Skip to content

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 shap

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

Options

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

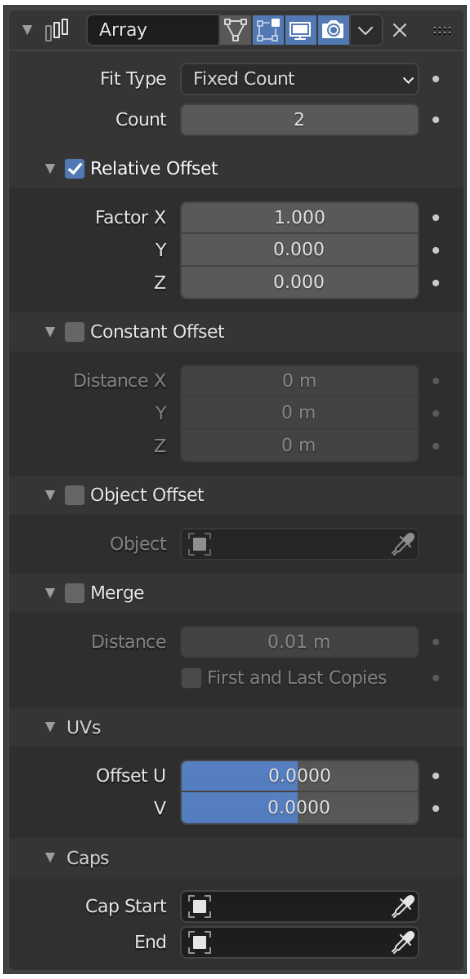
Generates enough copies to fit within the length of the curve object specified in *Curve*.

Fit Length

Generates enough copies to fit within the fixed length given by *Length*.

Fixed Count

Generates the number of copies specified in *Count*.



Array

Fit Type: Fixed Count

Count: 2

▼ ☑ Relative Offset

Factor X: 1.000

Y: 0.000

Z: 0.000

▼ □ Constant Offset

Distance X: 0 m

Y: 0 m

Z: 0 m

▼ □ Object Offset

Object:

▼ □ Merge

Distance: 0.01 m

□ First and Last Copies

▼ UVs

Offset U: 0.0000

V: 0.0000

▼ Caps

Cap Start:

End:

The Array modifier.

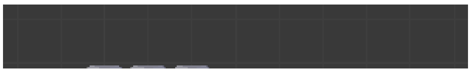
Note

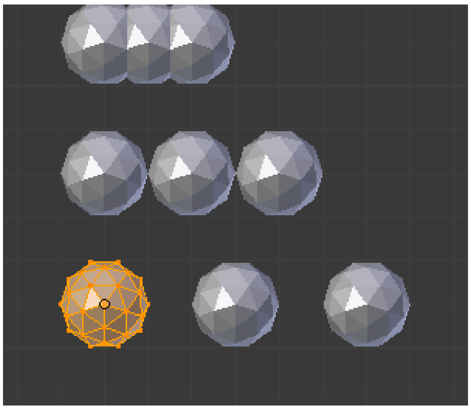
- Both *Fit Curve* and *Fit Length* use the local coordinate system size of the base object, which means that scaling the base object in Object Mode will not change the number of copies generated by the modifier.
- *Fit Curve* uses the local coordinate system length of the curve, which means that scaling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

Relative Offset

Factor 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.





Relative offset (0.5, 1.0 and 1.5) examples.

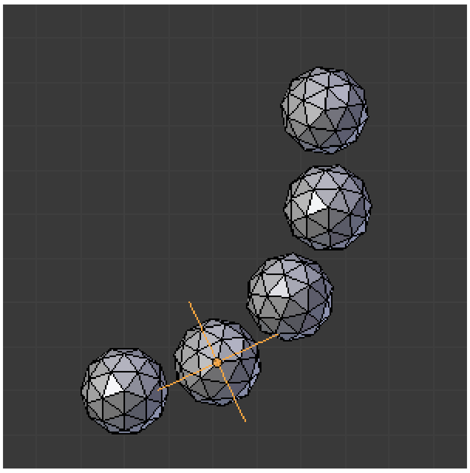
Constant Offset

Distance X/Y/Z

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

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 th initial object. E.g. by rotating this empty a circle or helix of objects can be created.



Object offset example.

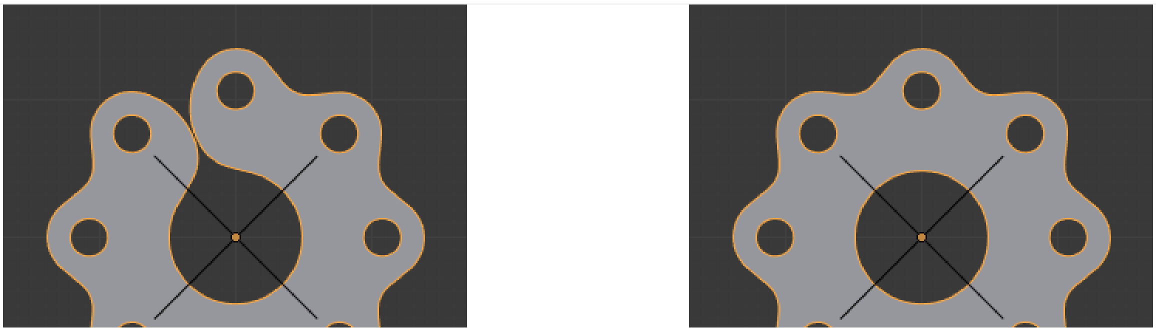
Merge

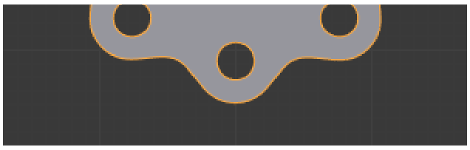
If enabled, vertices in each copy will be merged with vertices in the next copy that are within the given *Distance*.

First and Last Copies

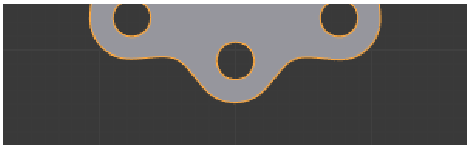
If enabled **and** *Merge* is enabled, vertices in the first copy will be merged with vertices in the last copy, again if they are within *Distance* range. Th is useful for circular objects.

First and Last Copies merge example.





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



Subdivision discontinuity eliminated by merging vertices between firs and last copies (*First and Last Copies* on).

Distance

Controls the merge distance for *Merge* and *First and Last Copies*.

UVs

Offset U/V

Shifts UVs of each new duplicate by a settable amount.

Caps

Cap Start, End

This allows either endpoints 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. on "array step" after the last "regular" array copy.

When *Merge* is activated, the *cap* vertices within the *Distance* threshold will be merged.

Note

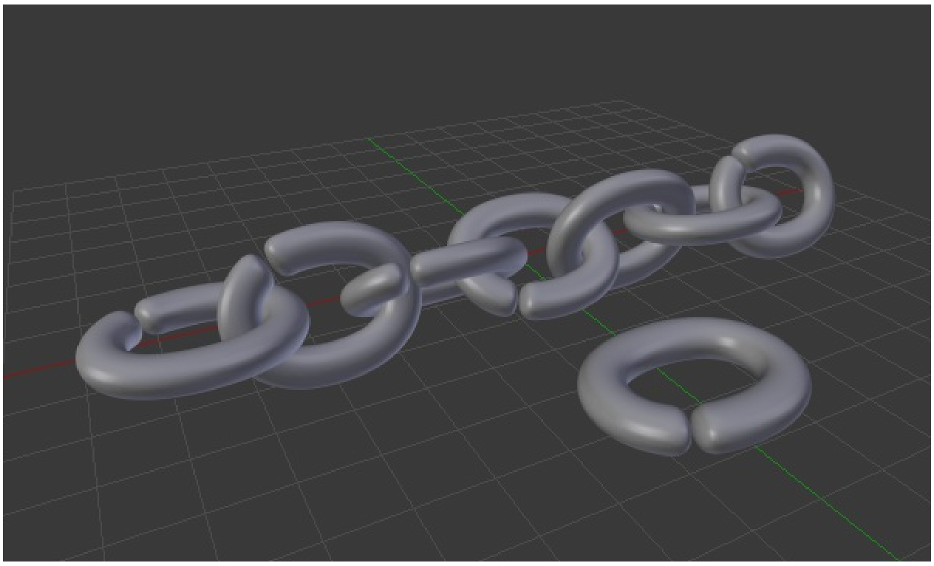
The start/end cap objects currently do not support the *First and Last Copies* option.

Hints

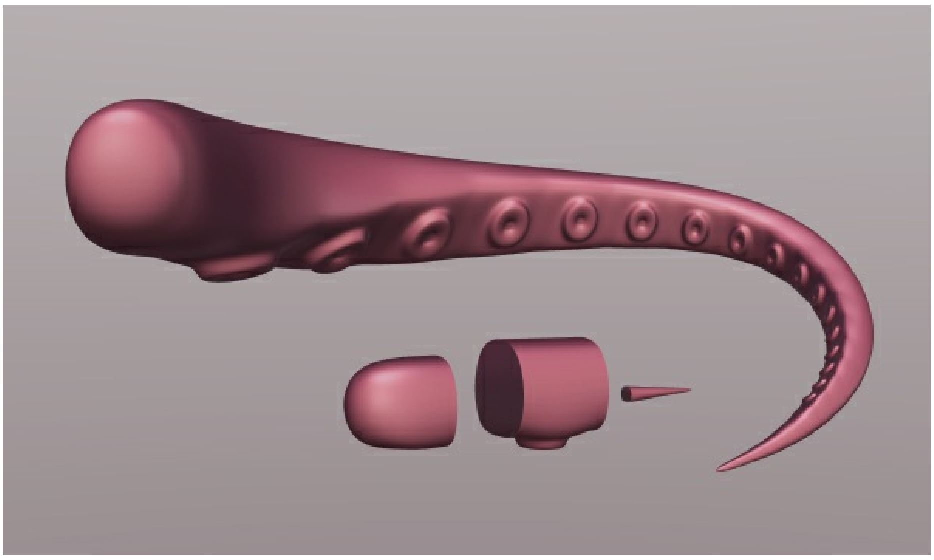
Offset Calculation

The transformation applied from one copy to the next is calculated as the sum of the three different components (*Relative*, *Constant* and *Object*), each 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



A chain created from a single link. Sample blend-file.

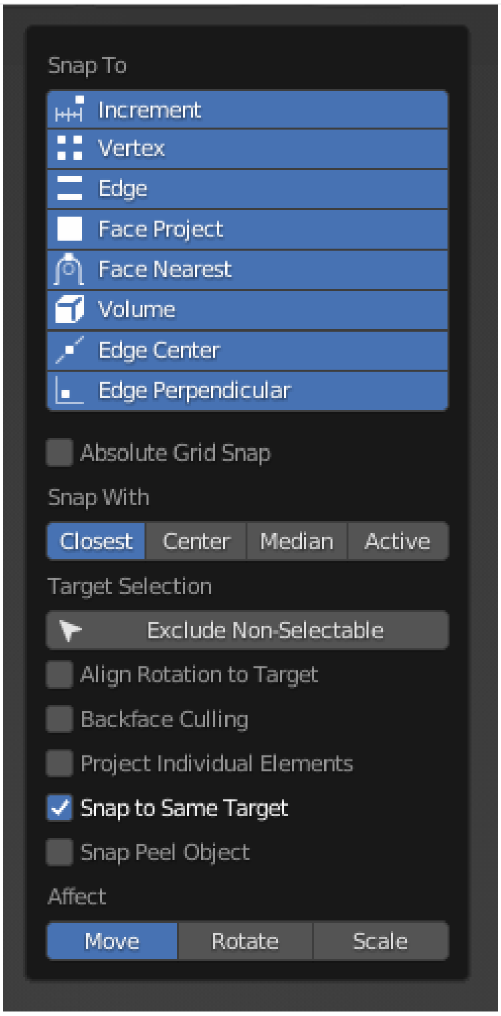


A tentacle created with an Array Modifier followed by a Curve Modifier.

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

Fractal

Multi-level array animated with motion blur.



Snap To

Increment
Vertex
Edge
Face Project
Face Nearest
Volume
Edge Center
Edge Perpendicular

□ Absolute Grid Snap

Snap With

Closest | Center | Median | Active

Target Selection

Exclude Non-Selectable

□ Align Rotation to Target
□ Backface Culling
□ Project Individual Elements
☑ Snap to Same Target
□ Snap Peel Object

Affect

Move | Rotate | Scale

Fractal created with multiple arrays. Sample blend-file.

Last updated on 2025-05-10

View Source
View Translation
Report issue on this page