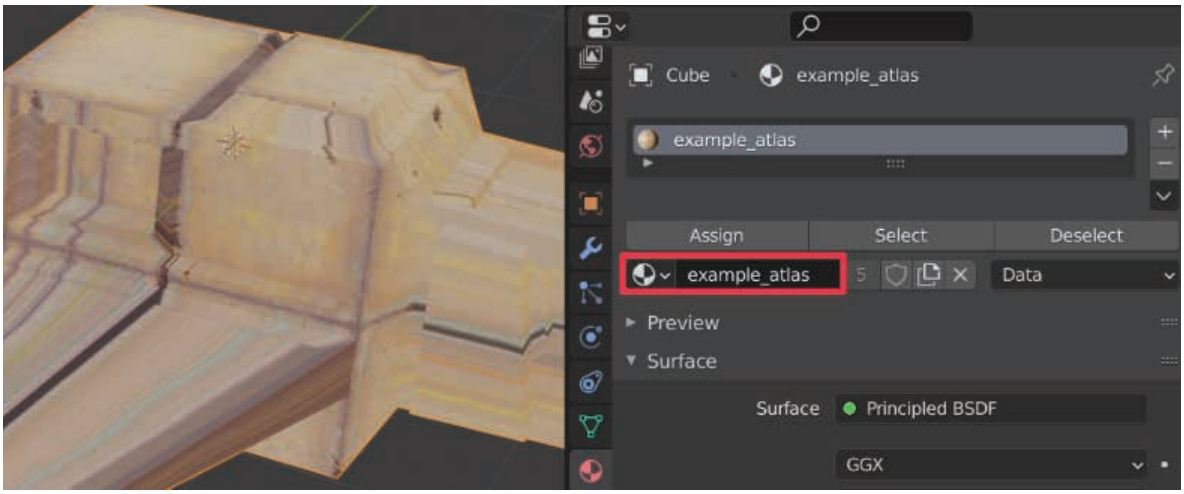
Now you'll have to add edge loops that coincide with where there are visual cues on the texture atlas. This is easy, and you'll only have to do it once. After that it's truly a one-click solution, I promise.

Just add edge loops in Edit Mode by **pressing Ctrl+R**. Start with the middle, then work your way down to the smaller segments. As you'll notice, they line up perfectly with the lines on the texture by default.



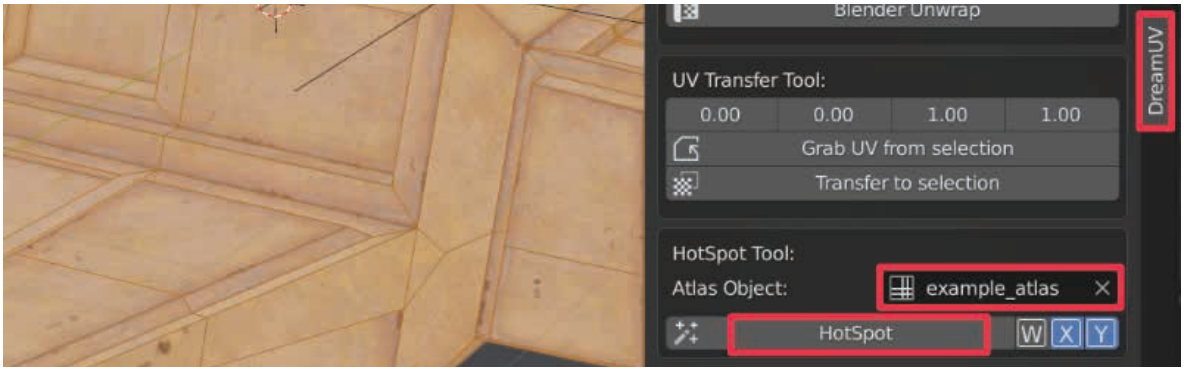
Now to use DreamUV, create or append a model that you want to texture and assign it the same material as the texture atlas. To do this, select the existing material from the drop-down menu in the material tab, with the object selected in Object Mode.

The texture will look very random now, because it's not properly UV-unwrapped. This is only temporary.



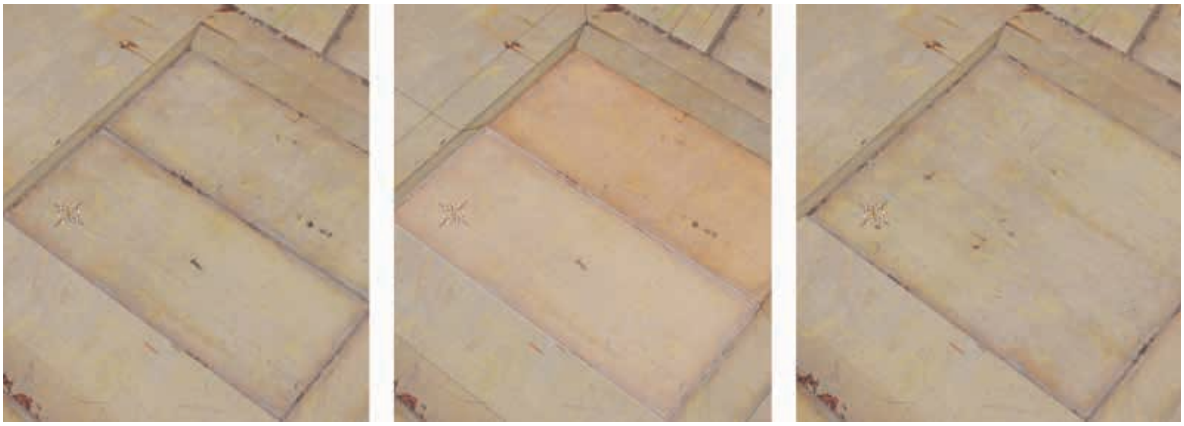
Then in Edit Mode, the DreamUV tab is visible in the Option panel. As the Atlas Object choose the **example atlas** (the plane with the texture).

Select all faces by **pressing A** and click on **Hotspot**. Now DreamUV has automatically assigned those parts of the texture that fit best with that particular Face.

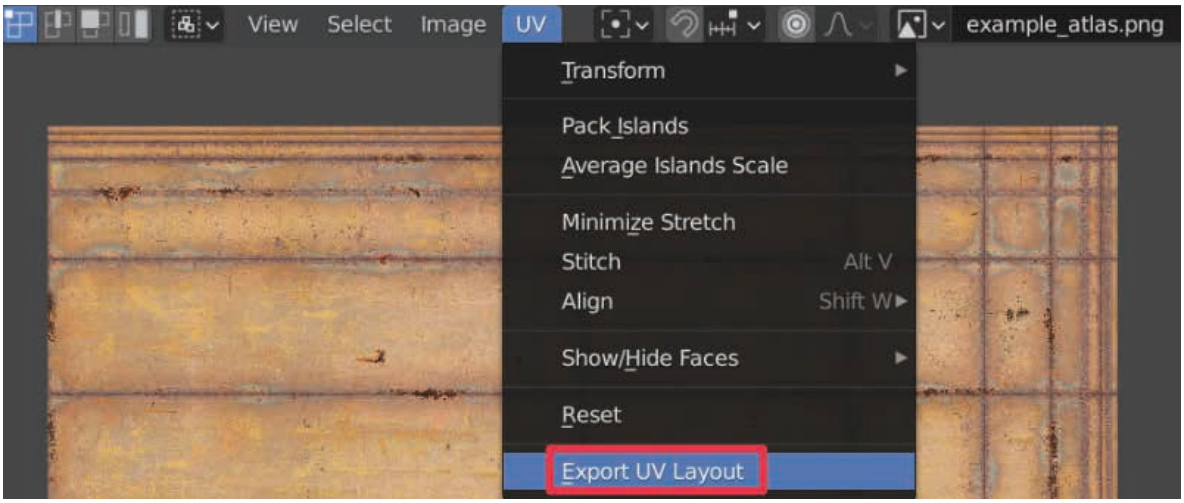


Now my entire model has textures assigned cleverly to its faces.

From now on, you can texture all objects by selecting faces in Edit Mode and clicking on HotSpot. You can also **select faces manually** and **clickHotspot** to change how they are textured.

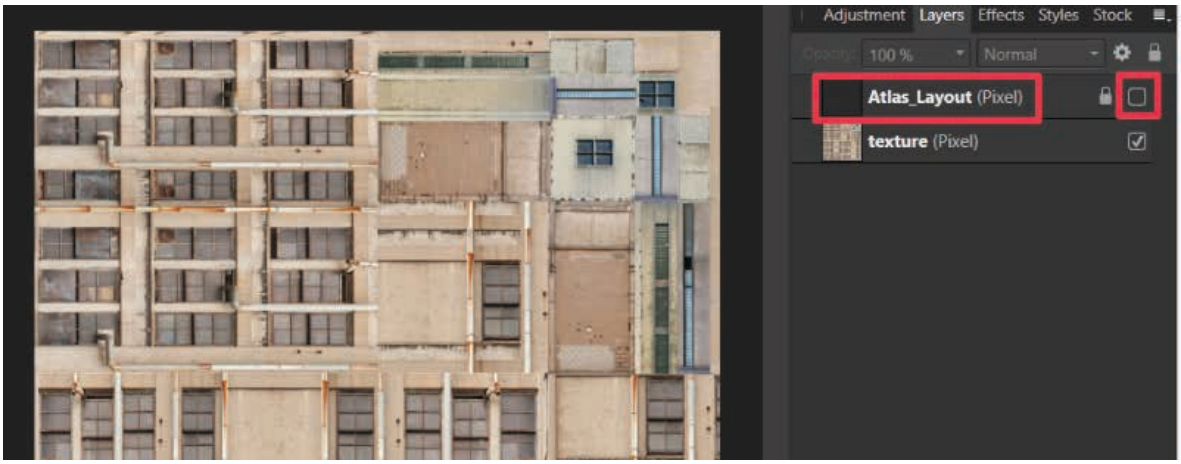


Save the Blend file with the texture atlas so you don't have to redo the initial work of creating the edge loops later. To create a texture atlas yourself, you can use this transparent UV layout I prepared as a **PNG file**, which you can download here. You can also **export the UV layout** of the texture atlas in Blender.

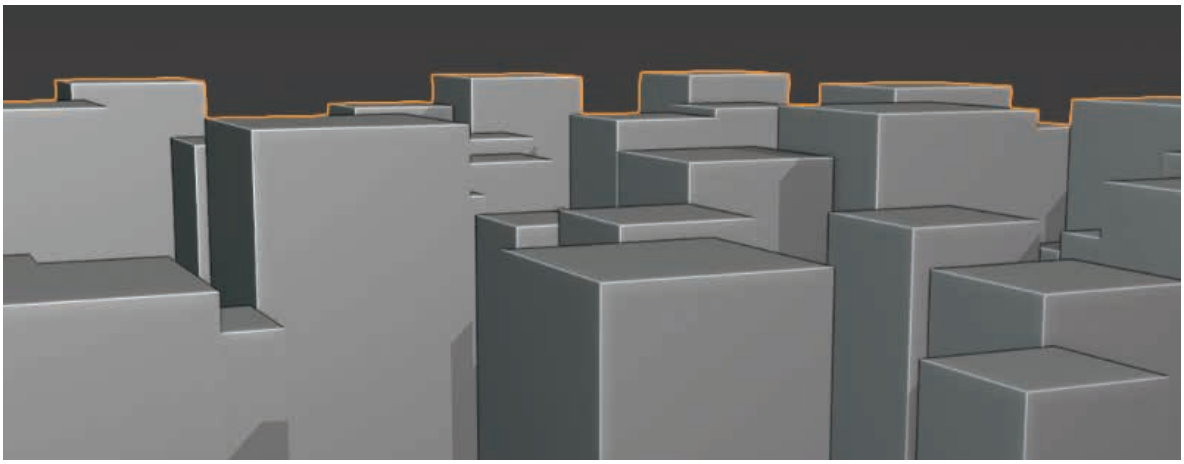


In your favorite image editing software create a new layer underneath the UV layout, then paint a texture or copy/paste selections from photographs, as I did in this example.

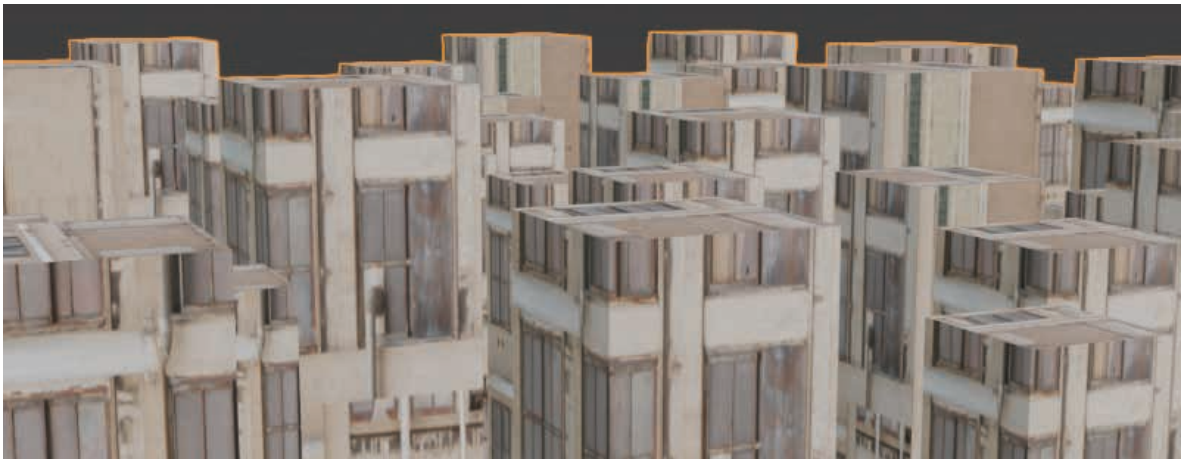
Hide the top layer with the UV layout before exporting the texture *(pictured on the next page is the software Affinity Photo, but you can also do this the same way in Krita or Gimp, both of which are free).*



I've quickly modeled a "city" by scaling the default cube along the Z-axis in Edit Mode, and duplicating it a bunch of times with **Shift+D**, each time adjusting the scale a bit.

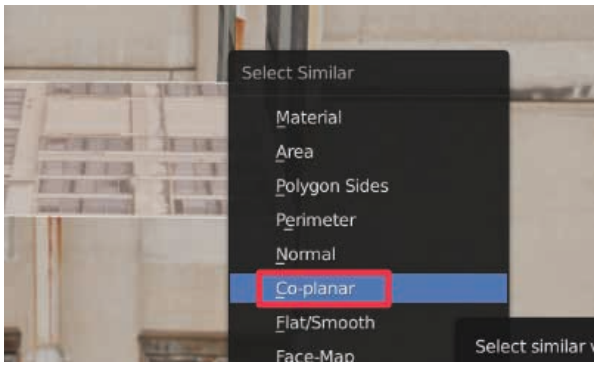


After importing the new texture atlas as a plane and adding the required edge loops as explained earlier, I use its material on the city model. Initially it looks like, well, crap:



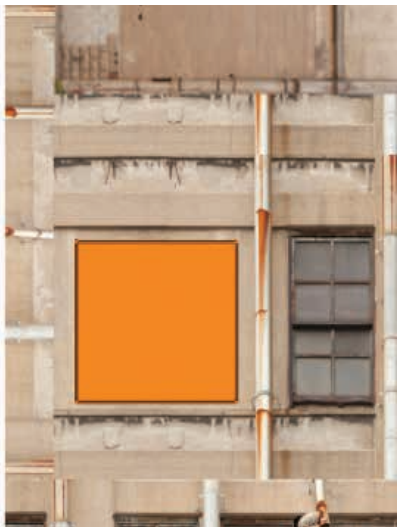
But after selecting all in Edit Mode and **pressing the Hotspot button** in the DreamUV panel, it looks a lot better!

As you can see, the rooftops also have windows, which is not really what you want.



Select one of the roof faces, **press Shift+G** and choose Co-planar to select all of the roof faces.

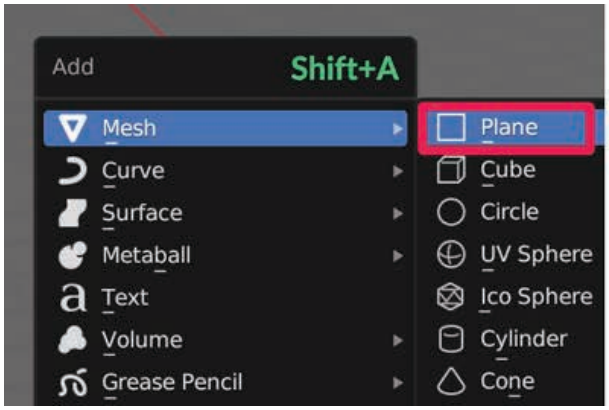
In the UV Editor window **press A** to select all of the roof UV's, then scale and move them to an area of the texture that could serve as the roof.

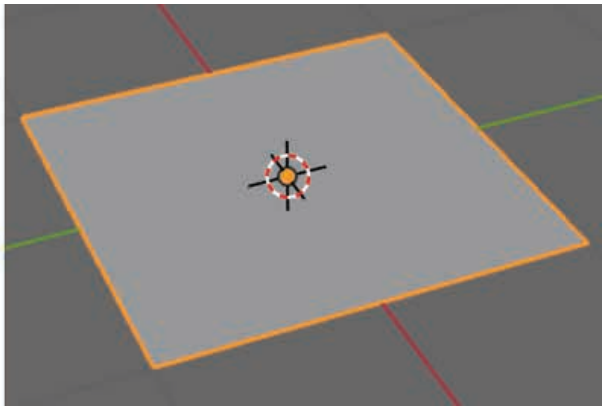


Turn a Photograph into a Tiling Texture

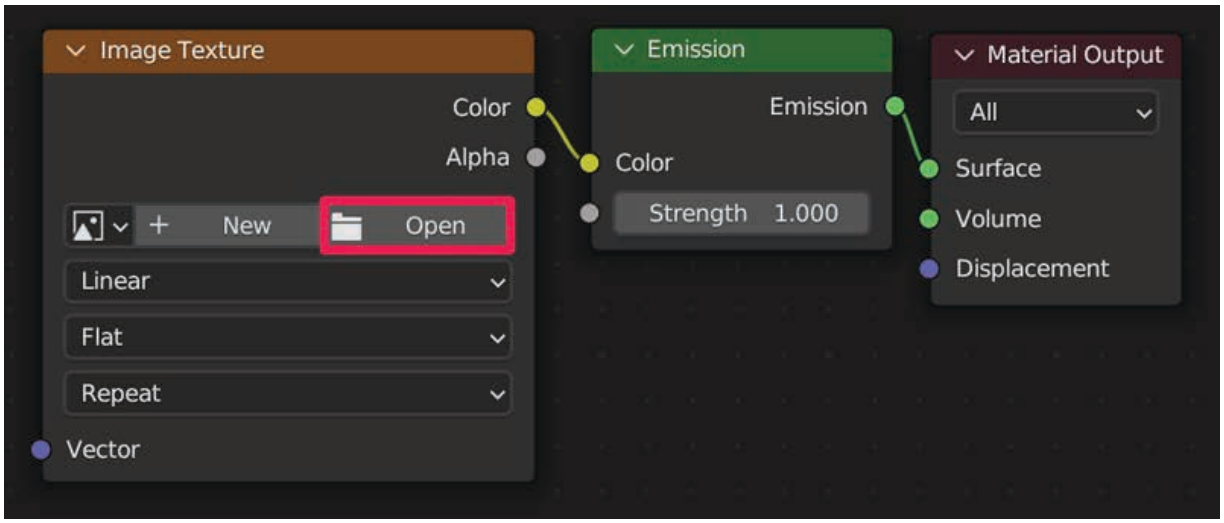
Let's say you have a photograph that you want to use as a seamlessly tiling texture.

Add a **Plane**.

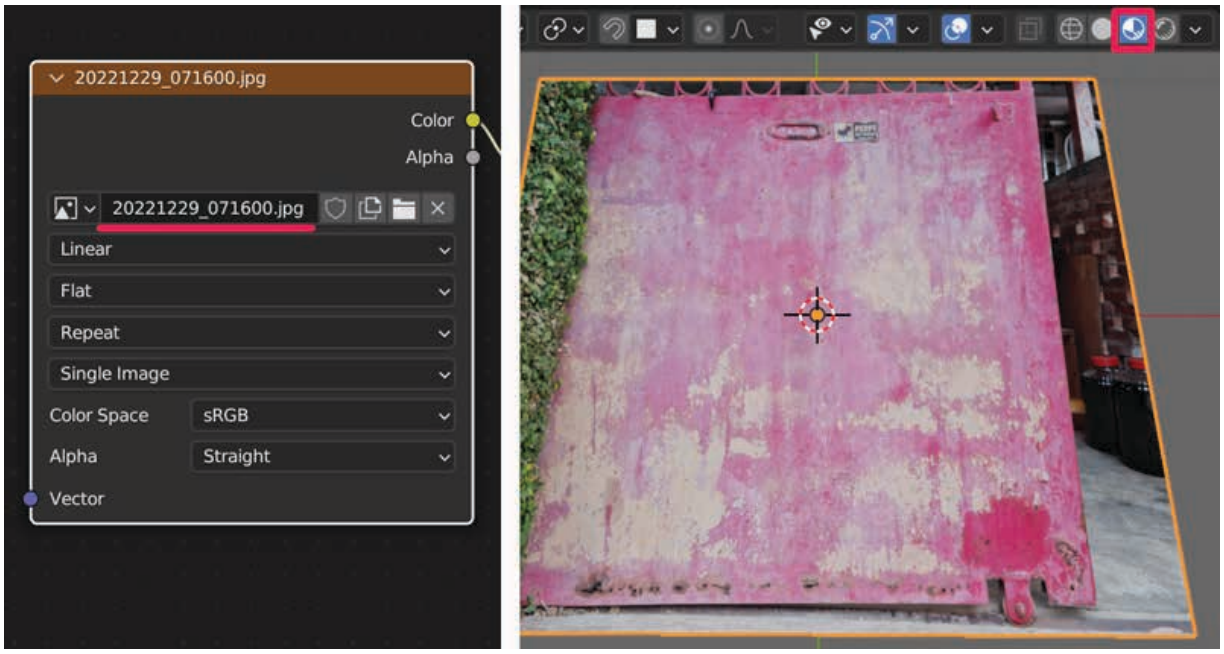




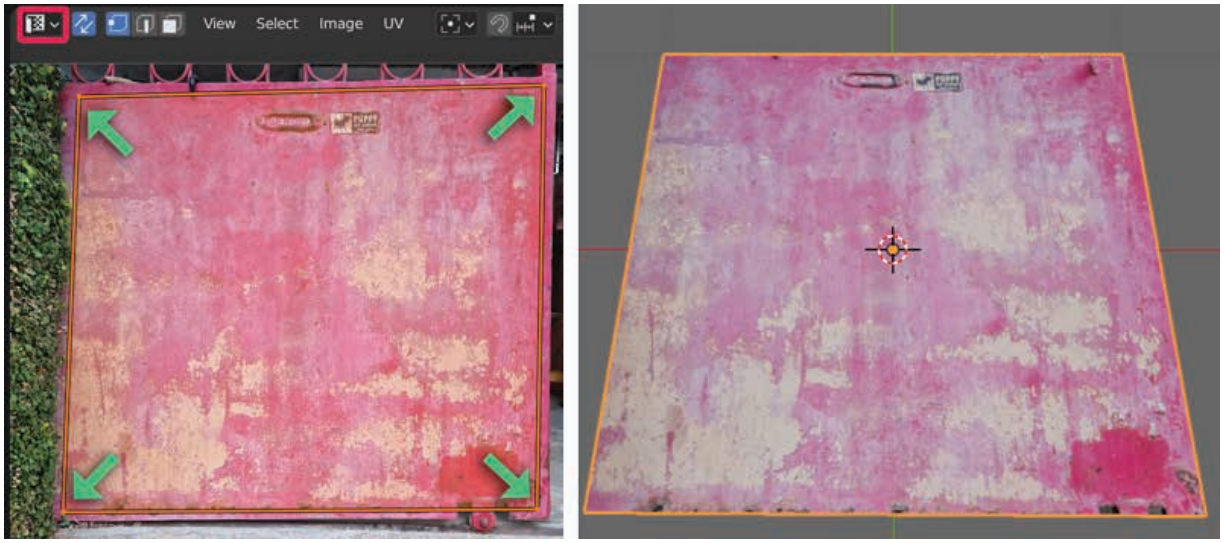
Add a Material to it with an **Emission** shader.



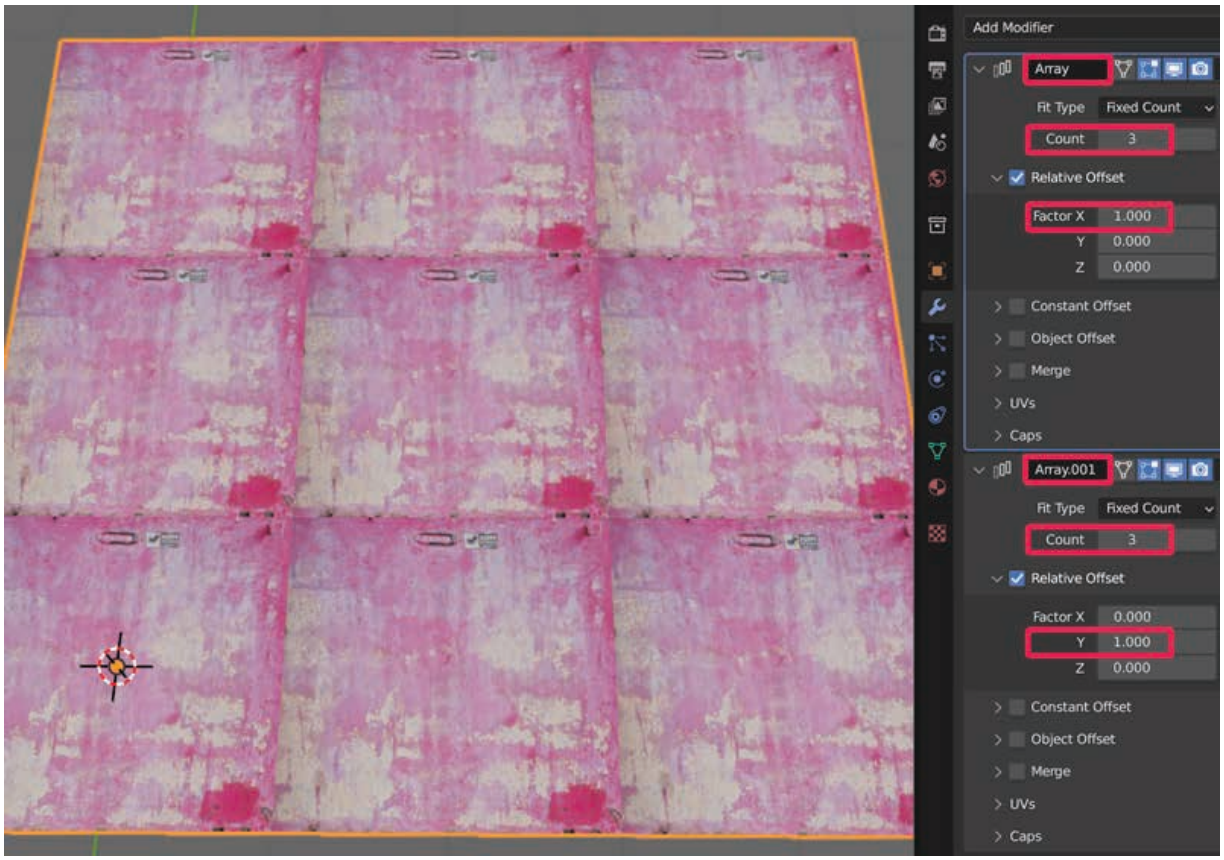
Load the photographic in the texture node. Switch to **Material preview** to see the



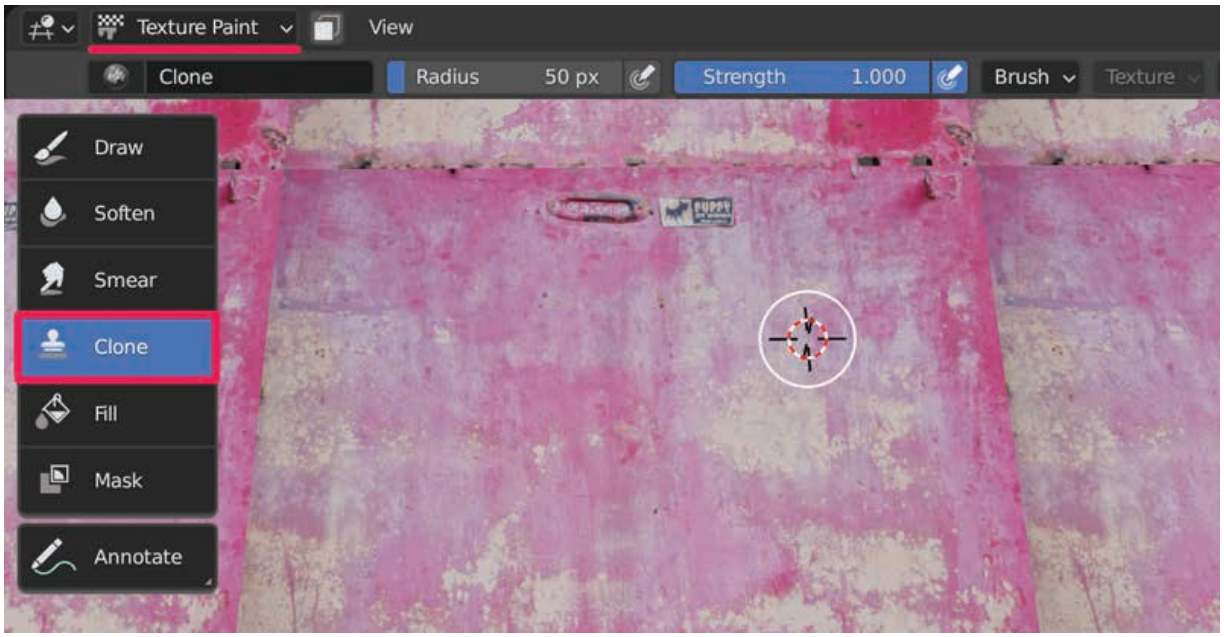
In the UV Editor you can do some perspective correction if necessary. Switch to **Edit Mode** in the viewport and select all vertices to be able to reposition the in the UV window.



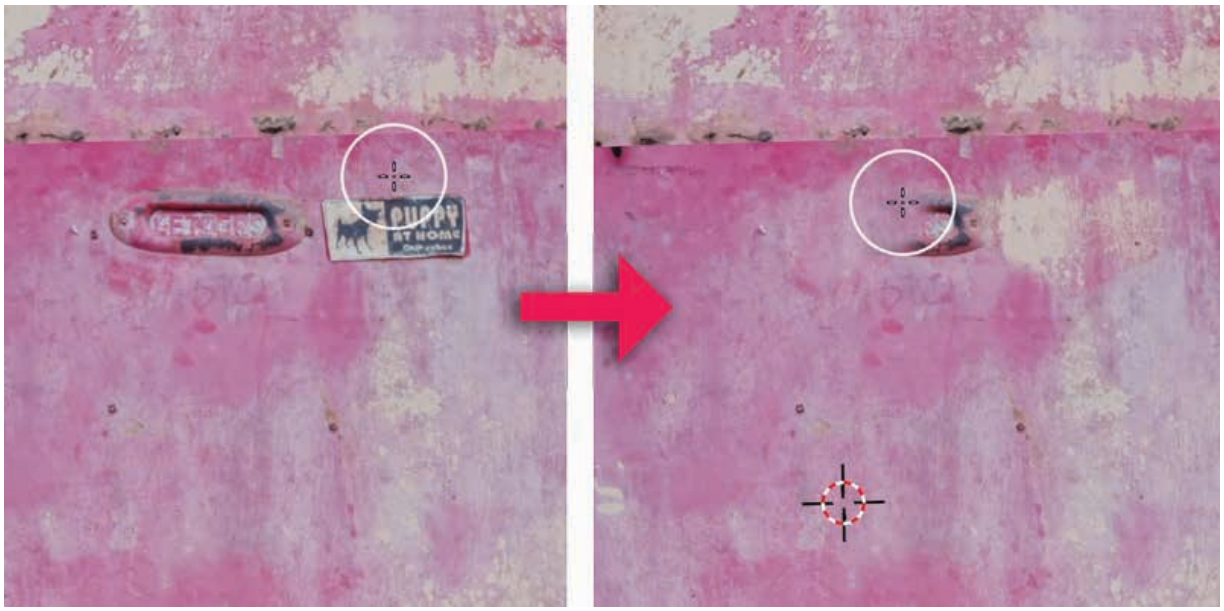
Add an **Array modifier** for the X-axis and another one for the Y-axis and increase the Count to 3 for both of the modifiers.



Then switch to **Texture Paint** mode. With the Clone brush, **hold Shift and right-click** on the area you want to sample from. I recommend working on the center instance in the Array, as you can clearly see all sides on it.

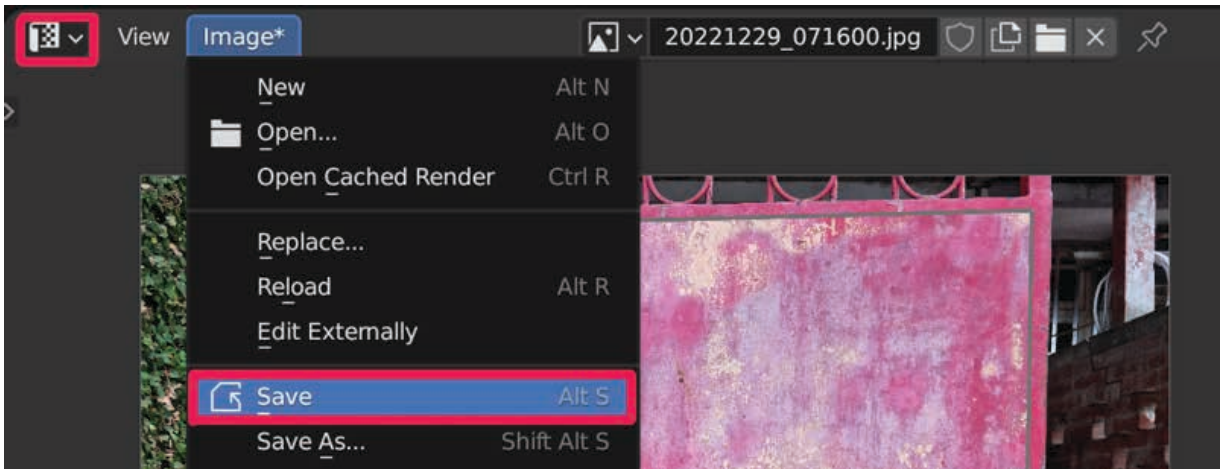


Then clone paint in the parts you need to fill or erase.



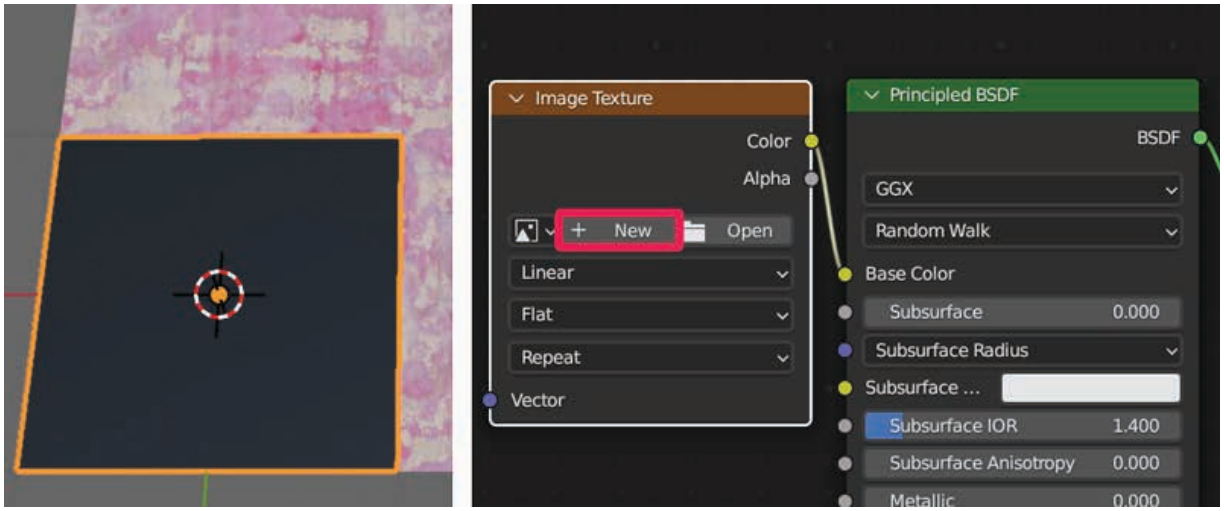
After a while, all the seams are painted over with cloned parts, and unwanted details are removed.

In the UV Editor window, save the changes to the original texture (make sure you have backed up the original first!). If you bake without saving this image, it will use the original texture to bake from, without using the clone paint work you did.

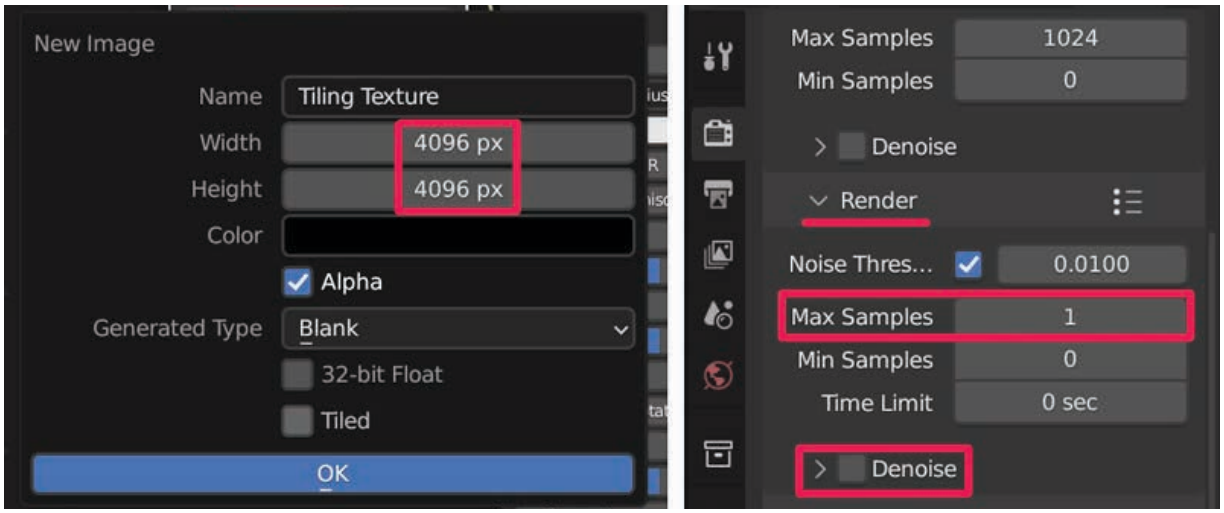


Reset the 3D cursor to World origin with **Shift+S** > Cursor to World Origin.

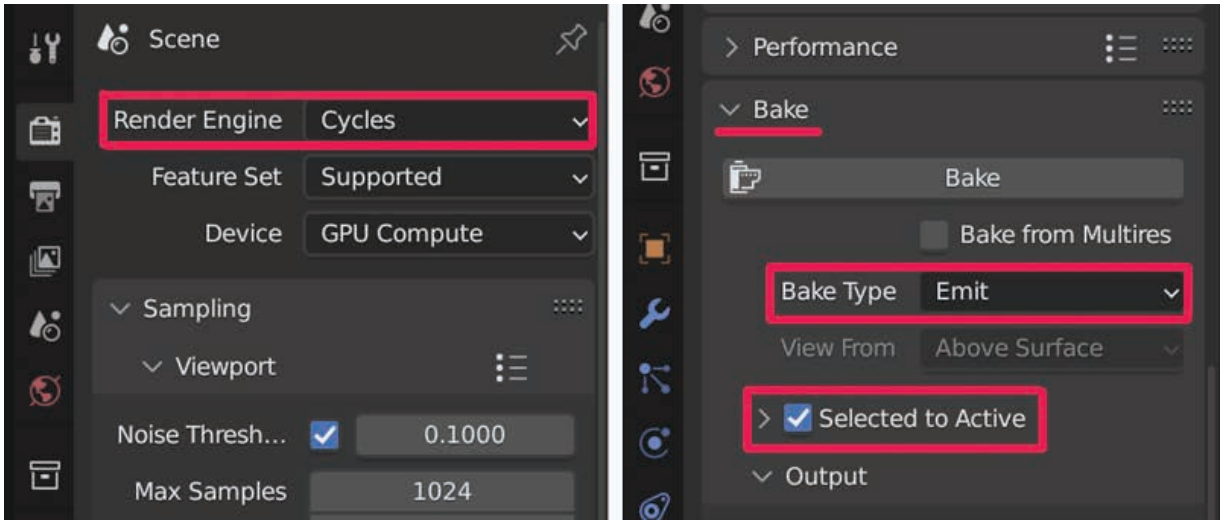
In Object Mode, add a new Plane with a material and image texture nodes. Give it a new blank texture.



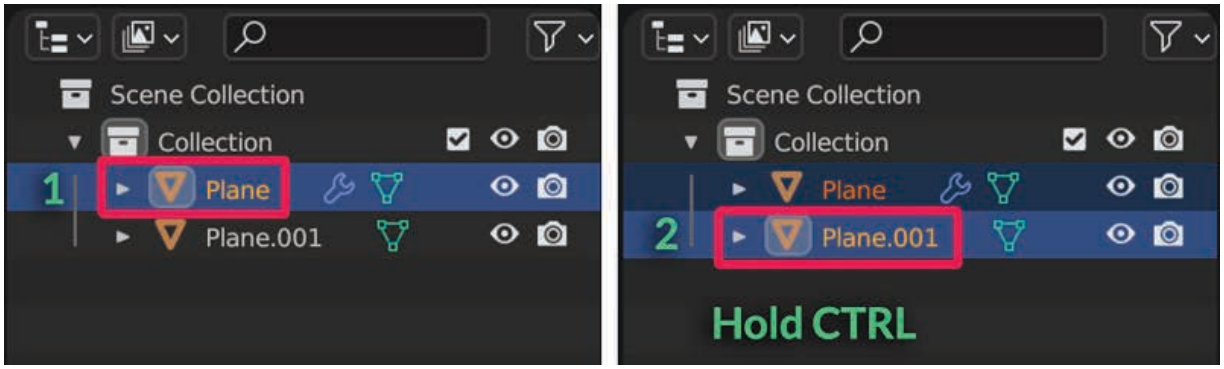
Make sure the new texture is not too low resolution - I recommend 4K. You may want to reduce the Samples before baking, as you're baking from an Emission shader you just need 1 sample. Turn off Denoising to save even more time.



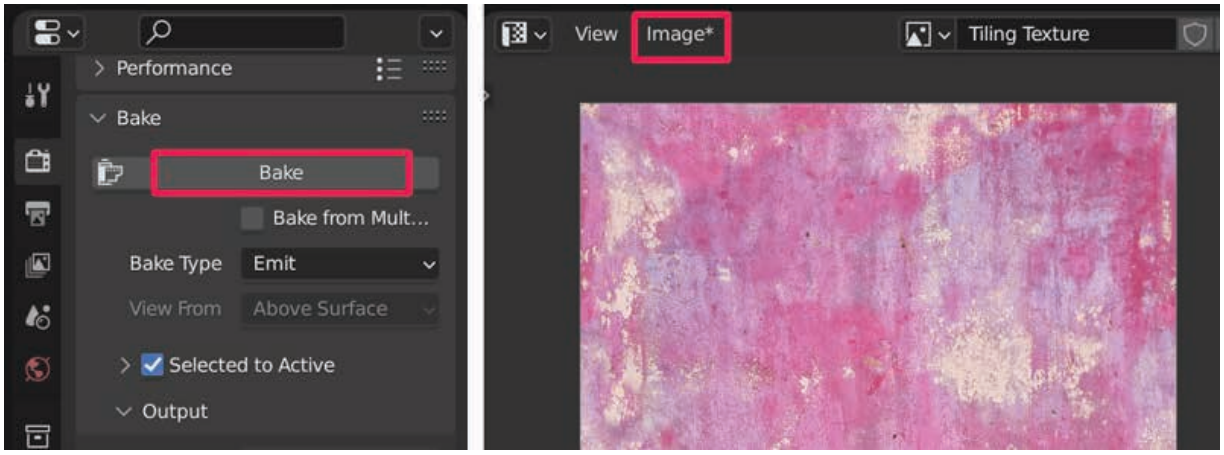
Switch to Cycles and scroll down to the Bake options. As Bake Type choose **Emit**. Enable **Selected to Active**.



Hold Ctrl (not Shift!) and in the Outliner, select the old Plane and then the new Plane so that the new Plane is "Active" (yellow instead of orange). Very important to get this right, otherwise it won't work.



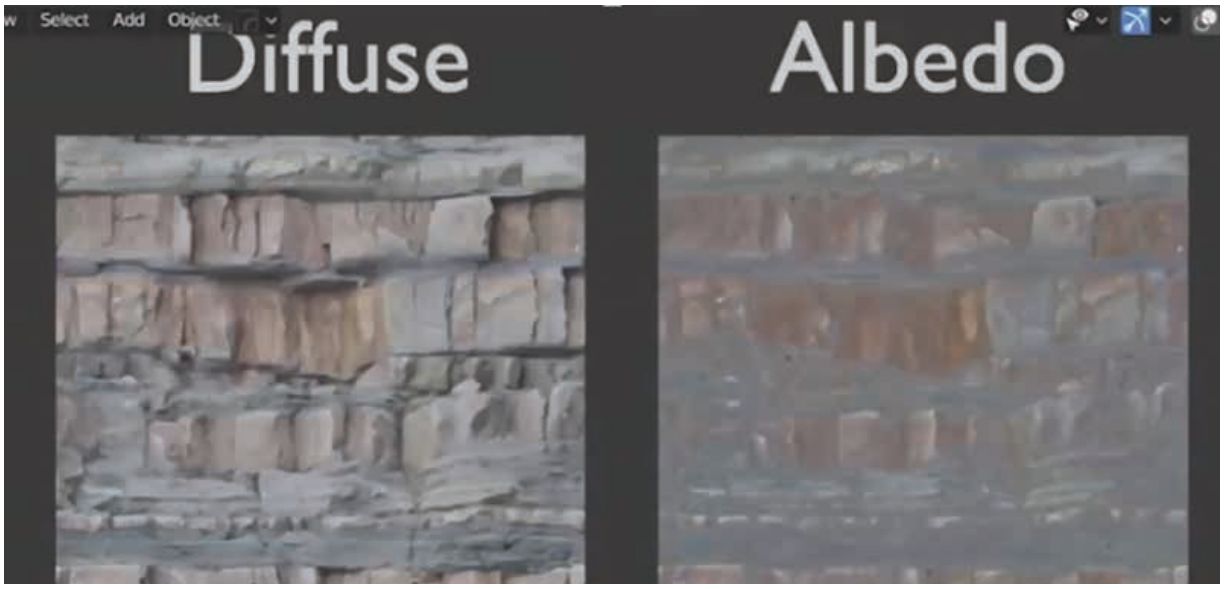
Click on **Bake**. Wait a moment while Blender does its thing (it's not very fast at baking). Save the resulting tiling texture.



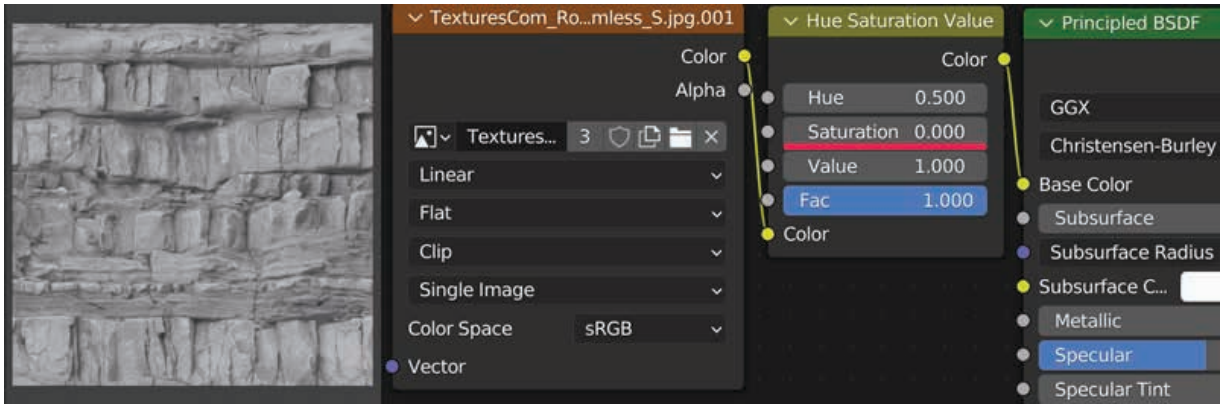
Photographic textures always look great, in my opinion! It's fun to keep them in mind when you're traveling or just having a walk outside. The below materials were made from photographs I took with my smartphone during a walk. They are all painted and weathered metal doors in Taiwan.

Create an Albedo texture in Blender

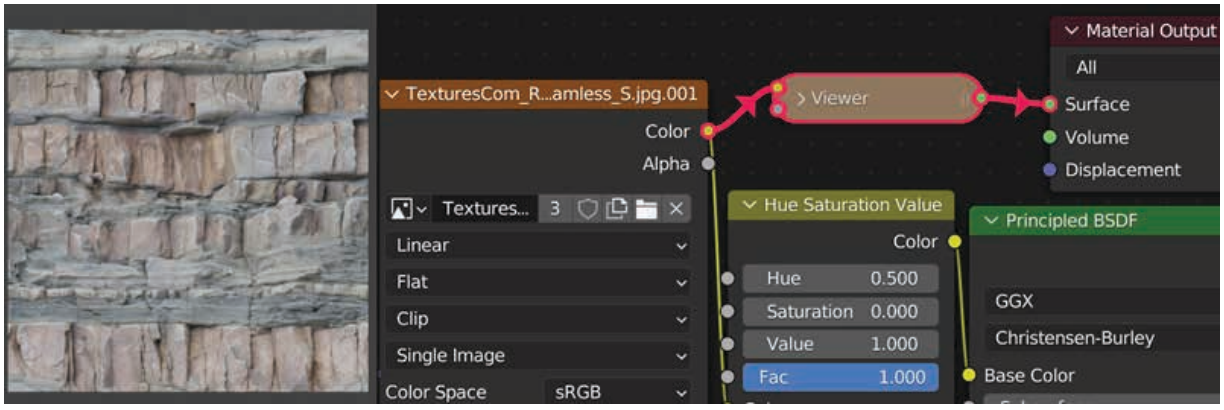
An **Albedo** map is basically a version of the **Diffuse** map but with the real-life baked in shadows removed as much as possible. When you buy a texture set online, this kind of map should be included. But if you make your own texture, from 3d scanning for example, you can also make your own albedo. To download this tutorial Blend file, go here.



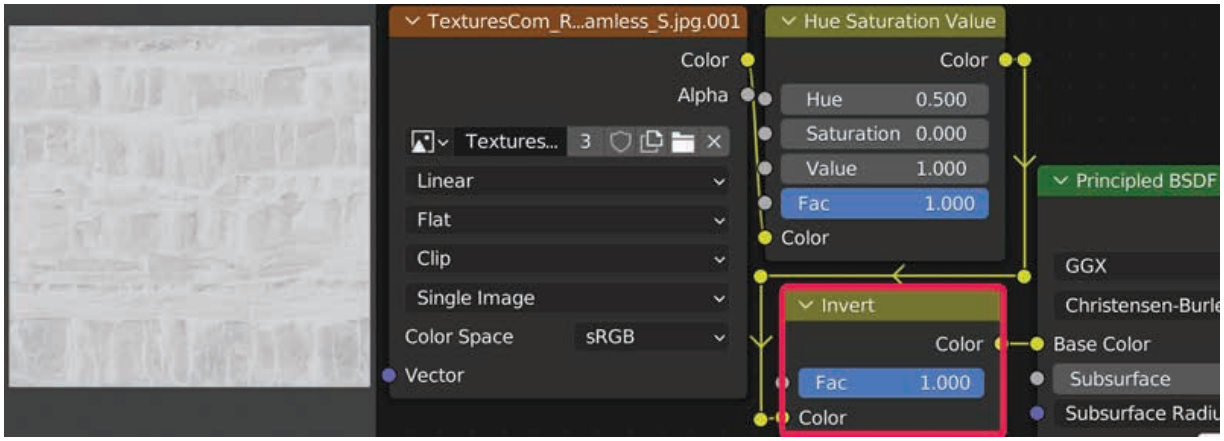
First get your diffuse map and add a **Hue Saturation node** and plug the color output of the texture into that. Reduce the **Saturation** to zero.



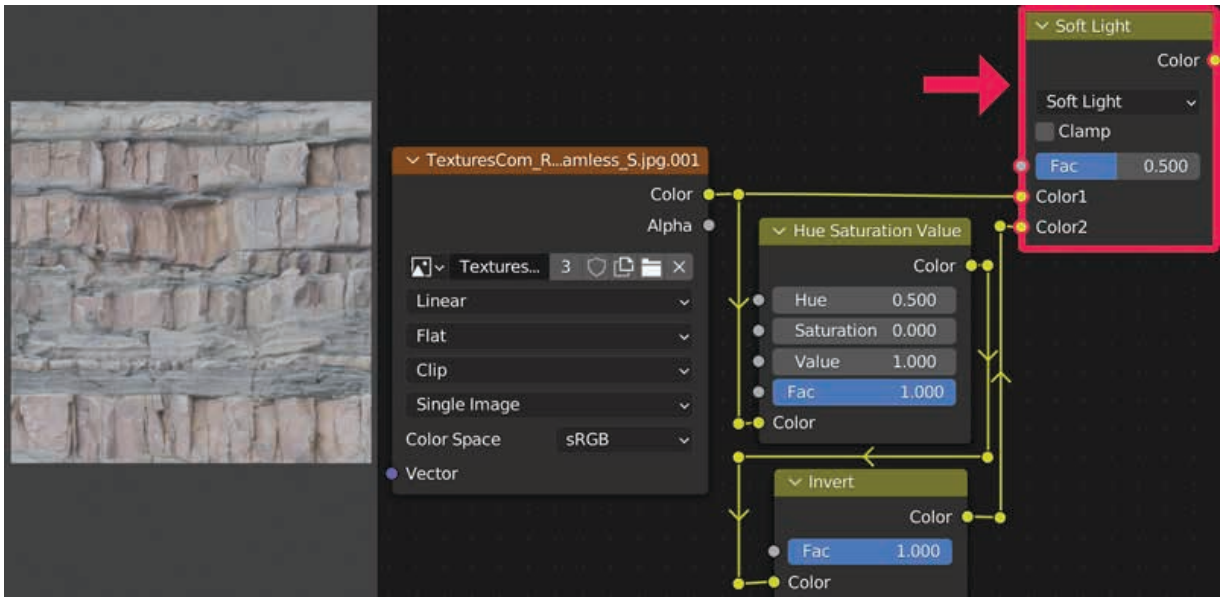
If you have **Node Wrangler** add-on enabled, you can always **press Shift+Ctrl+LMB** to preview the selected node.



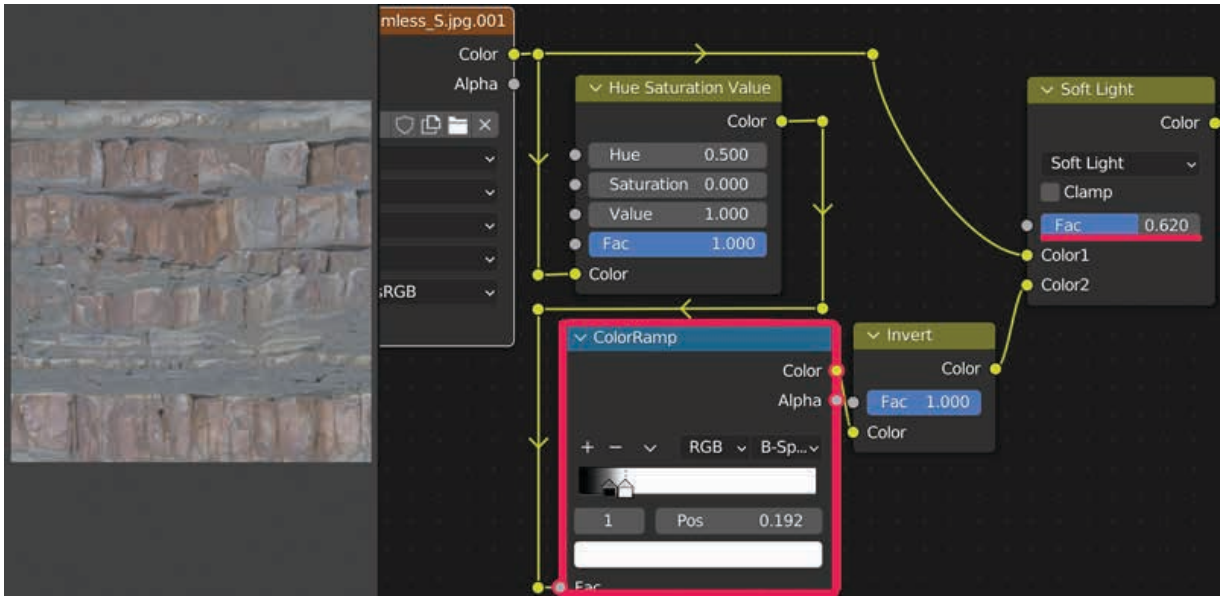
Now add an **Invert node** after the **Hue Saturation node**. As you can see, this makes all the dark parts of the image lighter, and vice-a-versa.



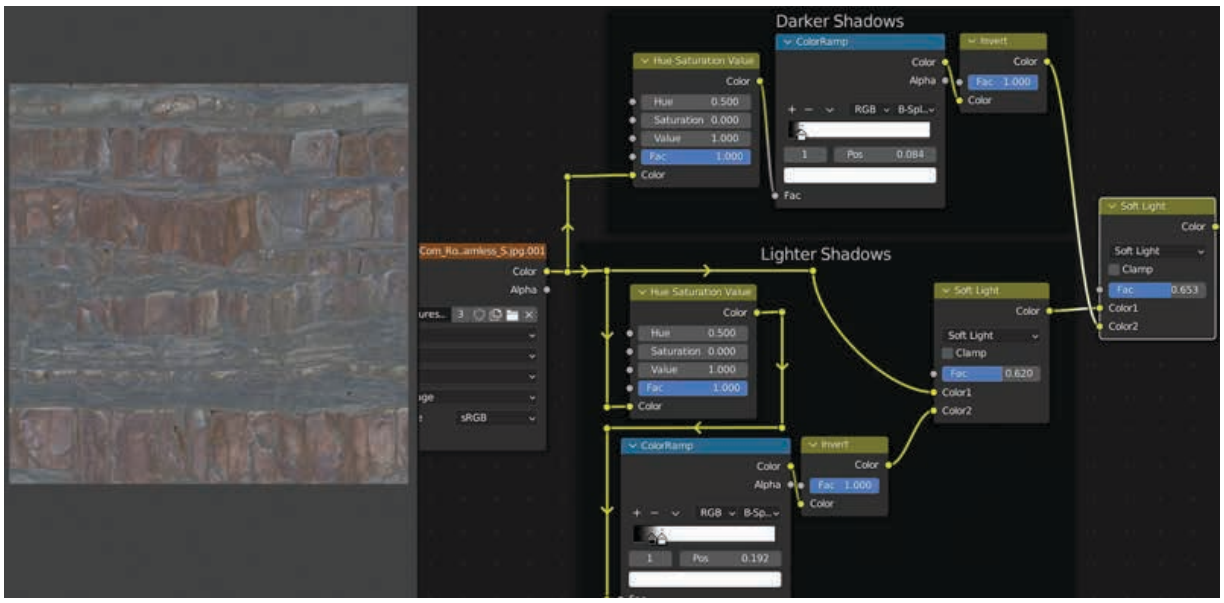
Add a mix **RGB Node** to combine the color output of the Diffuse texture with the nodes we've added. Set it to "**Soft Light**".



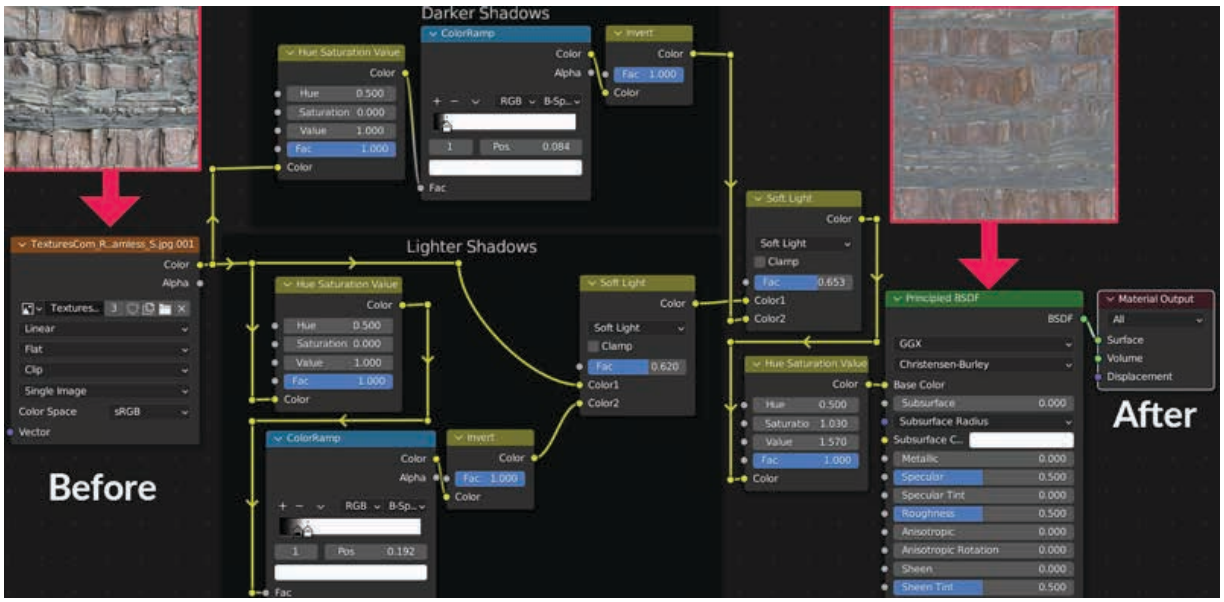
We can fine-tune the effect by adding a **Color Ramp node** in between the saturation and invert nodes. You can also tweak the mix factor of the soft light.



If there are some really dark shadows, you can **duplicate the nodes (Shift+D)** and tweak the **Color Ramp** to really target only those darkest shadows. Then mix the result back in with the other nodes.

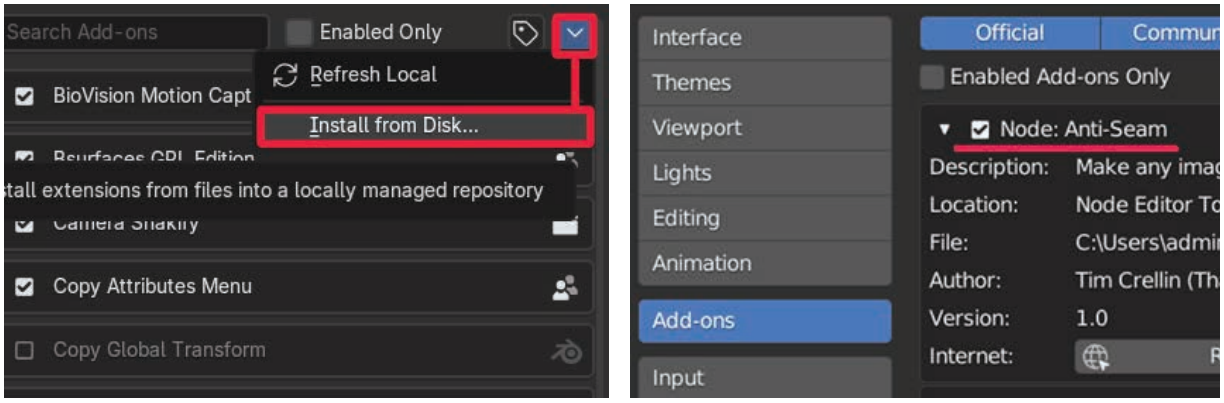


You may want to add another **Hue/Saturation node** at the end.

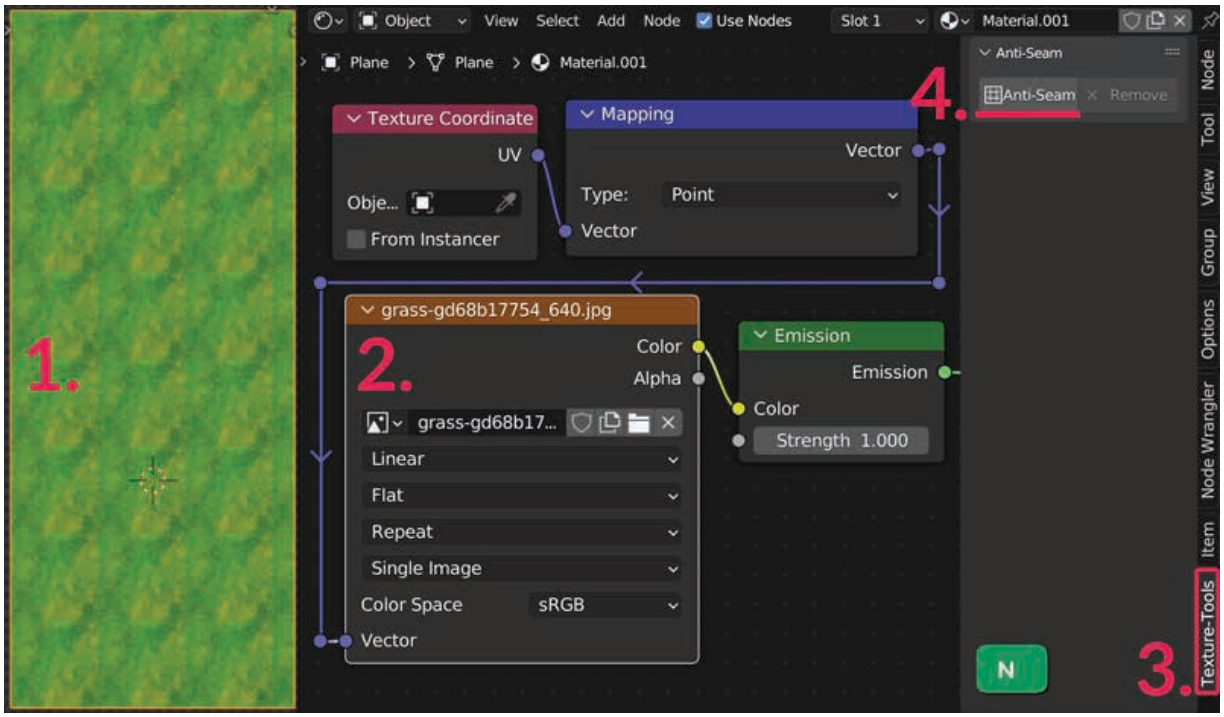


Seamless textures in one click with Anti-Seam

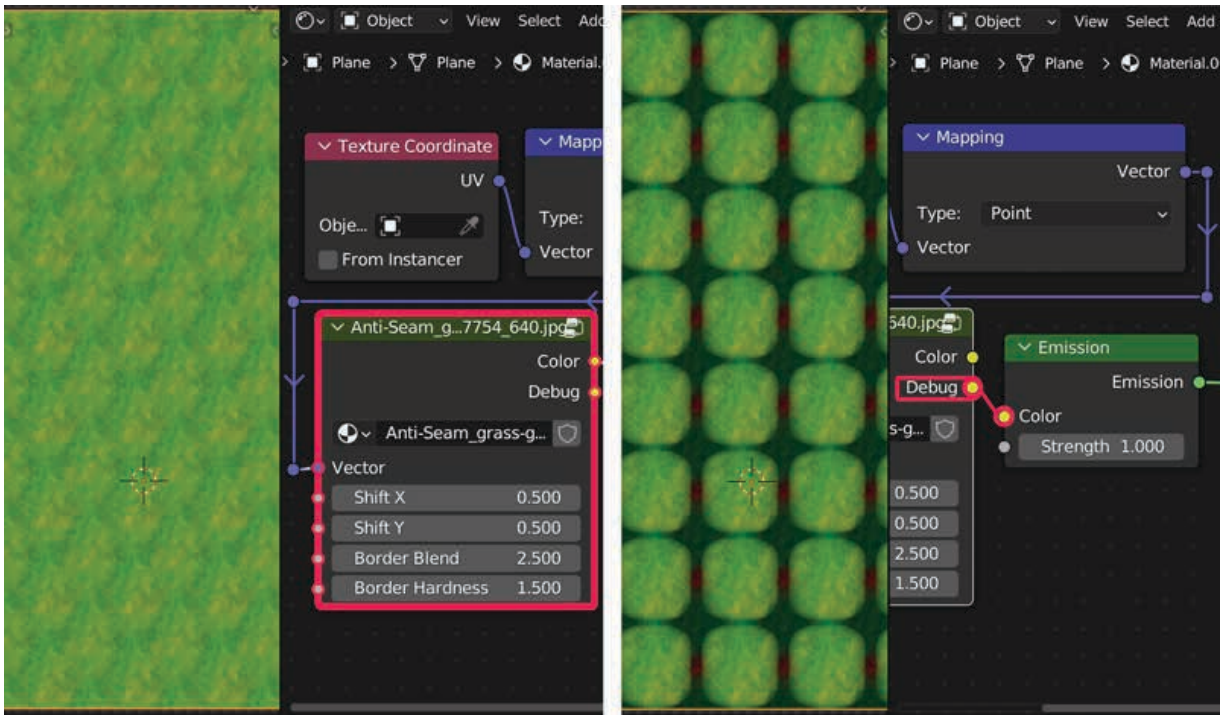
Anti-Seam is a free add-on that makes textures seamless inside of Blender with just one click. Download the add-on from Gumroad and install the zip file in the Add-ons tab in Preferences. It's free, but it's always a good idea to support independent creators so feel free to give them a tip.



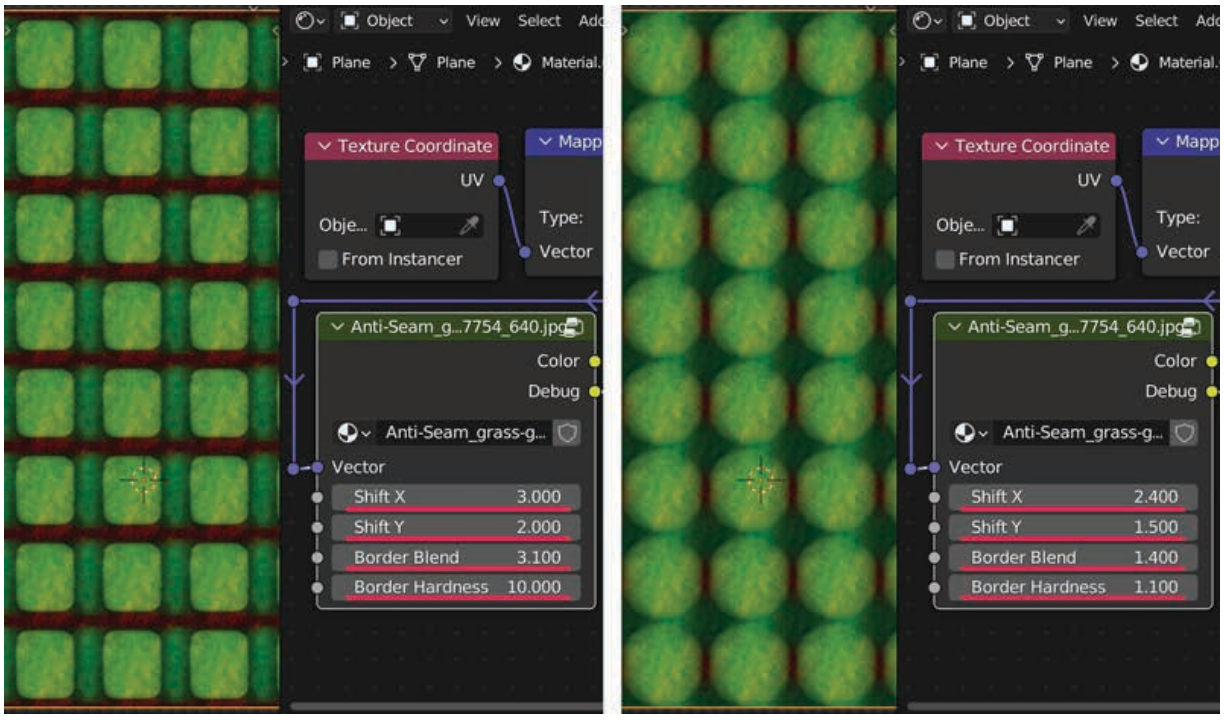
Then select your textured object. And in the shader editor window, **press N** to open the Option tab. Select the **Image Texture node**. Open the **Texture-Tools tab** and click on **Anti-Seam**.



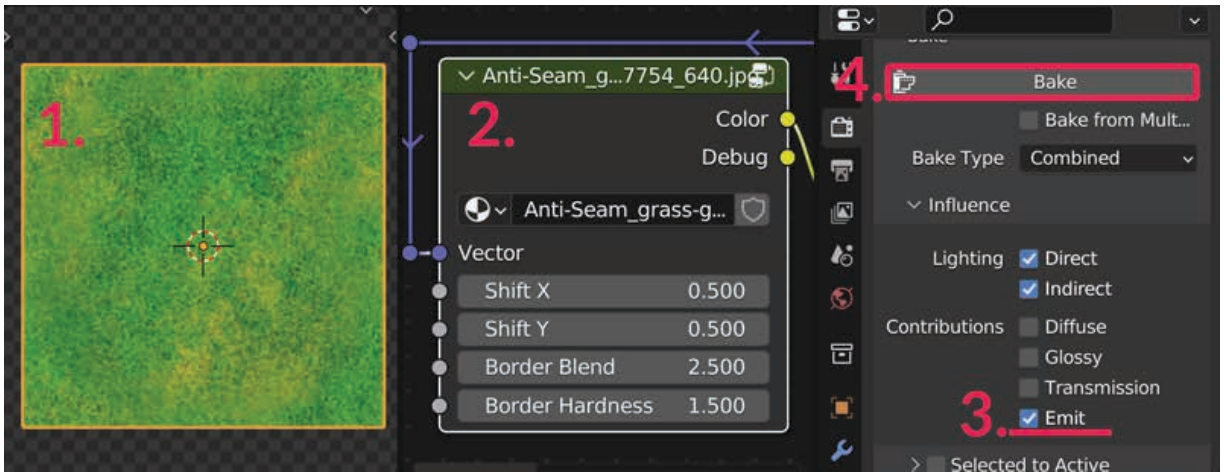
That's all - your texture is now seamless. The default values work well, but depending on your texture you may want to tweak these values. To see what they do, you can plug the **debug output** into the Material Node.



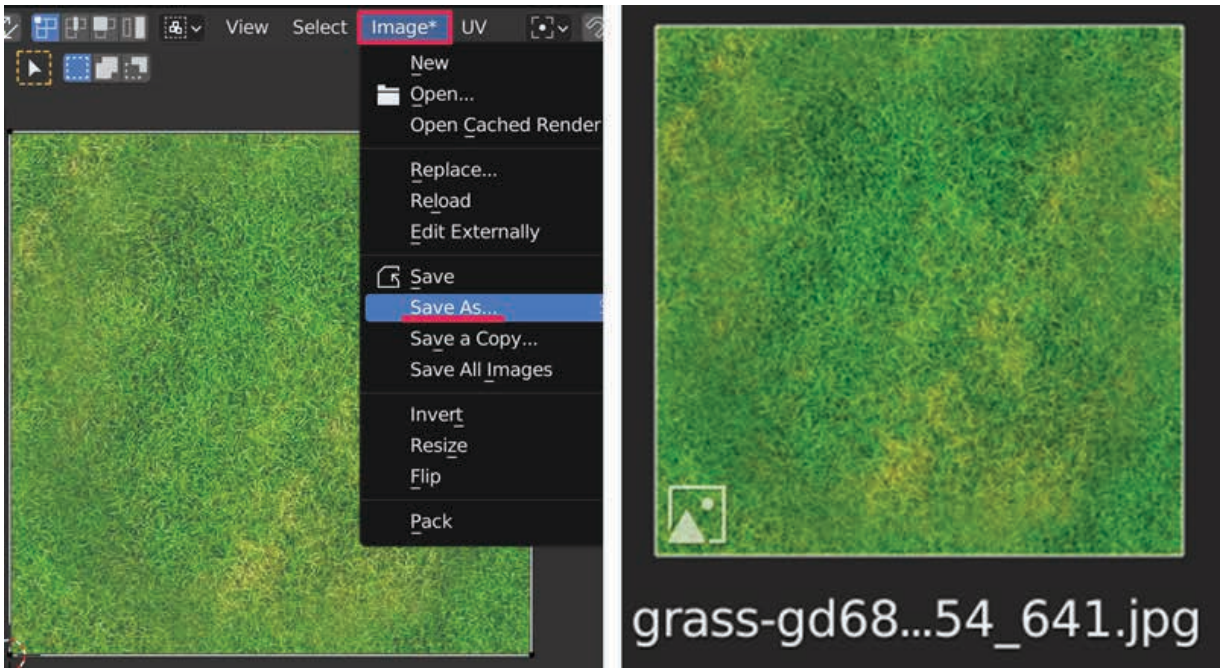
The Debug output is only used temporarily, to see how the borders of the tiles are blending. Once you're happy with the blending, switch back from the Debug to the **Color** output.



To export this texture, make sure you are in **Cycles** and then click on **Bake** in the Bake menu. It works best if you use an emission shader and check the Emit Box.



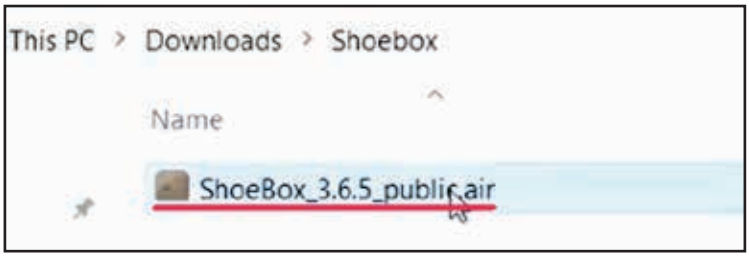
You will need to wait for Blender to finish, then you can save the baked seamless image from the **UV editor window**. It works best if you use an **Emission** shader and check the **Emit** box.



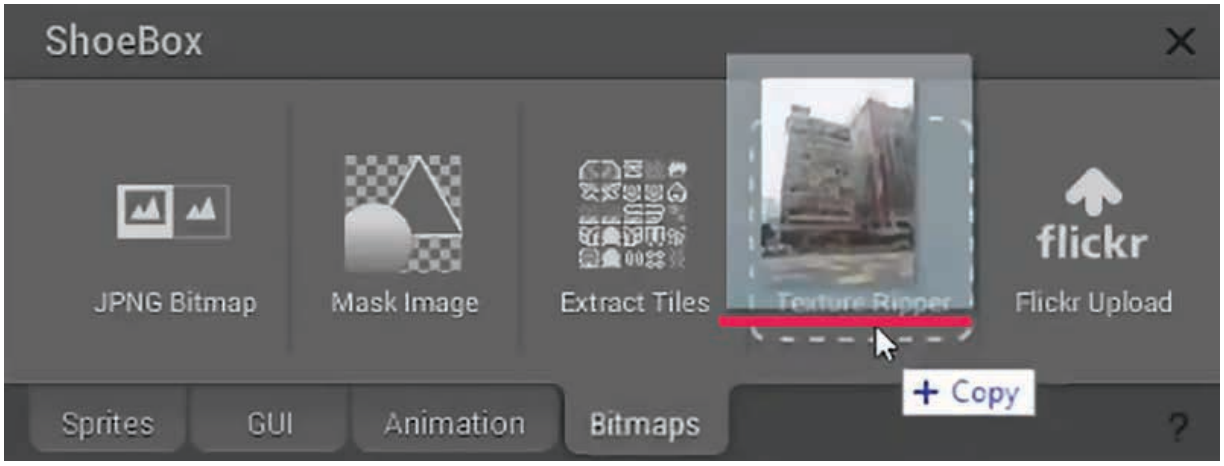
Texture Ripper

Texture Ripper is part of the free app Shoebox, from the creator of the popular Tex Tools add-on for Blender. It works separately from Blender. It's very useful if you want to get flat, perspective-free textures from photographs. Download Adobe Air. Then download Shoebox (includes Texture Ripper). Texture Ripper is another cool app by the creator of the essential TextTools add-on. I admit, this is not really a Blender tip, but it's quite useful if you want to use photographic textures for your models.

Click on the downloaded **Shoebox.air** file and open the Bitmap tab.



Open the **Bitmaps** tab and drag a photograph to the **Texture Ripper** field.



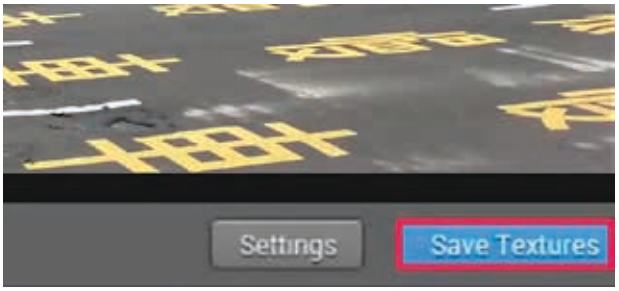
Now you can **click on the photo** and click again to create other points of rectangle.

Click here to return to the index.

This selection automatically gets rendered as a flat, perspective-free texture. You can zoom in and out with the mouse scroll wheel and refine your selection.

After this you can keep clicking to add more parts of the photograph to the texture.

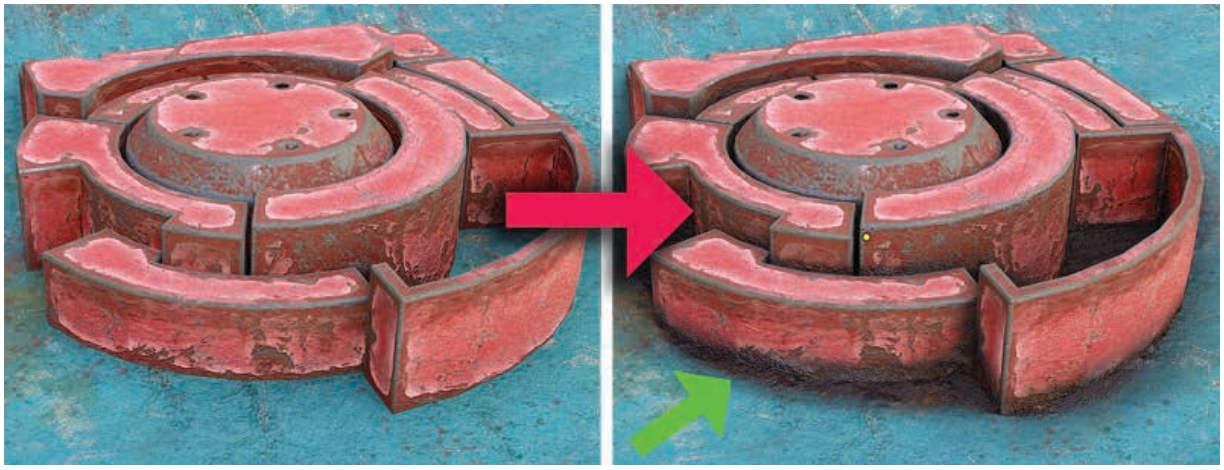
Once you're done, save your texture and build some stuff with it in Blender.



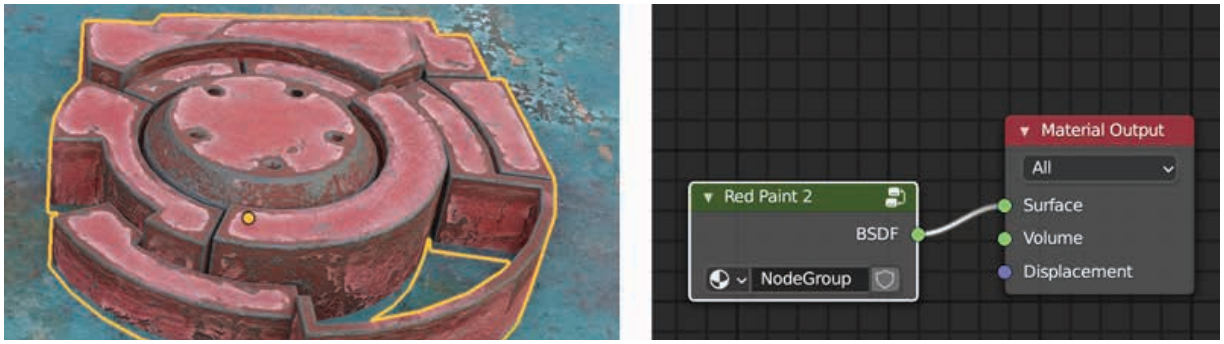
MATERIALS

Blending objects using Rust with the Ambient Occlusion node

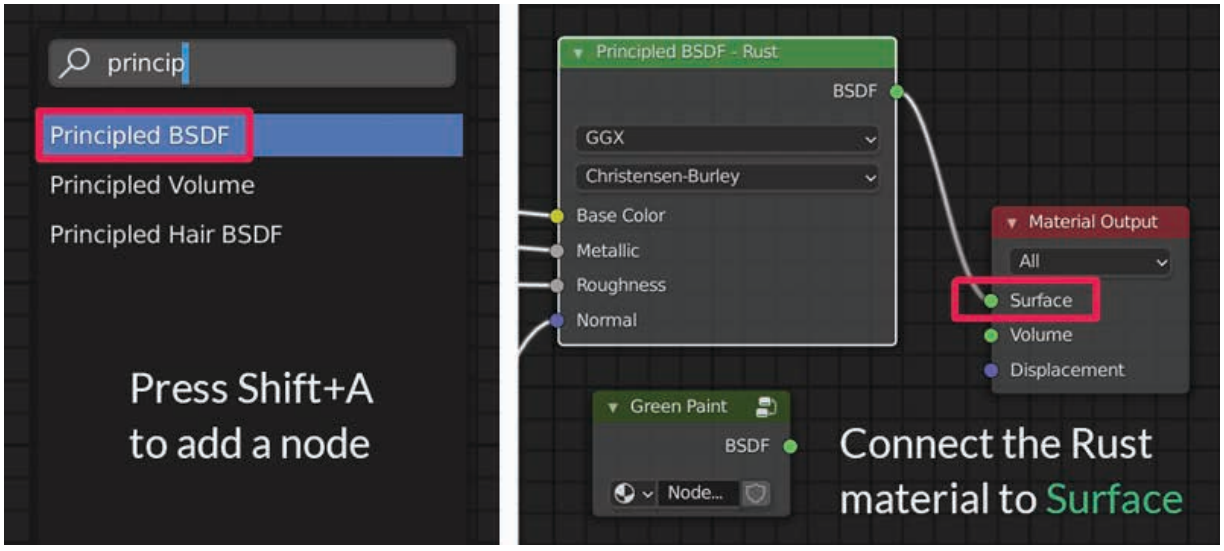
When you've got textured objects that you want to combine, there are several ways to "blend" the materials between the two objects. That way they look more integrated and realistic. In this case, we'll look at using rust (or dirt) to blend between the two.



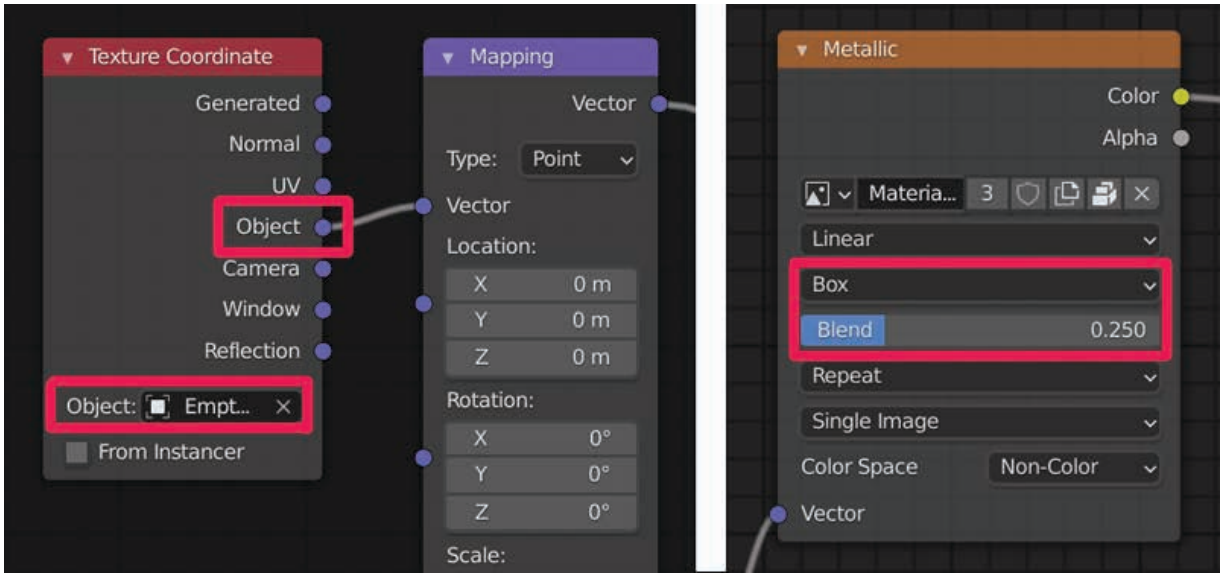
First, we'll assume that the objects already have materials. To make it a bit easier to see everything, the materials have been put in a **Node Group**. You can read more about how to do this in **the chapter "Node Groups" on page 927**.



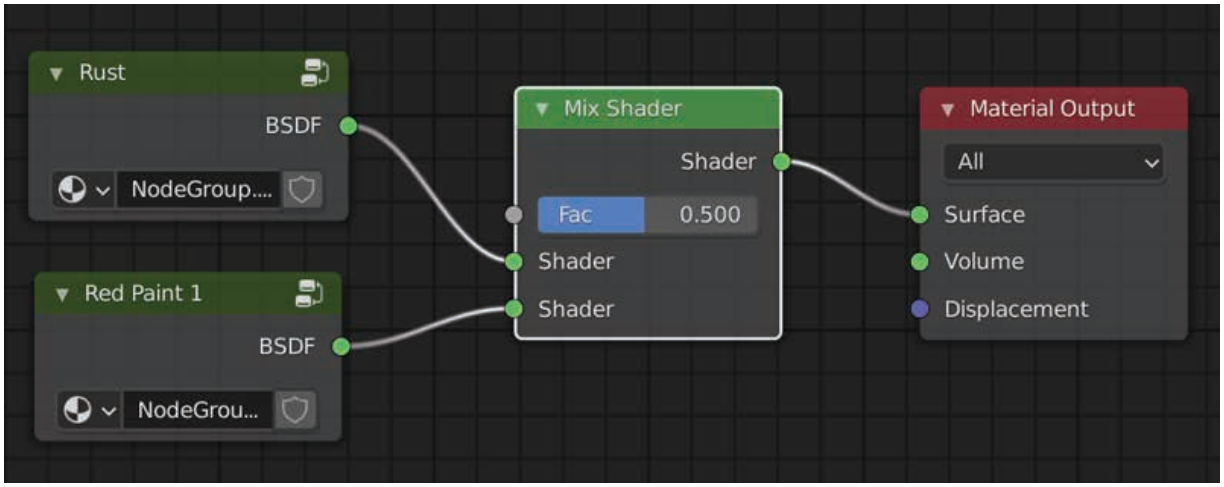
Now, with one of the objects selected, create a **rust material** in the Shader Editor window. In this case, I'm using a **Principled BSDF** with some textures. Unplug the existing material of the object (scuffed red paint in this case) and plug in the rust material to test how it looks.



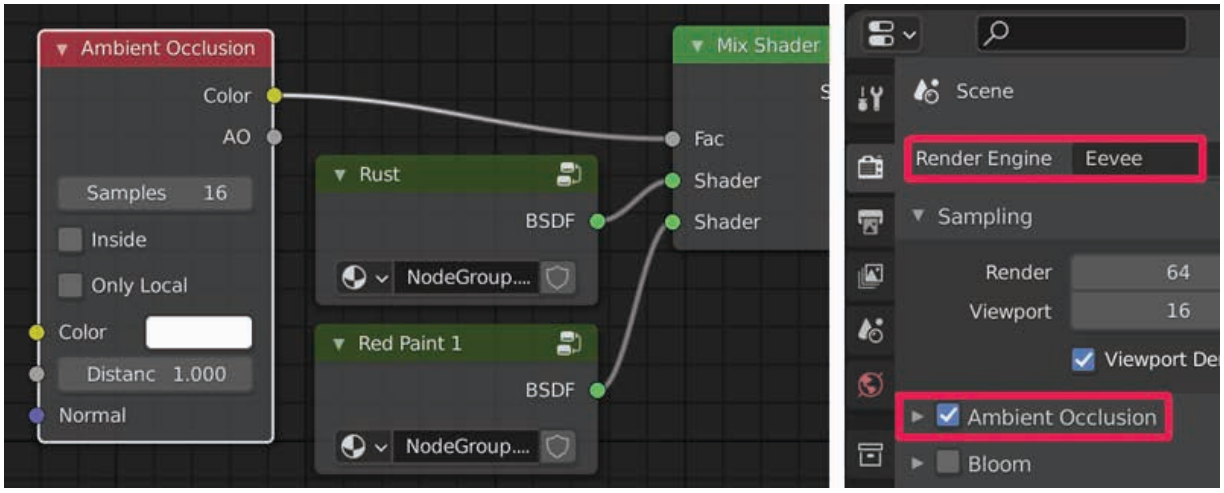
By setting the **Texture Coordinates** to **Object** and using an **Empty** as the object, the rust material looks uniform when applied to different objects. For all image textures that make up the rust material, set the **Projection** type to **Box** with a **Blend value of 0.25** (for more details, check the chapter about Box Mapping, "Texturing Without UV-Unwrapping" on page 1289.



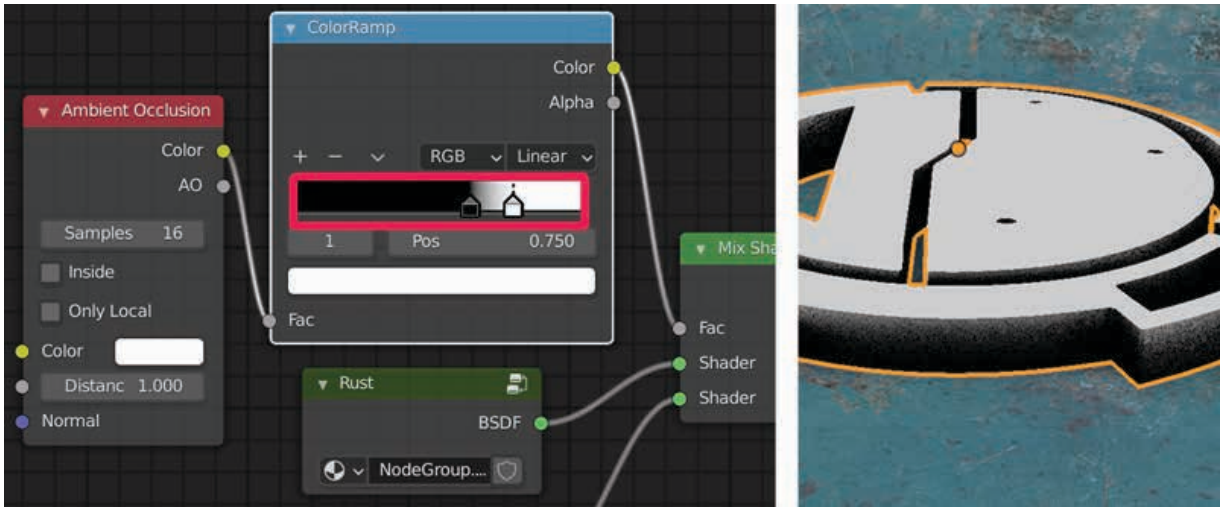
When you're satisfied with the rust material, add it to its own Node Group by selecting all its nodes and **pressing Ctrl+G**. Change the label name to **Rust** to keep things clear. Add a **Mix Shader** node and connect both materials to it. The Rust material should go into the top input. When you slide the **Fac** value, you can see that it fades into either the Rust or the Red Paint material across the entire object.



To control where we see one material and where we see the other, add an **Ambient Occlusion node** and plug it into the Mix Shader **Fac** input. If you're using Eevee, you need to enable Ambient Occlusion in the Render Settings to see the result. In Cycles, it works by default.

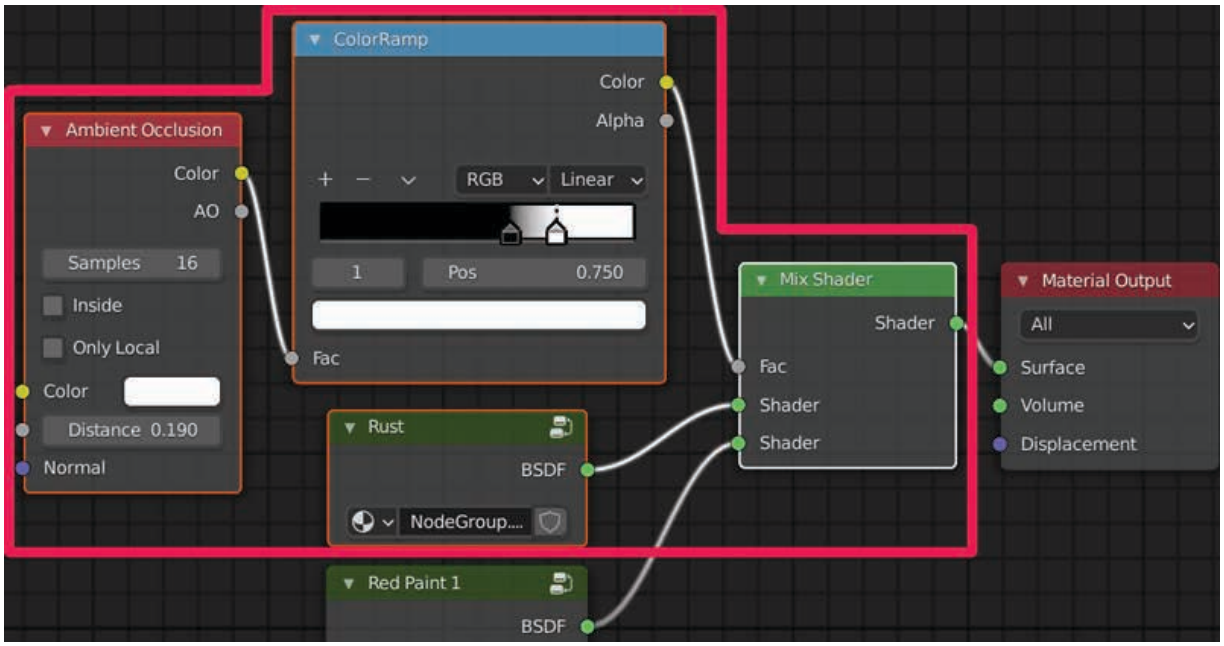


You probably still don't see much, until you add a **ColorRamp** node to increase the contrast of the Ambient Occlusion by moving the stops closer together. **Hold Ctrl+Shift and click** on the **Color Ramp** node to see what you're doing in the viewport (you need the **Node Wrangler add-on** to be enabled for this to work).



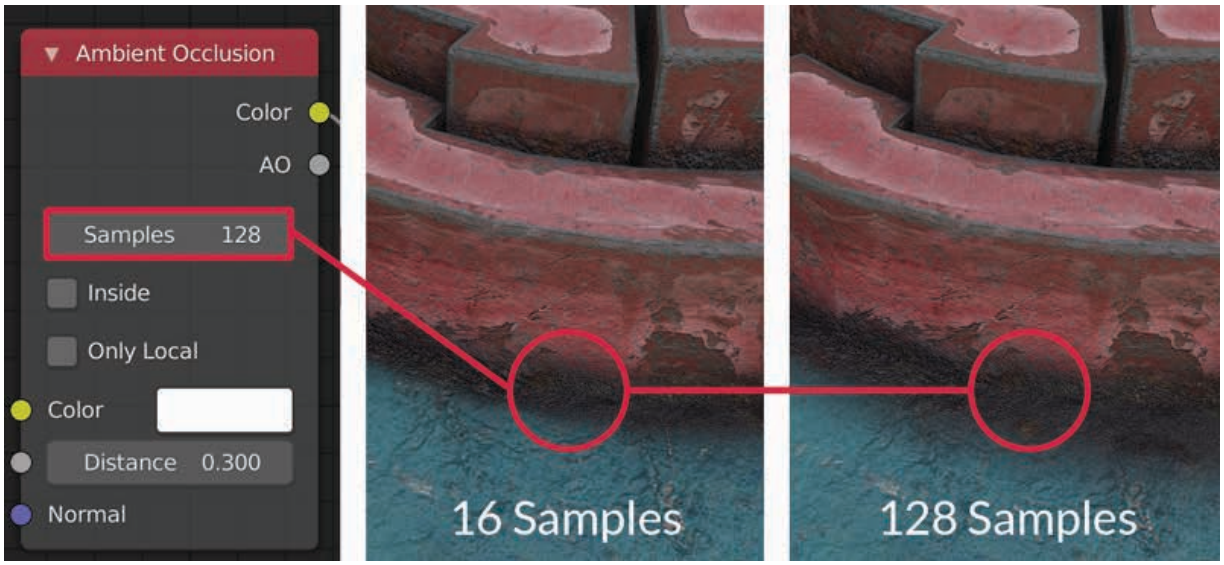
Then, finetune it to your liking by decreasing the **Distance** value. You can **hold Shift** for finer control when changing the value. This way you can control how far the rust goes.

Hold Shift+Ctrl and click on the Mix Shader node once you're done fine-tuning the Ambient Occlusion and ColorRamp values. Copy the necessary nodes *(Ambient Occlusion, ColorRamp, Mis Shader and of course the Rust node group)* by selecting them and **pressing Ctrl+C** and paste them in the **Shader Editor** window for the other objects by **pressing Ctrl+V.** Then connect them to the Material Output of that material.



What's nice with this method is that you can still move the objects, the rust blend will follow. And since the mapping coordinates of the Rust material are set to **Object,** you can use the **Empty** that is set as the object to control the scale of the rust.

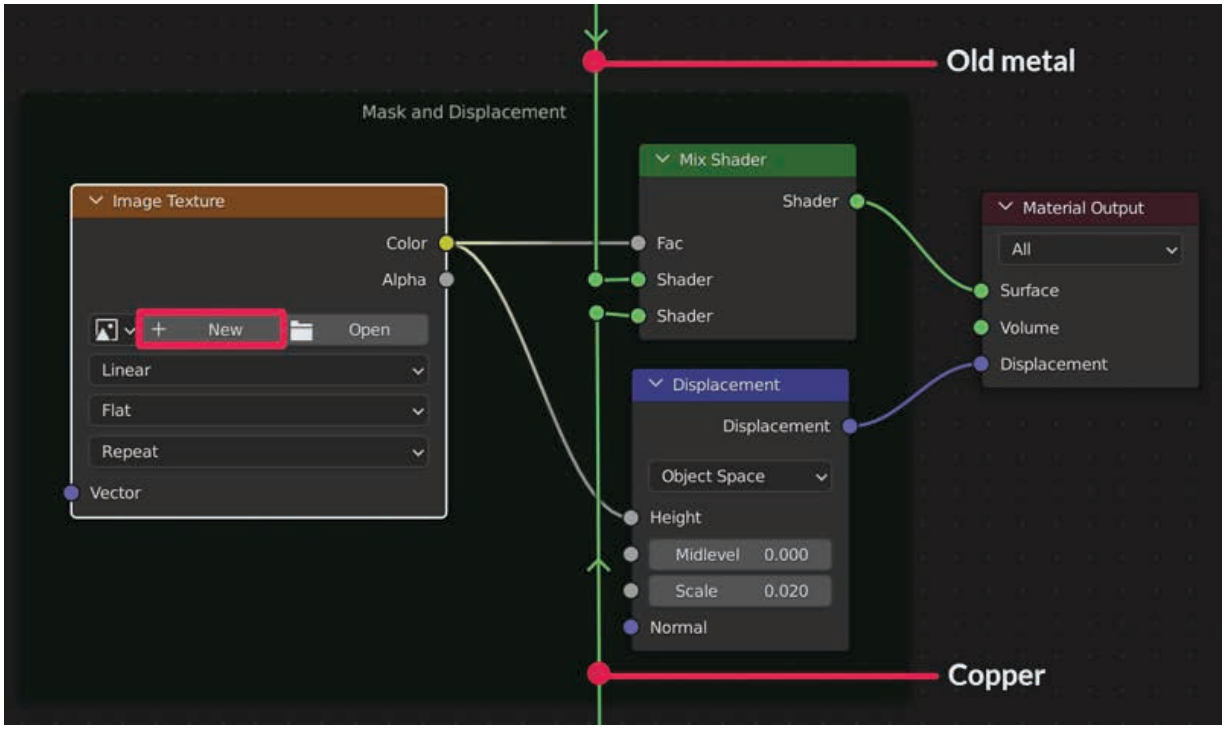
So you can still experiment with its look as well by scaling and transforming the empty. One final value of note is the amount of Samples. This can also be used to increase the distance the ambient occlusion reaches.



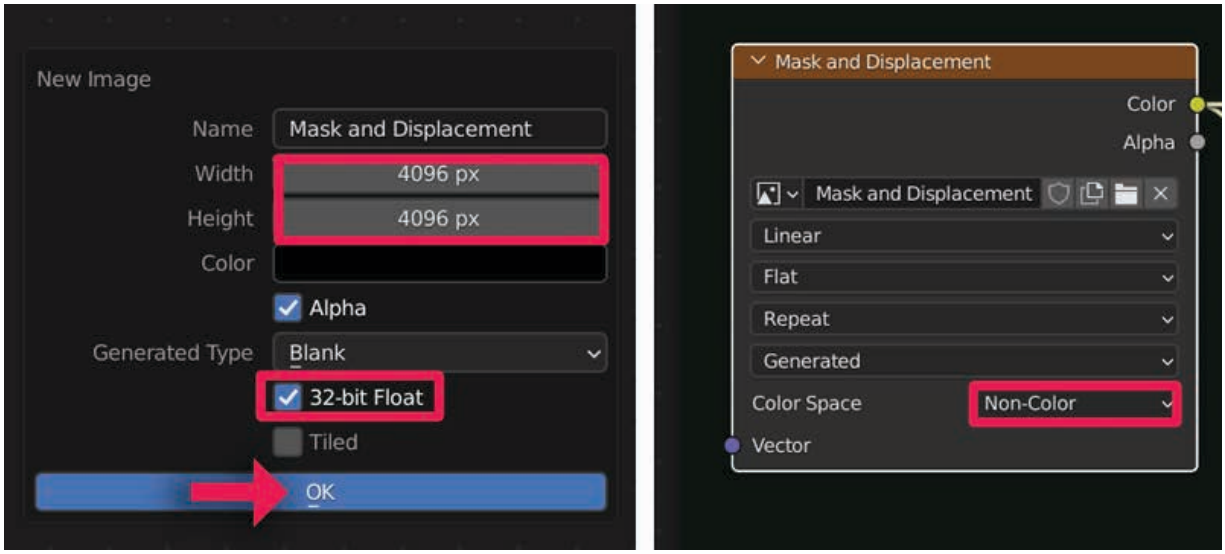
Mixing Shader Masks with Displacement Maps

Previously we looked at using textures as both masks and bump maps at the same time. Of course, the same thing can be done with Displacement maps. That way, you get actual displacement of geometry. In combination with alpha textures that you would normally use in sculpting, this can create an interesting effect.

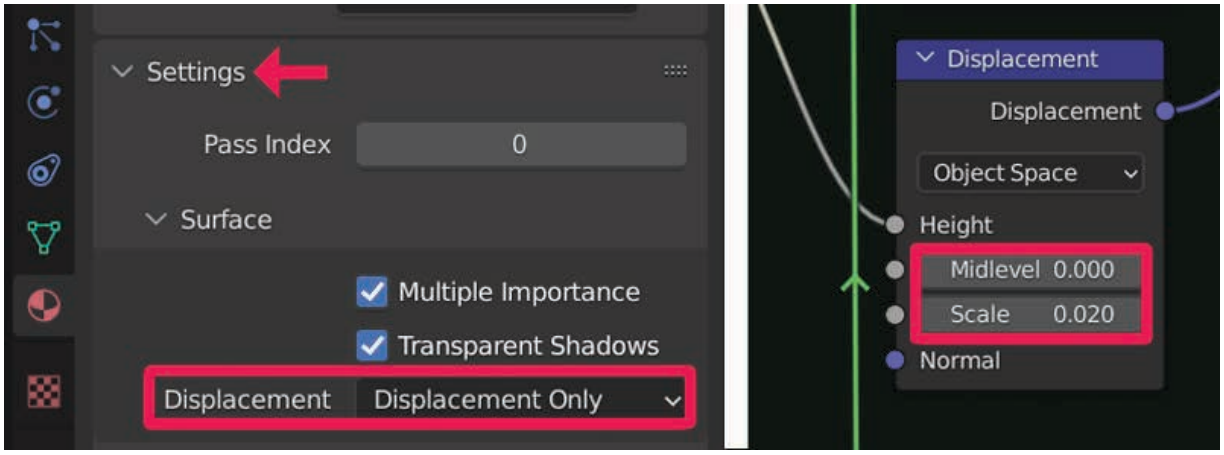
Just like in the previous topic, we have two materials plugged into a Mix Shader node. As the mix factor, we have an Image Texture node. The image texture also goes into a Displacement node. **Click on New** to create a new image texture.



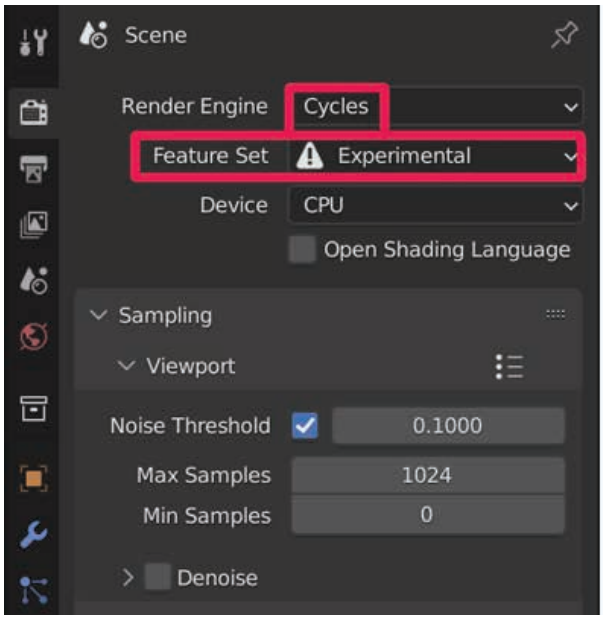
Call it something like “Mask and Displacement” and set it to 32-bit, 4K resolution. Click OK. Set the Image Texture node to **Non-Color**.

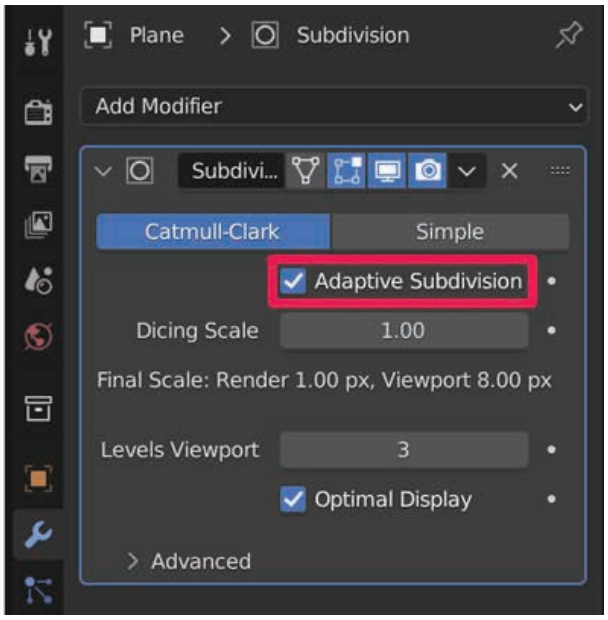


In the material settings, make sure Displacement is set to **Displacement Only**, otherwise it won’t work. Since we’re starting with a 100% black image texture, set **Midlevel to 0**. Normally, displacement maps are mid grey, but in this case we’re also using it as a mask so it makes more sense for it to be black. You can **reduce the Scale** as well. Otherwise the displacement will be far too extreme.

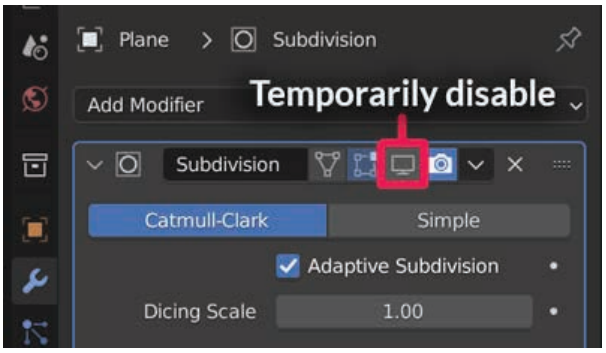


Make sure your object has enough geometry for displacement - ideally, you should use the **Experimental feature set** of Cycles and enable **Adaptive Subdivision** on the Subdivision modifier. This always gives the best displacement quality, as it subdivides things closer to camera more than those further away from the camera.

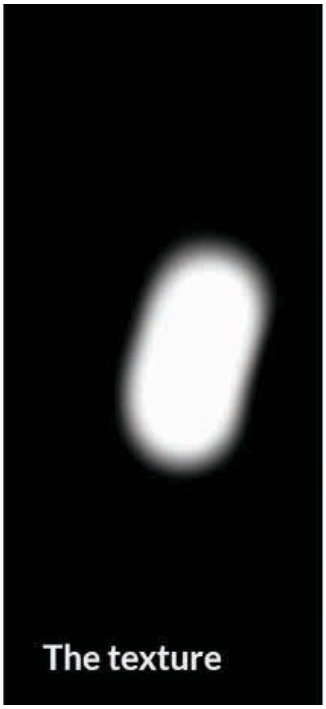




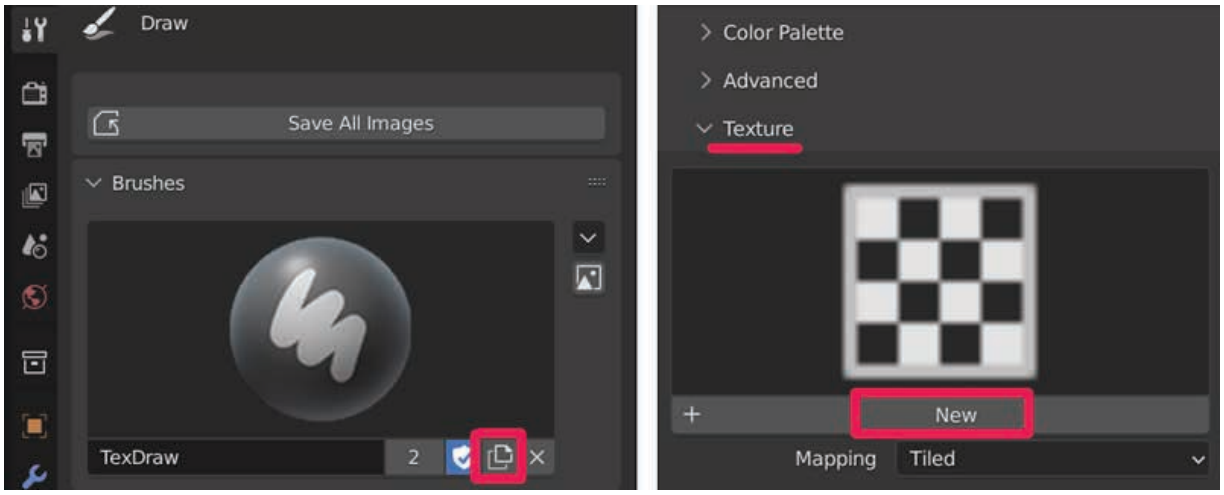
Turn the Subdiv modifier off for now (otherwise, texture painting will be painfully slow).



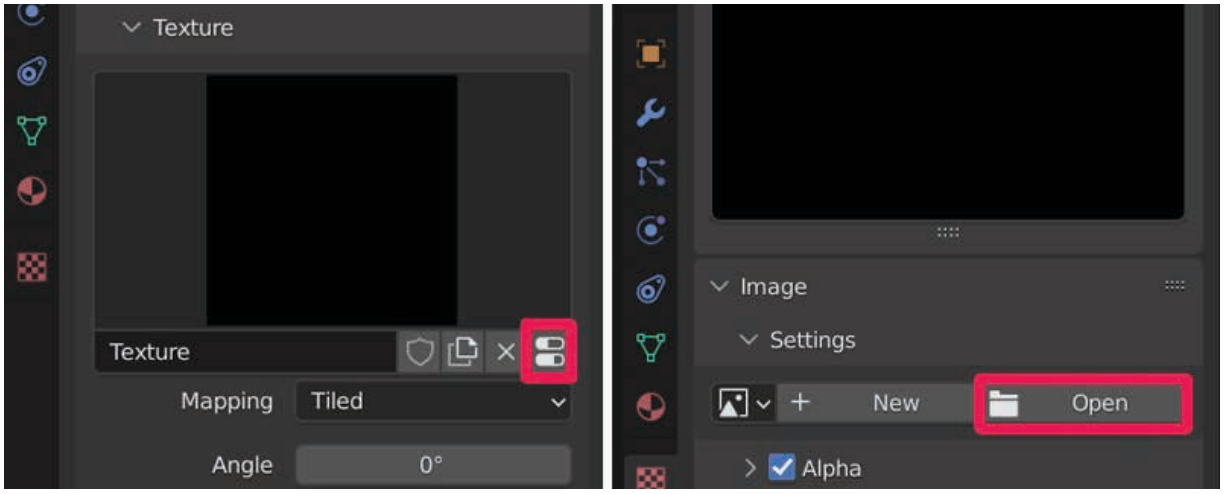
Now, when you paint a stroke in the viewport, you won't notice any displacement but you can see the mask revealing the copper material. The actual mask is just black and white. **Turn the Subdiv modifier on** again. In **Rendered preview**, you can see the displacement - as long as you're using **Cycles**!



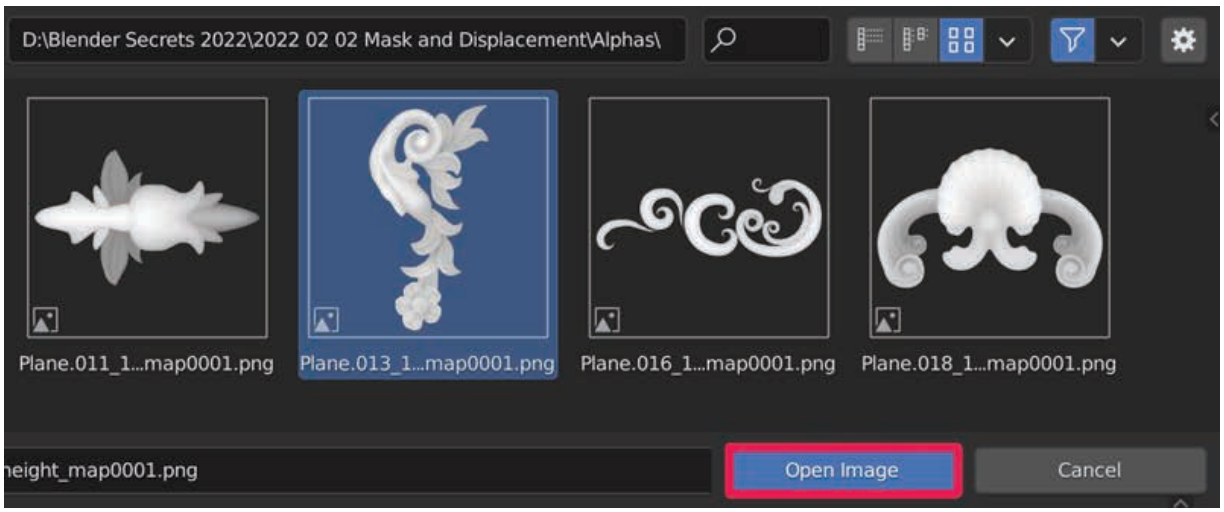
Okay, so it works - but let's create a more interesting displacement than just a stroke. Press Ctrl+Z to undo. You won't see the change until you go back to Material preview. **Click on the Add Brush icon** and go to the Texture settings of the duplicated brush. **Click on New**.



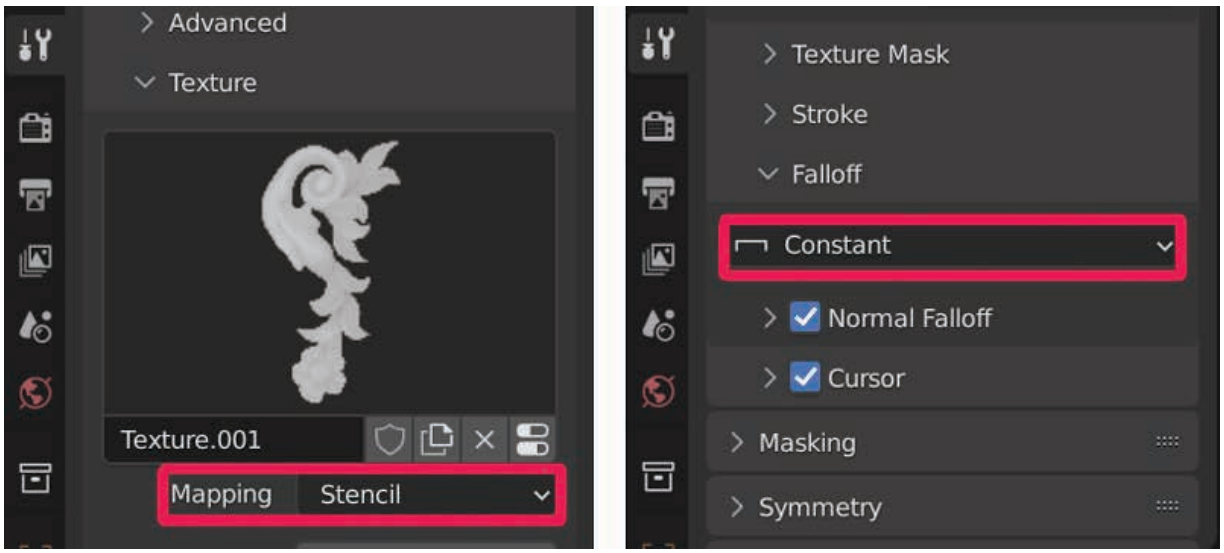
Click on the Texture properties icon and then **click on Open** to load a texture.



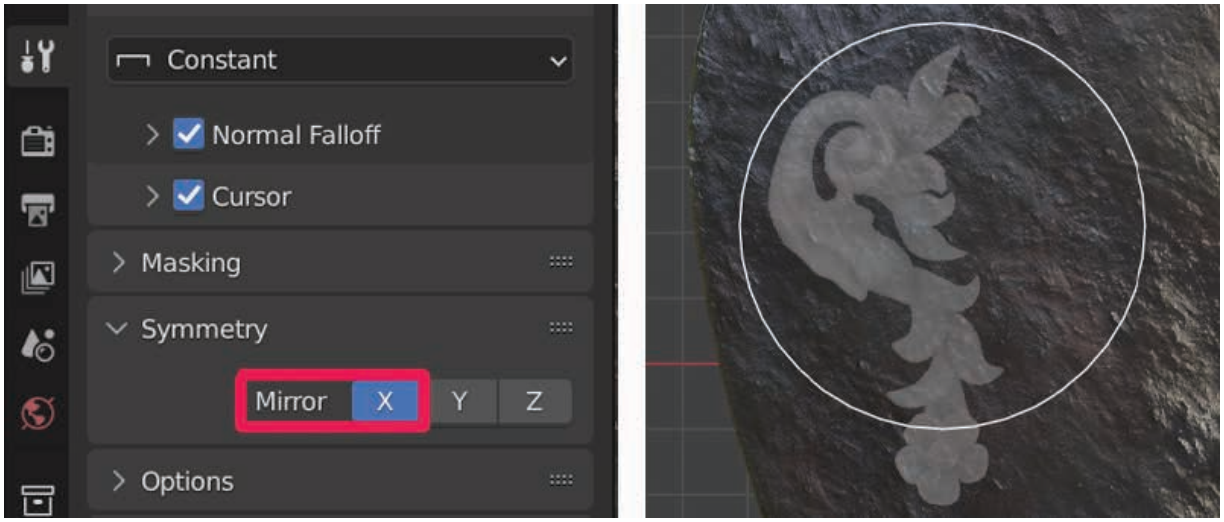
Choose an Alpha texture and click on **Open Image**. These are some alphas I created for an earlier topic. You can download them here.



Now, set **Mapping to Stencil** so that you have a preview of the texture in 3D view, which makes placing it where you want much easier. And set **Falloff to Constant** so that the entire image is used, without getting softer towards the edges.



Set Symmetry to **Mirror X** (if you want symmetry, that is). As you can see, we have a preview of the texture in the viewport. By holding the **Right Mouse Button**, you can move it to where you want. **Holding Ctrl+RMB** rotates it, while **holding Shift+RMB** scales it.

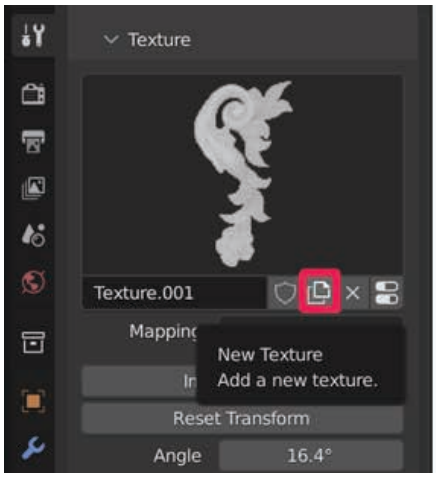


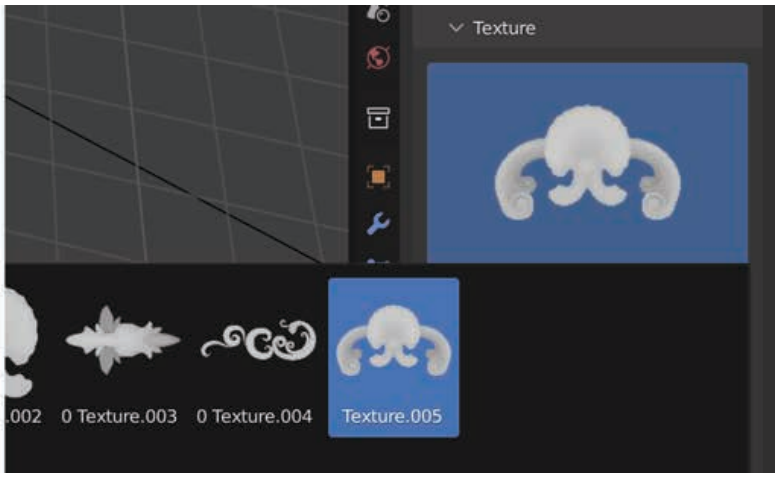
When you are happy with the placement of the stencil, make sure the **Radius** of the brush is just slightly bigger than the texture (otherwise part of it will be cut off) and that the Strength of the brush is high enough (I set it to 1 in this case).

Finally we can paint the alpha texture on the surface of the object by **clicking the Left Mouse Button**. It looks like the masking worked, but what about the displacement?

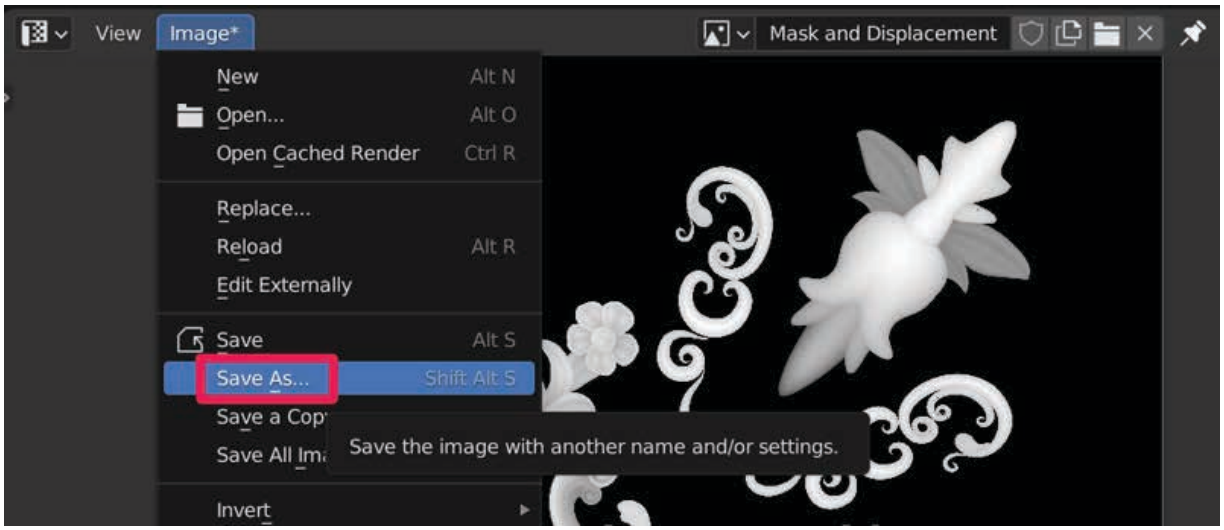
Let's switch over to **Rendered Preview**. Looks like the displacement is working nicely!

When you want to use other alpha textures, click on **Add new texture** and then load the one you want. When you click on the texture icon, you can see and choose the textures you've loaded thus far.



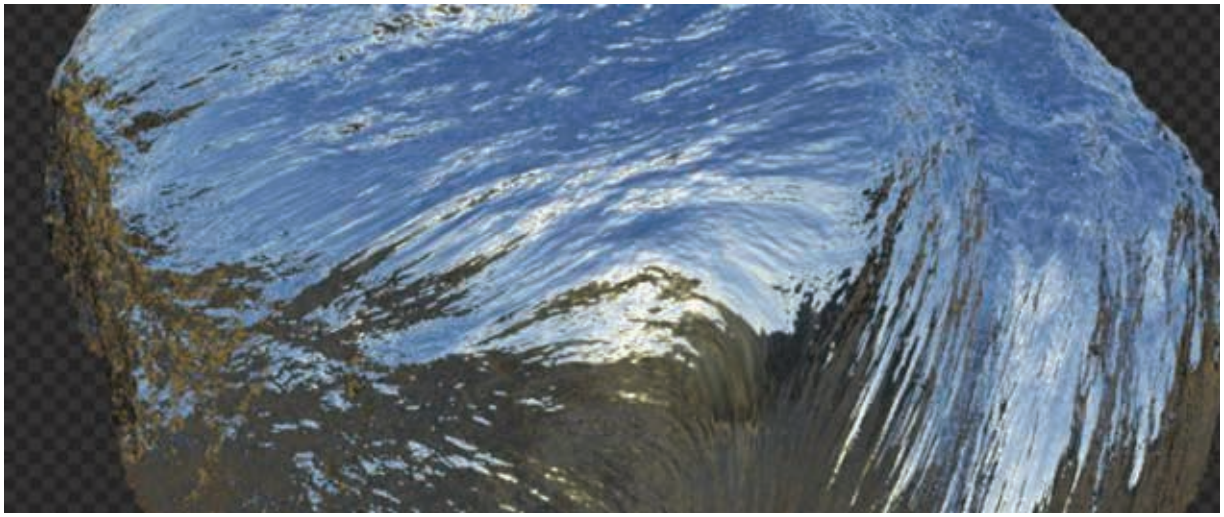


As always with texture painting, please **don't forget to save** your mask / displacement texture before closing Blender! You can do this from the **Image Editor** window or **UV Editor** window.

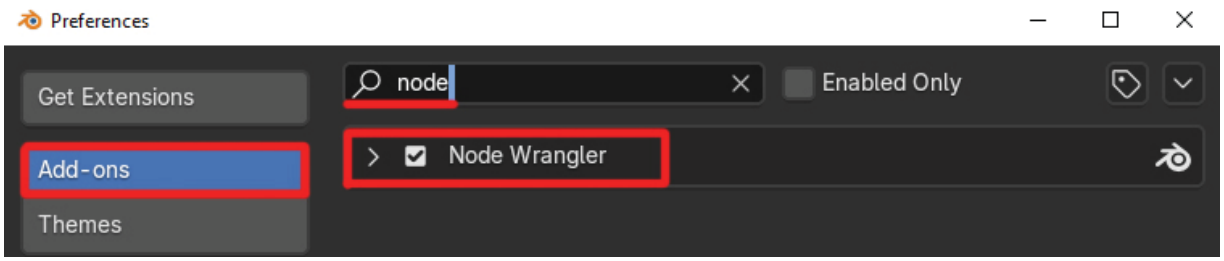


Micro-displacement

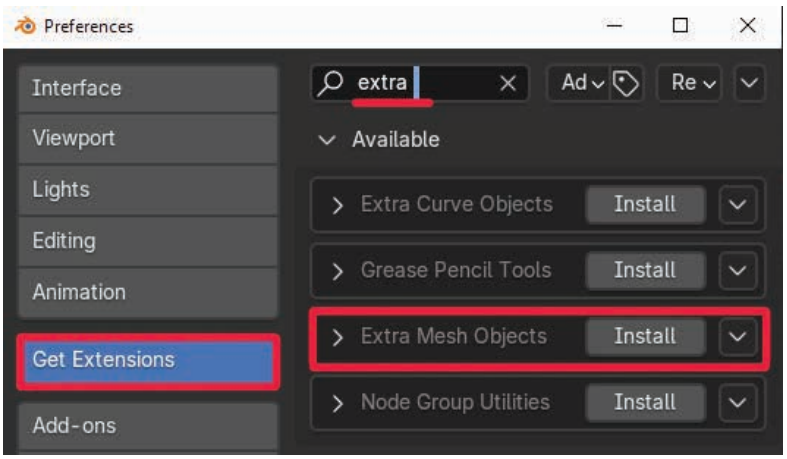
Micro-displacement is a method to get very fine detail in a very memory efficient way. It allows you to get more detail with faster render times.



For this tip I'm using the **Node Wrangler add-on** to be able to see the noise texture settings by previewing that node only.



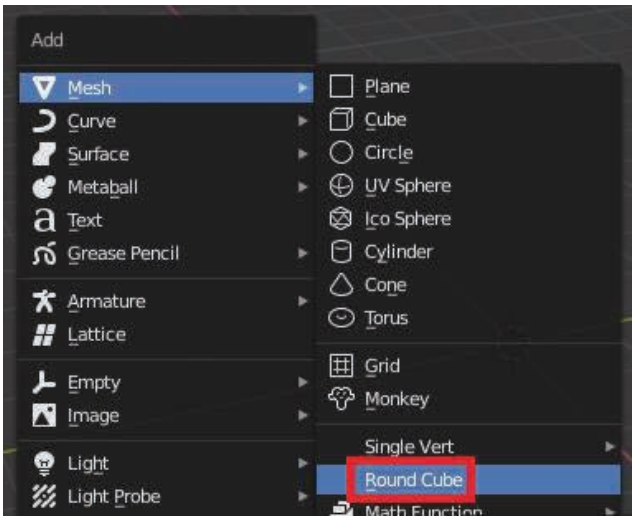
I'm also using the **Extra Objects** add-on for the round cube that I'm using for the example.



Both of these add-ons are included by default in Blender, you will however need to enable them in Preferences. You don't need the add-ons for micro-displacements to work. I use them to speed up my workflow.

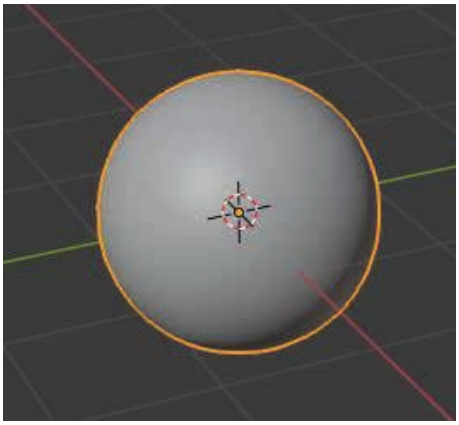
Please be aware that when using GPU rendering, microdisplacements might take a big chunk of memory and what you have on your graphics card it might not be enough.

You may need to use CPU rendering instead. Delete the default cube and add a **Round Cube** and subdivide smooth it twice.

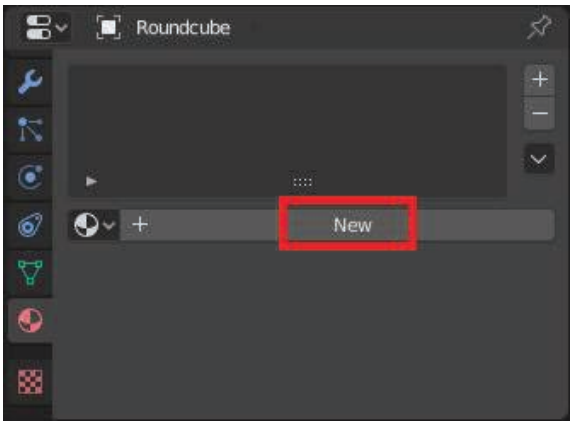


Press Ctrl+2 to add 2 levels of subdivision and make the Cube smoother.

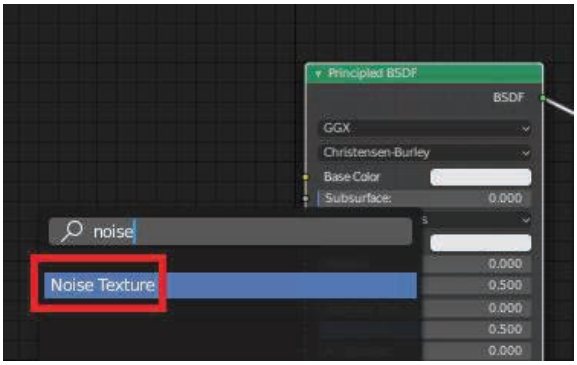
Right-click on the Round Cube and choose **Shade Smooth**.



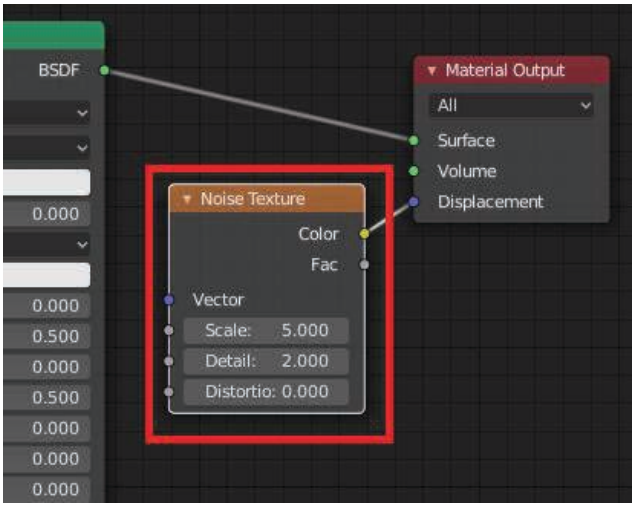
Add a new material by clicking on **New**.



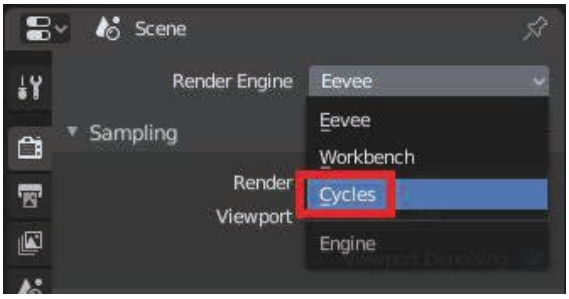
In the Shader Editor, press **Shift+A** and type **Noise** to find the procedural **Noise Texture**.



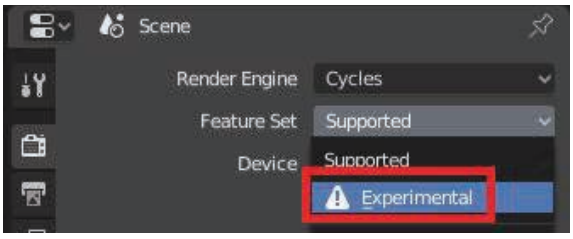
Plug the Noise Texture into the Material Output **Displacement** socket.



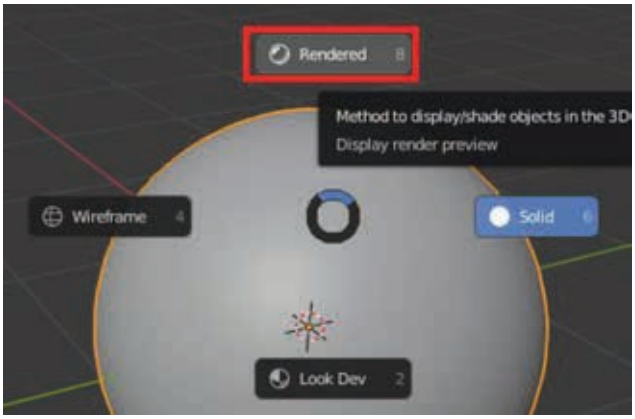
Make sure your render engine is set to **Cycles**.



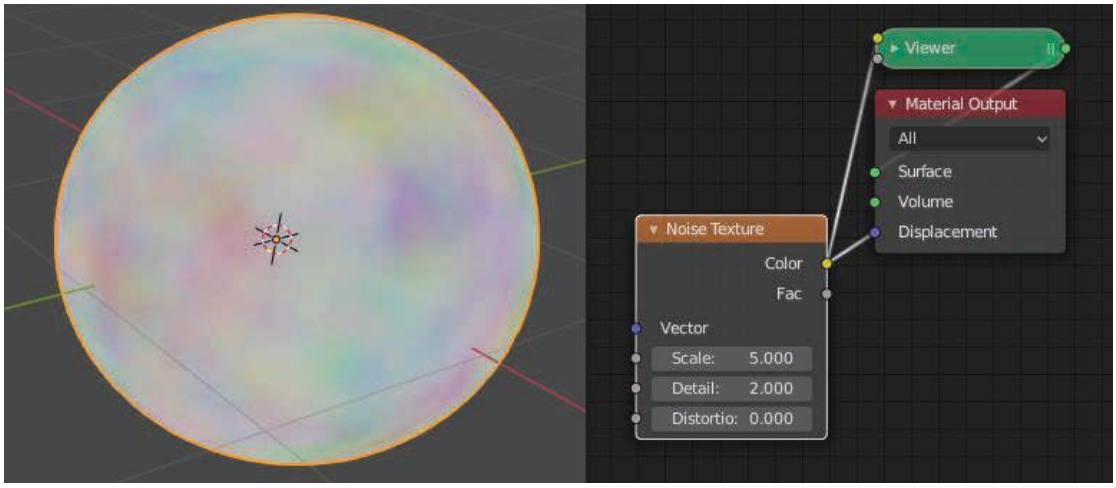
Feature Set should be set to **Experimental**.



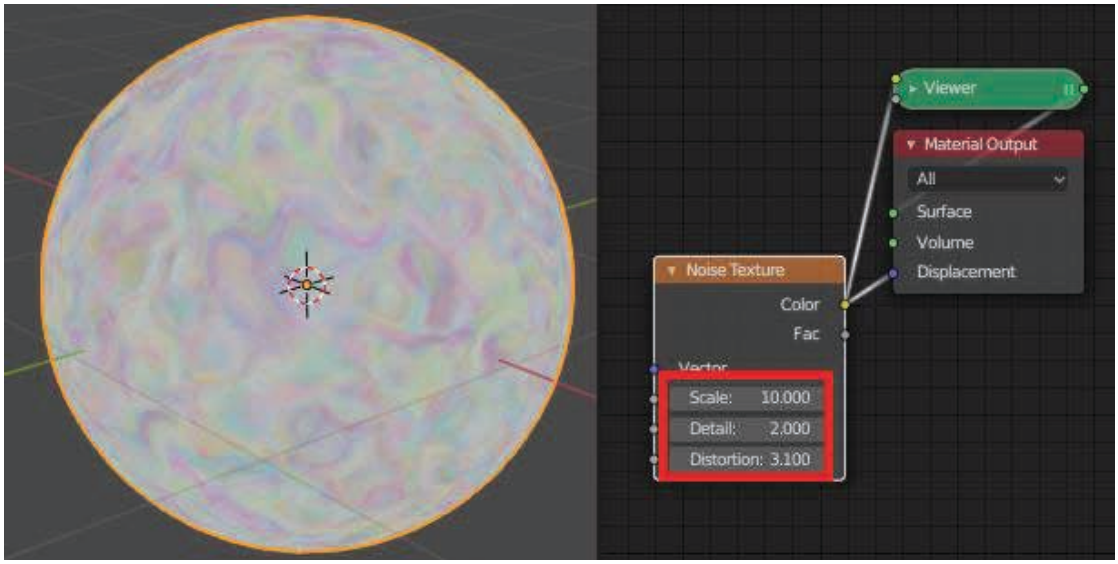
Set viewport shading to **Rendered**. You can do this by **pressing Z** and choosing Rendered from the menu.



By **Ctrl+Shift-clicking** on the Noise Texture node, you can preview what the noise itself looks like and fine-tune it to your liking by changing the Scale, Details and Distortion values.

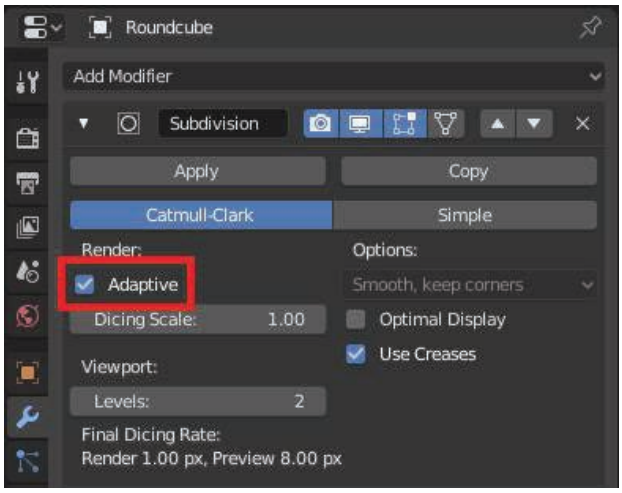


I think these values give an interesting result.

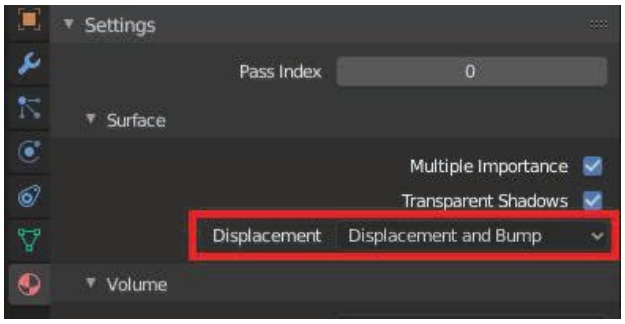


Ctrl+Shift+click on the Principled BDSF node to see the normal material again.

Turn on **Adaptive** in the Subdiv modifier.

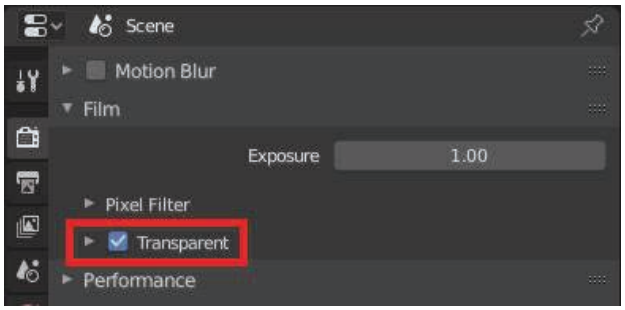


In the Material Settings, set Displacement method to **Displacement and Bump**.



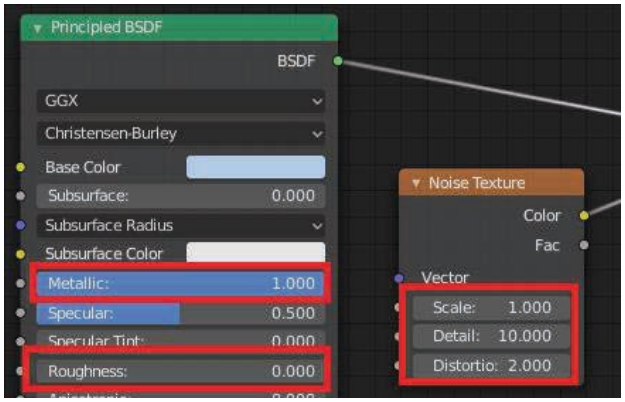
You'll notice the Round Cube changing dramatically in the viewport.

Note that I'm using an HDRI to make the light a bit more interesting and I've set the background to **Transparent**.



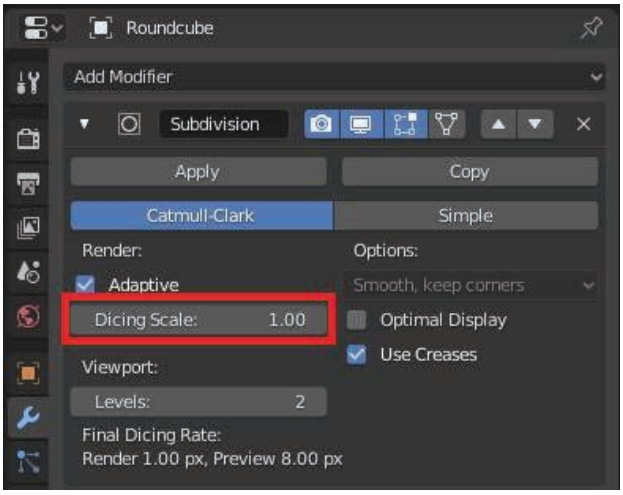
After seeing the result of the Displacement, I adjusted the Noise Texture values until the result was less extreme.

I also made the material really glossy by setting **Metallic to 1** and **Roughness to 0**.



Last but not least, the **Dicing Scale** value controls how much detail there is.

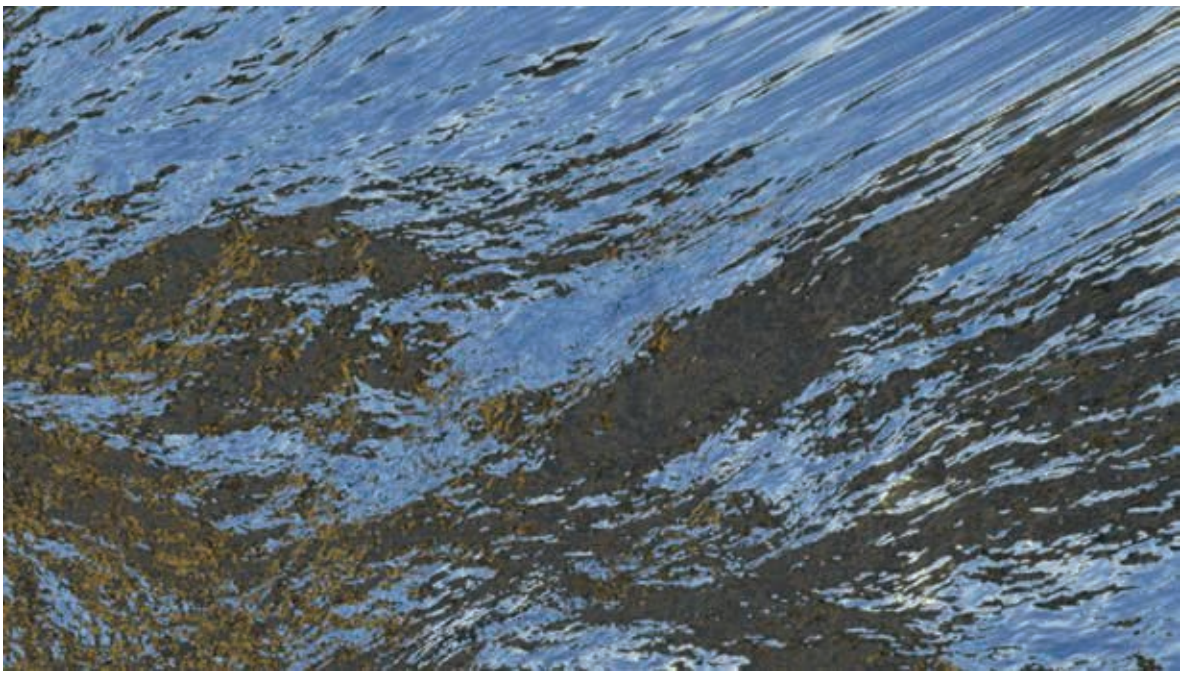
A lower value (1 or 0.5) means more detail.



The interesting thing about this technique is, Blender will calculate how much detail it needs considering how far the camera is from the object. So you get more detail as the camera gets closer to the object. This saves a lot of memory as the detail is only there for things close to the camera.

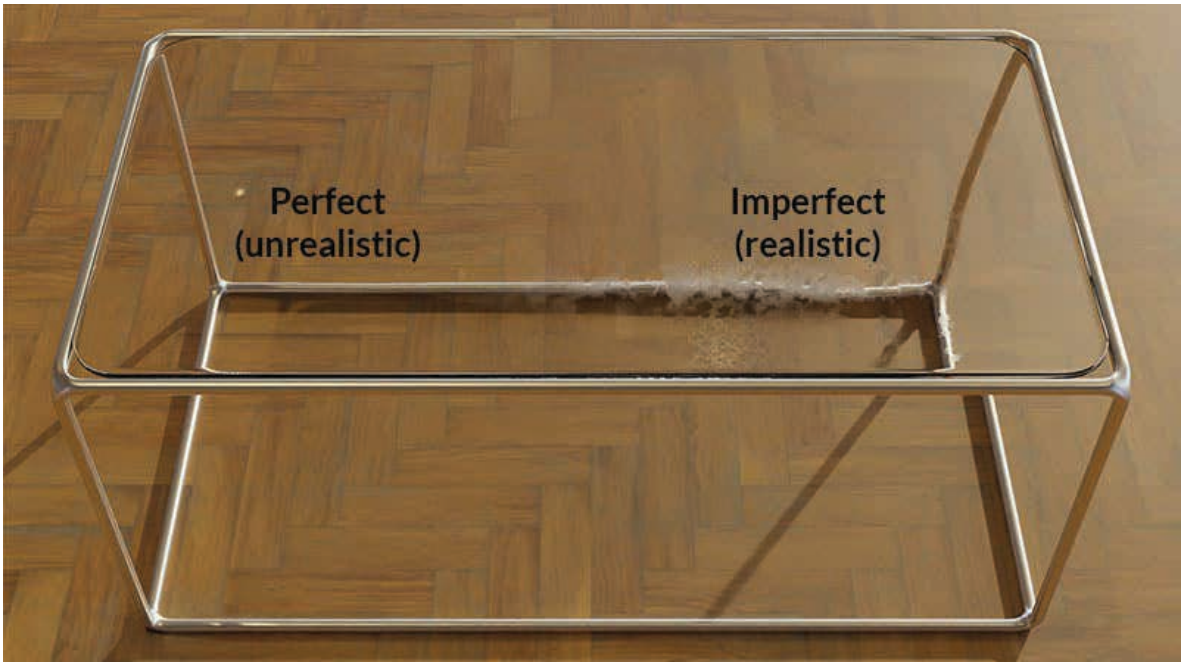
Even so, you may need to be careful not to run out of memory - that's why the Dicing Scale value is important.

When using the procedural noise texture and microdisplacements, I can get really close to my object and it will still have lots of detail.

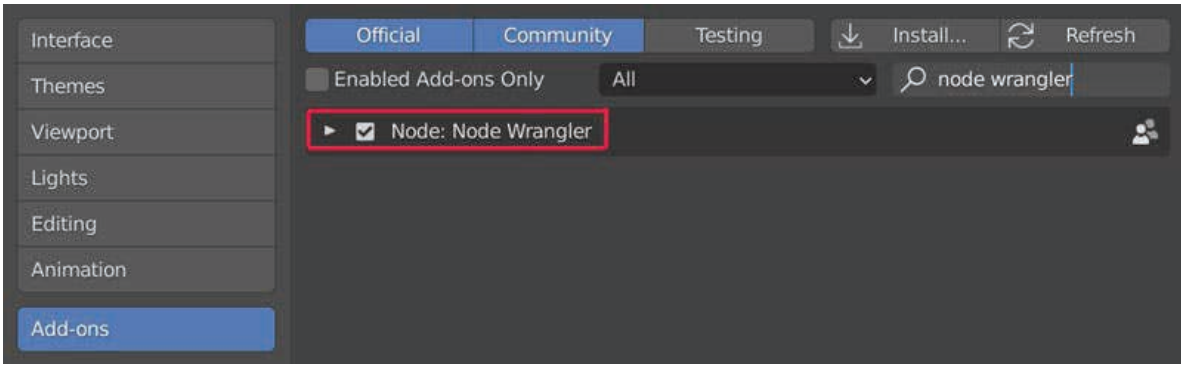


Higher realism through imperfections

It's easy to create a nice glass shader in Blender, especially when using Cycles. However, we can often see that it's fake because unlike real glass, it's 100% perfect. In the real world, glass is not actually perfectly flat, causing light to be reflected at ever so slightly different angles across its surface.

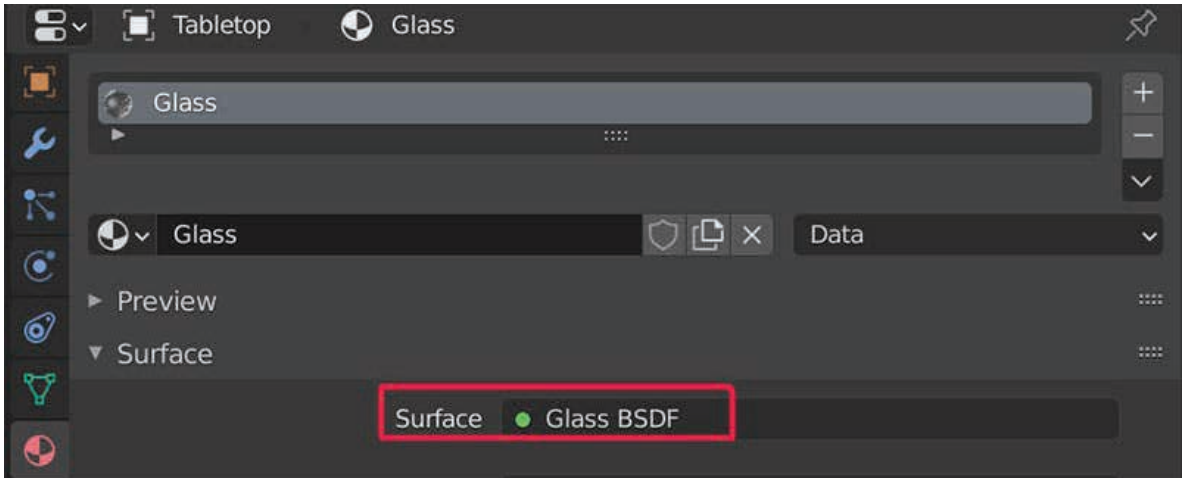


On top of that, it's not going to be perfectly clean, especially if it's an object that gets touched by hands, which makes the touched parts slightly less shiny. If you add these two elements to your glass shader in Blender, it will look better both in Cycles and Eevee. For this tutorial, I'm using the **Node Wrangler add-on** that comes with Blender by default. It merely needs to be activated in **Preferences**.

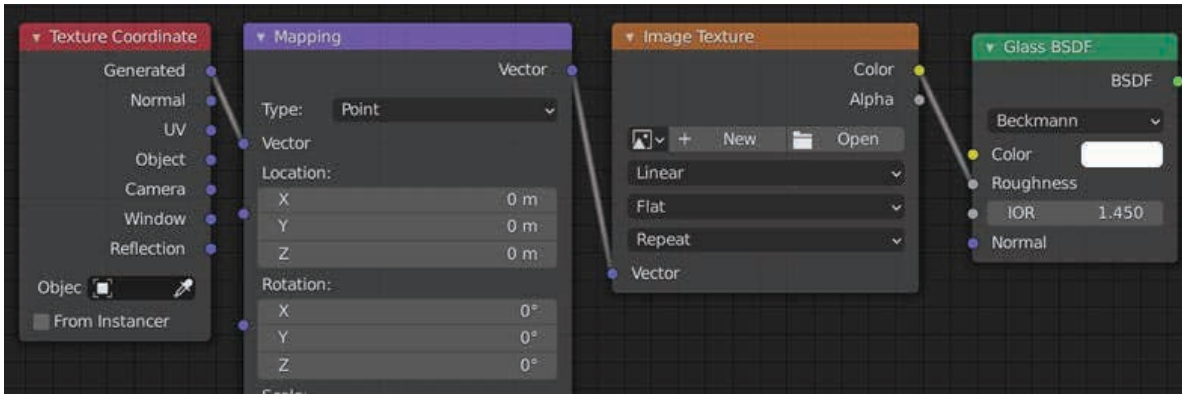


First, select the object that should be made of glass. In this case I've modeled a simple table with a glass tabletop.

With the object selected, add a new material and name it something like “Glass”. Choose the **Glass BSDF** as the material for the surface.

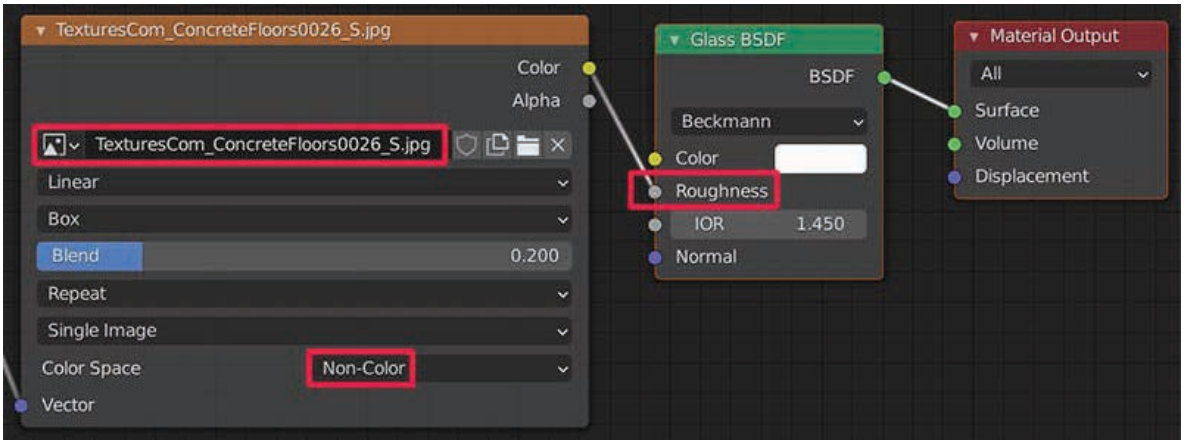


In the **Shader Editor**, select the **Glass node** and **press Ctrl+T** to add the Image texture, Mapping and Texture Coordinate nodes (this won’t work without Node Wrangler being enabled, though of course you could add these nodes manually one by one).

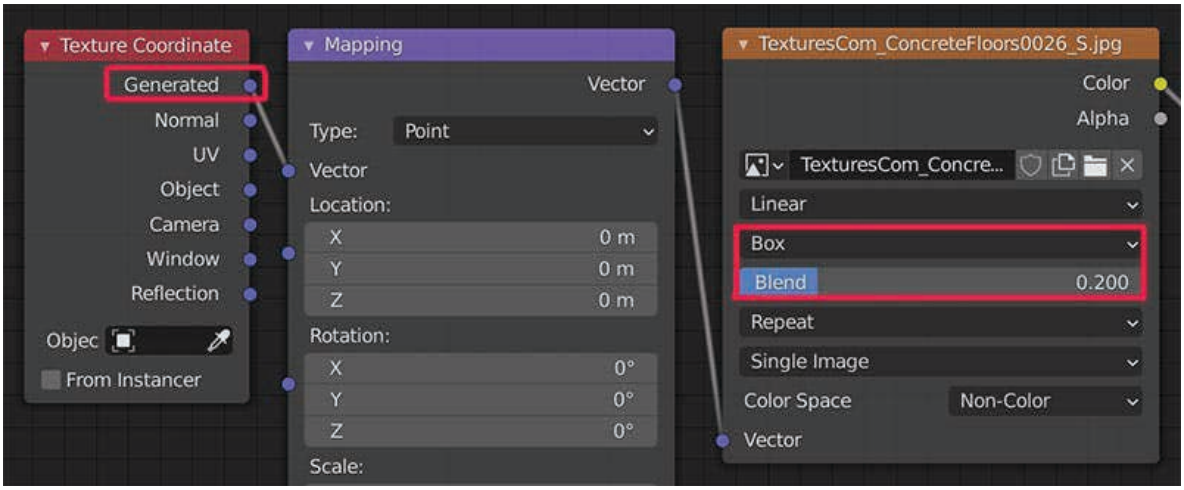


Add an image texture that you think might work well for surface imperfections. In this case I used a free photo of a concrete floor, "TexturesCom_ConcreteFloors0026", from Textures.com.

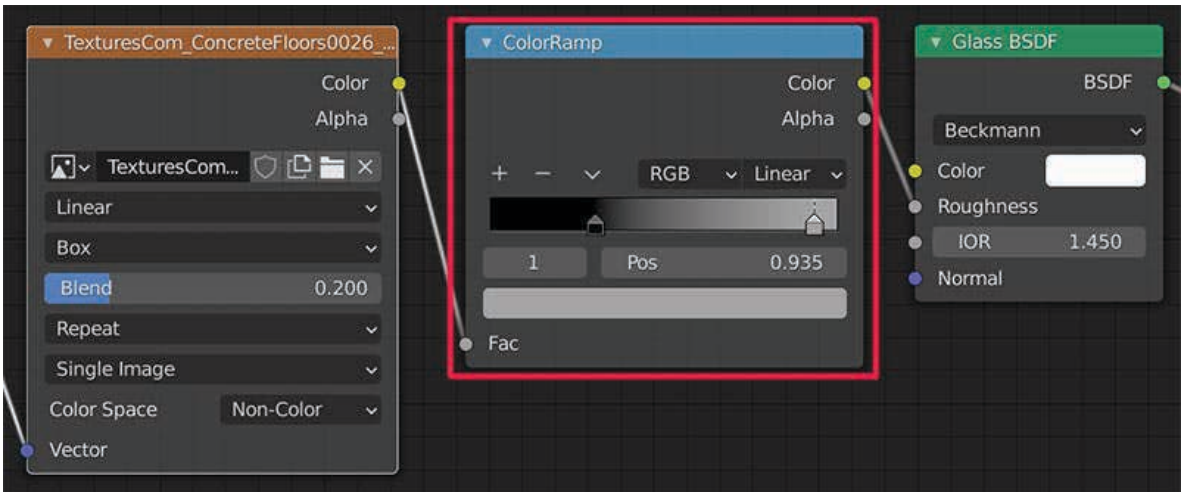
You can use something else, as long as it looks a bit random and organic. Set it to "**Non Color**". Plug it into the "**Roughness**" input of the Glass BSDF node.



For the **Texture Coordinates**, you can use "**Generated**" and in the Image texture node set the **projection type** to "**Box**", with a **Blend value of 0.2 or 0.25** - this makes sure the seams from the box projection are blended together.



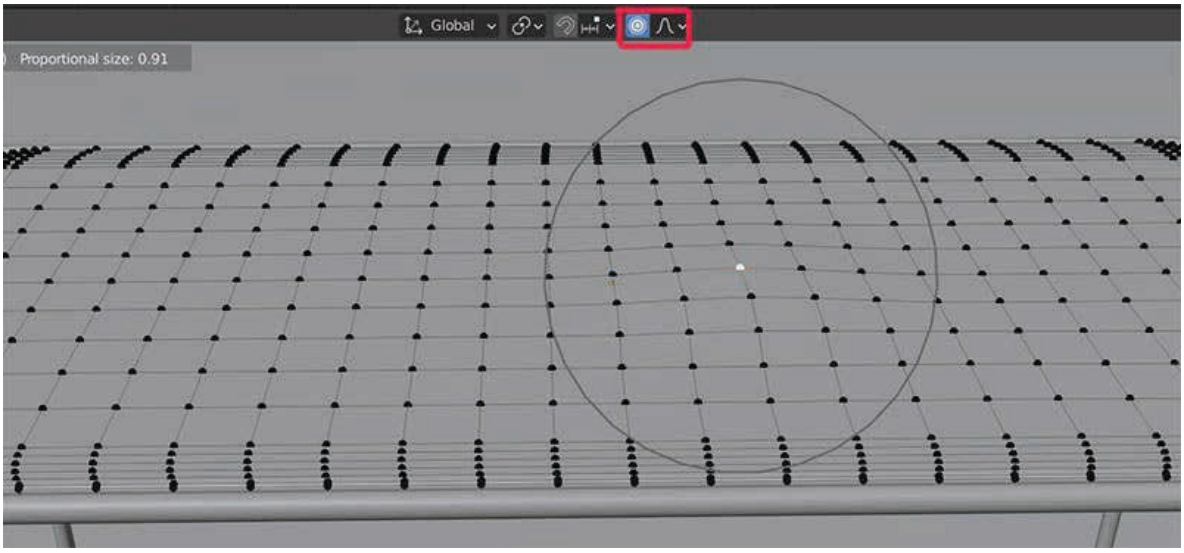
You can add a ColorRamp node to have more control and fine-tune the effect. You can do all this in Eevee so you can see what you're doing, and then go back to Cycles later to render the image for more realism.



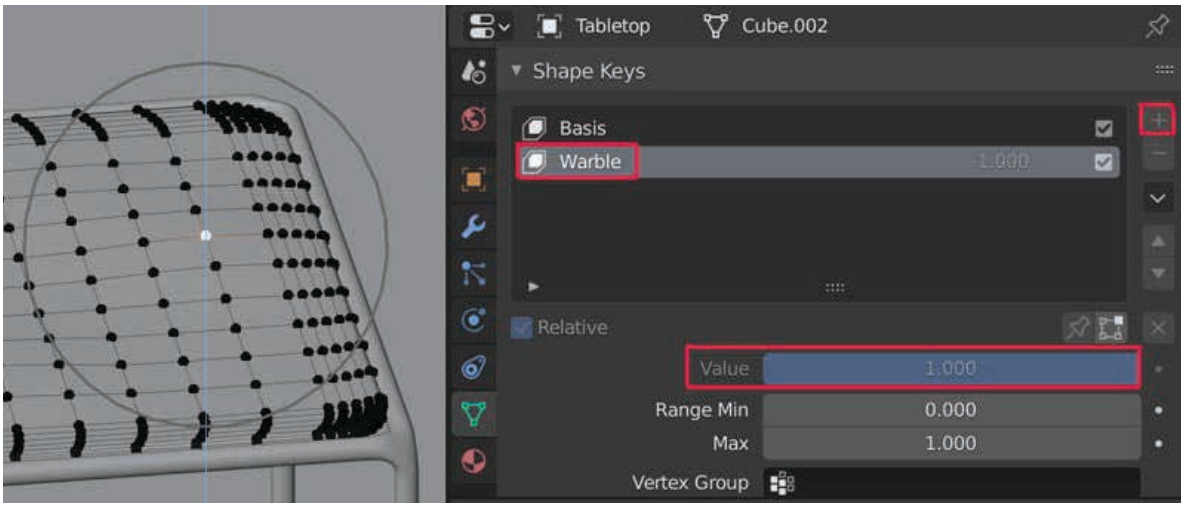
With **Node Wrangler** enabled, you can always isolate one node by **Shift+Ctrl+Left-clicking** on it, to see exactly what it's doing. This can be useful with the **ColorRamp** for example. To get back to the full shader preview, just **Shift+Ctrl+left-click** on the **Glass node**.

Now you get something like the glass in the image below. Mind you, this is exaggerated for the purpose of this tutorial - if your table is this dirty, please stop what you're doing and start cleaning your place :-) Normally tabletops (especially in Architectural Visualisation renders) aren't quite this dirty. You can dial the effect in as needed.

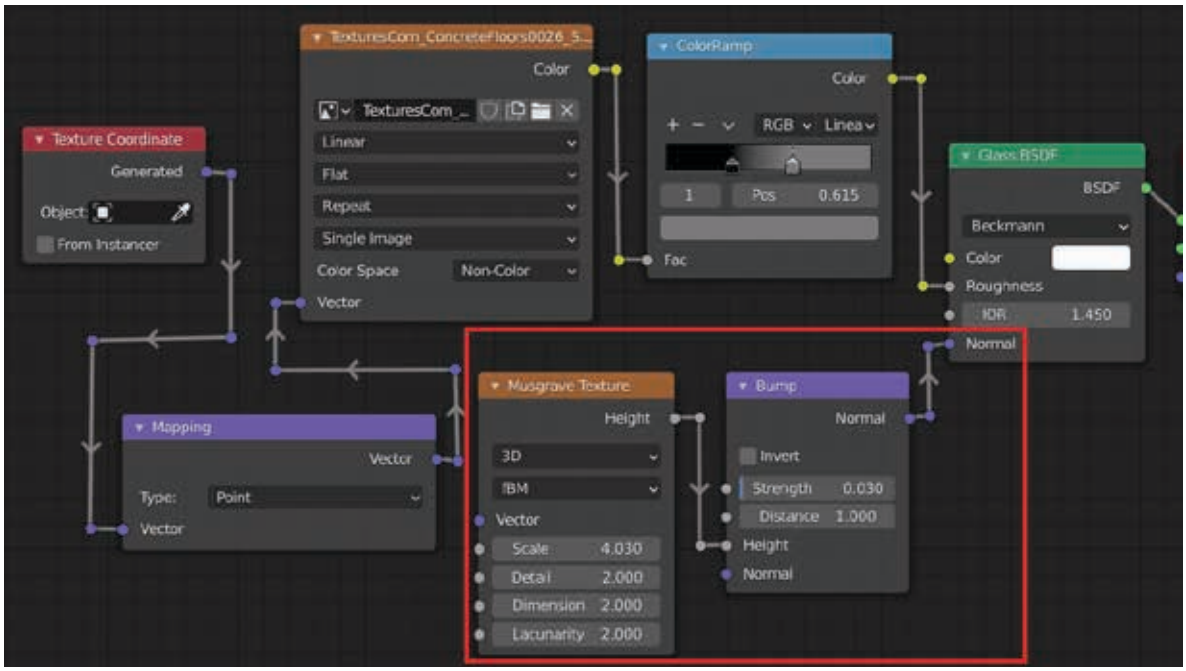
For the imperfect shape of the surface, I simply enable **Proportional Editing** (keyboard shortcut: o) and pull / push some vertices around. The trick is not to go too far!



If you want to make that a bit less of a destructive method, you can add a Shape Key, set it to 1 and do the Proportional Editing on that Shape Key, then afterwards you can still dial it back if you notice that the effect of the warbling is too much.



If you don't want to do any modeling using the Proportional Editing method (which is kind of destructive, after all) you can also use a **Bump node** with a **Musgrave Texture** to create some "warbling" of the glass. You'll need to experiment with a pretty low **Strength** value of the Bump.



If you want to download this table with the Glass Shader, you can find it here.

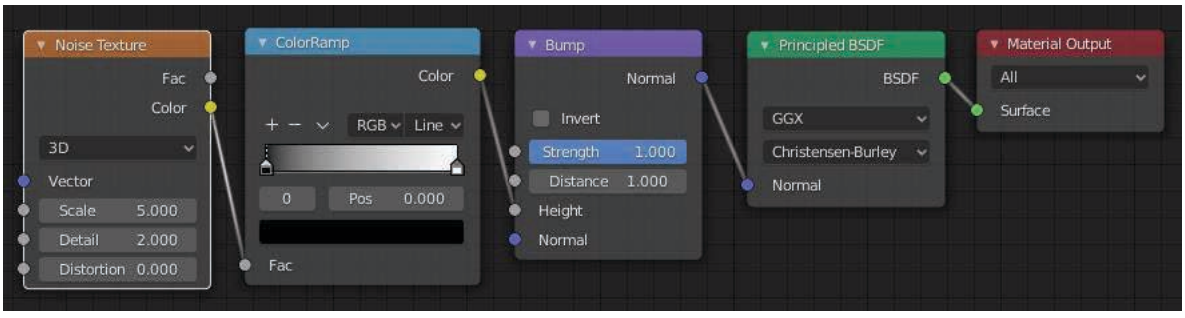
Procedural bump maps

Bump maps always make materials better, and luckily Blender comes with some built-in procedural noise textures that make for nice bump maps.

For this example I'm using a basic UV Sphere with 3 levels of subdivisions.

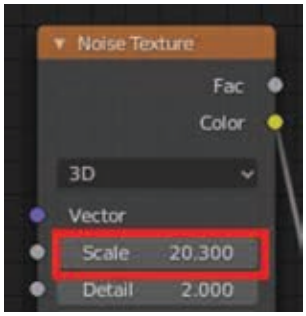
Below is the basic set-up, assuming there are no other maps (like an image texture). The Principled BSDF has been minimized with the **Ctrl+H** shortcut. On the left is the **Noise Texture Node**, which is one of the procedural textures in Blender.

You can add nodes in the Shader Editor window by pressing **Shift+A** and searching for the node name. Then just drag the lines in between from output to input to connect them.

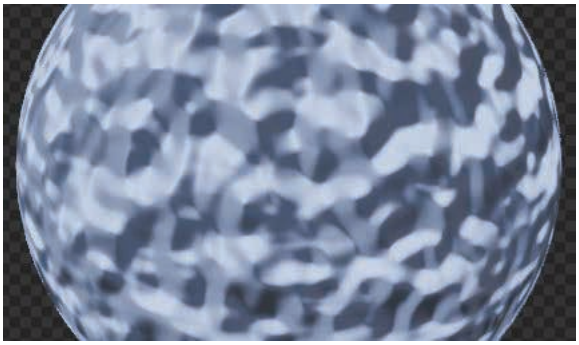


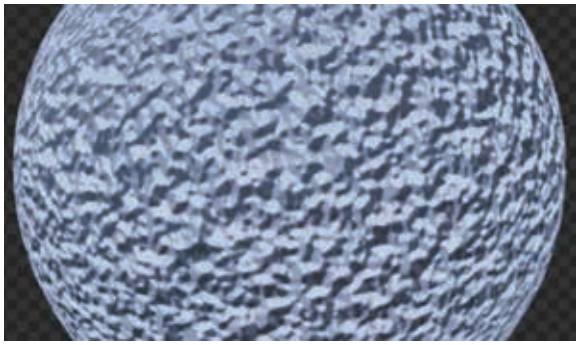
The **ColorRamp** node is added to give us extra control over the Noise Texture.

The **NoiseTexture** has several values that you can use to change its appearance, such as the **Scale** value.

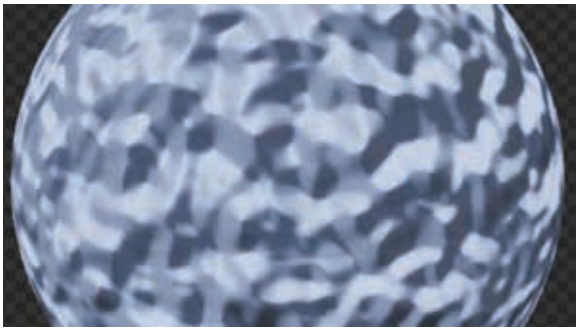


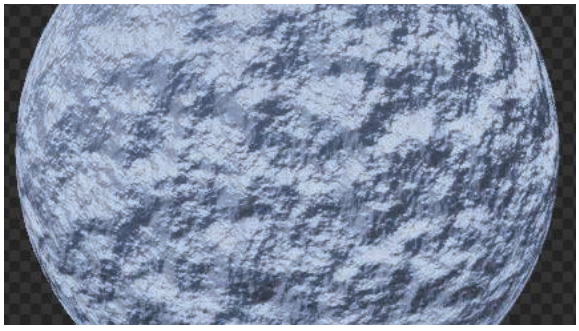
Below as an example of what the **Scale value** of the Noise texture does: on the left is a Scale of 10, right is a Scale of 30.



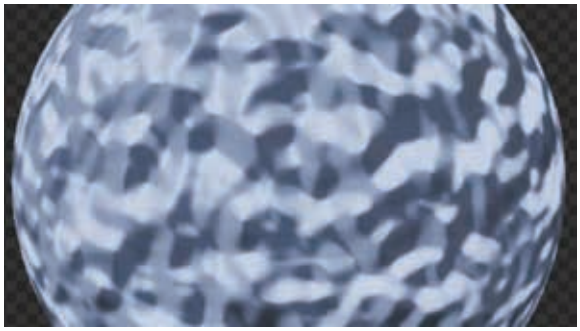


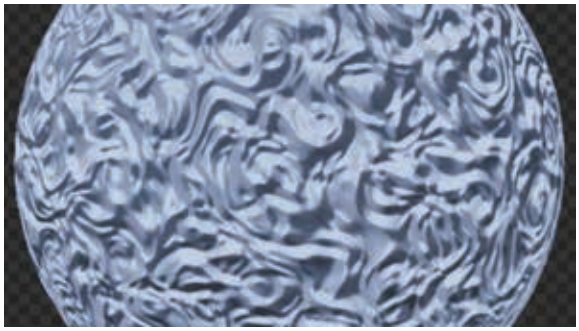
And below is an example of the **Detail value** of the Noise texture: on the left a Detail value of 1, and on the right a Detail value of 100.



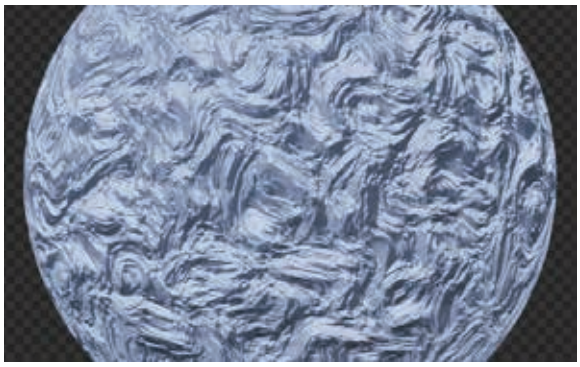


The **Distortion value** is also interesting, creating a swirling effect. Below left is a value of 0, and on the right a value of 4.

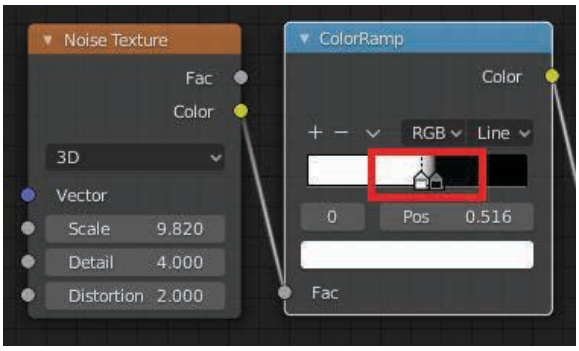




Of course you can combine all these values as well to create an interesting look.



The **ColoRamp** lets us limit the effect, creating a look of scratches on the surface.

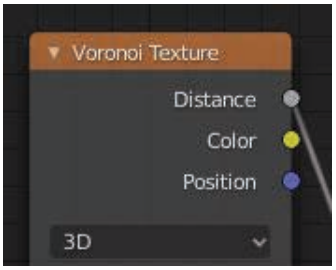


A quick look at some other procedural textures (plugged into the node setup instead of the Noise texture):

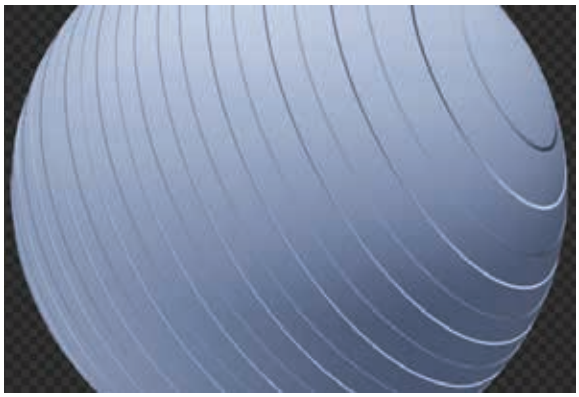
Magic texture.

Musgrave texture.

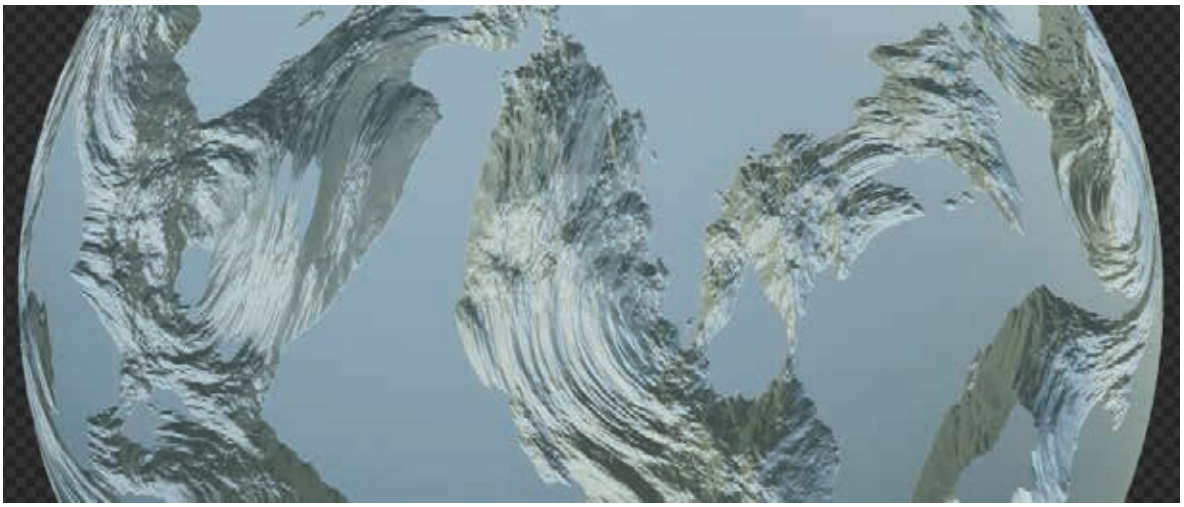
Voronoi texture. Here you can use 3 possible outputs, each with different results.



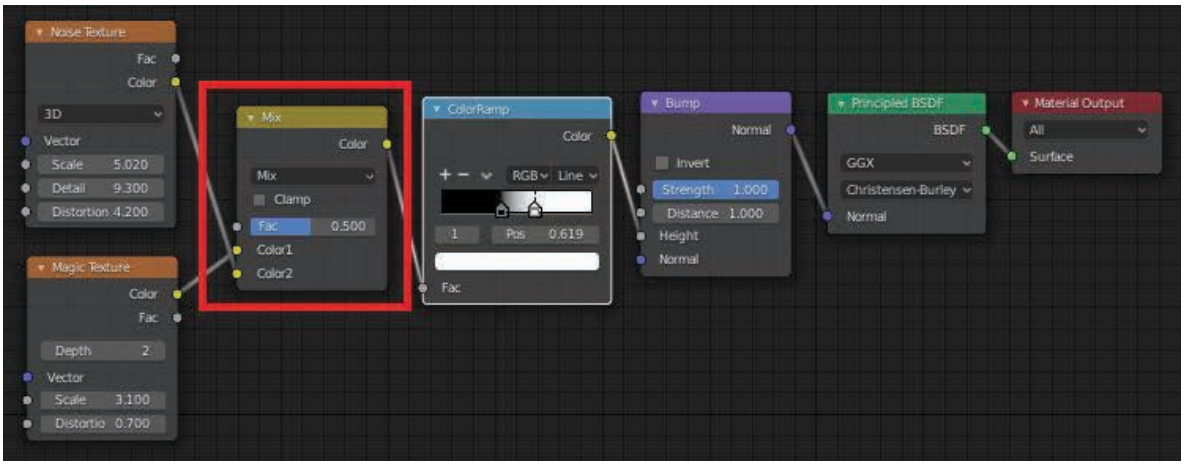
Wave texture.



We can **combine** different procedural textures to create different looks. Here we see a Magic texture and Noise texture combined.



For this we simply combine them using a **MixRGB node**.

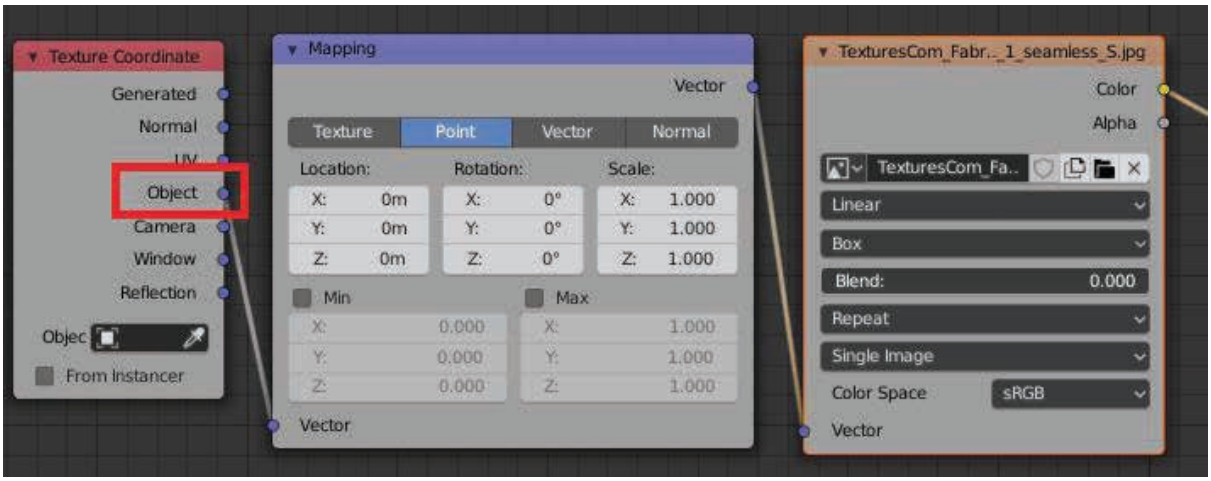


Don't forget that you can **animate** all these values as well for awesome motion graphics!

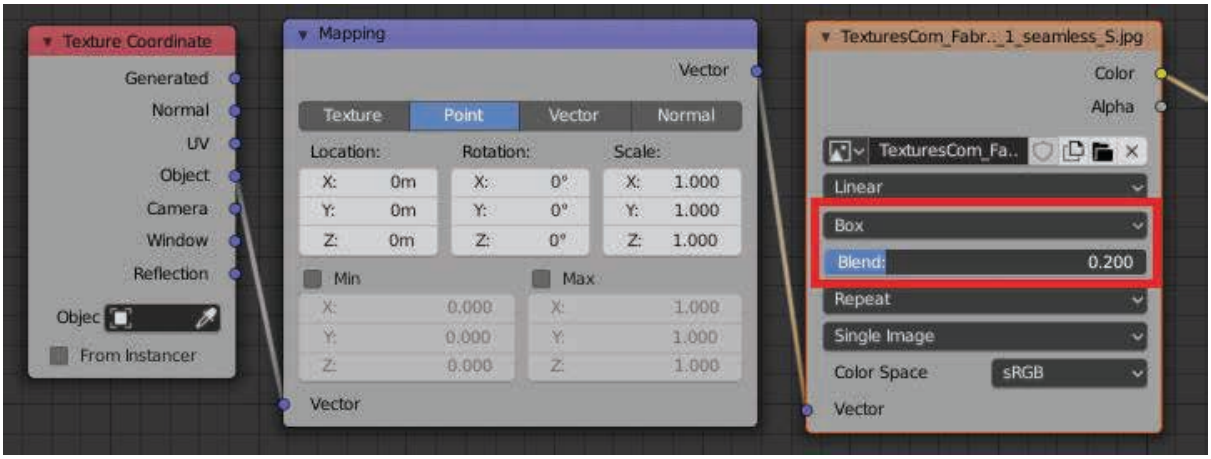
Texturing Without UV-Unwrapping

UV-unwrapping is a pain, let's be honest. What if you could just be lazy and skip that step? You can, in a lot of cases. With box-mapping (sometimes called "triplanar" mapping in other software).

Simply plug these nodes into your **Diffuse BSDF** (or in your Principled BSDF): Texture Coordinate, Mapping and an Image Texture node (with an image texture loaded). The Color output from the image texture goes to the input of the Diffuse or Principled BDSF node.



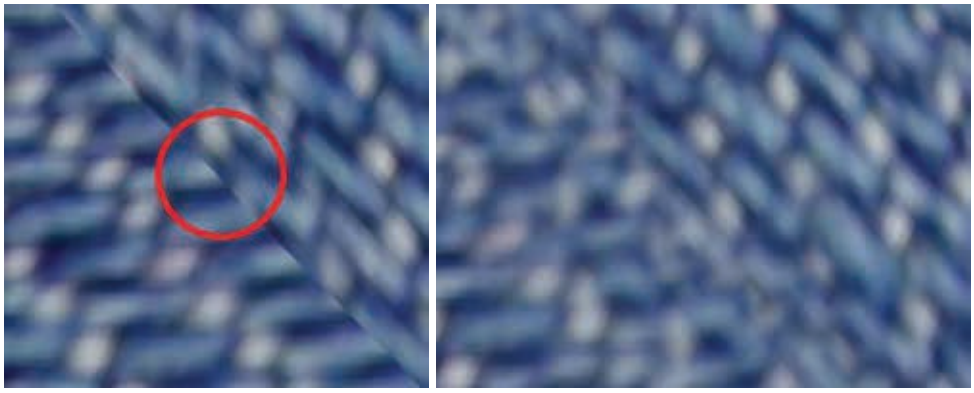
Make sure the Texture Coordinate output is "Object". Normally you would use the "UV" output, but we are skipping the UV unwrap process in this case.Now set mapping to "Box", and Blend value to something like 0.2.



What this does: It projects a texture from the different angles of the mesh, and where they intersect, the Blend-value blends between them.

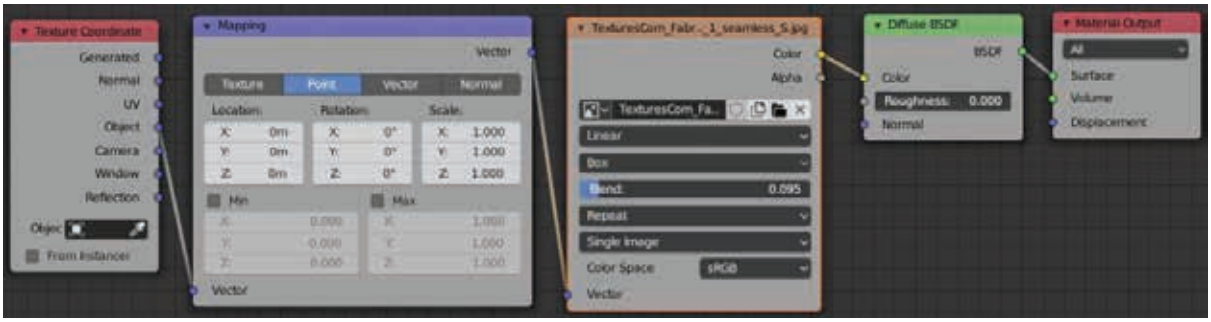
To see what the Blend value does:

Left: no Blend value (set to zero), we see a sharp border where the projected angles meet.



Right: a Blend value of 0.2 (a value that usually works well) gives us this smooth blend where the projected angles meet.

The whole node set-up:

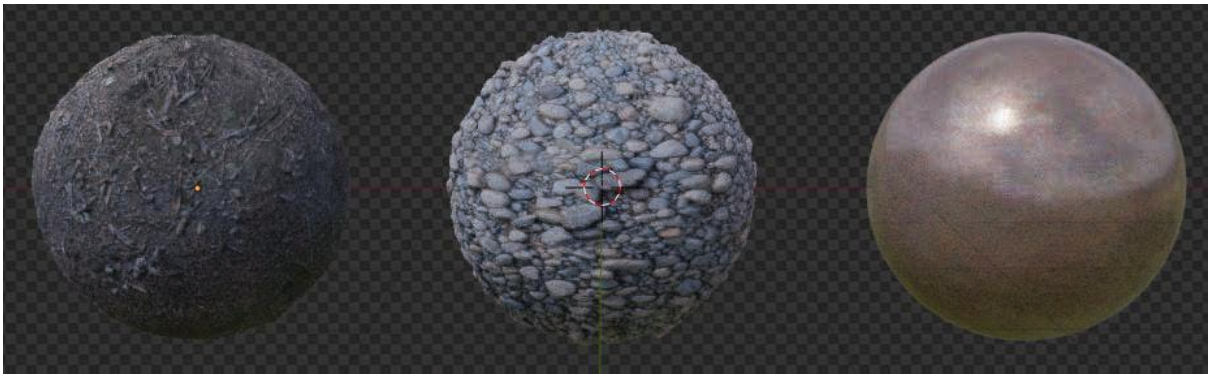


A lantern with a box-mapped texture, and Suzanne the monkey with a jeans-texture. When should you *not* use this method? Well, for example when you are painting a character, you'll want a good UV-unwrap so that the unwrapped texture makes sense when you open it in an image editing program. But for static objects, there's nothing wrong with good old Box mapping. It saves so much time!

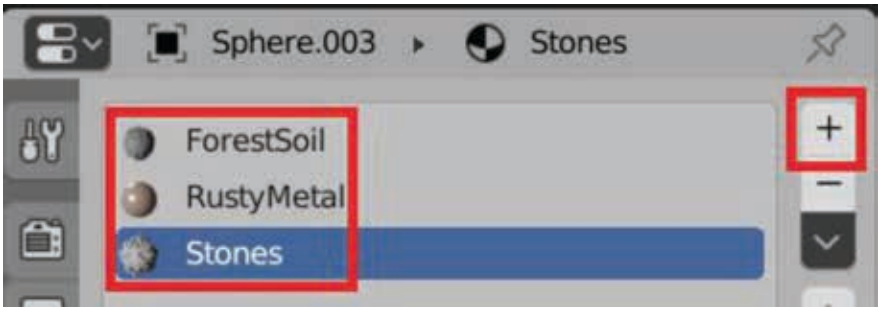
Tip: To save even more time, enable the *Node Wrangler add-on* that is included by default with Blender. That way, you don't need to set up those nodes shown above: simply select the material in the node editor and press **CTRL+T** to automatically add those other nodes. You'll only have to switch to Box mapping, set the blend value and plug the Object output into the Vector input.

Drag And Drop Materials

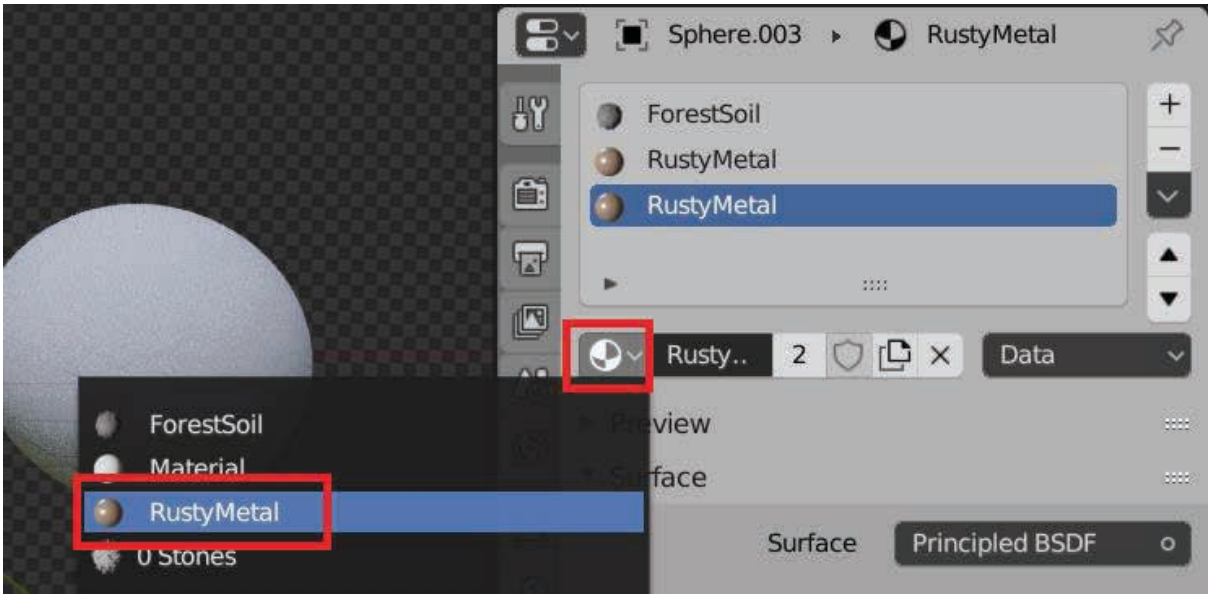
Did you know that you can drag-and-drop materials in Blender?



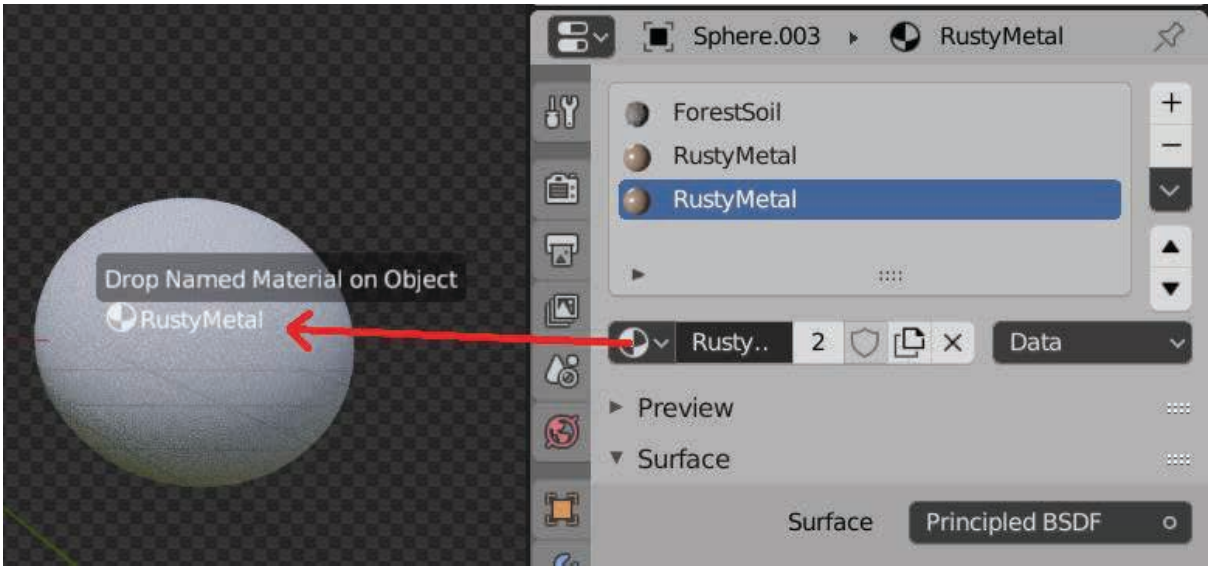
It's a good idea to **name your materials** so you can recognise them. First, put all your materials on one object. Add materials to it by **clicking the + icon**.



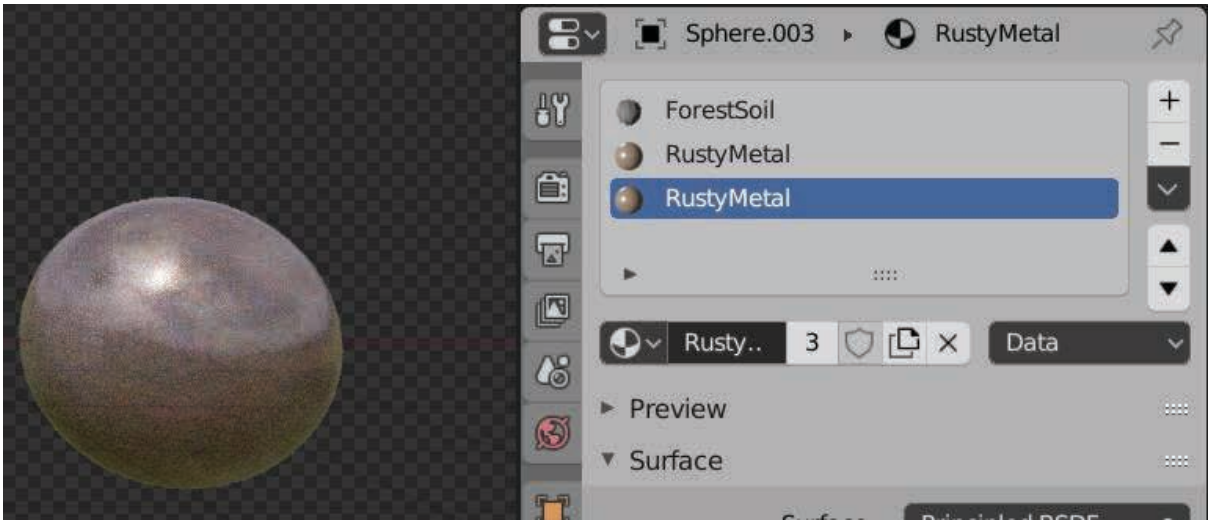
Now, choose a material from the small sphere's material drop-down list. Let's select this **RustyMetal shader**.



Having selected a material, you can now drag it from the icon (surrounded by a red box in the image below) to any other object in the scene. Just left-click on the material icon and drag and drop it on an object.



Now that object has the material assigned to it.



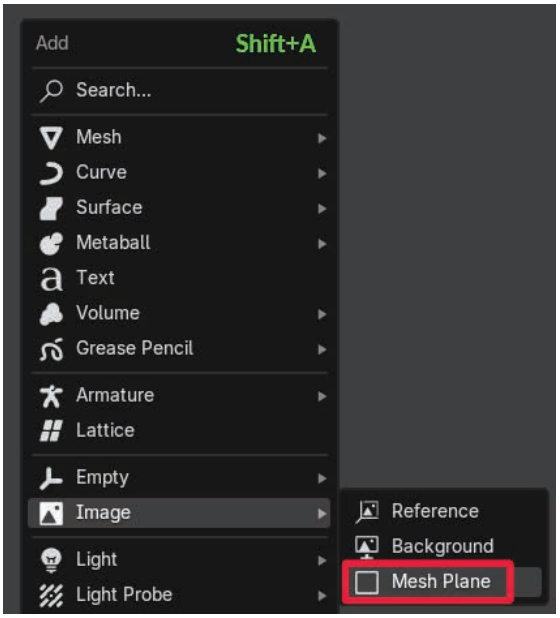
You can select other materials from the small sphere and drag them to other objects in the scene.

Using Shrink Wrap To Project A Sticker To A Surface

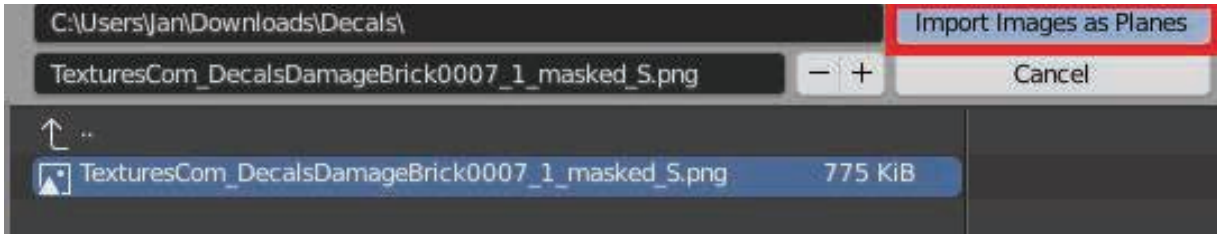
To add a sticker, logo, decal or some other graphic element to an object you can project it onto its surface as a separate object.

In this example I have a sphere and I want to put a decal on it that I've downloaded from textures.com.

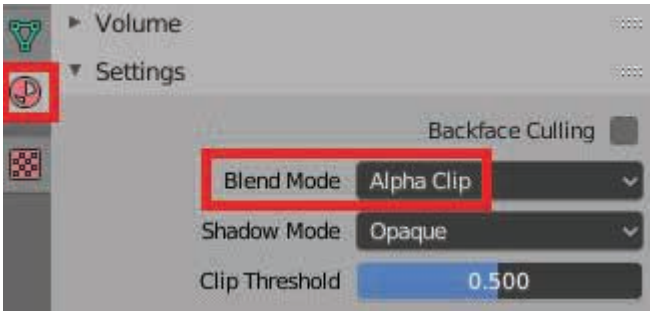
Add the decal image to your scene with Shift+A > Image > **Mesh Plane**.



Browse to your image and click on "Import Images as Planes".

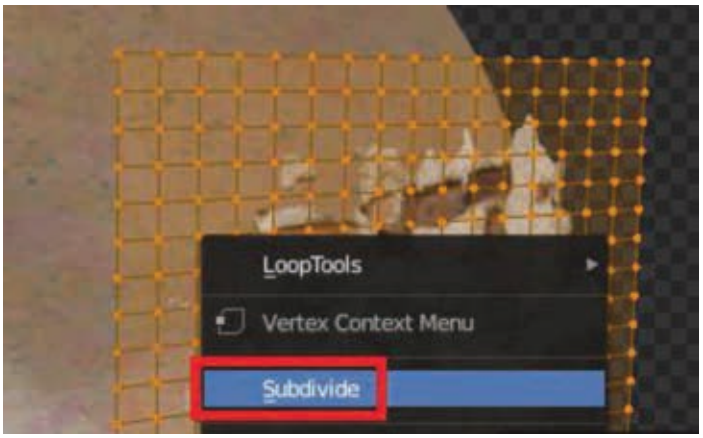


The add-on automatically sets up the nodes. In this case the image has an alpha channel for transparency. In Eevee however, we still need to change the opacity setting of the material. In the Material Settings, set **Blend Mode** to **Alpha Clip**. In Cycles this is not necessary.



Now the image has transparency where it is needed in Eevee.

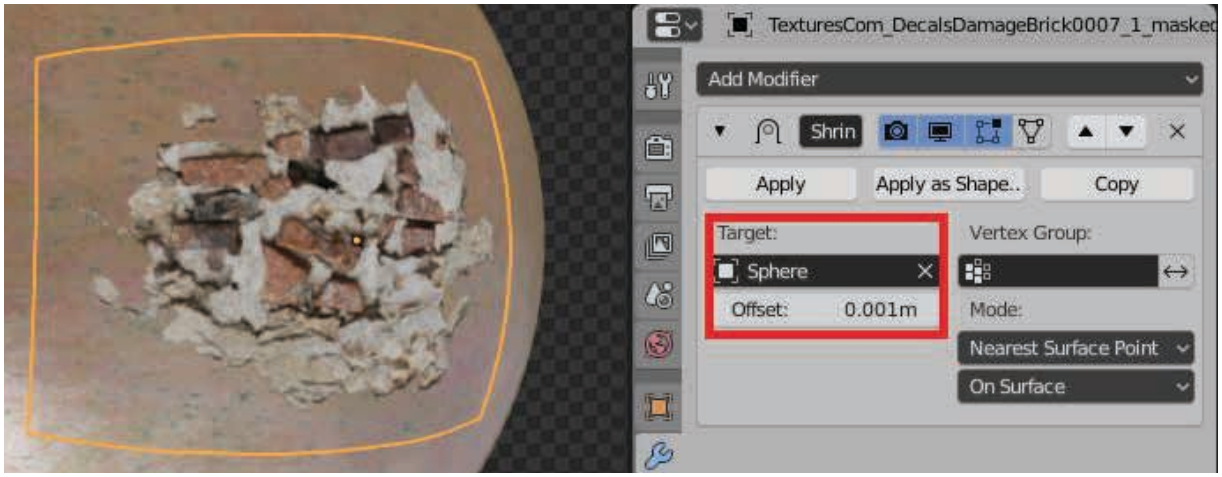
Move it closer to the object you want to project it onto and subdivide it a number of times.



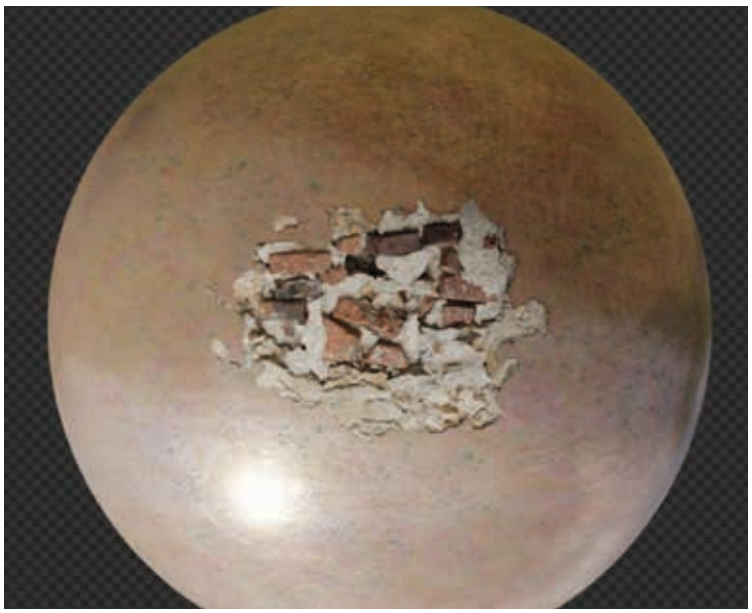
In Edit mode **Right-click** to open the Specials menu and choose Subdivide. Repeat a few times so we have more geometry.

You can also **press Shift+R** to repeat the Subdivide action (**Shift+R** repeats the last action).

In Object Mode add the **Shrinkwrap Modifier** to the sticker plane. In **Target** pick the other object (the sphere in this case). This is what the sticker will be "wrapped" onto. Set a tiny amount of Offset, otherwise the image won't be above the surface and will be invisible. Here I typed "1mm", which Blender converts to 0.001m.



If the sticker is distorted, add some more geometry to the object by adding a Subdivision Surface Modifier to the other object.



That's all there is to it! This can be really helpful if you want to add some details without having to add them to the object's own texture, or even without UV-unwrapping that object.

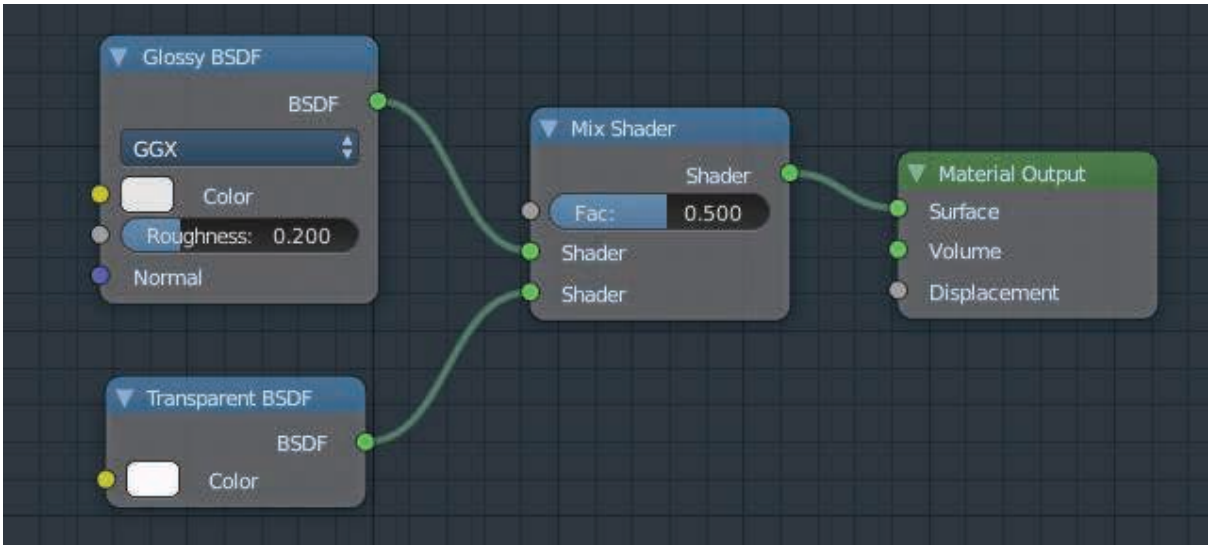
Glass Transparency With Alpha Channel

In this Cycles viewport display of a bus, you can see that an HDRI environment is visible through the glass. Even though the background is set to “transparent”. So why is the glass not transparent?

Well, that actually makes sense in a physically based render engine like Cycles. Glass changes the way you see the background through it. So it has to show the HDRI behind it in this case, on order to be accurate.

Okay, but what if we still want to really have a transparent glass shader so we can composite in a different background later? That’s possible! First, we create a new material for the glass.

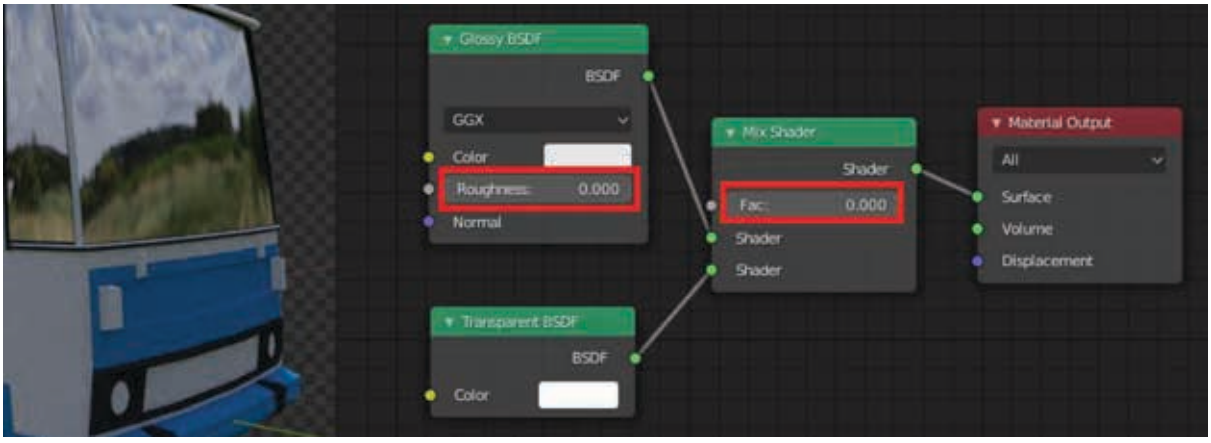
Select the glass object and go to the Shader Editor. Delete the Glass node (or whatever kind of shader is currently on the glass object) by selecting it and **pressing X**. Add a Glossy node, as well as a Transparency node and plug them into a Mix Shader.



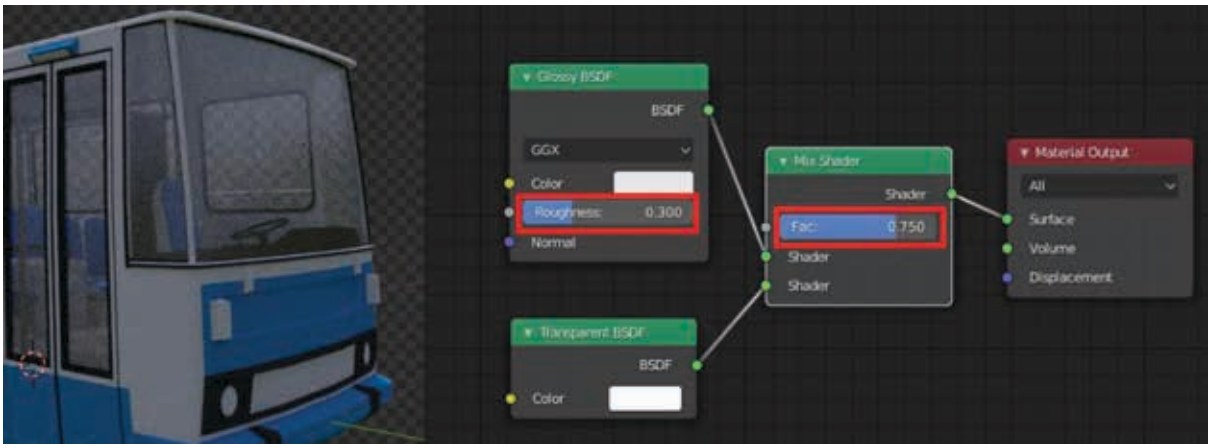
You can easily add nodes by **pressing Shift+A** (the way you would add objects in the 3D viewport) and searching for the name of the node you want.

Now you can see the "transparency" blocks of the background through the glass.

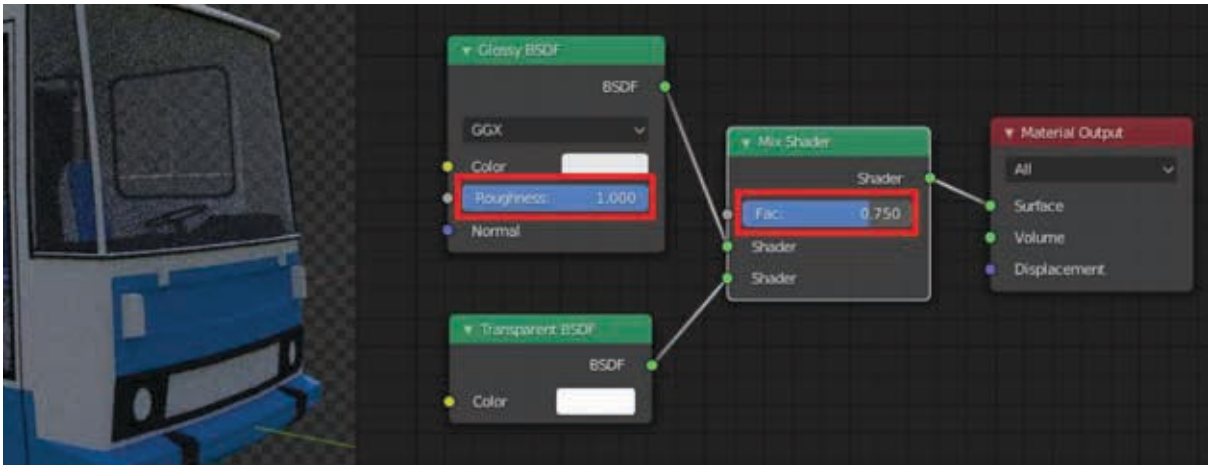
Here are some variations you can make with this shader setup by adjusting the values:



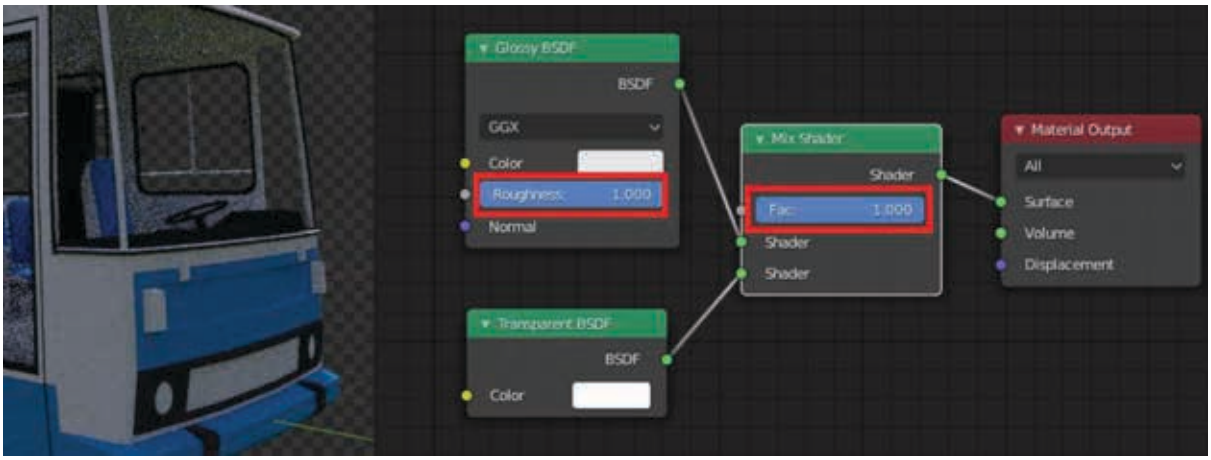
Zero roughness means 100% reflectivity.



A bit of Roughness, and the Mix Shader is balanced towards the Transparent BDSF.

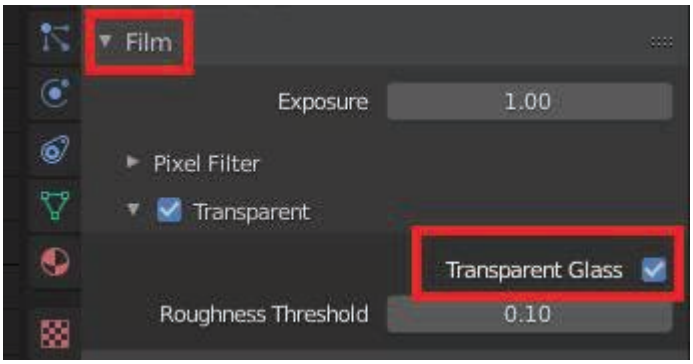


Roughness set to 1, which means there's no reflections at all. Mix Shader is balanced towards the Transparent BDSF.



The Mix Shader is 100% only giving the Transparent BDSF, so the glass is 100% transparent.

There is also an easier method (but with less control) where you just use a Glass DBSF shader and check the "**Transparent Glass**" box under the Film settings of the Cycles tab.

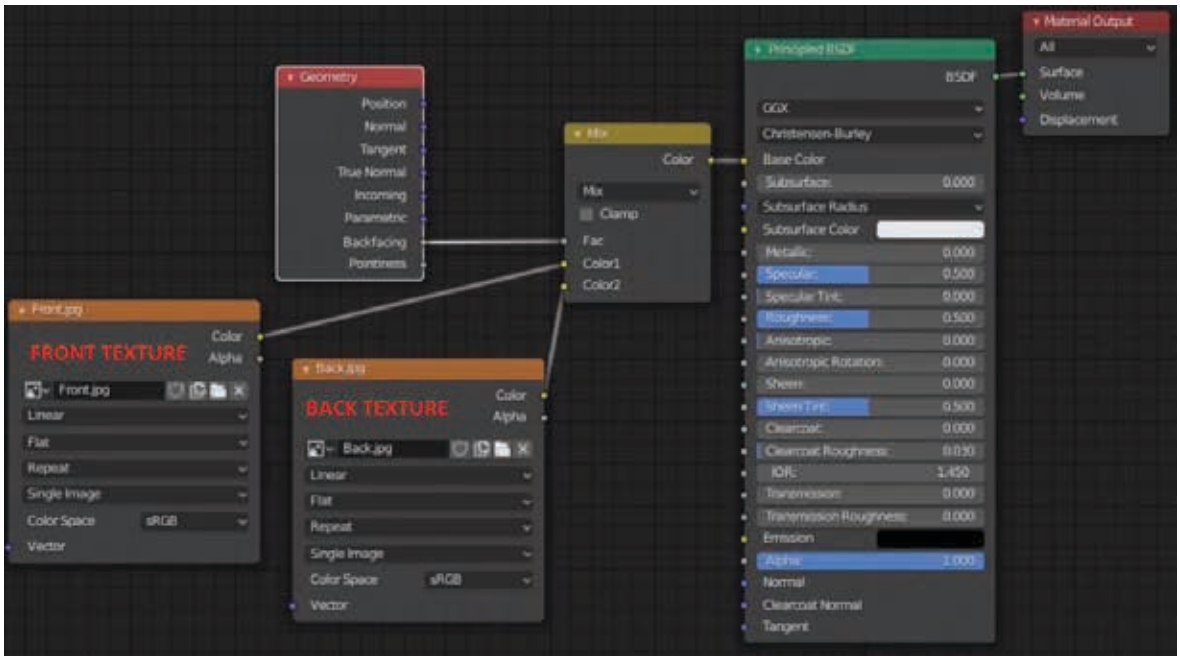


This beautiful bus was modeled by Blendswap-contributor swift502. You can download the model here: https://blendswap.com/blend/16380

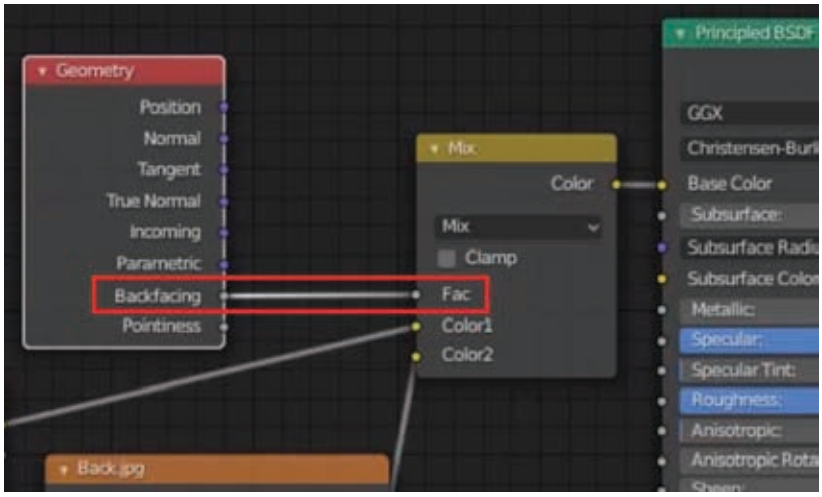
Double-sided material

This is a simple plane with one material, but it has different textures on the front and the back.

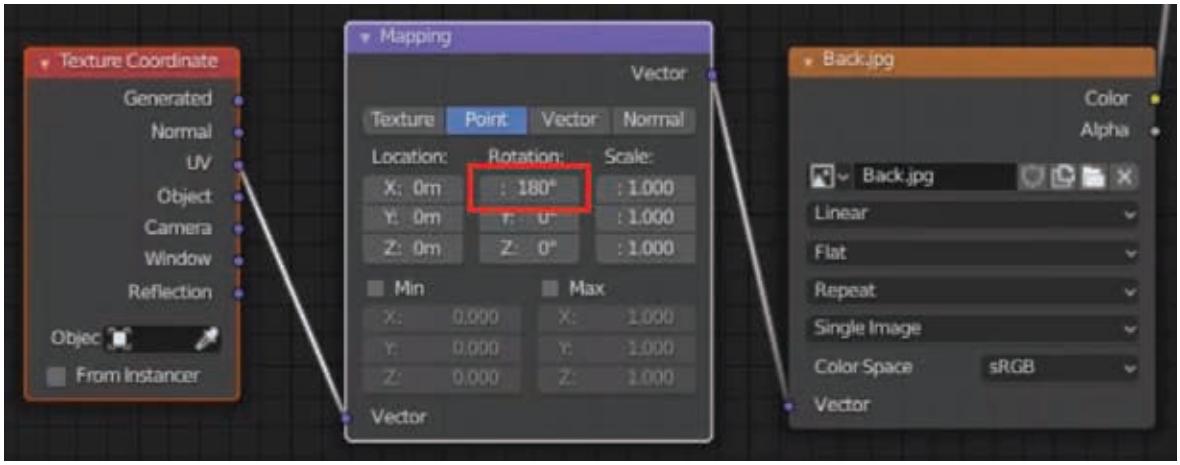
The node setup is simple, first we use two image texture nodes. One texture node for each side of the plane.



They are mixed with a **Mix RGB** node, and the mix factor comes from a Geometry node **Backfacing** output.



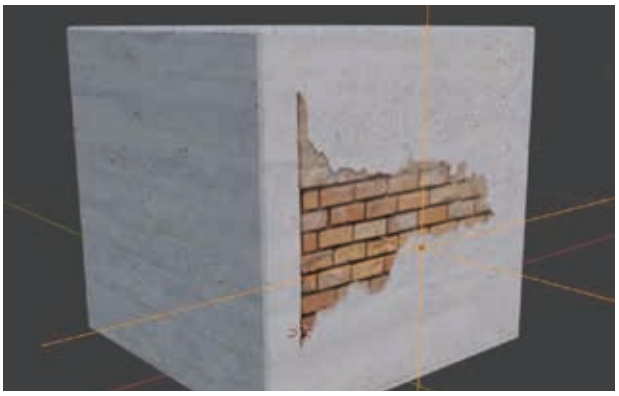
In case you need to flip the texture horizontally, you can add Mapping nodes (**press Ctrl+T** with the Texture node selected). In the below example the Back texture is flipped along the X-axis by rotating it 180°.



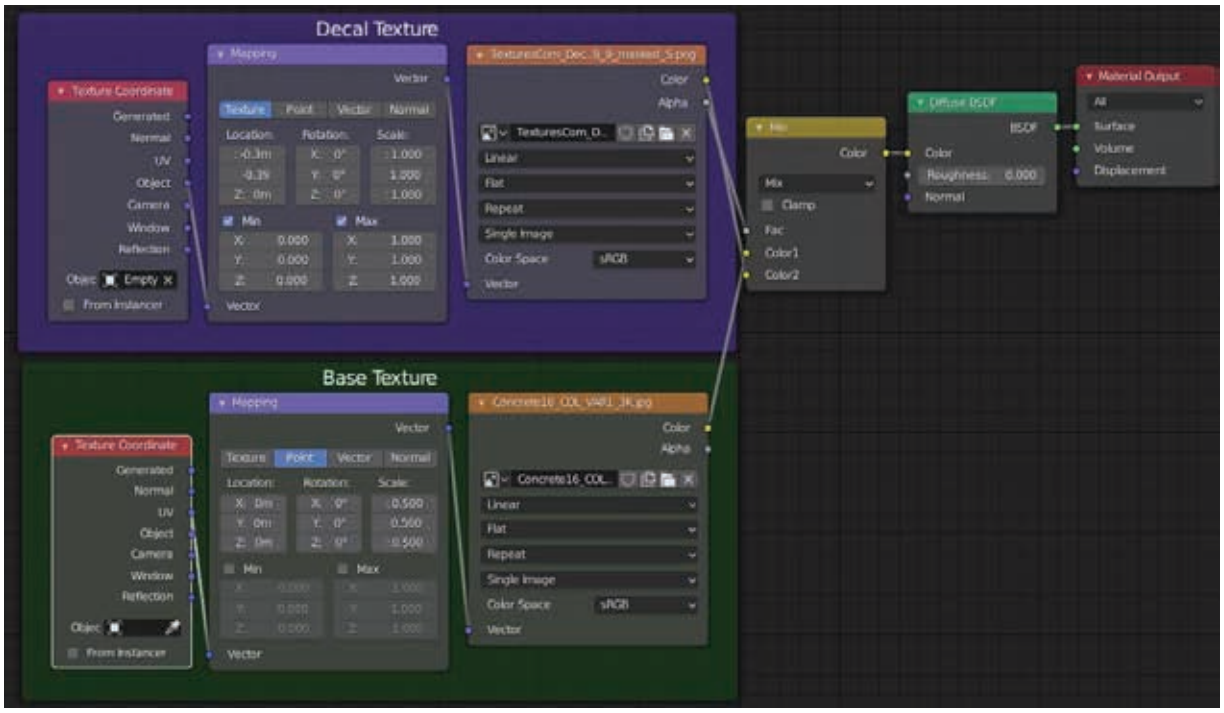
This can be useful when a texture has text on it, so it doesn't show the text mirrored.

Add a decal using only nodes

This is a method for adding a decal without using a method like Shrinkwrap or a separate mesh.



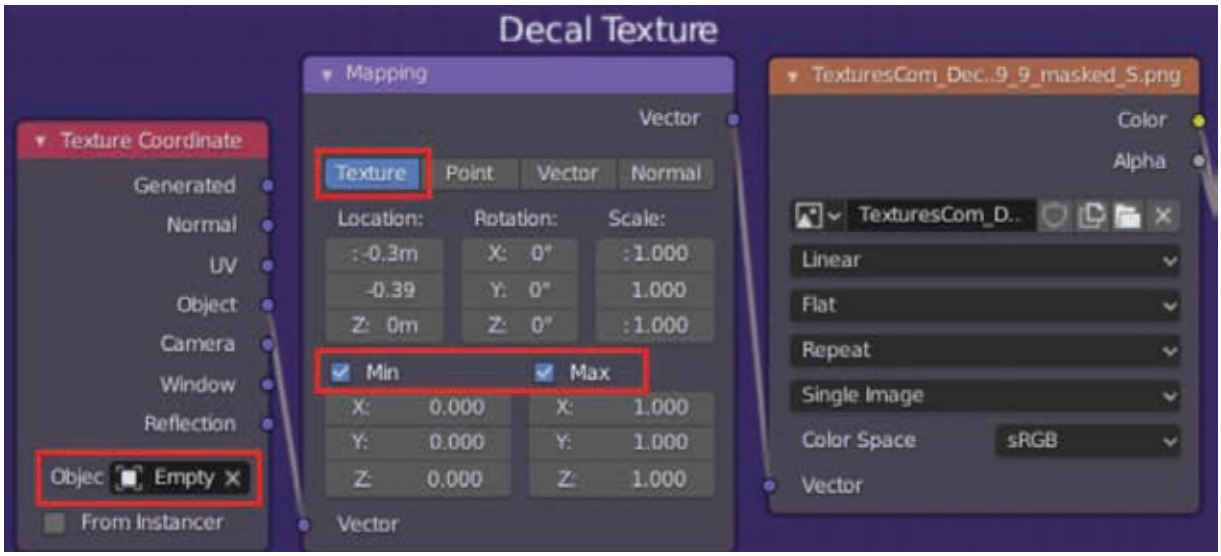
In the example on the right, the cube has a concrete texture, and the bricks texture is added on top through nodes. Here we see an overview of the entire node set-up for this material.



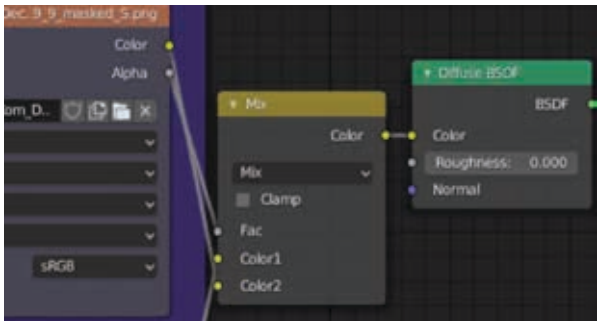
In the green frame we see the base texture, or just the normal texture of the object. Nothing unusual here. Just an image texture.

Click here to return to the index.

In the blue frame we have the decal texture. The mapping node is set to **Texture**.

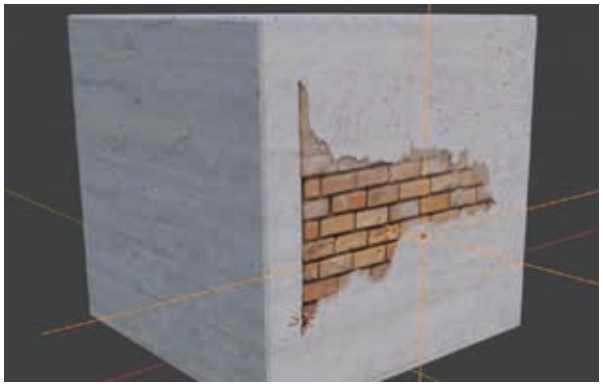


Min and Max being checked makes sure the Decal only shows once. The coordinates node is set to **Object**, and as the object we use an **Empty**.



The two textures are mixed with a **Mix node,** using the Alpha channel of the decal to mix them.

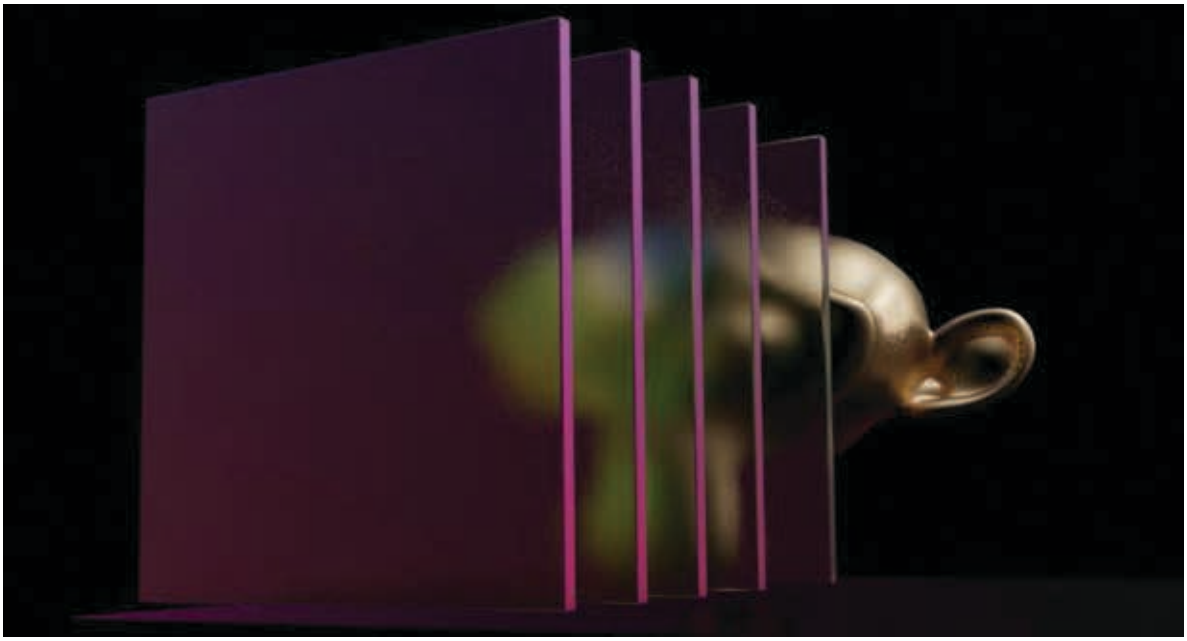
You may have to change the values of the Mapping location, depending on your case. Moving the Empty also moves the decal.



So you can parent the Empty to the object if you want. To do this, select the Empty and then thecube, **press Ctrl+P** and choose “Object (KeepTransform)”.

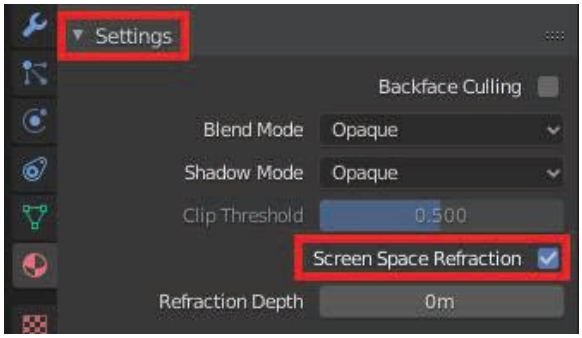
Glass in Eevee

Glass, even with several layers, is possible in Eevee with the following settings.

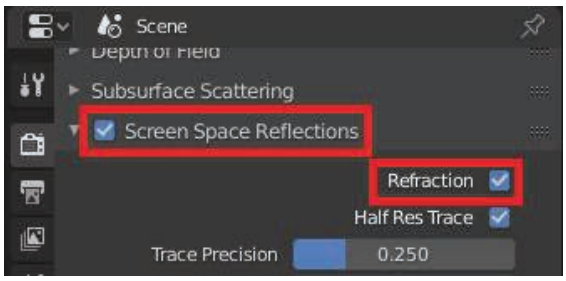


Add a new material to the glass object (in this case a flattened cube) and set **Transmission** to 1.

In material settings turn on **Screen Space Refraction**.



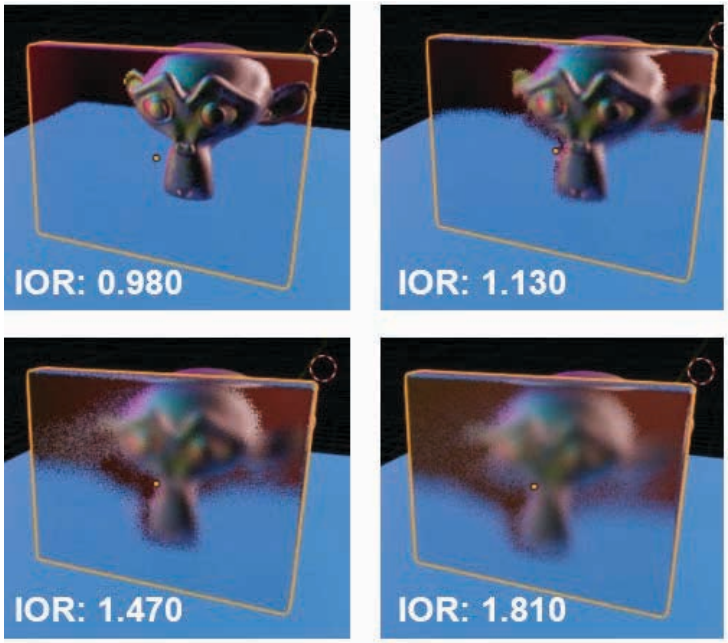
In the Render tab, turn on **Space Reflections** and **Refraction**.



Click here to return to the index.

You can experiment with the index of refraction (IOR) value and see the result in real-time.

Some example IOR values:



Materials all have their own Index of Refraction value. You can find these values on the Internet (for example on Wikipedia) to add realism to your renders.

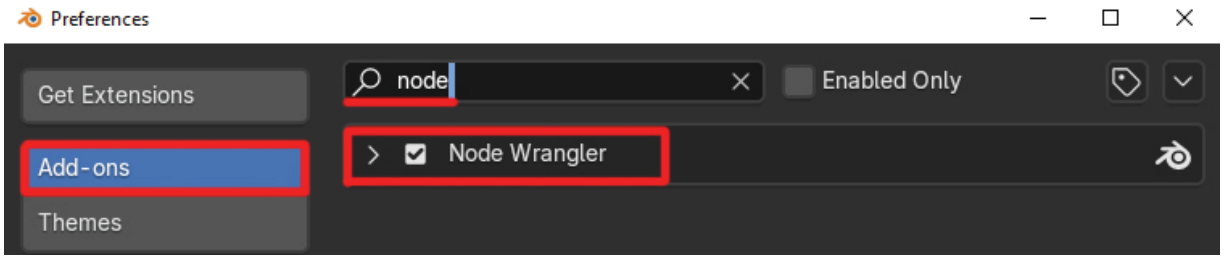
Below you'll find some examples of IOR values:

Water	1.333
Olive oil	1.47
Ice	1.31
Glass	**1.52 (though 1.450 is often used)**
Diamond	2.42
Clear plastic	1.400

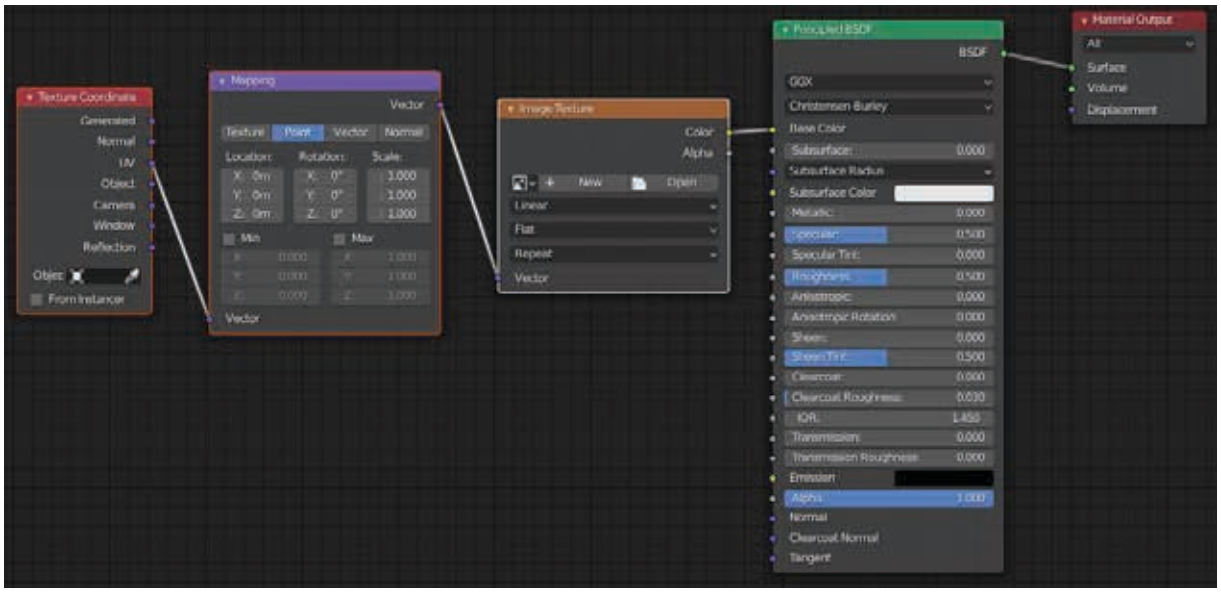
Quickly Add Material Nodes With Node Wrangler Add-On

Node Wrangler is a must-use addon that will save you a lot of time when making Shaders and adding textures to your models.

To use it, make sure **Node Wrangler add-on** is turned on in **Preferences** .



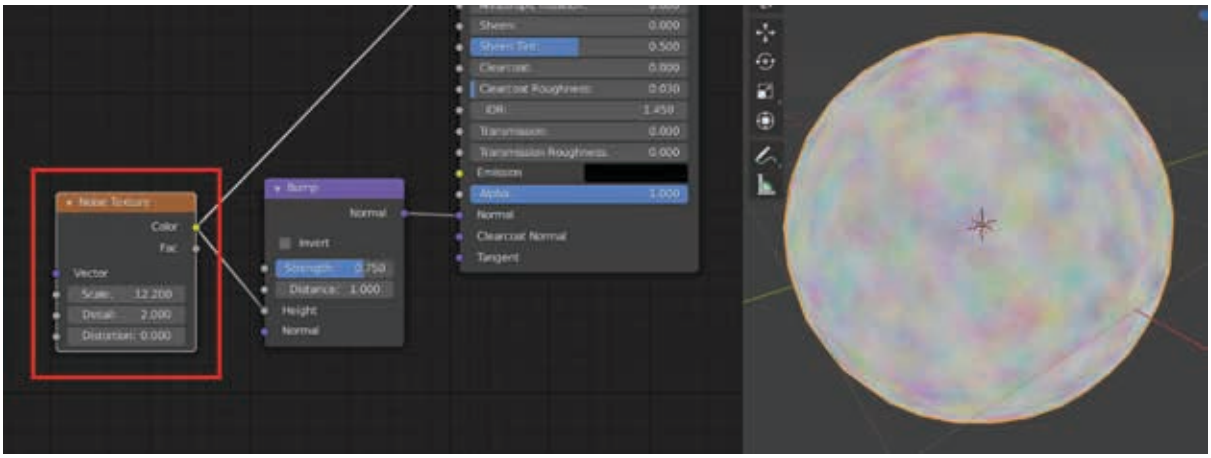
Select your material node (the Principled BDSF in the below example) and **press Ctrl+T**.



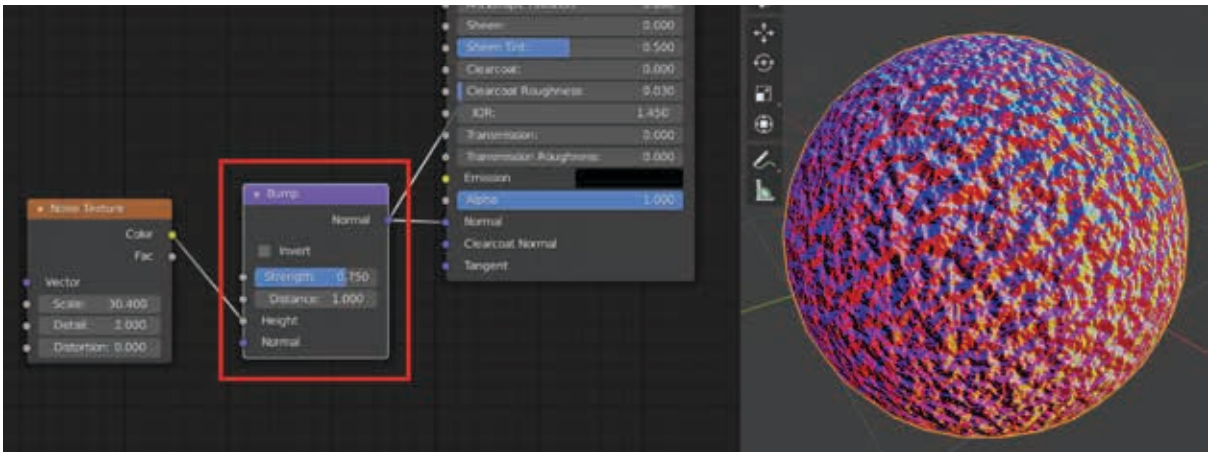
This adds the extra nodes you usually need for materials: Texture Coordinates, Mapping and Image Texture. Select and **Press G** to move the new nodes if necessary.

Another useful feature the Node Wrangler gives you:

Holding Ctrl+Shift and clicking on a node will preview just that node. Useful when adjusting the values of a Noise texture for example. Below you see that (in LookDev view) you see the noise pattern on the object because the Noise Texture has been temporarily connected to a Viewer node.

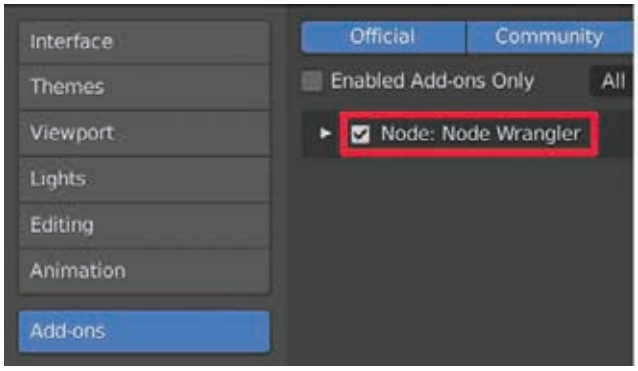


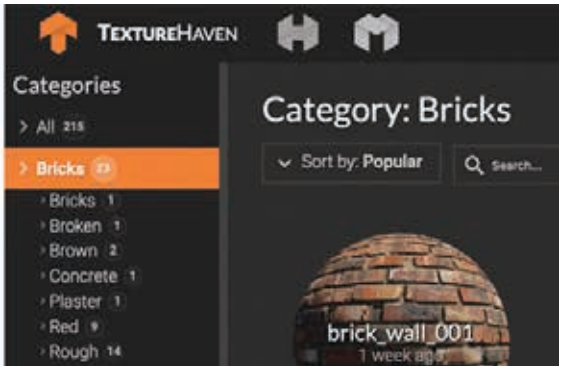
And here, the **Bump** node has been temporarily connected to a Viewer node, showing the bump (as a normal map). This way you can quickly and visually adjust individual nodes.



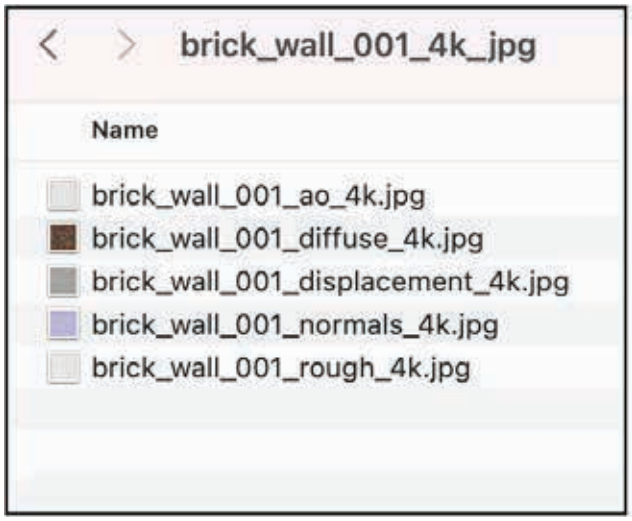
Easy PBR Textures With Node Wrangler

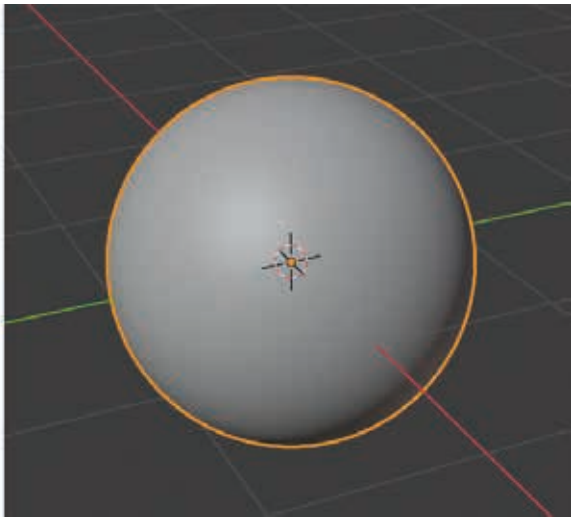
First, make sure Node Wrangler is enabled. Then find some PBR maps, for example on Texturehaven.com. Let's download "brick wall 001". I chose the 4K version.



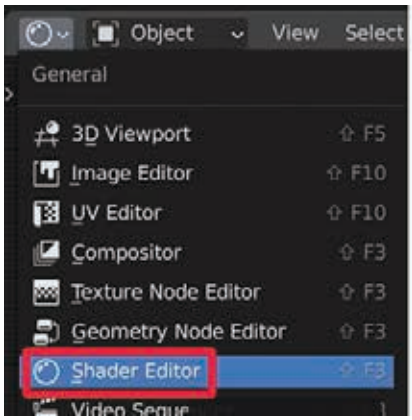


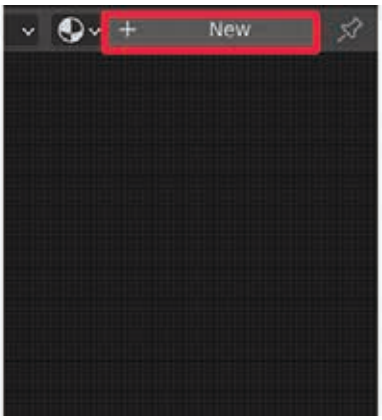
After unpacking the downloaded ZIP file, you should have a number of images in a folder. Let's test it with a **UV Sphere**. Give it a Subdivision modifier so there's some geometry for the displacement texture, and set it to Shade Smooth.

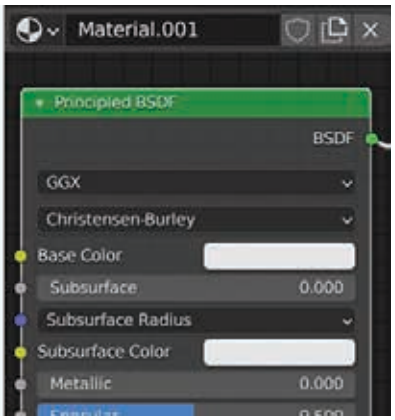




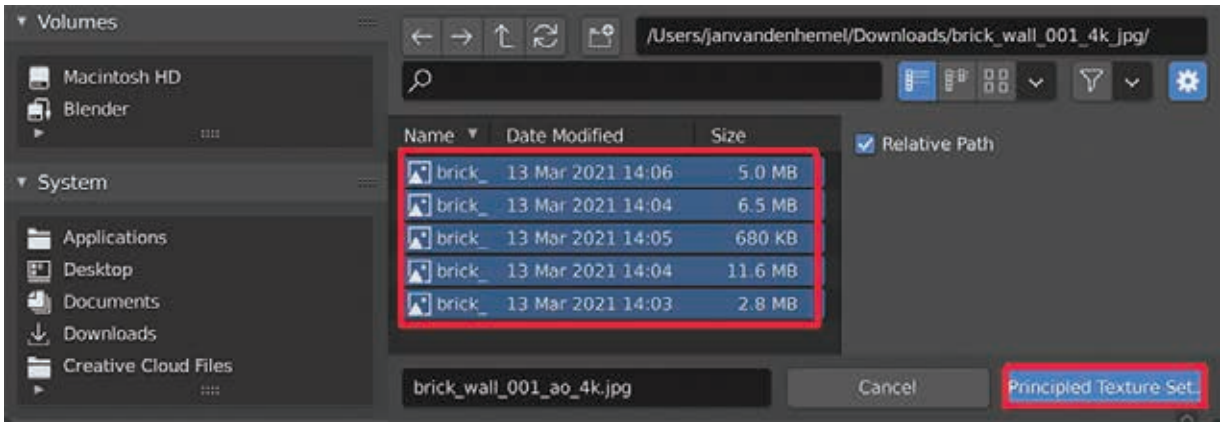
Open a **Shader Editor Window**. Create a new material and select the Principled BSDF node. **Press Shift+Ctrl+T**.



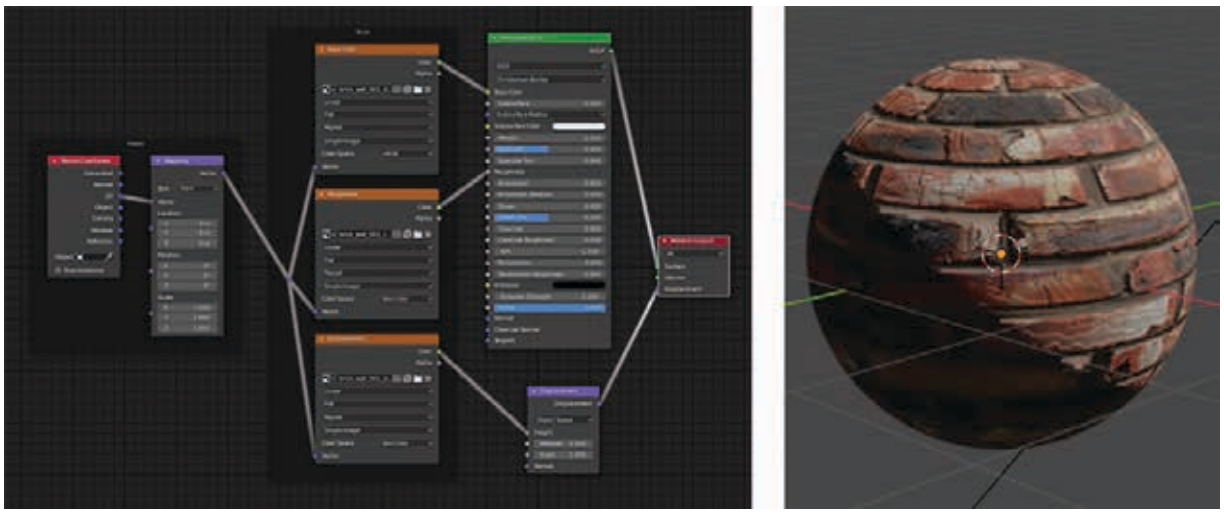




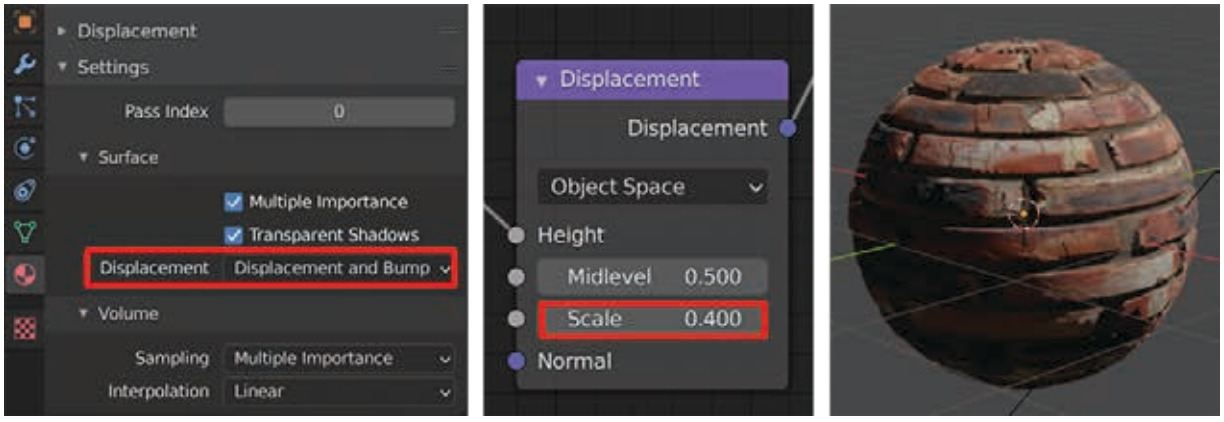
This will open a window where you can find your maps. Select them and **click on Principled Texture Set**.



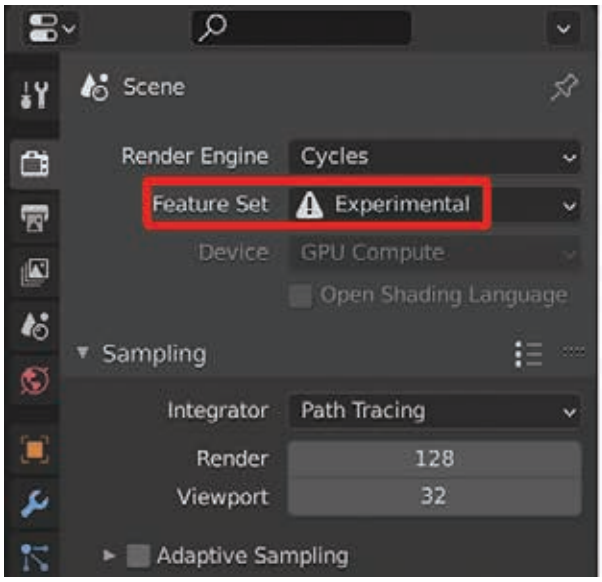
That's it! All your nodes are set up automatically. (Note that I've added a Sun lamp in the scene and switched to Rendered preview in the viewport)

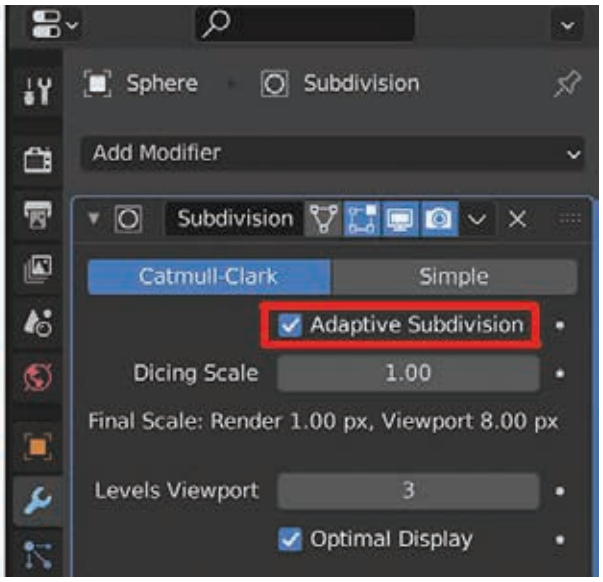


If you don't see Displacement in Rendered view, make sure you are in Cycles and that the material is not set to only Bump. You can change the Scale of the Displacement as needed.



Remember to use Cycles **Experimental Features** and **Adaptive Subdivision** for best Displacement results - this makes sure things close to the camera get more subdivisions, and things further away get less.





Keep your PBR textures in a separate Blend file, give them a clear name, then later you can append them when needed. It's worth noting that when using JPEG maps, the lack of bit depth will be visible in the displacement map up close. For close-up displacement, PNG maps are a better choice.

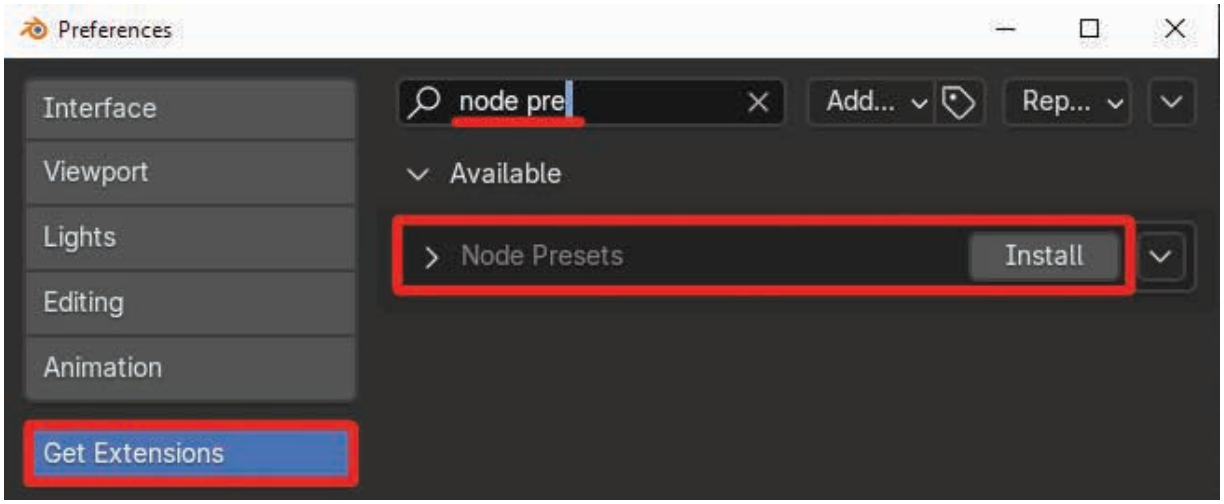
Here are some recommended sources of PBR texture sets:

Poliigon.com - Inexpensive and nice textures (affiliate link)
Texturehaven.com - Free textures, cc0 license so use as you see fit.
Textures.com - A very big database of textures. Many free ones.
cc0textures.com - Free textures with, as the name implies, cc0 license.
Freepbr.com - Guess what? Free PBR textures.
Lotpixel.com - A lot of nice textures made by photogrammetry, with many free ones.

Node Presets

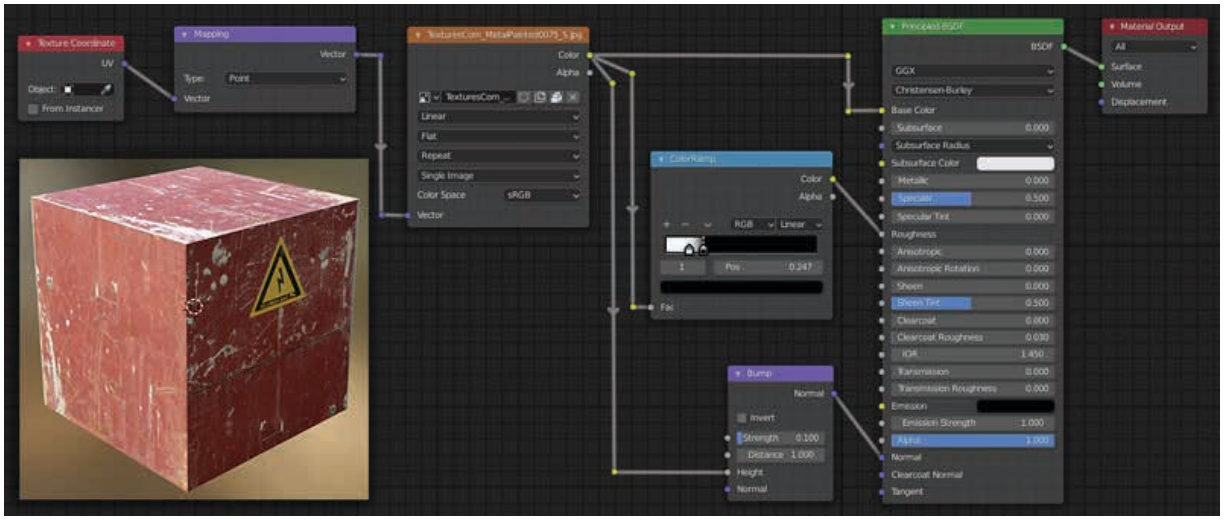
If there are combinations of nodes you use often, it makes sense to turn them into presets. This can be done with Node Presets, an add-on that comes with Blender by default. You just need to install it in the **Get Extensions** tab.

After that, you need to point it to the Blend file where you keep your node templates. In my case, I found that keeping it in a folder in the root directory of the same drive where Blender is stored worked. Before that, I tried to save the Blend file in my Documents folder, but this produced an error when trying to load a template.



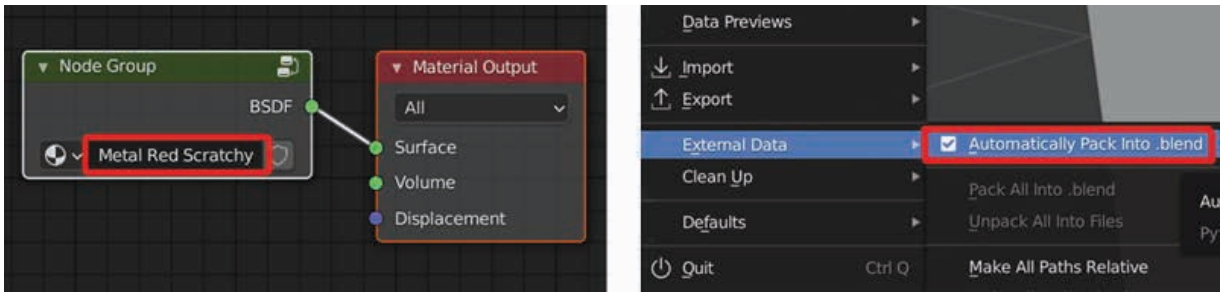
In case you're curious about that last sentence, here's what it means: you need to keep one central Blend file with all the node configurations that you want to have available as templates. The add-on will look at that Blend file and treat any node configurations in it as templates that can be used in any other Blend file.

You can start with just one or two things you can think of, and add more to them later. For me, something I do very often is this kind of fake PBR setup where I use a photo texture and use it for Roughness and Bump as well.



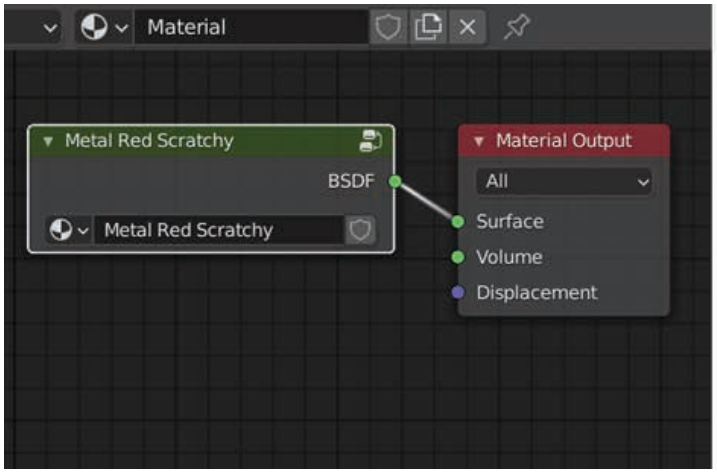
So I might as well make a template for it. I created this material in the Node Presets Blend file. Turn it into a Node Group by selecting the appropriate nodes (everything except the Material Output node) and **press Ctrl+G, then TAB** to create and collapse the Node Group.

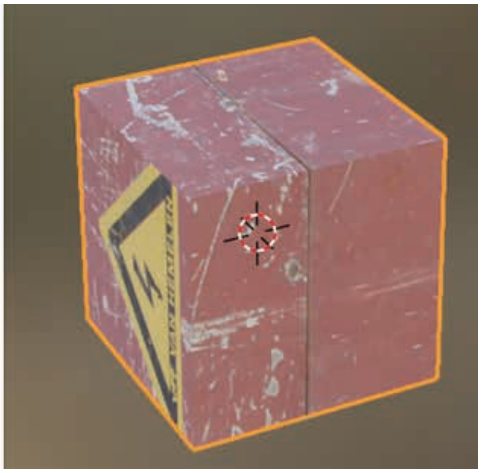
I give it an appropriate name and save and close the file. In this case the Node Group contains textures, so I checked "**Automatically Pack into .blend**" in the File menu.



Then I start a new Blend file to see if it worked. In the Shader Editor **press Shift+A** and choose add **Templates**. Then choose the Node Group preset that you want.

After that, delete the Principled BSDF by selecting it and **pressing X** (necessary in this case, since the Node Group already contains a Principled BSDF) and connecting the Node Group to the Material Output node. As you can see, even the texture remains (although the UV's of this object are different from the object in the presets Blend file).





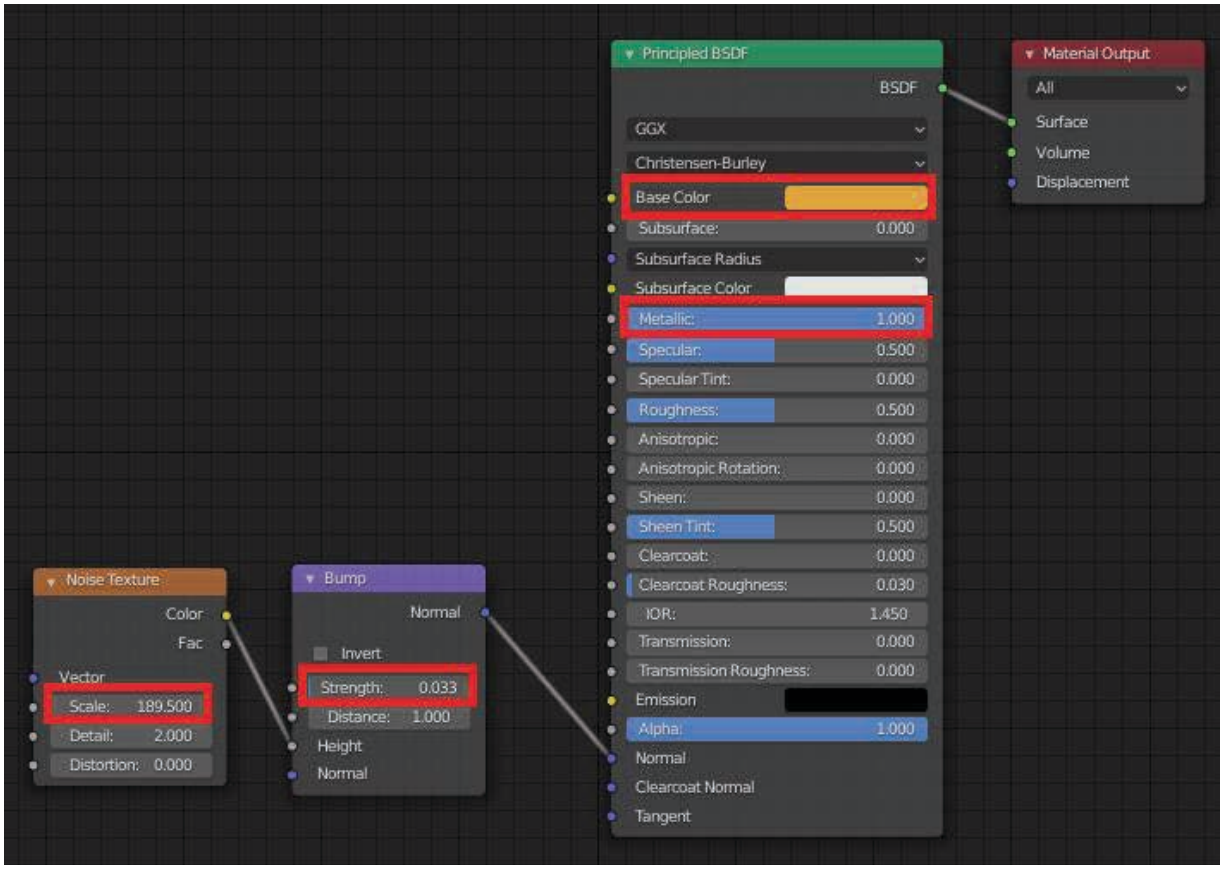
You're not limited to materials however, you can also use the Node Presets add-on for Compositing and more.

Material Basics: Pointiness

Blending two different materials together can have interesting results, especially if we use another special node to control how they are blended together.

Let's look at this material with worn gold-like edges.

First, I'll make a shiny gold-like material. It's easy enough, just create a new material for your object and set the color to be yellow/orangish and set "Metallic" to 1 like shown. To make things interesting, I like to add some bump. For this, add a Bump node and plug it into the Normal input of the Principled BDSF. Then plug a Noise Texture node into the Bump Height input. Just like shown in the illustration:



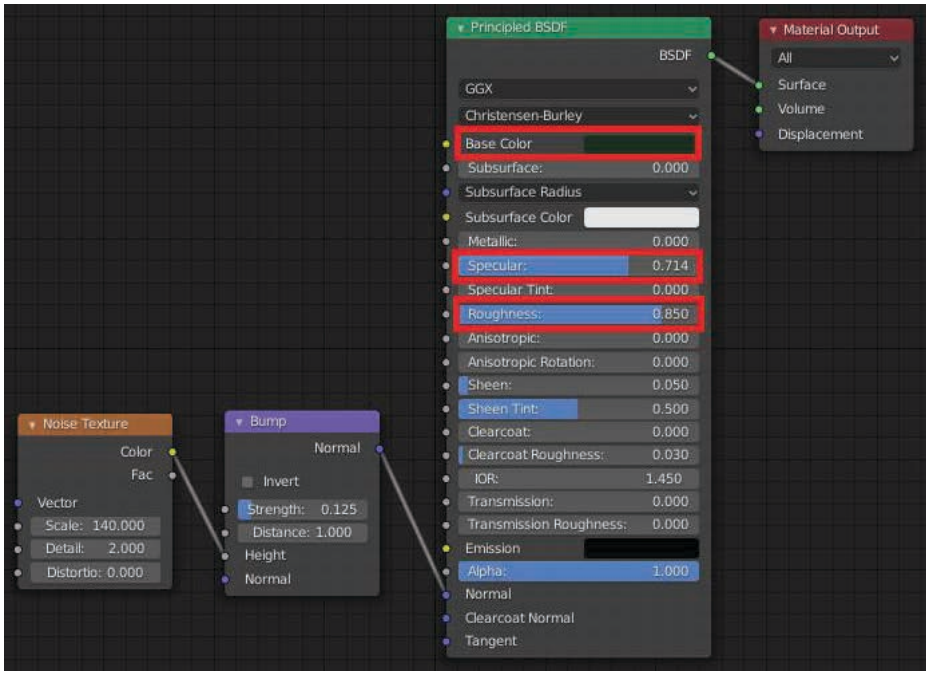
Play with the Noise Texture Scale first and Bump strength second to tweak the look. **Press Shift+Ctrl** and click on one of those nodes to see them visually while you tweak them (provided you have the Node Wrangler add-on enabled).

This is the result:

As you can see, the teaput with bump just has more surface roughness with which to catch light in an interesting way. It's a very easy thing to do, but it can make a world of difference to the realism in your scene.

Creating the diffuse material

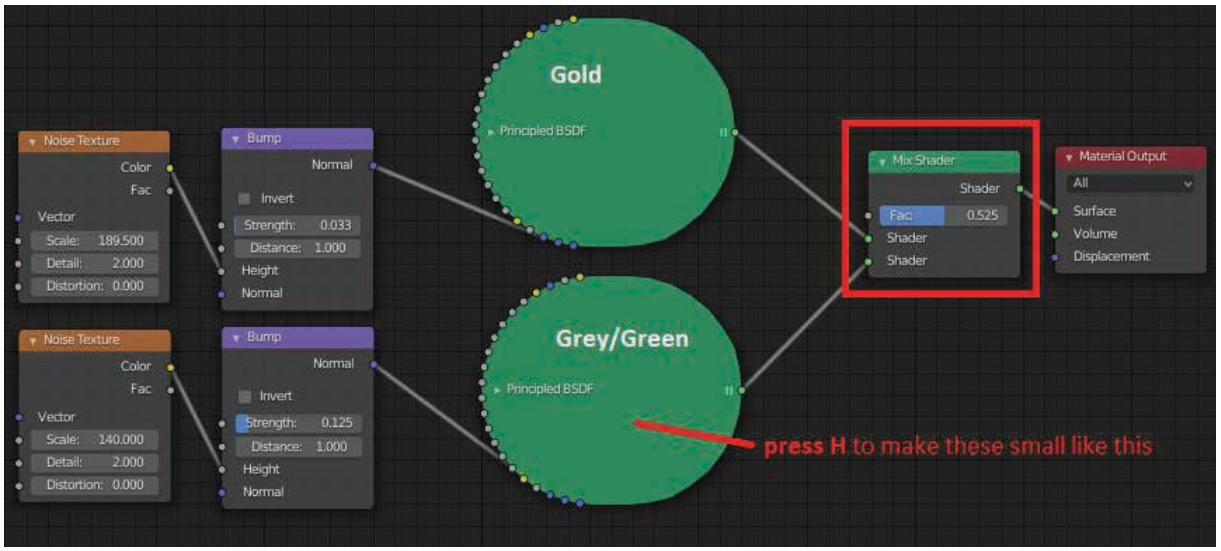
Select the Gold nodes and duplicate them (**Shift+D**). We can keep the bump node setup, and just change the settings like the **Base Color,** turn off the Metallic value, increase **Roughness** (to make it less shiny) and **Specular** (to still let it reflect some light, just not in a very shiny way).



This creates a dull-looking but slightly rough grey/green material.

We need a **Mix Shader** to combine these two materials.

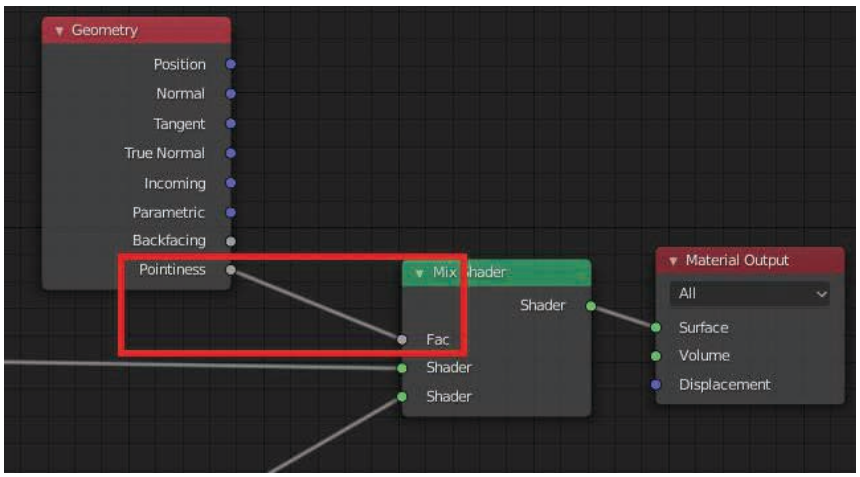
Tip: if you select a Principled BDSF node and **press H**, it will become smaller. Since we don't need to change their values anymore, just make both of them smaller by selecting them and pressing H. Then add the **Mix Shader node** (**press Shift+A** and search for it), then plug the nodes in like this:



As you can see, the materials are just blended half/half, which doesn't look very interesting. Now comes our Pointiness trick. Add a **Geometry node** (**Press Shift+A**, search for the node).

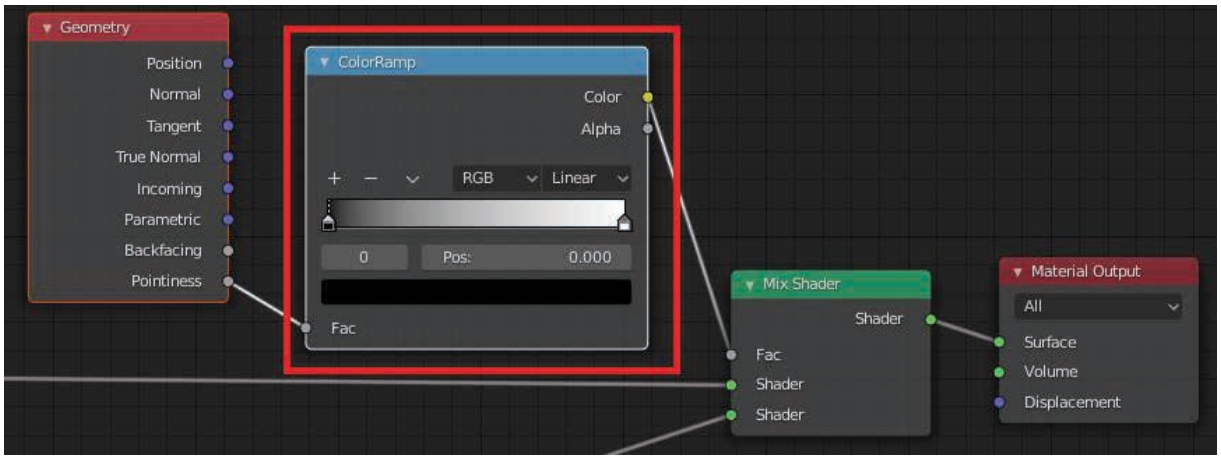
Important: The Pointiness node won't work unless you are in **Cycles** rendering mode. You'll also need to switch the Viewport to Rendered view in order to see the results.

The Pointiness output of the Geometry node goes into the Fac input of the **Mix Shader**.



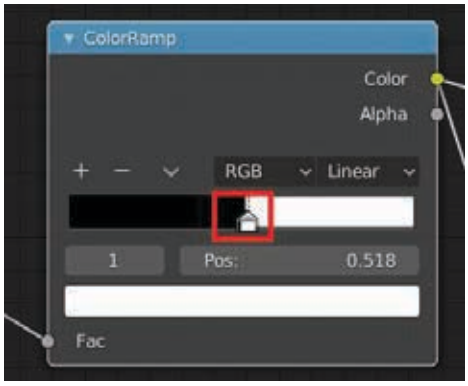
However, you won't notice much if any result right away. For that, we need another node.

Add a **ColorRamp node**. Drop it in on the connection between the Geometry node and the Mix Shader node and it will automatically get connected and make space for itself.



We need to fine-tune the ColorRamp node. To do that, **Shift+Ctrl-click** on the ColorRamp node so you see it visually in the viewport (your object should become grey now). Then bring the handles of the color ramp closer together to create stronger contrast.

You should see something like this after some careful fine-tuning.



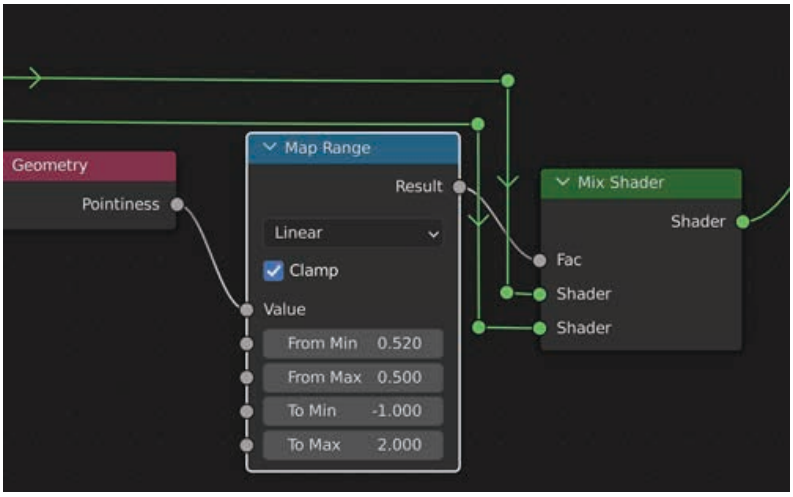
It takes some experimenting to find the right setting for the ColorRamp. It also helps to switch back to the Material Output (**Shift+Ctrl click** on the Material Output node) to see what the result will be.

It should look something like this:

A teacup with somewhat polished edges where the paint has worn off, showing the metal underneath.

Note, that the Color Ramp handles really need to be very close together. This is not easy. A newer node, the **Map Range node**, can help here.

At first glance, it seems less intuitive than the Color Ramp node.



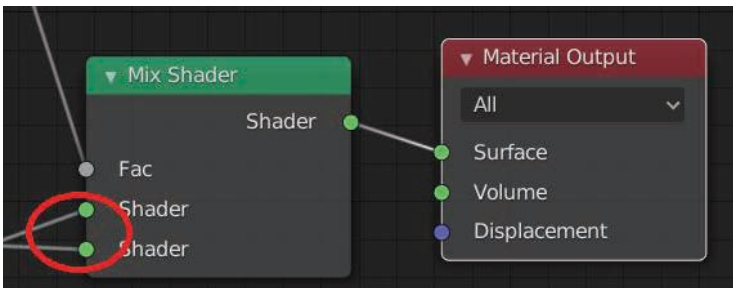
But it's really simple: **From Min** is the same as the left handle on the Color Ramp.

From Max is the same as the right handle on the Color Ramp.

In other words, bringing the **From Min** and **From Max** values close together as shown above, is the same as bringing the Color Ramp handles close together. This is of course easier than carefully dragging handles very close together. Also, lowering the **To Min** value and increasing the **To Max** value increases the contrast of the effect. This is something you can't do with the Color Ramp node.

It's possible that you need to switch the materials going into the **Mix Shader** node to get the correct result (otherwise it's gold all over, with some paint at the edges which doesn't make sense).

You might have to switch these inputs around.



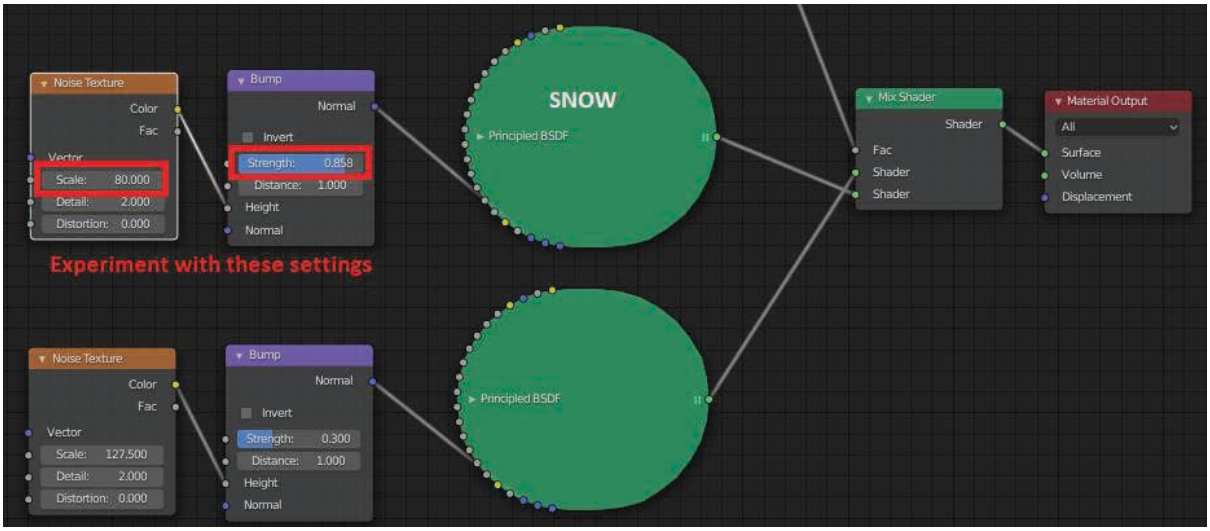
Another thing you can try if you don't get the desired result, is lowering the amount of subdivisions on the model.

Material Basics: Snow

If you understand the principle behind the previous tip, where we mixed two materials to create a worn edges kind of look, now we will try something a bit more ambitious: adding snow on top of a model. This can all be done simply with nodes.

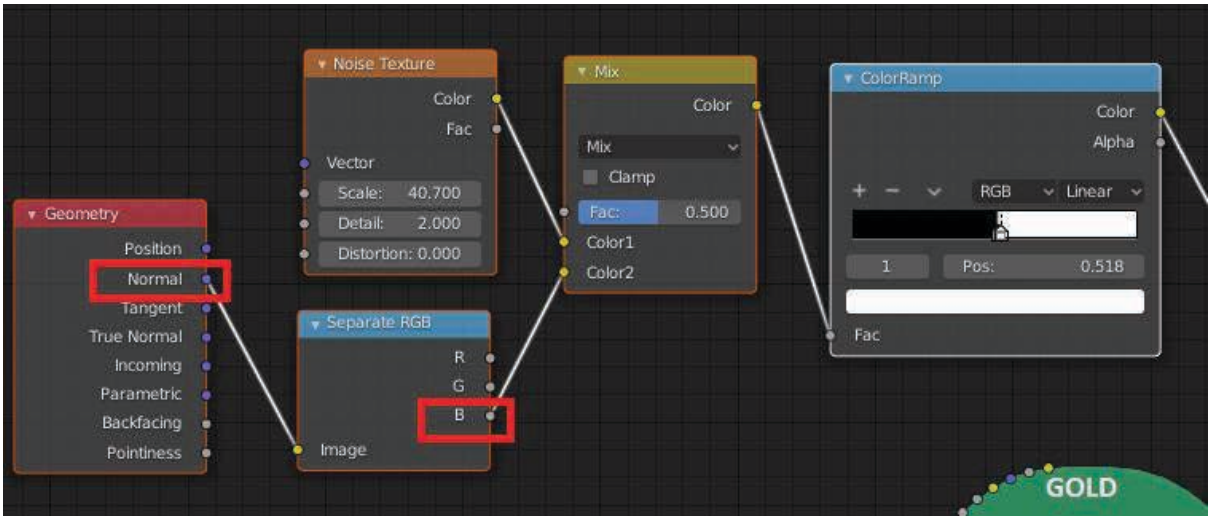
Instead of snow, you can also finetune the same node setup to create dust.

We start the same way as in the previous tip about worn edges, with two materials. We keep the dull grey/green one, and change the Gold into Snow by simply **increasing Roughness**, and changing the **Base color to white**(with a very small amount of blue- not 100% realistic, but it looks nice). You can also make the Bump a bit stronger by decreasing the Noise Texture Scale and/or Increasing the Strength of the Bump node.



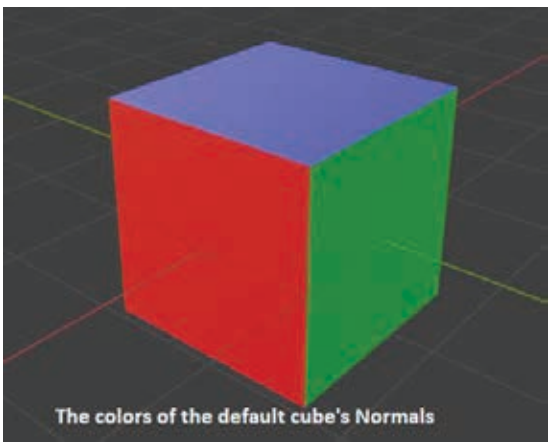
Select the Principled BDSF and **press H** to return it to its full size, so you can change the Base Color and Roughness values to make it more like snow. Then **press H** again to toggle it back to a smaller size.

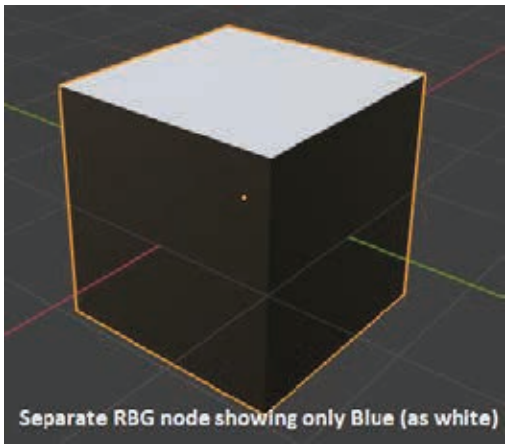
The Geometry Node which we also looked at in the previous tip, has an output called Normal.



The **Normal output** will show the X, Y and Z angles of the model as colors Red, Green and Blue. The **Separate RGB node** lets us filter out colors, with the result being a black and white output. In this case we only keep the Blue color. Blue will be white, the other colors will be black.

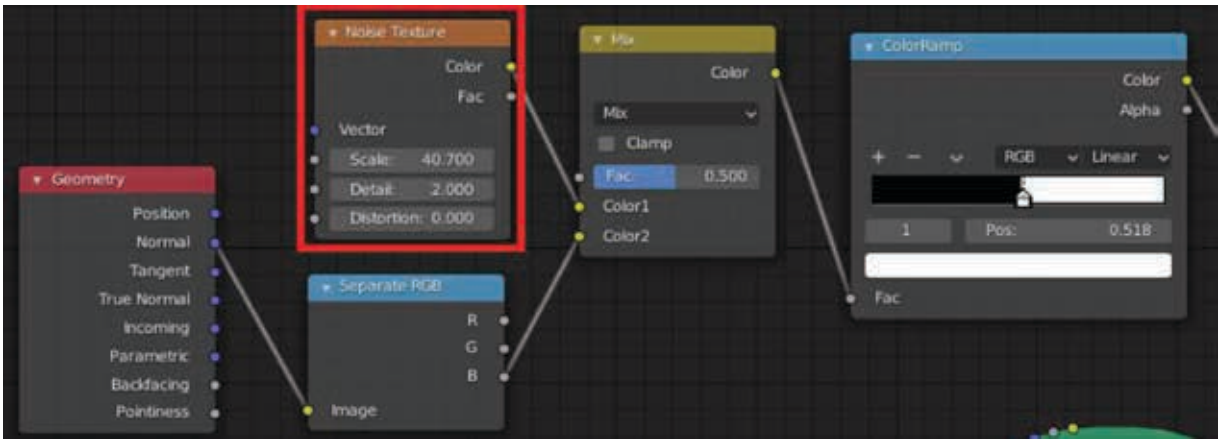
The reason we pick Blue is because that's the Z-axis. If we want to emulate snow, it will have to look like it fell down on the object from above, so the Z-axis makes sense.


The colors of the default cube's Normals


Separate RBG node showing only Blue (as white)

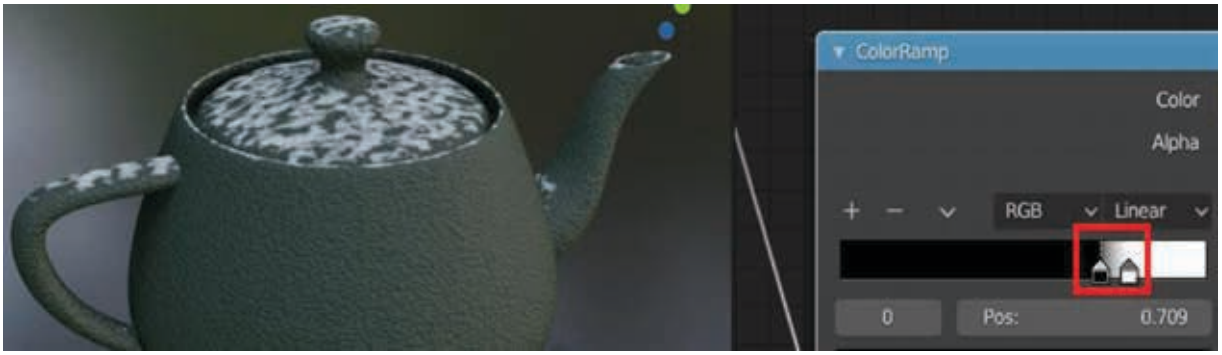
With our teapot, the Separate RGB node's blue channel gives us this:

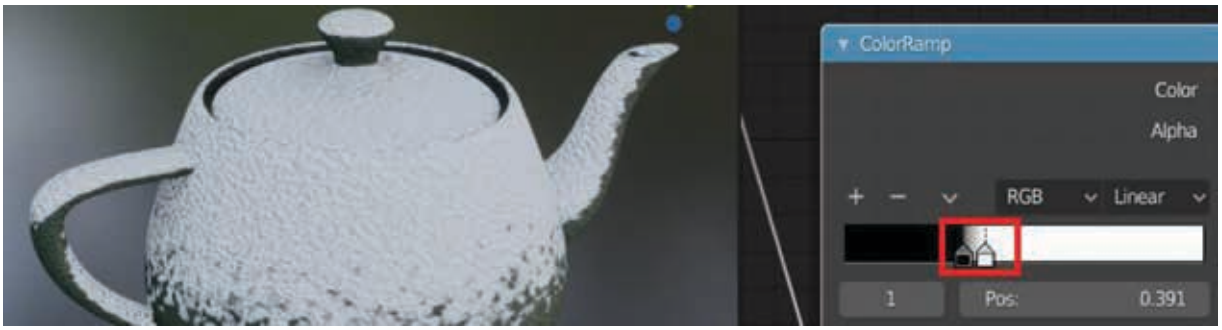
To make the mix between the snow and the other material more organic, we add a Noise Texture.

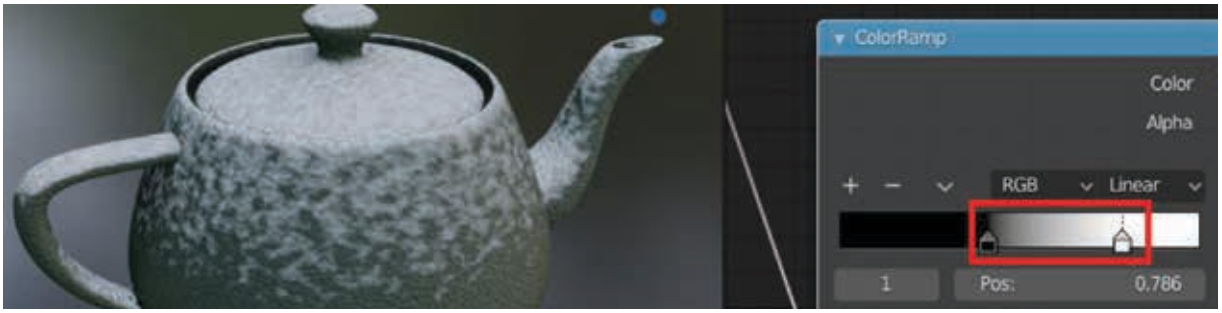


Remember that you can finetune and preview individual nodes by **Shift+Ctrl-clicking** on them. Pictured below: The Noise Texture (left) and the ColorRamp (right).

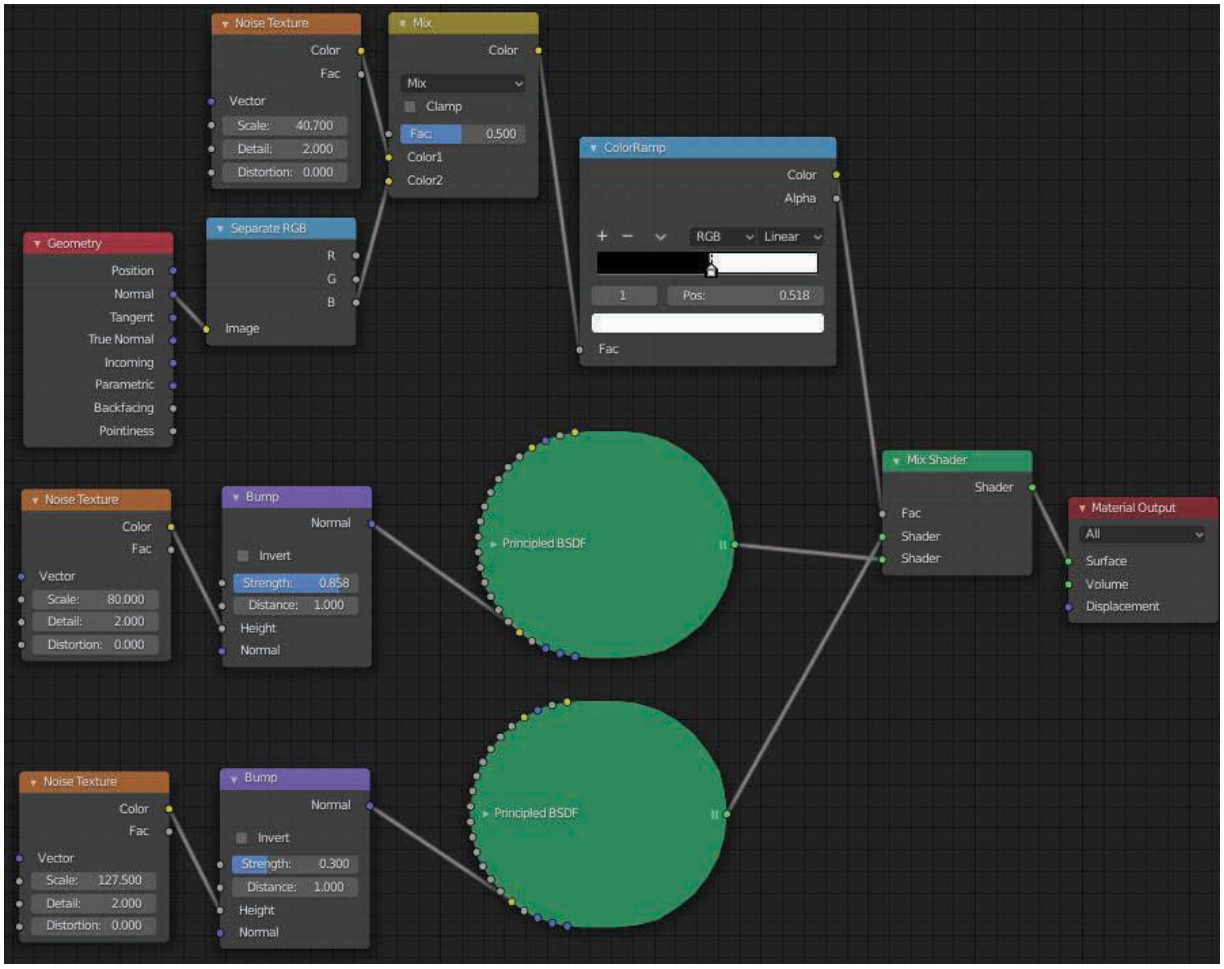
The ColorRamp controls the look of the snow. Some examples:



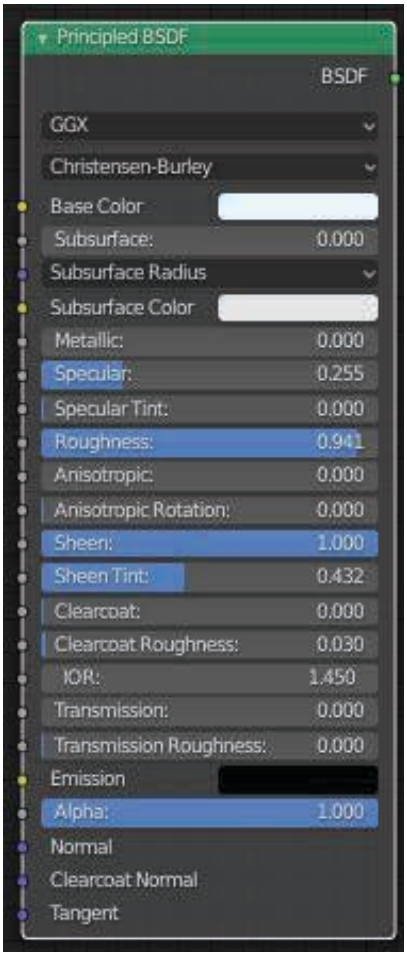


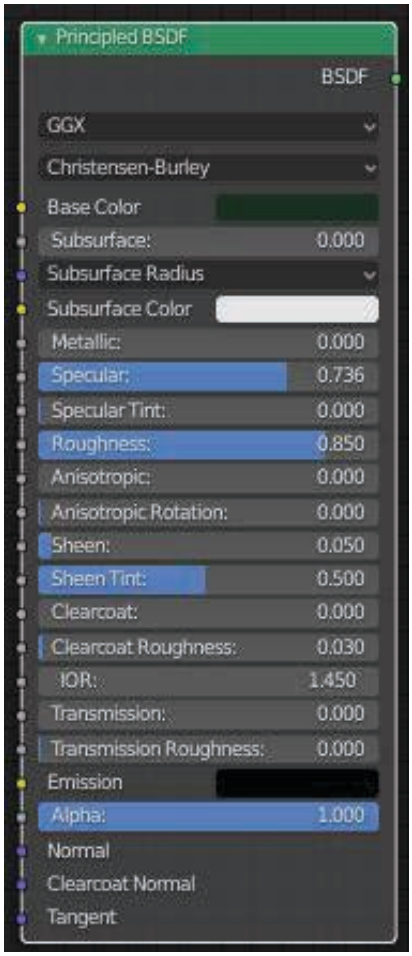


The full node tree:



The Snow material (below, left) is just a Principled BDSF with a white/blueish Base Color, some Sheen and Roughness. The green material has a green Base Color and some Roughness:

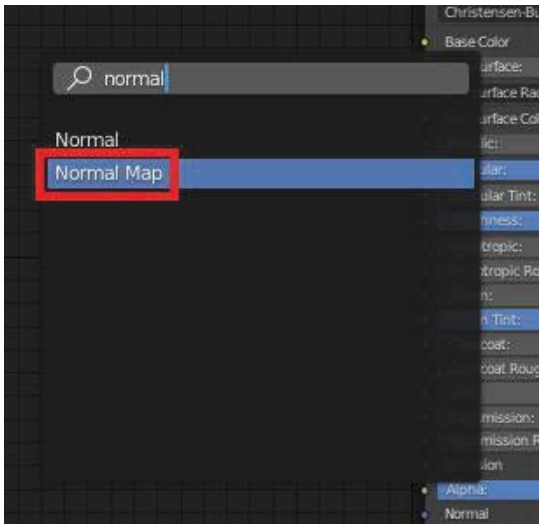




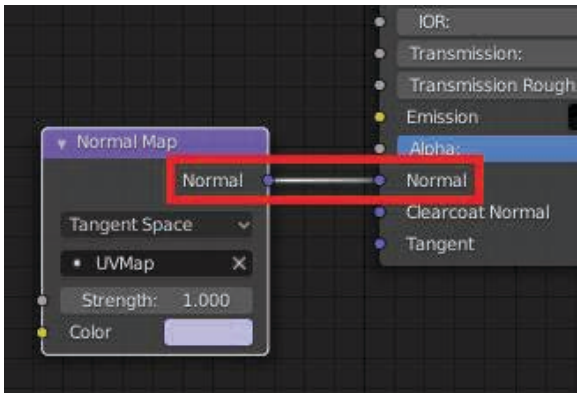
How to use normal maps

Create a plane (or use the bottom plane from the previous tip) by **pressing Shift+A** and choosing Mesh > **Plane**.

Give it a material and add a Normal Map node to your material.



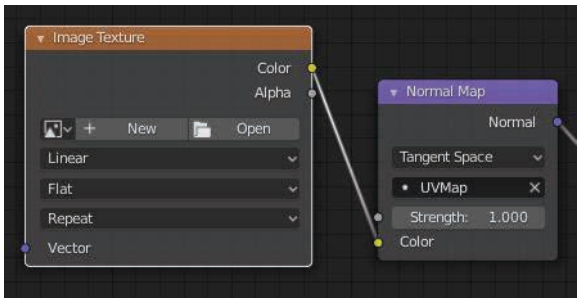
Connect the Normal Map to the Principled BSDF node's **Normal** input.



Add an Image Texture node by pressing **Shift+A** and searching for it.

You can also do this by **pressing Ctrl+t** with the Normal Map node selected using the **Node Wrangler add-on**.

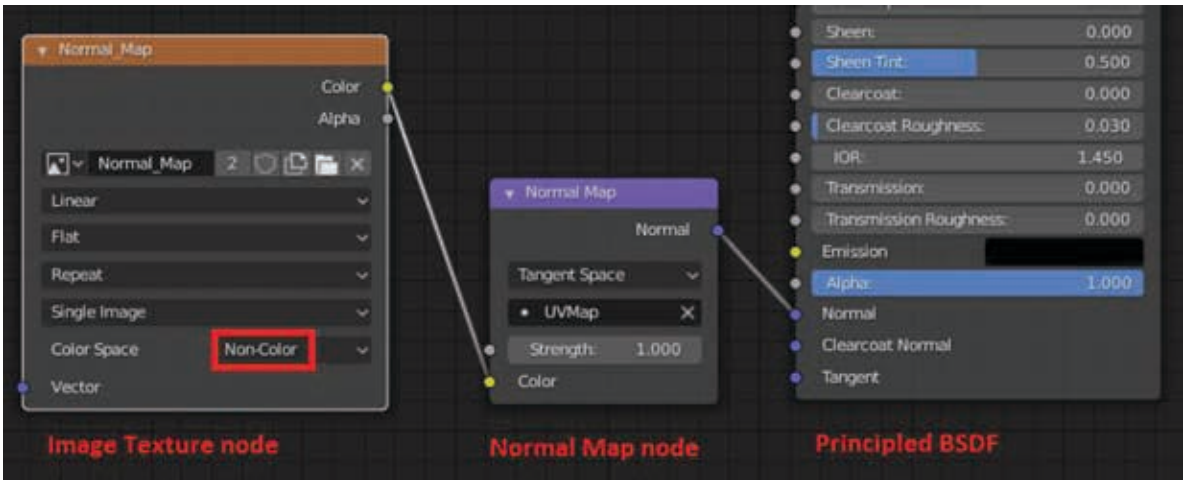
Click on **Open** and find the Normal Map image that you made.



Click here to return to the index.

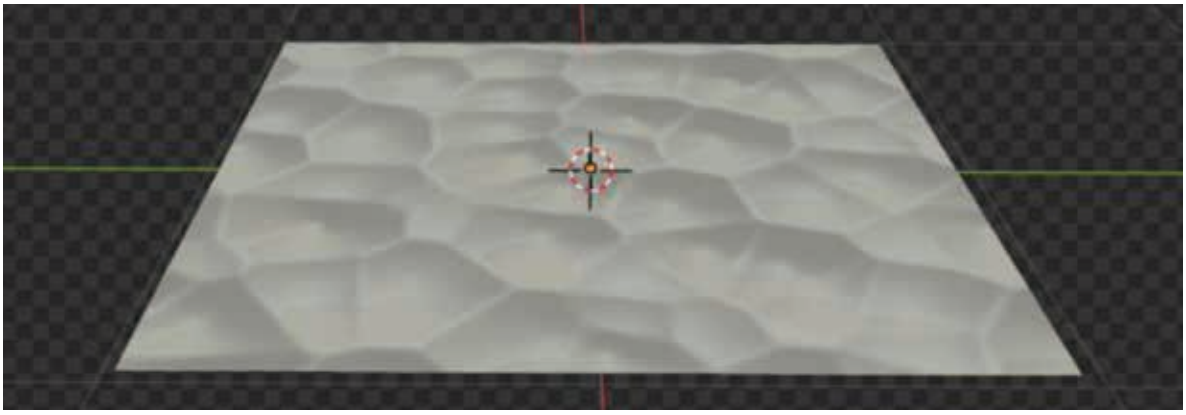
Make sure the Image Texture Color output goes into the Color input of the Normal map node. The Node Wrangler add-on doesn't do this correctly, so if you use that, make sure to reconnect the nodes manually in the correct way.

Make sure the Color Space of the Normal Map is set to **Non-Color**.

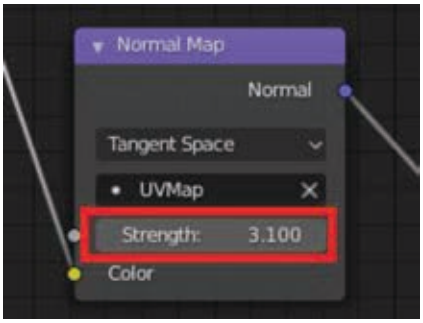


The result is different than with a displacement map. The geometry is not actually displaced, it's just faked. You will notice it at the edges of the plane, which are perfectly straight instead of displaced.

But remember that it is only one polygonal face! That's a lot of fake detail for such a low-poly (one-poly!) object. It's useful for adding fine detail to a simple mesh.



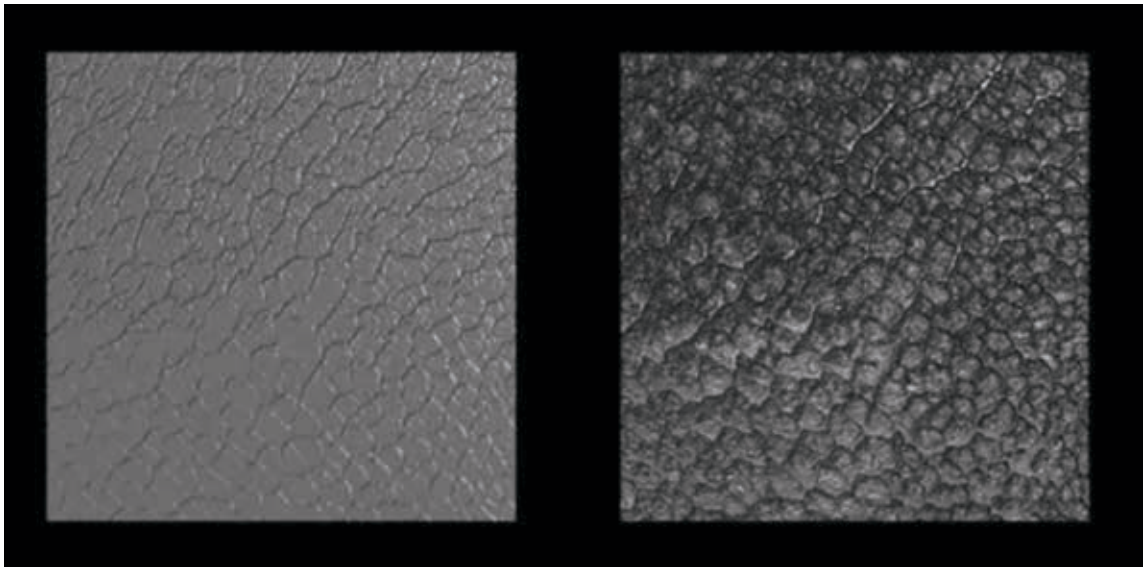
Note that in this case, I increased the **Strength value** of the Normal Map node to make it more noticeable.



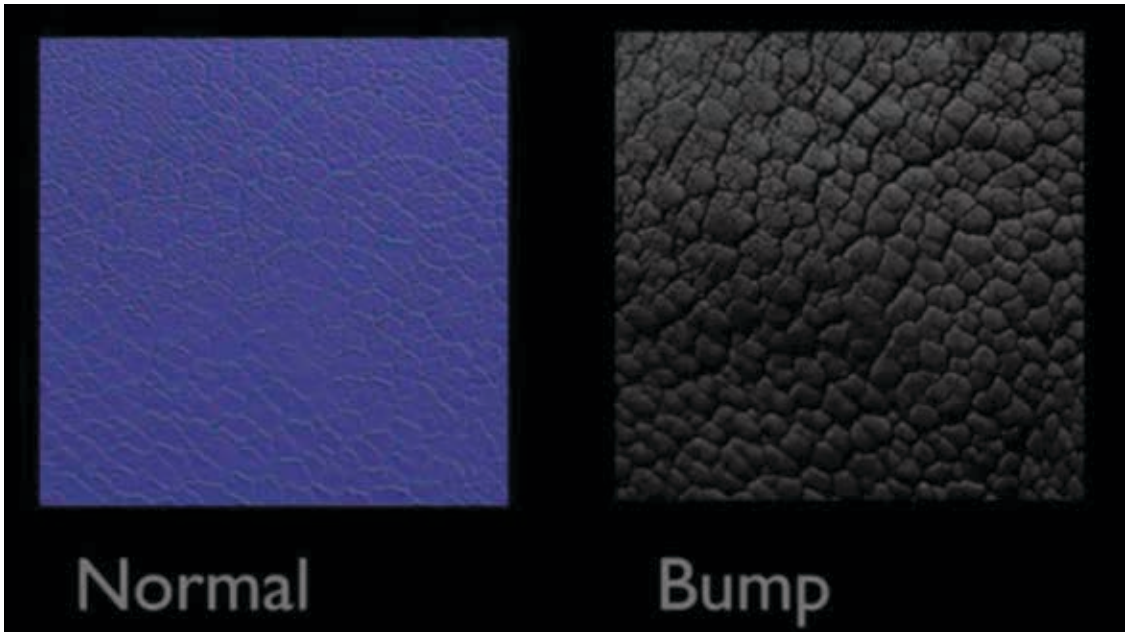
Bump maps or Normal maps?

Normal maps and Bump maps both do more or less the same. Normal maps are simply a more “modern” version of the Bump map.

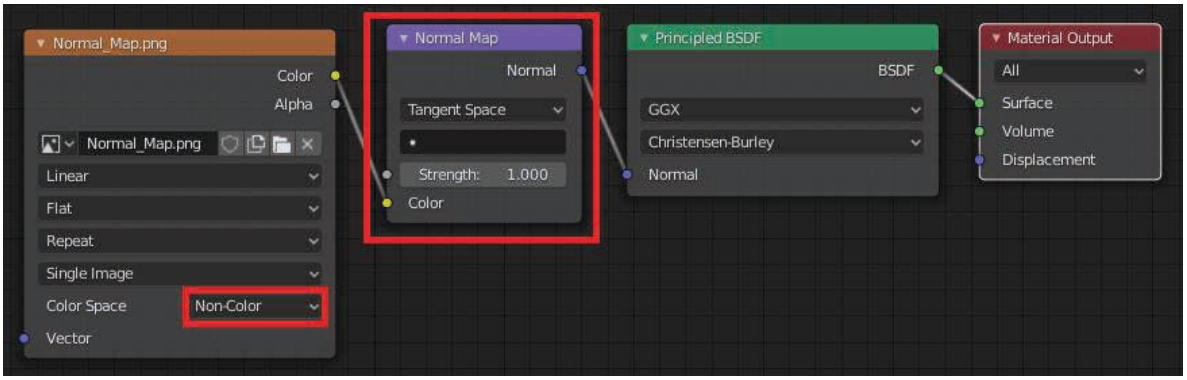
Both fake small details without needing extra geometry.



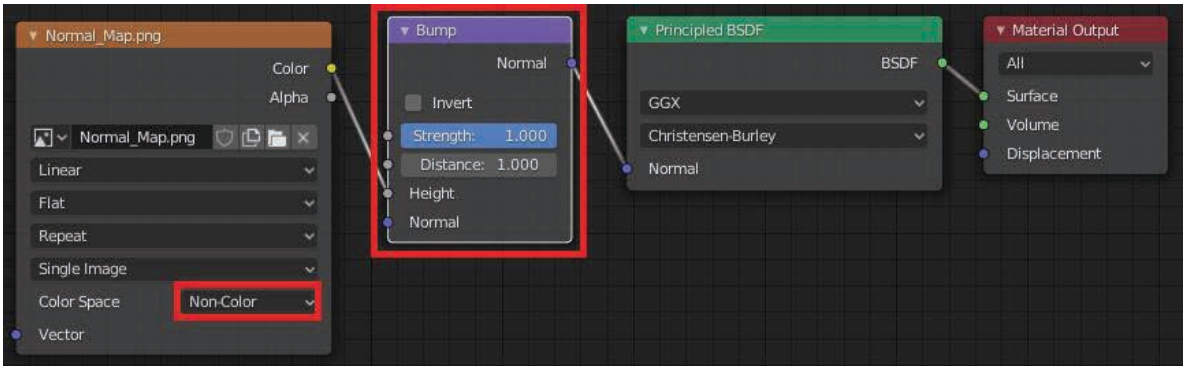
Normal maps react to light better because they use XYZ coordinates instead of only depth. Bump maps are converted to normal maps during rendering. With normal maps this is not necessary.



Their node setups are almost identical. This is a Normal Map node setup:



And this is a Bump map node setup. The only difference is a Bump node instead of a Normal Map node.



One advantage of Bump maps, is that any image can be a bump map. Whereas a Normal map has a very specific look. So you can in fact use your diffuse texture as a bump map. It doesn't always give a good result, but in some cases it's better than nothing.

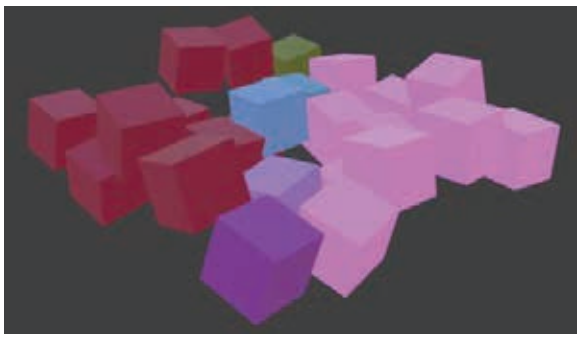
Bonus tip: there is an interesting online tool called NormalMap-Online where you can convert a grayscale map to a normal map.

You can find it here: http://cpetry.github.io/NormalMap-Online/

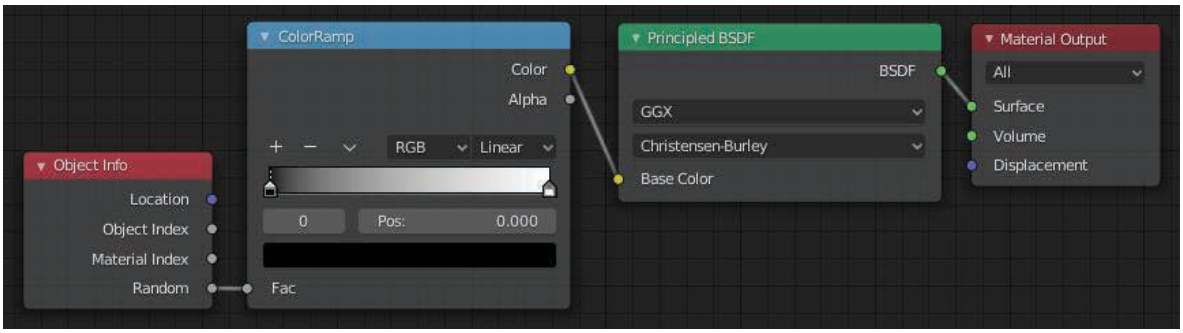
So what should you use? Well, a Normal map will react better to the lighting in your scene so if possible, go with that. Bump maps have the advantage that any image can be a Bump map. If you have time, convert the image to a Normal map with the above mentioned web-based tool.

Random colors with the Object Info node

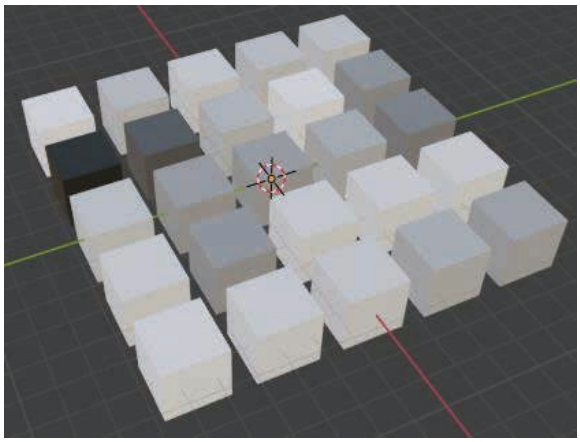
The Object Info node has a Random output. It can create random colors for objects.



This is interesting for motion graphics, for example. Objects with this material will have random colors, different values of grey in this case as a ColorRamp node is added in between the Object Info node and the material.



This results in the objects that have this material assigned (in this case a bunch of cubes) having different colors.



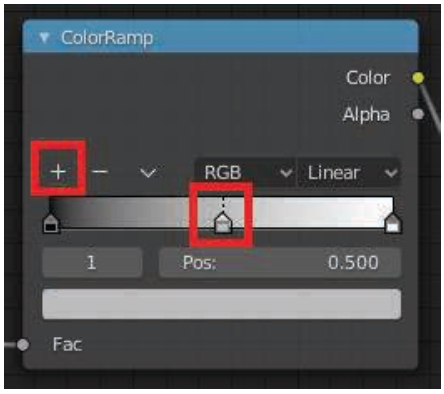
Because the **ColorRamp** is a default black to white gradient, you only get grayscale values.

You can add more colors on the Color Ramp.

To add a color, click on the **+ icon**. A new color “stop” is added to the ramp.

To remove stops, select them and press the-icon.

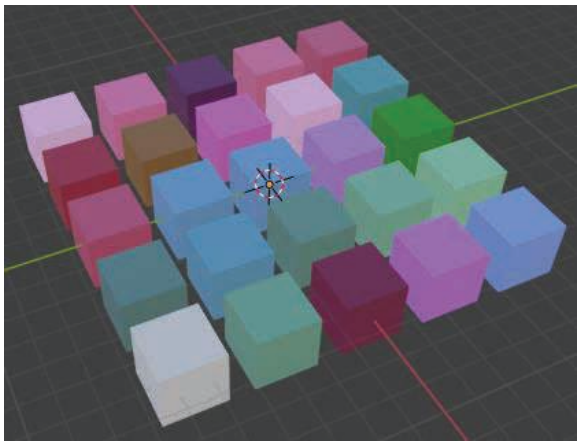
Clicking on a stop selects it. You can then change the color in the field below.



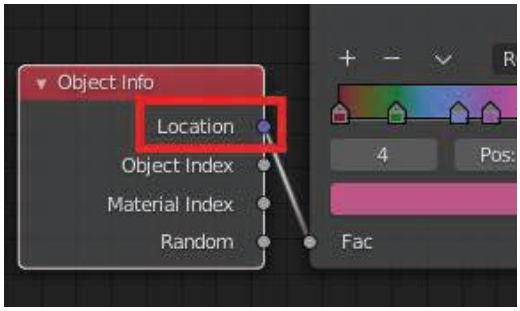
You can keep adding colors.

You can also move the stops if you want to, by selecting and dragging them.

The result is a colorful bunch of objects.

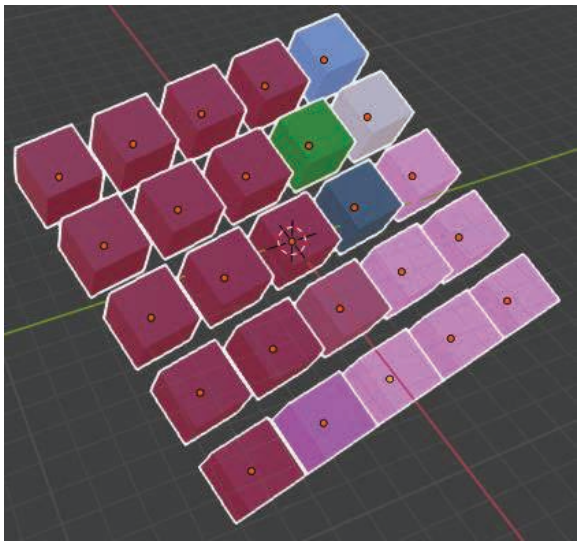


Instead of using the Random output you can use the **Location** output for a different result.

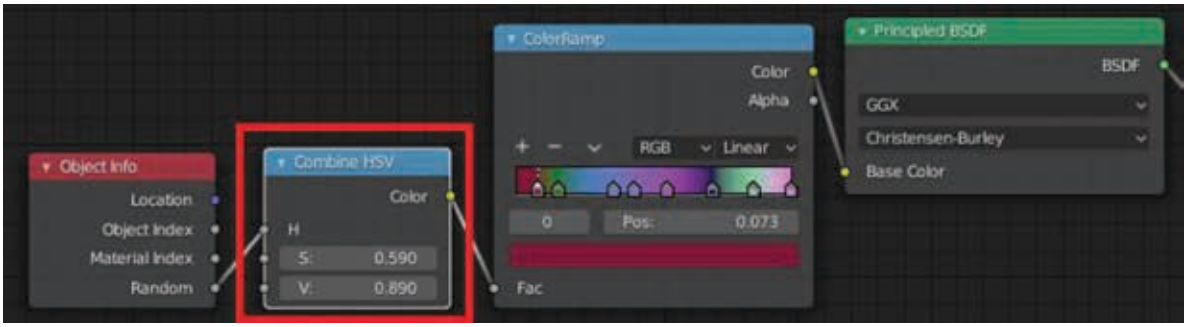


Now the objects will have a color assigned to them based on their location in 3D space.

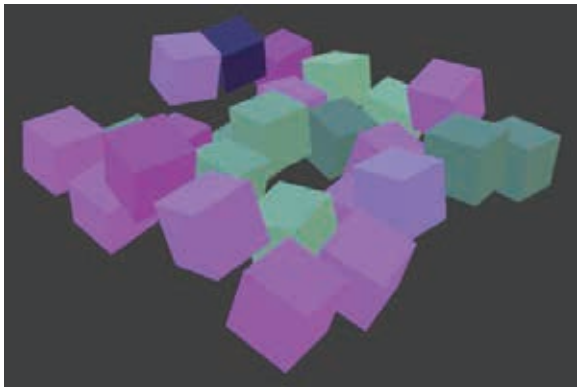
You will notice that if you move or rotate them, they will change color.



For a different result, you can add a **Combine HSV node** and plug the Random output into the **H input**. Then, experiment with the **S and V values** to see what you get.

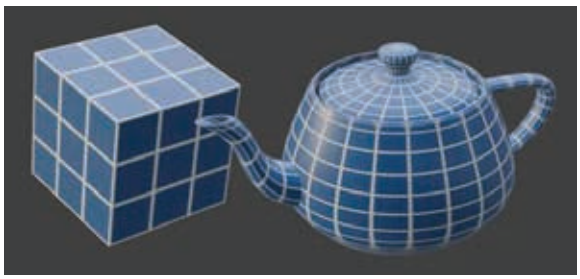


This gives you some additional animatable values in the Combine HSV nodes.



Easy wireframe shader

This is an easy way to create a wireframe shader.

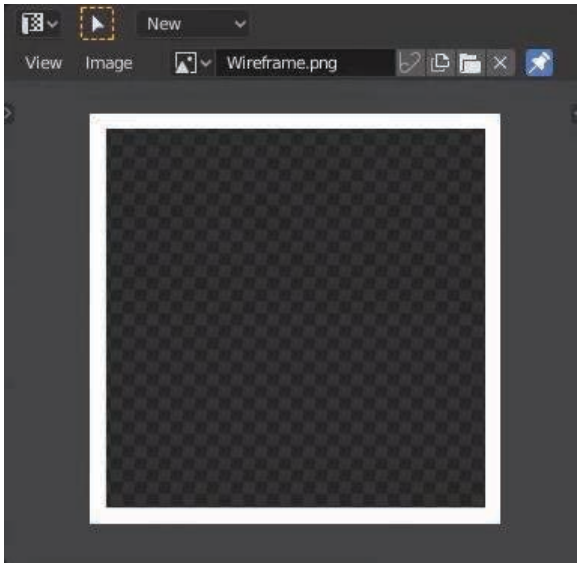


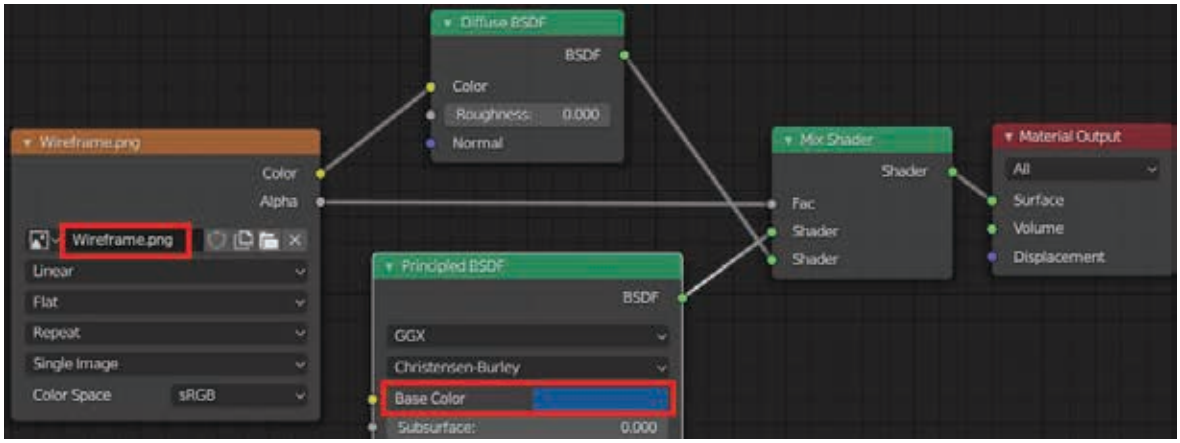
It requires a simple texture of wires: a PNG image with alpha channel for transparency (shown on the right).

It is mixed with a Principled BSDF that gives us the base color (blue in this example).

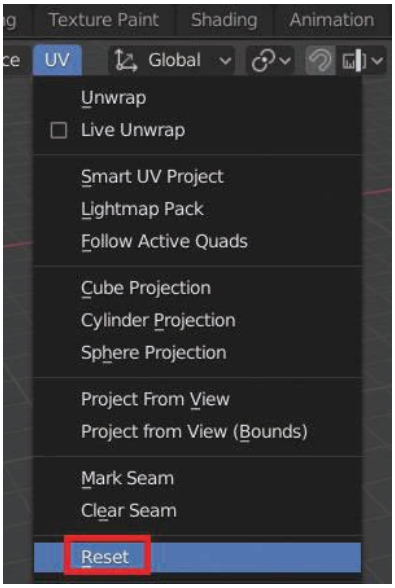
Below you see the shader set-up. Besides the color, nothing is changed in the Principled BSDF shader.

To add any shader, **press Shift+A** in the Shader editor and type the shader name.





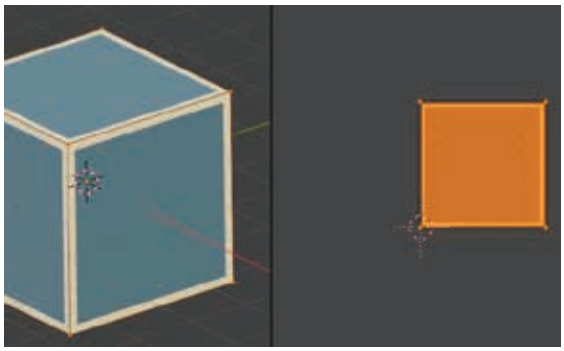
In Edit Mode select all faces and press U, select "Reset". Or choose it from the UV drop-down menu.

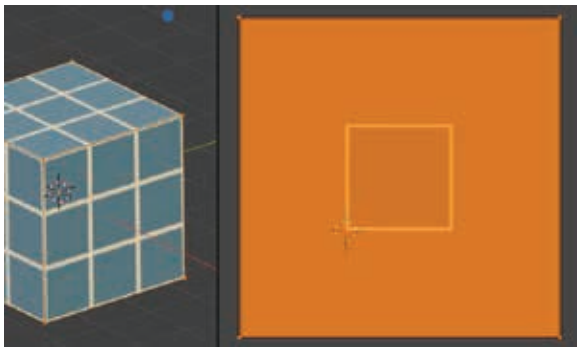


This resets each individual UV to fill the entire UV map, so in other words each Face of the model is unwrapped to fill the entire UV map.

Each Face therefore shows the entire texture.

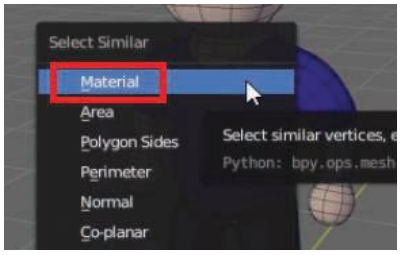
Scale the UV's in the UV editor, to change the wire thickness. Below left you see the reset UV filling each Face with the entire texture. Below on the right you see the UV's scaled up 3 times (press S+3) so the texture fills each Face 3 times.



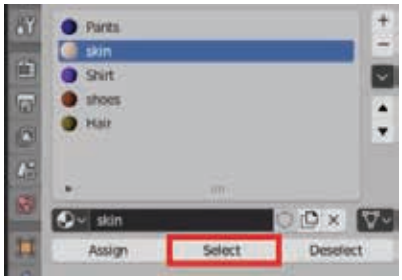


Select materials from characters

To select all faces with a certain material on your low-poly character **select one face** and **press Shift+G**. Choose **Material**.



Alternatively you can press the **Select** button under the material slot to select all faces using that material in Edit Mode.

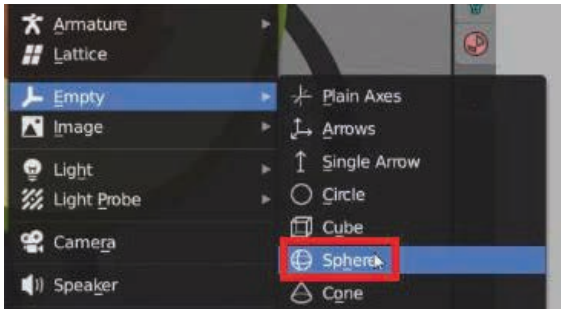


Using Empties to control textures

Previously we looked at using a texture with multiple facial expressions and animating them by changing the Mapping location values.

That works fine, but we always have to fiddle with values in the Mapping node.

Wouldn't it be nice if we could control this in the viewport? Add an Empty.



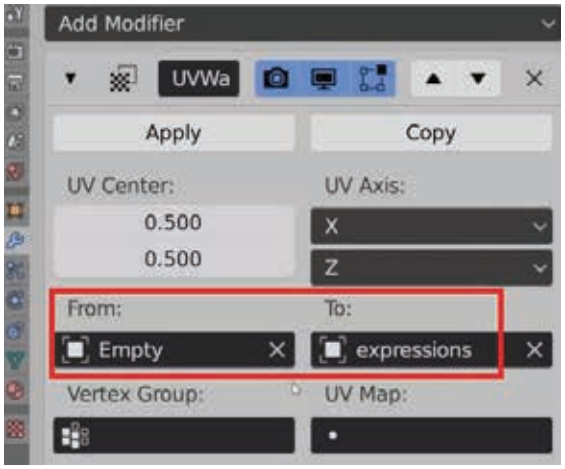
Any kind is fine, but the **Sphere Empty** is a bit more visible than the Plain Axes.

Add a **UV Warp modifier** to the "face" object (the plane with the facial expression texture).

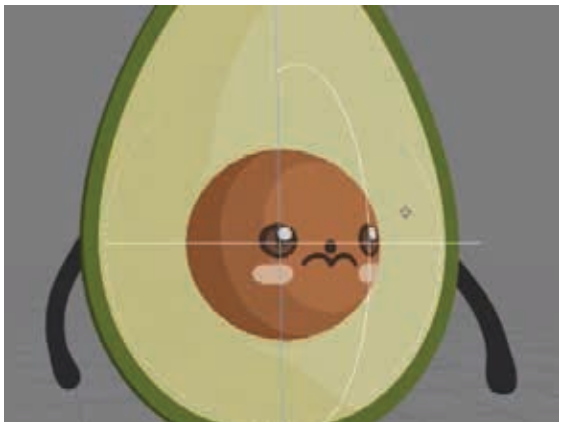
In the **From** field choose the **Empty**.

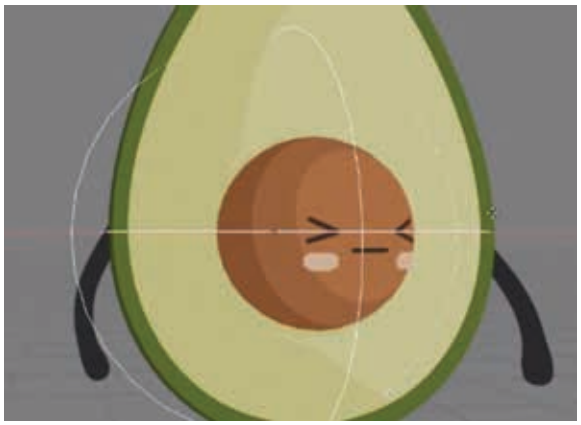
In the **To**field choose the face object (here called "expressions").

You may need to change the UV axis in your case like me, to X, Z.



Moving the Empty now controls the texture. You can parent the Empty to the body of the avocado with **Ctrl+P** so it stays with the avocado while it's walking for example.

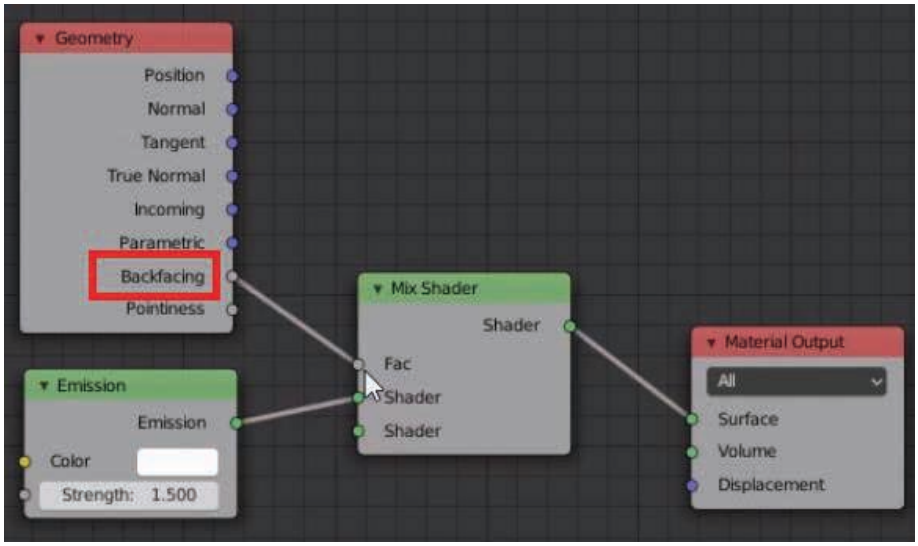




Single-sided Emission shader

In Cycles you can make an emission shadereasily. But what if you want to make it only emit on one side like in the picture on the right?

You need a Geometry node. We'll use the Backfacing output. Plug it into a Mix shader.

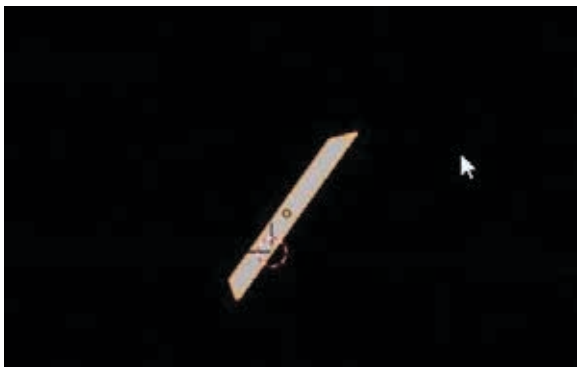


Now the Plane with the shader only emits light on one side.

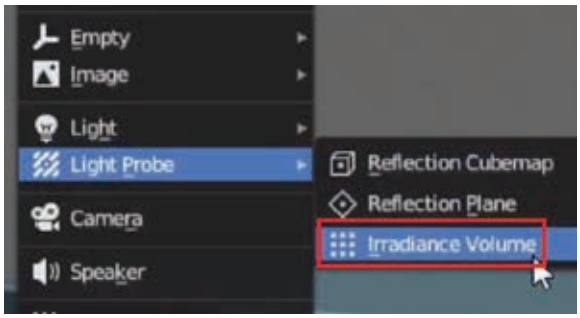
Emission shader in Eevee

In **Eevee** an object with an emission shader normally doesn't affect its surroundings.

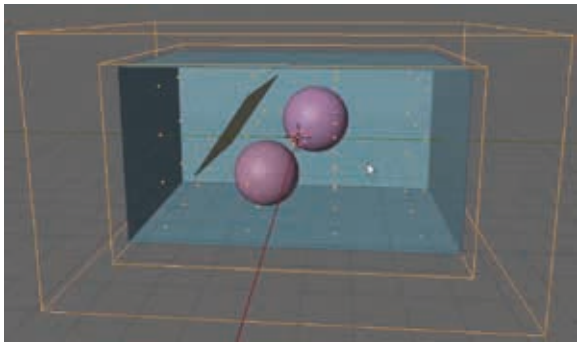
Here we have a plane with an emission shader in Eevee. You can see the plane, but the rest of the scene is in complete darkness.



We have to add a **Light Probe**, specifically an **Irradiance Volume**, around the scene.

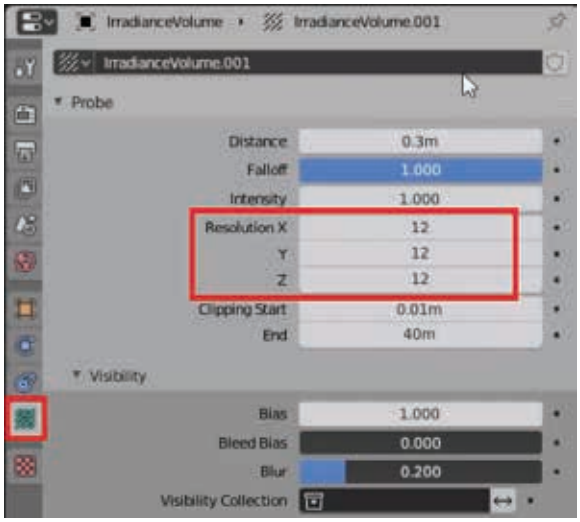


Scale it so the inner "cube" of the Irradiance Volume surrounds the objects you want to be affected.

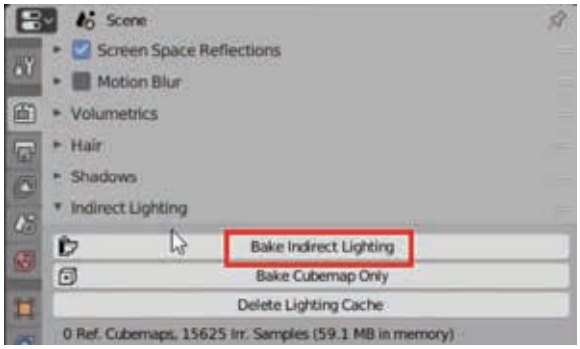


You can increase the **Resolution** for more precision, but it will require a longer baking time.

More resolution isn't always necessary, often the default values will do fine.



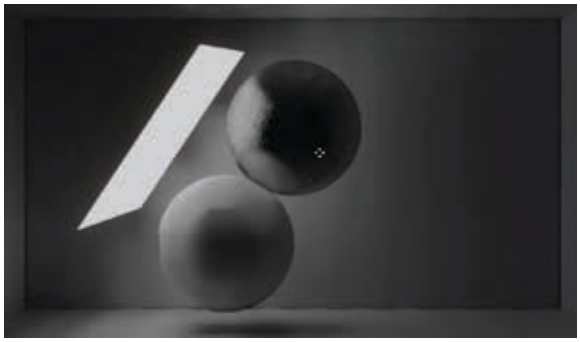
Then click on **Bake Indirect Light** and wait for it to calculate.



And there we go!

Now the plane with the Emission shader illuminates the scene in Eevee.

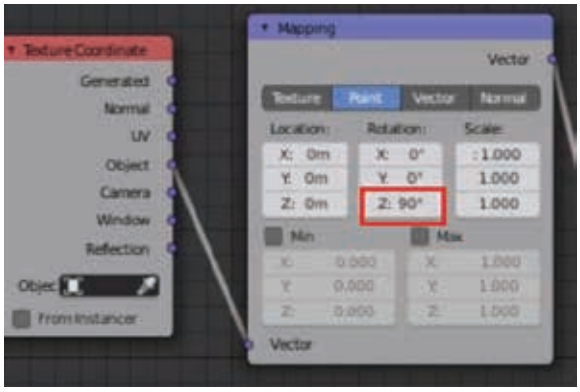
Unfortunately you can't really move things and have the light change interactively like in Cycles.



Rotate box-mapped textures

Here we have a simple Box-mapped or Triplanar mapped texture. What if we want to rotate the texture direction?

One idea is to rotate it on the Z-axis.



Looks okay, until we see the side - there is stretching of the texture. So this way doesn't really work. Let's undo that by **pressing Ctrl+Z**.

Instead go to Edit Mode by **pressing Tab**, select all by **pressing A** and rotate the mesh by 90° on the Z-axis by **pressing R+Z+90**. This rotates the mesh but not the textures.

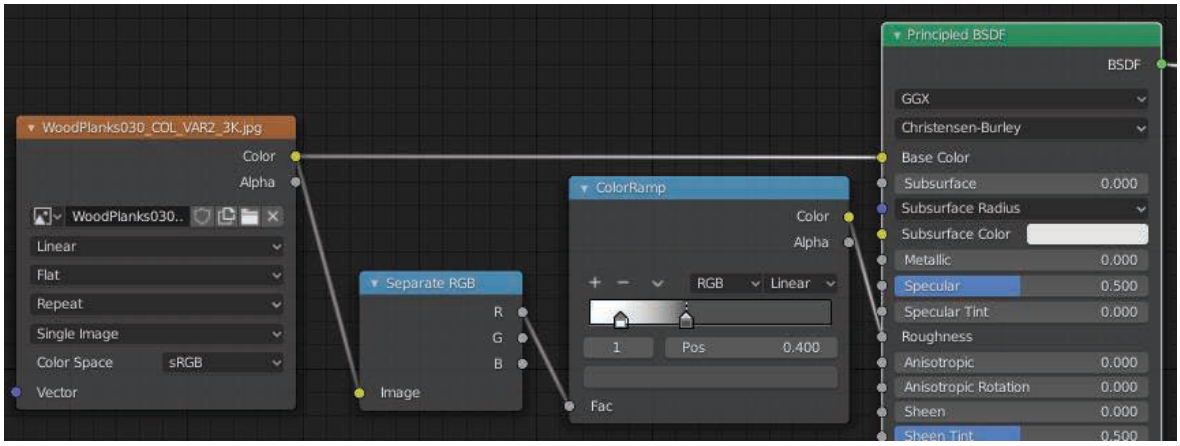
Next, press Tab to go back to Object mode and rotate it -90° on the Z-axis (so the opposite direction of the previous rotation) by **pressing R+Z+-90**.

That did the trick!

Using the Separate RGB Node

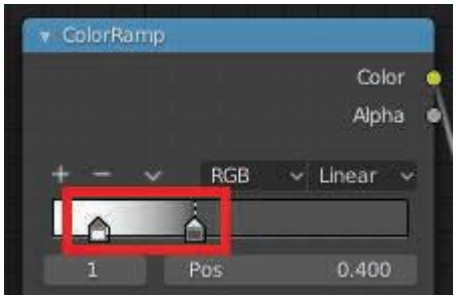
When you only have one image texture file, you can use the **Separate RGB node** to create fake roughness and bump maps.

For Roughness, add a **SeparateRGB node** and a **ColorRamp node**. Roughness controls what parts of the material are shiny, and how shiny those parts are.

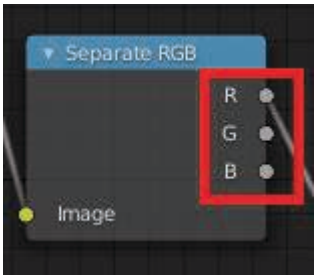


The **ColorRamp** controls what the Roughness looks like.

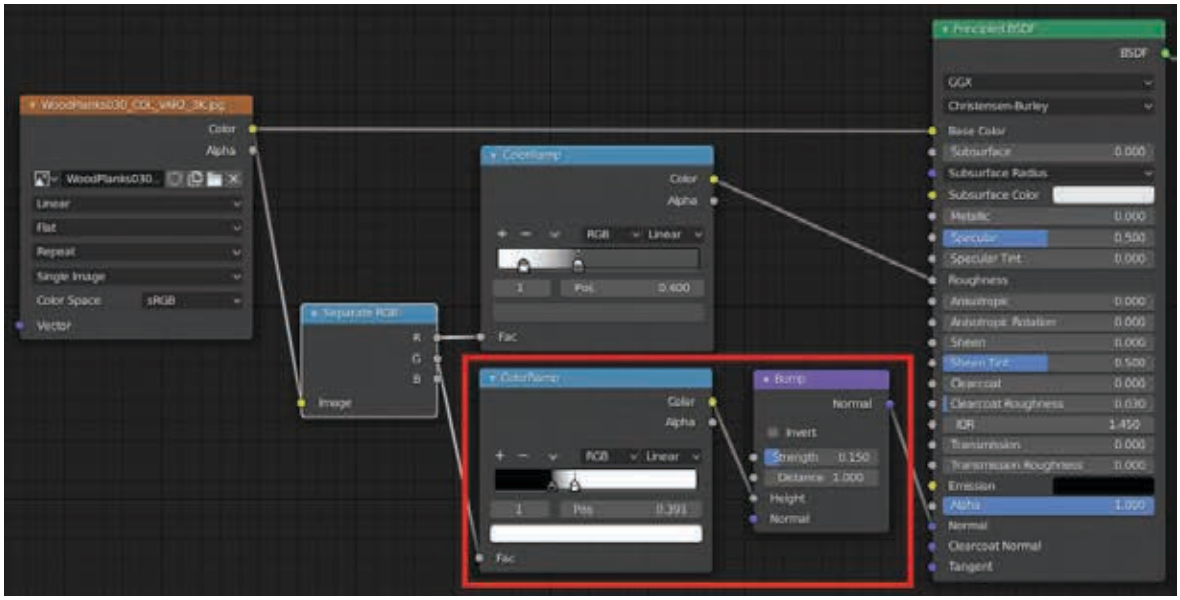
Check the result in real-time in the Eevee viewport.



You can use the **R, G or B channels** to drive the Roughness of the texture. They all look a bit different. Experiment to see what works best for the imagetexture you are using.

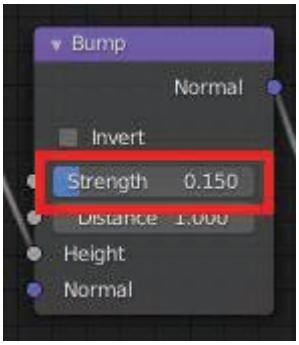


Duplicate the **ColorRamp node** by selecting it and **pressing Shift+D**. Plug it into a Bump node and connect them to the Normal input of the Principled BSDF.

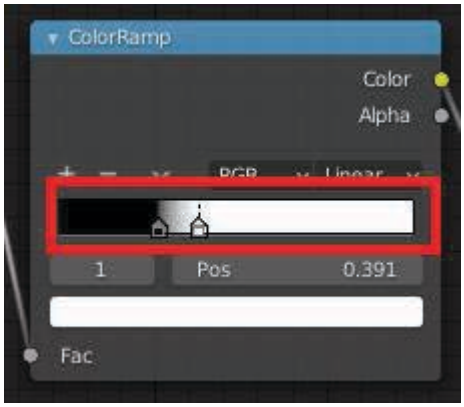


Here you can use R, G or B output as well, whichever looks best for your specific image texture.

You'll probably need to decrease the value of the Bump **Strength**.



Here as well you can experiment with the ColorRamp, switching the black or white values around and decreasing them.



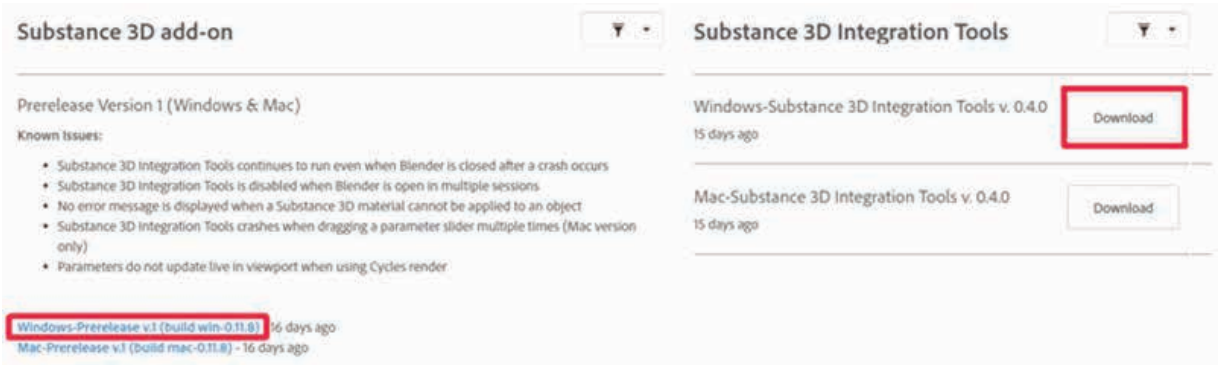
Now we have both Bump and Roughness, just from one Jpeg texture!

Substance Material Add-on for Blender

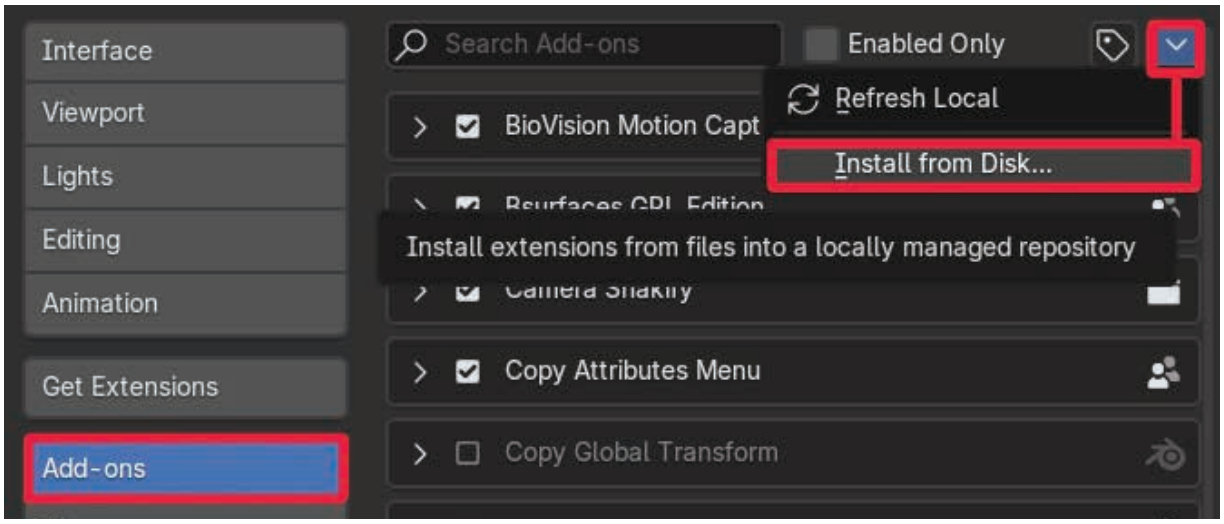
Without a doubt, **Substance Painter** is one of the most popular material editors out there. You can use it either for a monthly subscription price or you can get a version on Steam where you only pay once. However, you can now take advantage of the many free materials made by others and shared on Substance Share inside of Blender! Thanks to the add-on made available by Adobe.

You can download the free add-on here, after reading the licensing documents and agreeing to them. https://substance3d.adobe.com/magazine/the-substance-3d-add-on-for-blender-is-here/

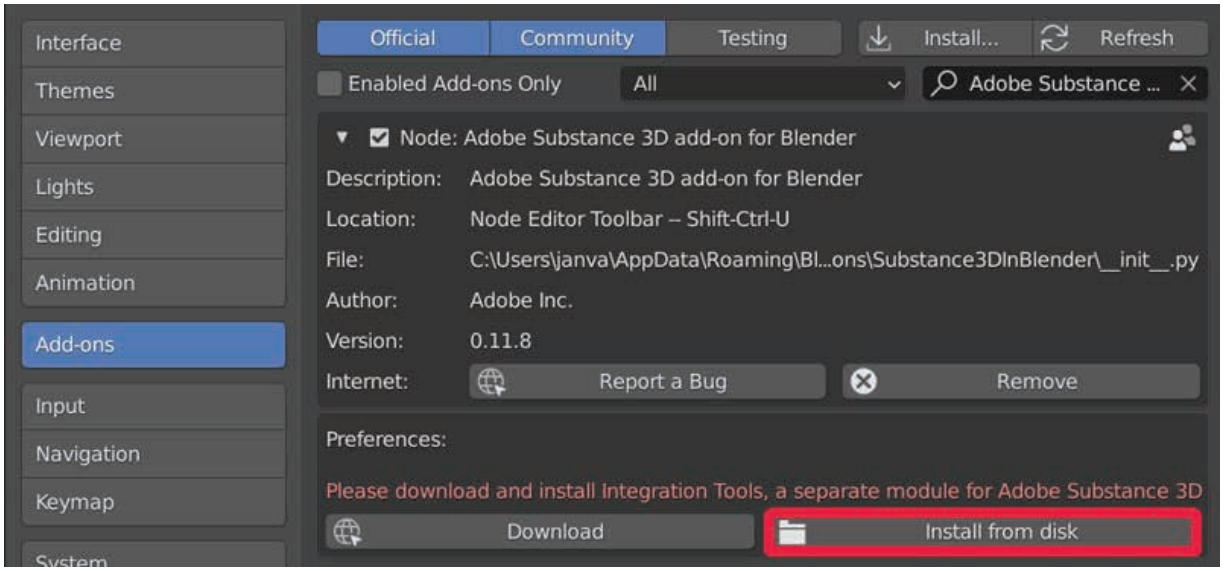
Note that it works for **Blender 2.9 and up**. Once you've agreed to their license, you go to a page where you can select your operating system and download the two required files. In my case, I downloaded the Windows versions.



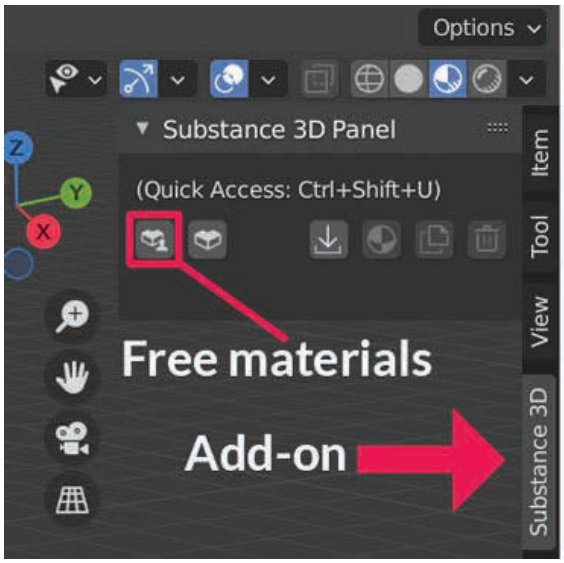
From Preferences under Add-ons, install from file the ZIP file "Substance3DInBlender-win32-0.11.8".

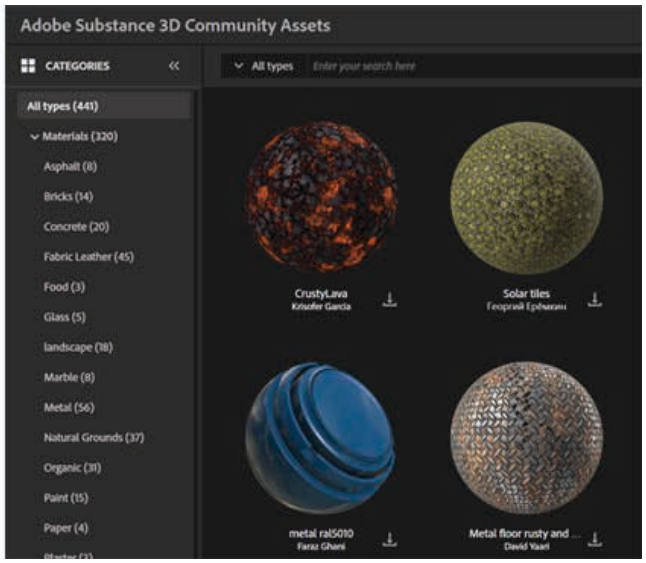


After this, expand the add-on in Preferences and click on **Install from disk** to install the other file you downloaded.



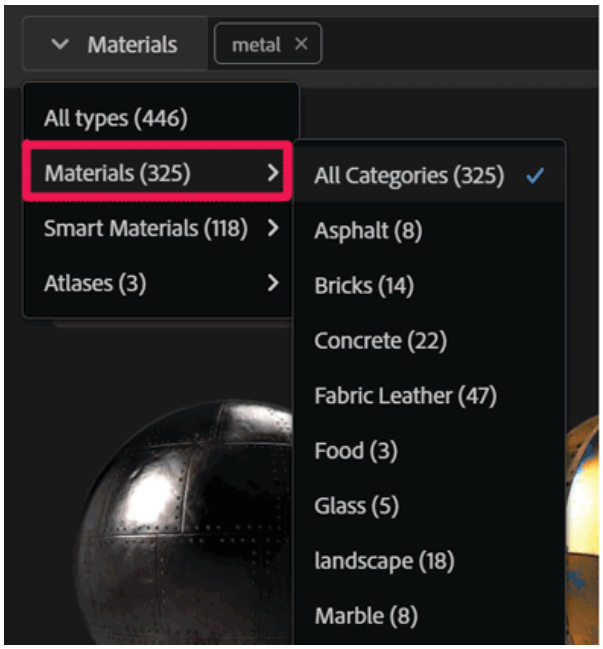
Now you can find the Add-on in its own “Substance 3D” tab in the Option Panel. Click on the icon to open the “Substance 3D Community Assets” page, which has a lot of free materials for you to choose from. Note, that you must be logged into a (free) Adobe account to download these.

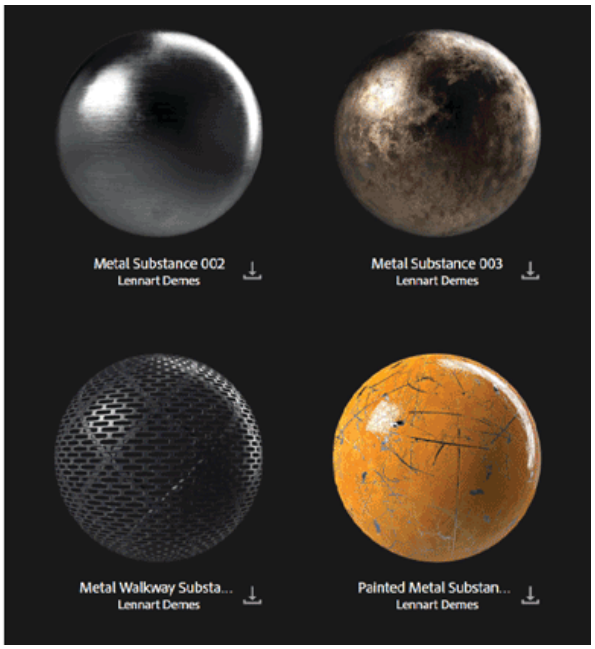




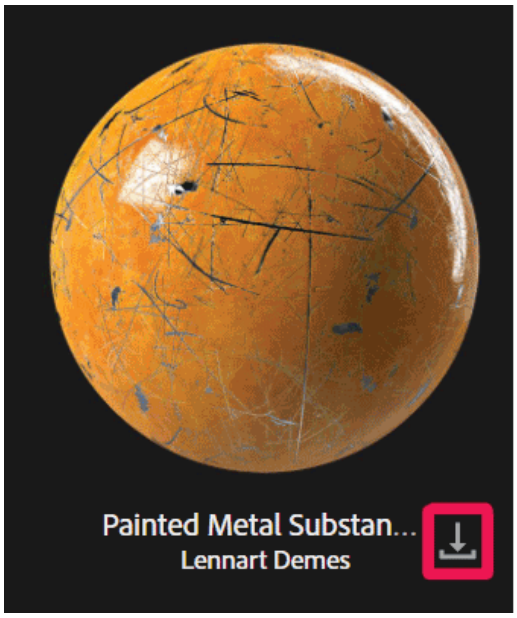
Click here to return to the index.

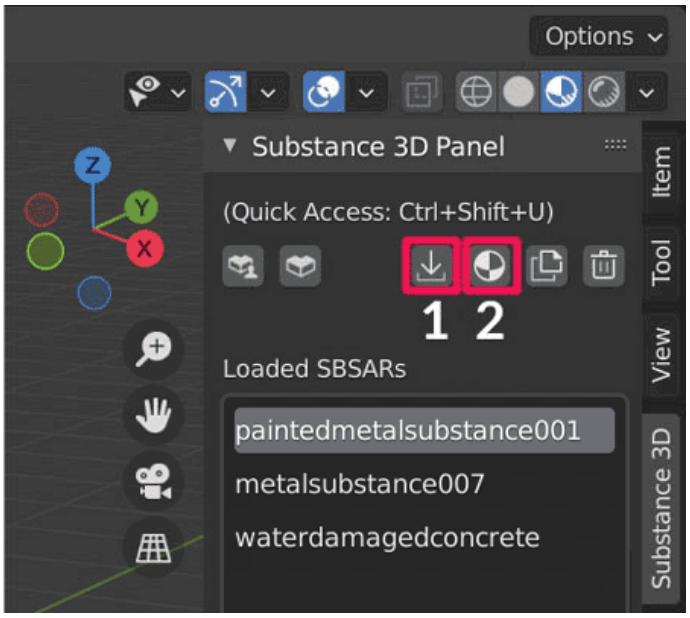
Choose a material you like and click on the download icon. Remember that some of these materials can be pretty big in terms of file size. Also, note that you can only use normal materials with the file format **.sbsar** and not "smart materials" with the file format .spsm. Luckily, there's **325 normal materials** available!



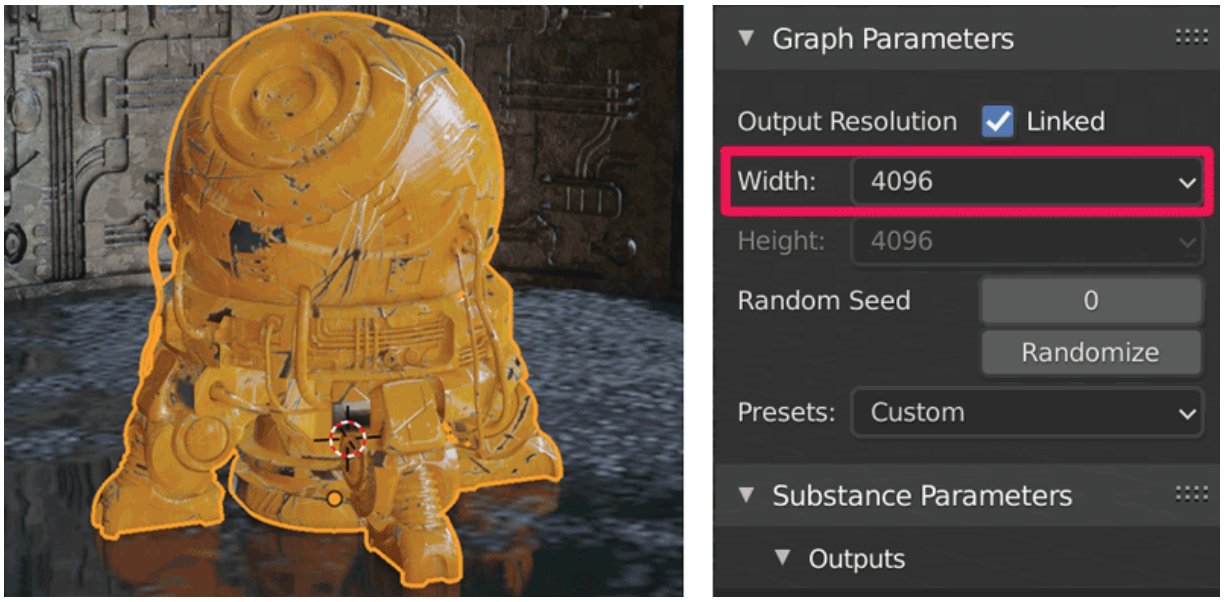


Once you've downloaded the material (I went with the yellow scratched "Painted Metal"), click on the Load Substance Material icon (**1 in the picture below on the right**), select an object you want to apply the material to and click on the Attach SBSAR icon (**2 in the picture below on the right**). Now your object has that material applied to it.

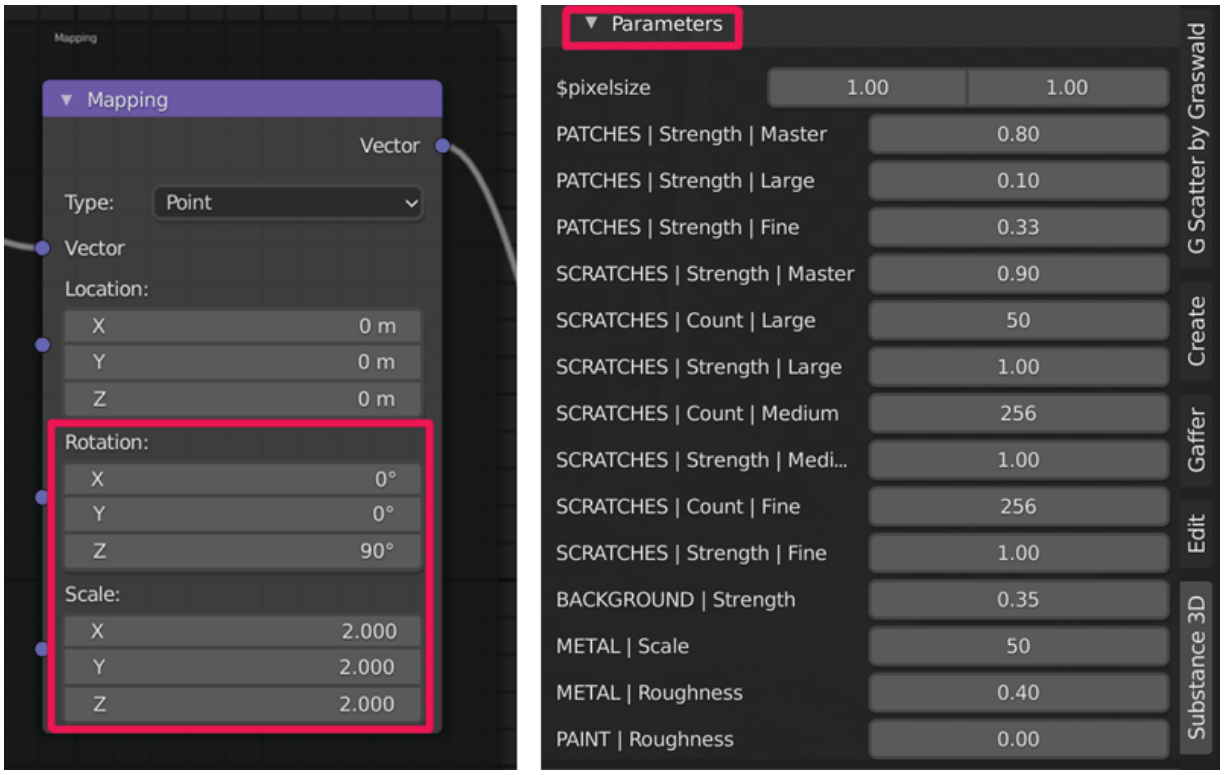




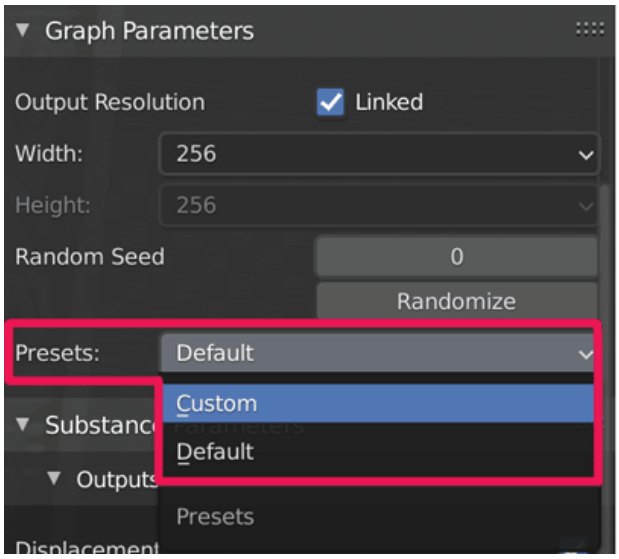
Below you can see that I've applied the material to my own custom shader totem. First thing to do is set the **Output Resolution Width** higher, because by default it's set to a paltry 256 pixels. Come on Adobe, you underestimate our computers! **4096** is more like it.

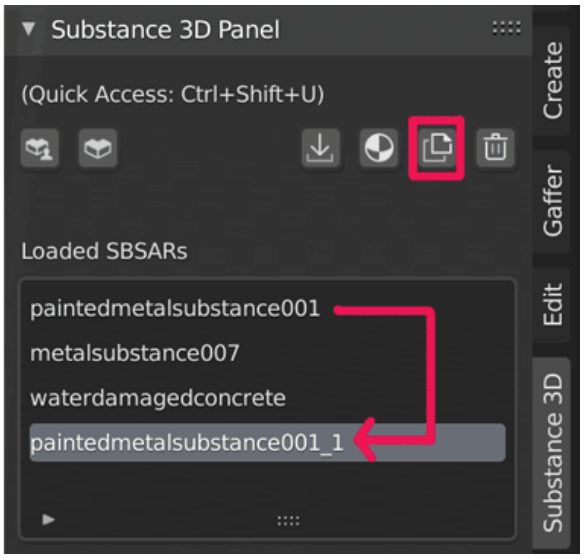


After that, it can be a good idea to change the **Scale** and **Rotation** in the **Mapping node** in the Shader Editor window to your liking. After that, it's a matter of tweaking all the parameters that come with the material you're using. Each Substance material has its own specific parameters. In this case you can adjust many attributed of the **scratches**, for example.

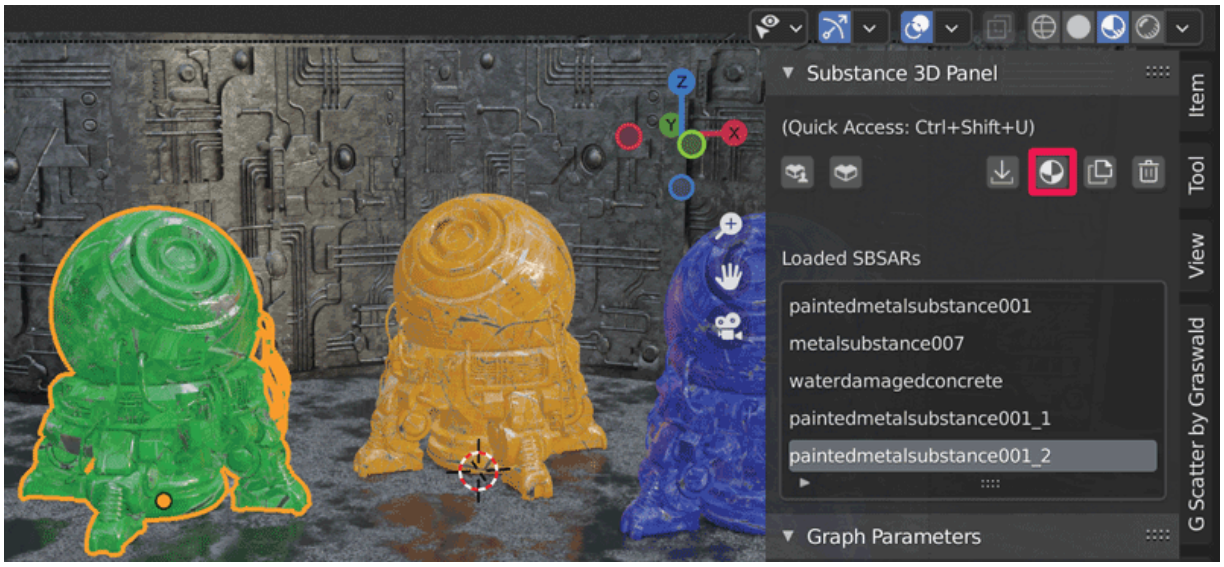


To reset the material to its default settings, choose **Preset > Default**. If you switch it back to **Custom**, you'll get the changes you made manually again so you can always switch between the two. You can also make a duplicate version of the material by clicking on the **Duplicate Selected SBAR** icon.





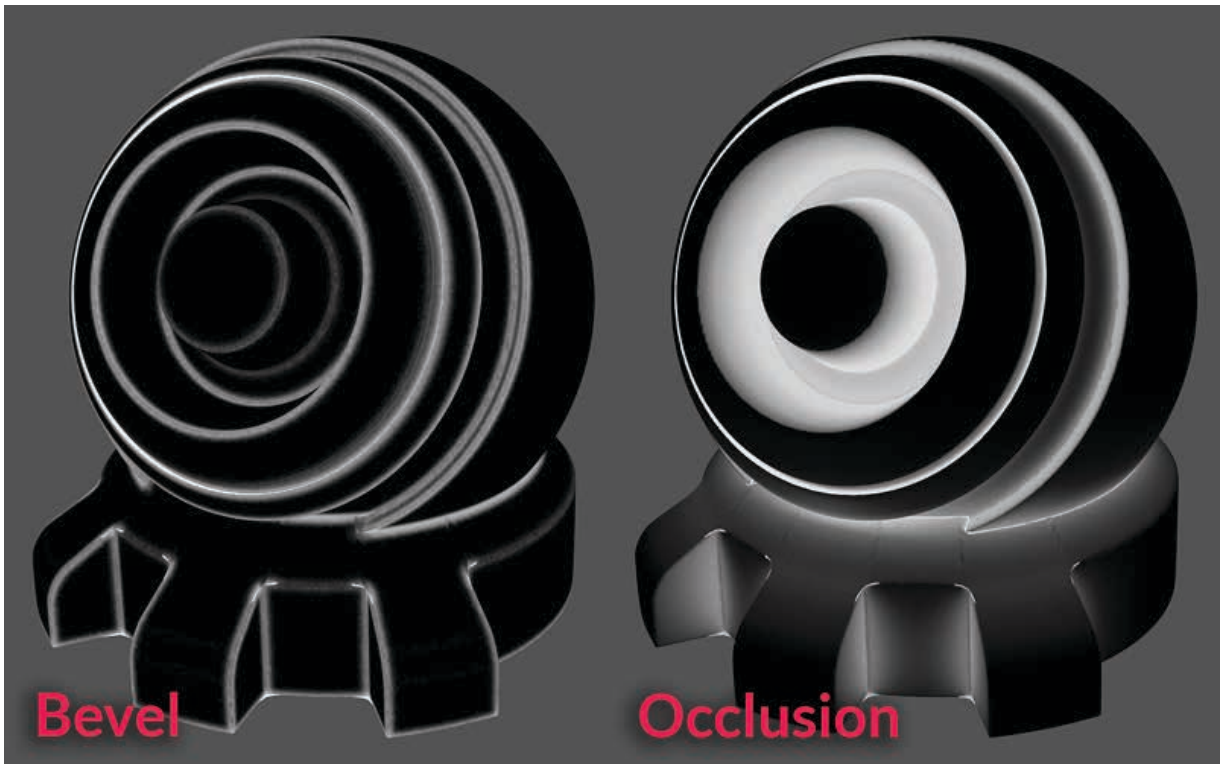
You can then change the values for that material, for example if you want to add a slightly different version of it to a different object in the scene. Just remember to select that object and **click the "Attach SBAR" icon** again (the globe icon). That's really all there is to it!



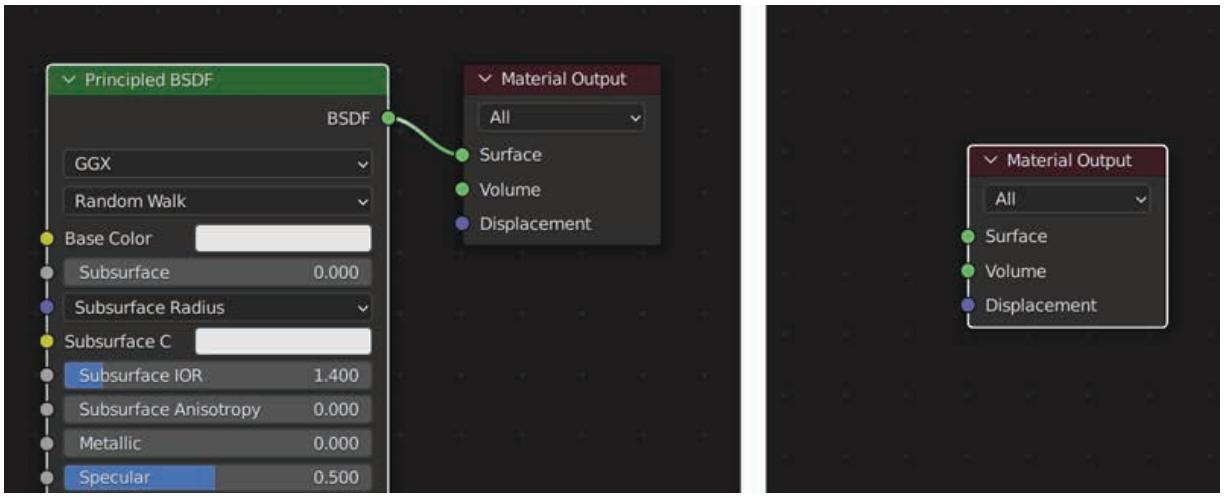
How to bake Bevel and Occlusion maps

If you want to have fine control over edge wear effects that works in both Eevee and Cycles, you should first bake maps that identify edges. This is also how programs like Substance Painter work - first, you need to bake what is called “mesh maps” before you can take advantage of real-time “smart” materials and add effects like edge wear.

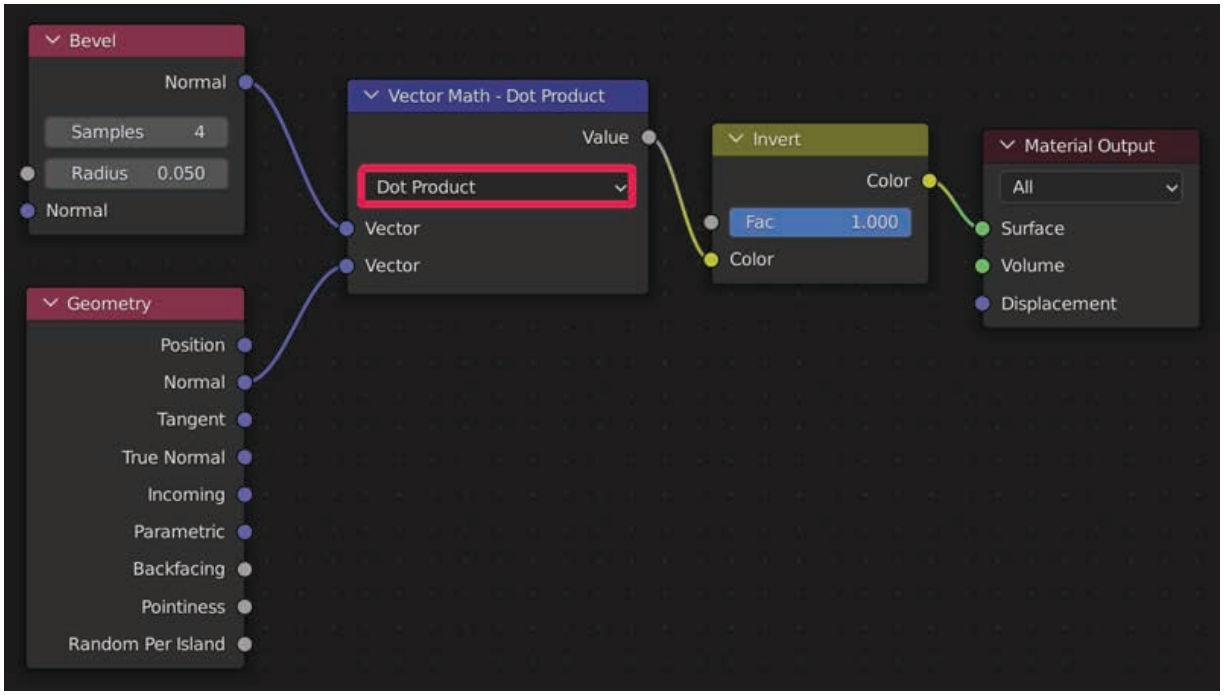
For edge wear, the most important maps (or masks, as they mask one material to reveal another) we can bake, are the **Bevel map** and the **Occlusion map**.



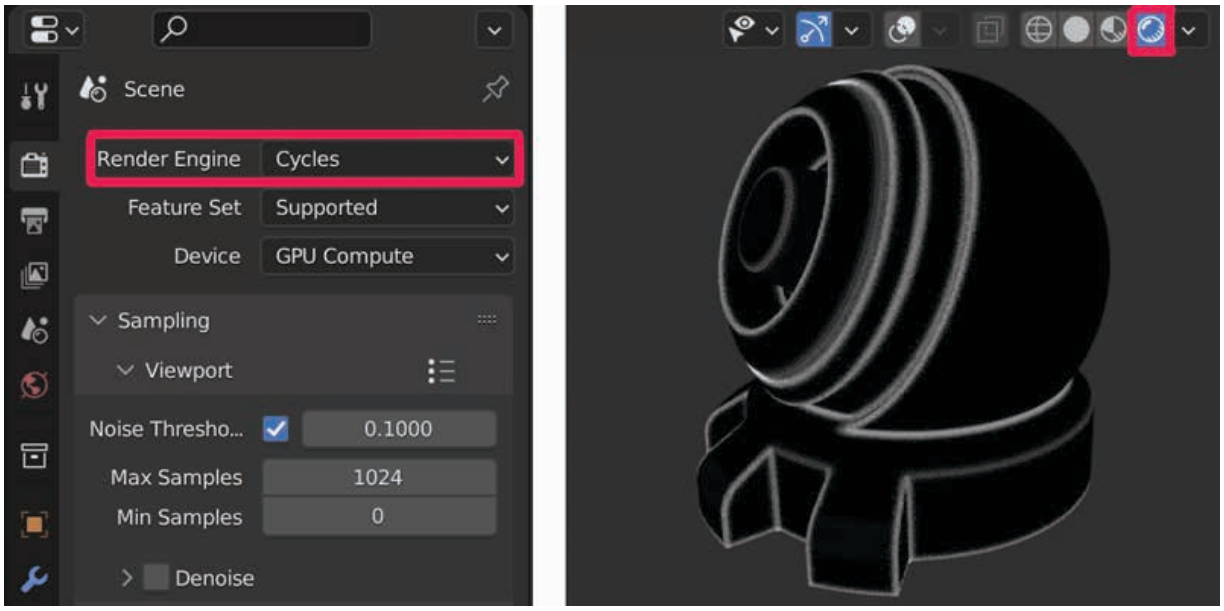
Once we've created these node set-ups, we can save them as node groups for later re-use. So you only need to create them once. Or, you can download the ones I've already made here. Then, you can always **append** them later. Assuming you've UV unwrapped your object and that it has a material, select its **Principled BSDF node** and delete it by **pressing Ctrl+X**. Now you only have the Material Output node.



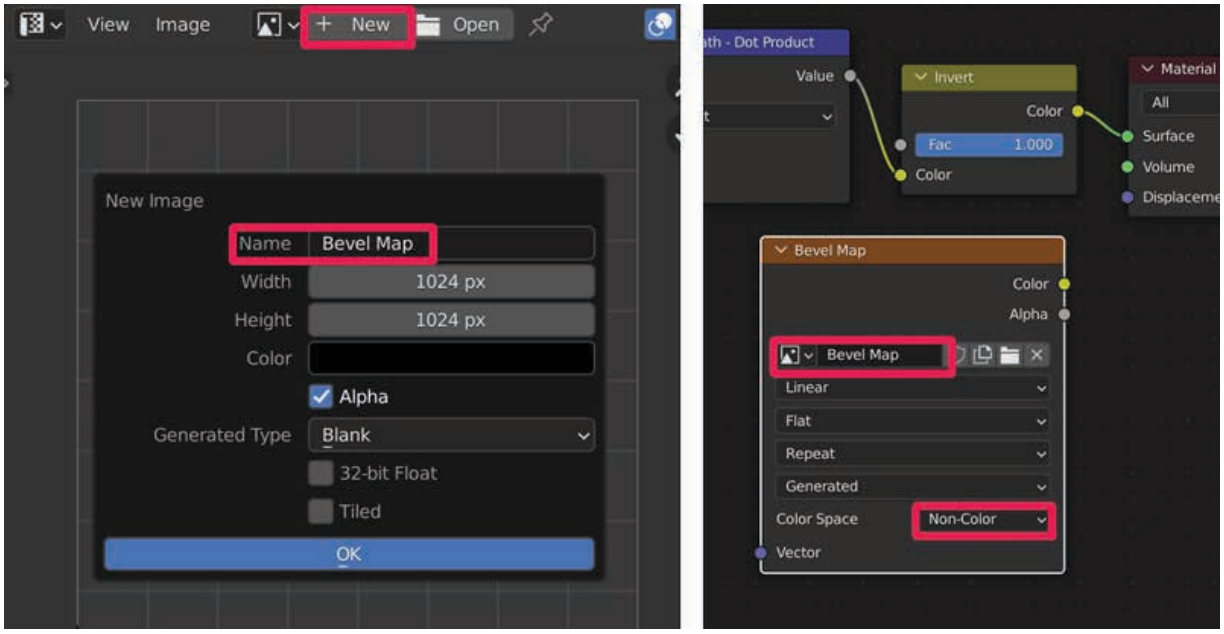
Then, add and connect these nodes. Set the **Vector Math node** to "Dot Product". The default values of the **Bevel node** are fine.



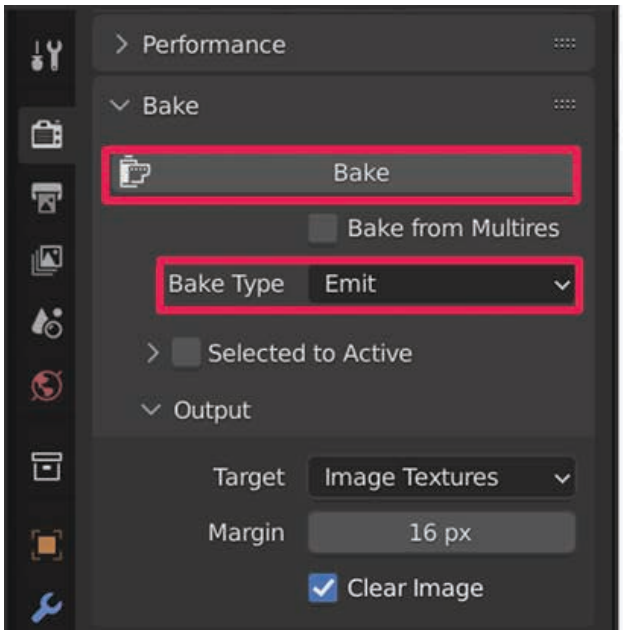
Now, as long as you're using the Cycles render engine and the 3D viewport is in **Rendered preview** mode, you'll see the edges as white and the rest of the object as black.

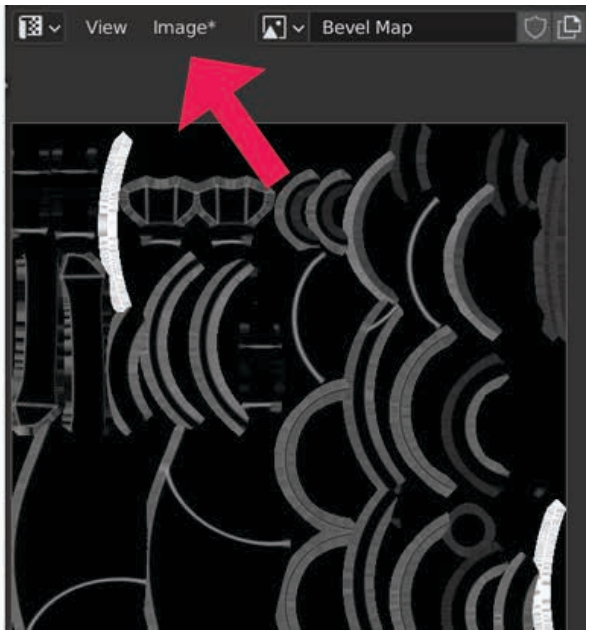


Now, create a new texture - 1024 x 1024 or 1K resolution is fine - and name it “Bevel Map”. Add an Image Texture node to the material, but don’t connect it to anything. Load the Bevel Map in it, and set it to Non-Color.

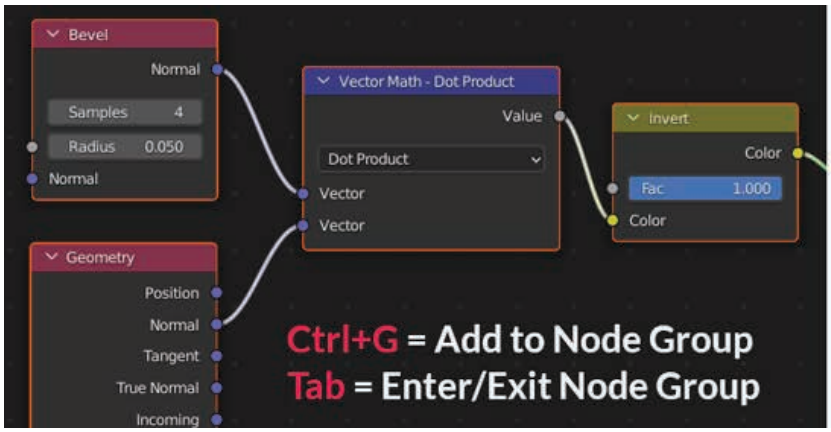


Make sure your object is selected. Now set Bake Type to **Emit** and click on **Bake**. After it’s done, save the image!

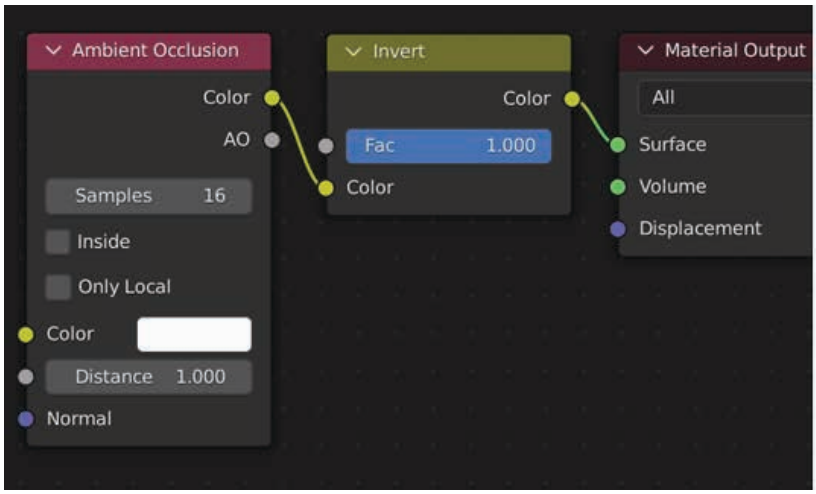




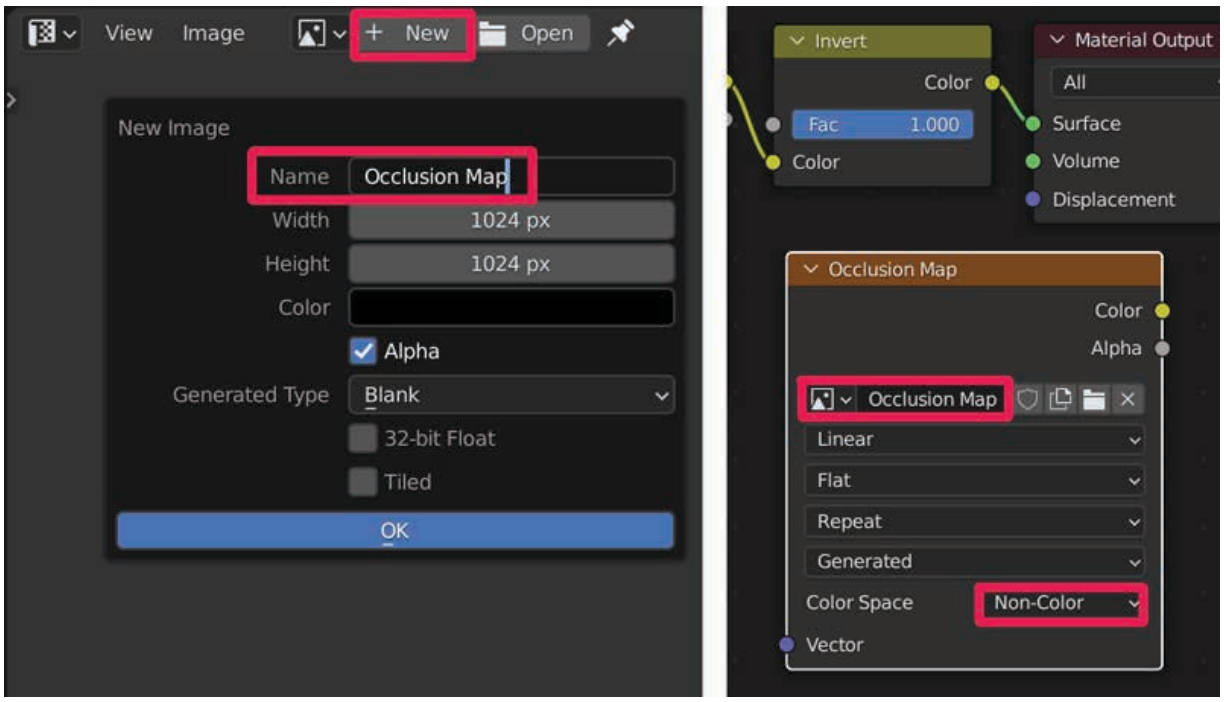
Now if you use that saved image as a texture, you'll be able to see the edge mask in Eevee as well as in Cycles. Select the nodes and **press Ctrl+G** to add them to a Node Group. Then **press Tab** to exit the Node Group, and give it a clear name.



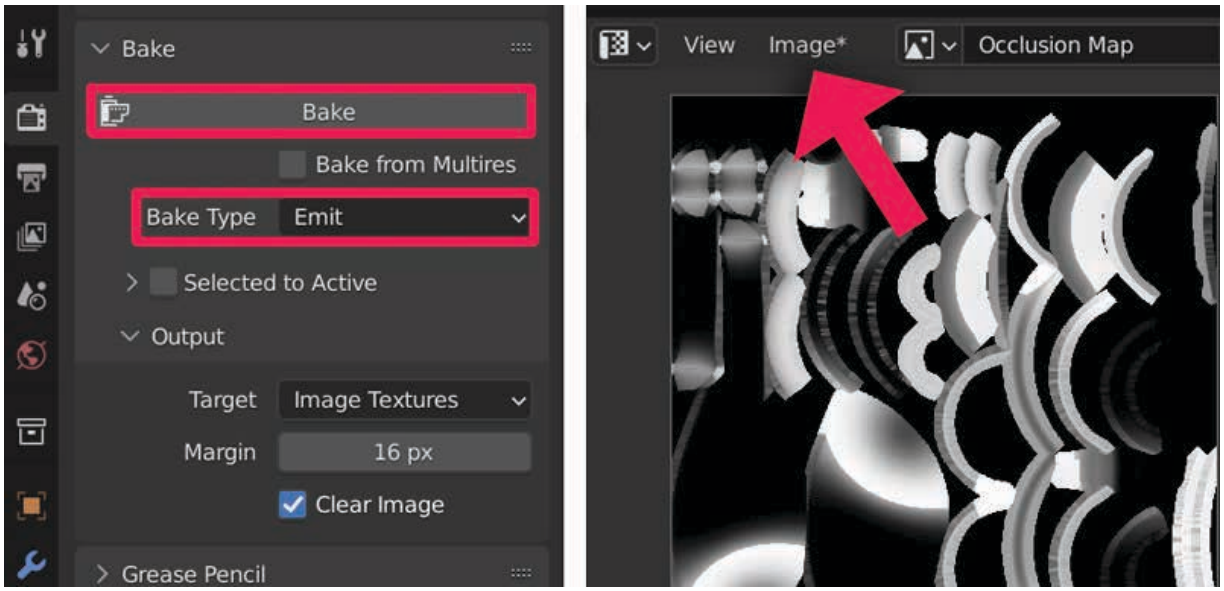
Next, we'll do the same but for Ambient Occlusion. This one is very simple. Just add an **Ambient Occlusion node**, and an **Invert node**. That's all!



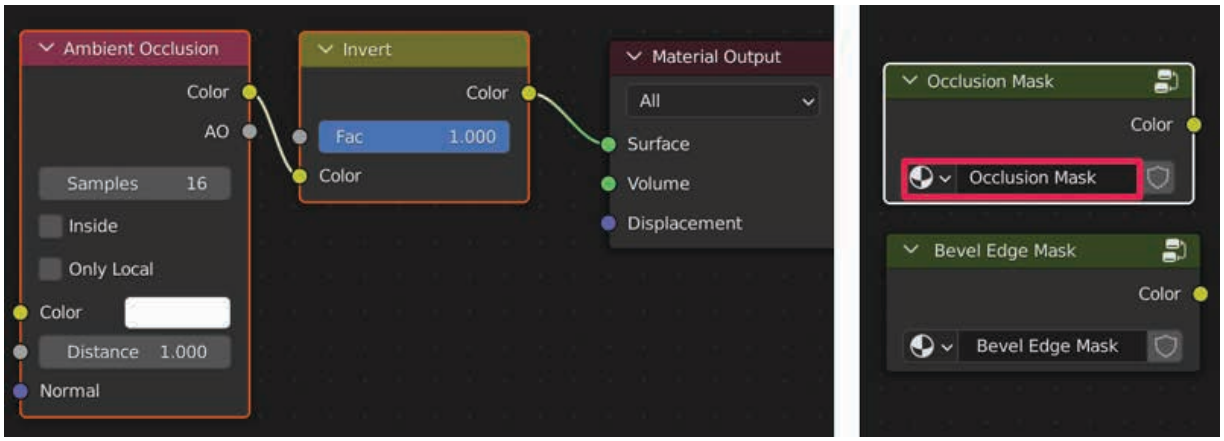
Create a new texture, name it “Occlusion Map”. You can reuse the same image texture node as before, as long as it’s not connected to anything. Load the Occlusion map in it, and set it to **Non-Color** (Blender defaults back to sRGB every time you load a texture).



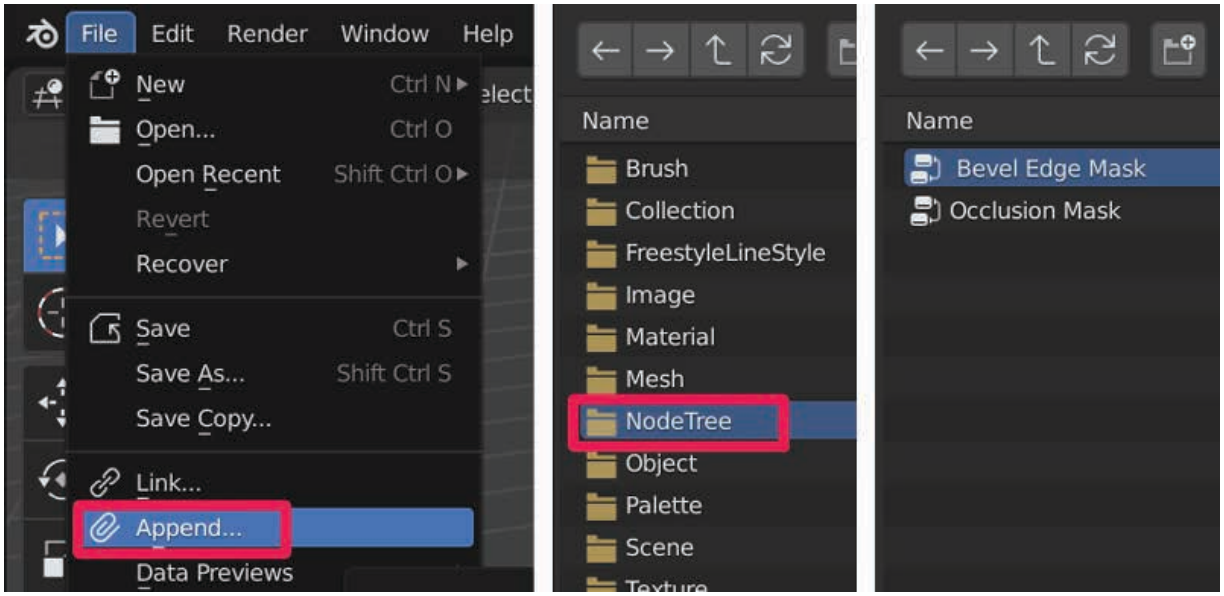
Then like with the Bevel map, select the object, **Bake** it and **Save** it.



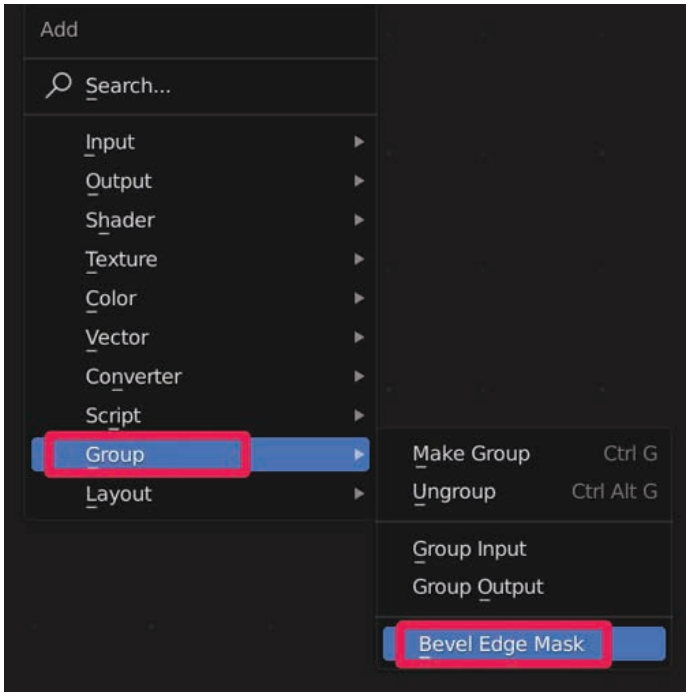
Like before, select the nodes for the Occlusion mask and **press Ctrl+G** to add them to a group, **Tab** to exit the group and then give it an appropriate name.



You don't need to create "fake users" for Node Groups, they will be saved with the blend file. So you can **append** them later to bake objects in your own projects.



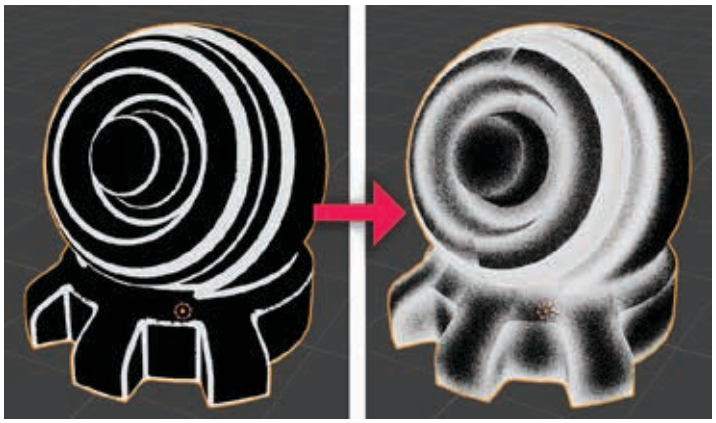
Once you've appended it, you can **press Shift+G** and go to **Group**, then it should be listed there.



Blur Node

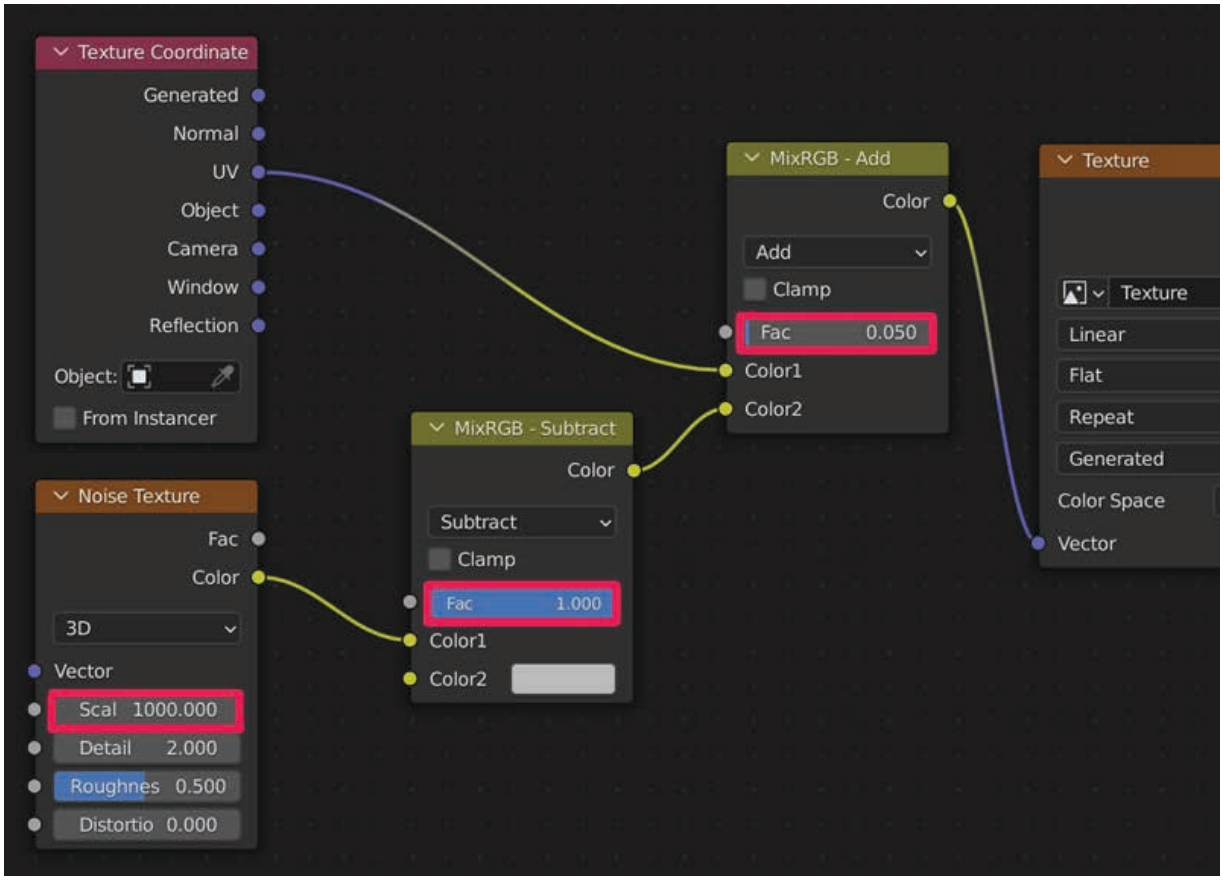
The ability to blur textures and texture masks is a very convenient, and some would say essential, feature.

A feature, Which unfortunately doesn't come with Blender by default. However, fear not: with a small amount of nodes grouped into a Node Group, we can create our very own "Blur Node".

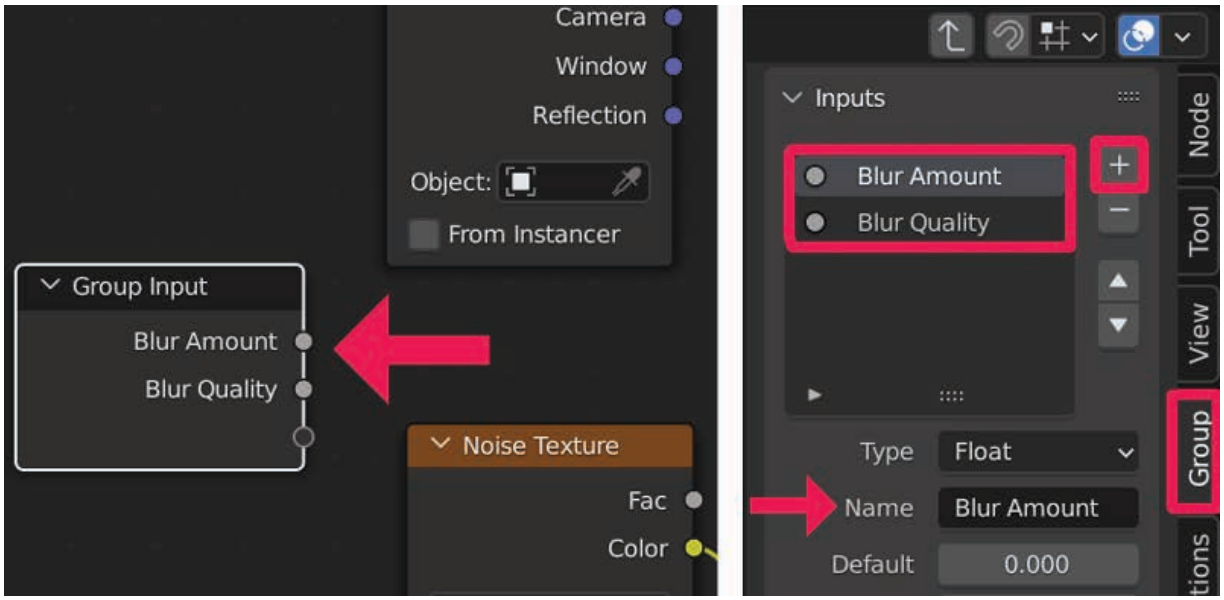


This method was first explained in a talk by Bartek Skorupa for a Blender conference, and the recording of which is also very interesting and delves into much more detail.

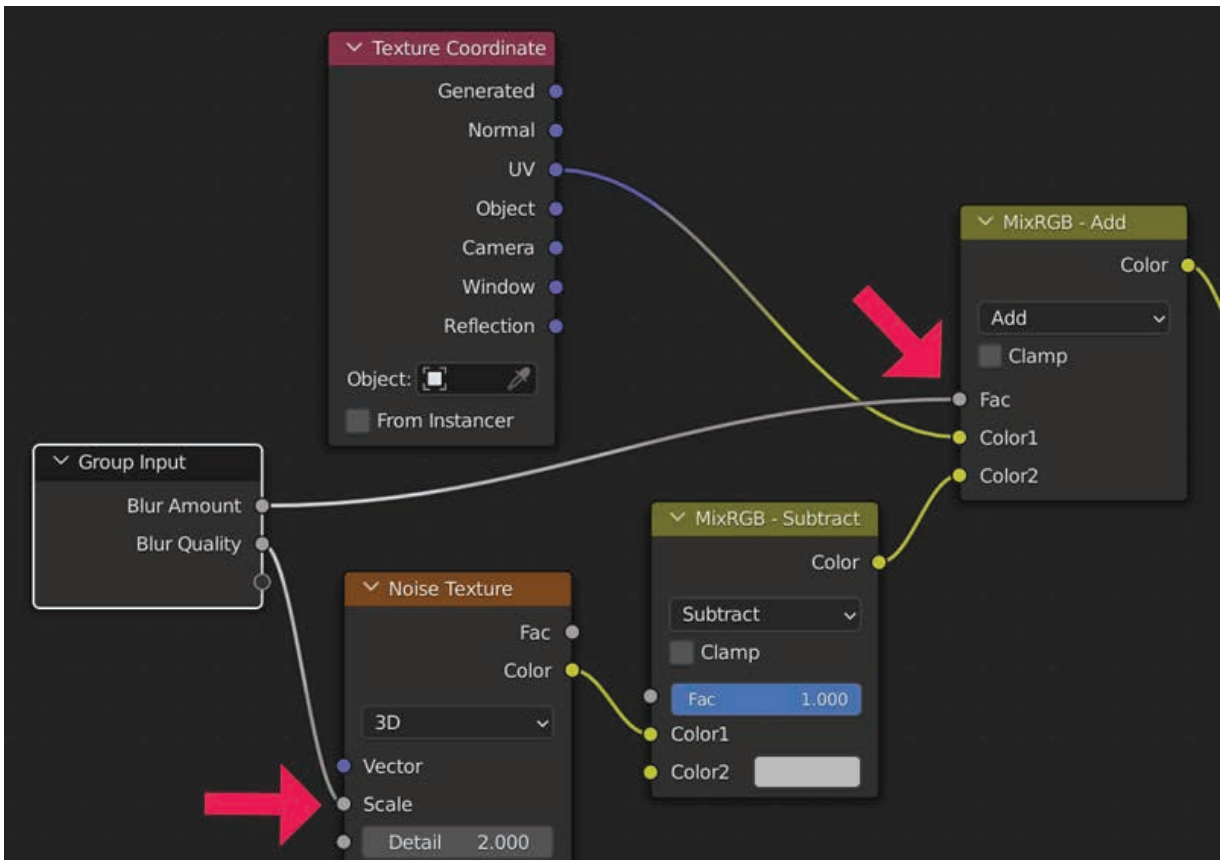
Anyway, onto the nodes! Below are the nodes you should add, for starters. These plug into your texture, in addition to the regular Texture Coordinate node. Note, the **Add** and **Subtract** ones are **MixRGB nodes**, *not Math nodes*! Although that would work too, but the MixRGB nodes give us value sliders which we can use to control the blur amount. Important: **Set the Subtract Fac value to 1**.



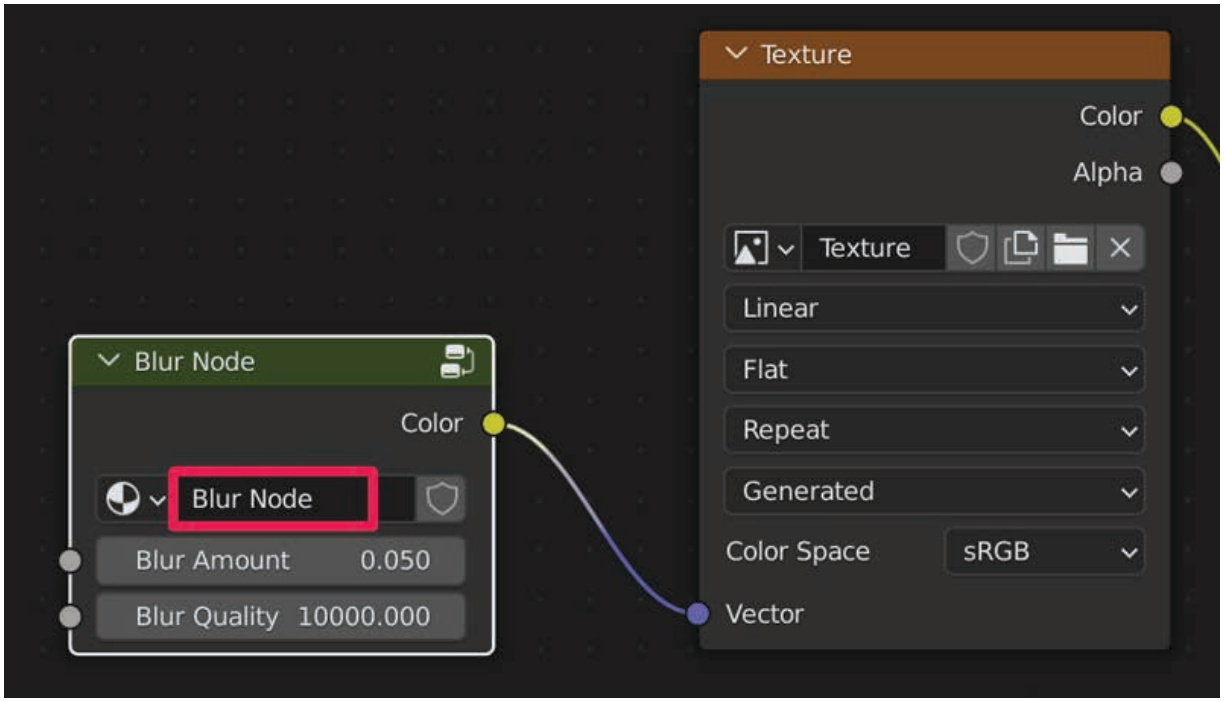
Select the Texture Coordinate, Noise Texture and both MixRGB nodes and **press Ctrl+G** to add them to a Node Group. You are automatically inside the node group, which is good because we want to adjust the **Group Input node.** In the Group tab in the Option panel, add an Input by **clicking on the + icon**. Name one "**Blur Amount**" and the other "**Blur Quality**".



Connect them as shown below. The **MixRGB Add** factor controls the amount of blurring, and the **Scale of the Noise texture** controls the quality. Less quality means more grainy, which can also be interesting sometimes - as shown later.

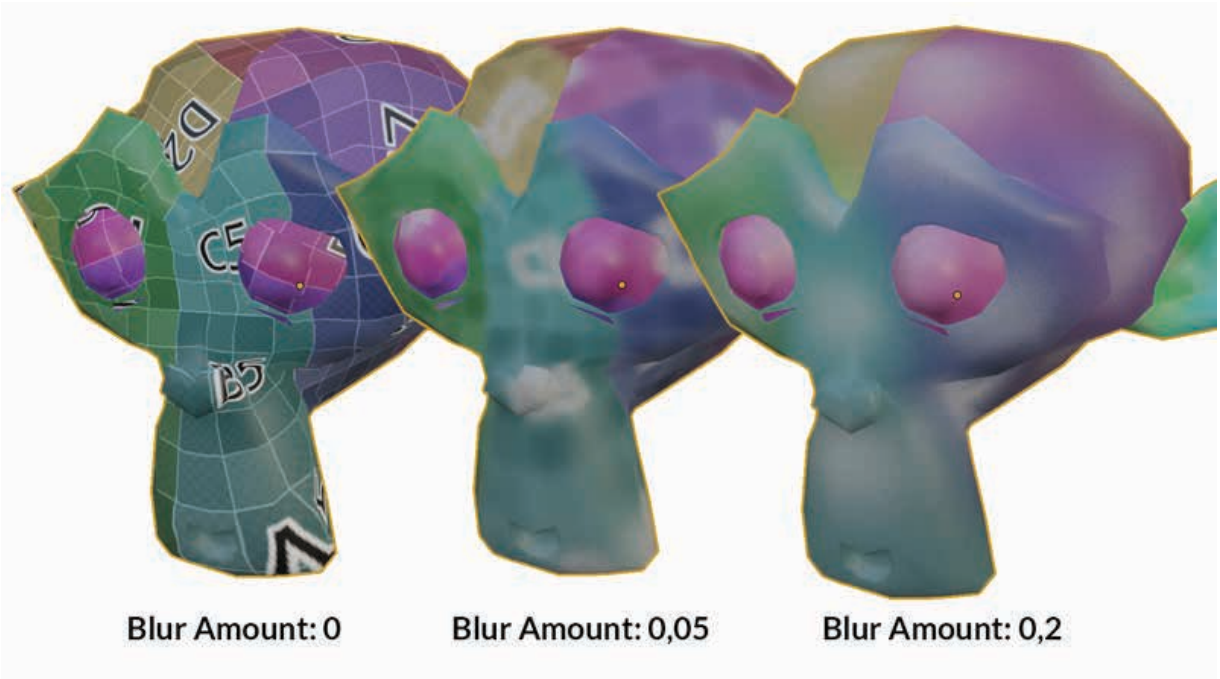


Press Tab to exit the Node Group. You can always select the Node Group and press Tab to enter / exit it, in case you need to make any changes. For example to change the mapping coordinates, depending on whether your object uses UV or Generated coordinates, for example. But with the two most important values connected to a Group Input, you should be able to use it as shown below. Final thing to do is adjust the name to "**Blur Node**".



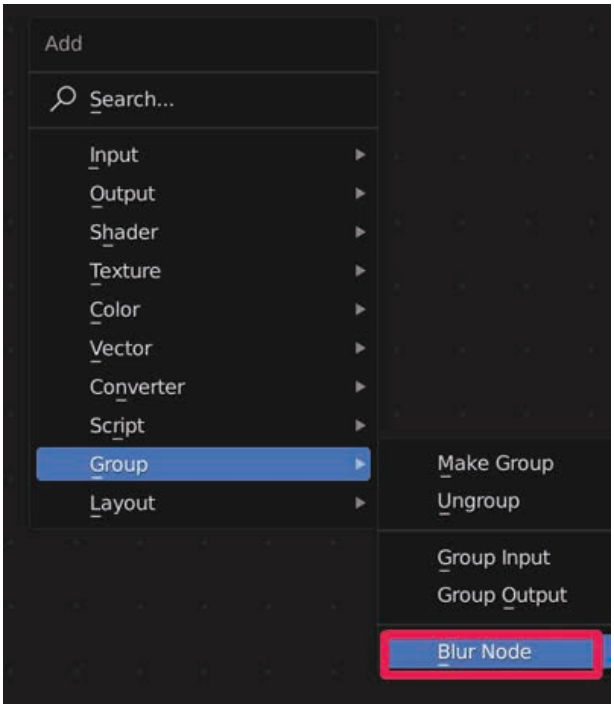
Below, you can see the difference the **Blur Quality** makes. For the higher values than 1000, it really depends on how close you plan to get with the camera whether you need even higher values or not.

The **Blur Amount** speaks for itself, nevertheless here is a comparison:



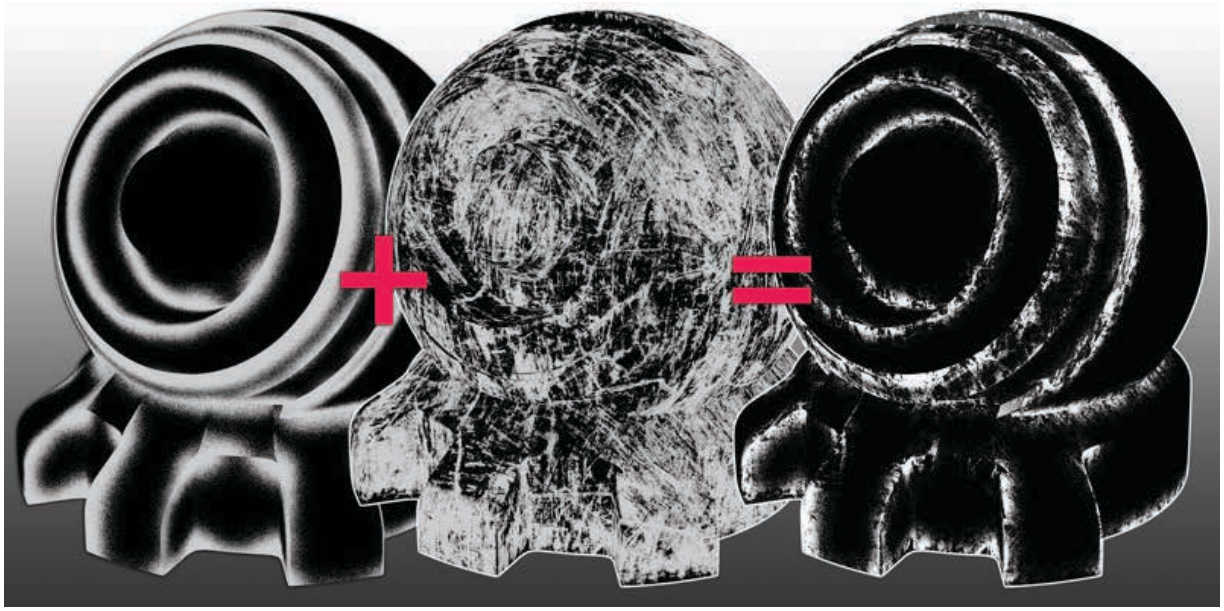
Save the Blend file you made this node group in.

Now any time you need it, you can **append** it from that file, and **press Shift+A** to find it in the Add Group menu.

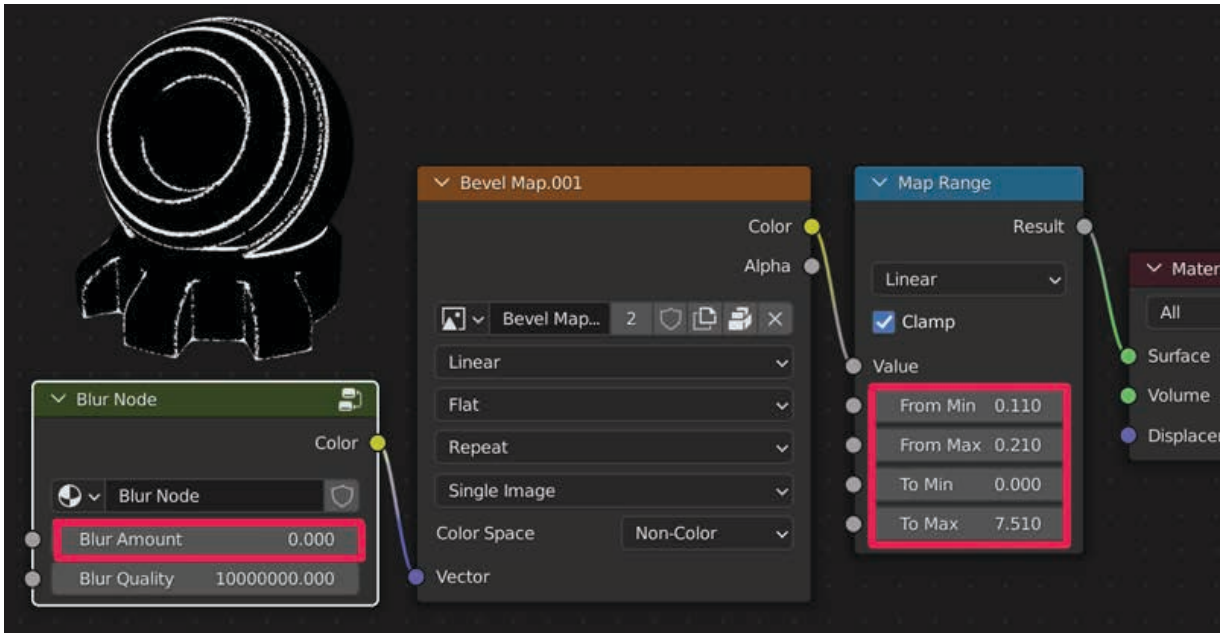


Substance Painter-like Edge Wear Effects

In previous topics we've learned how to **bake Bevel and Occlusion maps,** and we've created our own **Blur node**. These are the ingredients for creating more interesting edge wear and dirt effects. The key is using photographic grunge textures.

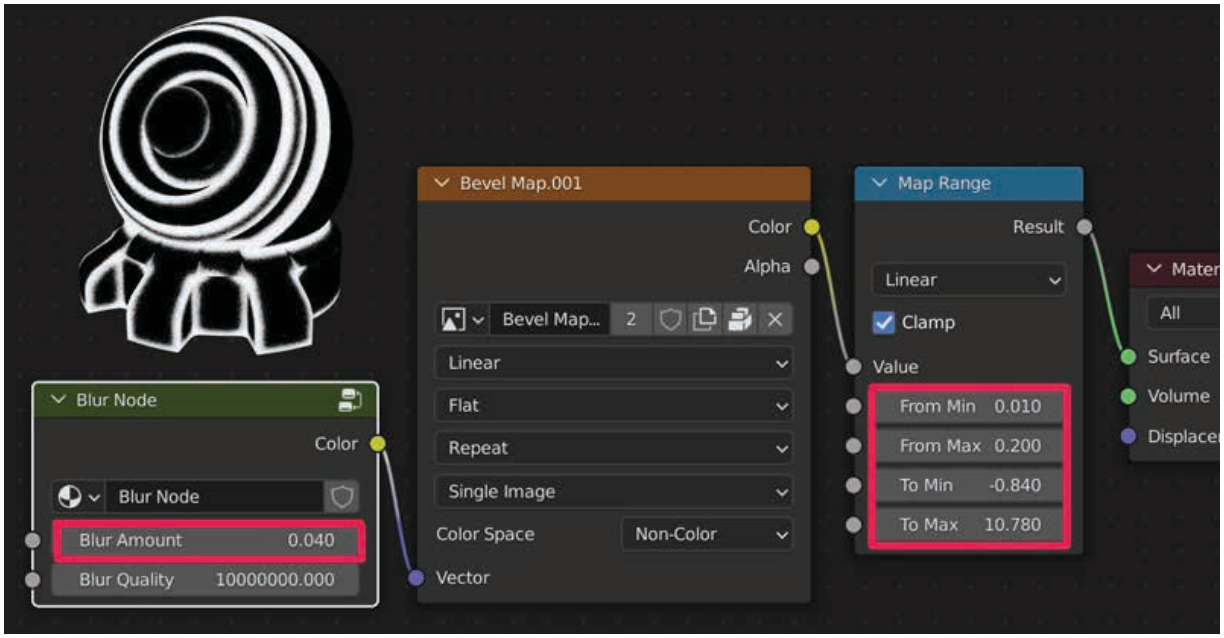


The result can then be used as a mask to mix for example, a painted metal surface with rusted or exposed metal edges. First, by adding a **Map Range node**, we can add **contrast** to the Bevel Map. So when we decrease the Blur, we can a thinner Bevel mask.

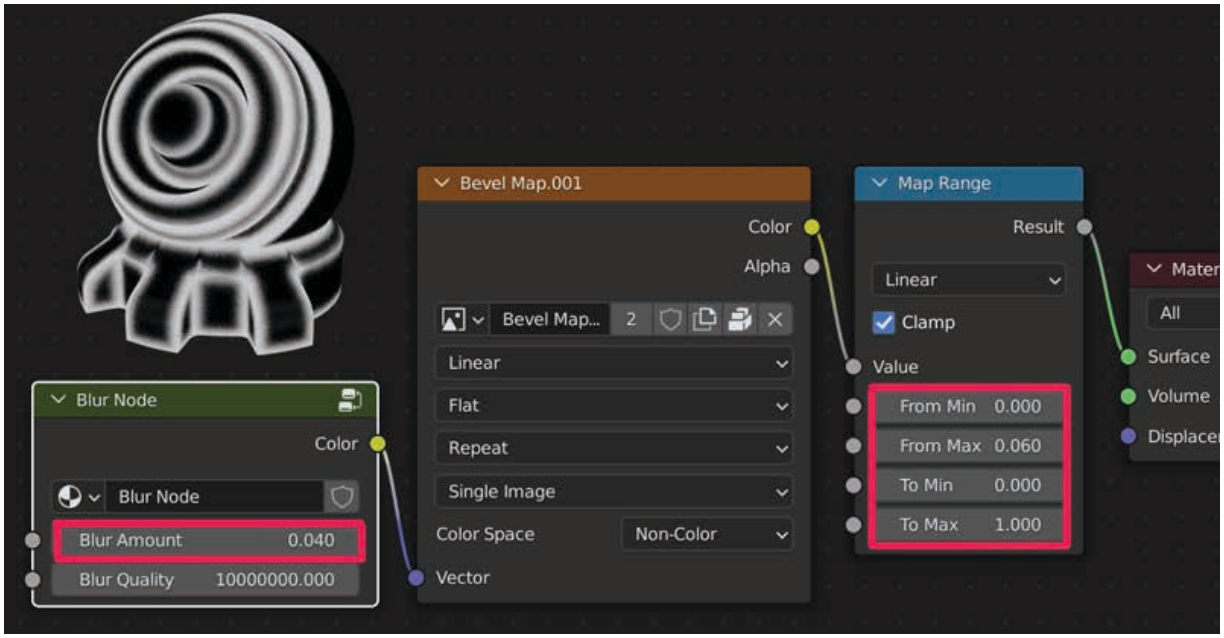


Click here to return to the index.

By increasing the **Blur Amount**, with strong contrast from the Map Range node, we can "grow" the Bevel mask.

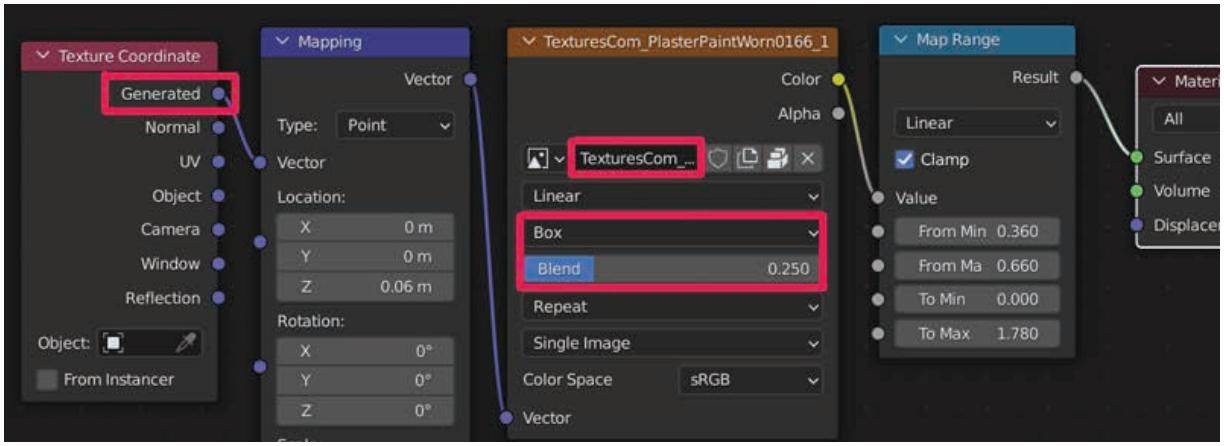


And by lowering the contrast added by the Map Range node, but with a lot of Blur, we also grow the Bevel mask but in a softer way. These can all be useful in different cases.

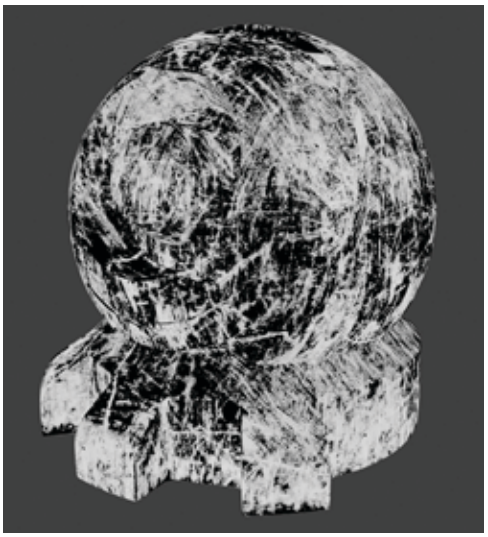


To make it even more interesting, we can add another texture to it. And instead of a procedural noise texture, let's go with something more real. If you search for "grunge" on the website Textures.com, you get some interesting results. I like "PLASTERPAINTWORN0166", a photographic texture that you can download for free.

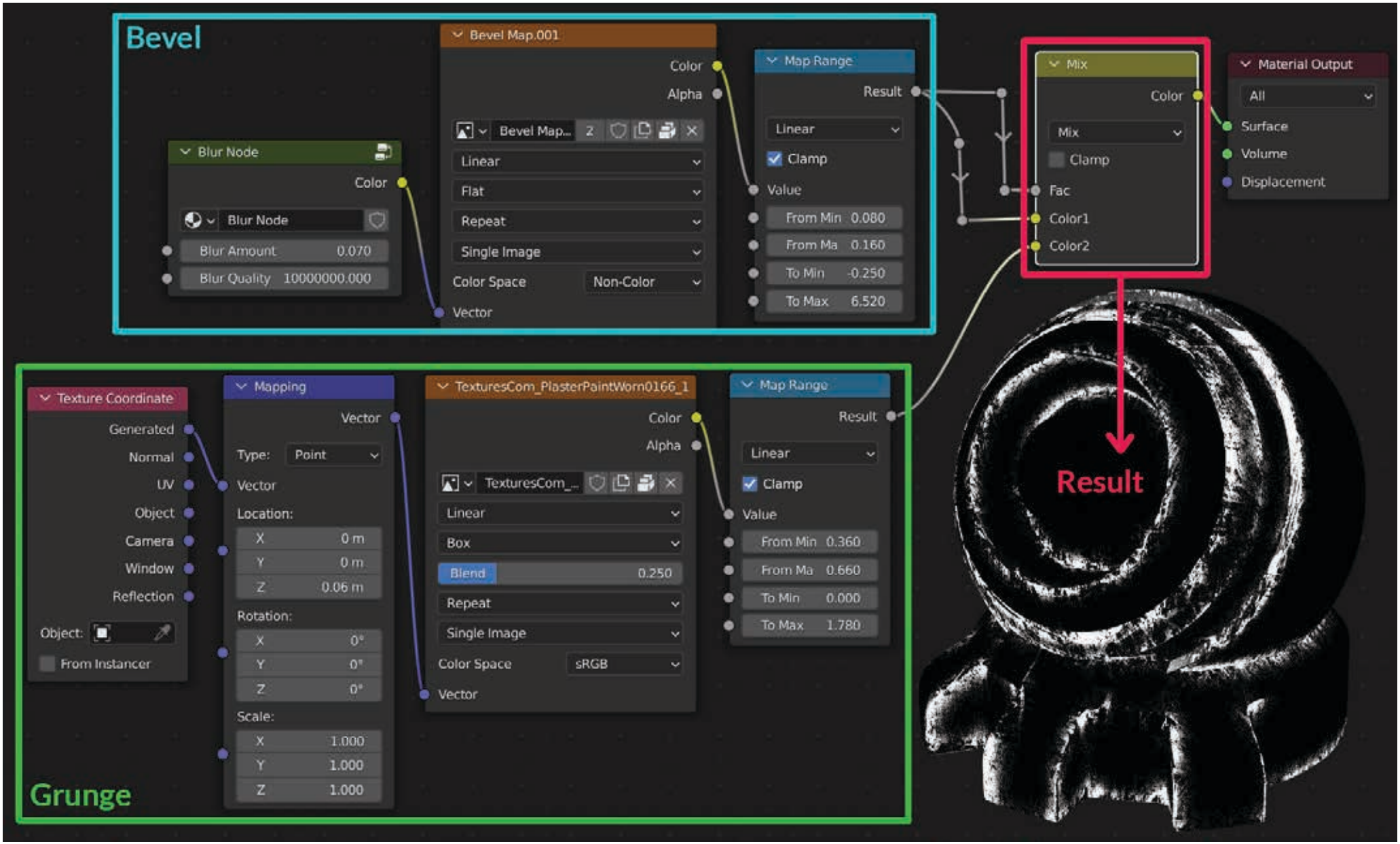
Add an **Image Texture node** and load this plaster texture in it. With the Texture node selected, **press Ctrl+T** to add Texture Coordinate and Mapping nodes *(assuming you have Node Wrangler add-on enabled, otherwise you'll have to add both nodes manually)*. Set it to Generated and Box Mapping. Adding a **Map Range node** turns it into a black and white image and gives you the ability to increase the contrast.



Now, the grunge texture looks like something like this - after some tweaking of the Map Range values.



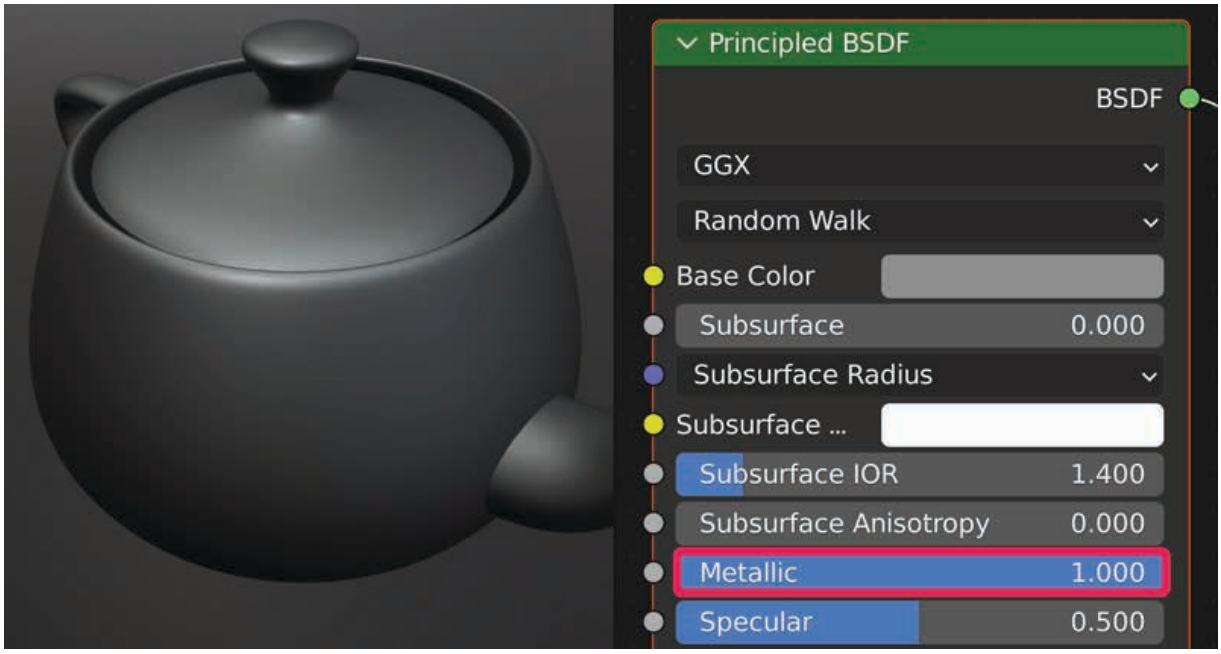
Now, let's combine this grunge map and our Bevel map using a **MixRGB node**. As the Fac input, we just use the Bevel map again. This way, we get something like his:



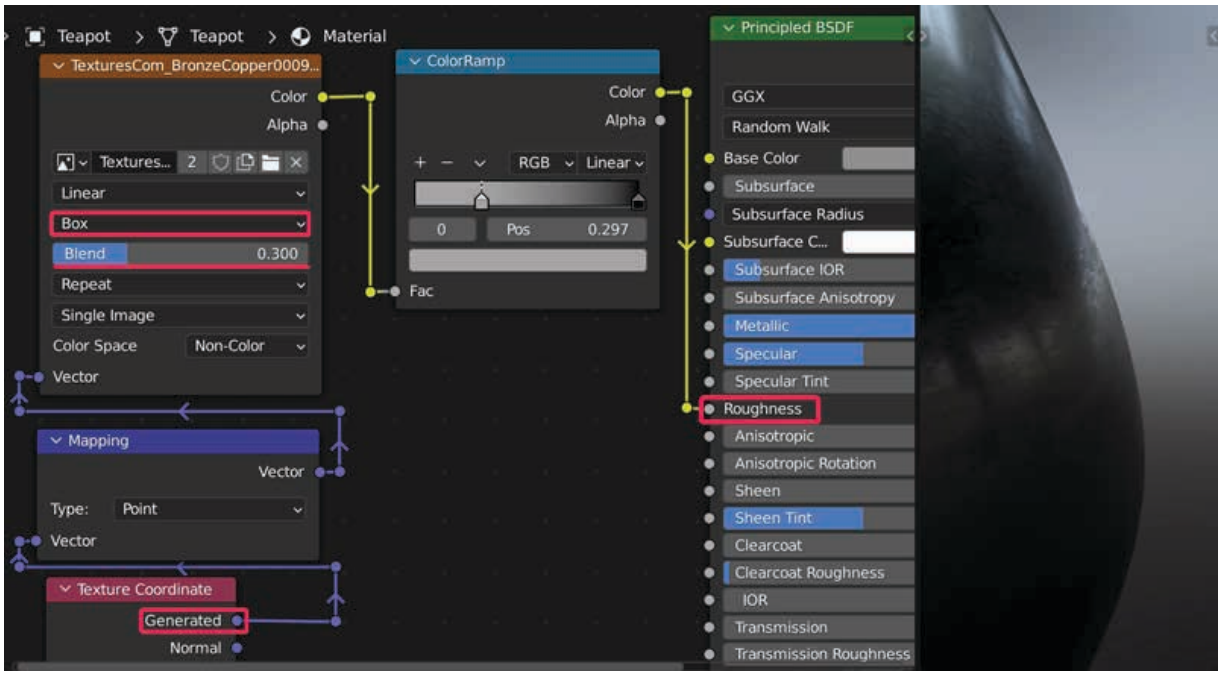
Click here to return to the index.

Dirty Metal material

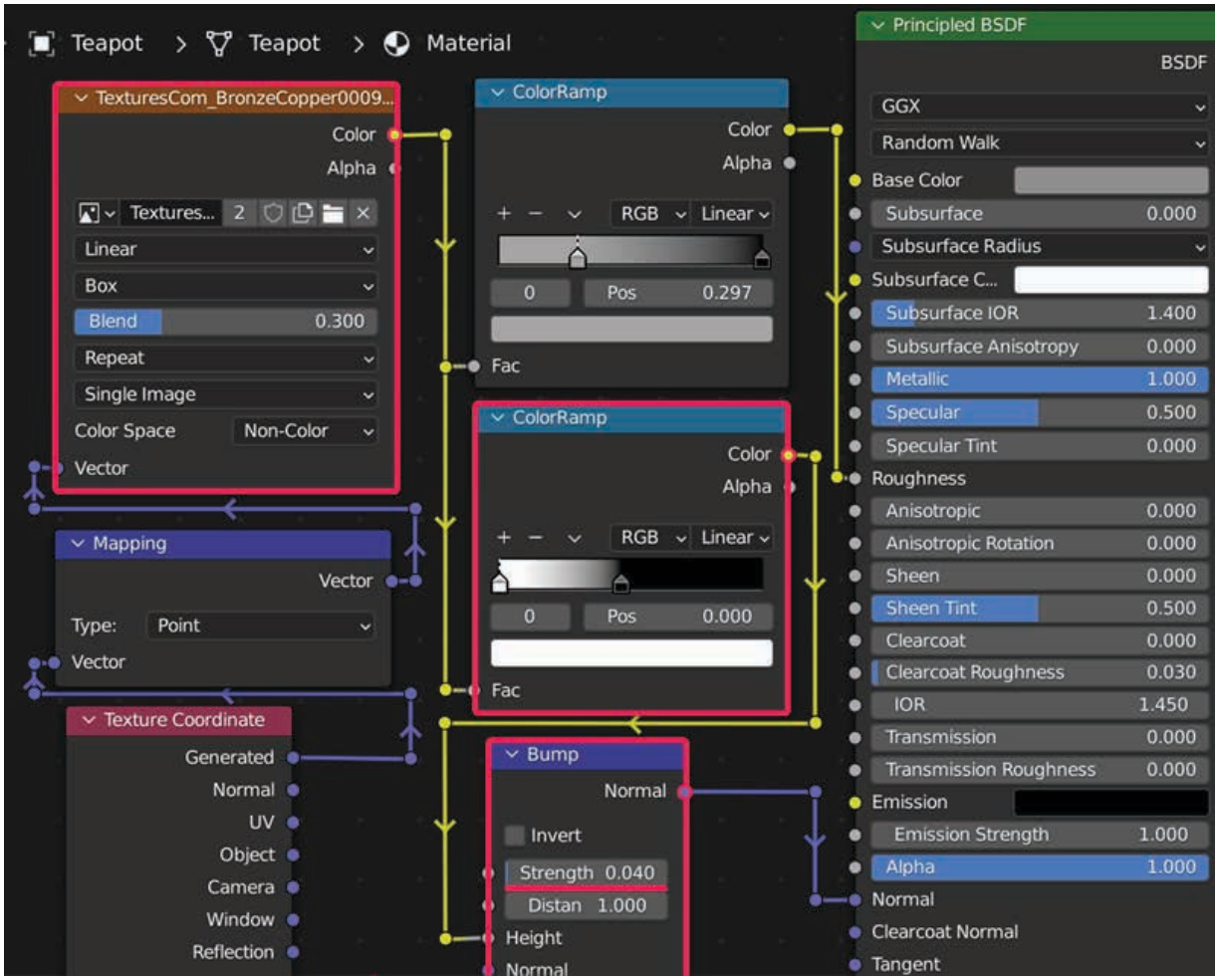
To make a nice old looking Metal shader, create a new material for your selected object and slide the Metallic value to 1.



To make it more realistic you'll need surface imperfections. Plug an image into the **Roughness** input. Set Texture Coordinates to **Generated** and Projection to **Box** with blend: **0.3**. That is, in case your object doesn't have UV coordinates. A **ColorRamp** in between controls the look of the roughness. You can try different images.



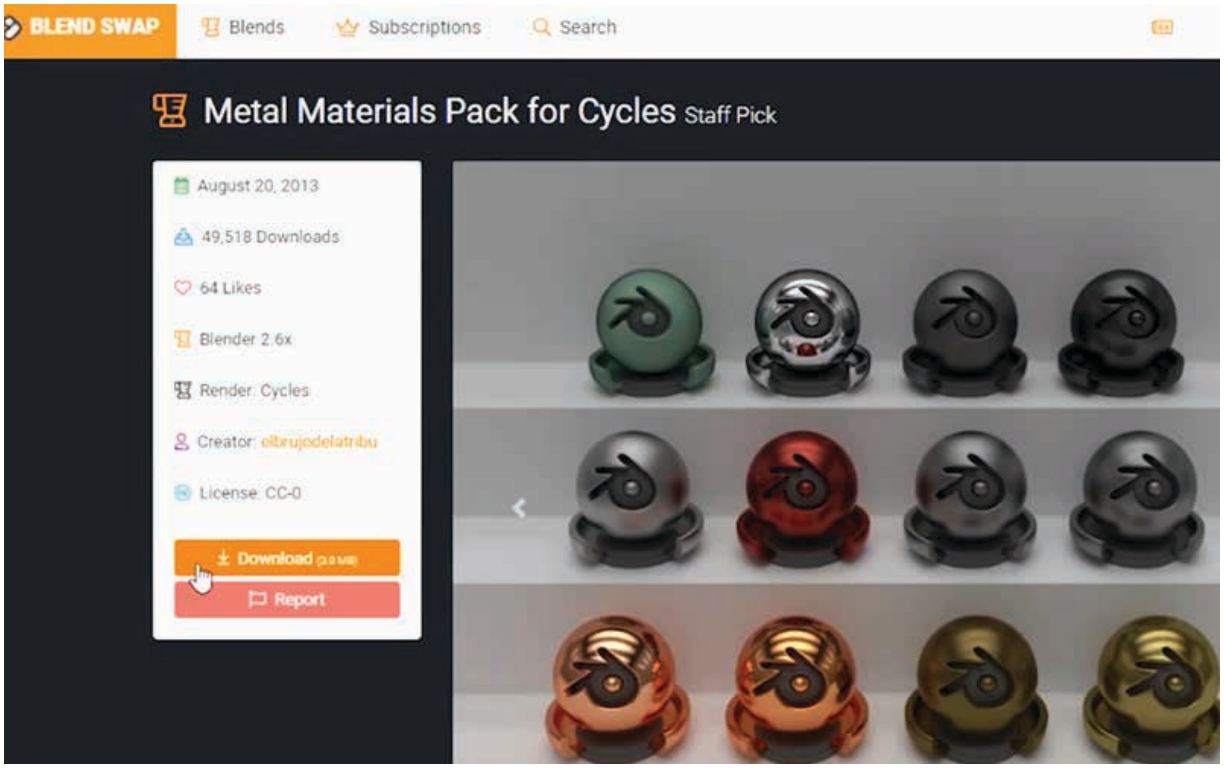
Use that same image to add **Bump** to the surface with a **Bump node**. Use another **ColorRamp node** in between for more control.



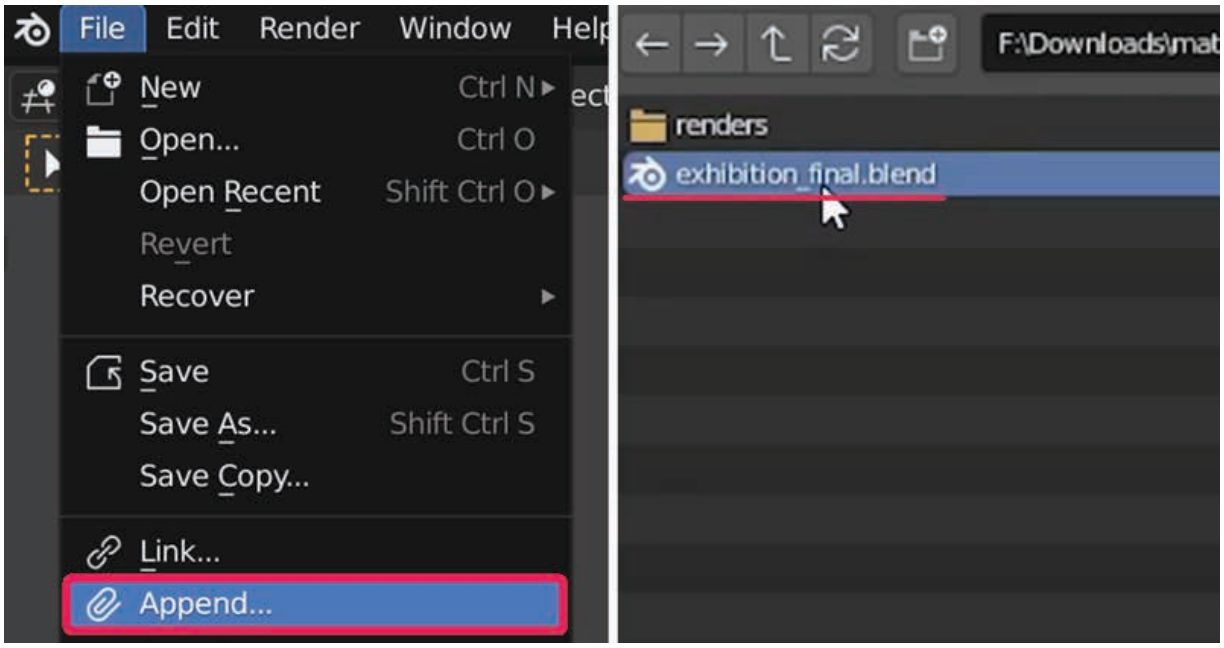
As with other shiny materials, using an HDRI for lighting tends to give more interesting results.

Append free materials

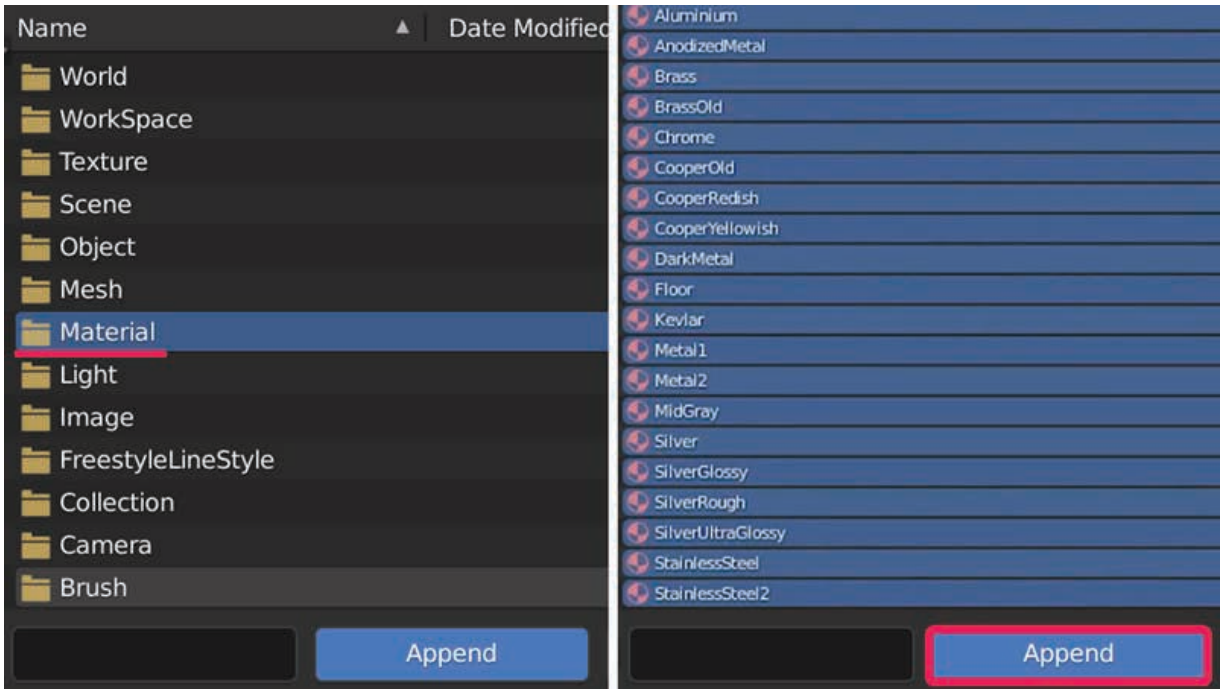
So if you're new to Blender, you may not know how to make fancy materials yet. Well good news you can download plenty of them for free. Blendswap is a good place to start. You can find many material sets there.



Once downloaded, go to **File > Append** to can append materials from that downloaded file to your own Blender scene.



In the appending window, go to **Material** and select the materials you want to append. Then **click on Append**.

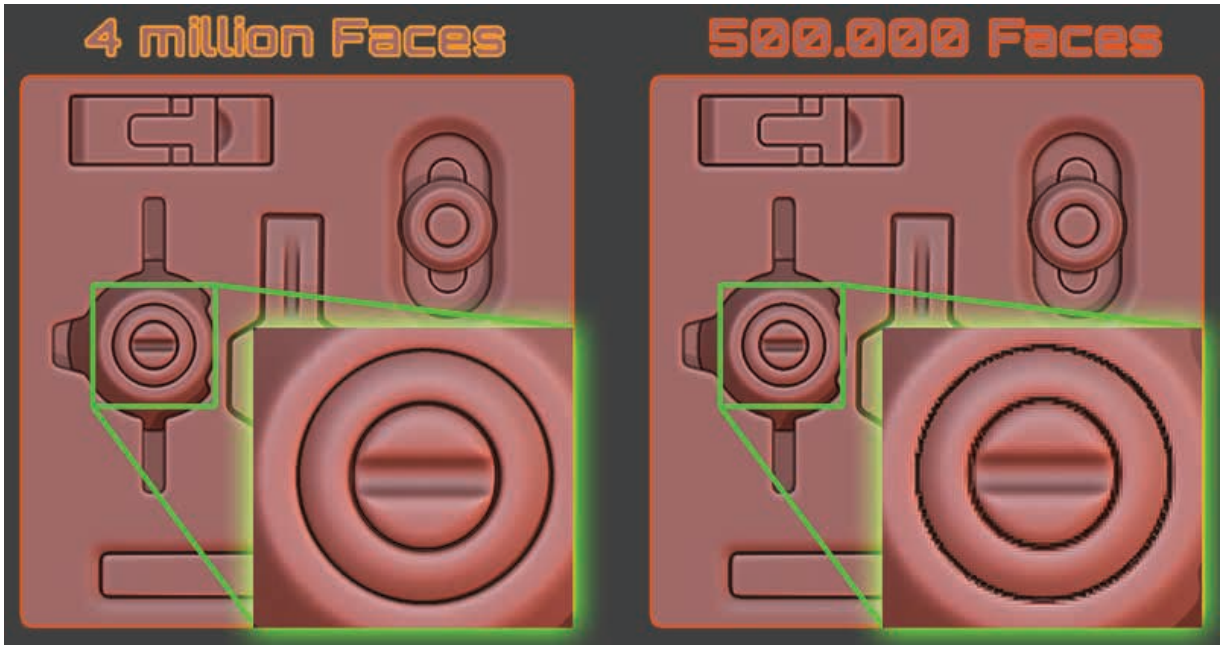


Then you can select your object and select a material to use. Procedural Rusty Paint Shader by CgAlpha on Blendswap.

Perfect Displacement maps with low poly models

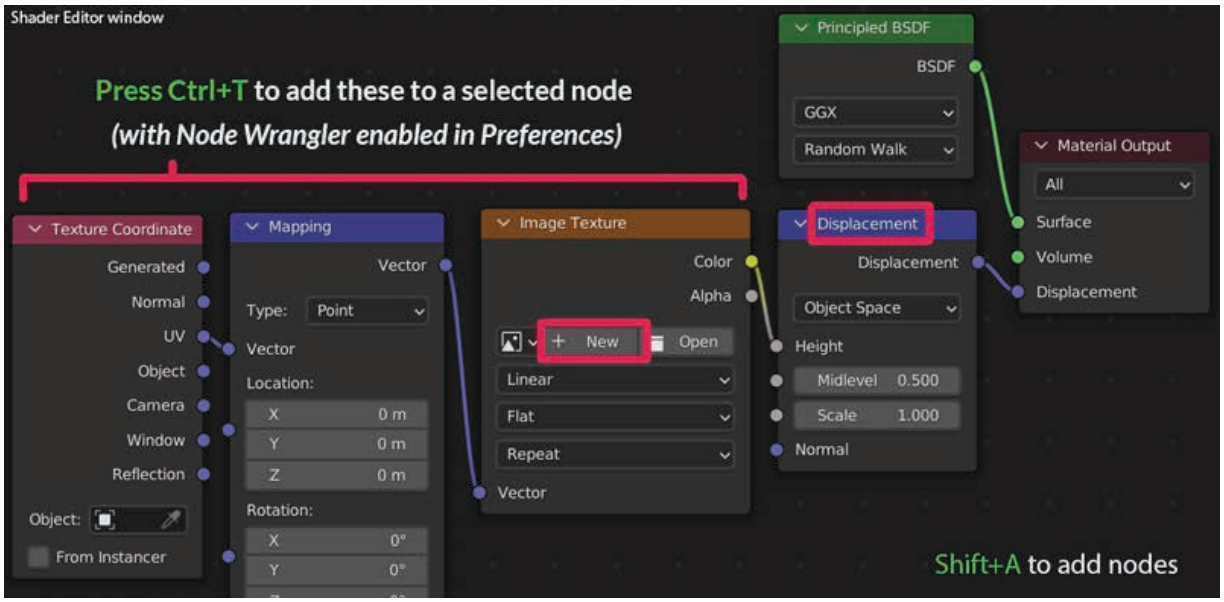
Previously we've looked at using Alpha textures to sculpt displacement and create a sword, as well as using them for Hard Surface detailing and how to create your own Alpha textures.

This is a fun way to work, but the downside is that you need a lot of geometry for it to look good. You need millions of Faces, or you'll see artefacts. You can't see it from a distance, but you can definitely see it when you get closer. On the left we have good detail, but at the cost of 4 million Faces. On the right there's "only" 500.000 Faces and we see some artefacts. 500.000 Faces is still quite a lot, especially for such a small area.

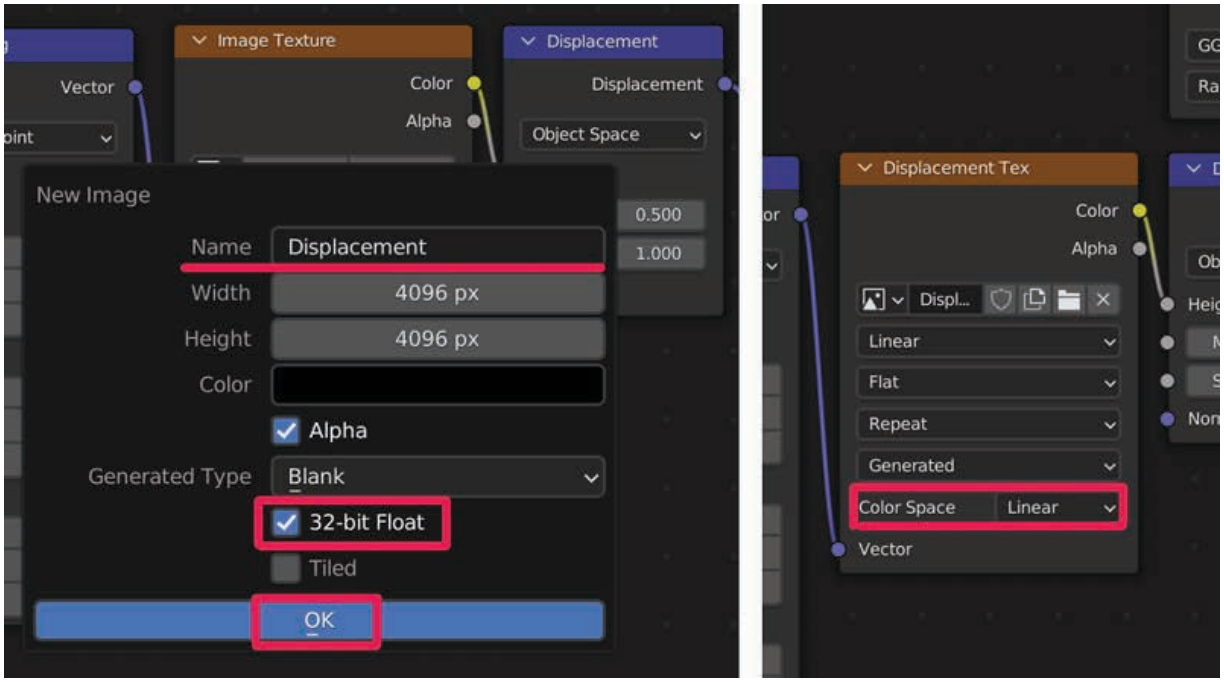


But there is another method that gives you a similar, or in fact even better, result with much less geometry. In fact for this to work well, it's best that we have a model with relatively low geometry. Because we're going to use texture painting, which works faster with lower poly models. How about *4 Faces instead of 4 million*? With better quality displacement?

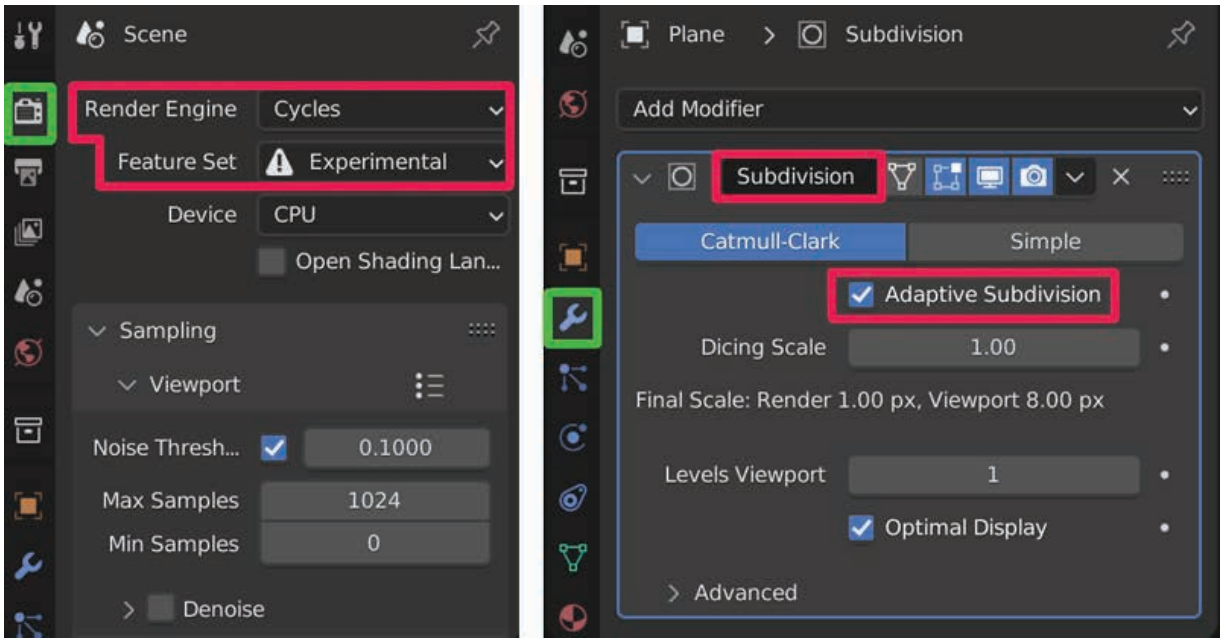
First, add a material to your mesh and give it a **Displacement node**. Then with the Displacement node selected, **press Ctrl+T** to add the necessary nodes for a texture *(provided you have Node Wrangler enabled in Preferences)*.



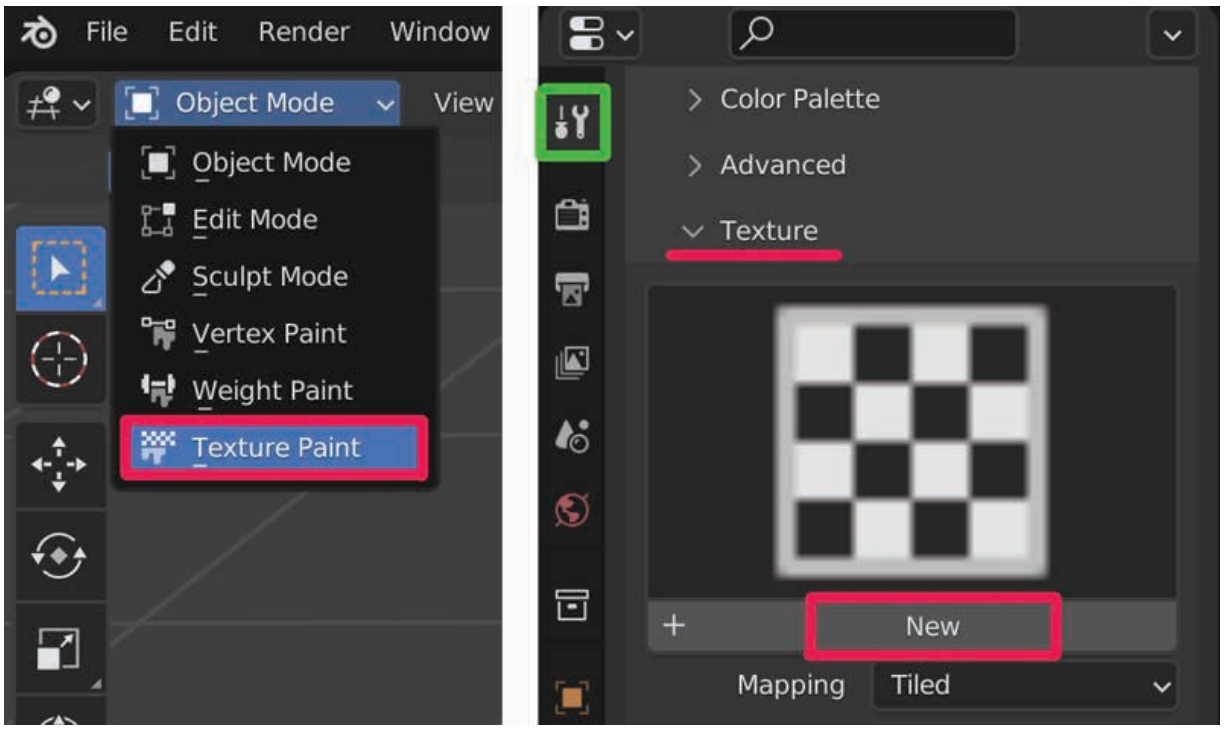
Now, **click New** on the image texture node, and create a 32-bit float Displacement texture. Set it to Linear. If the Principled BSDF is in the way, you can select it and press Ctrl+H to minimise it.



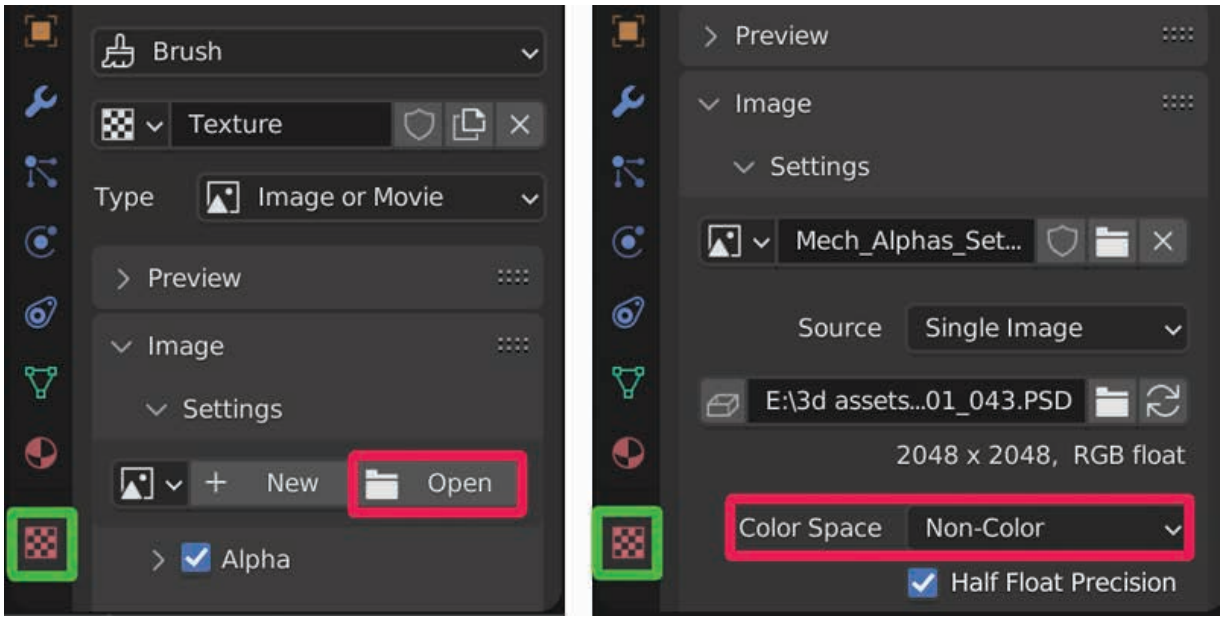
Make sure the render engine is set to **Cycles**. enable **Experimental features** in Cycles and add a **Subdiv modifier** to the mesh, with **Adaptive Subdivision** turned on. The Plane I'm using in the examples below has about 64 Faces. It's been subdivided in Edit Mode 3 times.



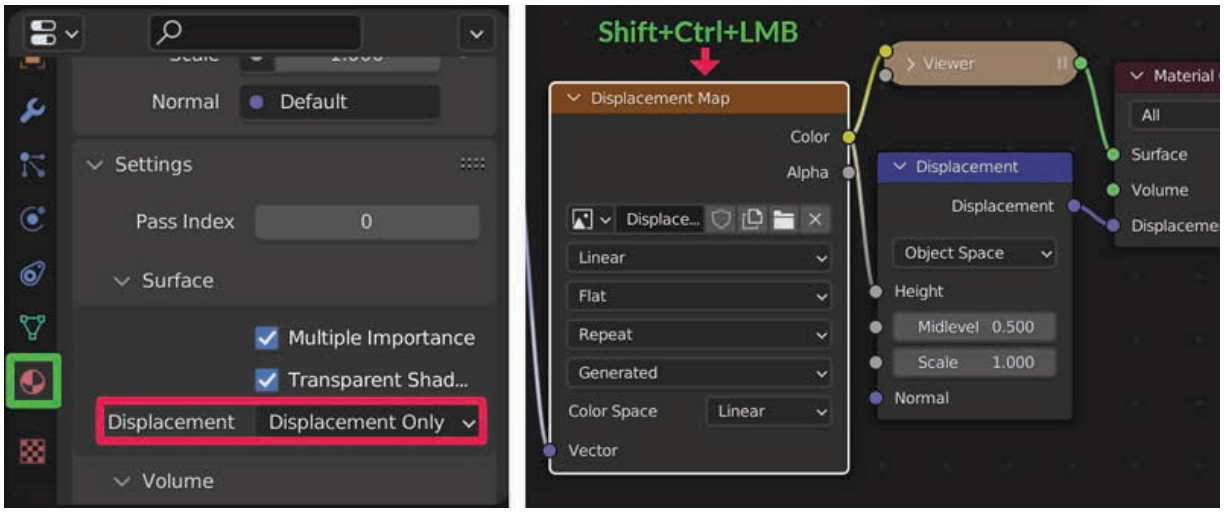
You can use the same Alpha textures that you would use in the Sculpting workflow. If you need some free ones to experiment with, I recommend these by JRO Tools. In my case, I'm using some that I purchased, made by Paul Dave Malla. Switch to **Material Preview** in the viewport. In **Texture Paint mode**, add one of the Alpha textures you downloaded or created to your brush by going to **Brush Settings > Texture > New**.



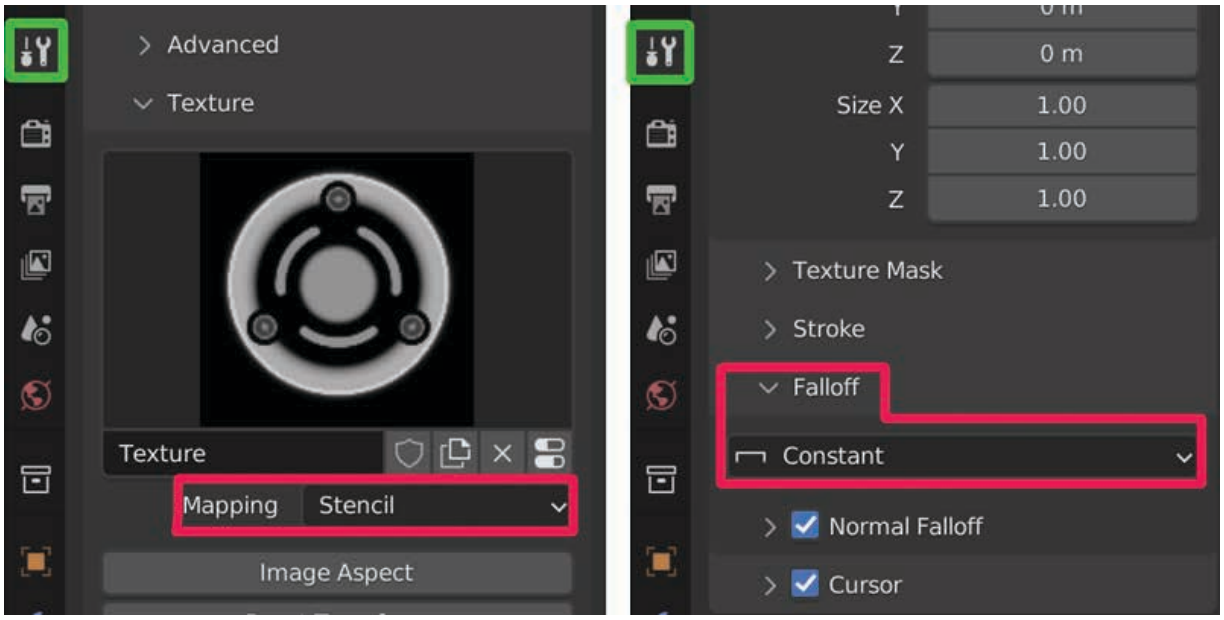
Set the Alpha texture to **Linear** or **Non-Color** instead of sRGB, otherwise you won't get all the detail out of them.



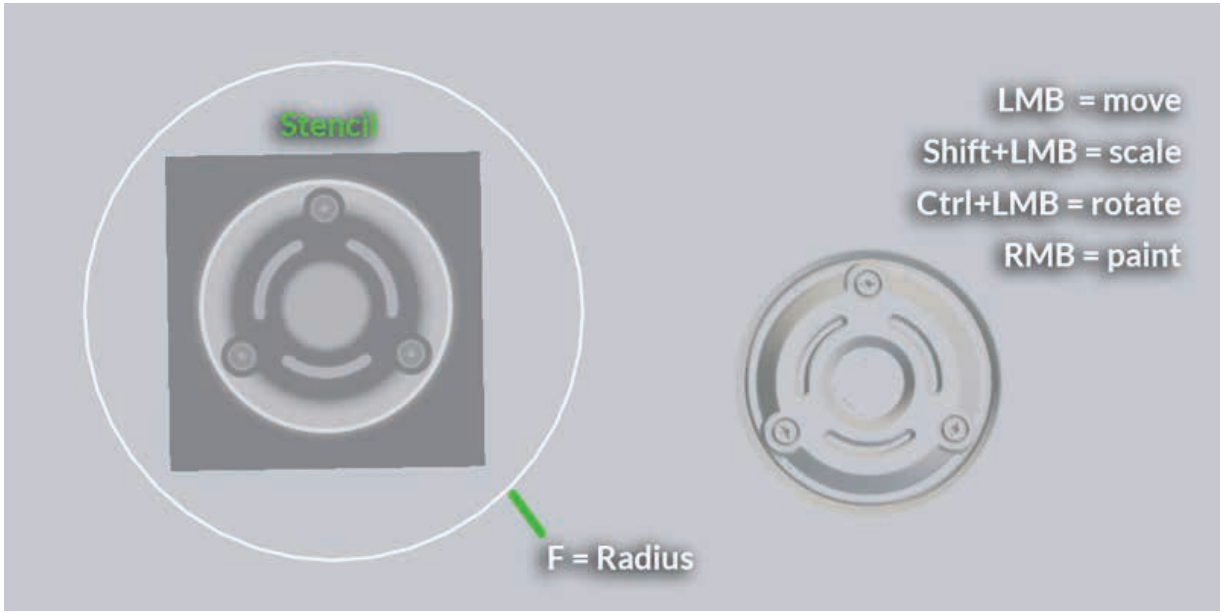
Make sure Displacement is set to **Displacement Only** in the material's Surface > Displacement setting. Otherwise the displacement won't be visible. **Shift+Ctrl+click** on the image texture node for the displacement texture to preview it on the model in Material preview mode. A **Viewer node** is temporarily added when you do this.



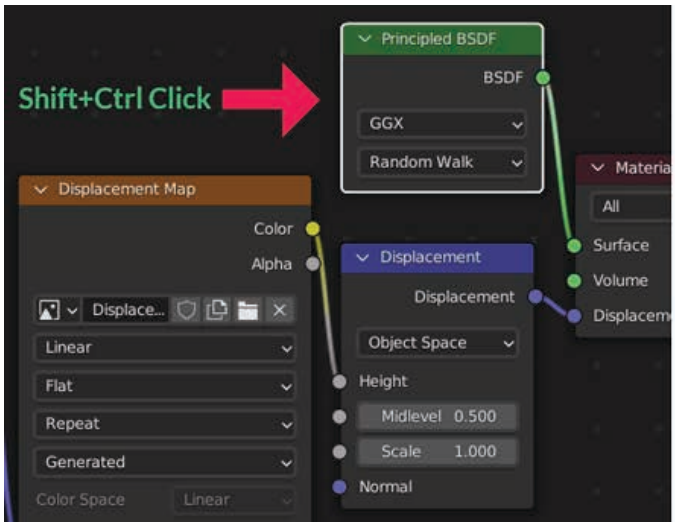
In **Texture Paint mode** you can now paint the alpha on the model using **Stencil** mapping. I recommend setting **Falloff** to **Constant** so the texture is used exactly as intended.



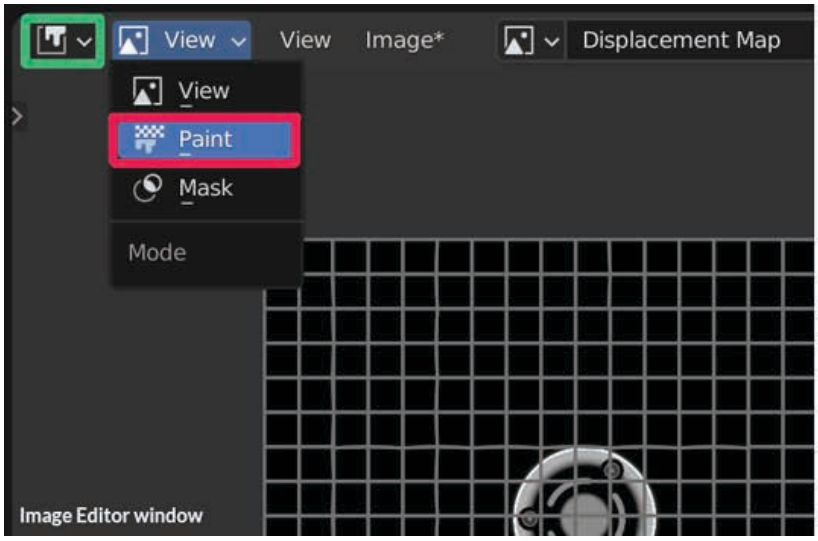
To use the Stencil mapping, move the stencil by **holding the RMB** and moving the mouse. **Shift+RMB** scales it and **Ctrl+RMB** rotates it. **Left-clicking** projects the stencil from view. Make sure the brush Radius is big enough so the edges of the stencil don't get cut off. You can change the Radius by **pressing F**.

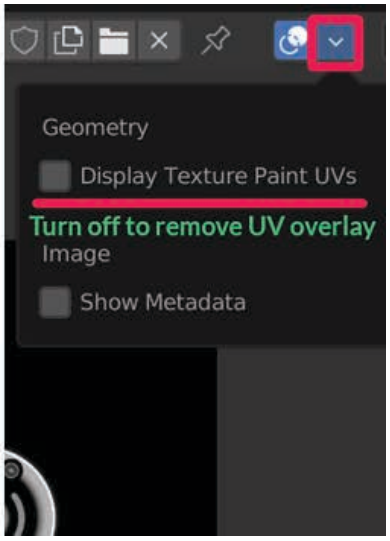


Shift+Ctrl+click on the Principled BSDF to stop previewing the Displacement texture. In Material preview mode it will look like you used a Normal map. But don't worry, in rendered view it will be real displacement.

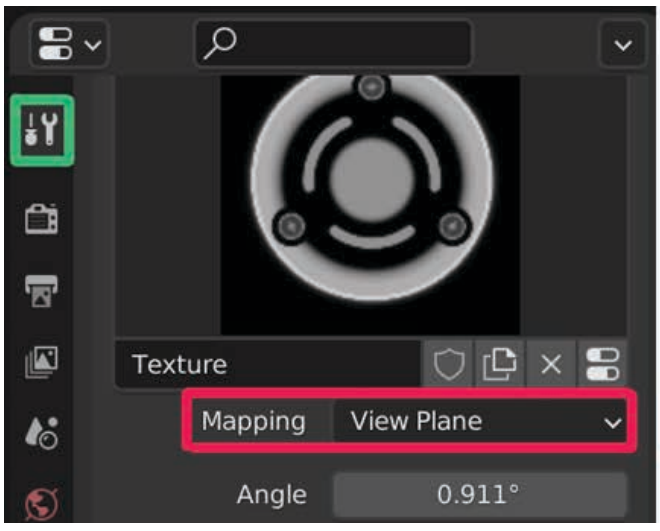


Instead of painting displacement directly on the model, you can also paint directly onto the textures in the **Image Editor**. This can be a good choice if you have a decent UV map. Set the Image Editor to **Paint**. In Image Editor overlays, **disable Display Texture Paint UVs** if you don't want to get distracted by the UV's.





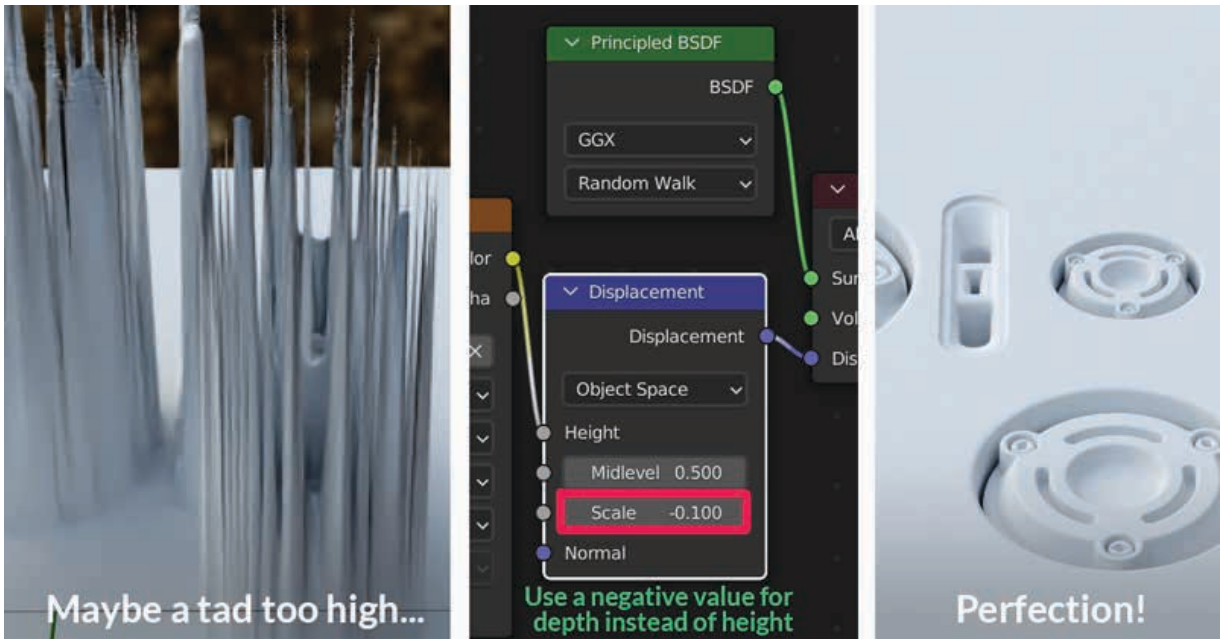
Then with the texture mapping of the brush set to **View Plane** instead of Stencil, you can add the texture directly to the UV's.



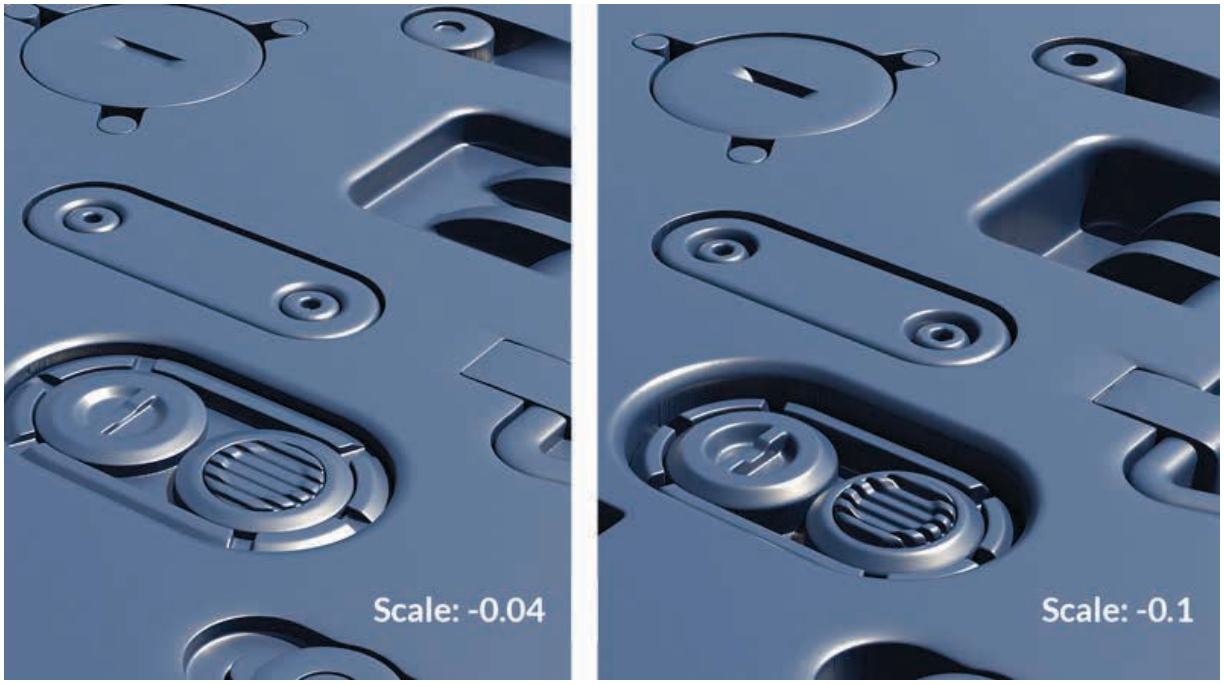
Set the Brush Blend mode to **Add**, so that you can draw close to other Alpha textures already on the texture, without erasing them with the black part of your brush.

Now you'll see the displacement in Cycles **Rendered view**. Amazingly, the model only uses a small amount of Faces. However, because of the **Adaptive Subdivision**, it looks better than if we'd used it on a very high resolution sculpted mesh with millions of Faces. Keep in mind however, that you'll need to render the Camera view to see all of the detail as the Adaptive Subdivision by default keeps the viewport detail level quite low. So don't worry if it looks blurry in the viewport, it'll look great in the render!

Also, you may want to lower the displacement **Scale** based on what you see in the viewport as by default Blender uses a rather high Scale, and use a negative value to create depth instead of height.



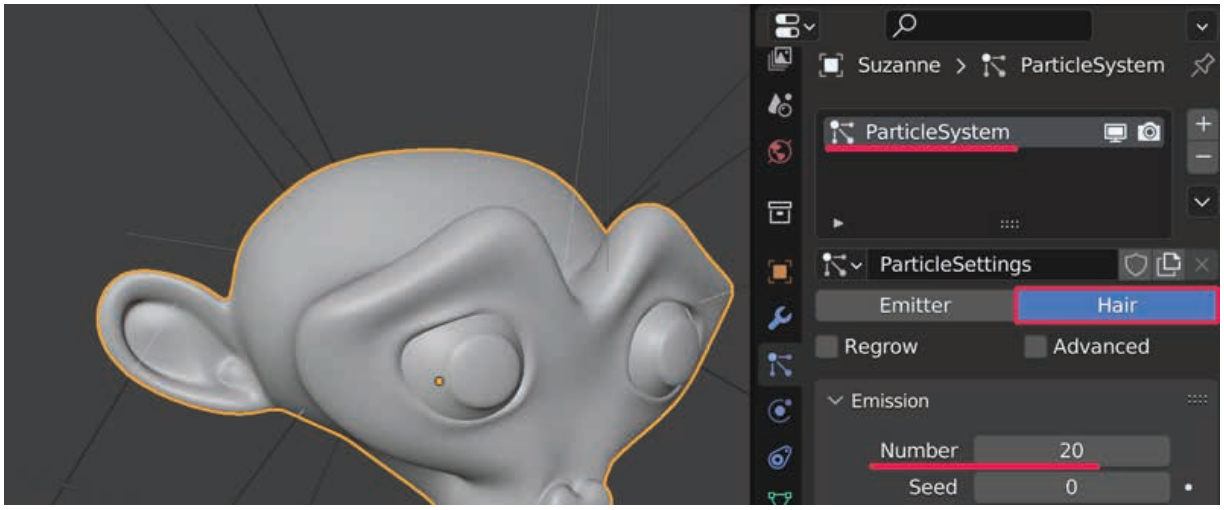
Another benefit of this method (instead of the sculpting one) is that you can still adjust the overall depth or height of the displacement effect by simply adjusting the **Scale** in the **Displacement node**. Plus you can disable the Subdivision modifier until Render time, which makes it less taxing on your computer.



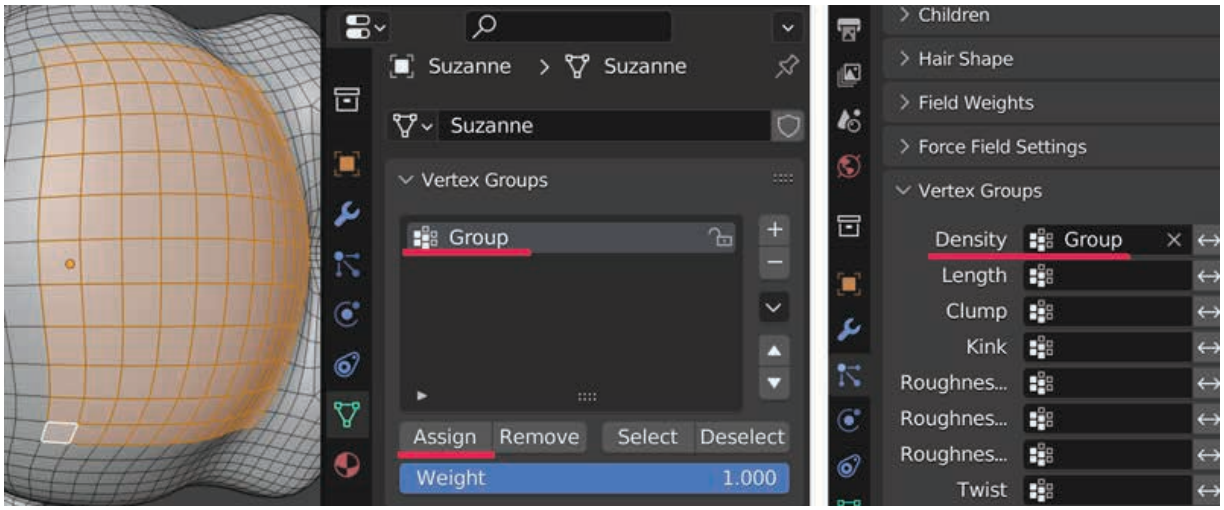
Tip: if you paint the displacement but you don't see it in Rendered view, switch back to **Material view** and then back to **Rendered view**. This updates the viewport and you'll normally see the displacement appear now. If you still don't see it, make sure Displacement is set to "Displacement only" in the material Settings.

Low-poly hair cards

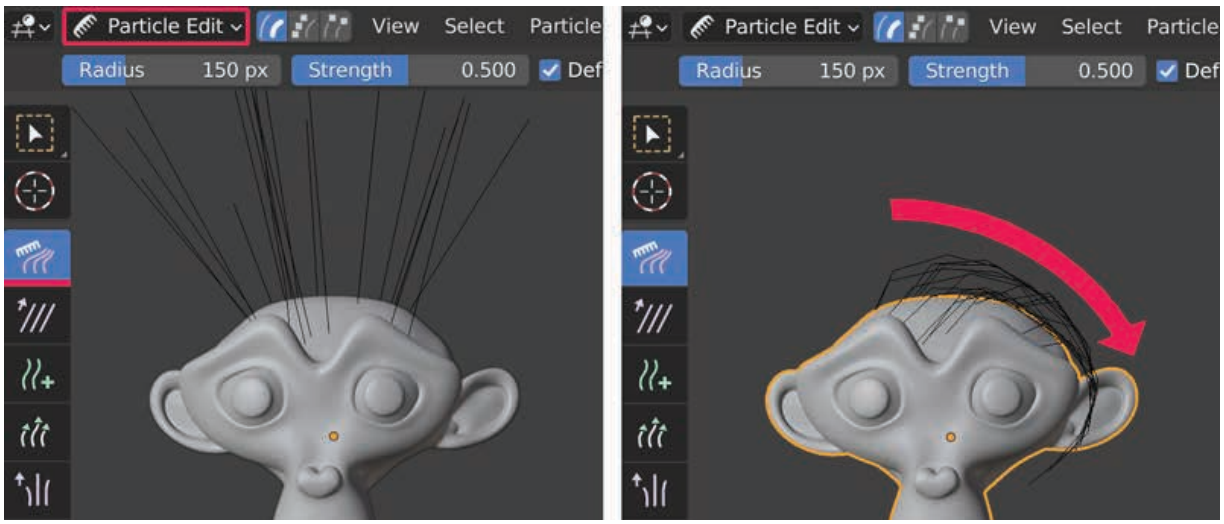
Hair particles can add a lot of render time to your project. This is where hair cards come in handy. They are simple low-poly geometry with textures on them. This is a quick guide on to use them. Add a Hair Particle system to your model. You don't need a lot of particles.



Use a Vertex Group in Density, so that it's only growing where you want it to.

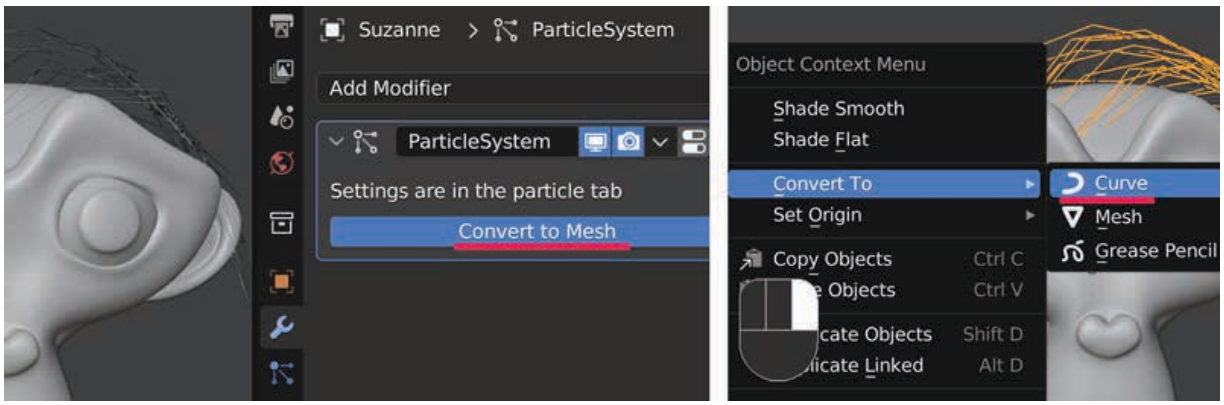


Groom the hair in Particle Edit mode.

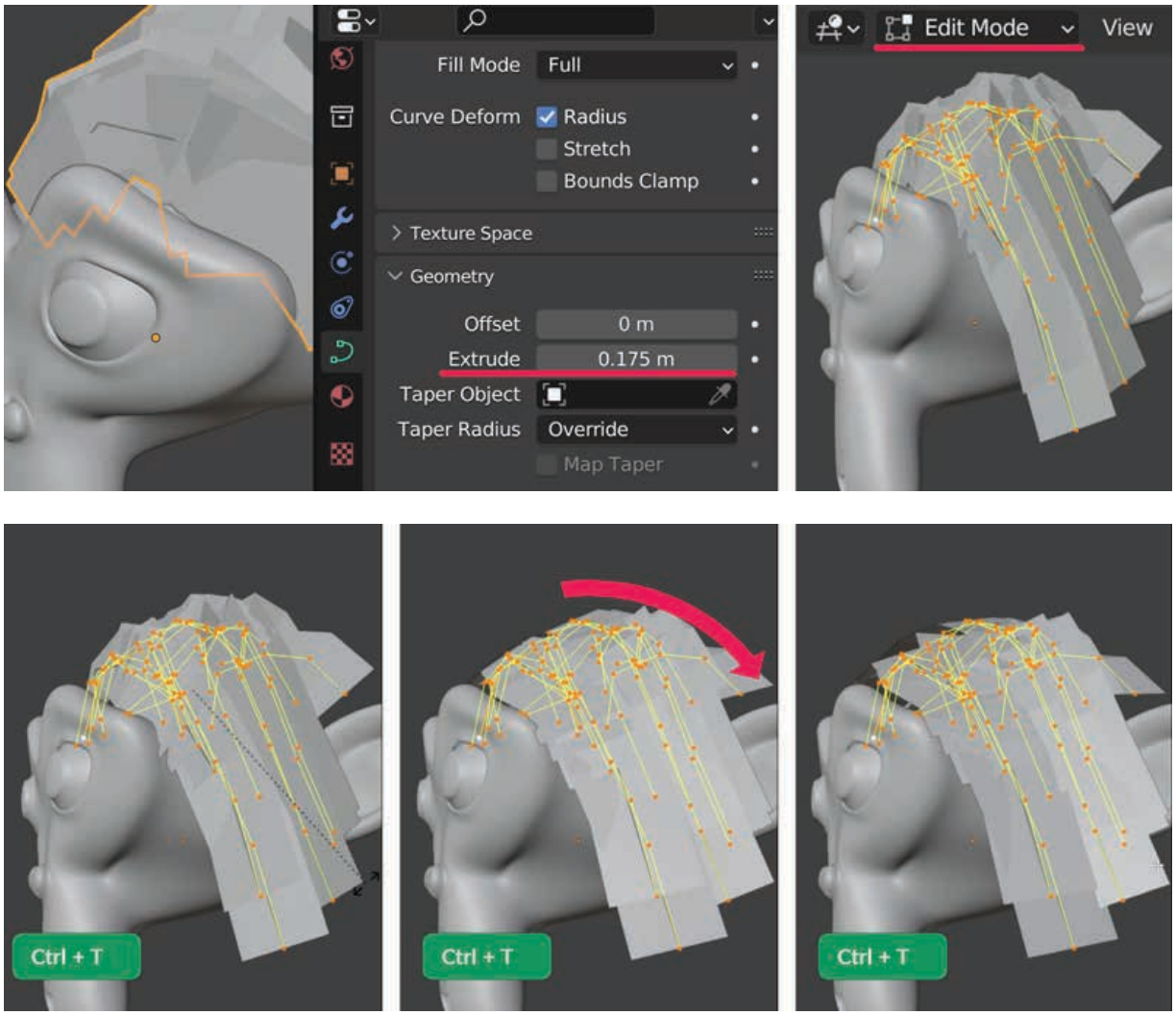


Click here to return to the index.

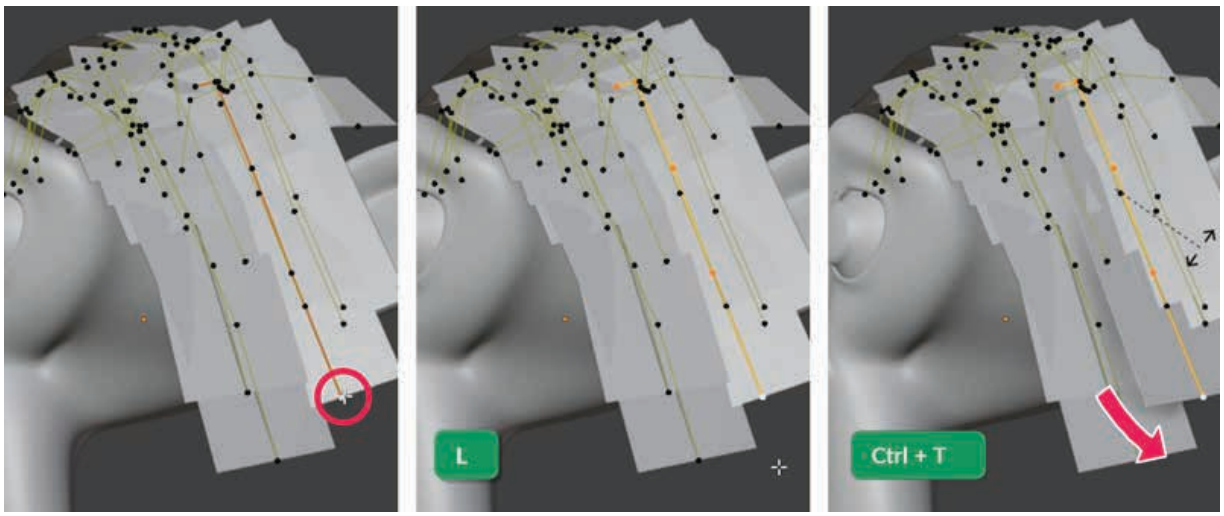
Once you're happy with the hairstyle, apply the Particle System by clicking on **Convert to Mesh**. Then convert the hair to **Curves**.



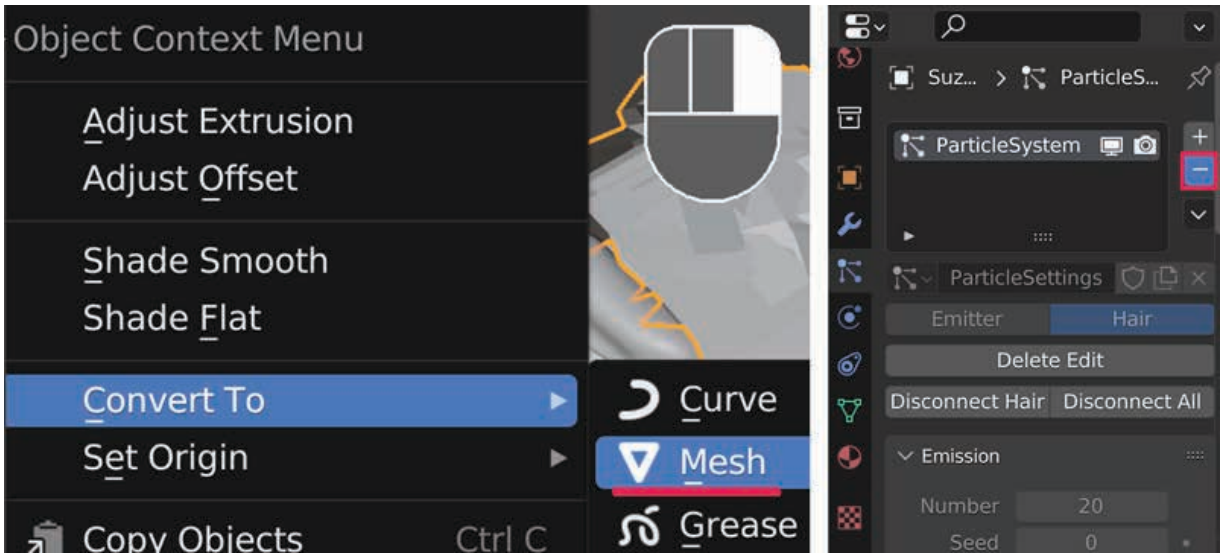
Extrude the hair to create some flat geometry. In Edit Mode, select them all and **press Ctrl+T** to twist them.



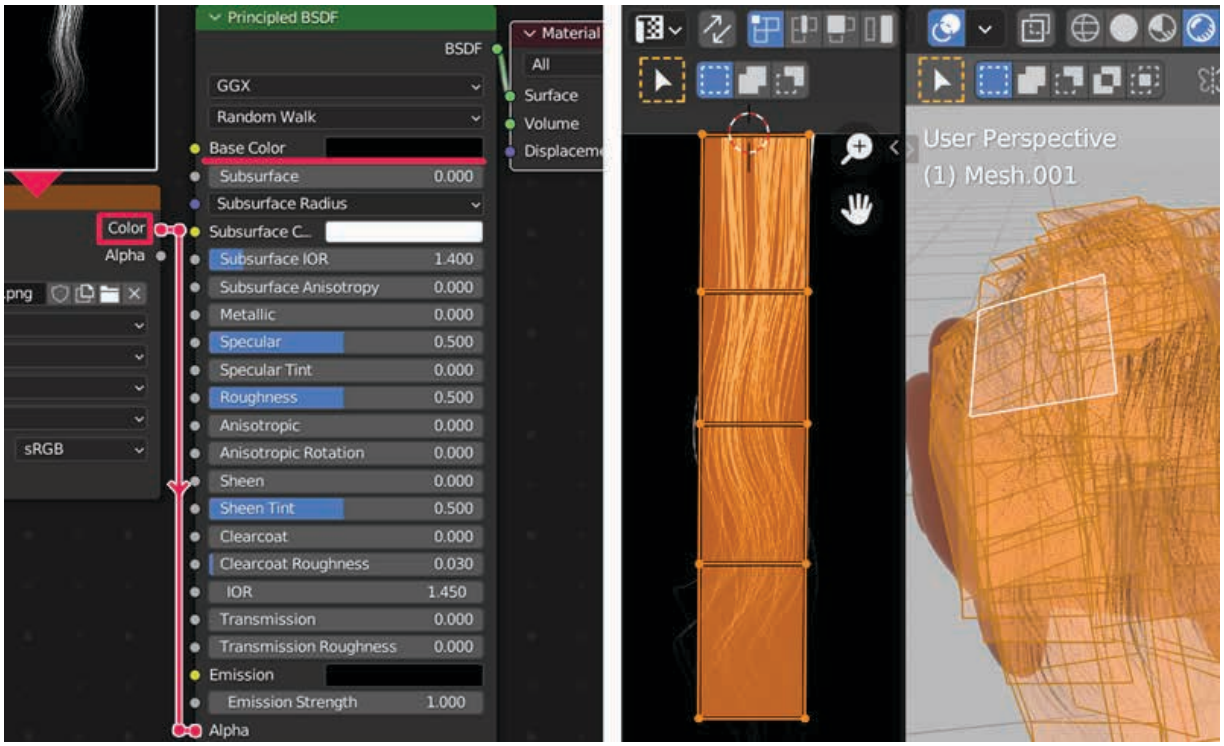
You may have to select individual strands and twist them to get them facing at a good angle.



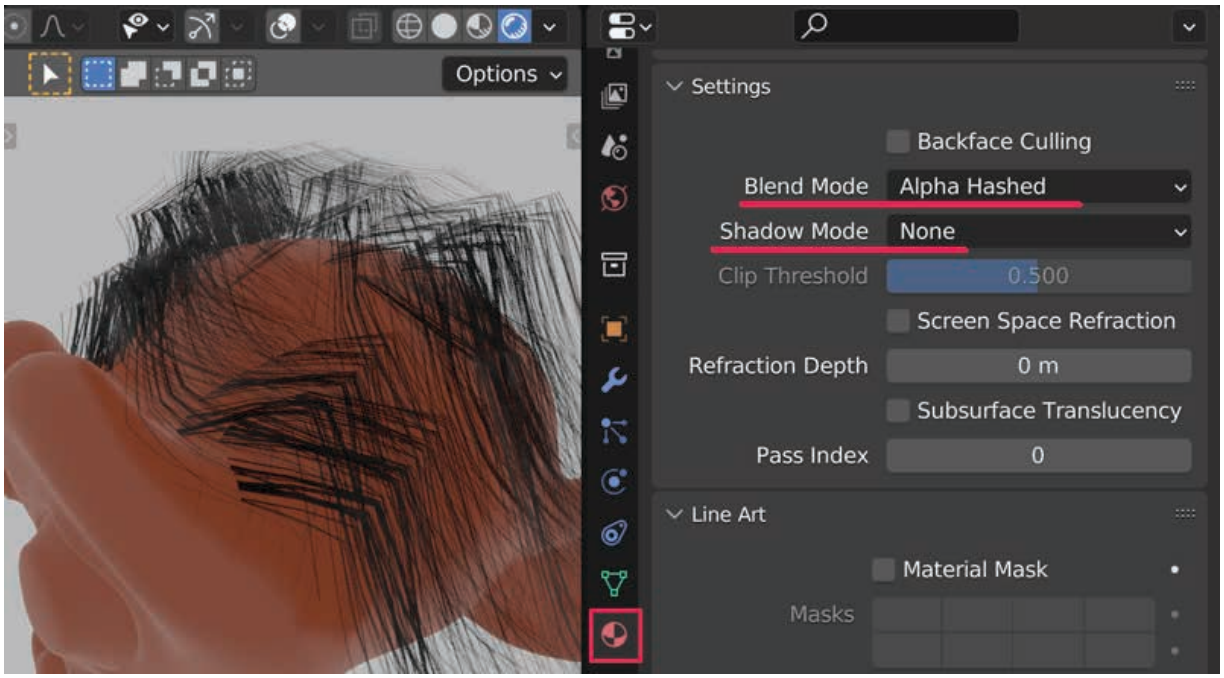
You can now convert the curves to mesh and remove the particle system.



For the hair texture, I'm using this public domain image I found on OpenGameArt.com. Create a new material, with the diffuse color that you want the hair to have and use the hair texture as the Alpha. In Edit Mode select all the faces and scale the Uv's in the UV editor to fit the texture.



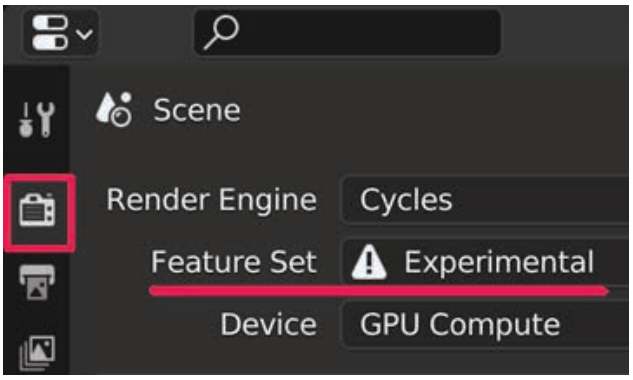
For Evee, set the Blend Mode to **Alpha Hashed** and Shadow Mode to **None.**



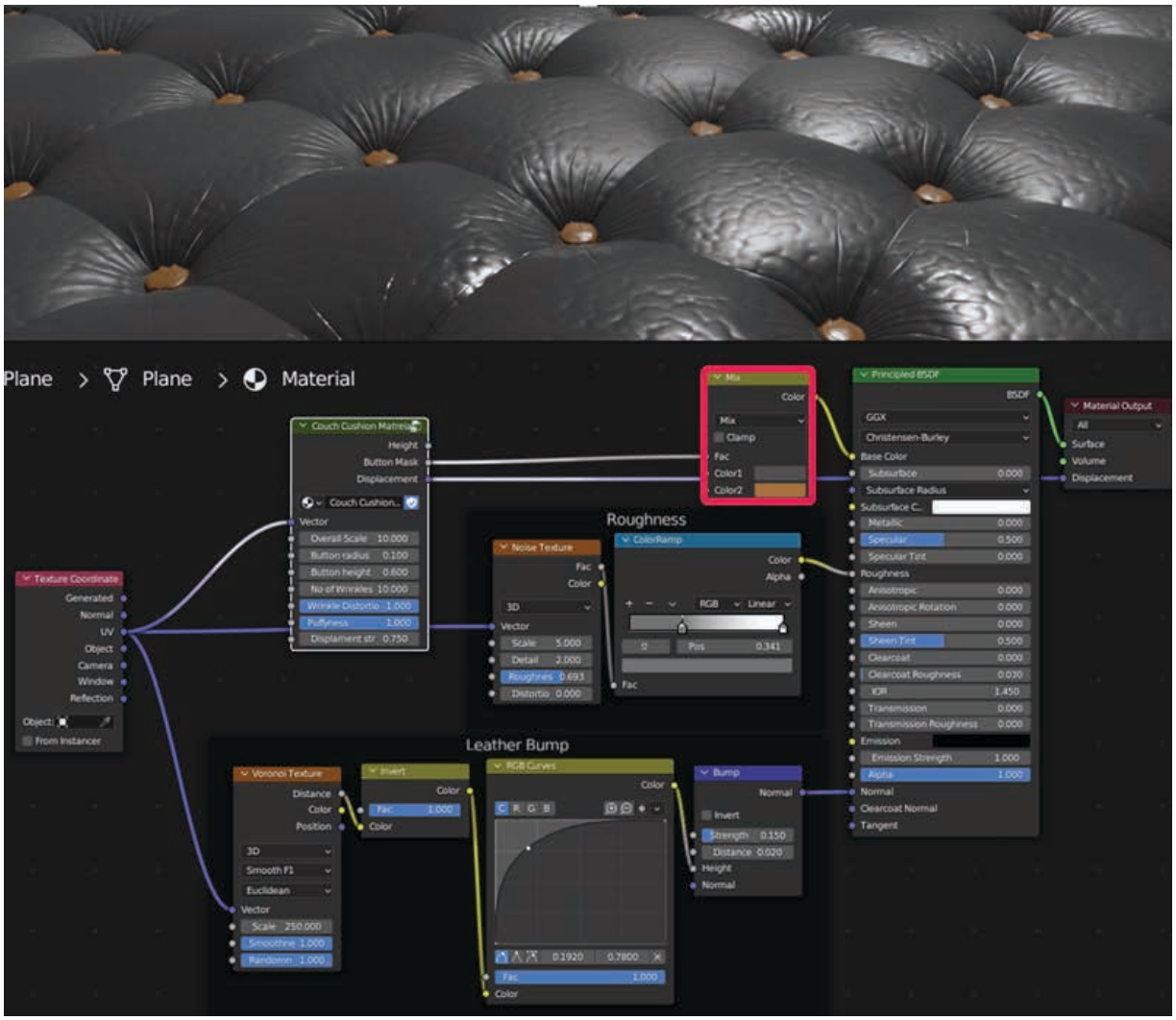
Procedural Leather Material

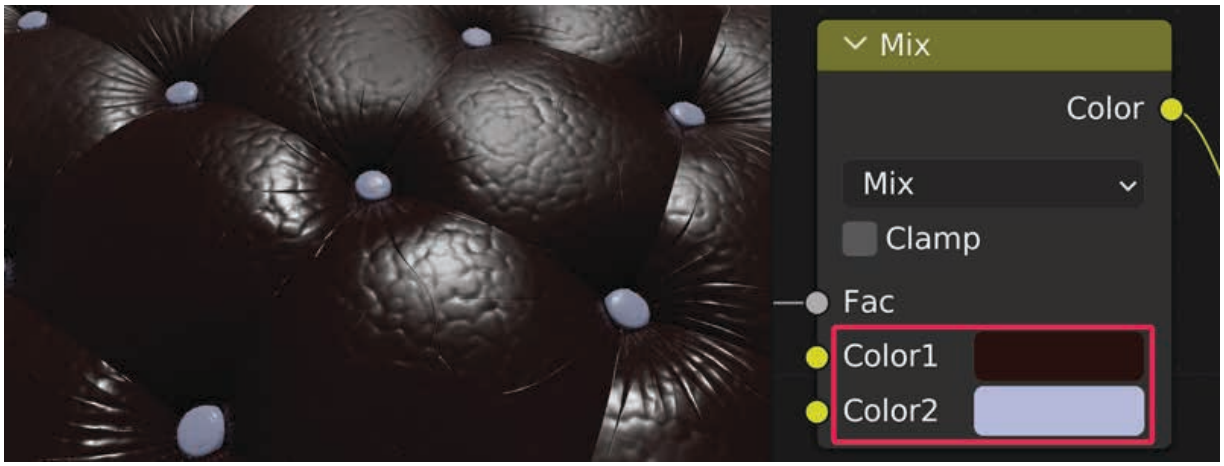
Download the Procedural Leather Couch Material from Gumroad. It's free!

To use this material, you'll need to use the Experimental Feature Set of Cycles.

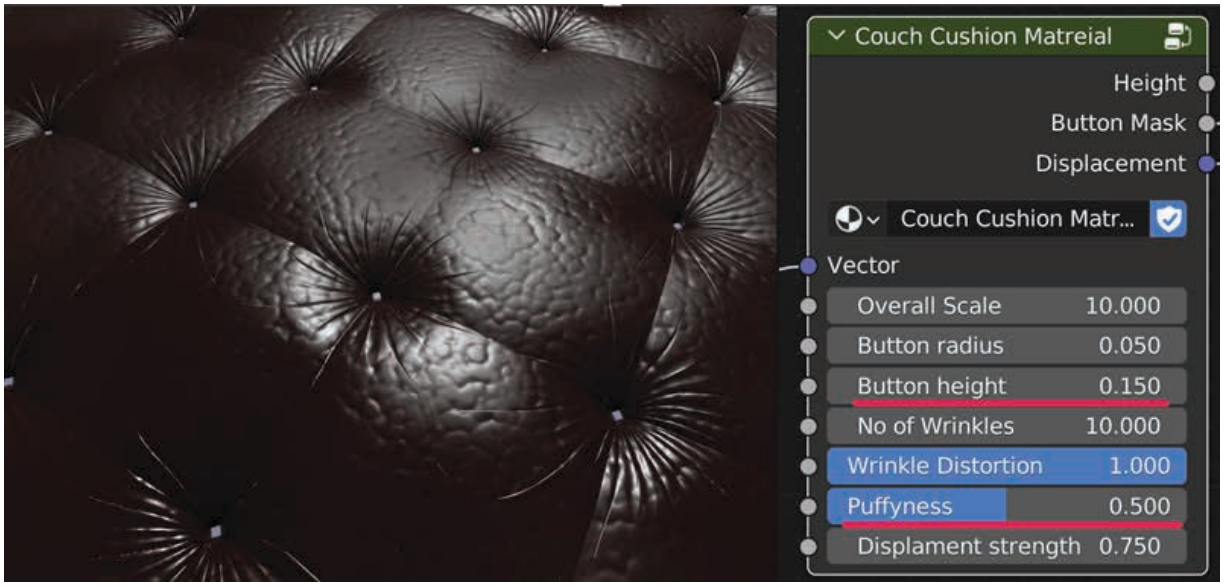


This **Mix RGB node** lets you choose different colors for the leather and the buttons.

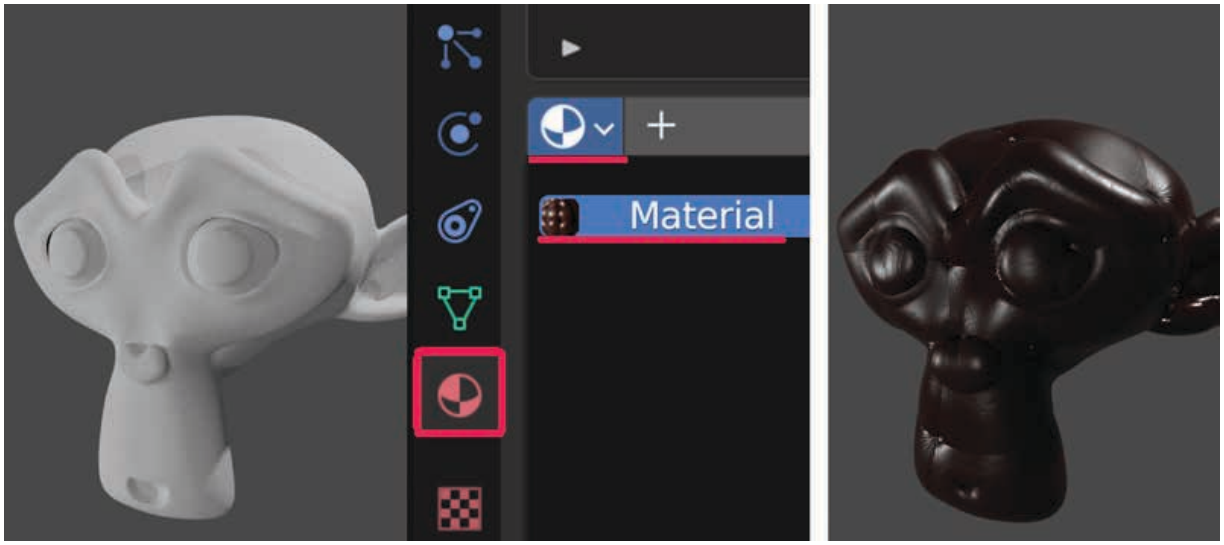




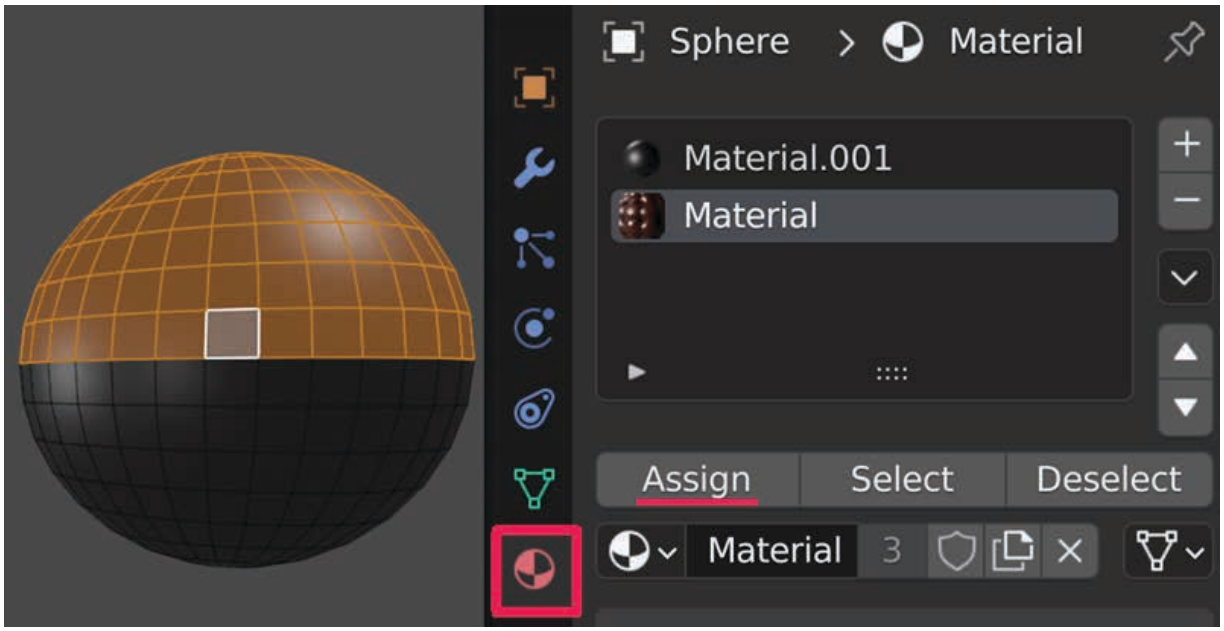
If you want to learn how it was made, you can open this Node Group. However you can control the most important values from the Collapsed Node Group. For example the Puffiness or the Button Height.



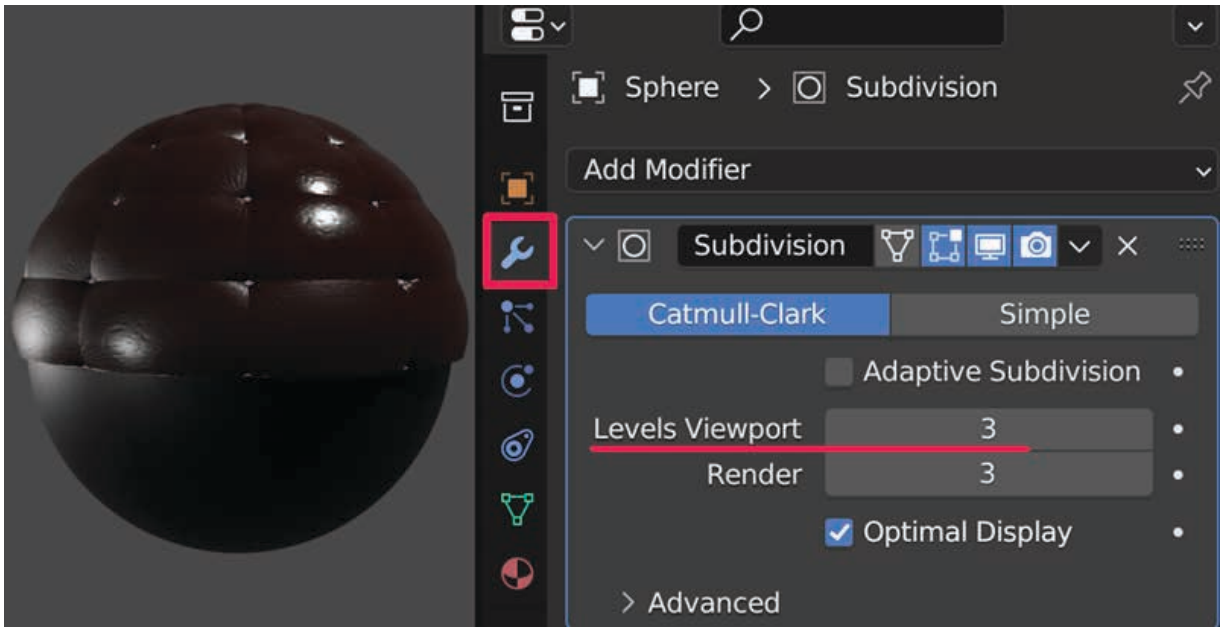
Applying the material to other objects is easy, just select it from the Material drop-down.



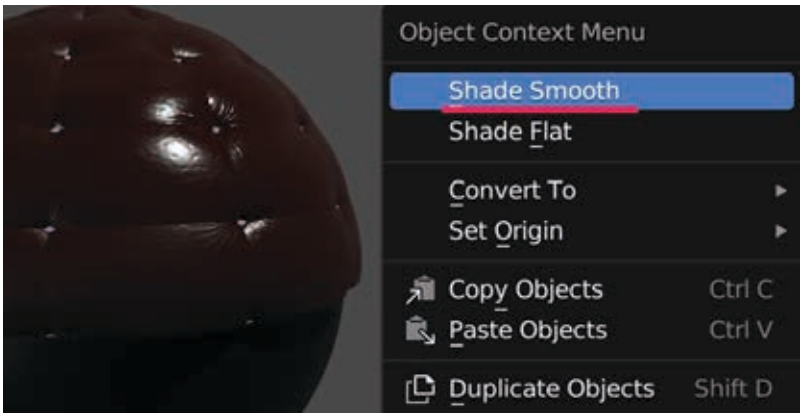
You can also apply it to specific faces of the model by selecting those and clicking Assign.



Keep in mind that you'll need enough geometry for the displacement to work.



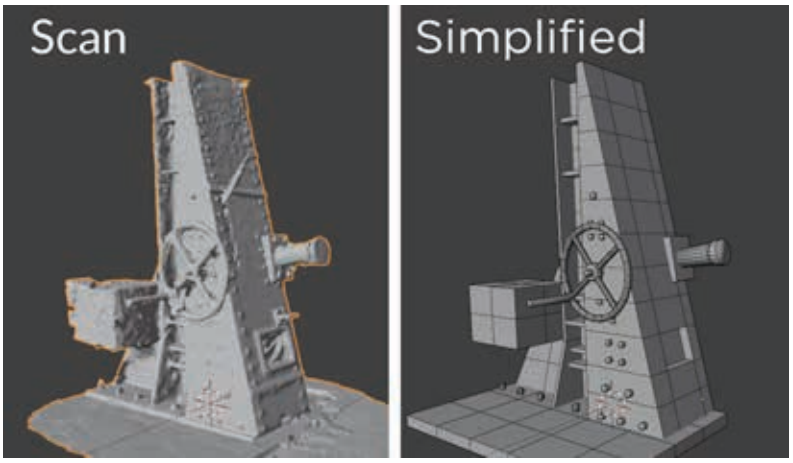
It also looks best if you use Smooth Shading.



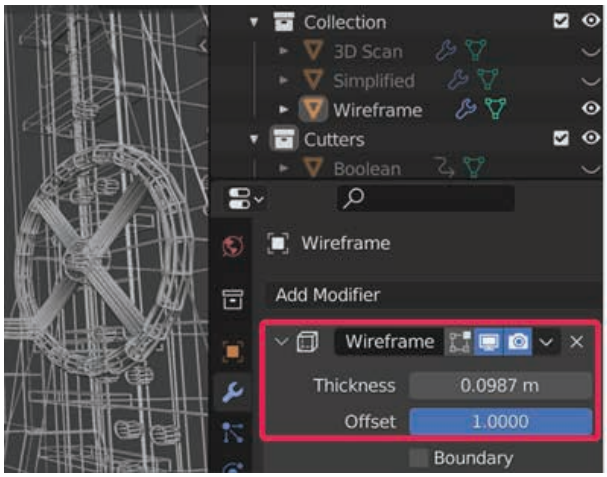
Wireframe Effect using modifiers

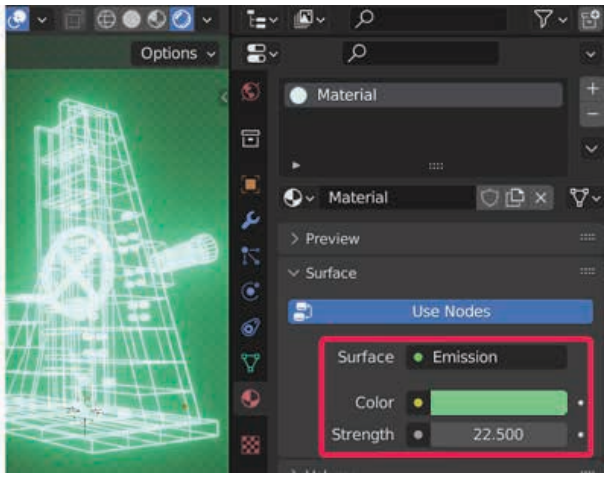
Here's a cool effect - using booleans, you can show different materials and models in slices. This topic explains how to set it up.

You can download the Blend file here. First, model a simplified version of your 3D scan.

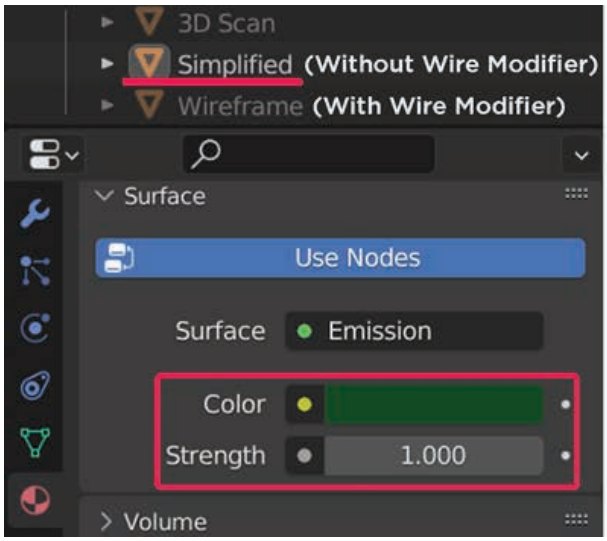


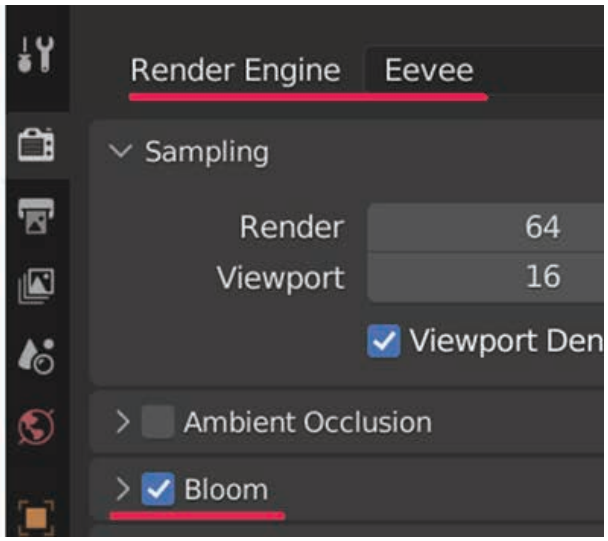
Add a **Wireframe modifier** and an **Emission** shader. Increase the Wireframe **Thickness** and **Offset** to make the wireframe more visible. Increase the Emission **Strength**.



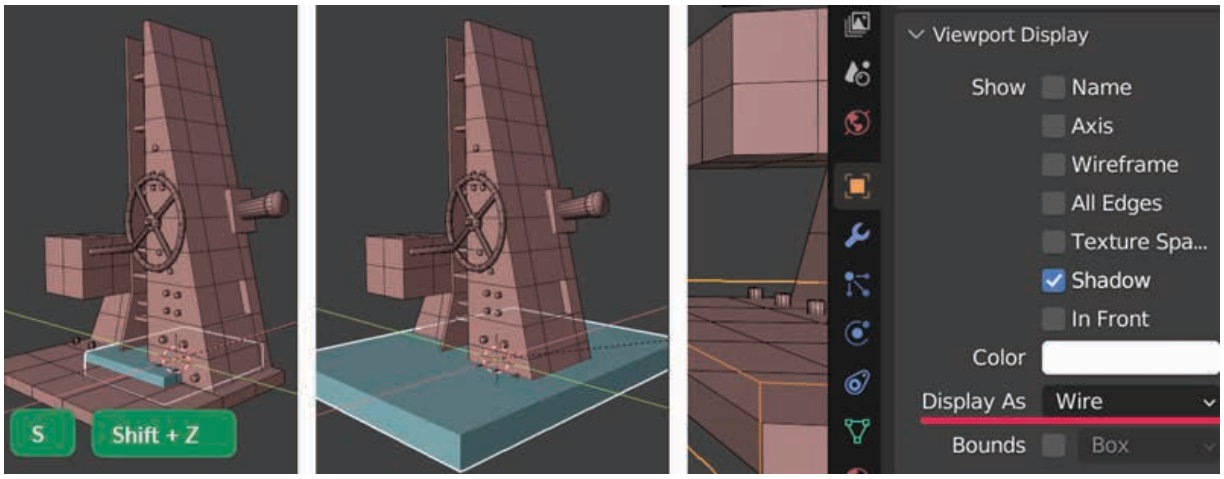


Duplicate the simplified model and remove the wireframe modifier. Give it an emission material as well, but make this one darker. In Eevee, turn on Bloom.

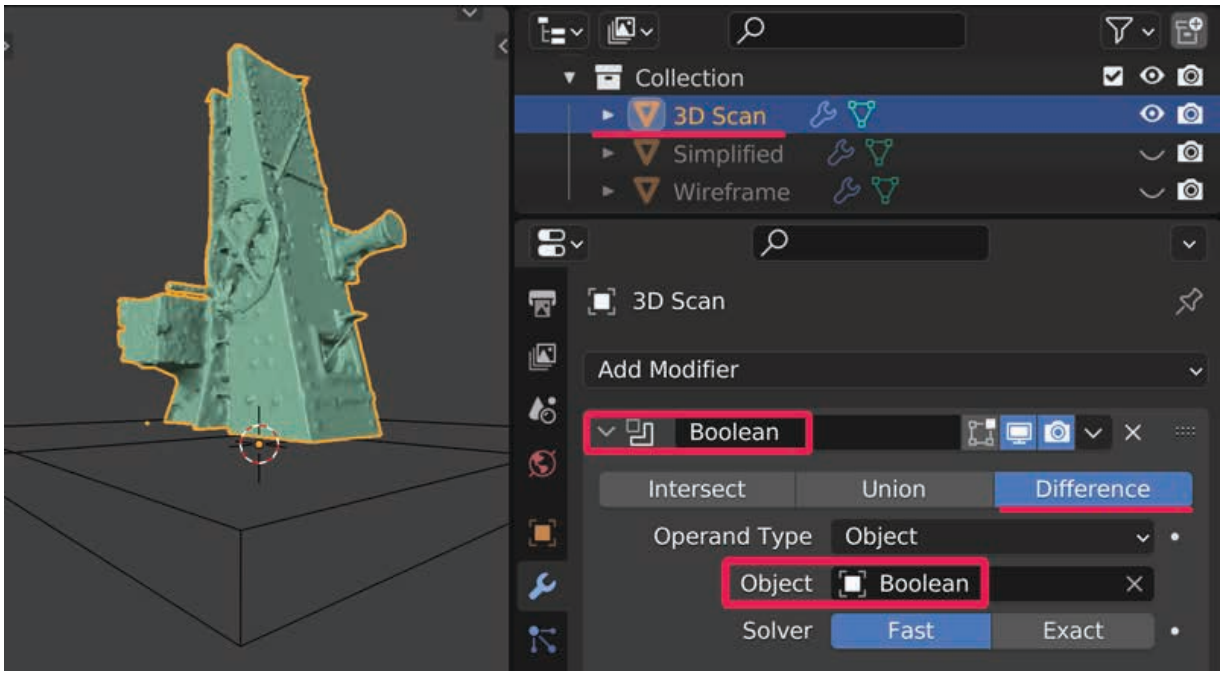




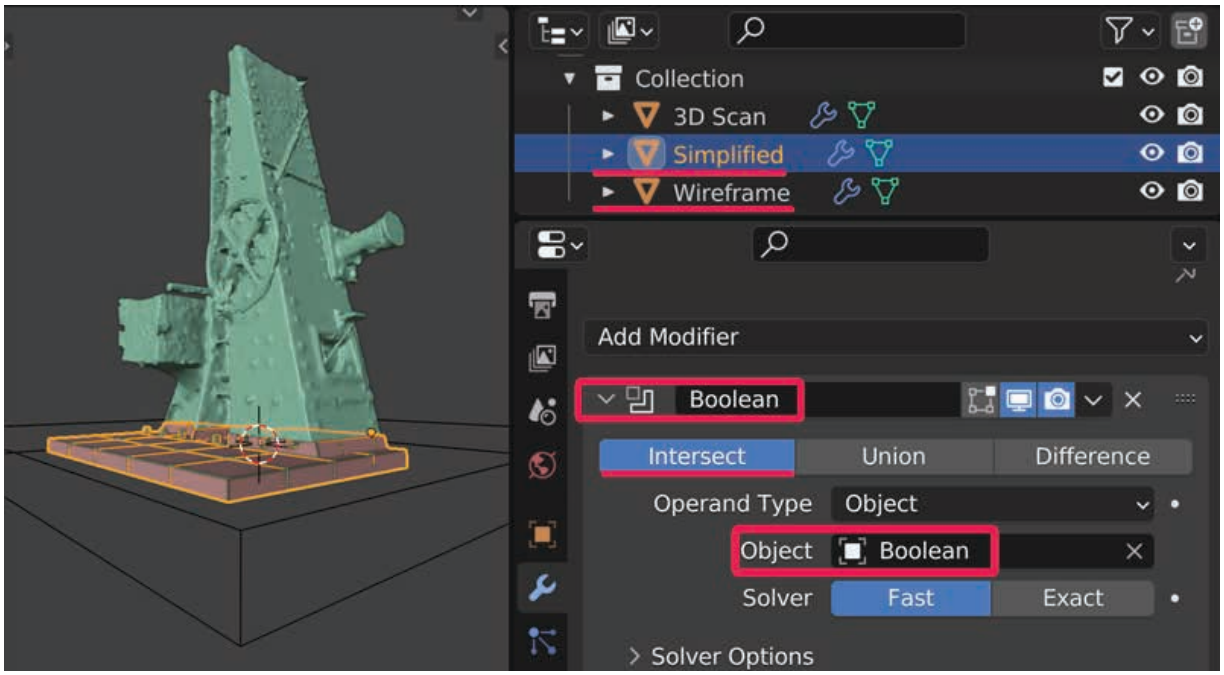
Add a cube and scale it in Edit Mode to be wider than the 3D scan. Name it “Boolean” to avoid confusion and set it to Display As **Wire**.



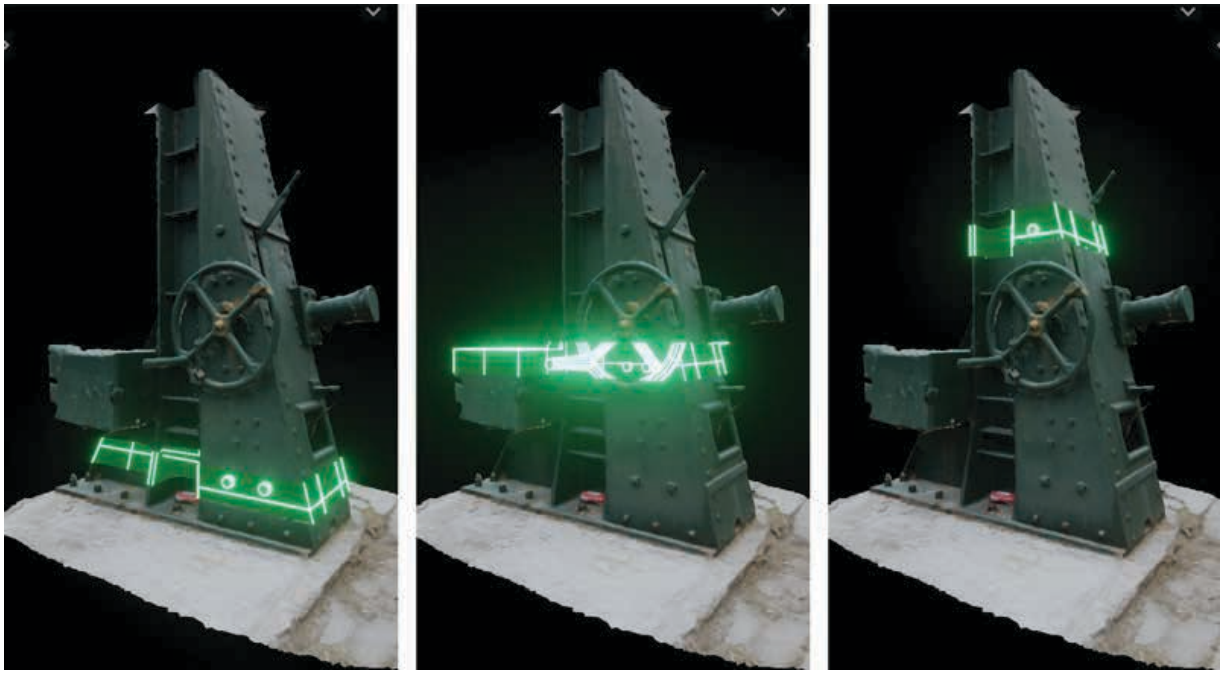
Unhide the 3D scan in the Outliner and add a **Boolean modifier** to it. Set it to **Difference** and choose the Boolean **Object**.



Select the Simplified model and add a Boolean modifier to it as well, but set this one to Intersect. Do the same for the Wireframe object.

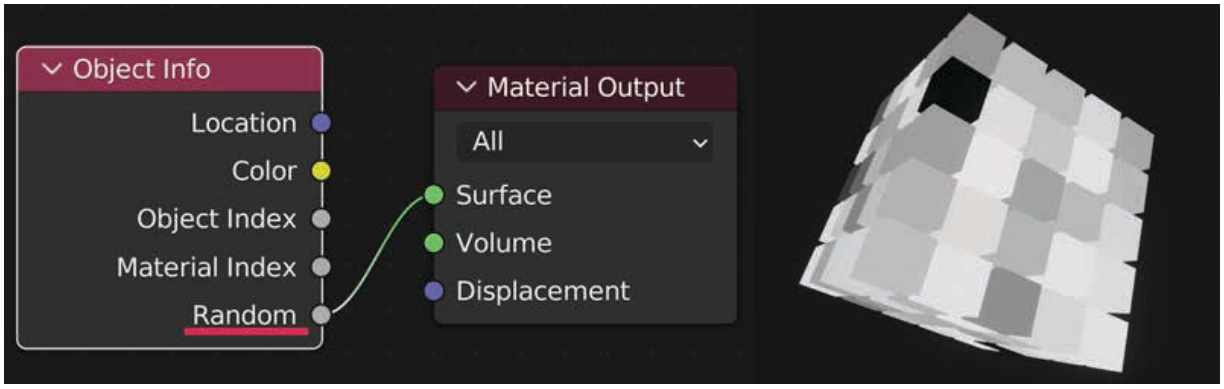


Now when you move the Boolean object it will reveal the wireframe beneath the 3D scan.

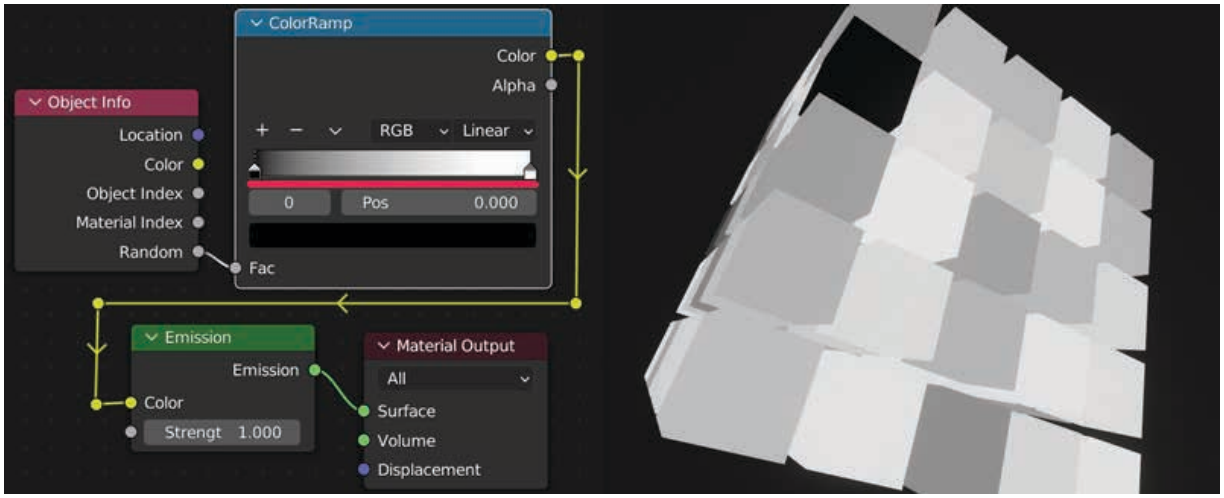


Random colors using nodes

The **Object Info node** has a **Random** output. It can create random colors for objects. This is interesting for motion graphics, for example. Download the Blend file here.



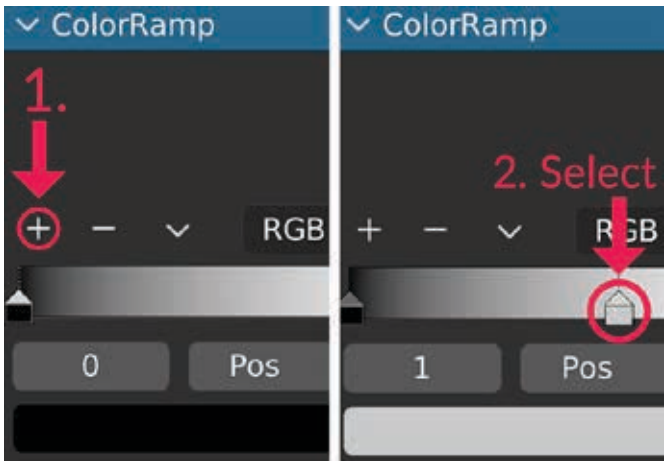
It can create random colors for objects. Objects with this material will have random colors different values of gray in this case as a **ColorRamp node** is added in between the **Object Info node** and the material. Because the ColorRamp is a default black to white gradient you only get grayscale values.



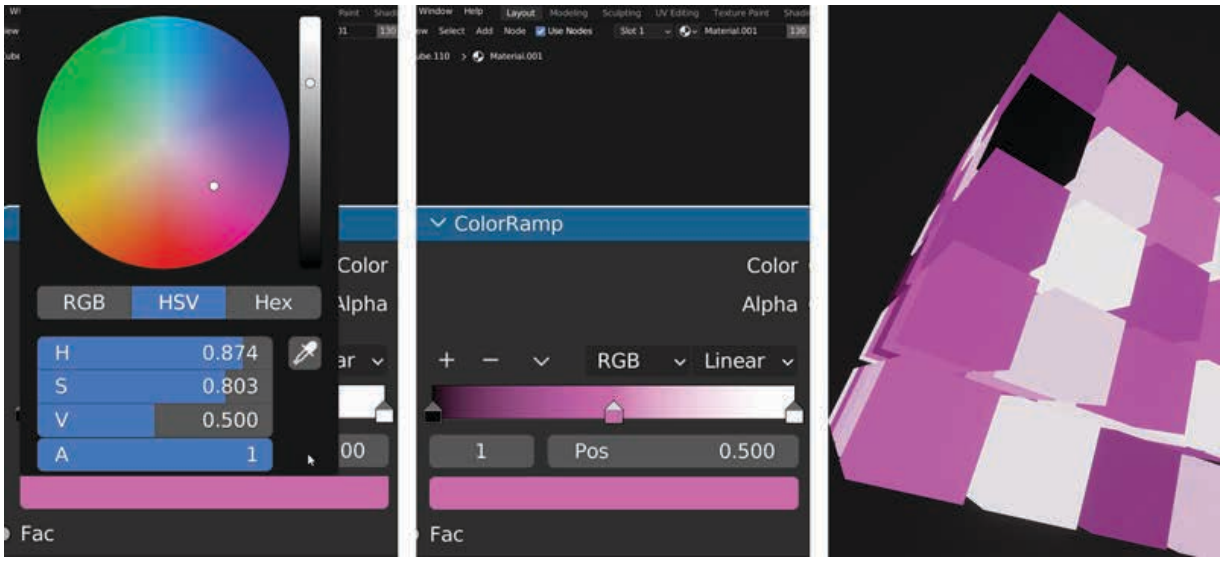
Click on the + icon.

A new color Stop is added to the ramp.

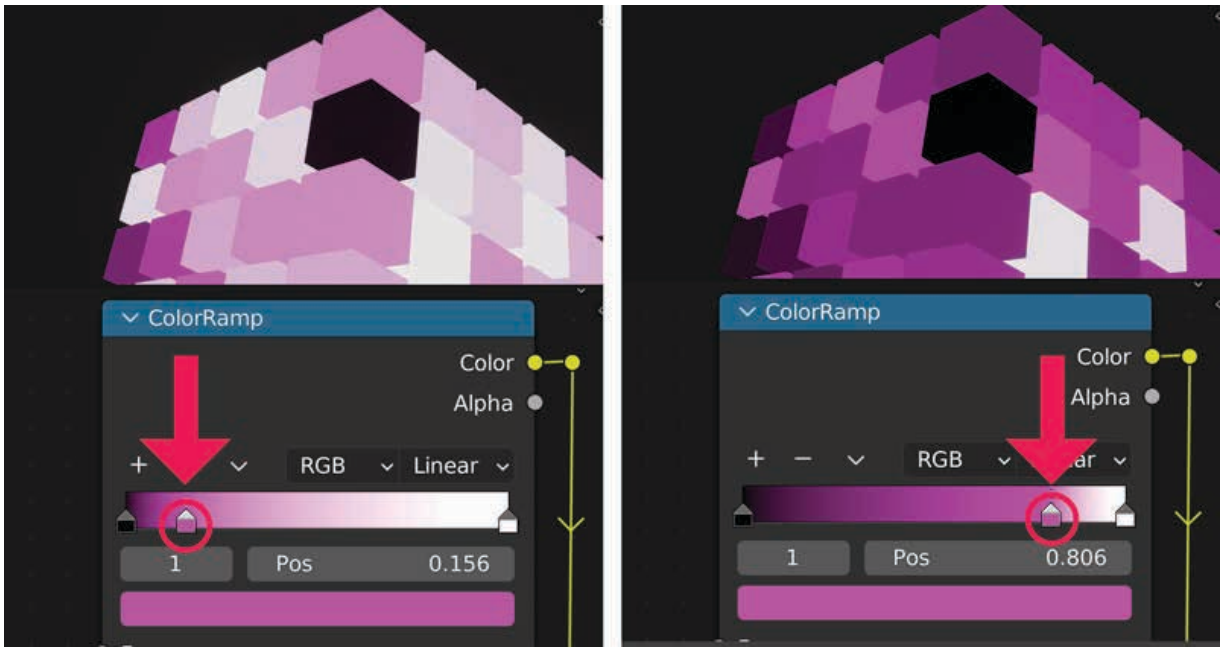
Clicking on a Stop selects it.



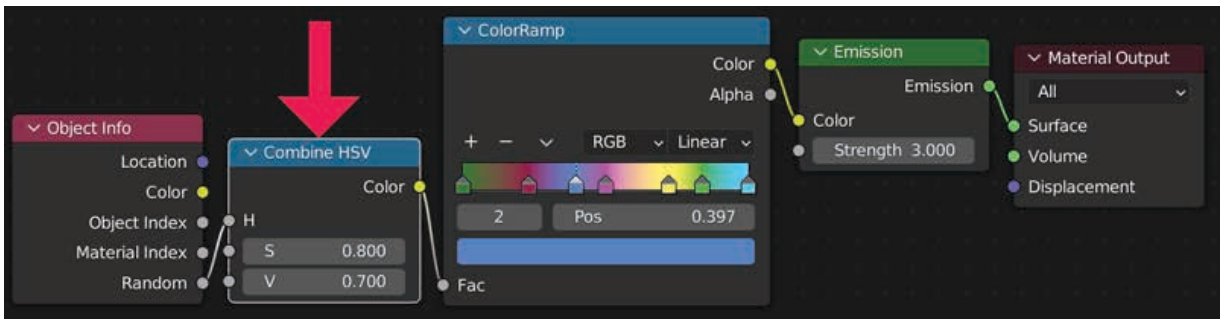
You can then change the color in the field below.



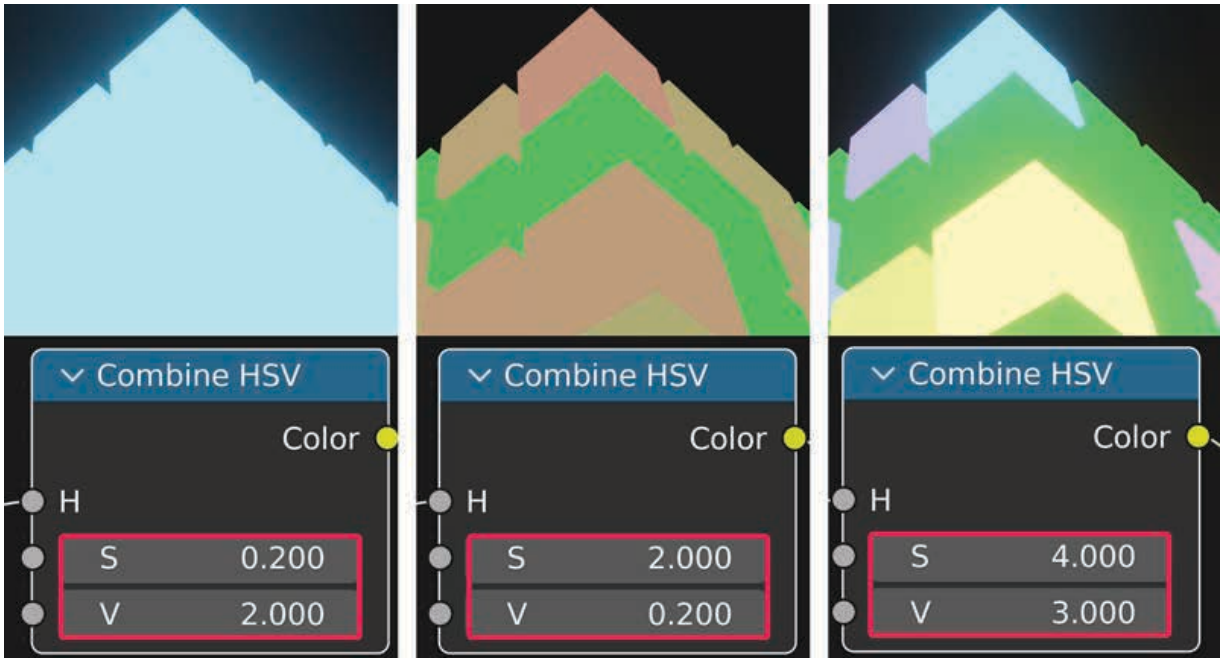
You can also move the stops if you want to, by selecting and dragging them. You can use the Location output for a different result.



Now the objects will have a color assigned to them based on their location in 3D space. For a different result, you can add a **Combine HSV node** and plug the **Random output** into the **H input**.

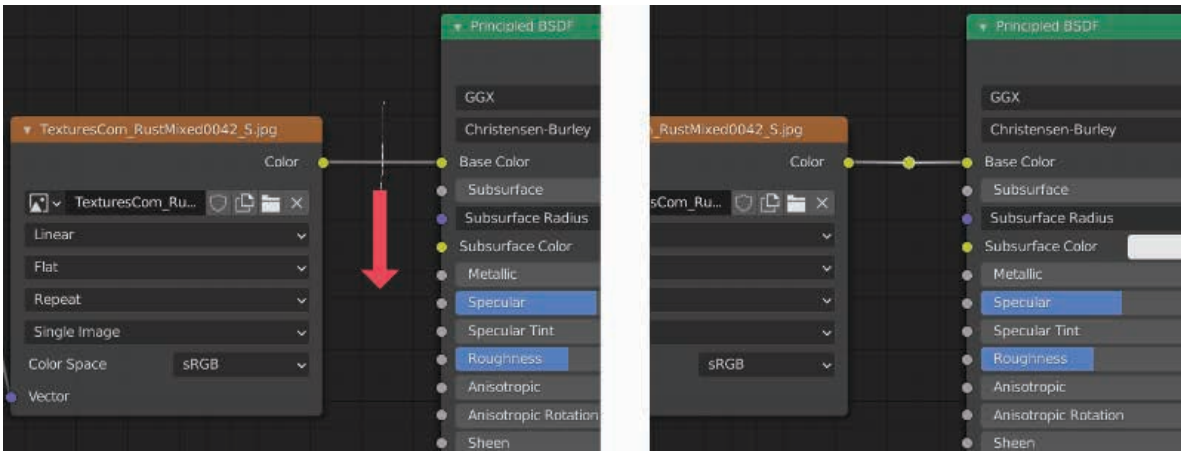


Then, experiment with the **S and V values** to see what you get.

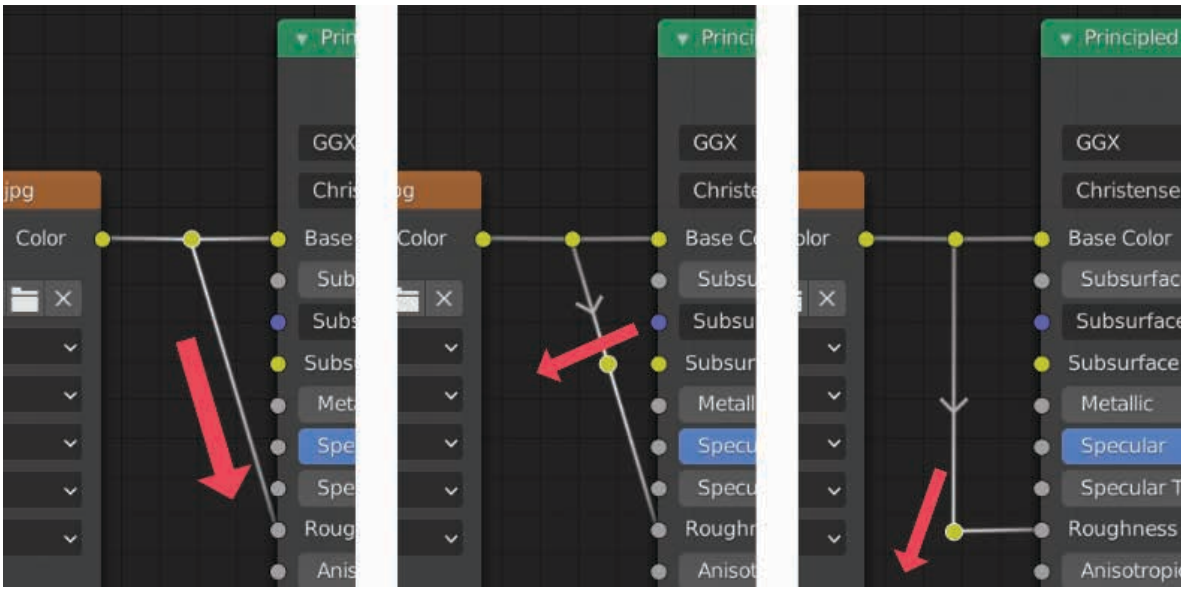


The Reroute Node

Connecting nodes is easy. Just drag one socket to the next. Node trees can get kinda crazy though. Using the Reroute Node we can at least connect the nodes more efficiently. To add a Reroute node **hold Shift and drag the right-mouse-button across a node connection**. This adds a yellow dot on the connection. This dot is actually a node.

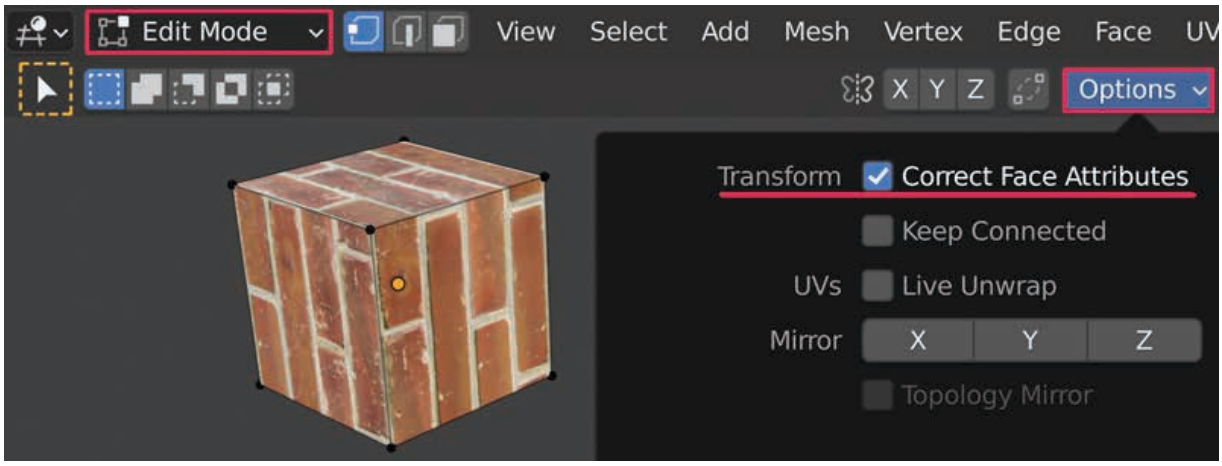


Now drag a new connection from the yellow dot to wherever you want to connect it (picture below left). You can add Reroute nodes to any connection and then **press G** to move the reroute node (picture below right) in order to create nice angular connections.

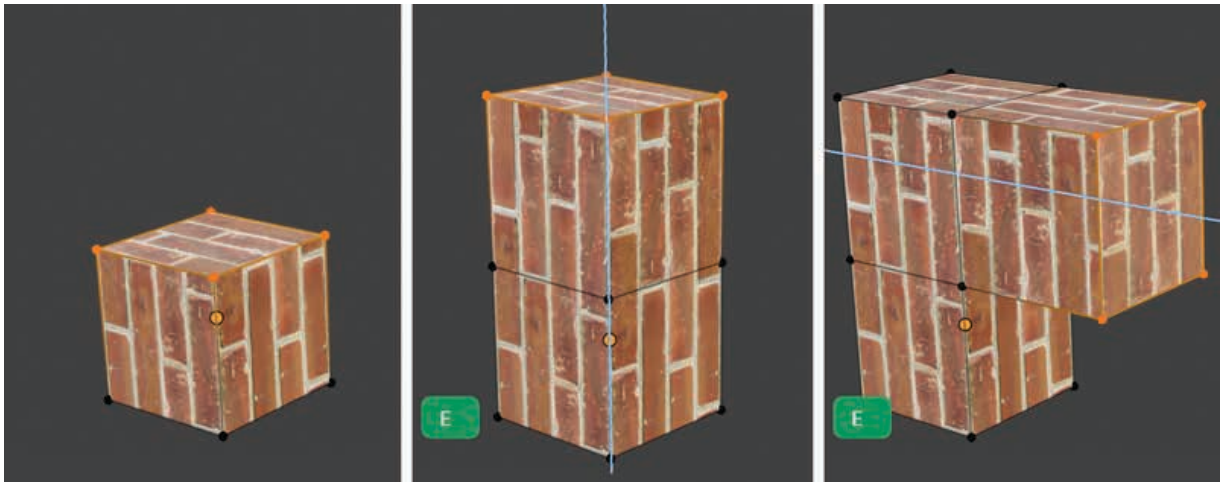


How to avoid Texture stretching

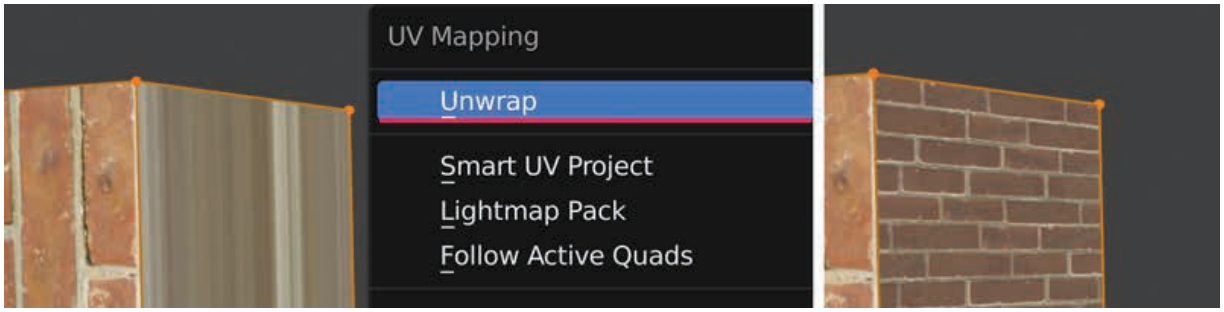
To make sure a UV map gets updated with new extruded geometry, make sure you are in Edit Mode, then go to Options and check "**Correct Face Attributes**".



Now you can move and even extrude geometry without stretching the texture.



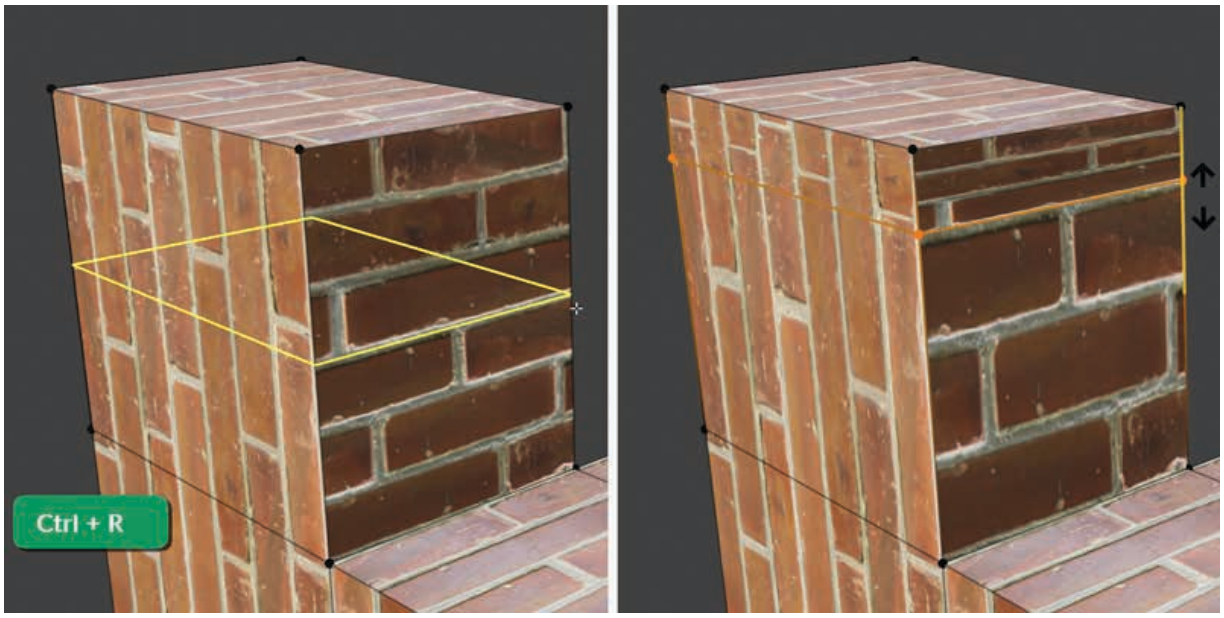
You'll notice that some textures are still stretched. In that case select the face or faces and **press U**, then choose **Unwrap**.



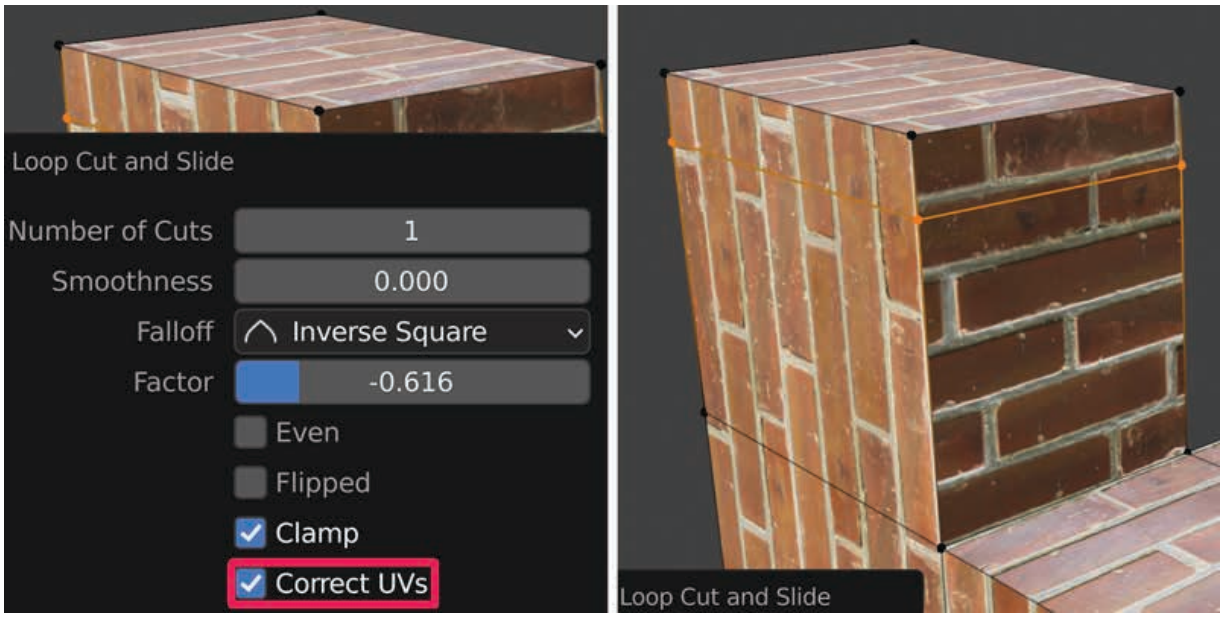
In the UV window **press A** to select all these newly unwrapped UV's...

... and scale them to the right size **by pressing S**.

If you add an edge loop by **pressing Ctrl+R** and you notice that the texture is stretching...



...open the Loop Cut panel and check **Correct UVs**.



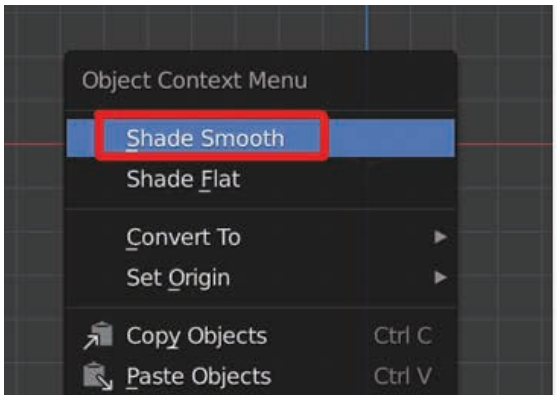
Now you can add edge loops without stretching the UV's.

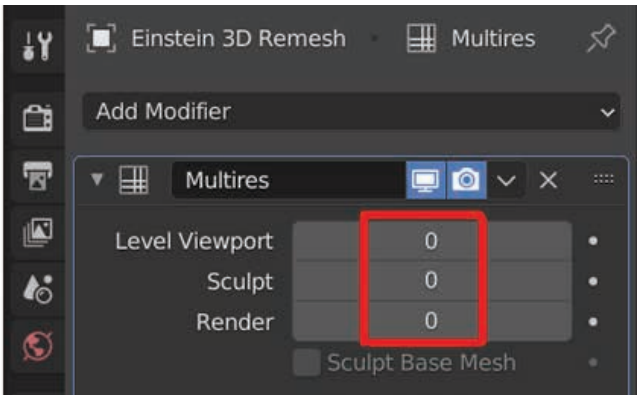
BAKING

Baking Normal Maps from the Multires Modifier

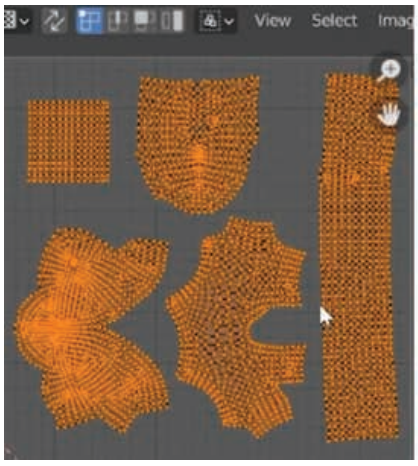
If you have a sculpt that uses the Multires modifier, you can bake the sculpted details from the higer Multires levels to a Normal map, to be used in combination with the lowest level of detail of the mesh.

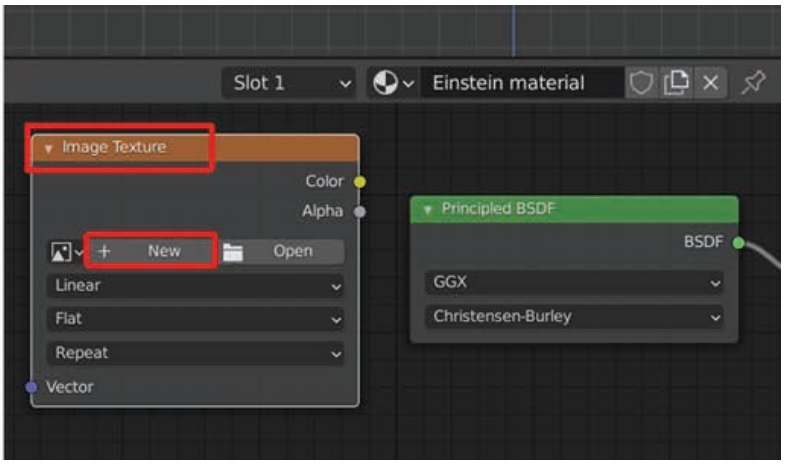
First, set the Multires Viewport **Level to 0** and set the model to **Smooth Shading.**



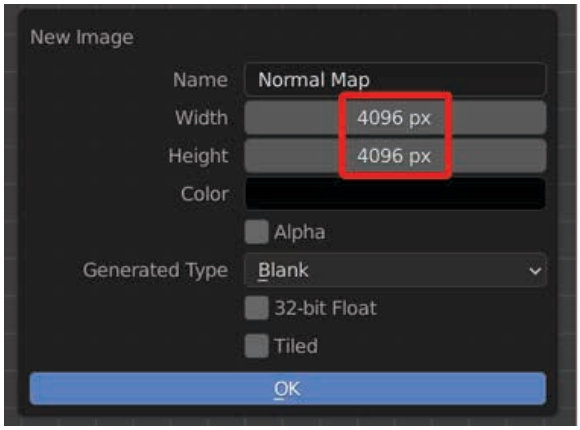


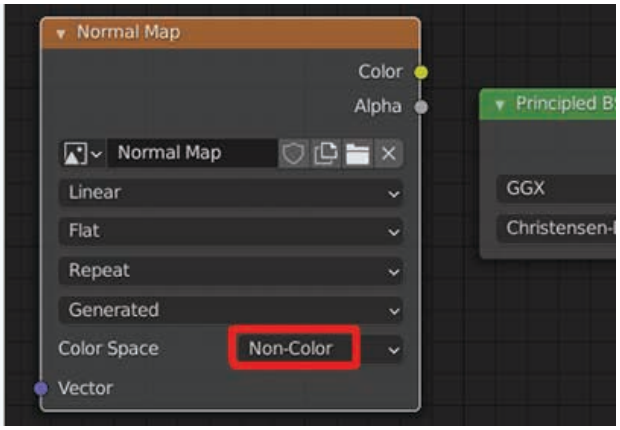
Make sure your model is **UV unwrapped** and that the UV islands have some margin in between them. Add an **Image Texture node** to the material of the object in the Shader Editor window but don't connect it to the material.



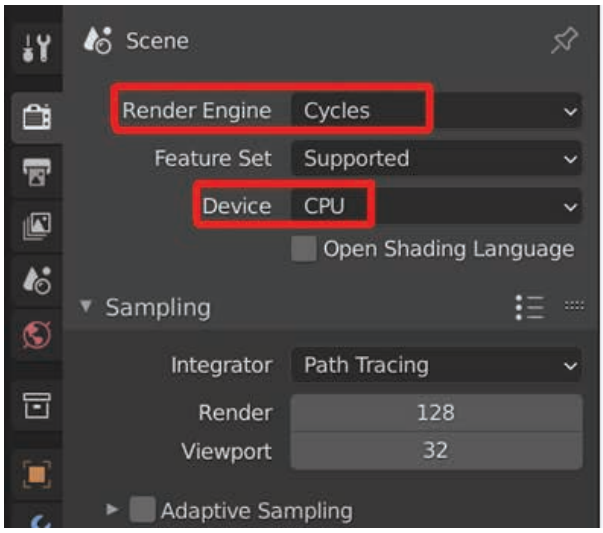


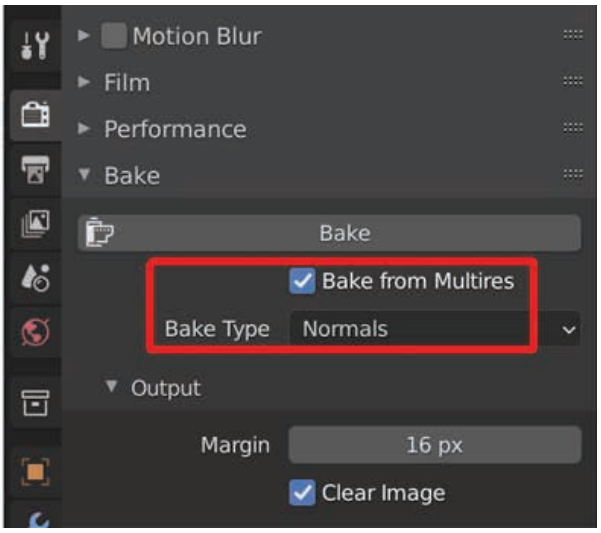
Create a new texture for the Normal Map. I recommend using a 4K texture, in other words 4096 x 4096. Then make sure the Color Space is "Non-Color" instead of sRGB.



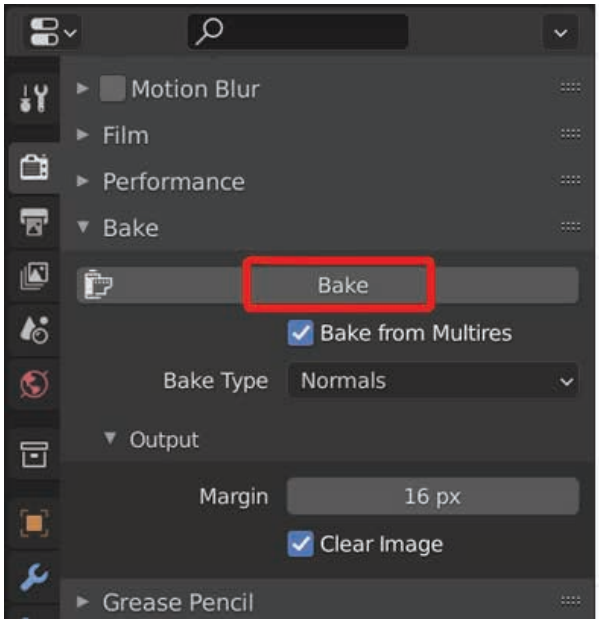


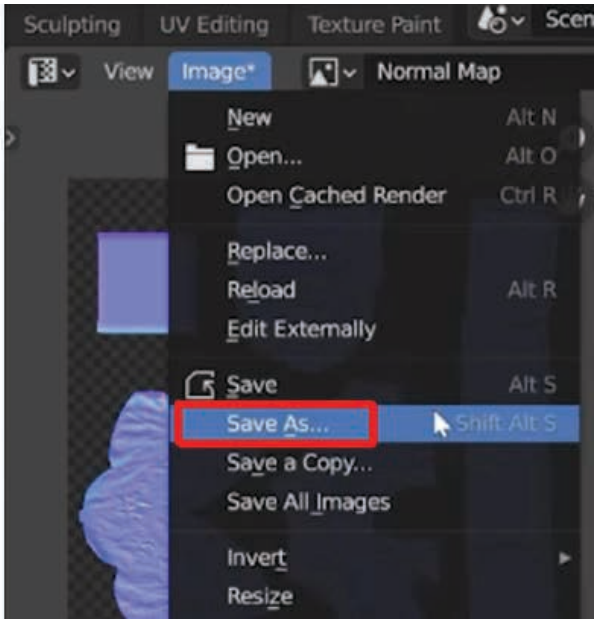
When you switch to **Cycles** render engine and use **CPU** as the device, you'll see the **Bake** options. Check "**Bake from Multires**" and set Bake Type to **Normals**.





Click Bake and after a while you'll see the result in the UV editor. Save it as an **OpenEXR** or **PNG** file.

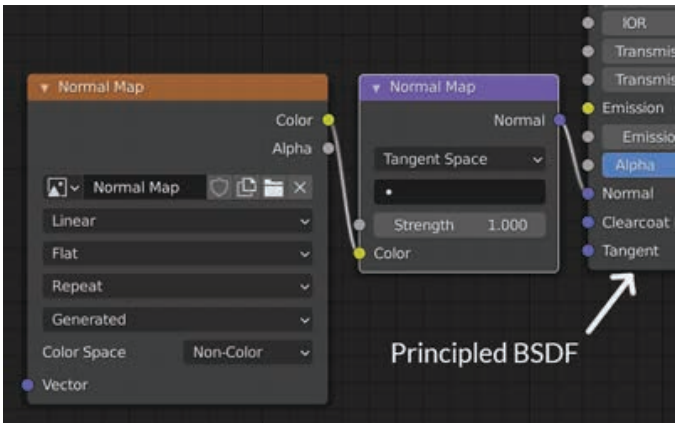




Now you can use it as a Normal map for the low poly model.

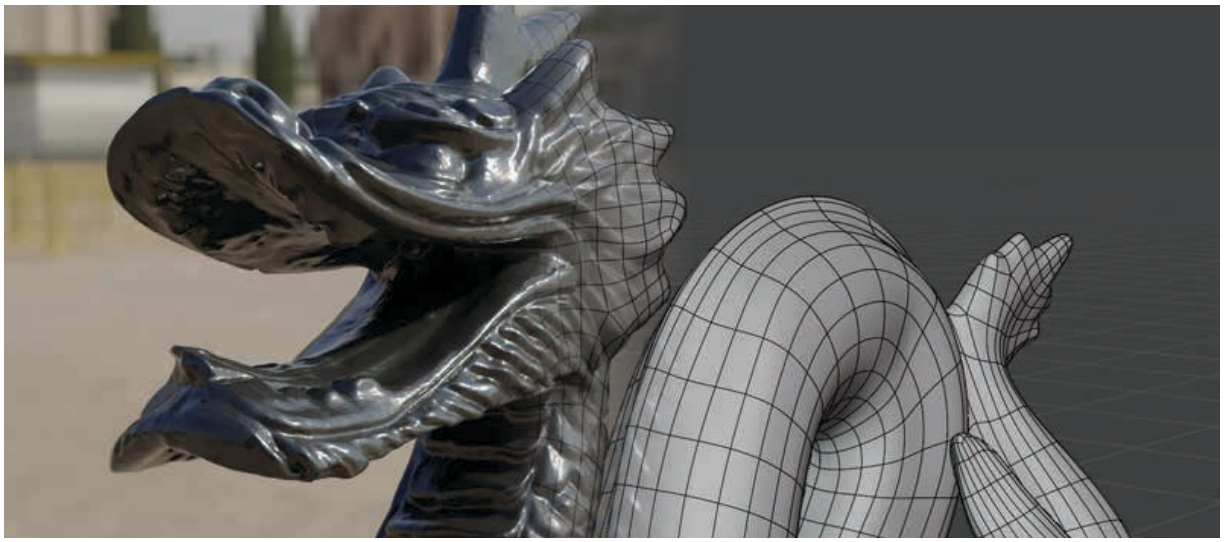
Keep in mind that while you'll see details on the model, there isn't any actual displacement. If you want that, bake a Displacement map instead.

How to bake a Displacement map is explained in the next pages.

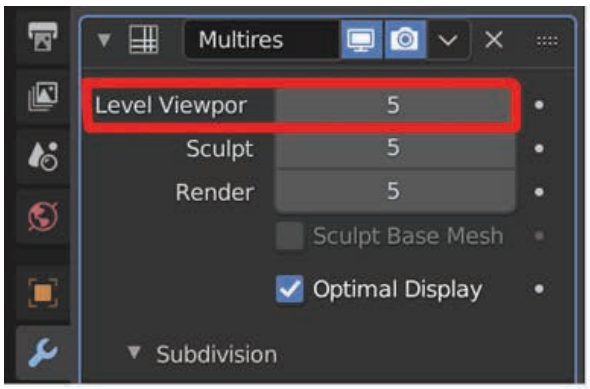


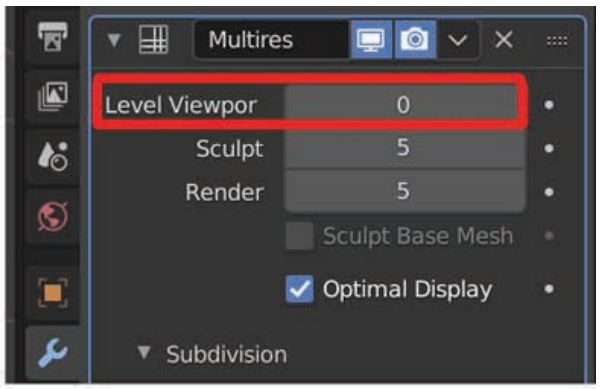
Baking Displacement maps

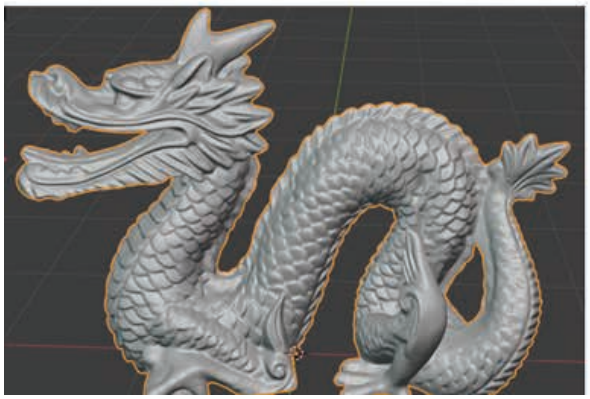
Baking Displacement maps has changed a bit from the 2.79 times, where we could use the Blender Internal render engine to quite easily bake a Displacement map. It took me a while to figure out how it worked in 2.8 and onwards, so I'm happy that I can finally share my new Displacement baking workflow with you.

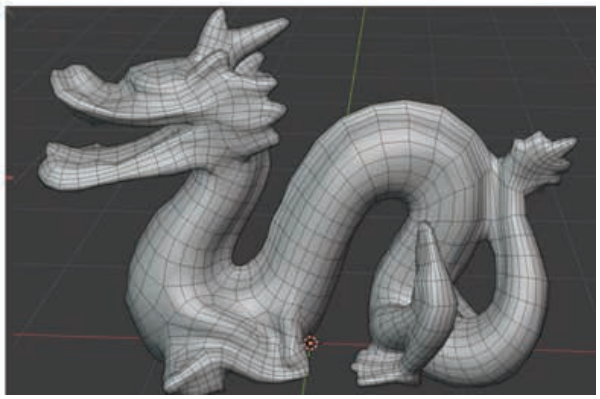


First of all, it's important that your model has a **Multires modifier** on it with different levels of detail from a base mesh all the way up to a highly subdivided, detailed mesh. If you only have a 3D scanned, high detail model for example, you can reconstruct the lower base mesh with the method explained in this video.

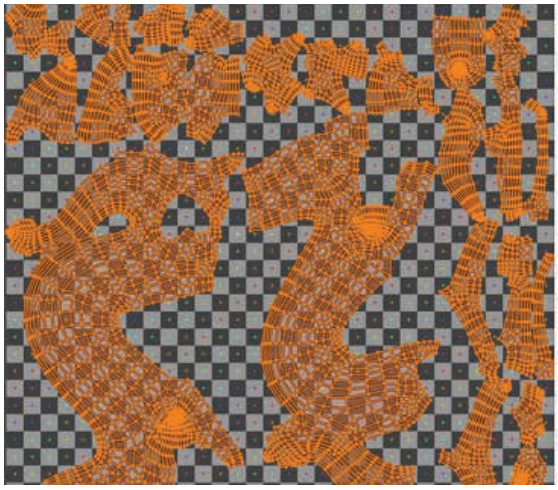


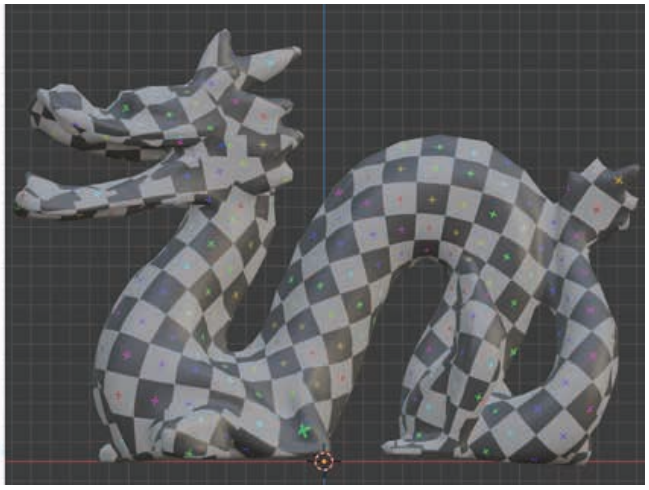




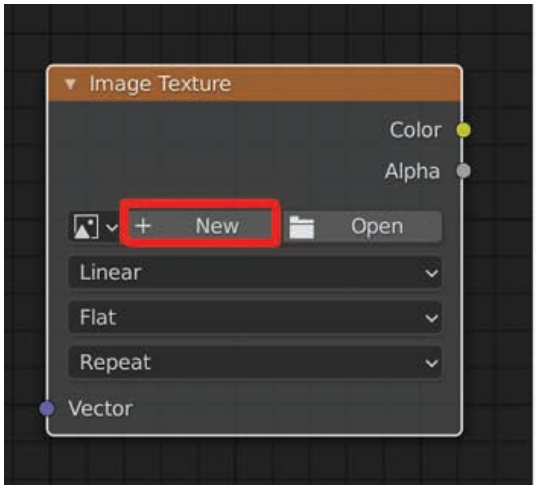


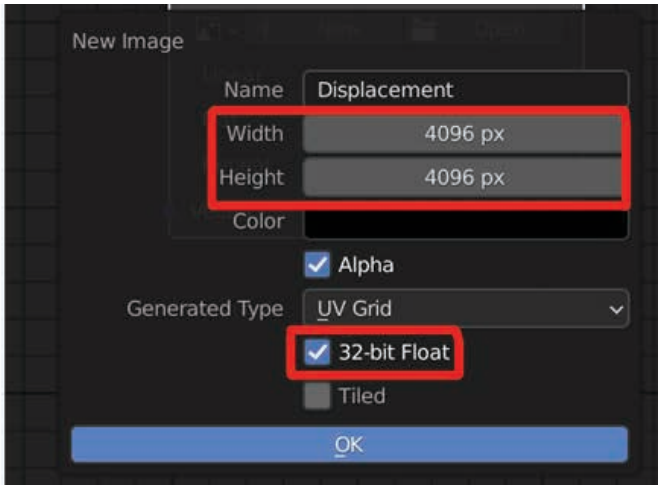
You also need to UV-unwrap the mesh.



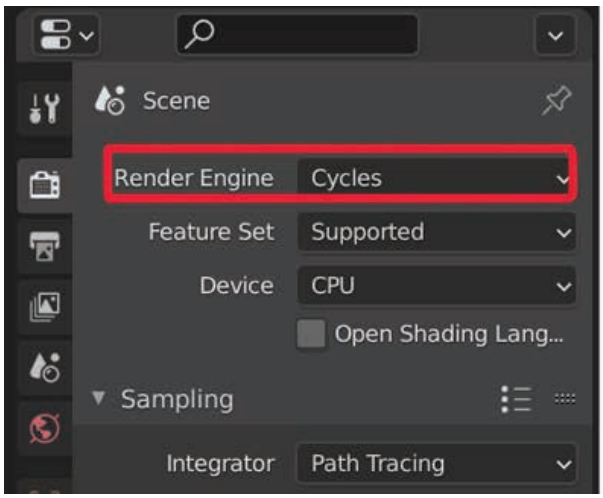


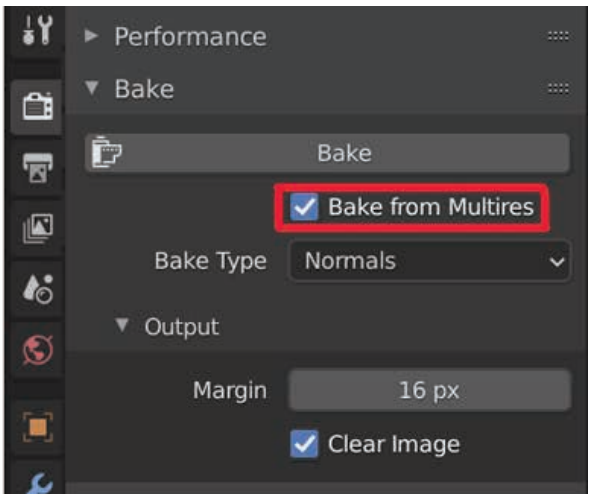
Now, create a new texture. I'd recommend going with at least **4096 x 4096** (4K) but more would be good too. You can always divide it into several UDIM textures, as explained here. Name it "displacement" or something similar, and check "**32 bit Float**". The texture doesn't need to be connected to anything.



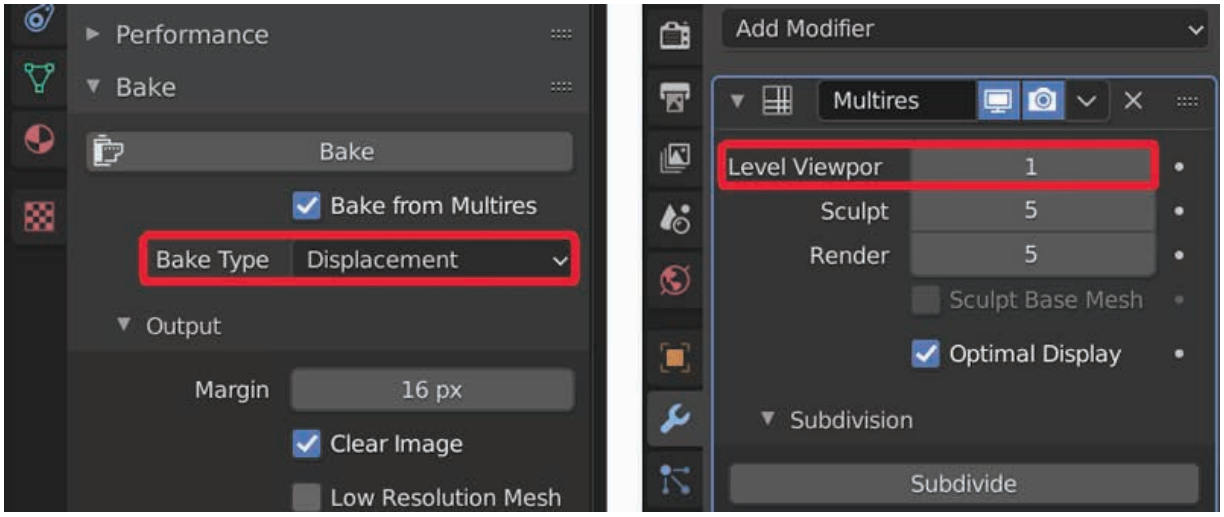


Now, very important: switch the Render Engine to **Cycles**. Then scroll down to the **Bake** section and check the "**Bake from Multires**" box.

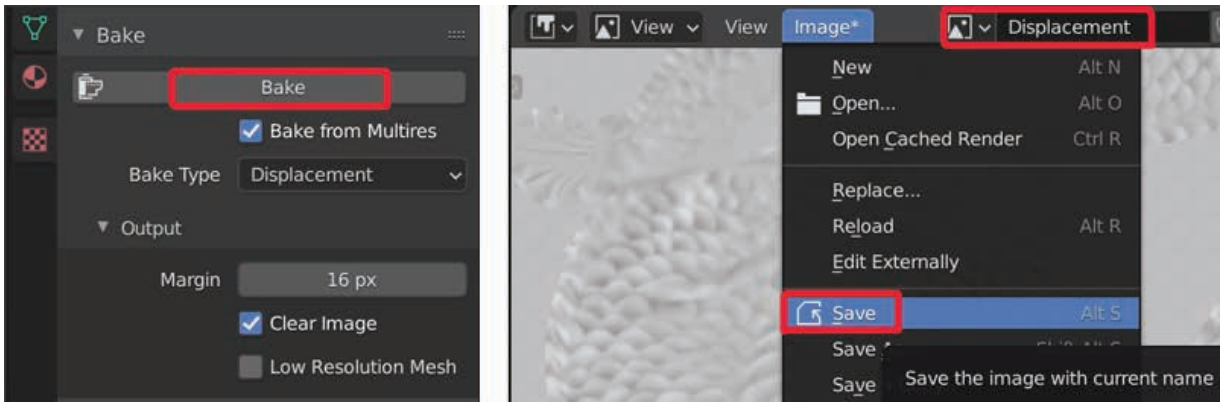




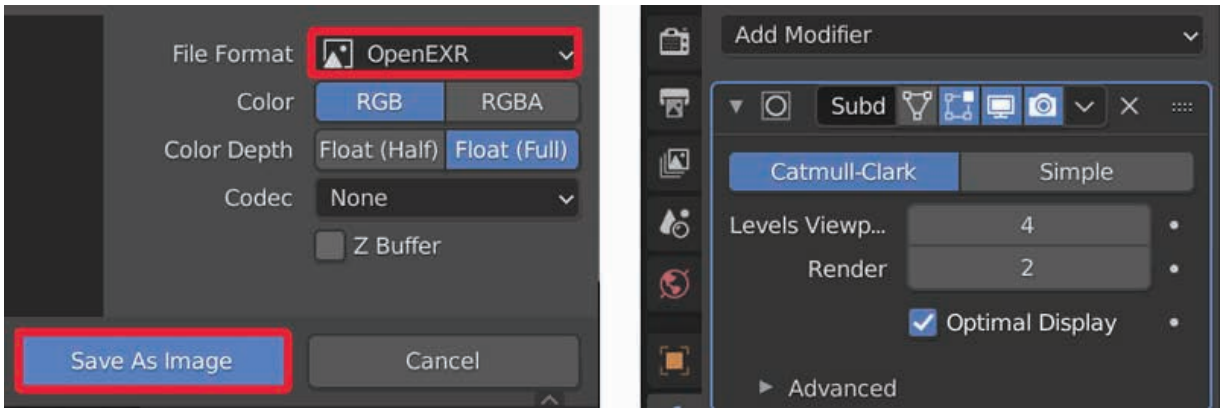
Set **Bake Type** to **Displacement**. In the Multires modifier, set the **Viewport Level to 1** (or 0, but that sometimes leads to some artifacts in the Displacement map).



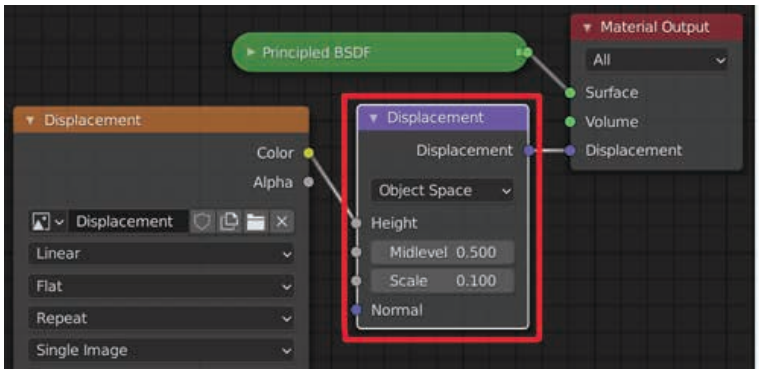
Now just **click on Bake** (make sure the object is selected, otherwise it won't do anything), and wait while Blender is baking the Displacement map. When that's finished, open an **Image Editor window** and select your Displacement texture from the drop-down, then **Save** it.

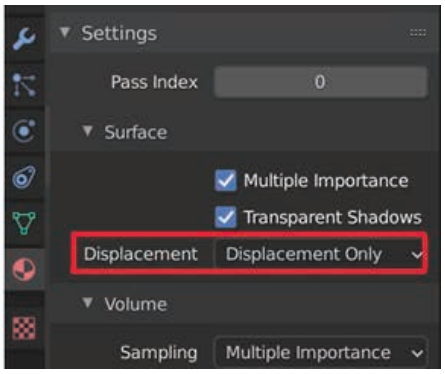


OpenEXR is a good file format for Displacement maps as it doesn't have compression. You can choose **Half Float** (16-bits) or **Full Float** (32-bit) when you save it. 32-bit is better, but the file size will be quite big. After saving a copy of this Blend file, remove the Multires modifier and add a regular Subdivision modifier instead.

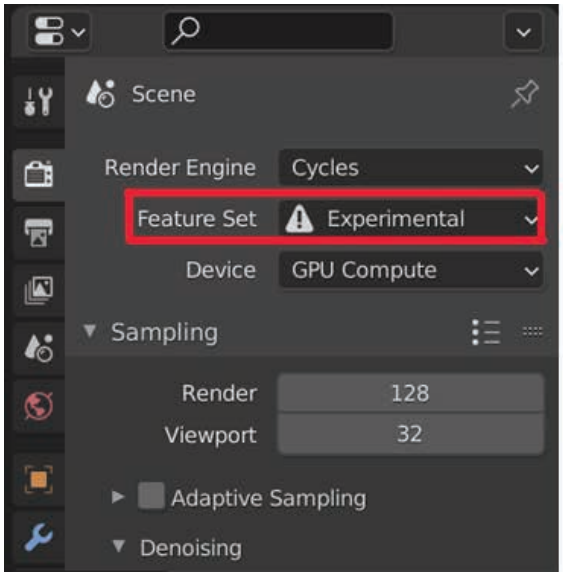


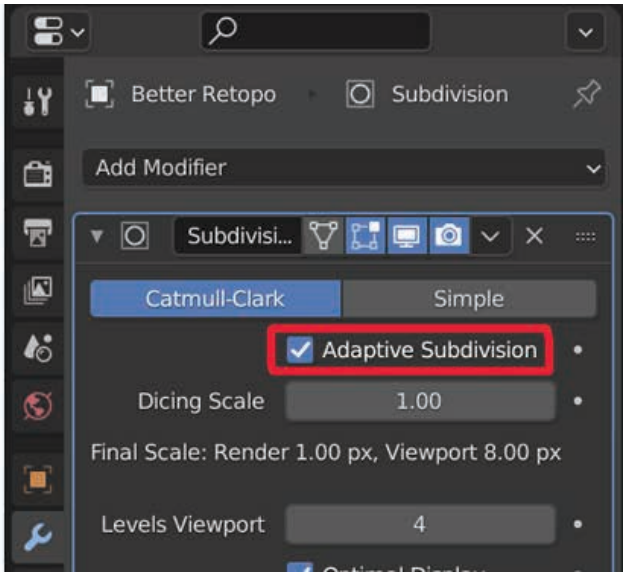
Add a **Displacement node** with the Displacement texture, then set Displacement to "**Displacement only**" in the **Settings** of the Material. Now you should see the displacement in Rendered Preview.





For best results with Displacement maps, set the feature set to "**Experimental**" and check "**Adaptive subdivisions**" in the Subdivision modifier. This way, things closer to the camera will receive more subdivisions, and things further away will receive less subdivisions. This saves on render time, while ensuring great looking displacement.



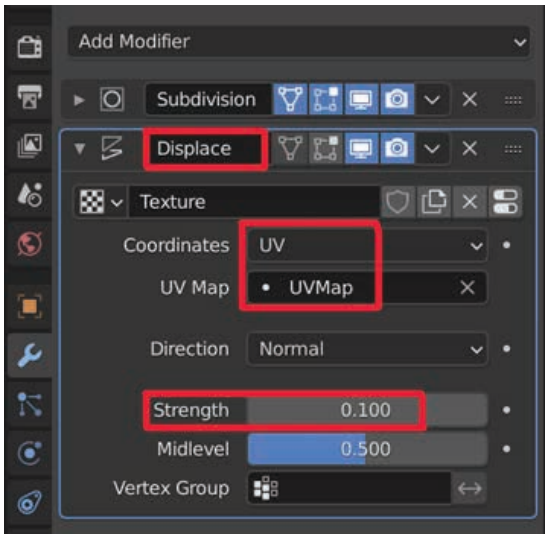


To use the Displacement map in **Eevee** is simple, you just need to use a **Displacement modifier** instead of a Displacement node.

Remove the node, and add the modifier, with the Displacement texture.

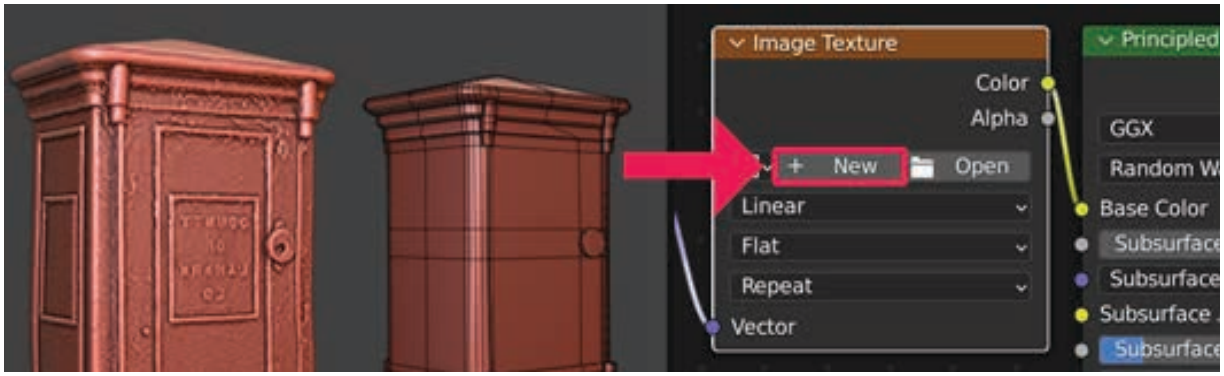
Set **Coordinates to UV** and then reduce the **Strength**.

I find that 0,1 is usually enough.



Bake 3D scan textures to Low Poly meshes

To transfer the texture from this 3D scan to this simple retopologized model, first give the retopo mesh a new material with an image texture. Create a new texture.

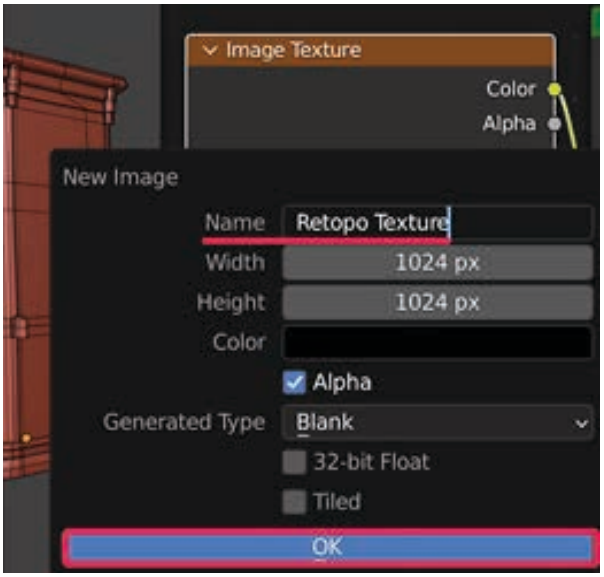


Give the texture an appropriate name.

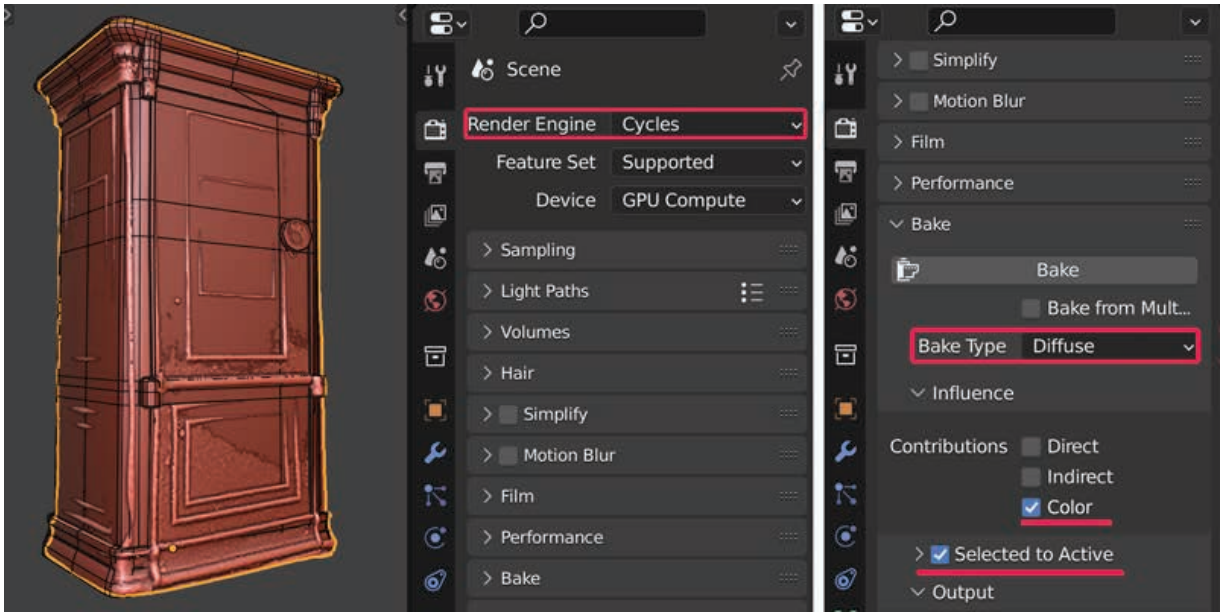
It's really important to use clear names, so that you don't get confused.

For example, you don't want to accidentally bake the empty new texture to the existing 3D scan texture, and override all that texture information by accident.

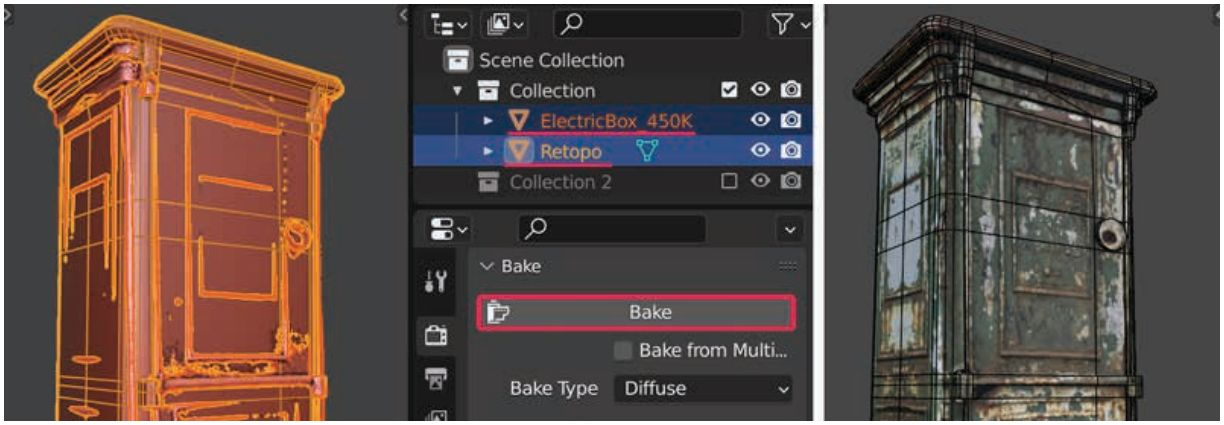
As long as the names are clear and you select in the right order, that won't happen.



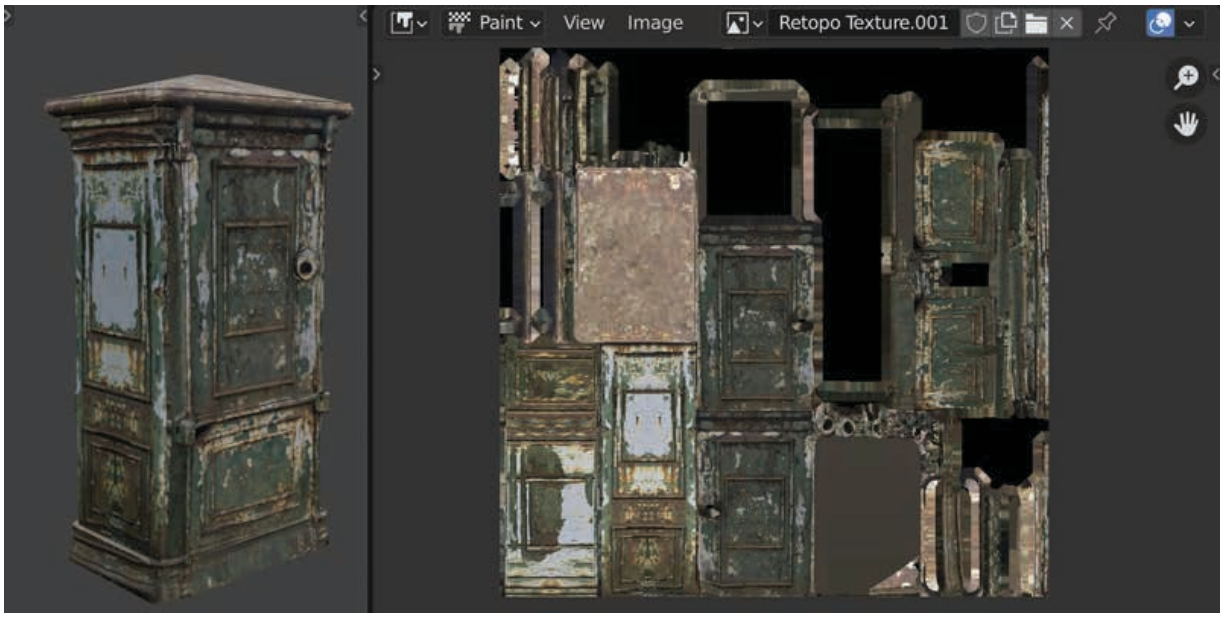
Make sure both meshes are in the same location. Go to the Render tab, set it to **Cycles** and go to the **Bake** options. Set Bake Type to **Diffuse** and set it to **Colour**. Check "**Selected to Active**".



Select the 3D scan and **hold Shift**, then select the retopologized mesh **in the 3D viewport.** Then click on **Bake**. Wait for a while as Blender is baking. *(Note: the same selection order applies in the Outliner but with holding Ctrl instead of Shift!)*

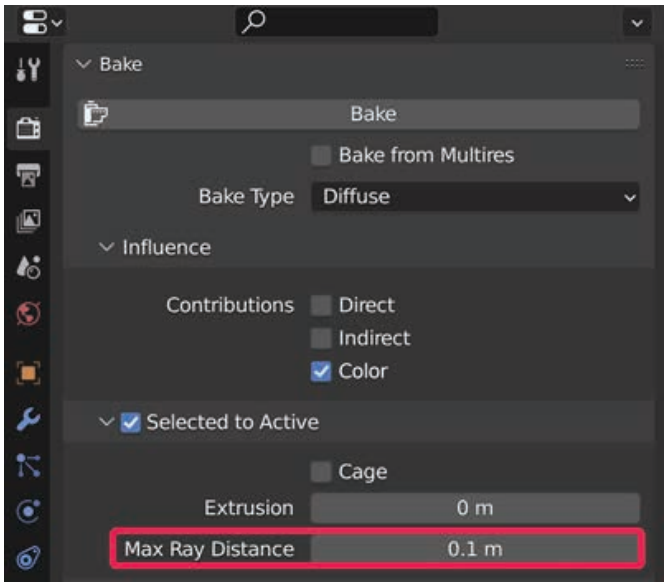


Now you can see that the retopologized low poly mesh has a nice baked texture with simple UV's.



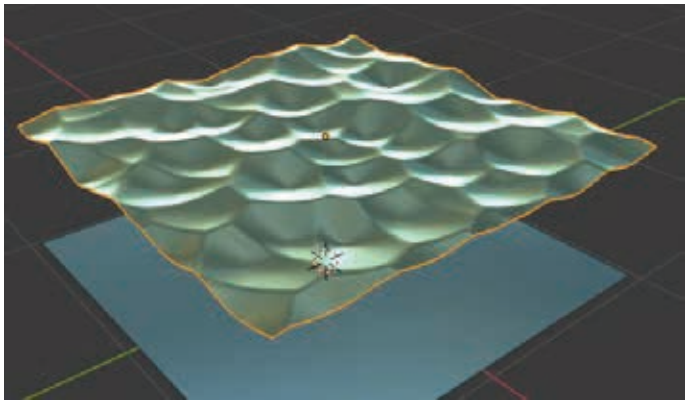
You may need to tweak the **Ray distance** a bit.

In this case 0.1 worked well instead of just 0.



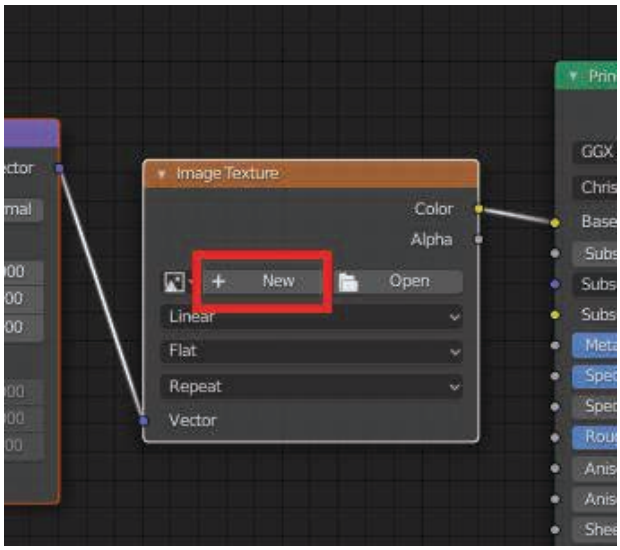
Baking normal maps

Normal maps are a seemingly magical way of adding small detail to surfaces without using any additional geometry.

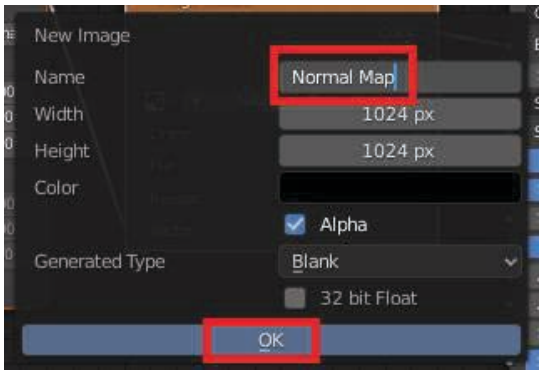


Here we have two planes. The one on top has lots of faces and displaced details. The plane below it is just one face, which is UV unwrapped.

Create the bottom plane a new material. Add the texture nodes by **pressing Ctrl+T** with the Principled BDSF selected. Create a new texture.

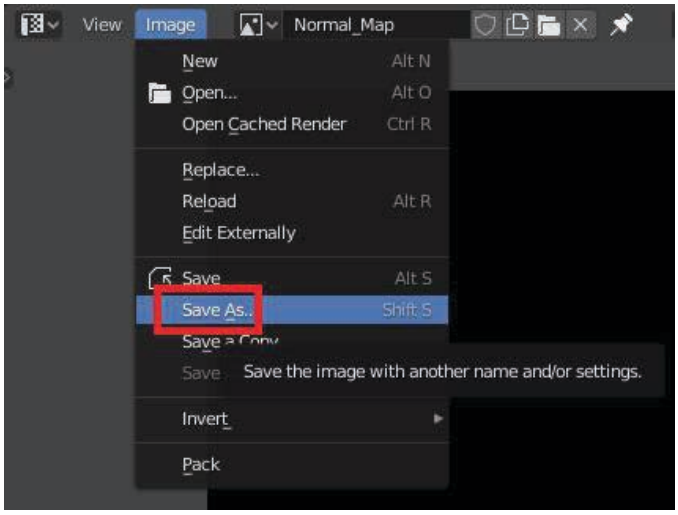


Give it a name like “Normal Map”, something clear.



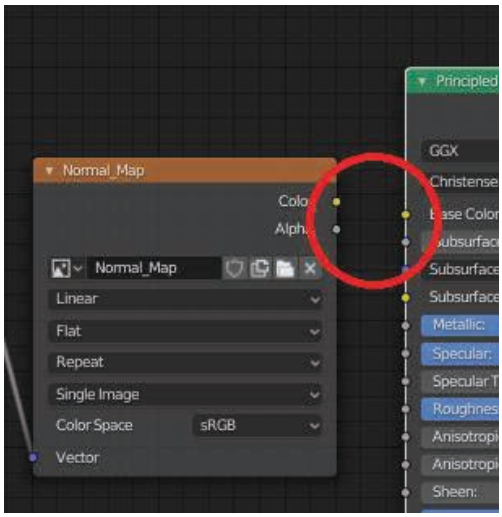
Click here to return to the index.

Open the texture in the UV Editor window and Save it somewhere.

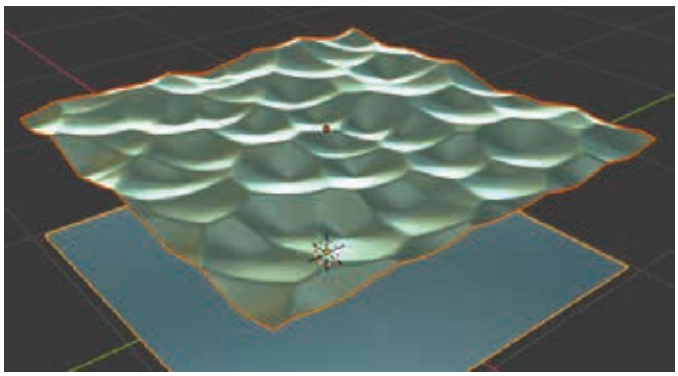


Disconnect the Image texture node from the material.

To do this, click on the yellow dot next to “Base Color” and drag it away, then let go.

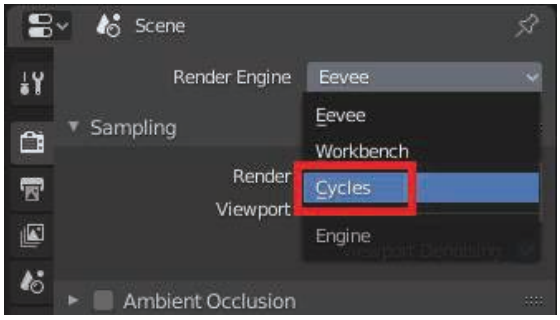


Select the top plane, then **Shift-left click** on the bottom one.



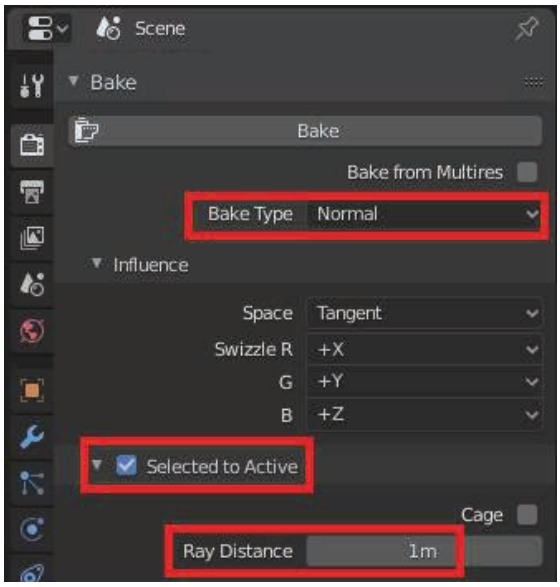
Set the render engine to Cycles. This will give you the Bake

options.

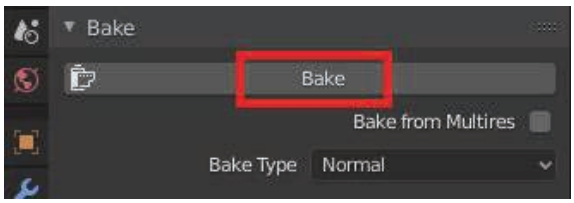


Set **Bake Type** to **Normal**.

Check **Selected to Active**. Set Ray Distance to 1.



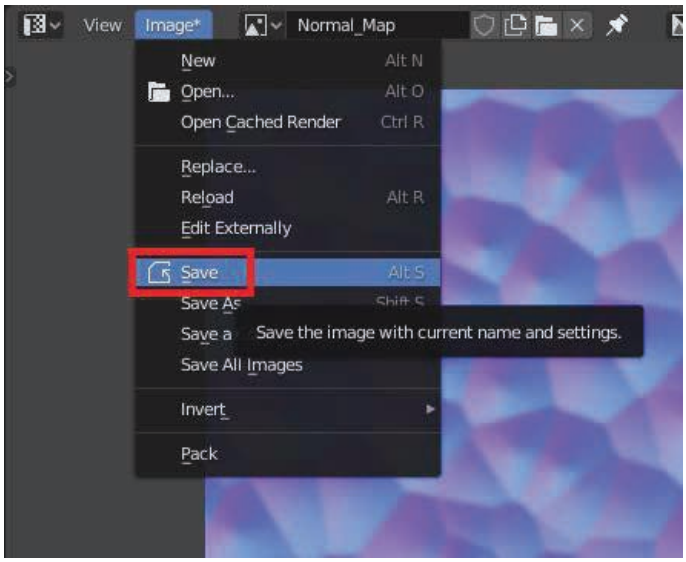
Click the **Bake** button.



You'll see the texture update to a colorful Normal map.

Make sure you **save** again.

Read the next tip to find out how you can use this Normal map.

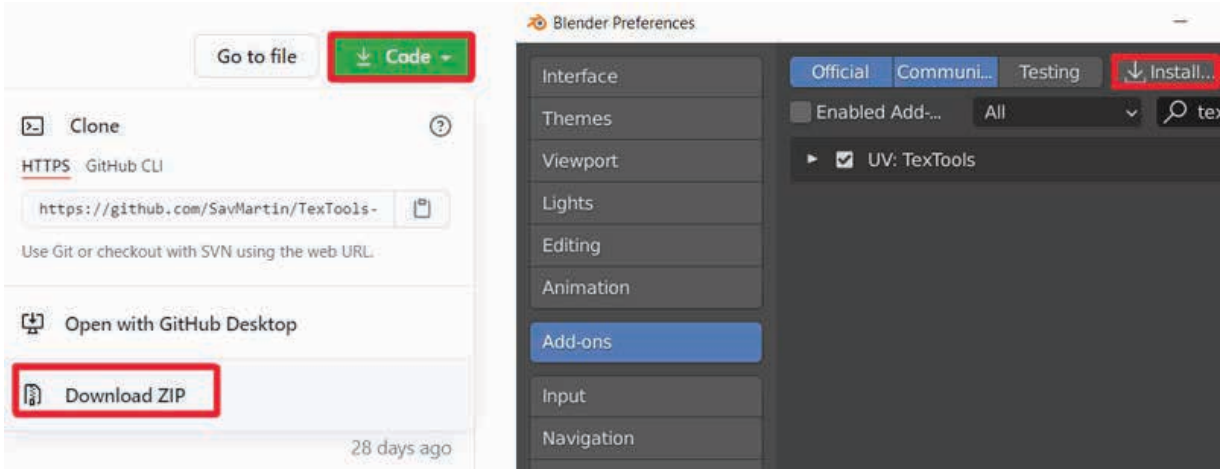


Baking textures with the free TexTools add-on

TexTools is a very useful free add-on originally developed for 3Ds Max and ported to Blender after its developer discovered Blender.

It adds a lot of useful functionality to the UV Editor window. Today, let's have a look at its baking options.

To use TexTools, you'll need to download it from Github first, as that's where you can find the version which works for Blender 2.9. Then, install the ZIP file from the Preferences.

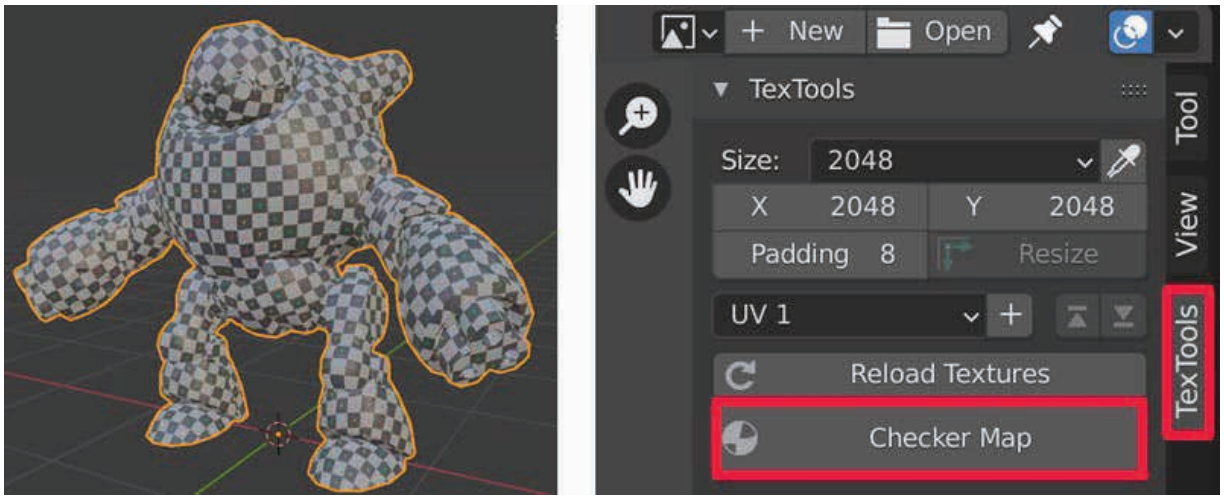


After this, you can find the TexTools tab in the **Option panel of the UV Editor window**.

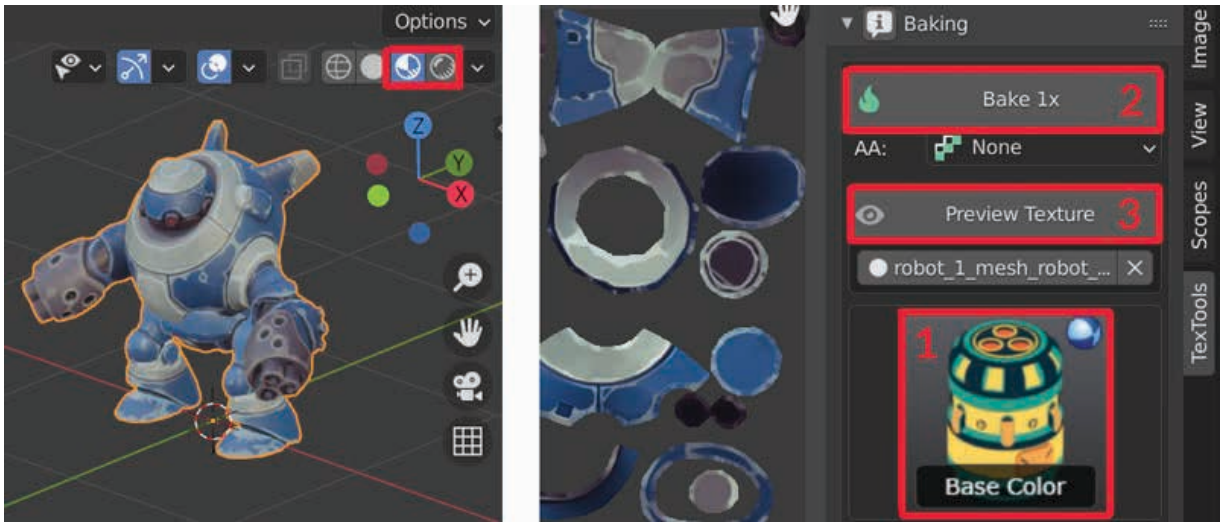
Open the Option panel by **Pressing O**.

It comes with many convenient baking options. You can bake normal maps, ambient occlusion, cavity and dust maps, and many more.

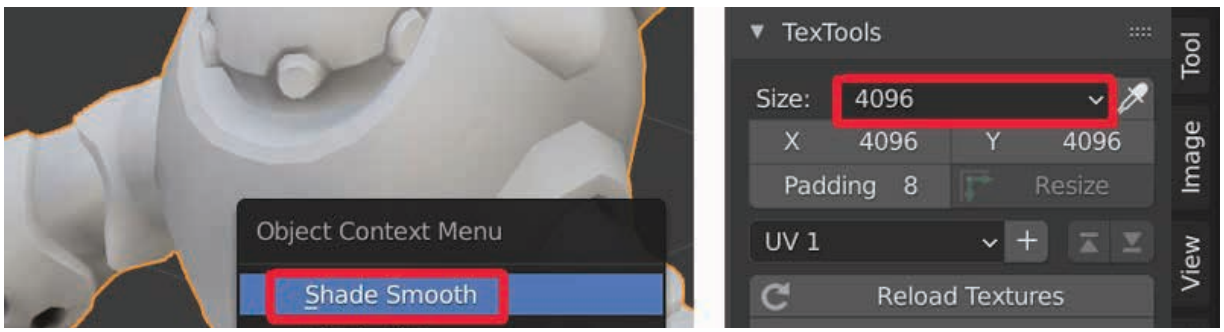
Press the **Checker Map button** to add a checker texture to your object. This is useful for seeing how the object is unwrapped before adding your final texture. Click the Checker Map button again to cycle through some different styles of checker maps.



To bake a texture is pretty easy. Click on the bake icon to select a type of bake, then **click on Bake 1x**. After that, **click on Preview Texture** to see the texture on your model in the viewport, provided that you are in either Material or Rendered viewport shading mode.

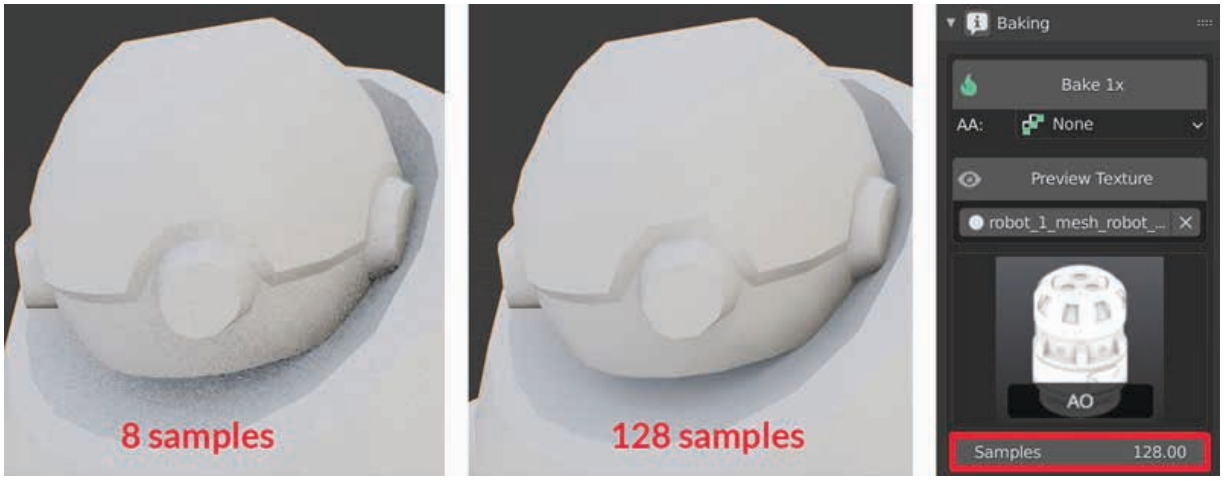


If the texture resolution is too low, you can always change the size and bake the texture again. For some textures, the result is better if you set your model to **Shade Smooth** first.

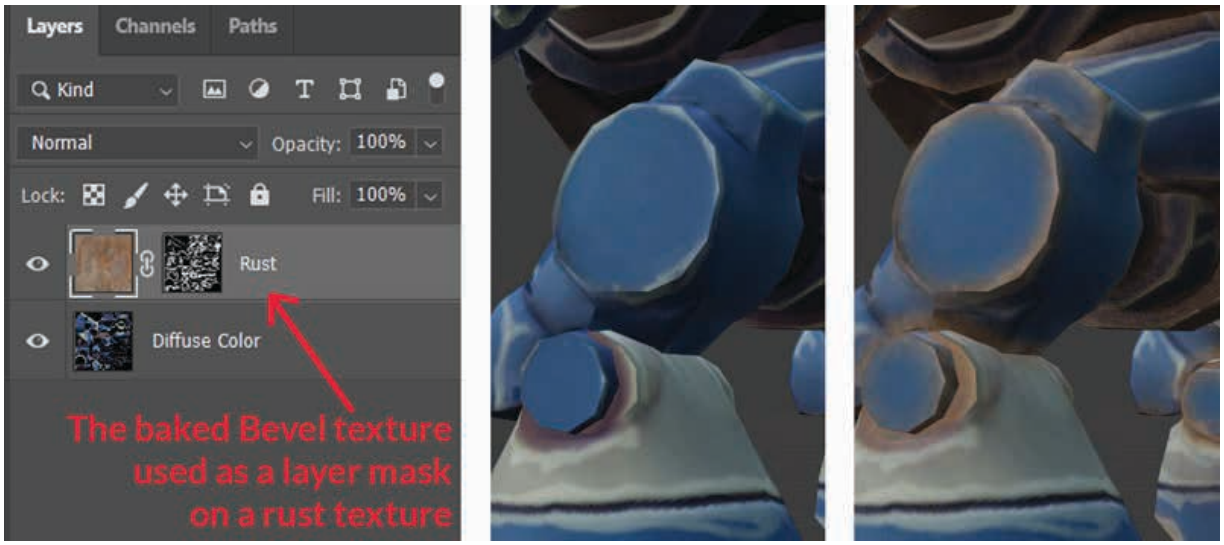


After baking, you need to save your texture from the UV Editor window, by going to **Image > Save As...** and saving it somewhere.

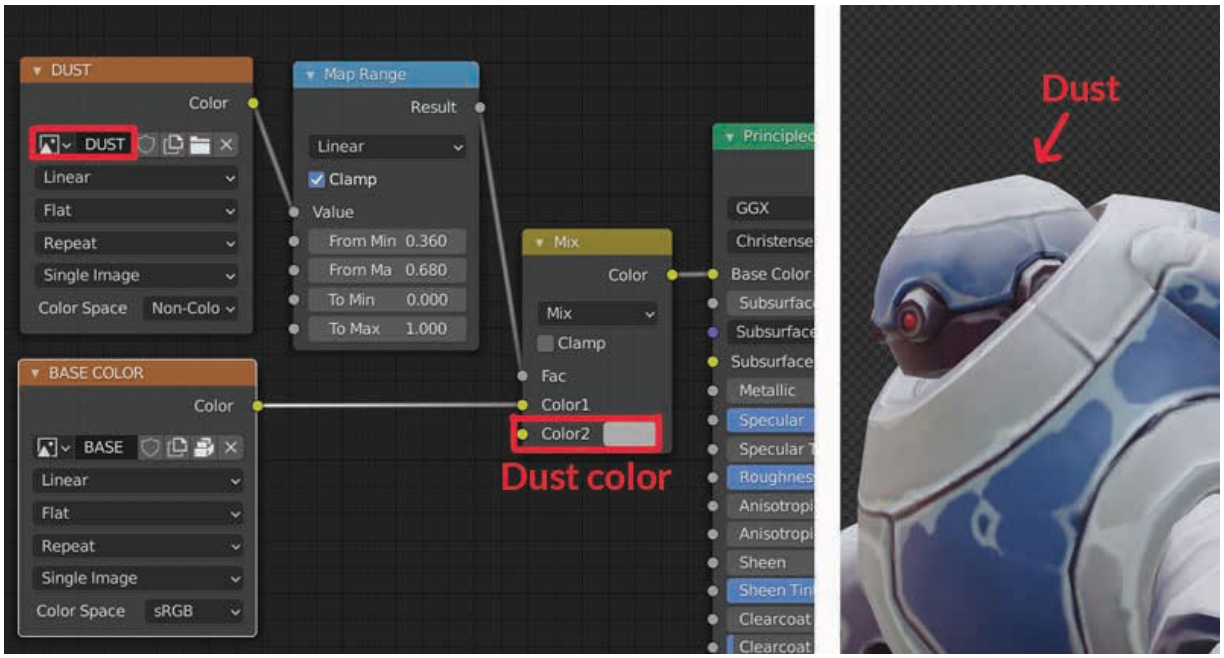
For some texture types such as AO (Ambient Occlusion) you will notice that the texture is a bit grainy when using the default settings. For these bakes you can increase the **amount of samples** in the baking menu to increase the quality.



Where a lot of these textures come in handy is with manual texture painting. You can use the Paint Base, Bevel, AO, Base Color textures for example in your favorite image editing software as layers, and combine them to create a base to paint further details on, or as masks for layers, like in the below example in Photoshop (you can do the same in free software like Krita or Gimp).

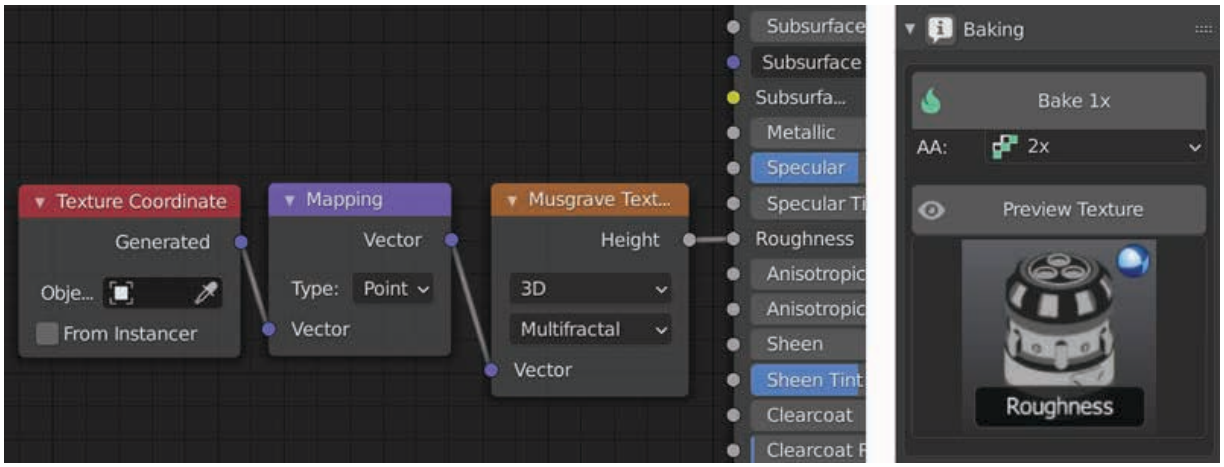


Of course you can also use them as masks to mix nodes in the Blender Shader Editor. The Paint Base, AO, Bevel Mask, Dust and Cavity textures are great for creating an aged effect on your model. In the below example, I've used a Dust texture to mix the Base Color texture with a light grey color, creating a dust-like effect on the model.



Another good reason to bake textures is for exporting your material to a game: imagine you have all kinds of special Blender-specific nodes set up to create an interesting noise pattern for the roughness of your object (below left).

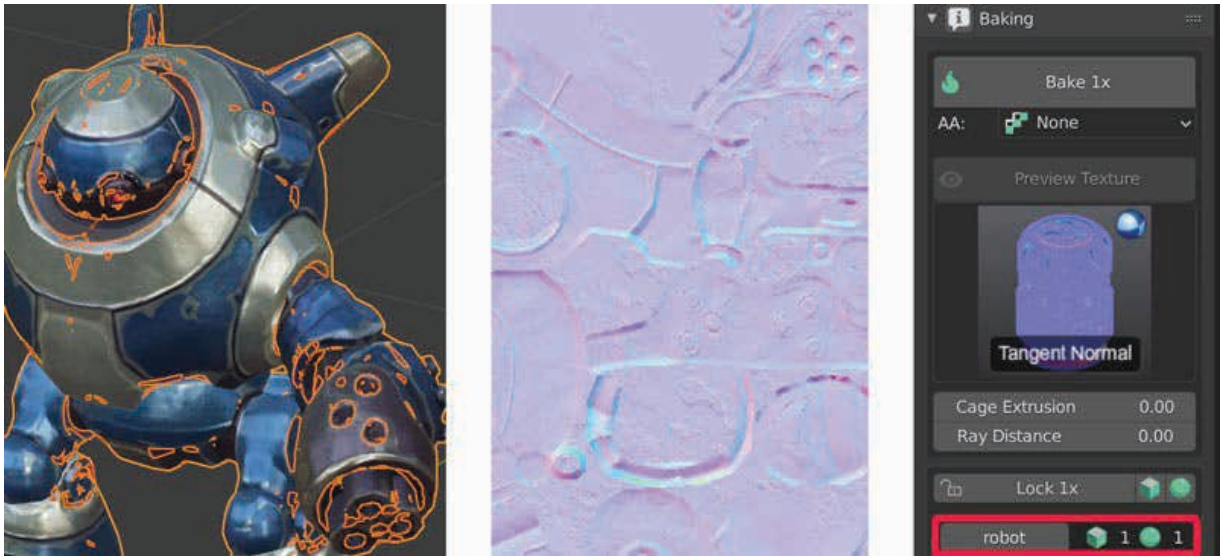
You can't export those nodes to your game engine, but you can bake them to a PBR-ready texture by selecting "**Roughness**" as the bake type and baking them to a single black and white texture.



How to bake a Normal map with Tex Tools

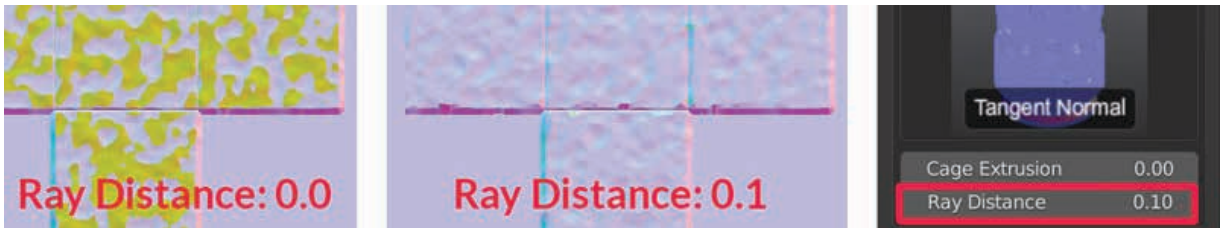
Baking normals assumes that you have both a **high poly and a low poly object**. Select both objects. The order in which you select them doesn't matter, however it's essential that the objects have "**_highpoly**" and "**_lowpoly**" affixed to their name, otherwise Tex Tools won't know which is which.

Choose "**Tangent Normal**". You can see that the "bake set" contains both objects. The low res object is symbolized by a cube icon, the high res object is symbolized by a sphere.

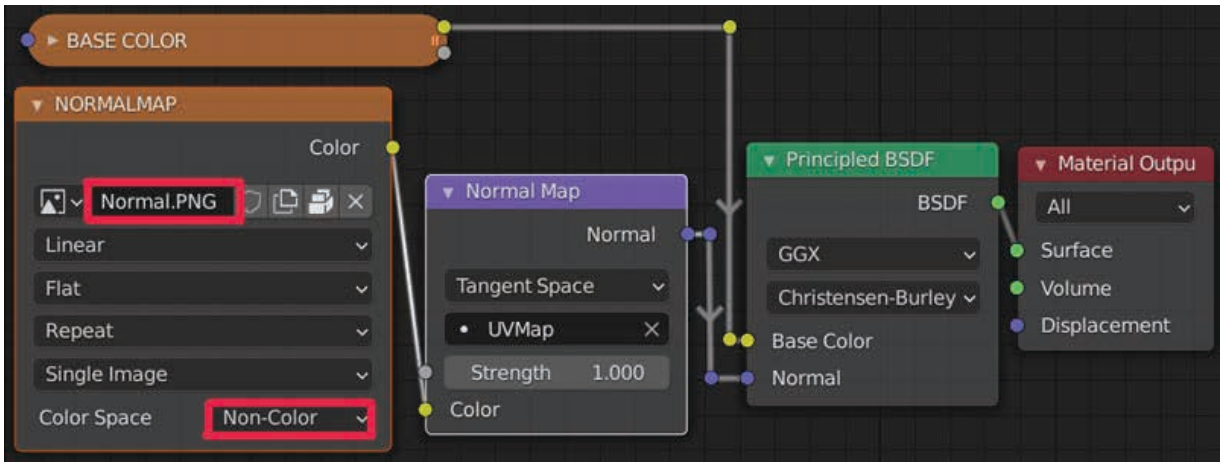


If textools gives you the error "Need High and Low" even though both a high and low poly object are selected, hover the mouse cursor over the Tex Tools interface, and it will update.

Now **click on Bake**. If there's this kind of yellow color in the resulting normal map, increase the **Ray Distance value** slightly (0.1 is usually enough in my experience) and bake again, until the yellow color is gone.

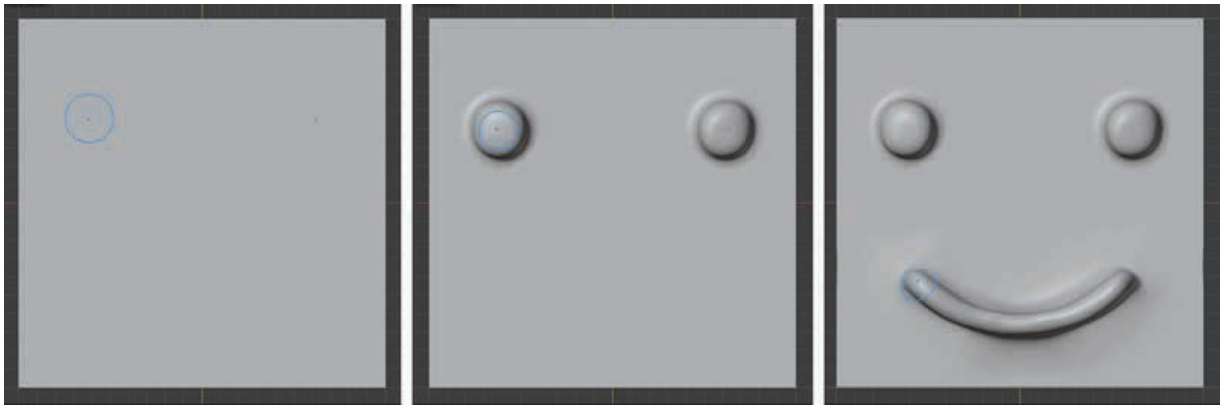


Save the Normal map from the UV Editor window. Now you can use it on the low poly model.



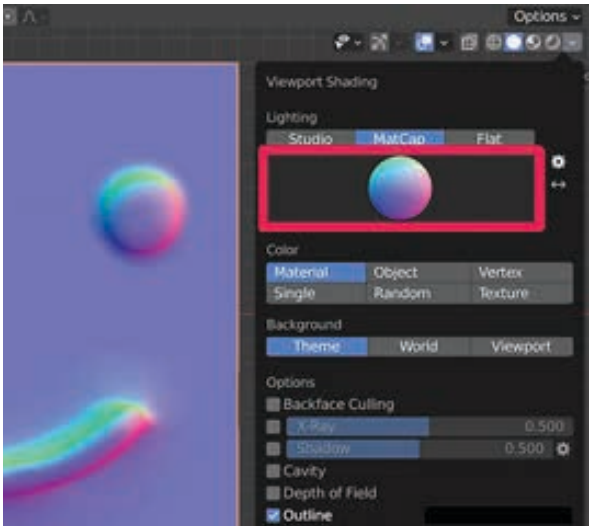
Make a Normal Map using a Matcap

Sculpt some detail on a subdivided plane.

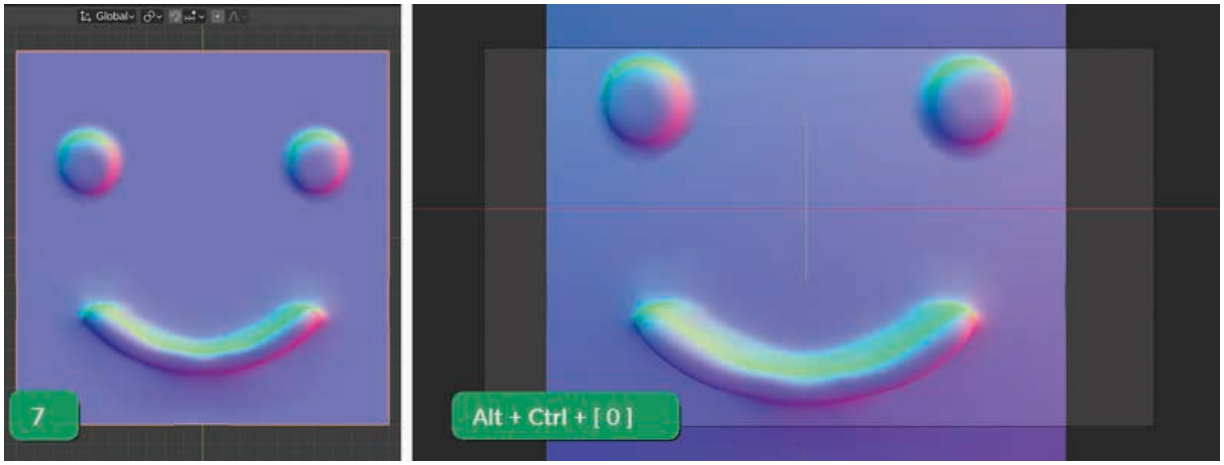


Turn on **Smooth Shading.**

Add this colorful **Matcap**.

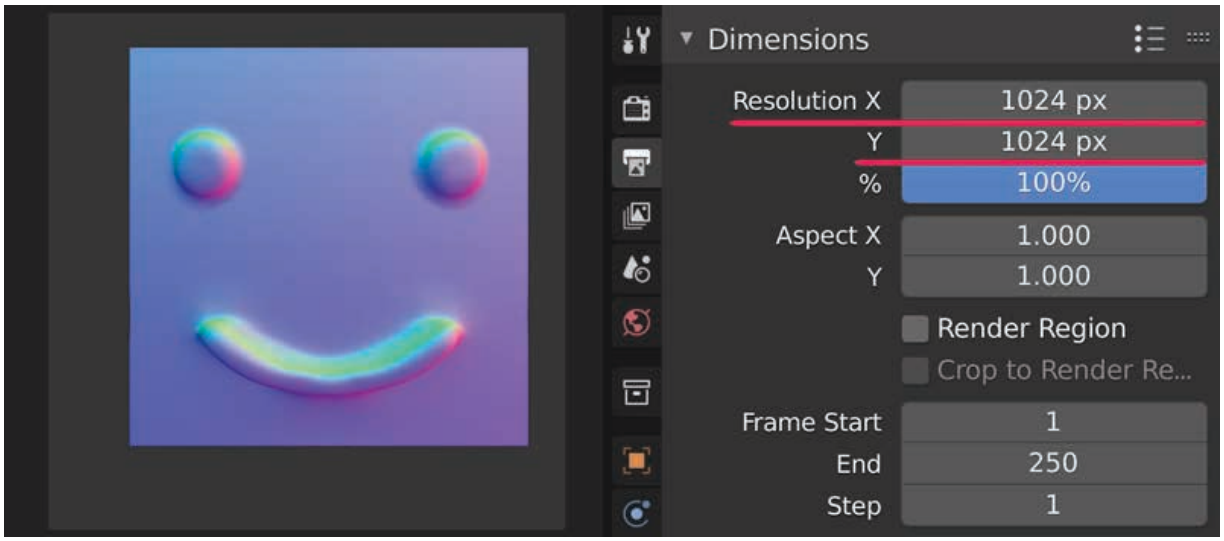


Go to the top view by **pressing 7 on the numpad** and **press Ctrl+Alt+0** to move the camera to match the top view.

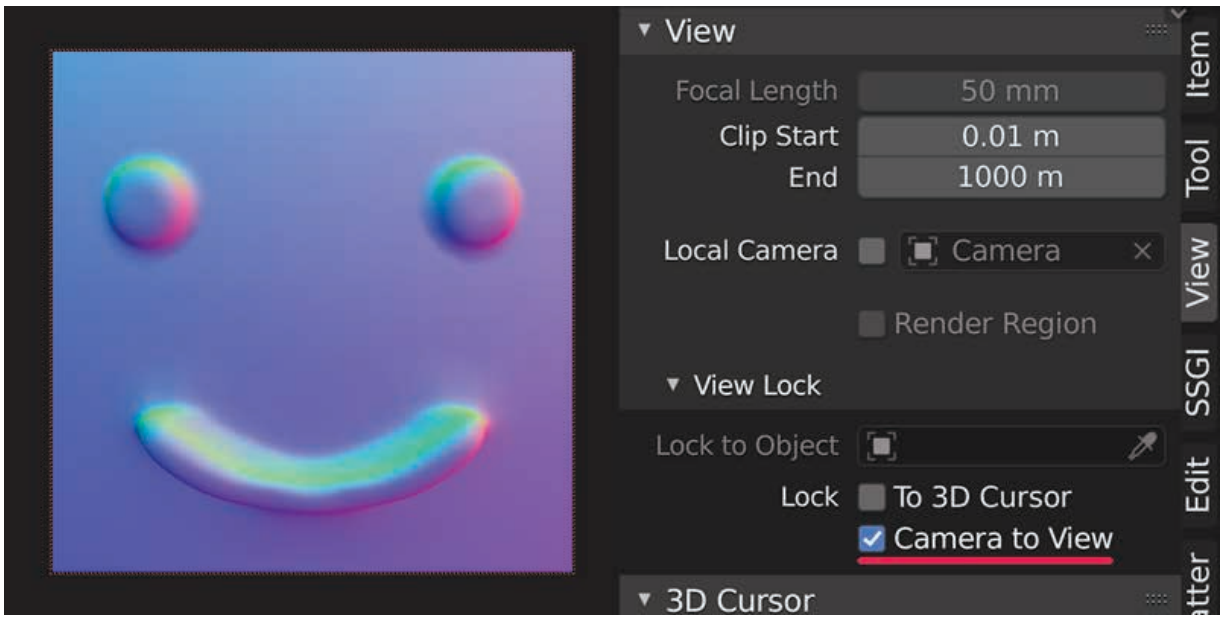


Click here to return to the index.

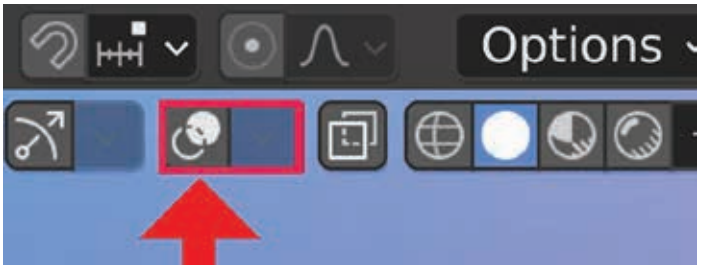
Set the resolution to 1024 x 1024.



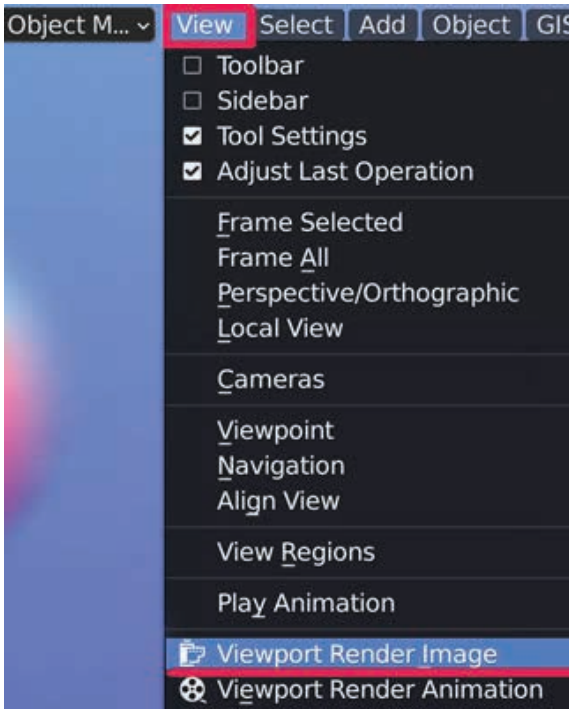
Lock **Camera To View** and move it so the plane fills the image.



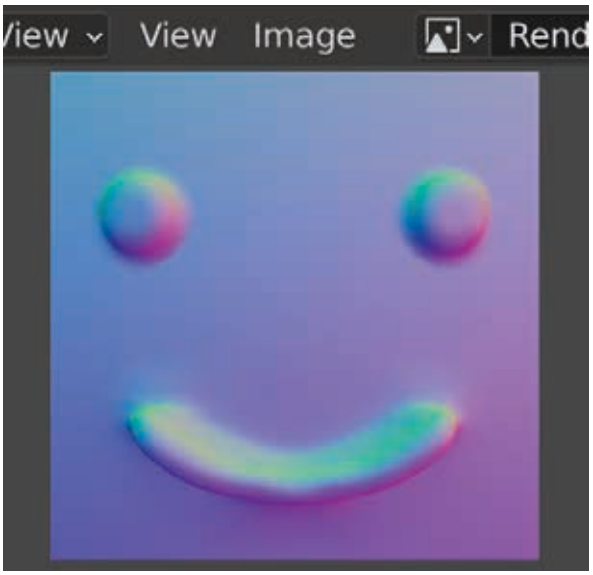
Turn off **Overlays**.



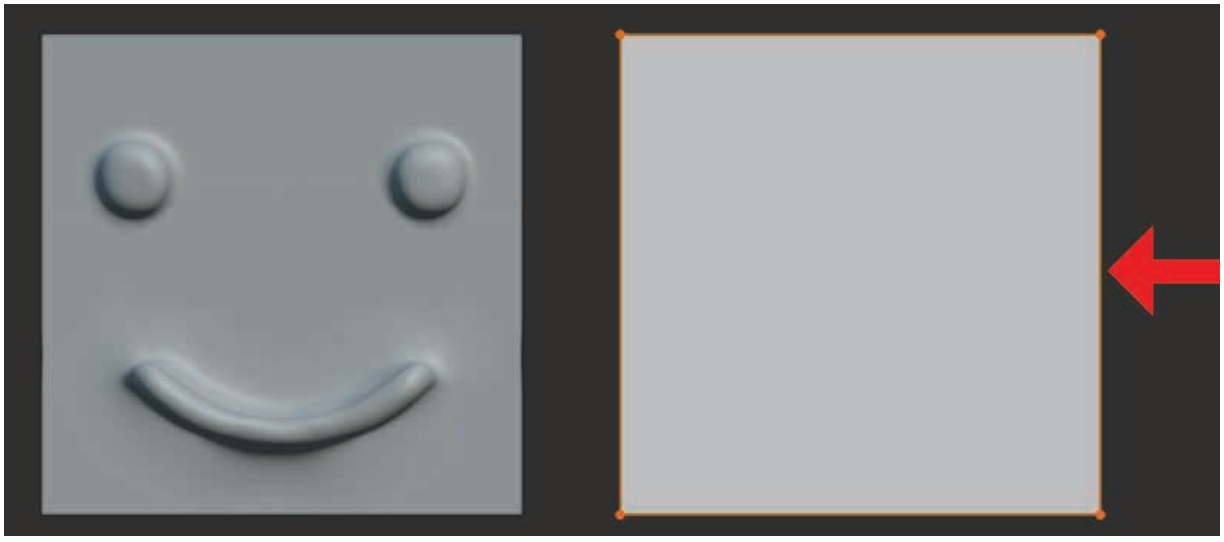
Now render the viewport with View > Viewport Render Image.



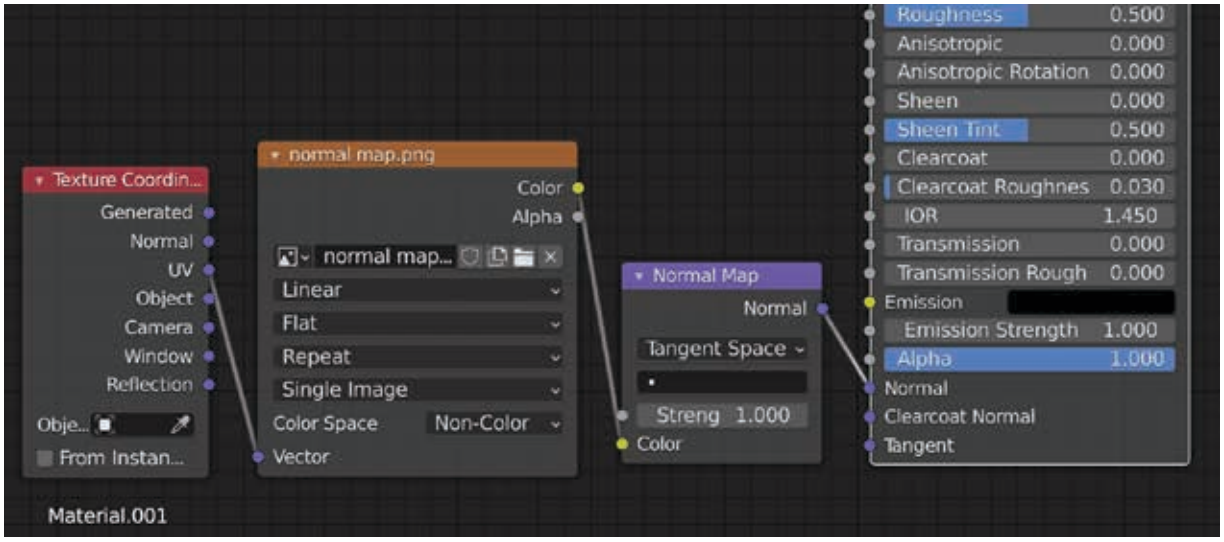
Now you've got a normal map!



Add a second plane with just one face. We'll use this one to test the Normal map we've created.



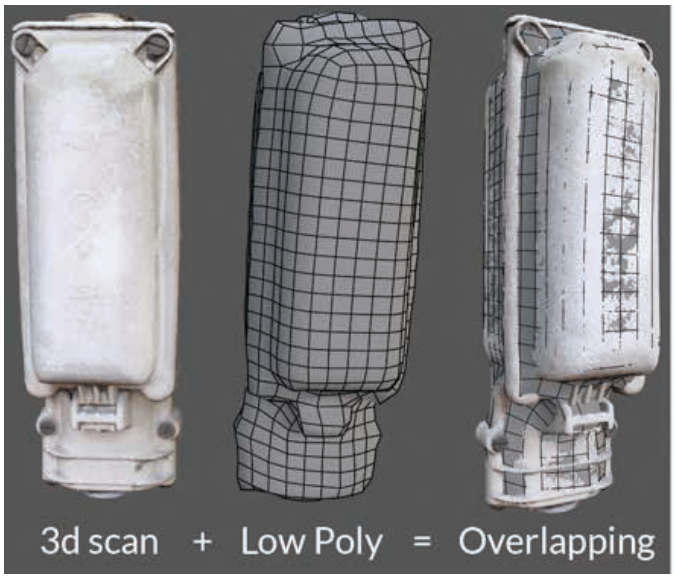
Add a material to it and plug in a Normal Map node and the image you saved. Set the image to **Non-Color Data**.

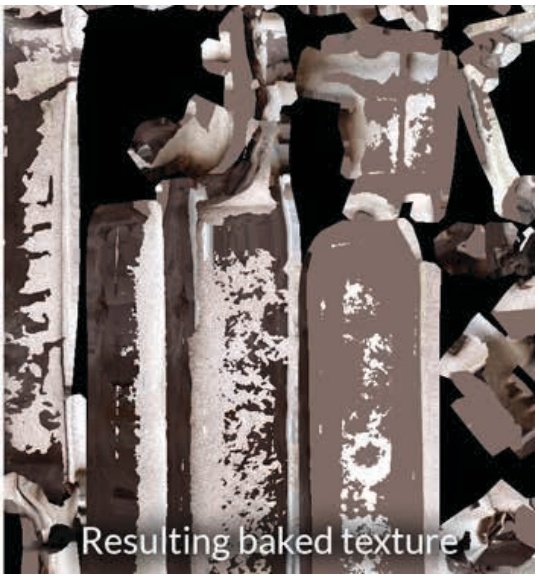


In Rendered view, we can see the Normal map working.

Baking with a Cage

When baking a colour texture from a 3D scan to a low poly model, the lack of geometry in the low poly mesh can cause some overlapping artefacts. For this topic, I'm assuming you know how to bake textures from 3D scans to low poly meshes as this is discussed previously. For more information about baking 3D scans to low poly objects, check this video.



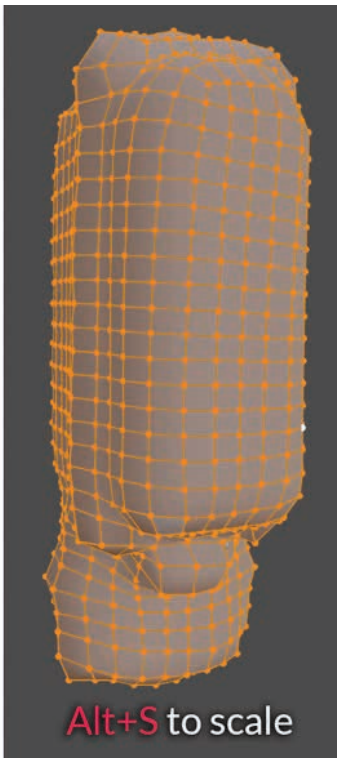


To solve this, we can use something known as a "cage".

It's a duplicate of the low poly object - it needs to have the same amount of Faces - that envelops the 3D scan.

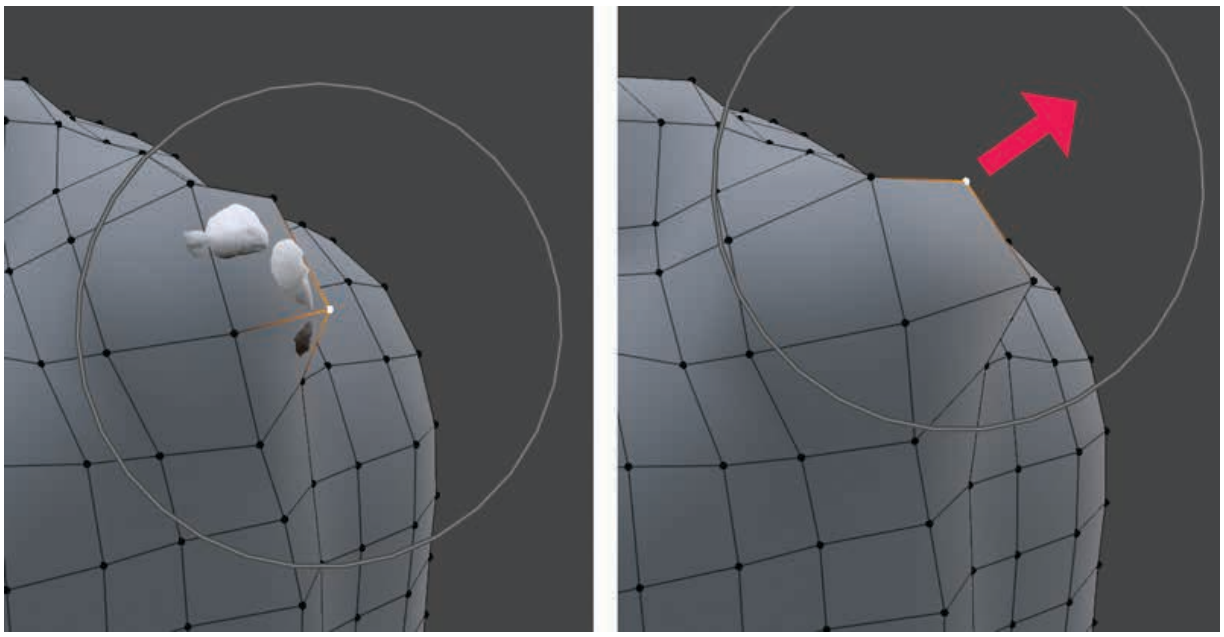
To make this, select the low poly mesh, **press Shift+D** to duplicate it and then **press F2** and rename it "Cage".

Now select all vertices of the cage in Edit Mode and **press Alt+S** and scale it so that it envelops the 3D scan.



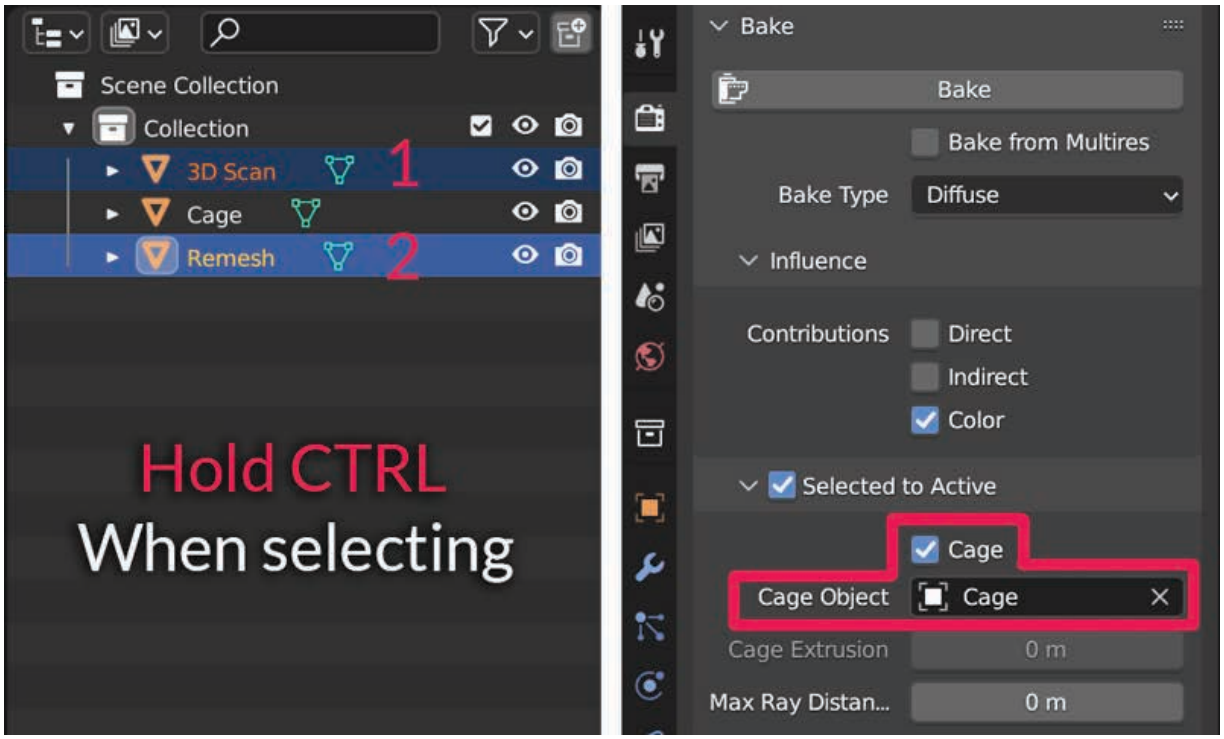
Alt+S scales the vertices along their normals, which "fattens" the mesh rather than conventionally scaling it. Which is perfect for creating a cage in a lot of cases.

However check if it really envelops the 3D scan everywhere, and if necessary do some manual tweaking by grabbing vertices and moving them using **Proportional Scaling** (toggle on/off by pressing o, and use the mouse wheel to change the radius of influence.)

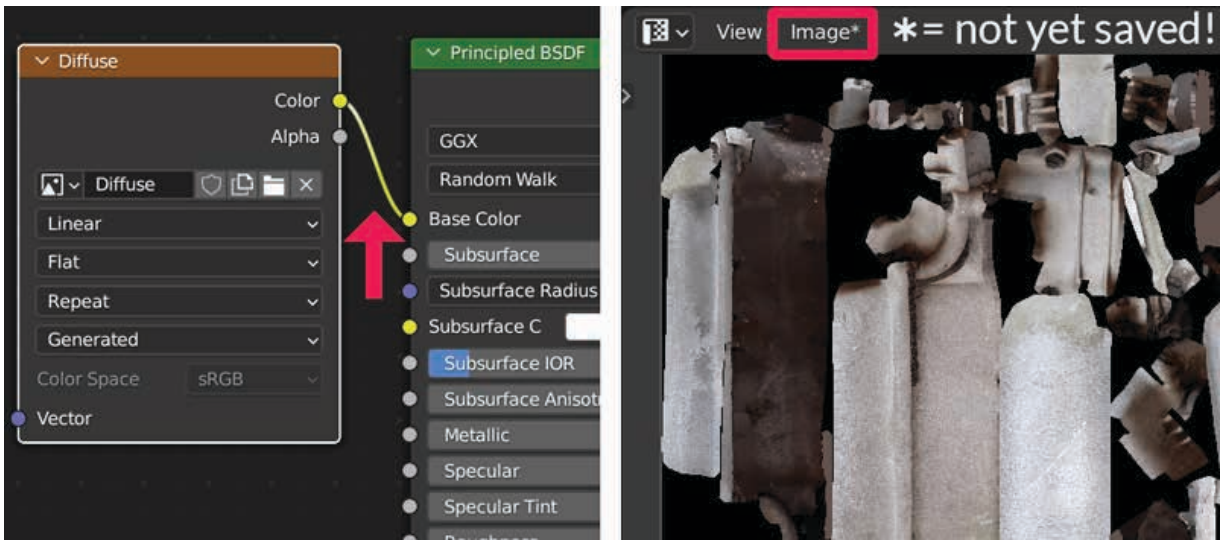


So having created the cafe from a duplicated low poly mesh, you now have 3 objects: the 3D scan, low poly mesh and cage. It might become a bit messy to select objects in the right order in the **3D viewport**, so we'll do that in the Outliner. It's really essential that you know this: when selecting in the 3D viewport, select the 3D scan, **hold Shift** and select the low poly mesh. But in the **Outliner**, select the 3D scan, **hold CTRL** and select the low poly mesh. This is a subtle difference that can lead to a lot of frustration.

Check the "**Cage**" option in the **Bake settings** and select the Cage mesh as the **Cage Object**. Now click **Bake**.



Now the texture bakes perfectly. **Don't forget to save it!** If you see an asterix (*) next to "Image", that means the changes are not saved. You can connect the image texture node used for baking to the Base Colour input of the Principled BSDF shader now.

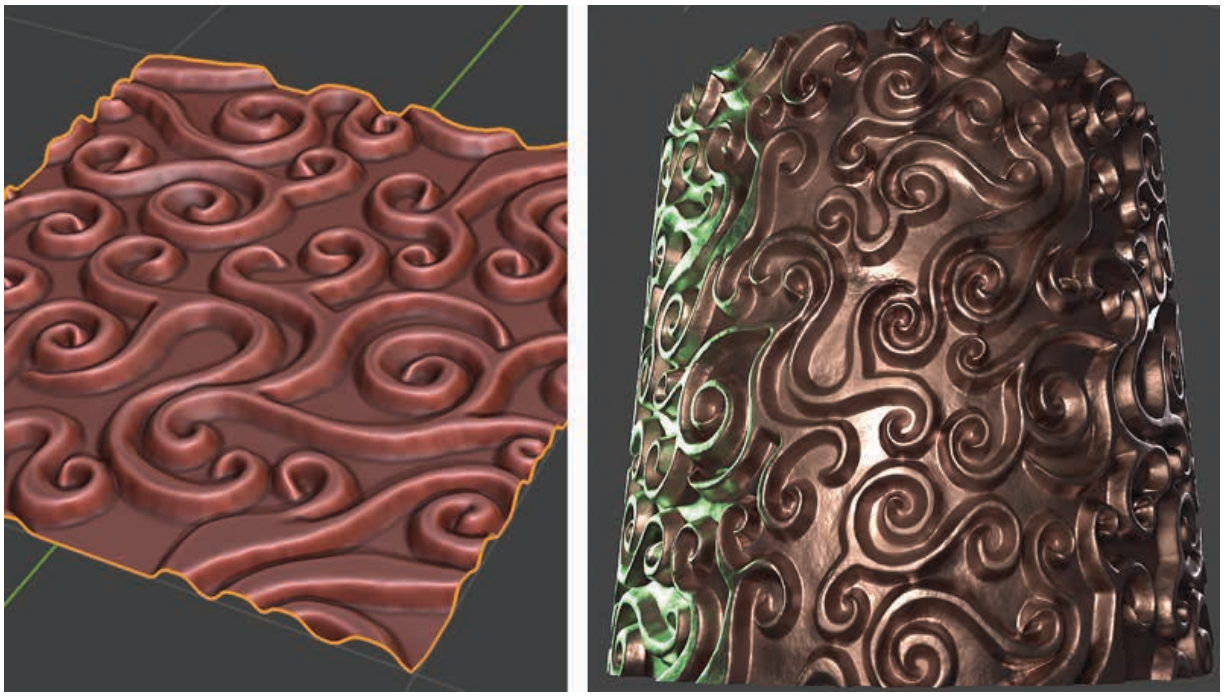


If it looks good, you can delete the cage. Here you see the result of the low poly mesh with the diffuse texture, and on the right with a displacement texture baked from a Multires modifier (more on that technique in this video).

*Alternatively, you could use a **Displacement modifier** instead of scaling the cage up with Alt+S. That way you have more non-destructive control over the amount of scaling. The end result is the same.*

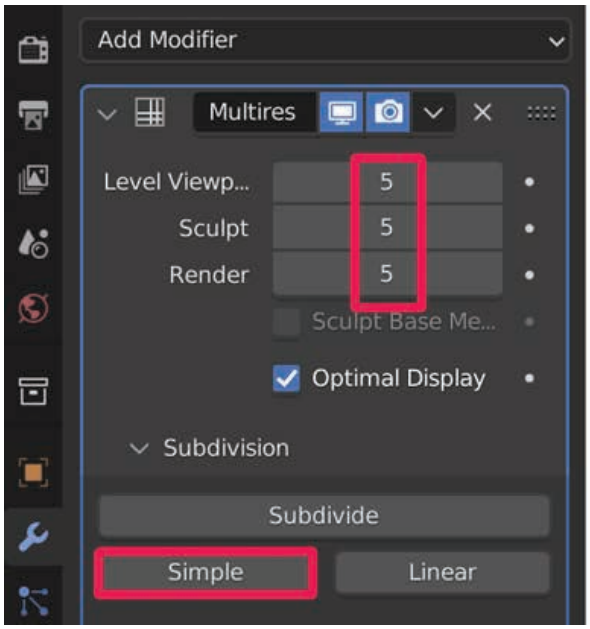
Baking Displacement from Camera View

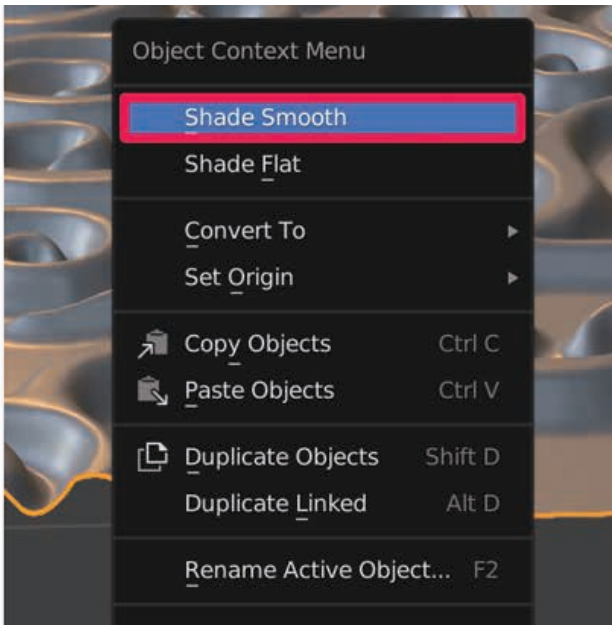
In another topic, we sculpted some seamlessly tiling geometry on a Plane. Now, let's bake this to a Displacement map. Then we can re-use it on any model.



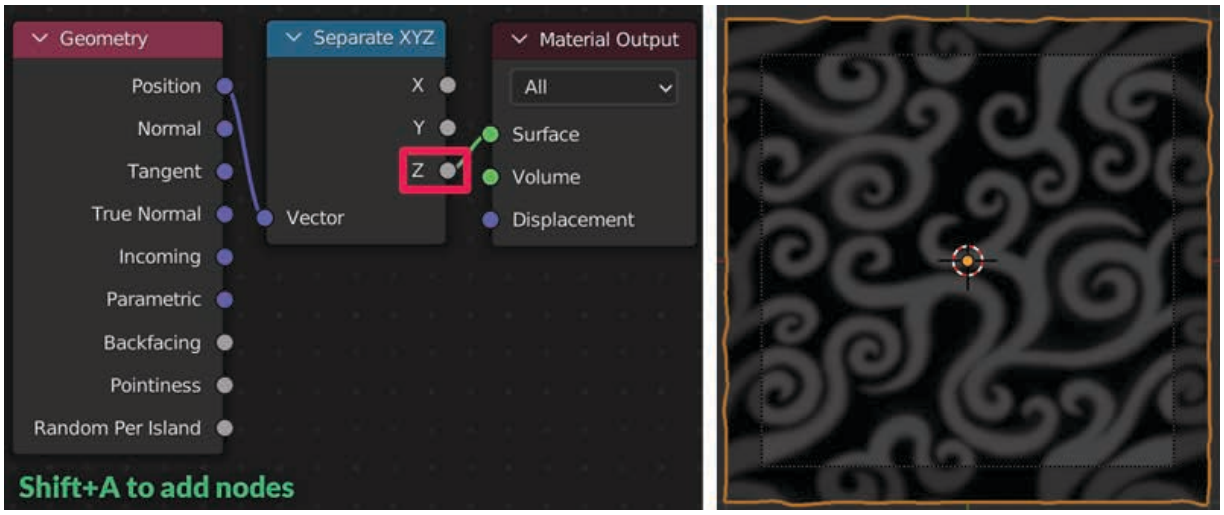
Above left: The sculpted Plane. Above right: The Displacement map in action, mixed with a subtle Noise Texture in a Bump map.

Make sure your **Multires modifier** is at the highest Level. Since you're finished with sculpting, you can try an even higher level by clicking on Simple to subdivide to level 5, for example. Also, make sure your sculpt is set to **Shade Smooth** in Object Mode.

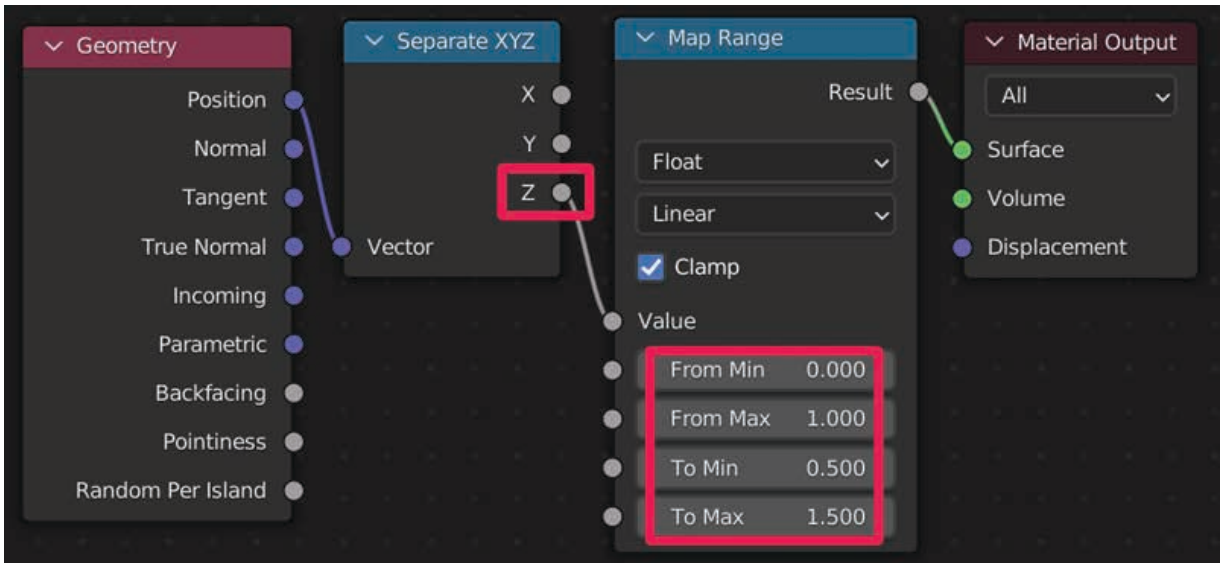




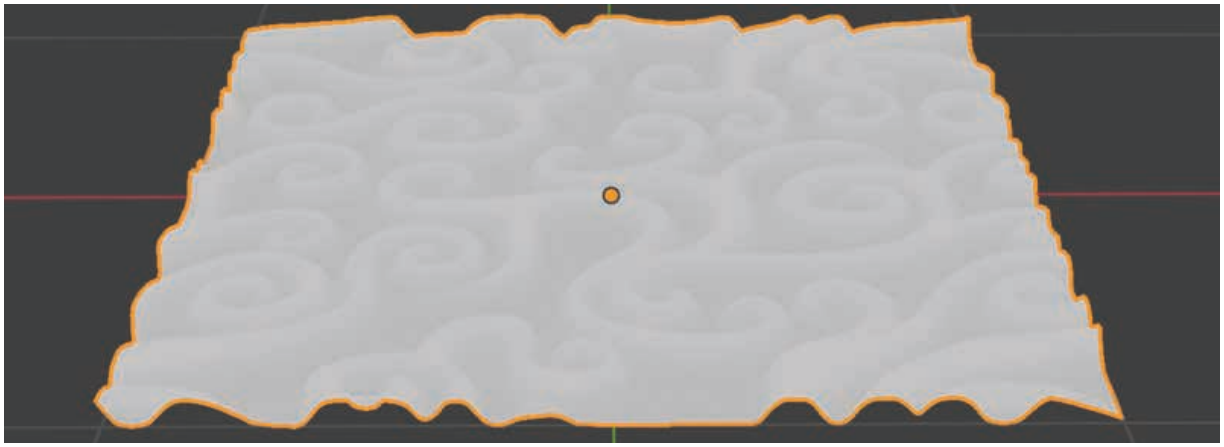
Add a material to the object, select and delete the **Principled BSDF** in the Shader Editor and add **Geometry** and **Separate XYZ** nodes. Use the Separate XYZ node's **Z output**. As you can see, this makes the low parts dark grey and the high parts light grey.



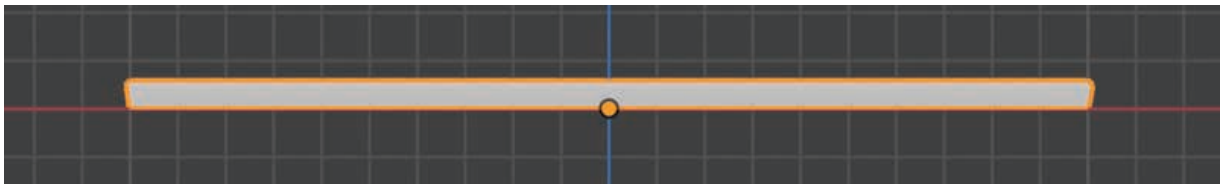
However if we use this as a Displacement Map, it will require us to change the values in the Displacement node each time we want to use it. Instead, let's make it perfect. Add a **Map Range** node. The Separate XYZ node's Z-output goes into the Value input of the Map Range node. This allows us to remap the **0 to 1** values to **0,5 to 1,5**.



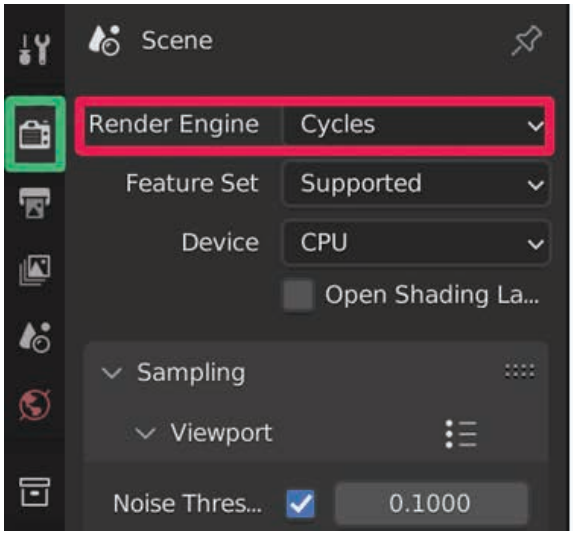
Now the result is more light grey.

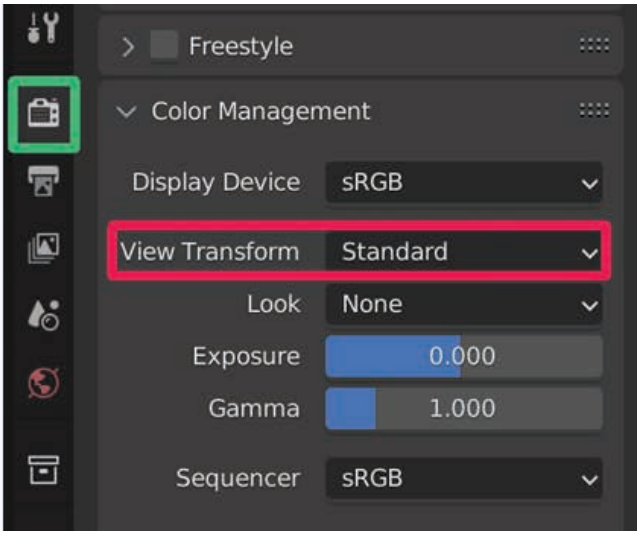


Make sure the object rests on or just above the floor of the world.



Set the Render Engine to **Cycles** and under **Color Management**, set **View Transform** to **Standard**.

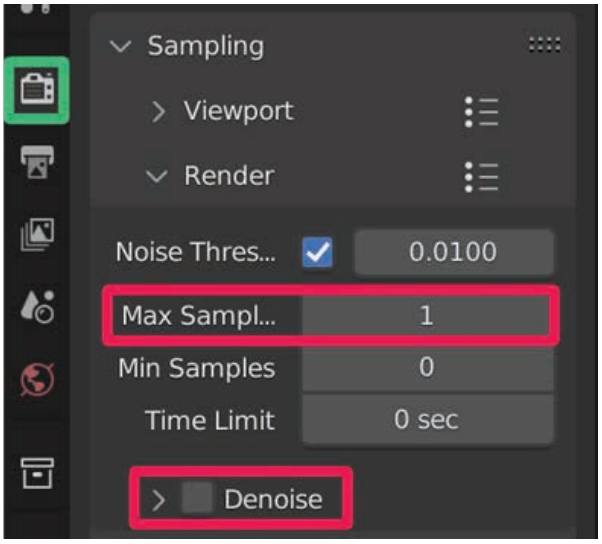




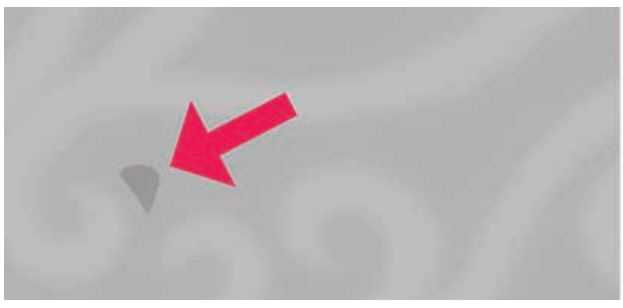
Press F12 to render the camera view (see the topic about sculpting seamless tiles for how to set up the Orthographic camera).

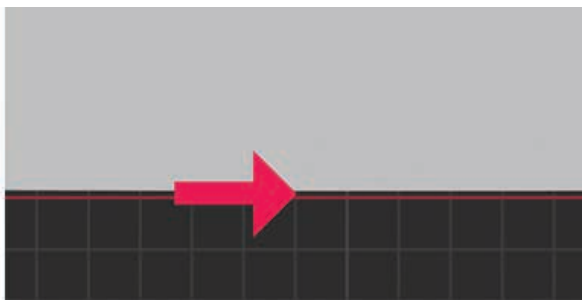
You don't have to have a lot of samples to render a Displacement map, so you can save some time by reducing the **samples to 1**.

It makes no difference in the quality of the render of the Displacement map. You can also **turn of Denoise** to save even more time.

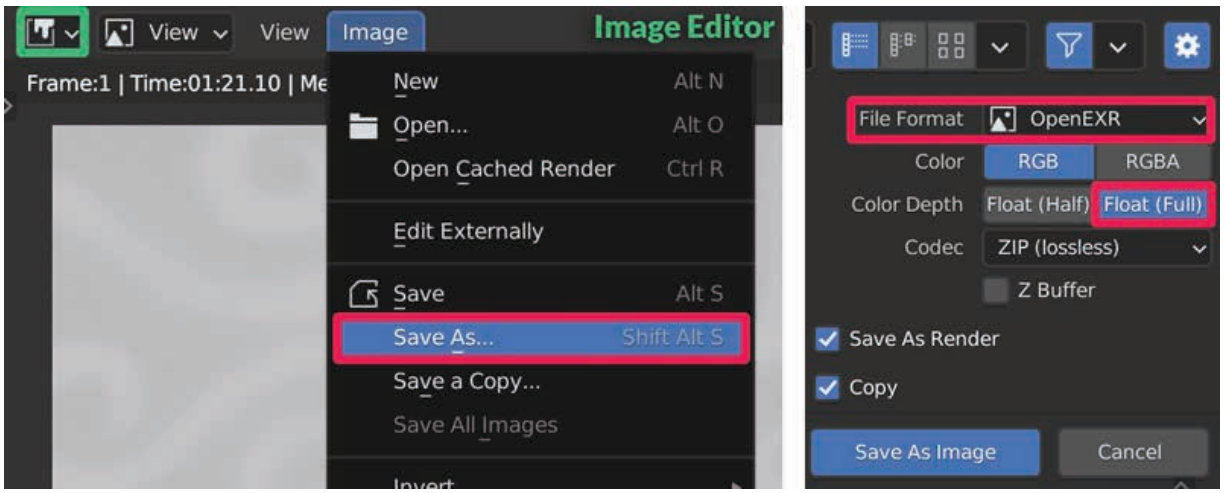


In case you have some weird spots on the resulting rendered Displacement map, that means some parts of the model are slightly below the world floor. So move the object up until it's on or just above the floor, and render again.





Once the result is rendered without artefacts, save it as a **Full Float (32-bits) OpenEXR** image. That way it's not compressed and has the most greyscale values.



Now you can load this as a **Displacement map** on other models, either in a Displacement node in a material, or with a Displacement modifier (for example if you're using Eevee, which doesn't have a Displacement node).

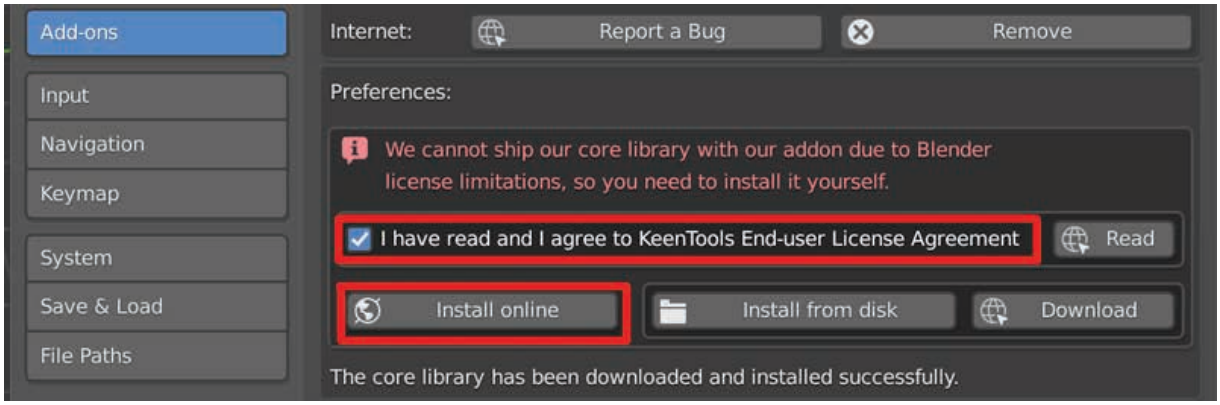
CHARACTERS

Modeling from Photos - Face Builder

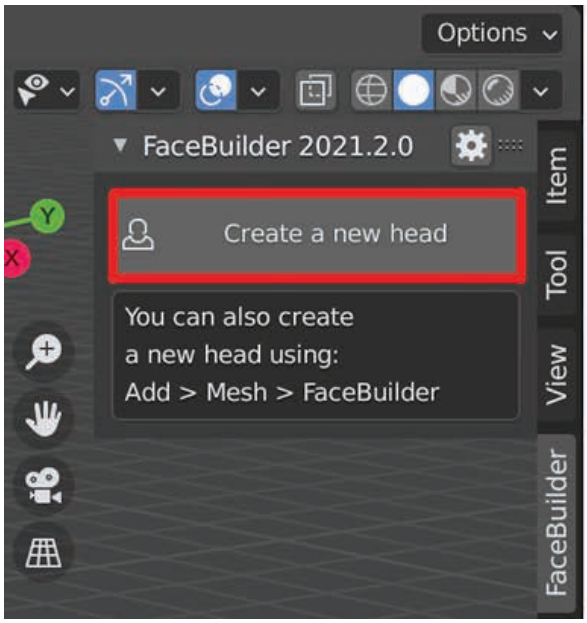
If you ever need to reconstruct a person's head but don't have enough photos or video to use a photogrammetry workflow, Face Builder is a very nice alternative. It's a paid add-on, however the demo version is fully functional and works for 15 days, so you can definitely get a lot of work done with it for free.

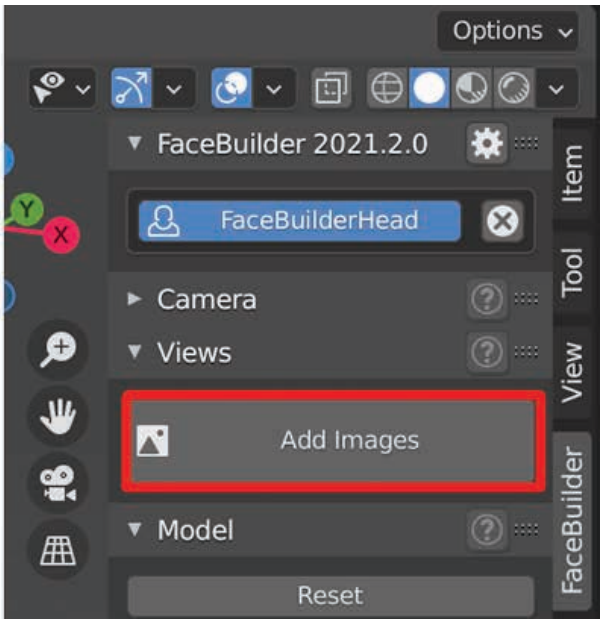
All you need is 2 or 3 photos. More is better, but really not necessary for a decent result. In this example, I used 3 photos of my own head.

First of all of course, you'll need to download the add-on. You can do so from this page: https://keentools.io/download/facebuilder-for-blender Installation is a bit different from most add-ons, in that once you've installed the add-on from Blender preferences, you need to open it's options in Preferences, check the End-user Licence Agreement box and install a "core" from there.

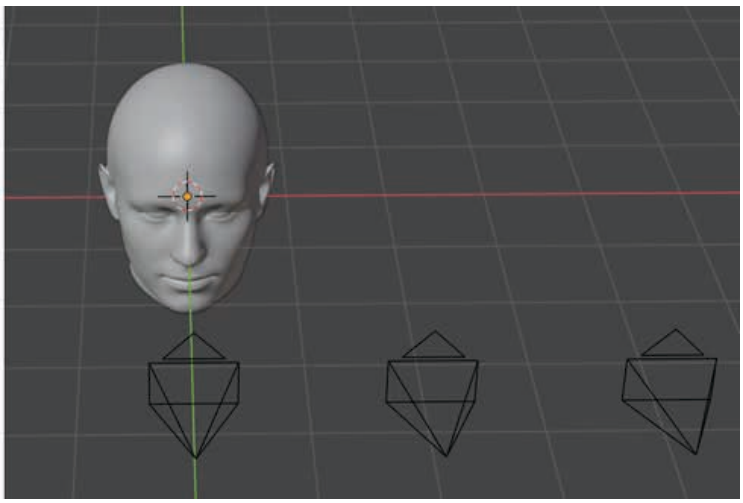


Once the installation of the add-on and core are complete, open the Option panel by pressing N and open the Face Builder tab. There's really only one option: **Build a new head**! This creates a generic head model. Next, **click on Add Images**.

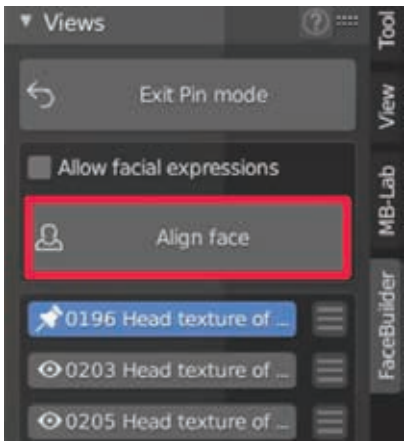




After adding 3 photos, we get a button for each photo which we can click on. Each photo gets its own camera in the 3D viewport which has that photo set as its background. For now, the camera's are just placed in a row.



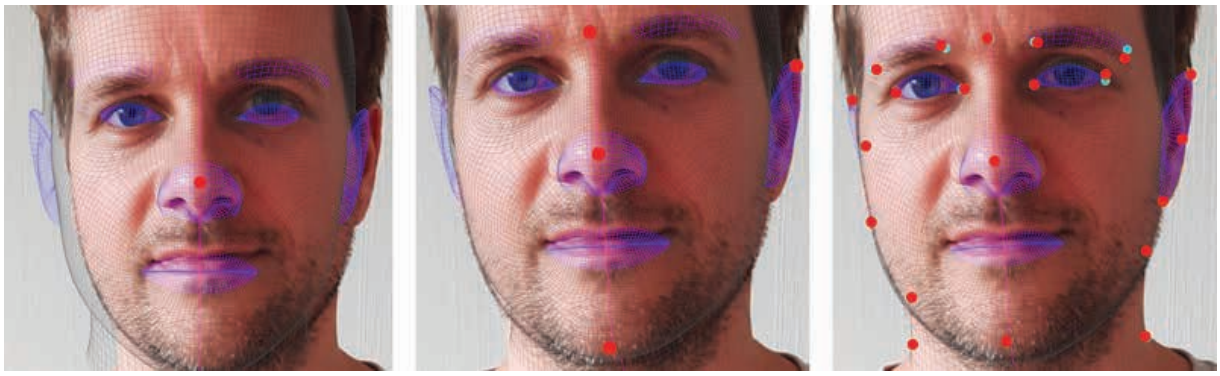
Click on one of the photo buttons and click on **Align Face**.



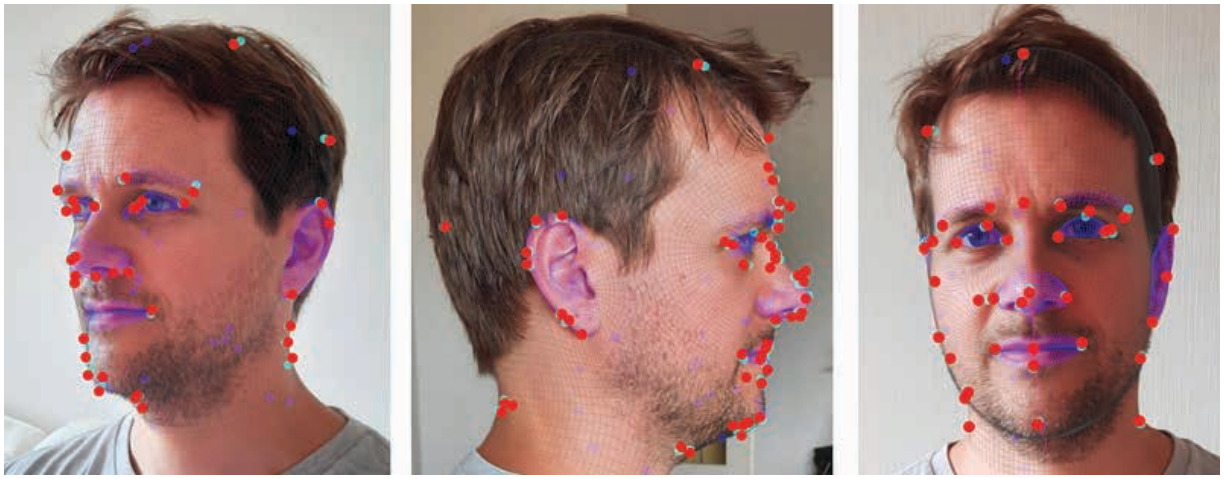
Now wait a bit for Blender to calculate, and after a while the head has been aligned to the photo.

Repeat this for the other photos.

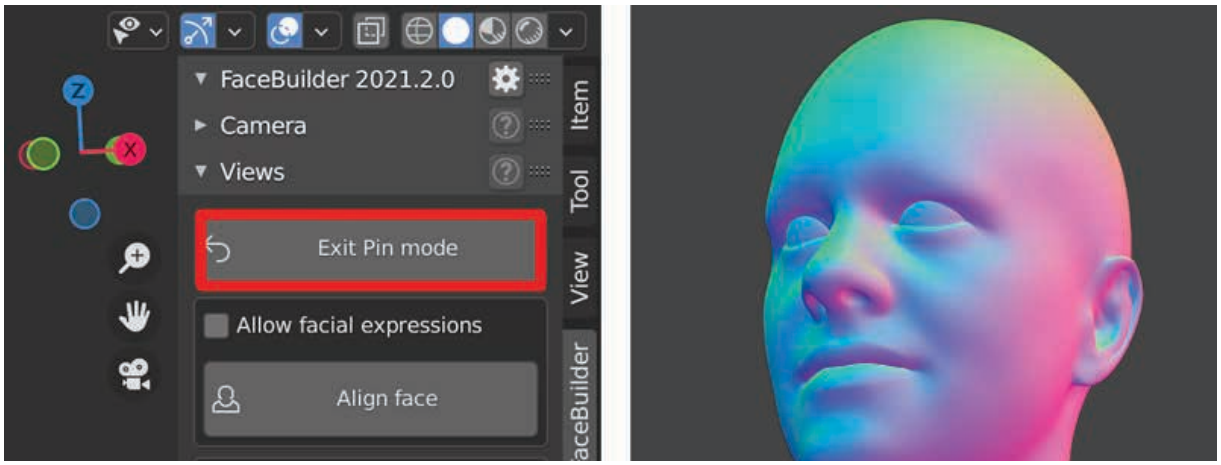
Now we can fine-tune it manually. This is done by clicking on a point on the model, and dragging it to the corresponding point in the photo. The more points you add manually the more the shape will match the photos. The order in which you do this, doesn't matter. To remove a pin, right-click on it.



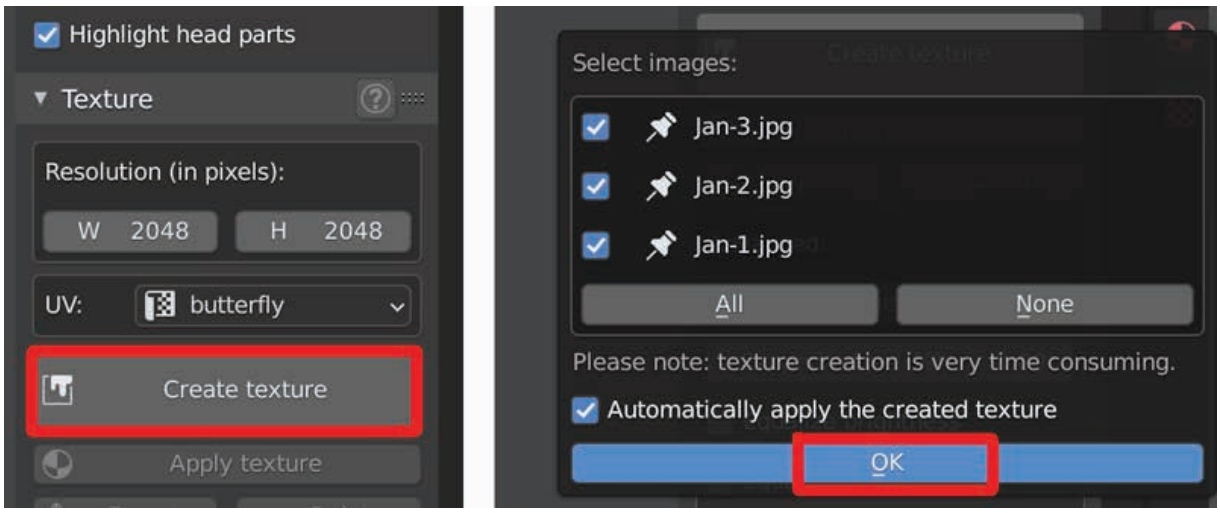
After this is done for one photo, move on to the next one. When they're all done, you may find that the model no longer matches perfectly in the first photo you worked on, because of changes you've made in the other photos. In that case, simply keep adjusting each photo - at some point, with enough points added, it will be good enough.



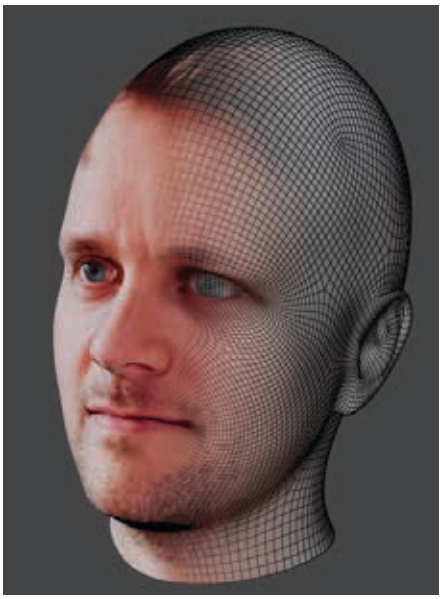
After you're satisfied with the pinning job, click on **Exit Pin mode**. Now you can see what the head model looks like.



To add the texture to the model, under Texture in the FaceBuilder tab click on **Create texture**. Then **click OK**. You can also limit which photos you use, in case some of them have very different lighting conditions.



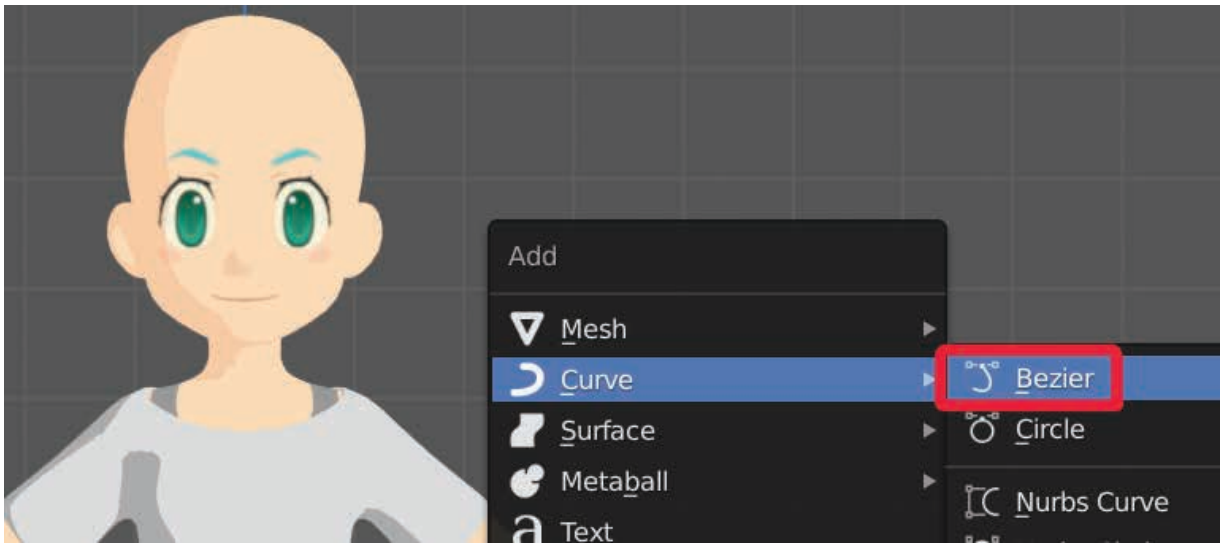
As long as the photos don't have too much baked in light, you should get a reasonably good result.



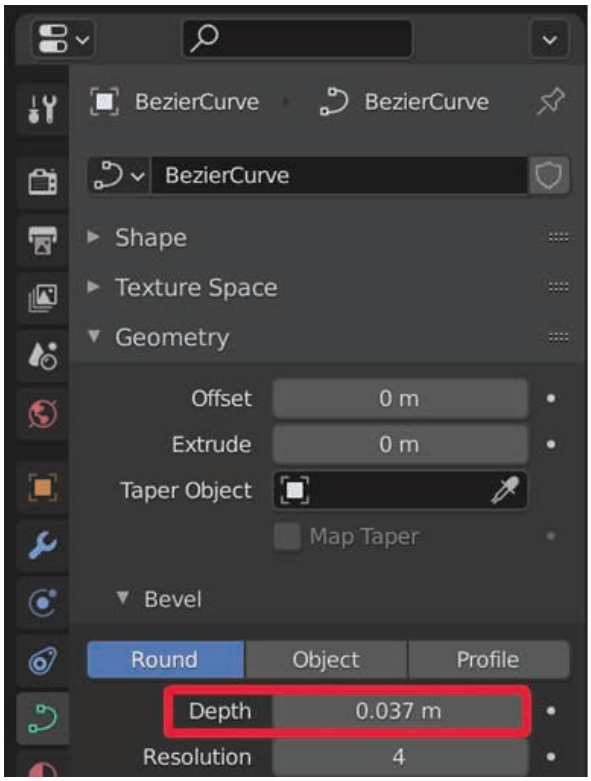
Easy Anime Hair

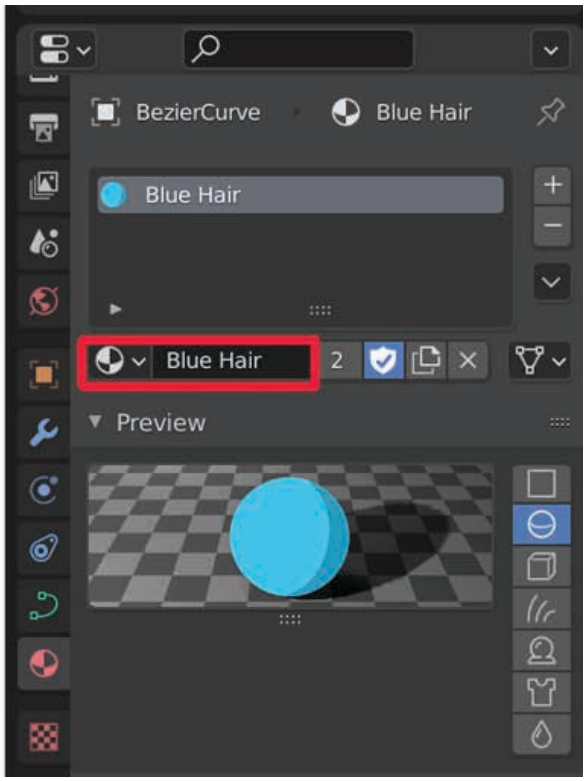
Creating stylized anime hair can be quite challenging it you do it manually. Luckily, there's a workflow that makes it fast and easy, using Curves drawn on surfaces.

Load your bald character, then **press Shift+A** and add a **Bezier Curve**.

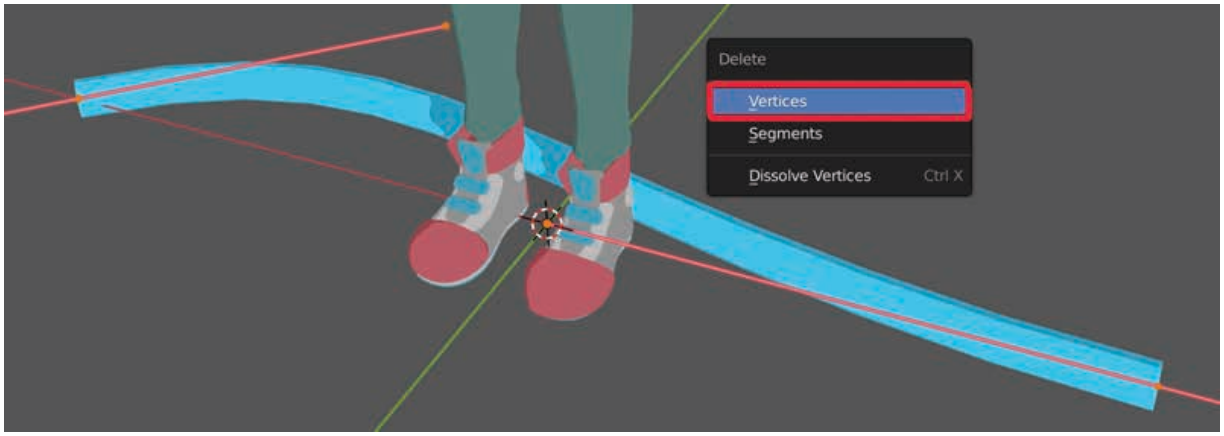


Give it a material (like this basic toon shader) and increase the **Bevel Depth**. Don't worry about the exact thickness this gives the Curve, you can change it later.

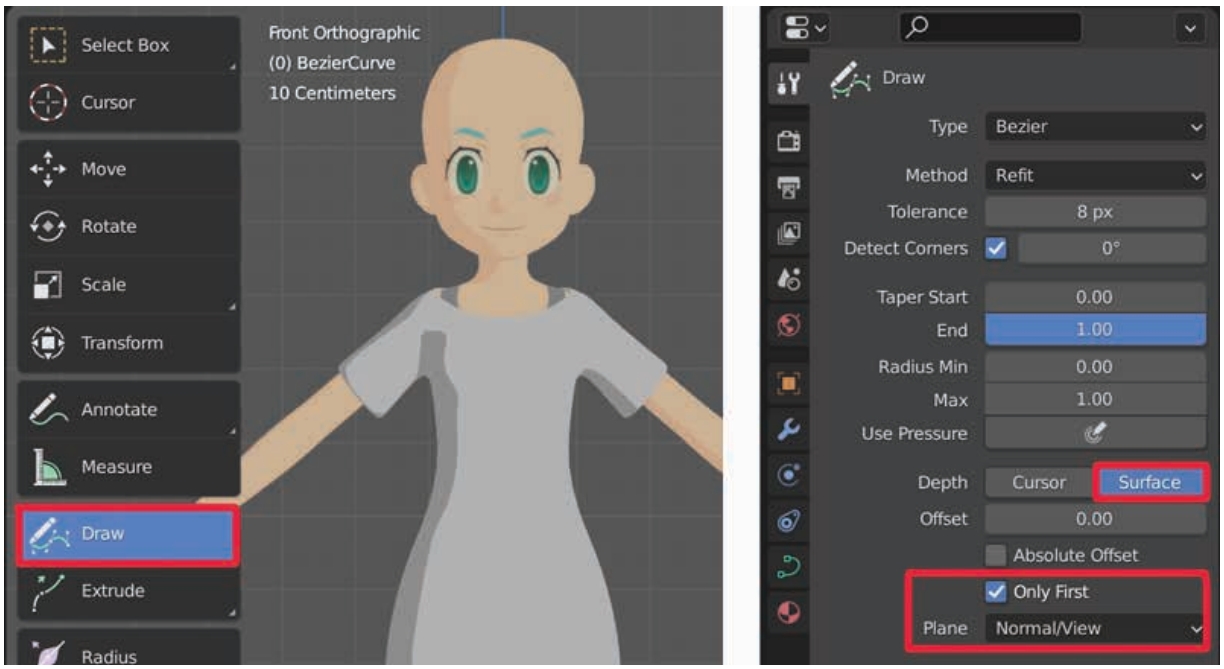




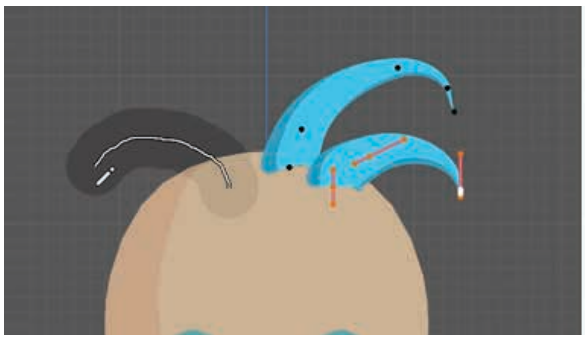
In Edit Mode, **select all and press X**, choose **Delete Vertices**.

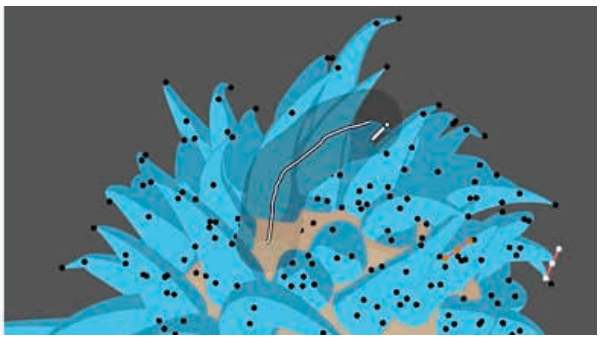


While still in Edit Mode, activate the **Draw Tool** and set Projection Depth to **Surface**. Set **Taper End to 1** so the hair tapers to a point. Check **Only First** and set it to **Normal/View**, so the hair is drawn in the direction of the Normals.

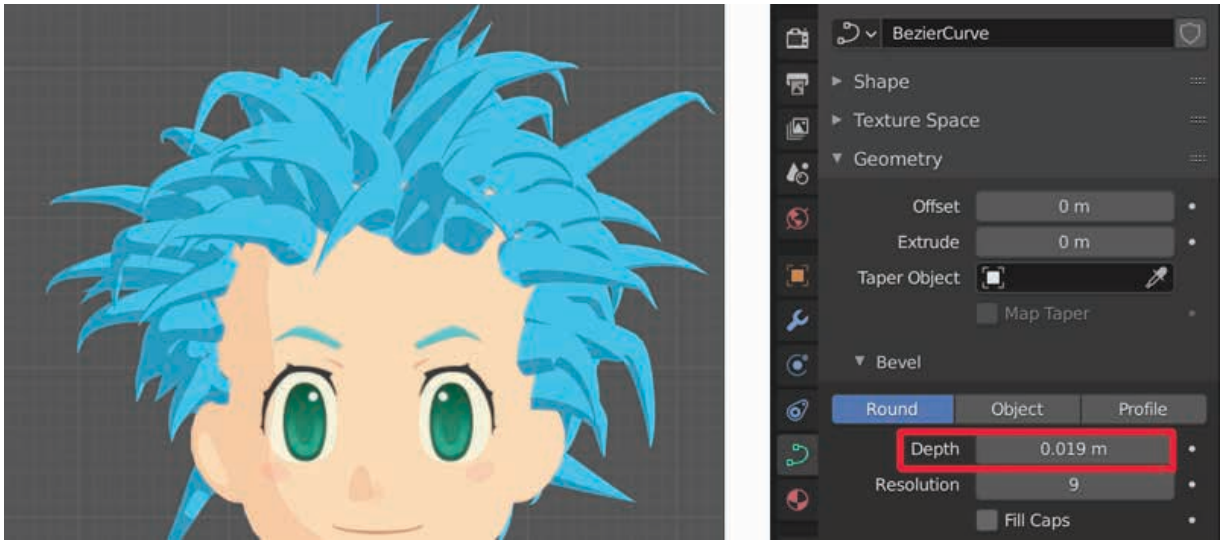


Now you can start drawing hair strands on the character. Don't worry about getting it perfect: if you mess up a curve, just **press Ctrl+Z** and try again. You can still adjust the strands later. Just fill that head full of hair first! Keep rotating the viewport so you can find all the bald spots.

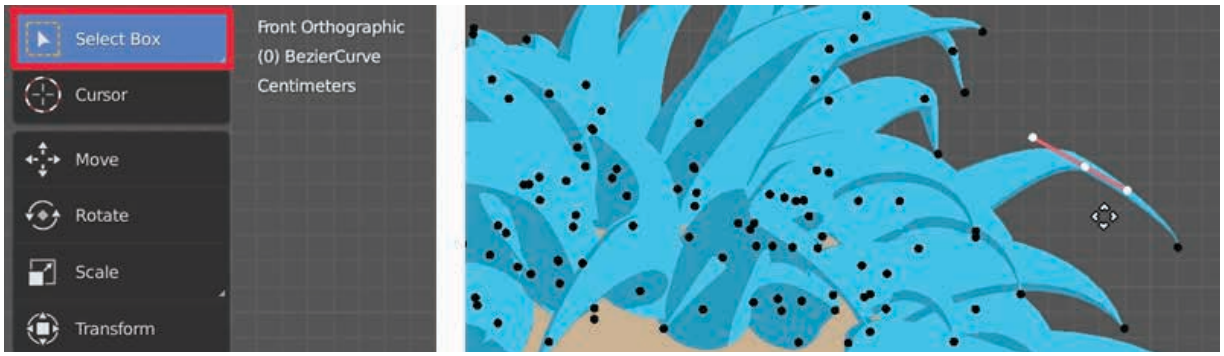




You can still adjust the **Bevel Depth** to change the hair thickness if necessary.



After you've got enough hair, you can rotate, move and scale the End points of the strands to make it really perfect.



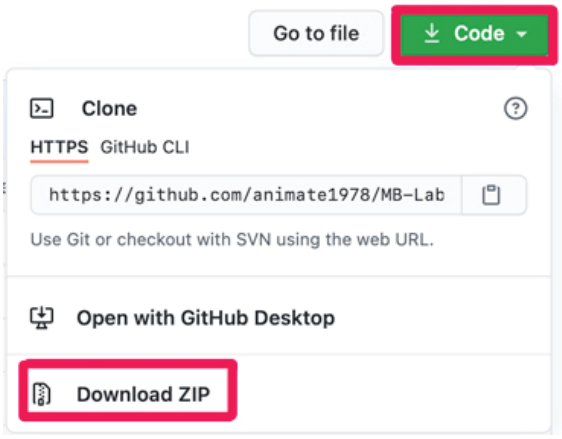
To change the shape of the strands, you can use a **Custom Profile**, although with this kind of Toon shader you probably won't see much difference.

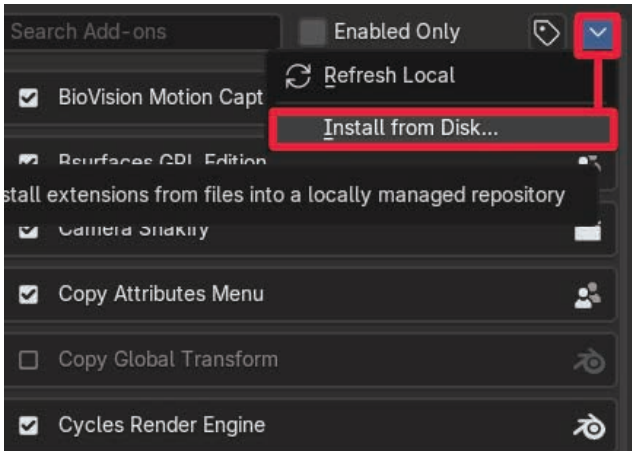
Create characters in just a few clicks in Blender

MB-Lab is a character creation tool for Blender that lets you create complex, realistic or stylized characters in just a couple of clicks, including clothing, hair and rigging!

Image by Manuel Bastioni, original author of the add-on (CC BY-SA 4.0)

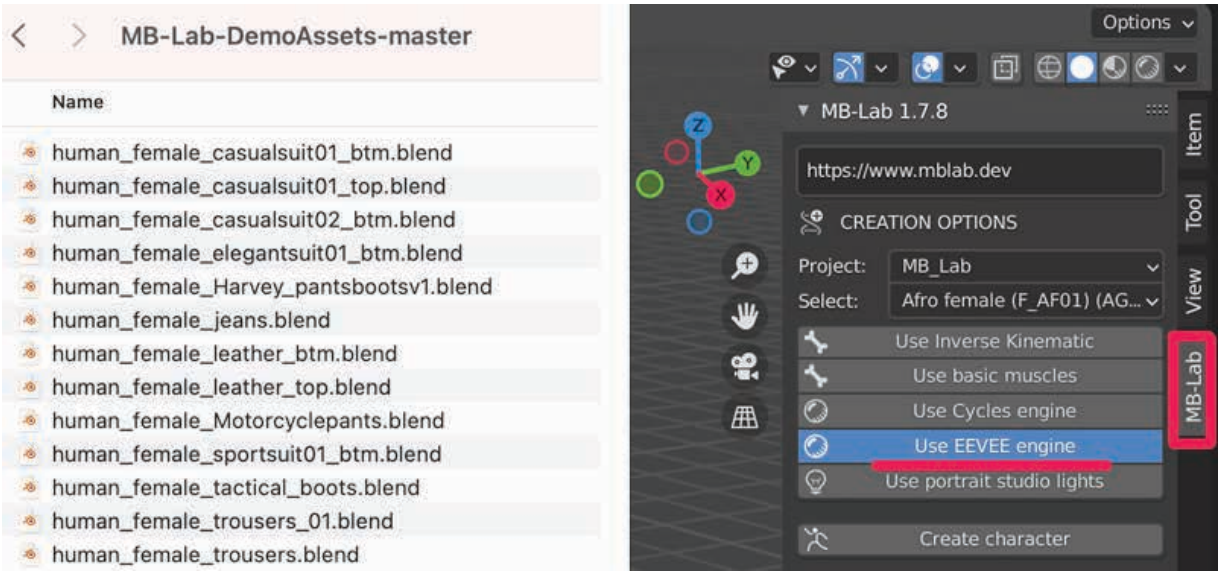
Get it from Github here: https://github.com/animate1978/MB-Lab. In Blender Preferences, install the ZIP file.



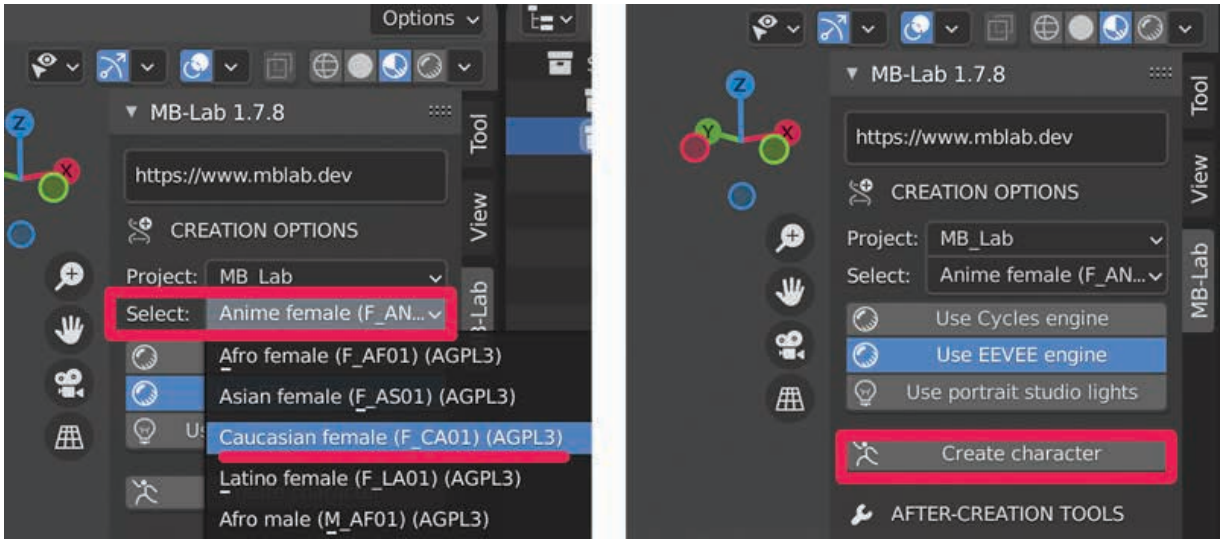


You can also download clothing assets from Github here: https://github.com/animate1978/MB-Lab-DemoAssets. In the case of the clothing assets you do have to unpack the ZIP file, and place the files somewhere easy to find.

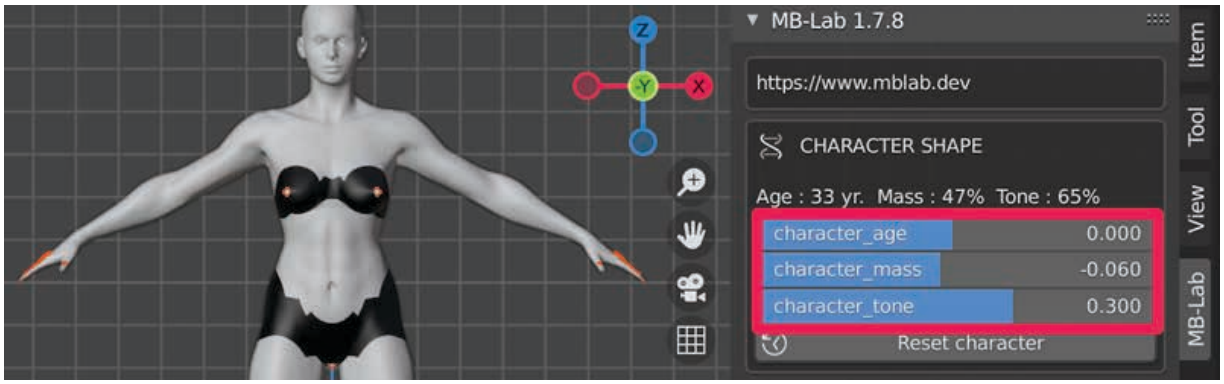
You'll find the add-on in the Option panel which you can open by **pressing N**. By default it's set to Cycles, but you can also set it to work with Eevee.



Now for the fun part! Select an initial character preset from the **Select** drop-down menu. In this example, I'm going with the "Caucasian female" option. Then, click on **Create character**.

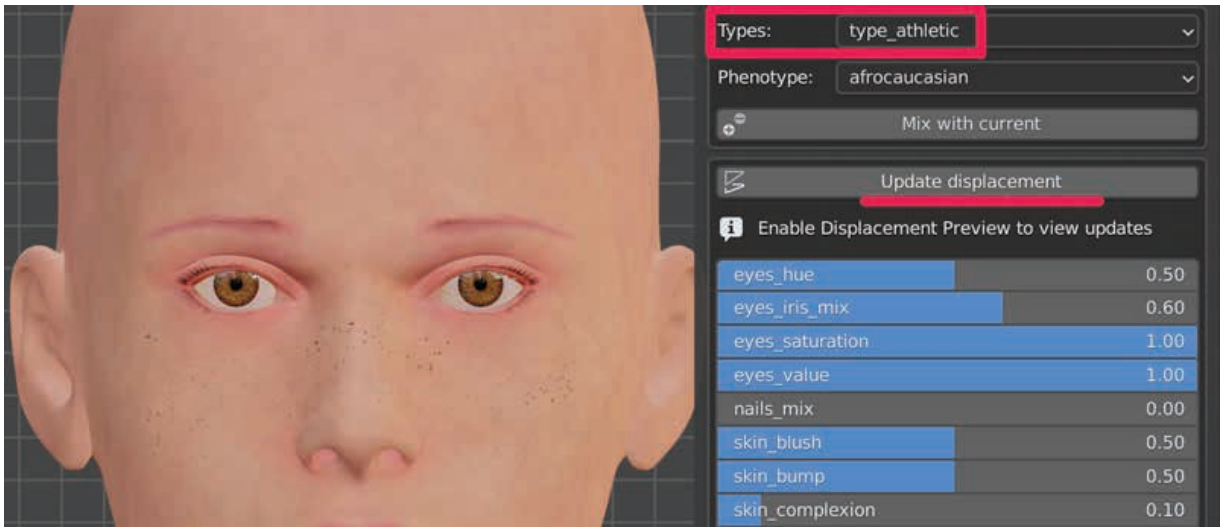


You can tweak the **age, mass and tone** values (tone is how muscular the character is). Switch to Material preview by **pressing Z > Material preview** to see what the skin and eyes looks like.



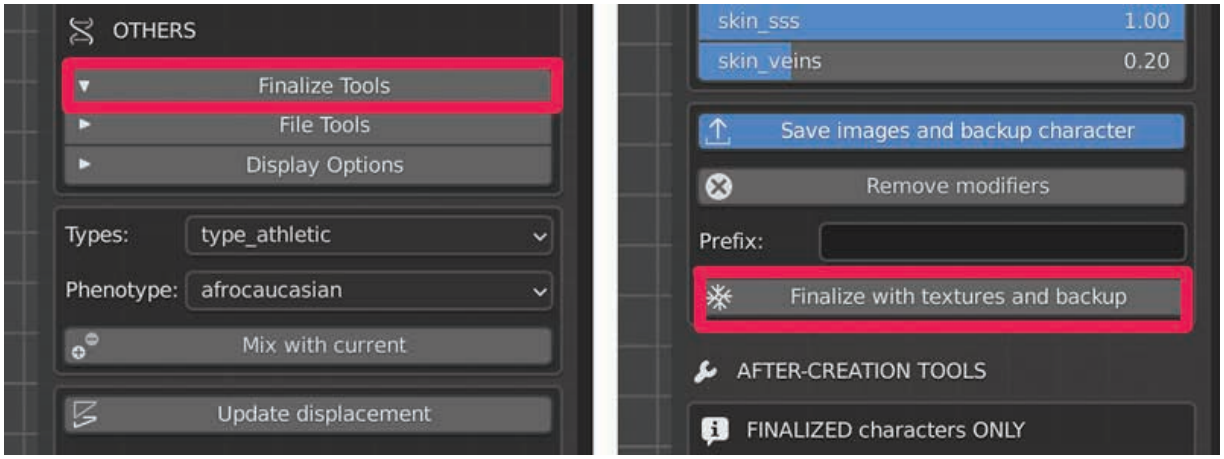
Instead of changing values manually, you can also try one of the presets under the **Types** dropdown menu.

Further down there are a lot of options for the skin and eyes that you can adjust with sliders. In case you change a value that influences displacement, such as **skin_bump**, you'll need to click on **Update displacement** to see the changes.



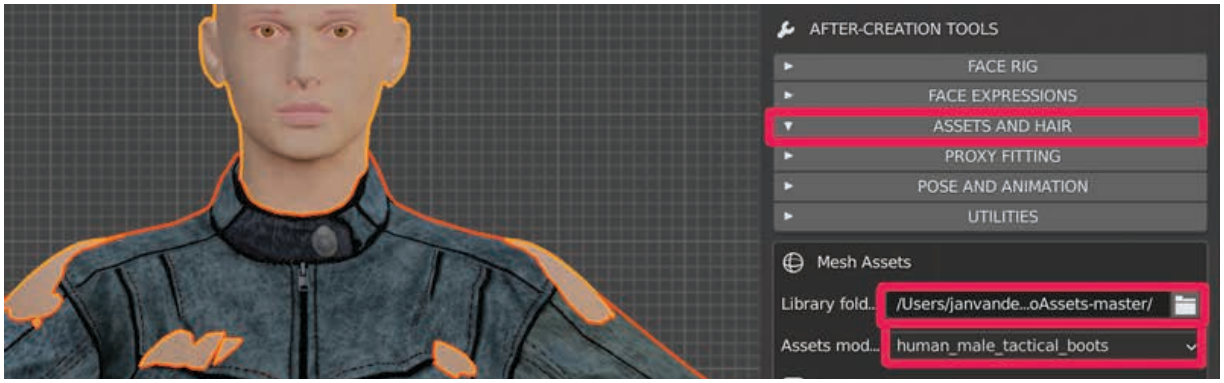
Once you're happy with the character, click on **Finalize Tools** and then on **Finalize with textures and backup**.

You'll be asked to save the texture somewhere, after which Blender will "bake" the character to a normal rigged character mesh. You can't undo this step, so don't click until you're certain!



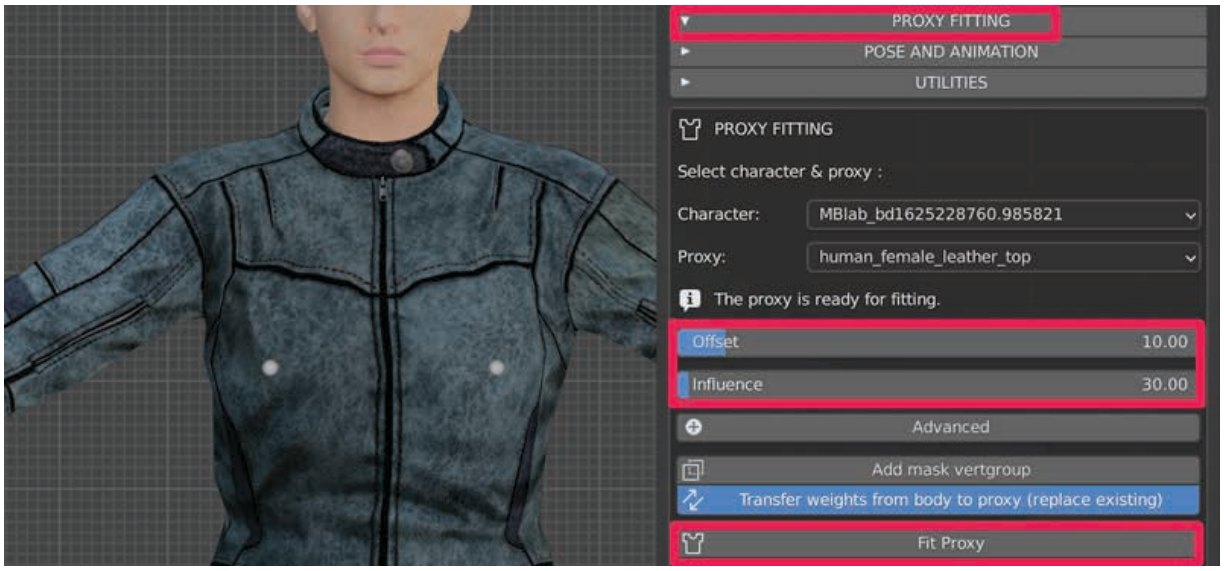
Now let's add some clothes. Click on **Assets and Hair** and point the **Library folder** field to the folder with the unzipped assets that you downloaded from Github earlier.

Choose the assets you want to use one by one from the **Assets model** drop-down menu.

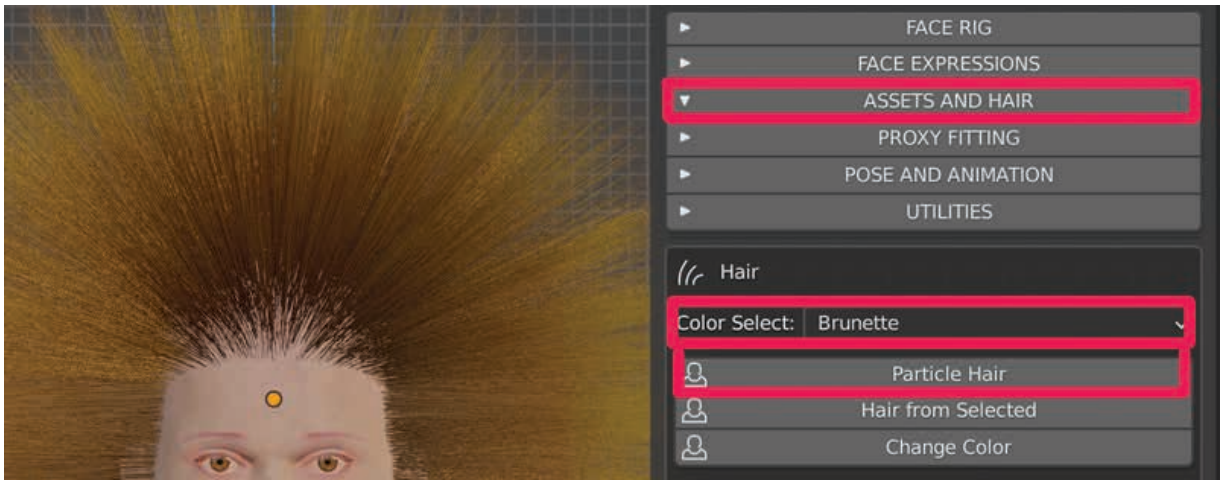


As you can see, the clothes don't fit by default. That's because we changed the shape of the body. To make them fit, open the **Proxy Fitting** options, select the clothing piece and click on **Fit Proxy**.

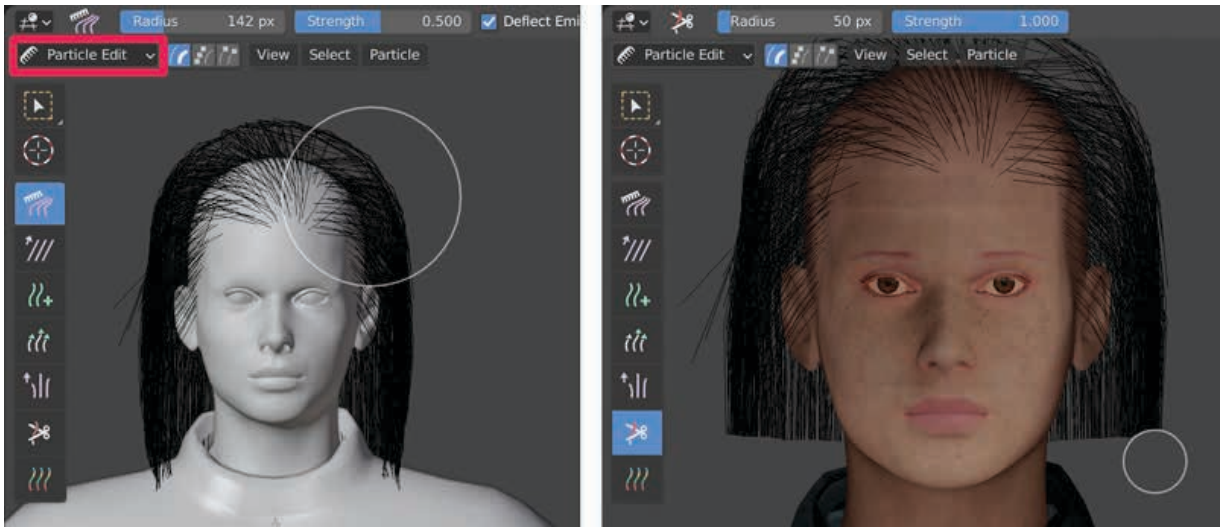
You may need to fine-tune it by adjusting the **Offset** and **Influence** values, and each time clicking on Fit Proxy to see the changes.



Finally let's add some hair! Click on the **Assets and Hair** options and choose a preset from the **Color Select** drop-down menu. With the character selected click on **Particle Hair**.



Now you can groom the hair with the standard Blender **Particle Edit** tools.



Although MB Lab can be quite useful, I'd like to recommend some other options as well:

There's the **Human Generator Add-on** for Blender, which is a paid tool. I haven't used it myself, but I've heard good things about it, and it looks a bit more user friendly than MB Lab, as it has for example pre-set hairstyles that you can use, so you don't have to do the particle editing yourself. https://blendermarket.com/products/humgen3d

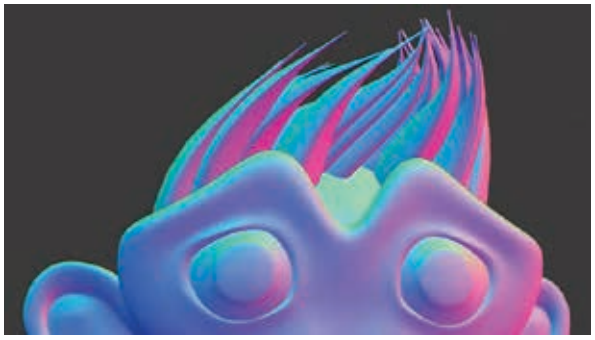
Epic has the free **MetaHuman Creator**, which is very easy and fun to use. You do need to register for an account, and install all kinds of stuff like the Epic launcher and to export the files as FBX, you'll need to install Quixel mixer. The results do look very detailed and good, so worth having a look at. https://www.unrealengine.com/en-US/digital-humans

Finally if all you need is a fairly basic model, perhaps as a base for sculpting or just some quick previsualisation animation, you can download any of the free **Mixamo** characters, which come fully clothed and textured. You do need to register an account, but it's free. https://www.mixamo.com/

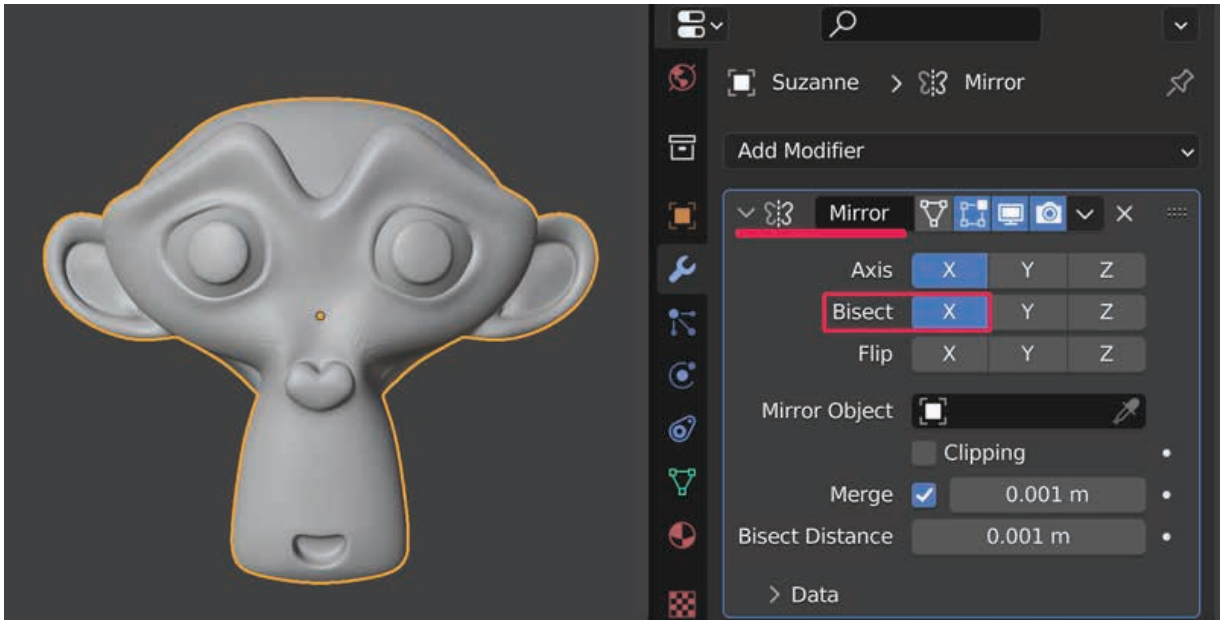
How to Model Stylized Hair

Knowing how to model stylized hair is an important step in creating a character.

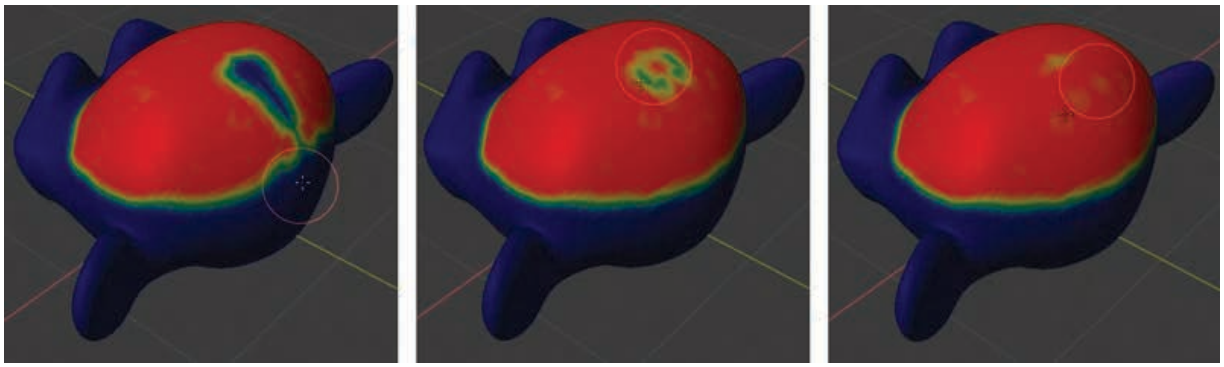
This is one quick way to do it using Curves.



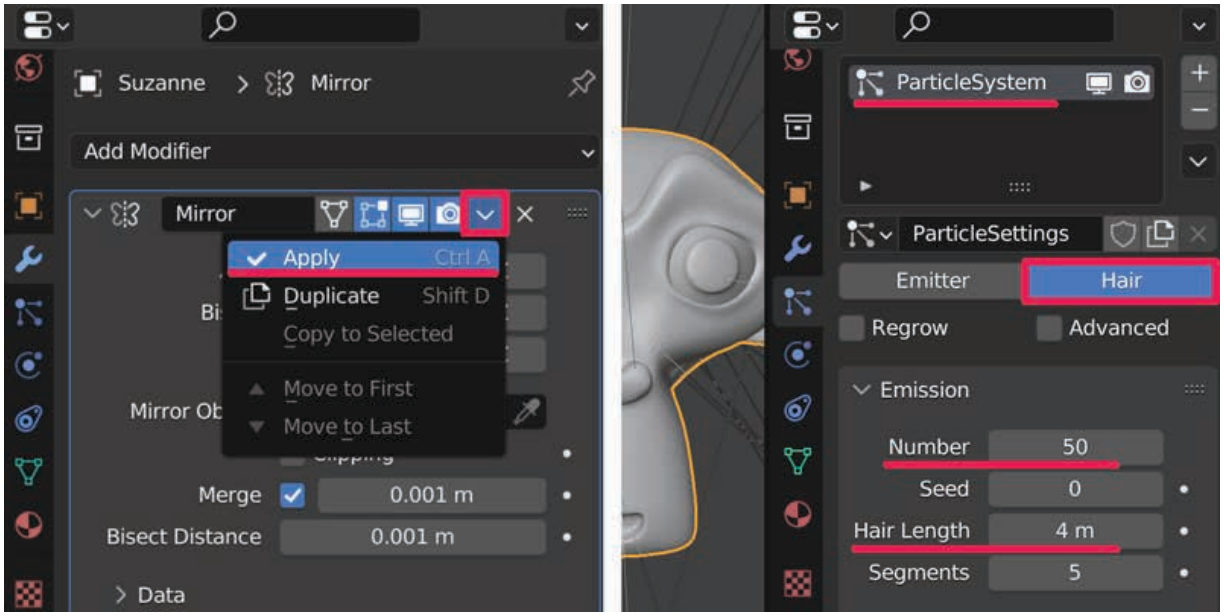
Add a **Mirror modifier**. Turn on **Bisect** so you don't have to delete half your mesh to mirror it.



Now you can weight paint symmetrically.

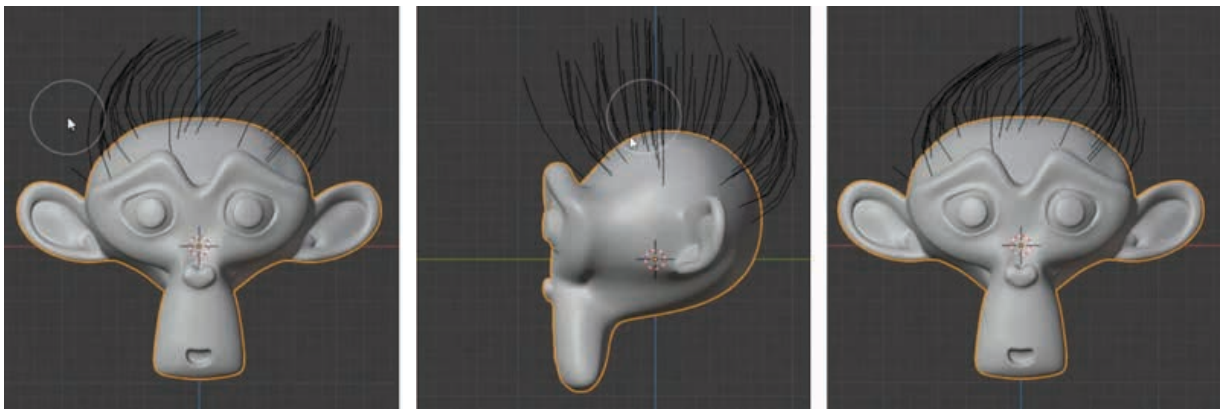


Apply the Mirror modifier and add a **Particle system**. Set it to **Hair** and reduce the amount of particles and the length.

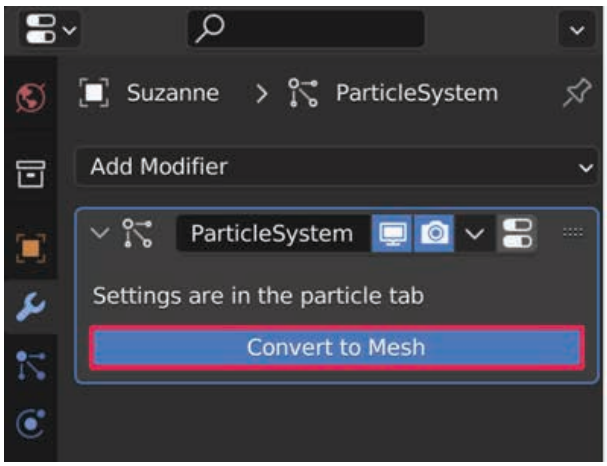


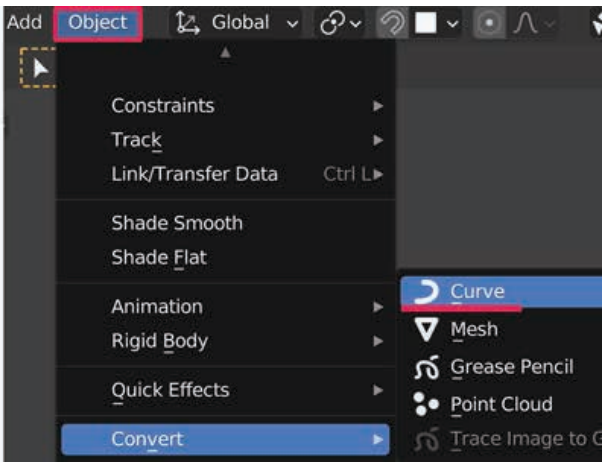
For **Density** use the weight paint group.

Style the hair in **Particle Edit** mode.

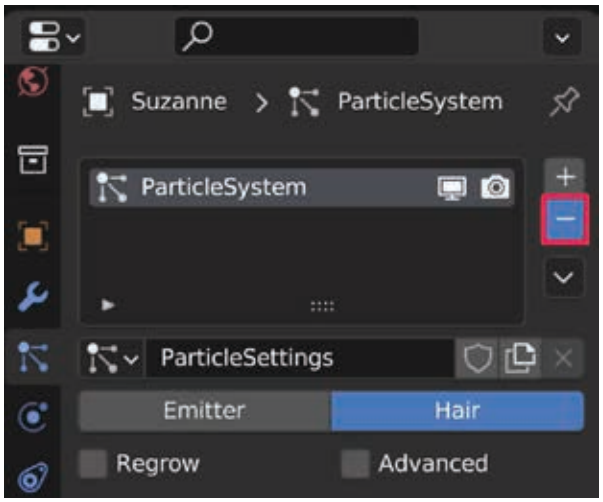


Apply the Particle modifier and convert it to **Curves**.

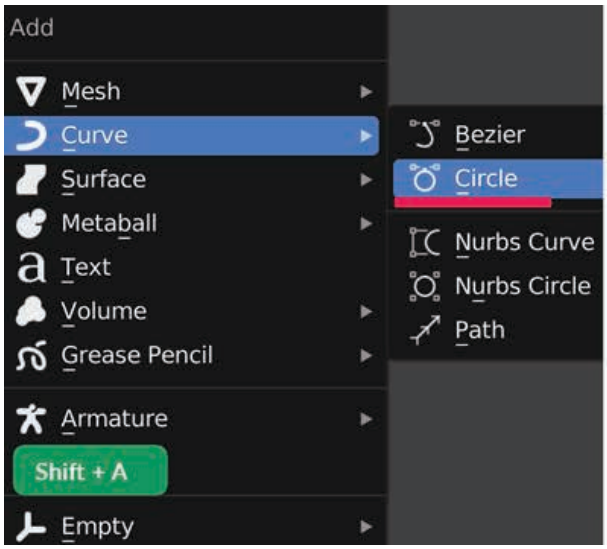


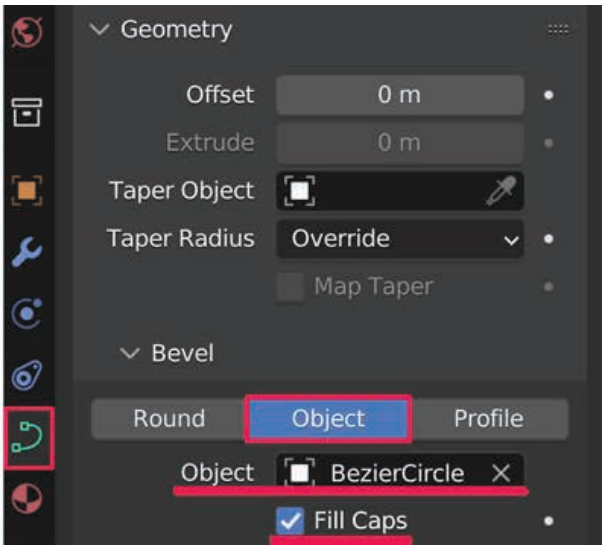


Remove the particle system by clicking on the - icon.

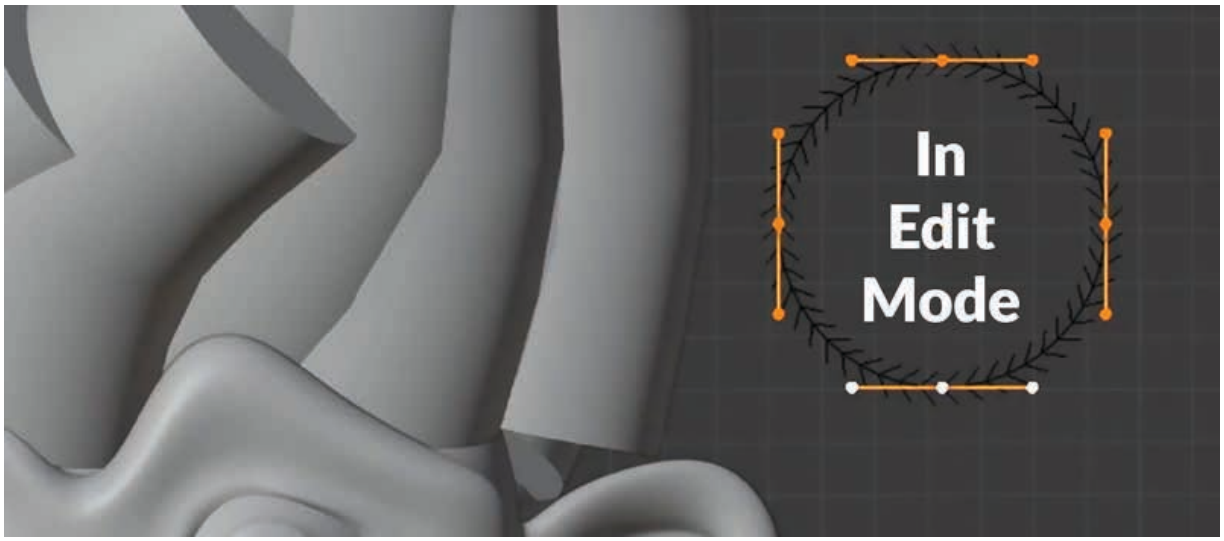


In Object Mode, add a **Bezier Circle** and use it as the **Bevel Object** for the hair curves. Turn on **Fill Caps** so the points are closed.

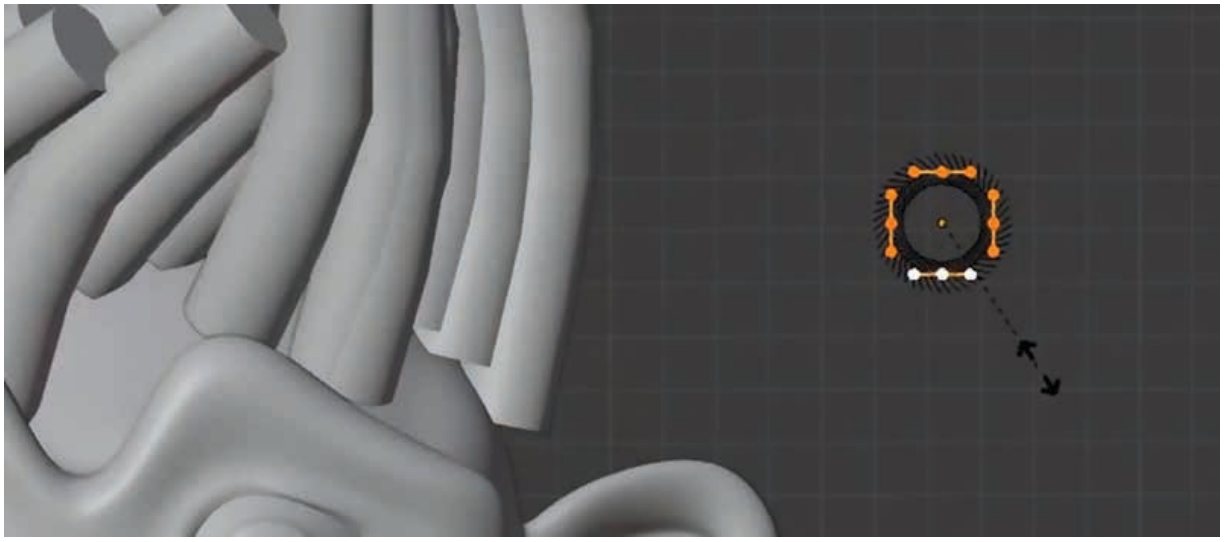




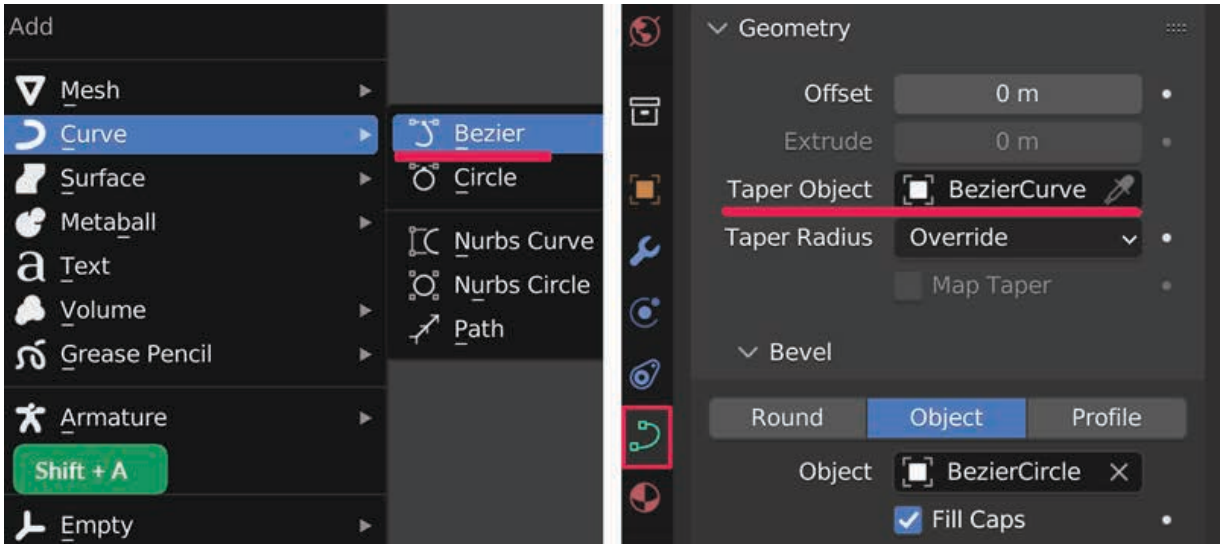
Scale the **Bezier Circle** to decrease the hair thickness.



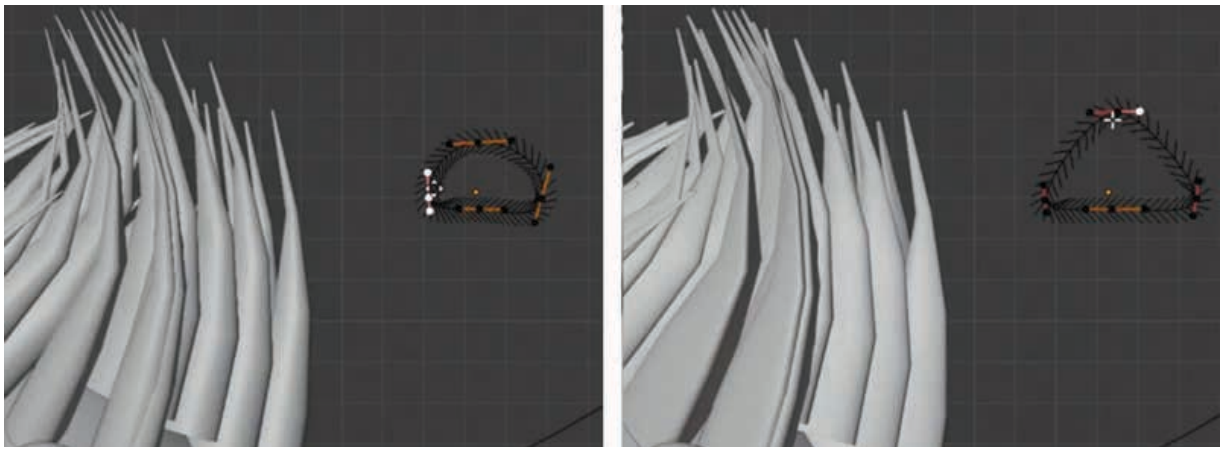
As you can see, the “hairs” will scale as you scale the Bezier Circle.



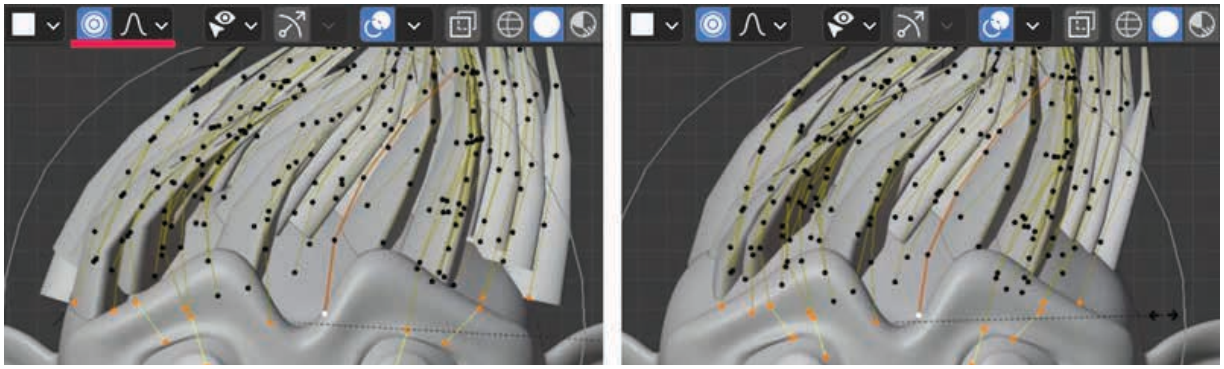
Add a **Bezier Curve** and use that as the **Taper Object**.



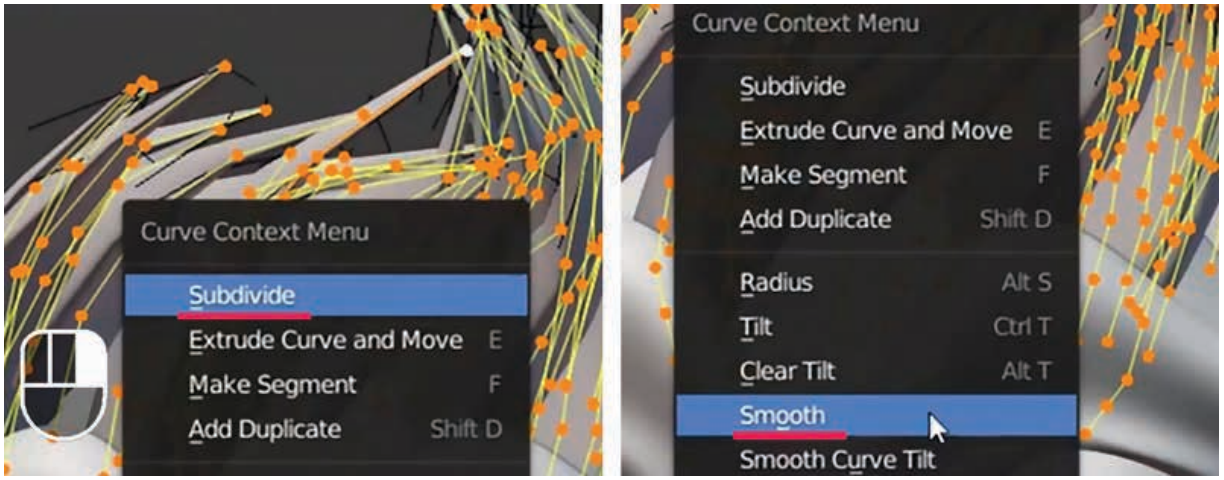
You can edit the circle to change the hair shape.



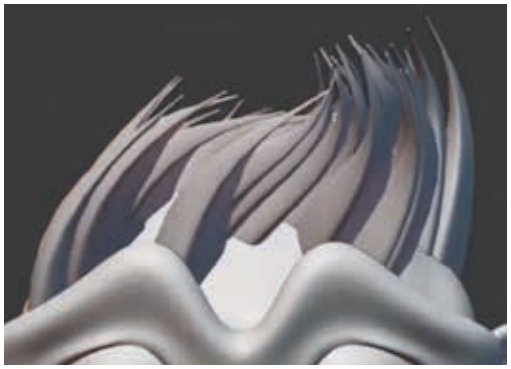
Using proportional editing, you can style the hair a bit more.



For more **Smooth** results, subdivide and smooth the curves.



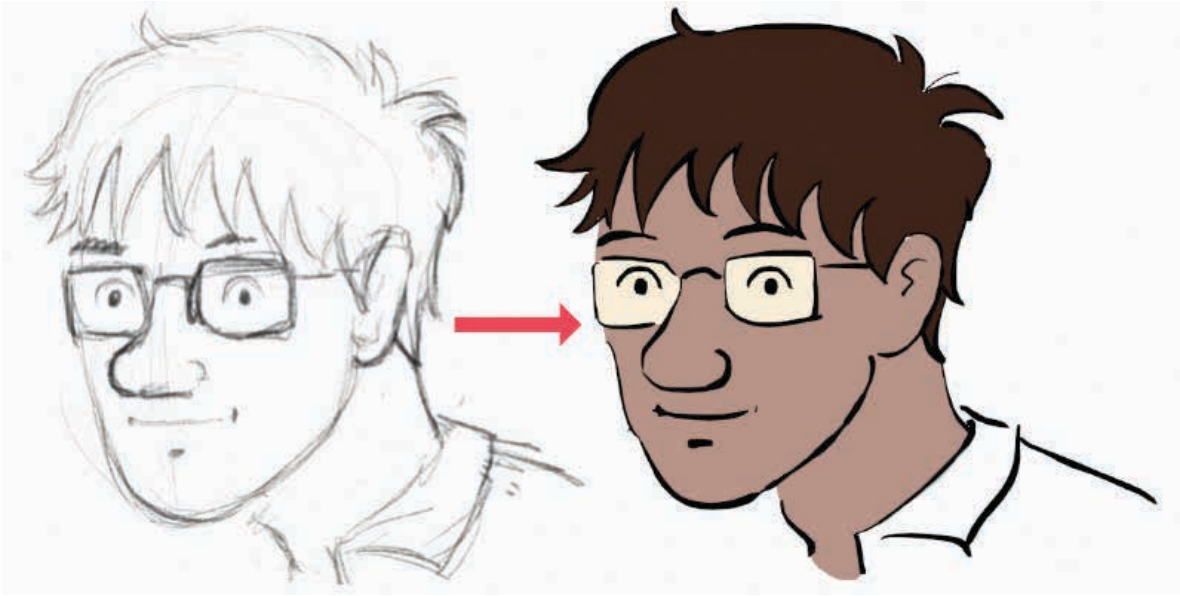
This is a quick one-minute result... but now that you know how to do it I'm sure you can do much better than me.



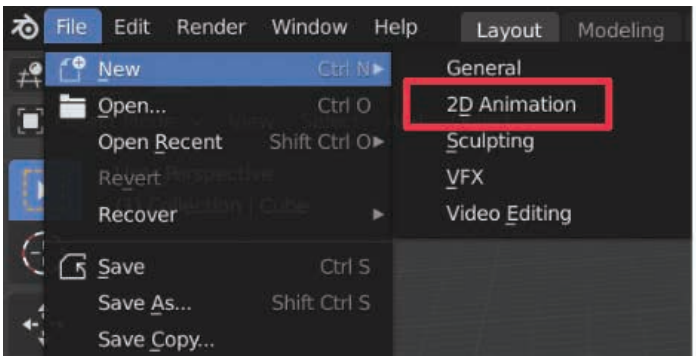
GREASE PENCIL

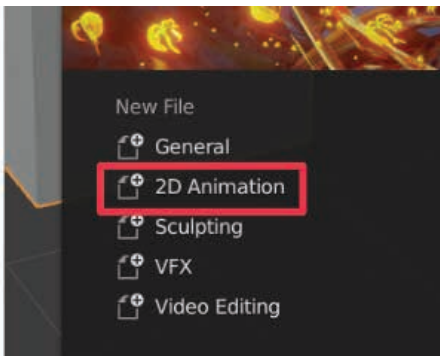
Grease Pencil Basics

On the following pages we'll look at the basics of drawing a sketch and turning it into a drawing with clean lines and colors using Grease Pencil.



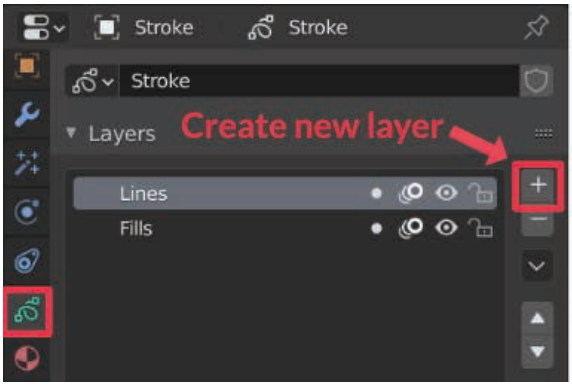
Blender comes with a convenient workspace for drawing 2D Animation using Grease Pencil. To access it, either go to **New > 2D Animation** or choose it from the Splash Screen on opening Blender. This gives you a nice clean canvas and a layout featuring everything you need for drawing with Grease Pencil.

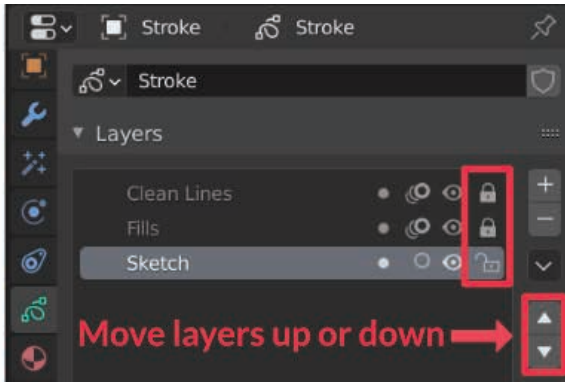




If you go to the **Stroke** tab, you will find the default layers called Lines and Fills. I like to rename Lines to **Clean Lines** (double-click on the name to be able to change it) and add another layer called **Sketch**. You can move layers up and down with the arrow icons.

Sketch should be at the bottom. Select the Sketch layer so you will draw on that layer only. If you're extra paranoid like me, you can lock the other layers just in case.



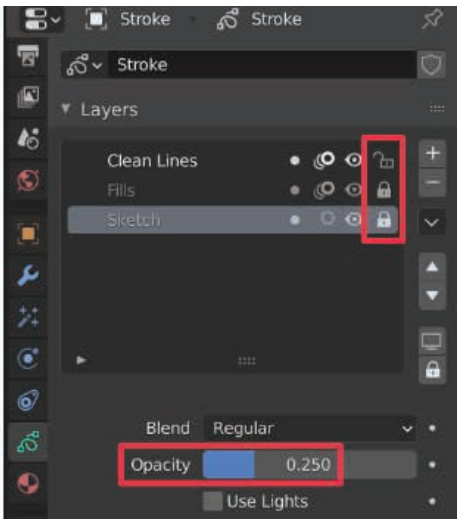


Now, draw the sketch of whatever you're going to animate, like a cartoon character for example. As it's a sketch, you don't have to worry about perfect lines, instead try to find expressive lines.

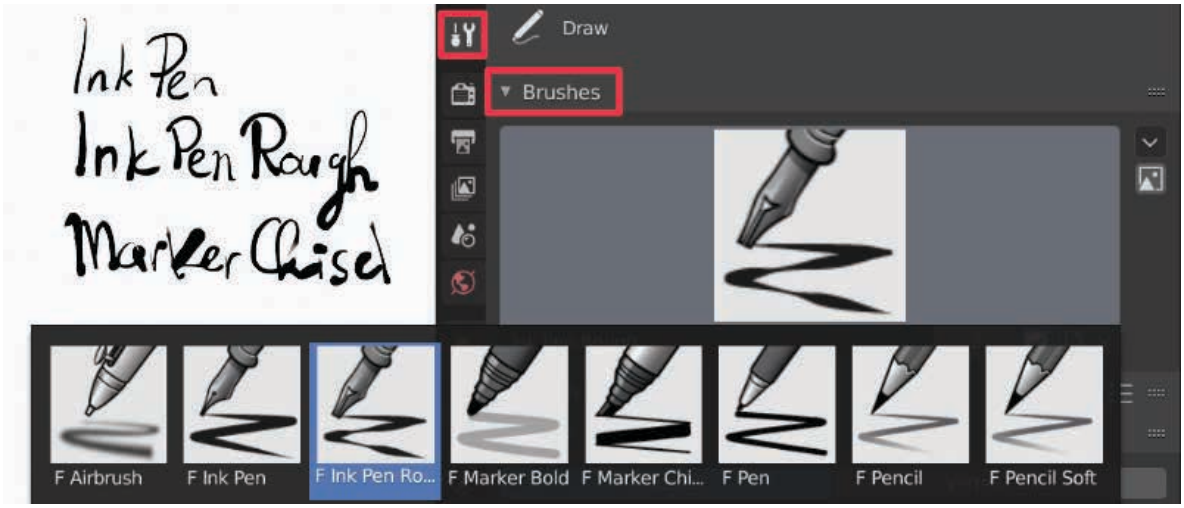
We can worry about making it all perfect later. Big nose, glasses... seems familiar, maybe it's a self-portrait.

Anyway, when you're satisfied with the sketch lowertheopacity of the Sketch layer. 25% opacity is usually enough to still be visible as reference.

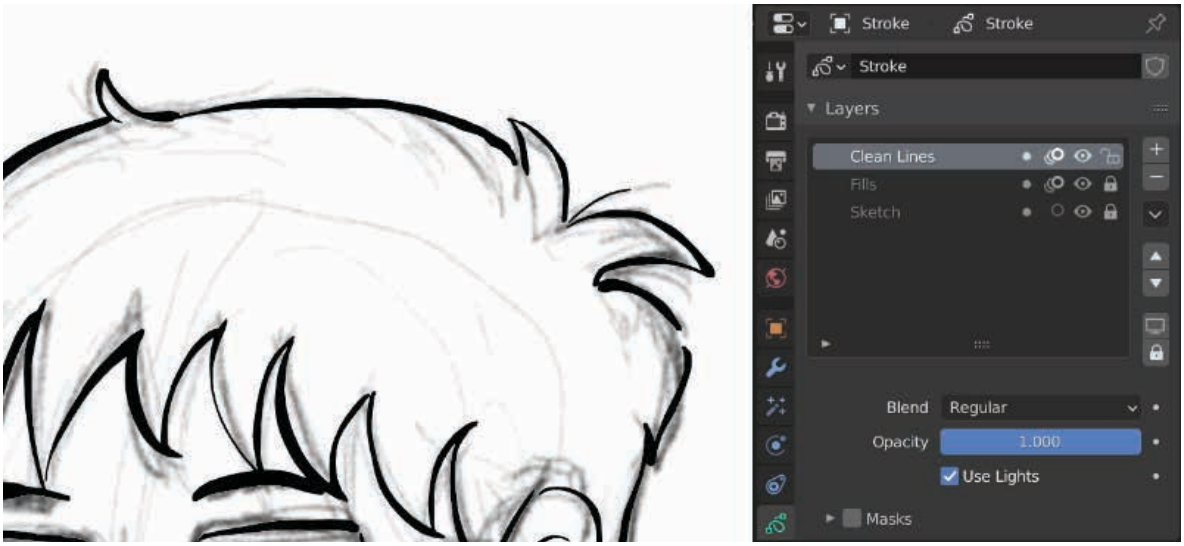
Then lock the Sketch layer so you don't accidentally draw your clean lines on it and unlock the Clean Lines layer. Clickonthe Clean Lines layer so it's selected.



For drawing clean lines I recommend **Ink Pen**, **Ink Pen Rough** or **Marker Chisel**, all of which give nice lines. Try them out for yourself to see which one fits your style.



Now draw clean lines on the Clean Lines layer using the sketch as your guide. If you are not happy with a clean line, just undo by **pressing Ctrl+Z** and try it again. Try to draw them in a quick stroke. Better to try a few quick strokes than to slowly make a perfect stroke. Quick strokes look more energetic and interesting.



A drawing tip: To draw really dynamic lines, try to make the gestures with your elbow as the pivot point (keep the rest of your arm stiff, so don't turn your wrist). It's easier to draw from your wrist, so it's tempting to do it like that.

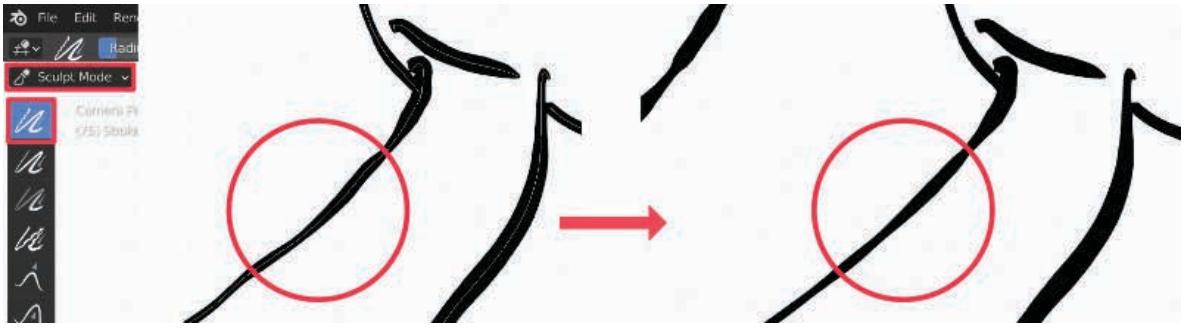
But you will not be able to create such smooth, long and dynamic strokes as when you master drawing by moving from your elbow. Drawing from your elbow is harder, but ultimately better.

Cheat like an artist!

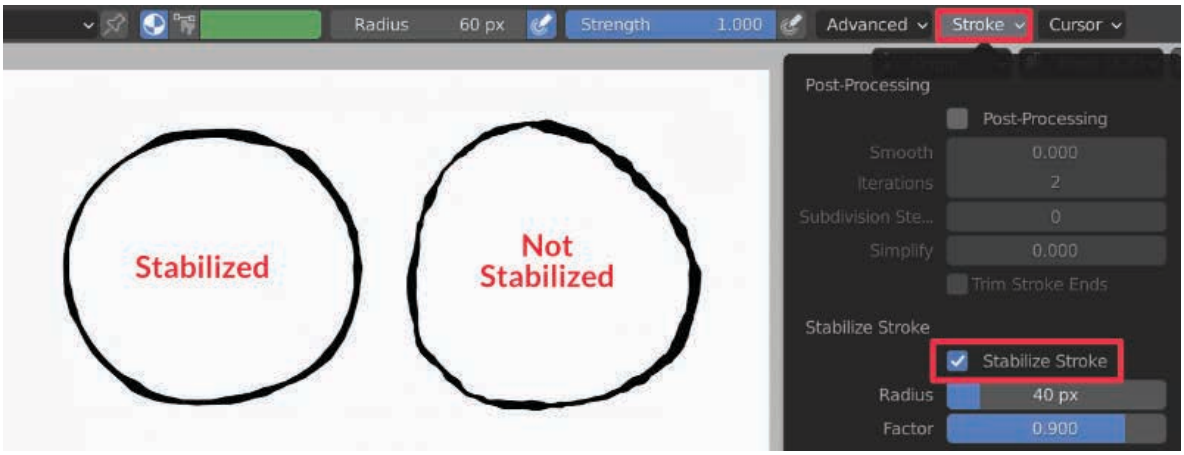
There's a couple of things we can do to "cheat" and make our art better. For example, it's one thing to draw one eye, but then you need to draw the other eye pretty much the same way, which can be challenging.

In **Edit Mode**, select the strokes that make up one eye, duplicate them with **Shift+D** and **press G** and move them over to the other side of the face. Just remember to go back to **Draw** mode afterwards.

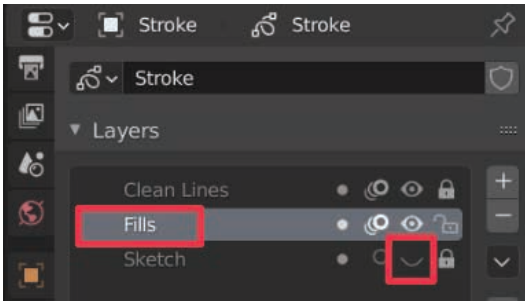
Another good trick to "cheat" and make your lines better is to use the **Smooth Brush** in Sculpt mode. With this brush, massage the lines you feel are too wiggly, and they will become smooth, as if you drew them full of conviction and years of drawing experience! Nobody will know your dark secret of having used the Smooth Brush. Except me.



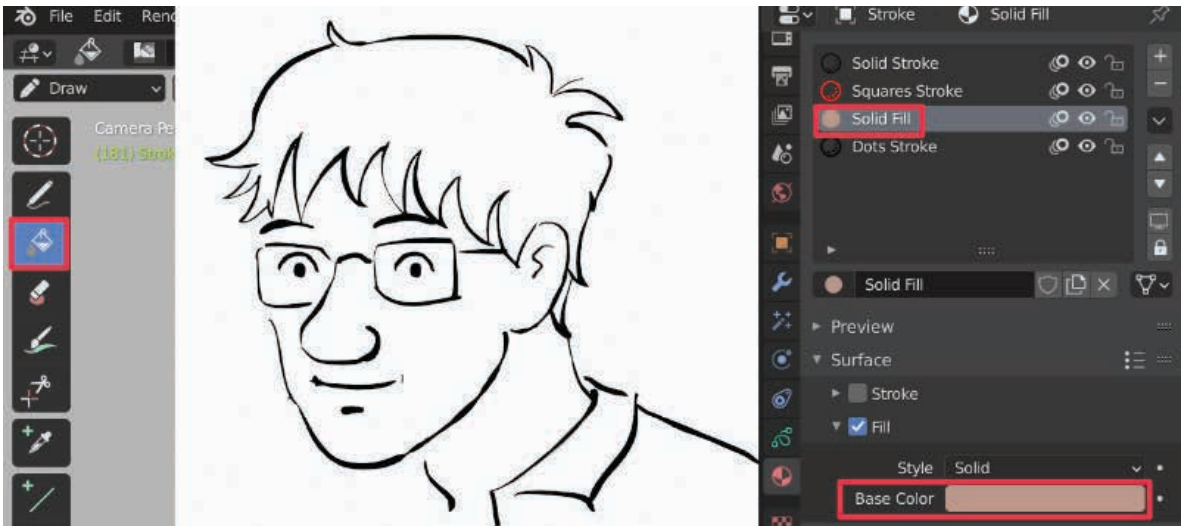
Another way to avoid squiggly lines altogether is using the **Stabilize** function. This is accessed by **holding down Shift** while drawing. There are some options that can be accessed from the Stroke menu, but I recommend using the keyboard shortcut with the default settings. Don't go too crazy with this though.



Once you've finished drawing the Clean Lines, you can simply turn off the visibility of the Sketch layer. Lock the Clean Lines layer and unlock and select the **Fill Layer**.

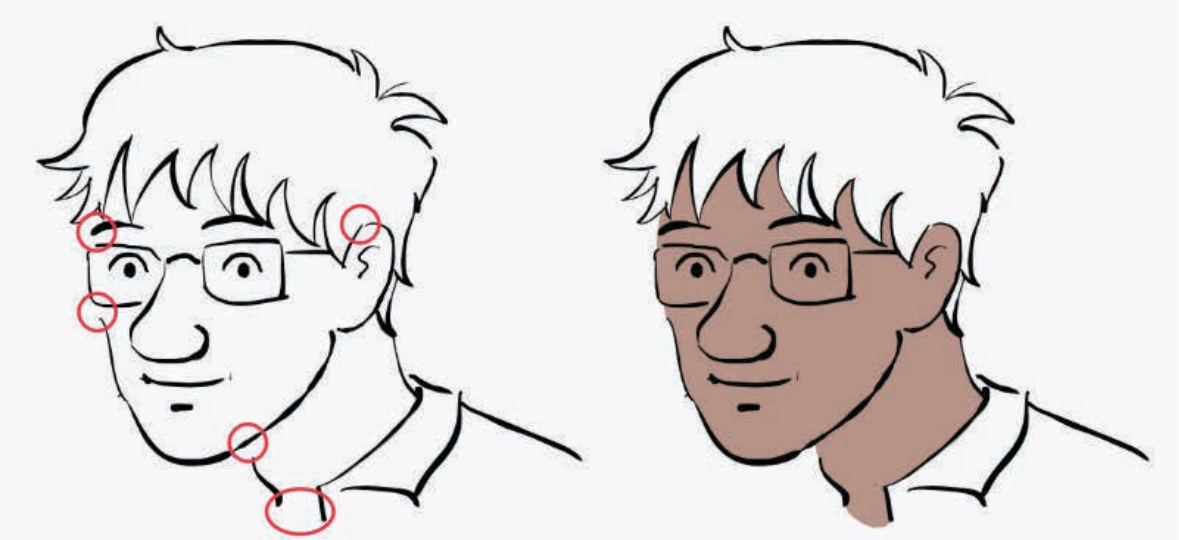


Now to fill, select the **Fill tool** in the Tool panel, then in the Materials tab select **Solid Fill** and change the **Base Color** to what you need.



Unfortunately, if you try to fill by clicking on the canvas, it will fill more than just what you want when there are gaps in the drawing like there are with mine. Fill needs to have really clearly defined close boundaries to work well.

Luckily there's a solution for this. **Holding Alt**, draw some invisible strokes at the areas where there are gaps in the drawing that the Fill would "leak" out of (circled below left). Now try the Fill tool again. It works!



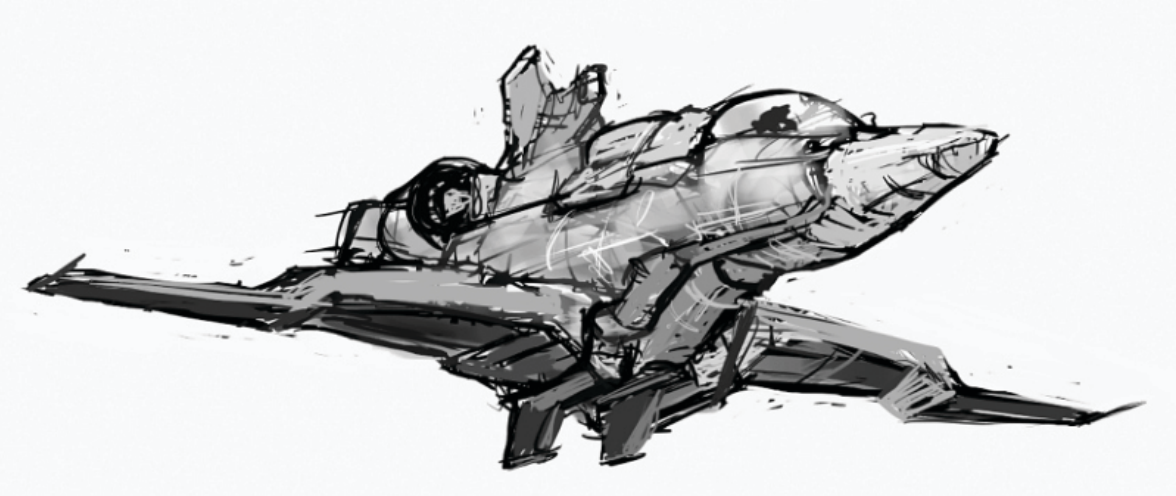
You'll need to create a new material for each color that you want to use as a fill. To do this **click on the [+] icon** in the Materials tab, give the new color a name and check Fill, uncheck Stroke. Then change its BaseColor. As before, you'll need to hold Alt to draw some invisible boundary strokes before you fill.

Grease Pencil for Concept Art

Concept art, the drawings and paintings created during pre-production of movies and games, were traditionally always made in 2D, either with traditional art techniques or by painting digitally on the computer.

Recently 3D is being used more to get "free" perspective, which gives artists more time to focus on the creative parts. Grease Pencil is a great tool for this, as it's 2D which also lives in the 3D space, meaning you can still change the camera angle or lens type even after drawing on top of the 3D shapes.

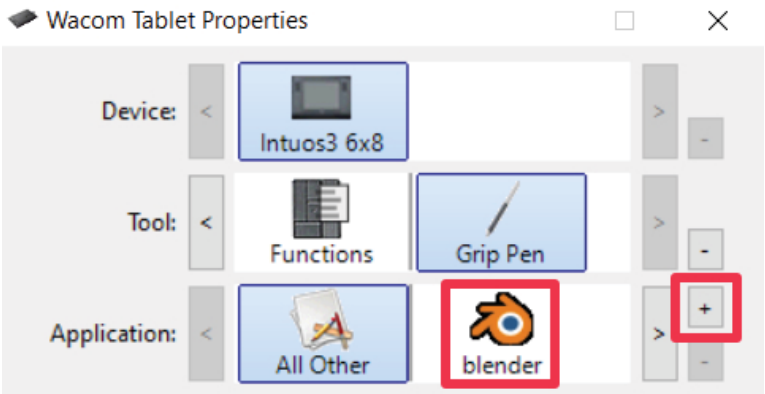
In this tutorial we take a look at the basic workflow for using Grease Pencil to create some quick 3D sketches that we can then paint over in a 2D package like Krita or Photoshop.



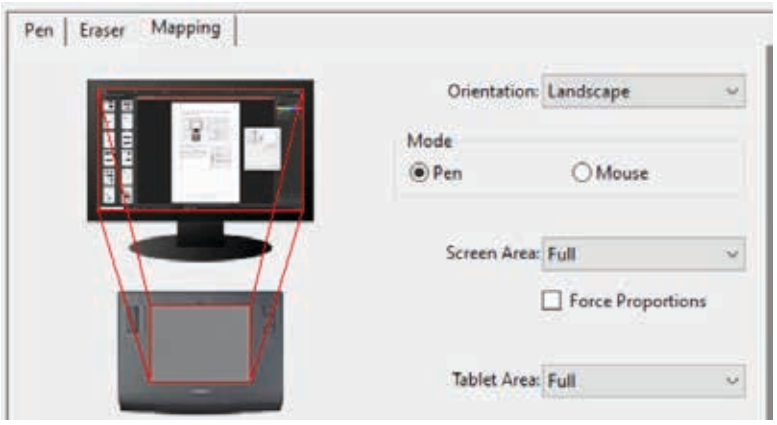
Before we get started, make sure our graphic tablet is set up in the most optimal way. If you're using a Wacom tablet like me, I recommend going into the properties (if you don't have the Wacom Properties app, download it from the Wacom website).

First of all, add Blender to the list of applications by **clicking on the + icon**.

Find the **Blender.exe** file and select it, so it is listed under Applications in the Wacom Tablet Properties.

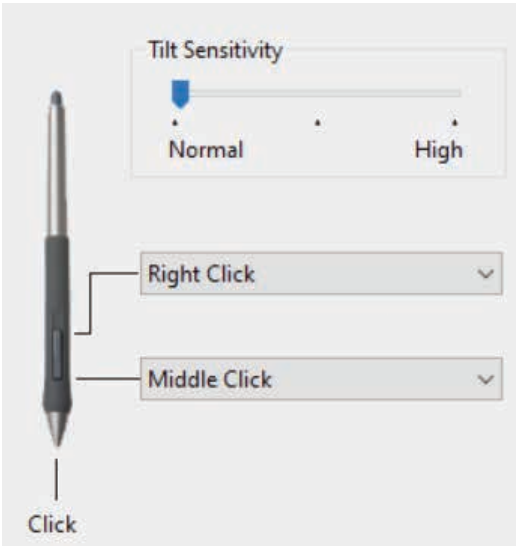


Also make sure that the tablet to screen relationship is 1 to 1, so you can move the cursor quickly across the screen.



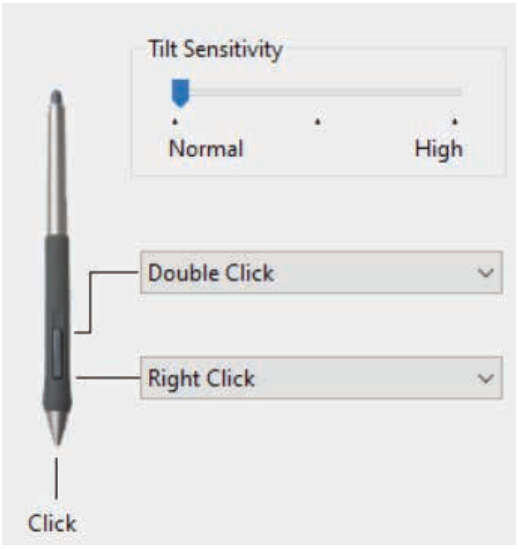
How you set up the buttons on the tablet is up to your personal preference.

Personally I have "**emulate 3-button mouse**" enabled in Blender's preferences, so I set up my tablet pen like the picture shown on the right.

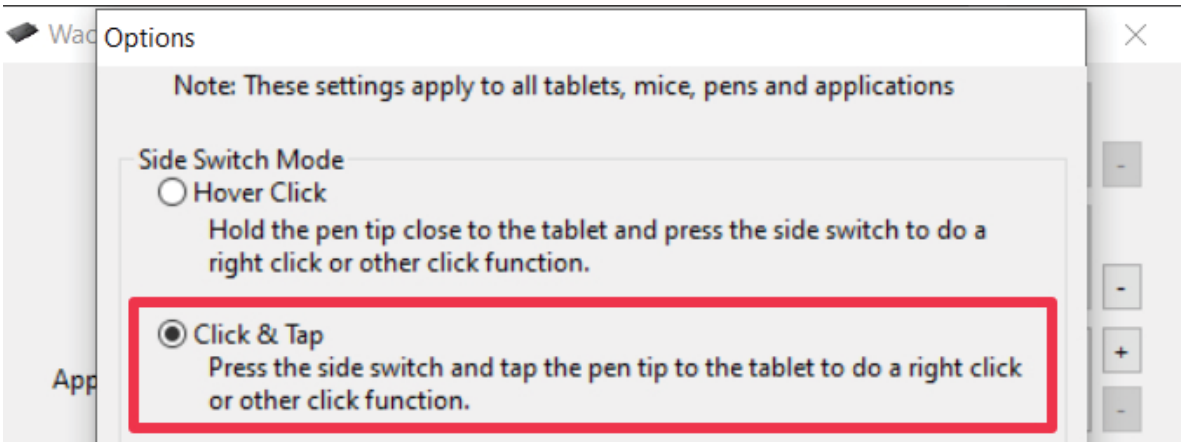


However if you use the default of Blender where you have to hold down the middle mouse button to rotate the viewport, I recommend the settings shown on the right.

That way, you can rotate the viewport by pressing down the lower part of the side button of your Wacom pen. It's probably best to use the defaults if you're new to Blender!



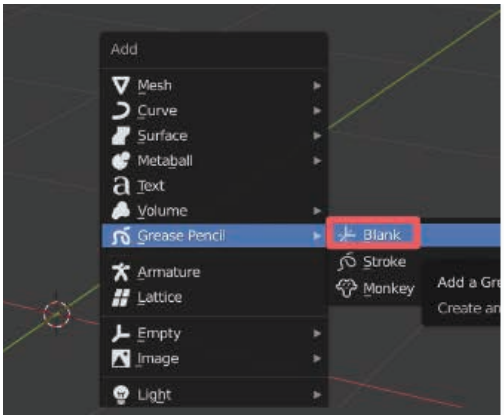
Finally I recommend that you click on **Options** and set the side button mode to the second option (if this is not the way by default). This basically means that you click and hold the pen down, then press on the side button of the pen to open the right-click menu in Blender.

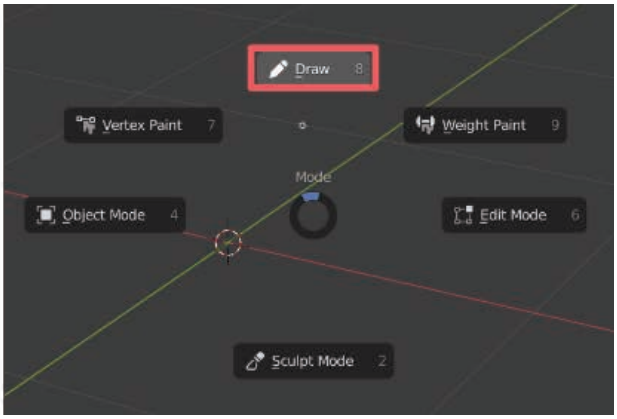


Now that we've made sure our graphic tablet is working as it should - how do you even **draw a Grease Pencil stroke**?

From Blender 2.8 on, Grease Pencil is an object just like the default cube or anything else, that you can add from the Add menu (press Shift+A to open this), after which you go to Grease Pencil > Blank.

Then you **press Ctrl+Tab** to open the pie menu and drag the mouse up for "**Draw**". After this you can draw strokes by **dragging the left-mouse-button**. To be able to scale, rotate and move the grease pencil object you **press Ctrl+Tab** again to open the same pie menu, this time moving the mouse cursor left to "**Object Mode**"

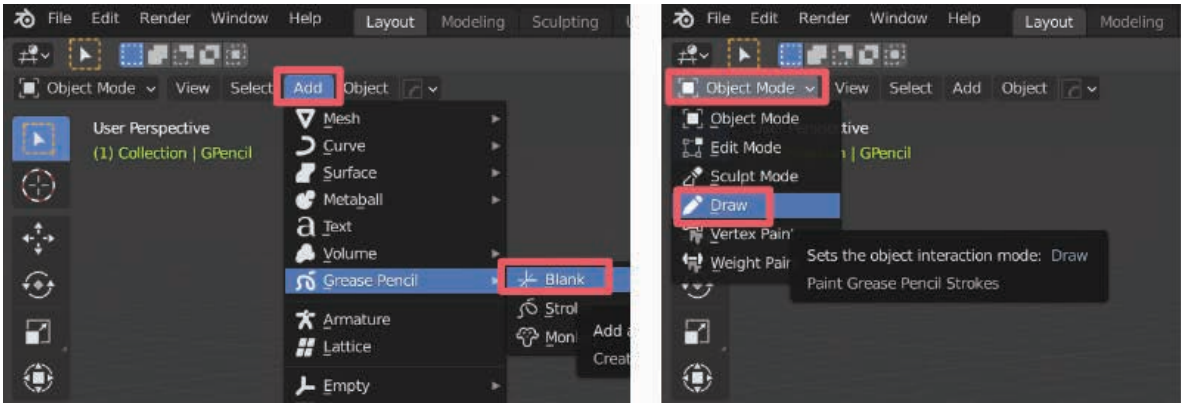




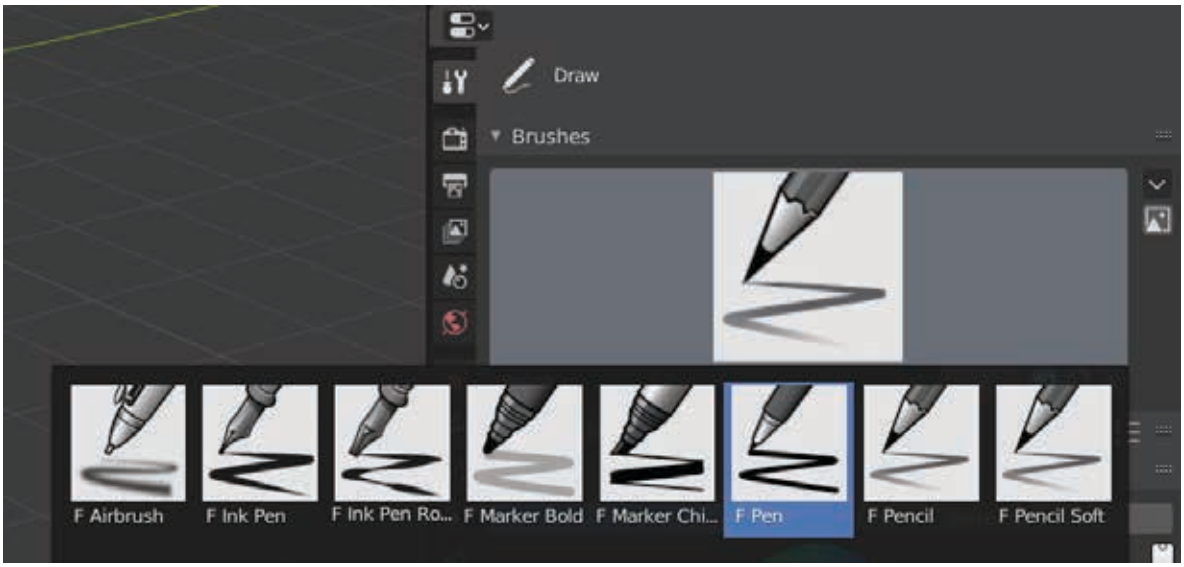
I recommend that you get used to this pie menu, as it's actually very quick once you are accustomed to it (especially when using a tablet pen).

If all these keyboard shortcuts and pie menus are making you sweat, don't worry- you can also find these options from drop-down menus in the top left corner of the viewport.

The result is the same, once you have a Blank Grease Pencil object you can start to draw strokes. It just takes a bit more time to always go to that drop-down menu when you need to switch between Object Mode, Draw and Edit Mode.

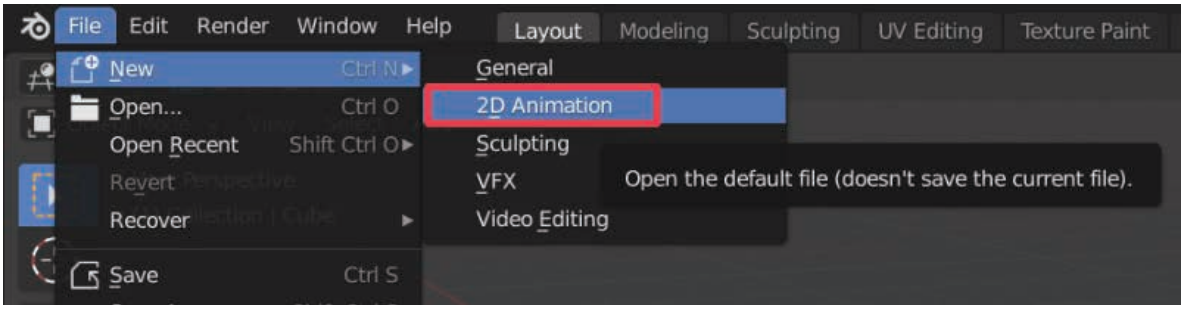


As soon as you're in Draw mode, you can draw strokes. And you can also choose different brushes by going to the **Active Tool and Workspace settings** tab (the top tab in the Properties panel) and clicking on the big brush icon.

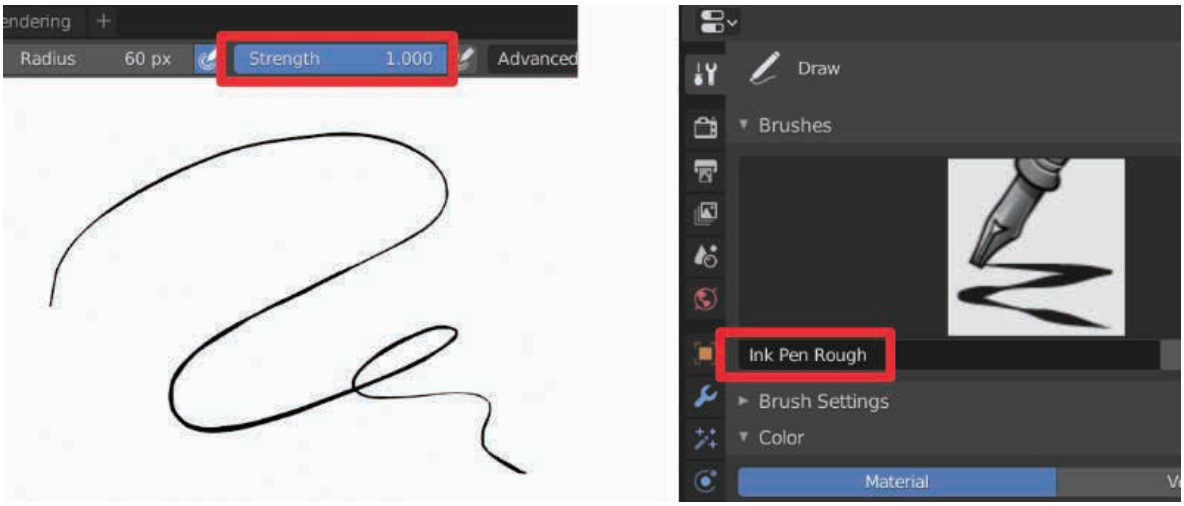


Try them all out to see which brushes you like to work with. You can erase by holding Ctrl (or select the EraserTool from the Tool menu, to erase faster)

These are the very basics of drawing a Grease Pencil stroke. I don't however recommend that you do all this in the regular 3D viewport. When you start up Blender you can go straight to a **2D Animation** project that sets up a nice blank canvas for you in the 2D Animation viewport layout. Or you can go to **File > New > 2D Animation** for the same thing. This is a really nice interface to sketch in using Grease Pencil where all the Grease Pencil options are easily accessible.

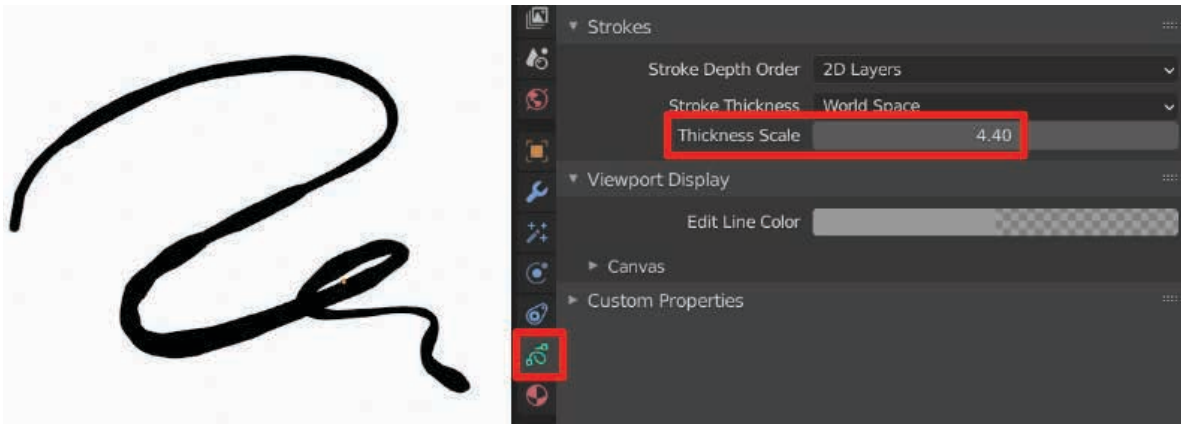


Personally I really like the **Ink Pen Rough brush**, because it creates these organic-looking strokes with varying thickness. I usually set **Strength to 1**. You can lower the layer opacity later if you want. While you can increase the **Radius** of the stroke in the field next to Strength, you can also **press F** to do this more quickly.



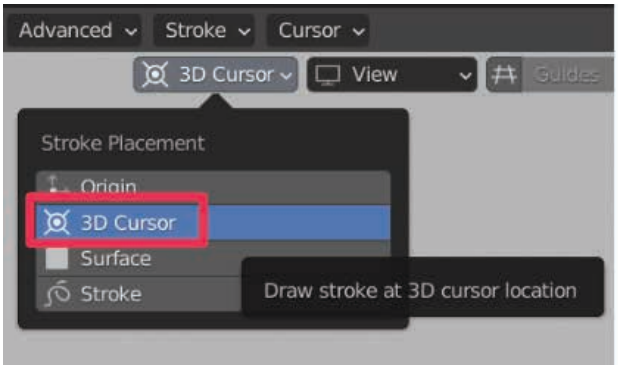
Something quite interesting is that you can also change the global thickness of all the strokes at once. Say you've made a drawing, but you want the strokes to be thicker or thinner, this can be done using the **Thickness Scale** value in the **Stroke** tab.

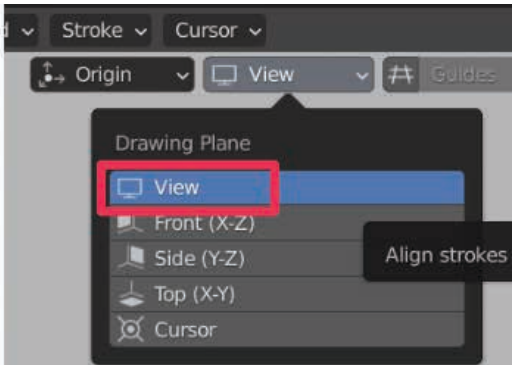
Where this may come in handy is for example, when you've duplicated some things and scaled them up or down. In such a case you might want to decrease or increase the stroke thickness to keep it consistent with the other Grease Pencil objects.



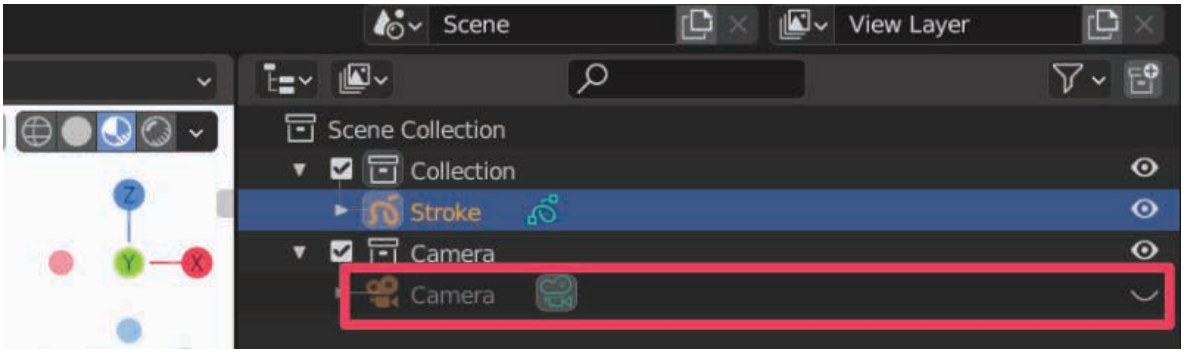
One more thing before we actually draw something. You can choose where the strokes are placed - Origin or **3D Cursor** are good choices for now. The Drawing Plane is how the stroke is placed in 3D space.

I recommend you set this to **View**, that way you can quickly add strokes along different 3D planes simply by rotating the 3D view or by switching to Front, Right or Top Orthographic view.

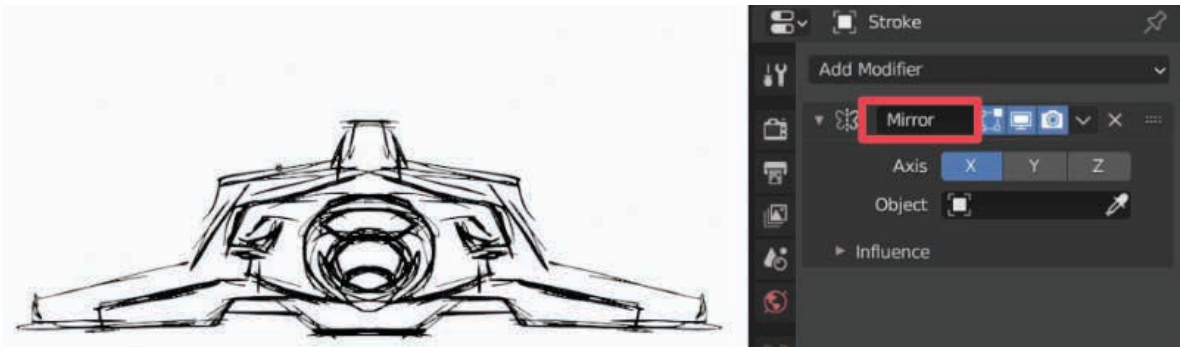




Okay, enough basics - time to create something! Go to **Front** Orthographic view by **pressing 1** on the numpad. You can hide the Camera in the Outliner so it doesn't get in the way.

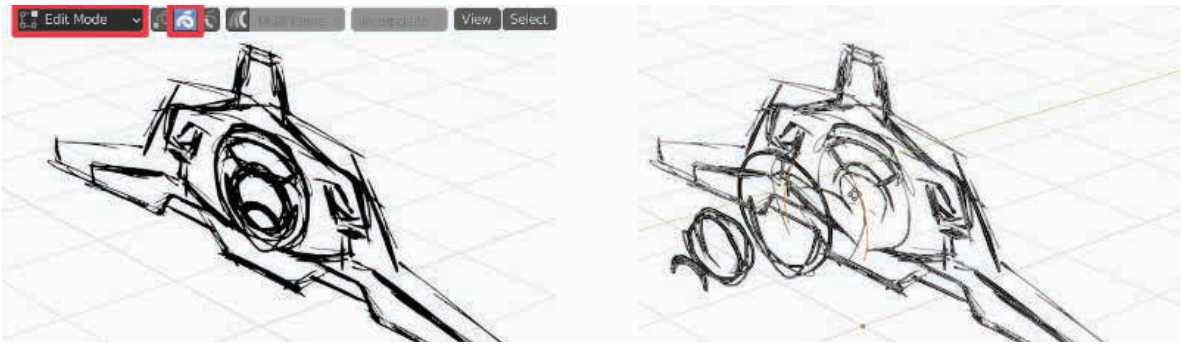


To draw something symmetrically, add a **Mirror modifier** to the Grease Pencil object. Now if you draw some strokes, they will mirror to the other side, which is very useful for quickly exploring visual concepts.

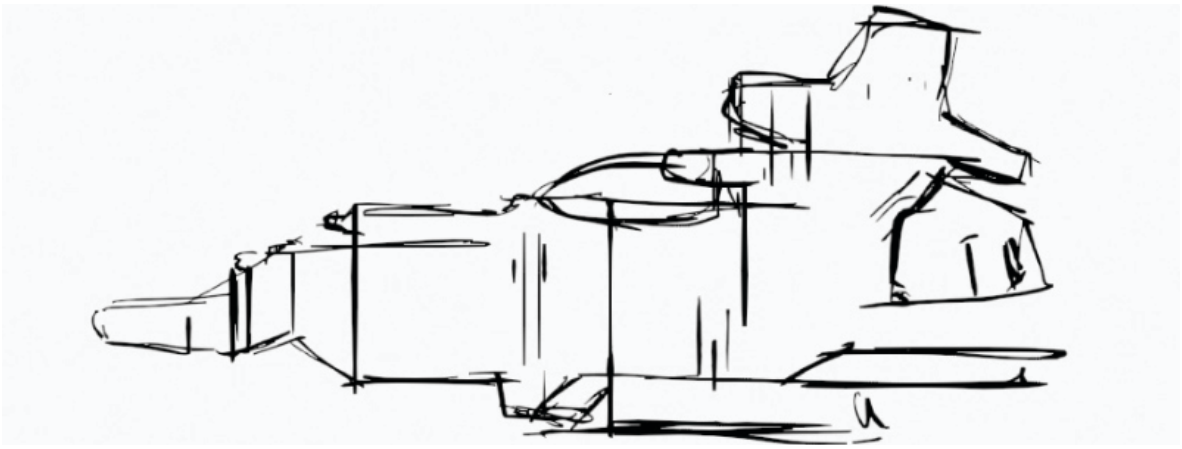


You can go into Edit Mode and "Select all stroke points" (which is kind of like Edge selection mode with regular 3D objects) to move the individual strokes of a Grease Pencil object.

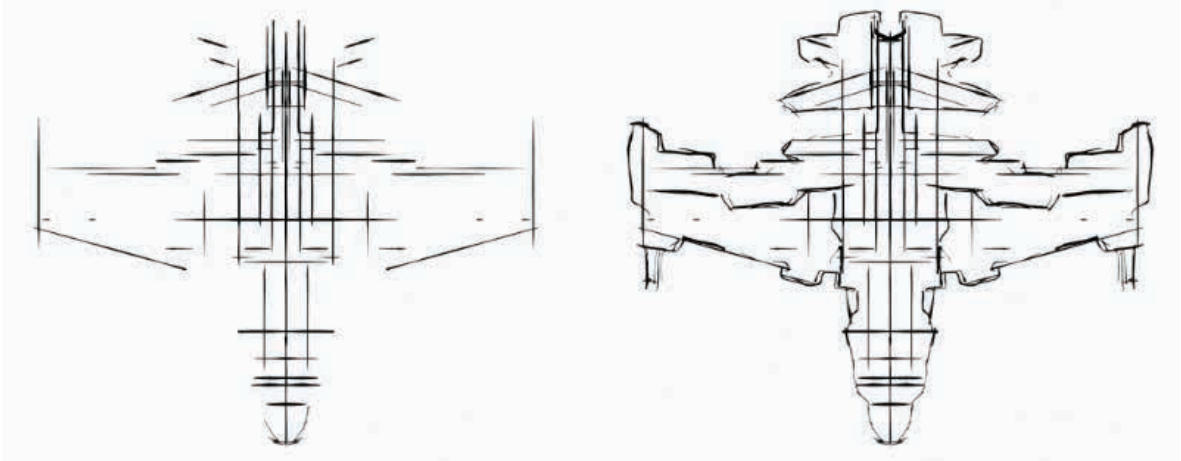
This is useful if you want to move where a stroke was placed in 3D. To make the drawing less flat, distribute the strokes in 3D space by selecting them and moving them with **G+Y** (if you want to move them on the Y-axis) or **G+X** (if you want to move them on the X-axis).



Then add more strokes from different angles like from the side:

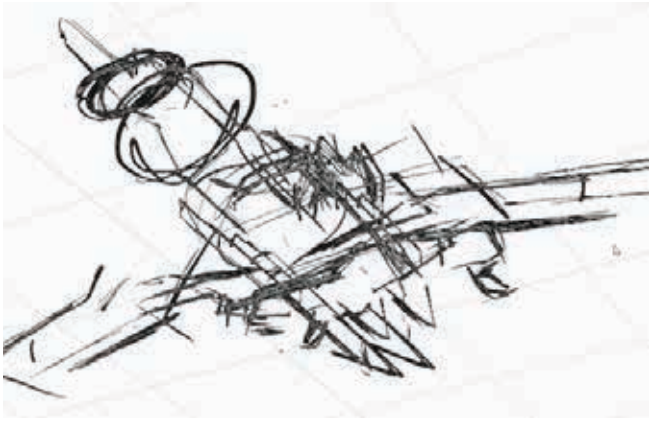


Move these strokes on the X-axis. They will also get mirrored, so you get two mirrored strokes as soon as you move them. Go to the top view and add some strokes there as well.

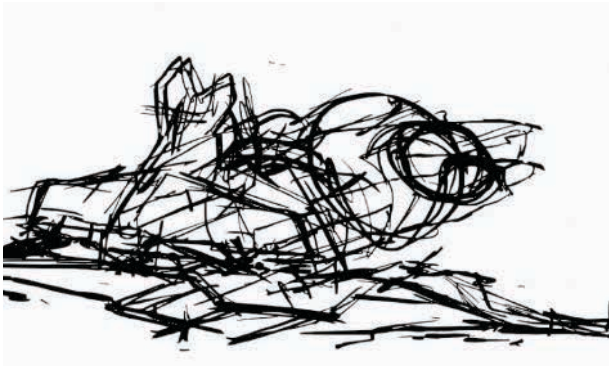


Again, select strokes in Edit Mode and move them to where you want them to be.

You can also rotate them and scale them. After a while you get something that looks like a 3D sketch.

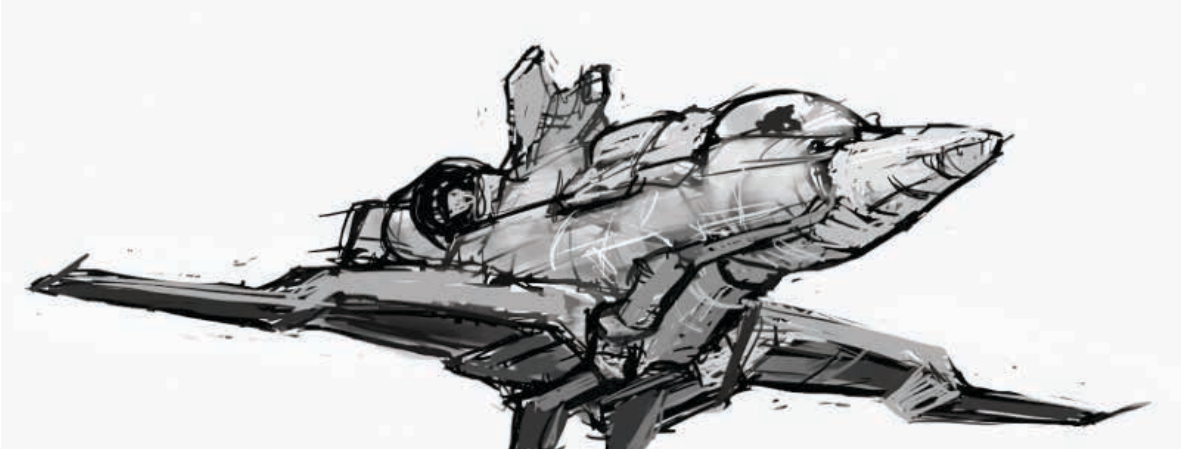


The benefit is that you have “free” perspective this way, and can quickly get the same sketch from all kinds of different angles, which would be a lot of work (and kind of a pain!) with traditional 2D drawing.



Of course you can use the best of both worlds - take some screenshots from different 3D angles, then import those in your favorite 2D drawing package (mine is Krita, which is free and open-source).

Then you can start drawing on top of the screenshot.

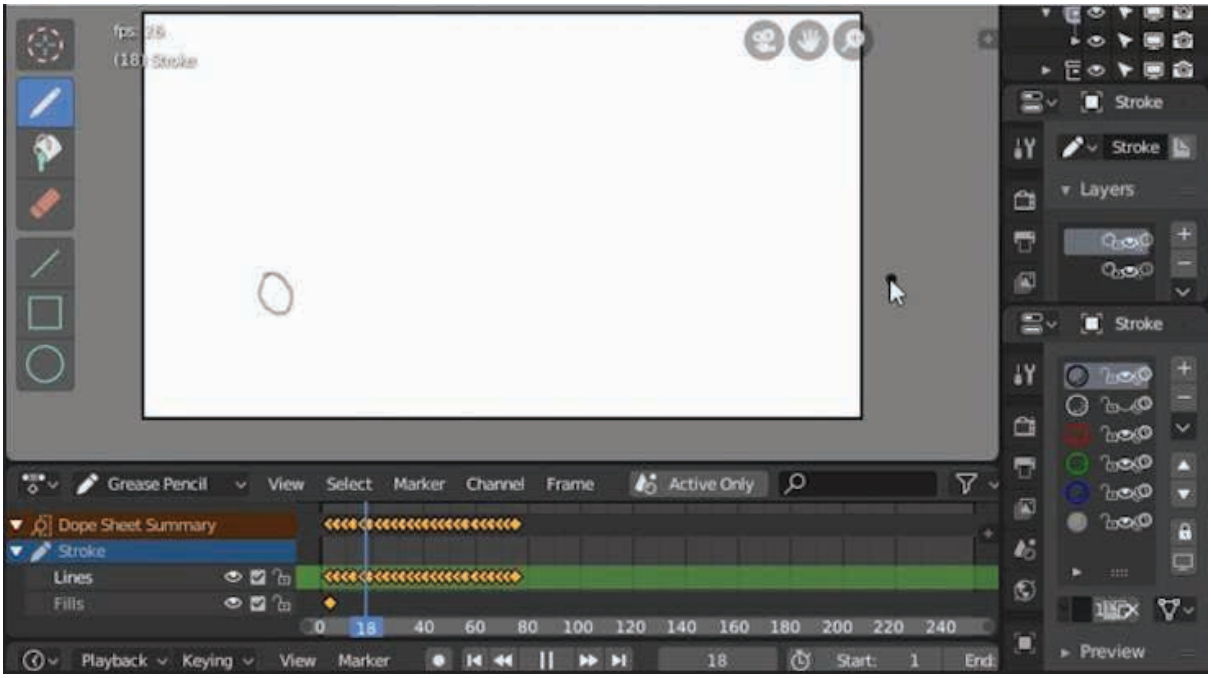


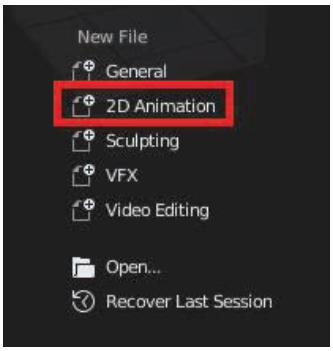
Alternatively to do a 2D paintover, you could add 3D geometry inside the Grease Pencil object and draw additional Grease Pencil detail strokes on the surface of that geometry.

That has the added advantage that you don't see "through" the drawing, though it will take some time to add the 3D geometry.

2D Animation Quick Start

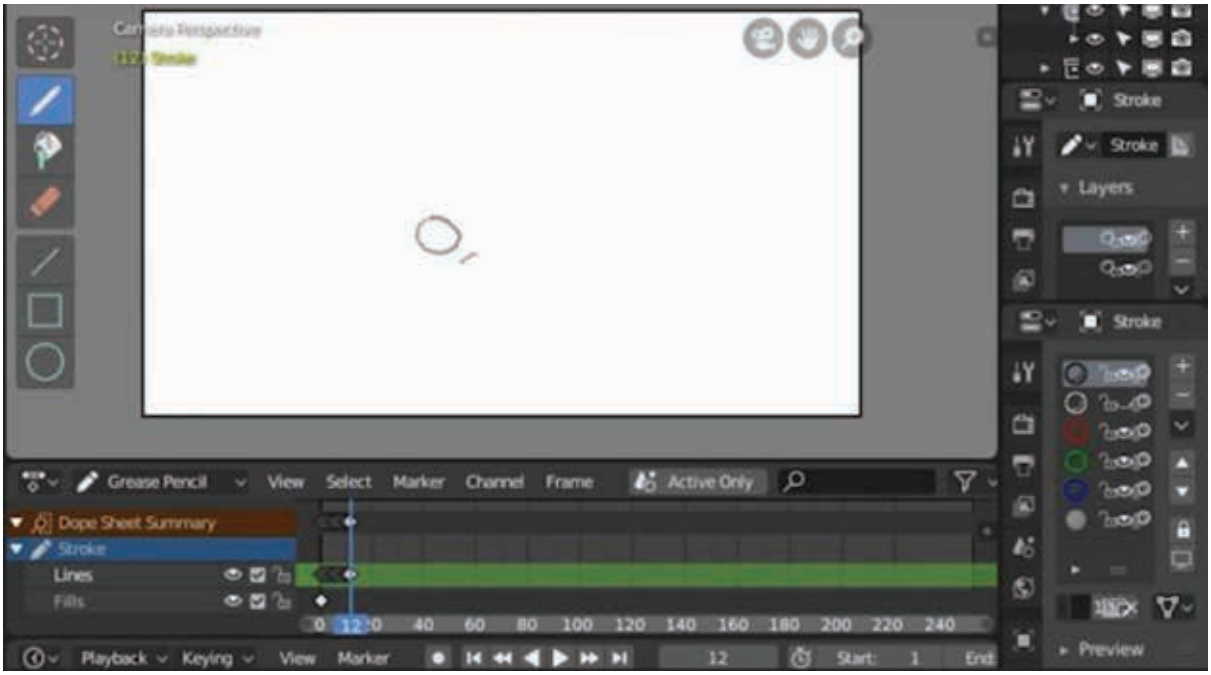
One of the improved parts of Blender 2.8 is the 2D Animation workflow.



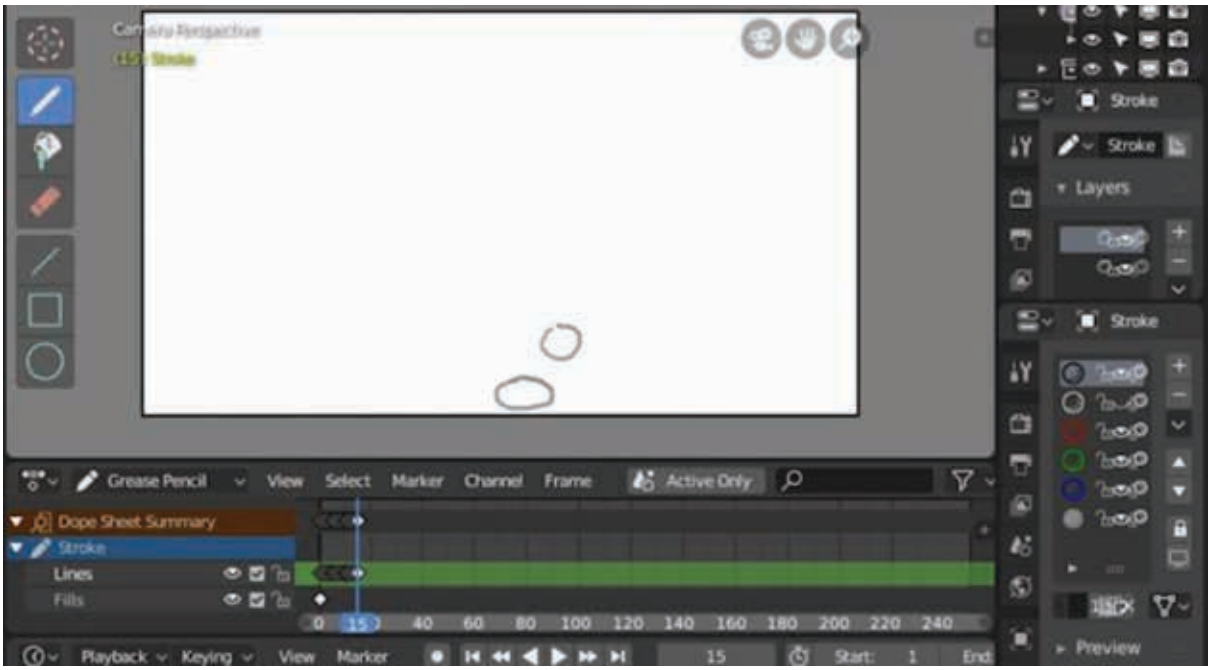


You can go straight to the 2D environment from the start-up screen by clicking on “2D Animation” from the start-up screen options.

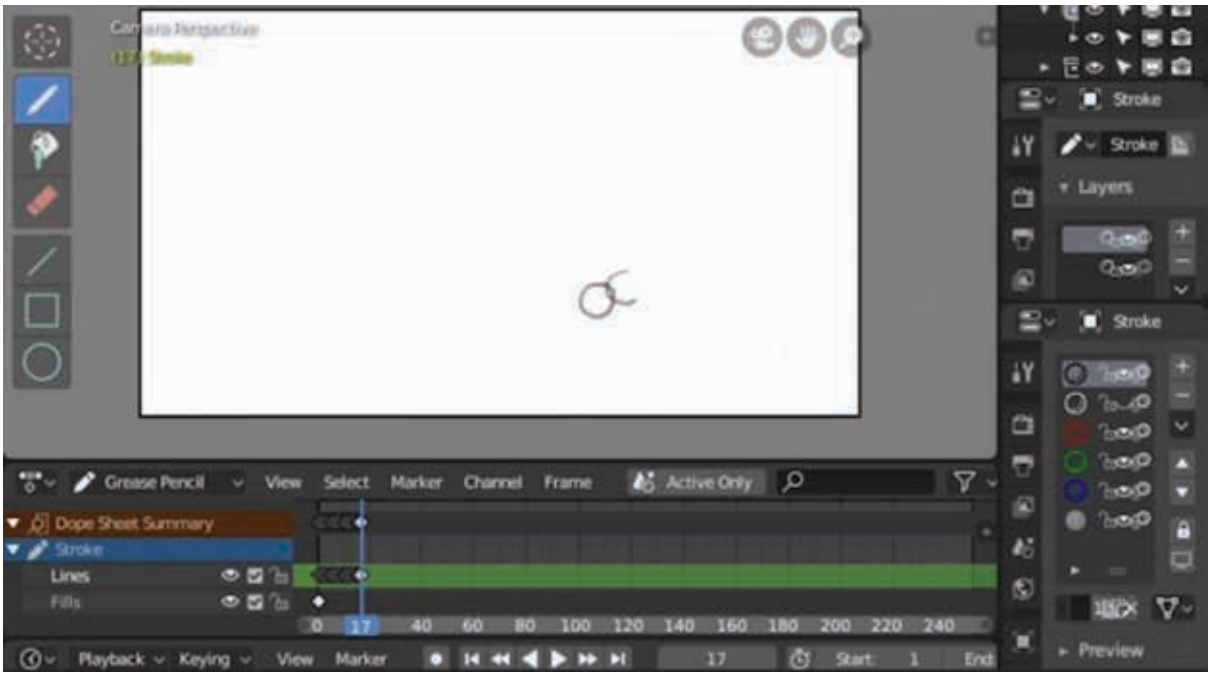
Click here to return to the index.



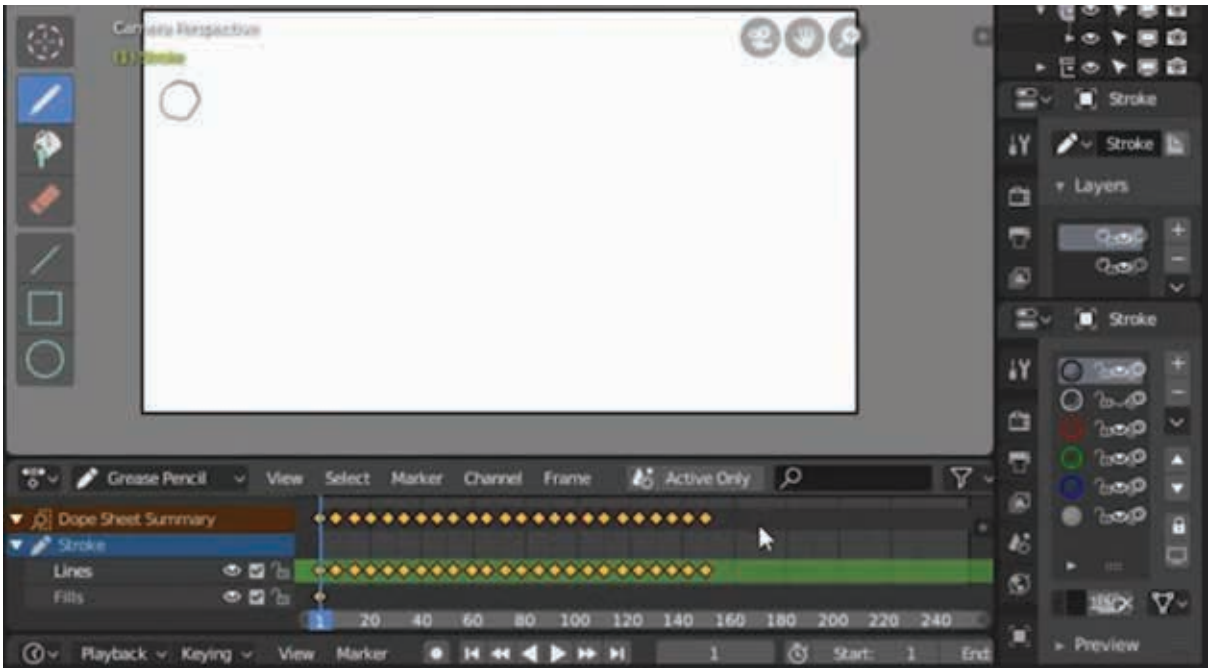
You can start drawing straight away and Blender will record a keyframe. Go back and forth with the arrow keys.



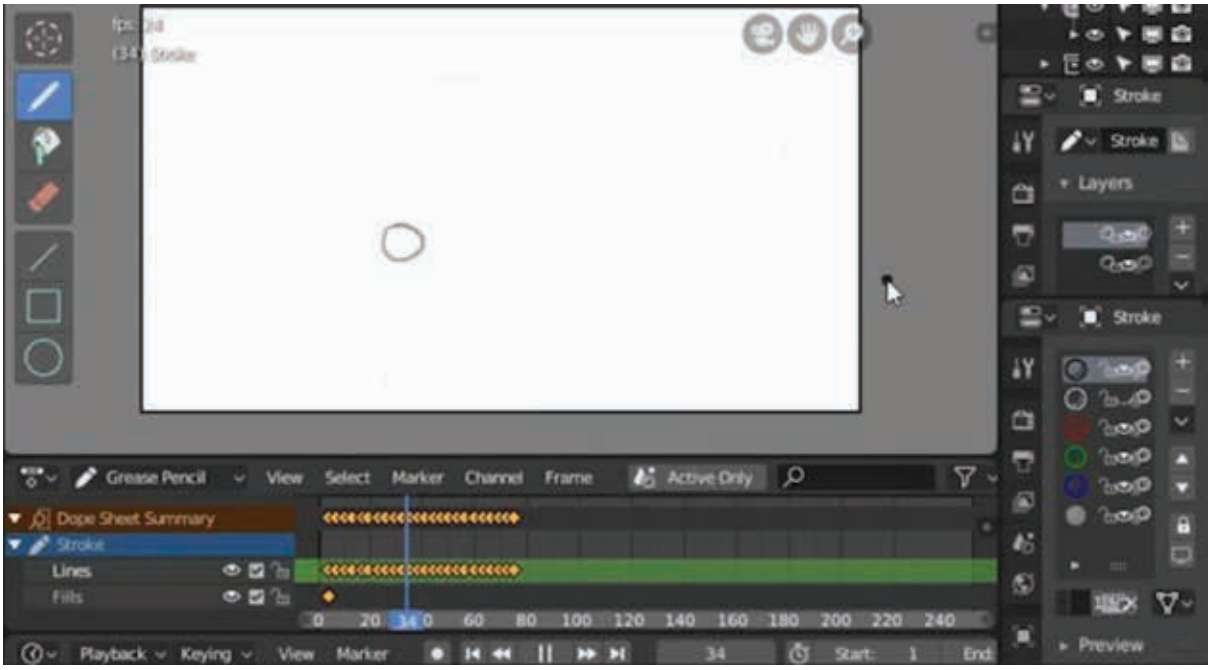
The **F-key** still gives control over the brush size. **Press shift + left-arrow key** to go back to the start frame.



You can select the keyframes and scale the length of the animation with the **S key**.

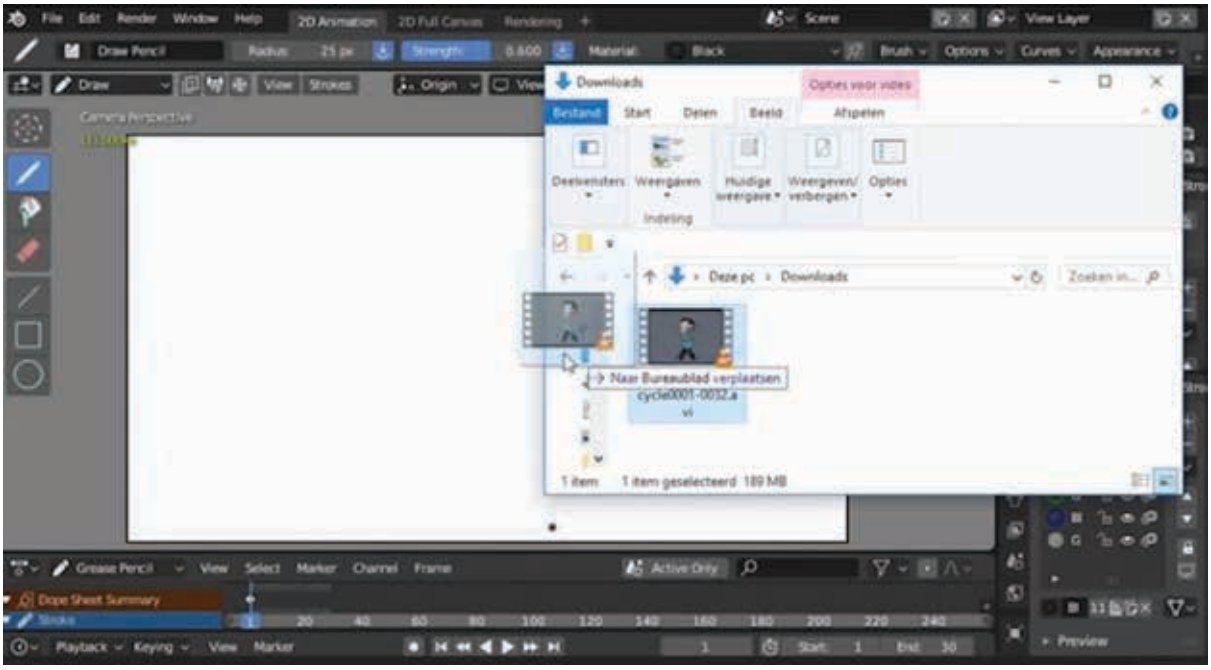


Pressing **Spacebar** now controls the playback of the timeline.

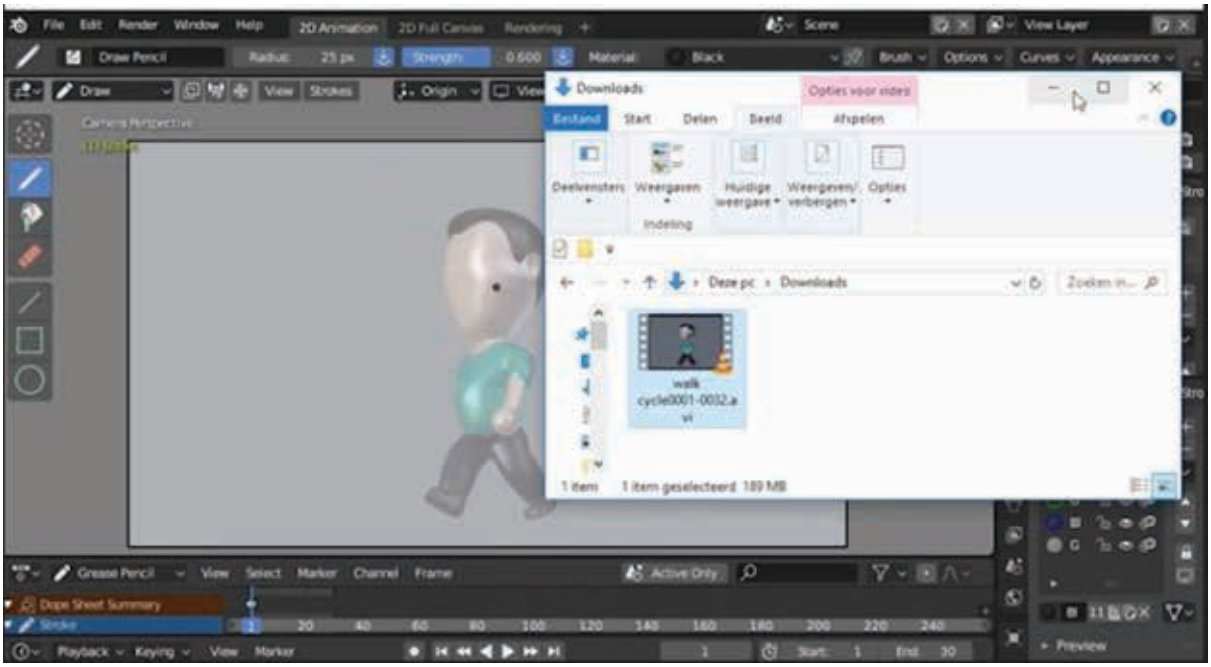


2D Animation From 3D Animation

Hand-drawn animation is really beautiful, but let's face it - it takes forever to make! However...



What about if we use 3D animation as a reference? You can now just drag video files into Blender 2.8.



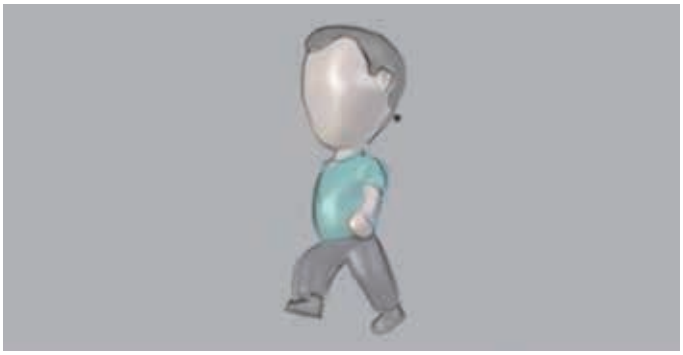
This is a viewport render of a 3D character walk cycle that I drag-and-dropped into the 2D animation interface.

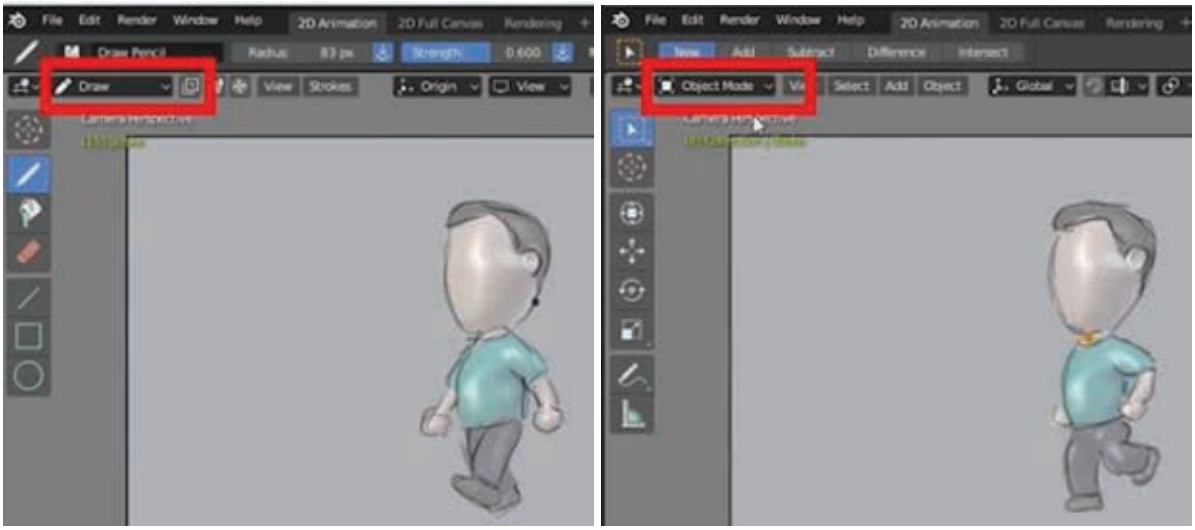
Now I just sketch on top of it. **Press F** to change the brush size and **press the arrow keys** to go back and forth.

This is certainly much easier than having a blank canvas.

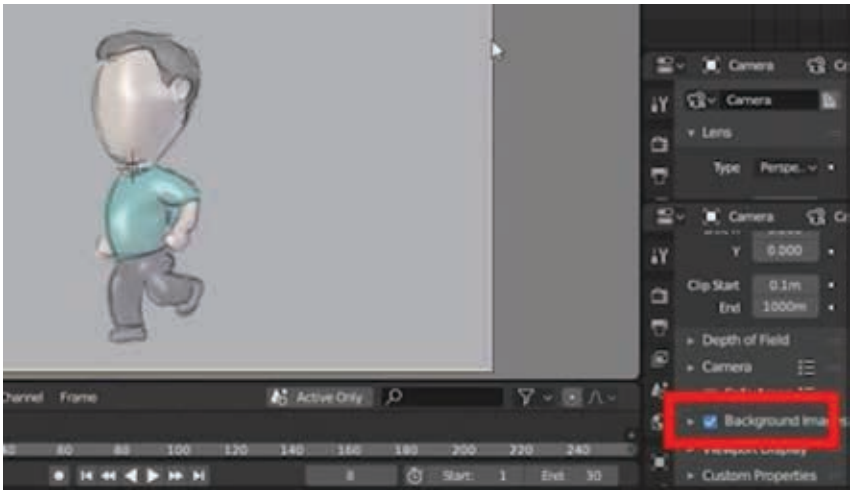
It only took me 5 minutes or so to draw all the frames for the walk cycle.

Every time you draw on a new frame, a keyframe will be created automatically.



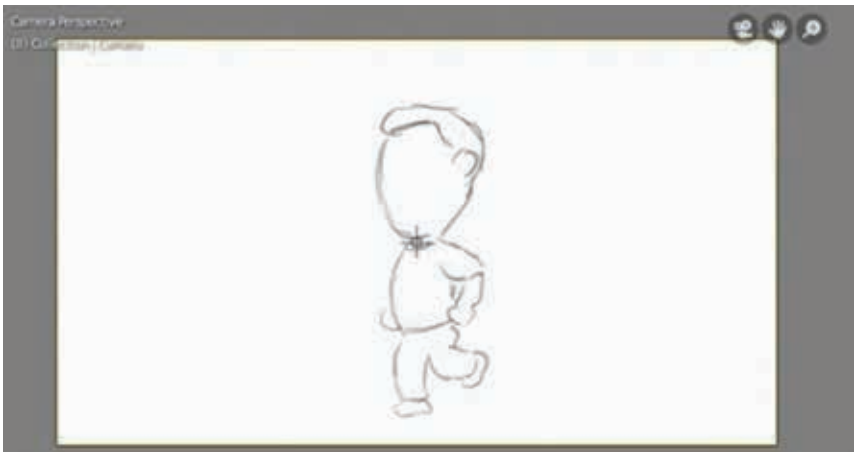


Switch from **Draw mode** to **Object mode** and select the camera. Turn off the background.



Play the animation to check the result.

Sweet! I personally really love the rough hand-drawn look. I did this very quickly and roughly and the result looks very interesting.



Grease Pencil Build Modifier

Grease Pencil has an awesome modifier called the Build Modifier. To use it, simply add the modifier after sketching something in Grease Pencil.

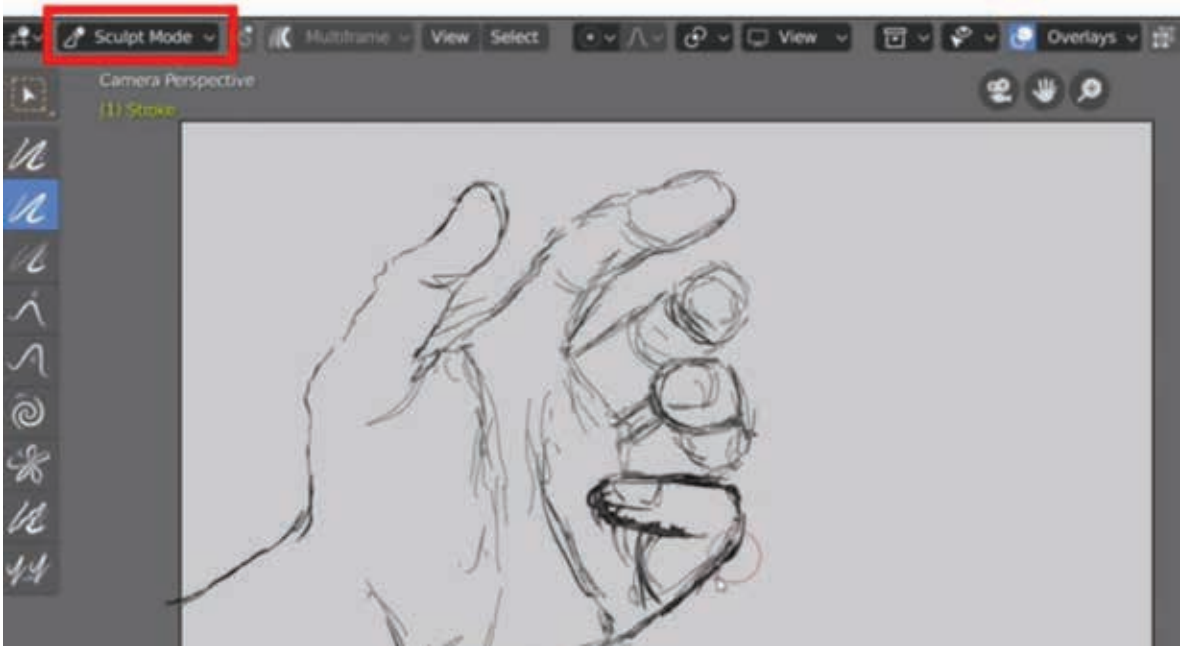
Then **press the Spacebar** to play the animation... you will see your sketching process played back. Awesome!

The Build modifier in action!

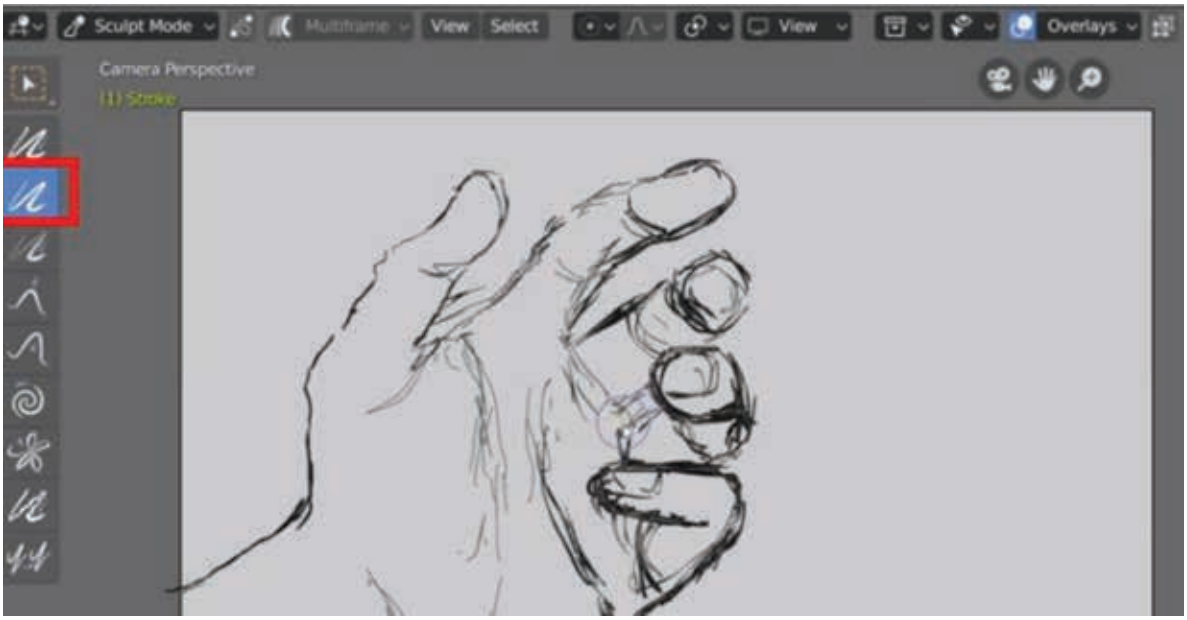
This is great for making for example "whiteboard" animation. A lot of businesses need this kind of animation - it's an affordable type of explainer video, where you just see a black and white drawing (usually with some sketched text) being drawn. This one modifier in Blender could make you some money!

Grease Pencil Sculpt Stroke Thickness

Grease Pencil has a **Sculpt Mode** that allows you to change the already drawn strokes in a number of ways.



The **Thickness** brush makes the strokes thicker.

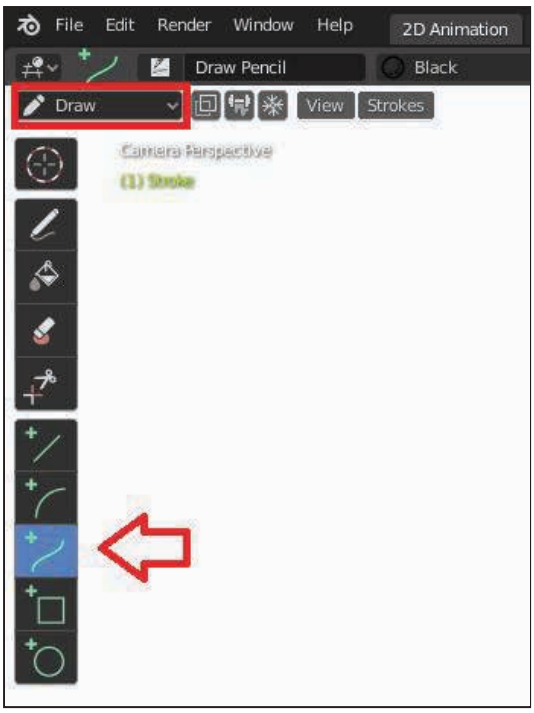


Holding Ctrl while using this brush will make thick strokes thin again.

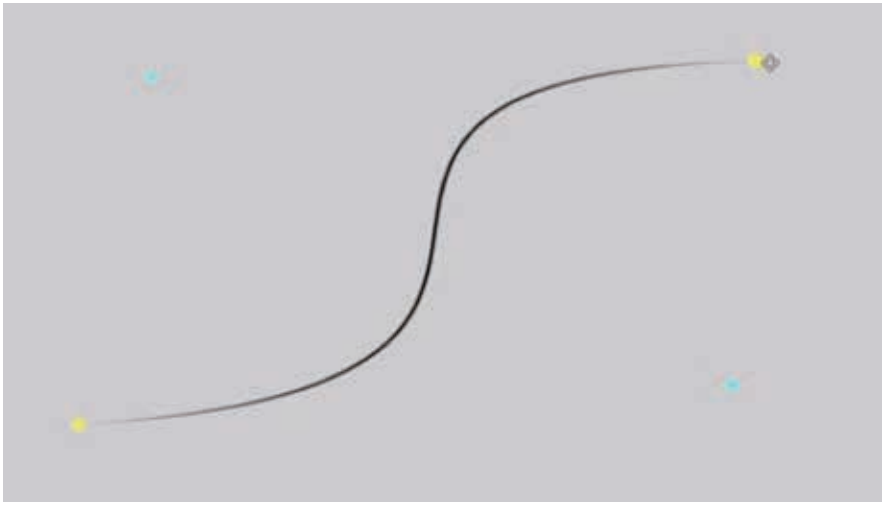
Curve Tool in Grease Pencil

To draw beautiful strokes, practice makes perfect. Spend many years drawing and you'll be really good at it! Or, you can cheat and use the Curve tool in Grease Pencil. It won't be as fast, but it gives you a lot of control over the exact shape of the curve, a bit like in a vector drawing program like Inkscape or Illustrator.

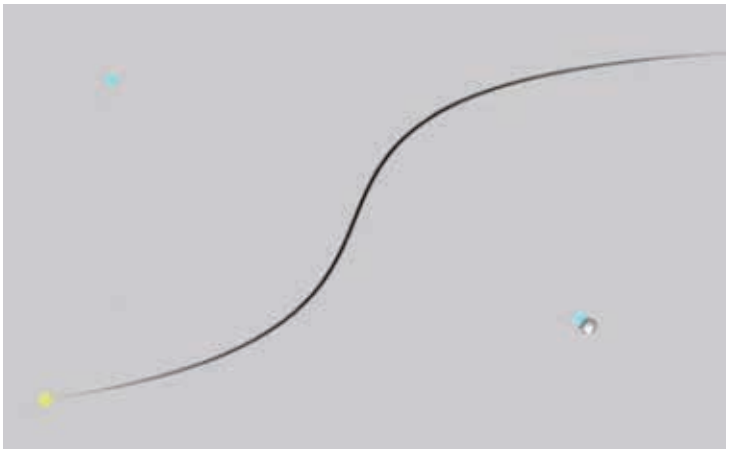
The **Curve tool** can be found in the **Tool palette** in **Draw mode**.



After drawing a curve, use the colored dots to manipulate it.

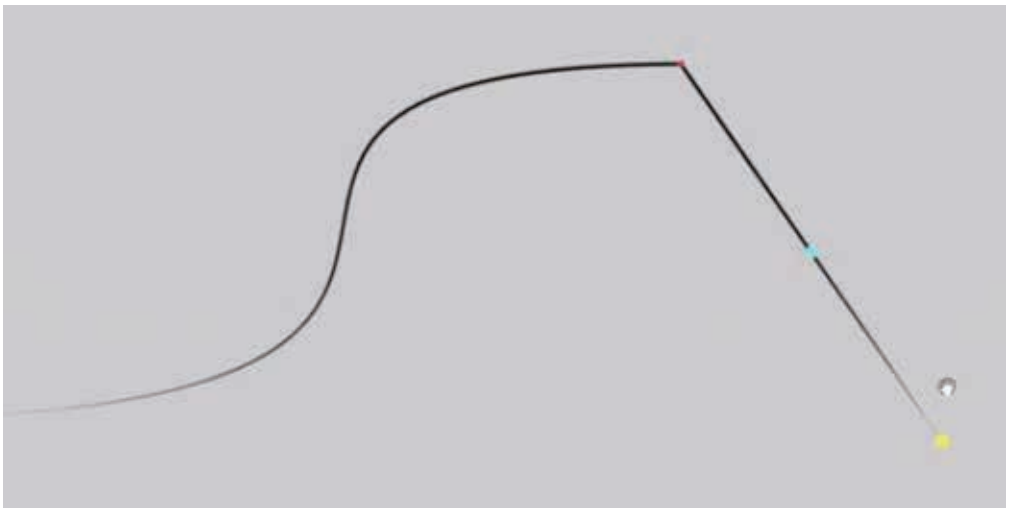


The blue dots change the bend in the curve.

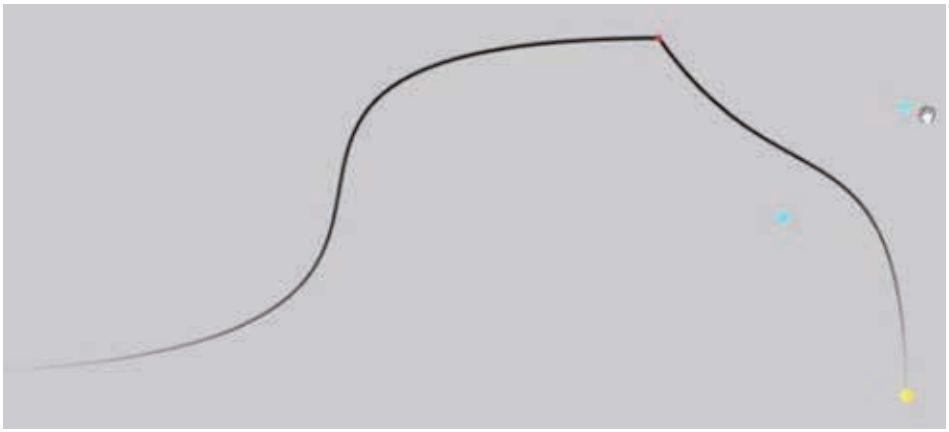


The yellow dots move the end points of the curve.

Press E and click somewhere to extrude from the end of the curve.

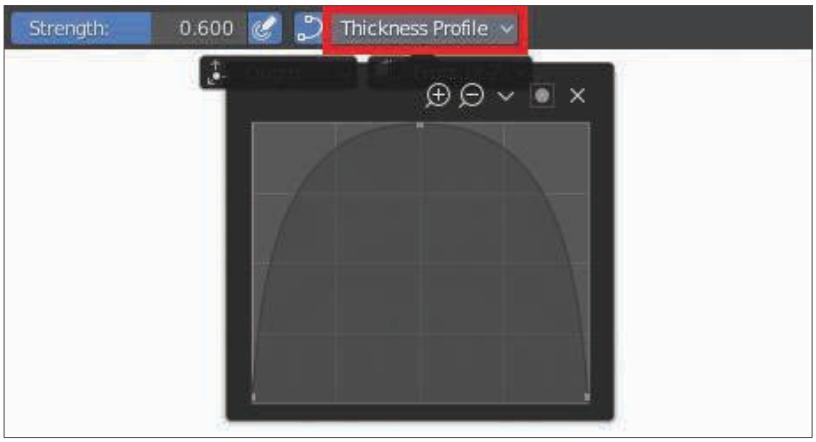


And then manipulate the extruded curve with the blue and yellow dots.

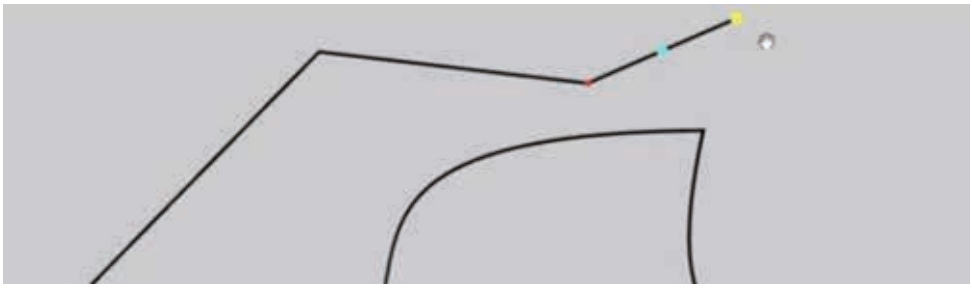


You can use the **Thickness Profile** like I have in this example, to make the stroke taper off towards the ends and have thickness in the middle. To turn this on, click on the icon next to “Thickness Profile”.

Then you can manipulate the profile manually by opening the drop-down by clicking on “Thickness Profile”.



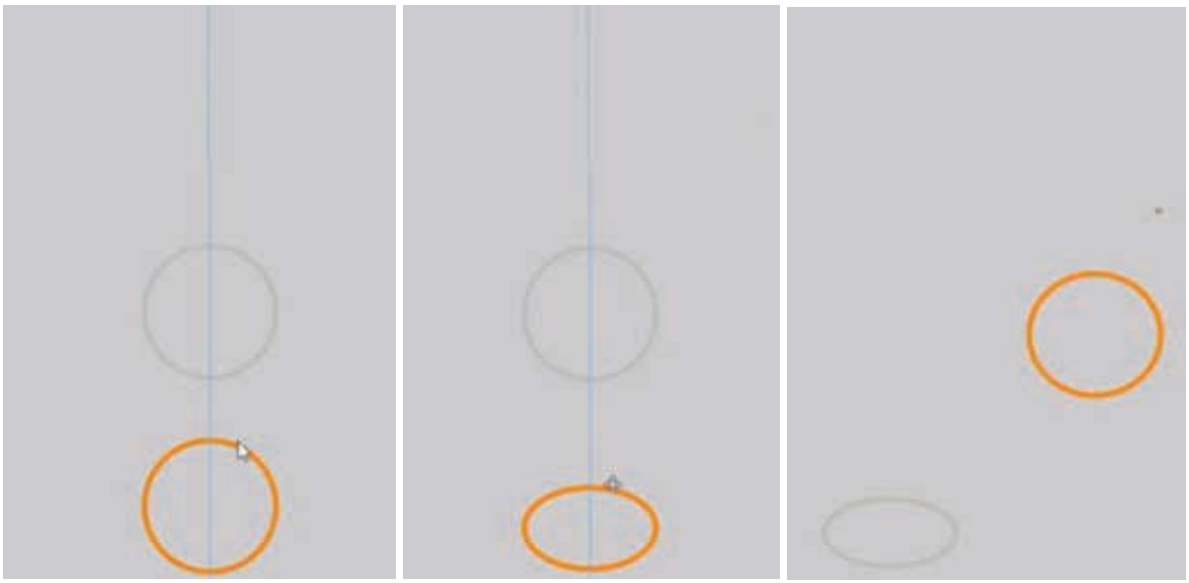
To draw straight lines hold Shift while drawing.



Grease Pencil Stroke Interpolation

Grease Pencil has a way to interpolate strokes. What that means is, it will automatically create one or more “inbetween” frames between two keyframes.

In this example I'm moving the stroke (a circle in this case) in Edit Mode three times. Each time a bit further along on the Timeline. First, I just move it down. Then I scale it on the Z-axis to squash it vertically. Then I move it to the right and scale it back to its normal circular shape. These are the very basic keyframes of a ball bouncing on the floor and bouncing back up to the right.

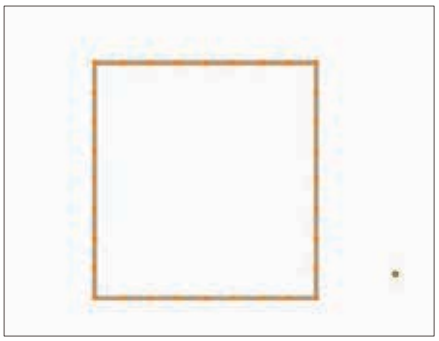


This has created three keyframes because Blender automatically records keyframes when you transform Grease Pencil objects.

Go to Draw > Strokes > Interpolate > **Sequence**.

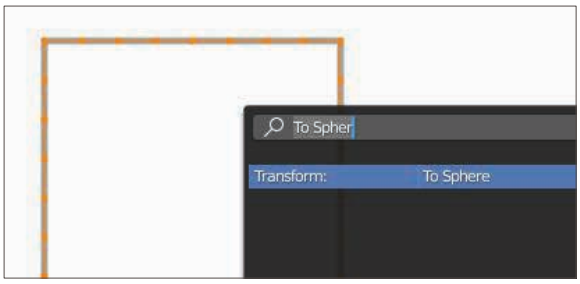
Now, Blender has created a sequence of interpolated in-between drawings. Magic!

For another example, I have this box-shaped Stroke.

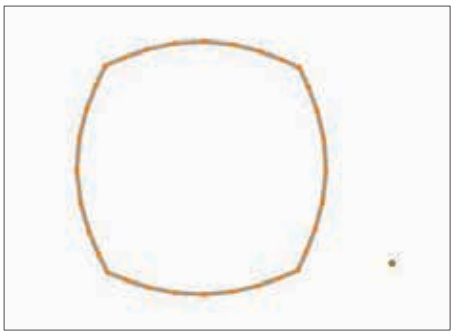


On a later frame on the Timeline, I make it spherical.

I do this by searching for the **To Sphere** command. **Press F3** and search "to sphere" and it will show up.

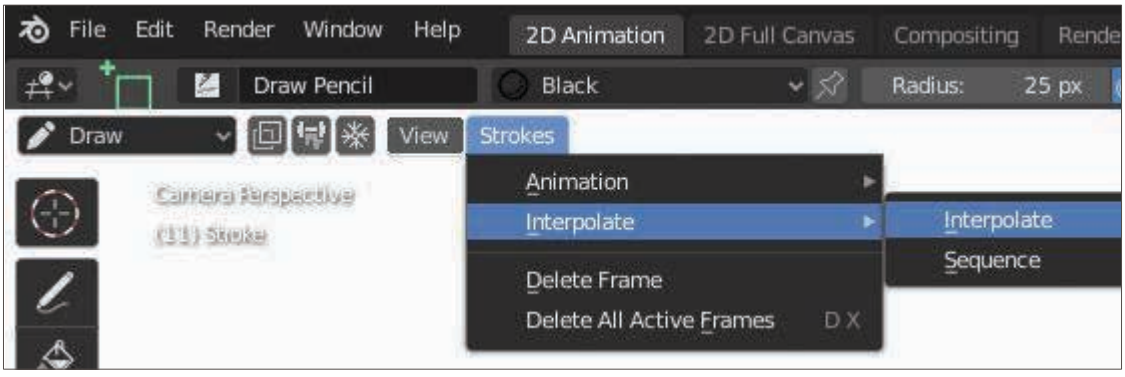


Drag the mouse to change the shape to spherical (This has to be done in Edit Mode).



As Blender Automatically records a keyframe for transformations in Edit Mode, we have two keyframes now for our Stroke.

Go to a point on the Timeline between both keys. In **Draw mode**, go to Strokes > Interpolate > **Interpolate**. This creates a single in-between keyframe (instead of a whole sequence).



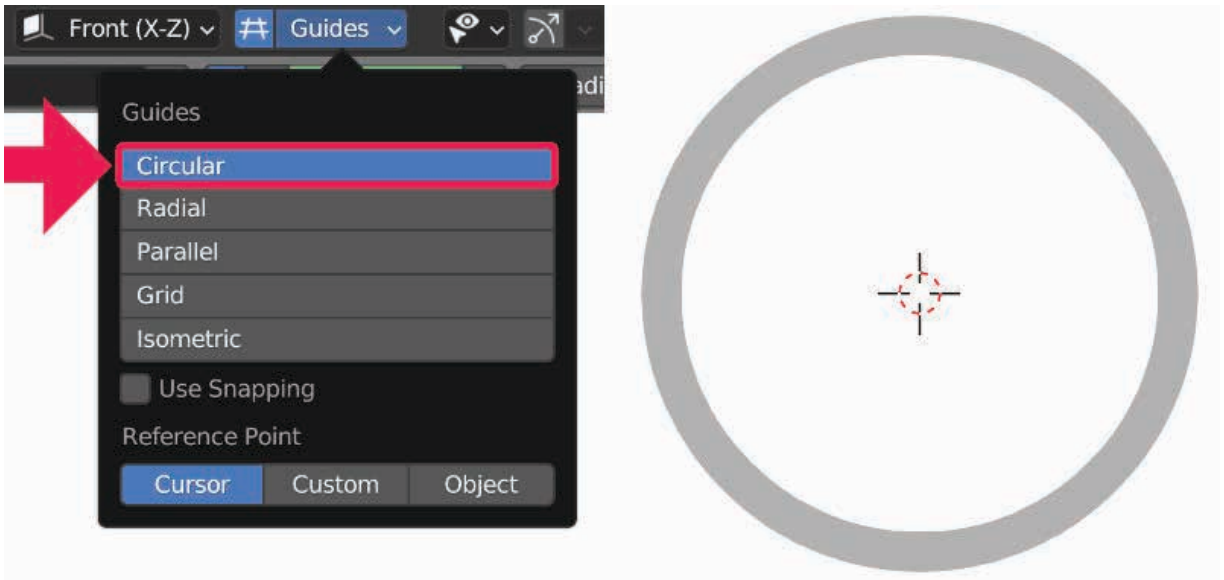
Grease pencil guides

This is a Grease Pencil circle drawn with a mouse.

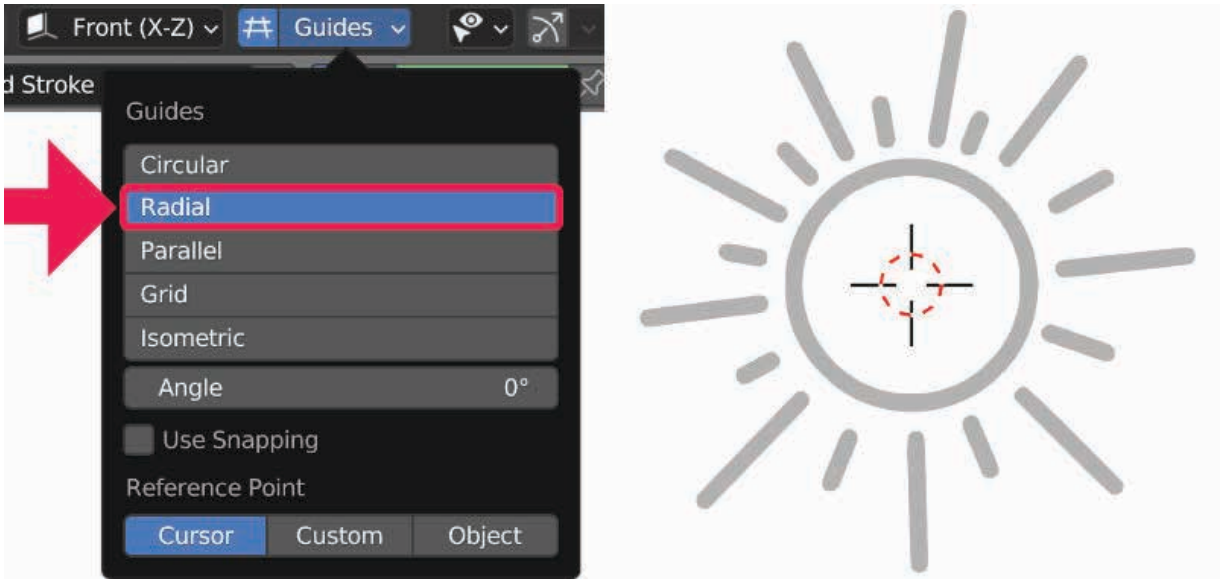
It's pretty bad!

Luckily there are some tools that make it easy to draw perfec circles..

Grease Pencil has several "guides". The **Circular** guide works around the **3D cursor** position to help you create a perfect circular stroke, even with a mouse!



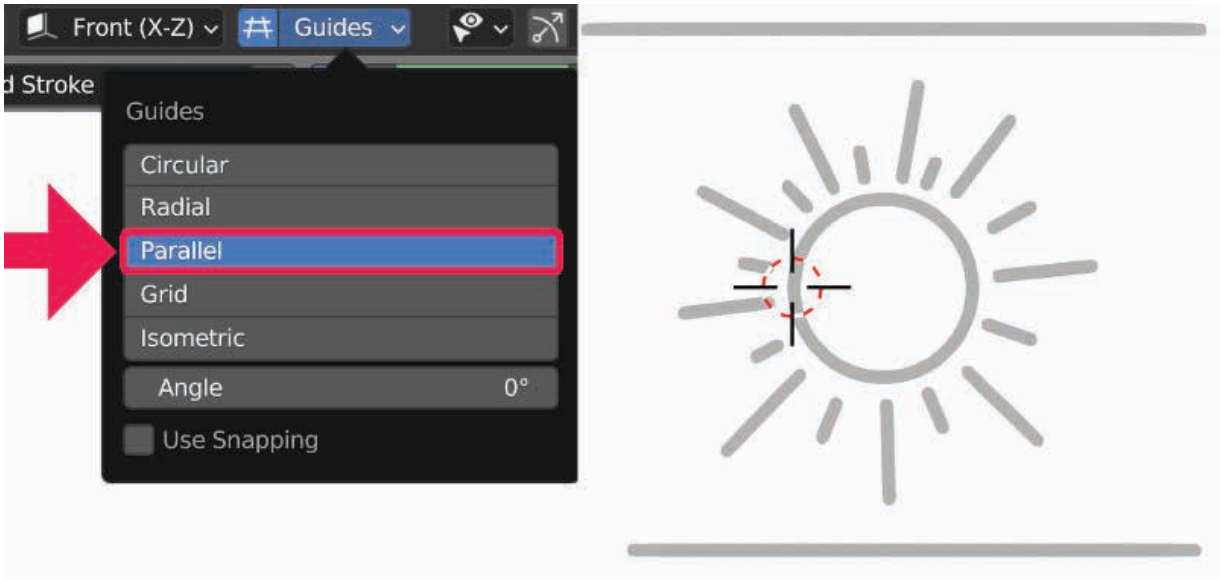
There is also a **Radial** guide.
It guides the strokes in a radius away from the **3D cursor**.



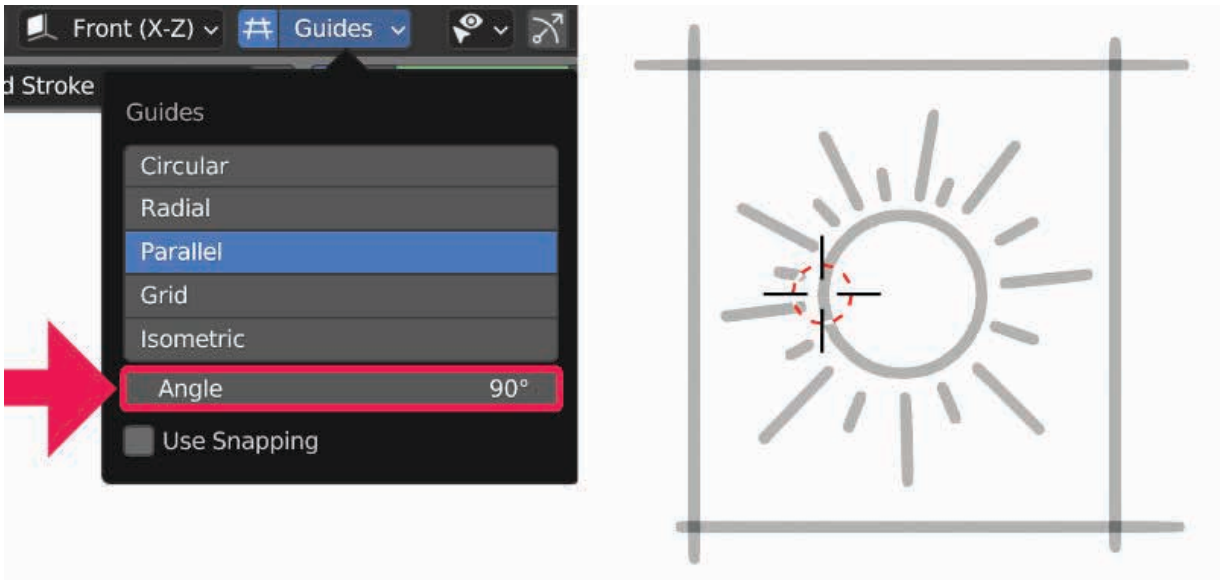
Click here to return to the index.

Place the **3D cursor** in a different location by **Holding Shift + right-clicking**.

There is also the **Parallel** guides option, to draw straight lines at an angle, for which you can set a value.



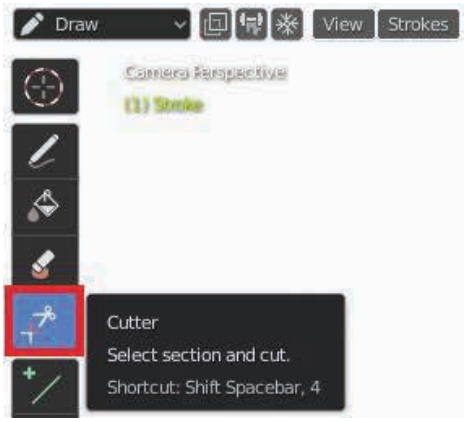
0° is horizontal, 90° is vertical.



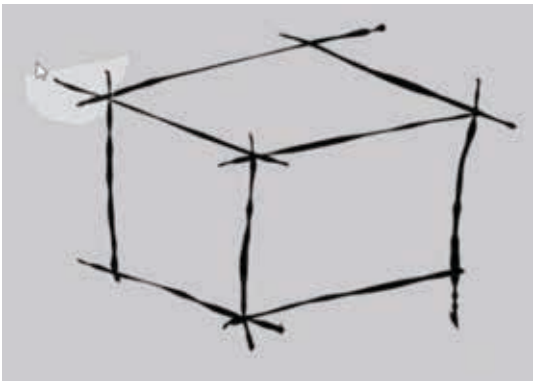
Finaly, the **Grid** guides let you draw horizontal and vertical lines.

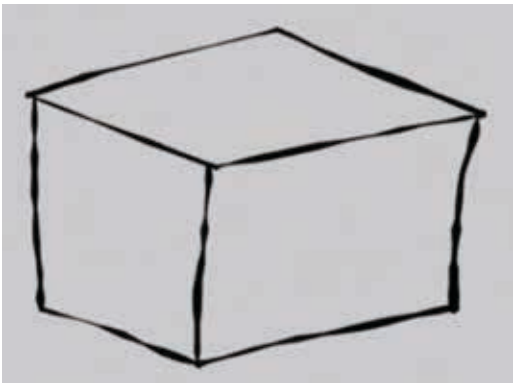
Grease pencil cutter tool

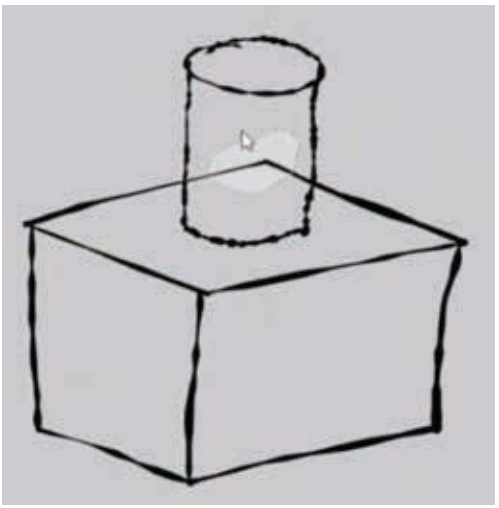
In Grease Pencil you can use the Cutter tool to cut off parts of strokes that you don't need. You can find it under the eraser tool.

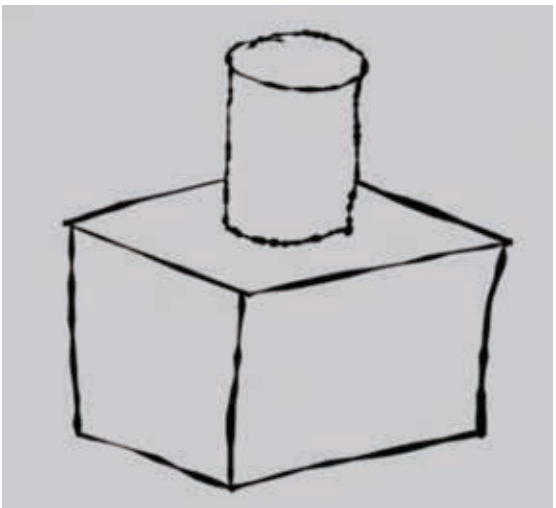


To use it, you draw a selection around the part you want to cut with the Cutter tool active. The selection doesn't have to be perfect, it's pretty clever at guessing what you want to cut.





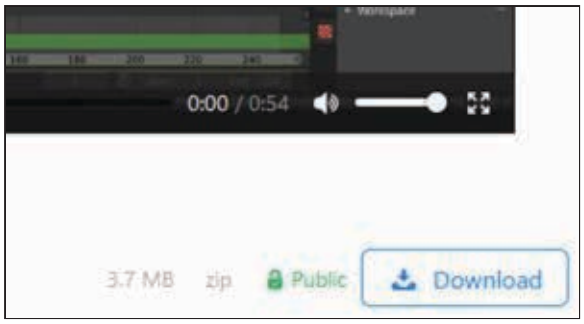




Installing textured brushes

On the Blender Cloud website you can download these brushes made by Daniel Martínez Lara following this link.

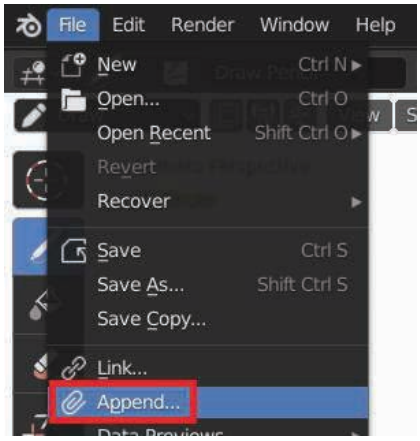
Just scroll down until you find the Download button. This downloads a Blend file.



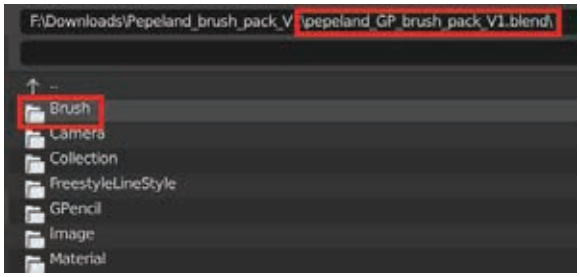
By default you only have these GreasePencil brushes.

Go to File>**Append**.

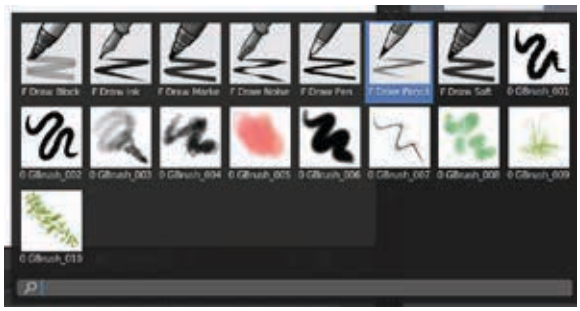
Search for the file Pepeland_GP_brush_pack_V1.blend.



And append the Brushes "GBrush_001" to "GBrush_010".



Now you have these new textured brushes.



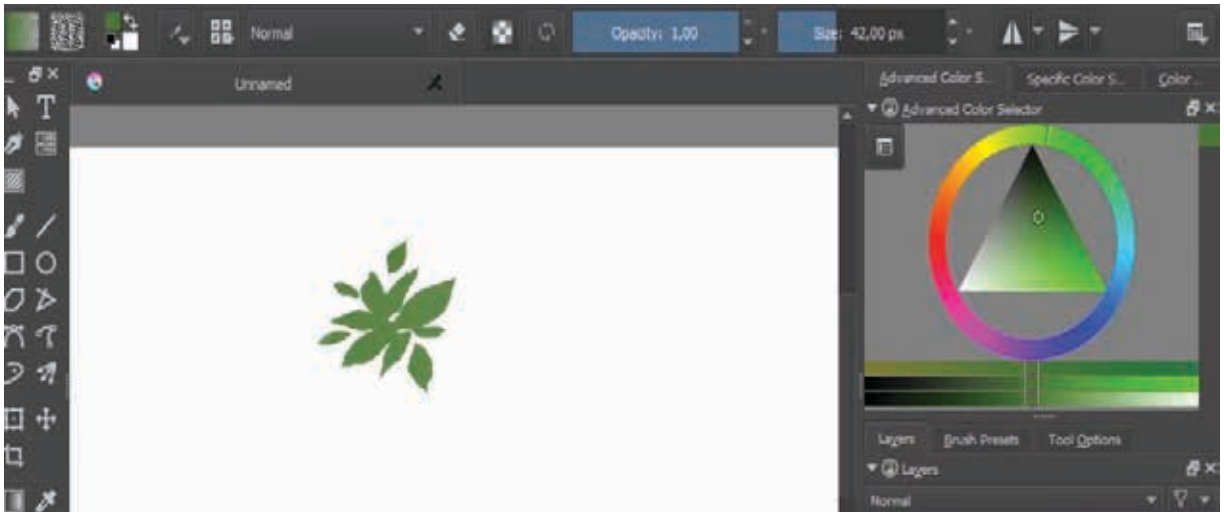
And you can paint with them!

If you don't see the texture, make sure you are in **LookDev** or Rendered viewport shading mode.

Creating custom textured brushes

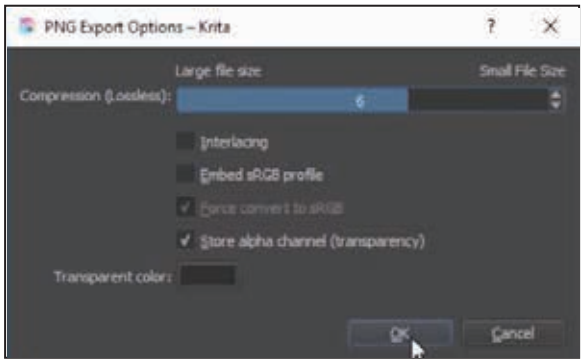
Paint a new texture in for example Krita or Gimp.
I like to draw leaves by using a lasso selection and then filling it with a color.

Much faster and easier than painting. This is Krita, it's a free and easy to use painting program.

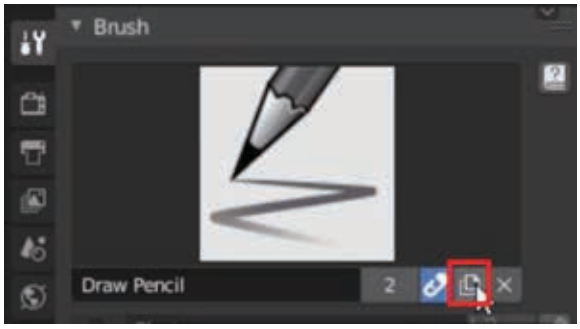


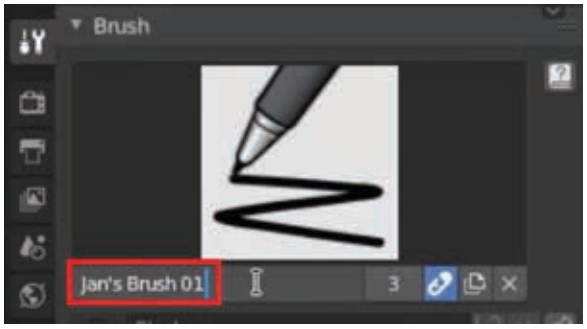
Crop it and make sure it has a transparent background.

Save as PNG with alpha.



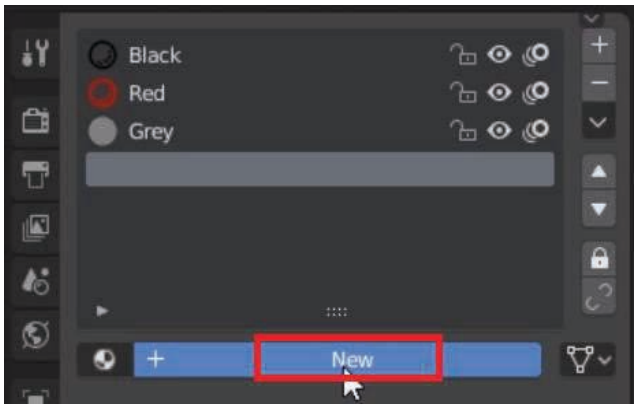
In Blender create a new brush and give it a name.



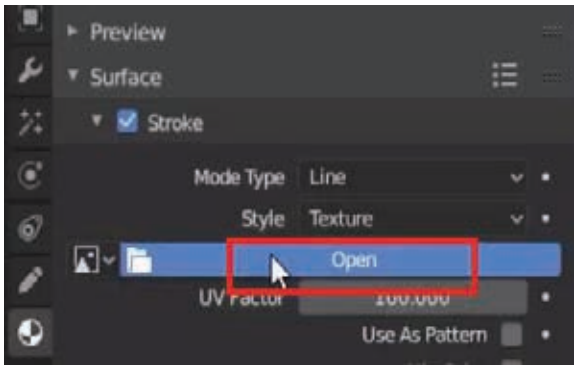


In the Material tab add a material slot by pressing the **+ icon**.

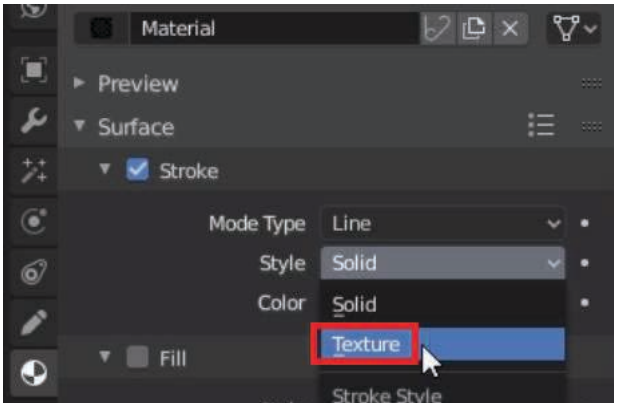
Click on **New,** set Stroke style to Textured.



Click on **Open** and choose your texture file.

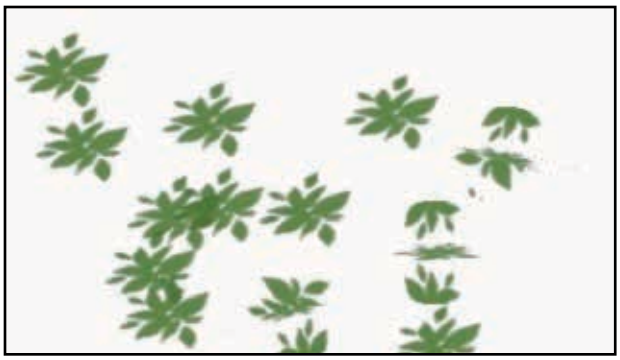


Set viewport color to **Textured**.



You may need to increase the brush size (**press F** and drag to change it).

Now you can use your brush!

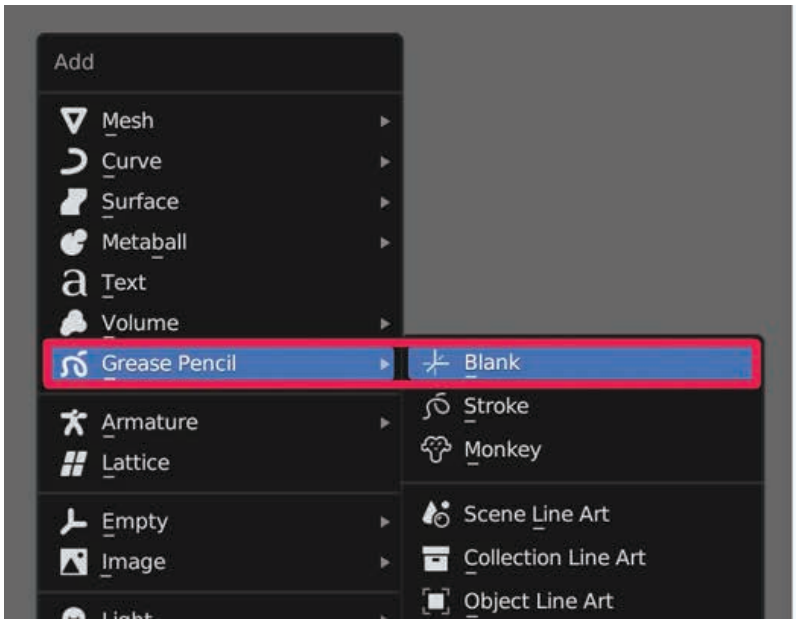


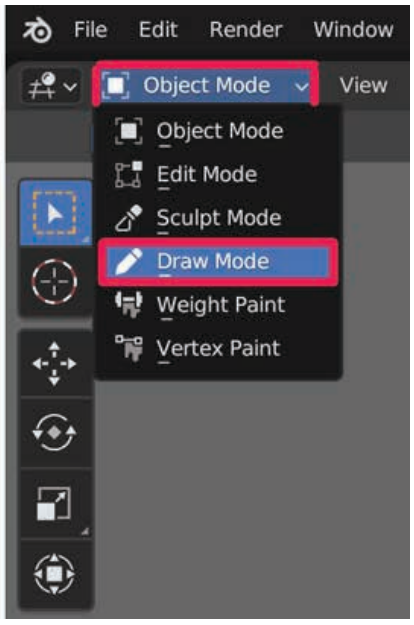
If you don't see the texture, make sure you are in **LookDev** or Rendered viewport shading mode.

Draw Grease Pencil On Surfaces without offset distance issues

Grease Pencil can be a good tool to draw designs on top of a surface. But if you zoom the 3D view in and out, the offset distance between the strokes and the surface changes, which can be annoying and unwanted. Here we look at how to draw on surfaces without changes in the offset distance.

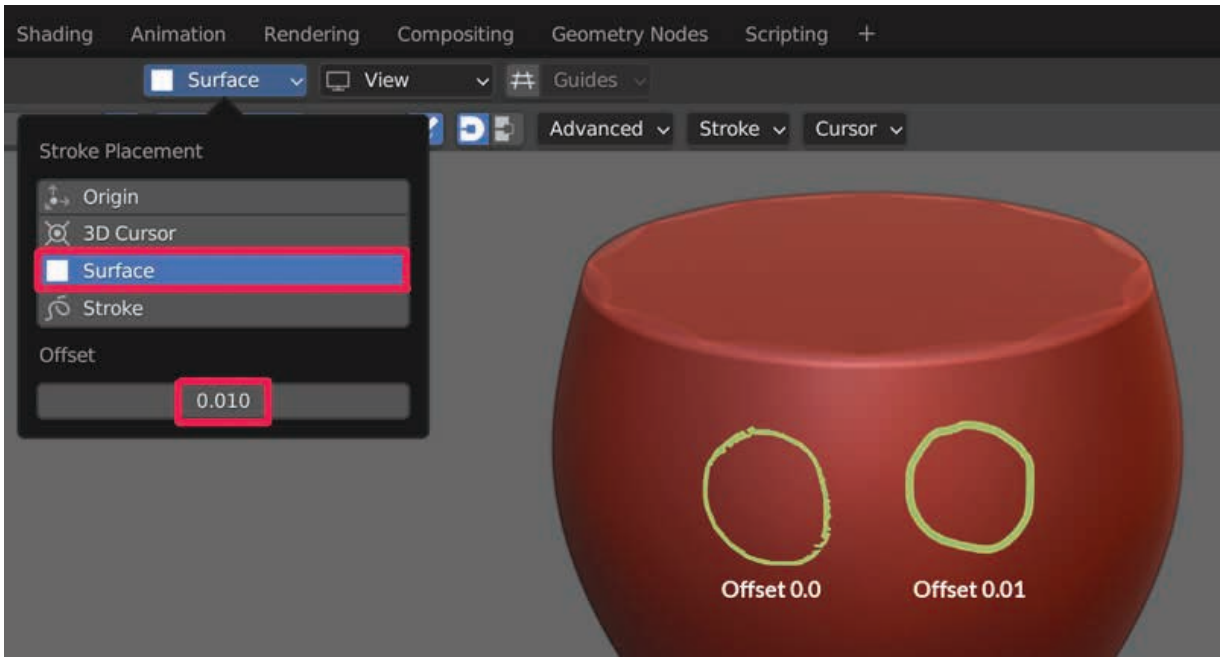
To draw on top of an object using Grease Pencil, add a **Blank Grease Pencil Object** in Object Mode. Once you have a Grease Pencil object you can switch to **Draw Mode**.



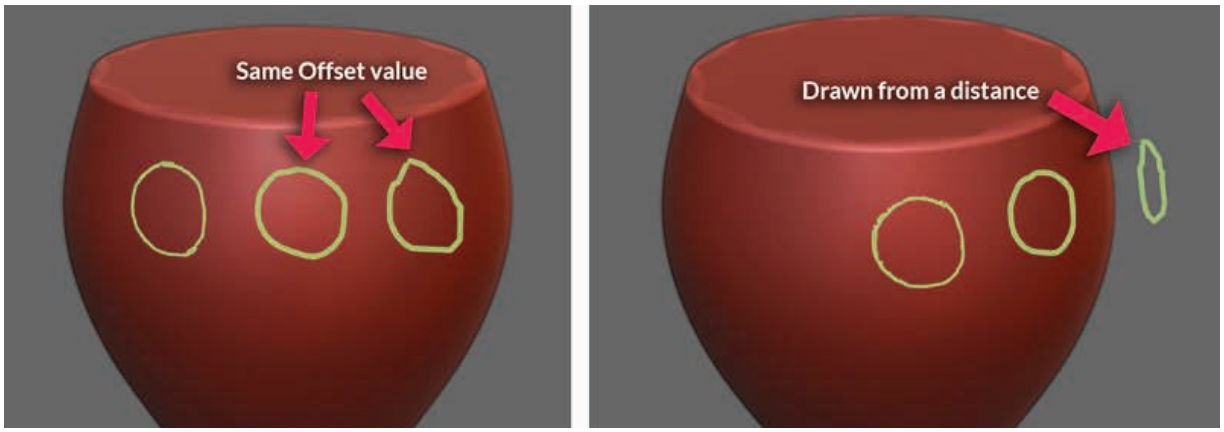


To see the stroke more clearly you can change the **Base Color** of the Grease Pencil material.

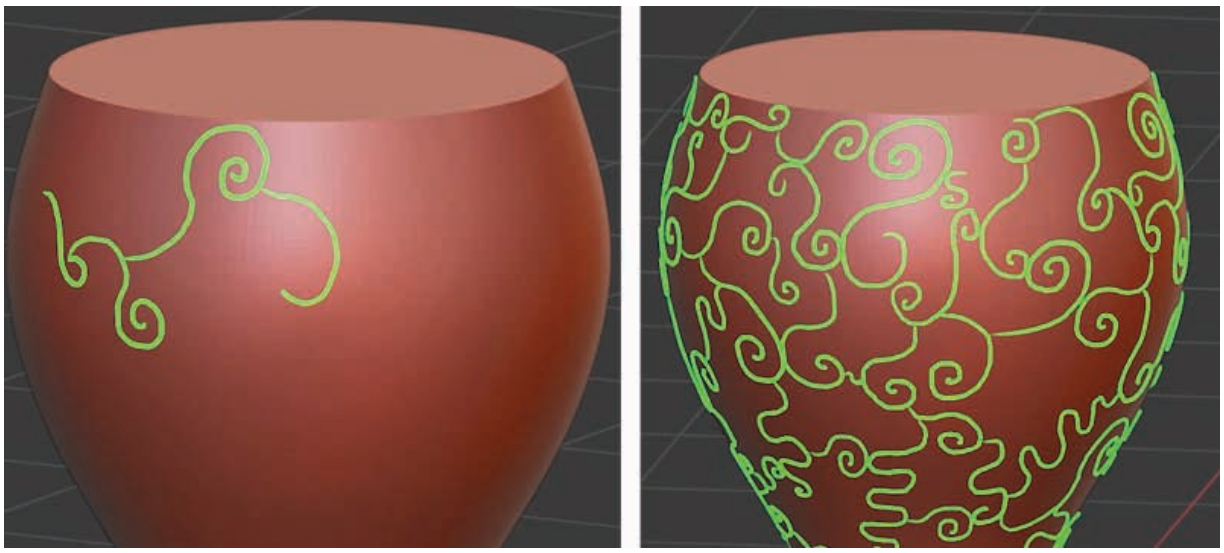
Set **Stroke Placement** to **Surface**. The Offset Value determines the distance between the Strokes and the Surface. If the value is zero your strokes will overlap with the object. A small value like 0.01 keeps it just above the surface.



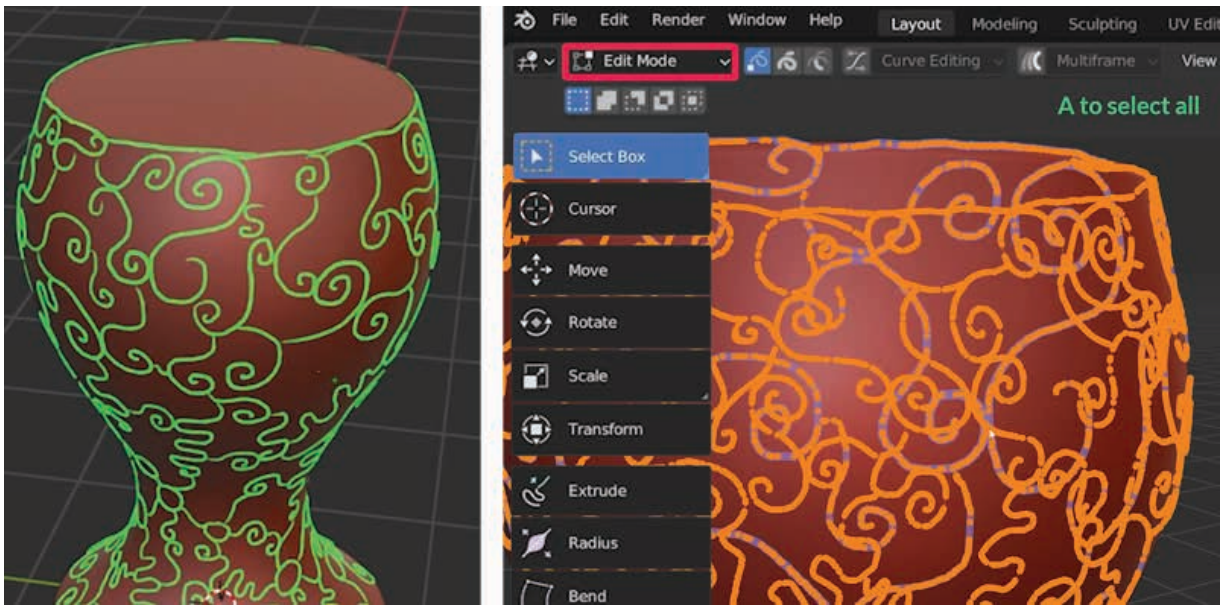
The Offset distance is also influenced by zooming in and out on the object.



To avoid this, **press 5 on the Numpad**. This toggles between Perspective and Orthographic view. Now you can draw without any offset distance issues. In **Orthographic view**, zooming in and out has no influence on the Offset distance.

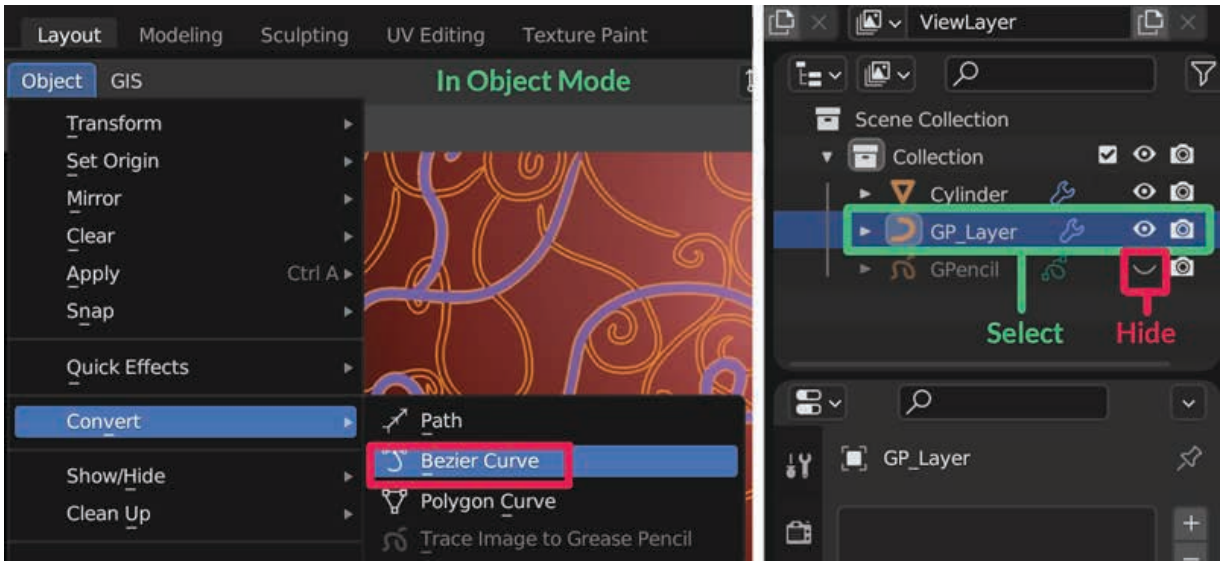


Press 5 again when you want to toggle back to Perspective view. **Press Tab** for Edit Mode and **press A** to select all the vertices.



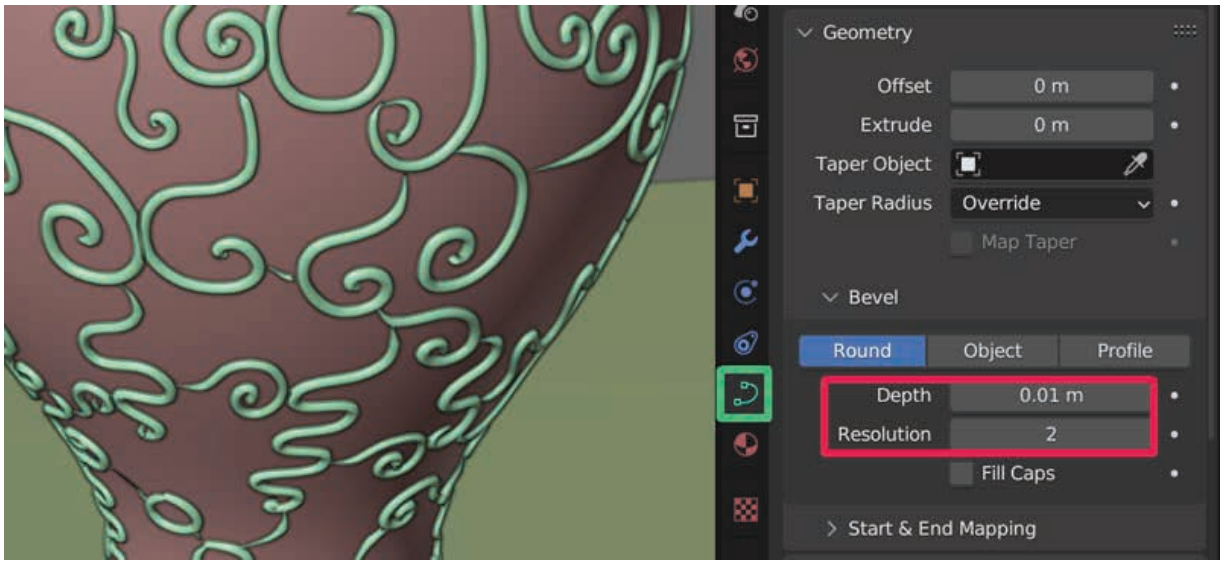
After drawing strokes on the surface, you can make the strokes more smooth by right-clicking and choosing **Smooth**. Repeat it a few times with **Shift+R** until it looks smooth enough. You can also use the Smooth Stroke pop-up menu to adjust the Repeat value.

You can turn the Grease Pencil object into Curves. Go to **Object > Convert > Bezier Curve** (in Object Mode). Disable the visibility of the GPencil object in the Outliner. That is the original Grease Pencil drawing you made. Select the newly created Curve, which is called GP_Layer.



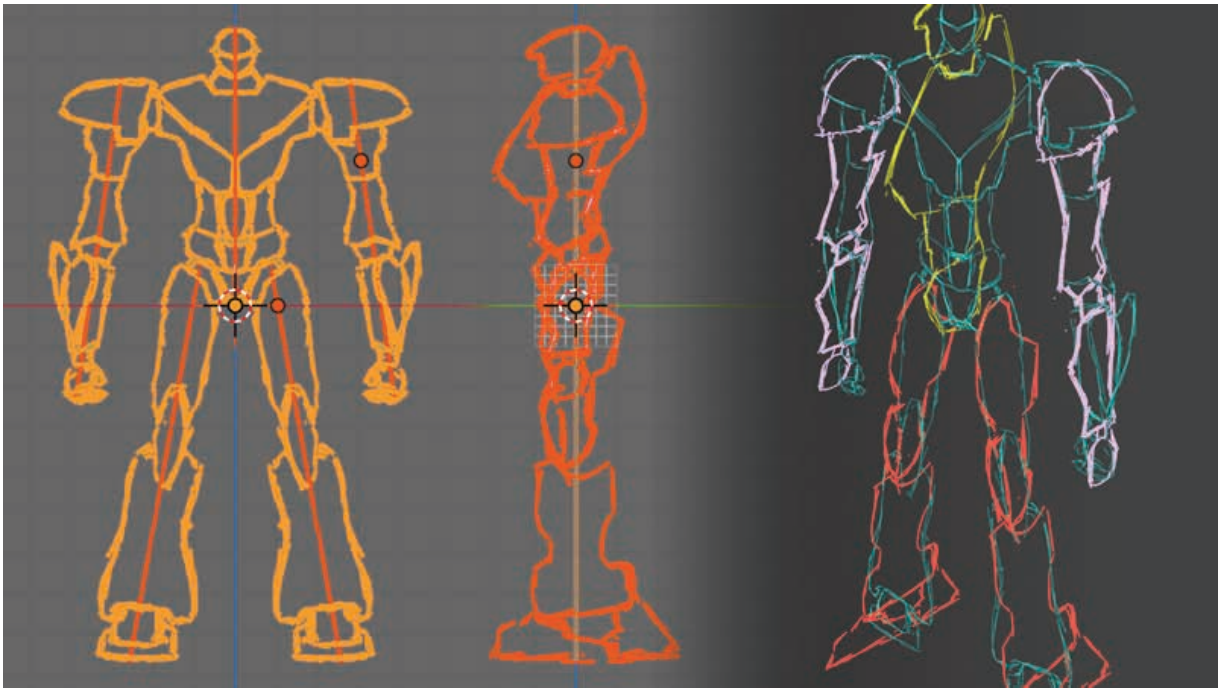
In Edit Mode, right-click again and choose **Decimate Curve** to reduce the complexity. I usually decimate to around 10 percent or 0.1.

Finally add thickness to the Curve by increasing the **Depth** value and increase its **Resolution** to make it look better.

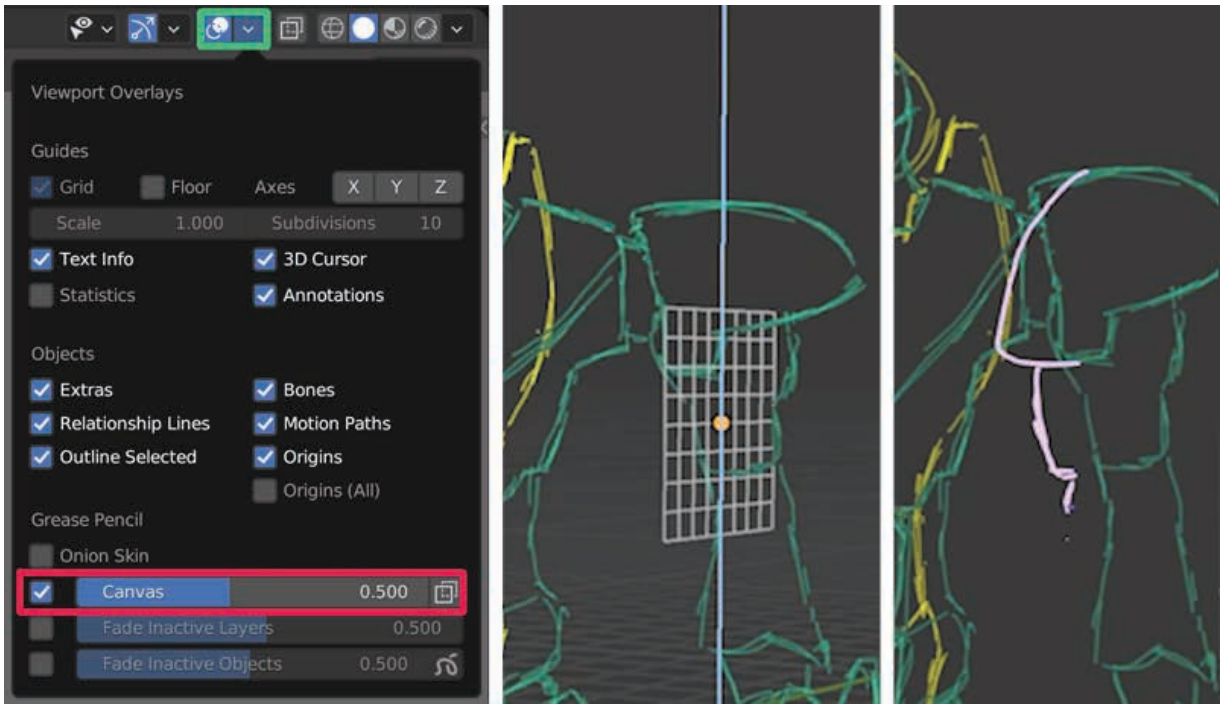


How to use the Grease Pencil Drawing Plane Canvas option

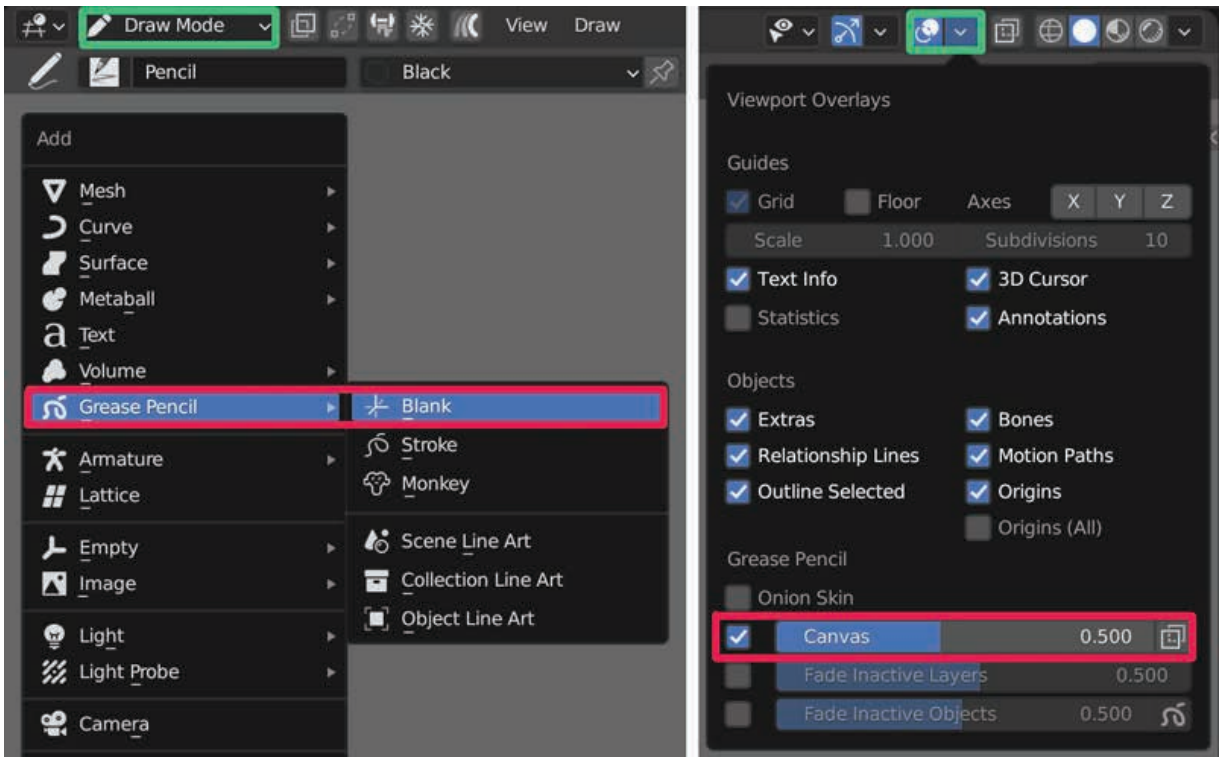
The Canvas is a visual aid that helps you see the Drawing Plane on which your Grease Pencil strokes will be placed.



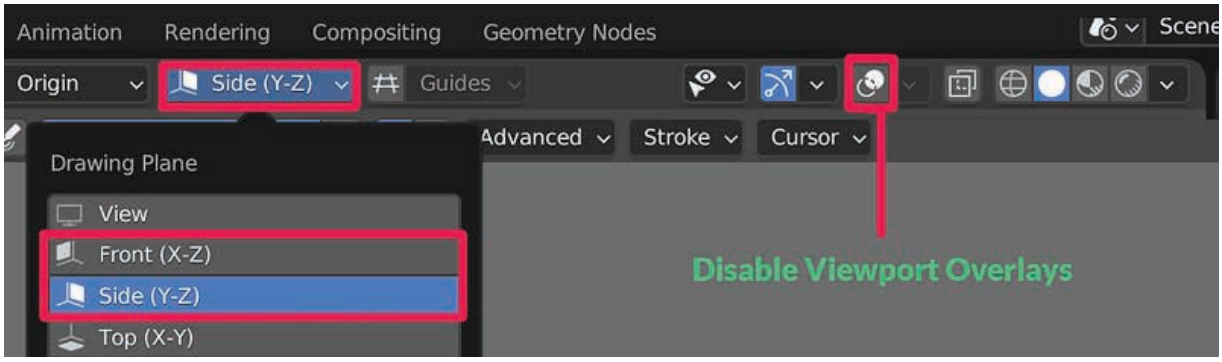
By moving and rotating the Drawing Plane you can decide where to draw your strokes.



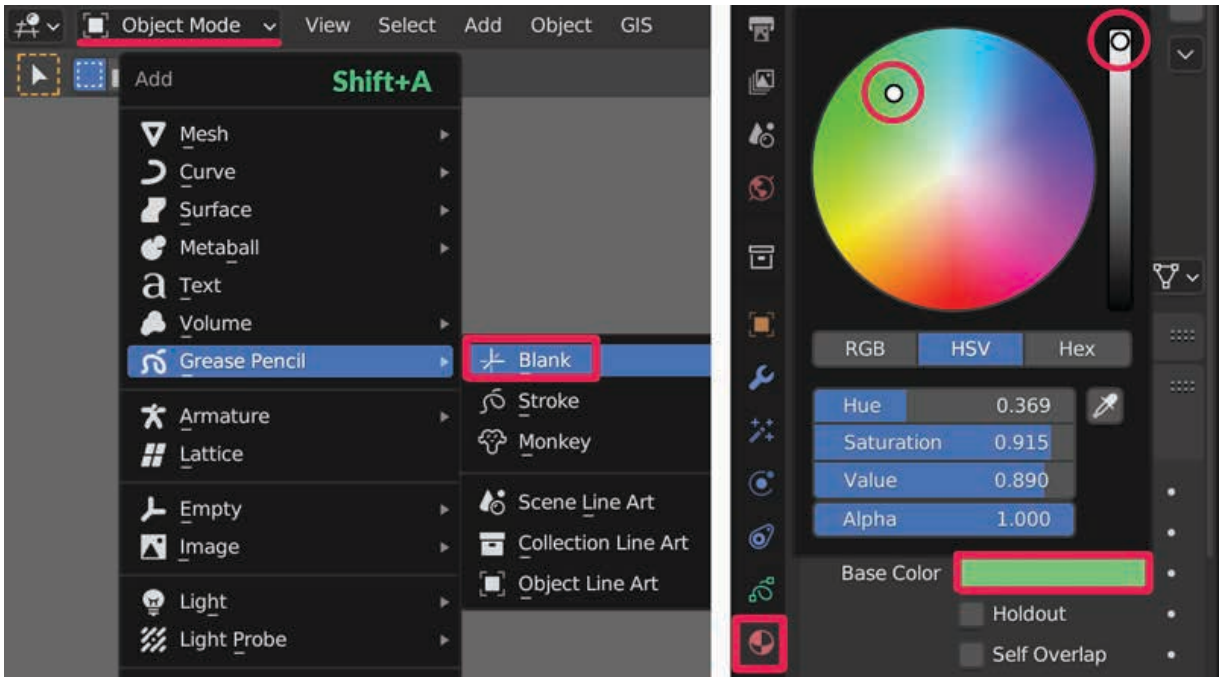
To use it, create a Blank Grease Pencil object. The Canvas checkbox is only available in Overlays when you are in Draw Mode. And Draw Mode is only available once you have at least one Grease Pencil object.



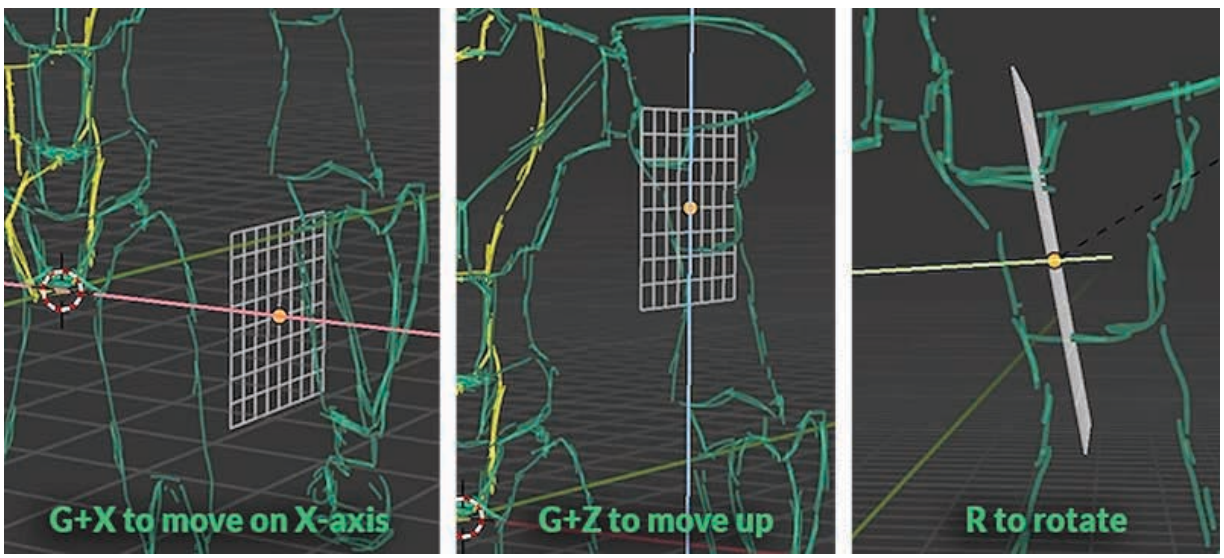
Set Drawing Plane to Front or Side. You can disable Overlays to reduce the visual distractions while drawing.



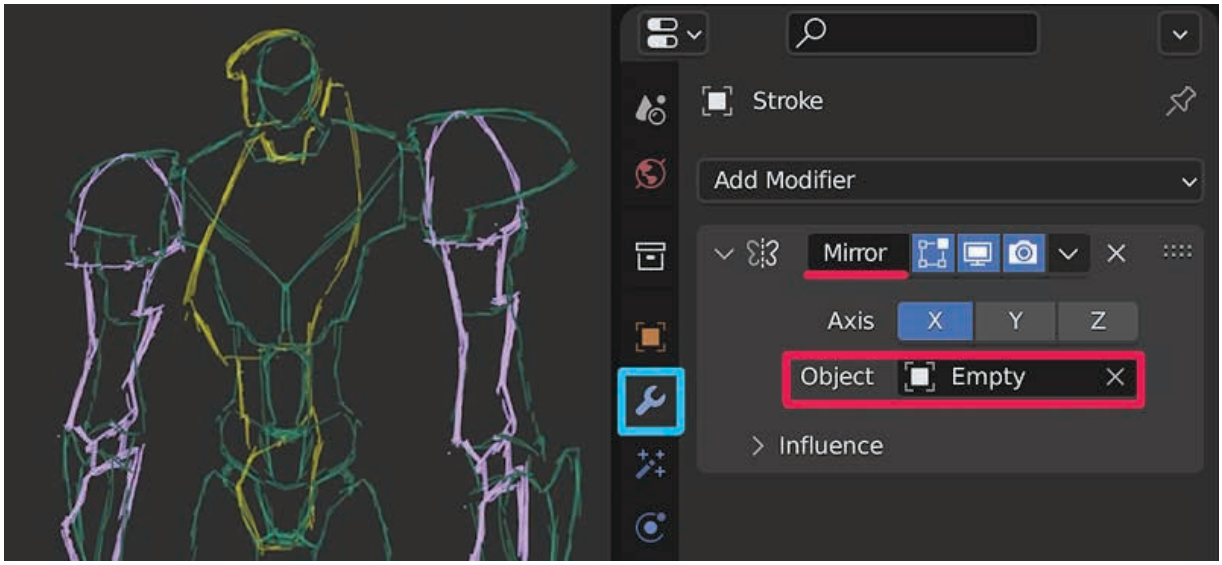
In Object Mode, add another Blank Grease Pencil object. To change the color, click on the Base Color field of the Grease Pencil Material. This will make it more distinct in the viewport so it's easier to see what you're doing.



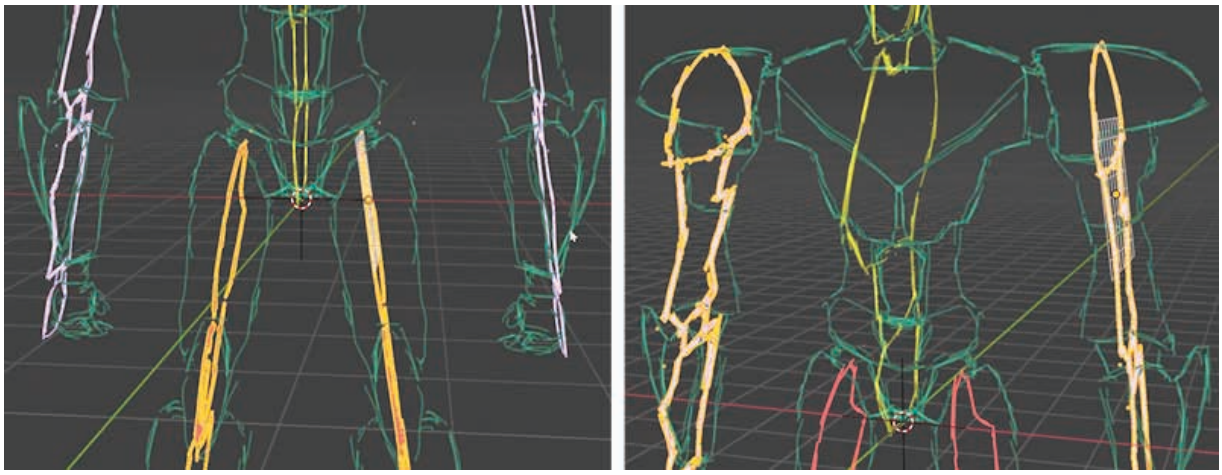
You can rotate and move it to change where you'll draw your strokes.



By using an Empty and a Mirror modifier, you can mirror the Grease Pencil strokes.



In Object Mode you can still select previously drawn Grease Pencil objects to rotate or move them.

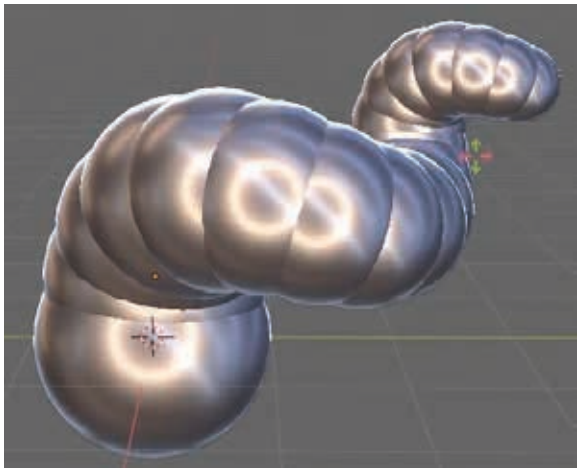


The Canvas overlay can be very useful when sketching a design in 3D, before you start 3D modeling.

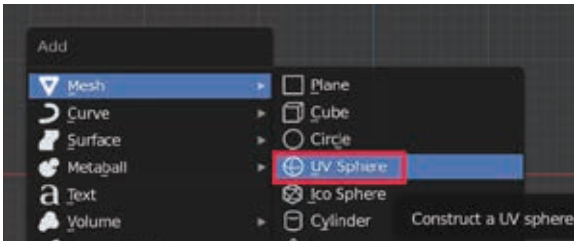
MODIFIERS

Array modifier with object offset

Using the **Array Modifier** you can create some interesting shapes.



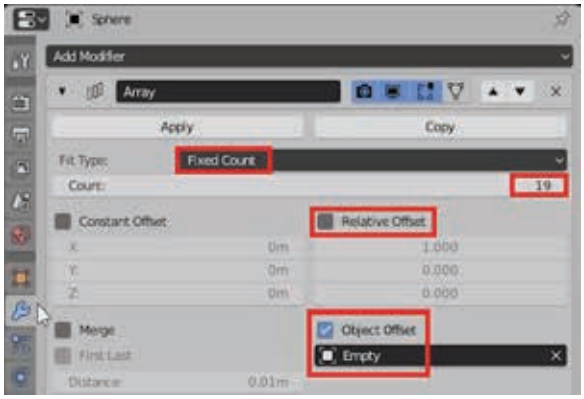
Press Shift+A and create a **UV Sphere**.



Add an **Array modifier** to the Sphere and increase the **Count**.

Turn off **Relative Offset**.

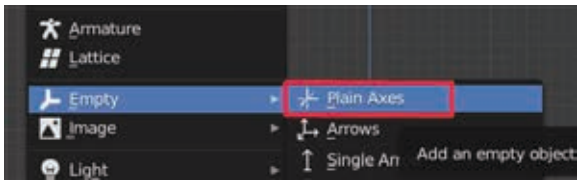
Turn on **Offset object** and select the Empty.



Press Shift+A and create an **Empty.**

In this case, I choose "**Plain Axes**", but it doesn't really matter what kind of Empty you use.

They all work the same but look differently, it just depends on what looks most clear to you.

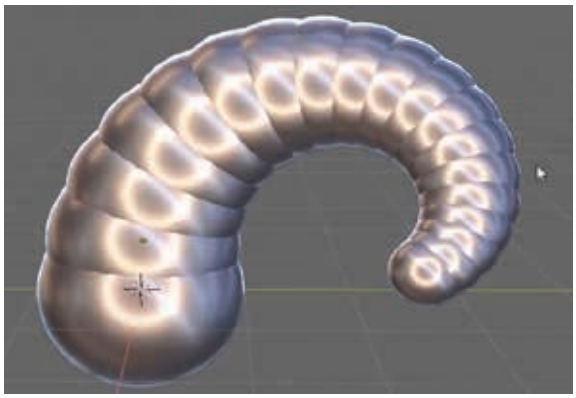


Right-click on the Sphere and set it to **Shade Smooth**.

Scaling, Rotating and Moving the Empty creates all kinds of shapes.

You can also keyframe animate the transformations of the Empty object.

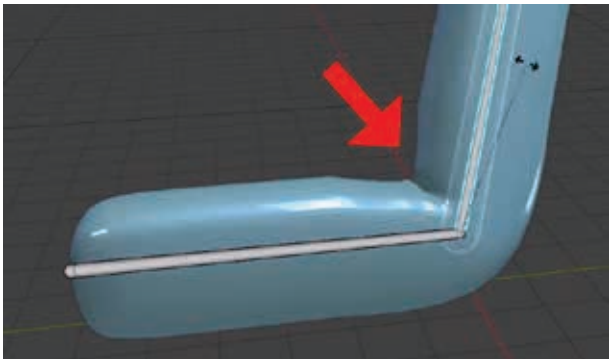
I'm using a MatCap here to make it shiny.



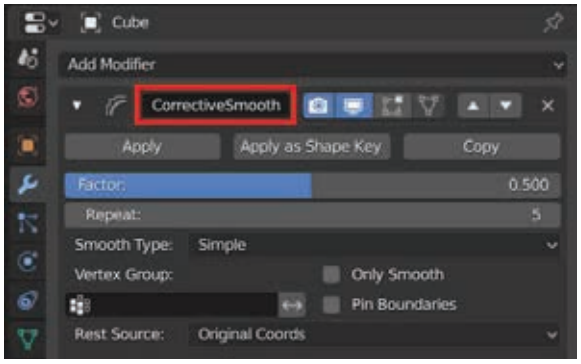
Click here to return to the index.

Corrective smooth modifier

Let's take a look at this simple rigged object. Note the ugly deformation.



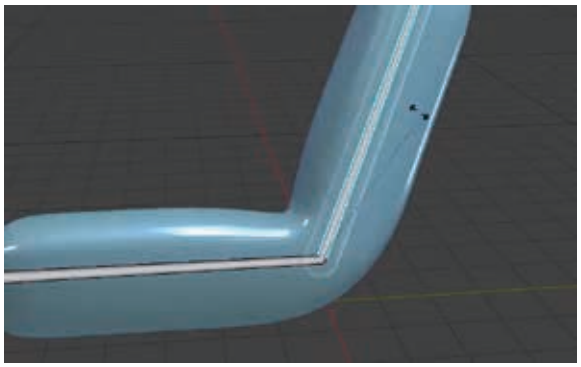
To solve it, in Object Mode add a **Smooth Corrective** modifier to the object.



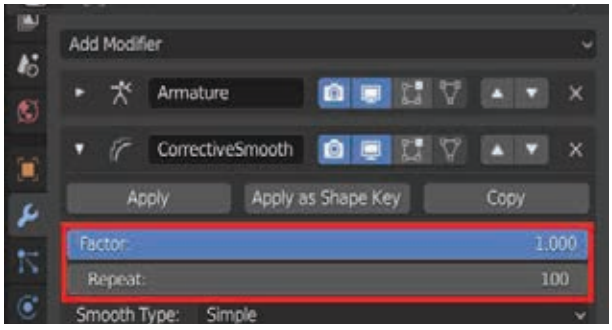
Make sure it is below the **Armature modifier**, otherwise it won't work correctly.

The **Corrective Smooth modifier** takes care of this deformation nicely.

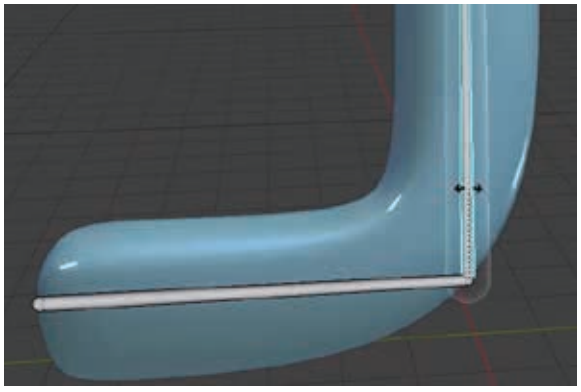
Especially when we increase the **Factor** value to 1.



Increasing the **Repeat** value makes it even smoother. Especially at a very high amount like 100.



It's almost like an easier to set up version of Bendy bones! Pretty nice...



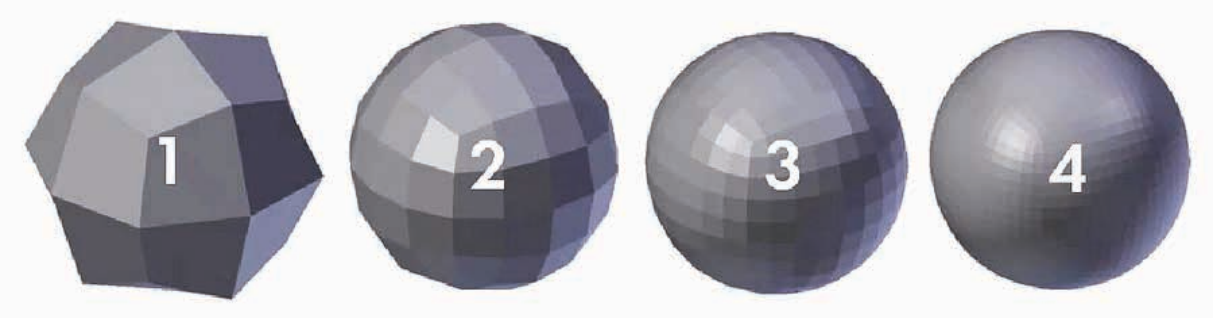
Toggle Subdivision Levels

To quickly add a subdivision modifier and set it to a specific level of subdivisions with one keyboard shortcut, **press Ctrl+0, 1, 2, 3, 4 or 5**.

For example: to add a subdivision modifier set to 2 levels of subdivisions, **press Ctrl+2** to immediately go to subdiv level 2.

Note: This works with the number keys above the alphabet keys, not the ones on the numpad.

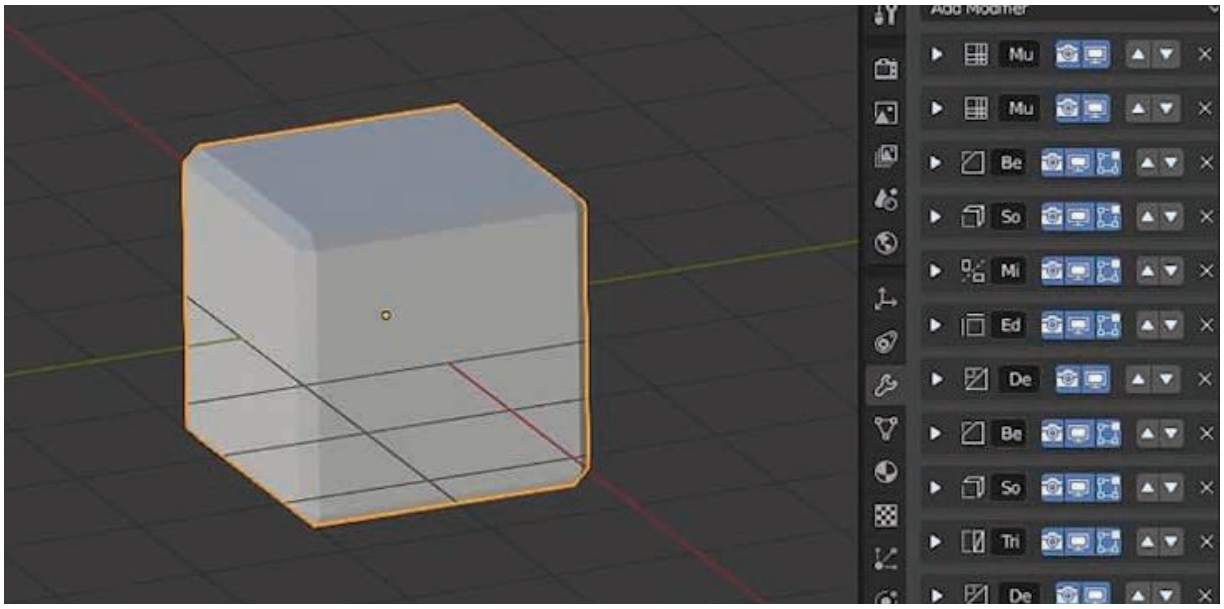
Once a subdivision modifier is applied, you can always change it to different subdiv levels as well, for example to turn it off **press Ctrl+0**.



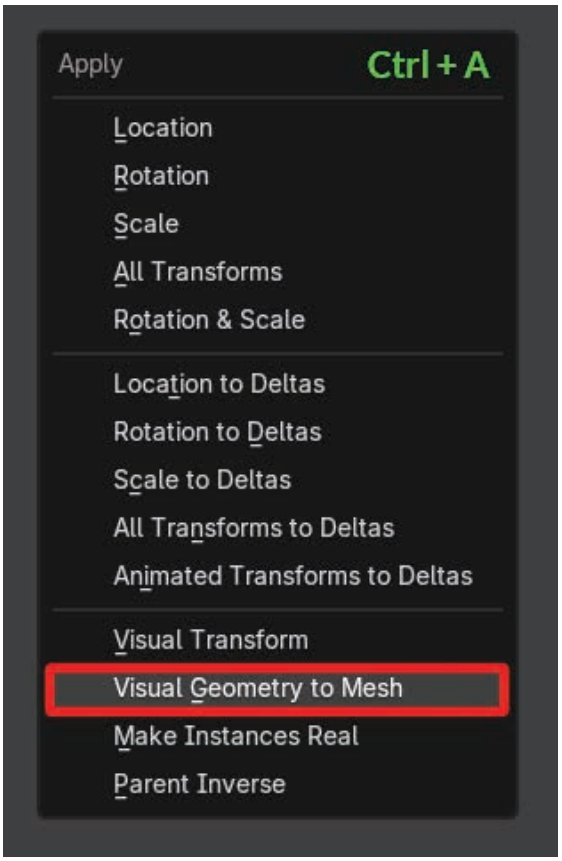
Different subdivision levels of a cube.

Apply All Modifiers At Once

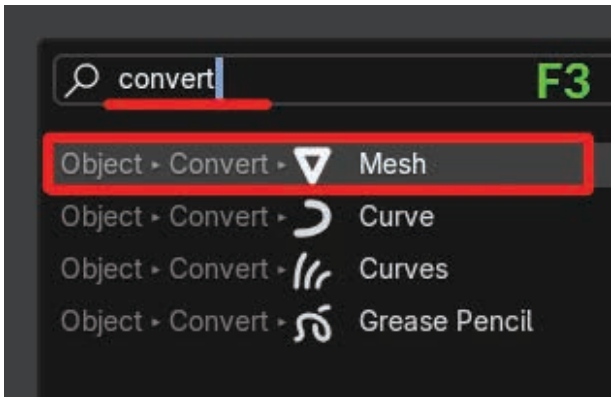
Look at all these modifiers. What if you need to apply them all, but don't want to hit "Apply" on each individual modifier?



Select the object in Object Mode. Press **Ctrl+A** and choose **Visual Geometry to Mesh**.



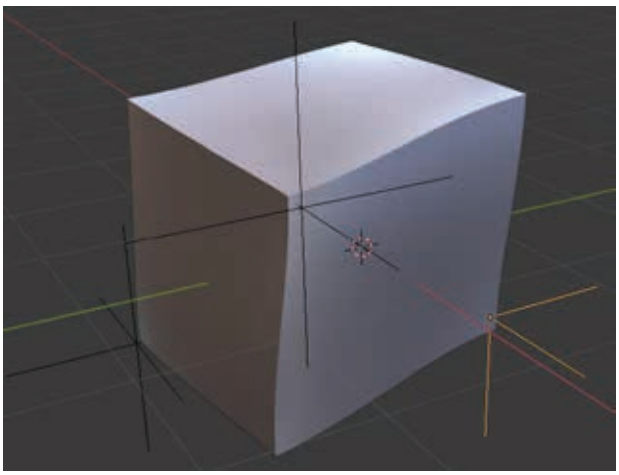
Another method, would involve **pressing F3** and typing "Convert", to get the **Convert to Mesh** option. This also applies all modifiers.



Another way would be to export it (as an OBJ file, for example) and checking the "Apply all modifiers" export option. That automatically applies the entire modifier stack as well.

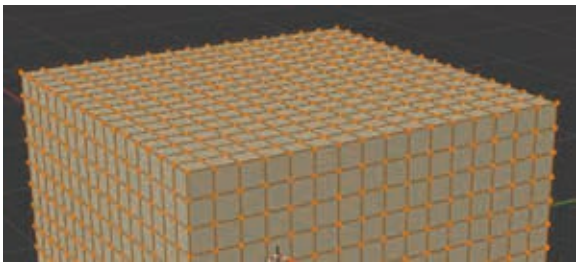
Add hooks and Laplacian Deformer

Hooks are an interesting way to control and deform objects.



For this example I've added some extra resolution to the default cube by Subdividing it 4 times in Edit Mode.

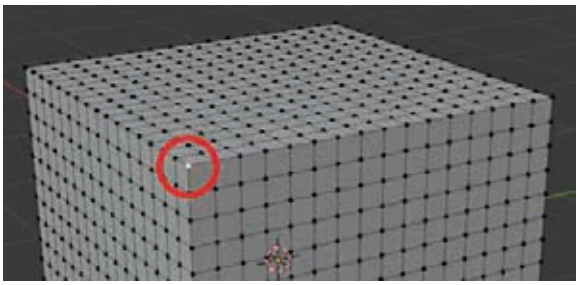
(to do this, **right-click, choose Subdivide** and then p**ress Shift+R** three times.)

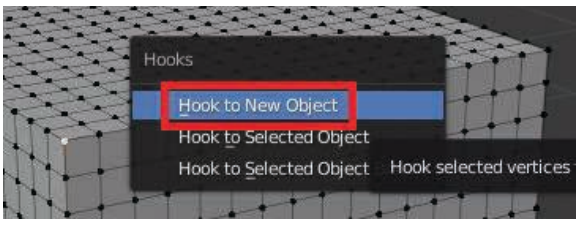


In Edit Mode select a corner vertex and **press Ctrl+H**.

This brings up the Hook menu. Choose **Hook to New Object**.

This connects a Hook to the selected vertex.

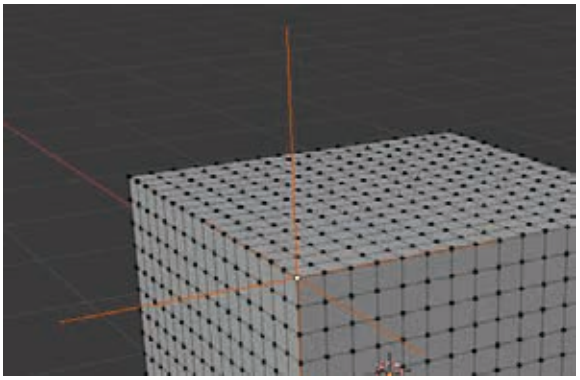




Click here to return to the index.

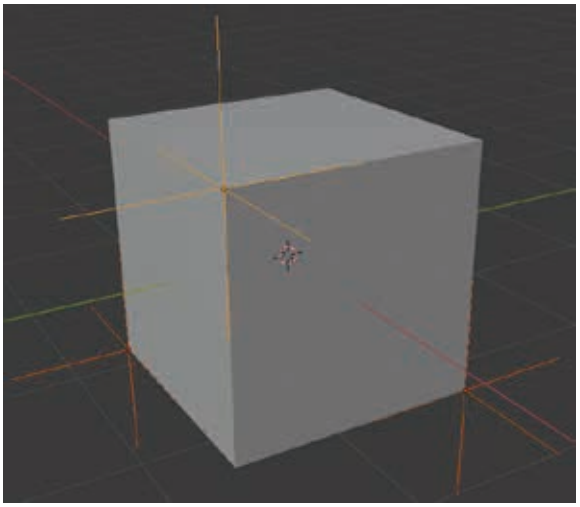
Now that vertex has a Hook attached to it. It looks similar to an empty.

It won't show up in your render.



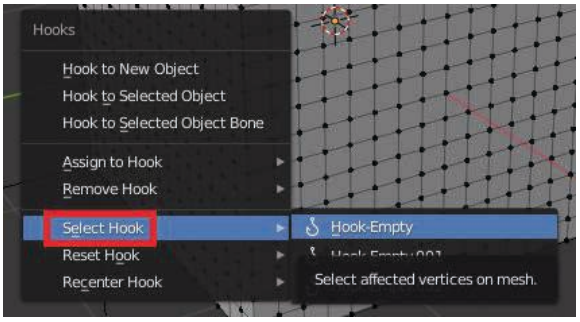
Repeat with some more vertices in different spots.

In this example I added hooks to 3 individual corner vertices.



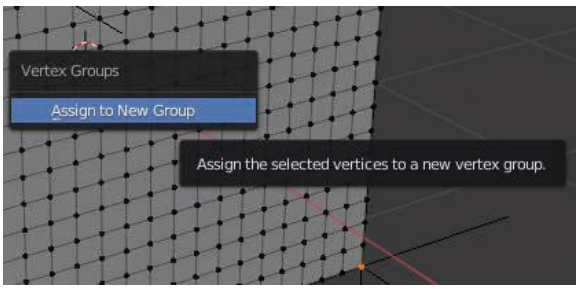
Select all the hooks from the same **Ctrl+H** menu.

You have to select them one by one.



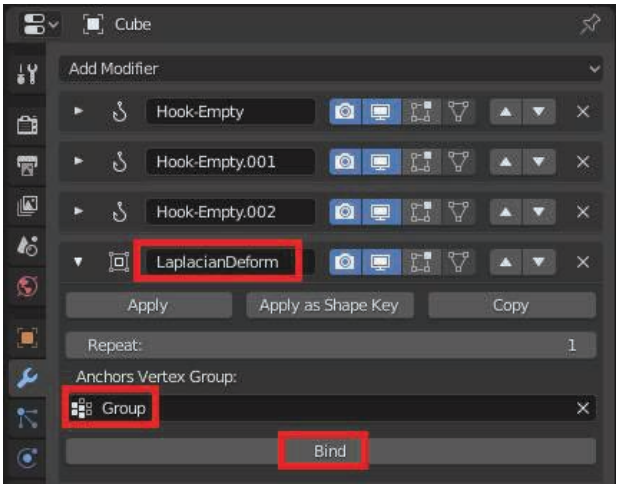
Press Ctrl+G to Group the selected hooks.

Choose "Assign to New Group".

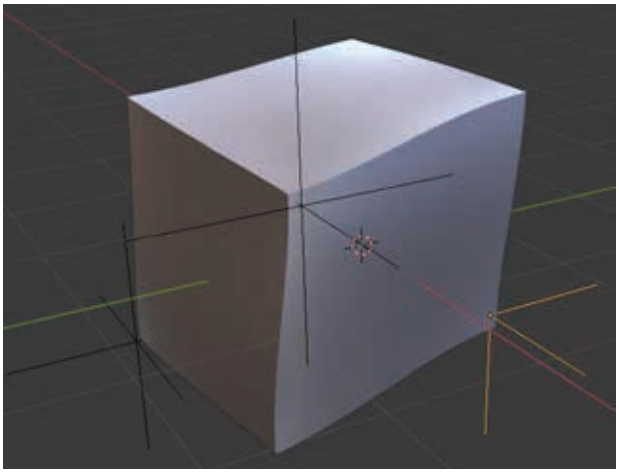


Add a **Laplacian Deform** modifier to the object.

Choose the **Group** you made earlier. Click on **Bind**.

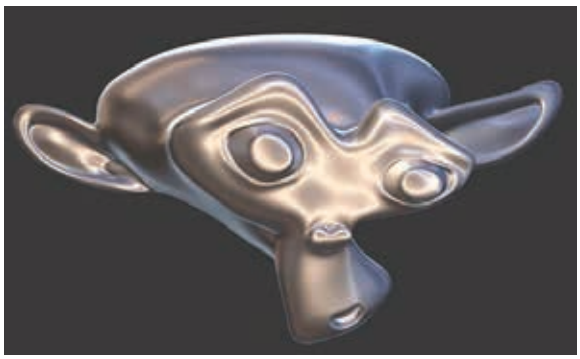


Now you can use the hooks to deform the cube or any other object you want to animate this way.

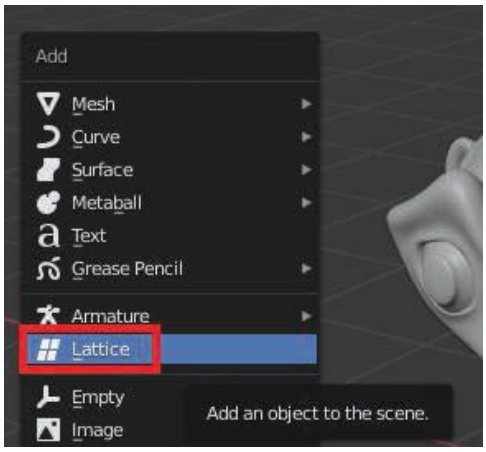


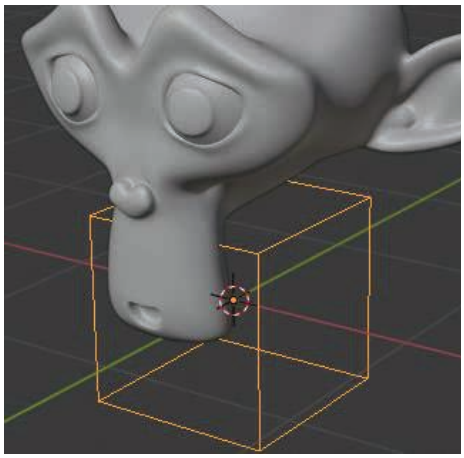
Lattice Deform Modifier

The Lattice Deform Modifier allows you to deform complex objects organically and easily.



To add a Lattice object to your scene **press Shift+A** and choose **Lattice**.

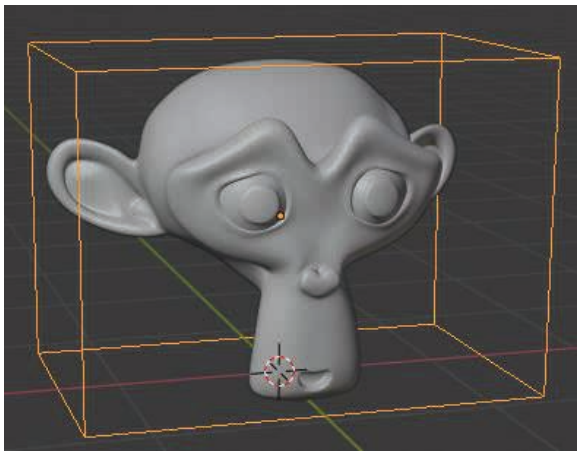




In **Object Mode** scale and move it to fit it around your mesh.

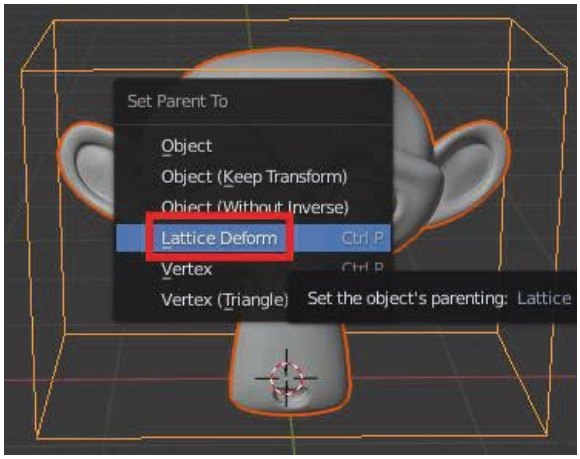
You can scale it along the individual axis by **pressing S+Y, X or Z** depending on which axis you want. That way you can make it fit around the object better.

Be careful **not** to scale it in Edit Mode, because the Lattice won't function correctly.



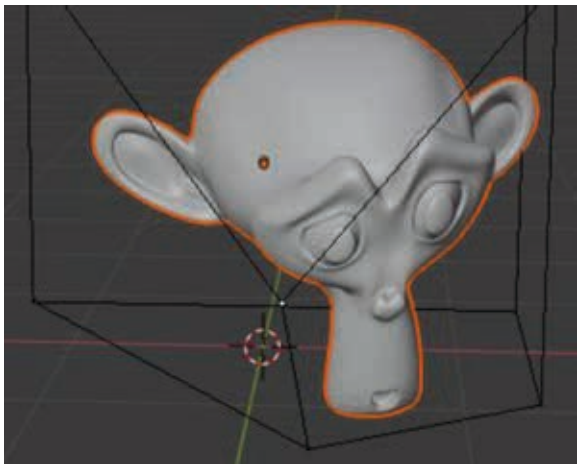
Holding Shift, select your mesh and then the Lattice.

Press Ctrl+P and choose **Lattice deform**.



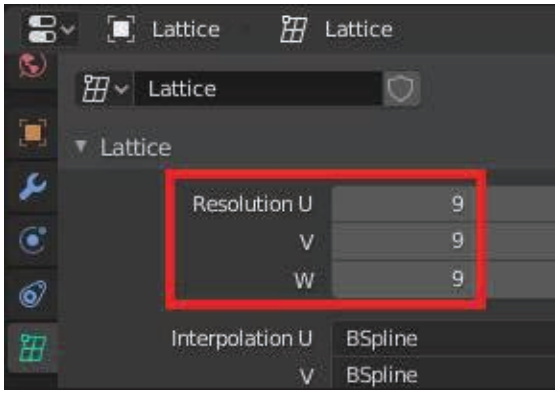
That's all!

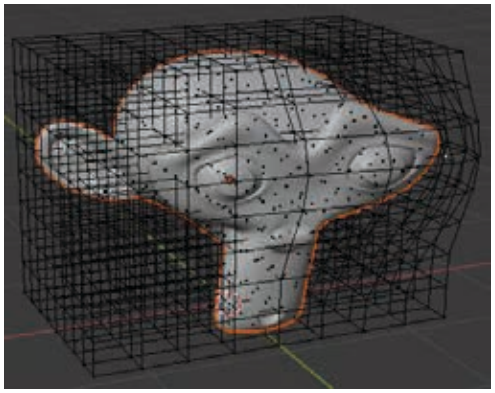
Press TAB to go into Edit Mode, you can now manipulate the Lattice, which in turn will deform the object.



In case you want more detail in your deformation cage you can add U, V and W edge loops to the Lattice. Then you can edit the individual points.

Add to that proportional editing (toggle on/off with keyboard shortcut **O**) and you can organically deform your mesh.

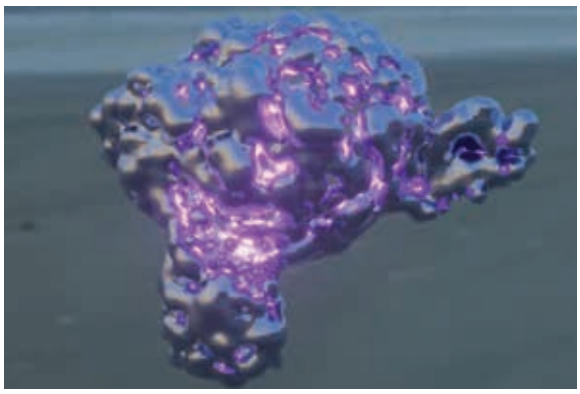




You can **press Apply** on the modifier afterwards if you want to keep the changes (though you don't have to).

Weird stuff with the Build Modifier

Let's make this weird blob!

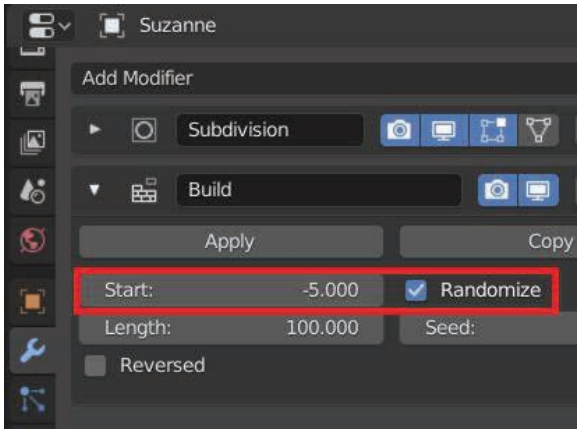


Press Shift+A and choose Mesh > Monkey to create a monkey.

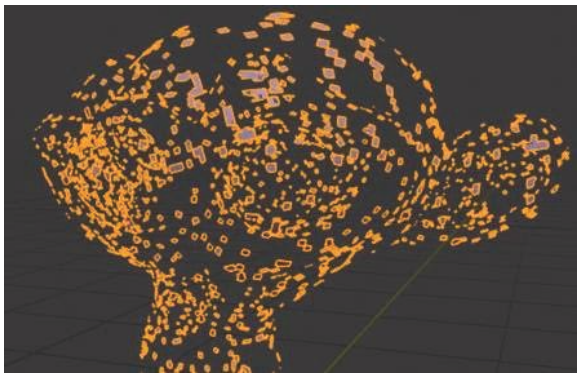
Add a **Subdivision modifier**and set it to View level 2 or 3 to add some more detail.

Add a **Build modifier** to it. Turn on **Randomize** and play with the **Start** value.

Make sure the Build modifier is below the Subdivision modifier.

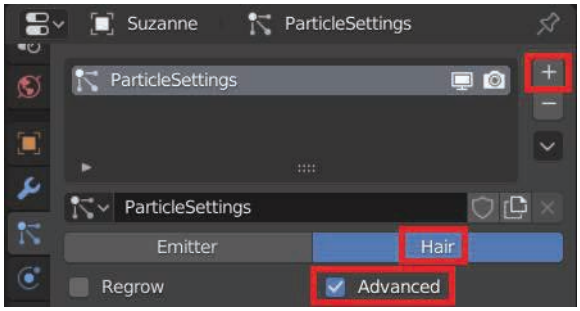


Apply the modifiers once you have something random that you are satisfied with (from top to bottom: first the Subdiv modifier, then the Build modifier).

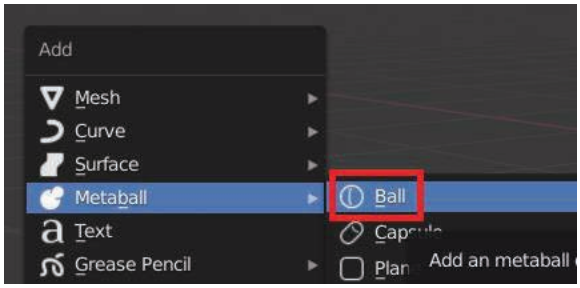


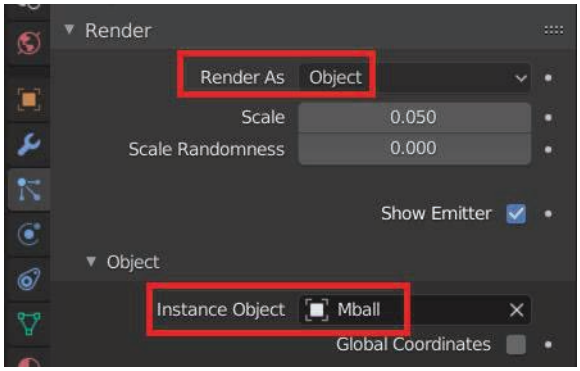
Click here to return to the index.

Add a **Particle system** to the monkey by clicking on the **+** icon, set it to **Hair** and **Advanced**.

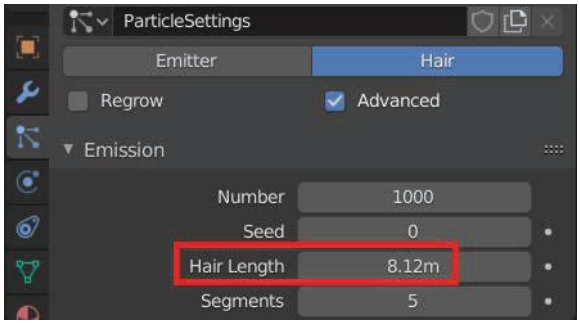


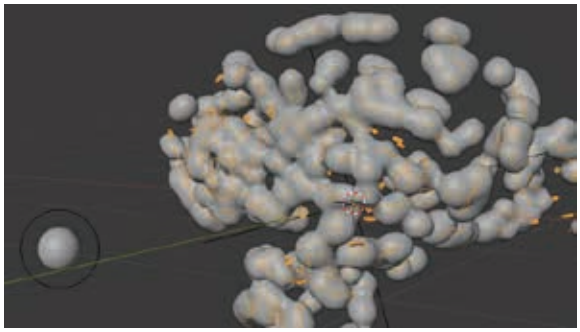
Press Shift+A and add a **Metaball.** In the Particle tab, set Render As to **Object** and use the Metaball as the **Instance Object**.





Play with the **Hair Length** until the Metaball size is the way you want.

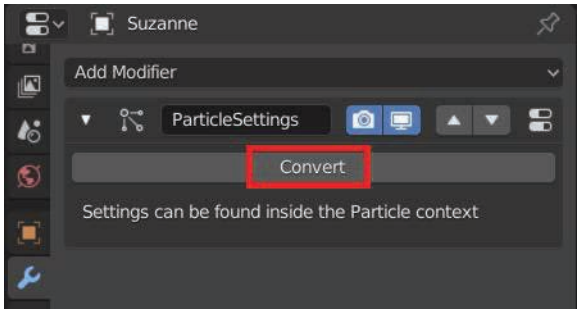


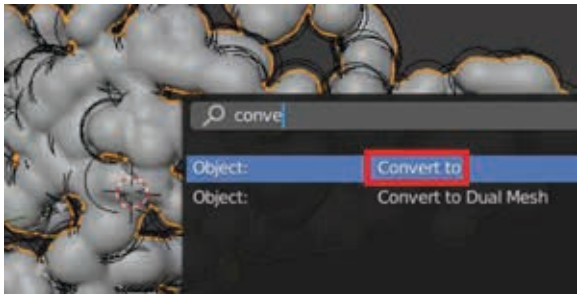


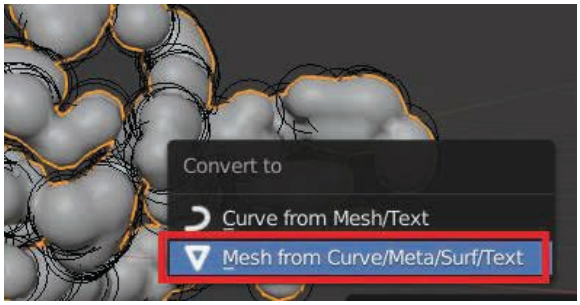
Apply the particle system by clicking on **Convert**.

Press F3 and search for **convert**, choose "Convert to" and then "Mesh from Curve…".

This converts the Metaballs to a regular mesh object.





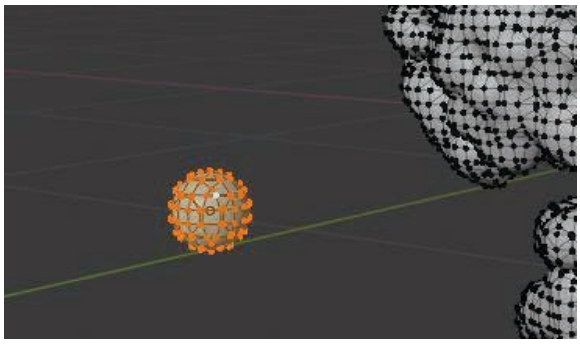


Press Ctrl+i to select everything that is not the metaball mesh, and **press X** to delete.

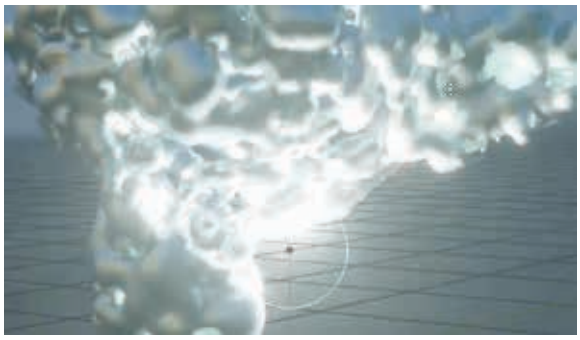
This gets rid of the leftover Monkey bits. You can select the original Metaball now,

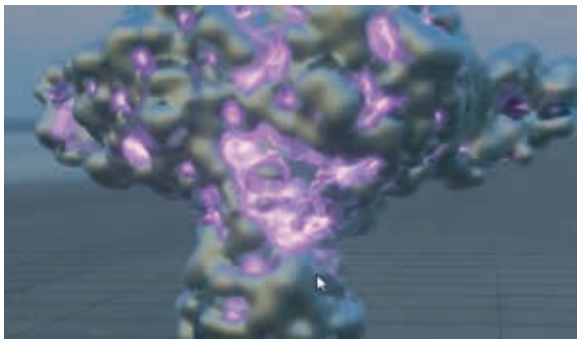
select all its faces or vertices and **press X**. One way to easily select all of it is mouse

hovering over it and **pressing L**.



In Eevee Rendered mode add some shaders, lights, an HDRI background...

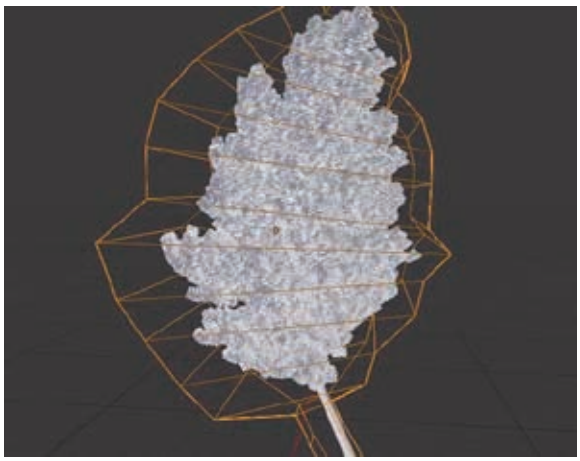




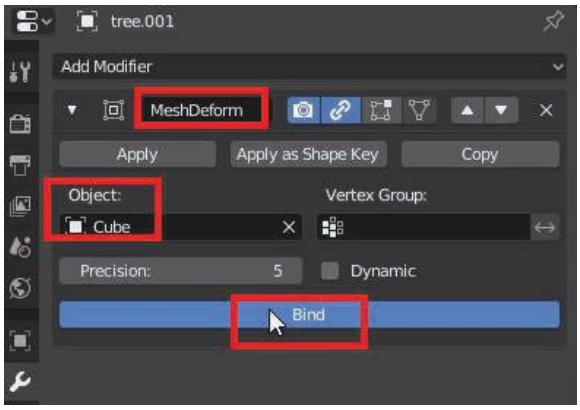
Mesh Deform modifier

Mesh Deform modifier is a lot like a Lattice modifier.

We model a simple shape around our model, which acts like a low polygon cage.

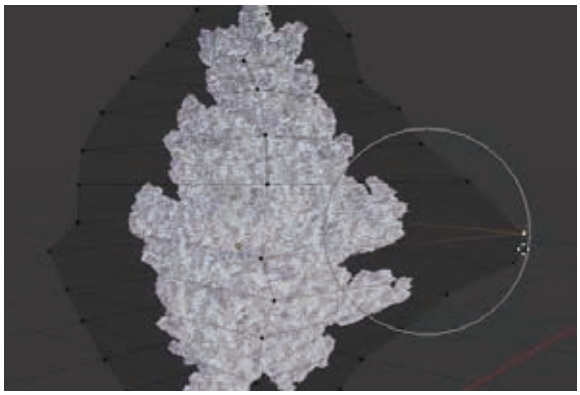


Add the **MeshDeform modifier** to your object, select the cage mesh, **click Bind**.



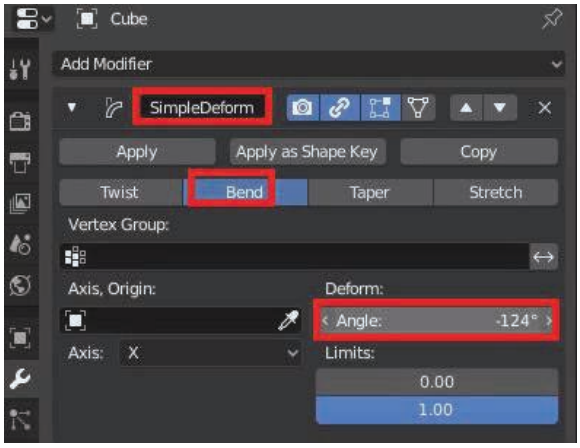
Now the cage mesh deforms the high-poly mesh.

Press o to toggle Proportional Editing for a more organic, proportional method of editing the cage mesh.

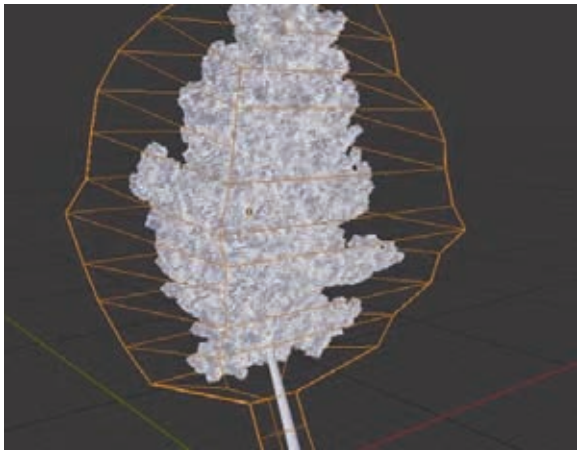


You can also add modifiers to the low poly cage object.

For example a **SimpleDeform modifier.** Set to Bend, and change the Angle value to animate the mesh.

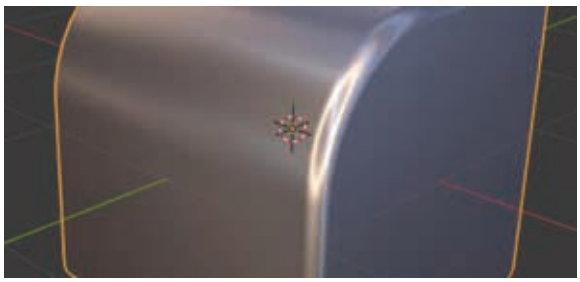


So this is an easy way to animate complex objects.

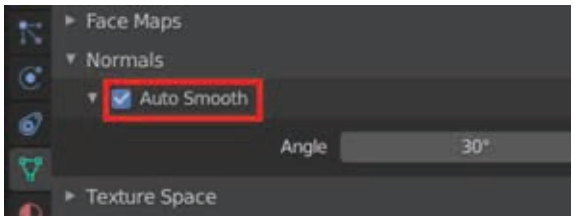


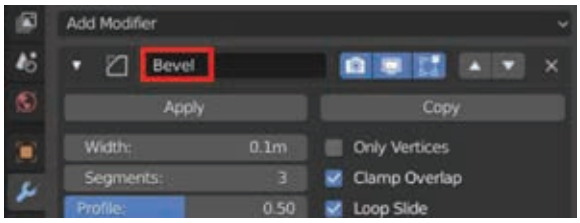
Weighted Normals modifier

Weighted Normals is a new modifier since Blender 2.8 that takes care of issues with Normals, resulting in better looking objects without time-consuming retopology.

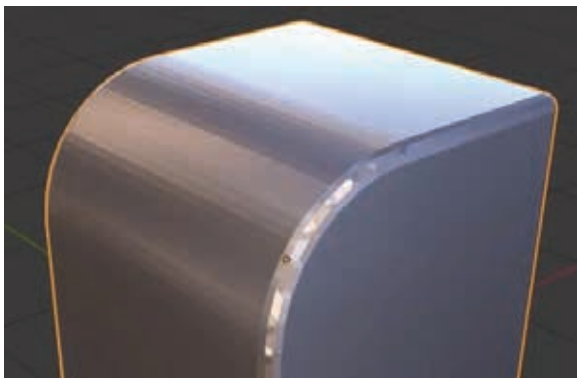


Turn on **Auto Smooth** (otherwise this modifier doesn't work) and add a **Bevel modifier**.

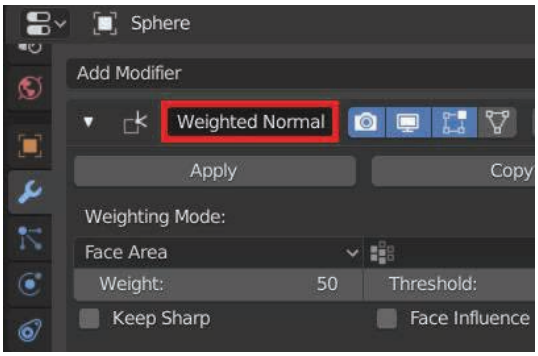


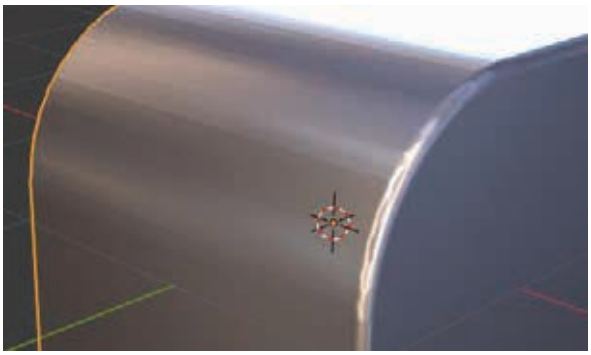


Bevels and Booleans don't always look smooth. As you can see the corner of this object has some weird artifacts caused by the Bevel modifier.



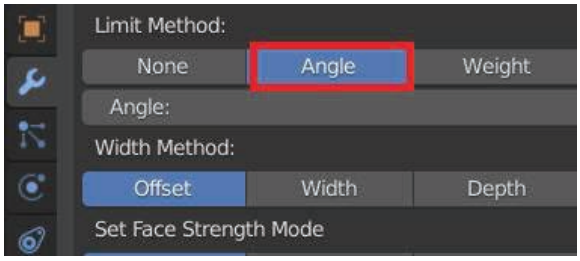
There is a new modifier that is an easy solution for this. Add the Weighted Normals modifier. Below on the right you see the difference it makes.

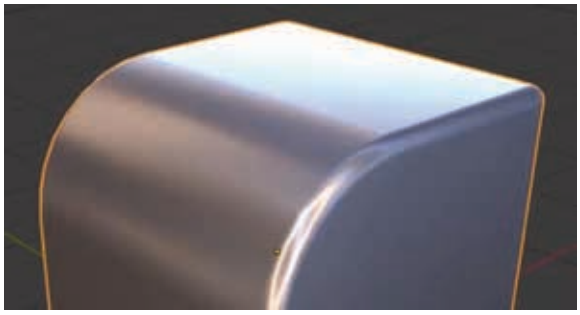




Click here to return to the index.

The Bevel is now smoother, but the rest of the object can still be better.
Change the Bevel method to **Angle**, this sometimes gives a better result.



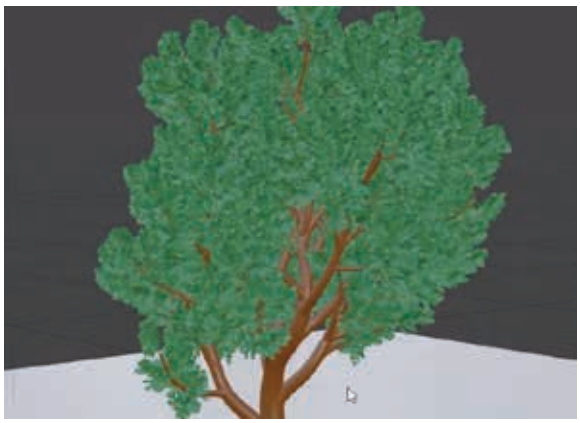


Play with the settings (weighting Mode and **Weight** specifically) until you get what works in your case.

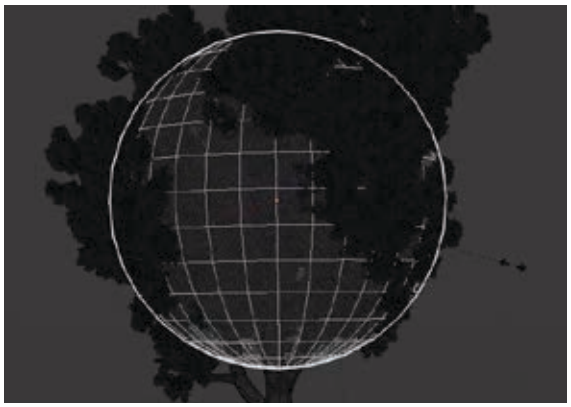
Editing Normals: Data Transfer

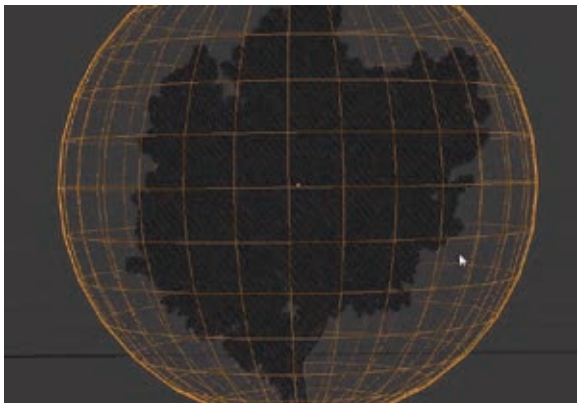
We can use the normals from a sphere for this tree.

This can be part of the method for a cartoon-style (NPR) look.



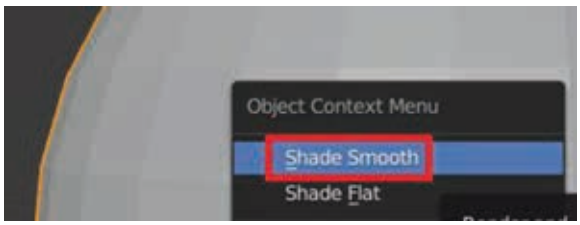
Create a sphere, in the same location as the tree. Scale it up to encapsulate the whole tree.



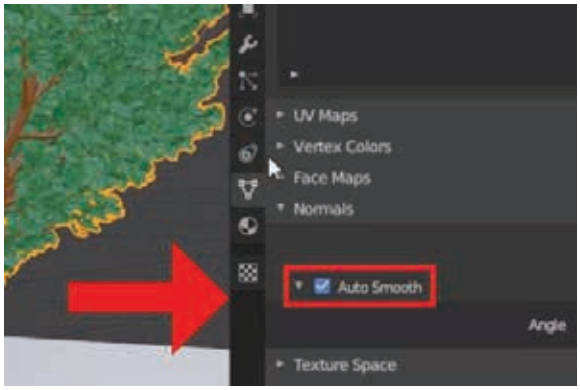


Right-click on the sphere and choose **Shade Smooth**.

Press H to hide the sphere.



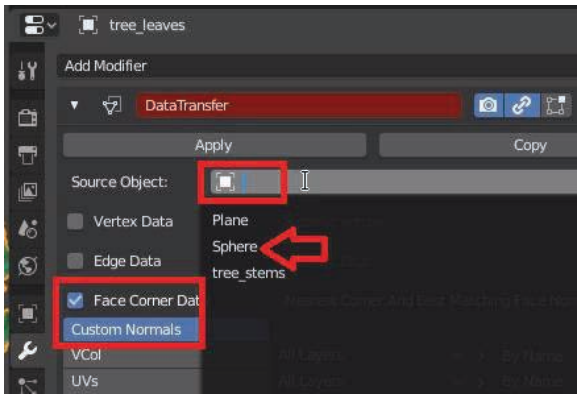
Turn on Auto Smooth for the leaves.



Add a **Data Transfer modifier**.

Turn on **Face Corner Data**.

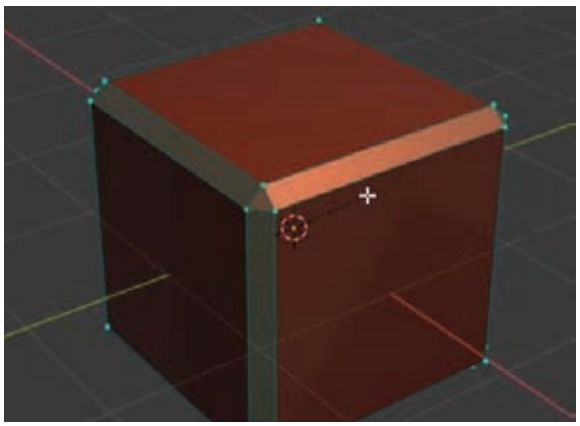
Click on **Custom Normals** and select the Sphere as the **Source Object**. Now the tree leaves use the normals from the sphere.



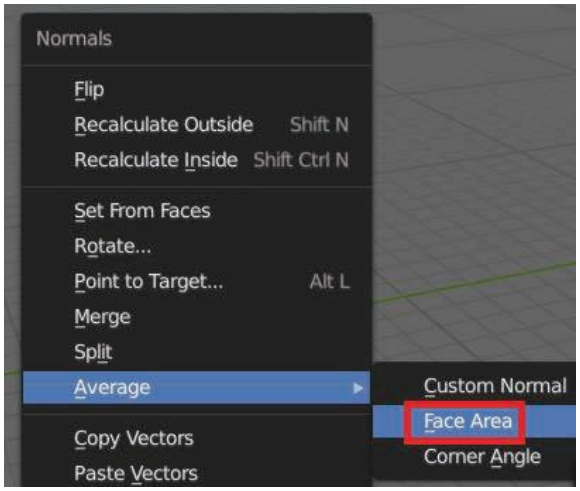
Now the leaf normals follow the angle of the sphere normals.

Smooth Edges with Average Normals

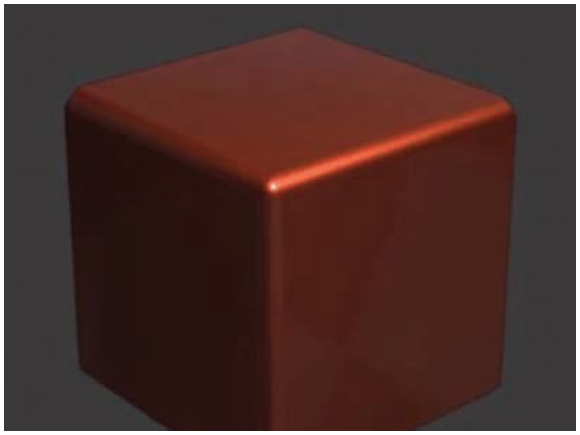
Smooth Edges with Average Normals
In Edit Mode add some low poly bevels with **Ctrl+B**.



Press Alt+N to bring up the Normals menu, choose Average > **Face Area**.



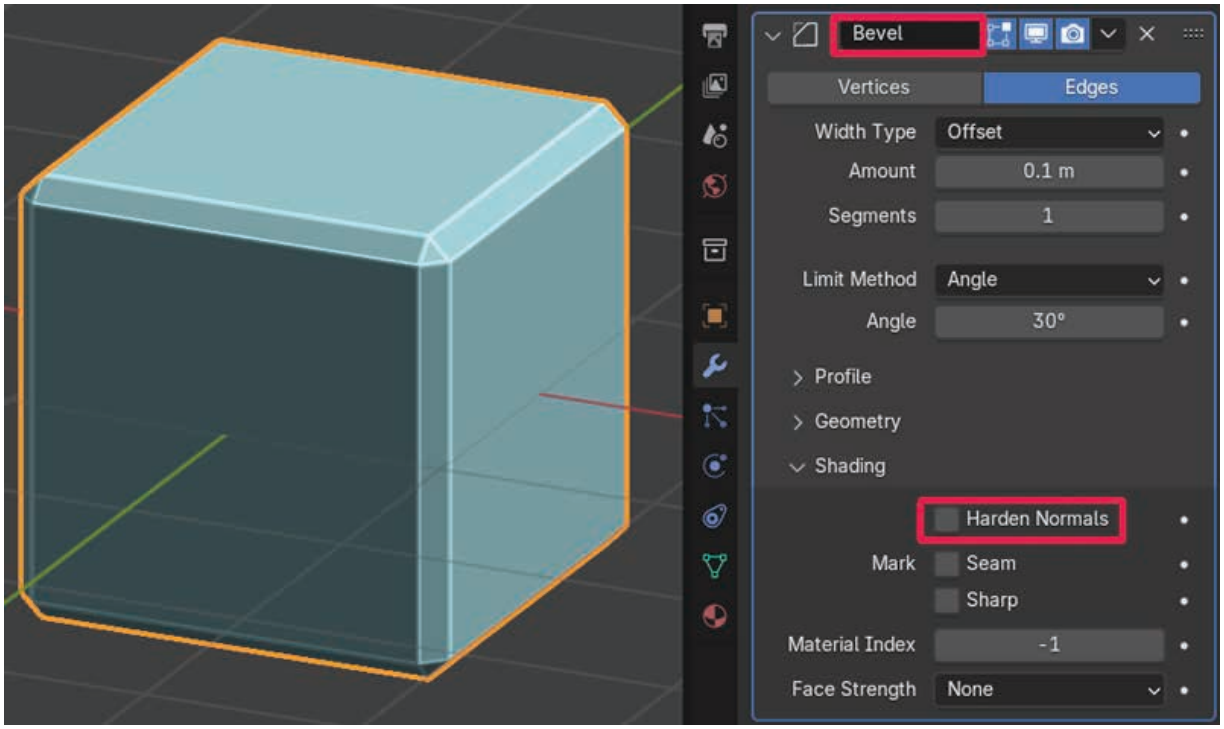
Now the cube looks very smooth, even though it is rather low poly.



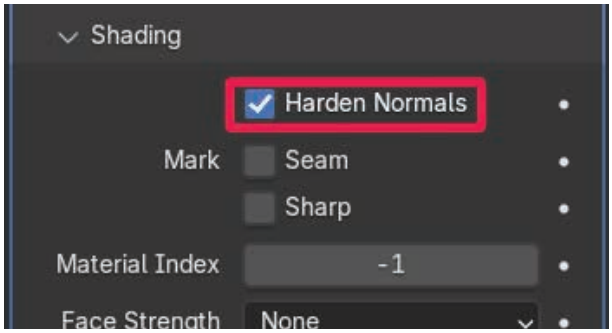
Click here to return to the index.

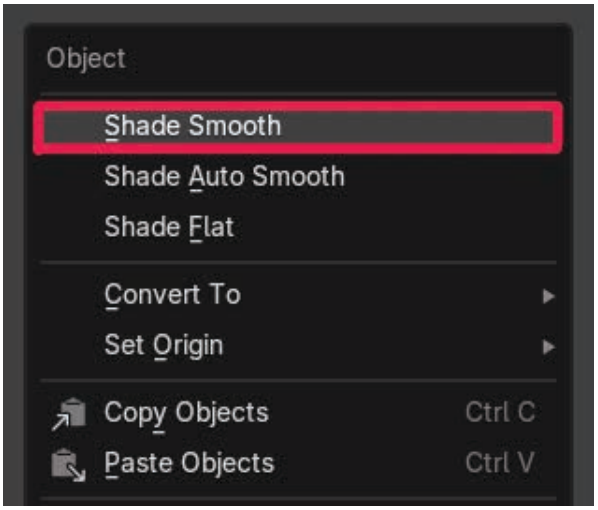
Smooth Edges with harden normals

You can achieve smooth beveled edges with minimal geometry using the Harden Normals option of the Bevel modifier. Here's a Cube with the Bevel modifier, and the Harden Normals option disabled (which is the default).



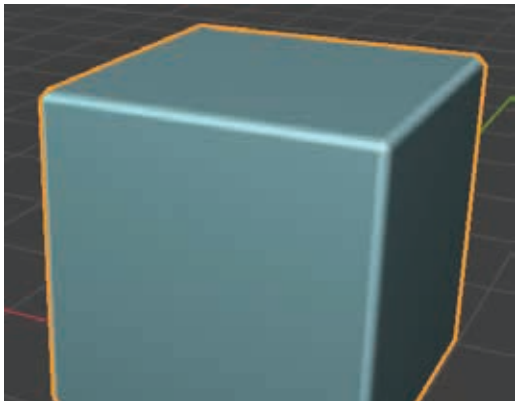
In Object Mode, right-click on the object and choose **Shade Smooth**. Enable the Harden Normals option in the Bevel modifier.





This is what the cube looks like now with a shiny Matcap.

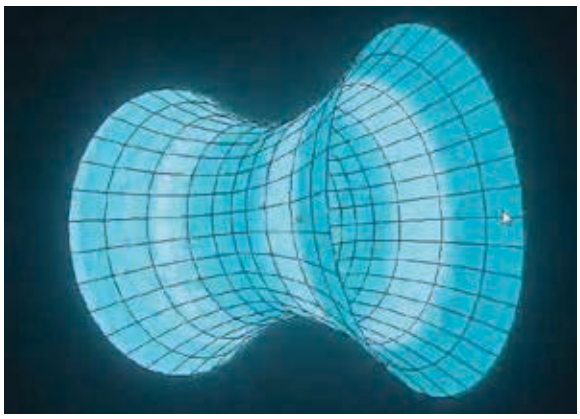
Pretty smooth edges with no additional geometry!



UV Project modifier

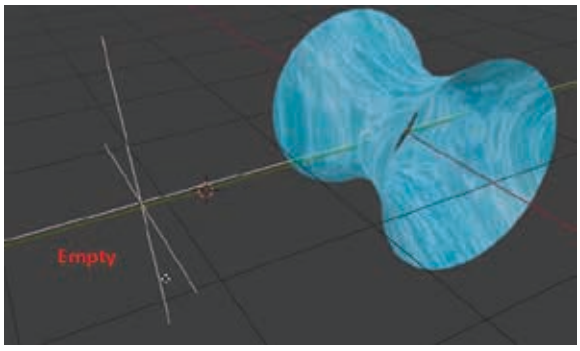
The UV Project modifier allows us to project UV's from different objects (like Empties).

Let's use it to create an animated texture for this wormhole model!



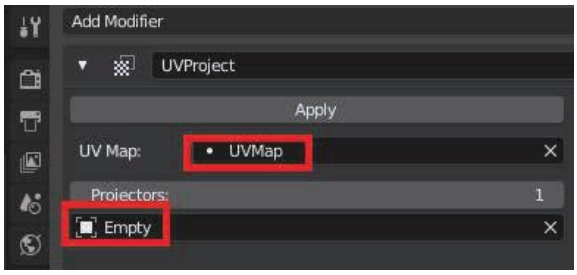
Here we've got a "wormhole" like cylinder which is UV-unwrapped and has a colorful texture.

Add an **Empty**, move it to the side.

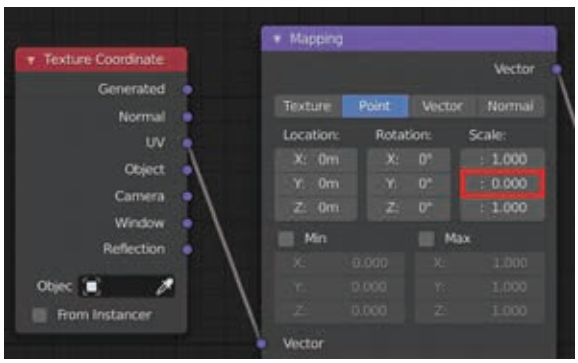


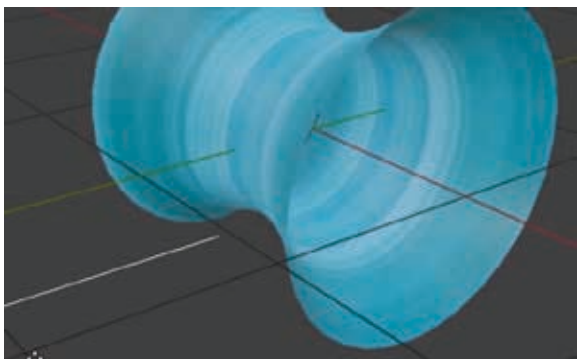
Add the **UV project modifier**, select the UV map and Empty.

Now the Empty controls the UV map.



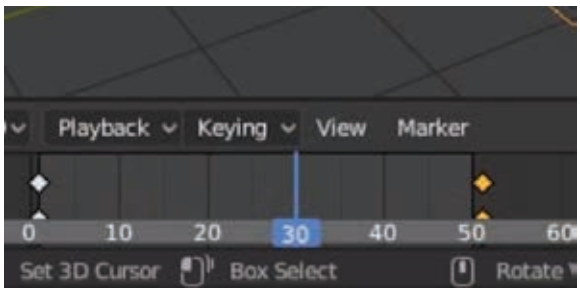
A quick adjustment to the Mapping node… we set the **Y-scale to 0**, so that we get the "rings" look on the model.





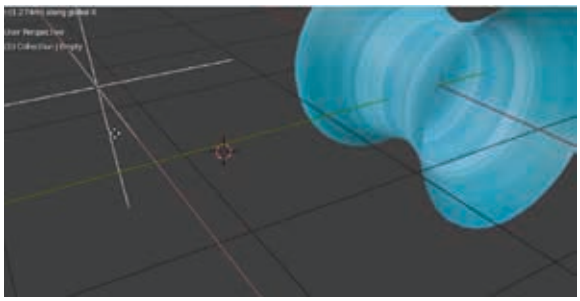
Click here to return to the index.

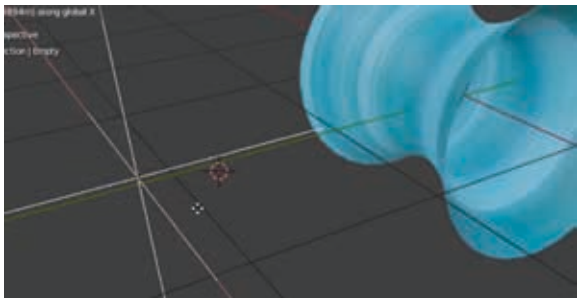
To loop it I just created two keyframes for the position of the Empty - one at the Start frame and one at oneframeafter the End frame.



Setting the second keyframe a frame later than the End frame will ensure a perfect looping animation.

So now when we **press the Spacebar** the Empty moves and the rings on the wormhole move with it.

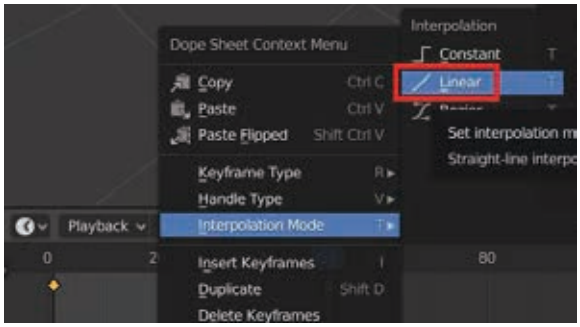




Set to **Linear Key interpolation**.

Select the keyframes in the Timeline and right-click, choose Interpolation Mode > **Linear**.

This makes sure the animation loops perfectly without slowing down.



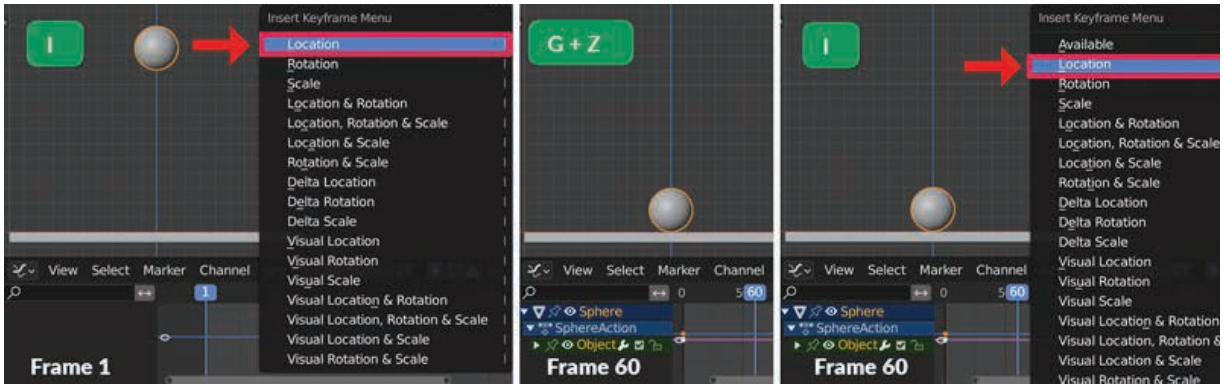
ANIMATION

3 ways to Bounce a Ball

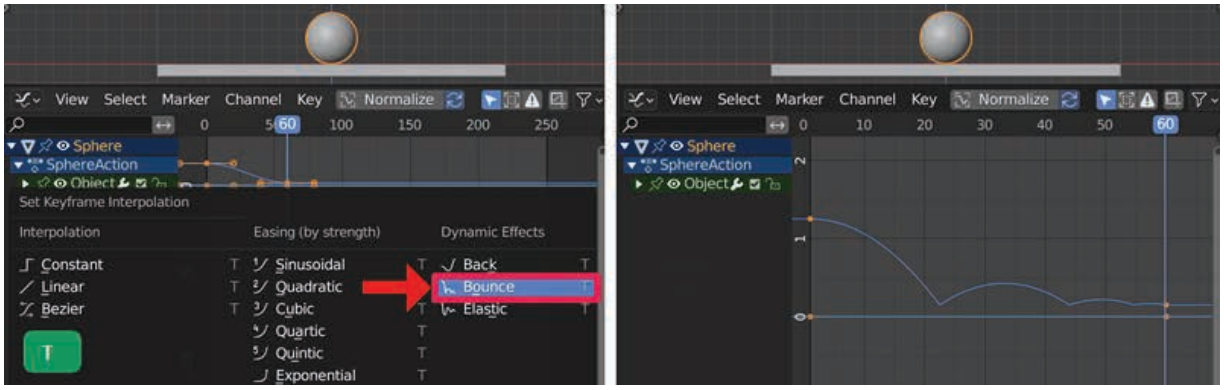
Here's 3 ways to do a Bouncy Ball.

Method 1

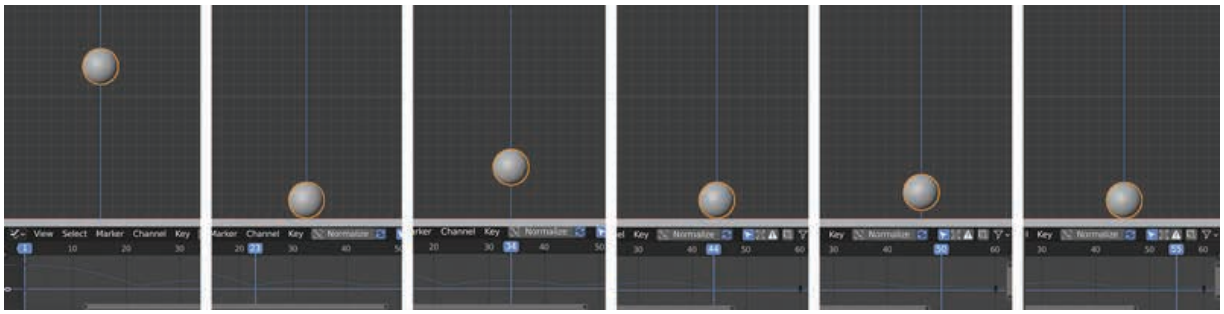
First, set some keyframes for the location by **pressing i** and moving the ball by **pressing G**.



Then in the Graph editor select the keyframes, **press T** and choose **Bounce.**

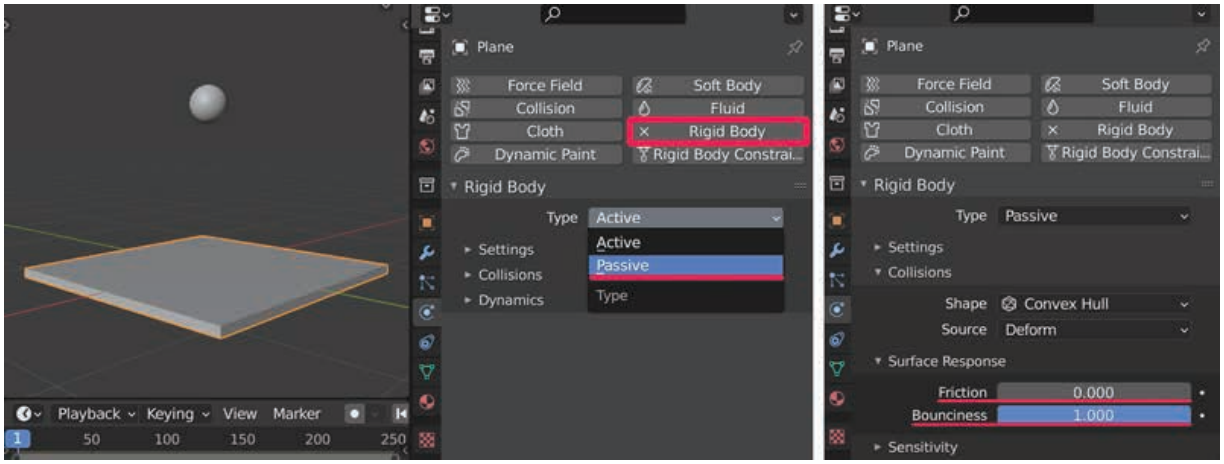


When you **press the Spacebar**, you'll see the ball bounce.

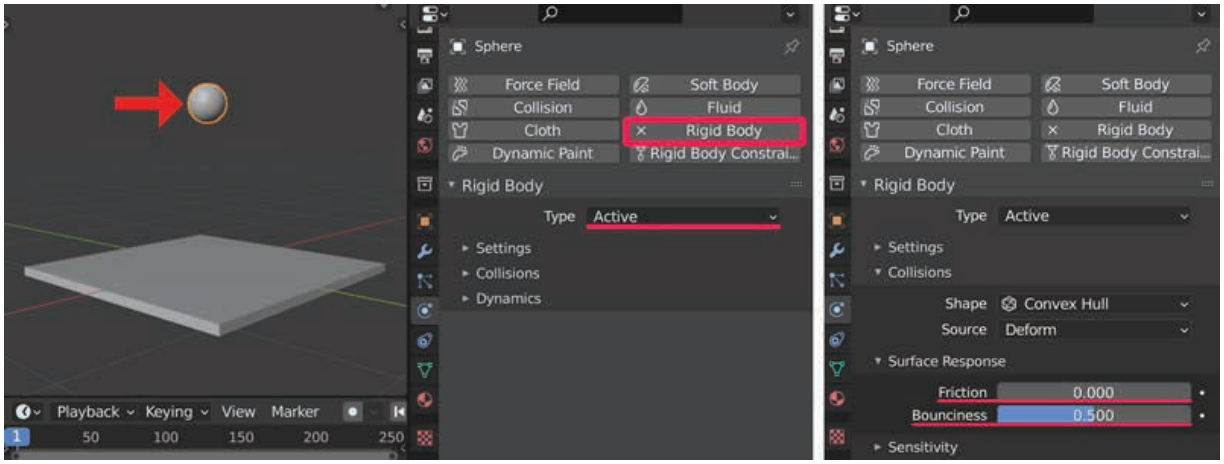


Method 2

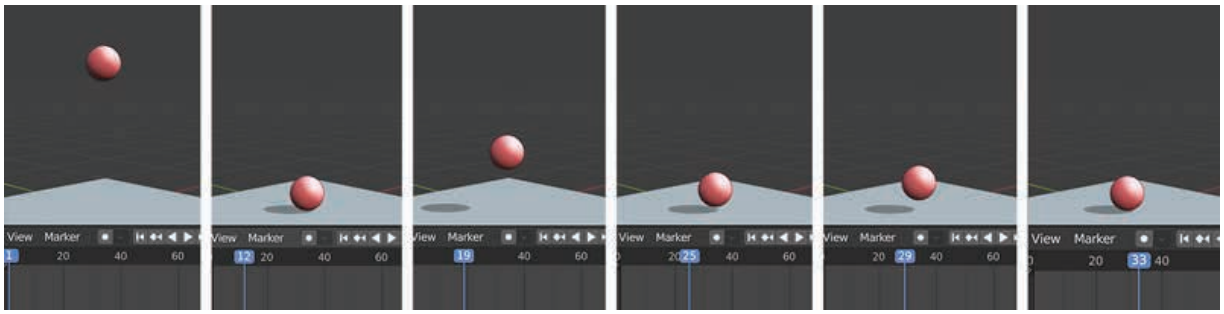
Set the ground plane to **Rigid Body** and make it **Passive**. Reduce the **Friction** and increase **Bounciness** to 1.



Set the Ball to **Rigid Body** and keep it as **Active**. Reduce the **Friction** and increase **Bounciness** to 0.5.



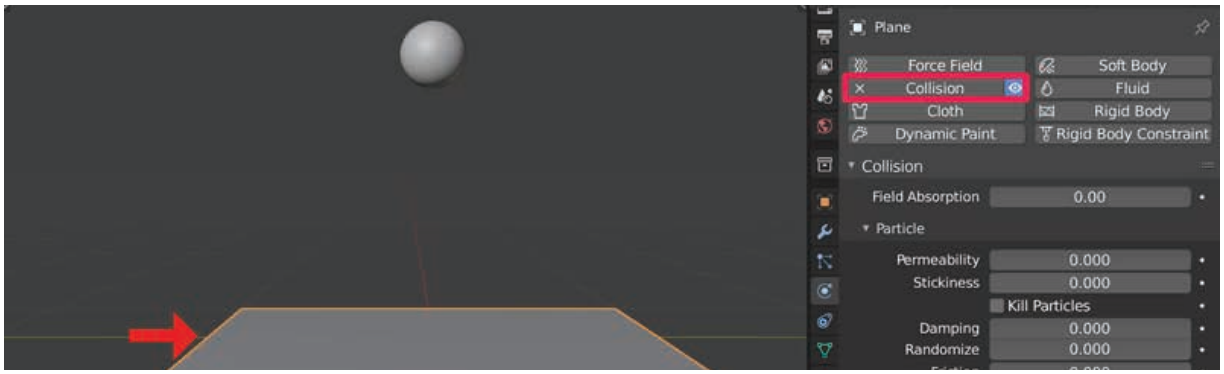
Now when you **press the Spacebar**, the bouncing animation starts.



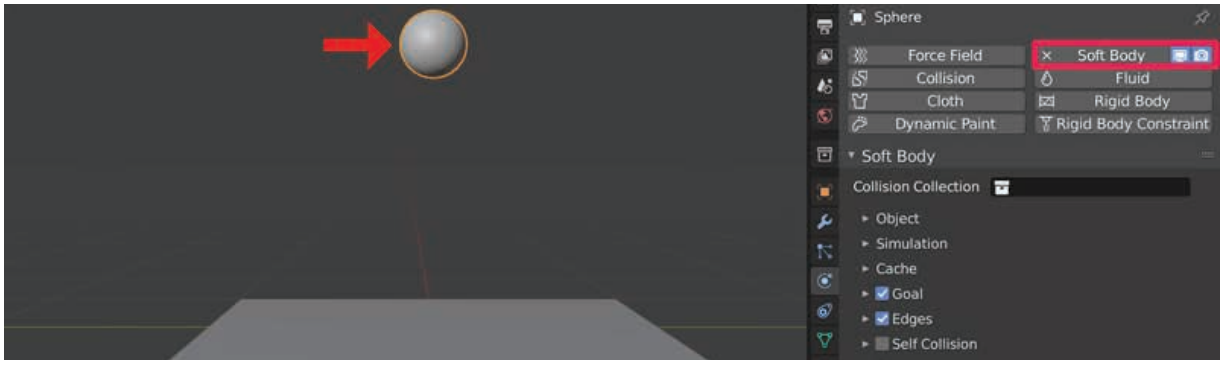
Click here to return to the index.

Method 3

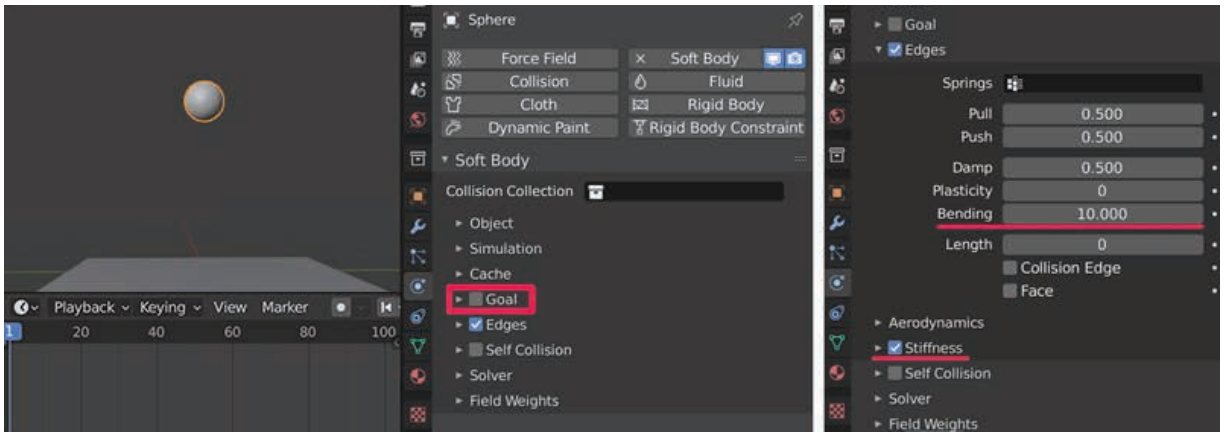
Enable **Collision** for the ground plane.



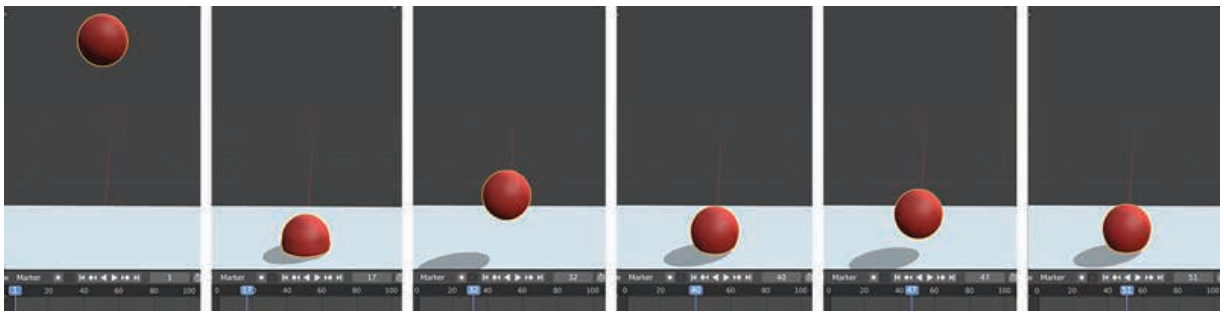
Enable **Soft Body** physics for the ball.



Turn off Soft Body **Goal** and set **Bending** to 10. Turn on **Stiffness**.

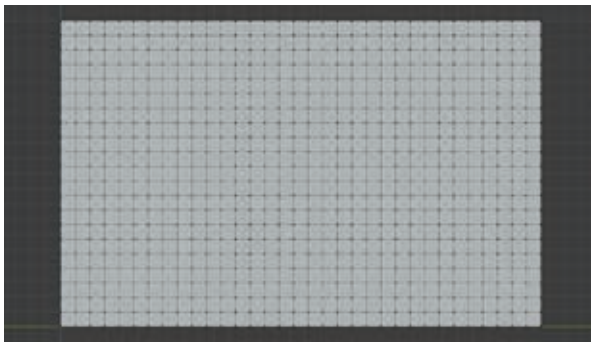


Press the Spacebar to see the bounciness of the ball.



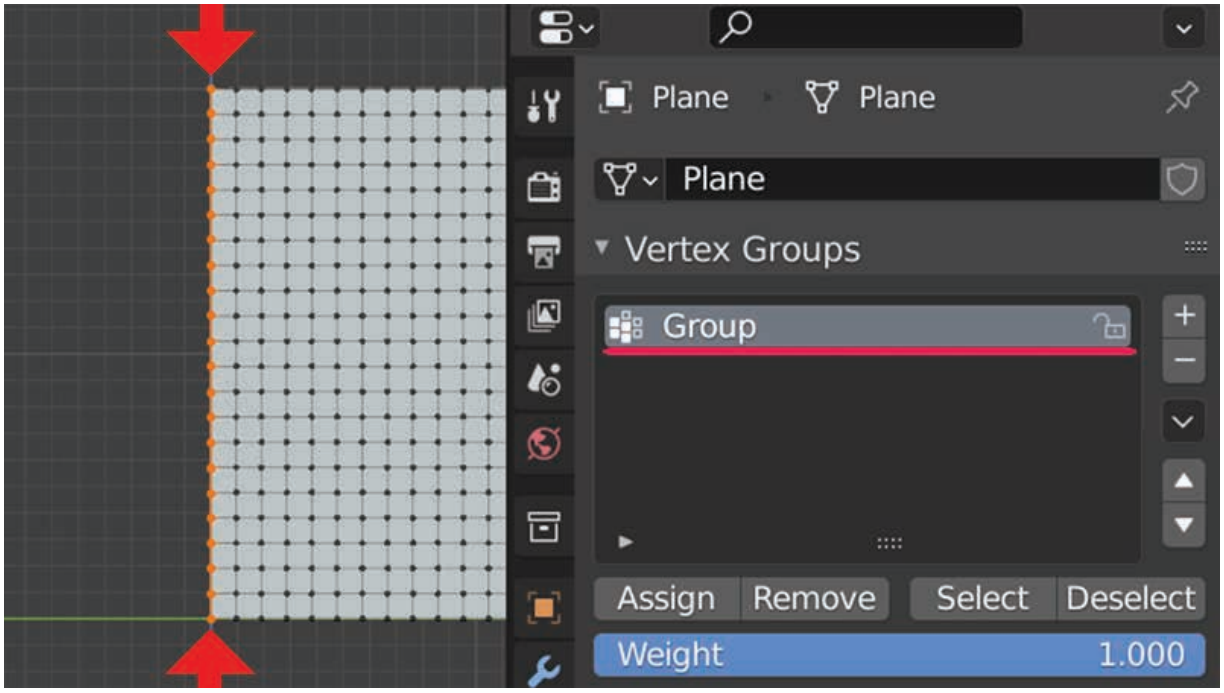
Animated Flag

Make a flag-shaped plane with enough subdivisions.

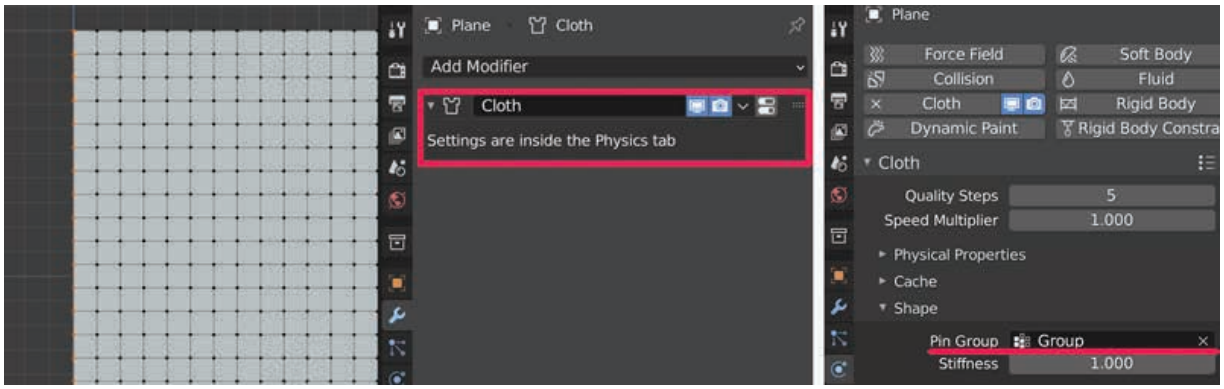


You can right-click and choose **Subdivide**, or add loops manually by **pressing Ctrl+R** and scrolling the mouse wheel up in Edit Mode.

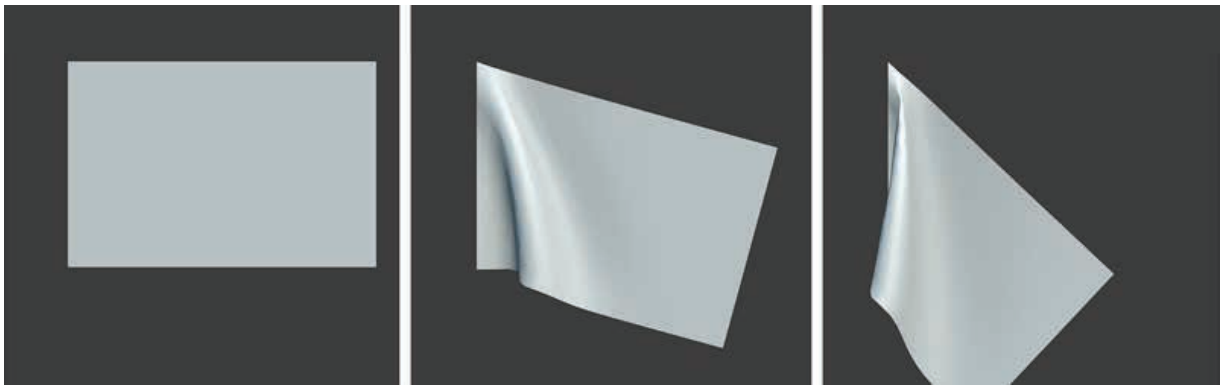
Add the vertices on one end to a vertex group.



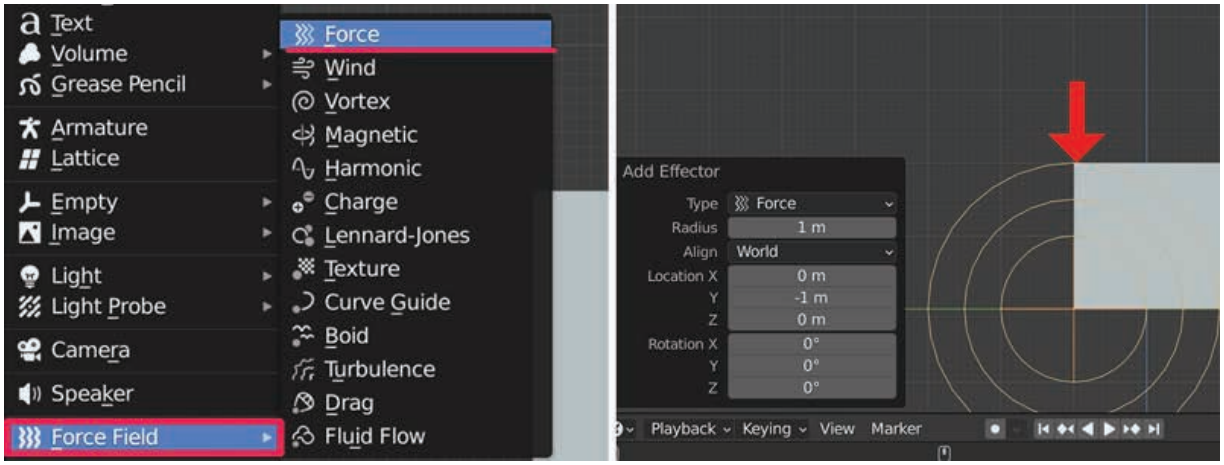
Add a **Cloth modifier** and add the vertex group to the **Pin Group**.



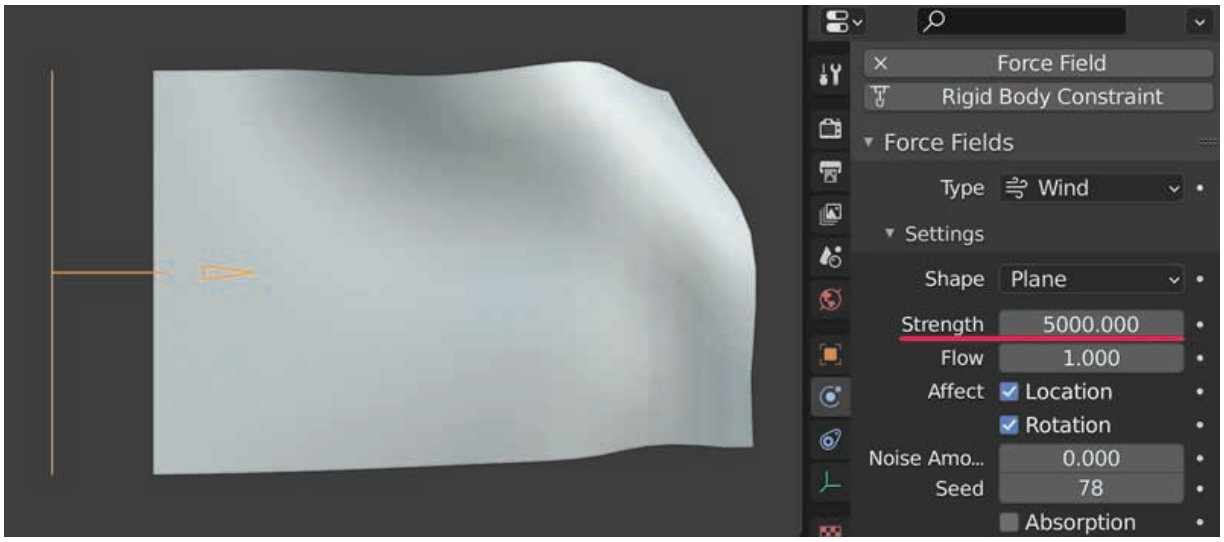
Start playing the simulation by **pressing Spacebar.**



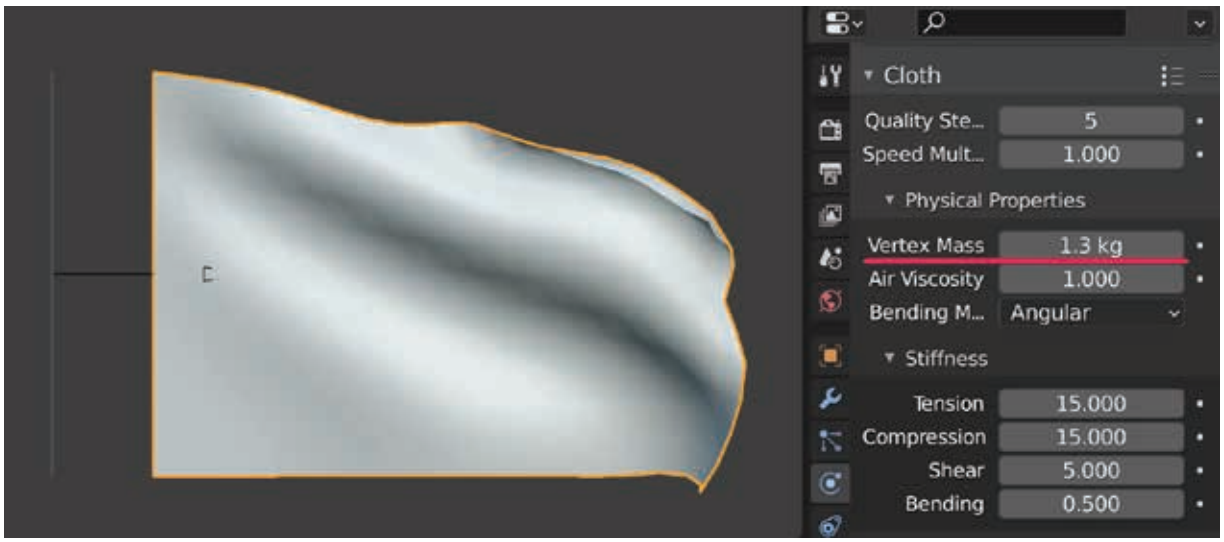
Add a **Force > Force** or **Wind** to the scene, either will work.



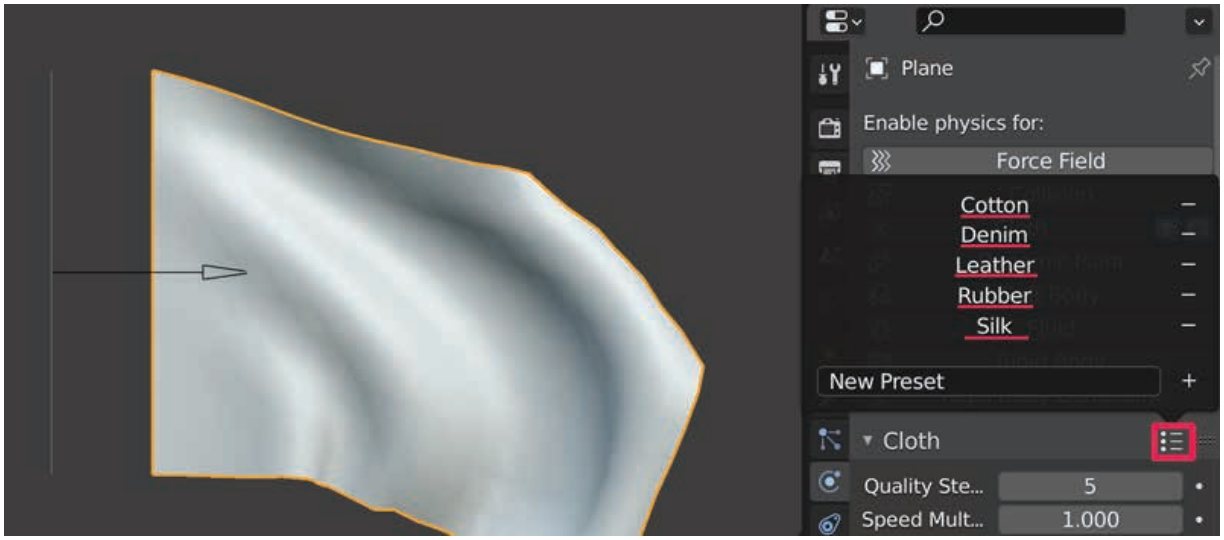
Add some more force to the wind by increasing the **Strength**. A value of 5000 for example.



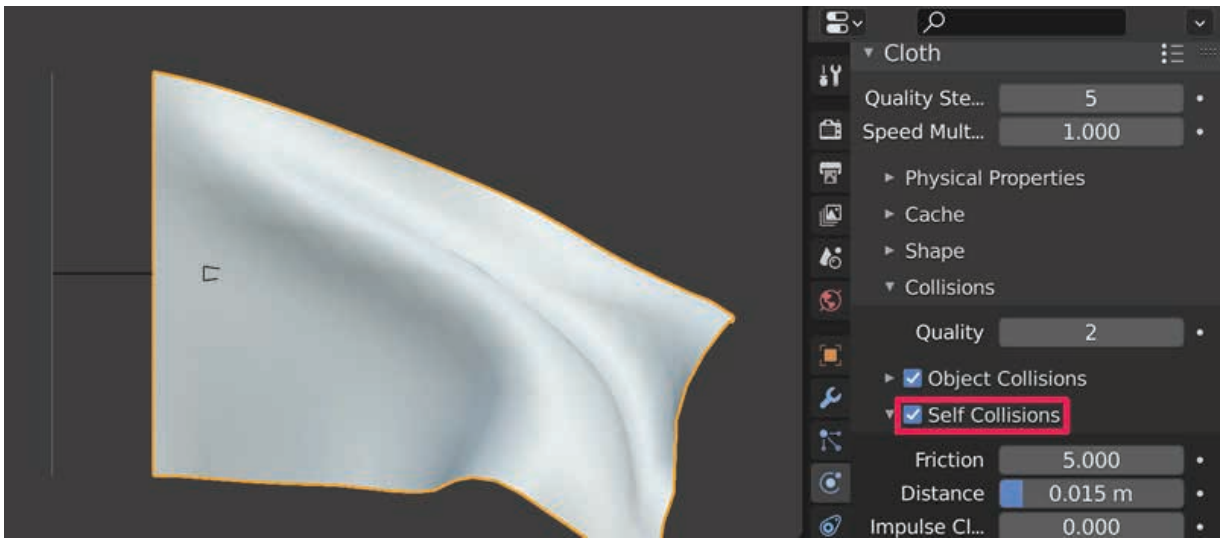
Experiment with the vertex mass. This value depends on your object.



You can experiment with the different **Cloth** presets.

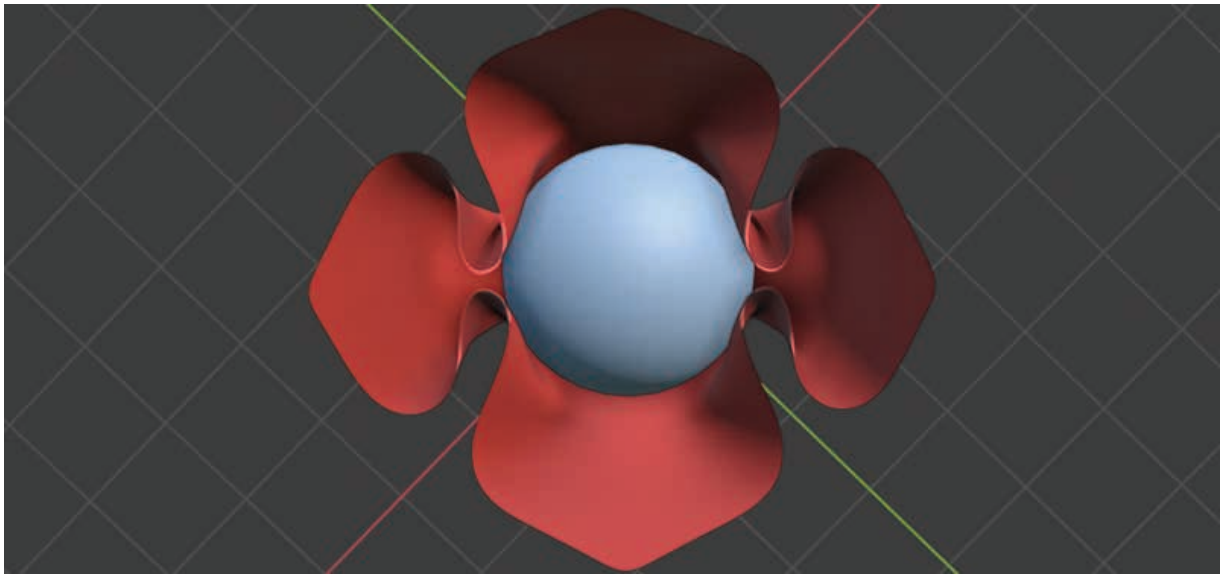


Turn on **Self Collisions** so the flag doesn't intersect with itself.

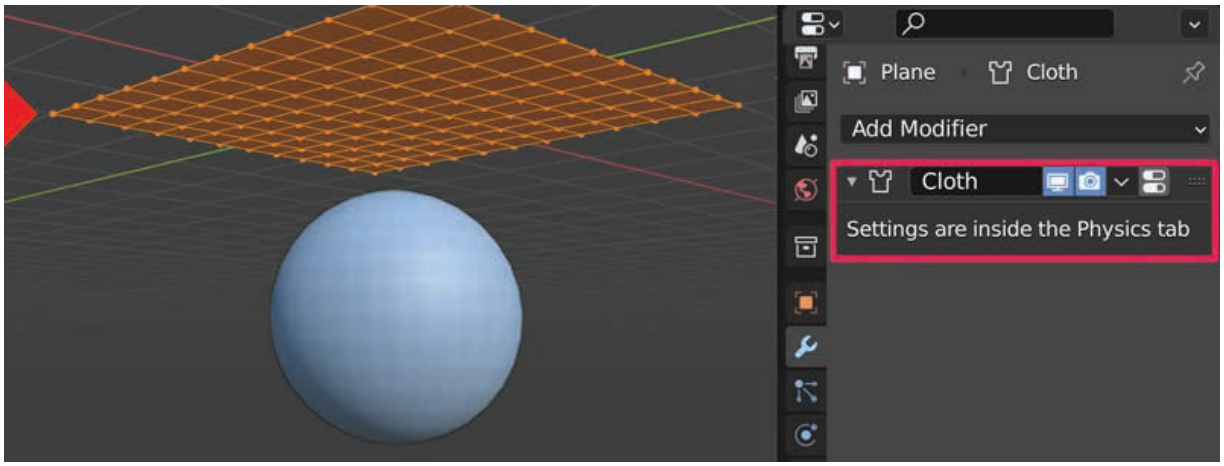


Slow-motion Cloth Simulation

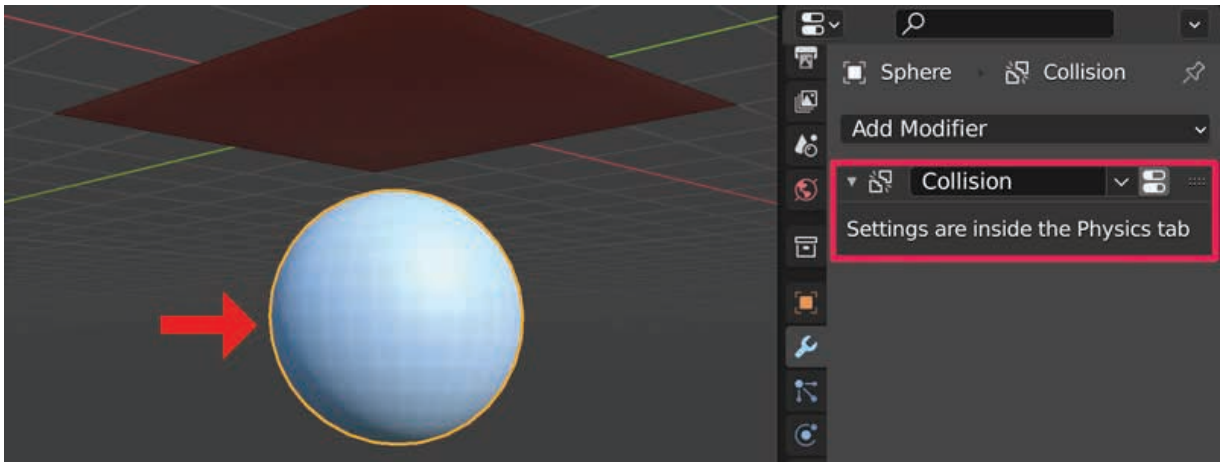
To simulate a piece of cloth moving in ultra slow motion, is actually very easy.



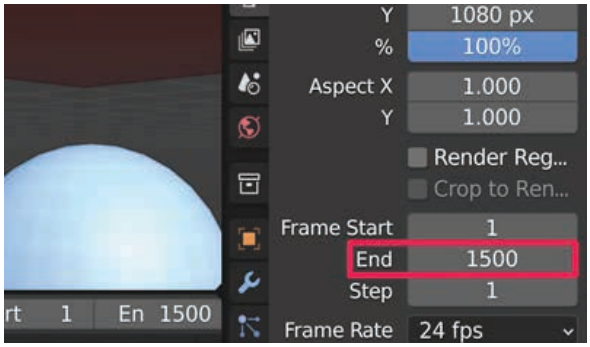
First we add a **Cloth modifier** to this subdivided plane.



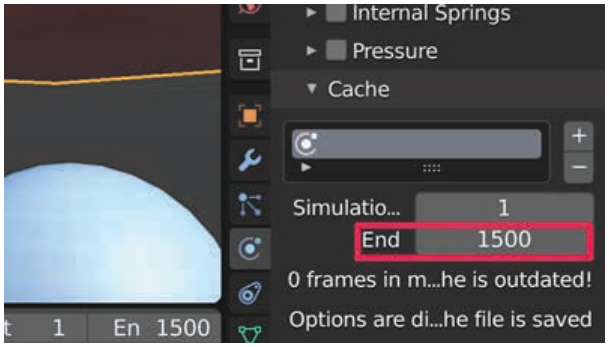
Next, add a **Collision modifier** to this sphere.



Then, set enough keyframes in the **Render** settings (in the Render tab).

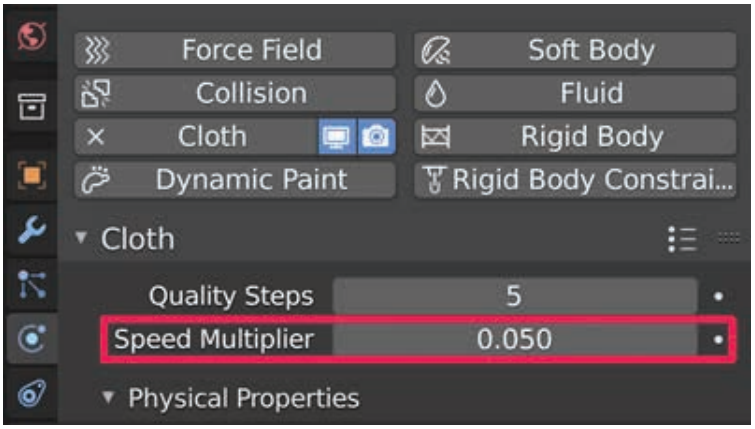


Similarly, set enough keyframes in the cloth **Cache** settings.

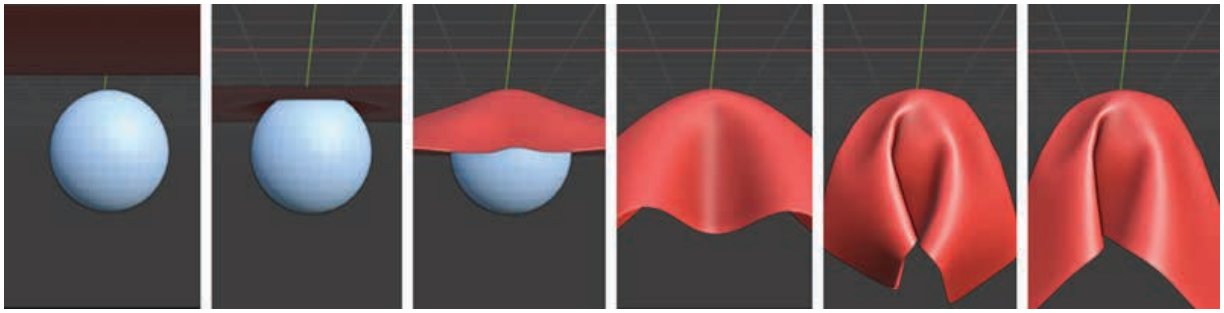


Now simply change the value in the **Speed Multiplier** to something smaller, like 0.005.

This is what actually makes it seem to move in slow motion.

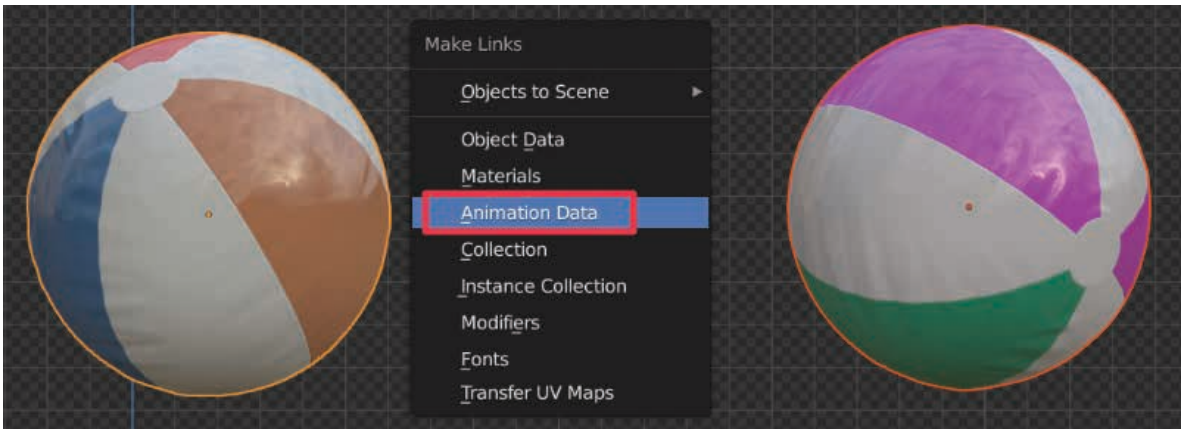


Then play the simulation by pressing the Spacebar.

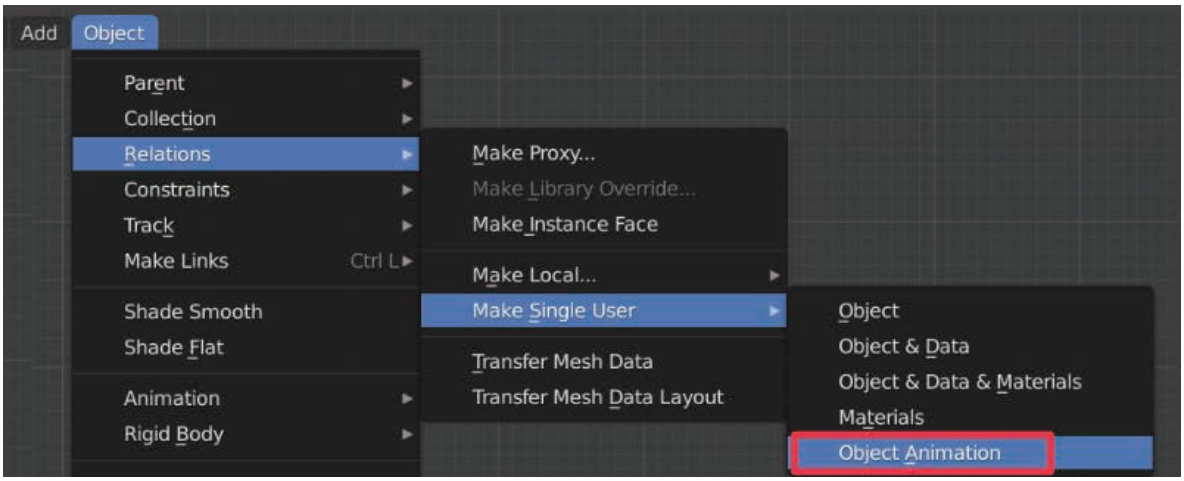


Copy Animation From One Object To Another

To copy animation from one object to another select the object you want to animate then select the object you want to copy animation from. Now **press Ctrl+L** and choose "**Animation Data**".



As the animation data is now linked changing the animation on one object also changes the animation on the other. To prevent this, make the object "**single user**". Go to Object > Relations > Make Single User > **Object Animation**.

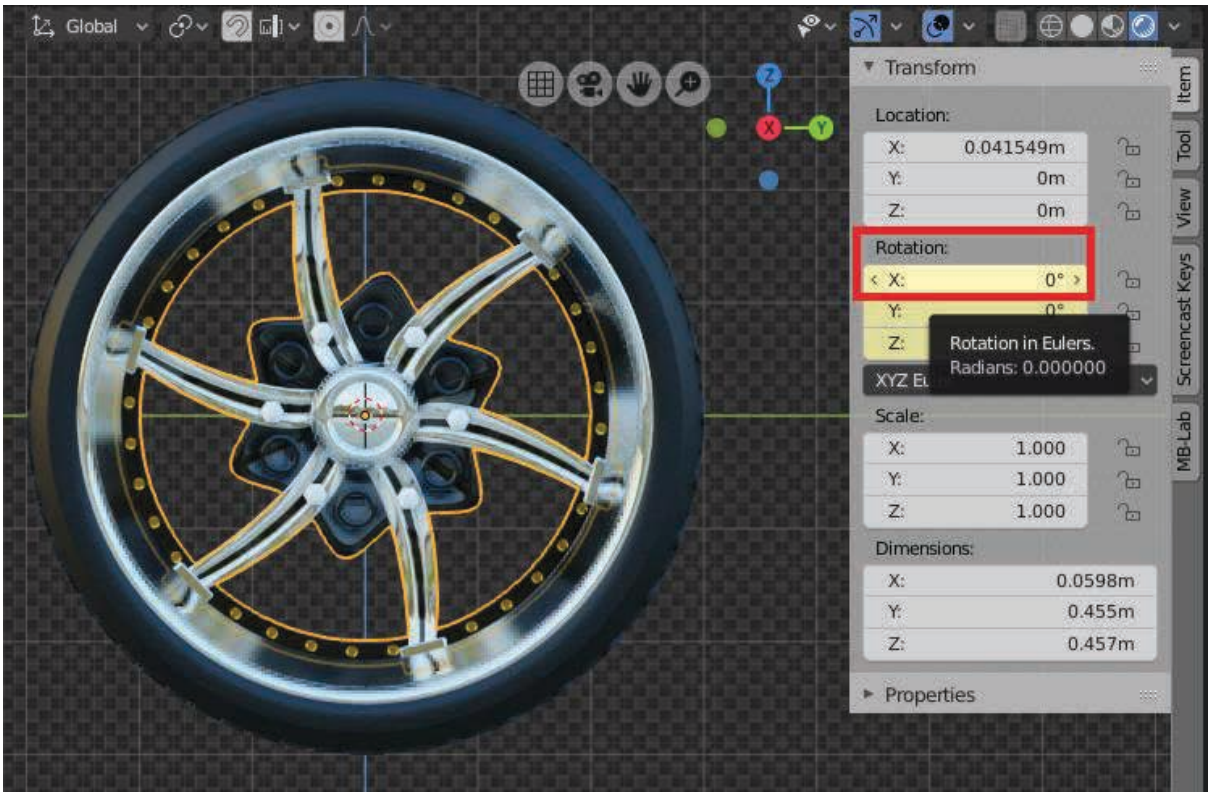


Now you can edit the animation keyframes of each object independently in the Graph editor. You can even do this with armatures, as long as they are the same.

Perpetual Controlled Speed Tire Rotation

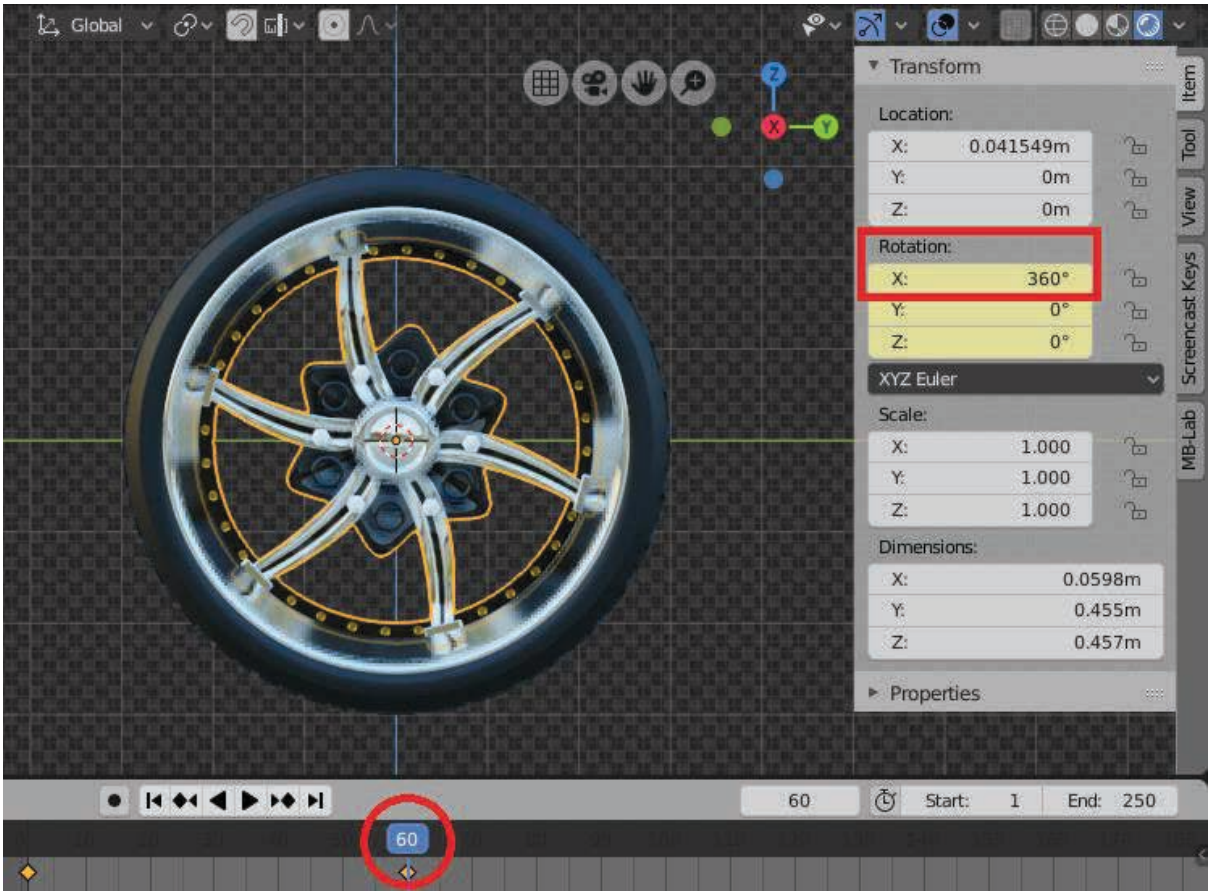
Sometimes you want things to rotate indefinitely or for a specific amount of time and with a specific speed, like when animating a wheel. There is a way to set this up in the NLA Editor, where you can also control the speed and amount of repetitions of the rotation.

First, set a keyframe at frame 0 by **pressing "i"** while the mouse cursor is above the "X Rotation" input in the Option panel.



Click here to return to the index.

Next, set a keyframe a bit later in the Timeline (at frame 60 for example) for 360° Rotation in the same field, X Rotation. **Press "i"** again to make sure the key is set (you will see the color of the field change to yellow).

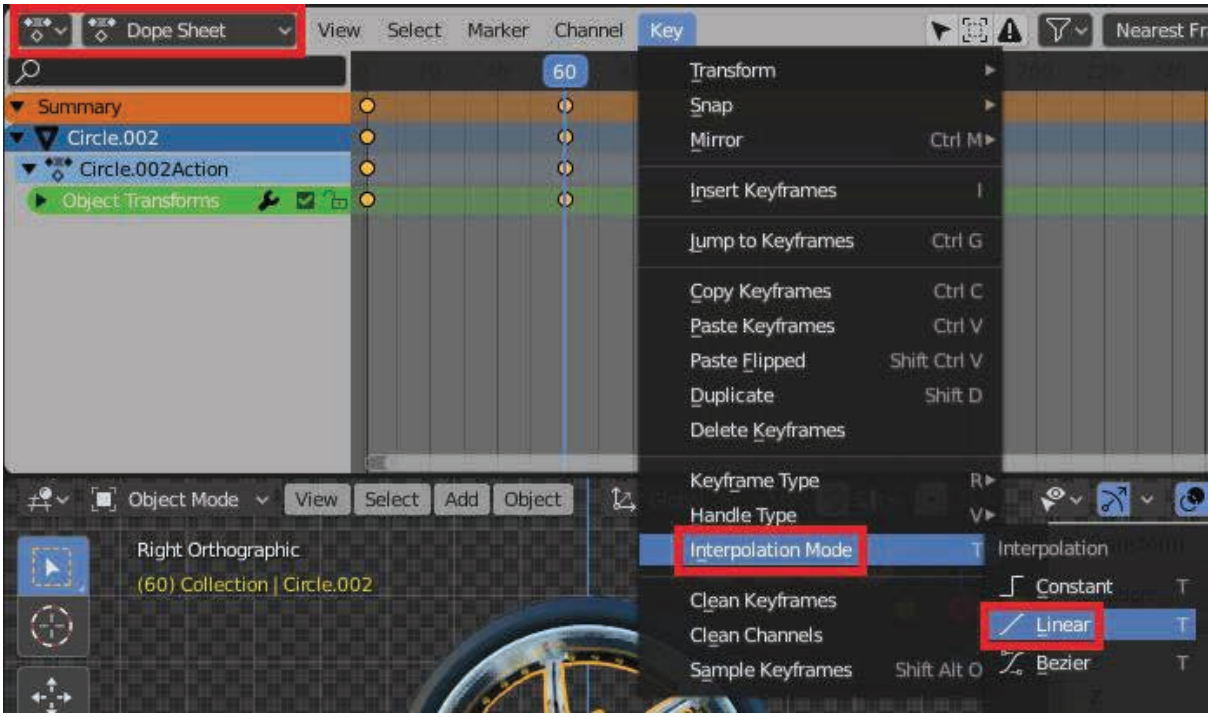


Press Shift and Left-key to go back to the first frame, and **press Spacebar** to play the animation you've recorded so far.

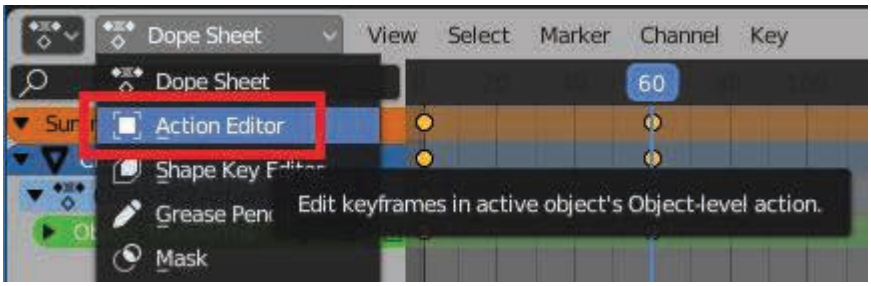
You'll notice the rotation starting slowly and slowing down at the end (this is called "easing" in animation). Normally, that's nice, but this won't look good when looping the animation of a continuously rotating wheel, so we need to fix this.

In the Dope Sheet, make sure the keys of the wheel animation are selected. **Press A** to select them all if necessary.

Go to **Key > Interpolation Mode** and choose "**Linear**". If you play back the animation now, you will notice that the rotation starts and stops abruptly. That's good, we're almost there!

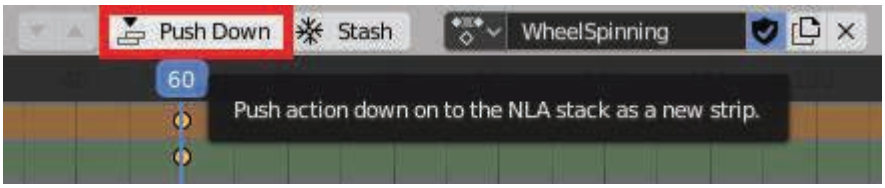


In the Dope Sheet drop-down menu, select the **Action Editor**.

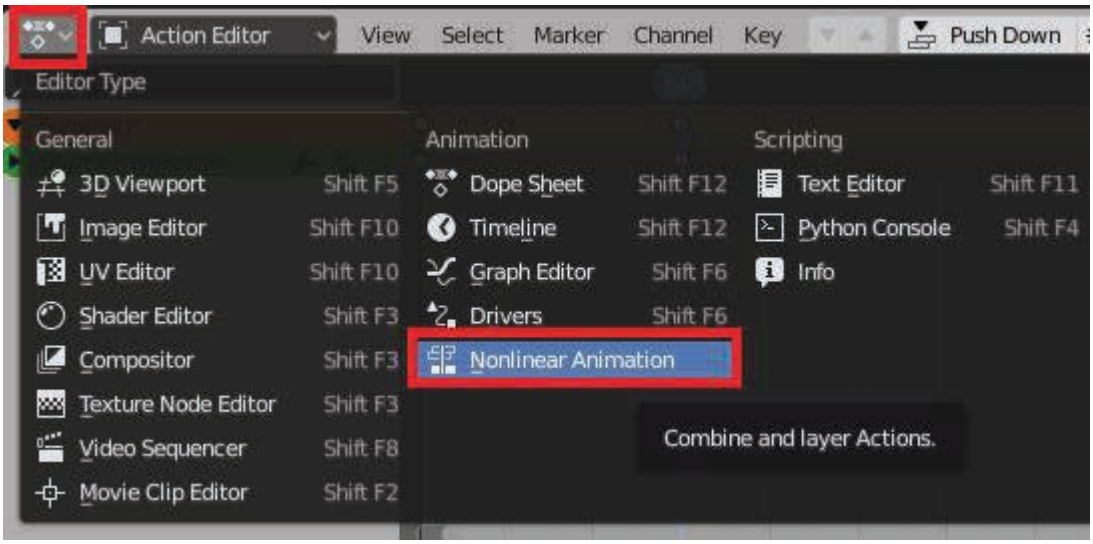


In the Action Editor, change the name of the action to something recognisable.

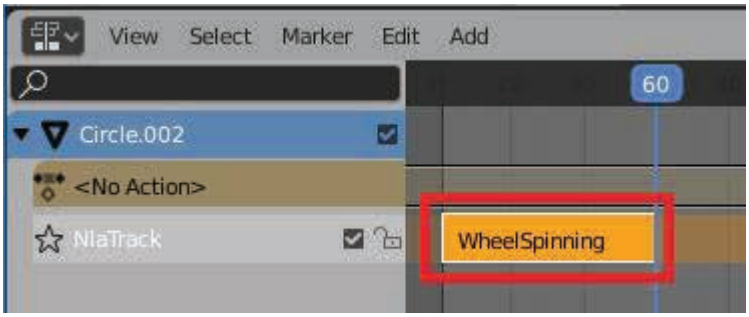
And then press that “Push Down” button. This “pushes” the action to the NLA (non-linear-animation) Editor as a strip.



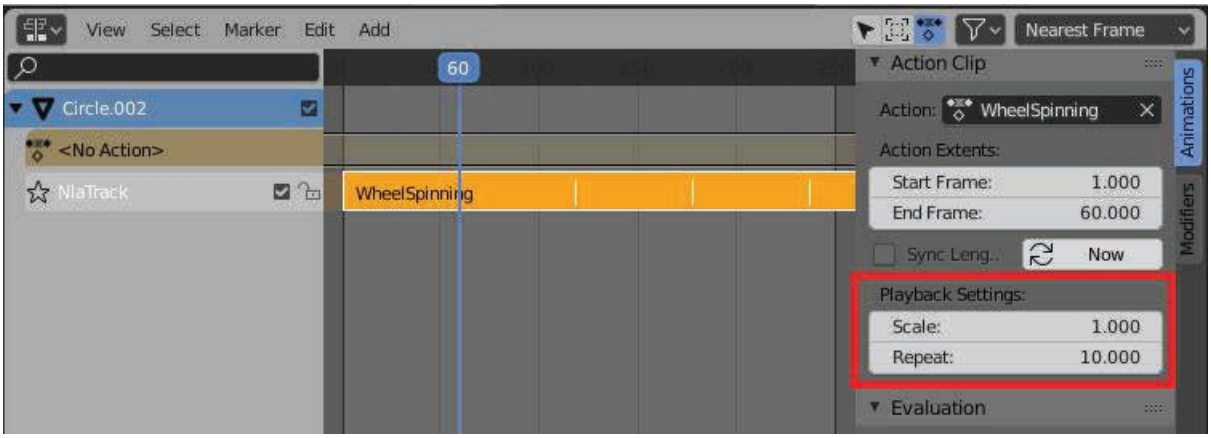
Now go to the NLA Editor (called “Nonlinear Animation” in the Editor Type drop-down menu).



In the NLA Editor we see that our Action which I named “WheelSpinning” is there as a strip. This strip is comparable to a video clip in video editing software. You can move it so it starts at a different place in time, and make it longer and shorter for example.



In the Option panel (**press n** to open it if you don't see it) you can now change the amount of repetitions of this action, as well as the speed (called "scale").



A repeat value of 10 like the above example, will repeat the action 10 times.

A larger scale means slower animation, a slower scale means faster animation.

For example, a scale of 10 will make the wheel rotate slowly, whereas a scale of 0.01 will make it rotate very fast.

That's it! It might seem difficult at first, but I recommend repeating these steps a few times to learn them. Once you've mastered this, you'll see that the world of "actions" is very powerful to use in animation, and is actually a very creative way to make animation.

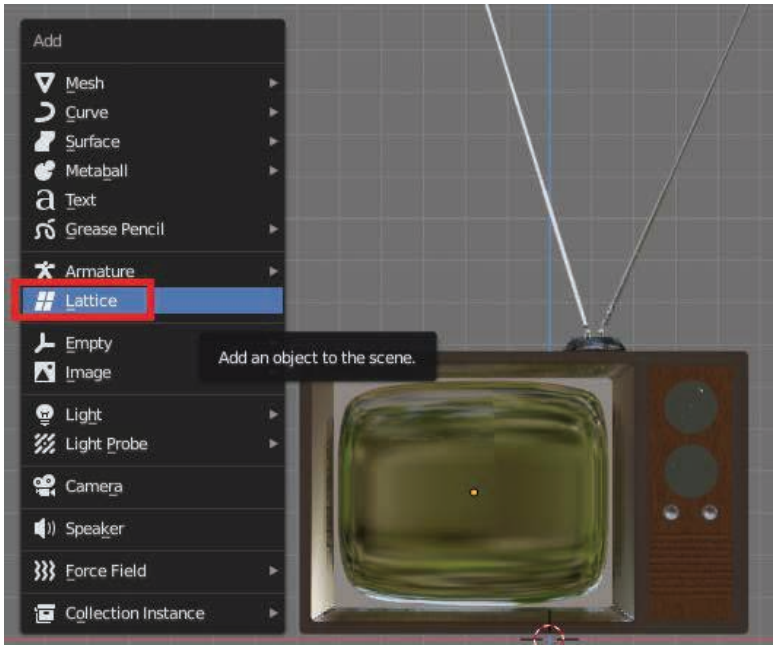
A walk cycle is another good example of where this repetition and speed control of actions will come in handy.

Thanks to Fenris on Blendswap.com for this nice tire model.

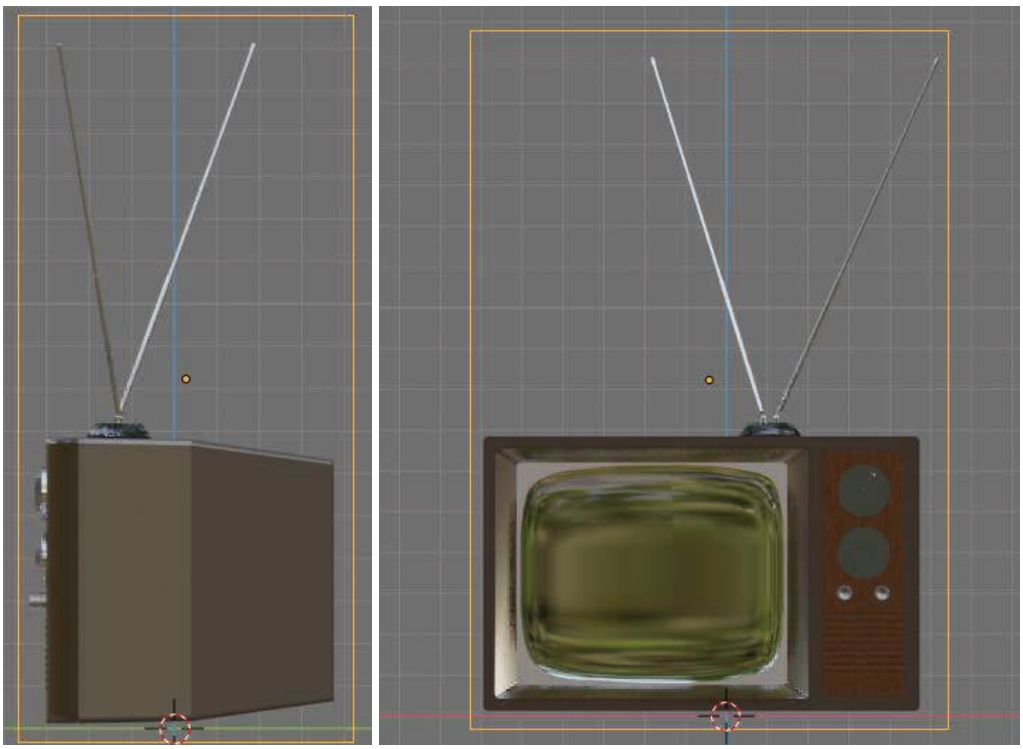
Easily Animate Objects With Lattices

Let's animate this cute TV! It would be challenging to use an armature with bones... but there is an easier way! That way is to use a Lattice object to deform it.

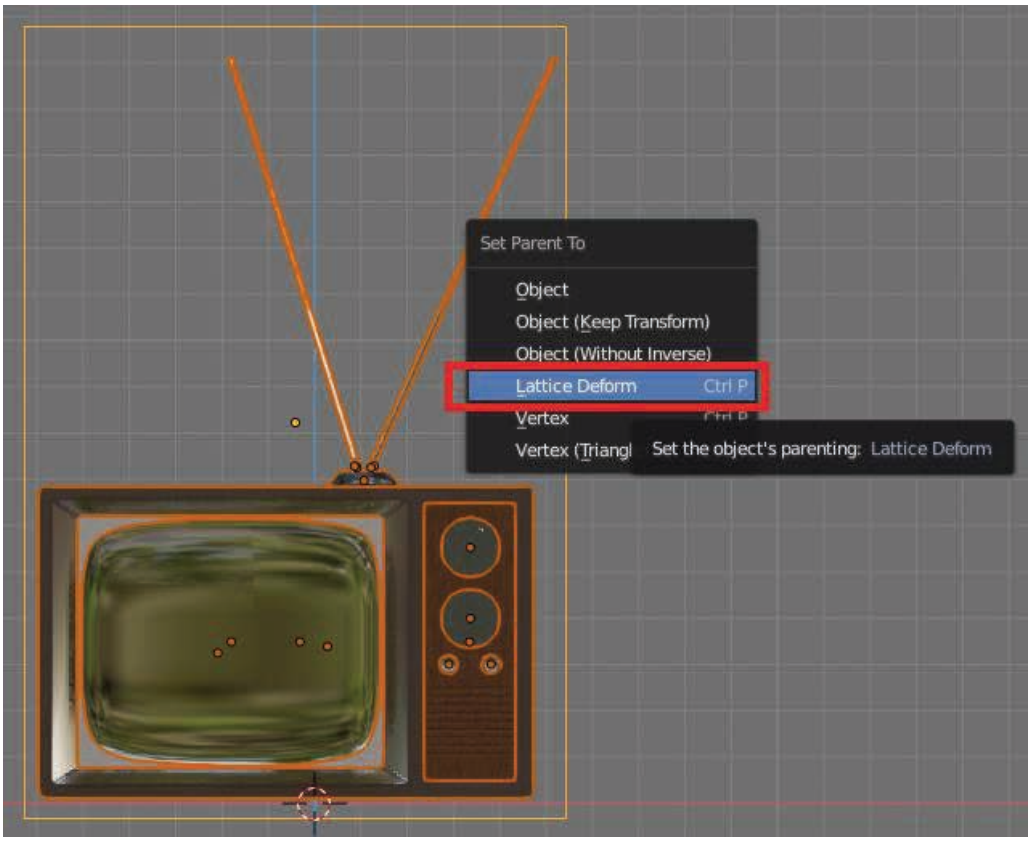
To create the Lattice object, **Press Shift+A**, choose **Lattice**.



Scale it in Object Mode around the object you want to animate. Don't scale it in Edit Mode! It won't behave correctly. Scale it to fit around the object you want to animate.

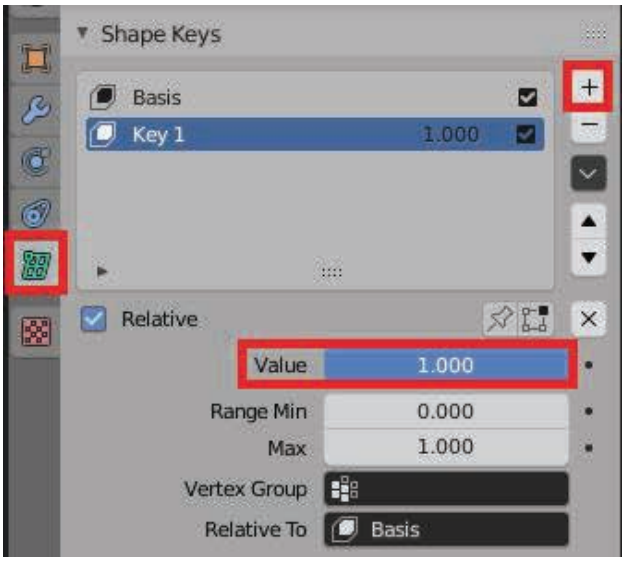


Select the object or objects inside the Lattice, then while **holding Shift** select the Lattice as well and **press Ctrl+P.** Choose "Lattice Deform".



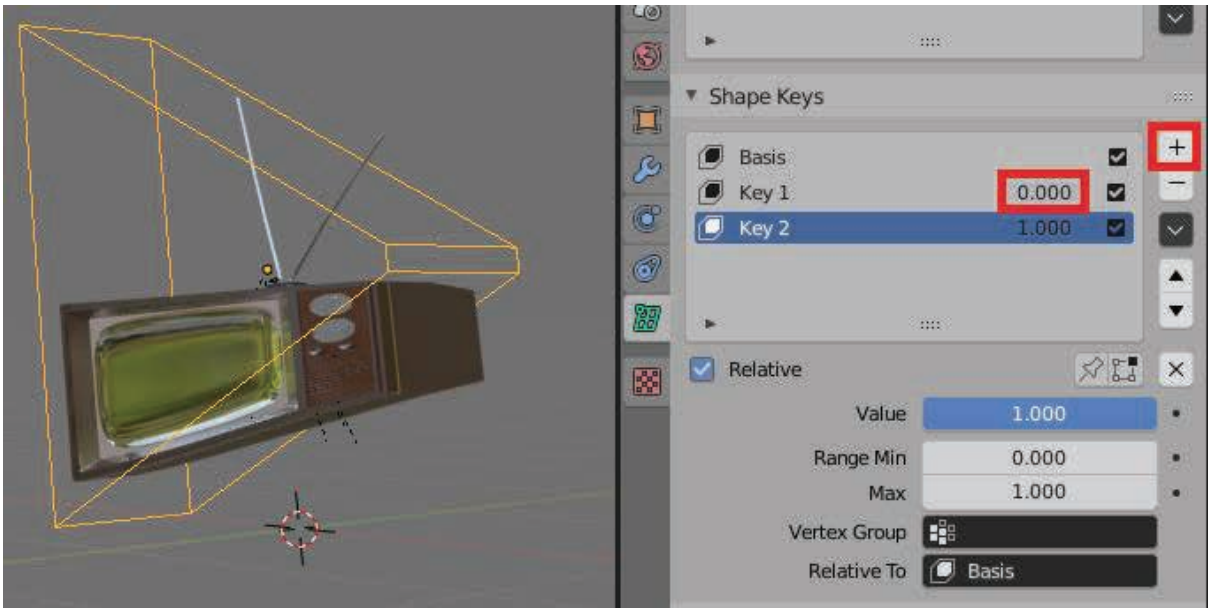
Parenting the object to the Lattice.

You can now deform the TV model by moving the vertices of the Lattice object in Edit Mode. To animate the TV we can create a number of Shape Keys for the Lattice object. In Object Mode with the Lattice selected create the "Basis" Shape Key by pressing the + button. Create another Shape Key ("Key 1") by pressing the + button again. Set its Value to 1.



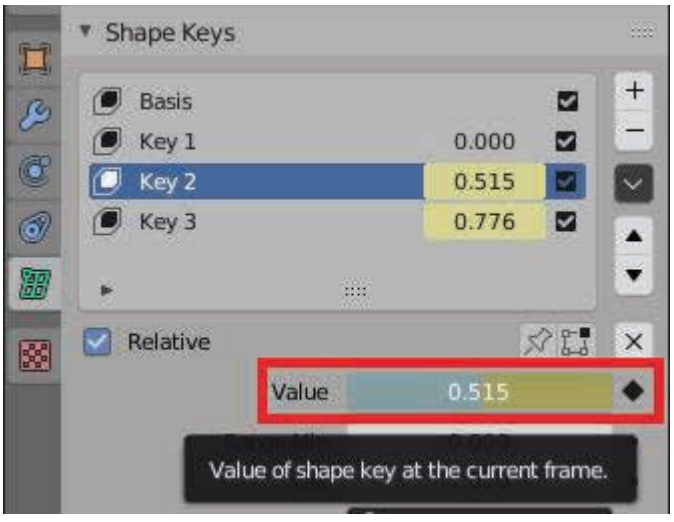
Edit the Lattice in Edit Mode to create the animation pose.

You can add another Shape Key by **pressing the + button**. For each Shape Key you can edit the Lattice for a different pose. Make sure the other Keys are set to zero when you do this.



You can add as many Shape Keys as you need.

Press i while hovering the mouse cursor over the Shape Key Value slider to set a keyframe. The slider will change color when a key is recorded. Alternatively you can press the diamond-shaped button next to the value, this also records the keyframe.



By inserting keyframes of different Shape Key values at different points on the Timeline, you create animation!

Television model made by “rblackwell” on Blendswap.com.

Animate A Fish Without Rigging

I can avoid time-consuming UV-unwrapping or rigging with complicated armatures, I will. not because I'm lazy, but because removing this kind of unnecessary technical busywork gives me more time for creativity, and for trying different things. The end result benefits from this. Of course, in some cases a decent UV unwrap or complicated armature is what you need. But often it is not really necessary.

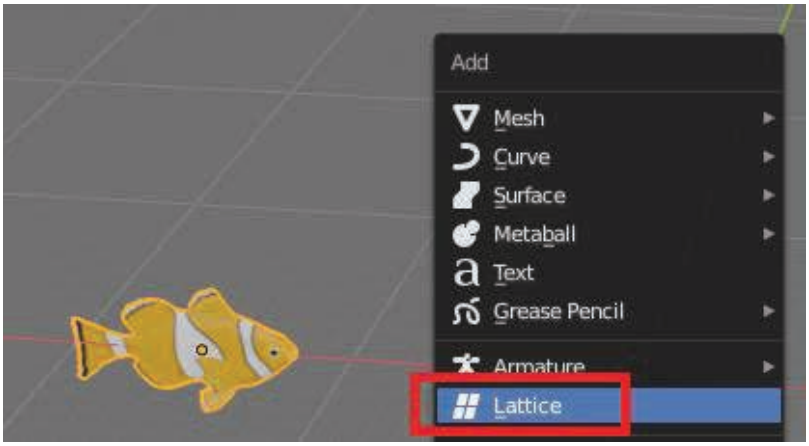
Let's create this fish animation.

Download or make a fish model (I'm using one from Blendswap). Set a couple of Location keyframes so it moves from one location to another. Don't worry about the fact that it is otherwise completely motionless for now.

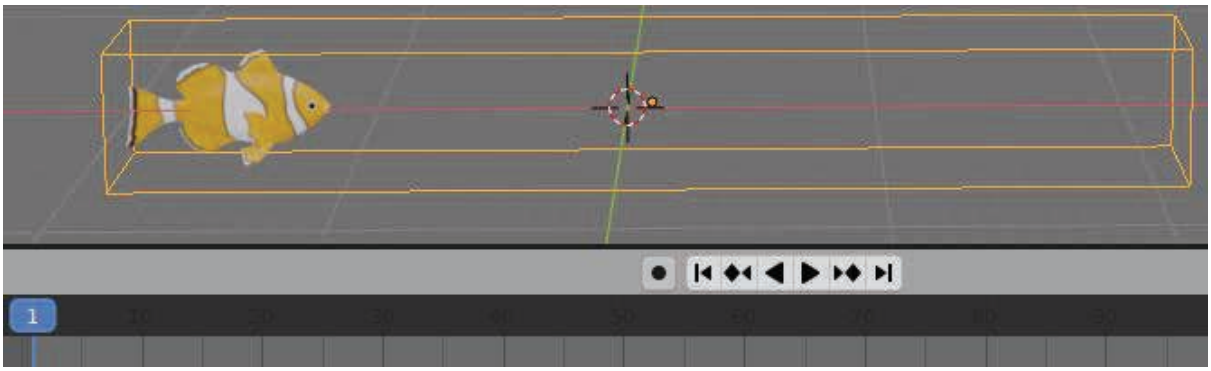
Press i to create a keyframe at frame 1. Choose "Location" as the value to create a keyframe for.

You can move the blue slider across the Timeline to frame 100 by grabbing its handle (left-click and hold) and moving it. **Press G + X** and move the fish along the X-axis and set a keyframe for its new location at frame 100.

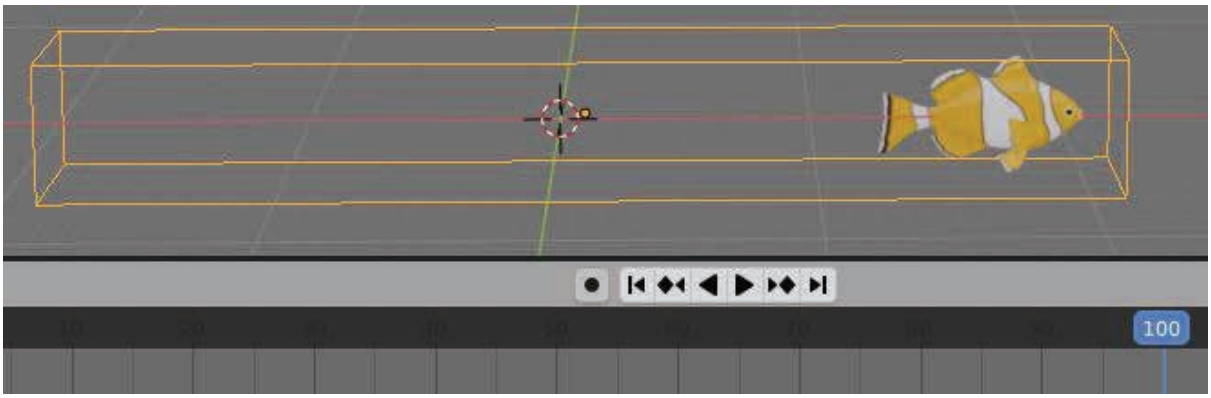
Tip: **Press Shift+[←]** to go backto the first frame. **Press Spacebar** to play. **Press Shift + A** and choose **Lattice**.



Scale the lattice in Object Mode (definitely not in Edit Mode!) around the fish and in front of it (around where it will move when it moves forward).

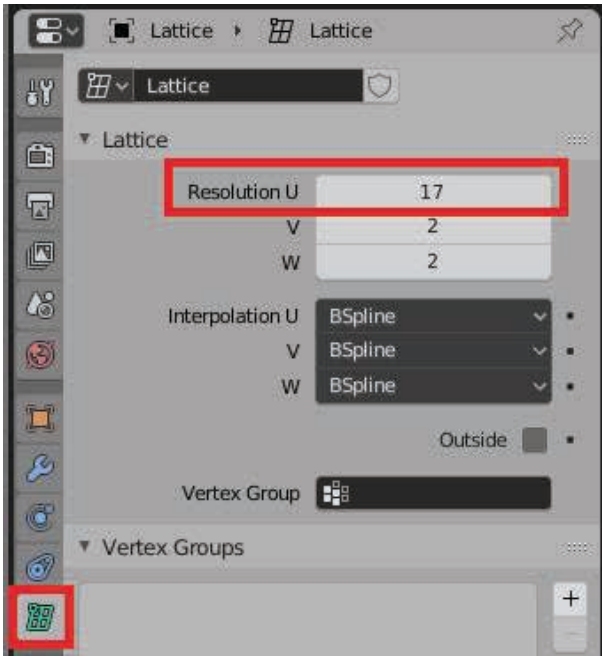


At frame 1.



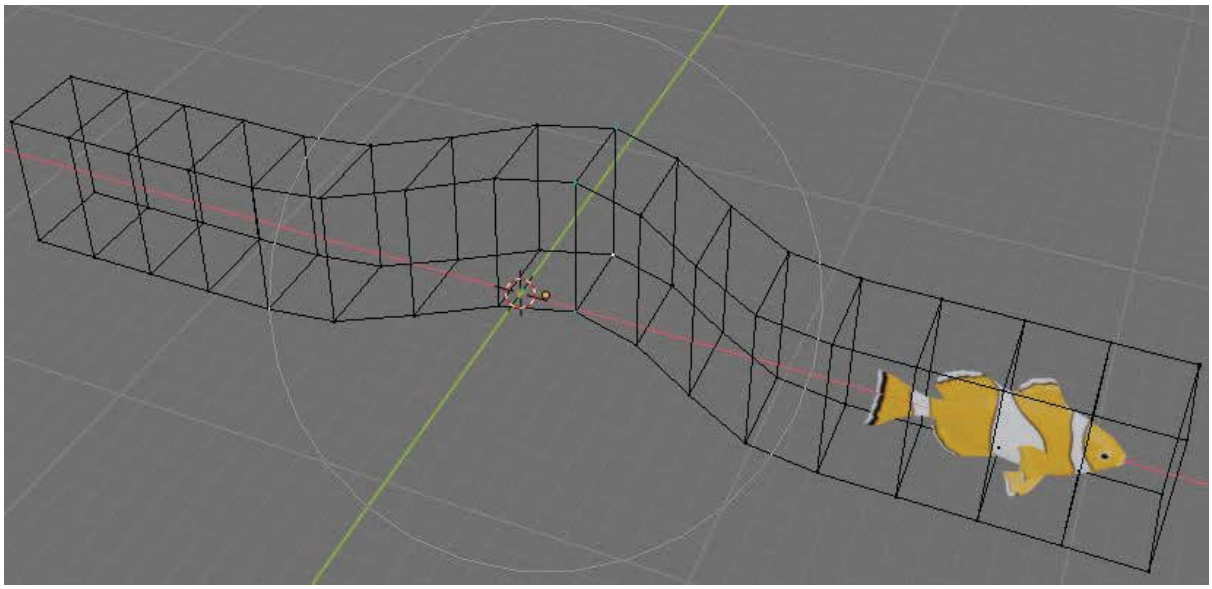
At frame 100. The Lattice is scaled to fit the trajectory of our heroic fish.

Add some more resolution in the U field in the Lattice tab. That way we can deform it better.



In Edit Mode, select the vertices of the Lattice and move them around. Note, that you won't have the traditional Face/Edge/Vertex options here, it's only vertices for Lattice objects.

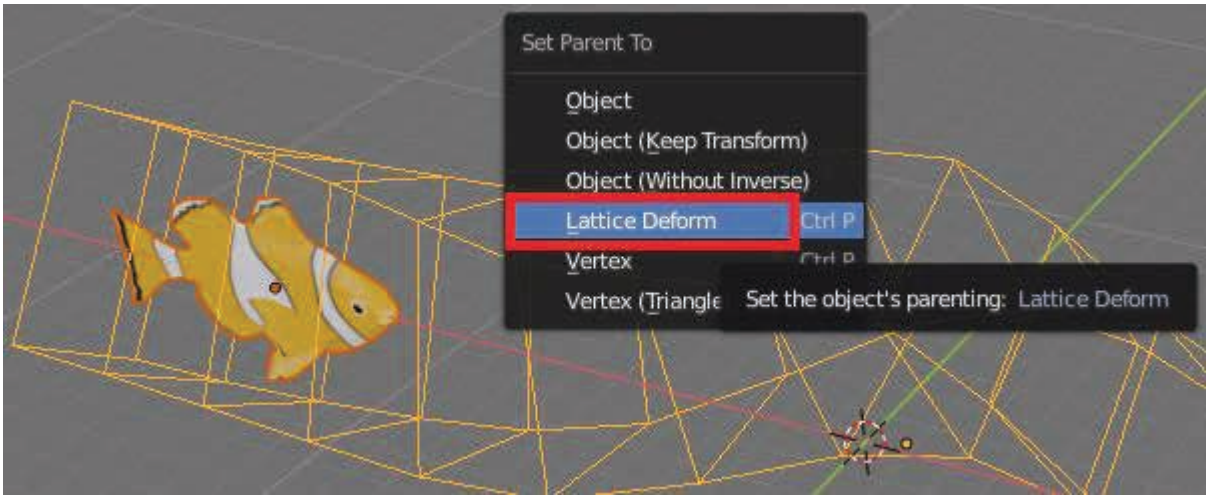
Press "o" to activate Proportional Editing and use the middle mouse button to change the sphere of influence. This way you can edit it in a more organic way, affecting several vertices at once.



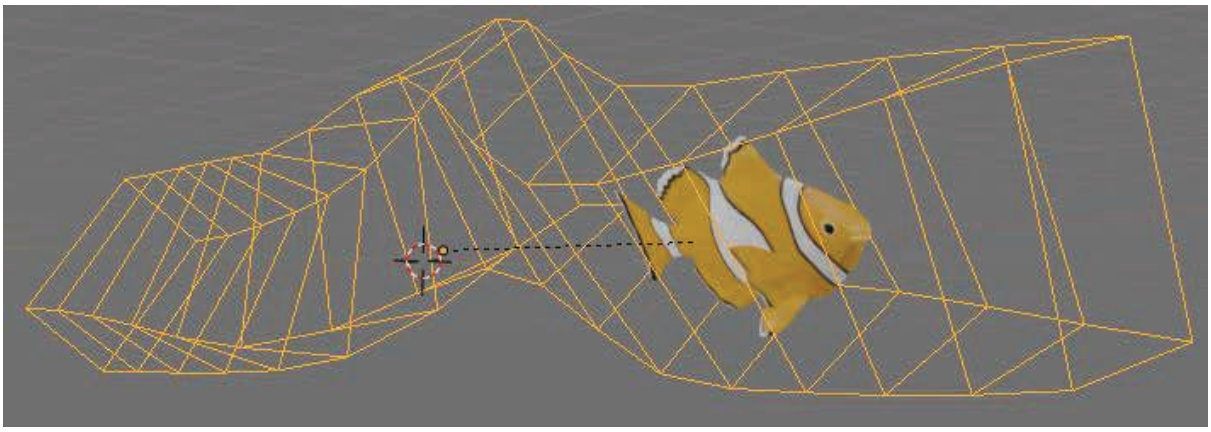
Dragging the middle mouse button to increase the sphere of influence of proportional editing.

When this icon in the top row of the viewport is blue, Proportional Editing is turned on. Toggle it on/off by **pressing "o".**

Select the fish and then the Lattice, **Press Ctrl+P** and choose "Lattice Deform".



Press Ctrl+A to view the animation.

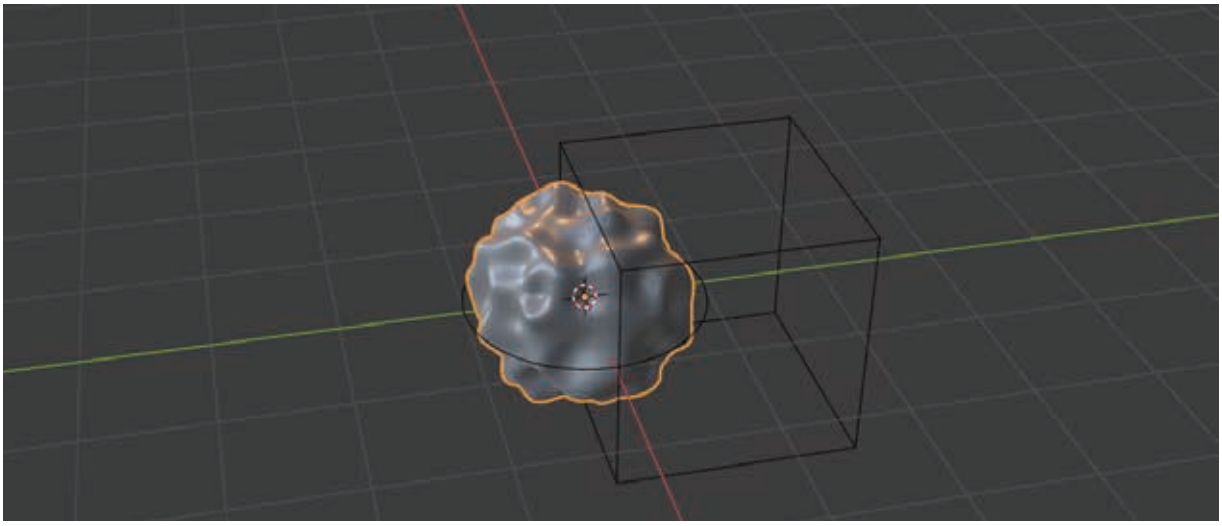


Look at him go! Time to make our own off-brand straight-to-streaming Pixar knock-off… I mean homage.

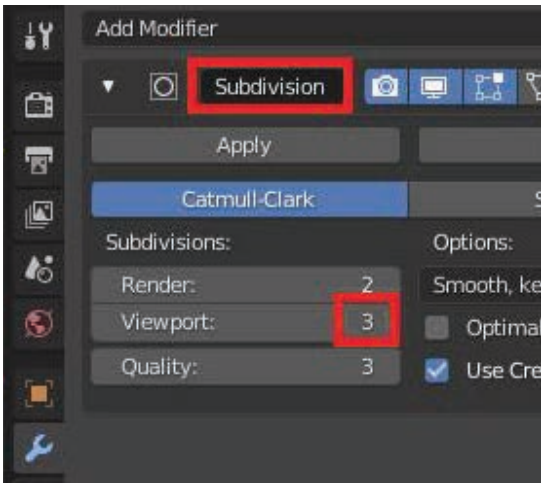
Clownfish model by "sumeetsaxena" on Blendswap.

Using Empty To Animate Displacement Modifier In A Loop

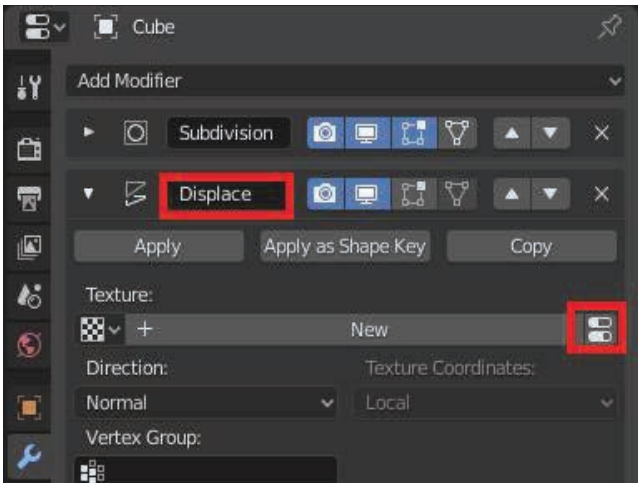
Let's look at a method for animating displacement in a perfect loop.



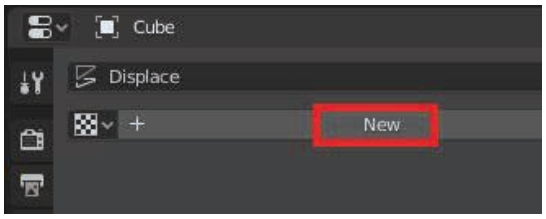
Add a **Subdivision modifier** to the default cube. Subdivide a few times, 3 is a good start.



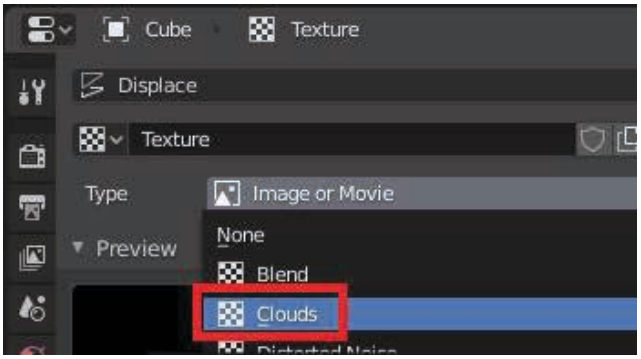
Next, add a **Displace modifier** to the cube. Click on the icon next to the Texture slot to go to the Texture settings.



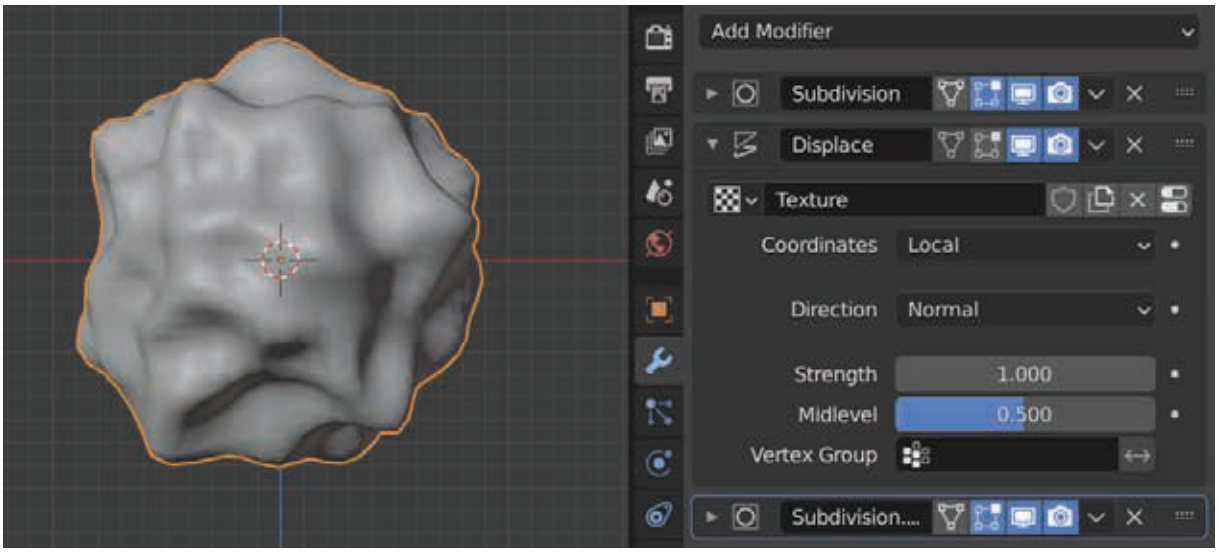
Add a Cloud Texture by **clicking on New** in the Texture tab.



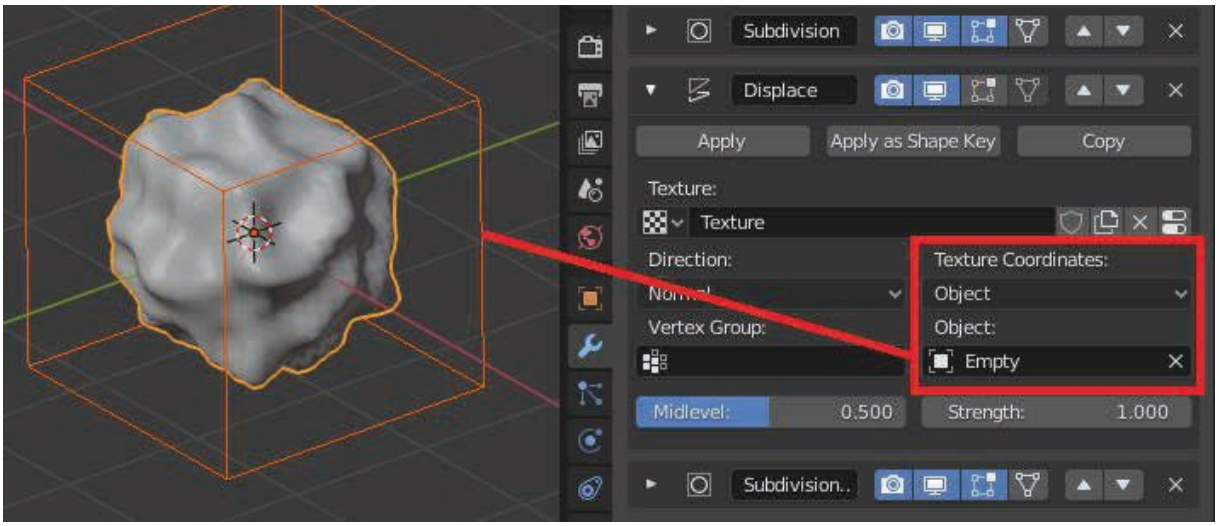
From the drop-down menu of the procedural textures, choose a **Clouds texture.**



Right-click on the cube and set it to **Shade Smooth**. Add a second Subdivision Modifier under the Displacement Modifier (subdivision level: 1).



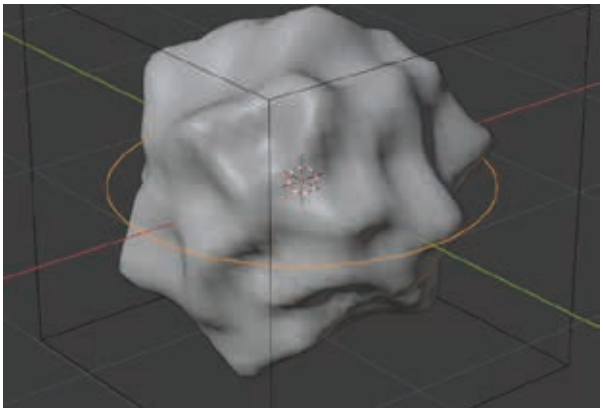
Add an empty to the scene. In the displacement modifier under Texture Coordinates choose “Object”, and as the Object choose the Empty. Note: I’m using a Cube-shaped Empty here, but the shape is irrelevant. Just choose what’s easy to see.



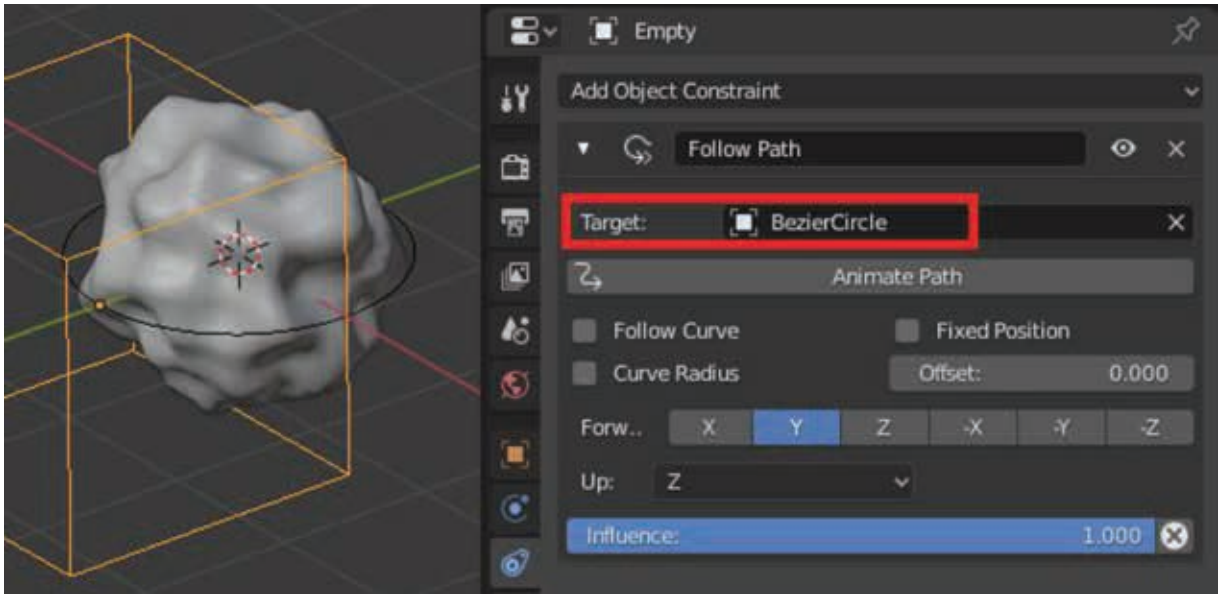
Try moving and rotating the Empty object and you’ll see that the Displacement effect starts to move.

Now we have to make this movement loop continuously.

Press Shift+A and add a Curve > **Circle**.



Add a Follow Path constraint to the Empty object and choose the Circle as it's **Target**. Now if you **press the Spacebar** you'll see the Empty following the Circle and the displacement on the object will be animated.

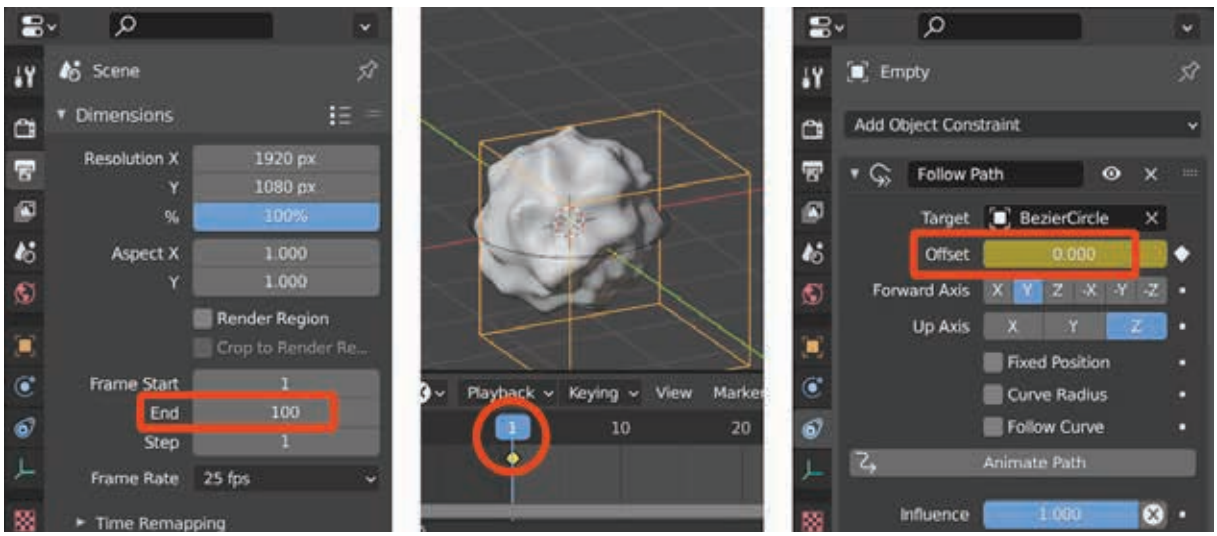


To make it loop perfectly we need to set some keyframes.

Set the **End** to 100 so that we have 100 frames in total.
I set the Frame Rate to 25 (the default frame rate in Europe).

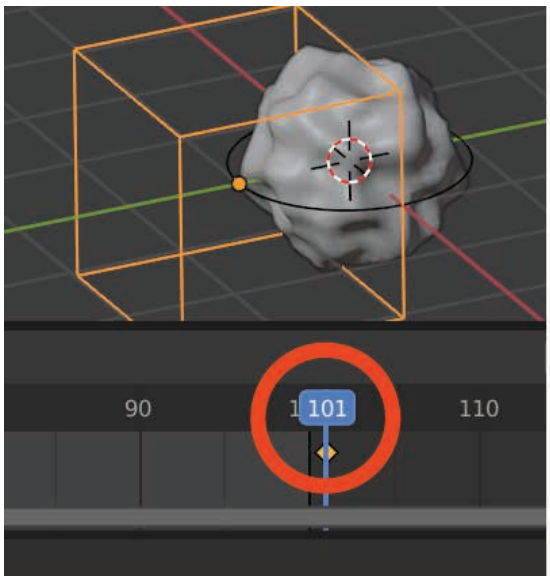
Now go to frame 1 (shortcut: Shift + left arrow key), and set the **Offset** value in the Follow Path constraint **to 0** as well.

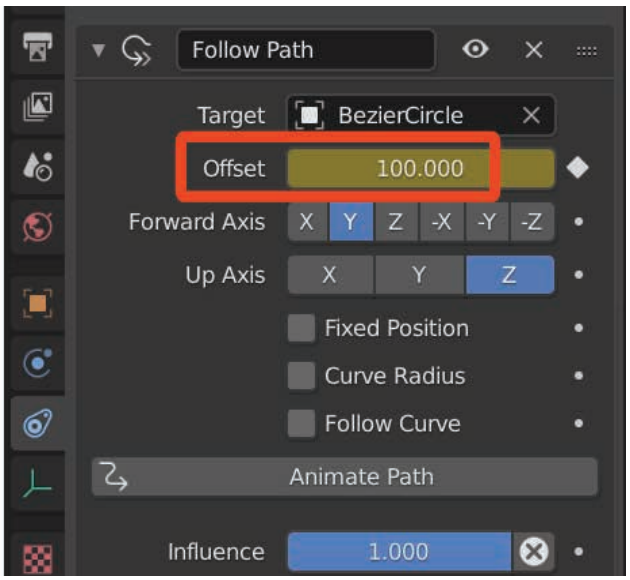
Add a keyframe by **pressing i** with the mouse cursor above the value field. It will change color to a kind of yellow-green.



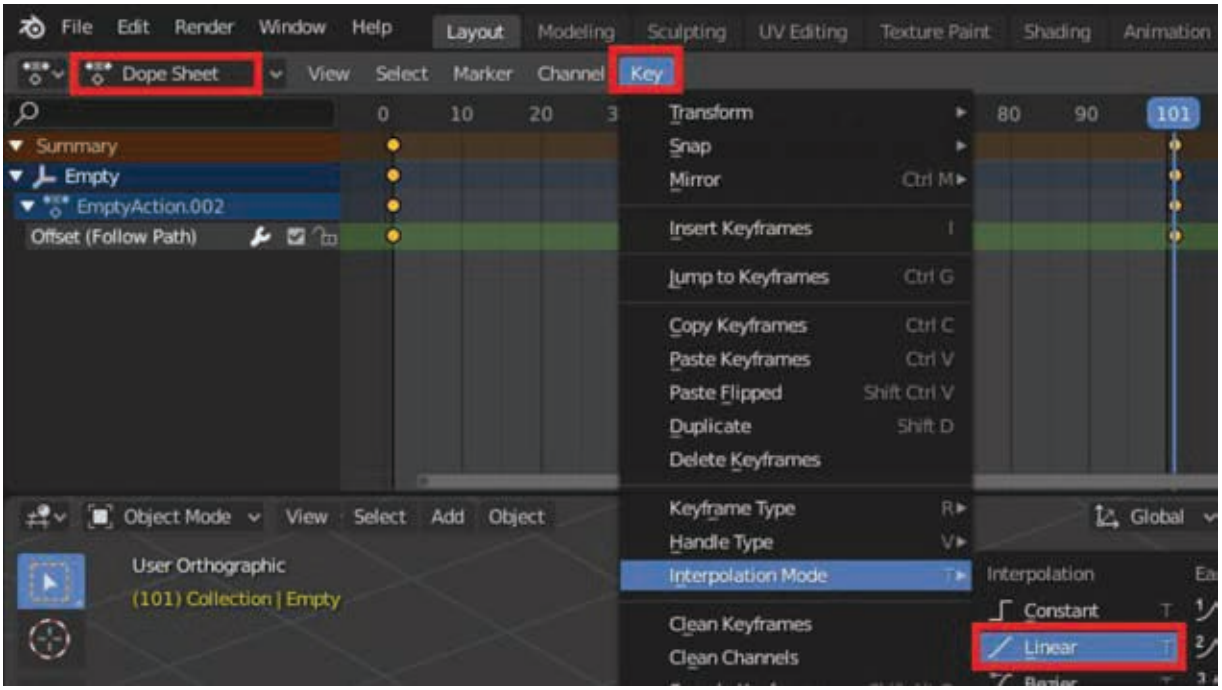
Now, **go to frame 100** (keyboard shortcut: Shift + right arrow key) and then **go one frame further** by pressing the right arrow key once. Now you're on frame 101.

Type **Offset value 100** and **press i** with the mouse cursor above that field to set a second keyframe. By setting the last key one frame *after* the last frame in the playback range, we make sure the loop is continuous.



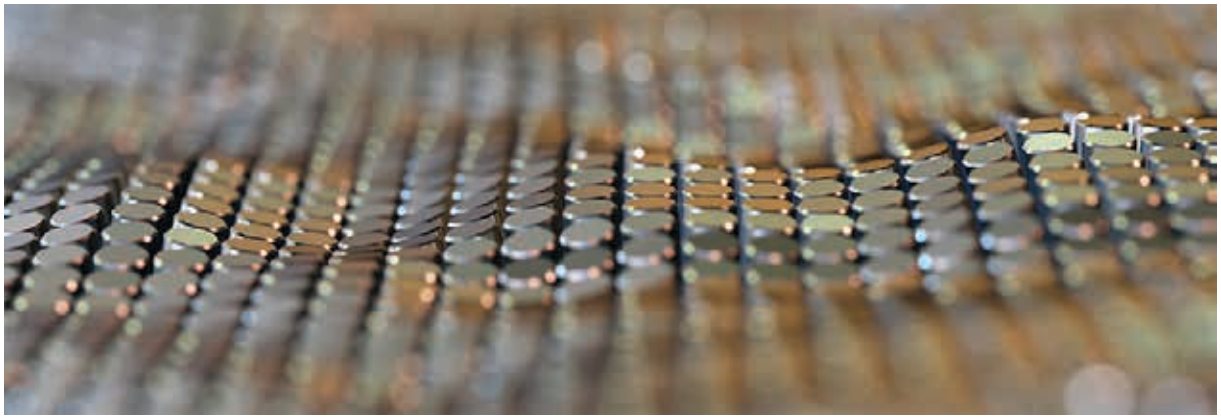


Set the Key Interpolation Mode to **Linear** in the Dope sheet so the animation loops at constant speed. Press **Spacebar** to play the animation.

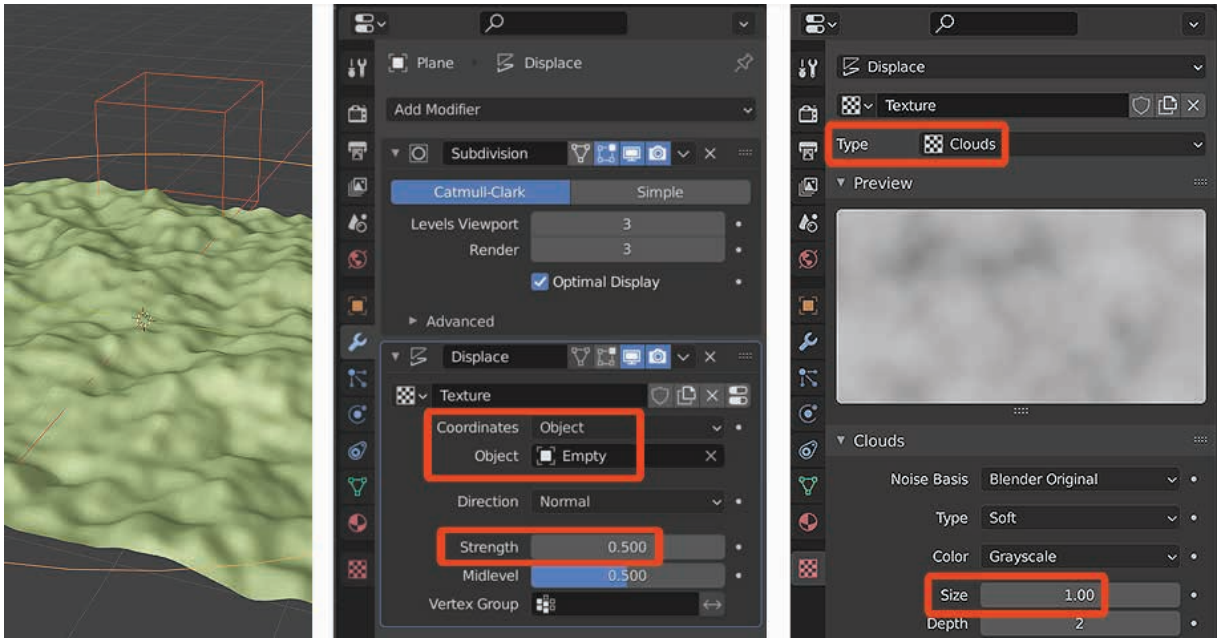


Dupliverts For Motion Graphics

Previously, we looked at how to set up a kind of looping displacement effect. This tip builds further on that effect. Instead of a Cube, we are using a **Plane**. An **Empty** that is constrained to a Circle creates the Displacement of the Plane, as explained in the previous chapter.

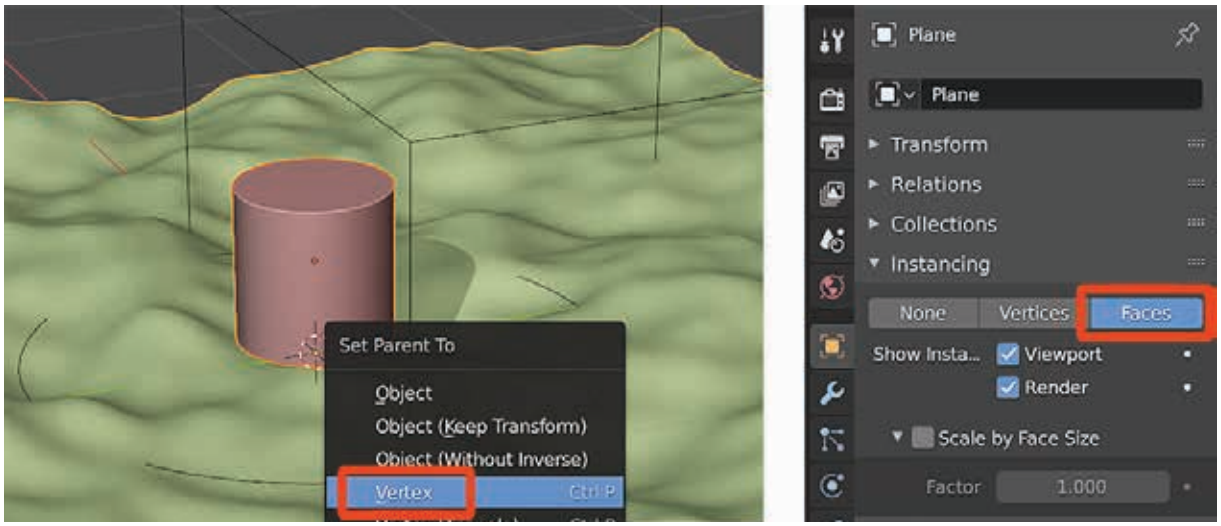


A Plane primitive is scaled up a bit and subdivided a few times in Edit Mode. A **Cloud texture** is used on a **Displacement Modifier**. The specific values I used (indicated below) are not so important - you can quickly the displacement simply by scaling the Empty. However it is important that when rendering your Subdiv level for rendering is the same as the Subdiv level for the viewport.



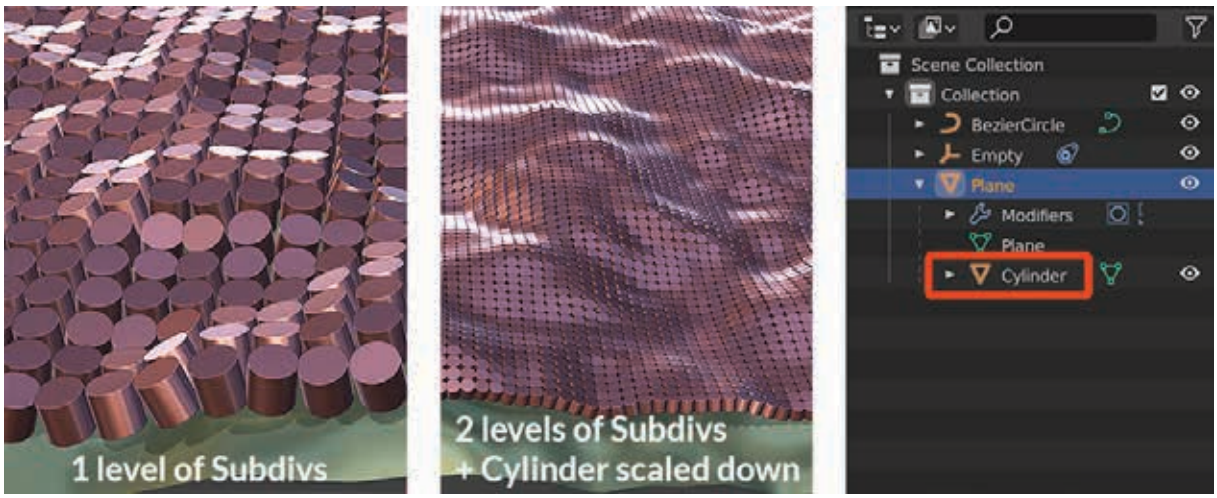
Create any mesh in Object Mode. In this case I added a **Cylinder** and set it to **Shade Smooth** and turned on **Auto Smooth**.

Select the Cylinder, and **holding Shift** select the plane as well. **Press Ctrl+P** to Parent and choose "**Vertex**" from the options. Select the plane, go to **Instancing** and choose Verts or Faces.

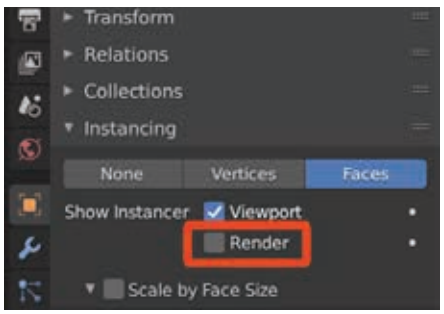


Now, depending on the scale of your Cylinder you may very well have to scale the original one up or down. Also, you can reduce the **Subdivision count** to have less instances.

If you can't find the original Cylinder, expand the Plane (which it's parented to) in the Outliner to find it. To control the scale of the displacement, you can select the Empty and scale it up or down.



If you don't want to see the Plane in your render, uncheck "Render" in the Instancing menu. The original Cylinder won't be visible in the render.

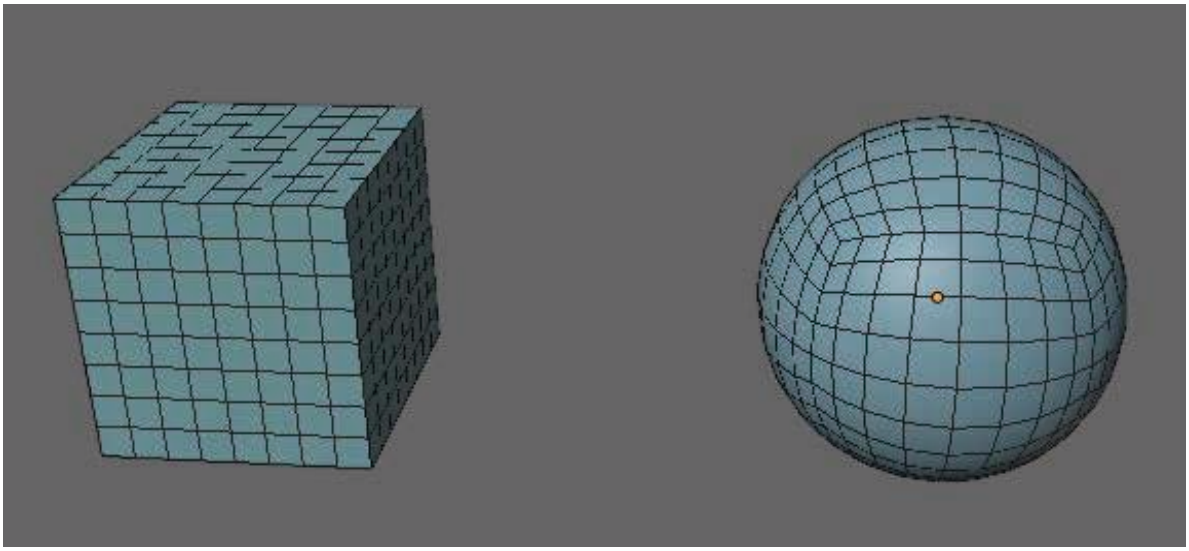


If you enjoy this kind of animation check out Midge "Mantissa" Sinnaeve's Youtube channel. He goes very in-depth with lengthy videos into this kind of animation.

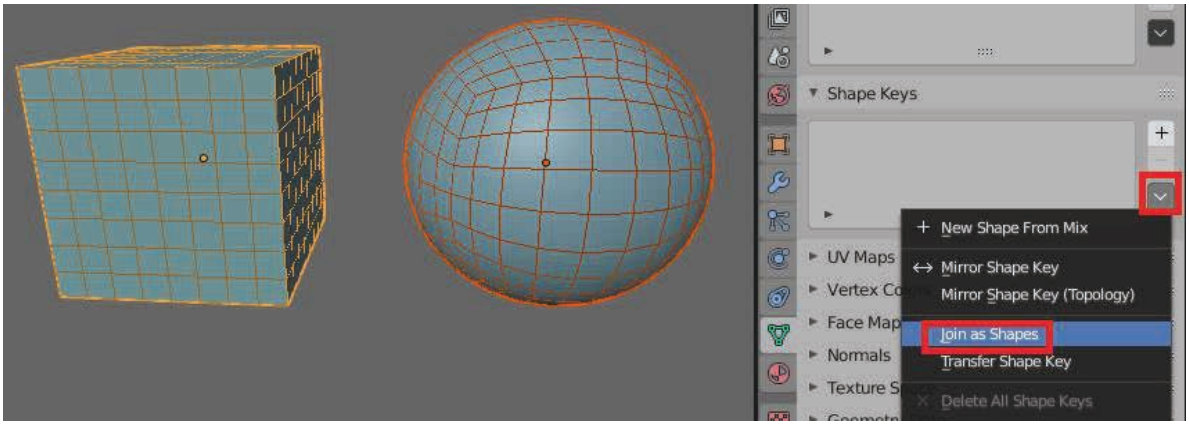
Shape Keys: Join As Shapes

Join As Shapes allows you to add Shape Keys to an object using different objects. This may be useful for example, if you have created different facial expressions from the same base head model, and they are all separate objects but now you want all those Shape Keys to be available on just one model.

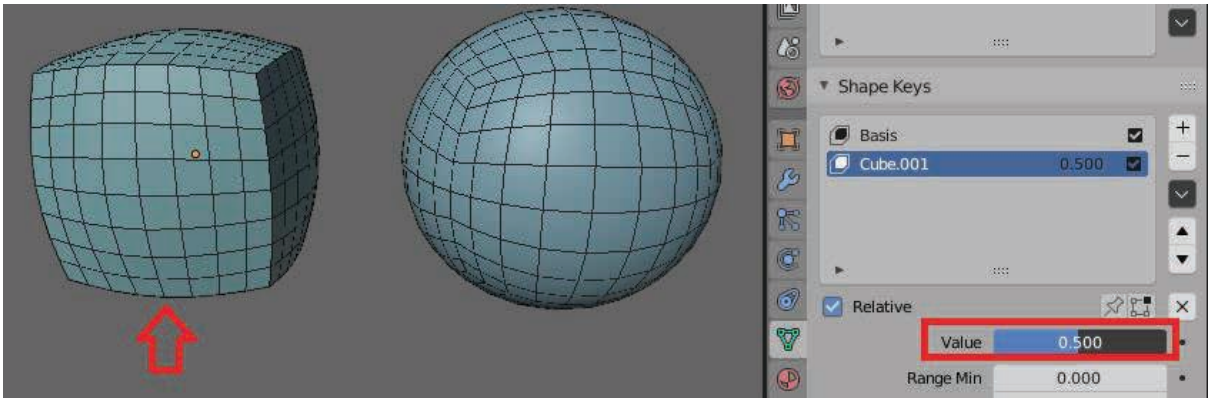
For this to work two objects need to have the same vertex count.



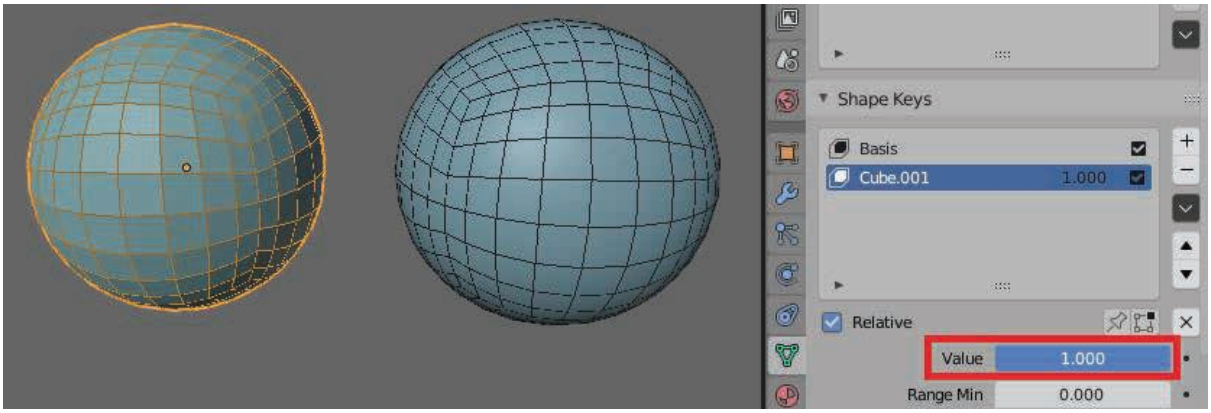
In the below example I want the Cube to be able to be spherical with a Shape Key. First I select the Sphere and while **holding Shift**, select the Cube, then in the Shape Keys menu click on the down arrow icon and choose **Join As Shapes**.



This creates a new shape key, named after the object used to create it, in this case the sphere. You can now use the Value slider to control the Shape Key value and blend between the two shapes.



With a Value of 0.5, only half the Shape Key is active, so the cube is only half-spherical.

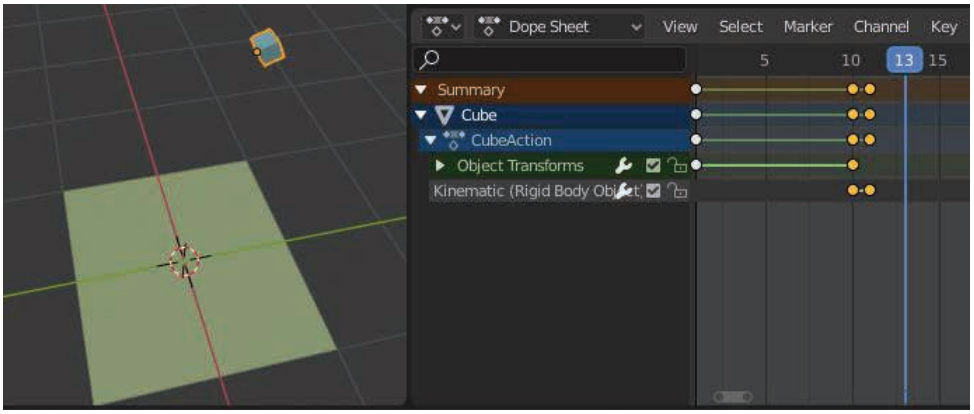


With a Value of 1, the Cube is now completely Spherical.

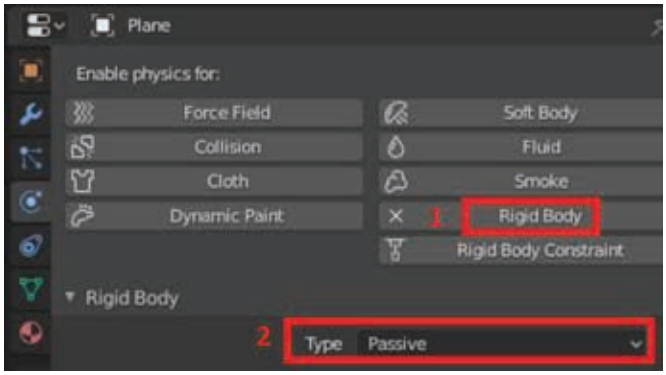
These Values can be keyframe-animated, and many more Shape Keys can be added. This is how you can add more facial expressions to your character for example.

How To Launch A Cube

We can combine keyframed animation and physics simulations for interesting results. For this example, we'll launch a cube and then have a gravity simulation take over to make it stop going up slowly, and fall down.



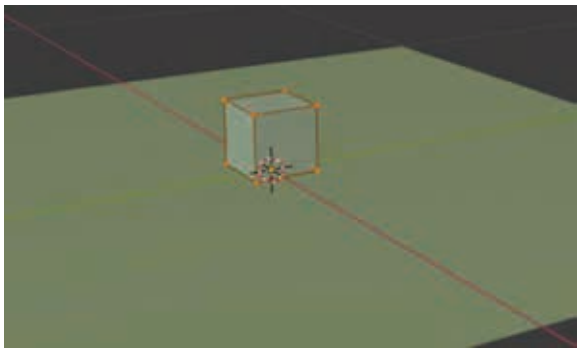
Create a plane by pressing **Shift+A** and choosing Mesh > Plane.



Enable **Rigid Body** physics.

Set Type to **Passive**. This ensures that it doesn't fall down when the simulation starts and that it can interact with other rigid body objects.

Press Shift+A and choose Mesh > Cube to create a cube.

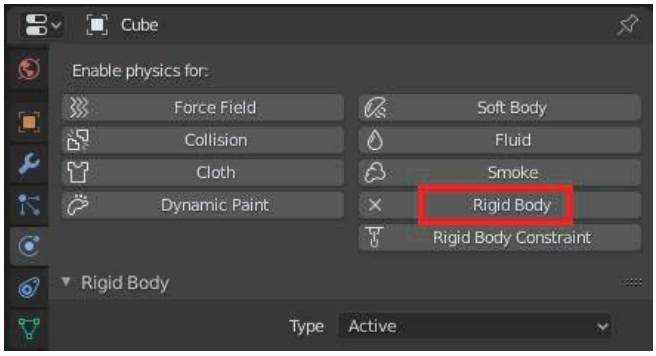


Press **TAB** to go into Edit Mode and scale the cube down.

Move it up above the plane.

Press **TAB** to go back to Object Mode.

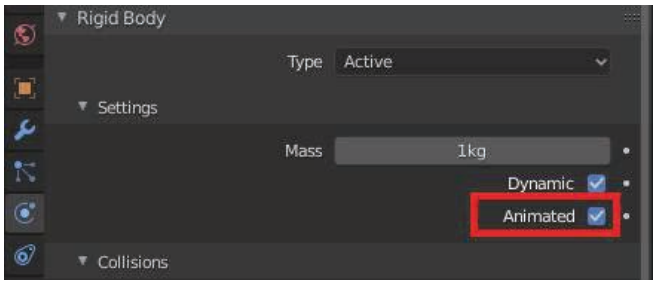
Enable **Rigid Body** physics for the cube. Leave it as **Active** Type.



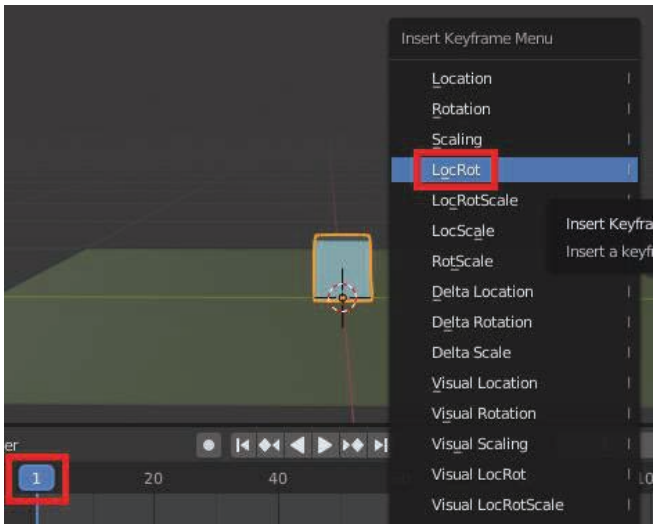
Check the Animated box. Otherwise we won't be able to add

keyframes to it for movement.

At frame 1, **press i** to set a keyframe for the cube.



Choose **LocRot** to set it only for location and rotation of the cube.



Go to a later frame (like 60) and move the cube up on the Z-axis by **pressing G Z** and moving the mouse.

Rotate it slightly on the X-axis by **pressing R X** and moving the mouse.

Move it a little to the side it's rotated towards by **pressing G Y** and moving the mouse.

Set another LocRot key.

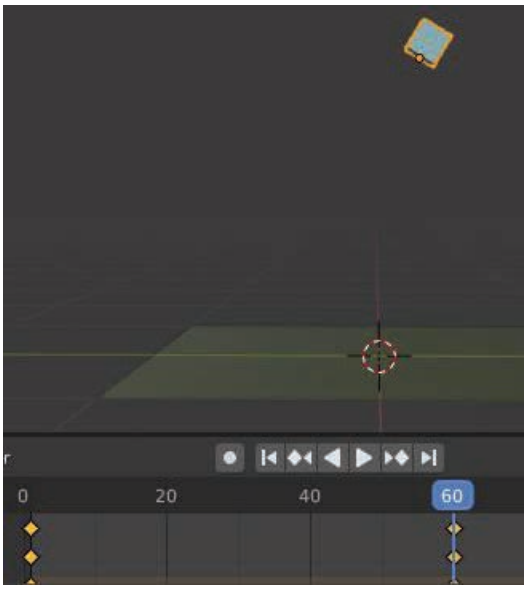
In Blender you can keyframe almost everything, even checkboxes.

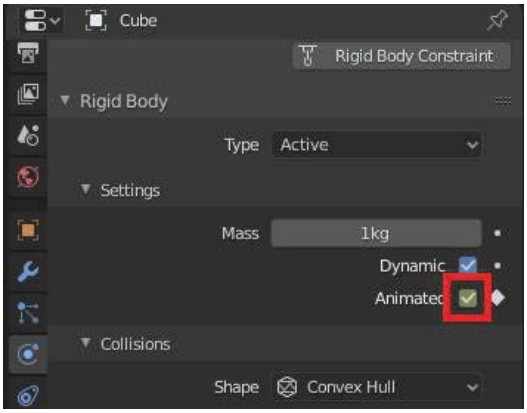
While mouse hovering over the **Animated** checkbox **press i** (don't *uncheck*it - just press i to keyframe that it is still checked ON at frame 60).

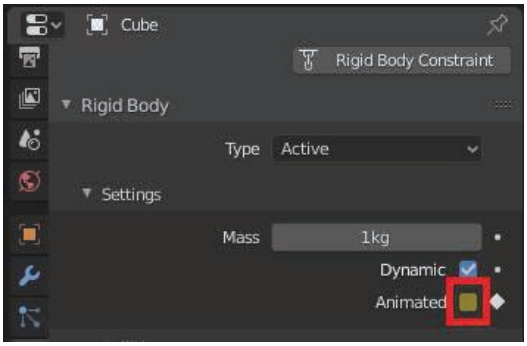
You will notice that it has changed color to a kind of yellow. That means there's a keyframe at that frame.

Go to the very next frame (frame 61) by pressing the right arrow key. Uncheck Animated and press i again.

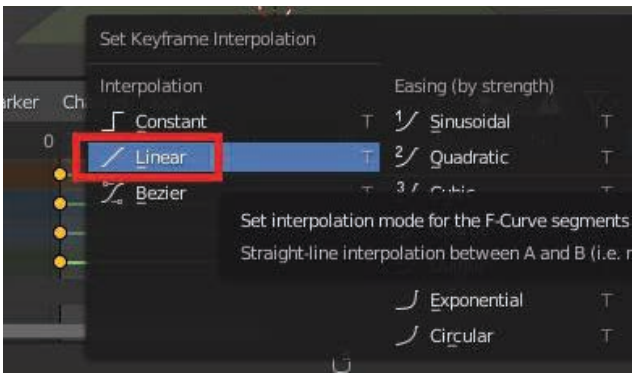
Now we've keyframed turning off the Animated checkbox at this point on the timeline and the simulation can take over.







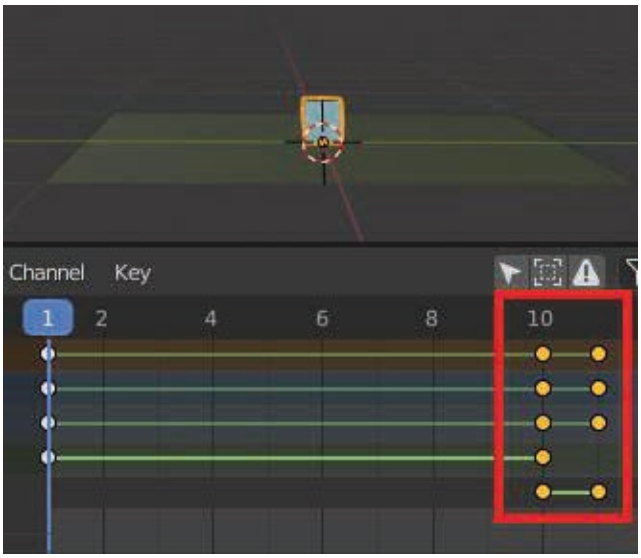
In the Dope Sheet, select all keyframes of the cube by pressing A (the cube needs to still be the active selected object).



Set Interpolation Mode to Linear by **pressing T** and choosing Linear.

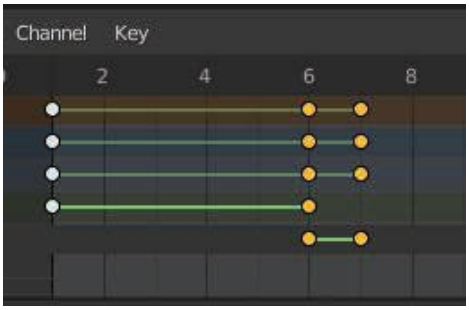
Now for some fun rocket science.

If you go to the beginning of the timeline (press Shift and left arrow) and press Spacebar to play the animation, you'll notice that the cube falls down like a brick after frame 60.



That's because it's initial velocity is too low. Grab the keyframes at frames 60 and 61 and move them closer to the beginning like frame 10.

Now you'll notice the cube has enough speed to keep going after the initial animation is finished.



Move those keys even closer to the start of the timeline and you'll increase how far your cube can fly.

Copy animation and Edit separately

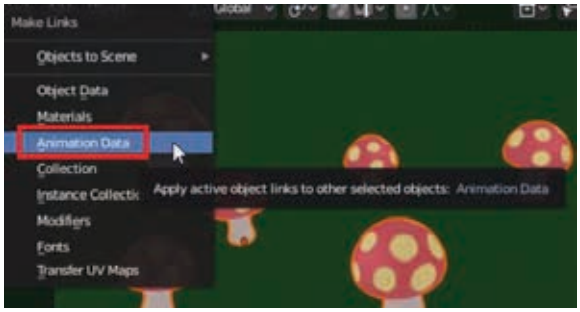
In case you have similar objects that need to have the same animation (like popping us like the mushrooms in this example), it's useful to just copy the Animation Data.

But what if you want to offset the animation between the objects?

Here we'll look at just how to do this. In this example, the mushroom at the top left has some keyframed scaling animation, the duplicates don't have any keyframes.

Select the duplicates that don't have animation first.

Then select the one that has animation already last. **Press Ctrl+L** and choose "animation data".

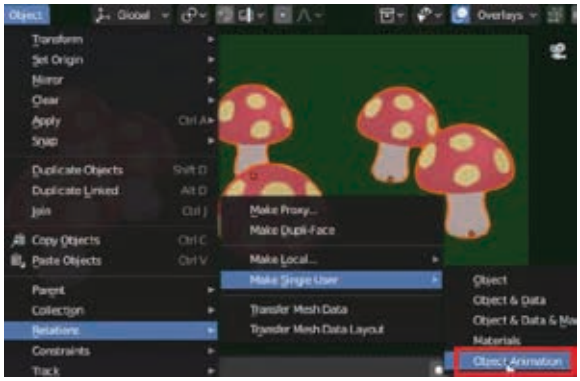


Click here to return to the index.

Now all the mushrooms have the same animation, and also the same timing.

Now let's look at offsetting the timing for each individual mushroom.

Select all the mushrooms, and go to Object > Relations > Make Single User > **Object Animation**.



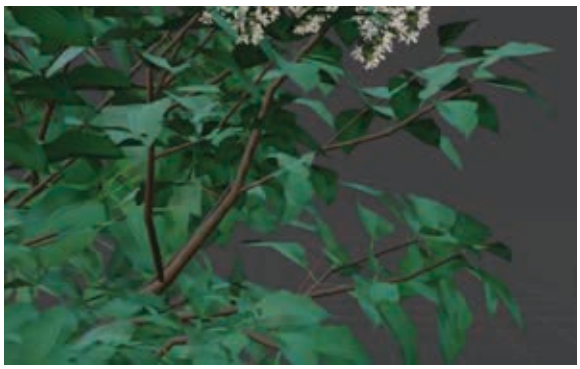
Then choose "All".

Now you can select any mushroom and move its keyframes in the dopsheet.

This way their animation is offset and they all move with their own timing.

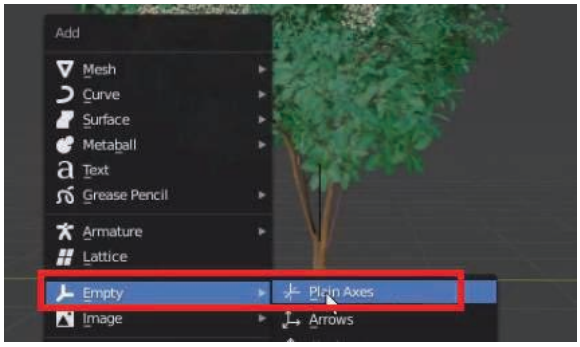
Easy animated leaves

Adding a little movement to the leaves of a tree adds a lot of realism to your animation and makes the scene more interesting.



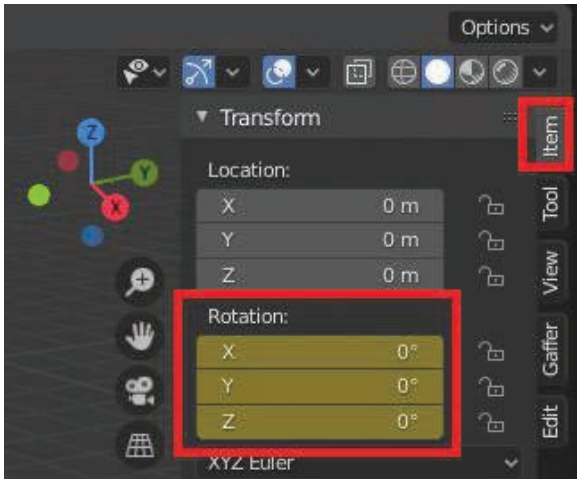
It's actually not difficult to do. Here we look at a case where the model was purchased from a model store like Turbosquid, and all the leaves are a single mesh instead of particles.

Add an Empty by **pressing Shift+A** and choosing **Empty > Plain Axes** (the type of Empty is actually not important).

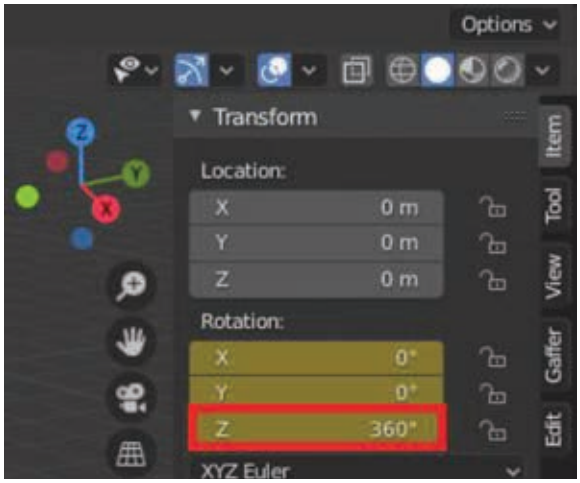


Set a rotation keyframe by pressing i with the mouse cursor above the Rotation values of the Empty in the Option panel at frame 1, with values of 0. Then set another keyframe at a later frame (frame 50 for example) with the Z-rotation value of 360°.

Frame 1:

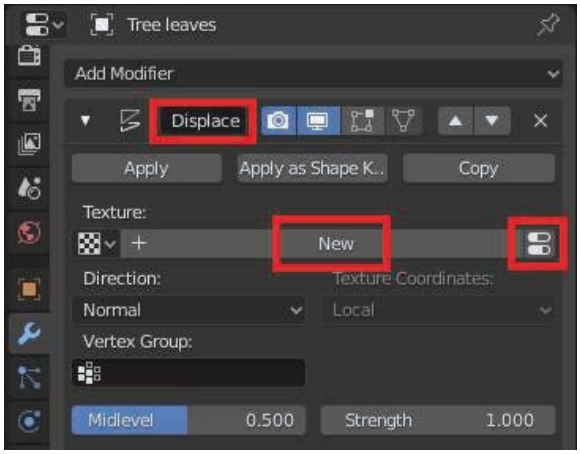


Frame 50:

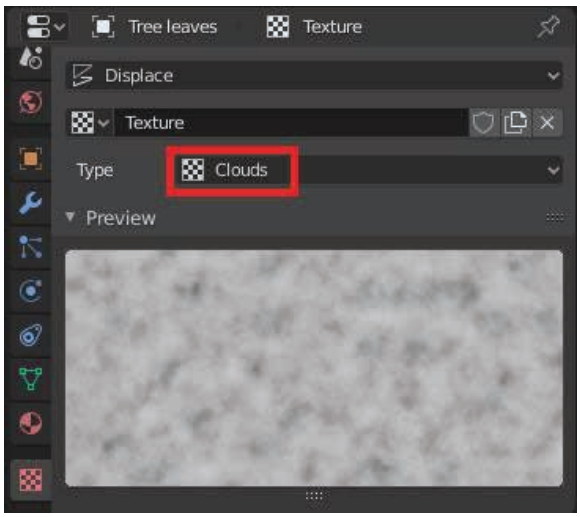


Select the leaves mesh and add a **Displacement modifier.**

Click on **New** to create a new texture, then click on the texture icon next to the New button. This takes you to the texture tab.



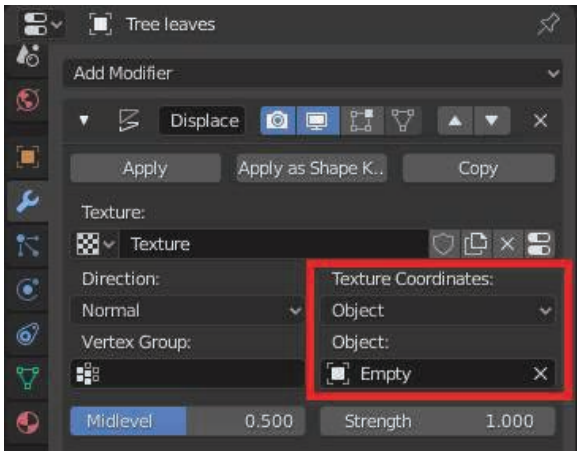
Choose a **Clouds** texture for the Displacement modifier.



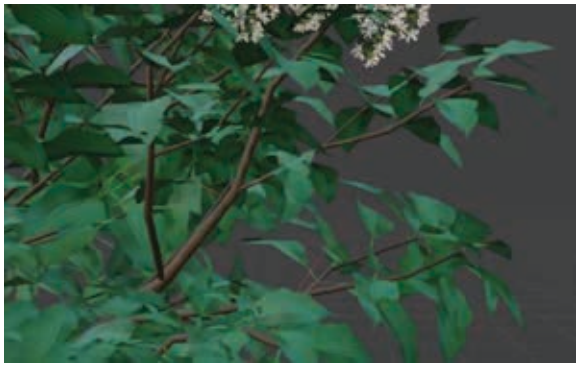
In the Displacement modifier, set Texture Coordinates to **Object**.

Use the **Empty** as the object.

This means that the position in 3D space of the Empty will drive or Displacement modifiers texture coordinates, so whenever the Empty moves, the texture moves.

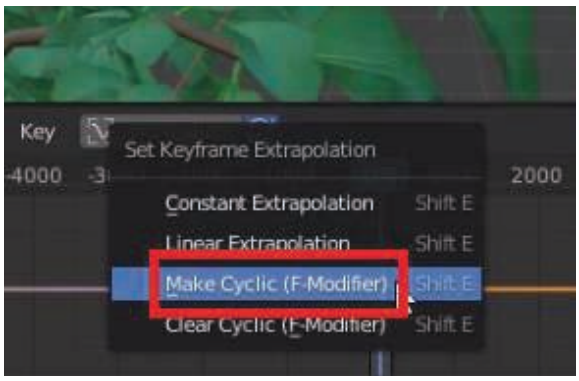


That's all! Easy leaf movement.



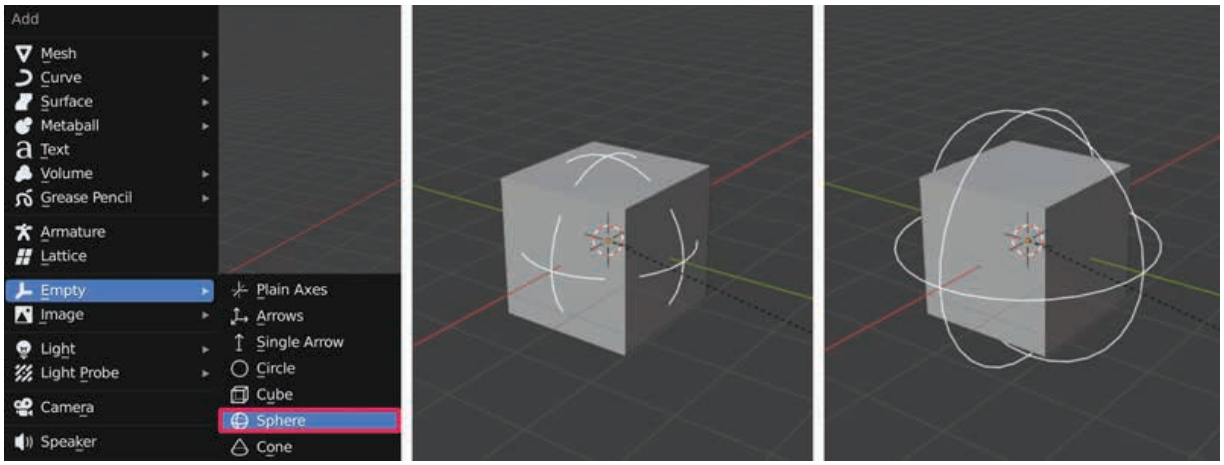
You can control the speed by moving the last keyframe. Moving it further in time makes the animation slower, moving it closer to the beginning makes it faster.

Bonus tip: to make the animation cyclical (i other works, loop infinitely), in the GraphEditorselect the keys and **press Shift+E**, choose **Make Cyclic** (F-Modifier).

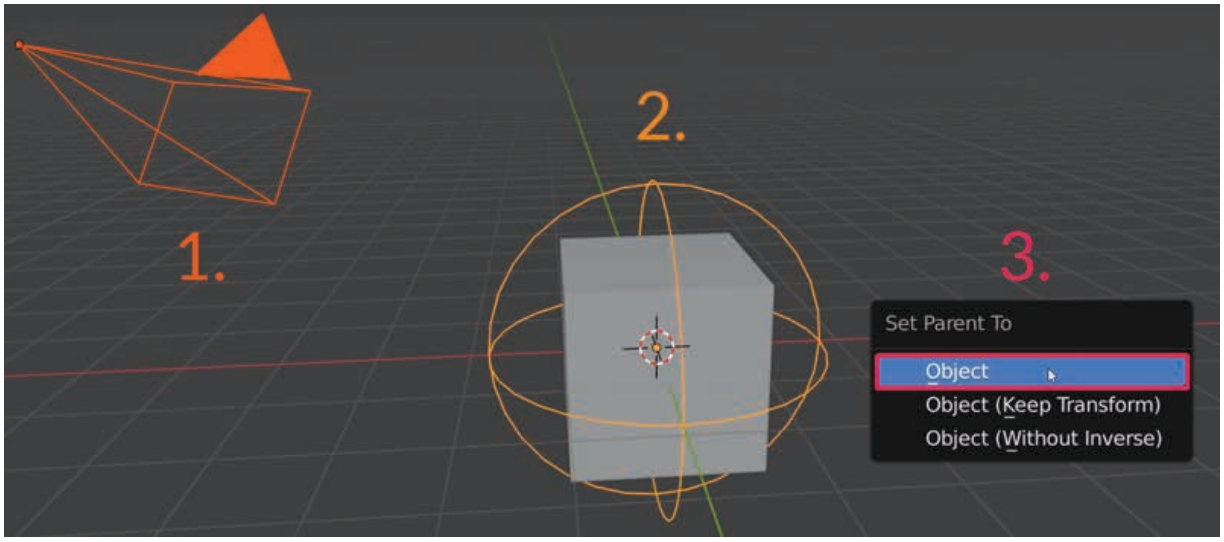


Camera rotation without keyframes

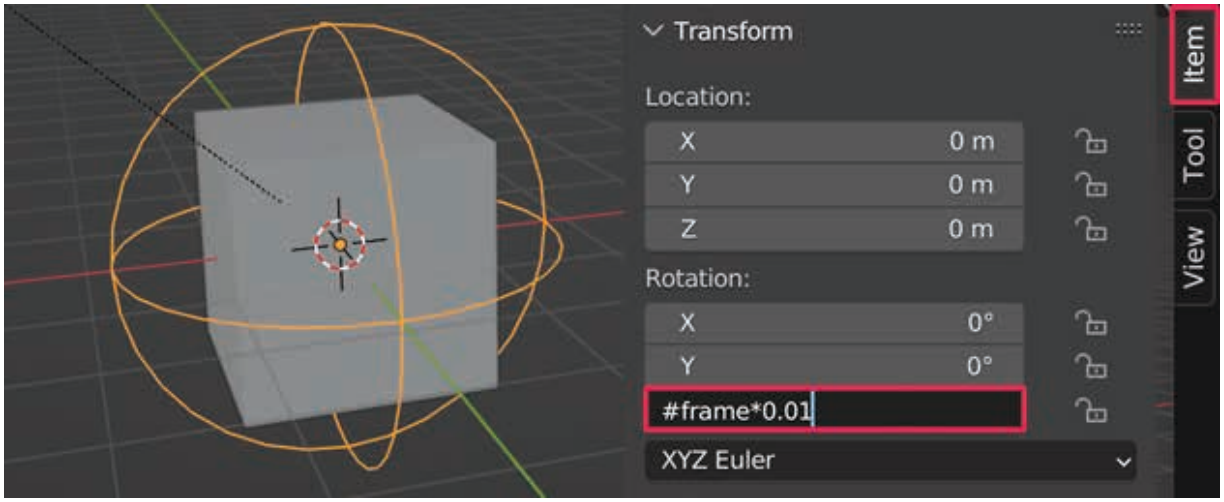
To create smooth rotating camera motion without keyframes, first add an Empty. In this case I choose a Sphere Empty but the shape doesn't matter. You may need to scale it up by pressing S.



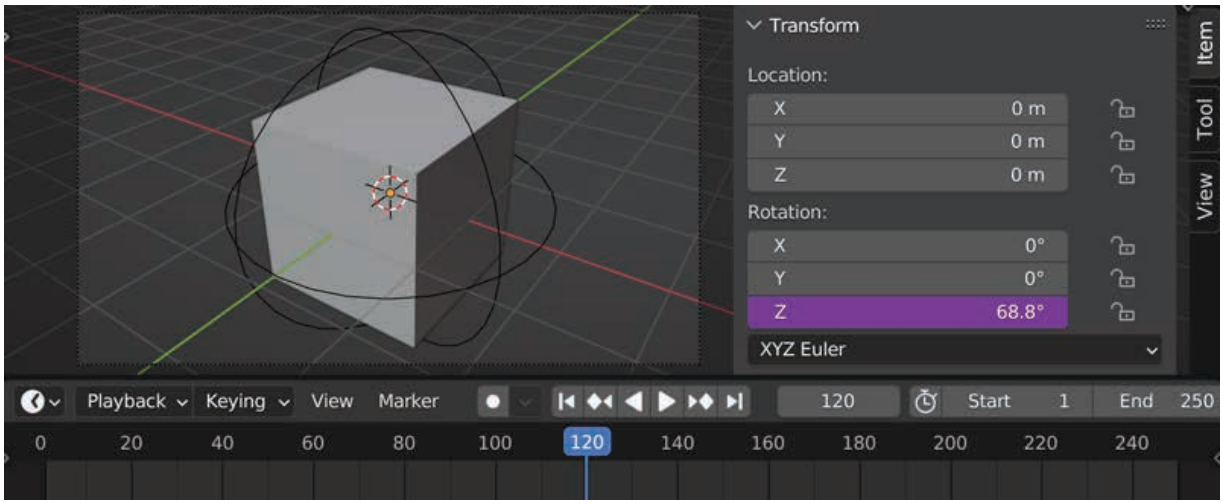
Select the camera and then the empty and **press Ctrl+P** to parent them.



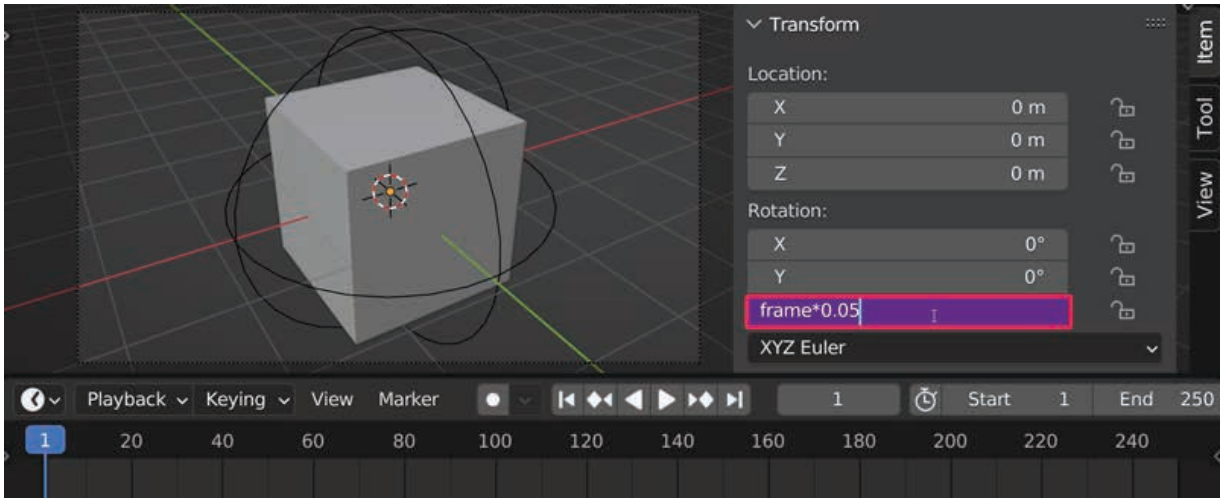
In the **Z rotation field** of the empty type **#frame*** and a number like 0.01.



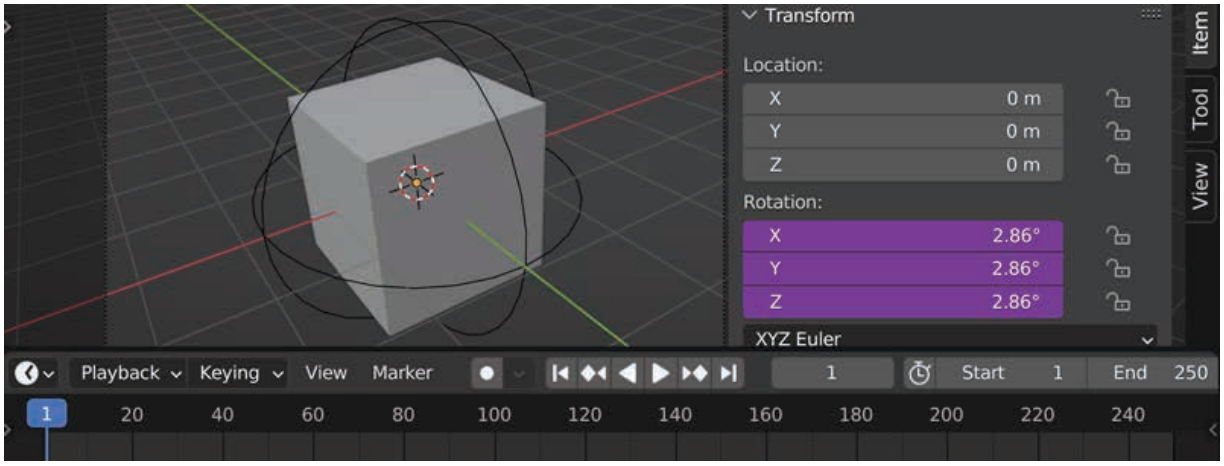
Then when you **press the Spacebar** the camera will rotate with the empty.



Changing the number changes the speed of the rotation.

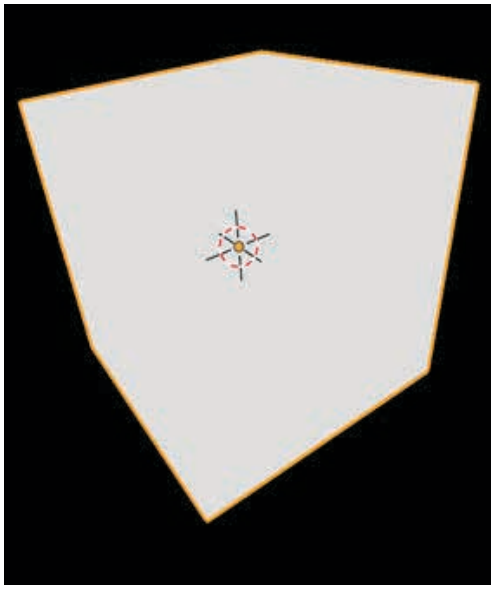


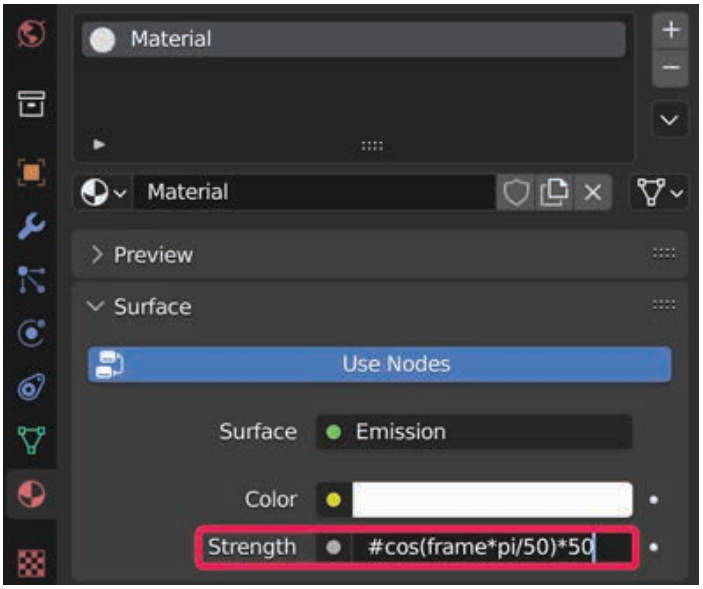
You can also input **#frame*** in the other rotation fields for some cool rotations.



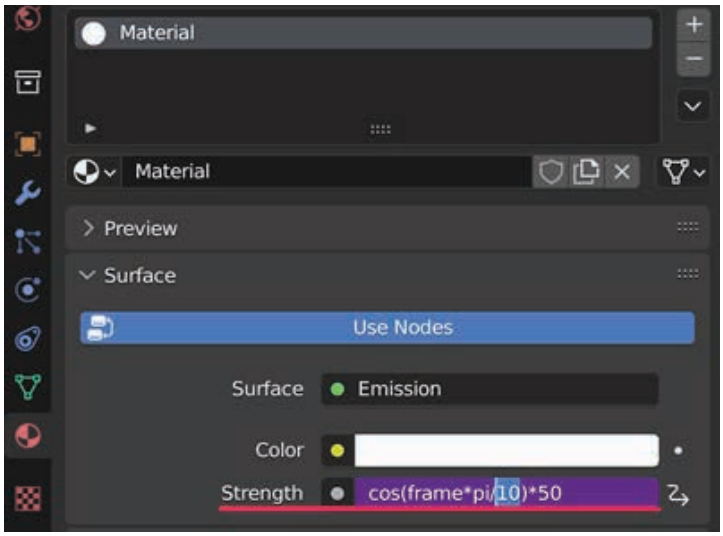
Pulsating light without keyframes

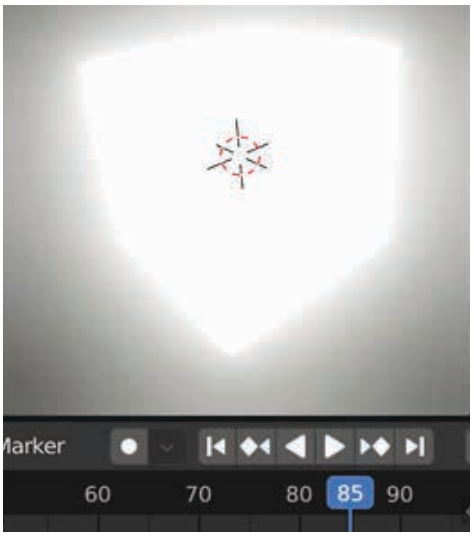
To make a pulsating light, first create an emission shader for your material. Then in the Strength field type this expression: **#cos(frame*pi/50)*50**



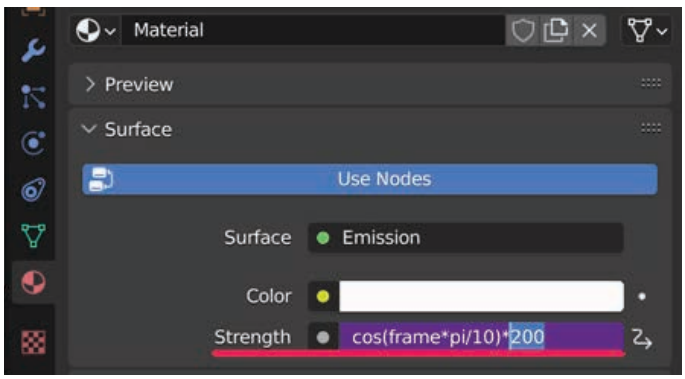


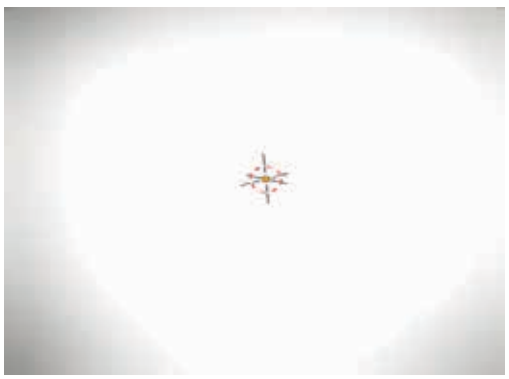
Tthe first number controls the speed of the pulsating light. The smaller this number, the faster it will pulsate.





The second number controls the strength of the emission. The bigger the number, the stronger the emission.



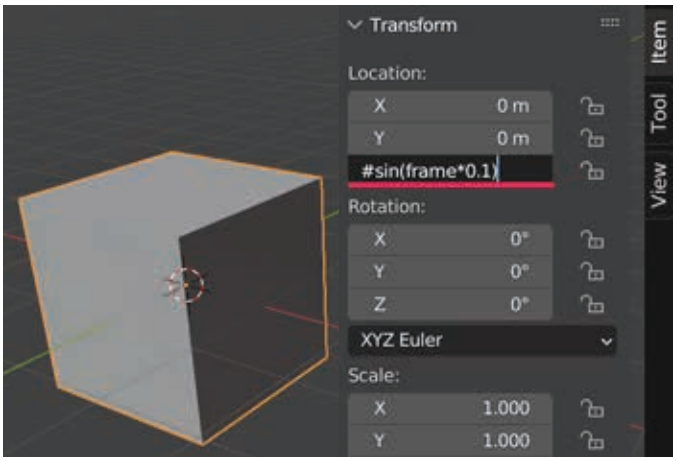


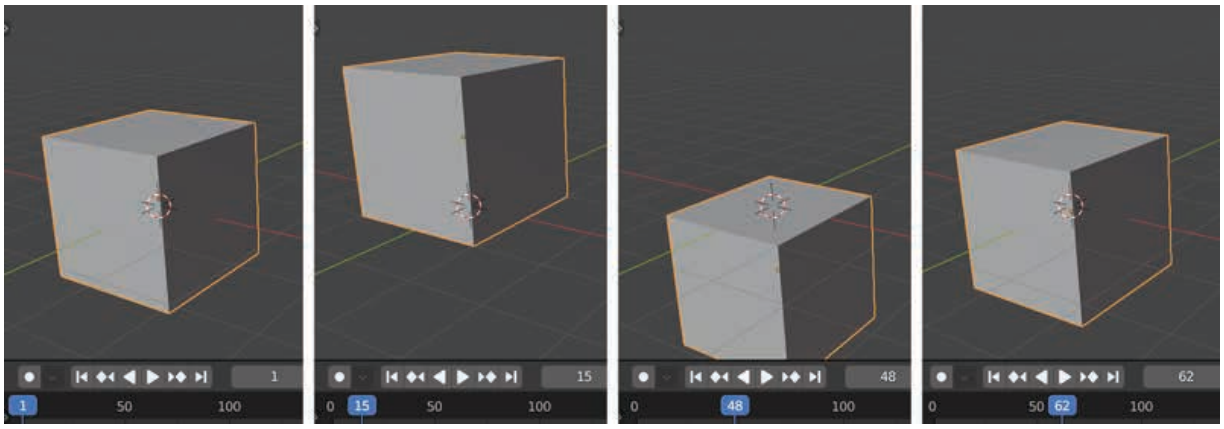
Click here to return to the index.

Wiggle Expression

To get an object to move back and forth on an axis type this expression in the Location field of that axis.

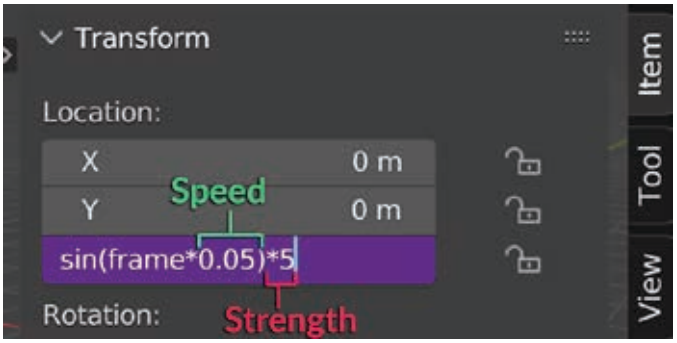
The number (0.1 in this case) presents the speed.

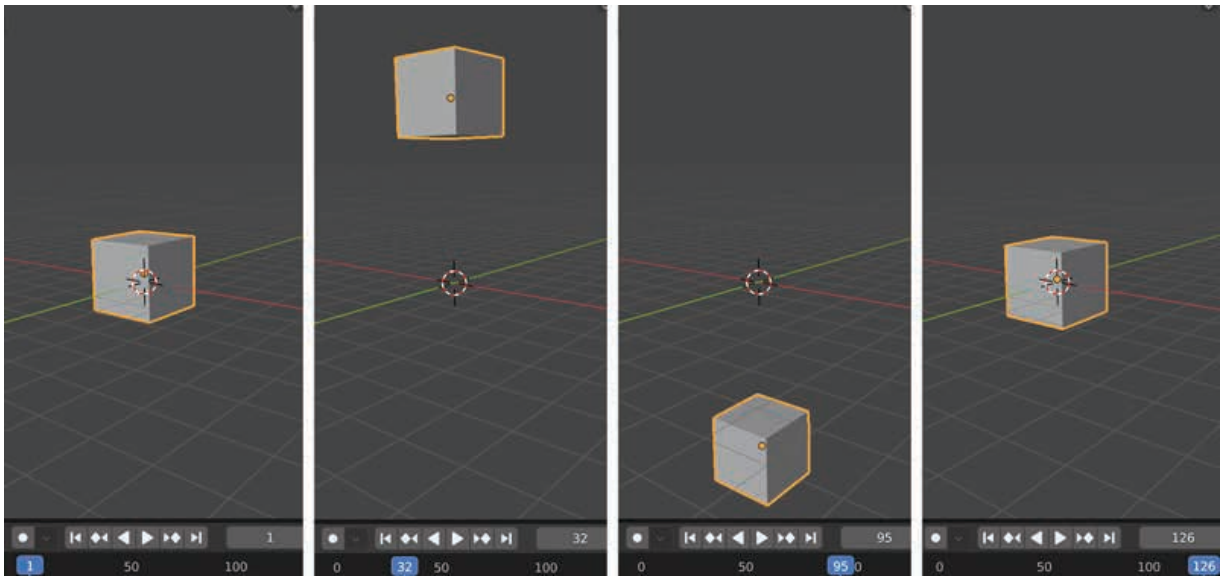




A smaller number means a smaller wiggle speed.

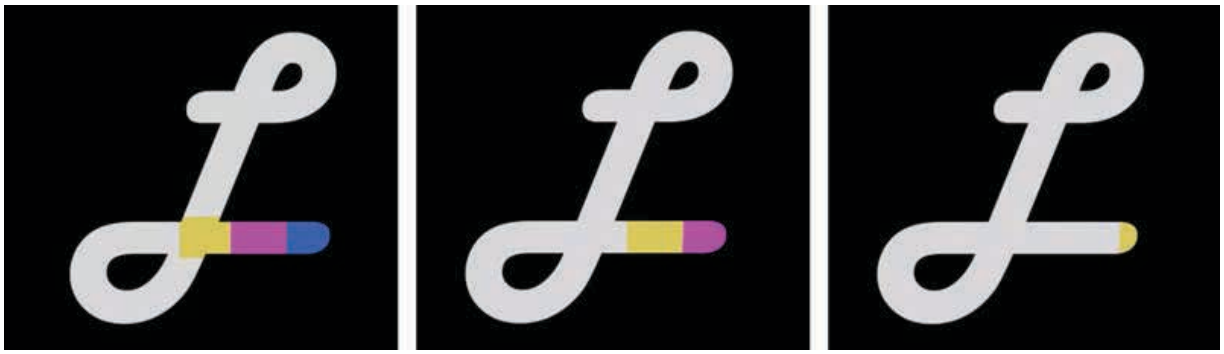
You can multiply all this to make the strength of the wiggle stronger.



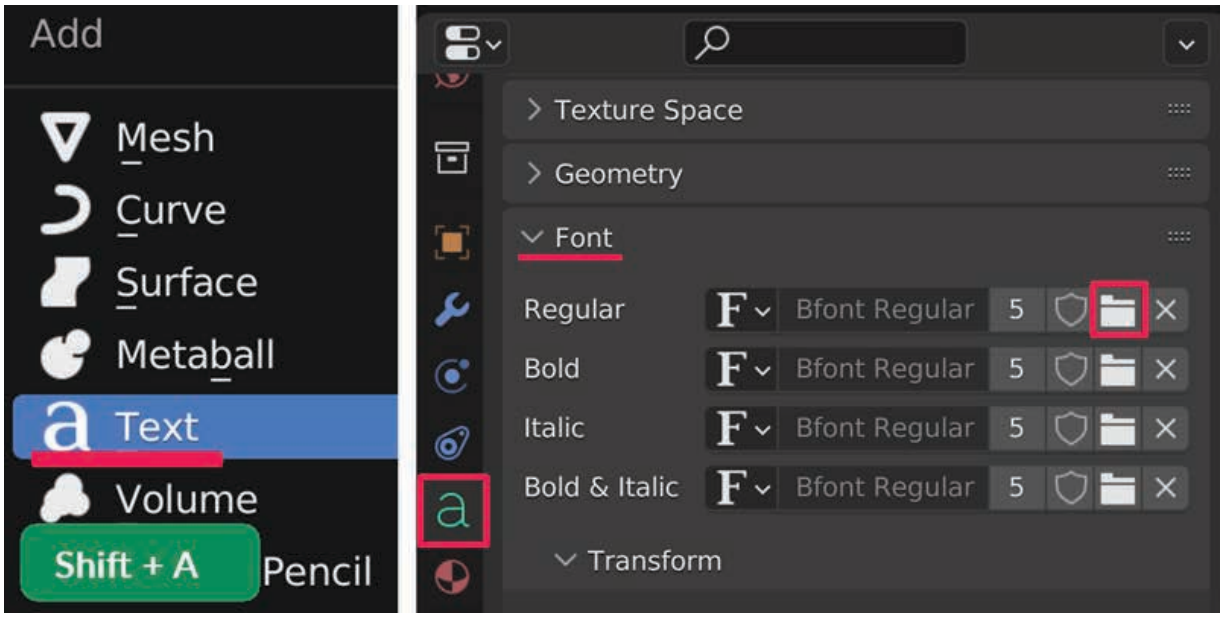


Click here to return to the index.

Liquid Colorful Text Reveal Motion Graphics

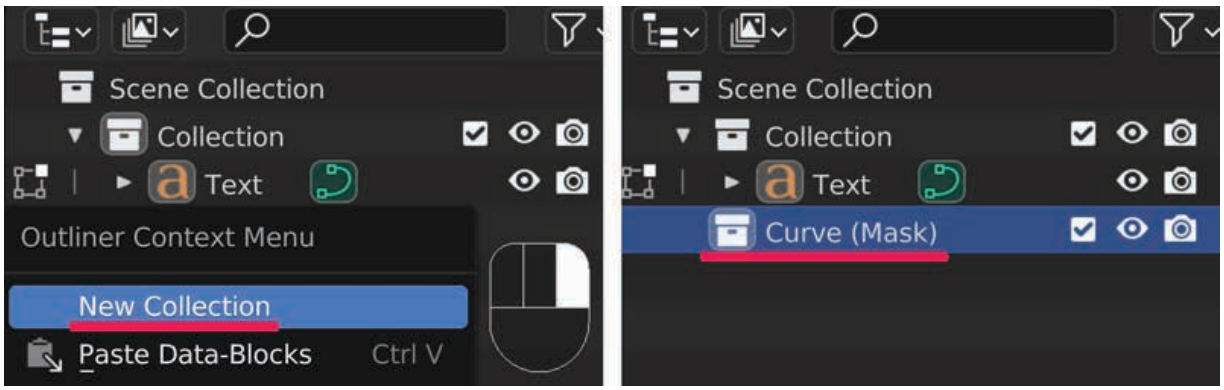


This is an effect that's usually created in programs like After Effects. Can we do it in Blender? Sure! To start, we can animate a curve and use that as a holdout mask. Create a Text object by **pressing Shift+A** and choosing **Text**.



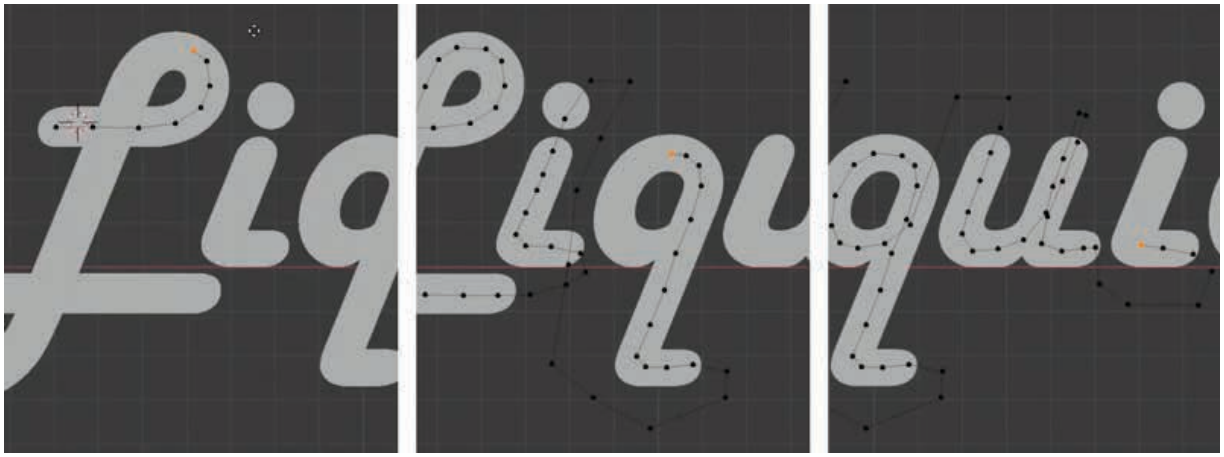
Select a font you like. If you want you can finetune the kerning with Alt and the arrow keys.

Create a new collection and select it so that anything new will be made in that collection.

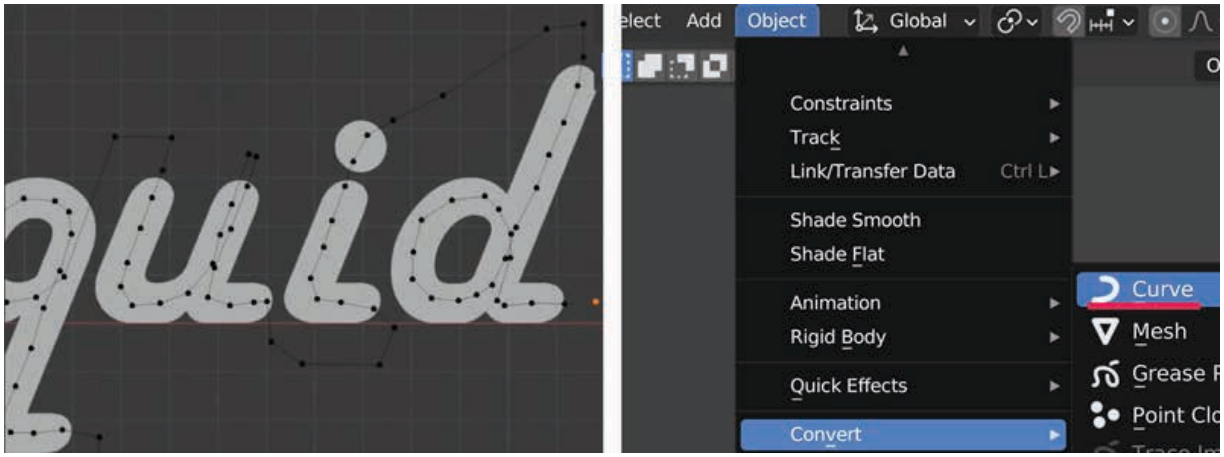


Create and extrude a Curve to cover the entire area of the text.

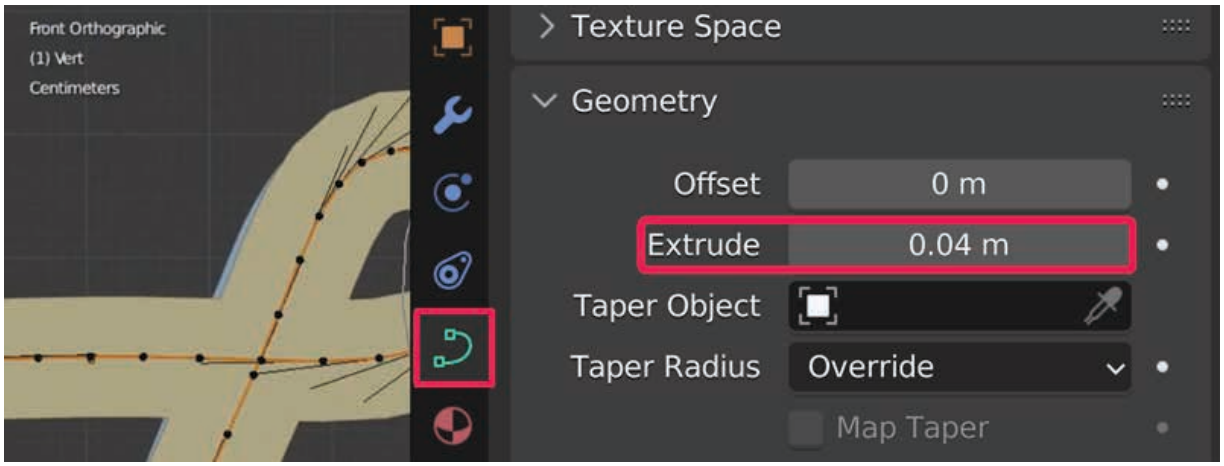
I like to extrude a vertex and convert that to Curve, I think it's easier.



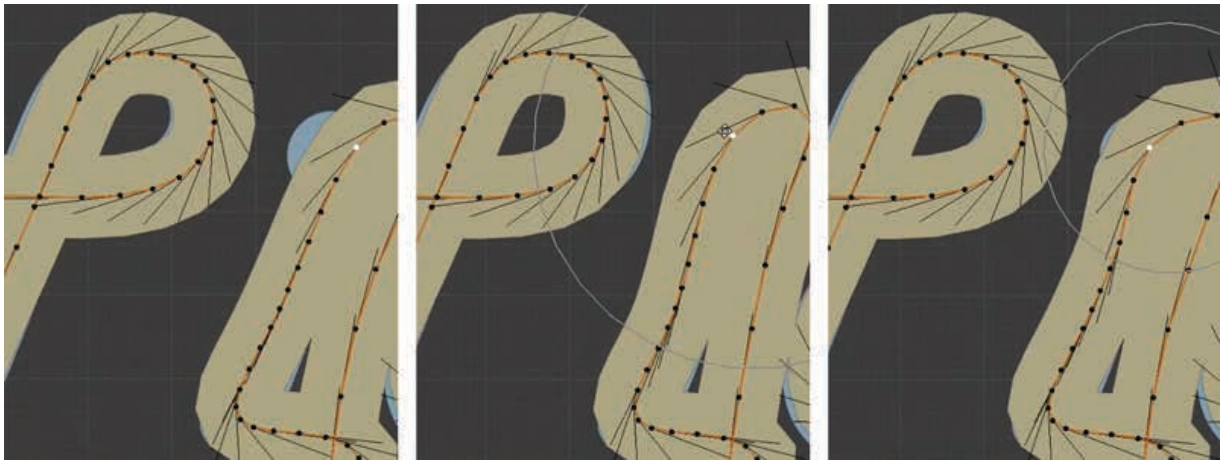
To convert it, go to **Object > Convert > Curve**.



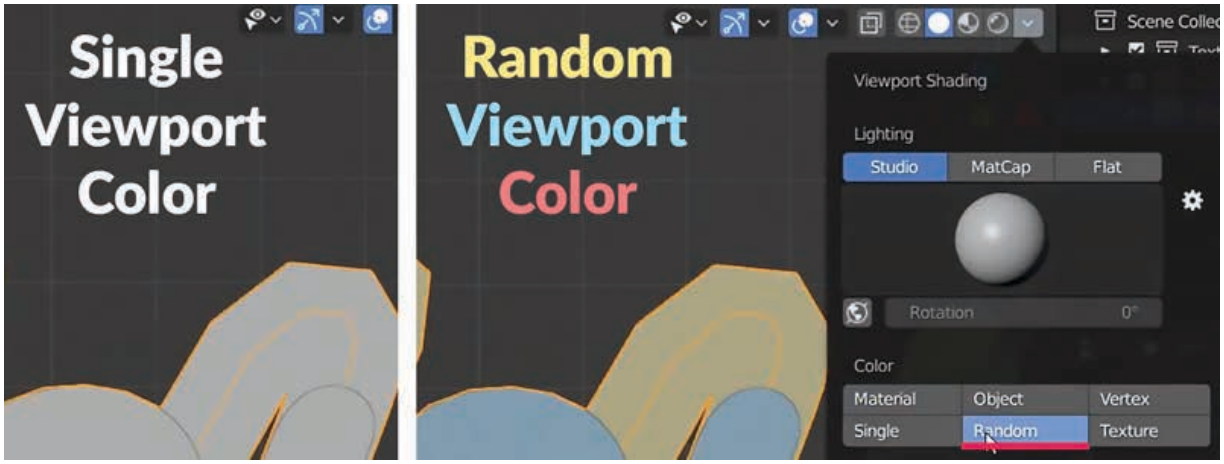
Extrude the Curve so it has some width to cover the text.



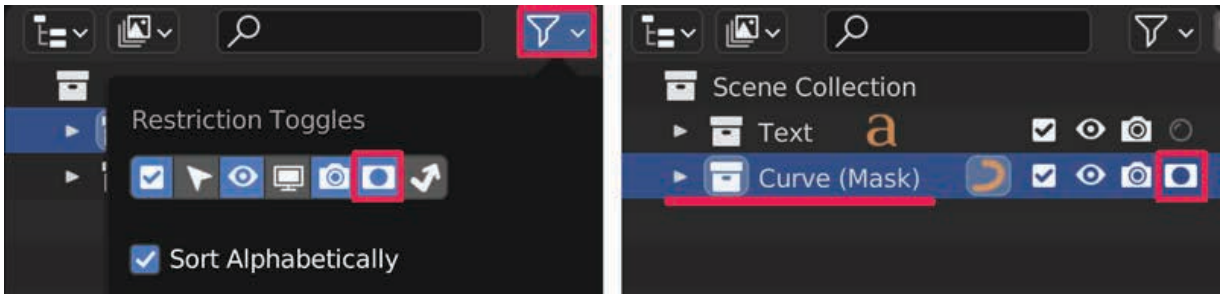
You'll probably need to finetune the curve a bit so it really covers the text.



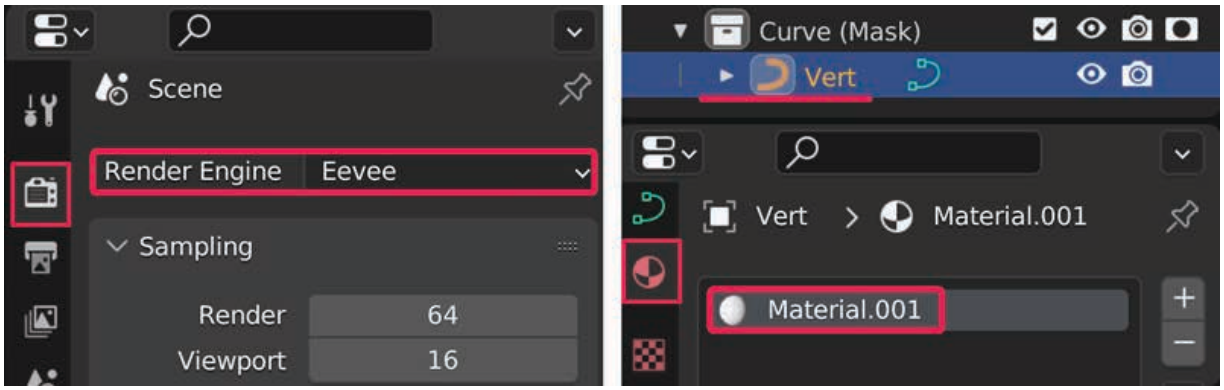
Turn on Random viewport colors so you can see more easily what you're doing.



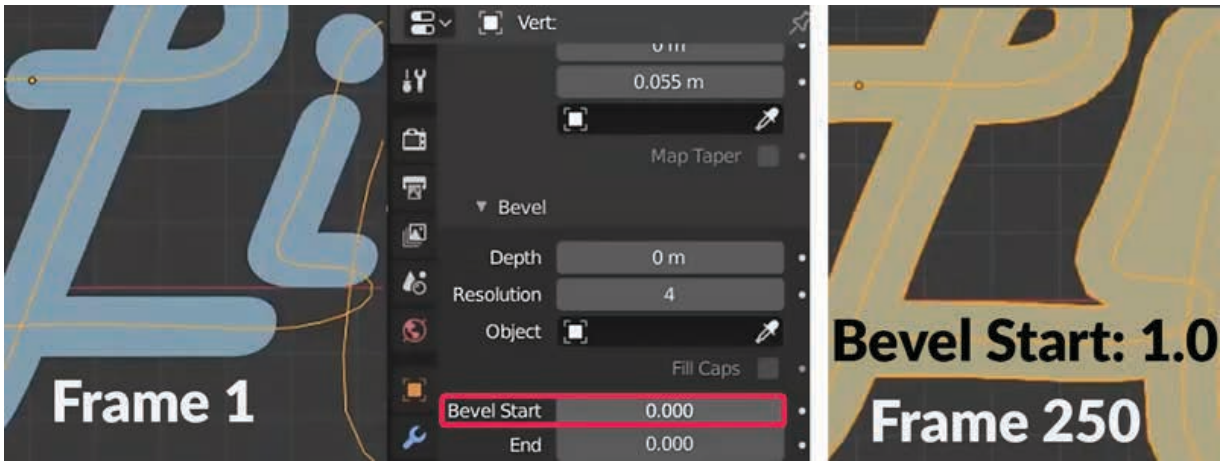
Enable the **Holdout** filter in the **Outliner** so the Curve becomes a mask for the text.



In Eevee, the Curve needs to have a material for the holdout to work.



Now you can set keyframes for the Bevel value to animate the curve which reveals the text.



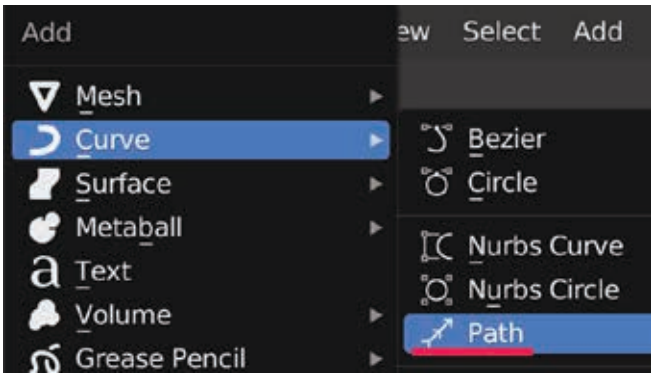
Render the animation with a few different colors and composite them back together with a few frames offset so you create this nice liquid color effect.

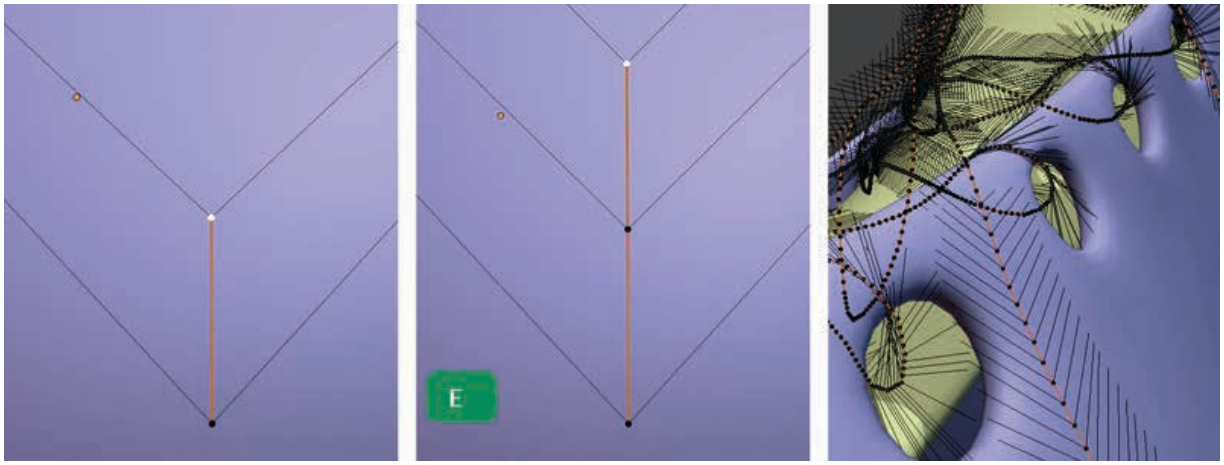
Shoelace animation

Motion Graphics is not something that Blender is not as known for, but it's definitely capable of some cool mograph effects! This shoelace animation was done by moving an array with a start and end cap along a curve. Download the Blend file here.

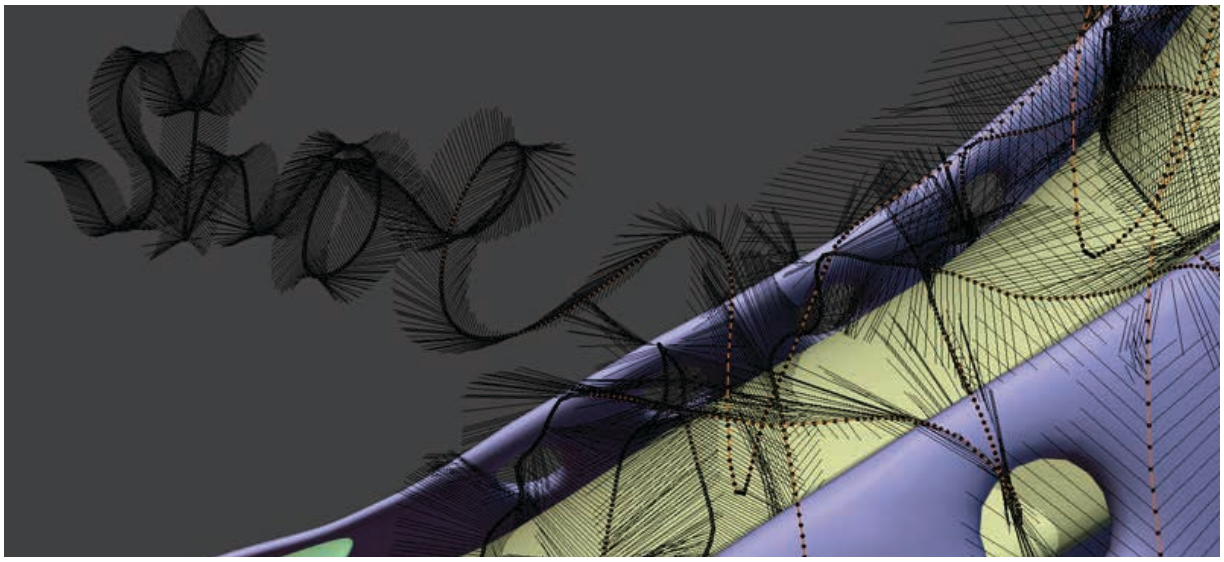
Model a path for the shoelace to follow by extruding and moving vertices of a Curve.

You can extrude vertices of a Curve by pressing E, just like with a Mesh. Move them by pressing G.

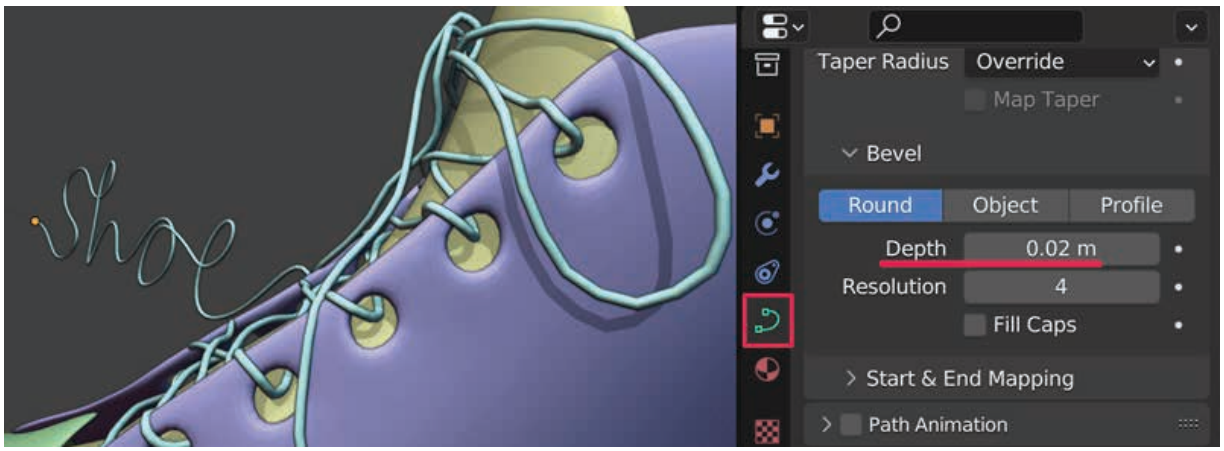




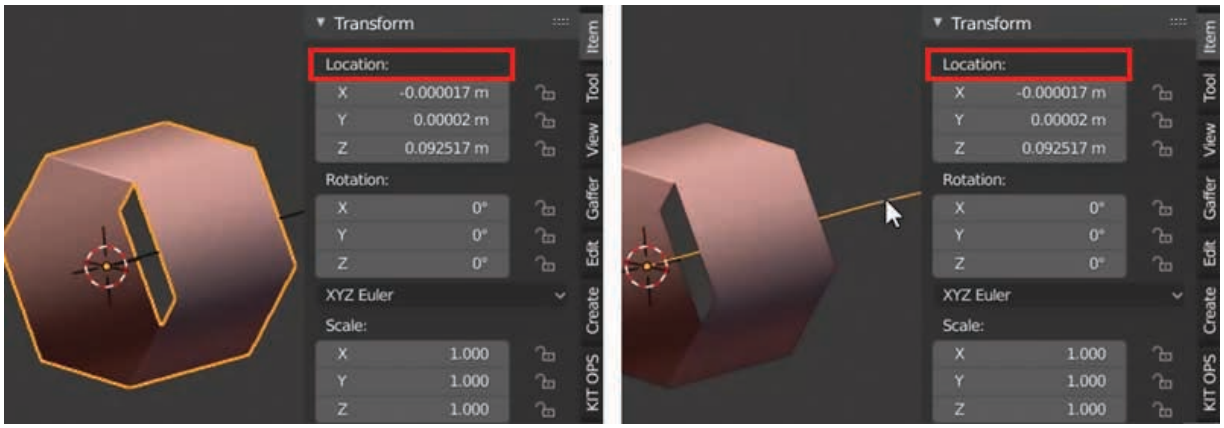
Click here to return to the index.



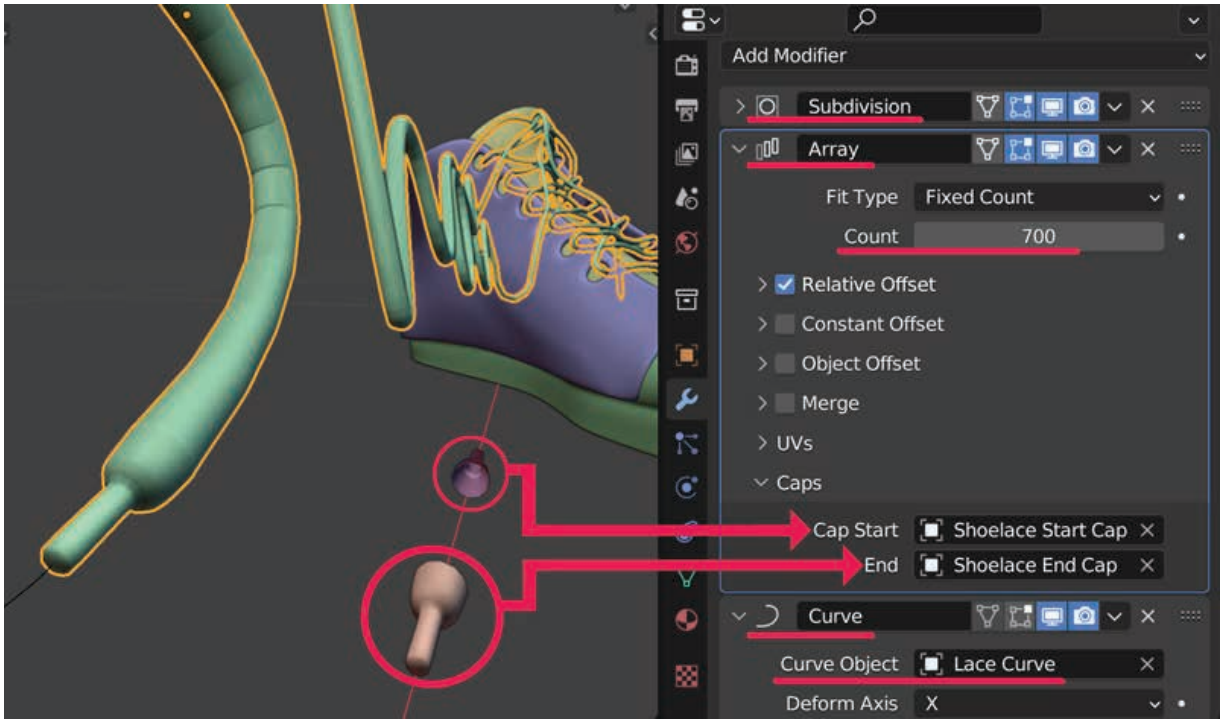
Give this Curve some temporary thickness by increasing the **Depth** value, as reference to fine-tune the curve.



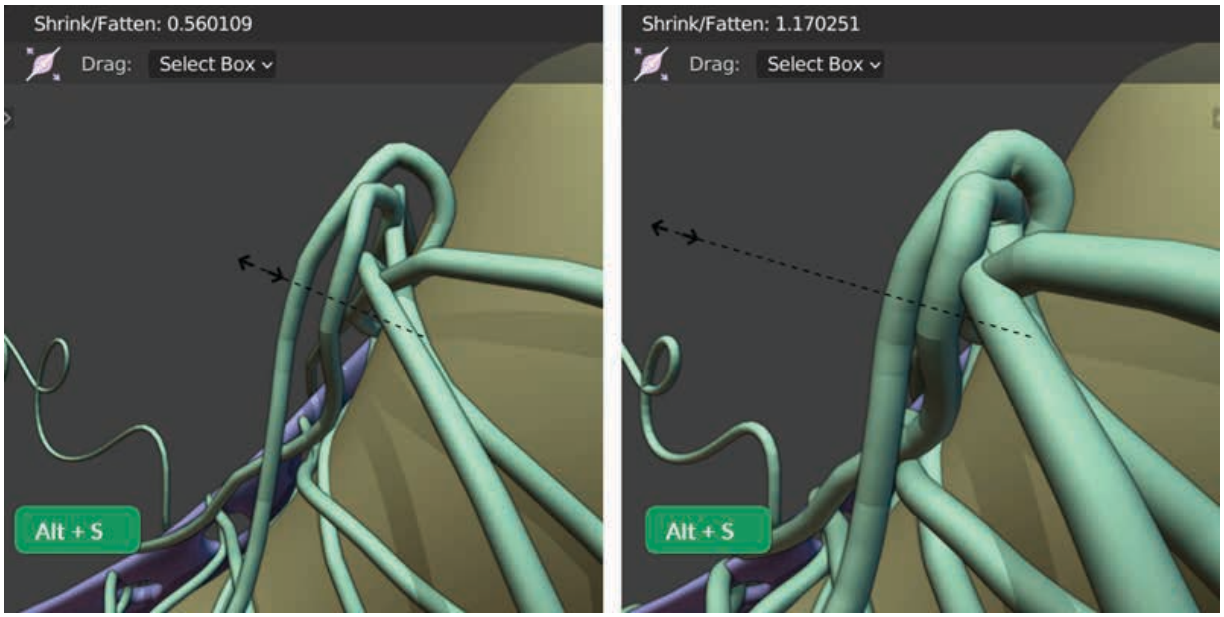
Once you're happy with the Curve, set the **Depth value back to 0**. Model a small segment of the shoelace. Place it at the start of the curve. It's important that the origin points of the Curve and the shoelace object are at the same location. Rotation and scale should also be applied on both objects.



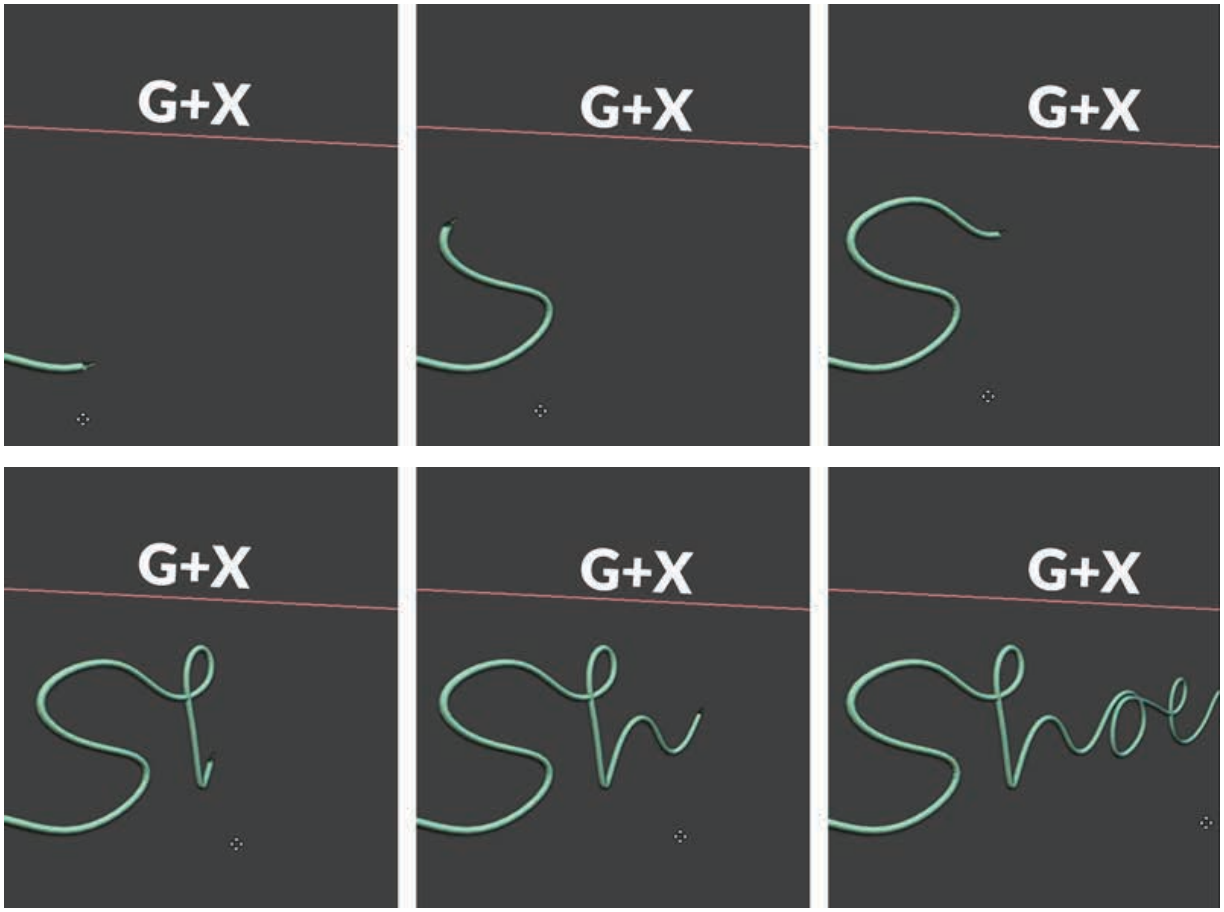
Add an **Array, Subdivision and Curve modifier** to the shoelace object. As the **Object** in the Curve modifier choose the extruded and modeled Curve. Model the start and end parts of the shoelace. Set those as the **Start and End caps** in the Array modifier. You'll need to increase the Array **Count** to create a long enough shoelace.



You can change the thickness of the shoelace by selecting the vertices of the Curve and **pressing Alt+S**.

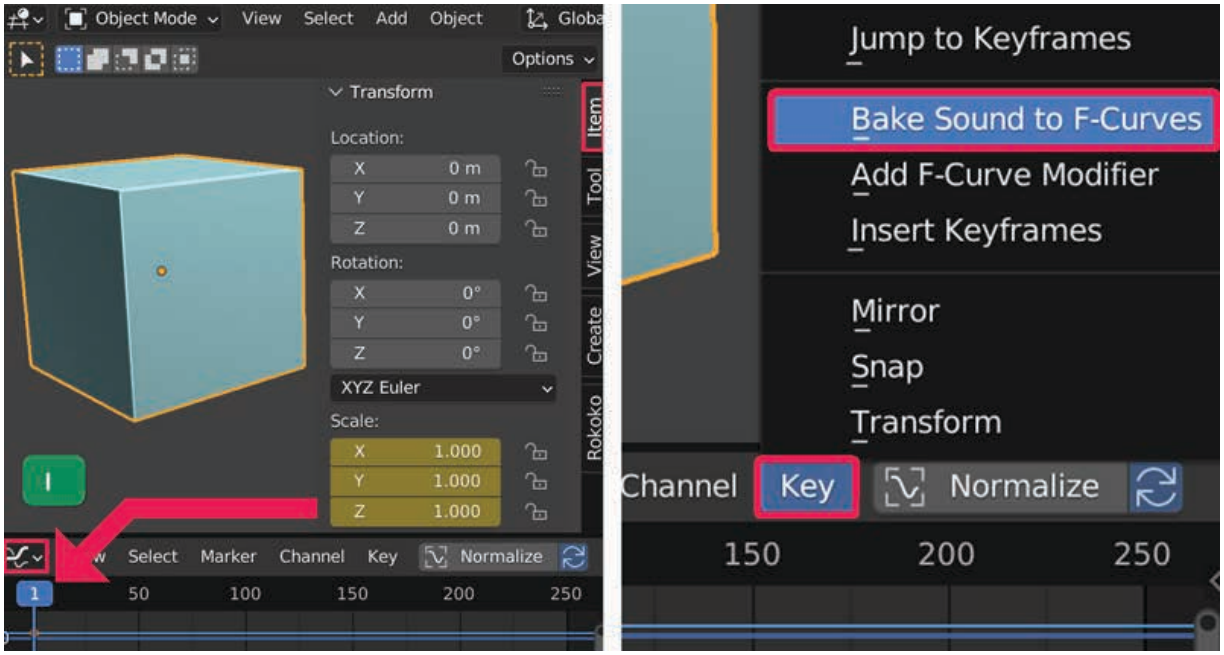


Set keyframes for the position of the shoelace segment to animate it.

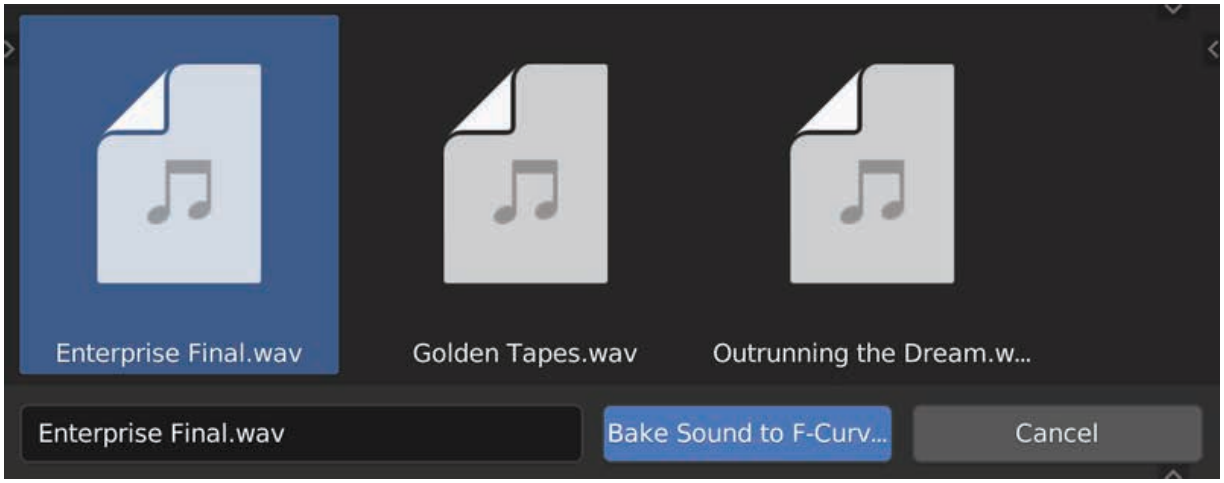


Using Sound to animate objects

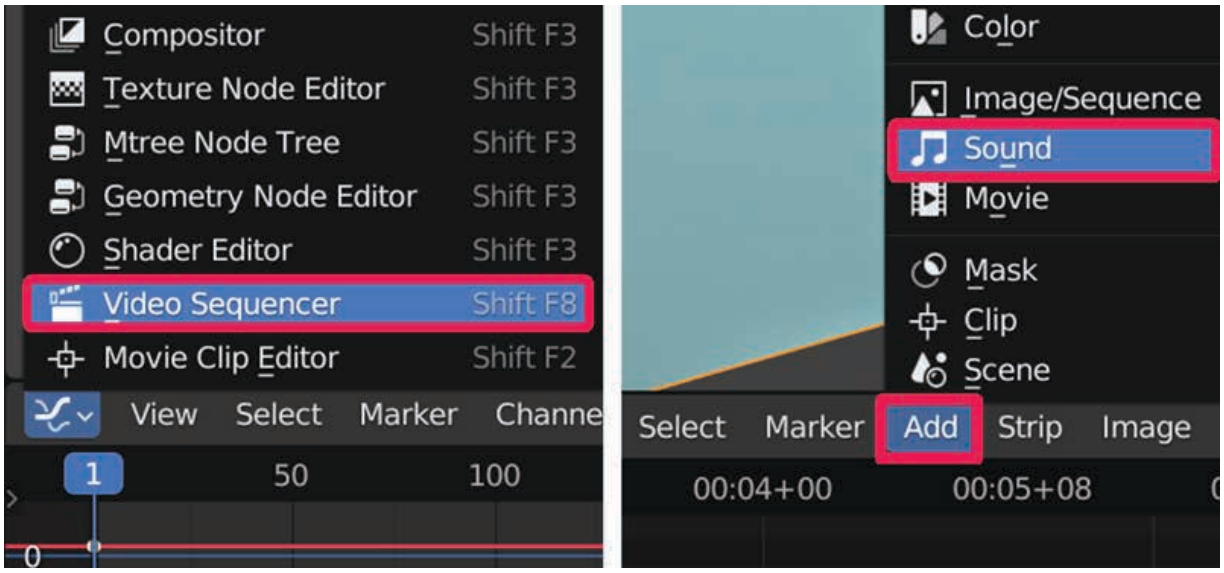
Did you know that you can drive animation with sound files in Blender? And it's pretty easy, too. You can use sound to animate modifier values, drivers and more for your motion graphics. This Default Cube has a keyframe set for the **Scale** by **pressing i** while hovering the cursor over object Scale value field. Open the **Graph editor window** and make sure the object is selected then go to Key > **Bake Sound to F-Curves**.



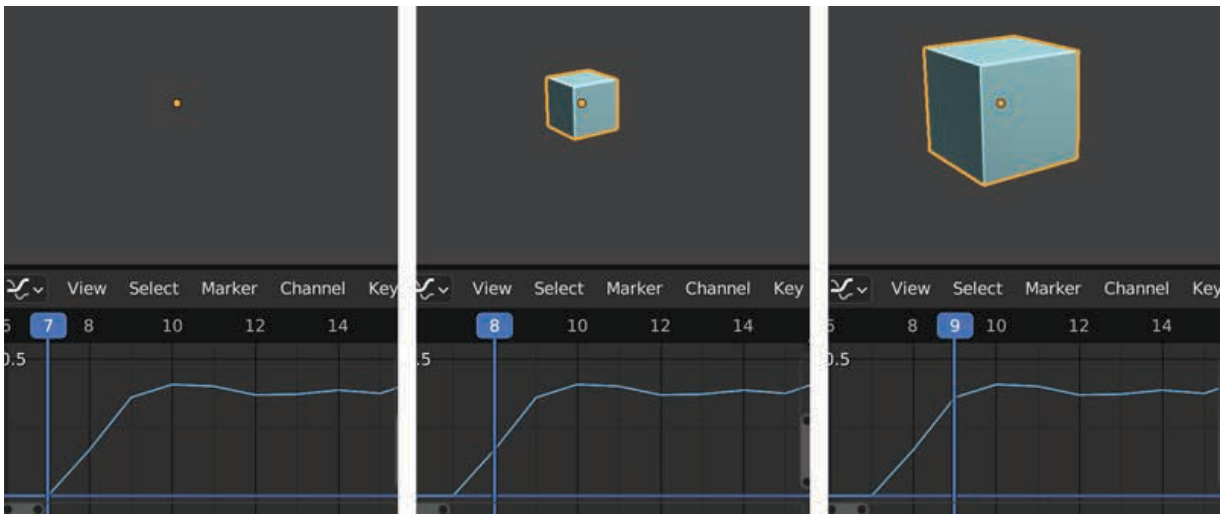
Select the music or sound that you want to use.



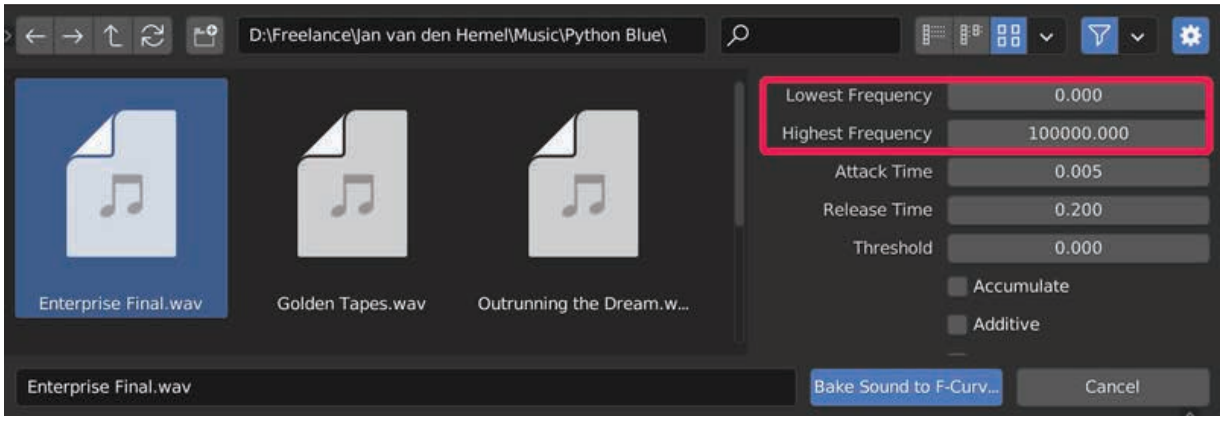
Now go to the Video Sequencer and go to **Add > Sound** and add the same music or sound again.



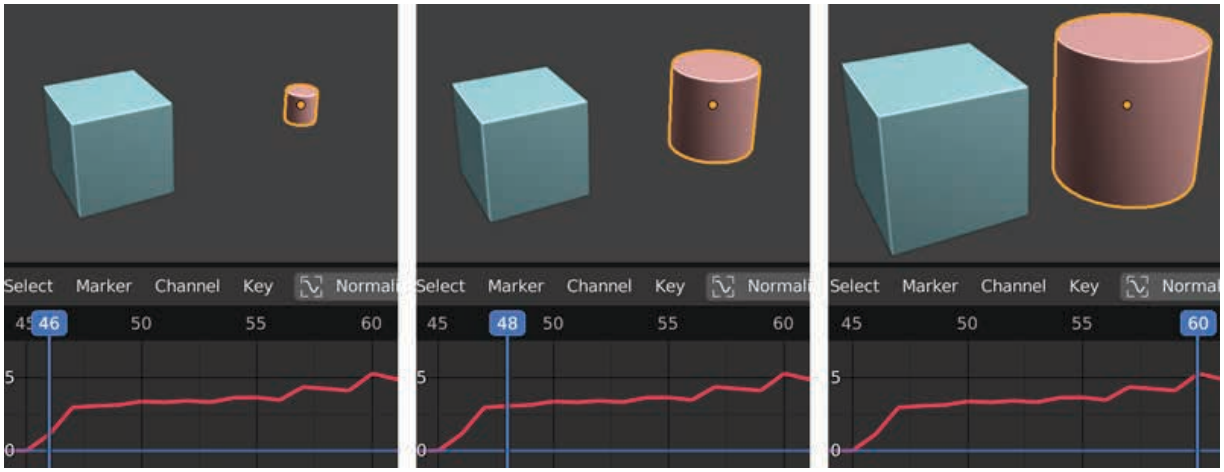
Now if you **press the Spacebar** the music will play and influence the keyframe.



You can use these Lowest and Highest frequency settings when baking the sound to F-curves to choose which frequency range of the music that you want to visualise.



That way different objects can visualise different frequencies of the music.

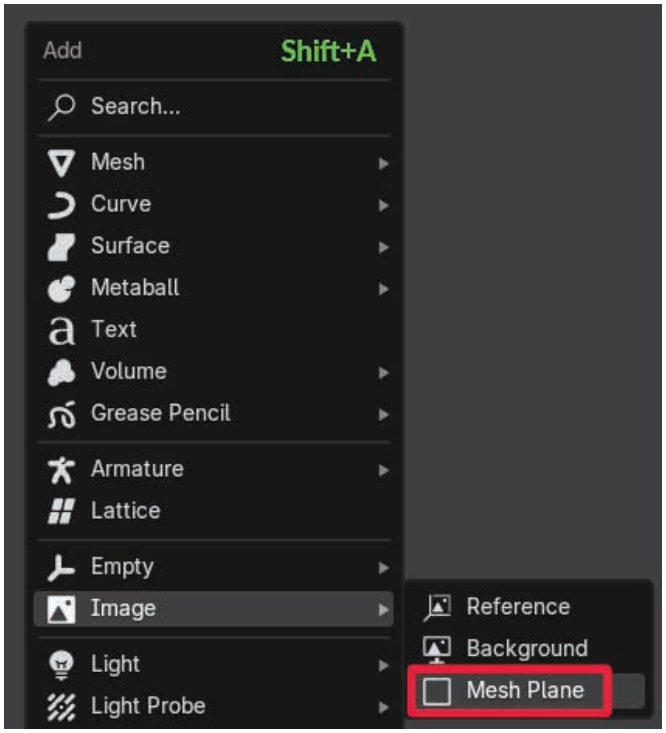


How to control the speed of image sequences

Image sequences on planes can be great for adding visual effects elements like smoke, fire and explosions.

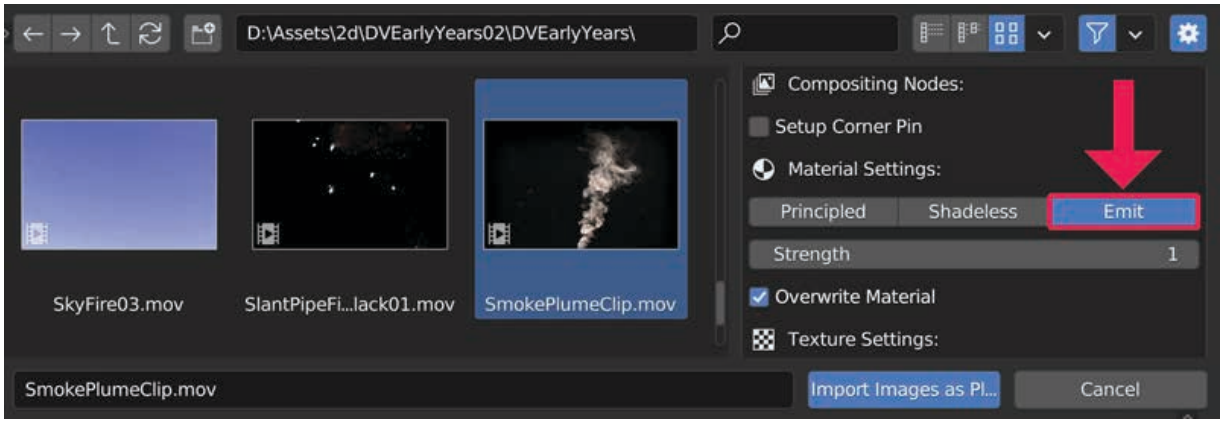
But how do you control the speed they play at?

First, import your video or image sequence as Plane by **pressing Shift+A** and choosing Image >**Mesh Plane.**

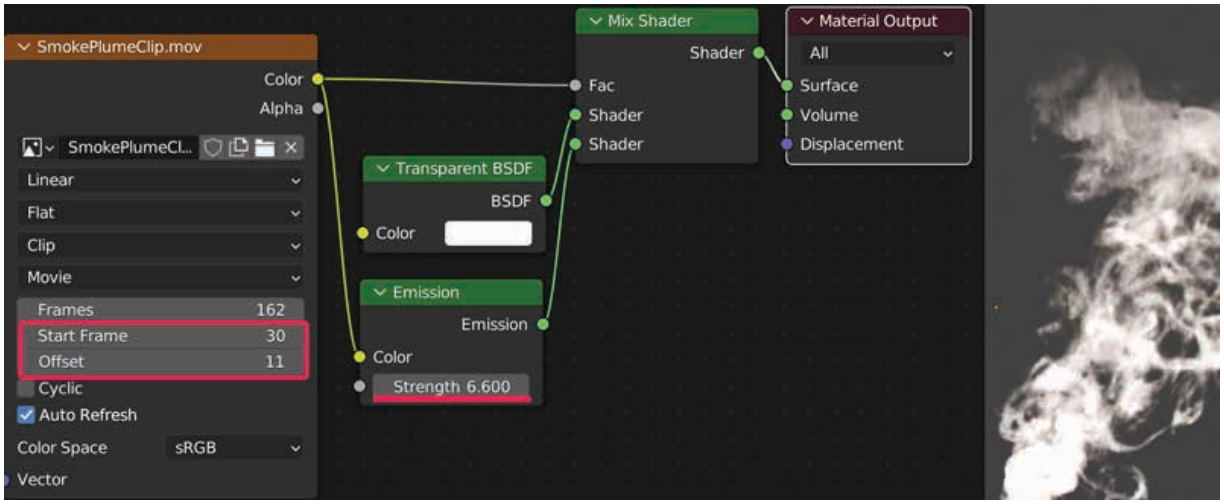


Click here to return to the index.

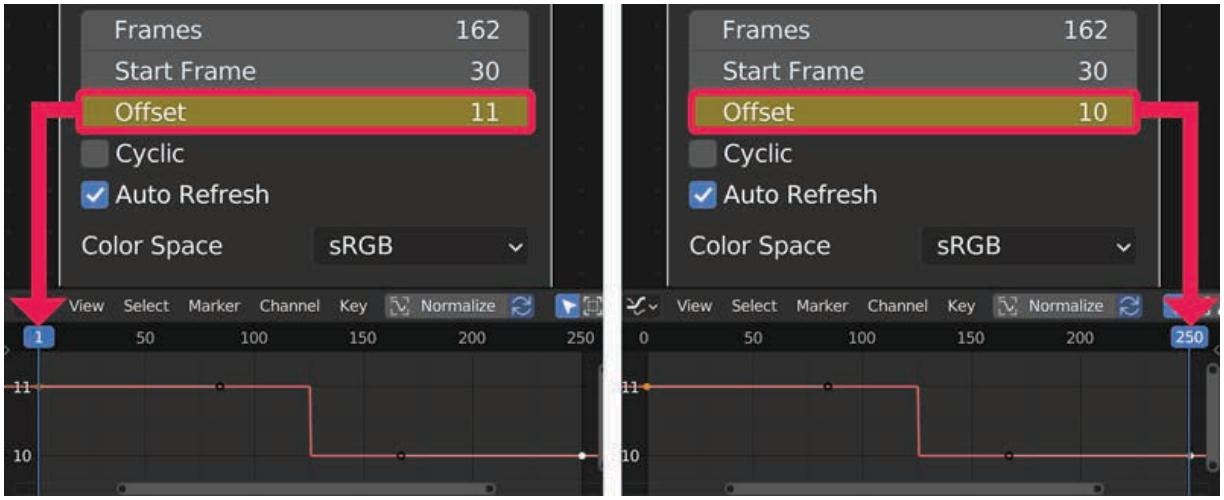
Set it to Emit.



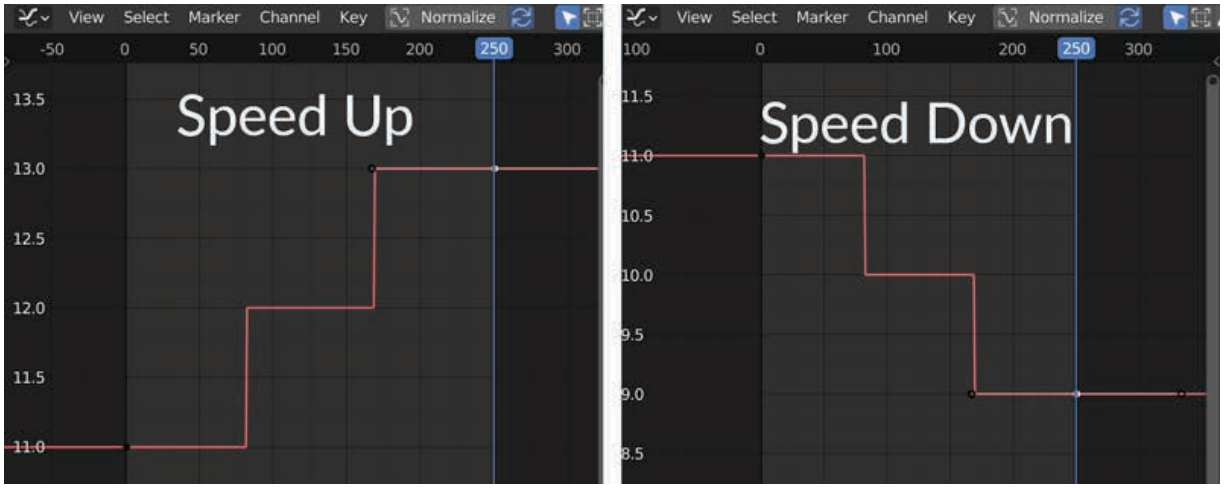
In this case my video doesn't have an Alpha Channel, so I use the **Color** output as the **Mix Factor**. I can increase the **Strength** to compensate. The part I want starts from Frame 11 so 1 set **Offset** to 11. Let's say I want to start this explosion at frame 30. Set the **Start Frame** to 30.



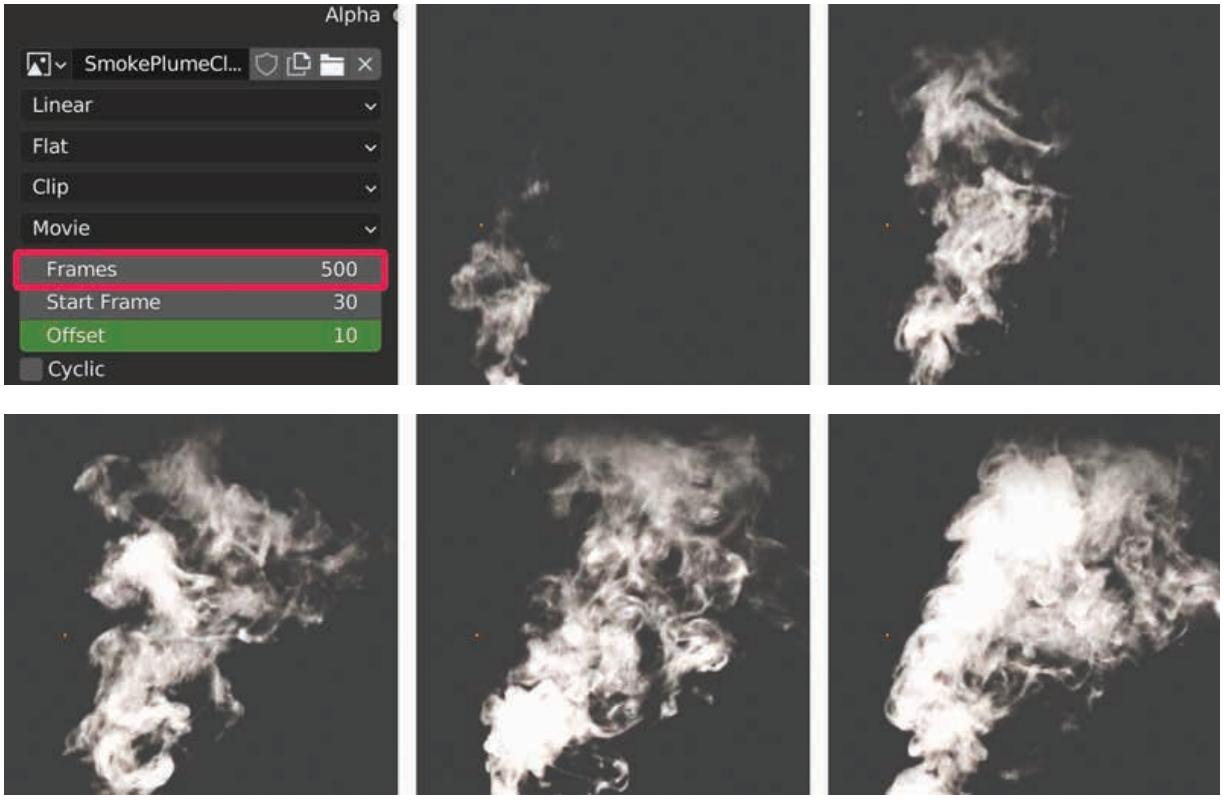
At this frame 30, insert a key for the offset value. Then at the last frame, insert a key for the lower offset value, like 10 in this case. Now open the **Graph Editor**.



Dragging the last key up speeds it up. Dragging the last key down slows it down.



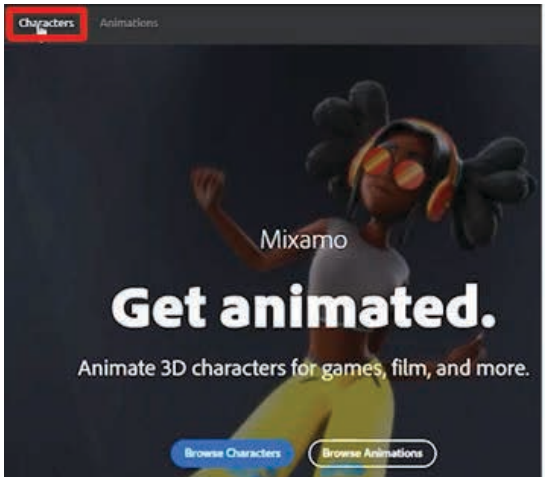
If you do slow it down, you need to extend the frames of the image sequence. Otherwise it will play backwards at some point.

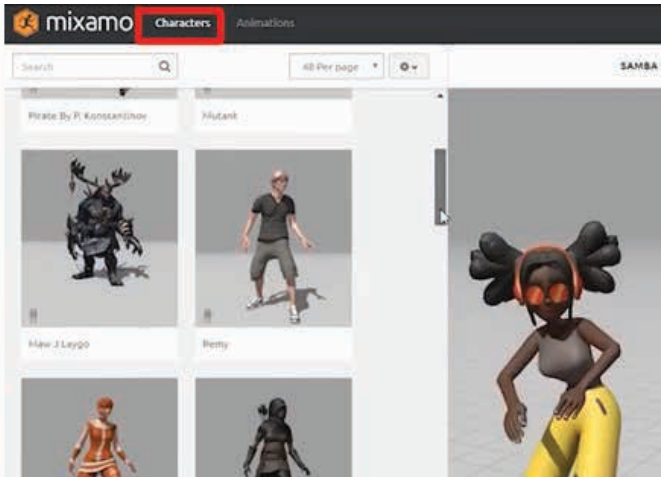


CHARACTER ANIMATION

Importing Mixamo character animation in Blender

First, log into the Mixamo website (www.mixamo.com). I'm using my Adobe cloud login, but if you don't have that you can register for a free account. **Click on "characters"** and select a character you want to use.

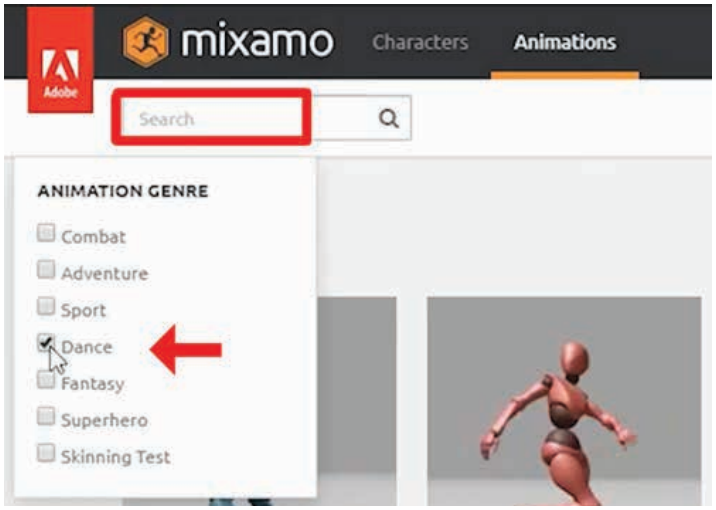




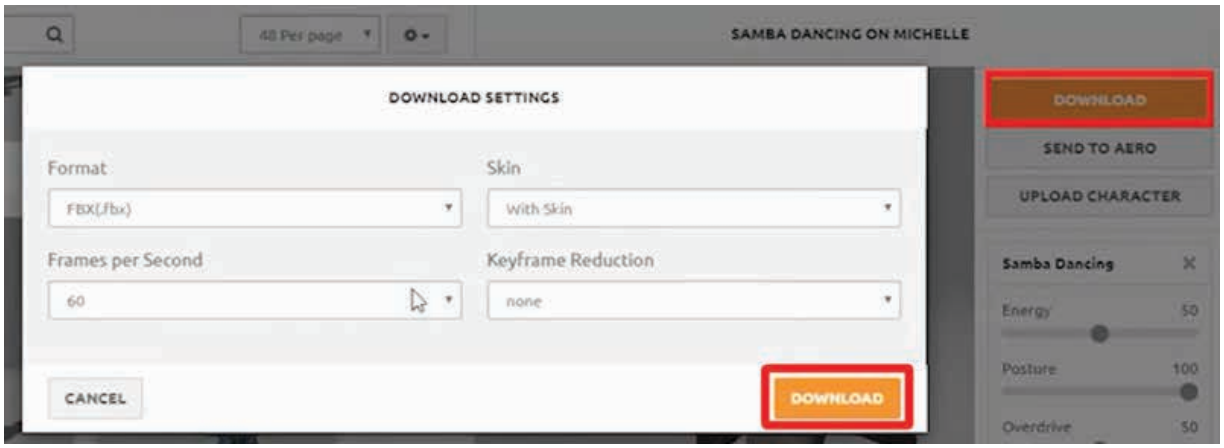
Next **click on "animations"** and select an animation.

There are a lot of animations and if you click on the **"Search"** field you can filter by **Genre**.

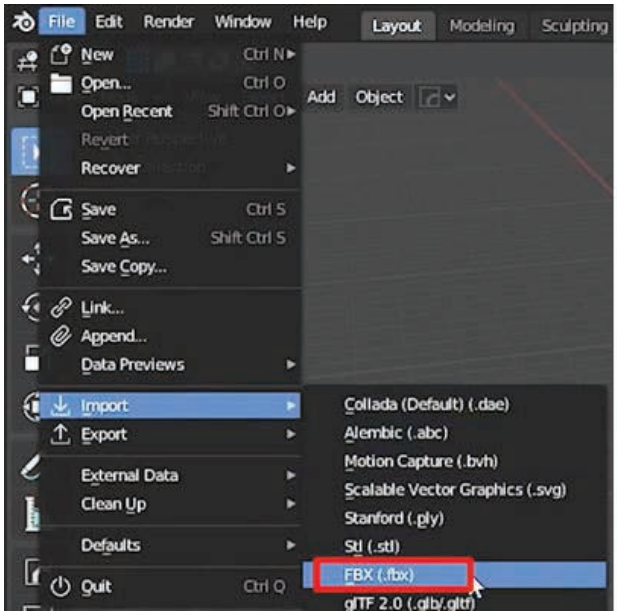
Here I choose Dance as the genre so that I can see all the dance animations.



Then **click on Download**. The default settings are fine, unless you want to change the framerate. By default the exported format is FBX, which is good.

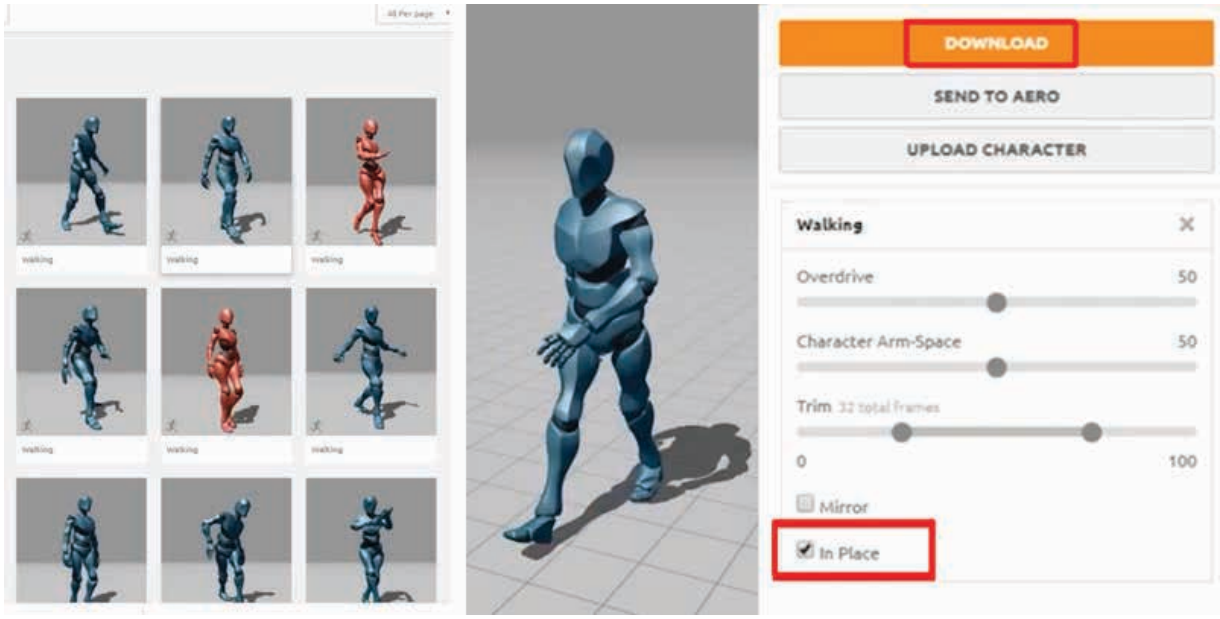


In Blender, go to **File > Import > FBX** and import the downloaded file. After waiting a bit, you can **press the Spacebar** and play the animation.

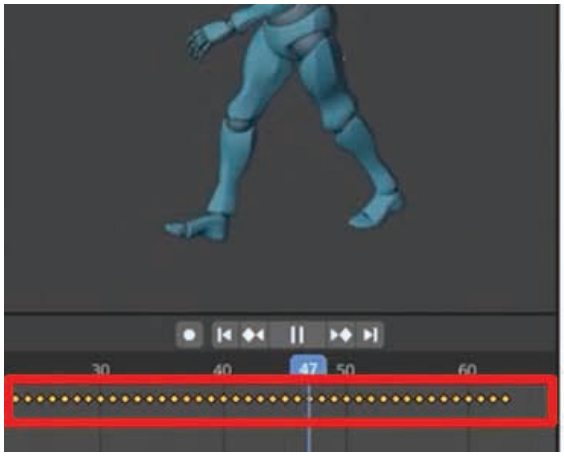


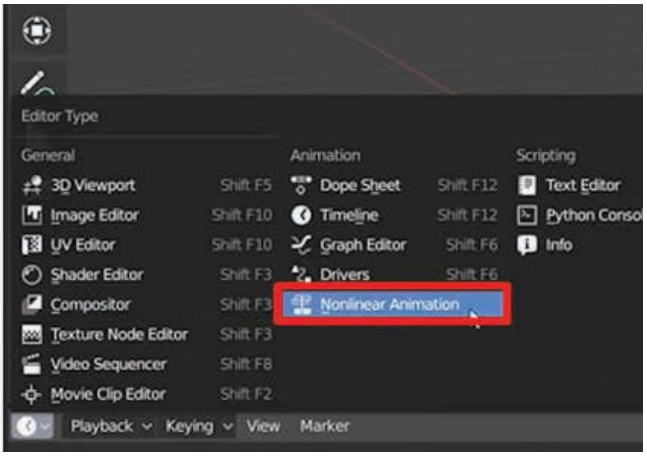
Mixamo walk cycle and how to create an Action from keyframes

First, find a walk cycle that you like on Mixamo. Make sure "**In Place**" is checked and download the animation.

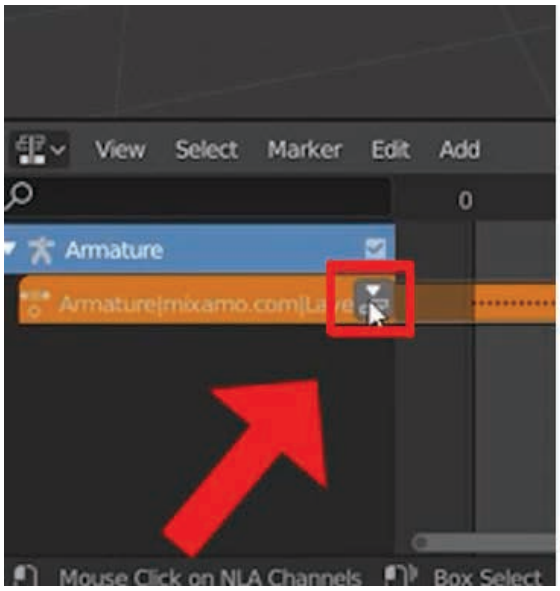


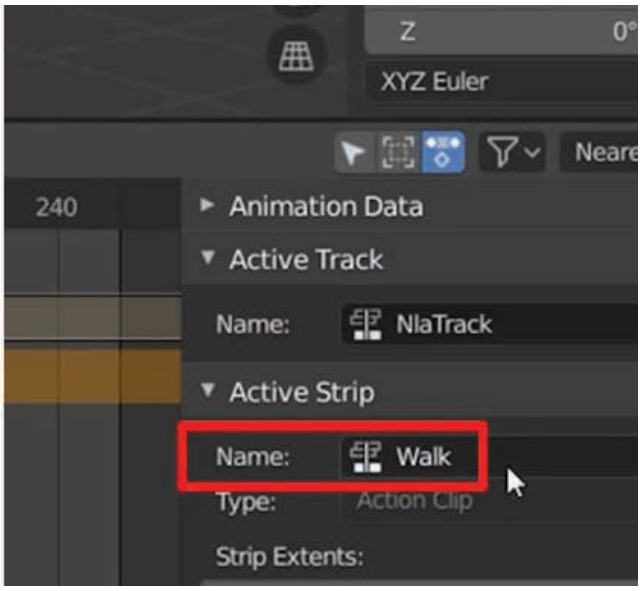
Import the file in Blender. The walk is short and not cyclical. We need to change this. Open the **Nonlinear Animation Editor**.





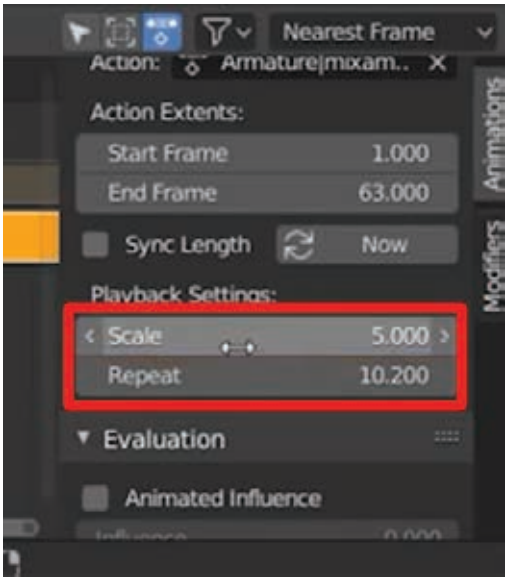
And click on this button to create an **Action**. It's always good to give actions a clear name.



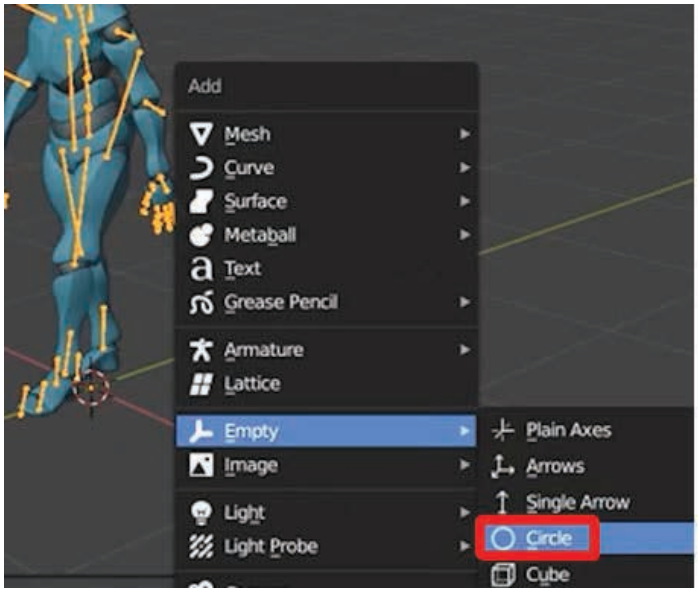


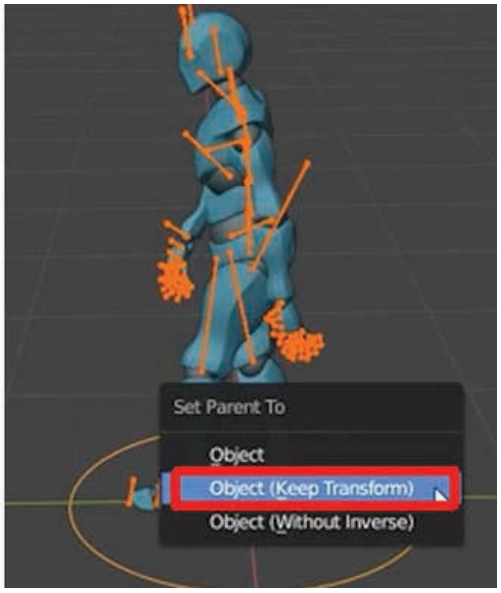
With the **Repeat** value you can set how many times the action gets repeated.

The **Scale** value makes the walk cycle faster or slower.



To move the character, create an **Empty** and parent the Armature to it in Object Mode. To do this, select the Armature, then **hold Shift** and select the Empty last. **Press Ctrl+P** and choose **Parent (Keep Transforms)**.



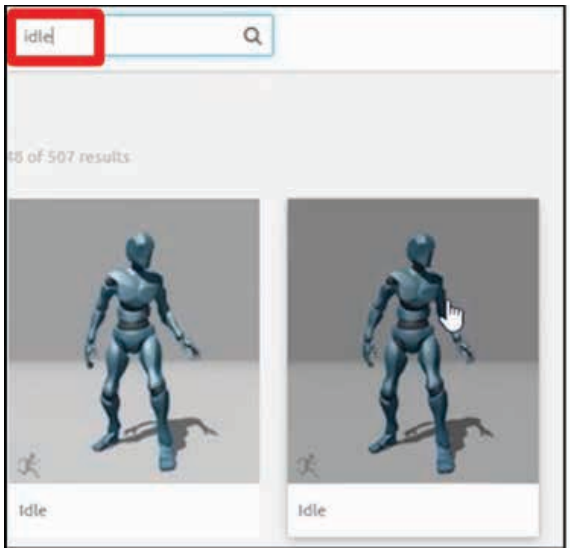


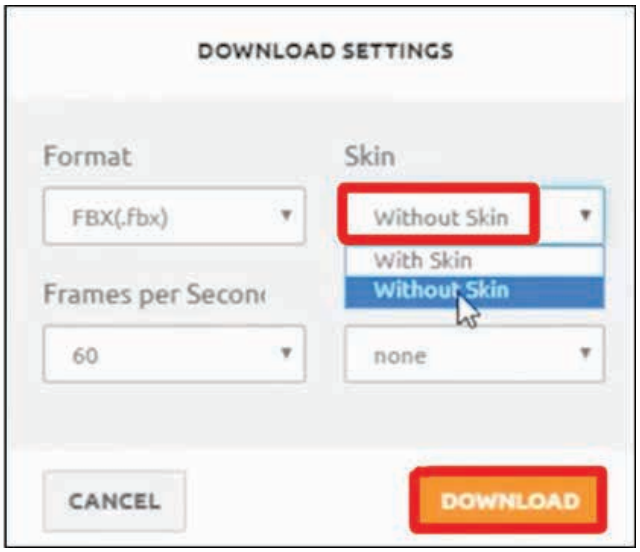
I used a Circle Empty here, but the type of Empty doesn't matter.

Now you can set keyframes for the **Location** (and Rotation, if you want) of the Empty and the character will follow.

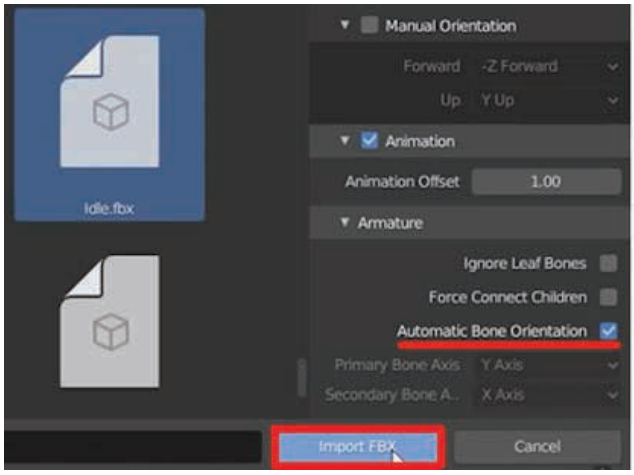
Combining Mixamo Animations

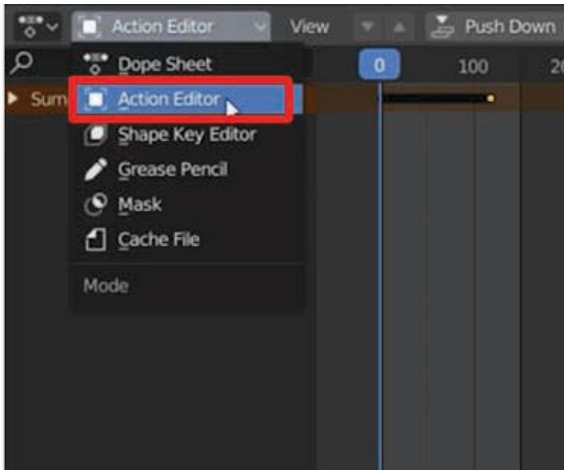
Search for an "idle" animation on Mixamo and download it. You don't need to download the skin. Idle means that the character isn't doing anything, just standing.





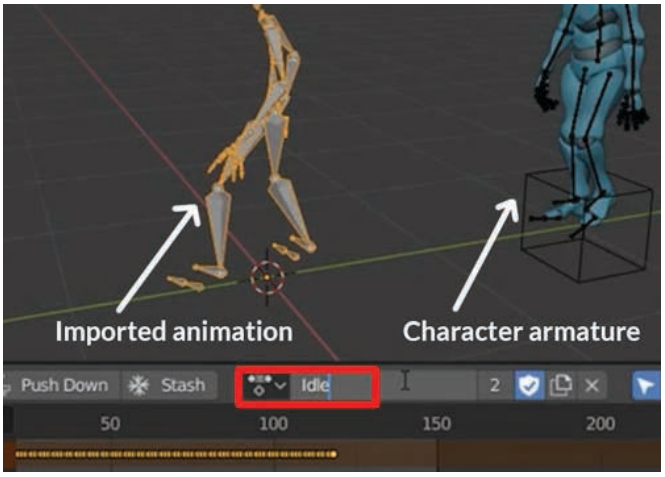
Import the idle FBX. Make sure to check "**Automatic Bone Orientation**". Open the **Dope Sheet** and then the **Action Editor**.



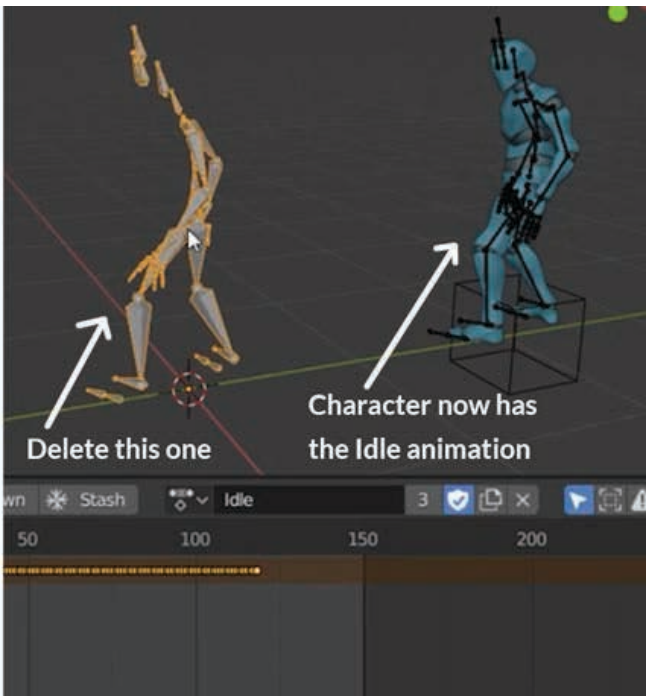


Give this idle animation a name.

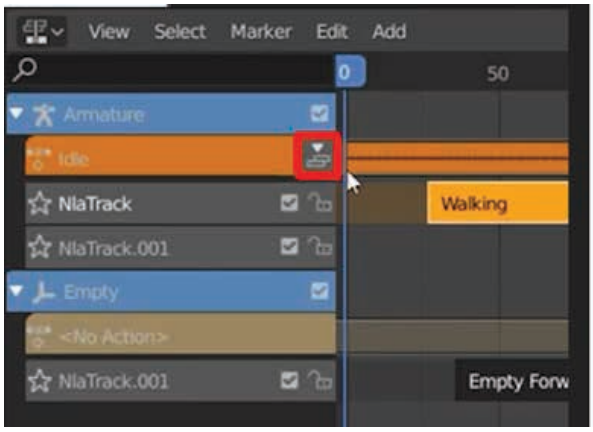
You now have the imported animation with its named "Idle" animation, as well as the character armature.

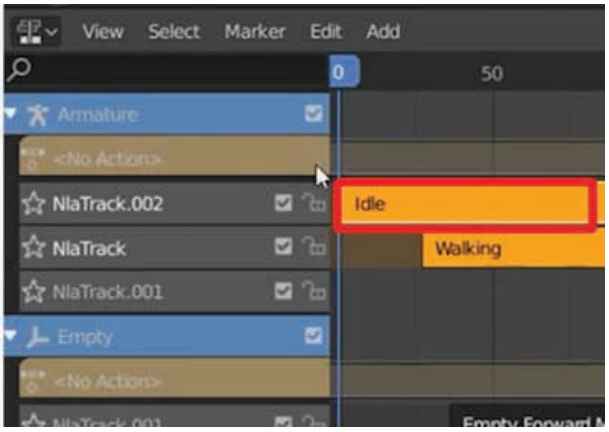


Select the character armature and then select the idle animation. You can now delete the second armature.



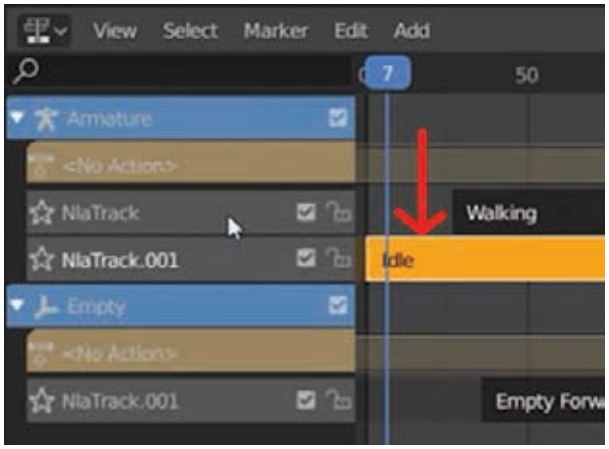
Go back to the **Nonlinear Animation Editor** and click on the button to make an action from the Idle animation. Now we have two actions on our character.



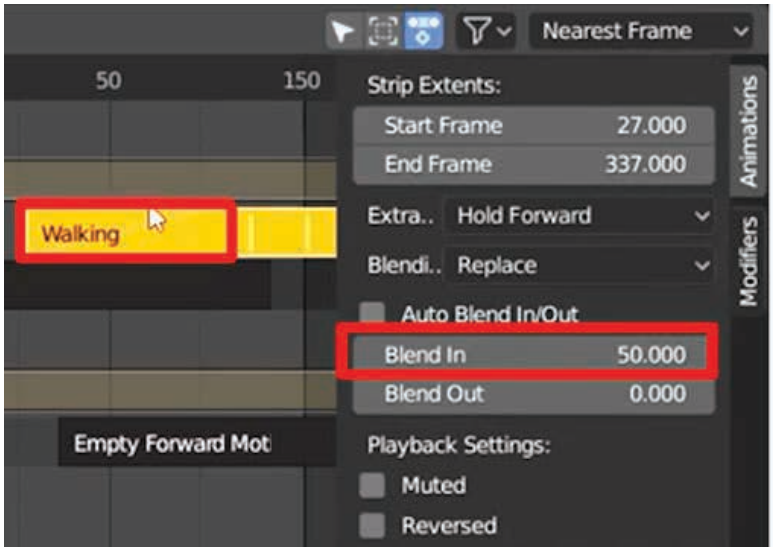


Whatever is on top in the Nonlinear Animation Editor takes priority.

Move the idle Action down by pressing **Page Down**.



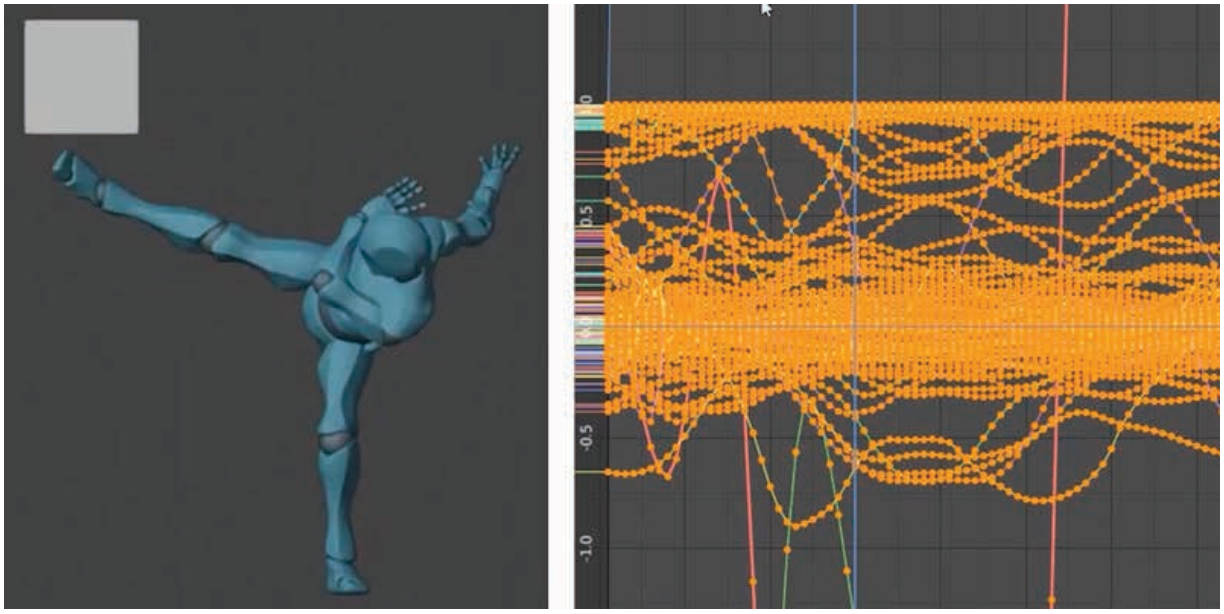
To "fade in" the walking we can set a **Blend In** value. The value stands for the **amount of frames** the action will blend in. Now the character stands still for a bit before he starts to walk.



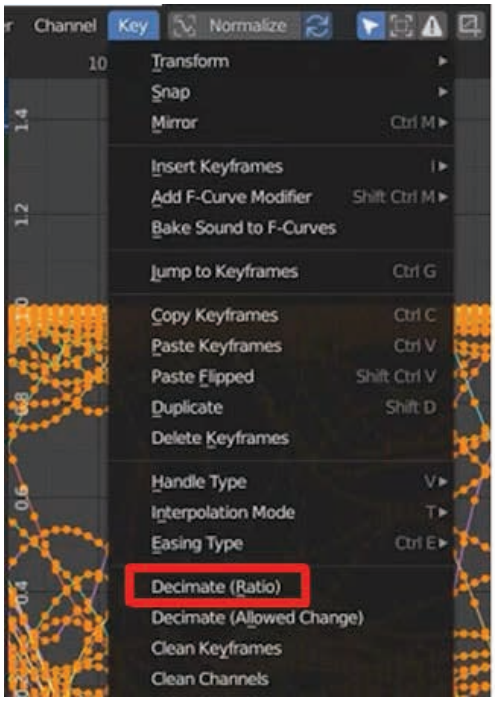
You'll need to experiment with the placement of the Actions, the **Blend** values and especially the **Scale** of the walk cycle until the feet don't slide anymore.

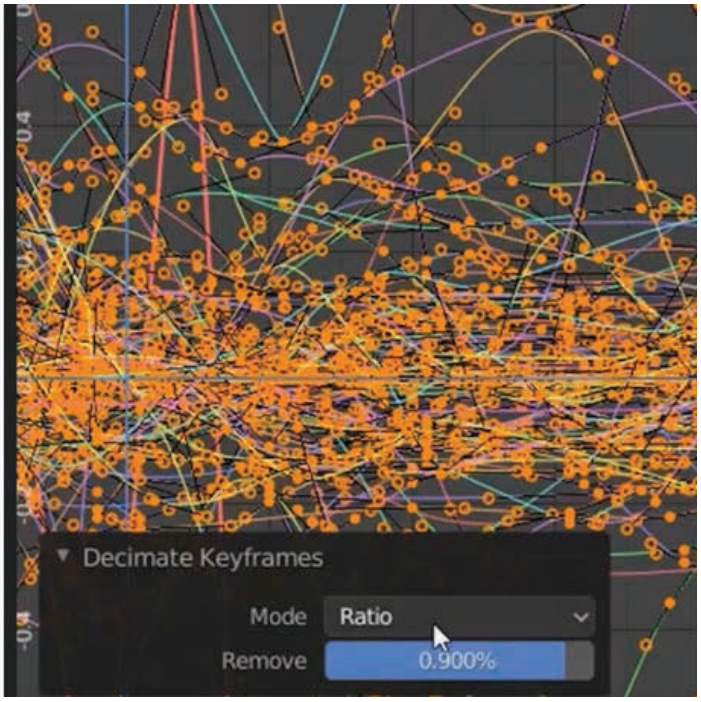
Adjusting Mixamo animation in Blender

Let's try to adjust this animation to make this character kick a bit higher to hit the cube. As you can see, the armature has a lot of keyframes.

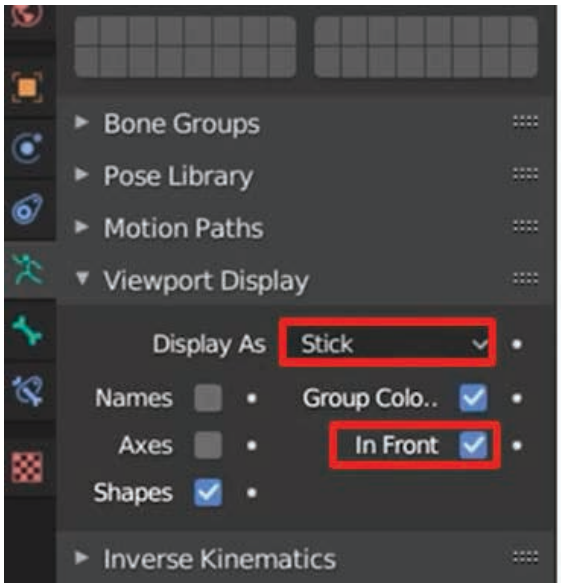


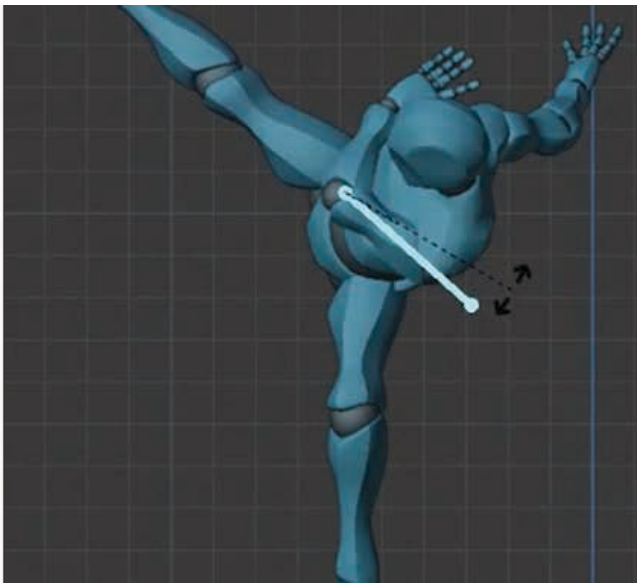
We can reduce them by using the **Decimate Ratio** function. You can decimate quite a lot and still retain the animation. In this case 90% decimation works out well.





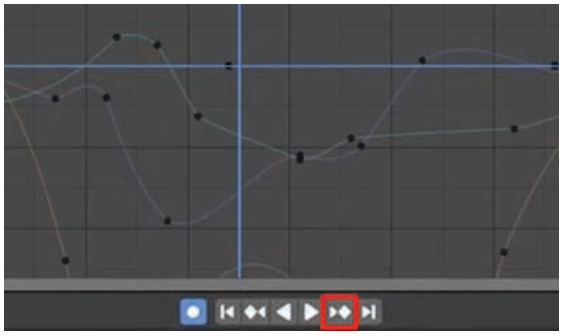
Turn on **In Front** for the Armature and set it to display as **Sticks** so we can see it better. Find the bone that you want to adjust and **press Shift+H** to hide the other bones.





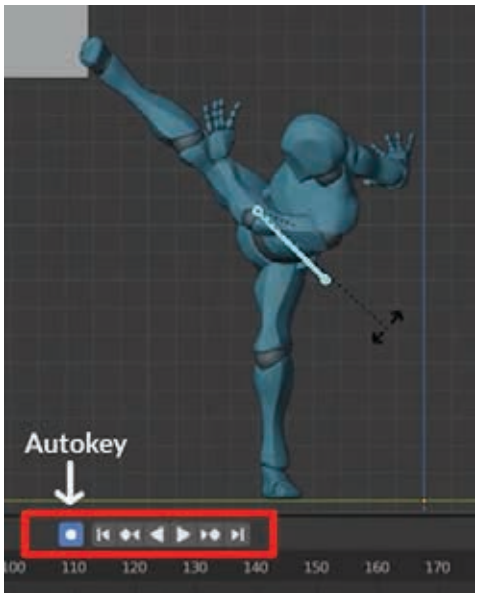
Now in the **Graph Editor** we only see the keyframes for that bone**.**

We can jump to the next keyframe and see which keyframes we need to change to make the character kick higher.



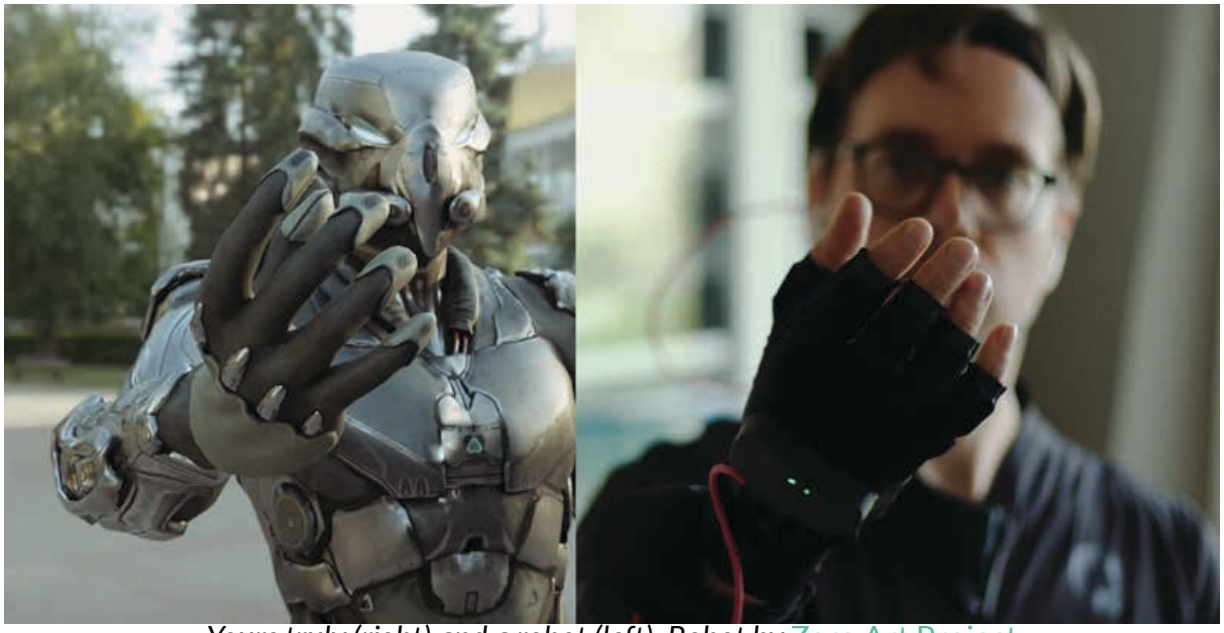
Turn on **Autokey** and just pose the bone the way you need to in Pose Mode.

Now the character kicks higher.



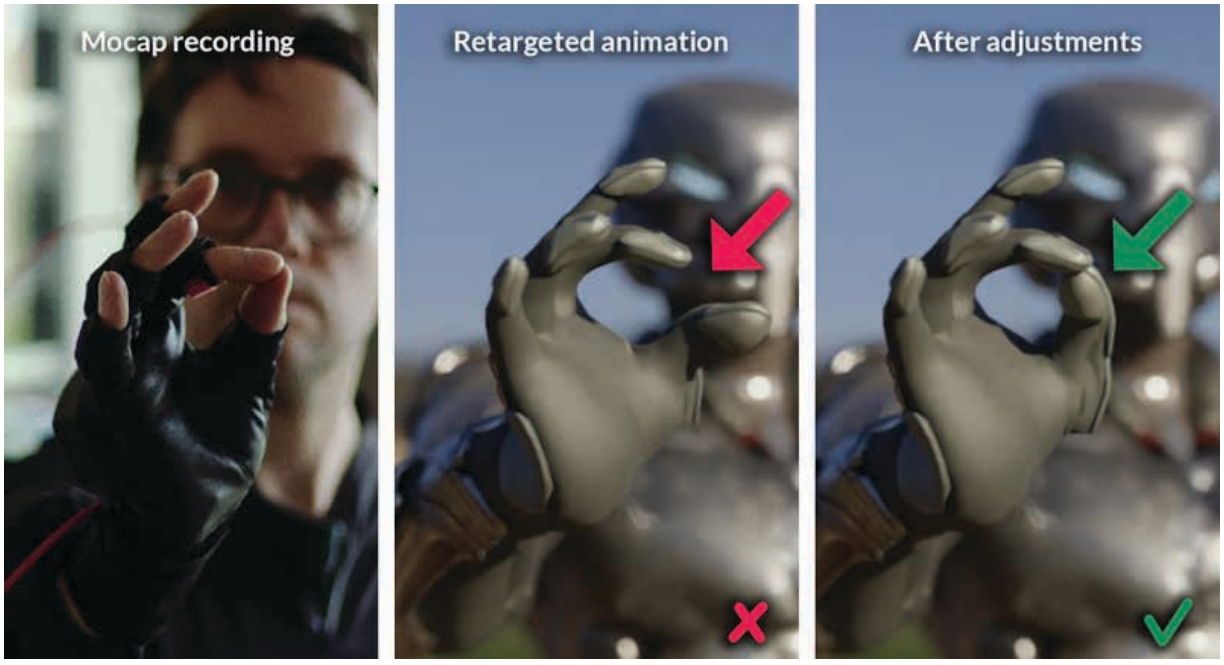
Cleaning up mocap and adding an adjustment layer

Although mocap software like Rokoko Studio contains filters to clean up your mocap recordings, retargeting them to a different armature in Blender can often mess up the finer details of the movement, often because of small differences in the **Rest Pose** of both Armatures.

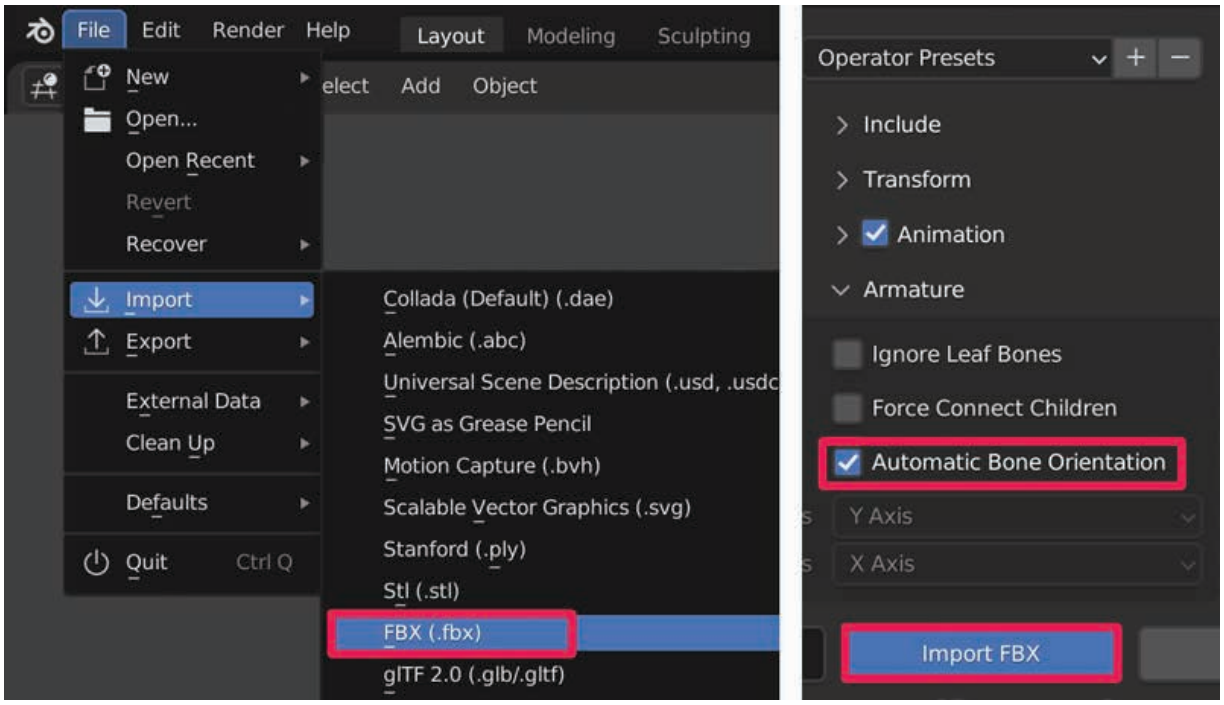


Yours truly (right) and a robot (left). Robot by Zero Art Project

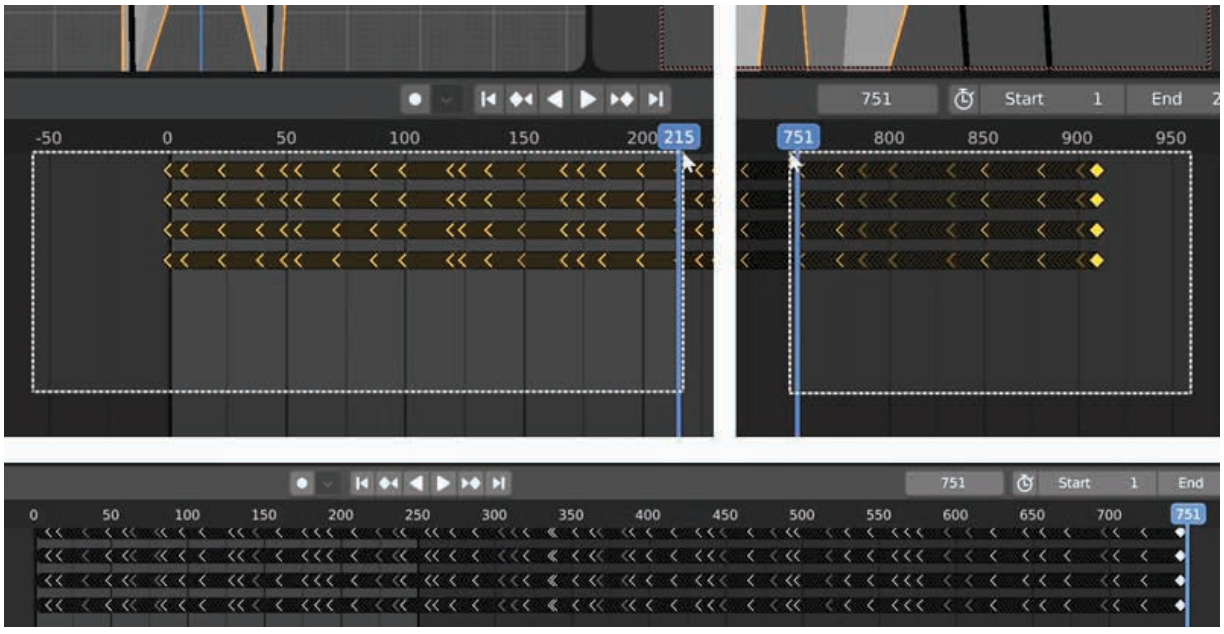
This is where knowing how to clean up and adjust animation with a second layer of keyframes comes in handy.



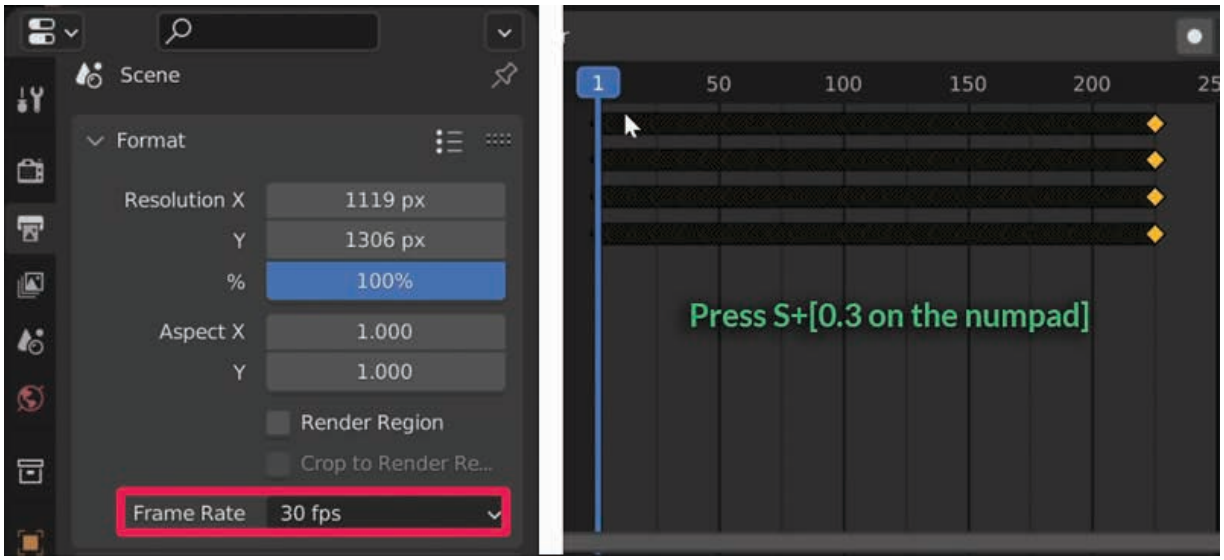
After exporting the mocap animation from your mocap software of choice (if possible, export only a Segment instead of the whole thing) import the FBX mocap file with **Automatic Bone Rotation**. *(Note for Rokoko users: In Rokoko Studio, I prefer to export using the default Rokoko skeleton or the Maya one - the Mixamo one seems to not be very reliable.)*



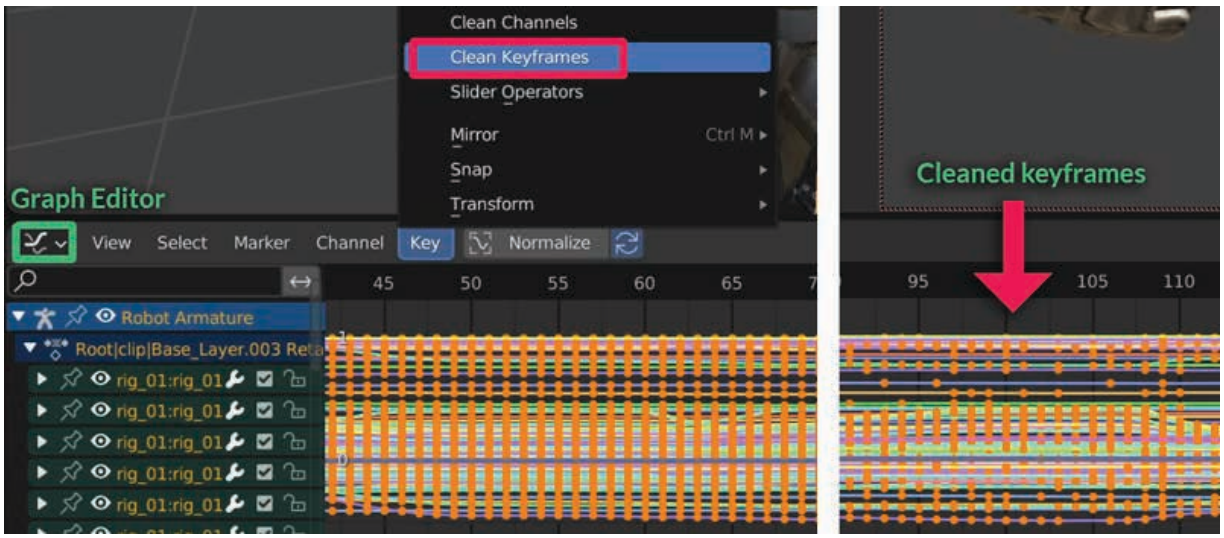
In the **Timeline**, delete any keyframes that you don't need by selecting them (you can drag a box selection around them) and **pressing X**. Select all keyframes and move them to the start of the Timeline by **pressing G** and moving the mouse.



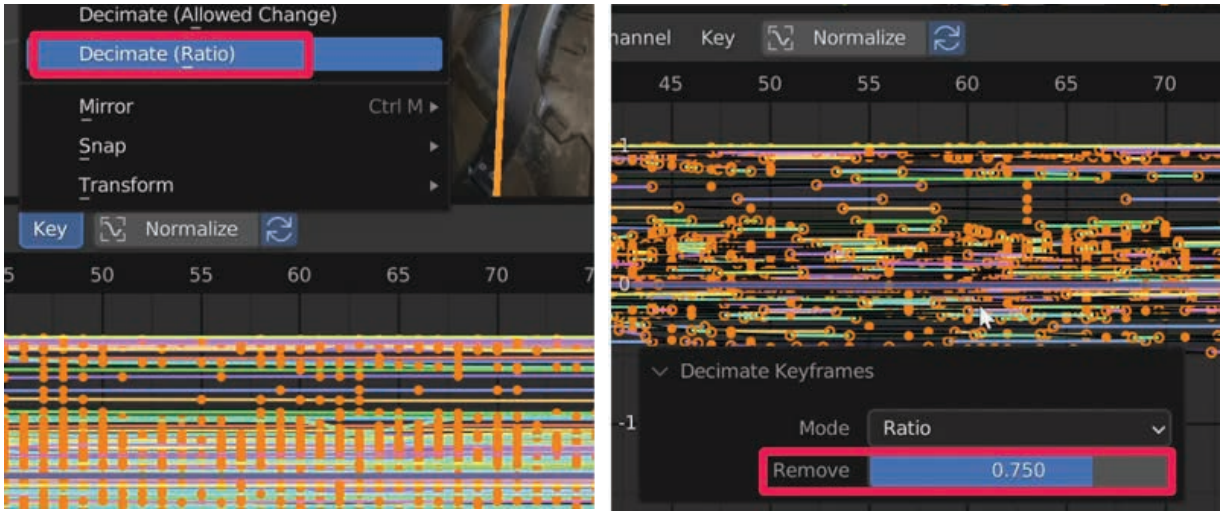
To scale a 100 frames per second recording (the default in Rokoko Studio) to 30 frames per second for a project set to that **Frame Rate**, select the keyframes and **press S+[0.3 on the numpad]**. Or for 25 frames per second, **press S+[0.25 on the numpad]**. That way the animation moves at the correct speed, and it'll be easier to compare it to video reference if needed.



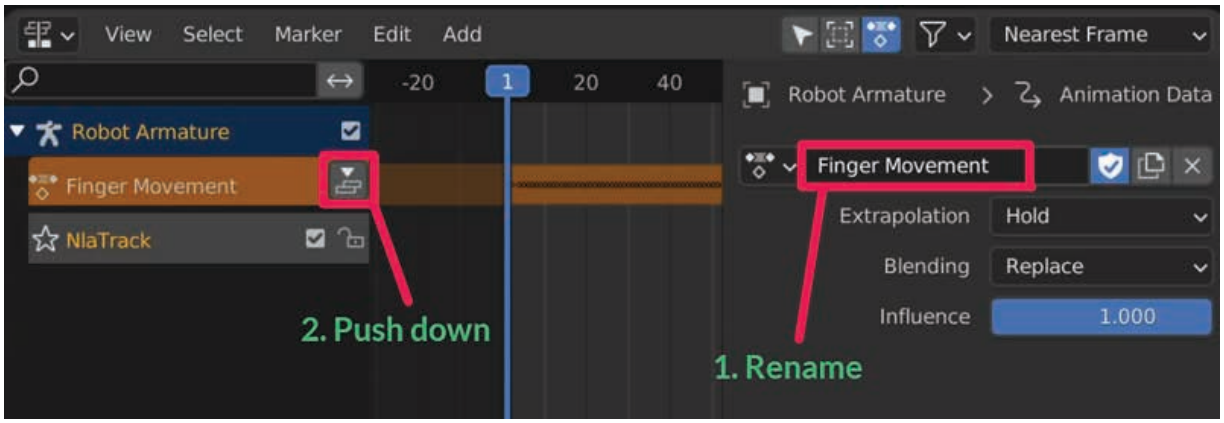
Retarget it to your character as explained in this previous video. In the Graph Editor, go to **Key > Clean Keyframes** to reduce the amount of keys.



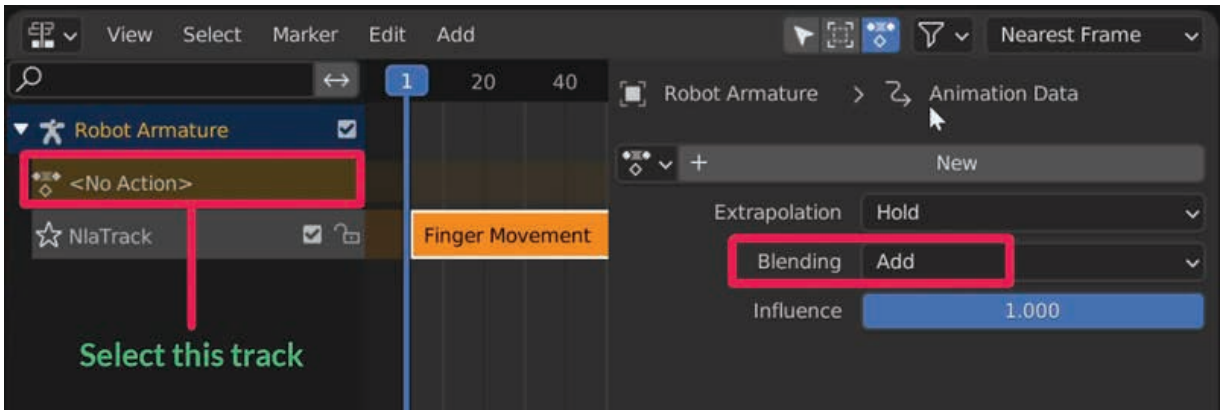
To reduce even more keyframes, go to **Key > Decimate Ratio** and **press Enter**. In the pop-up options, you can type a value and **press Enter**. A value of 0.75 for example decimates 75%, or leaves only 25% of the original frames. In your case, maybe less decimation is needed. It's something you can experiment with.



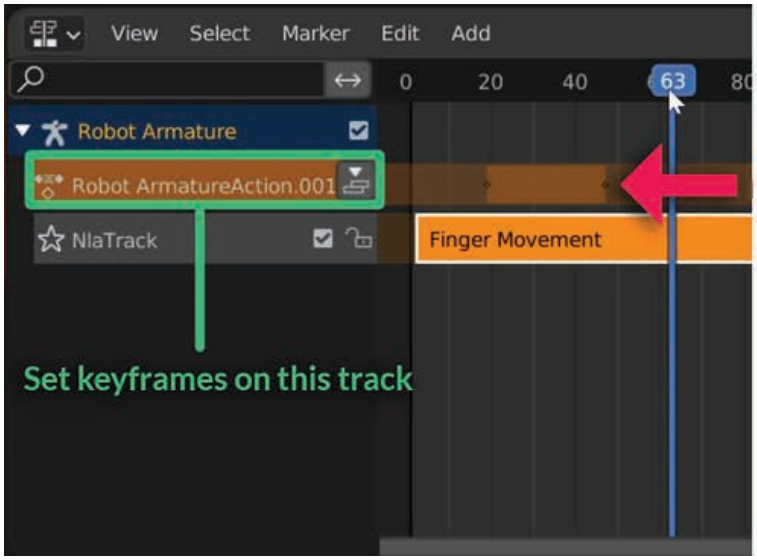
In the **NLA Editor**, rename the animation and press the result down to an **Action**.

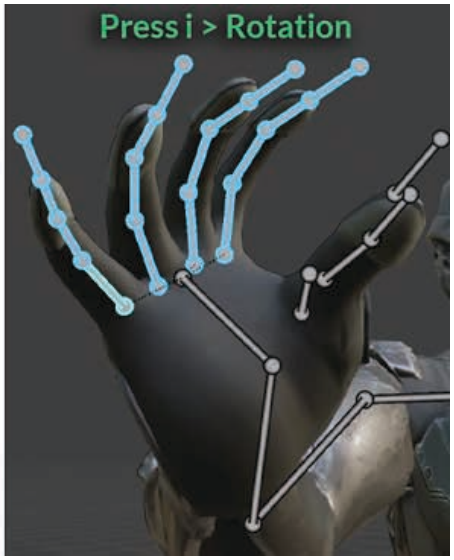


Set the **Blending** of the top layer to **Add**.

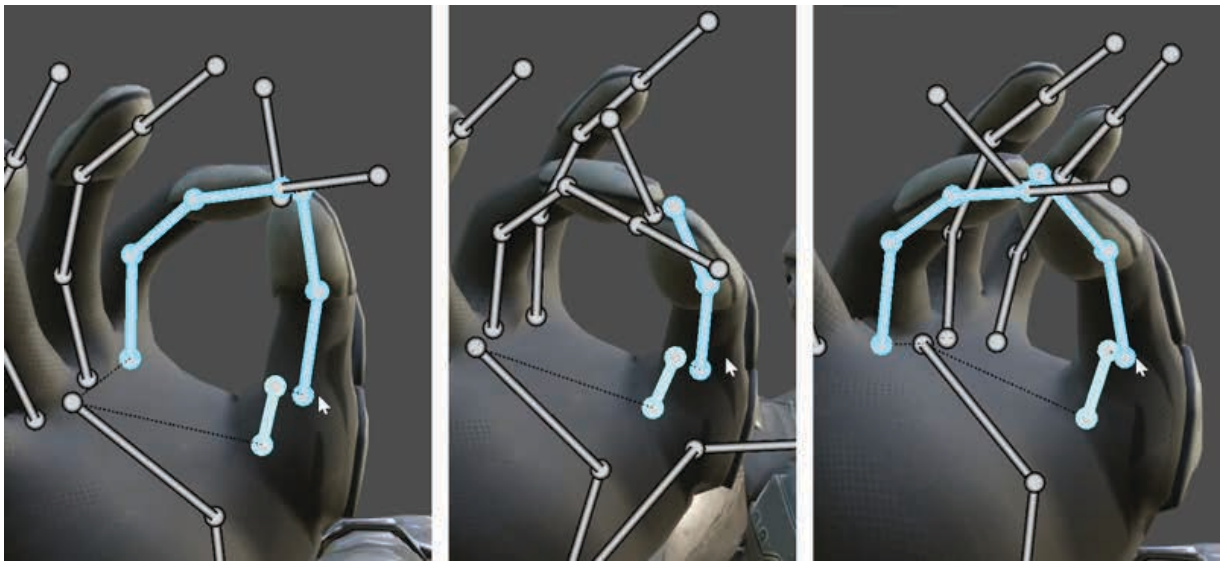


Now, first set keyframes for all bones where the animation is perfect. You can set keyframes for bones by selecting them in **Pose Mode** and **pressing i**, then choosing **Rotation**. This step protects the parts that don't need to be improved. In this case I'm only doing it for the finger bones minus the thumb, for the parts where the fingers are extended.





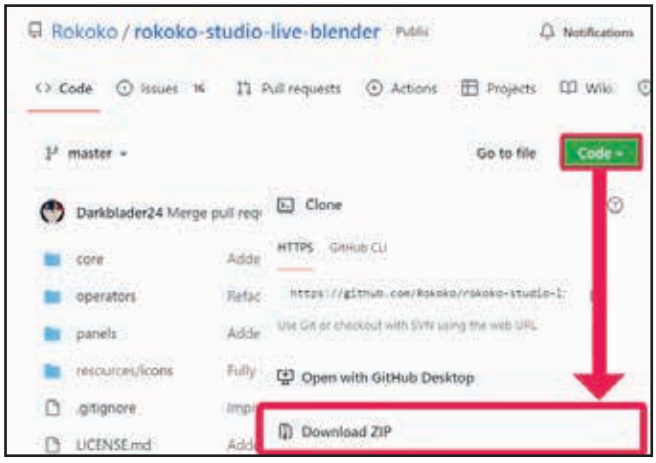
Then set additional keyframes in between those to adjust the animation as needed. This way we can really fine-tune the animation.



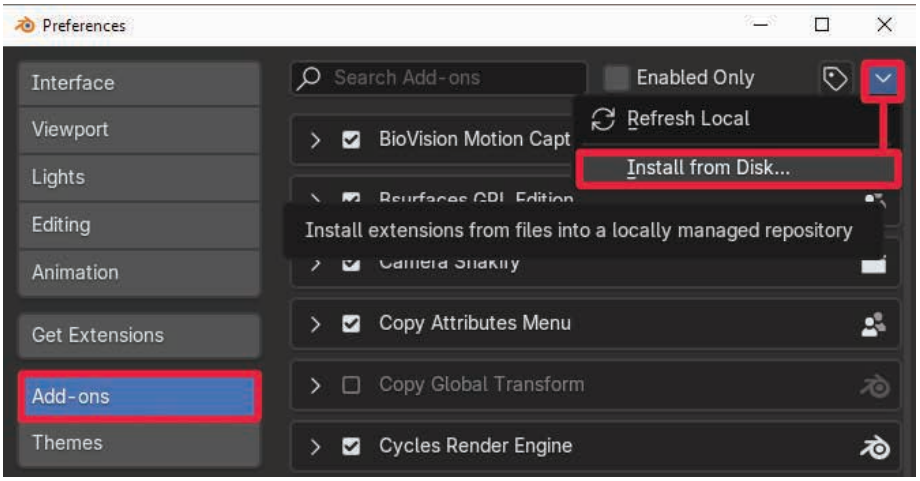
Retargeting with the free Rokoko add-on

Mocap suit makers Rokoko released this free add-on for retargeting existing mocap animations for use with your rigged characters in Blender. These can be made with Rigify or Mixamo for example. It works quite well!

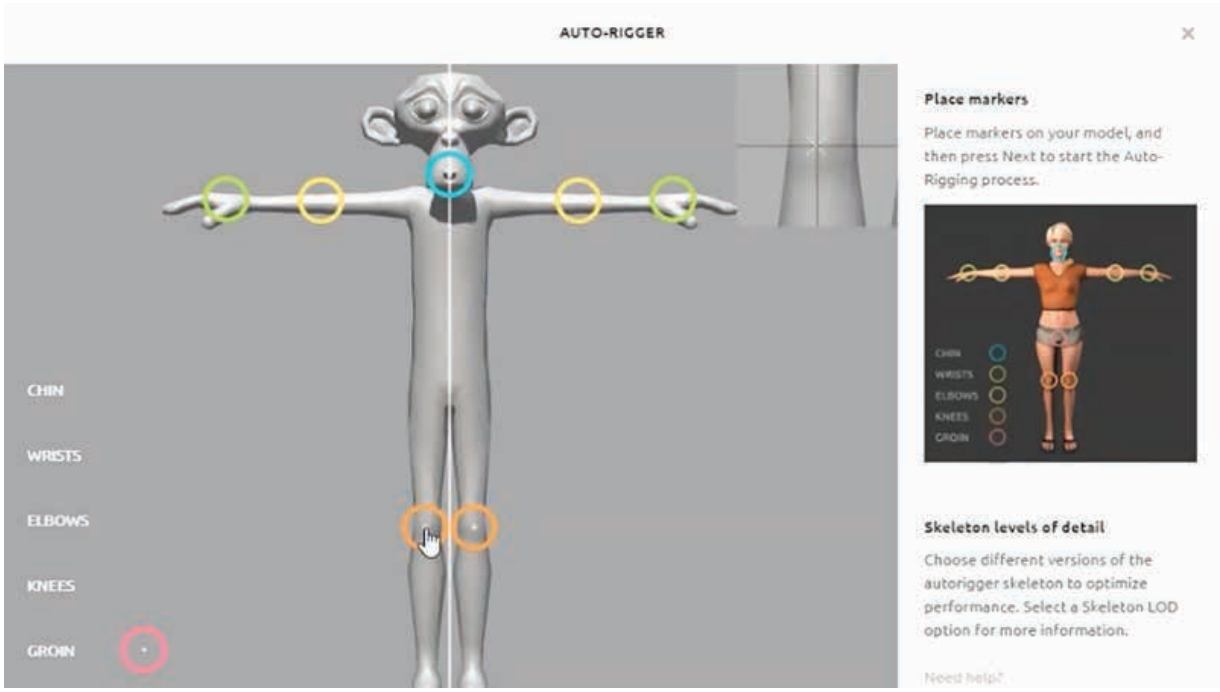
Download the add-on here. On Github, click on **Code** and choose **Download Zip**.



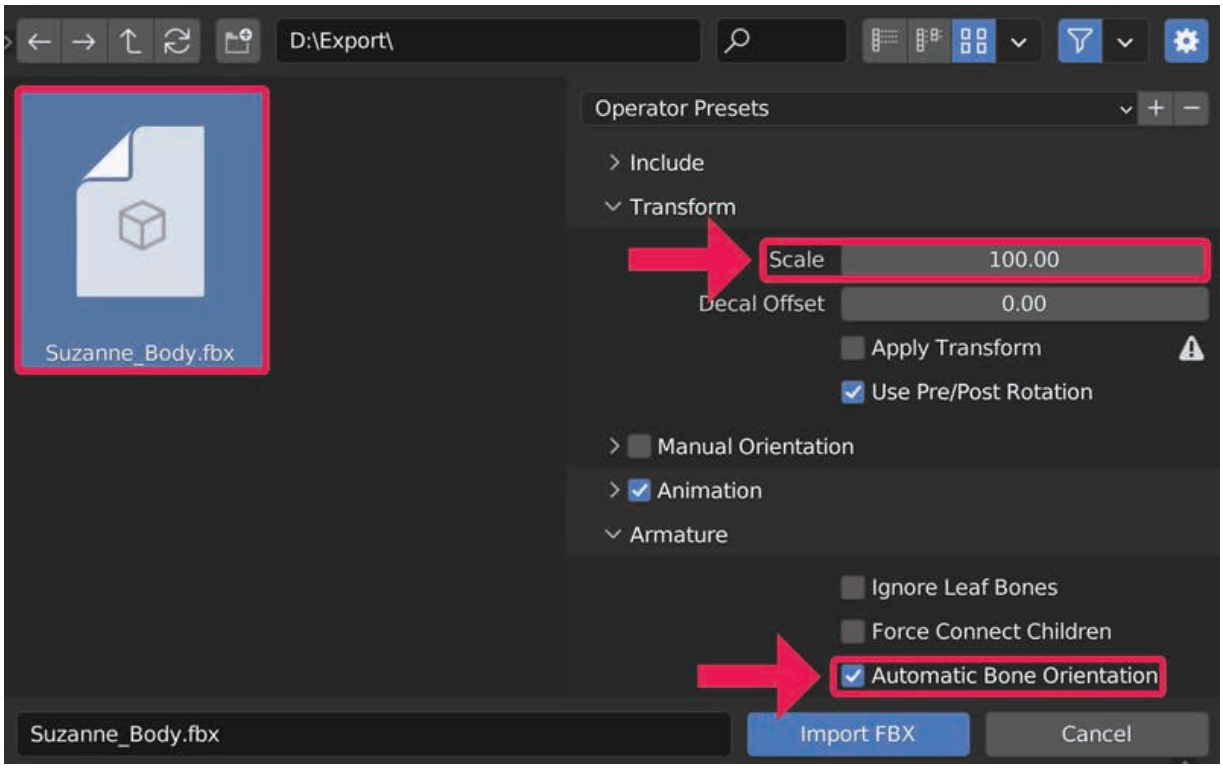
Don't unzip the download, but install it via the "**Install from Disk**" option in Preferences > Add-ons.



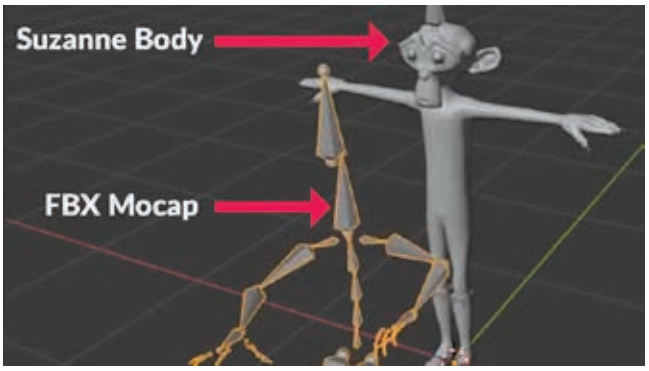
Rig your character with Mixamo. If you don't know how to do this, check out my video about rigging with Mixamo. Watch auto-rigging with Mixamo here.



Import your rigged mesh and check Automatic Bone Rotation and Animation. You may also have to change the scale to 100 otherwise the Mixamo mesh can be very tiny. The scale depends on where you got the file although the add-on can take care of the scaling automatically.

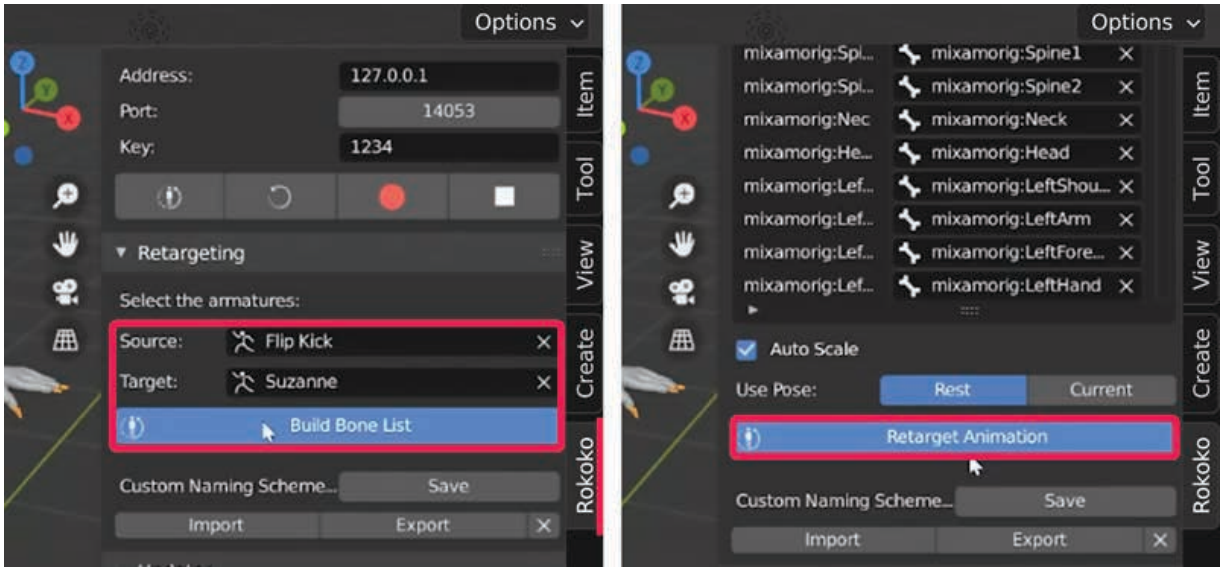


Then load your BVH or FBX mocap file.



You can find the Rokoko add-on in the Option tab. You'll need to register for a free account so you can log in.

Go to **Retargeting** and in **Source** choose the animation which you imported, then in **Target** choose the Armature of the character. Click **Build Bone List** and after that's finished, click on **Retarget Animation**.



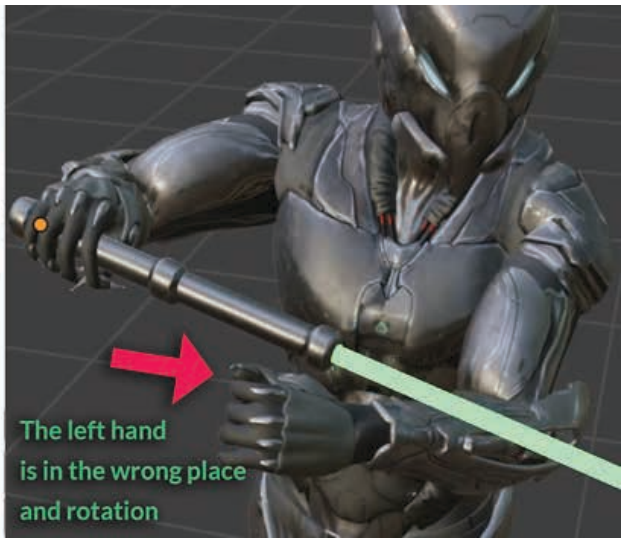
Normally this is all you need to do. If it didn't work, check if the Bone List has missing bones that you need to enter manually, then click Retarget Animation again.

Two-handed grip animation using Inverse Kinematics

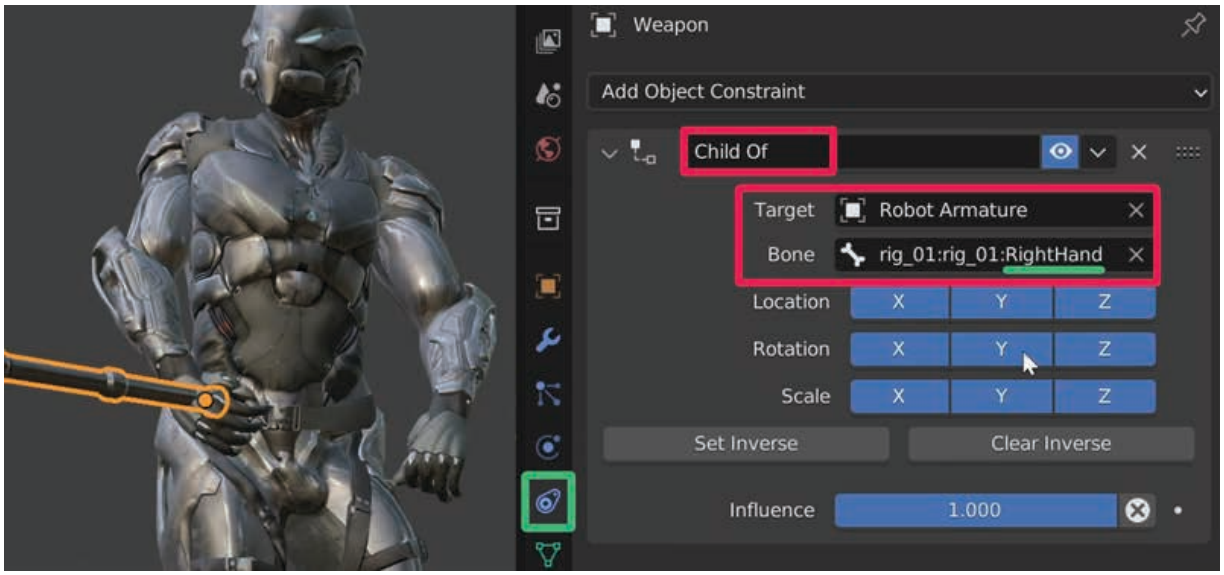
When you've recorded a motion capture animation of a character holding a sword or some other weapon with both hands, the hand positions may not be exactly where you want them to be.

Yours truly (left) and a robot (right). Robot by Zero Art Project

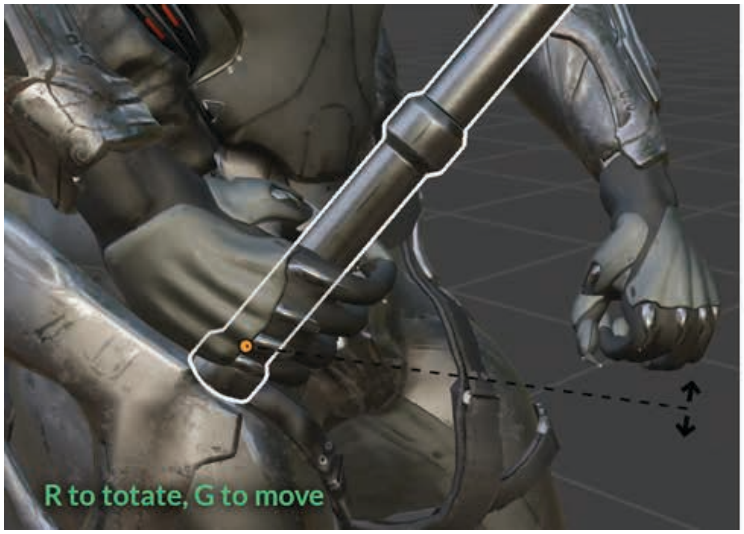
You can manually adjust the hand animation, but a much easier and more accurate method uses IK Bone Constraints. It's pretty easy to set up if you know how.



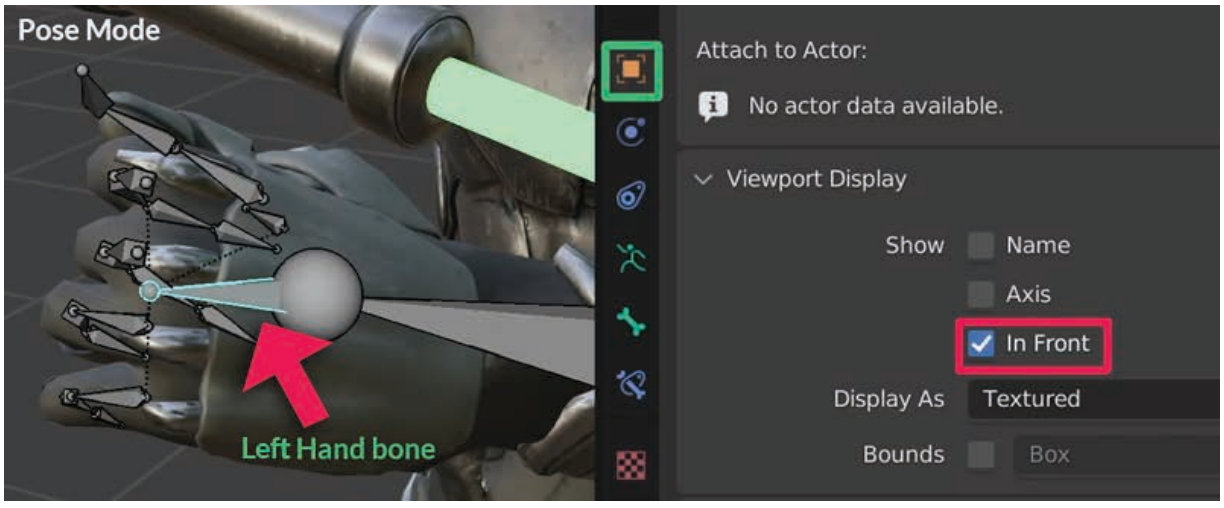
In this case, the character is holding a stick weapon… and both hands are firmly gripping it. After loading the .FBX animation file in Blender and retargeting it to a character (as explained previously) we parent the stick weapon to the right arm by putting a **Child Of Object Constraint** on the weapon object.



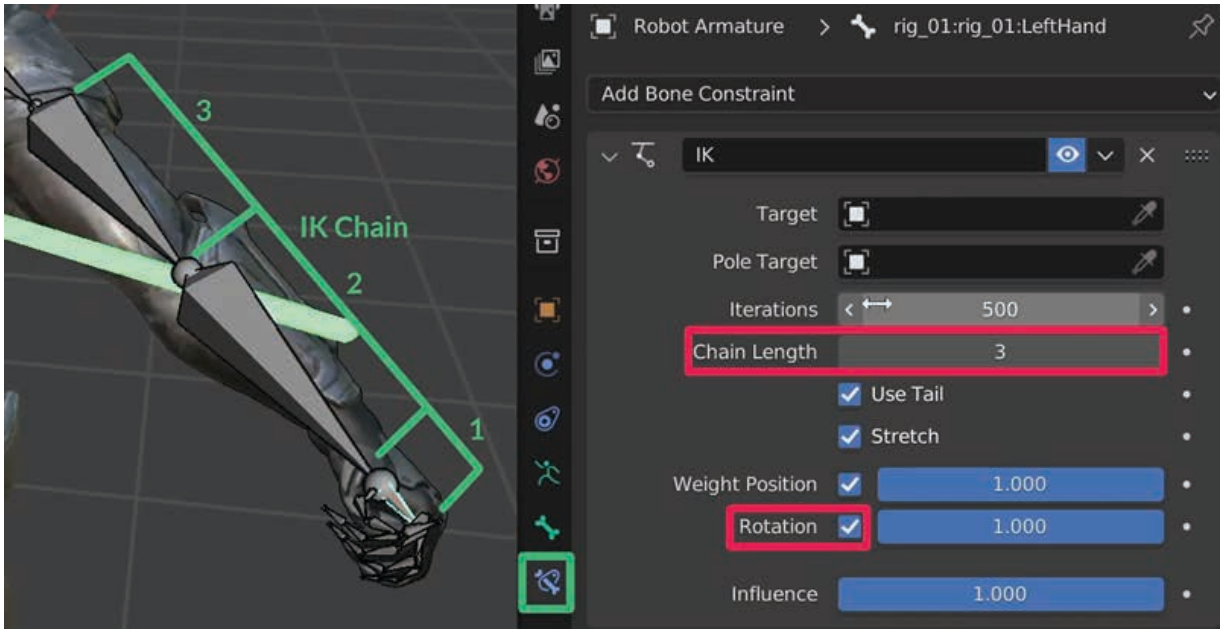
As soon as we pick the hand bone of the right arm as the Target, the weapon snaps to the correct location. We just have to rotate it correctly in Object Mode by **pressing R** and maybe tweak the exact location a little.



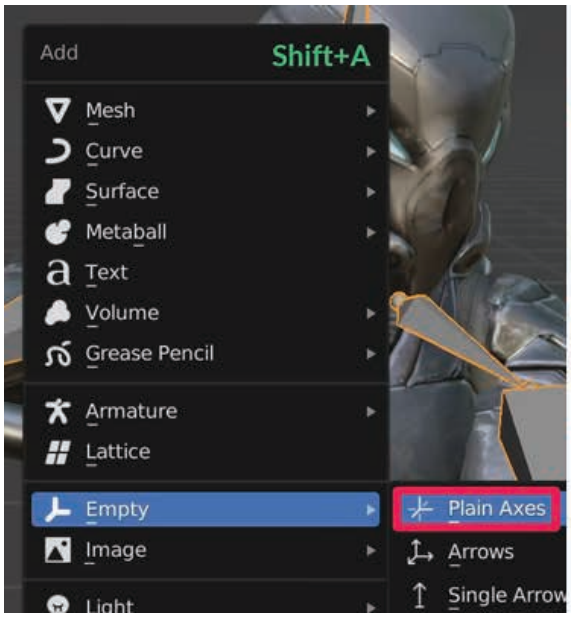
In Object Mode set the Armature to **In Front** in the Viewport Display options, so it's easier to see and select bones. In **Pose Mode**, select the **left hand bone**.

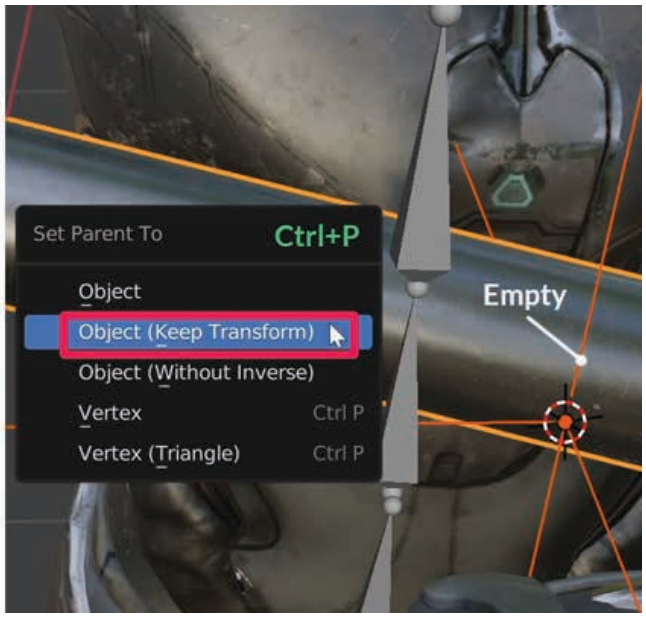


Add an **IK Bone Constraint** to it. Set the **Chain Length to 3** so that when you move the hand bone, the rest of the arm follows it. Check **Rotation** as well.

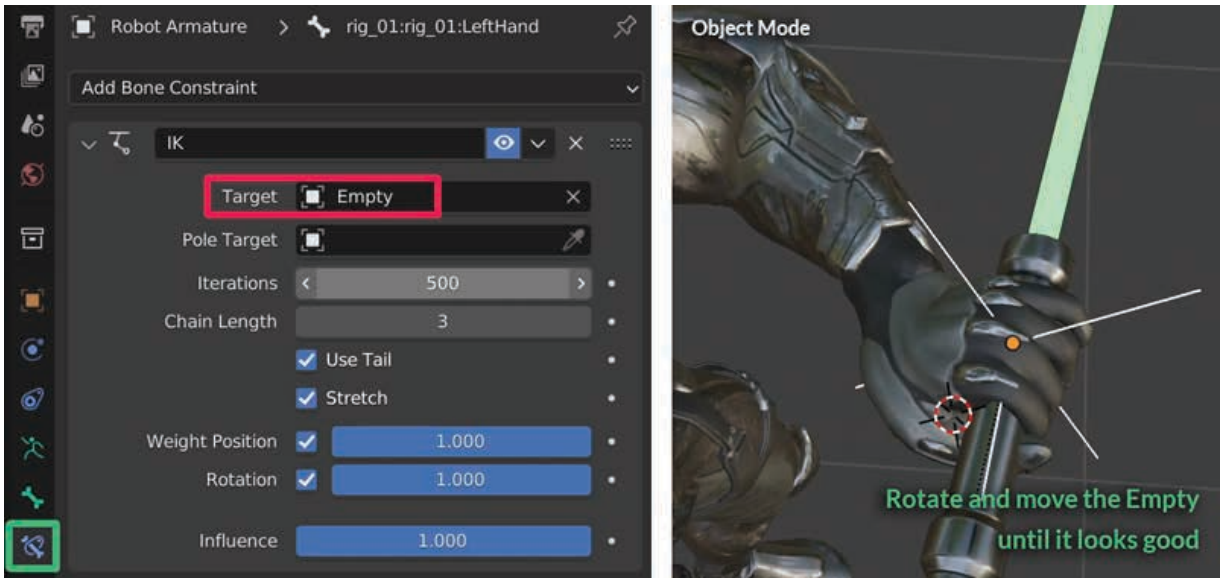


Add an **Empty** and place it near or on the prop where you want the second hand to grip it. Parent the Empty to the Weapon by selecting it, then the weapon, and **pressing Ctrl+P > Object (Keep Transform)**.





To snap the left hand bone to the weapon, set the Empty as the **Target**. You can control the exact placement and rotation of the hand by moving and rotating the Empty.

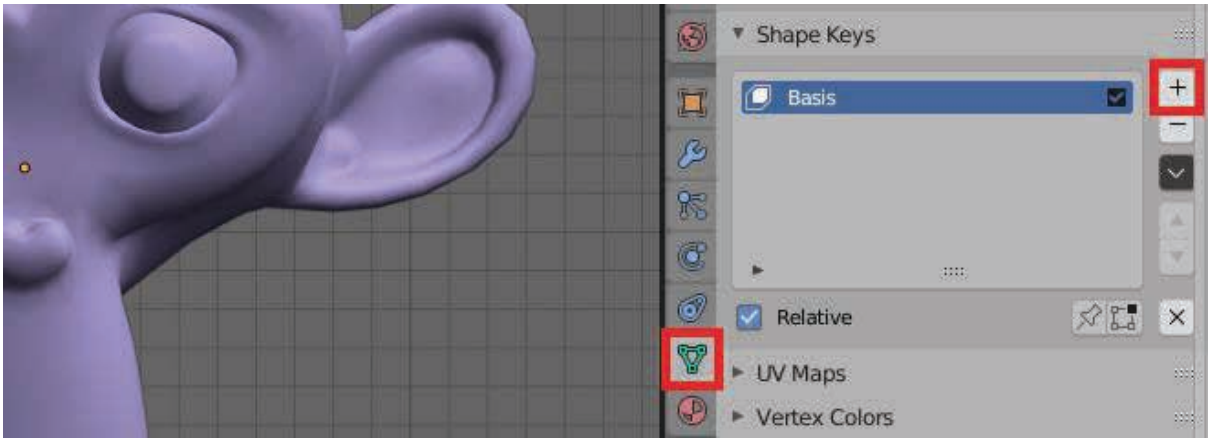


That's all, now the left hand is "stuck" to the weapon and follows it. You can change the Influence value of the **IK Bone Constraint** on the passive hand, in case you want to remove the IK influence. This value can be keyframed.

Shape Keys: Basics

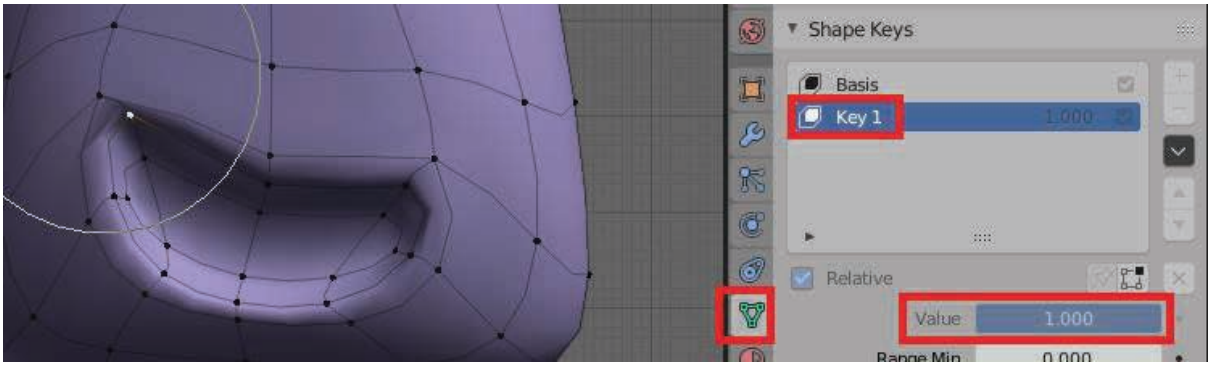
In the previous tip we looked at adding Shape Keys from other objects with the same vertex count. If possible, ideally you would just add your Shape Keys directly to the Object you want to be able to animate with the Shape Keys. In this tip we'll examine how to do that.

Select your object. Create a **Shape Key** by **pressing the + icon**. This is automatically named the "Basis" key, and is the default shape of your object.



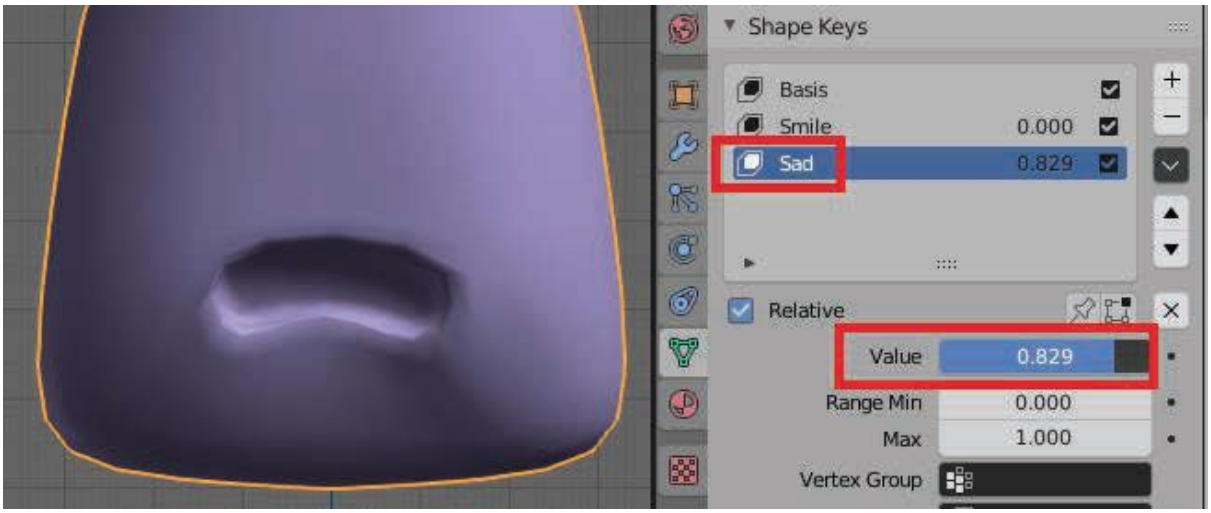
Create another Shape Key (it will be called "Key 1" automatically). Set the Value of that new Key to 1 and edit the mesh in Edit Mode.

Use proportional editing to edit the mesh (**press o** to toggle proportional editing and scroll the mouse wheel to increase/decrease the radius). After editing the shape key for the mouth (making a smile, for example) you can control the Value of the new shape with the Value slider in Object Mode.

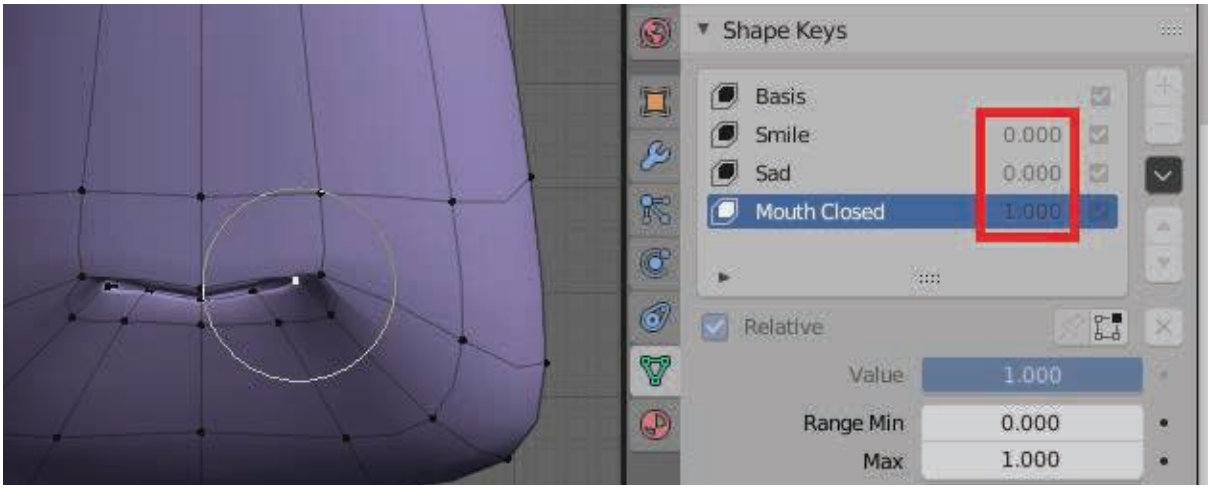


Editing the shape of the mesh in Edit Mode with Proportional Editing.

You can create many Shape Keys, all for example for different mouth shapes or facial expressions. Be sure to give them clear names, so you don't get confused later.

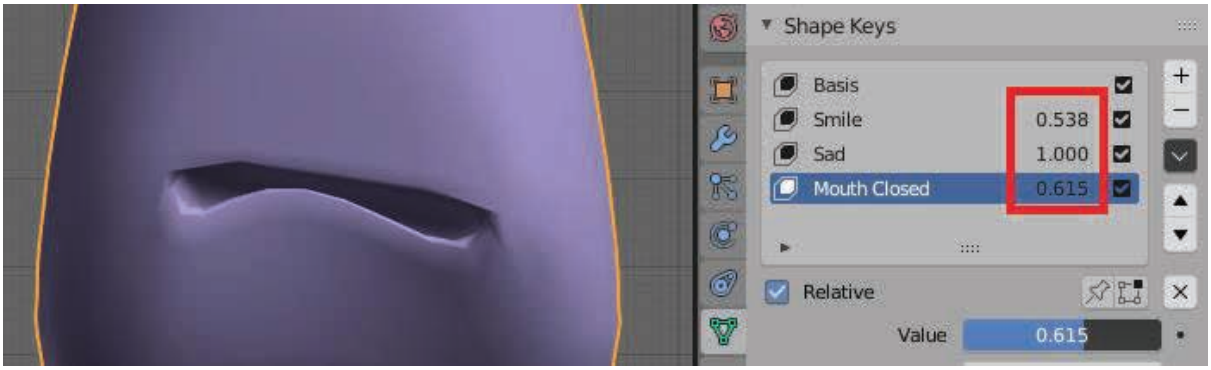


While making and editing a new Shape Key, make sure the other Shape Keys are set to 0, otherwise what you see is not the correct shape of the Shape Key you are currently working on, and it will get confusing later when mixing Shape Keys.



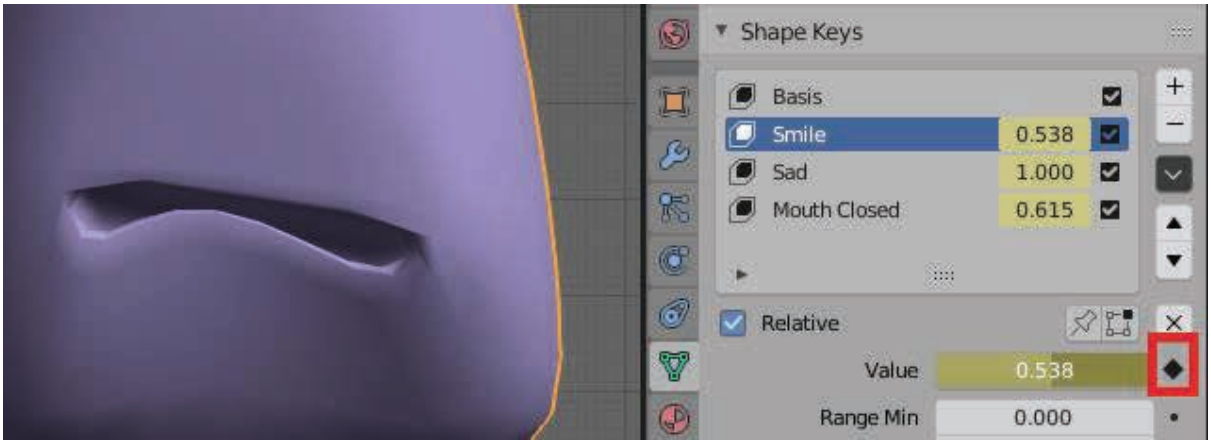
Proportional editing is your friend when editing facial shapes.

You can mix Shape Keys to create an infinite amount of unique shapes. This is really a fast and fun way to animate the face of a character without using bones.



A mix of three Shape Keys, creating a unique… expression.

For animation you need to record these Value changes over time. To record keys, press "i" while hovering the mouse cursor over the "Value" slider. Alternatively you can click the black dot next to the Value slider while you're at the right position on the Timeline. However keep in mind you'll need to do this for every Shape Key value individually.

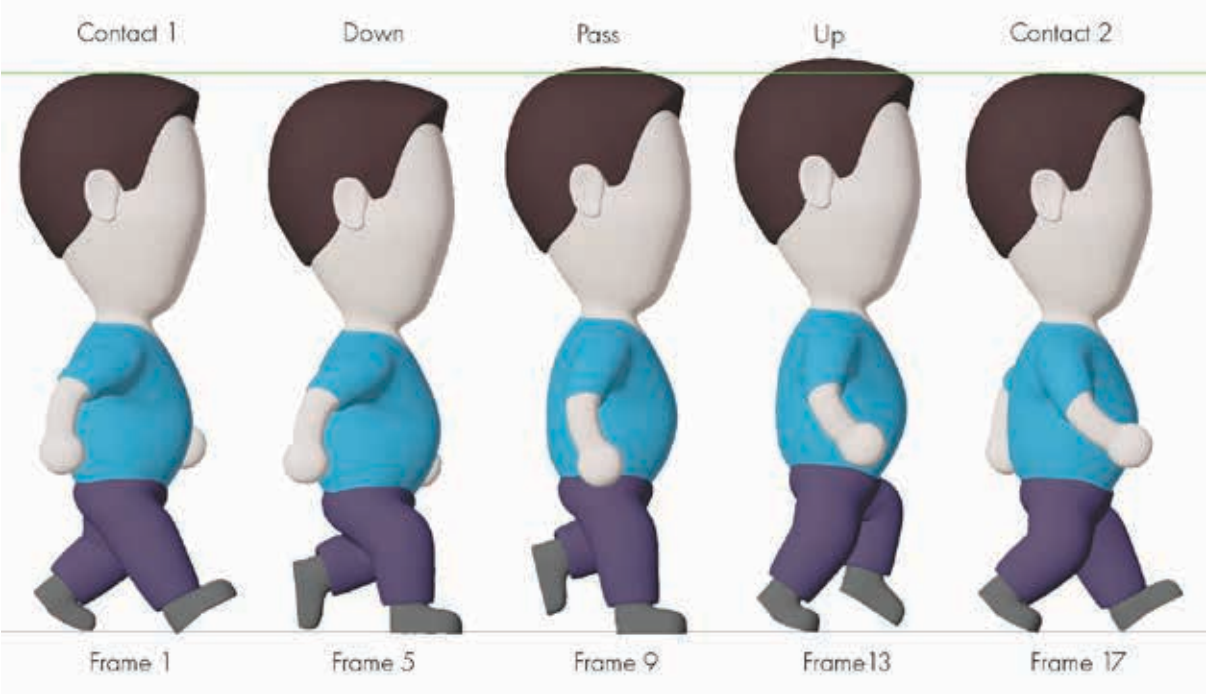


The three Shape Keys have been recorded as a key frame.

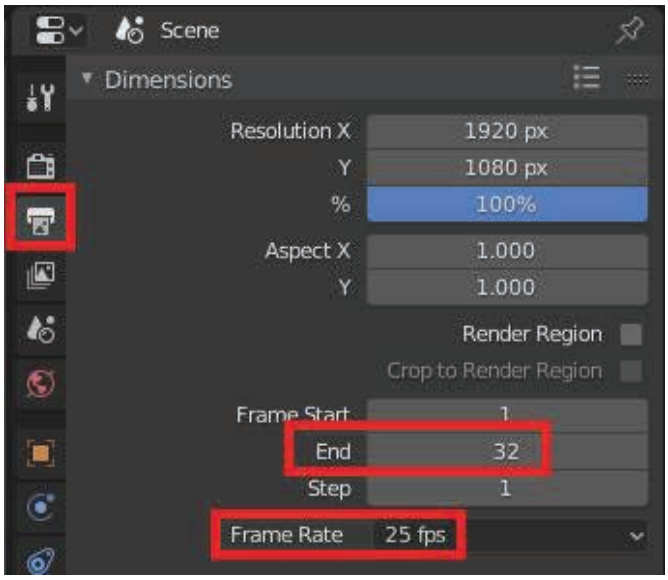
The color of the slider will change to indicate it has recorded the key frame. Go to a later frame in the timeline, set a different value and record another keyframe. This is how you can animate a face with Shape Keys.

Simple Character Walk Cycle

Let's make our simple character walk. A walk cycle is one of the most character-defining animations, and it is also one of the most difficult ones. If this is your first time animating a walk cycle, don't be disappointed if it doesn't look great on your first try. It really takes practice. I've made many walk cycles but I still have to peak sometimes at my "cheat sheet" of key poses:

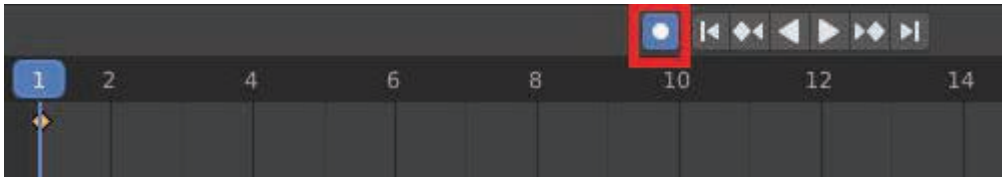


We will create a 32 frame loop. Set the Timeline to 32 frames, so we can easily see the loop.I'm using 25 frames per second as that is the default in Europe. For movies it's 24. I suggest that you use these settings until you've familiarised yourself with making walk cycles, before changing to your preferred or country-specific framerate.



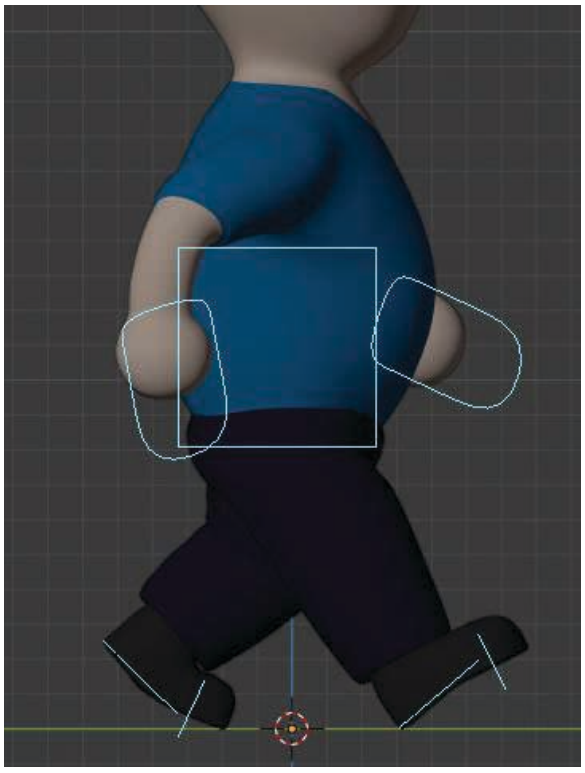
Contact Pose 1

Go to the left side view (**press Ctrl + 3** on your numpad) and start posing the first *contactpose*. Once you are happy with the pose, select all the bones by pressing A (they will all turn blue) and set a keyframe with "i". You could also turn on Auto keyframe recording.



The auto keyframe recording button is turned on.

Here I've hidden the unimportant bones - the visible bones are the ones I used to create this first Contact pose. The square in the middle lets you move the character down slightly. The hands and feet have been positioned. The arms and legs follow automatically.

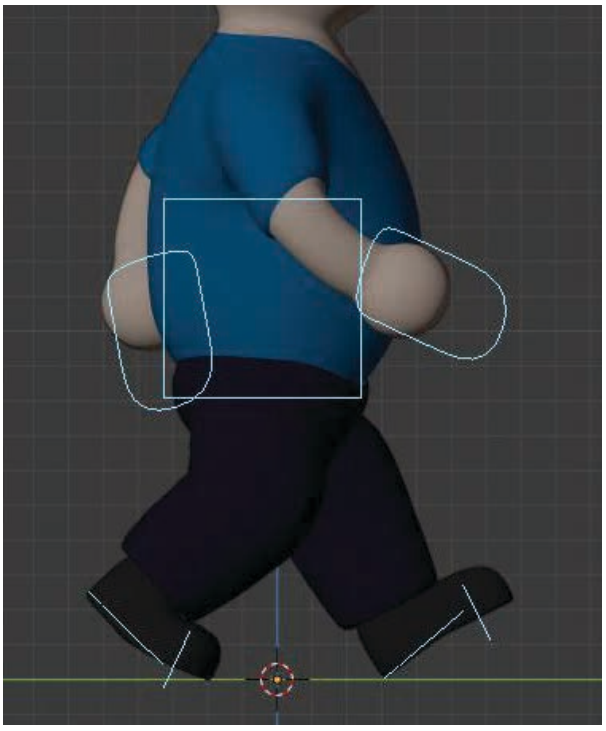


Contact Pose 1 on frame 1.

Contact Pose 2

With the bones still selected at frame 1, **press Ctrl+C**. Then go to frame 17, press **Ctrl+Shift+V** ("paste mirrored"). In case you don't have auto keyframing on, **press i** to record the keyframe. You'll notice that the character's pose has been mirrored.

Note: if you have the "Copy Attibutes" add-on installed, Ctrl+C gives you the Copy Attributes menu instead of just copying the pose. However, "copy pose" is the option all the way at the bottom of the Copy Attributes menu, so just choose that.



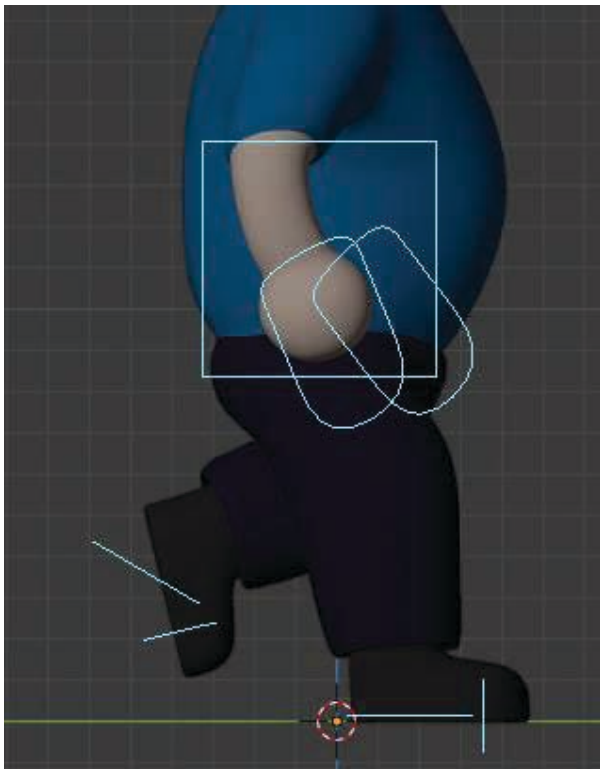
Contact Pose 2 on frame 17 - an exact mirror copy of Contact Pose 1 on frame 1.

If you scroll between frames 1 and 17, it already looks kind of like a walk cycle! But not quite dynamic enough, we need to have our character lift his feet up sometimes and move his body up and down to really sell the movement.

Pass pose

Set the *pass pose* on frame **9**. It's a bit higher than the previous pose and the character should have one foot flat on the ground. Set a keyframe.

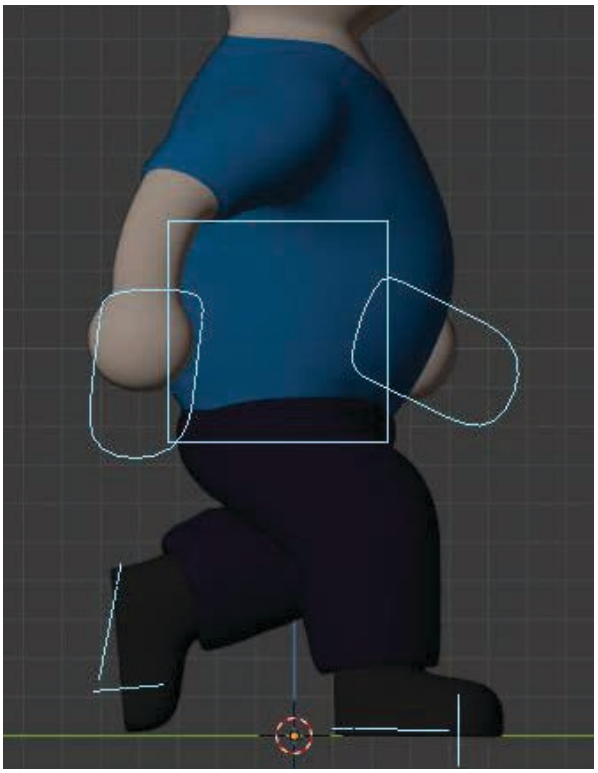
If it's confusing to you that the character has some influence from the previous keys, you can reset the bones to their default positions by selecting them all (**A**) and pressing **Alt+G** and **Alt+R.** That way you can create the pose from scratch.



The important bones for this pose are shown in blue. The rest are hidden. It helps to hide unnecessary bones while working - there are so many of them!

Down pose

Set the *down pose* on frame **5**. It's the pose where the character is the lowest in a walk cycle. Set a keyframe.



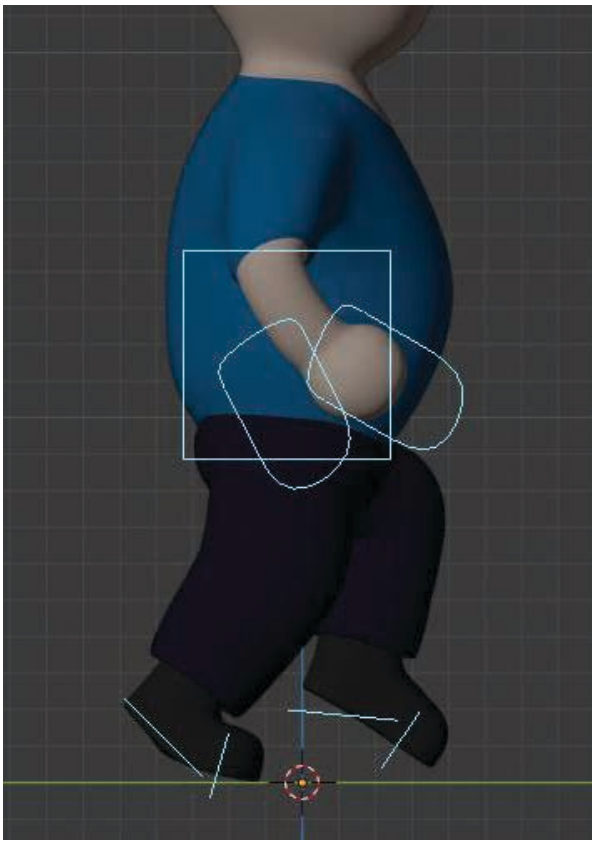
The Down Pose on frame 5.

Up pose

Set the *up pose* on frame **13**. It's the pose where the character is the highest in a walk cycle.

You may need to switch to right view (**press 3** on the numpad) to see if the arm on that side is not being stretched weirdly.

Go back to left view by **pressing Ctrl+3 on the numpad**.



The Up Pose on frame 13.

Second half

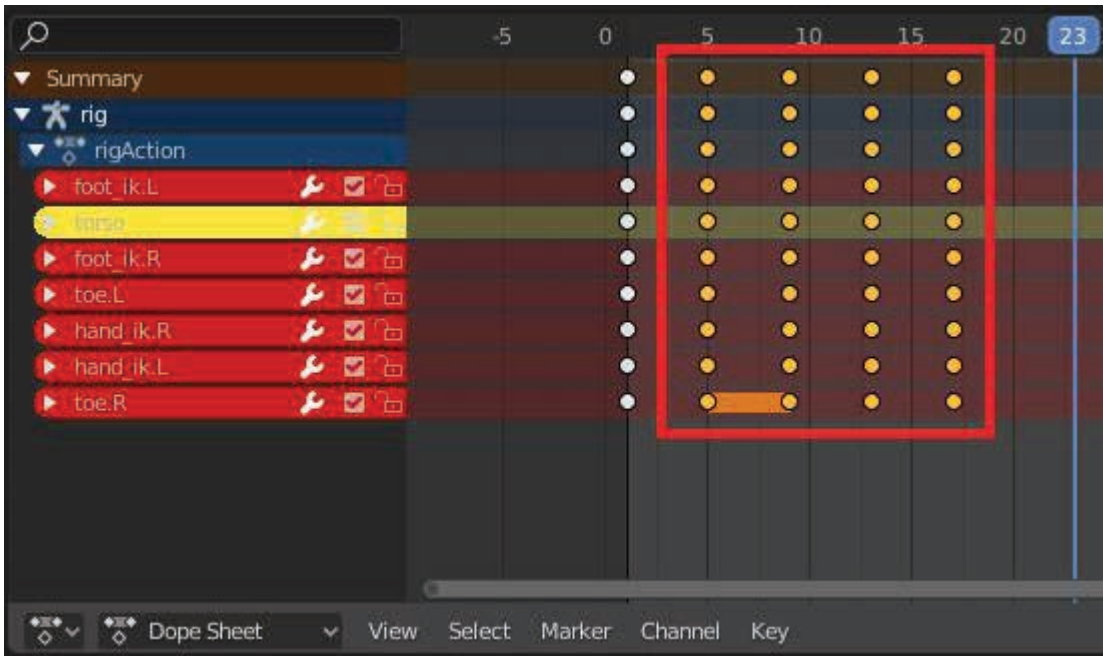
Now, you have the first half of your walk cycle - the second half will just be a mirrored version of this. But before we do that, let's make sure we are really happy with it so far. Move the timeline indicator back and forth and look at the movement of the character. Here it's very convenient to have "auto keyframing" turned on, because you can make quick, small changes to your bone positions and rotations, and you don't need to record the keys each time as it happens automatically.

Some things to double-check:

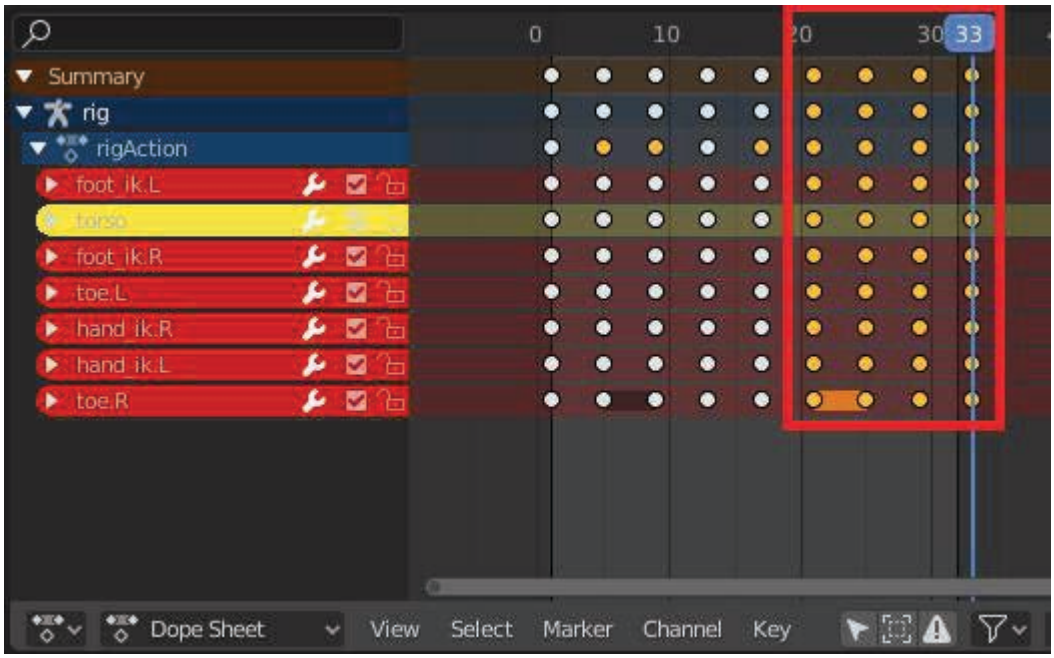
- Are the hands not moving too much (sometimes it can look like the character is punching the air while walking, when you exaggerate the hand movements too much).
- Do the feet touch the ground when they should?
- Is the up and down movement of the character not too strong? Or strong enough (depending on the style you want)?
- Do the hands rotate naturally when walking?
- Do the hands/arms not get stretched out too much when the character is in the Up pose? Make sure to check on both left and right views.

Duplicating and mirroring the keys

Once you are satisfied with this part of the walk cycle, duplicate all the keys except from those on frame 1 by box-selecting all those keys and **pressing Shift+D** in the dope sheet. Then move the duplicated keys over to the right. Note that the last keys should be on frame 33, so outside of our walk cycle on the timeline. That will make it loop perfectly.



*Select these keys (everything except the ones of the first frame) by dragging a selection around them. Duplicate them with **Shift+D** and move them to the right.*



The duplicated keys. Note that the last keys are outside of our Timeline range, on purpose.

If you plan the animation now it won't make any sense, because while we've duplicated the frames, we didn't mirror them yet. This we need to do for each frame individually.

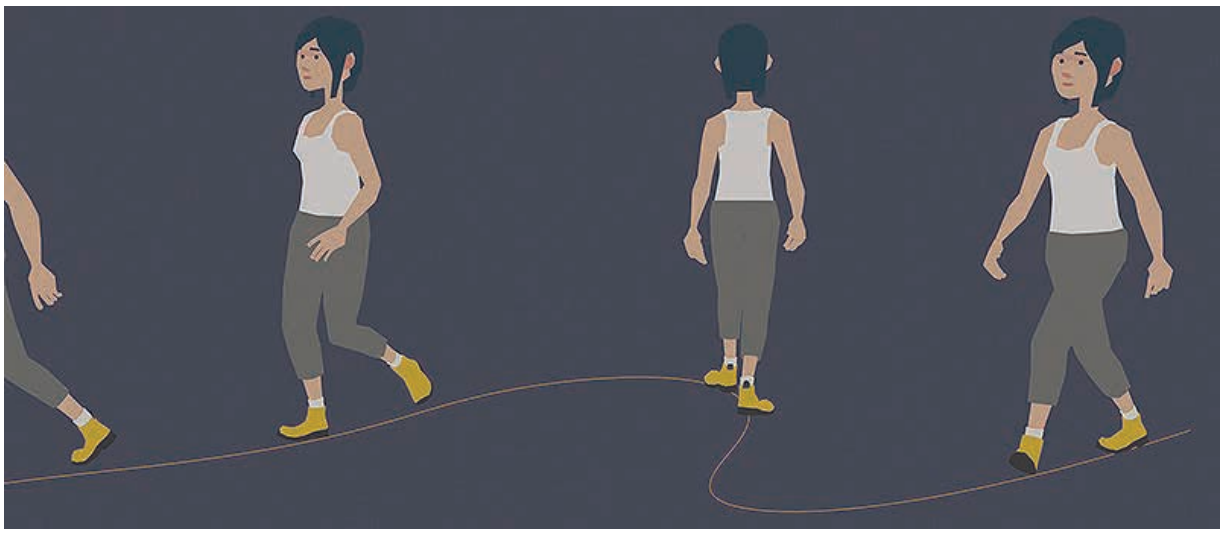
Go to frame 21, select all the bones, **press Ctrl+C**, then **press Ctrl+Shift+V** to paste the mirrored version of the copied pose at the same frame. Repeat the same step of copying and pasting mirrored poses for frames 25, 29 and 33. Now, when you play the animation it should be a perfectly looping walk cycle.

If it doesn't loop perfectly, make sure frame 1 and 33 are the same. If not, just copy the pose from frame 1 with **Ctrl+C** and past it with **Ctrl+V** on frame 33.

Tip: You can find many examples of walk cycles by searching on Google Images and Pinterest. It's a good idea to start by copying these, before you try to invent your own walk cycles.

Character Walking Along a Path

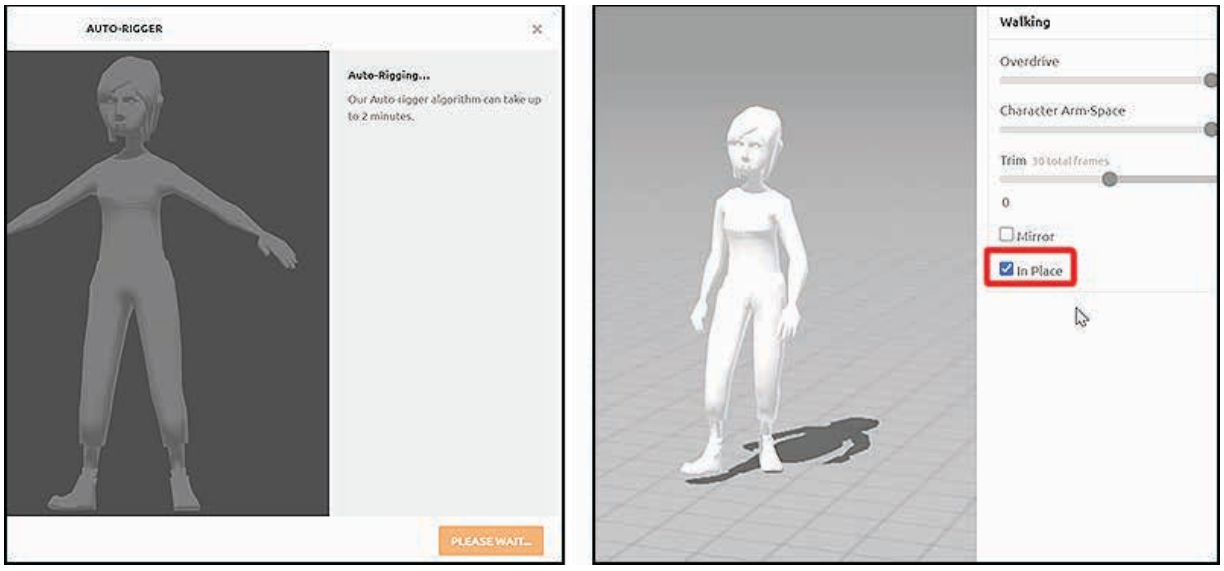
Getting a character to walk along a predetermined path is pretty easy!



Model by Sketchfab user Aki

First, it's important that your character walk cycle is animated to "walk in place". This means it's making walking motions but not going forward.

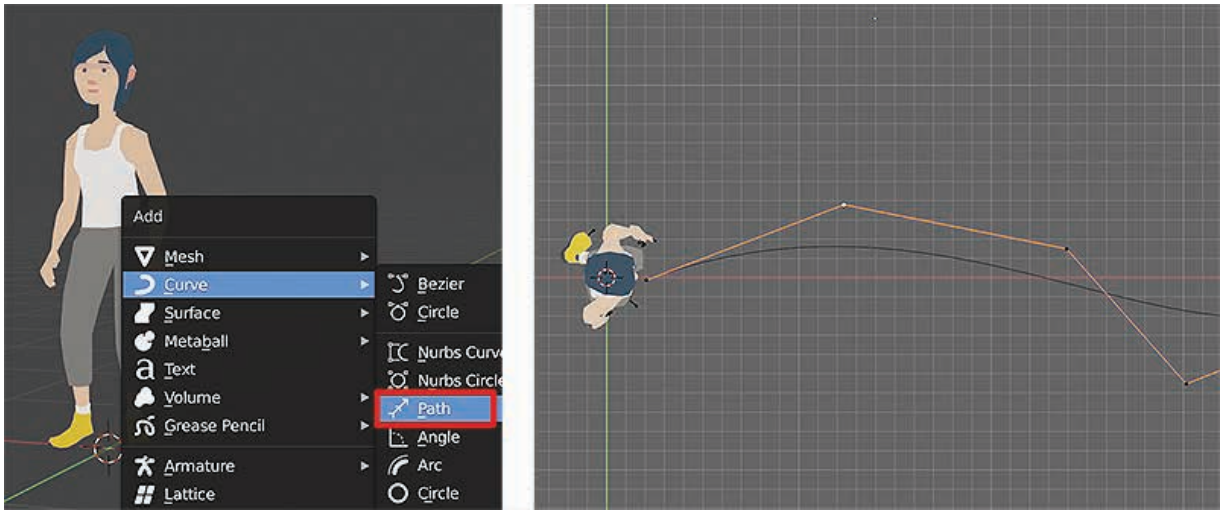
I rigged this particular character using Adobe Mixamo's free online "auto rigger", which took only a minute. I then found a suitable walk cycle and checked the **In Place** checkbox before downloading the animated model as an FBX file.



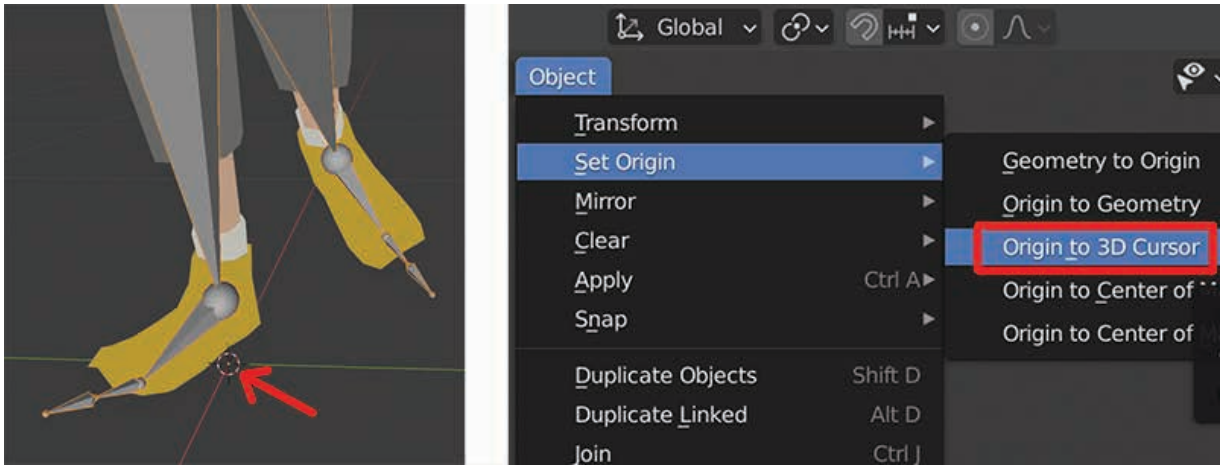
Auto Rigging the model in Mixamo

In Object Mode create a Path by **pressing Shift+A** and choosing **Curve > Path**. Edit the shape of the Path in Edit Mode by dragging the points that make up the Path.

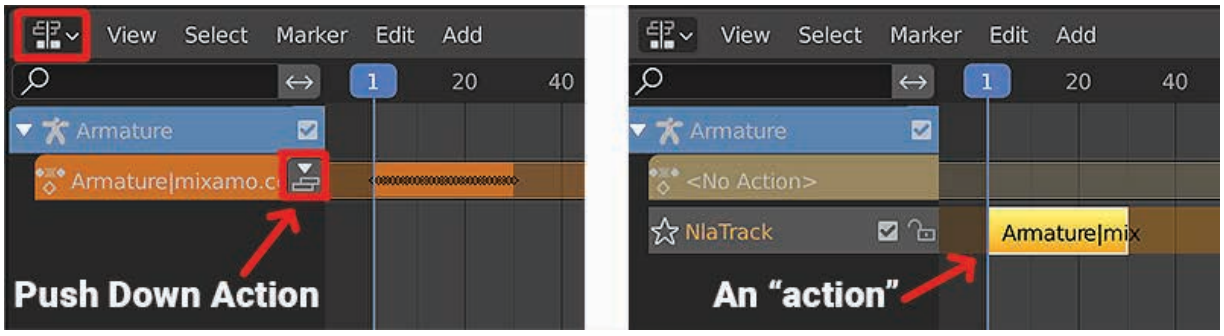
It's easy to do this from the Top view (**shortcut: 7 on the numpad**). Make sure it starts from where your character is standing.



By the way, make sure that your **Armature** has its pivot point where it should be (at the ground, between both feet). If that's not the case, **press Shift+S** and choose **Cursor to World Origin**, then from the Object drop-down menu choose **Origin to 3D Cursor**.



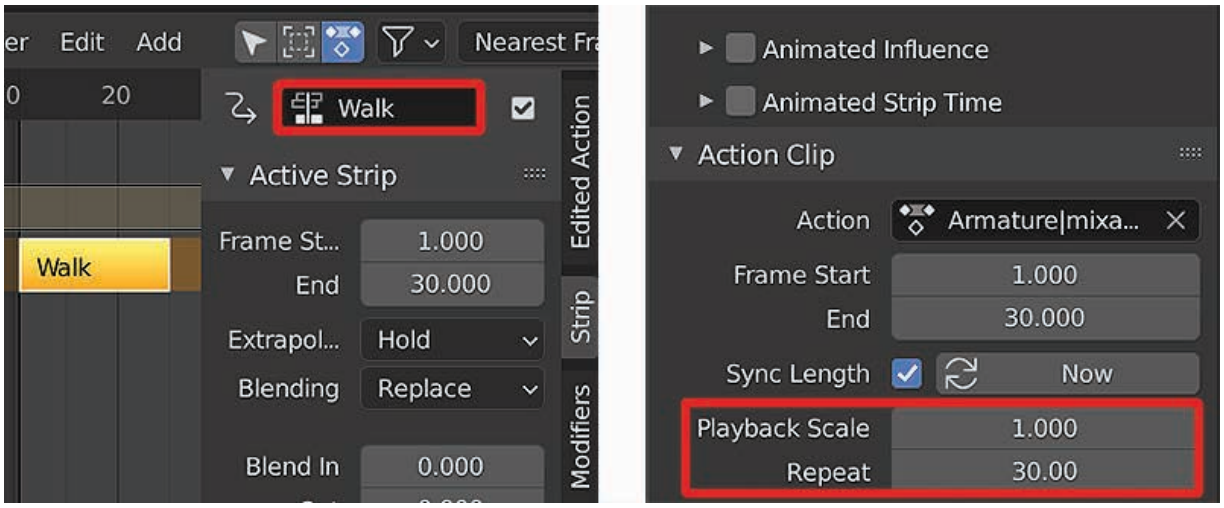
Open a **Nonlinear Animation** window. You'll see the keys of the walk cycle of your selected Armature there. Turn your keys into an Action by **clicking on the Push Down Action icon**.



Press N to open the Option Panel and give a clear name to your Action like “Walk”.

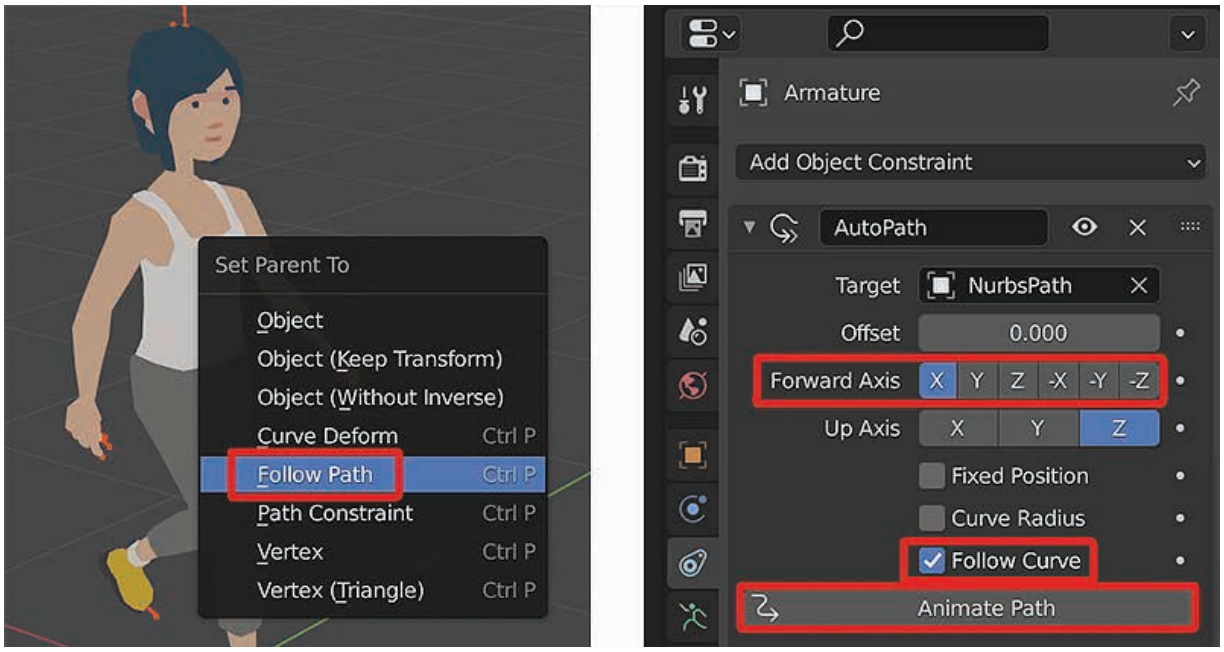
You can change the speed of the walk with the **Scale** value. For example a value of 10 will make it look like the character is walking in slow motion (maybe add a cool explosion behind them!) while a value of 0.1 will make them go really fast.

So the trick here is to be subtle - **hold Shift** while dragging the value to change it in smaller increments. The **Repeat** value repeats the Action, so increase it for the character to keep walking.



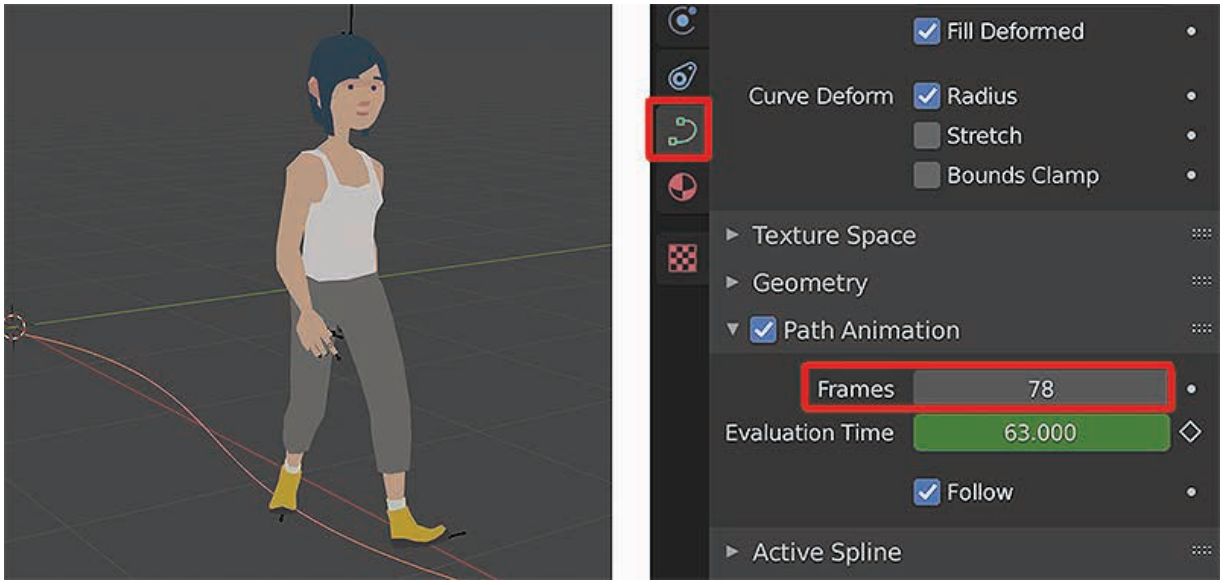
Holding Shift, select the Armature and then the Path. **Press Ctrl+P** and choose **Follow Path**. In the Armature’s Object Constraints tab, you’ll see the AutoPath constraint has been added.

Check **Follow Curve** and then click on **Animate Path**. You may need to choose a different **Forward** direction in your case (I had to change it to “X”) in order for the character to face the correct way.



Now if you **press the Spacebar**, you should see the character walk along the Path. You'll probably have a bit of "sliding" of the feet, which is easy to fix by changing the Curve's Path Animation **Frames value**.

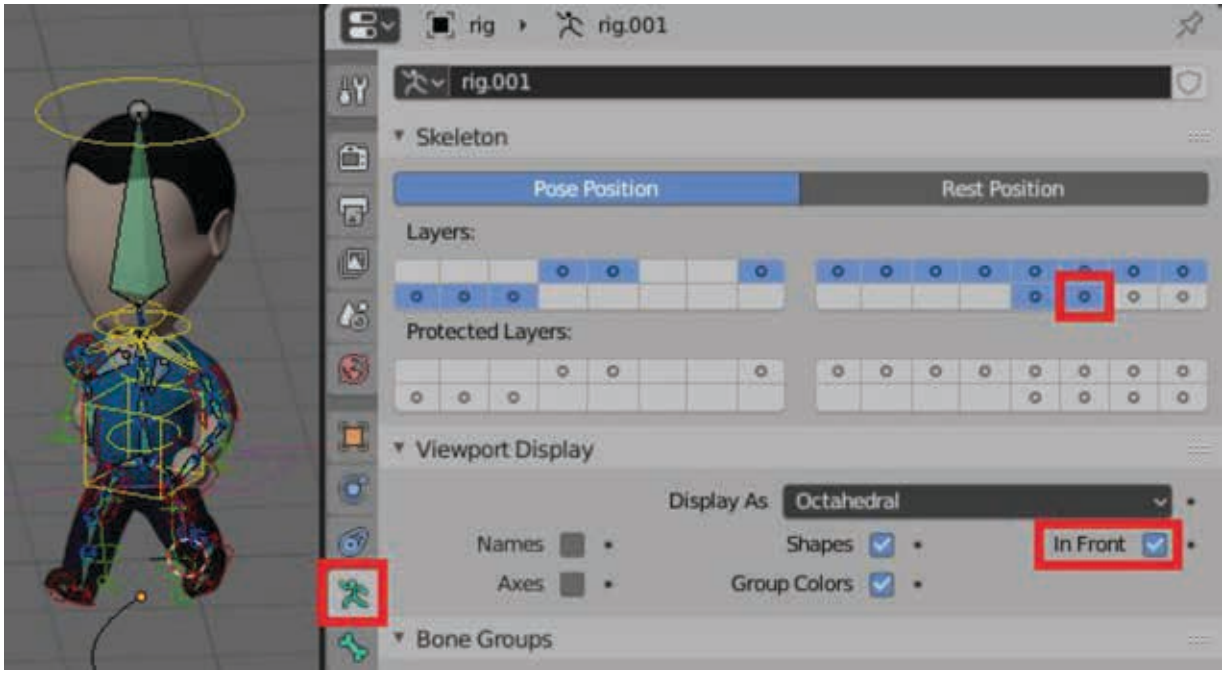
Change it in 10 frame increments at first, then in 1 frame increments until you can no longer detect any foot sliding. The specific value depends on the length of your Path as well as the Scale and specific animation of your walk cycle.



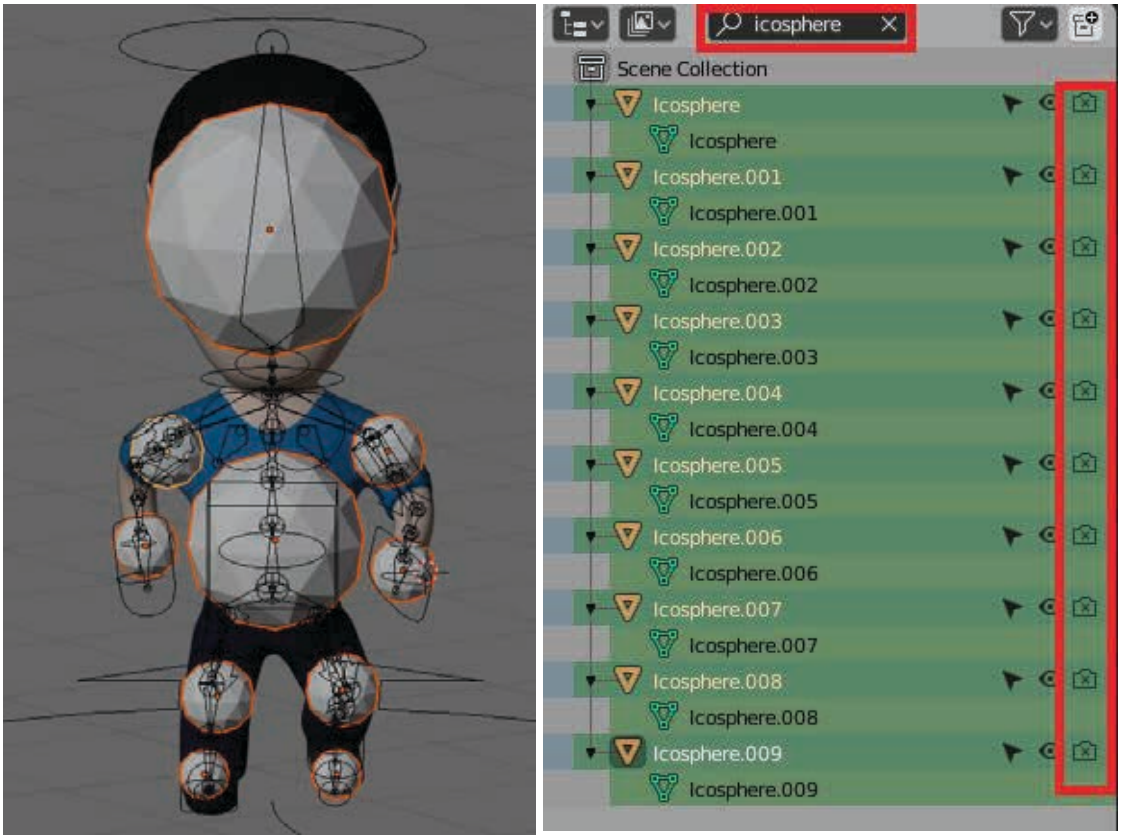
How To Have Characters Interact With Physics Simulations

Having a character interact with Rigid Body objects can have pretty interesting results! Normally a rigged character, armature or bones won't interact in a physics simulation. The character will just walk through a wall like a ghost. So in this tip we'll look at setting up the collision detection.

Unhide the armature layer which has the deformation bones on it, by holding Shift and clicking on it (the icon with the red border around it pictured below) Set it to "In Front" so we can see the Armature through the body.

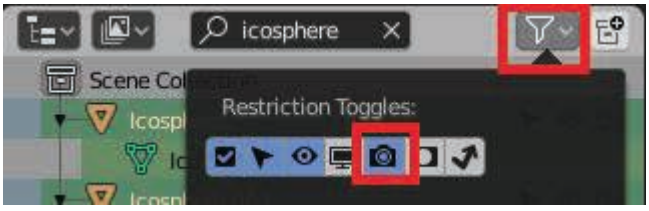


Now we are going to parent some simple collision objects to Bones of the Armature. For this we'll use simple Icospheres. You can also use other primitives, or model basic approximations of the body parts around those bones. In this case we'll keep it simple and use Icospheres everywhere.



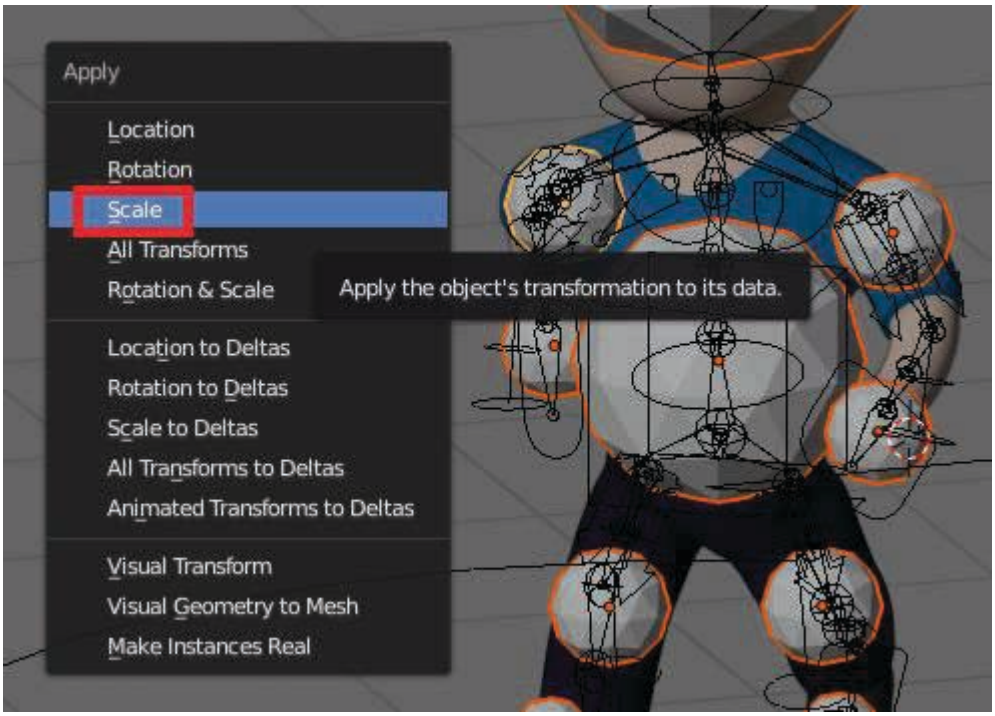
Left: Icospheres placed in key points of the character. Right: the Icospheres have their render visibility (the camera icon) turned off.

Make sure the Icospheres don't render by turning off the camera icons in the Outliner. You can search for "icosphere" and it will only list Icospheres. If you don't see the camera icon, turn it on in the Filter.

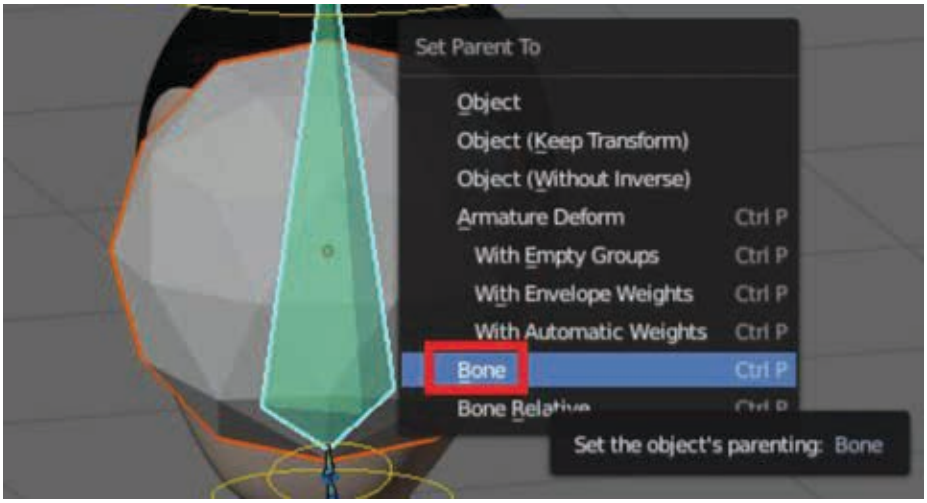


The Icospheres are also set to "In Front" in their Object > Viewport Display settings. This makes it easier for use to place them inside the character.

Before parenting the icospheres to the bones, make sure you've applied their scale in case you've scaled them up or down (which you probably have!). To apply the scale, **press Ctrl+A** and choose Scale.



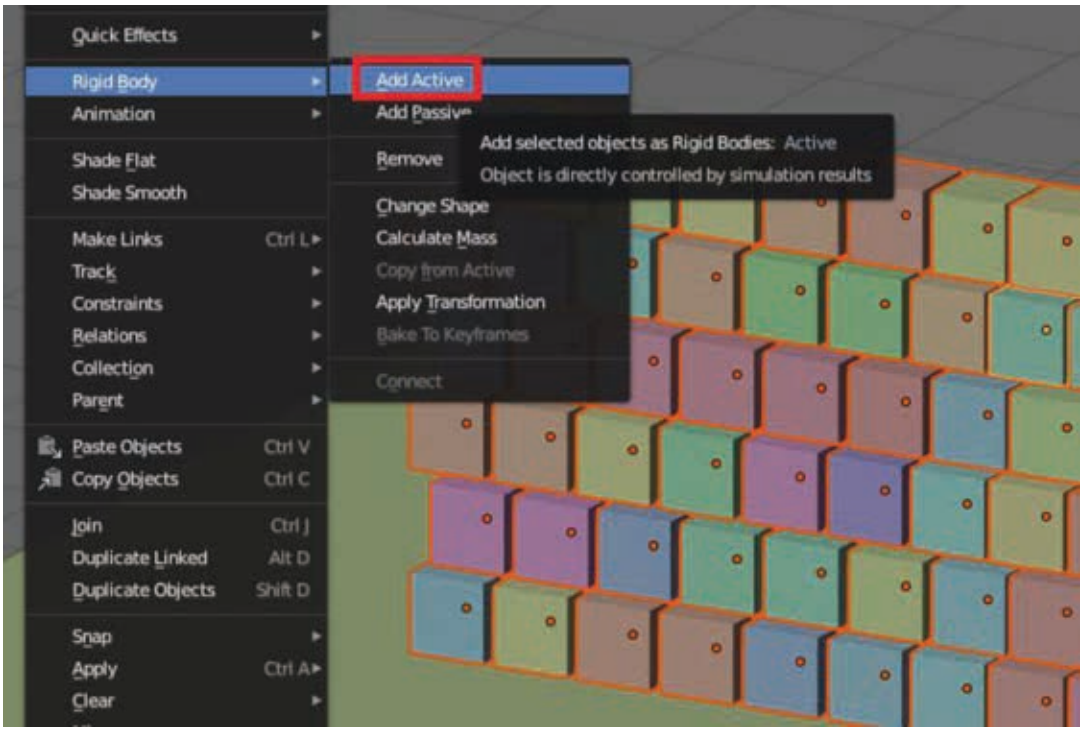
Then hold Shift while you select first the Icosphere and then the Bone (the Armature should be in Pose Mode before you select the Icosphere, otherwise you won't be able to select individual bones). **Press Ctrl+P** and choose the option "Bone".



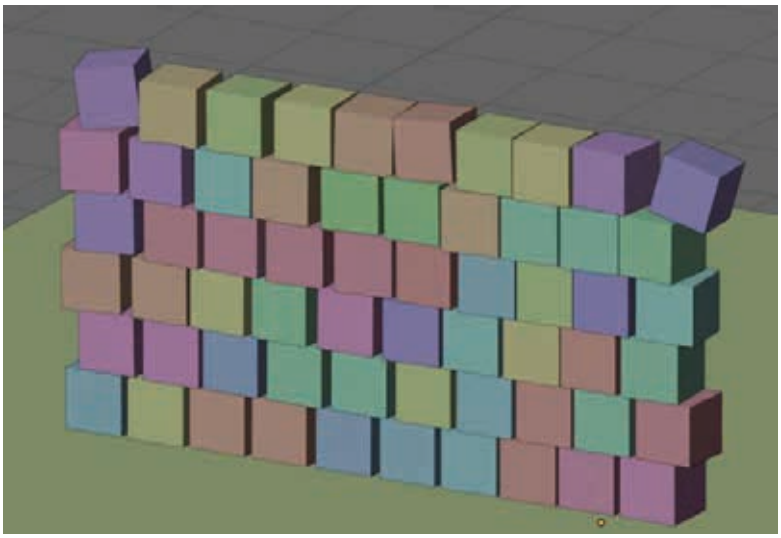
Note that this is the same way we parented the hat to the character earlier in this book. Except this time, the parented object will not be rendered.

Do the same for the parts of the body that will interact the most with the physics simulation of the wall (hands and feet for example). Afterwards we can hide the Armature. Play the animation to see if the Icospheres follow the character. If not, check that they are parented to their respective bones.

Create a Plane to serve as the floor, and make a little wall of Cubes. Make sure the Scale is applied after you've moved the Cubes into place and set them to Active Rigid Bodies.

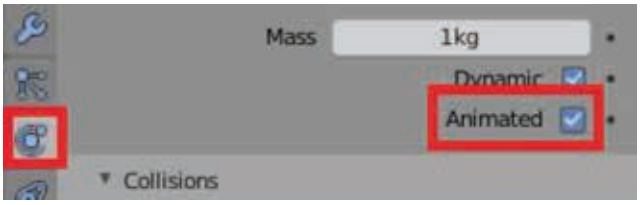


Now when you play the Timeline by **pressing the Spacebar**, the Cubes will fall into place (if you've left some space in between). If they fall through the floor, make sure you've set the floor Plane to be a Passive Rigid Body.



The wall cubes at the start of the animation.

Next, hide the character mesh and select all the Icospheres, make them all Active Rigid Body objects, same as we did with the wall Cubes. After this, select the Icospheres one by one and turn on "Animated" in their Rigid Body settings.

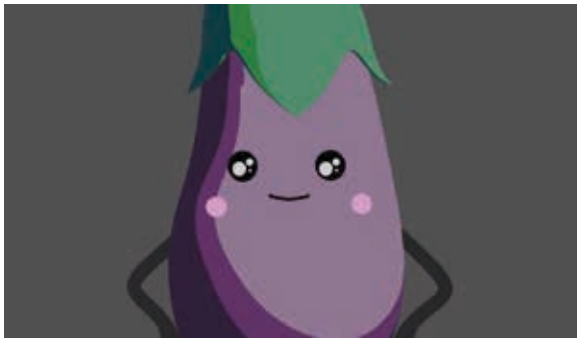


This is necessary because they are moving, not just from a physics simulation but because of keyframed animation. If we don't turn this on, they will just fall down like the Cubes in the start of the animation. (In fact, turn off the "Animated" setting for the head Icosphere and play the animation from the first frame, just for a quick laugh).

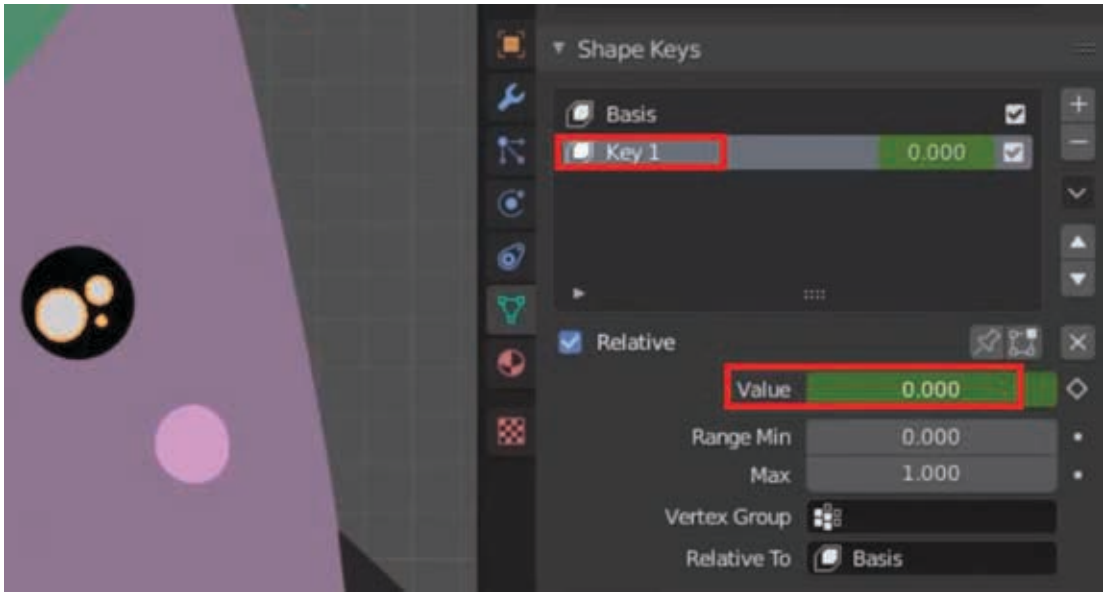
Now your character will crash through the wall when the animation is played!

Shape keys for facial expressions

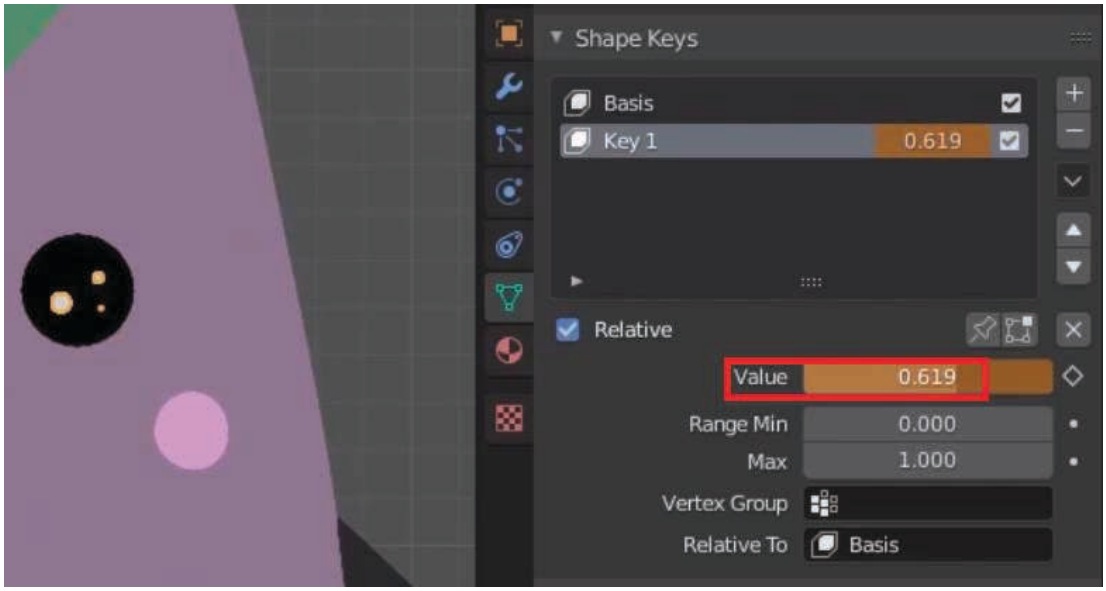
This is another way to do NPR style facial expressions.



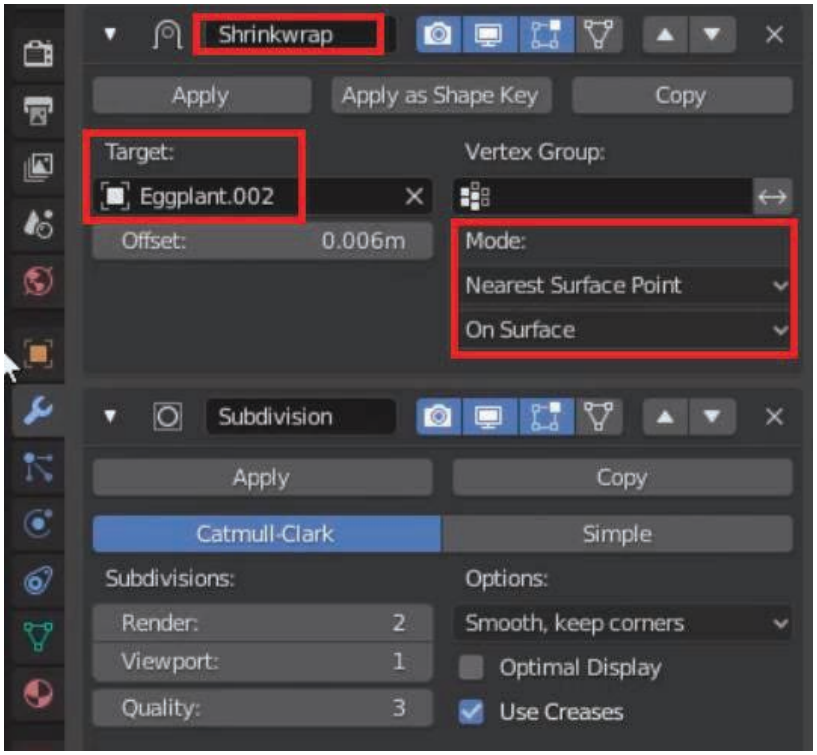
The face bits are meshes controlled with shape keys.



The “eye glimmer” object is a separate object with its own Shape Keys. When the value of Key 1 is 0, it has its default shape. When the value of Key 1 increases, the object becomes smaller.



A Shrinkwrap modifier makes sure they stay on the surface.

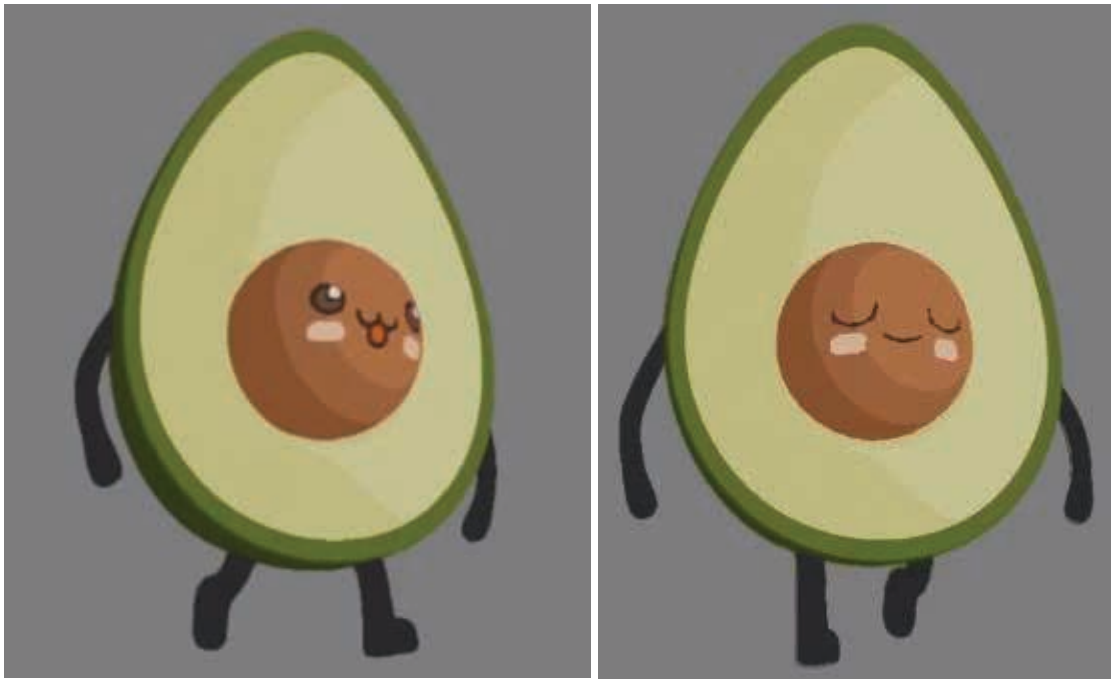


All the parts of the face: eyes, mouth, eye glimmer and even the pink cheeks are separate objects that have a Shape Key as well as a Shrinkwrap modifier to keep them on the surface of the eggplant body.

They are all parented to the main body mesh.

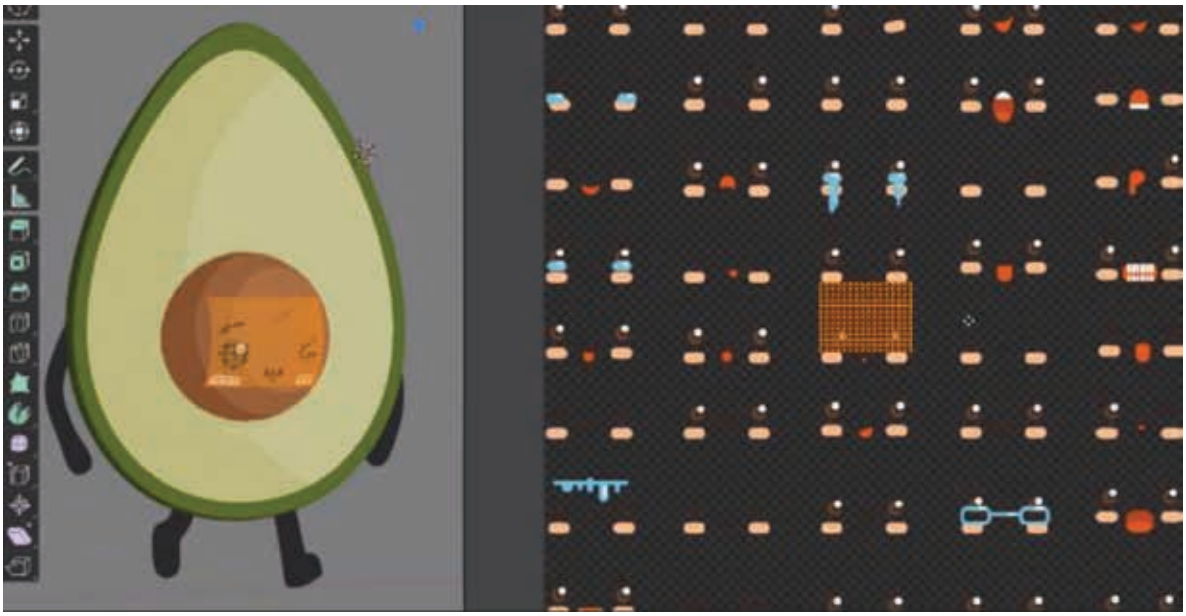
Facial Expressions

This tip looks at one way of doing facial expressions that suits low-poly or simple stylized characters.

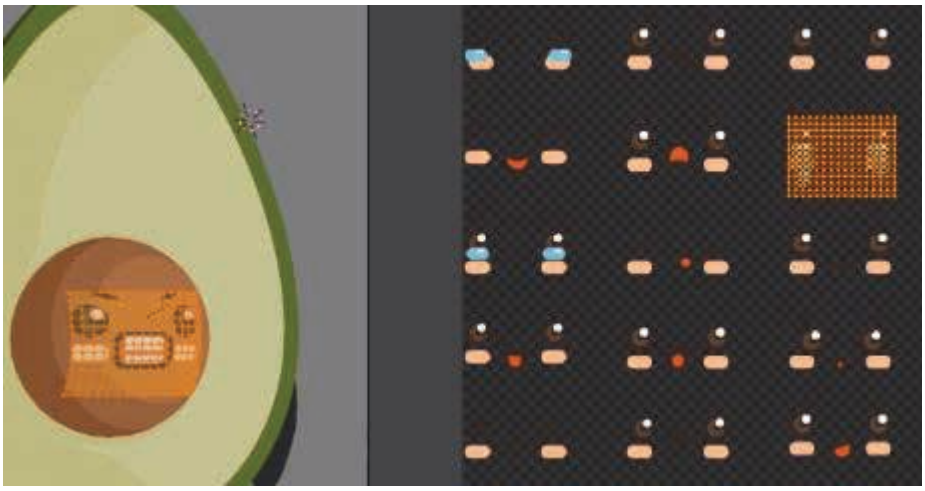


This is a single texture that contains all the necessary facial expressions for this character.

The UV Map is animated to switch between the faces.



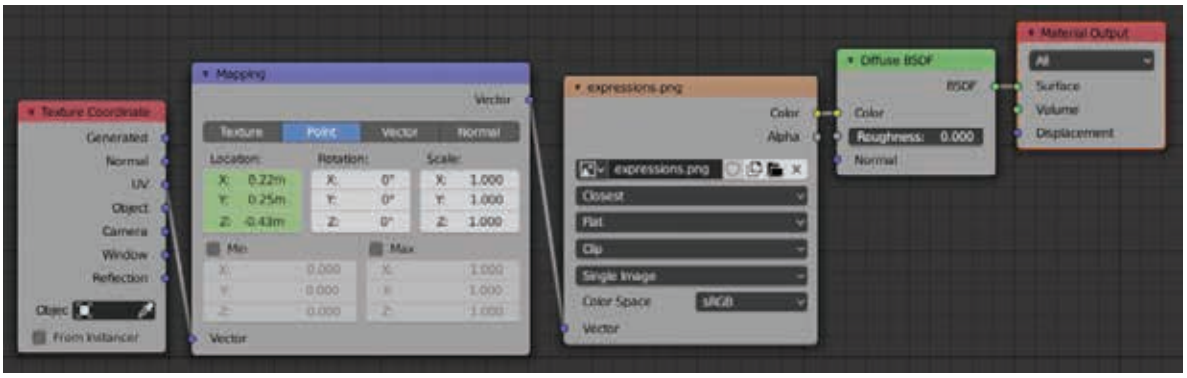
The face is a plane that is placed in front of the head, but you could also uv-unwrap the front section of the head.



Moving the UV island over the texture reveals the different expressions.

For the Snader node set-up I use the standard Mapping and Texture nodes that control the UV-map of the image texture. The plane that has the face texture is UV-unwrapped so we can use the UV-output socket of the Texture Coordinate node. Then the “Location” values of the Mapping node control the placement of the UV-island on the texture. This is what we need to keyframe-animate (from one frame to the next, so we don’t see the texture slide).

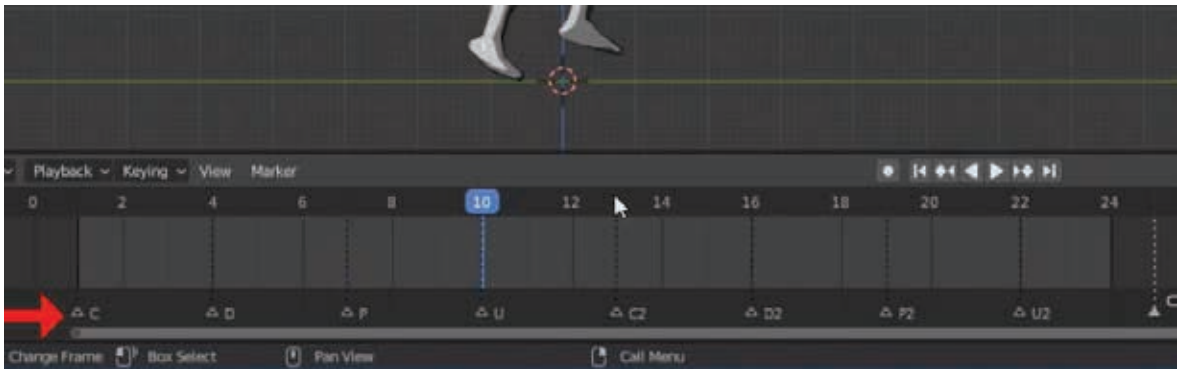
I’m also using a transparent BDSF Node and an Emission shader in the above example, because the face expressions texture has a transparency channel and I didn’t want to have any shading. However, this node set-up would also work:



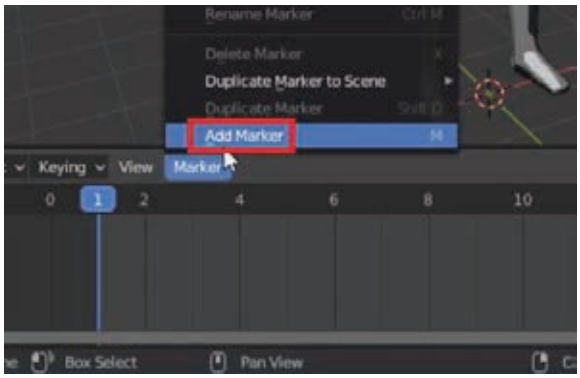
Walk cycle markers

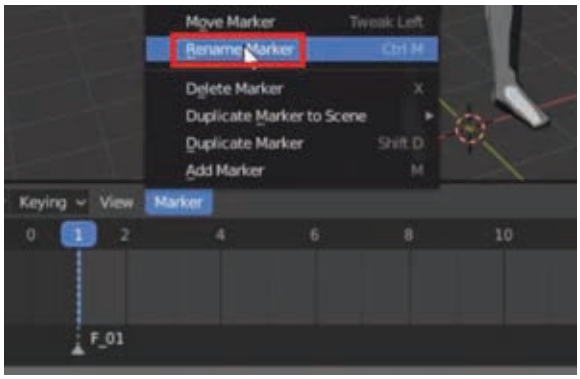
Before animating a walk cycle it's smart to place markers for each key pose.

Contact, Down, Pass, Up... just call them C, D, P, U, C2, D2, P2, U2 and finally C again.

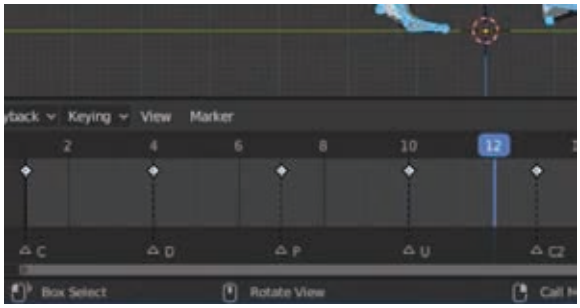


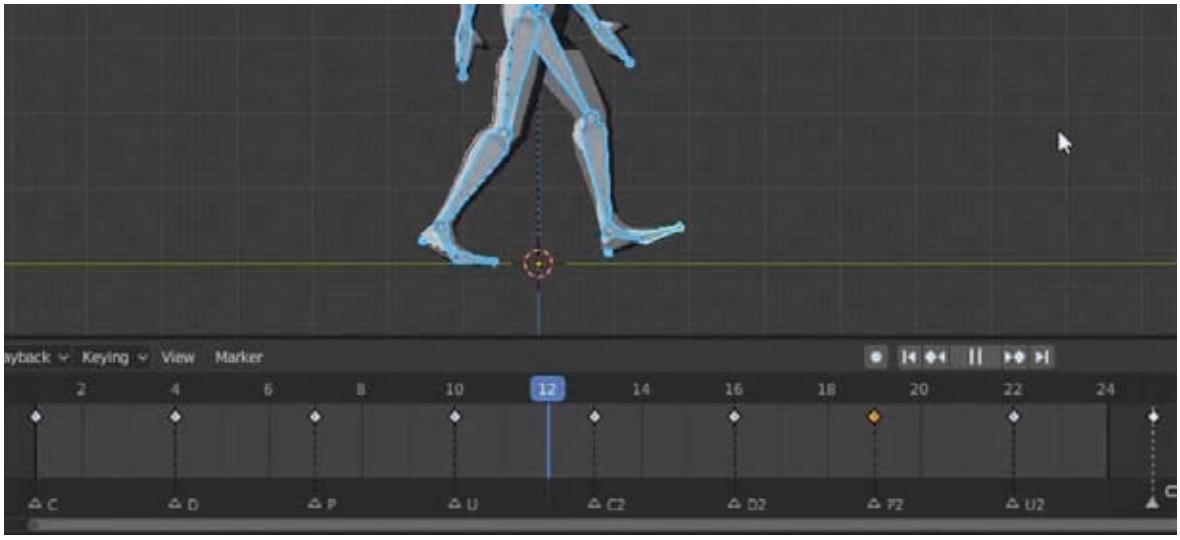
In the Timeline you can add and rename markers from the Marker menu.





Name them **C, D, P, U, C2, D2, P2, U2 and againC**. The last contact pose is identical to the first one.
I use 24 frames for a walk cycle, with 3 frames between each pose.

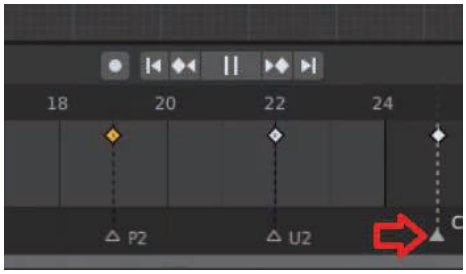




Contact, Down, Pass, Up and then the same again but mirrored.

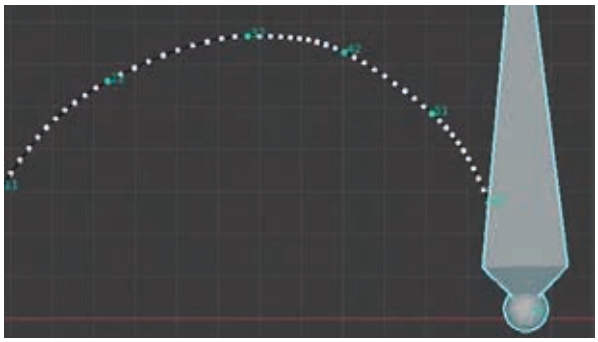
Thanks to the markers you can see easily where to make your key poses.

I repeat the first Contact Pose on frame 25 so that it loops perfectly.

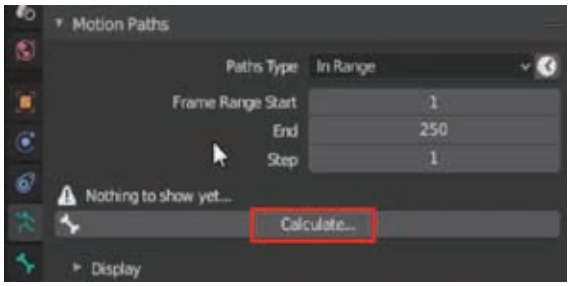


Motion Paths

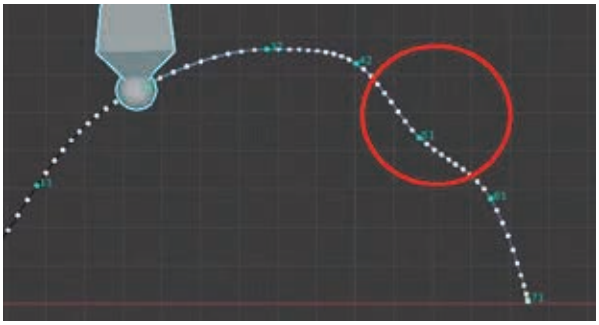
Motion Paths are a way to visualize your animation with a path.



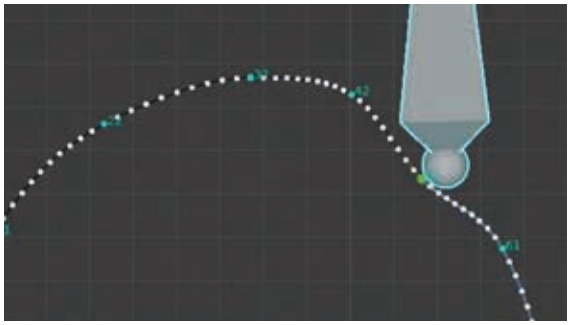
In Pose Mode select the bone you want to see a Motion Path for and press **Calculate**.



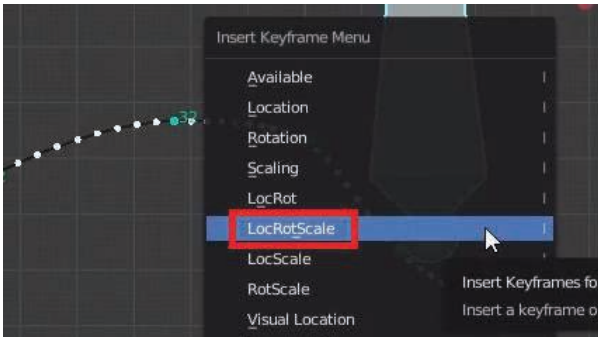
Now you can easily see parts of your animation that maybe need some work.



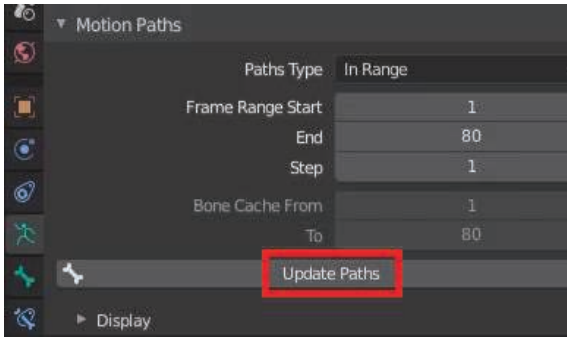
Adjust your animation by posing the bone in **Pose Mode.**



and record the keyframe by **pressing i** and choosing **LocRotScale**.



Press **Update Paths**.

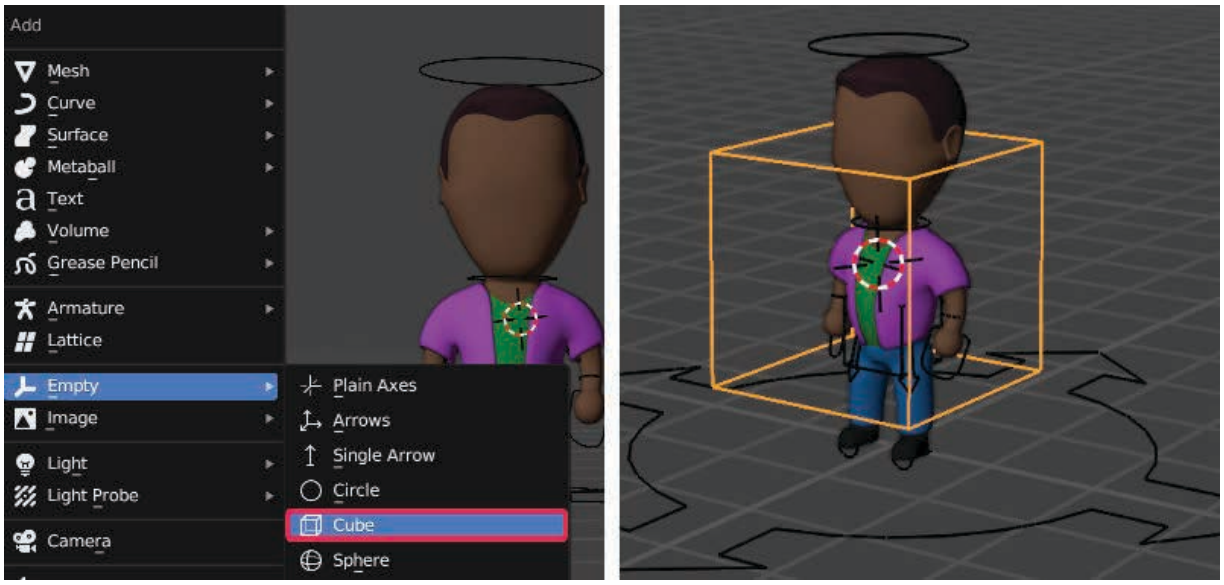


Click here to return to the index.

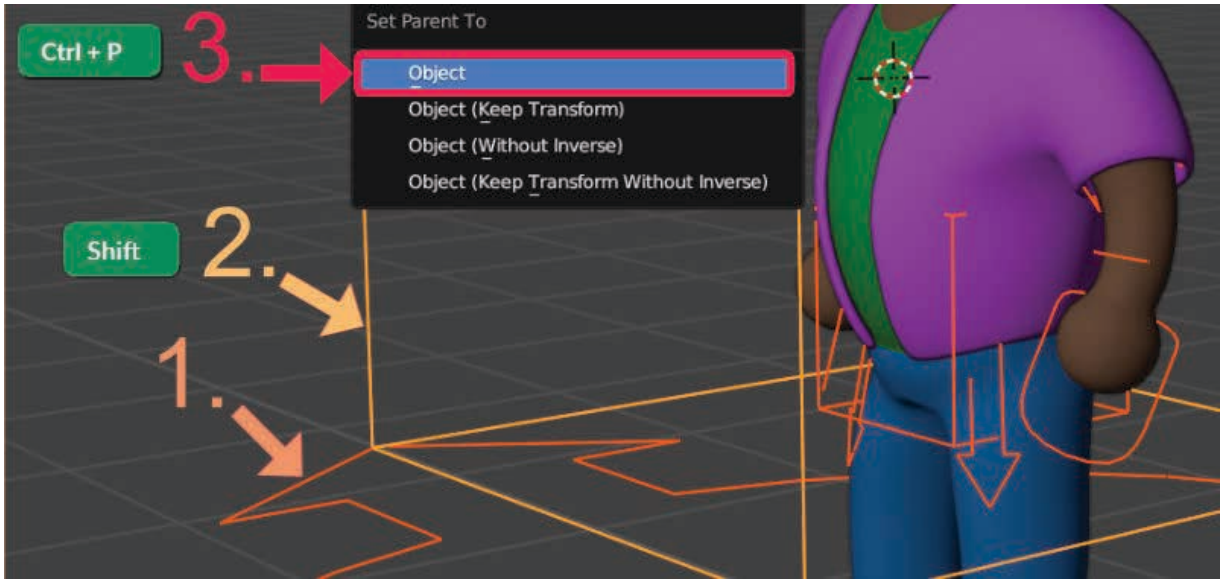
Rotating your character

To rotate a character it's much easier to parent the rig to an **Empty** and rotate the Empty. It's easier to choose the center of rotation and avoids rotation issues with the rig.

In this case I'm using a box-shaped Empty. To add an Empty to the scene, **press Shift+A** and choose an Empty.



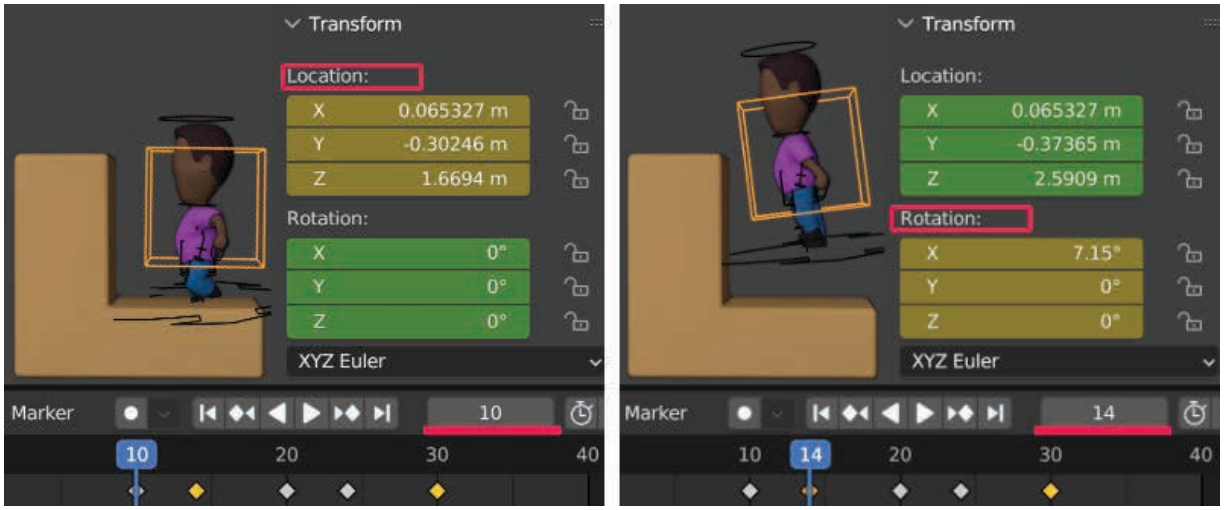
Select your Armature and **hold Shift**, then select the Empty. Then **press Ctrl+P** and choose Object from the Parent menu.



Set separate Location and Rotation keys for the Empty.

It's a good idea, to set the Rotation value manually in the Rotation field, then keyframe it by **pressing i** while hovering the mouse cursor over that field.

The Rotation and Location keys are out of sync to make the jump more interesting.

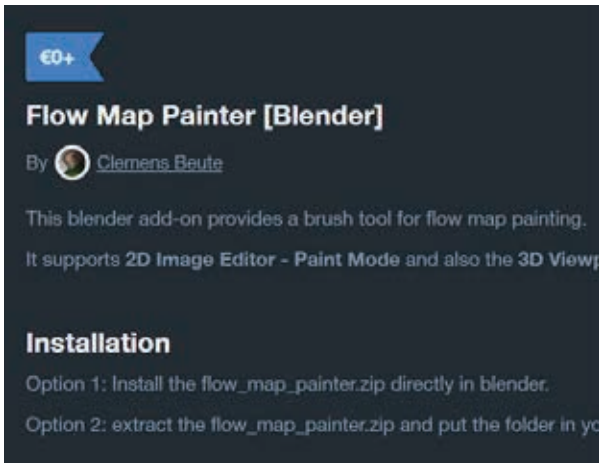


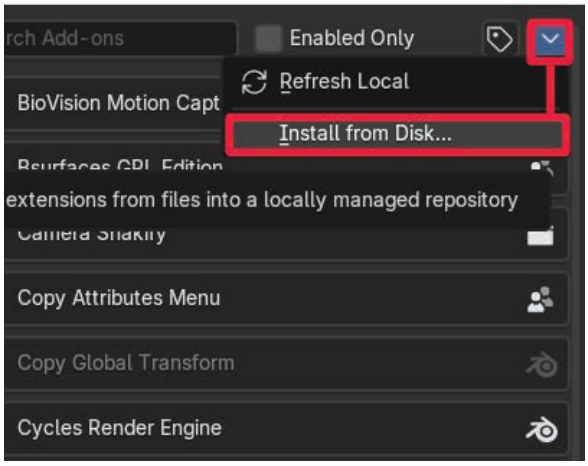
NATURE ANIMATION

Flowing water with Flow Maps

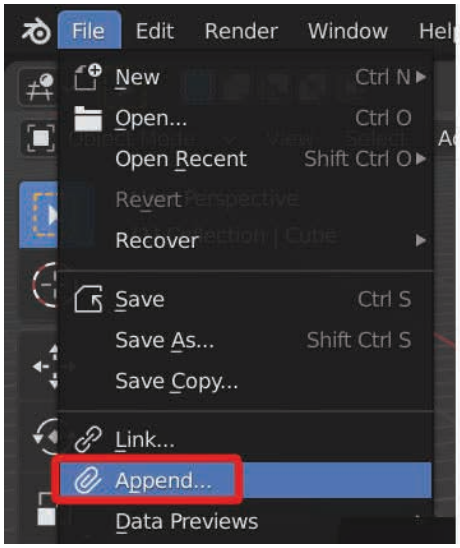
Flow map is an addon created by Clemens Beute for flow map painting. Flow maps can give a direction and movement to a texture.

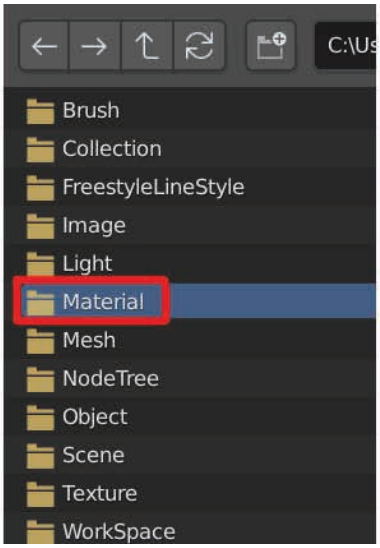
To get started, download the ZIP file from Gumroad and install it in Preferences.

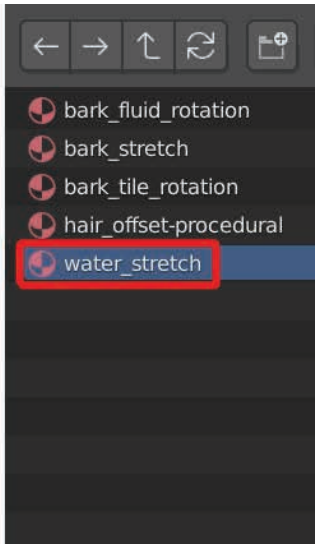




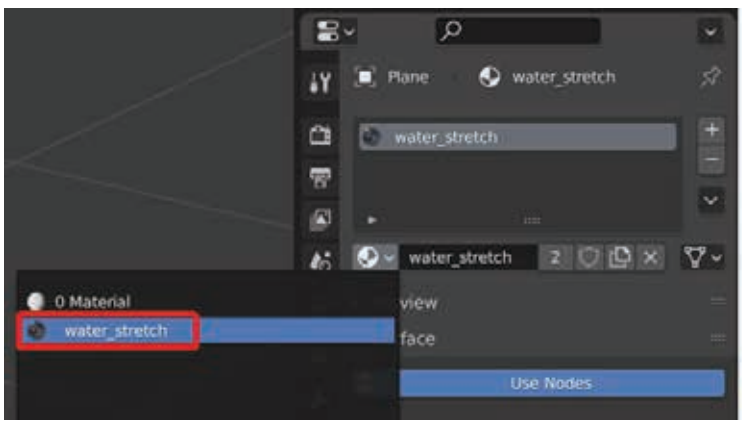
If you just want to use it for flowing water, you can simply append the water_stretch material from the example Blend file that comes with the add-on.

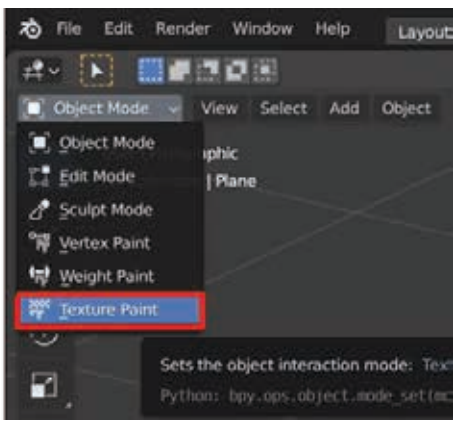






Add the material to your UV-unwrapped object by selecting it from the material drop-down menu of your object. Then, you need to be in Texture Paint mode.





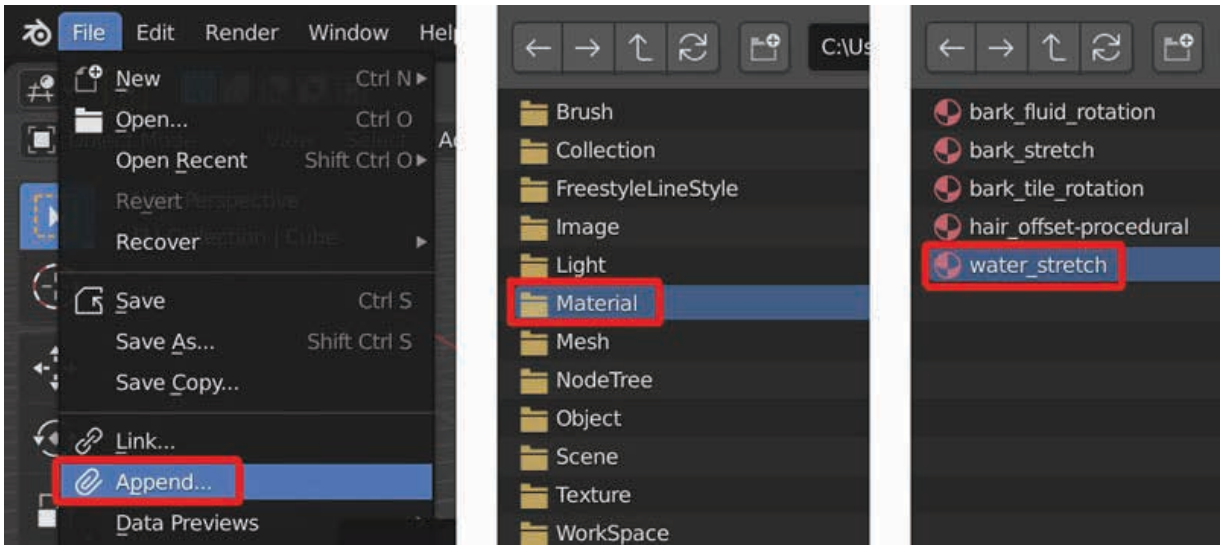
Open the Option Panel by **pressing N** and find the add-on in the **Tool panel**. Then, simply click on the Flowmap 3D Paint Mode button and start painting.

Press Spacebar to see the flowing movement. You can even keep painting while it's playing - in fact, I recommend that you work this way.

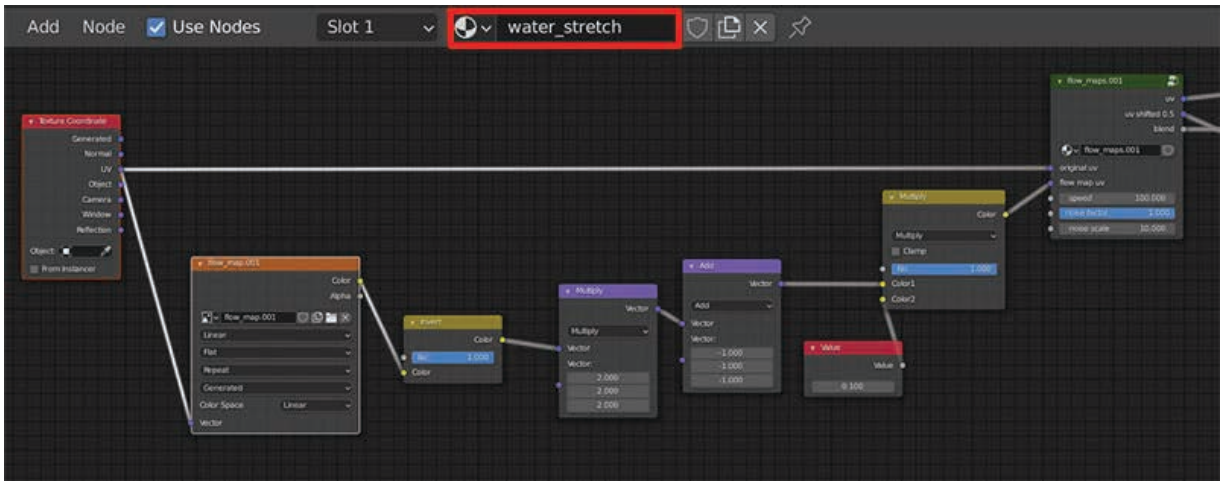
Animating clouds with Flow Maps

Previously we looked at using the example water material that comes with the Flow Map add-on. But what if you want to use your own textures, for example to create flowing gas clouds on Jupiter, or a moving cloudy sky?

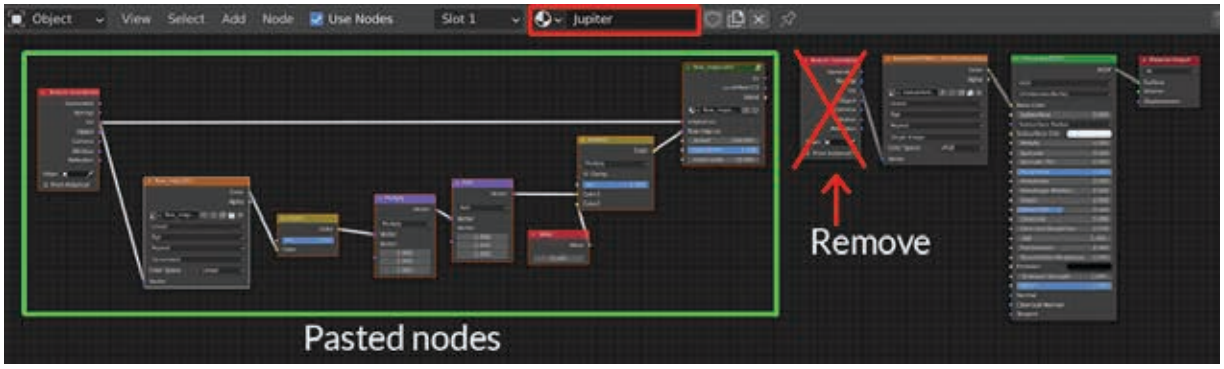
First, let's look at the planet surface. This is a UV Sphere (set to Shade Smooth) with a texture from NASA's website. Append the **water_stretch** material from the example file that comes with the Flow Map add-on.



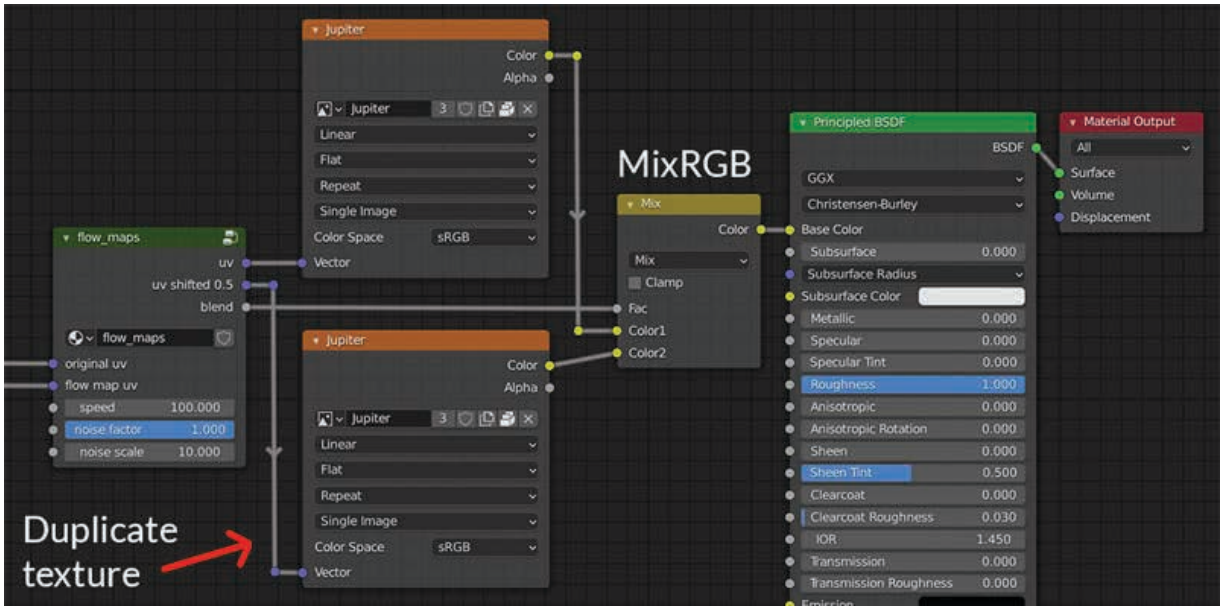
In the Shader Editor select the **water_stretch** material from the drop-down menu. Then select and copy these first 8 nodes with **Ctrl+C**.



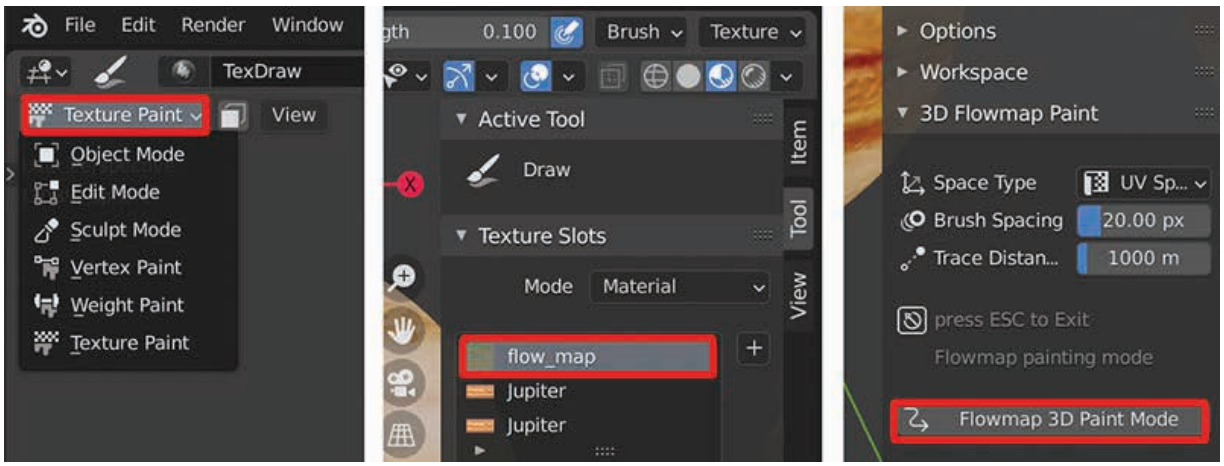
Switch back to the planet material and paste the copied node in the material of your object with **Ctrl+V**. You can remove the Texture Coordinates node from the planet material as the pasted nodes also contain this node.



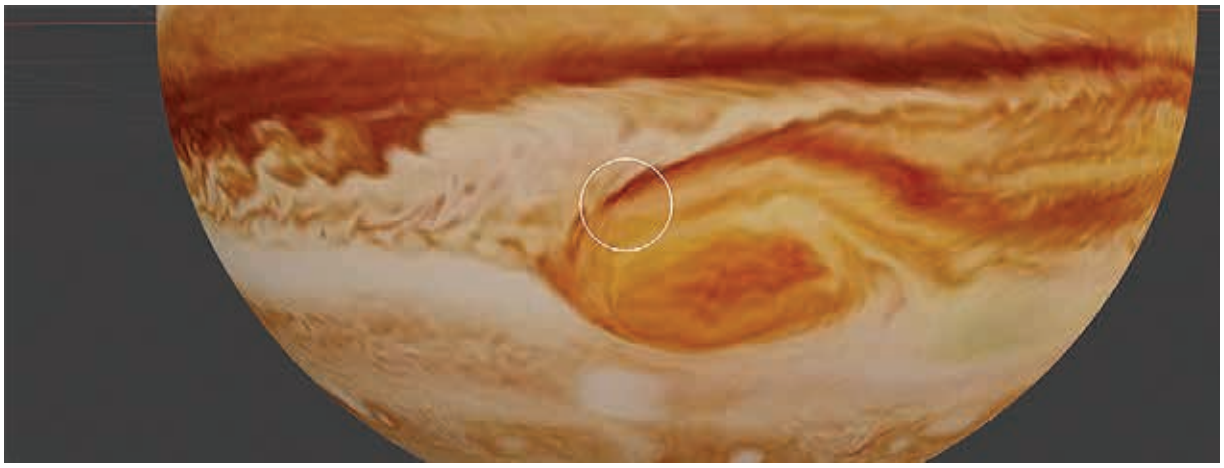
Duplicate your texture so that you have two copies, and combine them with a **MixRGB node**. As the blend factor for the MixRGB node, we use the Flow Map nodes.



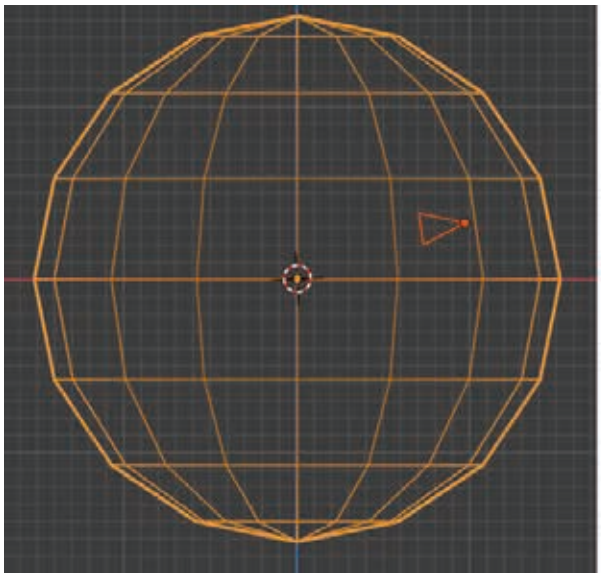
Then make sure you are in **Texture Paint mode** and that the **Flow Map** is selected. Click on **Flowmap 3D Paint Mode** and start painting.

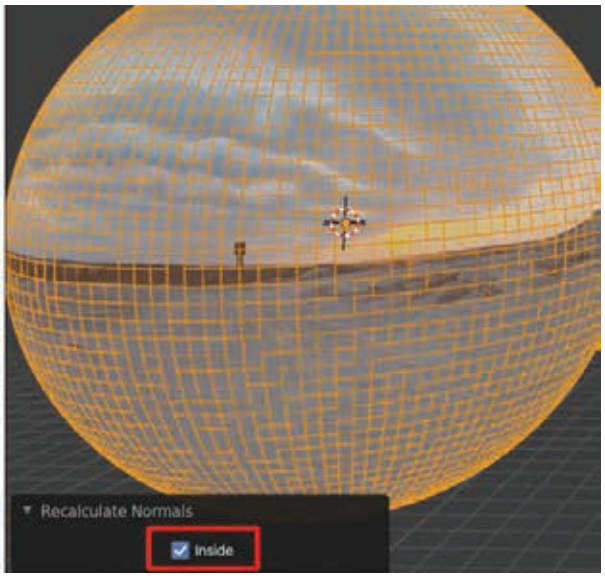


Press the Spacebar to see the flowing motion. This way you can see what the flow map is doing as you paint.

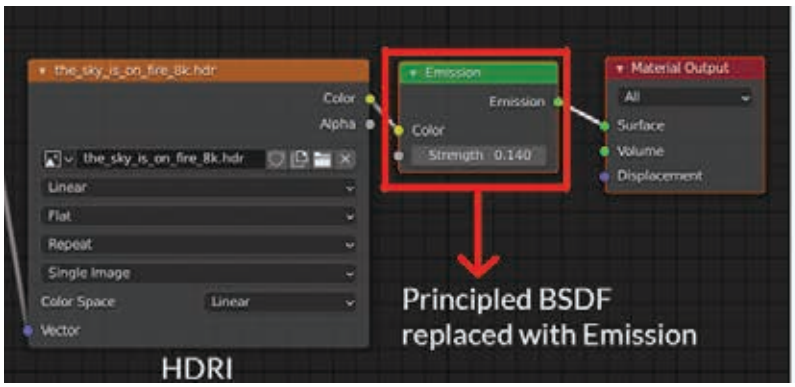


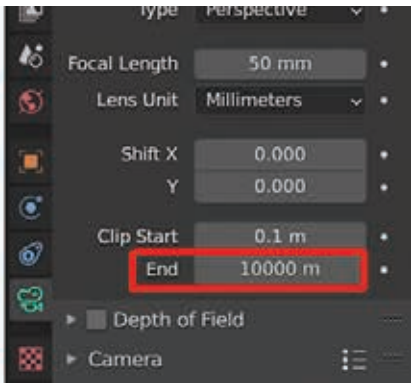
To animate clouds this way, is almost the same workflow as the planet. With a few notable differences: The camera is inside the sphere instead of on the outside. Because of this, it makes sense to increase the scale of the sphere. As you're going to paint the inside of the sphere, you need to flip the normals. To do this, select all in Edit Mode and **press Shift+N**, then check "inside".



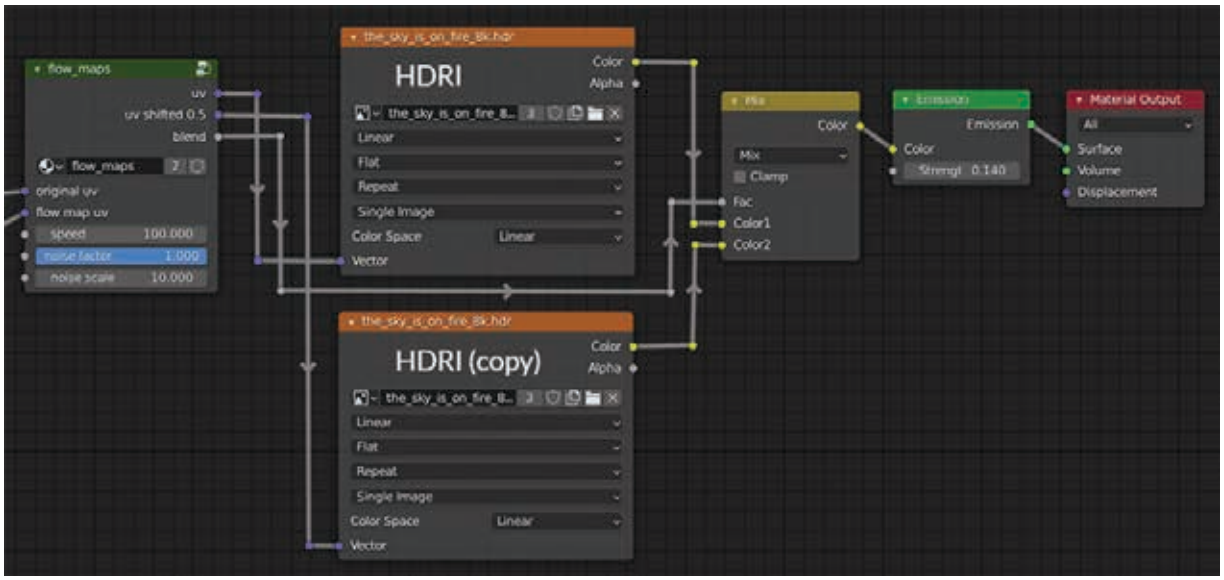


You may have to increase the **Clip End** value of the camera to see the sphere if it's really big. And you have to change the Principled BSDF to an **Emission** node.



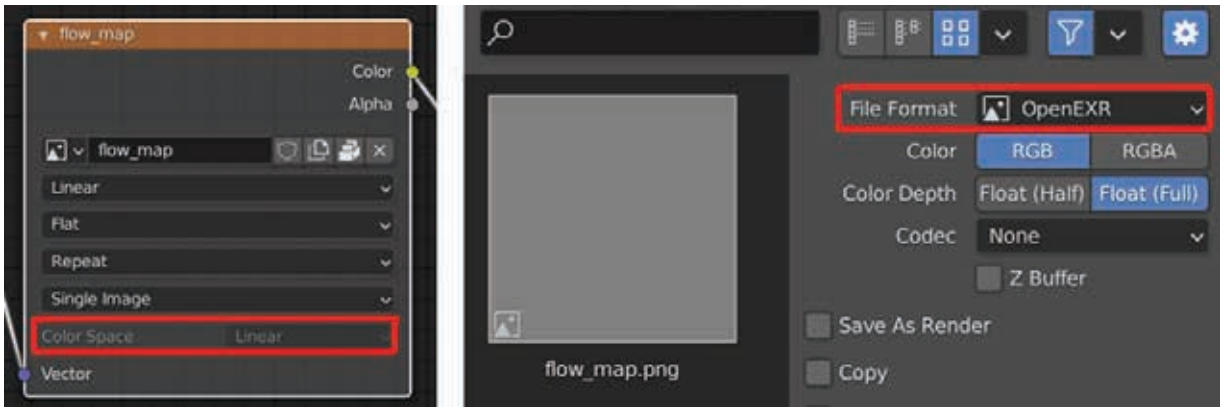


As a texture you can use an HDRI (I used this one by Sergej Majboroda), or you could paint the texture on the inside of the UV sphere by hand if you're going for more of a 2D animation look. For the rest, it's the same workflow as before: append the water material from the example Blend file, then copy/paste and connect the relevant nodes.



Finally, go to Texture Paint mode, make sure the Flow Map is selected and click on "Flowmap 3D Paint Mode". **Press Spacebar** to see the flowing motion while you paint.

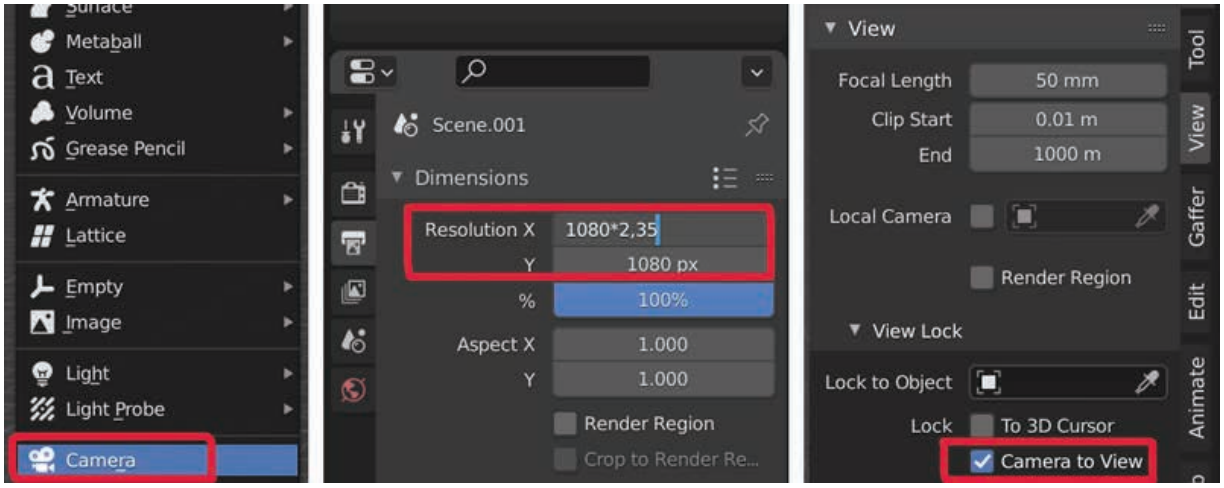
A final tip: The flow map texture needs to be set to **Linear** color space for best result, not to sRGB. Instead of a .PNG file, you can save the resulting flow map as an **.EXR** file as well. That's a great file format for containing a lot of data and as a bonus, Blender won't try to keep changing it back to sRGB every time you save.



Moving Cloud Shadows

Having cloud shadows moving accross the surface of your 3D environment is a great way to add movement and even dramatic lighting to an otherwise static and boring image. For this example, I'm using a photogrammetry scan of a city from Sketchfab.

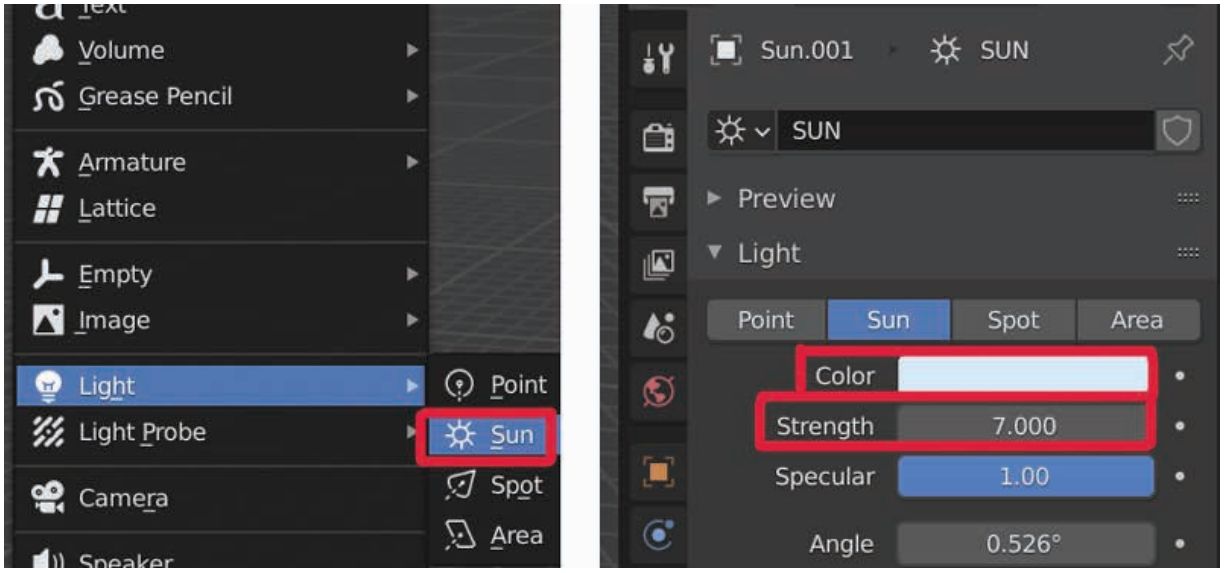
I added a **Camera** (Shift+A > Camera) and set the aspect ratio to cinemascope for a more cinematic look. Here's how: take the resolution Y value (1080 by default) and multiply it by 2,35. In this case that gives us a resolution of 2195 x 1080. This is not strictly necessary, the cinemascope aspect ratio is just really cool!



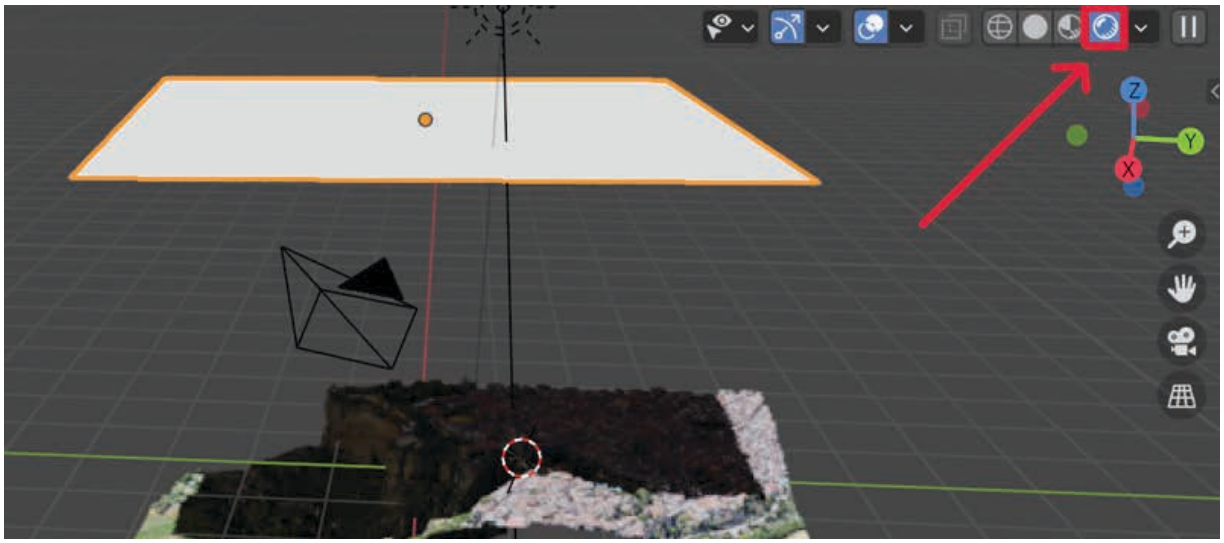
In the Sidebar, enable **Lock camera to view** and **press 0** on the numpad to position the camera easily.

Find a nice angle and once you're satisfied, **uncheck Lock camera to View**, so so you don't accidentally move the camera.

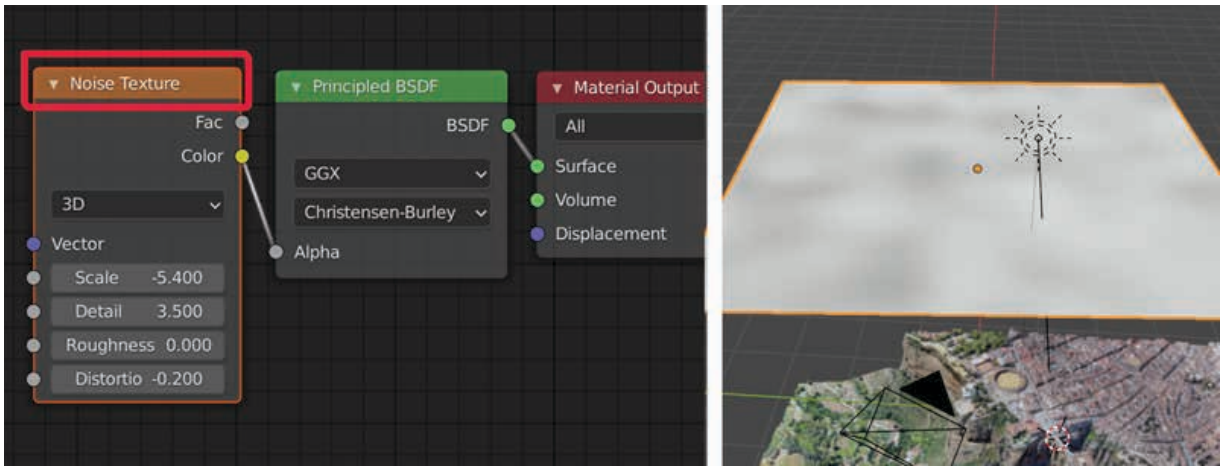
By adding a **Sun lamp** (Shift+A > Light > Sun) we add some light to the scene. Move the Sun lamp up, so you can see it more clearly and rotate it to change the way the environment model is lit. I found it to be a bit weak, so I increased its **Strength** and gave it a slightly blue tint.



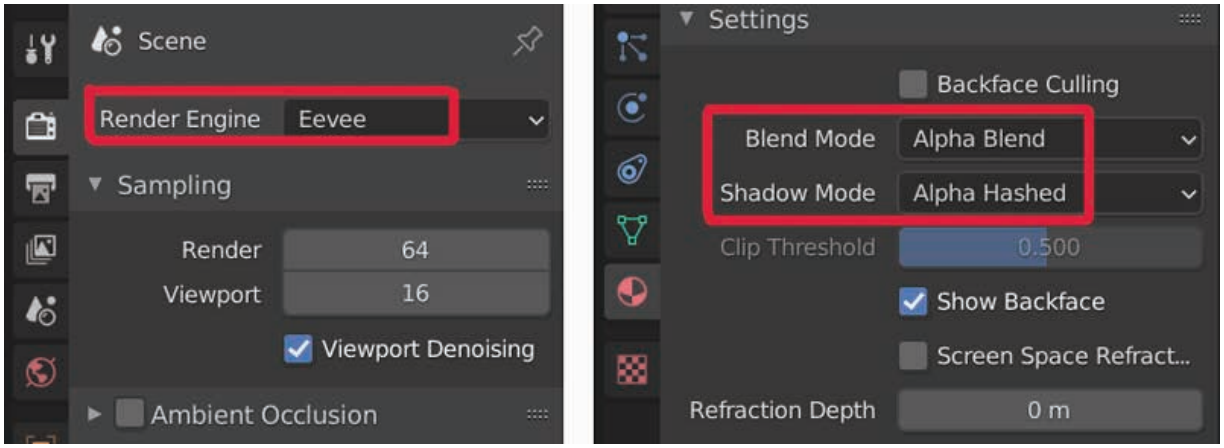
Now, if you add a **Plane** (Shift+A > Mesh > Plane) and place it above the landscape model, you'll notice a big square shadow, as if the town is being visited by some kind of square UFO. If you don't see the shadow, make sure you're in **Rendered preview**. Be careful that the plane is not in front of the camera. Scale it up so that its shadow covers the entire landscape.



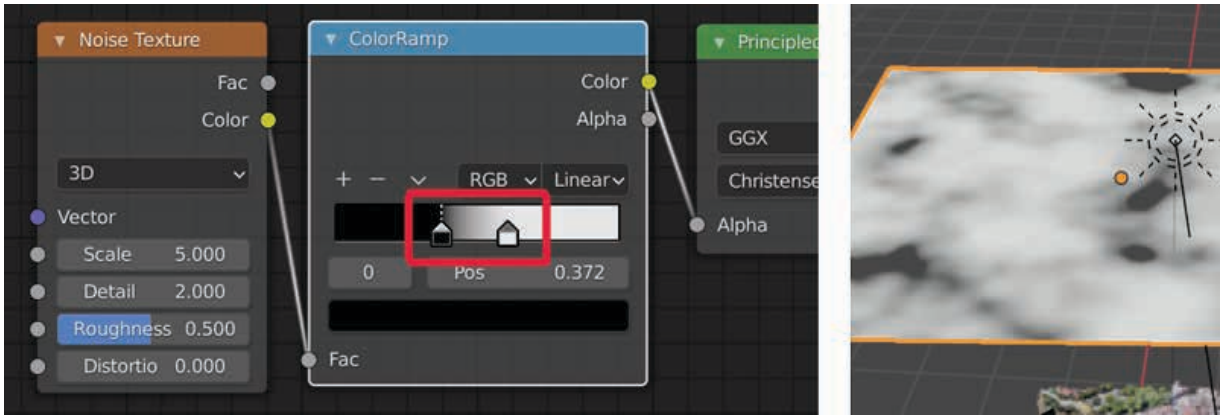
Give this Plane a **new material** and name it "Clouds" and in the **Shader Editor window** add a **Noise texture** (Shift+A > Texture > Noise Texture). Plug its color output into the **Alpha input** of the **Principled BSDF**. You'll notice that the plane has some kind of cloudy gray spots on it.



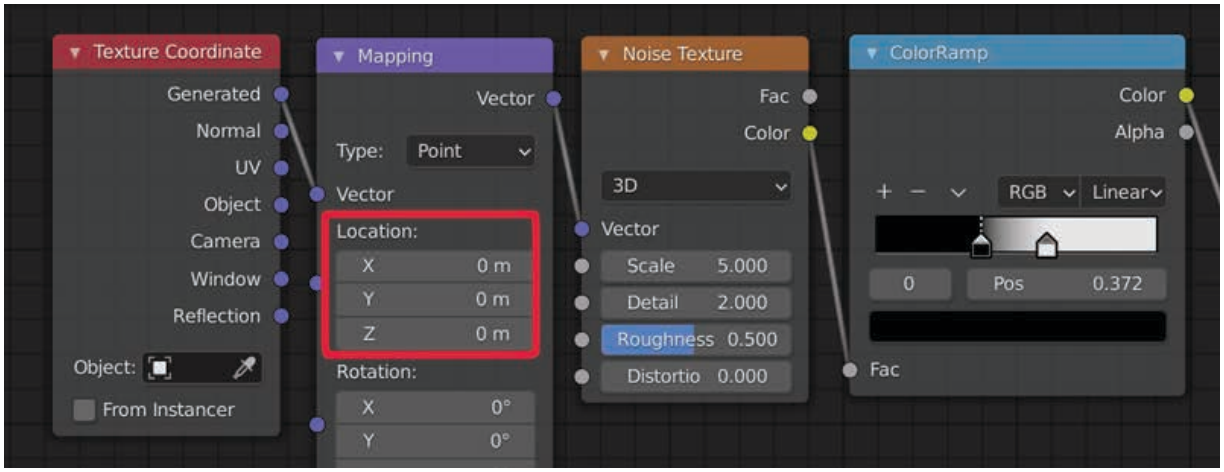
For this to work in Eevee, you'll need to set Blend Mode to **Alpha Blend** and Shadow Mode to **Alpha Hashed** in the Material Settings. Of course, you can also just use Cycles as the Render Engine.



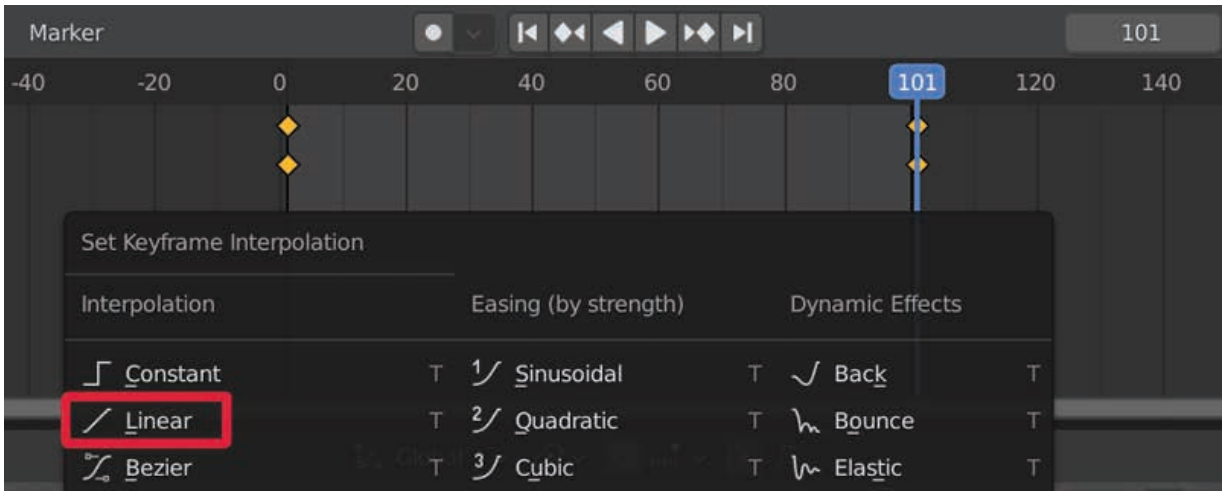
The effect is subtle. To make it more pronounced, add a **ColorRamp node** (Shift+A > Converter > ColorRamp). By bringing the two color stops closer together, you create a bigger contrast.



To animate the cloud movement, select the **Noise texture** and **press Ctrl+T** to add **Mapping and Texture Coordinate nodes** (provided you have Node Wrangler add-on enabled in Preferences). The mapping node has values that can be keyframed, location being the most obvious one.



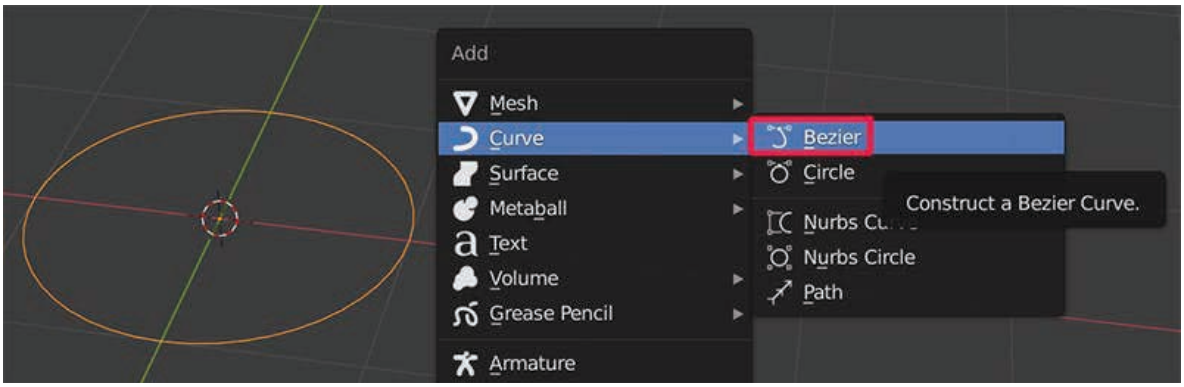
One thing to pay attention to, is when you set the keyframes for the Cloud location make sure to select them in the Timeline or Dope Sheet, press T and set them to Linear interpolation, so they don't ease in and out.



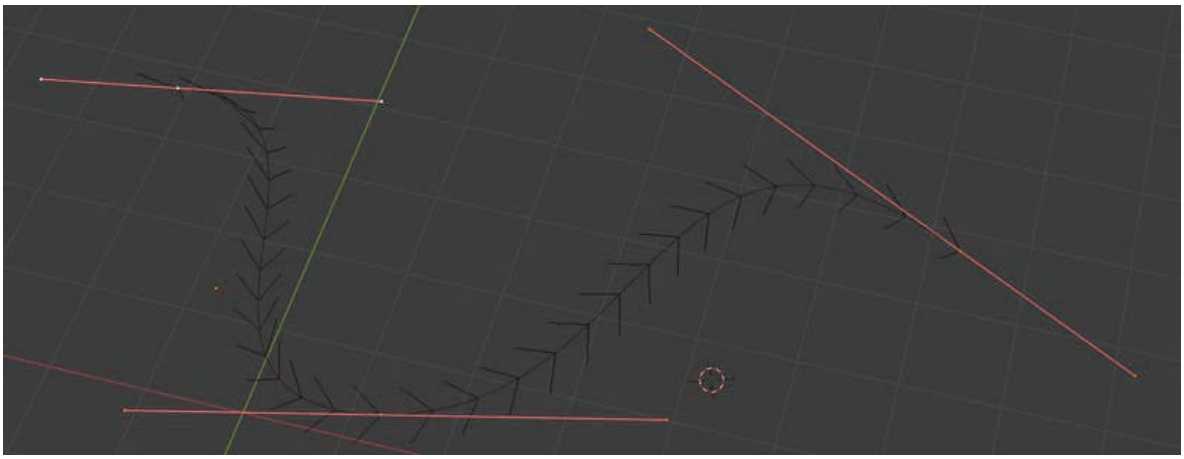
Animated growing Plant

In this tutorial we'll look at creating this growing plant animation, where leaves appear on a growing stem.

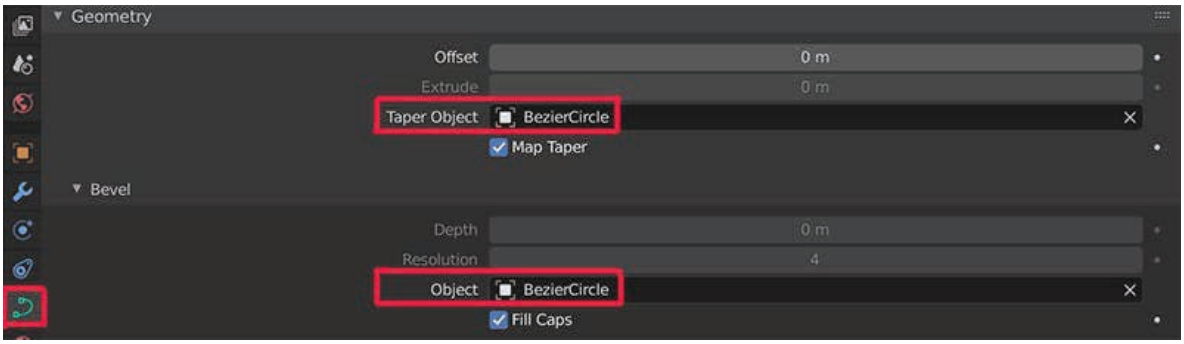
Press Shift+A and choose Curve > **Circle** to create a Circle. This will be used to give the stem its thickness. **Press G** and move it out of the way. **Press Shift+A** and choose Curve > **Bezier**. This will be the stem of the plant.



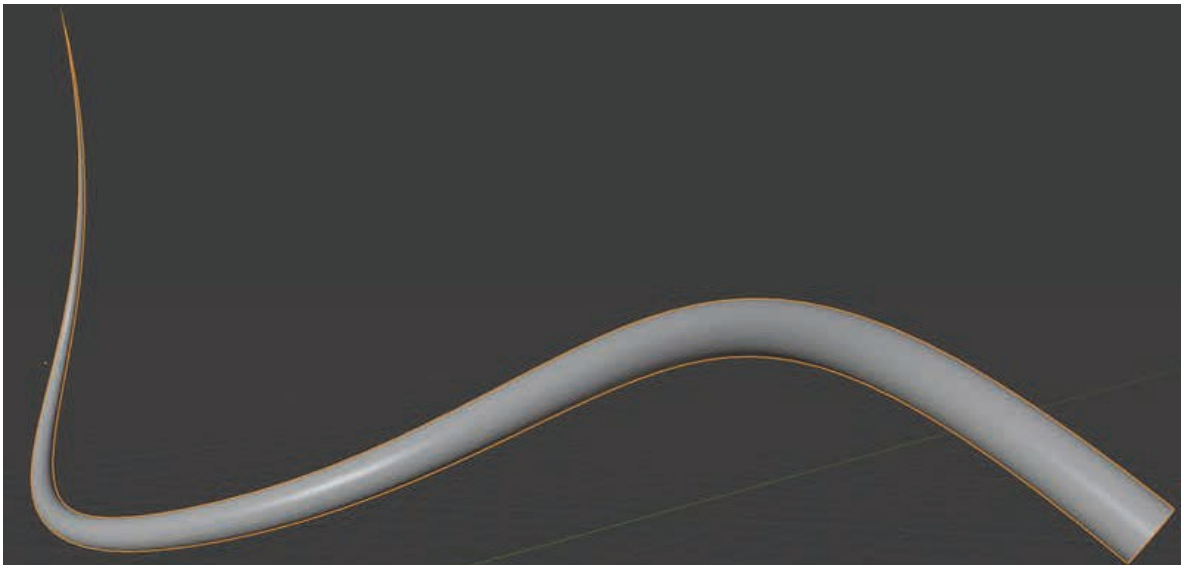
Press TAB to go into Edit Mode and edit the Bezier curve to have a more organic plant-like curve. You can edit the points with the usual commands: **G to move, R to rotate, S to scale**.



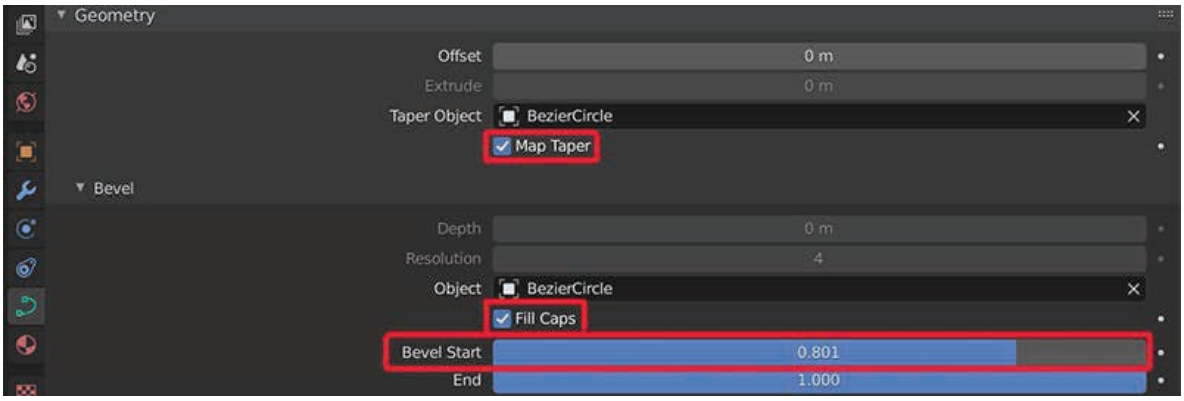
Press TAB to go to Object Mode. In the Curve tab, under Geometry in the **Taper Object** field pick the Circle. Pick the same circle again in the **Bevel Object** field. This will give the plant stem a gradual taper and roundness. Scale the Circle to change the thickness of the stem.



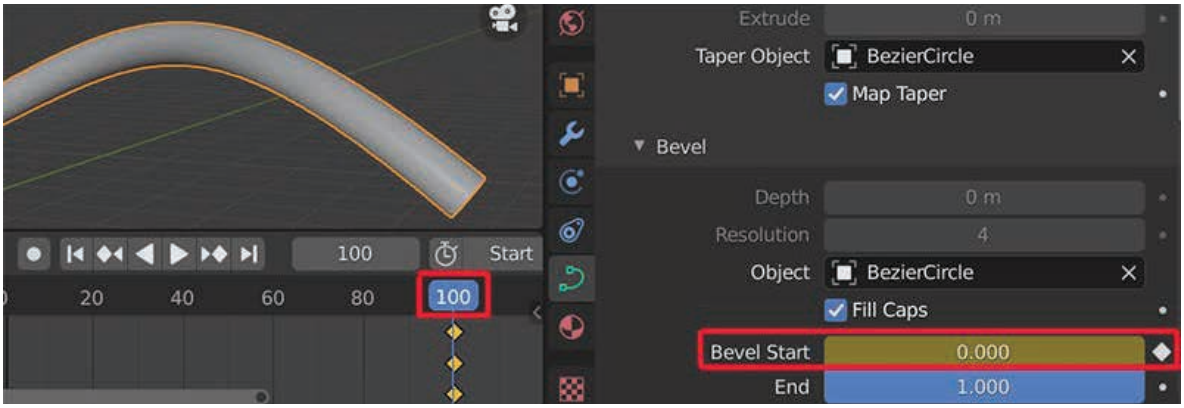
Now you'll have something like this. You can still **press TAB** to go into Edit Mode and edit the Curve.



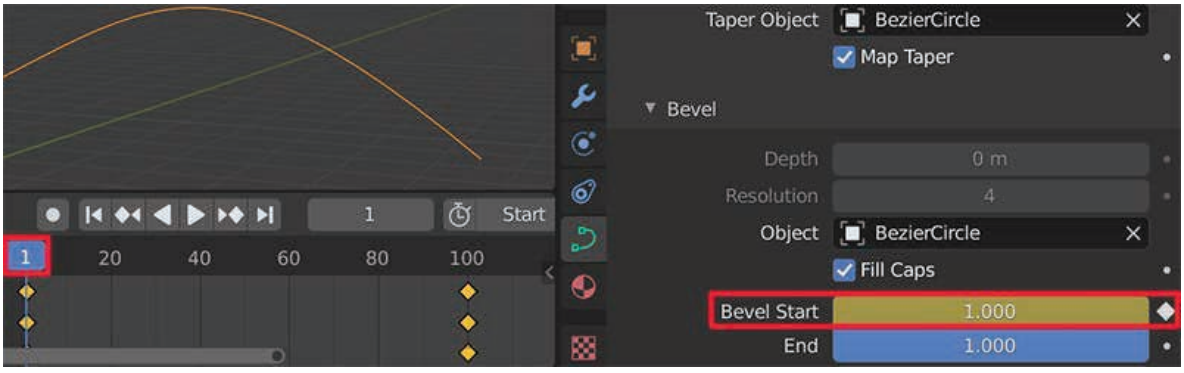
Now make sure you are in **Object Mode**. You'll notice that the **Bevel Start value** controls the plant's length. Changing the value from 1 to 0 "grows" the plant. Check **Fill Cap**s to make sure the plant cylinder has a cap at the end. Check **Map Taper**to make sure it's always tapered to a point no matter how long it is.



Go to frame 100 and set a keyframe of **Bevel Start value 0**, in other words for the plant to have reached its full length. **Press i** to set the keyframe while hovering the mouse over that field, or click on the key icon next to the field. It will turn yellow.

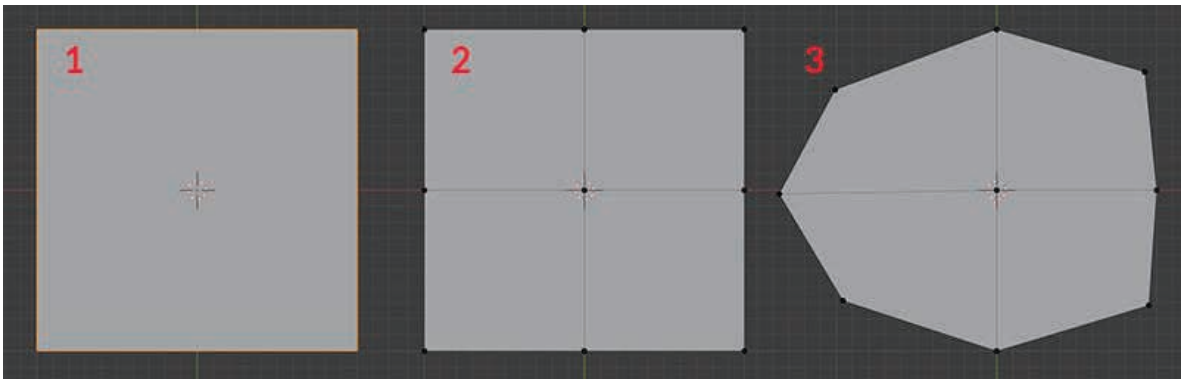


Similarly, set a keyframe at frame 1 for a **Bevel Start value of 1** (in other words, the plant has no length yet at frame 1).

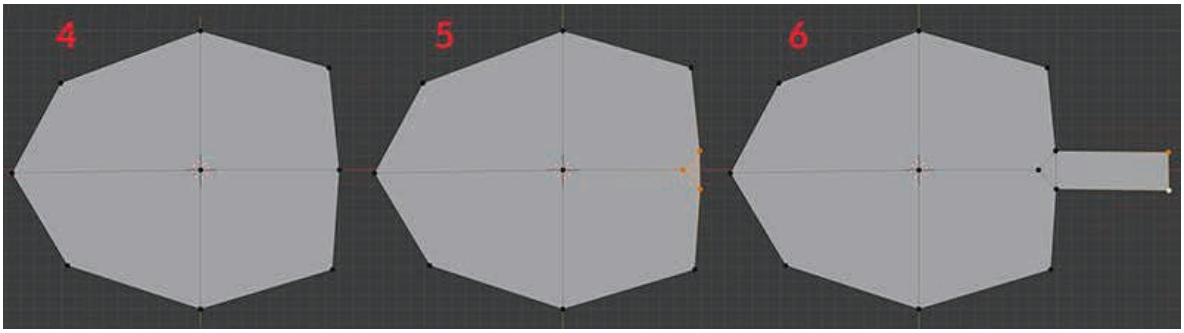


To quickly go back to the first frame on the Timeline, **press Shift + [left arrow]**. If you press Spacebar from frame 1, the plant will grow before your eyes. So far so good! Now that we've created the growing plant stem we can add leaves that grow as the stem gets longer.

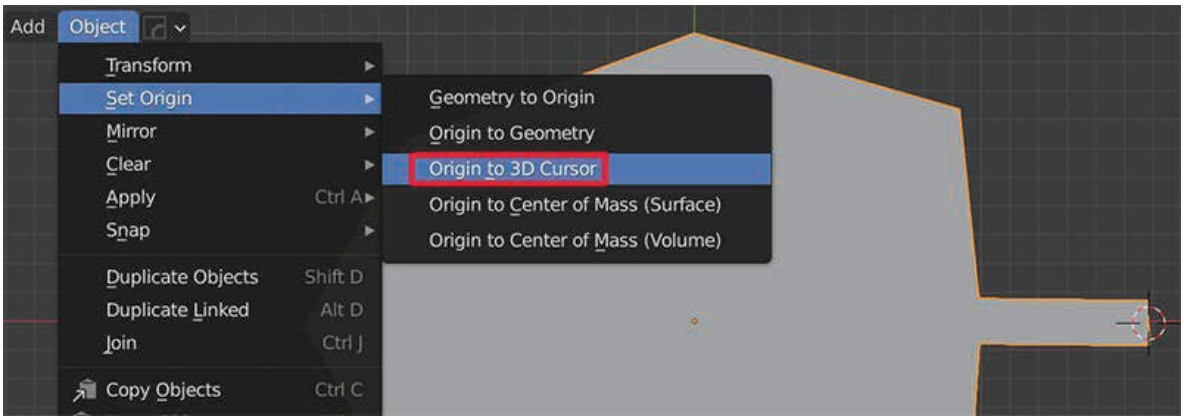
Press Shift+A and add a **Plane**. In Edit Mode right-click and Subdivide it. Move the vertices to give it more of a leaf shape.



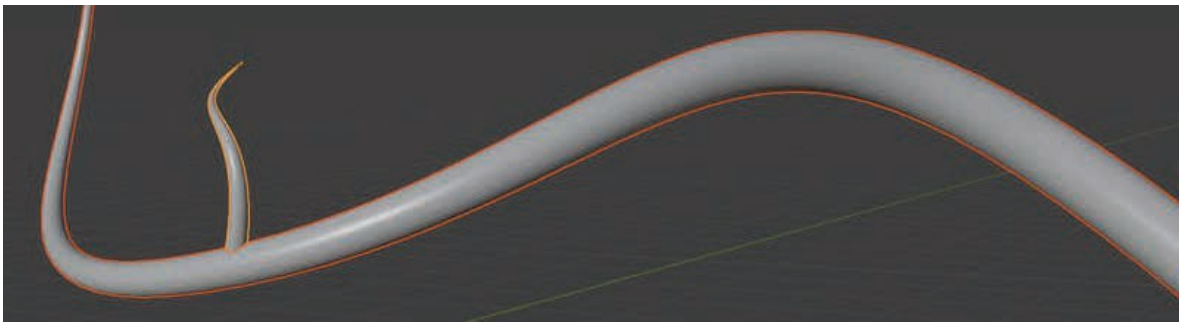
Then select the bottom vertex and **press Shift+Ctrl+B and drag** to create two more vertices. **Press E to extrude** these two bottom vertices to create the leaf stem.



Give the leaf a more organic, imperfect look by moving some of the vertices, either one by one or using Proportional Editing. Now we need to make sure the origin of the leaf is at the bottom of its stem. To do this select the bottom vertices of the stem and **press Shift+S**, choose **Cursor to Selected**. Next **Press TAB** to go to Object Mode. Go to Object > Set Origin > **Origin to 3D Cursor**.

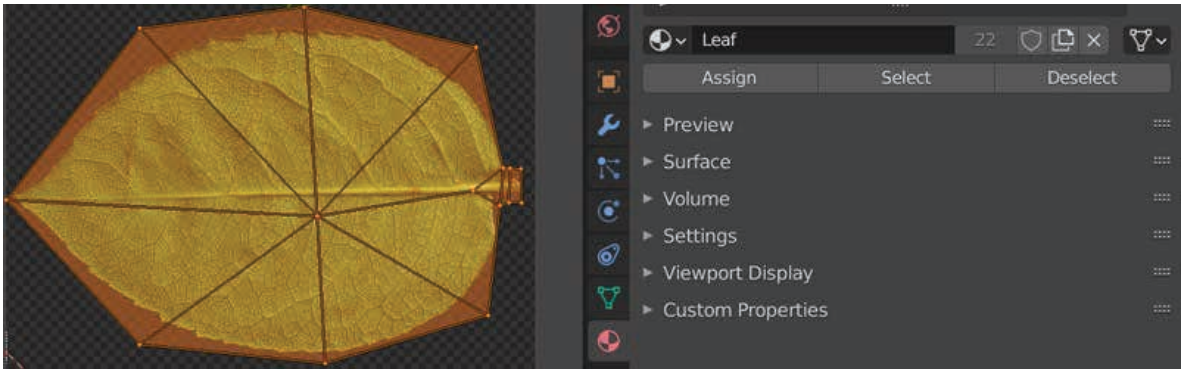


This is important for the animation of the leaves growing, as they need to grow from the origin, which should therefore be at the bottom of the stem. Optionally, using the techniques learned in the previous tutorial, we can create one or more additional twigs on the main stem.

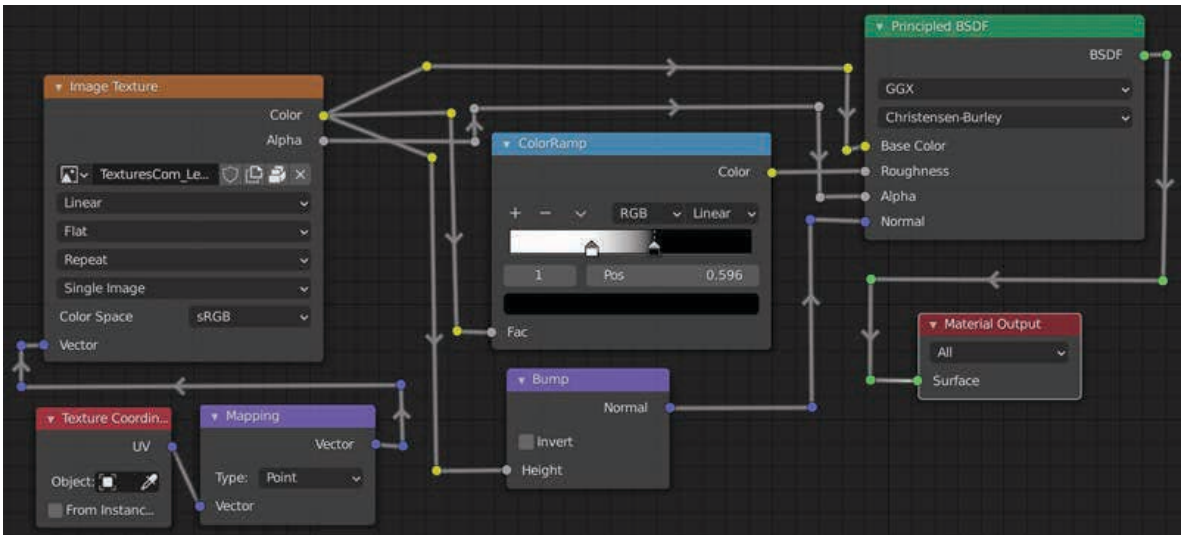


Now (making sure you are in Object Mode) position your leaf on the stem with **G (move)** and **R (rotate)**. **Right-click** on it and choose **Shade Smooth**.

Before we go further, let's give the leaf a material. In the **Shader Editor**(or in the Material Tab), click **New**. Then in Edit Mode select all, **press u** and choose **Unwrap**. I used a free leaf texture from textures.com, called "TexturesCom_Leaves0126". You can move the UV's in the UV Editor window to make sure the leaf texture fits inside the UV's.

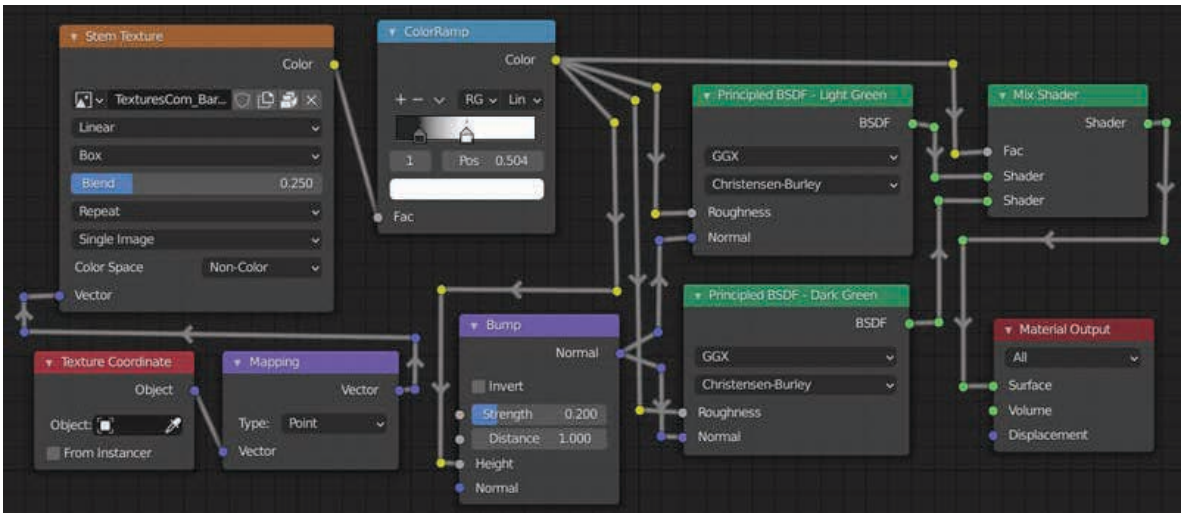


To add some bump, I used the same image texture and plugged it into a Bump node. I also used the same image for the Roughness, with an added ColorRamp node in between for a bit more control. You can see the shader set-up here:

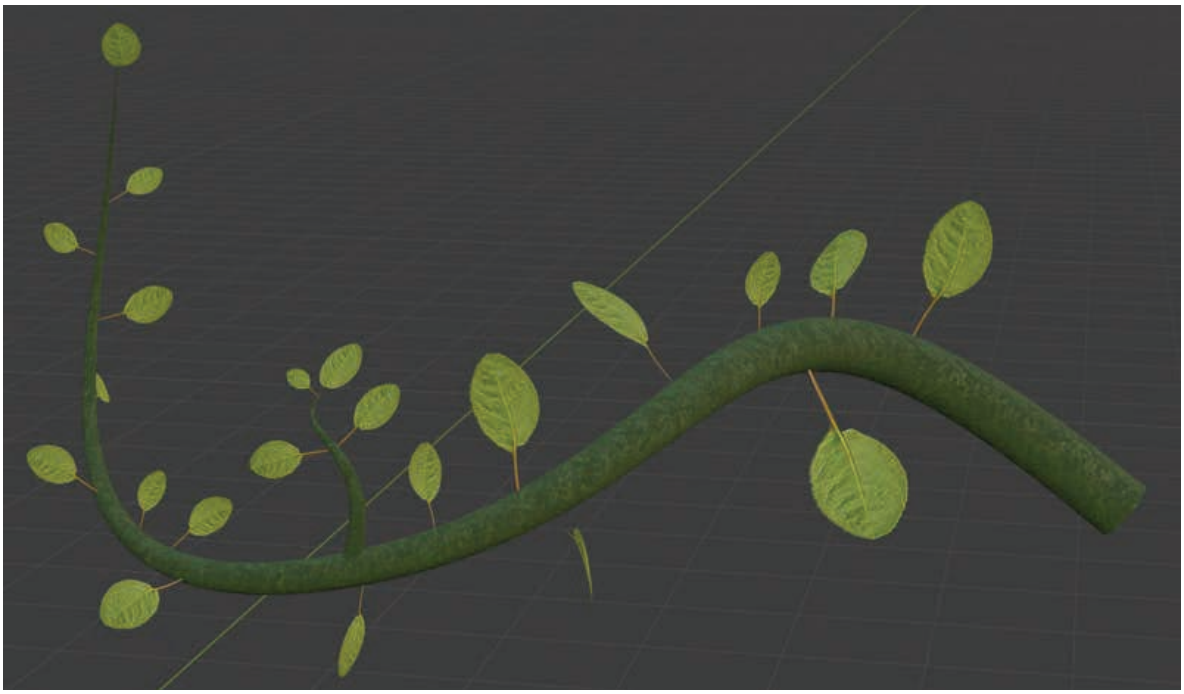


In case you're curious, to reroute the node connections like shown above, I used the **Reroute node**. Basically, you **hold Shift and drag across a node connection while holding the Right Mouse Button**. This adds a Reroute node and you can now **move that with G**. This way you can create neat node layouts that don't look so confusing (or less confusing, at least!).

Now that we're working on the materials, we might as well do the stem as well. For this, I used two **principled BSDF's** - one with a dark green, the other a lighter green-mixed with a **Mix Shader node**. As the factor to mix between the two, I used another free texture from Textures.com, "TexturesCom_BarkDecidious0174". As the **Texture Coordinates**, I used the **Object** instead of UV's. I also used the same texture for **Roughness and Bump**, like with the leaf shader. Below you can see the shader set-up.

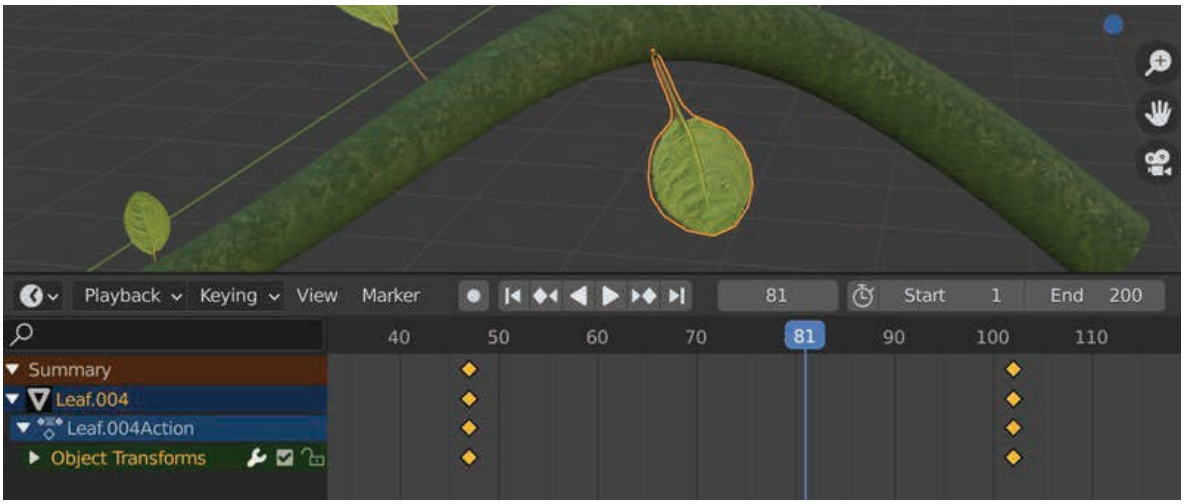


With the leaf selected **Press Shift+D** to duplicate it and place the duplicate somewhere else on the stem. Repeat a few times to place more leaves. Be careful: you need to use **Shift+D** (regular duplicate) and **not Alt+D** (linked duplicate) to duplicate otherwise you can't set individual keyframes for each leaf.



With the leaves placed, it's just a matter of setting **Scale key frames** for each individual leaf at different point of the Timeline. First set a keyframe **by pressing i and choosing Scale** for the fully grown leaf.

Then go back in time on the Timeline and scale the leaf to zero by **pressing S+0**. Then **press i and choose "Scale"** to set a scale keyframe. This way it will grow from 0 to 1. You can tweak the timing to perfection by moving the keyframes of each individual leaf on the timeline.

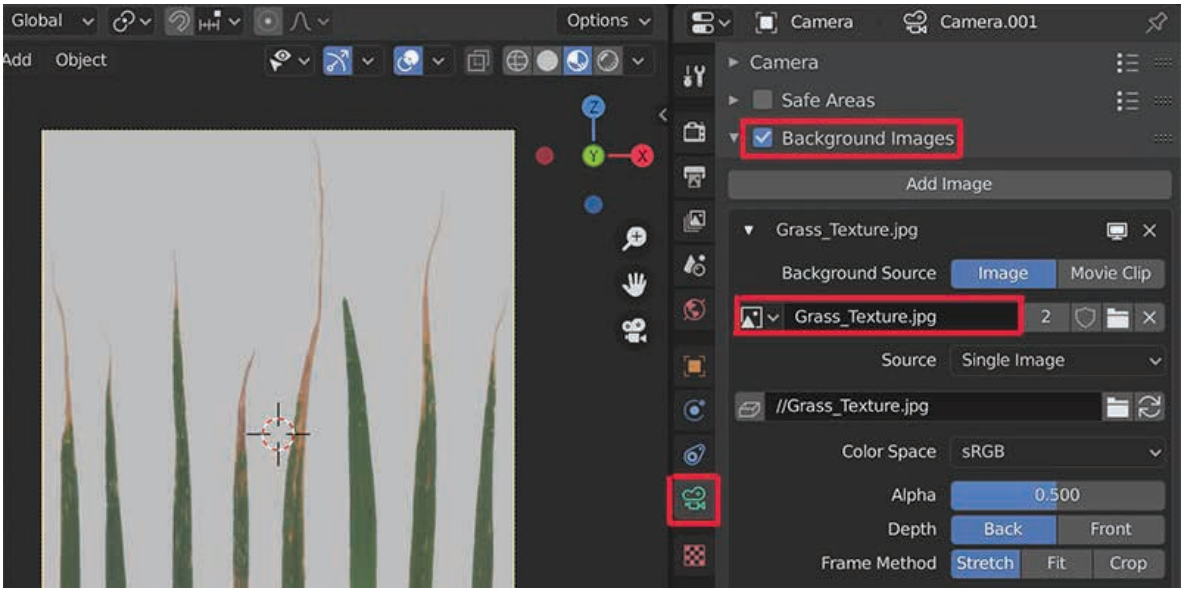


After that's completed, you've got your growing stew with leaves!

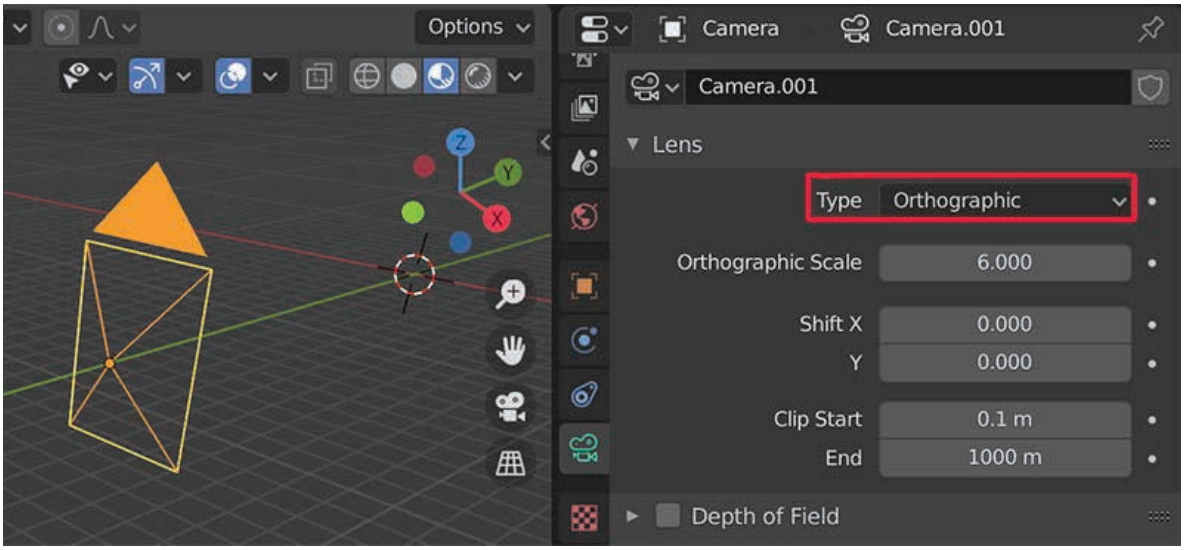
Grass moving in the wind

There are many different ways to create grass in Blender. However, when it comes to realism nothing beats a photographic texture. Real grass also moves, even if it's just barely visible, subtle movement. For maximum control over the animation, bones are the best solution. In this tutorial we'll create realistic animated grass this way.

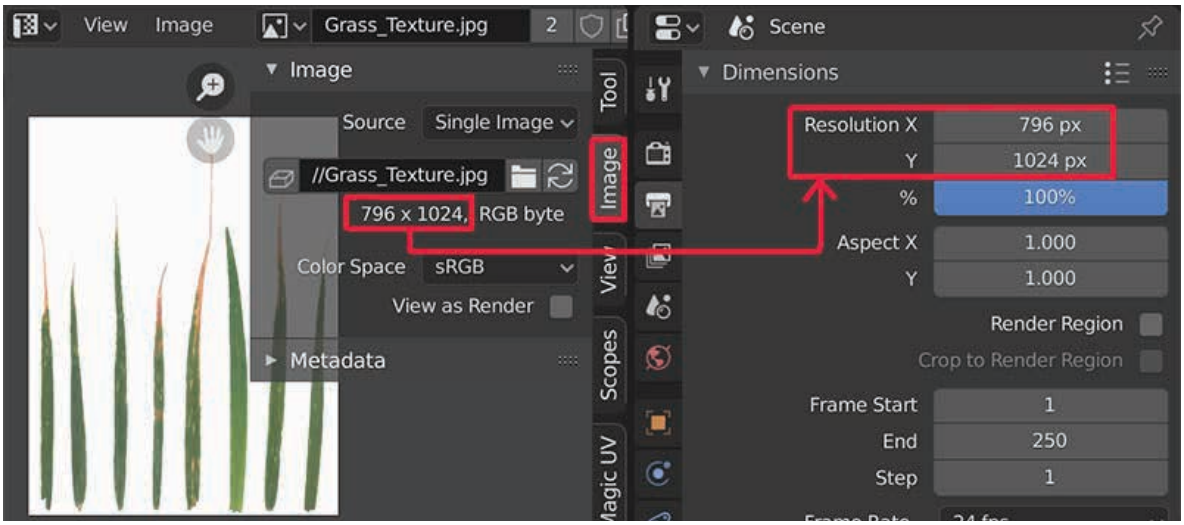
Download an image that features some grass strands. In my case, I couldn't find a free texture of grass strands so I went outside and picked some grass, which I then scanned! You can download my texture here. Check "**Background Images**" in the camera settings and load the grass texture. (Download the Blend file here).



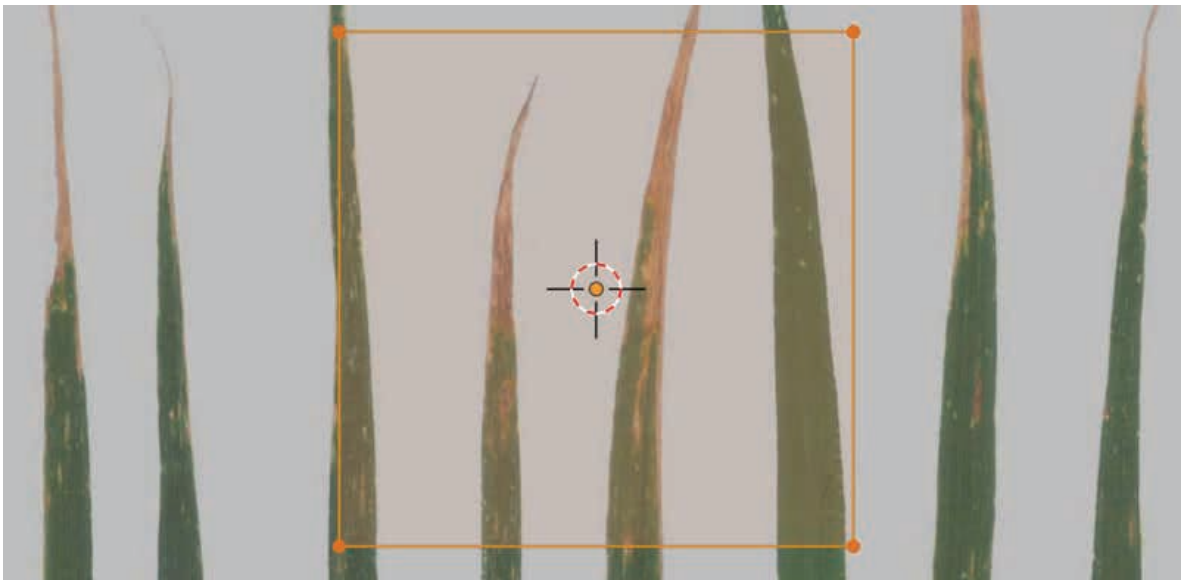
Go to the Front orthographic view by **pressing 1 on the numpad**. Now **press Ctrl+Alt+0** on the numpad to make that the Camera view. Go to the Camera tab and as **Lens Type** switch from Perspective to **Orthographic**.



Open a **UV Editor window** and load the grass texture there. Open the **Image** tab so you can see the image dimensions. Type these values into the **Scene Dimensions**, so the camera matches the background image. This will help ensure our texture gets projected correctly onto our grass models later.



Add a **Plane** in the Camera view. **Press Tab** to go into Edit Mode, select all by **pressing A** and rotate it 90° on the X-axis by **pressing R+X+90**. **Press Z** and choose **Wireframe** so the Plane becomes transparent.

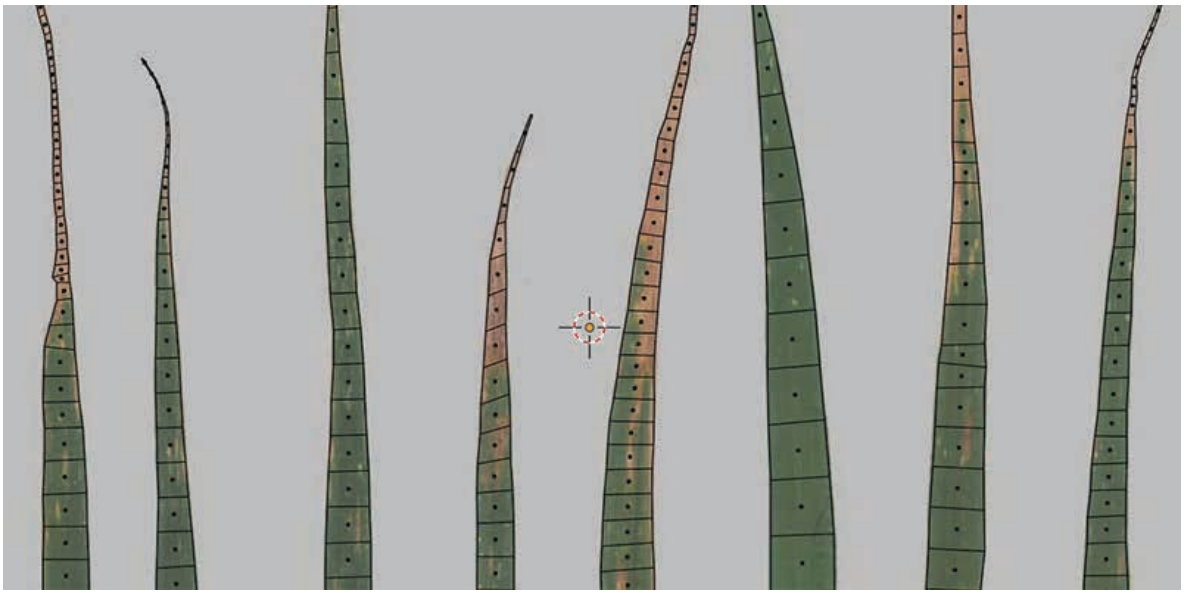


Now stretch the Plane along the Z-axis by **pressing S+Z** to fit one of the grass blades in the background image. Add edge loops by **pressing Ctrl+R** and move vertices with **G**.

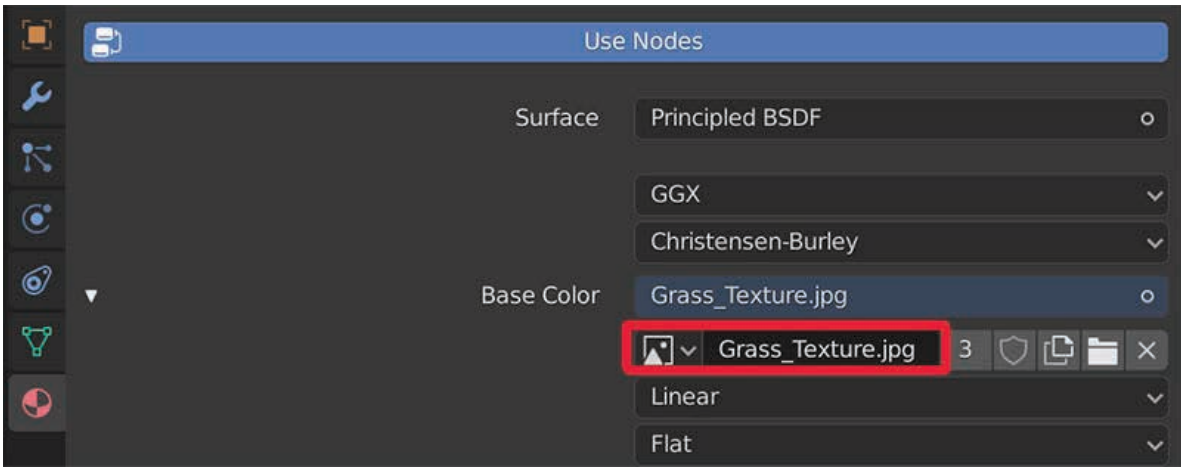
Add some more Edge Loops and continue to refine the grass shape. Be careful to keep the vertices inside of the grass blades, so there's no white edges from the empty white area of the image.

Select and duplicate one face of the grass blade and move it to the next grass blade in the image texture, then repeat the modeling steps. Do this for all the grass blades in the images until you've modeled all 8 of them.

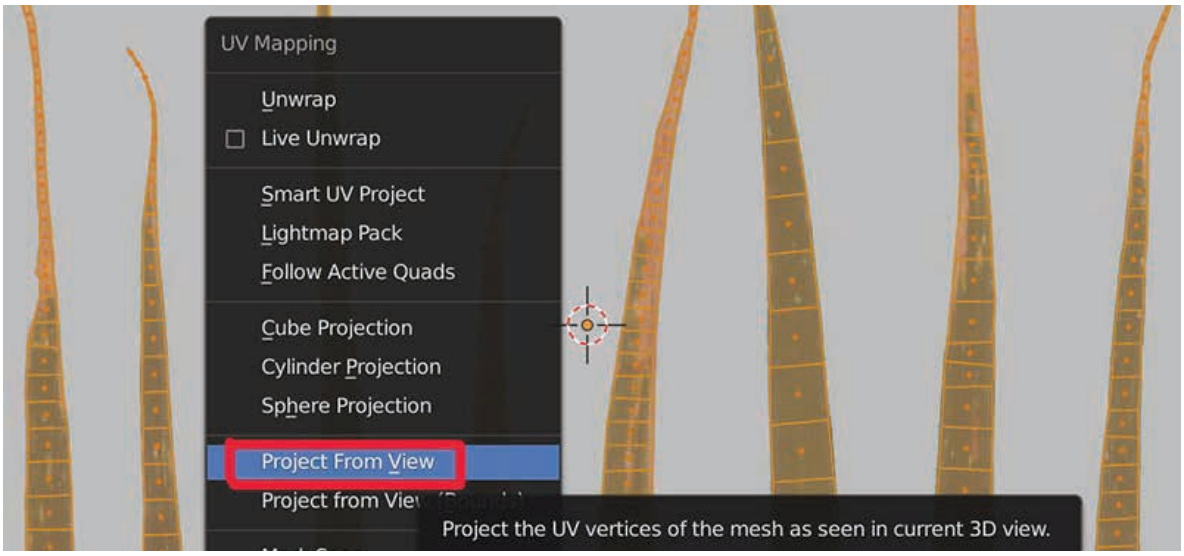
The good thing about using photographic reference and textures, is that you would never come up with these kind of organic shapes by yourself. It will add realism to do it this way.



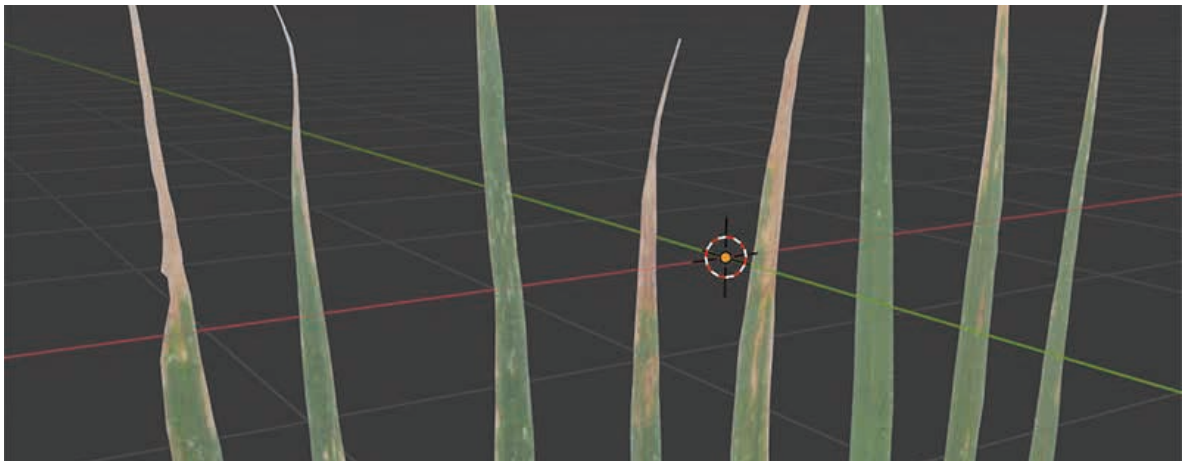
Give the grass blades a material and give that material the same image texture of scanned grass blades.



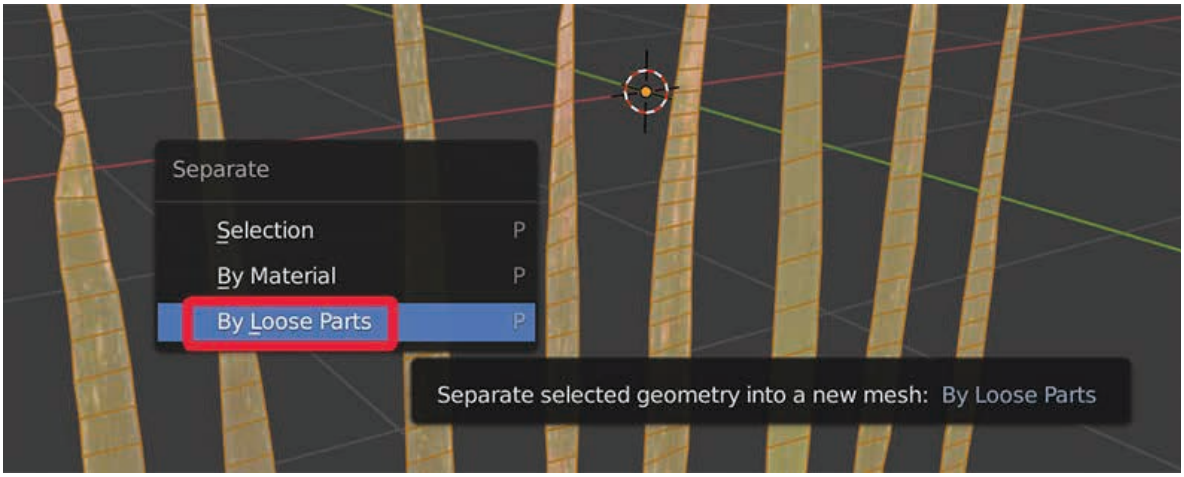
In Edit Mode select all and **press U**, then choose **Project From View**. As long as the scene dimensions match the image texture dimensions, this should give you a perfect UV map for the grass texture.



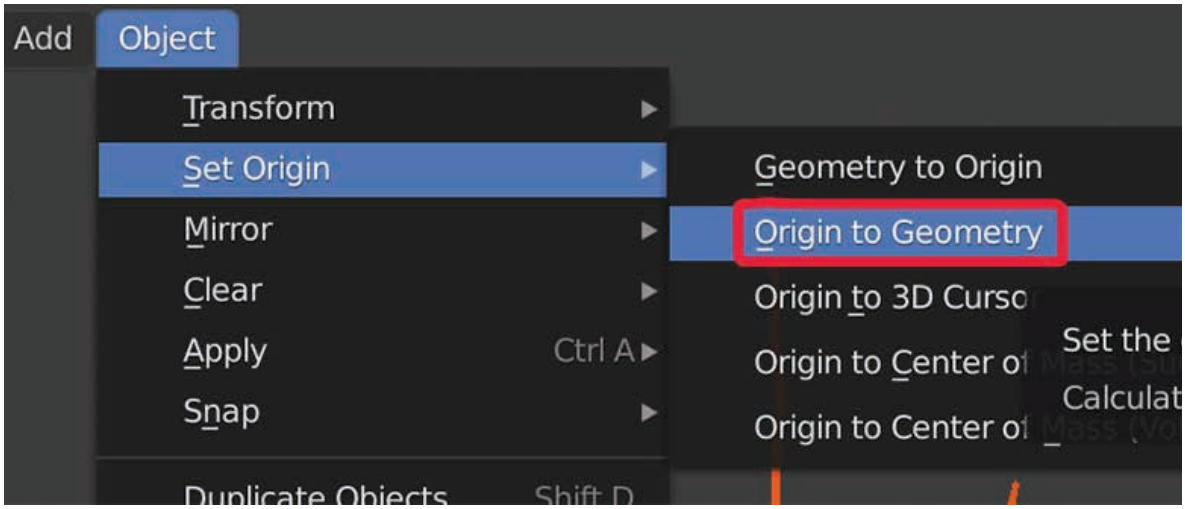
Leave the camera view and check the grass blades in **Material view**. As you can see they are now textured.



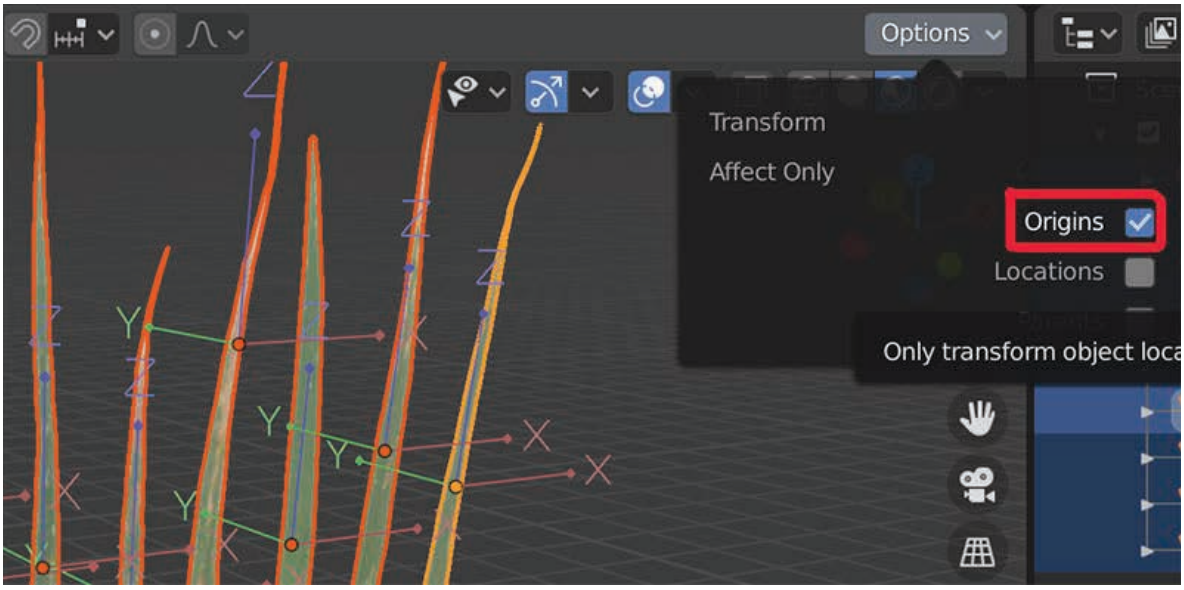
In Edit Mode select all and **press P**, then choose "**By loose parts**". This makes individual objects from all the grass blades.



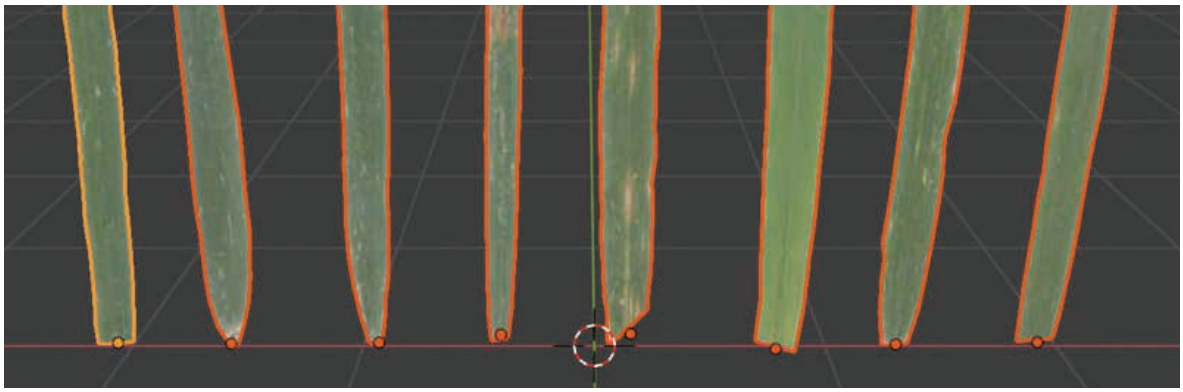
Select all the grass objects and go to Object > Set Origin > **Origin to Geometry**.



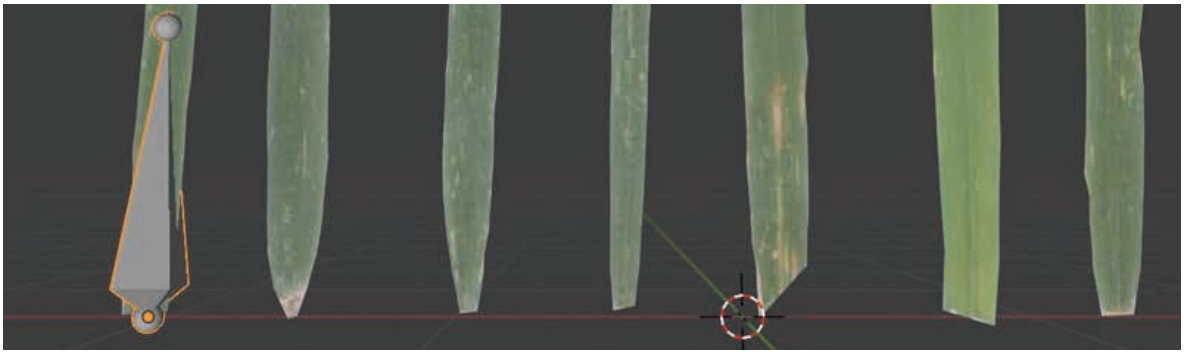
Now you need to move their origins to their root. Go to Object Mode and Under **Options** in the top right corner, check "**Origins**".



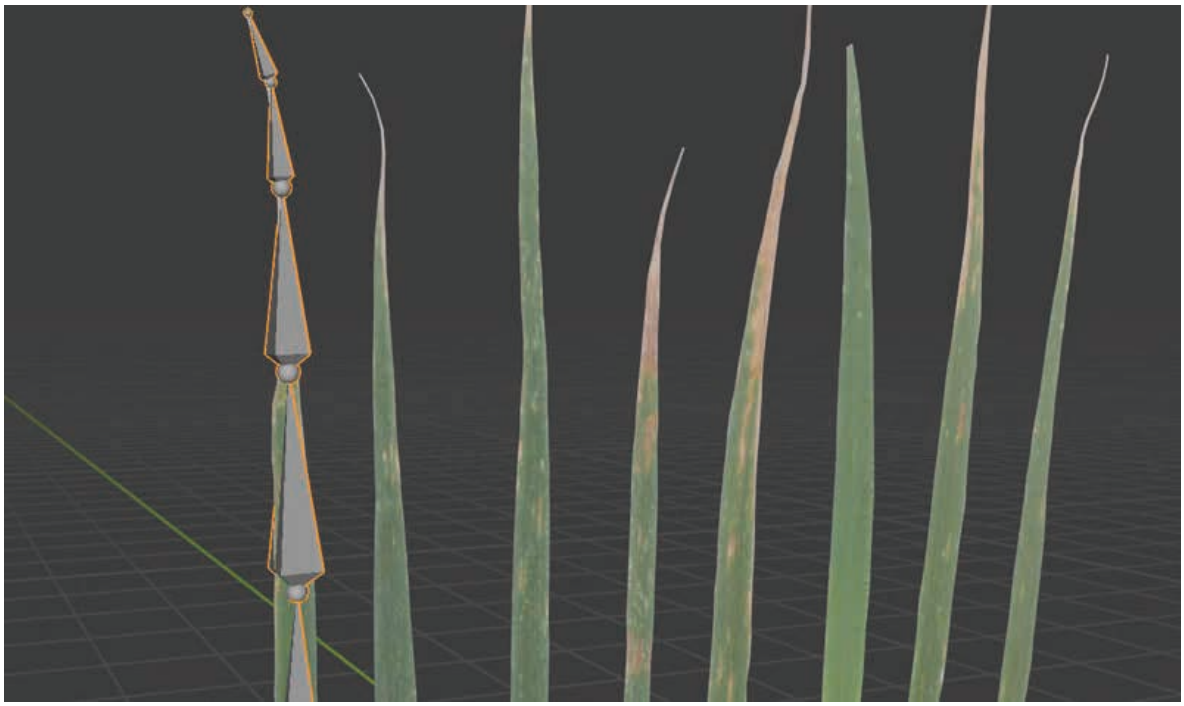
Now you can select a grass blade and move the origin point so that it is located at the base of the grass blade. Do this for all the grass blades. When you're done, uncheck the "**Origins**" box.



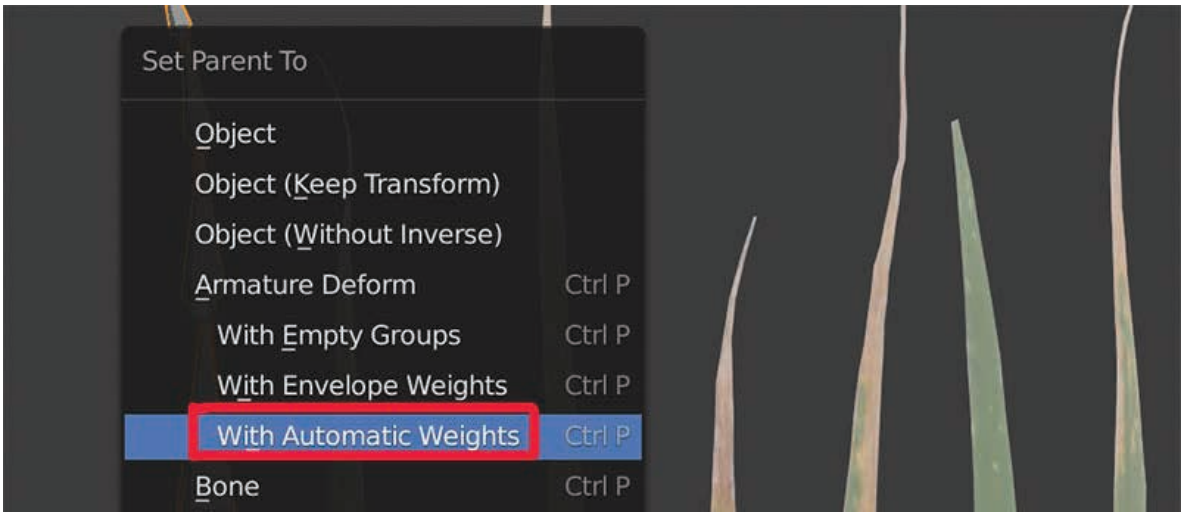
Press Shift+A and choose **Armature > Single Bone**. Move it to line up with the base of a grass blade.



Extrude the bone in Edit Mode a few times until you've got 5 bones all the way to the tip of the grass blade. If necessary adjust the bones in Edit Mode to follow the shape of the grass blade.

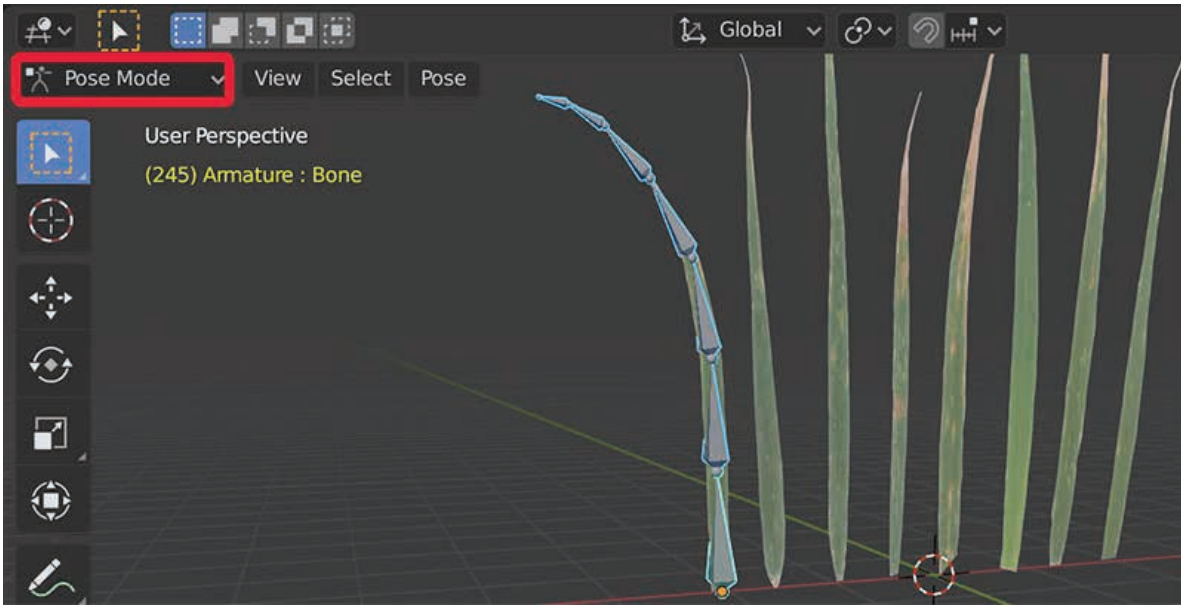


Select the grass blade, then while holding shift select the armature. **Press Ctrl+P** and choose "**with automatic weights**".

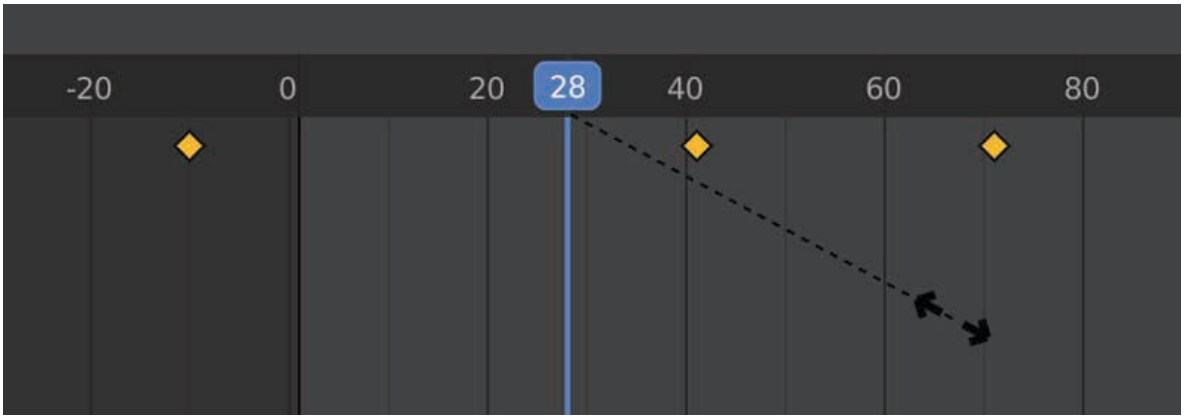


Now you can move the grass blade in **Pose Mode**. The armature bones won't show up in the render.

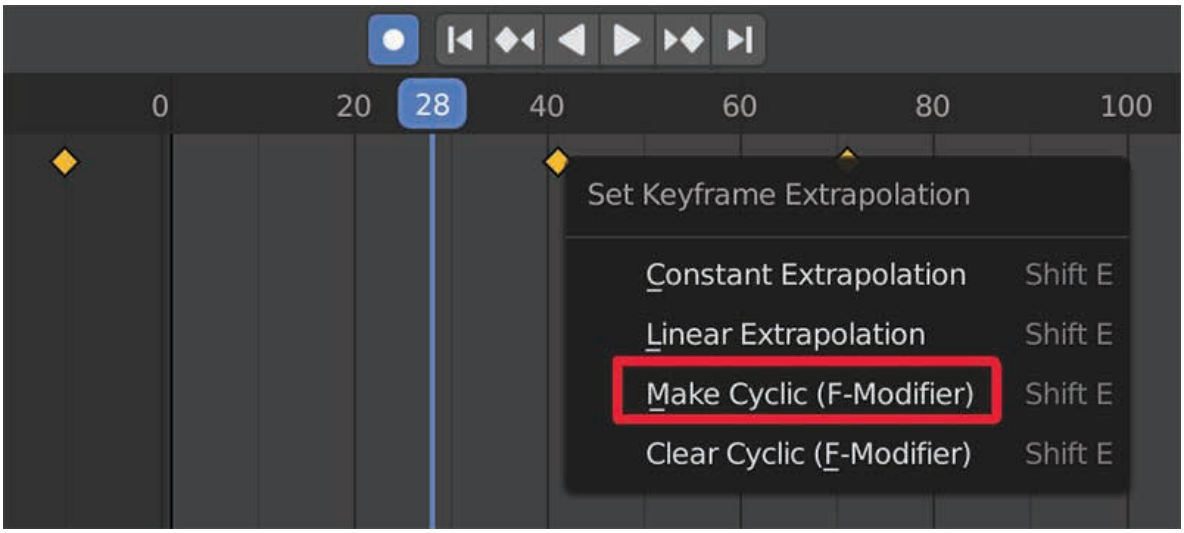
In Pose Mode, on frame 1 create an initial pose for your grass blade. Select all the bones in Pose Mode and **press i**, choose **LocRotScale**. This is also a good opportunity to make it look less flat. Go to a later frame like frame 100, and with all bones selected, **press i** and set another keyframe for **LocRotScale**. These two keyframes should be of the same pose. Then, set one or two more keyframes of different poses in between frame 0 and 100.



Check your animation by **pressing Spacebar**. Remember to keep it subtle and not too fast. You can always make the animation move slower by scaling the keyframes in the Timeline by **pressing S**and moving the keyframes further apart.

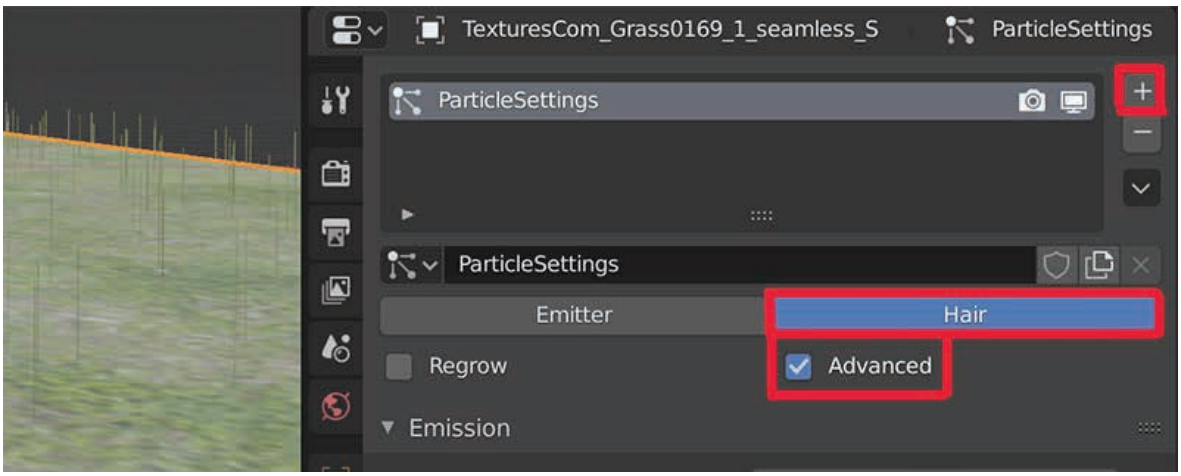


Once you think the grass animation is good enough, select all the bones and all the keyframes and in the Timeline **press Shift+E** and choose "**Make Cyclic**". This makes sure the animations keeps repeating.

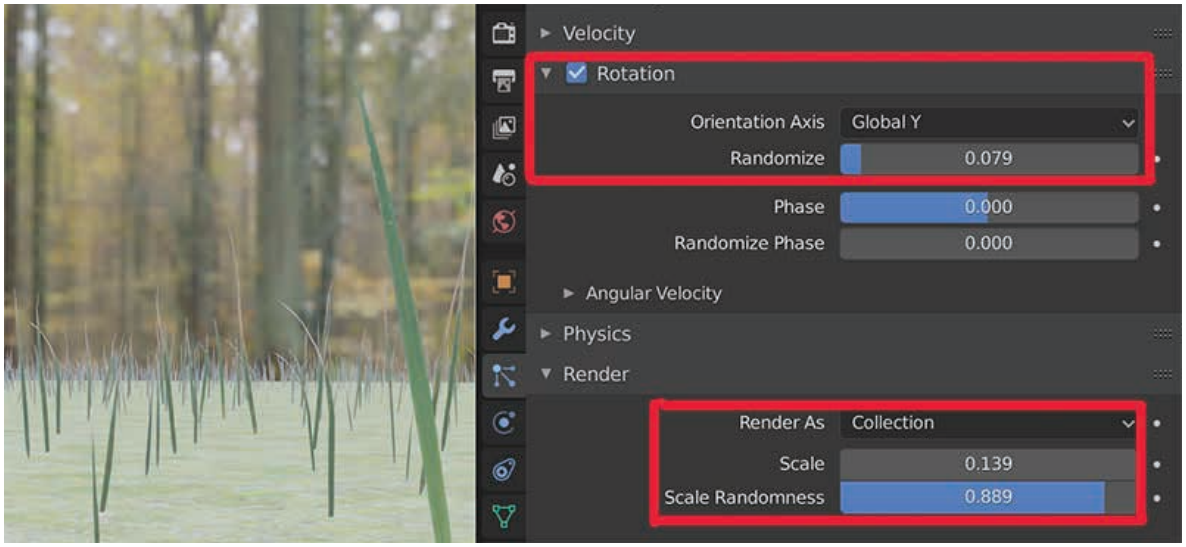


Repeat these steps of rigging and animating the other grass blades. Select all the grass objects in Object Mode and add them to a Collection by **pressing M** and typing a name for a new collection like "Grass".

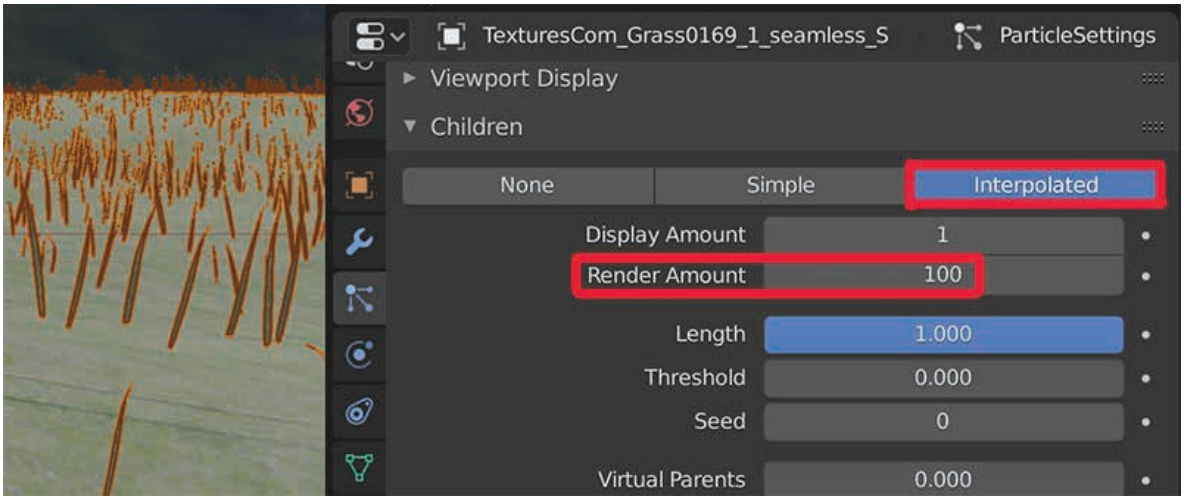
Now create a ground plane with a suitable texture. I used the free texture "Grass0169" from Textures.com. Add a Particle System to it and set it to **Hair** and check **Advanced**.



Under Render, choose **Collection** and use the "Grass" collection. Then, you'll notice that the grass lays flat on the Plane. Check **Rotation** and as the orientation axis choose **Global Y**. Then randomize the **Rotation** value a bit. Also randomize the **Scale** value. This makes it more organic.



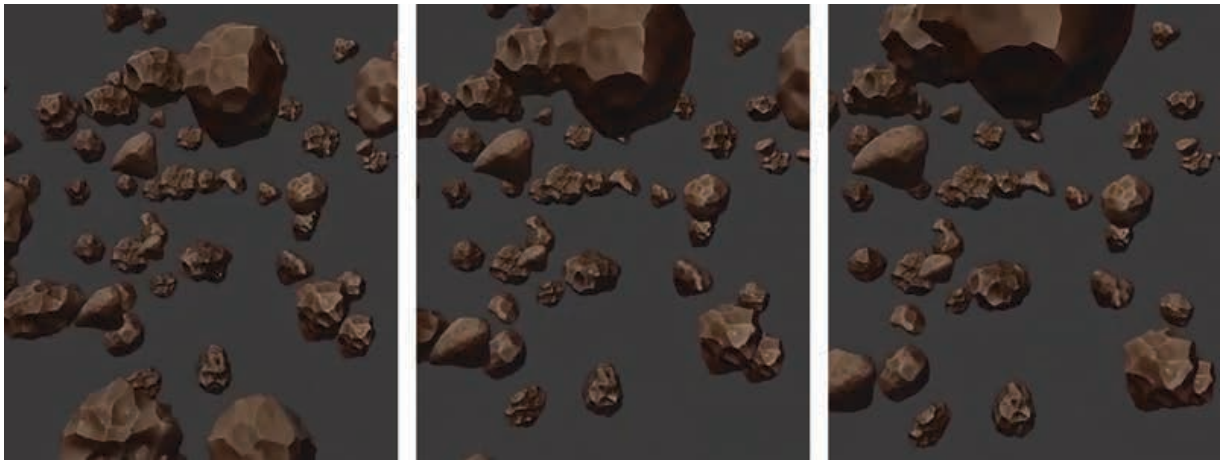
Obviously there's not enough grass on this field yet. In order to keep the scene manageable, we used a small amount of particles. We can configure this so that the render shows more particles. Under **Children** click on **Interpolated**. Set a higher **Render Amount** than the Display Amount.



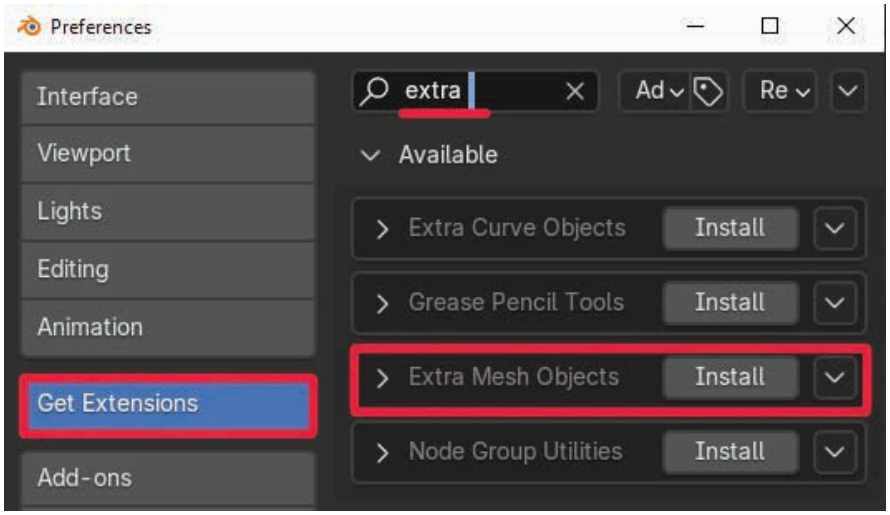
Now when you render, you get a lot more grass than what you see in the viewport (a hundred times more, in fact).

Asteroid field

Making an asteroid field is easy in Blender - there's even an "Asteroid" primitive object, if you know where to look. This video shows you how to get started with a simple space scene.



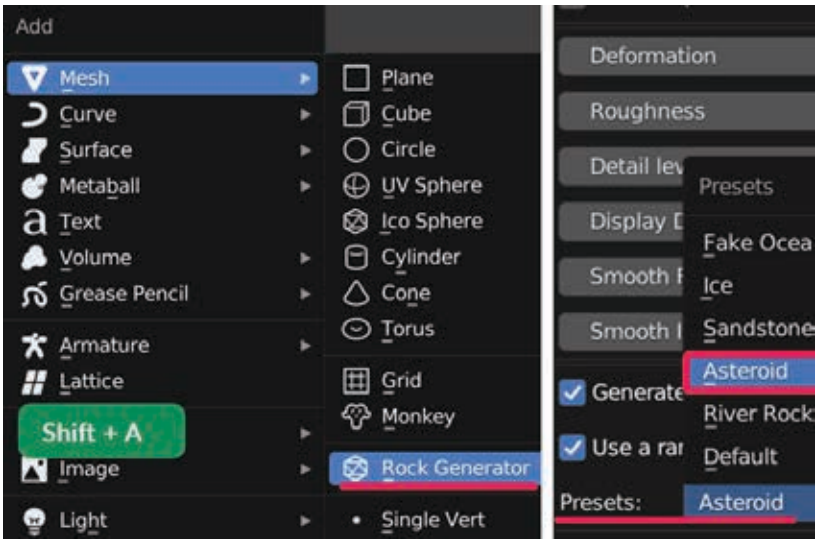
Enable the **Extra Mesh Objects** add-on and create one rock.



Press Shift+A to open the Add menu.

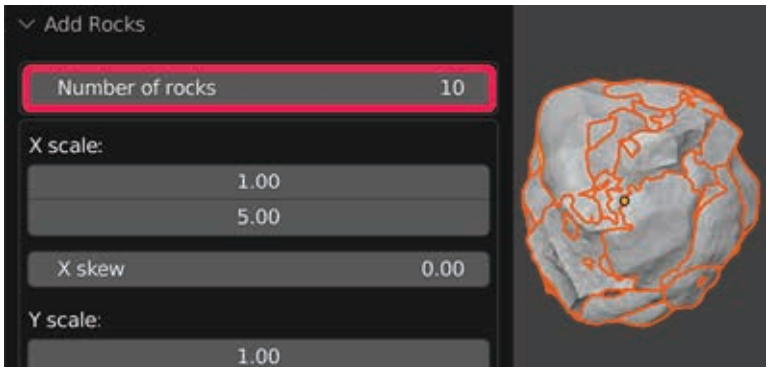
Choose Mesh > **Rock Generator**.

Open the options (**F9**) and choose the convenient **Asteroid** preset.

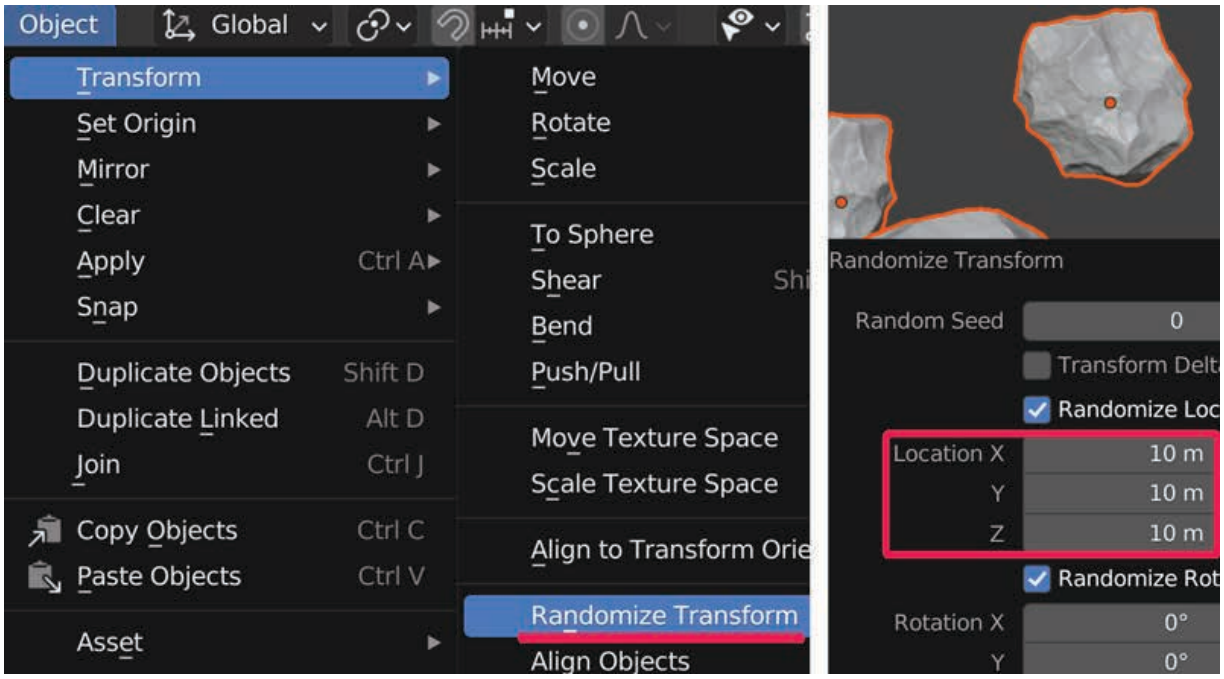


Click here to return to the index.

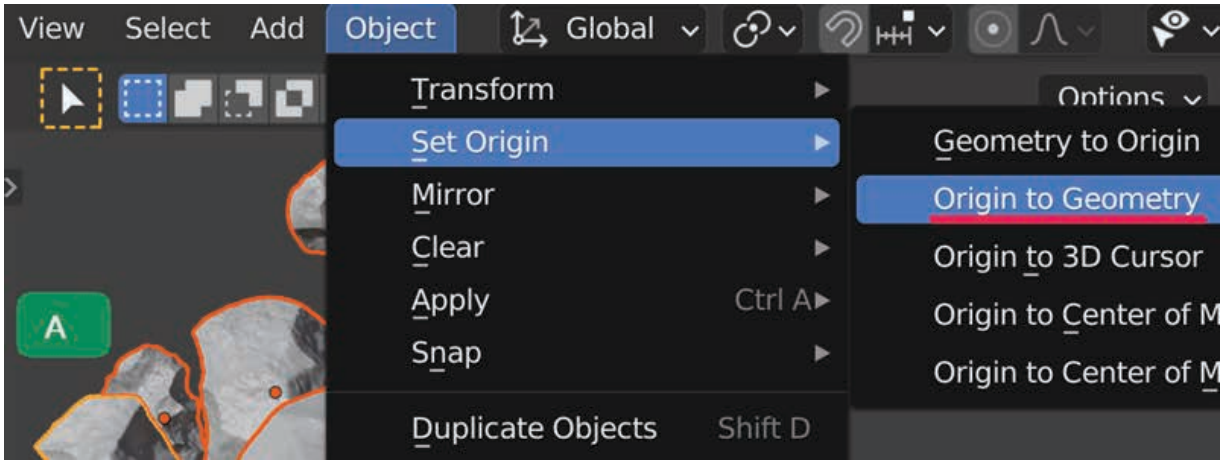
Increase the **Number of rocks**.



Select them all by **pressing A** and go to **Randomize Transform**. Increase the Location value.

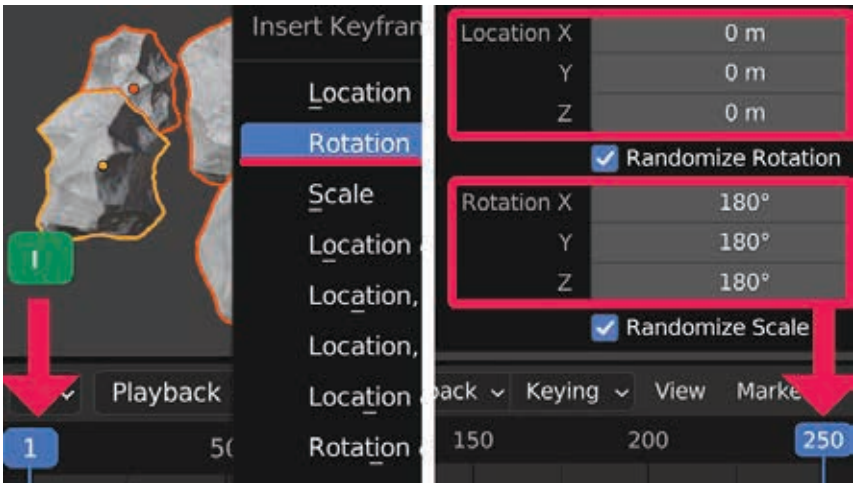


Select them all and **Set Origin to Geometry**.

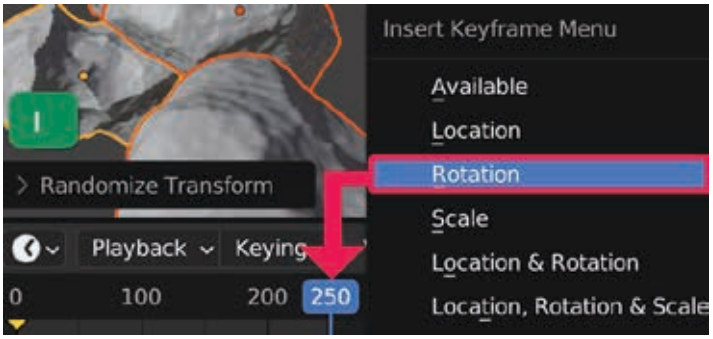


On the first frame, set a **Rotation** keyframe for all the rocks.

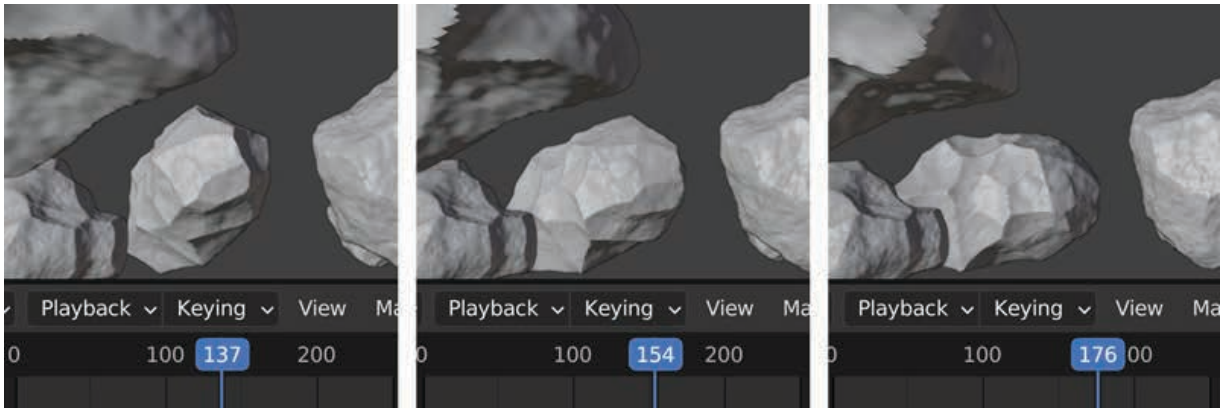
Go to the last frame and open the **Randomize Transform** panel. Uncheck **Location** and enter the maximum value in all the rotation fields.



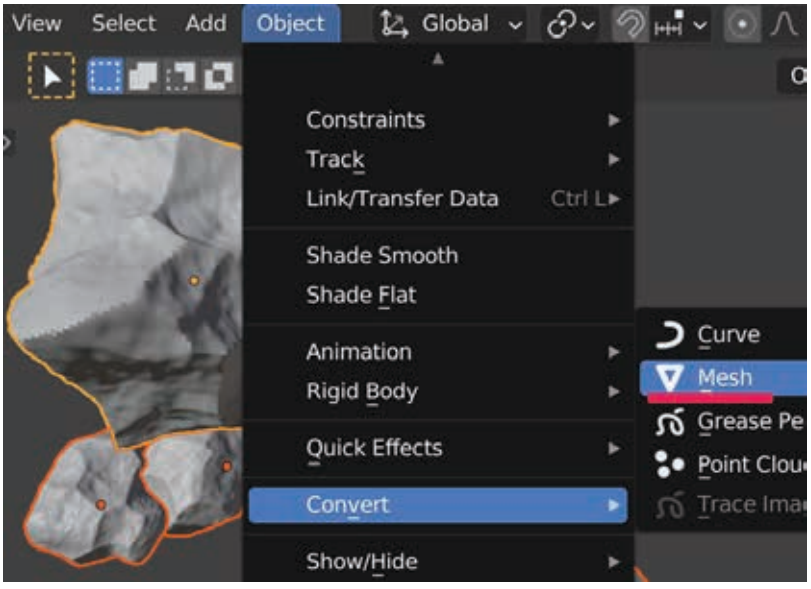
Set a rotation keyframe.



Now they all rotate slowly and randomly.

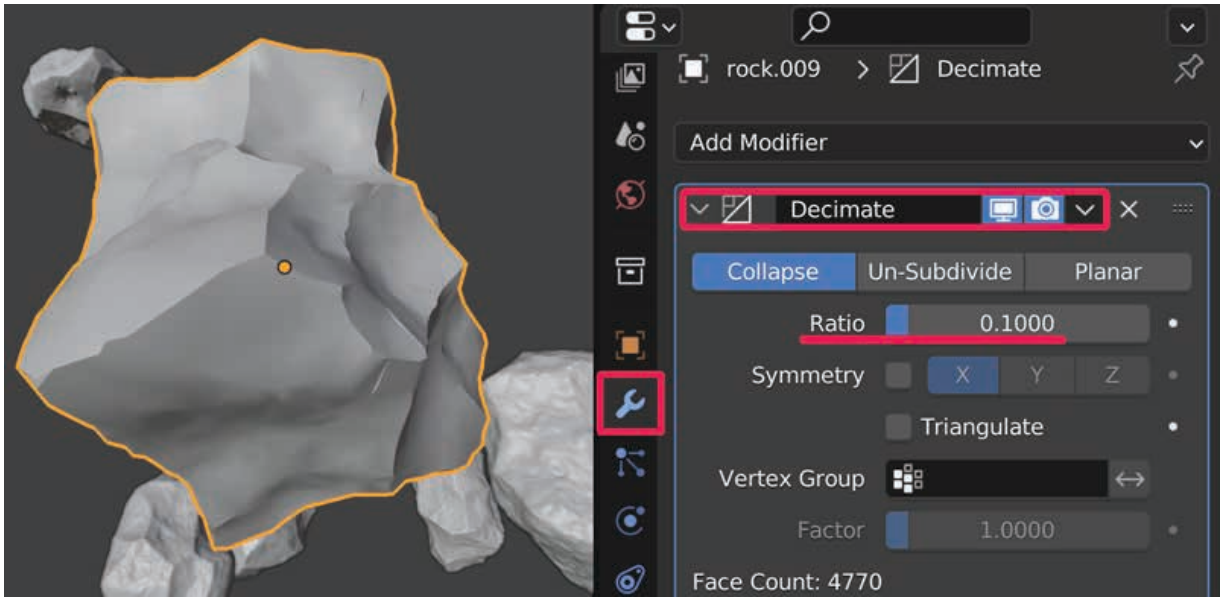


To collapse all the modifiers at once, convert all of the rocks to mesh.

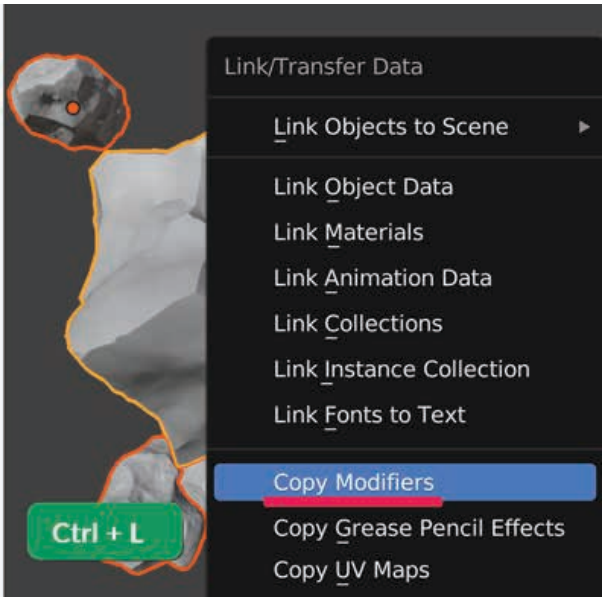


Click here to return to the index.

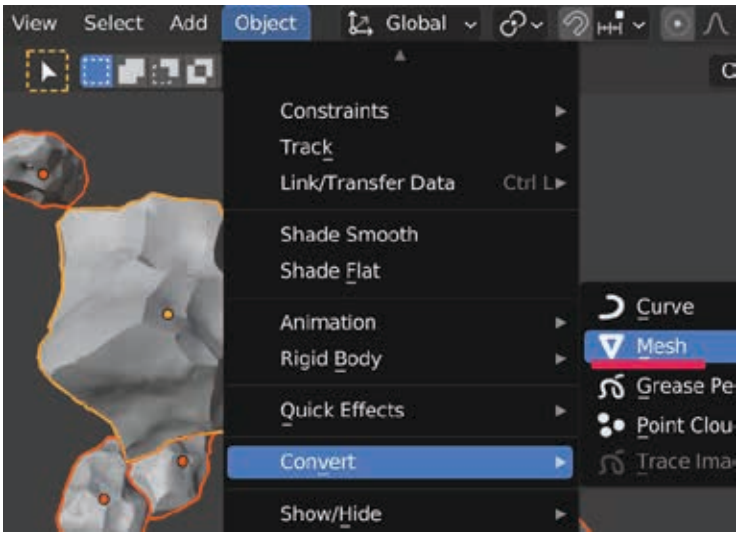
Add a Decimate modifier to one rock and set it to a low value like 0.1.



Select all the asteroids and select the one with the **Decimate modifier** again to make it active. Then **press Ctrl+L** and choose modifiers so they all share that Decimate modifier.



Convert them all to **Mesh** again to apply the Decimate modifiers.



SIMULATIONS, HAIR AND PARTICLES

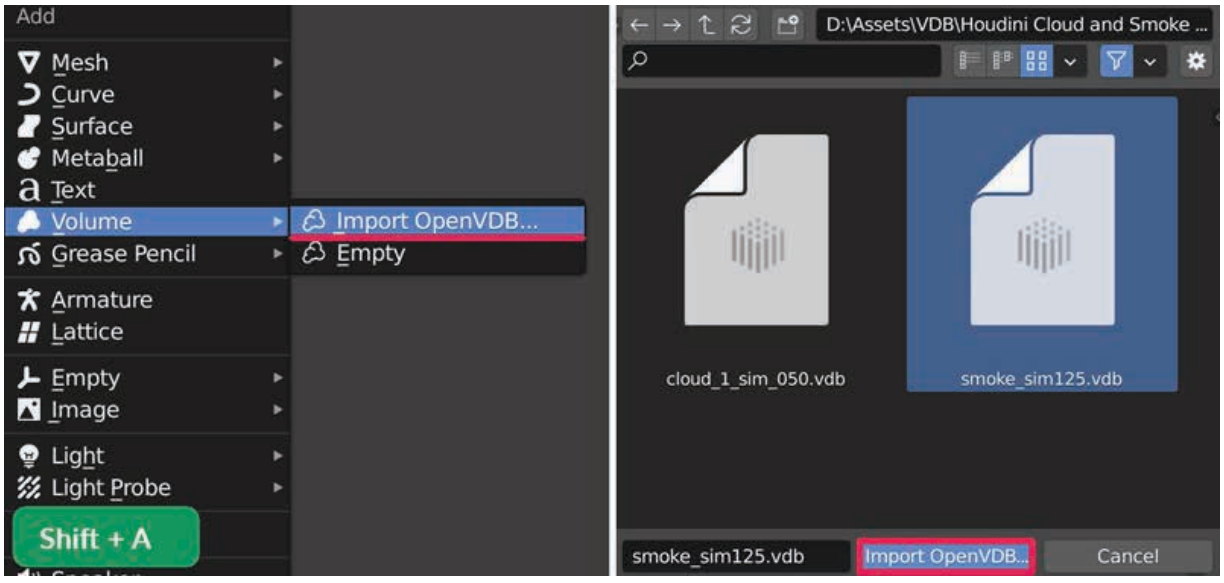
OpenVDB Quickstart

OpenVDB is a file format for animated simulations, that you can easily import into Blender for awesome results, without having to bake simulations. To get started, download an OpenVDB file. You can find some free ones here:

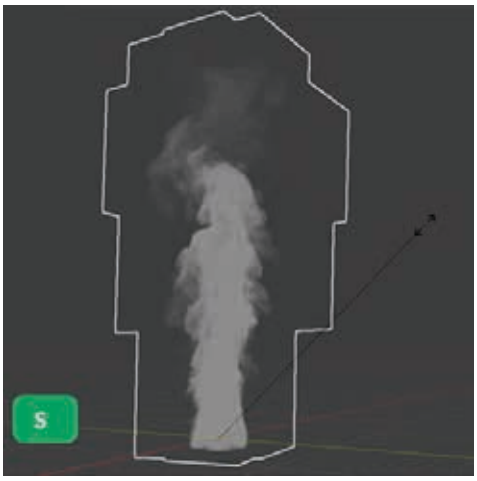
https://jangafx.com/software/embergen/download/free-vdb-animations/ (explosions!)
https://www.openvdb.org/download/ (explosion, fire and smoke)
https://gumroad.com/l/vdbsmoke (explosions)

Add some light, like a **Sun** lamp.

Press Shift+A and choose **Volume** and Import the OpenVDB file. Navigate to the OpenVDB file and **click on Import**.

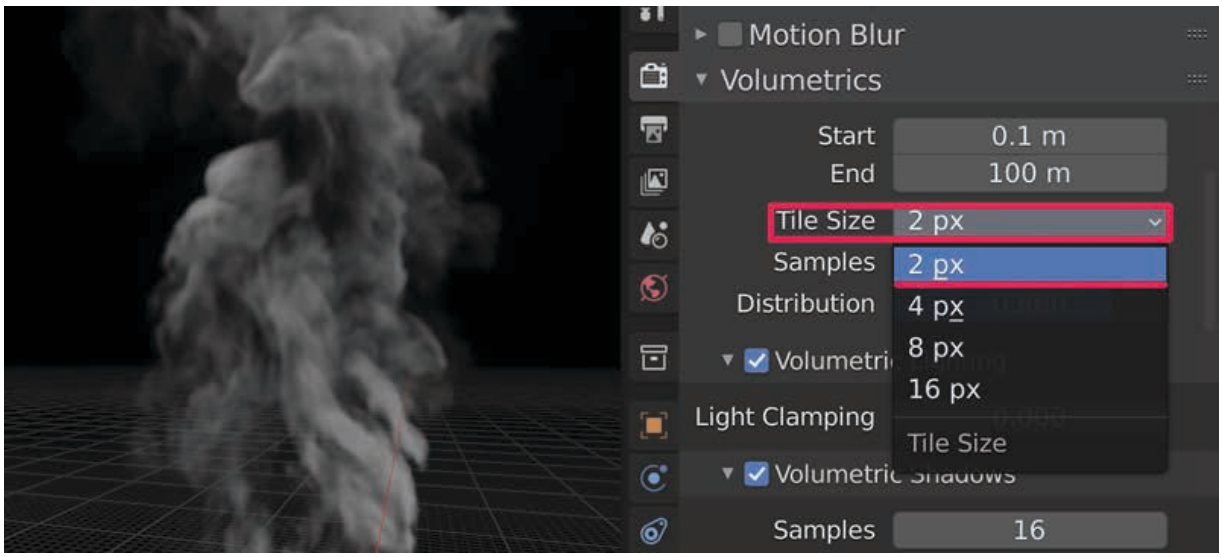


If you don't see anything, you may need to scale the object down.

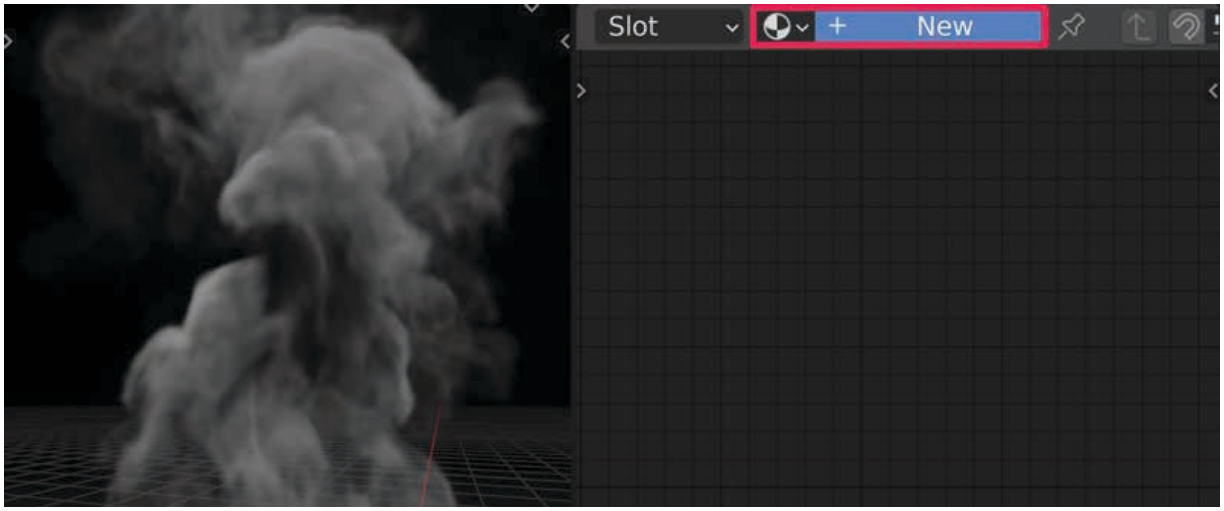


And turn on **Volumetric Shadows**.

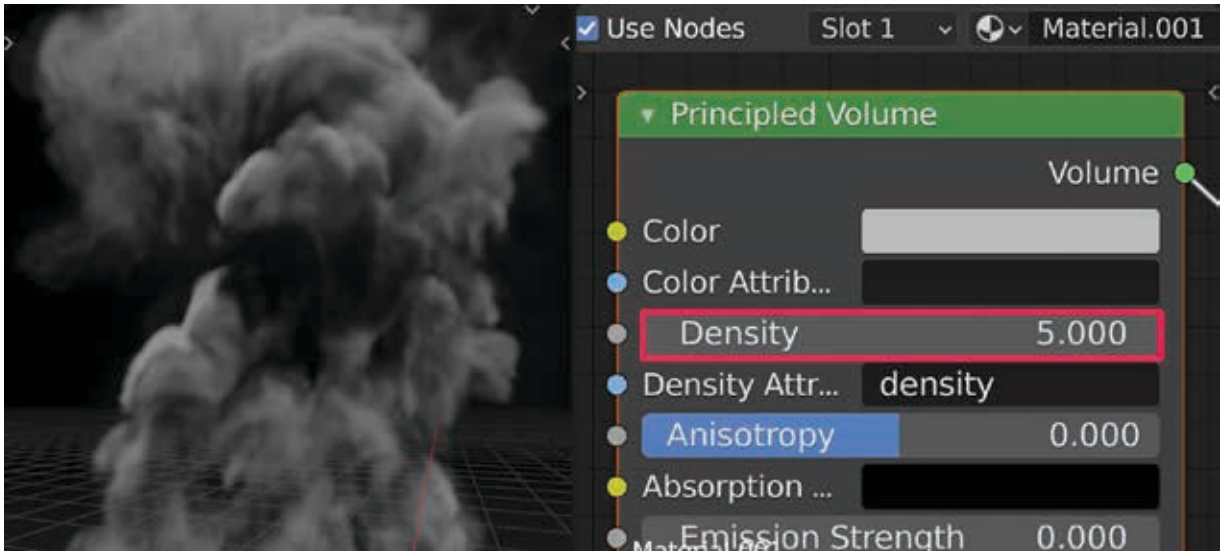
And lower the **Tile Size** for more detail.



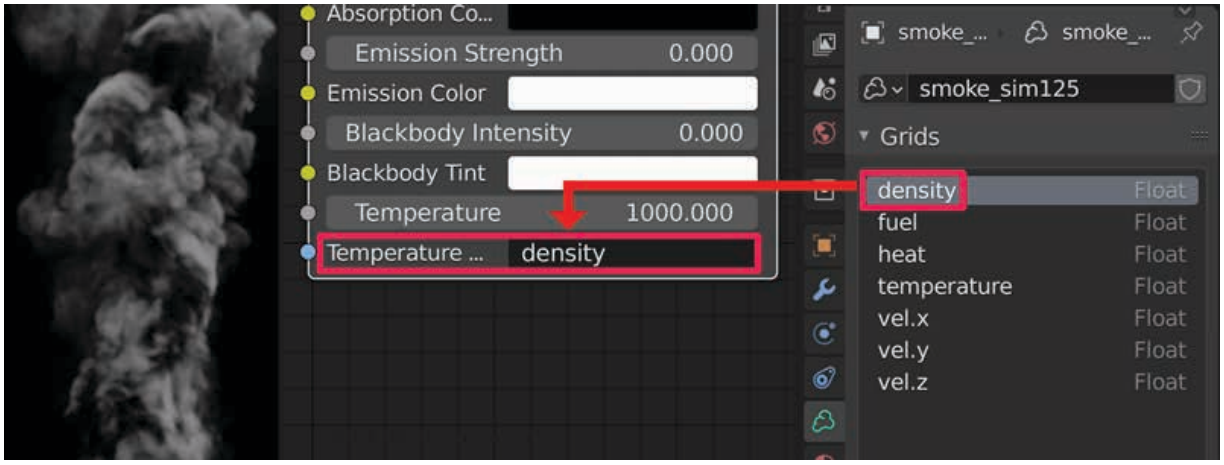
With the OpenVDB object selected, create a new material by **clicking on New** in the Shader Editor window.



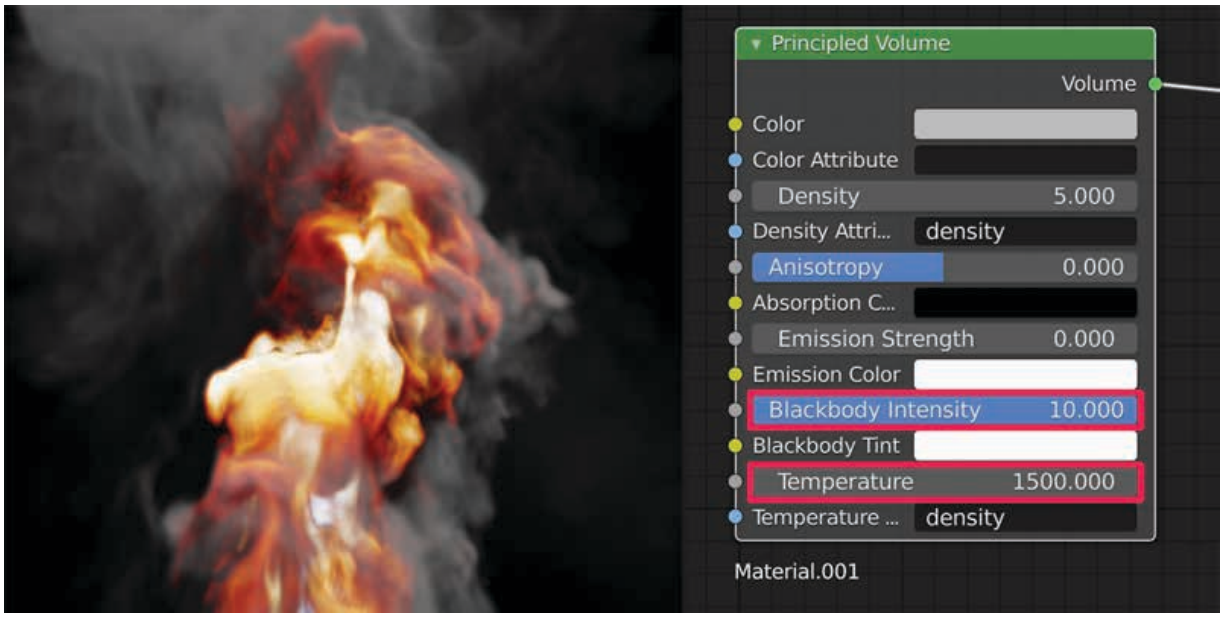
Use a **Principled Volume** node instead of the default Principled BSDF. Increase the **Density**.



Change the name in the Temperature Attribute in the shader to what it is in the Volume Grids panel.

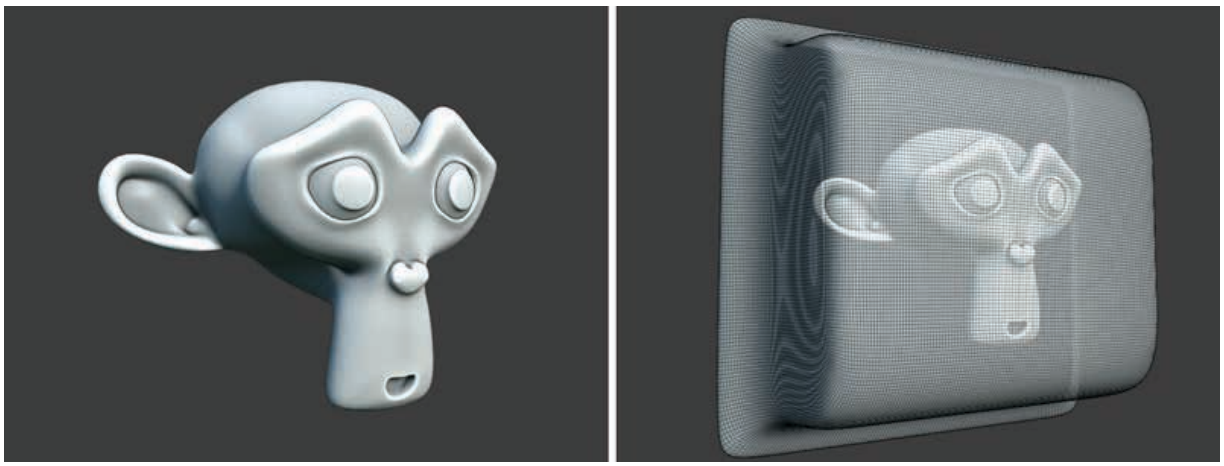


Increase the **Blackbody intensity** and the **Temperature**.

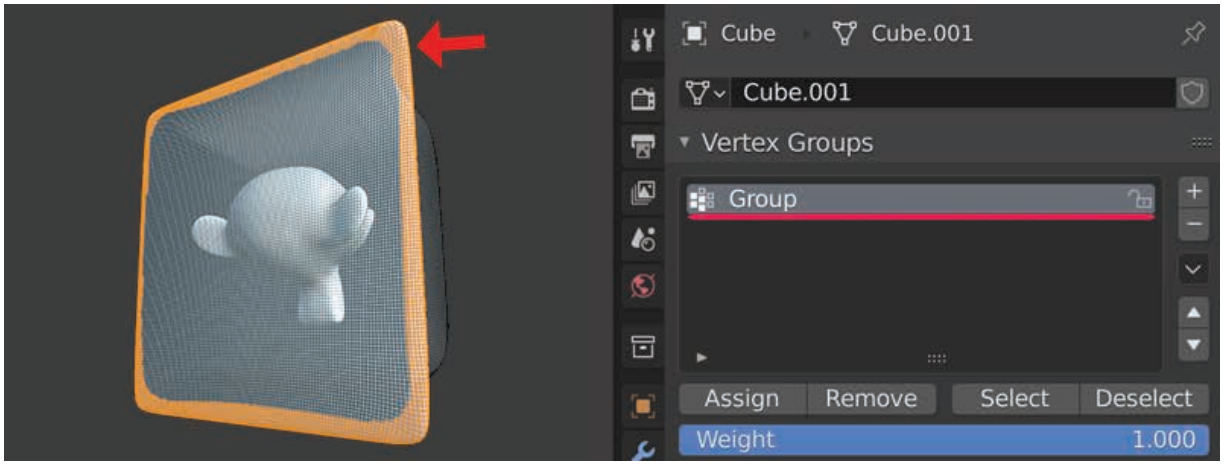


Vacuum Packing Objects using Cloth Pressure

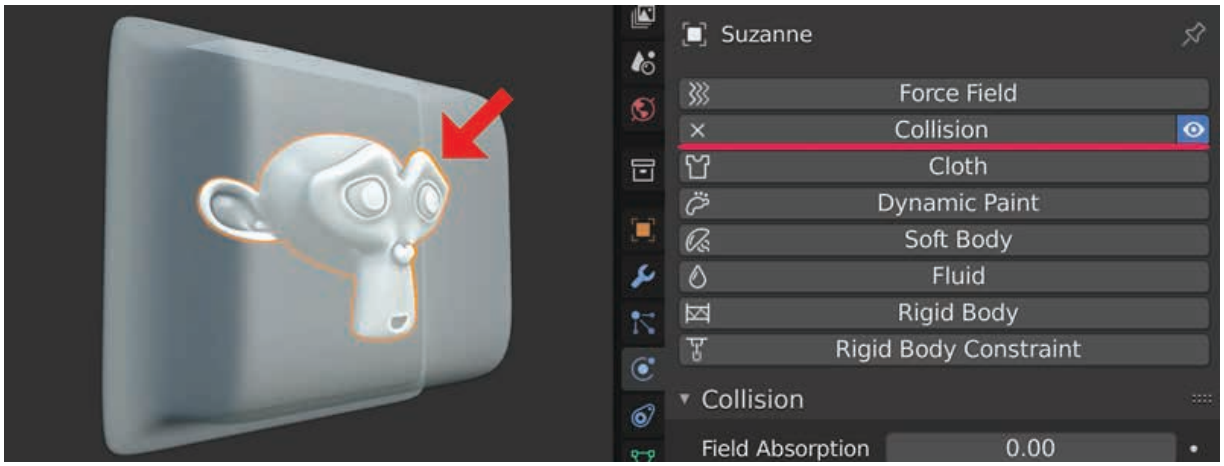
Create an object with enough subdivisions around the thing you want to vacuum pack.



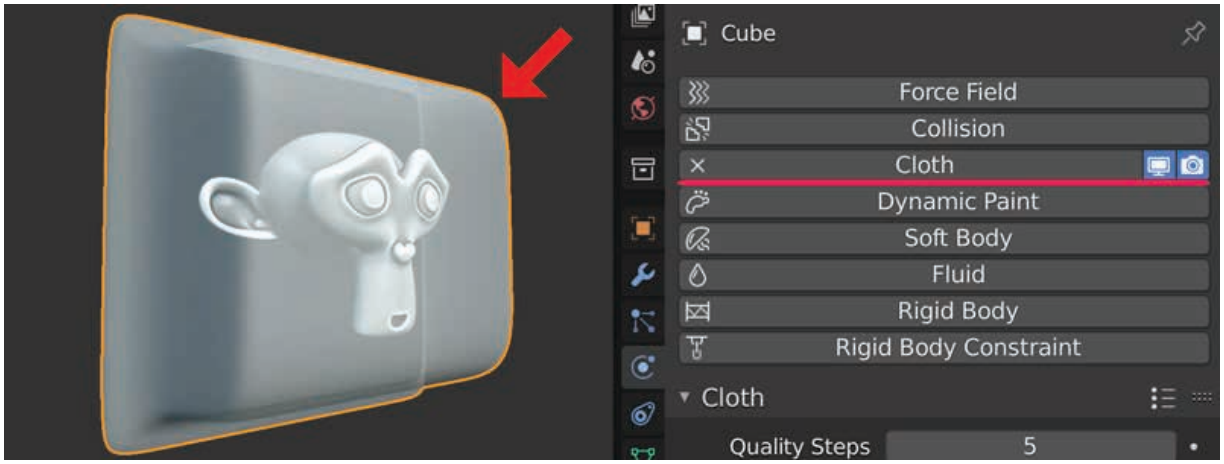
Select a rim of vertices. I like to do this kind of selection using **Circle select** which is activated by pressing C in Edit Mode and **X-ray viewport mode** (press Z and move the mouse cursor left to activate this mode). Add the vertices to a Vertex group.



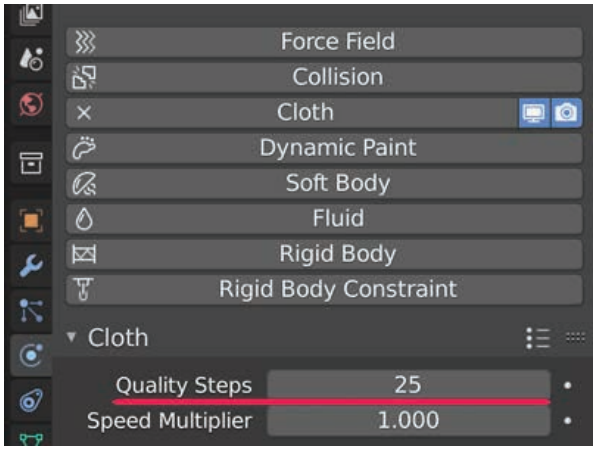
Enable **Collision** for the monkey.



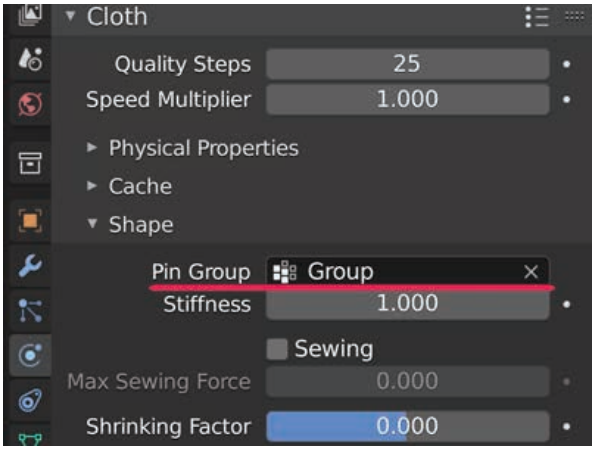
Turn on **Cloth** for the wrapping object.



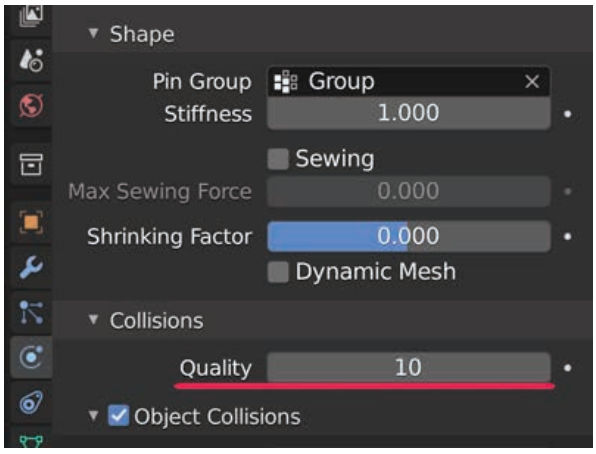
Increase the **Quality Steps** to 25.



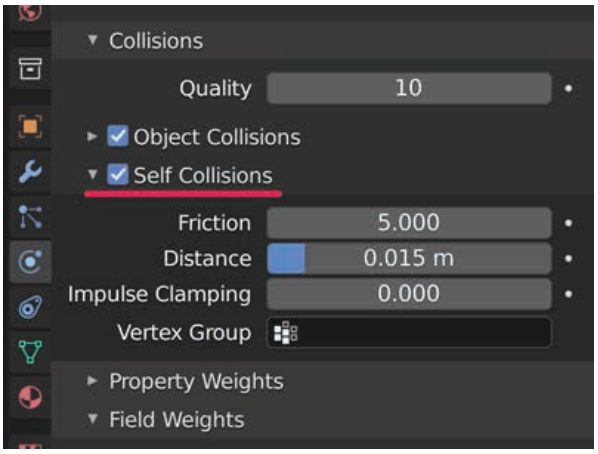
Add the Vertex Group to the **Pin Group**.



Increase the Collisions **Quality** to 10.

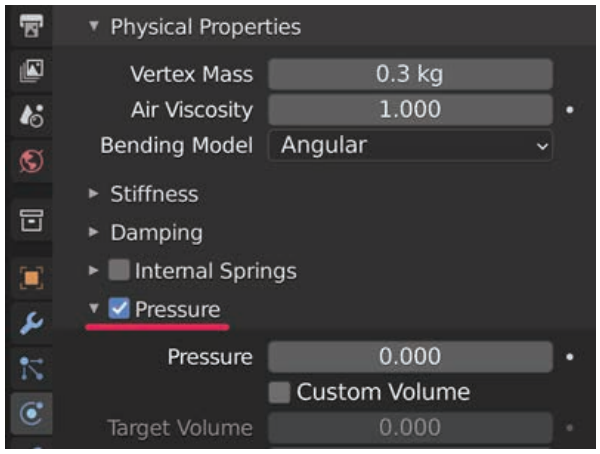


Turn on **Self Collisions**.

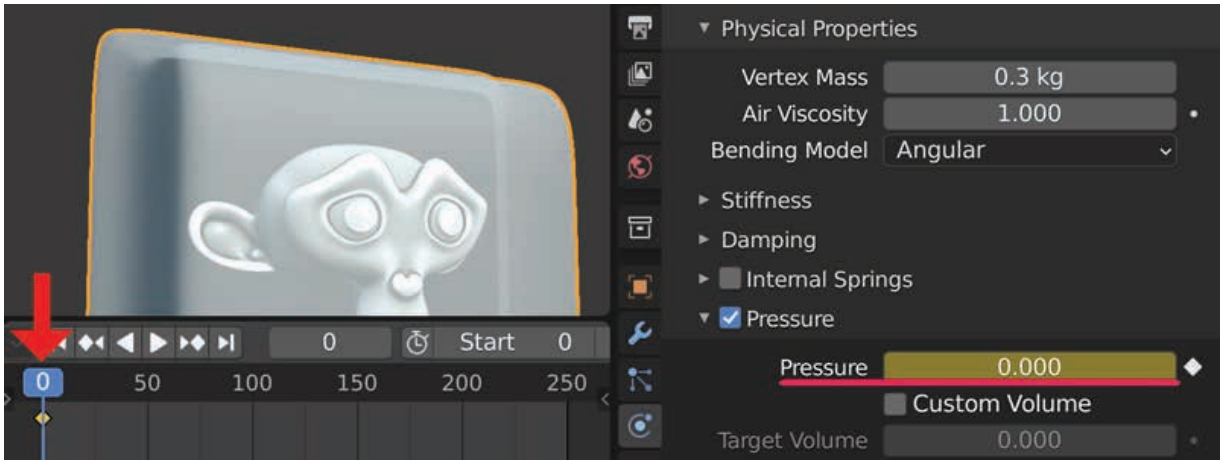


Turn OFF **Gravity**.

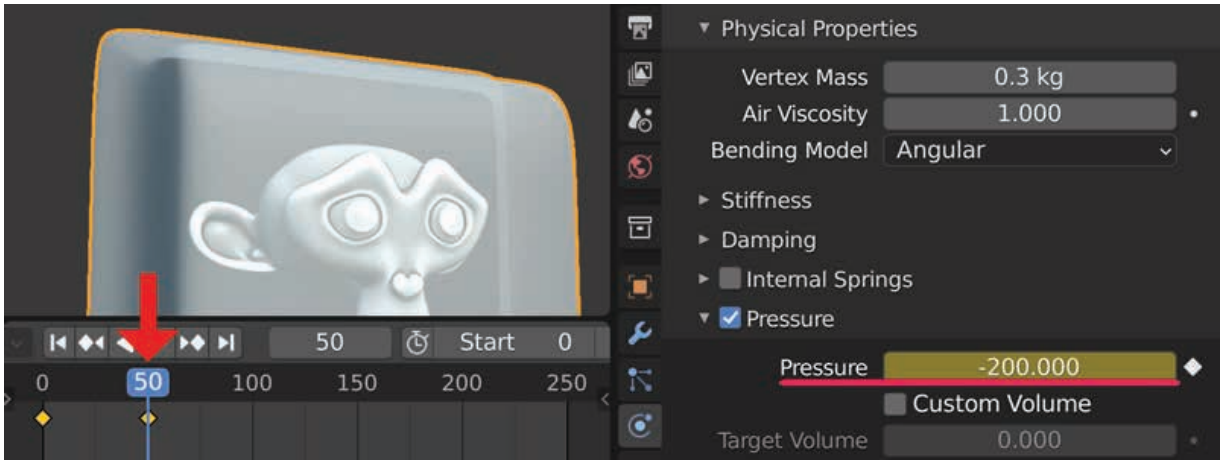
Turn ON **Pressure**.



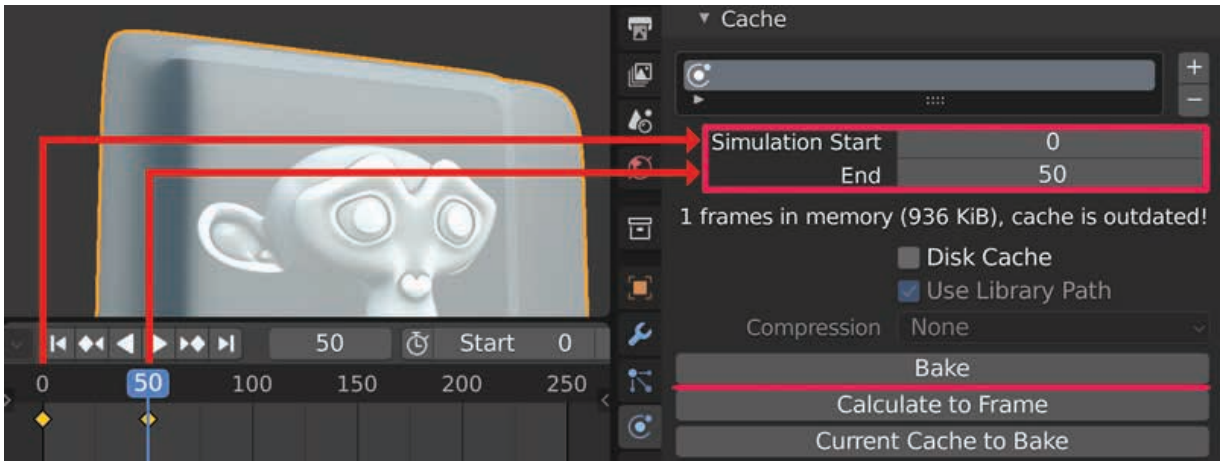
Now at **frame 0** set a **Pressure** keyframe of 0.



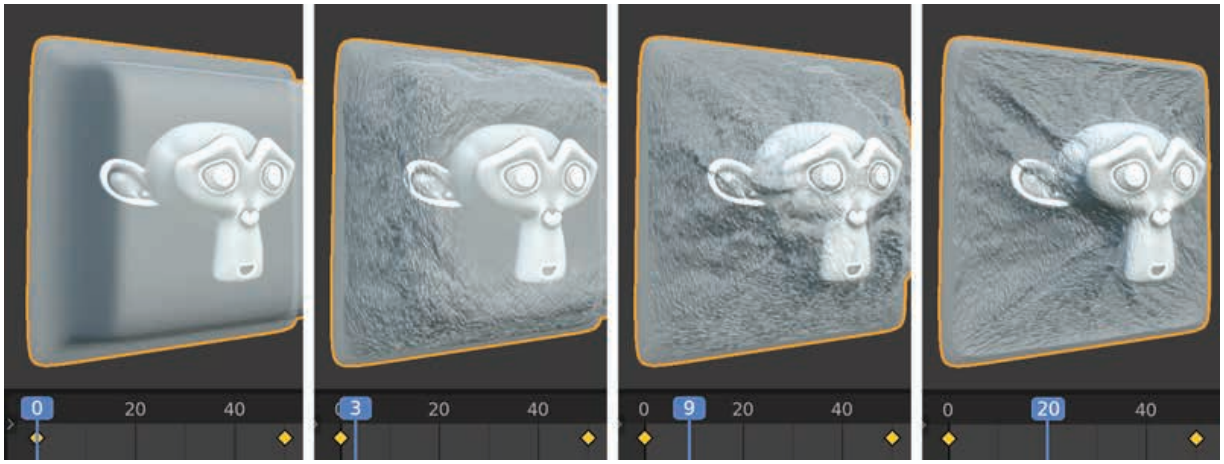
Then at a later frame set a Pressure keyframe of -100 or even -200.



Limit the **Cache** length to what is needed and bake the simulation.



Press the Spacebar to see the result.



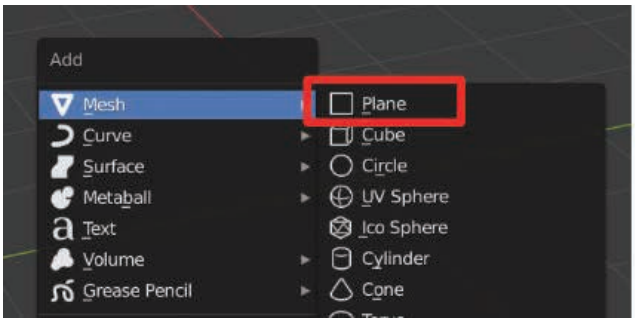
On the right is the result, rendered with a glass material in Cycles.

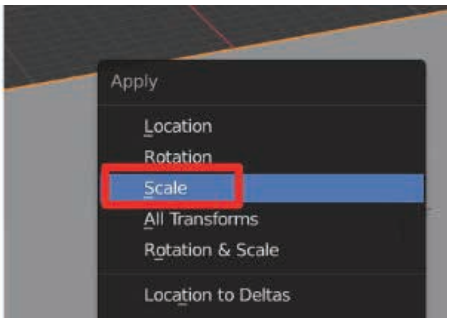
Once you've chosen the best looking frame on the Timeline, you can **Apply** the Cloth modifier.

Boat Floating on Ocean Waves

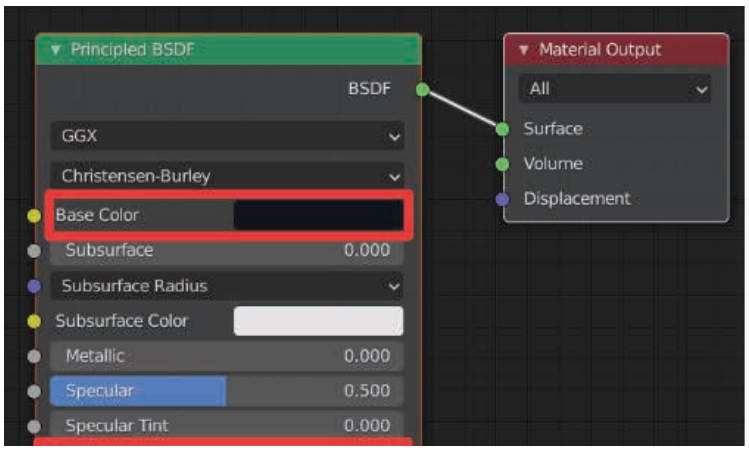
Have you ever wanted to create a 3D scene with an object that floats on the water, like a boat? This tutorial explains the basics of such a scene.

First we make a simple Ocean. **Press Shift+A** and add a **Mesh>Plane**. **Press F2** to open the renaming field and rename it to "Ocean" to avoid later confusion. **Press S+20** to scale it up and then **press Ctrl+A** and choose **Scale** to apply the scale.

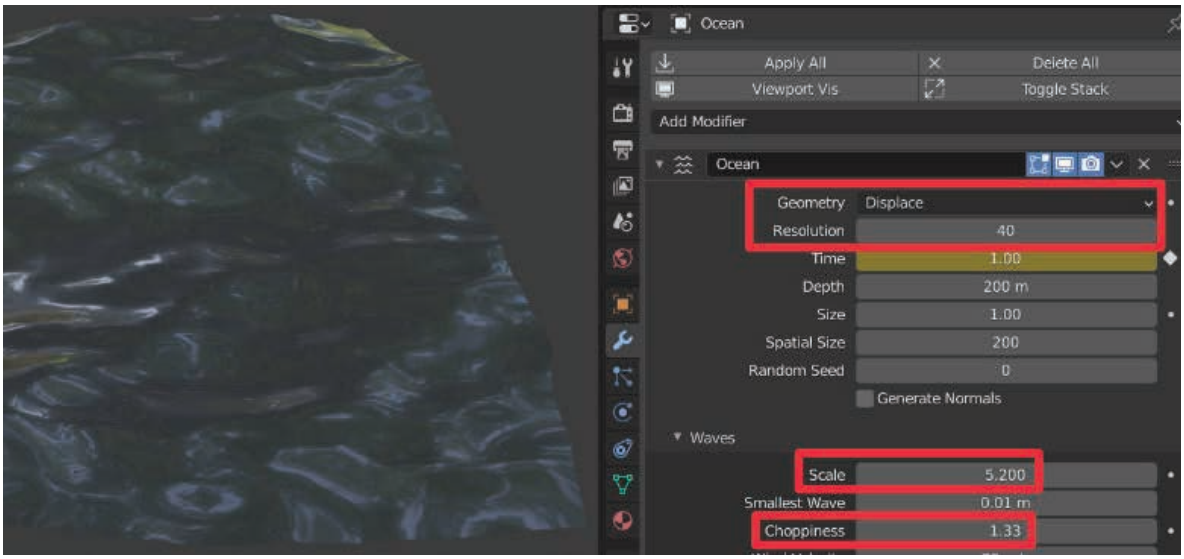




Press Tab to go to Edit Mode and **right-click** on the Ocean plane, and choose **Subdivide**. **Press Shift+R five times** to increase the subdivisions. Give the plane a new material and make it darker by changing the Base Color and reflective by reducing the Roughness. This makes it easier to see the waves. **Right-click** on the plane and set it to **Shade Smooth**.



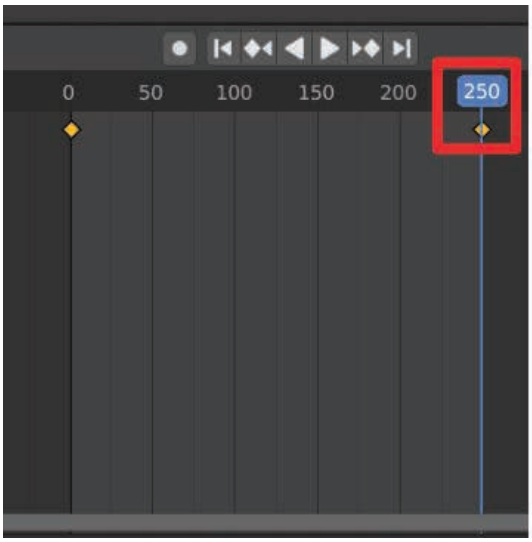
Add an **Ocean Modifier** to it. Set it to **Displace**. Increase the **Resolution** to 30 (or 40 if your computer can handle it). Increase the wave **Scale** and **Choppiness** to your liking.

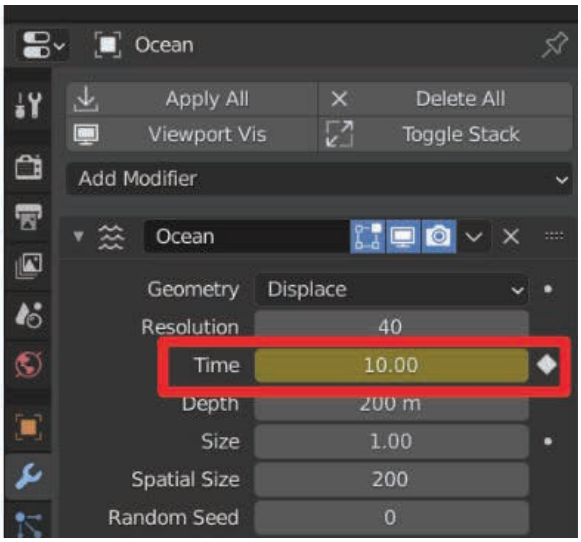


You might expect the ocean waves to start playing when you **press the Spacebar**, but that's not the case. Not until you add some keyframes. The value you want to keyframe is "Time".

So on frame 1 hover the mouse cursor over the **Time** value and **press i** to set a keyframe. Or click on the diamond shaped dot next it.

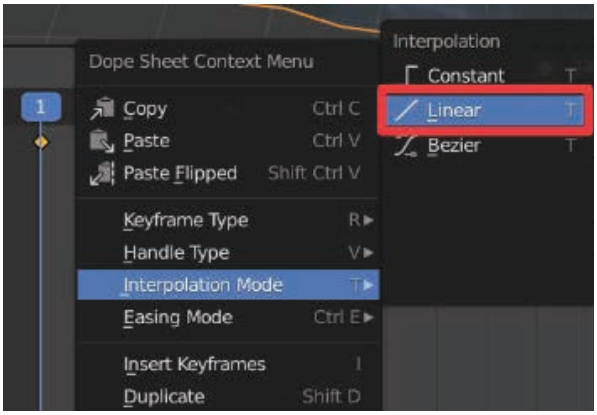
Next go to the last frame on the Timeline and set a different value for **Time** (10, for example) and set another keyframe for that.





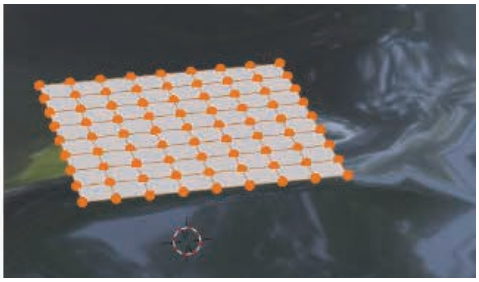
Now if you **press the Spacebar**, you'll notice that the ocean is animated. But it starts slow in the beginning, gains speed and then slows down again towards the end.

To fix this, select all the keyframes in the Timeline, **right-click and choose Interpolation Mode > Linear**.

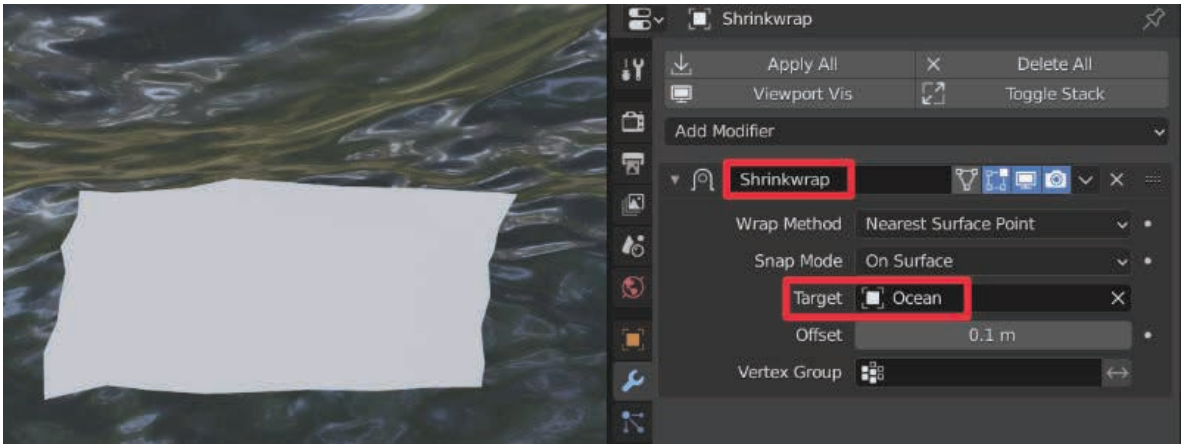


Press Shift+A to get the Add menu and add another **Plane**. You may need to move it up a bit to be able to see it by **pressing G+Z** and moving the mouse. **Press F2** and rename this one "Shrinkwrap".

Press Tab to go into Edit Mode and **right-click on the Plane**, then choose **Subdivide**. **Press Shift+R** a couple of times to repeat the subdividing.

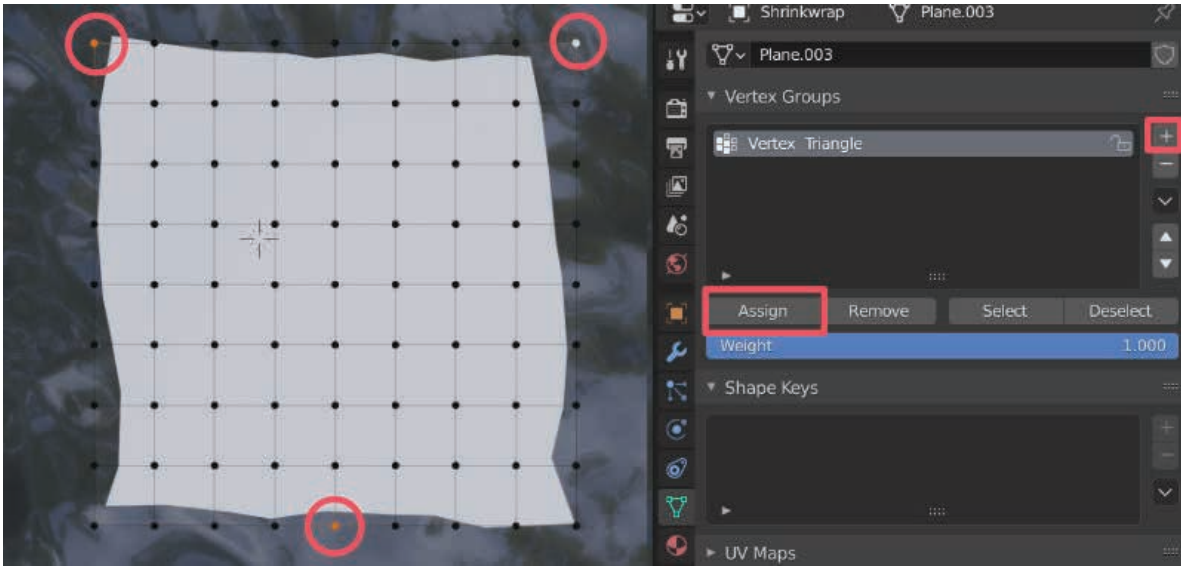


Press Tab again to exit Edit Mode and add a **Shrinkwrap modifier** to the new Plane. As the **Target** choose the "Ocean" object. Now the Plane dances along with the waves. It doesn't look pretty, but that's okay. We won't see the Plane in the render anyway, it's only there to help animate the boat.



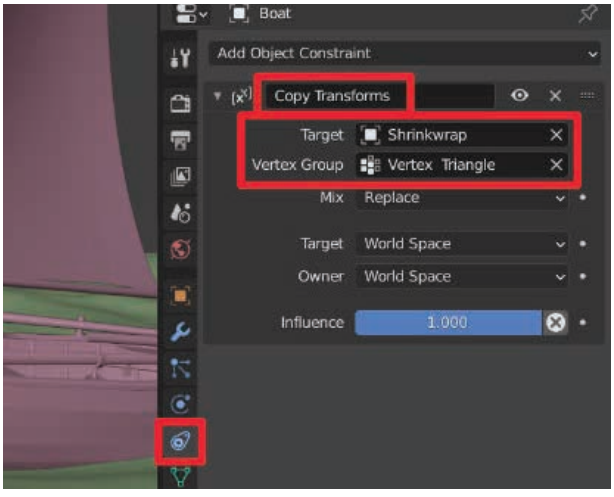
Now back Edit Mode, **select a triangle of three vertices**. It's very important that it's exactly **threevertices** - only two or four or more vertices and your boat will only bob up and down, without all the subtle side to side motion that it would realistically have.

Create a new **Vertex Group** by clicking on the **+ icon** and **Assign** all the vertices to it. If you want name it something clear like "Vertex Triangle".

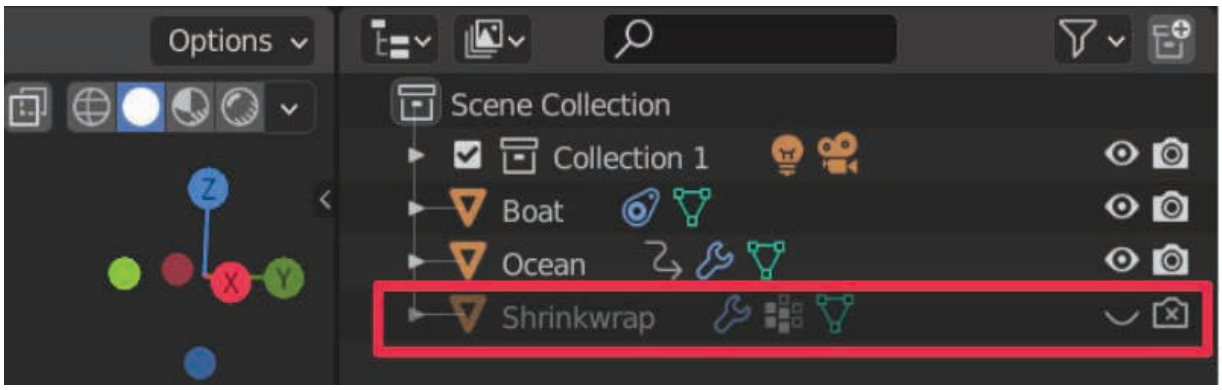


Now add your little boat (or whatever object you want to have float on the water) to the scene by Appending or Linking it from the file that contains your boat model (in my case I purchased a nice model made by carlcapu9 on Sketchfab).

In the **Constraints** tab of that add a **Copy Transforms constraint** and as the **Target** choose the "Shrinkwrap" Plane.
As the **Vertex Group** choose the **Group** (which we named "Vertex Triangle").

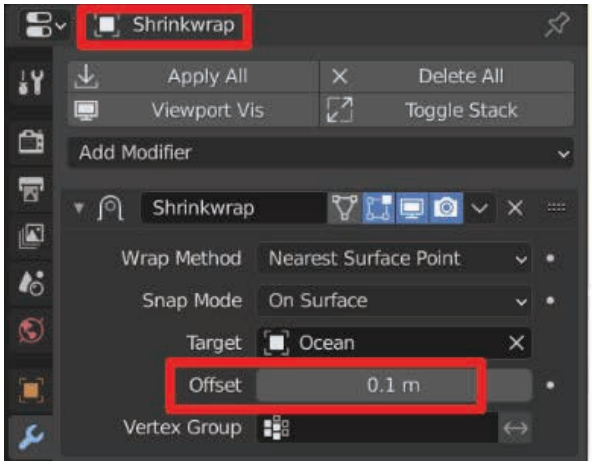


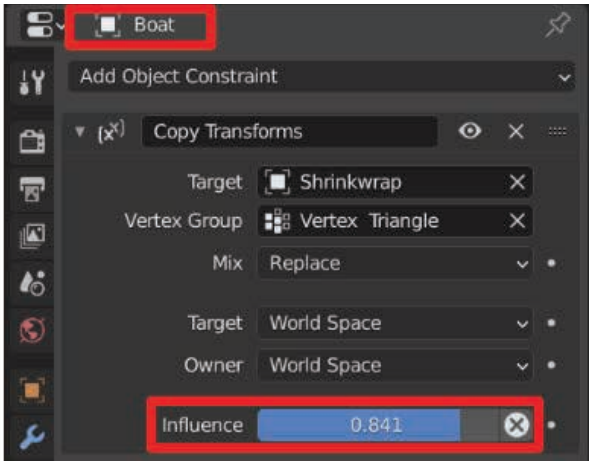
Now the boat follows the Plane. You can **turn off Render and Viewport visibility** in the Outliner for the Shrinkwrap Plane so it is no longer visible



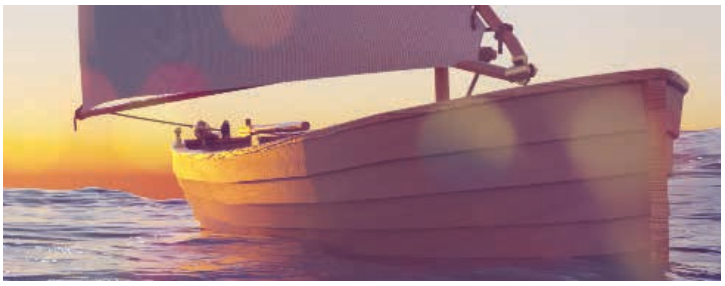
So far so good but… why is our little boat shaking so much? Is it afraid of giant sea monsters? In any case, this is easy to solve. In the **Outliner**, select the Shrinkwrap plane and in the Shrinkwrap modifier, **set Offset to 0.1**.

Now the boat moves smoothly without shaking. You can further finetune the amount of movement by the boat with the **Influence slider** of the Copy Transform Object Constraint.



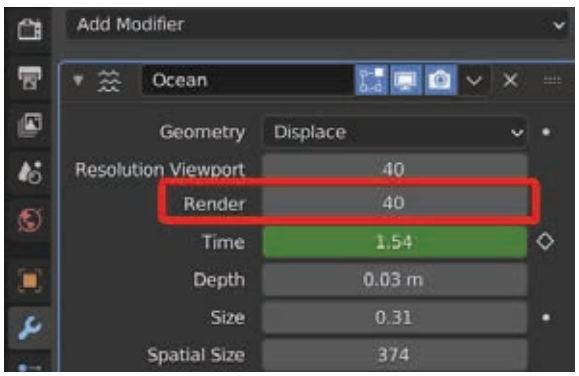


Now we've got our boat gently (or not so gently, depending on your Ocean settings) bobbing on the waves.



Troubleshooting

In case your waves don't show up in the render, make sure the **Render** value in your modifier is at least as high as, or higher than, the **Resolution Viewport** value.

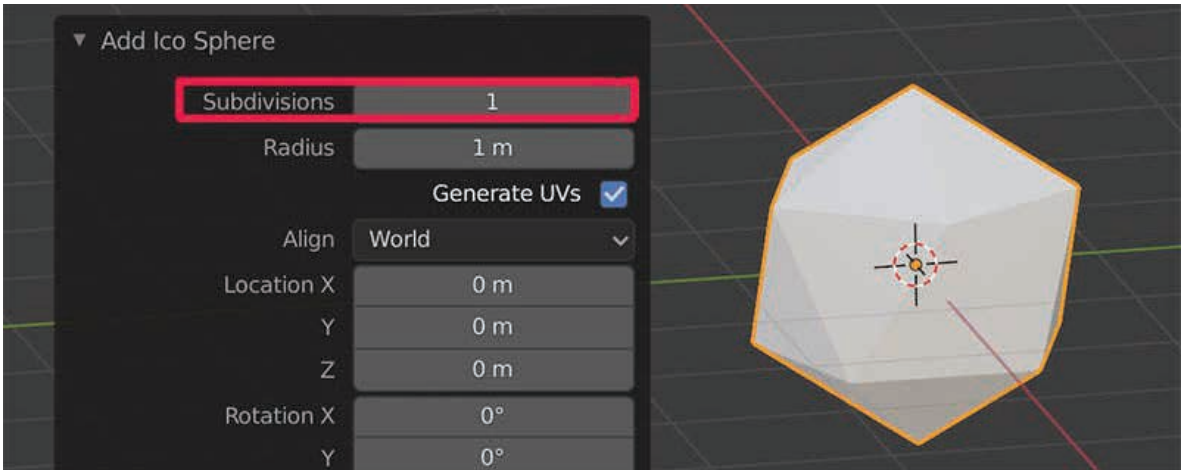


Rain with Dynamic Paint

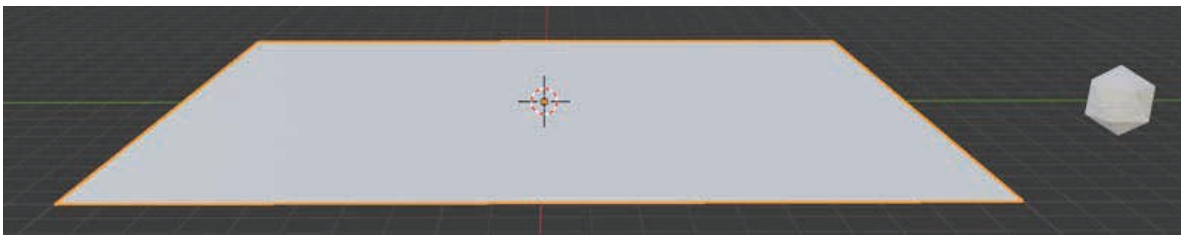
Have you ever created **raindrops** in Blender?

We can achieve this using a combination of Dynamic Paint and a Particle System. If that sounds complicated, don't worry - I explain all the steps below so you can just follow along.

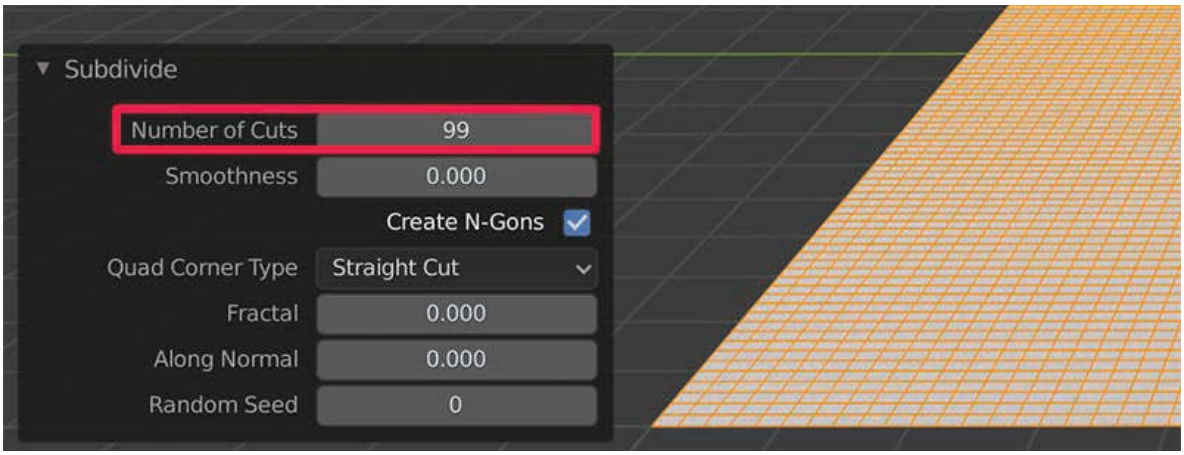
Press Shift+A and add an **Icosphere** with the lowest resolution, 1. This will be our "rain drop". Move it to the side along the Y-axis by **pressing G+Y** and moving the mouse.



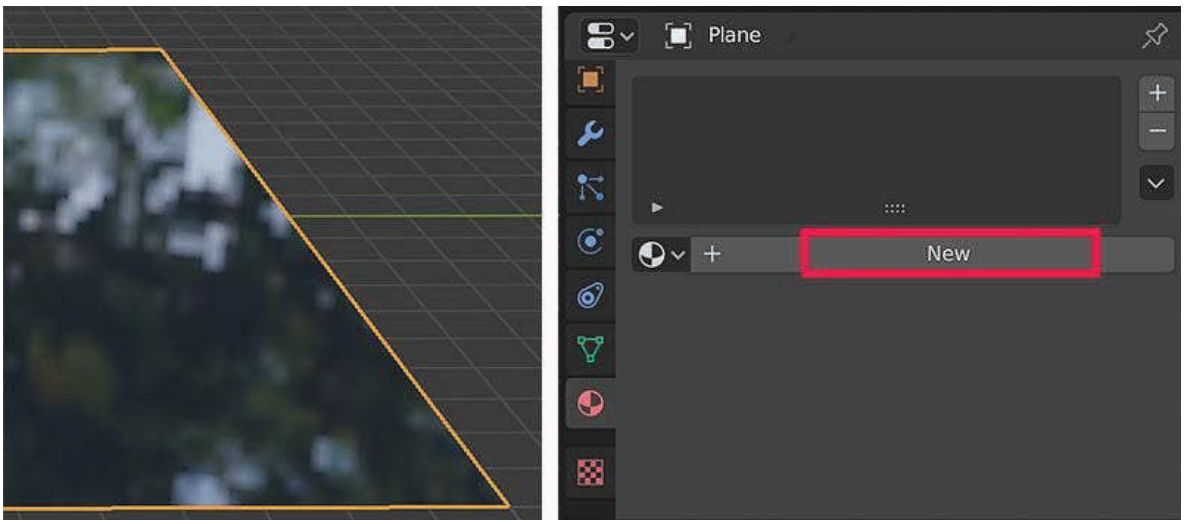
Press Shift+A again and add a **Plane**. Scale it up by **pressing S+10**.



Press TAB to go into Edit Mode. **Right-click** and choose **Subdivide**. In the Subdivide option panel that you can expand in the lower left corner, set **Number of Cuts** to **99**. You need this detail so that you can see the waves created by the rain. If you have an older computer and you think that these are too many subdivisions, try a lower number like 49.

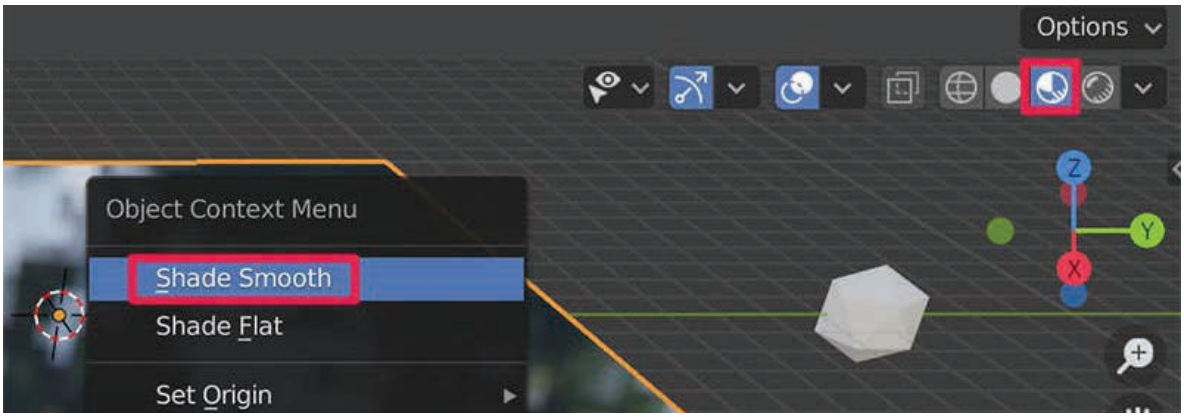


Press TAB to go back to Object Mode. To make the waves more visible, select the Plane and in the material tab click on **New**. make the material more reflective by setting the **Base Color** of the material to a dark color, decreasing **Roughness to 0** and increasing **Specular to 1**.

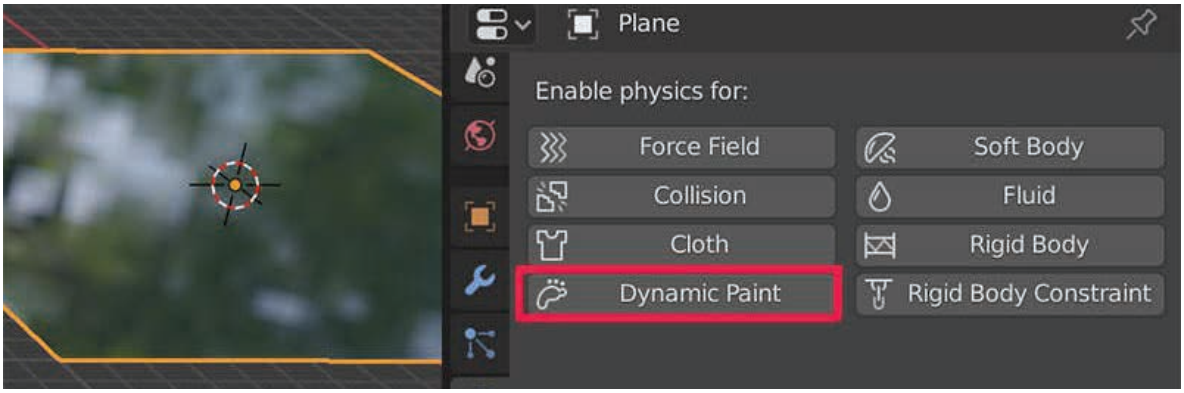


Set **Viewport Shading** to **Material Preview** so that you can see what the material looks like.

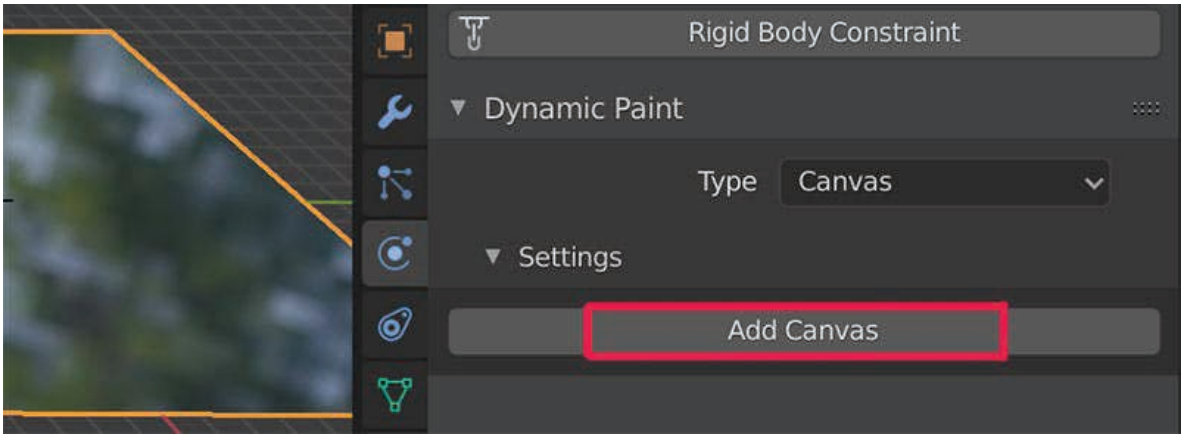
To do this you can click on the corresponding icon in the top right corner of the viewport, or **press Z** and choose **Material Preview** from the pie menu (For advanced Blender users: **press Z and move the mouse down** to do this quickly - this is the real benefit of pie menus, once you know them by heart).



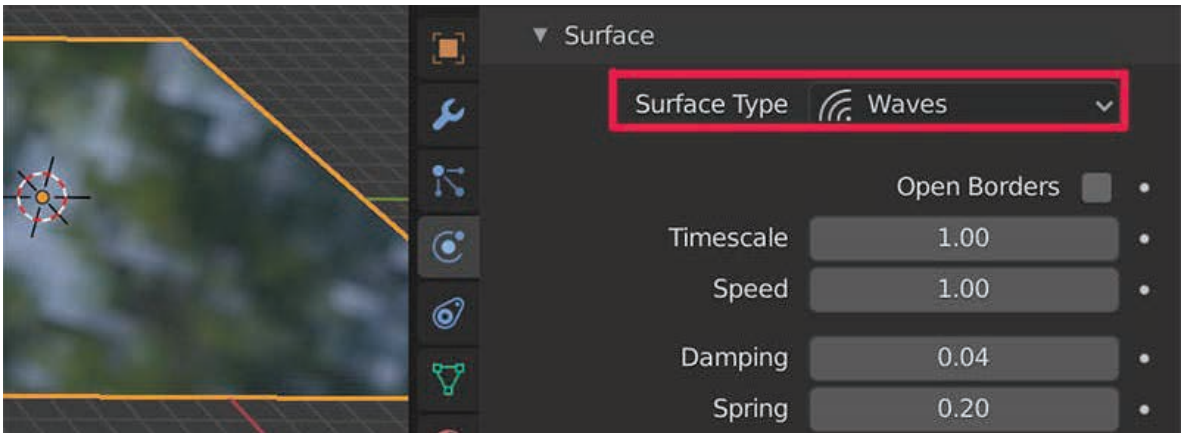
With the Plane selected click on **Dynamic Paint** to enable it.



Click on **Add Canvas**. Dynamic Paint has two types: **Canvas** (which receives the paint) and **Brush** (which applies the paint).

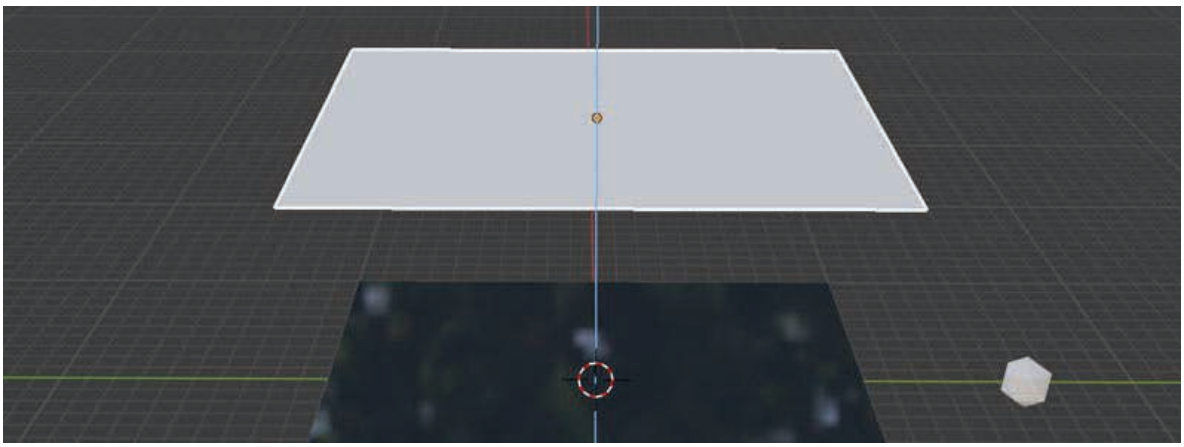


Set the **Surface Type** to **Waves**.

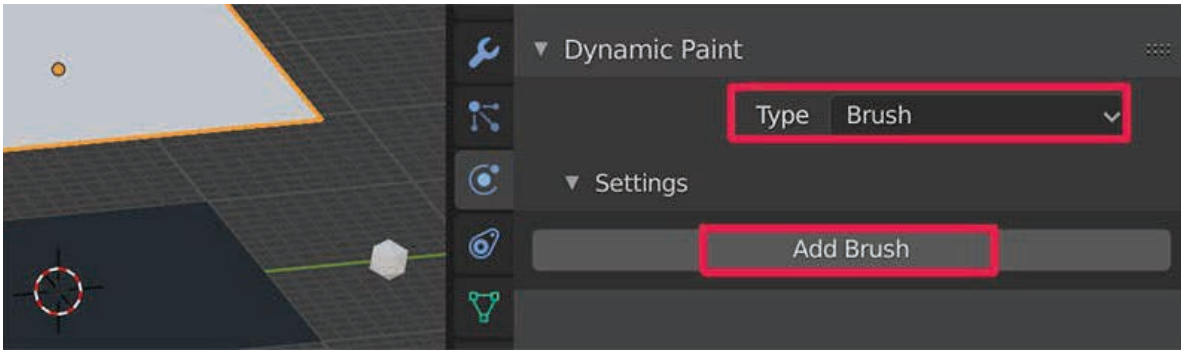


Press Shift+A and add a second **Plane**. Scale it up like before by **pressing S+10** and then **press G+Z** and move it up along the Z-axis.

We will use this to emit the rain from. Think of it as our rain cloud. You don't need to Subdivide this Plane nor give it a material.

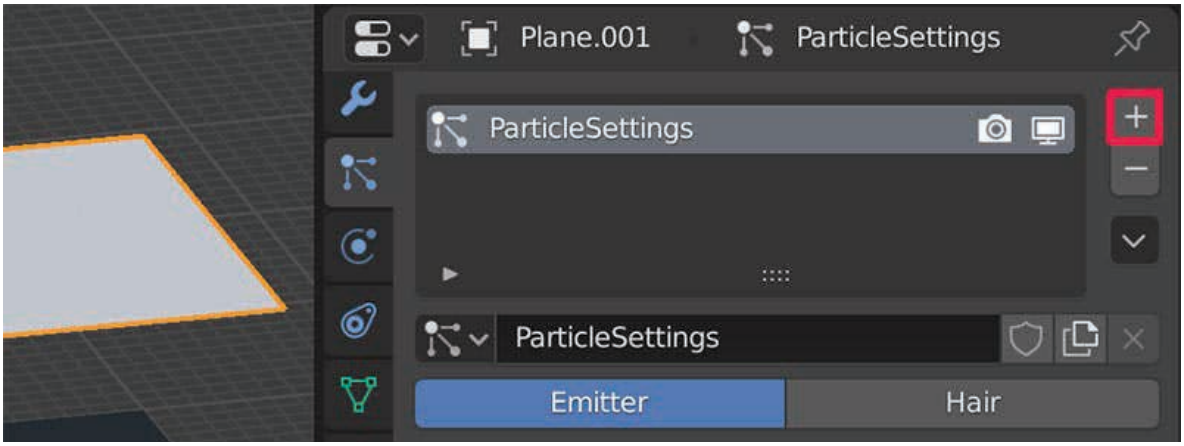


Enable **Dynamic Paint** for this Plane as well except in this case set it to **Brush**, then click on **Add Brush**.

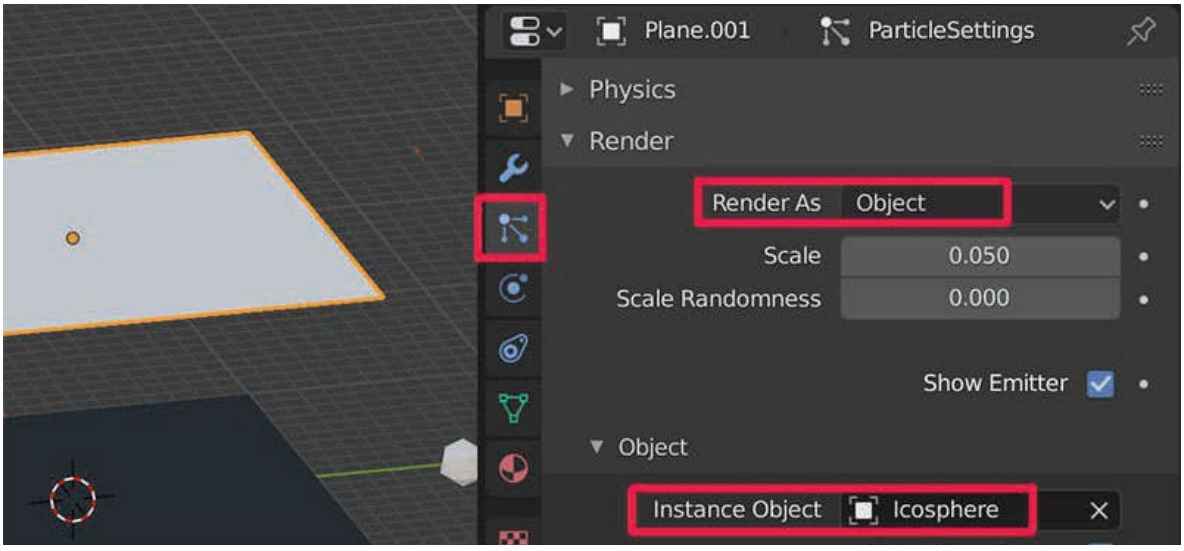


Set **Paint Source** to **ParticleSystem**. This means that the particles coming from this plane will act like a brush on the plane which is set to Canvas.

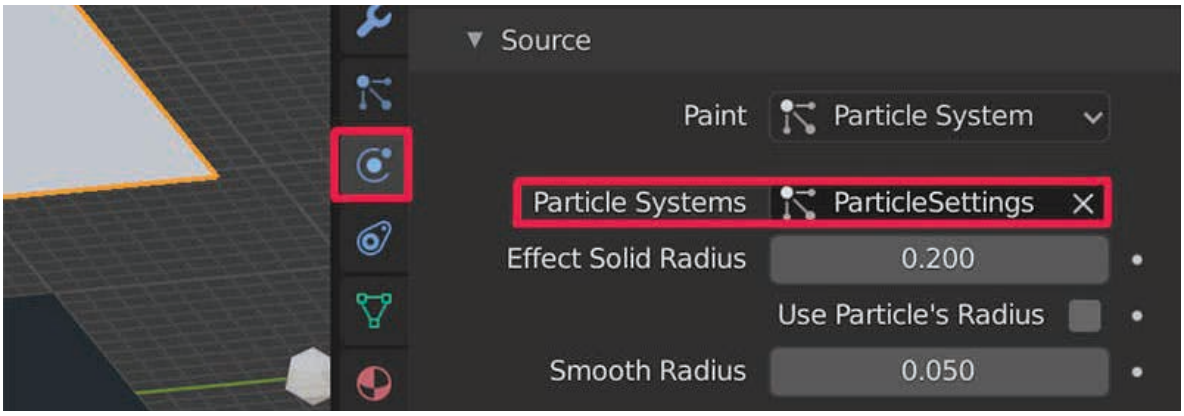
However, we don't have a Particle System yet, so add a **ParticleSystem** by clicking on the **+ icon** in the Particle System tab.



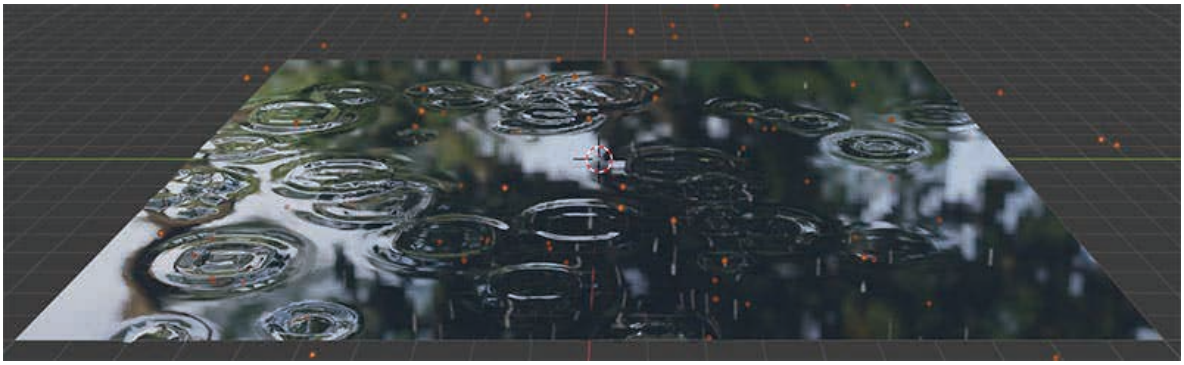
In the Render settings of the Particle System, set **Render As** to **Object** and set the **Instance Object** to the **Icosphere**. Now the Particle System uses the Icosphere as rain particles.



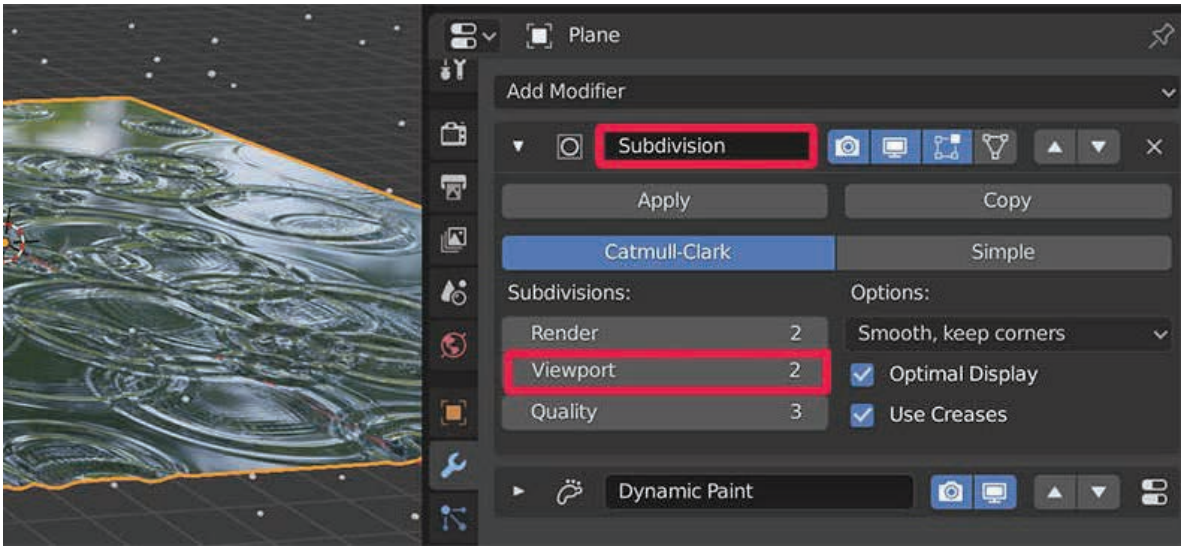
Back in Dynamic Paint, under **Source** you can now choose the **ParticleSystem**.



Press Spacebar to see the rain simulation.

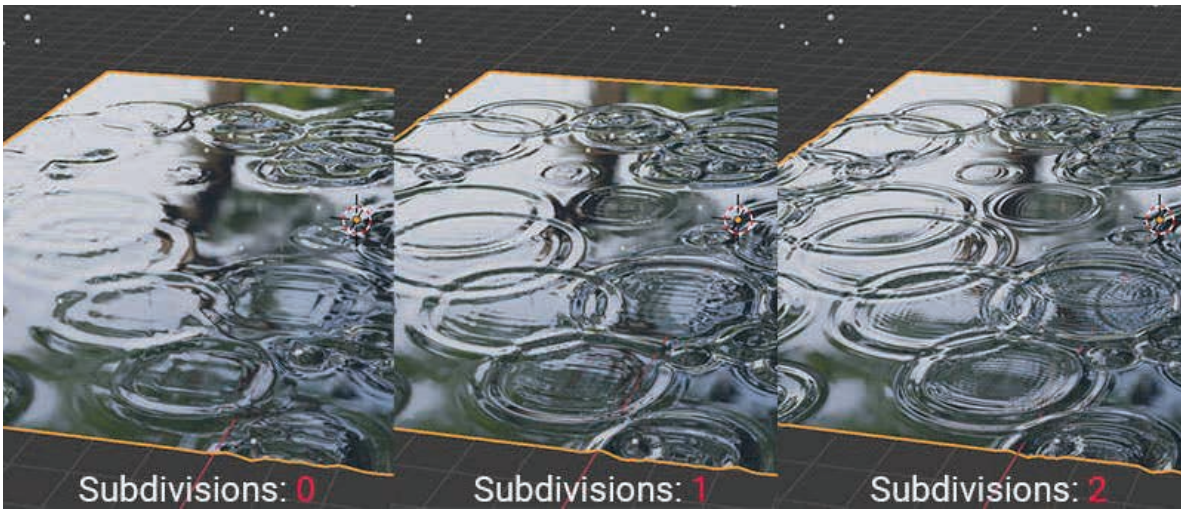


The more resolution the plane receiving the particles has, the more detailed the waves will be. Add a **Subdivision modifier** to the plane and increase the amount of **Subdivisions**. Place the Subdivision modifier above the Dynamic Paint modifier.

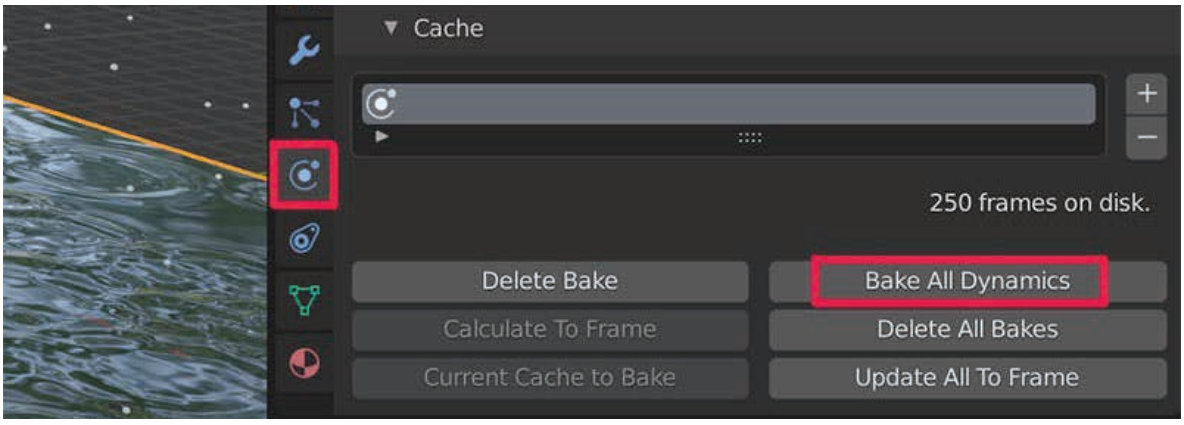


Press Shift and ←to go back to the beginning of the **Timeline** and **press Spacebar** to play the simulation each time you change a value.

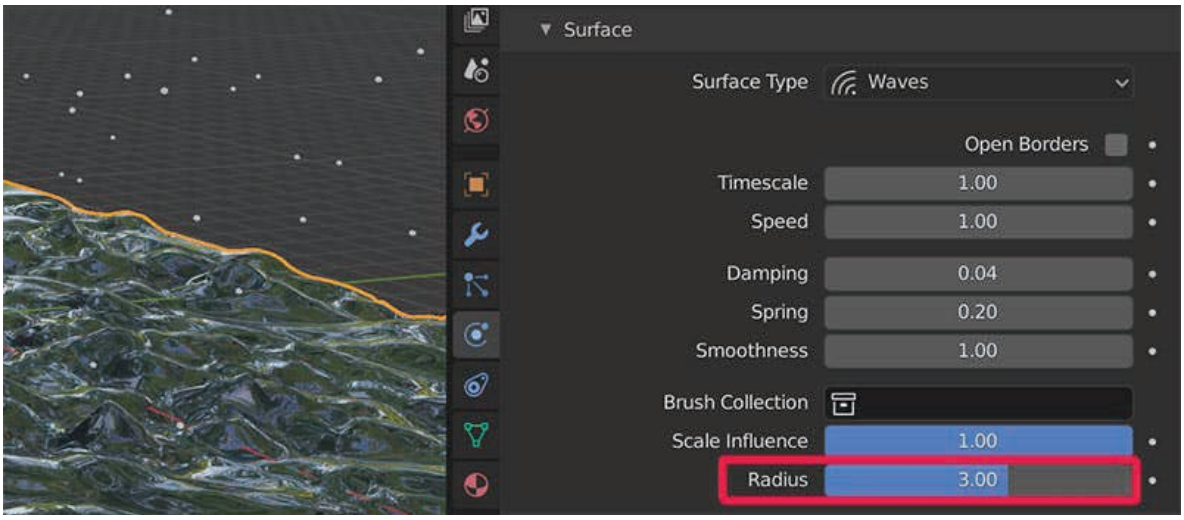
You will notice the difference the extra Subdivisions make in terms of the quality of the waves. This also increases the amount of calculations your computer has to perform, so depending on your computer you may have to find a compromise here.



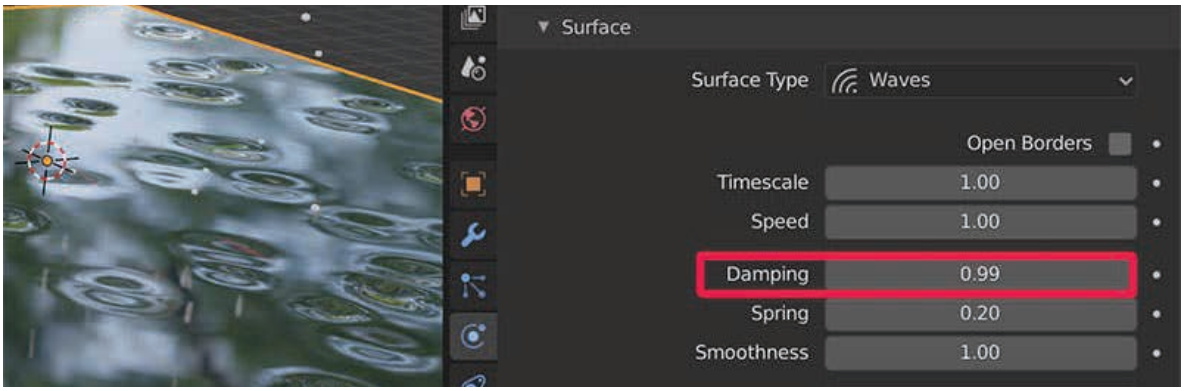
If you have trouble playing back in real-time I suggest that you save your Blend file and then Bake the simulation (the Bake options will be greyed out unless you save the file first).



If the waves made by the rain are not big enough to your liking you can increase the Radius value of the Surface. This increases how big the waves are.



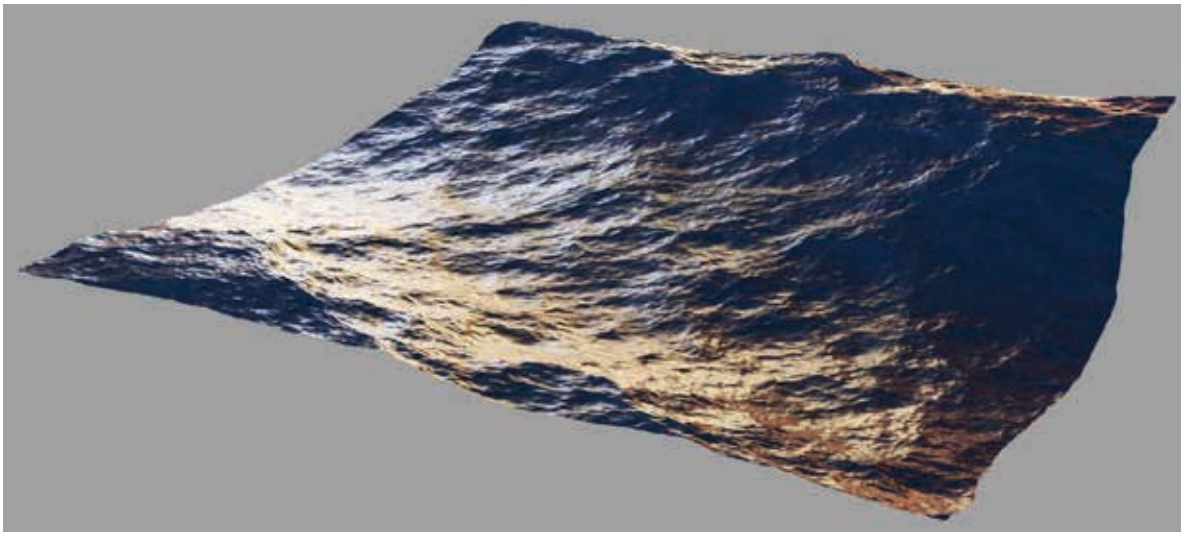
If you want to make the puddle more “sirupy” increase the **Damping** to **0.99**. Also try and set it to **1** and see what happens - it becomes more like snow!



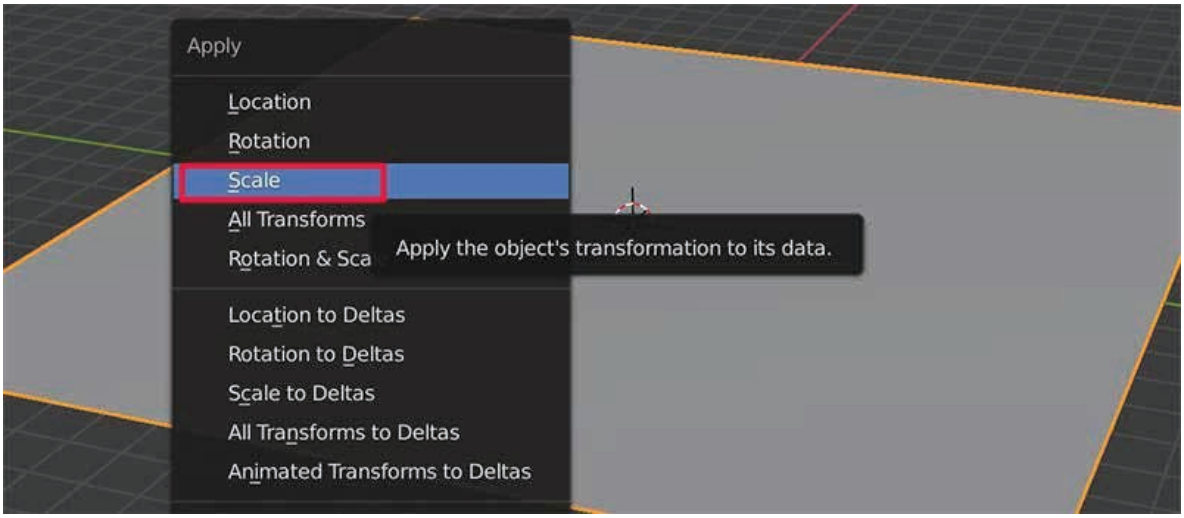
That's it, you've made it rain! These are the basics of creating rain with Dynamic Paint, but Dynamic Paint is useful for other things as well. Try setting the **Surface Type** of the Canvas to **Displace** instead of Waves to see a different kind of effect you can achieve.

Seamlessly Looping Ocean

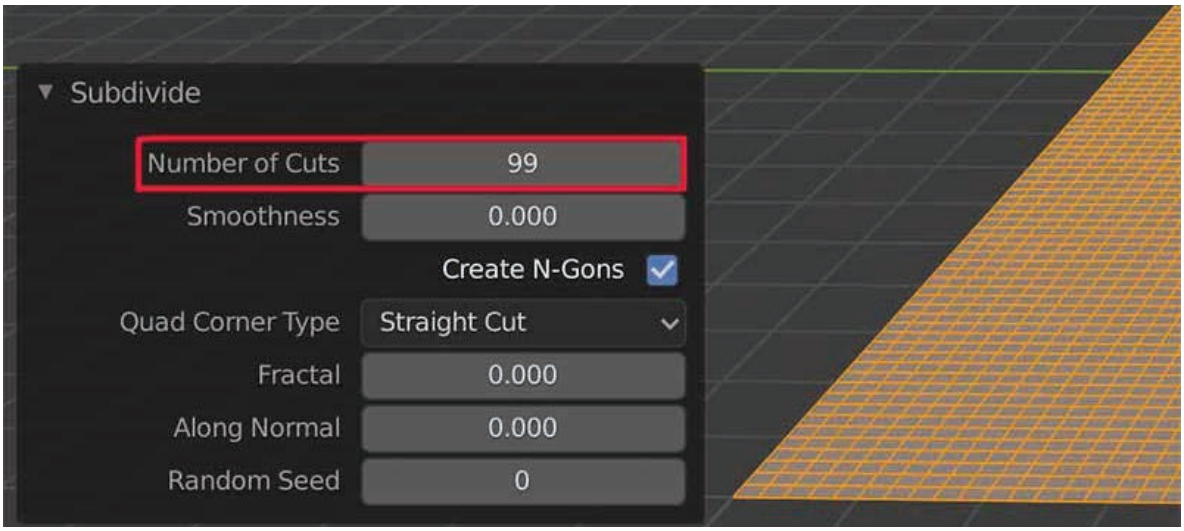
In this tip, we're going to look at creating a seamlessly looping animated ocean in Blender.



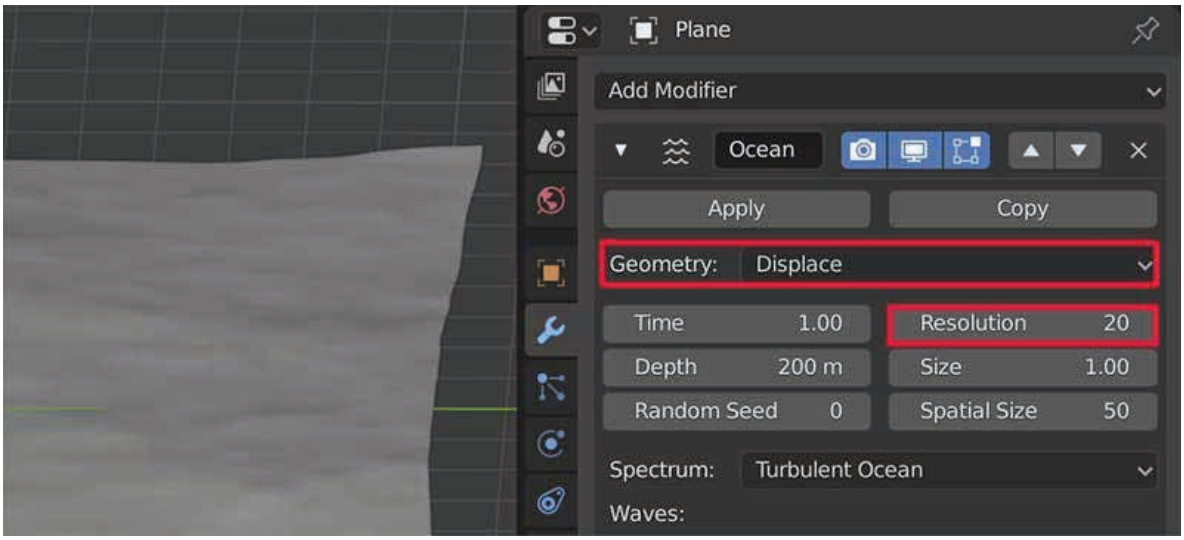
Create a **Plane** by **pressing Shift+A** and choosing Mesh>**Plane** and scale it up 10 units by **pressing S+10**. Then **press Ctrl+A** and choose **Scale** to apply the scale.



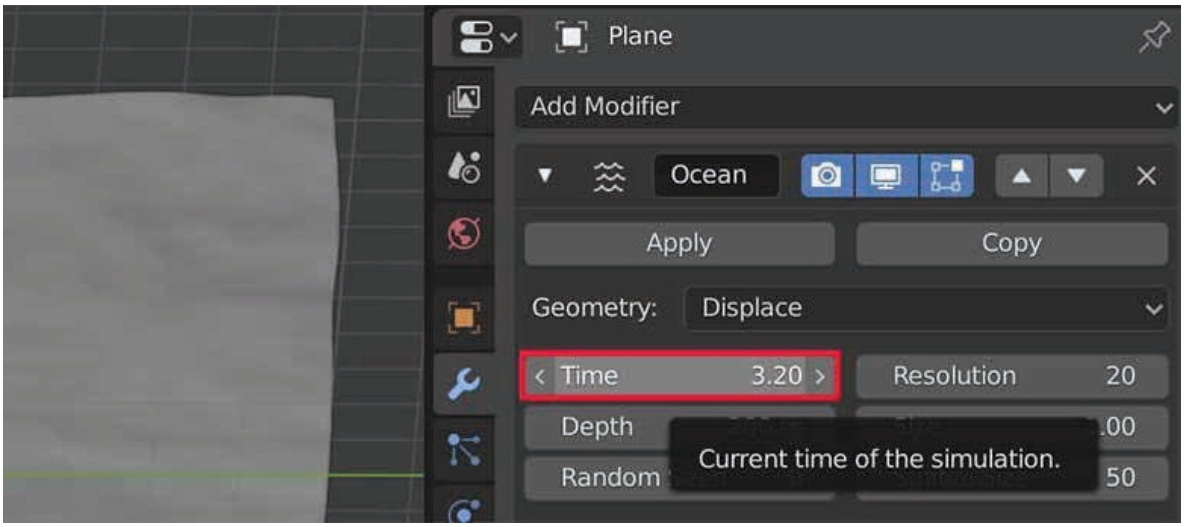
Press Tab to go into Edit Mode, **right-click** on the Plane and choose **Subdivide**. Then in the Subdivide option panel in the lower left corner of the viewport subdivide it with 99 cuts. We need this kind of detail to see the waves of the ocean.



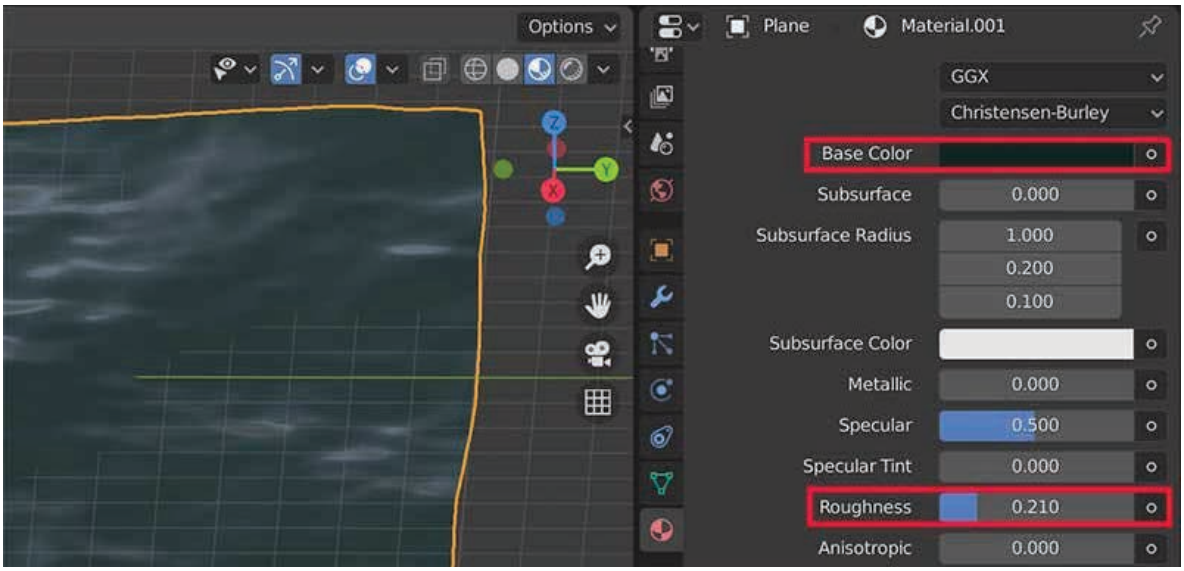
Press Tab to go back in Object Mode add an **Ocean modifier** to the Plane. Set **Geometry** to **Displace**, increase the resolution to 20, or a lower value if 20 turns out to be too much for your computer. Be careful with setting this value too high, as Blender might crash.



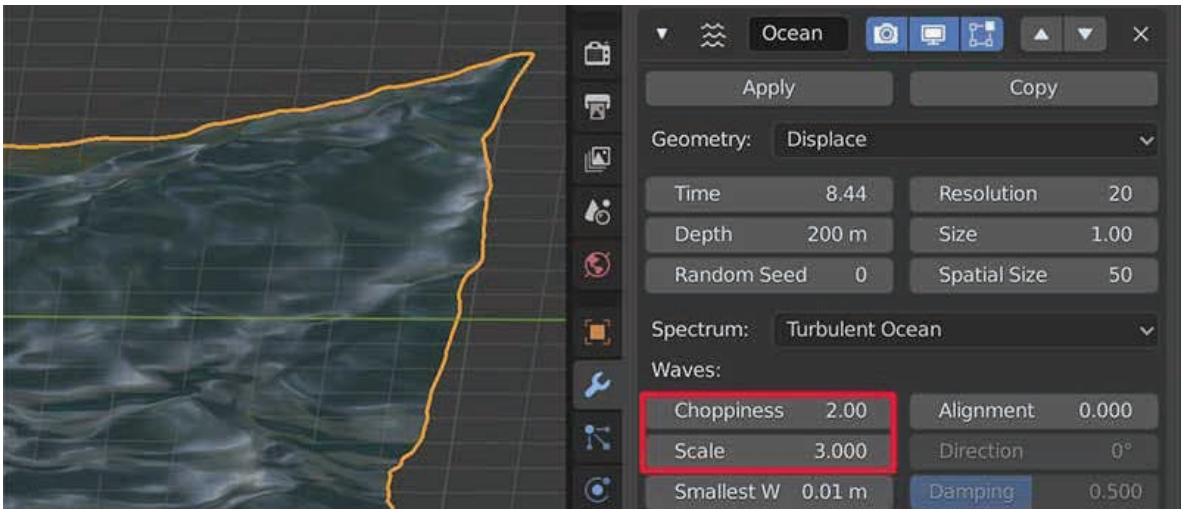
Right-click and choose **Shade Smooth** in the 3D Viewport so the waves look smoother. When you **press the Spacebar** you'll notice that the ocean doesn't animate. To animate the modifier we need to set keyframes for the **Time** value. If you drag the Time value you'll see the ocean move.



To make it more visually appealing, press Z and move the mouse down to go to the **Material view**, then add a new material to the Plane and set it to a dark green/blue color and a low **Roughness value** so it looks more like ocean water. This way it's easier to see the waves.

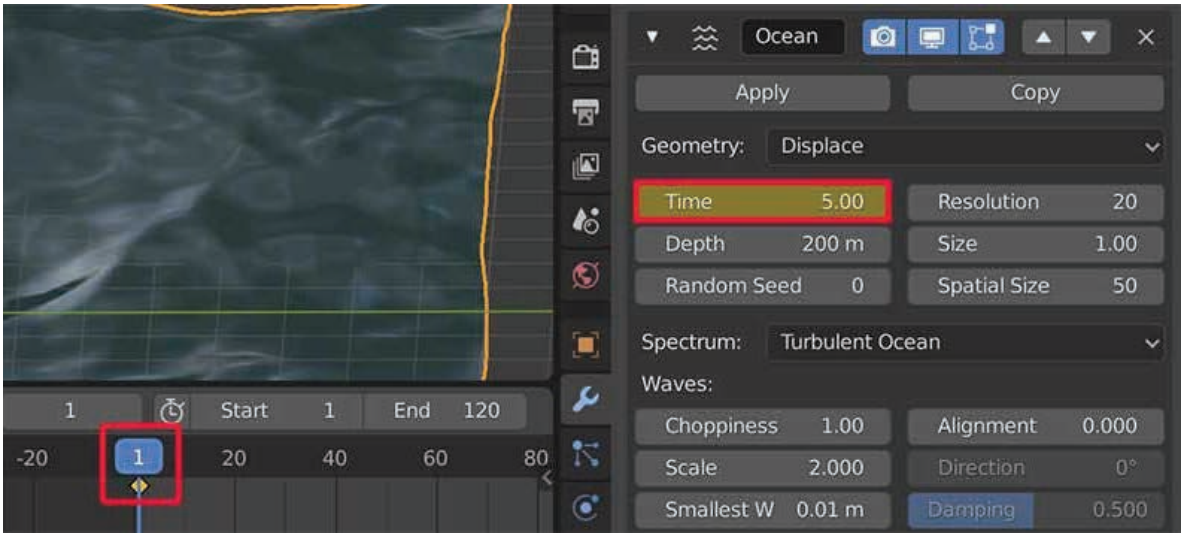


An important value in the ocean modifier is **Scale** (found under the Waves options). Set it to 2 or even higher to increase the waves. You can increase **Choppiness** as well but I actually prefer to keep it at 0 or 1. However try out some higher amounts so you understand what each value does.

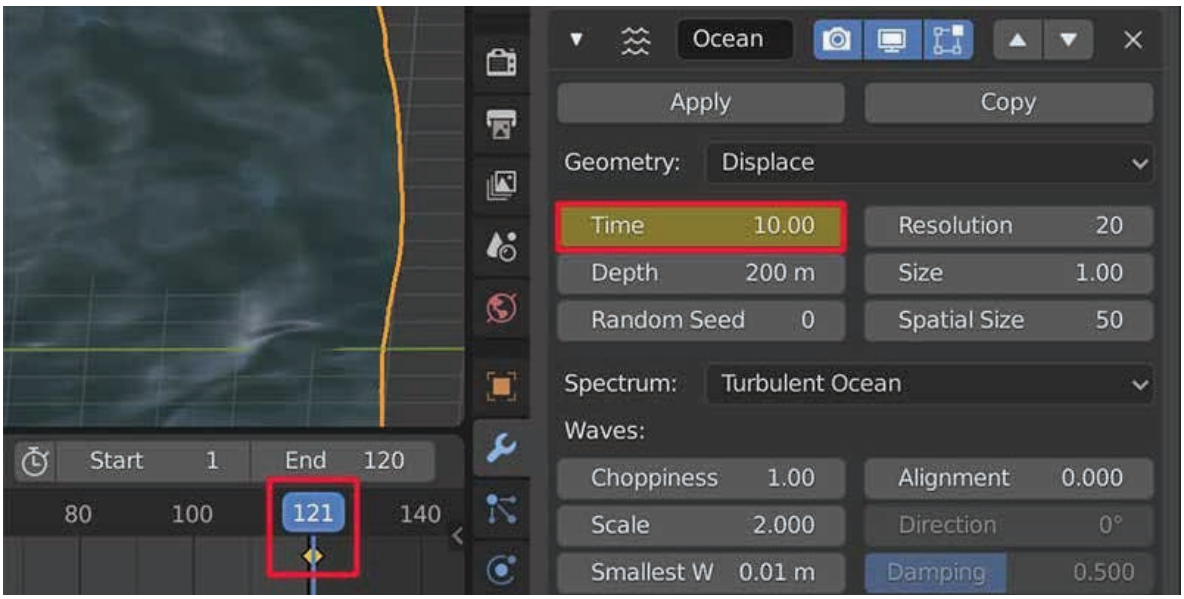


The **Scale** value is actually essential to create the looping animation of the ocean. Note that I've set my frame range to 120 frames.

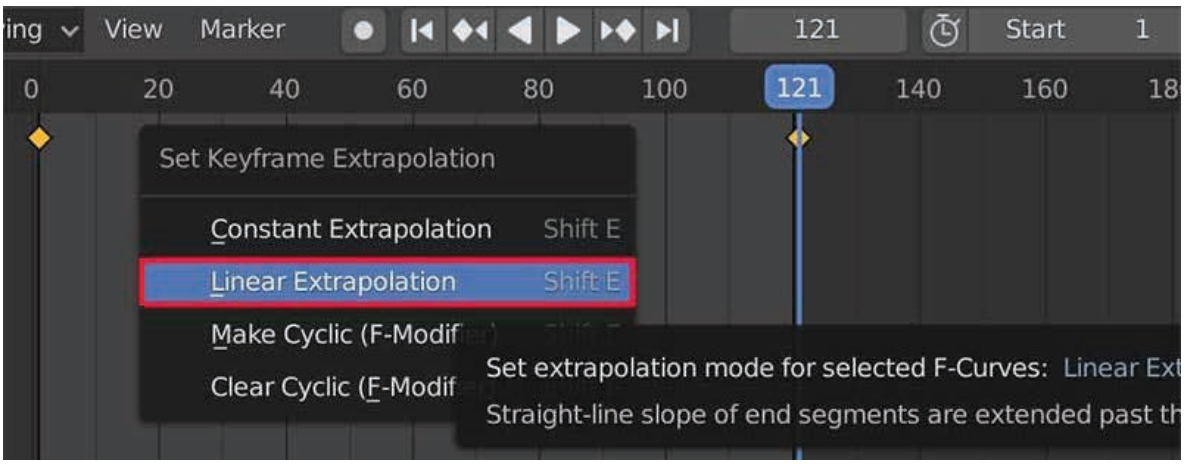
Press Shift+[left arrow key] to make sure you are at the beginning frame of the Timeline. There, set a keyframe for the **Time** value of **5** by typing 5 in the Time value field and **pressing i** while hovering the mouse cursor over that field (it will turn yellow to indicate the keyframe has been created).



Go to the end frame by **pressing Shift+[right arrow key]** and **press the [right arrow key]** one more time to go to frame **121**. Going one frame further than your frame range is an essential trick to creating perfect loops. Set another keyframe for the **Time** value, this time for the value **10**.



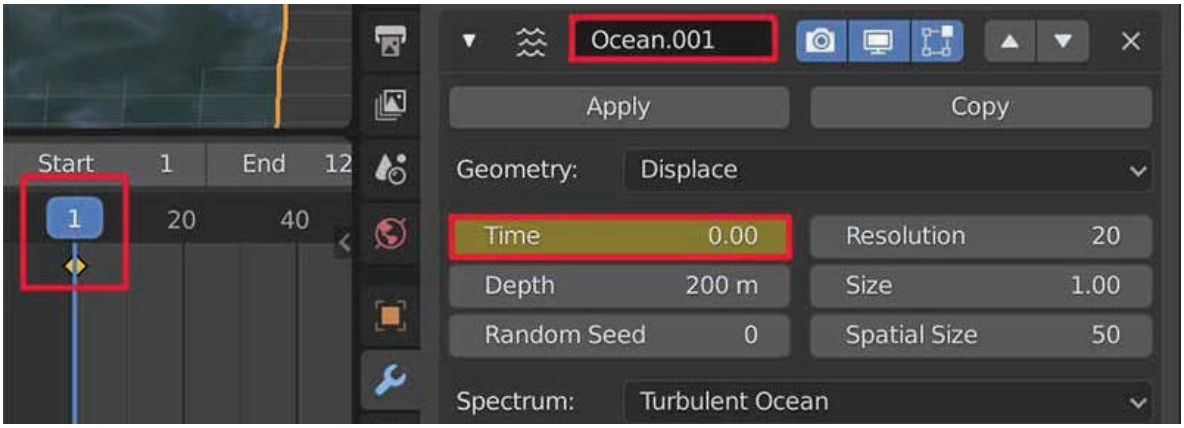
Select both keys in the Timeline and **right-click**, choose **interpolation Mode > Linear**. That way the animation doesn't ease in and out in the beginning and end, but rather keeps moving at the same speed. (Alternatively you can press **Shift+E** and choose **Linear Extrapolation**)



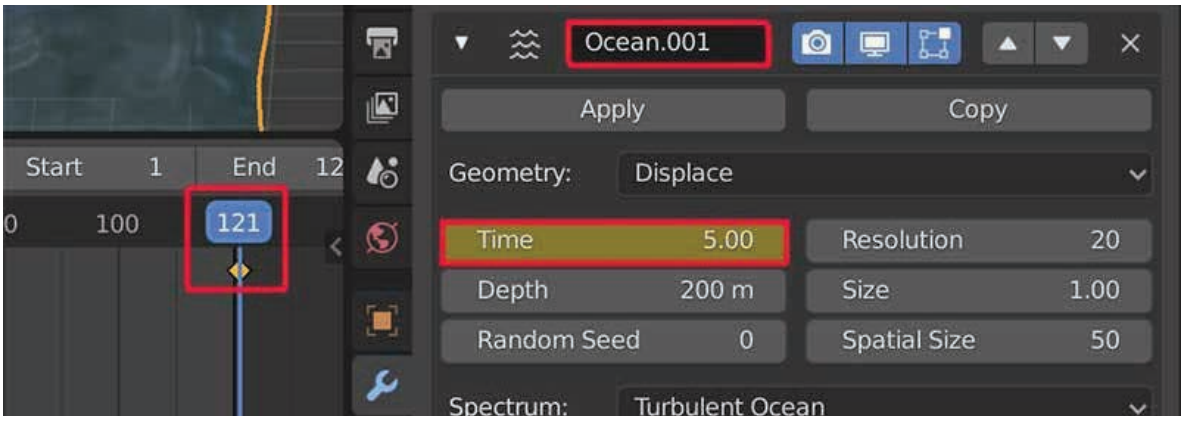
To make it possible to create a looping animation, we wil need a second Ocean modifier. To do this, hover the mouse cursor over the modifier and **press Shift+D.**

(Note: if you're using an older version of Blender than 2.9, click the Copy button on the modifier)

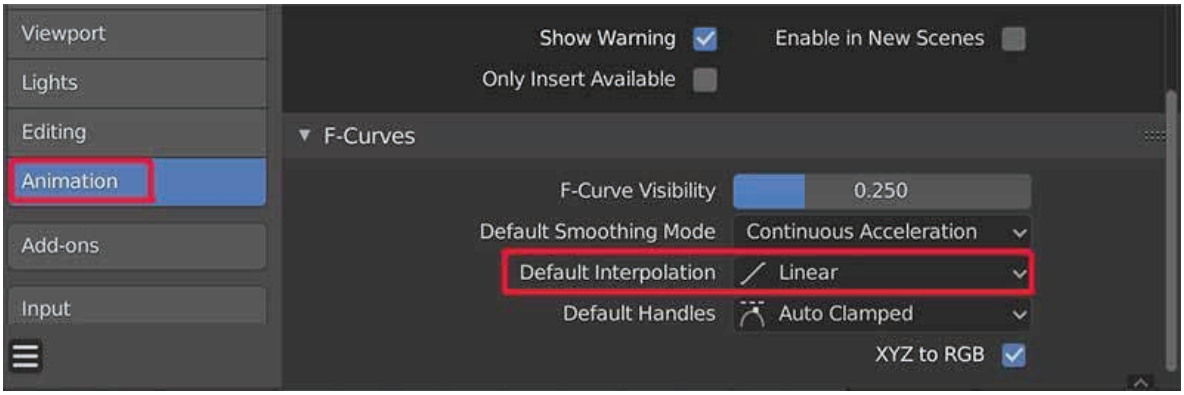
Now we set the keyframes for the second Ocean modifier (called "Ocean.001"). **Press Shift+[left arrow key]** to make sure you are at the beginning frame of the Timeline and set a keyframe for the **Time** value of **0** by typing 0 in the Time value field and **pressing i** while hovering the mouse cursor over that field.



Go to the end frame by **pressing Shift+[right arrow key]** and like before, go one frame further, to frame 121. Set another keyframe for the **Time** value, this time for the value of **5**.

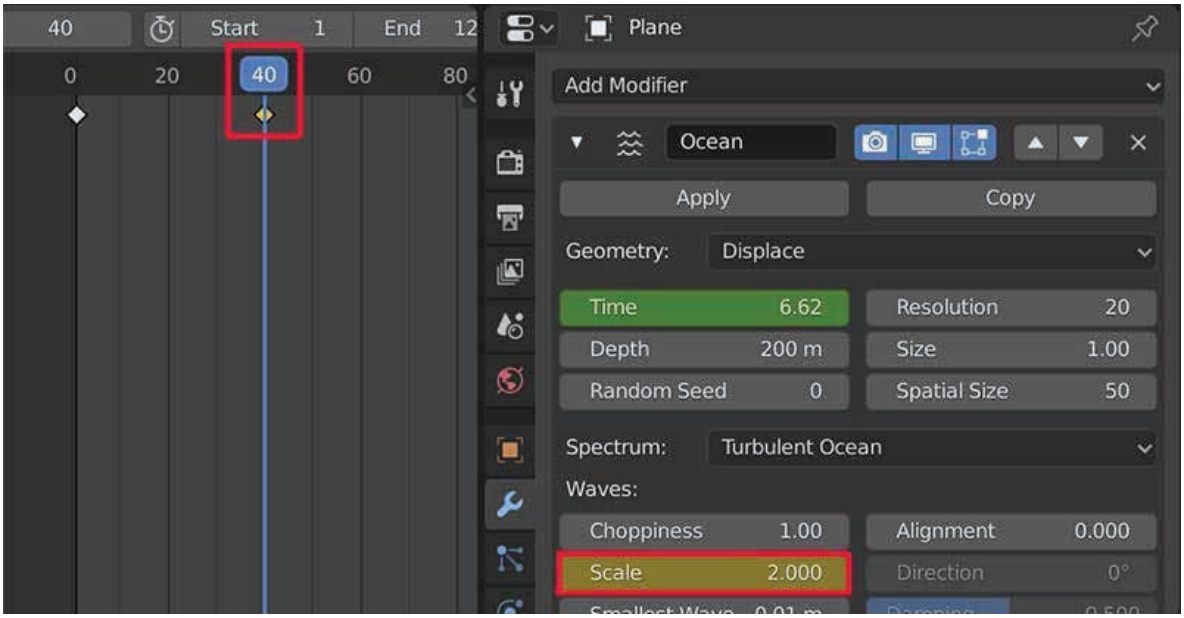


Like before set the keys to **Linear interpolation**. I actually have this set as the default in my preferences, which is the way it works by default in other applications like After Effects, so that I don't have to always change the key interpolation type.

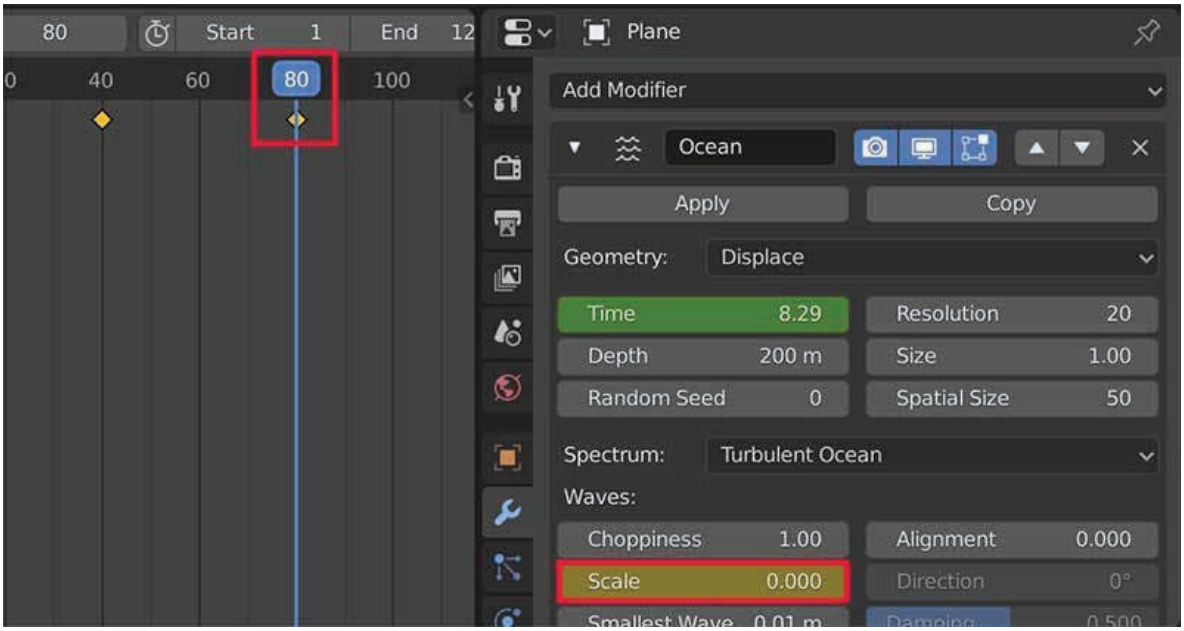


So now you've got one Ocean modifier that goes from values **5** to **10** and a second one that goes from values **0** to **5**. Now we just need to "blend" from one to another in the middle of the Timeline to make the loop seamless. We will do this by setting keyframes for the **Scale** values of both modifiers.

Go to frame 40 on the Timeline and set a **Scale** keyframe of **2** on the first Ocean modifier and a **Scale** keyframe of **0** on the second Ocean modifier.

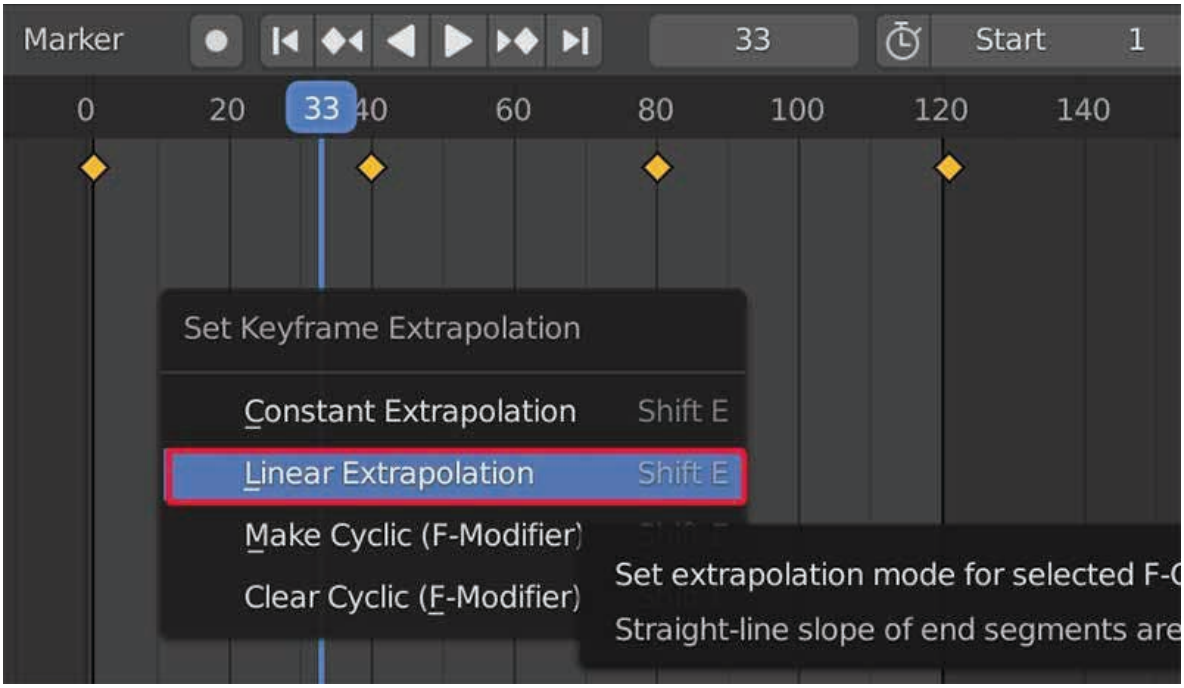


Then go to frame 80 and set the opposite keyframes: **Scale** of **0** on the first Ocean modifier and a **Scale** of **2** on the second Ocean modifier.



So as the Scale of modifier 1 goes from 2 to 0, the Scale on the second modifier goes from 0 to 2, effectively blending from one to the other between frames 40 and 80.

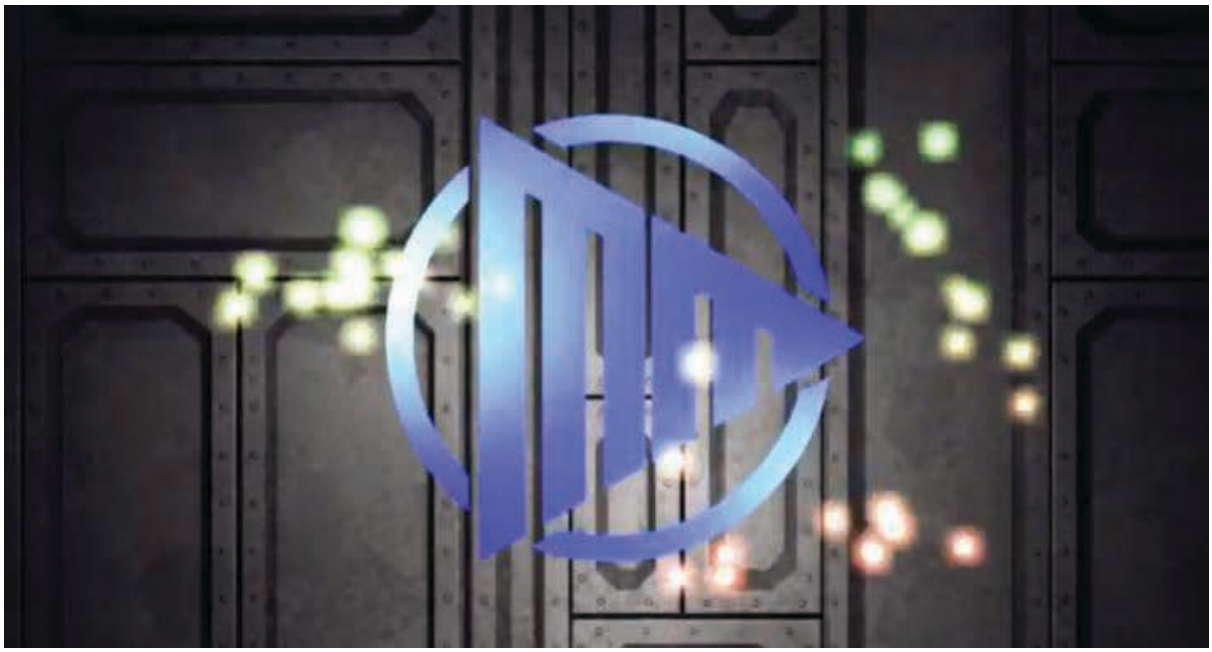
Again, select all the keyframes on the Timeline and **right-click**, choose **Interpolation > Linear** (or **press Shift+E** and choose **Linear Extrapolation**).



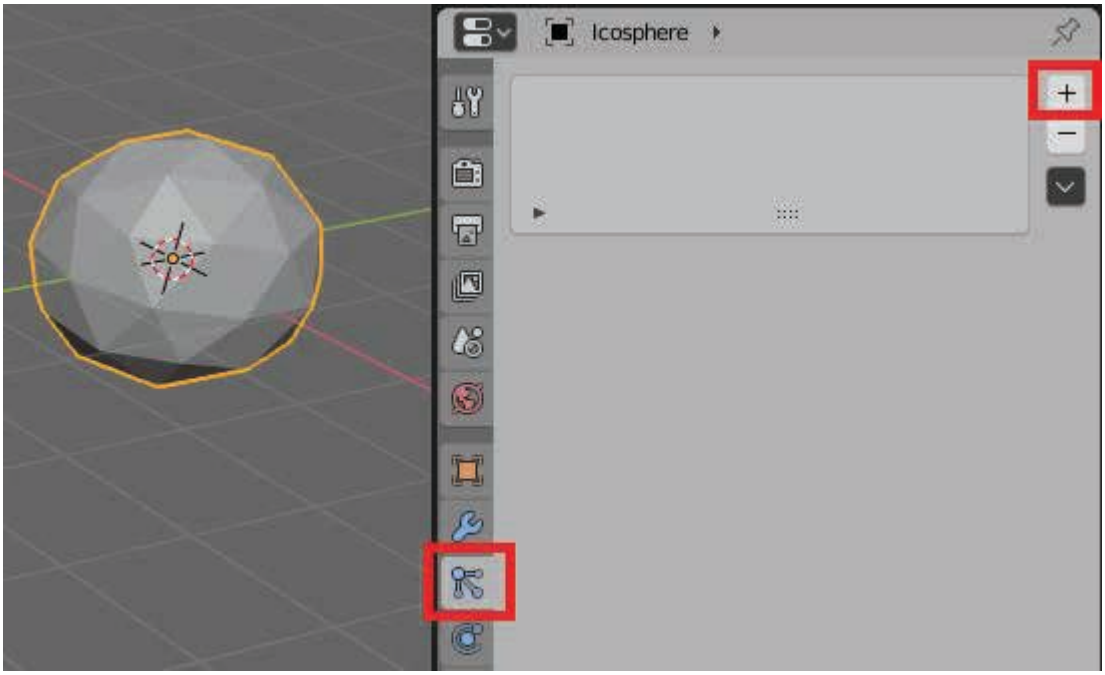
I learned this ocean blending method from Midge "Mantissa" Sinnaeve.

Particles Follow A Path

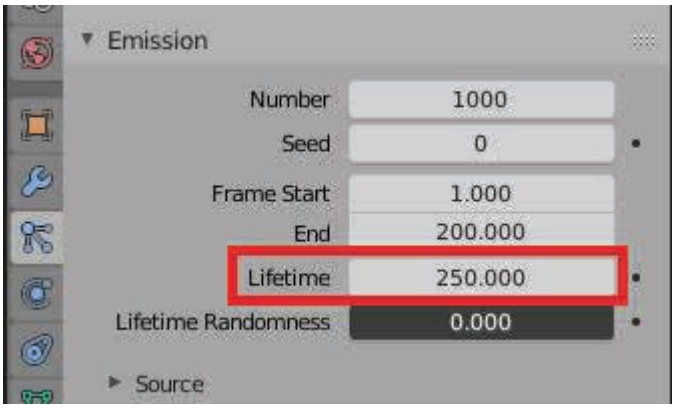
Did you know that you can make particles follow a path or curve? It looks pretty cool...



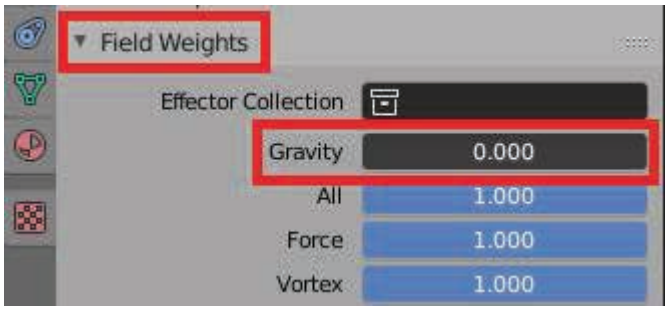
First, create an object (like an icosphere) and add a particle system to it in the Particle tab.



Create the Particle system by **clicking the + button**. Increase the lifetime so that the particles don't disappear immediately after being spawned.

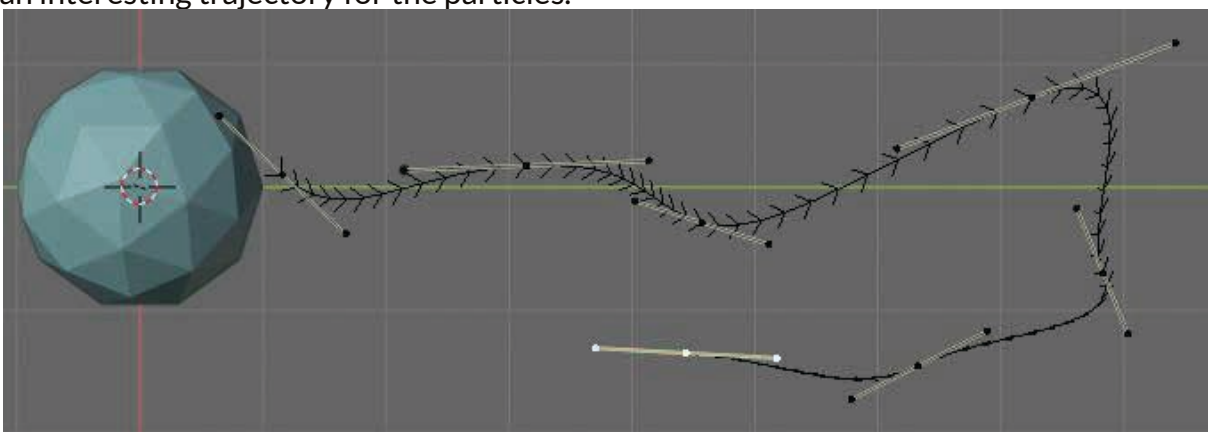


Turn Gravity to zero, so that the gravity has no influence on the particles (they won't fall down but float, as if they are in space).



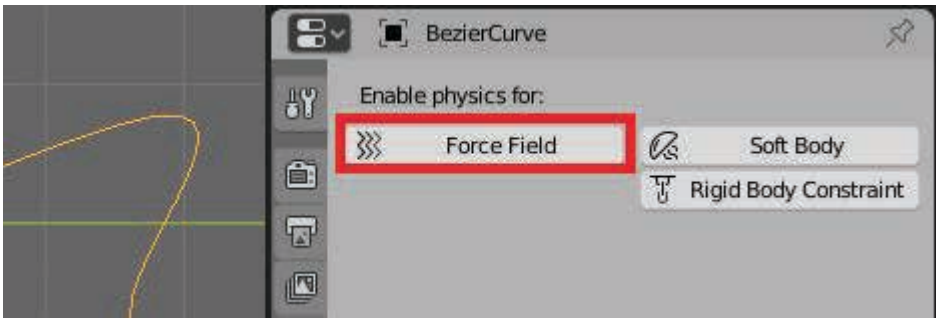
Then, create a Bezier curve. Press **Shift+A**, choose Curve > **Bezier**.

Put the curve with one end near the particle emission object (the icosphere), and create an interesting trajectory for the particles.

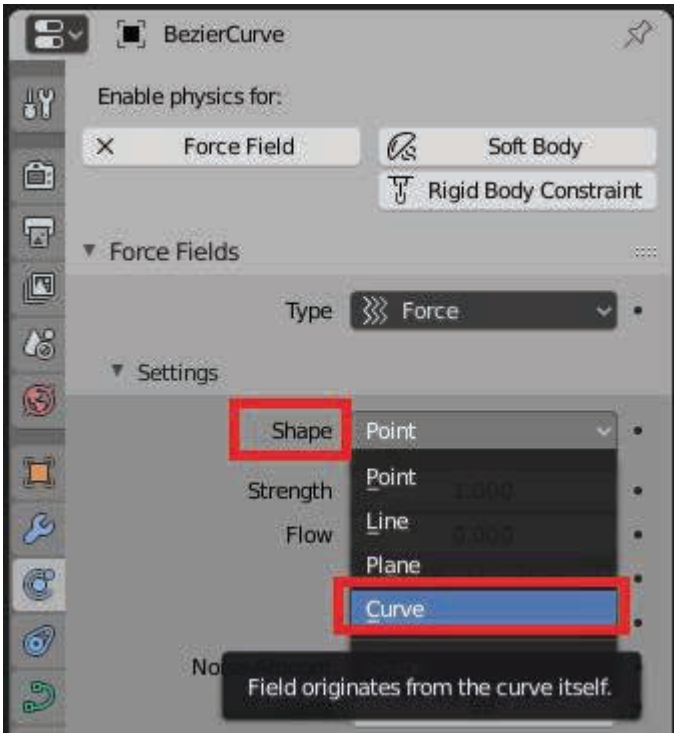


In Edit Mode, select the vertex at one end of the Curve and **press Ctrl+ right-mouse button** to extrude to where you click in the viewport. This way you can quickly extend the Curve.

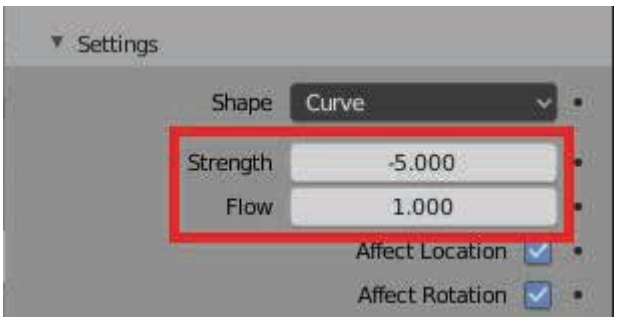
Now, in Object Mode add a **Force Field** to the curve in the Physics tab.



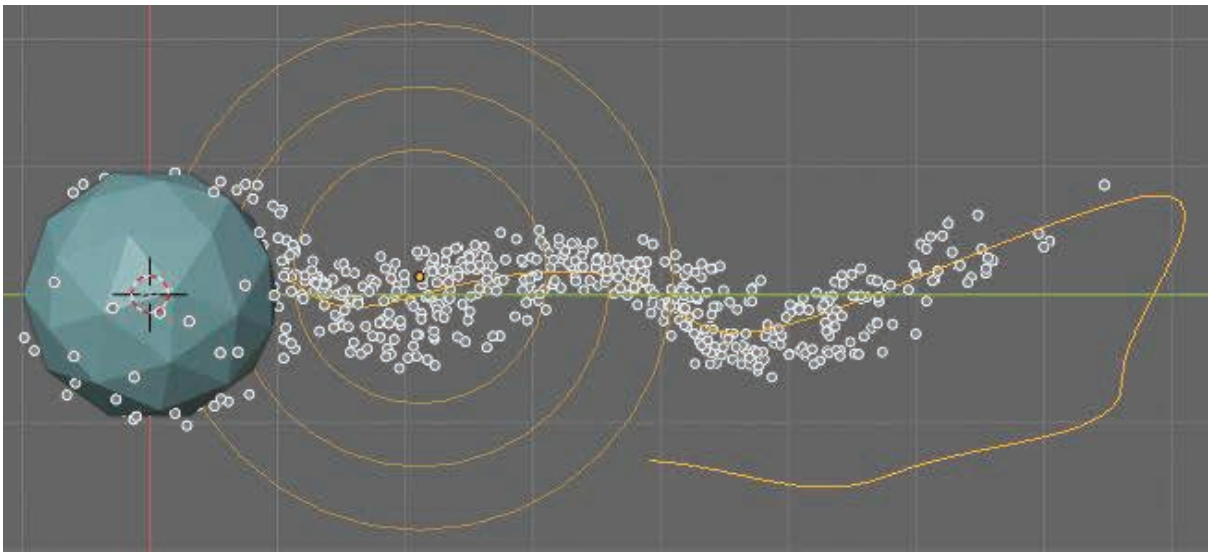
Set the Shape to "**Curve**".



Set the Strength to -5. Of course you can experiment with these settings, but this is a good starting point in my experience. Set Flow to 1.



Press the Spacebar to play the simulation. While playing, you can experiment with the Strength and Flow settings, and see what happens.



Quick Hair / Fur and How to Comb and Weight Paint

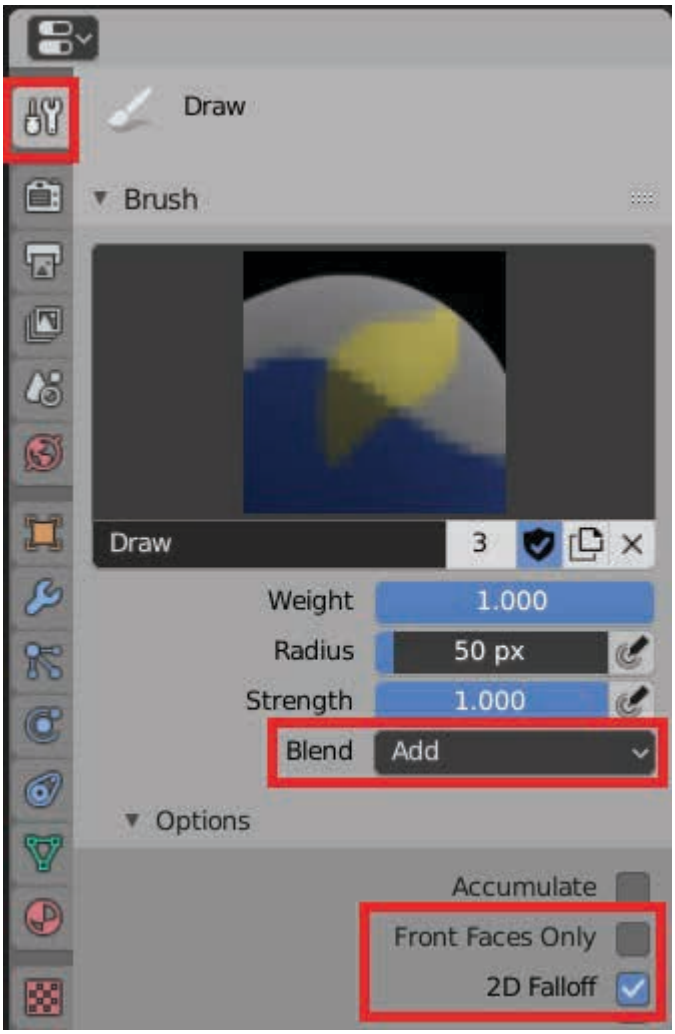
Let's add some hair to **Suzanne**, Blender's default monkey object.

Press Shift+A > Mesh > Monkey. Right-click, choose Smooth Shading. Move it up (G + Z) so it is placed above the grid. **Press Ctrl+1** so there's one level of subdivision.

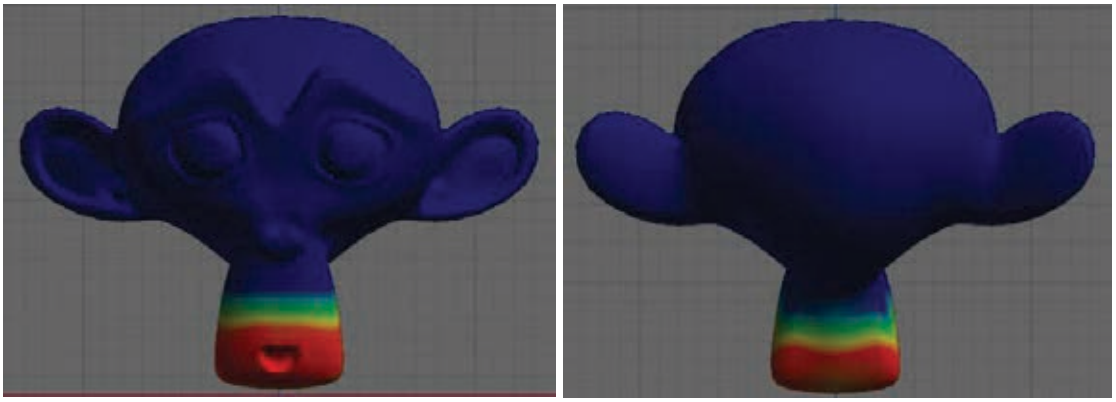
A smooth monkey.

Go to Weight Paint mode, and paint red where you want hair / fur…

Note: to paint “through” a model so the other side is also affected, turn off “Front Faces Only” and turn on “2D Falloff” in the brush options. Set the brush to “Add”.

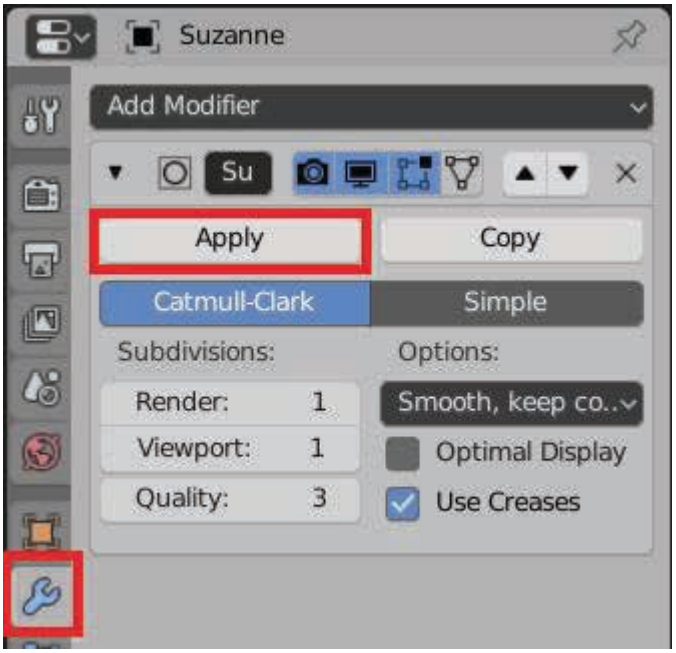


Now if we paint the front of the face, the back is also affected. Be careful: This works well for the chin, but not for the other parts of the face. In that case, use “Front Faces Only”.

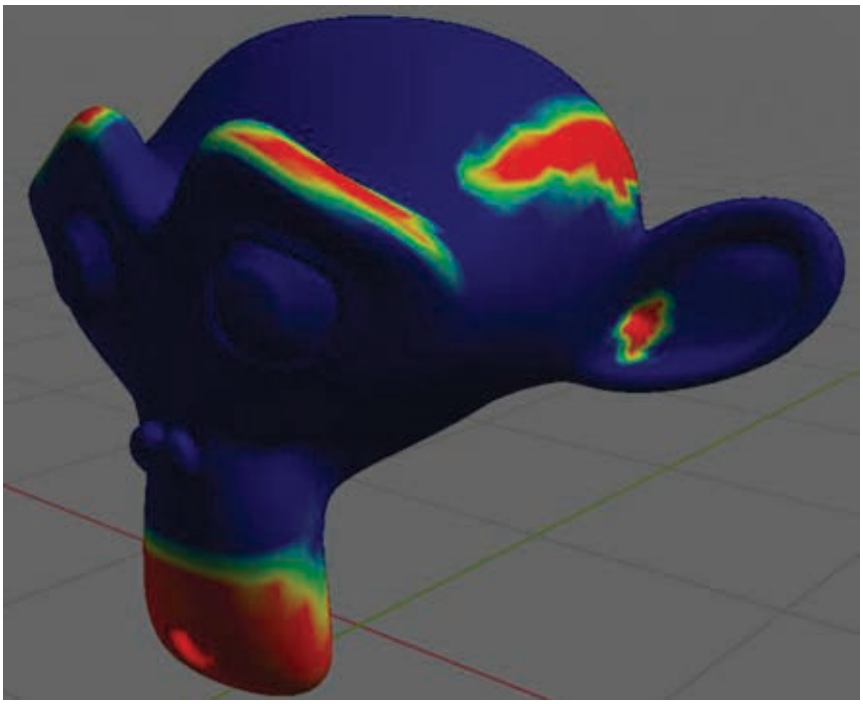


Press F to increase/decrease the size of the brush. Paint in add mode where you want hair to be distributed by the hair particle system. These spots should be red.

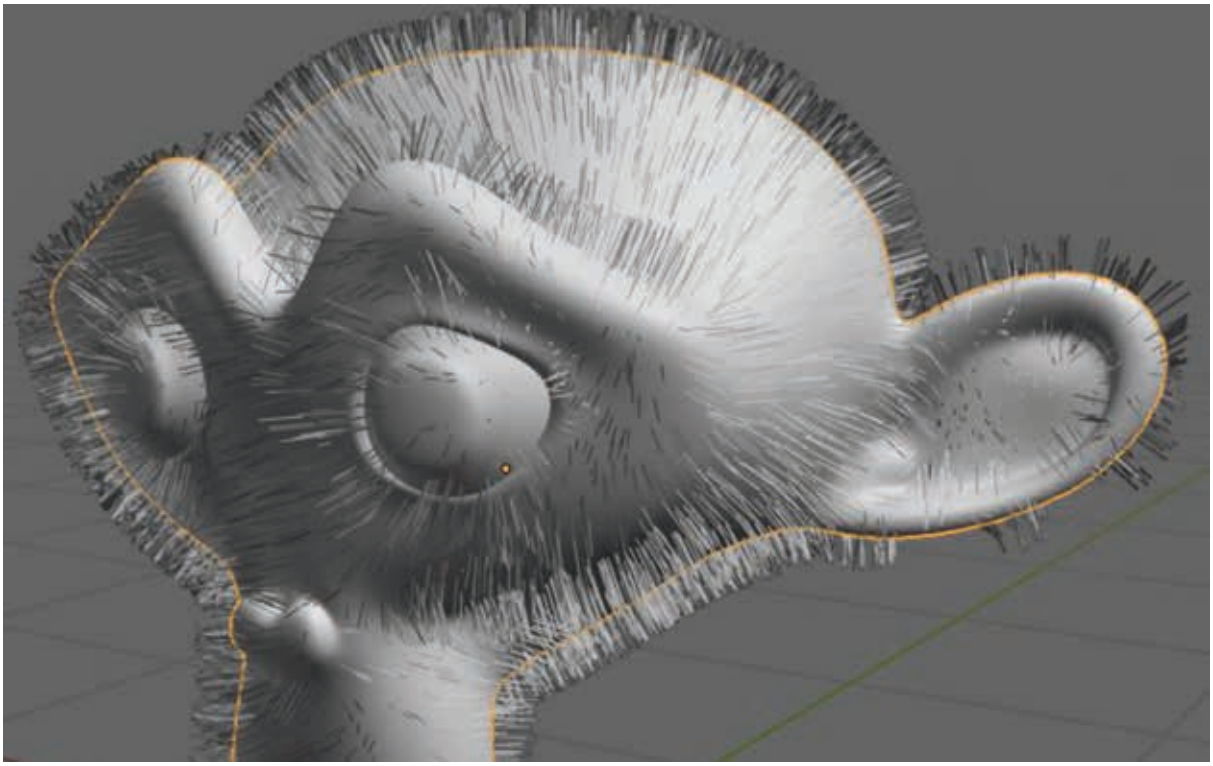
Note that the Monkey by default has very few polygon faces, making precise weight painting difficult. You can "apply" the modifier to permanently add those extra faces to the monkey, giving you more detail to weight paint. *(**In Blender 2.9** hover the mouse cursor over the modifier and **press Ctrl+A**.)*



To erase parts of your weight paint, set the brush from Add to Subtract. This works as a kind of eraser.

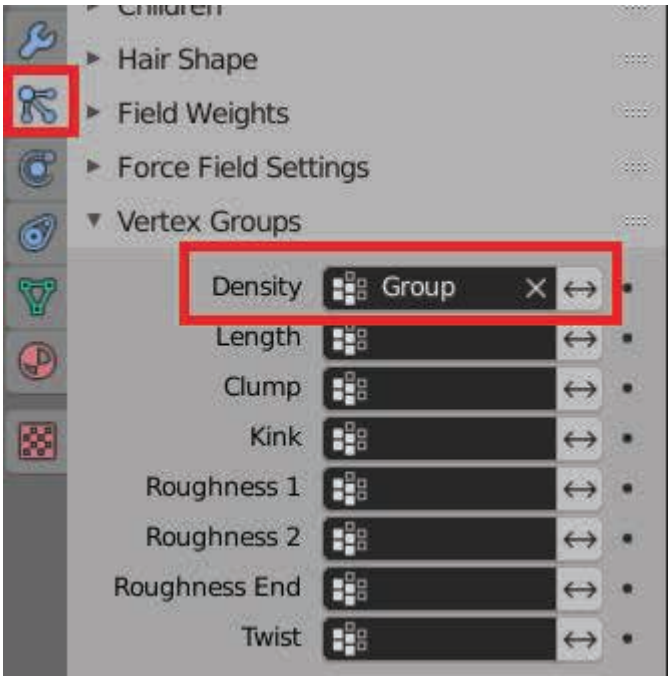


Go to Object Mode. Go to Object > **Quick Effects > Fur**.

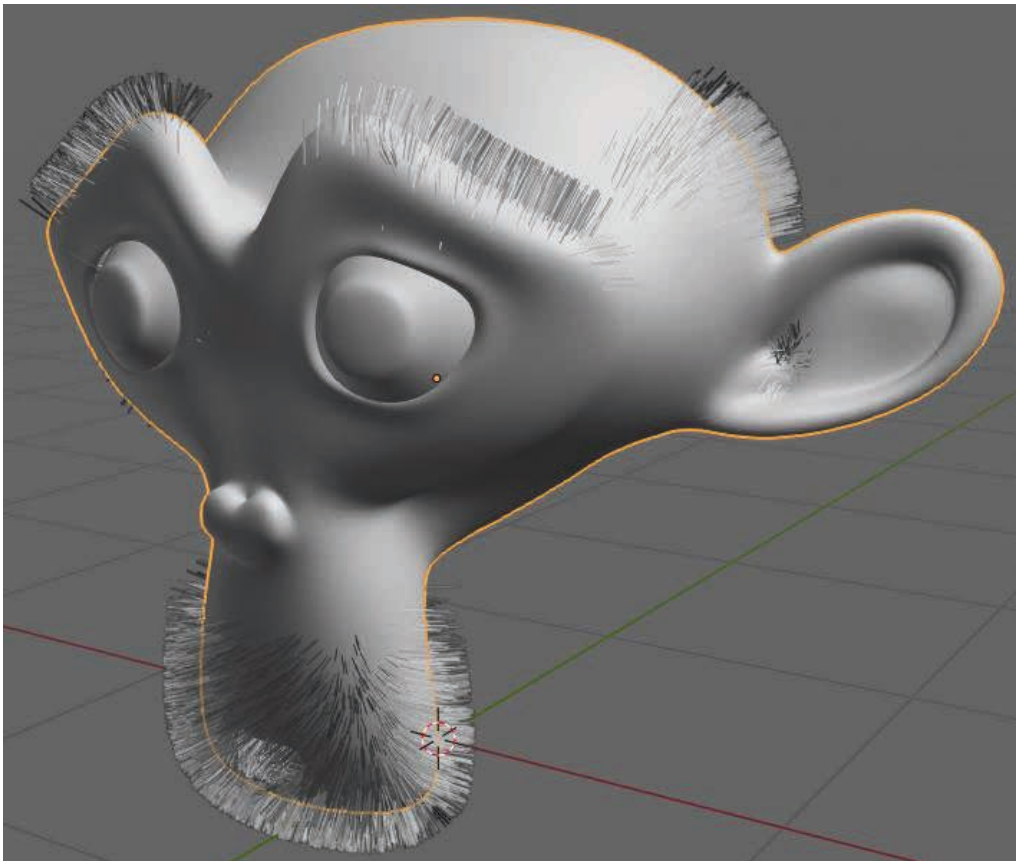


As you can see there's some kind of fur effect, but it is evenly distributed and doesn't use our weight painting.

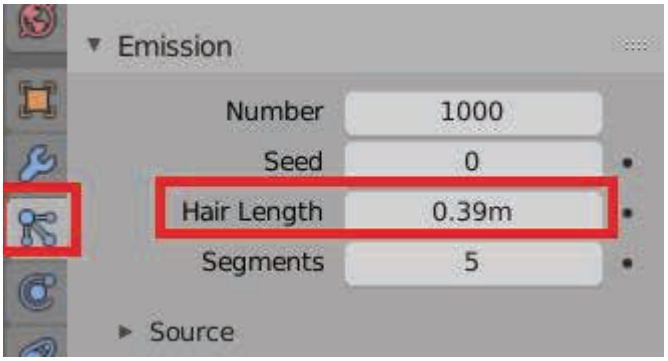
In the Particle settings (hair are "particles") - go to "Vertex Groups" and in Density choose the Weight Painted "Group" (there should only be the Group we made by weight painting).



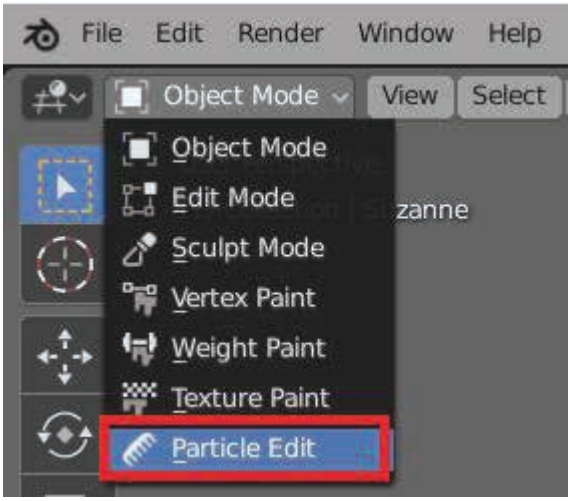
Now you immediately see the weight painting result.



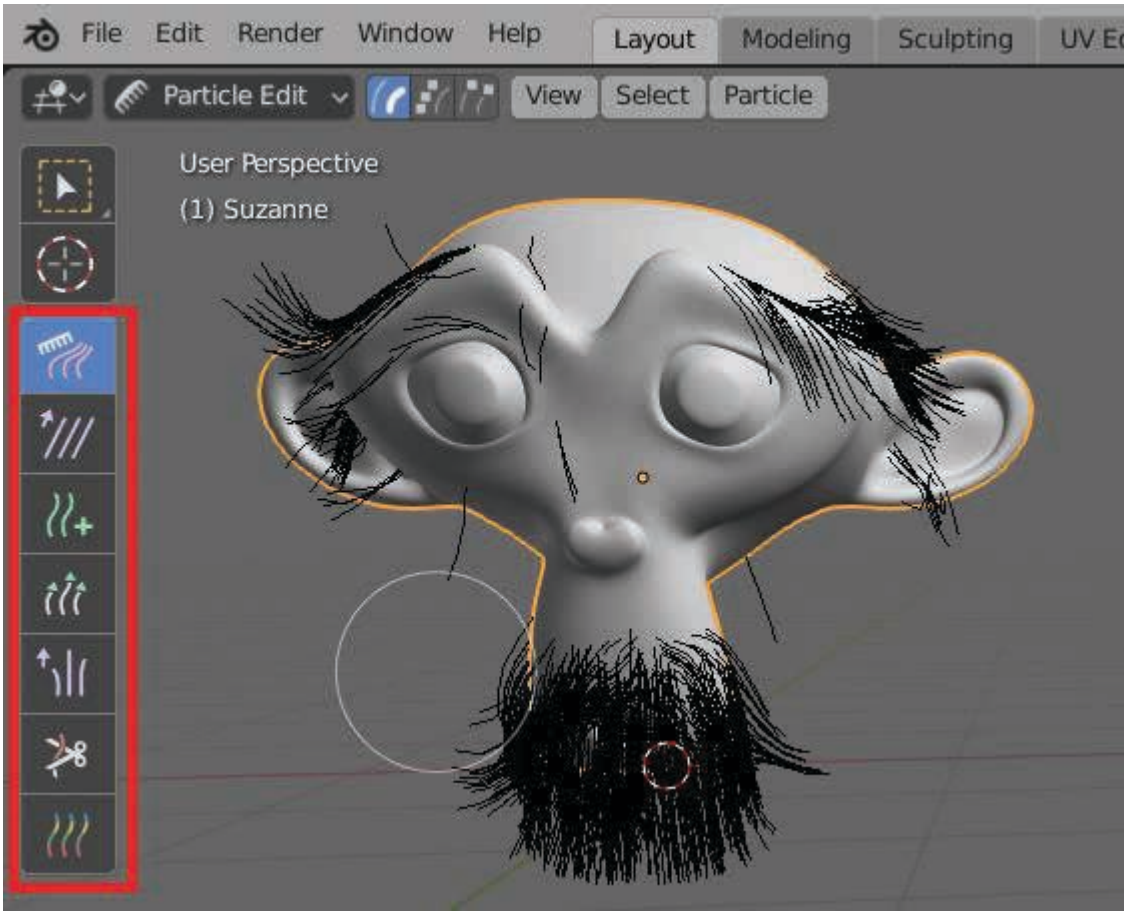
Play with Length in Particle Settings.



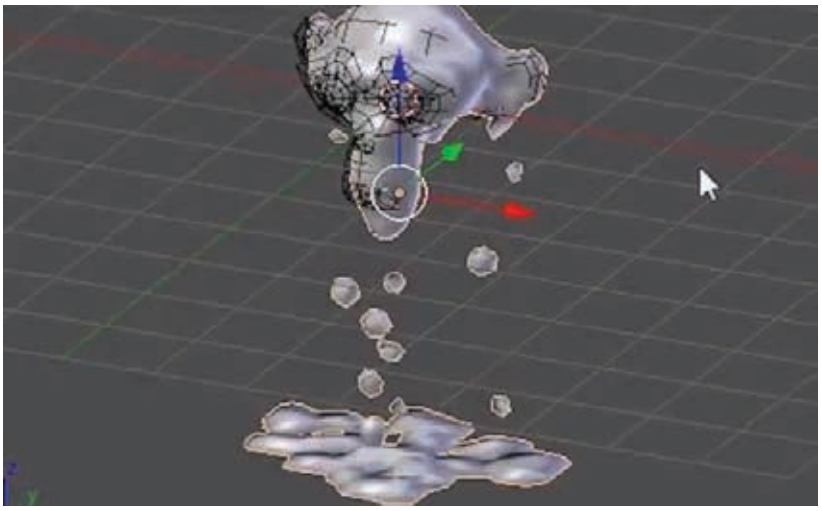
Ok, time to “style” the hair. Go to “Particle Edit” mode.



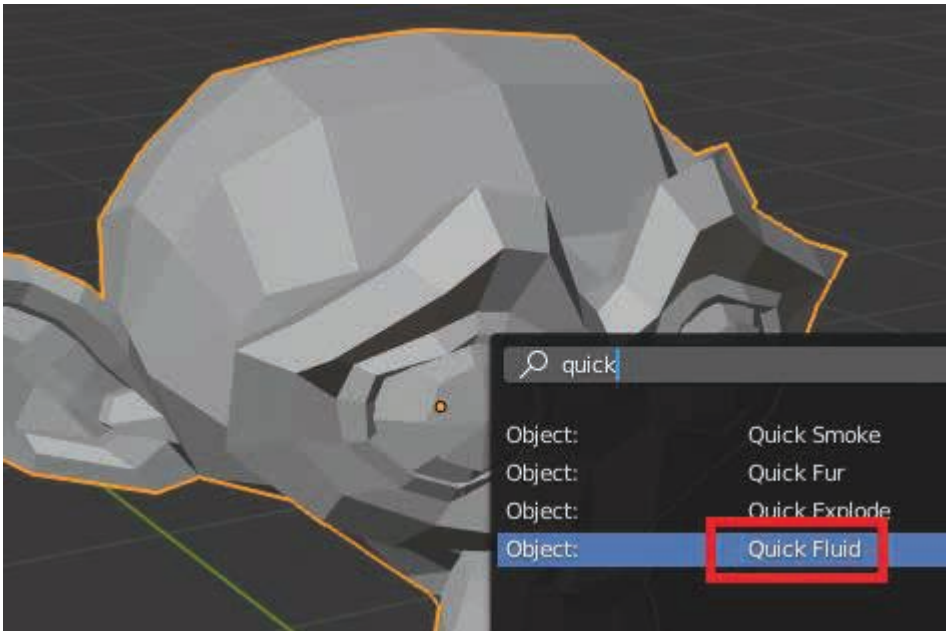
From the icons on the left you can comb, change the length, make it smooth, add more hair… if only real hair were so easy.



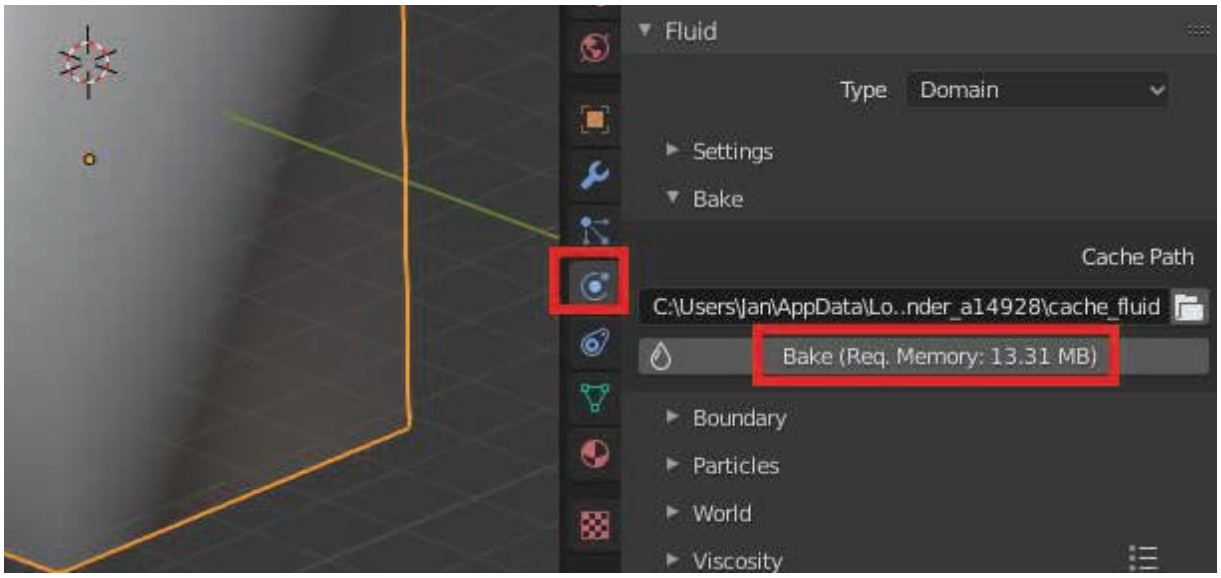
2 Types Of Quick Fluids



Press Shift+A, add a monkey… or other object if you like!
Press F3, type "quick fluid", choose Quick Fluid.



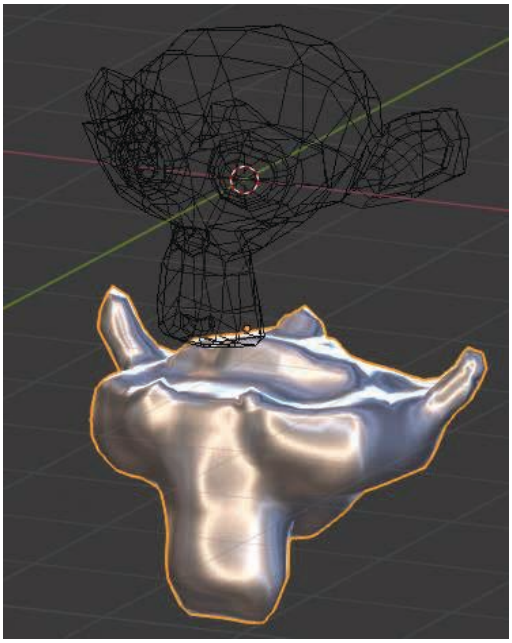
Everything is set up, just that easily. That cube is the “Fluid Domain”. **Click on Bake**. It will take a little while depending on your computer.

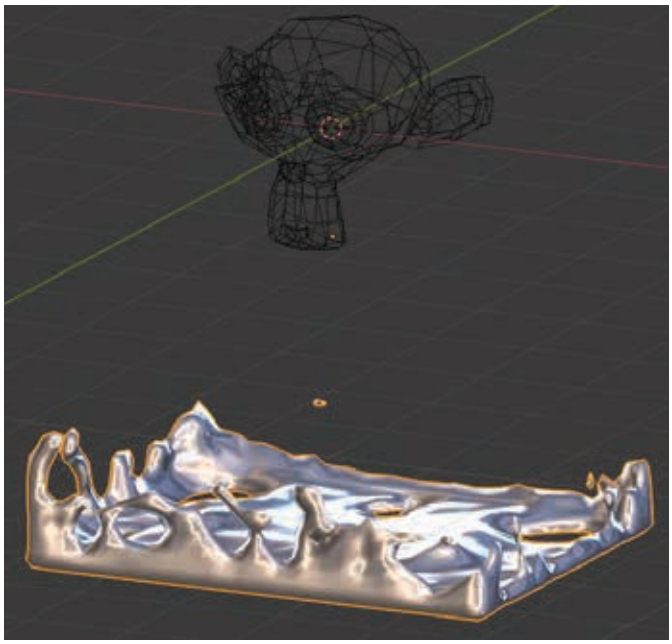


After it’s done Baking, play the simulation.

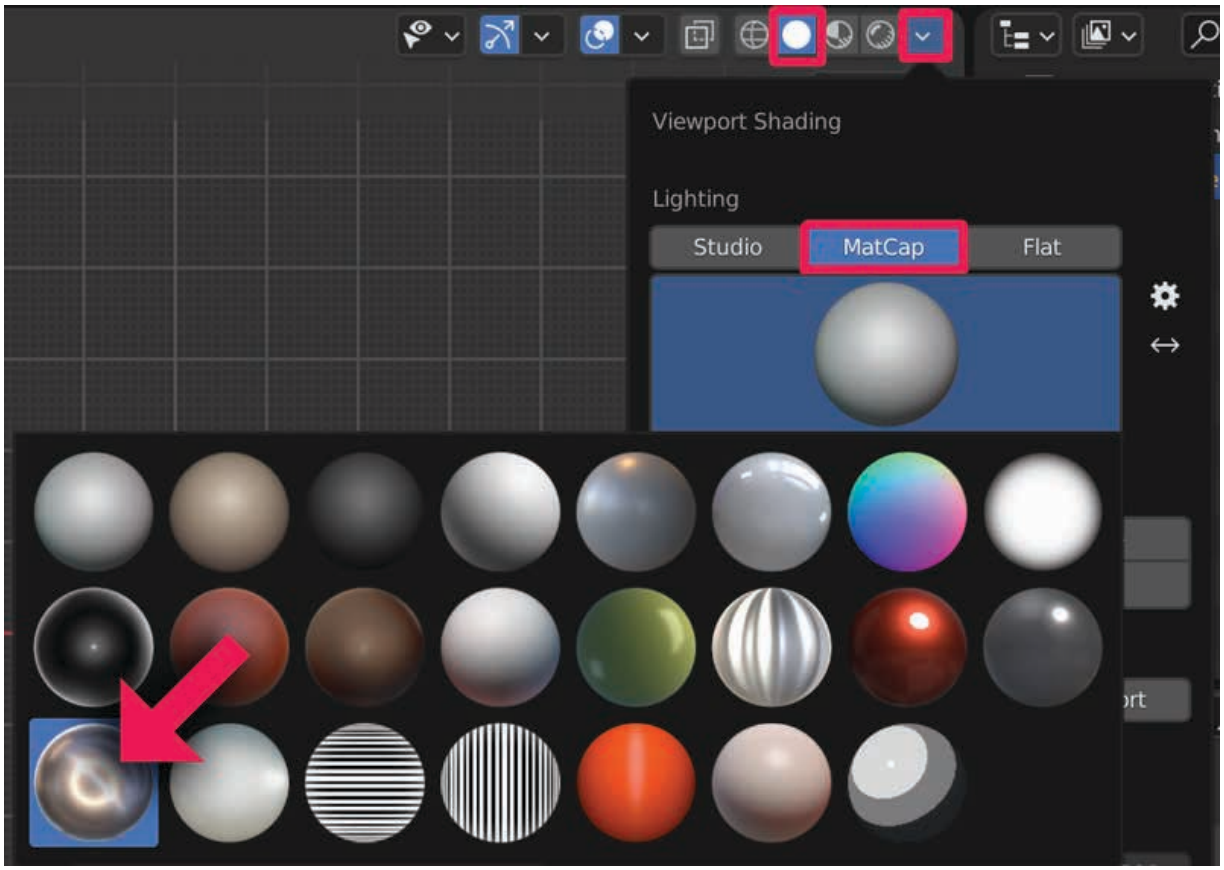
Some playback tips: **Press Shift + left arrow key** to go to the first frame. **Press Spacebar** to start playing the timeline.

Interestingly, this effect turns the entire object into liquid, which will then fall down and make a splash.





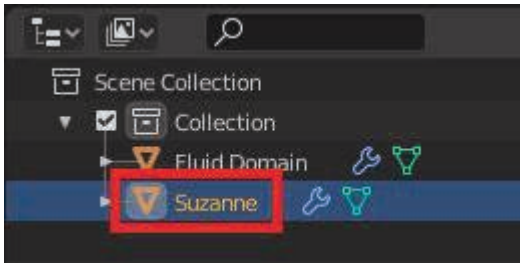
Note: I've turned on Matcap under **Viewport Shading** to make it look nice in the 3D viewport.



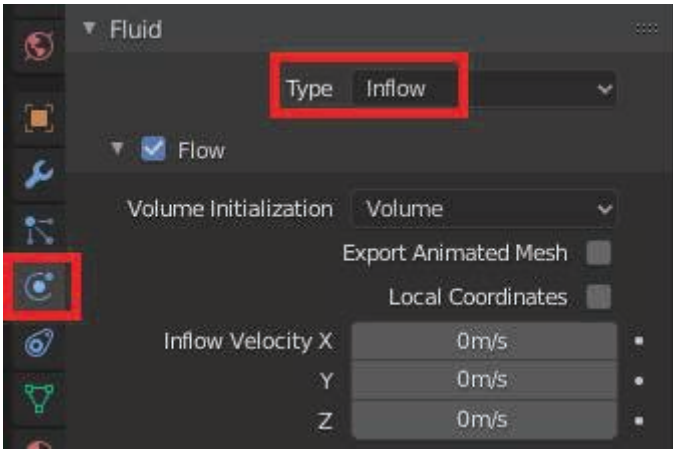
Let's try another kind of Fluid! Start with a new object, **press the Spacebar** and search for Quick Fluid… so far just the same steps as before.

Make sure the wireframe object is selected, your original object in other words.

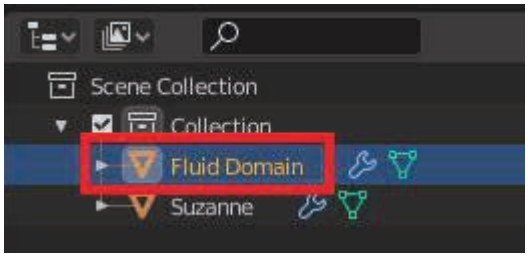
If you started with the Suzanne monkey, it should still be called Suzanne. If it's inside the domain cube, it's easier to select it in the Outliner.



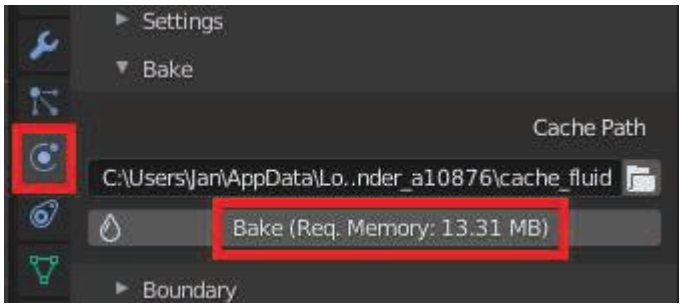
And in Type, this time choose **Inflow**.



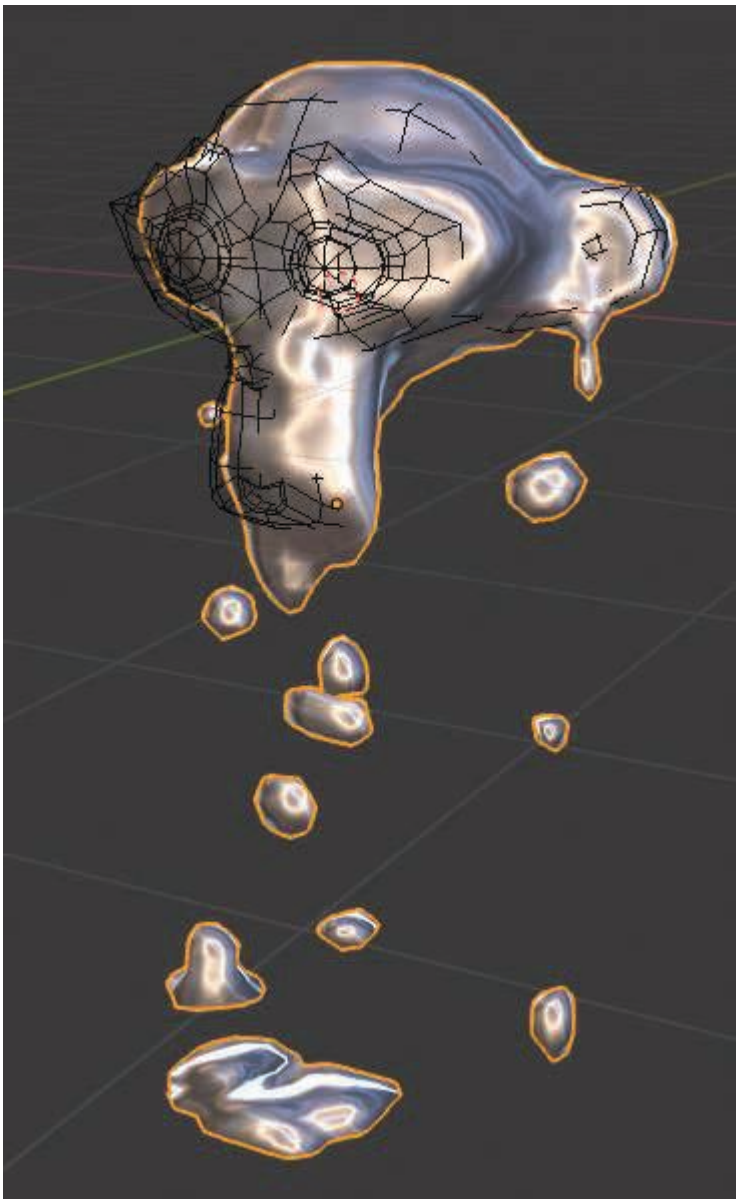
Select the other object (the Domain).



And press Bake.



Press Spacebar to see the fluid simulation. “Inflow” type causes a continuous flow of fluids to rain down from the object, which seems to be melting.

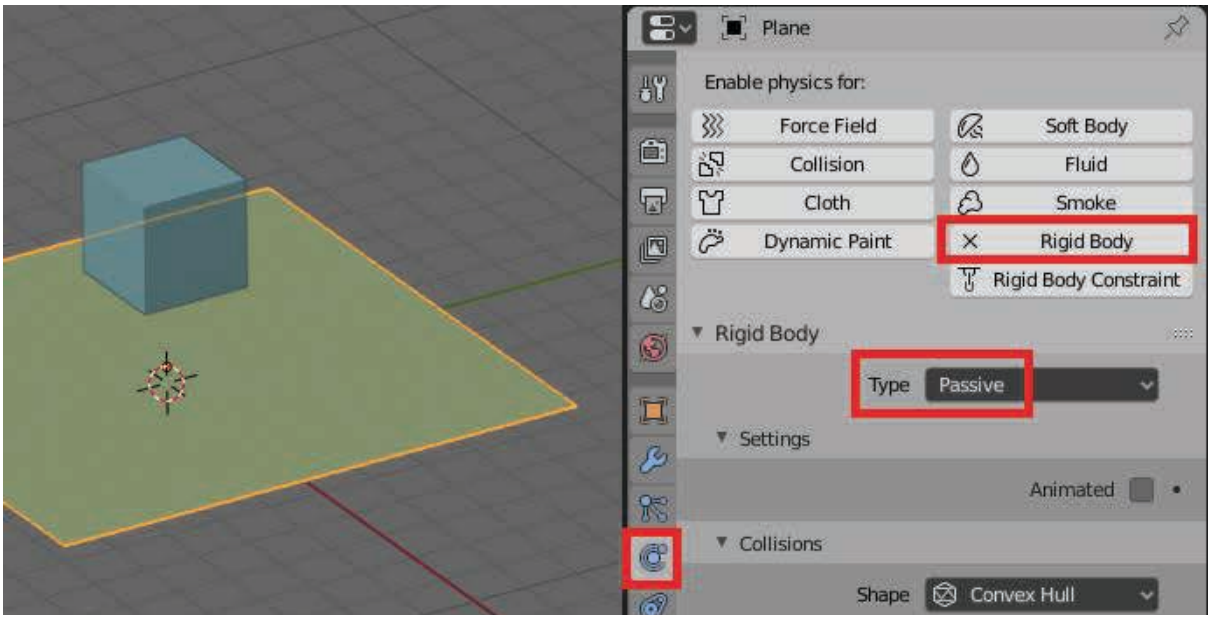


Quickly Fracture Stuff

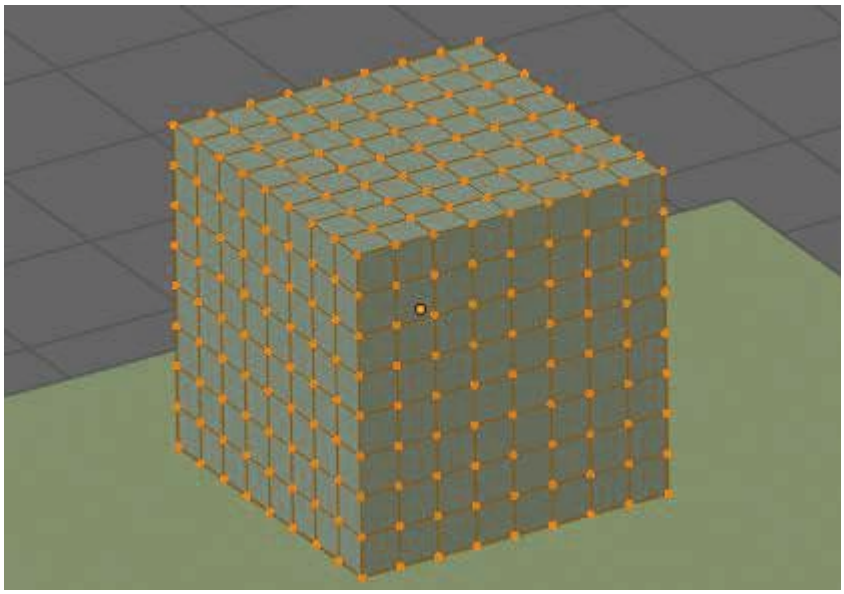
Fracturing and destroying things in 3D is fun, and can be quite satisfying! It can be difficult to get it all to work though. Luckily there's a quick method to start destroying in no time.

I've set up a simple scene with the default cube and a plane. Move the cube up so there's a bit of distance between the cube and the plane.

Set the plane to be a Passive Rigid Body.

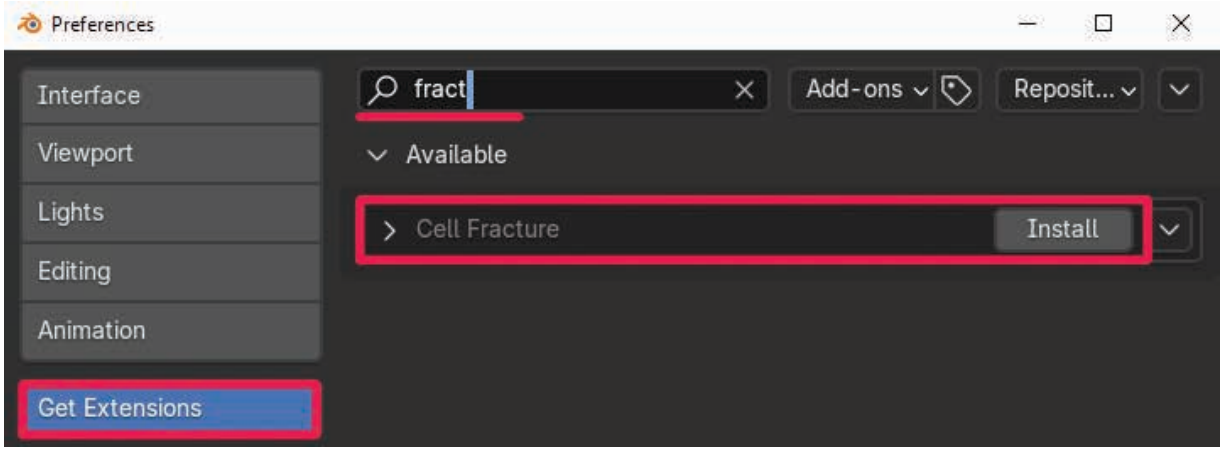


Make sure the object you want to destroy has enough geometry. If necessary subdivide it a few times in Edit Mode. **Right-click** and choose "Subdivide", then press **Shift+R** a couple of times to repeat the subdivide action.

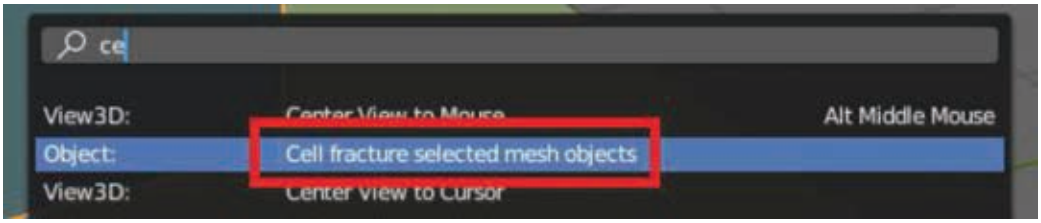


The cube has been subdivided a few times.

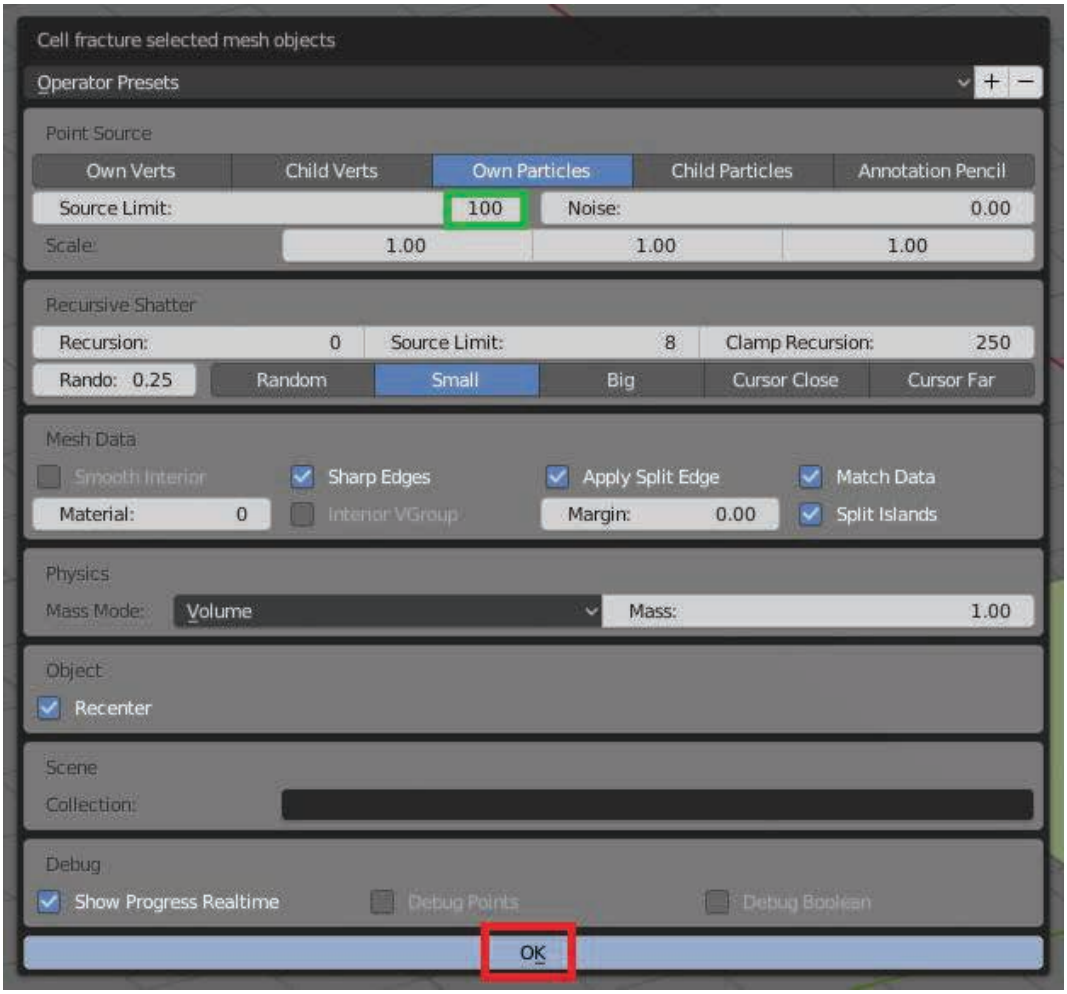
Install the **Cell Fracture** add-on in Preferences in the **Get Extensions** tab.



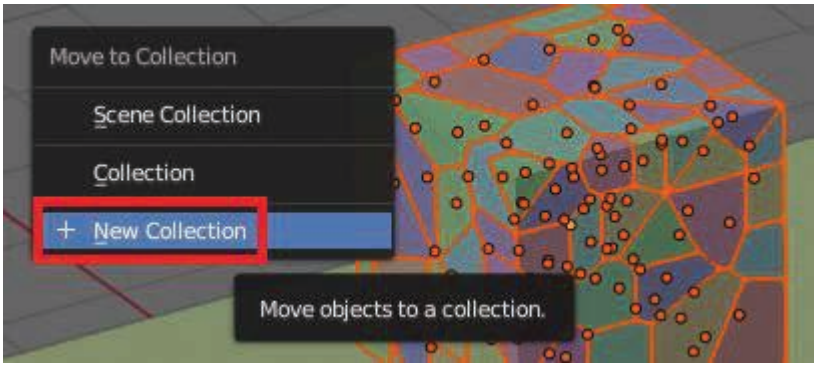
Select the cube in Object mode, **Press F3** and search for the "**Cell Fracture**" tool.



Select it and **press Enter**. Now you'll get this options pop-up. In fact the defaults are fine, you can just click on **OK**. But you can add the amount of shards in Source Limit if you like.



It may take a moment for the add-on to shatter your object into pieces. They will be selected once it is finished - take the opportunity to add them to a Collection by **pressing m** and choosing "**+ New Collection**". Name it something like "Shards" so it's easier to select all these pieces later.

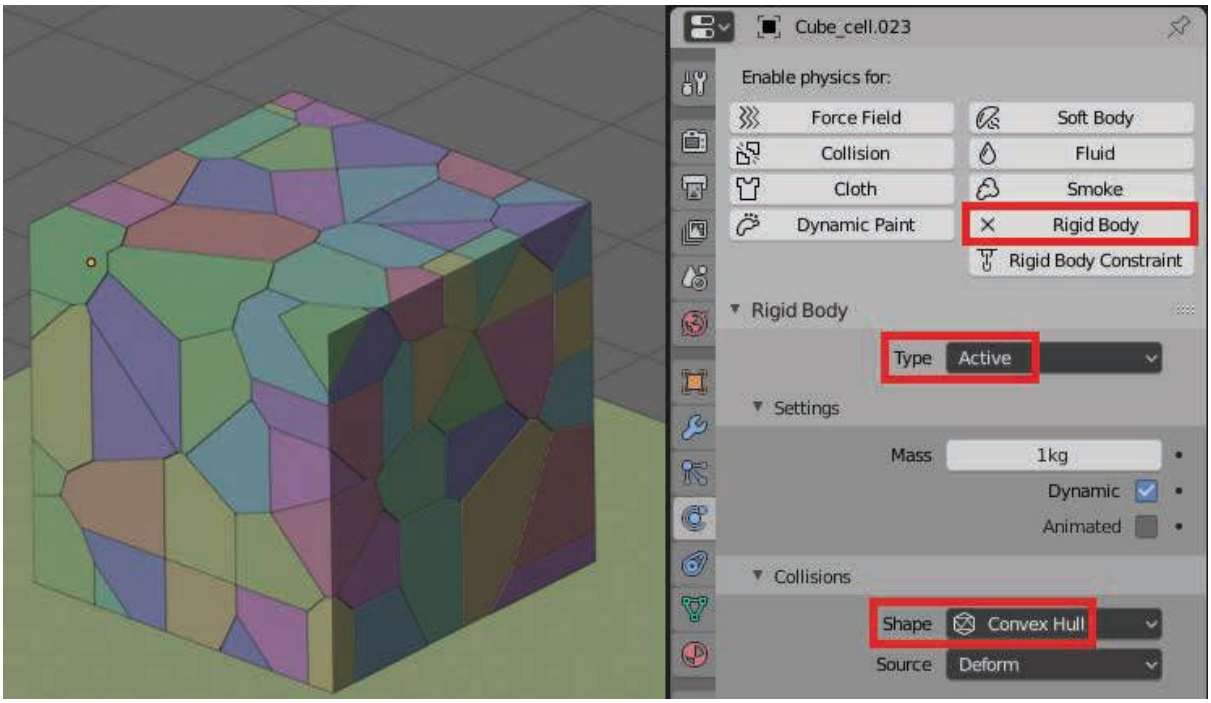


Now hide these pieces (**press h** or uncheck the eye-icon for that Collection in the Outliner) and delete the original cube (which is still there, hidden underneath all these shards). You don't need it anymore.

Unhide the shards by **pressing alt+h**.

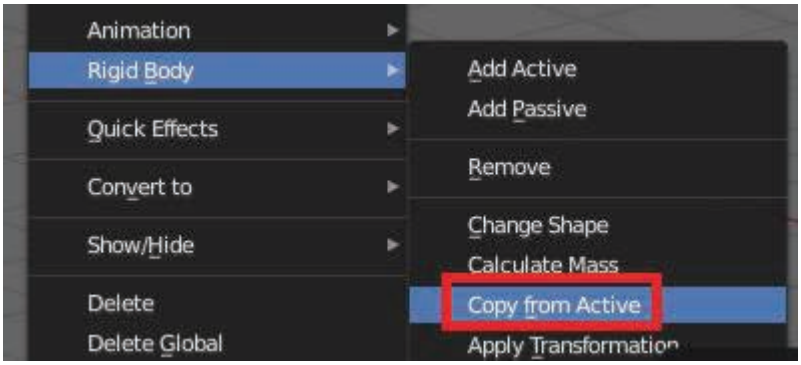
Select only one piece which is easy to select like a corner piece. In the Rigid Body tab **click on Rigid Body**, set it to Active and Collision Shape to **Convex Hull**.

In most cases the default settings work very well, in fact you may be tempted to set it to “Mesh” for more accuracy, but you’ll find all the pieces explode away from each other as soon as you play the simulation.



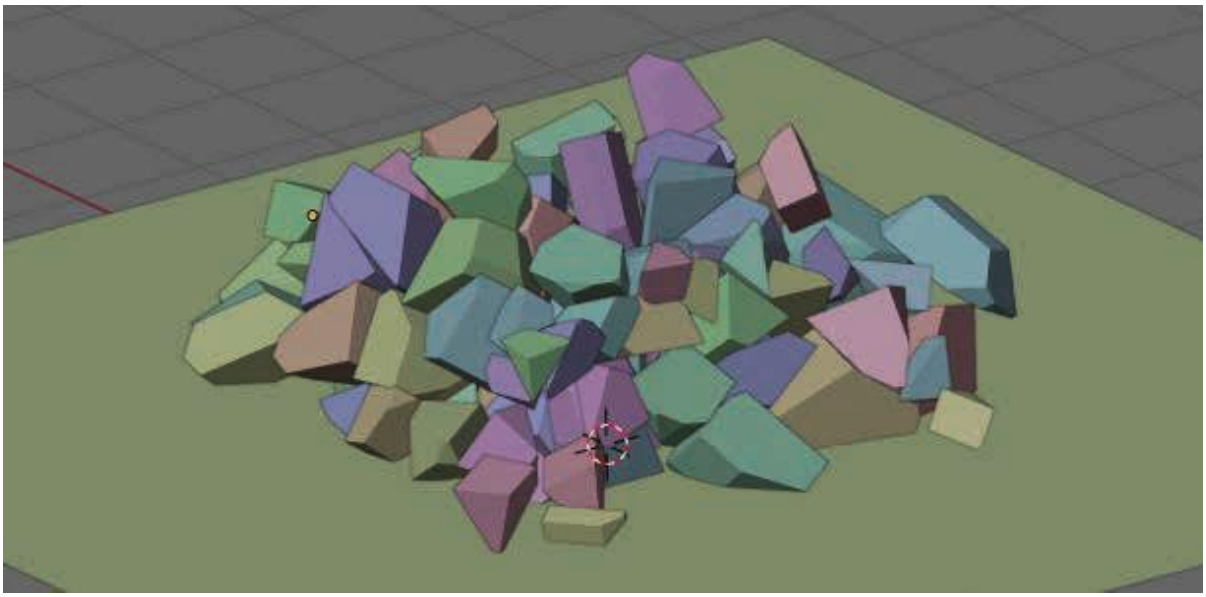
Now to select all the pieces, right-click on the Collection in the Outliner and choose “Select Objects”. This way all the pieces are selected. The corner piece is still the Active object (outlined in orange).

Go to Object > Rigid Body > **Copy from Active** to copy the Rigid Body settings from the Active object to all the other selected pieces.



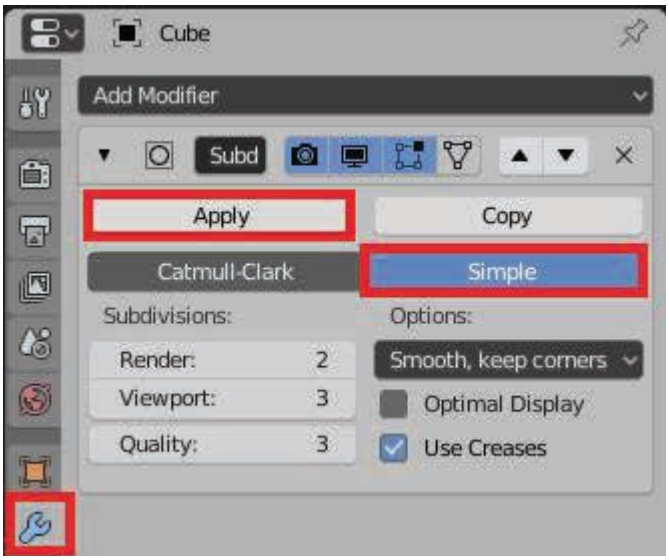
Play the simulation by pressing the **Spacebar**.

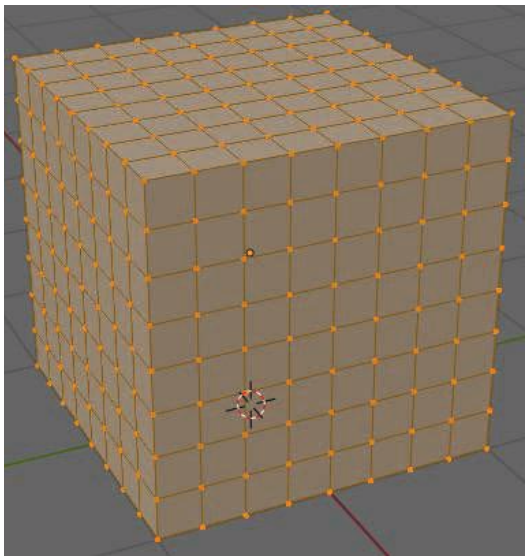
The cube falls and shatters on impact on the floor. Nice! There's something impressive about seeing the computer calculate the gravity and Rigid Body effects so well. Now, go and destroy some more 3D models… have fun!



More Fracture Stuff!

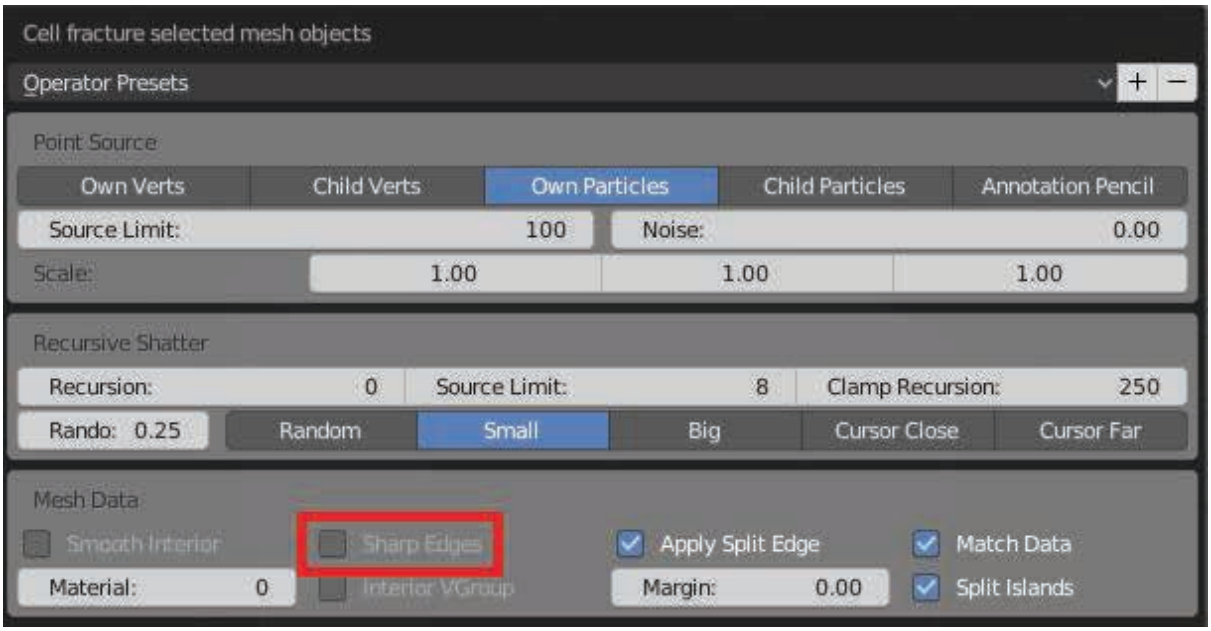
Add a **Subdivision Modifier** to a cube, Level 3, set it to simple and apply it.



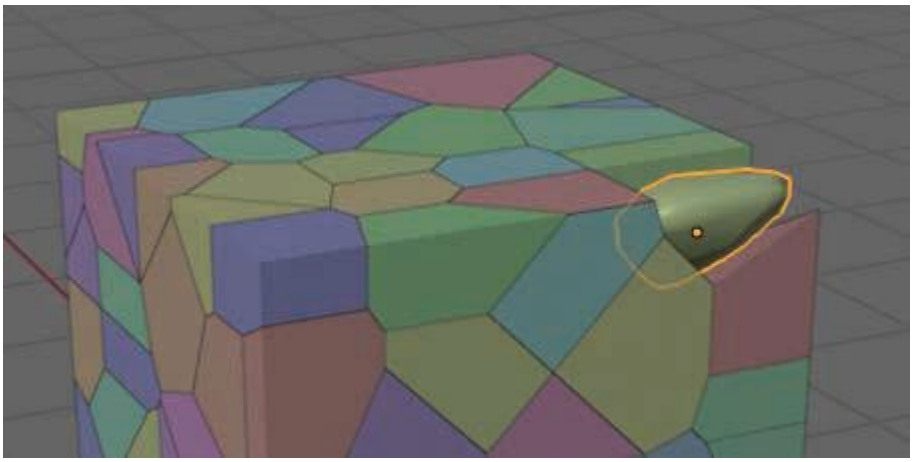


This creates a cube with more subdivisions but no smoothing.

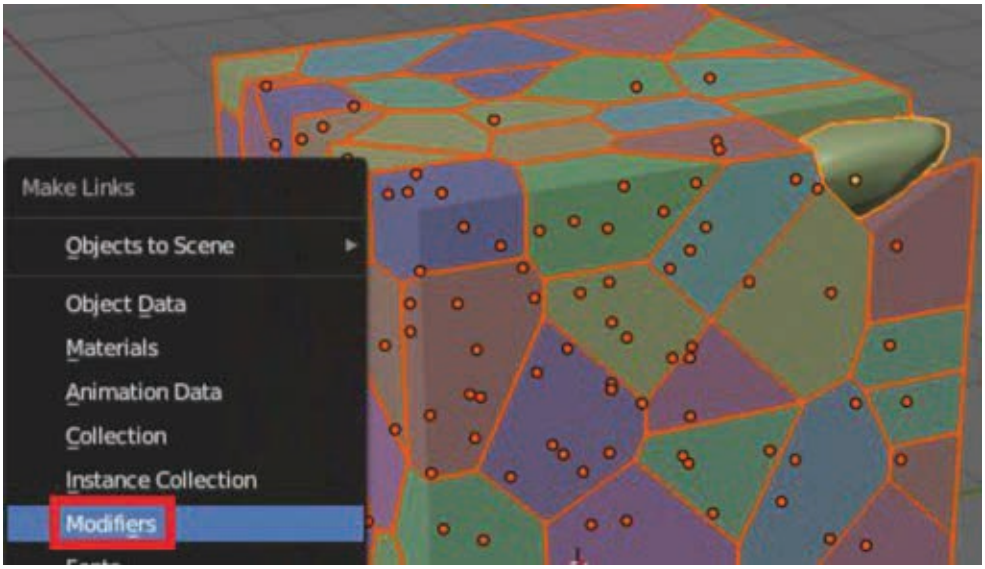
Like in the previous tip, use the Cell Fracture add-on on the cube. This time, turn off **Sharp Edges** in the options before **clicking on OK**.



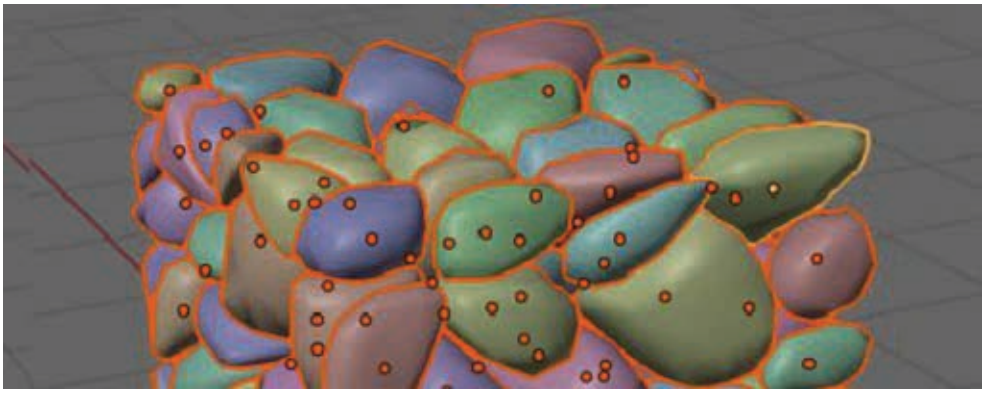
Find the fractured object on the next layer. Subdivide one of the parts by selecting it and **pressing Ctrl+1**, this adds a **Subdivision Modifier** at viewport level 1.



Invert the selection (**Ctrl+i**) and select the subdivided part, **press Ctrl+L** and choose "**Modifiers**" to copy the Subdivision Modifier from the Active object to the other objects.

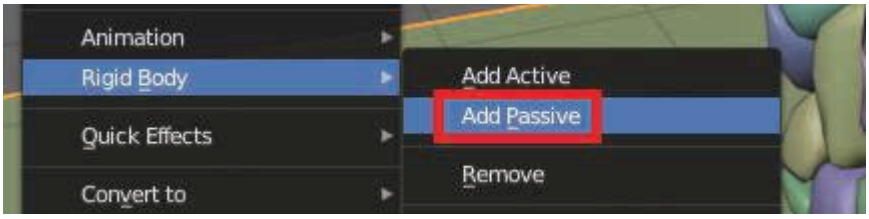


Right-click and set these objects to **Smooth Shading.**

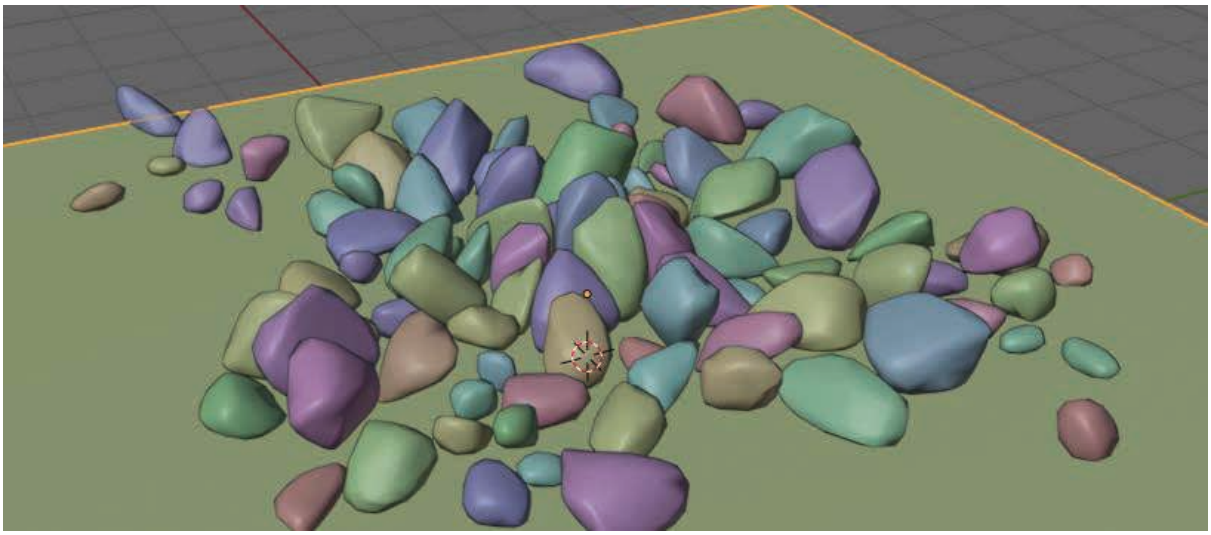


With all these pieces still selected, go to Object > Rigid Body > **Add Active.**

This gives all the selected objects the default Active Rigid Body settings. Create a floor plane and add a **Passive Rigid Body** to it.



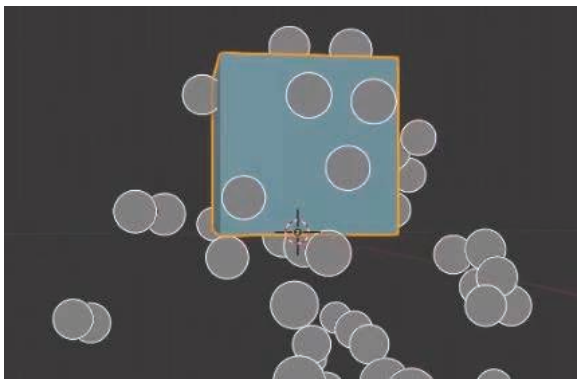
Play the simulation by **pressing the Spacebar**.



This is a quick way to create some rubble, or (if you don't add the Rigid Body settings) an organic looking stone wall.

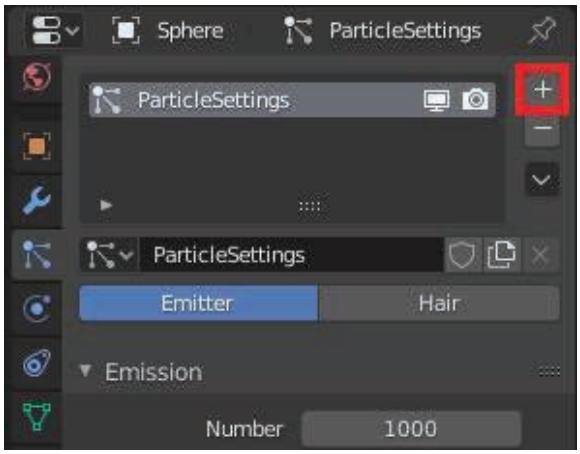
Slow motion particles

In some cases (like when you are making sparks for a slow-motion scene in a film) you need to be able to control the speed of the particles.

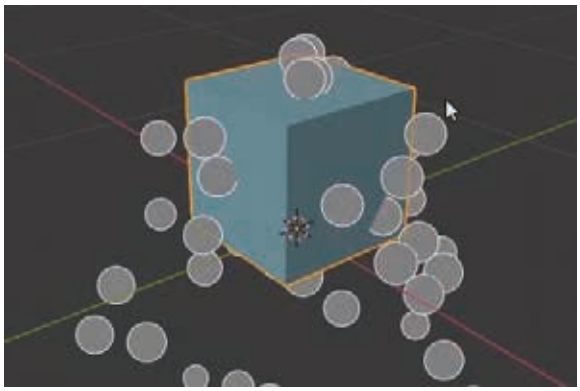


To demonstrate this, I'll quickly set up a simple particle system.

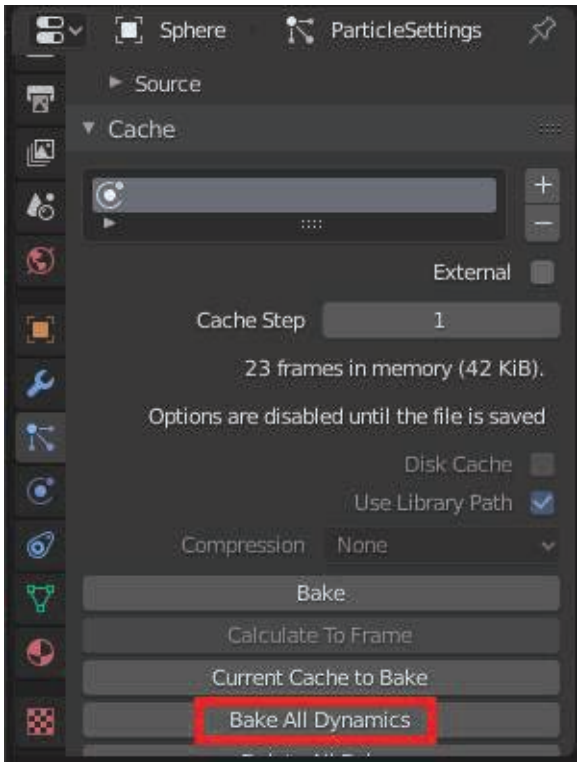
That is easy, we only need to select an object which will act as the emitter, and then in the Particle tab **click on the + icon**.



Now when we press the **Spacebar**, the cube starts to emit particles pretty fast.



In this case I bake the simulation by clicking on **Bake All Dynamics**.

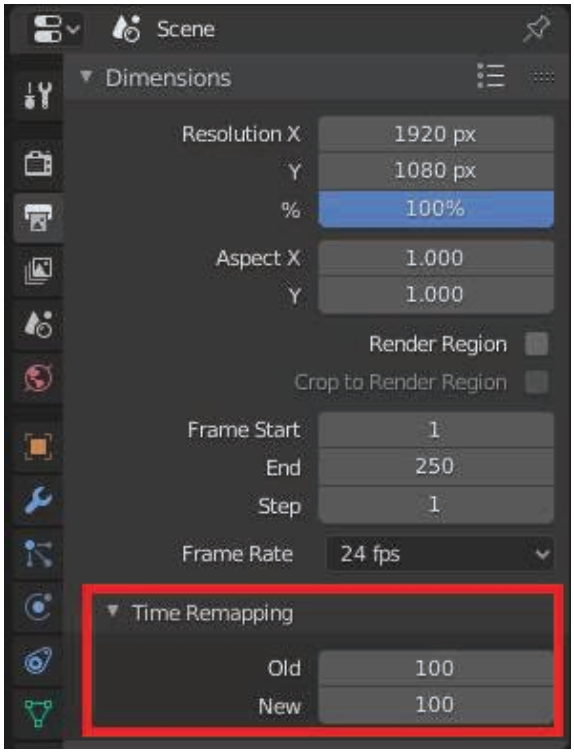


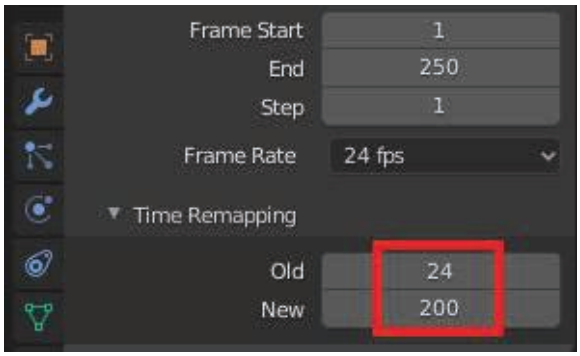
Now in the Output tab, under Dimensions look for the **Time remapping** options.

Both **Old** and **New** are set to 100 by default.

Set Old to your current frame rate and set New to a higher number (pictured below).

200 is really slow motion!

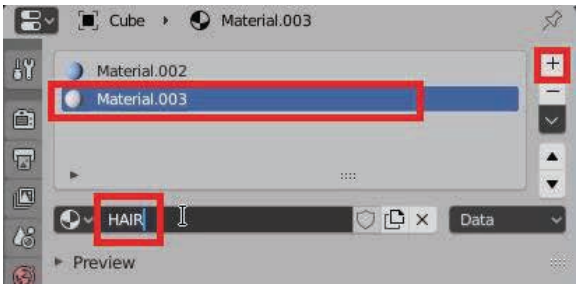




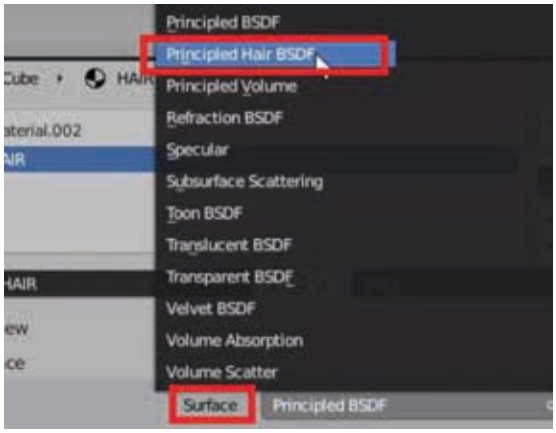
How to use the hair shader

This default cube has been given a nice haircut. But how do we give the hair and the cube separate materials?

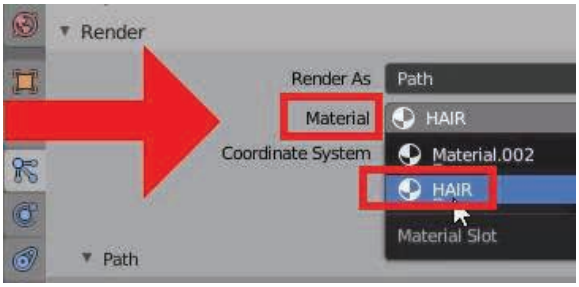
Add a second material to the cube by pressing the **+ icon** in the materials slot. Call it “Hair”.



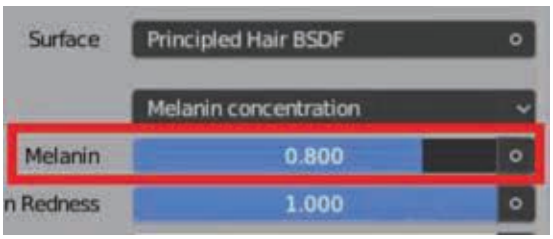
Change it to a **Principled Hair BSDF** shader.



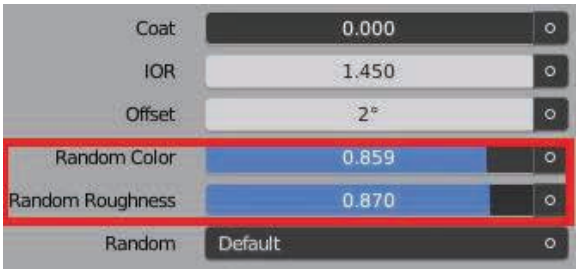
In the Particles tab change the **Material** to the “Hair” material in the Particle Render settings.



The shader itself is very easy to use. Some tips: change it to Melanin concentration for more options. Melanin is the amount of color pigment the hair has. Zero means white hair.



I like to increase the **Random color** and **Random Roughness** values as it increases realism. In reality every hair is a little bit different.



Click here to return to the index.

How to set the centre of weight in a rigid body

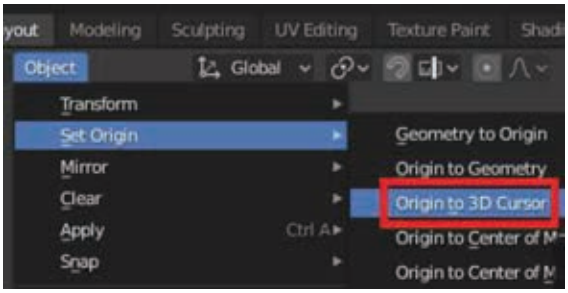
An active Rigid body has its centre of weight based on the object origin position. For the example in the image below left, I placed the origin of the object on the right side, so after falling down it leans to the right and falls over.

We can change the object origin to distribute the “weight” of the object differently.

To do this, first **press TAB** to go into Edit Mode, then select a vertex where you want to put the origin.

Press **Shift+S** and choose “Cursor to selected”. This moves the 3D cursor to the selected vertex location.

Then go to Object > Set Origin > **Origin to 3DCursor**.

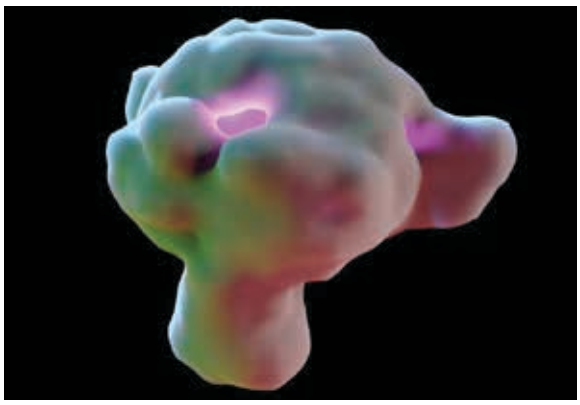


Now we’ve moved the object’s pivot point to that specific place, and playing the simulation we see that the object seems much heavier in that spot.

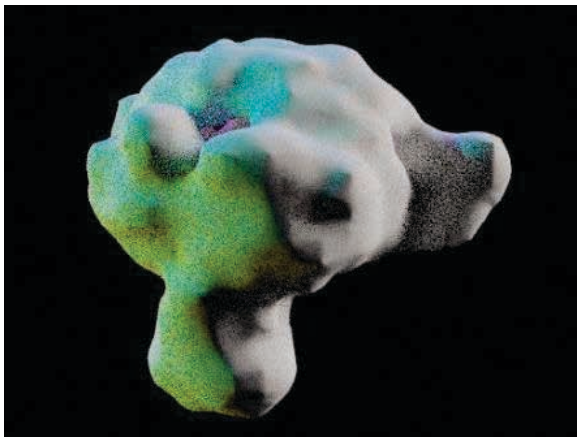
If you want objects to fall in a special way you can experiment with pivot point locations.

Volumetric Cloud

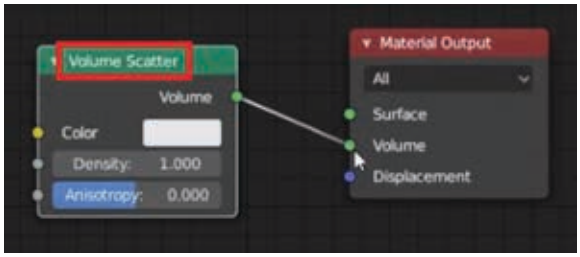
Continuing with the monkey head-shaped blob we created in the topic“Weird stuff with the Build modifier” on page 1473, let’s turn that into a volumetric cloud.



Model a cloud-like shape (here I'm using the blob from the previous tutorial).

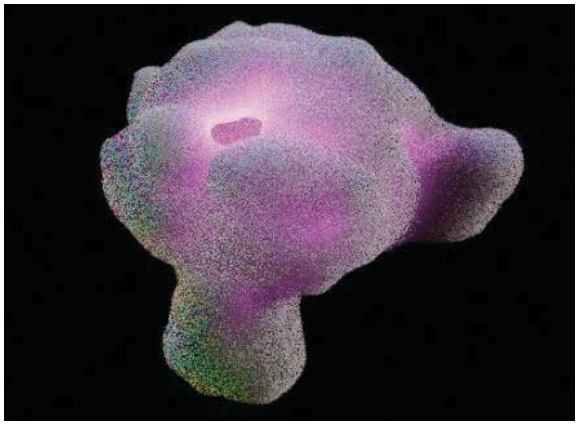


Remove the material input and add a **Volume Scatter node**.



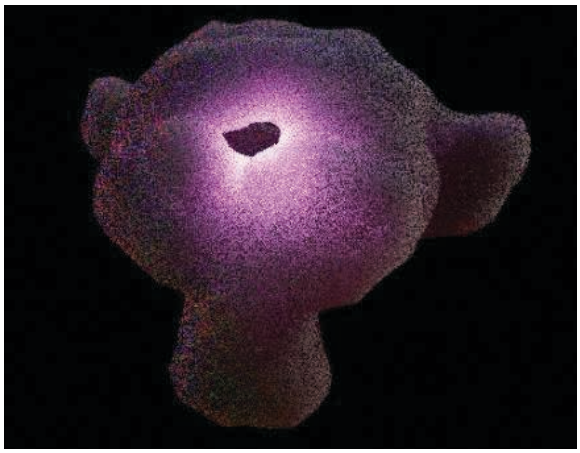
The **Density** setting controls how translucent the cloud is.

Here I just add a light inside because... it looks cool.



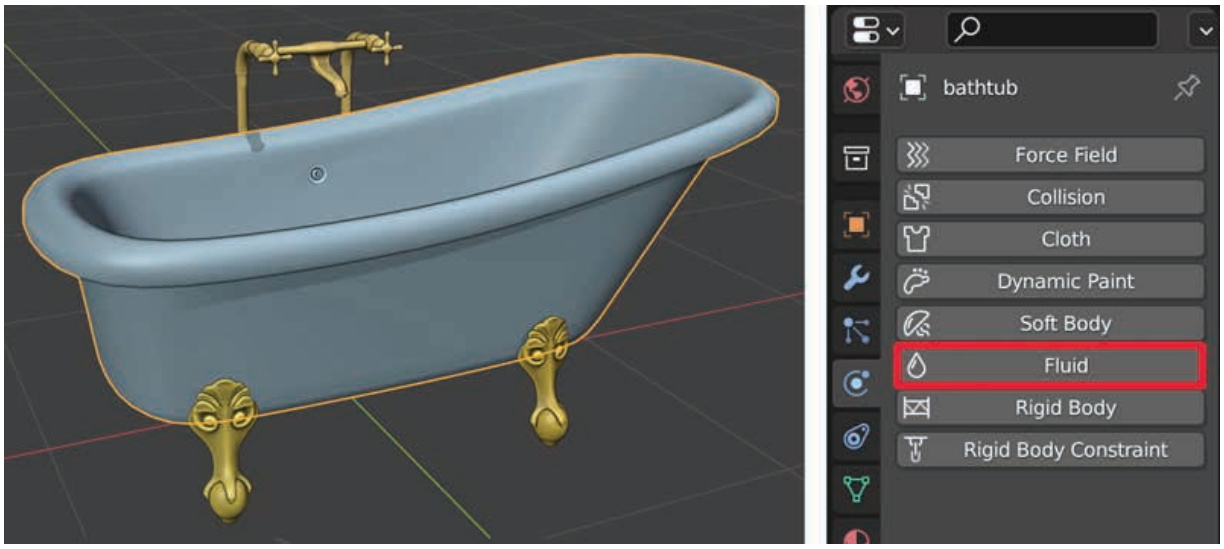
Playing with the density value of the Volume Scatter node.

This is in Cycles, not Eevee.



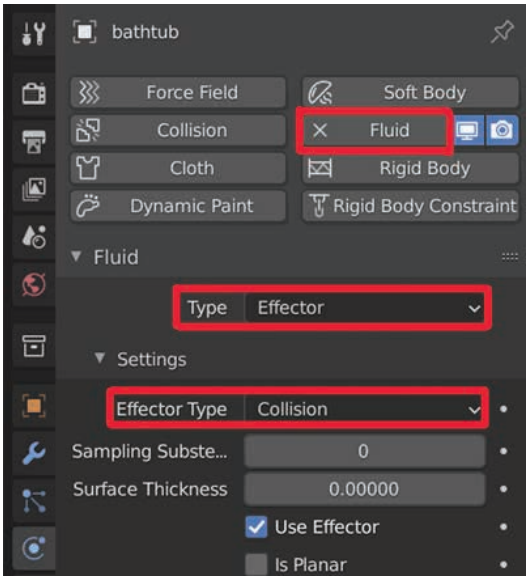
Fluid Simulations part 1 (basics)

In this tutorial we'll create a simple fluid simulation using the new Flip fluids system. It's easier than you might think! However, there are some things you need to know in order to achieve a good result. First, we need an object that receives the fluid. Select it, then click on **Fluid**.



(Bathtub model by bryanajones on Blendswap https://www.blendswap.com/blend/17052)

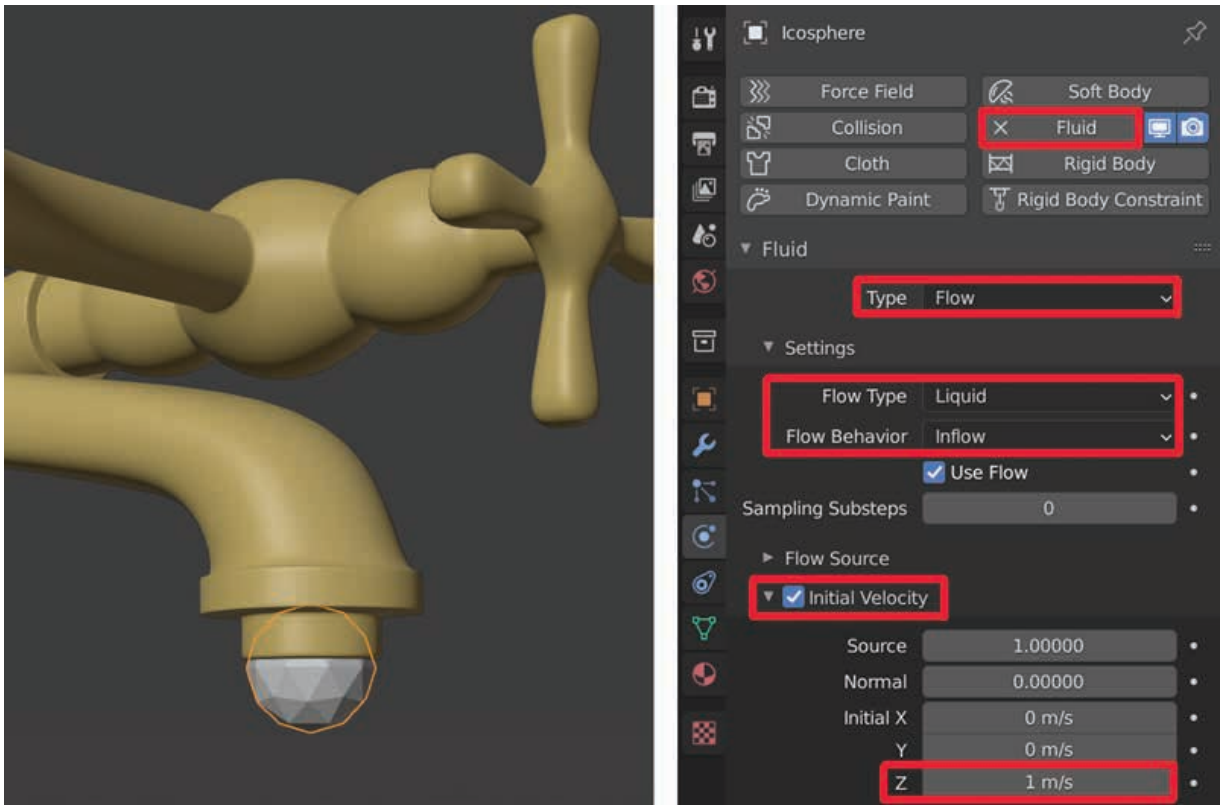
Use these settings: **Type: Effector** and **Effector Type: Collision**.



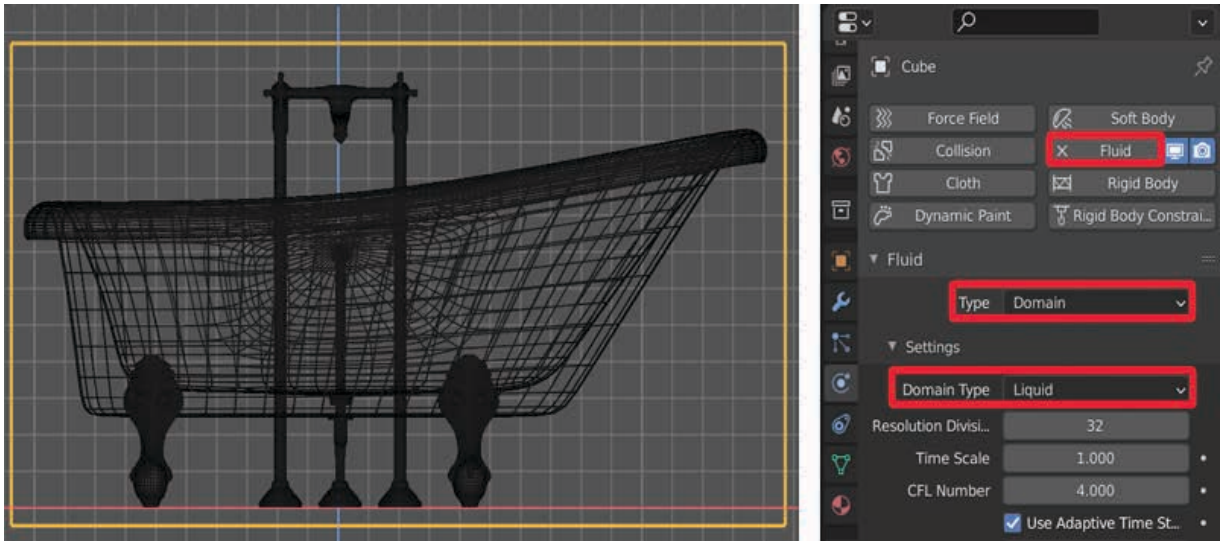
Second, we create the object that emits the liquid.

Press Shift+A and add an **Icophere**, scale it down and move it to where the water should be coming from.

Use these settings: **Type: Flow. Flow Type: Liquid. Flow Behaviour: Inflow**. Enable **Initial Velocity** and set **Z to -1 m/s**. This will give the water some speed as it's going down.

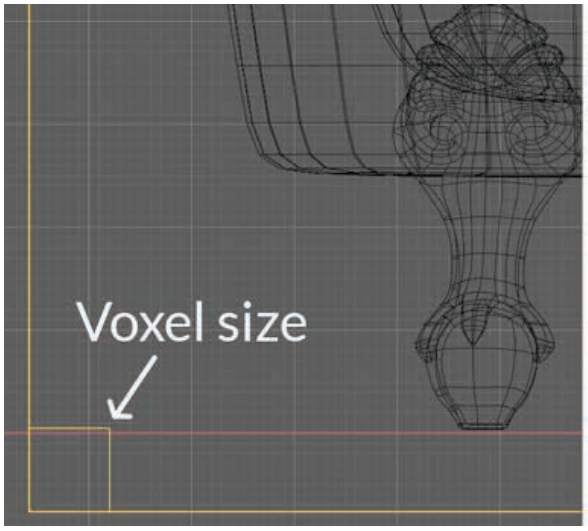


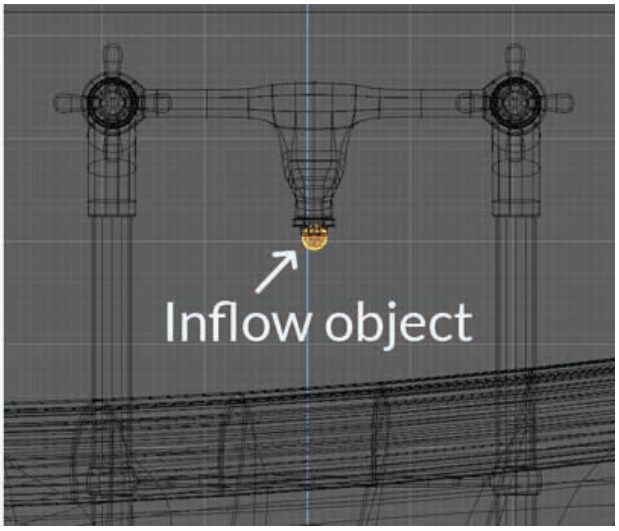
Add a **Cube** and scale it to fit just around the scene. **Press Z** and choose **Wireframe** so you can easily see all objects. This Cube will be our Domain, the area in which the simulation takes place. Click on **Fluid**, then set it to **Domain**. Use these settings: **Type: Domain** and **Domain Type: Liquid**.



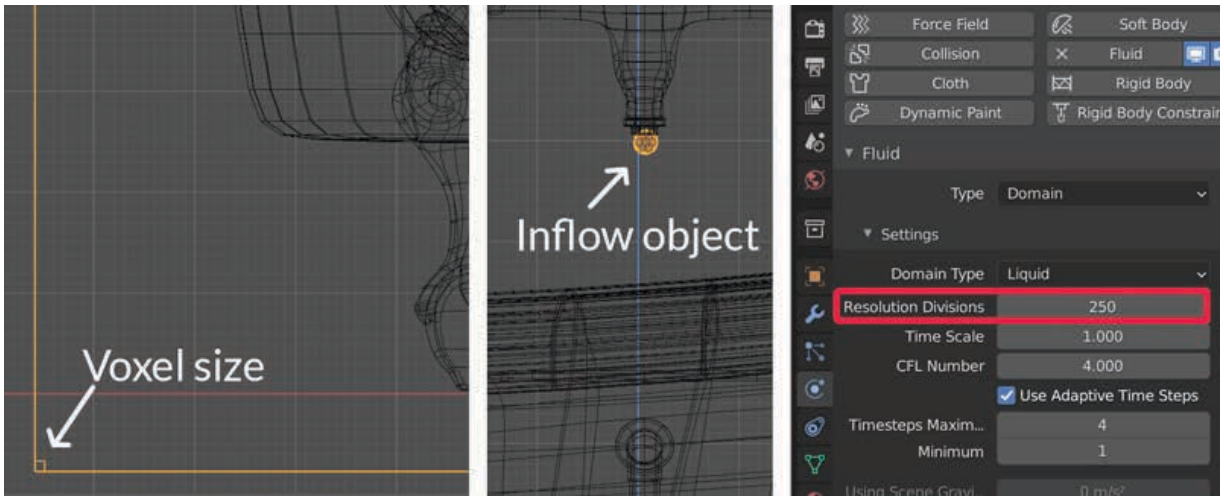
With a simple liquid simulation, if you pressed the **Spacebar** now to start the simulation you would already see it working (in Wireframe mode at least). But in this case, nothing happens...

Why? Some explanation about **voxel size, domains and inflow objects**: The little cube you see at the bottom corner of the domain, is the size of the voxel (the little cubes the liquid is made up, so to speak). **If your inflow object is smaller than this voxel, you won't see any liquid.**





The only thing you can do is either **increase the size of the inflow object**, or **decrease the size of the voxel** by increasing the **Resolution Divisions value**. Unfortunately, that also means that everything takes longer to calculate. But for this situation where the inflow object is tiny compared to the domain, that's our only choice.

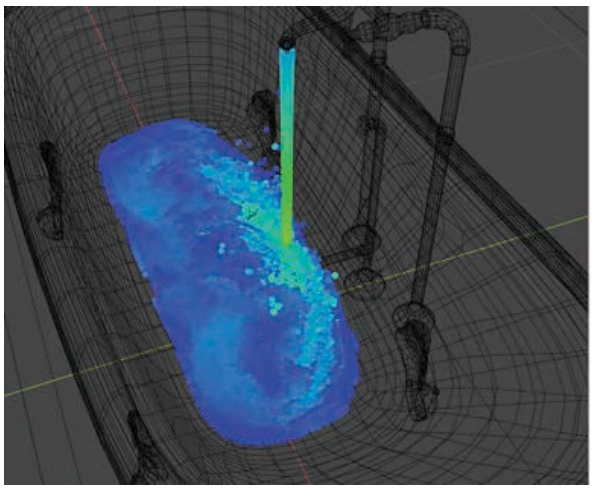


As you can see in the above screenshot, **setting the Resolution Divisions to 250** makes the voxel size smaller than the Inflow object.

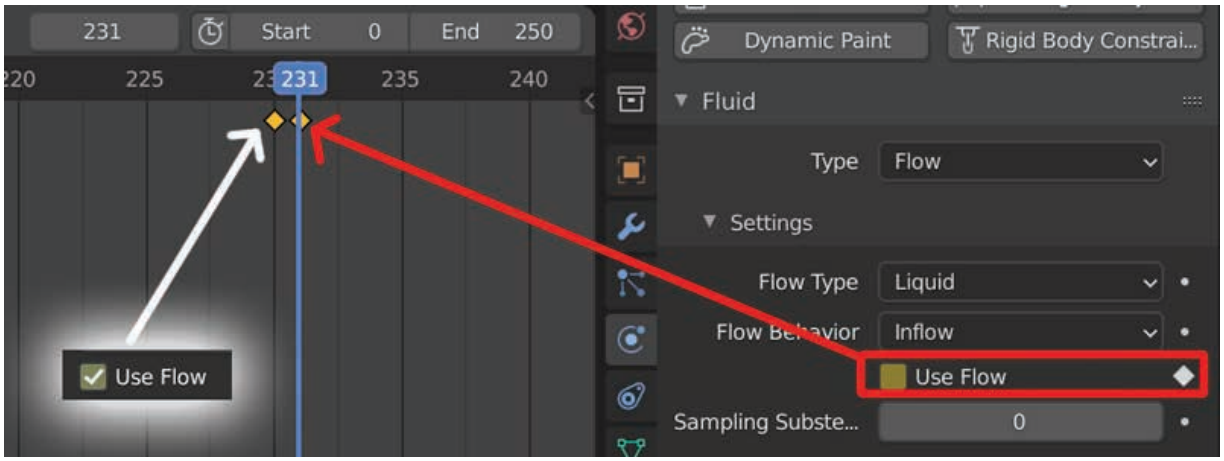
If you play the simulation from frame 1 on the Timeline, it will probably be quite slow, but you can see the simulation working now. In the next part, we will finetune the simulation.

Fluid Simulations part 2 (finetuning)

Another thing we need to do is to stop the simulation at some point, so the bath doesn't flow over, and then convert the particles to a mesh so that we can give it a material and render the result.

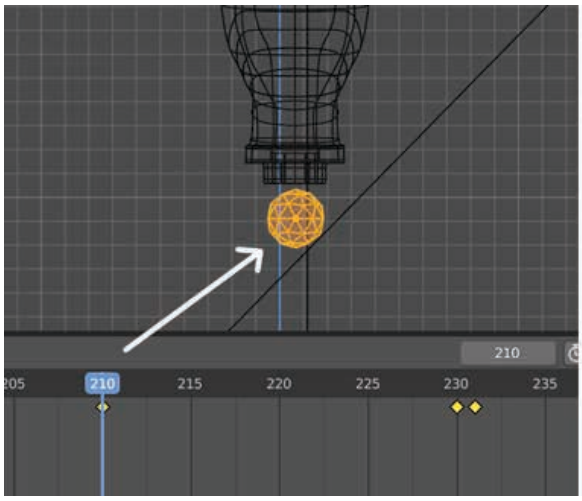


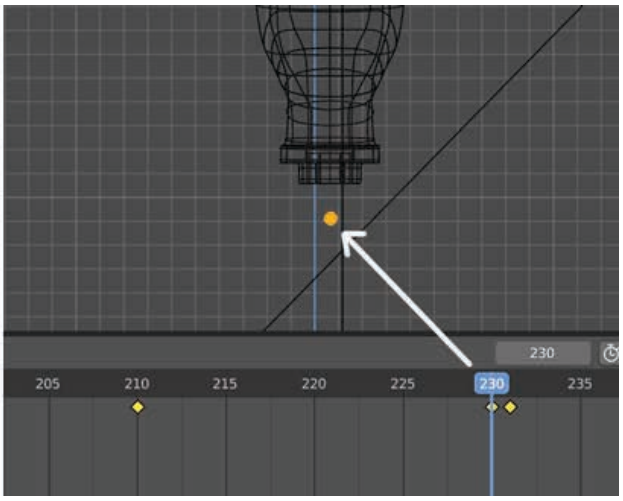
First of all, we can create keyframes for the **Flow** to turn it off at some point. You can do this by hovering the mouse over the Use Flow checkbox and pressing i to set a keyframe. Then go to the next frame on the Timeline and **uncheck that Use Flow checkbox** and set another keyframe.



This turns off the flow rather abruptly, so to make it more realistic we can **keyframe the scale** of the Flow object so that it scales down, reducing the flow in a more realistic way before stopping.

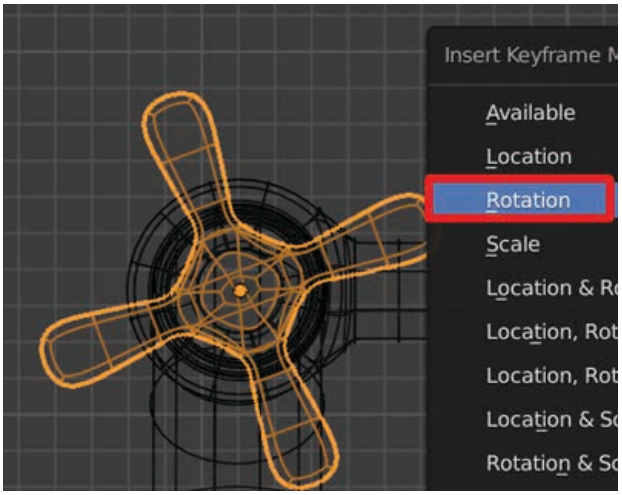
Go back twenty frames and select the **Icosphere** at the mouth of the faucet and **press i** to set a keyframe, choose **Scale**. Go to the frame where the Use Flow checkbox is keyframed and scale the Icosphere down, then set another keyframe for Scale.



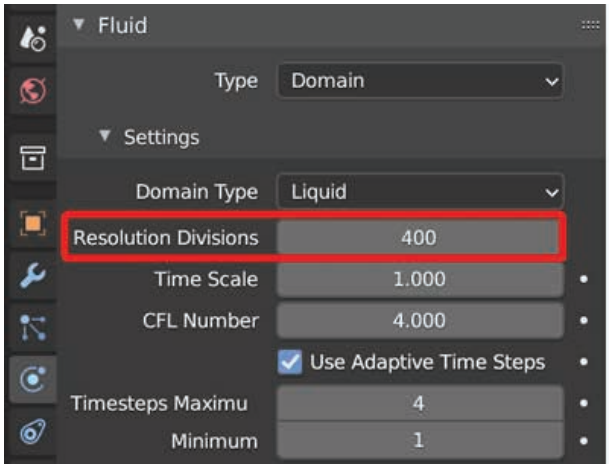


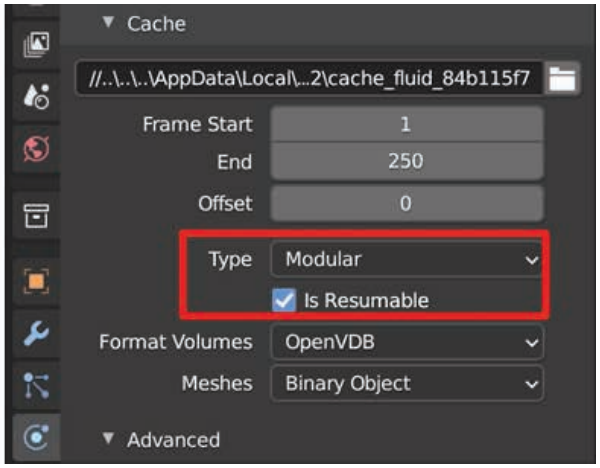
You can also animate the rotation of the bath faucet handle.

You can start this animation a bit earlier (5 frames or so) than the scaling down of the Inflow object, to make it a bit more realistic.



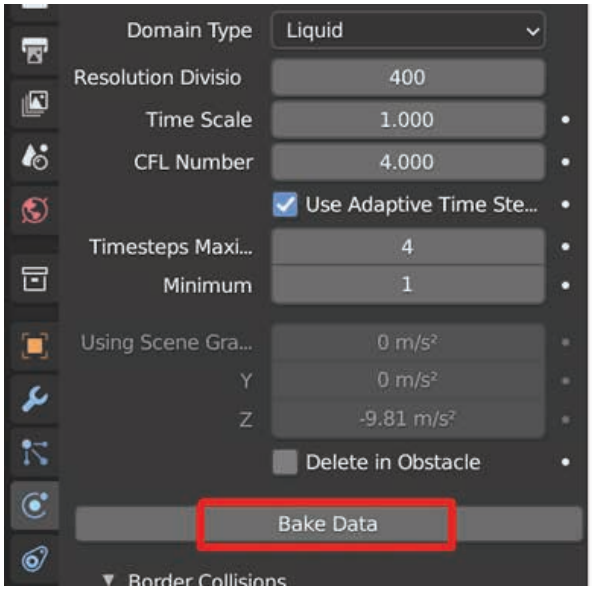
To get a better quality simulation, we need to increase the resolution. **Increase the Resolution Divisions** to for example **400** (or more… if your computer can handle it!). With the Domain selected, go to **Cache** and set Type to **Modular**. Enable the **Is Resumable** option.

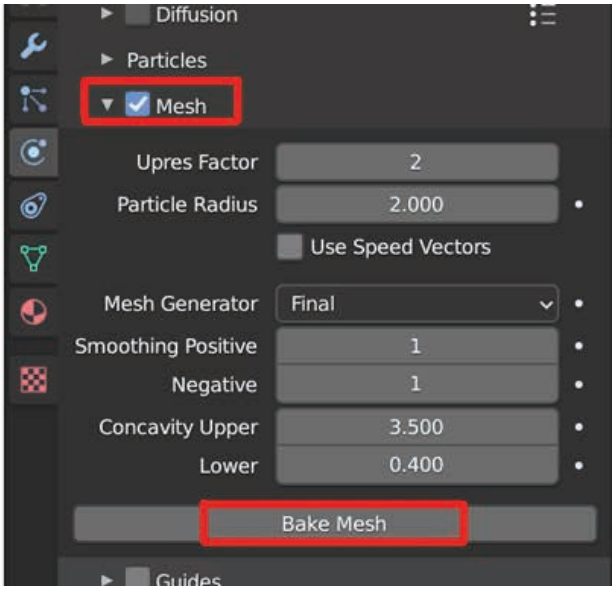




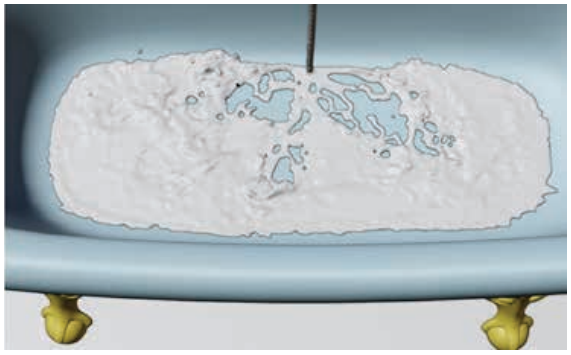
Scroll up a bit to find and click **Bake Data**. This will take some time to bake. Like, a lot of time. The good news is, by enabling "Is Resumable" you can always pause the baking process by pressing the Escape key, and then resume it at a later time (when you're taking a break for your computer, for example).

So you don't need to bake it all at once. After baking is finished, enable **Mesh** and click on **Bake Mesh**. This will create the geometry for the water so we can give it a material and render it. Here you'll need to be patient once more…



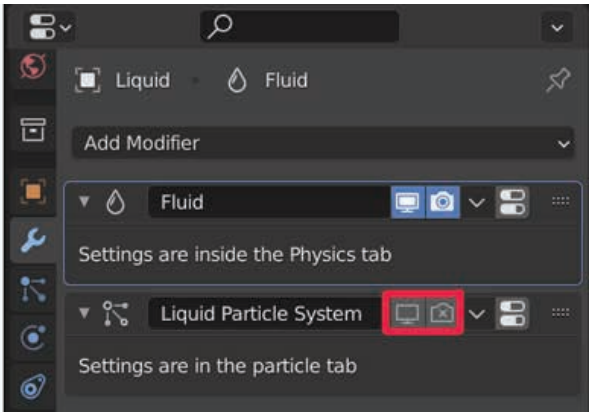


After baking, you can exit Wireframe mode (**press Z** > Object) to see the water geometry.



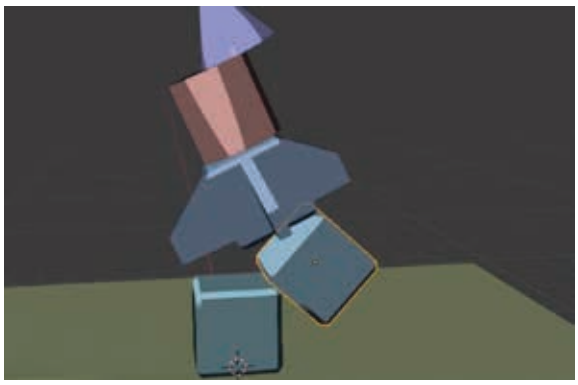
Now you can add materials to it. I'd recommend saving a backup of your scene, and then either removing the Icosphere that you used as the Inflow object or **turning its render visibility off** in the Outliner.

Also, turn off the viewport and render visibility of the Liquid Particle System in the modifier stack.

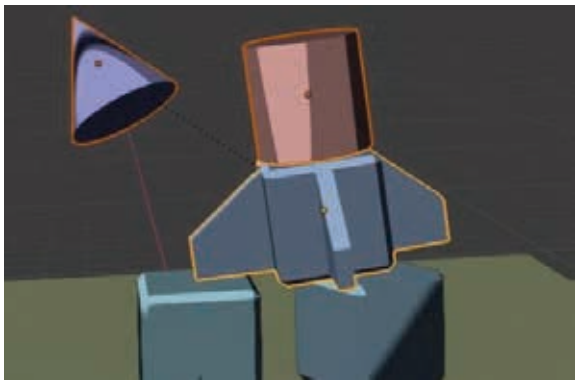


Connecting things for simulations

Sometimes you need to connect objects for a simulation, like this simple robot made up of 3 objects.

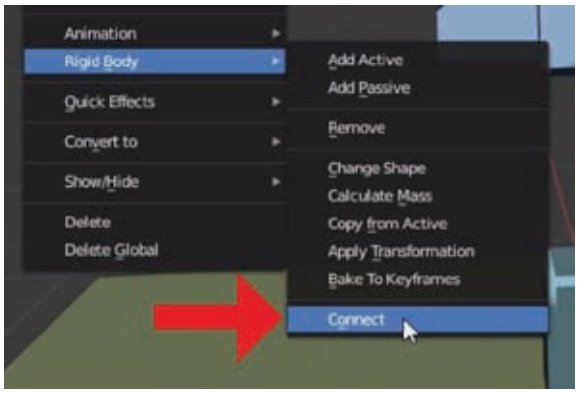


Just parenting objects with **Ctrl+P** can give some weird and unexpected results. In this case the top cone keeps moving in a glitchy way.

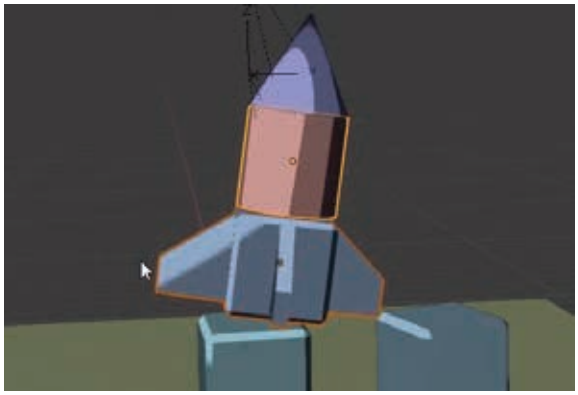


Instead select two objects, go to Object > Rigid Body > **Connect**.

Repeat for other objects to connect all the objects that need to stay together this way.



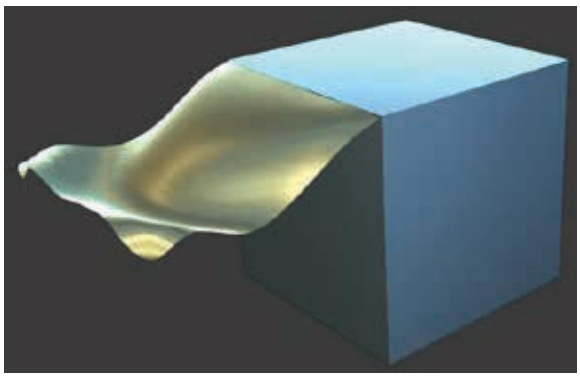
Press **Spacebar** to play the simulation. Now everything works as it should.



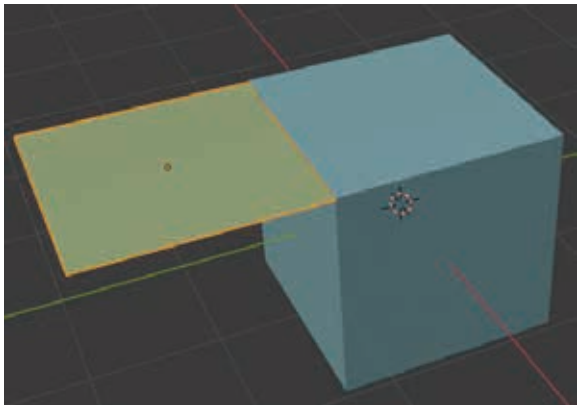
Click here to return to the index.

Heroic Default Cube

Instead of deleting the default cube, let's give it a little cape and make it look heroic by using a physics simulation with wind.

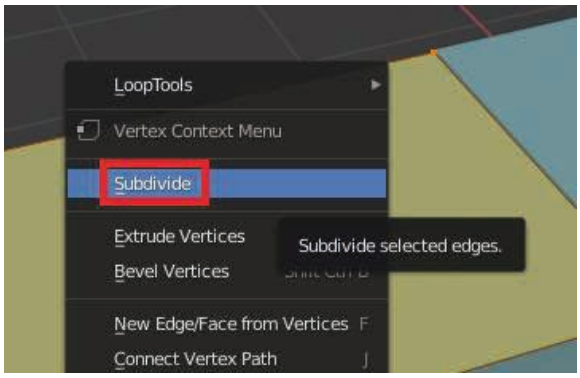


Press Shift+A and add a Mesh > **Plane**. Move it in place with keyboard shortcut **G**. If you **press Shift+X**, you exclude the X-axis from the movement, so you can position it more easily (since you only need to move it along the Z and Y axis in this case).

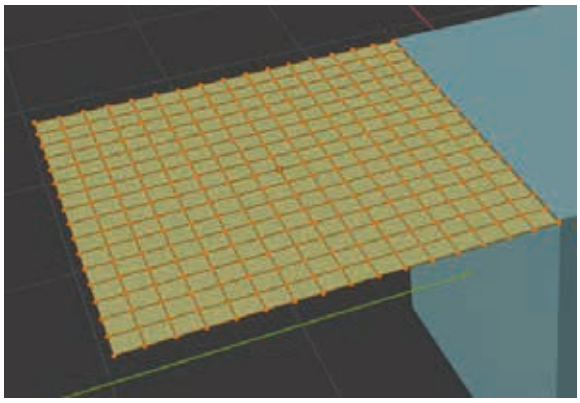


Press TAB to go into Edit Mode and **press A** to select all vertices.

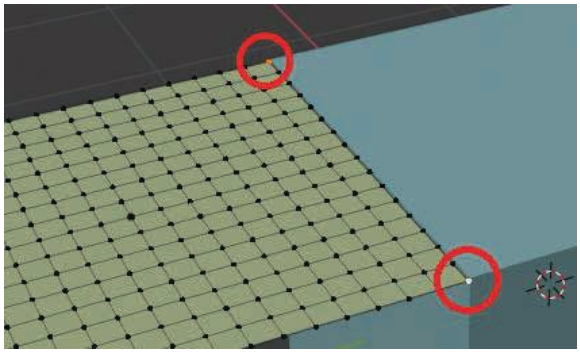
Right-click and choose **Subdivide** to subdivide the plane.



Press Shift+R 3 times to repeat the subdivision 3 more times.

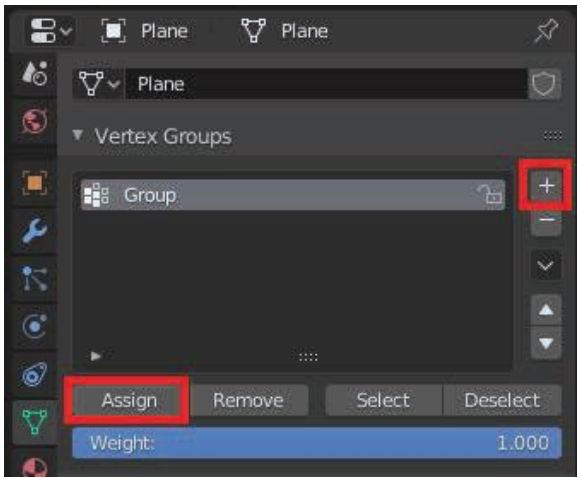


Select these two corner vertices of the Plane.

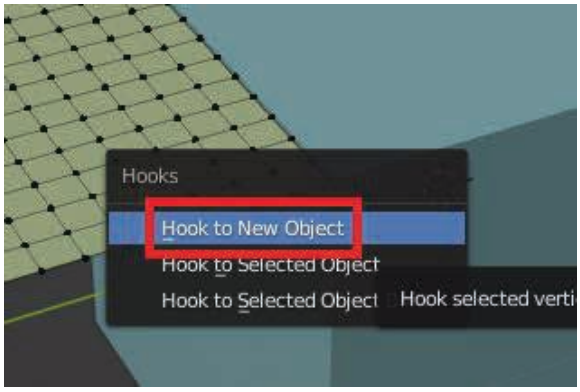


Add a Vertex Group by **clicking the + icon** and click on **Assign**.

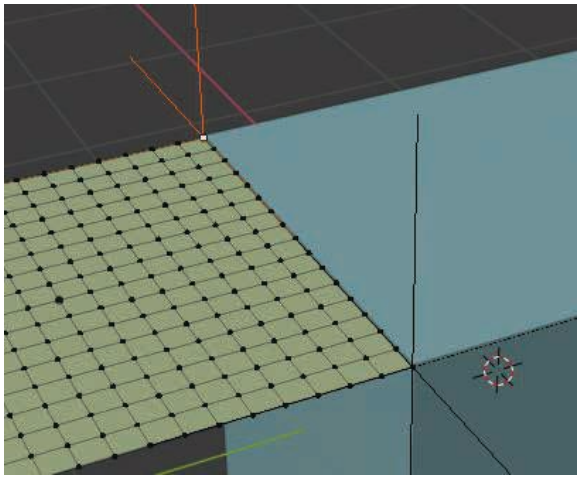
Now those two corner vertices are in a Vertex Group.



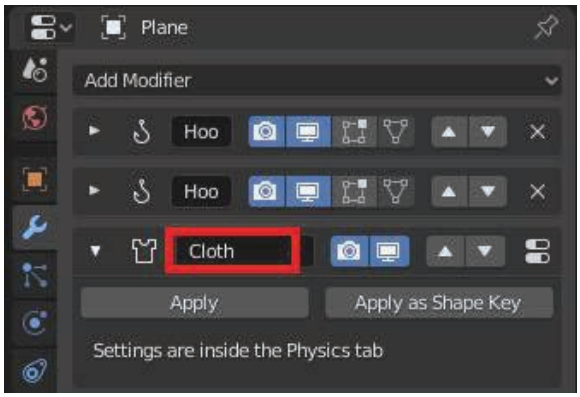
Select just one of the vertices and **press Ctrl+H** for the Hook menu, choose Hook to New Object.



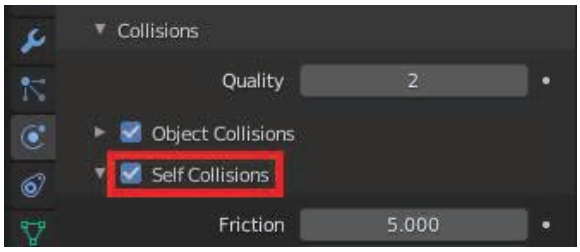
Repeat for the other vertex so they both have a Hook in place.



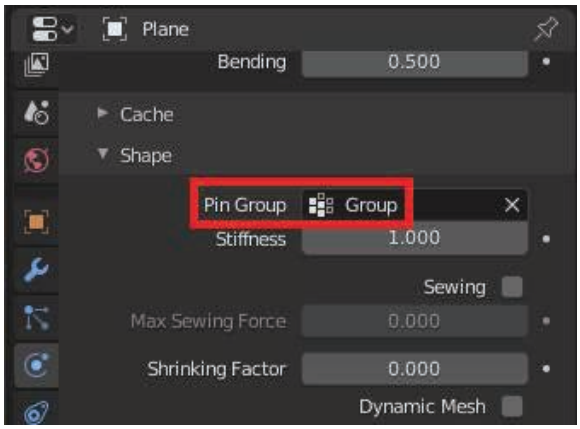
Press TAB to go to Object Mode, select the plane and add a **Cloth modifier** to the plane.



Turn on **Self Collisions** in the **Physics** tab.

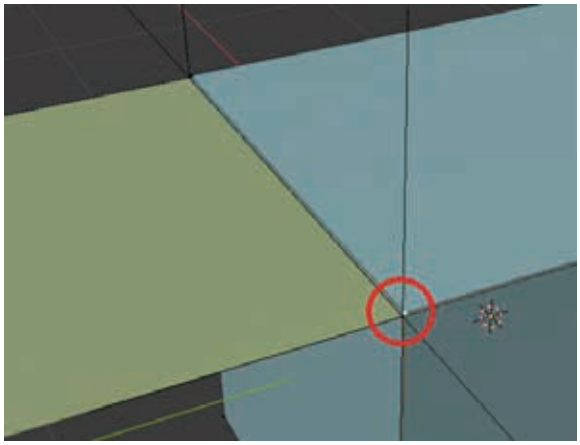


Under Shape, for the **PinGroup** choose the Vertex Group we made earlier.



Select the Cube.

In Edit Mode, select one corner vertex of the Cube (not the Plane!).

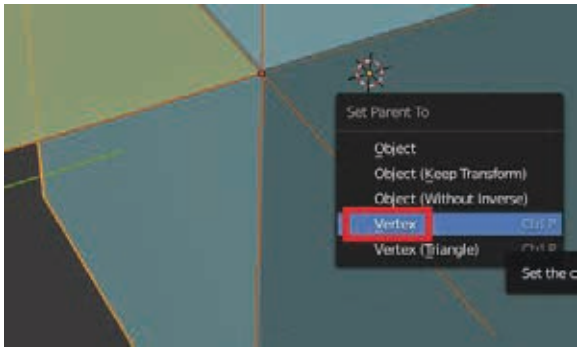


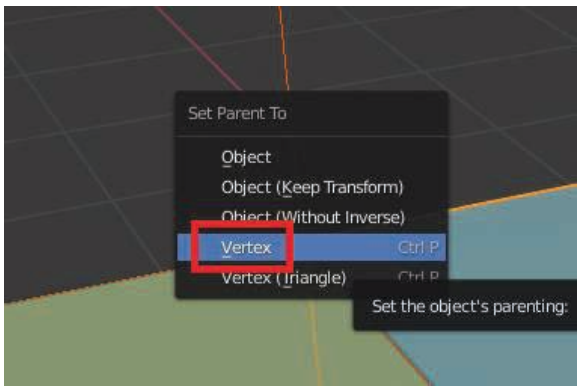
Press TAB to go to Object Mode.

Select the Hook at that corner, then the Cube.

Press Ctrl+P and choose **Vertex**. Repeat for the other Hook.

Select the hook, then the cube, then **press Ctrl+P** and choose **Vertex**.

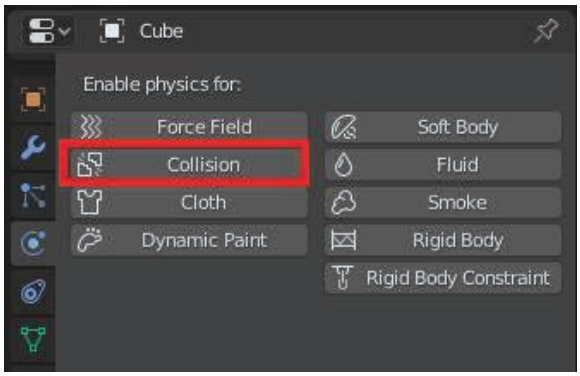




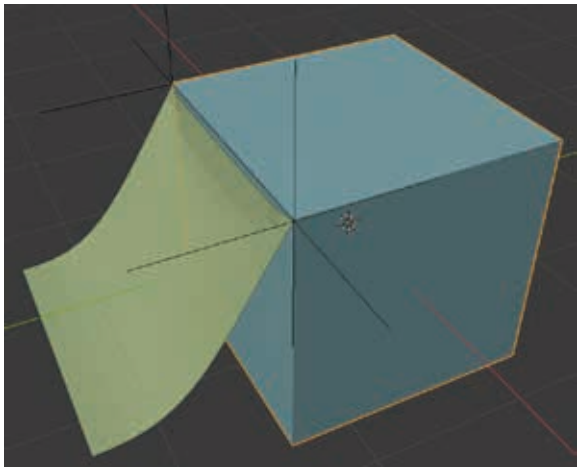
Select the Cube.

Enable Collision physics for the cube by **clicking on the Collision icon**.

This will ensure that the cloth doesn't go through the cube.



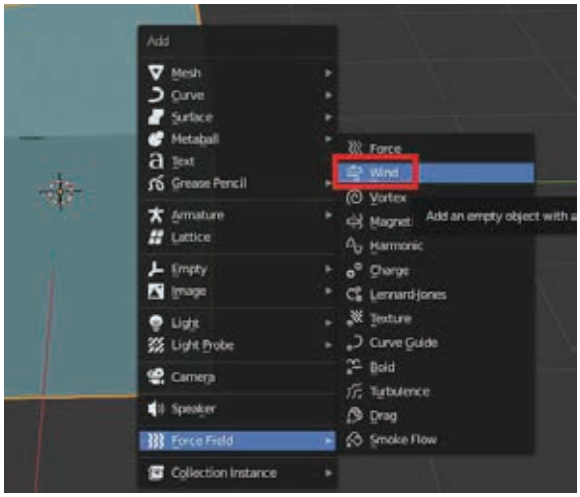
Press Spacebar to play.



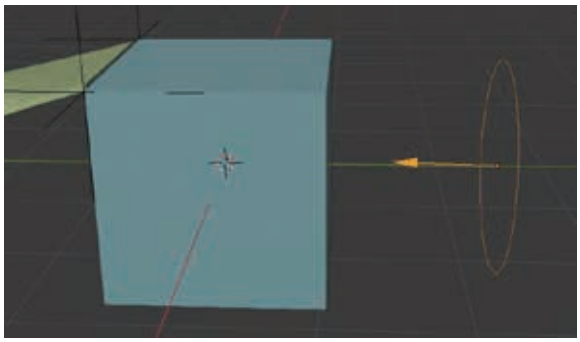
You will notice that the plane becomes cloth-like and falls down until it touches the side of the cube.

While the simulation is playing, you can also move the Cube with keyboard **shortcut G** and see how the cloth follows.

Add a Wind Force Field by **pressing Shift+A** and choosing **Force Field > Wind**.

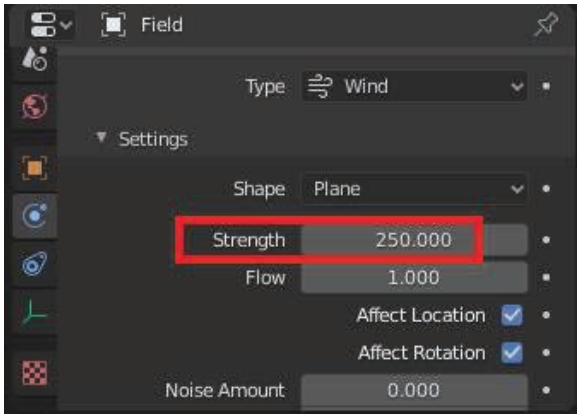


Move it along the Y-axis by **pressing G+Y** and rotate it along the X-axis by **pressing R+X** until the arrow points at the cube.

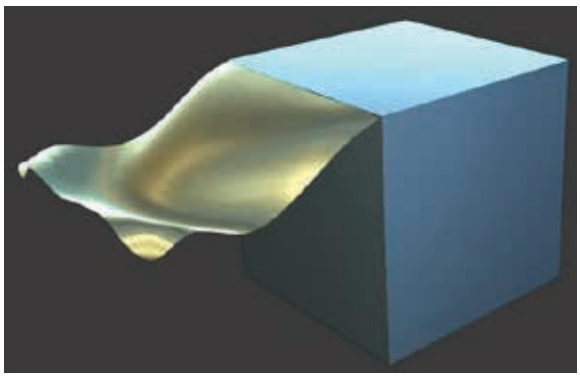


Increase the **Strength** until it is strong enough to make the cape stay up.

By playing the simulation (**press Spacebar**) you can do this interactively.

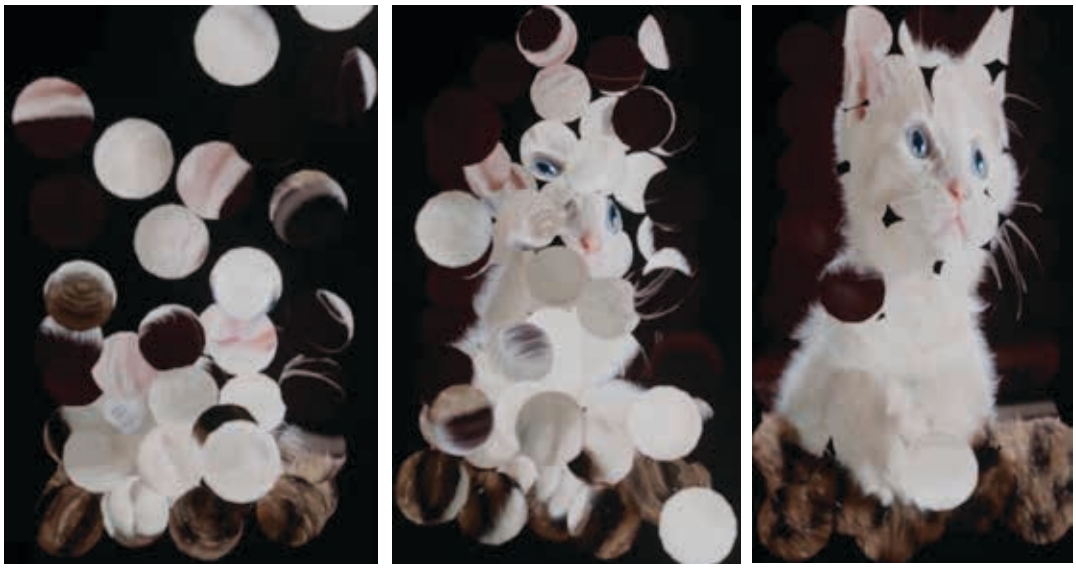


Right-click and turn on **Shade Smooth** for the Plane, then add a Matcap to make it look nice.

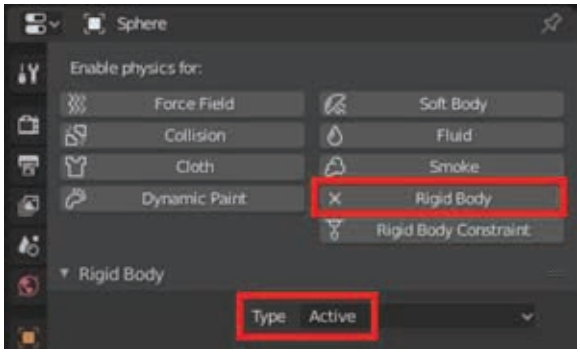


Reveal a Photo with a Physics sim

This is an effect I've seen a few times on Youtube and Instagram where a number of spheres fall into a box, revealing an image. The tutorials about this effect are always super long. It's really not that complex.

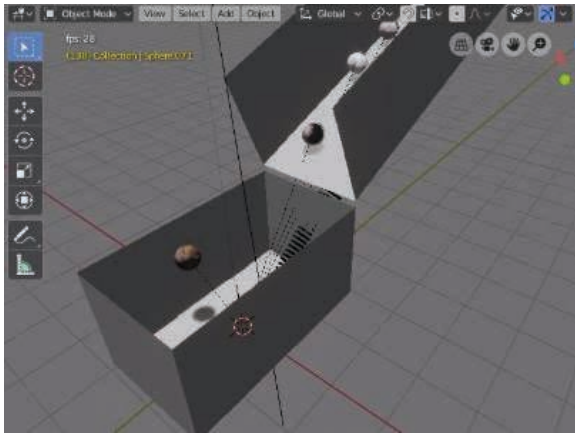


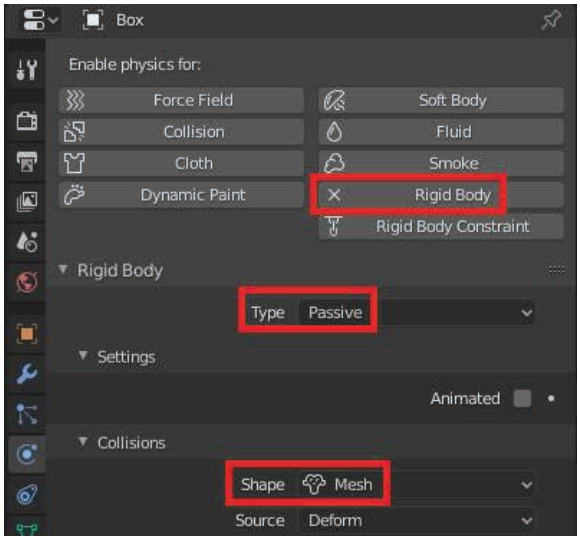
First, have some spheres going down a ramp into a box.



The spheres are **Active Rigid Bodies**. They are Active because they move.

The box and ramp are **Passive Rigid Bodies**, set to "Mesh".



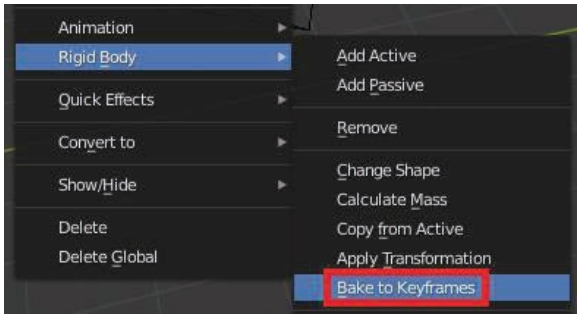


The box and ramp are **Passive** because they don't move.

Mesh shape means that the simulation has to take into account the actual shape of the object. Otherwise it would just look at the bounding box of that object, so in the case of the open box the balls would not go inside but bounce off the top instead.

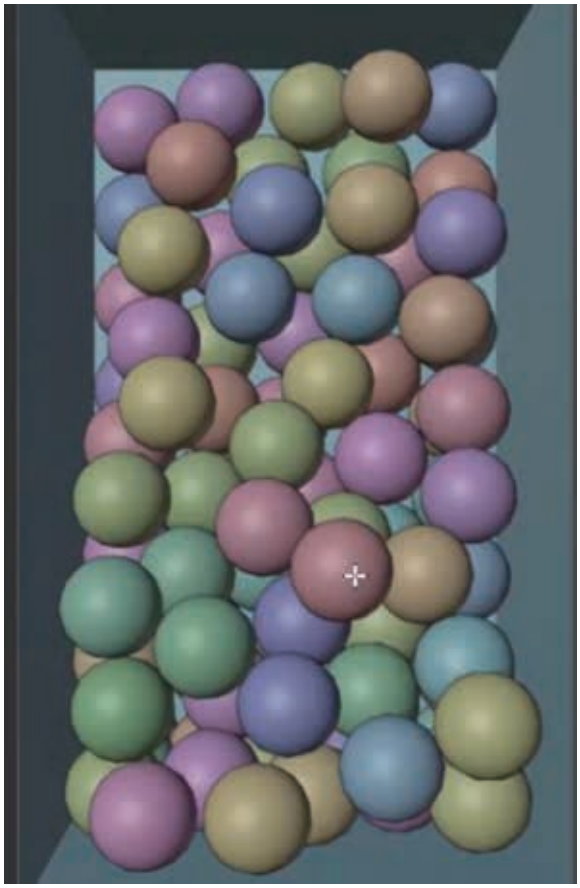
The balls need to have their simulation converted to keyframes.

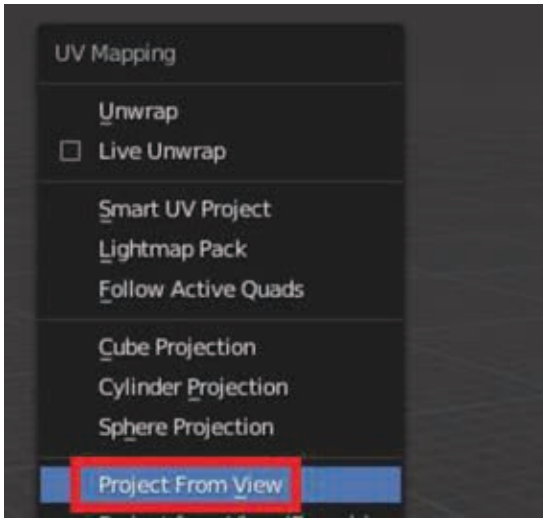
To do this select the spheres, go to Object > Rigid Body > **Bake to Keyframes**.



To put the image of the cat on all the different balls, give them all the same material which has a cat picture as an Image Texture.

Let the animation play until the end, then at that point select all the balls and in edit mode **press U**, then "**project from view**".



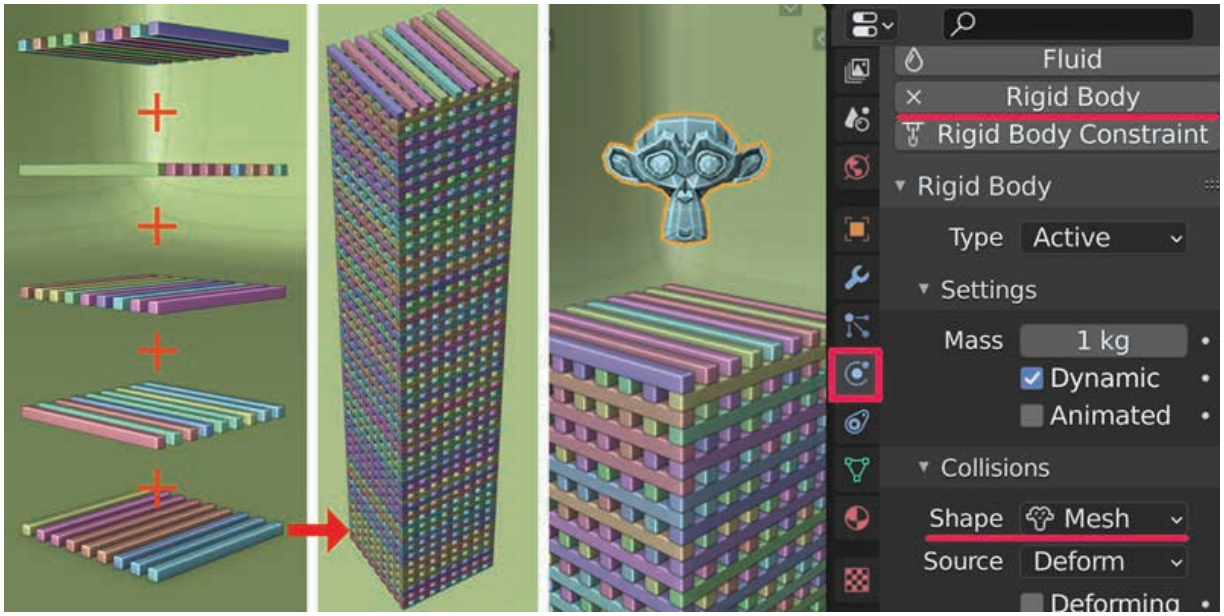


This unwraps them all from the view of the camera. Now they have the cat image on them.

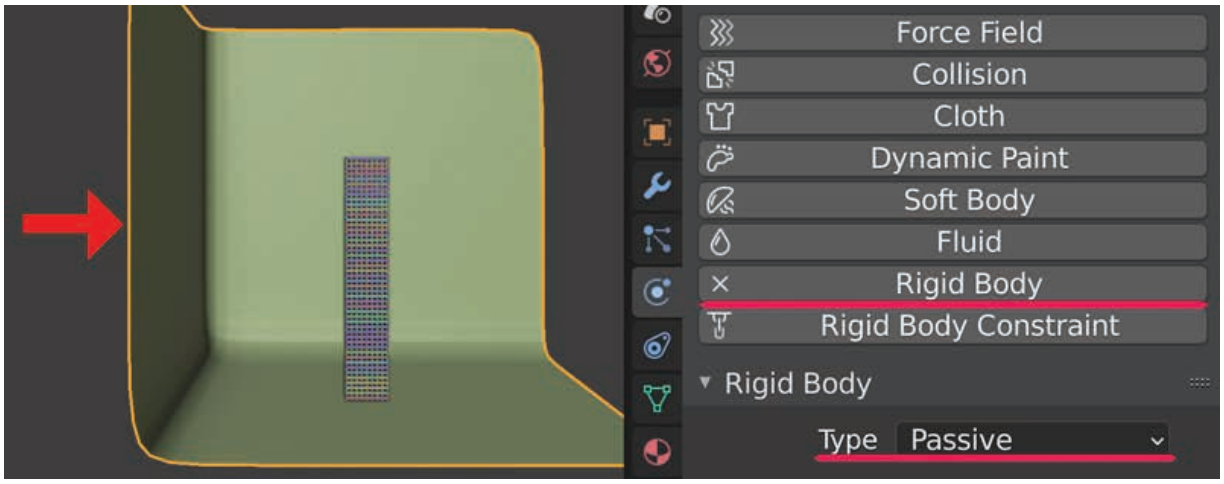
Make sure you use enough balls, otherwise you'll still have some gaps (as pictured in the result on the right).

Fun with Planks (Rigid Bodies)

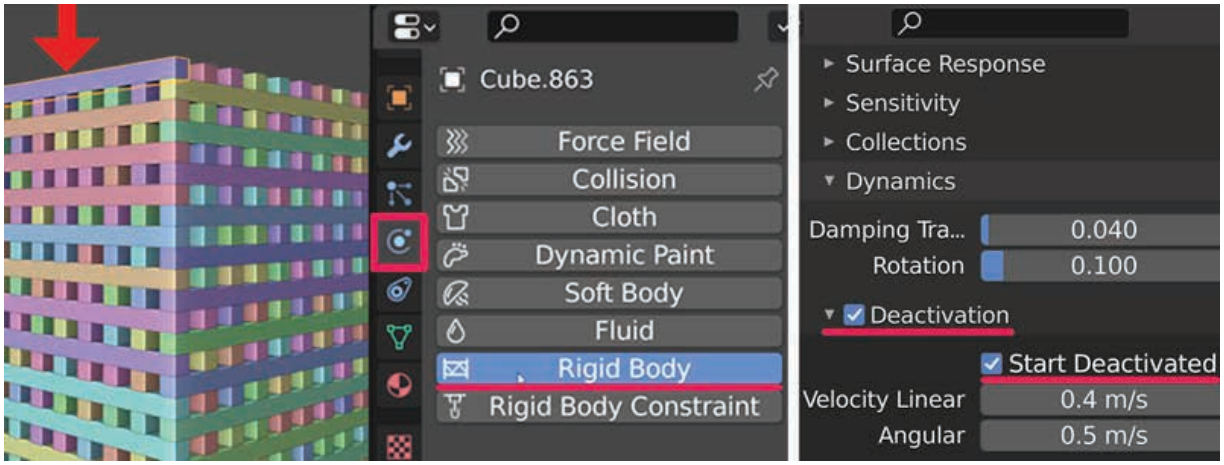
Create a stack of planks and a Rigid Body to interact with. Set that one to **Shape: Mesh**.



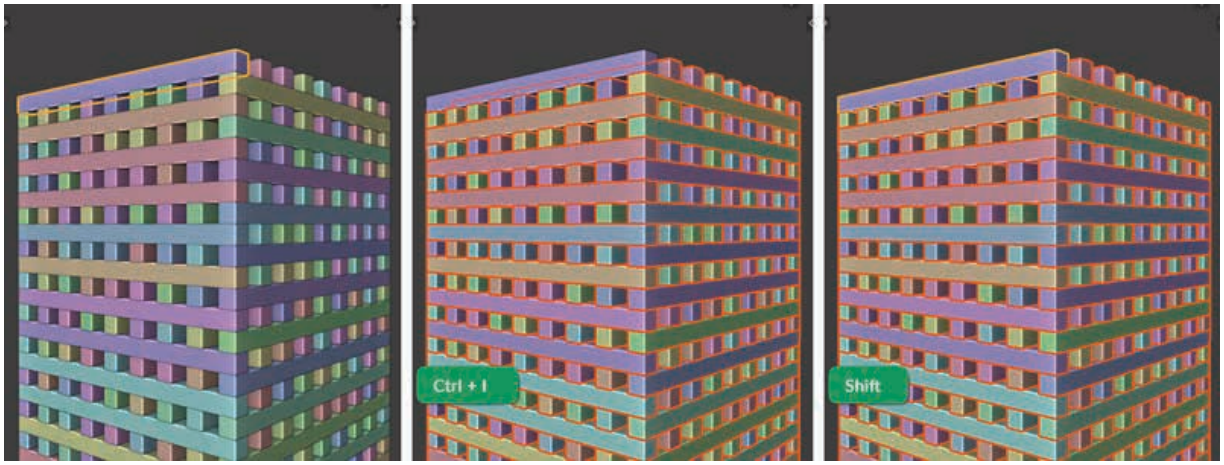
Set the Ground plane to Rigid Body: **Passive**.



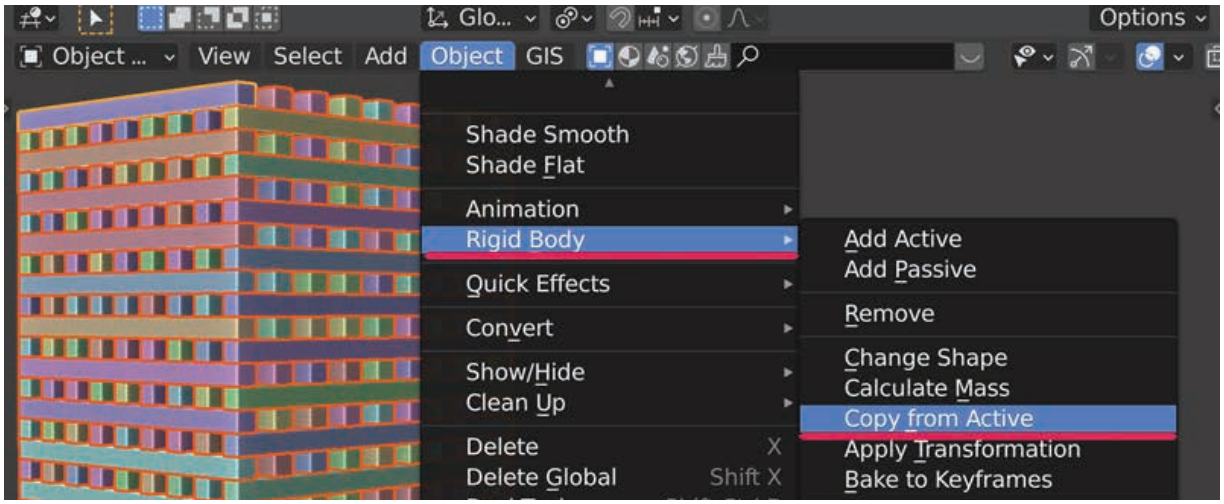
Hide the object and ground plane, then set one of the planks to Rigid Body. Turn on **Deactivation** and **Start Deactivated**.



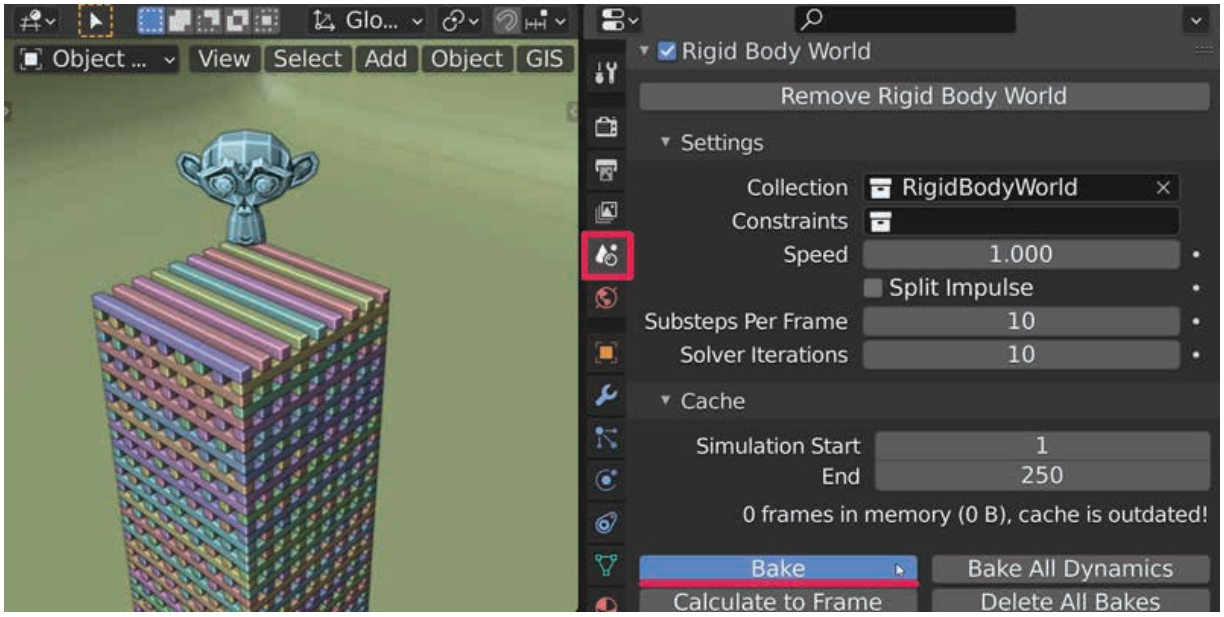
Invert the selection with **Ctrl+i** and select that first plank again while **holding Shift**, so it is the **Active Object** and everything is selected.



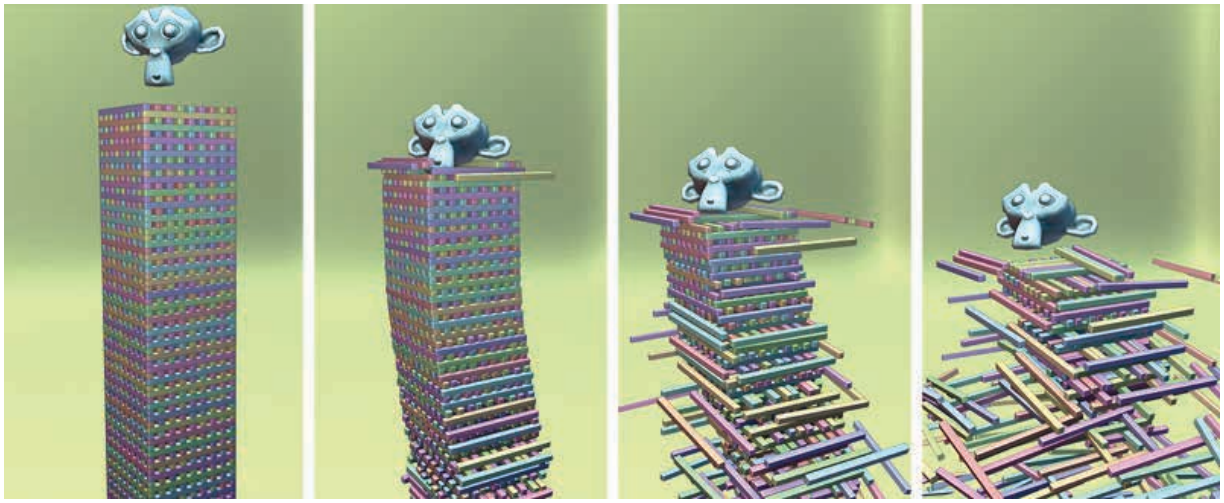
Go to Object > Rigid Body > **Copy from Active**.



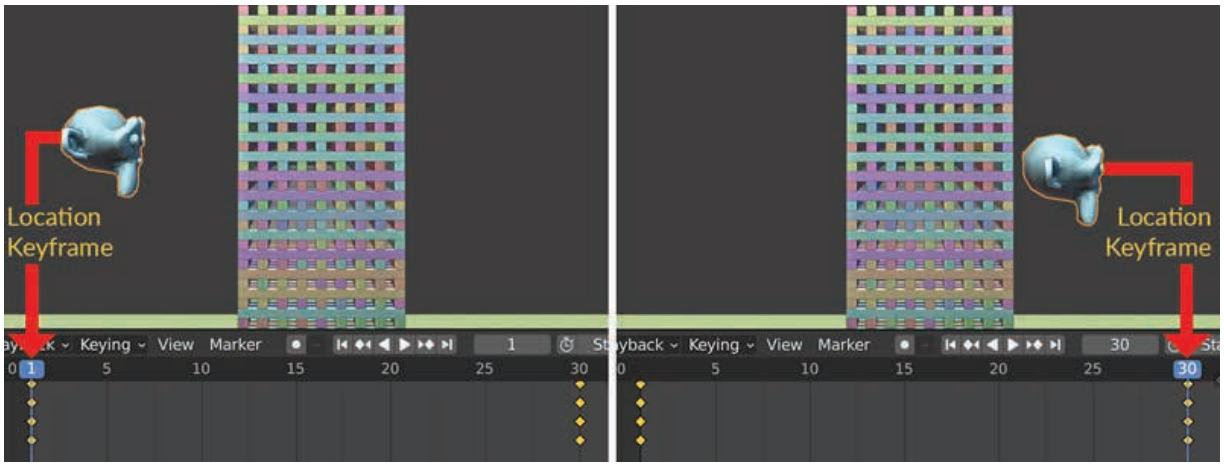
Bake the Simulation.



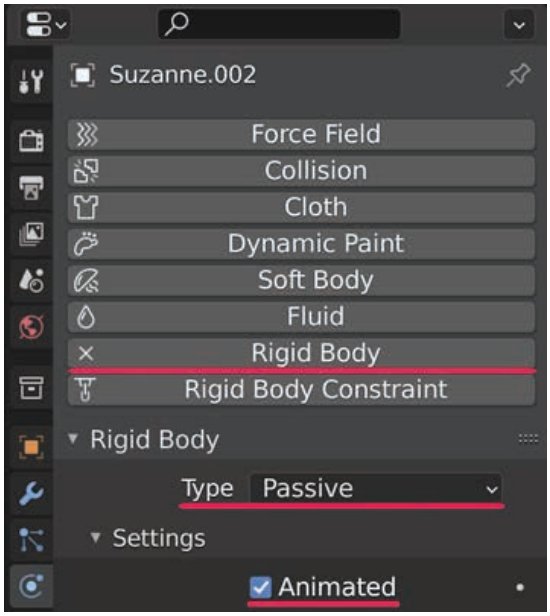
Then press the **Spacebar** to see the result from just using gravity.



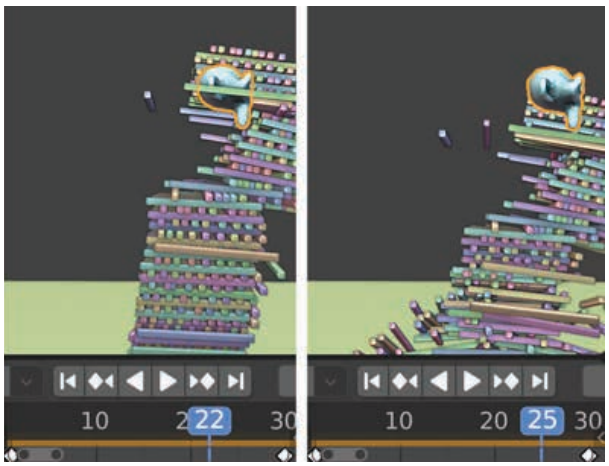
Let's make it even more interesting. Animate the object with some location keyframes.



Change the Object to a **Passive** Rigid Body and check "**Animated**".

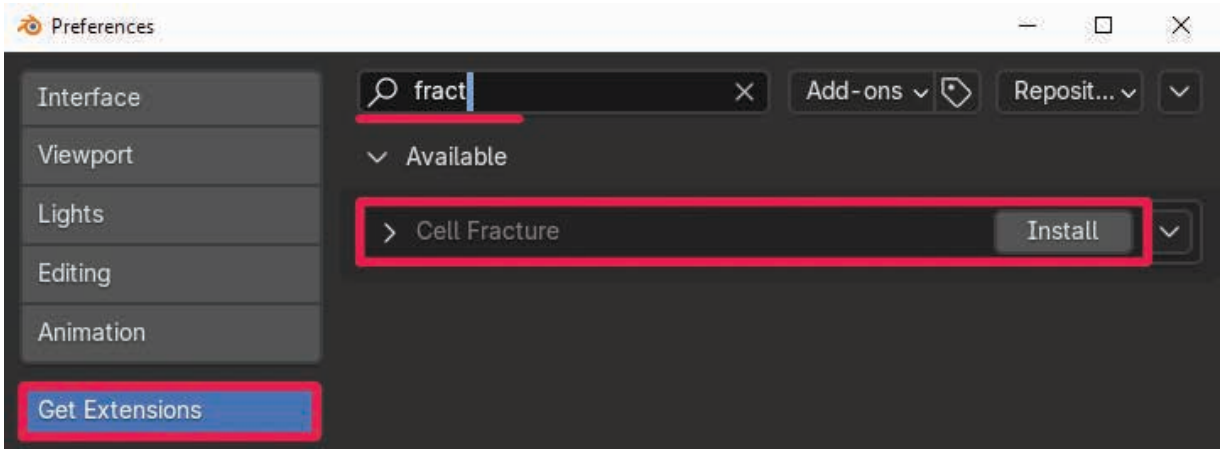


Then press the **Spacebar** to see the result.

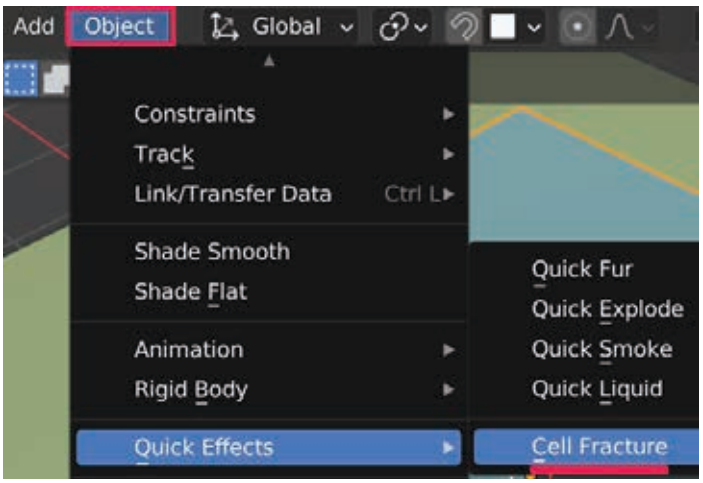


Cell Fracture Explosion

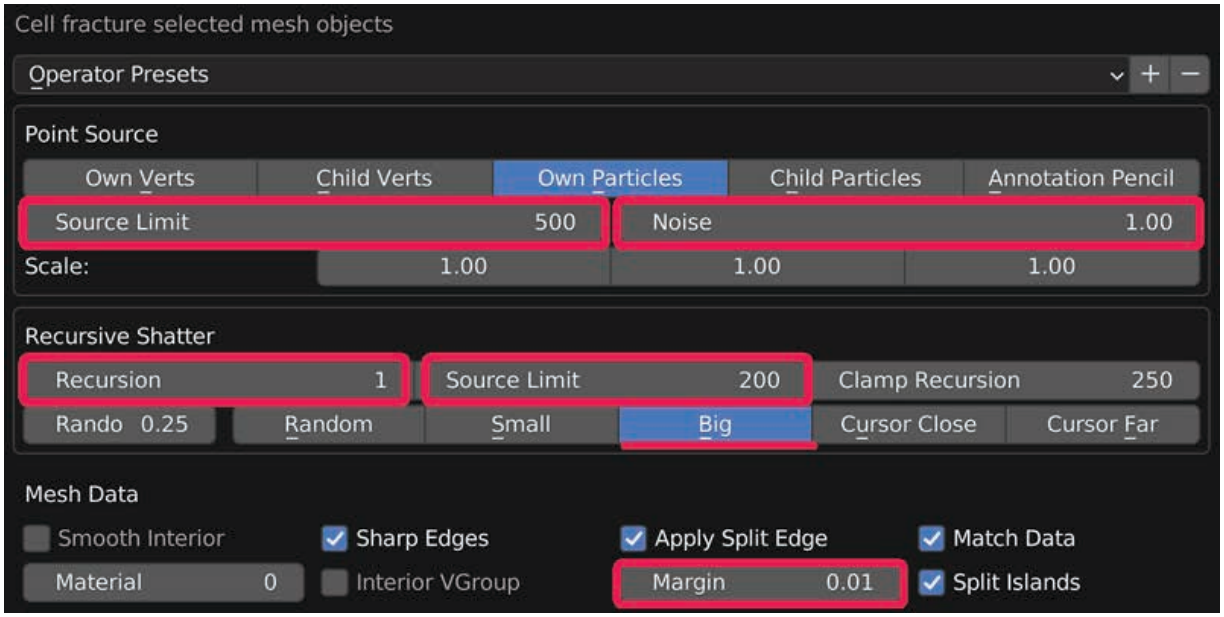
Cell Fracture is an add-on that ships with Blender by default. It's good for splitting objects into random chunks, which is a good start for creating explosions and other kinds of destruction. Turn on "**Cell Fracture**" in the **Get Extensions** tab in Preferences.



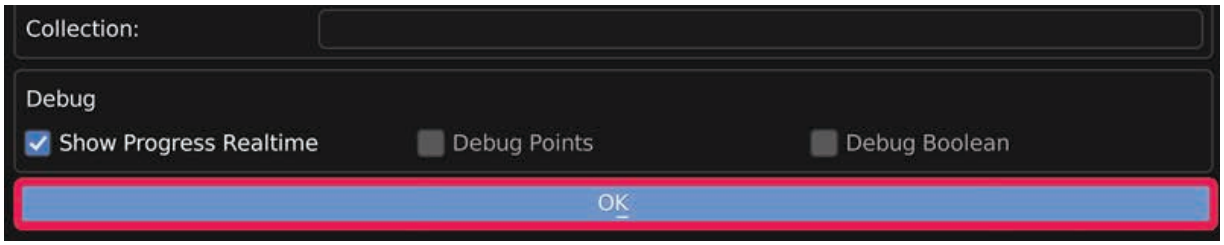
Select you object and go to Object> Quick Effects > **Cell Fracture**.



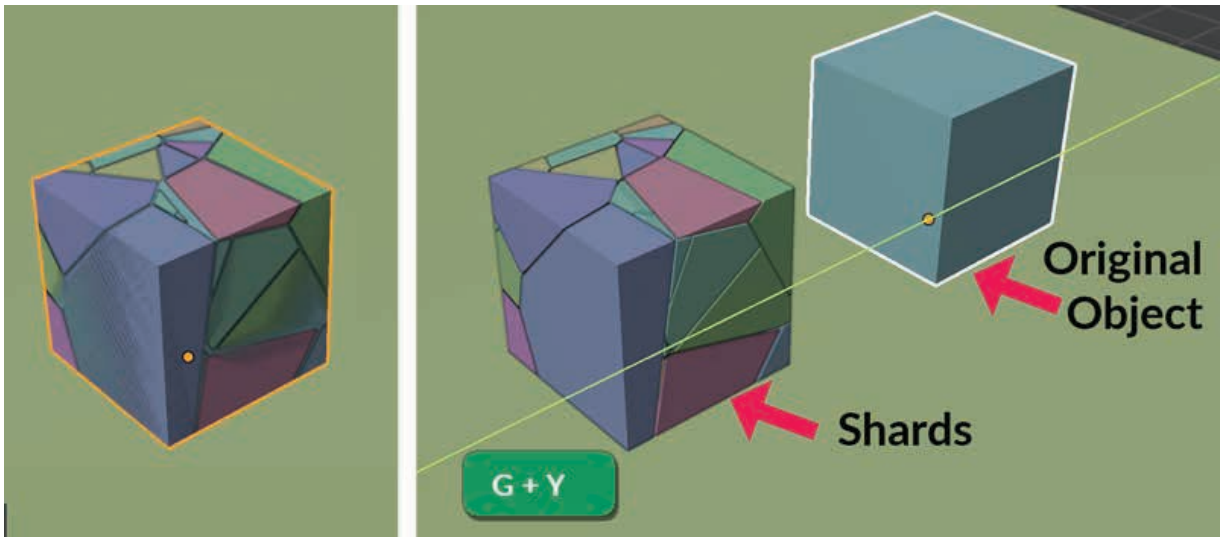
These are the settings I used.



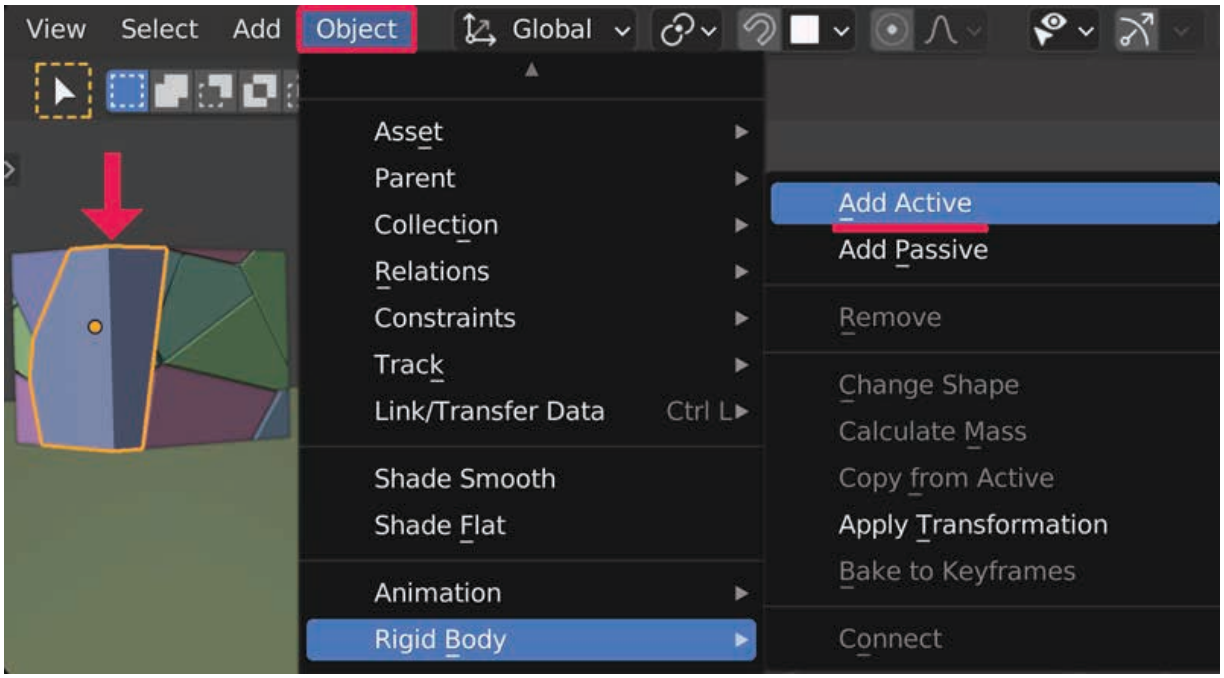
Click **OK** to apply the Fracturing.



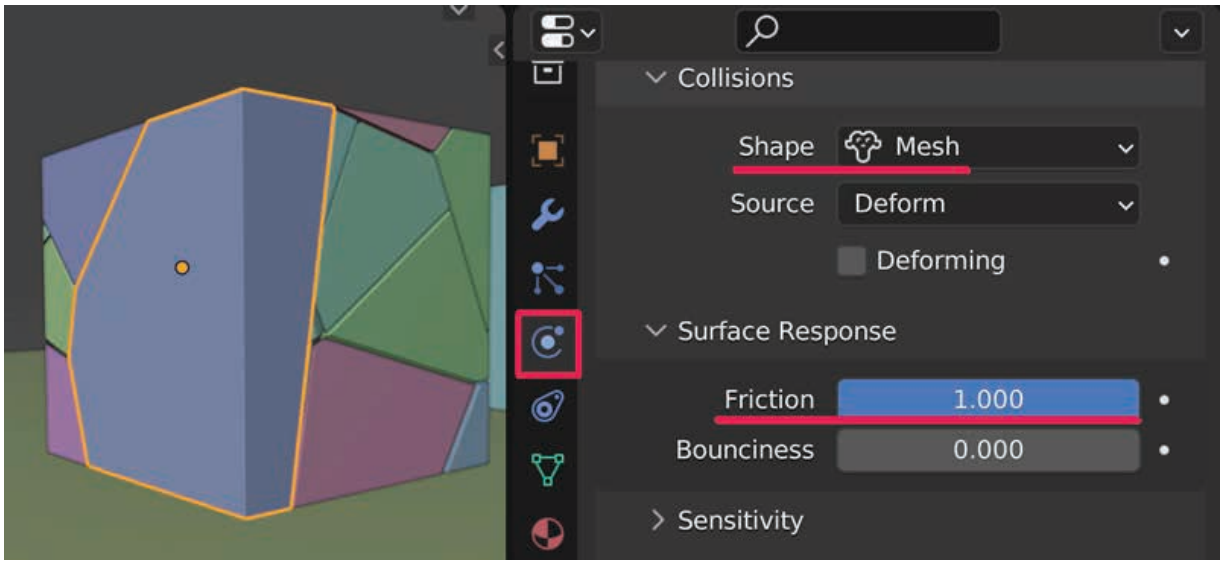
The original object remains. We can move it to the side for now.



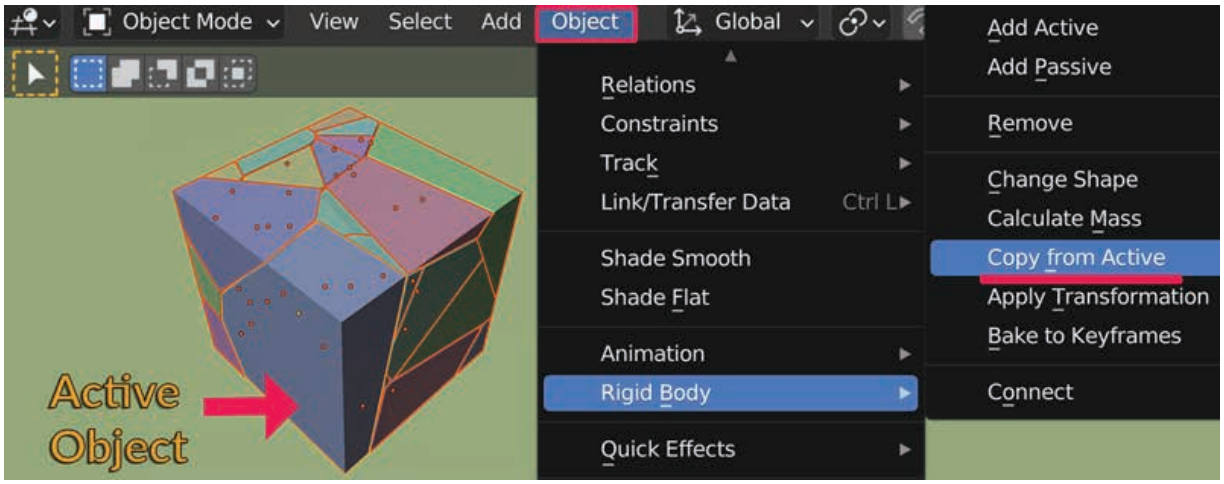
Select one shard, and go to Object> Rigid Body> **Add Active**.



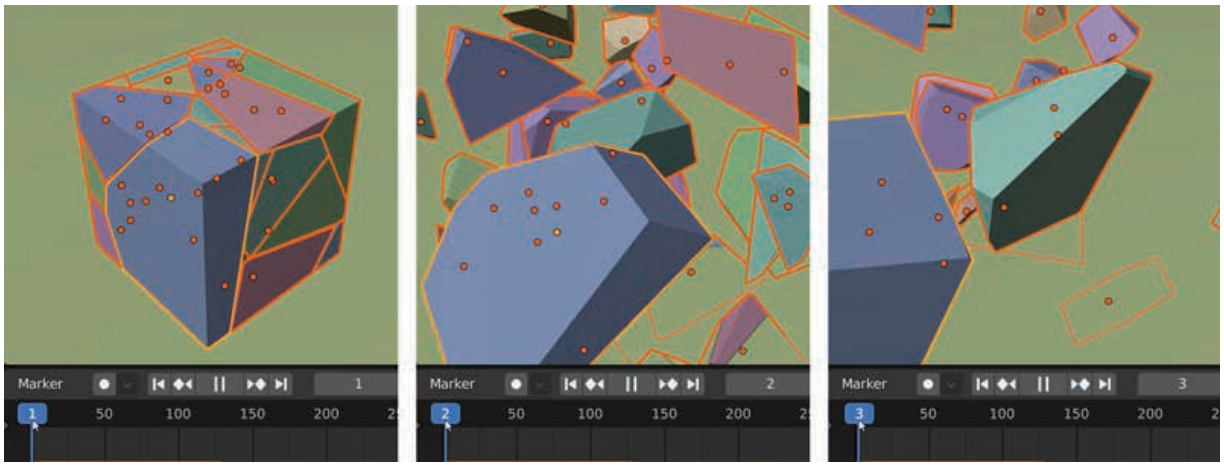
Set Collision Shape to Mesh and Surface Response> **Friction to 1**.



Select the other shards and then the Rigid Body. Go to Rigid Body> **Copy from Active**.

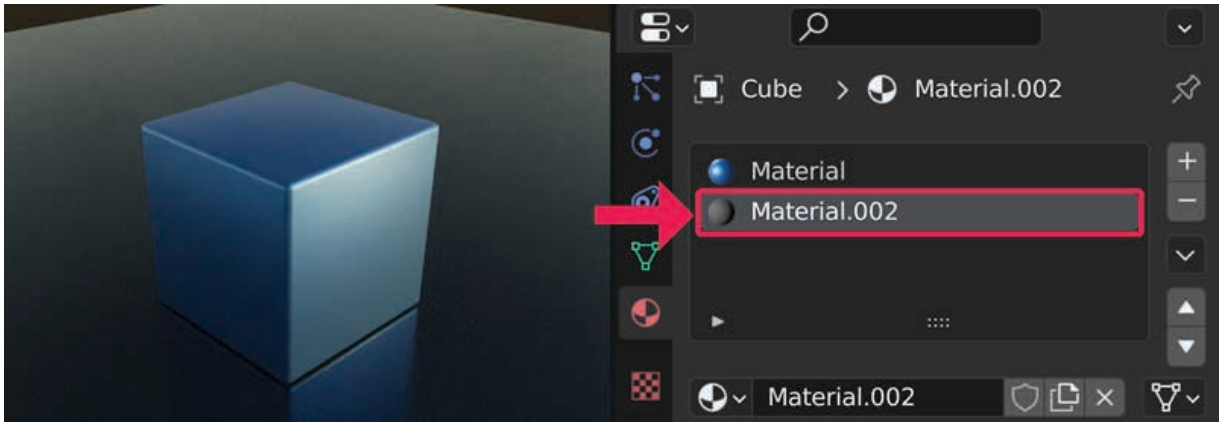


Now if you play the simulation by **pressing the Spacebar**, the cube will explode.

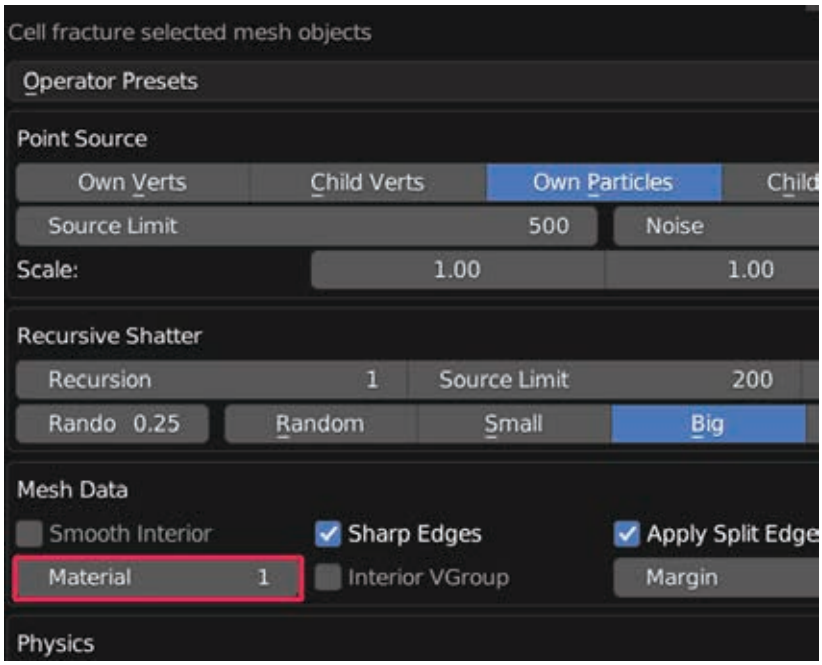


Cell Fracture Inner Material

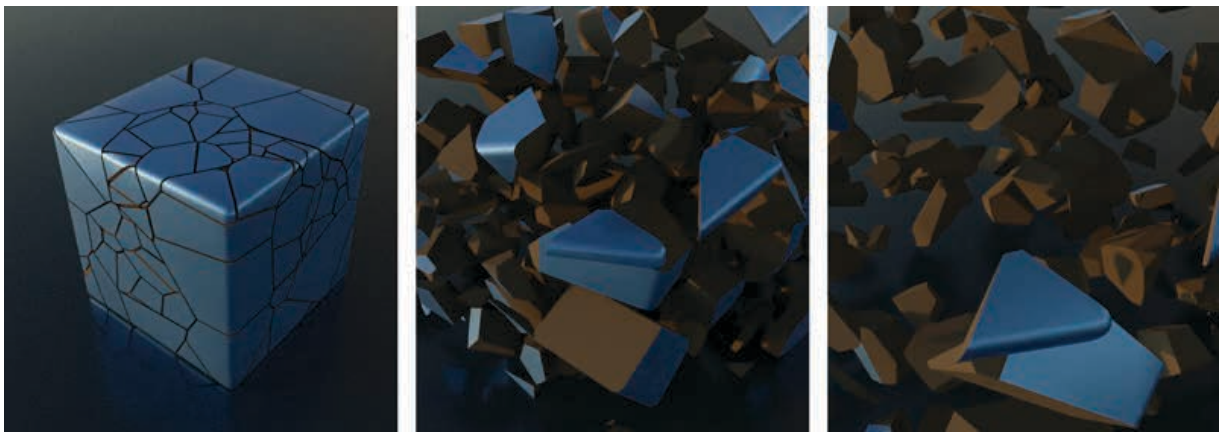
To have a different inner material from the outer material of your cell fractured object is easy, if you follow these simple steps. To add a material to the inside of your fractured object before fracturing it **add a second material**. You can do this by **clicking on the + icon**.



Then in the Cell Fracture panel, set the material to 1.



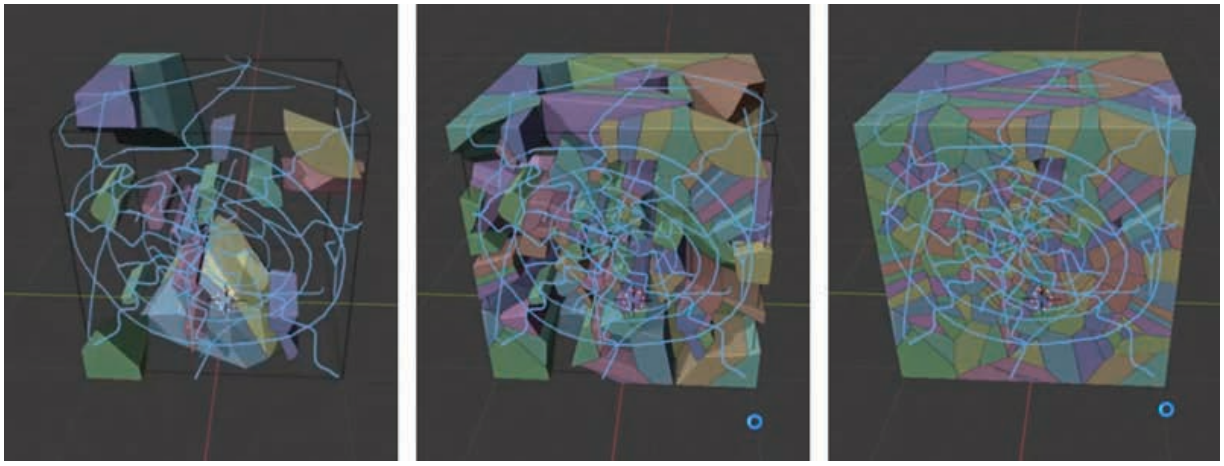
This way Blender knows to use the second material for the inside. Now after running the cell fracture add-on the inside of the model has a different material from the outside.



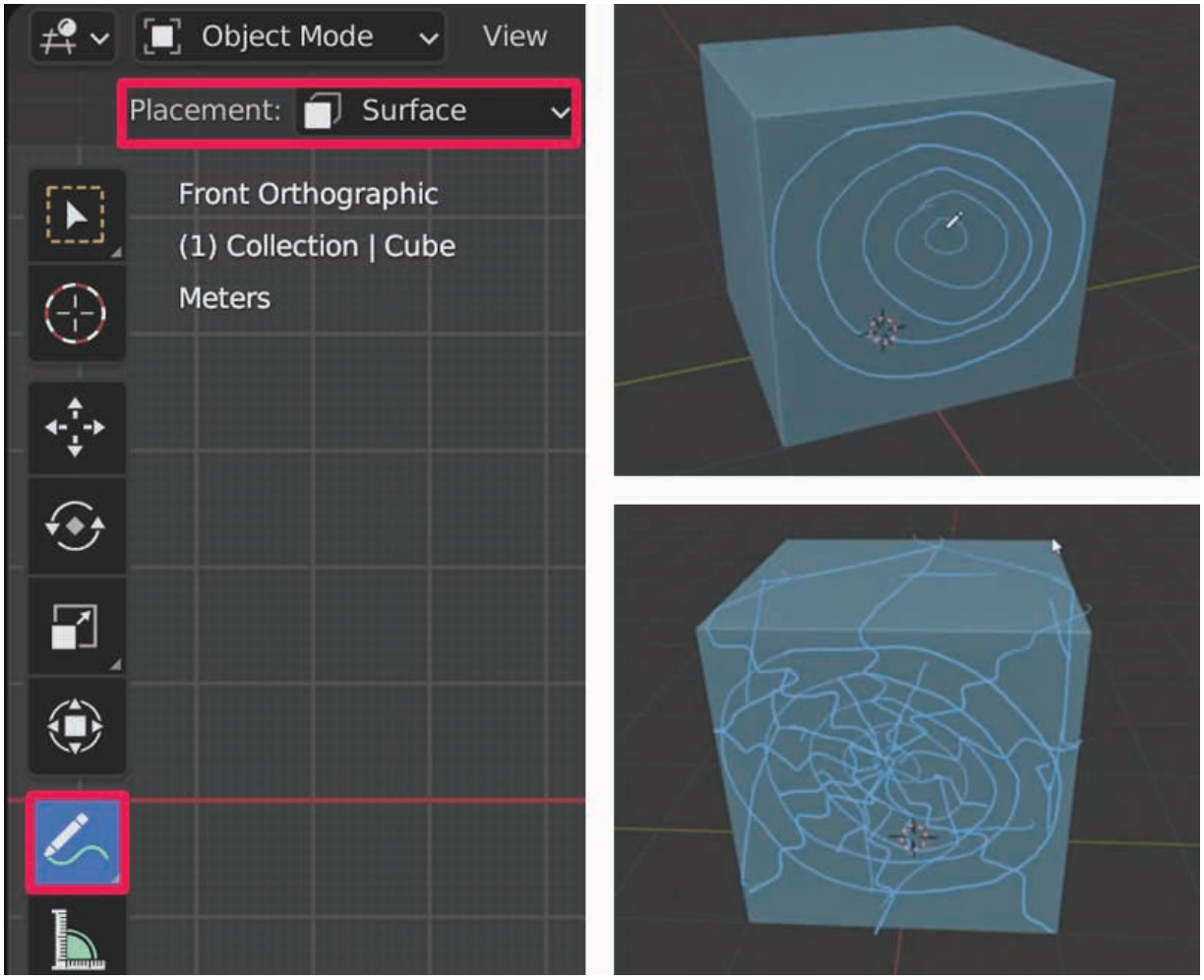
Click here to return to the index.

Annotate Cell Fracture

Previously we looked at using the Cell Fracture add-on to randomly destroy objects. If you don't want the fracture to be random, you can use the annotate tool to draw the fracturing pattern yourself.

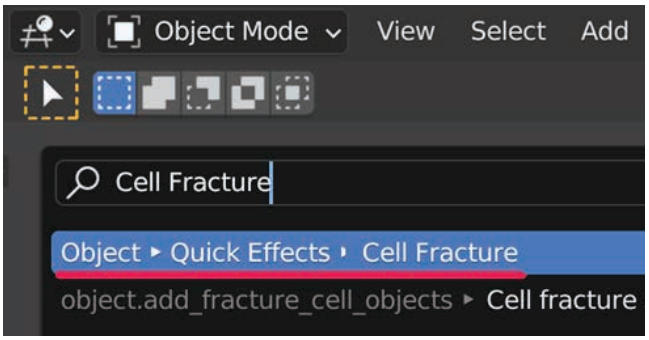


In Object Mode, turn on the **Annotate tool** by clicking on the pen icon in the Toolbar. Then for **Placement** choose **Surface** so the lines are drawn on the surface of the object. Now draw the destruction pattern that you want.

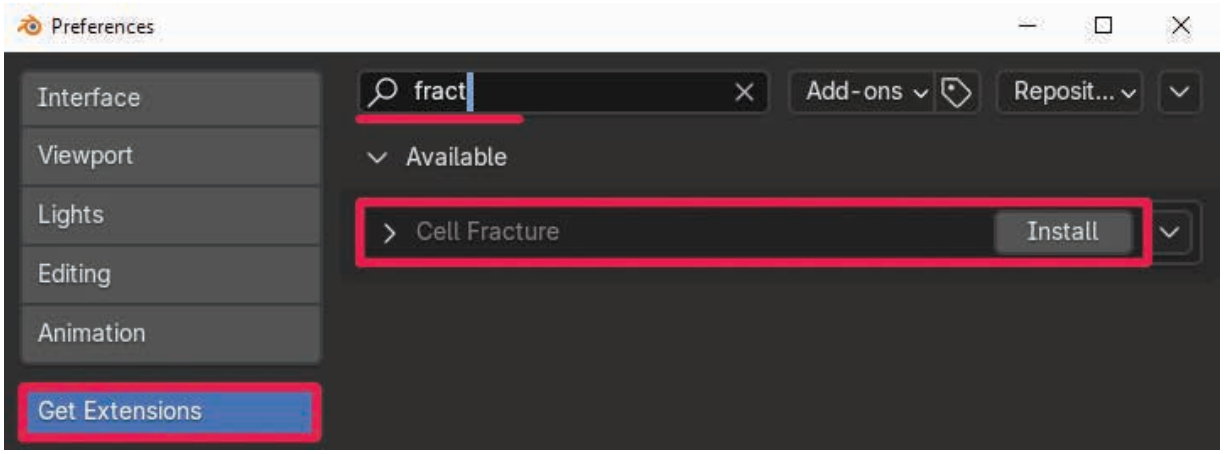


Click here to return to the index.

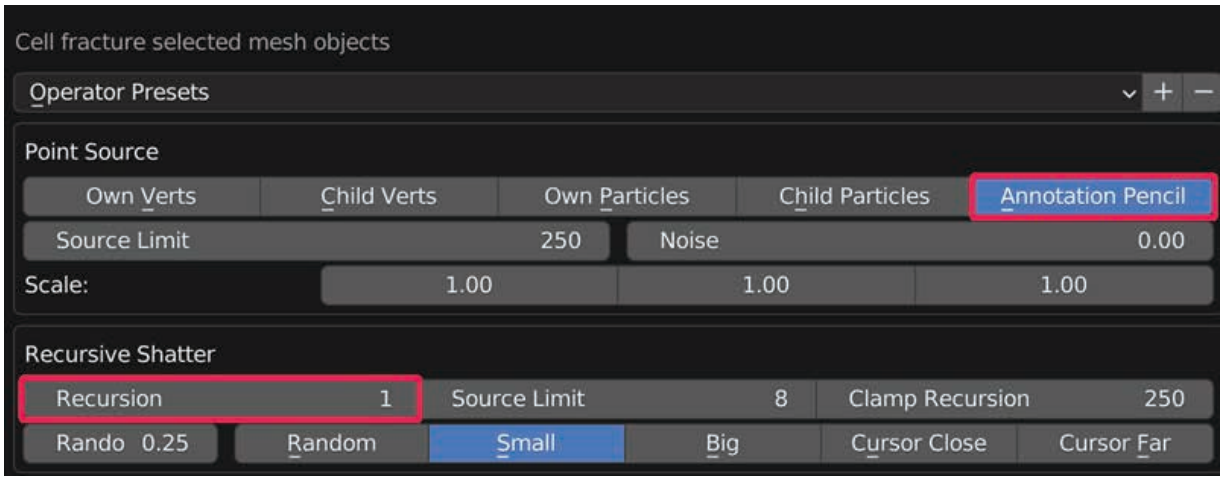
Make sure your object is selected, then **press F3** and search for the **Cell Fracture add-on**.



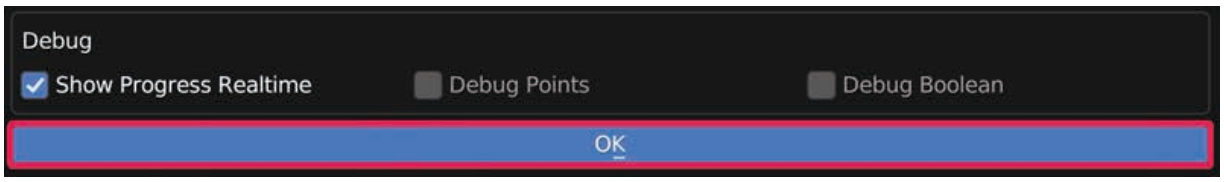
Remember that it needs to be activated in **Preferences** to show up.



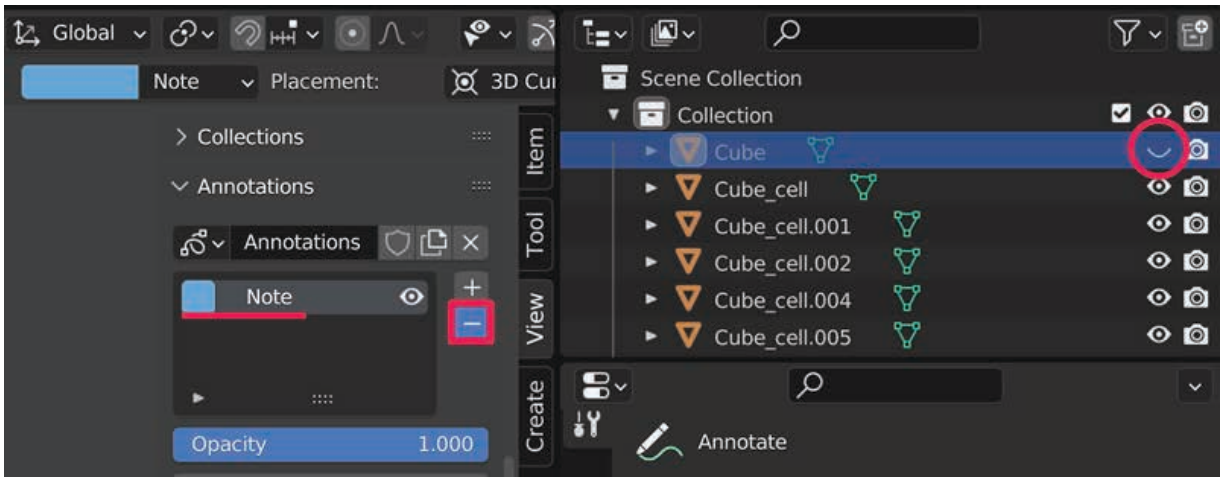
As the Point Source, choose **Annotation Pencil**. Turn on **Recursion** by setting it to 1, this breaks individual shards into even smaller pieces.



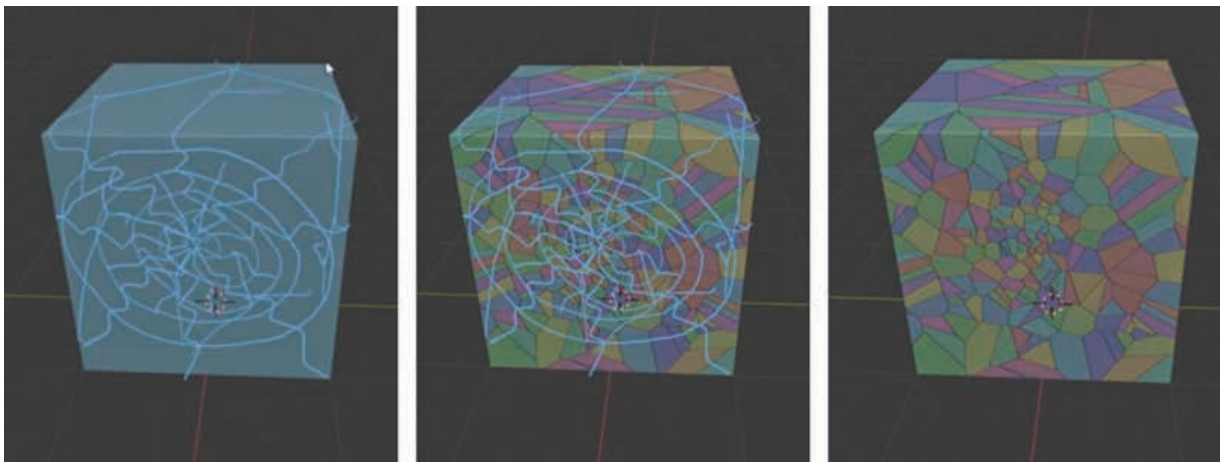
Finally **press OK**.



Now turn off the original object visibility and in the Option panel delete the annotation lines.

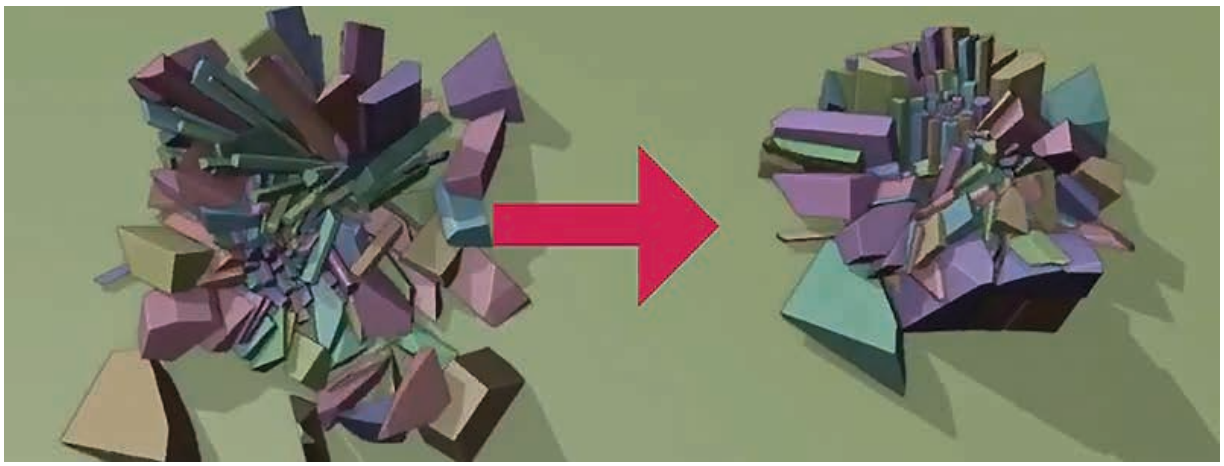


As you can see the cube has been fractured following your annotation.

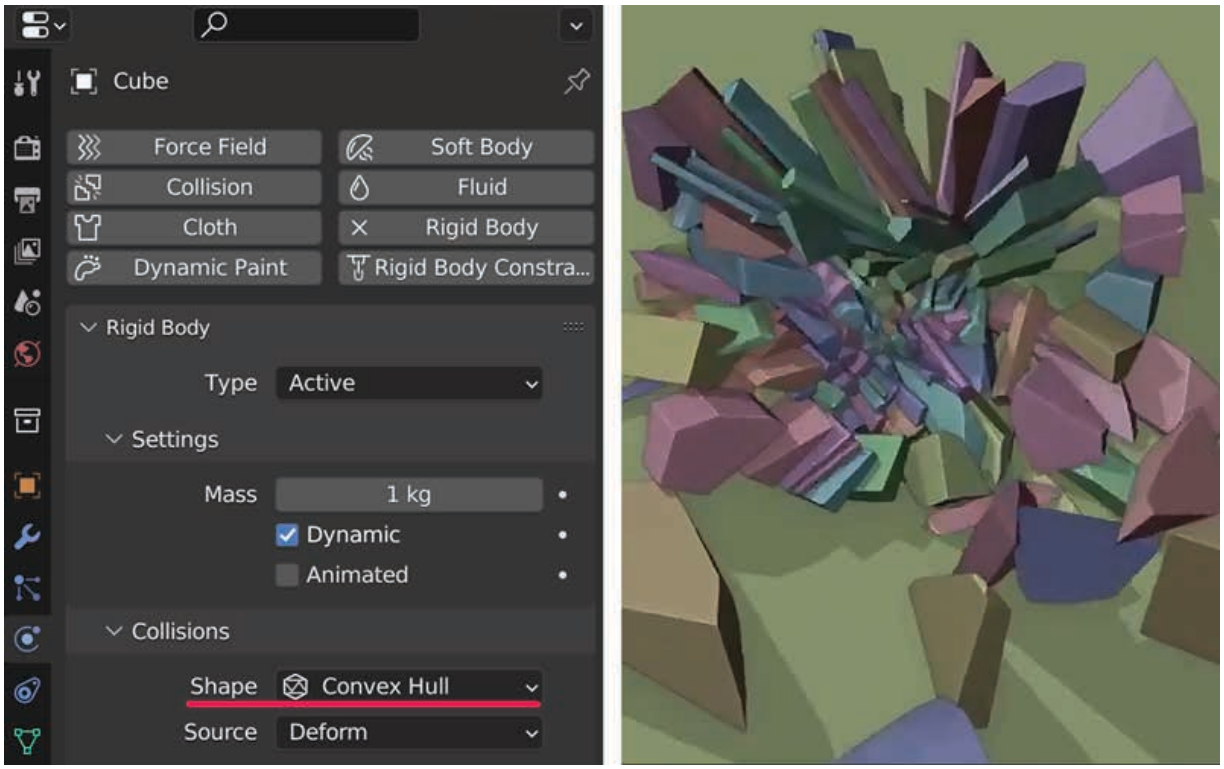


More realistic destruction and collision detection

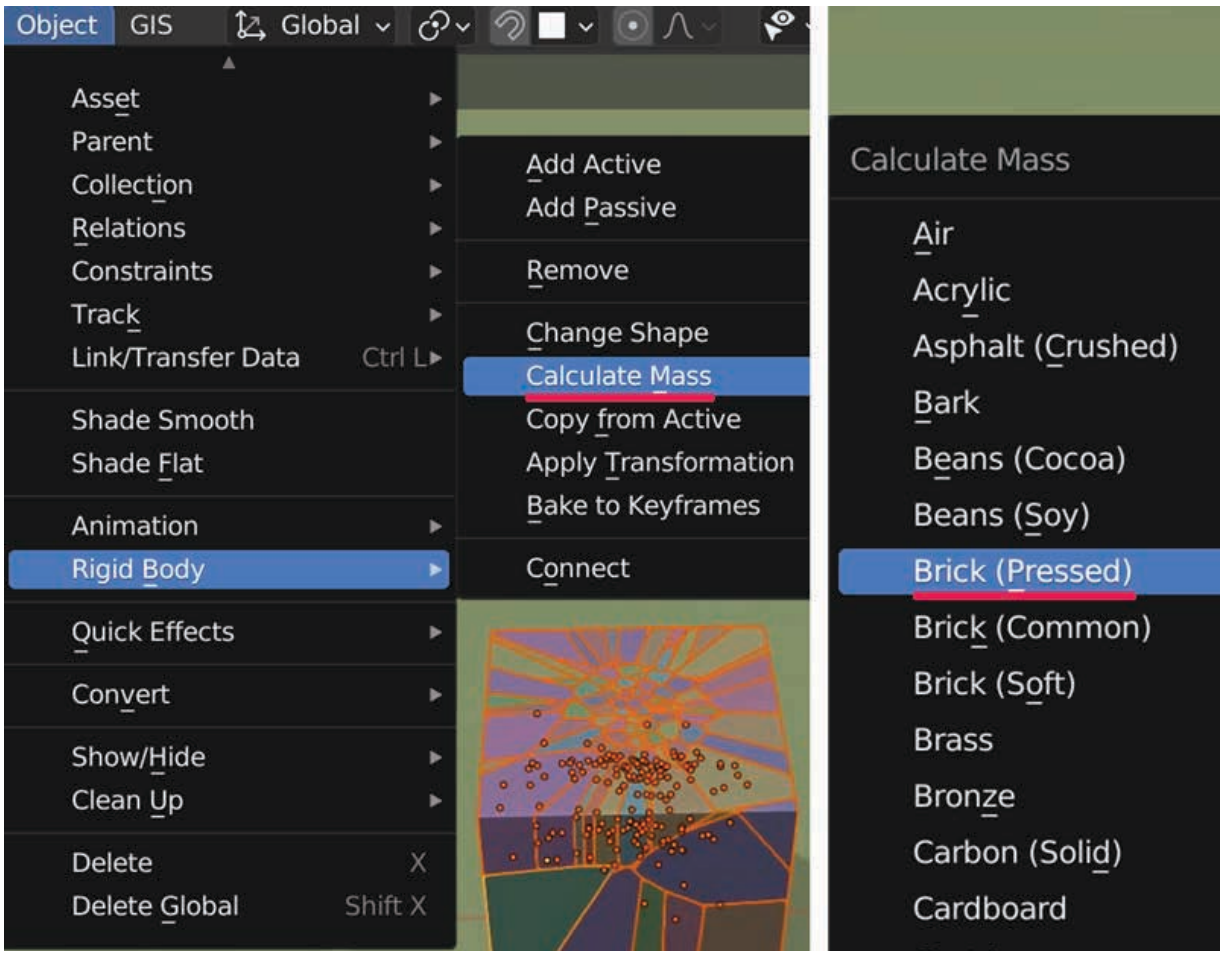
In this topic we look at more realistic weight for your destroyed objects, as well as having them be destroyed by collision detection.



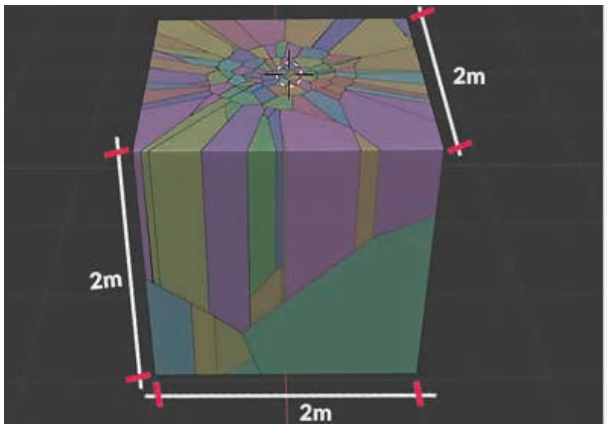
If you don't want your object to explode keep the Collision shape as "Convex Hull" instead of "Mesh" that way they will just fall apart once the simulation starts. Now all the pieces have the same weight, which doesn't really make sense.

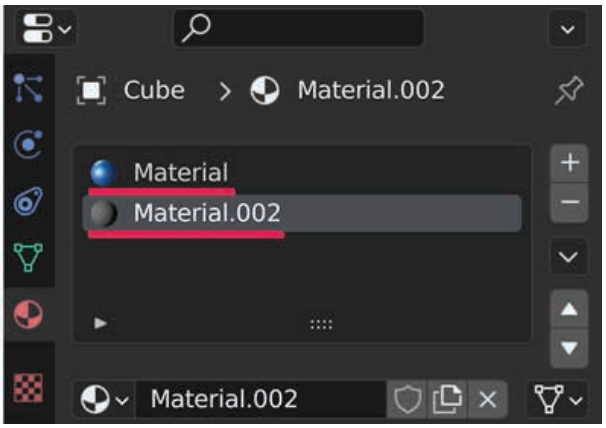


So select them all, and go to Object> Rigid Body> **Calculate Mass**.

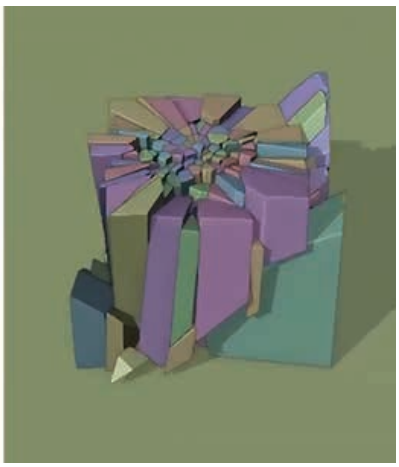


Now you can choose all kinds of materials so that Blender can assign the proper amount of weight to each piece. Keep the size of your object in mind as well for example the default cube here is 2 metres by 2 metres.

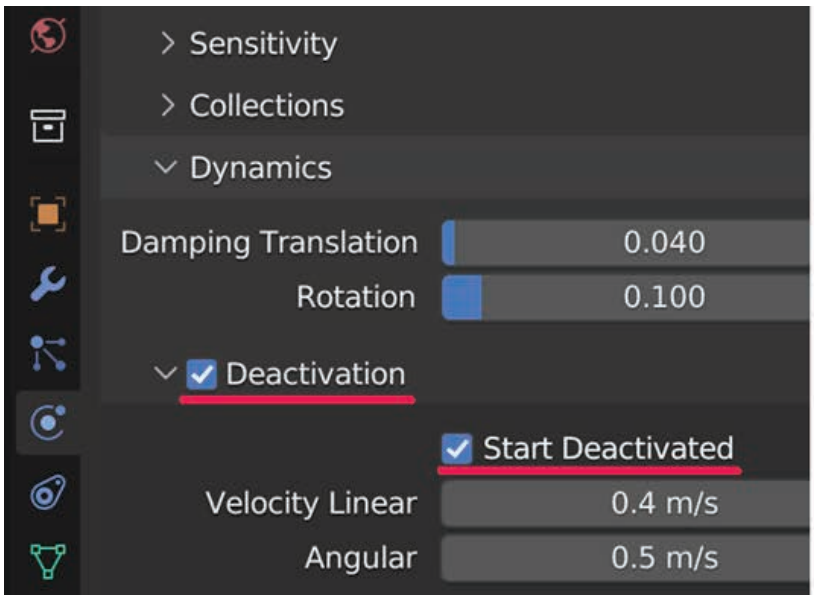


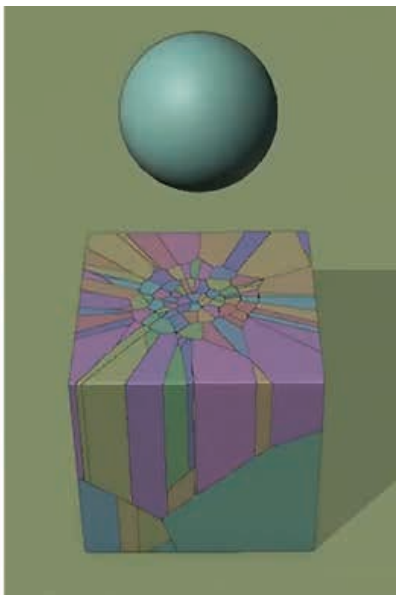


Now all the objects have their individual weight calculated and the simulation is more realistic.

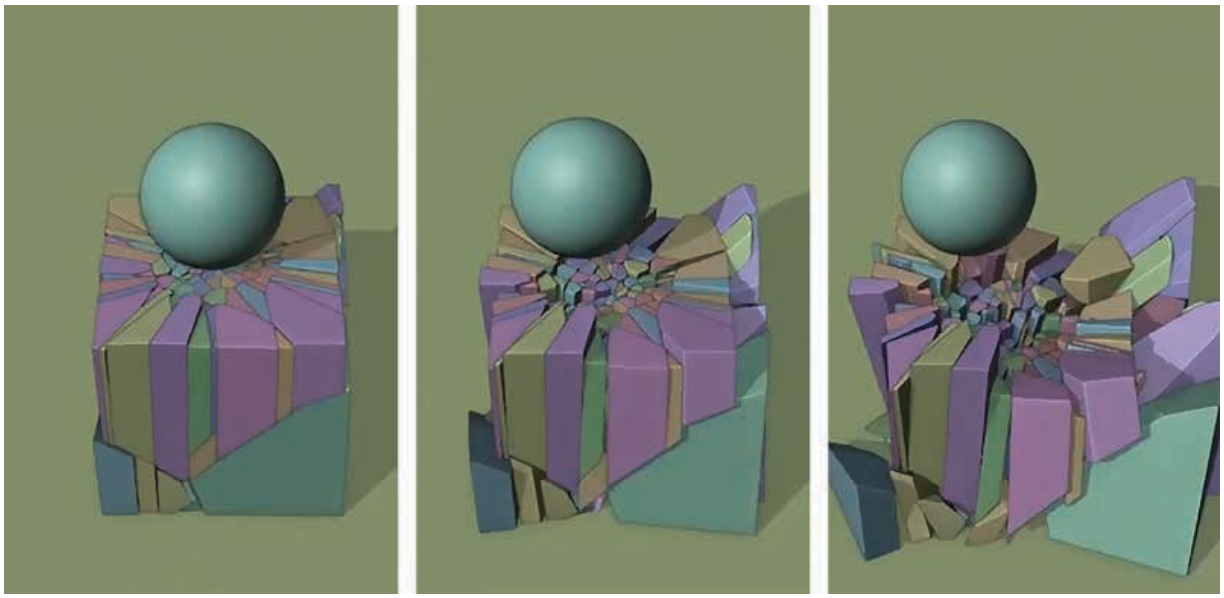


If you want the object to break once another object hits it, turn on "**Start Deactivated**" for each piece.



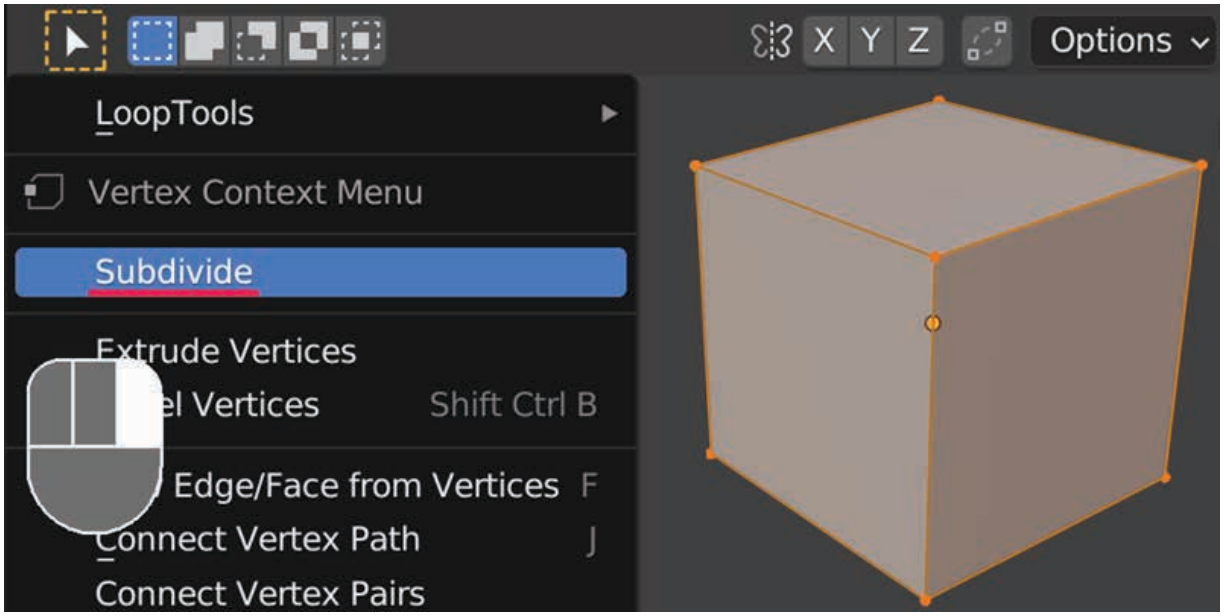


Now the fractured object doesn't start simulating until it's hit.

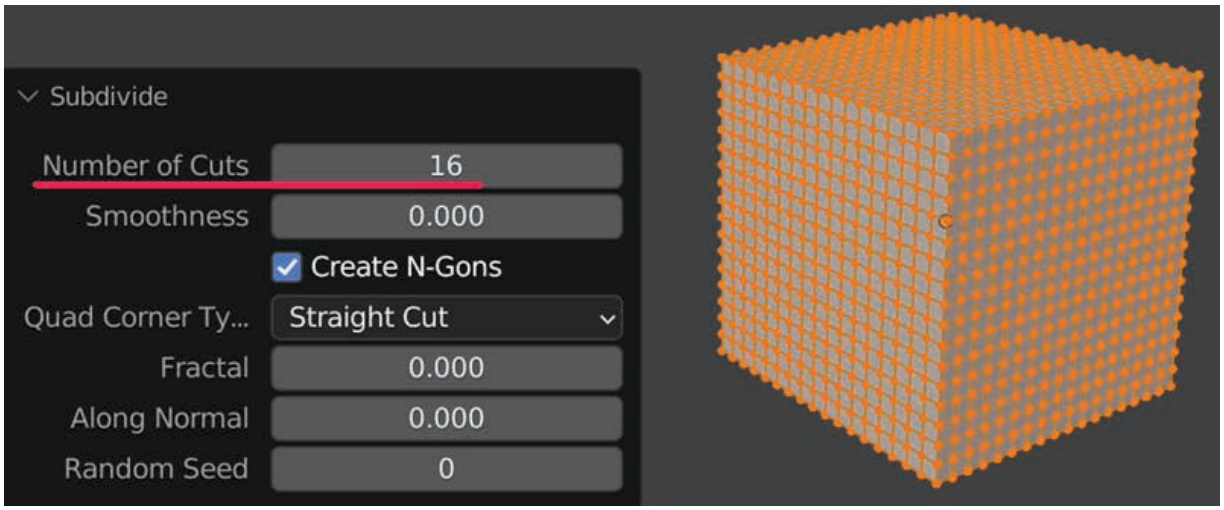


Exploding Default Cube

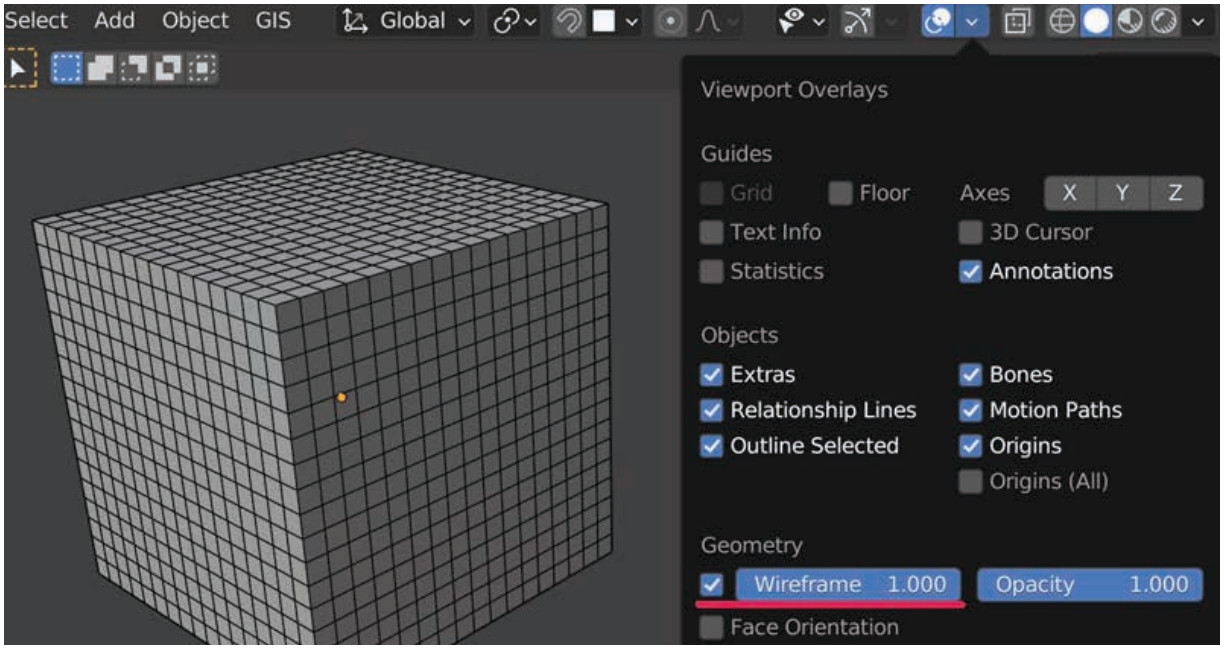
Exploding a default cube (or any other object) is easy with the Explode modifier. But how does it work exactly? First add some subdivisions to your cube. In Edit Mode, right-click and choose **Subdivide**.



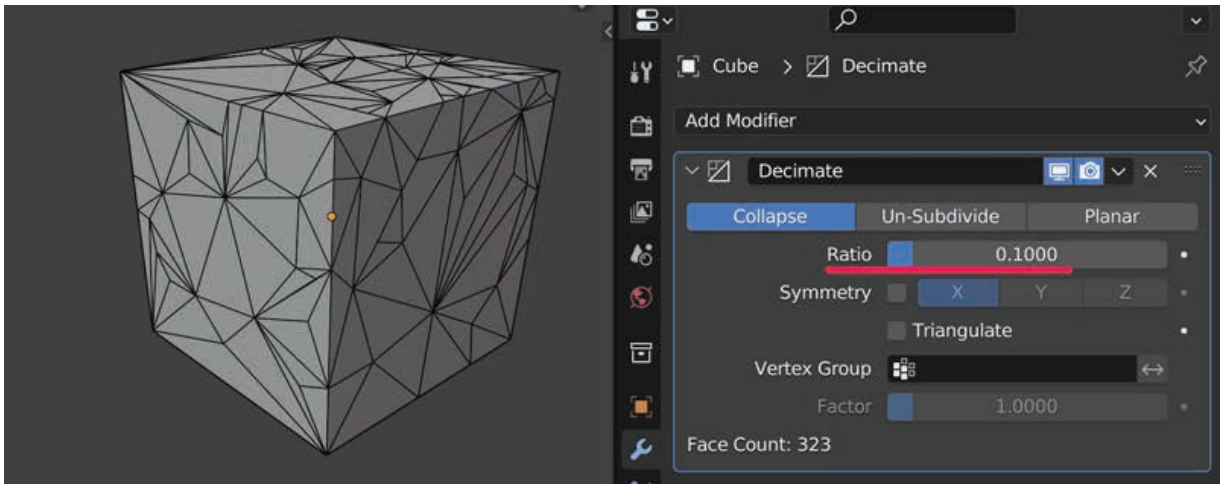
You can choose the **Number of Cuts** in the little option panel that appears. Press F9 if you accidentally close it. Alternatively, press Shift+R a number of times to repeat.



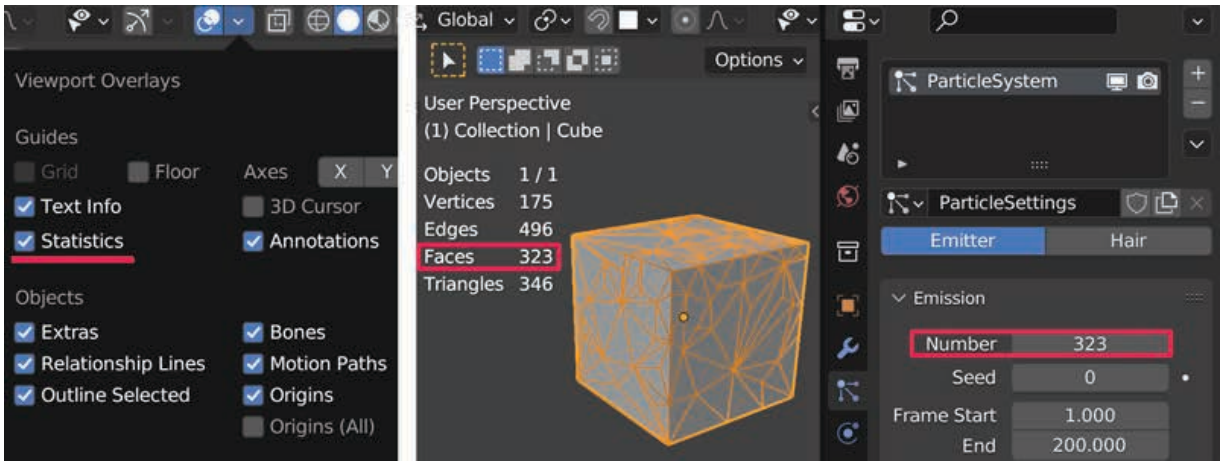
Enable Wireframe display in the Viewport overlays so you can see the fracture lines.



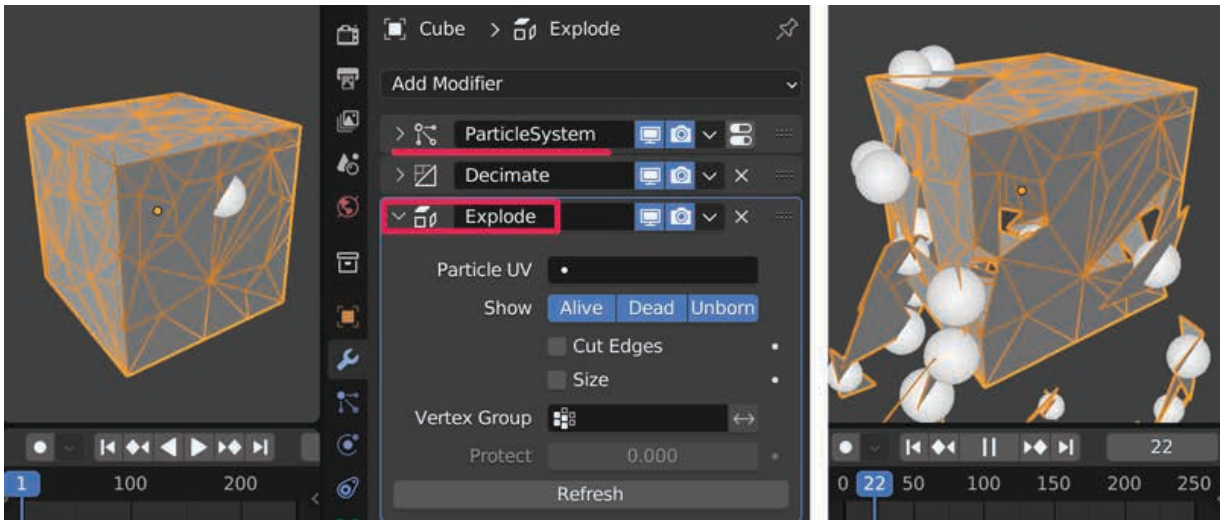
Add a **Decimate modifier** and lower the ratio so you get some more random faces.



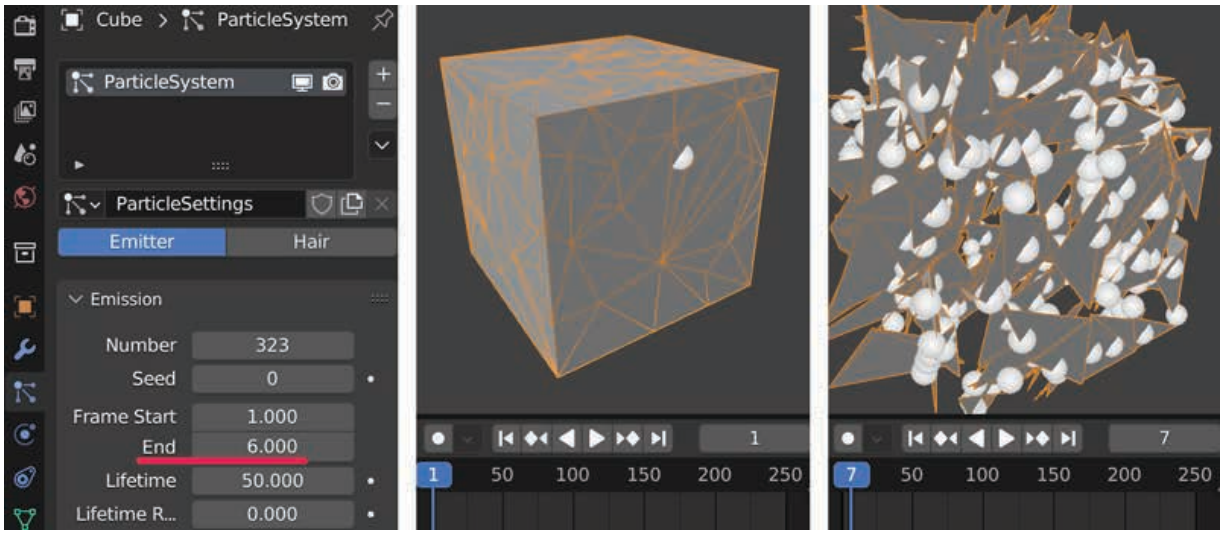
Enable Statistics in Viewport overlays to see the amount of Faces the obejct has. Add a **Particle System**. As the **Number** of particles, add the amount of Faces your object has.



You can see the number in the bottom right corner. Then, add an **Explode modifier**. It's important that the Particle system is at the top of the stack. Now if you **press the Spacebar** you can see that the cube is falling apart.

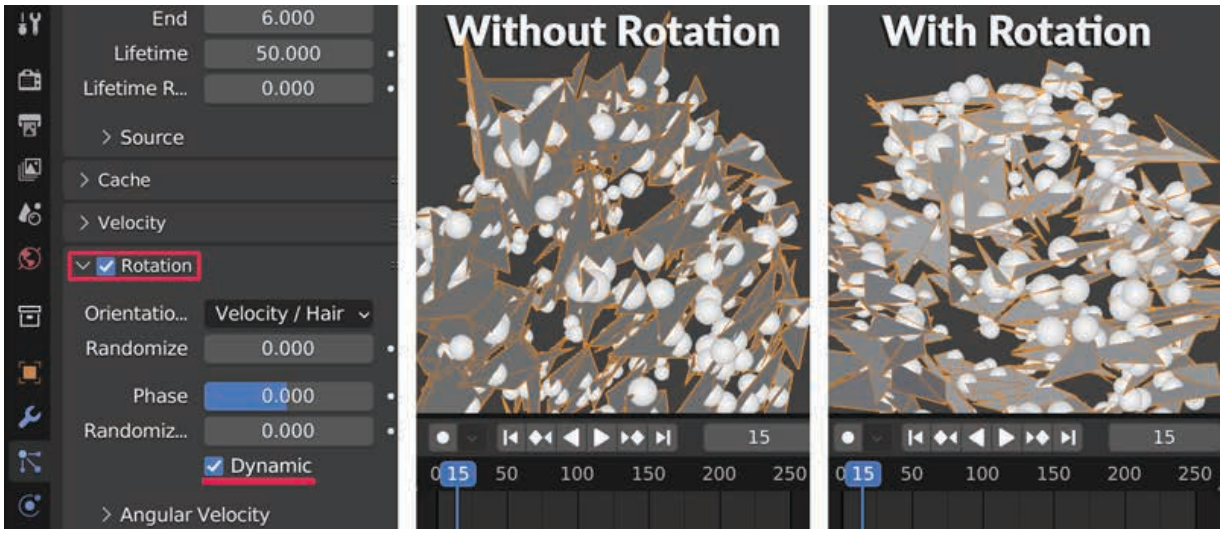


Reduce the **End** frame so all the particles appear faster.

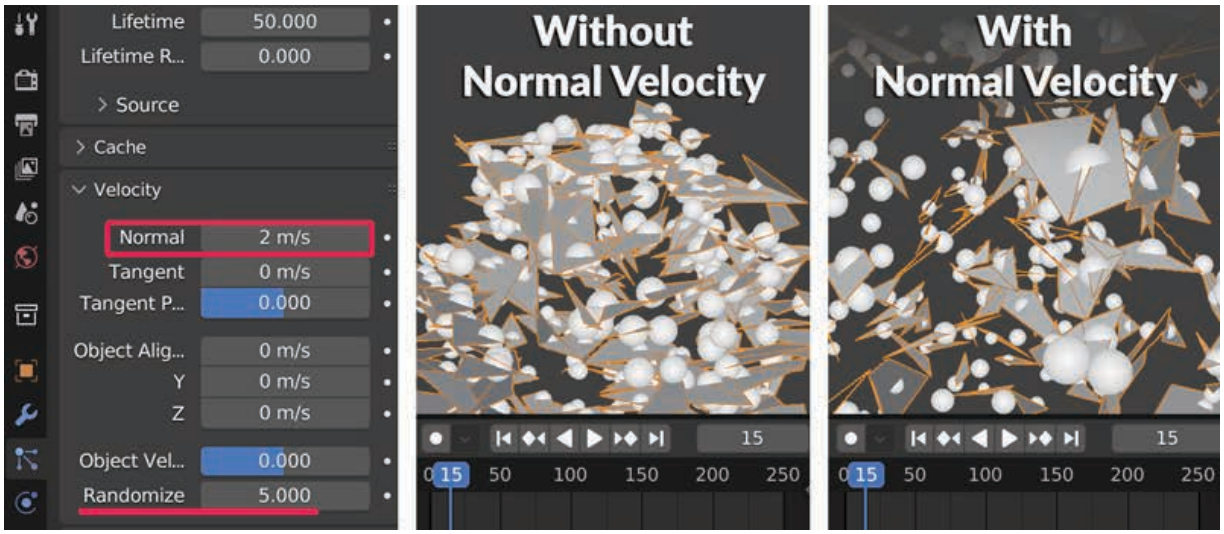


Click here to return to the index.

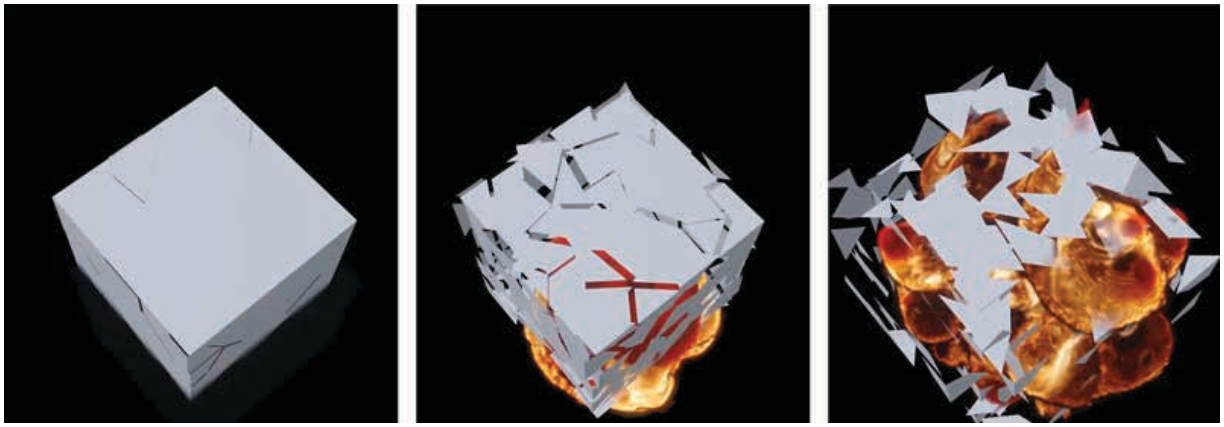
Turn on **Rotation** and **Dynamic** so the particles will rotate randomly.



Increasing the **Normal** Velocity will increase the power of the explosion. Then increase the Velocity **Randomize** value to make it even more chaotic.

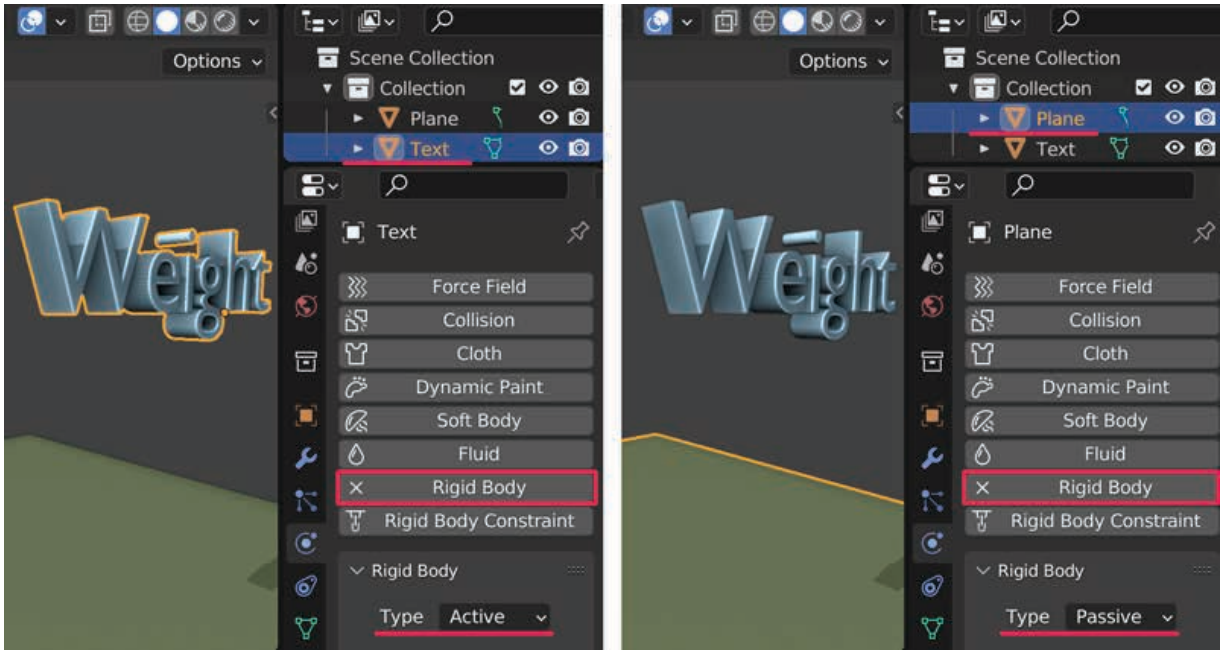


Now add an **OpenVDB explosion** as explained in a previous tutorial.

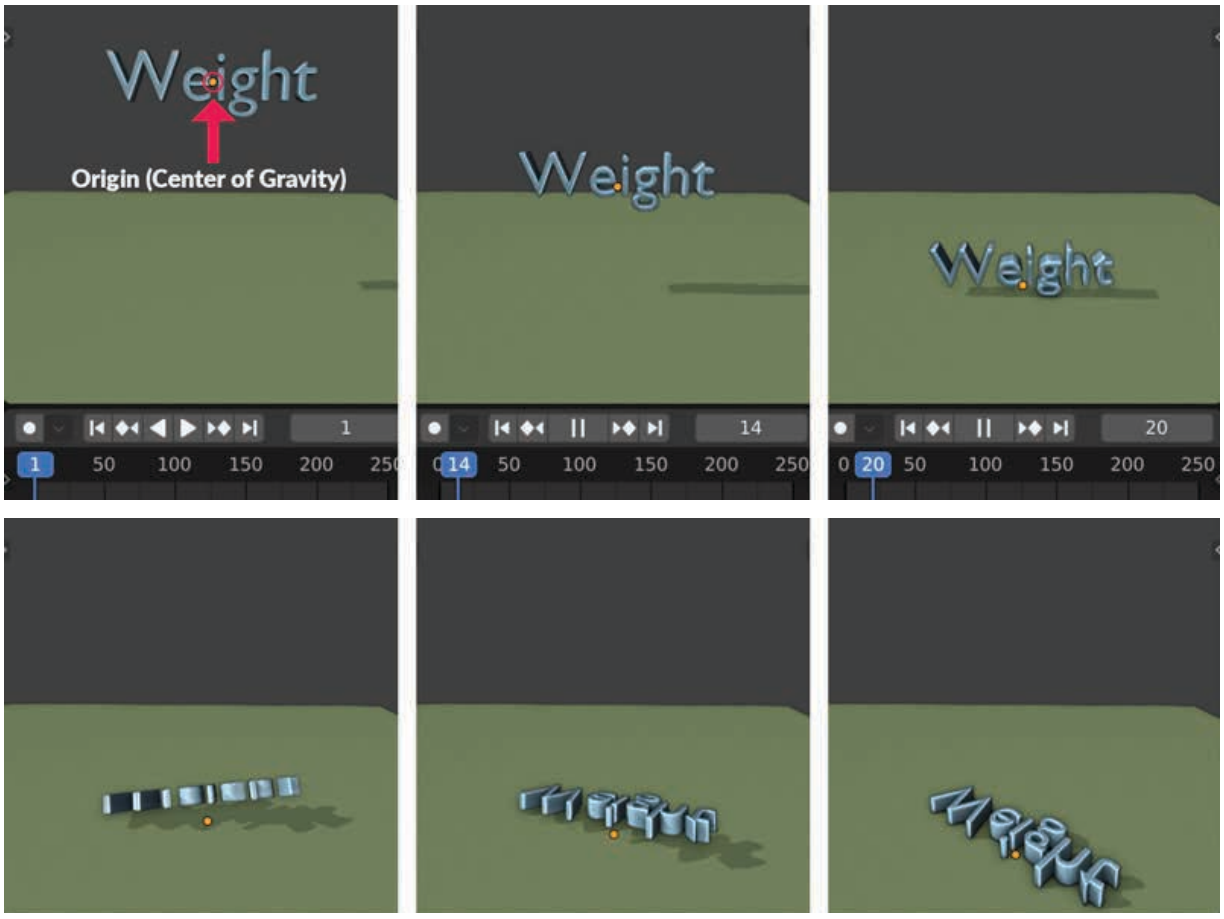


Distribute Weight of a Rigid Body

Did you know that the **origin** point of a Rigid Body object determines how its weight is distributed? Here are three ways to change the origin point so that you can let objects fall down differently. Set your object to **Active Rigid Body**. Then set the floor to a **Passive Rigid Body**.

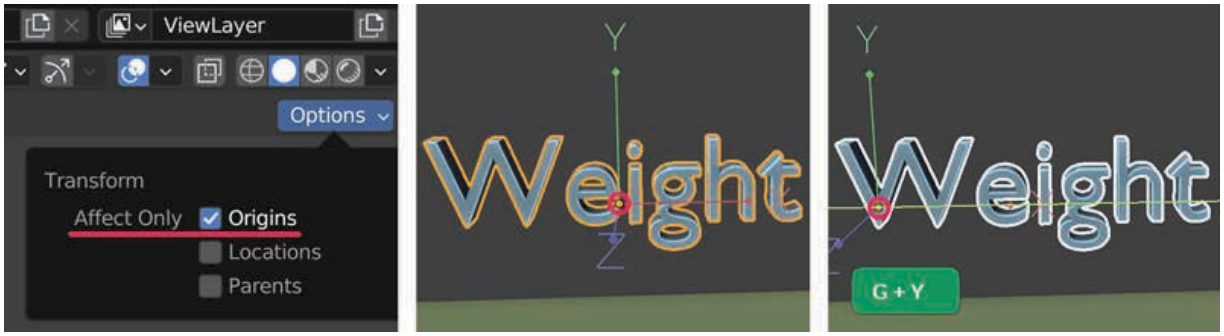


Now **press the Spacebar** to play the simulation. The falling object has its center of weight set to wherever the origin is.

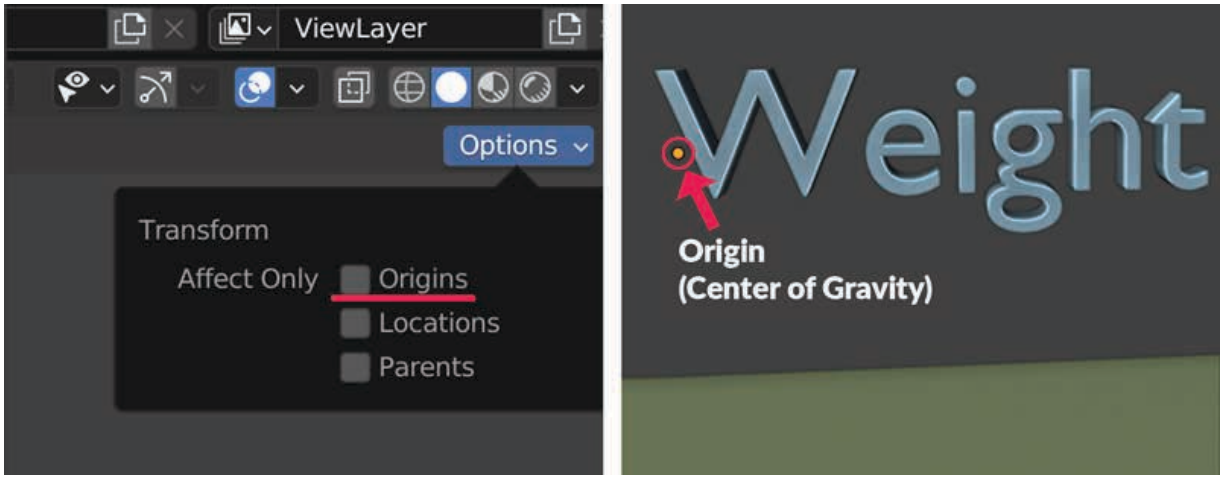


Click here to return to the index.

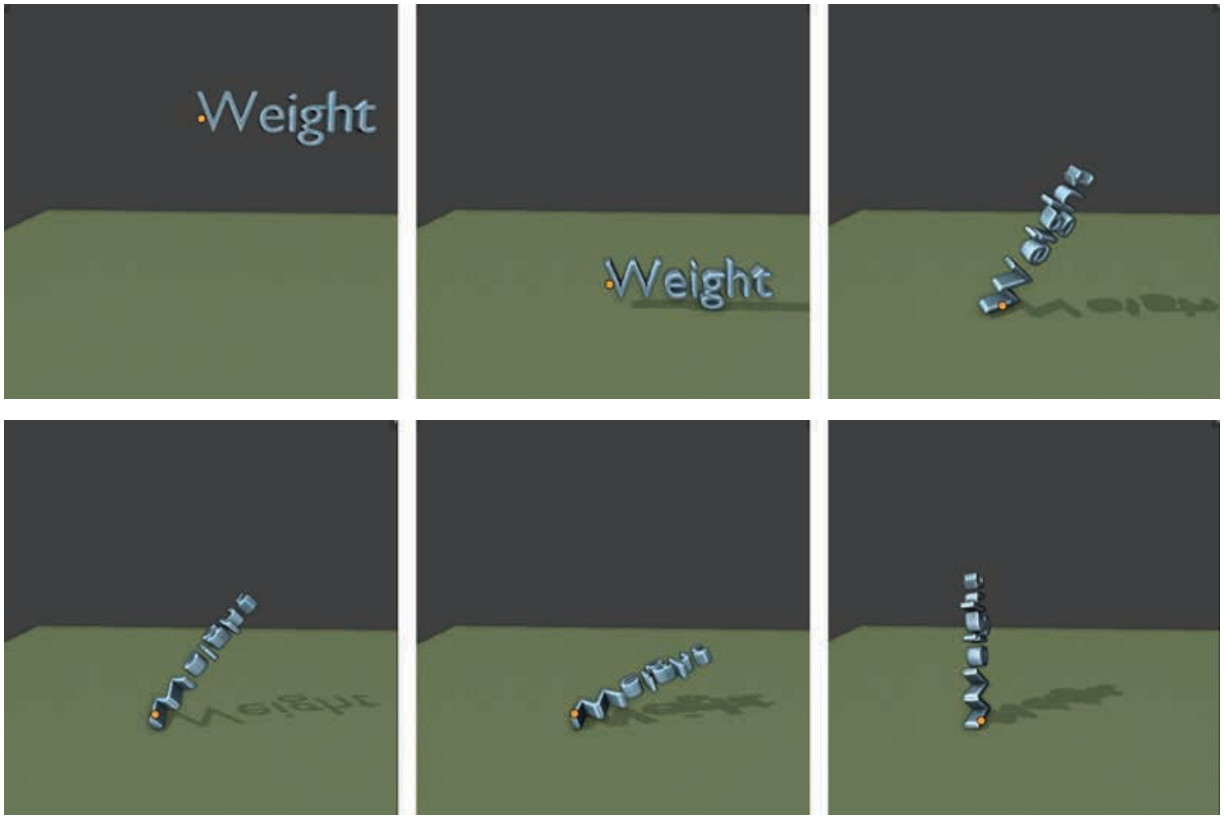
To change this, go to **Options** and enable **Origins**. Now you can transform only the origin position of the object.



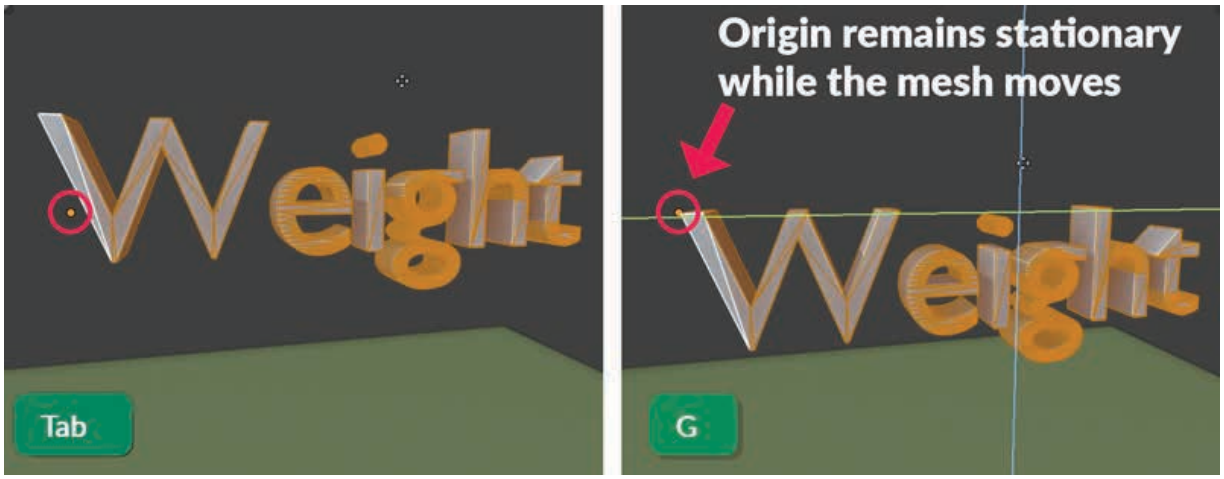
Uncheck it once you are done. The objects weight has shifted to where the origin is.



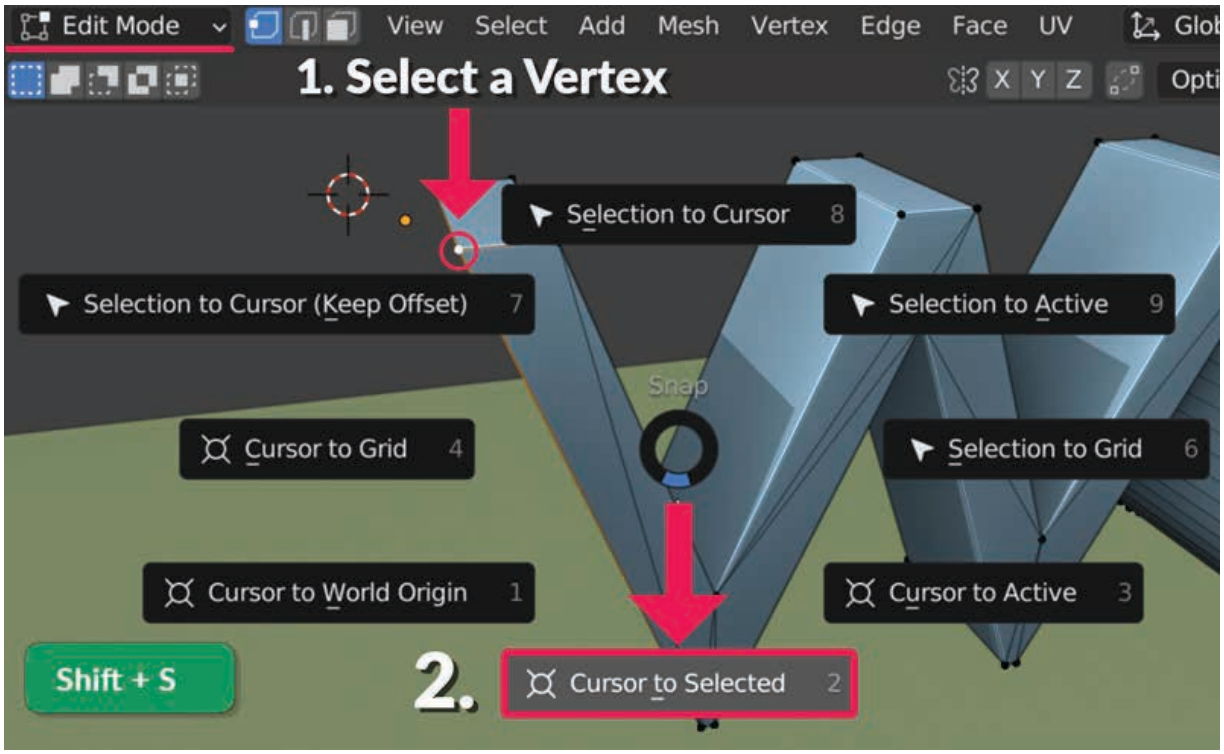
So it will fall differently when you press the Spacebar.



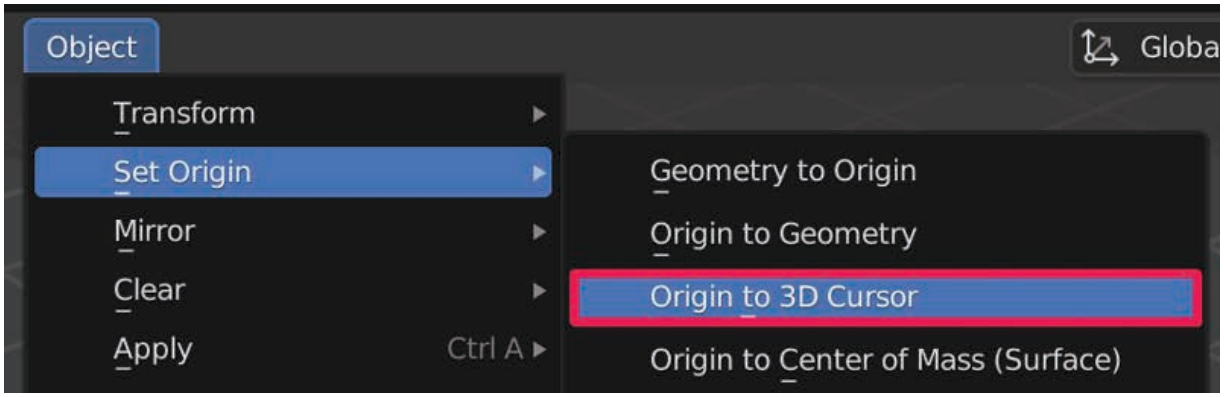
Alternatively you can go to Edit Mode and move the mesh around the origin.



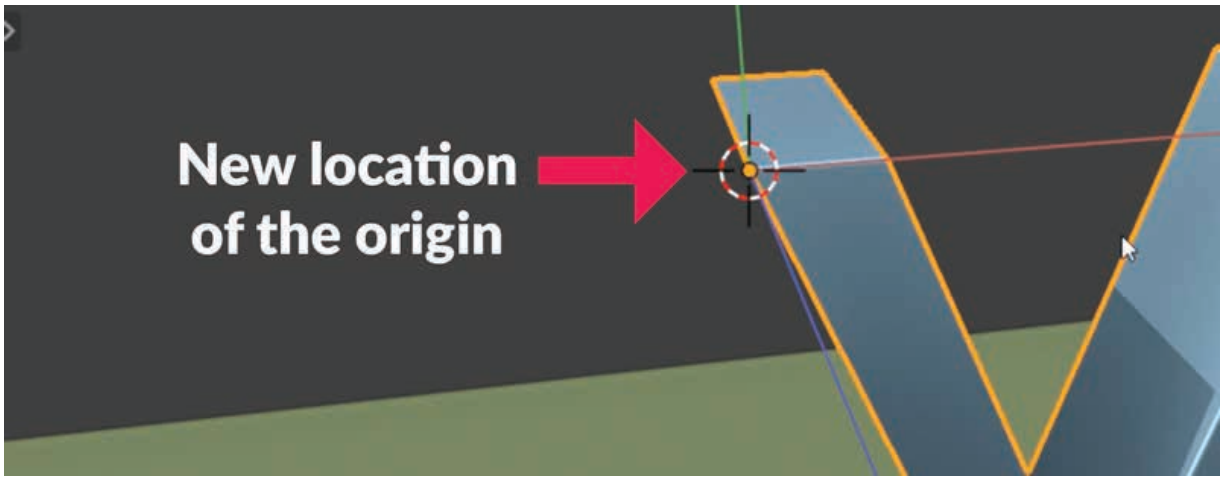
To be very precise you can also select a vertex in Edit Mode and then set the 3D cursor to there by **pressing Shift+S** and choosing **Cursor to Selected**.



Then in Object Mode set the origin to the 3D cursor by going to Object > Set Origin > **Origin to 3D Cursor**.



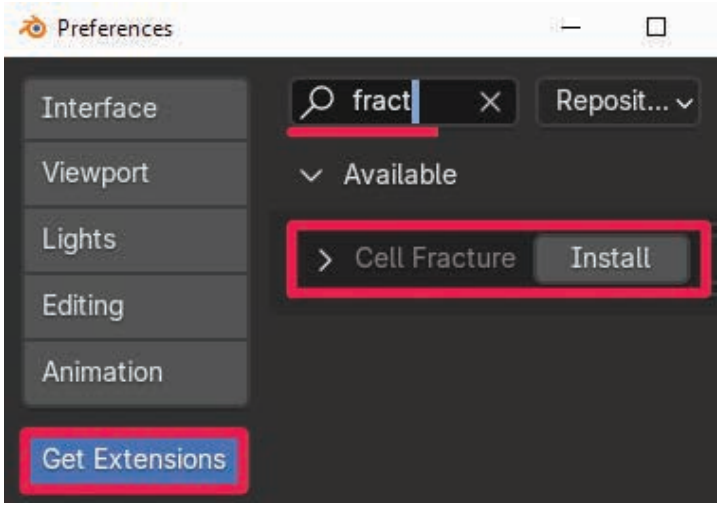
Now the origin is at the location of the vertex you selected.

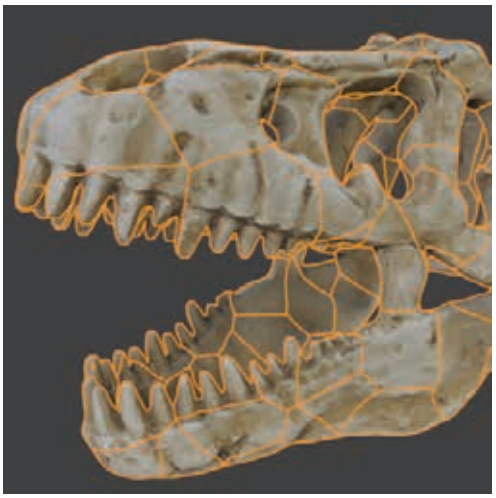


Hide cracks pre-fracture

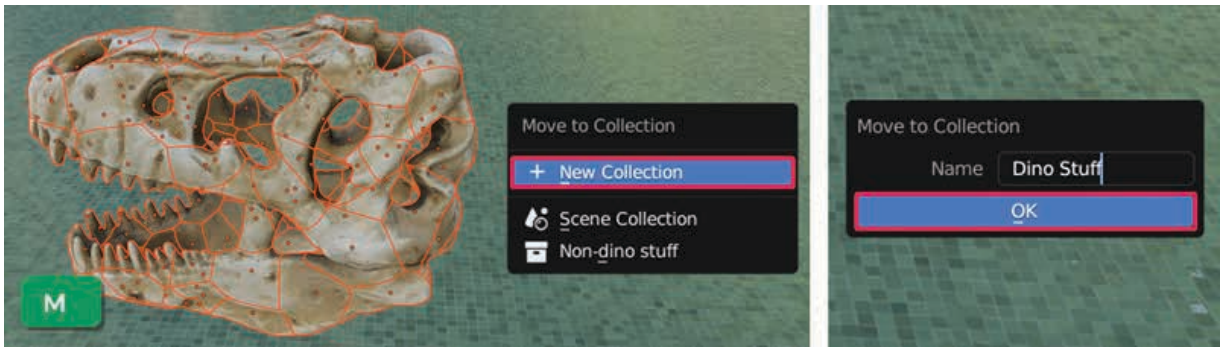
When you have a rigid body simulation in combination with a pre fractured model, you need to be able to hide those fractured pieces until just the right moment, at which point you hide the original (non-fractured) object. But, how do you set keyframes for the render visibility of a Collection? That's not possible... or is it?

Use the Cell Fracture add-on to prefracture the model.

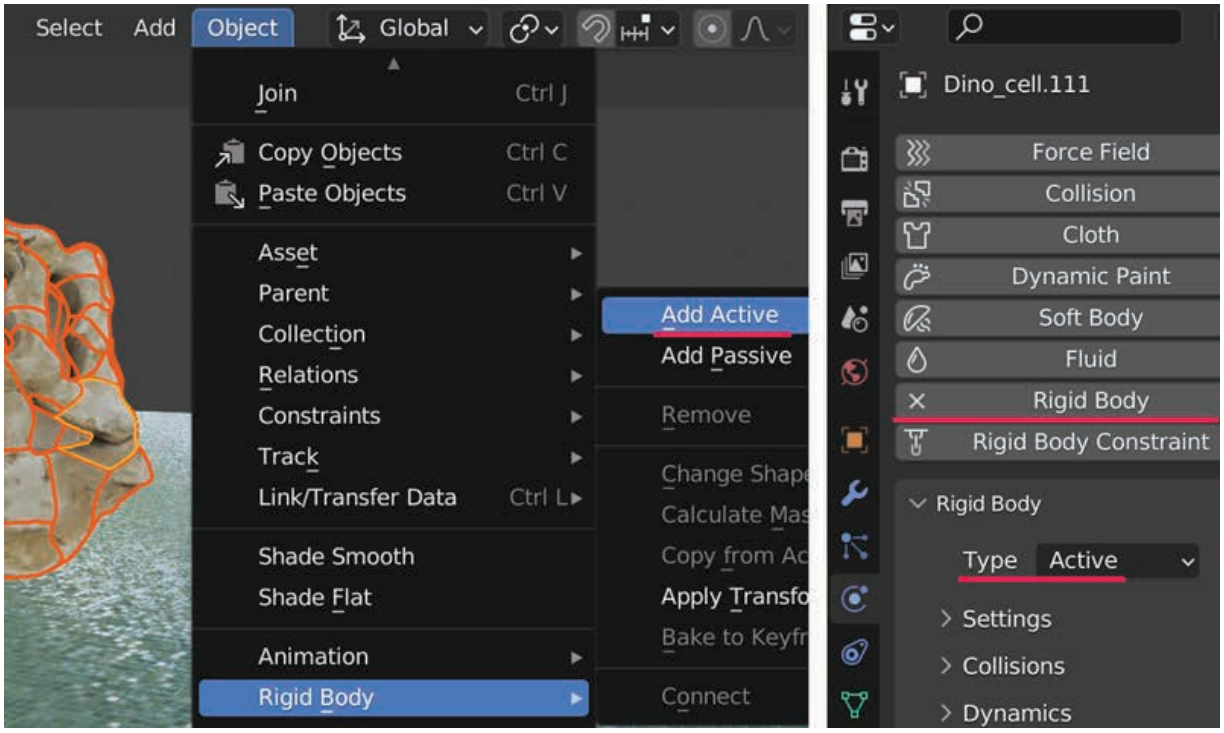




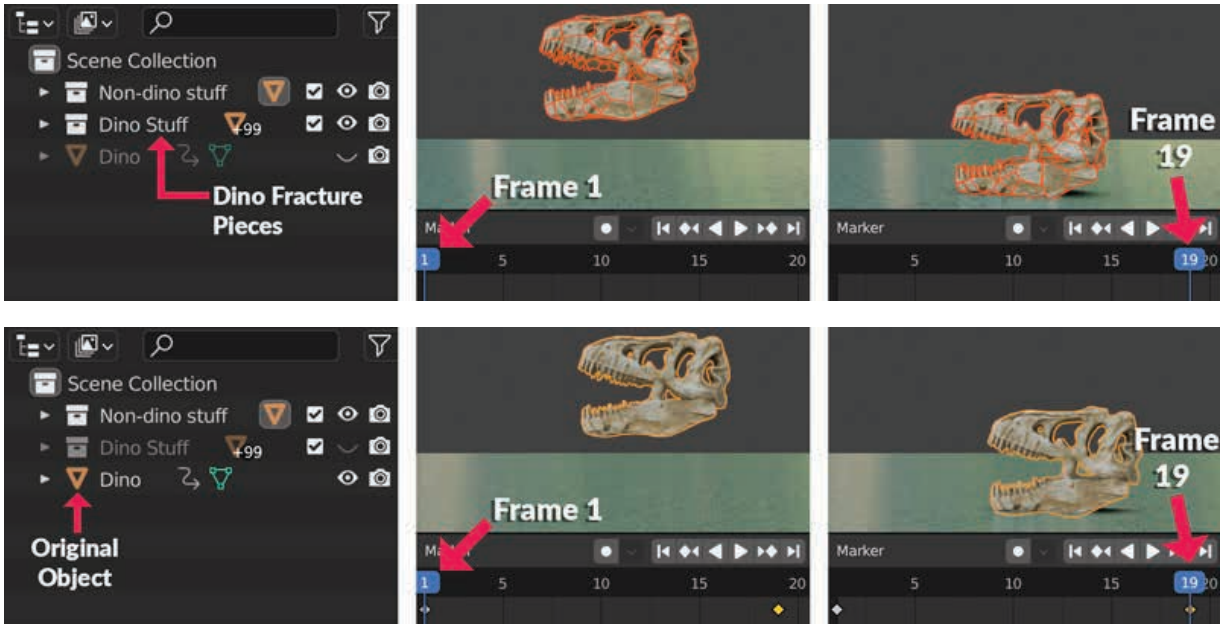
Press M to create a collection for these pieces.



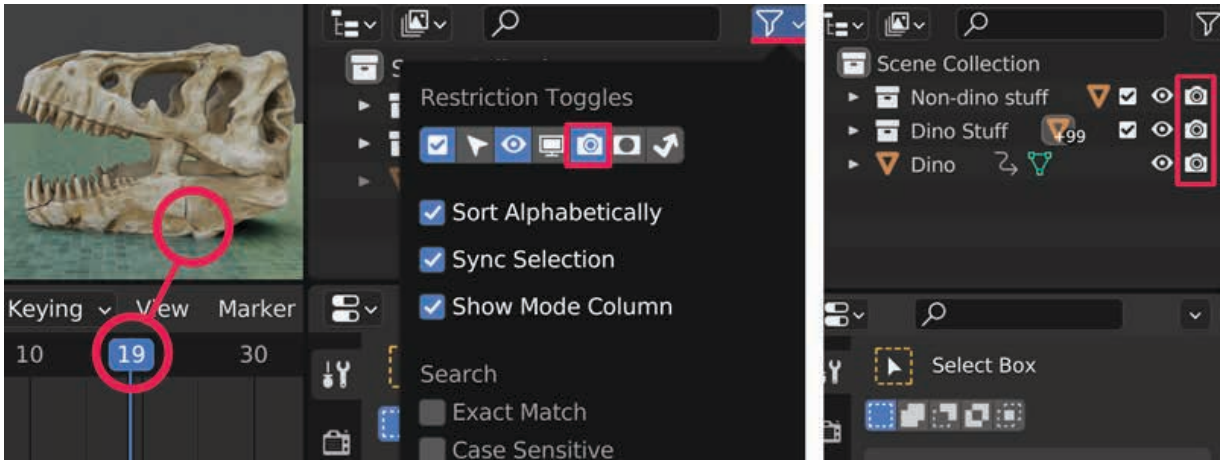
Go to Object > Rigid Body > **Active**.



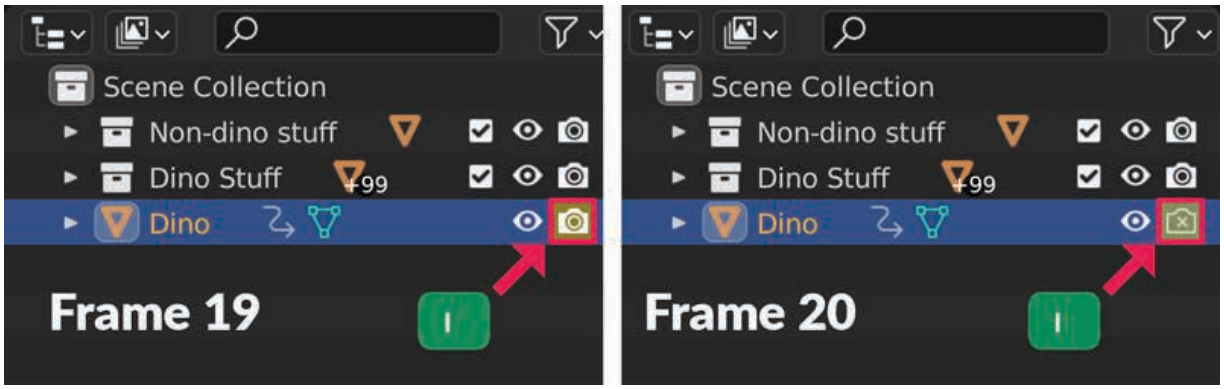
Set location keyframes for the original model to match the falling of the rigid bodies.



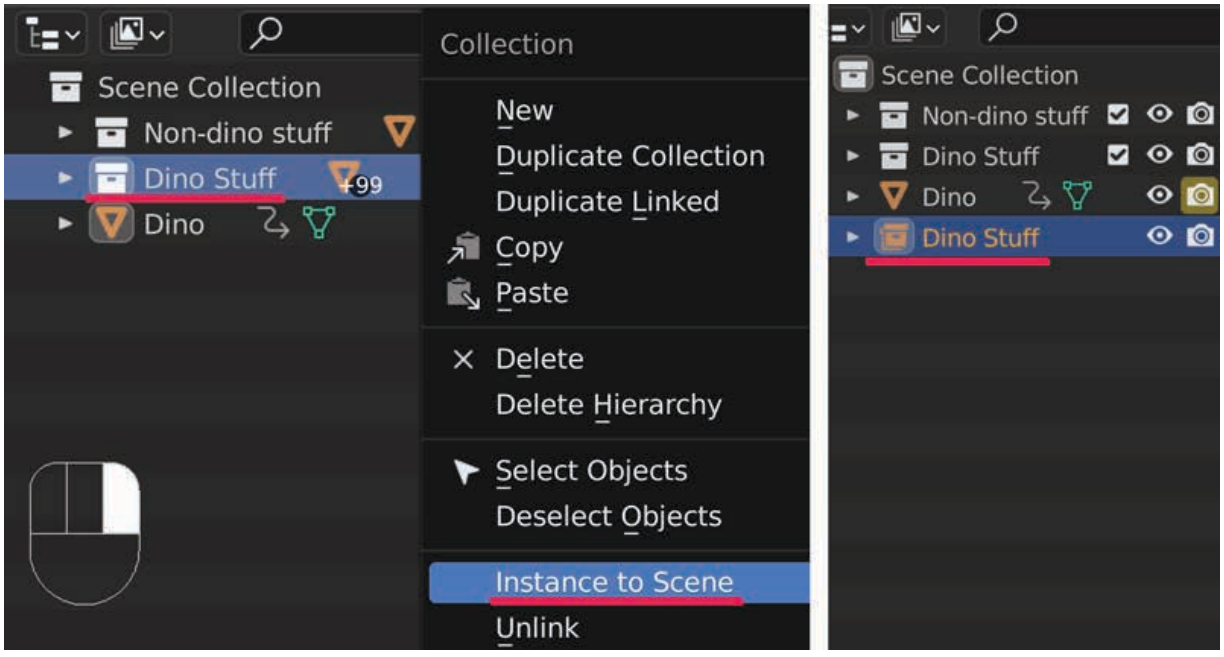
Enable the render visibility toggle. Go to the frame where the object needs to start fracturing.



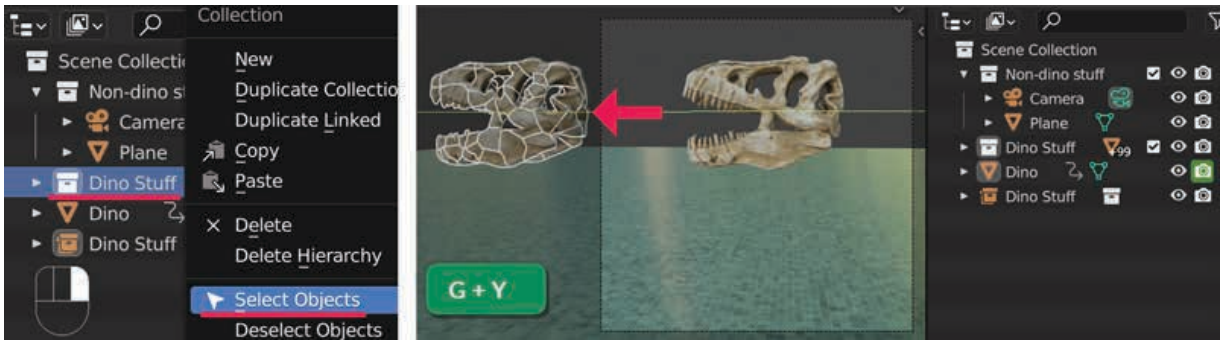
For the original object **press i** with the mouse cursor over the render visibility icon, then go to the next frame, turn off visibility and set another keyframe.



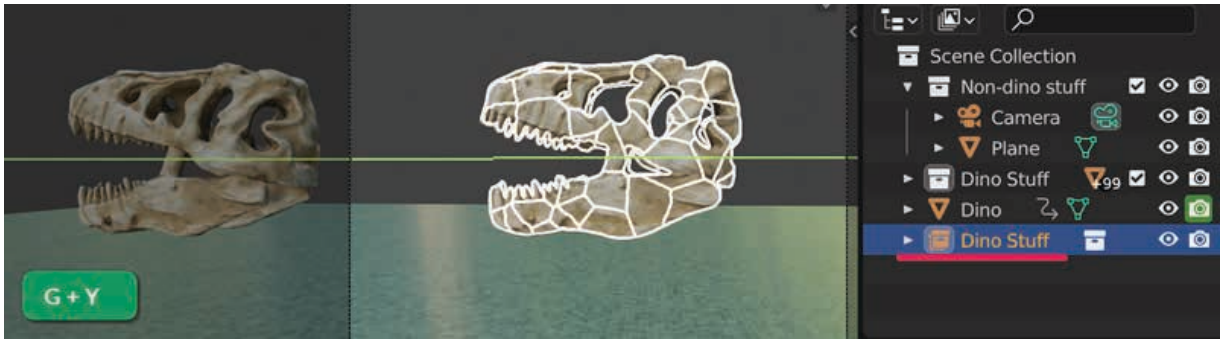
That doesn't work for a Collection unfortunately. So right-click on the collection with the fractured pieces and choose "Instance to Scene".



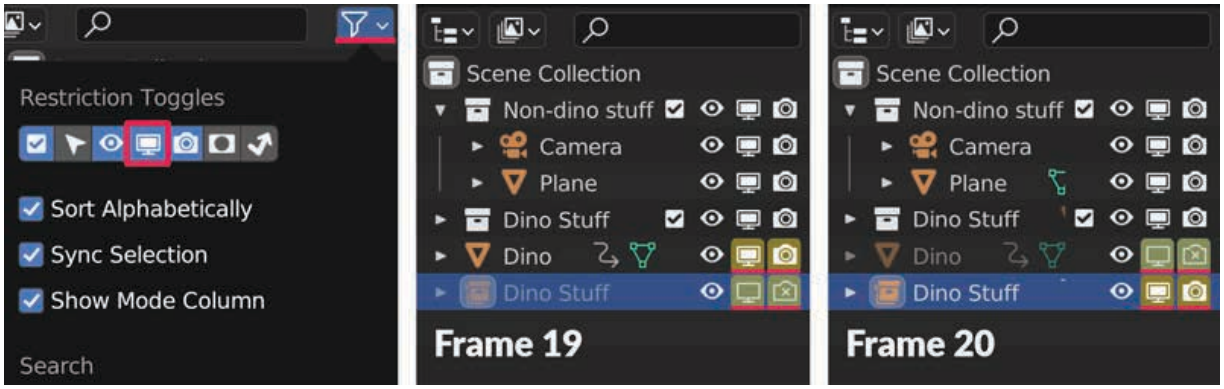
Select everything in the original collection and move it out of view somewhere.



Then move the collection instance to the right location and set keyframes so that it is visible in the render at the same frame that the original object becomes invisible in the render.



To avoid confusion you can also set identical keyframes for the viewport visibility.

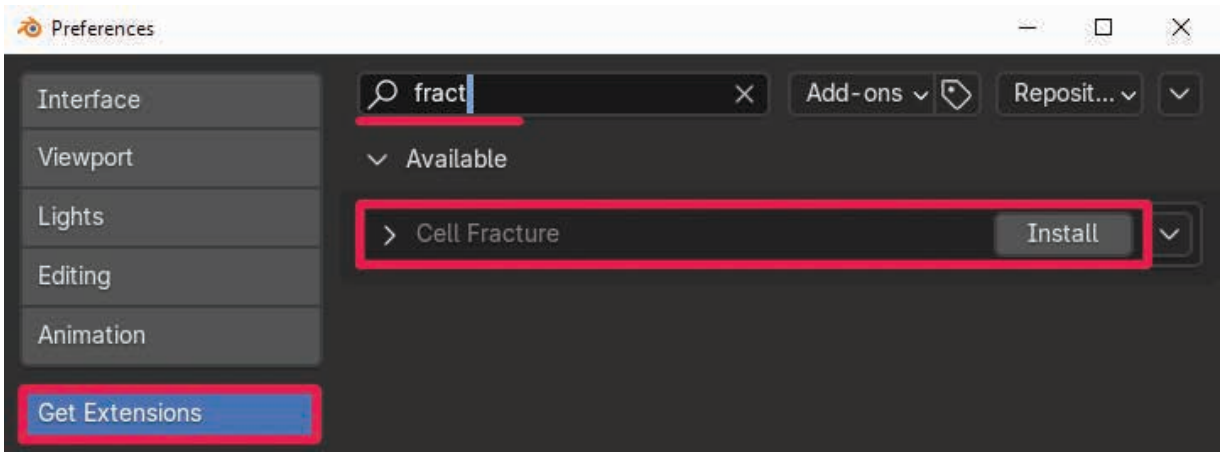


Now when you press the Spacebar, the object will fracture at the moment of impact without fracture lines being visible beforehand.

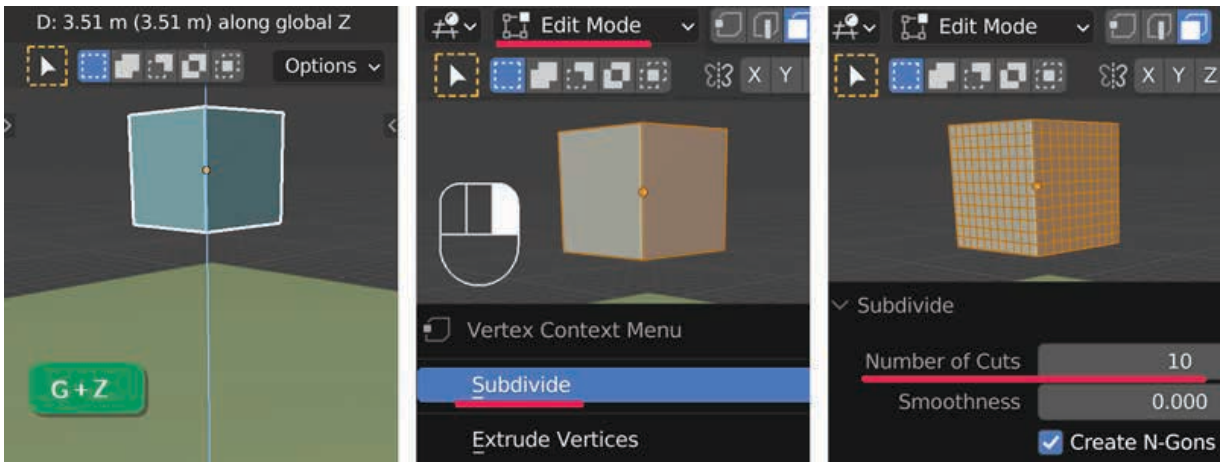
Cell Fracture Basics

Let's look at the easiest way to create cell fracture destruction, as well as how to set up an inside material.

In Preferences enable the Cell Fracture add-on.

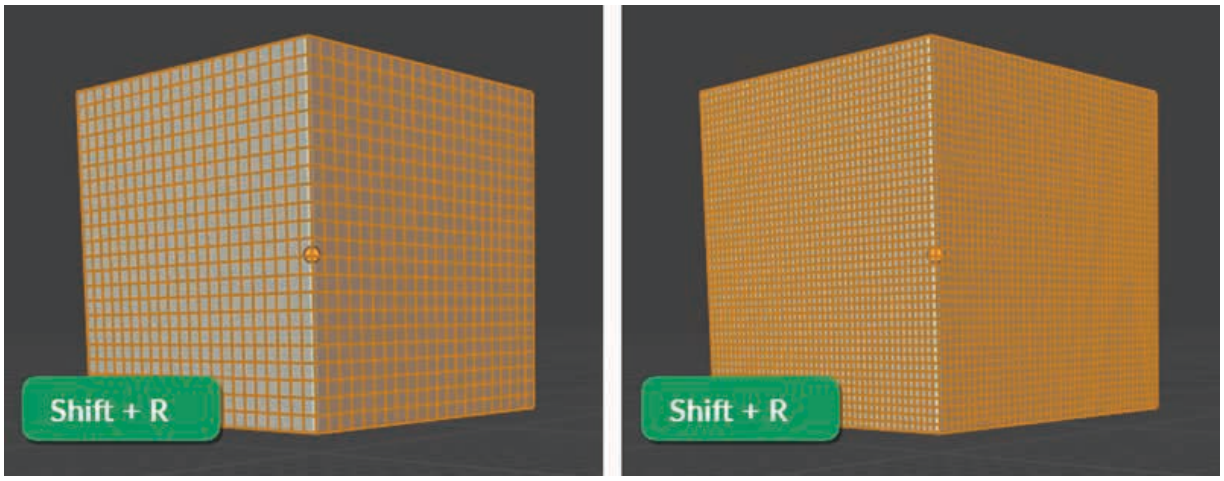


Move the default Cube up and in Edit Mode right-click and choose Subdivide.

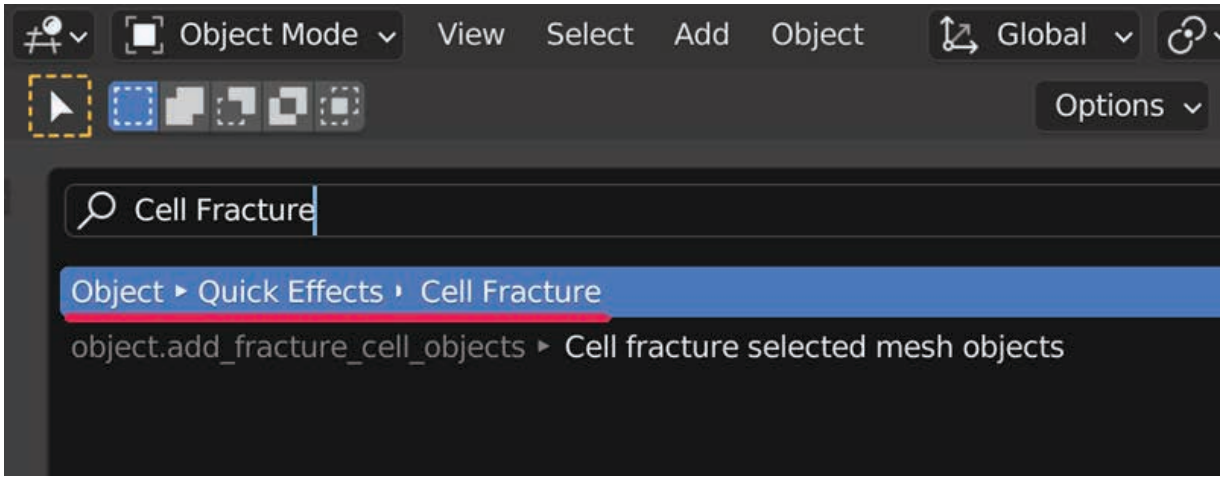


Click here to return to the index.

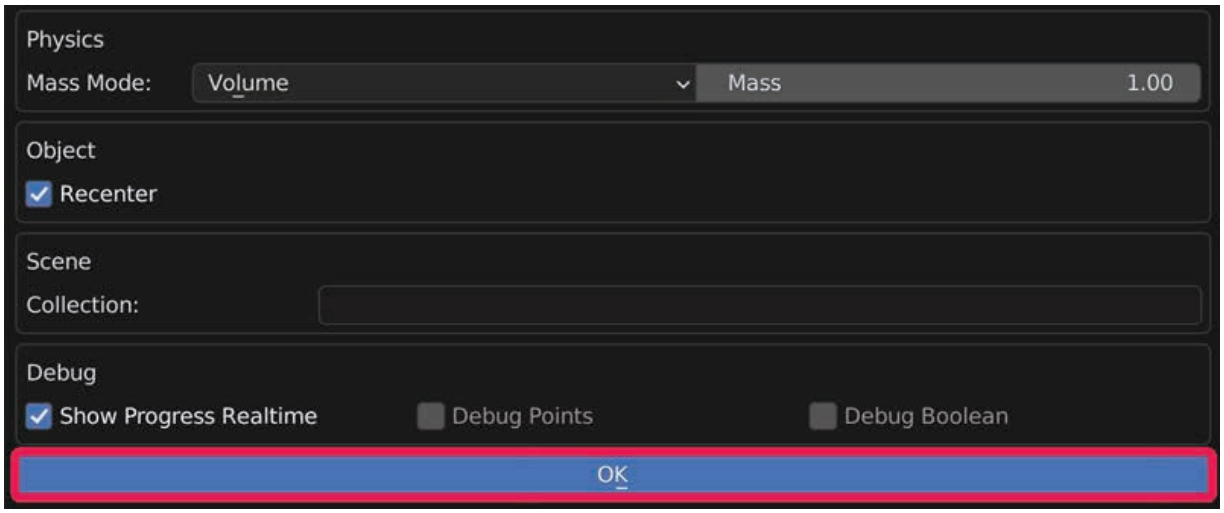
Press Shift+R a few times to repeat that last step.



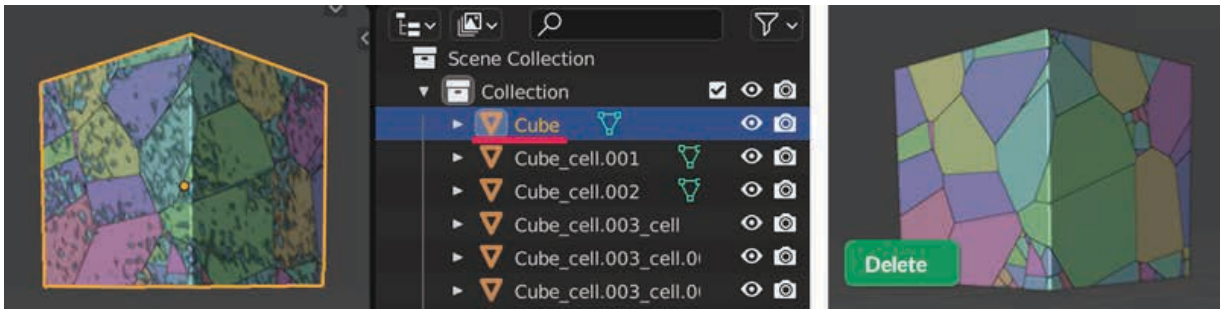
In Object Mode press F3 and search for Cell Fracture.



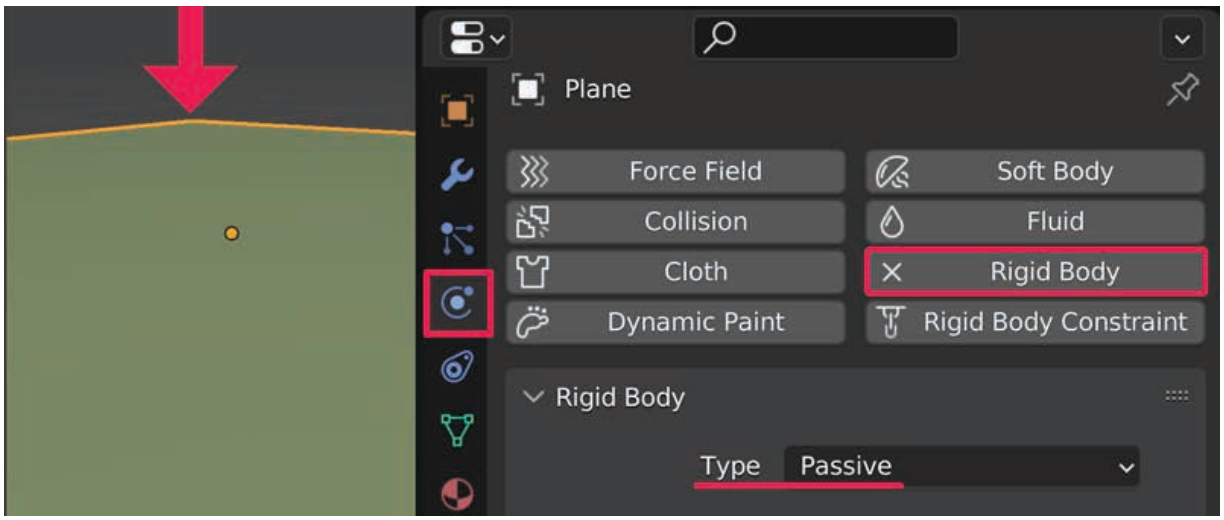
Click OK.



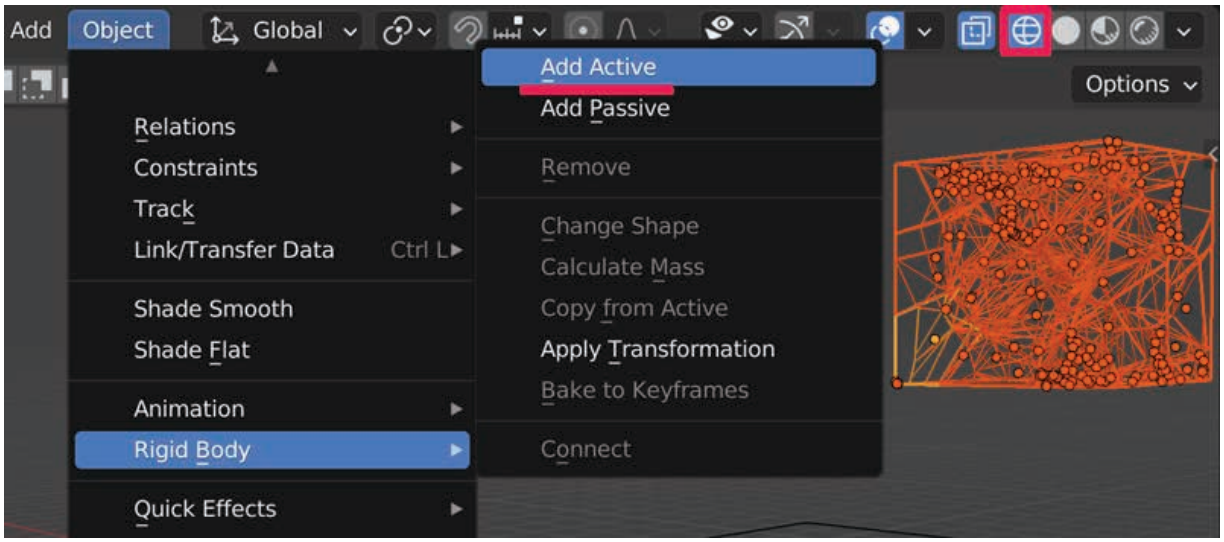
Select the original cube in the outliner and delete it.



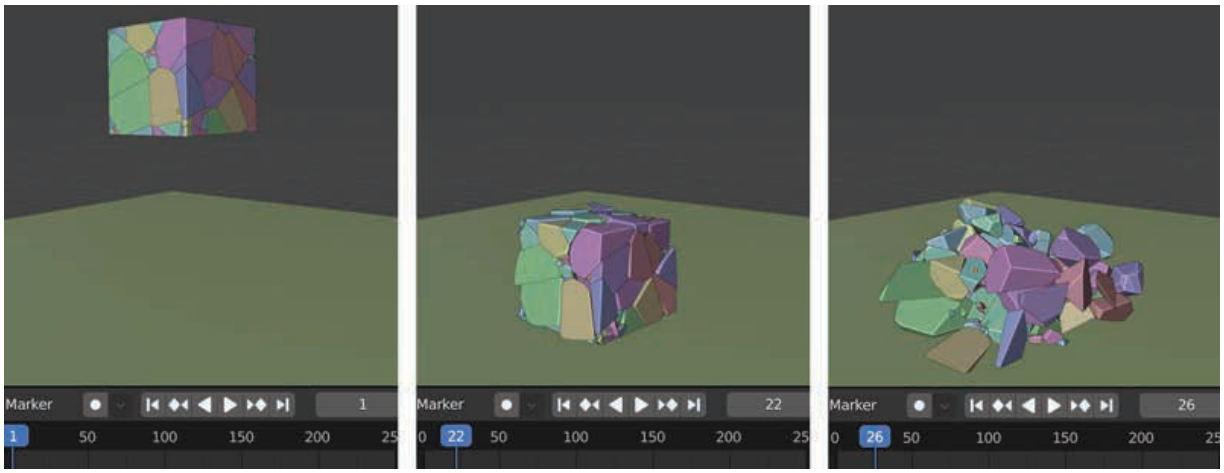
Add a Plane and enable Rigid Body, set Type to Passive.



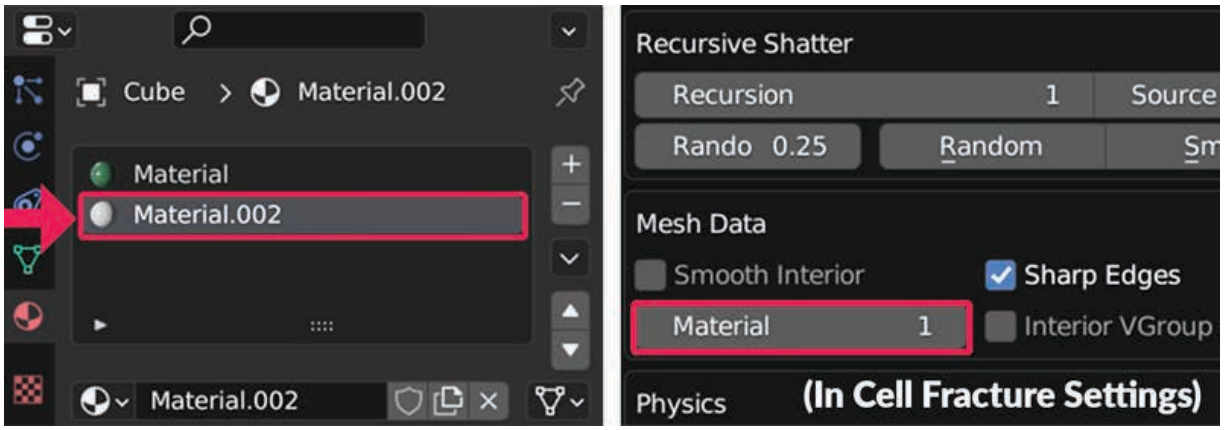
In Wireframe mode select all the shards and go to Object > Rigid Body > Add Active.



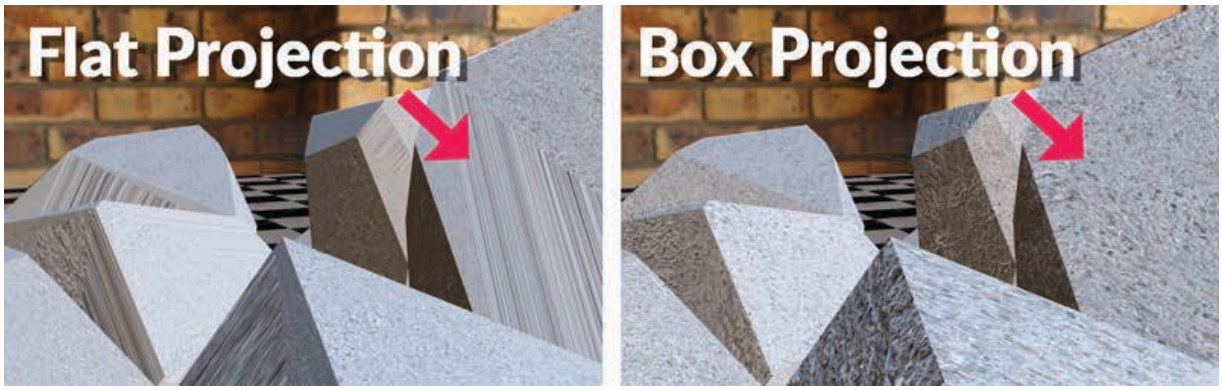
Now press the Spacebar and the cube will fall to pieces on the Plane.



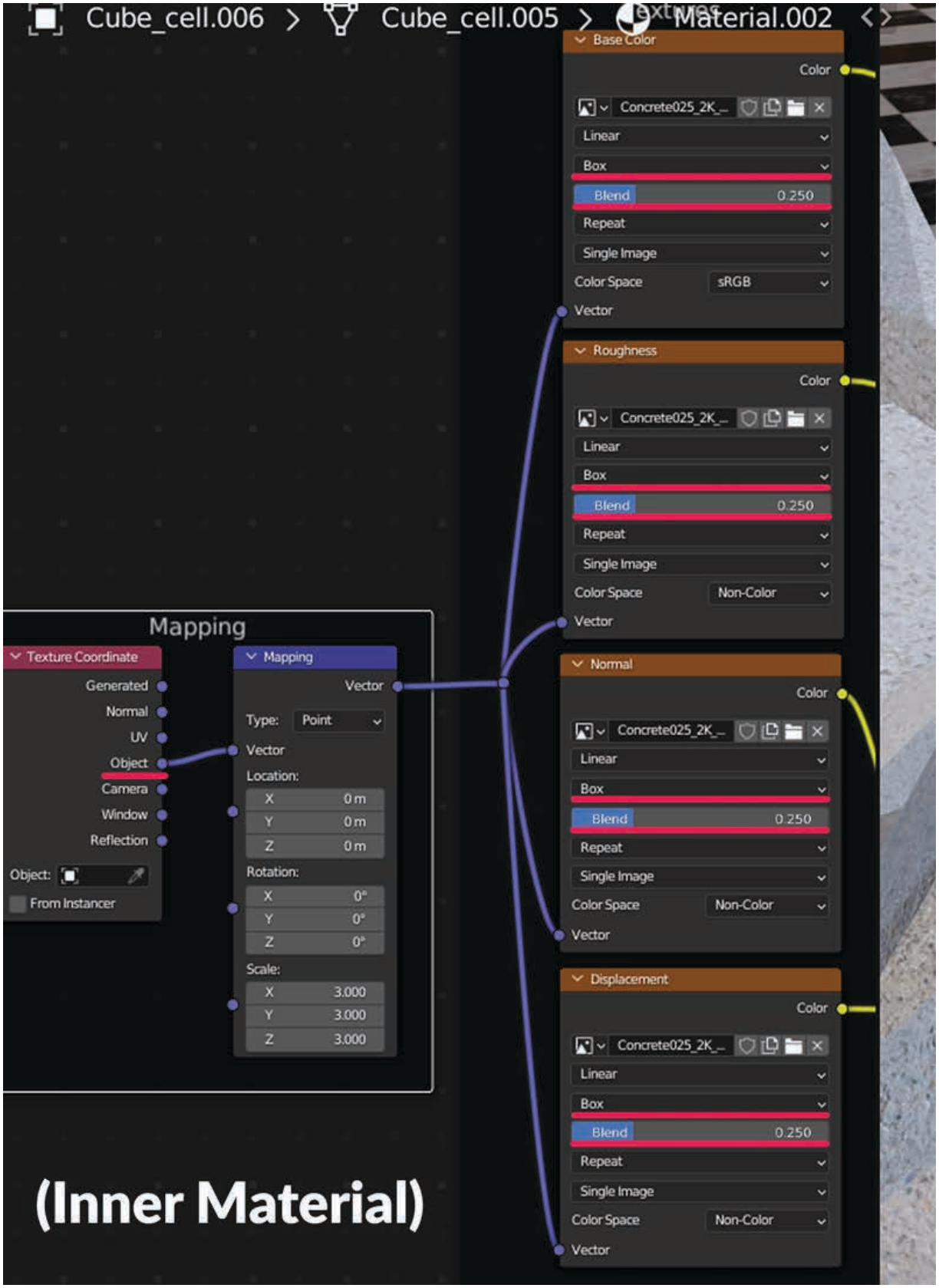
In case your object has two materials, one for the outside and one for the inside, you simply have to set Material to 1 in the Cell Fracture options before clicking OK.



Also, for the inside material you'll want to set Mapping to "**Object**" and Projection Type to "**Box**" with some blending (a Blend value of 0.25 works well in most cases). Because if it's set to "UV" it will just stretch like the picture below left.



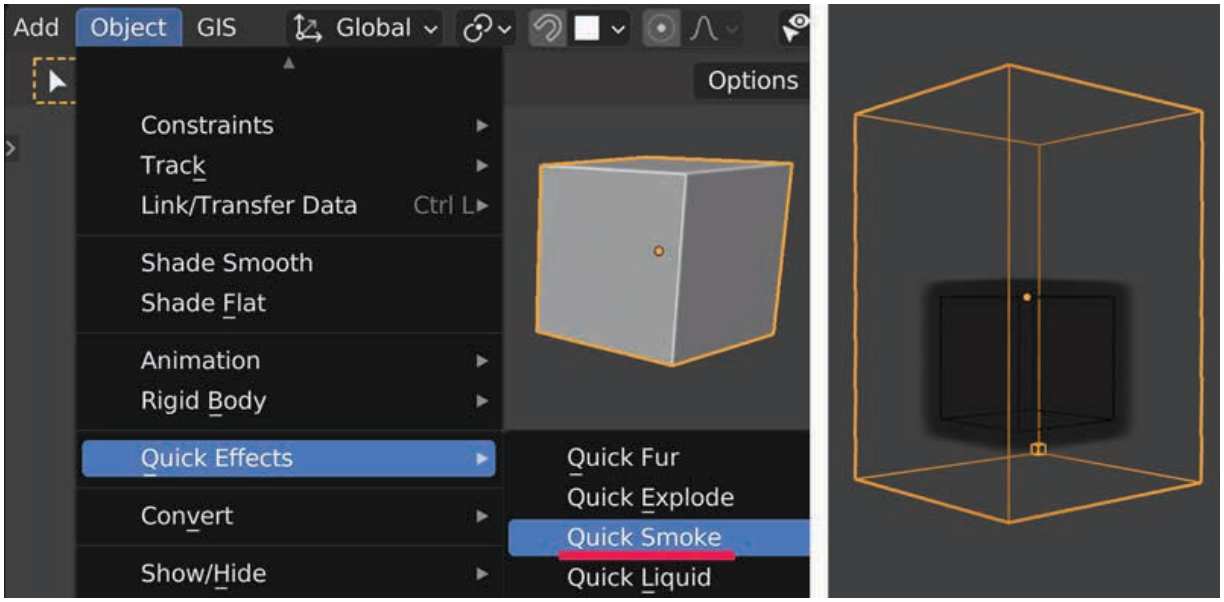
The nodes with Box Mapping and a Blend value of 0.25:



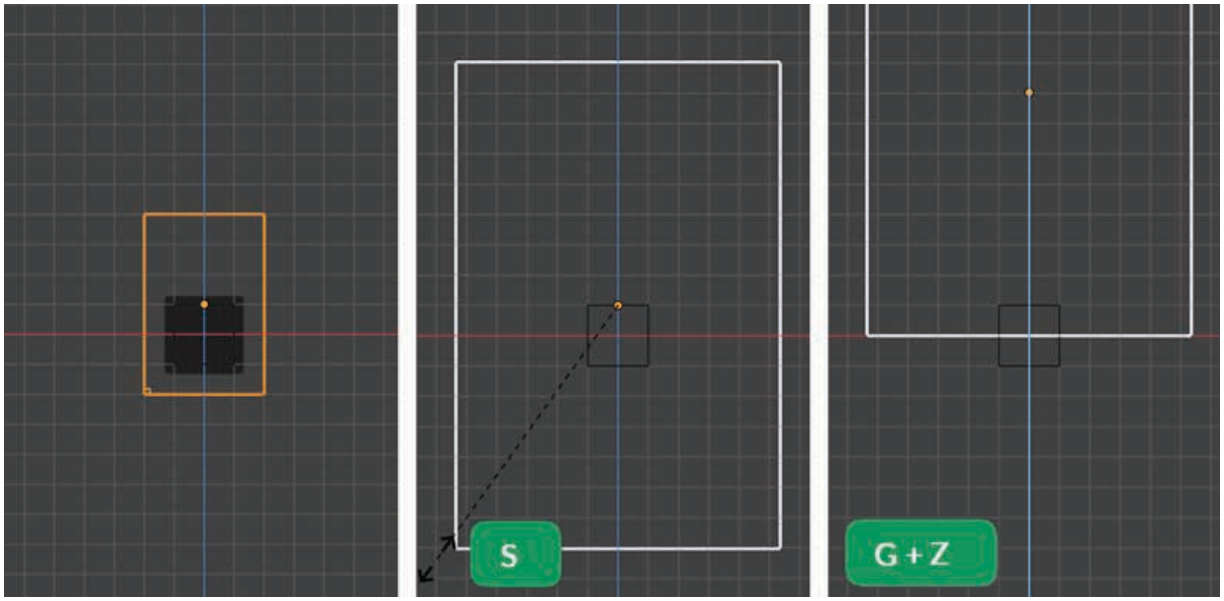
Quick Mantaflow Explosion

Mantaflow is the new liquid system in Blender. It has many possible settings and can seem daunting, but creating a basic explosion is actually not that much work. This render was done with only 1 sample in Cycles with the Optix Denoiser.

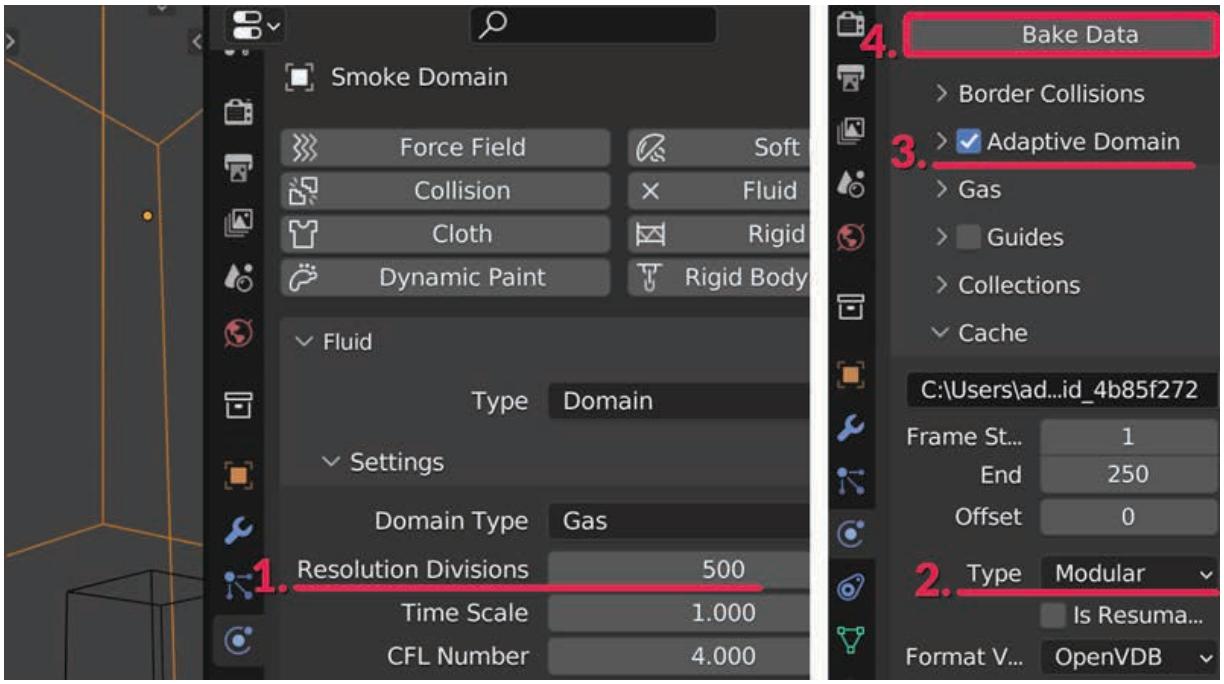
Select the default cube and go to objects > quick effects > smoke.



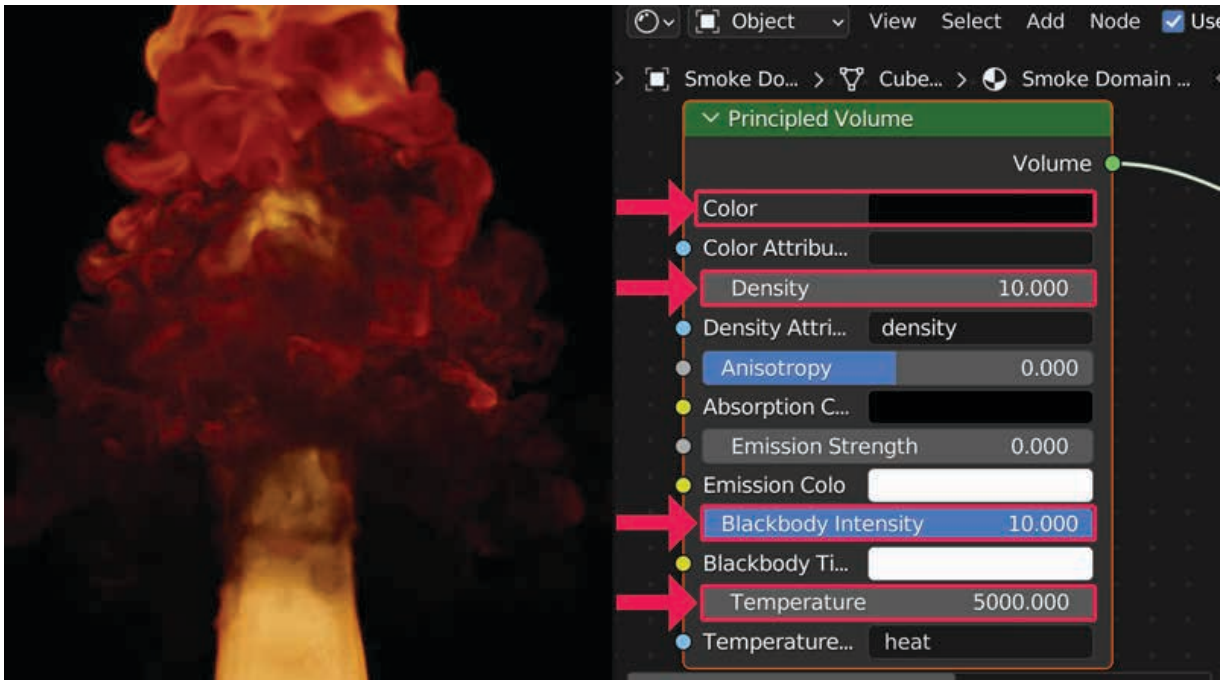
Scale the domain up and move it up along the Z-axis.



Turn on Adaptive Domain. Resolution divisions is the quality of the simulation, so higher values will be better but take longer to bake. Set cache type to modular and click on Bake... and wait.



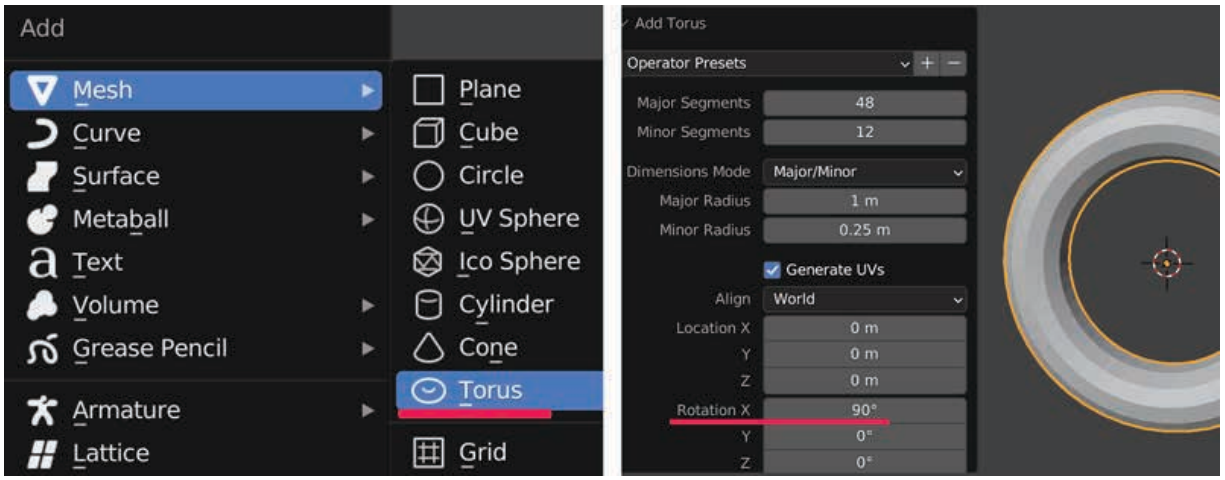
Open the shader editor and on the principled volume shader increase the density and the blackbody intensity. Increase the temperature and as temperature attribute set it to heat.



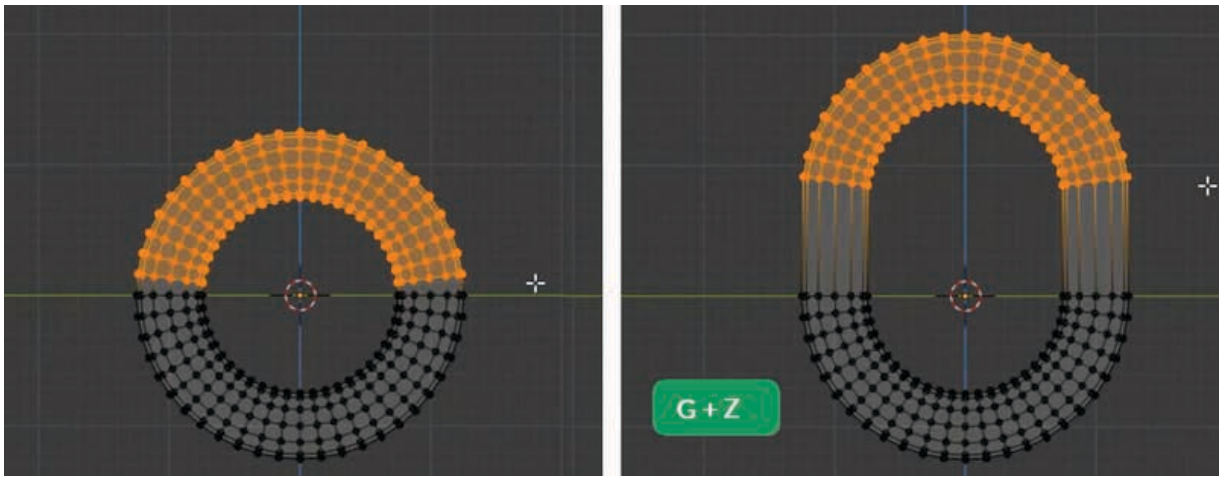
How to make chains

Here we look at creating chains that have rigid body dynamics. It's pretty easy!

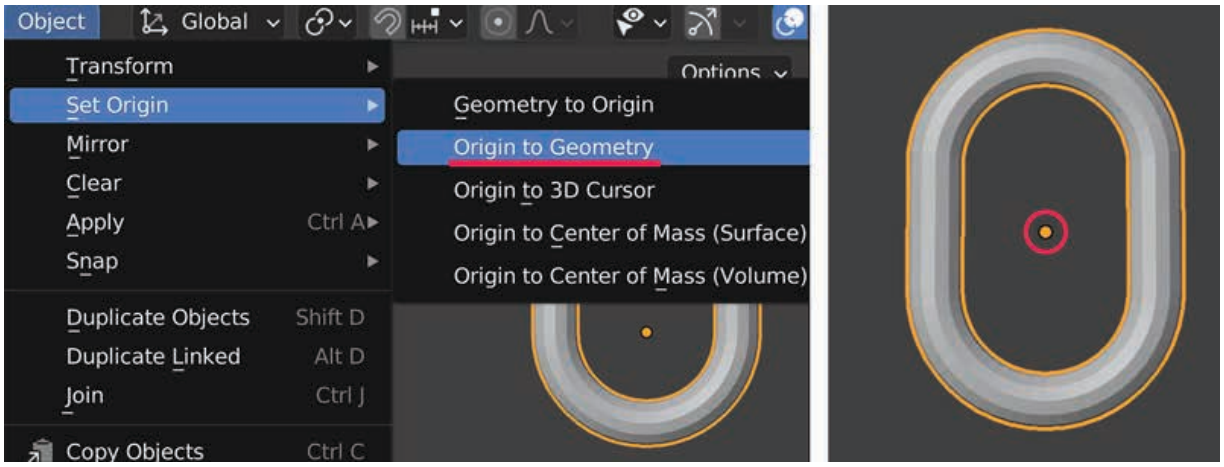
Add a Torus in Object mode by **pressing Shift+A** and choosing Mesh > **Torus**.



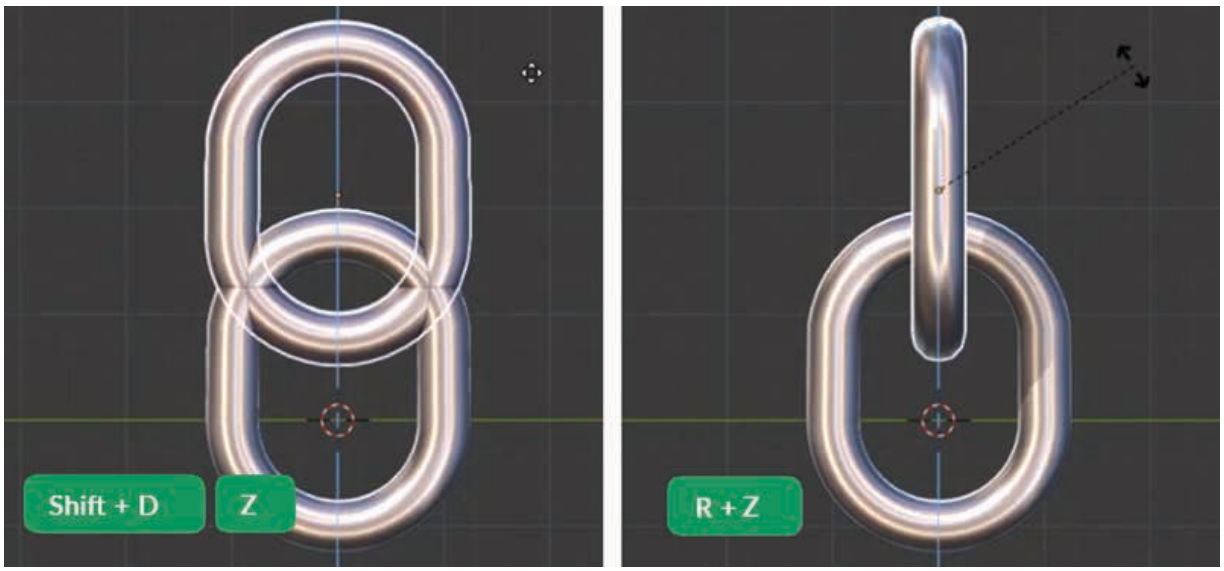
In Edit Mode, select the top part in **Front orthographic view** (1 on the Numpad) and in **X-ray mode** (press Z and choose Wireframe) so you also select the Vertices at the back. Move the selection up along the Z-axis by **pressing G+Z** and moving the mouse.



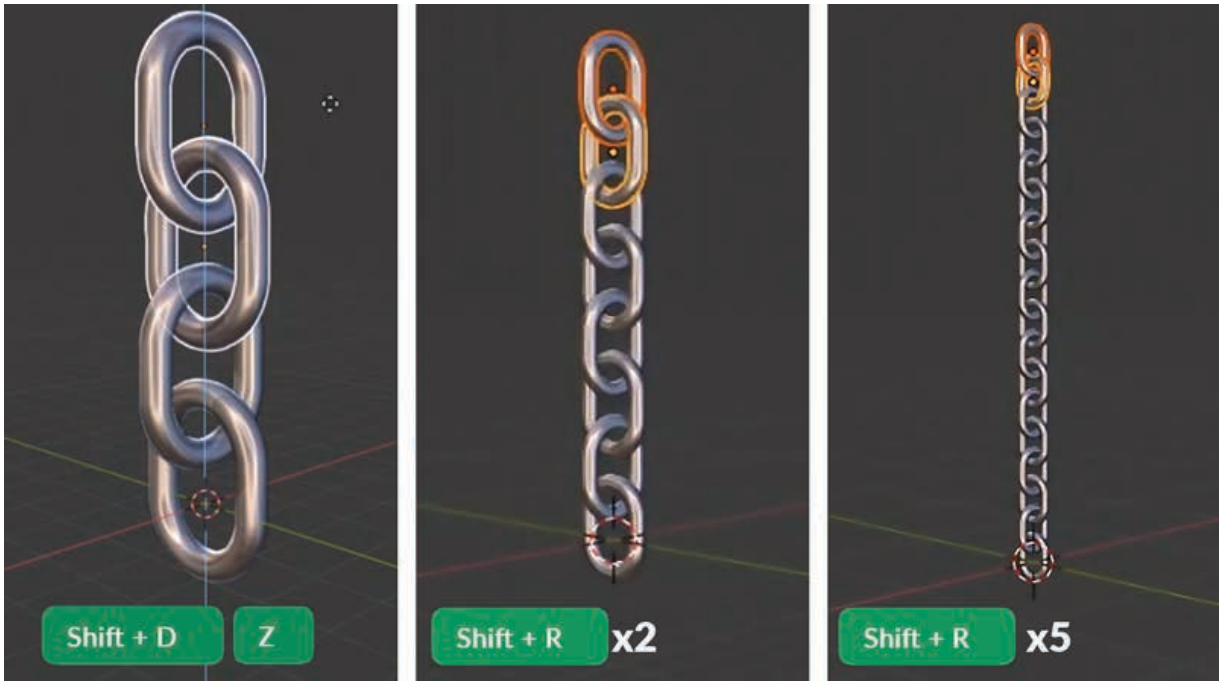
In Object Mode, go to Object> Origin> Set origin to geometry.



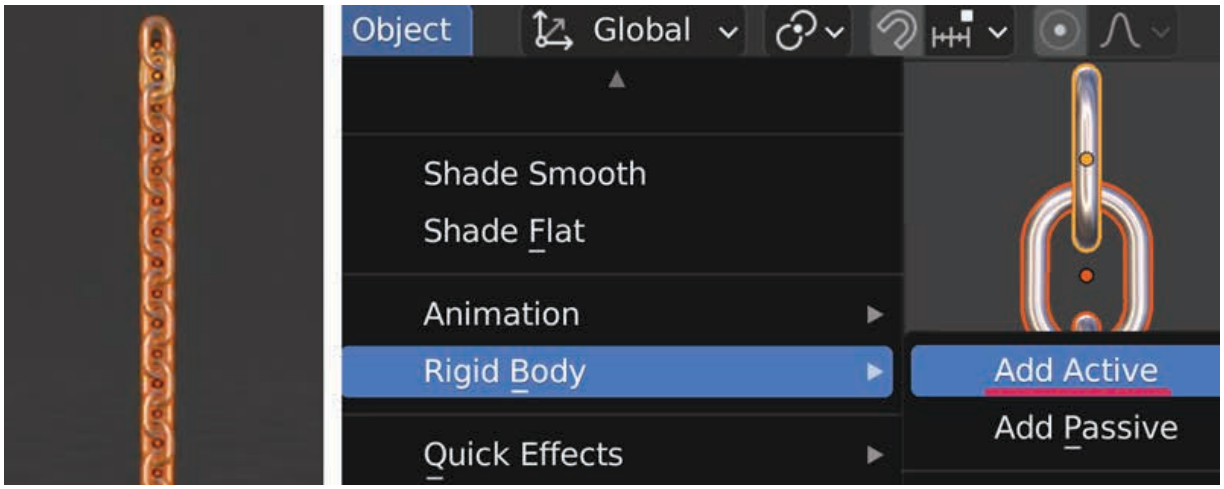
Duplicate the chain by **pressing Shift+D** and move it up and rotate it 90° by **pressing R+Z+90** and then Enter.



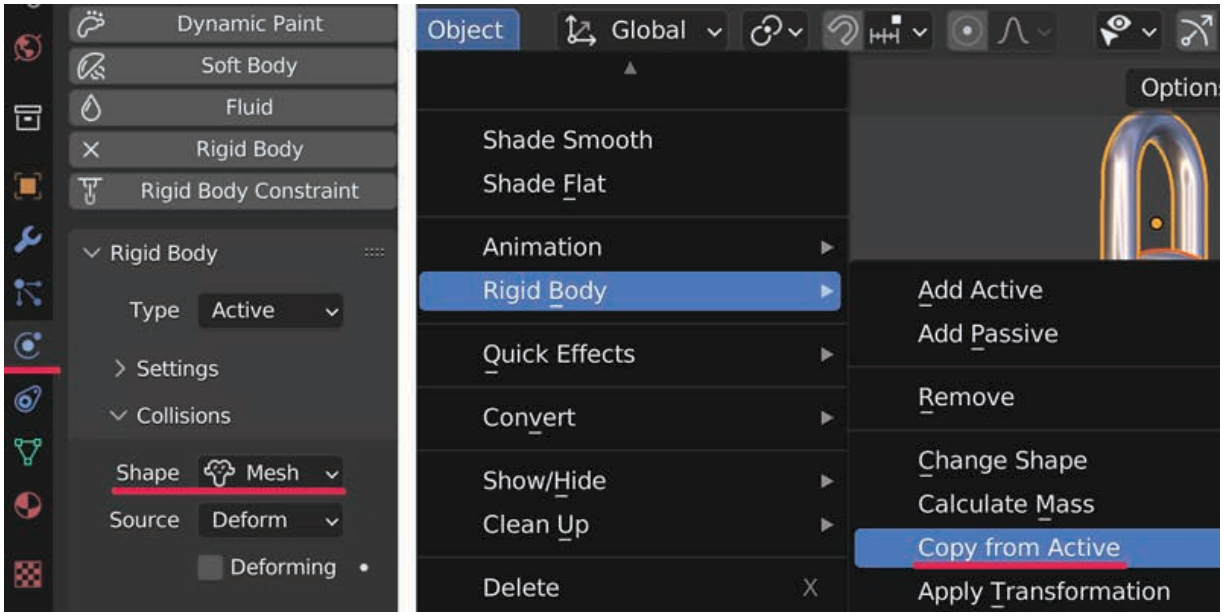
Select both of them and duplicate them and move them up. **Press Shift+R** a few times to create more chain links.



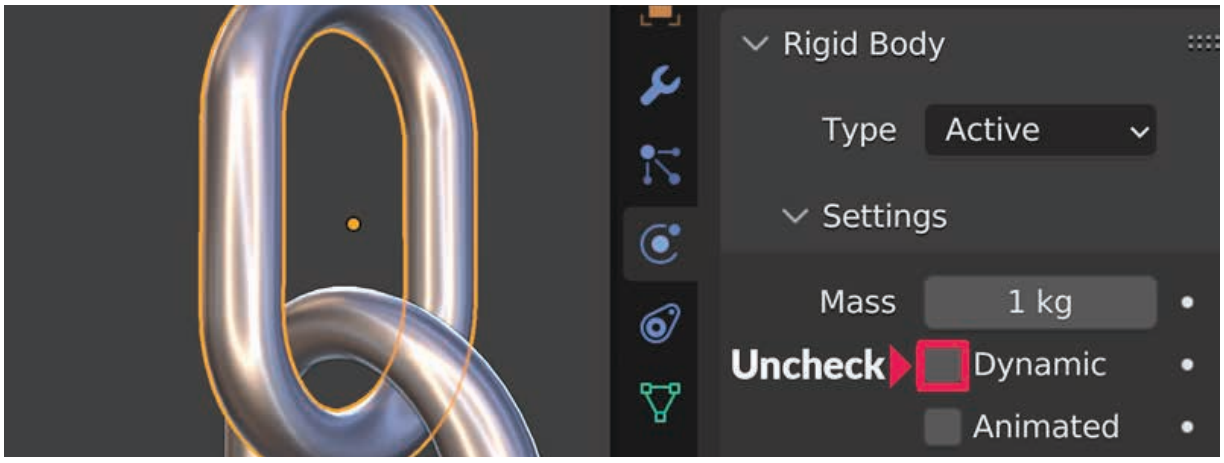
Select all of the chain links. Go to Object> Rigid Body> **Add Active**.



Change the Shape from Convex Hull to Mesh. Go to Object> Rigid Body> **Copy all from Active**.

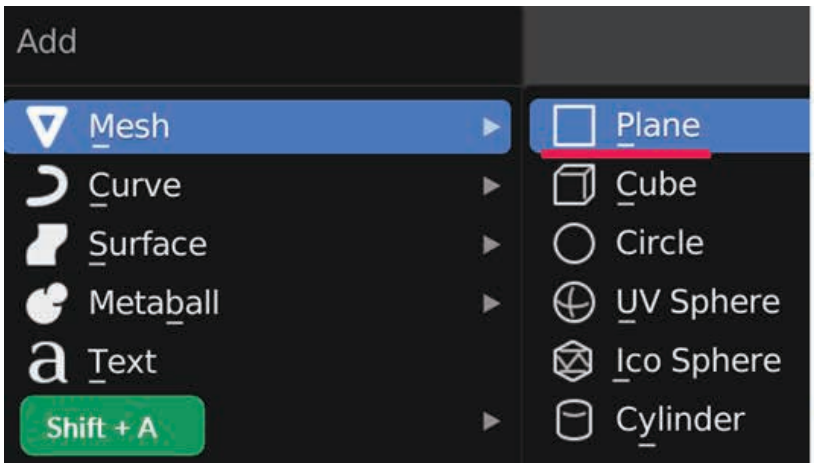


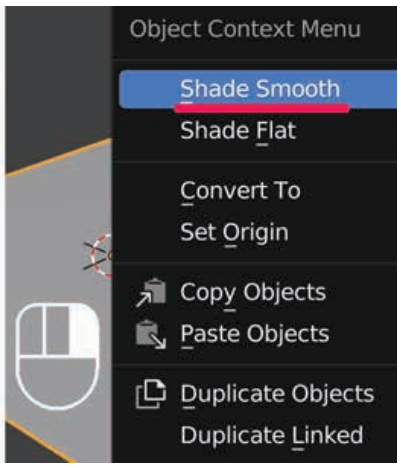
To make one chain be "pinned" (so it doesn't fall down or move), select it and turn off Dynamic. Now **press the Spacebar** to play the simulation.



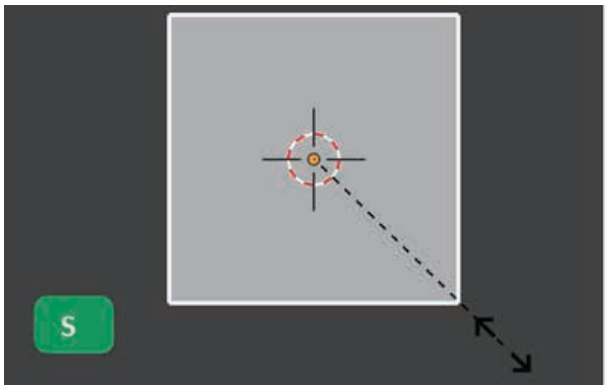
Ocean Modifier basics

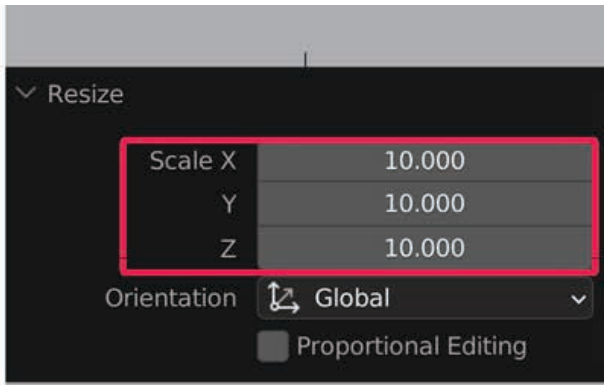
The **Ocean Modifier** in Blender is quite powerful and is based on the Houdini Ocean Toolkit. To get started all you need is a **Plane** with subdivisions and the Ocean modifier. But how do you animate it? Create a plane by pressing Shift+A and choosing Mesh > Plane.



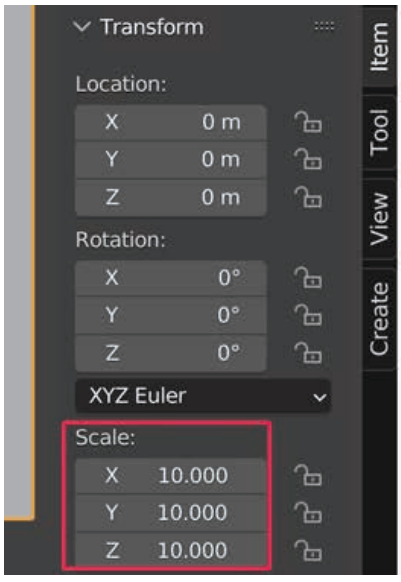


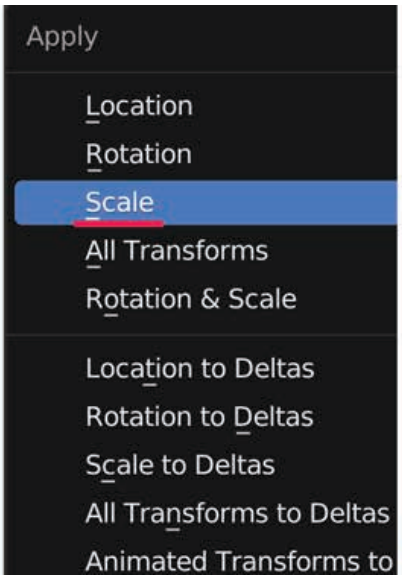
In Object Mode **press S and [10 on the Numpad]** to scale it up to 200 meters. Right-click and choose **Shade Smooth**.

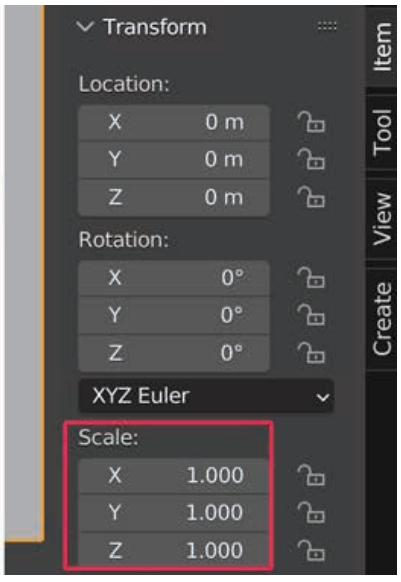




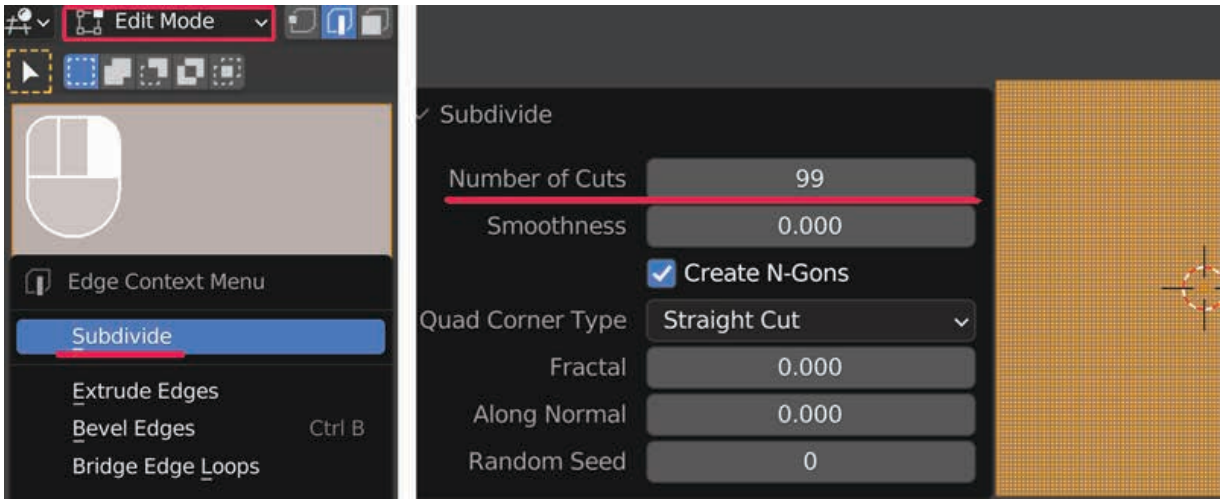
Press Ctrl+A and apply the scale.



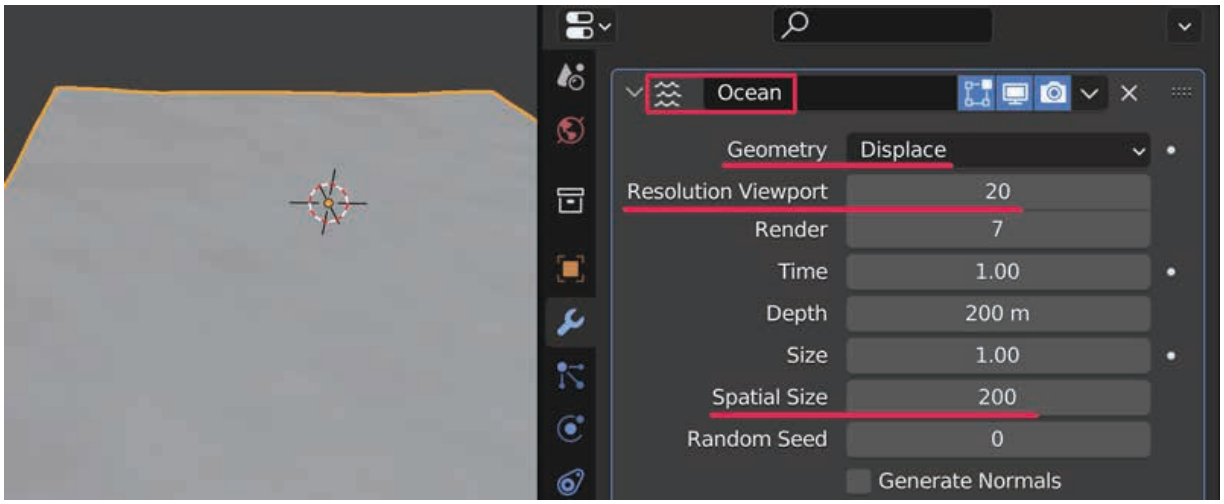




Subdivide it with 99 cuts and add an Ocean modifier. Set Geometry to Displace.



Set **Spatial Size** to 200 to match the size of the plane. Set **Resolution** to 20 or preferably even more. Note that you can set the resolution for the Viewport and Render individually, so you could set the Render Resolution high and Viewport Resolution low.



Be careful with this value: too high, and your computer won't be happy... if you set it too high (like 2000) it will just freeze...

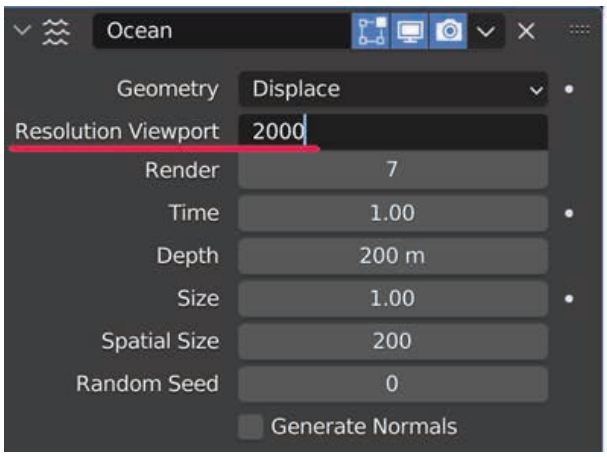
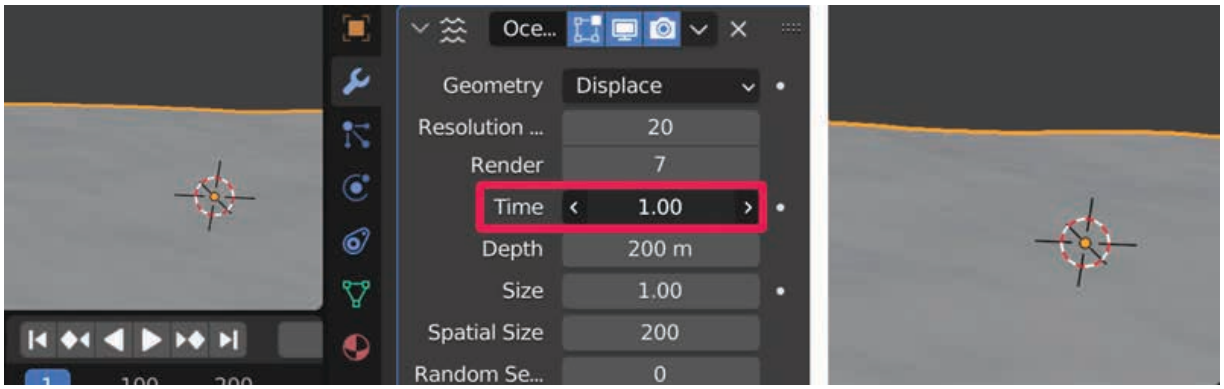
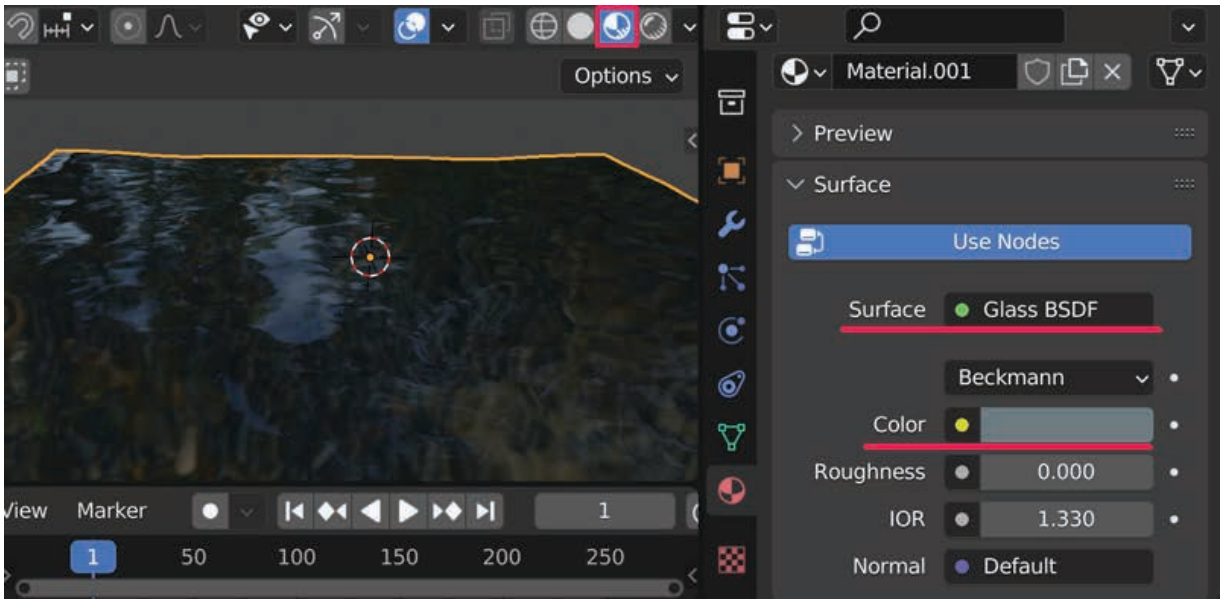


Illustration by Sarah Klockars-Clauser CC-BY-SA

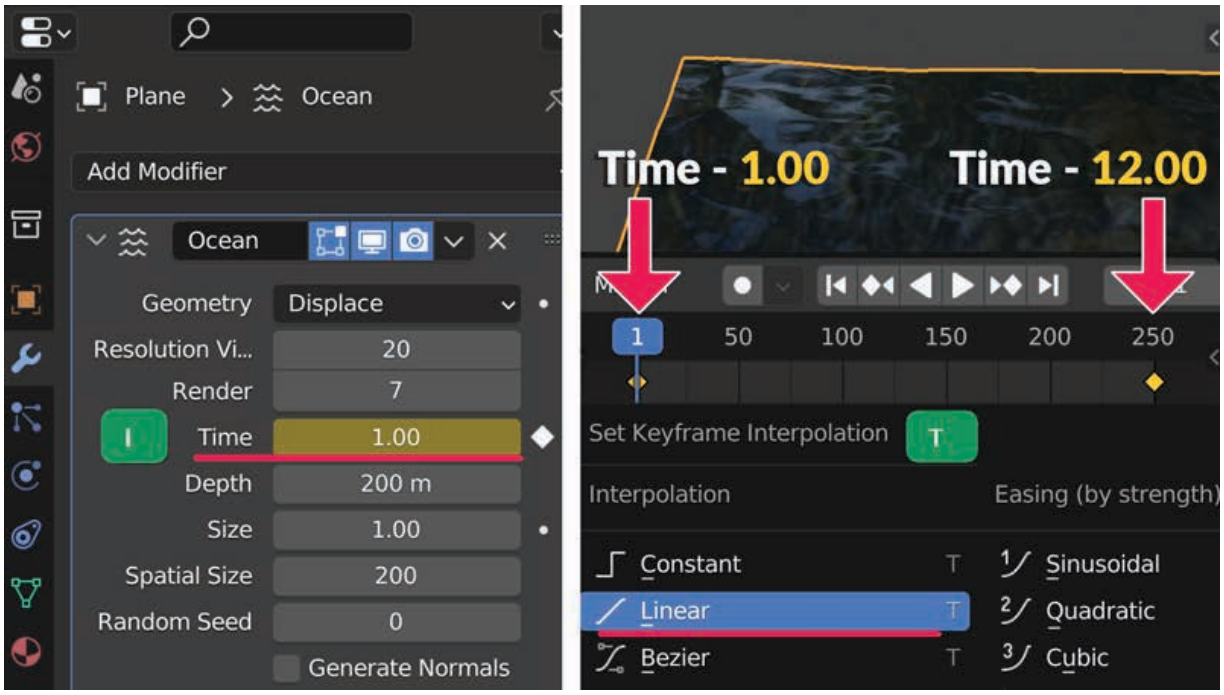
If you drag the **Time** value you'll notice that the ocean moves.



Actually it's hard to notice, so go to Material view and add a dark reflective material.

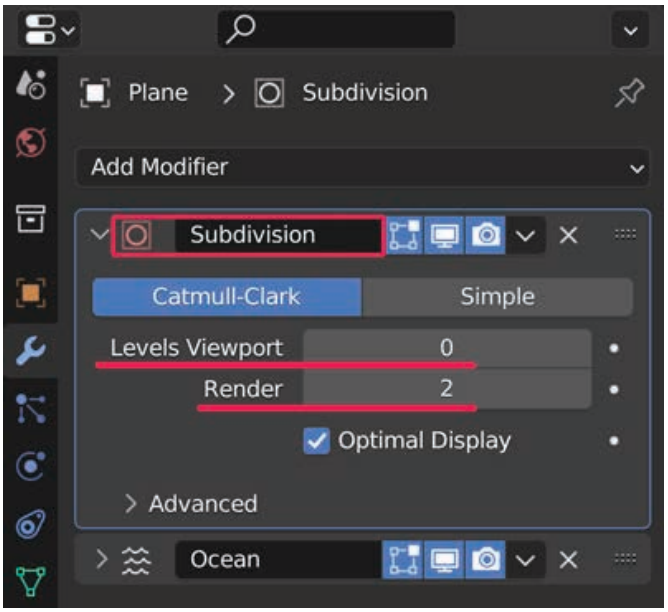


Set keyframes for the Time value if you want to animate the waves but be sure to set them to Linear.

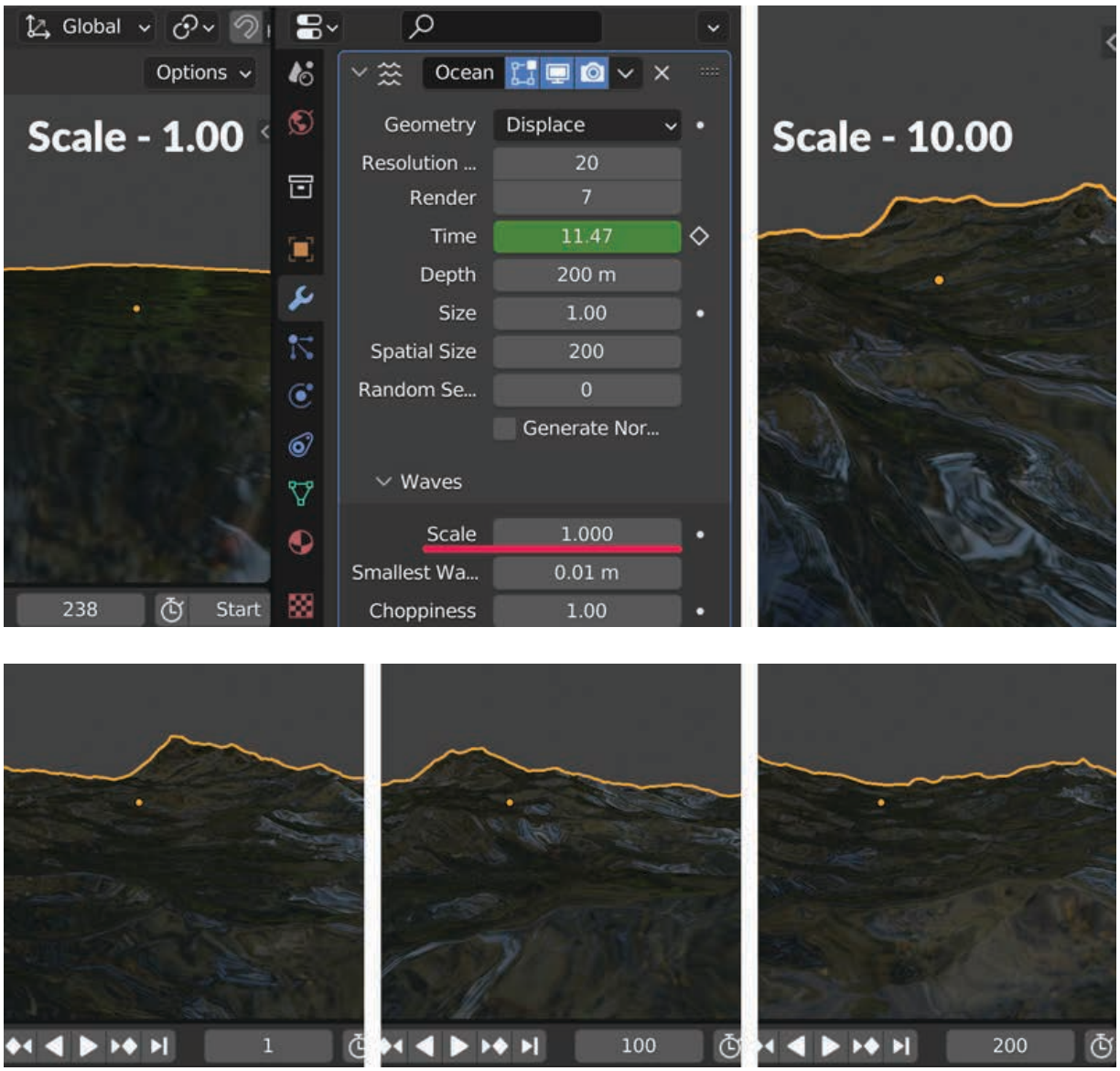


You can add a Subdiv modifier before the Ocean modifier to get a bit more detail from the Resolution that you set.

Like the Ocean modifier resolution, you can set this low for the viewport and high for the render.



Scale controls how high the waves are.

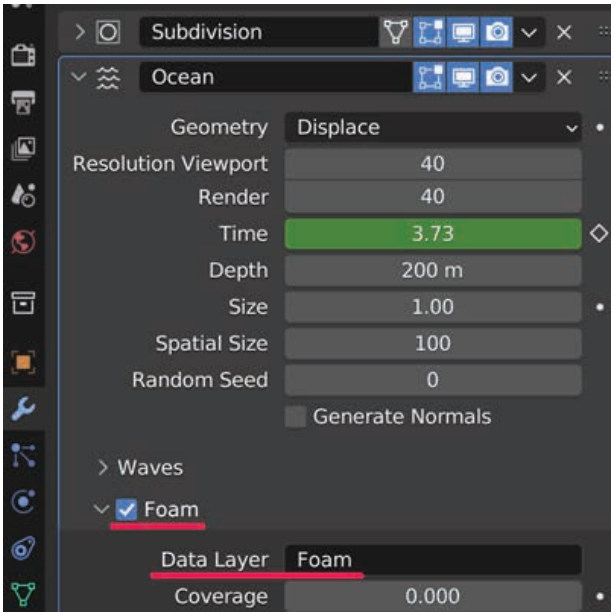


Ocean shader with Foam

Previously we had a look at creating an animated ocean using the Ocean Modifier. To create a material with foam is actually pretty easy!

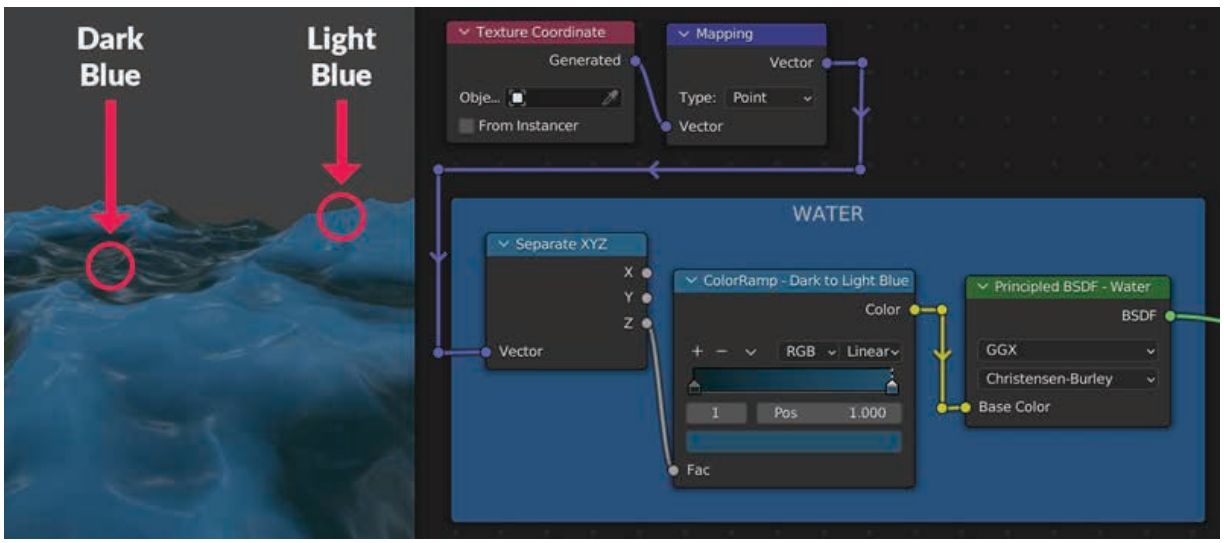
To create an ocean shader with foam is actually pretty easy!

In the Ocean modifier, check the Generate Foam box and name the Foam Data Layer “Foam”.



For the higher waves to have a lighter color you need a ColorRamp node. Add a dark blue and a light blue color to the Color Ramp and plug it into the Base Color of the Principled BSDF.

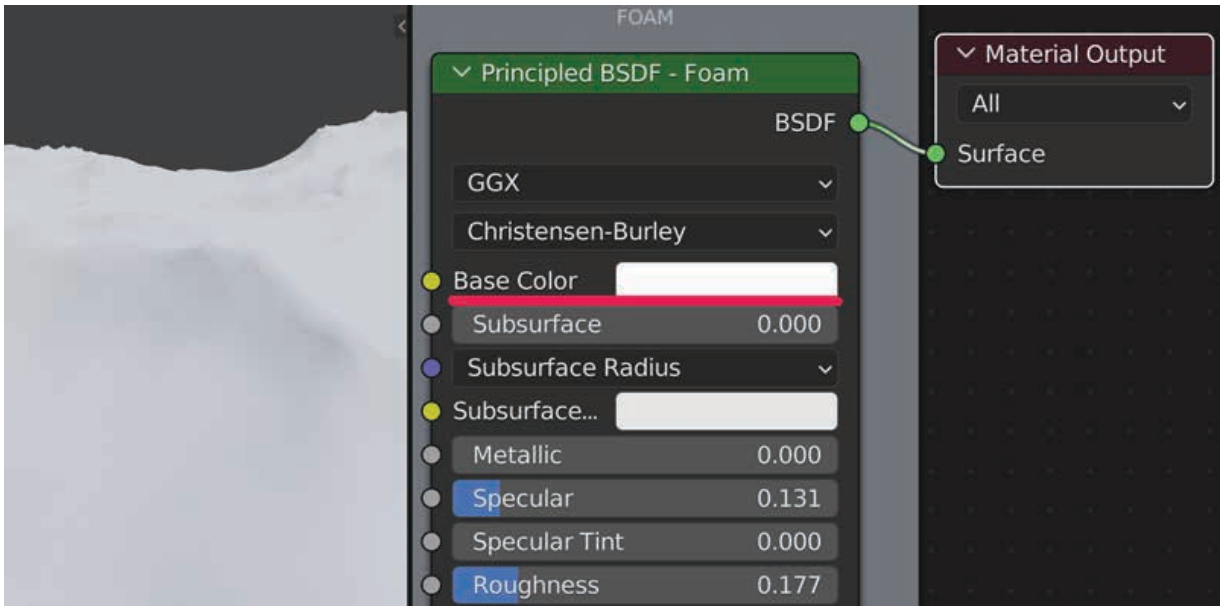
Then use a Separate XYZ node and plug the Z output into the ColorRamp. Now the waves show the light blue color for the peaks and the dark blue color for the valleys.



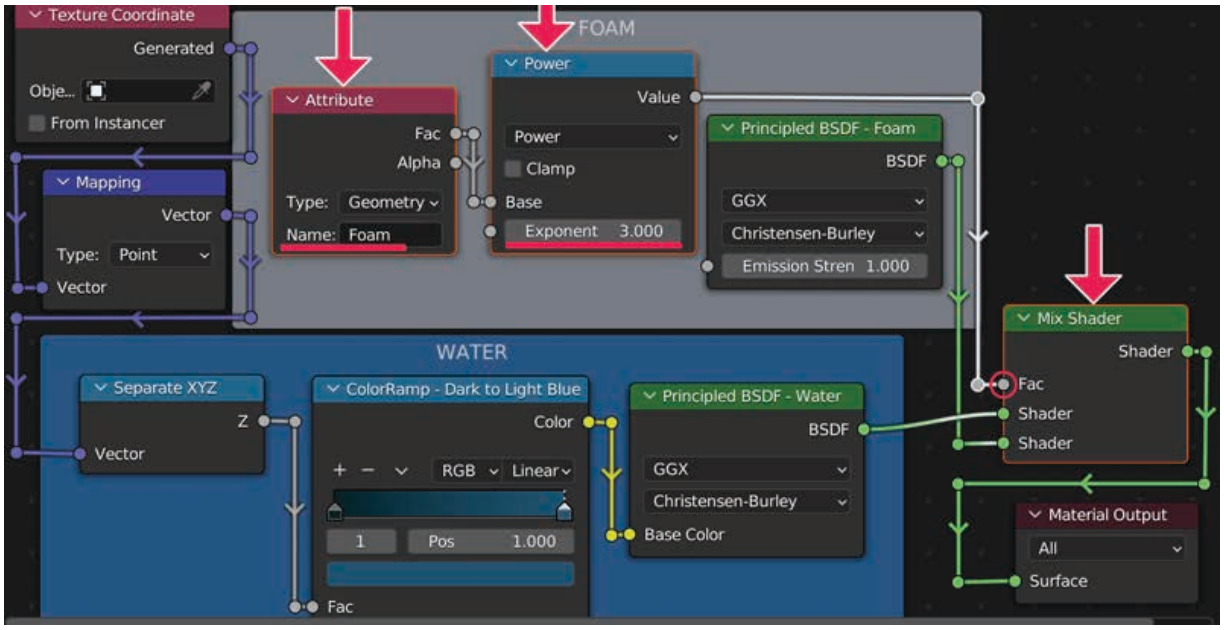
Click here to return to the index.

You can experiment with the Roughness value to get the right amount of glossiness.

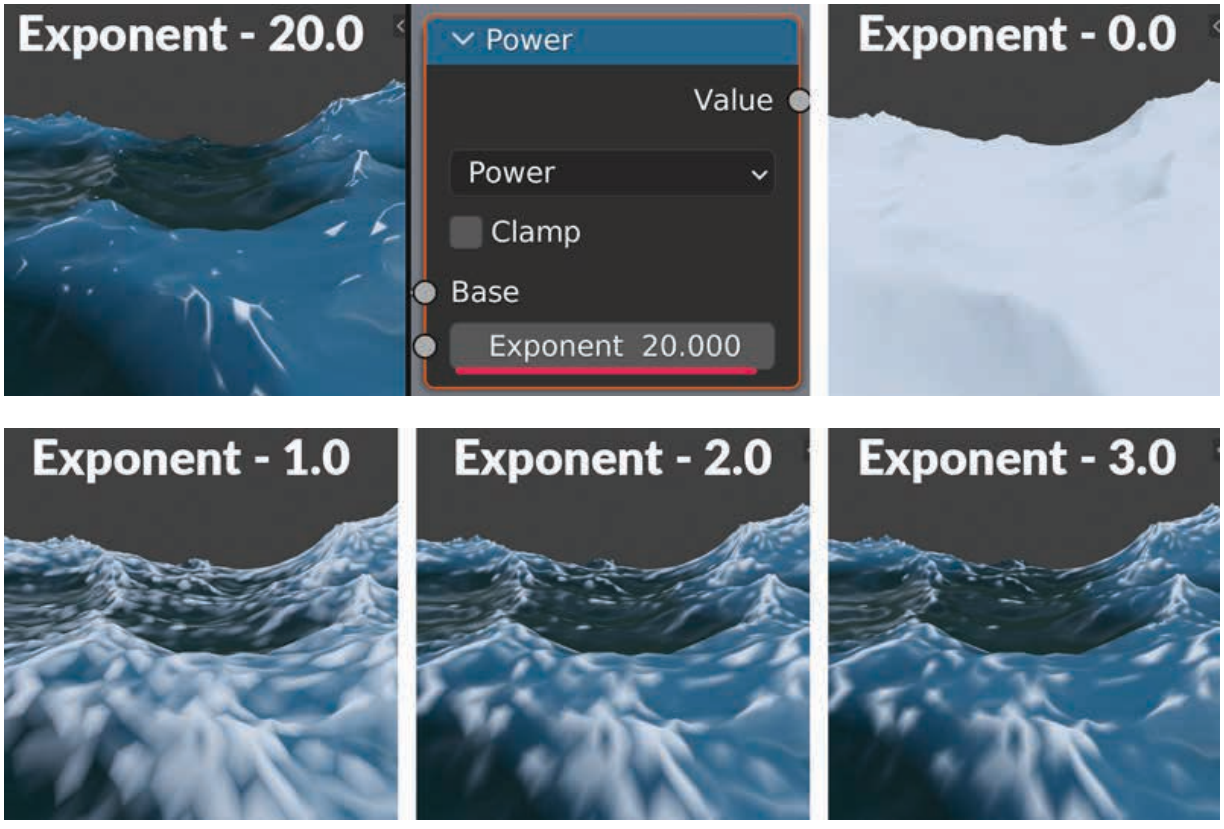
To create Foam, add another Principled BSDF and set the Base Color to white.



Add an Attribute node and type Foam in the Name field. Plug it into a Math node and set that one to Power. Use the output of the Math node for the Mix shader.



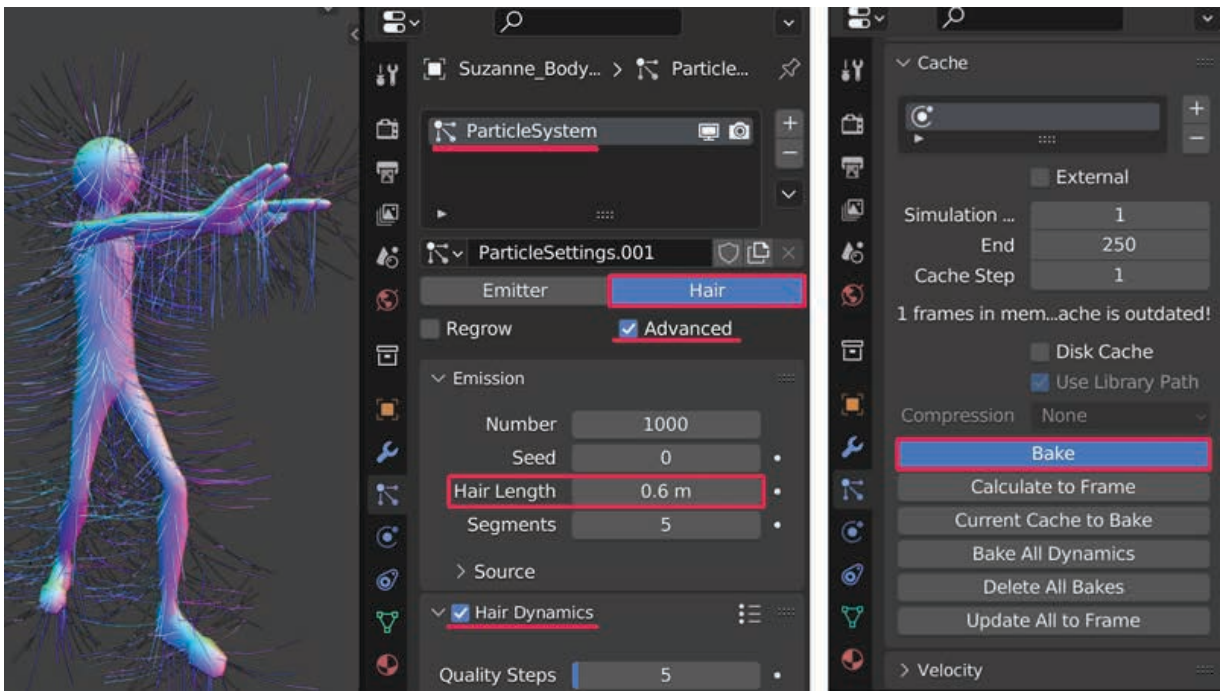
Experiment with the Exponent Value to get a result you like.



Make a Dancing Hair Monster with Hair Dynamics

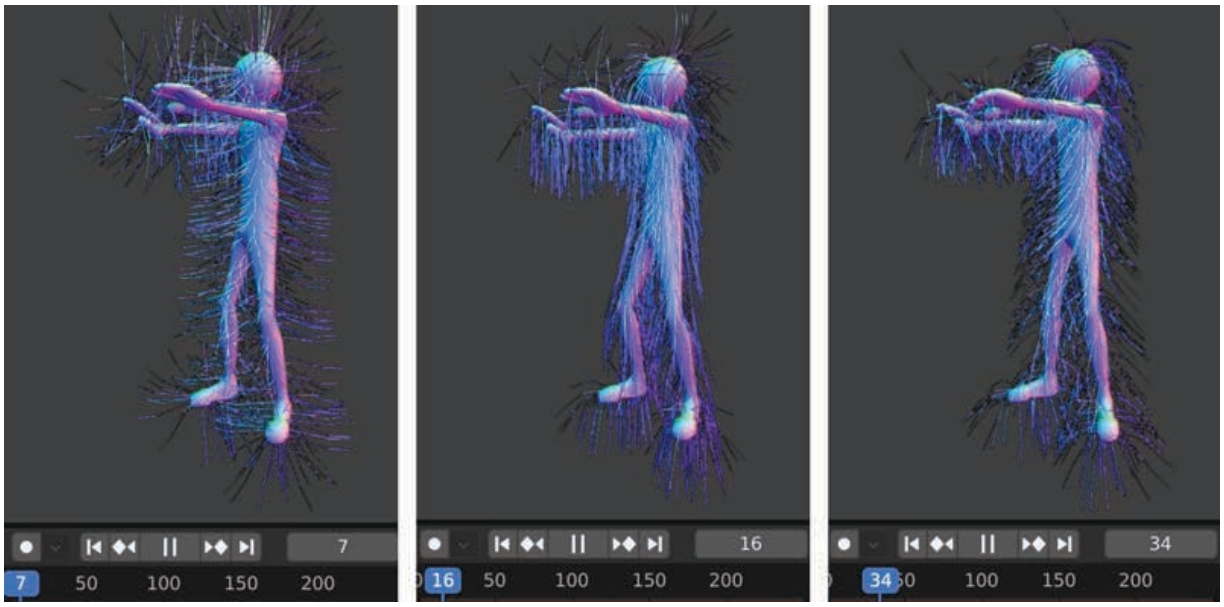
Make a dancing hair monster with Hair Dynamics! This video builds upon the topics where we looked at uploading your character to Mixamo as well as rigging and downloading it again with animation.

Previously we've looked at using mixamo to upload a character to rig it and give it some animation. With that animated character imported, add a particle system and set it to hair, advanced. Scale down the hair length and turn on hair dynamics. Now if you press bake it will calculate the simulation.

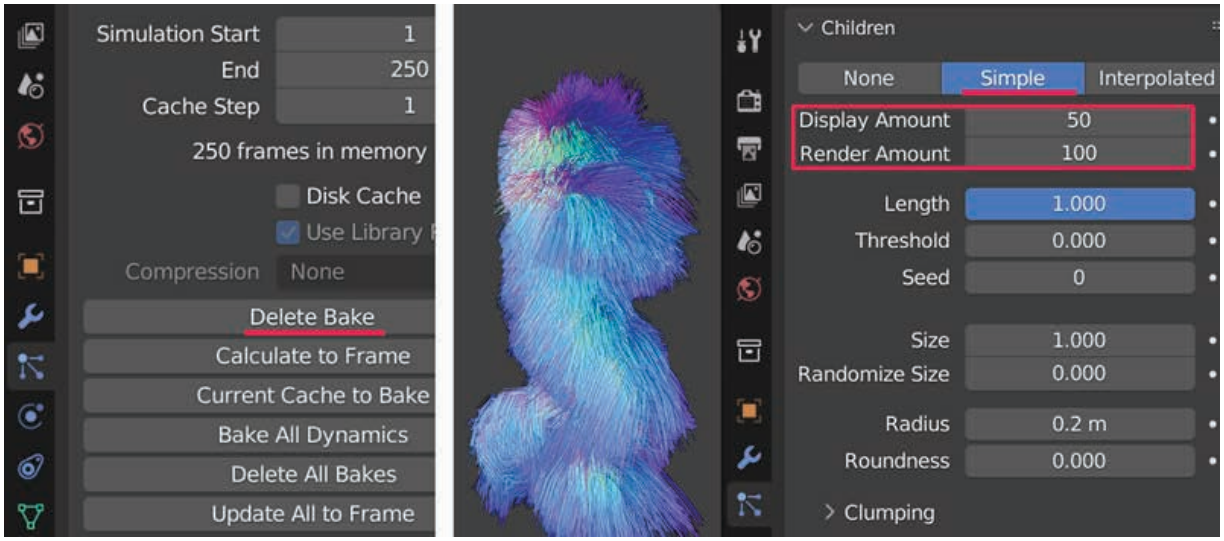


Click here to return to the index.

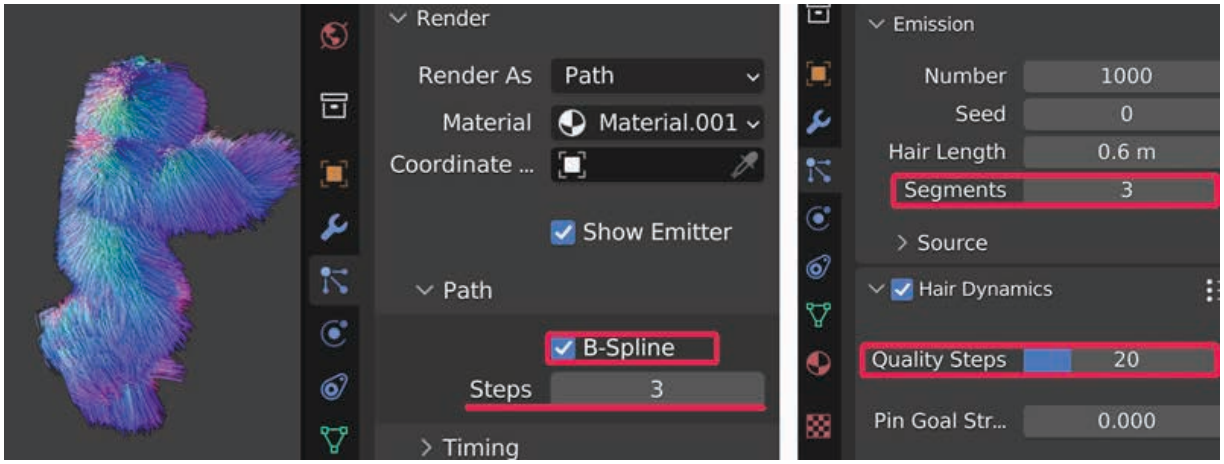
Press the Spacebar to view the simulation.



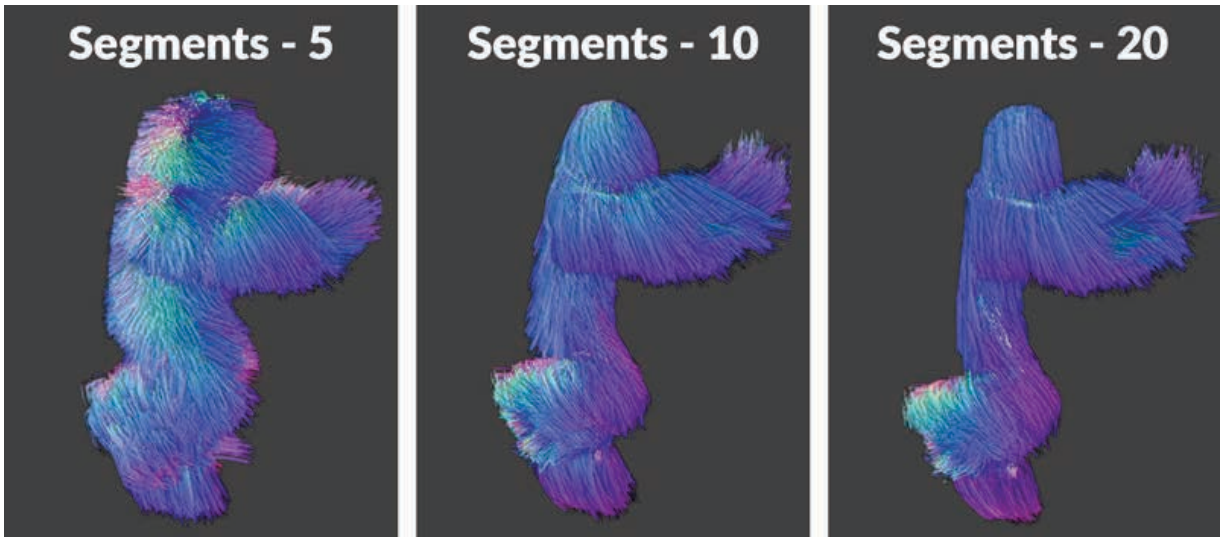
Delete the bake if you want to make adjustments. Go to children and turn on simple. Increase the amount. You can set the render amount higher so that you get more hair in the rendered animation.



Of course this also affects your render times so be careful. Under Render > Path you can turn on and increase the B-Spline number.

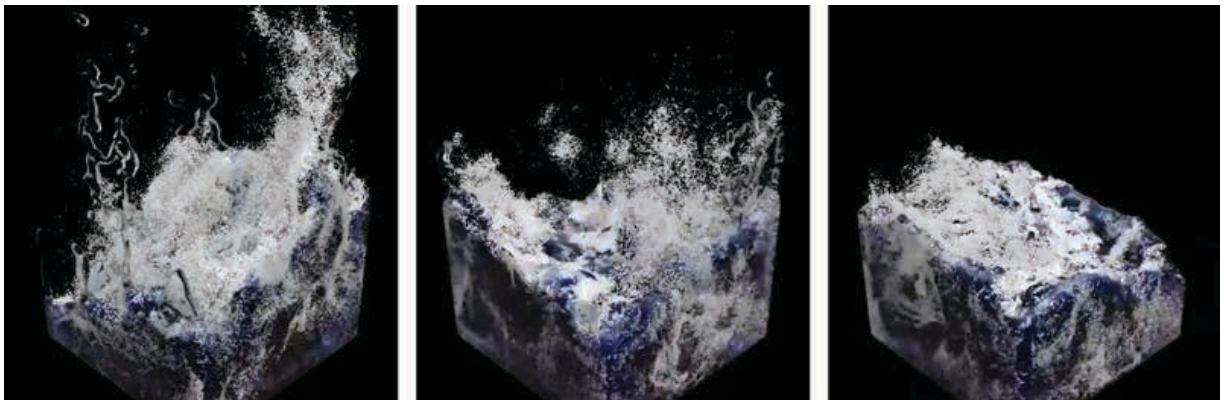


This makes the hair smoother. Increasing the segments makes the hair more flexible. Increasing quality steps increases the quality of the simulation.

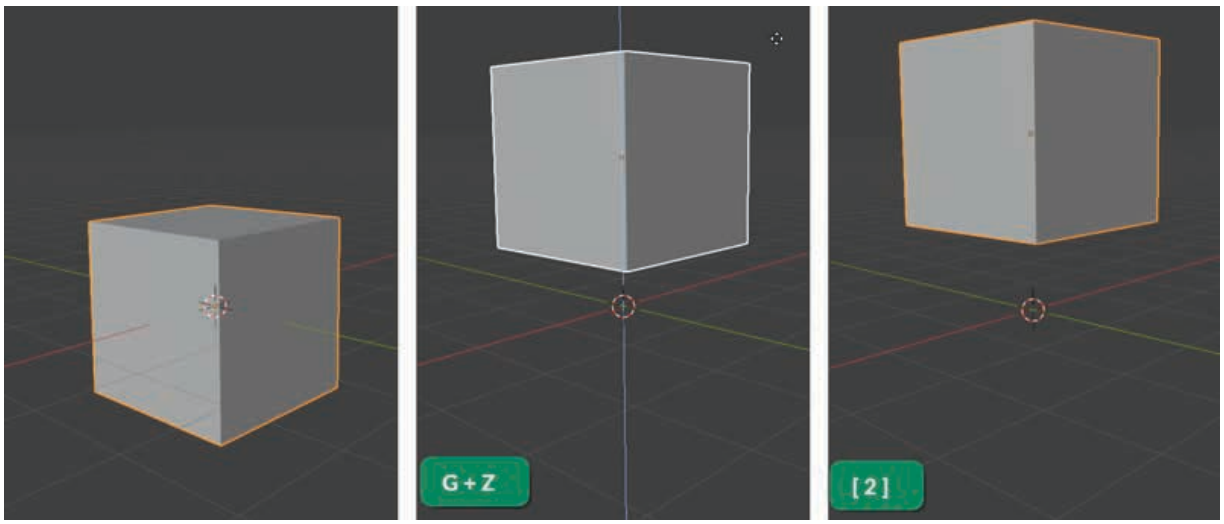


Quick Liquid with Spray, Foam and Bubbles

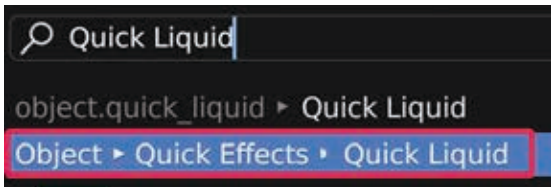
Thanks to the new Mantaflow system, liquid has never looked this good in Blender! Although there are a confusingly large amount of options, you can actually get started rather easily with the "Quick Liquid" which sets a lot of things up for you automatically. This short video explains.



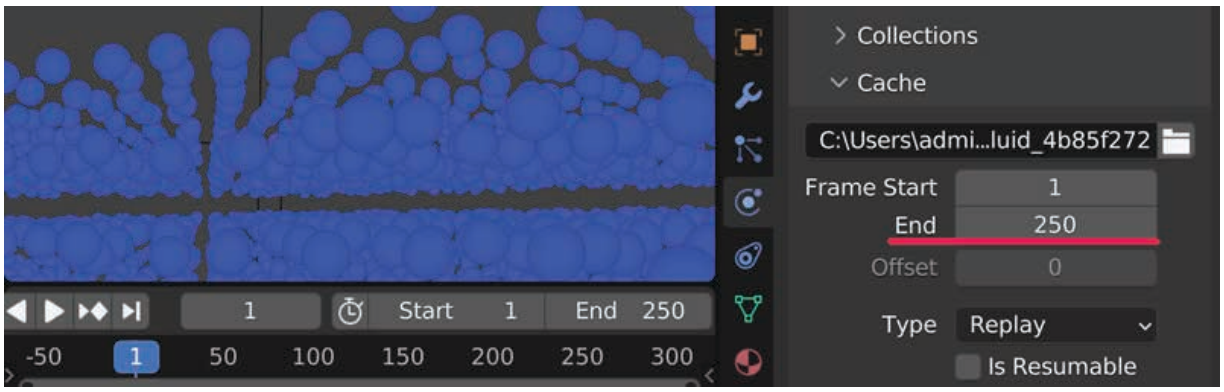
Select the default cube and move it up along the Z-axis by **pressing G+Z+[2 on the Numpad]**.



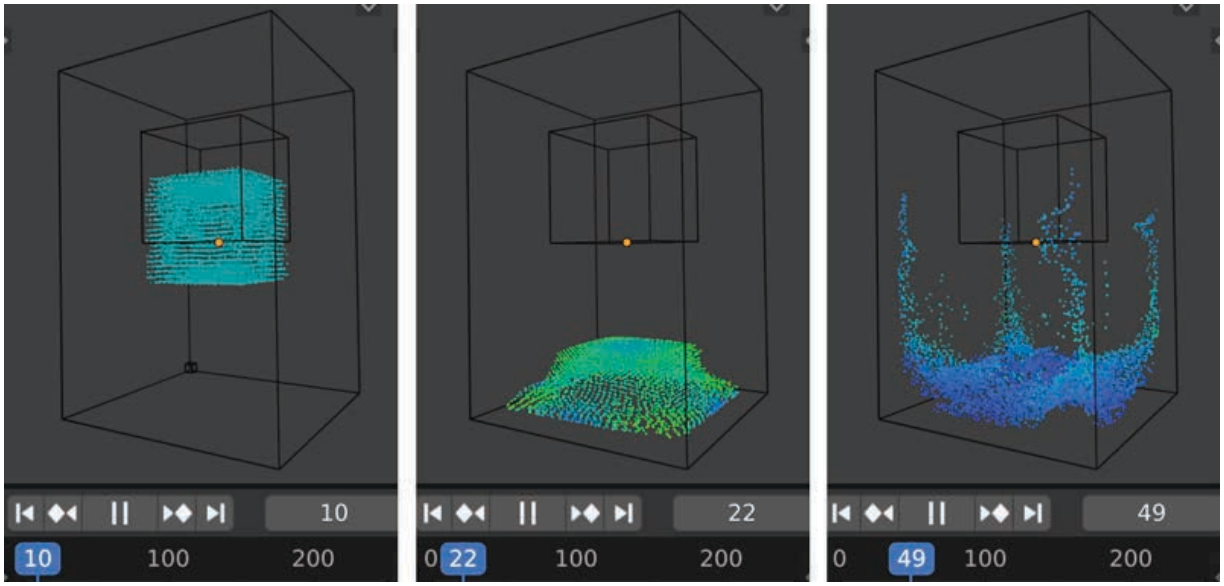
Press F3, type “Quick Liquid”. Choose **Quick Liquid**.



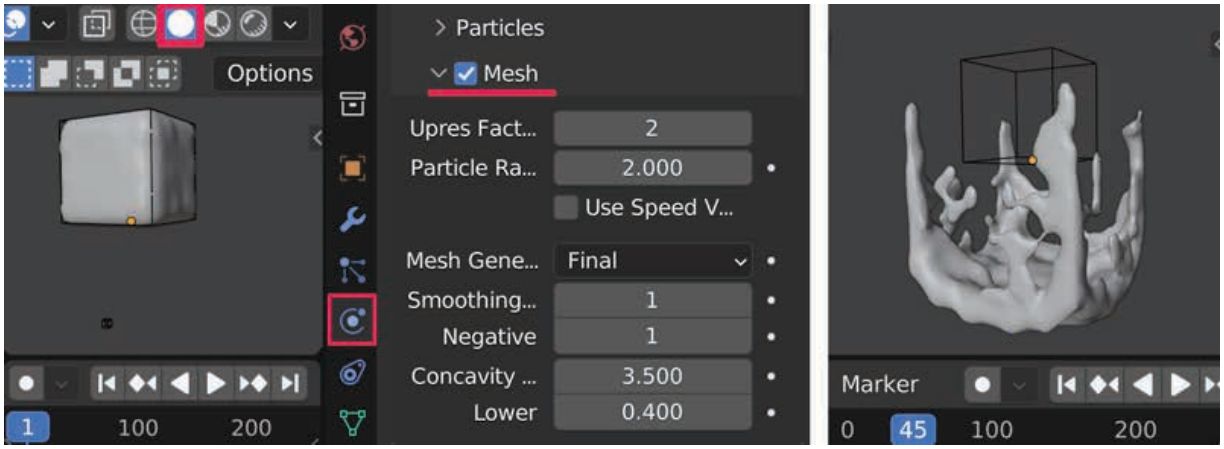
Under the **Cache** settings increase the **End** value so it doesn’t stop too abruptly.



Play the simulation by pressing the Spacebar.

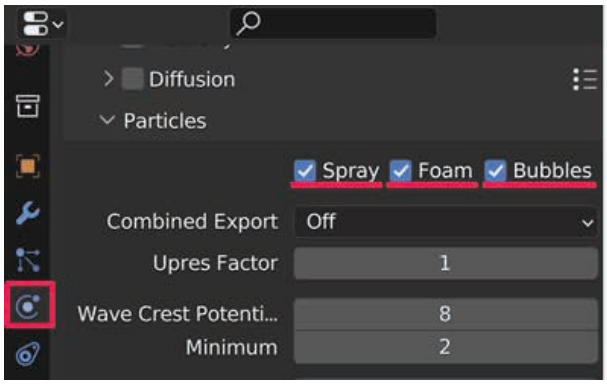


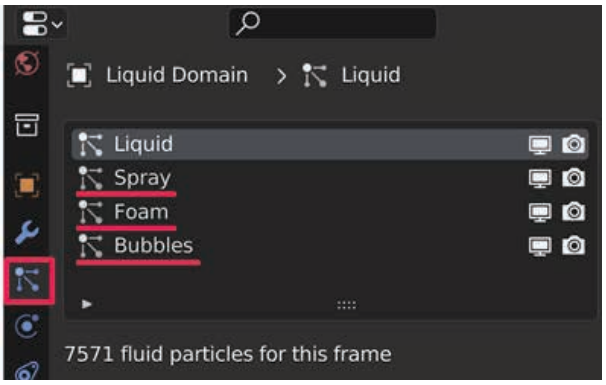
Press Shift and **left arrow key** to go back to the first frame. Check the “Mesh” box to turn the liquid into a mesh instead of little spheres.



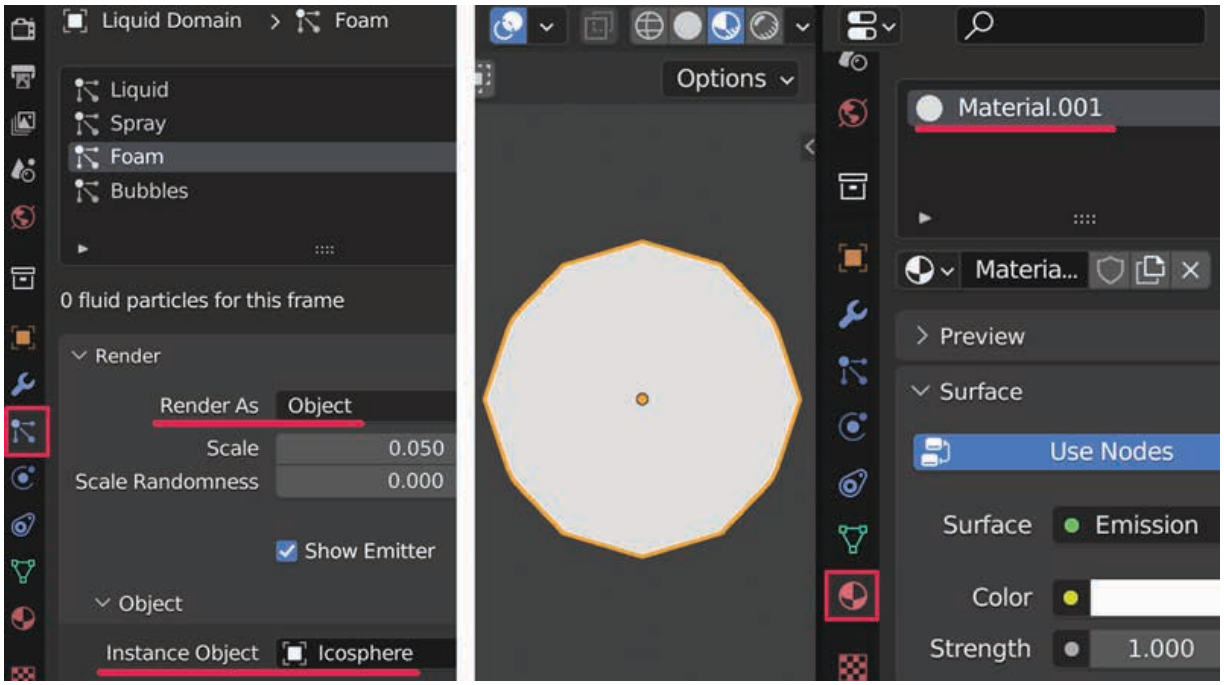
Click here to return to the index.

Under Particles turn on **Spray, Foam and Bubbles**. This adds Particle Systems for each of those elements.

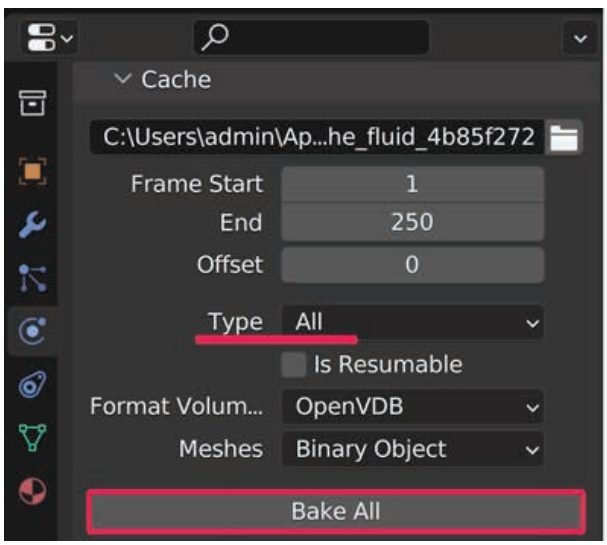


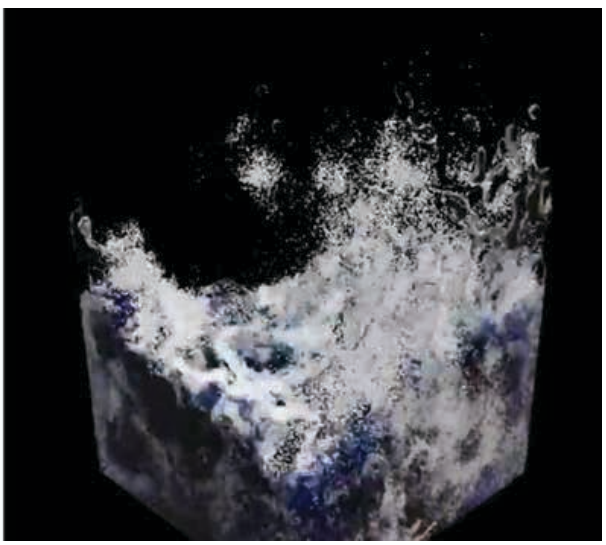


By default they are set to render as Halo but you can create objects like **Icospheres** and give those a material and then set the particle system to render as those objects. That way you can control the material of the Spray, Foam and Bubbles.



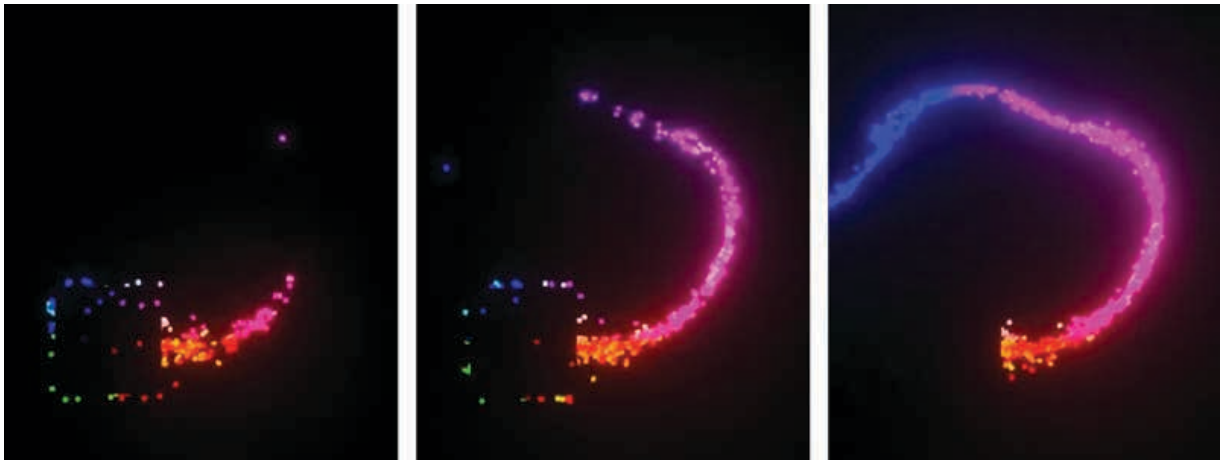
To bake the Liquid simulation you need to set Cache Type to Final. Then click Bake All.



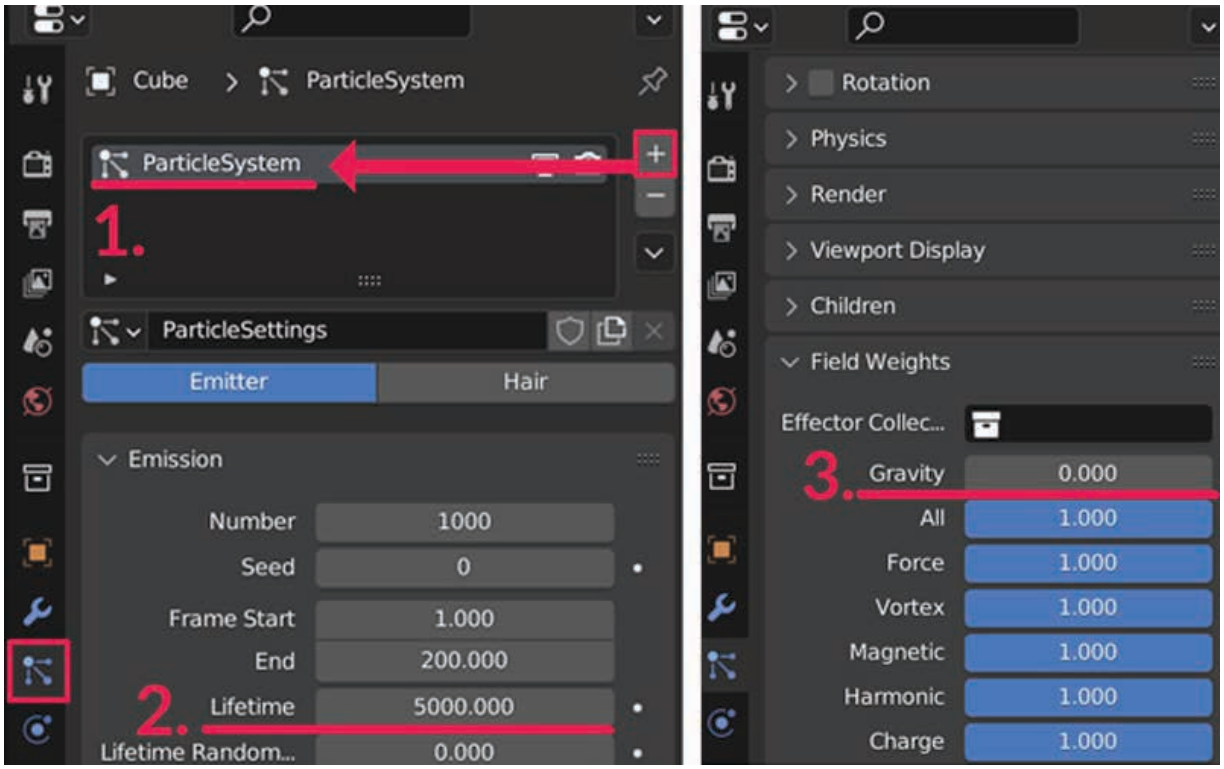


Particles along a curve with random colors in Eevee

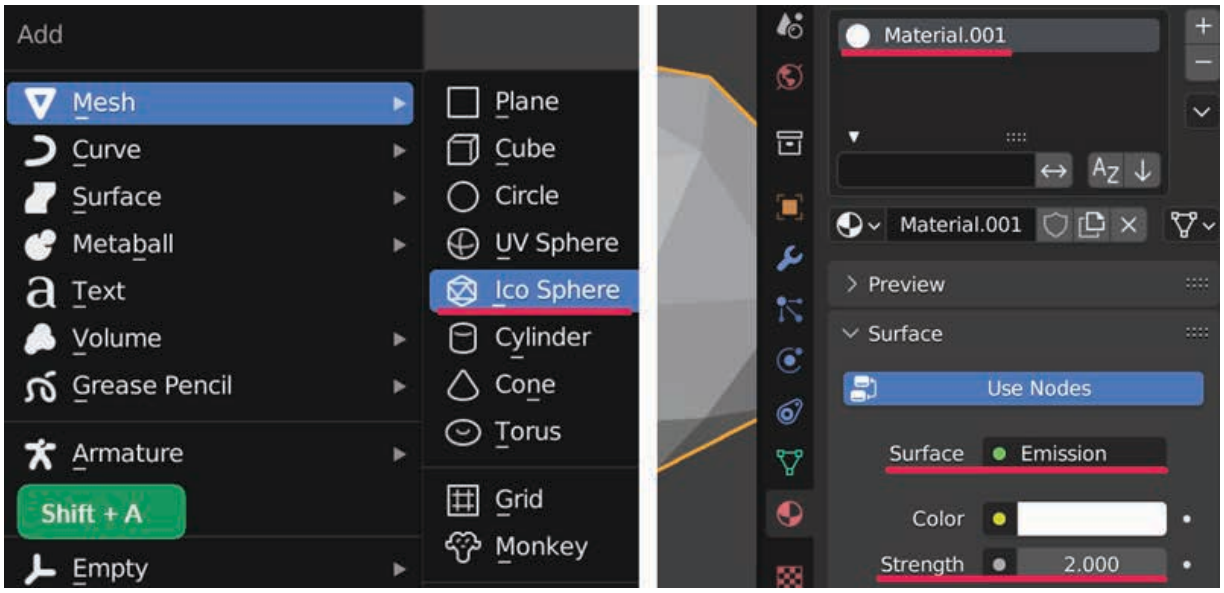
Did you know that you can make particles follow a Path or Curve? All you need is an Emitter object (the default cube will do) as well as a **Curve**. Then we can set up an **Ico Sphere** with random colors as the render object for the Particle System.



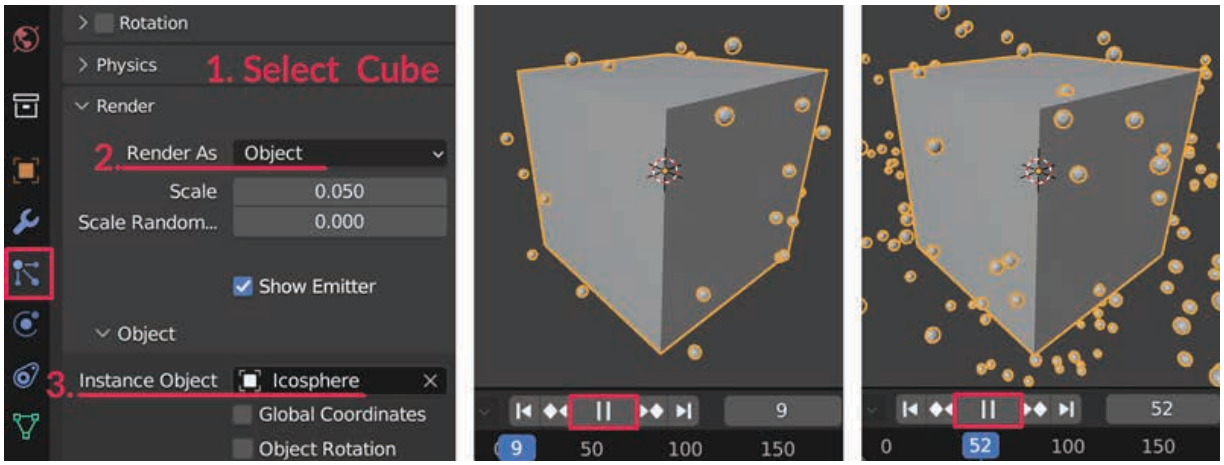
Select the default cube. Create the Particle system by pressing the + button. Increase the lifetime. That way the particles don't disappear so quickly. Set Gravity to zero so that the particles float weightlessly.



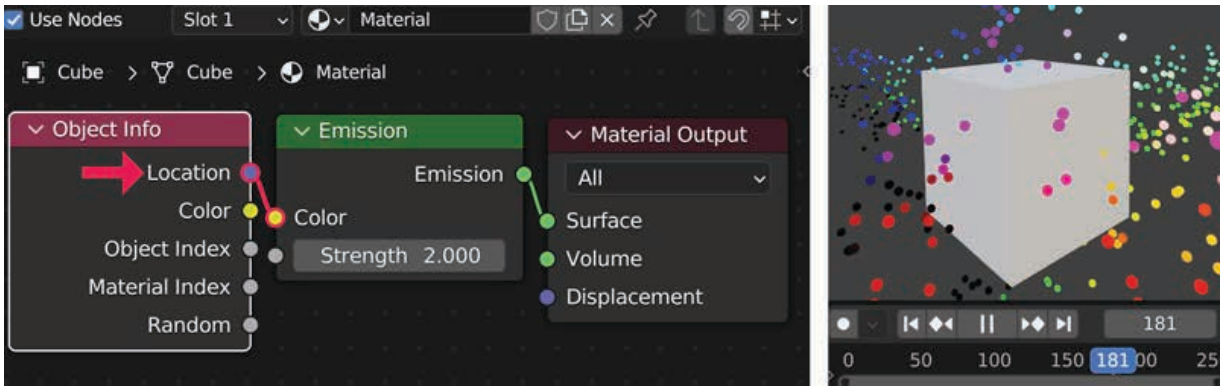
Press Shift+A and create an **Ico Sphere**. Give it an **Emission** material. Use the Ico Sphere as the Render object for the particles system.



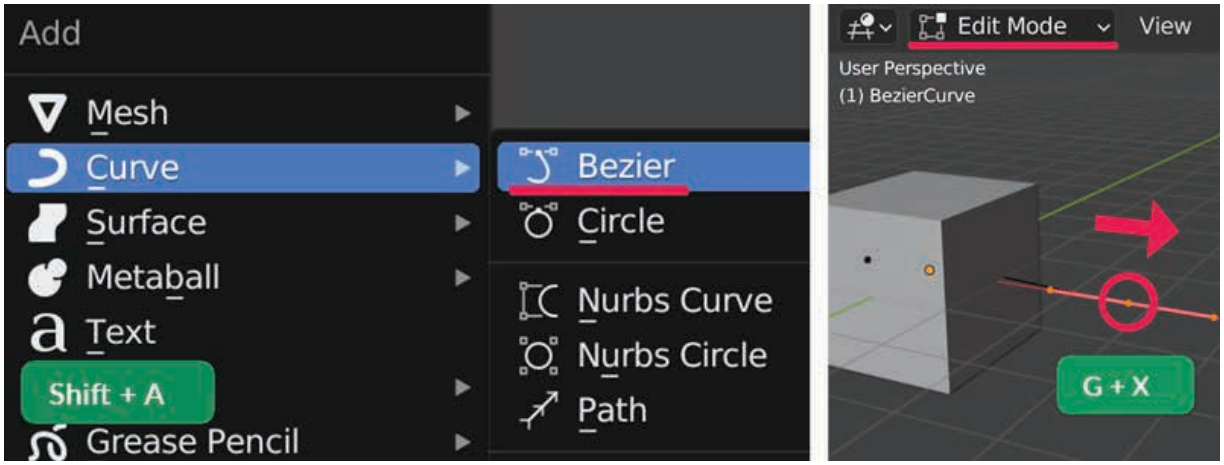
Press Shift+A and create an Icosphere. Give it an Emission material. Use the Icosphere as the **Instance Object** for the particles system.



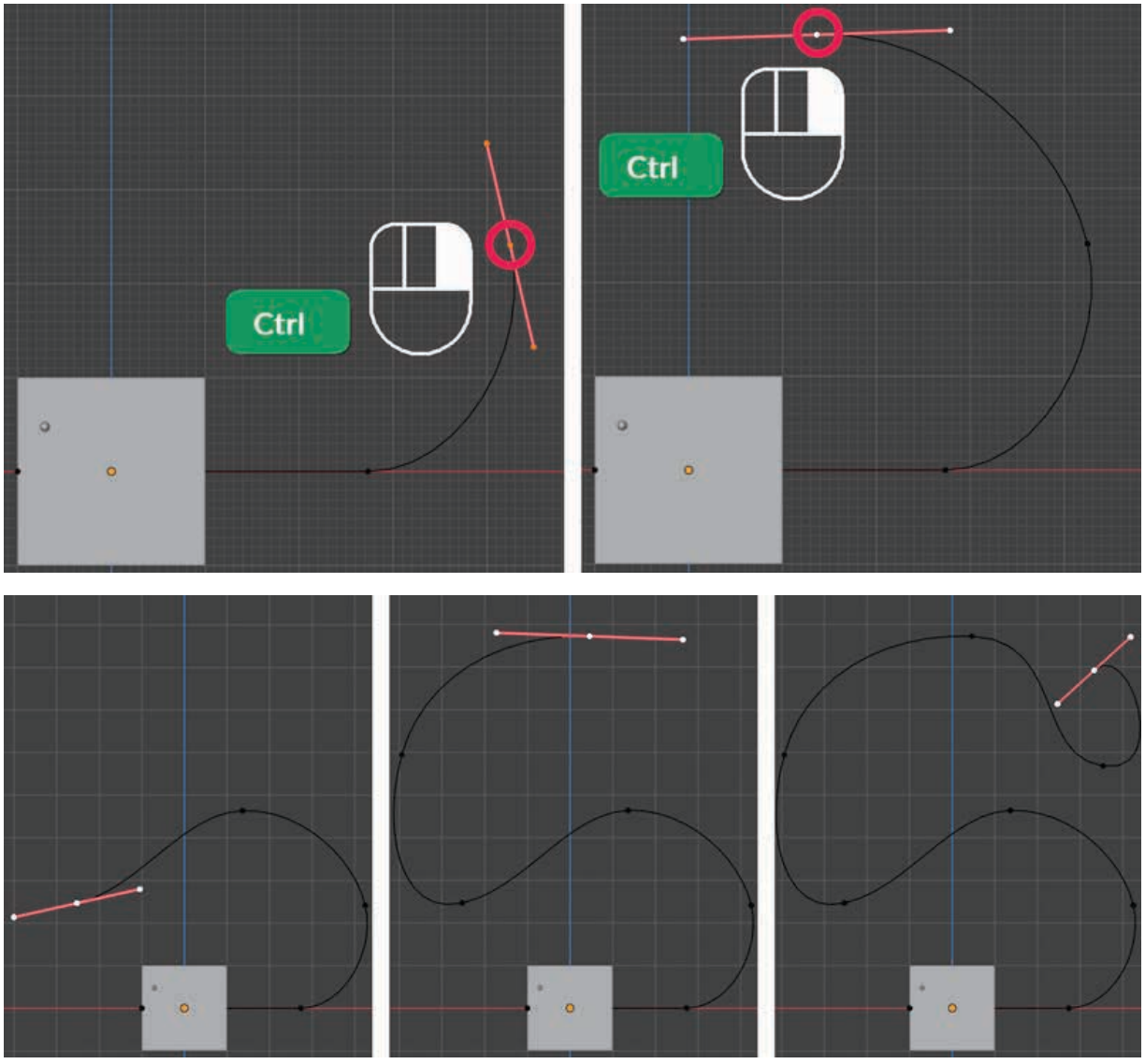
In the **Shader Editor** plug an **Object Info node** into the Emission Color input.



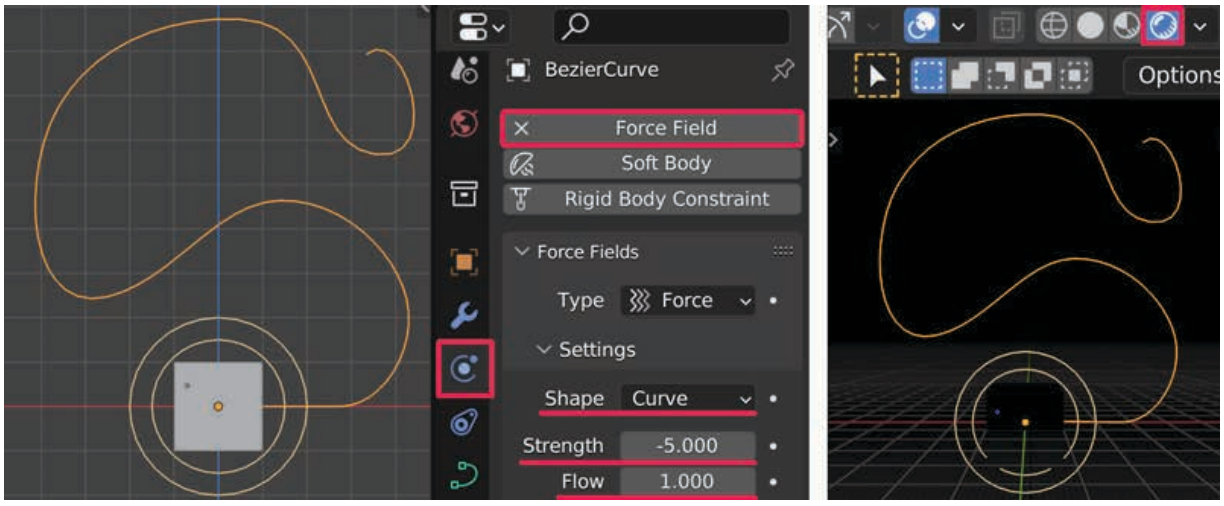
Press Shift+A and choose **Curve > Bezier**. In Edit Mode move one end of the Curve near the icosphere and create an interesting trajectory for the particles.



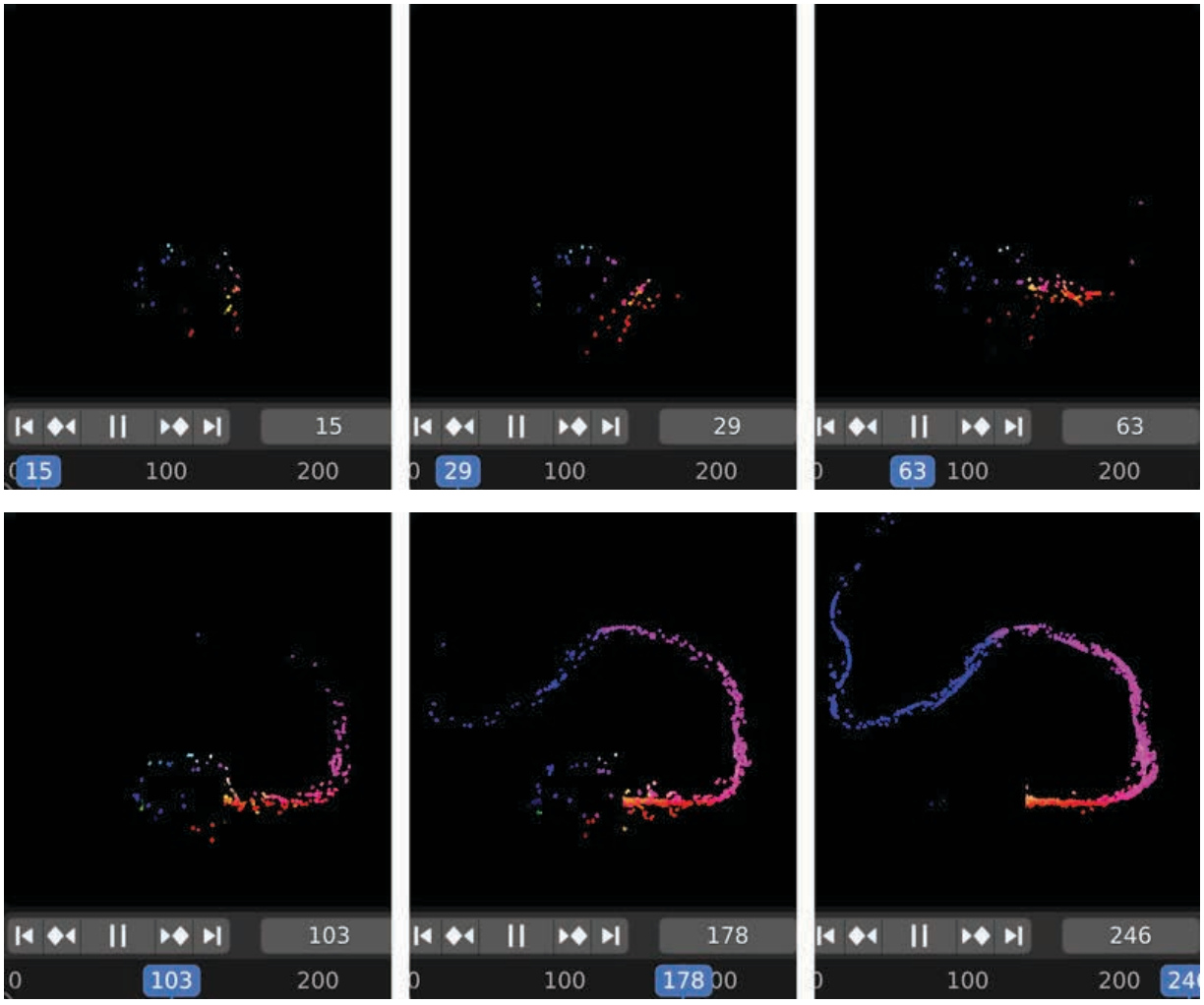
To quickly extend the curve select the vertex at one end of the Curve and **press Ctrl+RMB** to extrude to where you click in the viewport.



Now in Object Mode add a Force Field to the curve in the Physics tab. Set the **Shape** to Curve. Set the **Strength** to -5. Set **Flow** to 1.

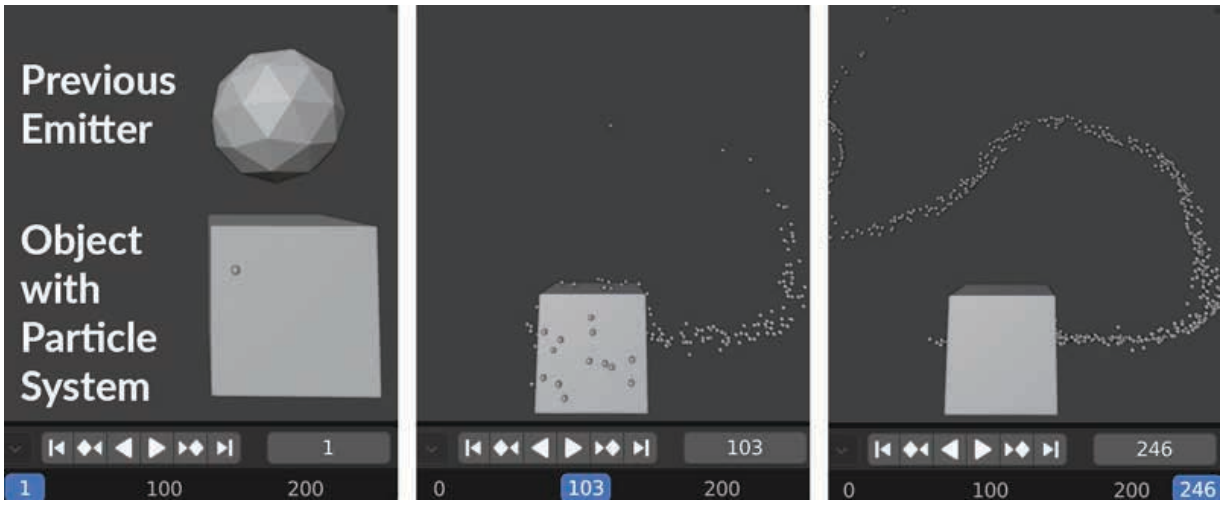


Press the Spacebar to play the simulation.

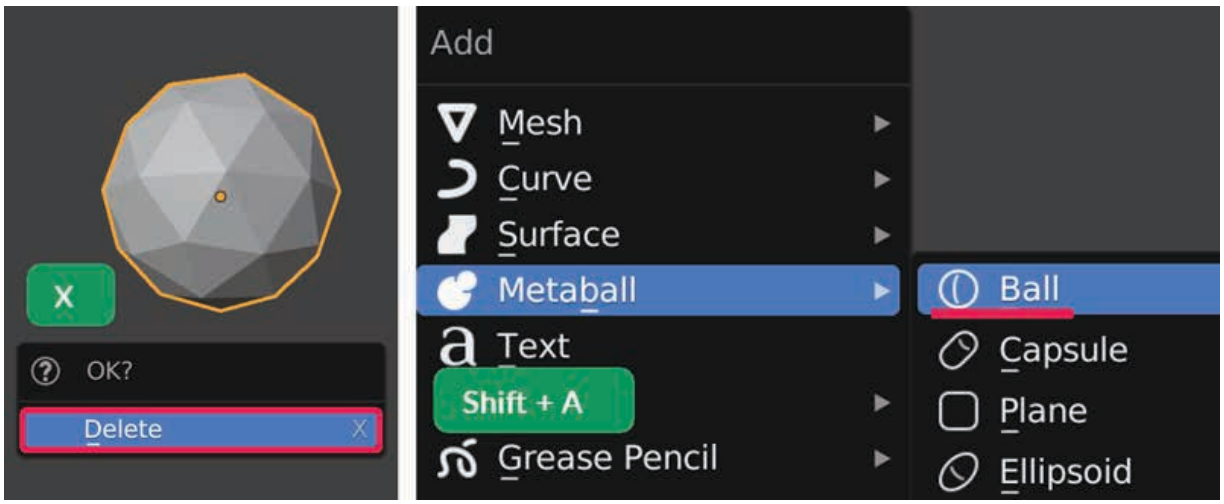


Liquid stream following a curve

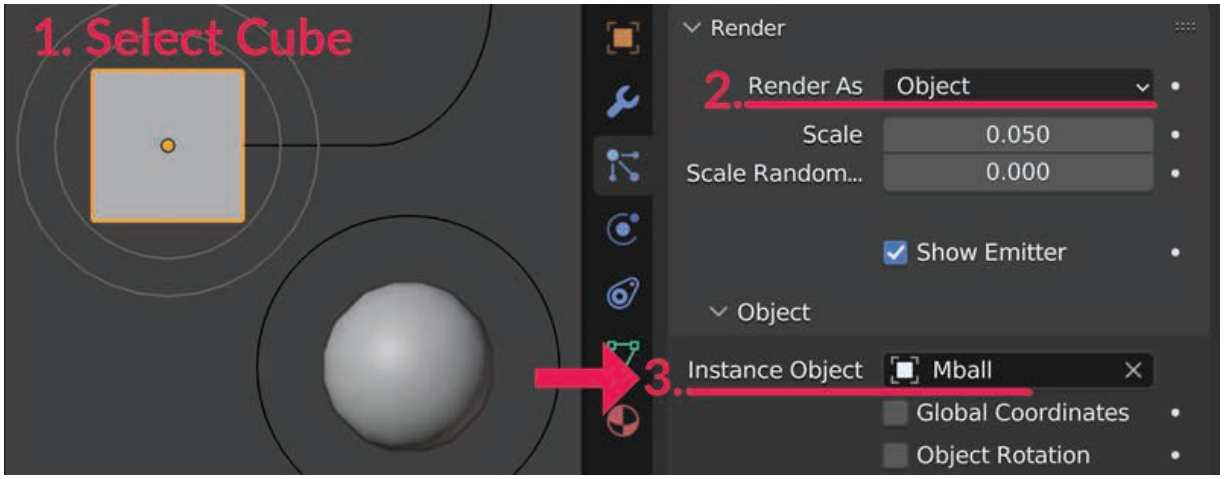
To create a liquid stream, first set up a particle system following a Curve as explained in the previous topic.



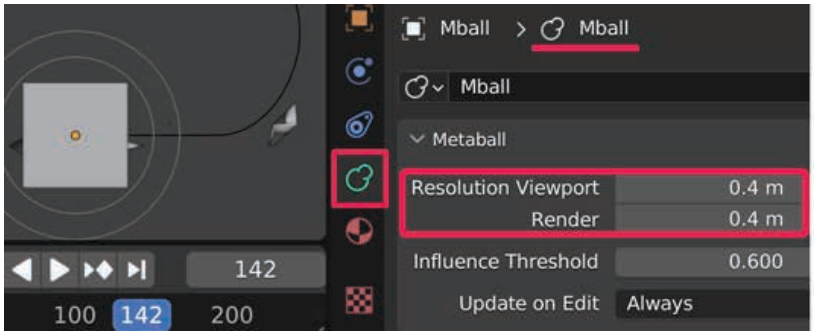
Then, remove the Ico Sphere and create a **Metaball** instead by **pressing Shift+A** and choosing Metaball > **Ball**.

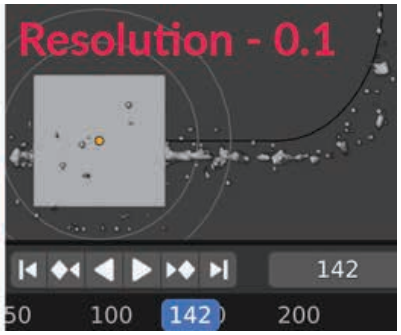


Use the Metaball as the **Instance Object** for the particle system.

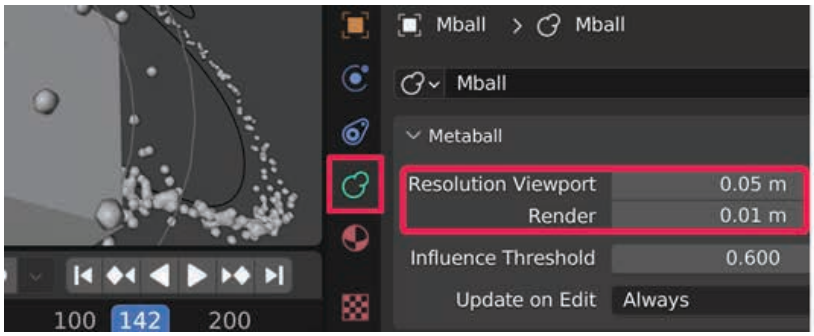


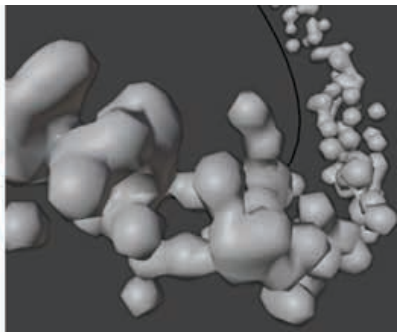
The Metaball resolution can really slow down your system, so don't set it too low. Lower values mean higher resolution.



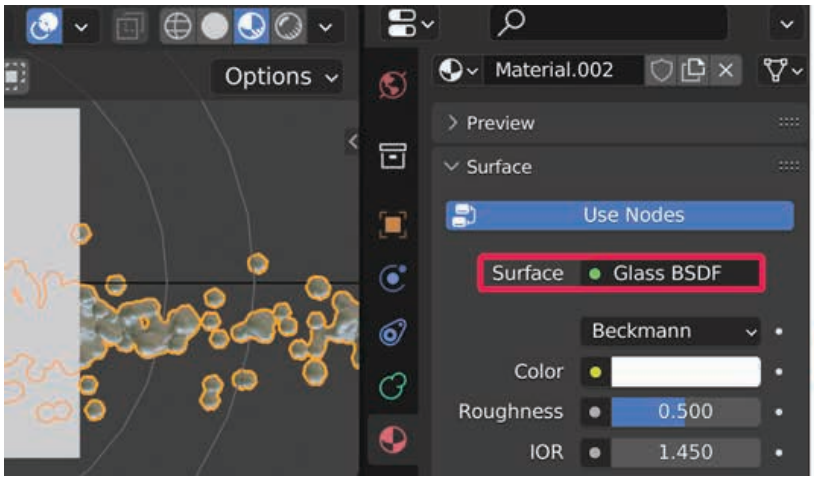


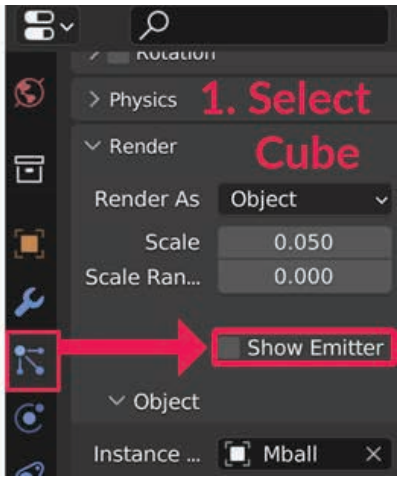
You can use high resolution for the renders.



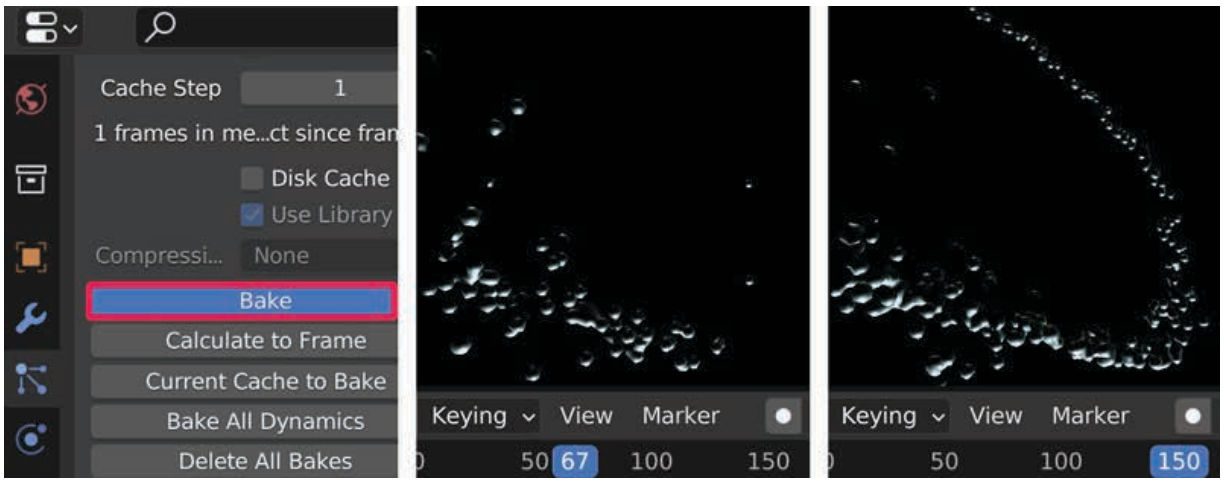


Give the Metaball a **Glass BSDF** material. In the particle system **turn off "Show Emitter"** so it doesn't render.

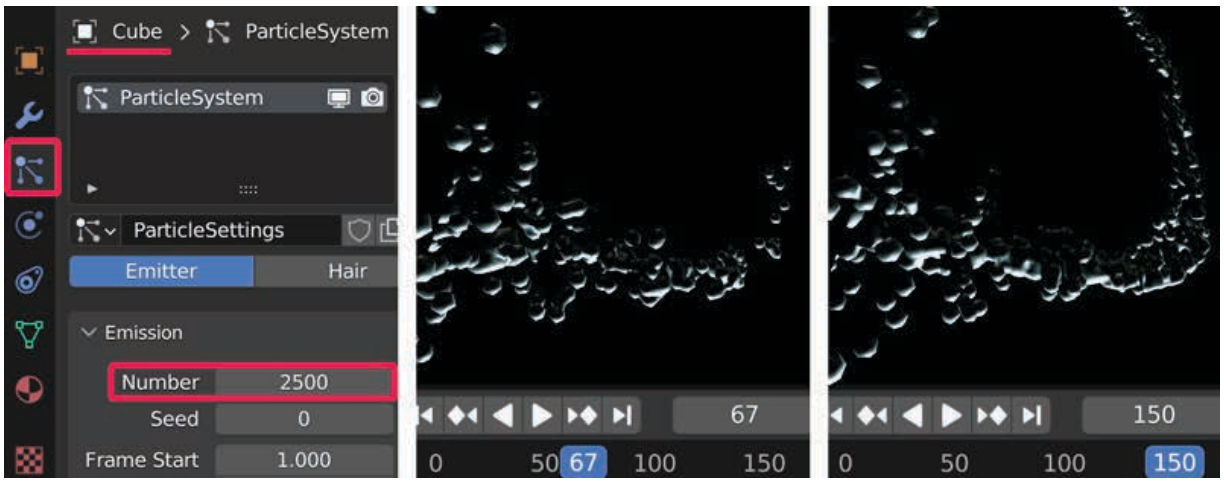




Now bake the simulation. Now you have a liquid stream following a Curve.



You may need to tweak the amount of particles to get the Metaballs close enough to each other to create a continuous liquid.



LIGHTING

A teacup with some nice glossy HDRI reflections. Fun fact: The teapot is for many 3D packages a default object, like the Suzanne monkey in Blender. It's also available in Blender, but you need to activate the "extra meshes" add-on in Preferences.

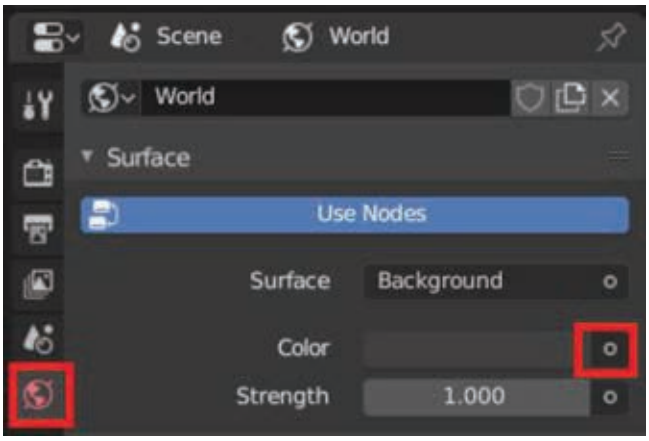
What are HDRI's?

HDRI's are an amazing and easy way to light scenes and objects in a very photoreal way without having to mess with lights (although you can still add more lights manually if you want). They are essentially 360° images that have multiple levels of exposure, so the render engine will use the bright exposures for the bright parts of your render and the dark exposures for the dark parts of your render. Kind of brilliant, really!

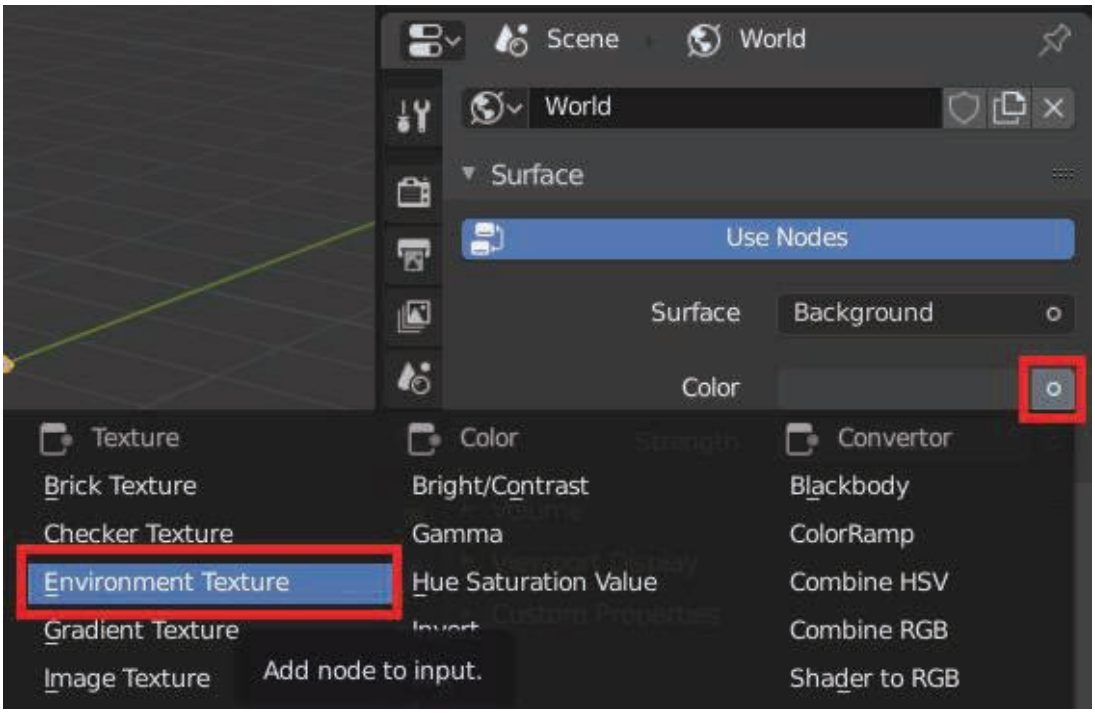
Great! Where do I get them?

These days many excellent free HDRI images can be found online, for example on Poly Haven (formerly HDRI Haven). Google "Free HDRI" and you'll find more.

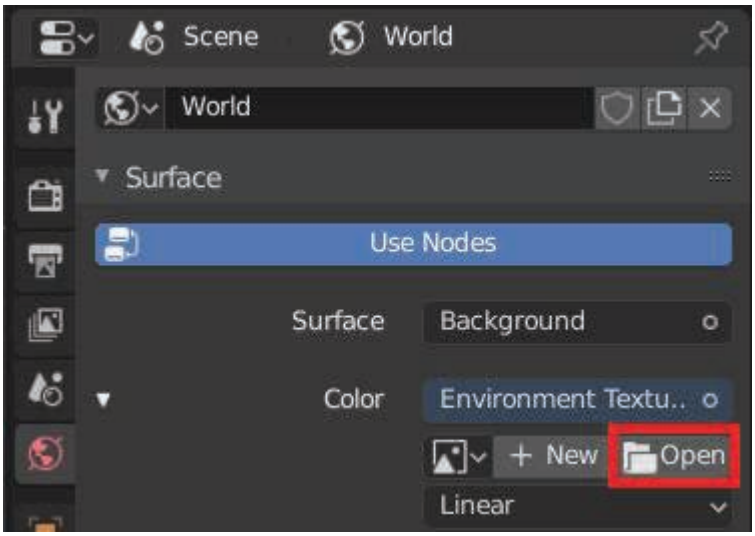
How to use an HDRI
To get started, click on the button next to "**Color**" in the **World tab**.



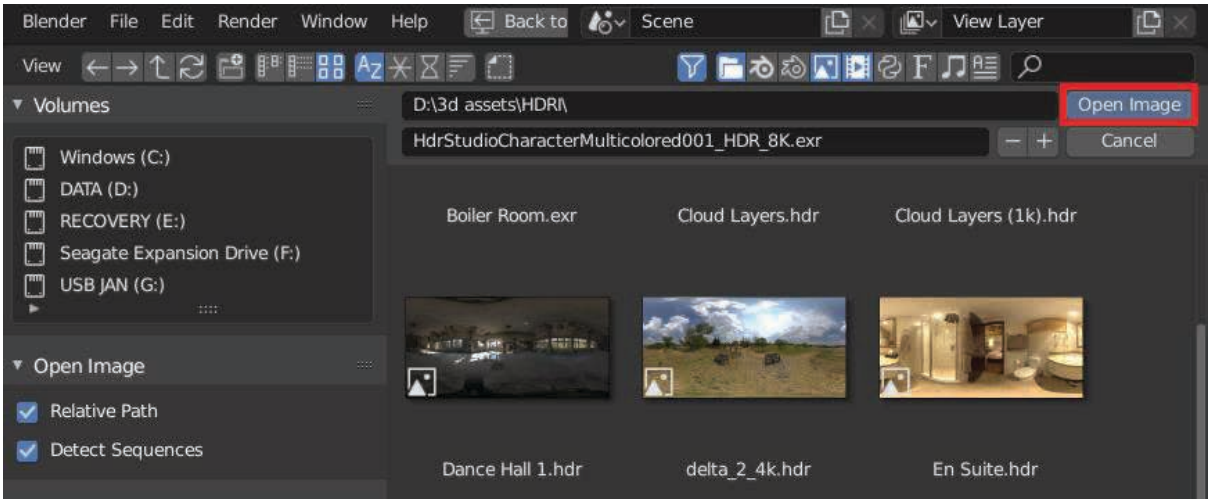
Then, choose "**Environment Texture**".



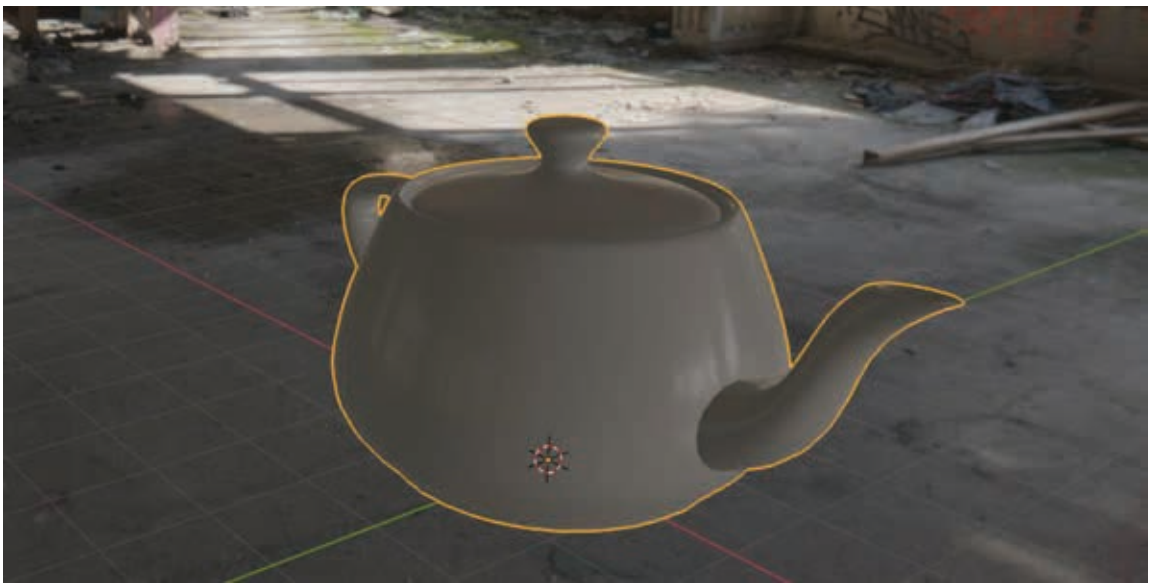
Assuming you've bought some HDRI's or downloaded some free ones, you now can click on "Open" to navigate to your folder containing HDRI images.



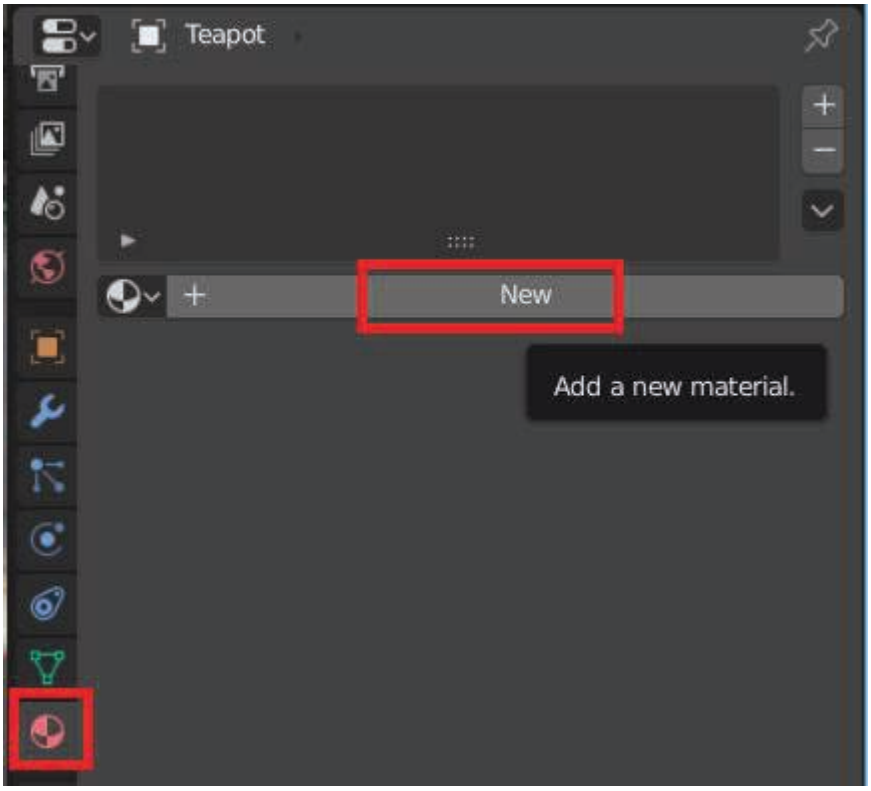
Find an HDRI you like and click on "Open Image". (There are some paid add-ons that make this process easier - but I tend to just try out a few different ones this manual way until I find one that works for my model).



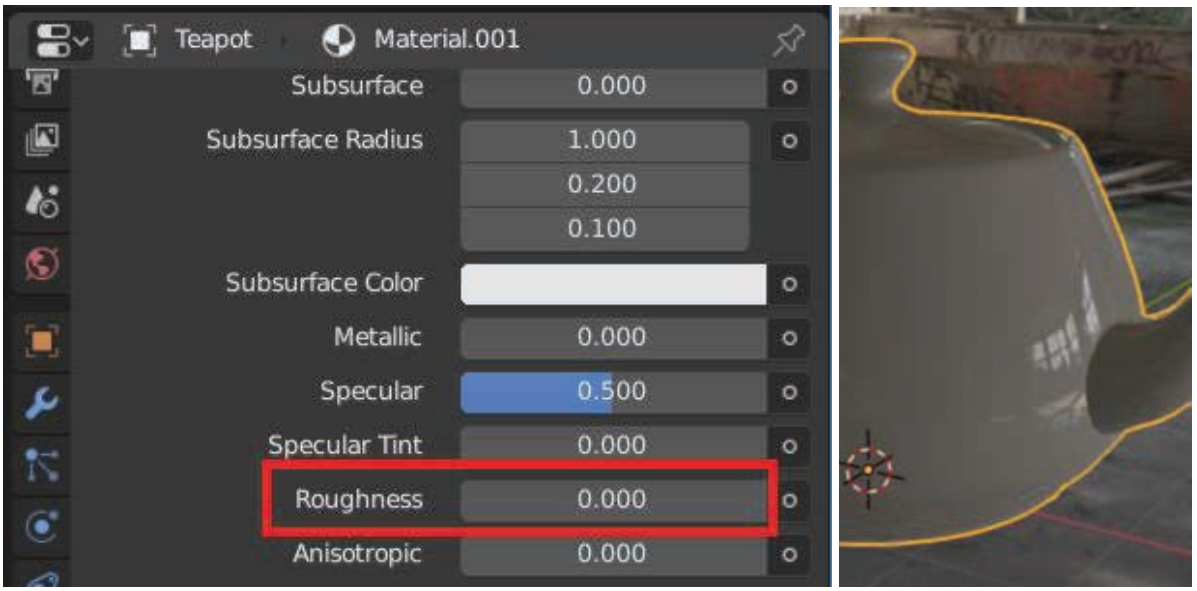
Press Shift+Z to toggle the viewport to Rendered. The HDRI is visible in the background, and rotates as it should when we rotate the viewport. But let's give our teapot a nice glossy material to see the HDRI reflections.



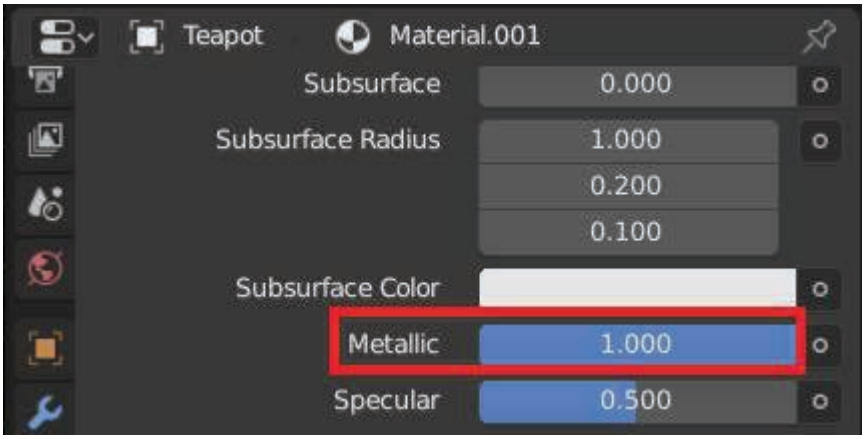
With the teacup selected, in the **Materials** tab, click on **New**.



Change the **Roughness** to 0, to make it super glossy (shiny). Some reflections already become visible in the teapot.



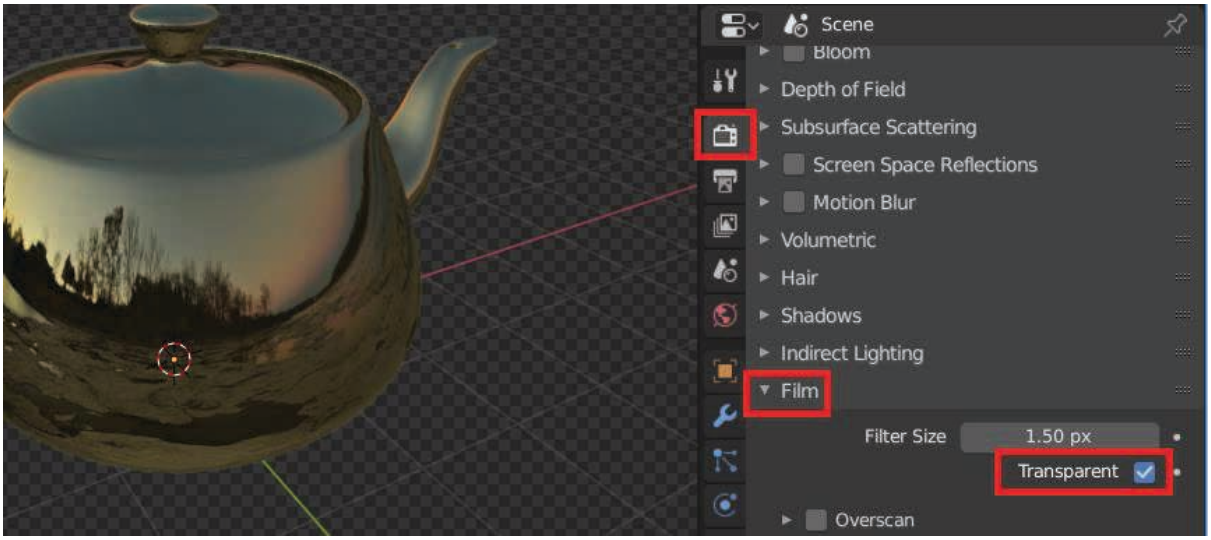
Let's turn **Metallic** on, set it to 1. Now the teapot is really super glossy.



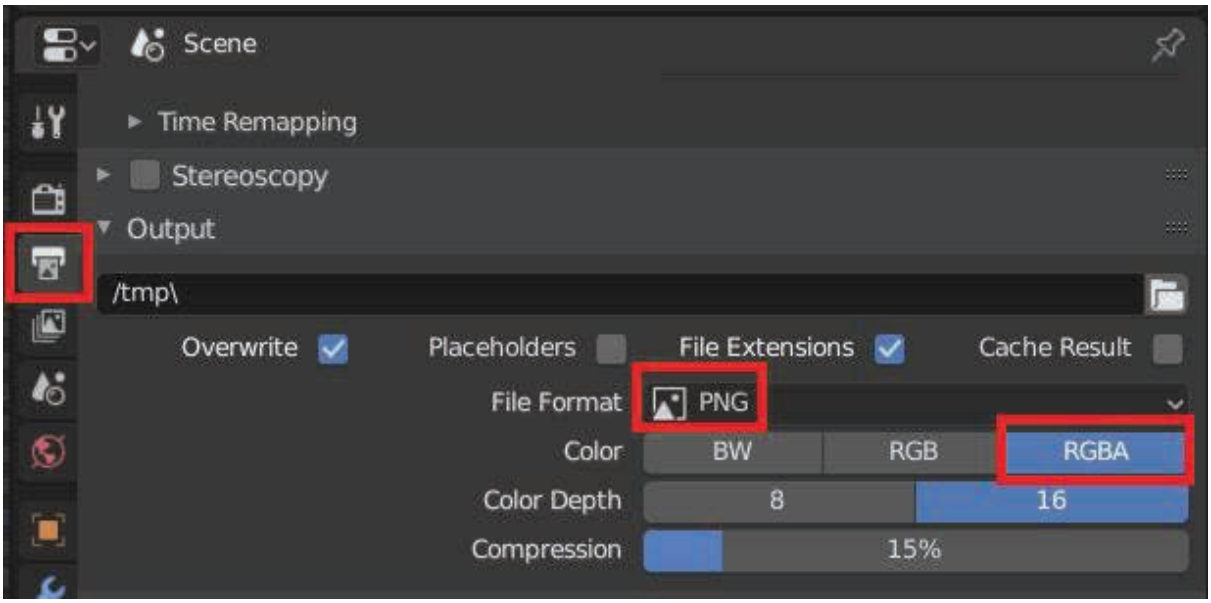
You can see the HDRI environment being reflected.

Not all HDRI's are based on photographs. Here's one that emulates studio lighting. I bought it from Poliigon (affiliate link). It's nice for car renders, for example.

That's all! In the Render Tab under "**Film**" you can set the background to "**Transparent**".



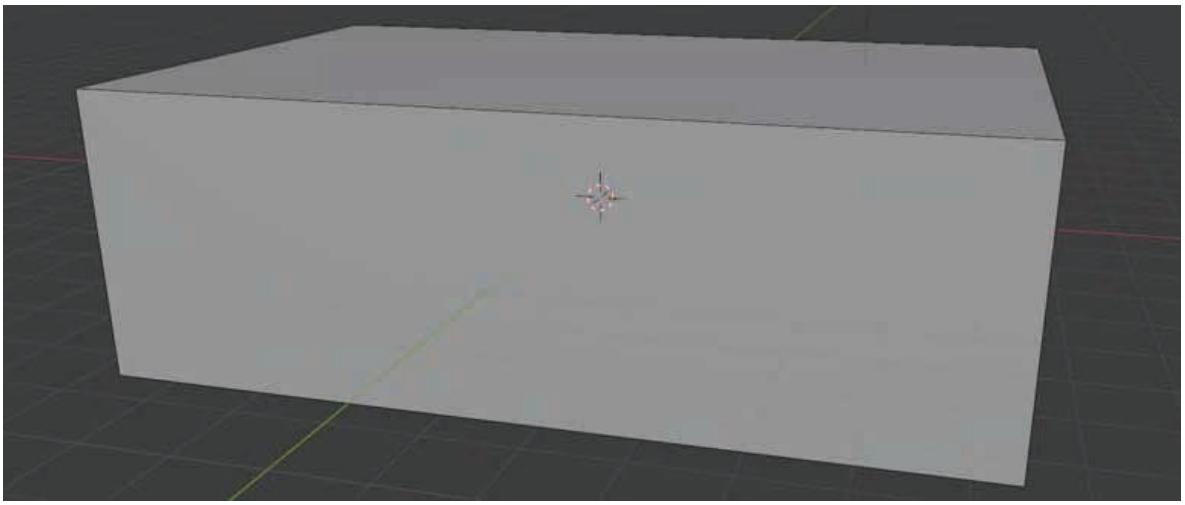
Press F12 to render a still image. Choose **PNG** (or EXR) and **RGBA** in the Output settings to keep the transparent background. PNG's are images that support alpha channels (transparency).



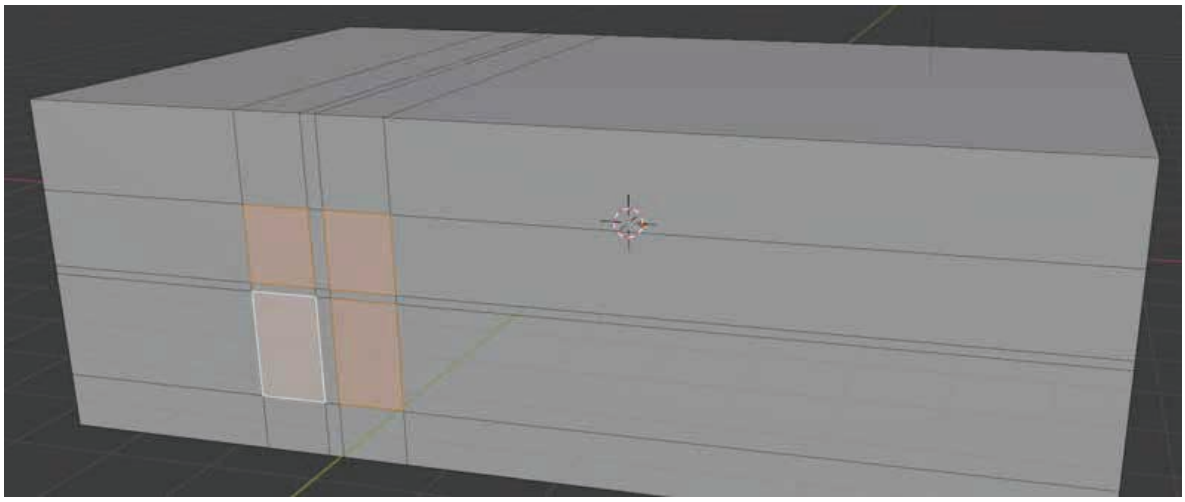
Sunset Timelapse with Sky Texture

One of the features that were introduced with Blender 2.9 is the **Nishita Sky Texture**, which allows us to make beautiful sunset animations in Cycles.

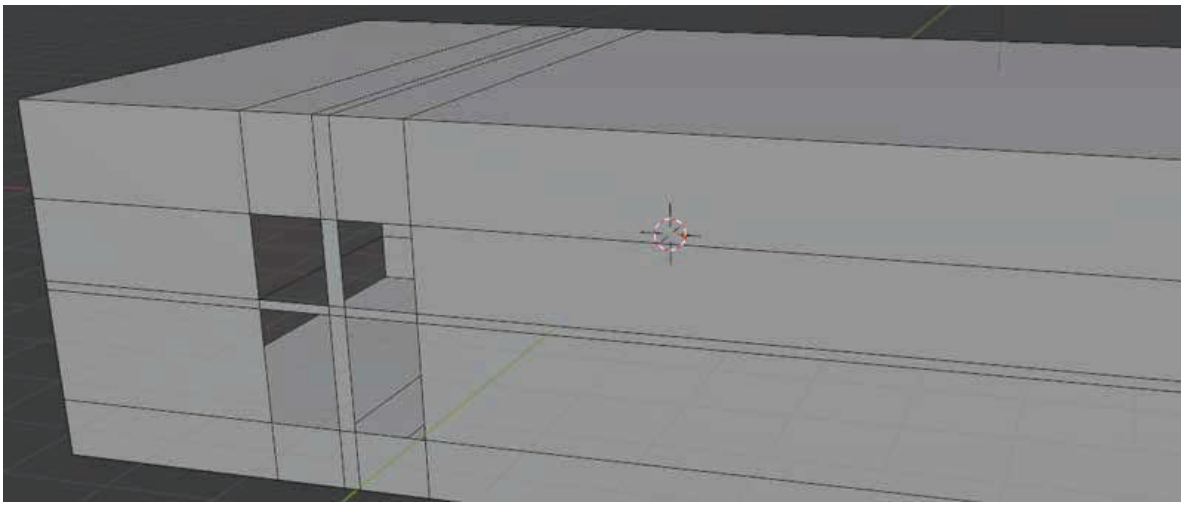
To create this room, I use the default cube and scaled it along the X and Z axis to get more of a room shape. **Press S+ an axis (X, Y or Z)** to scale only along that axis, for example **S+X** to scale along the X-axis only.



It's always best to scale in Edit Mode, that way any modifiers we place later will behave more predictably. By **pressing Ctrl+R**, you can add some loop cuts so that you can cut out the windows.

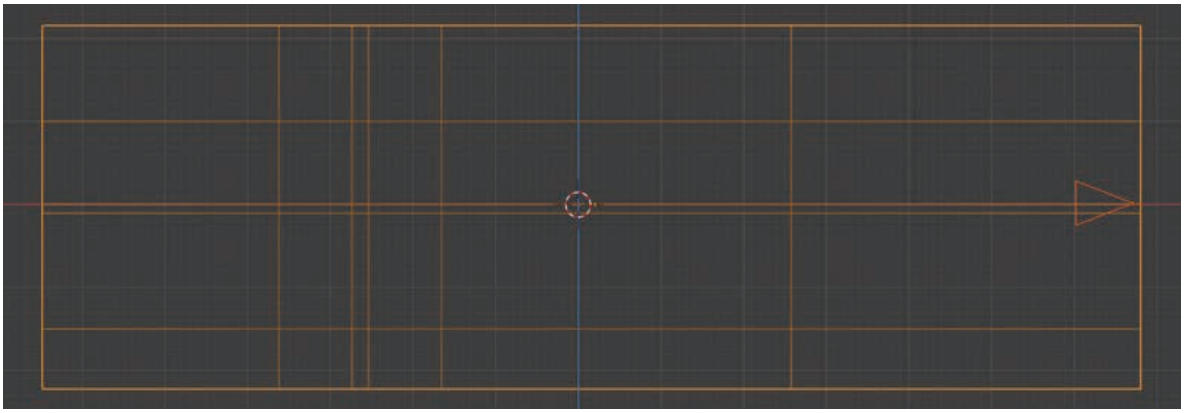


Switch to **Face selection mode** (you can **press 3** on the keyboard to switch to this mode quickly), select the faces that will become the window holes and **press X**, then choose **Faces** to delete them.

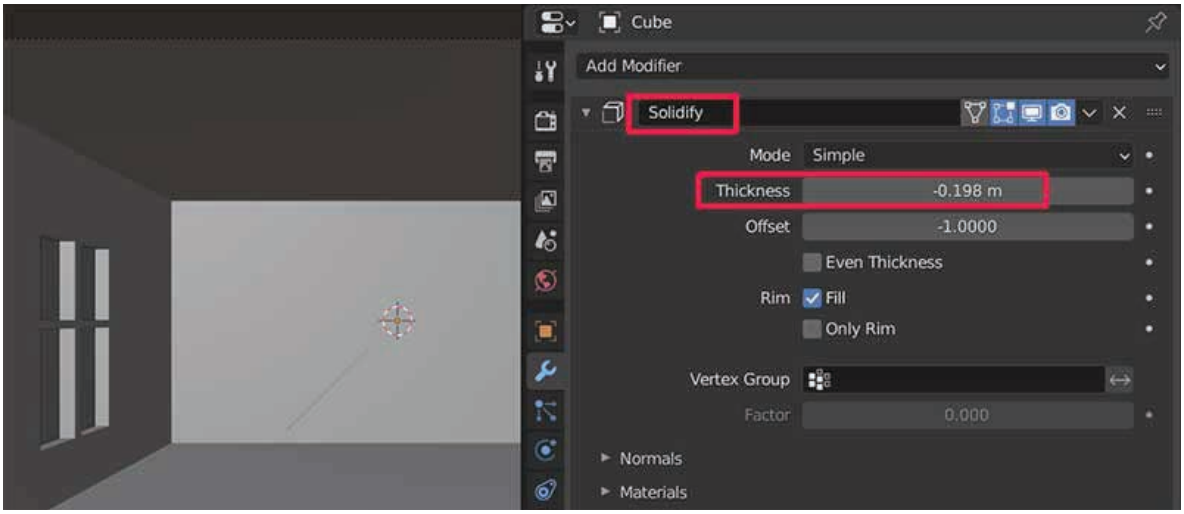


Move the camera to inside be the cube. You can do this manually by **rotating (R)** and **moving (G)** the camera, then **press 0 on the numpad** to switch to camera view.

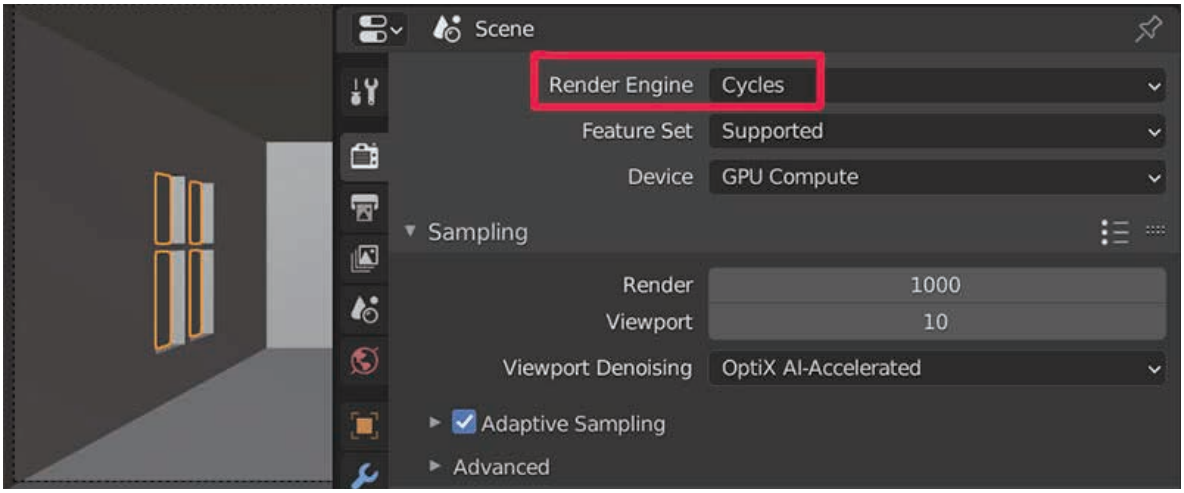
Alternatively, move the viewport view to how you want the camera to see the scene and **press Shift+Ctrl+0 on the numpad** to align the camera to that view (this is my preferred method).



Then add a **Solidify modifier** to the cube and adjust the **Thickness** value so the wall has some thickness. This can help to avoid light leaking in some cases.

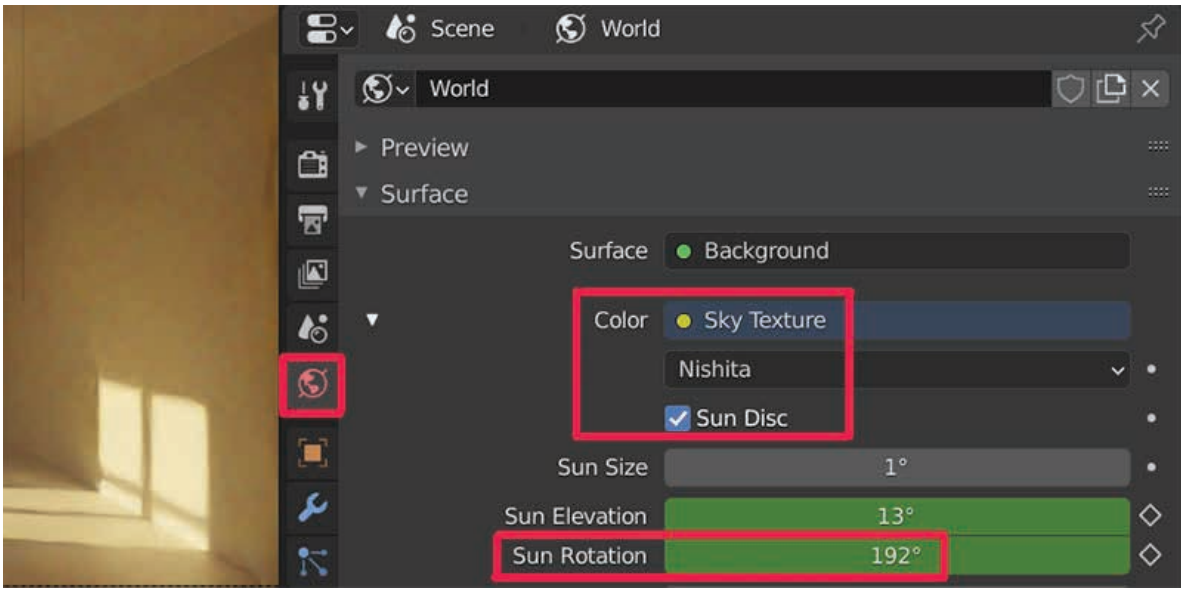


Set Render engine to **Cycles**. I've turned on **GPU** as the device as well as **OptiX** AI-enhanced denoising and **Adaptive Sampling**. This is optional and not necessary for the Sky Texture to work, it just helps everything go faster. To have the OptiX option, it needs to be enabled in Preferences as well (under "System").



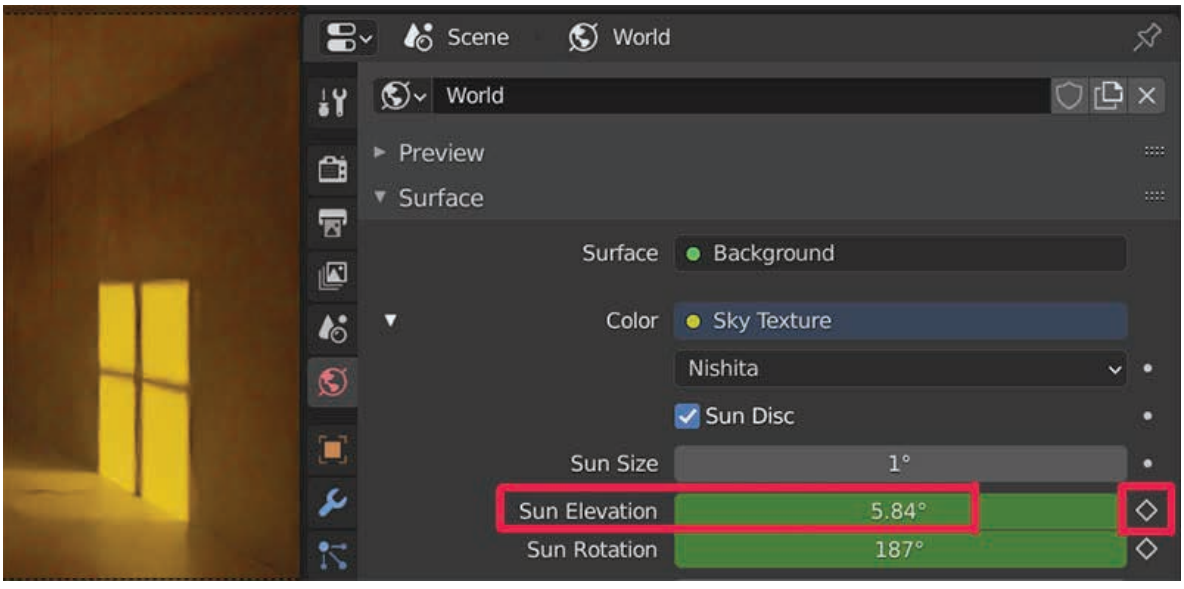
In the **World** tab, set **Color** to **Sky Texture**. Set it to **Nishita** with the **Sun Disc** enabled.

Sun Rotation changes the horizontal position of the sun. You probably need to change this so that the sun shines through the window.

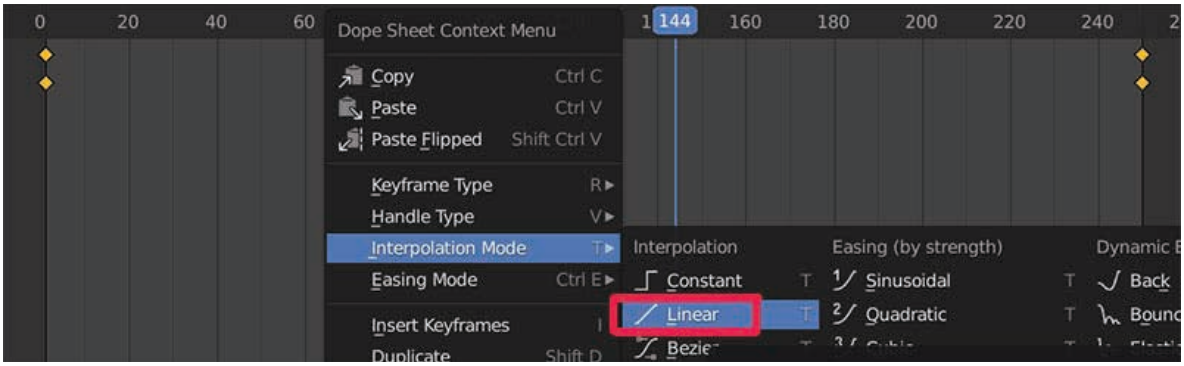


Right away you get some very nice lighting. To create a sunset timelapse animation, you can set keyframes for the **Sun Elevation** value.

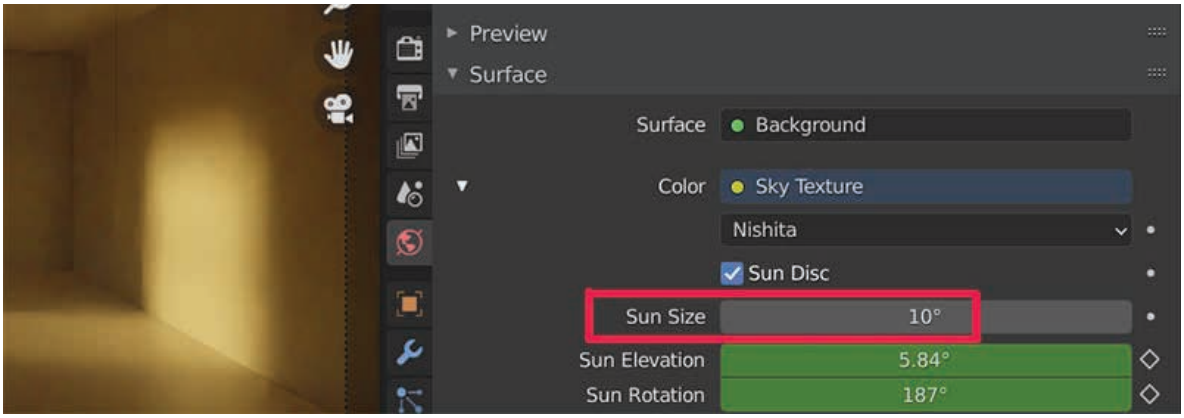
You can go to the first frame on the Timeline and enter a Sun Elevation value and click on the keyframe icon, then go to the last frame on the Timeline and enter another value, and click the keyframe icon again.



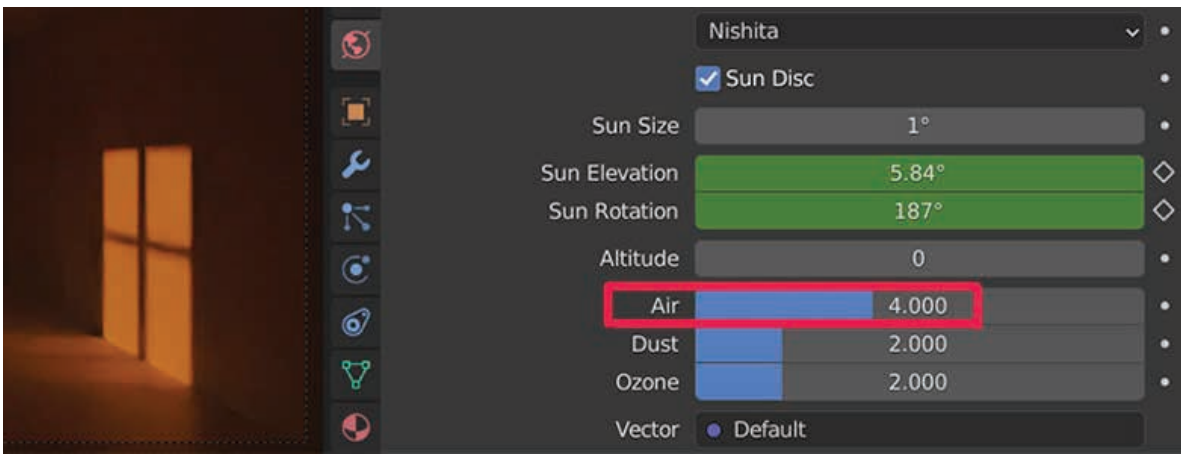
Set the keyframes to **Linear** for more realistic animation (I personally prefer to have the keyframes set to Linear by default in my Preferences). To do this **select the keyframes**, **right-click** and choose Interpolation Mode > **Linear**. Alternatively **press Shift+E** and choose **Linear Interpolation**.



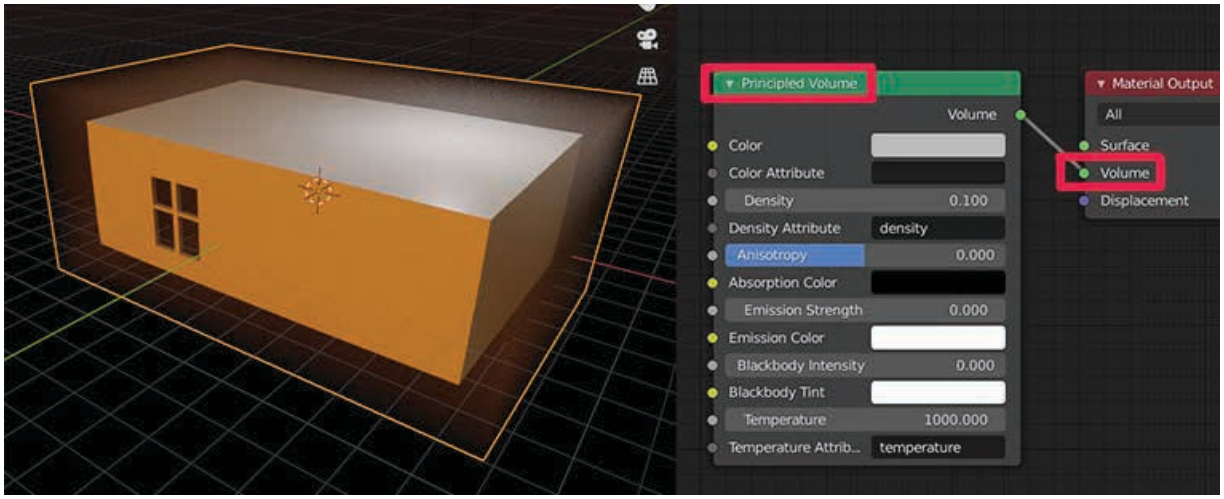
Sun Size affects the sharpness to softness of the shadows. The bigger the sun the softer the shadows. Of course, this is not really realistic (the sun tends to not get bigger in real life, fortunately) but it can be a quick "cheat" to get beautiful soft shadows. For example here the sun has an exaggerated size.



The **Air**, **Dust** and **Ozone** values control how much the air quality influences the light. Increasing the Air value gives more dramatic sunsets.



For some nice volumetric effect, add a cube around the cube and give it a **Principled Volume** material. Just make sure to plug the Principled Volume output into the Volume input of the Material node. This doesn't happen by default when choosing Principled Volume as the material.

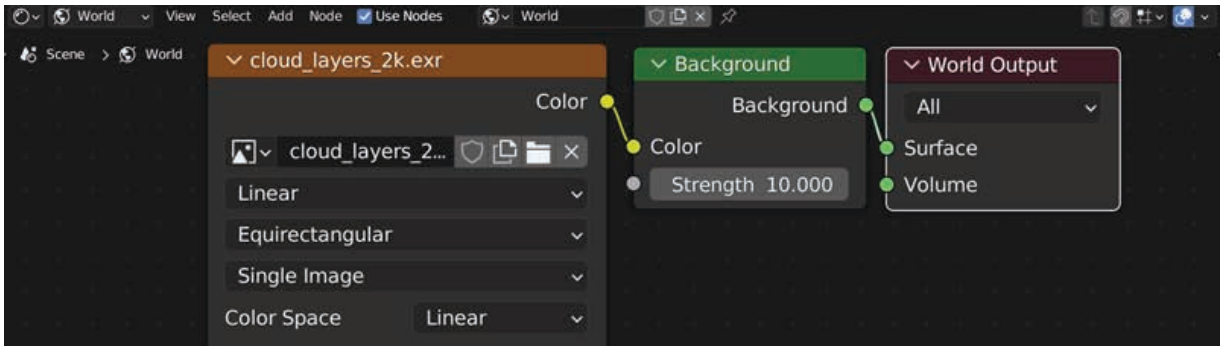


Volumetric background to HDRI transitions

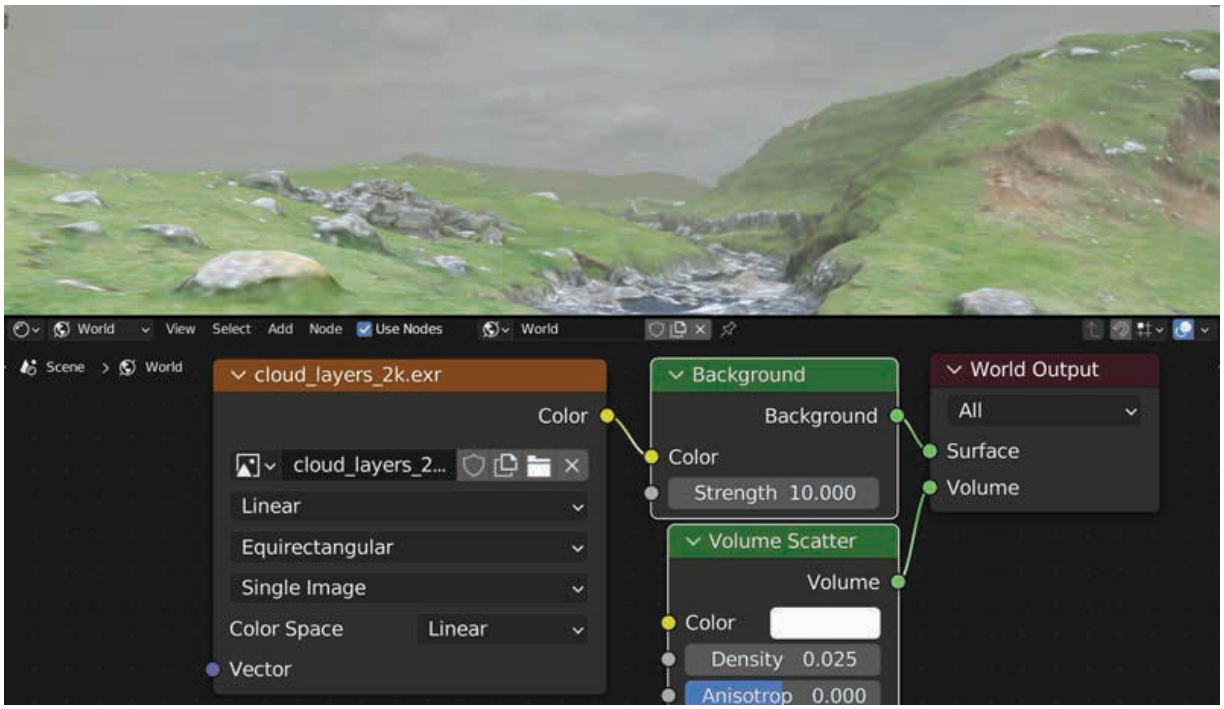
To combine a 3D scanned foreground like this model from Sketchfab with an HDRI background, it helps to transition between them with a subtle volumetric effect.

You can find this 3d scanned environment on Sketchfab.

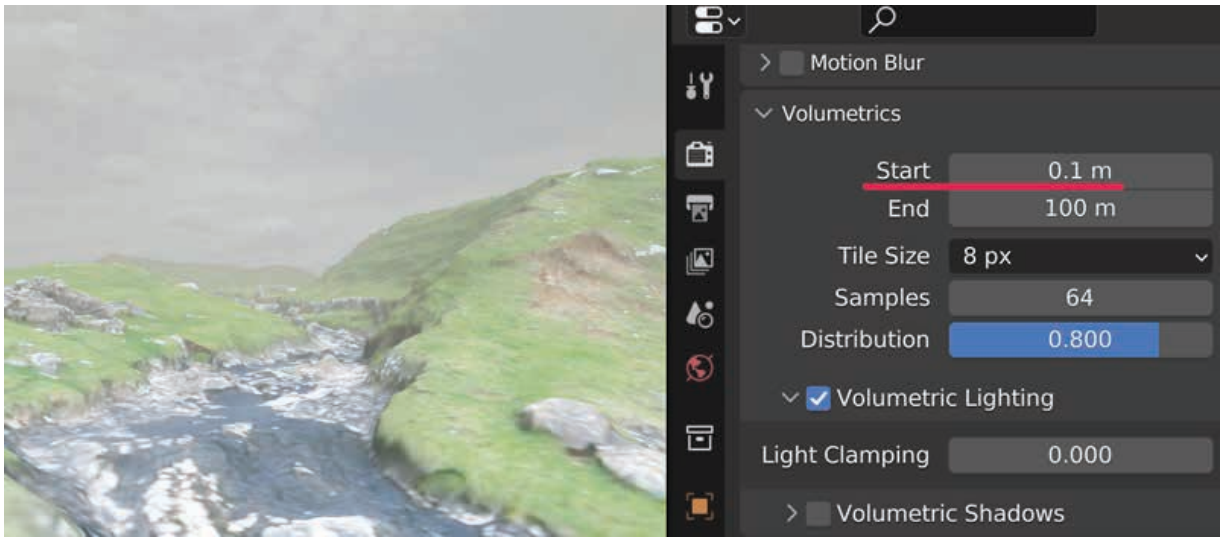
This is a 3D scanned landscape model from sketchfab with an **HDRI** as the world background. You can find many free HDRI images on Poly Haven.

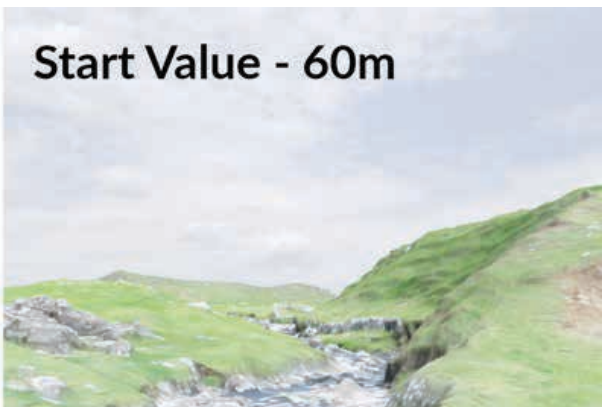


In Eevee, to make the transition between foreground and HDRI a bit more interesting we can add a volumetric effect. In the world tab add a **Volume Scatter node** and plug it into the **Volume** input. Decrease the **Density**.



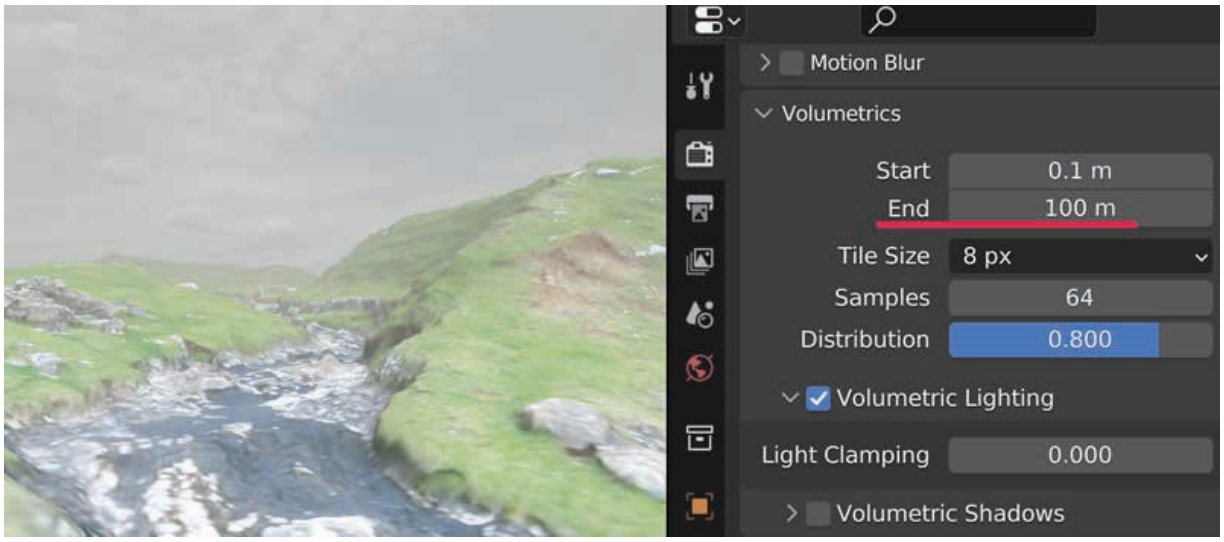
Now you can push the volumetric effect back by increasing the **Start** value.



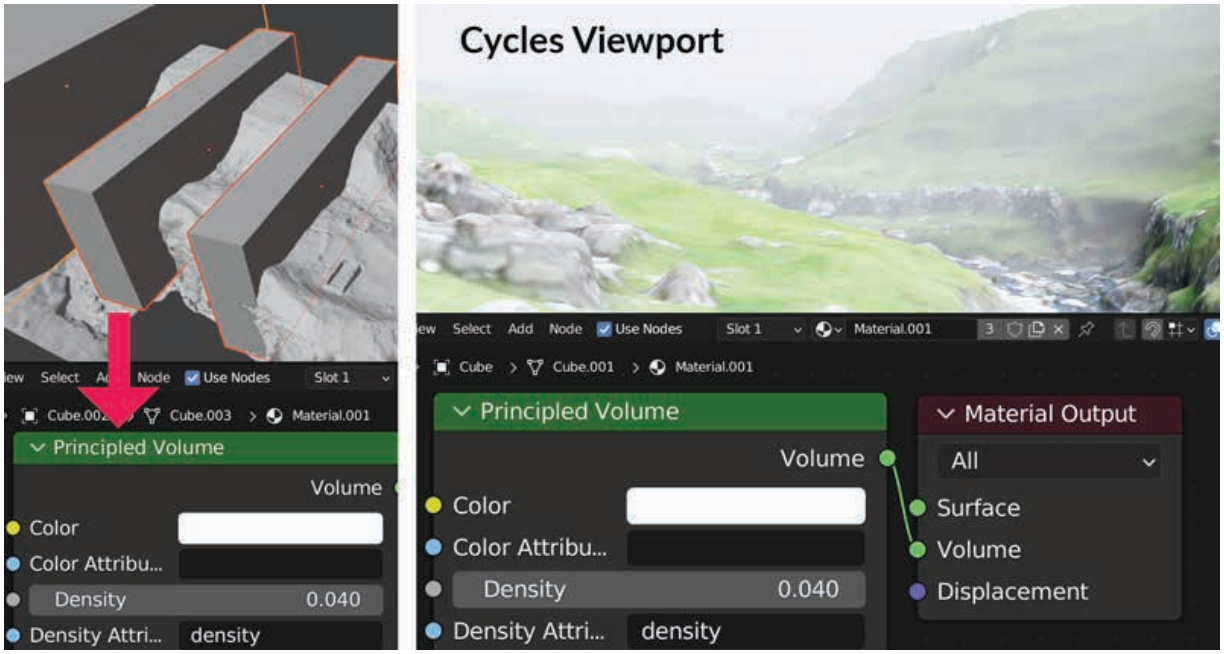


Click here to return to the index.

You can also extend it further into the distance so the sky becomes less clear by increasing the **End** value.



For an alternative that works well in cycles. You can simply place a cube (or multiple cubes) with a **Principled Volume shader** on it in the scene. Be sure to connect the Volume output to the Volume input and to lower the Density.



How to use IES lights

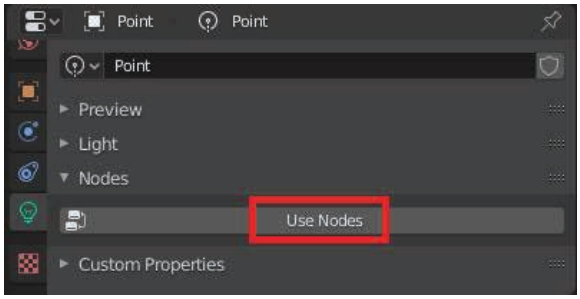
IES files are text files that describe how specific lights look in real life.

They are often used in Architectural Visualisation for their realism.

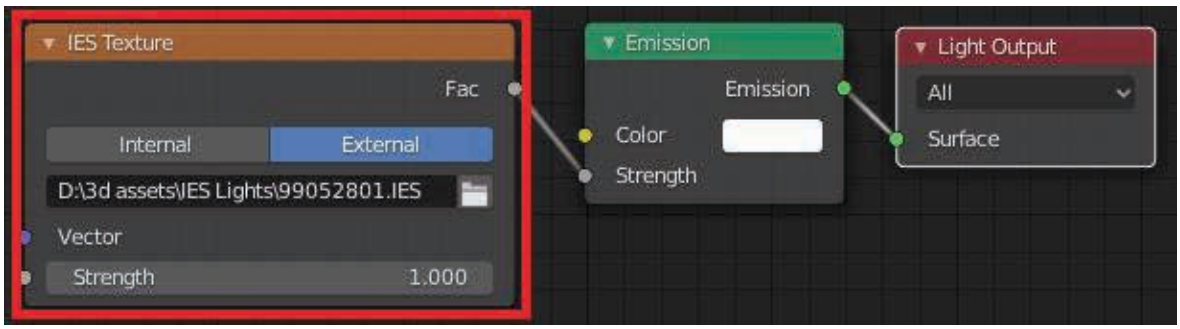
Many IES filed can be found for free online.

To create an IES light, first add a **Point Light** in Blender by **pressing Shift+A** and choosing **Light > Point**.

Click on Use Nodes in the light settings. IES only works in Cycles so make sure you are not in Eevee render mode. If you are in Eevee you won't have this Use Nodes option.



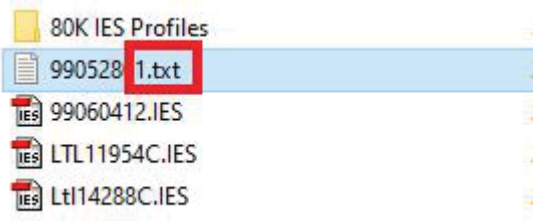
In the Shader Editor **press Shift+A** and find the IEStexturenode. Connect its Fac output to the Strength input of the Emission node. Then click on External and choose the IES text file you want to use.



One extra tip for those of you who use a renderfarm. It's safer to set the IES node to "Internal" in this case and open a txt file that's already loaded into Blender. I've noticed that IES lights don't work otherwise, even when they are uploaded on the renderfarm in a subfolder of the .Blend file.

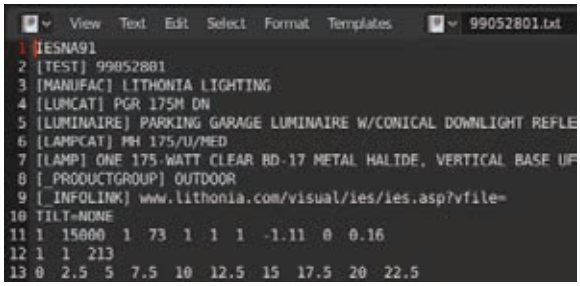
To do this first you have to change the extension of the IES file you want to use to ".txt".

That way Blender can read it with its text editor.



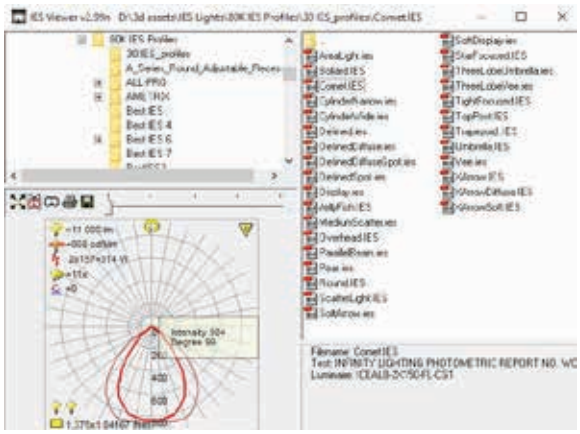
Then open a **Text Editor window** in Blender and open the .txt file.

After this you can load it into the IES Texture node.



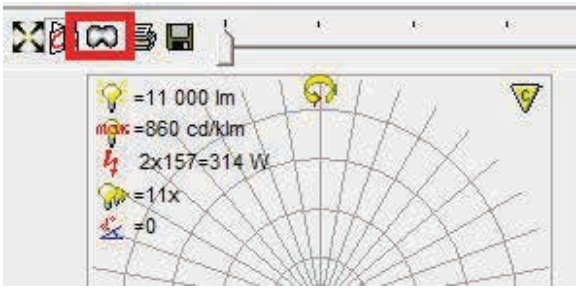
Bonus IES tip: To quickly see what the IES files look like, use the free program IESViewer.

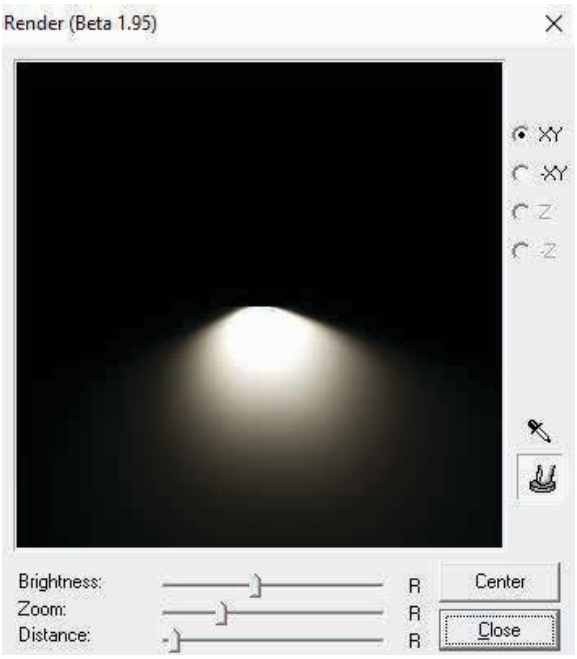
Get it from
http://photometricviewer.com/



Click here to return to the index.

It even has a render function, to see what the light may look like in your render.



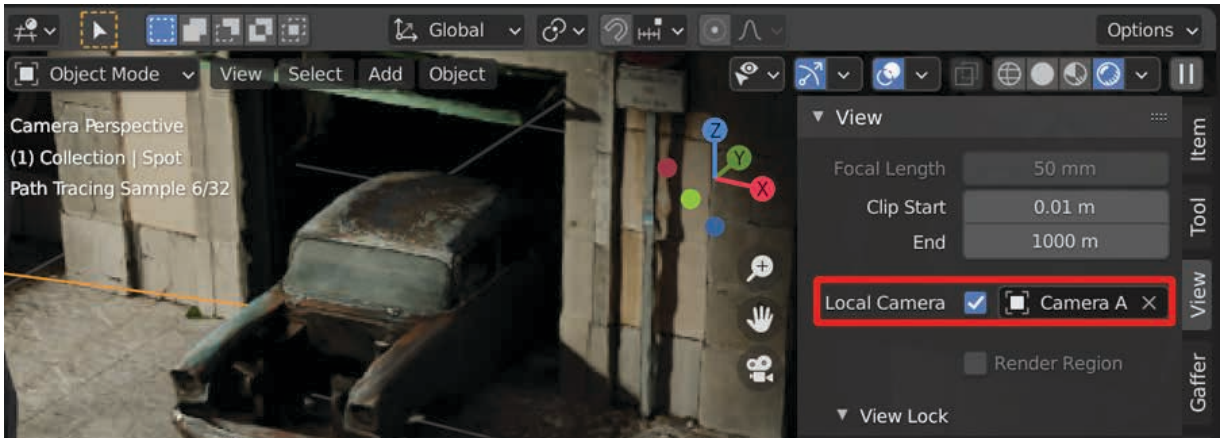


And finally, to get **80.000 free IES files**, go here: https://gumroad.com/l/FXsj

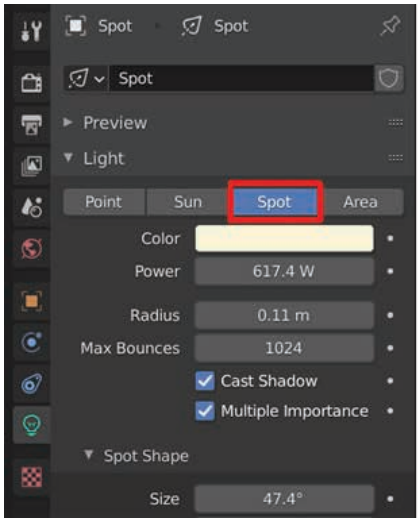
CAMERA TIPS

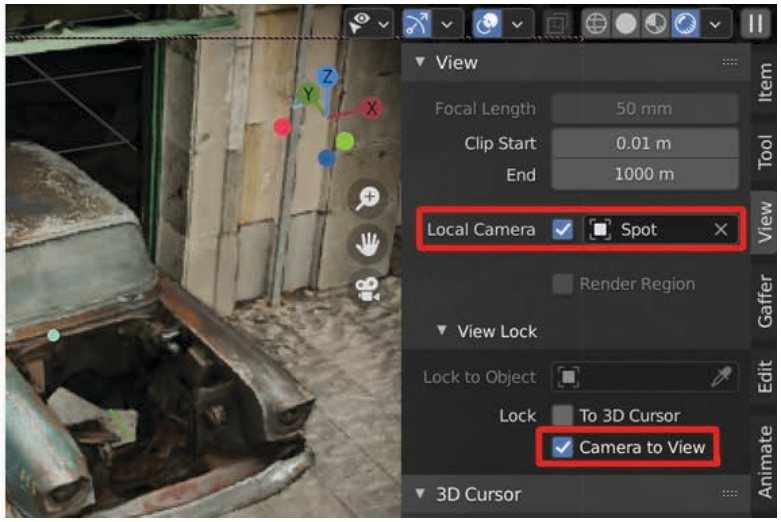
Different cameras per viewport & adjusting lights

If you have multiple cameras in your scene and want to use a different camera per viewport, open the Option panel by **pressing o**. Then check **Local Camera** and select the camera you want.



You can choose a different Local Camera for each viewport. As a bonus tip, you can select a light instead of a camera (really any object can be a "camera"). For example a Spot light. You can always change Light types in the Light's Object Data Properties tab. Check **Lock Camera to View**.

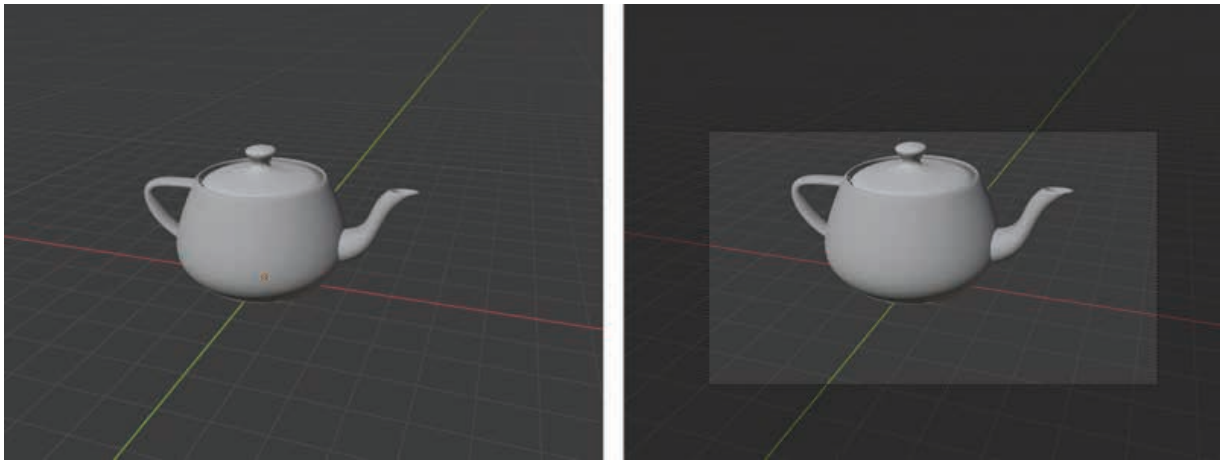




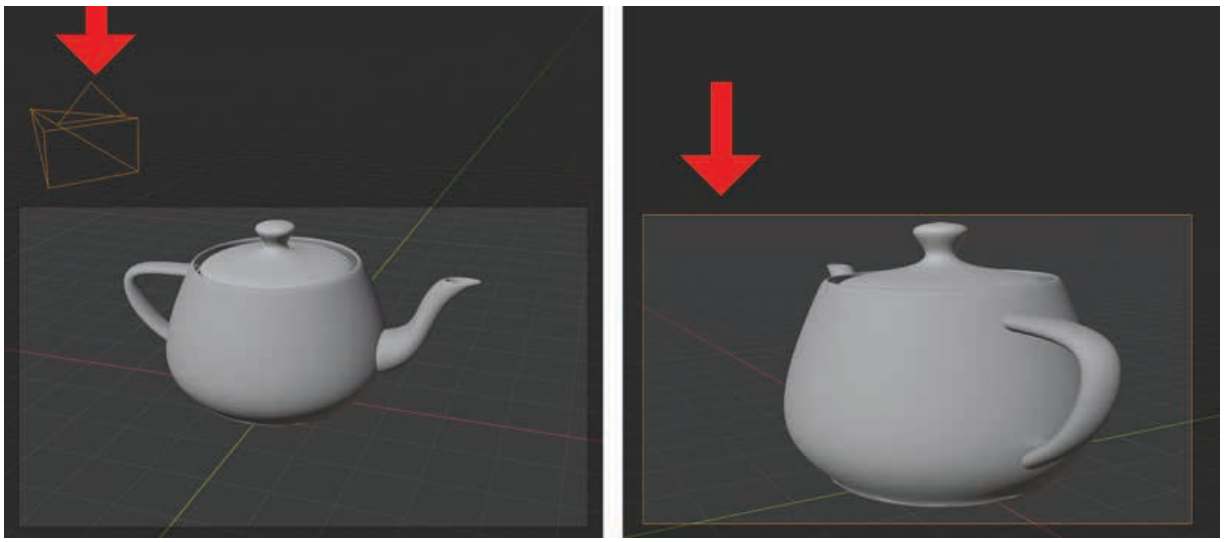
Then to enter the light's "point of view", **press 0 on the numpad**. Then **hold the middle mouse button** to aim the lamp. In the Spotlight settings, you can make the spot softer or sharper by adjusting the Radius and Blend values.

3 Camera Tips

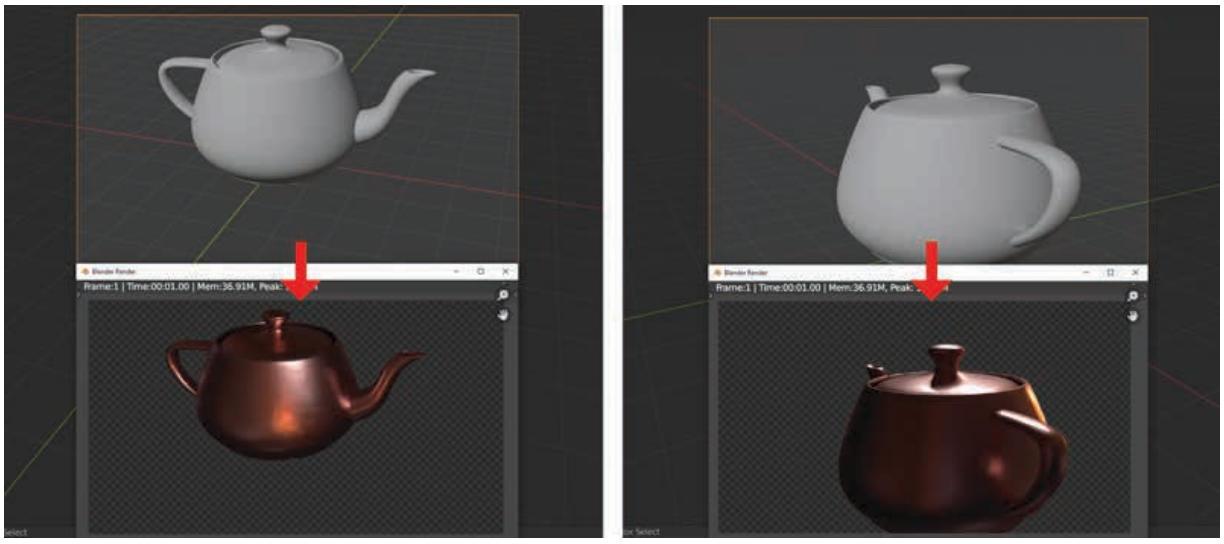
If you want the camera to match what you see in the viewport press Alt+Ctrl+O. This will move the camera to match what you're seeing in the viewport.



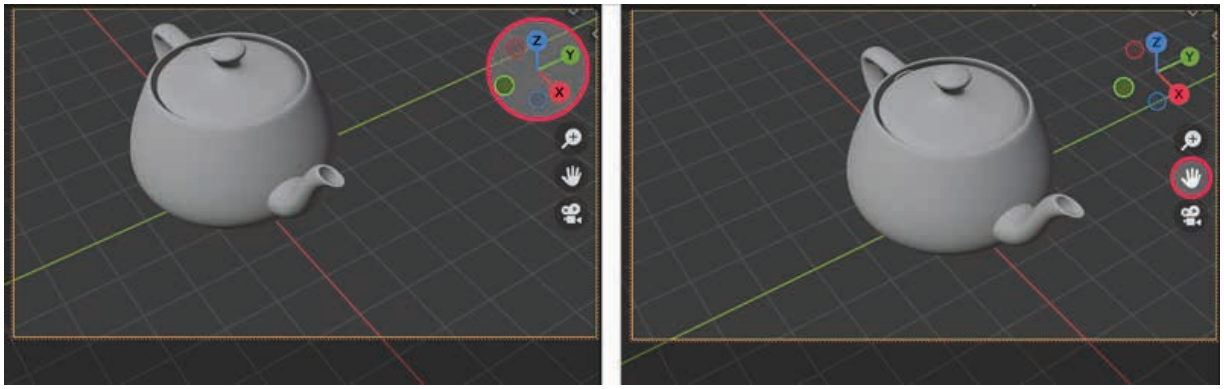
To change the active camera, select the camera you want and press Ctrl+O.



When you render, it will be what the active camera sees.



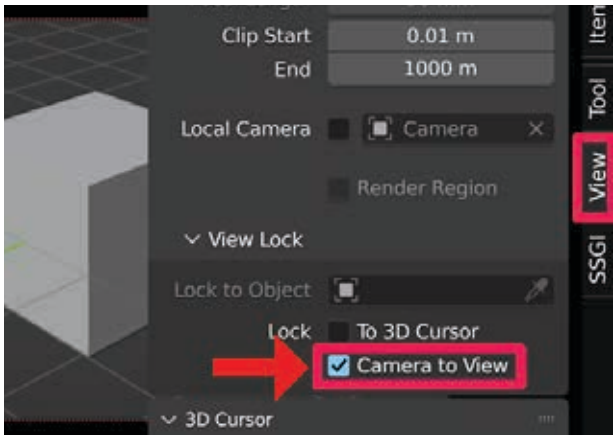
Check the box next to Lock Camera to View to be able to move the camera view much like the viewport.

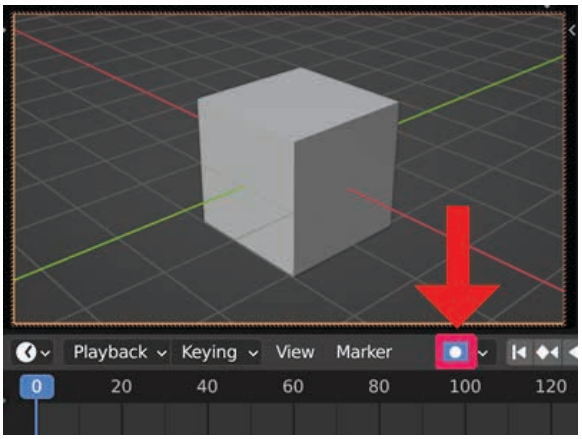


Easily make smooth camera moves

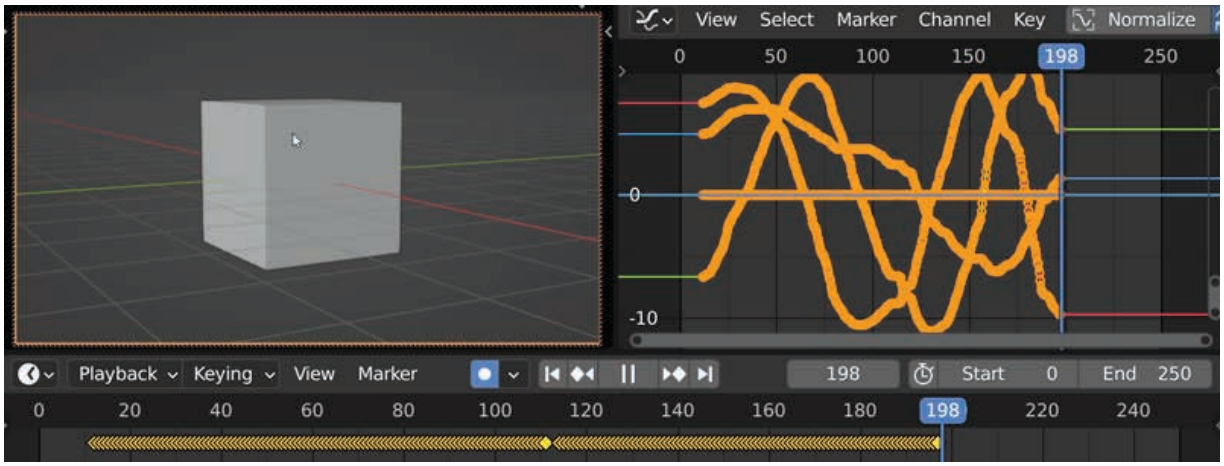
In the Option panel enable **Lock Camera To View**.

Turn on Auto Keying.

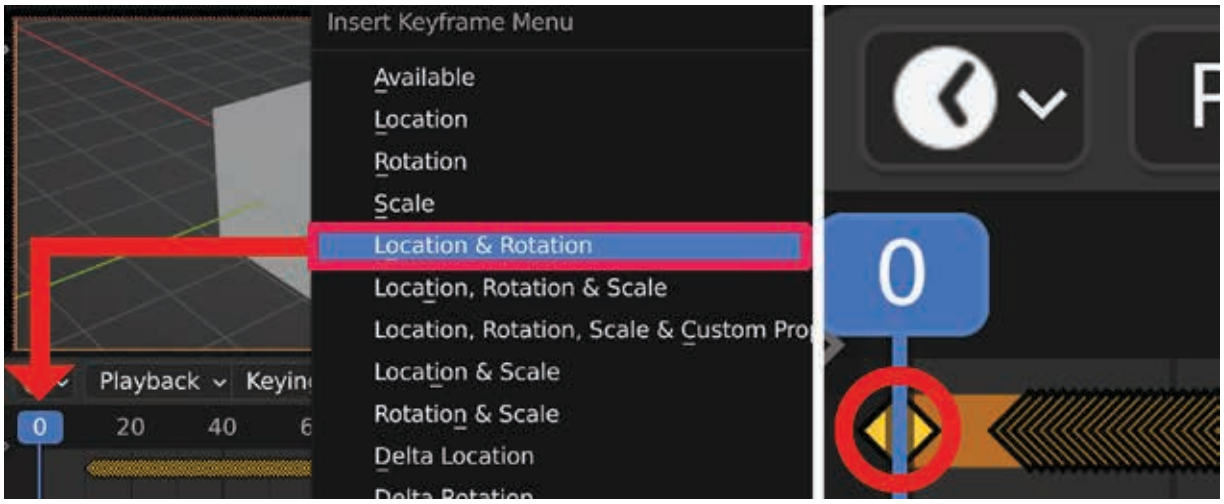




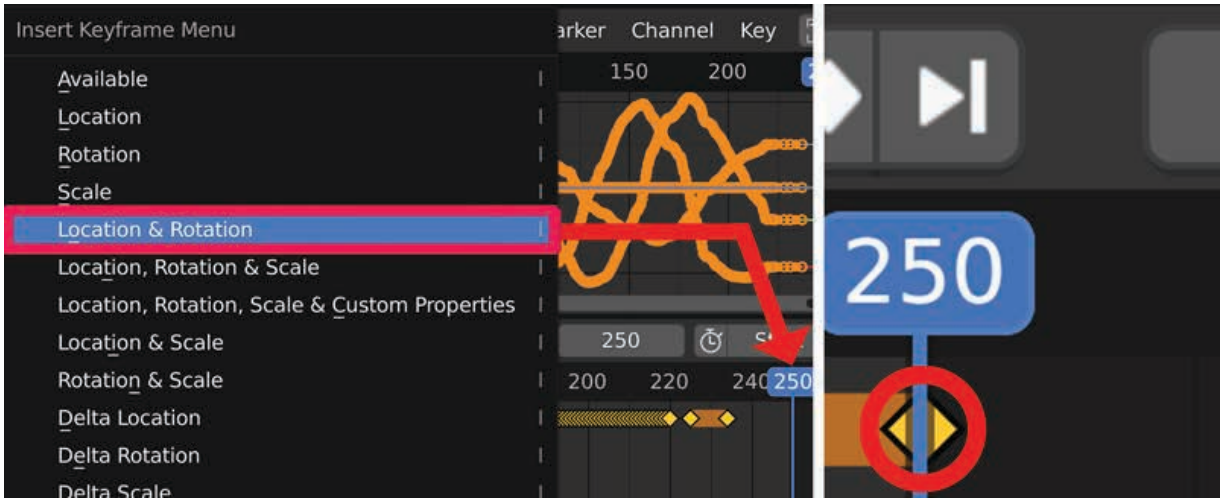
Then press the **Spacebar** and record your crazy camera move by moving the view as you normally would. You can see many keyframes being recorded. Press the Spacebar again to stop recording.



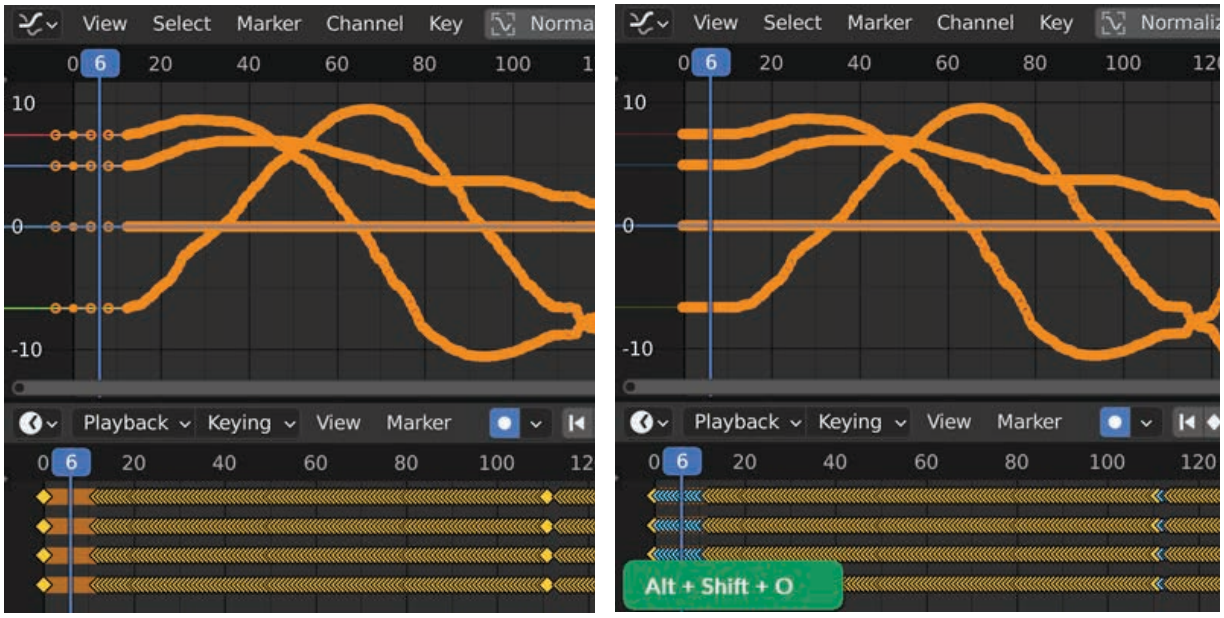
Now the Dopesheet will have a lot of keyframes. Insert a Location & Rotation keyframe at the start of the Timeline by **pressing i** and choosing **Location & Rotation**.



And add another Location & Rotation keyframe at the end.

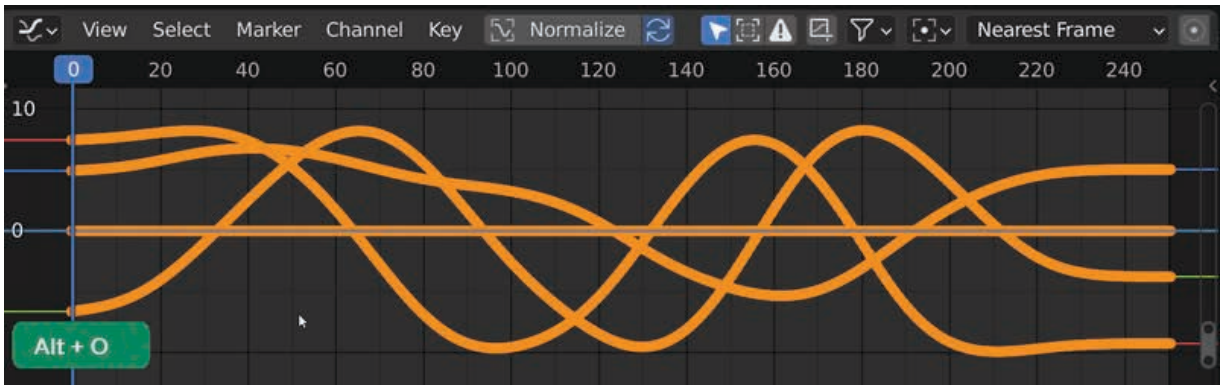


Press Shift+Alt+o to add keyframes where nothing was recorded.



Click here to return to the index.

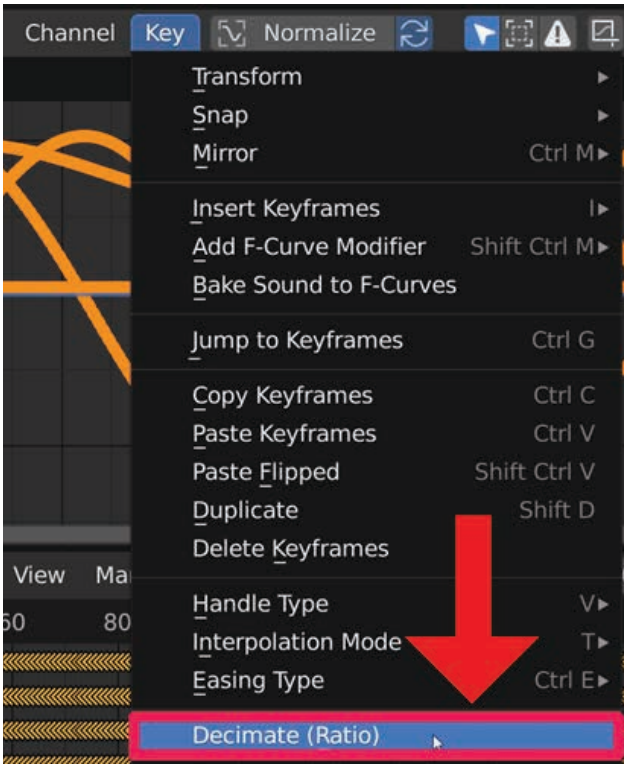
Below you see what the animation looks like in the Graph Editor window. **Press Alt+o** to make the animation smoother. Repeat pressing the Alt+o shortcut to make the curves smoother. Or go to **Key > Smooth Keys** if you don't want to memorise this shortcut.



Now if we press the Spacebar we can see that our camera move looks quite smooth!

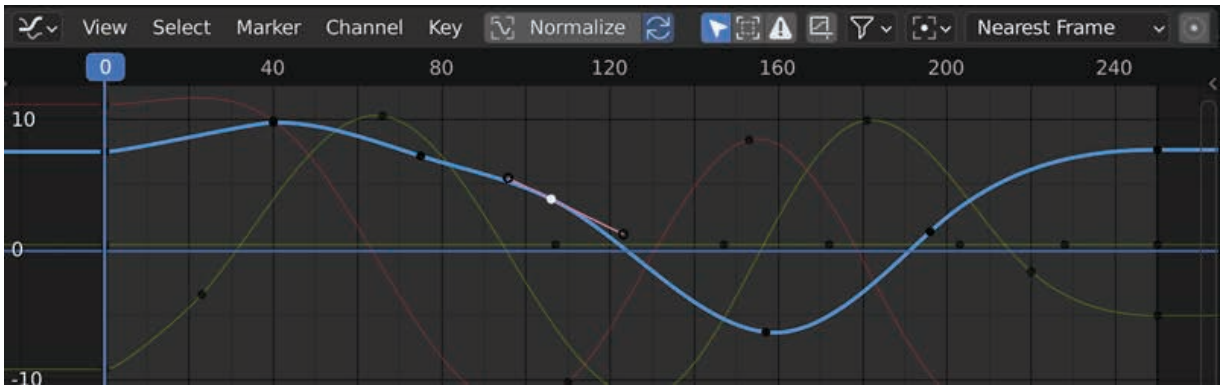
However we have too many keyframes if we want to be able to adjust the camera animation manually.

To fix this go to **Key** and choose **Decimate (Ratio).**



Move the mouse cursor to the right slowly until you get a decimation ratio that looks good.

With only the essential keyframes for the animation left you can tweak them individually to make the camera movement perfect.



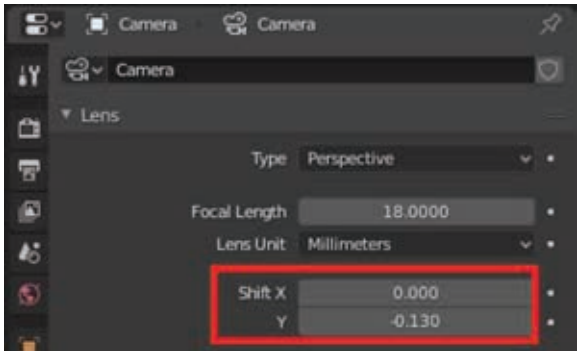
Camera Shift

Let's say you have an architectural visualisation (archviz) scene and you want to see less of the ceiling and more of the floor.

If you don't want that, use the **Shift** setting instead.

For example, changing the **Y value** shifts the camera up and down without a change in perspective.

If you tilt the camera down, you get a lot of perspective distortion because of the wide angle of the lens. Notice the vertical lines especially no longer go straight up.



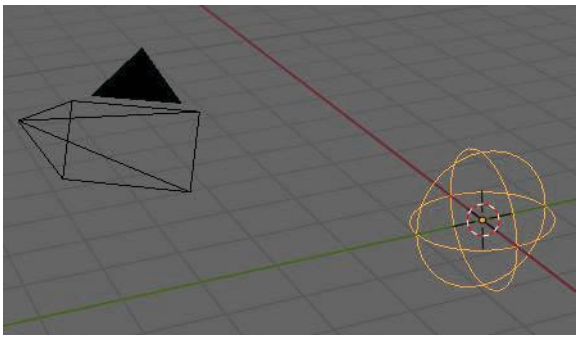
Now the composition changes without getting a change in the perspective.

Notice that in this case the vertical lines remain perfectly vertical.

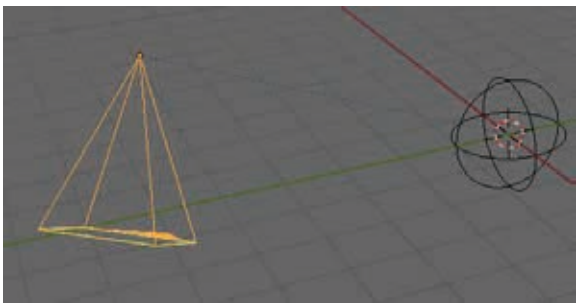
Credits for this Archviz scene: Marek Moravec.

Camera track to constraint

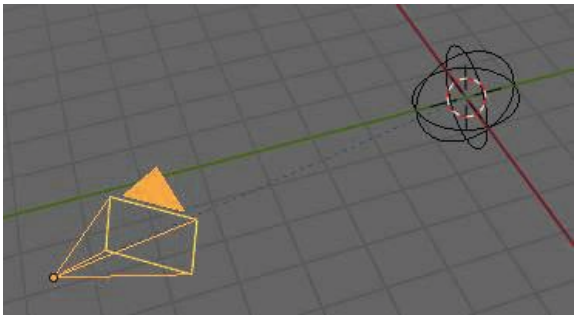
Press Shift+A and choose **Empty** > Plain Axes (or any kind of Empty - for this example I chose a Sphere Empty because it's easier to see in the picture).



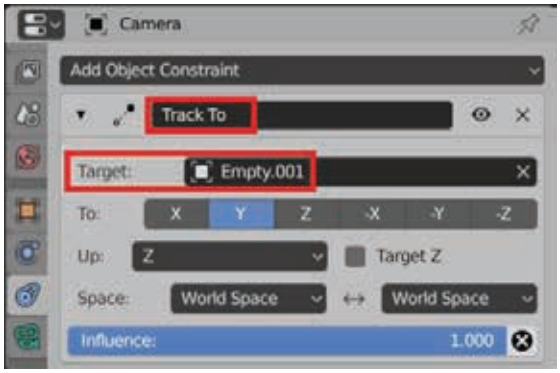
Now the camera is constrained to the Empty, but it's pointing down.



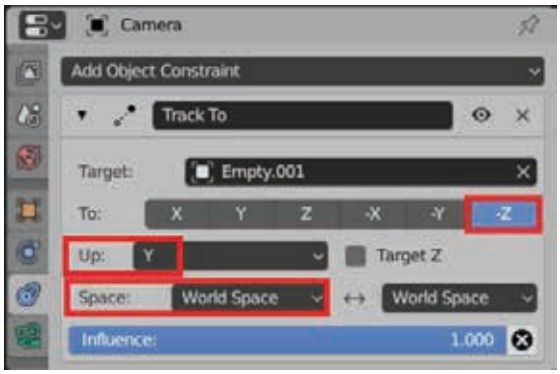
Now the camera will point wherever the Empty is.



Select the camera, go to **Object Constraints** and choose a **Track To** constraint.



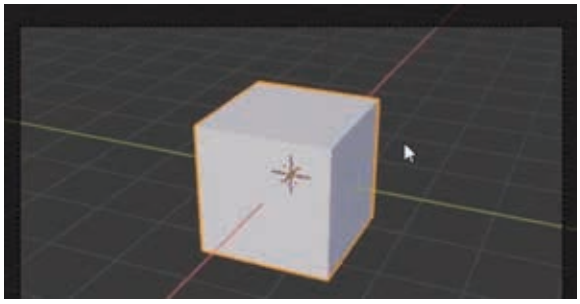
To fix this, make sure To is set to **-Z, Up is set to Y** and Space is set to **World Space**.



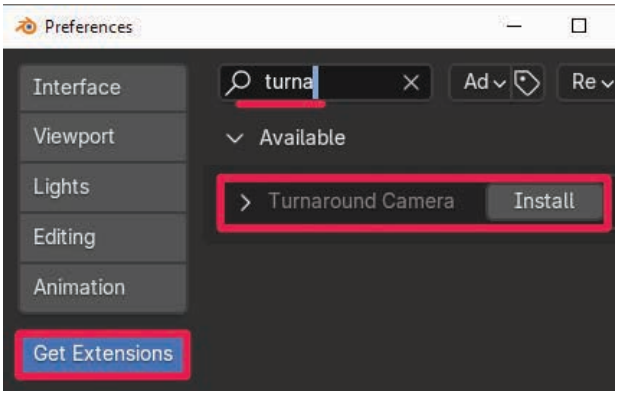
If you want to make sure it not only follows the Empty location but also its rotation, turn on **Target Z**.

Camera turnaround

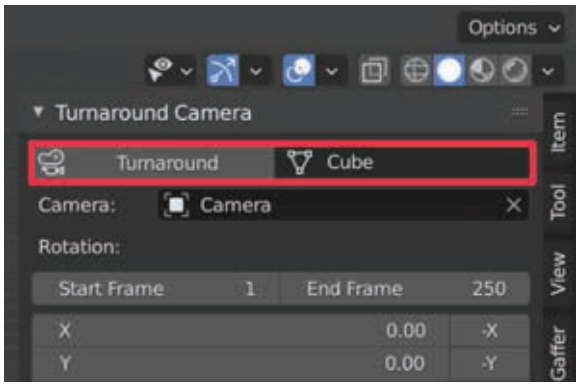
Creating a camera turnaround animation is important for showing off our finished models, but it's kind of a pain to set up.



But it becomes a lot easier when you install the **Turnaround Camera add-on** in Preferences (in the Get Extensions tab).

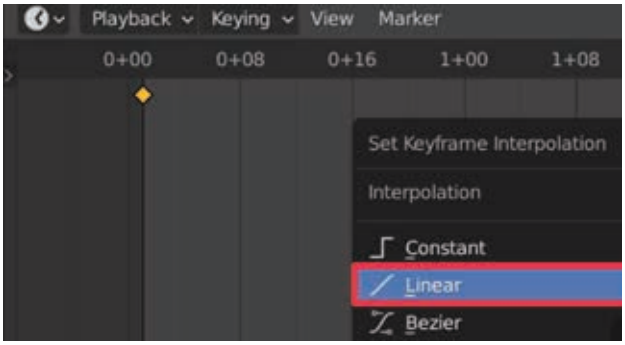


The **Option panel** now features Turnaround Camera options inside the **Animate** tab. Select an object and click on **Turnaround.**

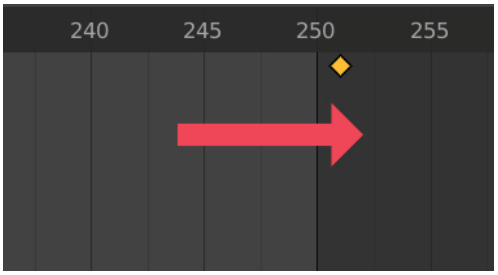


Now if you **press Spacebar**, the camera will start to rotate around the object. Notice that the camera movement is not constant, it slows down at the end. To fix this, first select the **Rotation target** in the **Outliner**.

Set the keyframes to Linear in the Timeline by **pressing T** and choosing **Linear**.

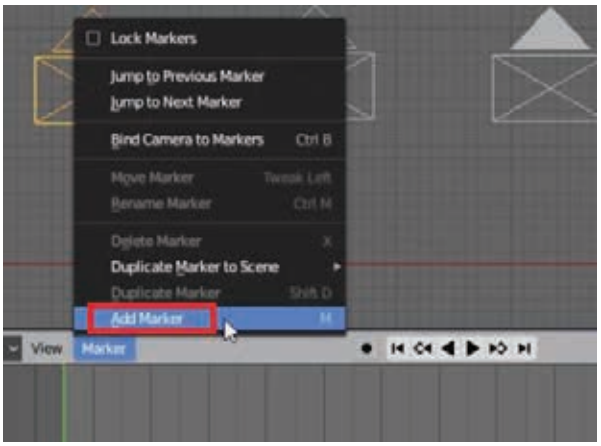


Move the last keys one frame past the last frame by **pressing G+1**. Now it loops perfectly.

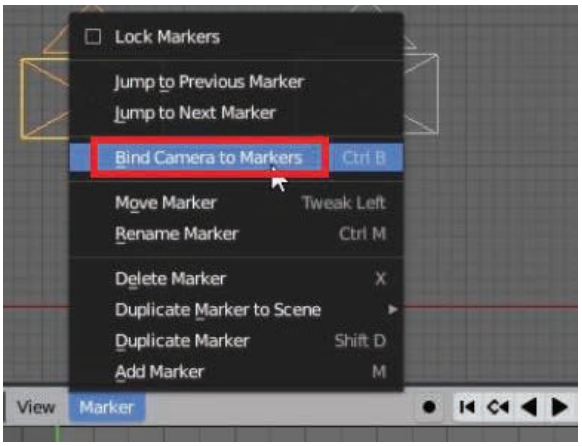


Camera switching while rendering using markers

Select the camera and in the Timeline add a Marker.



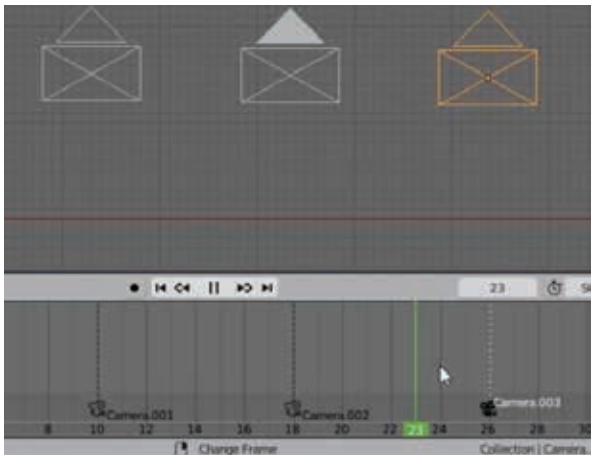
After that select Bind Camera To Marker. Alternatively **press M** to make a marker and **Ctrl+M** to bind to camera.



This way you can switch between cameras in one file. Whenever the frame is reached where there is a marker, Blender will automatically switch to the camera that has been bound to that marker.

This is useful if you want to upload to a render farm.

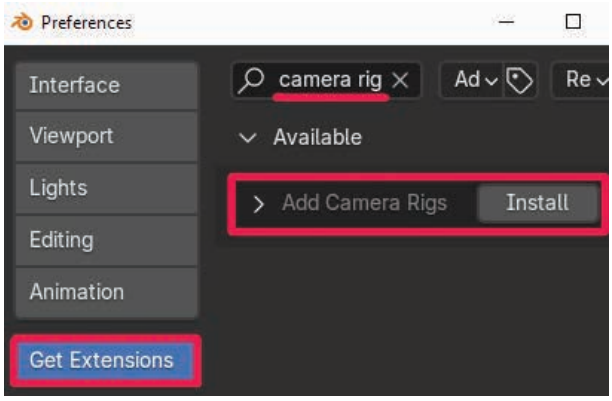
That way you don't need to upload a file for each camera.



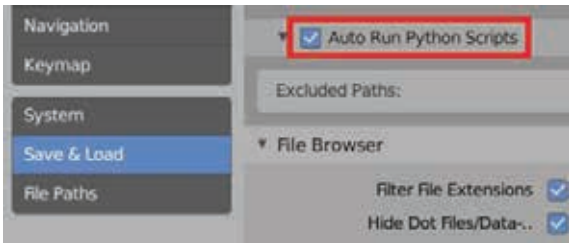
Click here to return to the index.

Camera crane rig add-on

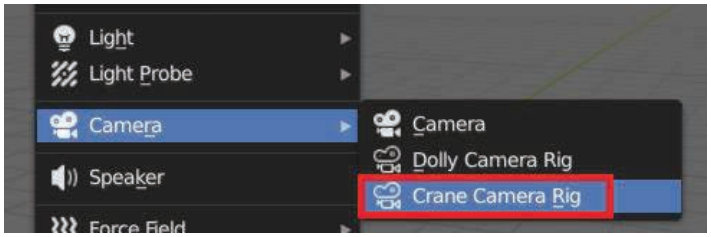
Install the **Add Camera Rigs** add-on in Preferences under Get Extensions.



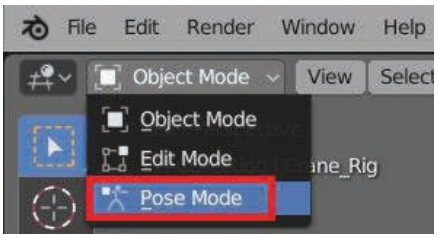
Under Save & Load in Preferences, turn on **Auto Run Python Scripts**.



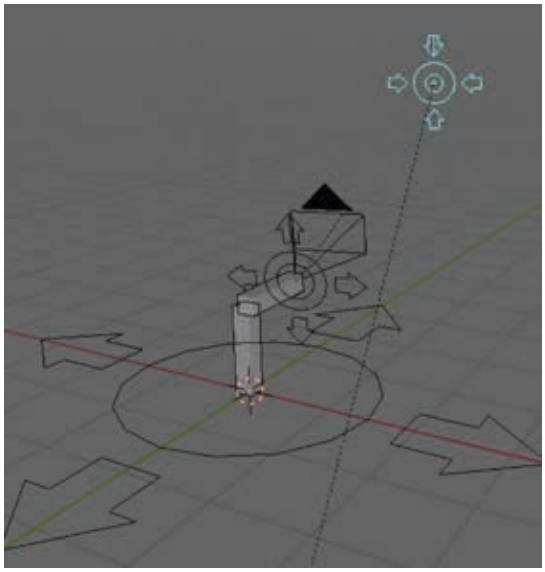
Add the Camera Rig with **Shift+A** > Camera > **Crane Camera Rig**. You can also add a Dolly Camera Rig.



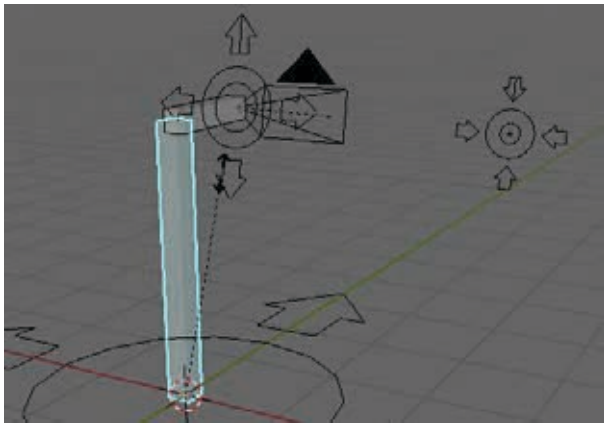
You need to be in **Pose mode** to use the camera rig.



This is the Crane rig in **Pose mode**. The light blue part is the target, which is what the camera will always point at.

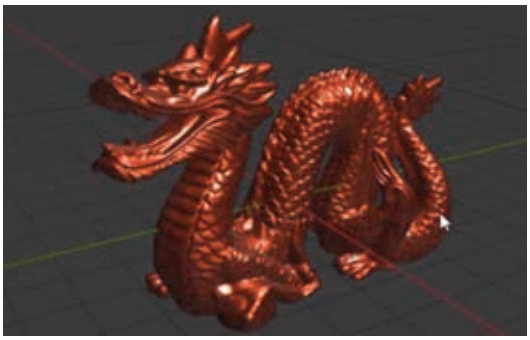


You can transform the individual parts for different kinds of camera moves by **pressing S, G and R** for scale, move and rotate. **Press i** to set a keyframe at a point in the timeline to animate the camera movements.

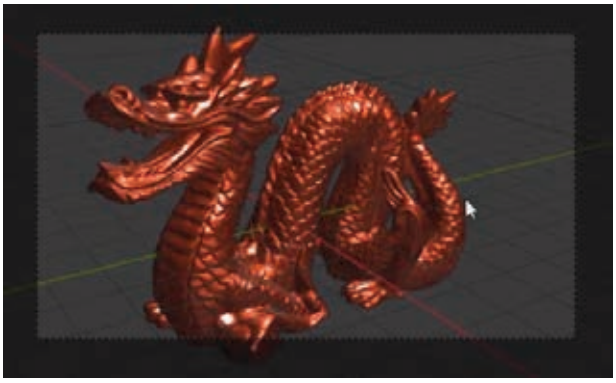


Set Viewport to Camera View

To change to a camera view from the same perspective press **Alt+Ctrl+0** (numpad zero).

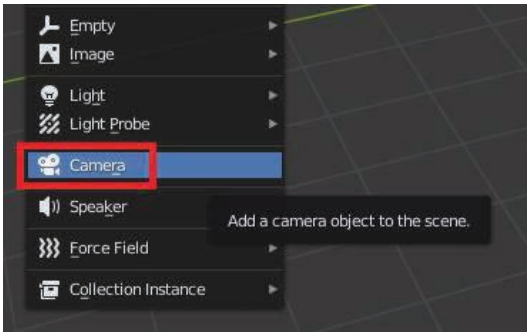


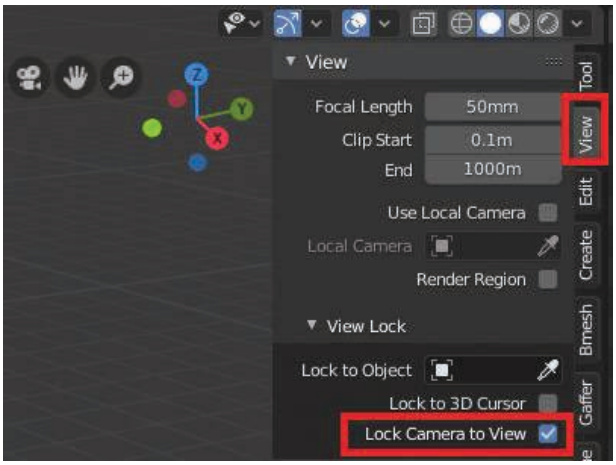
Note that this only works when there's at least one camera in the scene.



Check **Lock Camera to View** to control the camera when using the normal viewport navigation methods.

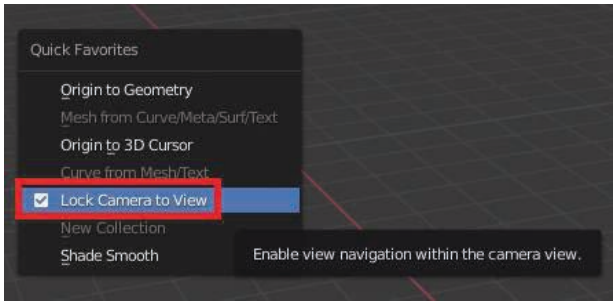
To add a camera, **press Shift+A**, choose Camera.





Did you know you can add checkboxes to the Quick Favorites?

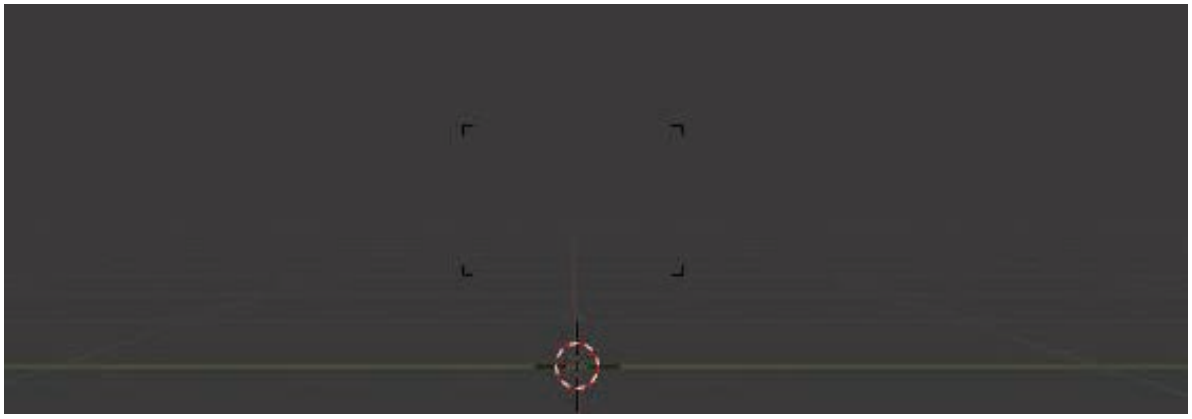
Lock Camera to View is something I use often, so I've added the checkbox to my Quick Favorites menu.



Fly Mode

FlyMode is a videogame-like mode of moving the viewport "camera" in Blender. It can be especially handy when working with a large interior scene, such as an architectural visualisation.

Activate Fly Mode by **pressing Shift+F**. You will see a kind of "target" display in the center of the viewport and it will point at where you move your mouse.



Using the scroll wheel on your mouse to scroll up, you'll start to fly forward. Scrolling down will stop and eventually make the viewport reverse.

The **Q and E** keys make the viewport move up and down (press repeatedly to increase the speed). The **A and D** keys make the viewport move left and right (press repeatedly to increase the speed). You can also use the **arrow keys** on the keyboard to move left, right, forwards and backwards.

Clicking anywhere with the **left or right mouse button** will exit Fly Mode.

Here's a handy reference:

Shift+F	Initiate Fly Mode
A or Keyboard ArrowLeft	Move left
D or Keyboard ArrowRight	Move right
Q or Keyboard ArrowDown	Move down
E or Keyboard ArrowUp	Move up
Right or Left MouseButton	Exit Fly Mode

RENDER TIPS

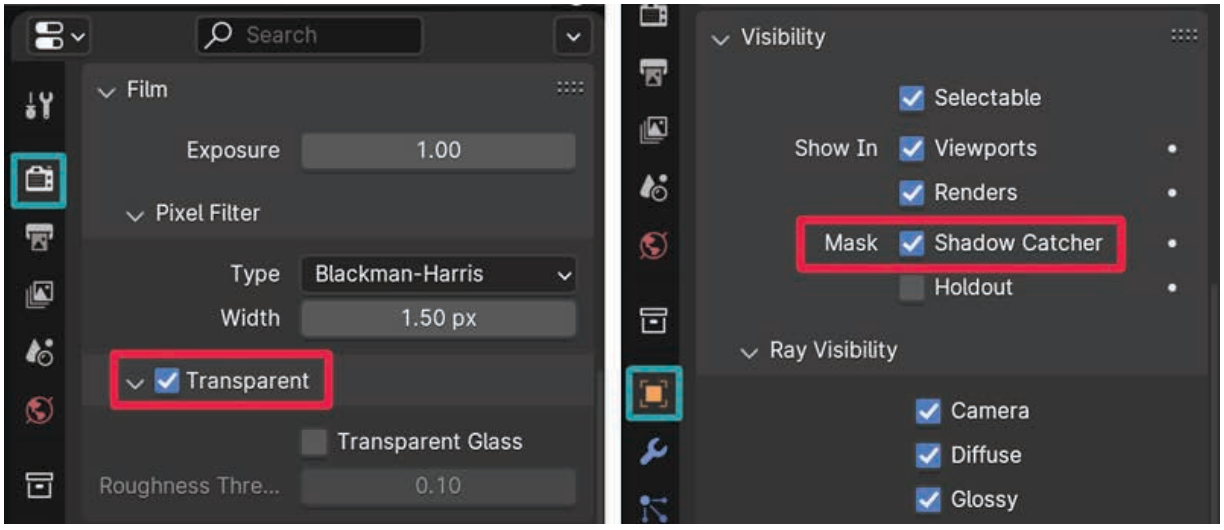
Faster Render Times

If you have static shot where the background doesn't change, you can save a lot of time by rendering the background separately.

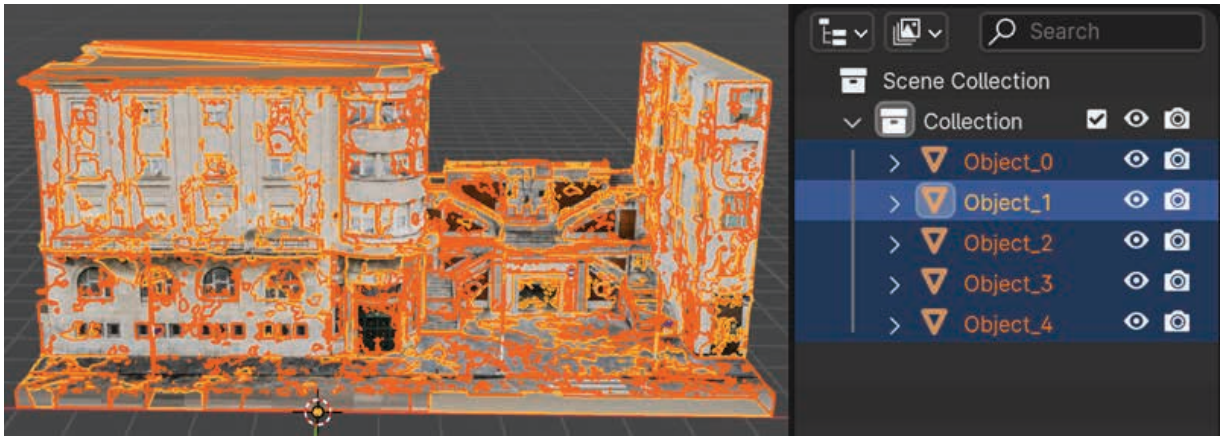
Robot by JoseDiaz, background by Miguel Bandera, both from Sketchfab.

I recommend that you render an **image sequence** for your foreground animation and a **still render** of the background.

To do this, under Film choose "Transparent". Select the background and under Visibility check Shadow Catcher. If you don't have this option, make sure you are in **Cycles** and not Eevee.

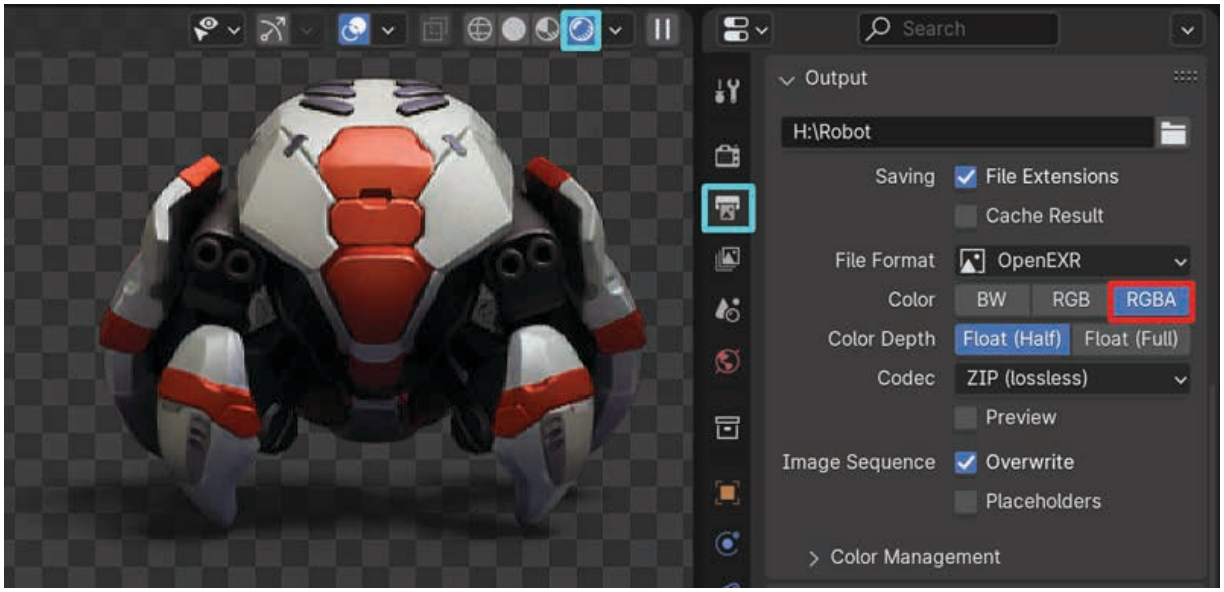


In case you have multiple background objects selected, **hold Alt** when clicking the checkbox for the Shadow Catcher option to get them all at once.



Now you can render the foreground object, including its shadow, as an image sequence. You can see the transparency already in **Rendered preview mode**.

Use an Output format that supports transparency with an alpha channel.

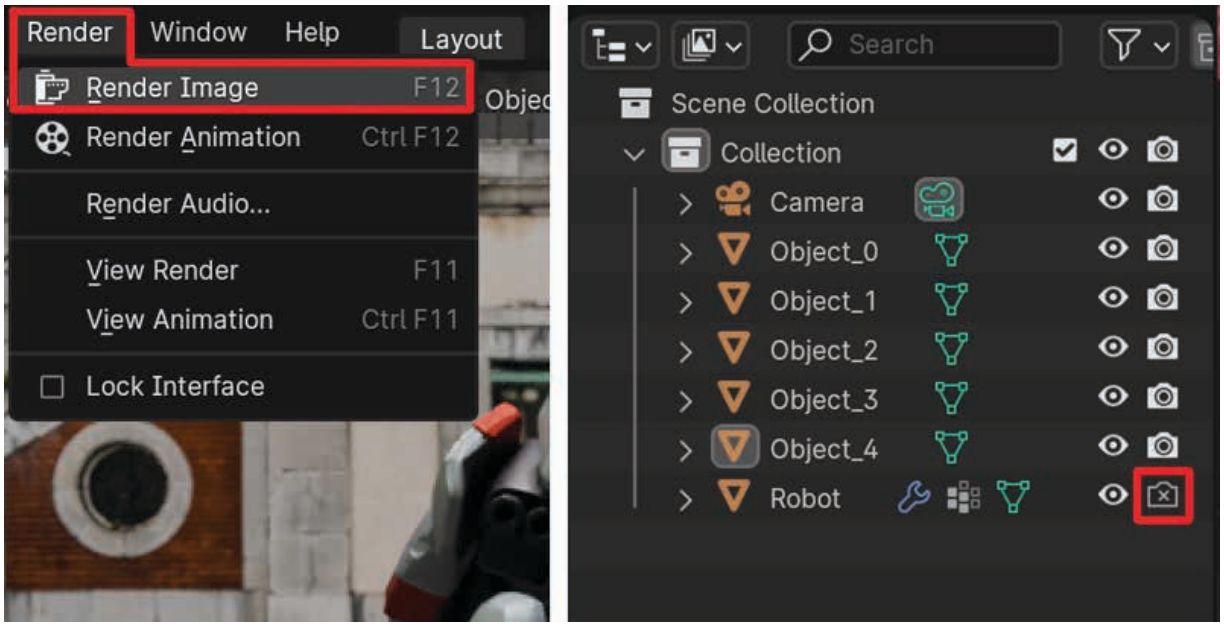


The robot's and its shadow are visible, but the background is transparent.

After that, turn off Shadow Catcher for the background object(s).

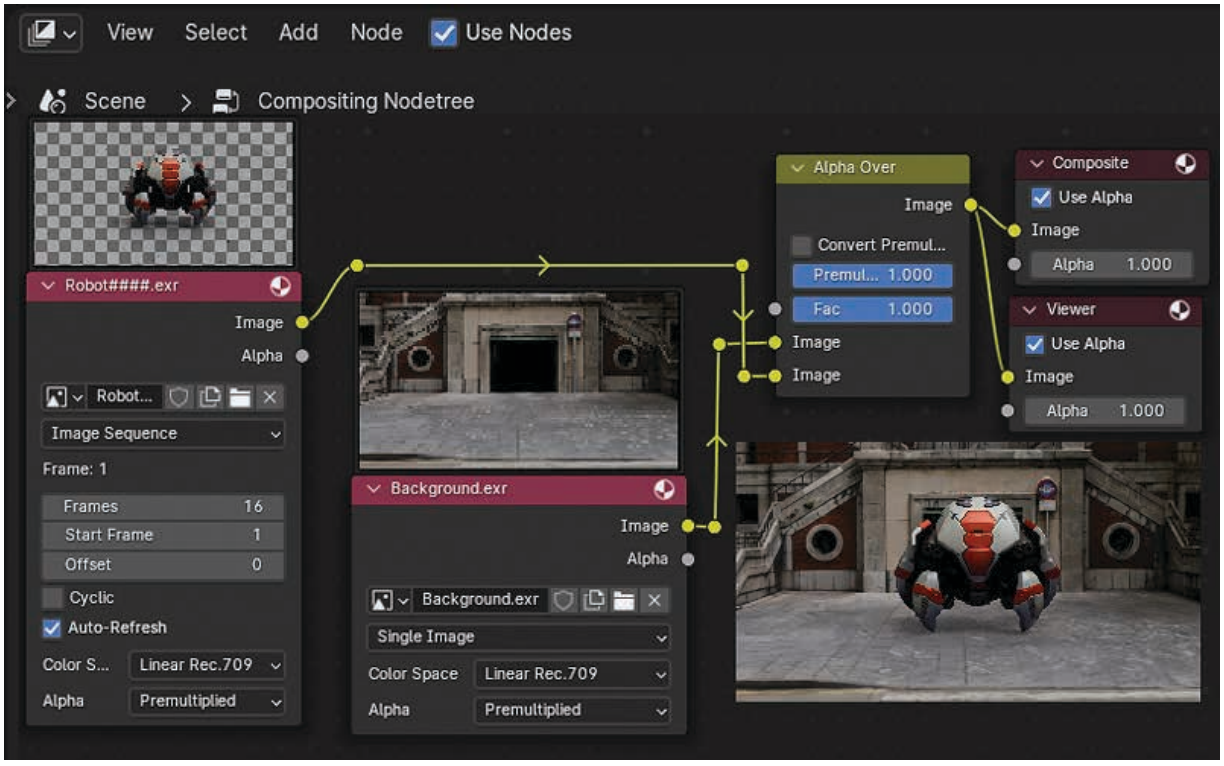
Disable **render visibility** (the camera icon) on the foreground object in the **Outliner**.

And render a still image of the background by **pressing F12** or going to Render > Render Still Image.



Finally, after the background and foreground are rendered separetely, combine them in the **Compositor**.

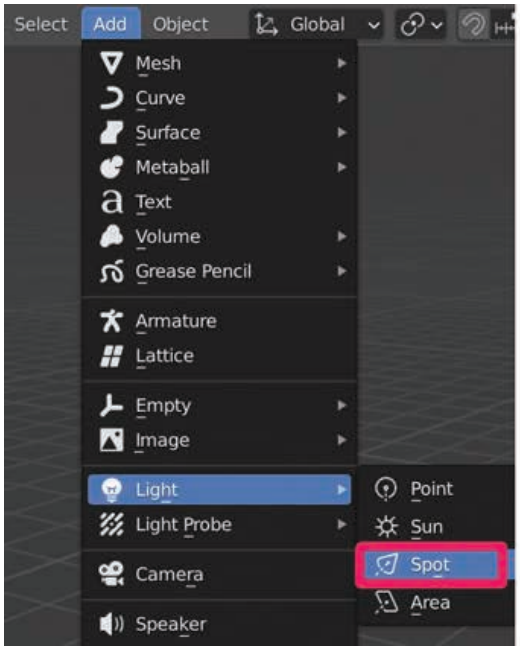
Use an Image Sequence node for the foreground element, and an Image node for the background. An **Alpha Over node** combines them.

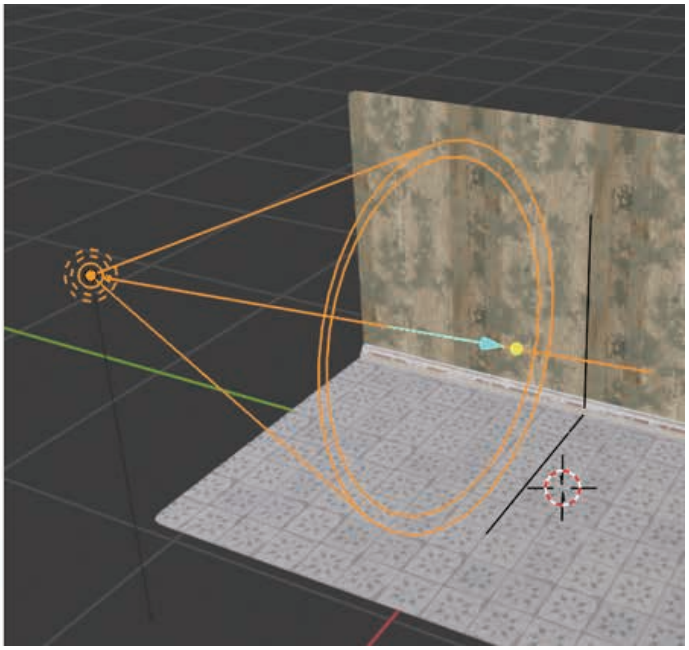


Added realism with Textured Lights

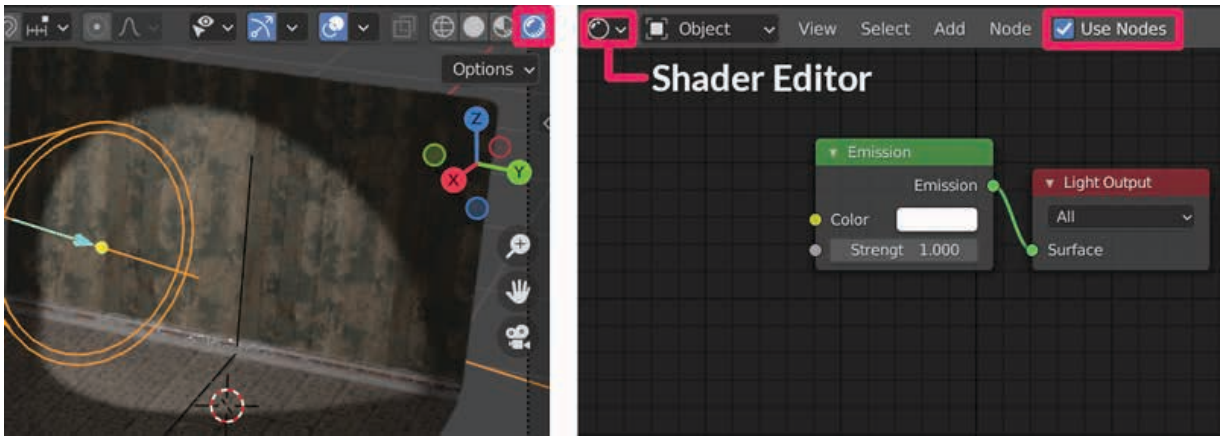
CG art often looks too perfect and clean. That's why we try to make materials look worn and sometimes even dirty, to achieve realism. The same goes for lights: if they're too perfect, they don't contribute to realism. One way to make lights more interesting is to add textures to them.

Add a Spotlight to your scene.

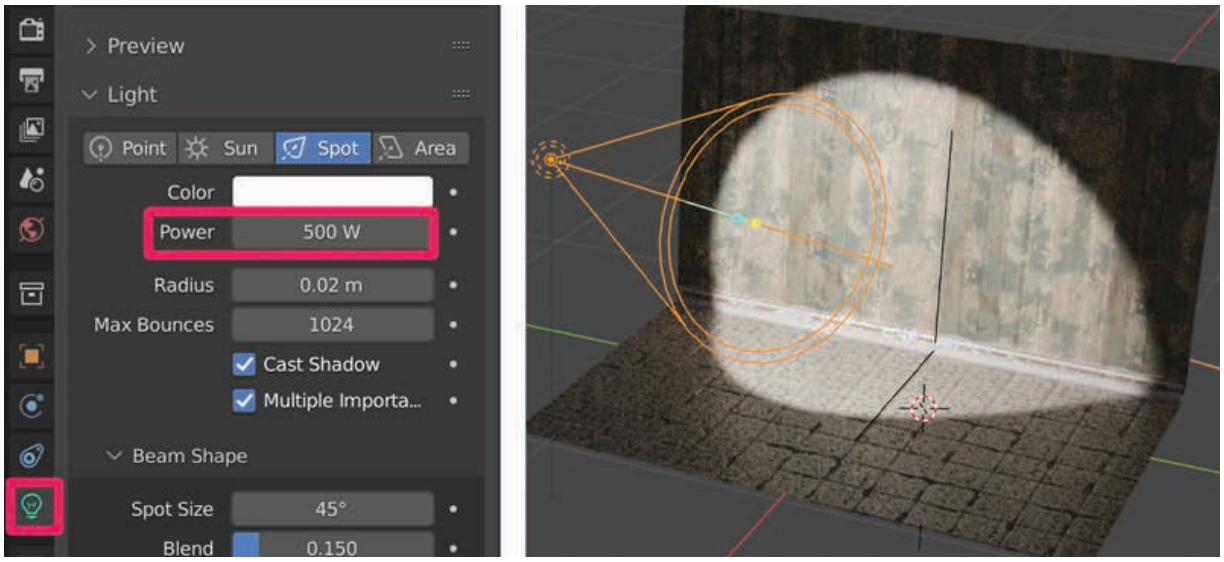




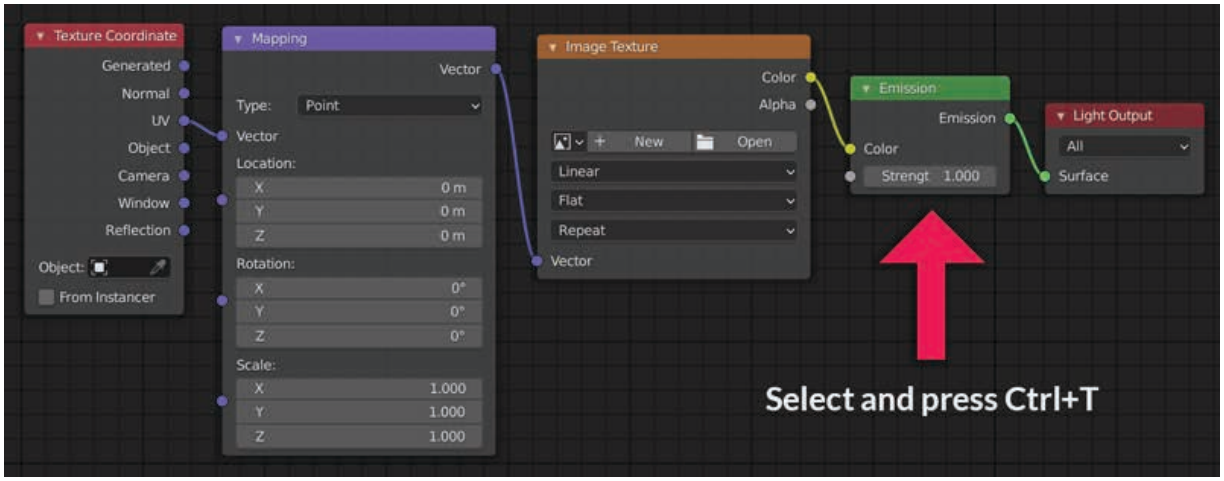
In the Shader Editor window enable **Use Nodes**. Switch to **Rendered View** to see the lighting in the viewport.



Increase the **Power** of the Spotlight so that you can see it affecting the scene. Don't be afraid to use a high value like 500 or even 1000. After plugging a texture in the light, it will get slightly dimmer.



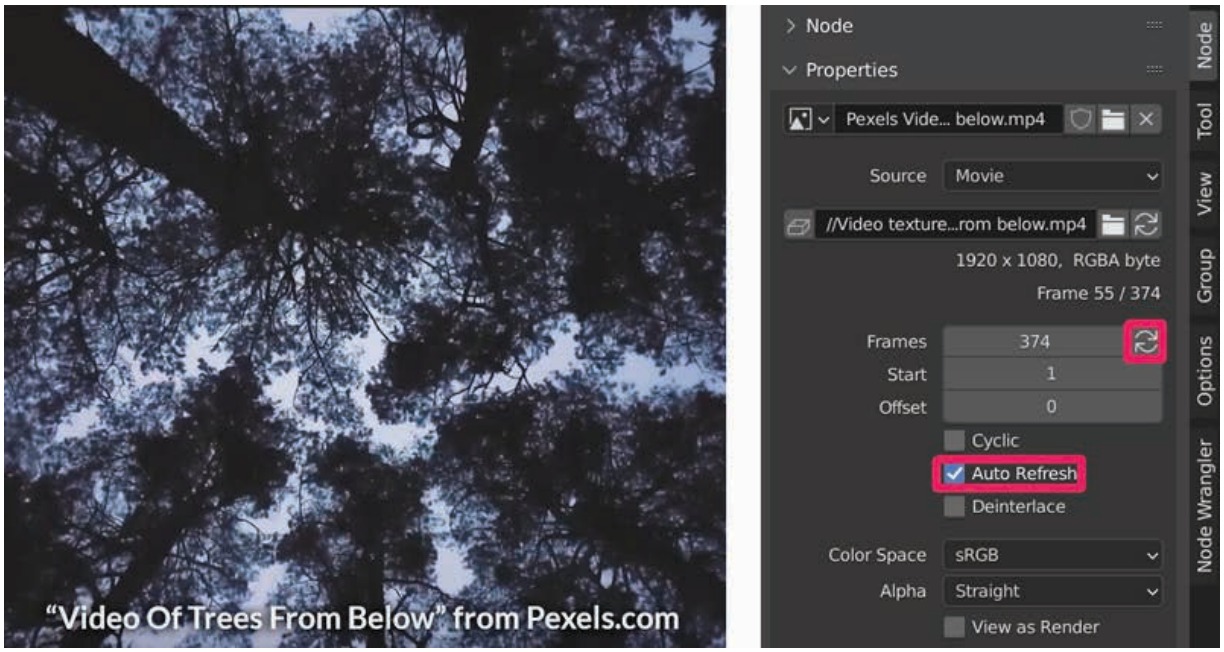
Select the light, and add a texture to it in the Shader Editor window by **pressing Ctrl+T** with the **Emission** node selected. (Note that for this keyboard shortcut to work, the Node Wrangler add-on needs be be enabled in Preferences).



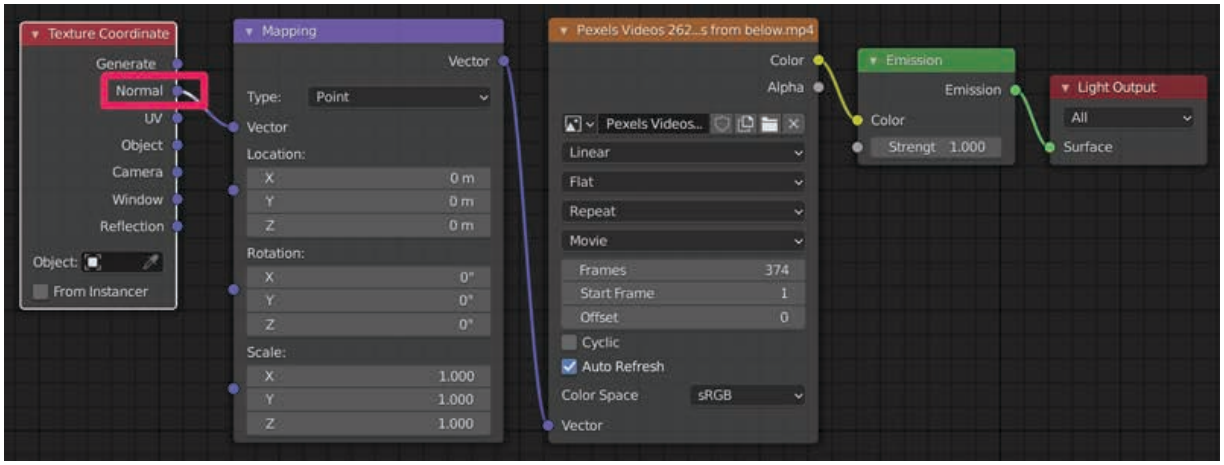
You can also use a video as a texture. In this case I'm using a video of some slowly moving autumn leaves (link to the video below this article). In the Shader Editor window, open the **Option panel** by **pressing N**.

Then open the **Node panel** and under Properties click on the **Match Movie Length** icon next to the Frames value. This way, all the frames of the video will be played instead of just a static image. Enable **Auto Refresh** as well.

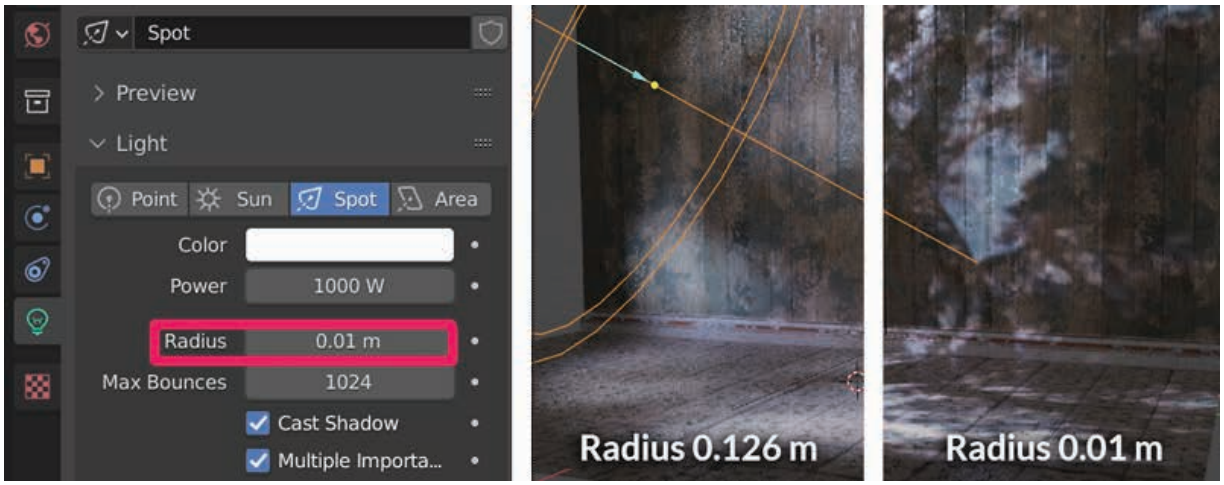
Note, that the Image Texture node doesn't have this Match Movie Length button, you can find it in the Properties panel in the Shader Editor window, which you can toggle on or off by pressing N. "Auto Refresh" won't work until you've clicked Match Movie Length.



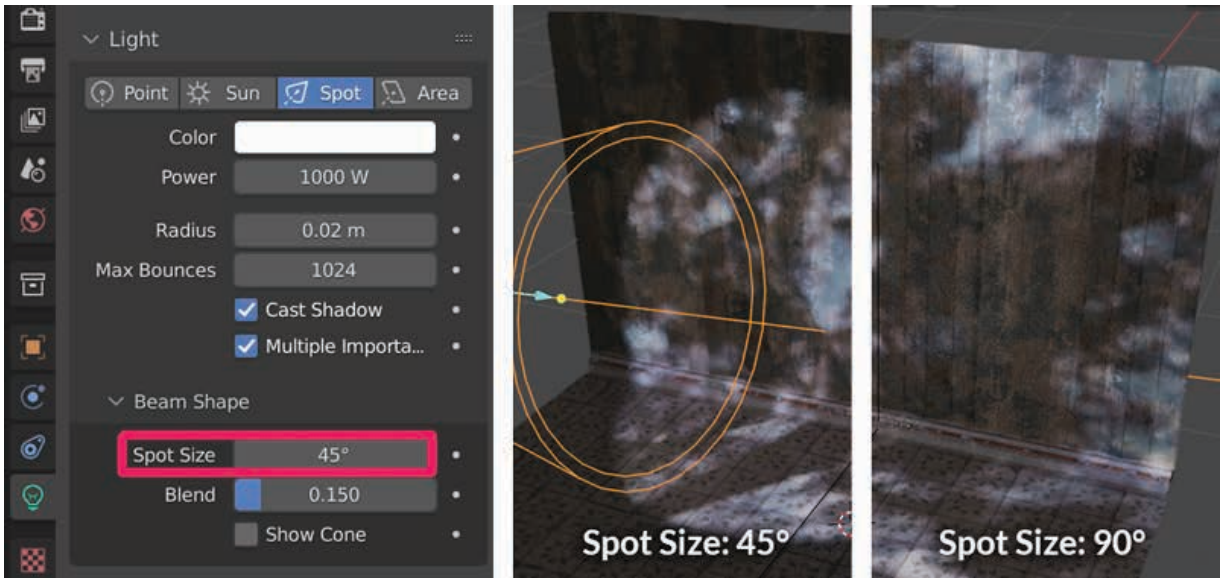
Use the **Normal** texture coordinate. If you're using **Cycles**, you will now see the textured light.



To make the texture more blurry, increase the Light > **Radius** value (or decrease it to make the texture sharper).



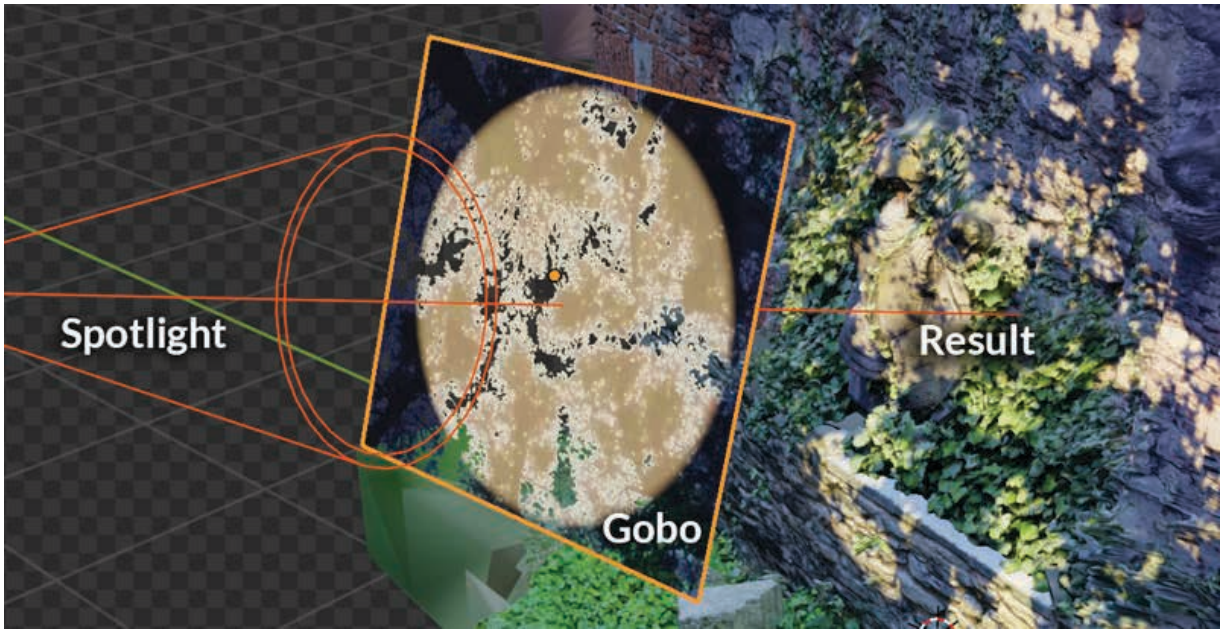
To make the texture cover more area, increase the Beam Shape > **Spot Size** value.



Video Of Trees From Below from Pexels.com
https://www.pexels.com/video/video-of-trees-from-below-853839/

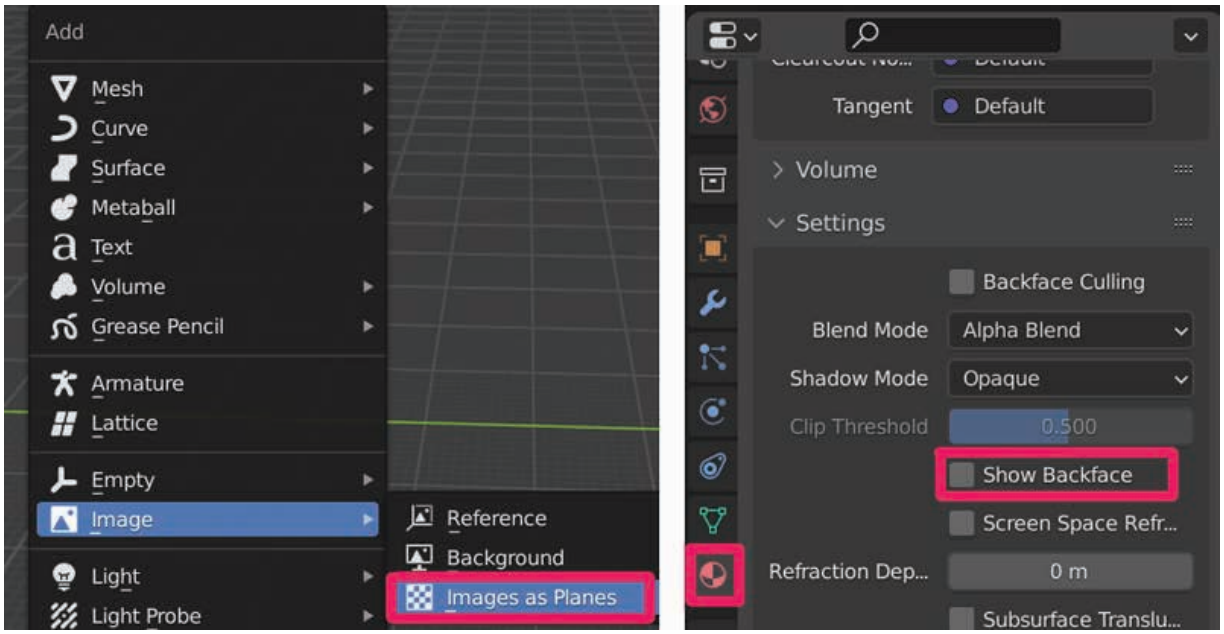
More Realistic Renders in Cycles and Eevee using Gobos

Another way of adding realism and interest to lights besides light textures, is using **gobos**. Gobos are just black and white alpha images, where the white lets lights through and the black parts block light. This way you can create the illusion of tree leaves casting shadows, for example.



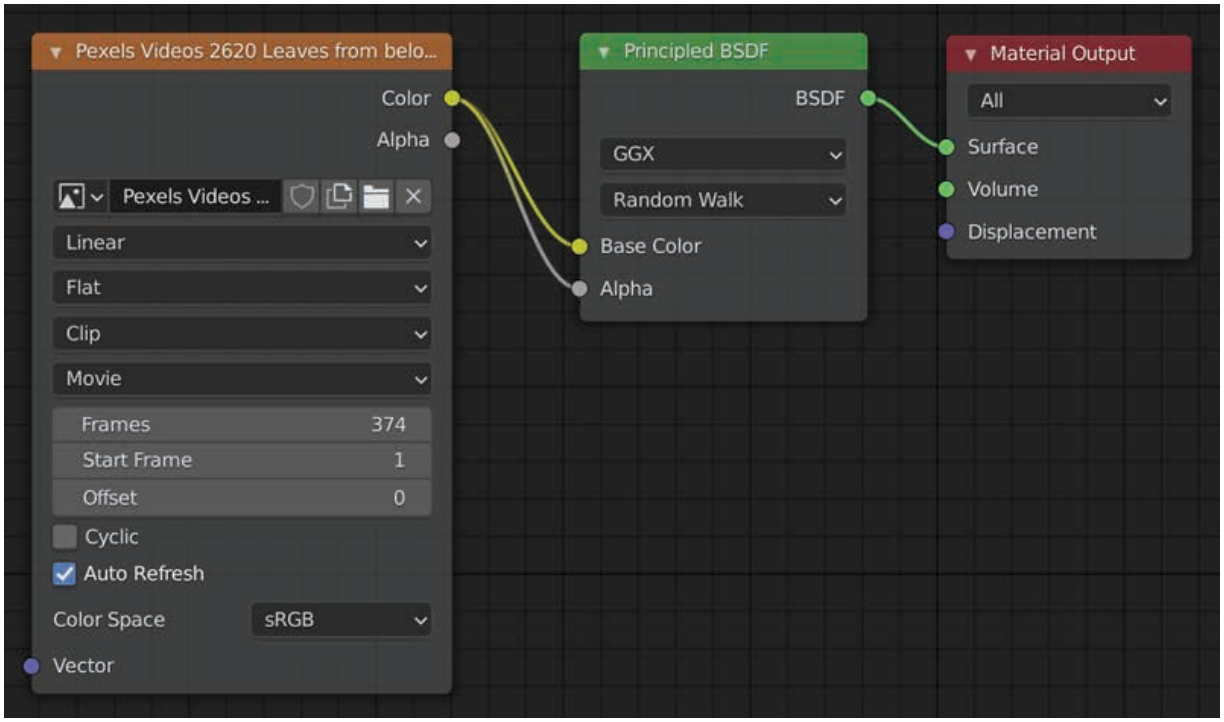
The benefit of gobos, is that you can also use them in **Eevee**, which doesn't support the previous method of textured lights.

Import the image texture or video you want to use as a gobo using the **Import Images as Planes** option (for this the Images as Planes add-on needs to be enabled in Preferences). In this case I'm using the same leaves video as from the last video. In the Material settings, uncheck **Show Backface**.

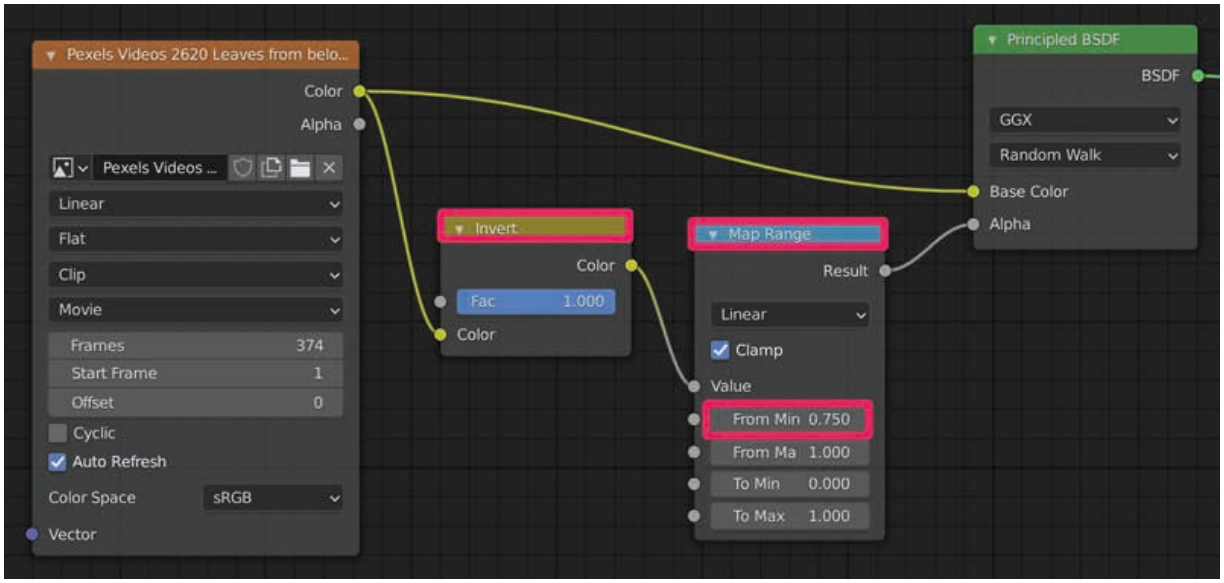


By default Blender assumes the image or video has an **Alpha channel** and connects the nodes with that in mind. However in this case I'm using a video without an Alpha channel.

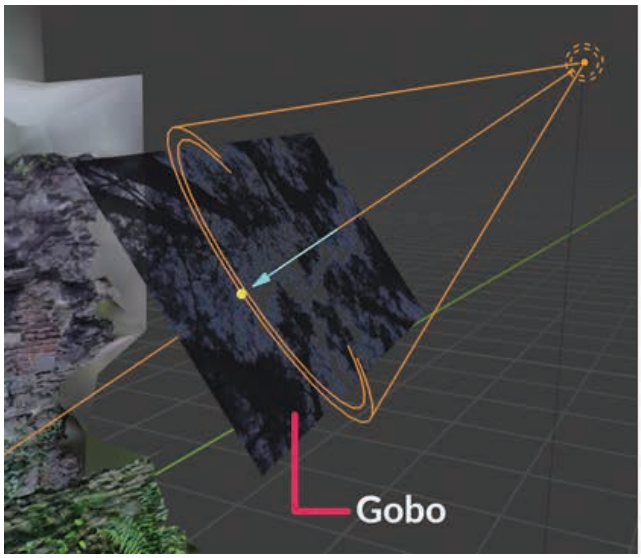
I do however want to have an alpha channel so I'm just going to pretend to have one by plugging the **Color output** of the video into the **Alpha input** of the **Emission** shader. *(Note that for screenshot purposes, I've minimized the Principled BSDF node)*



However, now the wrong part of the image is transparent. So add an **Invert node** to invert that and make the right part transparent. Also increase the contrast using a **Map Range** node, with the **From Min** value increased until it looks like there's enough really transparent parts. In this case the value is 0.75. You could replace the Map Range node with a **ColorRamp** node if you're more familiar with that.



Add a **Spotlight** if you haven't got on in the scene yet. Make sure your Spotlight is strong enough to see the effect of the light and the gobo. Note that gobo's don't work with a Sunlight. Manually position the gobo so that it's in front of the Spotlight by **rotating it with R and scaling it with S**. And of course make sure that it's outside of the camera view.

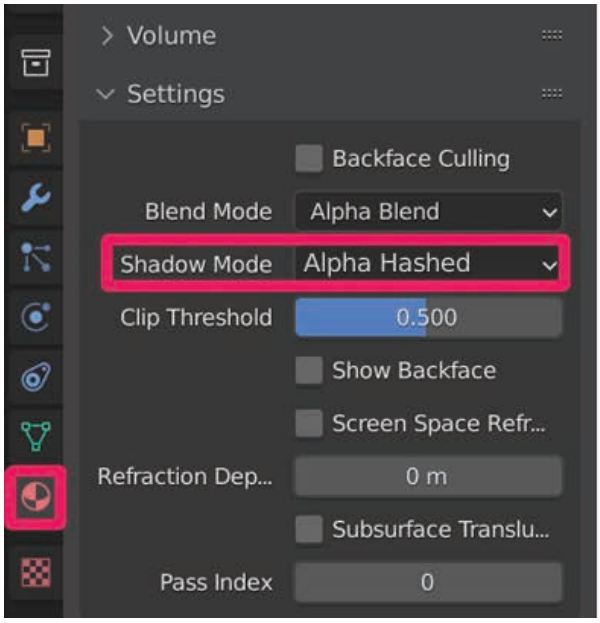


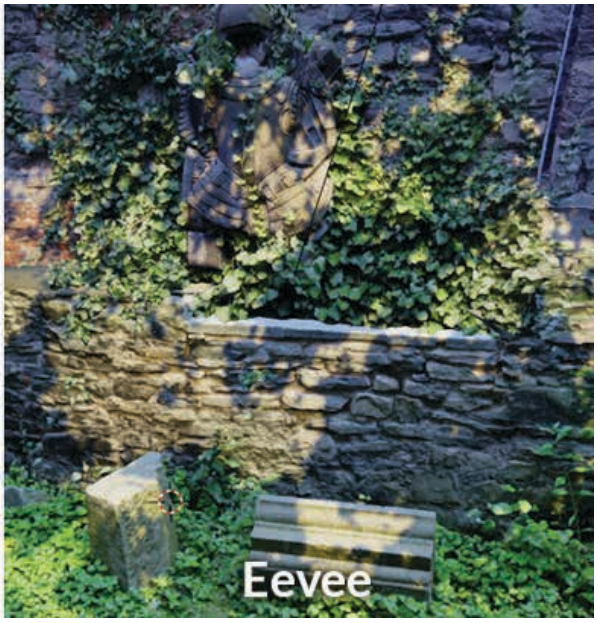
You may need to really crank up that light intensity because the gobo is blocking a lot of it. Now you've got some subtle leaves shadows on your models. You can also use an HDRI for some overall light, and then use the Spotlight to add some warm light by giving it a bit of an orange tint.

Just like with the previous topic about Light Textures, to make the texture more blurry, increase the **Light > Radius** value (or decrease it to make the texture sharper).

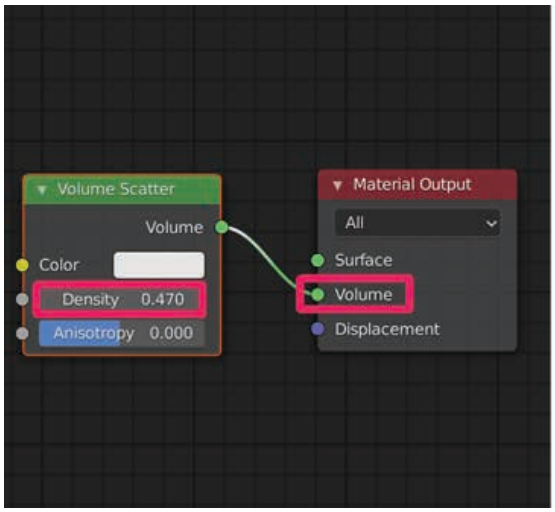
To make the texture cover more area, increase the **Beam Shape > Spot Size** value.

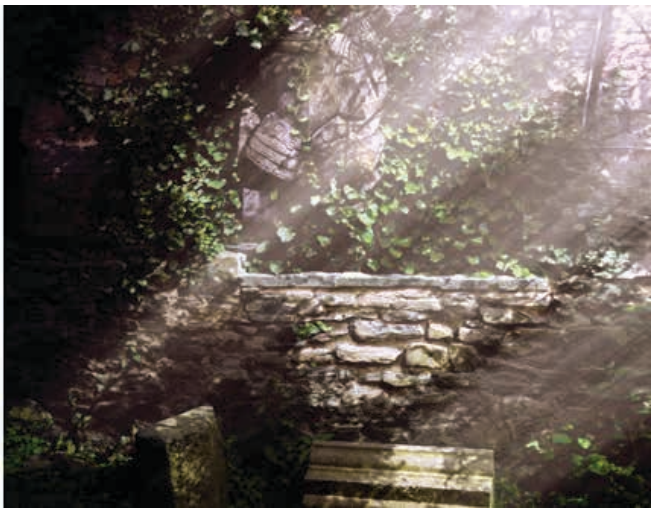
In **Eevee** to see the effect of the Gobo you'll need to select the gobo object and in its Material properties, set **Shadow Mode** to Alpha Clip or **Alpha Hashed**. Both options have a slightly different look, so choose what you prefer. In my opinion, Alpha Hashed looks better.





To add a Volumetric effect to the light, simply add a Cube around your scene (not too big, just big enough to envelop what's seen in camera) and add a **Volume Scatter node** to the Cube material (delete the Principled BSDF). Remember to plug it into the **Volume input**! Then reduce the **Density** until it looks good. This works the same for Cycles and Eevee.

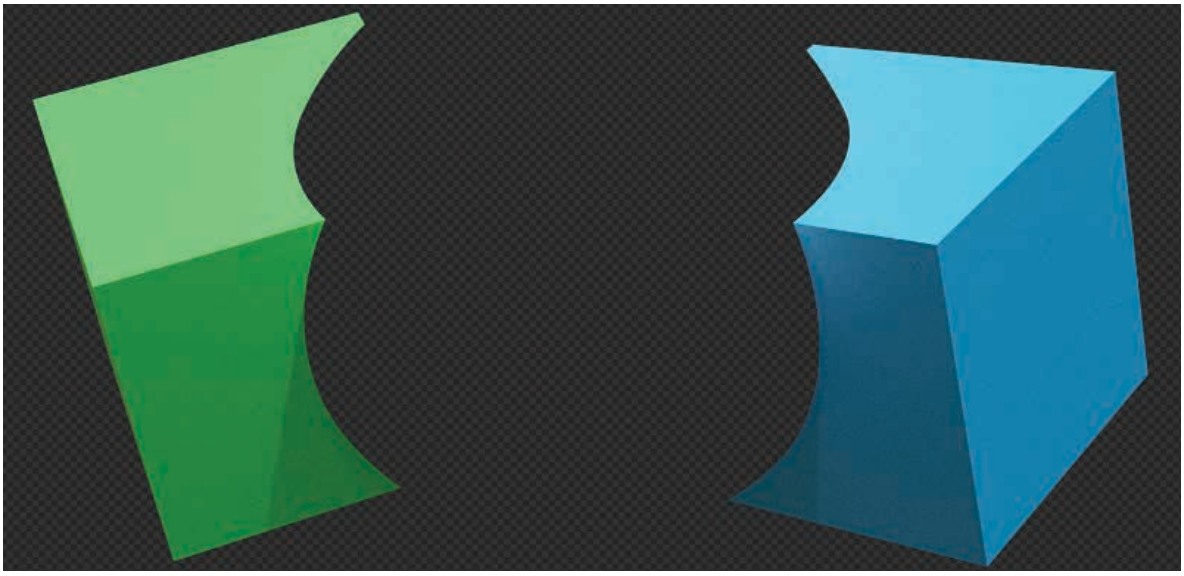




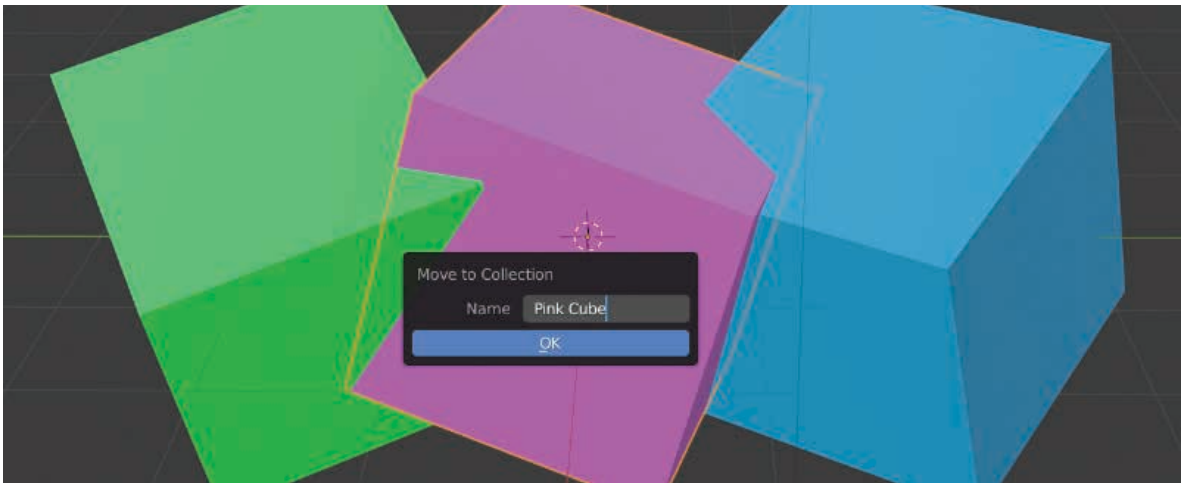
Video Of Trees From Below from Pexels.com
https://www.pexels.com/video/video-of-trees-from-below-853839/

Create masks with Holdout collections

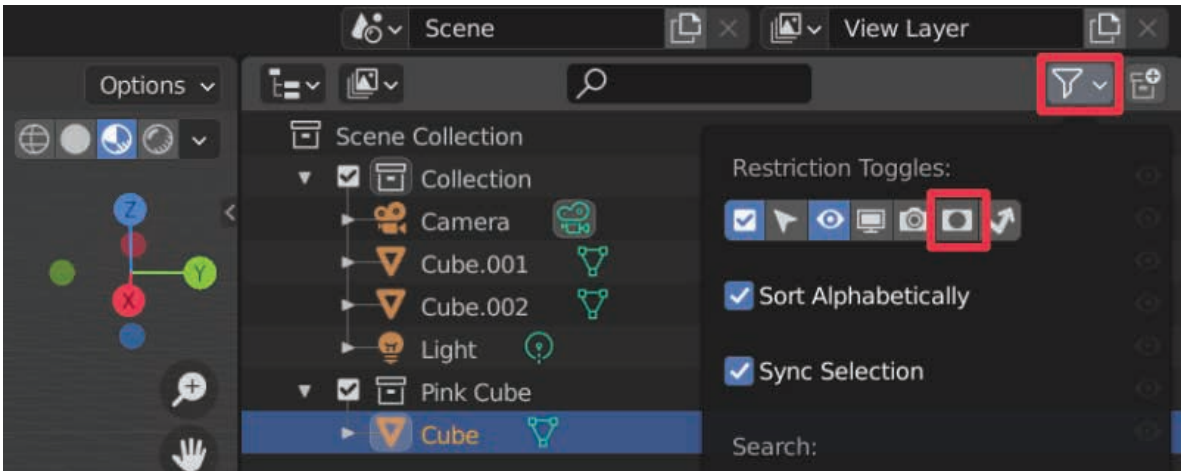
A holdout is an object that acts like a mask, creating transparency. Below you see the result of a Sphere inside a collection set to Holdout. It cuts a hole in the other cubes, a bit like a Boolean would, except it's a render effect rather than a modeling effect.



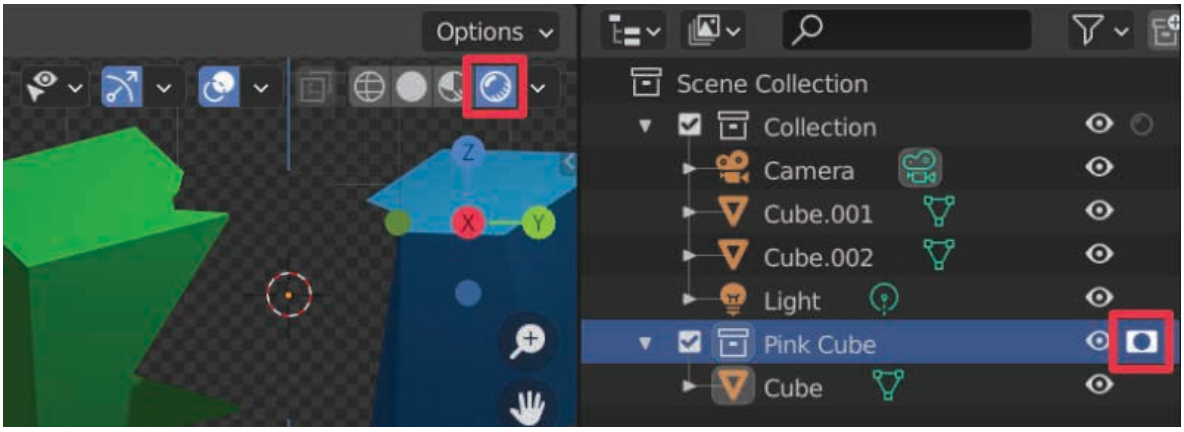
Select the object you want to use and **press M** to add it to a collection. Make sure to always name your collections clearly, to avoid later confusion.



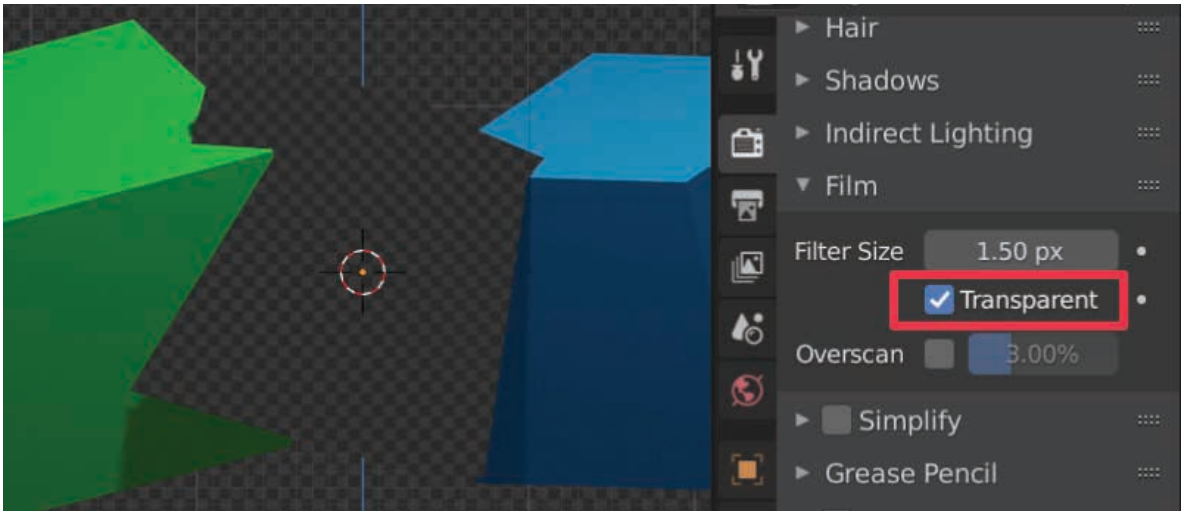
Go to the **filter** icon in the **Outliner** and make the Holdout icon visible.



Then click on the Holdout icon for the Collection you want to use as a mask. You'll also need to turn on Rendered Viewport Shading to see the effect of the holdout. It works in both Cycles and Eevee.

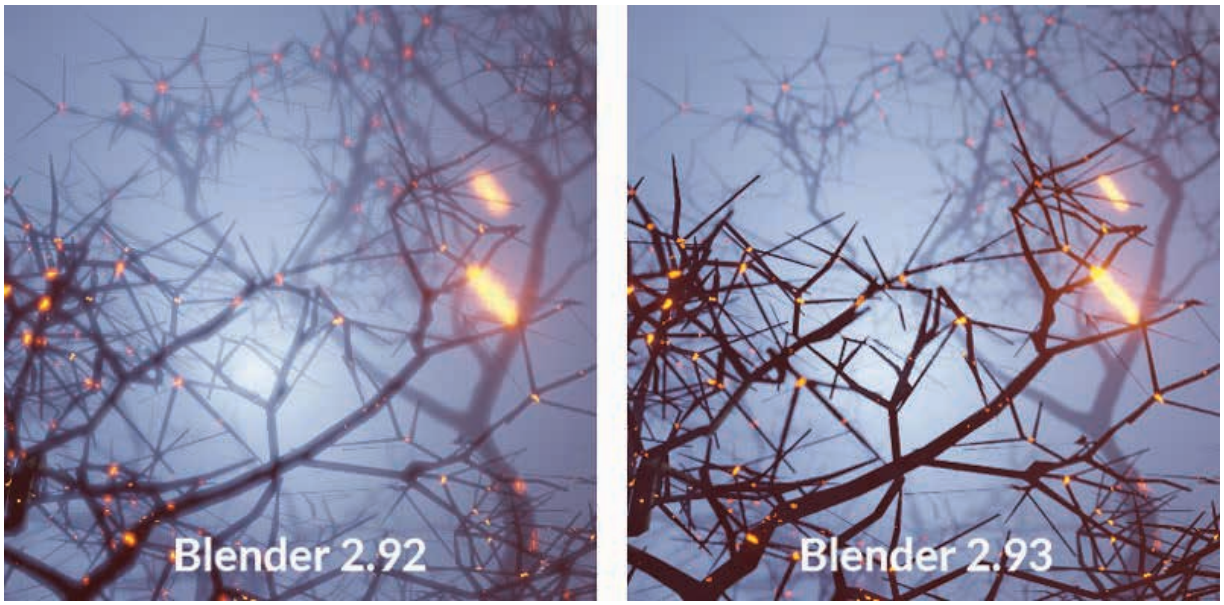


It doesn't work until you turn on Transparent under the Film options.



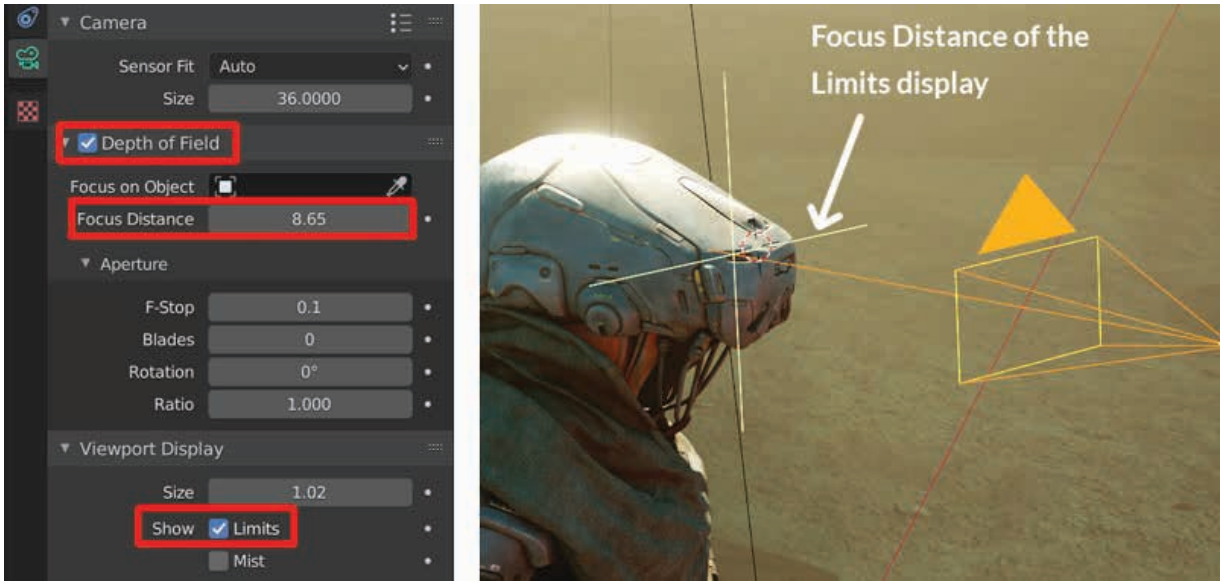
Shallow Depth of Field in Eevee

One of the improvements that Blender has gotten since version 2.93 (the current stable release) is in the way **shallow focus** (or shallow **depth of field**, the look of having a short distance between the nearest and farthest object in focus) looks in Eevee. Here we can see a quick comparison between Blender Eevee in 2.92 and 2.93:



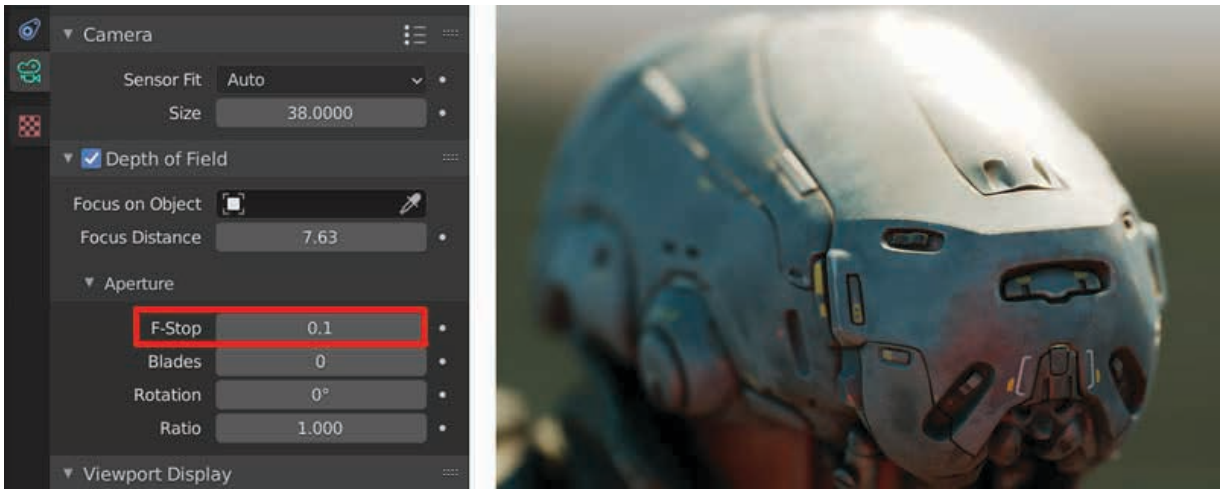
Forest scene by Mike Pan

As you can see it handles highlights, close-up blurred objects and outlines much better than before. The way to create shallow focus hasn't changed. First of all, you'll need a **camera** to see through. Then, you need to enable **depth of field** in the Camera settings. Then there are two ways to control the depth of field, first is by showing the **Limits** and settings a **Focus Distance**.

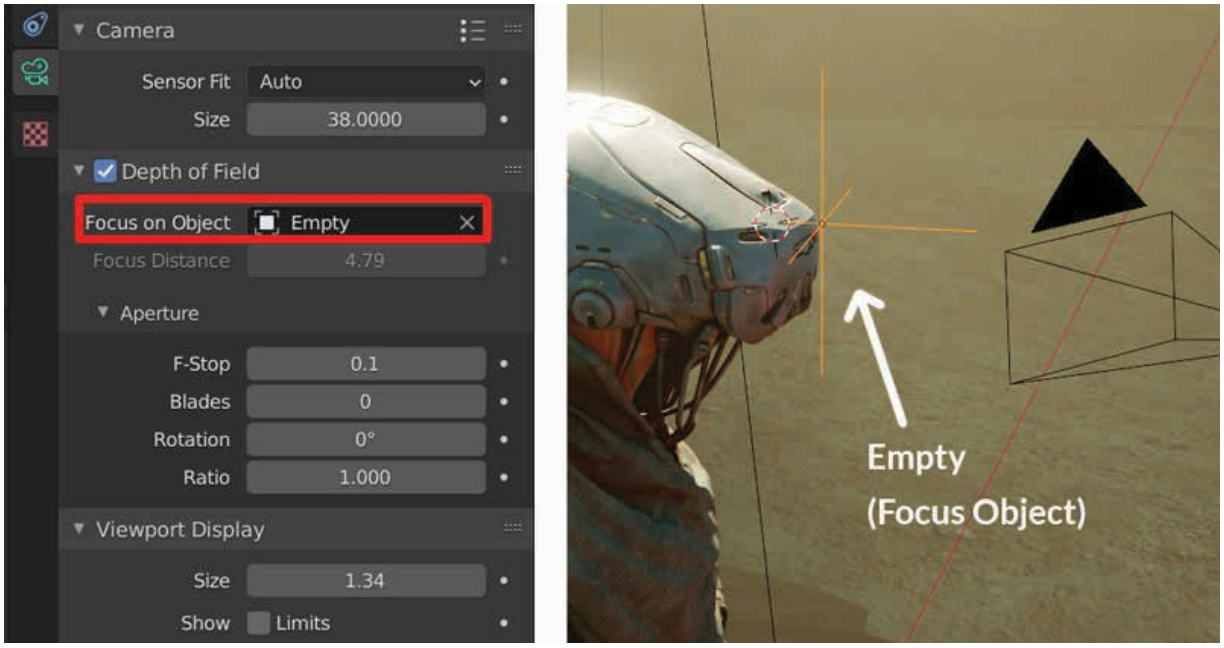


The Wanderer scene by Danield

To control how shallow the depth of field is, you can adjust the **F-Stop value**. It's best to have the objects in your scene be at the correct scale, so you don't need to enter a really low value like in this case (the robot here is about 10 metres high).

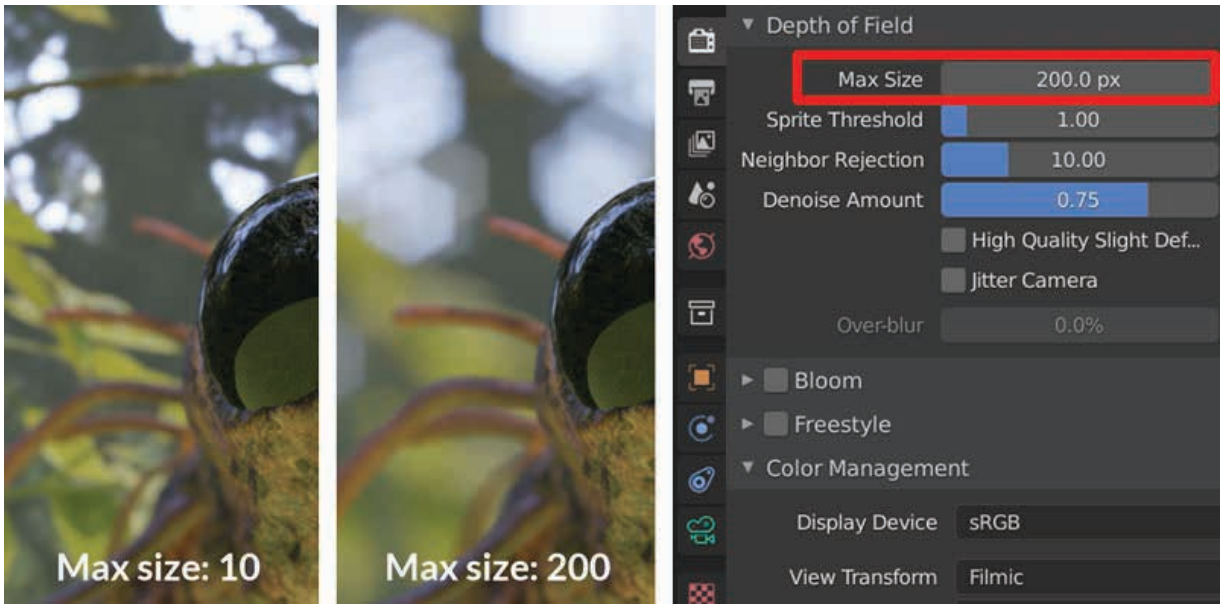


If you plan to animate the camera position I'd recommend using a second method, which is by placing an **Empty** where you want to focus, and choosing that Empty in the **Focus on Object** field. This is a bit more easy to control, as the Empty is not constrained to the camera like the Focus Limit is.



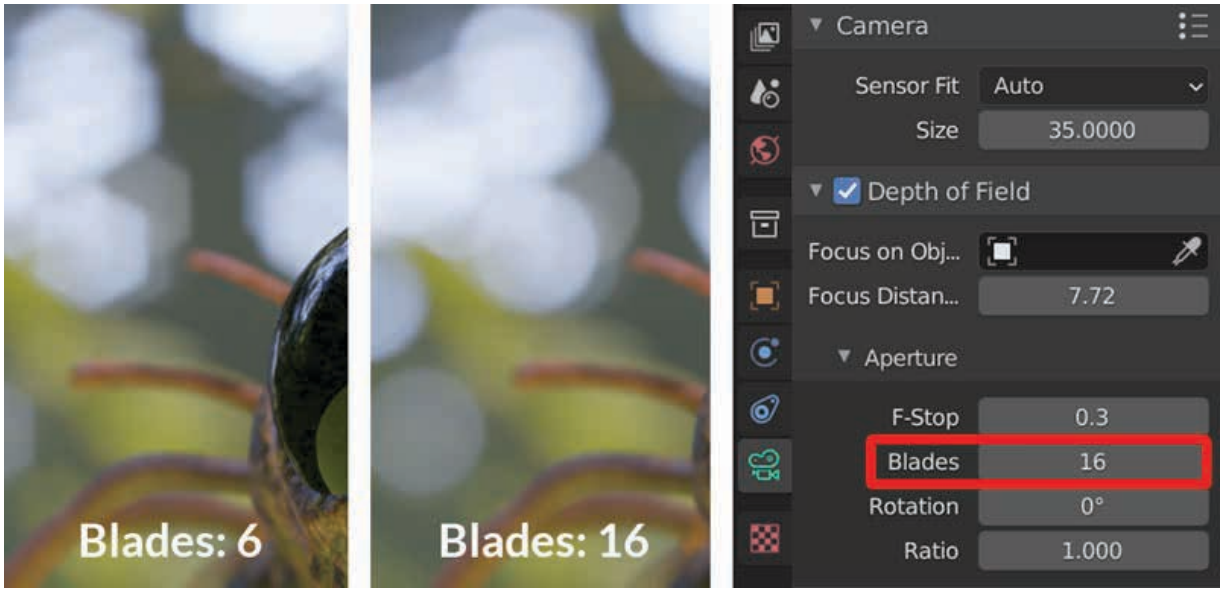
Okay, so after that refresher on how to achieve shallow focus, let's look at what's new. Besides Eevee handling depth of field better than before, there are now several post-processing settings to tweak to make the quality of the unsharp parts (or the quality of **the "bokeh"**) even better.

Some of them are very subtle and best left at the defaults, so let's look at the most important ones. **Max Size** increases the pixel size of the post-process, making the blurred background more spread out. The higher this value, the longer it will also take to render.

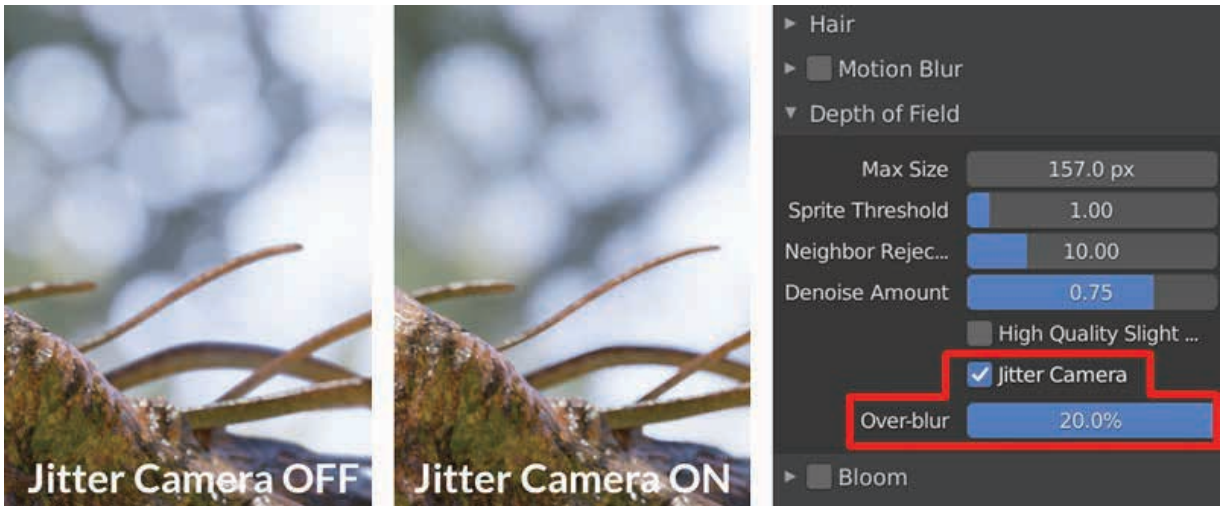


Tree Creature scene by Danield Bystedt

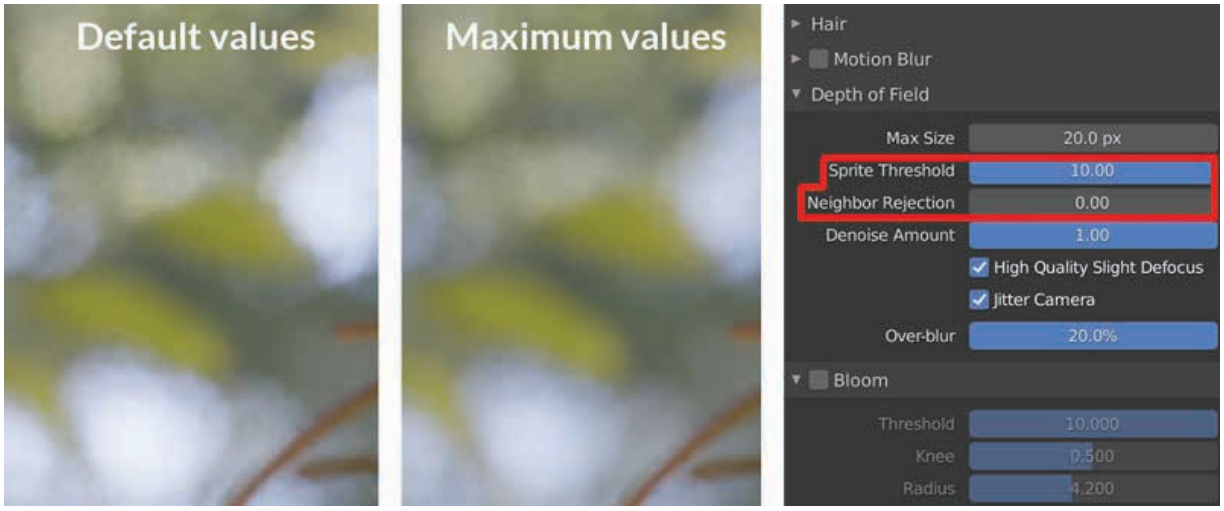
Notice how the unsharp highlights have a geometric shape. This can be controlled by the **Blades** value in the Camera Depth of Field settings. The more blades, the rounder these shapes will be.



Jitter Camera actually renders the frame several times from slightly different random angles, which leads to a more realistic depth of field, at the cost of longer render times. In case this leads to a kind of pattern of dots in the highlights, you can increase the **Over-blur** value to blur the dots together.



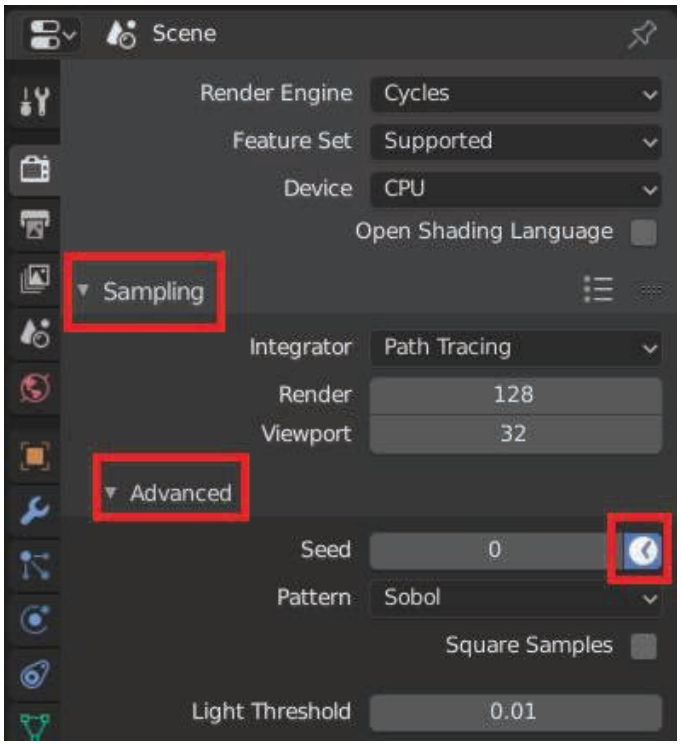
Those are the values that will have the most dramatic impact on the look of the unfocused areas. More subtly, higher **Sprite Treshold** and lower **Neighbour Rejection** values improve the highlights at the cost of render time and reduced brightness.



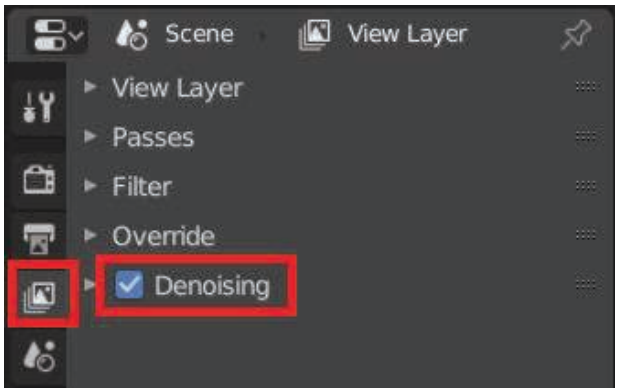
Denoise Amount and **High Quality Slight Defocus** both reduce flickering during animation. I recommend leaving these last 4 values at their defaults, unless test renders show a significant difference. Keep in mind also that these are artistic choices, not just technical ones: do what do you think looks best!

Randomize Grain

If you don't use "denoise", you should set grain to "random" by pressing this clock icon in the Render Settings.



It causes the grain to be different each frame, like we are used to from film. Be aware, that when using "denoise", you should not randomize the grain. Denoise needs to have a predictable grain pattern for the best result.



It depends on the project and personal preference which one you choose. I personally don't mind a bit of grain: it makes it look like film. So I turn on the Randomized Seed.

But some projects need to look very "clean", in that case I'll use Denoising. Just make sure you never use both at the same time.

Bevel Node

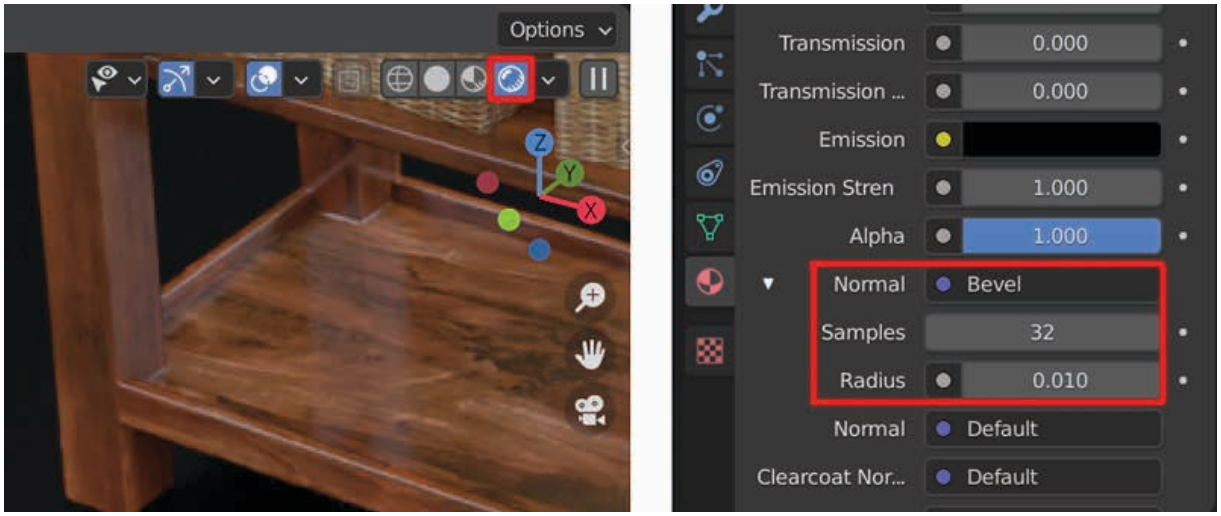
If you want to quickly add nice shiny beveled edges to objects, but don't want to spend the time doing it by modeling, a quick solution would be to use the **Bevel node**.

The benefit of this, is that it's a non-destructive solution which you can toggle on and off. It also works when adding a bevel through modeling or the Bevel modifier might just not work, because of the particular geometry.

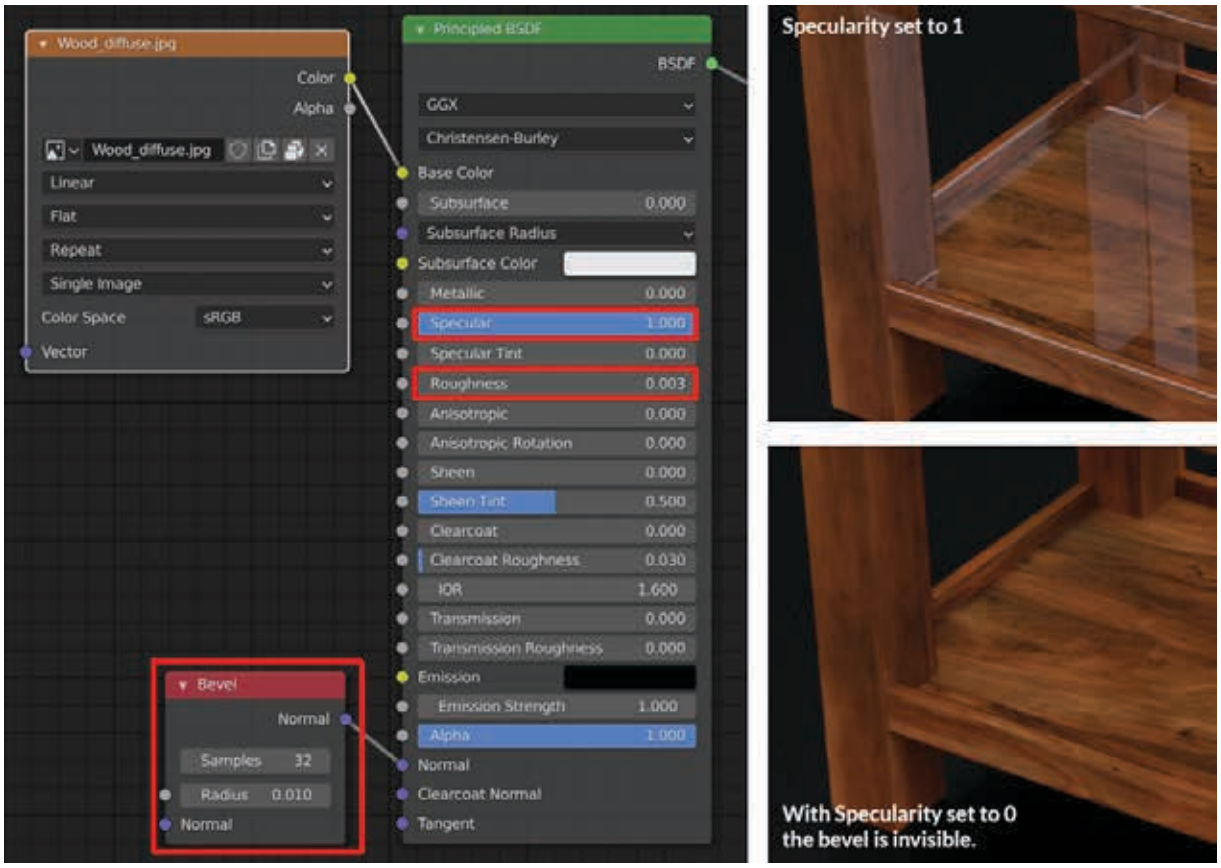
It works very nicely with shiny, specular objects like this kind of lacquered wood. Here you can see the difference more easily:

It's easy enough to enable this. First of all, make sure you are in **Cycles** and that you are in **Rendered** preview. For the moment at least, it does not work in Eevee.

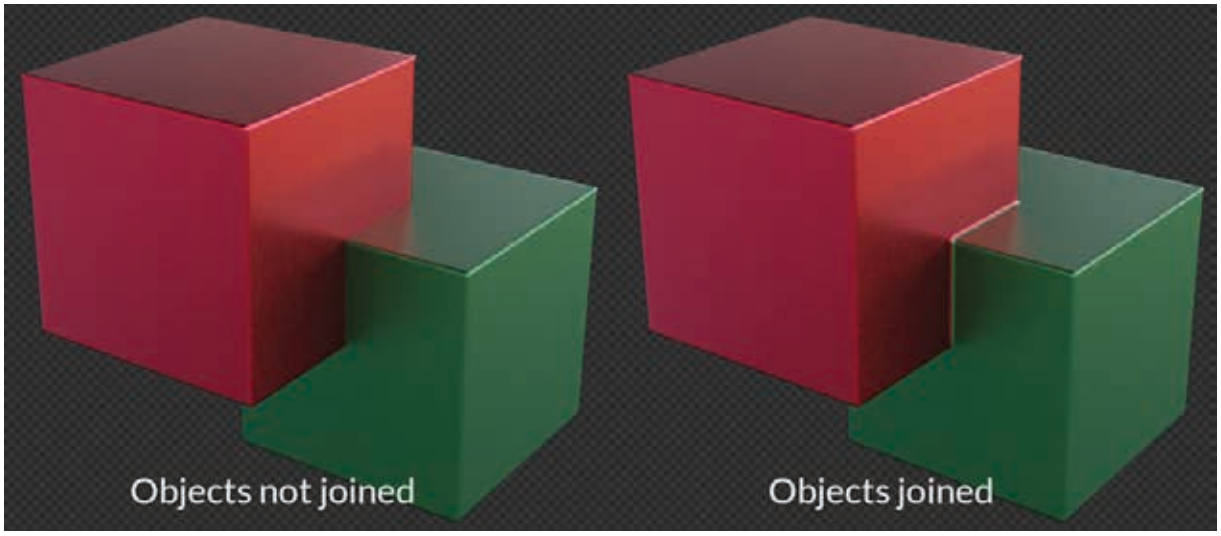
Then, in the Material tab **set Normal to "Bevel"** and enter a **Samples** value (the more the better, but also the longer it takes to render), and experiment with the **Radius**. I find that with real-world sized furniture for example, the Radius needs to be pretty low: 0,01 for example.



Alternatively you can plug the "Bevel" node into the Normal input of the Principled BSDF. The beveled edge looks best with really shiny materials: high Specular values, and low Roughness values.



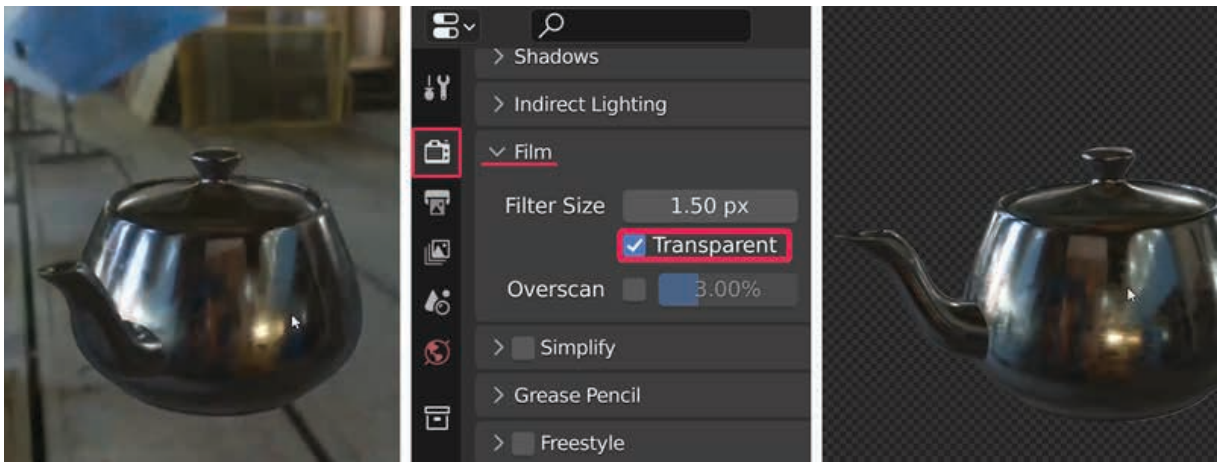
Below you see the difference between joining the objects (with **Ctrl+J**) or leaving them separate. In the latter case, you don't get a bevel where the objects join.

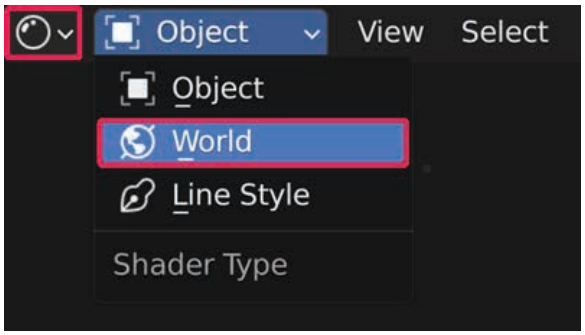


The Bevel node works in Cycles, but not in Eevee. Hopefully it will at some point!

Use HDRI only for foregrounds

To use an HDRI to cast light on a foreground object without showing up as background, one option is to turn on Transparent under the film options.



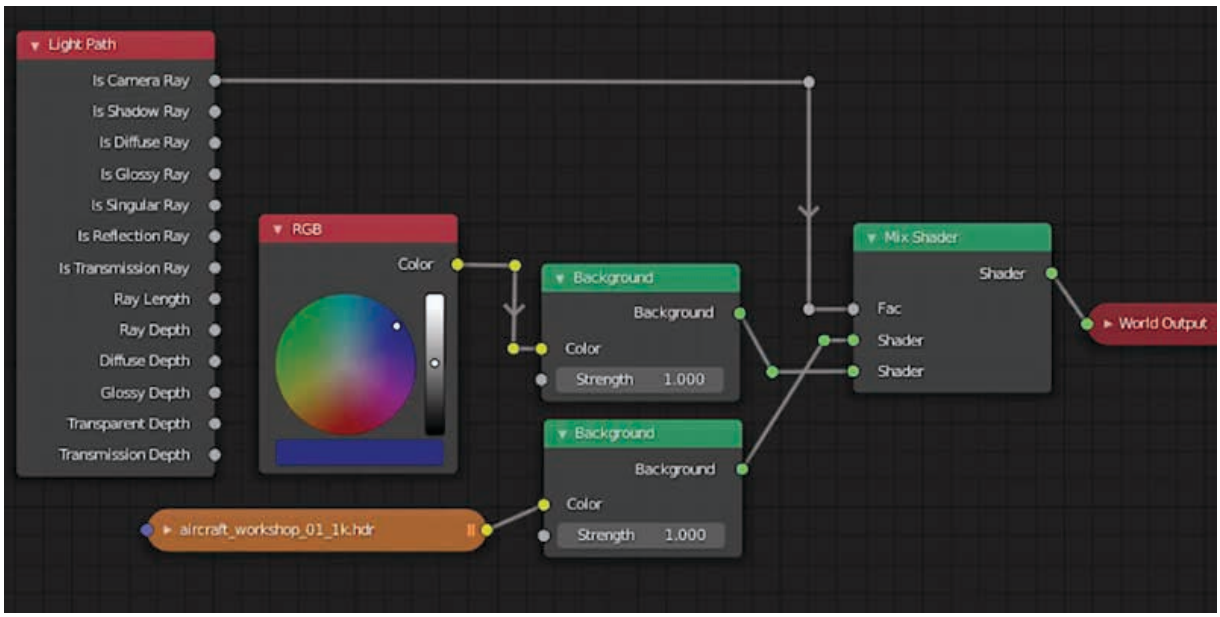


But that gives you a transparent background.

And what if you just want some color instead?

Go to the Shader window and as shader type choose World.

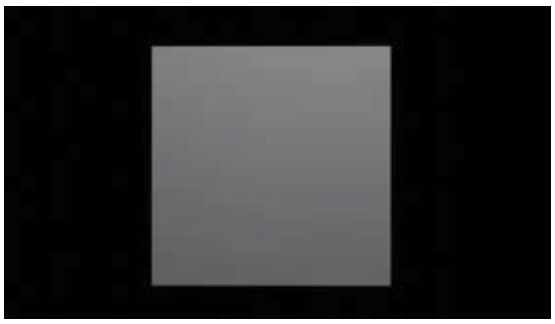
Use the Light Path node as a factor between the HDRI and an RGB color. Use the Is Camera Ray output.



Download this file if you want to experiment with this set up. Or to see how I've made the metal material.

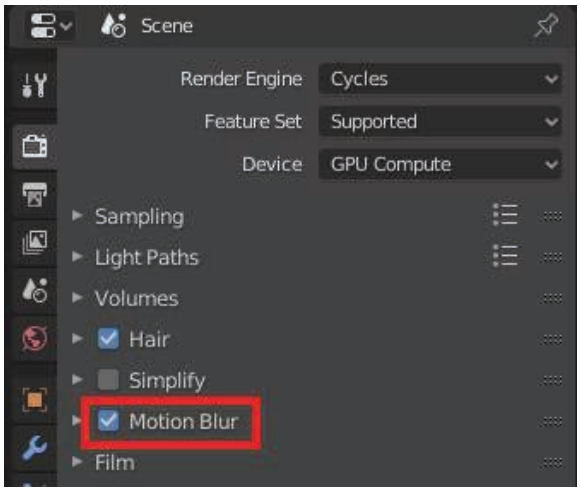
How to use motion blur

This cube is animated to move from left to right really fast.

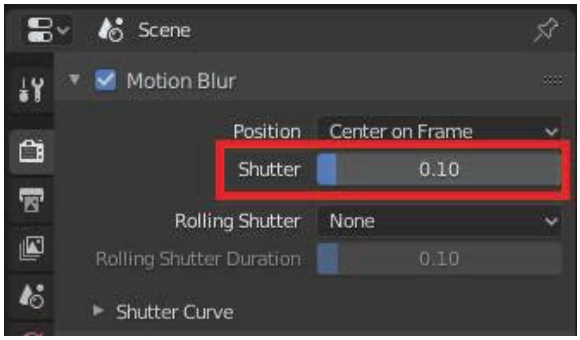


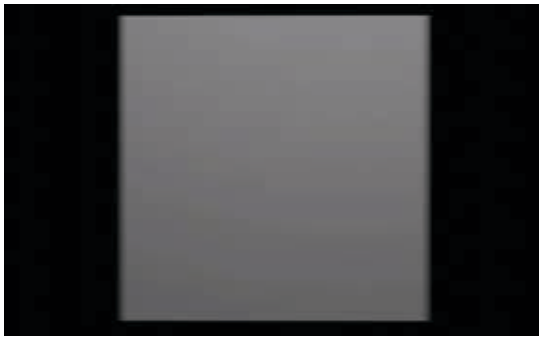
However if motion blur is not turned on, fast moving objects like this will have no blur at all if you look at them in a still frame like this.

Turning on **Motion Blur** in the **Scene tab** gives you the default settings, which are fine in most cases.



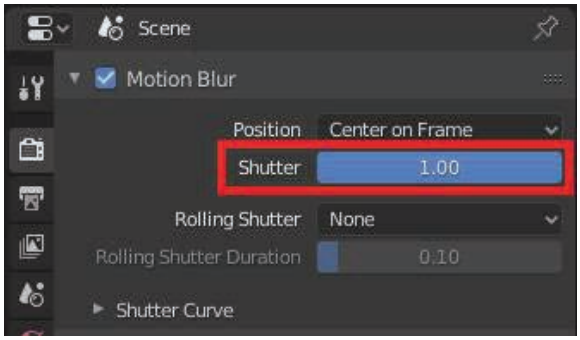
Reducing the **Shutter value** will reduce the amount of blur.



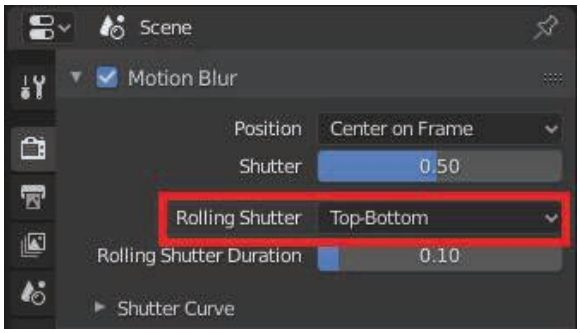


Increasing the **Shutter value** will increase the amount of blur.

Click here to return to the index.

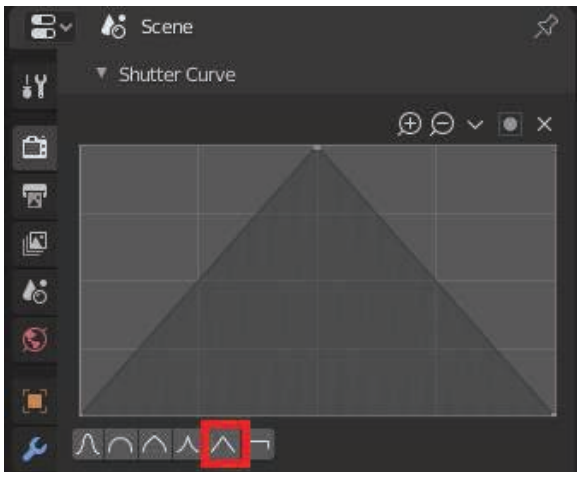


Turning on **Rolling shutter** will imitate the look of filming with for example a smartphone camera.

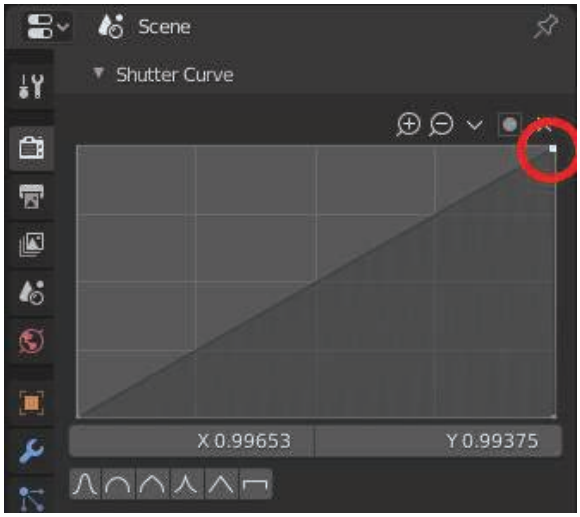


You can even create a custom shutter curve.

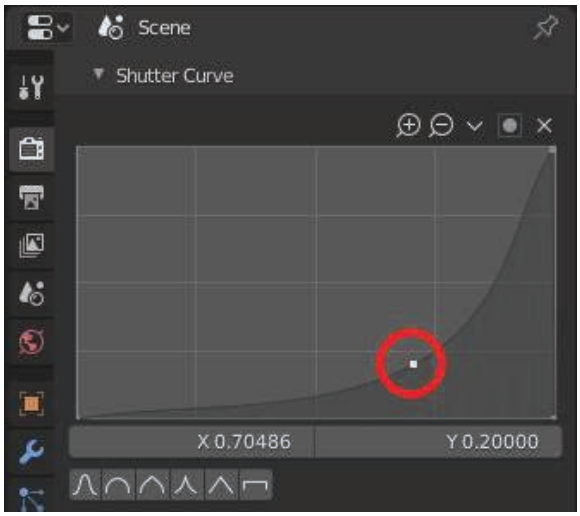
Let's make a custom shutter curve. First click on one of the presets like this triangle shaped curve.



Grab the top point and drag it almost entirely to the right.



Click in the middle of the curve to add a point and drag it down.



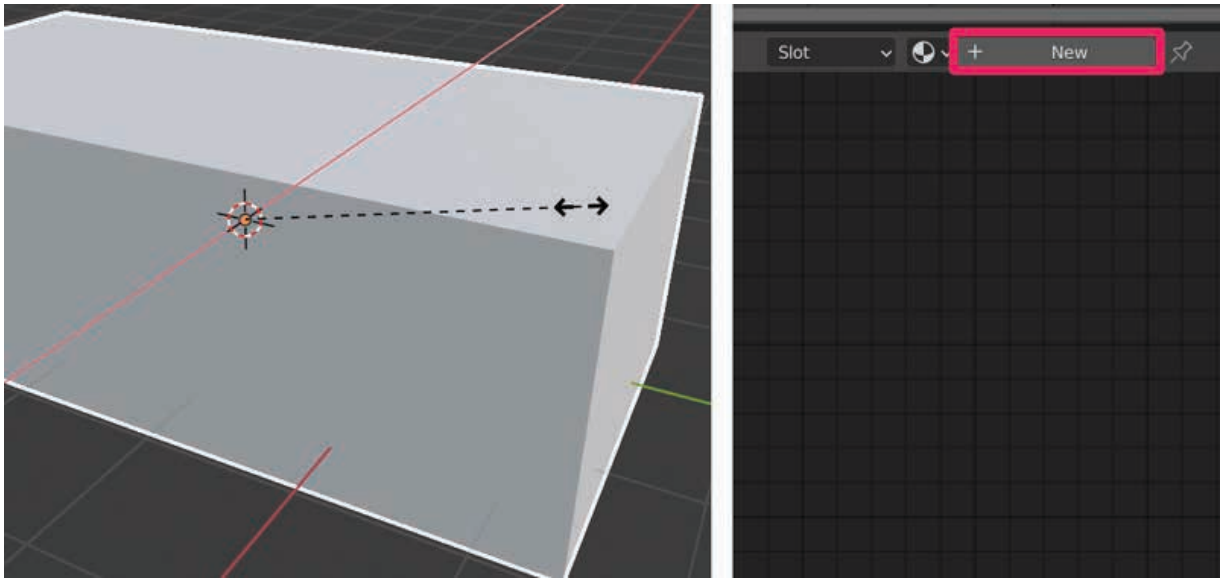
The resulting motion blur looks like this. Of course this is just an example, the idea is that you can make exactly the kind of motion blur you want.

Click here to return to the index.

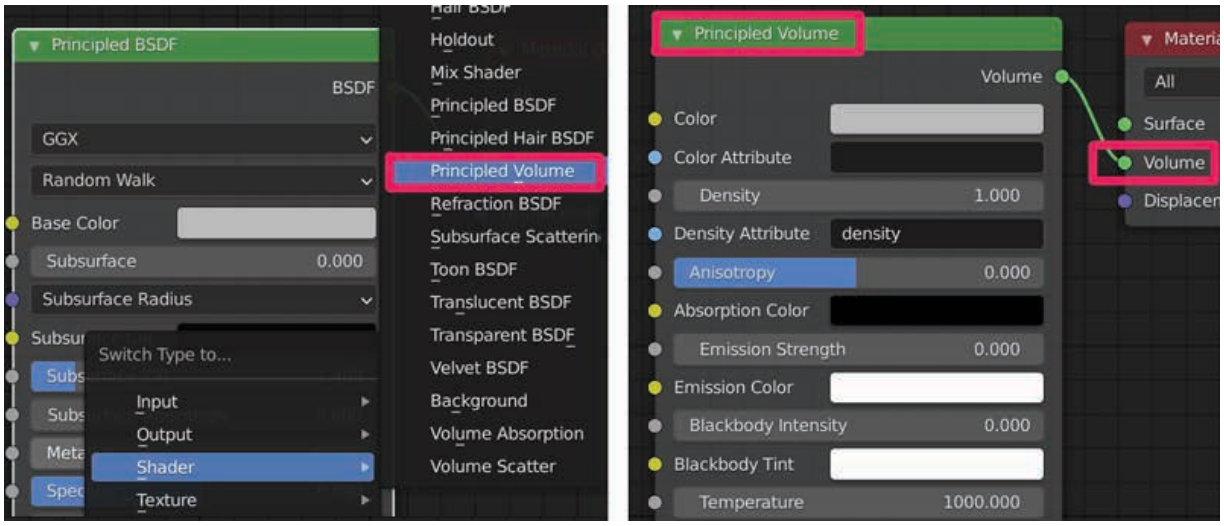
Easy animated volumetric fog in Eevee

Adding some animated Fog in Eevee is easy, using a Principled Volume material and some procedural noise. *(Please note: this technique also works in Cycles, although of course at longer render times).*

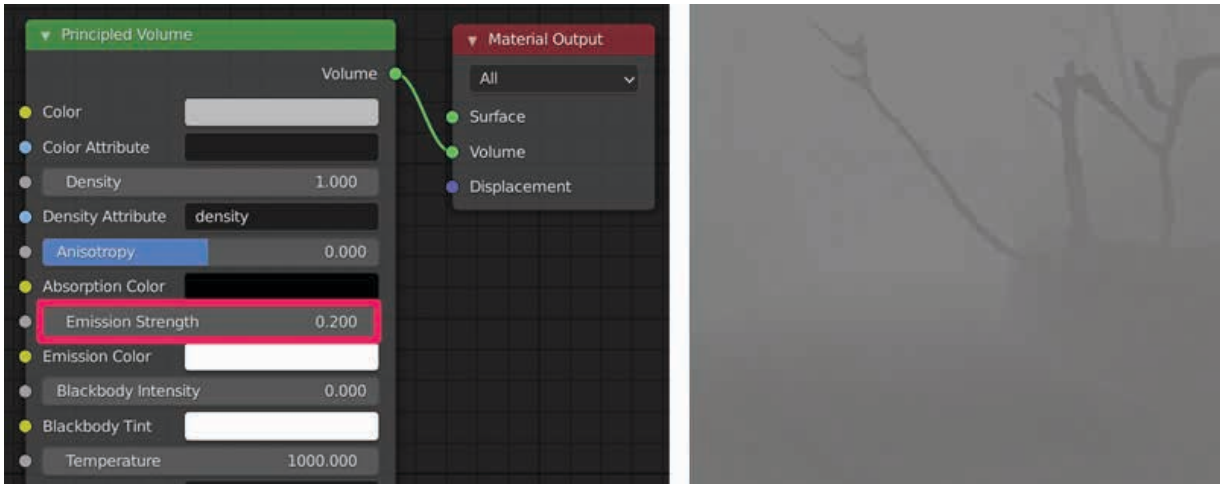
Add a **Cube** and scale it around the part of your scene that's visible in camera. Give the cube a material by **clicking on New** in the Shader Editor window.



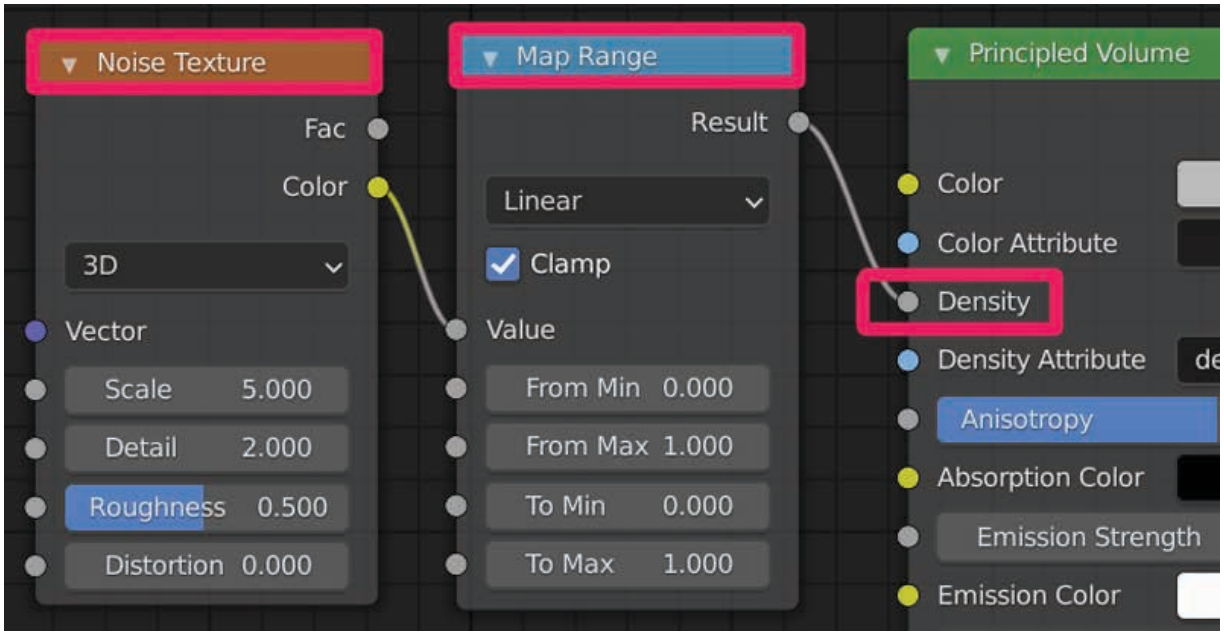
Select the **Principled BSDF** node and **press Shift+S**. This shortcut replaces the selected node with whatever you choose next. Select Shader > **Principled Volume**. Connect it to the **Volume input** instead of the Material Output node.



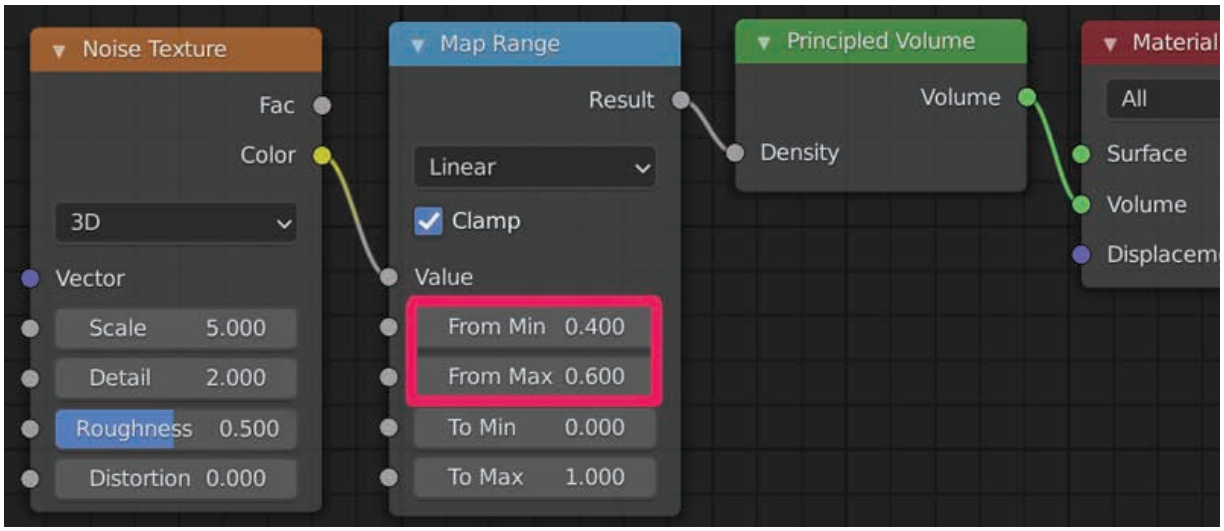
Increase the **Emission Strength** until you get light grey mist.



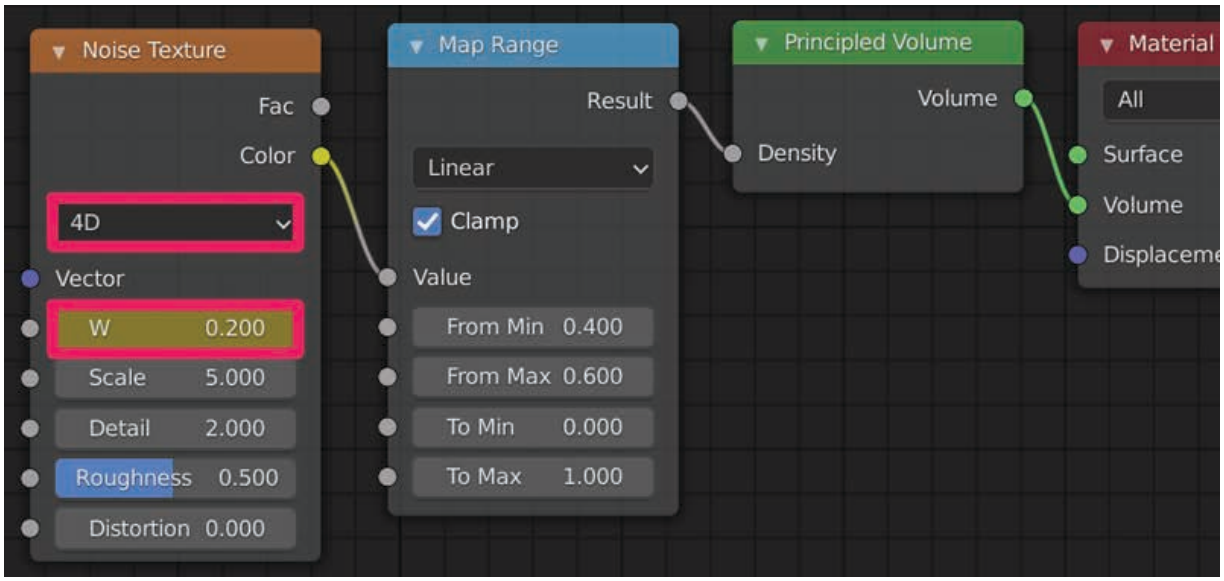
Press Shift+A and search for a **Noise Texture** node and a **Map Range** node. Connect them to the Volume Scatter node **Density** Input.



Increase the contrast by bringing the **From Min** and **From Max** values of the **Map Range** node close together *(note that I've collapsed the Principled Volume node for the purpose of screenshot efficiency by selecting it an pressing Ctrl+H).*



Set the Noise Texture to 4D and set keyframes for the W value on the Timeline. You don't need to have very different values for W, for example I keyframed it from 0 to 0.2 in the video example of this topic.

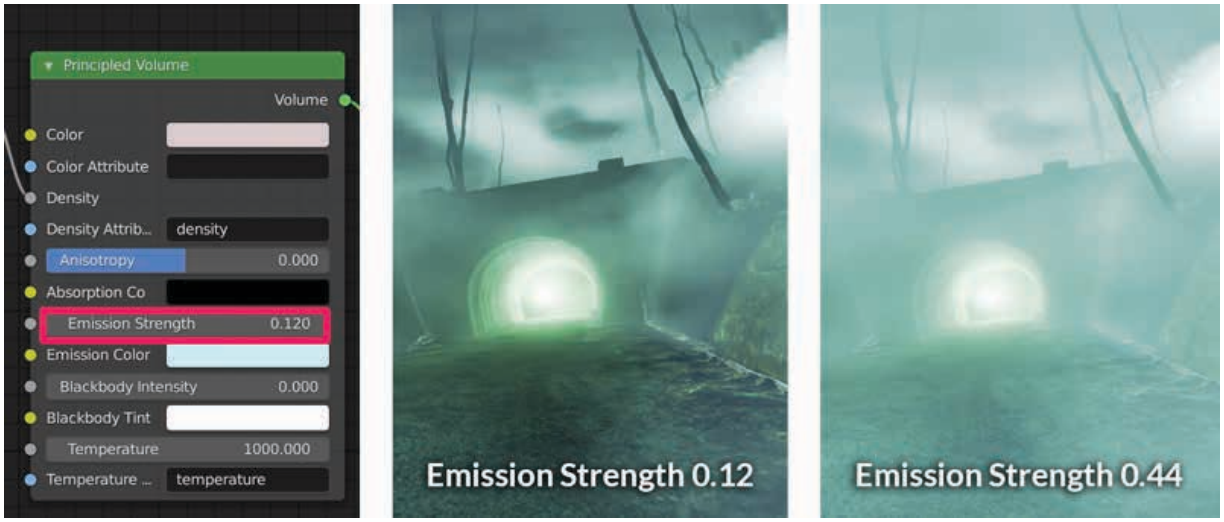


Finally add a Sun lamp or an HDRI to add some more light.

Now you've got animated creepy mist.

The more lights you add, the more visible the mist will be.

If you're not getting the look you want, the most important value to experiment with is the **Emission Strength** of the Principled Volume.



After that, try adjusting the **Scale** of the Noise Texture.

3D scene "Futatsugoya Tunnel Abandoned road Japan" (https://skfb.ly/6Sp8O) by Fuji on Sketchfab.

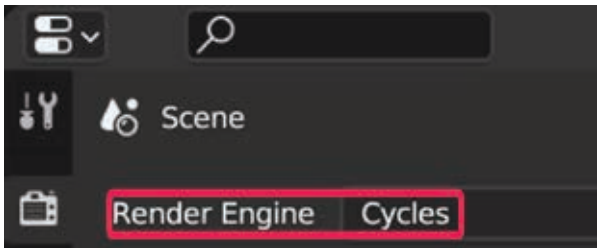
Shadow Catcher in Cycles

Sometimes you want to render the foreground and background separately, in order to be more efficient or to have more control over your shot during the compositing stage.

But what about shadows? They may be considered a part of the foreground element(like a character), but they are in fact rendered as part of the background.

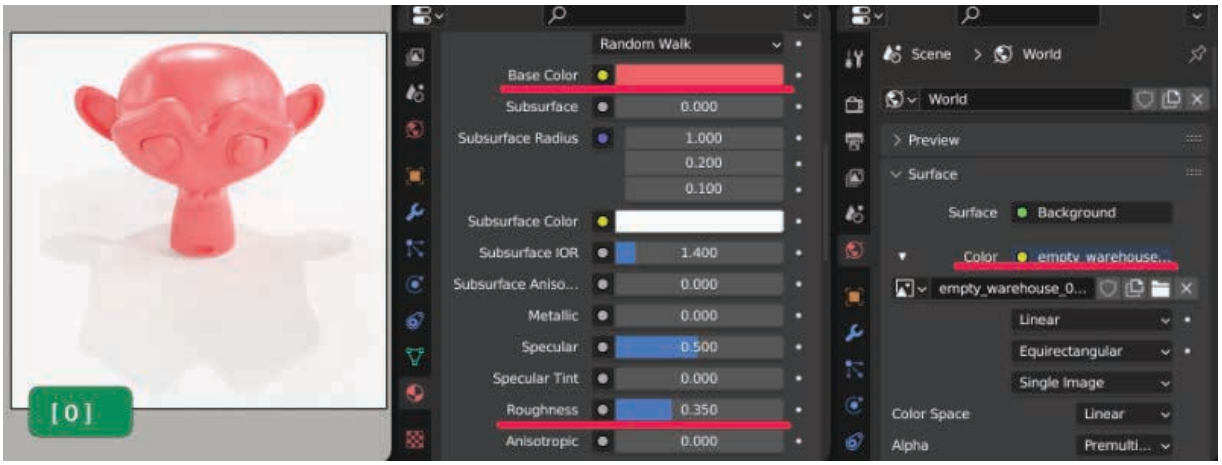
So how do you render the shadow, but not the background? With a “Shadow Catcher”.

First of all, make sure you are in **Cycles**.



As an example, I’ve created a simple scene: A Plane and a Monkey. The monkey has a low Roughness value, for the rest it’s just a standard Principled BSDF with a slight reddish tint.

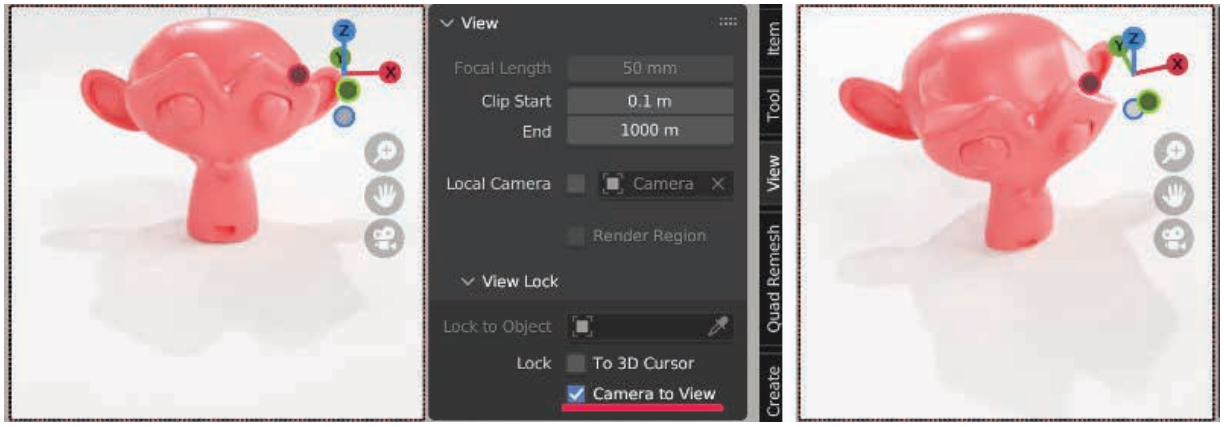
I’m using a free HDRI for the background (empty_warehouse_01_2k from HDRI Haven).



Click here to return to the index.

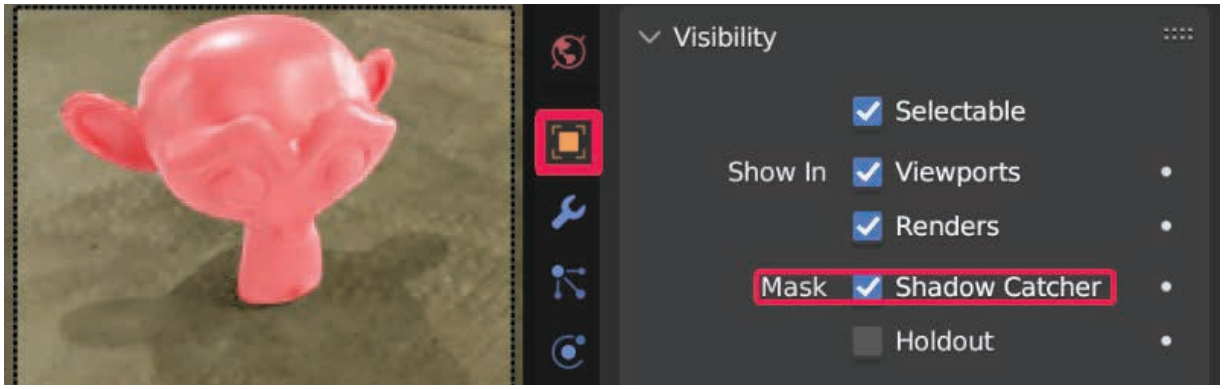
Make sure you have a camera so that you can render an image. Press 0 (zero) to toggle the camera view.

To move the camera like moving the viewport, **Lock Camera to View** needs to be checked in the Option panel (**press N** to open the option panel).This is an easy way to move the camera.

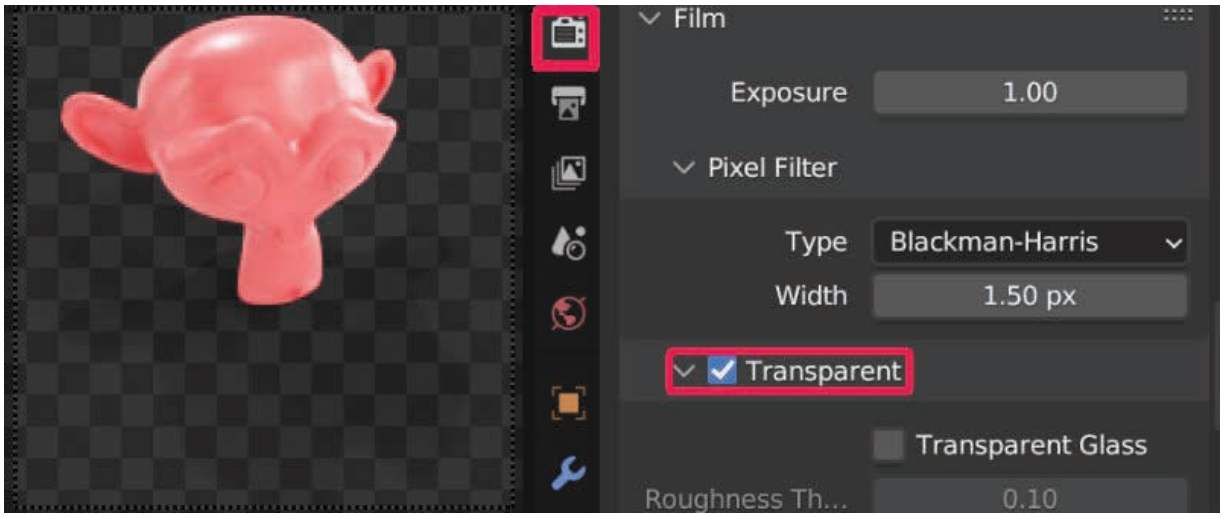


To render a transparent shadow first select the plane that the monkey rests on.

In the Object tab, under Visibility turn on Shadow Catcher.



In the Render tab, under Film turn on Transparent.



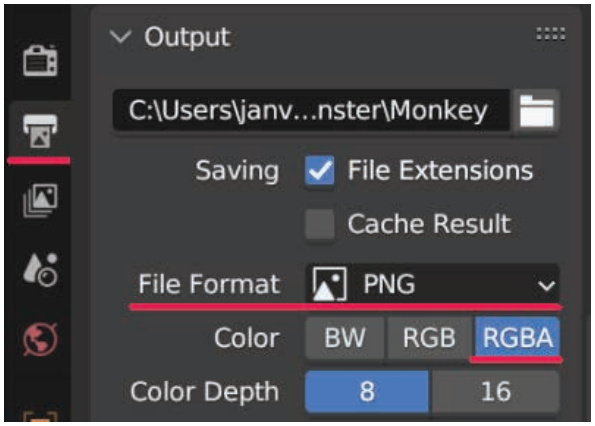
Could it really be that easy?

We already see the transparent shadow in the viewport.

Render by **pressing F12**. You will see the transparent shadow in the rendered image as well.

As long as you save the image with an **alpha channel**, this can be composited on a background render.

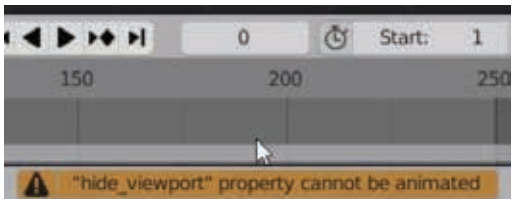
For example, a PNG with RGBA Colors.



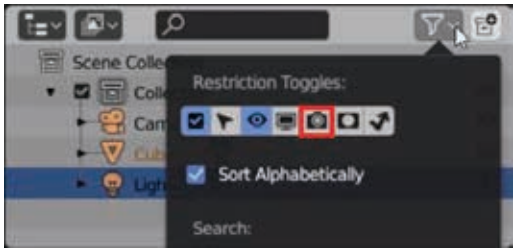
Keyframe render visibility

In Blender 2.79 it was easy to keyframe animate the render visibility of an object. You just pressed "i" with the mouse cursor above the camera icon in the Outliner.

But in Blender 2.8 and up, **pressing i** on the eye icon gives an error message.

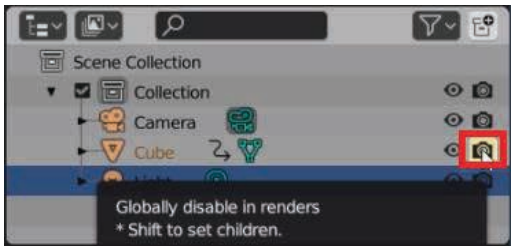


In fact, you need to use the **Render visibility toggle**, which you can turn on through this Outliner filter.



Now you can **press i** while hovering the mouse over the icon to keyframe animate the render visibility in Eevee. You will notice that the icon turns yellow once it has a keyframe. To remove a keyframe from an icon or checkbox, hover the mouse cursor over it and **press Alt+i**.

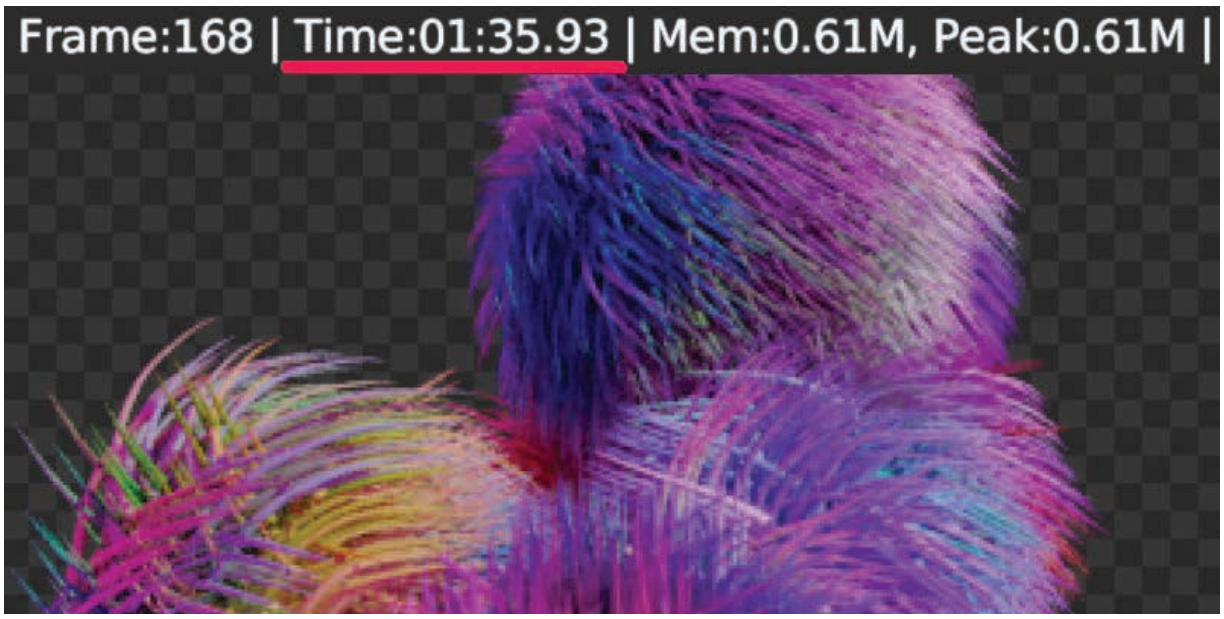
Set a key for visibility on, then on the next frame a key for visibility off.



You won't see the effect in the viewport, only in the render.

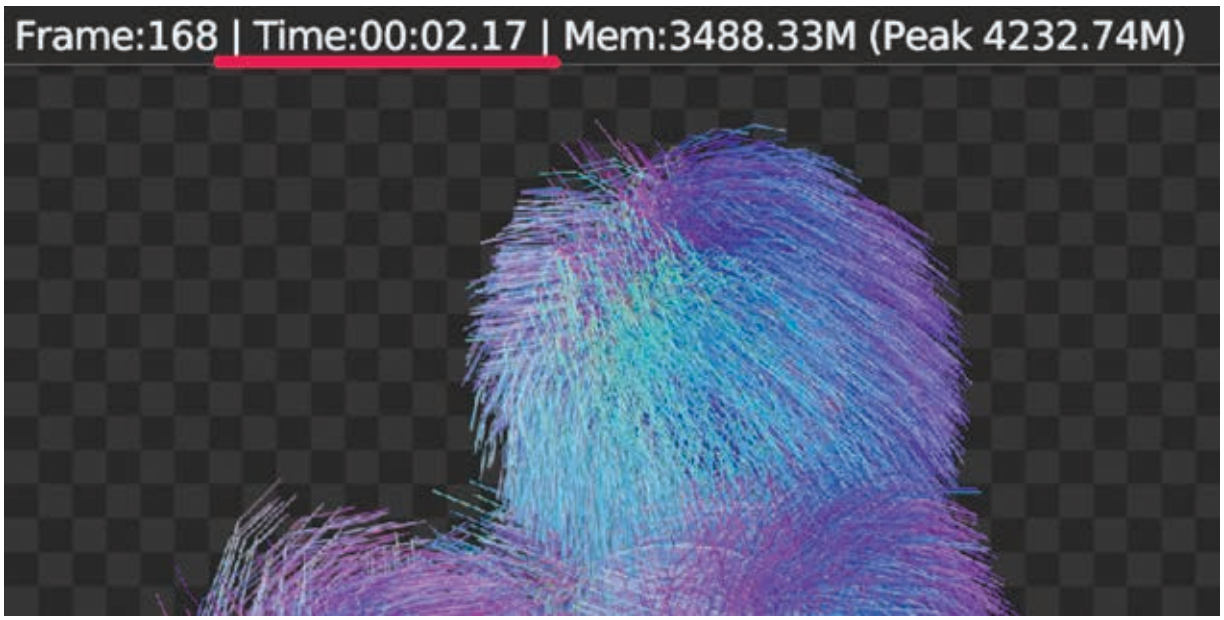
Viewport Animation Render

Rendering takes a lot of time.



You should always do a low resolution viewport render first.

This way you can see if the animation looks good or needs improving, without wasting a lot of time on rendering.

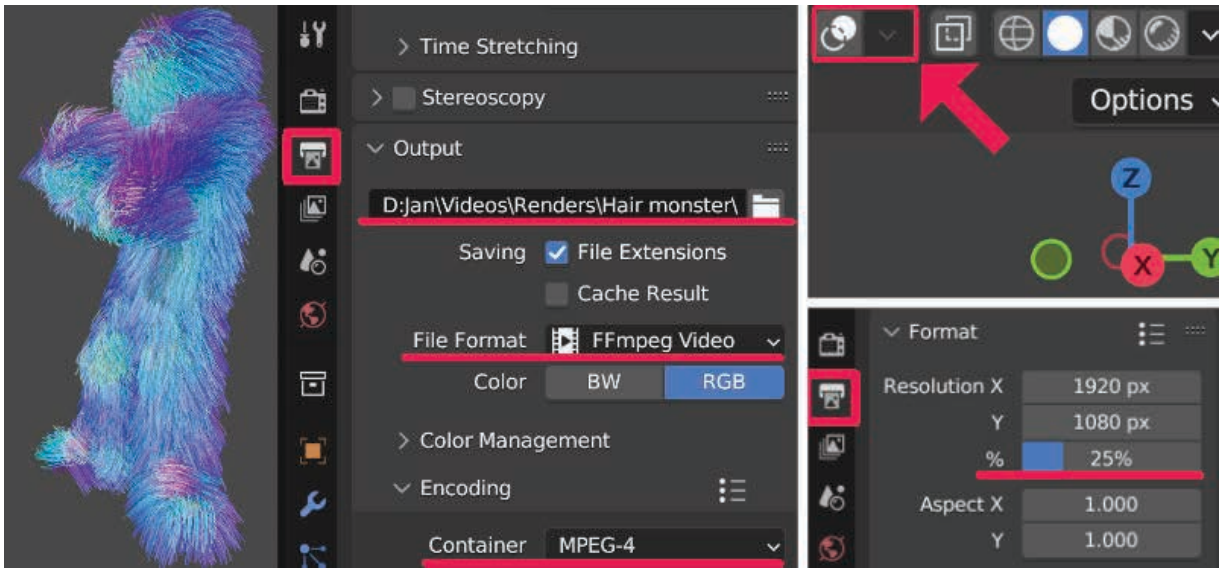


You might as well save a file.

I like to use FFmpeg, H.264, MPEG-4 video files. The files are small and the quality is still okay - good enough certainly for a test render.

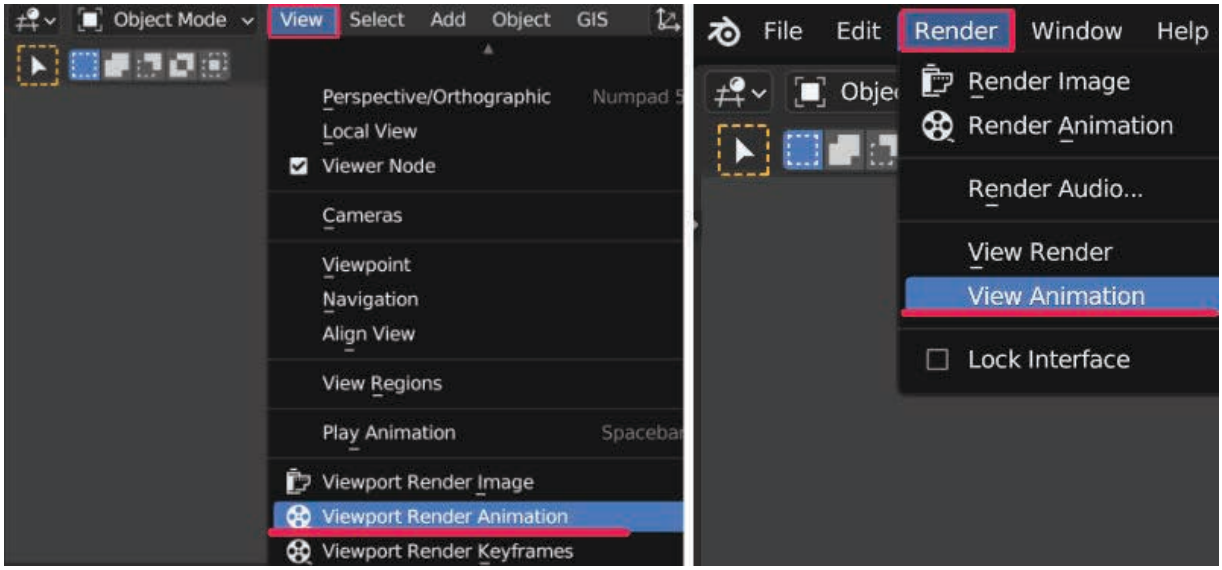
You can turn off the Viewport Overlays, otherwise they will be visible in the render.

Set the resolution to 25%.



This renders faster and an even smaller file size. It's still good enough to judge the animation.

Then go to View > **Viewport Render Animation**. After its finished rendering go to Render > **View Animation**.



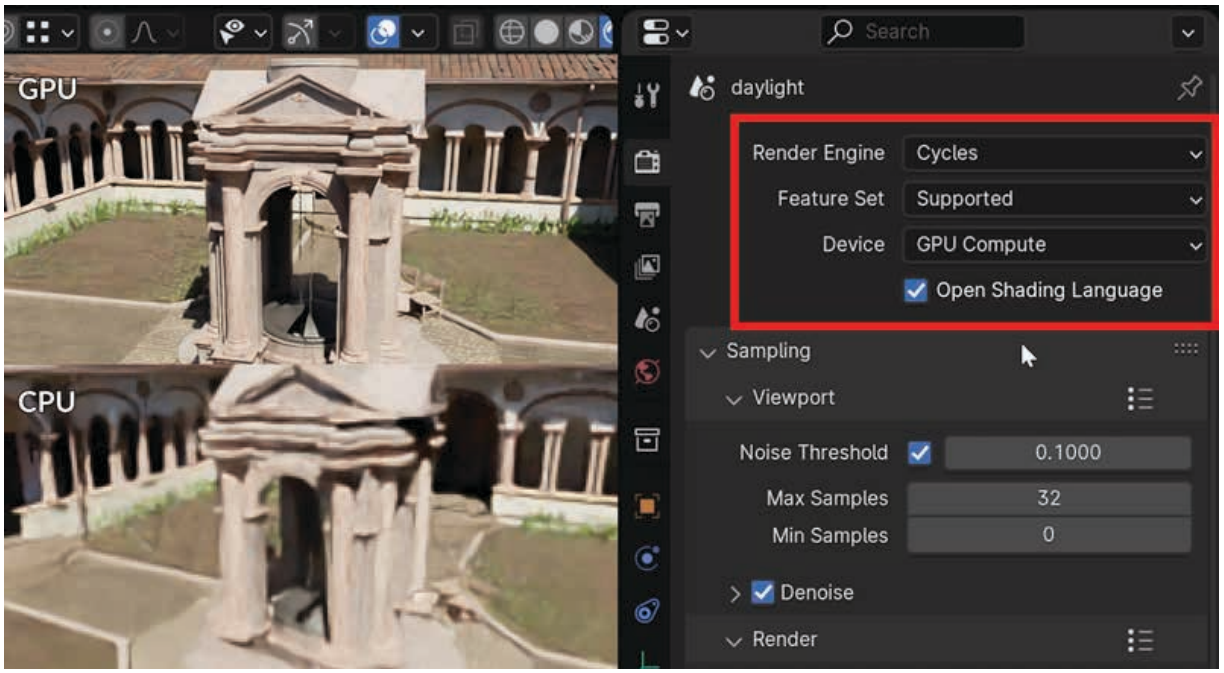
This is a great way to judge the animation before rendering in final quality.

Cycles Rendering Basics

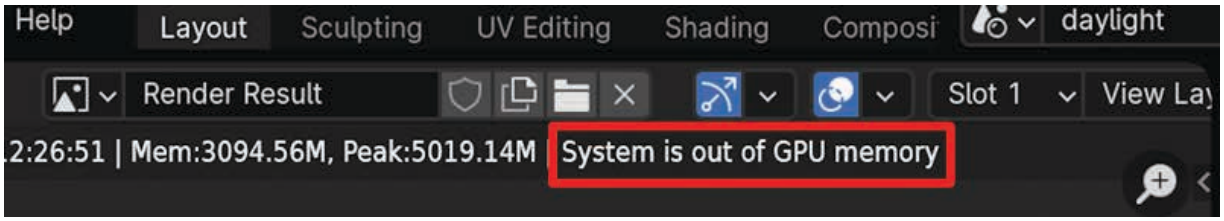
Cycles is Blender's built in physically-based path tracer. Fun fact, it's actually an add-on. But one that is enabled by default.

Art by Metin Sevin, based on a concept by Blake Stevenson

After enabling Cycles, set it to **GPU Compute** if your graphics card supports it, for faster renders.



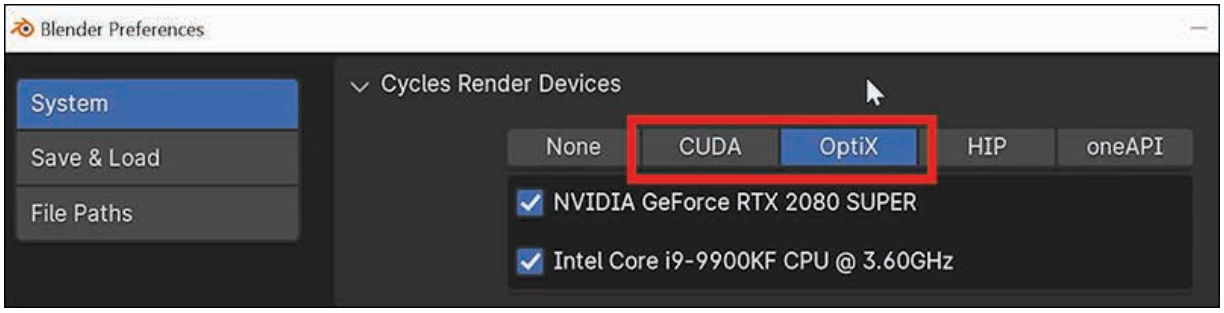
When you do want to use the CPU to render, is when the memory of your graphics card isn't sufficient for the scene you're trying to render, and it gives you an out of memory error: "System is out of GPU memory". That can happen when you have a lot of textures.



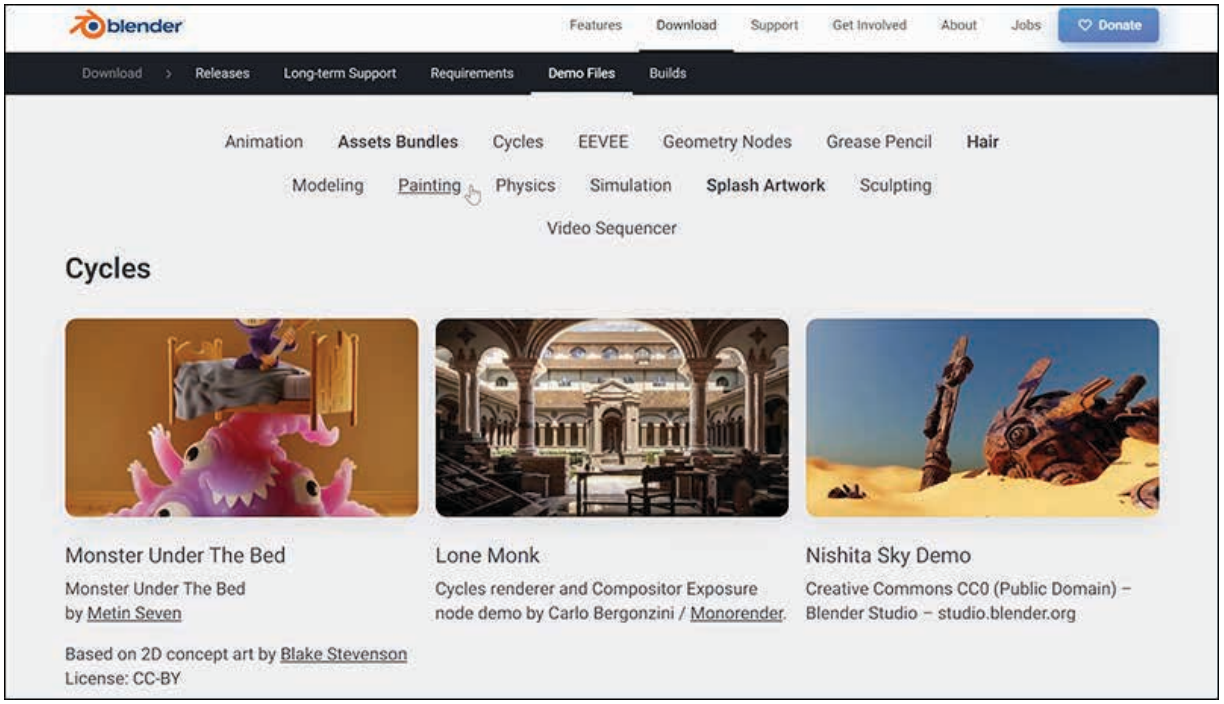
In Preferences, choose the best Render Device for your computer. If the **OptiX** render device is available in Preferences, choose that as it is usually faster. Otherwise, **CUDA** is also fine.

HIP is only for when you have an AMD graphics card (CUDA and Optix take advantage of NVIDIA hardware) and **oneAPI** requires a specific Intel GPU. You can only select the option for which you have the correct hardware.

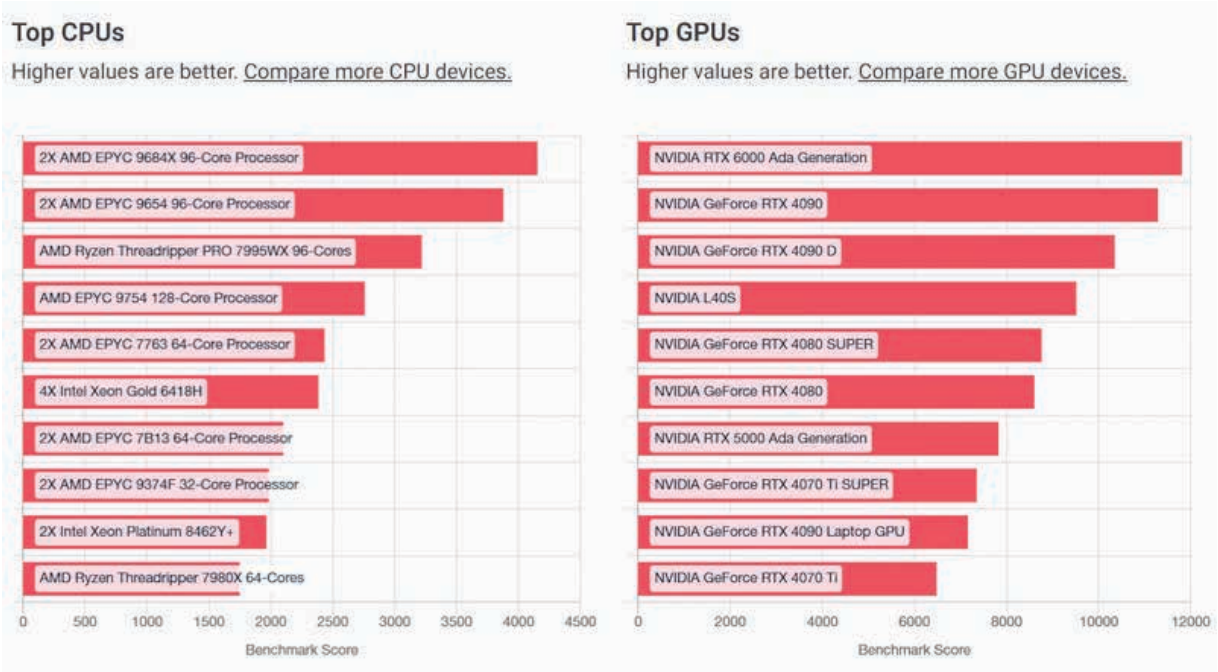
On Apple, you can choose **Metal**, if you have the right hardware. But keep in mind, that some features such as Shadow Caustics are not supported.



You can always download a **test scene,** to see if these things actually make a difference on your specific computer. There is a dedicated page with demo scenes that you can download.

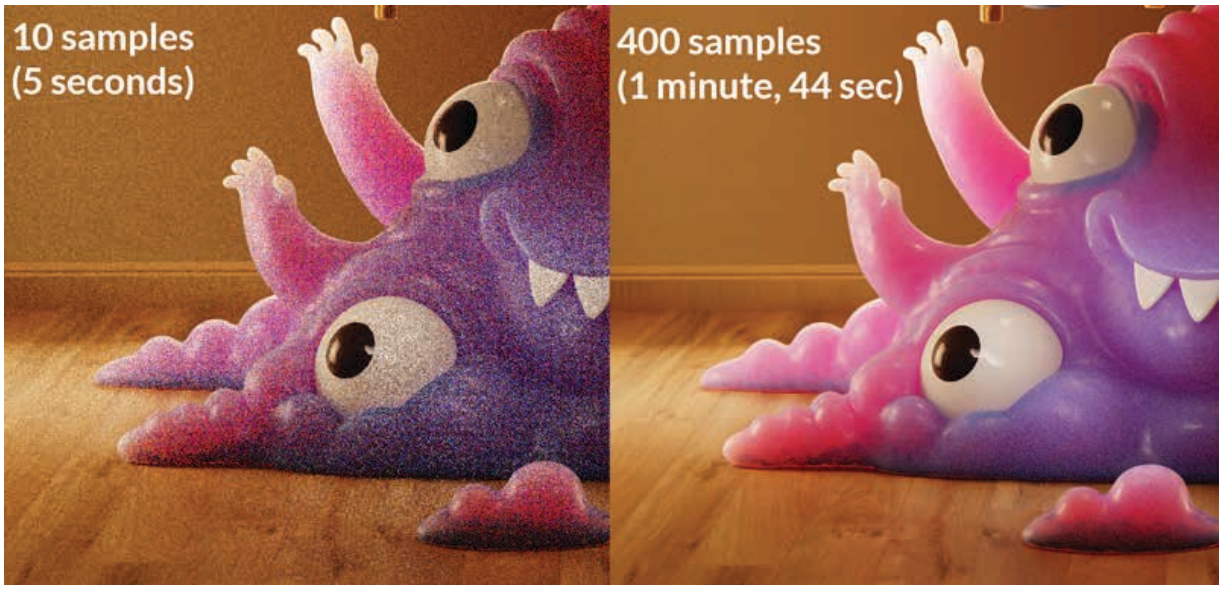


They also have an overview of tested hardware. The top GPU listed costs something like 10.000 euro.

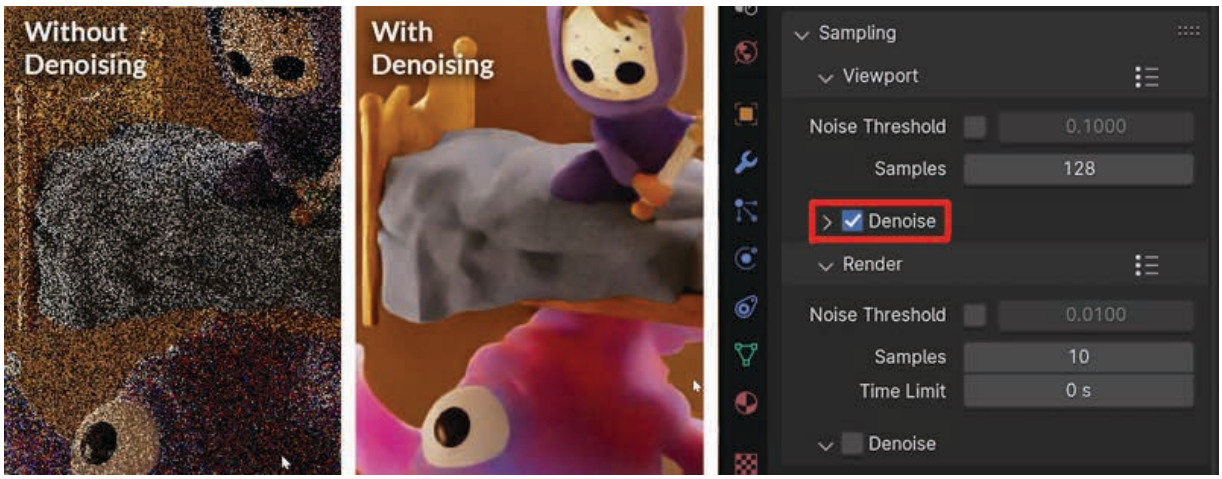


More render **Samples** means less noise and a more accurate render, at the cost of more time spent rendering. So you have to find the least amount of samples you can get away with to get a nice image.

If you set viewport sampling to 0, it will just keep rendering until you manually end it. That's a good option if you just want to render a still image, while you're away from your computer for a short break.

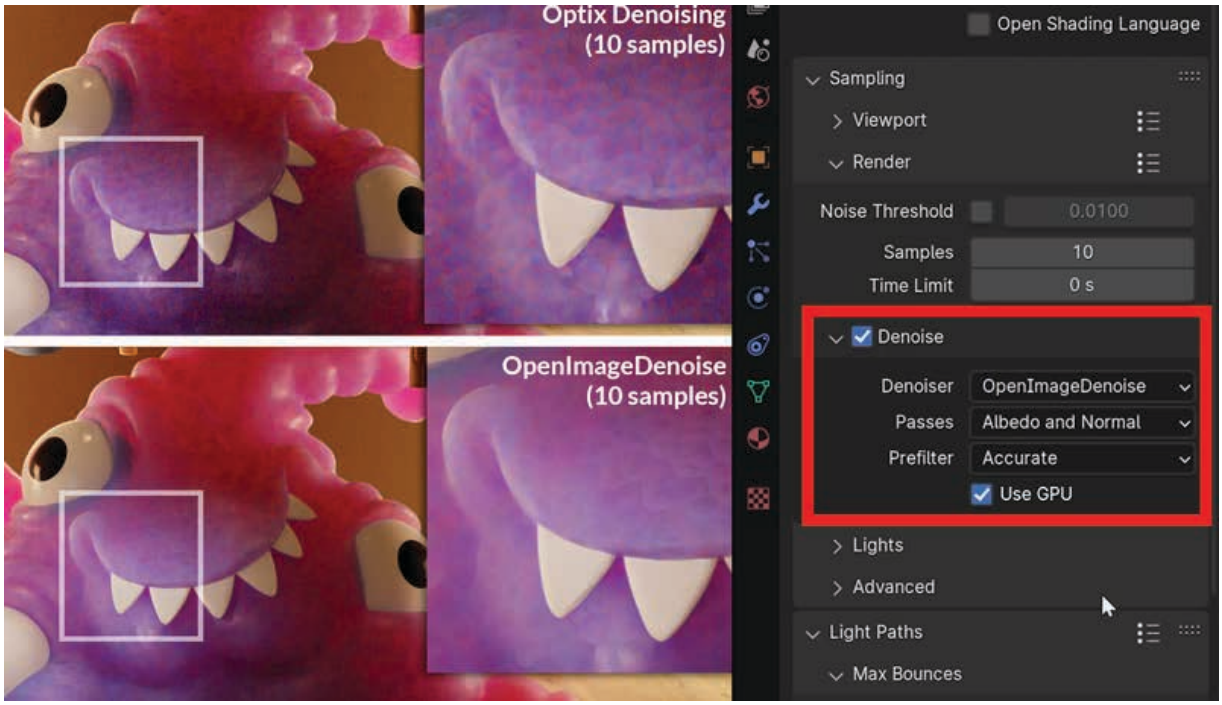


But you can also enable **denoising**, which helps in reducing the required samples.



For Denoising, you can choose between **Optix** and **OpenImageDenoise**. OpenImageDenoise seems to have a better quality result in most cases.

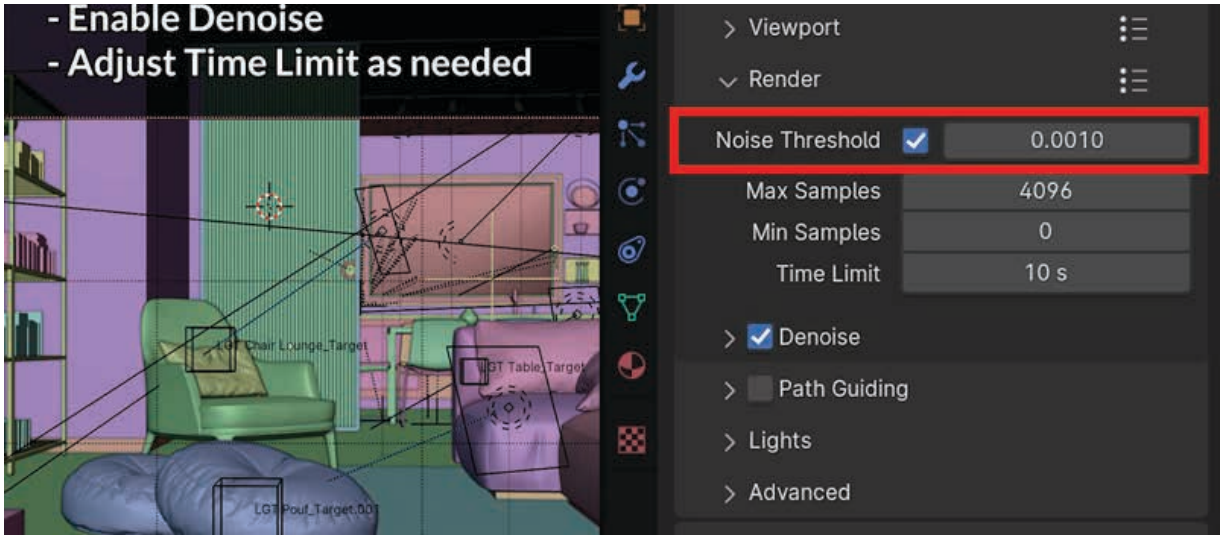
In my test certainly, with only 10 samples, the OpenImageDenoise had higher quality results. Especially in things like subsurface scattering.



If you want to limit render time, you can set a **Time Limit**.

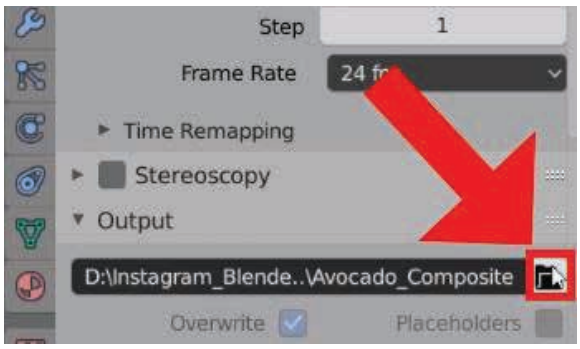
The **Noise Treshold** tells Cycles when a pixel is good enough so that it can stop sampling it. Typically this is set between 0.1 and 0.001.

If you set it to zero, it will automatically guess a value. Do keep in mind, that this is only the time spent rendering. It doesn't include the preparations Blender goes through to start rendering.



Quickly open your renders in a browser

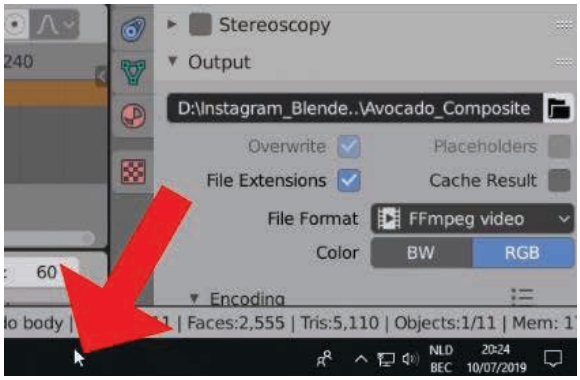
To open a browser window containing your renders **hold Alt** and click on the file location icon.



This opens the folder containing your renders.

It's a real time-saver, as you don't need to open the browser window and navigate to the correct folder.

Weirdly this doesn't always work. If it doesn't, click out of Blender (on the taskbar for example) and it will work again.



Click here to return to the index.

Track Billboards to the camera

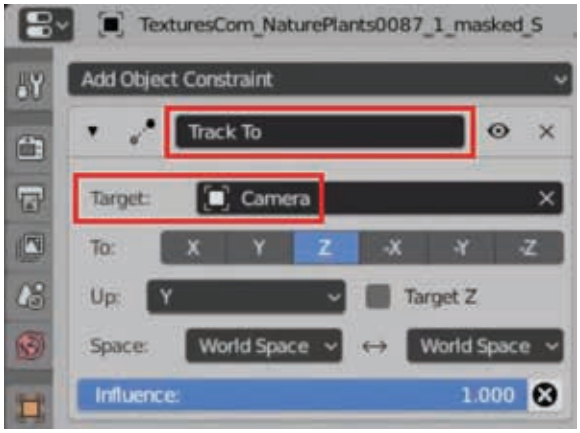
These are images on planes that always face the camera.

They add detail to a scene with very little memory use.

They are flat and only "work" when seen through the camera.

Add a **Track To constraint** to each plane. As **Target** choose the **Camera**.

Settings in this case: **To: Z**, **Up: Y** (this might differ in your case, so experiment to see what works).



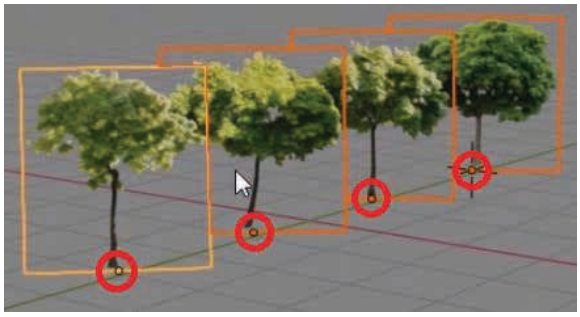
Click here to return to the index.

Particle System Billboards

The previous tip showed you how to make Billboards.

Now we'll look at setting up a particle system made with billboards, to make a whole forest.

Don't forget to make sure the origin is at the bottom.



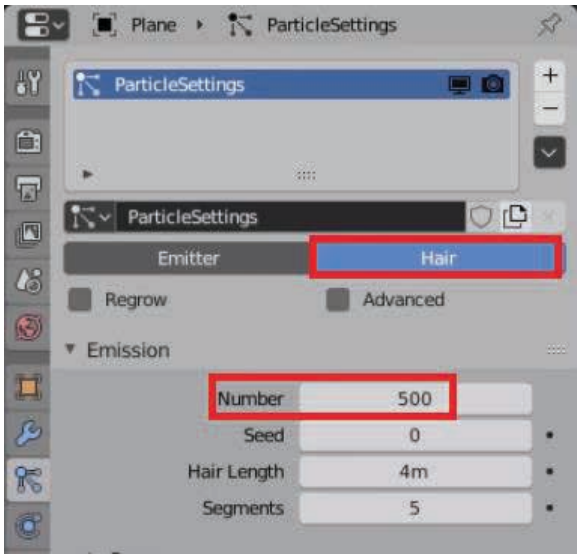
Select them and make a new Collection by **pressing m**.

Name it "Trees" or something easily identifiable.

Create a new particle system by pressing the **+ icon** in the Particle tab.

Set it to **Hair.**

You can increase the number of particles now or later.



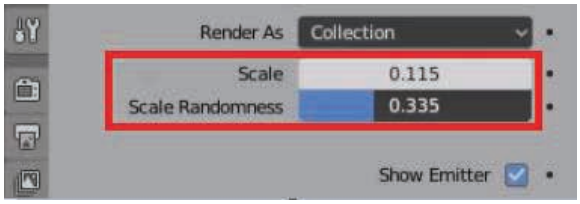
Click here to return to the index.

Under **Render As** choose Collection. As the **Instance Collection** choose the Trees collection we created earlier. Turn on **Object Rotation**.

As you can see the particles now follow the camera position so that they are facing the camera.

Under Scale you can change the size of the particles, and **Scale Randomness** makes the scale more, you guessed it- random.

Now you've got yourself a very memory-efficient forest!



Optimize Environment Scenes with the LOD add-on

When creating environments in Blender, a scene can quickly become heavy - especially when dealing with particle system for trees, shrubs, flowers and grass. Of course you can disable those particle systems in the viewport to keep a scene manageable.

Forest scene from Turbosquid

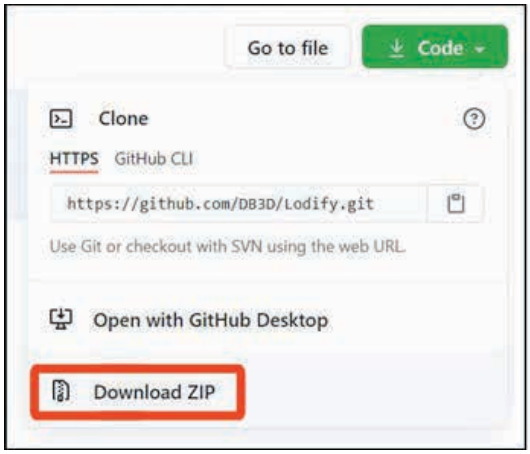
Alternatively you can use a different level of detail for your material preview, render preview and final render, thanks to the free Lodify add-on, created by the maker of the Scatter add-on. This is an interesting alternative to using proxies.

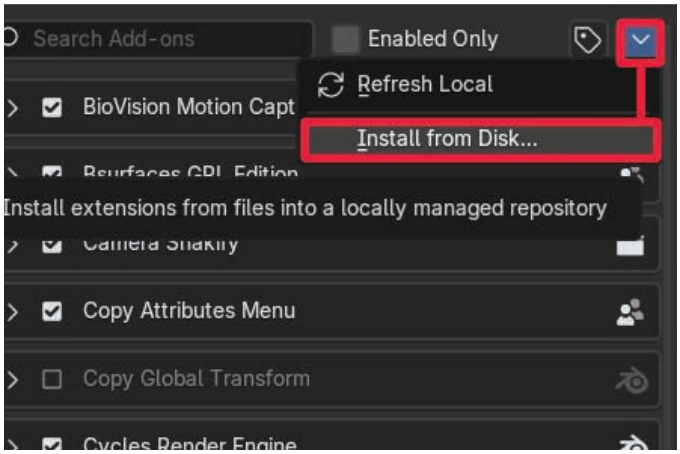
Different LOD levels

This way, you can still get a preview of all the particles, but with different levels of detail *(LOD stands for Level Of Detail)* for the different preview modes, rather than turning them off in the viewport and having to guess what they'll look like in the render.

Different LOD levels of a little shrub

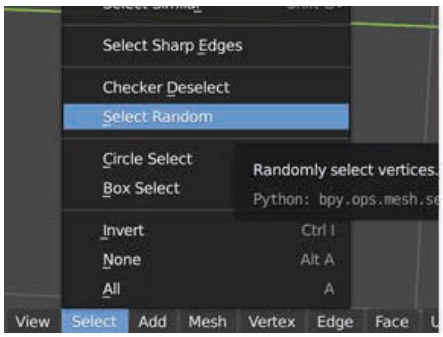
To install Lodify, go to the website https://github.com/DB3D/Lodify and **click on Code**, then choose **Download Zip**. Unpack the zip file. Go to Blender's Preferences and choose Install From Disk in the Add-ons tab. Find the unzipped folder and choose the **Lodify.py** file. Now the add-on should appear in the add-ons.

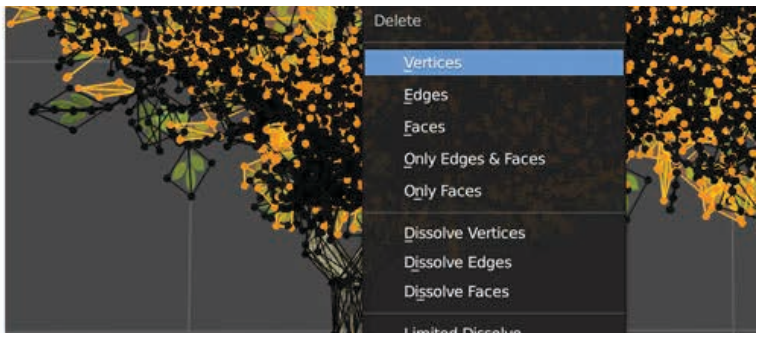




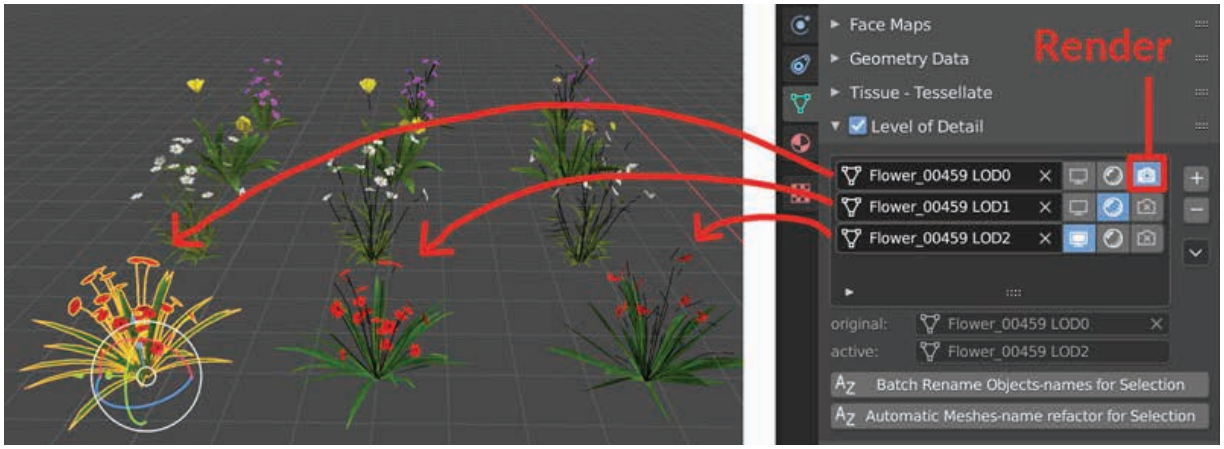
The easiest and fastest way to make these lower levels of detail is to simply use the **Decimate modifier on a duplicate** of the object.

With shrubs, trees and flowers you can also select random leaves or petals in Edit Mode and delete those. To do this to your duplicated object, deselect everything in Edit Mode, then go to **Select > Select Random** and **press Ctrl+L** to select the rest of the randomly selected leaves. Then **press X** and choose **Vertices**.

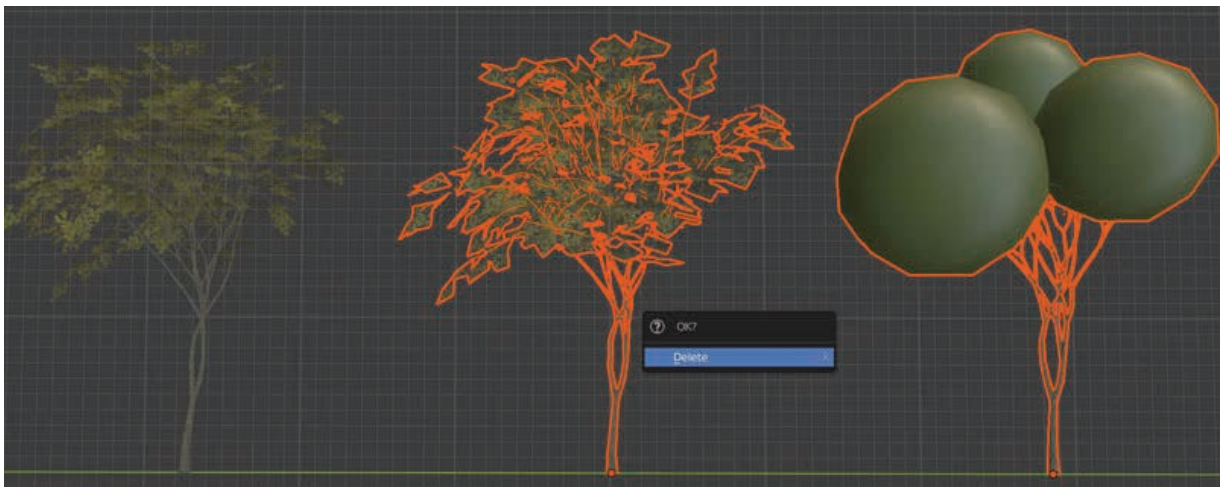




You'll find the add-on in the **Object Data Properties** tab under **Level of Detail**. Select your high quality object, check the Level of Detail checkbox to enable it for that object and **click the + button** 3 times to create 3 slots. Then you can select the high quality object and use the picker to choose the objects for each slot. Then choose which mode that LOD is visible in.



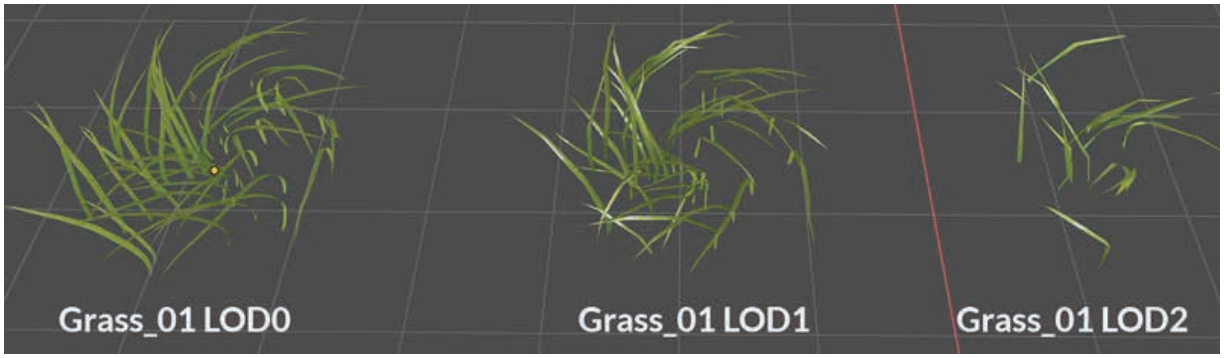
After assigning a mesh to each slot, you can actually delete those LOD1 and LOD2 objects, or you can keep them around in a hidden Collection for safekeeping if you prefer. But the different levels are in fact now stored in the original object. You can even append an object with different LOD levels in a new scene, and it will still work.



Once you've done this for each object, you can work very quickly with low poly models in the viewport, then switch to rendered view to get a better idea of what it will look like, and then finally have the high quality model show up in your renders. Note that with big scenes, it might take a few seconds for the LOD levels to switch in your viewport after switching a preview mode.

In fact, I've noticed that sometimes you need to select some random object before it will switch everything to the correct level of detail.

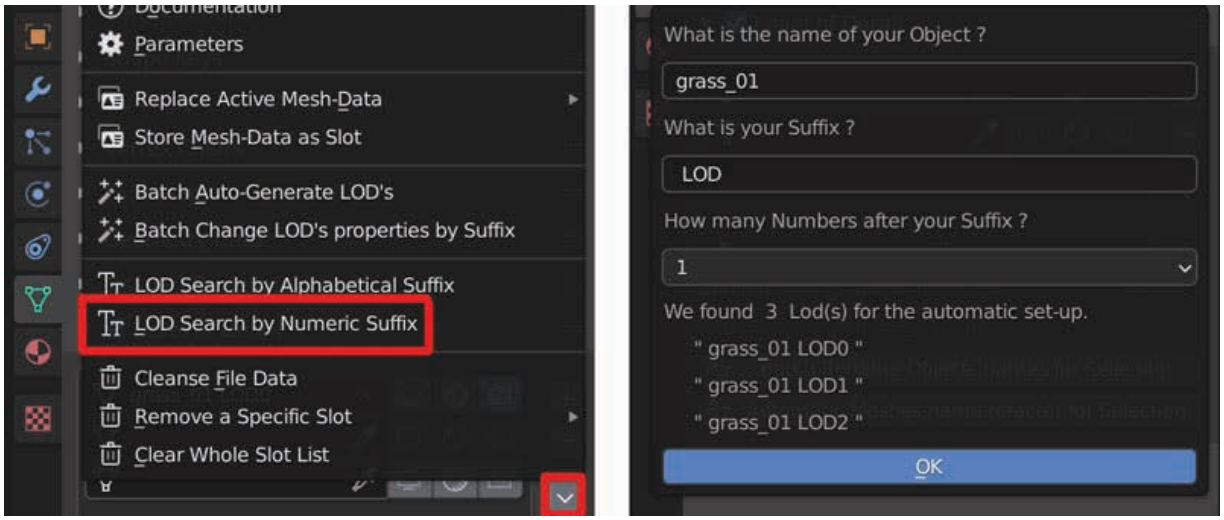
If you have a lot of objects, instead of doing this manually you can let Lodify search for and automatically add LODs in your scene. For that to work, your objects need to have consistent naming. All levels of detail of a grass clump model should be named the same, “Grass” for example, with a suffix like “LOD” and a number (in other words: LOD0, LOD1, LOD2).



LOD levels for a clump of grass

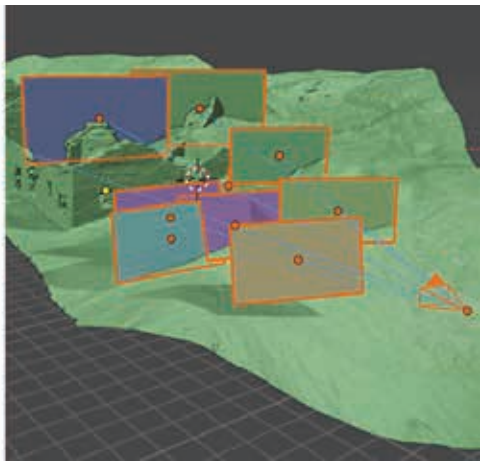
To use this automatic function, click on the dropdown menu in the LOD panel and choose “**LOD Search by Numeric Suffix**”. Then you enter the name of the object, the suffix *(be careful: if there’s a space in front of the suffix, type that too)*, and now many numbers are after the suffix (normally just 1).

Then after it’s done its thing **click OK**. Now it’s added the other LOD levels to the object and you just have to set up when each should be visible.



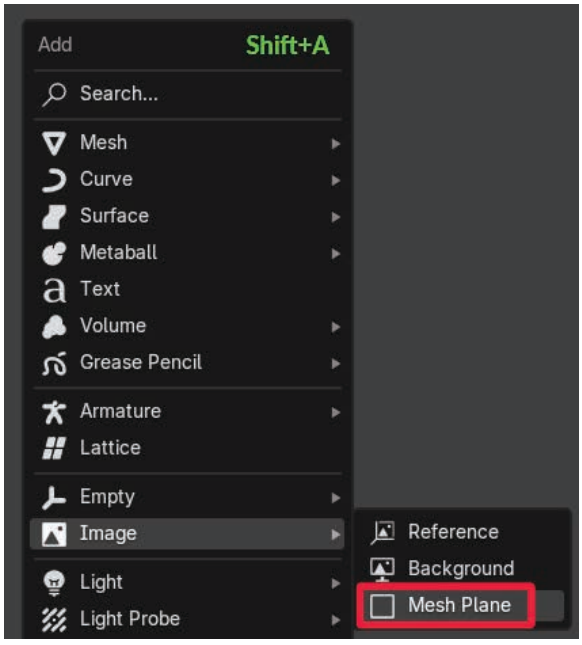
Proxies for Image Sequences on Planes

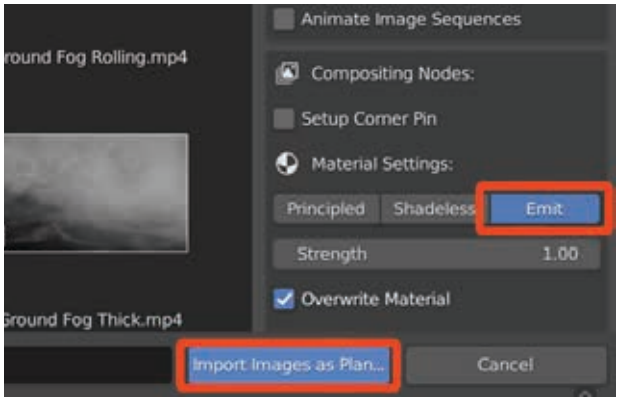
Fog and smoke can add a lot of interesting mood, not to mention photorealism to your scene. In this case, let's add some free videos of fog to a photoscanned scene found on Sketchfab.



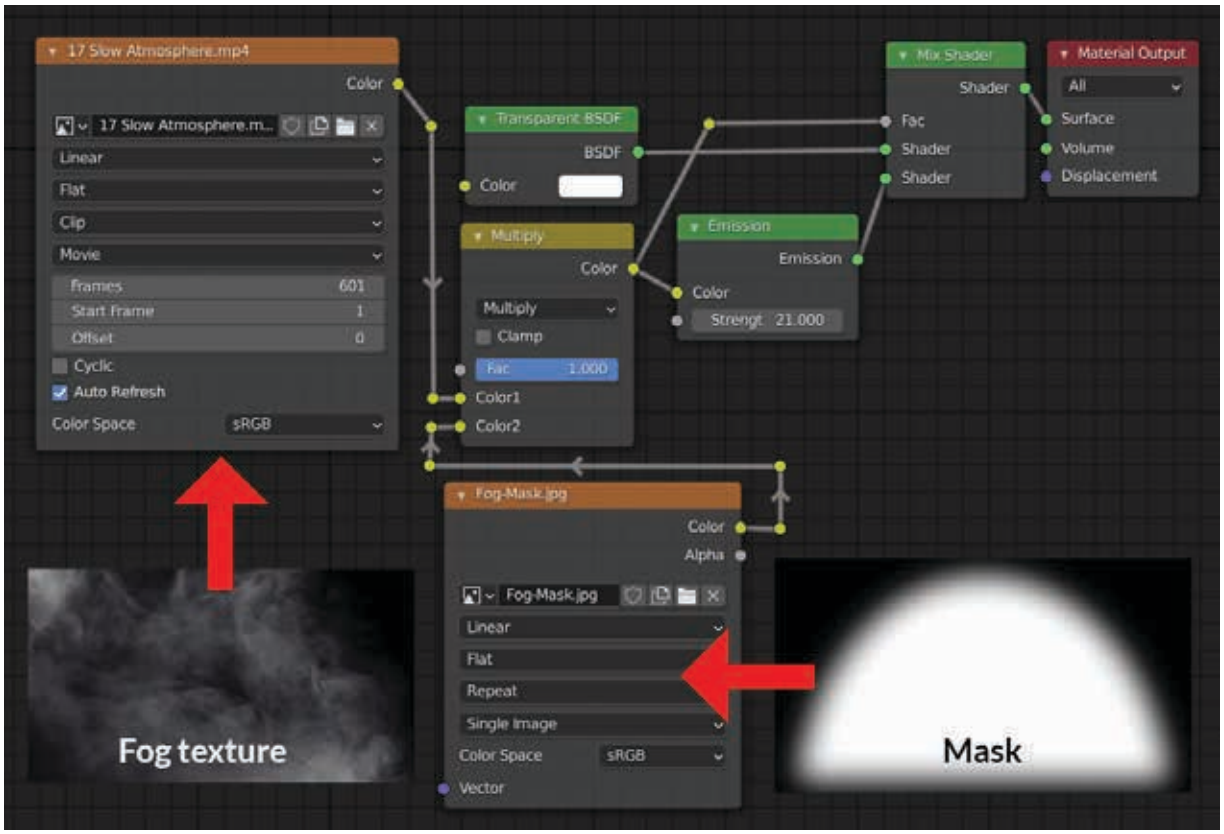
"Surrender Lead Ore Smelt Mill" by Altogether Archaeology is licensed under Creative Commons Attribution.

By using the Mesh Plane option, I can import one of the free 4K fog videos from Premiumbeat. I just press **Shift+A > Image > Mesh Plane** and choose one of the files. They are video files, but that works as well as an image. It's automatically set to the correct frame rate. On import you can choose to use an Emission material instead of the default Principled BSDF.





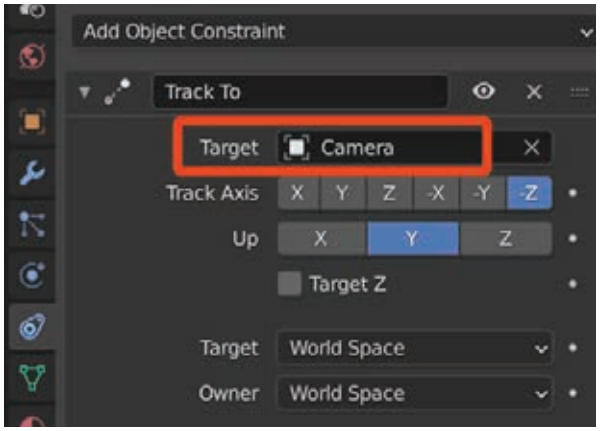
The fog in the footage goes outside of the frame, so in order to avoid it looking a bit strange I created a black and white mask that I then used as an image texture multiplied with the fog texture, like so:



You can see the difference below: on the left, the image plane without the mask. The fog gets cut off at the edges of the plane. On the right, the mask causes it to fade towards the edges.

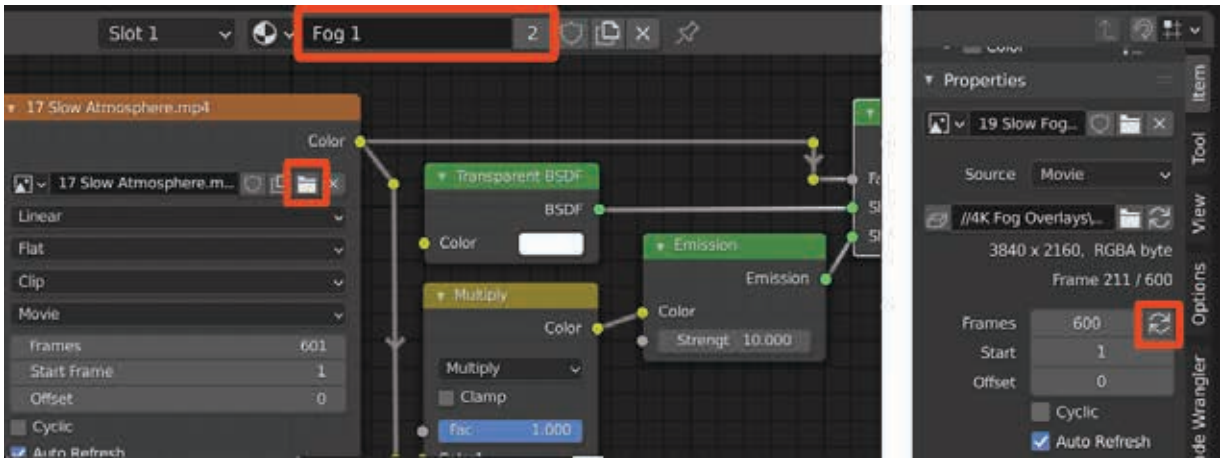
You may need to increase or decrease the Emission Strength a bit depending on the footage.

After this, you can add an Object Constraint to the Plane with as the Target the Camera, so that the Plane always faces the camera.

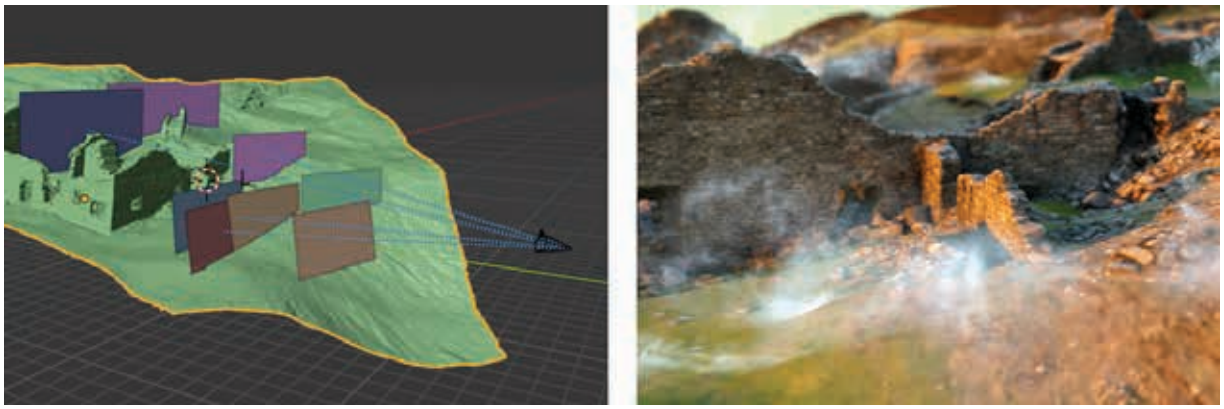


Now if I select the Plane and **press Shift+D** I can duplicate it and move it somewhere else. The Object Constraint to the Camera keeps working. For some variation, let's add a different fog video to the duplicate. In the Shader Editor click on the number next to the Material name.

This creates a new material based on the existing one. Type a new name for it, like "Fog 2" for example. Change the file as well. In the Properties of the Item tab you can update the Frames count.

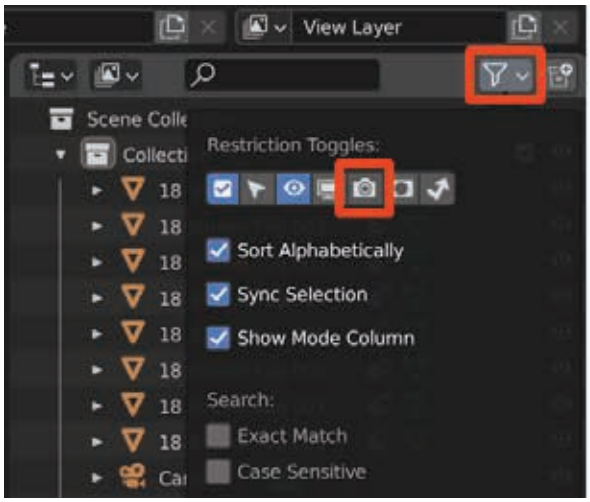


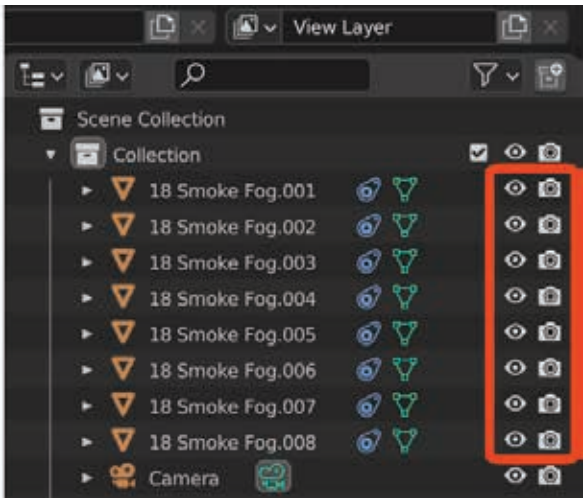
Now we've got a second plane with different fog. We can add a bunch of other planes this way. Each time you may have to adjust the Emission Strength based on the individual fog footage you use.



After about 6 of these 4K fog planes however, I start to notice my computer slowing down. I only get about 2 frames per second playback in Material View. Which is no surprise, all these 4K video textures are quite heavy to store in memory.

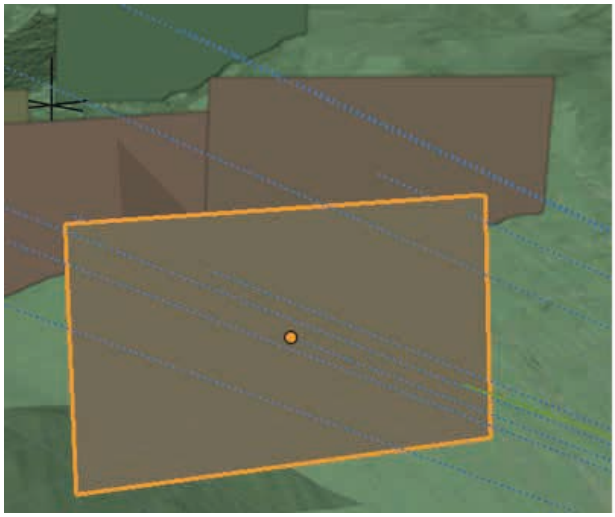
In fact this might happen much sooner depending on your computer specs. So what do we do about this? First of all, make sure you can enable and disable render visibility in the Outliner.

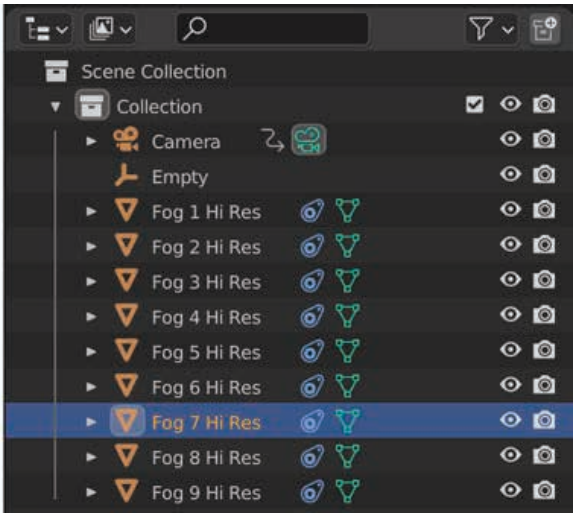




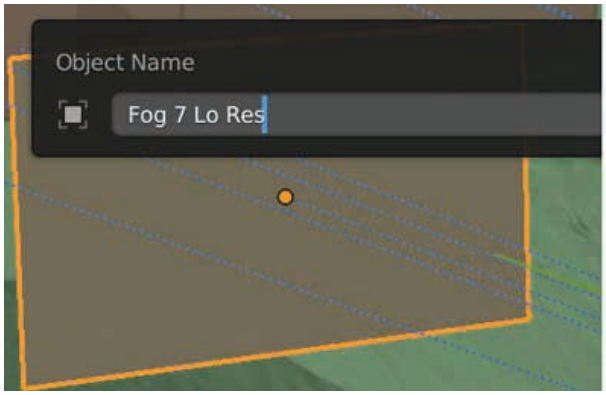
Now we can create lower quality duplicates or "proxies" of the image planes, that we parent to the originals. The original is disabled in the viewport, and enabled for rendering. (Watch a video explanation of this part here)

The proxy is set up the opposite way: enabled in the viewport, and disabled for rendering. Make sure you name them clearly to avoid confusion. Now to make a proxy select a plane, **press Shift+D** to duplicate it.



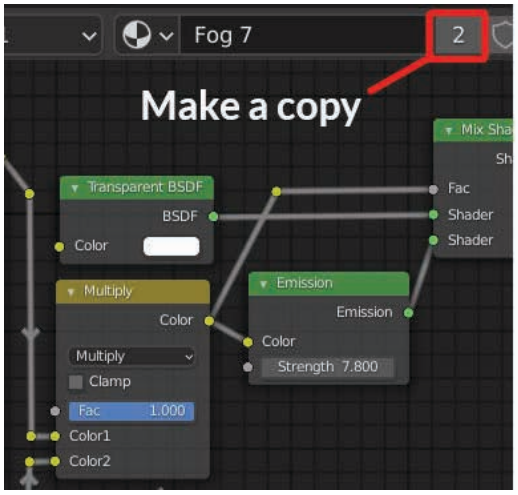


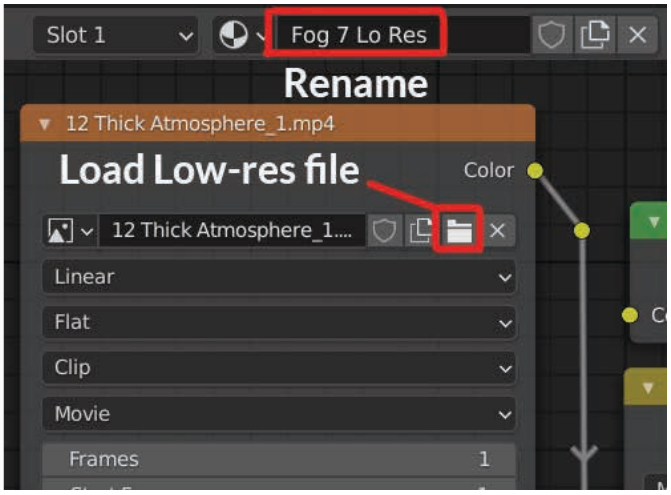
With the duplicate still selected, left-click to confirm the duplication action and then **press F2** to rename the duplicate. Add the affix "_proxy" or "Lo Res" to it (for example) so that you know it's the proxy version.



Then in the material of the proxy object, make a copy of the material by clicking on the number next to the name and add a lower resolution version of the texture. Give the copied material an affix like “Lo Res” as well.

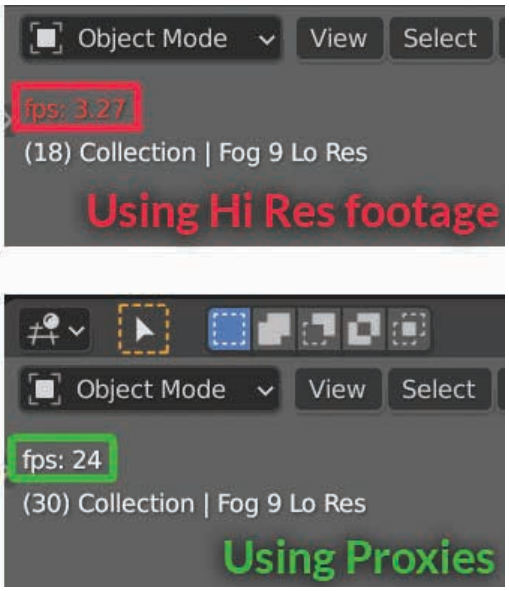
The lower resolution texture can be either a still or a really low resolution video. I convert my videos in Adobe Encoder, but if you’re looking for a free alternative, you could do worse than to use Mpeg Streamclip (which is open source as well!).



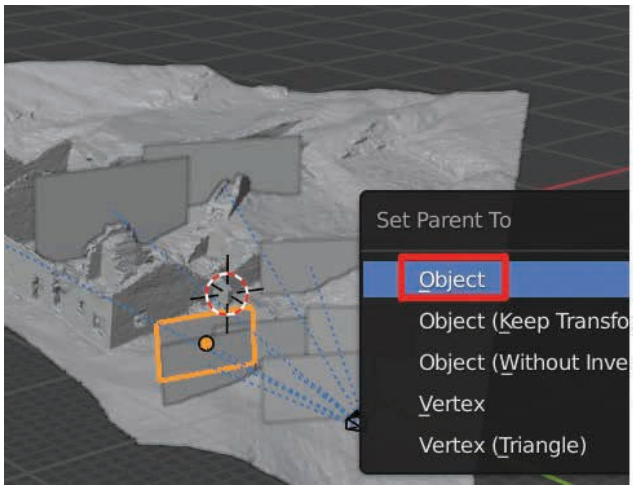


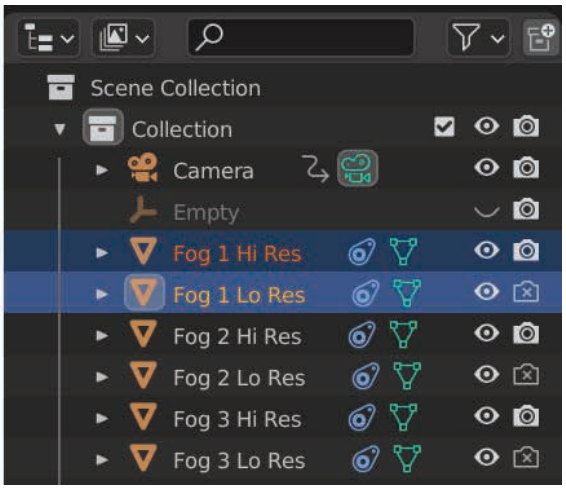
Repeat these steps until you’ve made proxies for all the image planes. Now set these proxy objects to be invisible in render, but visible in the viewport and set the high quality objects to be visible in render, but invisible in the viewport. You’ll see that the viewport now responds much faster *(in Eevee, you won’t notice much difference in the Cycles rendered preview).*

As you can see, with the Hi Res image planes I’m only getting about 3 frames per second of playback in my viewport. With the proxies, I easily reach the goal framerate of 24 fps (and could probably go higher if I wanted to).

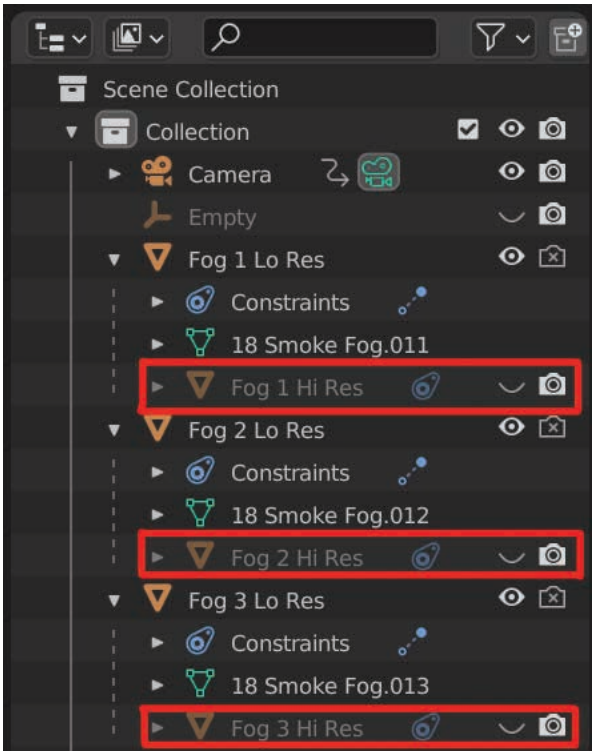


If you want to be able to move the image planes, you'll need to parent the high res one to the proxy. For that, the high res one needs to be temporarily set to be visible in the viewport (otherwise you can't select it). Then, select the low res one, **hold Shift** and select the high res one. Now with the mouse cursor hovering in the 3D viewport, **press Ctrl+P** and choose Object.





Now when you expand the low res image plane in the Outliner, you'll see that the high res one is parented to it. Turn off viewport visibility for it again.



NON-PHOTOREALISTIC RENDERING

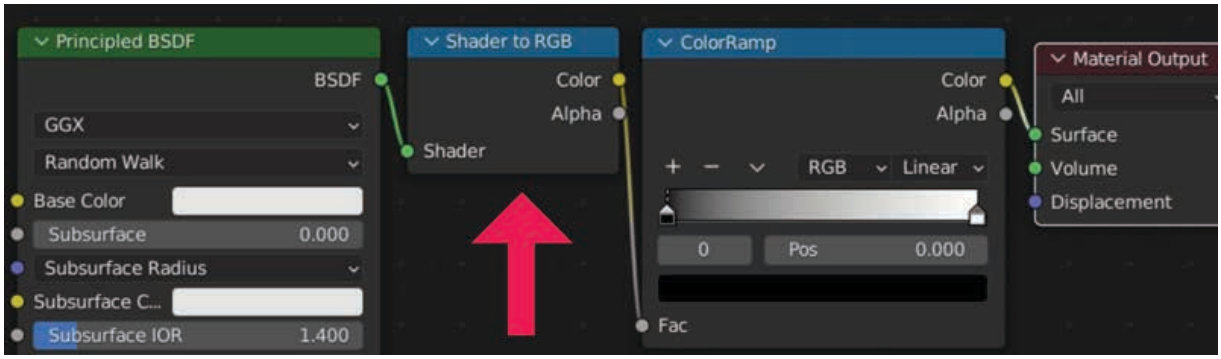
Basic Toon Shader

Creating a basic Toon shader is easy if you follow these steps.

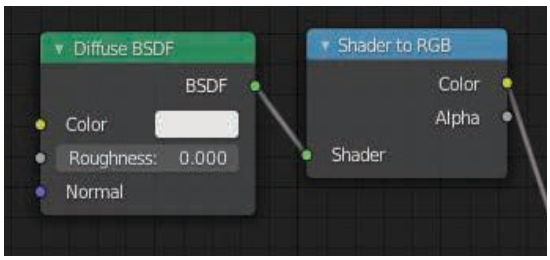
You can add as many colors to it as you want to create a stylized look.

Note, that the shader described here works only in Eevee because it uses the **Shader to RGB node**.

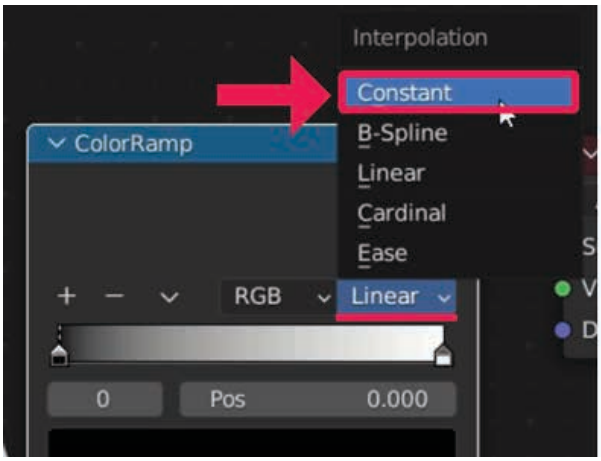
Make sure your object is shaded smooth by right-clicking on it in Object Mode and choosing Shade Smooth. Then, add a Shader to RGB and ColorRamp node to its material.

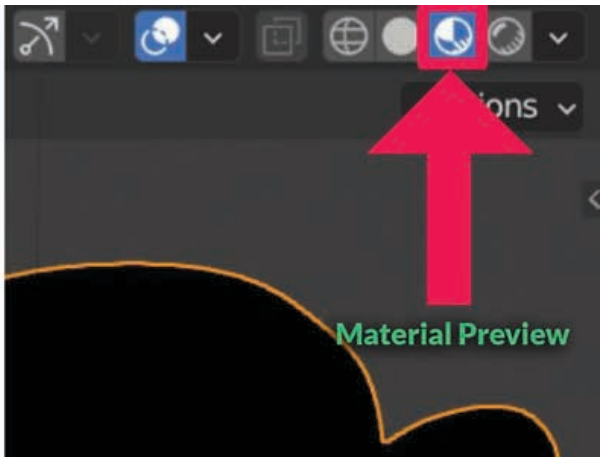


Instead of a Principled BSDF, you can also use a **Diffuse BSDF**.

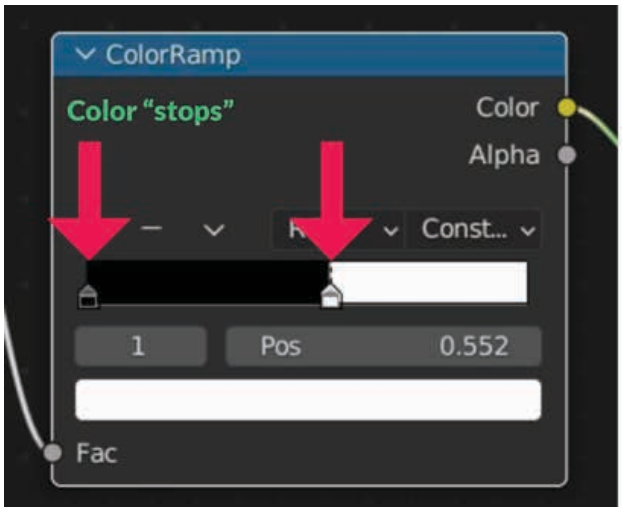


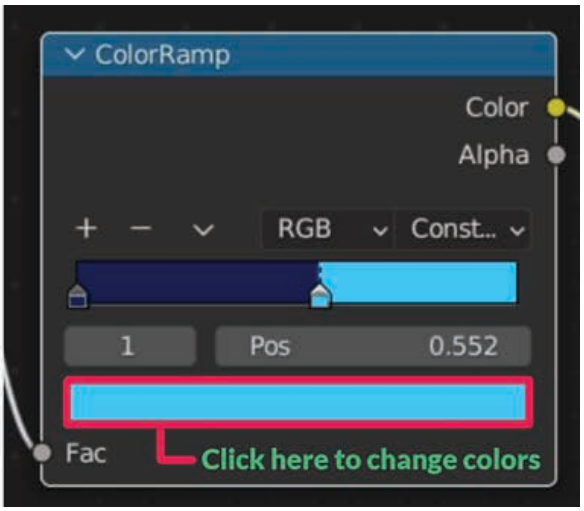
Change the ColorRamp node from Linear to **Constant**. Make sure you are in Rendered mode in Eevee (**press Z** and choose **Rendered** in the viewport) or **Material preview** to see what you are doing.



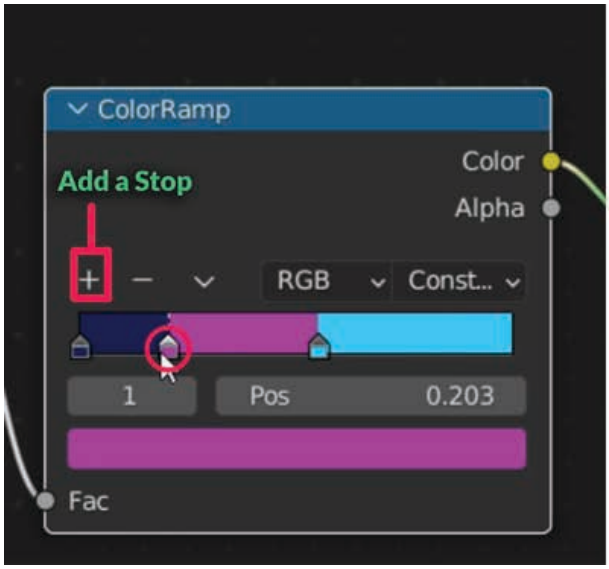


By clicking on a **stop** (the little pointers on the ColorRamp) you can then change the color in the field below.

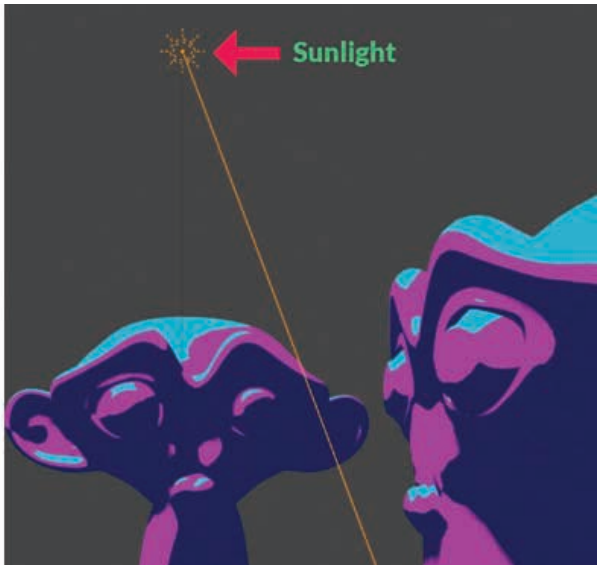




Click on the [+] button to add more colors if you like. In Rendered view, you will need to add a Light.



The position of the Light (a **Sun** in this example) influences the way the material looks as well.



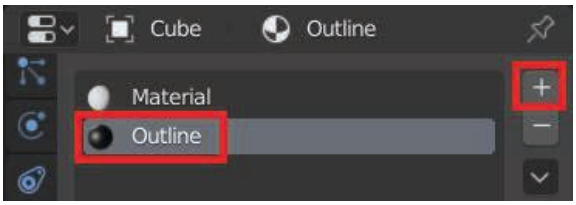
Easy Outline with inverted hull

There are several ways to create a cartoon-like outline around an object in 3D. A popular method (especially for games, since it works in real-time) is the so-called “inverted hull method”.

It creates a duplicate of the model which is slightly bigger, has flipped normals and is the color of the outline.

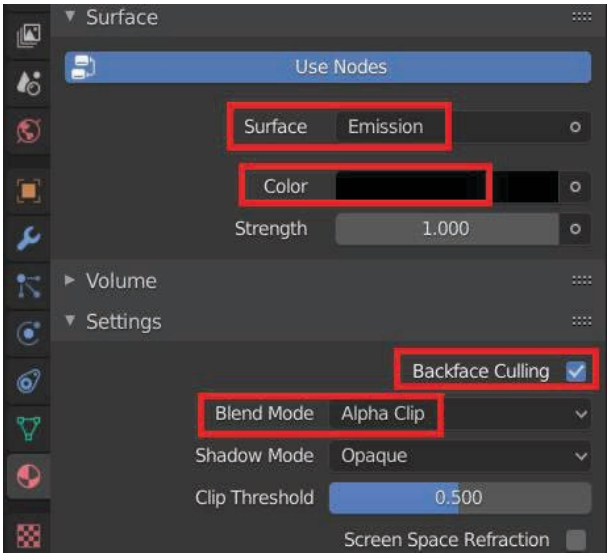
Note, that this method is for Eevee.
A Cycles-specific version can also be found in this book (after this topic).

In Blender we can do this very easily. Create a second material by **clicking on the [+] icon** and name it Outline.



For the Outline material, set it to either Emissionor Toon BSDF and make the colorblack.

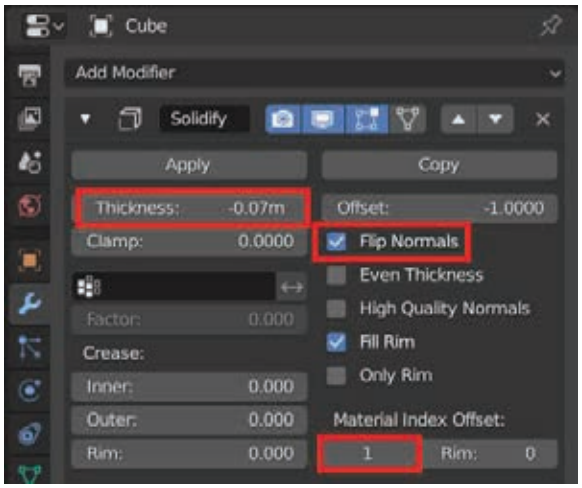
In material Settings turn on **Backface Culling** (backface culling allows us to see the flipped normals) and as Blend Mode set it to **Alpha Clip**.



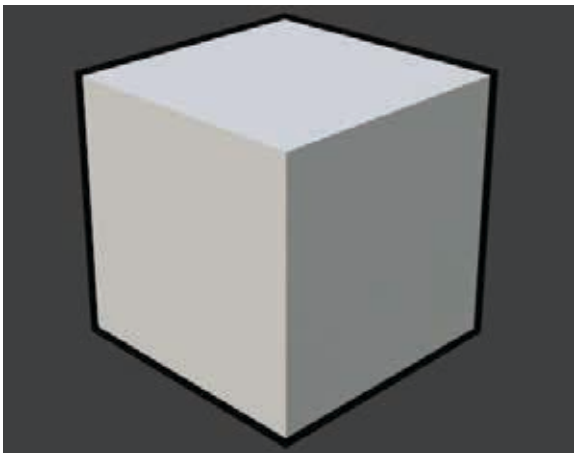
Add a **Solidify modifier** and check **Flip Normals**.

Flipping the normals turns the model inside out.

Set **Material Index Offset** to **1** (this forces the outline to use the second material of the object, in this case the black outline).

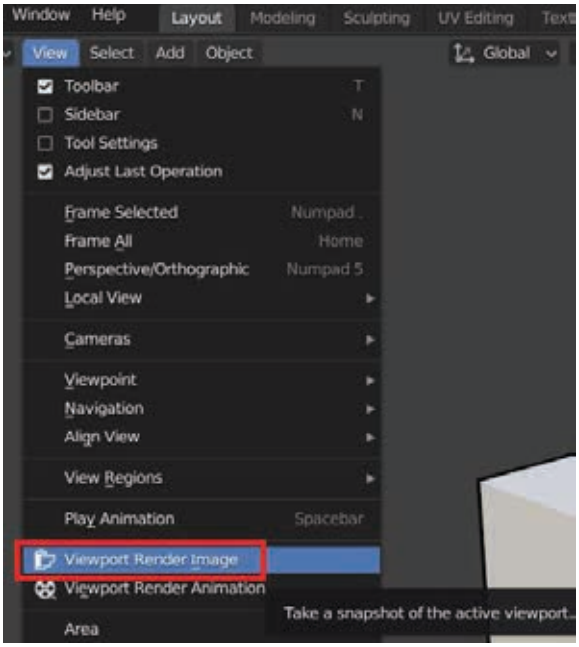


In Material Preview mode you should now see the outline.

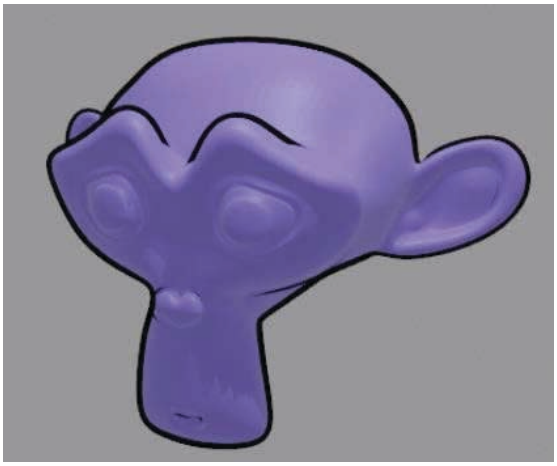


If you don't see the outline, try changing the Thickness value of the Solidify modifier.

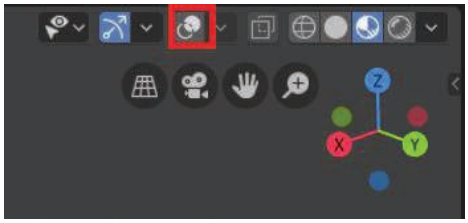
To render, go to View and choose **Viewport Render Image** (or Animation, if you want).



Works on monkeys as well...

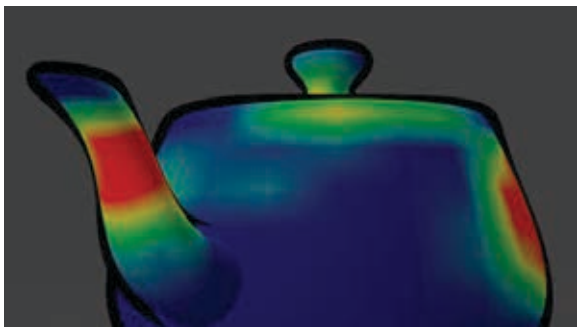


If you want to do a viewport render you may want to turn off **Overlays**, as those also show up in a viewport render.



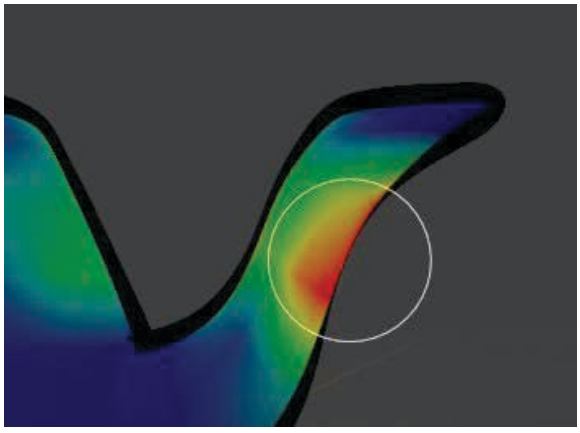
Control outline thickness

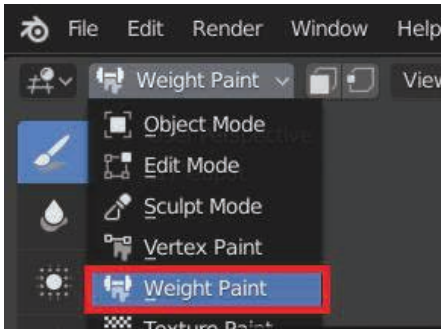
In the previous pages we looked at creating a Toon Outline with the inverted hull method.



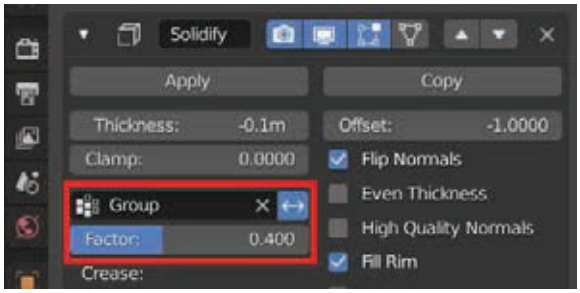
Now we will go one step further and control the variable thickness of the outline using Weight Painting.

By switching to **Weight Paint** mode we can paint the thickness map of the outline. Make sure Overlays are turned on, otherwise you can't see the weight painting.





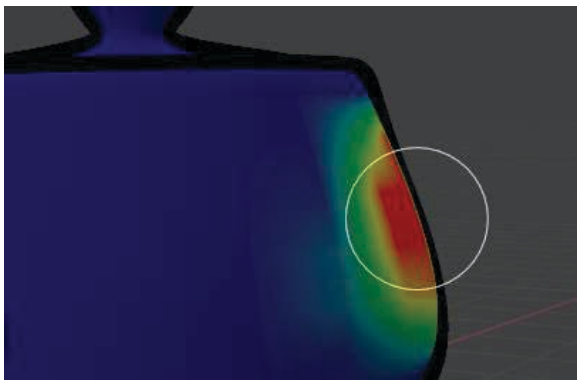
Choose the new **Group** (this was automatically created when we did our weight painting) in the **Solidify** modifier.



In case you want to inverse the effect (thick/thin) click on the **double arrow ←→icon** next to the Group field.

Factor controls the influence of the weight painting on the thickness.

Once the group is loaded in the modifier, you can Weight paint interactively.

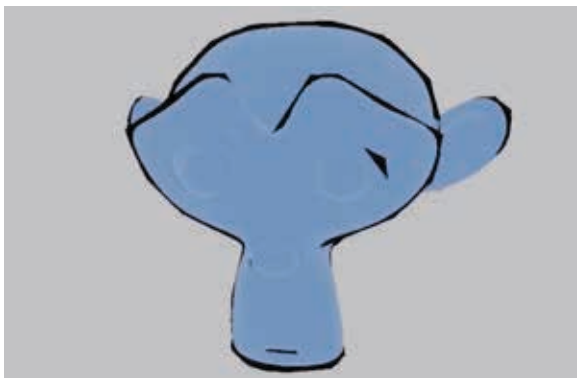


Sketchy moving toon outline

We already learned in a previous section about the “inverted hull method” for creating a real-time outline.

And we also looked at creating a variable thickness outline.

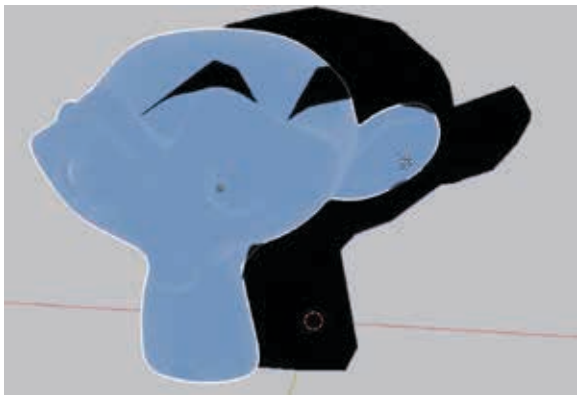
What if you want to animate that outline, to create a really sketchy, hand-drawn animation look? Read on!



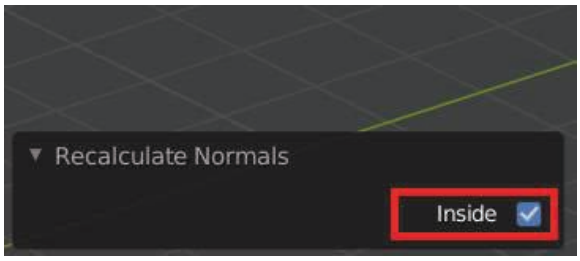
This technique uses a second, parented duplicate object which is completely black.

In other words, we don’t simply use a Solidify modifier like before but actually have two objects.

The second object has flipped normals.

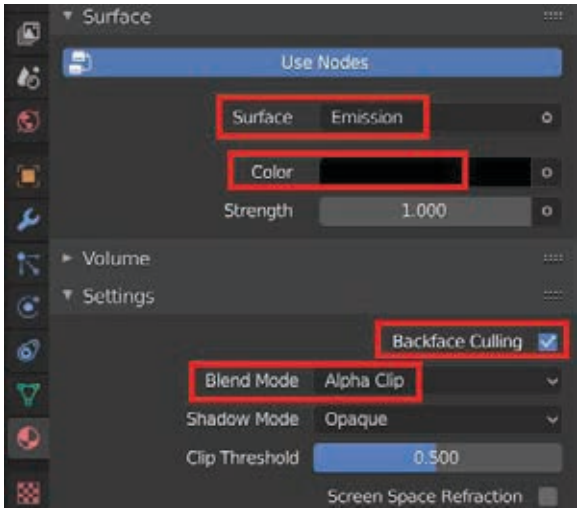


To do this, select all its Faces in Edit Mode and **press Shift+N**, then choose **Inside**. This flips the normals to the inside of the object.



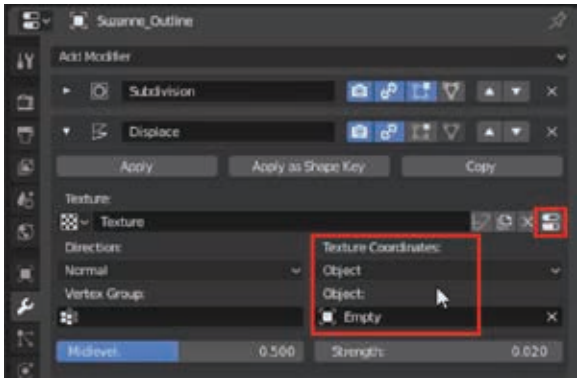
For the Outline material, set it to either Emission or Toon BSDF and make the color black.

In material Settings turn on **Backface Culling** (backface culling allows us to see the flipped normals) and as **Blend Mode** set it to **Alpha Clip**.

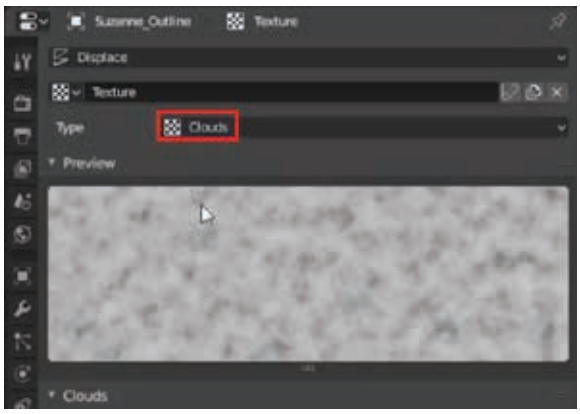


The Outline object has a Displacement modifier on it.

Instead of UV's, we use another object to give this material its **Texture Coordinates**.

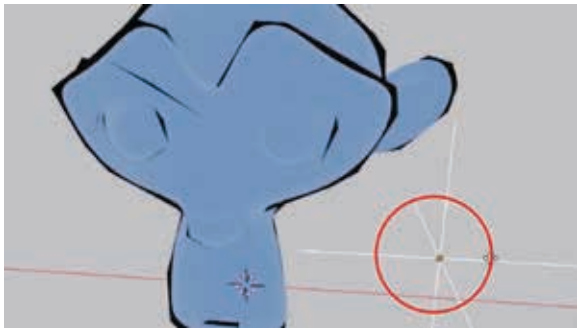


For the texture, we use a simple procedural **Clouds** texture.

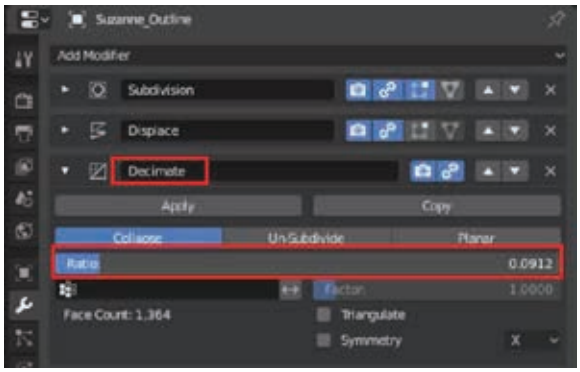


An **Empty** drives the movement of the Displacement modifier.

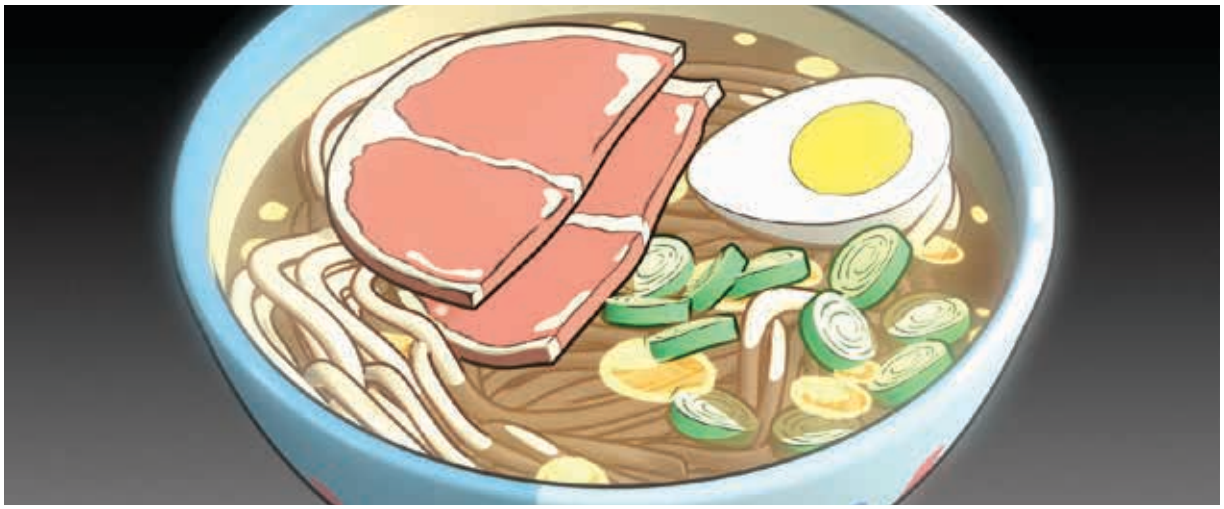
To add an Empty, **press Shift+A** and choose any kind of **Empty**. When you move the Empty, the outline will change, as long as we use a Decimate modifier.



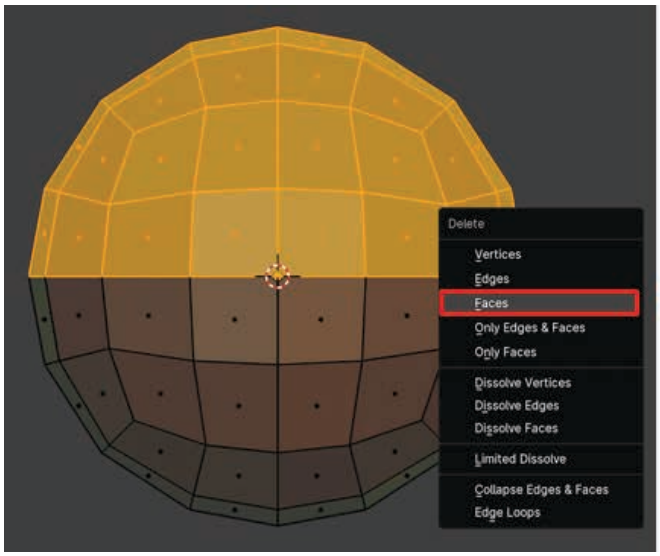
The **Decimate modifier** (placed underneath the Displace modifier) makes the outline "sketchy", especially when the Ratio of decimation is ridiculously low.

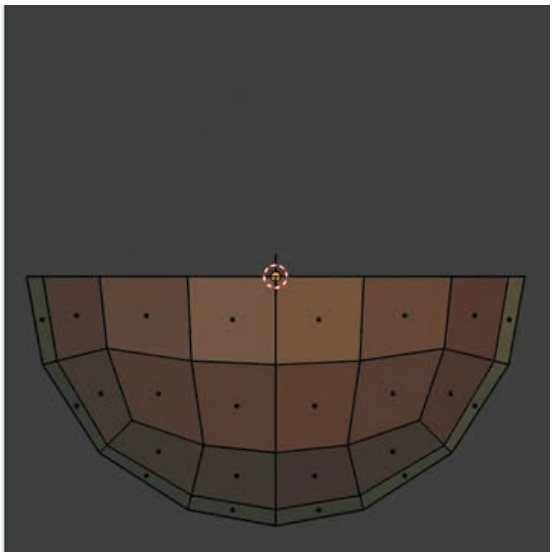


Anime Style Ramen in Blender

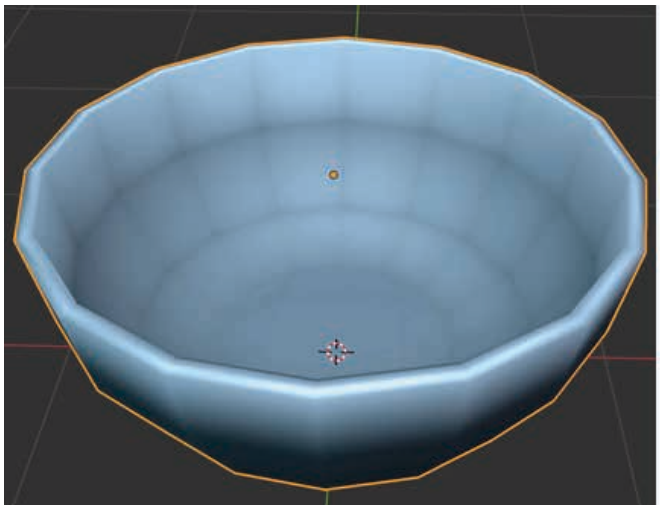


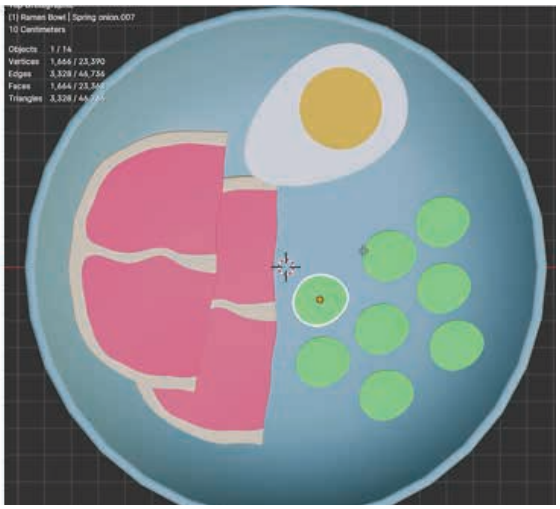
First, model a bowl and some toppings. You can start by adding a **Sphere** primitive, then deleting the top half of Faces in Edit Mode.



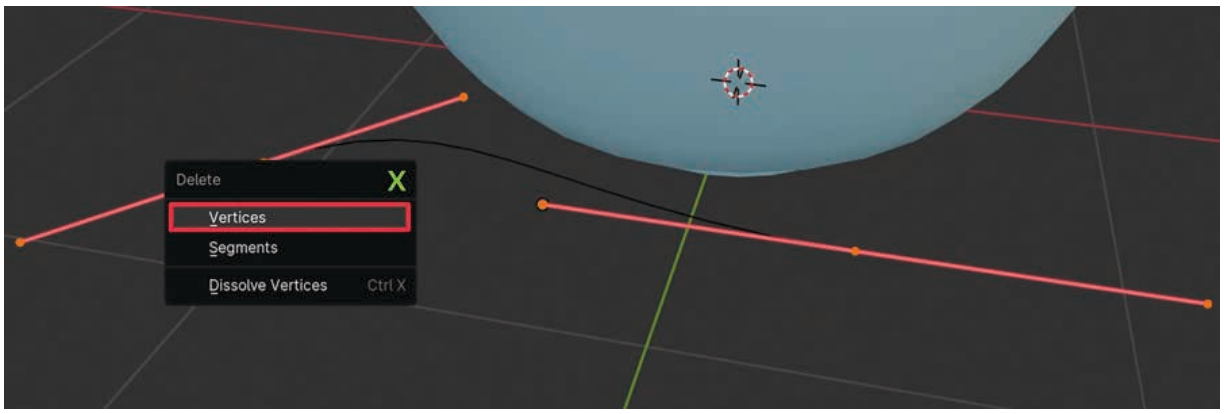


Add a **Solidify** modifier to the bowl to add thickness. In fact, you can do the same for the meat and pieces of green onion. Create them flat first, then add thickness. Duplicate the toppings as needed with Alt+D.



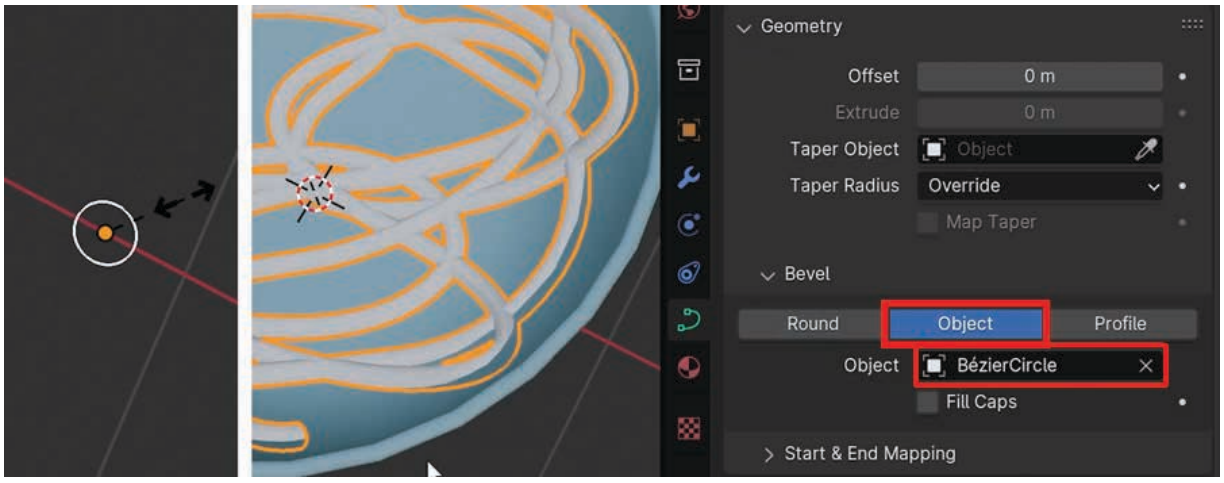


For the noodles, add a **Bezier Curve** (Shift+A > Curve > Bezier). In Edit Mode, select and **delete** all Vertices of the Curve. Now the Curve still exists, but has no Vertices.



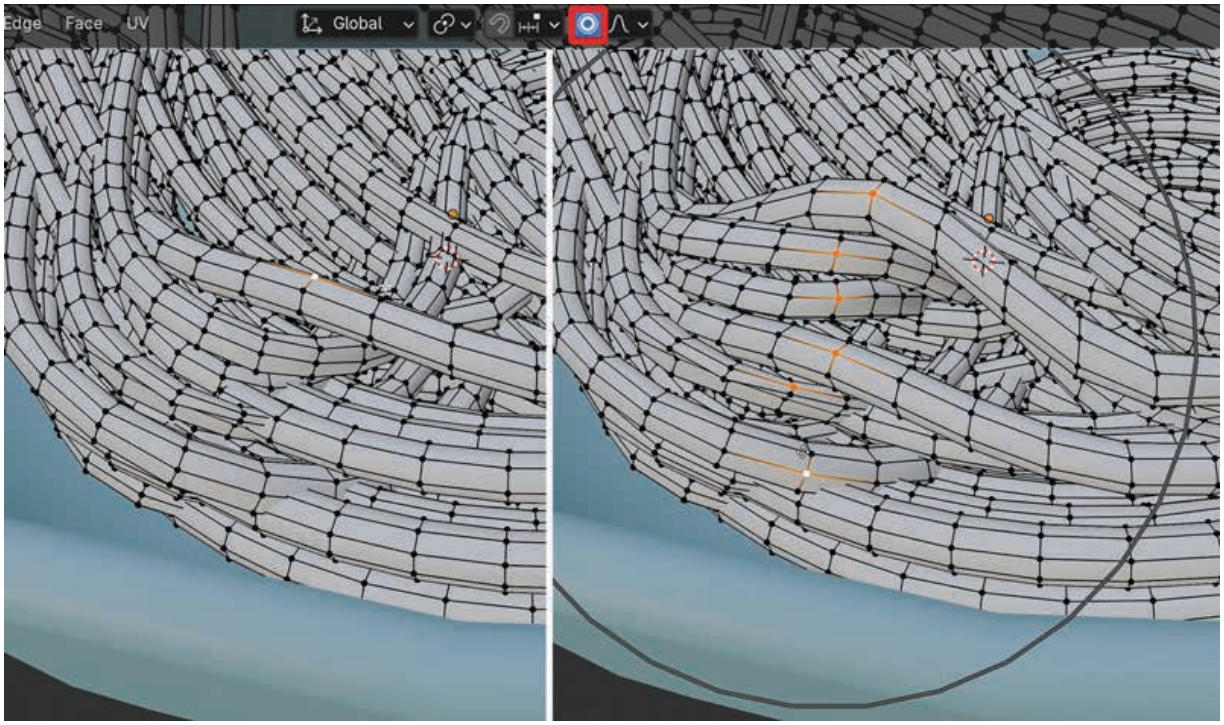
Using the **Draw** Tool in Edit Mode, draw some Noodles on the bowl. With some practice, you can do this with the mouse. It doesn't have to be perfect: messy chaos is fine!

As the **Bevel** object for the noodles Curve, add a **Bezier Circle**. Scale down the Bezier Circle to control the thickness of the noodles.

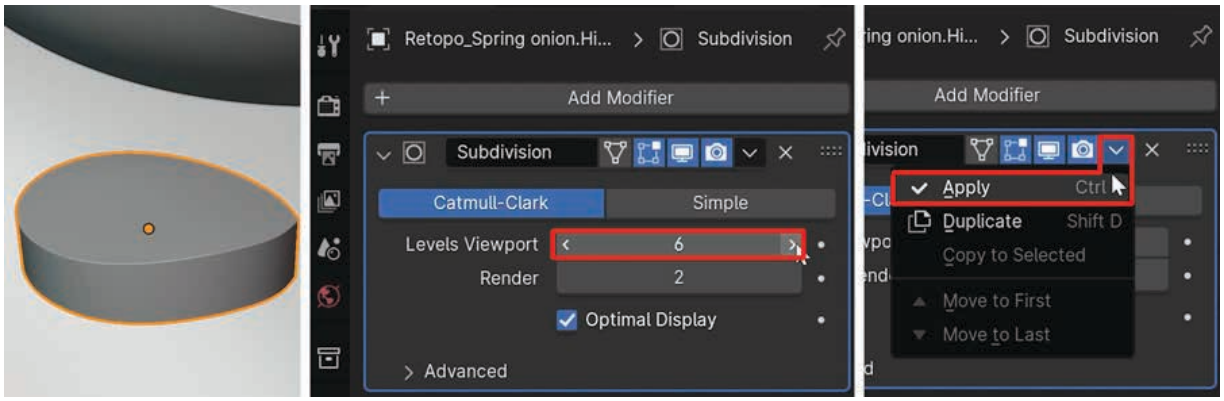


Duplicate the Circle in Edit Mode a few times (Shift+D) to create more noodles along the same Curve.

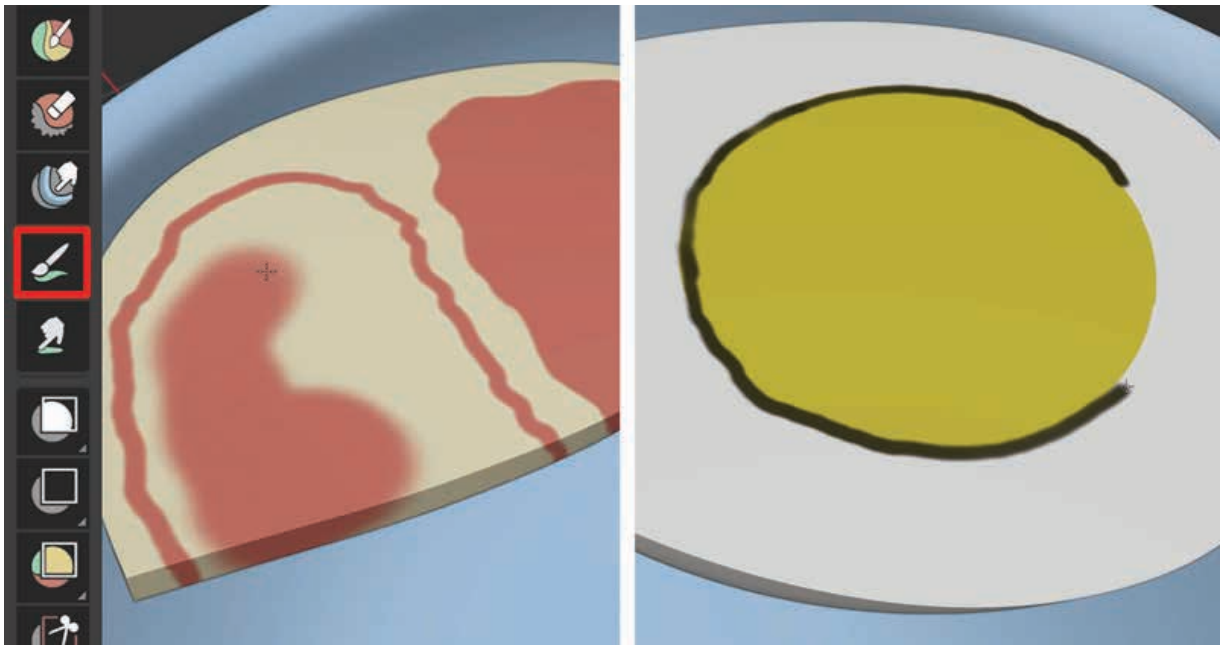
With **Proportional Editing** (shortcut: the letter "o") you can adjust the noodles so they look better and don't intersect as much. Just select one Vertex on a noodle that you want to move, and move it with the shortcut **G** (Grab).



Duplicate the models and increase the geometry so you can paint on them in Sculpt Mode. You can keep the originals in a hidden Collection. I recommend keeping the low-poly versions so you can bake textures on them later. Add a **Subdiv modifier** to the duplicates and **apply the modifier** after increasing the levels.

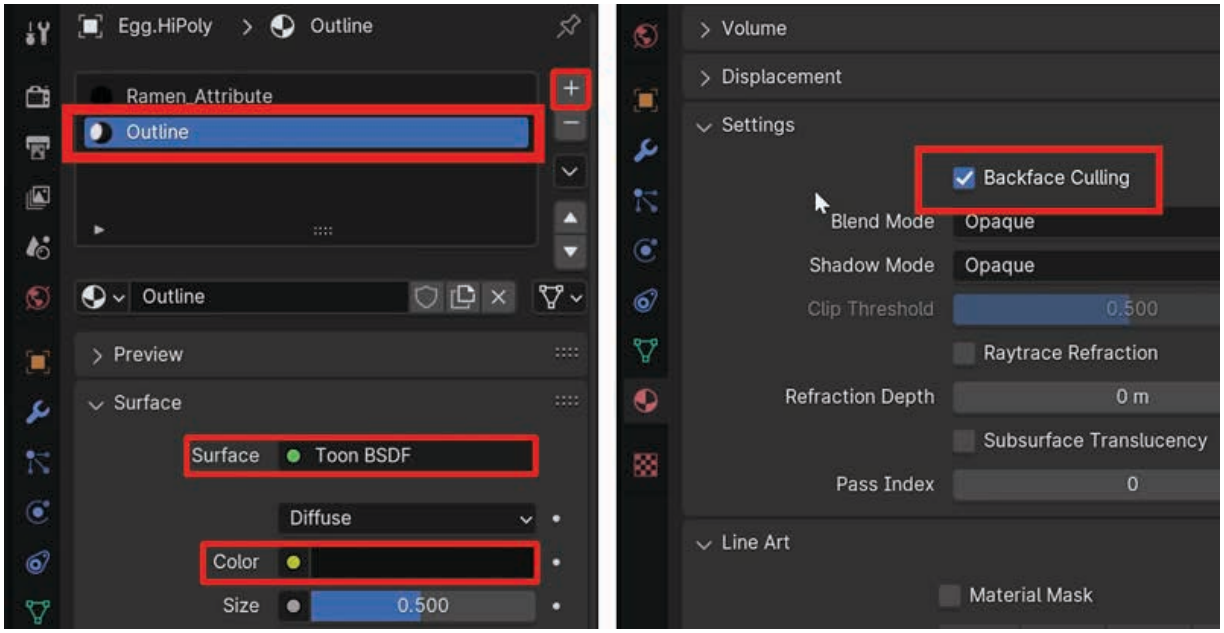


Using the **Paint** brush in **Sculpt Mode**, you can paint the high resolution duplicates.



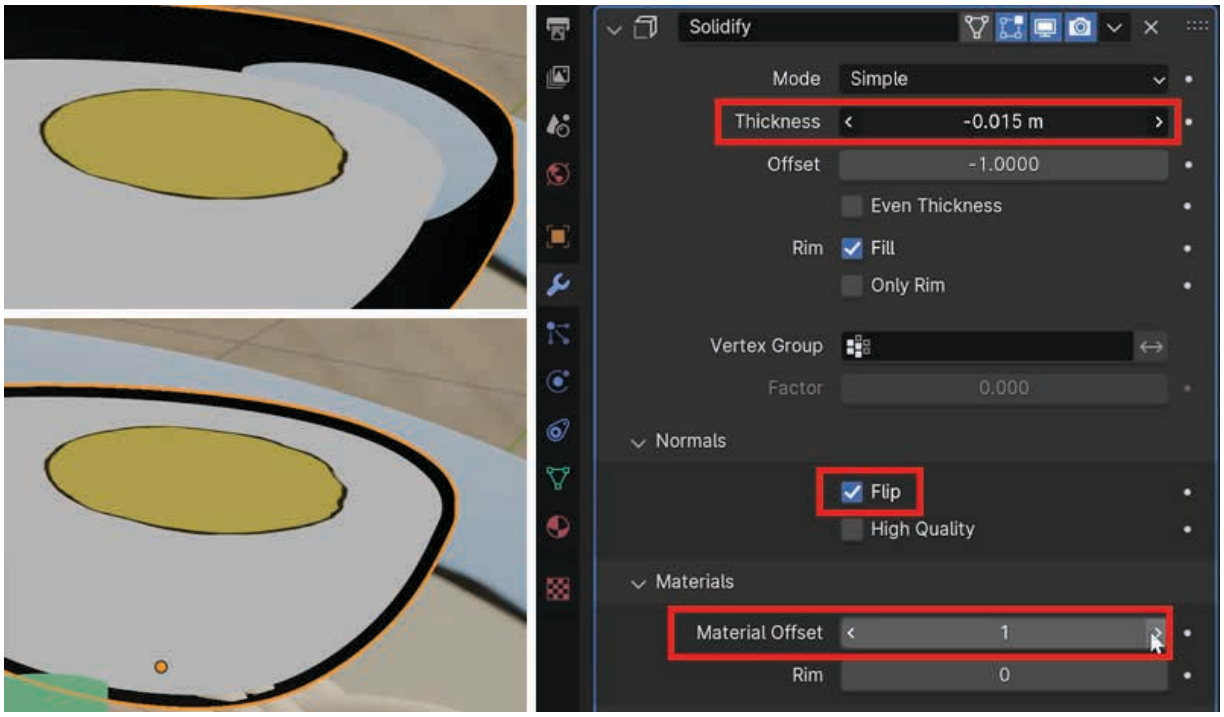
Create an **Outline material** and enable **Backface Culling**. That option makes sure that we can see the material "from behind", in other words after the normals are flipped.

As the type of surface you can use the **Toon BSDF** shader. The color can be black, or some other color depending on the style you're going for.

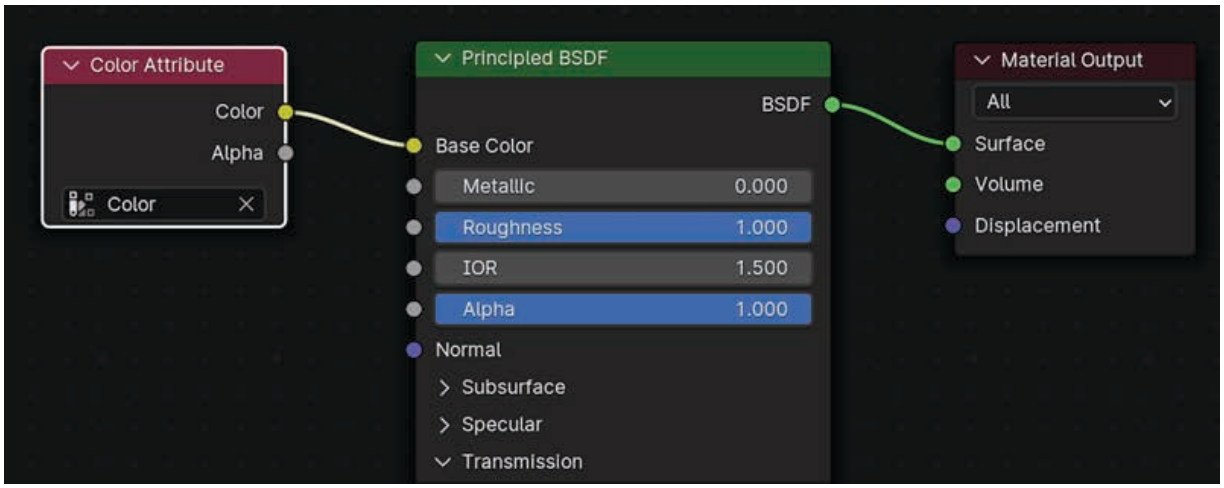


Add a **Solidify** modifier, **flip** the normals (so the thicker part of the object is “inside out”) and offset the material. By **offsetting** the material to 1, it uses the second material which is the Outline material you’ve just created.

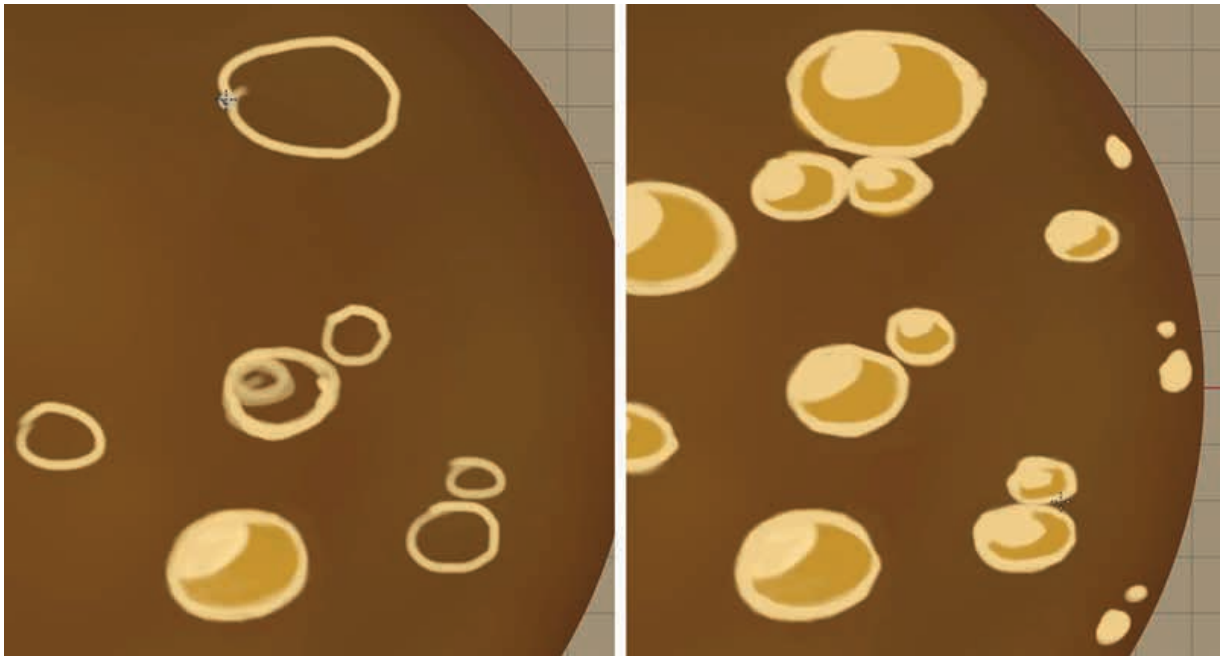
By increasing the **Thickness** value, you adjust the thickness of the outline as needed.



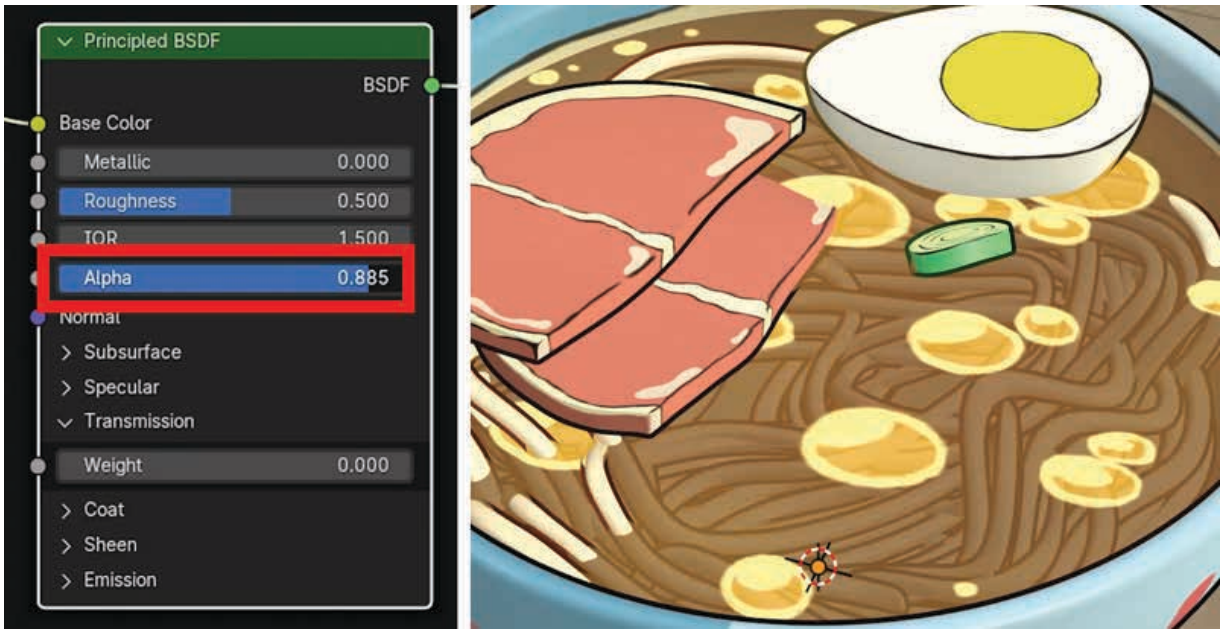
To see the paint in rendered mode, add a **Color Attribute node** to the material as the Base Color.



For the soup, paint the greasy bubbles with the Paint brush in Sculpt mode.



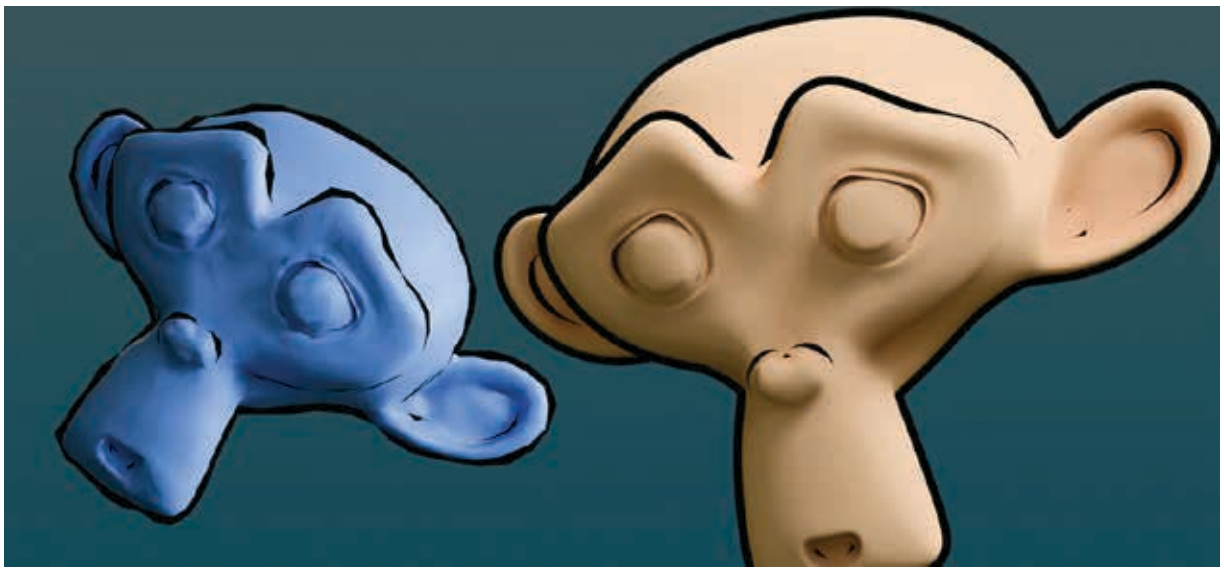
Then reduce the **Alpha** value to make the soup somewhat transparent.



Inverted Hull Outline in Cycles

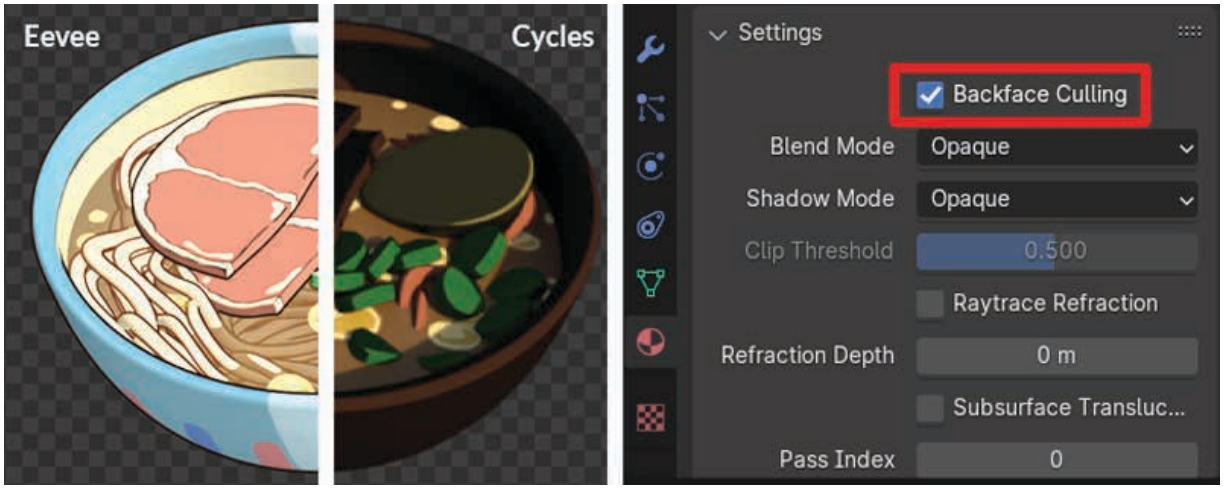
Previously we've explored how to create a toon outline using the inverted hull method in Eevee.

But what if you want to take advantage of some of the benefits of Cycles?



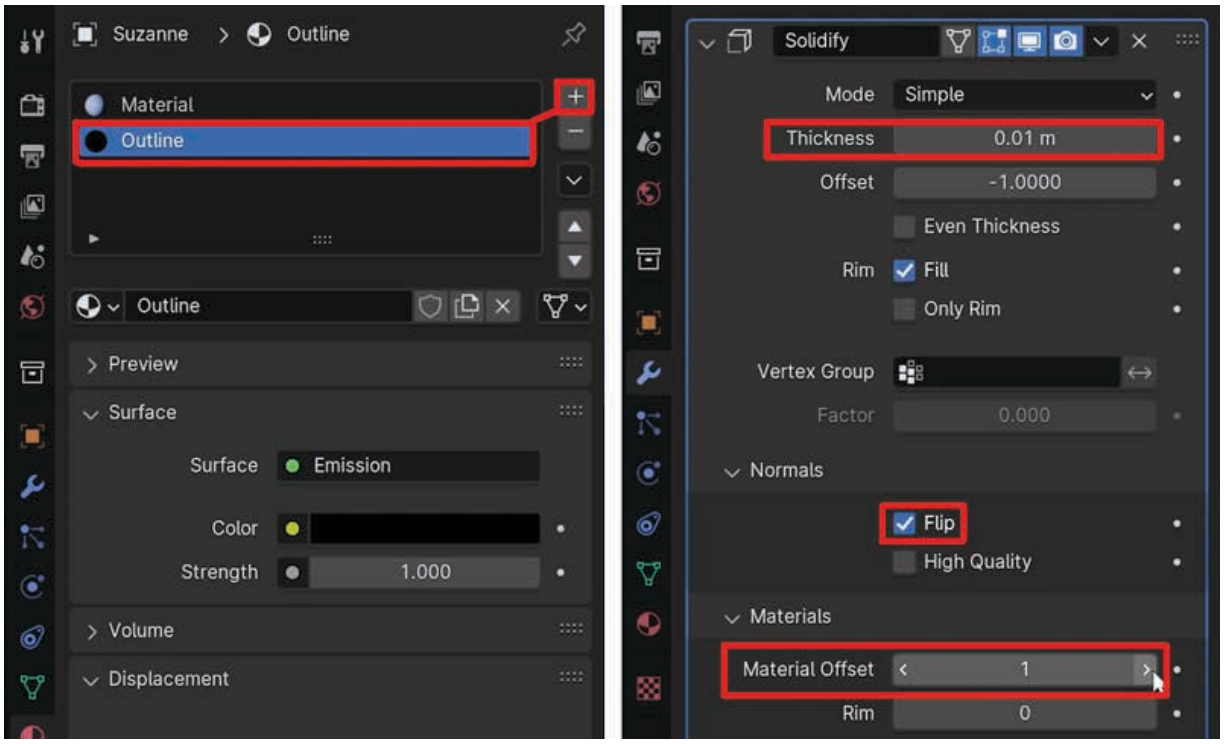
Unfortunately, unlike in Eevee, you can't enable Backface Culling in a material in Cycles. So the Eevee method doesn't work.

However, with some additional shader nodes you can also achieve the inverted hull outline in Cycles.



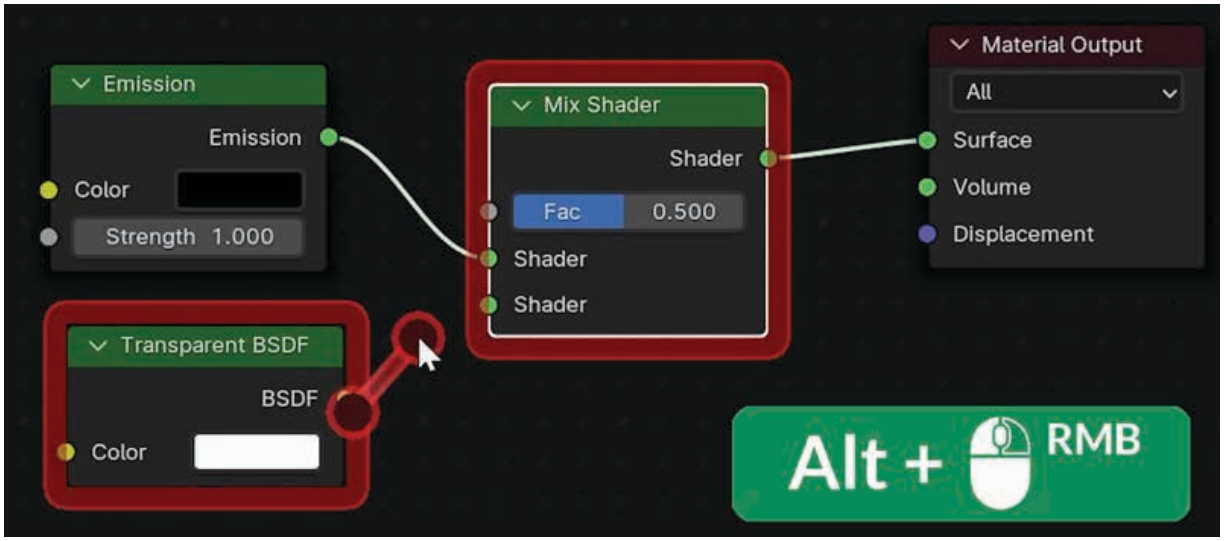
Just like with the Eevee method, add a **second material for the outline**.

Then use a Solidify modifier, under Normals enable **Flip** and **offset the Material**. So far, it's all the same as in Eevee.



Now open the second material in the Shader Editor. Mix an Emission material with the Outline color you want, with a Transparent BSDF.

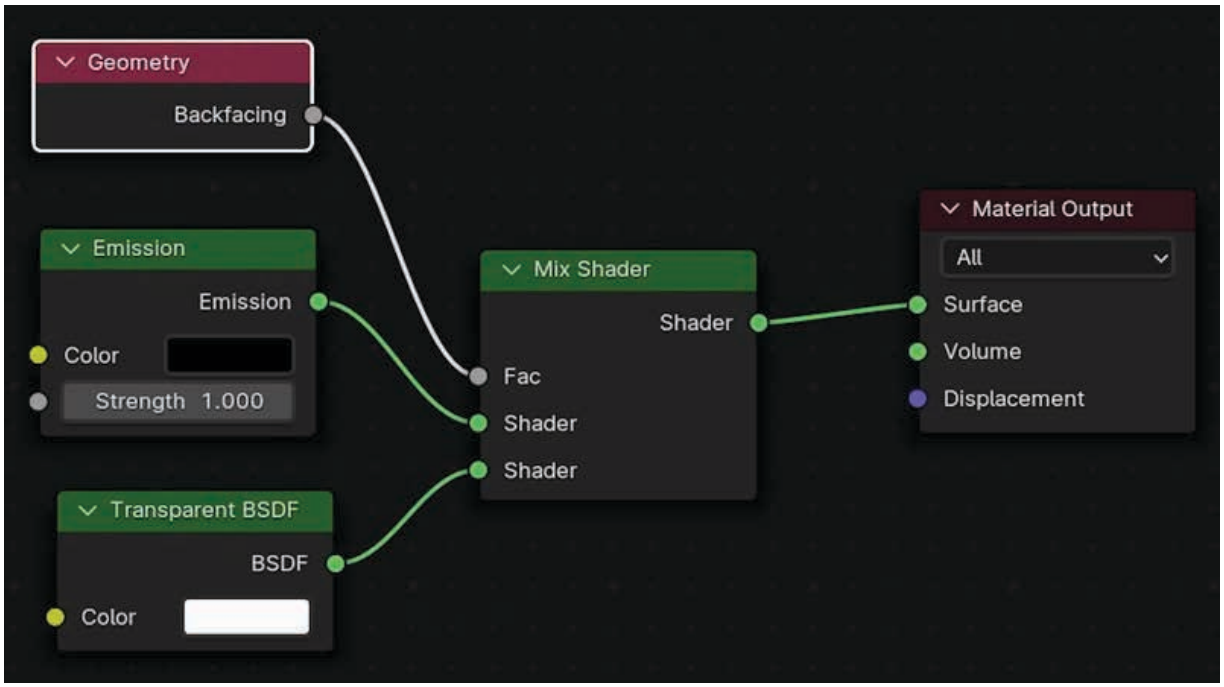
You can hold Alt and Right Mouse Button-drag from the output of one node to roughly where the other node is, to make a connection.



Use the **Backfacing** output of a **Geometry** node to mix them. This shows the Emission material where faces are flipped, and it shows transparency everywhere else.

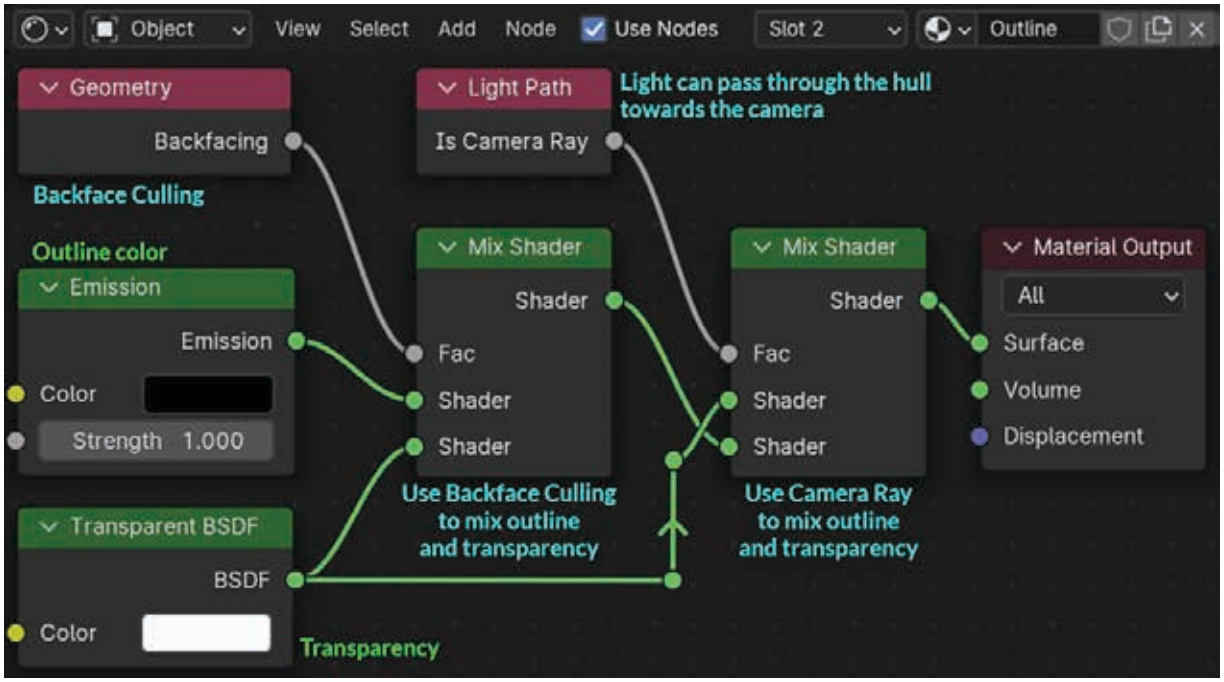
To make a node with a lot of unused outputs (like the Geometry node in this case) smaller, select it and **press Ctrl+H**.

You can expand it again by pressing Ctrl+H again, if needed. But this helps keep everything more neat and easy to look at.



Then mix that with the **Transparent BSDF** again, but use a **Light Path - Is Camera Ray** output to mix it.

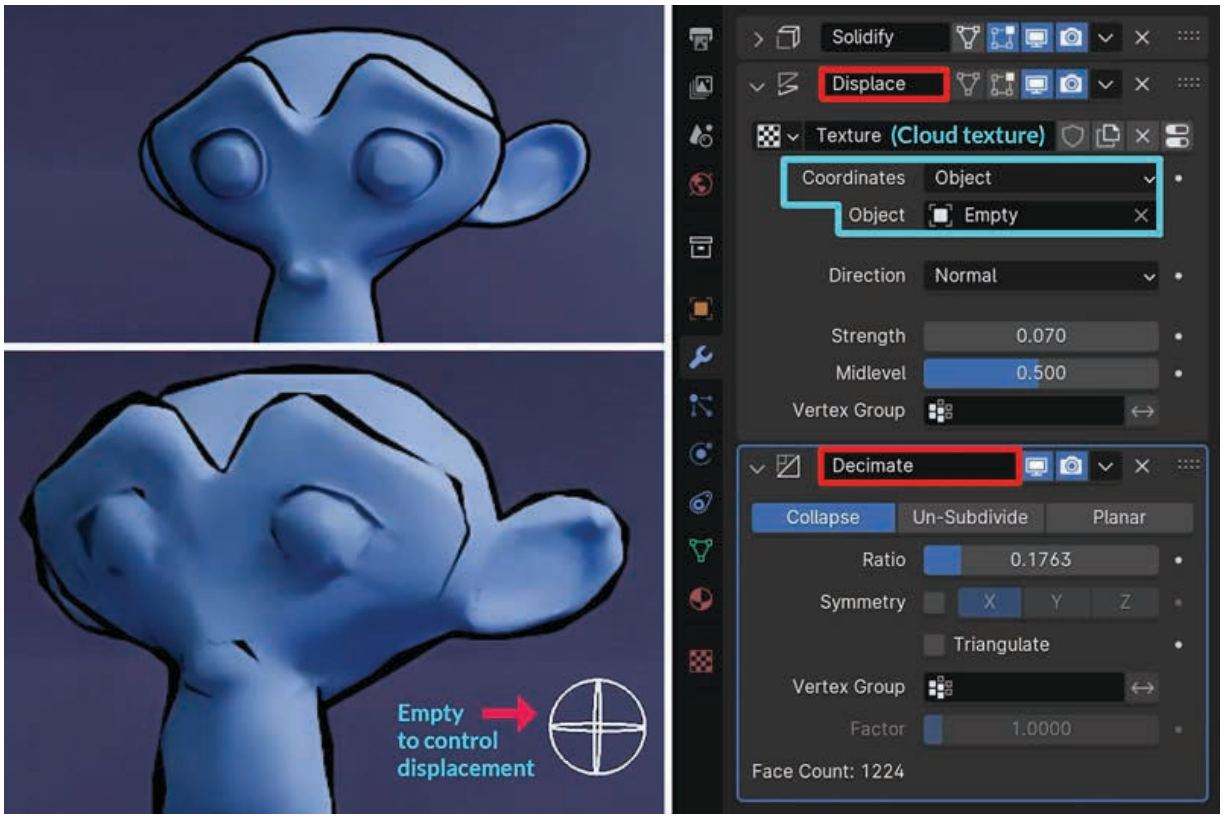
This way, the light can pass through the hull at the mesh inside, at least from the Camera view. In this picture I've annotated what each node does.



Now you have a toon outline that works in Cycles renders.

Finally, we can add some modifiers to make the outline look more interesting.

We can use a **Displace modifier** with a **Cloud texture,** and an **Empty** to drive its animation (move the Empty to animate the displacement) plus a **Decimate modifier** to make the lines more simplified.

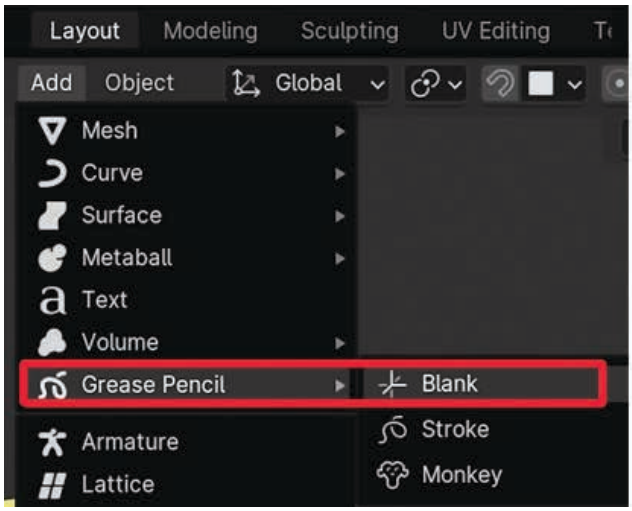


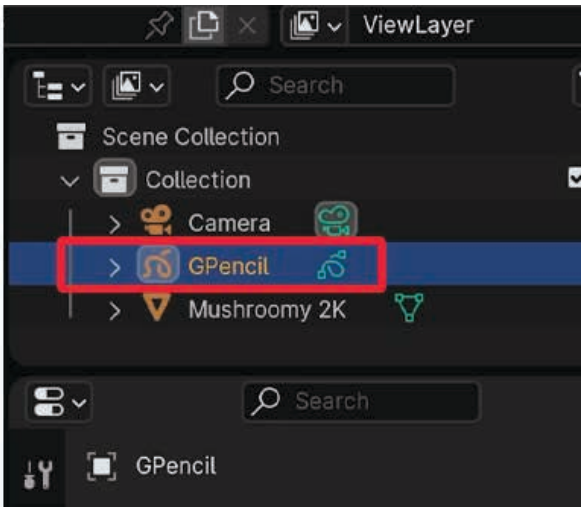
Create an outline with the Line Art Modifier

The Line Art modifier adds an outline to objects. It offers a different method of creating an outline, which has the benefit of having several modifiers that control its look. For example, you can add noise, random line thickness and a texture to the outline.

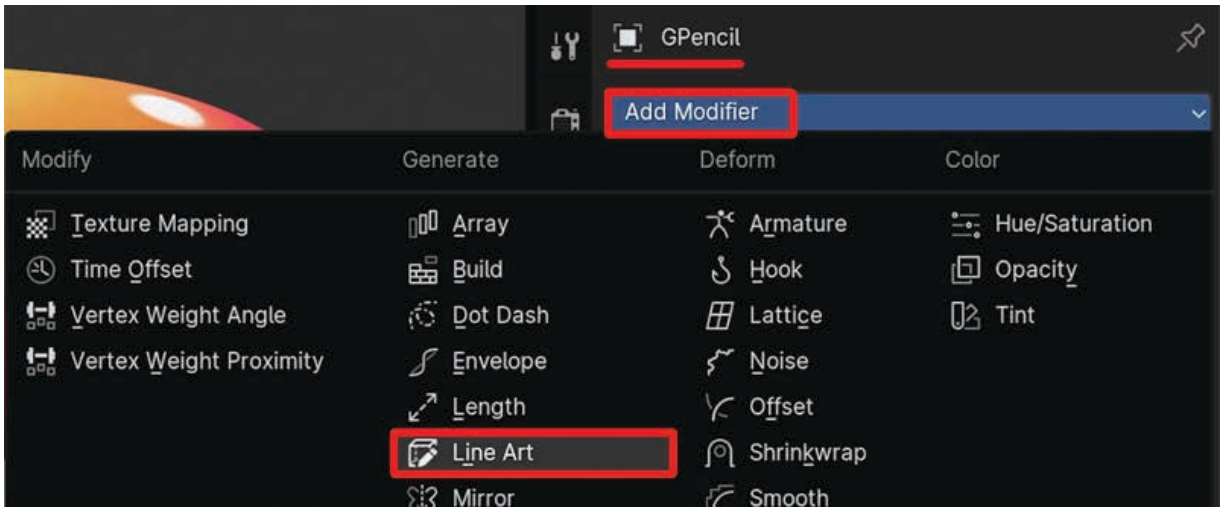
The downside is that it can be quite slow, especially when there is a lot of geometry.

To use it, first add a **Blank Grease Pencil** object to the scene. You can find it from the **Add** menu, or by pressing **Shift+A**.

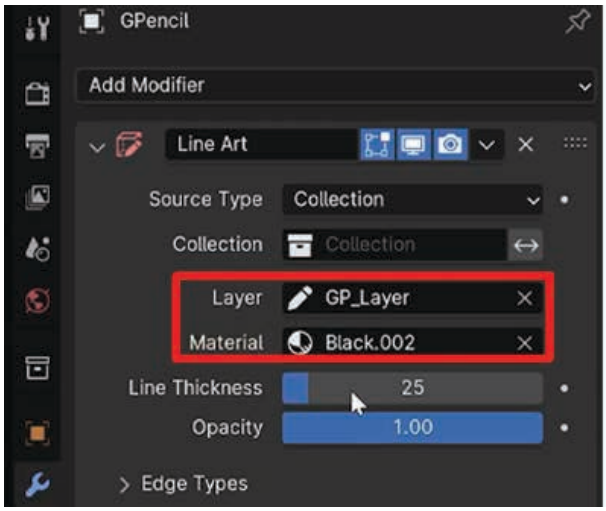


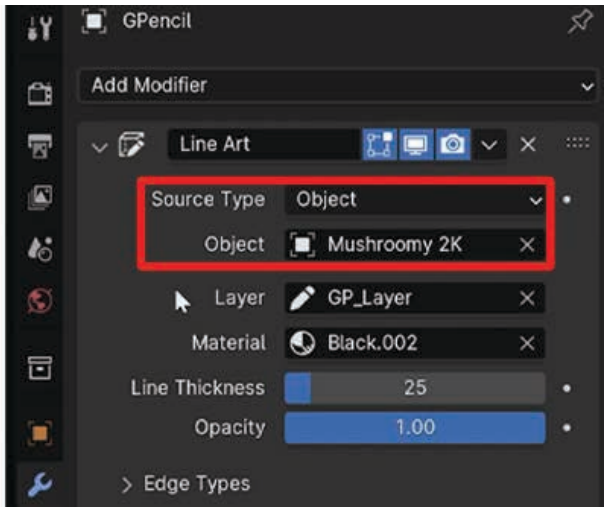


Add the **Line Art modifier** to that Grease Pencil object.

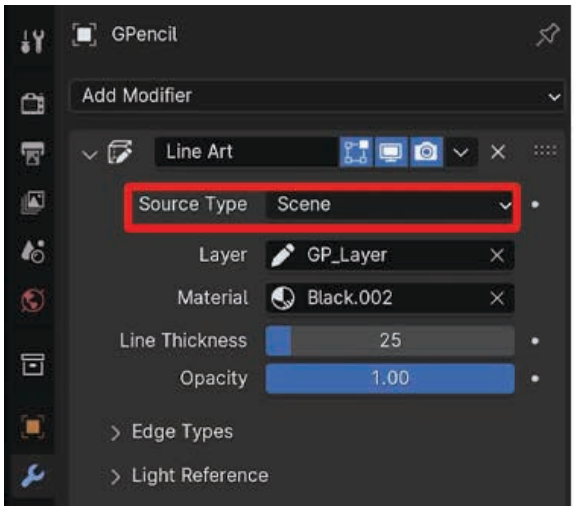


Choose the Grease Pencil **Layer** and **Material**. As the **Source Type**, either choose an individual object to give the outline to…

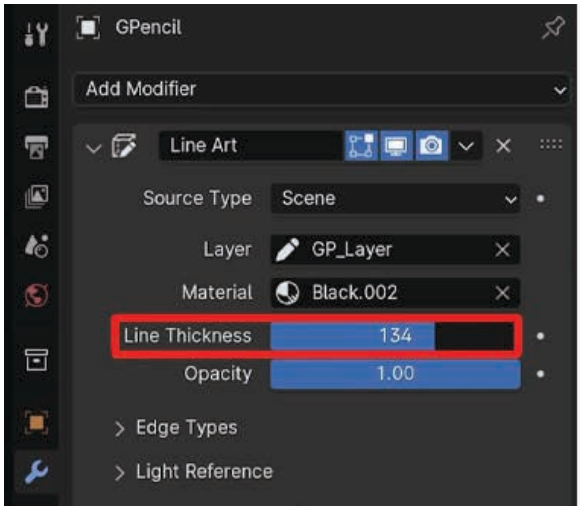




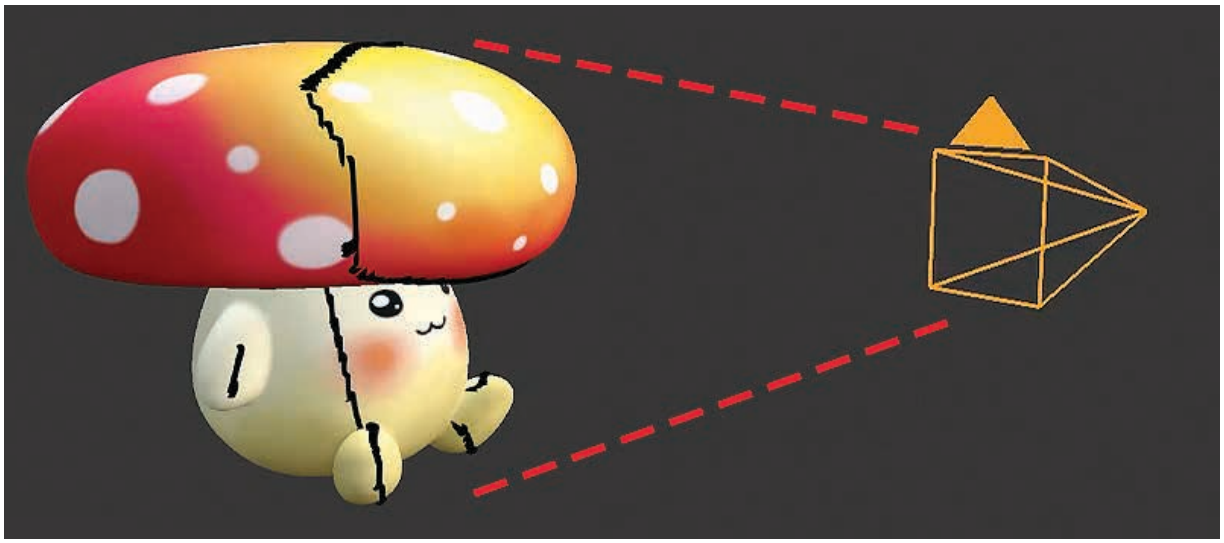
… or just apply it to everything in the Scene. You should be able to see the outline around your object now.



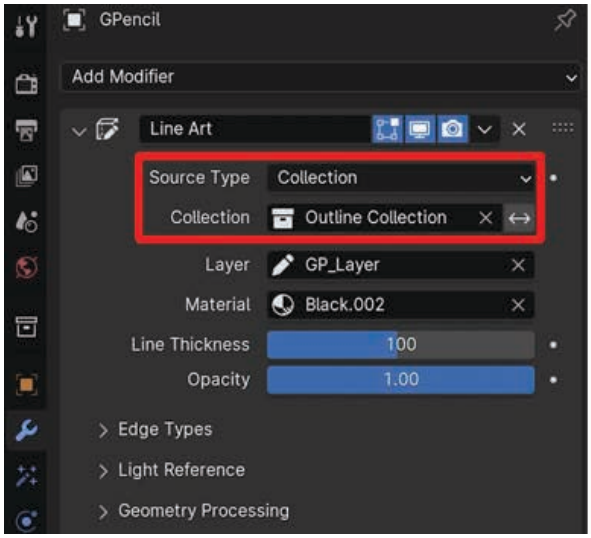
Then you can increase the **Line Thickness** if needed.

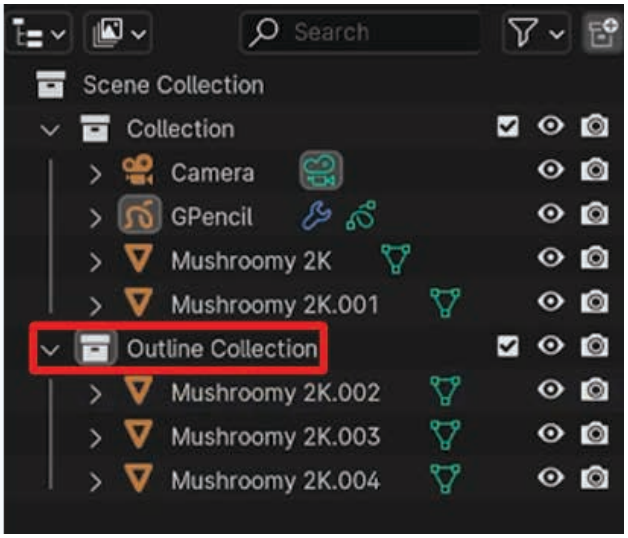


The effect works from the Camera view, so it only looks good from that angle.



To only apply the Line Art to objects in a specific Collection, set it to Source Type **Collection**.



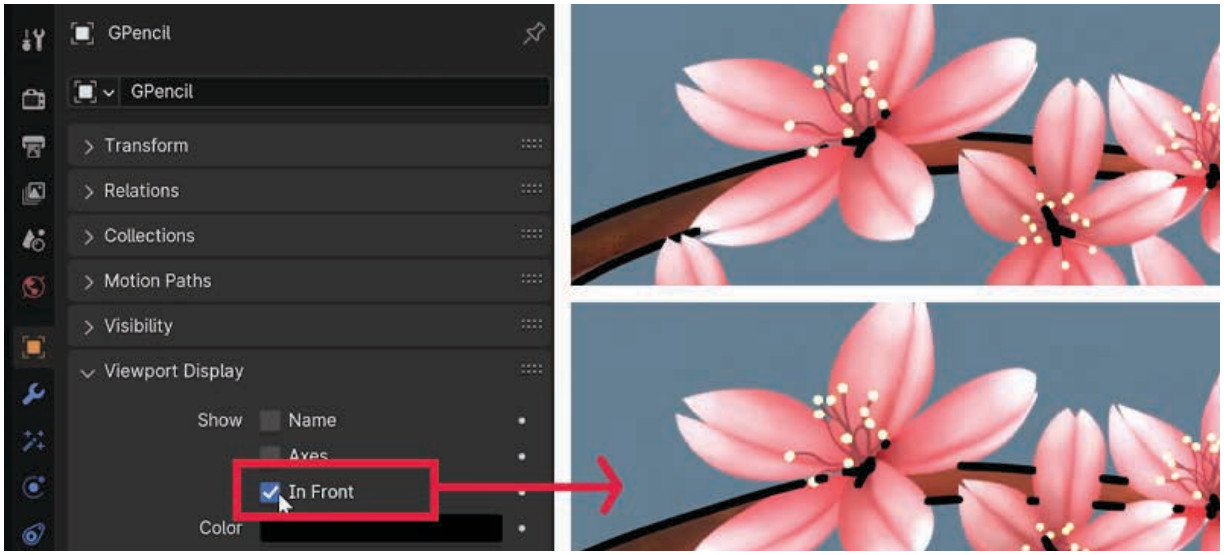


I have found that sometimes, you may have to exit and re-enter Render preview in the 3D viewport to see the changes made in the Source Type.

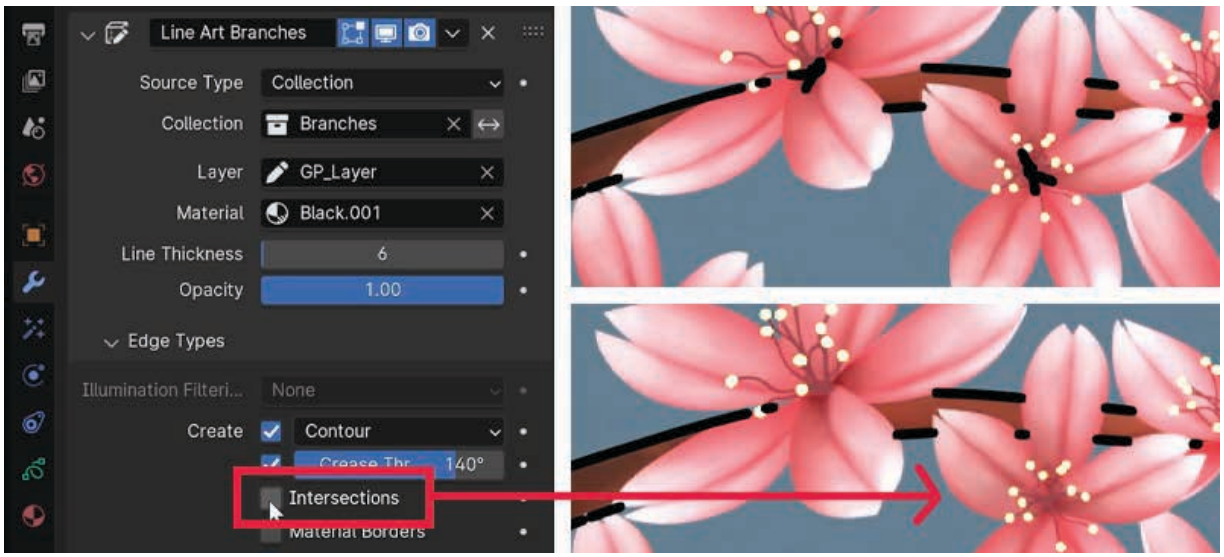
Add noise to Line Art Modifier strokes

Previously we looked at adding Grease Pencil strokes to objects using the Line Art modifier. To improve how it looks, here's a few tips.

Enabling **In Front** can help to make the strokes look more consistent. Can you spot the difference? In some cases it's more useful than others, so decide based on your model whether you want to enable this or not.



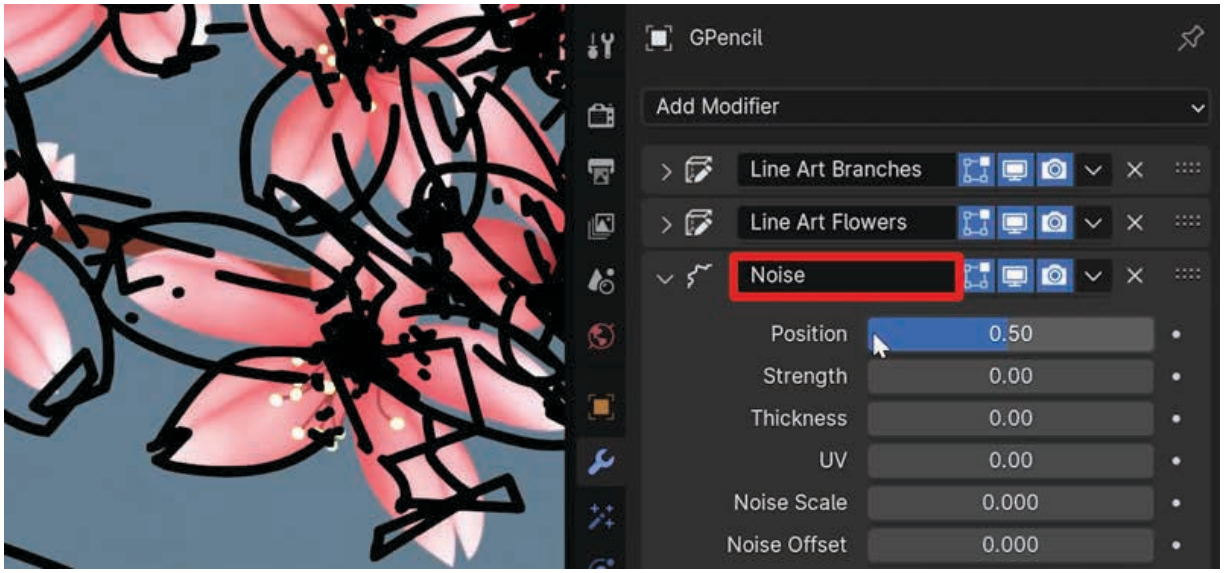
You may also want to disable **Intersections**. Again, the difference can be subtle.



Click here to return to the index.

Add a **Noise modifier** if you want the lines to be less perfect.

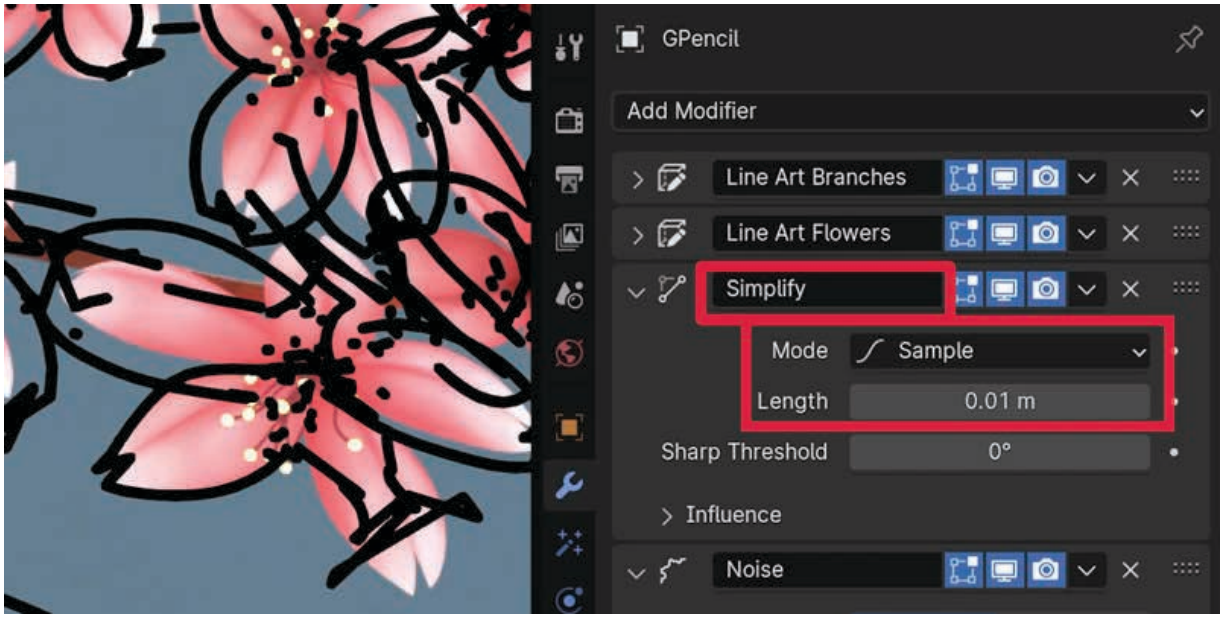
With the default settings, it will look very chaotic, but don't worry - we can fine-tune it.



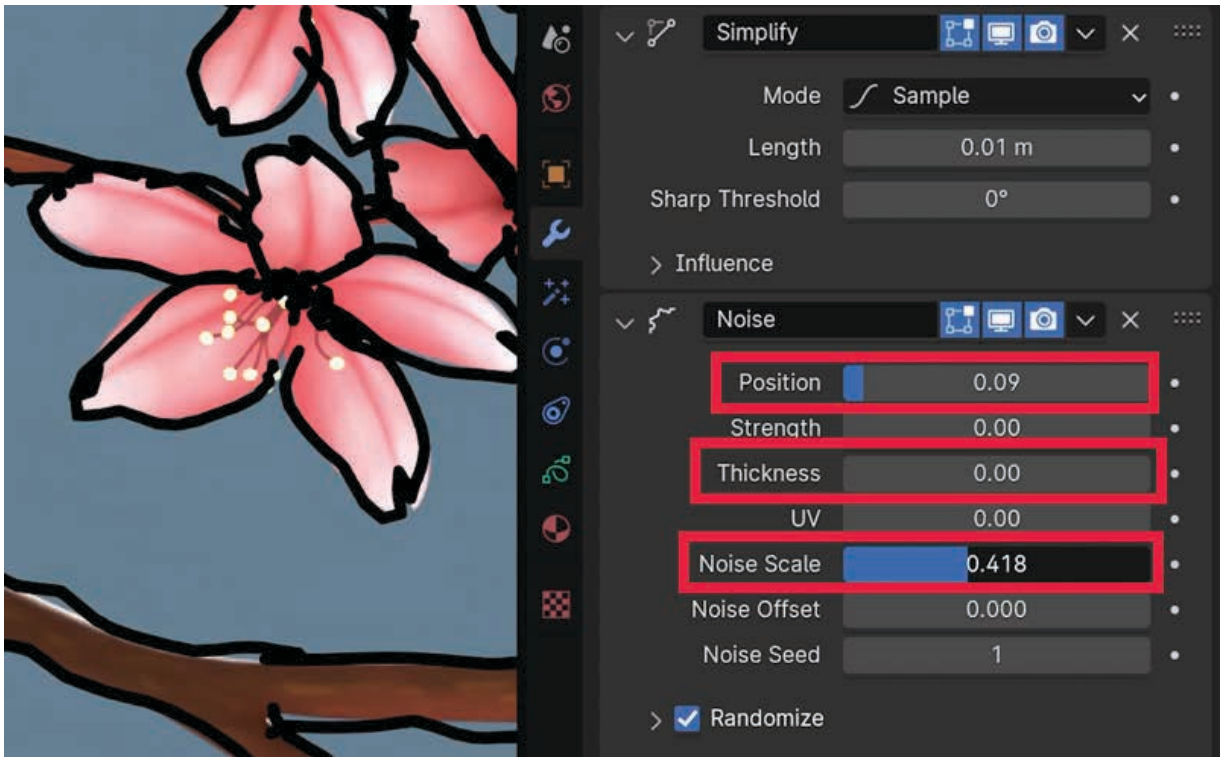
This works best with a **Simplify modifier** before it, to give the Noise modifier more to work with.

Set the Mode to **Sample** and reduce the **Length to 0.01 m**.

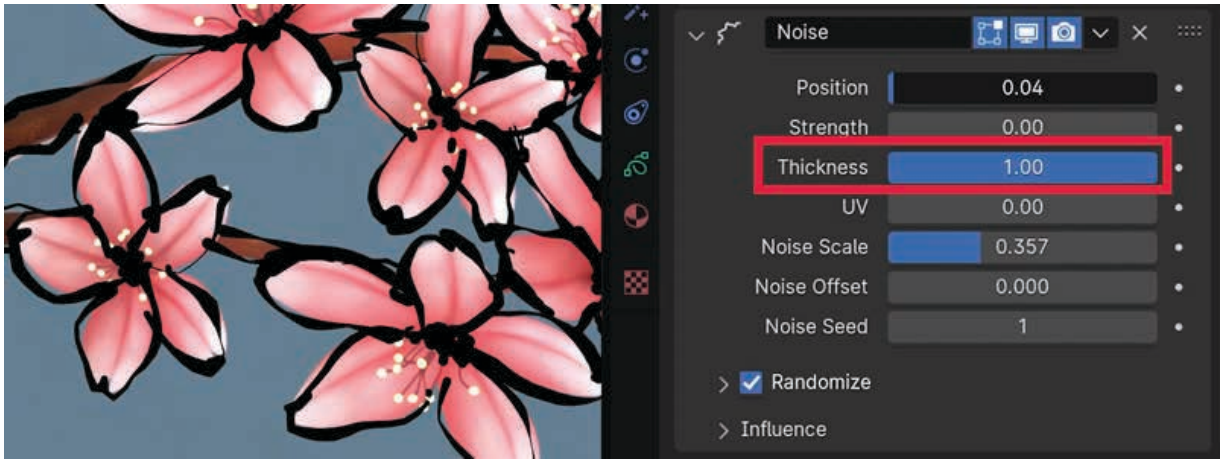
This gives us an even distribution of many points on the line, that we can use to fine-tune our noise.



Position and **Noise Scale** influence the intensity and the scale of the noise.



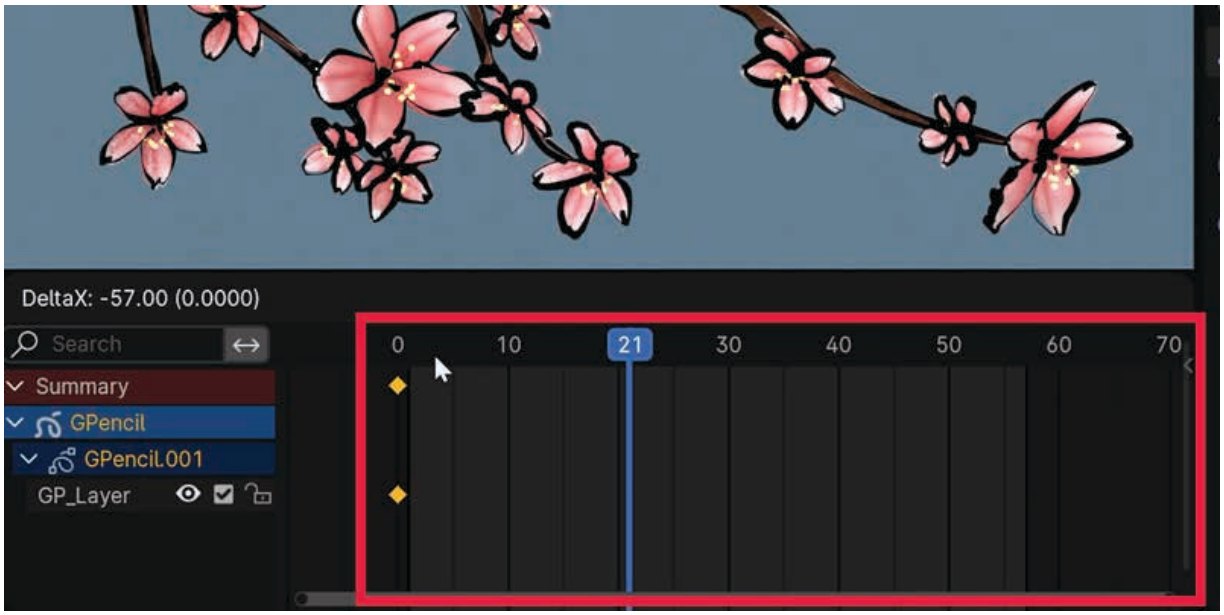
Increasing the **Thickness** value gives you a varied line thickness.



Make sure that the Grease Pencil keyframe is on the first frame of your animation.

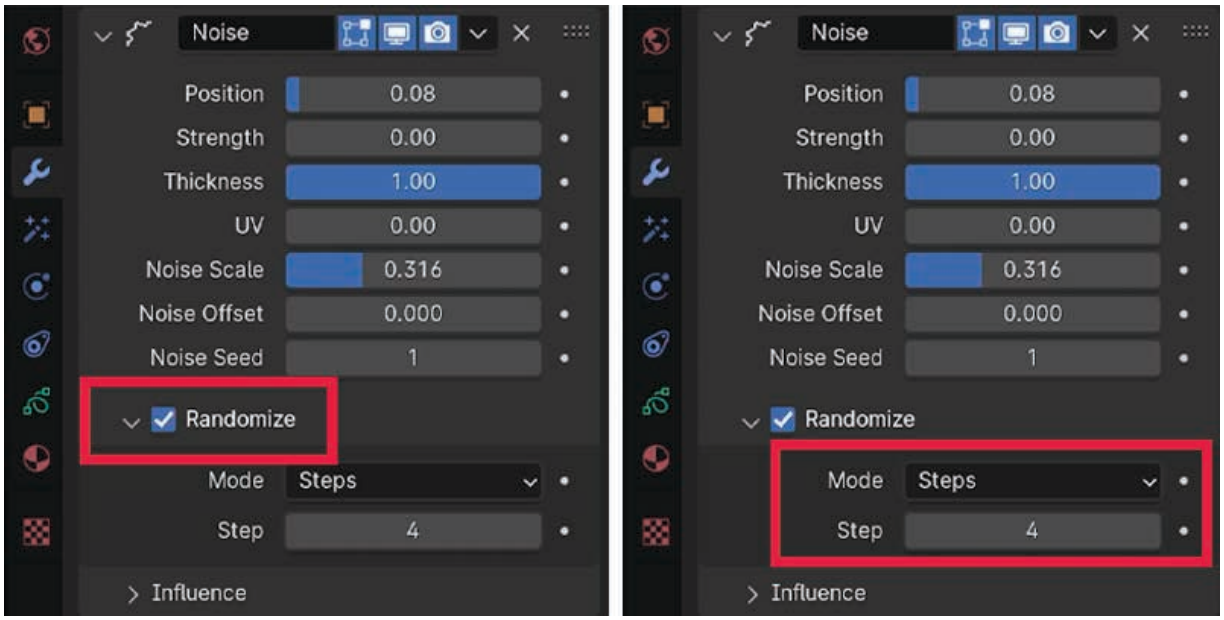
That way, the strokes will show up from the beginning.

You can move a keyframe by clicking on it to select it, and **pressing G** to move.



The noise is randomized every few frames, unless you uncheck **Randomize**.

Per how many frames it's randomized is influenced by the **Steps** value.



Baking the Line Art Modifier

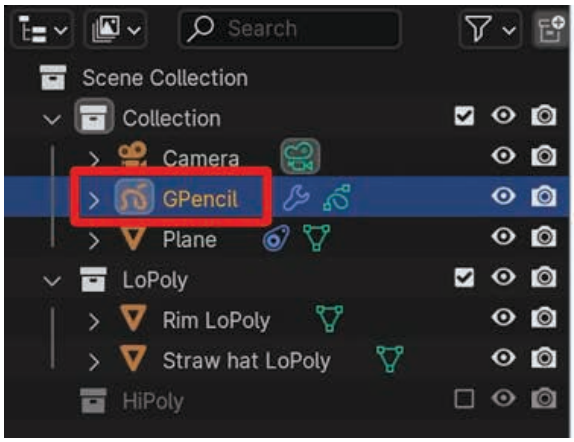
The **Line Art modifier** is great, if you want to render stylized graphics with outlines. But it can be quite resource-demanding in my experience.

Because it takes some effort for your computer to calculate the effect, it makes sense to "bake" it. That way, everything moves much more smoothly.

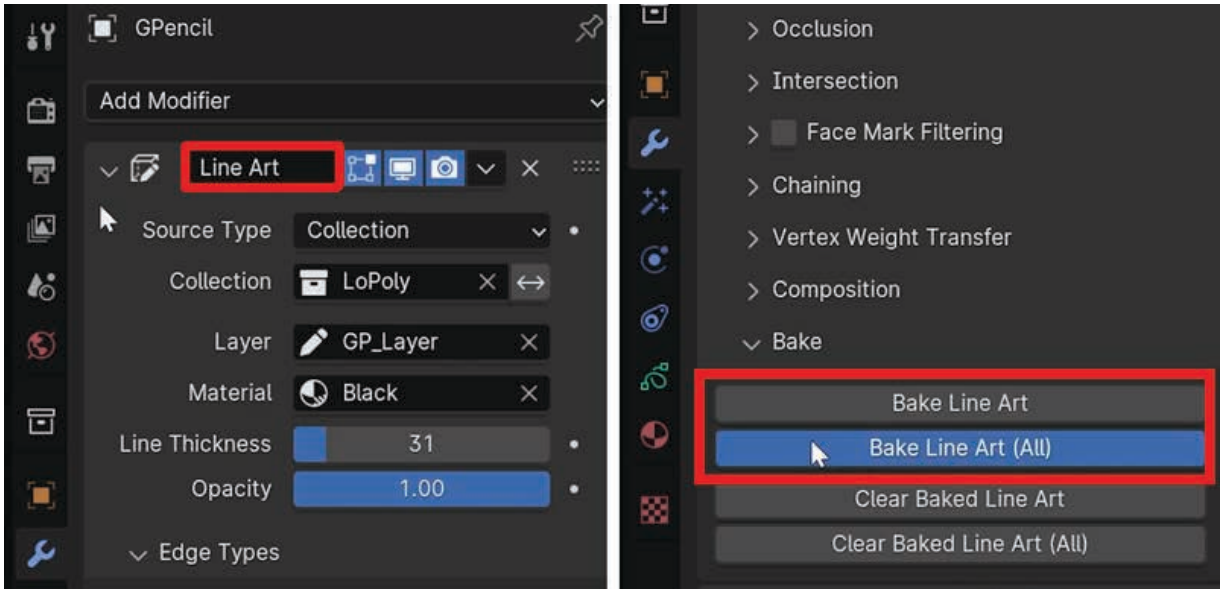
Keep in mind that this is no longer affected by the camera position, so it only looks good from the Camera perspective that you baked from.

It can be useful if you plan to tweak the outline strokes manually, erase parts, or add strokes manually using the normal Grease Pencil tools. I think for 2D animation this can be a very powerful workflow.

To do this, select the **Grease Pencil object** and go to the modifier tab.

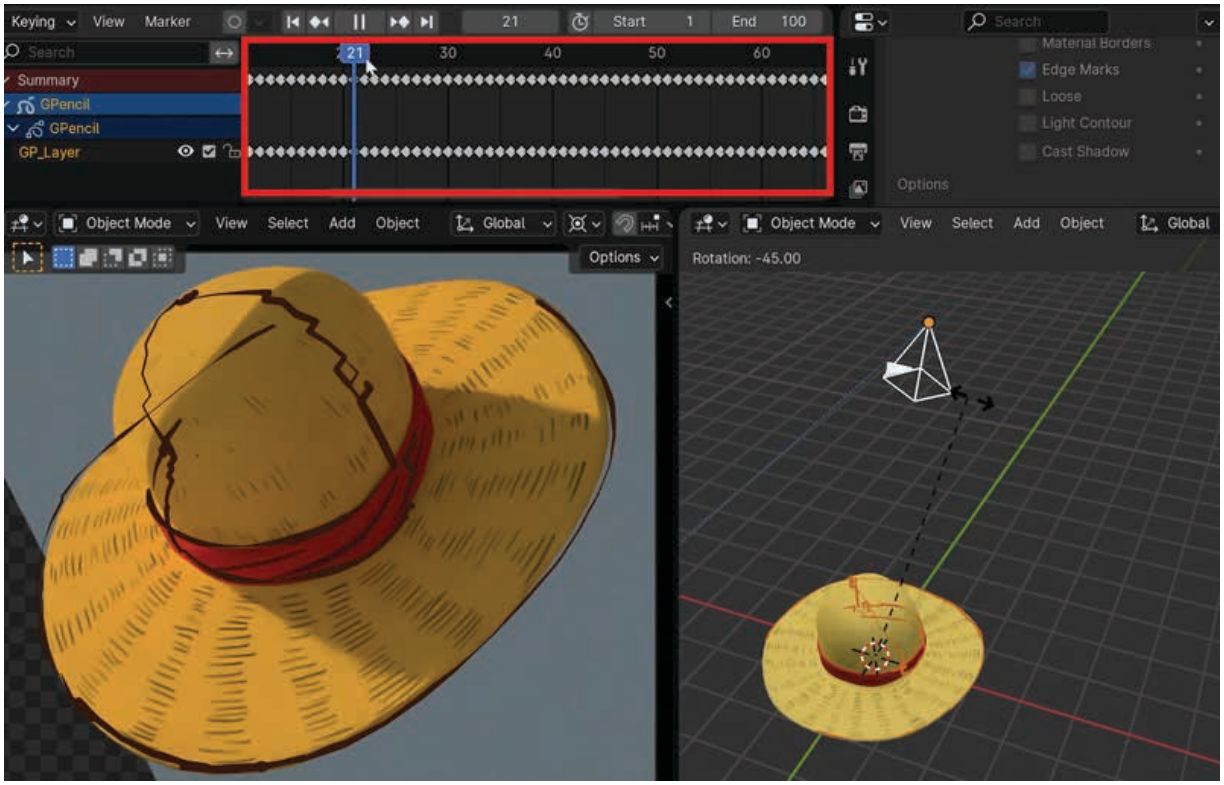


In the **Line Art modifier** expand the **Bake** tab. Click on **Bake Line art**.



Now the outline is baked to **keyframes**.

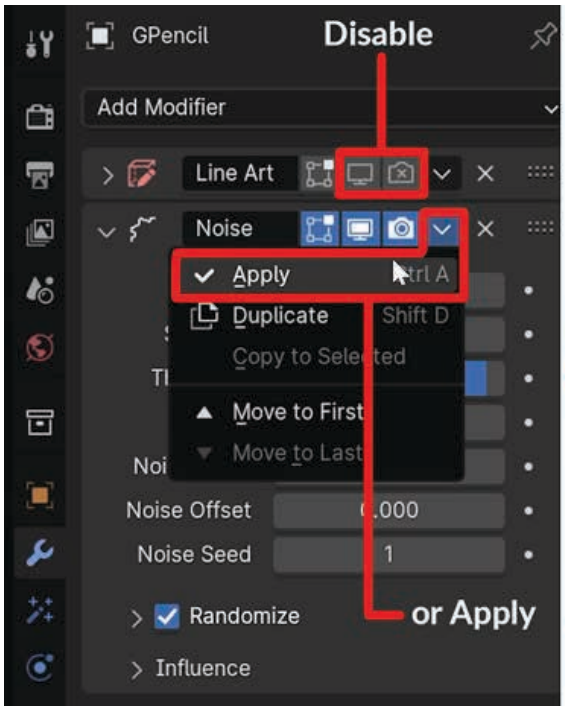
You can see that the Camera position no longer influences it.

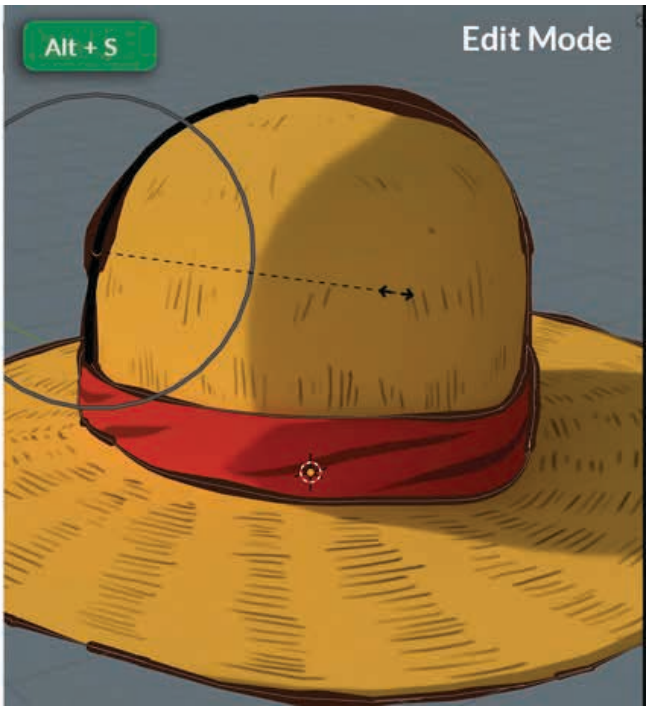


Since you no longer need it, you can disable or remove the modifier.

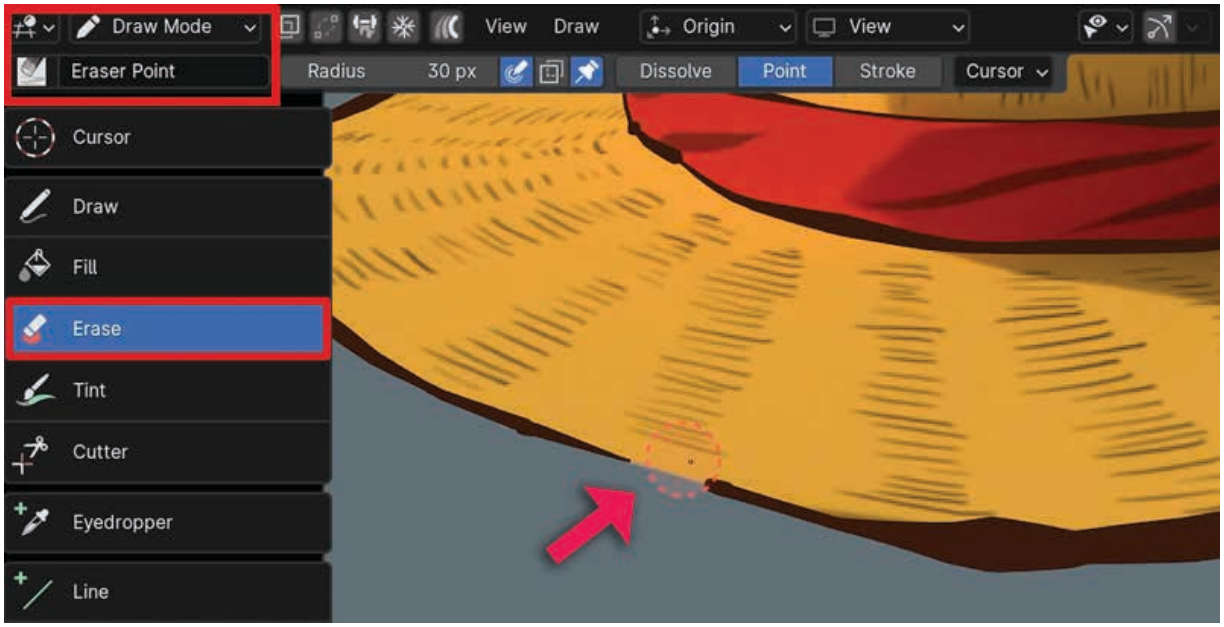
Now you can treat the outline like any other Grease Pencil object.

So you can edit it in **Edit Mode**, for example with the shortcut **Alt+S** to increase the thickness.

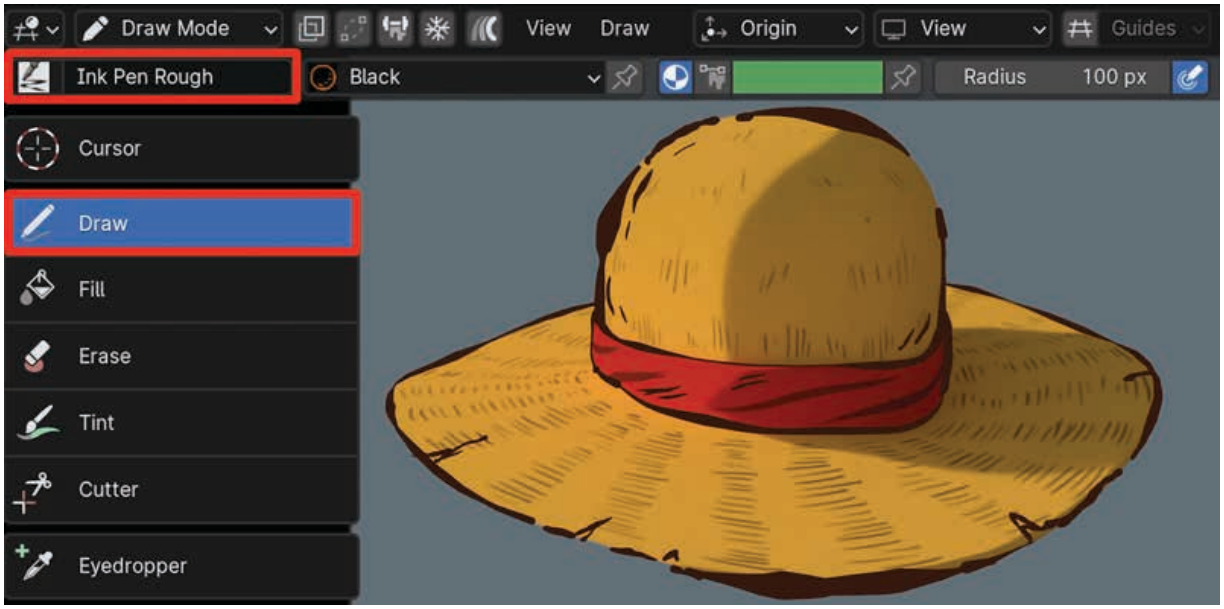




Or in **Draw Mode** you can **Erase** the stroke…



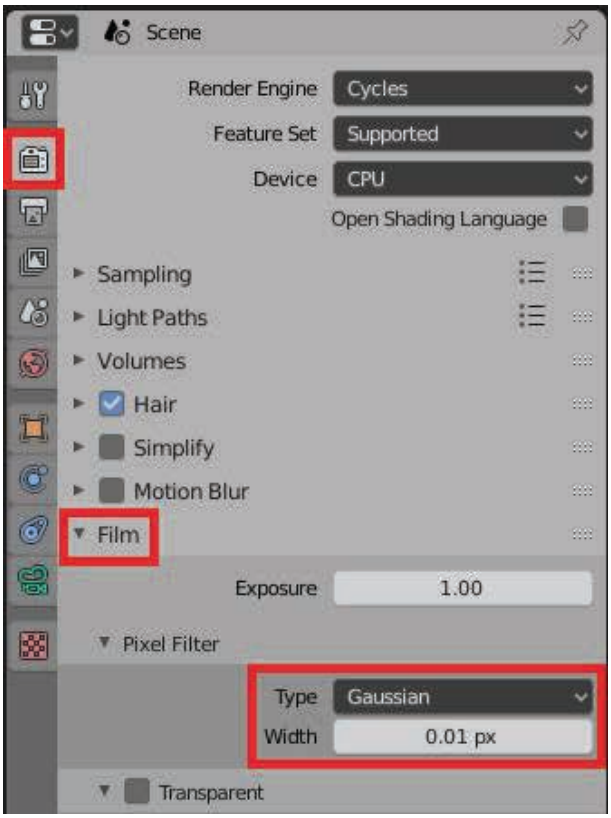
...or add to it with various brushes.



Pixelated Renders

To get a videogame-like look to your renders, you need to pixelate them. There are a couple of ways to do this. For Cycles, read further on this page, and for Eevee go to the next page.

Isometric scene by Laurynas Jurgila on Blendswap.

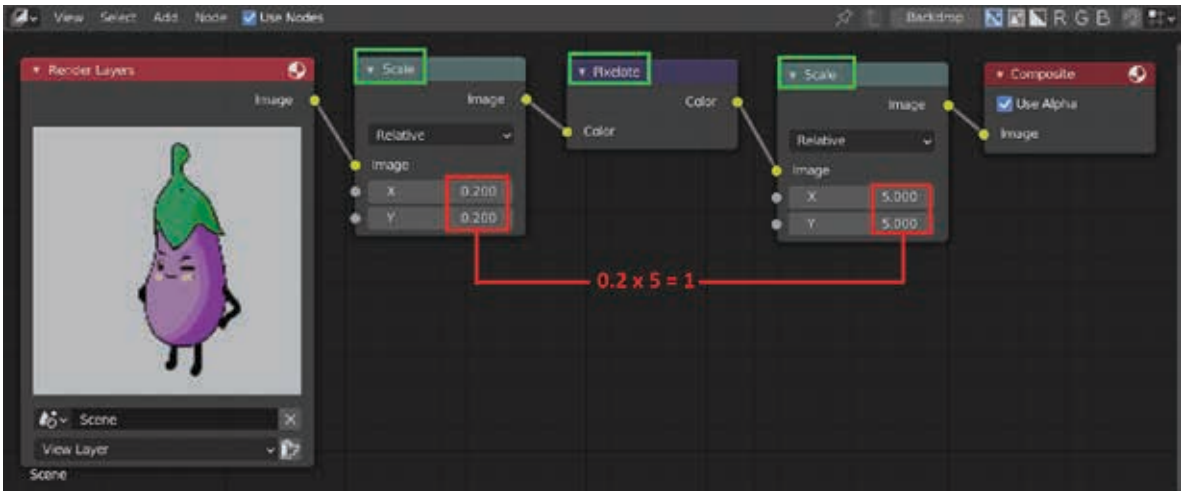


In the **Render Properties** tab, set the Resolution to 10%.

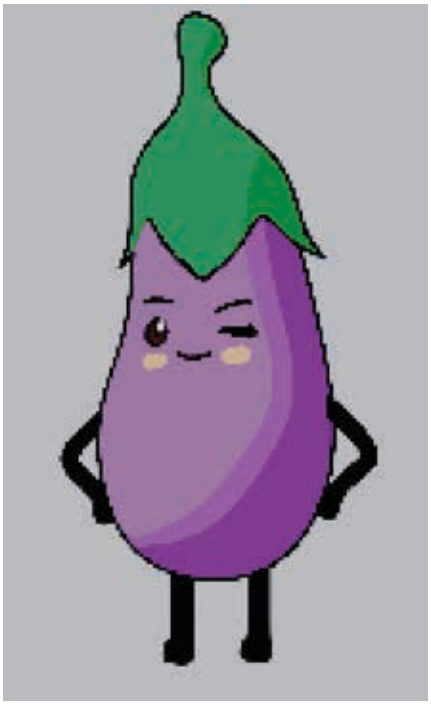
In the **Output Properties** tab under "Film", set **Gaussian Width** to **0.01** instead of the standard 1.5.

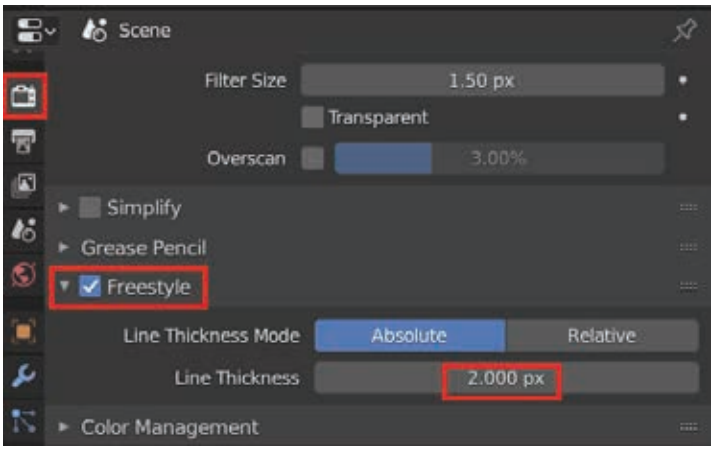
To achieve a pixelated render with Eeveeyou can use the Pixelate node in the Compositor. Leave the renders settings to their defaults settings. The only difference in this example is that I'm using a square image of 1080 by 1080 pixels.

Open a Compositor window and add a Pixelate node sandwiched in between two Scalenodes. Now set the first scale node to a lower value and the second one to a higher value. The lower and higher value should multiply to have 1 as their result.



To get an outline turn on Freestyle. In this case I went with a Line Thickness of 2. Now **press F12** and after a while (after the initial normal render appears) you'll see your pixelated render with freestyle line.





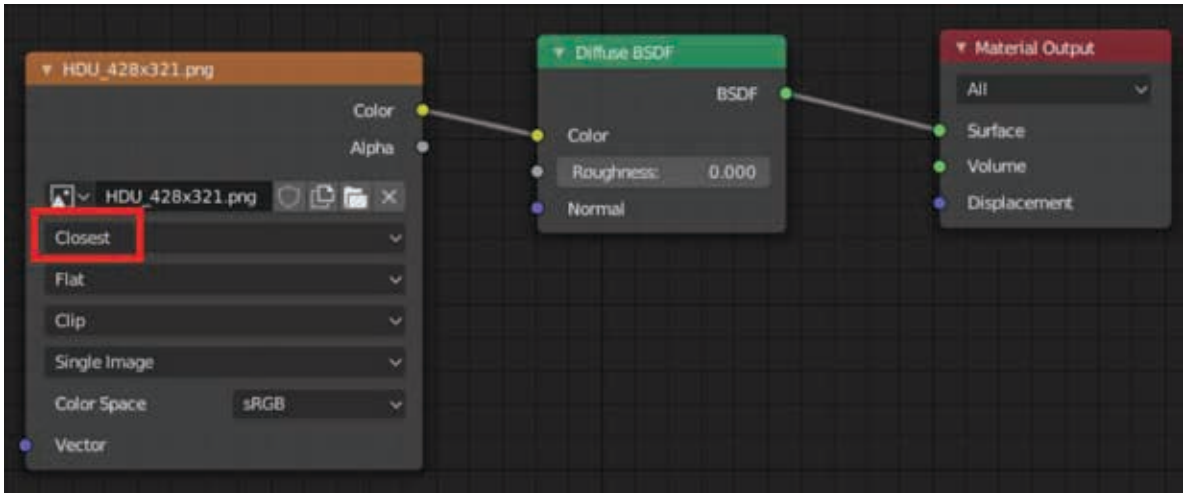
Correctly display pixel art textures

By default Blender blurs textures a little bit. For pixel art, this may not be what you want at all.

The blurry “Linear” setting.

The crispy, pixely “Closest” setting.

In the Shader Editor, set the Image texture node Interpolation to Closest. Now the image will be displayed with all its blocky pixels intact.



Picture by Buch from OpenGameArt.org.
https://opengameart.org/content/a-blocky-dungeon

Freestyle calligraphy render

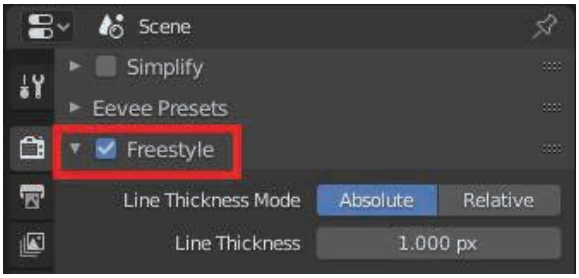
This is a project I worked on before, which has a nice render style made by using the "Freestyle" settings in Blender cycles. Of course there are newfangled ways of doing this in Blender 2.8 and Eevee, but Freestyle still has its charm! It can really create unique looks.

The sheep model has a simple color texture and an Emitter material so that it's shadeless. In other words, it renders "flat".

The freestyle lines are added during render. I think it actually looks kind of neat this way as well, before the lines.

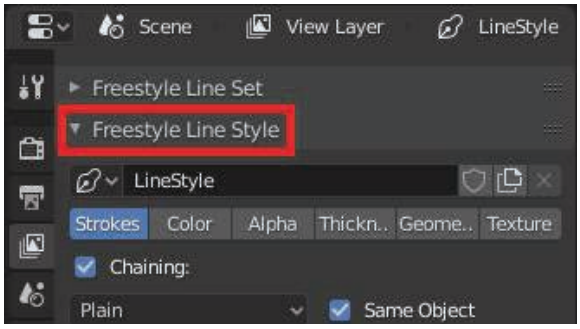
Freestyle needs to be turned on in the Render tab.

Note that if you are using Eevee, you'll still need to render before seeing the freestyle lines. That's because Freestyle is a post-render effect. You can use Cycles or Eevee.



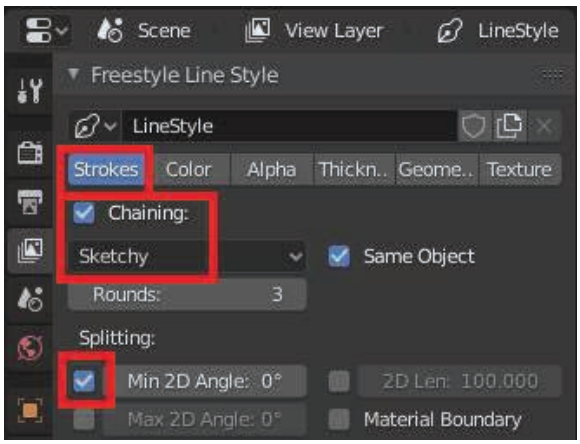
After turning Freestyle on, the settings can be found in the View Layer tab.

Go to **Freestyle Line Style**.



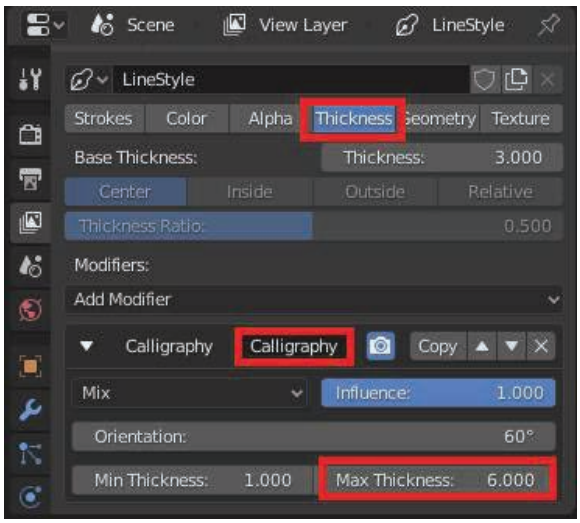
The Line Style has the following **Strokes** settings:

Chaining should be set to **Sketchy**. **Splitting** should be turned on for Min 2D Angle: 0°.



Under **Thickness** add a **Calligraphy** modifier. This creates a more organic looking line which tapers off at the ends.

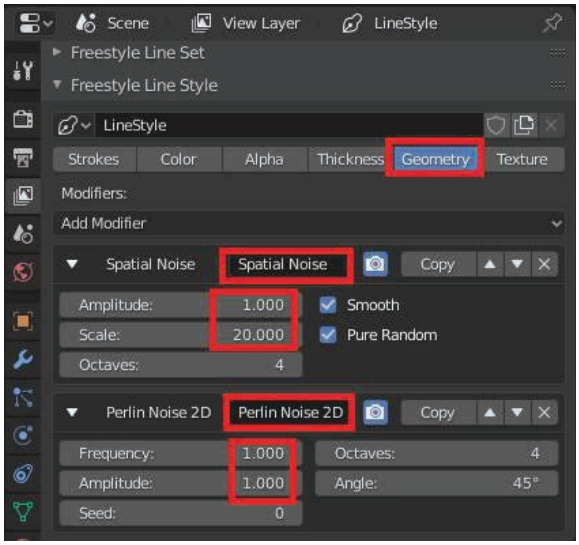
For this example I reduced the **Max Thickness** to 6. You can experiment with the thickness you like. This is how thick the lines will be at their thickest point.



Under **Geometry** I added a Spatial Noise modifier as well as a Perlin Noise 2D modifier, with the settings as pictured.

This is all a matter of trying out different modifiers and values and seeing what looks nice.

Press F12 to render and see the result, then tweak as required. It's faster to try these settings on something simple like a Monkey or even just a UV Sphere before using them in a bigger scene.



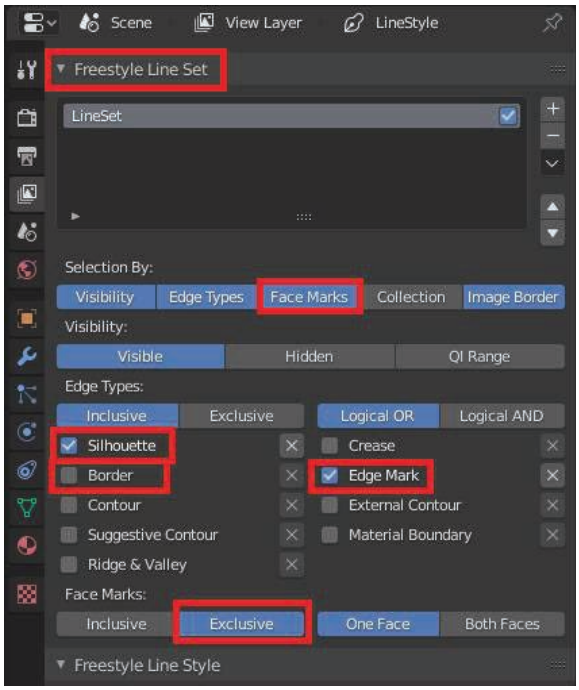
Now go to the **Freestyle Line Set** options.

Face Marks is turned on.

Combined with Face Marks: **Exclusive** this allows us to select faces and make them not render, causing an organic kind of interruption of the lines in some parts of the model.

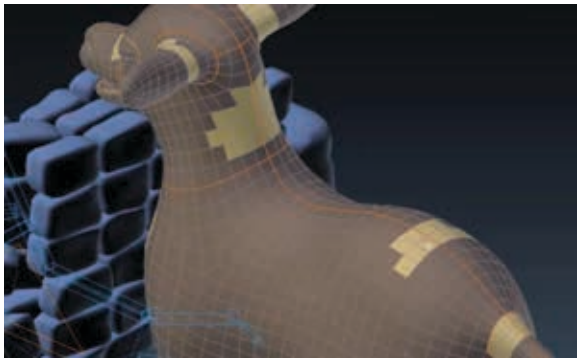
Edge Mark causes the stroke interruptions as well, making it look more hand-drawn.

Silhouette is turned on, **Border i**s turned off. This just tells Freestyle to render lines around the silhouette of the model, but not at borders of open mesh edges like the eye sockets of the monkey.



Excluding edge marks works by selecting edges on the model in Edit Mode, **right-clicking** and choosing **Mark Freestyle**.

Similarly, the yellow faces on the model are faces that are excluded from the Freestyle render.



The result, after rendering.

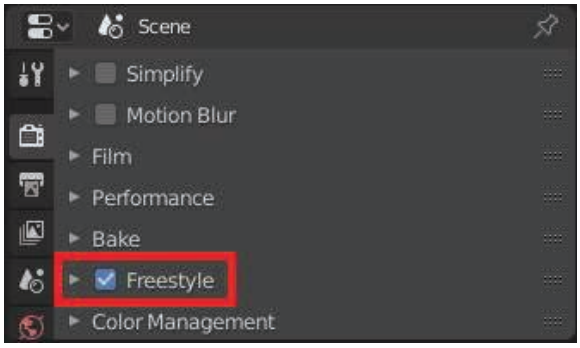
Pencil lines with freestyle

In this tutorial, we'll look at creating these kinds of sketchy, hand-drawn lines using Freestyle.

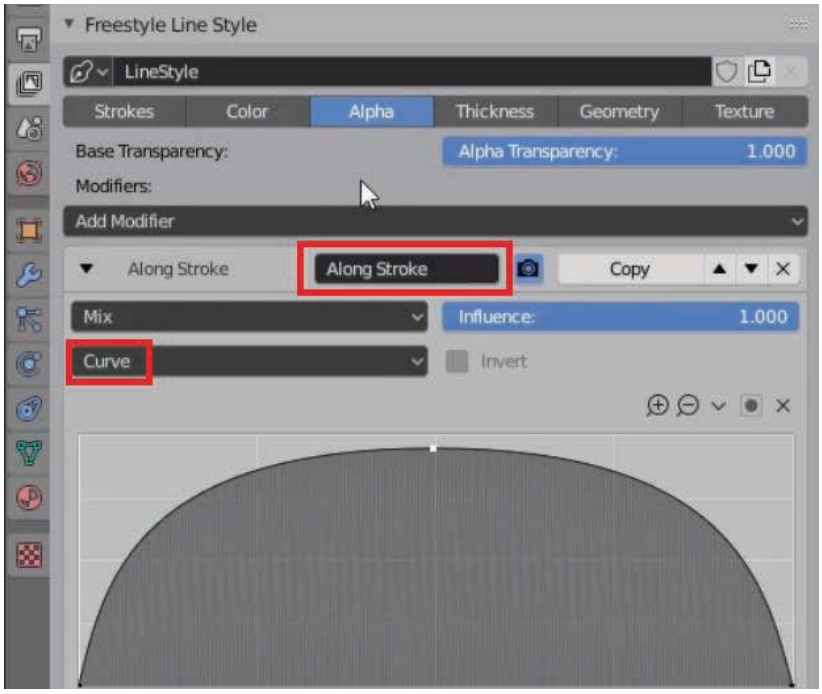
In this example I'm using Cycles, but Freestyle works in Eevee as well. Just know that you won't see the lines until after the rendering has finished.

Step 1: Turn on **Freestyle**.

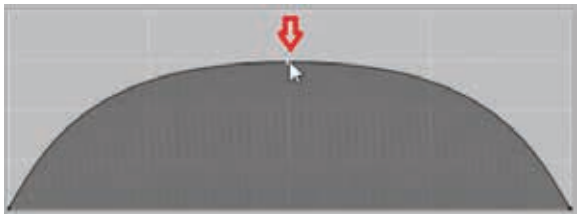
1 pixel **Line Thickness** is fine.



In the Line Style, add Alpha modifier: **Along Stroke**. Set to **Curve** and shape the Curve like shown below.



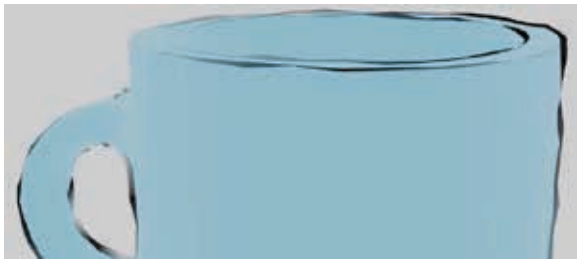
Drag it down to make the line more transparent.



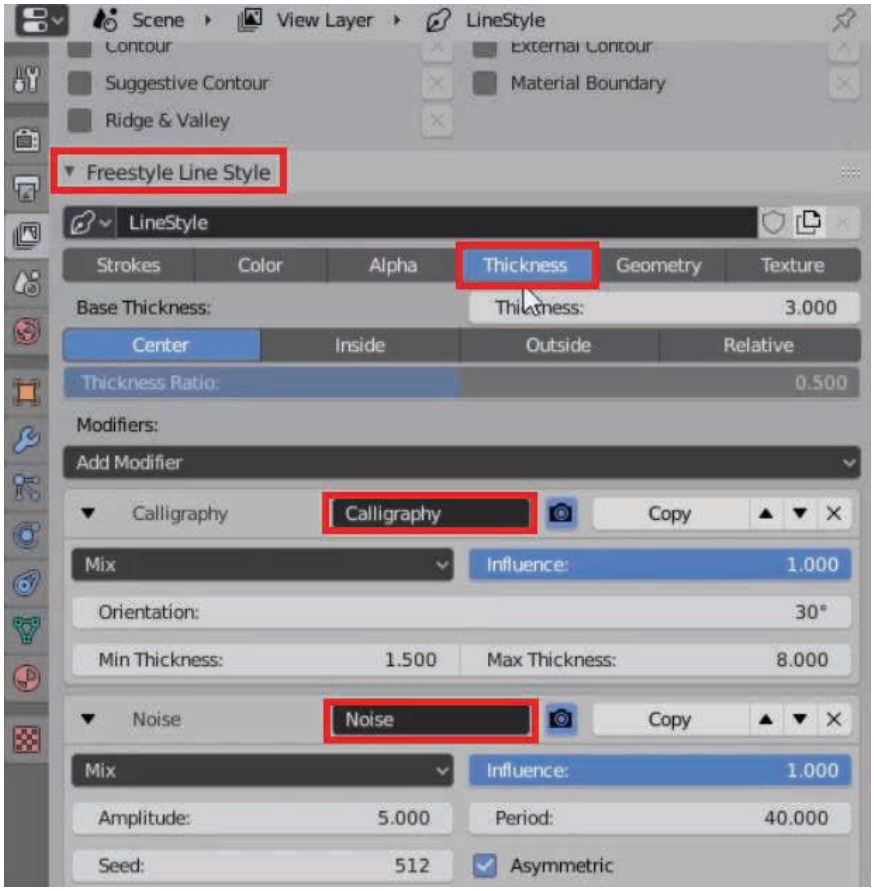
As you can see the result is more transparent lines, a more subtle effect.

Drag the curve back up, all the way to the top.

As you can see, the lines are no longer transparent.



The particular sketchy look of the lines is the result of adding these 2 modifiers to the **Thickness setting** of the **Line Style**, with the following settings.



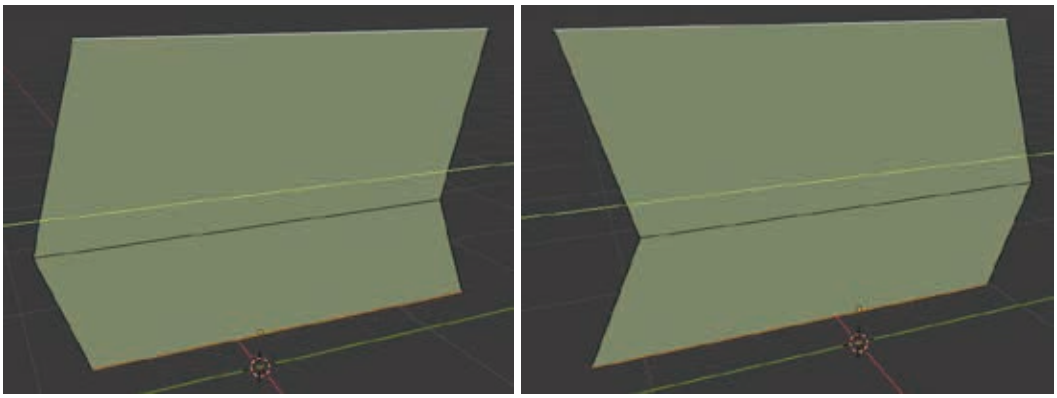
Calligraphy gives it the tapered front and end look, **Noise** introduces some randomness to the lines.

Easily Add Camera Movement To A 2D Painting

It's easy to add movement to a picture with depth like this picture from Shutterstock.

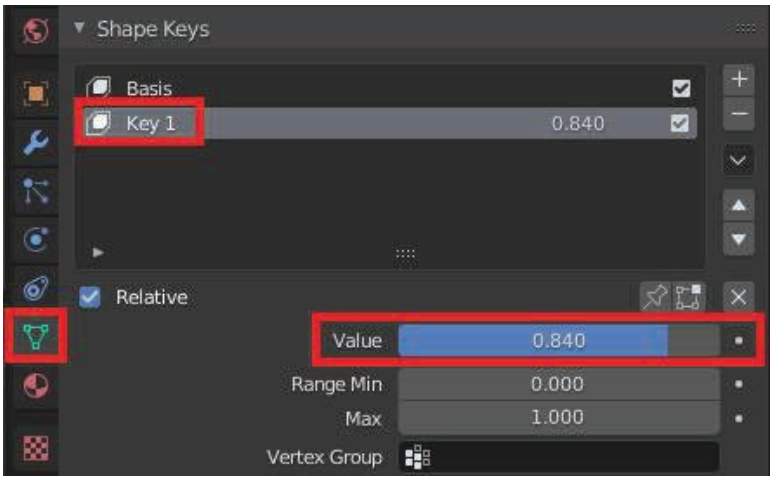
The background and foreground have been separated in an image editing program. The background consists of a plane with an edge at the horizon. The car is on another plane just in front of it.

The background is animated with a Shape Key. The horizon stays the same, but the top and bottom edge are moved from right to left.

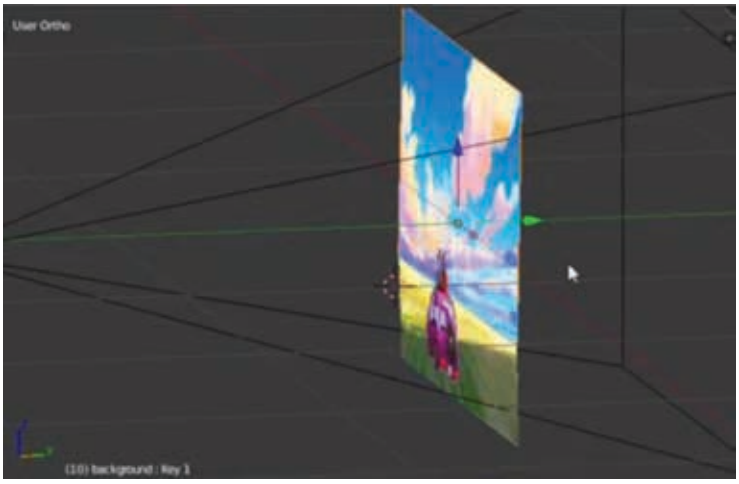


To keyframe the movement, adjust the Value slider of the Shape Key and **press i** while hovering the mouse cursor above the Value slider to set the keyframe. Move to a different time in the Timeline and change the Value, then keyframe it again.

Alternatively you can press the dot next to the Value slider to set a keyframe.



From the side you can see these are just flat planes. The car needs to only be slightly in front of the background plane.

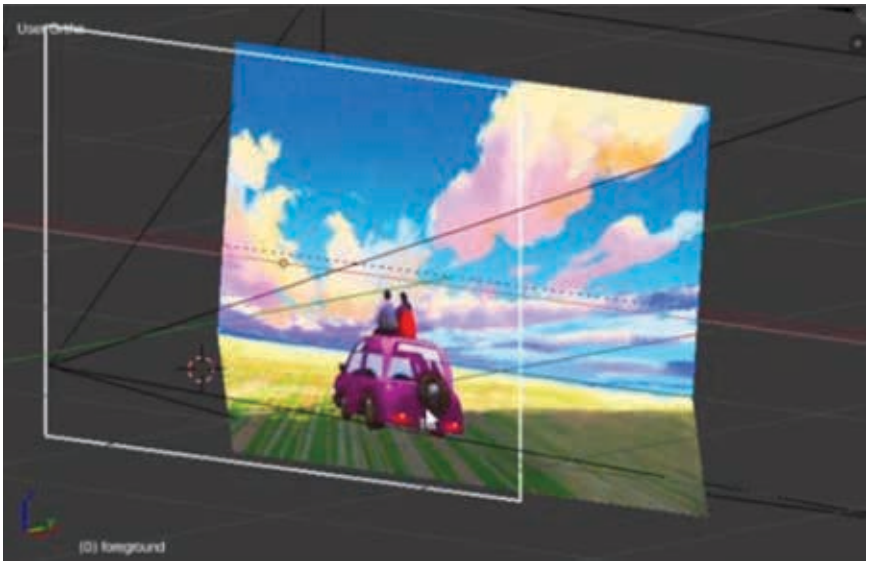


I used the **Import Images As Planes addon** to import the images and automatically place them on planes of the correct sizes. It comes with Blender by default and needs to be activated in User Preferences.

Here, the top part (the sky and clouds) are skewed with a Shape Key and keyframed from value 0 to 1 over time.

The same thing is done with the lower part (the ground).

The car in the foreground is a .PNG image file with an alpha channel for the transparency. It moves slightly from right to left, along with the movement of the ground plane.



From the camera view it looks like the background has perspective and depth.

Cut-out Animation

Cut-out animation is a quick way to make animation using drawings or images that have been prepared for this method of animation by cutting them into smaller pieces. They also require to have a transparent background.

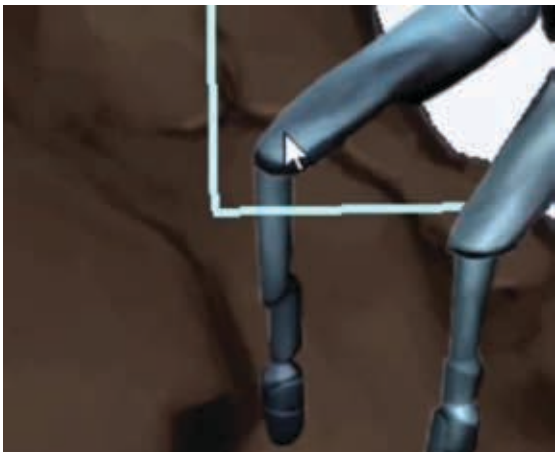
The benefit is that you can use special looking art and that it's a very quick way to produce animation as it's so simple. Some examples of cut-out animation: South Park, Terry Gilliam's animated sequences in Monty Python's Flying Circus, the opening sequence of the movie "Weird Science"...

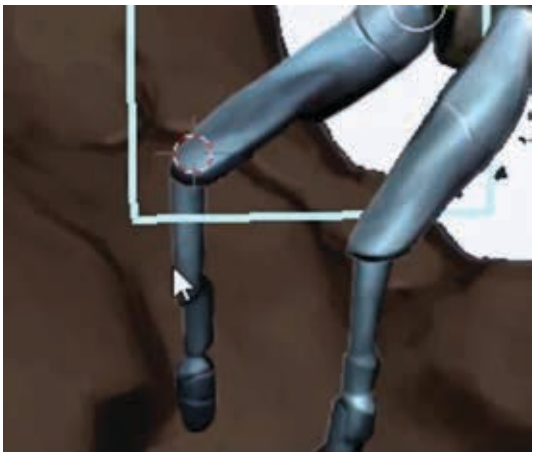
Cut-out animation is made by cutting an image into the separate parts (body, upper arm, lower arm, hand, maybe even all the finger parts, and so on). Typically these separate parts will be saved as PNG images with an alpha channel.

These can easily be imported into Blender by activating the **Import Images As Planes** add-on. It comes with Blender by default and only needs to be activated in the User Preferences.

So let's start making our character ready for cut-out animation!

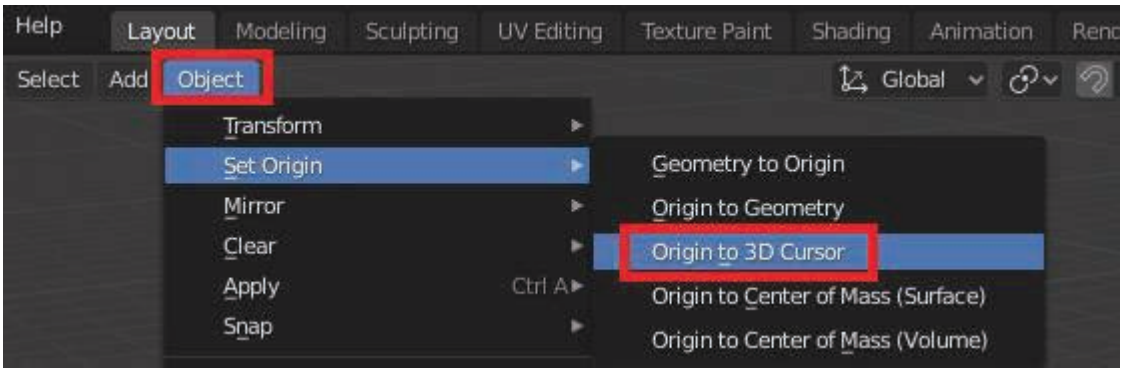
Place the 3D cursor at the point where you want the Pivot Point of the cut-out part to be by **holding Shift** and clicking with the **Right Mouse Button** at the desired location.



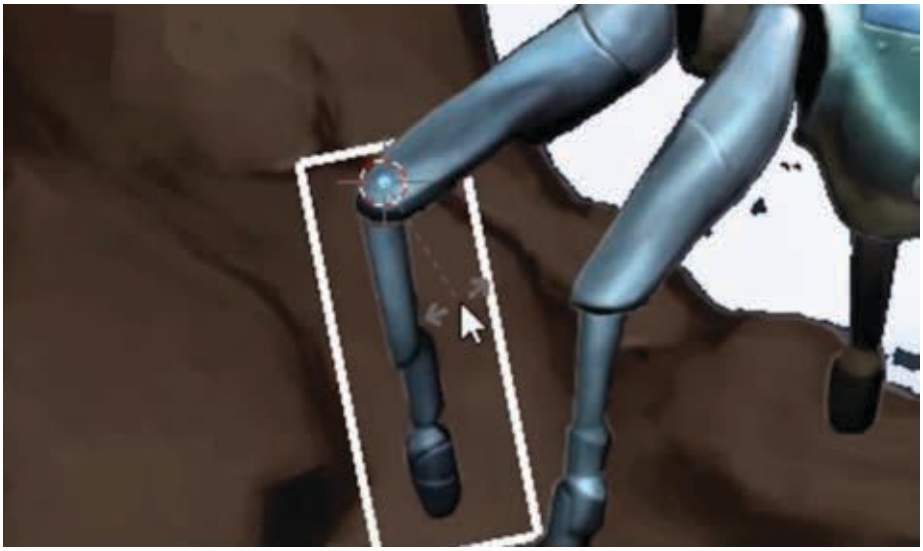


Now the 3D cursor is at the desired location.

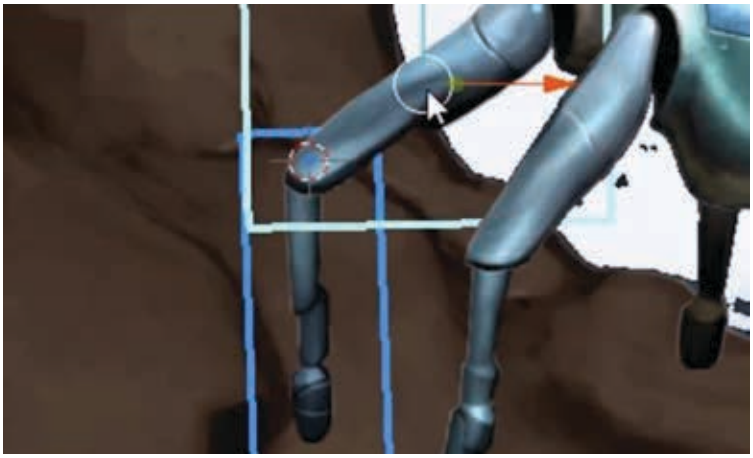
Make sure that the right object is selected, and go to Object > Set Origin > **Origin to 3D Cursor** (you may want to add this one to your Quick Favorites).



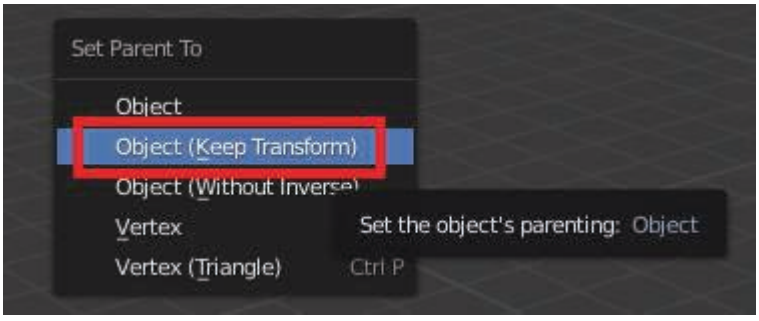
You can now rotate the part correctly from the pivot point.



That's how you set the pivot point. Now we need to establish a relationship between that lower leg and the upper leg. Shift+select the upper leg so that one is the Active object.



Press **Ctrl+P** and choose "Object (Keep Transform)" to make a parent-child relationship between the two parts. By selecting the upper leg last, that one is the parent of the lower leg.



Now that the body parts are parented to each other, they can be animated with keyframes by rotating them by **pressing R** and recording the keyframe by **pressingi** (choose “Rotation”). Repeat these steps for all body parts and connect the top leg parts to the body.

You can now animate this character for your sci-fior fantasy epic!

Bonus knowledge: In a case like this robot walker, to keep the feet nicely stable on the ground (without having to manually keyframe their position for each frame, which would be quite challenging) you may want to use IK or Inverse Kinematics instead.

IK sounds complex but all it means is that the relationships of the body parts are reversed. If in the above example the body of the robot is the “parent”. Where the body goes, the legs have to follow. When the legs move, the feet have to follow. Reversing those relationships means the feet are the boss, and then the legs and finally the body. That’s Inverse Kinematics.

In this case, that means you can place the feet somewhere, and animate the legs and body until you need to move the feet again. It will make the placement of the feet

completely stable. Depending on the situation, you need IK or FK (Forward Kinematics).

In a more elaborate rig for a character, normally there is an option to switch between both modes, even for individual body parts (the legs can be IK, the rest FK for example. Or the arms can be IK and the rest FK, like when a character is hanging on something by their hands. This cut-out animation robot is the simplest kind of animatable character that you can make, and as such has its limitations.

Anime tree in Krita and Blender

Let's make an NPR-style tree. **NPR** stands for **Non Photo-Realistic**, and is often used to describe renders that resemble traditional art styles or hand-drawn animation.

In Blender there are many ways to achieve this. In this case, we'll look at a hybrid 2D / 3D approach that uses the free painting software Krita to create some painted elements, and then Blender to bring it together and create a slight 3D feeling.

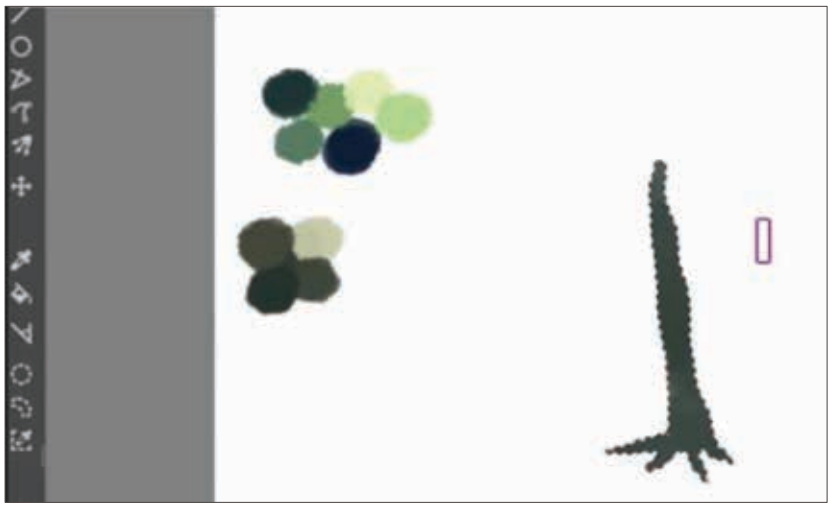
Of course, you could paint these elements in Blender as well. But I recommend having a look at Krita if you don't know it. I've tried many painting programs, some paid, some free, and I just really enjoy working in Krita. It's easy to learn and the result is somehow always better than if I use Photoshop.

Left: an NPR-style tree in all its glory.

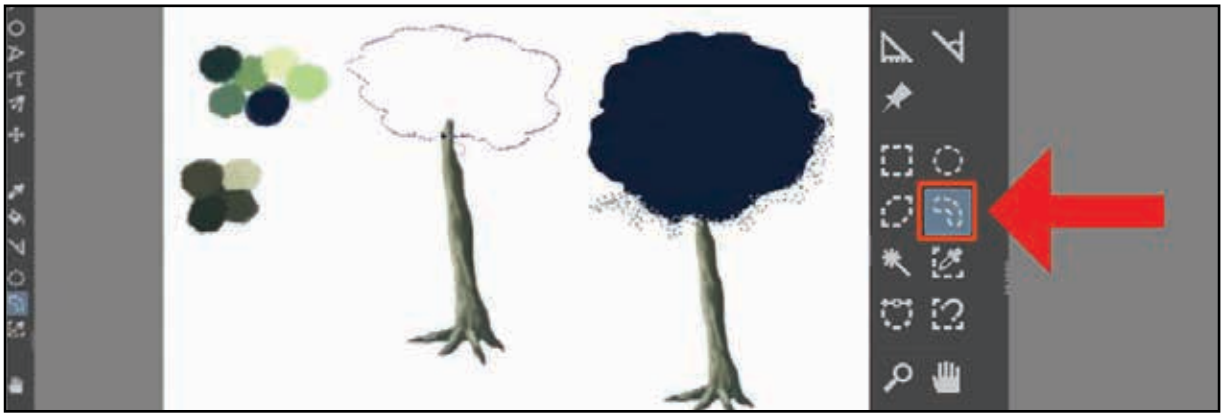
On the far right: the same tree, but rendered in a very low resolution.

This kind of gives it a retro-game look.

In Krita, draw shapes with the marquee-select tool.



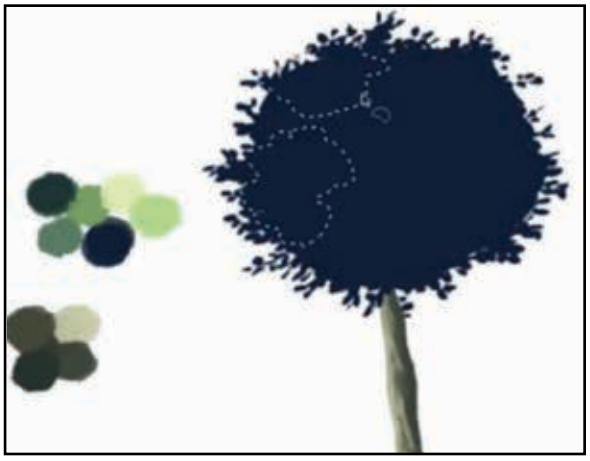
Of course you are welcome to paint every single leaf by hand (this is how I used to paint before knowing better) but trust me… it's much, much faster to just make a marquee-selection and then fill it with a color using the Brush tool. You can find the Marquee-selection tool in the left tool panel.

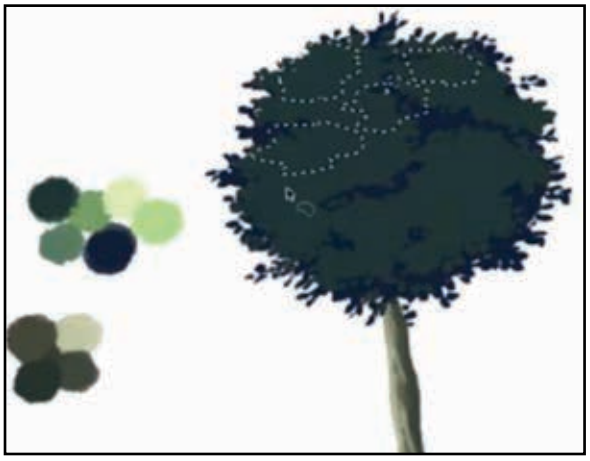


In Krita, you can pick colors from the palette with the **Ctrl** key. I usually paint a small palette of colors that I plan to use for easy color picking, or just pick the colors directly from the painting.

Go from dark to light colors.

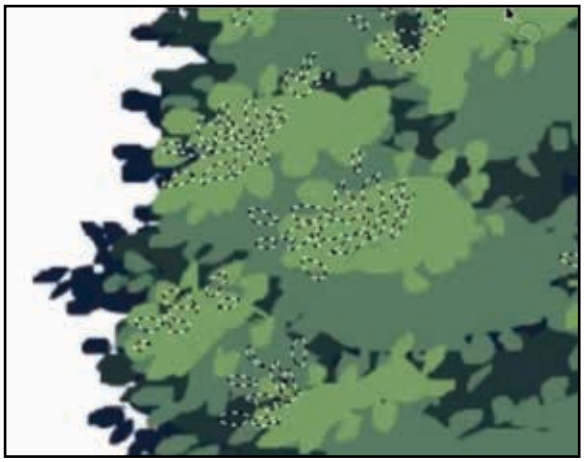
Decrease amount of leaves (in other words, decrease the size of your marquee selections) as they get lighter.



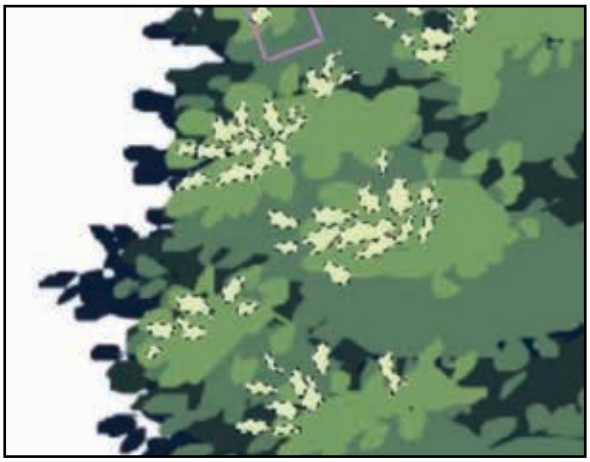


Below you can see that as I get to lighter colors, I also decrease the size of the marquee selection. Each color needs to be on a separate layer. The trunk should be on its own layer as well.

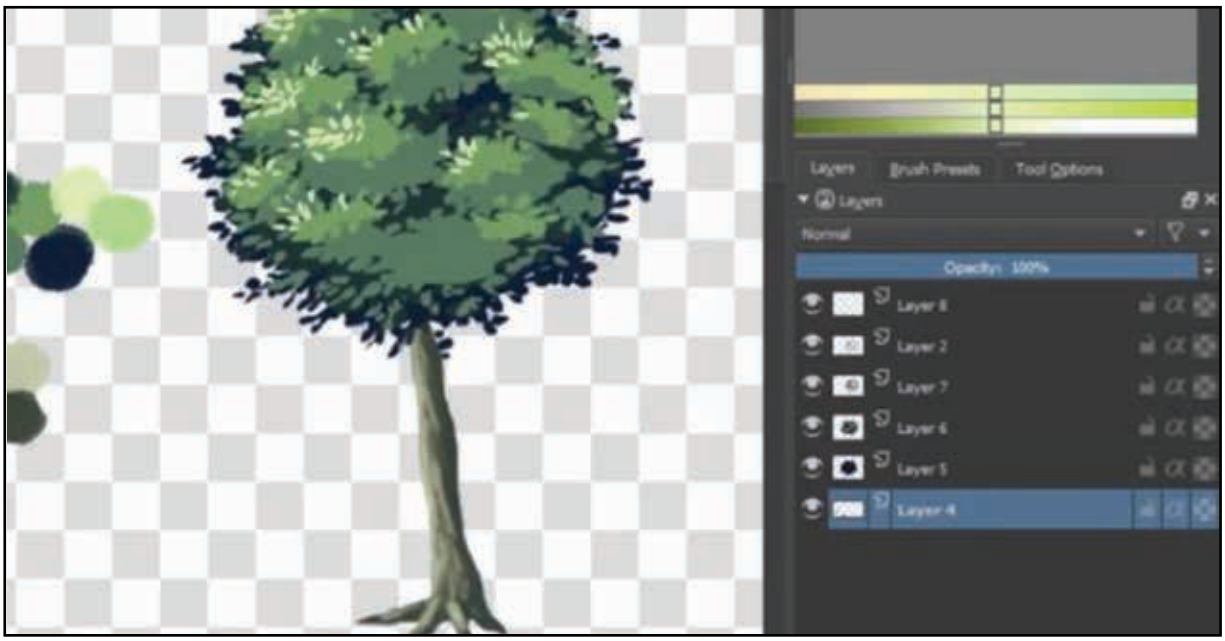
Here you see close-up selection drawing shapes.



The same selection after being painted in with a light green.

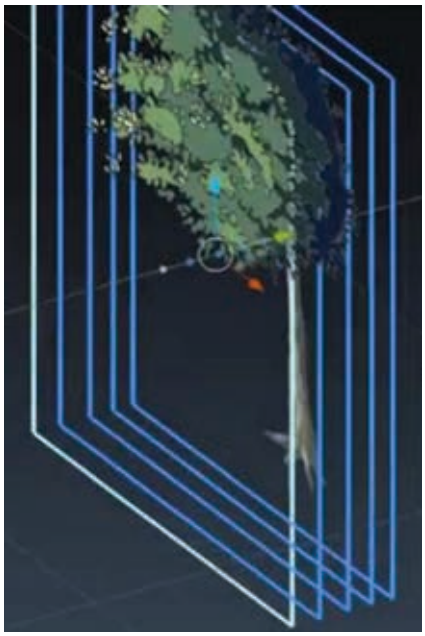


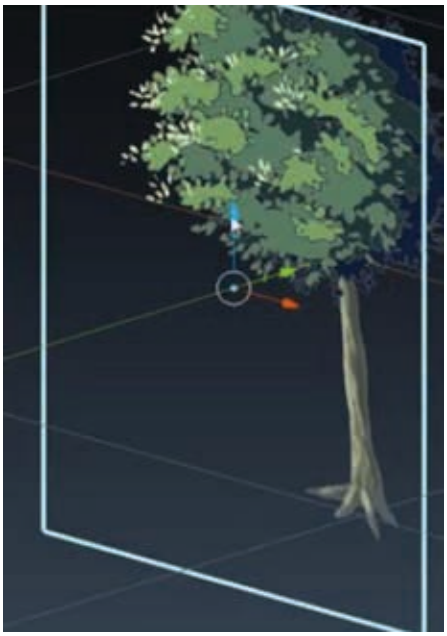
All of this is on separate layers. **Export each layer as a PNG file** with transparency. To do this, you need to have only that layer visible (the eye icon for the other layers needs to be turned off), then go to File > Export and choose PNG as the file format.



In Blender use the **Import Images As Planes** addon which comes with Blender by default but needs to be turned on, to import the images.

As you can see, I've moved each image plane so there is a bit of distance between each. It should go from light to dark, front to back.

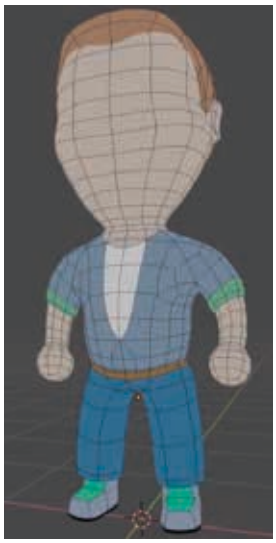




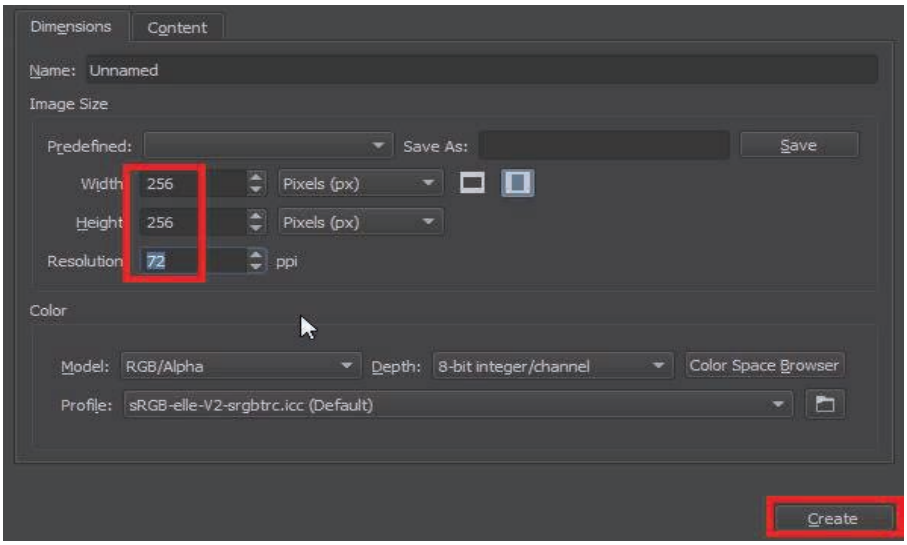
Instead of a Diffuse shader I use an **Emission shader** so the light or shadows don't affect the tree. There is a slight parallax effect as you move the camera around the tree, making it both look 3D and 2D.

Low poly texture atlas

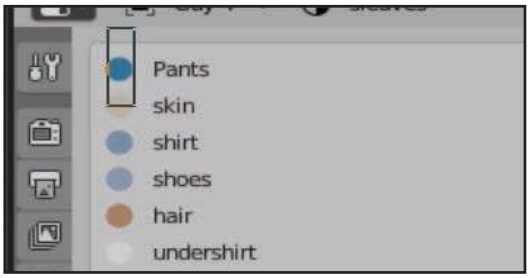
This character has faces with one color materials assigned to it. Instead of having a dozen materials, what we are going to do is use one material with a simple texture called a **texture atlas**.



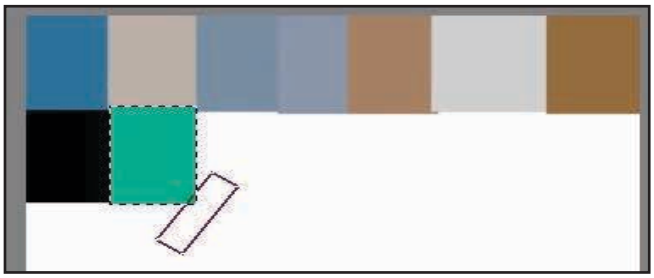
In Krita color create a new 256x256 texture.



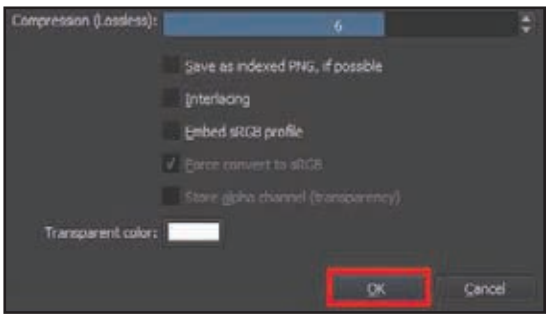
In my case I took a screenshot of the materials in Blender, then color-picked from that screenshot.



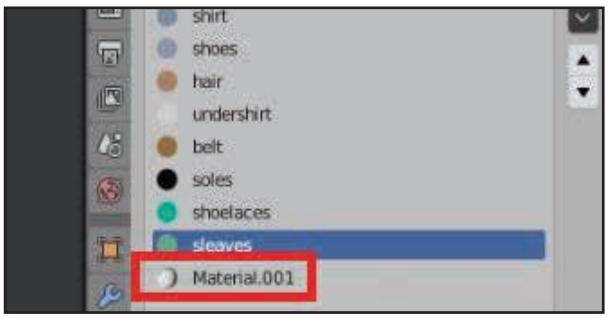
Then painted those colors in box selections, to create simple color blocks in my Krita document. You can color-pick in Krita by **holding Ctrl and left-clicking**.



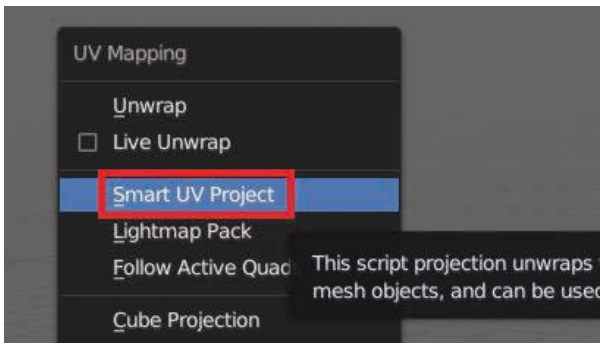
Export as a PNG and load it in a new material in Blender.



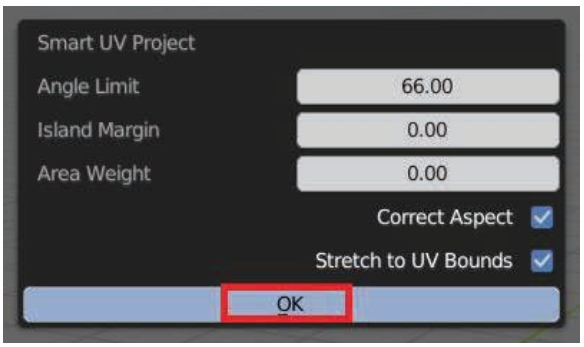
Load the PNG texture in the new material.



In Edit Mode select all faces of the model by **pressing A**, then **Press U** and choose **Smart UV Unwrap**.



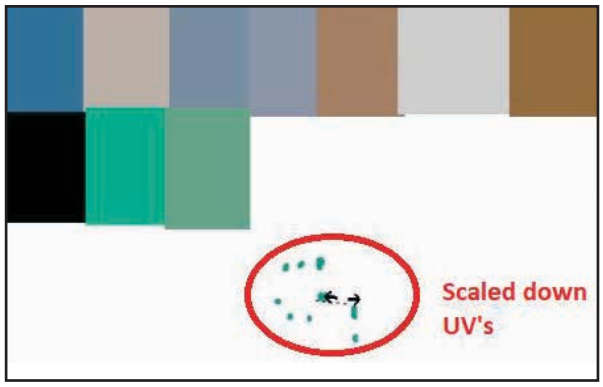
Click on **OK** (the default settings are fine).



Now select all the faces per material and scale them to the color block on the texture.

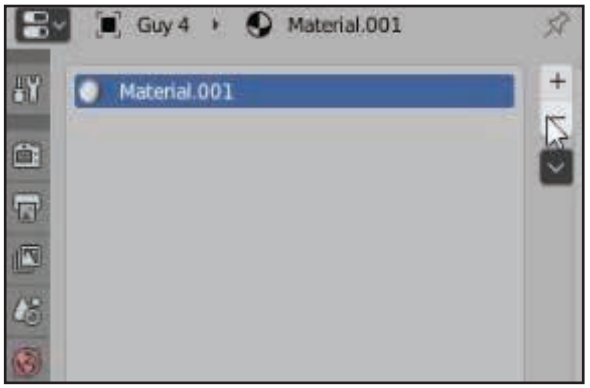
In other words, the size or shape of the UV's is not important as long as they are scaled down to fit inside the correct color block.

Do this with the UV's assigned to each color material.



Now you can delete all the individual color materials, all you need is the texture now.

Press the - icon until only the new Material with the PNG texture is left.



COMPOSITING

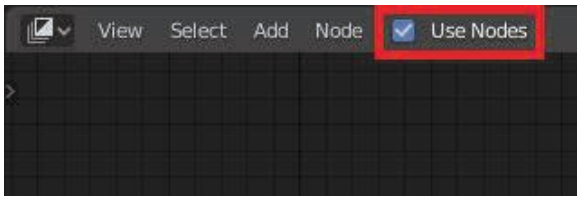
Rendering passes as separate files

For compositing it's nice to have at least an AO pass (for extra shadow control) and Mist (for depth of field etc).

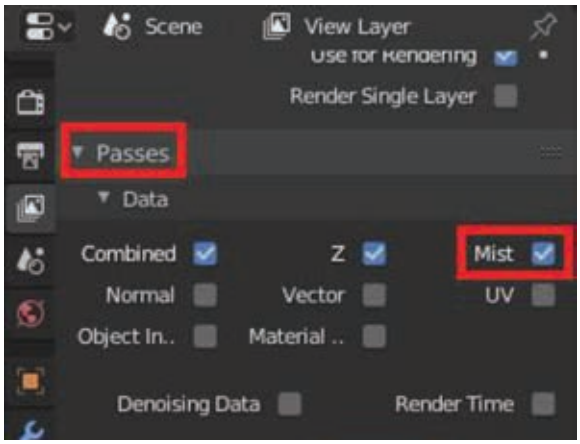
I recommend using **Multilayer EXR files**, because then you can have all your render passes in a single file.

Sometimes that's not an option (on some render farms for example). In such a case, render the passes as separate files.

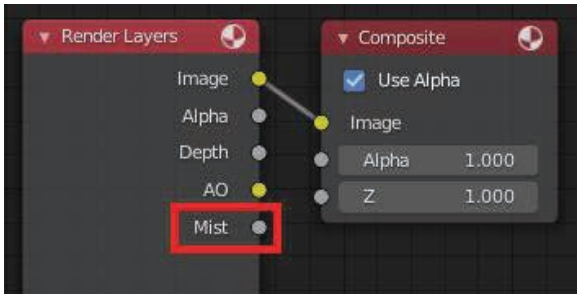
In the Compositor, click on **Use Nodes**.



In the **Passes** section of the Scene tab, enable the passes you need, Mist for example.

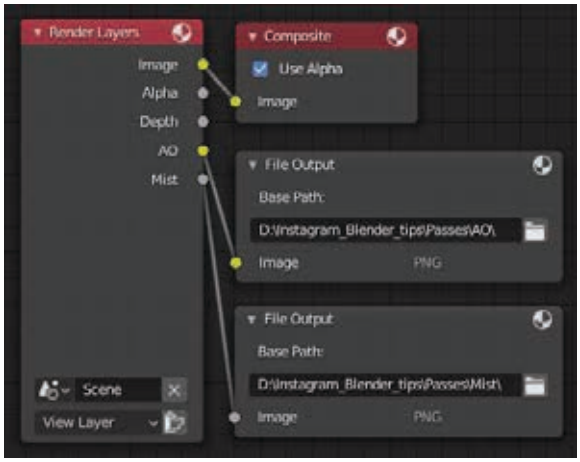


You will notice that the RenderLayers nodein the Compositor updates to show the Mist output socket.

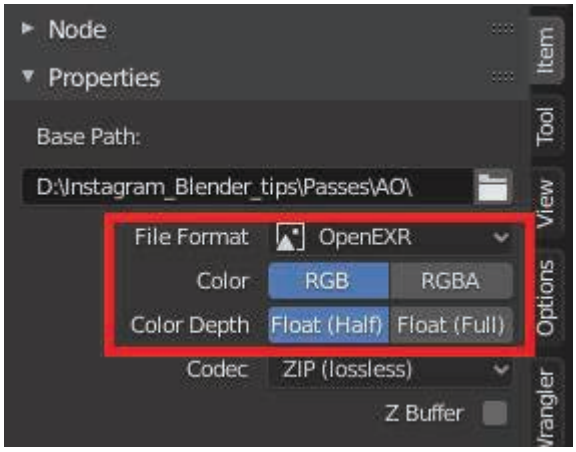


In this case there are two extra passes: **AO** (Ambient Occlusion) and **Mist**.

Add **File Output nodes** for both, and set a path where you want these nodes to write their files to.



In the Properties tab of the Compositor window, you can change the properties of the selected File Output node.



In case you are compositing inside of Blender, I recommend using **OpenEXR**. In case you use After Effects, I would use PNG instead.

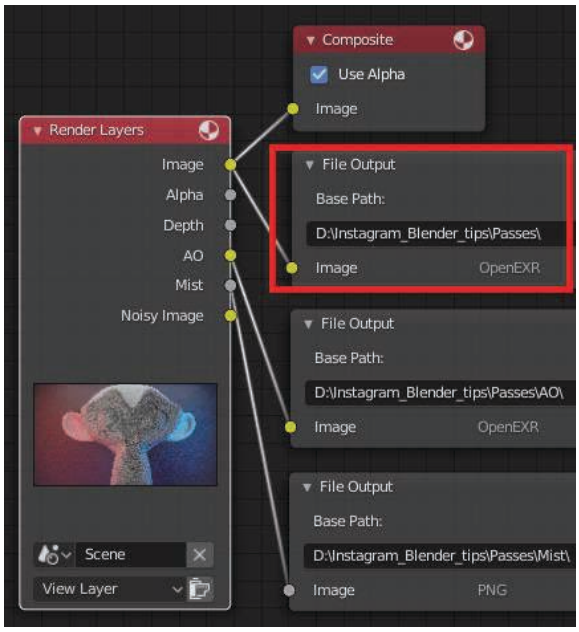
If you render now, the extra passes will be written to separate files.

Please note that Blender does not write a still image render to a file by default.

So your extra passes will get written to files but not your main combined render, unless you save it manually!

Solution is to either add a File Output node to the **Image output**, or to render a **one rame length animation**. That will get written to a file.

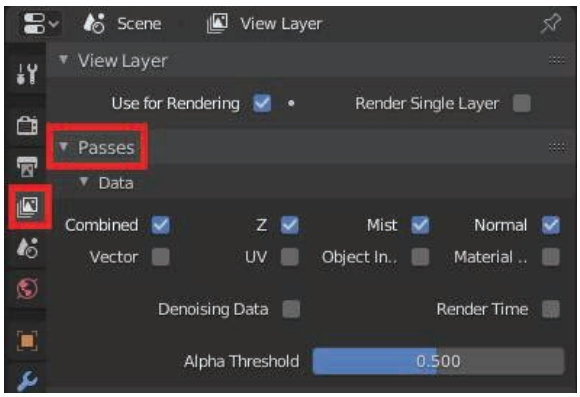
Pictured: a File Output node added for the Image output, so everything gets written to a file.



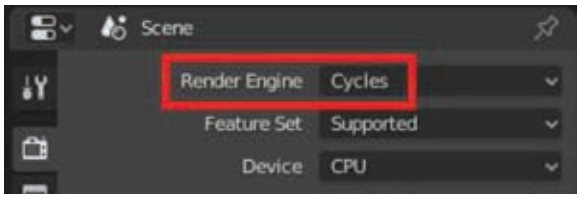
Click here to return to the index.

Render Passes

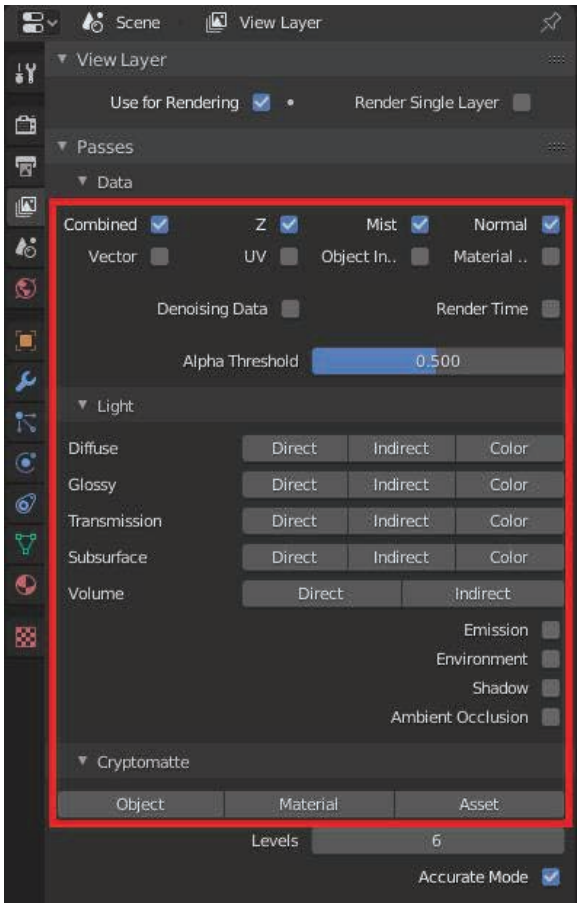
Render passes are a powerful part of Blender that can take your renders to the next level.



Note that while Eevee has some render passes, Cycles has many more.



You can turn on which ones you need. These are all the Cycles render passes.



The passes under Data are based on the 3D information of the scene. For example the Mist pass is based on the Z-depth in the scene.

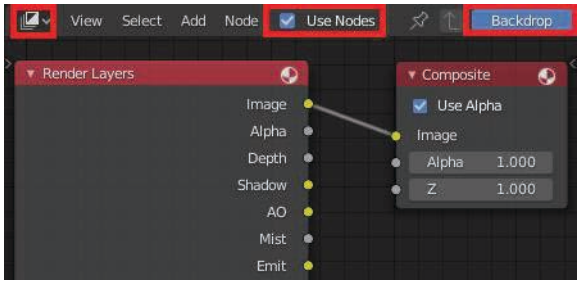
Under Light you can use the passes to split up the render in the different parts of the lighting. This gives you more control during compositing by combining them back together in the way that looks best.

The Emission, Environment, Shadow and AO passes are very useful for more compositing control as well.

Finally Cryptomatte are used to create masks.

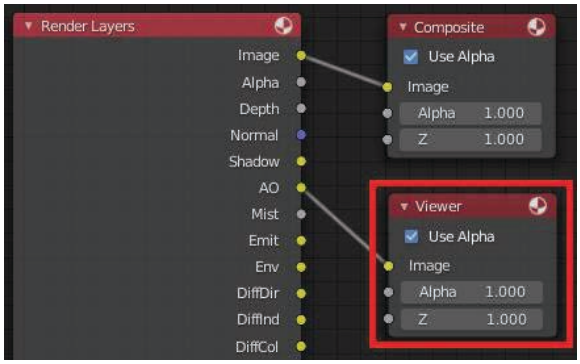
To generate any of these passes, just turn them on before rendering.

After we render our images, we go to the Compositor.



Make sure **Use Nodes** and **Backdrop**are turned on.

The Render Layers node in the Compositor has outputs for all the render passes that you've turned on before rendering.



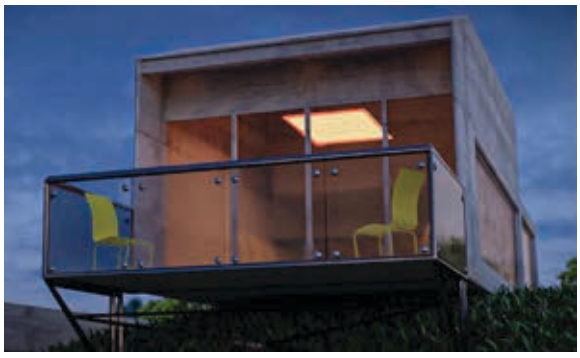
Add a Viewer node by **pressing Shift+A** and searching for "**viewer**". This allows us to see the different passes in the Compositor window (as long as something's been rendered).

Press Alt+V to zoom in the backdrop and **press V** to zoom out.

Here you see the AO pass. It can be used to make the shadows a bit more exaggerated and separate objects visually in a more readable manner.

Free Archviz scene by Emiliano Colantoni.

In the next section we'll look at compositing such a pass, the Mist pass, to make our render look more interesting.



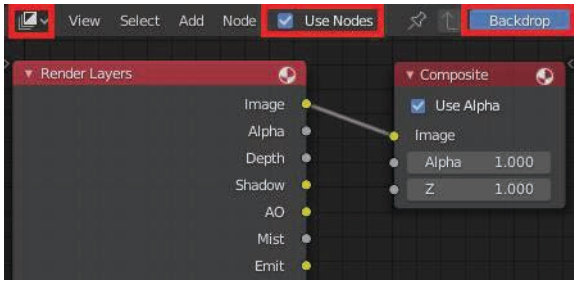
Click here to return to the index.

Compositing: Mist pass

In the previous tutorial we looked at the different render passes.

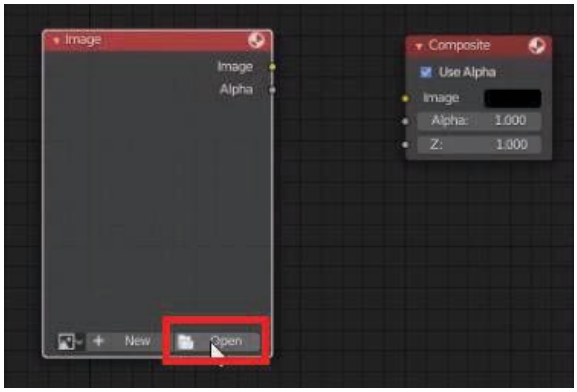
Here we will use the Mist pass to create this composite image.

In the Compositing layout, enable **Use nodes** and **Backdrop**.



Delete the Render layers node.

Press Shift+A and search for an **Image node**. On the Image node click on Open to load the previously rendered image. We're using an **OpenEXR image**, which has all the render layers embedded in one file.



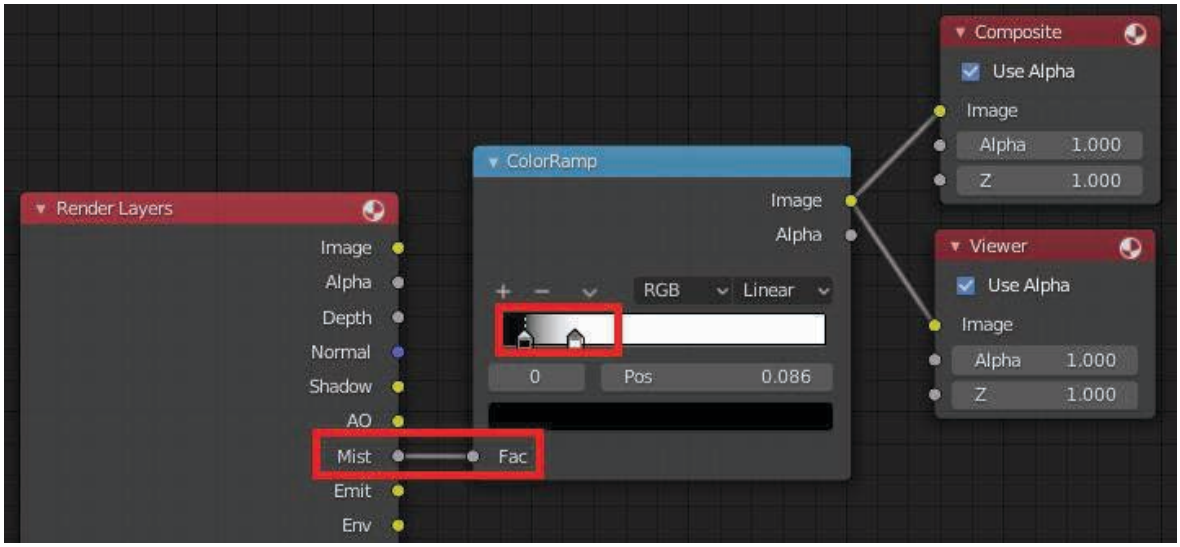
Press Shift+A and find a **Viewer node** so we can see what we are doing.

Connect any of the passes to the Viewer node to see what it looks like. The **Combined pass** is what you normally get when rendering.

The Mist output gives us the Mist pass. **Press Shift+A** and find the **ColorRamp node**.

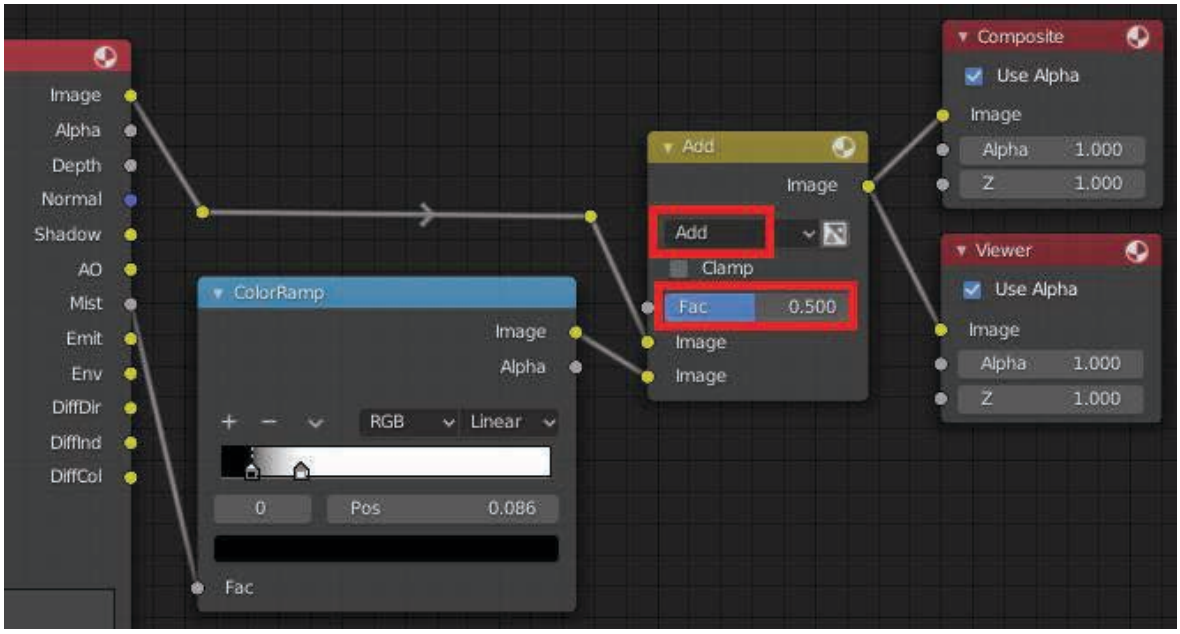
Add it in between the render layers Mist output and the Viewer node. You can also connect it to the Composite node, this is the node used for the final render.

Changing the ColorRamp values changes the mist appearance.



Bringing the ColorRamp values closer together creates more contrast, which affects the look of the Mist.

Now to combine the Mist and the normal image we add a **Mix node** and set it to **Add**.



The **Fac** value of the **Mix node** gives us more control over the appearance of the mist in the image.

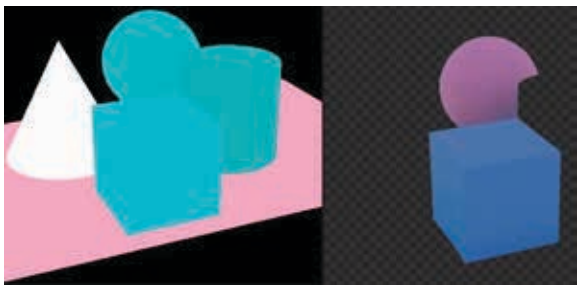
Now we have the regular rendered image, mixed with mist. This creates an interesting look! Here you see anexaggerated version of this effect, but you can use it in a subtle way to add realism.

As long as the Composite node gets the same input as the Viewer node we just have to **press F12** to render the composite image.

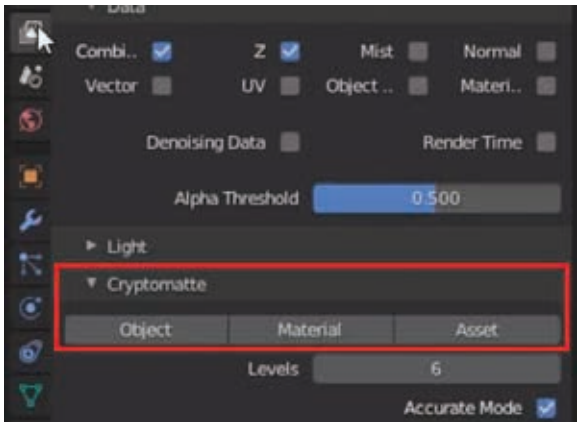
Archviz scene by Emiliano Colantoni.

Compositing: Cryptomatte

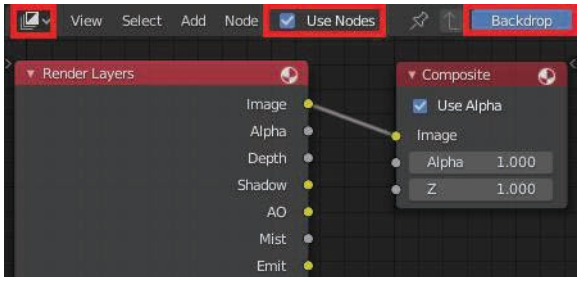
Cryptomatte is a clever new node that lets you pick objects or materials to mask from a render.



To use it, enable one or more Cryptomatte render passes and render your image.

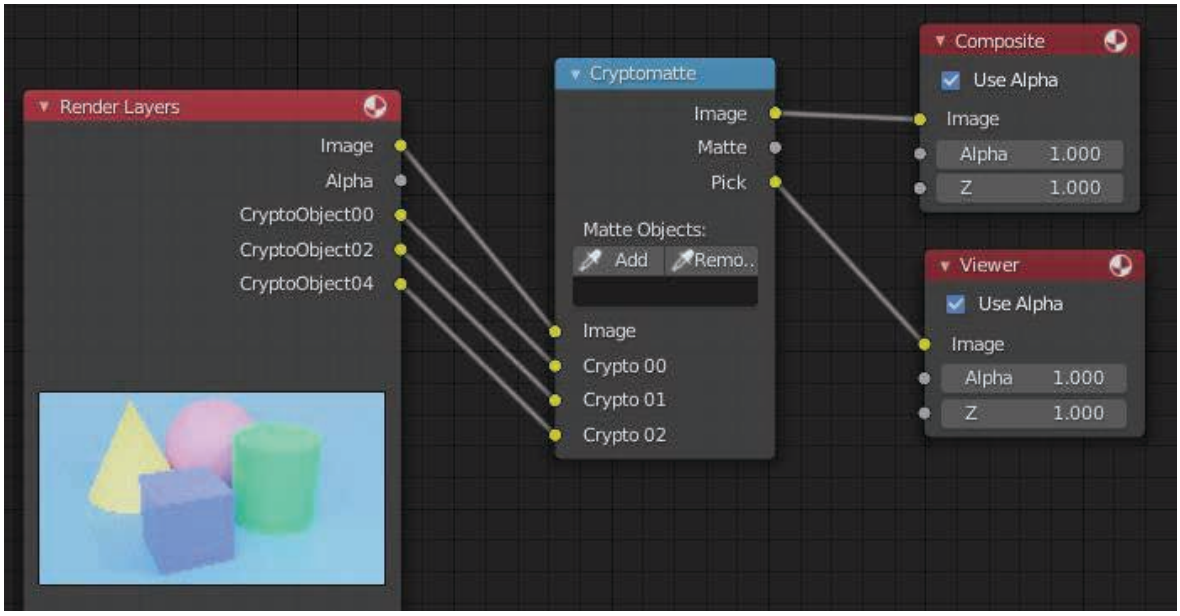


Make sure **Use Nodes** and **Backdrop** are activated in the Compositor.



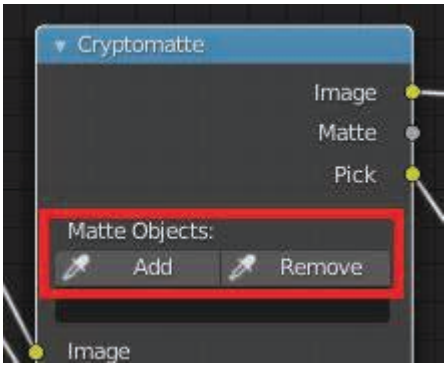
In the Compositor, add a **Viewer node** and a **Cryptomatte node**.

Click here to return to the index.



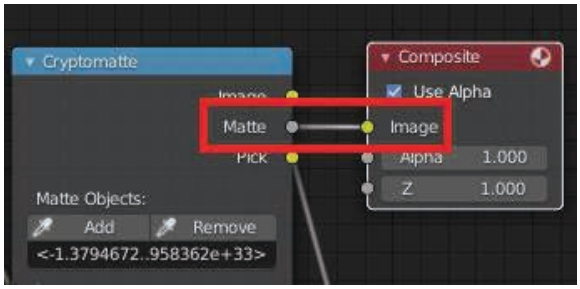
Connect the **Pick output** to the Viewer node. Connect the Image output to the Composite node. As shown in the image above.

Now you can click to **Add** or **Remove** and Mask objects.



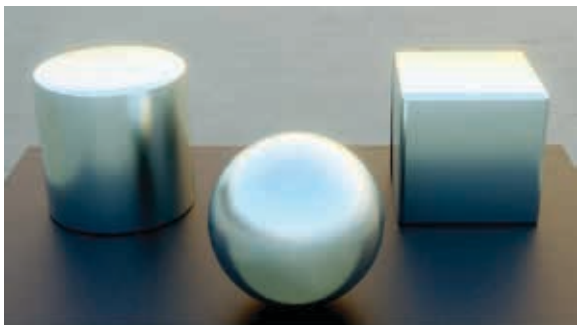
Click on **Add** to get the picker mouse cursor, then click on the backdrop image (below left) on the items you want to include in your render image. The render result (below right) will update to reflect your choices. Click on **Remove** to pick items you want to remove.

You can also use the Matte output if you need a black and white mask image. Connect it to the Composite node to render the mask (for later use in another Compositing program for example, or when rendering on a render farm that doesn't export to a multipass format like OpenEXR).

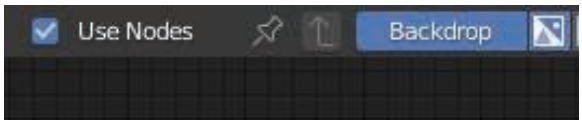


Compositing: Color Correction

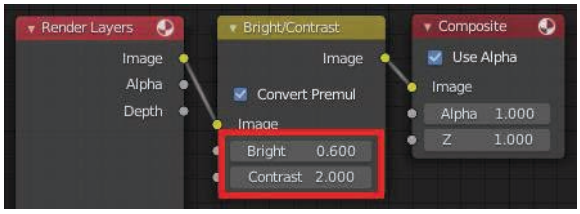
After you finish your render, you can make it look nicer in the Compositor with some node magic.



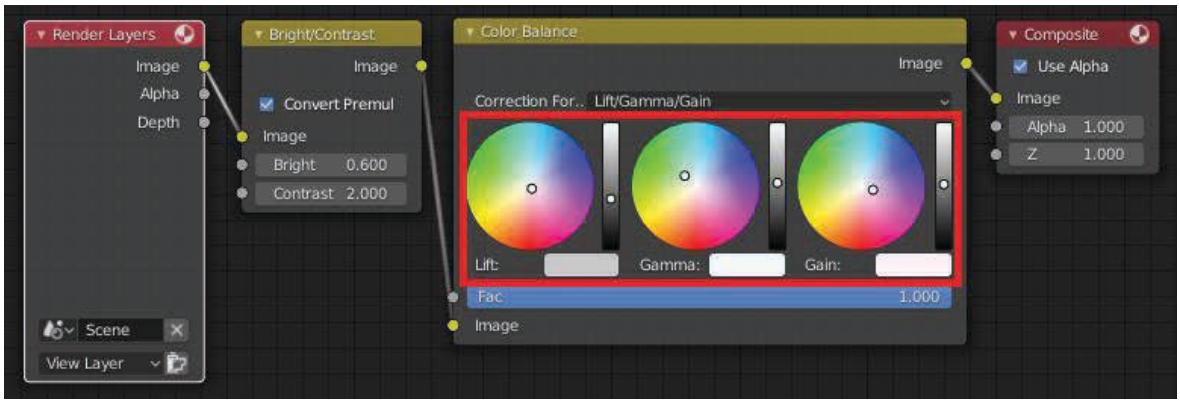
Remember to turn on **Use Nodes** and the Backdrop in the Compositor window, so you can see what you are doing.



A **Bright/Contrast node** always helps. Here I increase the contrast and make it a bit brighter as well.

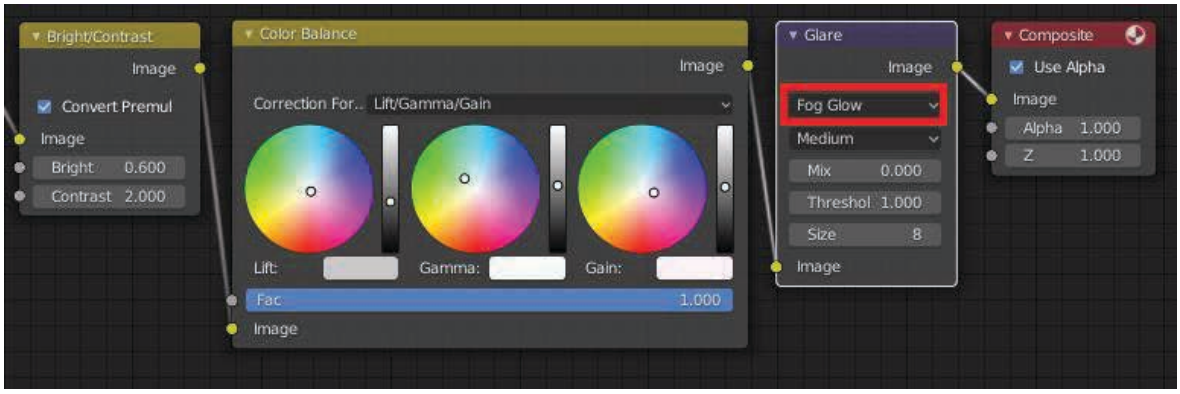


Next, let's add a **Color Balance node**. I like to adjust the middle values, and leave the other two as they are.

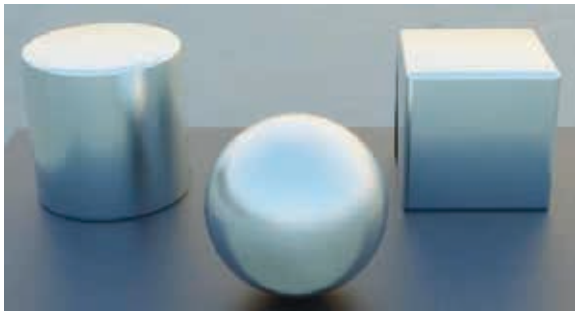


For the lower and brighter values (the left and right color spheres) I tend to only change the brightness.

A **Glare node** also adds some extra visual interest. I think "**Fog Glow**" looks nice.

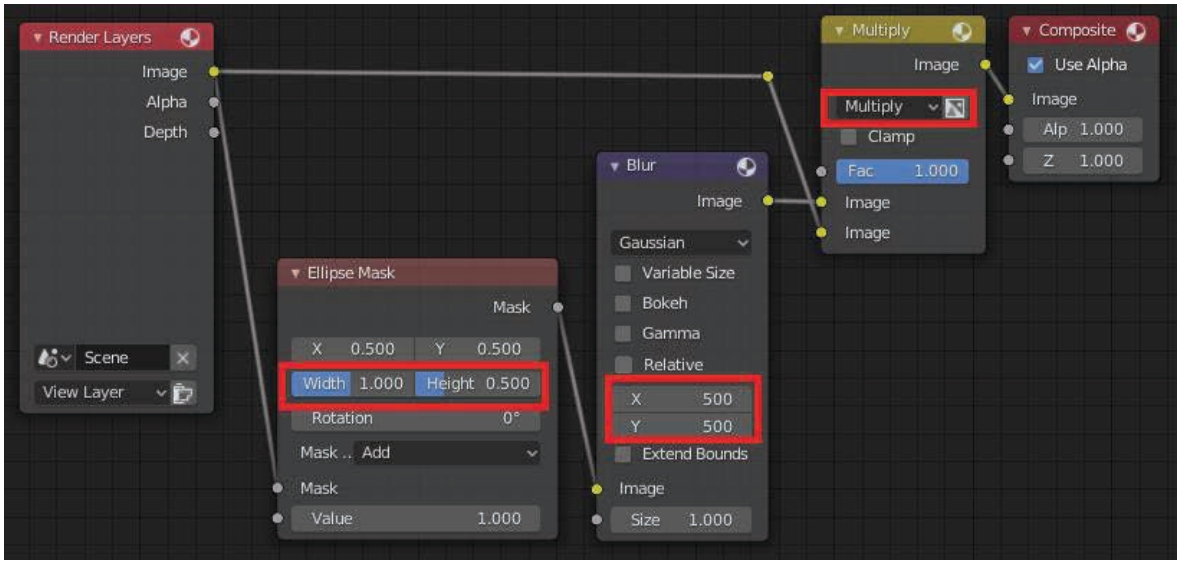


Below you can see the difference without these nodes (left) and with them (right). The difference is subtle, but all these little things together can make a much better render.



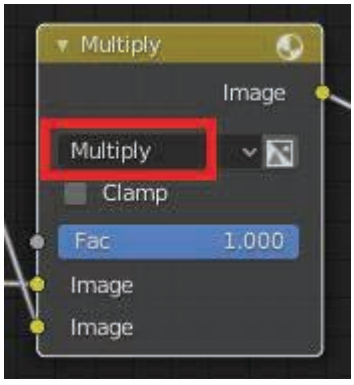
Compositing: Vignette

Adding a Vignette adds some more visual interest to our render.

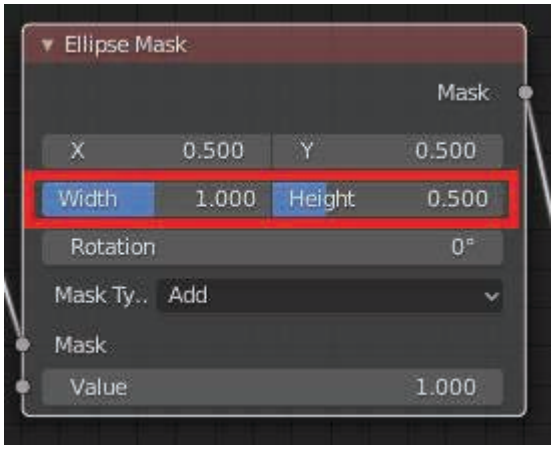


To start add a **Mix node** and set it to **Multiply**.

Plug the **Render Layers** as well as the output of the Blur node into the Mix node.

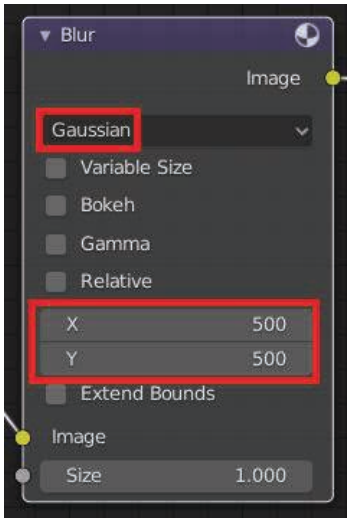


Add an **Ellipse Mask node**, plug it into the Mix node. The **Width** and **Height** values are important for the shape of the ellipse that creates the vignette. They should reflect (more or less) the aspect ratio of your image.

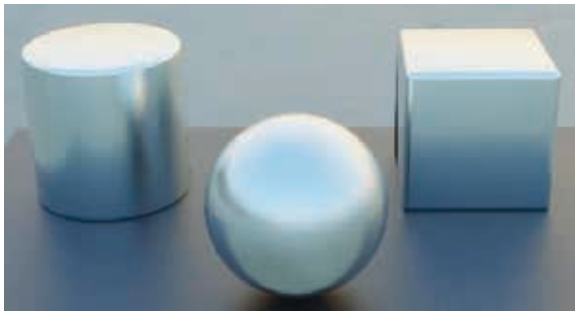


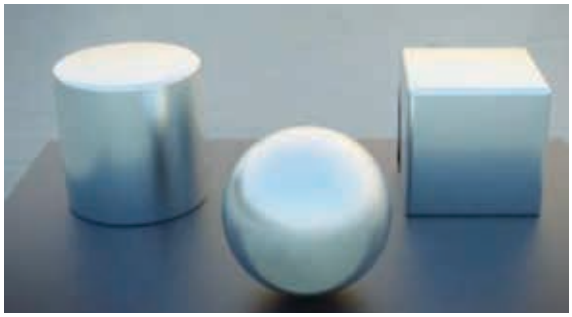
Finally add a **Blur node** and set it to a high enough value (500 in this case on both axis) so that the Ellipse Mask is blurred enough.

Gaussian is a blur method that looks good.



Here you see the render without vignette (left) and with vignette (right). It's very subtle, but it makes the image just a little bit less flat by making the image brighter in the center and darker around the edges of the frame. It is a subtle color correction effect very commonly used in movies.

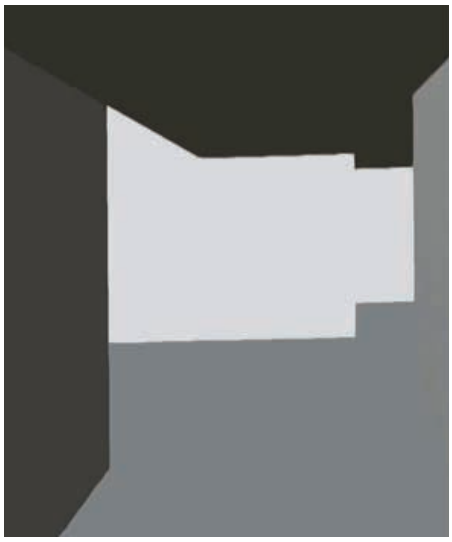




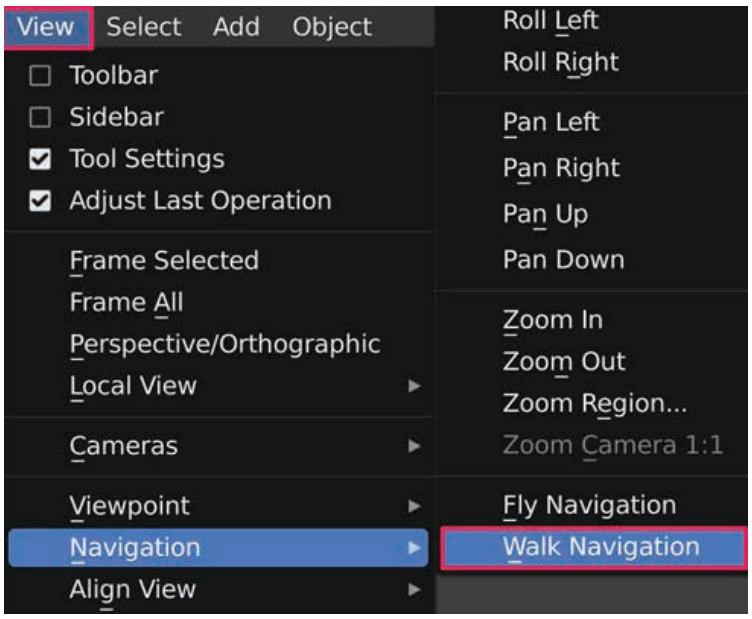
WORKFLOW TIPS

2 tips for navigating in a big environment scene

Navigating in an environment can be tricky once the scene is big.

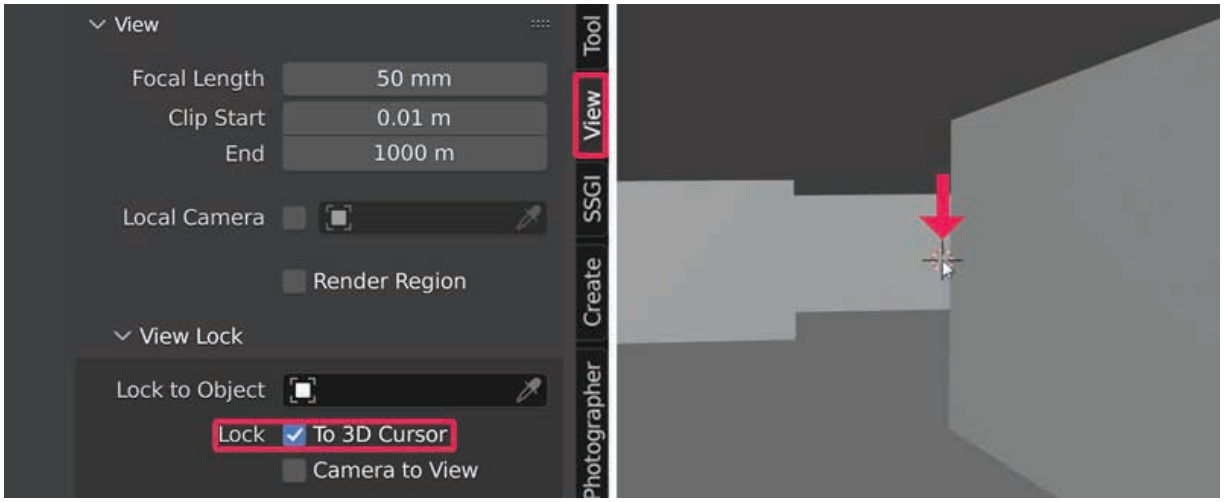


In such a case you can use Walk mode.



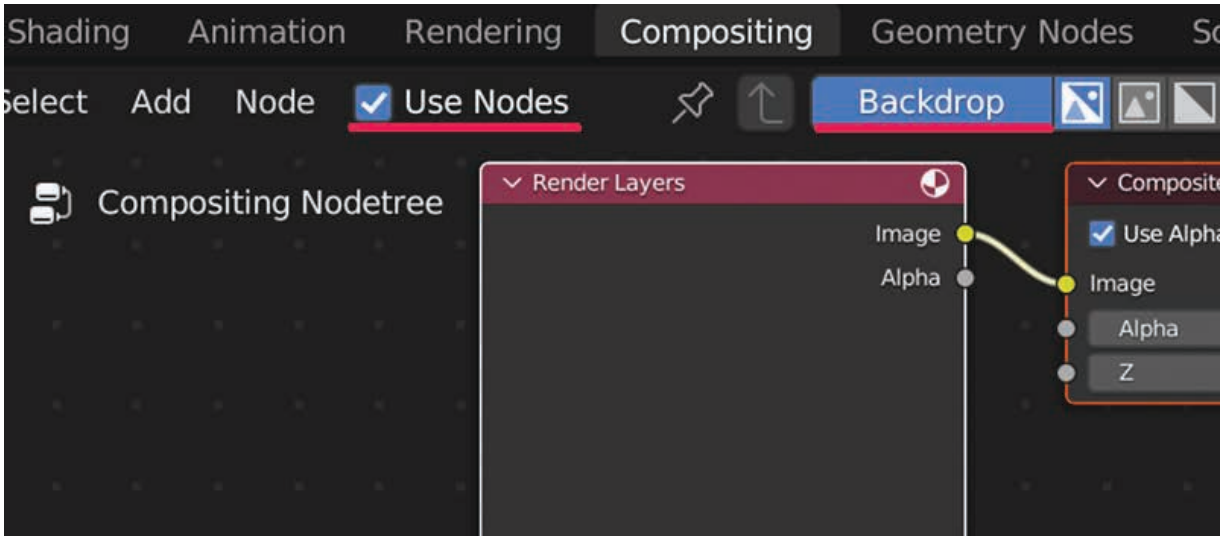
Press G to activate gravity. Just make sure you're actually inside the model, or you will fall down in the 3D void... Now you can walk around in your scene with the arrow keys and the mouse. It's a bit like an old videogame. Wolfenstein 3D, but without textures!

Another useful feature is to turn on **Lock to 3D cursor**. With this turned on, the view will center around where you put the 3D cursor.

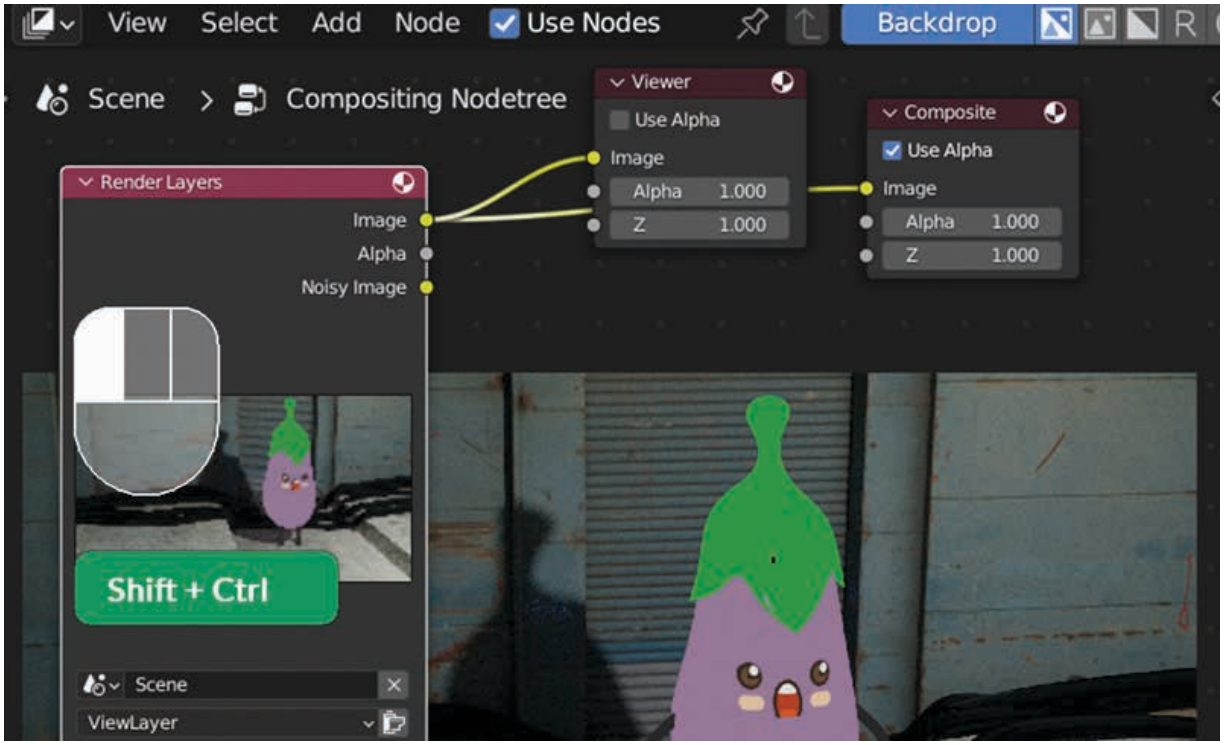


Flip Node

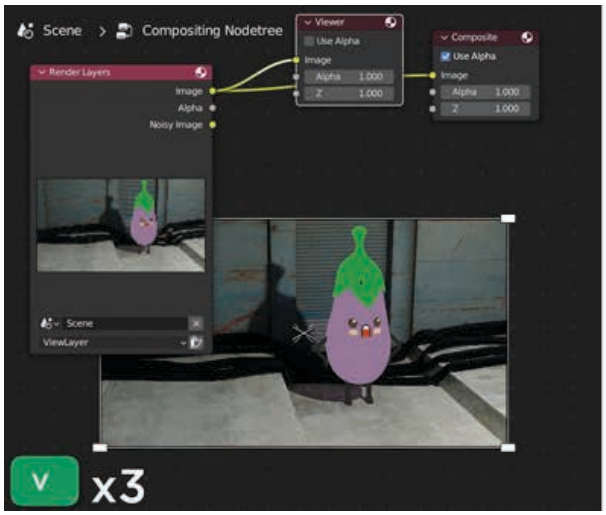
When creating a render it can be useful to flip the image horizontally as it helps you spot problems with the composition. To do this render the image and in the compositor turn on **Use Nodes** and **Backdrop**.



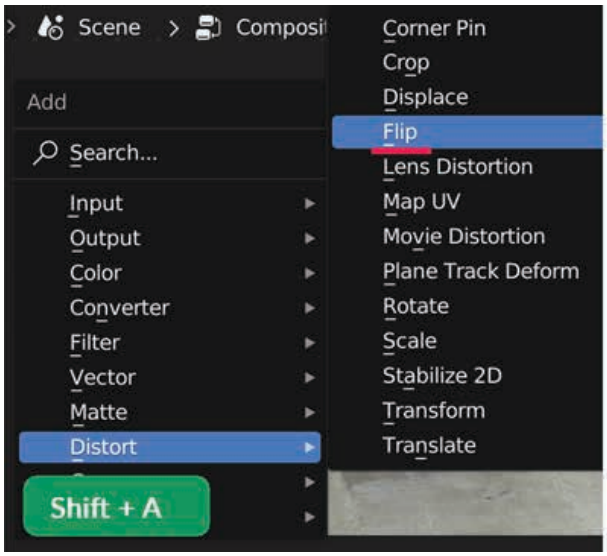
Shift+Ctrl+LMB click on the **Render Layers** node to create a **Viewer** node.

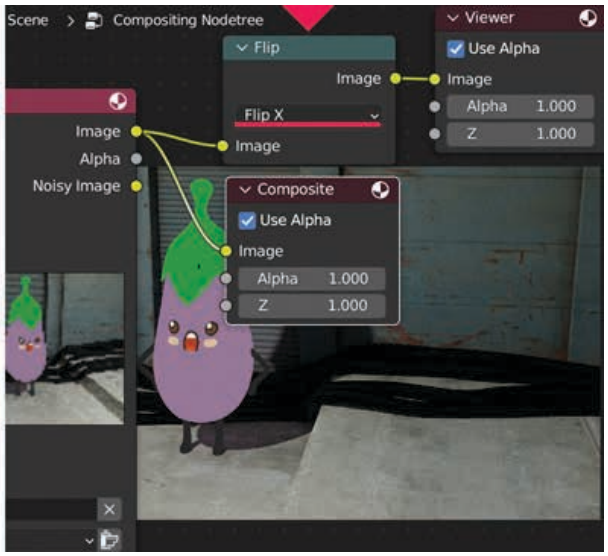


You can zoom the rendered image in and out using the keyboard shortcuts **v and Alt+v.**

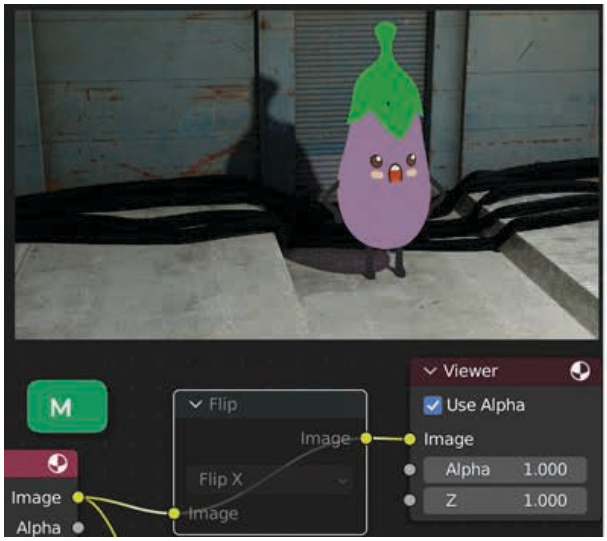


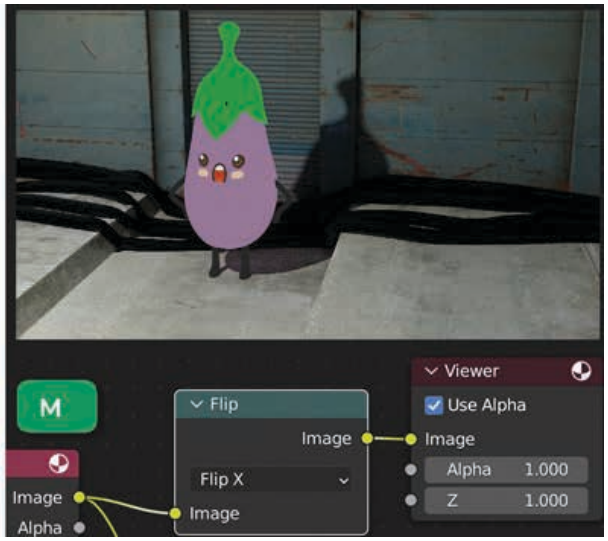
Now add a **Flip node** in between the Render Layers and the Viewer.





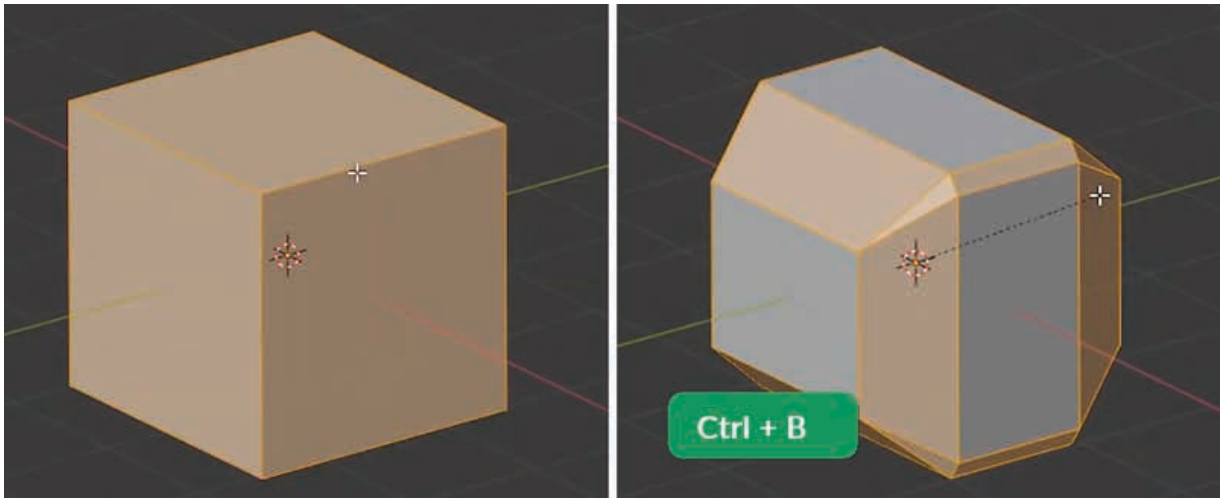
You can now toggle the Flip node on and off by selecting it and **pressing M** (for Mute).



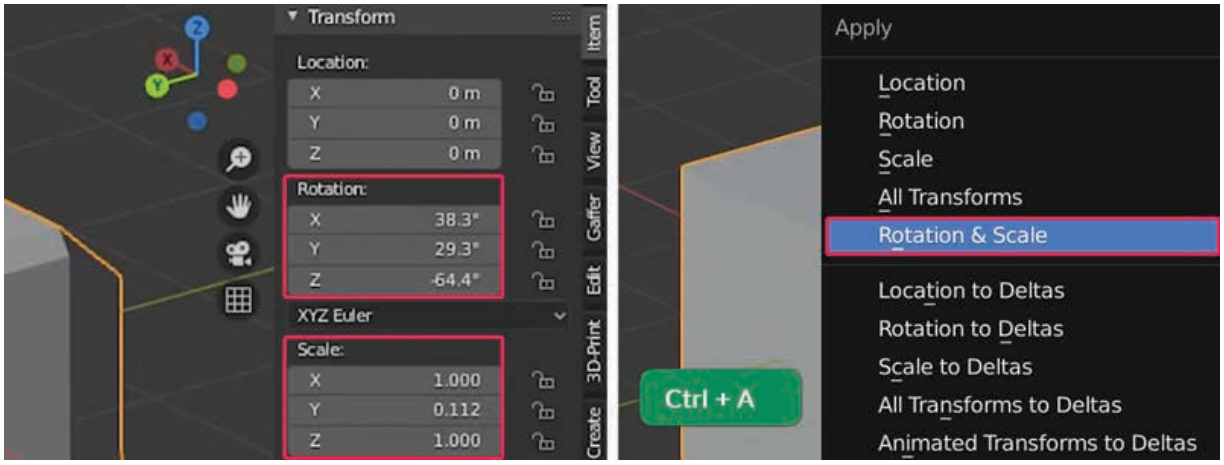


Top 3 solutions to Blender issues

As a Blender tutorial guy, people often send me messages asking for help with Blender issues. I've noticed that often they are the same problems, so here are my top 3 solutions. **Number one: Non-uniform Bevels**.

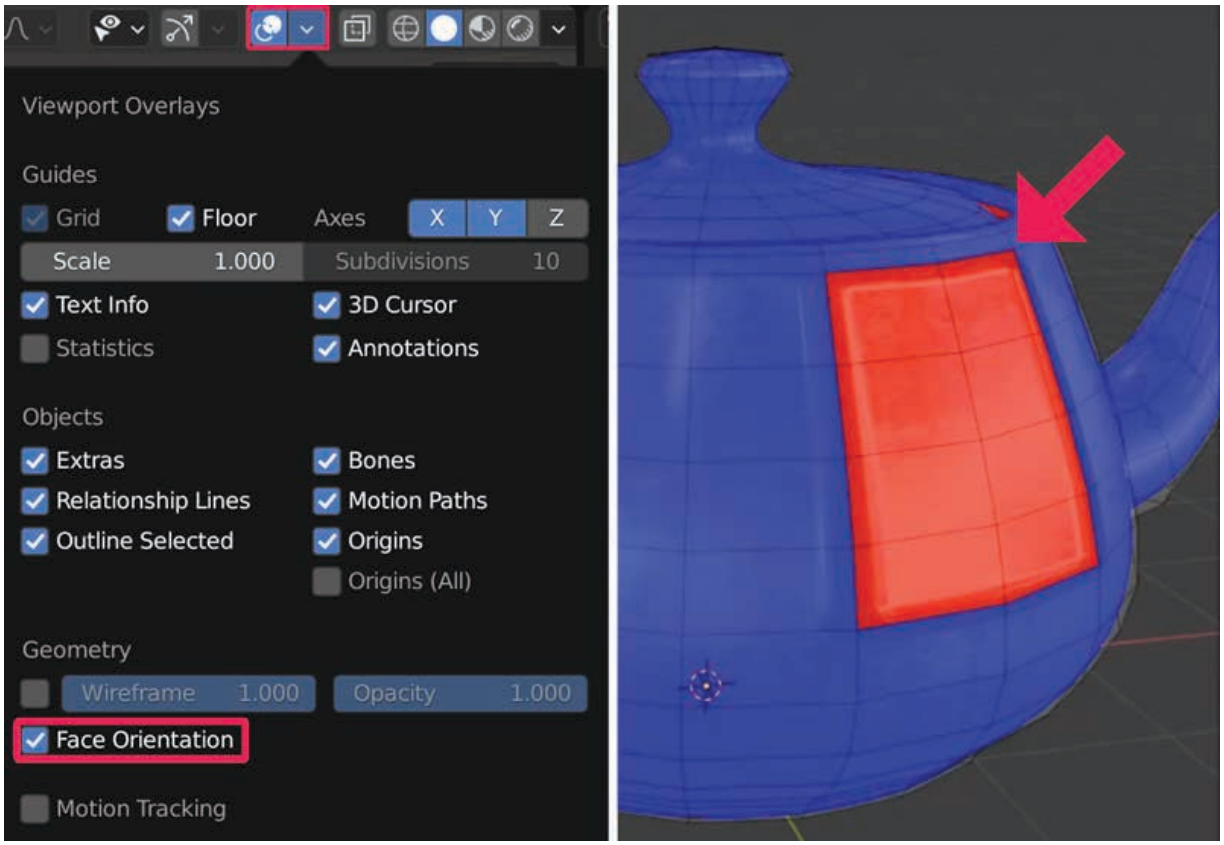


Check if the scale and rotation of your object are applied in the Option panel. If not, **press Ctrl+A** and choose "**Scale**" or "**Rotation and Scale**" in Object mode. Now, your bevels should be uniform. You don't necessarily need to apply the rotation to solve this issue, but you might as well as not doing so can also lead to some confusion.

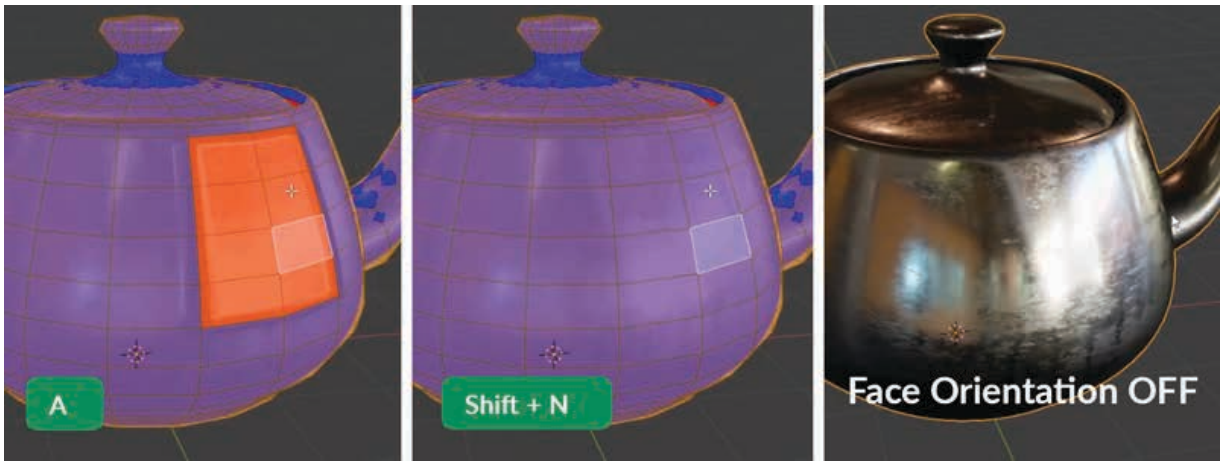


Number two: Flipped normals. This can be seen as darker parts of the material, or just plain weird shading like this:

In **Overlays**, check **Face Orientation**. If there's faces in different colors that means some normals are facing the wrong way.

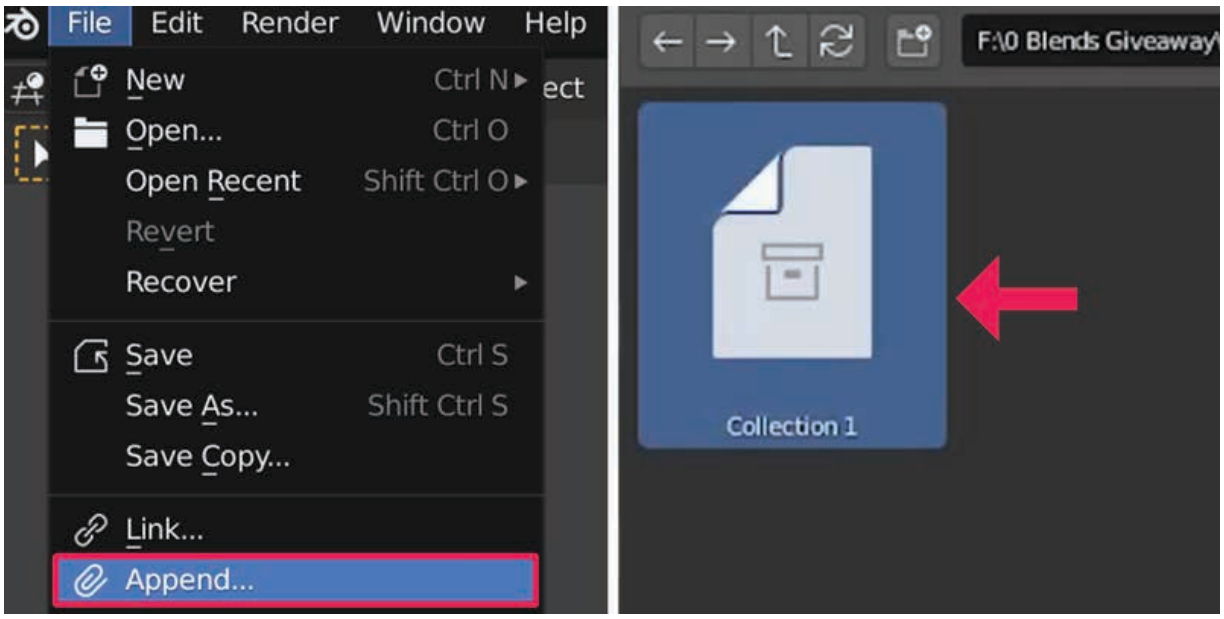


Select all Faces by **pressing A** in Edit Mode and **press Shift+N** to recalculate them.



Number three: Random issues that you just can't put your finger on.

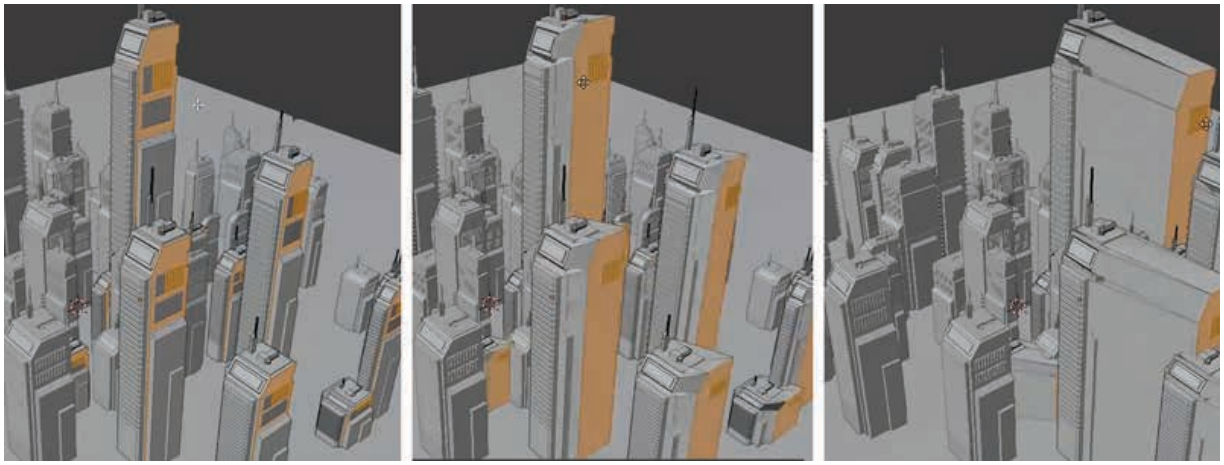
I find that 90% of issues can be solved by starting **a fresh Blend** file and **Appending** whatever Objects or Collection you were working on from your previous file.



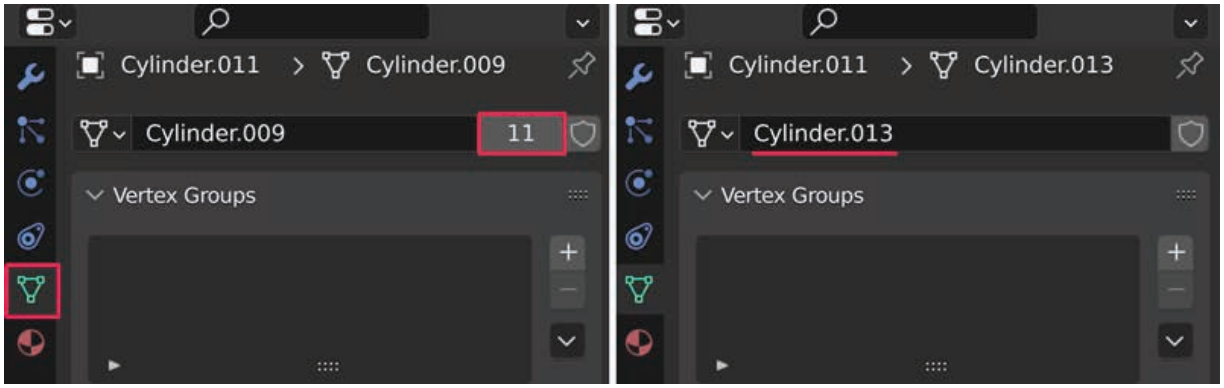
It's simple but this literally solves most of the problems people e-mail me about... *without even looking at their Blend files.*

Make linked duplicates unique with one click

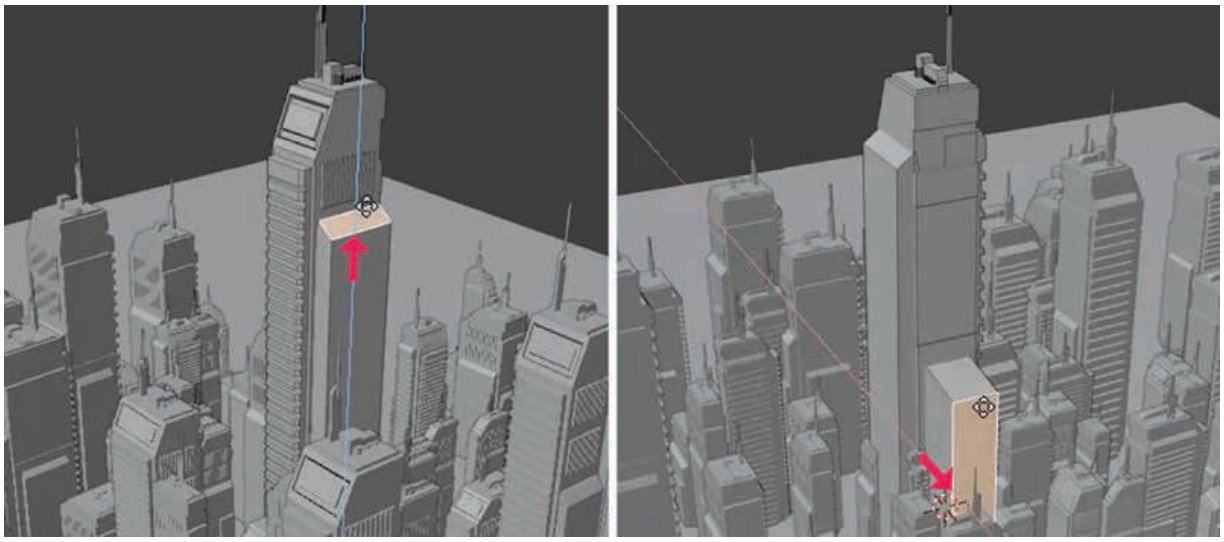
If your scene has a lot of Linked Duplicates, to make one unique so that you can make changes to it in Edit Mode.



All you have to do is to the **Object Data Properties** tab. Then, click on the number next to the name of the object.

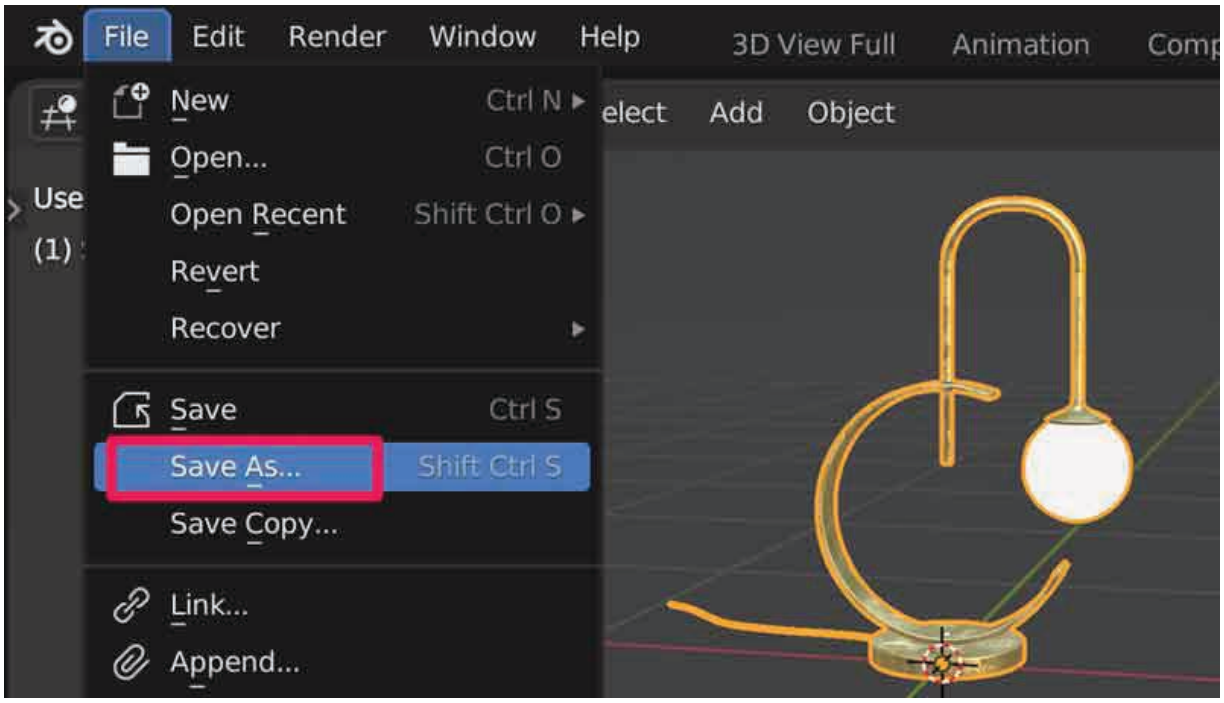


Now you can edit the object in Edit Mode without affecting all the other duplicates.

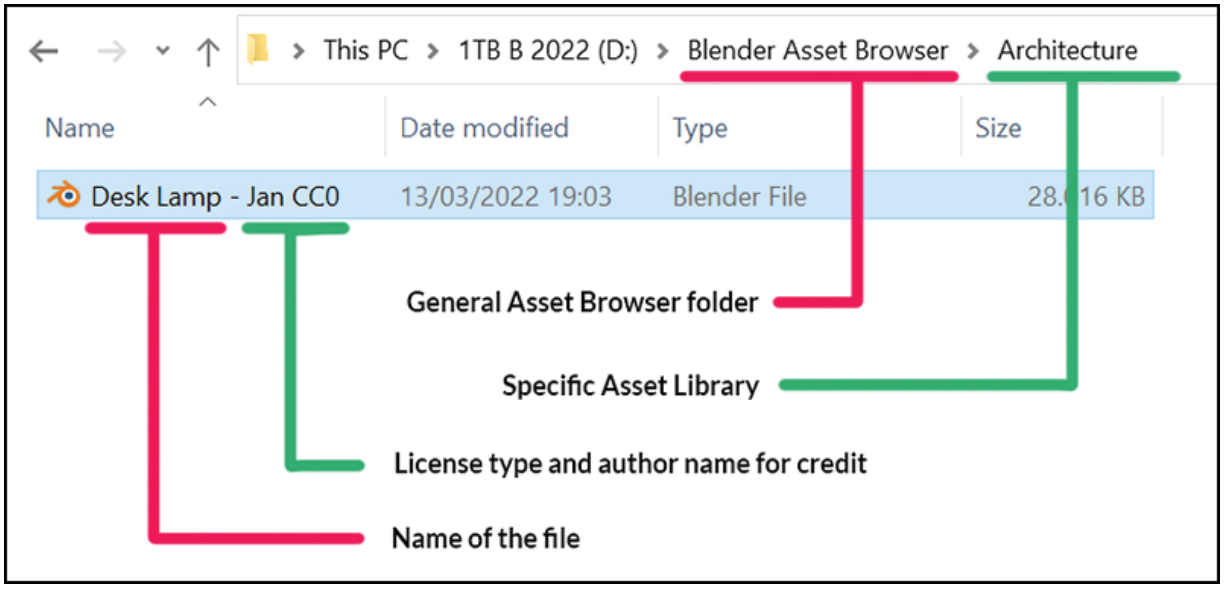


Drag and Drop with the Asset Browser

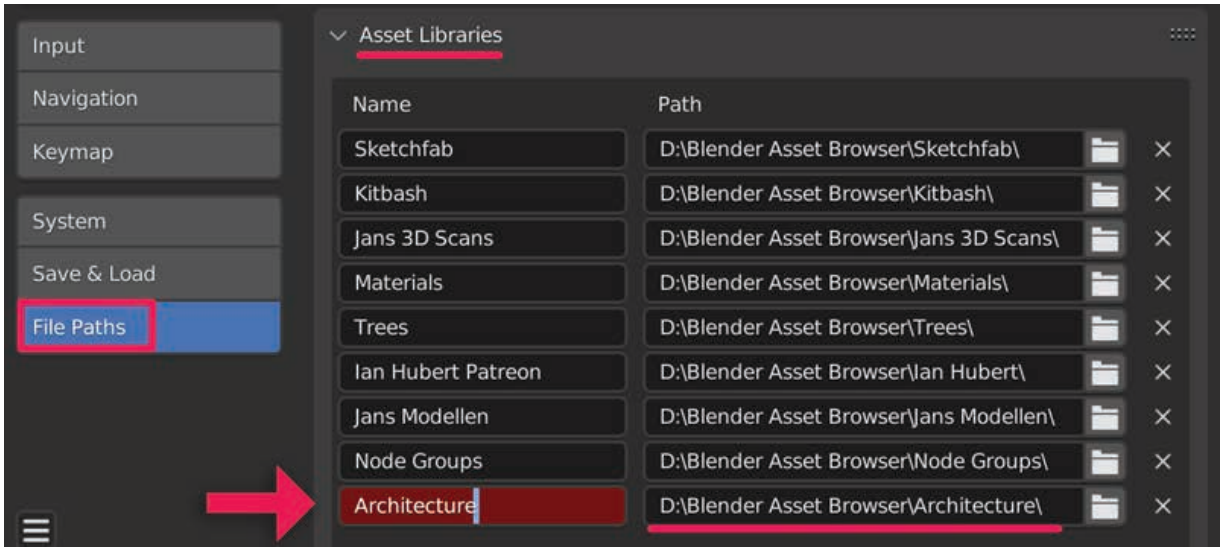
The Asset Browser is now (since Blender 3.1) a default part of Blender since the official release of Blender 3.0. To add an object to the Asset Browser, **save the .blend-file** containing that object in a folder that has a name pertinent to that object. For example, a desk lamp model could go in a folder called "**Architecture**".



I tend to put the author and the licence type in the name of the file. For example, "Jan CC0" means the author is named **Jan**, and the licence type is **CC0** (more info about licence types here). That way I know how to credit the creator of the file later and what I can use it for.

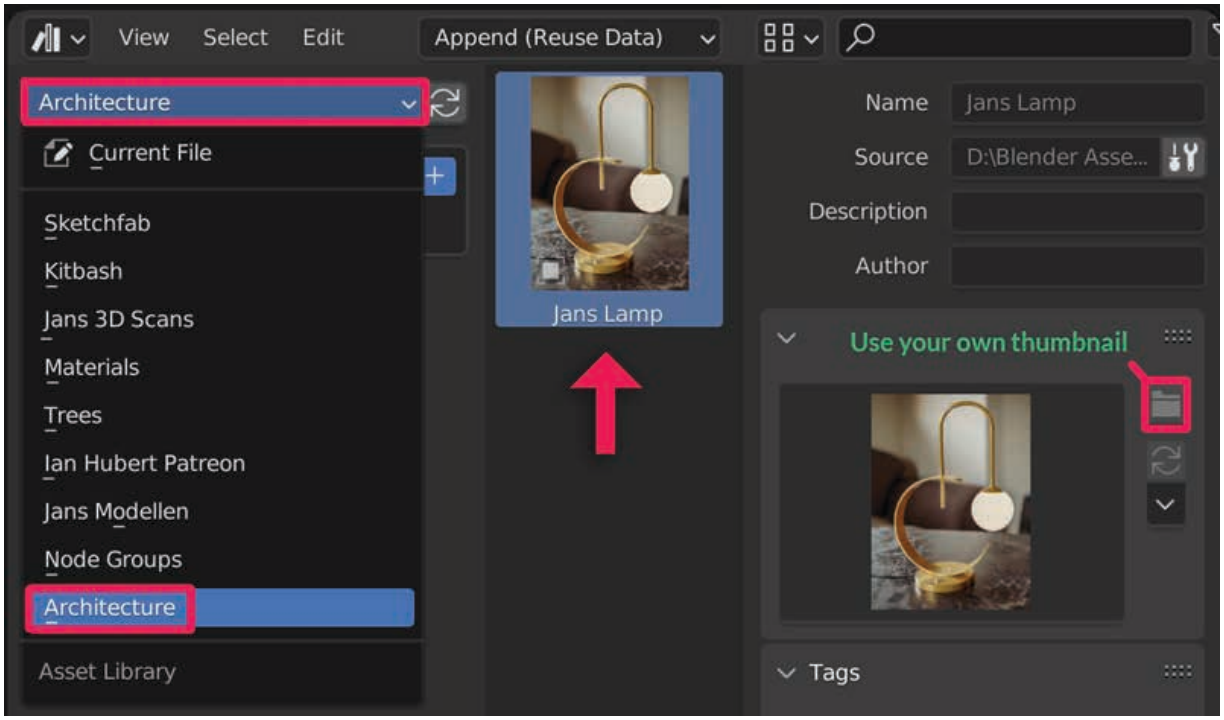


If you prefer, you can also store this information in the asset, in the **Description** and **Author** fields in the Option Panel of the Asset Browser. Then add this folder in **Preferences** under **File Paths > Asset Libraries** and give it a name in the field on the left of the file path. I usually just use the same name as the folder.



Behold… my stuff!

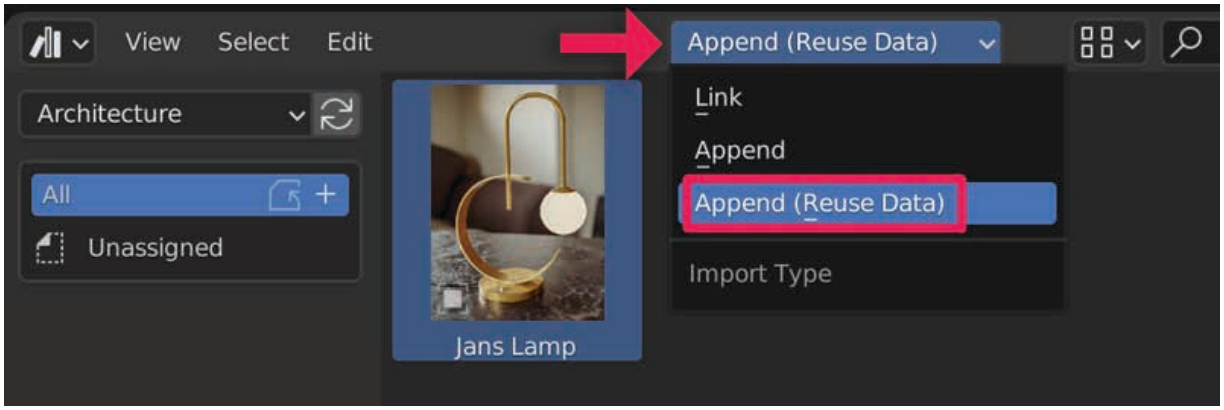
From then on, this is an **Asset Library** that is available by default in the Asset Browser when you open a new Blend file. From there you can drag it to the 3D viewport. You can also load your own thumbnail image if you don't like the one Blender automatically creates.



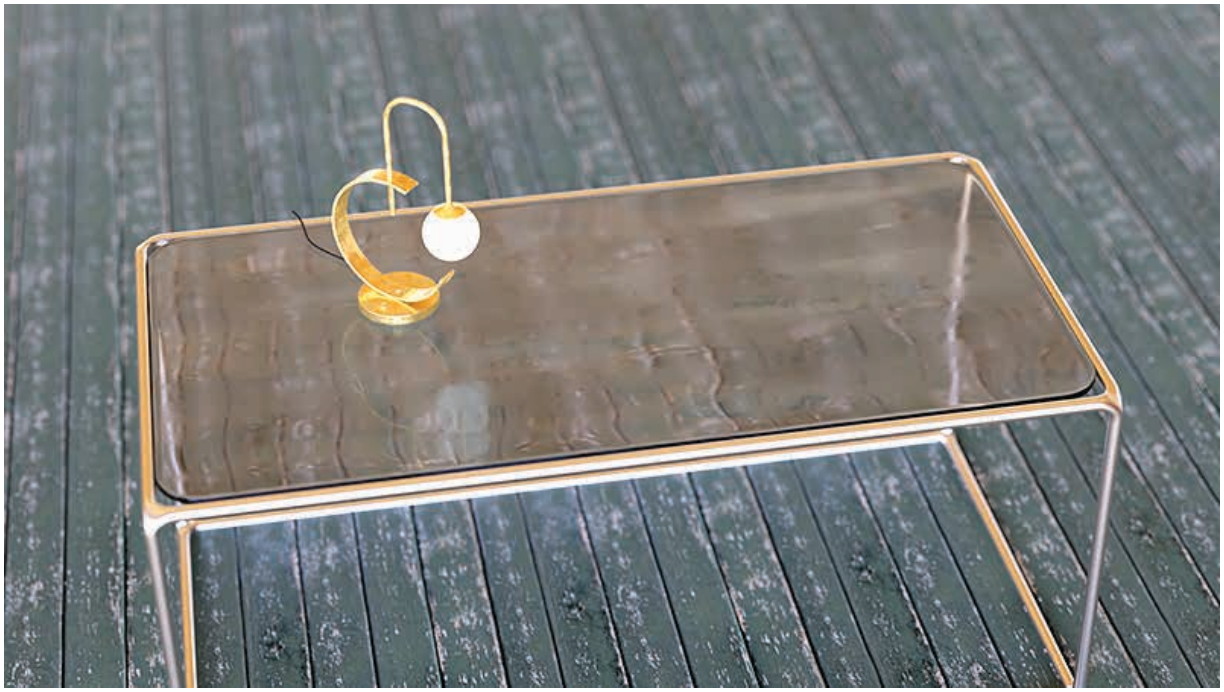
By default, objects from the Asset Browser are **appended** with the **Reuse Data** option. This means that they are really appended to your project, so changing the object in the original file has no influence on that new copy.

But if you drag it into the 3D viewport again, that will be a linked duplicate of the one you already have in the scene. So if you change one of the copies in the scene, they will all change. This is a very memory-efficient method of working.

If you set it to **Append**, each copy will be a unique data block, which after a while will cost a lot of memory. If you set it to **Link**, it will only link to the original file. So if you want to change something about it, you need to do that in the original file. This is also very memory-efficient, but can be impractical as well. My recommendation is to use the default, **Append (Reuse Data)**.



Now you can drag and drop that desk lamp very easily!



Use Collections and parented objects in the Asset Browser

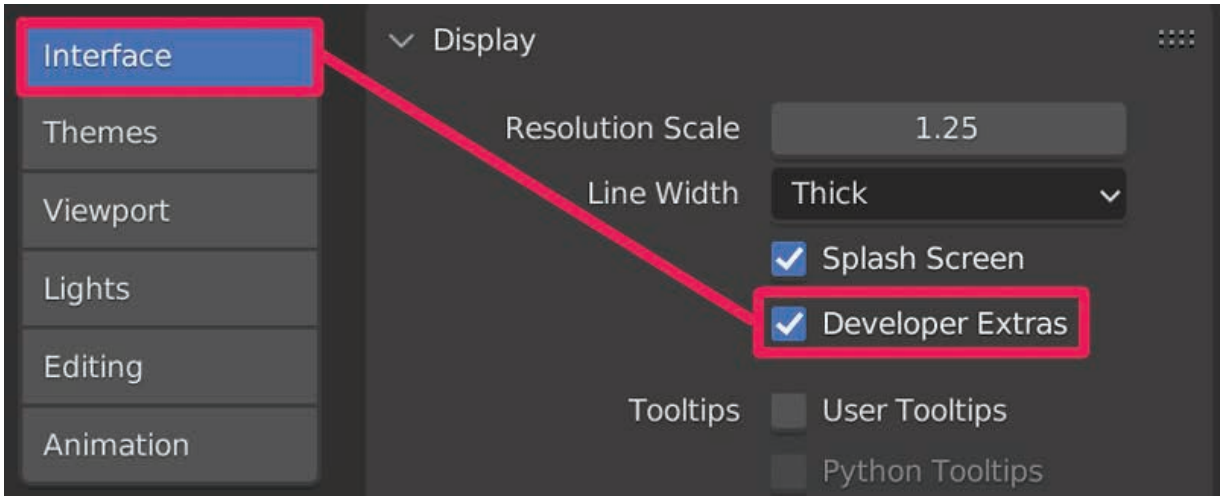
If you've been trying out the Asset Browser, you may have discovered that **Collections** or simply **parented objects** such as an Armature and a Character, or just a number of objects parented to an Empty, can't be stored as a single Asset. However, it's certainly possible! Here are two methods that you can use right now.

This sofa seat consists of several separate objects in a Collection.

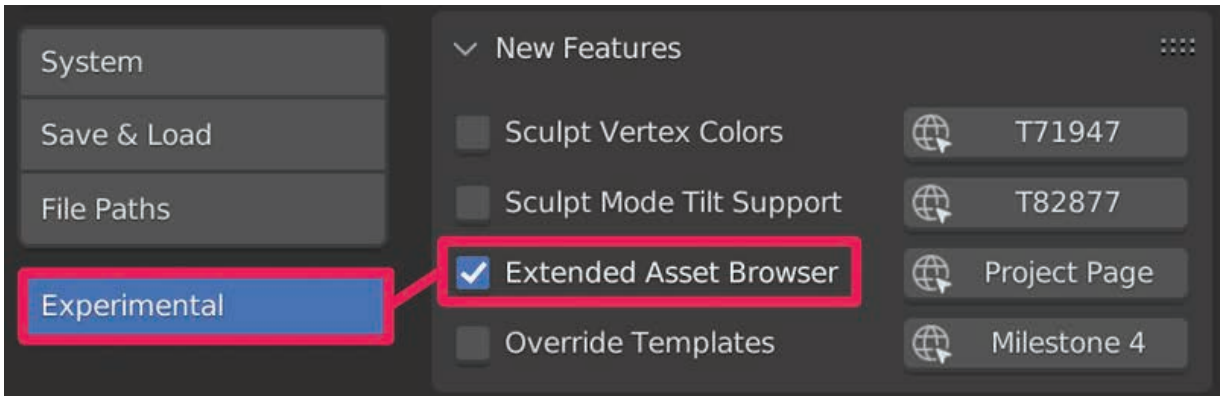
If you've been trying out the Asset Browser, you may have discovered that **Collections** or simply **parented objects** such as an Armature and a Character, or just a number of objects parented to an Empty, can't be stored as a single Asset. However, it's certainly possible! Here are two methods that you can use right now.

Method 1: Extended Asset Browser.

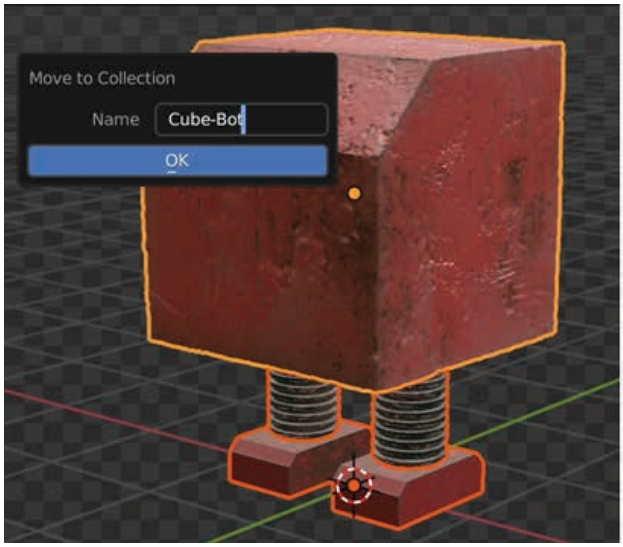
The first method uses **Blender 3.2 Alpha**. You can download it here. In Preferences > Interface enable "**Developer Extras**".

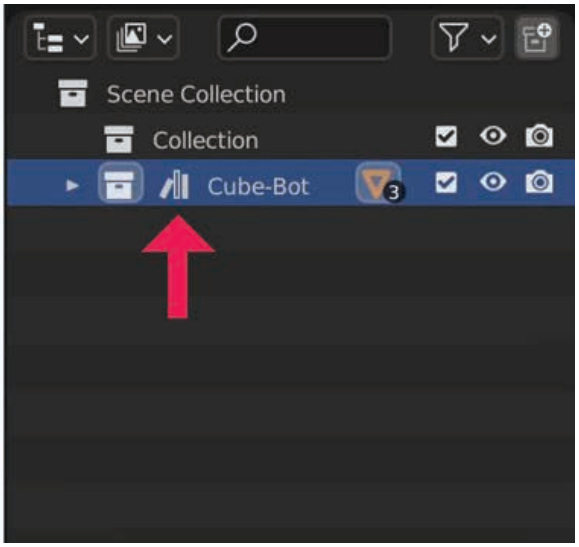


Now the Experimental tab appears. In it, enable "**Extended Asset Browser**".



Now you can add and use Collections in the Asset Browser. Add things to a Collection by selecting them and **pressing M**, then in the Outliner right-click on the selected Collection and choose **Mark as Asset**.

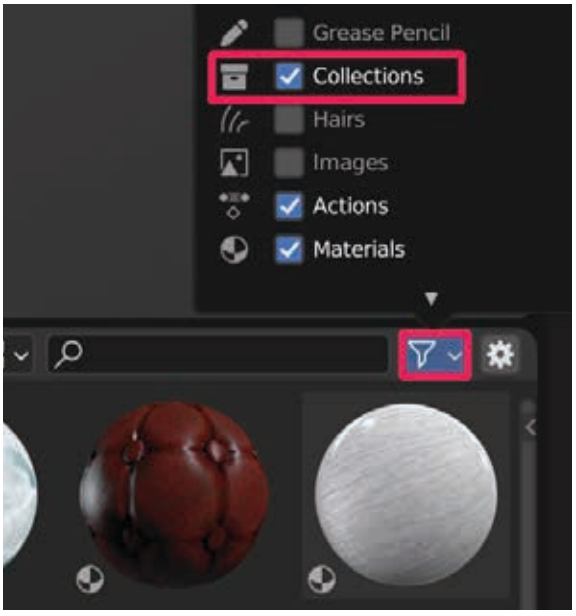




Then save the .blend-file in a folder listed in File Paths under Asset Libraries as usual with assets.

To see Collections in the Asset Browser, check the **Collections** box under **Filter**.

I did notice that this checkbox gets unchecked every time you close Blender, which will hopefully be fixed in a future release when this feature becomes standard.

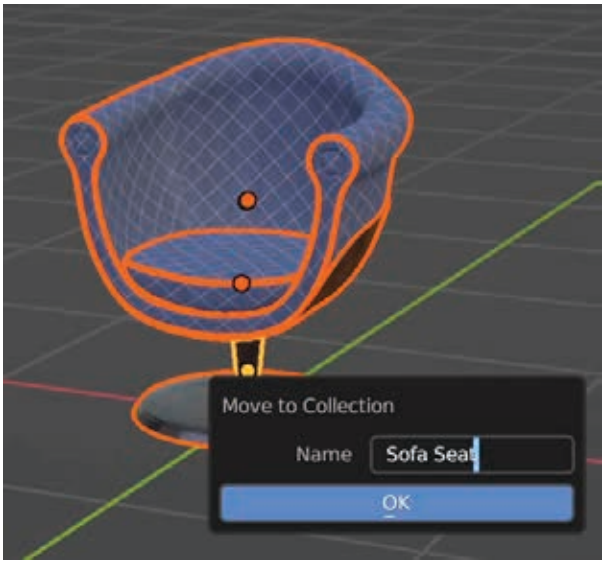


Method 2: Workaround in Blender 3.1 Stable

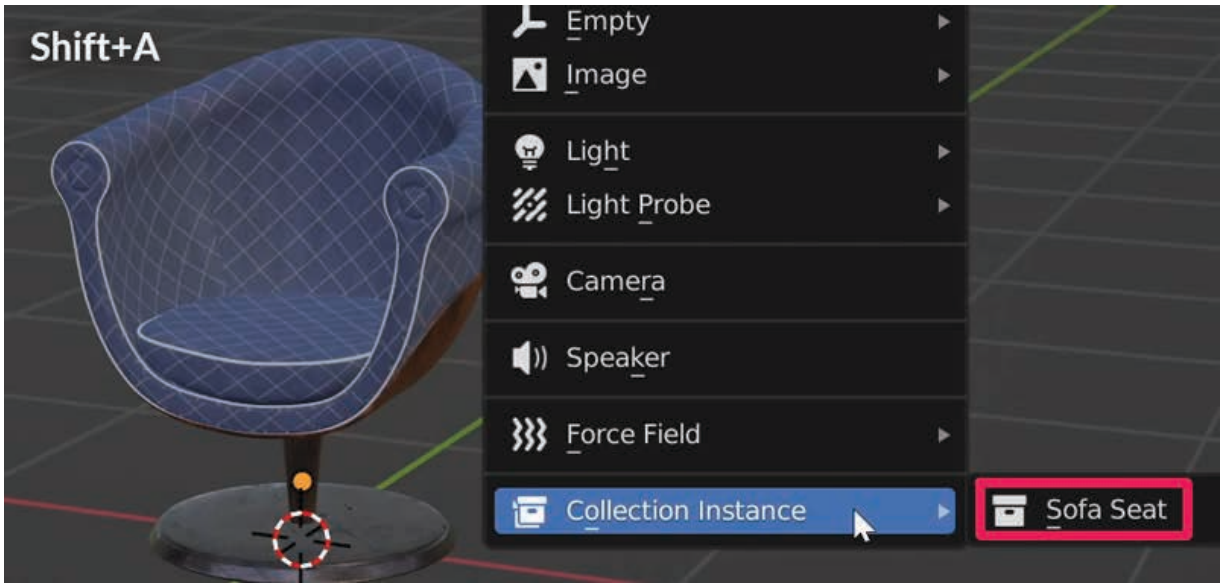
If you don't want to use an experimental build of Blender, you can use the following workaround in the stable release of Blender 3.1.

Credit for discovering this workaround goes to Brandon's Drawings on Youtube. Select the objects you want to bundle as one asset and **press M**.

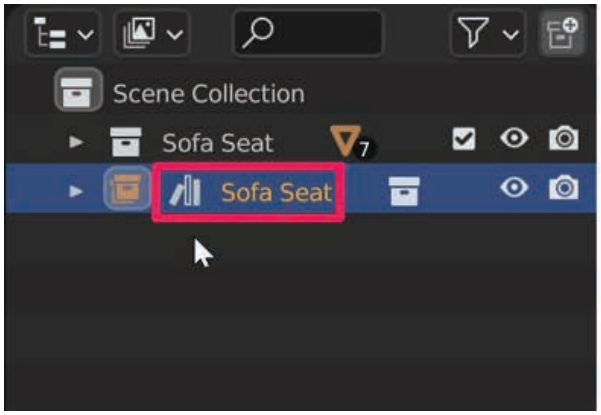
Add them to a collection.

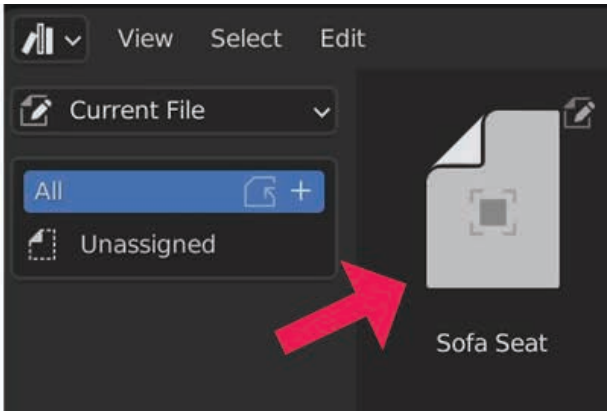


Press Shift+A in the 3D viewport for the Add Menu and add a **Collection Instance** of that Collection.

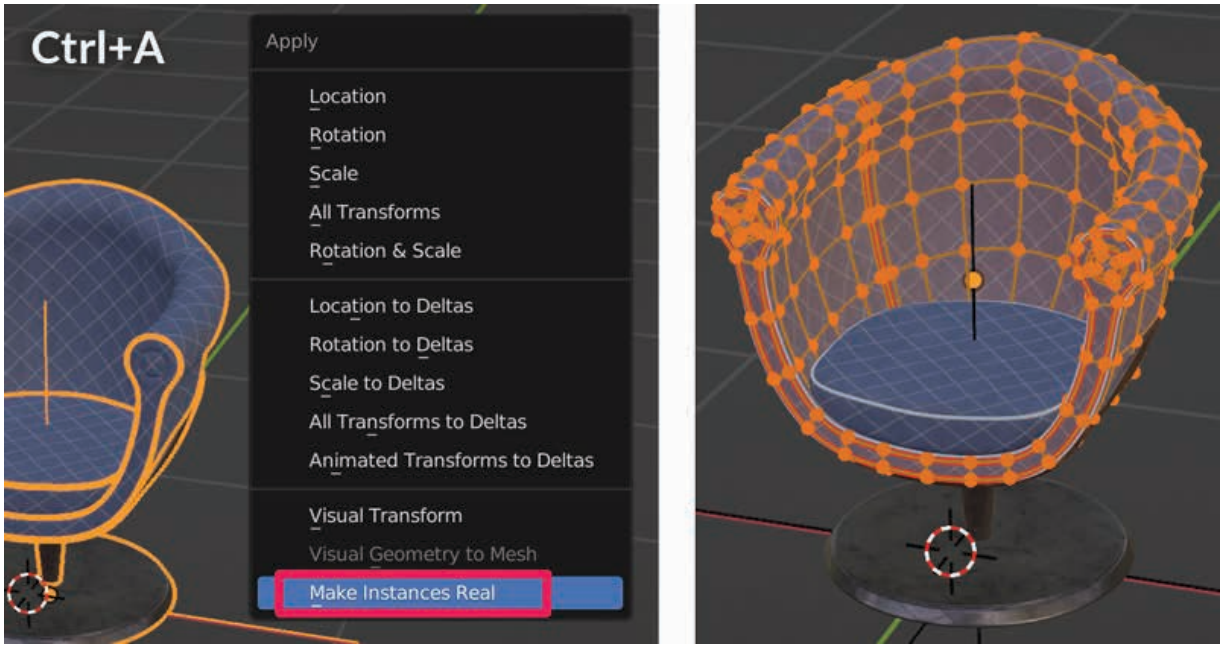


Right-click on the Collection Instance in the **Outliner** and choose Mark as Asset. Now you will see it appear in the Current File selection of the Asset Browser. It's that simple!





With both methods, you'll most likely need to use a custom thumbnail. If you want to edit the instance, **press Ctrl+A** in the 3D viewport and choose **Make Instances Real**.

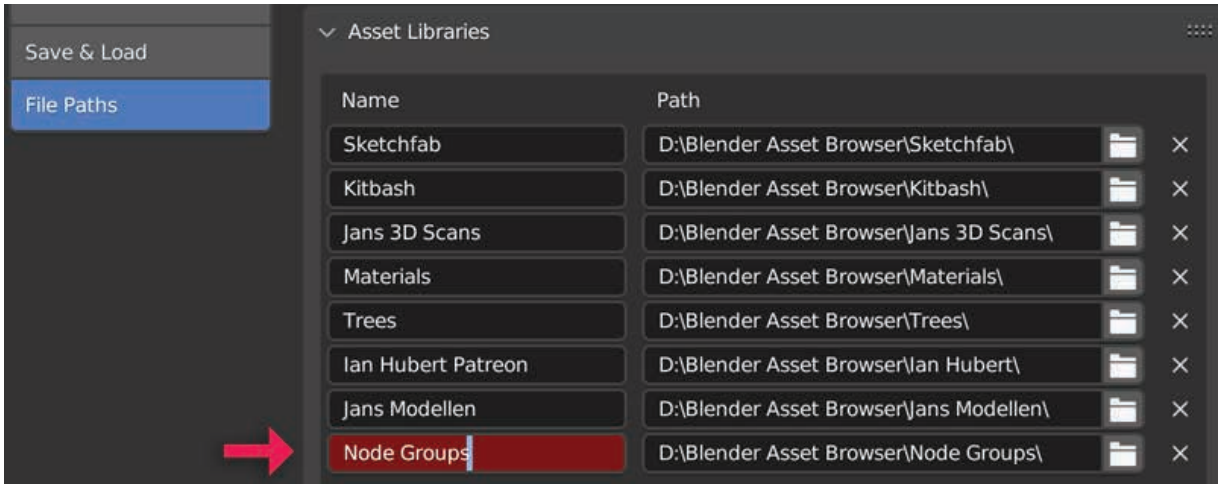


As always with Asset blend files, enable File > External Resources > **Automatically Pack Resources** if the asset contains textures. And save the blend file in a folder listed under Asset Library File Paths in Preferences.

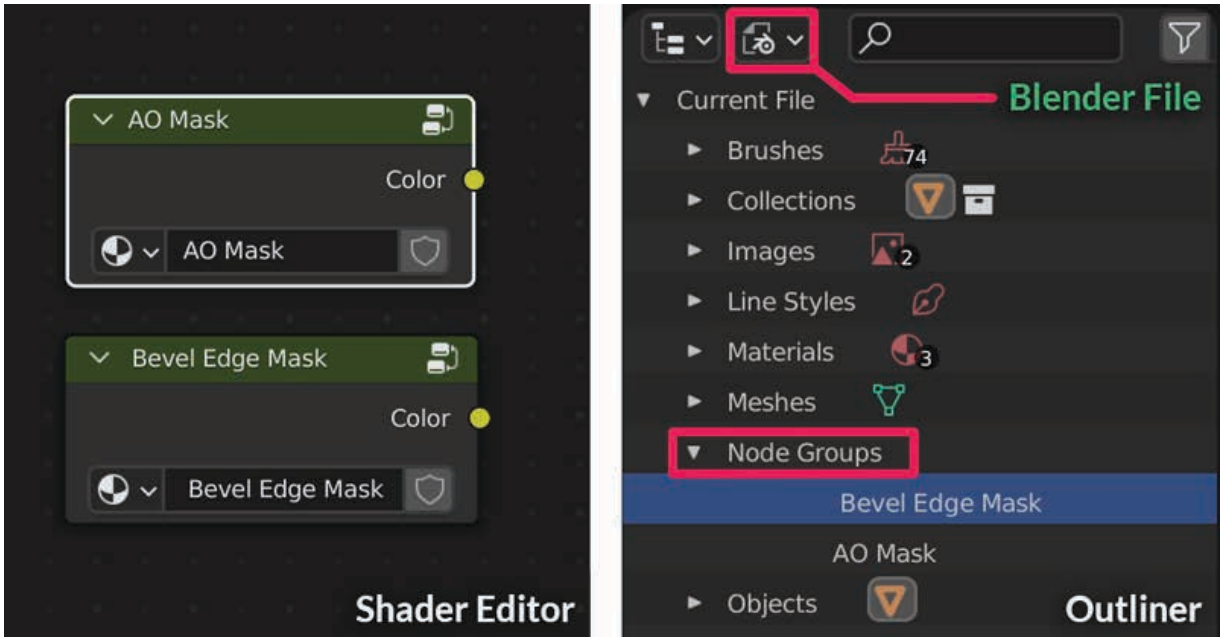
Asset Browser: How to mark Node Groups as Assets

Node Groups are great to have at your finger tips in the Asset Browser, but when you right-click on them in the Shader Editor, "Mark as Asset" isn't an option. However, you can definitely save them as Assets by following the steps outlined below.

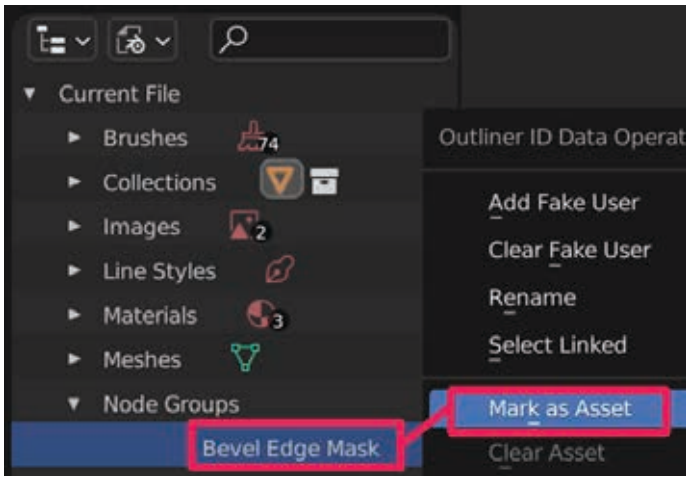
Save your file containing the Node Groups in a subfolder of the folder where you keep your Asset Browser assets. Then add that path in Preferences.



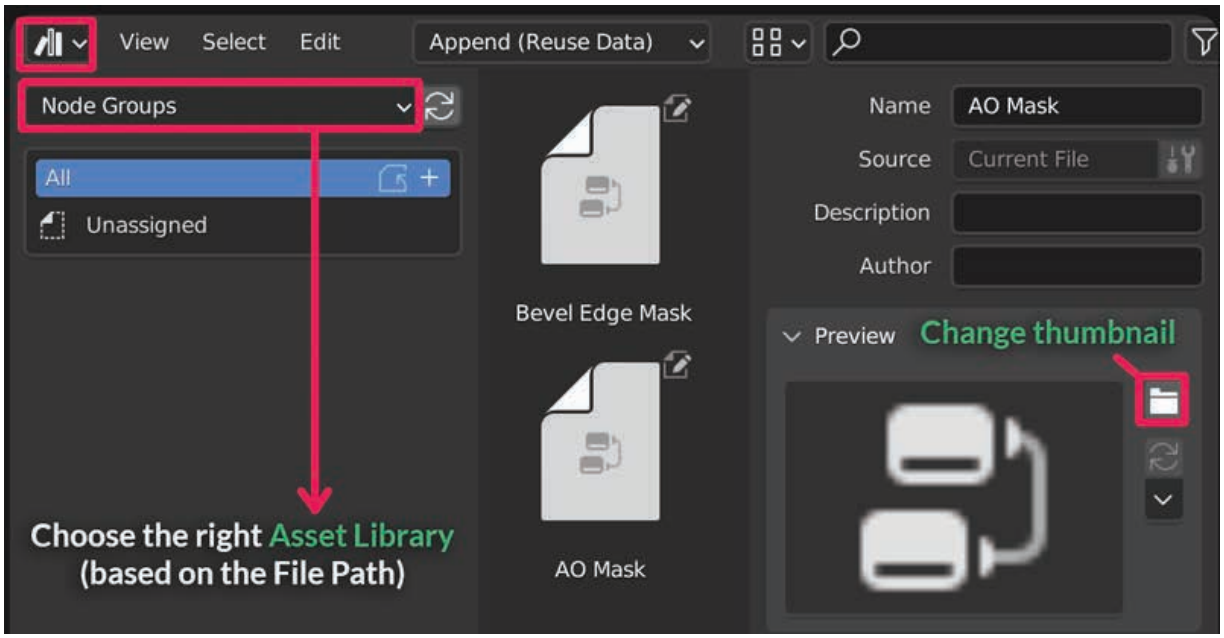
You can't mark Node Groups as assets in the Shader Editor. But you can in the Outliner. To do so, set the Display Mode to "Blender File" (see picture below right). Then, expand the Node Groups to find your Node Groups.



There, right-click on a Node Group and choose **Mark as Asset**.

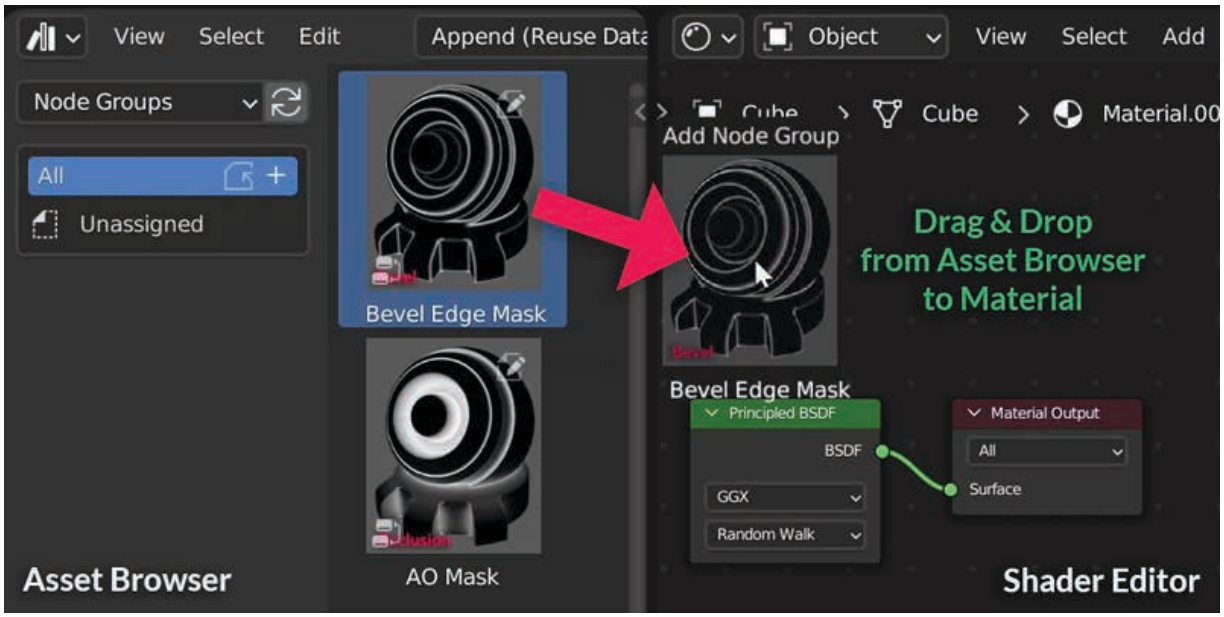


Having marked the Node Groups as Assets, as long as this file is saved in a folder named "Node Groups" which is also listed in the File Paths in Preferences, you'll find the Node Groups in the Asset Browser in the right Asset Library.



Click here to return to the index.

Since Node Groups don't automatically get an interesting thumbnail, you can open the Option Panel by **pressing N** in the Asset Browser, select a Node Group and click on the file icon of the Preview (see above). Then you can choose an image you've created yourself as the thumbnail. Then, just **drag & drop** Node Groups as needed to materials in the Shader Editor.

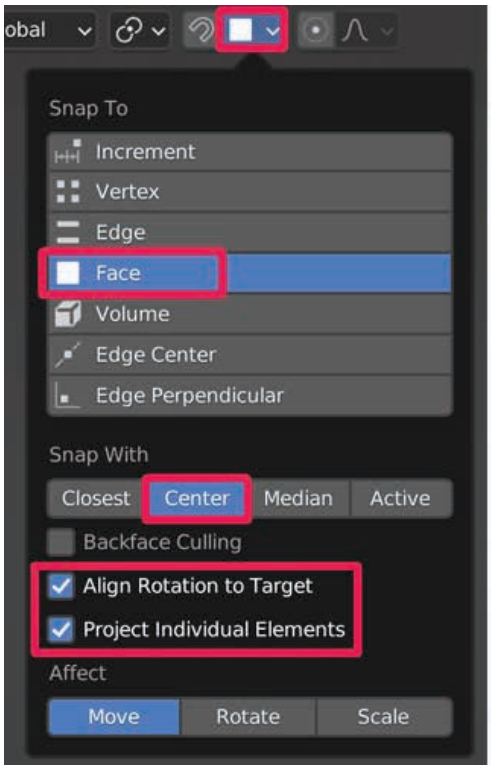


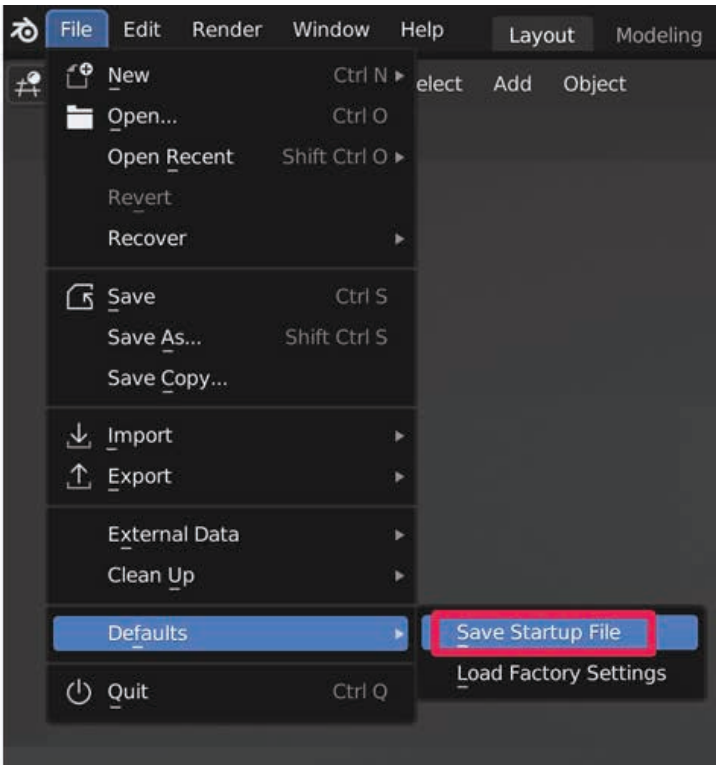
Now you never have to create or append those Node Groups again.

Drag-and-drop Decals from the Asset Browser

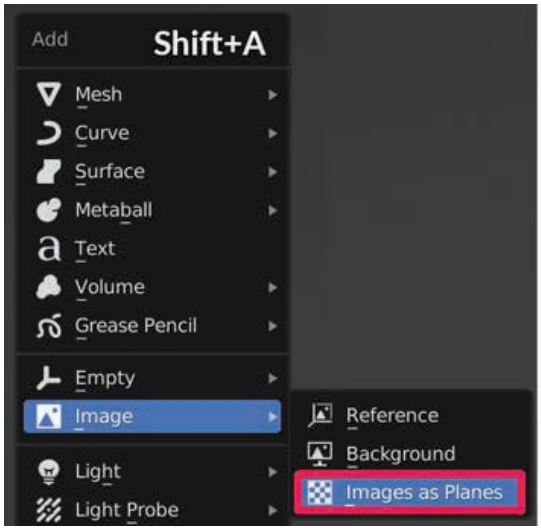
Using the Asset Browser, we can now easily drag and drop decals from the Asset Browser to objects in the viewport. First, create (or buy) some PNG images with alpha transparency. Make sure the **Images as Planes add-on** that comes with Blender is enabled.

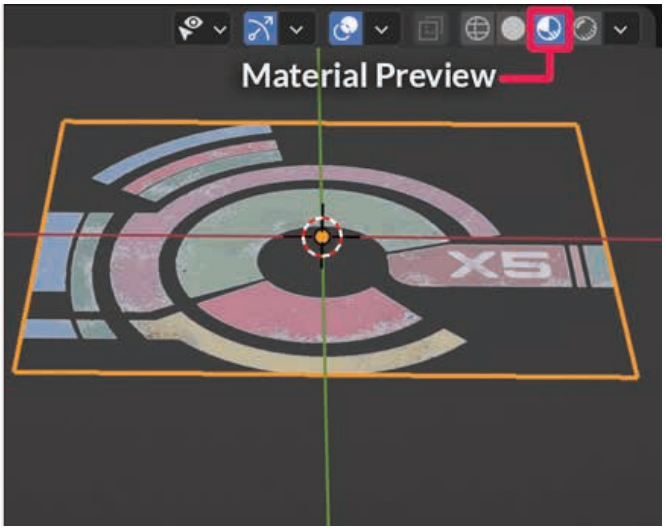
Enable the Snapping options: to **Face**, **Snap With Center**, and **Align Rotation to Target**. Also enable **Project Individual Elements**. Leave snapping turned off, you can enable it when needed by holding Ctrl. If you want to make this your default settings (like I do, and recommend you do too), go to **Defaults > Save Startup File**. Now you never have to choose those settings again.



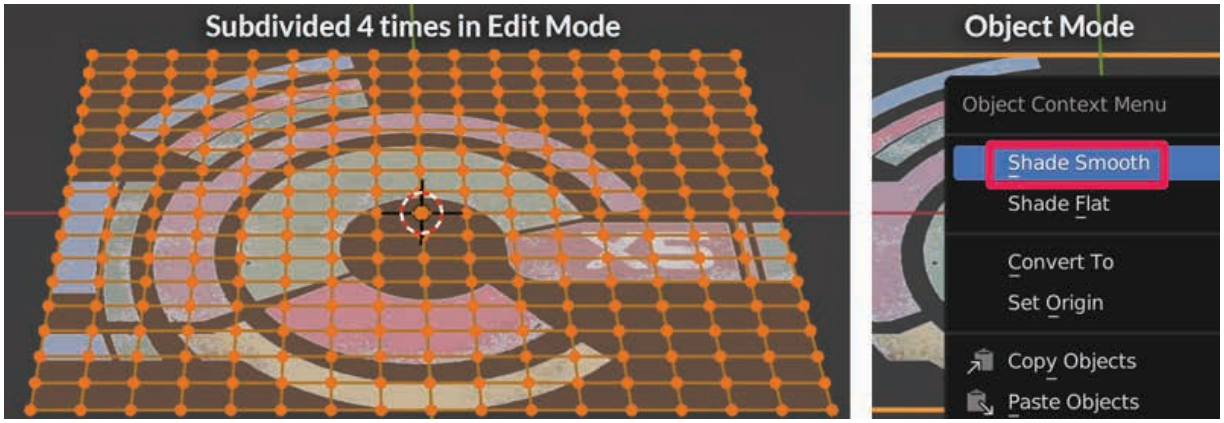


Enable **Material Preview** mode so you can see textures in the viewport. **Press Shift+A > Image as Planes** and load one of the PNG images.

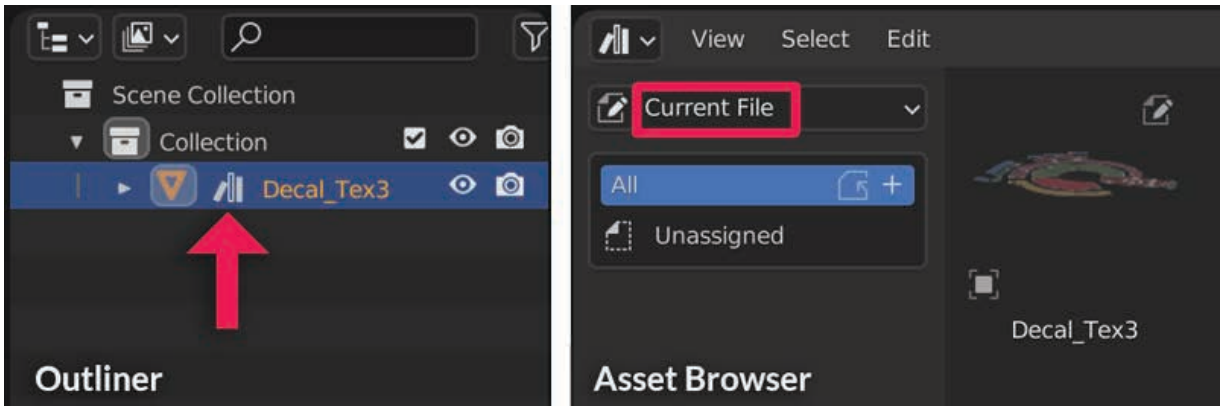




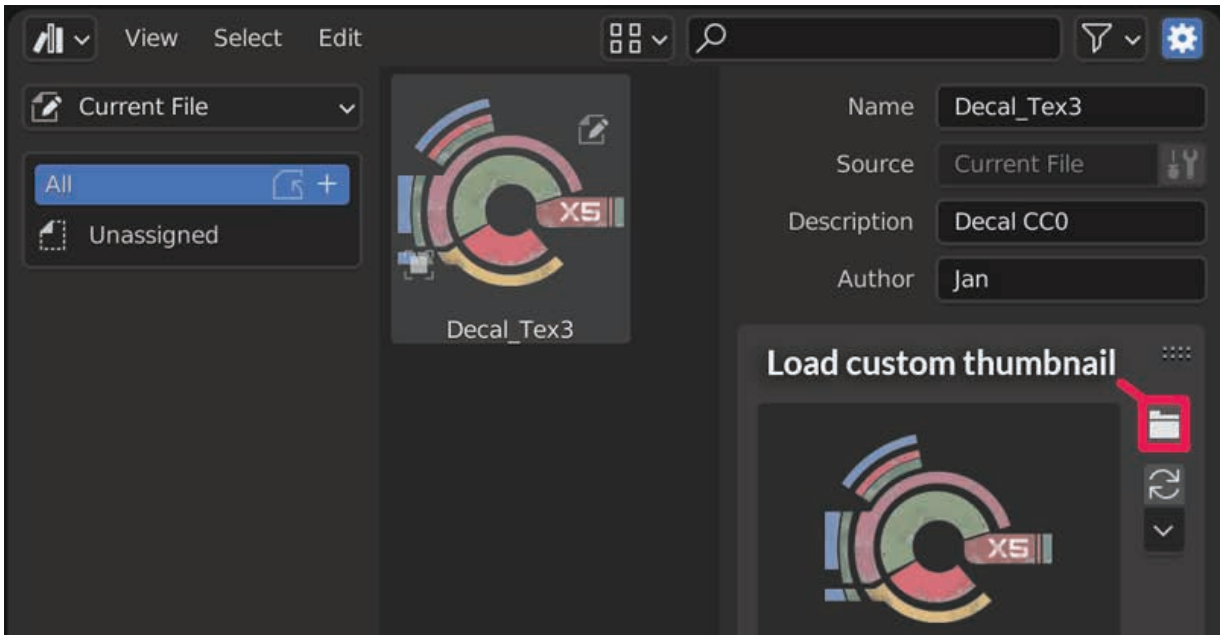
In **Edit Mode**, right-click and choose **Subdivide**. Repeat by **pressing Shift+R** a couple of times (4 times in this example). This gives the decal enough geometry to wrap around surfaces. Set it to **Shade Smooth** to make it even smoother.



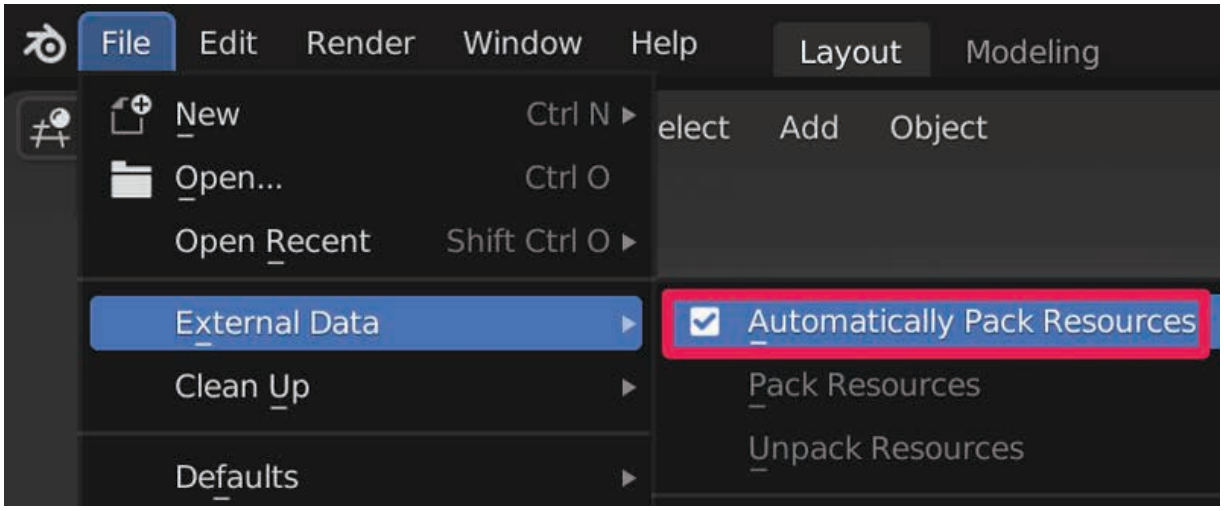
Right-click on the plane in the **Outliner** and mark it as an **Asset**. Now you can find it in the "**Current File**" library in the Asset Browser.



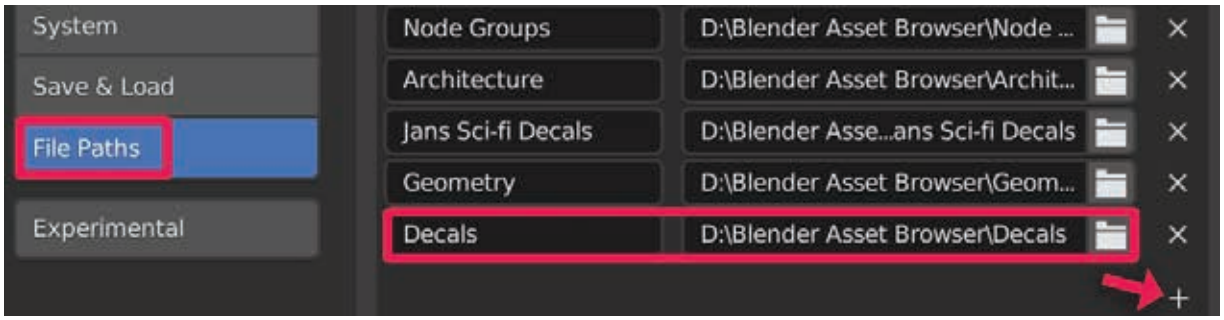
Although thumbnails are generated automatically for objects in the Asset Browser, since we already have the PNG image that we used for the texture, we can load that as the **custom thumbnail**. It will be more fun to use this decal in the future when you can see clearly what it looks like.



Load some more mages as planes and subdivide them in Edit Mode. Mark them as assets and give them custom thumbnails, as shown before. Don't forget to **Pack Resources** so you don't lose the textures.

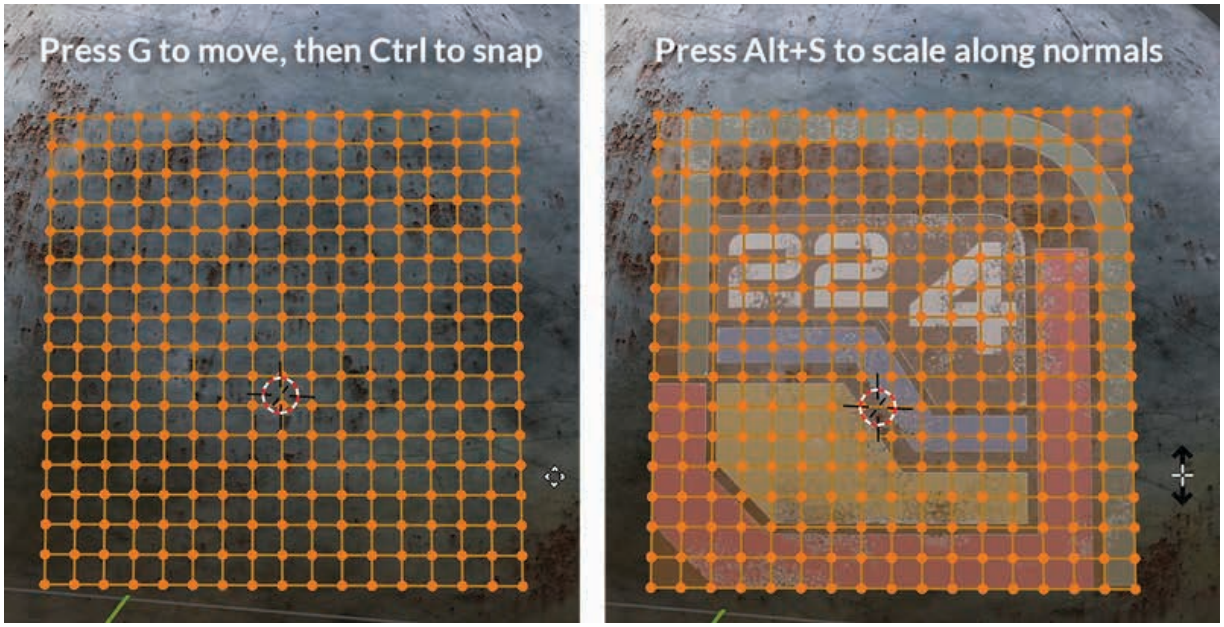


Then save this .blend-file in a folder called "Decals" (or something which is clear to you) and make sure that folder is listed in the Asset Library **File Paths** section in Preferences. Now when you load a new file, the Decal assets should be there in their own library.

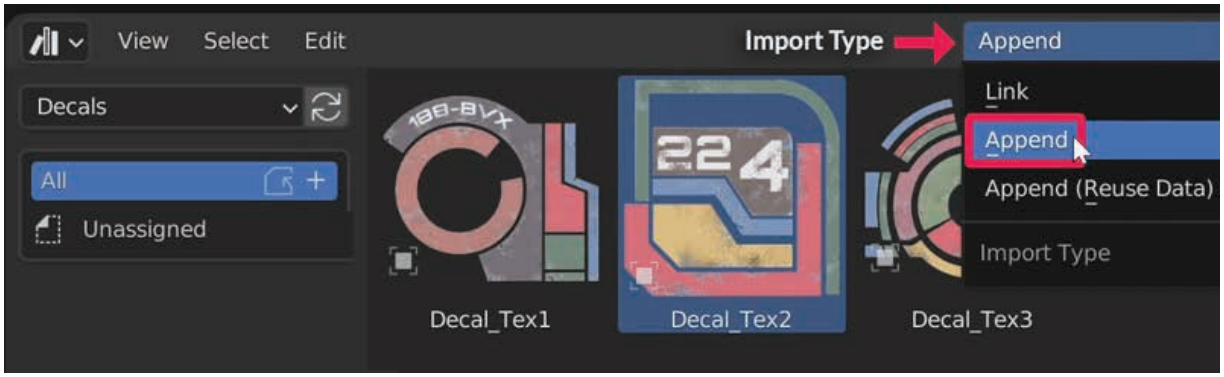


Now you can drag a decal from the Asset Browser, and move it along the surface where you want to place it. **Press S** to scale it, and **press R and Z twice** to rotate it along its normal.

Finally in Edit Mode **press G, then hold Ctrl and left-click** to snap the individual vertices to the surface. It will look like the decal disappears, because the vertices are at the same location as the surface. You can fix that by **pressing Alt+S**, scaling the decal up slightly along its normals until it's just above the surface.

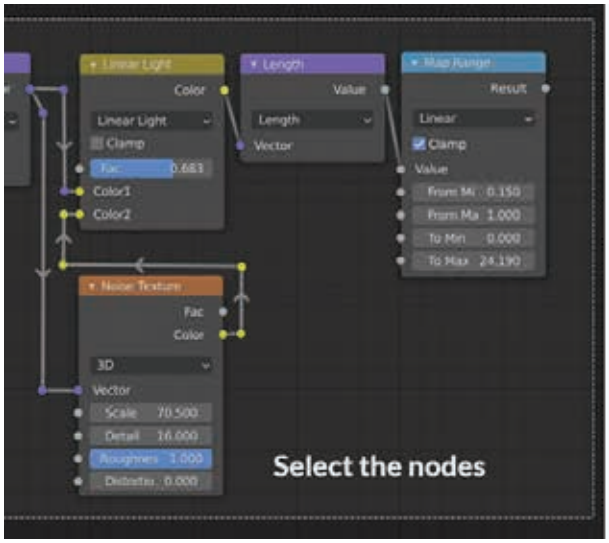


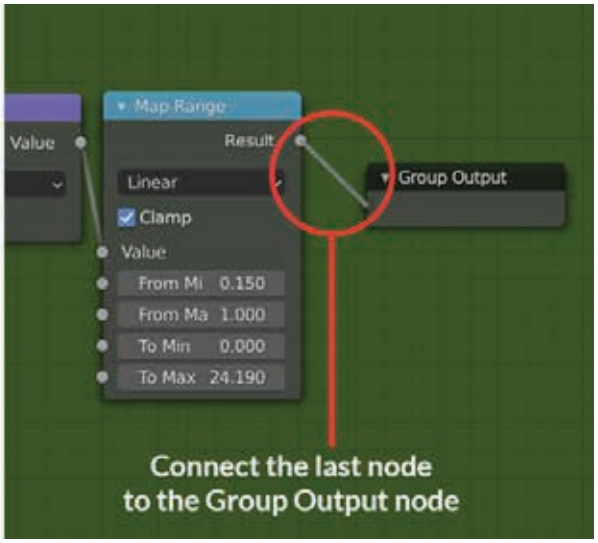
If you want to use a decal more than once in a scene, choose the **Append** Import Type and not **Append (Reuse Data)**, otherwise changing one in Edit Mode will also change the others, which you don't want in this case.



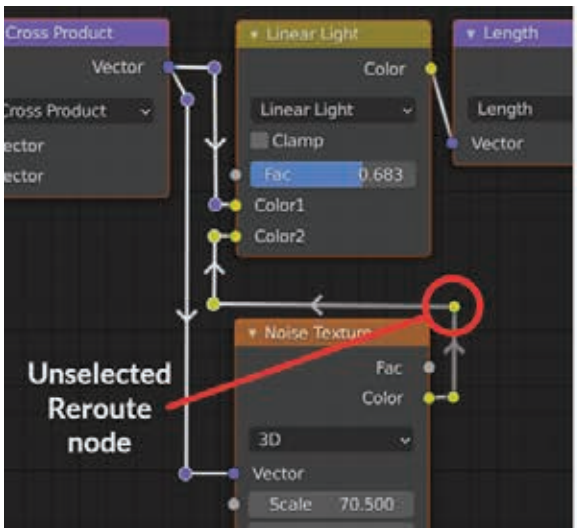
Node Groups

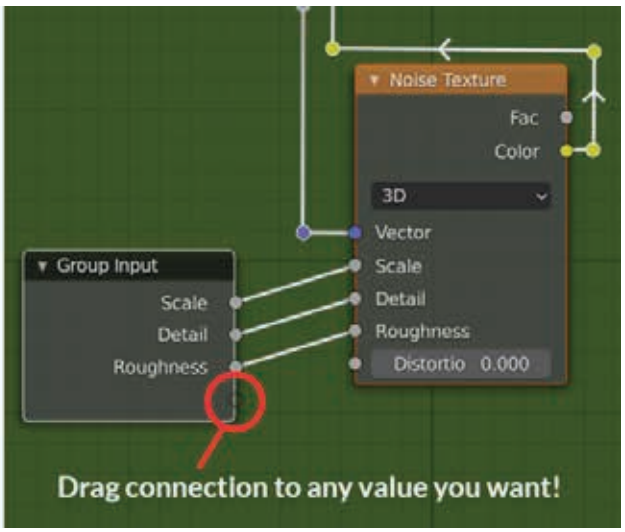
To keep complex node trees organised you can use Node Groups. To create a group, select nodes by dragging a box around them and then **press Ctrl+G**. You'll see that the nodes are now against a green background. You're now "inside" the Node Group. Connect the last node to the Group Output.





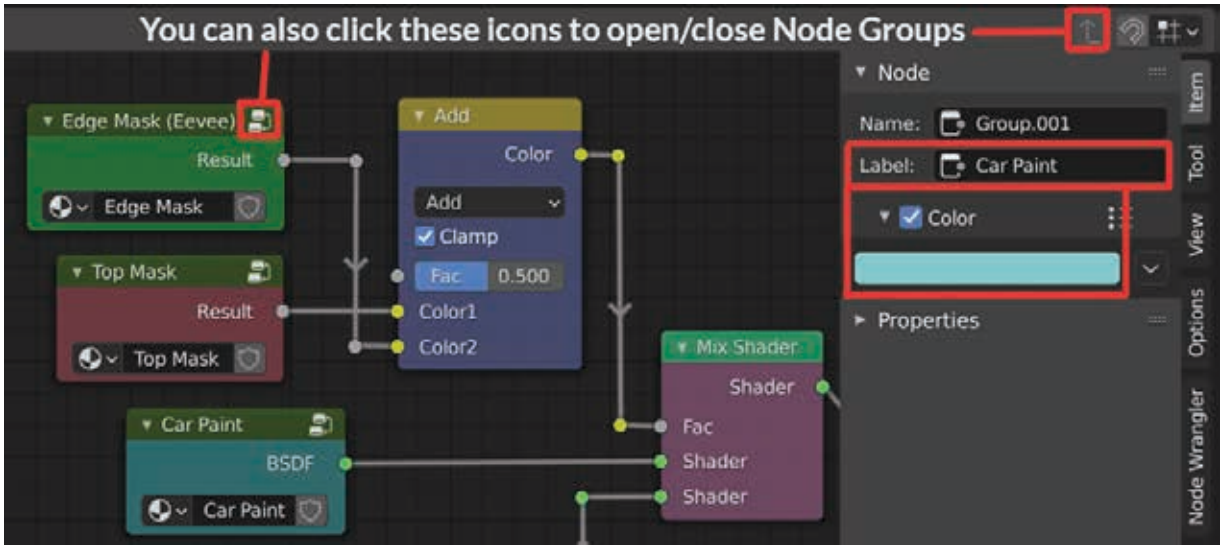
If nothing happens when pressing Ctrl+G, make sure everything is selected, especially when using Reroute Nodes. To control specific values of the node group, connect the Group Input node to whatever value you want to control. You can connect several values.





With the node group selected, **press TAB to expand or collapse it**.

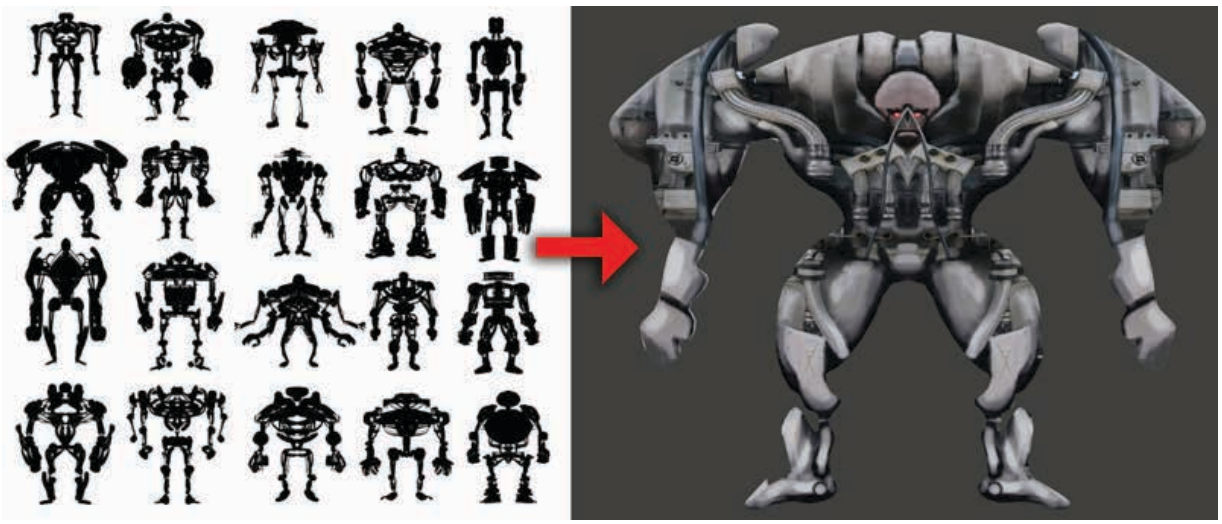
You can give nodes a label and color, so you can do this with Node Groups as well. Instead of pressing TAB you can also click on icons as demonstrated in the image below, to open/close (expand/collapse to use the proper terms) Node Groups.



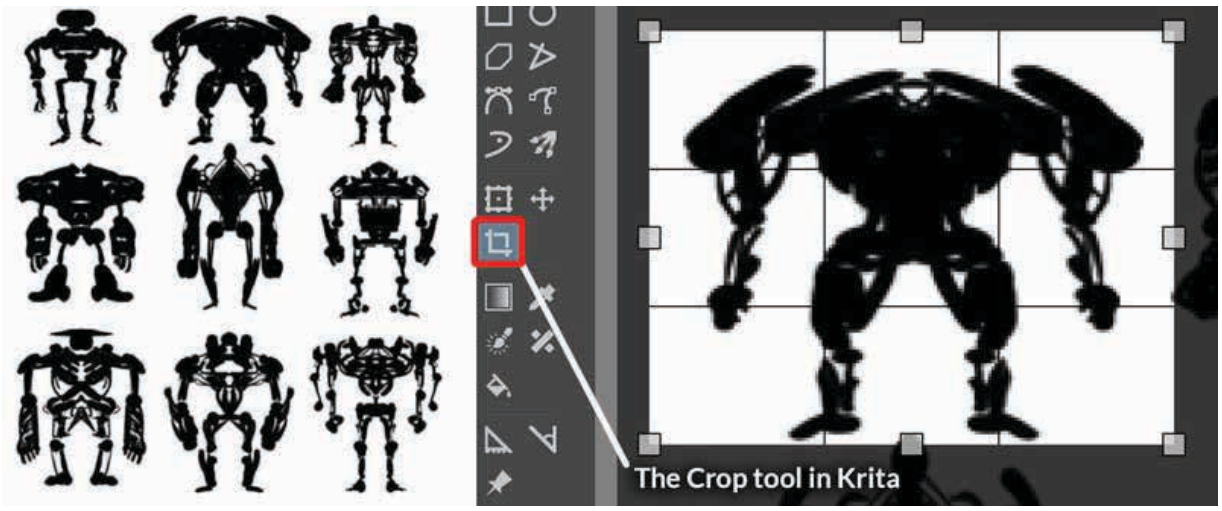
Click here to return to the index.

Concept Art Workflow - Trace image to Grease Pencil

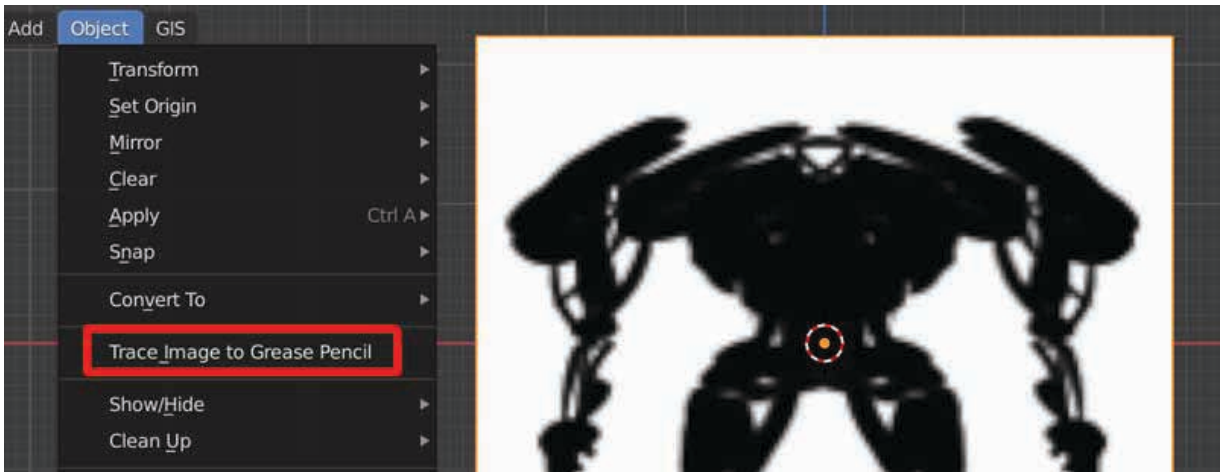
Although the painting tools in Blender are limited, there are also some advantages in Blender that 2D painting apps like Photoshop don't have, such as easily mirroring textures. This, in combination with the recent "Trace Image to Grease Pencil" function makes Blender a powerful tool for painting concept art.



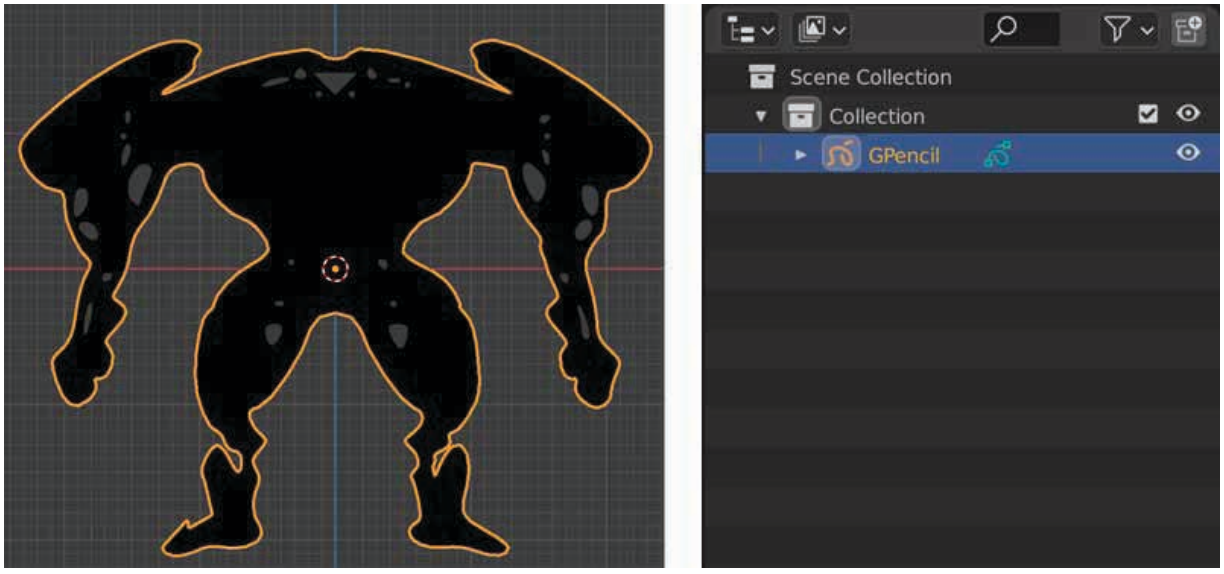
After you've made some quick thumbnails in your favorite painting app save the one you like best as an image. Here I've used Krita, but it doesn't really matter - you could also just use pen on paper and take a picture of it with your phone. As long as it's a simple black and white silhouette.



Now, drag and drop that image into Blender, in the **Front** orthographic view. **Press Alt+G** to center the image in the 3D world. Go to Object > **Trace Image to Grease Pencil**. Then **click OK** in the pop-up menu that follows.

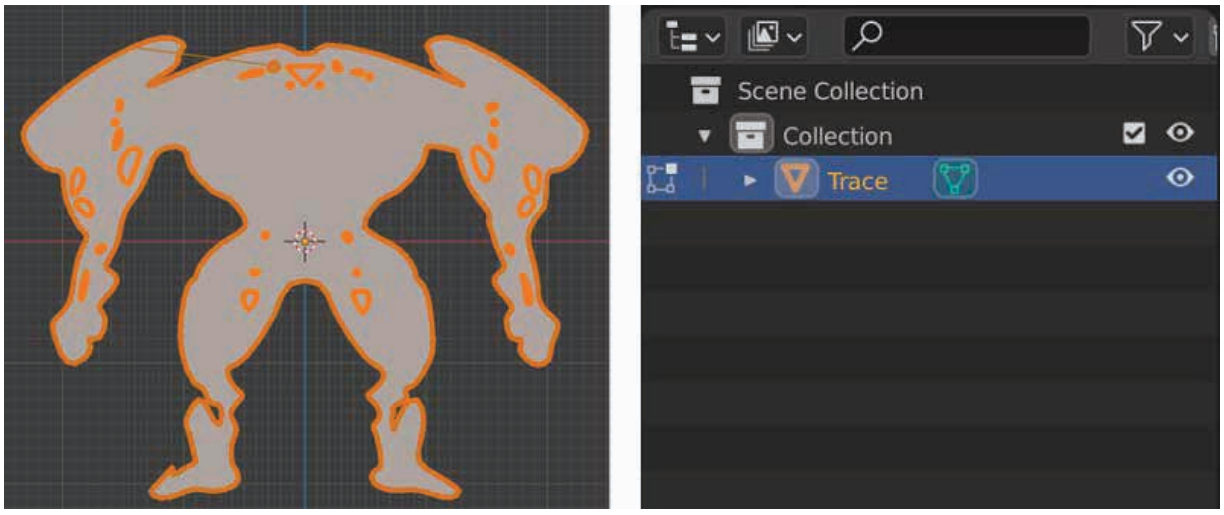


Now your image has been converted to a Grease Pencil object and you can remove the original image (which is listed as an Empty in the Outliner).

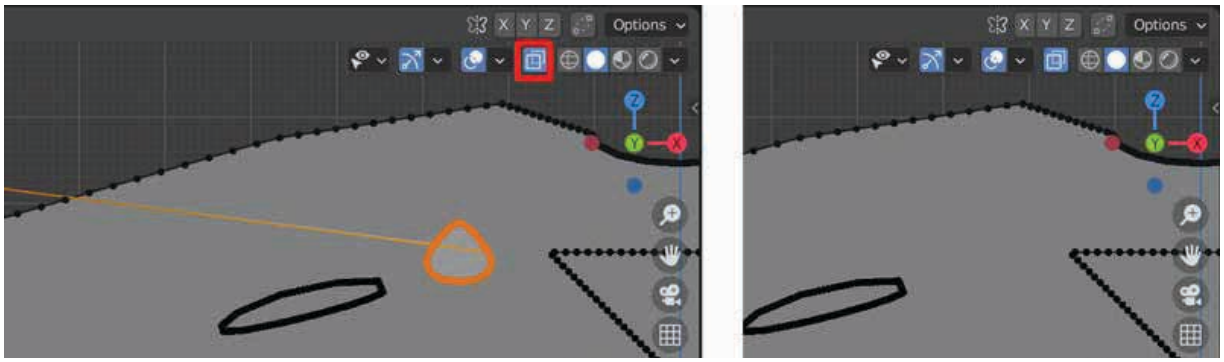


Then convert the Grease Pencil object to a Path, and convert that Path to a Mesh.

You can delete the Grease Pencil object from the Outliner. The only thing left should be the mesh, which will be listed as "Trace" in the Outliner. Fill the mesh by selecting all in Edit Mode and **pressing F**.

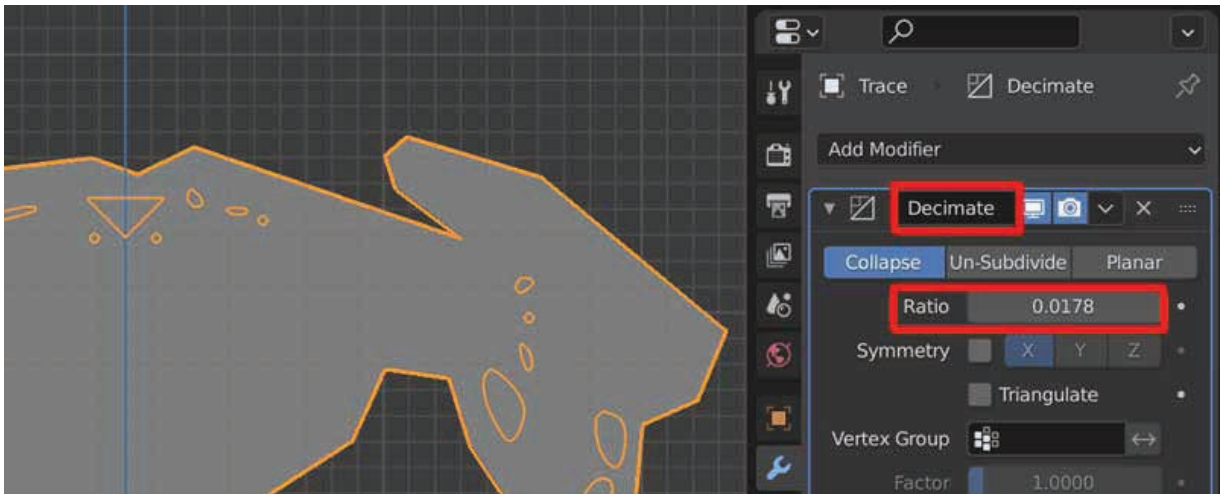


You may have to clean it up a bit by dissolving some problematic vertices. This is easily done by going to **X-ray mode**, then dragging a selection around the vertices, and **pressing Ctrl+X** to dissolve them.

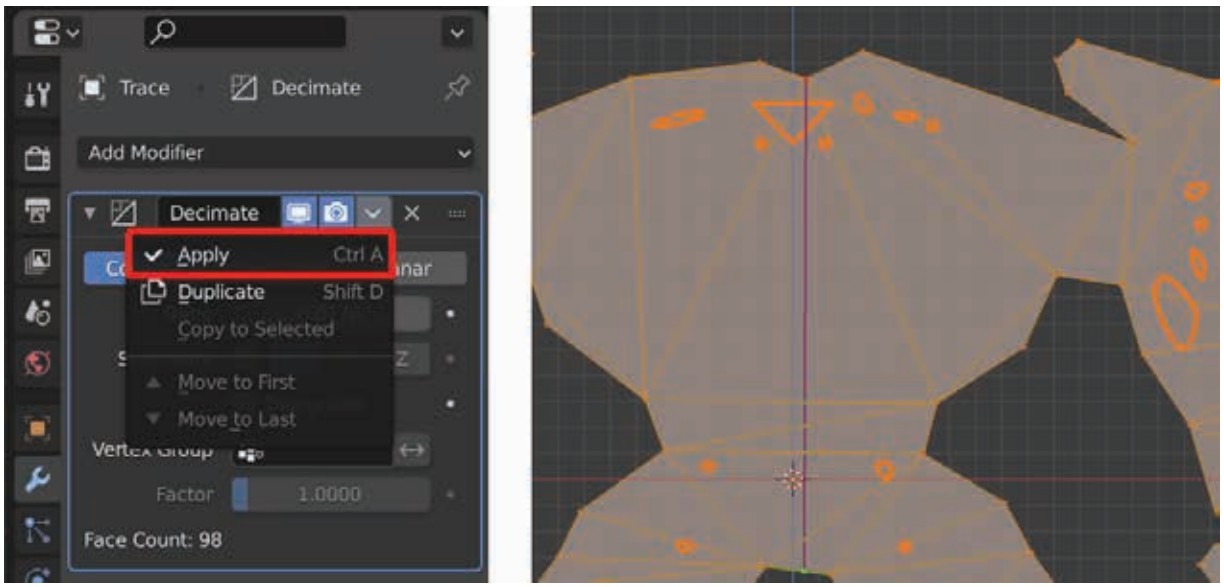


Optionally, you can triangulate the mesh by **pressing Ctrl+T** in Edit Mode and add a **Decimate modifier** to it in Object mode, then reduce the **Ratio** of decimation until the silhouette has somewhat sharper angles.

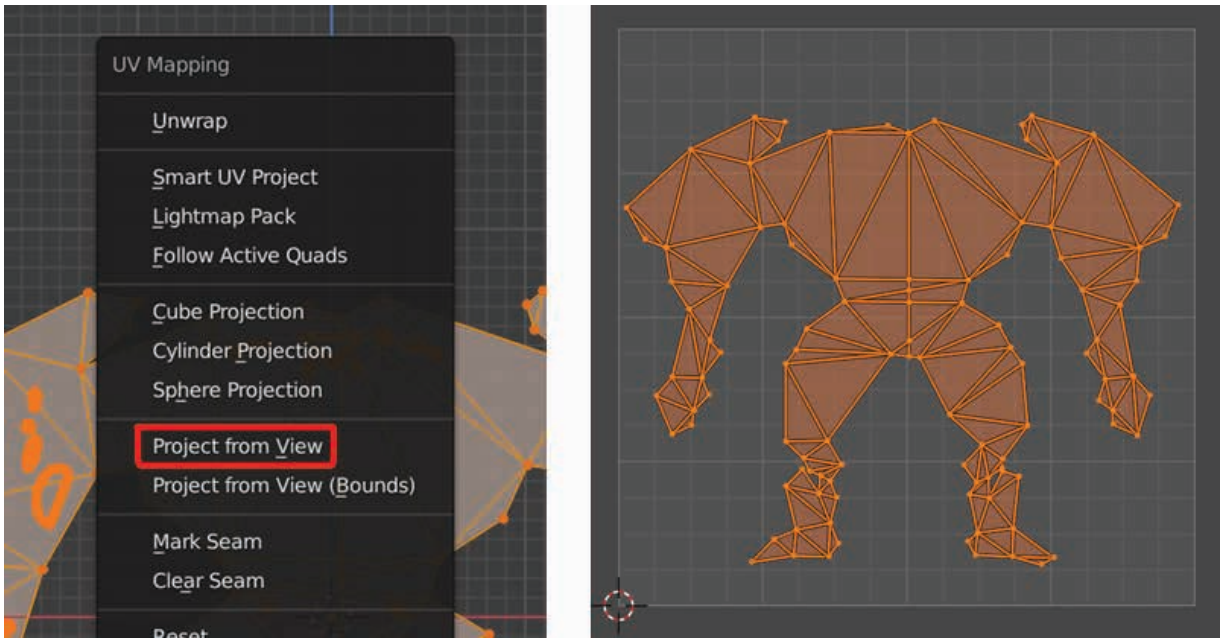
This can be more fitting for a mechanical design, but it's a creative choice which is up to you. The reason you need to triangulate first, is that the Decimate modifier won't work on just one Face.



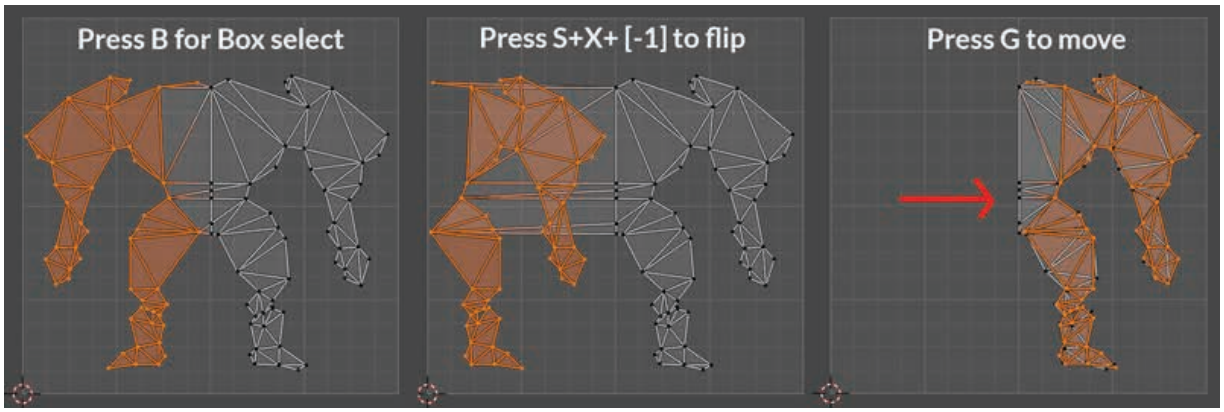
Apply the Decimate modifier. Then **press K** to activate the **Knife Tool** in Edit Mode and make a vertical cut in the middle of the mesh. **Press Enter** to confirm it.



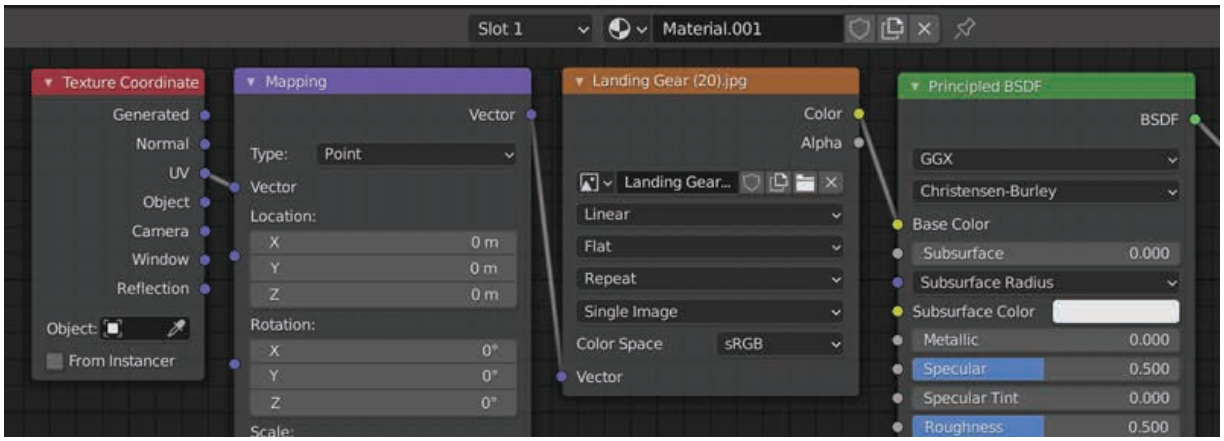
Select all in Edit Mode and **press U**, then unwrap with **Project from view.** Open a **UV Editor Window**. Scale the UV's up if necessary so they fill most of the UV Editor.



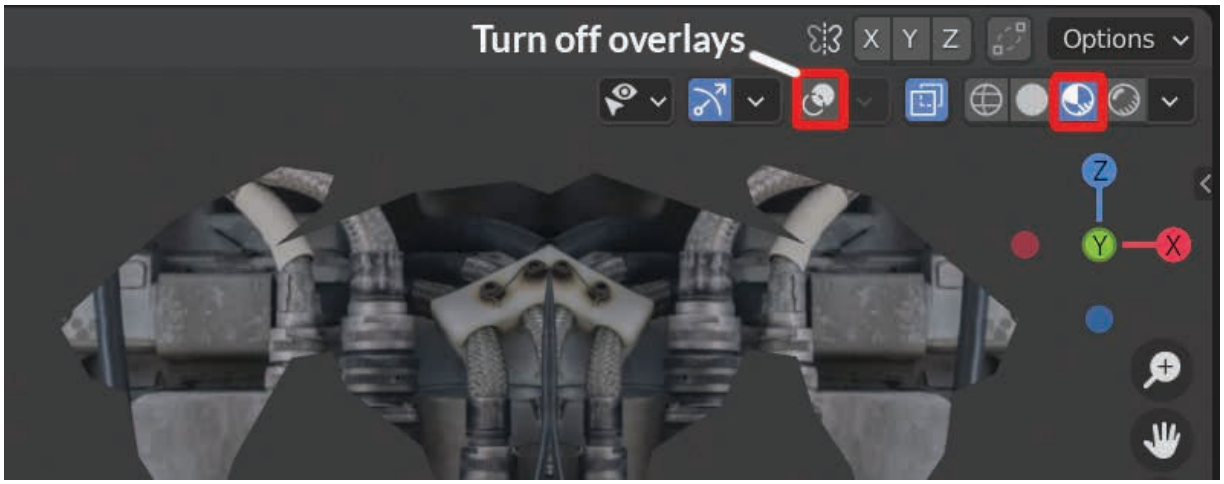
Select one half in the UV Editor window by **pressing B** and dragging a box selection around that part. Scale it -1 along the X-axis by **pressing S+X+[-1]**. Then move those flipped UV's to overlap the other half. Now any texture you use on the mesh will be mirrored.



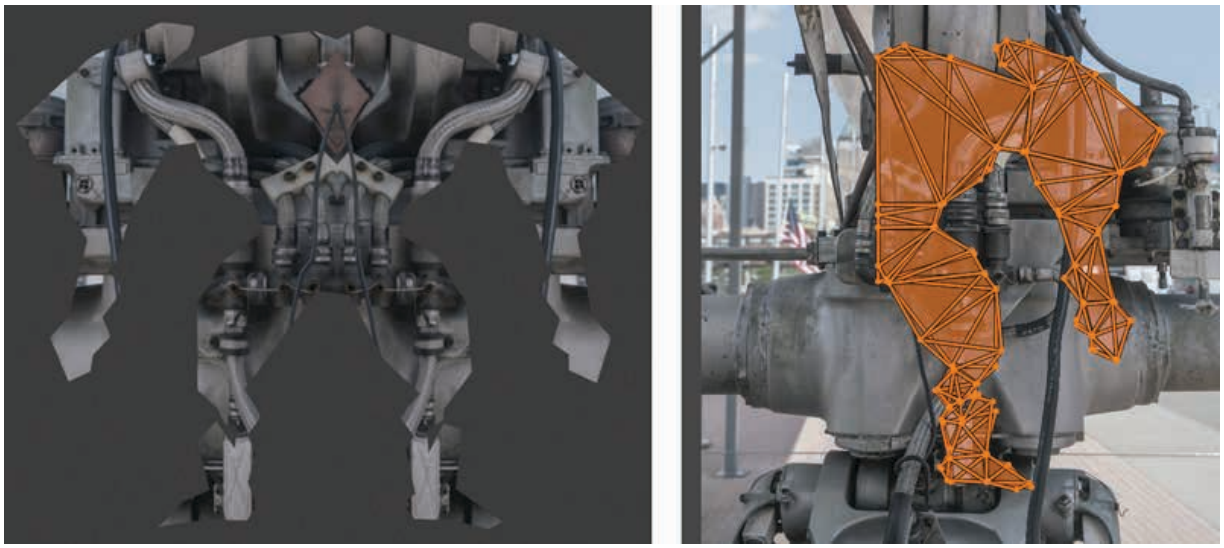
Give the mesh a new material with an image texture. I recommend turning **Specular to 0** and **Roughness to 1** so the material is not shiny.



Load an interesting photograph into the image texture node. Make sure the viewport is in Material Preview mode. You can **turn off Overlays** to see better what you're doing.



In the UV Editor window, select all and then rotate, move and scale the UV's until you find something that looks interesting.

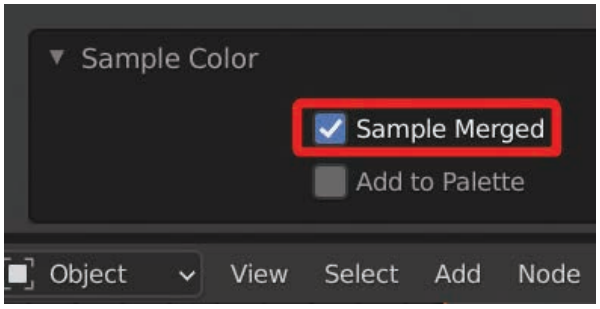


If you want you can also select parts of the model like the legs and arms, and move those around separately to find different parts of the texture that fit better. It doesn't need to be perfect or anything - what you want is something that can help you get inspired for painting, so you're not faced with a blank canvas.

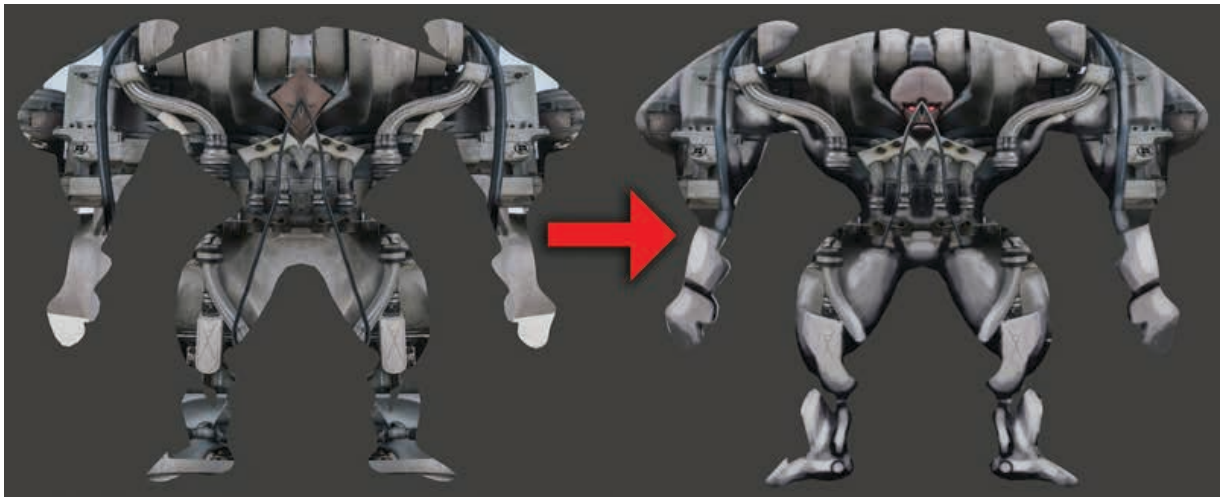
When you're happy with the placement of the photographic texture, go to **Texture Paint mode** and paint on top of the texture, picking colors by **pressing S.** Of course you could also take a screenshot of this and paint on that in Krita, where you have more paint tools.

When sampling colors, check the "**Sample Merged**" button.

This will improve greatly the sampling of the colors.



Here's a comparison between the photographic texture (left) and the result after twenty minutes of painting on it in Blender (right).



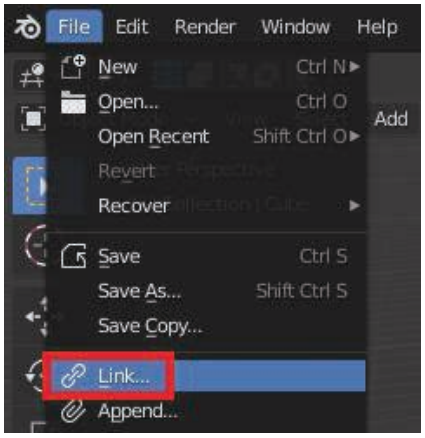
Linking Characters

If you want to use a character as a **Linked asset** in another Blend file, add the character to a **Collection** by selecting all of it in Object Mode (the rig, custom rig object, the character mesh).

Press M, then choosing a name for the collection and finally **pressing OK**.

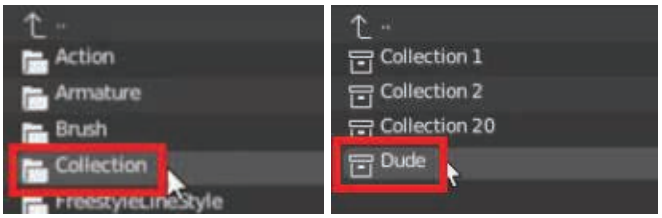
Make sure you save your file!

In a new file or wherever you want to link the character into, go to File > **Link**.



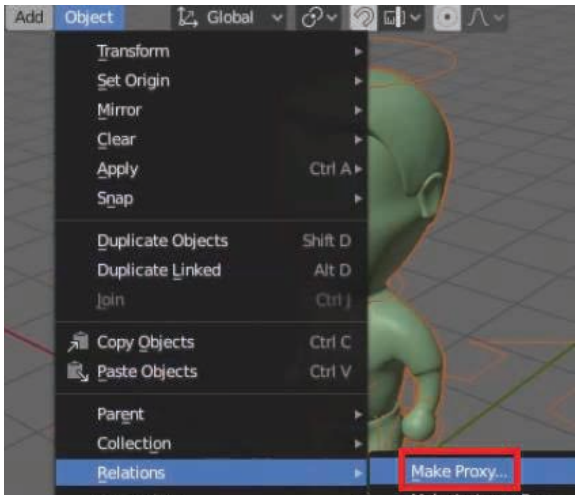
Click on the Blend file that contains the character.

Inside it, find the Collection you made in the Character file.



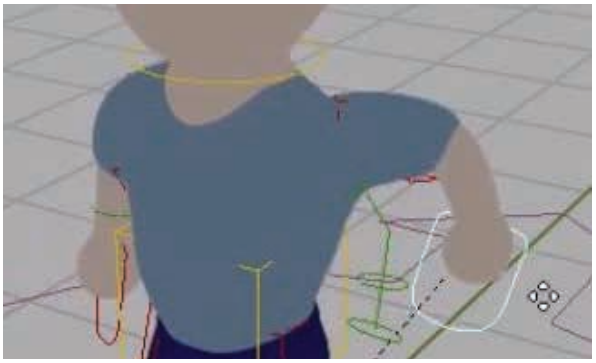
Now our character is loaded, but looks weird and we can't pose it yet.

With the linked character selected, go to Object > Relations > **Make Proxy**.



Choose the Rig.

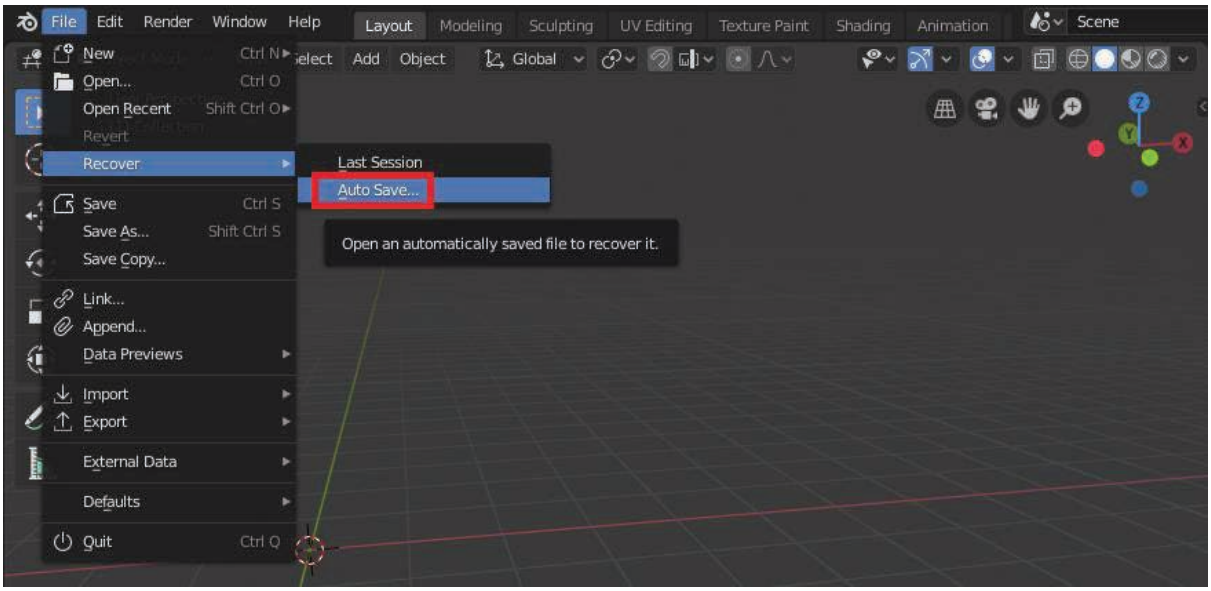
That's all, now you can go to **Pose Mode** and pose the character.



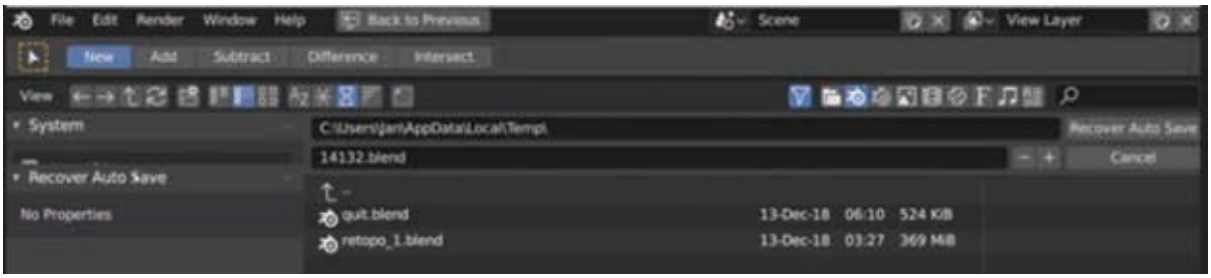
Recover Lost Blend-Files

It happens to the best of us: you accidentally quit Blender (the Q key being dangerously close to the S key that is used in the **Ctrl+S** shortcut for saving your precious work!)... and you forgot to save your hours of work! Oh no…

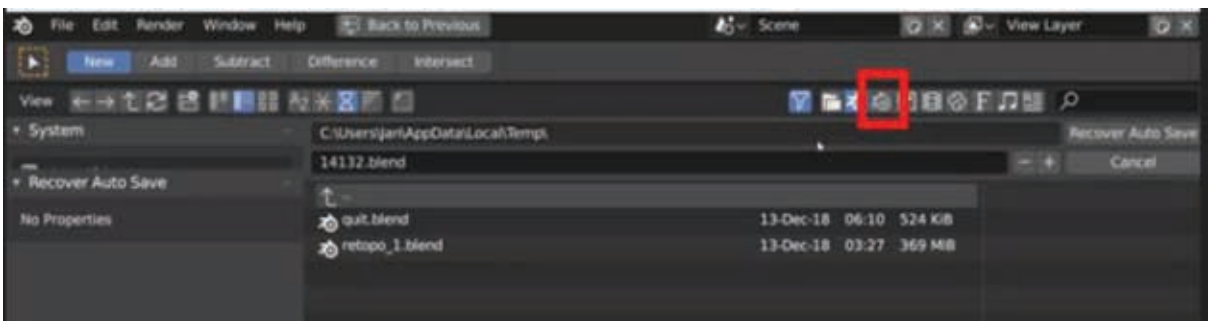
Luckily Blender is quite good at **Auto Saving**. In the **File** dropdown menu you'll find this:



If you choose "**Auto Save**" from that menu, you will see something like this (though probably with more files if you've been working for a while):

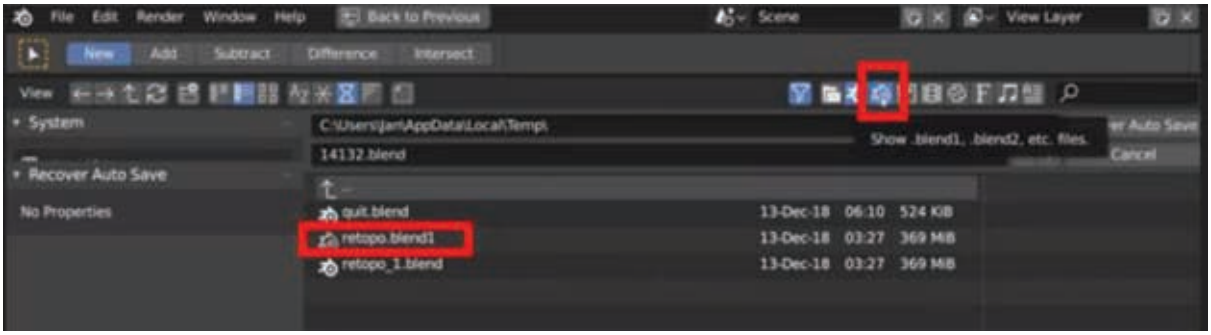


Here you will find the autosaved files as well as **.blend1** files, which are normally hidden.

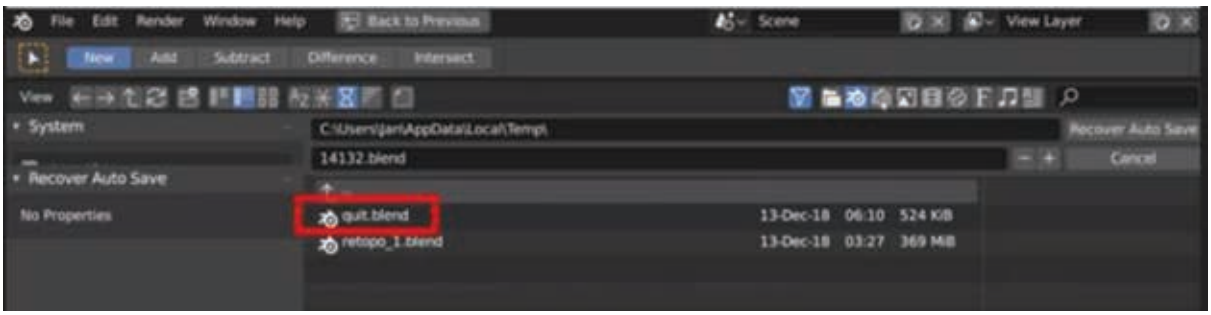


Click on this icon to see the **.blend1** files. These are always created when you work on a project, and are like a backup version of your file, that can come in handy in case you mess up.

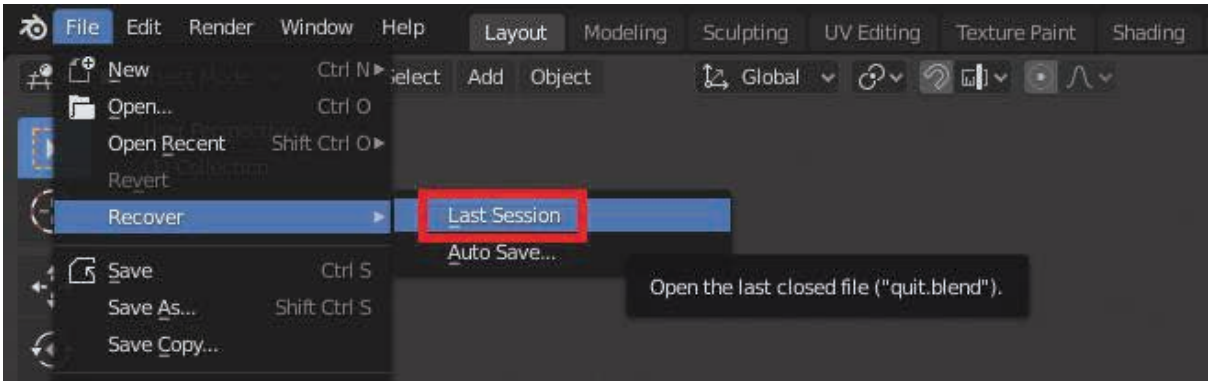
Normally .blend1 files are hidden, so you won't see them unless you click on that icon.



There's also the "**quit.blend**" file, the last thing you worked on when you quit Blender.



You can also open the quit.blend file by going to File and choosing Recover **Last Session**.



Auto-saved files have saved my butt many times in Blender.

Purge All Orphan Data

After working on a project for a while, not all materials, textures and mesh objects stored in memory are really used in the scene anymore.

But because they are still around in the file, slowing down your renders.

Get rid of them to boost your render times.

Go to the Outliner window and choose "**Orphan Data**".

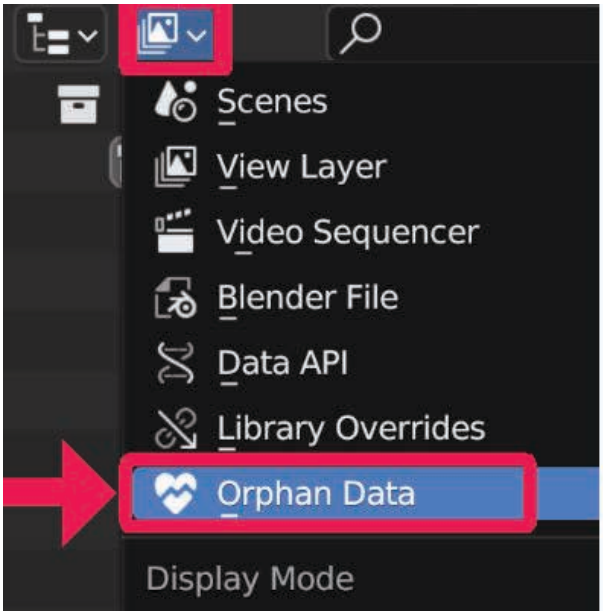
As you can see, there's 2 unused images and 10 textures in this example.

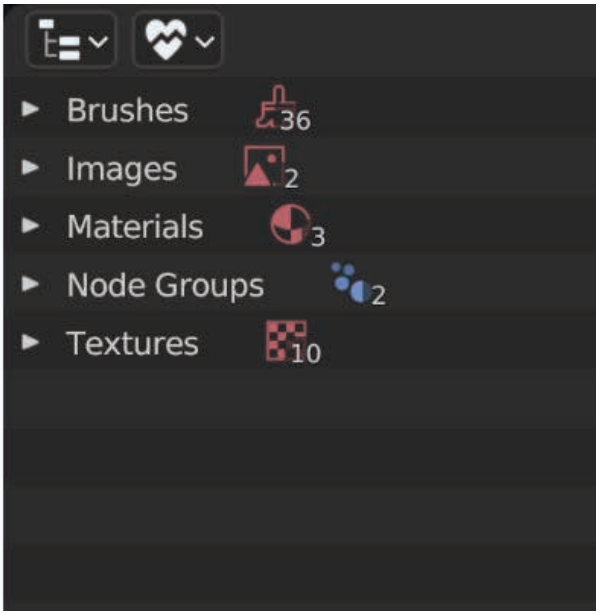
These were reference images as well as textures that I tried but didn't end up using for the final scene.

They still take up precious memory.

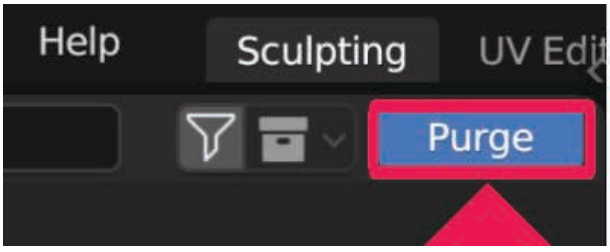
Below that you can also see that there are 2 node groups that are no longer used.

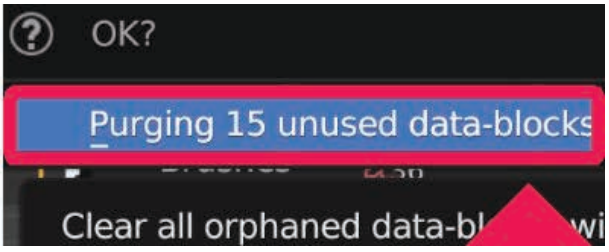
These are shaders that are no longer assigned to any objects in the scene.





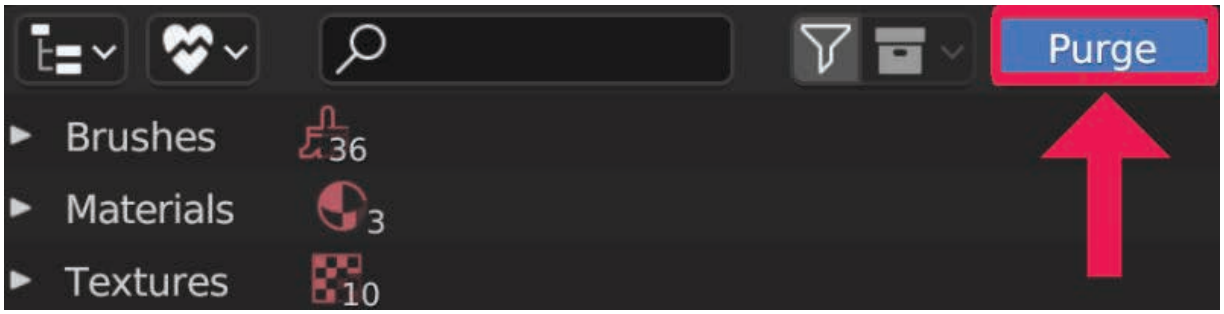
Click on the **Purge** button. You will get a warning that you are about to purge the unused data-blocks: 4 nodes, 14 images in this case. Proceed.





Oddly enough, after purging this there are 10 more orphaned images.

Those are images used by the 3 shaders (node groups) that we purged, so without the shaders they are now “orphaned”. We need to click **Purge** once more.

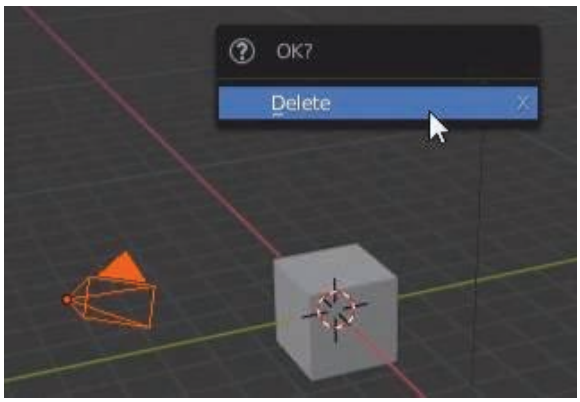


Remember, with great power comes great responsibility... think about whether you can really delete all these unused materials. You may actually regret it later! It’s always good to make a backup of your Blend-file before doing this.

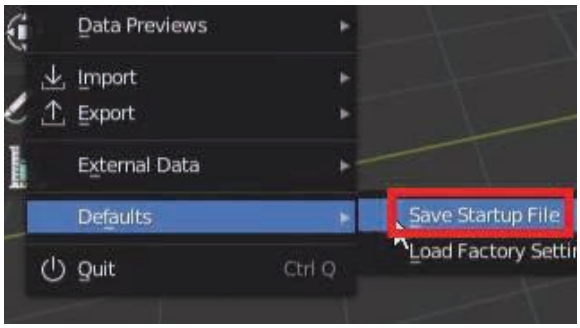
Another more "clean" way to reduce a project size or clean it up, may be to open a new file, and **append** all the objects that you need from the original file, which you can keep around as a backup.

Custom Startup Settings

If there are things you do each time you start Blender, like deleting the camera and lamp, setting the viewport to Random Colors, and so on.

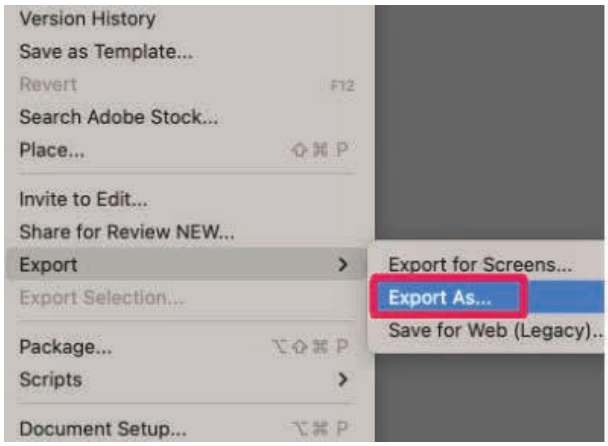


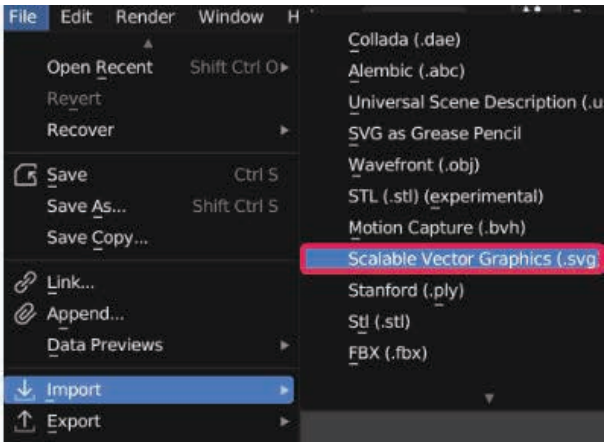
Save it as the **Startup File** so you never have to do it again.



Import SVG vectors with colors intact

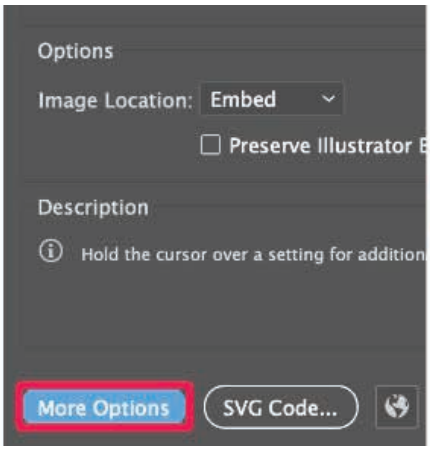
You can export an **SVG** file from illustrator (below, left) and import it in Blender (below, right).

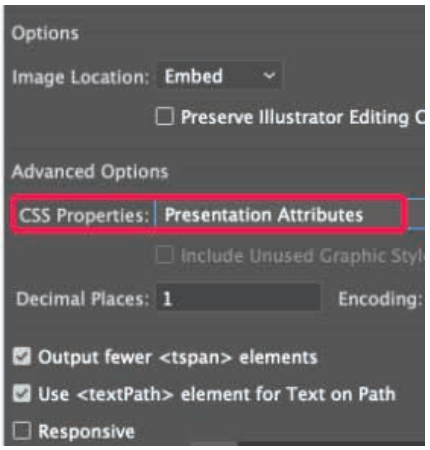




However all the colors will be lost, everything will be black.

How to solve this: First, don't export. Instead choose **Save As** from Illustrator. Use the following options, especially **Presentation Attributes**.





Now your imported SVG file will still have all its colors intact.

You may still need to offset the different layers of the illustration, in case there are parts that are overlapping in 3d space.

Bonus tip: set **View Transform to RAW** and **Gamma to 2**. This will make the colors look more like they did in Illustrator.

Red Fox illustration by Stampf on Pixabay.

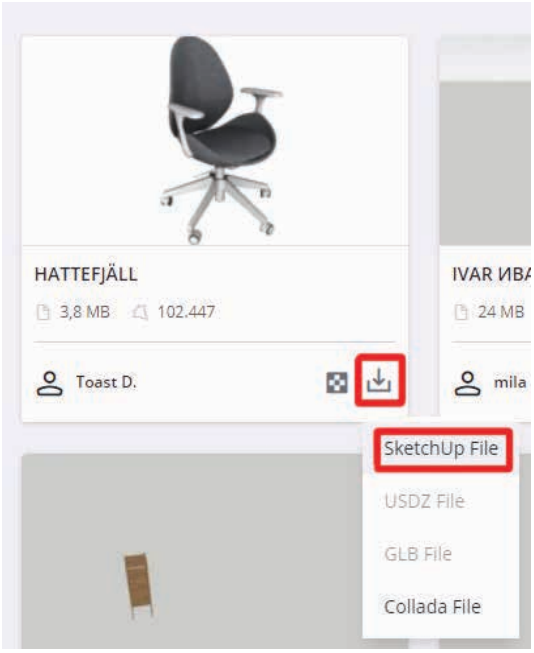
Import SketchUp models

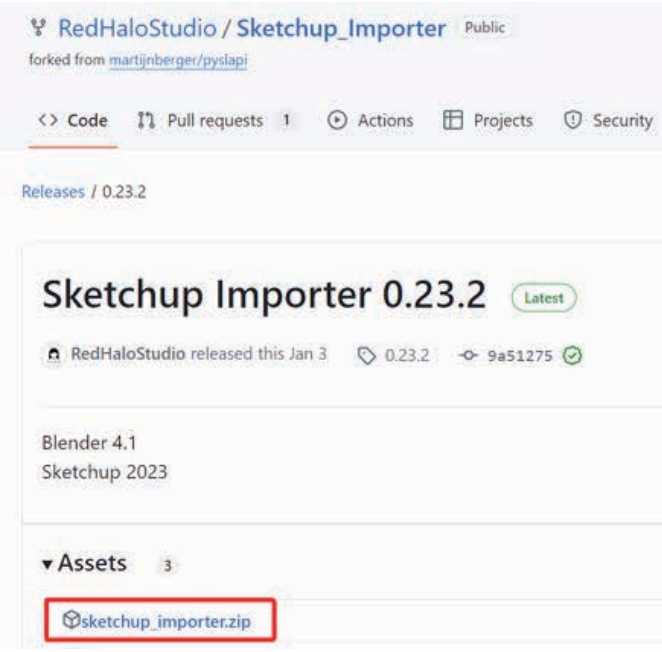
There's a free SketchUp importer for Blender.

It allows you to import the many free models from the SketchUp Warehouse website.

You'll need a free "Trimble" account to download from the Warehouse, but you can also use your Google account to log in. Then download models to your hearts content.

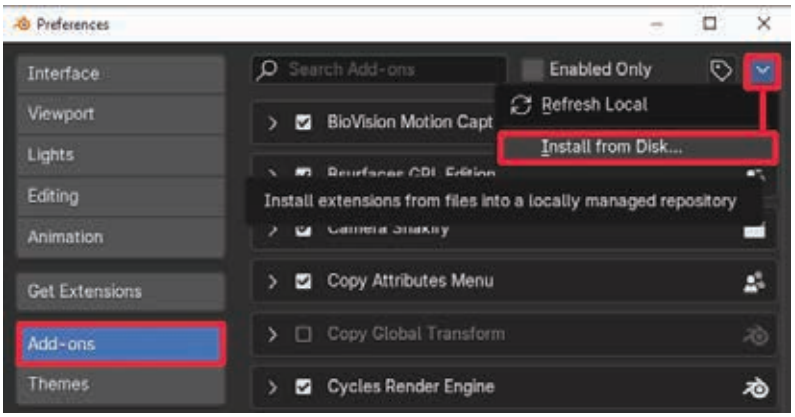
And also download the free add-on from RedHaloStudio on Github.



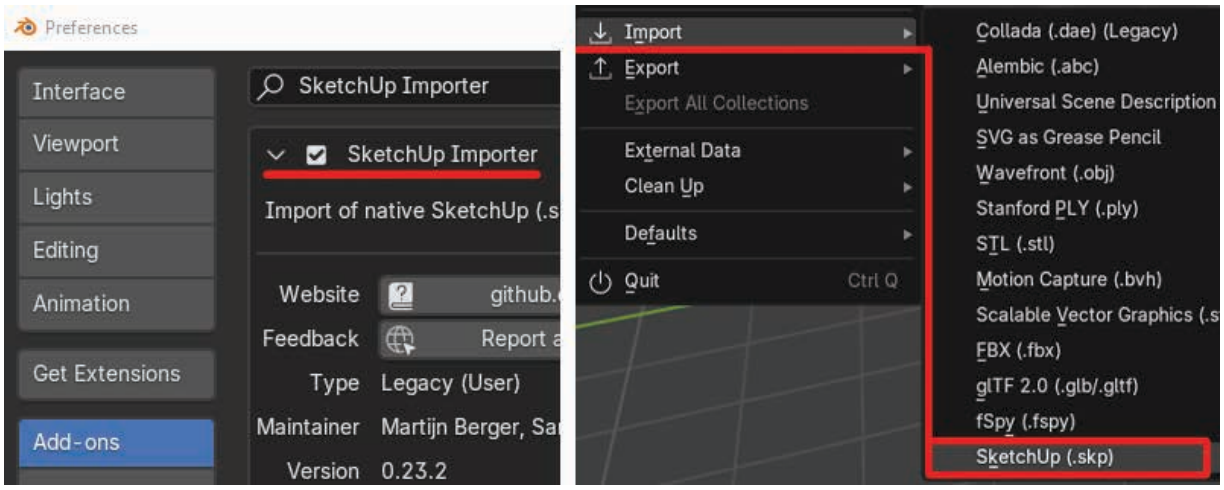


To install the add-on, click on the drop-down icon in the Add-ons tab in Preferences.

Then choose Install from Disk and select the downloaded ZIP file.



Once the add-on is installed, it will be automatically enabled. Then you can go to File > Import > SketchUp to import the downloaded model.

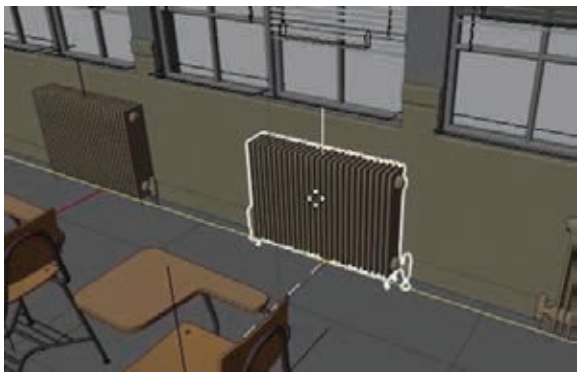


The SketchUp Warehouse also has a Collada download option. However in my experience, these usually require a lot more work to use. The SketchUp file comes with textures, and only needs some fine-tuning of the parts that only have a shader, such as the legs of the chair in this example.

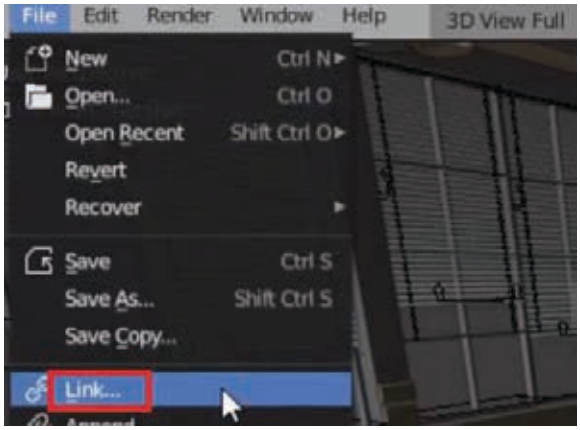
Workflow tip: Linked Libraries

When working on an environment with many objects, make use of Linked Library objects.

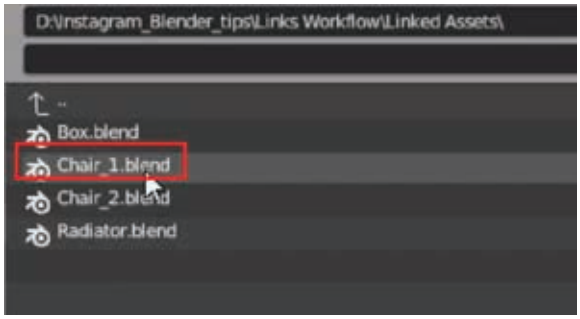
In other words have your main file and then use separate files for each object.

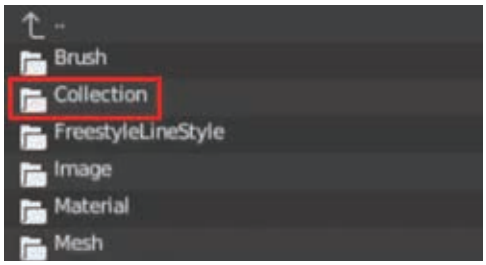


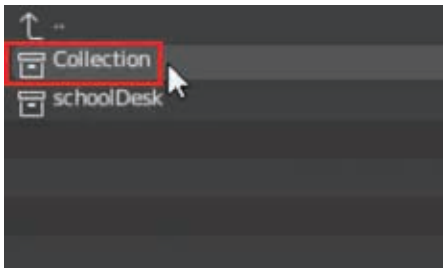
Simply import the Linked asset from File > **Link**.



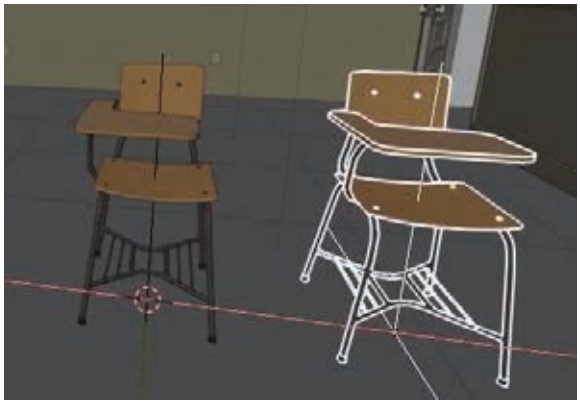
Navigate to the .Blend file, open the **Collection** folder and choose the collection you need.





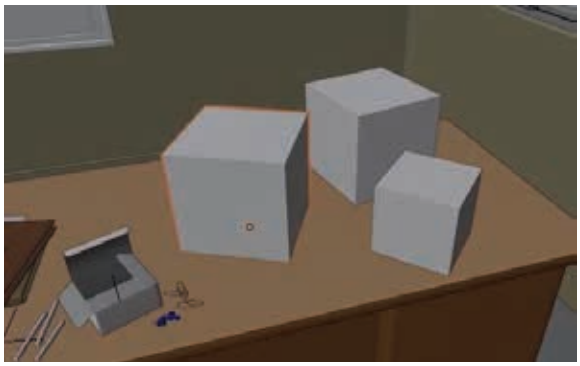


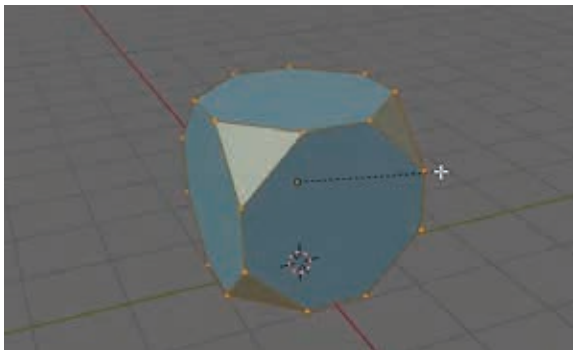
If you want more of the same asset in your scene just use **Alt+D** to make more **Linked Duplicates**.

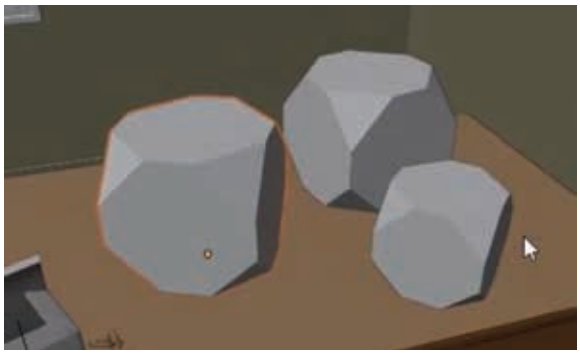


Any changes to the object have to happen in the original objects' file.

The changes will then be visible in the main scene file.

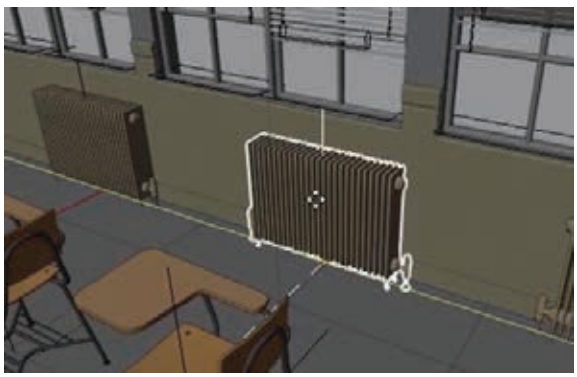






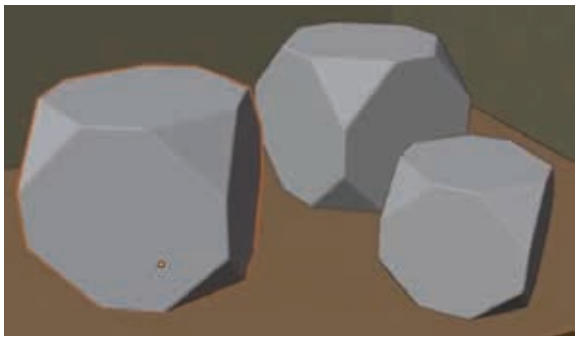
Benefit: it's easier to model an object in its own file, without any other objects blocking your view.

It also makes your main scene file much lighter and easier to manage.

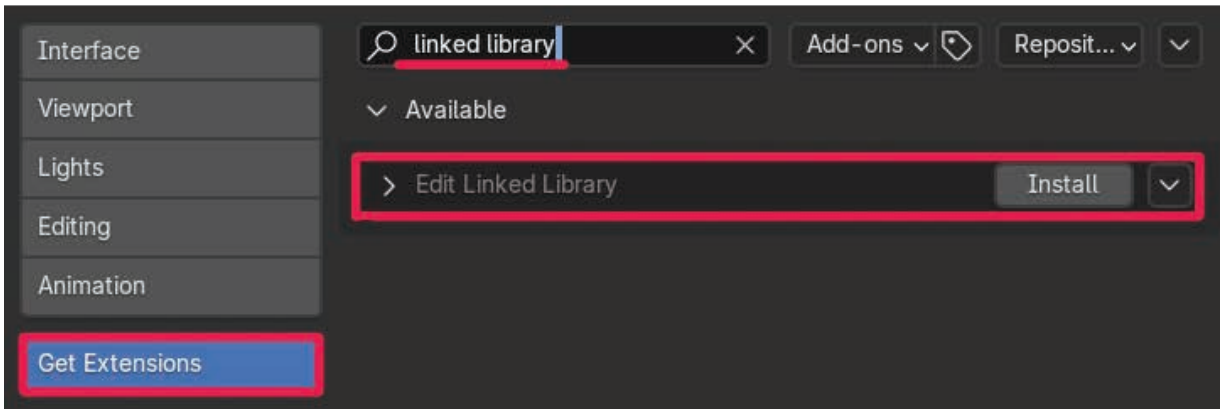


Editing Linked Libraries

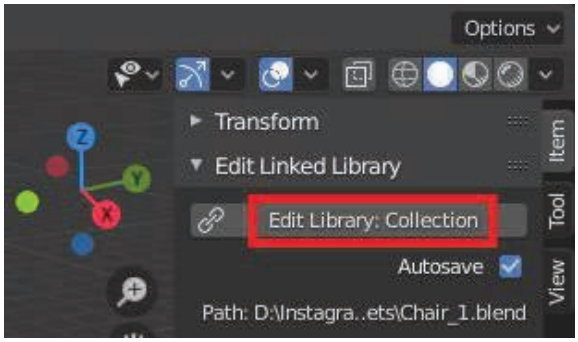
While using Linked Assets is great, it may be inconvenient if you want to edit them when already linked.
Perhaps you don't want to have to search for the asset file each time you need to edit it.



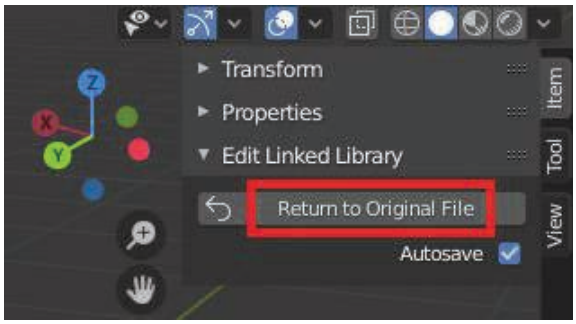
Luckily the **Linked Library add-on** gives us a quick way to edit the Linked assets.



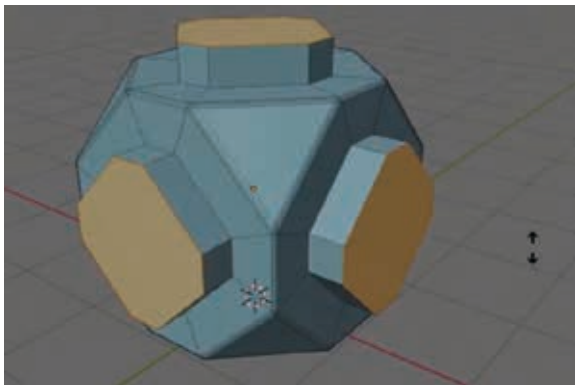
In the Option Panel, click on **Edit Library: Collection** with the Linked Asset selected. Note that your main scene file must be saved to be able to do this.



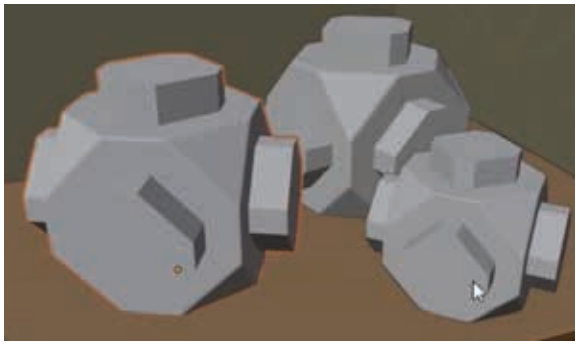
Then in the Option panel click on **Return to Original File**.



This transports us to the Linked asset file, where we can do the necessary editing.

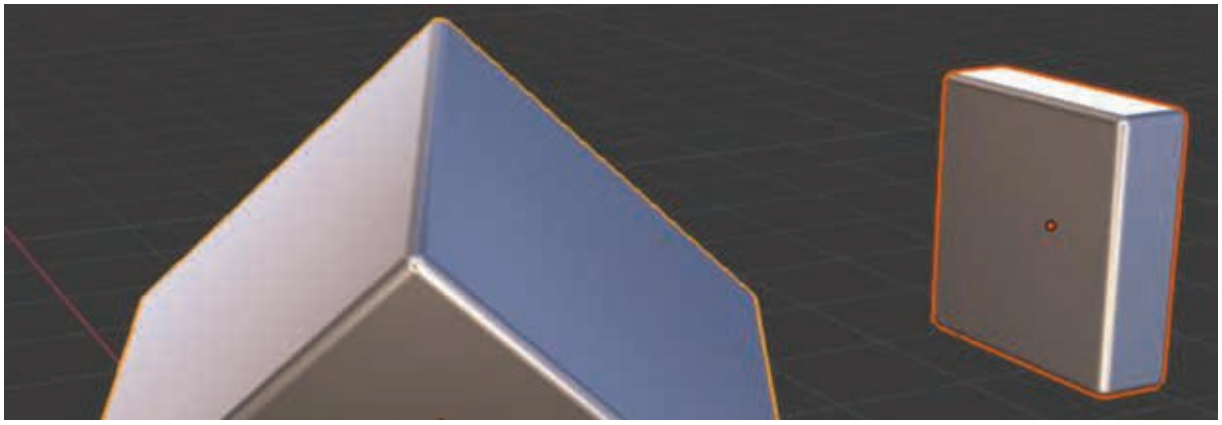


And our Linked Assets are updated in the main scene file. This saves quite a bit of time when using Linked assets.

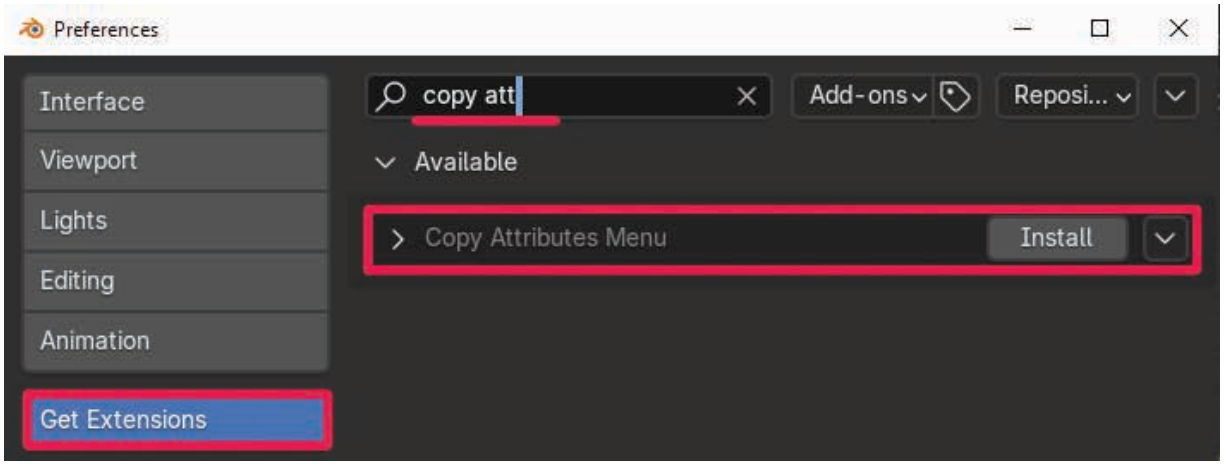


Copy Rotation with Copy Attributes Menu Add-On

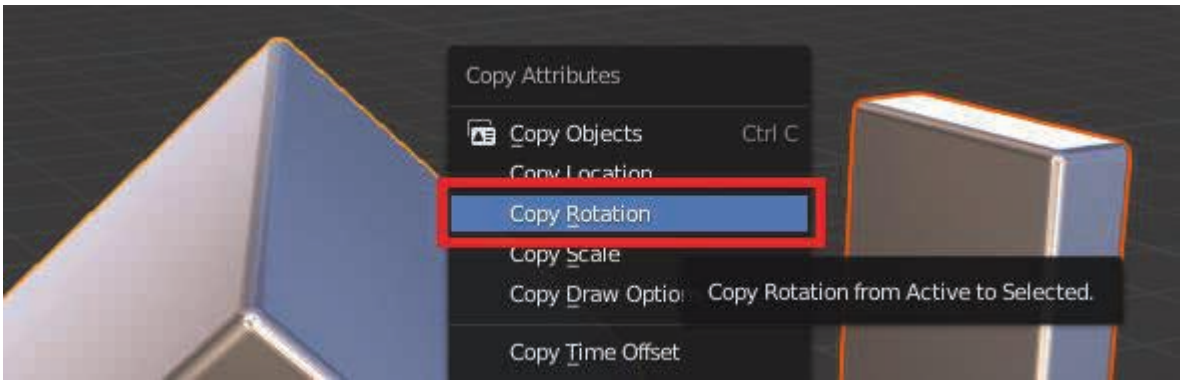
What if you want to line up these two objects perfectly?



You need the **Copy Attributes Menu** add-on to be able to do this. Open Preferences, go to the **Get Extensions** tab and find the add-on it in the search bar. Click on Install.

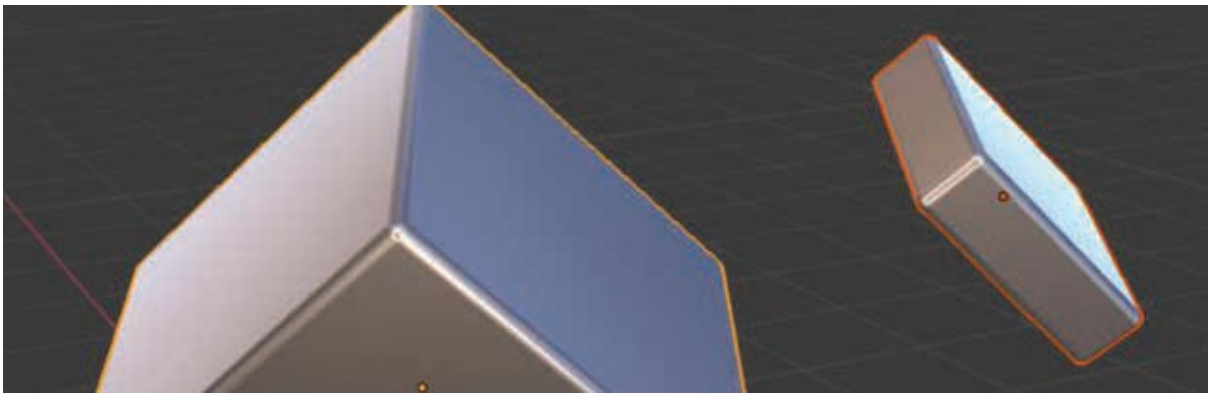


While **holding Shift**, select the object for which you want to change the rotation and then the object that you want to match it's rotation to (so in this case, first the small cube, then the big cube). **Press Ctrl+C** and choose **Copy Rotation**.



The Copy Attributes menu.

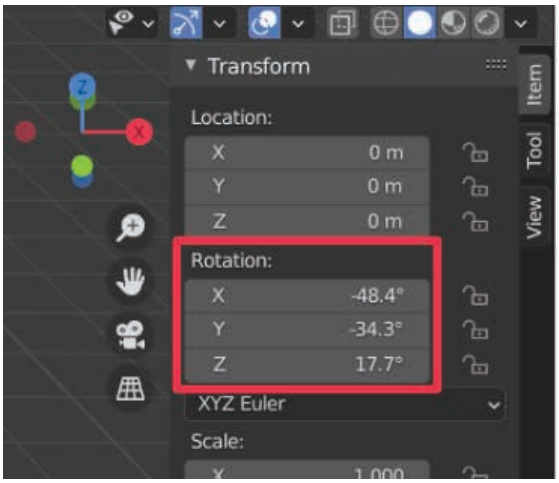
The small cube now matches its rotation angle to the big cube.

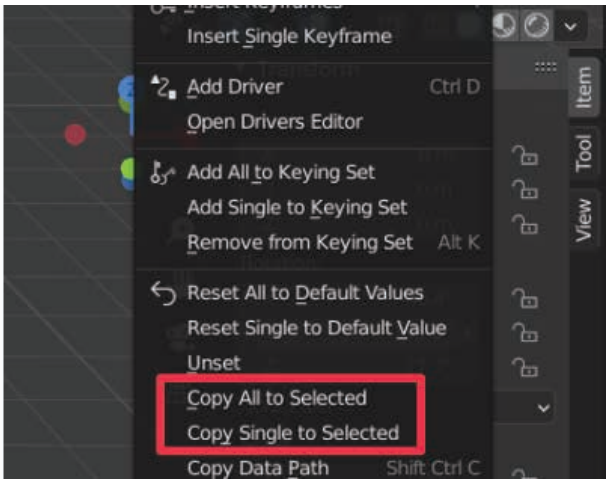


Bonus tip: this is also possible without the Copy Attributes add-on.

Select the object you want to rotate and then the object you want to copy the rotation from. Go to the **Option panel**. Hover the mouse over the Rotation field and right-click.

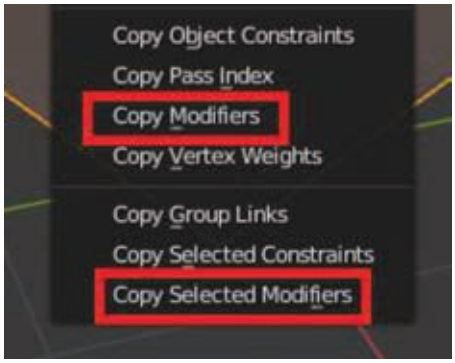
Then choose "**Copy All to Selected**" to copy all rotation values or "Copy Single to Selected" to copy only one of them. This can be a good method if you only want to copy one axis of rotation instead of all three.





The **Copy Attributes** menu is a powerful addition to your Blender tools.

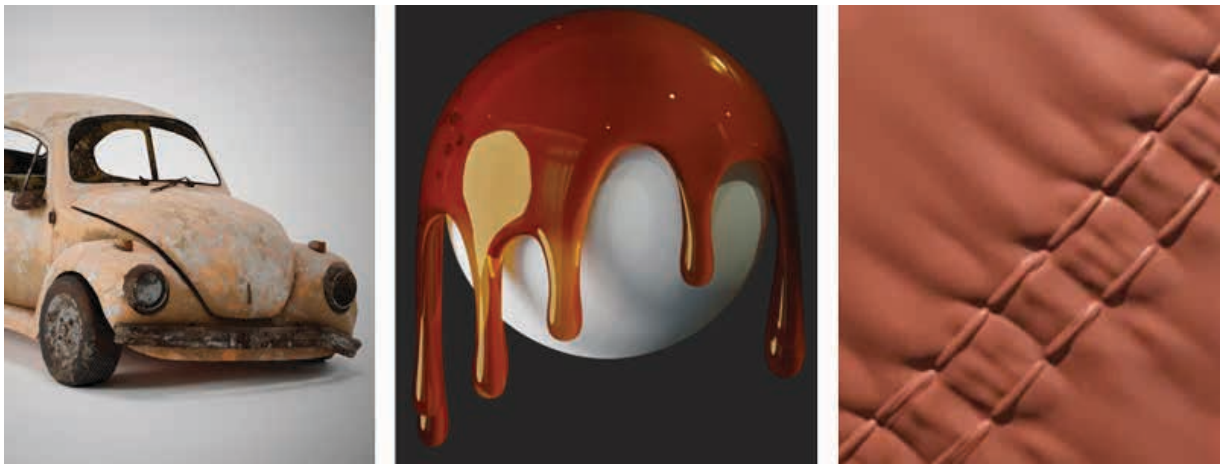
I like to use it to copy Modifiers from one object to several other objects. It even gives you the option to choose which modifiers to copy, and which not to (with the "**Copy Selected Modifiers**" option).



Get 6000 free assets and an Asset Manager with Blenderkit

Blenderkit is an add-on that works much like the new Asset Browser, except it's already been around for much longer, and comes with a lot of assets. It allows you to browse assets, specifically **sculpt brushes, models and materials**, with visual icons.

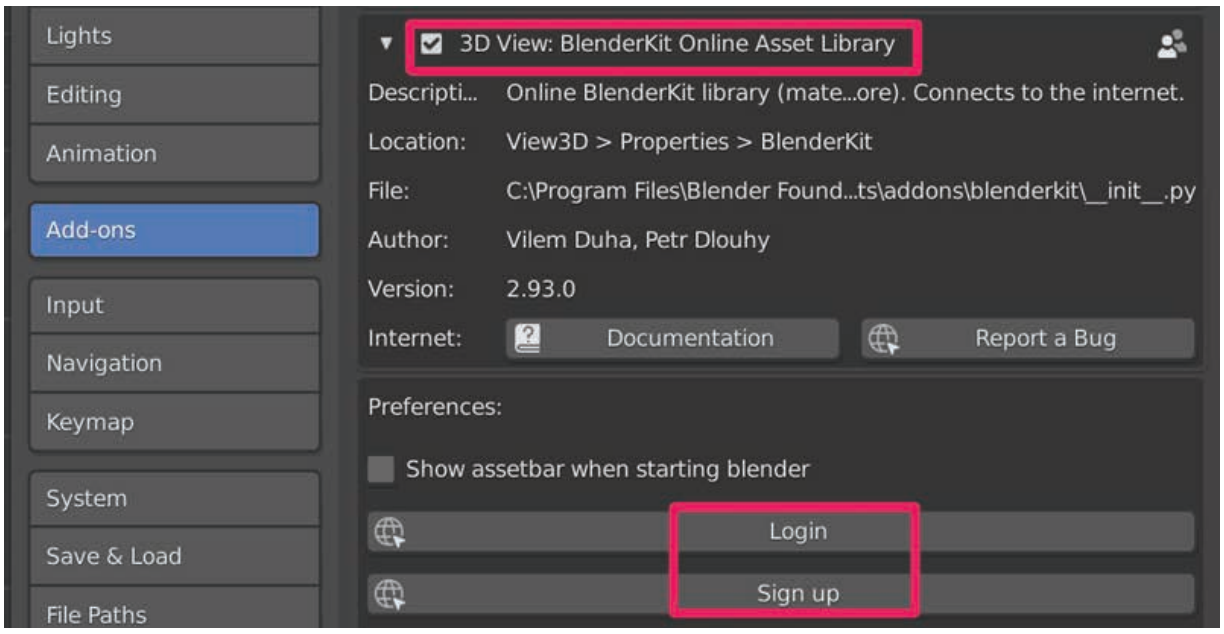
There are more than 12000 assets presently in the library, of which more than half are free. So that's over **6000 free assets**, directly accessible from inside of Blender, and people can sell their own assets directly in Blender. There is also a very affordable and simple premium plan that gives access to everything else and supports asset creators and Blender development. Download the add-on here (note, this is an affiliate link).



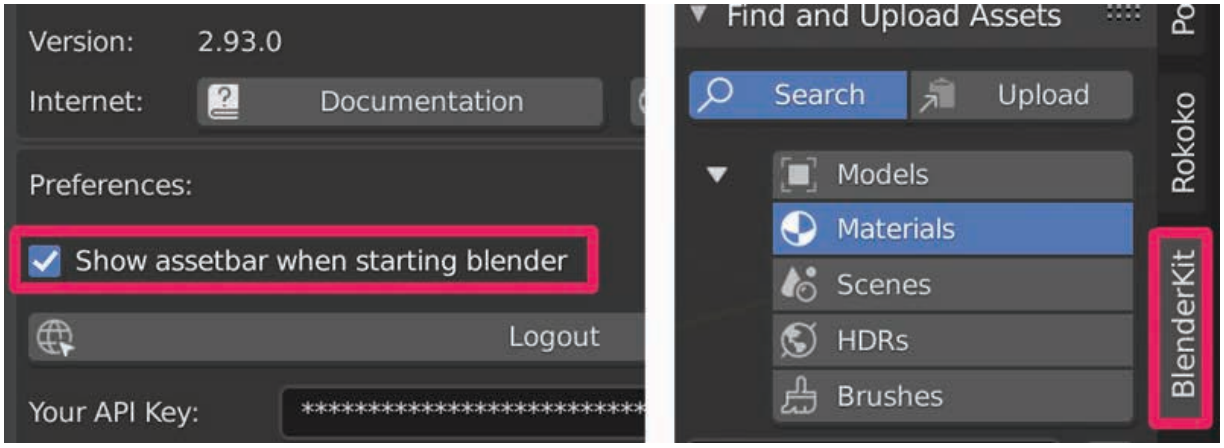
Free model, material and sculpt brush.

After downloading, install the add-on via de drop-down in the Add-on tab in preferences and the **Install from Disk** option.

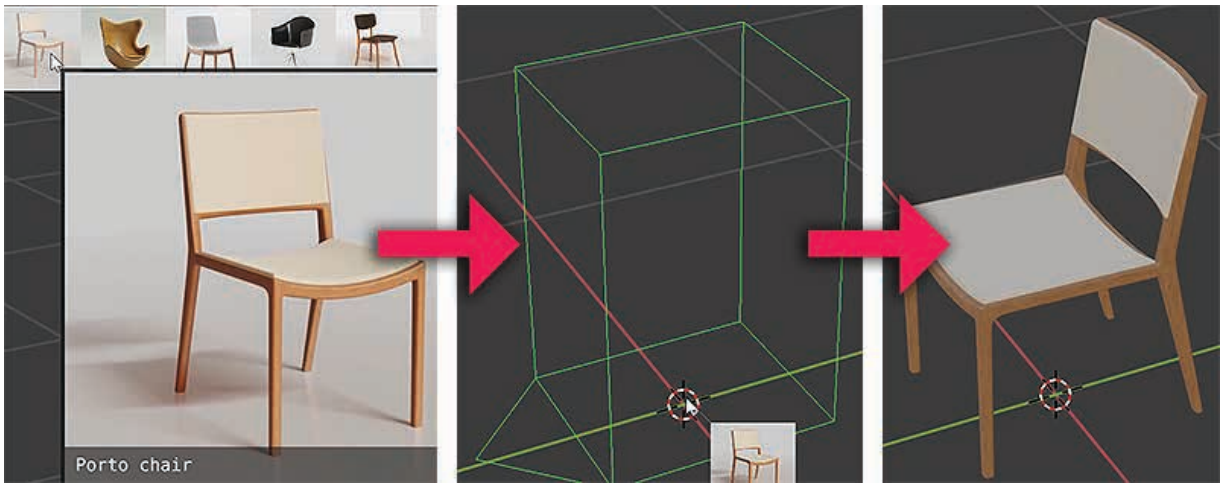
Although you don't have to register to use the add-on, it gives you some extra options like the ability to rate assets. Of course if you decide to get the premium plan, you do need to register. You can **Sign up and Login** from the add-on options in Preferences.



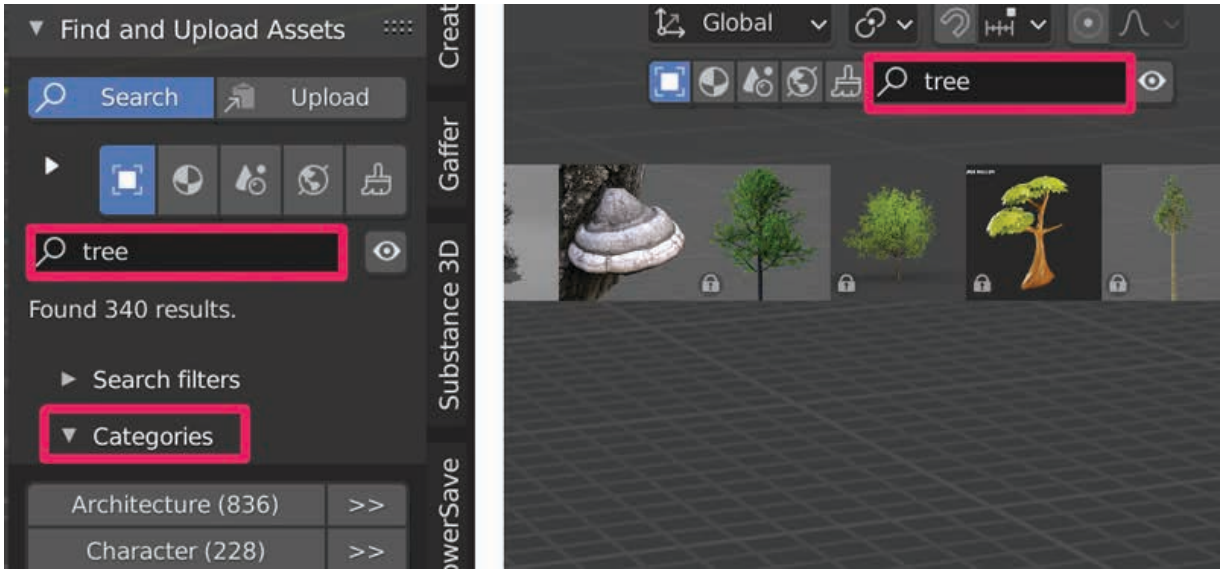
You can also check options that start the Blenderkit assetbar every time you start Blender. Personally, I think that's a good way to avoid forgetting about the add-on. Having enabled the add-on, we can find it in the **Option panel** (open that by **pressing N**).



The assetbar lets you scroll through models, and then drag and drop them to the 3D Viewport.

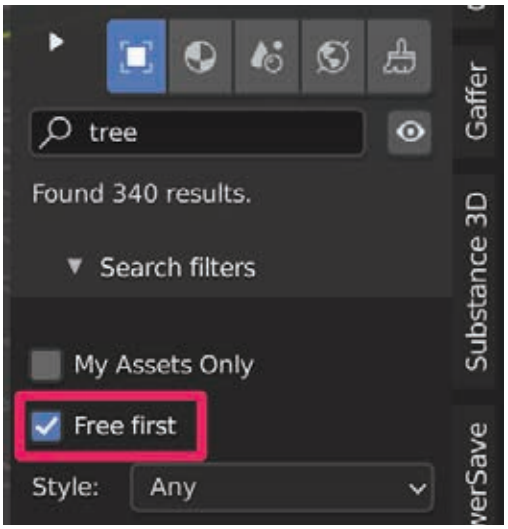


You can also choose a **category** in the Option Panel, and even search for a specific keyword in the **search bar**.

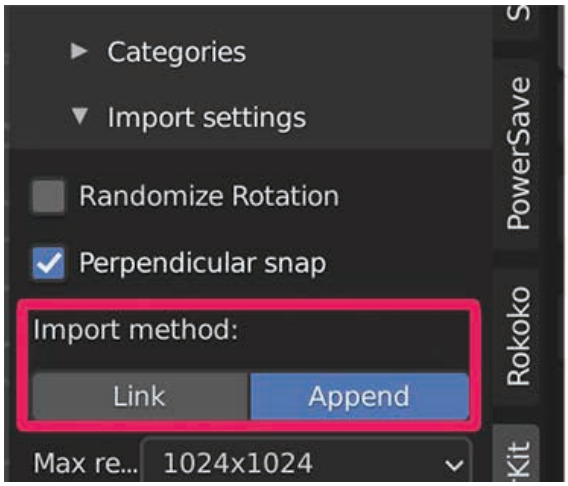


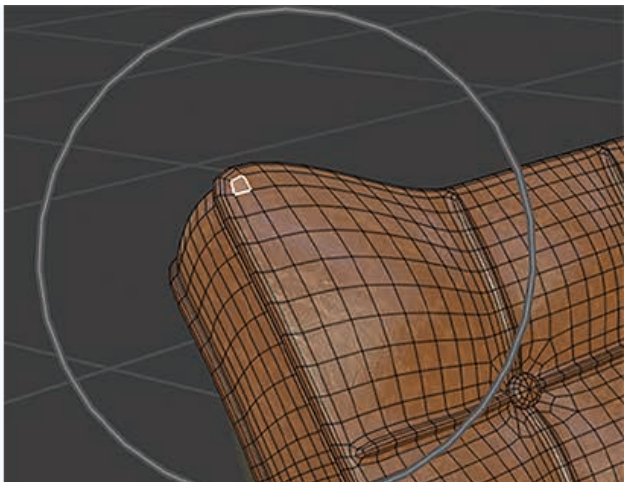
After which you can scale them up and transform them any way you like.

To filter it so you only see the free assets, check the **Free first** box.

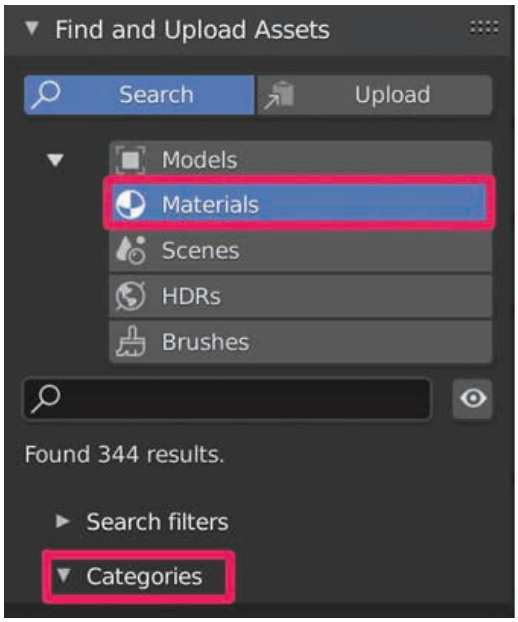


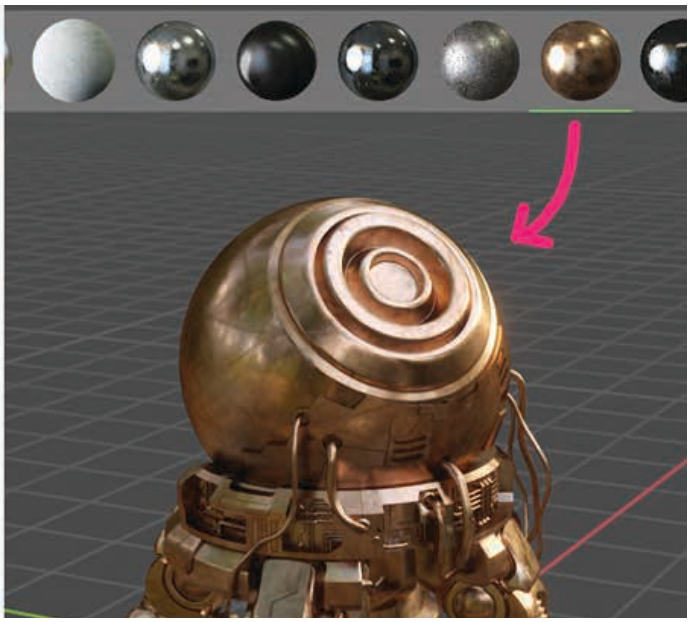
Under **Import Settings** you can choose whether to **Link or Append** the assets (see below). Linking means that the asset is an instance linked from another file, and you can't edit the asset directly. If you want to be able to edit the asset in Edit Mode, use **append** instead.





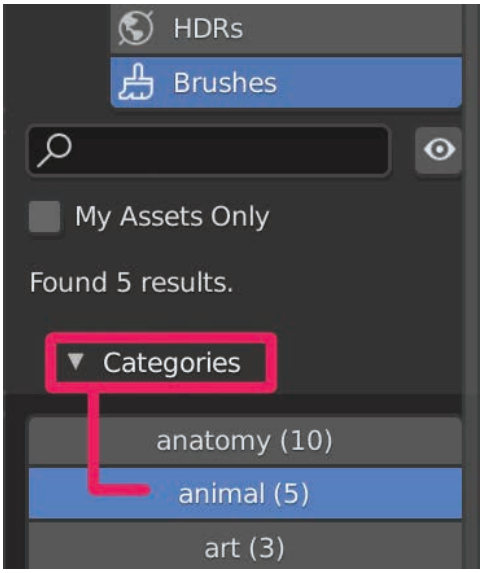
Just like models, **materials** can be dragged-and-dropped, in that case to an object in the 3D viewport. Wait a moment for the asset to download, and it should appear on the model (as long as you're in Material preview or Rendered preview). Click on **Materials** and then choose a **Category**, for example Metal.





For the sculpt brushes, **left-click on a brush thumbnail** and it will start to download. When the line underneath it is green, it should be ready to use. **Right-click** on a brush thumbnail in case you want to read more information about it or to rate it.

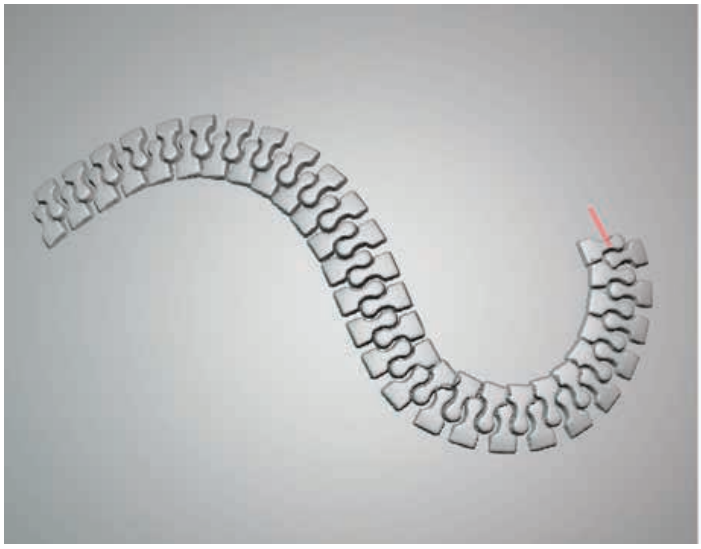
In my experience, you need to **select a category** in order to see any Sculpt brushes.



The brushes are loaded with all the optimal settings for that brush. For example, some are set to work as Stencils (like the baroque decorations. (To learn more about how to use Stencils in sculpting, check this video).

Using the Stencil Brush to add sculpted detail

Some brushes are repeated with space in between like the Bolts with washers or the **Zipper brush** (pictured below left), and some are anchored like the **Stitched Skin wound** (below right).

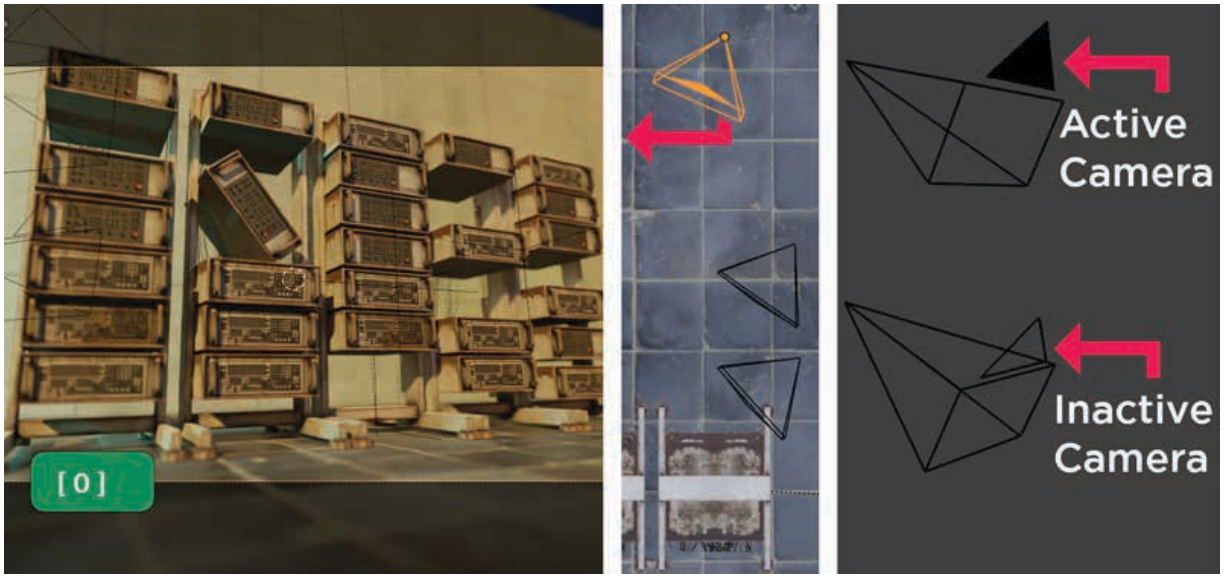


You will notice that some assets are “locked” (indicated with a little lock icon). That means these are part of the premium plan. As an affiliate partner, I can share a link which gives you **10% discount:** https://blenderkit.com/r/blendersecrets/ on the premium version of Blenderkit.

By paying the small monthly fee, you also support the artists who created the assets as well as Blender development. This is of course all completely optional, and 6000+ free assets should keep you busy for a while.

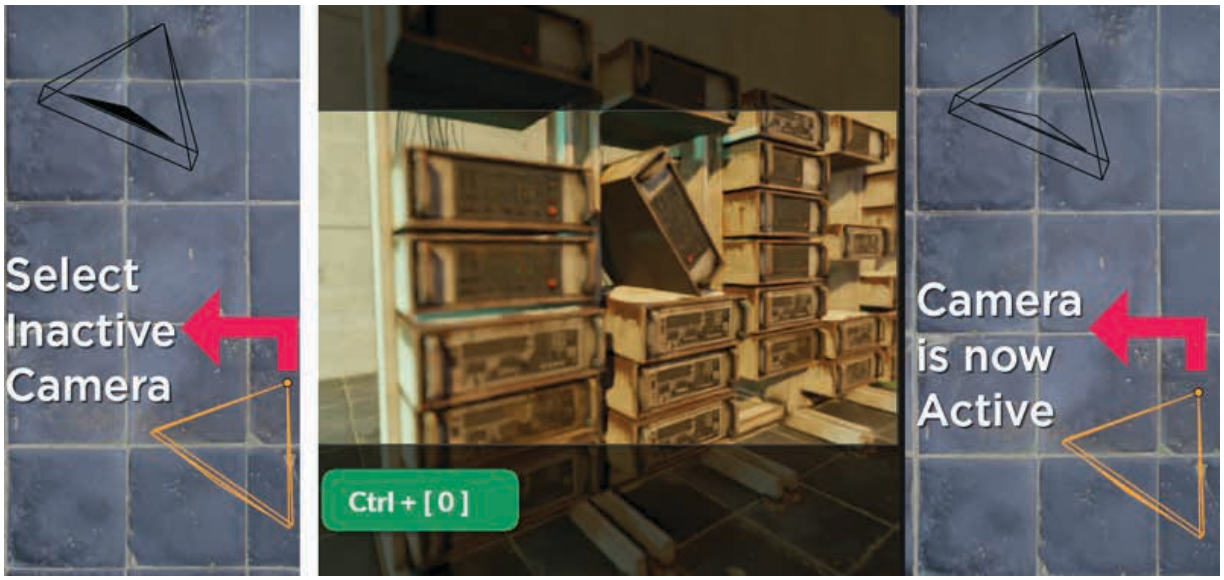
Switching between different cameras

In Blender the **Active camera** is the one which is rendered. **Press 0 on the numpad** to see through the Active camera.



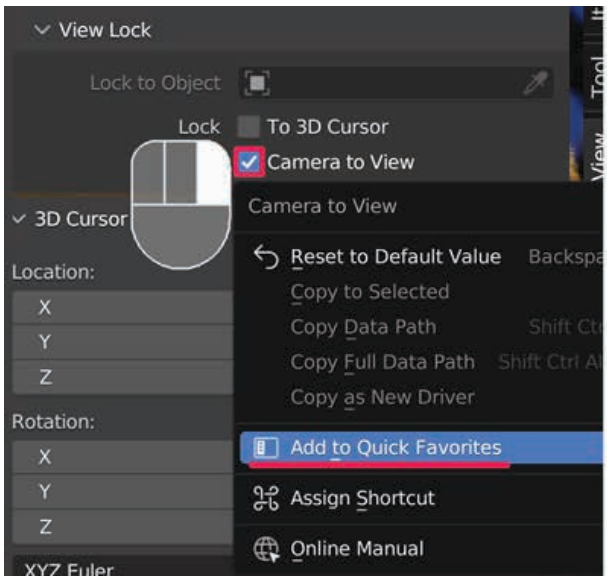
3D model by Renafox on Sketchfab.

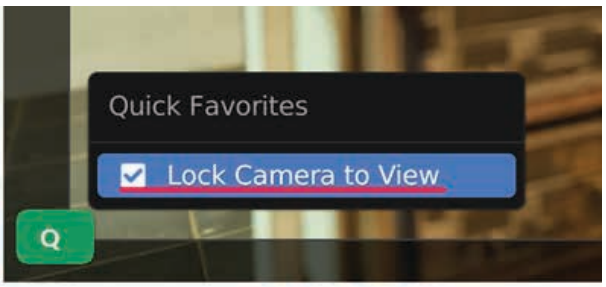
But what if you want to switch to a different camera? Select that camera and **press Ctrl+0 on the numpad**. This will make that selected camera the active one.



If you check "**Lock Camera to View**" you can move the camera just like you would the 3D viewport.

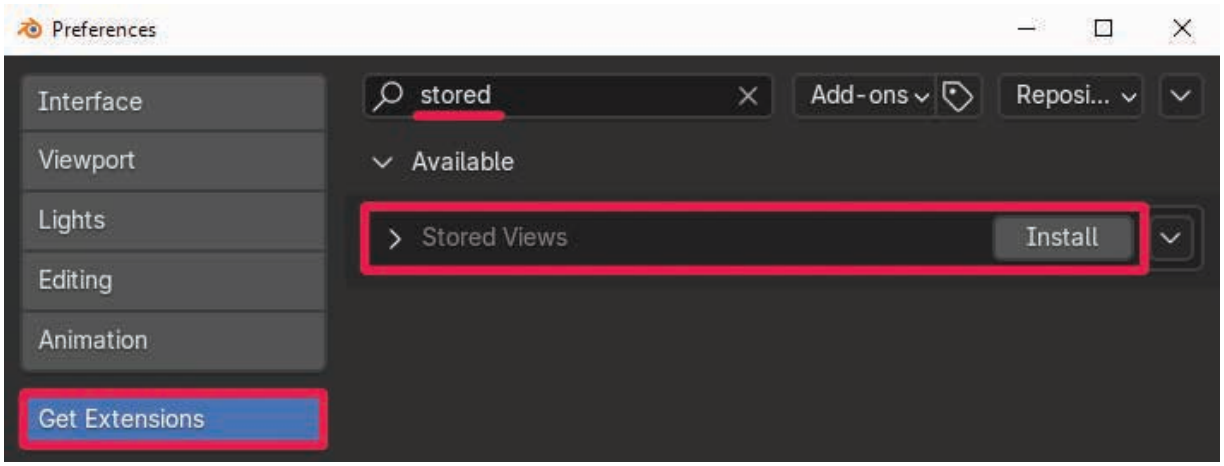
I like to add this checkbox to my quick favourites.



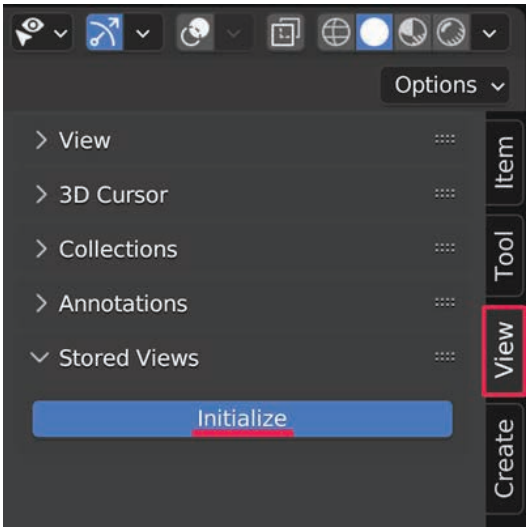


Stored Views

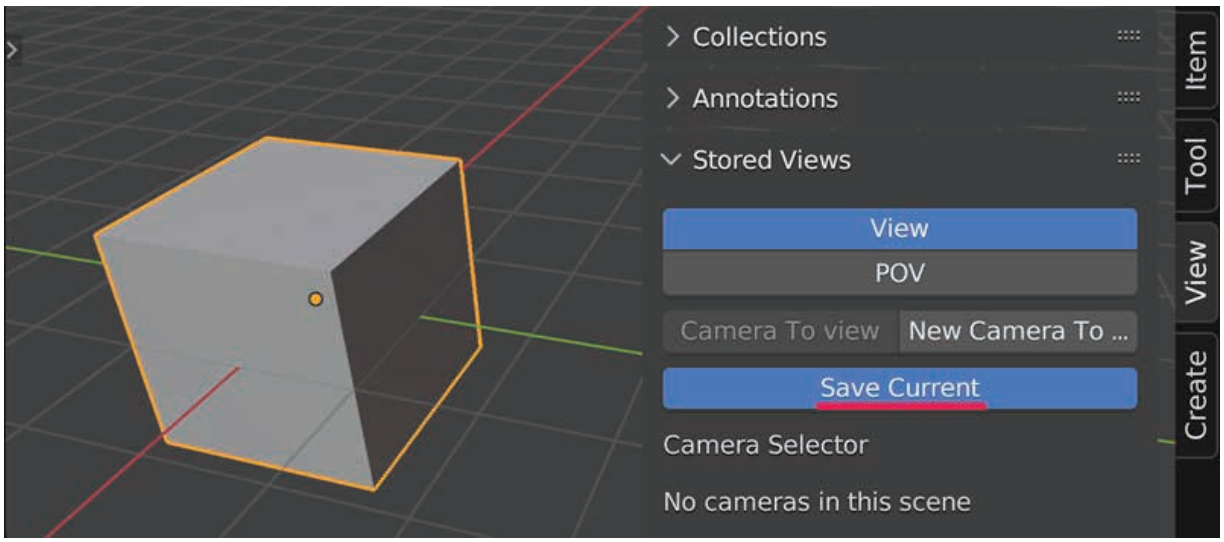
In Preferences, find and install the **Stored Views** add-on in the Get Extensions tab.



In the **Option Panel** click on the **View** tab and under Stored Views, click on **Initialize**.

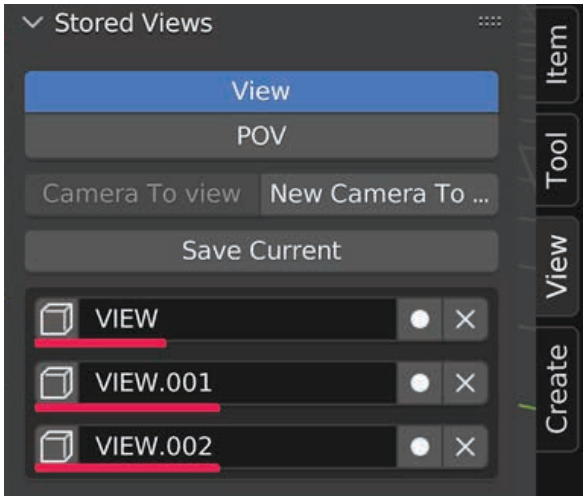


By clicking on **Save Current**, you create a kind of bookmark for that view.

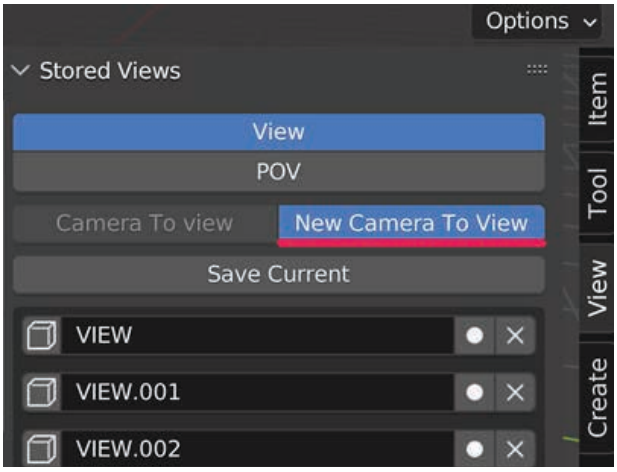


Click here to return to the index.

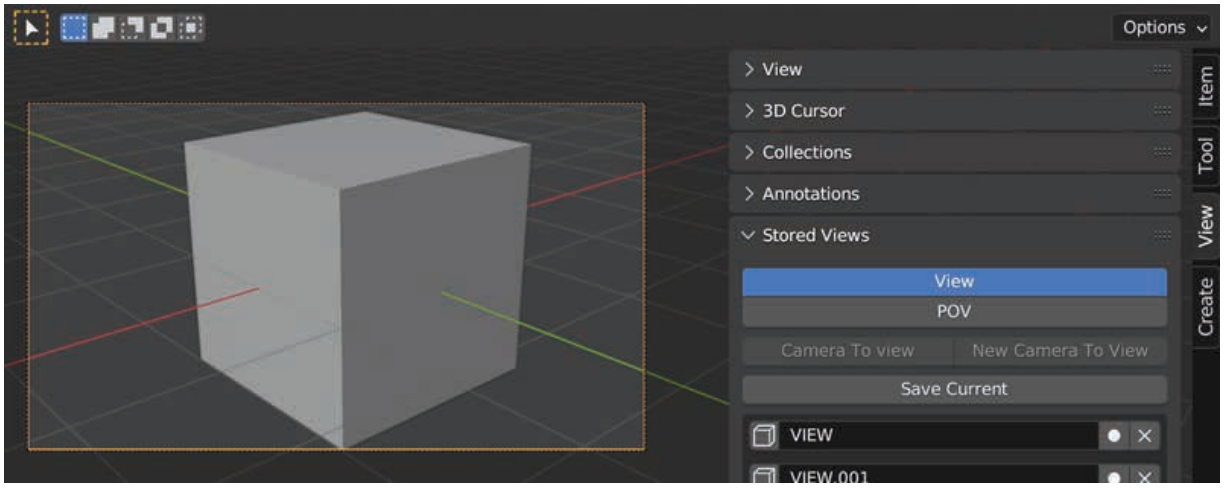
You can create several views.



Click on "**New Camera to View**" to create a Camera and set it to the current view.



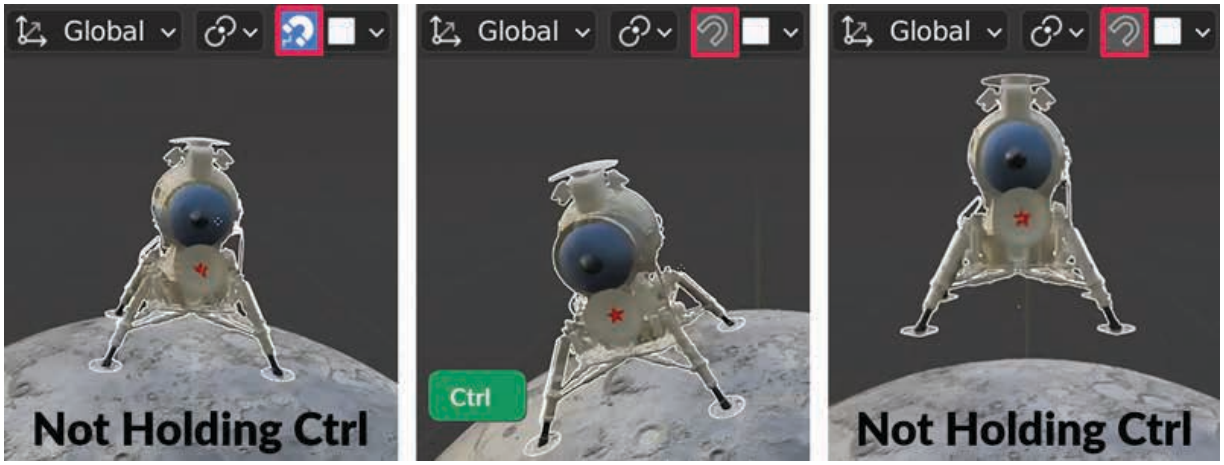
You can create several camera's this way and switch to the camera you like by pressing on the relevant button.



You can create several camera's this way and switch to the camera you like by pressing on the relevant button.

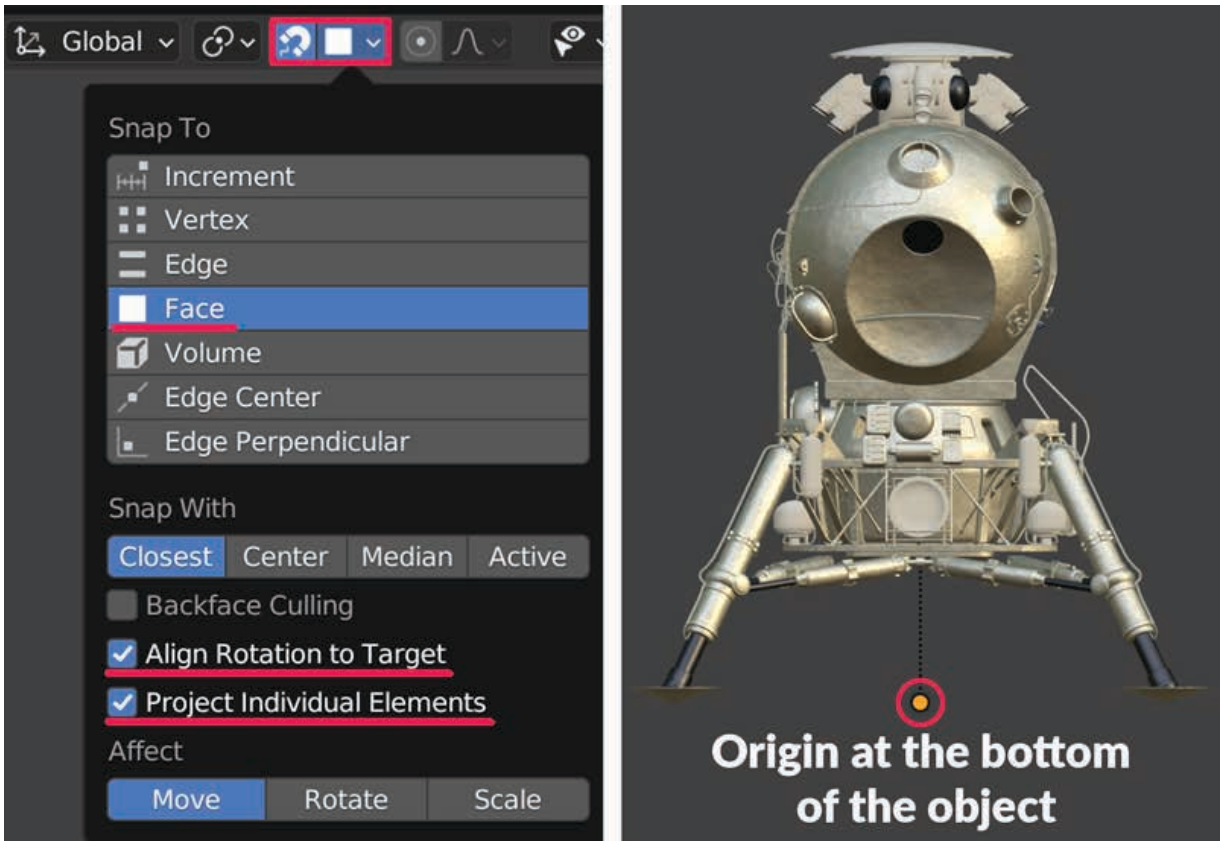
Snapping and Origin transform shortcuts

You probably know that you can turn **snapping** on from the viewport menu. But did you know that you can also just **hold Ctrl** while moving an object to snap it to the surface? When you let go of Ctrl the snapping is turned off again.

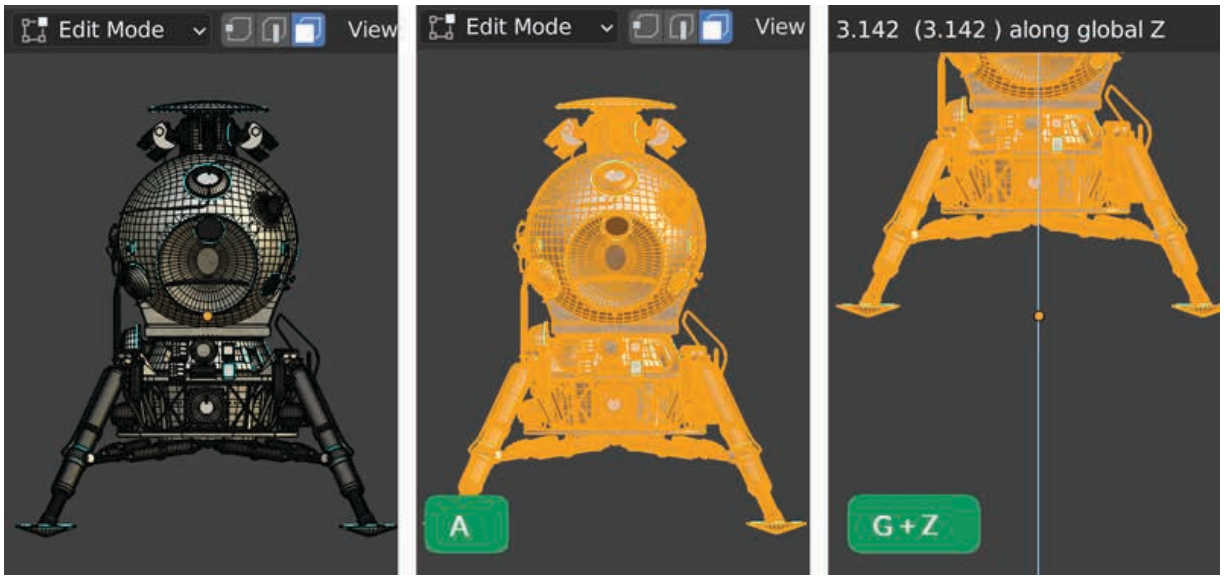


Lunar Lander model by Shontoloyo on Blendswap.

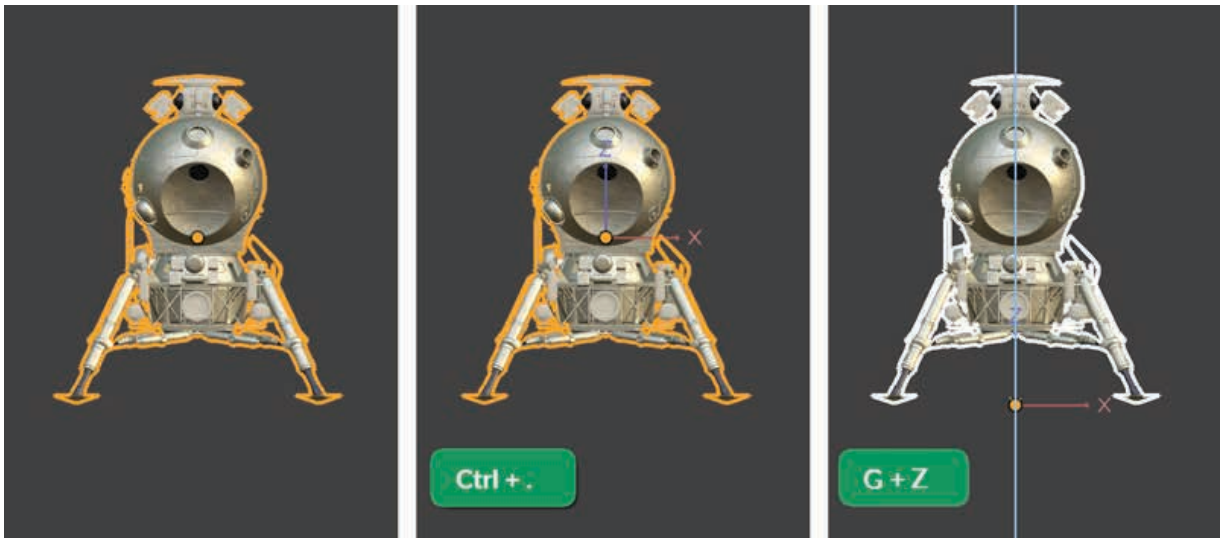
It works best when snapping is set to Faces, with Align Rotation and Project turned on. The location of the origin point is also important when snapping an object to a surface.



You can move the **Origin point** in Edit Mode either by selecting everything and moving the mesh so the origin point ends up in a different place.

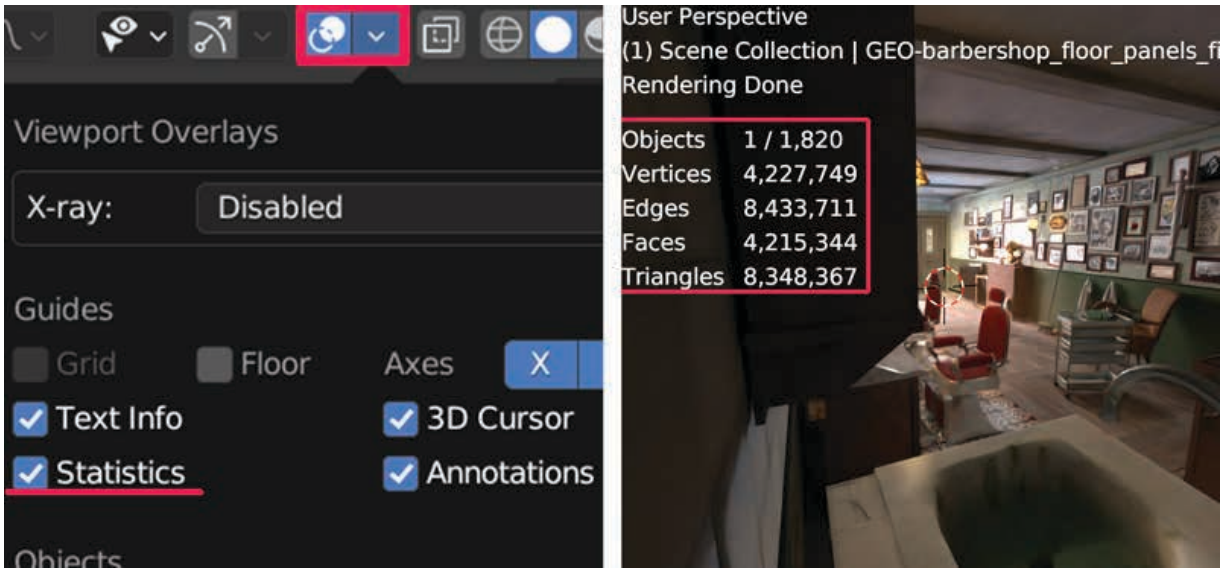


Or in Object Mode **press Ctrl and .** (period). then **press G** to move just the origin.



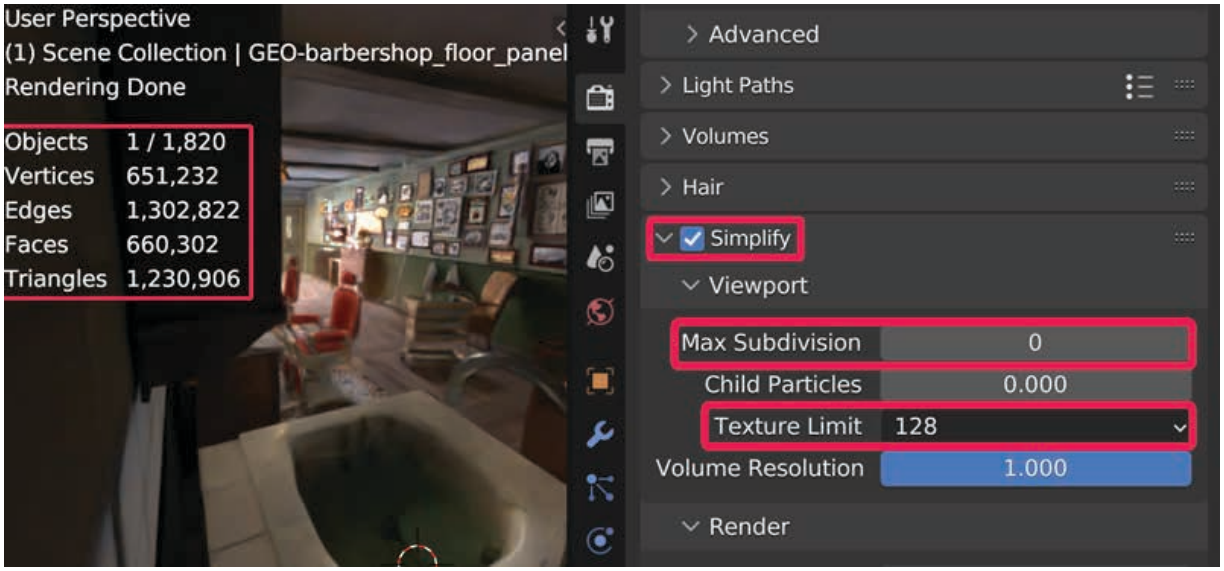
Using Simplify, Camera Culling and Clipping Regions

When working with heavy scenes that feature many subdivided objects and particles, even the best computers can have trouble showing everything in the viewport. Using Simplify, Culling and Clipping Regions to speed up your viewport is necessary and helps to keep it fun to work on your scene. First of all: if you turn on Statistics in the Viewport Overlays, you can see just how heavy your scene is.

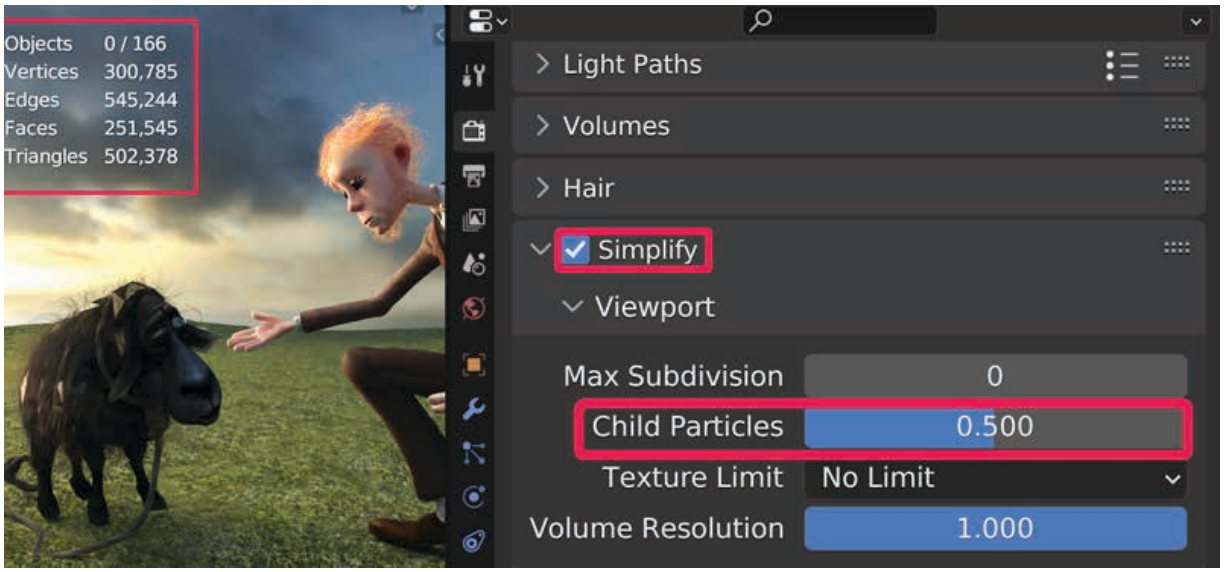


Demo files by Blender Foundation can be found here.

Simplify makes complex scenes lighter and easier to work with without affecting the render quality. To use it, in the **Scene** tab turn on **Simplify** and set Viewport > **Max Subdivision** to zero, to turn off all subdiv modifiers in the viewport. In Cycles you can also set a **Texture Limit**, so your texture is automatically reduced to a lower size until render time.

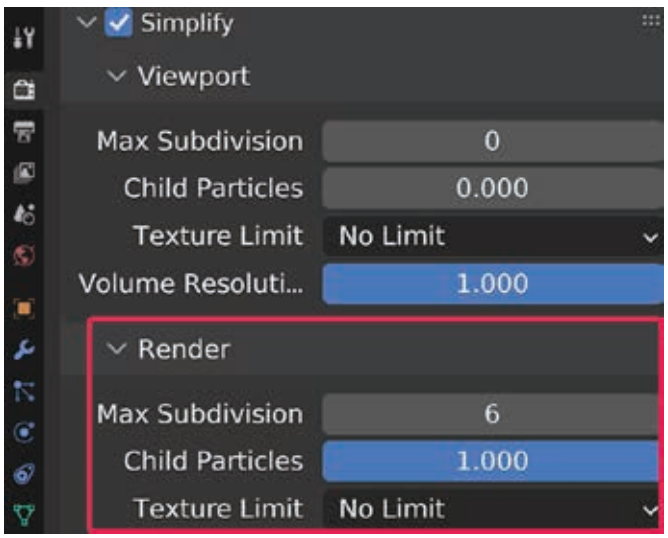


You can also limit the amount of **Child Particles** that are displayed in case you are using a particle system for grass for example.

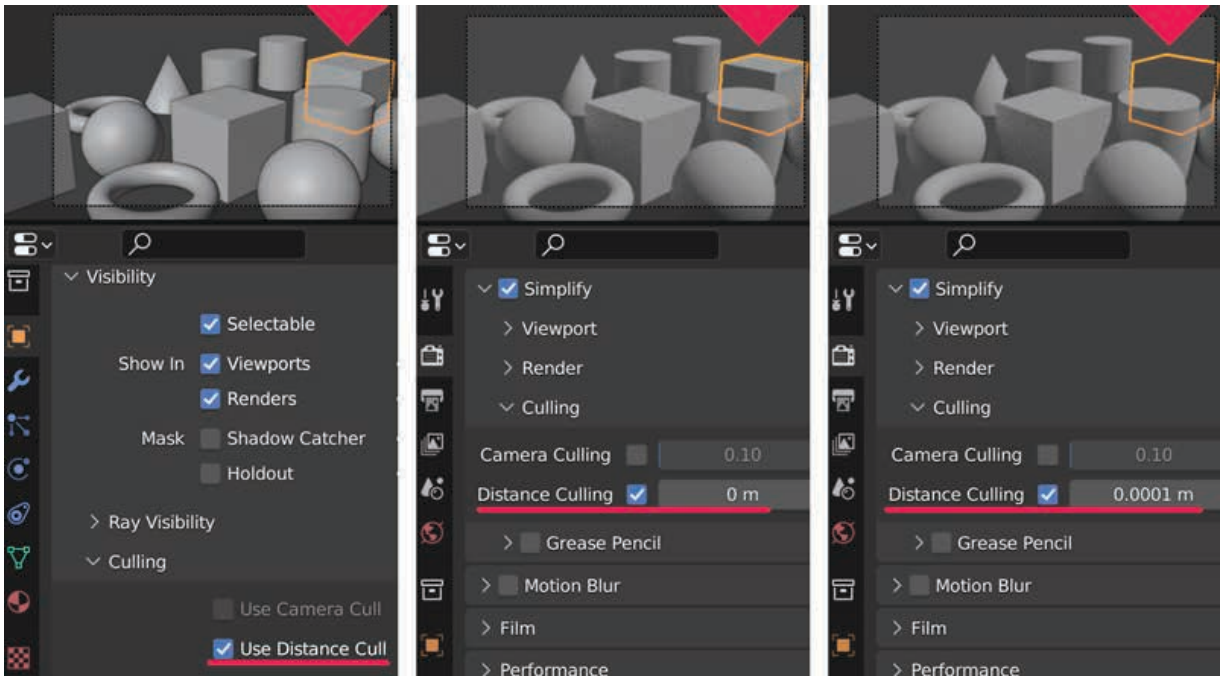


Here you can see the difference (in the Viewport) when using less Child Particles. Note, that this doesn't have an effect on the final render. It just makes working in your Viewport easier and faster.

You can also decrease render times by lowering the Render values although this will be at the cost of render quality.

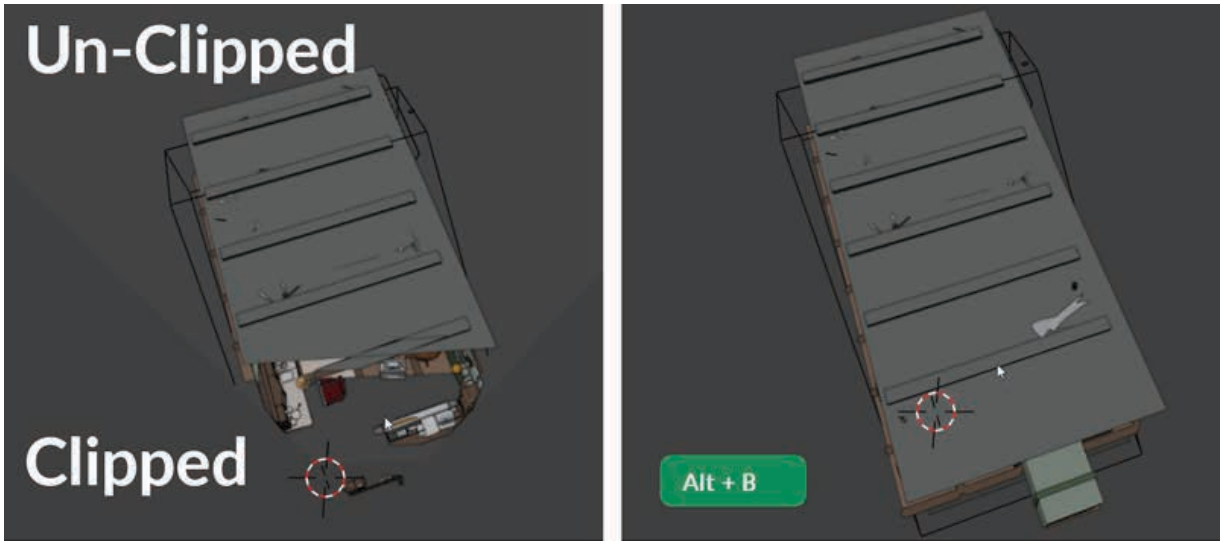


The **Cull** settings simplify the scene by hiding objects that are outside of a certain camera distance, but the way it works is not very intuitive.



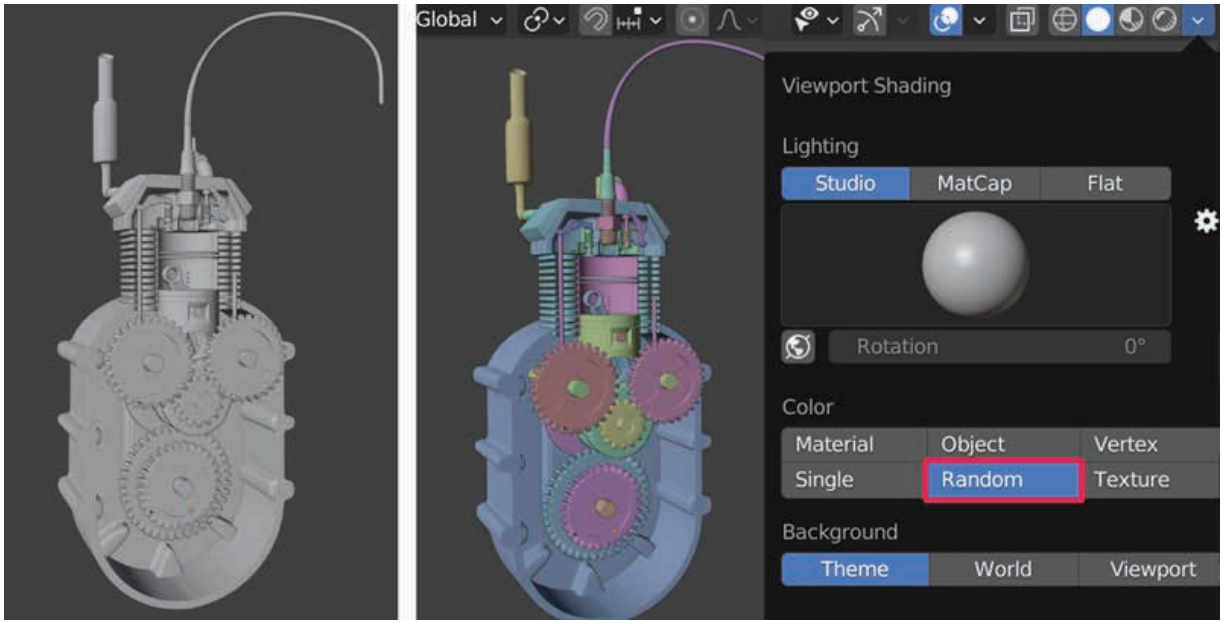
An easier alternative for culling is to go to your camera view and **press Alt+B** to select a clipping region so only the things in camera view are visible.

Press Alt+B again to unhide.



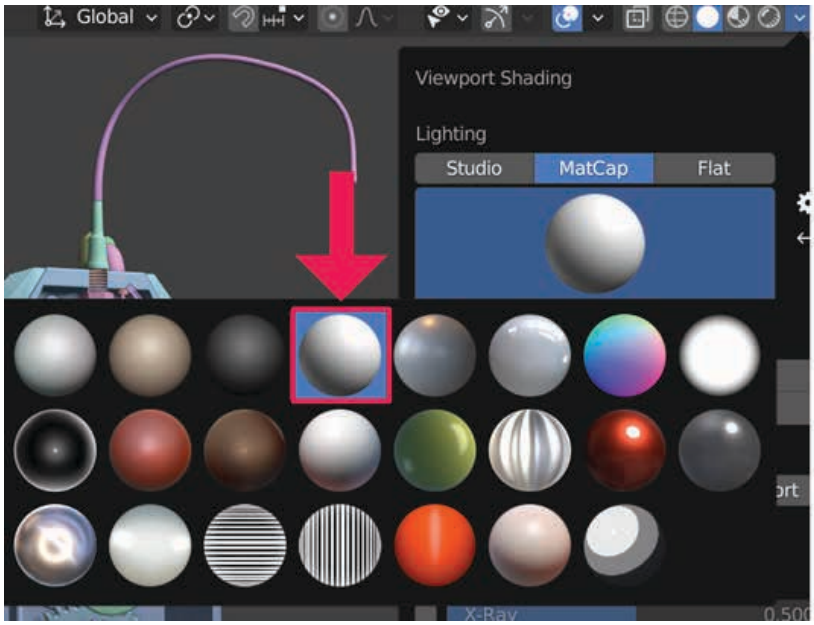
High Quality Eevee Viewport by Default

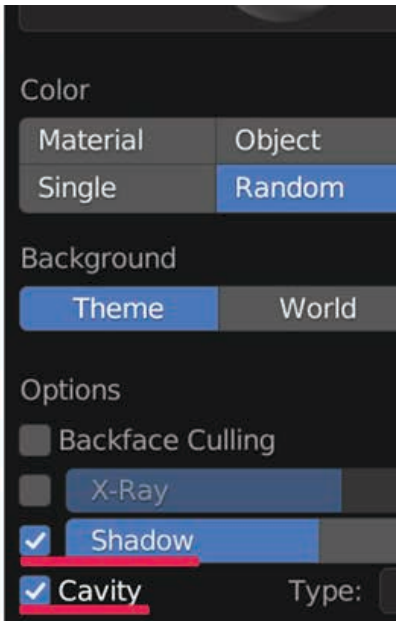
By default everything in Blender is rather gray. It can be tiring for you eyes to work like this so let's make the viewport more easy to look at. First, enable random colors. **Click on Random** in the **Viewport Shading** menu.



Mechanism modeled by Blendswap-user shontoloyo.

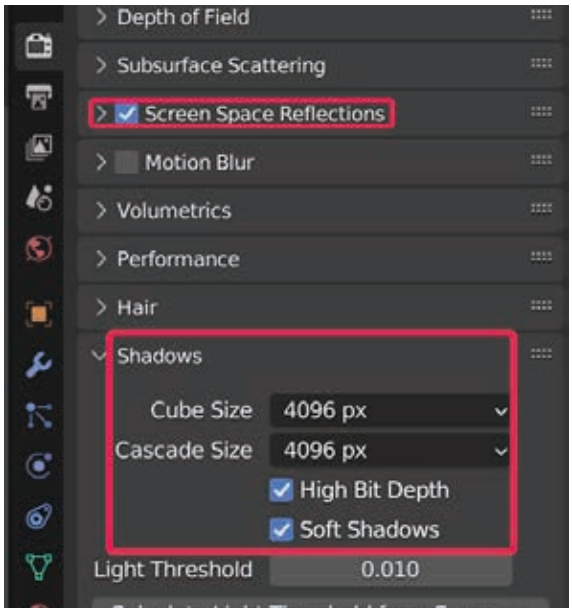
Next click on the **Matcap** sphere to select one. I choose this white one as I can still see the random colors that way. In the same menu, check the box for **Cavity** as well as **Shadows**. This all makes it easier to see what you are doing. Not just for mechanical parts but also with organic shapes, especially when you are sculpting for example.

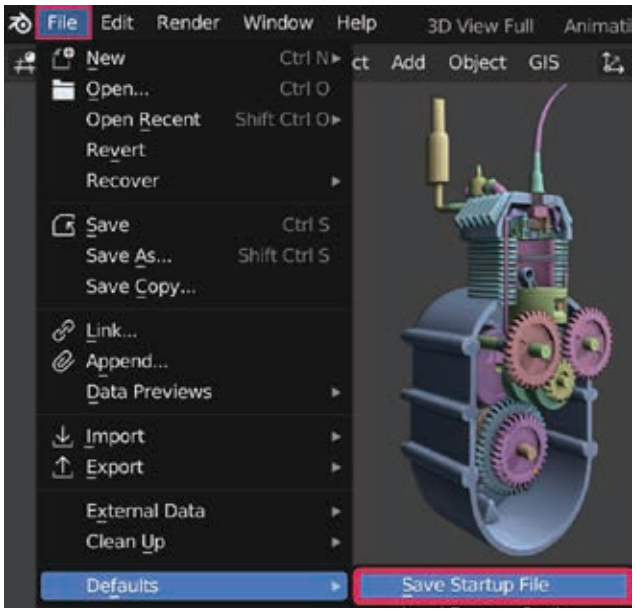




Another thing you can do is turn on **Screen Space Reflections** and increase The shadow quality.

If you want this to be your default every time Blender loads, simply go to File > **Defaults > Save Startup File**.

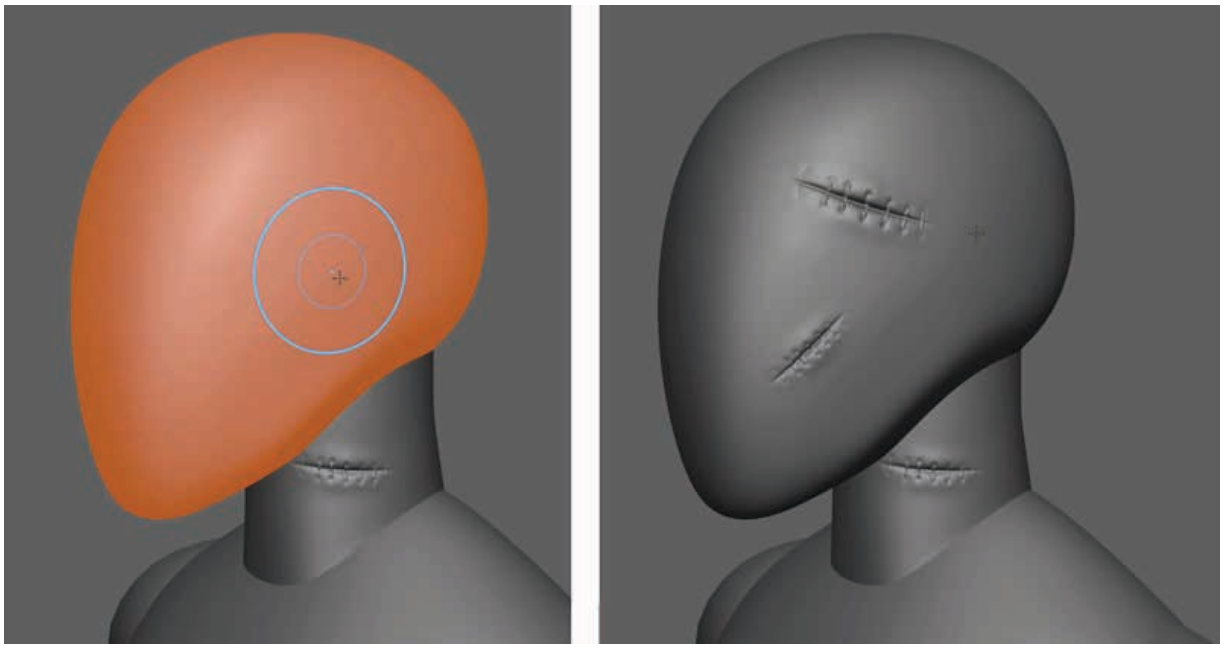




INTERFACE TIPS

Quickly Switch Objects While Sculpting and Texture Painting

When **Sculpting**, sometimes you need to switch from one object to another. Instead of switching to Object Mode to select the other object, hover the mouse over the object you want to work on and **press Alt+Q**. The selected object will momentarily light up to indicate being selected. After that you can sculpt on it.

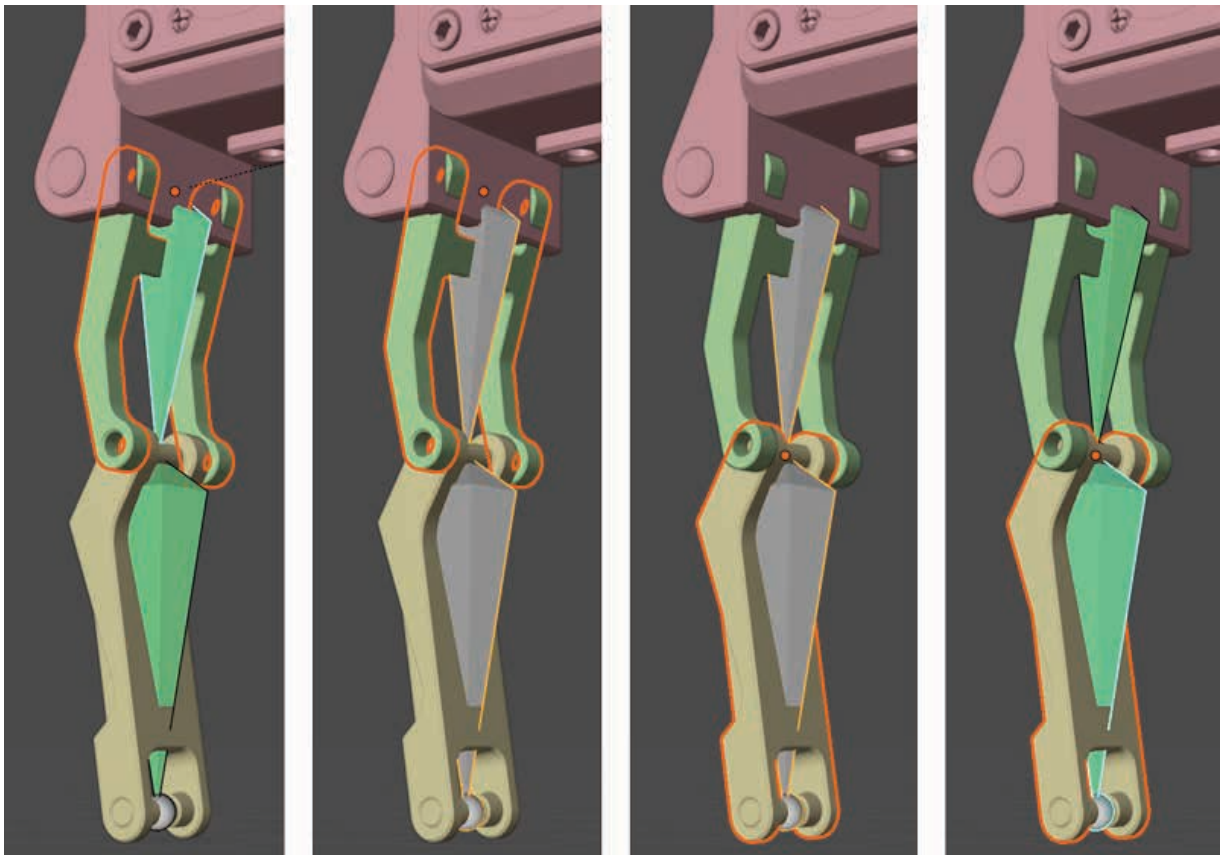


Selecting the head to sculpt on it. Model by VeryHotShark on Gumroad

This shortcut works in every mode, so you can use it while sculpting, texture painting and so on.

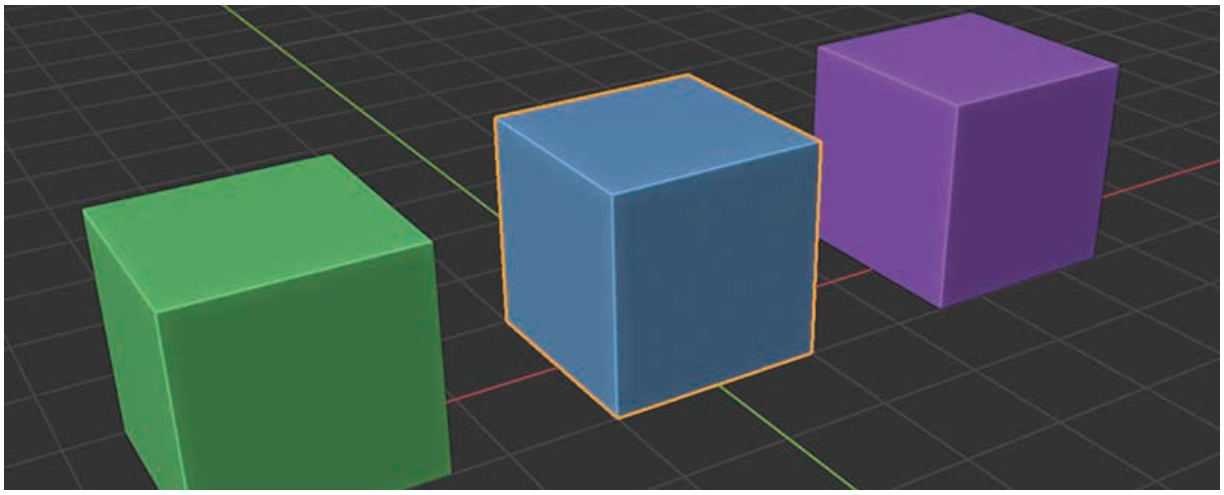
Selecting the handle to paint on. Model by MrLiang on Sketchfab

Only **Pose Mode** works slightly differently. To quickly switch between Pose and Object mode, **press Ctrl+Tab**, to choose a different object to parent to a bone for example. Note that you must have both the Armature and the Object selected for this toggling to work.

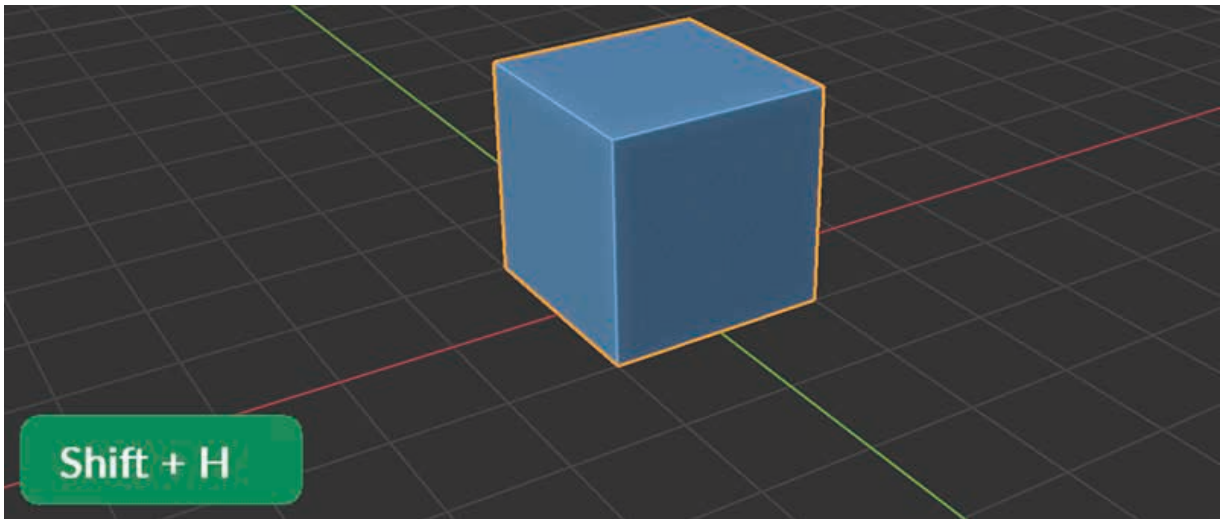


3 ways to isolate objects

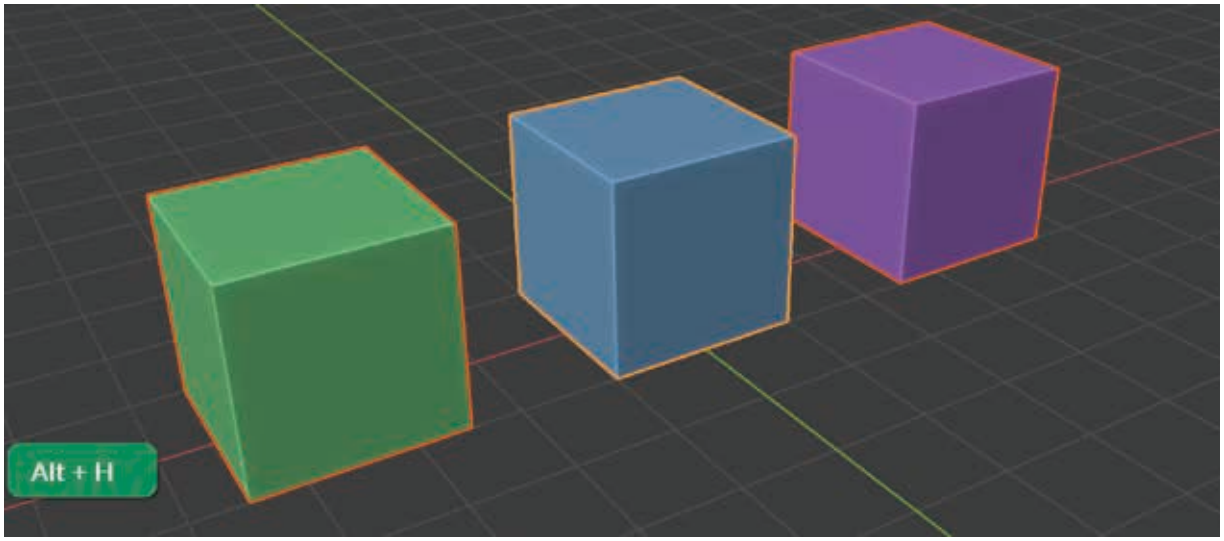
If you need to focus on just one object, there's a couple of things you can do.



With the object you want to work on selected, **press Shift+H** to hide everything else.

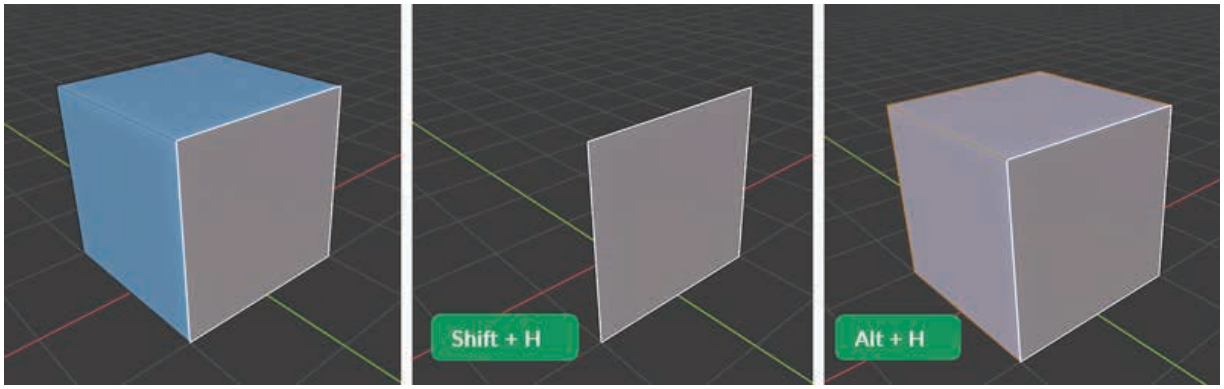


Pressing Alt+H unhides them again.

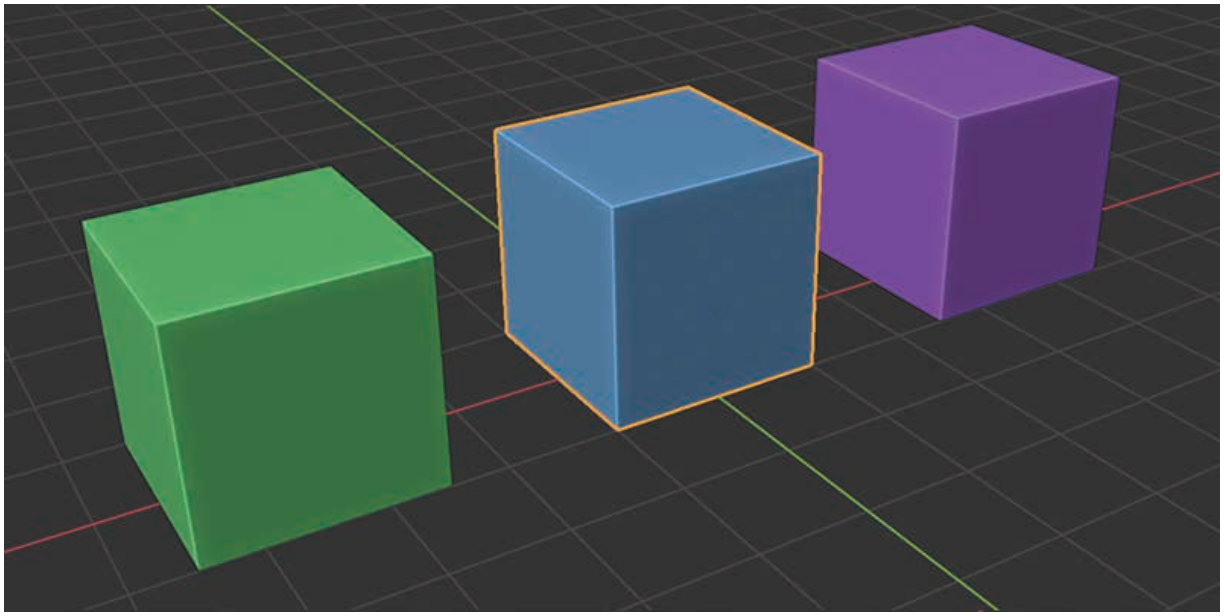


Click here to return to the index.

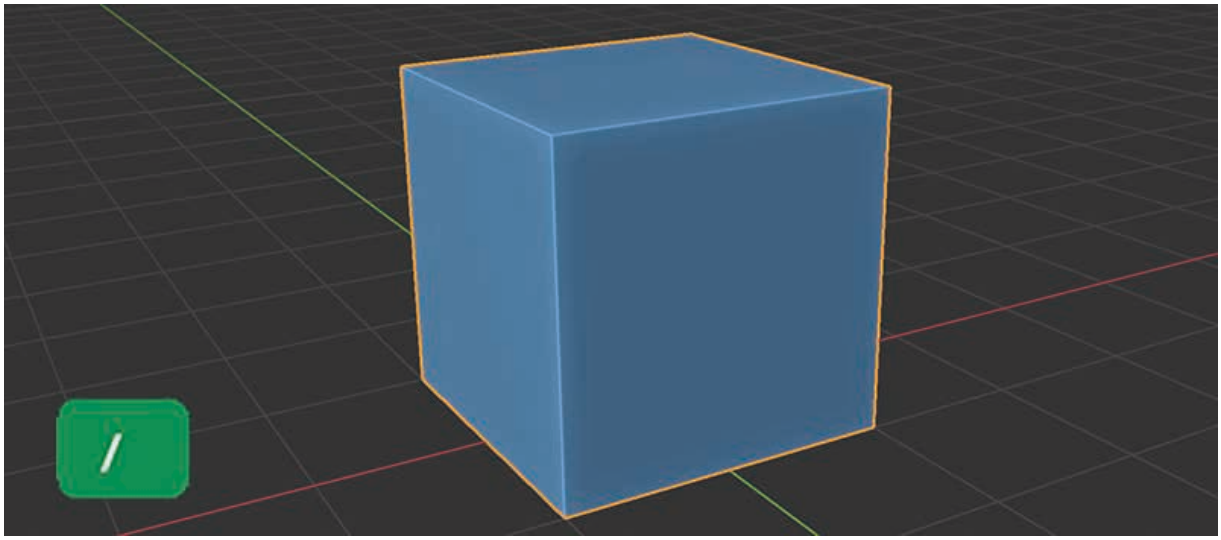
This also works in Edit Mode.



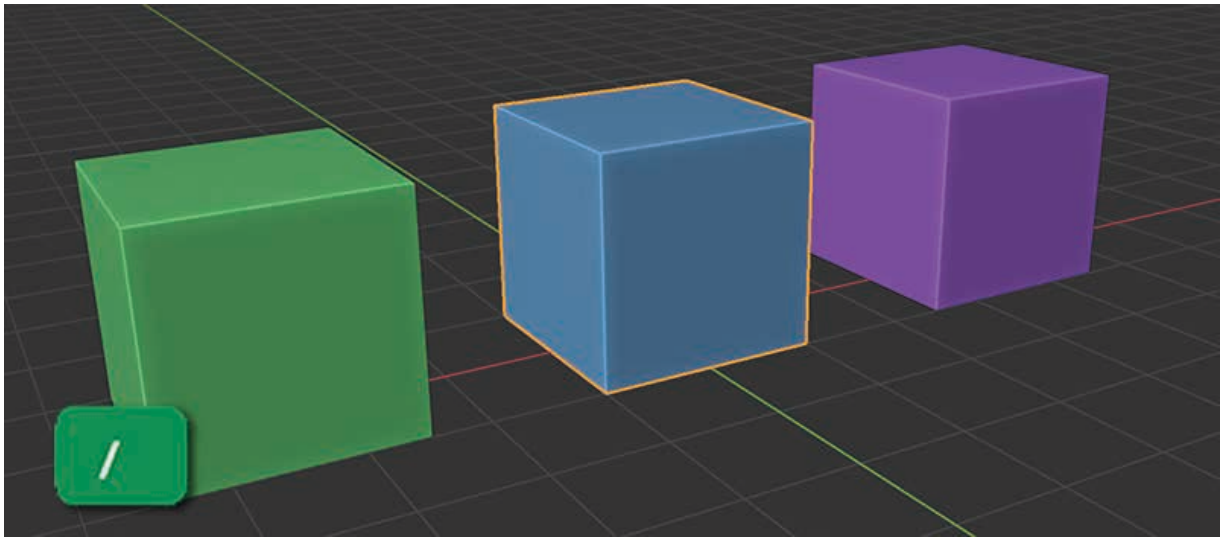
Another thing you can do to isolate an object is **press /** (forward slash).



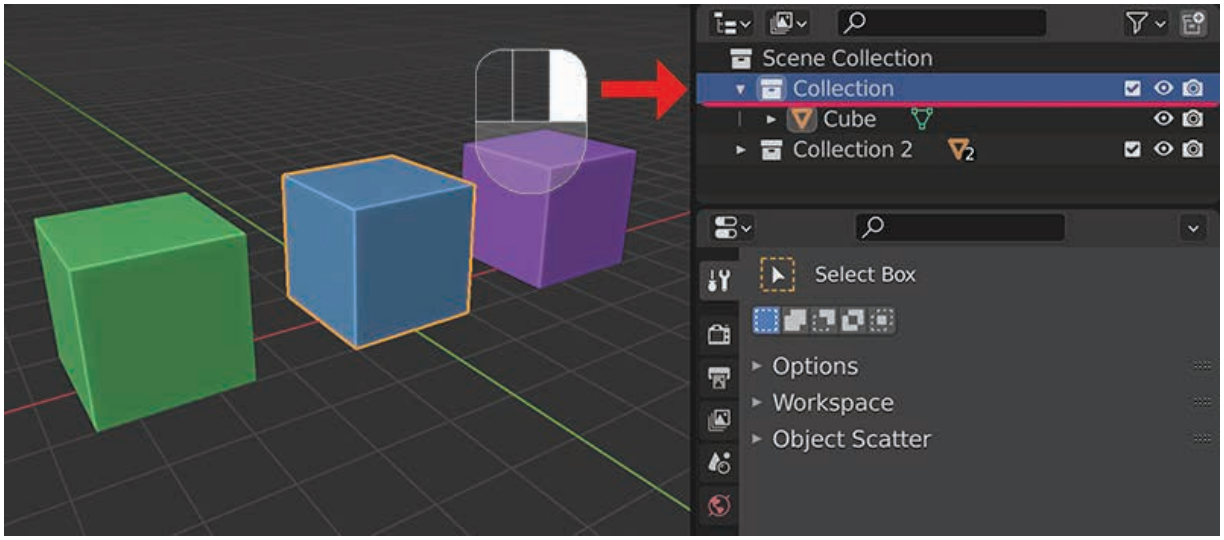
That will isolate the selected object and zoom in on it as well.



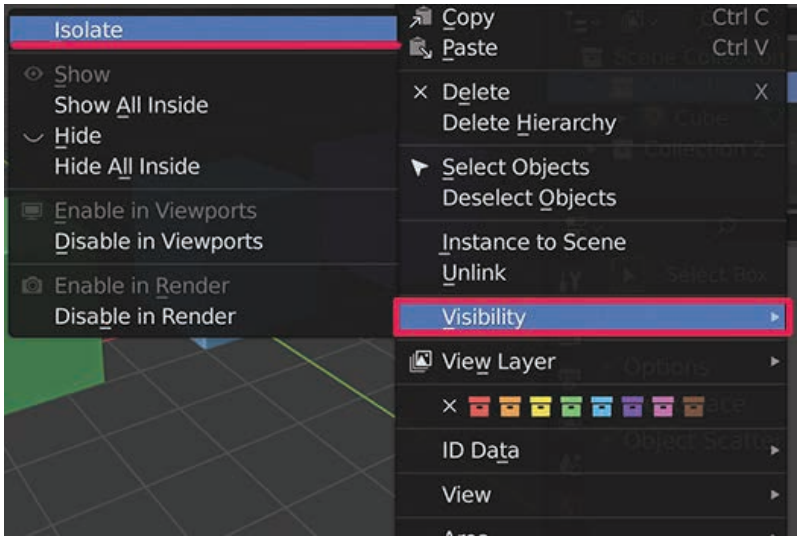
Press / (forward slash) again and the view will go back to what it was before.

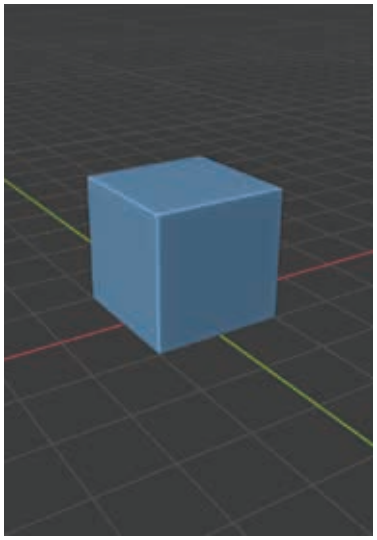


Finally, if your object is in a **Collection**, you can right-click on it in the **Outliner**.



Then go to **Visibility > Isolate**.

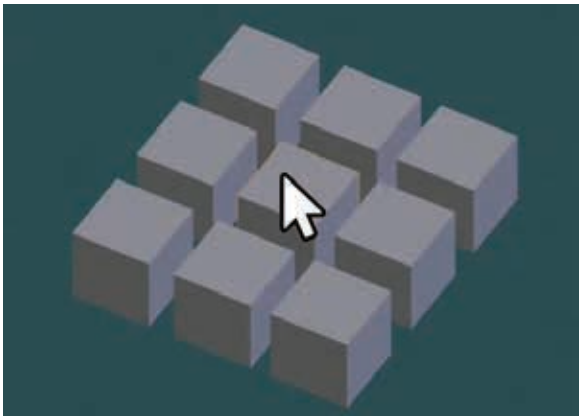




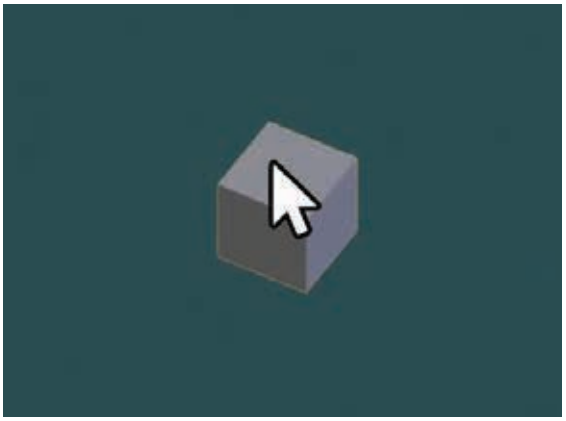
Hide Everything Except Selected

Instead of selecting all the things you want to hide and **pressing H**, you can select the one thing you don't want to hide, and **press Shift+H**.

Select the middle cube and **press Shift+H**.



The surrounding cubes are hidden. **Press Alt+H** to unhide everything.



Isolation Mode

Press / (forward slash) on the numpad to isolate the currently selected object.

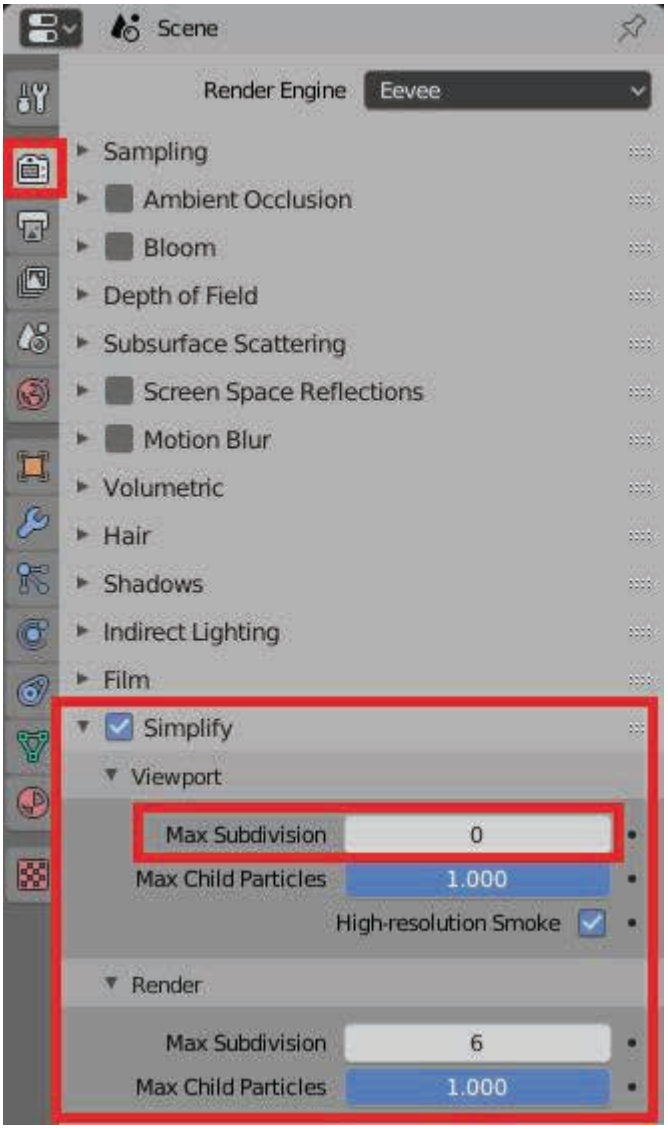
Isolation Mode will hide everything else and zoom in on the selected isolated object. It's great to quickly do some work on a little part of a complex project, as an alternative to hiding everything else by **pressing Shift+H**, because it also zooms in on the object and then zooms out again when you don't need it anymore.

Robot from Blendswap, © Blender Foundation - Tears of Steel

Simplify Setting

Some scenes can be very heavy, with a lot of detail slowing down your viewport. It's not fun to work with that kind of scene because it is very unresponsive. Turning off subdivision modifiers one by one unfortunately is a big waste of time.

Luckily, there's a single box we can check to turn off the subdivisions globally for all objects! The Simplify option is found in the Render tab. Check that box! Then set the maximum amount of Subdivision you want to see in the viewport. You can turn this all the way down to 0, as long as you don't change the maximum Render subdivisions the render will still have all the subdivisions your scene really needs.

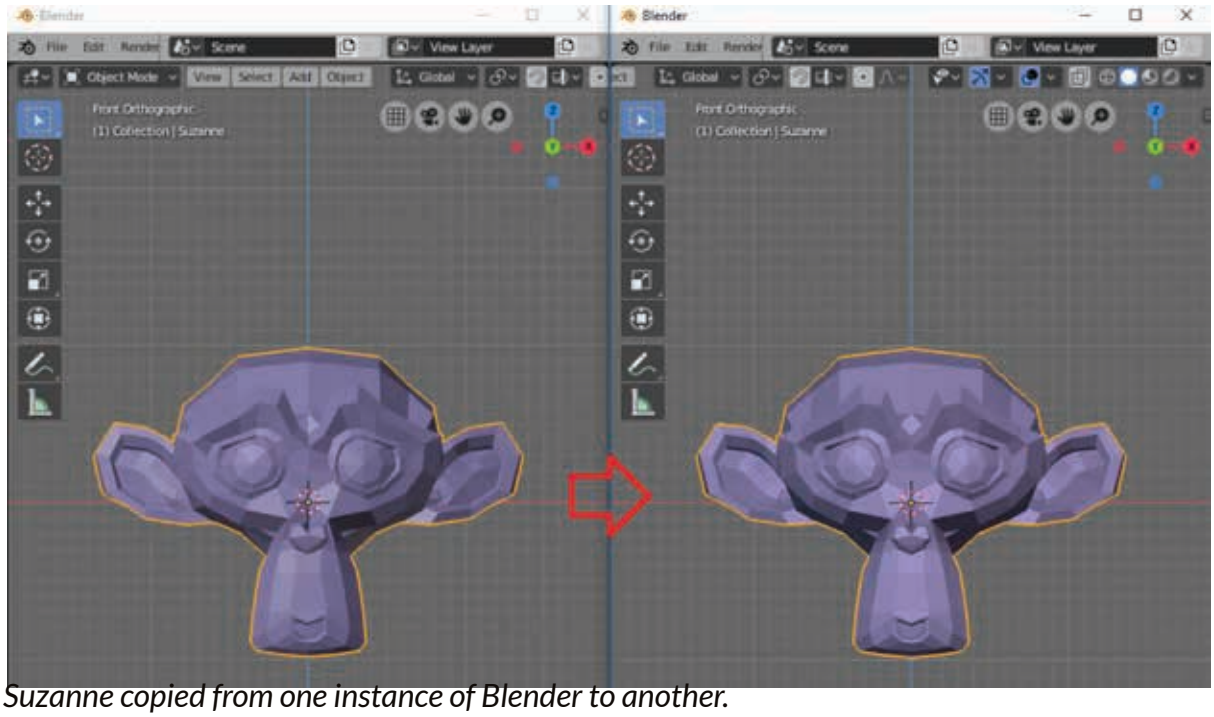


As you can see, you can also change the amount of Child particles displayed (from a Particle system), and you can influence how detailed smoke is displayed in the viewport. All of this will save memory and make your work easier.

Copy Objects From One Blender Window To Another

You can have two separate Blender files open and copy objects from one to the other.

To do this, copy from one window with **Ctrl+C** and paste it in the other window with **Ctrl+V**.



Suzanne copied from one instance of Blender to another.

(Note: On Apple computers, running two instances of Blender was previously not possible, although recently this seems to have changed).

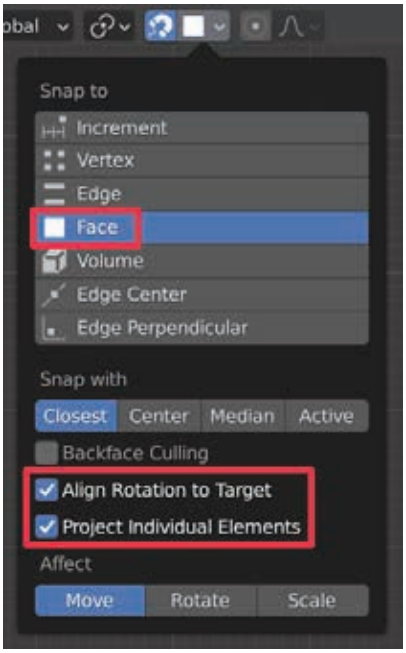
Hold CTRL To Snap Objects

To place the tree on the surface easily, **hold Ctrl** while moving the object to snap it to thesurface. **Holding Ctrl** in other words,temporarily turns snapping on.

This is useful for placing for example this tree on the ground.

Release Ctrl go to turn off the snapping. Without snapping, the tree can go “through” the surface.

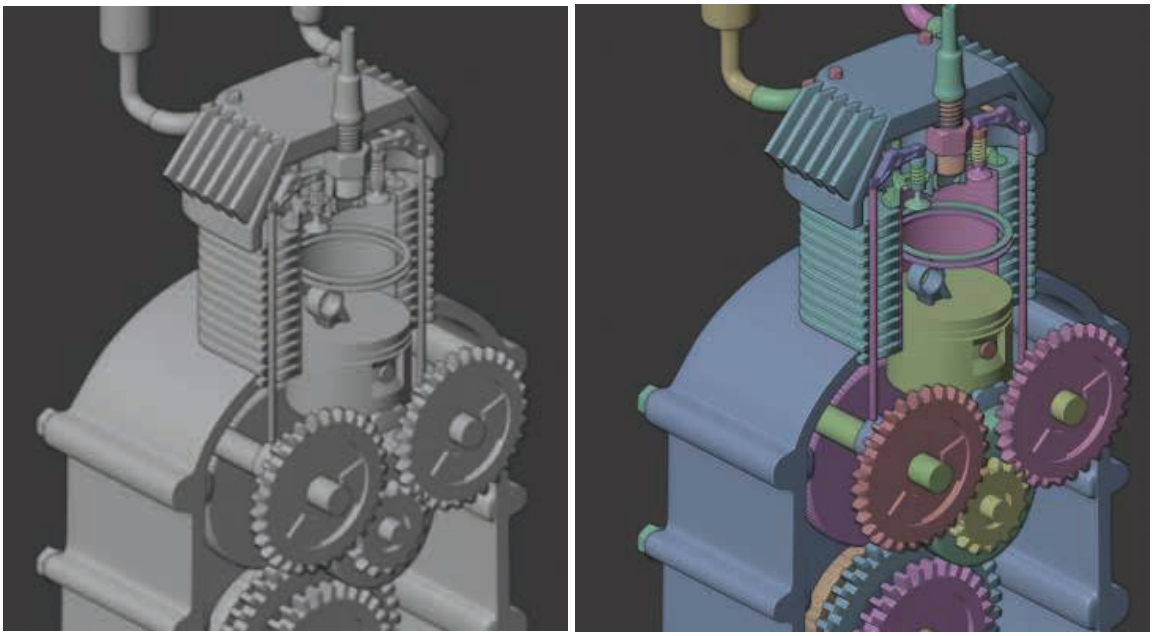
Make sure Snapping is set to **Face**, and that **Align Rotation to Target** and **Project Individual Elements** are turned on in the Snapping menu for best results.



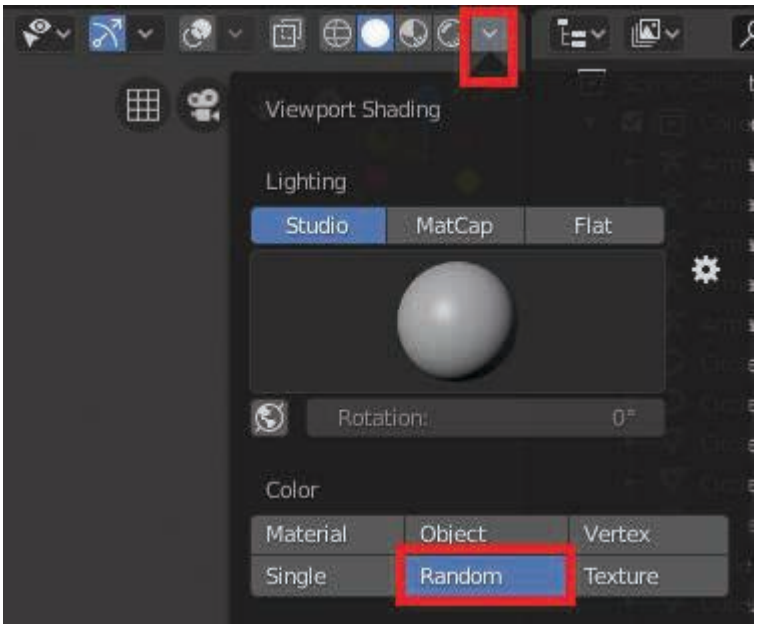
Click here to return to the index.

Viewport Visibility Tips

This is a thing I often do when working on a scene with many objects, or an object that has many parts. I will give different objects different Viewport Colors. This has nothing to do with the render color. It just makes things more clear in the viewport so your eyes don't get tired.

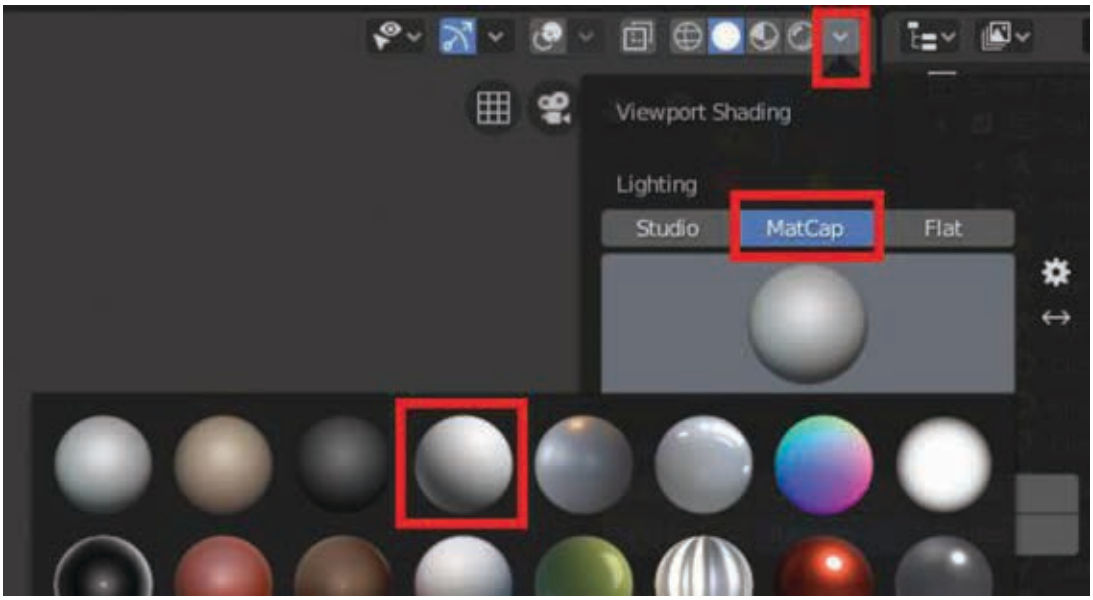


Left: the default "everything grey" display mode. Right: with random viewport colors enabled.
To enable these random colors, simply click on Random in the Viewport Shading menu.

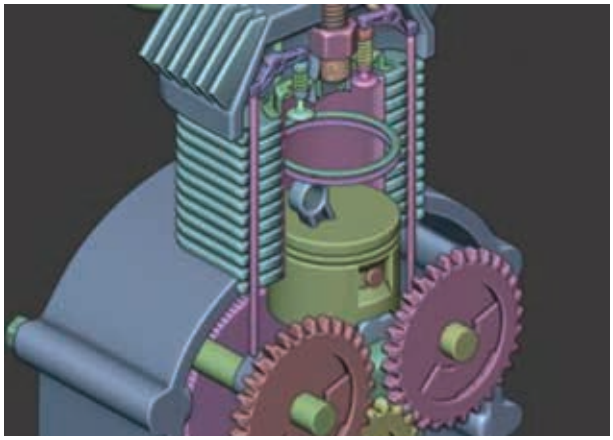


Mechanism modeled by Blendswap-user shontoloyo. Download it here: https://www.blendswap.com/blends/view/72354

Another good trick is to use a Matcap. Click on the Matcap sphere to select one. I choose this white one as I can still see the random colors with it.



In the same menu, check the box for "Cavity". This is similar to "Ambient Occlusion" in older versions of Blender. This adds shadow in creases as well as highlights on edges.



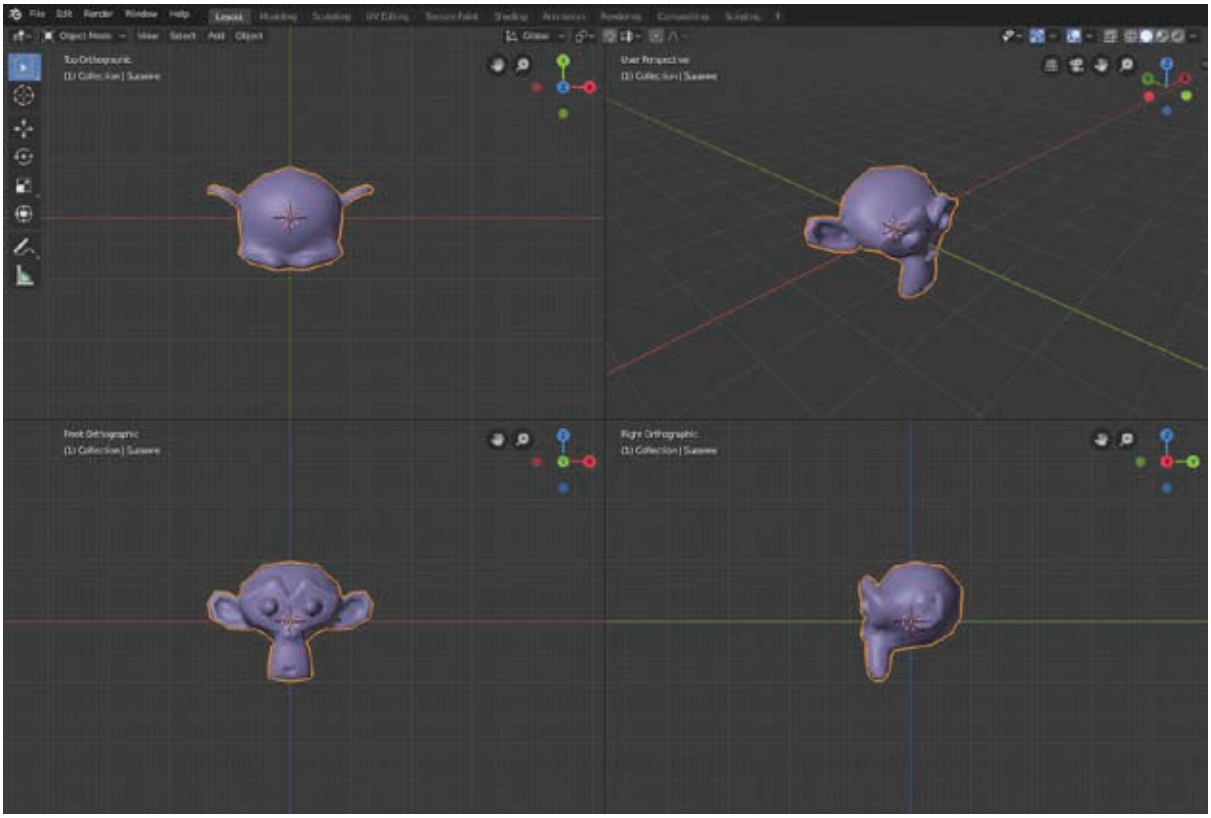
The combination of Matcaps and Cavity makes it easier to see what you are doing. Not just with mechanical parts but also with organic shapes, especially when you are sculpting for example.

Quad View Toggle

Quad View is a convenient viewport layout with 4 windows. Generally it's one perspective view and three orthographic views - top, front and one of the side views.

When doing box modeling, especially using reference that has been prepared in advance to show different sides of a model, this can be a very useful layout as it shows you what you are doing from the most important angles of the model.

To toggle the "quad view" layout, **press Ctrl+Alt+Q**.



Zoom To Selected

Getting closer to an object in Blender can be done in different ways. Sometimes you just need to quickly zoom in on a far away object or part of a scene.

One way to do this would be to select the object and **press "."** (period key on the numpad), but it's not always easy to select stuff that's far away, and you might select the wrong object.

By **pressing Shift+B** you can drag a bounding box around an object and the viewport will quickly zoom in on the area you have selected.

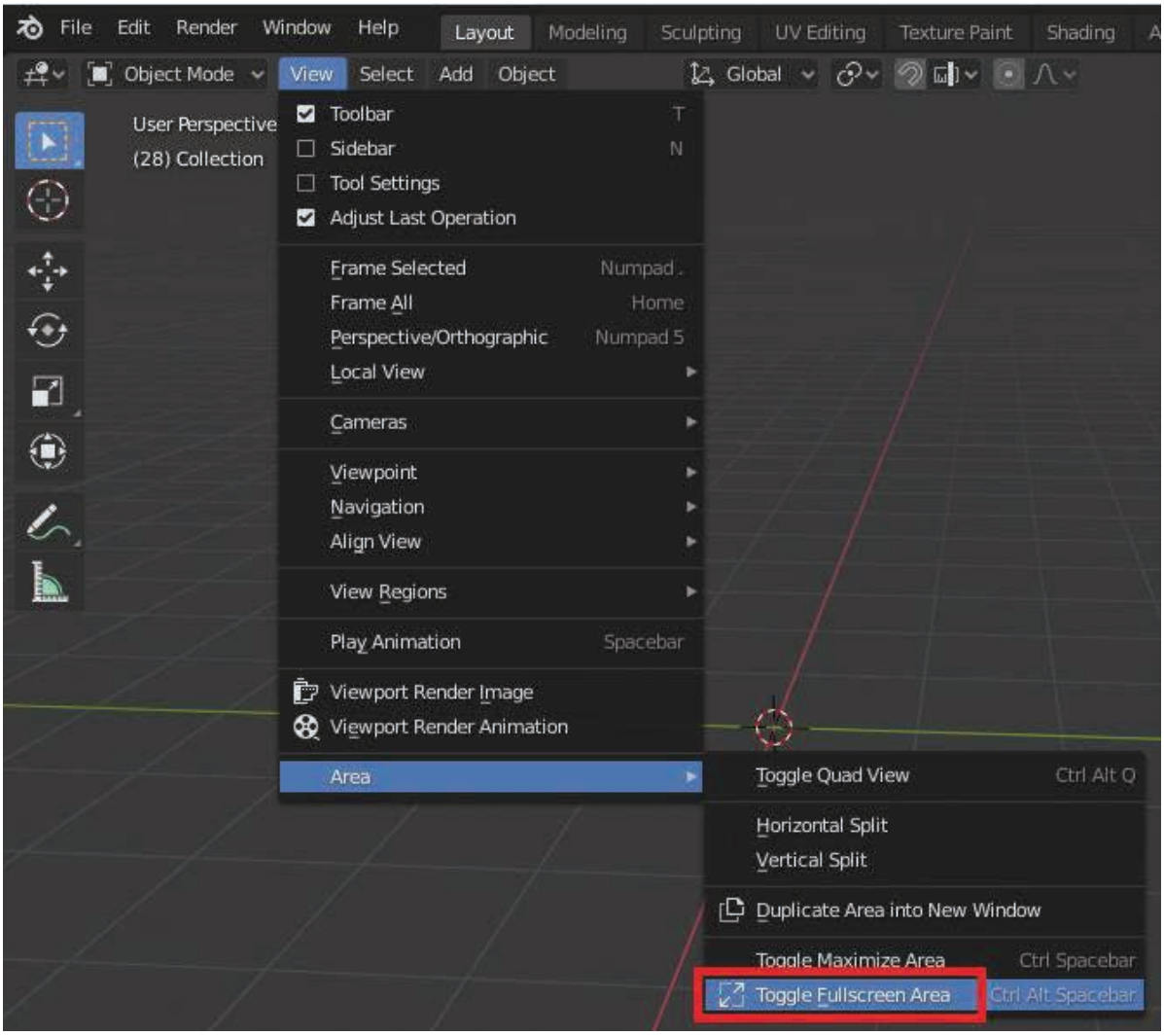
Whoosh! We're quickly brought close to the object.

Toggle Fullscreen or Maximize Area

To toggle Fullscreen mode **press Ctrl+Alt+Space** in Blender 2.8. This makes the window your mouse is currently on fullscreen until you **press Ctrl+Alt+Space** again. Quite useful if you've only got one screen with a lot of things going on (Shader Editor, Compositor, Timeline etc). That way you can keep those windows small and just expand them temporarily when needed by making them full screen.

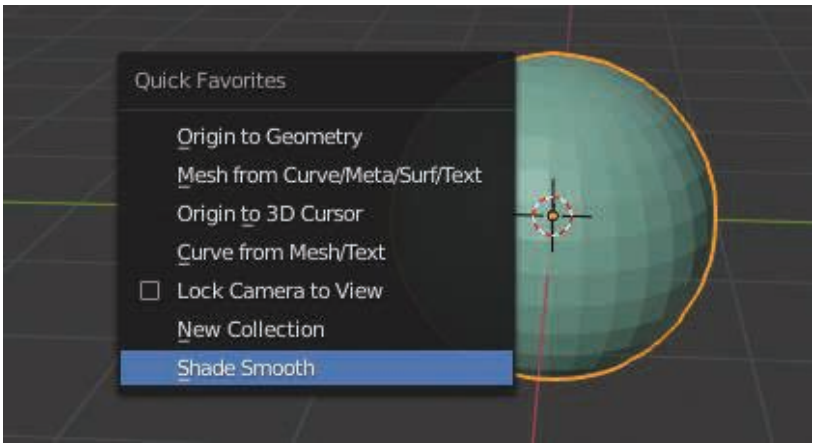
The downside is that it's quite an uncomfortable and long keyboard shortcut. Instead of Full Screen you can also just **press Ctrl+Spacebar** to "maximize" the window the mouse is currently on. It's pretty similar to Fullscreen, but you can still see the top header bar.

In case you want to add Maximize or Full Screen to your **Quick Favorites**, you can find them under View > Area > **Toggle Fullscreen Area** (just right-click and choose "Add to Quick Favorites".



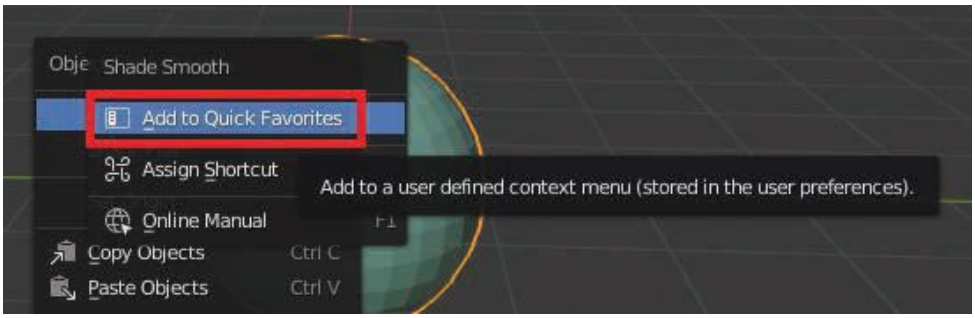
Quick Favorites

A great feature we've had since Blender 2.8 is the "Quick Favorites" menu. This is a custom menu you can bring up by **pressing Q**.



My current Quick Favorites. Yours can be whatever you want.

To use Quick Favorites, **right-click** on a menu item you want to add - something you use often, like "Shade Smooth" or "Subdivide" for example. Then pick "Add to Quick Favorites".

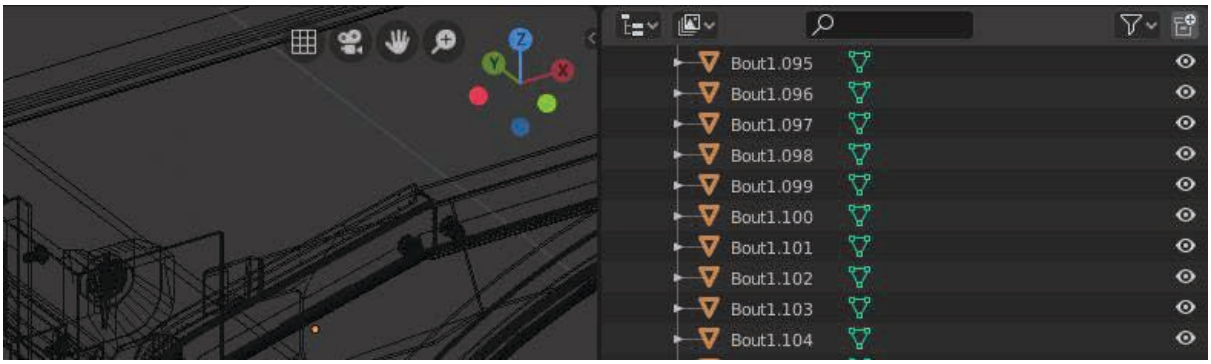


Now when you **press Q** the Quick Favorites menu will pop up and you'll find your menu items there.

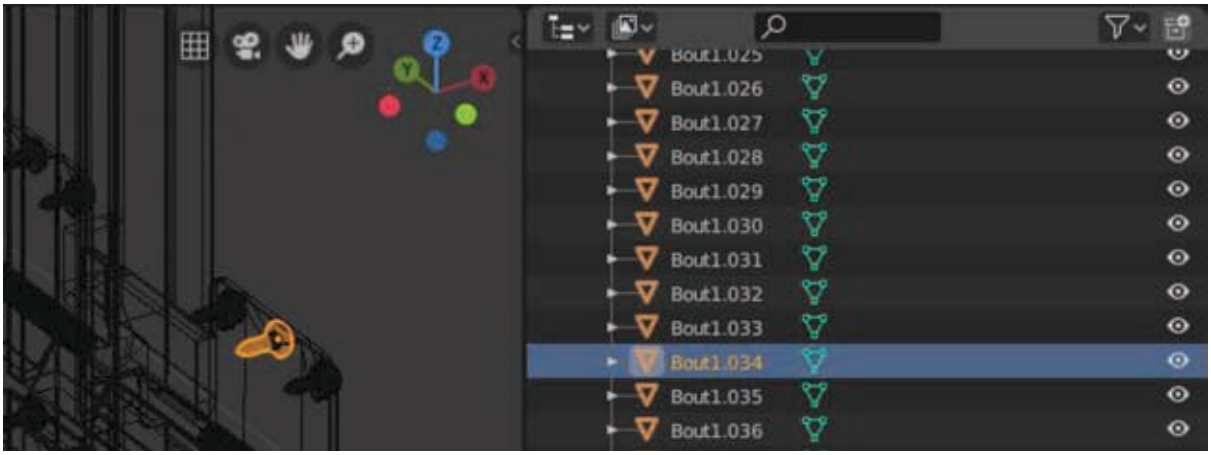
Very useful, especially since in 2.8 a lot of the old familiar keyboard shortcuts from 2.7 are gone. Plus not having to remember all those keyboard shortcuts can only be a good thing.

Quickly Find Objects In The Outliner

Sometimes you need to turn an objects render or viewport visibility off. But what if there are hundreds of similarly-named objects in the Outliner? How do you find it quickly without scrolling down that long list?

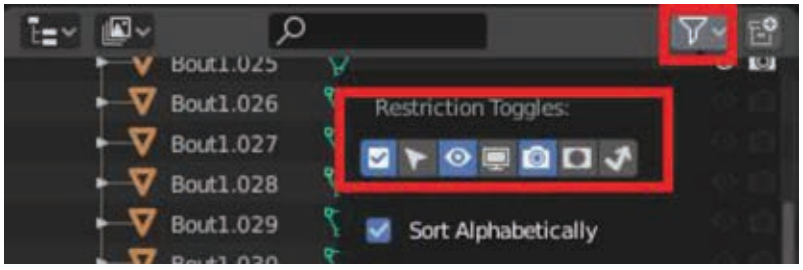


Here's how. First select it in the viewport. Then hover the mouse over the Outliner, **press the period key "."** and the Outliner will jump to that object.



This can be quite useful when dealing with many tiny objects in a complex scene, for example a mechanical model with a lot of screws or bolts.

Bonus tip: In case you are wondering where the toggle for Render Visibility is, by default it is now hidden in Blender since version 2.80. But you can find it and other Restriction Toggles by clicking on the filter icon:



Snap to orthographic view

You probably know that the numpad keys switch between views.

1	Front view
3	Side view

Press Ctrl+ one of these numbers to get the opposite view, **Ctrl+1** for Back view for example.

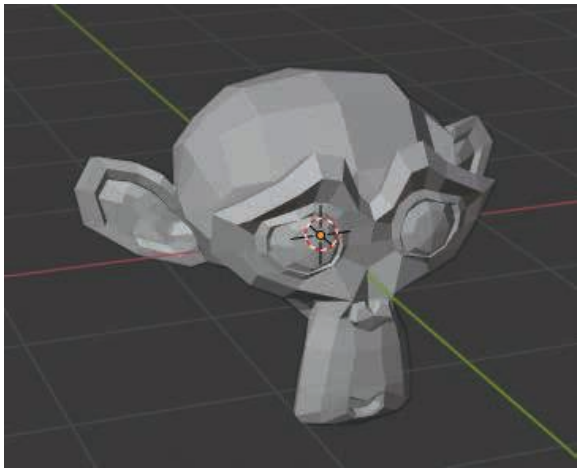
But how about when you are on a laptop without numpad keys?

Not to worry!

You can snap to the orthographic views.

You can do this by **pressing ALT** when rotating the view and getting close to what the orthographic view would be.

This will snap the view to the closest orthographic view (and automatically switch from Perspective mode to Orthographic mode).



If you are using the **Emulate 3-button mouse** input option in Preferences (like I

am) then you already use ALT + left-mouse button to rotate the viewport.

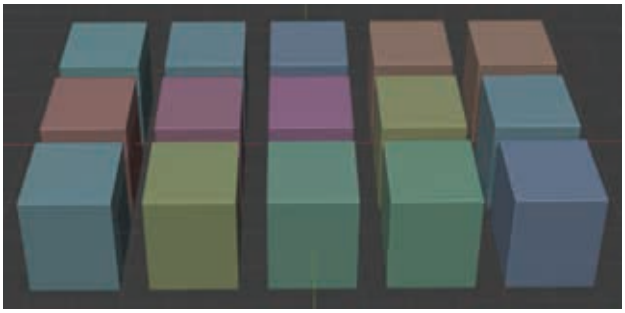
In this case you will need to let go of the ALT button before pressing it again. After that it works as described above.

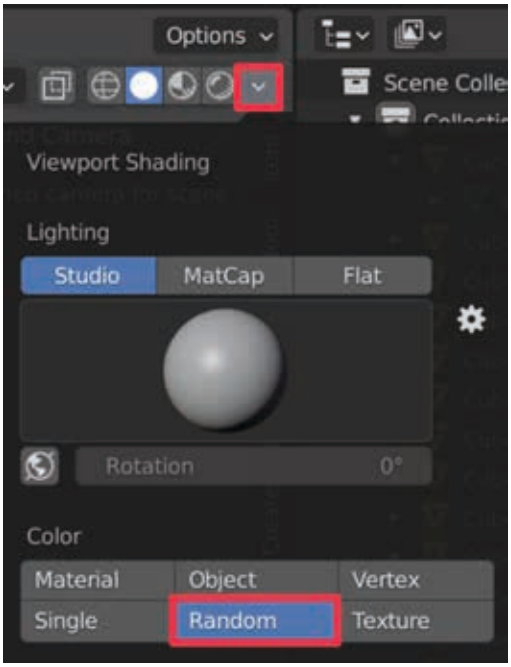
Random Object Colors in the viewport

It's a good idea to give objects different colors in the viewport. This allows you to more easily distinguish all the different parts, especially with a complicated scene.

Since Blender 2.8 you no longer have to do this manually for each object as was the case in previous versions of Blender.

To enable random viewport colors, click on the Viewport Shading drop-down menu and enable "Random".



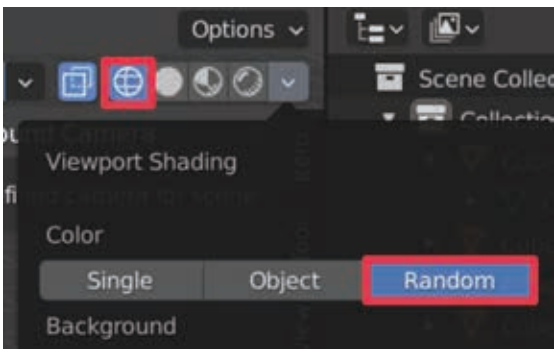


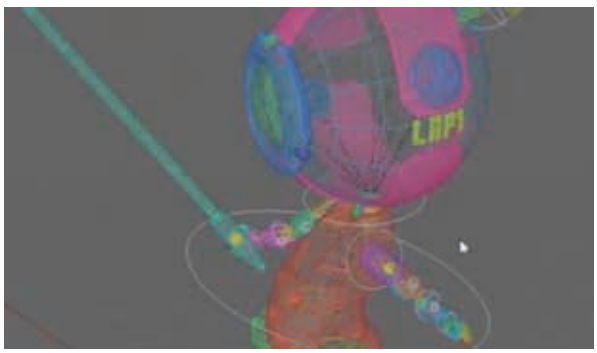
Now each object automatically gets a random color in the viewport. This is certainly much easier for your eyes to read than the dull grey color objects normally have in the viewport. Note, that this has nothing to do with the render colors. It is merely a viewport thing.

Colorful Wireframes

Since Blender 2.8 the viewport has the optionof colored wireframes! Go to Wireframe mode. Go to Shading > Random.

This makes it much clearer to see separate objects. (Robot model by Lucas Falcao, design by David Revoy. Get it from Blendswap.com)

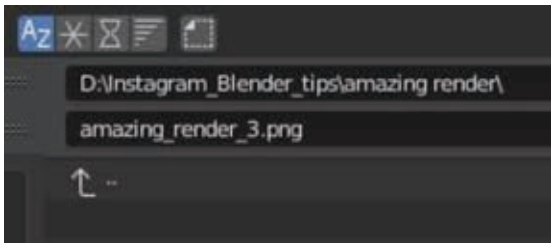




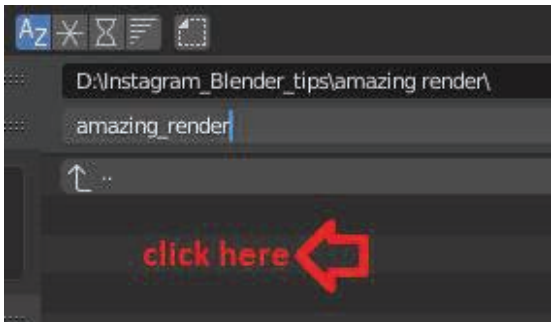
Incrementing your file names

Incrementing file names can be important when you are making different test versions of a render. I used to do this by carefully selecting the number in the filename, deleting that part and typing a new number. Those days are over!

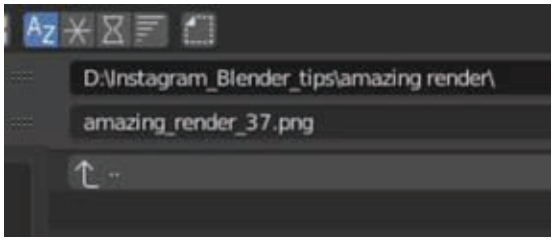
To quickly increment file names when saving, press the + or - keys on the numpad while holding the mouse cursor over the filename.



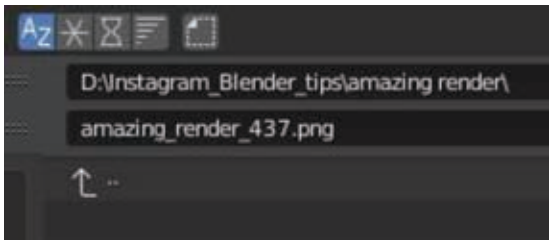
If it doesn't work - click in the field below the filename until you see the filename extension like .PNG appear



Hold Shift when pressing + to increase numbers by 10.



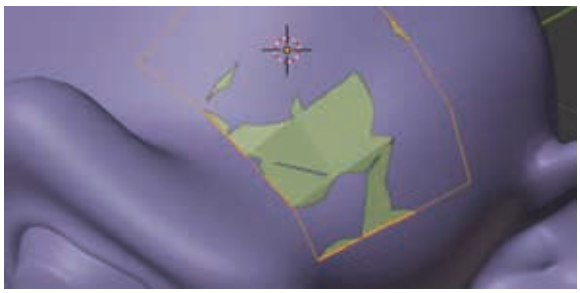
Hold Ctrl when **pressing +** to increase numbers by 100.



Click here to return to the index.

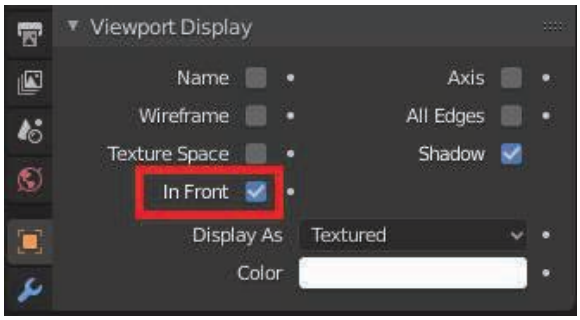
Retopology visibility tips

When you retopo by snapping to faces, the meshes intersect.

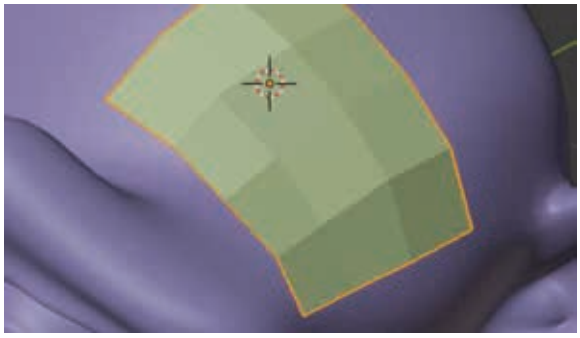


That makes it difficult to see what you're doing.

Turn on **In Front** in the Object tab's Viewport Display options.



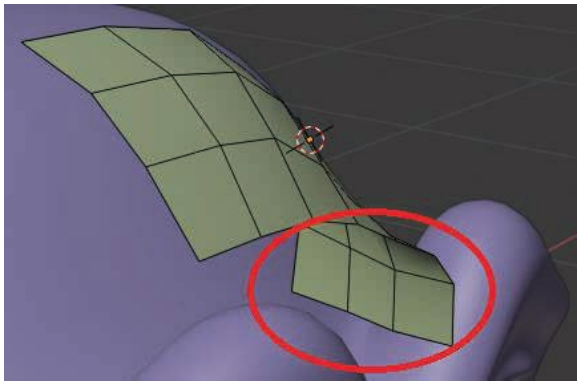
Now you can see the retopo mesh much better.



Right-click and turn on **Smooth Shading**.

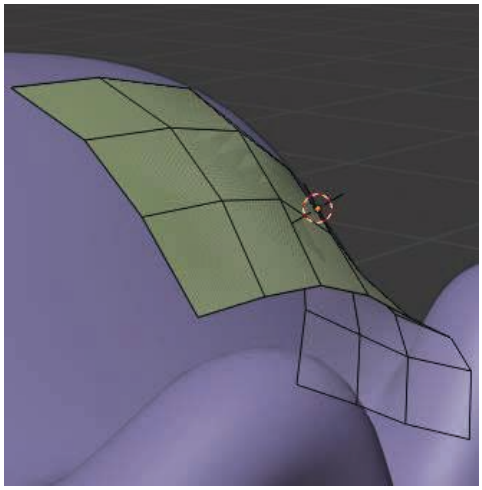
Now the problem is that we see the backface of the mesh.

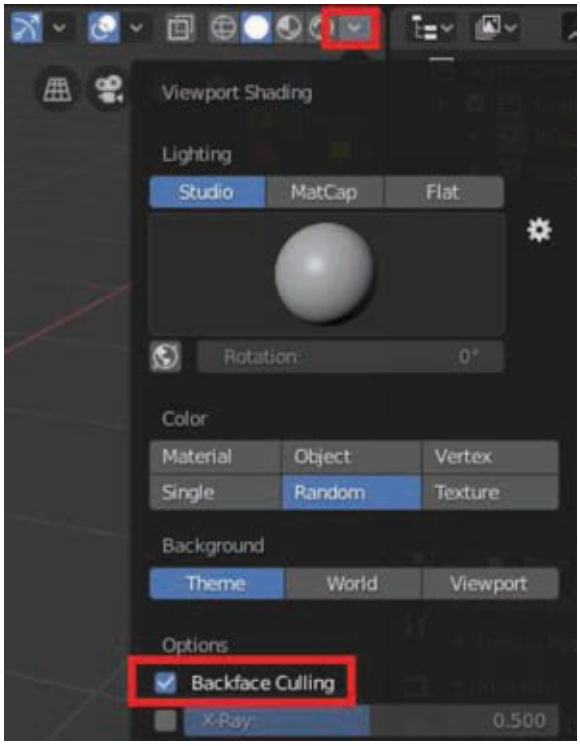
This can be confusing as it's blocking the view of things we still have to retopo.



To solve this, turn on **Backface Culling** in the Viewport Shading options.

This takes care of most of the backfaces, we now only see the wireframe.

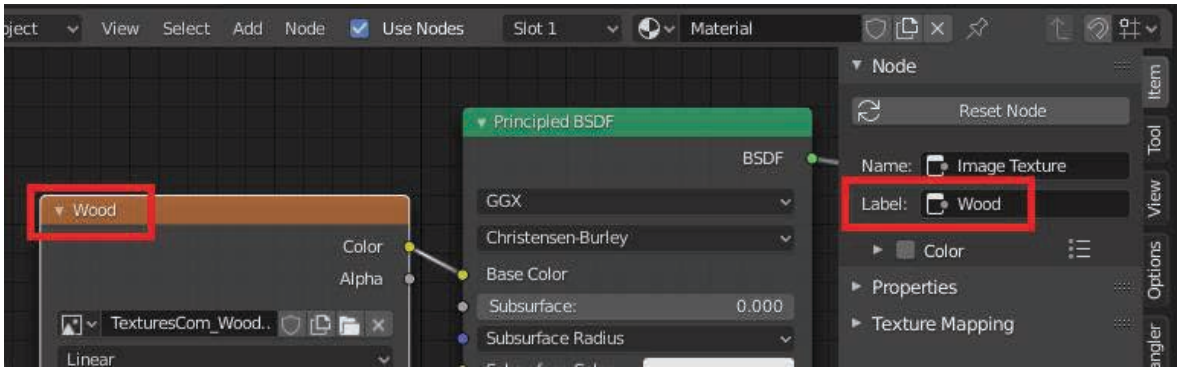




Node labels and colors

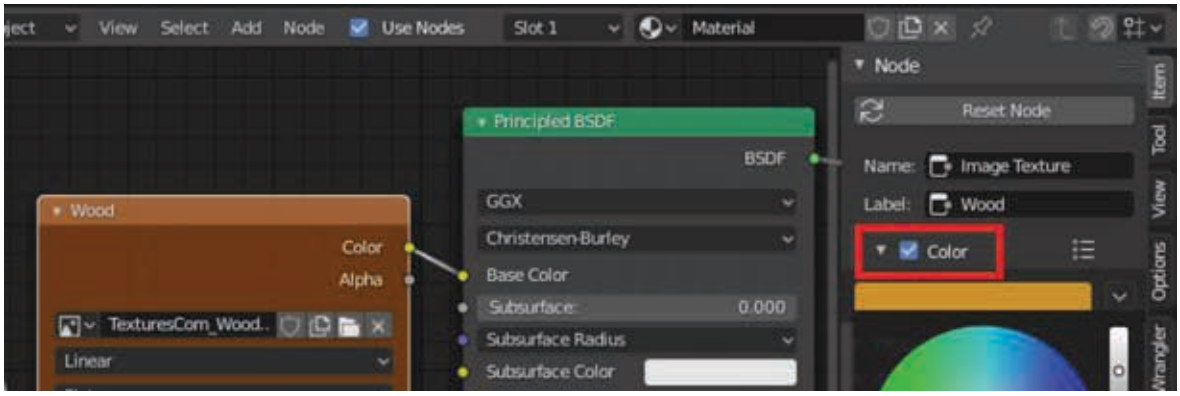
Do you want to make your nodes easier to find? You can give them a unique name in the Option Panel (**press n** to open the Option panel).

Select the node you want to give a unique name to. Type the new name in the **Label** field in the Option panel.

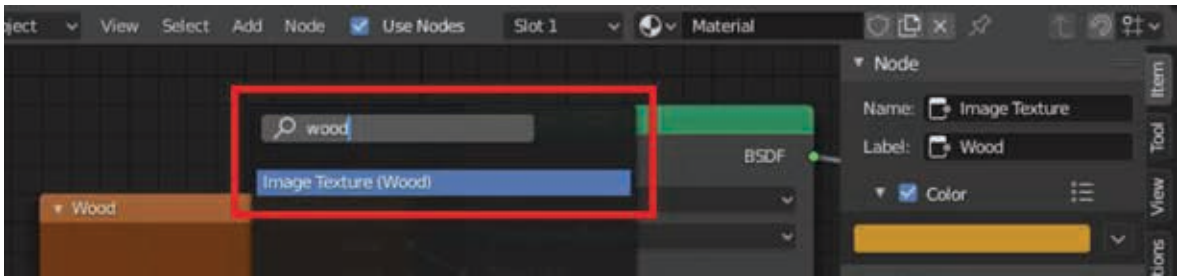


This will make it more clear what certain nodes do later on when you come back to the project after some time.

You can also give them a color to stand out more. Just check the **Color** box and choose a color.

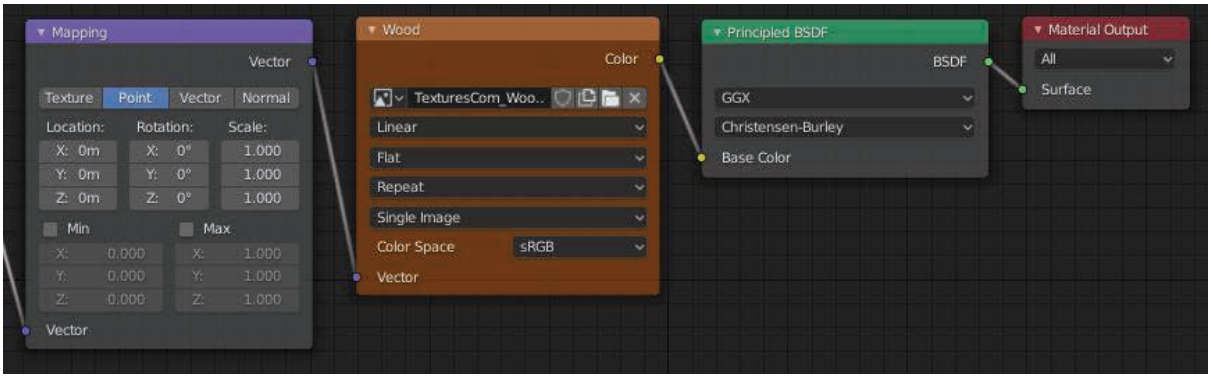


Press Ctrl+F to find the nodes you have given a name.



More Node Tips!

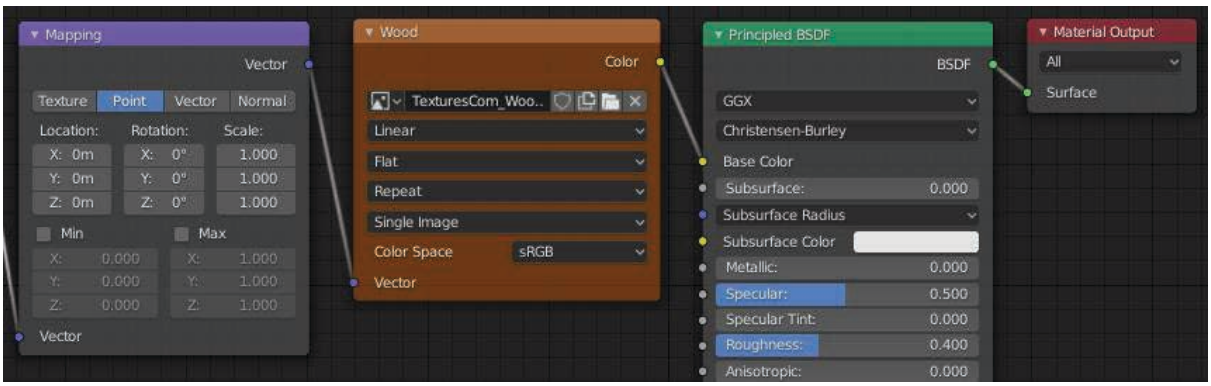
Press Ctrl+H to hide the unused sockets of selected nodes. For example below, the Principled BSDF only shows the Base Color input and BSDF output sockets.



Press H to make the selected nodes even smaller by hiding all their details.

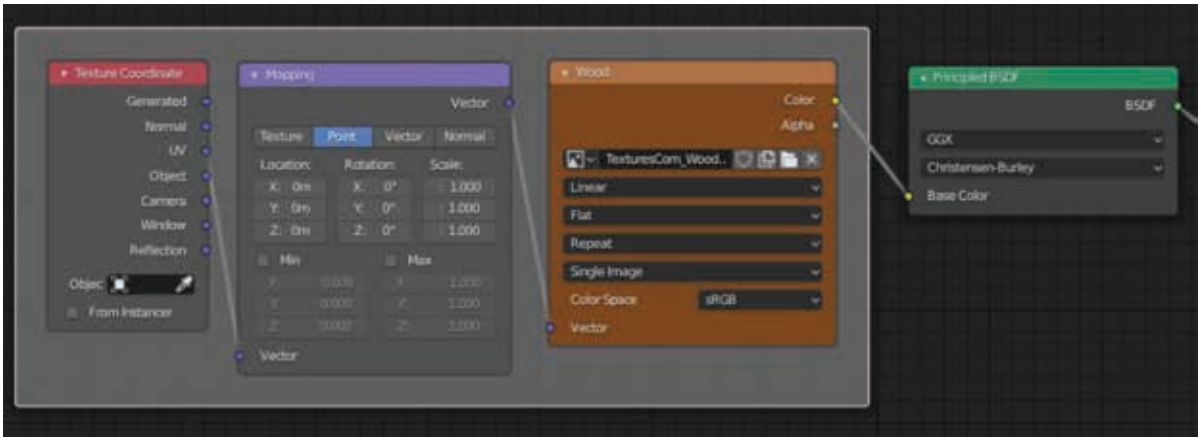
Press H again to unhide the node details.

Press Ctrl+H again to unhide the unused sockets.

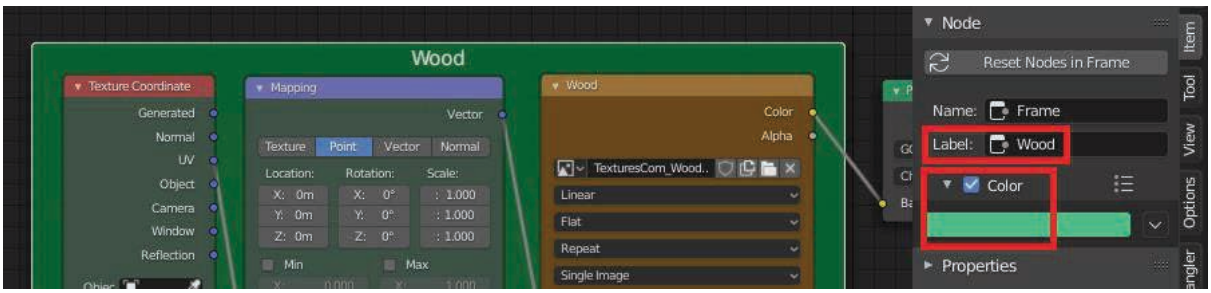


Click here to return to the index.

Press Shift+P to create a frame around a selection.

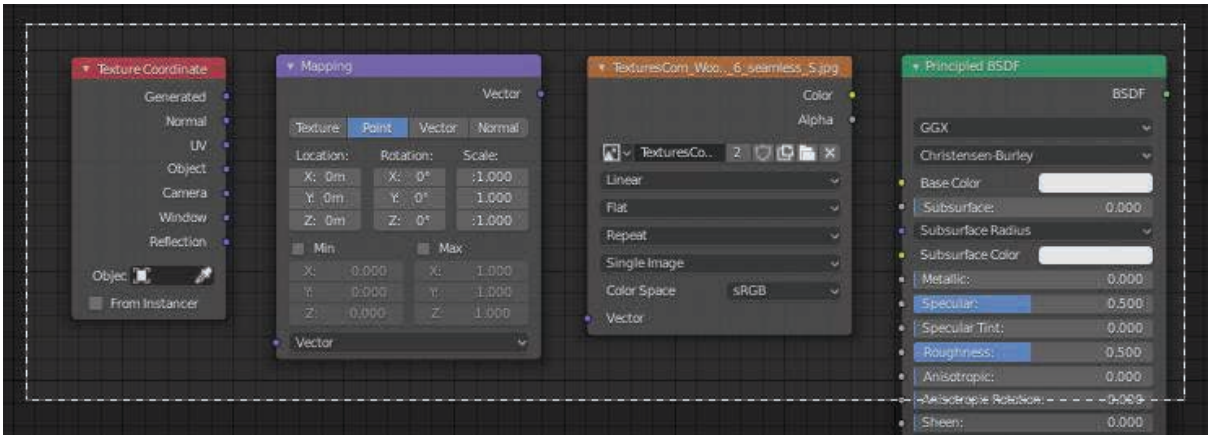


You can give the frame a name and a color. This all helps to keep things more organised. That is important when working together with other people, or just to help your future self understand what you were doing with all those nodes.

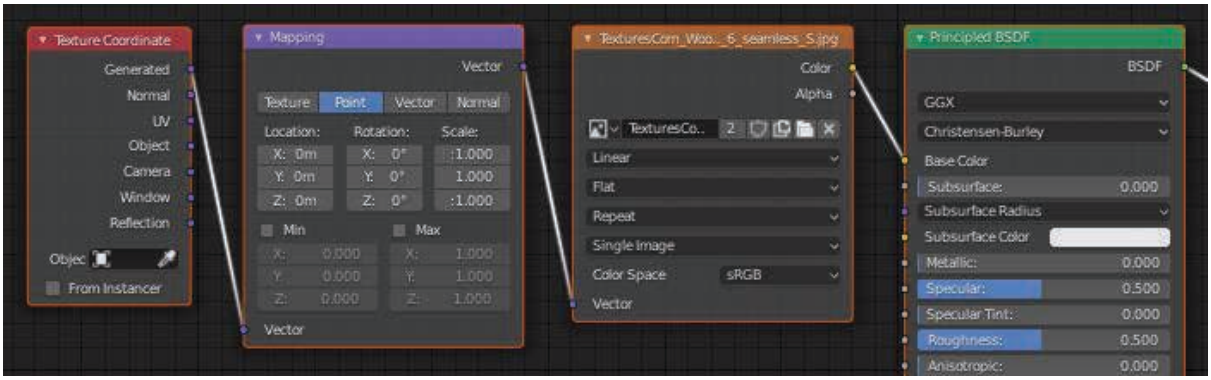


Even more Node Tips!

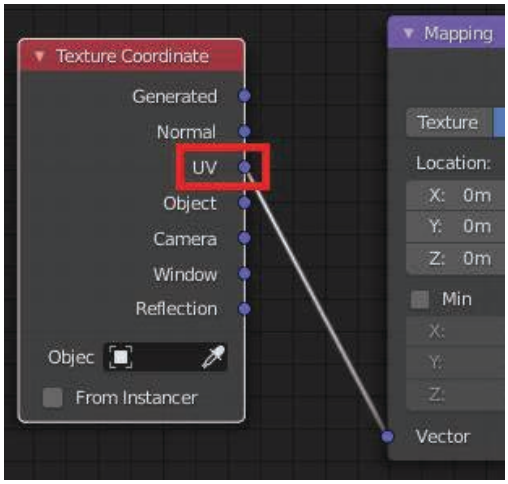
Select the unconnected nodes by dragging a selection around them.



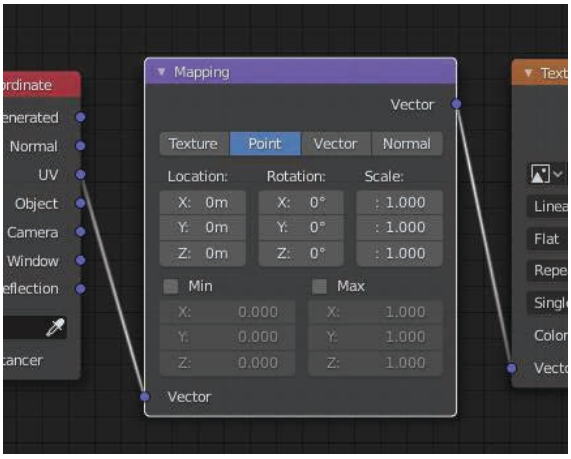
Press F to automatically link the nodes.

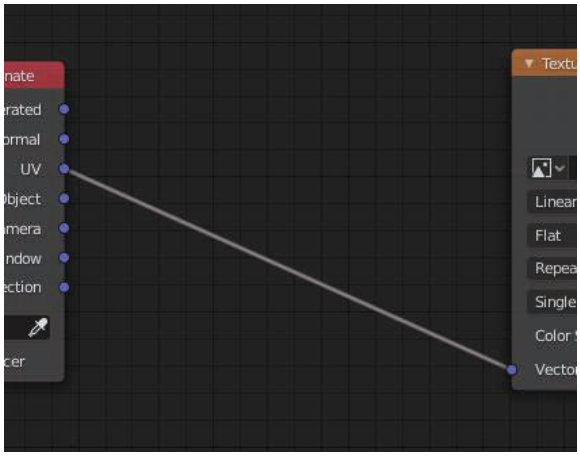


In the above example, the Texture node uses the Generated output. You may want to change this to UV in case your object has a UV map.



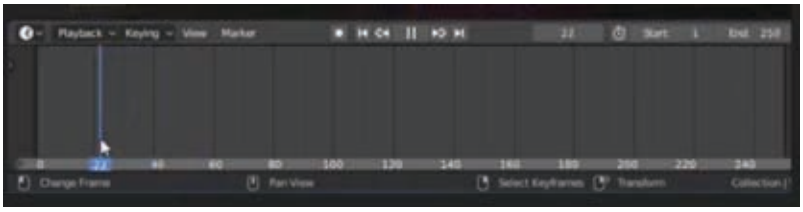
Instead of pressing X to delete a node **press Ctrl+X** to remove the selected node, keeping the link between the two other nodes. Otherwise the node connection will be broken.





Set in- and out-points in the timeline

You can now set in- and outpoints in the Timeline.



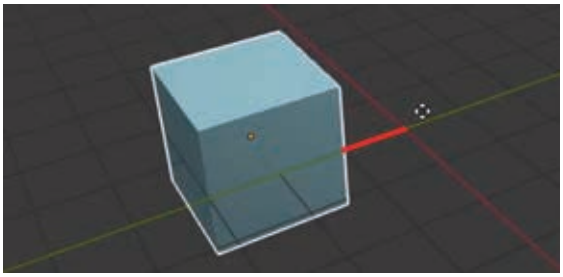
Press Ctrl+Home to set an in-point.

Press Ctrl+End to set an out-point.

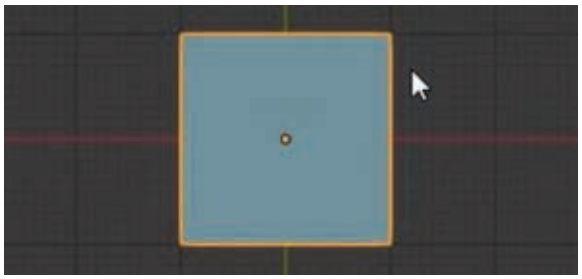
Note that these keyboard shortcuts only work in the Timeline window.
Not if the mouse cursor is in a different location.

Grid snapping

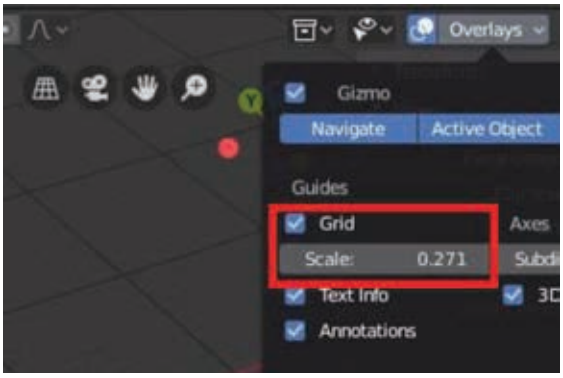
Snapping can be very useful during modeling. When **holding Ctrl, press G** to move the object and snap it to the grid lines.



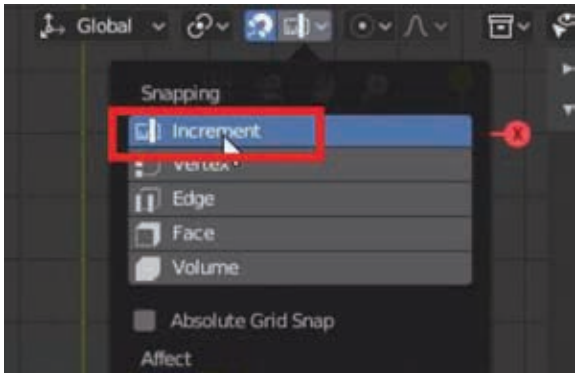
Switch to an orthographic view (for example by **pressing 7** for Top orthographic view) to snap to smaller increments of the grid.



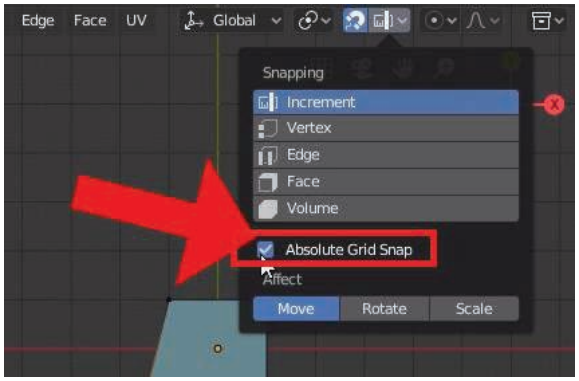
For smaller increments in a perspective view, scale the grid down in Overlays.



You can also turn snapping on in the menu. Set to Increment to snap to grid.



Turn on **Absolute Grid Snap** if you want the vertices to snap to the actual grid lines at all times.

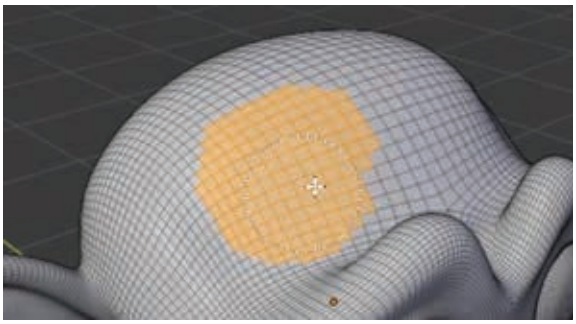


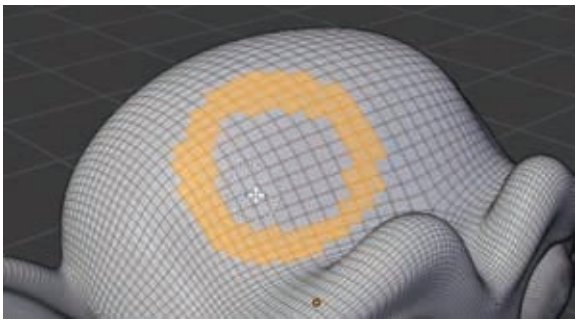
Click here to return to the index.

Draw select and lasso select

Press C in Edit Mode to use Circle Select mode. **Middle mouse button** changes the circle size.

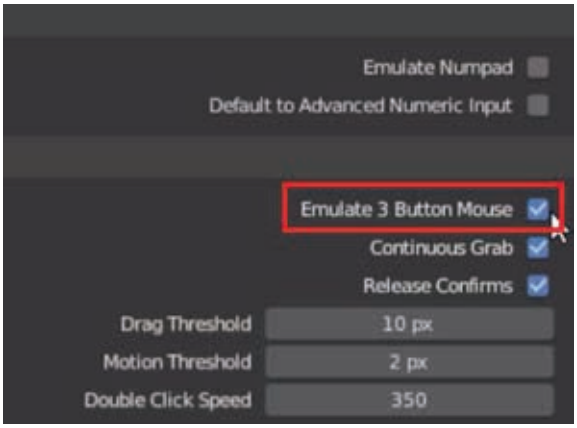
Holding Shift erases from the selection.



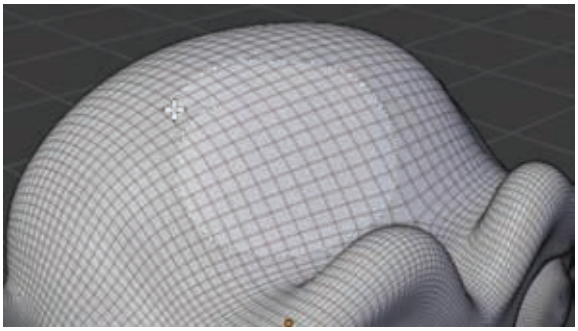


Right-click to exit Circle select mode.

Alternatively Holding Alt erases from the selection when you use “Emulate 3 Button Mouse”

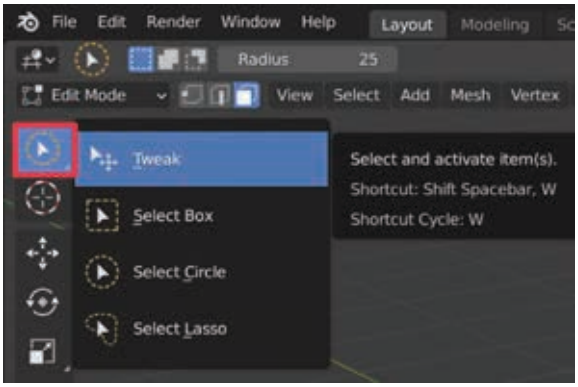


Another way to select is by using the lasso. **Hold Ctrl** and select with the right mouse button.



Alternatively **pressing W** cycles through the possible selection modes, including box select and lasso select.

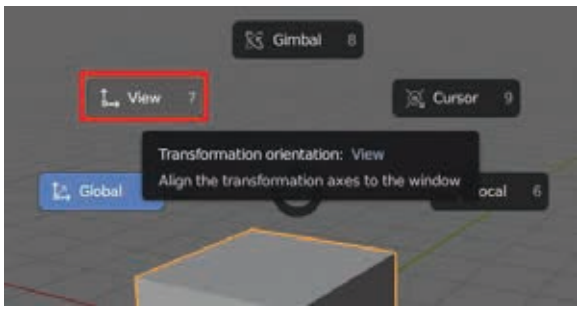
You can also click and hold the **Select Tool** in the Tool panel to see the possible Select modes.



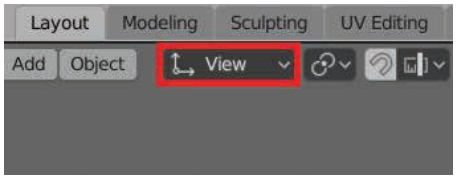
Align objects to camera view

Press [,] to bring up the Transformation orientation menu.

Choose **View** from the menu that pops up.

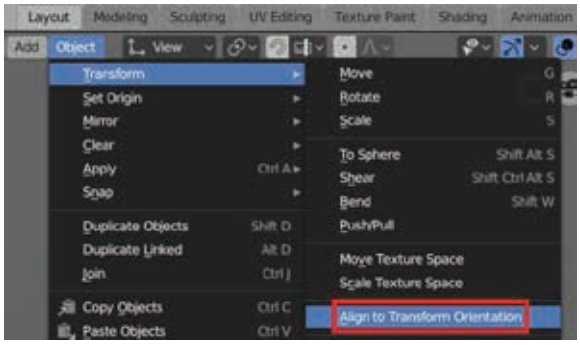


Alternatively you can choose View from the Transformation orientation drop-down menu in the interface.

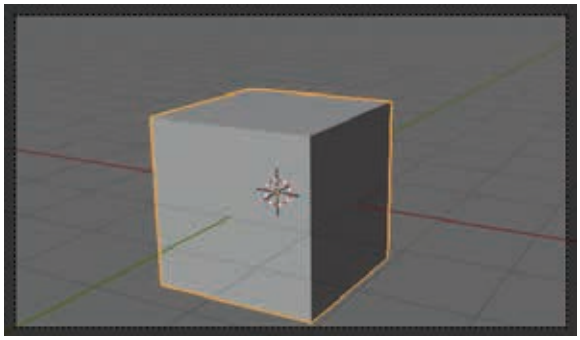


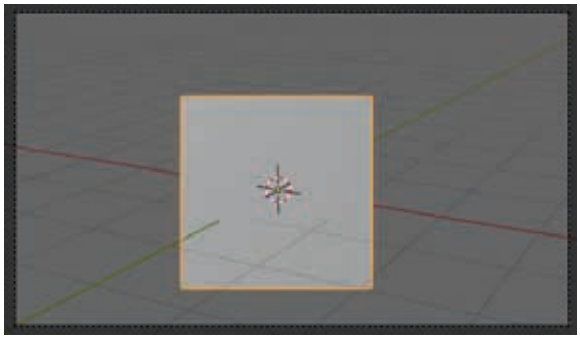
Select an object you want to transform to face the camera.

Go to Object > Transform > **Align to Transform Orientation**.



Now the object is aligned to the camera view.

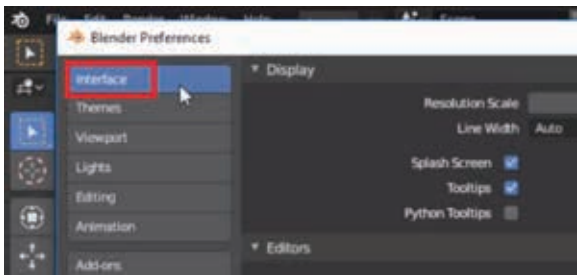




Click here to return to the index.

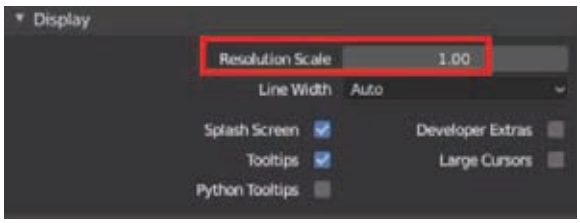
Interface scaling

Sometimes it's useful to make the interface more readable by increasing the scale of the interface elements. For example when recording a tutorial to make things more easy to see.

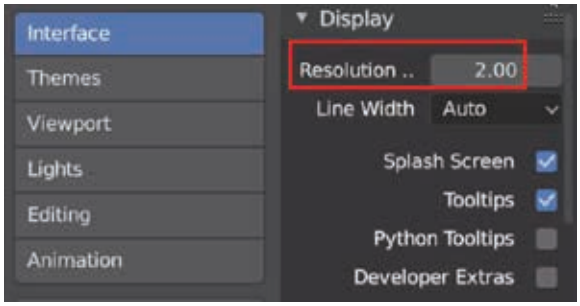


Go to Edit > Preferences > **Interface**.

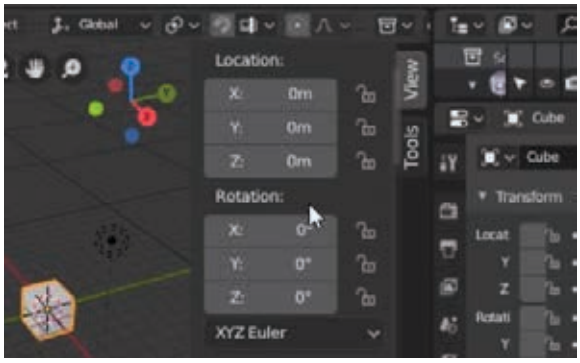
In **Resolution Scale** you can scale the interface.



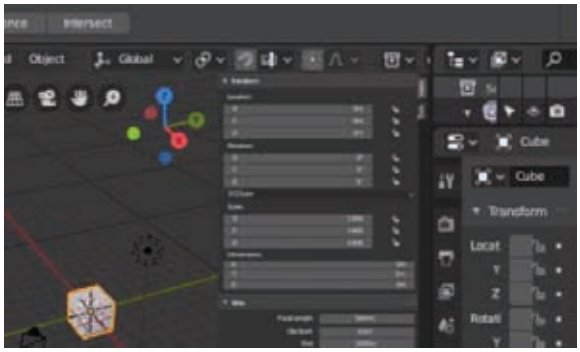
1 is the default value. You can increase or decrease the value.



You can also scale individual panels.



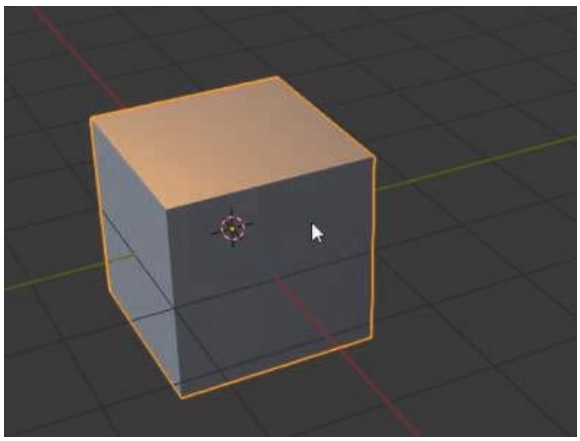
Press Ctrl+ middle mouse button and drag to change the scale of the panel.



Click here to return to the index.

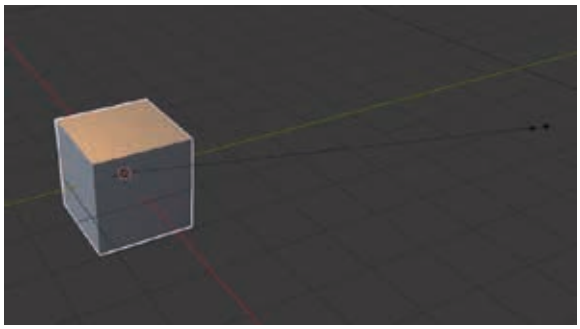
Scaling Tips

You probably know by now that **pressing S** and moving the mouse scales an object.

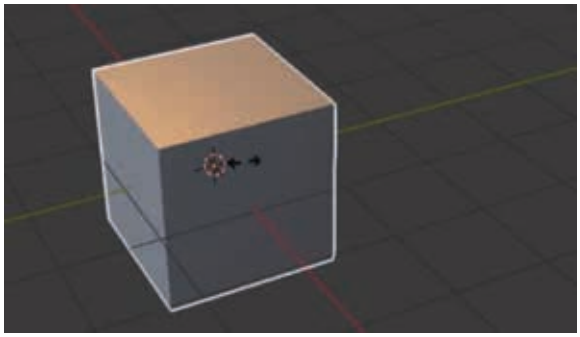


But there are ways to control how fast or slow that scaling is, to give you more control.

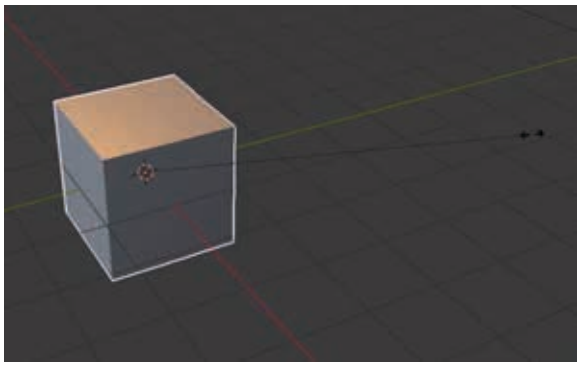
For example, Scaling near the pivot point goes in big steps.



Scaling from far away goes in small steps.



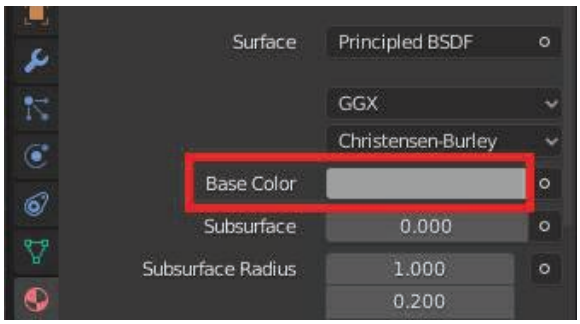
Holding Shift while scaling makes the steps even smaller.



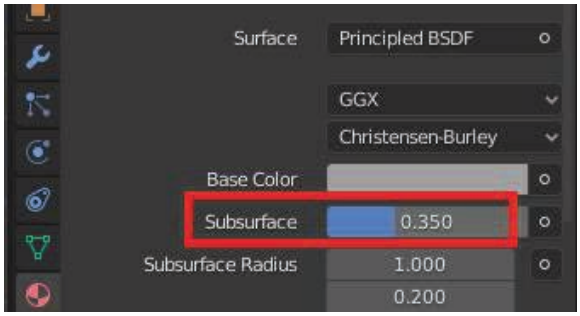
Scaling from far holding Shift is the most controlled.

Blender UI Value scrolling

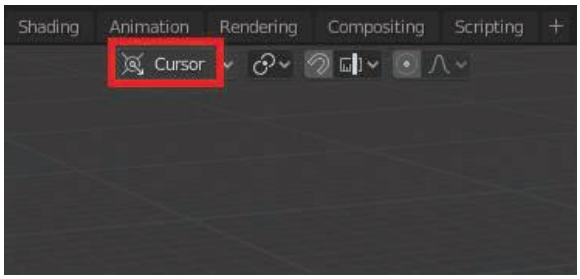
You can scroll through values in the user interface.



Simply **hold Ctrl** while scrolling the middle mouse wheel.



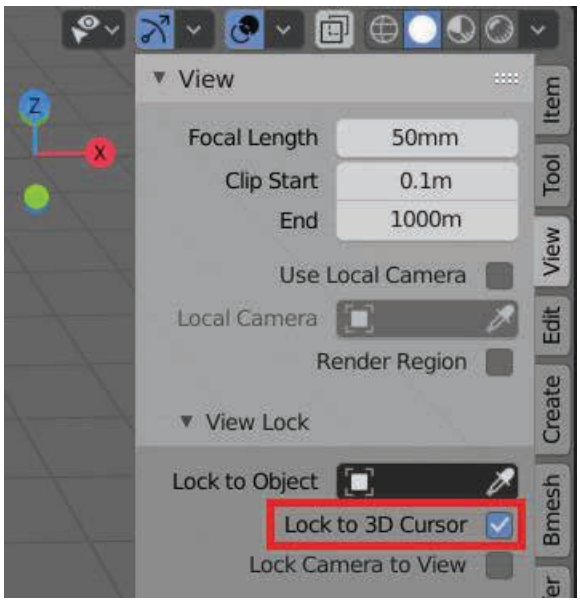
It works in most parts of the user interface.



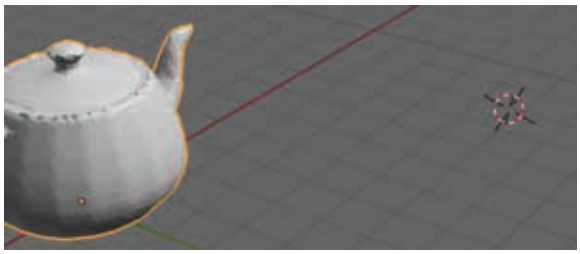
Click here to return to the index.

Lock to 3D cursor

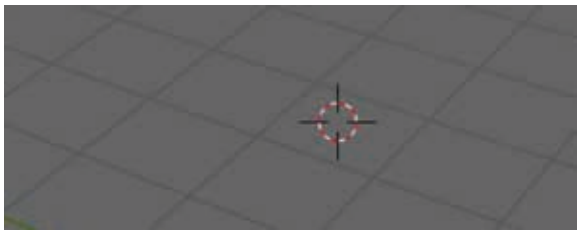
Press N to open the option panel. In the **View** tab, turn on **Lock to 3D Cursor**.



Now the viewport will rotate around the 3D cursor.



Set its location with **Shift+right-click**.

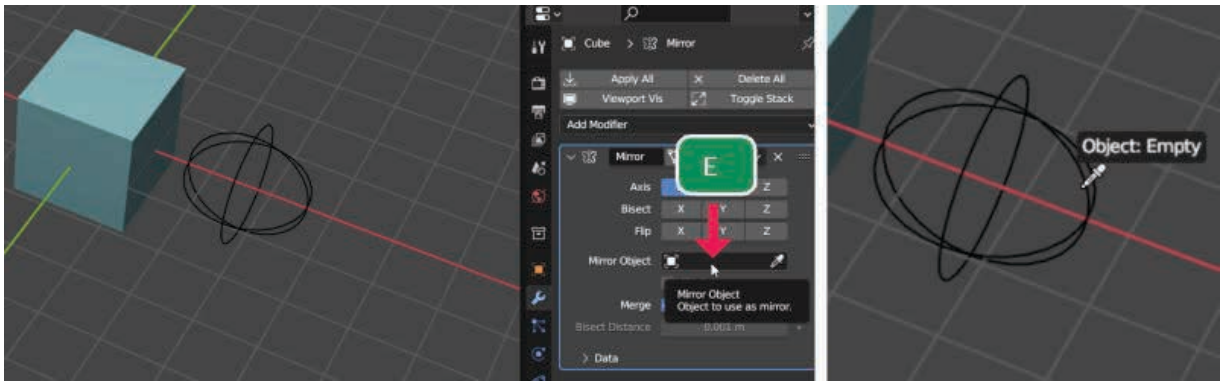


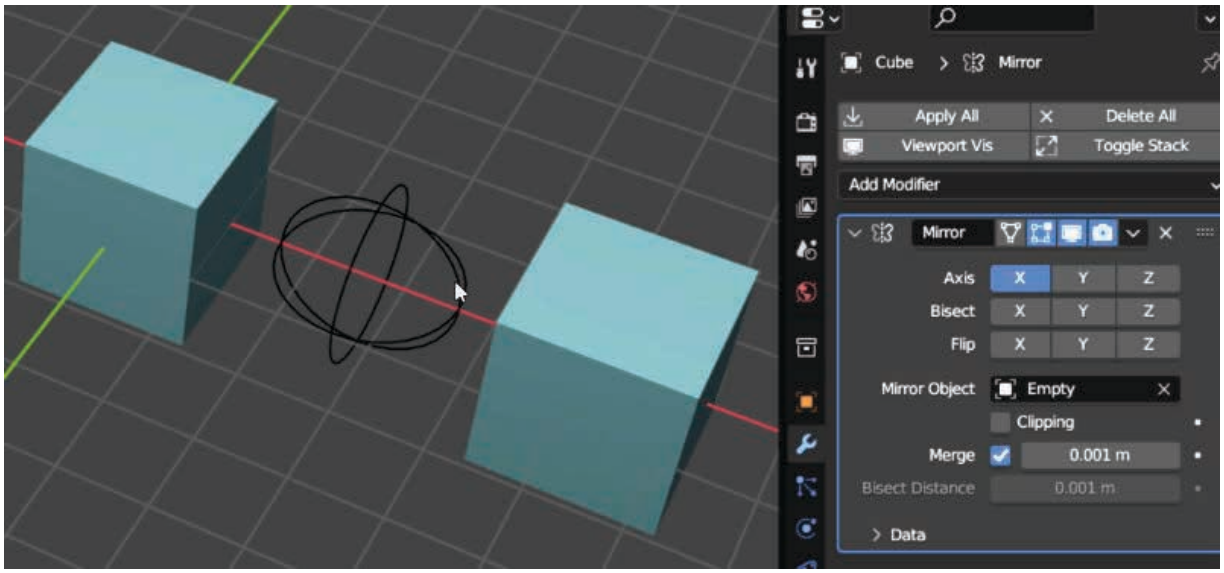
Press Shift+S and choose **Cursor to World Origin** to set the cursor back to the world origin.

Alternatively **press Shift+S+1**.

Maximum undo & Picker shortcut

Did you know that you can pick objects with the **keyboard shortcut E**? Just **press E** above the field for which you need to enter an object or value and pick it.

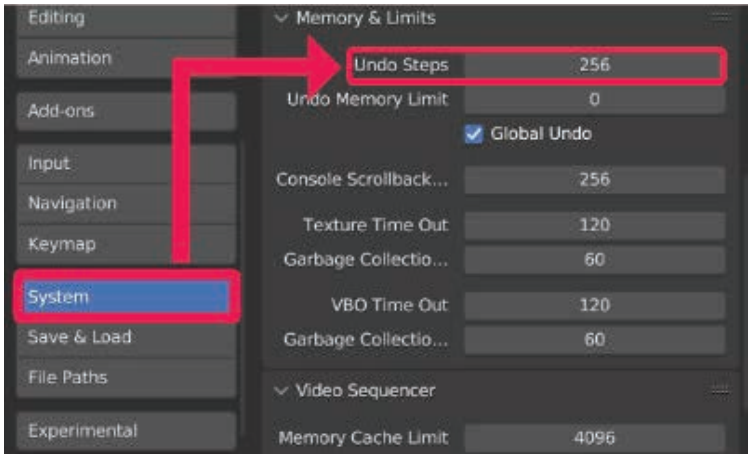




Another tip: Do yourself a favor and increase the **Global Undo level**. The maximum is 256.

Be careful though, too many might also increase the memory usage too much.

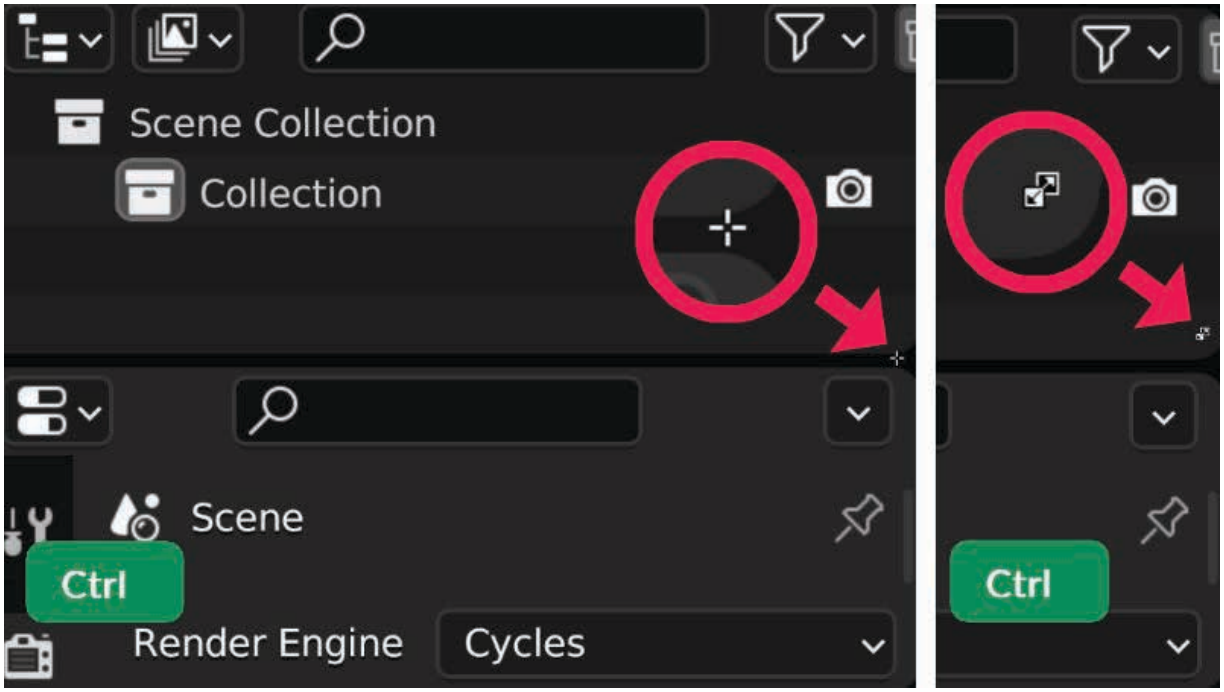
You know best what your computer can handle.



Click here to return to the index.

Swap interface windows

Did you know that you can swap interface windows? It works by dragging (with the left mouse button) from the corner of one window to another window while holding **Ctrl**.

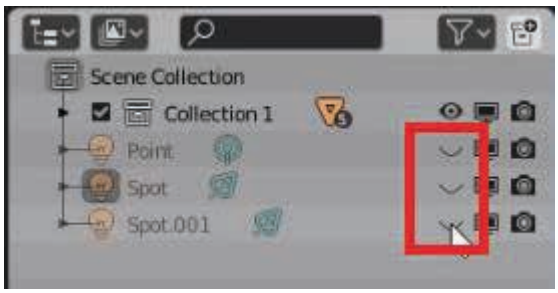


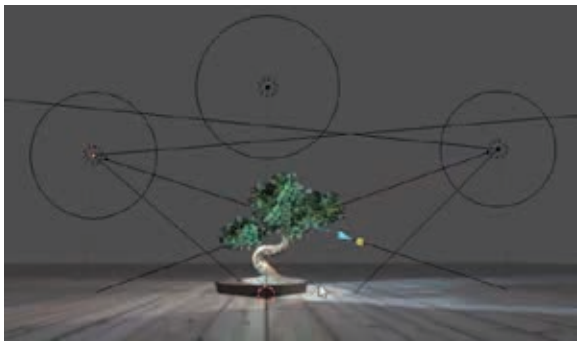
Once you drag to the other window, let go and the windows will swap.

Click here to return to the index.

How to hide lights in Eevee

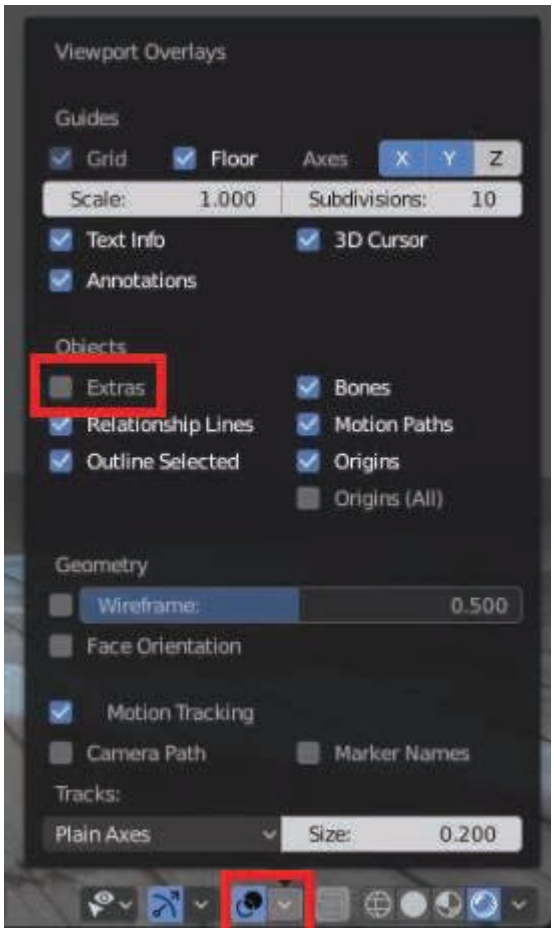
In case you want to hide the Light widgets in the viewport in Eevee, but keep the effect of the lighting, turning off the lamps viewport visibility won't work because it also turns off the lamps.





Instead go to the Overlay options and turn off the Extras checkbox.

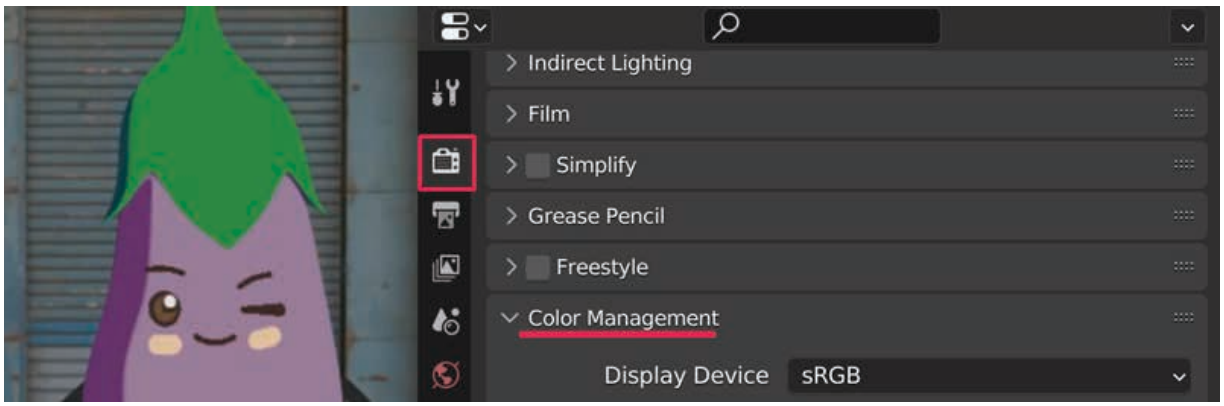
Now you still have the lights, but the lights overlays are not visible.



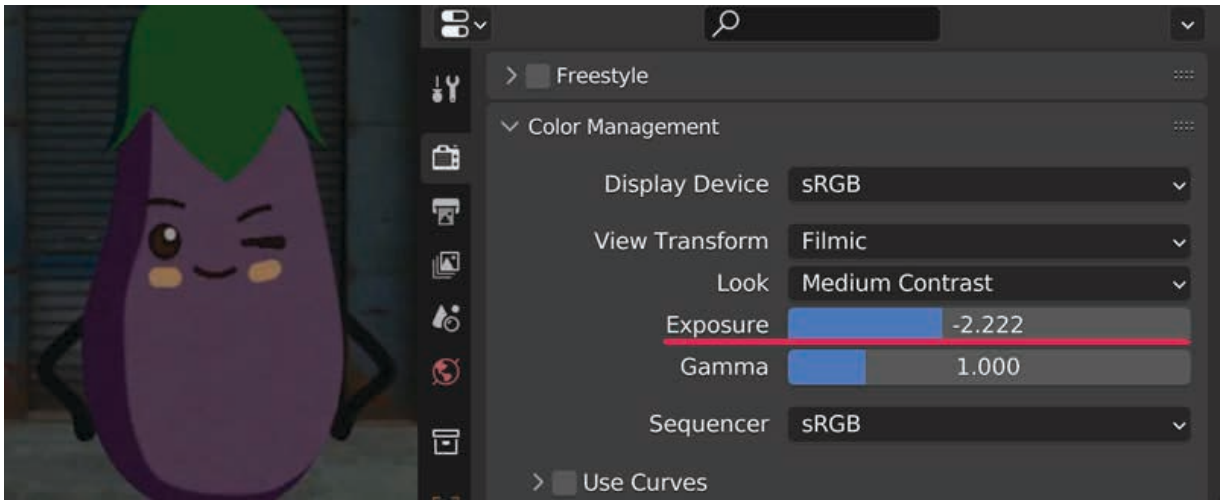
Bonsai tree model by Carlos Folch from https://resources.blogscopia.com.

Eevee Viewport Color Management

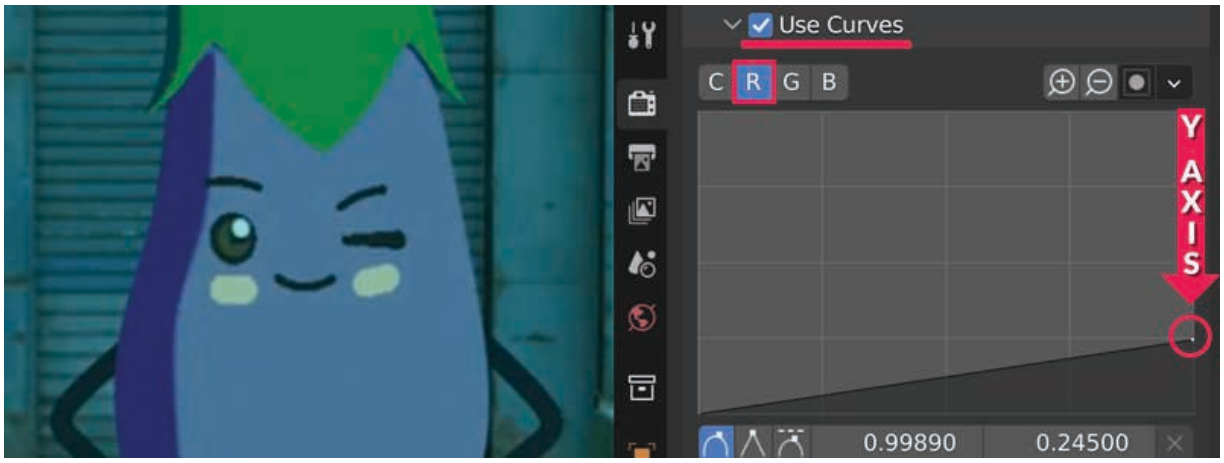
Go to Scene Color Management.



Here you can play with the **Exposure** and **Gamma**.

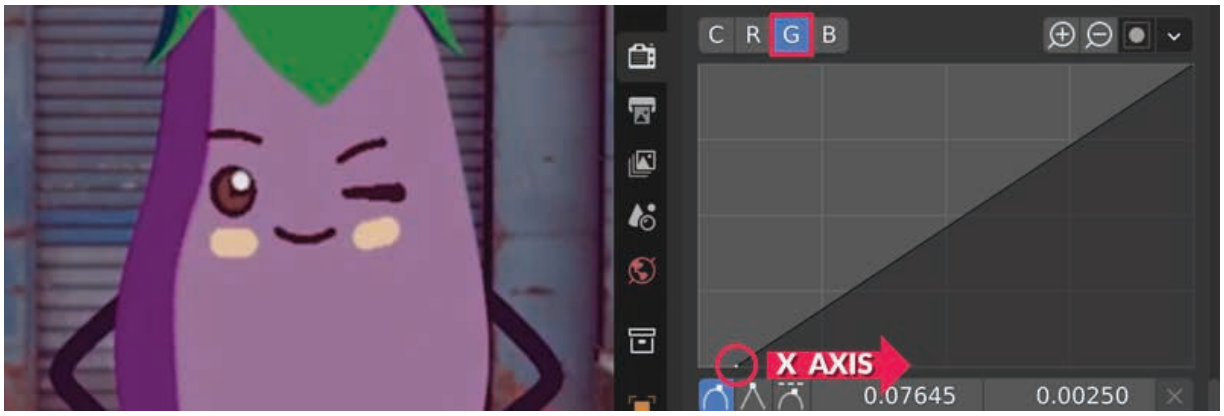


Click on "**Use Curves**". Click on a color you want to tweak. Dragging the point on the **Y-axis** down gives you the opposite color: Cyan, Magenta and Yellow for Red, Green and blue respectively.



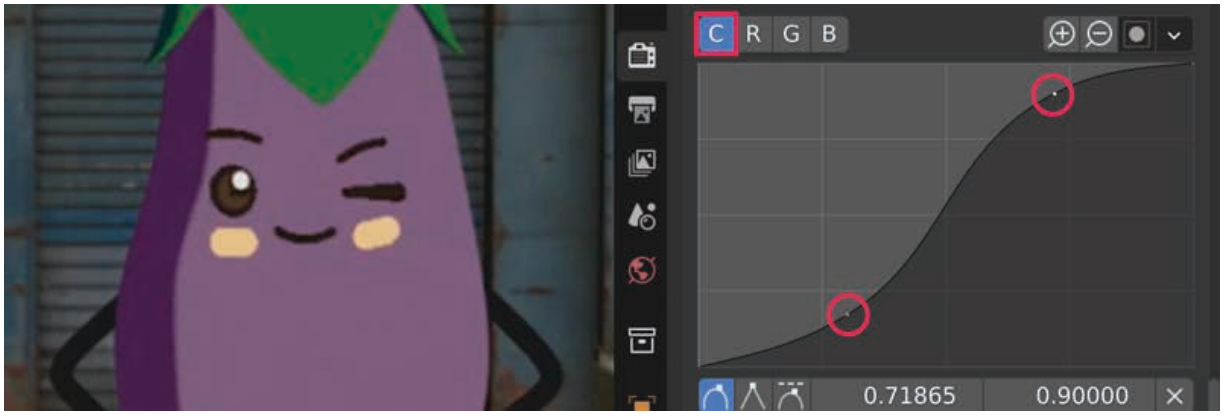
Click here to return to the index.

If you drag the point on the X-axis up you get more of that color in the darker parts of the image.

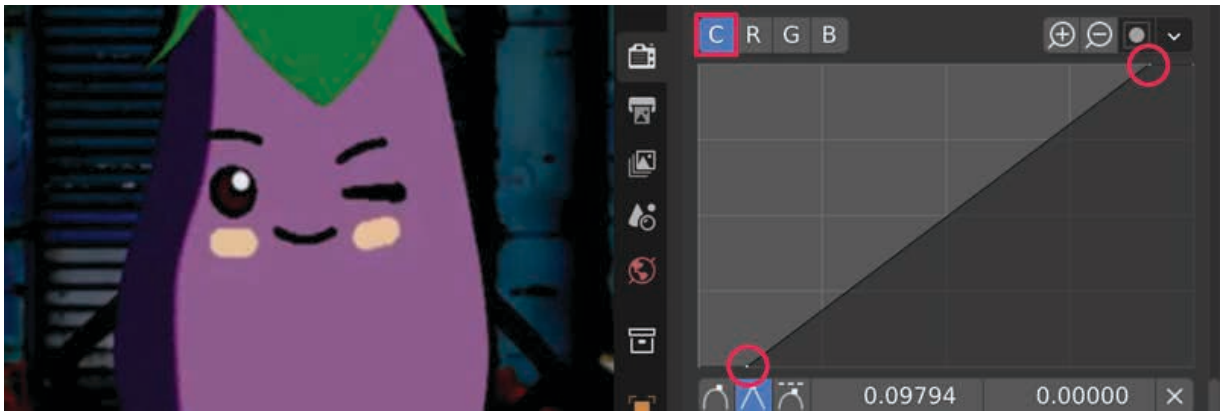


So if I want some Cyan in the shadows drag the darkest point of the Red curve to the right for example.

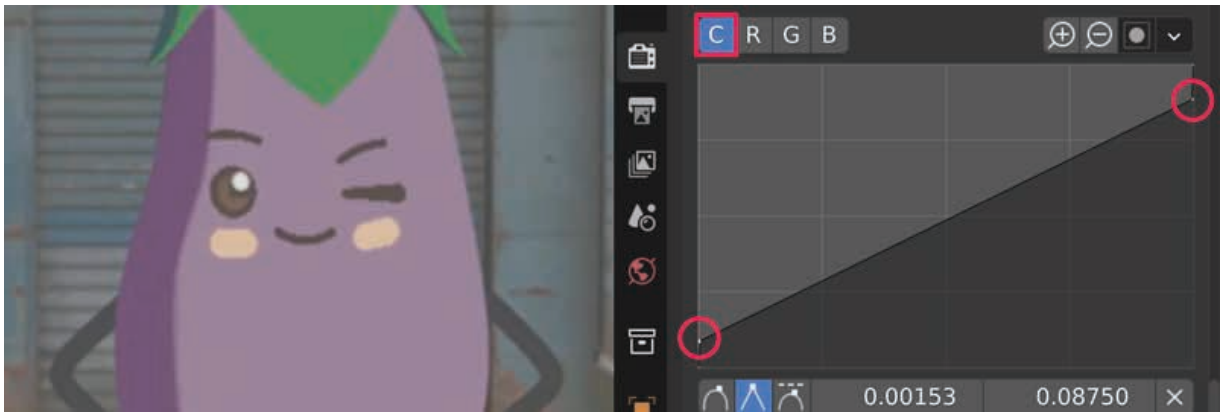
You can also **click on C** to adjust the luminance and make the darks a bit darker and the brights a bit brighter to add some contrast.



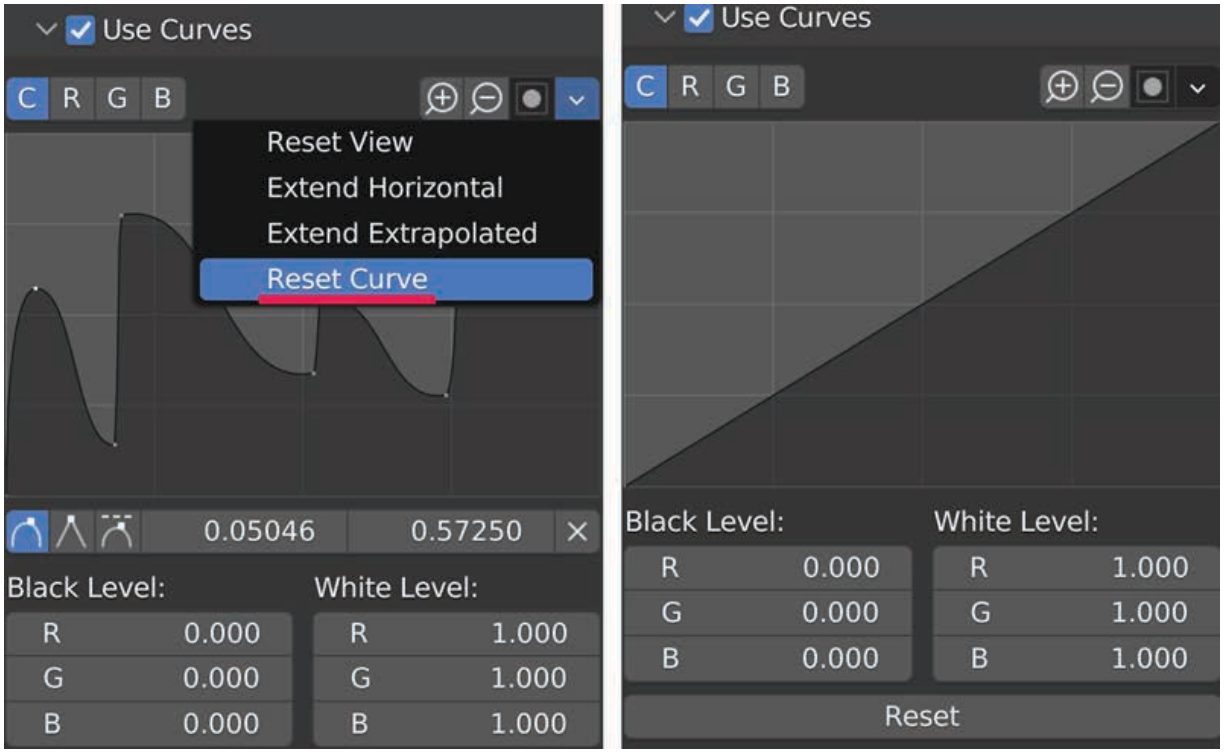
Or go for even more extreme contrast and bring the dark and bright points closer to the center.



You can also bring the dark part up and the bright part down, to decrease contrast.



If you want to undo your changes you can click on the drop-down menu and choose "**Reset Curve**".



3D Vocabulary

Below you'll find a kind of glossary of terms you'll encouter in the context of Blender or 3D art in general. If you have a word that you'd like to see explained here, let me know.

Active
The last selected object is the "Active" object. This is important when parenting objects together, or when copying properties from one object to another or to several others.

Albedo
The color texture of a material, without any baked in shadow information.

All-quads
In some pipelines, a model which is all-quads is desired. This means that the model has only Faces that have 4 Vertices and 4 Edges, so no triangles or N-gons (Faces with more than 4 Vertices or Edges). This can be necessary for various reasons.

For example, a quad-based model tends to deform better when animated, works more predictably with Subdivision Surface smoothing, and can be used more readily for a sculpting pass where detail is added. It's also easier to mix materials using only Vertex Painting, when a model has evenly distributed quad topology.

Alpha
The texture that gives the information about which parts of a material should be transparent.

Area Light
Area Lights are a type of light that is characterised by having a shape (as opposed to being just a point). For example, they can be a rectangle, or a disc. This is useful when simulating light coming from a TV screen or a window. Area Lights give softer shadows than point lights.

Armature
The (non-spooky) skeleton of your 3D character. The Armature consists of "bones" that can be posed in Pose Mode, and animated with keyframes. The character mesh is parented to the Armature using weight painting.

Asset
Things like models, materials, brushes, that can be re-used. They can be stored in the Asset Browser.

Backface Culling
The hiding of Faces that are facing away from the camera. This can be important for example during retopology, so you don't get confused by seeing geometry on the other side of your model, or when creating a stylized toon outline with the "inverted hull method".

Baking
Things such as fluid simulations, lighting, rigid body simulations can be heavy to calculate in real-time. In order to make it easier to continue working on scenes containing such elements, or to make rendering go faster, it's a good idea to pre-compute them.

This pre-computing is called baking. Typically, simulations are baked to keyframes, after which they can be played back smoothly. Baking is also a term used to describe the converting of materials to distinct PBR-textures, which are a standard system of texture maps used in game engines.

Complex shader set-ups which can not be exported are "baked" to images that combine to give the same result, but are easy to export and use in other software packages.

Bevel
The various ways of making corners of an object less sharp.

BNPR
Short for Blender Non-Photorealistic Rendering. This usually means rendering styles such as toon shading, or anime-style rendering. Literally it means anything which is not intended to be photorealistic.

Boolean
The technique of cutting into objects using other objects (or adding them in a way that fuses them).

Boundary Box
The simplest representation of a 3D object, usually meant for making the viewport more efficient. Once the rendering starts, the actual full detail object will be used instead of the bounding box.

Box mapping
If you want to add a texture to an object which is not UV-unwrapped, you can do so with "Triplanar" mapping. This simply projects the texture on the object from 3 planes (hence, tri-planar): x, y and z.

Then, there is some blending where they meet so there is no sharp visible seam. In Blender, this is called "Box mapping". It is achieved by choosing the "Box" Projection type in the Image Texture node, and setting a Blend value.

Usually the Blend-value is pretty low, like 0.1 or 0.2, depending on what looks good.

Box Modeling
The workflow of starting with a box, and adding more and more detail to it based on reference images, until you get the desired shape. A very old school method.

Bump map
A Bump map is a texture that adds fake height detail to a material, without the need for additional geometry. Any black and white image can be used.

Cap
The "lid" on for example a Cylinder. When creating a Cylinder primitive, you can choose: no cap (there will be a hole on either end of the Cylinder), or an N-gon cap.

Child
The object that is parented to another object (it's "parent"). It will follow the transformations done to the Parent object. You can parent objects by selecting both and pressing Ctrl+P.

Color Depth
The amount of colors used in an image file. Most images you see on the internet are 8-bit, as that's enough for our eyes. But when an image has to represent height detail, such as for a Displacement texture, 16-bit or even 32-bit may be preferable to prevent a stair stepping artefact.

Crease
Instead of adding more geometry to make edges of an object sharper (so-called "support loops" or "boundary protection"), we can also just select those Edges and "Crease" them. This tells any subdivision surface modifier, that those Edges need to remain sharp.

This is done by selecting the Edges, and pressing Shift+E+ a numerical value, 1 being the highest. Alternatively you can do this in the Option panel, in Edit Mode in the Item tab with the Mean Crease value.

Cycles
The render engine in Blender used for photorealistic renders. It's much slower than Eevee, but the results are often better.

Depth of Field
The area in which objects are sharp / in focus, as opposed to blurry / out of focus. This is created in the Camera settings by enabling Depth of Field and choosing an object to focus on, and adjusting the F-Stop value.

Diffuse
Another name for the color texture of a material.

Displacement map
The texture that adds height detail to an object through its material. Contrary to the Bump or Normal map, this texture does require additional geometry (usually supplied with a Subdivision modifier) to work.

Edge
The connection between two Vertices. Connect 3 Edges in a triangle and you have... a triangle. You're on your way to making a 3D model!

Edge Loop
an Edge or connected loop of Edges that goes all the way from one side to another, or in a loop around an object. Add an Edge Loop in Edit Mode with Ctrl+R.

Eevee
A render engine in Blender that's quite fast, but less photorealistic than Cycles, the other render engine.

EXR
EXR or OpenEXR is a file format, that very efficiently stores images with high Bit Depth, so it's useful for Height Maps that need a lot of colours to avoid visual artefacts when displacing geometry.

EXR files can even store several layers of an image, so you can render separate passes to one EXR file and then separate them again in your compositing software of choice.

Extrude
The technique of creating a 3-dimensional extrusion from a 2-dimensional selection (for example, a selection of one Face). The shortcut for extruding is E.

Face
One of the building blocks of a 3D object, a Face is the space that fills up connected Edges and Vertices.

Fields
Fields are specific nodes in the Geometry Nodes workspace, that are a piece of code that you can use again and again. Nodes with a diamond input socket, can accept input from fields.

Geometry
The arrangement of Vertices, Edges and Faces that make up an object. "It has a lot of geometry" for example can mean that a model has a lot of Vertices.

Geometry Nodes
In Blender, you have "modifiers" that do things to your objects, for example add subdivisions to them. With Geometry Nodes, you can create your own modifier. This is done by using nodes in the Geometry Node Editor, after adding a Geometry Nodes modifier to an object.

Global Illumination
When rendering, the way light bounces and illuminates objects in a realistic manner is called Global Illumination. Without it, it looks a lot less realistic. The first movie to use this technique? Why, it was Shrek 2.

GPU
The processor on your graphics card. When rendering with GPU, it's often faster than with CPU (the processor of your computer). However, GPU often has memory limits for textures, that the CPU doesn't have.

Hard Surface model
Any kind of non-organic model. Often used when talking about stuff like robots, weapons and other metal objects. But a chair is also a hard-surface object (unless it's a particularly soft and squishy chair...).

HDRI
A High Dynamic Range Image. Basically, a photograph that contains multiple exposures. From under to overexposed. The render engine then uses what it needs from that image for lighting a scene, leading to more realism.

Height map
The same as a Displacement map.

Low-Poly modeling
A 3D model that makes the most of very limited amount of Faces, or even triangles. In some cases, the low-poly nature of the model may be part of its design and charm. Low-poly objects may be a requirement of certain videogames.

Matcap
MatCap (short for Material Capture) is a way of quickly and easily applying a material and lighting set-up to your scene. This is done by loading a spherical image that contains all this material and lighting information.

MatCaps are useful when sculpting, as they often show the shape and its lumpy surfaces more clearly than the normal grey solid viewport mode does.

Material
The properties contained in one or more nodes, that make up the look of a 3D object. Materials can (but don't have to) use image textures, or procedural textures. Also called a "shader".

Mesh
A construction that consists of Vertices, Edges, and Faces. A 3D model, essentially!

Mudbox
Mudbox is software that is dedicated to 3D sculpting (and painting). It was developed during production of the Lord of the Rings movies to speed up the sculpting process. It was sold to Autodesk in 2007.

Unfortunately, updates of the software are very underwhelming, with the 2025 version having just one bug fix, after years of waiting for new features. So, kind of the opposite of Blender in that sense.

However, Mudbox is really easy and fun to use for sculpting and painting, and it boasts a great layer-system. This makes the sculpting process in Mudbox more non-destructive. Definitely worth a try, and not that expensive at 100 USD per year or 15 dollars per month. So if you love high resolution 3D sculpting, you may want to try it.

N-gon
A Face made up of more than 4 Vertices or Edges.

Nodes
Nodes are what make up a material (for example an Image Texture node, that contains an image texture), or a geometry nodes modifier, or a compositing set-up. It's a visual representation of how data is processed. Nodes usually contain input and output sockets, to connect them together.

Noise
Procedurally generated noise can be used instead of an image texture, to create patterns or detail in a material. This is done with a Noise texture.

Normal map
Similar to a Bump map, a Normal map adds fake height detail to a material, without the need for additional geometry. It's a colourful image that is created by baking the detail from a high-resolution model to a low-poly model.

Normals
The direction a Face is pointing in. This is important for how the shadows are displayed on an object, especially for non-photoreal renders like toon shaders.

NURBS
A different way of creating 3D objects, instead of Faces made up of Vertices and Edges, it calculates the curved surface connecting those points. Which gives an infinitely smooth surface. It's less used nowadays.

Octane render
A third-party render engine which also has a version for Blender. The Blender version actually consists of a separate build of Blender.

OpenVDB
A file format that can store things like explosions, clouds and other kinds of fluid simulations.

Organic modeling
The opposite of hard surface modeling: Everything that's soft and squishy, like humans and animals.

Orthographic
When there's absolutely no perspective, we talk of an orthographic view, or orthographic camera.

Parent
An object that is above another object (it's "child") that is parented to it. You can parent objects by selecting both and pressing Ctrl+P.

Particles
When simulating rain, dust, small debris, leaves falling from a tree and other such small objects in large quantities, particles are usually used.

These are not much more than 3D coordinates following the rules set by a "Particle System", and replaced by a specified object (such as a model of a leaf) or a collection of objects, at render time.

PBR
PBR is short for Physically Based Rendering. It's usually used to talk about materials using a specific set of image textures that together achieve a photorealistic result: Some of these textures are Roughness, Albedo, Normal, Specularity, Height, Metalness.

Perspective
The 3D viewport has two modes: Perspective and Orthographic. Perspective means that things get smaller as they are further away, just like how we perceive things with our eyes. In the Option panel under View, you can adjust the Focal Length of the viewport "lens", if you want.

If you want a completely Orthographic view, in other words a view without any perspective, press 5 on the Numpad. Pressing 5 toggles between Perspective and Orthographic mode.

Pivot Point
The Pivot Point of an object is the part relative to which it transforms, in other words relative to which it rotates and scales. This could be the Origin of the object (the small orange/yellow dot) but it could also be the 3D Cursor. You choose this from the Transform Pivot Point menu, which is located in the top / middle of the interface.

Planar
Making something "Planar" usually means making it perfectly flat. For example, if you have a surface that has some bumps caused by Vertices that are not exactly on the same plane as the other Vertices that make up that surface, you can make the surface "planar" by scaling flattening them. This can be done for example with the LoopTools menu in Edit Mode.

Plane
A Plane (in the context of Blender) is a flat mesh consisting of at least one Face. It's one of the Primitive mesh objects that you can add via the Add menu.

Point Cloud
A Point Cloud is a set of points in 3D space that make up a 3D object. Point Clouds are usually generated through photogrammetry or 3D scanning, after which they are stitched together into a surface.

Point Light
Point Lights are a type of light source in Blender that emit light in all directions.

Polygon
The same thing as a Face. In most other 3d software, people talk about polygons rather than Faces. Triangles, Faces and N-gons are all polygons.

Primitive
3D modeling software usually comes with “primitive objects”, these are simple 3D objects such as a Cube, Sphere, Cylinder and so on. Typically a more complex model such as a tea pot is also included, however in Blender this is a monkey head instead. Using primitives you can quickly block out more complicated shapes.

Procedural
Procedural can stand for things that are created entirely by nodes, for example a procedural noise texture. The advantage of a procedural texture, is that there is no image file associated with it - it’s just math.

So you can’t accidentally break a material by moving files and forgetting to move the texture associated with it. The downside, is that procedural textures look quite recogniseable.

Quads
Quads are Faces that contain 4 Vertices and 4 Edges, no more and no less. See also “All-quads”

Raytracing
Raytracing is a rendering method where a ray gets cast into the 3D scene, and the render engine looks at what the ray hits, in order to know what to show. This is a slow process, though recently there have been graphics cards like the nVidia Turing, that can do raytracing in real-time.

Refraction
Refraction is the way the light bends as it goes through a transparent material. This makes materials such as glass or water look more realistic.

Render Pass
Instead of rendering your image all in one go, you can split it up into several parts: For example the roughness, diffuse, depth and so on. Then you have more control over how they are combined later on, resulting in potentially a better final image.

Rendering
The process of creating images from your 3D scene, from the view of the virtual camera.

Resolution|
How many horizontal and vertical pixels an image has is called its resolution.

Rigging
The process of preparing a model for animation, by adding a kind of virtual skeleton to it, which is called an Armature. An Armature consists of bones. The model is connected to these bones by a process called Weight Painting, which tells the bones how much influence they have over each part of the mesh.

Rigid Body Dynamics
When you want to simulate objects reacting to gravity, such as a pile of bricks crumbling down, that's typically done with Rigid Body Dynamics. Rigid Bodies are the objects that take an active part in the simulation. There are also objects that they can collide with, that don't move - these are considered passive rigid bodies.

Roughness
How shiny an object is, is determined by the Roughness channel of its material. This can be influenced by a simple value from 0 to 1 (0 being very shiny, 1 being non-shiny or "rough"). Or you can use a black and white texture to influence this more precisely.

Samples
As you render something, you'll see the image appear in a very rough, grainy form at first. These are the samples. The more samples, the more clearly defined and noise-free the image will be. And also, the longer it takes to render. You can make things a little faster by using denoising.

Seams
When you UV-unwrap something, you do so by marking Edges as "Seams". Think of it like taking scissors and cutting the 3D model in specific places, allowing you to flatten it.

Shader
Often used as a synonym for a material, but really a shader is the code that determines how the properties of a material are applied to a 3D object. People also talk about shaders when referring to materials that only use mathematical properties, such as procedural noise textures, and no image textures.

Simulation
Things like liquid, smoke, and objects reacting to gravity are usually done with simulations in 3D. This makes them move in a realistic way. Simulations tend to be quite processor intensive, and slow. So it makes sense to bake them after they've been calculated, so you can save the result as keyframes and don't have to run the simulation again.

Specular
A specific value of a shader that makes the material more shiny. However, this is in most cases a value you should leave alone. To make things more or less shiny, use the Roughness input instead.

Sphere
Spheres are one of the primitive objects that you can add to a scene. There's UV Spheres, which are convenient for mapping textures of planets for example, Icospheres, which are useful for things like particles, and Quad Spheres, which are good for subdividing and sculpting due to their evenly distributed quad topology.

Spotlight
A Spotlight is a specific light type in Blender that has a "cone" and a direction. You can make the edge of the spotlight projection hard or soft.

sRGB
Depending on what kind of monitor you use when working in Blender, you may need to choose a different display device in the Color Management options. However, most monitors use sRGB which is the default.

If it's something else, that means you have a very expensive monitor and you probably know to change this setting. However Mac devices often have Display P3 as their display device.

Substance painter
A dedicated texture painting program. As such, it has many available texture brushes, and clever methods of quickly generating masks such as edge wear. It also has a big marketplace of user-made materials on places like Artstation.

It has a monthly license from Adobe, but you can still buy the perpetual license on Steam. It just doesn't get updated after one year.

Subsurface Scattering
With organic stuff like skin, the light doesn't just bounce right off it, but goes a bit deeper and bounces around inside before being reflected back. This gives organic material its specific look. In Blender this is done with "subsurface scattering".

Taper or tapering
When an object becomes smaller in one direction, you can say that it "tapers" down. For example, a piramid tapers towards the top. If you use a SimpleDeform modifier, it can be set to taper your model.

Texture
An image file, that supplies visual information to the material of a 3D object. Materials usually have several textures: The color texture, the roughness texture, the Normal texture... sometimes, people also call those textures "maps".

Tilt
When working with Curves, you can "twist" the individual vertices of the curve. This can only be perceived when the cure has some depth, or thickness. This twisting is called the "tilt". This value can be found in the Option panel in the Transform tab. Or by pressing Ctrl+T.

Timeline
The part of the interface where keyframes for animation are displayed. You can press the Spacebar to start/stop playing animation, and you'll see that the playhead moves from left to right on the Timeline.

Toon shader
A toon shader is a material that gives your object a stylized look. This can be used for 2D-animation style rendering.

Transmission
The Transmission value of a material in Blender is the amount of light that can pass through it. A value of 0 makes an object completely opaque (not transparent), and a value of 1 makes it fully transparent.

Transparency
Transparency in a material is achieved with the Transparent BSDF node. It's generally not used for something like glass, but to tell Blender which parts of geometry are fully transparent, as if there was really no geometry there. This can be useful for alpha transparency, or for some specific effects like an inverted hull material in Cycles.

Tri or triangle
A Face made up of 3 Vertices or Edges. The simplest possible Face.

Triplanar
If you want to add a texture to an object which is not UV-unwrapped, you can do so with "triplanar" mapping. This simply projects the texture on the object from 3 planes (hence, tri-planar): x, y and z. Then, there is some blending where they meet so there is no sharp visible seam.

In Blender, this is called "Box mapping". It is achieved by choosing the "Box" Projection type in the Image Texture node, and setting a Blend value. Usually the Blend-value is pretty low, like 0.1 or 0.2, depending on what looks good.

Unwrapping
Unwrapping is the process of unfolding a 3D object to a 2D layout, for the purpose of more easily editing the texture in a 2D space. It is achieved by adding selecting Edges to mark as "seams" first, and then selecting the Faces you want to unwrap and choosing unwrap (In Blender this is done in Edit Mode by pressing U and choosing Unwrap).

UV map
The unwrapped, 2-dimensional representation of a 3D object for the purpose of texturing.

UVs
UV's are the coordinates on a UV Map (see UV Map). As a UV Map exists in 2D-space instead of 3D-space, we use UVW instead of XYZ for coordinates to show that it's 2D we're talking about. And W is left out, as in 2D we only need two coordinates.

Vertex
The simplest object in 3D space, a point made up of X, Y, Z coordinates. Connect two Vertices and you have an Edge.

Viewport
The part of the interface where you can see the 3D objects visually, and where you can navigate in 3D space.

Volume
Volume tends to refer to such things as smoke, clouds, and fire. The Principled Volume shader can achieve realistic fire and smoke.

Volumetric
Volumetric can refer to for example Volumetric lighting: the visible beams of light that enter through a window in a room from outside, because of some smoke or dust inside the room.

Weight Map
A weight map is a Vertex Group created using weight painting. What does all that mean? The vertices of a model can have "weight" values between 0 and 1, which can be used in various ways. One way, is to tell an Armature (The skeleton of a model used in animation) how much it can influence those Vertices.

A Weight Map is simply a visual representation of these values, where red is 1 and blue is 0 (with shades of green for the values in between). This visualisation is seen in Weight Paint mode.

Weld
Welding in Blender tends to refer to the welding or merging together of Vertices. You can use a Weld modifier on an object and set a value for how close together Vertices need to be to be merged (or welded) together. This can help eliminate double vertices, or reduce the complexity of a model.

Wireframe
3D-models consist of Faces, Edges and Vertices. Displaying the Edges on top of the model shows us what's known as the "Wireframe". It's an overlay of the Edges of the model.

This is a good way to see the topology of the model, so it's often used to display the work that was done creating nice topology. In Blender you can enable a Wireframe overlay in the viewport overlay options.

World Origin
The center of the 3D world space. Press Shift+S and choose Cursor to World Origin to place the 3D cursor there.

Z-Depth
Z-Depth is the depth of the 3D world from the camera to the distance. You can render this depth as a grayscale image, where black is close to the camera and white is the far distance. With greyscale values in between.

This can be used in compositing, to add mist or other volumetric effects to the image. Z-Depth helps to make the renders look more realistic, and less flat.

ZBrush
ZBrush is software initially intended for the creation of 2D concept art, using a proprietary technology for handling enormous amounts of vertices for the creation of "2,5D" images.

Because of its ability to handle so much geometry however, it has become the industry standard 3D sculpting tool. In ZBrush you can handle 40 million vertices easily on a computer that can't handle that much geometry in other packages.

Nowadays, hardware has caught up, making it possible to handle large vertex counts in other software such as Blender as well.

However, sculpting in ZBrush is still buttery smooth and the brushes work really well. The downside of ZBrush is its legendarily strange User Interface, which hasn't really seen much improvement in 2 decades. See also: Mudbox.

Hi There! That's all for now.
You will be notified of all future updates (an email will be sent to the mail address that you used for your Gumroad or Blender Market purchase). Just in case, I'll mention updates on my social media as well, so you can't miss it.

If you have any questions, feel free to write to hi@blendersecrets.org - I always try to answer everybody but if you mention you bought the book that will help speed up my reply. I really hope you get a lot out of the book and that it will boost your skill level.

You can also always write a comment on Youtube, I normally try to answer all comments and Youtube makes it easy to keep track of them.

If you make something with these tips, please share it with me! I'll be happy to share it on Instagram or Twitter. It's challenging for me to find messages on Instagram due to the way Instagram works (once you have a lot of followers it's almost impossible to find the messages between the likes), so definitely get in touch on the other social platforms instead, with Youtube being the best choice.

instagram.com/blendersecretsdotorg
twitter.com/JanvandenHemel
youtube.com/c/BlenderSecrets
blendersecrets.org

ISBN: 9789464337303