3D Wed Gallery for Game

Title: 3D Web Gallery for Game

By: Ali Akbary

Chapter 6 **MODIFIERS**

Learning Outcome

Objectives of this chapter are: -

- Modifiers
- Types of Modifiers

MODIFIERS

Introduction

Modifiers are automatic operations that affect an object's geometry in a non-destructive way. With modifiers, you can perform many effects automatically that would otherwise be too tedious to do manually (such as subdivision surfaces) and without affecting the base geometry of your object.

Figure 1 Blender Modifiers menu.

They work by changing how an object is displayed and rendered, but not the geometry which you can edit directly. You can add several modifiers to a single object to form The Modifier Stack and Apply a modifier if you wish to make its changes permanent.

They can be added to the active object using the Add Modifier dropdown menu at the top of their properties tab. new modifiers are always added at the bottom of the stack (i.e., will be applied last).

There are four types of modifiers: -

Modify

These are tools similar to the Deform ones (see below), however, they usually do not directly affect the geometry of the object, but some other data, such as vertex groups.

Generate

These are constructive/destructive tools that will affect the whole topology of the mesh. They can change the general appearance of the object, or add new geometry to it…

Deform

Unlike Generate ones above, these only change the shape of an object, without altering its topology.

Modifiers in Action

We will try to apply modifiers to objects

Array Modifier

The Array modifier creates an array of copies of the base object, with each copy being offset from the previous one in any of a number of possible ways. Vertices in adjacent copies can be merged if they are nearby, allowing smooth Subdivision Surface frameworks to be generated.

This modifier can be useful when combined with tileable meshes for quickly developing large scenes. It is also useful for creating complex repetitive shapes.

Multiple Array modifiers may be active for an object at the same time (e.g. to create complex three-dimensional constructs).

Options

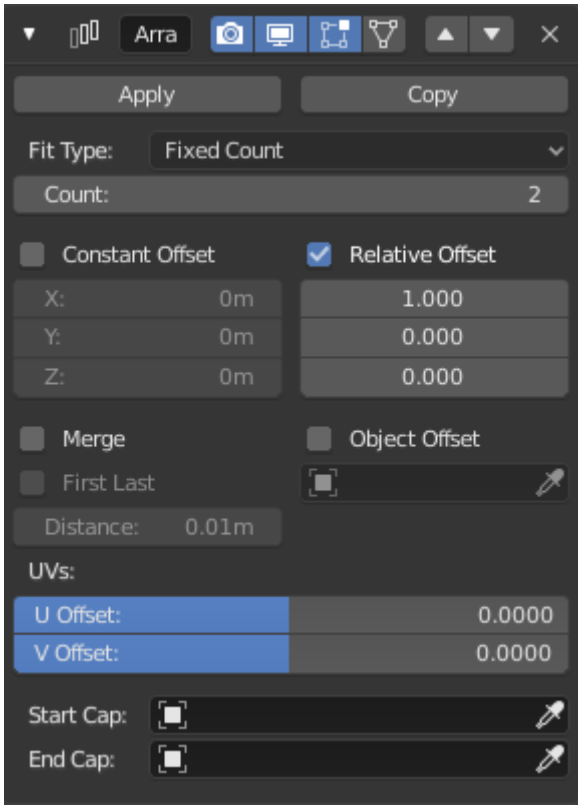


Figure 2 The Array modifier.

Fit Type - Controls how the length of the array is determined. There are three choices, activating respectively the display of the Curve, Length or Count settings explained below:

- Fit Curve - Generates enough copies to fit within the length of the curve object specified in Curve.
- Fit Length - Generates enough copies to fit within the fixed length given by Length.
- Fixed Count - Generates the number of copies specified in Count.

Note

- Both Fit Curve and Fit Length use the local coordinate system size of the base object, which means that scaling the base object in Object Mode will not change the number of copies generated by the modifier.
- Fit Length uses the local coordinate system length of the curve, which means that scaling the curve in Object Mode will not change the number of copies generated by the modifier.
- Applying the scale can be useful for both.

Offset

Constant Offset, X, Y, Z - Adds a constant translation component to the duplicate object's offset. X, Y and Z constant components can be specified.

Relative Offset, X, Y, Z - Adds a translation equal to the object's bounding box size along each axis, multiplied by a scaling factor, to the offset. X, Y and Z scaling factors can be specified.

Figure 3 Relative offset (0.5, 1.0 and 1.5) examples.

Object Offset - Adds a transformation taken from an object (relative to the current object) to the offset. It is good practice to use an empty object centered or near to the initial object. E.g. by rotating this empty a circle or helix of objects can be created.

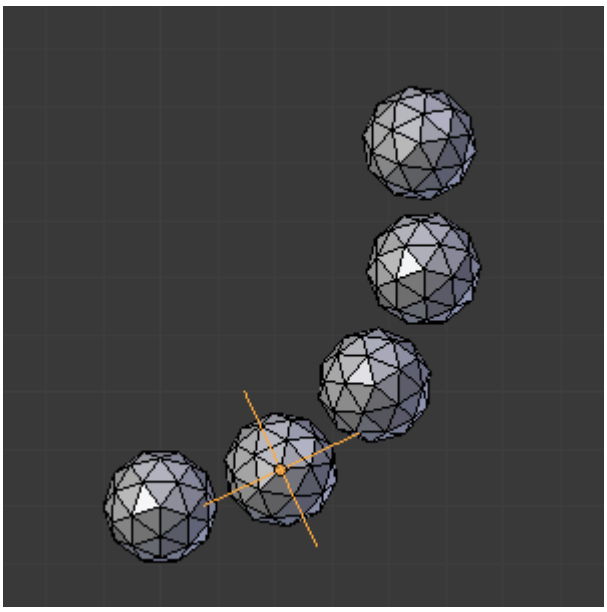


Figure 4 Object offset example.

Merge - If enabled, vertices in each copy will be merged with vertices in the next copy that are within the given Distance.

- First Last - If enabled and Merge is enabled, vertices in the first copy will be merged with vertices in the last copy, again if they are within Distance range. This is useful for circular objects.

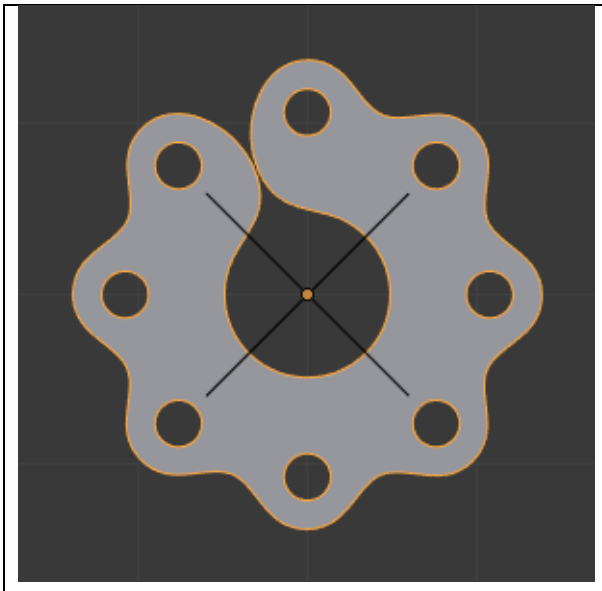


Figure 5 Subdivision discontinuity caused by not merging vertices between first and last copies (First Last off).

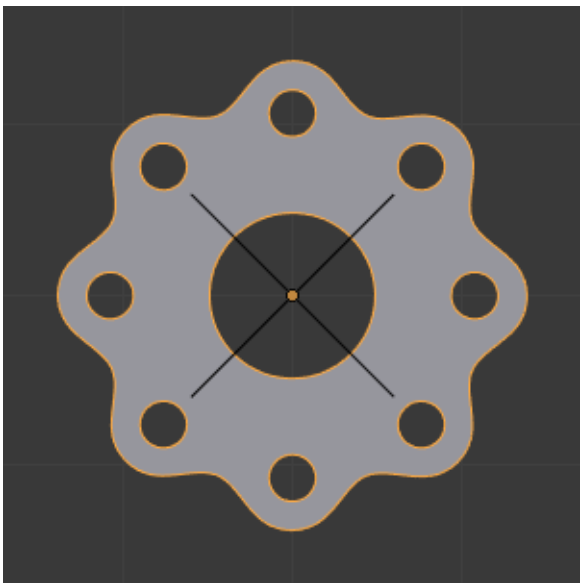


Figure 6 Subdivision discontinuity eliminated by merging vertices between first and last copies (First Last on).

Distance - Controls the merge distance for Merge and First Last.

UVs

U Offset, V Offset - Shifts UVs of each new duplicate by a settable amount.

Cap

Start Cap / End Cap - This allows either end points of the array to have a different mesh subsisted.

For the start: as if it was in position -1, i.e., one "array step" before the first "regular" array copy. For the end: as if it was in position n + 1, i.e., one "array step" after the last "regular" array copy.

When Merge is activated, and the cap vertices are within the Distance threshold, they will be merged.

Note

The start/end cap objects currently do not support the First Last option.

Hints

Offset Calculation

The transformation applied from one copy to the next is calculated as the sum of the three different components (Relative, Constant and Object), each of which can be enabled/disabled independently of the others. This allows, for example, a relative offset of (1.0, 0.0, 0.0) and a constant offset of (0.1, 0.0, 0.0), giving an array of

objects neatly spaced along the X axis with a constant 0.1 unit between them, whatever the original object's size.

Examples

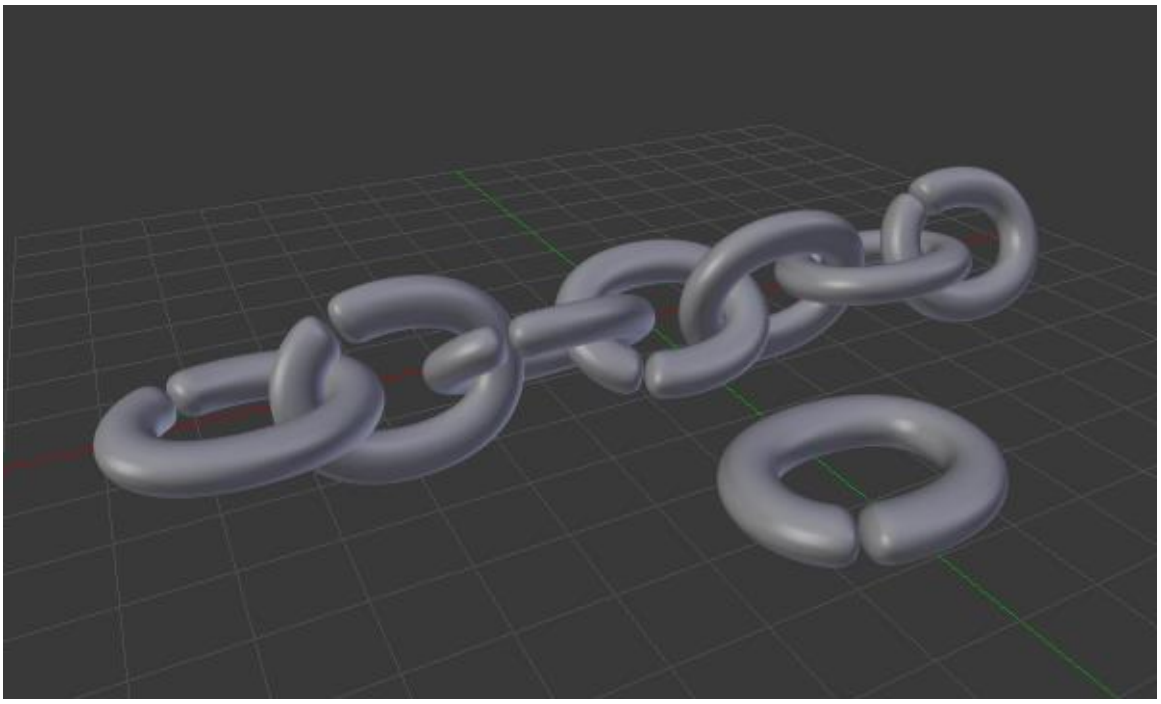


Figure 7 A chain created from a single link

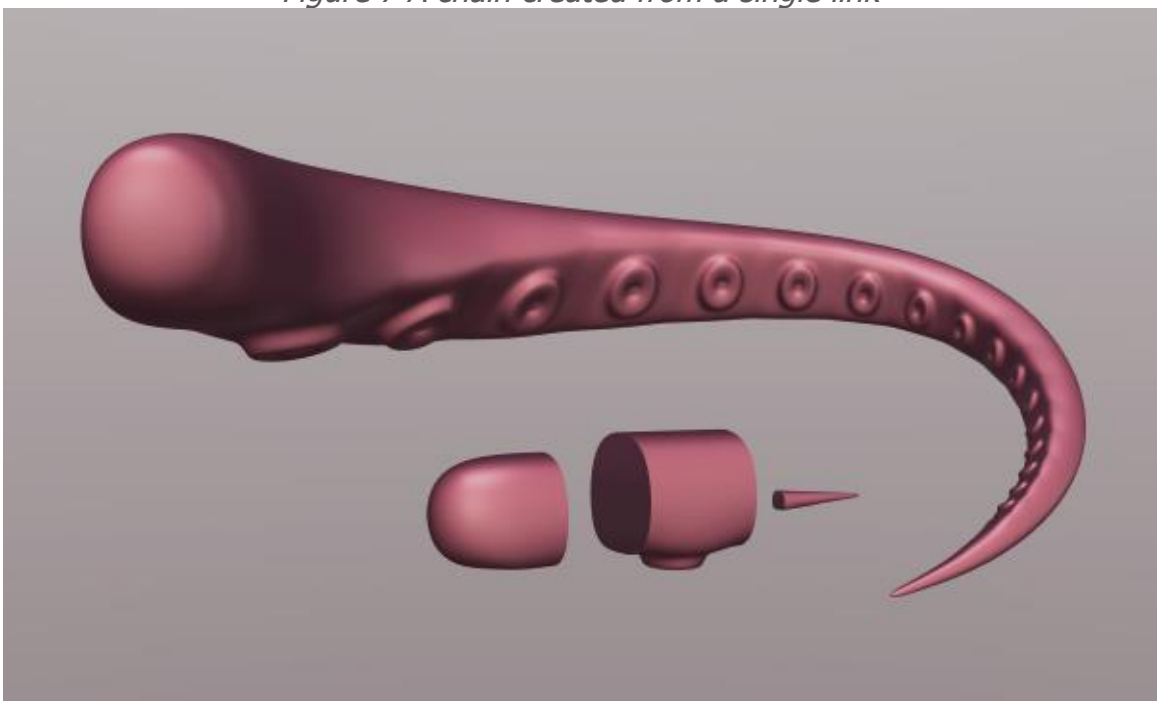


Figure 8 A tentacle created with an Array Modifier followed by a Curve Modifier.

The segment in the foreground is the base mesh for the tentacle; the tentacle is capped by two specially-modelled objects deformed by the same Curve object as the main part of the tentacle. Sample blend-file.

Fractal

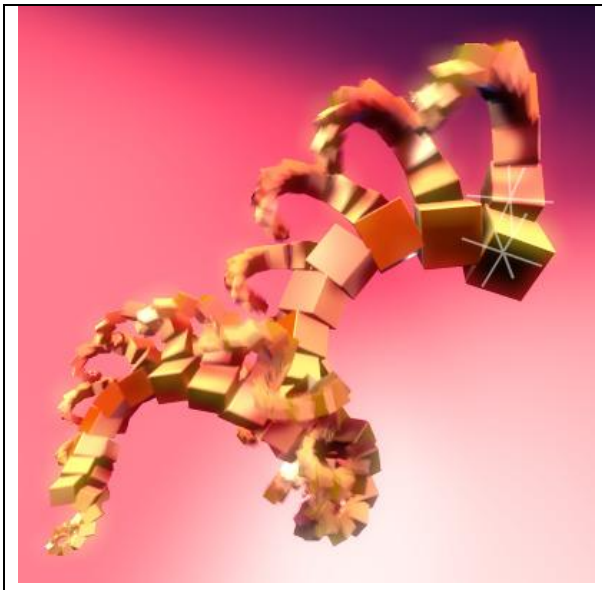


Figure 9 multi-level array animated with motion blur.

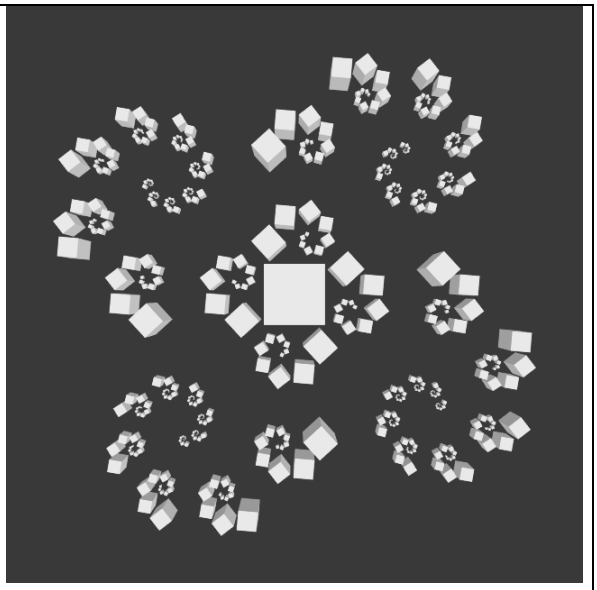


Figure 10 Fractal created with multiple arrays.

Bevel Modifier

The Bevel modifier bevels the edges of the mesh it is applied to, with some control of how and where the bevel is applied to the mesh.

It is a non-destructive alternative to the Bevel Operation in Edit Mode.

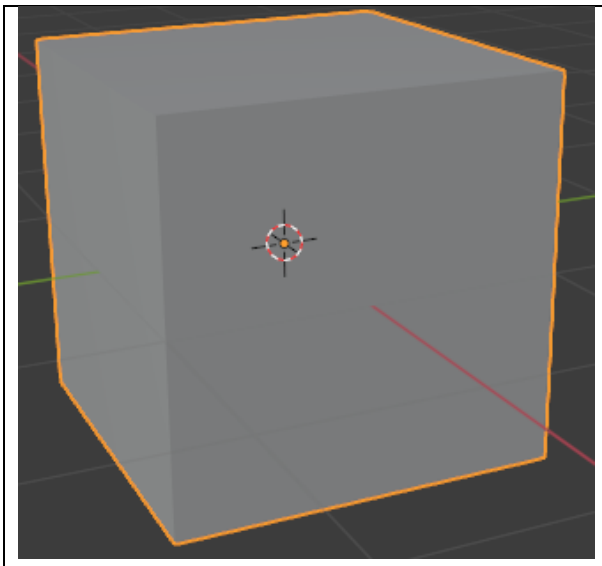


Figure 11 Not beveled.

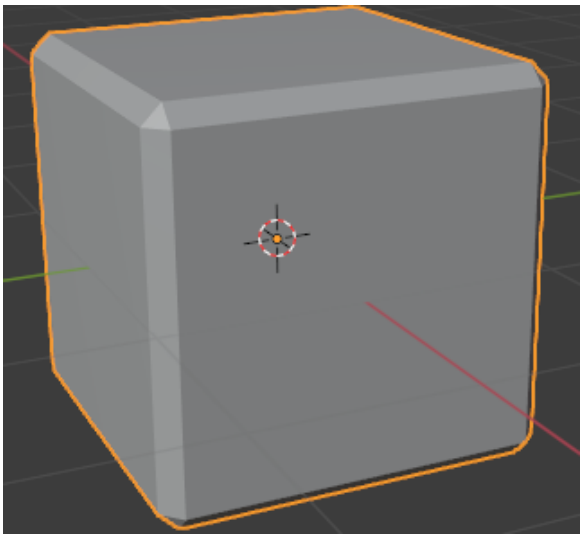


Figure 12 Beveled.

Options

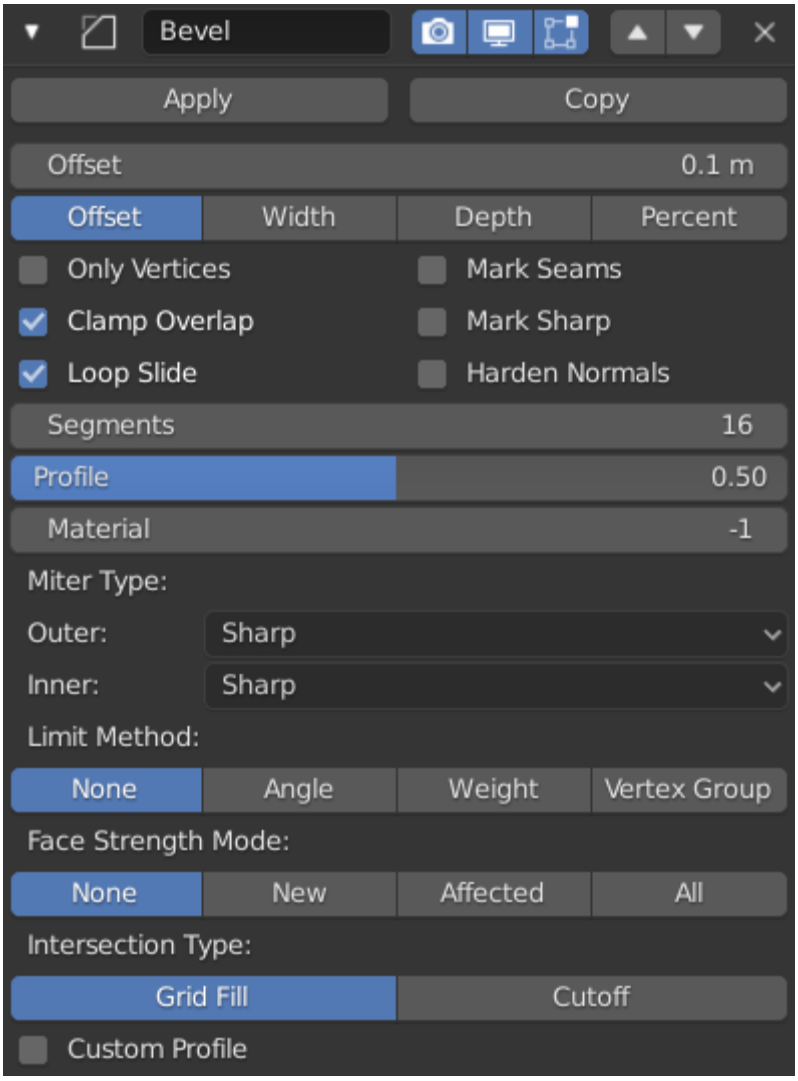


Figure 13 The Bevel modifier.

Width - The size of the bevel effect. See Width Method below.

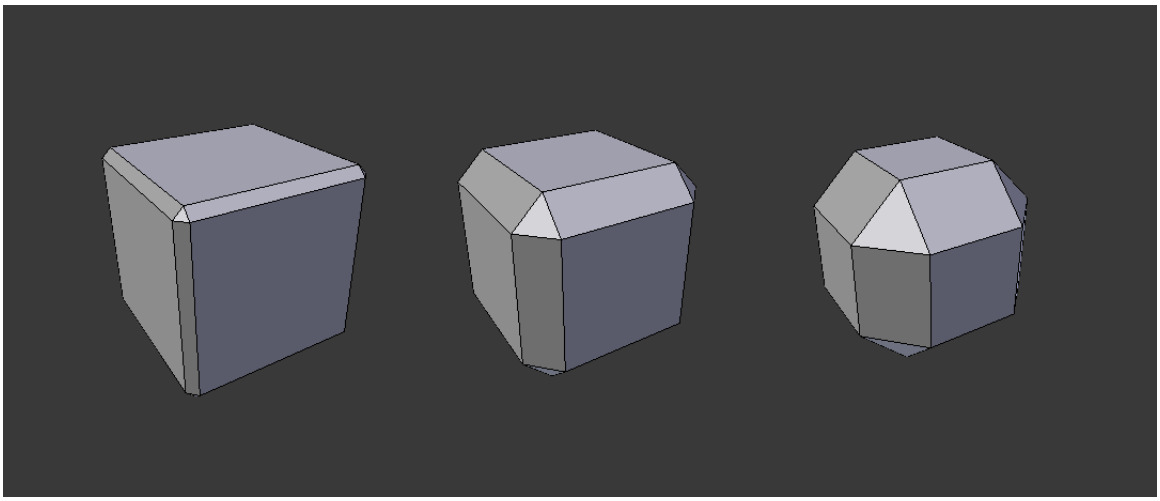


Figure 14 Three Cubes with 0.1, 0.3 and 0.5 bevel widths.

Segments - The number of edge loops added along the bevel's face.

Profile - The shape of the bevel, from concave to convex. It has no effect if Segments is less than 2.

Material - The index of the material slot to use for the bevel. When set to -1, the material of the nearest original face will be used.

Only Vertices - When enabled, only the areas near vertices are bevelled, the edges remain unchanged.

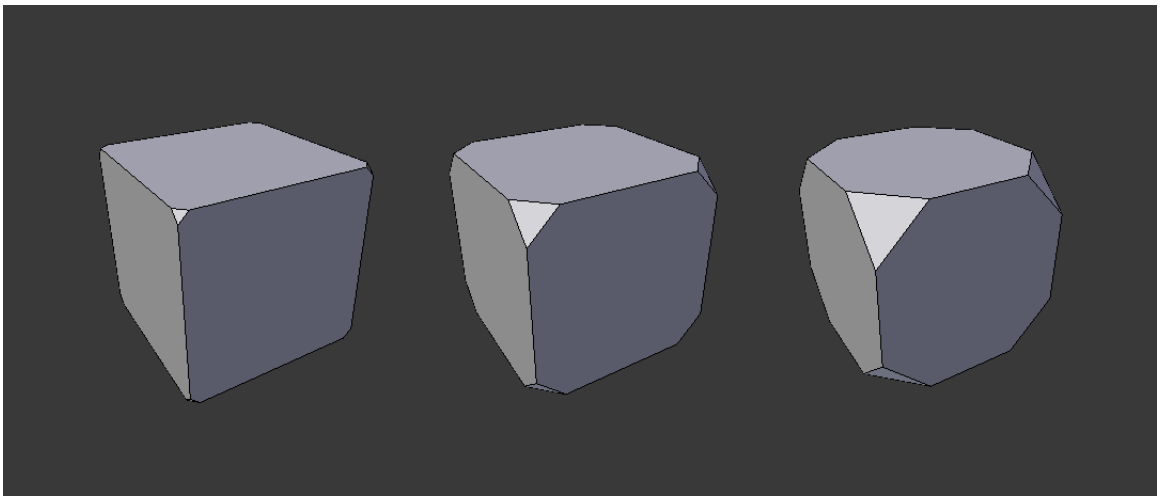


Figure 15 Three cubes with 0.1, 0.3 and 0.5 bevel widths, with Only Vertices option enabled.

Clamp Overlap - Limits the width of each bevelled edge so that edges cannot cause overlapping intersections with other geometry.

Loop Slide - If there are unbevelled edges along with bevelled edges into a vertex, the bevel tries to slide along those edges when possible. Turning the option off can lead to more even bevel widths.

Mark Seams - If a seam edge crosses a non-seam one and you bevel all of them, this option will maintain the expected propagation of seams.

Mark Sharp - Similar to Mark Seams, but for sharp edges.

Harden Normals - When enabled, the per-vertex face normals of the bevel faces are adjusted to match the surrounding faces, and the normals of the surrounding faces are not affected. This will keep the surrounding faces flat (if they were before), with the bevel faces shading smoothly into them. For this effect to work, you need custom normals data, which requires Auto Smooth option to be enabled (see Normals).

Limit Method - Used to control where a bevel is applied to the mesh.

None - No limit, all edges will be bevelled.

Angle - Only edges where the adjacent faces form an angle smaller than the defined threshold will be bevelled. Intended to allow you to bevel only the sharp edges of an object without affecting its smooth surfaces.

Weight - Use each edge's bevel weight to determine the width of the bevel. When the bevel weight is 0.0, no bevel is applied. See here about adjusting bevel weights.

Vertex Group - Use weights from a vertex group to determine the width of the bevel. When the vertex weight is 0.0, no bevel is applied. An edge is only bevelled if both of its vertices are in the vertex group. See here about adjusting vertex group weights.

Invert - Inverts the influence of the selected vertex group, meaning that the group now represents vertices that will not be deformed by the modifier.

The setting reverses the weight values of the group.

Width Method - Declares how Width will be interpreted to determine the amount of bevel.

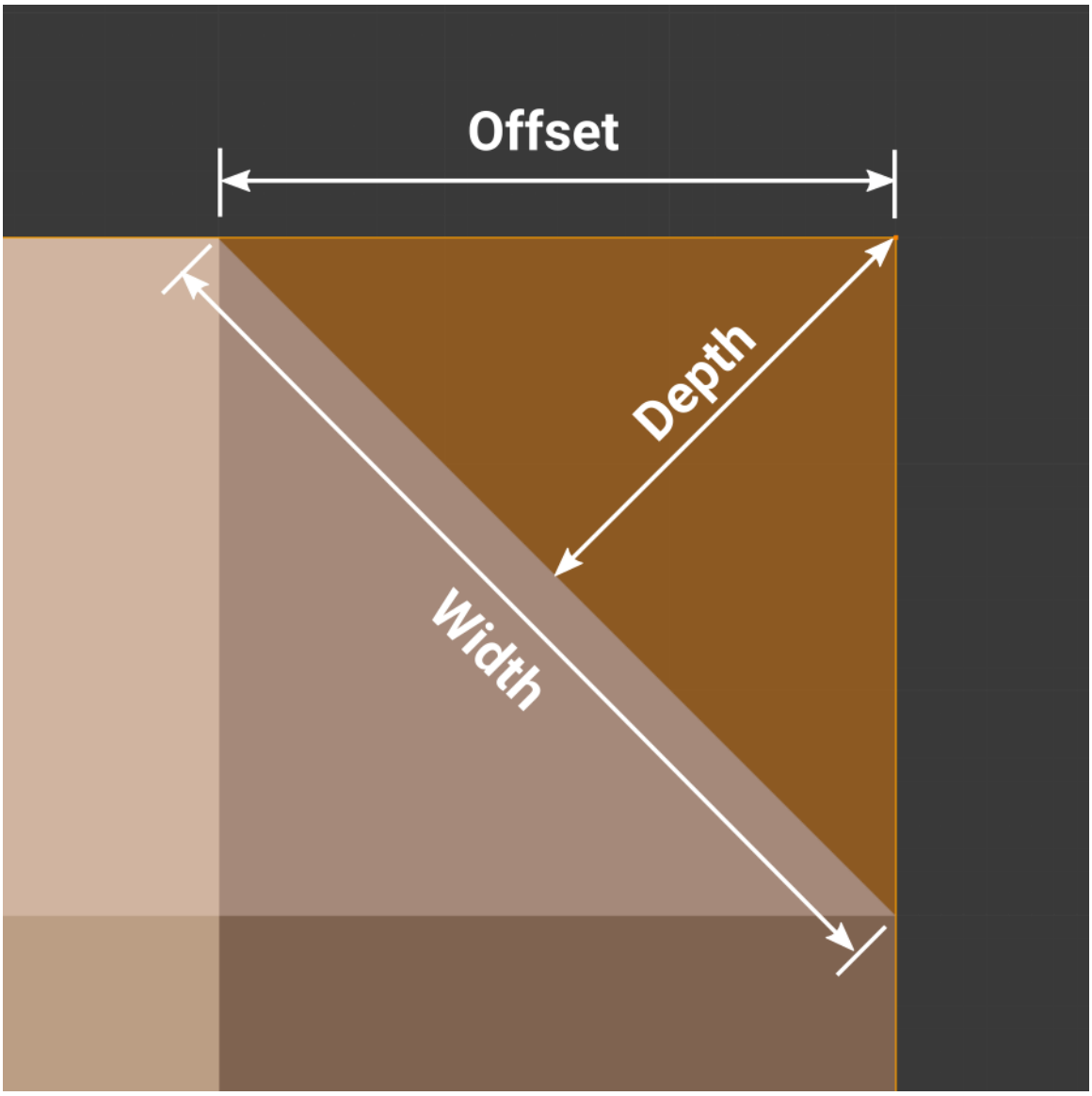


Figure 16 Width methods.

- Offset - Value is interpreted as the distance from the original edge to the edge of the bevelled face.

- Width - Value is interpreted as the distance between the two new edges formed by the bevel.
- Depth - Value is the perpendicular distance from the new bevel face to original edge.
- Percent - Similar to Offset but the value is interpreted as a percentage of the adjacent edge length.

Set Face Strength Mode - Set Face Strength on the faces involved in the bevel, according to the mode specified here. This can be used in conjunction with a following Weighted Normals modifier (with the Face Influence option checked).

- None - Do not set face strength.
- New - Set the face strength of new faces along edges to Medium, and the face strength of new faces at vertices to Weak.
- Affected - In addition to those set for the new case, also set the faces adjacent to new faces to have strength Strong.
- All - In addition to those set for the Affected case, also set all the rest of the faces of the model to have strength Strong.

Miter Patterns - A miter is formed when two bevelled edges meet at an angle. On the side where the angle is greater than 180 degrees, if any, it is called an outer miter. If it is less than 180 degrees, then it is called an inner miter. The outer and inner miters can each be set to one of these patterns:

Sharp - Edges meet at a sharp point, with no extra vertices introduced on the edges.

Patch - Edges meet at a sharp point but in addition, two extra vertices are introduced near the point so that the edges and faces at the vertex may be less pinched together than what occurs in the Sharp case. This pattern does make no sense for inner miters, so it behaves like Arc for them.

The Spread slider controls how far the new vertices are from the meeting point.

Arc - Two vertices are introduced near the meeting point, and a curved arc joins them together.

The Spread slider controls how far the new vertices are from the meeting point.

The Profile slider controls the shape of the arc.

Spread - The value used to spread extra vertices apart for non-sharp miters.

Intersection Method - When more than two beveled edges meet at a vertex, a mesh is created as a way to complete the intersection between the generated geometry. This option controls the method used to create that mesh.

Grid Fill - The default method for building intersections, useful when a smooth continuation of the bevel profile is desired. Without Custom Profile enabled, the curve of the profile continues through the intersection, but with a custom profile it just creates a smooth grid within the boundary of the intersection.

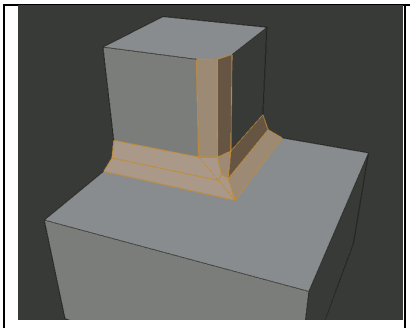


Figure 17 Sharp outer miter.

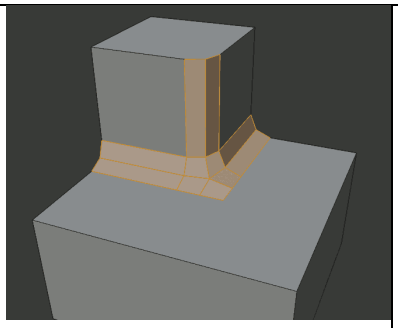


Figure 18 Patch outer miter.

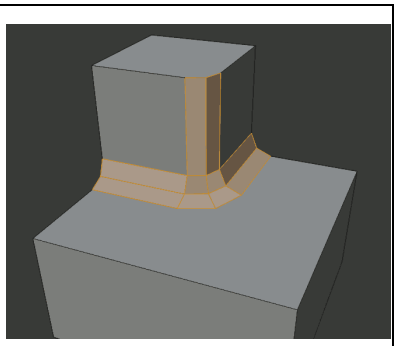


Figure 19 Arc outer miter

Cutoff - Creates a cutoff face at the end of each beveled edge coming into the vertex. This is most useful for custom profiles when the new intersection is too complex for a smooth grid fill.

With a three-way intersection, when the inner corners of the cutoff profiles faces meet at the same location, no center face is created.

The direction of the cutoff faces depends on the original vertex's normal.

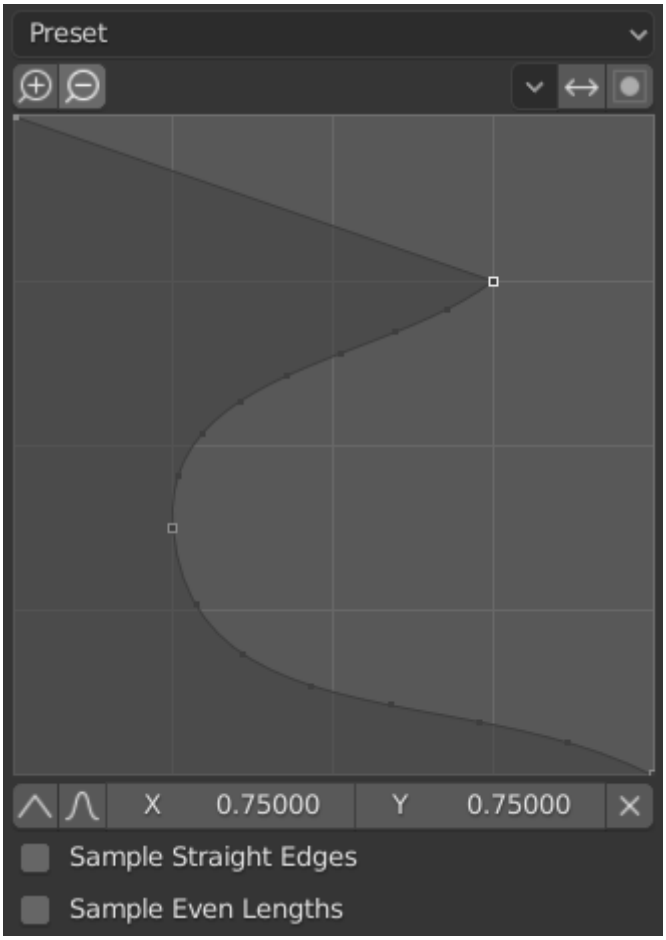


Figure 20 The custom profile widget.

This widget allows the creation of a user-defined profile with more complexity than with the single profile parameter. The modal tool allows toggling the custom profile, but the shape of the profile is only editable in the options panel after the operation is confirmed.

The profile starts at the bottom right of the widget and ends at the top left, as if it were between two edges meeting at a right angle. Control points are created in the widget and then the path is sampled with the number of segments from the bevel modifier.

- Presets - The Support Loops and Steps presets are built dynamically depending on the number of segments in the bevel. If the number of segments is changed, the preset will have to be re-applied.
- Reverse - The Reverse button flips the orientation of the profile for all bevelled edges.
- Clipping - The Clipping toggle allows control points to be moved beyond the initial boundary, allowing the bevel to add volume to the mesh rather than just removing it.

Note

The Profile slider is still useful when miters are enabled because it still controls the shape of the miter profiles.

Sampling - Samples will first be added to each control point, then if there are enough samples, they will be divided evenly between the edges. The Sample Straight Edges option toggles whether the samples are added to edges with sharp control points on either side. If there aren't enough samples to give each edge the same number of samples, they will just be added to the most curved edges, so it is recommended to use at least as many segments as there are control points.

Boolean Modifier

The Boolean modifier performs operations on meshes that are otherwise too complex to achieve with as few steps by editing meshes manually. It uses one of the three available Boolean operations to create a single mesh out of two mesh objects:

This modifier needs a second object to be the target (the second operand) of the operation.

Warning

Only manifold meshes are guaranteed to give proper results, other cases (especially "opened" meshes, non-manifold but without any self-intersections) will usually work well, but might give odd glitches and artifacts in some cases.

You should also avoid any co-planar faces (or co-linear edges) between both operands, those also tend to give issues currently.

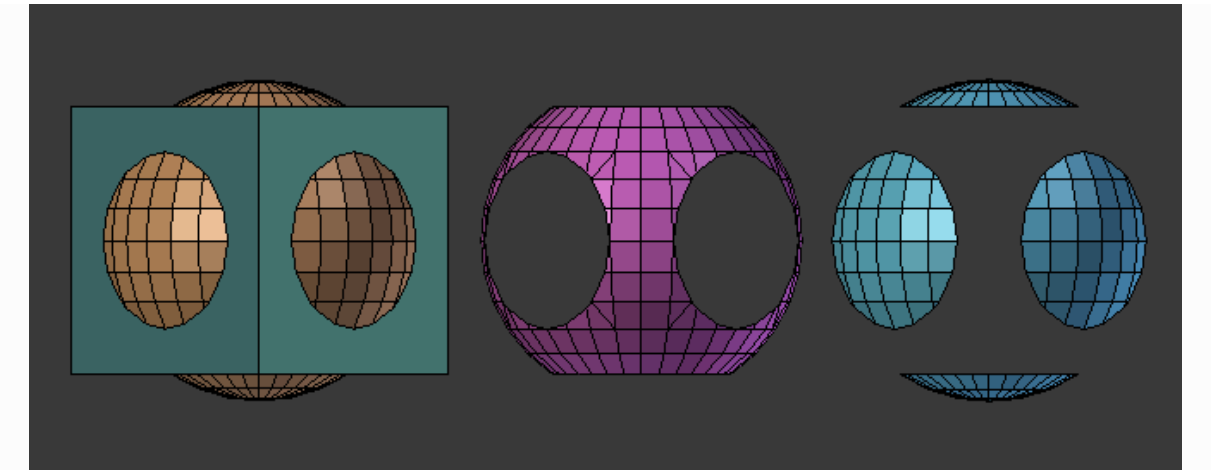


Figure 21 The Union, Intersection and Difference between a Cube and a UV Sphere, with the modifier applied to the sphere and using the cube as target.

Tip

If you have marked your objects to show the edges (in Properties Editor ▸ Object ▸ Viewport Display, enable Wireframe), you will see the edge creation process while you are moving your objects around. Depending on your mesh topology, you can also enable X-Ray and Transparency and see the topology being created in real-time.

Options

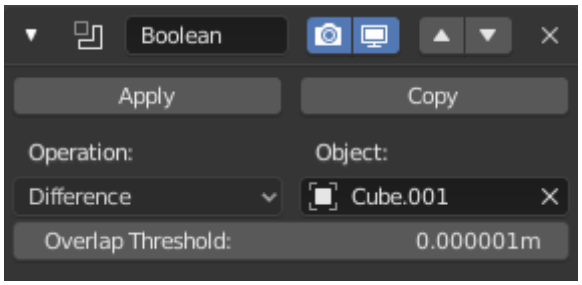


Figure 22 The Boolean modifier.

Operations

Operation - Which Boolean operation will be used.

- Difference - The target mesh is subtracted from the modified mesh (everything outside of the target mesh is kept).
- Union - The target mesh is added to the modified mesh.
- Intersect - Opposite of Difference (everything inside of the target mesh is kept).

Object - The name of the target mesh object.

Overlap Threshold - Maximum distance between two faces to consider them as overlapping.

Mirror Modifier

The Mirror modifier mirrors a mesh along its local X, Y and/or Z axes, across the Object Origin. It can also use another object as the mirror center, then use that object's local axes instead of its own.

Options

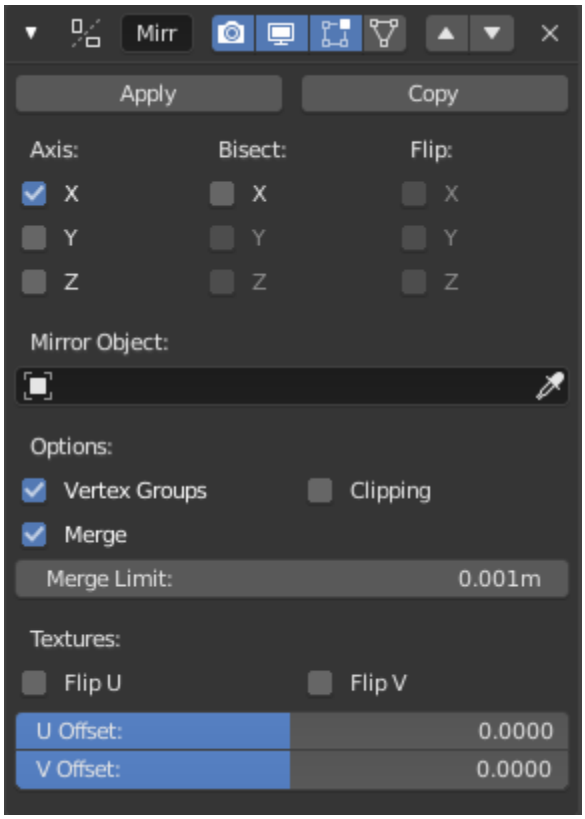


Figure 23 The Mirror modifier.

Axis - The X, Y, Z axis along which to mirror, i.e., the axis perpendicular to the mirror plane of symmetry.

To understand how the axis applies to the mirror direction, if you were to mirror on the X axis, the positive X values of the original mesh would become the negative X values on the mirrored side.

You can select more than one of these axes. And will then get more mirrored copies. With one axis you get a single mirror, with two axes four mirrors, and with all three axes eight mirrors.

Bisect - If the mesh is already on both sides of the mirror plane, it is cut by that plane, and only one side (the "negative" one by default) is kept to perform the mirror process.

Flip - When bisect is enabled on an axis, you can use this setting to switch the side kept and mirrored (i.e., when it is enabled, the "positive" side will be kept, instead of the "negative" one).

Mirror Object - An Object Selector to select an object (usually an empty), which position and rotation will be used to define mirror planes (instead of using the ones from the modified object).

You can animate it to animate the mirror effect.

Vertex Groups - Try to mirror existing vertex groups.

A very nice feature, but one that has very specific prerequisites: -

- The vertex groups you want to mirror must be named following the usual left/right pattern (i.e., suffixed by something like ".R", ".right", ".L", etc.).
- The mirror side vertex group must already exist (it will not be created automatically). It must also be completely empty (no vertices assigned to it).

Merge - Where a vertex is in the same place (within the Merge Limit distance) as its mirror it will be merged with the mirrored vertex.

Merge Limit - The maximum distance between a vertex and its mirror copy at which they are merged together (being snapped on the mirror plane). Needs Merge to be enabled.

Clipping - Prevents vertices from moving through the mirror plane(s) while the user is transforming them in Edit Mode.

If it is enabled but vertices are beyond the mirror plane and outside of the Merge Limit, the vertices will not be merged. But as soon as the vertices are within Merge Limit they are snapped together and cannot be moved beyond the mirror plane.

Note

Vertices on the mirror plane will be unable to move away from the mirror plane as long as Clipping is enabled. You must disable it to be able to move the vertices along the mirror axis again.

Flip UV - With this option you can mirror the UV texture coordinates across the middle of the image.

E.g., if you have a vertex with UV coordinates of (0.3, 0.9), its mirror copy will have UV coordinates of (0.7, 0.1).

UV Offsets - Amount to shift mirrored UVs on the U/V axes.

It's useful for baking (as overlapping UVs can cause artifacts to appear in the baked map), so the UVs can be moved outside the image and not used for baking, but still be used for display.

Hints

Many modelling tasks involve creating objects that are symmetrical. This modifier offers a simple and efficient way to do this, with real-time update of the mirror as you edit it. Once your modelling is completed you can either click Apply to make a real version of your mesh, or leave it as-is for future editing.

Accurately Positioning the Mirror Plane

To apply a Mirror modifier, it is common to have to move the object's origin onto the edge or face that is to be the axis for mirroring. This can be tricky when attempted visually.

A good technique to achieve an exact position is to select the edge, then snap Cursor to Selection. This will position the 3D Cursor in the center of the edge.

Finally, use the Set Origin menu, and select Origin to 3D Cursor. This will move the object's origin (and thus, the mirror plane) to where the 3D cursor is located, and the mirroring will be exact.

An alternative is to use an empty as a Mirror Object that you move to the correct position.

Screw Modifier

The Screw modifier is similar to the Screw tool in the Toolbar, in that it takes a profile object, a mesh or a curve, to create a helix-like shape.

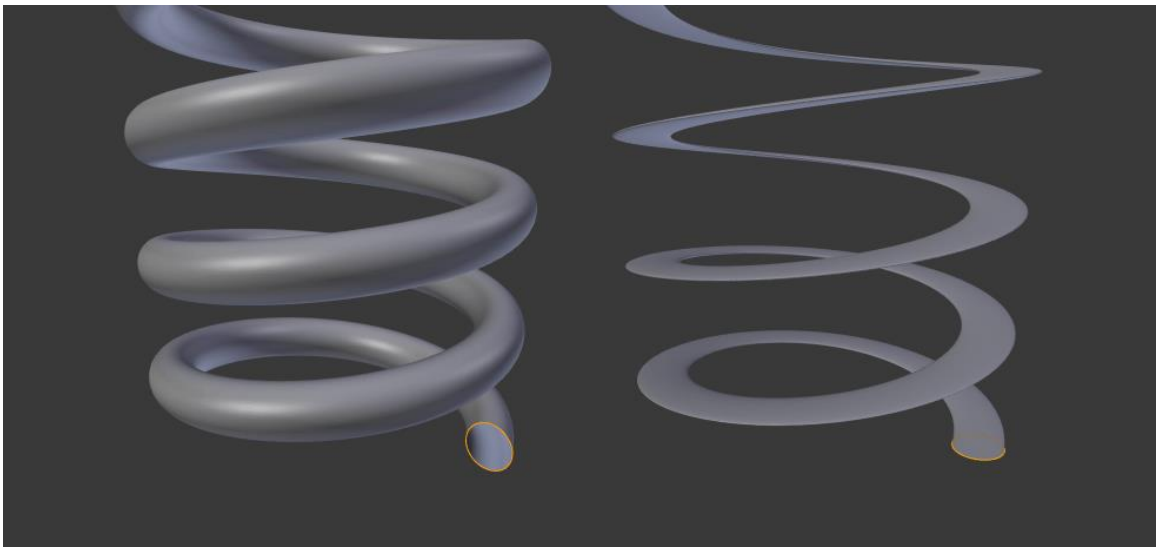


Figure 24 Properly aligning the profile object is important.

The profile should be properly aligned to the cardinal direction of the object rather than to the screw axis.

Options

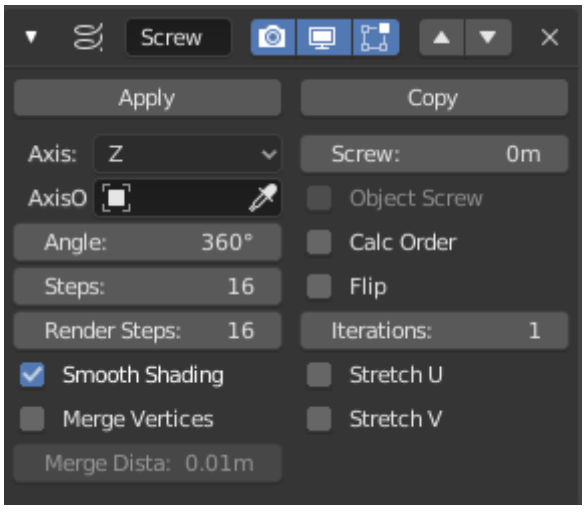


Figure 25 The Screw modifier.

Axis - The axis along which the helix will be built.

- Screw - The height of one helix iteration.

Axis Object - The name of an object to define the axis direction.

- Object Screw - Use the distance from the Axis Object to define the height of one helix iteration.

Angle - Degrees for a single helix revolution.

Steps - Number of steps used for a single revolution displayed in the 3D Viewport. Beware of setting this higher than Render Steps, which is the value used for rendering.

Render Steps - As above, but used during render time. Increase to improve quality.

Smooth Shading - Output faces with smooth shading rather than flat shading. The smooth/flat shading of the input geometry is not preserved.

Calculate Order - Order of edges is calculated to avoid problems with normals and shading. Only needed for meshes, not curves.

Flip - Flip normals direction.

Iterations - Number of revolutions.

Stretch U/V - Stretch the UV coordinates from (0.0 to 1.0) when UVs are present.

Merge Vertices - Merge vertices that lie on the axis of rotation. Use this to close off end points with a triangle fan.

- Merge Distance - Vertices under this distance to the axis are merged.

Solidify Modifier

The Solidify modifier takes the surface of any mesh and adds depth, thickness to it.

Mode Independent Properties

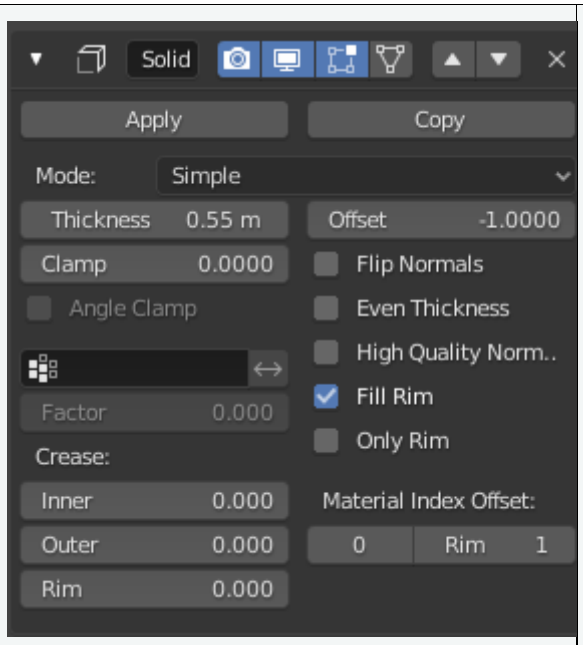
Thickness - The depth to be solidified.

Important

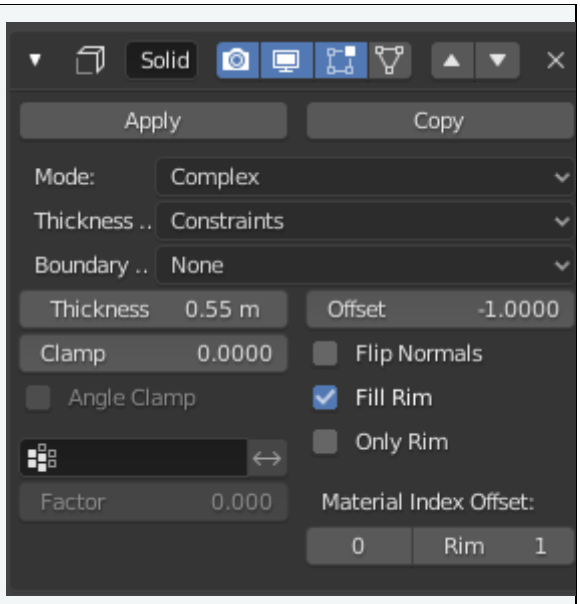
The modifier thickness is calculated using local vertex coordinates. If the object has a non-uniform scale, the thickness will vary on different sides of the object.

To fix this, either Apply or Clear the scale.

Options



The Solidify modifier in simple mode.



The Solidify modifier in complex mode.

Clamp - A value between (0 to 2) to clamp offsets to avoid self-intersection. The amount is determined by the length of the shortest adjacent edge.

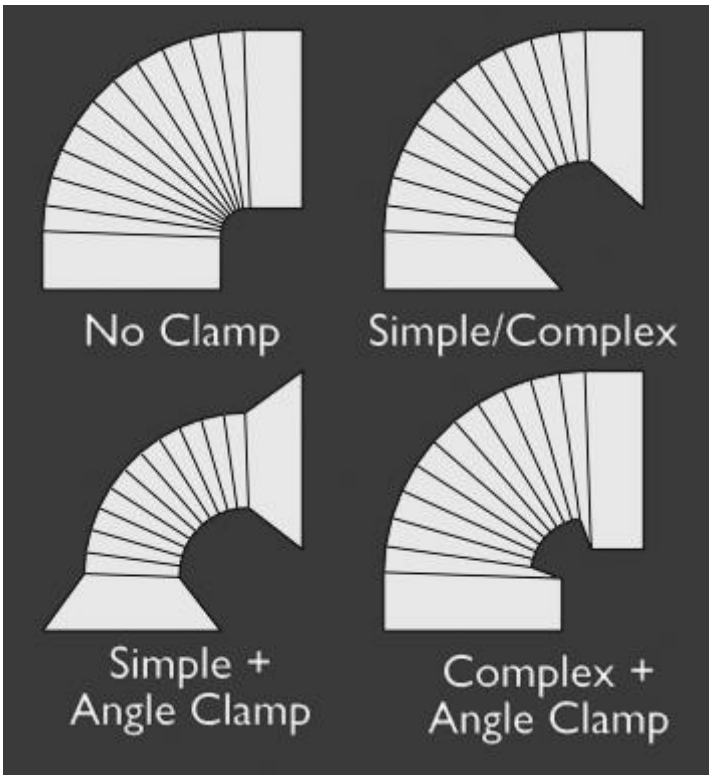


Figure 26 Clamp Offset.

- Angle Clamp - If enabled clamping will also consider angles in the geometry, not only lengths.

Vertex Group - Only vertices in this group are solidified. Their weights are multiplied by the thickness, so vertices with lower weights will be less thick.

- Invert - Reverses the vertex group, so that only vertices which are not in the vertex group are solidified.
- Factor - How much the vertex weights are taken into account.
    - On 0.0, vertices with zero weight will have no thickness at all (creates duplicate vertices).
    - On 0.5, vertices with zero weight will be half as thick as those with full weight.
    - On 1.0, the weights are ignored and the Thickness value is used for every vertex.

Bevel Convex - Edge bevel weight to be added to outside edges.

Offset - A value between (-1 to 1) to locate the solidified output inside or outside the original mesh. The inside and outside is determined by the face normals. Set to 0.0, the solidified output will be centered on the original mesh.

Flip Normals - Reverse the normals of all geometry (both the inner and outer surfaces).

Fill Rim - Fills the gap between the inner and outer edges.

Only Rim - In Simple Mode: Will not extrude surfaces parallel to the original one, but instead will only add the perpendicular rim.

In Complex Mode: Will only leave the generated perpendicular rim.

Note

Fill Rim and Only Rim only make a difference on non-manifold objects, since the rims are generated from the borders of the original geometry.

Material Index Offset

Choose a different material to use for the new geometry. This is applied as an offset from the original material of the face from which it was solidified.

- A value of 0 means it will use the same material.
- A value of 1 means it will use the material immediately below the original material.
- A value of -2 means the material two positions above the original material will be used.

These are clamped to the top-most and bottom-most material slots.

- Rim - Similarly, you can give another material to the rim faces.

Shell Vertex Group - Vertex group that the generated shell geometry will be weighted to. This allows you to use other modifiers to only affect the shell geometry by using a that modifier's vertex group influence control.

Rim Vertex Group - Same as Shell Vertex Group, but for the generated rim geometry.

Simple Mode

This is the default solidify algorithm, which simply extrudes the geometry. This algorithm does not work on geometry where edges have more than two adjacent faces.

Crease - These options are intended for usage with the Subdivision Surface modifier.

Figure 27 Edges which will get creases marked.

- Inner - Set a crease to the inner edges.
- Outer - Set a crease to the outer edges.
- Rim - Set a crease to the rim.

Even Thickness - Maintain thickness by adjusting for sharp corners. Sometimes improves quality but also increases computation time.

High Quality Normals - Normals are calculated to produce a more even thickness. Sometimes improves quality but also increases computation time.

Important

If the normals of adjacent faces don't point into the same general direction, simple mode will not be able to solidify the boundary between those. This happens if the normals are not recalculated or for example on one-sided surfaces like a Möbius strip.

Complex Mode

This is a new solidify algorithm which can handle every geometric situation to guarantee a manifold output geometry. This algorithm is able to solidify shapes like Möbius strips, Klein bottles, architectural wall layouts and many more which the standard implementation isn't able to do. If the special cases are not present it is recommended to choose Simple Mode because the extra logic makes this algorithm much slower.

Note

There are no options for crease in the Modifier tab because crease is handled in a dynamic way. The modifier will transfer the creases of the original mesh in a smart way to the output mesh to work with the Subdivision Surface modifier.

Thickness Mode - Choose the kind of thickness handling (thickness solver)

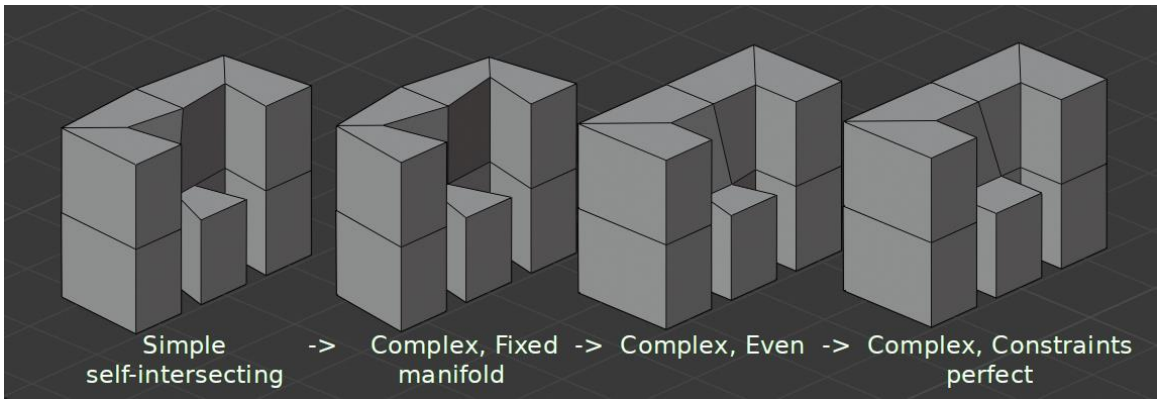


Figure 28 Different thickness options on a non-manifold mesh.

- Fixed - This is similar to Simple Mode without Even Thickness. The new vertices are always in a fixed distance to the old ones.
- Even - This is similar to Simple Mode with Even Thickness and High Quality Normals. It adjusts for sharp corners, but may not always work when more than three faces come together.
- Constraints - This is a more advanced model to try to always get the perfect thickness everywhere. For up to three faces, it is always guaranteed to find a perfect solution.

Boundary Shape - Choose the kind of boundary that suits the model the most.

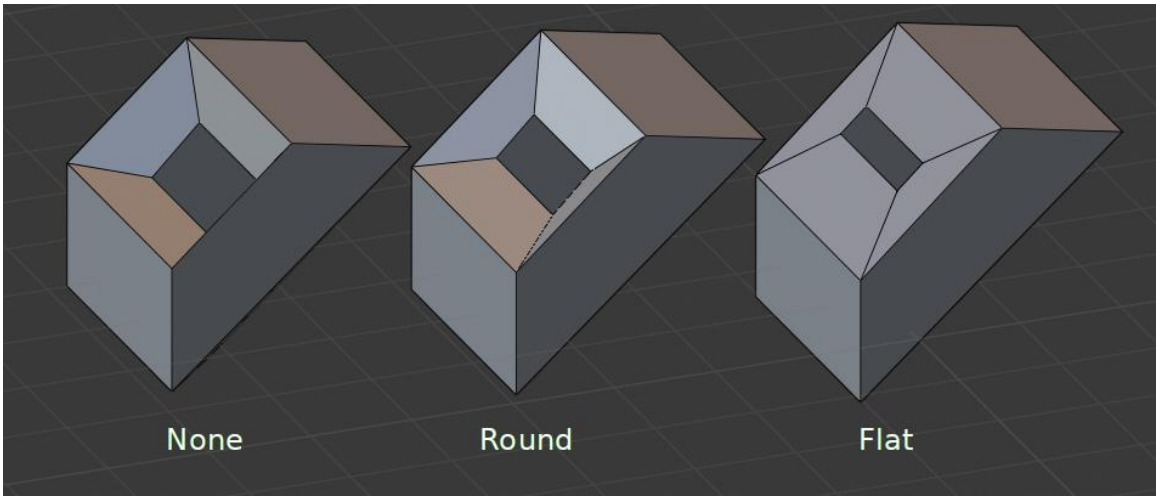


Figure 29 Different boundary options with a matCap.

- None - No boundary fix is applied. Results are stable.
- Round - Adjusts the boundary for an opening to face inwards (like a hole in an egg).
- Flat - Adjusts the boundary of a planar opening to be a flat (like a cut sphere).

Flat Faces - Make faces use the minimal vertex weight assigned to their vertices to ensure that new faces remain parallel to their original ones. This is slow, so disable it when it is not needed.

Merge Threshold - Distance within which degenerated geometry is merged.

Known Limitations

Even Thickness - Solidify thickness is an approximation. While Even Thickness and High Quality Normals should yield good results, the final wall thickness is not guaranteed and may vary depending on the mesh topology. Especially for vertices with more than three adjacent faces.

In order to maintain a precise wall thickness in every case, we would need to add/remove faces on the offset shell, something this modifier does not do since this would add a lot of complexity. The best option to preserve wall thickness is complex mode with constraints thickness mode, but it is also not guaranteed to work perfect in every case.

Subdivision Surface Modifier

The Subdivision Surface modifier (often shorten to "Subdiv") is used to split the faces of a mesh into smaller faces, giving it a smooth appearance. It enables you to create complex smooth surfaces while modelling simple, low-vertex meshes. It avoids the need to save and maintain huge amounts of data, and gives a smooth "organic" look to the object.

As with any modifier, order of execution (position in the modifier stack) has an important bearing on the results.

Keep in mind that this is a different operation than its companion, Smooth Shading. You can see the difference between the two in the grid image below.

Figure 30 Subdivision levels 0 to 3, without and with Smooth Shading.

Tip

The Subdivision Surface modifier does not allow you to edit the new subdivided geometry without applying it, but the Multiresolution modifier does (in Sculpt Mode).

Note

This modifier now uses the OpenSubdiv library as a backend.

Options

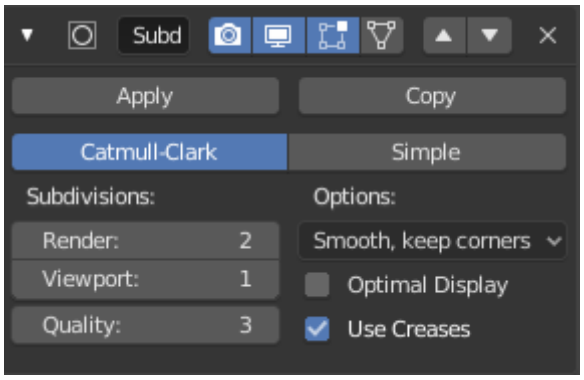


Figure 31 The Subdivision Surface modifier.

Type - This toggle button allows you to choose the subdivision algorithm:

- Catmull-Clark - The default option, subdivides and smooths the surfaces. According to its Wikipedia page, the "arbitrary-looking formula was chosen by Catmull and Clark based on the aesthetic appearance of the resulting surfaces rather than on a mathematical derivation."
- Simple - Only subdivides the surfaces, without any smoothing (the same as the Subdivide operator, in Edit Mode). Can be used, for example, to increase base mesh resolution when using displacement maps.

Subdivisions - Recursively adds more geometry.

The right combination of these settings will allow you to keep a fast and lightweight approximation of your model when interacting with it in the 3D Viewport, but use a higher quality version when rendering.

Warning

Higher levels of subdivisions result in more vertices, which means more memory will be occupied (both system RAM, and video memory for display). Blender could potentially crash or hang if you do not have enough available memory.

Render - The number of subdivision levels shown in renders.

Viewport - The number of subdivision levels shown in the 3D Viewport.

Quality - How precisely the vertices are positioned (relatively to their theoretical position of an infinitely subdivided mesh), can be lowered to get a better performance.

Using higher values does not necessarily mean real improvement in quality, ideal results might be reached well before the maximum Quality value.

Tip

Be careful not to set the Viewport subdivisions higher than the Render subdivisions, this would mean that in the 3D Viewport the quality will be higher than the rendered.

Options

UV Smooth - How to handle UVs during subdivision.

- Smooth, keep corners - UV islands are smoothed, but their boundary remain sharp.
- Sharp - UV remain unchanged.

Optimal Display - When rendering the wireframe of this object, the wires of the new subdivided edges will be skipped (only displays the edges of the original geometry).

Use Creases - Use the Weighted Edge Creases values stored in edges to control how smooth they are made.

Keyboard Shortcuts

To quickly add a Subdivision Surface modifier to one or more objects, select the object(s) and press Ctrl-1. That will add a Subdivision Surface modifier with Viewport subdivisions set to 1. You can use other numbers too, such as Ctrl-2, Ctrl-3, etc, to add a modifier with that number of subdivisions. Adding a Subdivision Surface modifier in this fashion will not modify the Render subdivisions.

If an object already has a Subdivision Surface modifier, doing this will simply change its subdivision level instead of adding another modifier.

Control - Catmull-Clark subdivision rounds off edges, and often this is not what you want. There are several solutions that allow you to control the subdivision.

Weighted Edge Creases

Weighted edge creases for subdivision surfaces allows you to change the way the *Subdivision Surface* modifier subdivides the geometry to give the edges a smooth or sharp appearance.

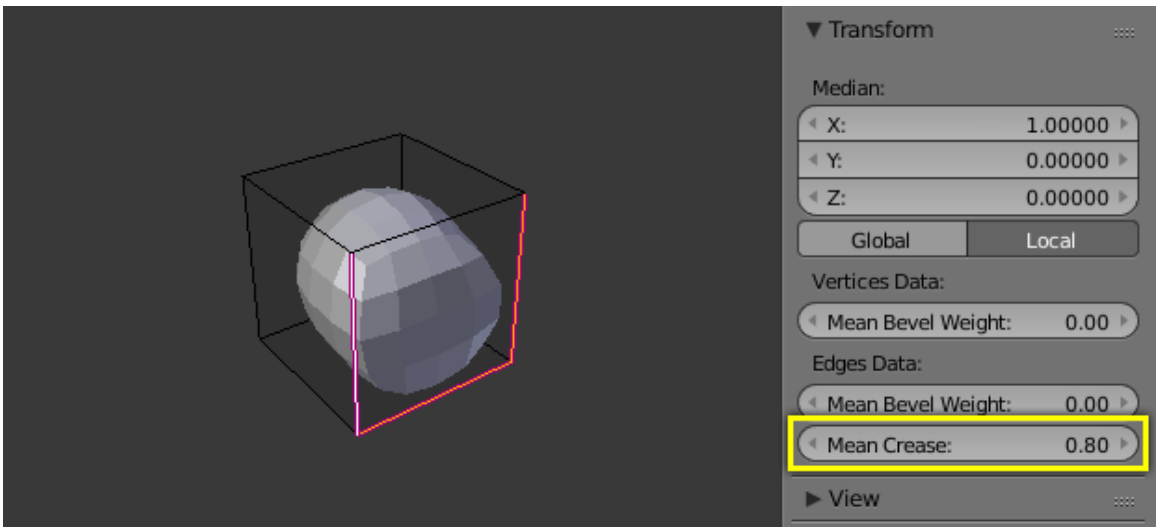


Figure 32 A subdivided cube with creased edges.

The crease weight of selected edges can be changed in the Transform panel, Sidebar of the 3D Viewport. The scale-like dedicated tool Shift-E can also be used to adjust the crease weight. A higher value makes the edge "stronger" and more resistant to the smoothing effect of subdivision surfaces.

Edge Loops

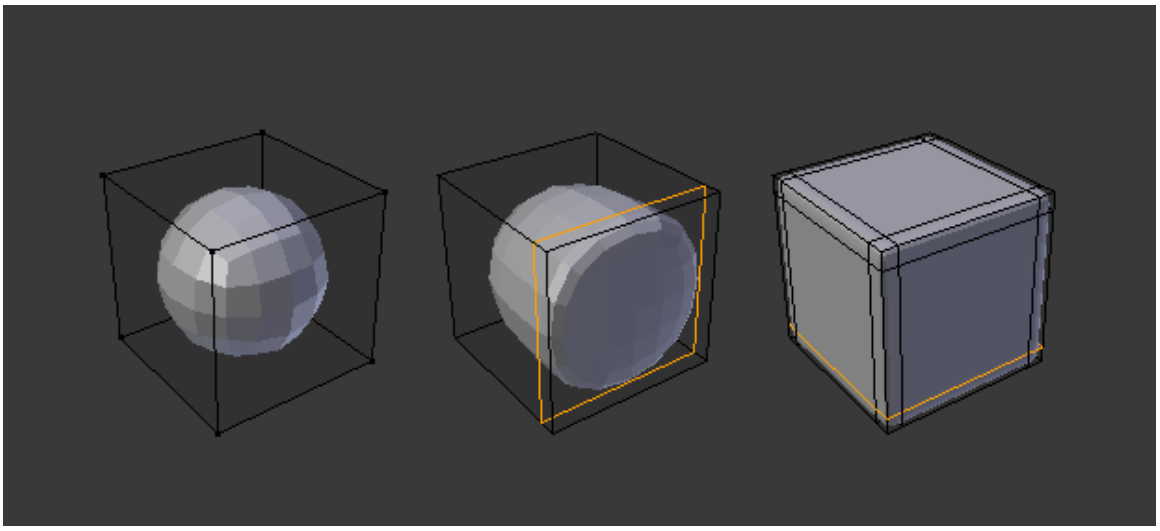


Figure 33 Subdivision Level 2 cube, the same with an extra Edge Loop, and the same with six extra Edge Loops.

The Subdivision Surface modifier demonstrates why good, clean topology is so important. As you can see in the figure, the it has a drastic effect on a default cube. Until you add in additional loops (with e.g. Loop Cut and Slide), the shape is almost unrecognizable as a cube.

A mesh with deliberate topology has good placement of edge loops, which allow the placement of more loops (or their removal) to control the sharpness/smoothness of the resultant mesh.

Known Limitations

Non-Contiguous Normals - Blender's subdivision system produces nice smooth subdivided meshes, but any subdivided face (that is, any small face created by the algorithm from a single face of the original mesh), shares the overall normal orientation of that original face.

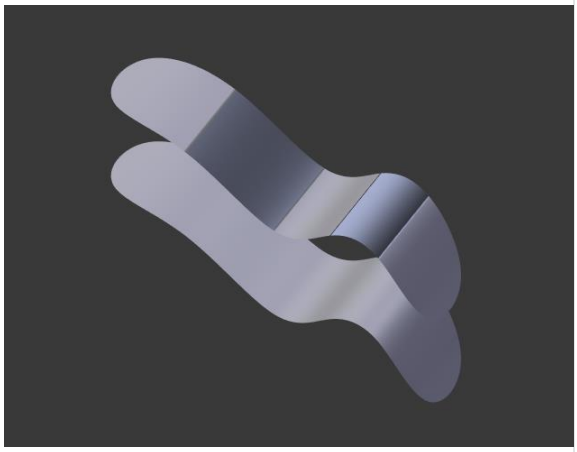


Figure 34 Comparison of good normals and bad normals.

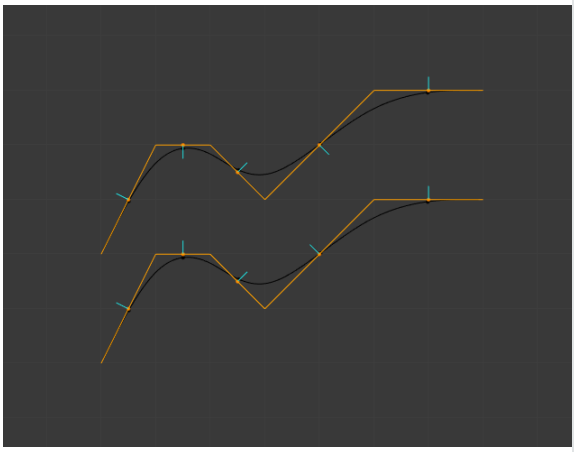


Figure 35 Side view of image on the left.

Abrupt normal changes can produce ugly black gouges even though these flipped normals are not an issue for the shape itself.

A quick way to fix this is to use Blender's Recalculate Normals operation in Edit Mode.

If you still have some ugly black gouges you will have to manually flip the normals.

Wireframe Modifier

The Wireframe modifier transforms a mesh into a wireframe by iterating over its faces, collecting all edges and turning those edges into four-sided polygons. Be aware of the fact that your mesh needs to have faces to be wireframed. You can define the thickness, the material and several other parameters of the generated wireframe dynamically via the given modifier options.

Thickness - The depth or size of the wireframes.

Offset - A value between (-1 to 1) to change whether the wireframes are generated inside or outside of the original mesh. Set to zero, Offset will center the wireframes around the original edges.

Vertex Group - Restrict the modifier to only this vertex group.

- Invert - Inverts the vertex group weights.

- Factor - Percentage that the vertex has influence over the final wireframe result.

Crease Edges - This option is intended for usage with the Subdivision modifier. Enable this option to crease edges on their junctions and prevent large curved intersections.

- Crease Weight - Define how much crease (0 to 1, nothing to full) the junctions should receive.

Options

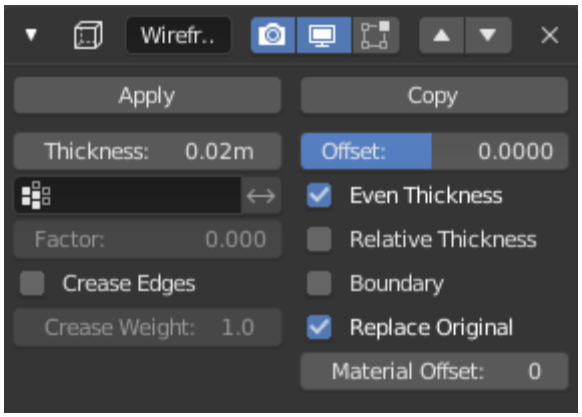


Figure 36 The Wireframe modifier.

Even Thickness - Maintain thickness by adjusting for sharp corners. Sometimes improves quality but also increases computation time.

Relative Thickness - Determines the edge thickness by the length of the edge. Longer edges will be thicker.

Boundary - Creates wireframes on mesh island boundaries.

Replace Original - If this option is enabled, the original mesh is replaced by the generated wireframe. If not, the wireframe is generated on top of it.

Material Offset - Uses the chosen material index as the material for the wireframe; this is applied as an offset from the first material.

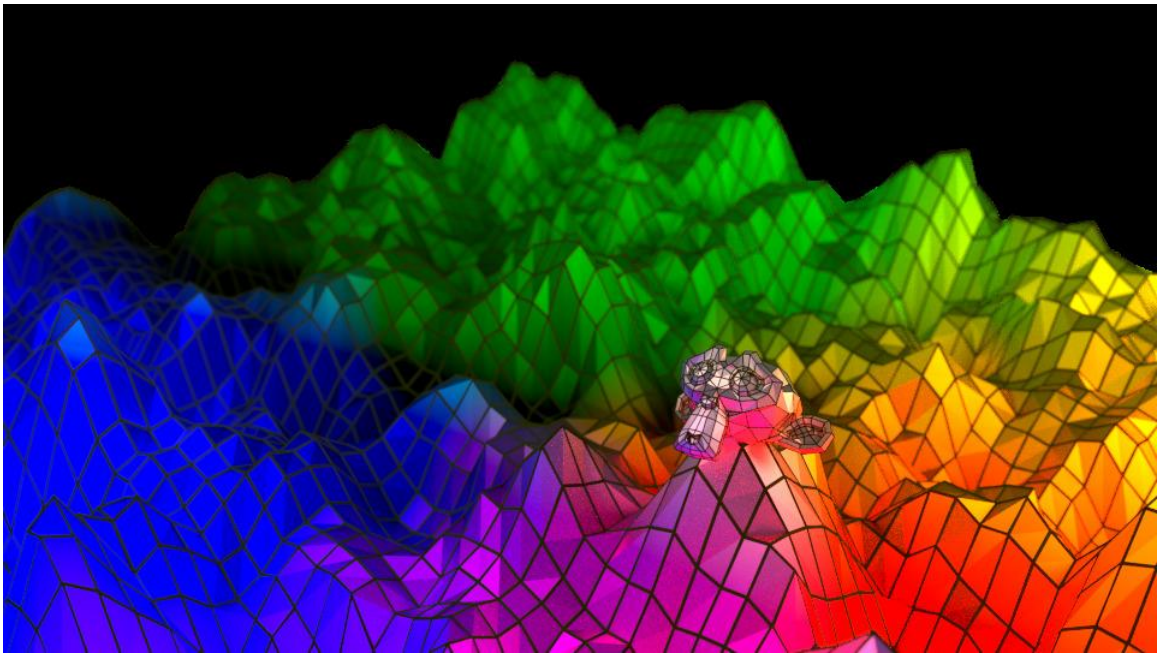


Figure 37 Wireframes on a displaced plane.

In this example, the wireframes carry a second (dark) material while the displaced plane uses its original one.

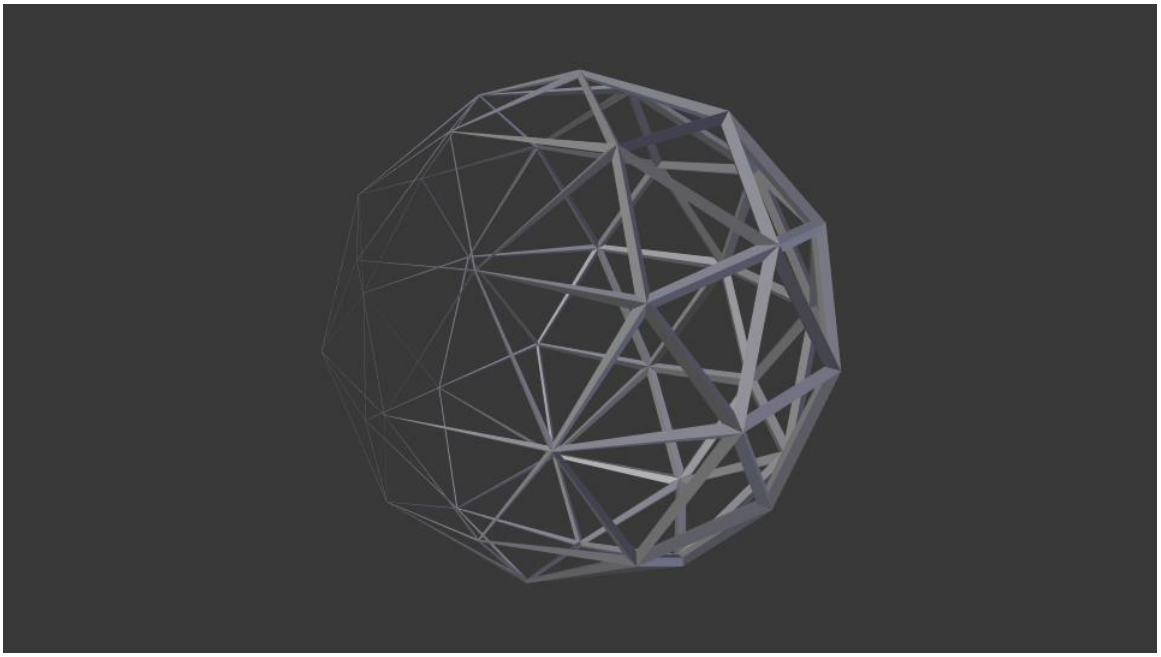


Figure 38 Vertex group weighting.

The weights of the vertex group gradually change from 0 to 1.

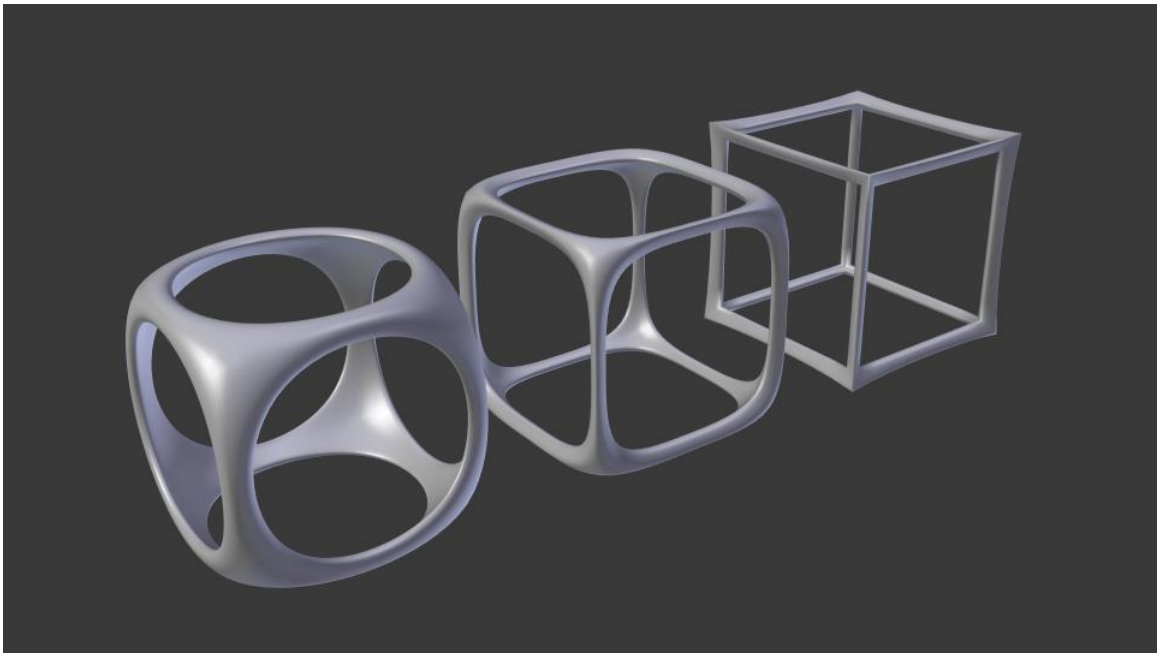


Figure 39 Wireframe and Subdivision Surface modifier.

Cube with enabled Crease Edges option. The Crease Weight is set to 0, 0.5 and 1.

Build Modifier

The Build modifier causes the faces of the mesh object to appear or disappear one after the other over time.

By default, faces appear in the order in which they are stored in memory (by default, the order of creation). The face/vertex order can be altered in Edit Mode by using Sort Mesh Elements.

Options

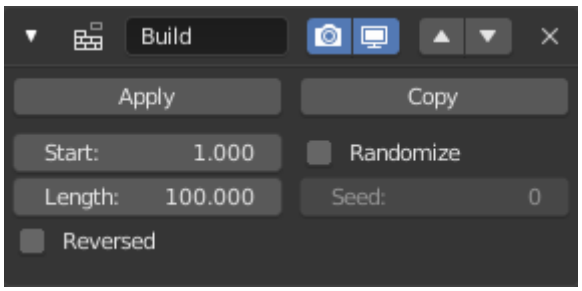


Figure 40 The Build modifier.

- Start - The start frame of the building process.
- Length - The number of frames over which to rebuild the object.
- Randomize - Randomizes the order in which the faces are built.
- Seed - The random seed. Changing this value gives a different "random" order when randomize is checked. This order is always the same for a given seed/mesh set.
- Reversed - The modifier will operate in reverse, essentially allowing it to be used as a "deconstruction" effect. This is useful for making a set of instancing objects gradually disappear.

Example

The Build modifier is often useful when needing a way to get a large number of items to progressively appear, without resorting to animating the visibility of each one by one. Examples of this include a mesh containing vertices only, which is used as an Instance Verts emitter, and has the Build modifier on it. Such a setup is a workaround/technique for being able to art-direct some semi-random layout of a collection of objects (i.e., leaves/balls forming a carpet of sorts) when doing so with particles is not desirable (e.g., due to undesirable distribution of items leaving random gaps and overlapping in other places).

Edge Split Modifier

The Edge Split modifier splits, duplicates edges within a mesh, breaking 'links' between faces around those split edges.

The edges to split can be determined from the edge angle (i.e., angle between faces forming that edge), and/or edges marked as sharp.

Splitting an edge affects vertex normal generation at that edge, making the edge appear sharp. Hence, this modifier can be used to achieve the same effect as Auto Smooth, making edges appear sharp when their angle is above a certain threshold. It can also be used for manual control of the smoothing process, where the user defines which edges should appear smooth or sharp (see Mesh Smoothing for other ways to do this). If desired, both modes can be active at once.

Note

This modifier is kept mostly for historical/compatibility reasons. Everything it can do in shading, and much more, can now be achieved using custom normals.

Unless you really need the topology changes it generates, it is not advised to use it in new projects.

Note

Splitting edges can also be performed manually in Edit Mode.

Options

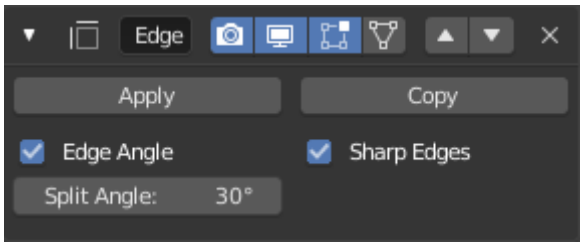


Figure 41 The Edge Split modifier.

Edge Angle - When enabled, edges will be split if the angle between its two adjacent faces is greater than the Split Angle.

- Split Angle - On 0: all edges are split. On 180: no edges are split.

Sharp Edges - When enabled, edges will be split if they were marked as sharp.

Note

Non-manifold edges will always be split.

Examples

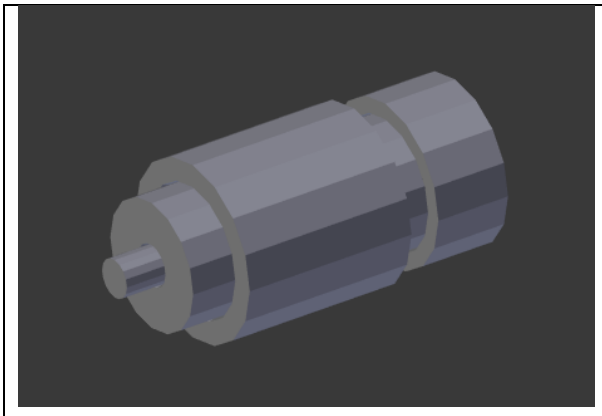


Figure 42 Flat shading.

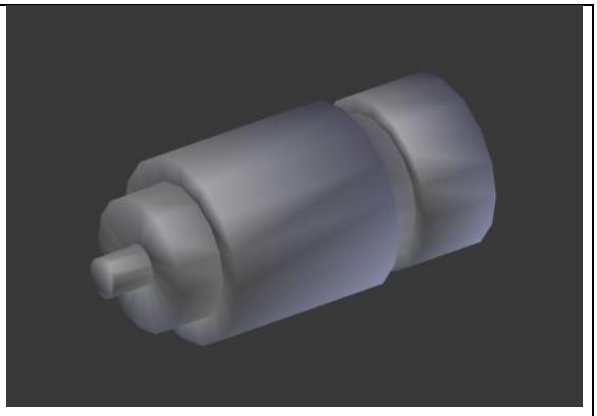


Figure 43 Smooth shading.

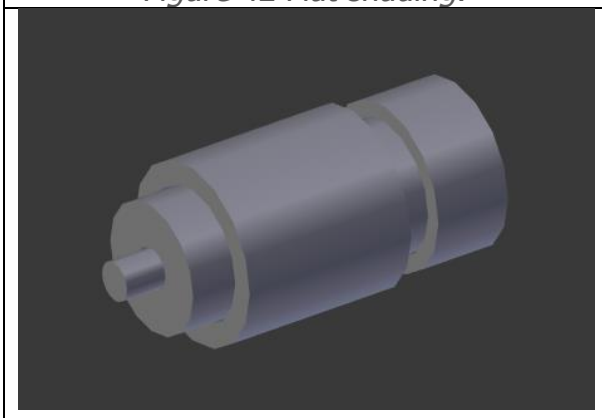


Figure 44 Smooth shading with Edge Split.

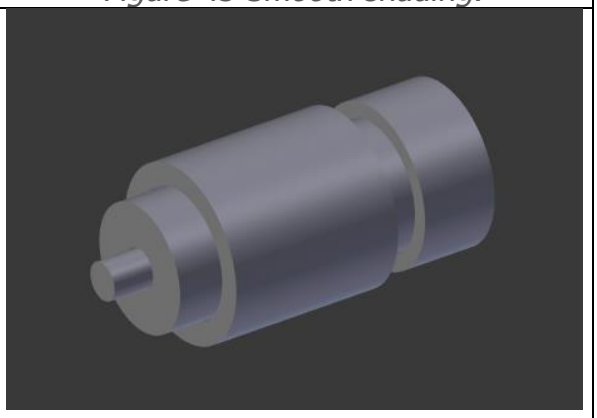


Figure 45 Smooth shading with Edge Split and Subdivision Surface.

Mask Modifier

The Mask modifier allows vertices of an object to be hidden dynamically based on vertex groups.

Options

Mode - The Mask Modifier can hide parts of a mesh based on two different modes, selectable from this select menu.

Vertex Group - When the Vertex Group option is selected, all vertices belonging to the chosen vertex group (with a weight above zero) will be visible, and all other vertices will be hidden.

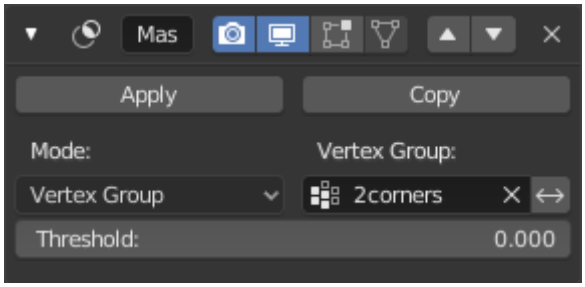


Figure 46 The Mask modifier in Vertex Group mode.

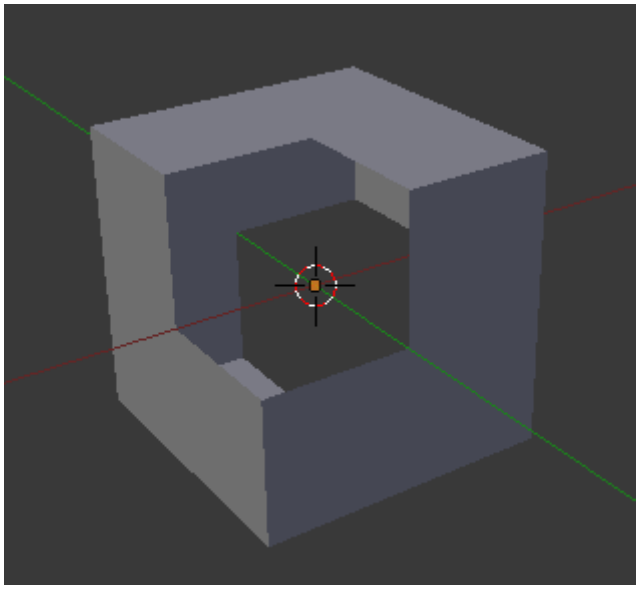


Figure 47 The Mask modifier in Vertex Group mode.

Armature - When in Pose Mode, vertices belonging to the vertex group associated with the active bone (same names) will be visible. Vertices not in that group will be hidden.

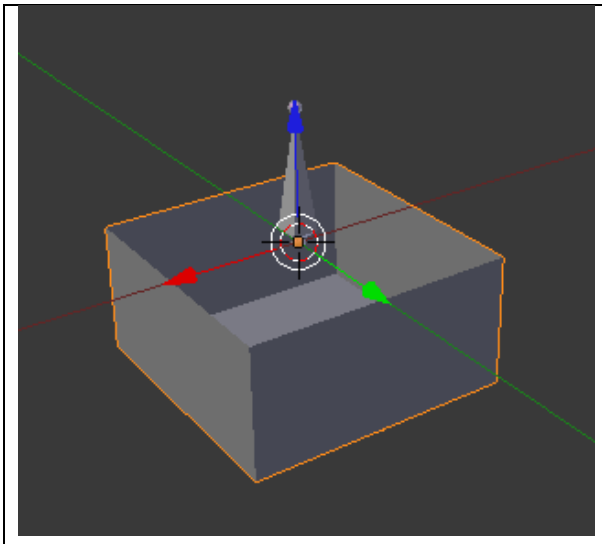


Figure 48 The Mask modifier in Armature mode.

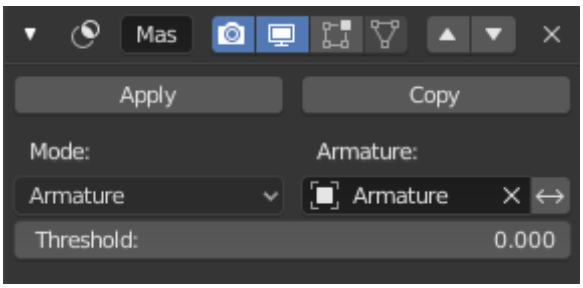


Figure 49 The Mask modifier in Armature mode.

Inverse - Normally, vertices belonging to the selected vertex group (or group associated with the active pose bone) will be shown. The Invert toggle allows you to reverse this behavior, instead only showing vertices which do not belong to the vertex group.

Threshold - Vertices with weights less or equal to this value will be hidden

Multiresolution Modifier

The Multiresolution modifier (often shortened to "Multires") gives you the ability to subdivide a mesh similarly to the Subdivision Surface modifier, but also allows you to edit the new subdivision levels in Sculpt Mode.

Note

Multiresolution is the only modifier that cannot be repositioned in the stack after any modifier that will change geometry or other object data (i.e.,all Generate, some Modify and some Simulate modifiers cannot come before the Multiresolution one).

Options

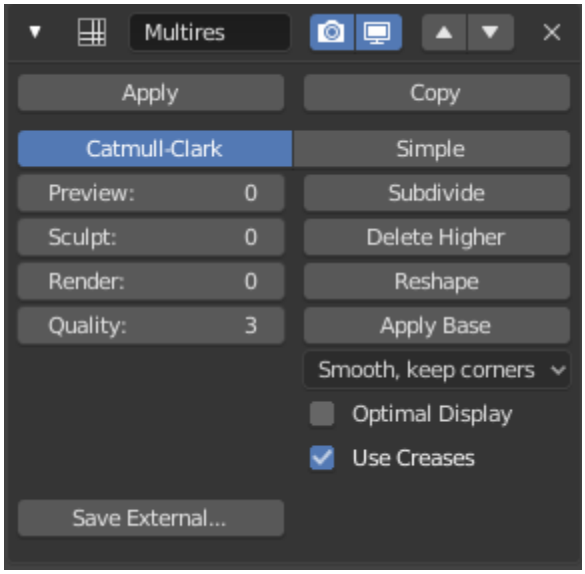


Figure 50 The Multiresolution modifier.

Type - Sets the type of subdivision.

- Simple - Maintains the current shape, and simply subdivides edges.
- Catmull-Clark - Creates a smooth surface, usually smaller than the original, using the standard Catmull-Clark subdivision surface algorithm.

Levels - The left column of settings control the general quality generated by the modifier in various contexts.

- Preview - Set the level of subdivisions to show in Object Mode.
- Sculpt - Set the level of subdivisions to use in Sculpt Mode.
- Render - Set the level of subdivisions to show when rendering.
- Quality - How precisely the vertices are positioned (relatively to their theoretical position), can be lowered to get a better performance when working on heavy meshes.

Operations

The right column gathers several operators necessary to manage multiresolution workflow.

Subdivide - Adds another level of subdivision.

Delete Higher - Deletes all subdivision levels that are higher than the current one.

Reshape - Copies vertex coordinates from another mesh.

To use it, first select a different mesh object with matching topology and vertex indices, then shift select the object you wish to copy vertex coordinates to, and click Reshape.

Apply Base - Modifies the original unsubdivided mesh to match the form of the subdivided mesh.

Further Options

UV Smooth - How to handle UVs during subdivision.

- Smooth, keep corners - UV islands are smoothed, but their boundary remain sharp.
- Sharp - UV remain unchanged.

Subdivide UVs - When enabled, the UV maps will also be subdivided. (i.e., Blender will add "virtual" coordinates for all sub-faces created by this modifier.)

Optimal Display - When rendering the wireframe of this object, the wires of the new subdivided edges will be skipped (only displays the edges of the original geometry).

Save External - Saves displacements to an external .btx file.

Remesh Modifier

The Remesh modifier is a tool for generating new mesh topology. The output follows the surface curvature of the input, but its topology contains only quads.

Options

Mode - There are three basic modes available in the Remesh modifier. The output topology is almost identical between the three modes, what changes is the smoothing.

Blocks - There is no smoothing at all.

Smooth - Output a smooth surface.

Sharp - Similar to Smooth, but preserves sharp edges and corners.

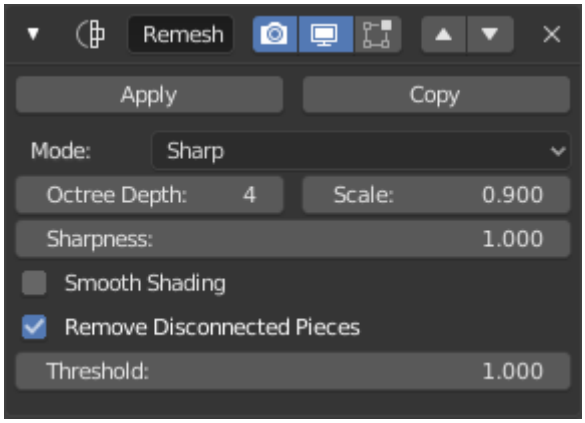


Figure 51 The Remesh modifier.

- Sharpness - Higher values produce edges more similar to the input, while lower values filter out noise.

Voxel - Uses an OpenVDB to generate a new manifold mesh from the current geometry while trying to preserve the mesh's original volume.

- Adaptivity - Reduces the final face count by simplifying geometry where detail is not needed. This introduces triangulation to faces that do not need as much detail.
- Smooth Shading - Outputs faces with Smooth Shading instead of flat shading.

Octree Depth - Sets the resolution of the output. Low values will generate larger faces relative to the input, higher values will generate a denser output.

Scale - The result can be tweaked further by this; lower values effectively decrease the output resolution.

Smooth Shading - Output faces with smooth shading rather than flat shading. The smooth/flat shading of the input faces is not preserved.

Remove Disconnected Pieces - Filter out small disconnected pieces of the output.

Thin parts of the input mesh can become lose, and generate small isolated bits of mesh. This option will remove those.

- Threshold - Use this to control how small a disconnected component must be to be removed.

Note

The input mesh should have some thickness to it. If the input is completely flat, add a Solidify Modifier above the Remesh one.

Examples

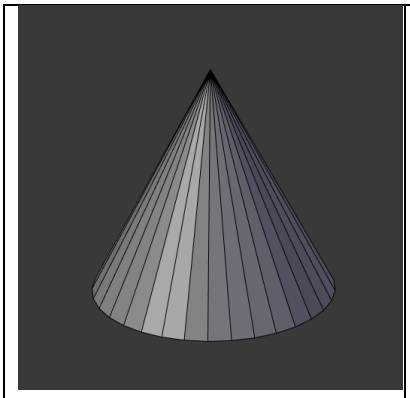


Figure 52 Unmodified mesh.

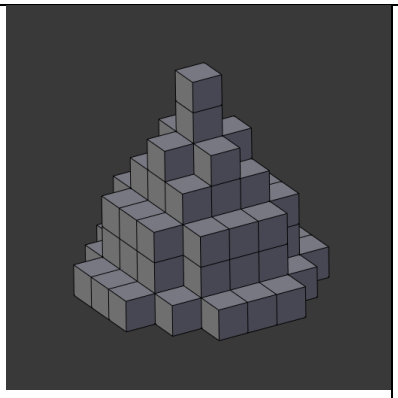


Figure 53 Blocks mode with Octree Depth 3.

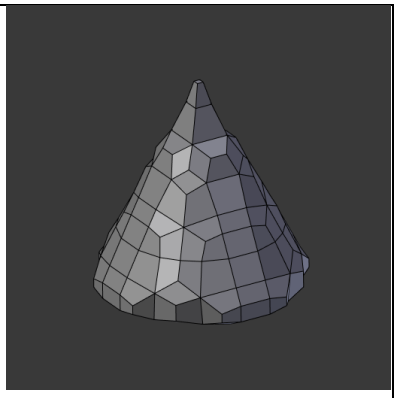


Figure 54 Smooth mode with Octree Depth 3.

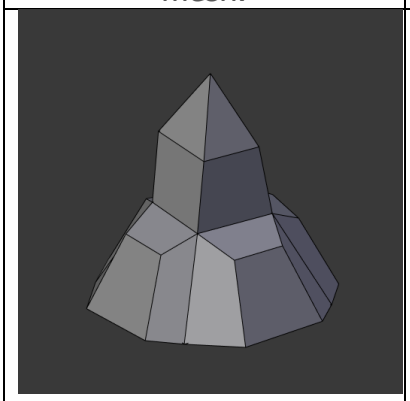


Figure 55 Sharp mode with Octree Depth 2.

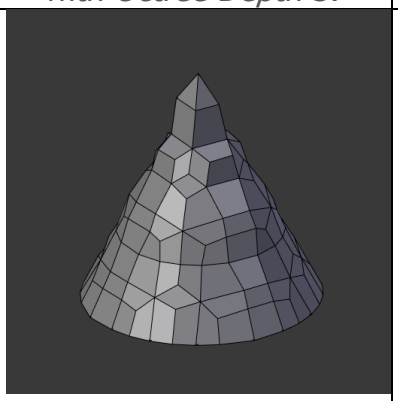


Figure 56 Sharp mode with Octree Depth 3.

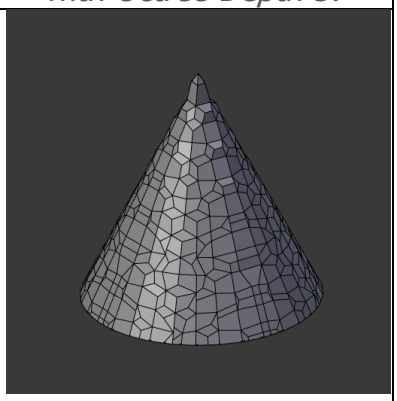


Figure 57 Sharp mode with Octree Depth 4.

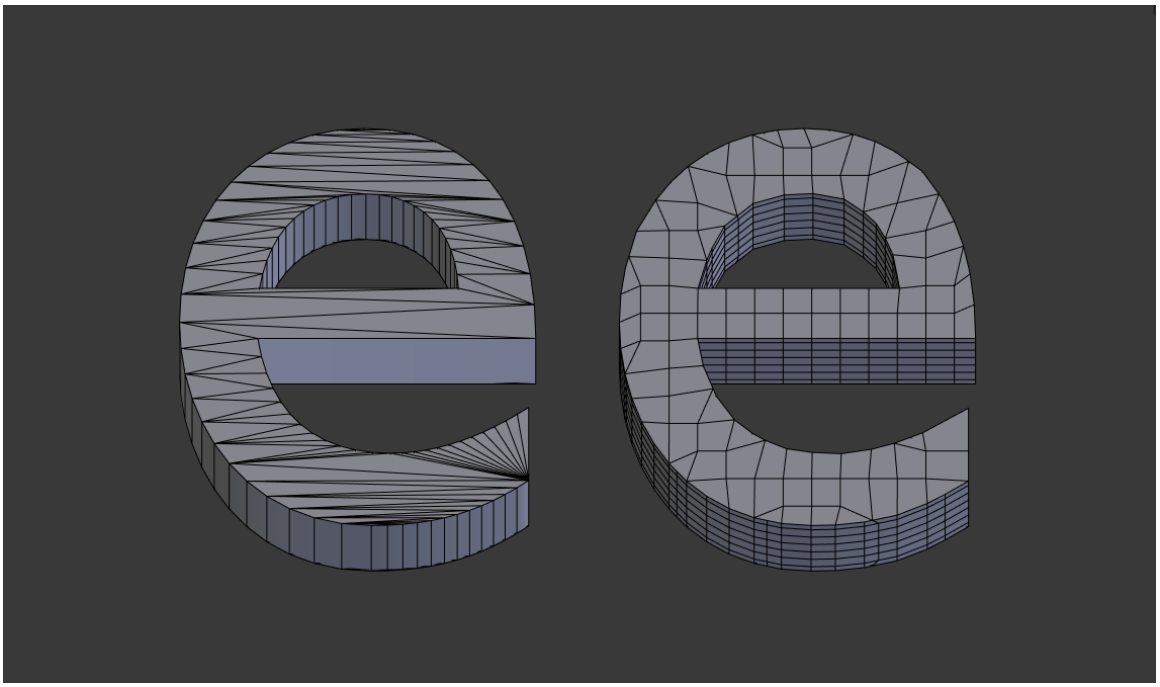


Figure 58 The Remesh Modifier applied to a text to improve its topology.

Skin Modifier

The Skin modifier uses vertices and edges to create a skinned surface, using a per-vertex radius to better define the shape. The output is mostly quads, although some triangles will appear around intersections.

It is a quick way to generate base meshes for sculpting and/or smooth organic shapes with arbitrary topology.

Note

Faces in the original geometry are ignored.

Options

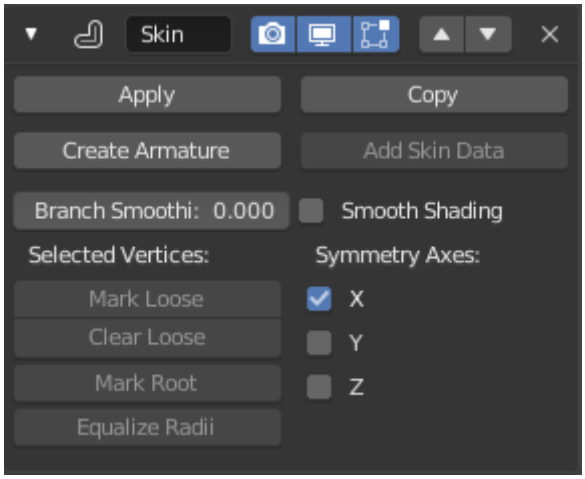


Figure 59 The Skin modifier.

Branch Smoothing - A branch point is a vertex with three or more connected edges. These areas tend to produce more complicated topology, some of which may overlap. This setting relaxes the surface around these points, with the side effect of shrinking it.

Smooth Shading - Output faces with smooth shading rather than flat shading. The smooth/flat shading of the input geometry is not preserved.

Symmetry Axes X/Y/Z - These checkboxes are used to keep the output topology symmetrical in their respective axes. In other words, using it avoids merging triangles across an axis unless the triangles form a symmetric quad.

Note

They do not add geometry flipped across an axis. For that, the Mirror modifier should be used, typically placed above the Skin one.

Add Skin Data

This modifier uses a custom set of data in the mesh, that is generated automatically when you add the modifier the first time.

However, you may remove that data, or loose it some way or the other. That operator will generate it again.

Selected Vertices

Those operators modify the original mesh 'control data' for the Skin modifier. They help manage its behavior.

Mark/Clear Loose - By default, a branch vertex (vertex with three or more connected edges) will generate extra edge loops along adjacent edges in order to keep the output tight. Branches can be made loose by clicking Mark Loose, which will allow the output to stretch between all adjacent vertices. This can be disabled again by clicking Clear Loose.

Mark Root - Marking a vertex as root causes that vertex to be used for calculating rotations for connected limbs. Root vertices also affect the armature output, they will be used as the origin for the root bones.

Each set of connected vertices should have one root node (one is selected by default if you do not assign any manually). Mark Root enforces the one-root per set rule, so it is not necessary to manually unmark roots.

Equalize Radii - Makes the skin radii of selected vertices equal on each axis.

Create Armature - Create an armature on top of the object. Each edge becomes a bone.

Note

If the root vertex has more than one adjacent edge, an extra bone will be created to serve as the root.

This tool does the following: -

1. A new armature object is added with bones matching the input mesh. The active selection is switched to the new armature.
2. Weight groups are added to the input mesh. The Skin modifier propagates these weights to the output as well.
3. An Armature modifier is added directly below the Skin one. Note that the Armature modifier is being applied after the Skin one because it should only deform the output, whereas if it were above, it might change the resulting topology.

Skin Mesh Data

That modifier needs a set of specific data in the original mesh to work properly. This data allows to define the root vertices of each tree, which ones are loose (see Selected Vertices above), and the size (radius) of the skin at each vertex. The radii of input vertices can be individually scaled in Edit Mode with the Skin Resize.

Examples

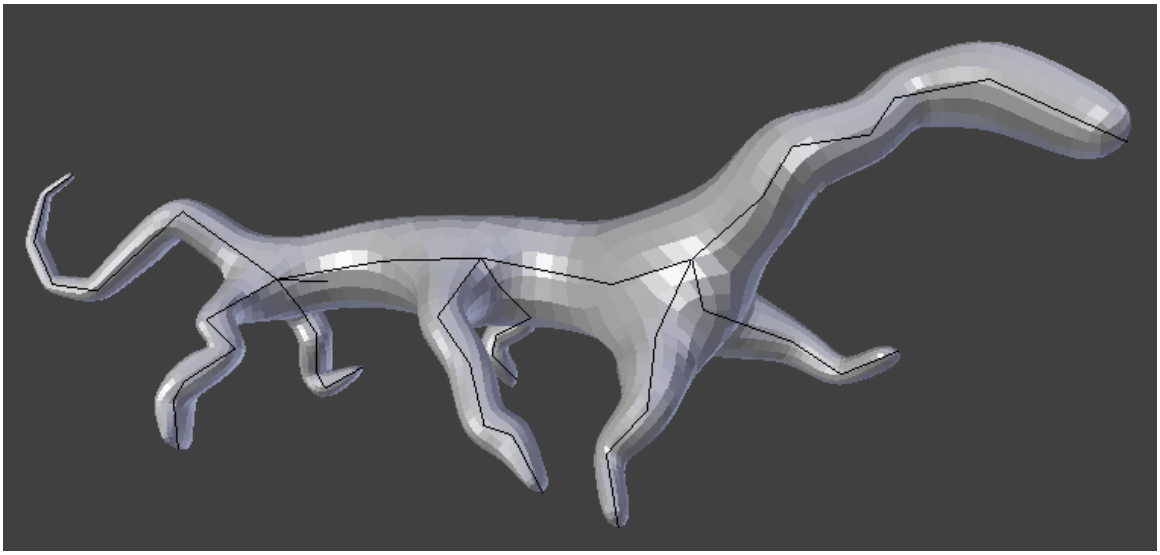


Figure 60 Simple creature, made with only the Skin and Subdivision Surface modifiers.

Triangulate Modifier

The Triangulate modifier converts all faces in a mesh (quads and polygons) to triangular faces. It fulfils the exact same function as the Triangulate tool in Edit Mode.

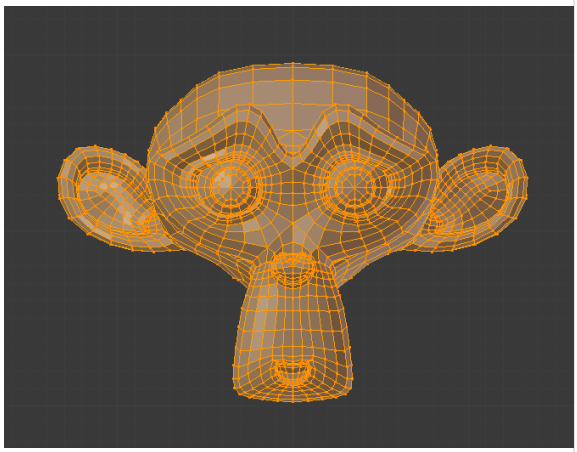


Figure 61 Mesh before Triangulate modifier.

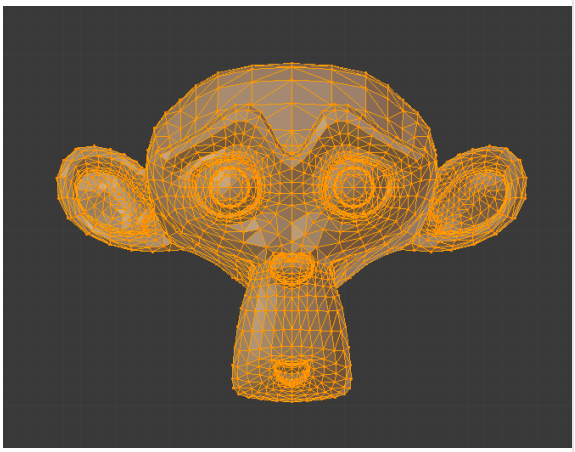


Figure 62 Mesh after Triangulate modifier.

Options

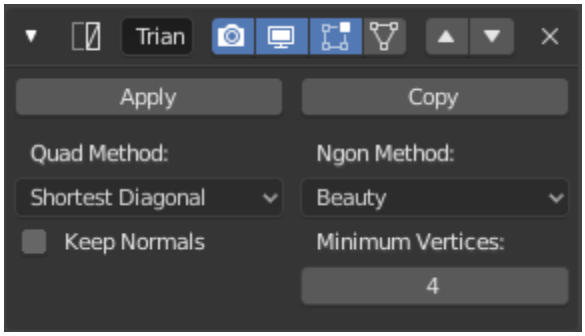


Figure 63 The Triangulate modifier.

Quad Method

Beauty - Split the quads in nice triangles, slower method.

Fixed - Split the quads on their 1st and 3rd vertices.

Fixed Alternate - Split the quads on their 2nd and 4th vertices.

Shortest Diagonal - Split the quads based on the diagonal distance between their vertices.

N-gon Method

Beauty - Arrange the new triangles nicely, slower method.

Clip - Split the polygons using an ear-clipping algorithm (gives similar results to the tessellation used for the viewport rendering).

- Keep Normals - When using custom normals, try to preserve the same shading as before triangulation.
- Minimum Vertices - Minimum number of vertices a face must have to be triangulated. For example, setting this value to 5, will prevent triangulation of quads and only triangulate N-gons.

Weld Modifier

The Weld modifier looks for groups of vertices within a threshold and merges them, collapsing the surrounding geometry.

Options

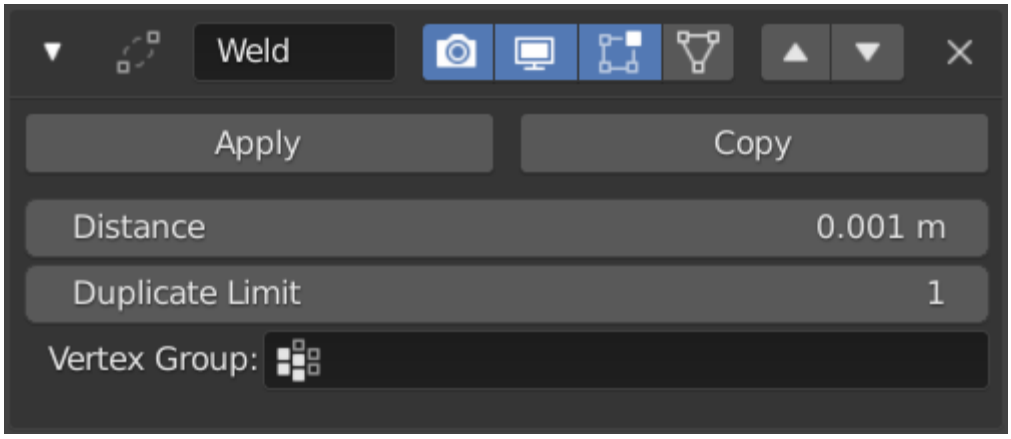


Figure 64 The Weld modifier.

Distance - Maximum distance that the vertices must have each other to be merged.

Duplicate Limit - For a better performance, limits the number of elements found per vertex. 0 makes it infinite.

Vertex Group - When the Vertex Group option is selected, only vertices with weight above zero will be affected by the modifier.

Invert - Inverts the influence of the selected vertex group, meaning that the group now represents vertices that will not be deformed by the modifier.

The setting reverses the weight values of the group.