Menu *Weights → Locks*

Hotkey K

Vertex groups can be locked to prevent undesired edits to a particular vertex group.

Tip: Bones that belong to a locked vertex group are displayed in red the 3D Viewport.

Lock All Locks all vertex groups.

Unlock All Unlocks all vertex groups.

Lock Selected Locks selected vertex groups.

Unlock Selected Unlocks selected vertex groups.

Lock Unselected Locks unselected vertex groups.

Unlock Unselected Unlocks Unselected vertex groups.

Lock Only Selected Lock selected and unlock selected vertex groups.

Lock Only Unselected Unlock selected and lock unselected vertex groups.

Invert Locks Inverts the locks on all vertex groups.

2.7 Grease Pencil

2.7.1 Introduction

Grease Pencil is a particular type of Blender object that allow you to draw in the 3D space. Can be use to make traditional 2D animation, cut-out animation, motion graphics or use it as storyboard tool among other things.

The object act as a container of strokes that you can create using drawing tools in *Draw Mode*. More detailed editing of the strokes is done in *Edit Mode*, and *Sculpt Mode*.

Tip: To obtain best results with Grease Pencil is highly recommended to use a *Graphics Tablet*.

2.7.2 Object

Visibility

Reference

Panel *Object Properties → Visibility*

See also:

There are several other *general visibility* properties.

Use Light

Enables the Grease Pencil object to be affected by lights.

This property affect the whole object, for more control with lights you can enable or disable the use of lights by layers. See *Layers* for more information.

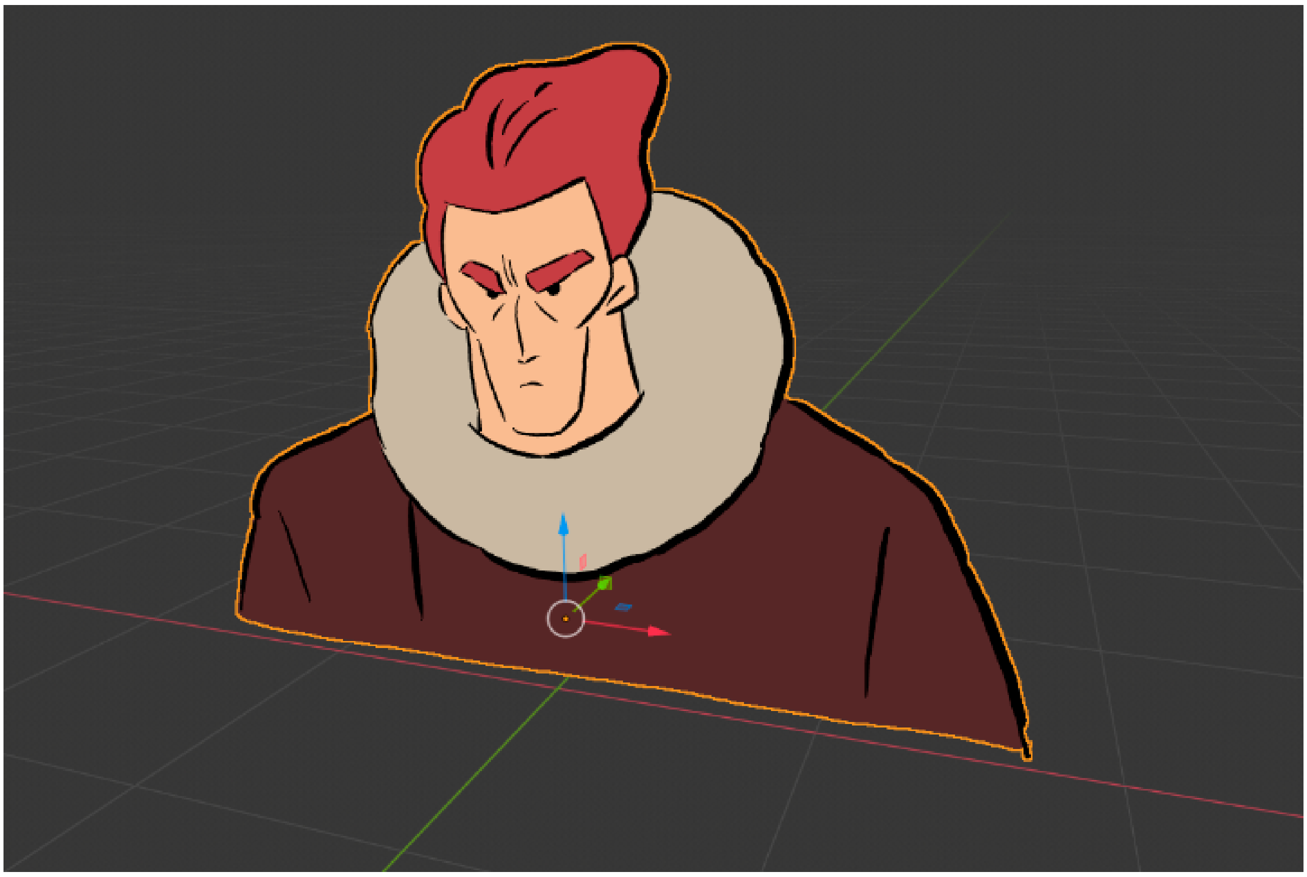


Fig. 1360: An example of a Grease Pencil object in the 3D environment.

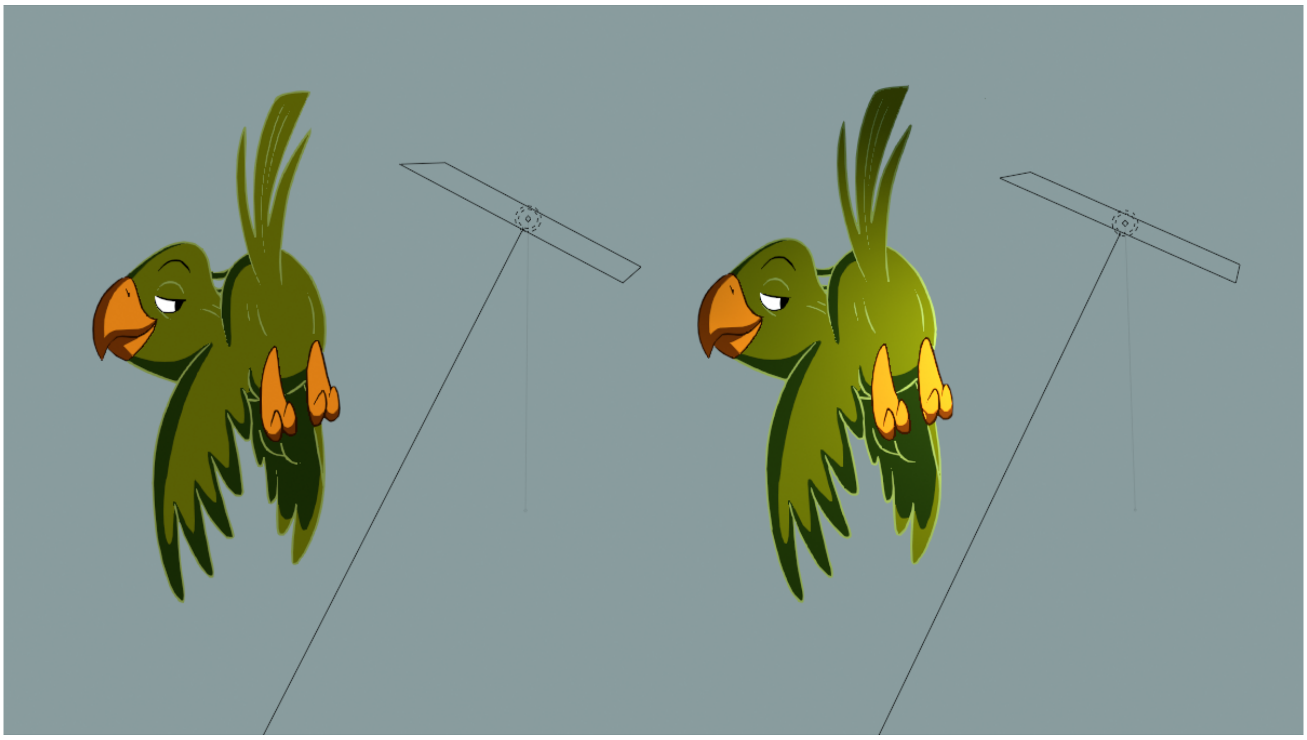


Fig. 1361: Lights disabled (left) and enabled (right).

2.7.3 Structure

Grease Pencil object has three main basic components: *points*, *edit lines* and *strokes*.

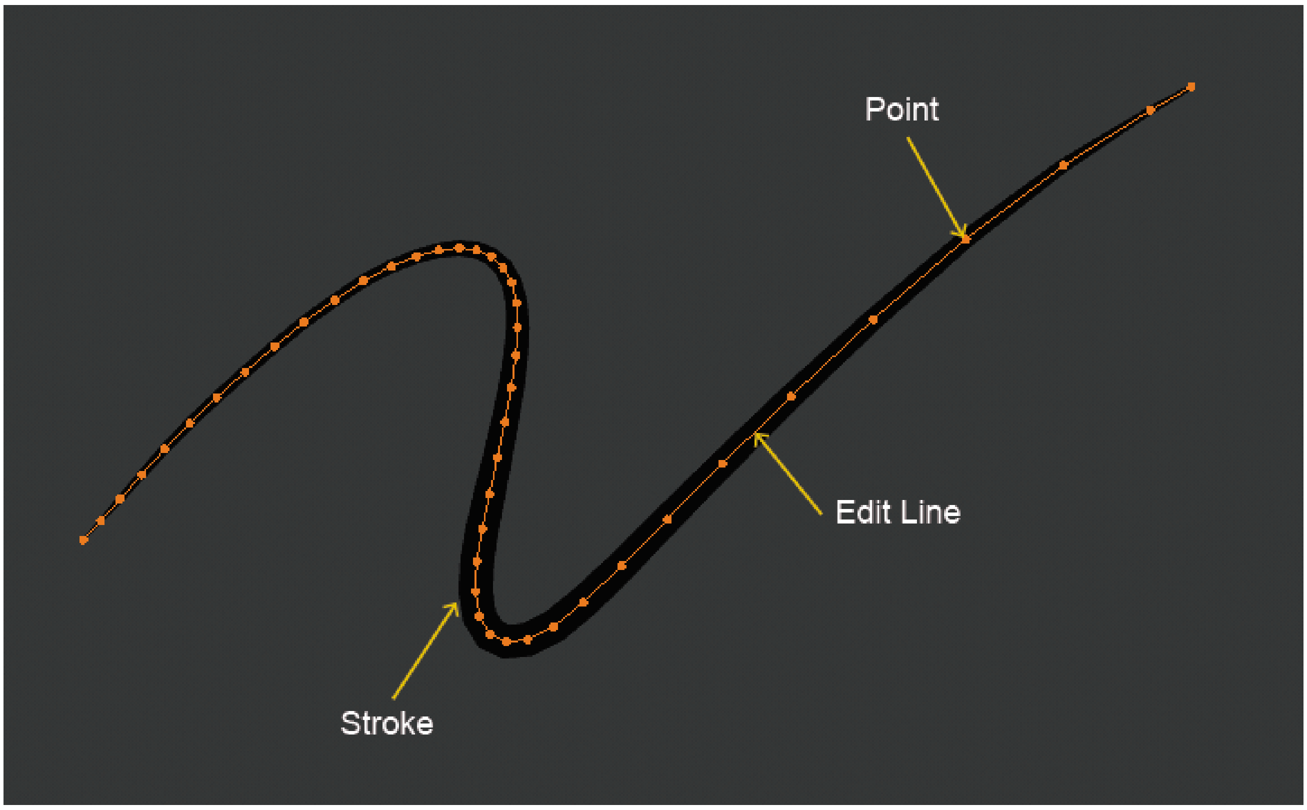


Fig. 1362: Example of Grease Pencil structure.

Points

The main element used in editing Grease Pencil objects are points. Points represent a single point in 3D space.

Each point stores all the properties that define the final appearance of the strokes as its location, thickness, alpha, weight and UV rotation for textures.

Note: Point (Grease Pencil) and Vertex (meshes) are equivalent names.

Edit Lines

Points are always connected by a straight line, which you see when you are editing in Edit Mode or when you look at a stroke in wireframe view. They are invisible on the rendered image and are used to construct the final stroke.

Strokes

The stroke is the rendered image of the points and edit lines, using a particular *Grease Pencil material*. (Grease Pencil materials are linked at stroke level.)

2.7.4 Primitives

Reference

Mode Object Mode and Edit Mode

Menu *Add → Grease Pencil*

Hotkey `Shift-A`

In Object Mode, the *Add Grease Pencil* menu provides three different Grease Pencil primitives:

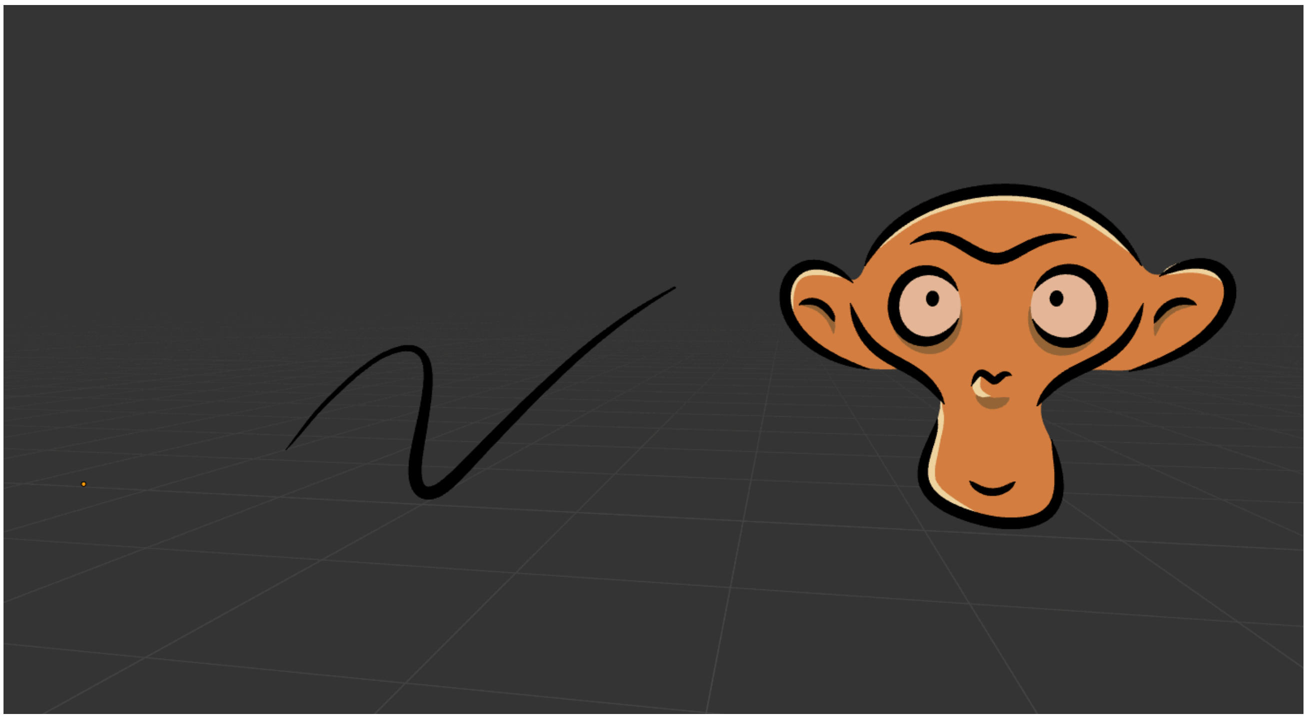


Fig. 1363: Grease Pencil primitives.

Blank

Adds a Grease Pencil object without any stroke.

Stroke

Adds a Grease Pencil object with a simple stroke as a reference.

Monkey

It creates a 2D monkey head. The Monkey's name is "Suzanne" and is Blender's mascot. 2D Suzanne is very useful as a standard test.

Note: Materials and 2D Layers

Adding Stroke or Monkey primitives also adds a set of preset materials and 2D layers to help start drawing with the new object.

2.7.5 Selecting

Reference

Mode Edit Mode

Menu *3D Viewport Header → Select Mode*

Hotkey 1, 2, 3

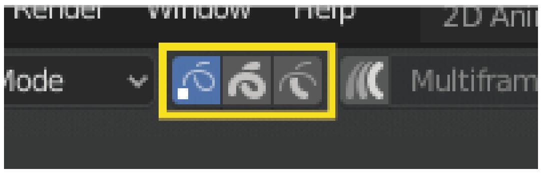


Fig. 1364: Edit Mode selection buttons.

In Edit Mode there are three different selection modes. You can enter the different modes by selecting one of the three buttons in the header.

Points To select individual points.

Strokes To select an entire stroke.

Points in Between To select all points that are between other strokes.

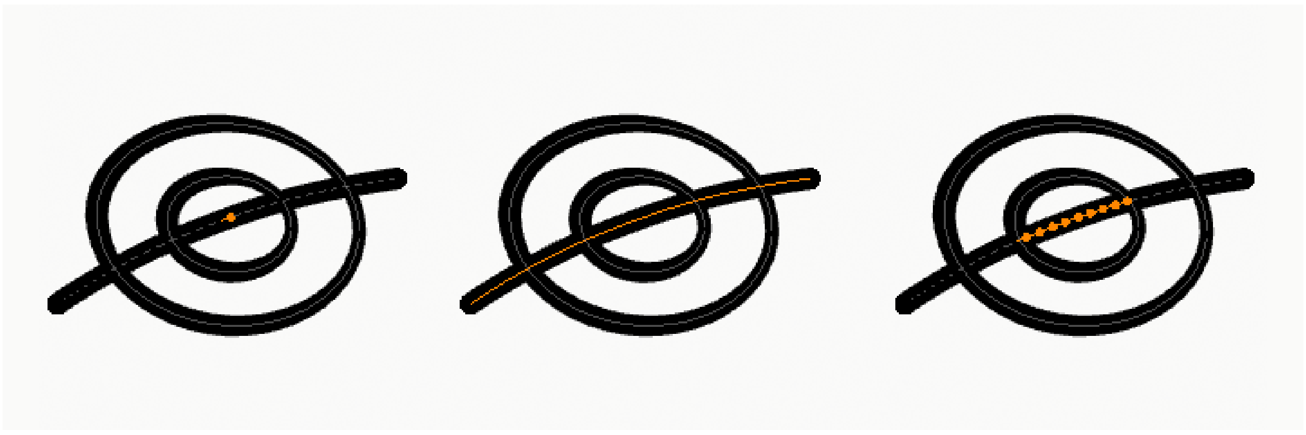


Fig. 1365: Points, stroke and in between stroke selection sample.

Select Menu

Box/Circle/All/None/Invert Select All these options have the same meaning and behavior as in *Object Mode*.

Select Linked

Reference

Mode Edit Mode

Menu *Select → Linked*

Hotkey L, Ctrl-L

L (or Ctrl-L for all) will add to the selection the cursor's nearest control point, and all the linked ones, i.e. all points belonging to the same stroke.

Select Alternated

Reference

Mode Edit Mode

Menu *Select → Alternated*

Hotkey `Shift-L`

Selects alternate points in the selected strokes.

Select Grouped

Reference

Mode Edit Mode

Menu *Select → Grouped*

Hotkey `Shift-G`

Layer Selects all the points/strokes on the same layer.

Material Selects all the points/strokes that share the same material.

Select Vertex Color

Reference

Mode Vertex Paint Mode

Menu *Select → Vertex Color*

Selects all points with a similar vertex color as the current selection.

Tolerance How similar colors are allowed to be; higher values select a wider range of colors.

Select First/Last

Reference

Mode Edit Mode

Menu *Select → First/Last*

These operators will toggle the selection of the first or last point(s) of the stroke(s) in the object. This is useful to quickly find the start of a stroke.

Select More/Less

Reference

Mode Edit Mode

Menu *Select → More/Less*

Hotkey `Ctrl-NumpadPlus`, `Ctrl-NumpadMinus`

The purpose of these tools is to reduce or enlarge the current selection within a stroke (i.e. they will never "go outside" of a stroke or "jump" to another stroke in the same object).

More For each selected point, select *all* its linked points (i.e. one or two…).

Less For each selected point, if *all* points linked to this point are selected, keep this one selected. Otherwise, deselect it.

Hint: When *all* points of a stroke are selected, nothing will happen (as for *Less*, all linked points are always selected, and of course, *More* cannot add any). Conversely, the same goes when no points are selected.

2.7.6 Multiframe

Multiframe allows you to edit, sculpt, or weight painting on several frames at the same time. Extremely useful to avoid repeating a task one frame at a time when animating.

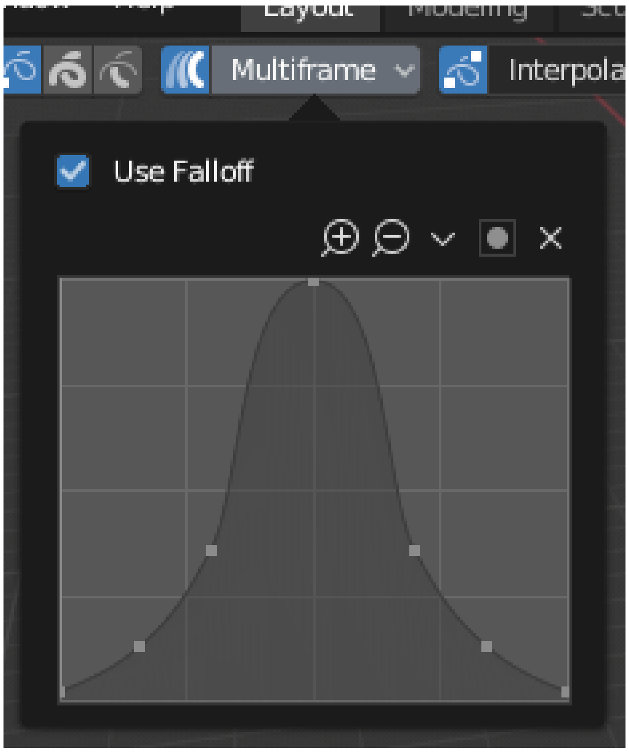


Fig. 1366: Multiframe panel.

Usage

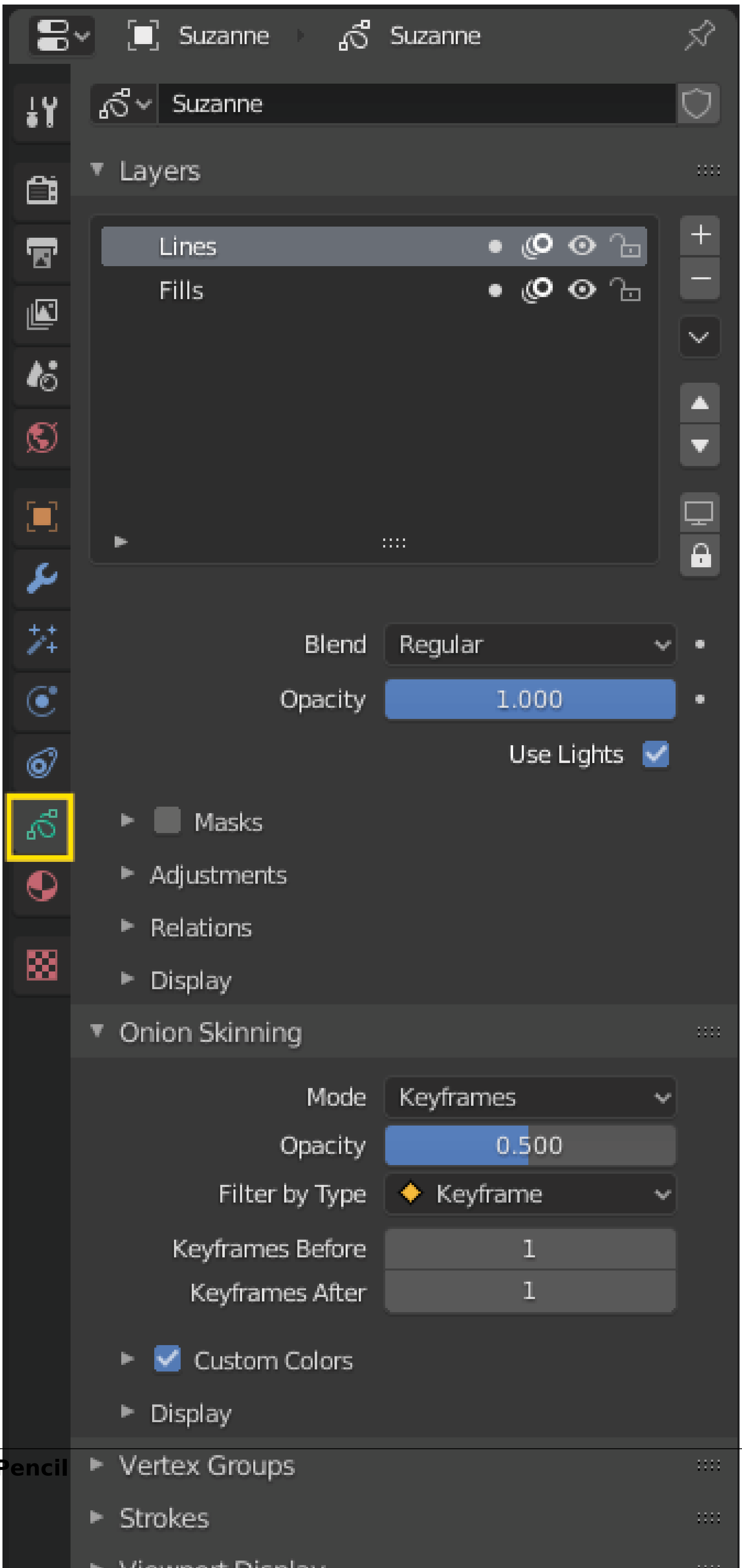
1. Select the desired keyframes to edit or sculpt at the same time.

2. Activate the Multiframe tool in the 3D Viewport's header with the toggle button (faded lines icon).
3. Once activated you can:
   - Select the points in all the selected keyframes and make your editions.
   - Start sculpting. The sculpt brushes will affects all the strokes in the selected keyframes.
   - Start weight painting. The weight paint brush will affect all the strokes in the selected keyframes.

Use Falloff When enabled, the effects on the strokes start to falloff from the current frame as defined by a *curve widget*.

2.7.7 Properties

Object Data



Suzanne

Blend: Regular

Opacity: 1.000

Use Lights: ✓

Mode: Keyframes

Opacity: 0.500

Filter by Type: Keyframe

Keyframes Before: 1

Keyframes After: 1

Custom Colors: ✓

Grease Pencil The Grease Pencil *data-block menu* can be used to link the data between objects.

2D Layers

Strokes can be grouped in 2D layers, a special Grease Pencil layers that help to organize the drawing order and visibility of the strokes.

See *2D Layers* for more information.

Onion Skinning

Onion skinning is used in animation to see several frames at once and make decisions or edits based on how the previous/next frames are drawn.

See *Onion Skinning* for more information.

Vertex Groups

Vertex groups can be used to assign a group or weighted group to some operator. An object can have several weight groups and can be assigned in *Weight Paint Mode*.

See *Vertex Groups* for more information.

Strokes

General settings for Grease Pencil strokes.

See *Strokes* for more information.

2D Layers

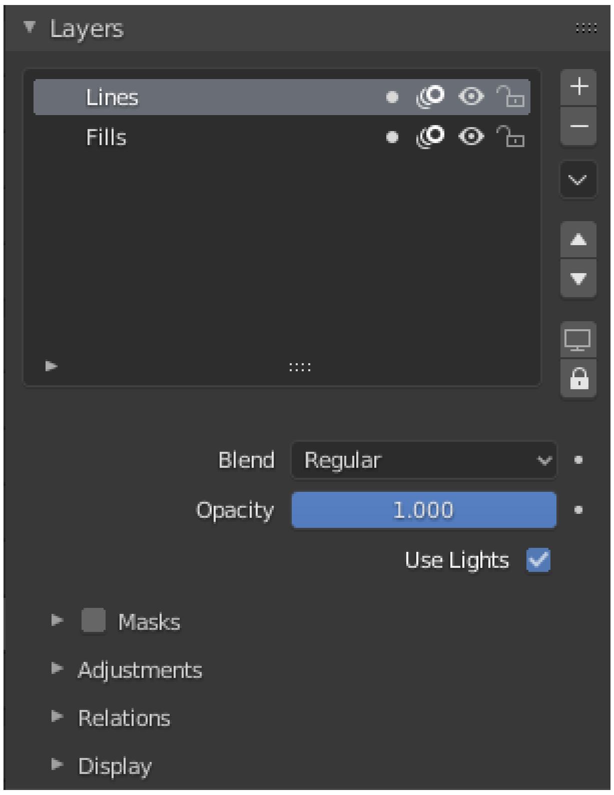
Reference

Mode All Modes

Panel *Object Data tab → Layers*

Hotkey Y

Layers List



Blend: Regular
Opacity: 1.000
Use Lights: ✓

Fig. 1368: Grease Pencil Layers panel.

Grease Pencil objects each have a list of 2D layers for grouping and arranging strokes in a *List view*. Any stroke can only belong to a single 2D layer. There is always only one active layer in the list (the selected one). When you draw, the new strokes are added to the active layer. By default the view order of the layers in the viewport is top to bottom.

Every layer correspond to a channel in the Dope Sheet editor (in Grease Pencil mode). See *Dope Sheet* for more information. Layers can also be used together with Modifiers to only affects part of your drawing. See *Modifiers* for more information.

Tip: Sometimes the layers you are not working on can be a distraction. Activate *Fade Layers* in overlays to control the opacity of the non-active layers. See *Overlays* for more information.

Next to the layer name there are four icons buttons that control common properties of the layer:

Mask (mask icon) Toggle the *Masks* visibility in the layer.

Onion Skinning (onion skin icon) Toggle the use the layer for *Onion Skinning*.

Viewport/Render Visibility (eye icon) Toggle layer visibility in the viewport and in render.

Lock (padlock icon) Toggle layer from being editable.

Below the layers list there are additional common settings:

Blend The layer blending operation to perform. See *Color Blend Modes*.

Opacity Used to set the opacity of the layer.

Use Lights When enabled, the layer is affected by lights.

Specials

Duplicate Layer Makes an exact copy of the selected layer appending a number to differentiate its name.

Show All Turns on the visibility of every layer in the list.

Hide Others Turns off the visibility of every layer in the list except the active one.

Lock All Locks edition of all the layers in the list.

Unlock All Unlocks edition of all the layers in the list.

Autolock Inactive Layer Locks automatically the edition of every layer in the list except the active one. This way you avoid to make unwanted changes in other layers without the need to lock them every time.

Merge Down Merge the selected layer with the layer below, the new layer keeps the name of the lower layer.

Copy Layer to Object Makes a copy of the layer and move it to the selected Grease Pencil object.

Lock & Visibility General Controls

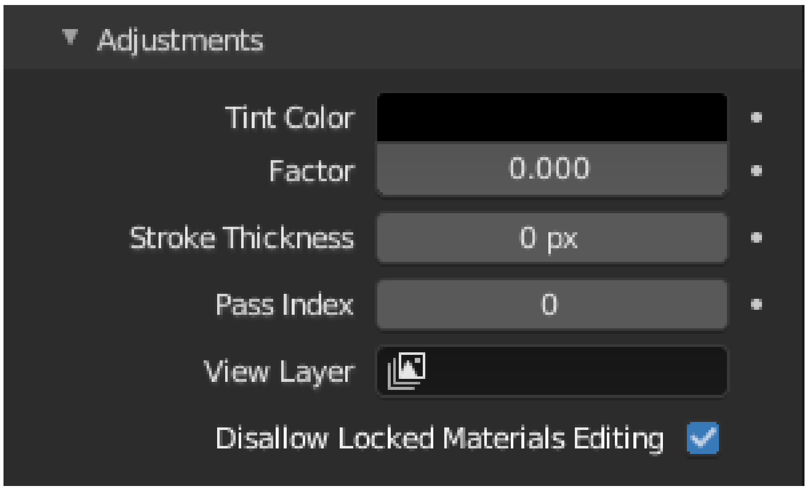
Lock (padlock icon) Toggle whether the active layer is the only one that can be edited.

Visibility (screen icon) Toggle whether the active layer is the only one that can be edited and is visible.

Masks

In a *List view* of layers affected by a layer mask. See *Masks* for more information.

Adjustments



Tint Color
Factor 0.000
Stroke Thickness 0 px
Pass Index 0
View Layer
Disallow Locked Materials Editing ✓

Fig. 1369: Layers adjustment panel.

Tint Color/Factor Color that tint any material colors used in the layer. The *Factor* controls the amount of tint color to apply.

Stroke Thickness Thickness value that override the strokes thickness in the layer.

Pass Index The layer index number can be used with some modifiers to restrict changes to only certain areas.

See *Modifiers* for more information.

View Layer Defines the View Layer to use for the Grease Pencil layer. If empty, the layer will be included in all View Layers. This is useful to separate drawings parts for *compositing*.

Disallow Locked Materials Editing Avoids editing locked materials in the layer. When disabled, any material can be edited even if they are locked in the material list.

Relations

Parent/Type Select a Parent object and Type to manipulate the layer. The layer will inherit the transformations of the parent, this is especially useful when rigging for cut-out animation.

Display

Custom Channel Color Sets the color to use in the channel region of the *Dope Sheet*.

Show Only On Keyframed Makes the layer visible in the viewport only if it has a keyframe in the actual frame. This helps for example when you are in the inking process using the *Fill* tool and want to only see the strokes that are in the actual frame to avoid fill in unwanted regions.

Masks

Layers List

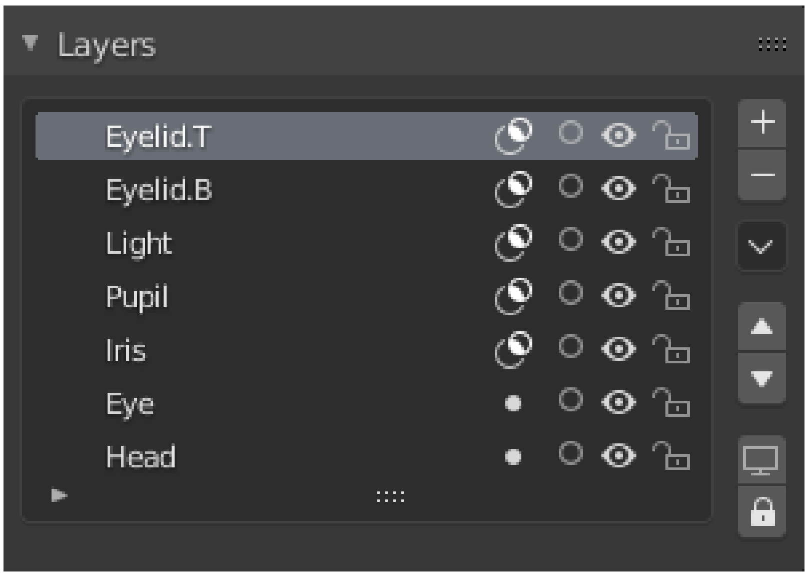


Fig. 1370: Layer list with masked layers.

In Grease Pencil there are no special mask layers, any layer can act as a mask for other layers. The mask system is flexible enough to allow top-bottom and bottom-top masking.

Layers used as mask can use all the blend modes and different opacity values like any other layer.

Note: If you want to make a full transparent masking you will have to set the mask layer(s) opacity to 0.

By activating the mask toggle (mask icon) next to the layer name or using the checkbox on the masks panel header the layer becomes prepared to be masked by other layer(s).

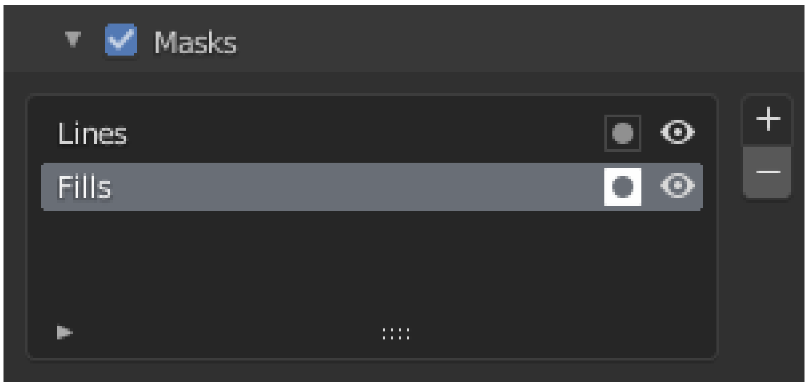


Fig. 1371: Masks list view.

Masks List

The layer/s that will act as mask of the current layer could be added to the Mask *list view*.

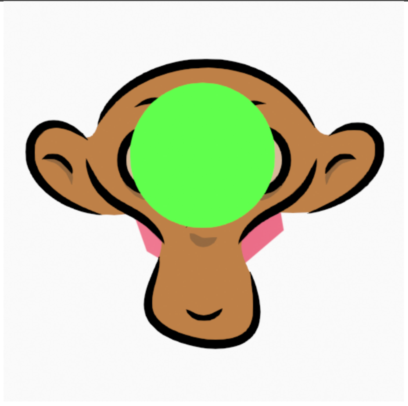
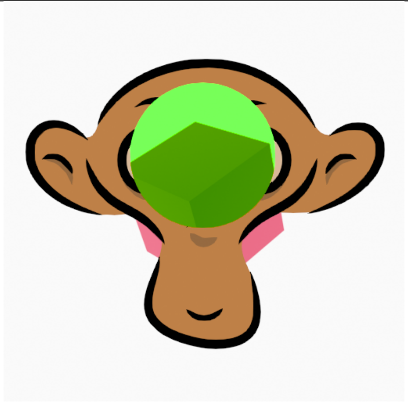
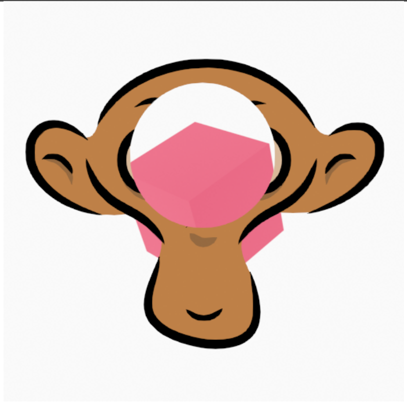
In the Masks list next to the layers name there are two icons buttons that control common properties of the layer mask:

Invert (mask icon) Inverts the mask.

Viewport/Render Visibility (eye icon) Toggle layer visibility in the viewport and in render.

Example

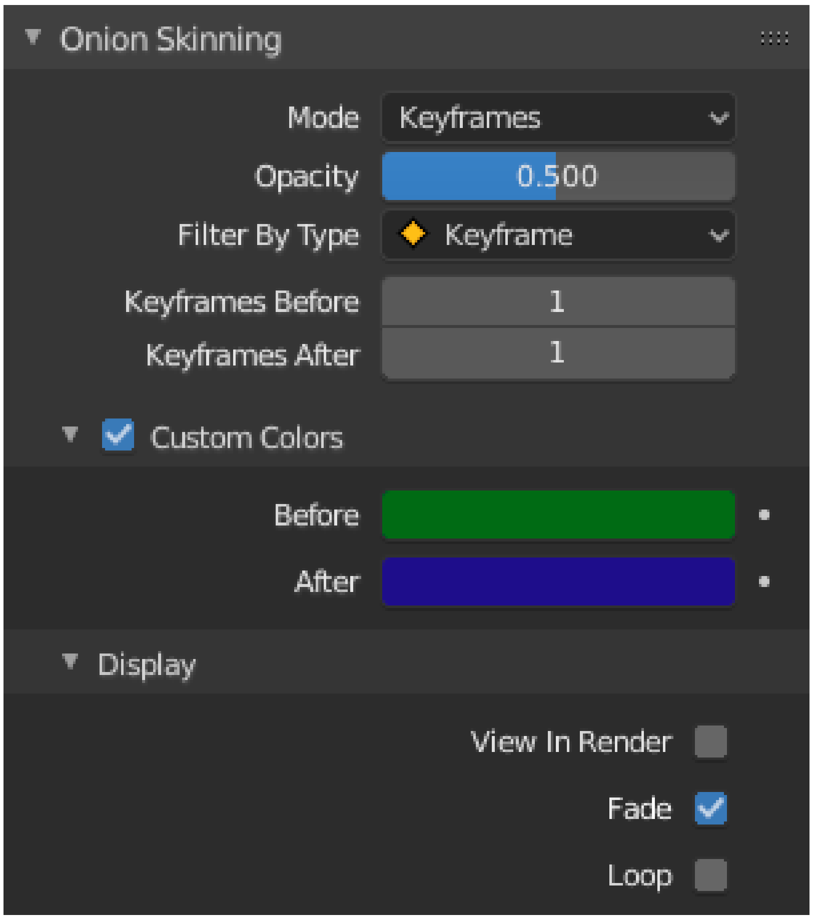
Table 70: Mask (green circle) samples.

 Fig. 1372: Original image (Blend: Regular, Opacity: 1).	 Fig. 1373: Blend: Hard Light, Opacity: 1.	 Fig. 1374: Blend: Regular, Opacity: 1.

Onion Skinning

Onion Skinning show ghosts of the keyframes before and after the current frame allowing animators to make decisions in the animation sequence.

The main switch to show/hide Onion Skinning is in the *Viewport Overlays*, but Grease Pencil Onion Skinning is per-layer and the visibility can be toggle in the layer list. See *2D Layers* for more information.


Onion Skinning

Mode: Keyframes

Opacity: 0.500

Filter By Type: Keyframe

Keyframes Before: 1

Keyframes After: 1

Custom Colors ✓

Before 🟩

After 🟦

Display

View In Render ☐

Fade ✓

Loop ☐

Fig. 1375: Onion Skinning panel.

Options

Mode

- **Keyframes** Shows Keyframes in the range determined by the *Before/After* settings.
- **Frames** Shows Frames in the range determined by the *Before/After* settings.
- **Selected** Shows only on the manually selected keyframes in the Dope Sheet.

Opacity Control the opacity of the ghost frames.

Filter by Type Filters what type of frames to show in the Onion Skinning range.

Keyframes Before/After Sets how many frames or keyframes, depending on the *Mode*, to show before and after the current frame.

Custom Colors

Before/After Color to use before and after the current frame on ghost frames.

Display

View in Render Show the onion skinning in final render image e.g. for a motion blur effect.

Fade Opacity of the ghosts frames decrease the further away from the current frame.

Loop Help working on loop animations showing the first keyframe/frame as ghost when you are on the last frame of your animation.

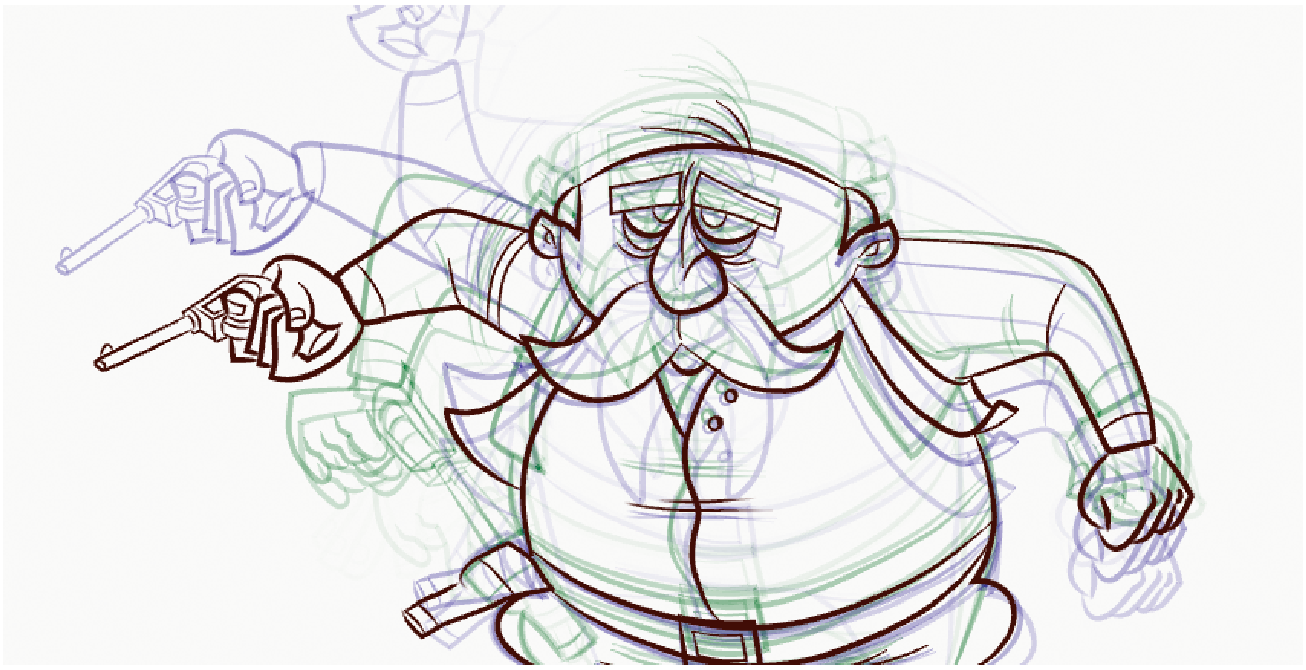
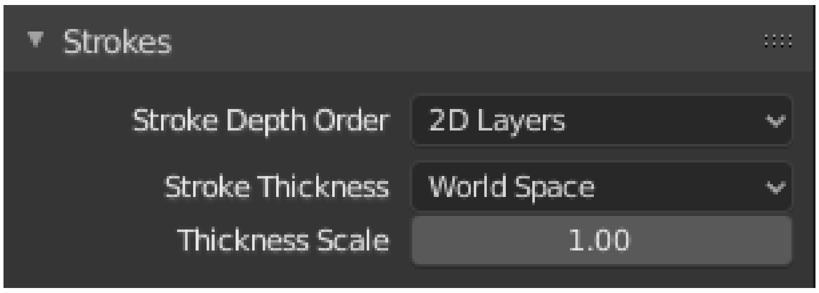


Fig. 1376: An example of Onion Skinning activated.

Strokes



▼ Strokes

Stroke Depth Order: 2D Layers

Stroke Thickness: World Space

Thickness Scale: 1.00

Fig. 1377: Strokes panel.

General settings for Grease Pencil strokes.

Stroke Depth Order

2D Layers The Strokes drawing order respect the order of the 2D layers list (top to bottom) and ignores the real position of the strokes in 3D space. See *2D Layers* for more information.

3D Location The strokes drawing order is based on the stroke location in 3D space.

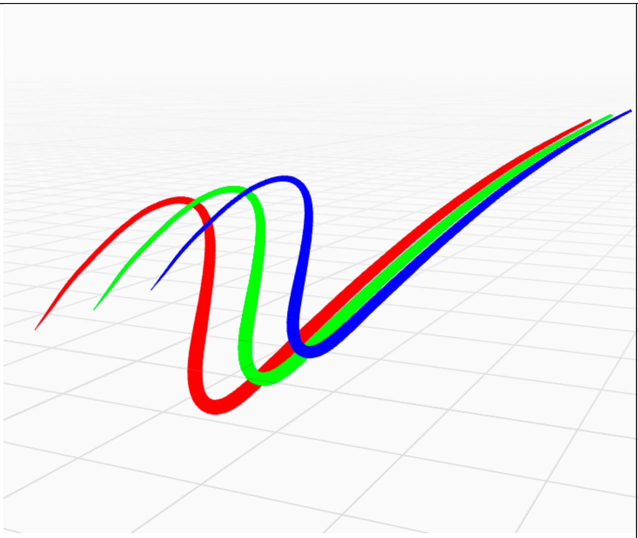


Fig. 1378: Blue, Green and Red strokes in three different layers using 2D Layers depth order.

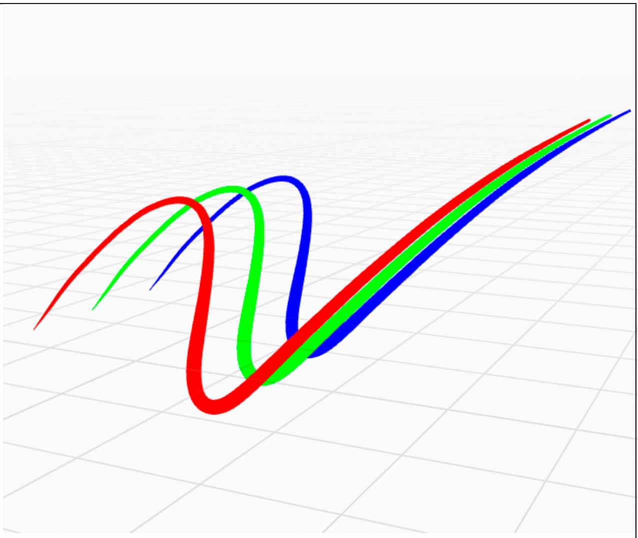


Fig. 1379: Blue, Green and Red strokes in three different layers using 3D Location depth order.

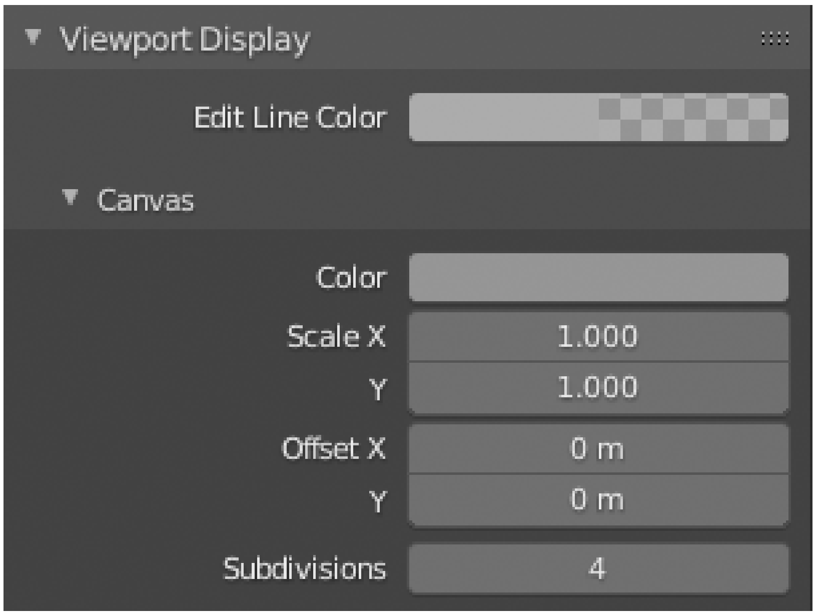
Stroke Thickness

World Space The thickness is relative to world space. Stroke thickness change with the screen zoom factor.

Screen Space The thickness is relative to screen space. Stroke thickness remains the same regardless of the screen zoom factor.

Thickness Scale Sets a thickness scale factor for all strokes.

Viewport Display



Viewport Display

Edit Line Color

Canvas

Color

Scale X 1.000

Y 1.000

Offset X 0 m

Y 0 m

Subdivisions 4

Fig. 1380: Viewport Display panel.

Display settings for Edit Lines in *Edit Mode* and *Sculpt Mode*.

Edit Line Color Sets the color of the Edit Lines.

Canvas

In 3D space sometimes is difficult to assess on which plane are you drawing. The Canvas is a display overlay helper that shows a grid at the current *Drawing Plane*. You can enable the Canvas visualization in the *Viewport Overlays*.

See *Drawing Plane* for more information.

Color Color of the Canvas grid lines.

Scale X/Y Defines the X and Y scale of the Canvas.

Offset X/Y Sets the Canvas position offset from the object's origin.

Subdivisions Specifies the number of subdivisions to use for the grid.

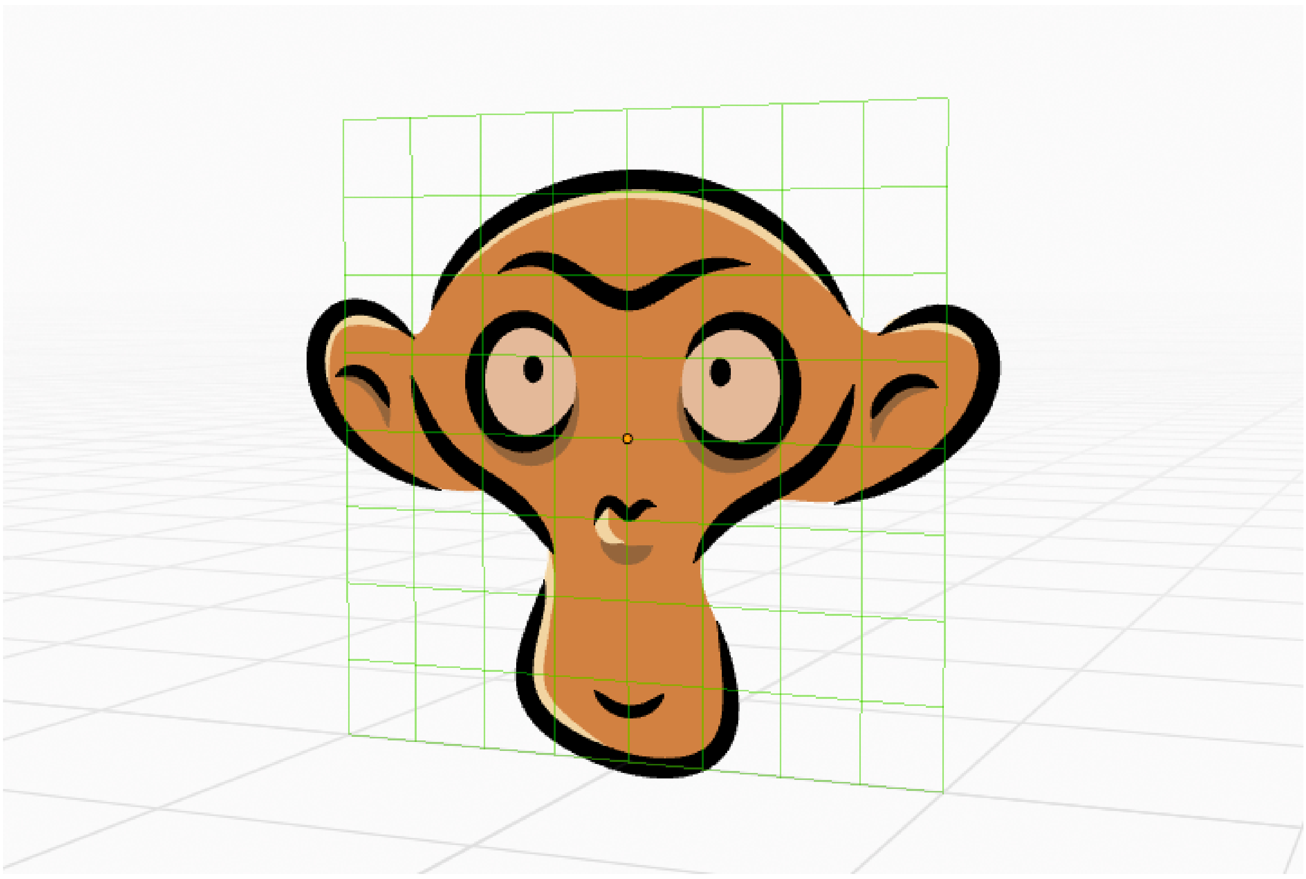


Fig. 1381: Canvas example on the XZ drawing plane using a green color grid.

2.7.8 Modifiers

Introduction

Reference

Panel *Properties → Modifiers*

Grease Pencil has their own set of modifiers. Modifiers are automatic operations that affect an object in a non-destructive way. With modifiers, you can perform many effects automatically that would otherwise be too tedious to do manually and without affecting the base geometry of your object.

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 forming the modifier stack and *Apply* a modifier if you wish to make its changes permanent.

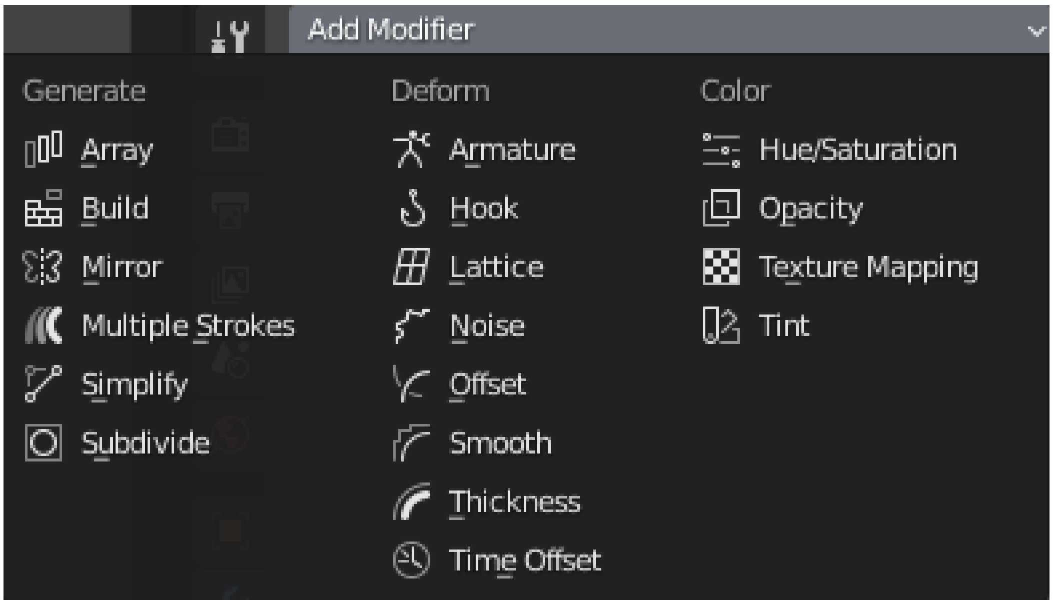


Fig. 1382: Grease Pencil Modifiers menu.

There are three types of modifiers for Grease Pencil:

Generate The *Generate* group of modifiers includes constructive tools that either change the general appearance of or automatically add new geometry to an object.

Deform The *Deform* group of modifiers only changes the shape of an object without adding new geometry,

Color The *Color* group of modifiers change the object color output.

Interface

Each modifier's interface shares the same basic components like modifiers for meshes.

See *Modifiers Interface* for more information.

Note: Grease Pencil strokes, unlike meshes, still can not be edited directly in the place.

Influence Filters

Most of the modifiers share some special properties that restrict the effect only to certain items.

Layer Restricts the effect only to one layer or to any layers that share the same pass index.

Material Restricts the effect only to material that share the same material or pass index.

Vertex Group Restricts the effect only to a vertex group.

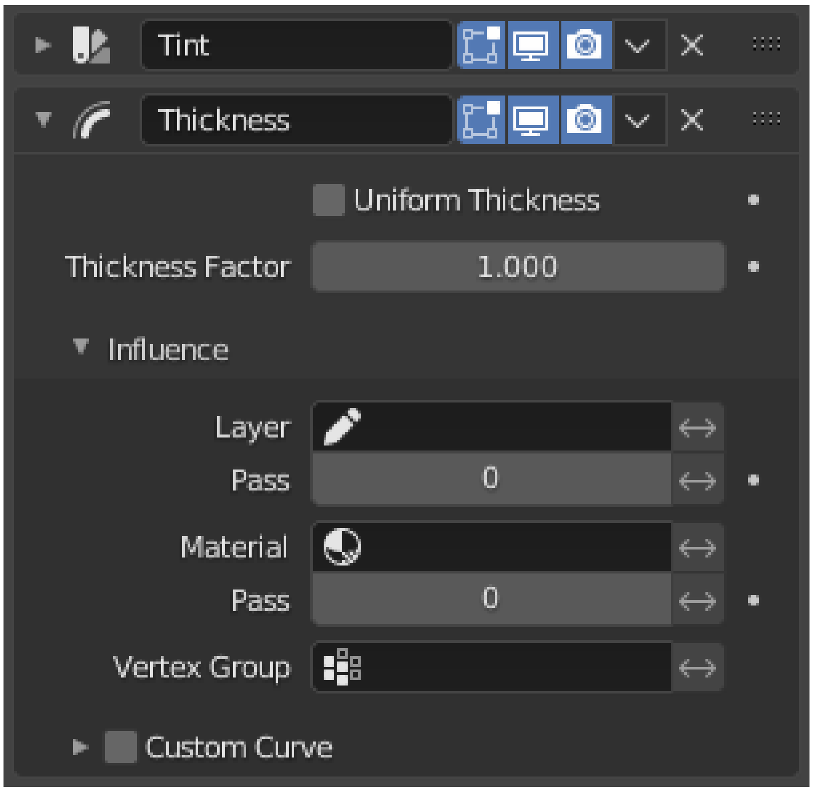


Fig. 1383: Panel layout (Thickness modifier as an example).

Custom Curve When enabled, use a custom curve to shape the effect along the strokes from start to end points.

The Invert toggle <-> allows you to reverse the filters behavior.

Generate

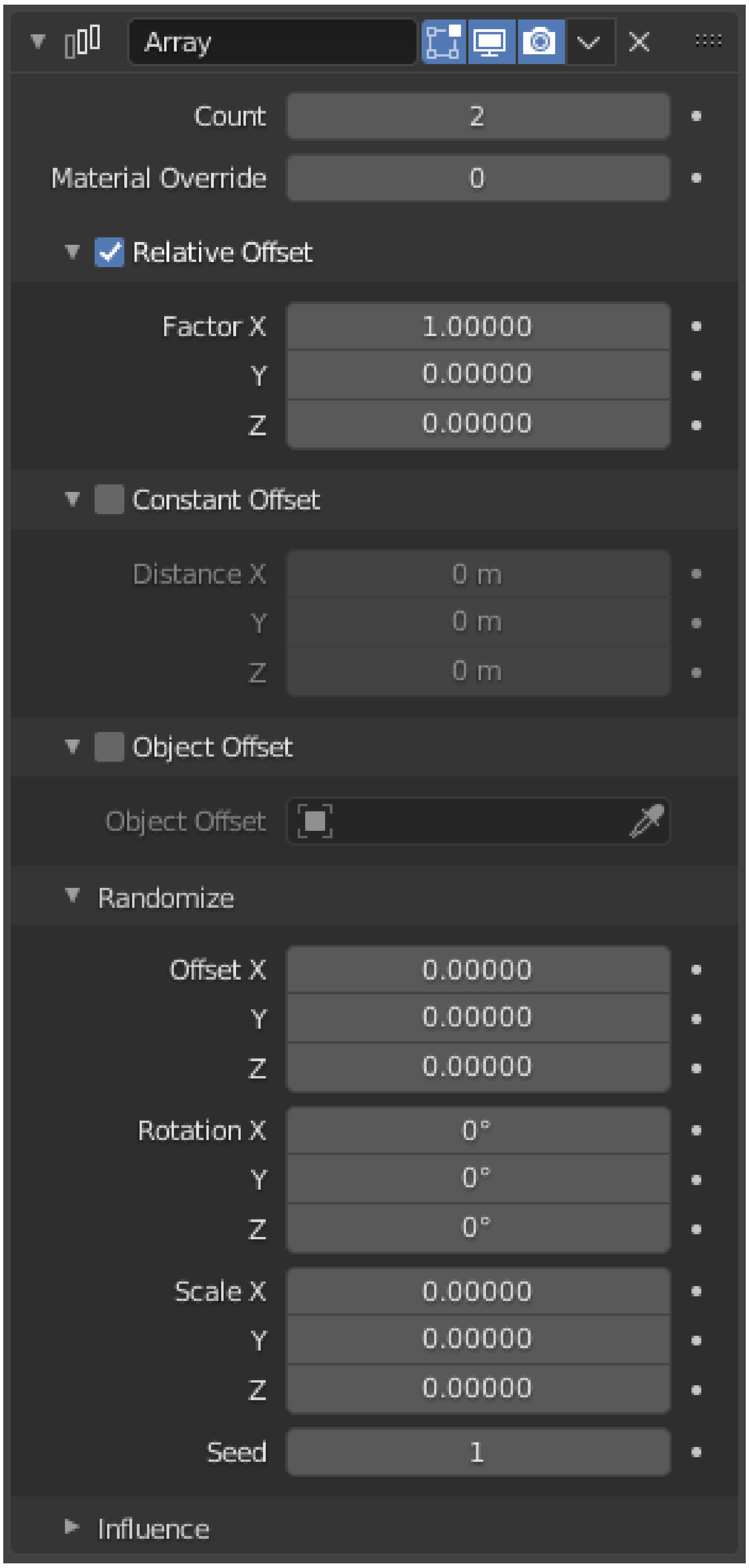
Array Modifier

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.

Useful for creating complex repetitive drawings.

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

Options



Array

Count 2

Material Override 0

Relative Offset ✓

Factor X 1.00000

Y 0.00000

Z 0.00000

Constant Offset

Distance X 0 m

Y 0 m

Z 0 m

Object Offset

Object Offset

Randomize

Offset X 0.00000

Y 0.00000

Z 0.00000

Rotation X 0°

Y 0°

Z 0°

Scale X 0.00000

Y 0.00000

Z 0.00000

Seed 1

Influence

Fig. 1384: The Array modifier.

Count Total number of copies.

Material Override Index of the material to use on duplicated strokes (0 use strokes original materials).

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.

Constant Offset

Factor 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

Distance X, Y, Z 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.

Randomize

Offset X, Y, Z Add random offset values to the copies.

Rotation X, Y, Z Add random rotation values to the copies.

Scale X, Y, Z Add random scale values to the copies.

Seed Seed used by the pseudo-random number generator.

Note: The *Depth Order* used in the Grease Pencil object has an influence on the visualization of the strokes when using the Array modifier. See *Depth Order* for more information.

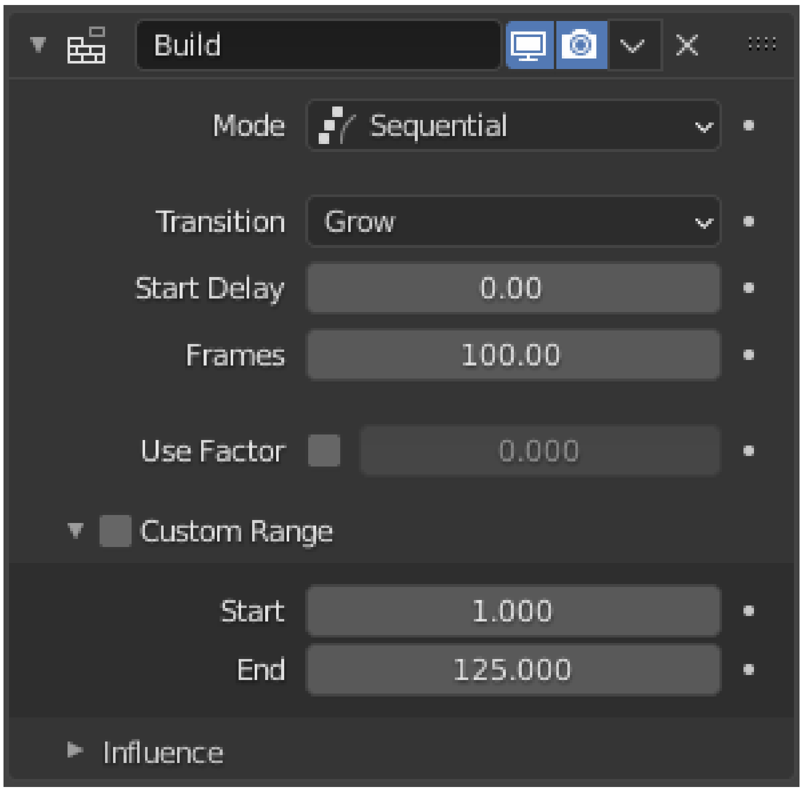
Influence

See *Influence Filters*.

Build Modifier

The *Build* modifier make strokes appear or disappear in a frame range to create the effect of animating lines being drawn or erased.

Options



Mode: Sequential
Transition: Grow
Start Delay: 0.00
Frames: 100.00
Use Factor: 0.000
Custom Range
Start: 1.000
End: 125.000

Fig. 1385: The Build modifier.

Mode Determines how many strokes are being animated at a time.

Sequential Strokes appear/disappear one after the other, but only a single one changes at a time.

Concurrent Multiple stroke appear/disappear at a time.

If enabled you can set the *Time Alignment*.

Time Alignment

Align Start All stroke start at the same time (i.e. shorter strokes finish earlier).

Align End All stroke end at the same time (i.e. shorter strokes start later).

Transition Determines the animation type to build the strokes.

Grow Shows points in the order they occur in each stroke, from first to last stroke. (Simulating lines being drawn.)

Shrink Hide points from the end of each stroke to the start, from last to first stroke. (Simulating lines being erased.)

Fade Hide points in the order they occur in each stroke, from first to last stroke. (Simulating ink fading or vanishing after getting drawn.)

Start Delay Number of frames after each Grease Pencil keyframe before the modifier has any effects.

Frames Maximum number of frames that the build effect can run for. (Unless another Grease Pencil keyframe occurs before this time has elapsed.)

Use Factor Use a defined percentage factor to control the amount of the stroke that is visible.

Custom Range

If enabled, only modify strokes during the specified frame range.

Start/End Determines the start and end frame for the build effect.

Influence Filters

See *Influence Filters*.

Mirror Modifier

The *Mirror* modifier mirrors the strokes 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

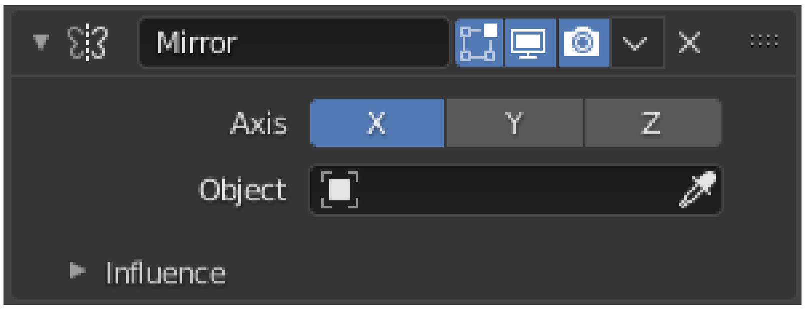


Fig. 1386: The Mirror modifier.

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 stroke 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.

Object A *Data ID* 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).

Influence

See *Influence Filters*.

Multiple Strokes

The *Multiple Strokes* modifier generate multiple parallel strokes around the original ones.

Options

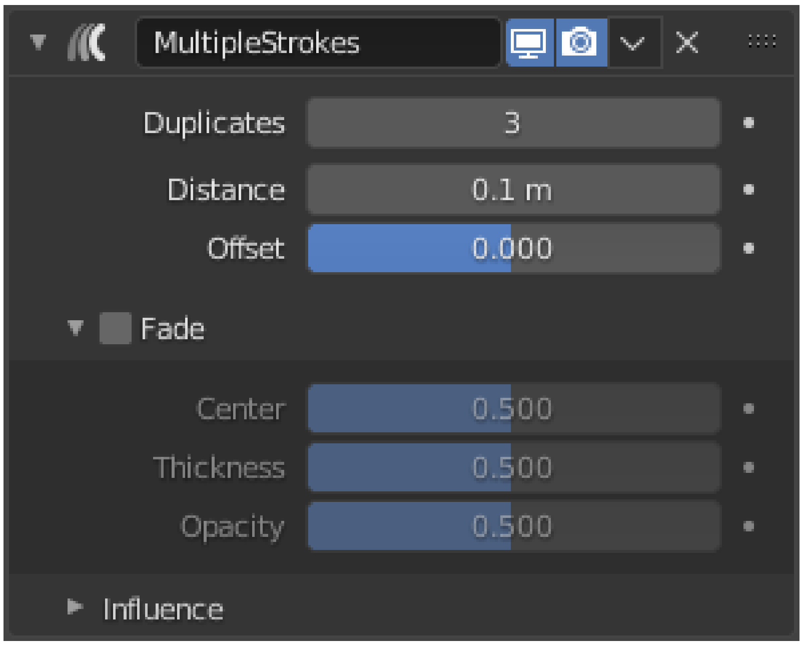

MultipleStrokes
Duplicates 3
Distance 0.1 m
Offset 0.000
Fade
Center 0.500
Thickness 0.500
Opacity 0.500
Influence

Fig. 1387: The Multiple Strokes modifier.

Duplicates The number of additional strokes.

Distance Distance between the original and the duplicate strokes.

Offset Control the offset position (inner or outer) for duplicate strokes.

Fade

When activated, the duplicate strokes fades out using their opacity or thickness.

Center Control the initial position for the fading.

Thickness Fade influence on strokes thickness.

Opacity Fade influence on strokes opacity.

Influence

See *Influence Filters*.

Simplify Modifier

The *Simplify* modifier allows you to reduce the amount of points in the strokes. The goal of this modifier is reduce points while maintaining the lines shape.

Apply the modifier can help to obtain a better performance (more FPS) while animating.

Options

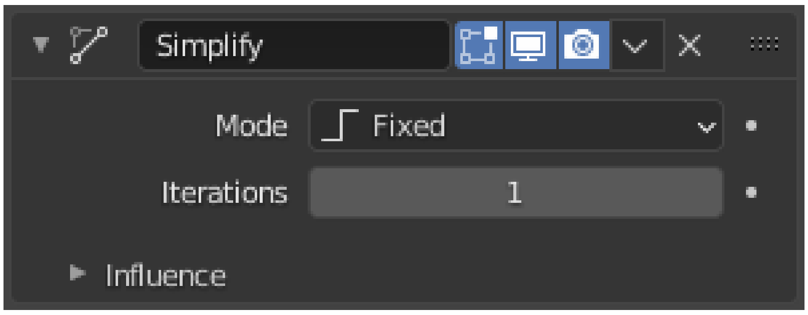

Simplify
Mode: Fixed
Iterations: 1
Influence

Fig. 1388: The Simplify modifier.

Mode Determines how to reduce points in the strokes.

- **Fixed** Deletes alternated points in the strokes, except the start and end points.
  - **Iterations** Number of times to repeat the procedure.
- **Adaptive** Uses the RDP algorithm (Ramer-Douglas-Peucker algorithm) for points deletion. The algorithm try to obtain a similar line shape with fewer points.
  - **Factor** Controls the amount of recursively simplifications applied by the algorithm.
- **Sample** Recreates the stroke geometry with a predefined length between points.
  - **Length** The distance between points on the recreated stroke. Smaller values will require more points to recreate the stroke, while larger values will result in fewer points needed to recreate the curve.
- **Merge** Simplifies the strokes by merging points that are closer than a specified distance to each other.
  - **Distance** Sets the distance threshold for merging points.

Influence

See *Influence Filters*.

Example

Table 71: Fixed Mode sample.

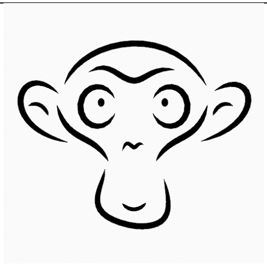
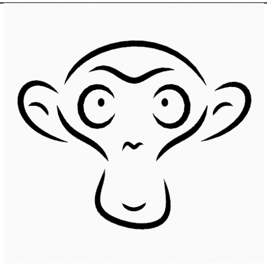
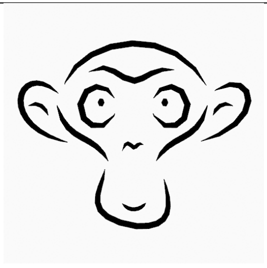
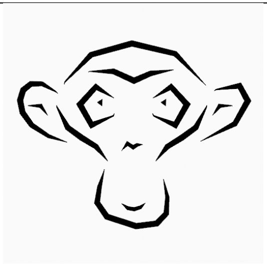
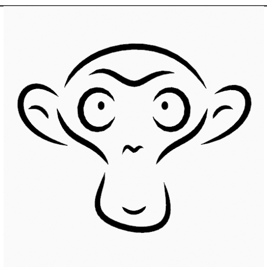
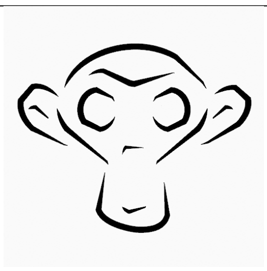
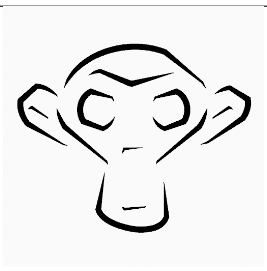
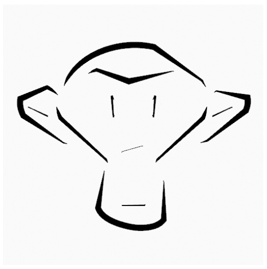
 Fig. 1389: Original Model.	 Fig. 1390: Iteration: 1.	 Fig. 1391: Iteration: 2.	 Fig. 1392: Iteration: 3.

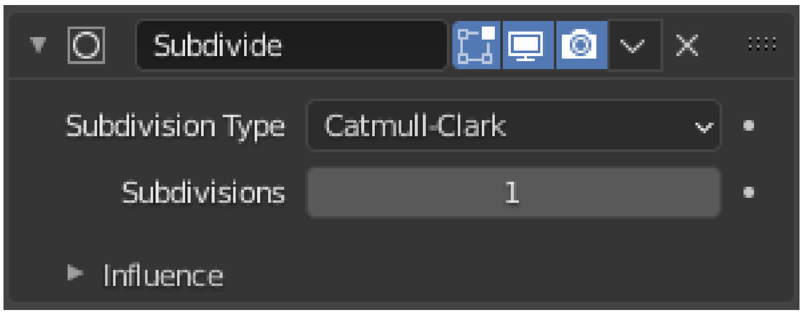
Table 72: Adaptive Mode sample.

 Fig. 1393: Original Model.	 Fig. 1394: Factor: 0.1.	 Fig. 1395: Factor: 0.5.	 Fig. 1396: Factor: 1.

Subdivide Modifier

The *Subdivide* modifier subdivide the strokes by inserting points between other points to the lines.

Options



Subdivision Type: Catmull-Clark
Subdivisions: 1

Fig. 1397: The Subdivide modifier.

Subdivision Type

> **Catmull-Clark** The default option, subdivides and smooths the surfaces.
>
> **Simple** Only subdivides the surfaces, without any smoothing.

Subdivisions Recursively adds more points.

Influence

See *Influence Filters*.

Deform

Armature Modifier

The *Armature* Modifier is used for building skeletal systems for animating the poses of characters and anything else which needs to be posed.

By adding an armature to an object, this object can be deformed accurately so that geometry does not have to be animated by hand.

See also:

For more details on armatures usage, see the *armature section*.

Options

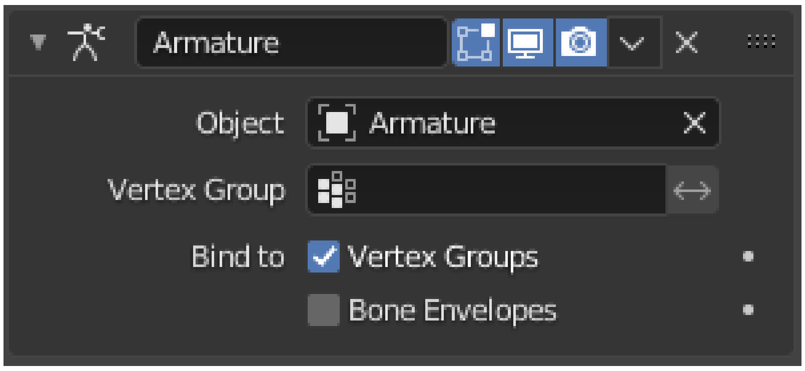


Fig. 1398: The Armature modifier.

Object The name of the armature object used by this modifier.

Vertex Group The name of a vertex group of the object, the weights of which will be used to determine the influence of this Armature Modifier's result when mixing it with the results from other *Armature* ones.

Only meaningful when having at least two of these modifiers on the same object, with *Multi Modifier* activated.

Invert <-> Inverts the influence set by the vertex group defined in previous setting (i.e. reverses the weight values of this group).

Bind to

Vertex Groups When enabled, bones of a given name will deform points which belong to *vertex groups* of the same name. E.g. a bone named "forearm", will only affect the points in the "forearm" vertex group.

The influence of one bone on a given point is controlled by the weight of this point in the relevant group. A much more precise method than *Bone Envelopes*, but also generally longer to set up.

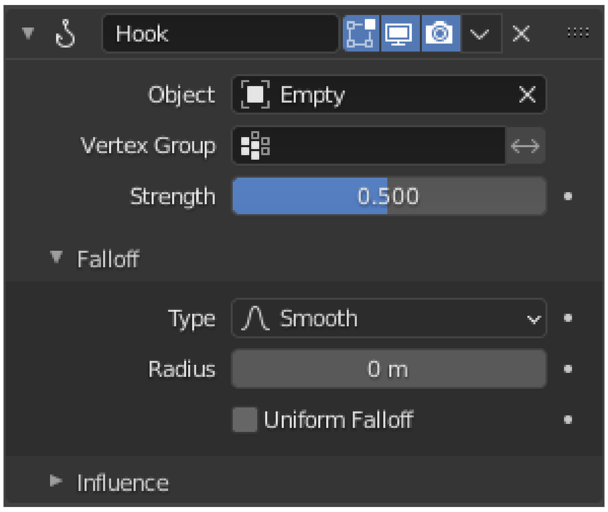
Bone Envelopes When enabled, bones will deform points or control points near them, defined by each bone's envelope radius and distance. Enable/Disable bone *envelopes* defining the deformation (i.e. bones deform points in their neighborhood).

Hook Modifier

The *Hook* Modifier is used to deform stroke points using another object (usually an empty or a bone but it can be any object).

As the hook moves, it pulls points from the strokes with it. You can think of it as animated *Proportional Editing*.

Options



Object: Empty
Vertex Group:
Strength: 0.500
Type: Smooth
Radius: 0 m
Uniform Falloff

Fig. 1399: The Hook modifier.

Object The name of the object to hook points to.

Vertex Group Restricts the effect only to a vertex group.

Strength Adjust this hooks influence on the stroke points, were (0.0 to 1.0) (no change to fully follow the hook).

Falloff

Type This can be used to adjust the type of curve for the Strength attenuation. You can also define a custom curve to get a much higher level of control.

Radius The size of the hooks influence.

Uniform Falloff This setting is useful when using hooks on scaled objects, especially in cases where non-uniform scale would stretch the result of the hook.

Influence

See *Influence Filters*.

Note: The Hook Modifier stores points indices from the original strokes to determine what to affect; this means that modifiers that generate geometry, like a Subdivision Surface Modifier, should always be applied **after** the Hook Modifier; otherwise the generated geometry will be left out of the hook's influence.

Example

Lattice Modifier

The Lattice modifier deforms the base object according to the shape of a *Lattice* object.

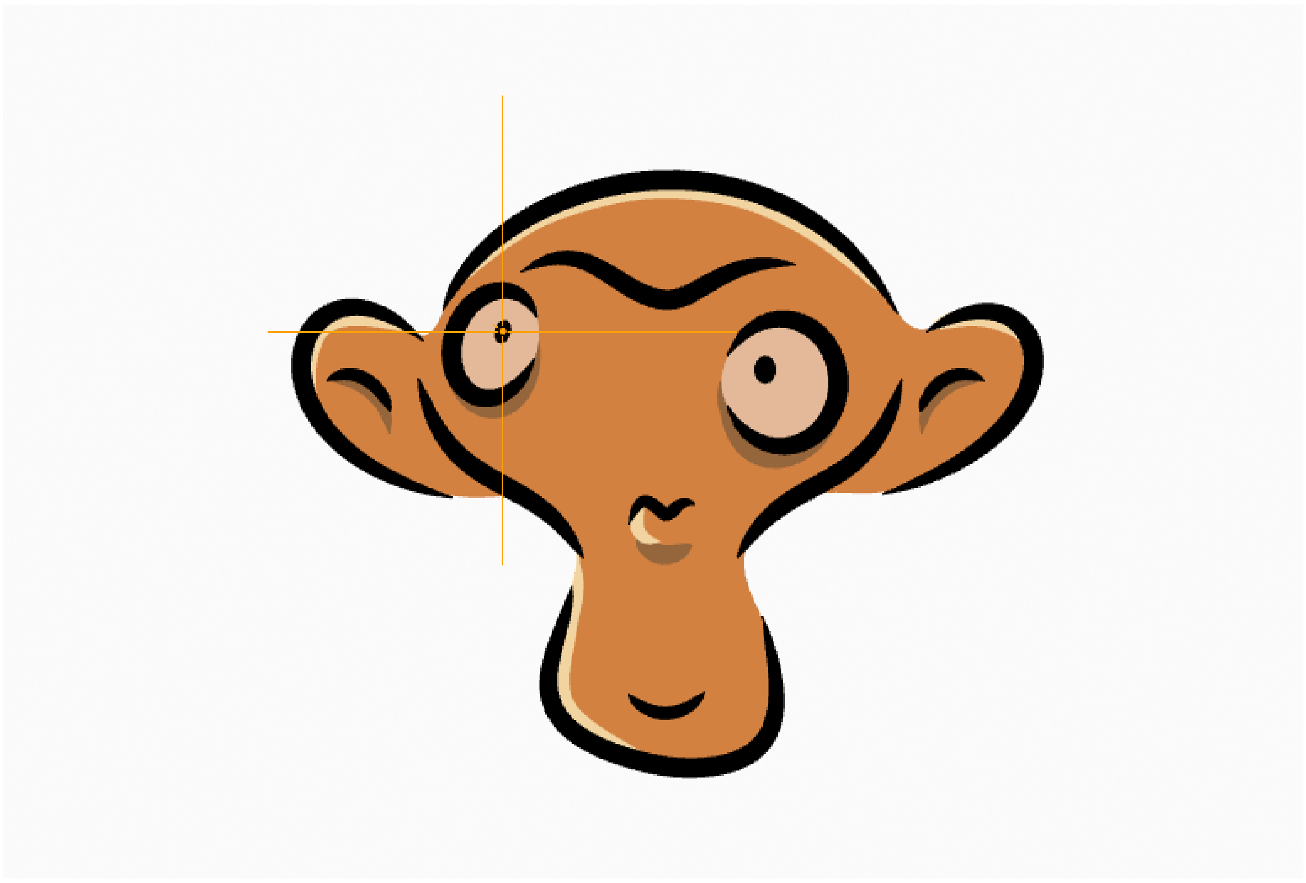
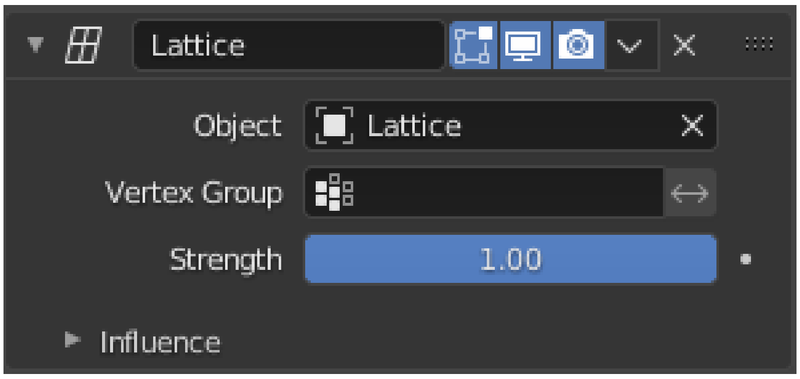


Fig. 1400: Empty used as a hook to manipulate a vertex group (right eye of the monkey).

Tip: A Lattice Modifier can quickly be added to selected objects by selecting them all, then selecting the lattice object last and pressing `Ctrl-P` and choosing *Lattice Deform*. This will both add Lattice Modifiers to the selected objects and parent them to the lattice.

Options


Lattice

Object: Lattice

Vertex Group

Strength: 1.00

Influence

Fig. 1401: Lattice Modifier.

Object The *Lattice* object with which to deform the base object.

Vertex Group Restricts the effect only to a vertex group.

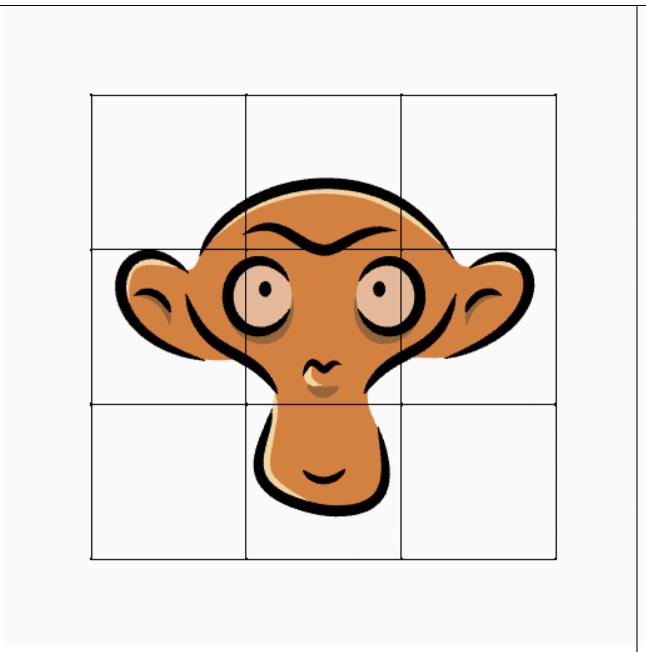
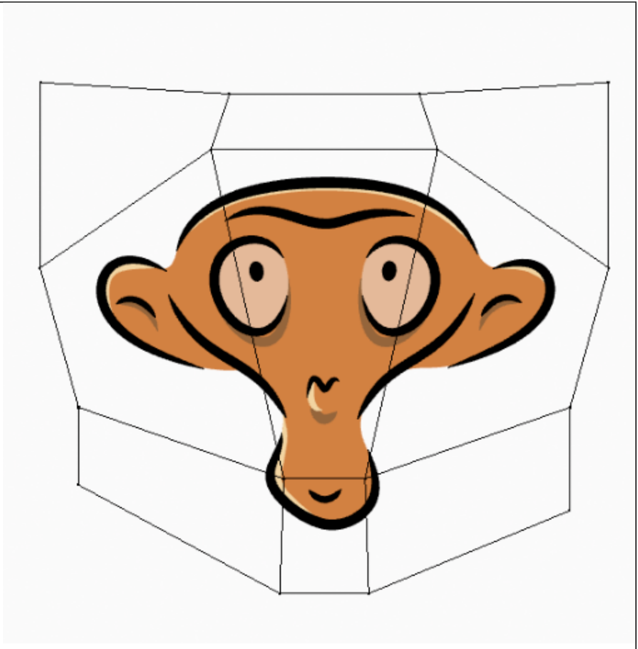
Strength A factor to control blending between original and deformed points positions.

Influence

See *Influence Filters*.

Example

Table 73: Lattice modifier example.

 Fig. 1402: Original model.	 Fig. 1403: After lattice edition.

Noise Modifier

The *Noise* Modifier changes the value of one or more stroke/points properties like: location, strength, thickness or UV texture position by adding varied values that make the line unstable and noisy.

Random values can be used for the noise factor for more vivid effects.

Options

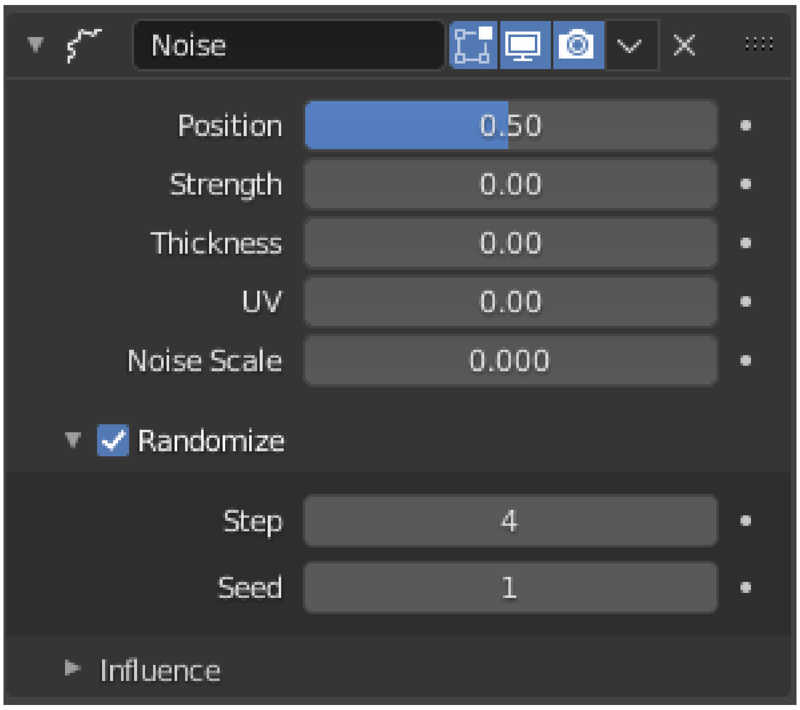


Fig. 1404: Noise Modifier.

Position Strength of the noise effect over the point location.

Strength Strength of the noise effect over the point strength (opacity).

Thickness Strength of the noise effect over the point thickness.

UV Strength of the noise effect over the point UV rotation.

Noise Scale Control the noise frequency scale.

Randomize

When enabled, the noise uses a random value over time.

Step Number of frames before using a new random value.

Seed *Seed* used by the pseudo-random number generator.

Influence

See *Influence Filters*.

Offset Modifier

The *Offset* Modifier changes the strokes location, rotation or scale starting from the object origin.

Options

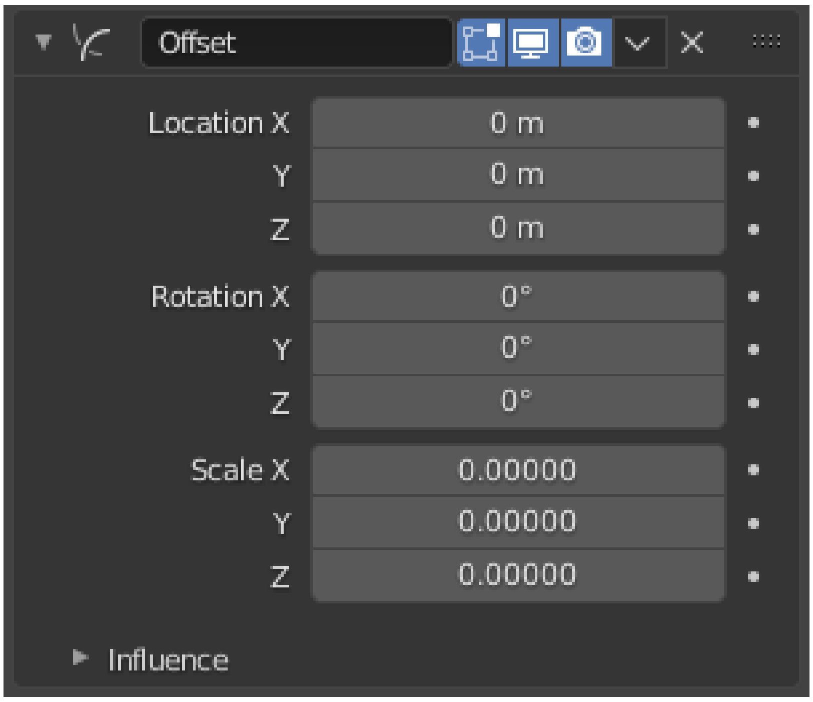


Fig. 1405: Offset Modifier.

Location X, Y, Z Sets strokes location offset from its object origin.

Rotation X, Y, Z Sets strokes rotation.

Scale X, Y, Z Sets strokes scale.

Influence

See *Influence Filters*.

Smooth Modifier

The *Smooth* Modifier changes the value of one or more stroke/points properties like: location, strength, thickness or UV texture position trying to maintain similar values that make the line fluid and smoother.

Options

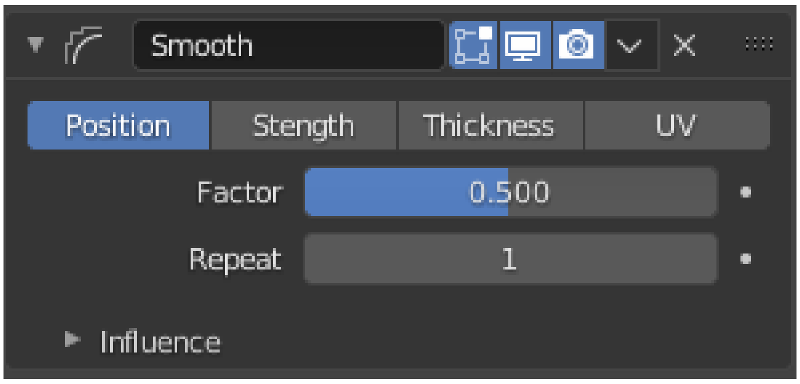


Fig. 1406: Smooth Modifier.

Affect Combination of stroke/points properties that will be affected by the smooth factor.

> **Position** Smooth affect the point location.
>
> **Strength** Smooth affect the point strength (opacity).
>
> **Thickness** Smooth affect the point thickness.
>
> **UV** Smooth affect the point UV rotation.

Factor Strength of the smooth effect.

Repeat The number of smoothing iterations, equivalent to executing the Smooth tool multiple times. High values can reduce the animation performance (FPS).

Influence

See *Influence Filters*.

Thickness Modifier

The *Thickness* Modifier change the stroke points thickness.

Options

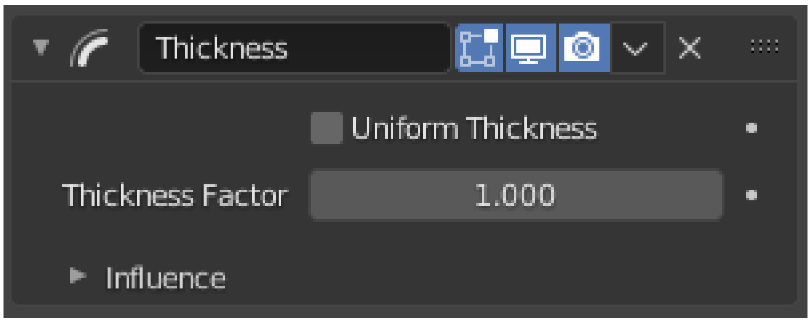


Fig. 1407: Thickness Modifier.

Uniform Thickness When enabled, makes the thickness equal for the entire strokes.

Thickness Absolute Thickness for the stroke points.

Thickness Factor Value to add or subtract to the actual points thickness.

Influence

See *Influence Filters*.

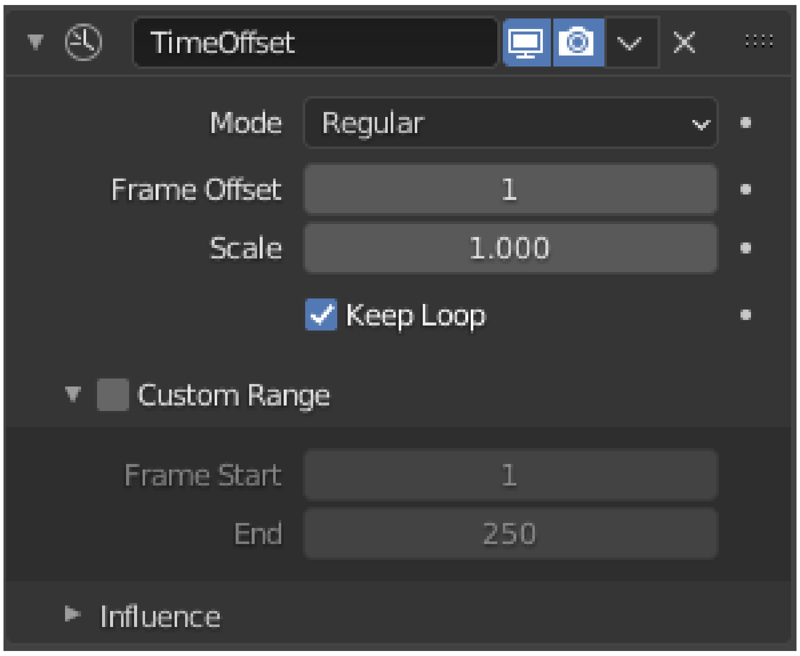
Time Offset Modifier

The *Time Offset* Modifier offsets the position of Grease Pencil keyframes.

For example can be used to start the same animation loop at different times and avoid an unappealing synchronization of the loops.

Or if you have different character poses in several keyframes, the Time Offset Modifier can be use to select which pose to show at a particular time in the animation. This is especially useful for cut-out animation.

Options


TimeOffset

Mode: Regular

Frame Offset: 1

Scale: 1.000

✓ Keep Loop

Custom Range

Frame Start: 1

End: 250

Influence

Fig. 1408: Time Offset Modifier.

Mode

Regular Offsets keyframes in default animation playback direction (left to right).

Reverse Offsets keyframes in inverse animation playback direction (right to left).

Fixed Frame Keep the selected frame fixed and do not change over time.

Frame Frame number to use.

Frame Offset Number of frames to offset the original keyframes.

Scale Evaluation time (in seconds).

Keep Loop Moves end frame to the animation start to keep animation in a loop.

Custom Range

When enabled, uses a custom range of frames.

Frame Start/End Sets the range start and end frames.

Influence

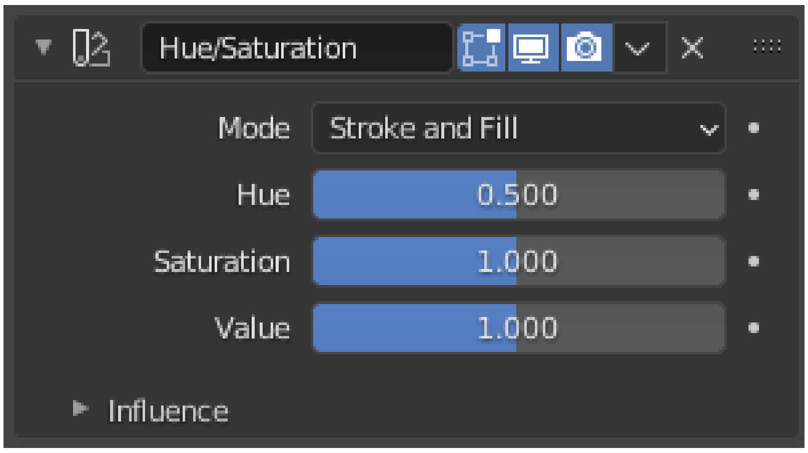
See *Influence Filters*.

Color

Hue/Saturation Modifier

The *Hue/Saturation* Modifier applies a color transformation to the object output color.

Options



Mode: Stroke and Fill
Hue: 0.500
Saturation: 1.000
Value: 1.000

Fig. 1409: Hue/Saturation Modifier.

Mode The color transformation will be applied on the stroke and/or the fill color.

Stroke and Fill, Stroke, Fill

Hue Specifies the hue rotation of the image. 360° are mapped to (0 to 1). The hue shifts of 0 (-180°) and 1 (+180°) have the same result.

Saturation A saturation of 0 removes hues from the image, resulting in a grayscale image. A shift greater than 1.0 increases saturation.

Value Value is the overall brightness of the image. De/Increasing values shift an image darker/lighter.

Influence

See *Influence Filters*.

Opacity Modifier

The *Opacity* Modifier change the opacity (alpha) value of the stroke points.

The alpha value in Grease Pencil is stored per-point. The modifier can alter these values to go from totally transparent points to totally opaque points.

Options

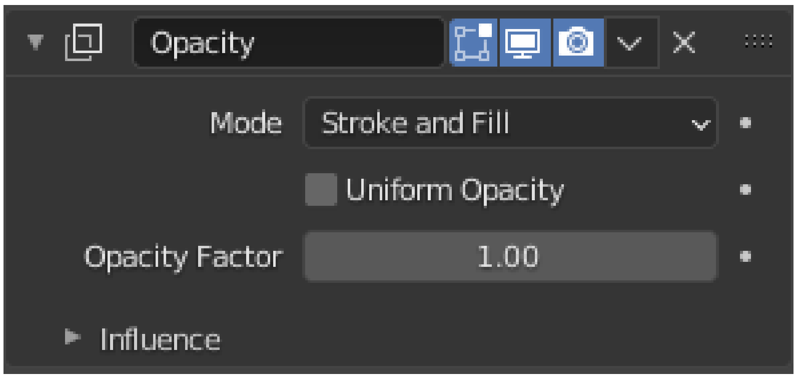


Fig. 1410: Opacity Modifier.

Mode The color transformation will be applied to the stroke/fill color or stroke Hardness. When Hardness is selected the opacity affects the stroke's transparency (alpha) from the center to the border.

Stroke and Fill, Stroke, Fill, or Hardness

Uniform Opacity When enabled, makes the opacity equal for the entire strokes.

Strength Absolute opacity for the stroke points.

Opacity Factor Controls the opacity value of the stroke points. A value of 1.0 respect the original alpha value of the points, a shift less than 1.0 make the points more transparent than originally, and a shift greater than 1.0 make the points more opaque than originally.

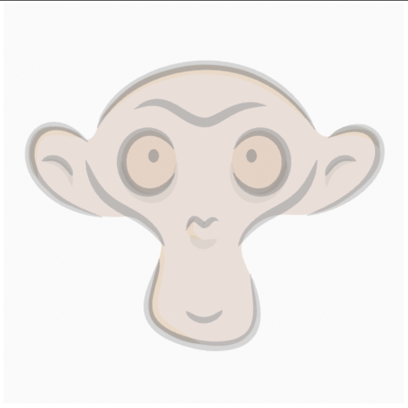
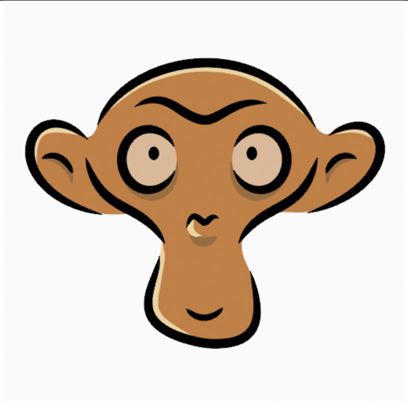
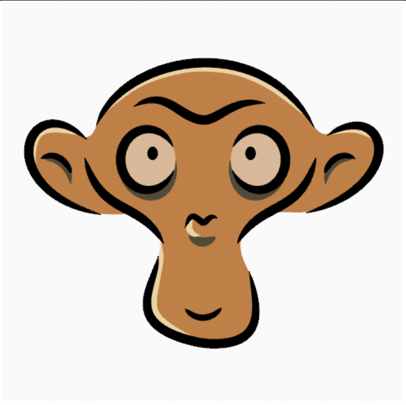
Sets value to 2.0 makes the points alpha fully opaque.

Influence

See *Influence Filters*.

Example

Table 74: Opacity Factor samples.

 Fig. 1411: Opacity Factor: 0.3.	 Fig. 1412: Opacity Factor: 1.0 (original alpha).	 Fig. 1413: Opacity Factor: 2.0 (fully opaque).

Texture Mapping Modifier

The *Texture Mapping* Modifier change the strokes texture UV position.

Options

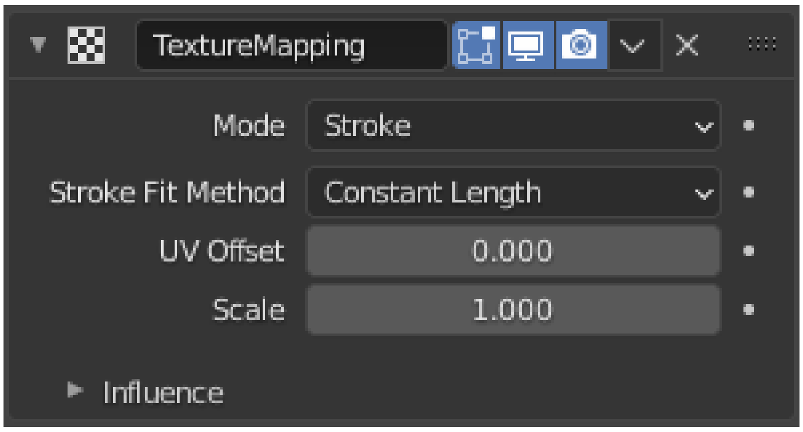

TextureMapping
Mode: Stroke
Stroke Fit Method: Constant Length
UV Offset: 0.000
Scale: 1.000
Influence

Fig. 1414: Texture Mapping.

Mode The texture transformation will be applied to the stroke/fill or stroke UVs.

Stroke

Stroke Fit Method Selects the texture fitting method.

Constant Length The texture keep a consistent length along the strokes.

Stroke Length The texture is normalized to fit the stroke length.

UV Offset Moves the texture along the strokes.

Scale Factor for the texture scale.

Fill

Fill Rotation Sets the texture angle.

Offset Moves the texture origin.

X, Y

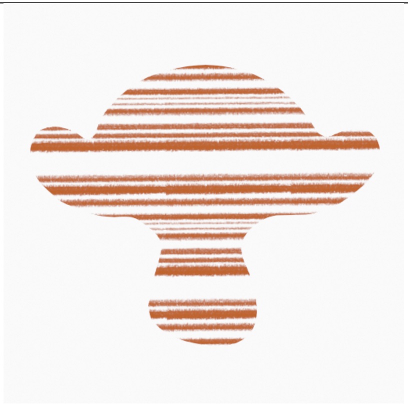
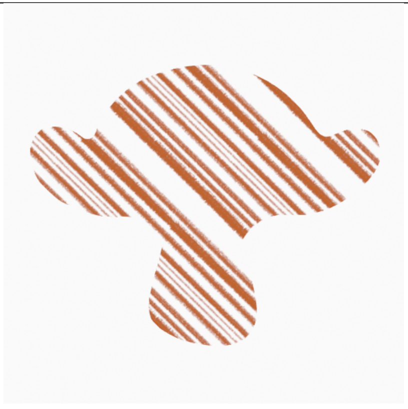
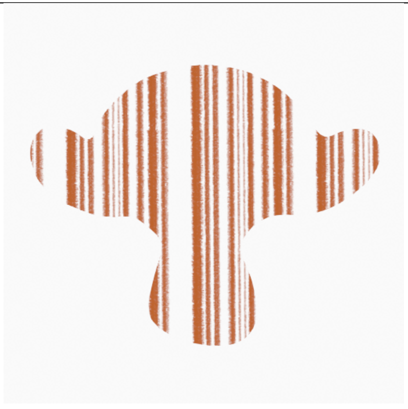
Scale Factor for the texture scale.

Influence

See *Influence Filters*.

Example

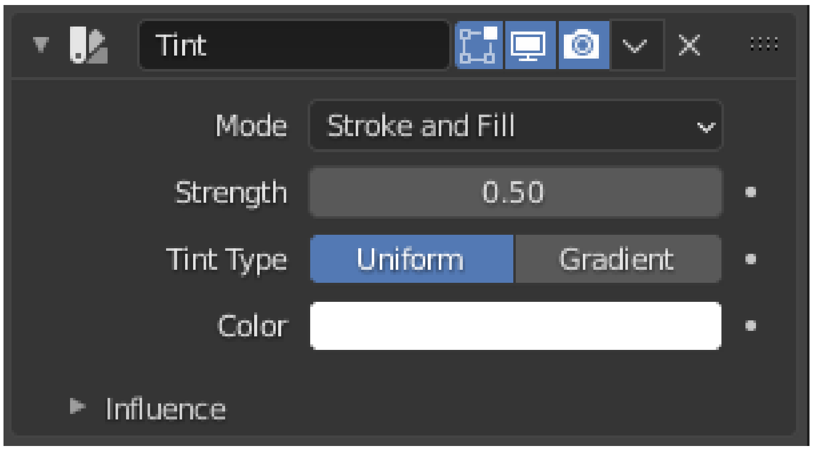
Table 75: Opacity Factor samples.

 Fig. 1415: Rotation: 0°.	 Fig. 1416: Rotation: 45°.	 Fig. 1417: Rotation: 90°.

Tint Modifier

The *Tint* Modifier colorize the original stroke or fill with a selected color.

Options



Mode: Stroke and Fill
Strength: 0.50
Tint Type: [✓ Uniform] Gradient
Color

Fig. 1418: Tint Modifier.

Mode The color transformation will be applied on the stroke and/or the fill color.

Stroke and Fill, Stroke, Fill

Strength Controls the amount for the color mixing.

A value of 0 respect the original strokes vertex color, a value of 1.0 totally replace the original color with the tint color.

A shift greater than 1.0 will make the points alpha less transparent than originally (2.0 is fully opaque).

Tint Type

Uniform

Color Defines the tint color for mixing with the original color.

Gradient

Color Ramp Defines the tint gradient color for mixing with the original vertex color. For controls see *Color Ramp Widget*.

Object A *Data ID* to select an object (usually an empty), which position and rotation will be used to define the center of the effect.

Radius Defines the maximum distance of the effect.

Influence

See *Influence Filters*.

Example

Table 76: Tint uniform color sample.

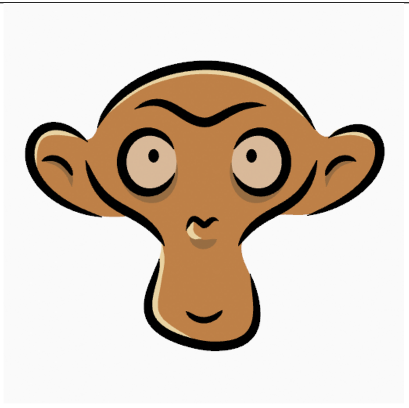
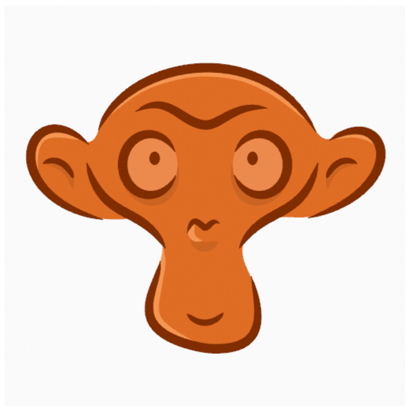
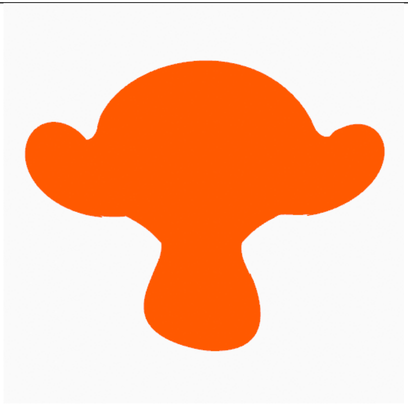
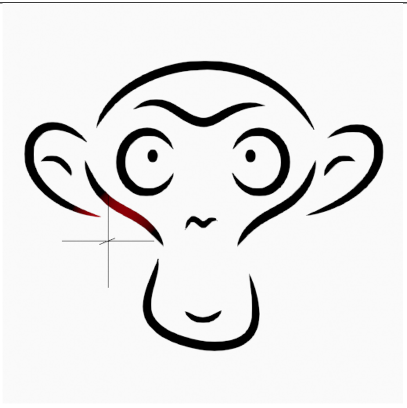
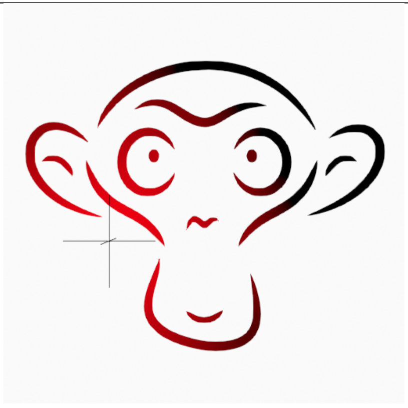
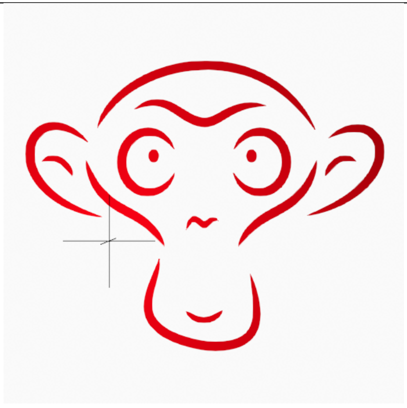
 Fig. 1419: Strength: 0 (original color).	 Fig. 1420: Strength: 0.5.	 Fig. 1421: Strength: 1.0 (fully tinted).

Table 77: Tint gradient color sample.

 Fig. 1422: Radius: 1, Strength: 1.	 Fig. 1423: Radius: 5, Strength: 1.	 Fig. 1424: Radius: 10, Strength: 1.

2.7.9 Visual Effects

Introduction

Reference

Panel *Properties → Visual Effects*

Grease Pencil has a special set of viewport real-time visual effects that can be apply to the object.

These effects treat the object as if it was just an image, for that reason they have effect on the whole object and cannot limit their influence on certain parts like layers, materials or vertex group as with modifiers. Also unlike modifiers, they can not be applied to the object.

Their main purpose is to have a quick way to apply visual effects on your drawings like blurring, pixelation, wave distortion, among others.

Note: Visual Effects best fit for quick viewport visualization. You can use it for final renders but if you want more precision with effects it is still recommended to use the *Compositor*.

Interface

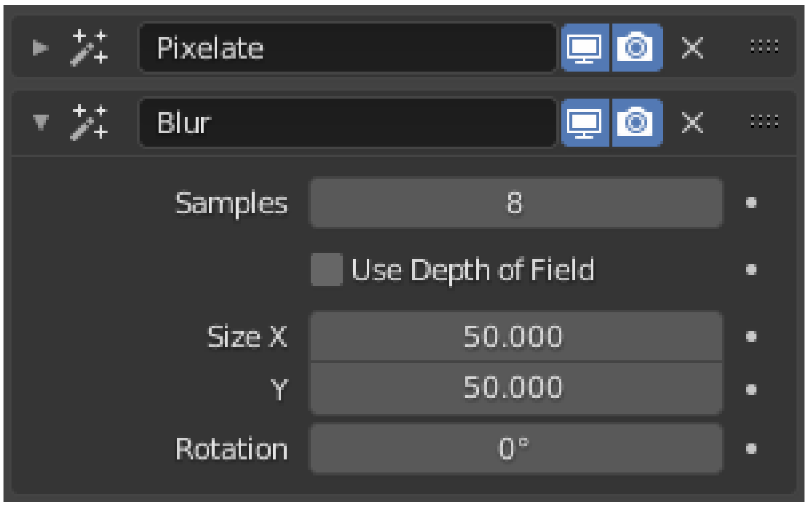
The visual effects panels and interface are similar to modifiers. Each effect shares the same basic interface components similar to modifiers for meshes.

See *Modifiers Interface* for more information.

Types

Blur Visual Effect

The *Blur* Visual Effect applies a Gaussian blur to the object.



Samples 8

Use Depth of Field

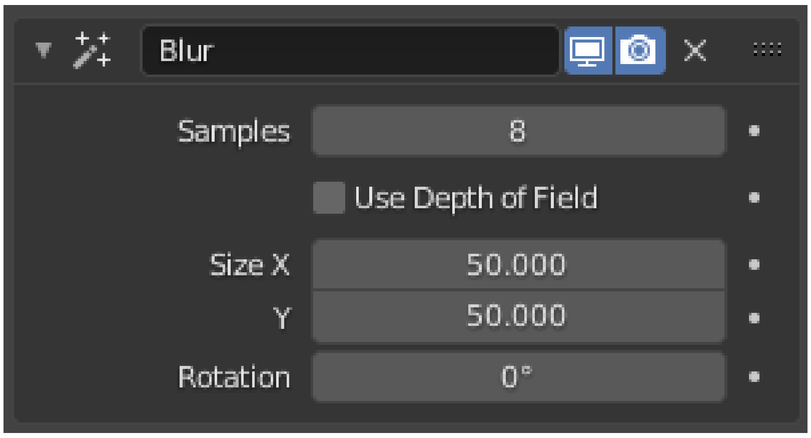
Size X 50.000

Y 50.000

Rotation 0°

Fig. 1425: Panel layout (Blur effect as an example).

Options



Samples 8

Use Depth of Field

Size X 50.000

Y 50.000

Rotation 0°

Fig. 1426: Blur Visual Effect.

Samples Number of blur samples (0 disabled the blur effect).

Use Depth of Field When enabled, the blur effect uses the focal plane distance of the actual camera to calculate the object blur. Only available in camera view.

Size Control the blur scale in pixels on the X and Y axis.

X, Y

Rotation Control the Rotation of the blur.

Example

Table 78: Blur Effect samples (Samples: 8).

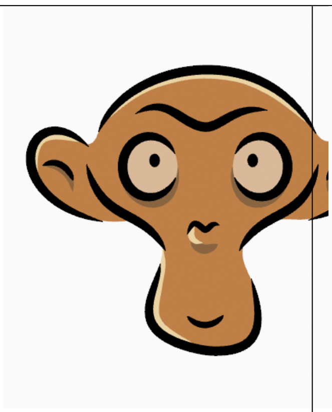
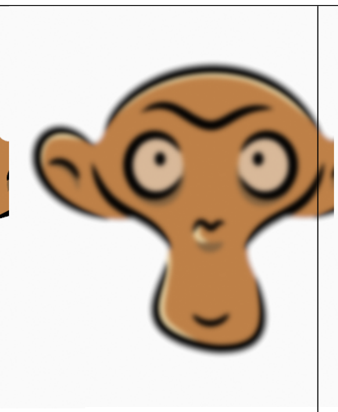
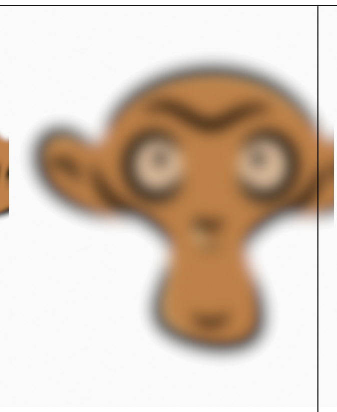
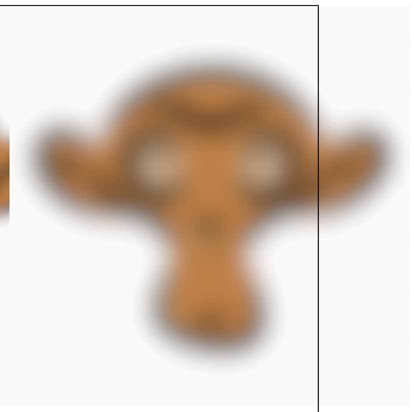
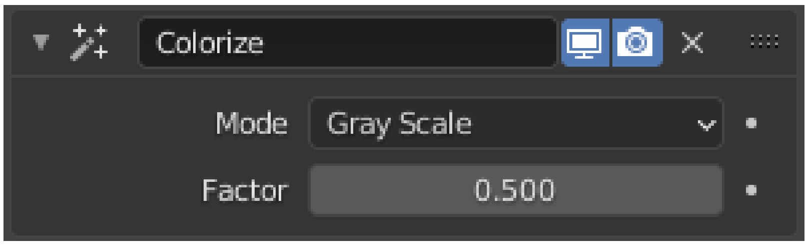
			

Fig. 1427: Original Model.	Fig. 1428: Factor: 10, 10.	Fig. 1429: Factor: 50, 50.	Fig. 1430: Factor: 100, 100.

Colorize Visual Effect

The *Colorize* Visual Effect applies different preset colorizing effects to the object.

Options



Mode Gray Scale
Factor 0.500

Fig. 1431: Blur Visual Effect.

Mode

Grayscale Converts to a grayscale image.

Sepia Converts to a sepia tone image.

Duotone Converts to a posterize image with high contrast and brightness.

Low Color Primary color.

High Color Secondary color.

Transparent Add color transparency.

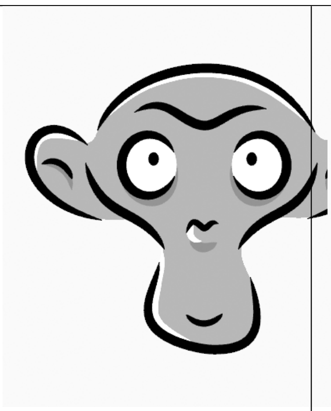
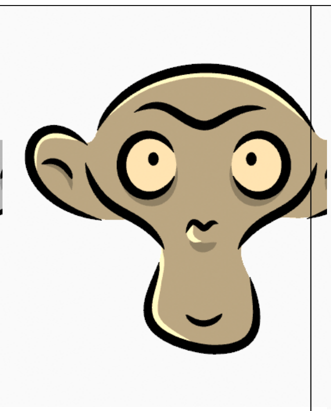
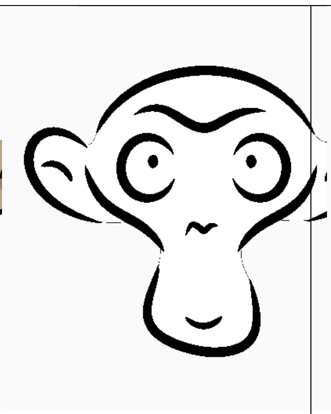
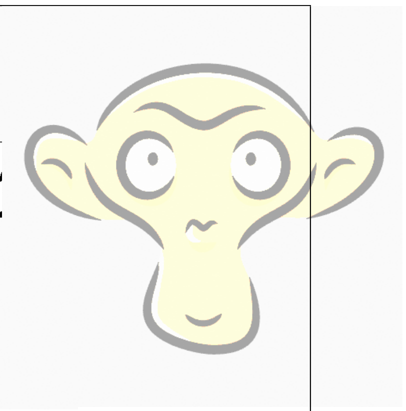
Custom Allows to define a tint custom color.

Color Sets the tint color.

Factor Control the mix value.

Example

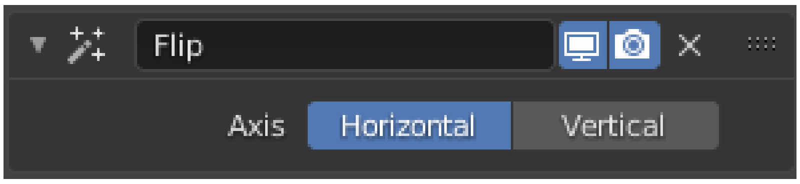
Table 79: Colorize Effect samples.

 Fig. 1432: Mode: Grayscale.	 Fig. 1433: Mode: Sepia.	 Fig. 1434: Mode: Duotone.	 Fig. 1435: Mode: Transparent.

Flip Visual Effect

The *Flip* Visual Effect shows the object flipped horizontally and/or vertically.

Options



Axis: Horizontal (selected) | Vertical

Fig. 1436: Flip Visual Effect.

Axis Which axis or axes to flip the object about.

> **Horizontal** When enabled, shows the object flipped horizontally.
>
> **Vertical** When enabled, shows the object flipped vertically.

Glow Visual Effect

The *Glow* Visual Effect add a glowing rim around the object.

Options

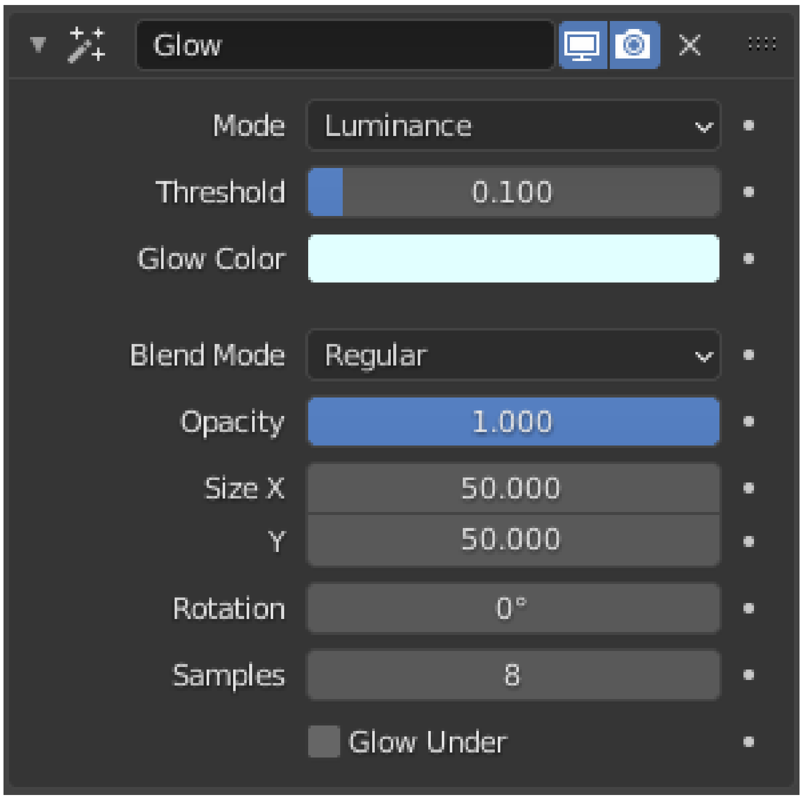

Glow
Mode: Luminance
Threshold: 0.100
Glow Color
Blend Mode: Regular
Opacity: 1.000
Size X: 50.000
Y: 50.000
Rotation: 0°
Samples: 8
Glow Under

Fig. 1437: Glow Visual Effect.

Mode Determines the mode for the glow effect.

> **Luminance** The glow light illuminates the entire object.
>
> > **Threshold** Limits the colors affected by the glow light. (A value of 1 means no colors affected.)
>
> **Color** The glow light only affect a single color.
>
> > **Select Color** Allows to select a single color to apply the glow light.

Glow Color Defines the glow color.

Blend Mode The mask blending operation to perform. See *Color Blend Modes*.

Opacity Control the Opacity of the glow over the object.

Size X, Y Control the glow scale in pixels on the X and Y axis.

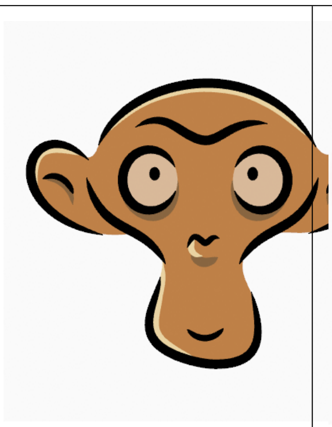
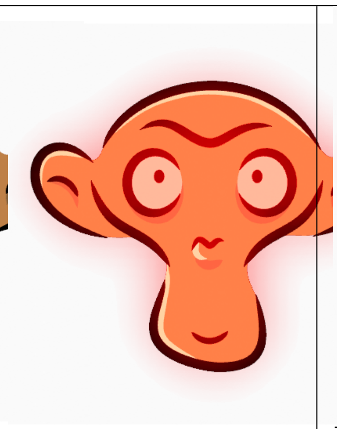
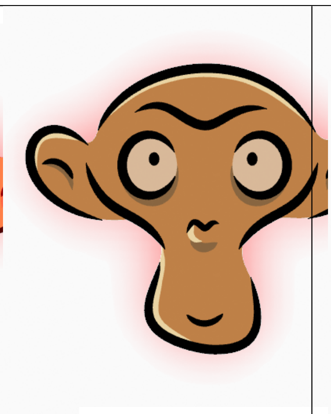
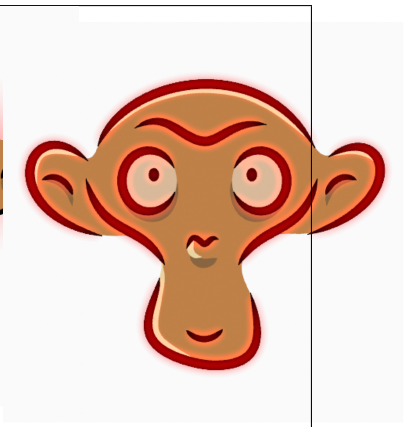
Rotation Control the Rotation of the glow.

Samples Number of Blur samples (0 disabled the blur effect).

Glow Under When enabled, glow only affects alpha areas.

Example

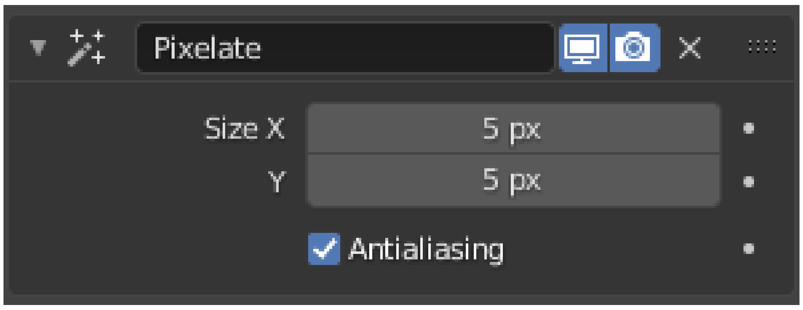
Table 80: Glow Effect samples.

 Fig. 1438: Original image.	 Fig. 1439: Mode: Luminance.	 Fig. 1440: Mode: Luminance (Glow Under).	 Fig. 1441: Mode: Color (Black lines).

Pixelate Visual Effect

The *Pixelate* Visual Effect shows the object as a pixelated image.

Options


Pixelate

Size X 5 px

Y 5 px

☑ Antialiasing

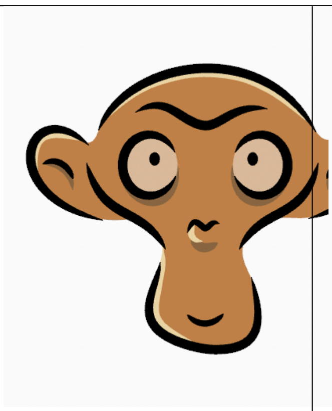
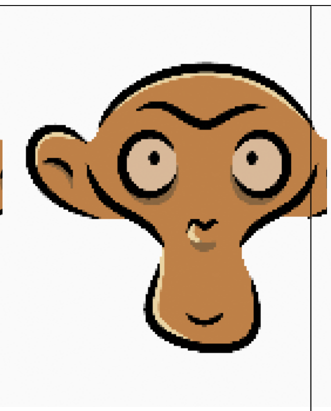
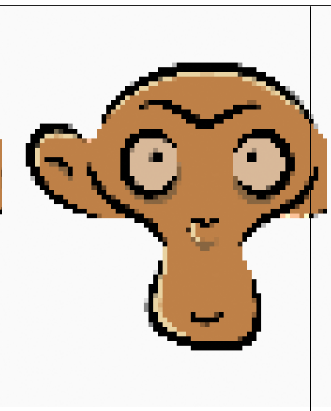
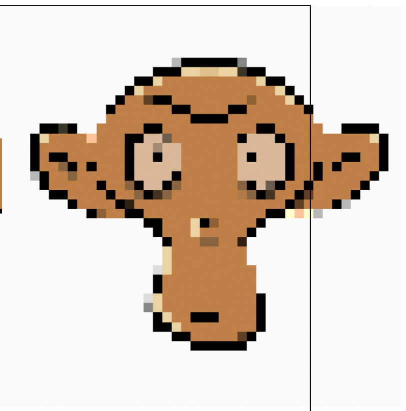
Fig. 1442: Pixelate Visual Effect.

Size X, Y Horizontal and vertical size of the final pixels to apply.

Anti-aliasing Applies an anti-aliasing effect to the resulting pixels.

Example

Table 81: Pixelate Effect samples.

 Fig. 1443: Original image.	 Fig. 1444: Size: 20 px.	 Fig. 1445: Size: 100 px.	 Fig. 1446: Size: 200 px.

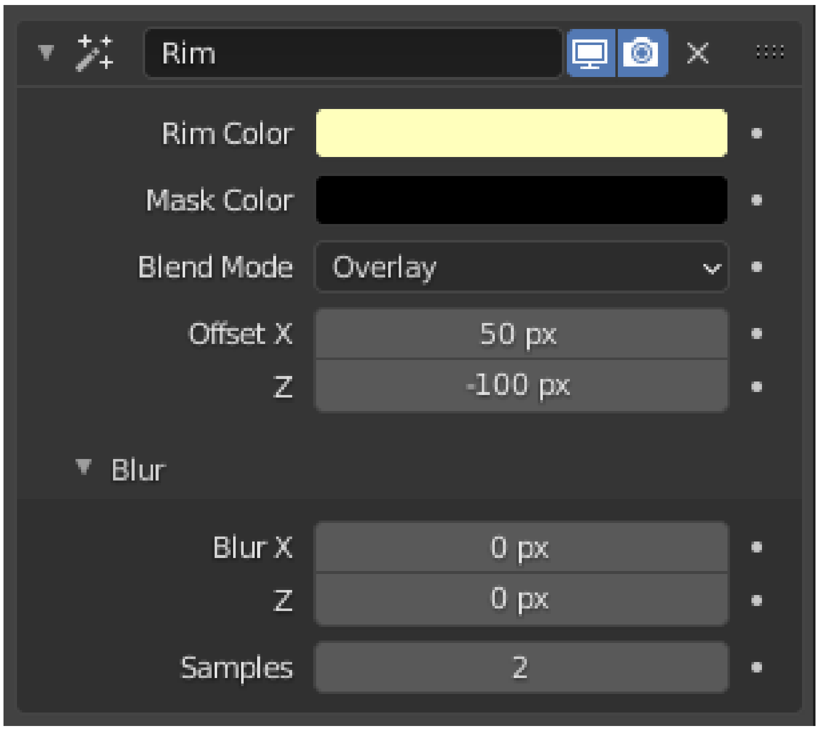
Rim Visual Effect

The *Rim* Visual Effect shows a simulated rim light on the object contour.

For simulating the rim light, a masked color silhouette of the object is displaced in horizontal and/or vertical direction.

Many blending modes can be applied to the resulting mask.

Options



Rim

Rim Color: 🟨
Mask Color: ⬛
Blend Mode: Overlay
Offset X: 50 px
Z: -100 px

Blur
Blur X: 0 px
Z: 0 px
Samples: 2

Fig. 1447: Rim Visual Effect.

Rim Color Defines the rim light color.

Mask Color Defines a color to keep unaltered.

Blend Mode The mask blending operation to perform. See *Color Blend Modes*.

Offset X, Y Control the color mask displacement in pixels on the X and Y axis.

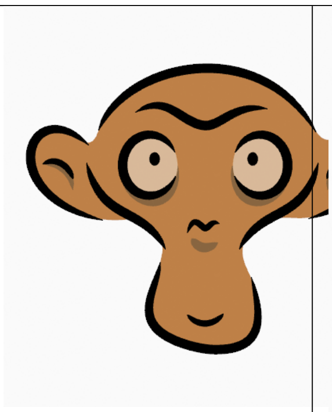
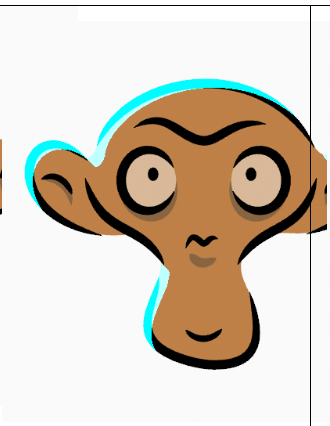
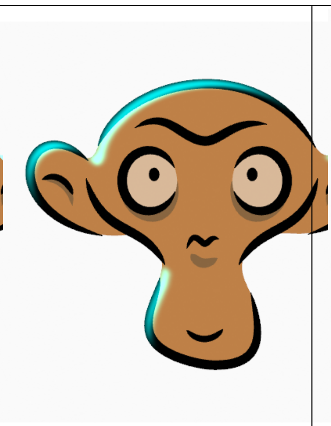
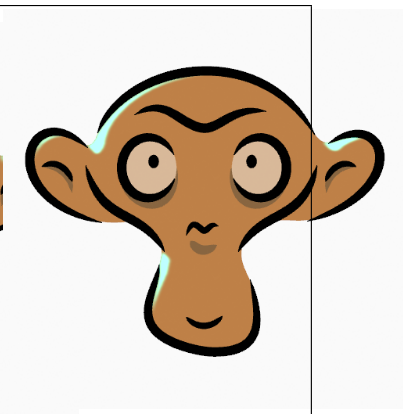
Blur

Blur X, Y Control the blur scale in pixels on the X and Y axis.

Samples Number of blur samples (0 disabled the blur effect).

Example

Table 82: Rim Effect samples (Mode: Add).

 Fig. 1448: Original image.	 Fig. 1449: No blur.	 Fig. 1450: Blur.	 Fig. 1451: Mask color: Black.

Shadow Visual Effect

The *Shadow* Visual Effect shows a simulated shadow casting by the object.

For simulating the shadow a color silhouette of the object is displaced in horizontal and/or vertical direction on the back of the object.

Options

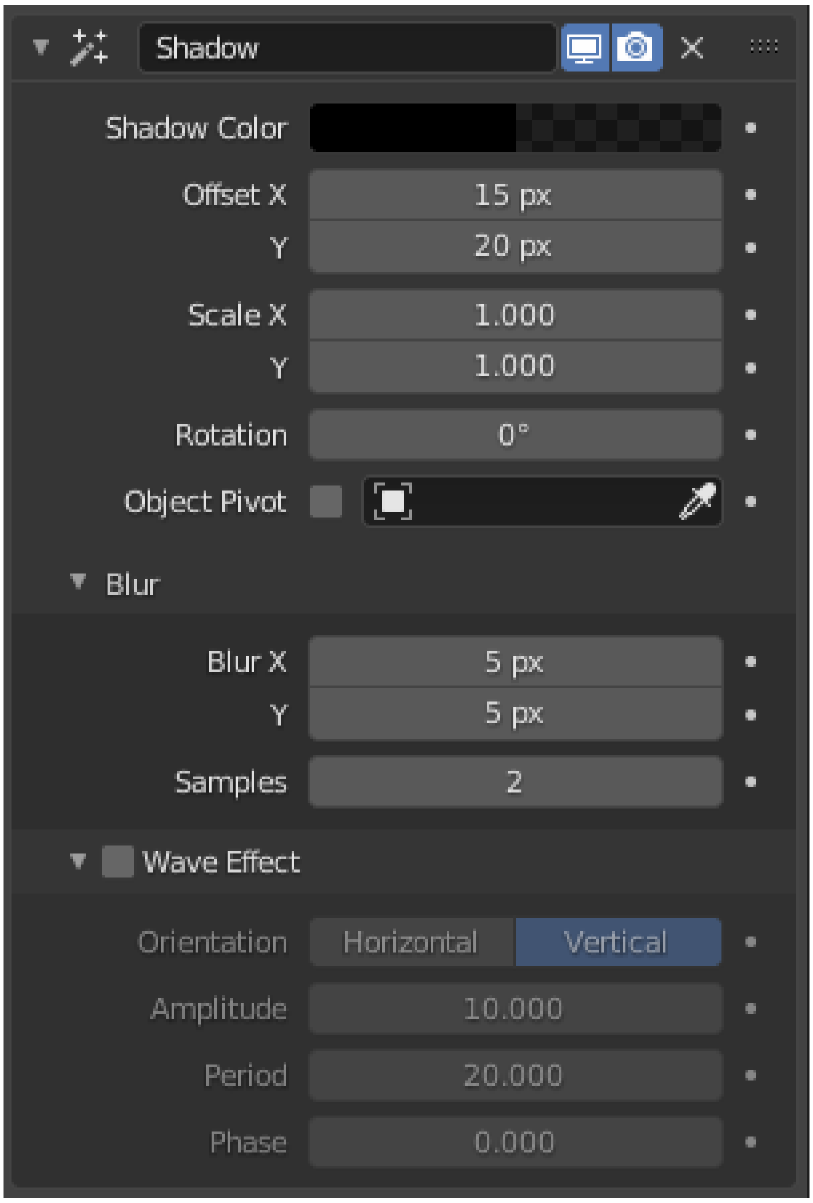

Shadow
Shadow Color
Offset X 15 px
Y 20 px
Scale X 1.000
Y 1.000
Rotation 0°
Object Pivot
Blur
Blur X 5 px
Y 5 px
Samples 2
Wave Effect
Orientation Horizontal [✓ Vertical]
Amplitude 10.000
Period 20.000
Phase 0.000

Fig. 1452: Shadow Visual Effect.

Shadow Color Defines the shadow color.

Offset X, Y Control the shadow displacement in pixels on the X and Y axis.

Scale X, Y Control the size of the shadow on the X and Y axis.

Rotation Sets the shadow rotation around the Grease Pencil object center or another object when *Use Object As Pivot* is enabled.

Object Pivot When enabled, an *Object* is used by the shadow as the center of rotation.

Blur

Blur X, Z Control the blur scale in pixels on the X and Z axis.

Samples Number of blur samples (0 disabled the blur effect).

Wave Effect

When enabled, apply a wave distortion to the shadow.

Orientation Sets horizontal or vertical direction for the waves.

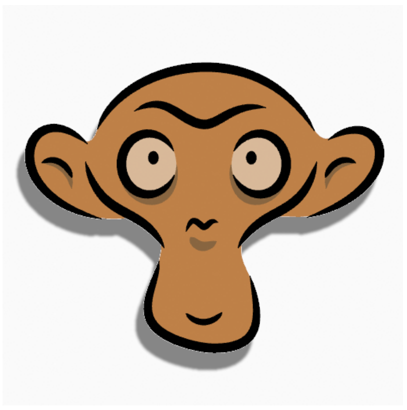
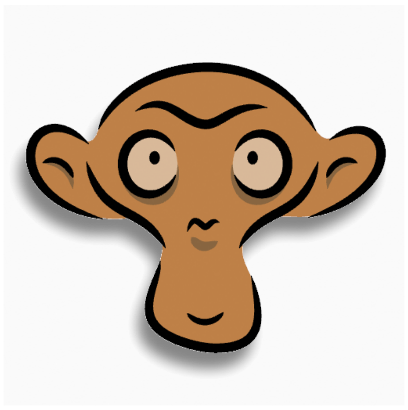
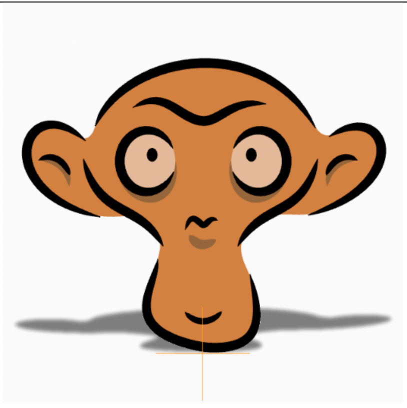
Amplitude Controls the strength and the depth of the wave.

Period Controls the wave period. The time it takes to complete one cycle.

Phase Shifts the wave pattern over the shadow.

Example

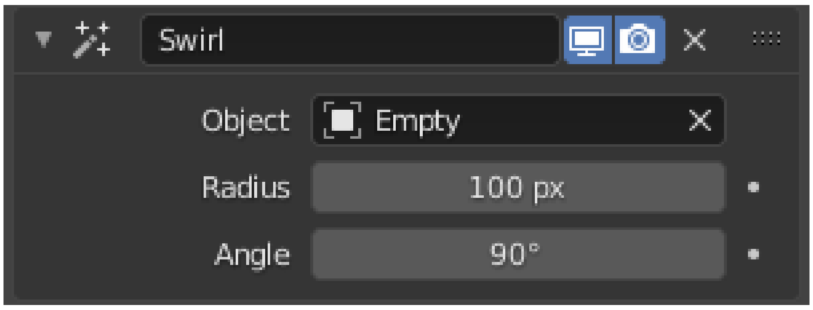
Table 83: Shadow Effect samples.

 Fig. 1453: Simple Shadow.	 Fig. 1454: Blurred Shadow.	 Fig. 1455: Stretched shadow with an empty as center of rotation.

Swirl Visual Effect

The *Swirl* Visual Effect applies a swirling pattern to the object. The effect use an object as the center of the swirl.

Options



Swirl

Object: Empty

Radius: 100 px

Angle: 90°

Fig. 1456: Shadow Visual Effect.

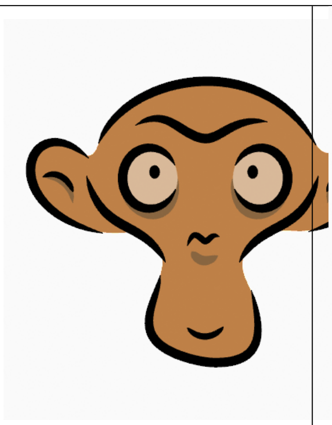
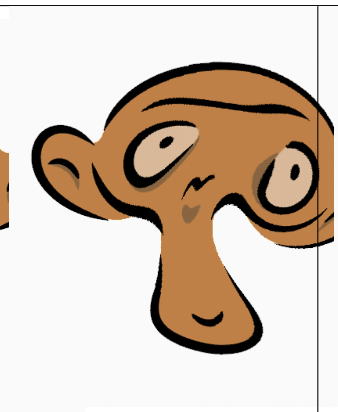
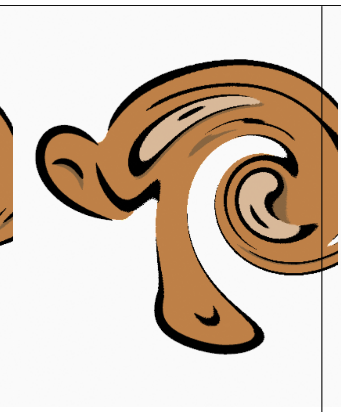
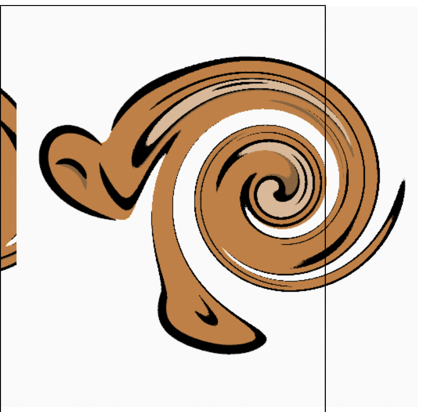
Object Sets the object to use as the center of the swirl.

Radius External radius size of the swirl.

Angle Rotation angle of the swirl. A value of 0 shows no swirl.

Example

Table 84: Swirl Effect samples (with a Radius of 100 px).

 Fig. 1457: Angle: 0°.	 Fig. 1458: Angle: 15°.	 Fig. 1459: Angle: 45°.	 Fig. 1460: Angle: 90°.

Wave Distortion Visual Effect

The *Wave Distortion* Visual applies a wavy effects to the object.

Options

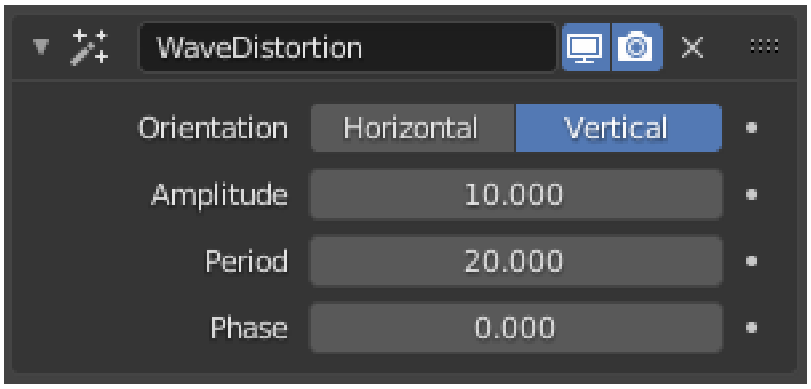


Fig. 1461: Wave Distortion Effect.

Orientation Sets horizontal or vertical direction for the waves.

Amplitude Controls the strength and the depth of the wave.

Period Controls the wave period. The time it takes to complete one cycle.

Phase Shifts the wave pattern over the Object.

Example

Table 85: Wave Distortion Effect samples.

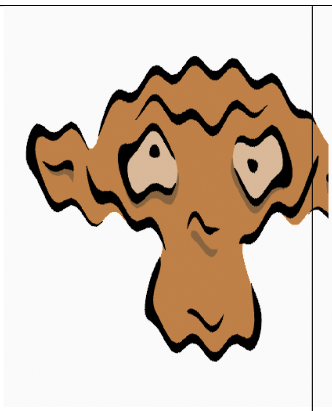
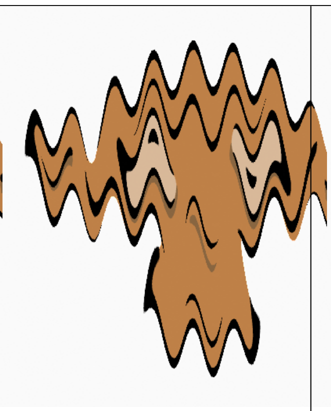
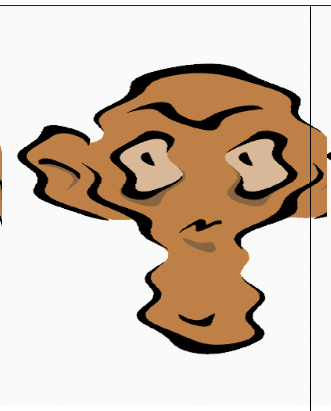
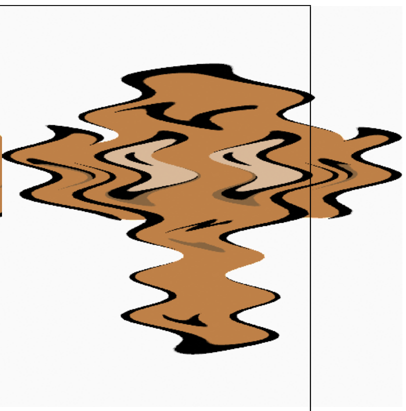
			

Fig. 1462: Amplitude: 10 (horizontal).	Fig. 1463: Amplitude: 30 (horizontal).	Fig. 1464: Amplitude: 10 (vertical).	Fig. 1465: Amplitude: 30 (vertical).

2.7.10 Materials

Introduction

Materials control the appearance of the Grease Pencil object. They define the base color and texture of the strokes and filled areas.

There is always only one active material in the list (the selected one). When you draw, the new strokes use the active material.

You can override the base material color using the tools in *Vertex Mode* or the Draw and Tint tool in Draw Mode.

The material always remains linked to the strokes, this means that any change in a material will change the look of already drawn strokes.

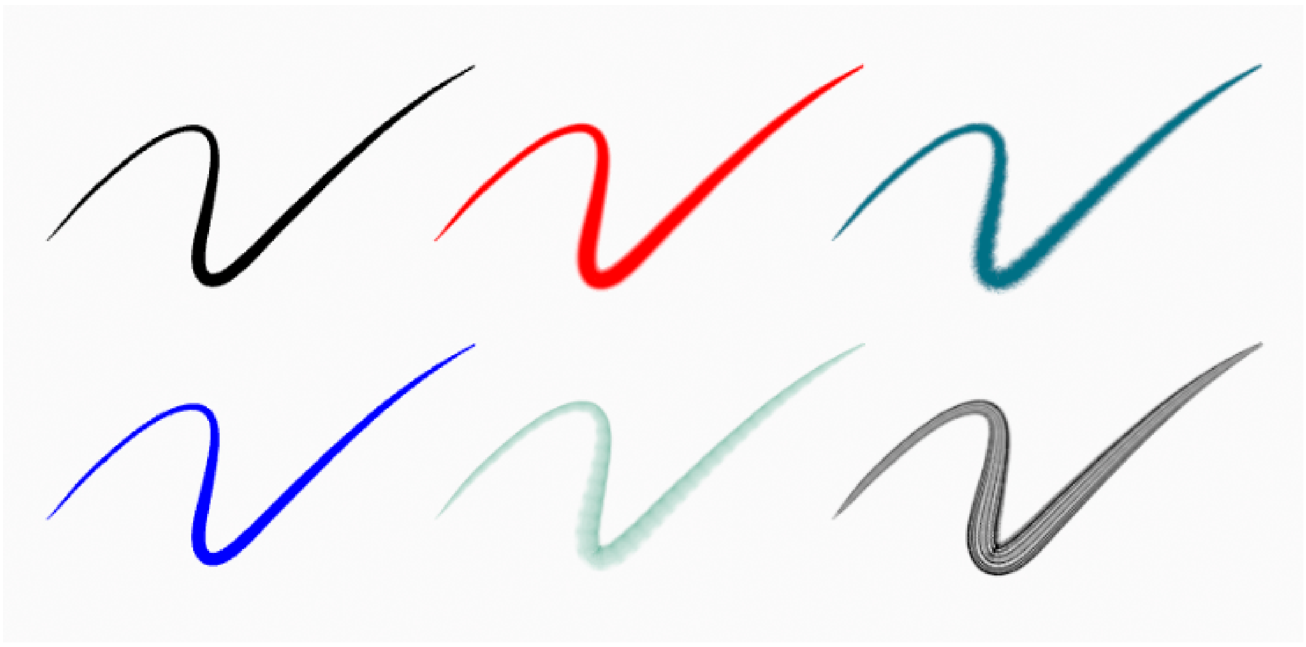


Fig. 1466: Same stroke linked to different materials.

Setting Up Materials

Reference

Mode Drawing Mode

Panel *Material → Material Slots*

Hotkey `U`

Grease Pencil materials can be created in the *Material properties* as any other materials in Blender. See *Material assignment* for more information.

The 3D Viewport can be set to Material Preview or Rendered shading, to interactively preview how the material looks in the scene.

Grease Pencil materials are data-blocks that can be *assigned* to one or more objects, and different materials can be assigned to different strokes.

In Grease Pencil the *brush* settings together with the material used will define the look and feel of the final strokes.

Materials slots in the *List view* also have some extra controls that help to work with materials while drawing or editing lines.

Common Settings

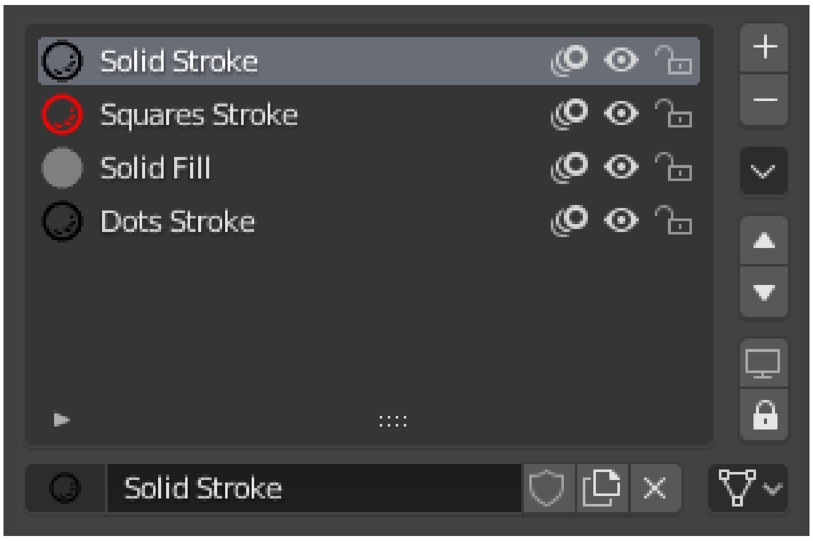


Fig. 1467: Grease Pencil material slots panel.

Next to the material name there are three icons buttons that control common properties of the material:

Lock (padlock icon) Toggle material from being editable.

Viewport/Render Visibility (eye icon) Toggle material visibility in the viewport and in render.

Onion Skinning (onion skin icon) Toggle the use of the material for *Onion Skinning*.

Specials

Show All Turns on the visibility of every material in the list.

Hide Others Turns off the visibility of every material in the list except the active one.

Lock All Locks edition of all the materials in the list.

Unlock All Unlocks edition of all the materials in the list.

Lock Unselected Locks all materials not used in the selected strokes.

Lock Unused Locks and hides all unused materials.

Remove Unused Slots Remove all unused materials.

Merge Similar Combines similar materials in the list and replace the strokes that use the one of the merged materials with the new one.

Convert Materials to Vertex Color Only keeps necessary materials and convert all materials base color to vertex color.

Extract Palette from Vertex Color Add all used vertex color to a new Color Palette. See *Color Palette*.

Lock & Visibility General Controls

Lock (padlock icon) Toggle whether the active material is the only one that can be edited.

Visibility (screen icon) Toggle whether the active material is the only one that can be edited and is visible.

Grease Pencil Shader

Grease Pencil materials use a special *shader* that define the appearance of the surface of the stroke and fill.

Grease Pencil Shader

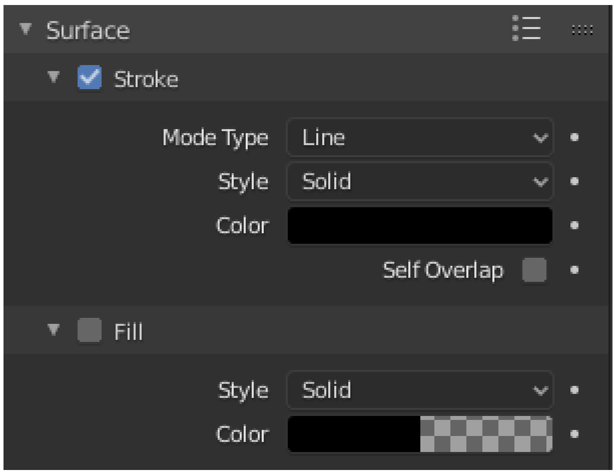

Surface
Stroke ✓
Mode Type: Line
Style: Solid
Color
Self Overlap
Fill
Style: Solid
Color

Fig. 1468: Shader panel with only Stroke component activated.

The Grease Pencil shader creates a material that can work with strokes and/or filled areas of a Grease Pencil object.

Stroke and fill components has it own section panel and they can be enabled with a checkbox on the panel header.

Stroke only has effect on the lines and *Fill* only on the areas determined by closed lines (by connecting the lines start and end points).

Note: The shader is not yet a BSDF capable shader and can only be setting up on the Material Properties panel (it is not a shader node).

Properties

Stroke

When enabled, the shader use the stroke component. The *Stroke* component controls how to render the edit lines.

Mode Type Defines how to display or distribute the output material over the stroke.

- **Line** Connects every points in the strokes showing a continuous line.
- **Dots** Use a disk shape at each point in the stroke. The dots are not connected.
- **Squares** Use a square shape at each point in the stroke. The squares are not connected.

Style The type of the material.

- **Solid** Use a solid color.
  - **Base Color** The base color of the stroke.
- **Texture** Use an image texture.
  - **Base Color** The base color of the stroke.
  - **Image** The image data-block used as an image source.
  - **Blend** Texture and Base Color mixing amount.
  - **UV Factor** The image size along the stroke.

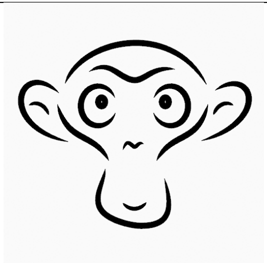
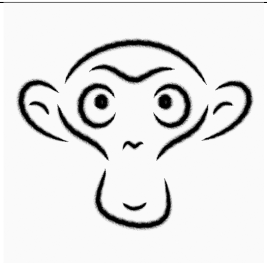
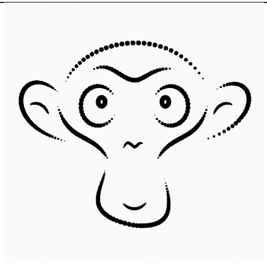
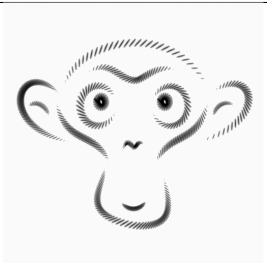
Holdout Removes the color from strokes underneath the current by using it as a mask.

Alignment Defines how to align the *Dots* and *Squares* along the drawing path and with the object's rotation.

- **Path** Aligns to the drawing path and the object's rotation.
- **Object** Aligns to the object's rotation; ignoring the drawing path.
- **Fixed** Aligns to the screen space; ignoring the drawing path and the object's rotation.

Self Overlap Disables stencil and overlap self-intersections with alpha materials.

Table 86: Samples of different material strokes mode types and styles.

 Fig. 1469: Mode Type: Line, Style: Solid.	 Fig. 1470: Mode Type: Line, Style: Texture.	 Fig. 1471: Mode Type: Dot, Style: Solid.	 Fig. 1472: Mode Type: Dot, Style: Texture.

Fill

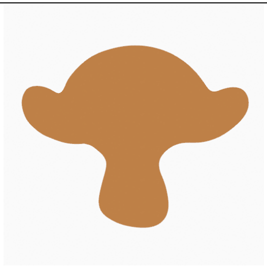
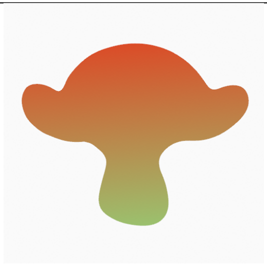
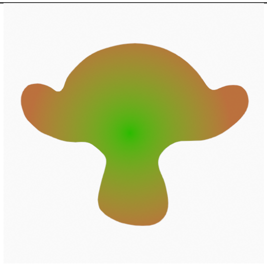
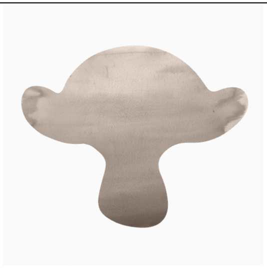
When enabled, the shader use the fill component. The *Fill* component control how to render the filled areas determined by closed edit lines.

Style The type of material.

- **Solid** Use solid color.
  - **Base Color** The base color of the fill.
- **Gradient** Use a color gradient.
  - Gradient Type
    - **Linear** Mix the colors along a single axis.
    - **Radial** Mix the colors radiating from a center point.
  - **Base Color** The primary color.
  - **Secondary Color** The secondary color.
  - **Blend** Base Color and Secondary Color mixing amount.
  - **Flip Colors** Flips the gradient, inverting the Base Color and Secondary Color.
  - **Location** Shifts the gradient position.
    - X, Y
  - **Rotation** Rotates the gradient.
  - **Scale** Scales the gradient.
    - X, Y
- **Texture** Use an image texture.
  - **Base Color** The base color of the fill.
  - **Image** The image data-block used as an image source.
  - **Blend** Texture and Base Color mixing amount.
  - **Location** Shifts the image position.
    - X, Y
  - **Rotation** Rotates the image.
  - **Scale** Scales the image.
    - X, Y
  - **Clip Image** When enabled, show one image instance only (do not repeat).

Holdout Removes the color from strokes underneath the current by using it as a mask.

Table 87: Samples of different material fill styles.

 Fig. 1473: Style: Solid.	 Fig. 1474: Style: Gradient (Linear).	 Fig. 1475: Style: Gradient (Radial).	 Fig. 1476: Style: Texture.

2.7.11 Animation

Introduction

Animating with Grease Pencil

The main goal of Grease Pencil is to offer a 2D animation tool full immersed in a 3D environment.

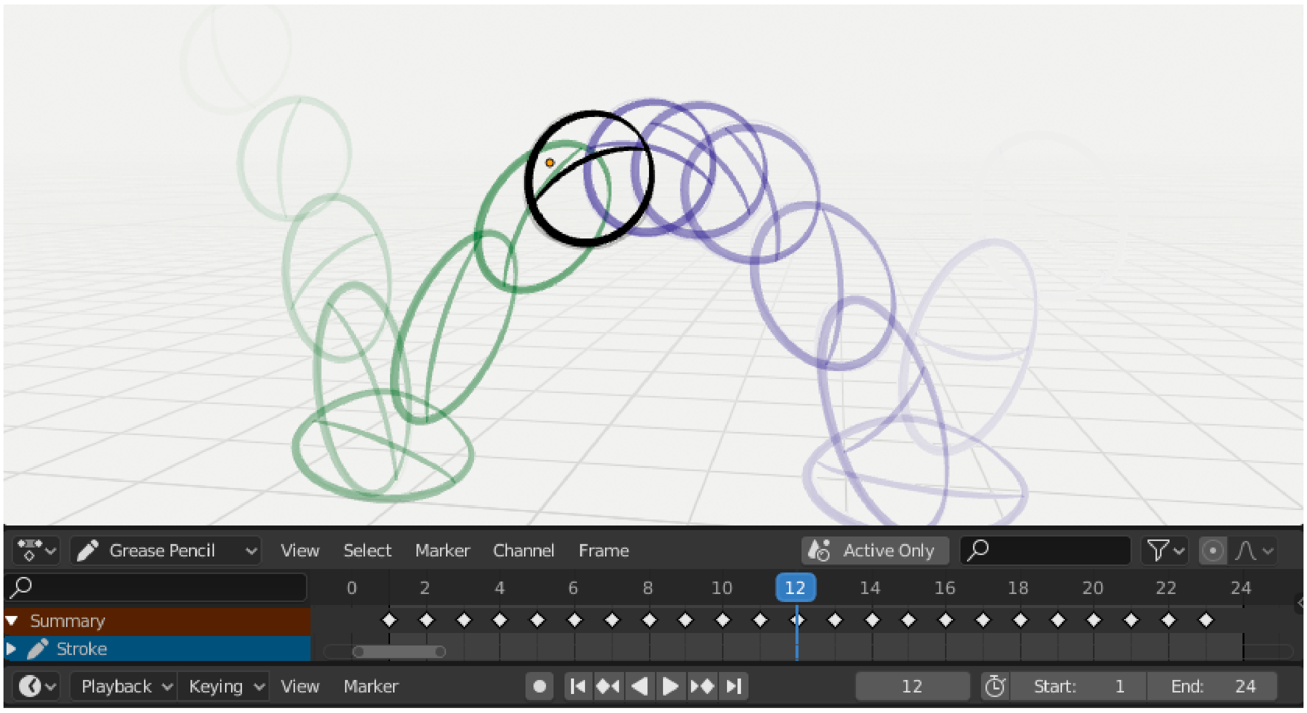


Fig. 1477: Sample animation showing Grease Pencil object keyframes in the Dope Sheet with onion skinning enabled.

In Blender, Grease Pencil objects can be animated in many ways:

Moving as a whole object Changing their position, orientation or size in time;

Drawing frame by frame Drawing one frame at a time (traditional animation).

Deforming them Animating their points;

Inherited animation Causing the object to move based on the movement of another object (e.g. its parent, hook, armature, etc.). Useful for cut-out animation for example.

For a complete overview of animation in Blender please refer to the *Animation & Rigging* chapter.

2D Traditional Animation

Keyframes

Traditional animation in Grease Pencil is achieved with the use of *keyframes* that hold the strokes information at a particular frame.

Every time you create a stroke in Grease Pencil object Draw Mode a new keyframe is added at the current frame on the active channel.

Note: The channels in the Dope Sheet correspond to the active 2D layer of the Grease Pencil object.

Grease Pencil has its own mode in the Dope Sheet to work with keyframes. See Grease Pencil mode in the *Dope Sheet* section for more information.

There are also several tools on the Stroke menu to work with keyframes and strokes. See *Animation tools* for more information.

Onion Skinning

One key element in traditional animation is the use of onion skinning. Grease Pencil offer a lot of flexibility and options for this tool. See *Onion Skinning* for more information.

Animation Options

Draw Mode

In Draw Mode there are two option related to the animation workflow that you can use.

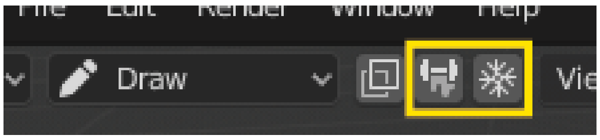


Fig. 1478: General drawing/animation options.

Add Weight Data When enabled, new strokes weight data is added according to the current vertex group and weights. If there is no vertex group selected, no weight data is added.

This is useful for example in cut-out animation for adding new drawing on the same vertex group without the need to creating it afterwards.

See *Weight Paint Mode* for more information.

Additive Drawing When creating new frames, the strokes from the previous/active frame are include as a basis for the new one.

Edit Mode

In Edit Mode there are two option related to the animation workflow that you can use.

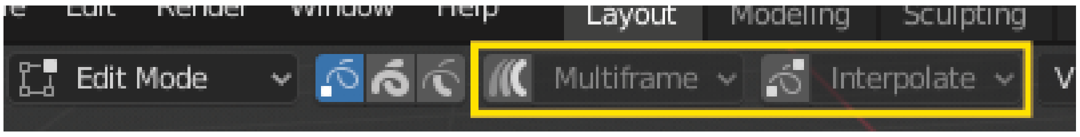


Fig. 1479: Multiframe edition.

Multiframe Sometimes you may need to modify several frames at the same time with edit tools, for example to repositioning drawings in an animation.

You can activate multiframe edition with the Multiframe button next to the modes selector (faded lines icon). See *Multiframe* for more information.

Interpolate When you are animating simple shapes with strokes that have the same amount of points, you can use the Interpolate tool to automatically add new breakdown keyframes. See *Interpolation* for more information.

Examples

Traditional Animation

This example shows you how to animate a bouncing ball with a traditional 2D animation technique and Grease Pencil.

First, go to menu *File* → *New* → *2D Animation* to start with a new 2D animation template. The template is ready to quick start your animation with a Grease Pencil object already created, onion skinning activated and in camera view.

1. Set the range of the animation in the Timeline from 1 to 24.
2. In the 3D Viewport draw a ball on the upper left corner with the Draw Tool (extreme).
3. Move to frame 12 and draw a squashed ball in the bottom center (breakdown).
4. Move to frame 24 and draw a ball in the top right corner of the 3D Viewport (extreme).
5. Keep drawing all the inbetweens frames you want using the onion skinning ghost as a reference.

To test the animation, press `Spacebar` to play.

Interpolation

Interpolate Panel

When you are animating simple shapes you can use the interpolate tools to automatically add new breakdown keyframes.

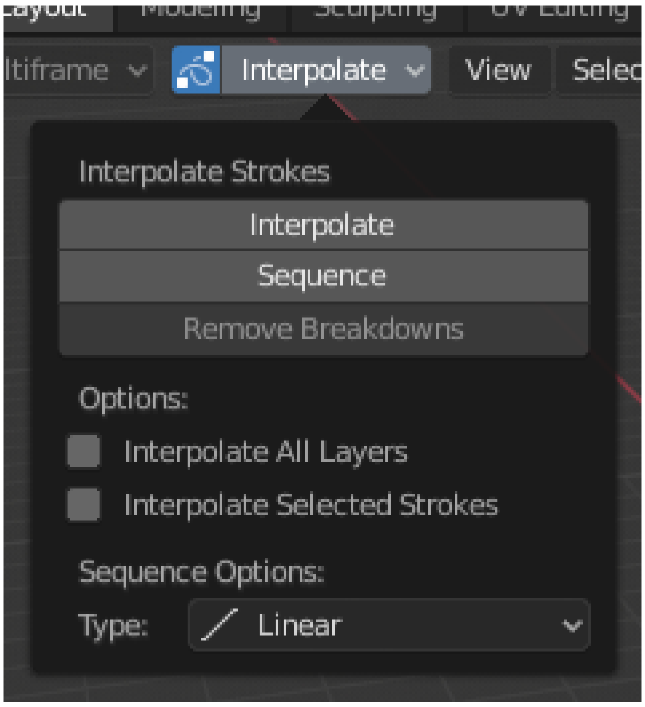


Fig. 1480: Interpolate panel.

Interpolate `Ctrl-E` Interpolates strokes between the previous and next keyframe by adding a *single* keyframe. When you are on a frame between two keyframes and click the Interpolate button a new breakdown keyframe will be added. This way you define the final interpolation for the new stroke.

Sequence `Shift-Ctrl-E` Interpolate strokes between the previous and next keyframe by adding *multiple* keyframes. When you are on a frame between two keyframes and click the sequence button a breakdown keyframe will be added on every frame between the previous and next keyframe.

Step The number of frames between generated interpolated frames.

Type Interpolation method to use for the sequence.

Remove Breakdowns Removes the breakdowns generated by the Interpolate tool.

Interpolate All Layers When enabled, interpolated keyframes will be created on all layers, not only the active one.

Interpolate Selected Strokes When enabled, only selected strokes will be interpolated.

Selection Mask When enabled the interpolation only affect selected strokes.

Note: The *Interpolate* and *Sequence* tools work better when the strokes in the previous and next keyframes have the same amount of points. For example when there are duplicated strokes on different keyframes only with different location, rotation or scale.

Animation Tools

Insert Blank Keyframe

Reference

Mode Draw Mode, Edit Mode, Sculpt Mode

Menu *Stroke → Animation → Insert Blank Keyframe (Active Layer) Stroke → Animation → Insert Blank Keyframe (All Layers)*

Hotkey `Shift-I`

Active Layer Add a new blank keyframe to the active layer at the current frame. If there is already a keyframe at the current frame, a new blank keyframe will be added on the next frame.

All Layers When enabled, Blank keyframe will be created on all layers, not only the active one.

Duplicate Active Keyframe

Reference

Mode Draw Mode, Edit Mode, Sculpt Mode

Menu *Stroke → Animation → Duplicate Active Keyframe (Active Layer) Stroke → Animation → Duplicate Active Keyframe (All Layers)*

Duplicates the strokes on the last keyframe by copying them to the current frame.

Mode

Active Duplicate only the active layer.

All Duplicate all the layers.

Delete Active Keyframe

Reference

Mode Draw Mode, Edit Mode, Sculpt Mode

Menu *Stroke → Animation → Delete Active Keyframe (Active Layer) Stroke → Animation → Delete Active Keyframes (All Layers)*

Hotkey `Shift-X`

Deletes the last keyframe in the Dope Sheet or the current keyframe if you are on one.

Interpolation

Interpolate

Reference

Mode Draw Mode, Edit Mode, Sculpt Mode

Menu *Stroke → Animation → Interpolate → Interpolate*

Hotkey `Ctrl-E`

Interpolates strokes between the previous and next keyframe by adding a *single* keyframe. When you are on a frame between two keyframes and press `Ctrl-Alt-E` a new breakdown keyframe will be added. This way you define the final interpolation for the new stroke.

Sequence

Reference

Mode Draw Mode, Edit Mode, Sculpt Mode

Menu *Stroke → Animation → Interpolate → Sequence*

Hotkey `Shift-Ctrl-E`

Interpolate strokes between the previous and next keyframe by adding *multiple* keyframes. A breakdown keyframe will be added on every frame between the previous and next keyframe.

Note: The *Interpolate* and *Sequence* tools work better when the strokes in the previous and next keyframes have the same amount of points. For example when there are duplicated strokes on different keyframes only with different location, rotation or scale.

Bake Mesh to Grease Pencil

Reference

Editor 3D View

Mode Object and Pose Modes

Menu *Object → Animation → Bake Mesh to Grease Pencil…*

Converts each frame of a mesh animation within a selected frame range to a Grease Pencil object keyframed strokes. The *Bake Action* tool computes the final animation of the selected objects with all those modifiers, drivers, and constraints applied, and keyframes the result.

Start Frame, End Frame Start/End frame for the baking process.

Frame Step Frame steps for the baking process

Thickness Strokes thickness.

Threshold Angle Threshold value that determine the strokes end.

Stroke Offset Sets offset to separate strokes from filled strokes.

Only Seam Edges Convert only edges marked as seam.

Export Faces Convert faces as filled strokes.

Target Object Select the target Grease Pencil object for the baked animation or a new one if there is nothing yet.

Target Frame Target destination frame for the baked animation.

Reproject Type Sets the reprojection type to use for the converted strokes.

2.7.12 Modes

Draw Mode

Introduction

Draw Mode is the mode in Grease Pencil that allows you to draw in the 3D Viewport. This mode is actually the only one in which new strokes can be created.

Already made strokes can not be selected in Draw Mode, for editing strokes you must use the *Edit Mode* or *Sculpt Mode*.

Draw Mode

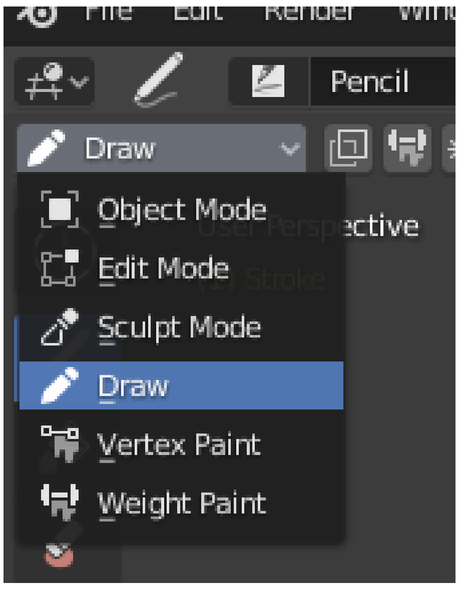


Fig. 1481: 3D Viewport Mode selector: Draw Mode.

Draw Mode is selected with the *Mode* menu in the 3D Viewport header. Once Draw Mode is activated, the Toolbar of the 3D Viewport will change to Draw Mode specific panels. Also a circle with the same color as the active material will appear and follow the location of the cursor in the 3D Viewport.

To create new strokes you have to select one of the drawing tools in the Toolbar. The most common one is the *Draw tool* for free-hand drawings but there are many other tools for drawing, filling areas and erasing strokes. There are also some tools to create primitives shapes like lines, arcs, curves, boxes and circles.

See *Toolbar* for more details.

Strokes Location and Orientation Controls

Drawing in a 3D space is not the same as drawing on a flat canvas. When drawing with Grease Pencil you have to define the location and orientation of the new strokes in the 3D space.

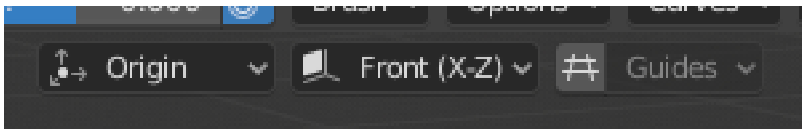


Fig. 1482: 3D Viewport header Controls for strokes.

Stroke Placement

The Stroke Placement selector defines the new strokes location in 3D space.

See *Stroke Placement* for more information.

Drawing Planes

The Drawing Planes selector defines the plane (orientation) to which the new strokes will be restricted.

See *Drawing Planes* for more information.

Guides

Different Guides types can be activated to assist you when drawing new strokes.

See *Guides* for more information.

Drawing Options

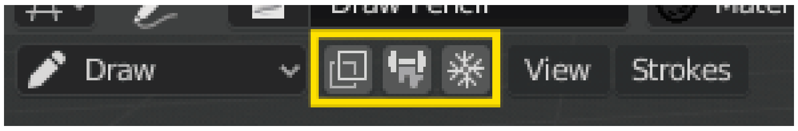


Fig. 1483: General drawing options.

Draw on Back When enabled, new strokes are drawn below of all strokes in the layer. For example when you want to paint with a fill material below line strokes on a character and they are on the same layer.

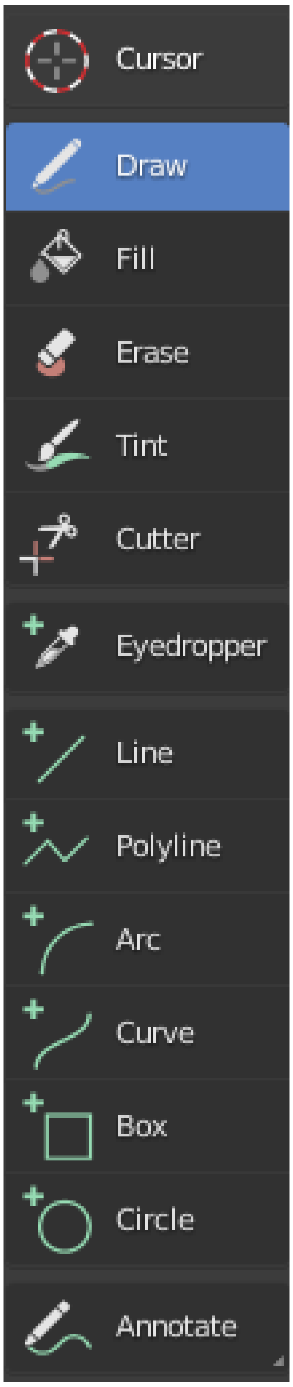
Add Weight Data When enabled, weight data is added to new strokes according to the current vertex group and weight. If there is no vertex group selected, no weight data is added.

Useful for example in cut-out animation for adding new drawing on the same vertex group without the need to creating it afterwards.

See *Weight Paint Mode* for more information.

Additive Drawing When creating new frames adding strokes with drawing tools, the strokes from the previous/active frame are include as a basis for the new one. When erasing existing strokes using Additive Drawing a new keyframe will be added.

Drawing Tools



Cursor Change the location of the 3D Cursor.

Draw Draw free-hand strokes.

Fill Automatic fill closed strokes areas.

Erase Erase strokes.

Tint Colorize stroke points.

Cutter Cut strokes in between others.

Eyedropper Eyedropper to create new materials or palette color based on sampled colors in the 3D Viewport.

Line Draw straight lines.

Polyline Draw straight multiple lines.

Arc Draw simple arcs.

Curve Draw complex Bézier style curves.

Box Draw rectangular shapes.

Circle Draw oval shapes.

Annotate Draw free-hand annotation.

> **Annotate Line** Draw straight line annotation.
>
> **Annotate Polygon** Draw a polygon annotation.
>
> **Annotate Eraser** Erase previous drawn annotations.

Tools

Draw Tool

Reference

> **Mode** Draw Mode
>
> **Tool** *Toolbar → Draw*

The Draw tool allows you to draw free-hand strokes.

Brush Settings

Material Data-block selector for the *material*.

Radius The radius of the brush in pixels.

> F allows you to change the brush size interactively by dragging the pointer or by typing a number then confirm.
>
> **Use Pressure (pressure sensitivity icon)** Uses stylus pressure to control how strong the effect is. The gradient of the pressure can be customized using the *curve widget*.

Strength Control the stroke transparency (alpha). From fully transparent (0.0) to fully opaque (1.0).

> You can change the brush strength interactively by pressing `Shift-F` in the 3D Viewport and then moving the pointer and then `LMB`. You can also enter the size numerically.
>
> **Use Pressure (pressure sensitivity icon)** Uses stylus pressure to control how strong the effect is. The gradient of the pressure can be customized using the *curve widget*.

Advanced

Input Samples Controls how often the input device is read to generate points on the stroke. Higher values give a higher precision (more points) but produce an irregular stroke, while lower values give a lower precision (fewer points) but produce a soften stroke. (0 disabled extra input device samples.)

You have to set up this value according to your input device to obtain the right balance between accuracy and softness for your strokes. See *Input Device* for more information.

Active Smooth The number of smoothing iterations to apply to the stroke while drawing.

Angle Direction of the input device that gives the maximum thickness to the stroke (0° for horizontal).

Factor Amount of thickness reduction when the stroke is perpendicular to the *Angle* value.

Hardness Amount of transparency (alpha) to apply from the border of the point to the center. Works only when the brush is using stroke materials of *Dot* or *Box* style.

Aspect Ratio Controls the width and height of the alpha gradient.

X, Y

Stroke

Post-Processing

Post-processing methods that are executed on the strokes when you finished drawing, right after releasing the `LMB` or `Pen` tip. You can toggle the use of post-processing using the checkbox in the section panel header.

Smooth Strength of smoothing process on the points location along the stroke.

Iterations The number of smoothing iterations to apply to the stroke.

Subdivision Steps Number of subdivisions to apply to newly created strokes.

Simplify Reduces final points numbers in the stroke with an adaptive algorithm.

Trim Strokes End Automatically trim intersection strokes ends.

Randomize

Adds randomness to the position of the points along the stroke. You can toggle the use of Randomize using the checkbox in the section panel header.

Radius The amount of randomness to apply using the pressure of the input device.

Strength The amount of randomness to apply to the stroke strength value (alpha).

UV The amount of randomness to apply to the UV rotation.

Hue, Saturation, Value Randomizes the hue, saturation, and value of the stroke's *Color*.

Jitter The amount of jittering to add to the stroke.

Common Options

Stroke Random (stroke icon) Use randomness only at stroke level.

Use Pressure (pressure sensitivity icon) Uses the stylus pressure to control how strong the effect is. The gradient of the pressure can be customized using the *curve widget*.

Stabilize Stroke

Stabilize Stroke helps to reduce jitter of the strokes while drawing by delaying and correcting the location of points. You can toggle the use of *Stabilize Stroke* using the checkbox in the section panel header.

Radius Minimum distance from the last point before the stroke continues.

Factor A smooth factor, where higher values result in smoother strokes but the drawing sensation feels like as if you were pulling the stroke.

Cursor

The cursor can be disabled by toggling the checkbox in the *Cursor* header.

Show Fill Color while Drawing Shows the brush linked material color in the viewport.

Usage

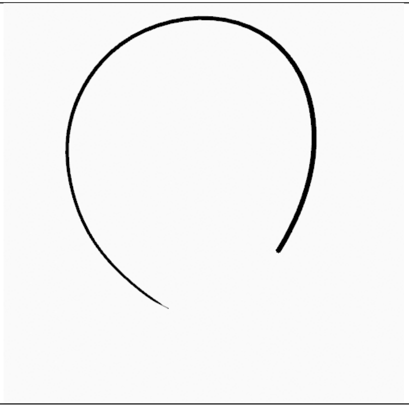
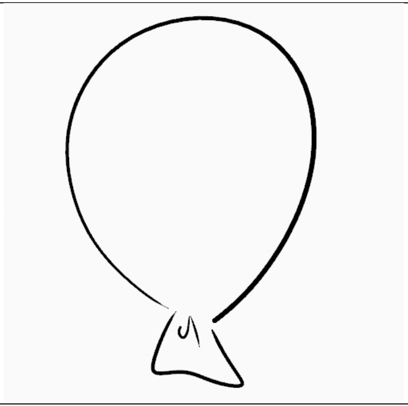
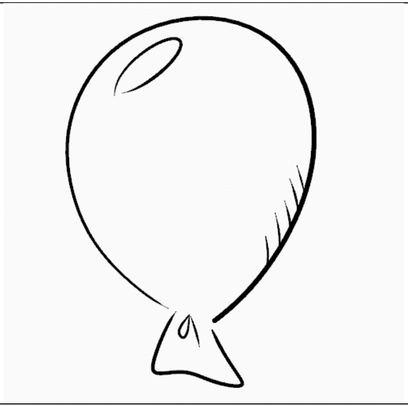
Selecting a Brush and Material

In the Tool Settings select the brush, material and color type to use with the tool. The Draw tool uses *Draw Brush* types. See *Brush Settings* for more information.

Free-hand Drawing

Click and hold LMB or use the pen tip to make free-hand drawing on the viewport.

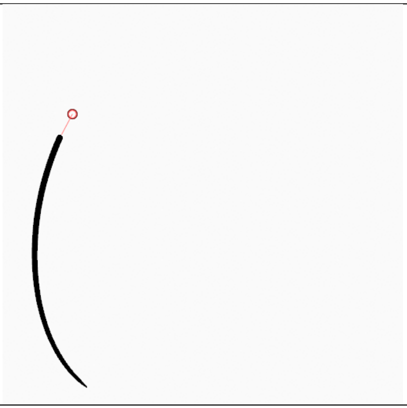
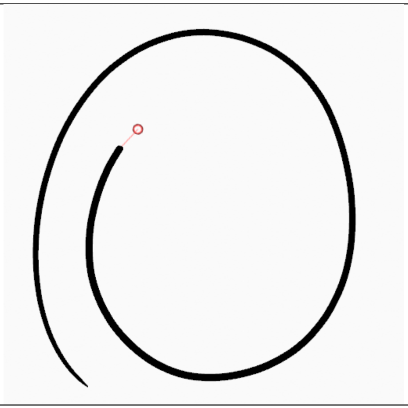
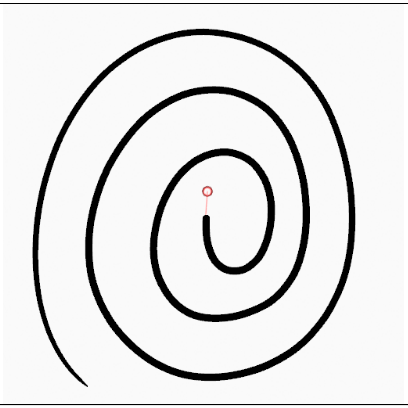
Table 88: Drawing free-hand strokes.

Stabilize Stroke

Shift-LMB toggle the use of *Stabilize Stroke* on the brush to have more control while drawing and get smoother lines.

Table 89: Drawing strokes using *Stabilize Stroke*.

Straight Lines

`Alt-LMB` Constrains the drawing of the strokes to horizontal or vertical straight lines.

Switching to the Erase Tool

`Ctrl-LMB` changes temporally to the active Erase tool. See *Erase Tool* for more information.

You can also use `B` to delete all the points in the selected drawing area.

Fill Tool

Reference

  Mode Draw Mode

  Tool *Toolbar* → *Fill*

The Fill tool is used to automatically fill closed strokes areas.

Brush Settings

You can also configure the brush main settings exposed on the Tool Settings for convenience.

Direction `Ctrl` The portion of area to fill.

  Normal Fills the area inside the shape under the cursor.

  Inverted Fills the area outside the shape under the cursor.

Leak Size Size in pixel to consider the leak as closed.

Thickness The thickness radius of the boundary stroke in pixels.

Simplify Number of simplify steps to apply to the boundary line. Higher values reduce the accuracy of the final filled area.

Advanced

Boundary Sets the type of fill boundary limits calculation to perform.

All Use the thickness of the strokes and the editing lines together.

Stroke Use only the thickness of the strokes (ignore edit lines).

Line Use only the edit lines (ignore strokes).

Show Lines (grid icon) Toggle show help lines to see the fill boundary.

Layers Determines which *Layers* are used for boundary strokes.

Visible Calculates boundaries based on all visible layers.

Active Calculates boundaries based on the active later.

Layer Above Calculates boundaries based on the layer above the active layer.

Layer Below Calculates boundaries based on the layer below the active layer.

All Above Calculates boundaries based on all layers above the active layer.

All Below Calculates boundaries based on all layers below the active layer.

Resolution Multiplier for fill resolution. Higher values gives better fill boundary accuracy but slower time for calculations.

Ignore Transparent When enabled, strokes with transparency does not take into account on fill boundary calculations.

The value slider controls the threshold to consider a material transparent.

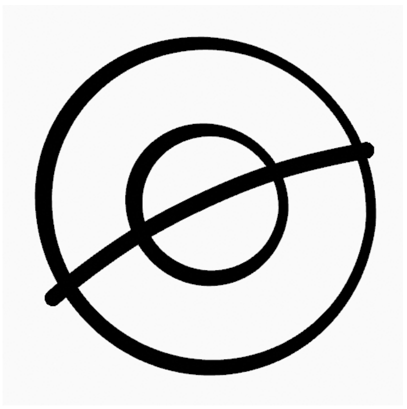
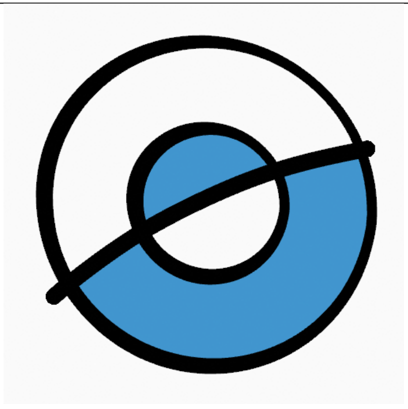
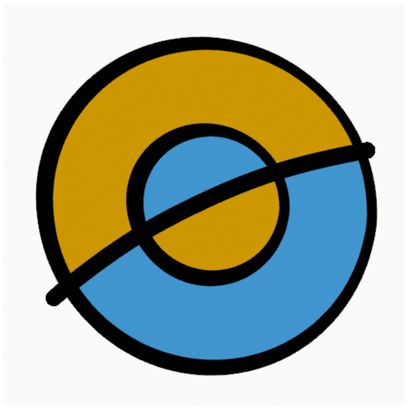
Usage

Selecting a Brush and Material

In the Tool Settings select the brush, material and color type to use with the tool. The Fill tool uses *Fill Brush* types. See *Brush Settings* for more information.

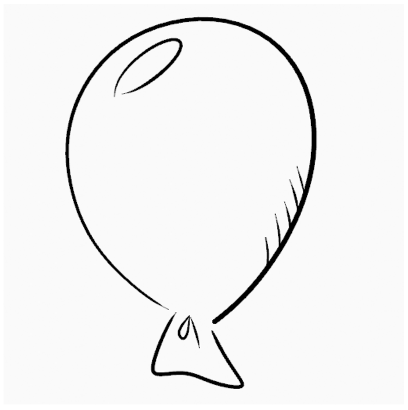
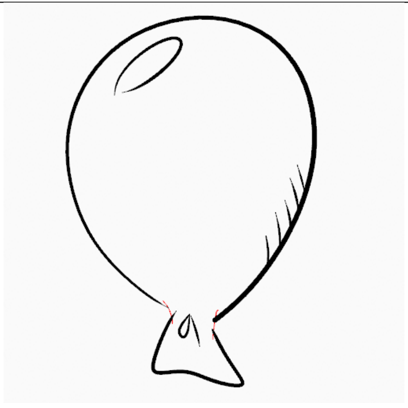
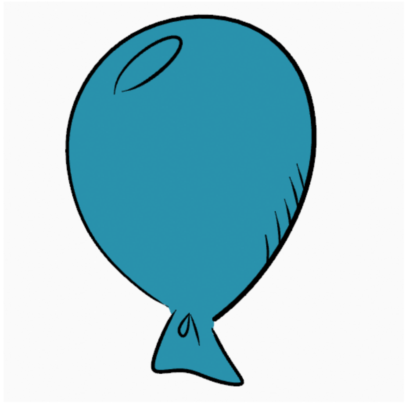
Filling Areas

Click LMB in a closed stroke area. The tool will automatically calculate the boundary and create a new closed stroke filled with the material selected.

 Fig. 1484: Original Drawing.	 Fig. 1485: Use the fill tool to leak materials on closed areas.	 Fig. 1486: Final filled drawing.

Boundary Strokes

If you have a large gap in an area that you want fill, you can use *Boundary Strokes*, a temporary help lines for closing open shapes. To create a Boundary Strokes use `Alt-LMB` and draw a line to close the desired area.

 Fig. 1487: Original Drawing.	 Fig. 1488: Add Boundary Strokes to close open areas (red lines).	 Fig. 1489: Use Fill Tool to leak material on the new closed area.

When you are satisfied with the fill result you can delete the Boundary strokes using the *Clean Up* tool in the *Grease Pencil Menu* in Edit Mode.

Switch to Draw Tool

Use `Ctrl-LMB` to change temporary to the active draw tool. For example to manually cover small areas difficult to reach for the Fill tool. See *Draw Tool* for more information.

Erase Tool

Reference

Mode Draw Mode

Tool *Toolbar* → *Erase*

The Erase tool erase already drawn strokes.

Brush Settings

Radius The radius of the brush in pixels.

`F` allows you to change the brush size interactively by dragging the pointer or by typing a number then confirm.

Use Pressure (pressure sensitivity icon) Uses stylus pressure to control how strong the effect is.

Occlude Eraser (overlapping squares icon) Erase only strokes visible and not occluded by geometry.

Mode Determines how the erase tool behaves.

Dissolve To simulate a raster type eraser, this eraser type affects the strength and thickness of the strokes before actually delete a point.

Strength Control how much will affect the eraser has on the stroke transparency (alpha).

You can change the brush strength interactively by pressing `Shift-F` in the 3D Viewport and then moving the pointer and then `LMB`. You can also enter the size numerically.

Use Pressure (pressure sensitivity icon) Uses stylus pressure to control how strong the effect is.

Affect Stroke Strength The amount of deletion of the stroke strength (alpha) while erasing.

Affect Stroke Thickness The amount of deletion of the stroke thickness while erasing.

Point Delete one point at a time.

Stroke Delete an entire stroke.

Display Cursor Shows the brush shape in the viewport.

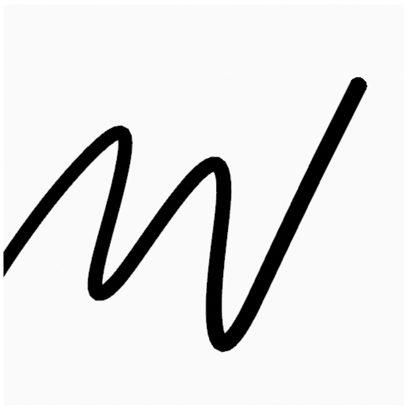
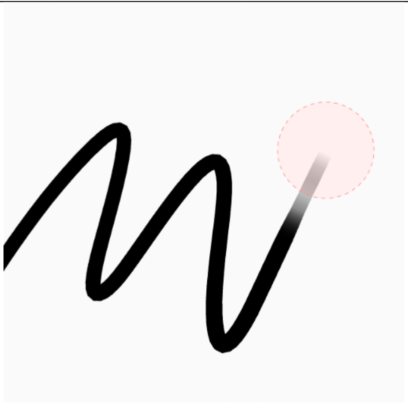
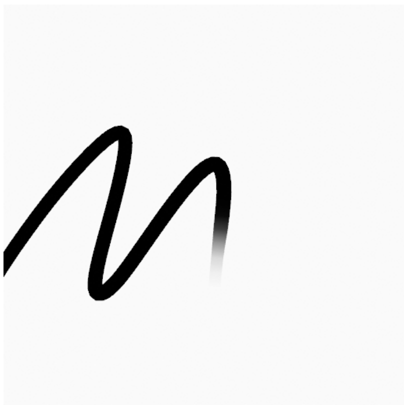
Usage

Selecting a Brush

In the Tool Settings select the brush to use with the tool. The Erase tool uses *Erase Brush* types (soft, point and stroke).

Dissolve Erasing

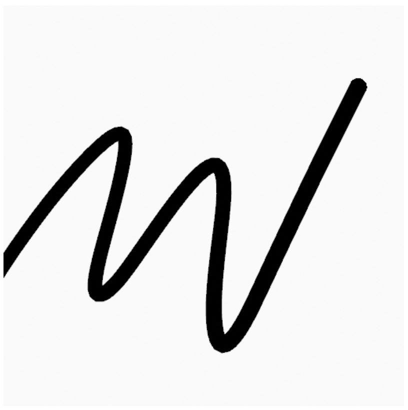
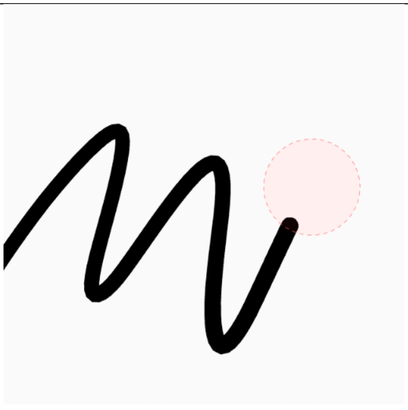
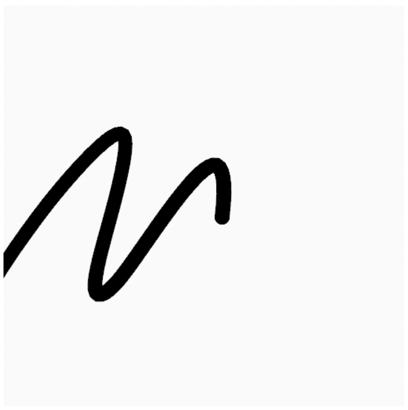
- Select an erase brush of type Soft/Hard.
- Adjust brush settings.
- Click and hold `LMB` or use the `Pen` tip to delete strokes on the viewport.

 Fig. 1490: Original drawing.	 Fig. 1491: The eraser affect the transparency of the strokes.	 Fig. 1492: Final result.

Point Erasing

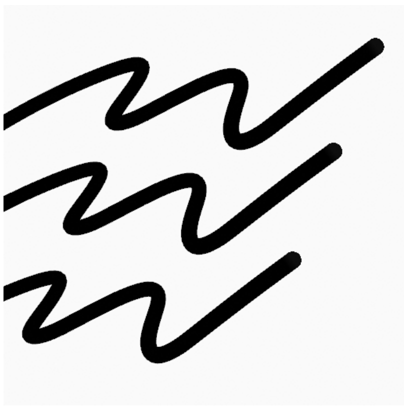
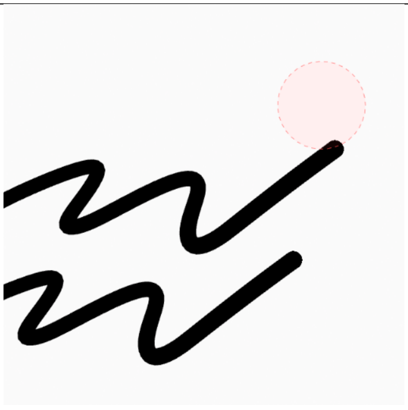
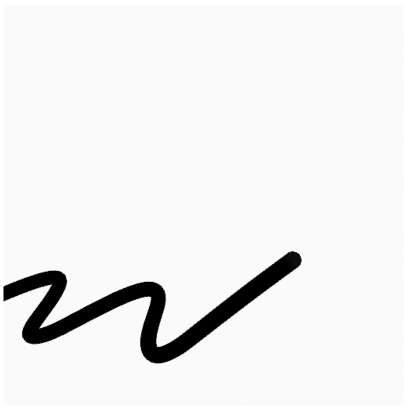
- Select an erase brush of type Point.
- Adjust brush settings.

- Click and hold LMB or use the Pen tip to delete strokes on the viewport.

 Fig. 1493: Original drawing.	 Fig. 1494: The eraser delete one point at a time.	 Fig. 1495: Final result.

Stroke Erasing

- Select an erase brush of type Stroke.
- Adjust brush settings.
- Click and hold LMB or use the Pen tip to delete strokes on the viewport.

 Fig. 1496: Original drawing.	 Fig. 1497: The eraser delete one stroke at a time.	 Fig. 1498: Final result.

Tint Tool

Reference

Mode Draw Mode

Tool *Toolbar → Tint*

The Tint tool allows you to paint onto strokes point mixing the material base color with a selected vertex color.

Usage

Selecting a Brush, Color & Mode

In the Tool Settings select the brush, color and mode to use with the tool.

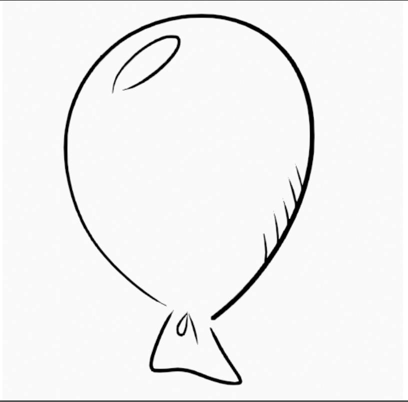
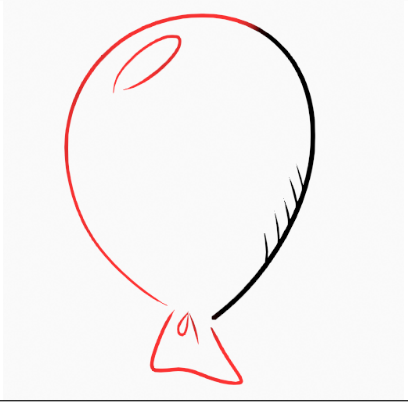
You can configure the brush main settings included in the Tool Settings for convenience. For the vertex paint brushes configuration and settings see *Vertex Paint Brush*.

`Ctrl-LMB` erase the vertex color.

Painting

Click and hold `LMB` or use the pen tip to paint onto the stroke points.

Table 90: Vertex painting stroke points.

Cutter Tool

Reference

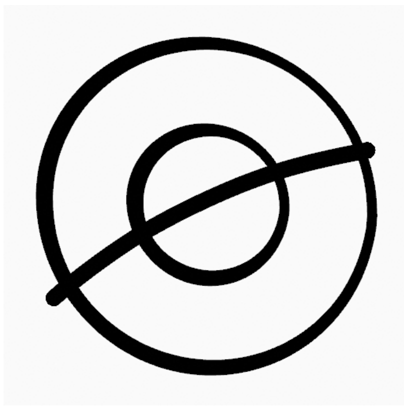
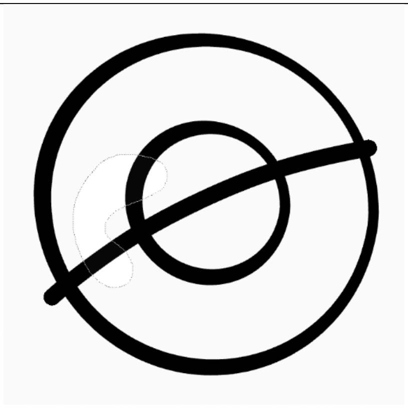
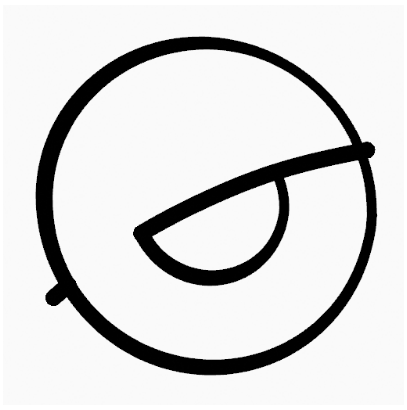
Mode Draw Mode

Tool *Toolbar → Cutter*

The Cutter tool delete points in between intersecting strokes.

Usage

Draw a dotted line around the strokes you want to be cut. After releasing the pointer all the points on the selected strokes will be deleted until another intersecting stroke is found.

 Fig. 1499: Original drawing.	 Fig. 1500: Lasso Selecting the strokes to be cut.	 Fig. 1501: Final result.

Eyedropper

Reference

Mode Draw Mode

Tool *Toolbar → Eyedropper*

The Eyedropper tool is used to create materials or palette color based on sampled colors in the 3D Viewport.

Usage

- `LMB` Create a stroke material.
- `Shift-LMB` Create a fill material.
- `Shift-Ctrl-LMB` Create both a stroke and fill material.

Line Tool

Reference

Mode Draw Mode

Tool *Toolbar → Line*

The Line tool create straight lines.

Tool Settings

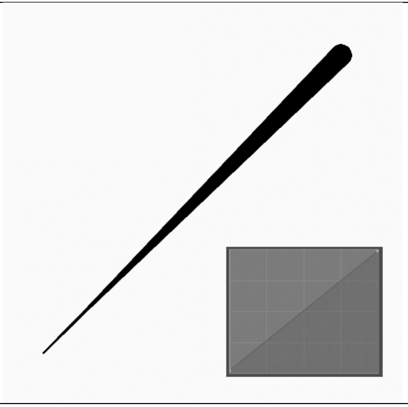
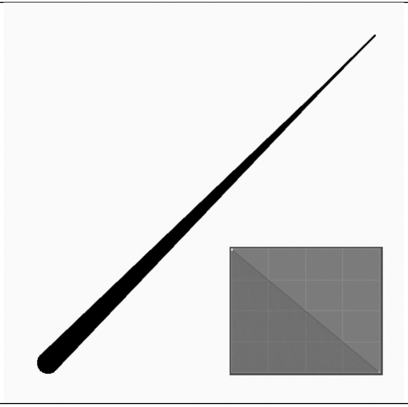
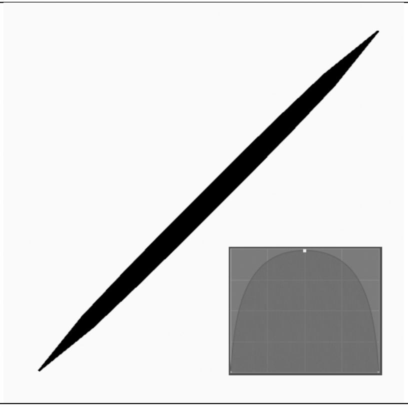
You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: *Draw Brush*.

Subdivisions The number of stroke points between each stroke edge.

Thickness Profile Use a *curve widget* to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve When enabled, the stroke use a curve profile to control the thickness along the line.

Table 91: Different thickness profile samples.

Usage

Selecting a Brush and Material

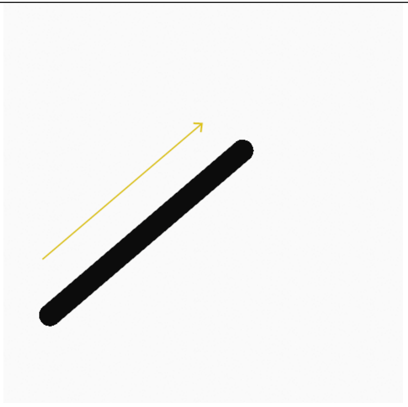
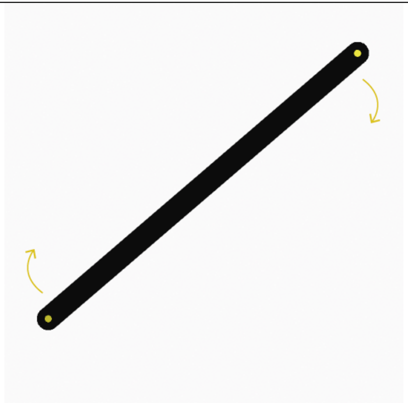
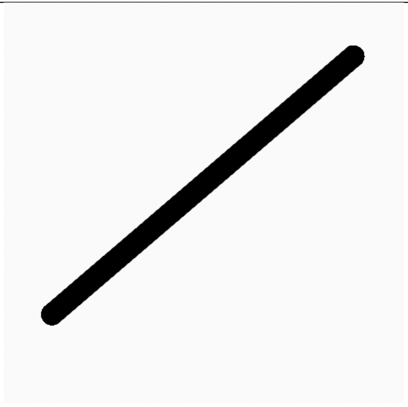
In the Tool Settings select the brush, material and color type to use with the tool. The Line tool uses *Draw Brush* types. See *Brush Settings* for more information.

Creating Lines

1. Click (LMB or the Pen tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can move the start and end point by clicking and dragging on the yellow manipulators.
4. Then confirm (Return/MMB) or cancel (Esc/RMB).

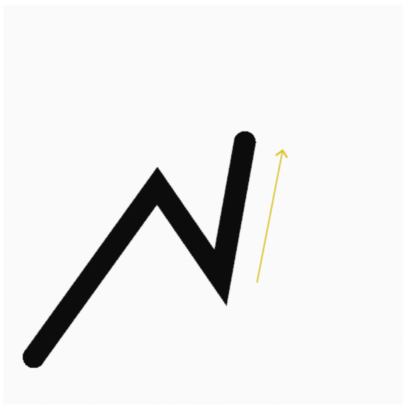
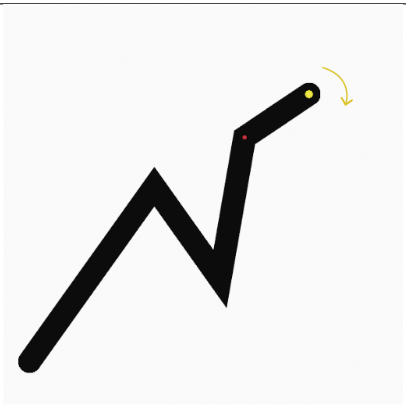
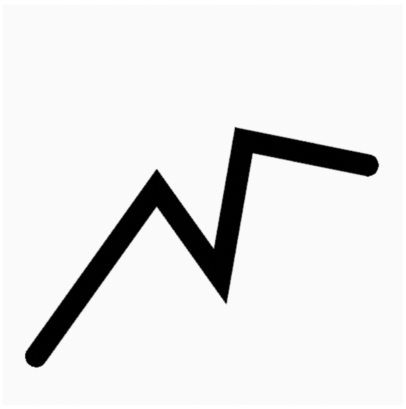
While dragging you can use Shift to snapping the line to horizontal, vertical or 45° angle or use Alt to create the line from a center point.

NumpadPlus and NumpadMinus or using the mouse Wheel will increase or decrease the amount of points in the final line.

 Fig. 1502: click and dragging the start point.	 Fig. 1503: Moving start and end points with manipulators.	 Fig. 1504: The line after confirming.

Extruding

Before confirming you can use `E` to extrude the end point of the line to generate multiple connected lines.

 Fig. 1505: End point extruding.	 Fig. 1506: Moving the end point of the last line with the manipulator.	 Fig. 1507: The connected lines after confirming.

Polyline Tool

Reference

Mode Draw Mode

Tool *Toolbar → Polyline*

The Polyline tool create multiple straight lines.

Tool Settings

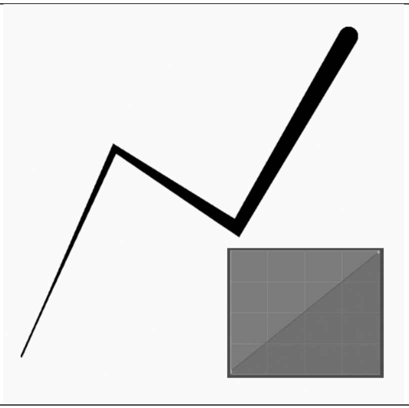
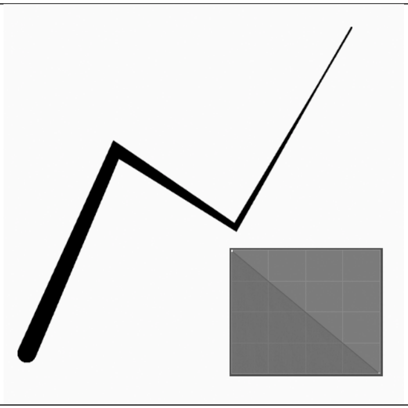
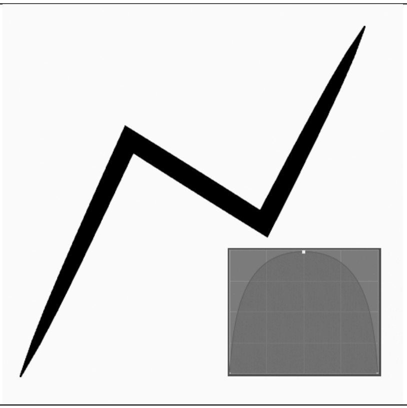
You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: *Draw Brush*.

Subdivisions The number of stroke points between each stroke edge.

Thickness Profile Use a *curve widget* to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve When enabled, the stroke use a curve profile to control the thickness along the line.

Table 92: Different thickness profile samples.

Usage

Selecting a Brush and Material

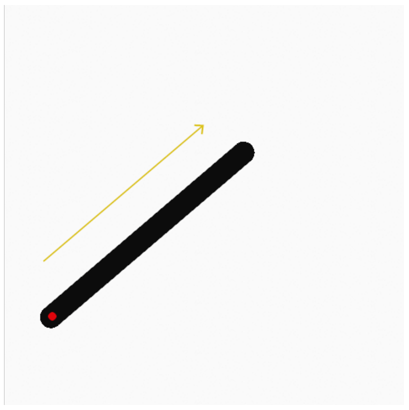
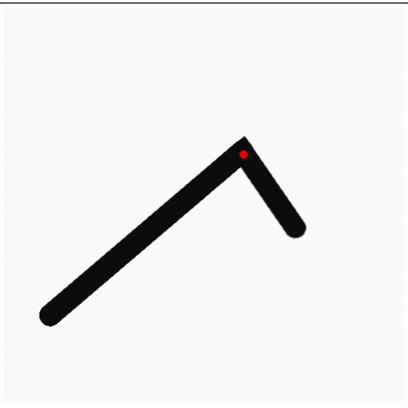
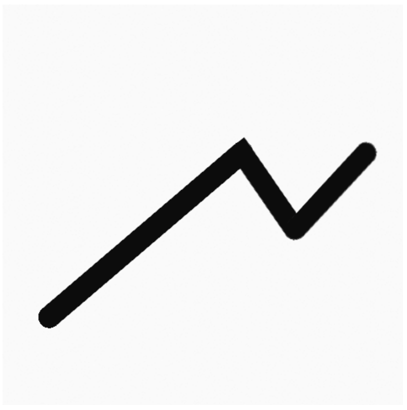
In the Tool Settings select the brush, material and color type to use with the tool. The Line tool uses *Draw Brush* types. See *Brush Settings* for more information.

Creating Polylines

1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. Click multiple times on different locations to create multiple connected lines.
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

While dragging you can use `Shift` to snapping the line to horizontal, vertical or 45° angle.

`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final line.

 Fig. 1508: click and dragging the start point.	 Fig. 1509: Click multiple times to create multiple connected lines.	 Fig. 1510: The polyline after confirming.

Arc Tool

Reference

Mode Draw Mode

Tool *Toolbar → Arc*

The Arc tool create simple arcs.

Tool Settings

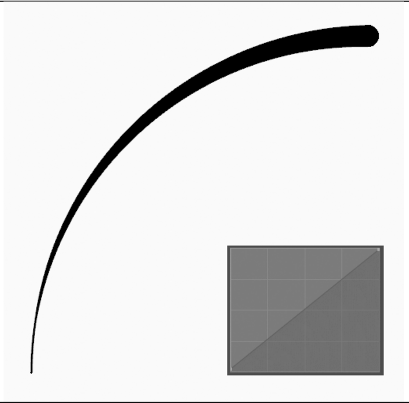
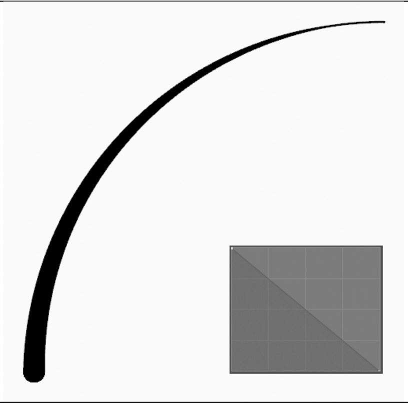
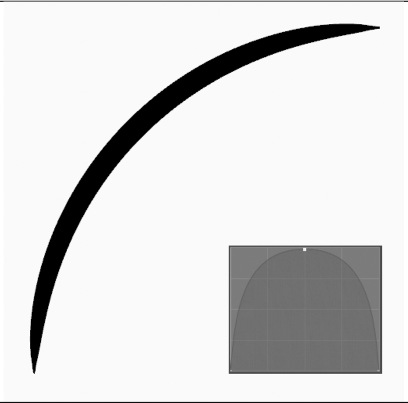
You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: *Draw Brush*.

Subdivisions The number of stroke points between each stroke edge.

Thickness Profile Use a *curve widget* to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve When enabled, the stroke use a curve profile to control the thickness along the arc.

Table 93: Different thickness profile samples.

Usage

Selecting a Brush and Material

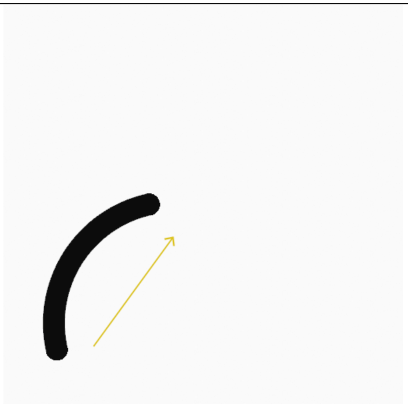
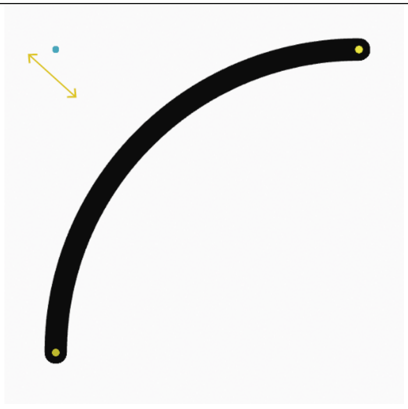
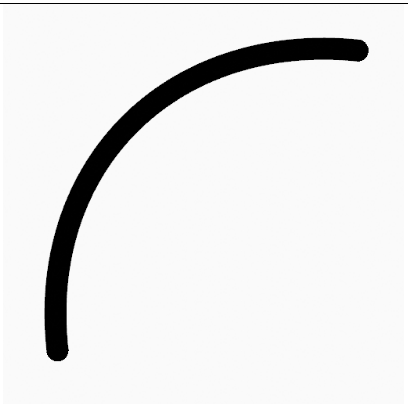
In the Tool Settings select the brush, material and color type to use with the tool. The Arc tool uses *Draw Brush* types. See *Brush Settings* for more information.

Creating Arcs

1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can tweak the arc using a single cyan manipulator (hand icon).
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

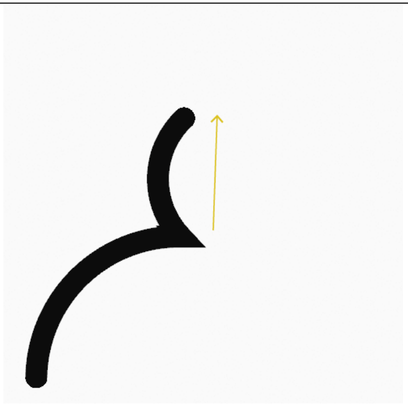
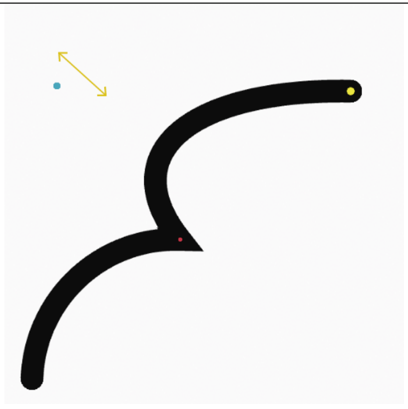
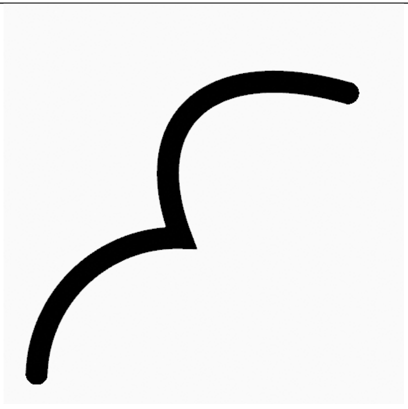
While dragging you can use `Shift` to make a perfect arc, use `Alt` to create the arc from a center point or `M` to flip.

`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final arc.

 Fig. 1511: click and dragging the start point.	 Fig. 1512: Tweaking arc with the manipulator.	 Fig. 1513: The arc after confirming.

Extruding

Before confirming you can use E to extrude the end point of the arc to generate multiple connected arcs.

 Fig. 1514: End point extruding.	 Fig. 1515: Tweaking the last arc with the manipulator.	 Fig. 1516: The connected arcs after confirming.

Curve Tool

Reference

Mode Draw Mode

Tool *Toolbar* → *Curve*

The Curve tool create complex Bézier style curves.

Tool Settings

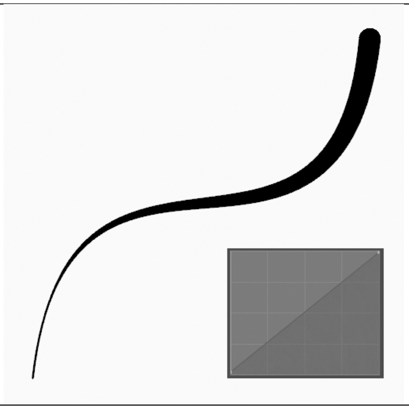
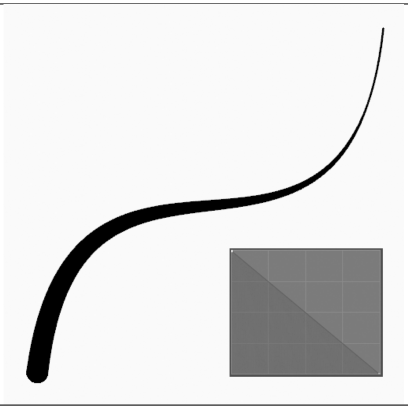
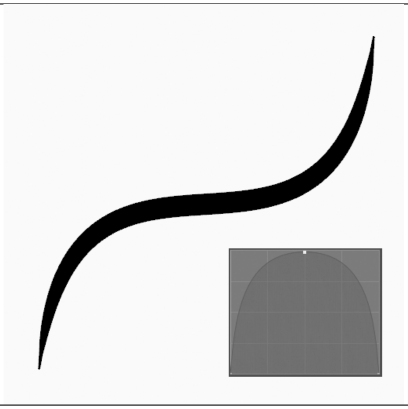
You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: *Draw Brush*.

Subdivisions The number of stroke points between each stroke edge.

Thickness Profile Use a *curve widget* to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve When enabled, the stroke use a curve profile to control the thickness along the curve.

Table 94: Different thickness profile samples.

Usage

Selecting a Brush and Material

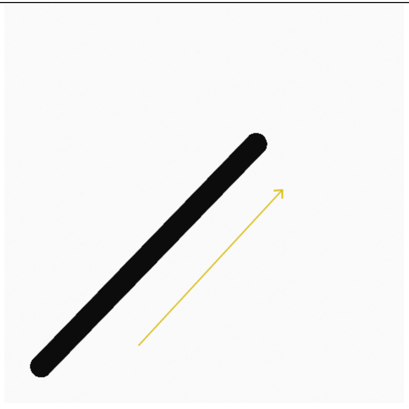
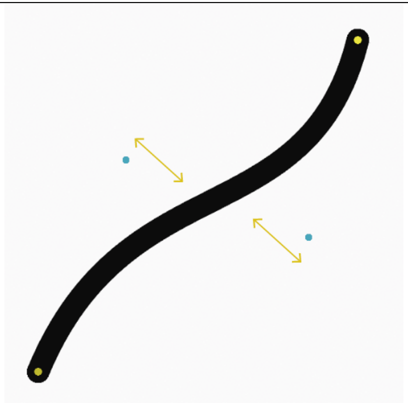
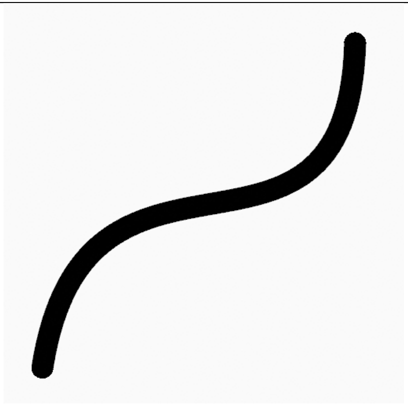
In the Tool Settings select the brush, material and color type to use with the tool. The Curve tool uses *Draw Brush* types. See *Brush Settings* for more information.

Creating Curves

1. Click (`LMB` or the `Pen` tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can tweak the curve using two cyan Bézier like manipulators.
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

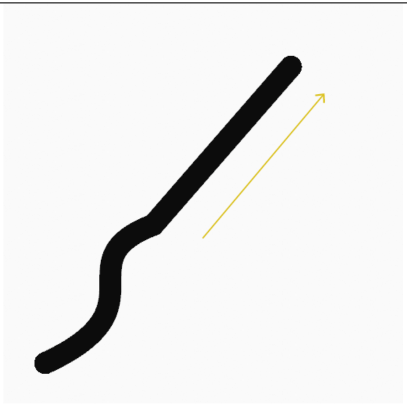
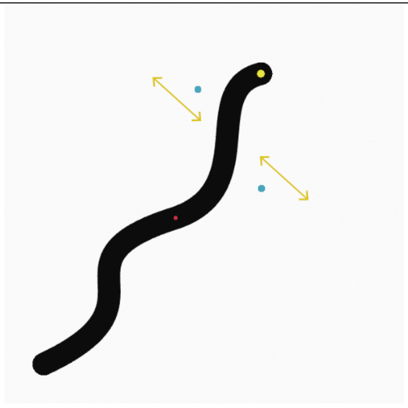
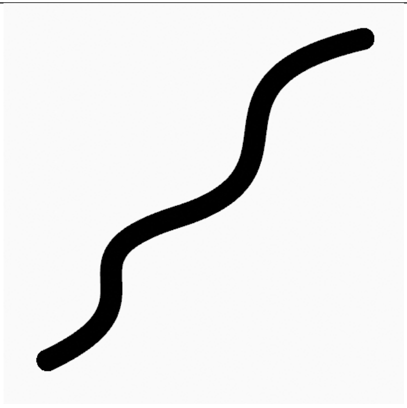
While dragging you can hold `Shift` to use only one manipulator to tweak the curve (like the Arc tool), use `Alt` to create the arc from a center point or `M` to flip.

`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final curve.

 Fig. 1517: click and dragging the start point.	 Fig. 1518: Tweaking curve with the manipulators.	 Fig. 1519: The curve after confirming.

Extruding

Before confirming you can use E to extrude the end point of the curve to generate multiple connected curves.

 Fig. 1520: End point extruding.	 Fig. 1521: Tweaking the last curve with the manipulators.	 Fig. 1522: The connected curves after confirming.

Box Tool

Reference

Mode Draw Mode

Tool *Toolbar* → *Box*

The Box tool create rectangular shapes.

Tool Settings

You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: *Draw Brush*.

Subdivisions The number of stroke points between each stroke edge.

Thickness Profile Use a *curve widget* to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve When enabled, the stroke use a curve profile to control the thickness along the line.

Usage

Selecting a Brush and Material

In the Tool Settings select the brush, material and color type to use with the tool. The Box tool uses *Draw Brush* types. See *Brush Settings* for more information.

Creating Boxes

1. Click (LMB or the Pen tip) and drag the start point.

2. Release on the desired end point.
3. After releasing you can move the start and end point by clicking and dragging on the yellow manipulators.
4. Then confirm (`Return`/`MMB`) or cancel (`Esc`/`RMB`).

While dragging you can use `Shift` to make a perfect square or use `Alt` to create the box from a center point.

`NumpadPlus` and `NumpadMinus` or using the mouse `Wheel` will increase or decrease the amount of points in the final box.

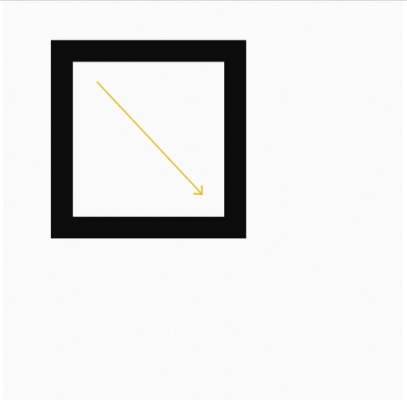


Fig. 1523: click and dragging the start point.

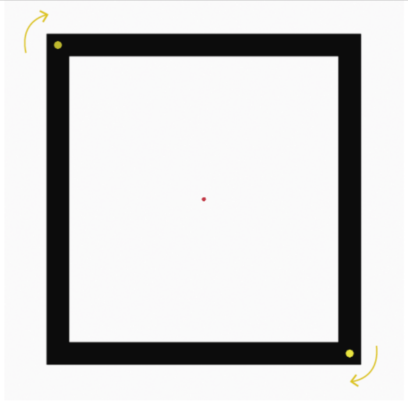


Fig. 1524: Moving start and end points with manipulators.

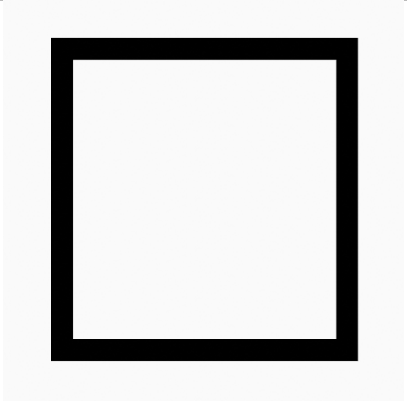


Fig. 1525: The box after confirming.

Circle Tool

Reference

Mode Draw Mode

Tool *Toolbar* → *Circle*

The Circle tool create oval shapes.

Tool Settings

You can configure the brush main settings exposed on the Tool Settings for convenience. For the draw brushes configuration and settings see: *Draw Brush*.

Subdivisions The number of stroke points between each stroke edge.

Thickness Profile Use a *curve widget* to define the stroke thickness from the start (left) to end (right) of the stroke.

Use Curve When enabled, the stroke use a curve profile to control the thickness along the line.

Usage

Selecting a Brush and Material

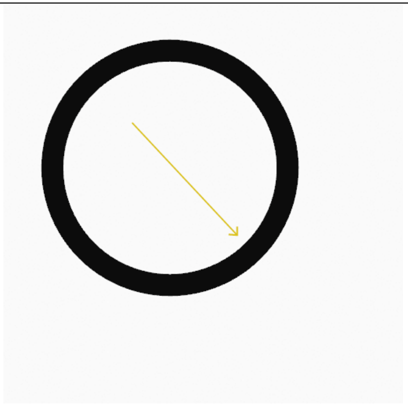
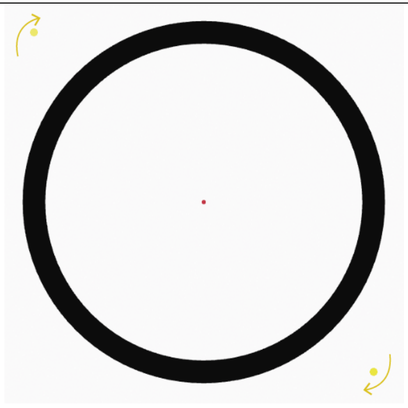
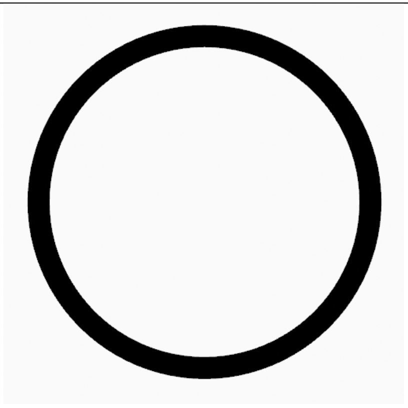
In the Tool Settings select the brush, material and color type to use with the tool. The Circle tool uses *Draw Brush* types. See *Brush Settings* for more information.

Creating Circles

1. Click (LMB or the Pen tip) and drag the start point.
2. Release on the desired end point.
3. After releasing you can move the start and end point by clicking and dragging on the yellow manipulators.
4. Then confirm (Return/MMB) or cancel (Esc/RMB).

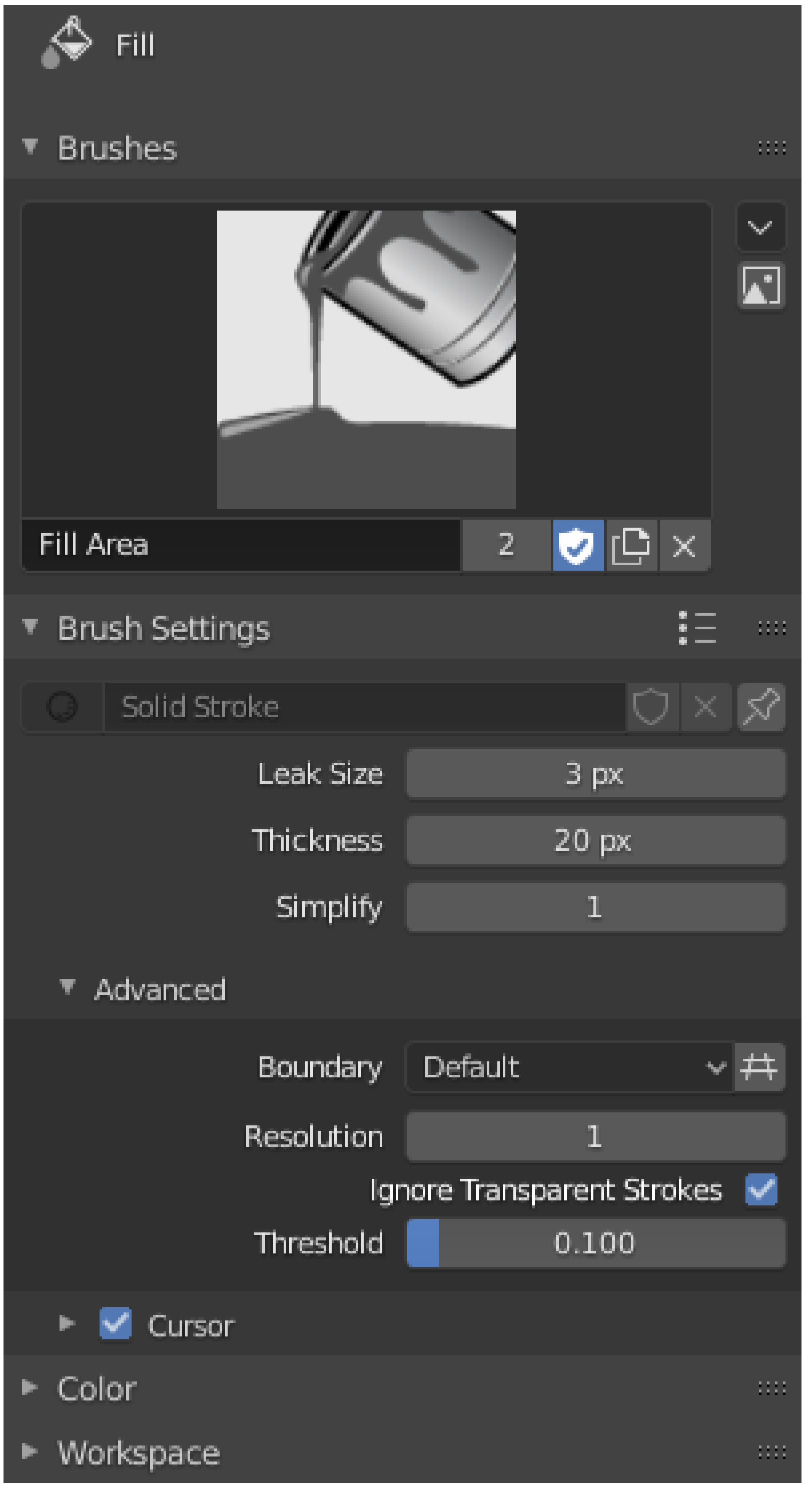
While dragging you can use Shift to make a perfect circle or use Alt to create the circle from a center point.

NumpadPlus and NumpadMinus or using the mouse Wheel will increase or decrease the amount of points in the final circle.

 Fig. 1526: Click and dragging the start point.	 Fig. 1527: Moving start and end points with manipulators.	 Fig. 1528: The circle after confirming.

Tools Settings

Brushes



Fill

Brushes

Fill Area 2

Brush Settings

Solid Stroke

Leak Size 3 px

Thickness 20 px

Simplify 1

Advanced

Boundary Default

Resolution 1

Ignore Transparent Strokes ✓

Threshold 0.100

✓ Cursor

Color

Workspace

Fig. 1529: Brush data-block panel.

Brush The *Data-Block Menu* to select a preset brush type or a custom brush.

Add Brush When you add a brush, the new brush is a clone of the current one.

Brush Specials

Reset Brush Reset the current brush to its default settings.

Reset All Brushes Reset all brushes to their default settings.

Custom Icon Allows definition of a custom brush icon.

Image Path Defines the path to the image to use as custom icon.

Note: In order to save a custom brush in a blend-user, enable *Fake User*.

Brush Types

Draw Brushes

Draw brushes are the special type of brushes that uses Grease Pencil for drawing tools. The brush can be changed in the Tool Settings. The different draw brushes (pencil, Ink, marker, etc.) are settings variations of the same *Draw Brush*. You can create many brushes, each with unique settings to get different artistic result while drawing.

Fill Brushes

Fill brushes are the special type of brushes that uses Grease Pencil for the *Fill* tools. The brush can be changed in the Tool Settings. The different fill brushes are settings variations of the same *Fill Brush*. You can create many brushes, each with unique settings to get different result when filling areas.

Erase Brushes

Erase brushes are the special types of brushes that uses Grease Pencil for *Erase* tools. The brush can be changed in the Tool Settings. Soft and hard eraser brushes are settings variations of the same *Erase Brush*. You can create many brushes, each with unique settings to get different effects while erasing. The *Erase Brush* has also other two special eraser types: point and stroke.

Brush Settings

Material Data-block selector for the *material*. Except for the *Erase* tool of course.

Pin Material (pin icon) Pin the material to the brush.

The final appearance of the strokes is a combination of the brush and material used, binding the material to the brush gives more control and avoids a lack of coordination between the two.

Color

Controls the source of the stroke color. The mode can be pinned to the brush by enabling the Pin icon in the Tool Settings header.

Material

Use the stroke/fill base color material.

Vertex Color

Use Vertex color.

Color Picker The color of the brush. See *Color Picker*.

Color Palette Active Color Palette. See *Color Palette*.

Mode

- **Stroke** Only paint over strokes.
- **Fill** Only paint over fill areas.
- **Stroke and Fill** Paint over strokes and fill areas.

Mix Factor Mixing factor between the selected color and the base material color.

Stroke Placement

Reference

- **Mode** Draw Mode
- **Header** *Stroke Placement*

The Stroke Placement selector helps to select the location in which the newly created strokes are drawn.

Note: The Stroke Placement selected has effect only for new strokes and does not affect the existing ones.

Placement Options

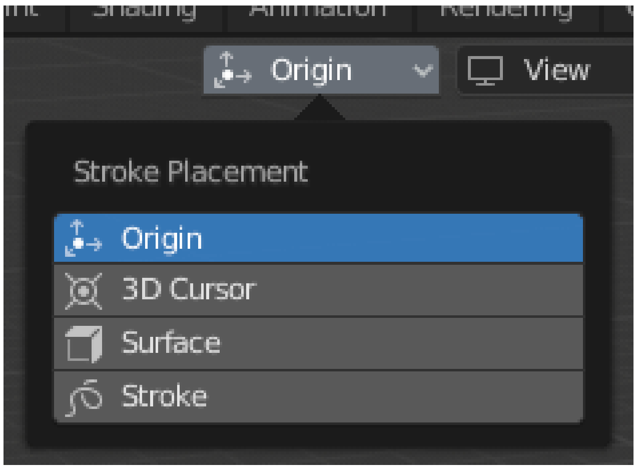


Fig. 1530: Stroke Placement selector on 3D Viewport header.

Origin Strokes are placed at Grease Pencil object origin.

3D Cursor Strokes are placed at 3D cursor.

Surface Strokes will stick on mesh surfaces.

- **Offset** Distance from the mesh surface to place the new strokes.

Stroke Strokes will stick on other strokes.

Target

All Points All the points of the new stroke sticks to other strokes.

End Points Only the start and end points of the new stroke sticks to other strokes.

First Point Only the start point of the new stroke sticks to other strokes.

Examples

Table 95: Stroke using different Stroke Placements.

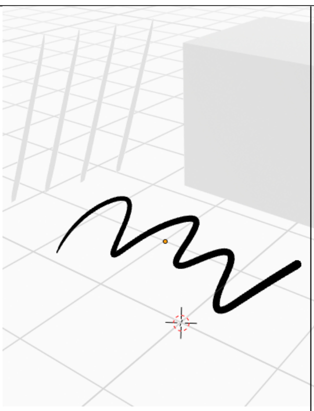
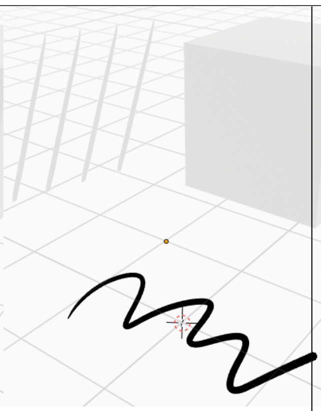
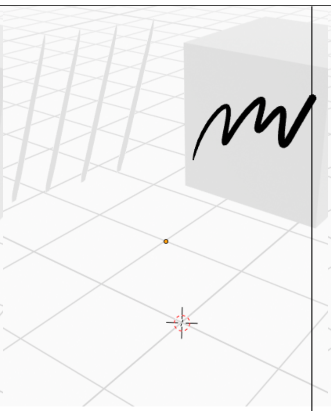
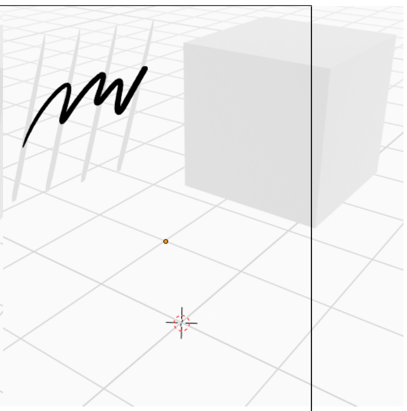
			

Fig. 1531: Origin.	Fig. 1532: 3D Cursor.	Fig. 1533: Surface.	Fig. 1534: Stroke.

Drawing Planes

Reference

Mode Draw Mode and Sculpt Mode

Header *Drawing Planes*

The Drawing Planes selector helps to select the plane in which the newly created strokes are drawn.

To see which plane you are using when drawing strokes, you can enable *Canvas* in *Viewport Overlays*. See *Viewport Display* to know more about Canvas settings.

Note: The Drawing Plane selected has effect only for new strokes and does not affect the existing ones.

Plane Options

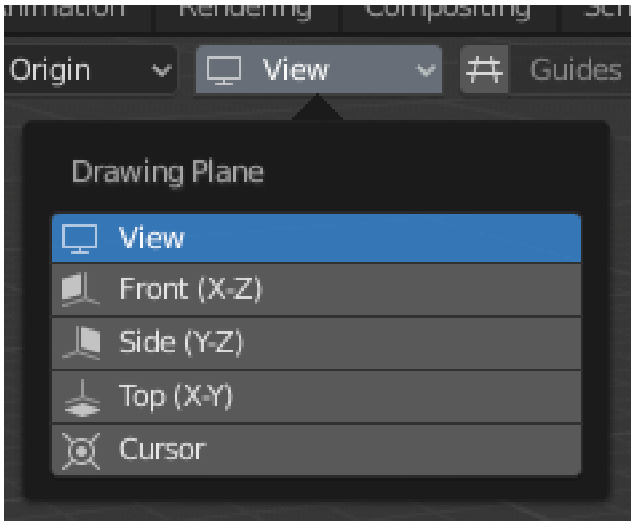


Fig. 1535: Drawing Planes selector in the 3D Viewport header.

Front Strokes are drawn on the plane determined by the XZ axes (front view).

Side Strokes are drawn on the plane determined by the YZ axes (side view).

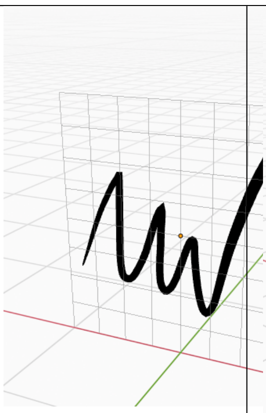
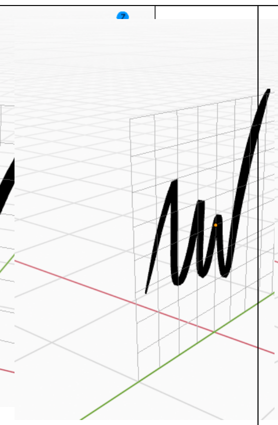
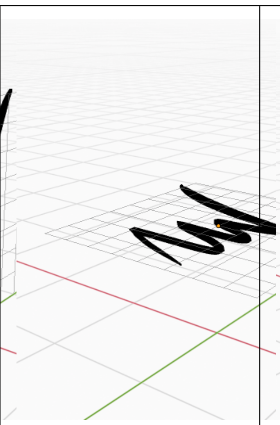
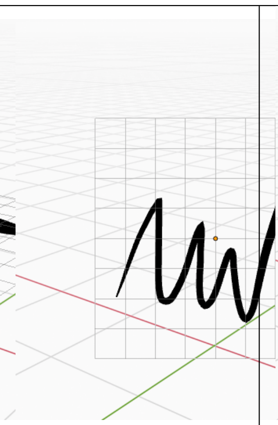
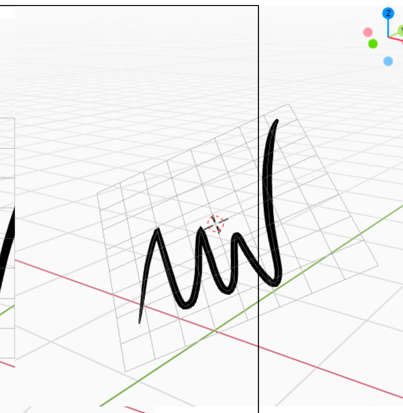
Top Strokes are drawn on the plane determined by the XY axes (top view).

View Strokes are drawn with the current 3D Viewport orientation.

Cursor Strokes are drawn with the current 3D cursor orientation.

Examples

Table 96: Stroke using different Drawing Planes with Canvas overlay activated.

 Fig. 1536: Front.	 Fig. 1537: Side.	 Fig. 1538: Top.	 Fig. 1539: View.	 Fig. 1540: Cursor.

Guides

Reference

Mode Draw Mode

Header *Guides*

Guides are drawing aids that make it easier to create different types of strokes. The Guides can be activated with the button next to the selector (grid icon).

Guide Types

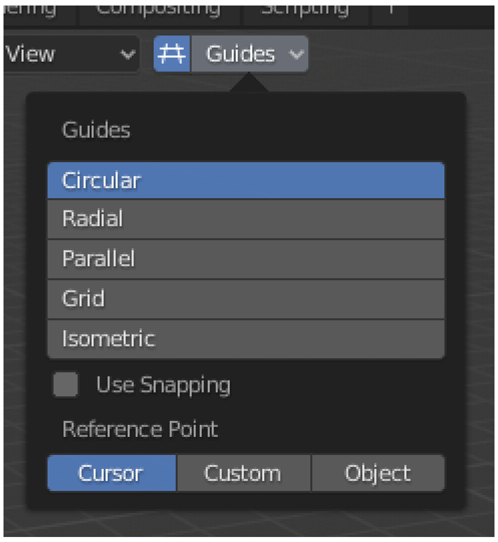


Fig. 1541: Guide selector activated in the 3D Viewport header.

Circular Constrains the drawing of new strokes to form rings from the selected reference point.

Radial Constrains the drawing of new strokes to form rays from the selected reference point.

Parallel Constrains the drawing of new strokes to form parallel lines.

Angle Angle direction of the parallel lines.

Grid Constrains the drawing of new strokes to form parallel horizontal or vertical lines.

Isometric Constrains the drawing of new strokes to vertical or isometric lines.

Angle Angle direction of the isometric lines.

Common Options

Use Snapping When enabled, snap the drawn strokes to an angle or spacing.

Spacing Guide spacing.

Reference Point Determines the origin point to use for the creation of the lines. Applies only for *Circular* and *Radial* guides.

Cursor Use the cursor as a reference point.

Custom Use a custom location as a reference point.

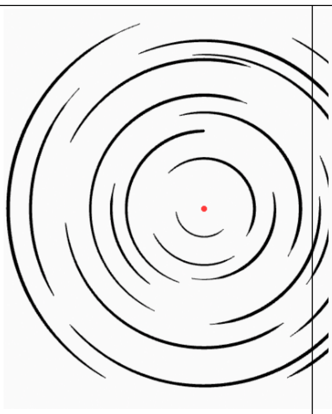
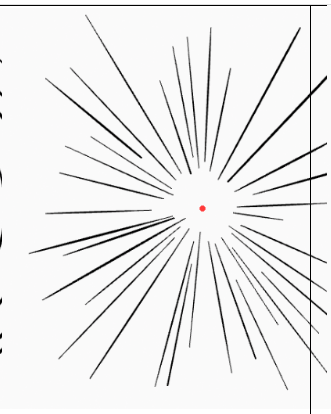
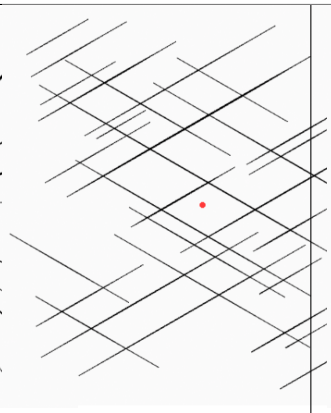
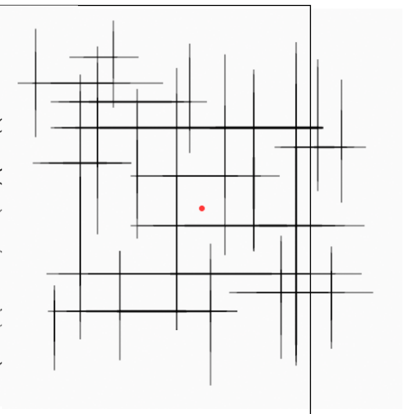
Custom Location X, Y Z

Object Use an object as a reference point.

Object An *Data ID menu* to select the object (usually an empty), which location will be used as a reference point.

Examples

Table 97: Examples of strokes using different types of Guides.

 Fig. 1542: Circular Guides.	 Fig. 1543: Radial Guides.	 Fig. 1544: Parallel Guides (30° Angle).	 Fig. 1545: Grid Guides.

Sculpt Mode

Introduction

Sculpt Mode is similar to *Edit Mode* in that it is used to alter the shape of a drawing, but Sculpt Mode uses a very different workflow: Instead of dealing with individual elements (points and edit lines), an area of the model is altered using a brush. In other words, instead of selecting a group of points, Sculpt Mode manipulates the drawing in the brush region of influence.

Sculpt Mode

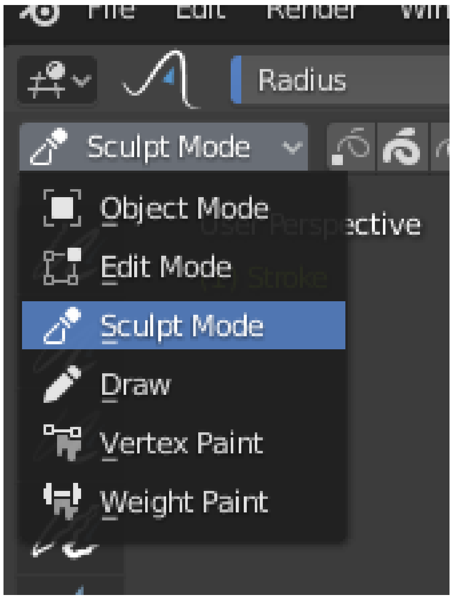


Fig. 1546: 3D Viewport Mode selector: Sculpt Mode.

Sculpt Mode is selected from the *Mode* menu in the 3D Viewport header. Once Sculpt Mode is activated, the Toolbar of the 3D Viewport will change to Sculpt Mode specific panels. A red circle will appear and follow the location of the cursor in the 3D Viewport.

Sculpting Options

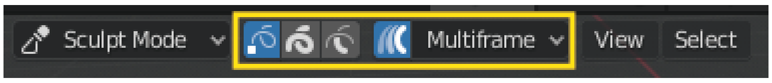


Fig. 1547: General sculpting options.

Selection Mask Sculpt Mode in Grease Pencil allows you to select points or strokes to restrict the effect of the sculpting tools to only a certain areas of your drawing.

You can use the selection tools in the Toolbar for a quick selection. You can restrict sculpting only on the selected points or strokes with the Selection mode buttons.

Multiframe Sometimes you may need to modify several frames at the same time with the sculpting tools.

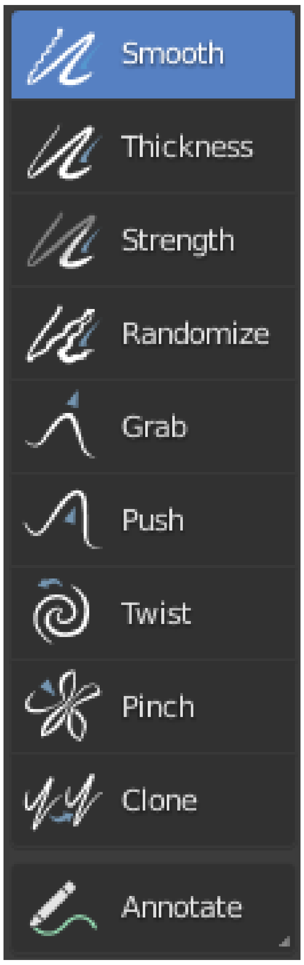
You can activate multiframe edition with the Multiframe button next to the modes selector (faded lines icon). See *Multiframe* for more information.

Keyboard Shortcuts

- Invert stroke toggle `Ctrl`

Sculpting Tools

For Grease Pencil sculpt modes each brush type is exposed as a tool, the brush can be changed in the Tool Settings. See *Brush* for more information.



Smooth Eliminates irregularities in the area of the drawing within the brush's influence by smoothing the positions of the points.

Thickness Increase or decrease the points thickness in the area of the drawing within the brush's influence.

Strength Increase or decrease the points transparency (alpha) in the area of the drawing within the brush's influence.

Randomize Add noise to the strokes in the area of the drawing within the brush's influence by moving points location in a random way.

Grab Used to drag a group of points around. Unlike the other brushes, Grab does not modify different points as the brush is dragged across the model. Instead, Grab selects a group of points on mouse-down, and pulls them to follow the mouse. The effect is similar to moving a group of points in Edit Mode with Proportional Editing enabled.

Push Moves points in the direction of the brush stroke.

Twist Twist the points in counter-clockwise (CCW) or Clockwise (CW) rotation.

Pinch/Inflate Pulls points towards the center of the brush. The inverse setting is Inflate, in which points are pushed away from the center of the brush.

Clone Adds copies of the strokes in the clipboard in the center of the brush. You have to copy the selected strokes you want into the clipboard with `Ctrl-C` before using the tool.

Annotate Draw free-hand annotation.

Annotate Line Draw straight line annotation.

Annotate Polygon Draw a polygon annotation.

Annotate Eraser Erase previous drawn annotations.

Brush Settings

Reference

Mode Sculpt Mode

Panel *Sidebar → Tool → Brush Settings*

Radius This option controls the radius of the brush, measured in pixels. `F` allows you to change the brush size interactively by dragging the mouse and then `LMB` (the texture of the brush should be visible inside the circle). Typing a number then enter while using `F` allows you to enter the size numerically. Brush size can be affected by enabling the pressure sensitivity icon, if you are using a *Graphics Tablet*.

Strength Controls how much each application of the brush affects the model. For example, higher values cause the *Randomize* brush to add noise to the strokes more quickly, and cause the *Smooth* brush to soften the strokes more quickly.

You can change the brush strength interactively by pressing `Shift-F` in the 3D Viewport and then moving the brush and then `LMB`. You can enter the size numerically also while in `Shift-F` sizing. Brush strength can be affected by enabling the pressure sensitivity icon, if a supported tablet is being used.

Use Falloff When enabled, use Strength falloff for the brush. Brush Strength decays with the distance from the center of the brush.

Sculpt Strokes Filters the effect of the brush over the strokes. Applies to *Smooth* and *Randomize* brushes only.

Note: If there is no filter selected *Affect Position* is the default behavior.

Affect Position Toggles the brush effect on the position of the stroke points.

Affect Strength Toggles the brush effect on the strength (alpha) of the stroke points.

Affect Thickness Toggles the brush effect on the thickness of the stroke points.

Affect Pressure Toggles the brush effect on the pressure values of the stroke points.

Affect UV Toggles the brush effect on the UV rotation of the stroke points.

Direction The influence direction of the brush. This can be Add or Subtract.

Cursor

The cursor can be disabled by toggling the checkbox in the *Cursor* header.

Color Set the color of the brush ring. Depending on the current mode there will be options to set a single Color or a Color for Adding/Subtracting.

Edit Mode

Introduction

Blender provides a variety of tools for editing Grease Pencil strokes. These are tools used to add, duplicate, move and delete elements.

Accessing Stroke Editing Tools

These are available through the different tools in the Toolbar, the Stroke menu in the 3D Viewport header, and context menus in the 3D Viewport, as well as individual shortcut keys.

Toolbar

When you select a stroke and `Tab` into Edit Mode, the *Toolbar* changes from *Object Tools* to *Stroke editing Tools*. These are only some of the stroke editing tools.

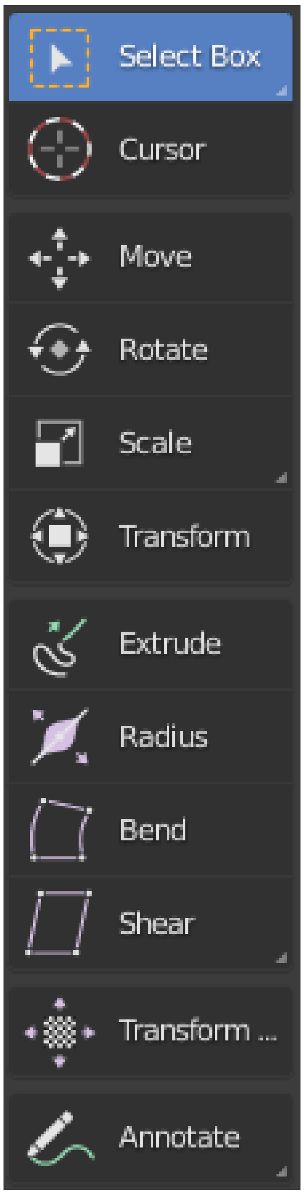
Menus

The *Stroke Menu* is located in the header.

Context Menu

`RMB` brings up the complete *Stroke Menu*.

Editing Tools



Select Select or moved.

Select Box Select geometry by dragging a box.

Select Circle Select geometry by painting on it.

Select Lasso Select geometry by drawing a lasso.

Cursor Change the location of the 3D Cursor.

Move Translation tool.

Rotate Rotation tool.

Scale Scale tool.

Scale Cage Change the scale of an object by controlling its cage.

Transform Tool to adjust the objects translation, rotations and scale.

Extrude `E` Extrusion tools duplicate points, while keeping the new geometry connected with the original points.

Radius `Alt-S` Expand or contract the thickness radius of the selected points.

Bend `Shift-W` Bend selected points between the 3D cursor and the pointer.

Shear `Shift-Ctrl-Alt-S` Shear selected points along the horizontal or vertical screen axis.

> **To Sphere** `Shift-Alt-S` Move selected points outward in a spherical shape around the selected strokes' center.

Transform Fill Change the Translation, Rotation and scale of strokes fill.

Annotate Draw free-hand annotation.

> **Annotate Line** Draw straight line annotation.
>
> **Annotate Polygon** Draw a polygon annotation.
>
> **Annotate Eraser** Erase previous drawn annotations.

Grease Pencil Menu

Transform

Strokes can be edited by transforming the locations of points.

Move, Rotate & Scale

Reference

> **Mode** Edit Mode
>
> **Tool** *Toolbar → Move, Rotate, Scale*
>
> **Menu** *Grease Pencil → Transform → Move, Rotate, Scale*
>
> **Hotkey** `G`, `R`, `S`

Like other elements in Blender, points and strokes can be moved `G`, rotated `R` or scaled `S` as described in the *Basic Transformations* section. When in *Edit Mode*, *Proportional Editing* is also available for the transformation actions.

Transform Snapping

Basic move, rotate and scale transformations for selected points/strokes. See *Move, Rotate, Scale Basics* for more information.

Tools

Reference

> **Mode** Edit Mode
>
> **Menu** *Grease Pencil → Transform*
>
> **Tool** *Toolbar → Bend/Shear*

The *Bend*, *Shear*, *To Sphere*, *Extrude* and *Shrink Fatten* transform tools are described in the *Editing tools* section.

Mirror

Reference

Mode Edit Mode

Menu *Grease Pencil → Mirror*

Hotkey `Ctrl-M`

The *Mirror* tool is also available, behaving exactly the same as with *mesh vertices*.

Snap

Reference

Mode Edit Mode

Menu *Grease Pencil → Snap*

Hotkey `Shift-S`

Mesh snapping also works with Grease Pencil components.

Active Layer

Reference

Mode Edit Mode, Draw Mode

Menu *Grease Pencil → Active Layer*

Hotkey `Y`

Select the active layer.

Animation

Reference

Mode Edit Mode, Draw Mode

Menu *Grease Pencil → Animation*

Hotkey `I`

The stroke animation tools are described in the *Animation* section.

Interpolation

Reference

Mode Edit Mode, Draw Mode

Menu *Grease Pencil → Interpolation*

The stroke animation tools are described in the *Animation* section.

Duplicate

Reference

Mode Edit Mode

Menu *Grease Pencil → Duplicate*

Hotkey `Shift-D`

Duplicates the selected elements, without creating any connections with the rest of the strokes (unlike *Extrude*, for example), and places the duplicate at the location of the original elements.

Split

Reference

Mode Edit Mode

Menu *Grease Pencil → Split*

Hotkey `V`

Splits (disconnects) the selected points from the rest of the stroke. The separated points are left exactly at the same position as the original points but they belong to a new stroke.

Copy

Reference

Mode Edit Mode

Menu *Grease Pencil → Copy*

Hotkey `Ctrl-C`

Copy the selected points/strokes to the clipboard.

Paste & Paste by Layer

Reference

Mode Edit Mode

Menu *Grease Pencil → Paste, Grease Pencil → Paste by Layer*

Hotkey `Ctrl-V`

Type

Paste to Active Pastes the points/strokes copied from the clipboard into the active layer. This is the default behavior and the mode used when using *Grease Pencil → Paste*.

Paste by Layer Pastes the points/strokes copied from the clipboard into the layer they were copied from.

Separate Strokes

Reference

Mode Edit Mode

Menu *Grease Pencil → Separate Strokes*

Hotkey P

Separate the selected elements into a new Grease Pencil object.

Selected Points Separate the selected points into a new object.

Selected Strokes Separate the selected strokes into a new object. If one point of a stroke is selected, the entire stroke will be separated.

Active Layer Separate all the strokes in the active layer into a new object. See *2D Layers* for more information.

Clean Up

These tools help to cleanup degenerate geometry on the strokes.

Delete Loose Points

Reference

Mode Edit Mode

Menu *Grease Pencil → Clean Up → Delete Loose Points*

Removes unconnected points.

Delete Duplicate Frames

Reference

Mode Edit Mode

Menu *Grease Pencil → Clean Up → Delete Duplicated Frames*

Removes any duplicated animation frames.

Merge by Distance

Reference

Mode Edit Mode

Menu *Grease Pencil → Clean Up → Merge by Distance*

Merge by Distance is a useful tool to simplify a stroke by merging the selected points that are closer than a specified distance to each other. Note, unless using *Unselected*, selected points must be contiguous, else they will not be merged.

Merge Distance Sets the distance threshold for merging points.

Unselected Allows points in selection to be merged with unselected points. When disabled, selected points will only be merged with other selected ones.

Boundary Strokes

Reference

Mode Edit Mode

Menu *Grease Pencil → Clean Up → Boundary Strokes, Boundary Strokes All Frames*

Removes boundary strokes used by the *Fill* tool. See *Fill tool* for more information.

Mode

Active Frame Only Removes boundary strokes from the current frame.

All Frames Removes boundary strokes from all frames.

Reproject Strokes

Reference

Mode Edit Mode

Menu *Grease Pencil → Clean Up → Reproject Strokes*

Sometimes you may have drawn strokes unintentionally in different locations in the 3D space but they look right from a certain plane or from the camera view. You can use Reproject Strokes to flatten all the selected strokes from a certain viewpoint.

Reprojected Type

Front Reproject selected strokes onto the front plane (XZ).

Side Reproject selected strokes onto the side plane (YZ).

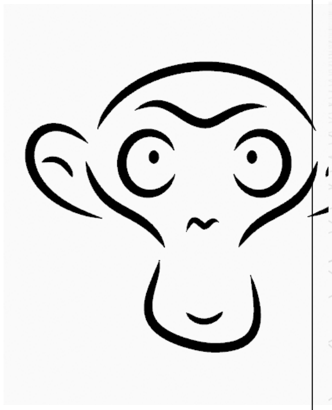
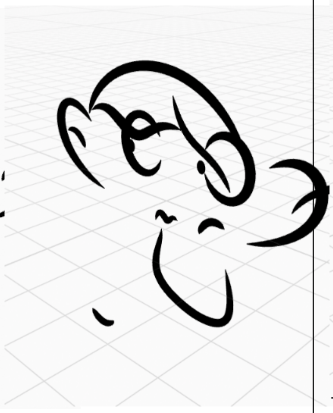
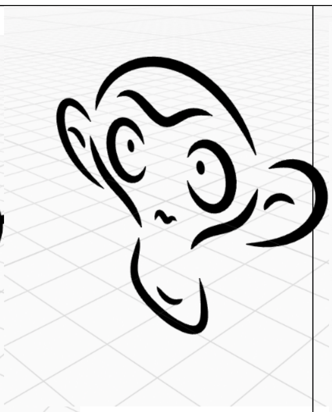
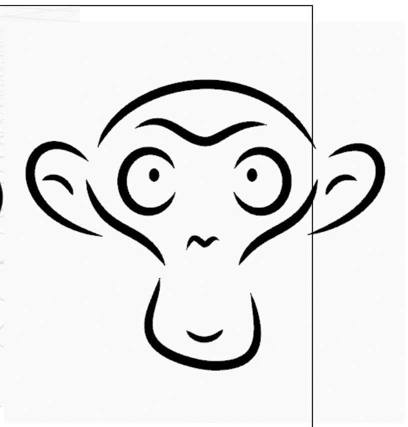
Top Reproject selected strokes onto the top plane (XY).

View Reproject selected strokes onto the current view.

Surface Reproject selected strokes onto the mesh surfaces.

Cursor Reproject selected strokes onto 3D cursor rotation.

Keep Original Maintains the original strokes after applying the tool.

 Fig. 1548: Original drawing from the front view.	 Fig. 1549: Original drawing in the 3D Viewport.	 Fig. 1550: Strokes reprojected onto the front plane to fix strokes misalignment.	 Fig. 1551: Drawing after reprojection operation from the front view.

Delete

Reference

Mode Edit Mode

Menu *Grease Pencil → Delete*

Hotkey `X`, `Delete`, `Ctrl-X`

Options for the Erase pop-up menu:

Points Deletes the selected points. When only one point remains, there is no more visible stroke, and when all points are deleted, the stroke itself is deleted.

Strokes Deletes all the strokes that selected points belongs to.

Frames Deletes all the strokes at the current frame and in the current layer/channel.

Dissolve `Ctrl-X` Deletes the selected points without splitting the stroke. The remaining points in the strokes stay connected.

Dissolve between `Ctrl-X` Deletes all the points between the selected points without splitting the stroke. The remaining points in the strokes stay connected.

Dissolve Unselect `Ctrl-X` Deletes all the points that are not selected in the stroke without splitting the stroke. The remaining points in the strokes stay connected.

Delete All Active Frames Deletes all the strokes at the current frame in all layers/channels.

Stroke Menu

This page covers many of the tools in the *Strokes* menu. These are tools that work primarily on strokes, however, some also work with point selections.

Subdivide

Reference

Mode Edit Mode

Menu *Stroke → Subdivide*

Subdivides the strokes by inserting points between the selected points.

Number of Cuts The number of subdivisions to perform.

Smooth The amount of the smoothness on subdivided points.

Repeat Number of times to repeat the procedure.

Selected Points When enabled, limits the effect to only the selected points within the stroke.

Position When enabled, the operator affect the points location.

Thickness When enabled, the operator affect the points thickness.

Strength When enabled, the operator affect the points strength (alpha).

UVs When enabled, the operator affect the UV rotation on the points.

Simplify

Reference

Mode Edit Mode

Menu *Stroke → Simplify*

Reduce the amount of points in the strokes.

Fixed Deletes alternated points in the strokes, except the start and end points.

Steps The number of times to repeat the procedure.

Adaptive Uses the RDP algorithm (Ramer-Douglas-Peucker algorithm) for points deletion. The algorithm tries to obtain a similar line shape with fewer points.

Factor Controls the amount of recursively simplifications applied by the algorithm.

Sample Recreates the stroke geometry with a predefined length between points.

Length The distance between points on the recreated stroke. Smaller values will require more points to recreate the stroke, while larger values will result in fewer points needed to recreate the curve.

Trim

Reference

Mode Edit Mode

Menu *Stroke → Trim*

Trims selected stroke to first loop or intersection.

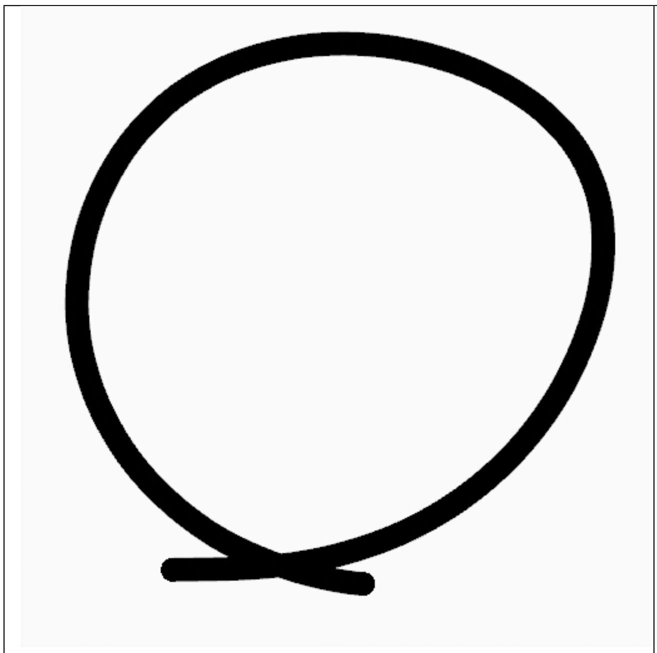


Fig. 1552: Original stroke.

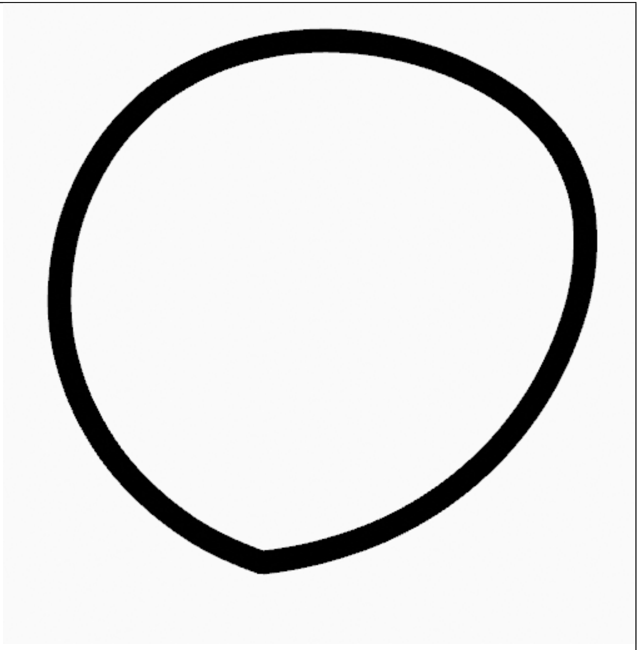


Fig. 1553: Result of trim operation.

Join

Reference

Mode Edit Mode

Menu *Stroke → Join → Join, Join and Copy*

Hotkey `Ctrl-J`, `Shift-Ctrl-J`

Join two or more strokes into a single one.

Type

Join `Ctrl-J` Join selected strokes by connecting points.

Join and Copy `Shift-Ctrl-J` Join selected strokes by connecting points in a new stroke.

Leave Gaps When enabled, do not use geometry to connect the strokes.

Move to Layer

Reference

Mode Edit Mode

Menu *Stroke → Move to Layer*

Hotkey `M`

A pop-up menu to move the stroke to a different layer. You can choose the layer to move the selected strokes to from a list of layers of the current Grease Pencil object. You can also add a new layer to move the selected stroke to.

Assign Material

Reference

Mode Edit Mode

Menu *Stroke → Assign Material*

Changes the material linked to the selected stroke. You can choose the name of the material to be used by the selected stroke from a list of materials of the current Grease Pencil object.

Set as Active Material

Reference

Mode Edit Mode

Menu *Stroke → Set as Active Material*

Sets the active object material based on the selected stroke material.

Arrange Strokes

Reference

Mode Edit Mode

Menu *Stroke → Arrange Strokes*

Change the drawing order of the strokes in the 2D layer.

Bring to Front Moves to the top the selected points/strokes.

Bring Forward Moves the selected points/strokes upper the next one in the drawing order.

Send Backward Moves the selected points/strokes below the previous one in the drawing order.

Send to Back Moves to the bottom the selected points/strokes.

Close

Reference

Mode Edit Mode

Menu *Stroke → Close*

Hotkey F

Close or open strokes by connecting the last and first point.

Type

Close All Close all open selected strokes.

Open All Open all closed selected strokes.

Toggle Close or Open selected strokes as required.

Create Geometry When enabled, points are added for closing the strokes. If disabled, the operator act the same as *Toggle Cyclic*.

Toggle Cyclic

Reference

Mode Edit Mode

Menu *Stroke → Toggle Cyclic*

Toggles between an open stroke and closed stroke (cyclic).

Type

Close All Close all open selected strokes.

Open All Open all closed selected strokes.

Toggle Close or Open selected strokes as required.

Create Geometry When enabled, points are added for closing the strokes like when using the *Close* tool. If disabled, the stroke is close without any actual geometry.

Toggle Caps

Reference

Mode Edit Mode

Menu *Stroke → Toggle Caps*

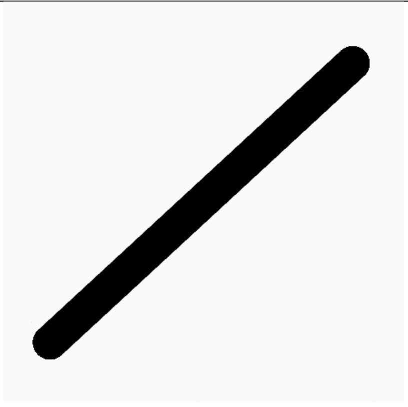
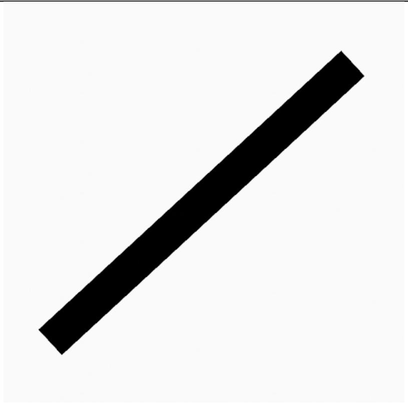
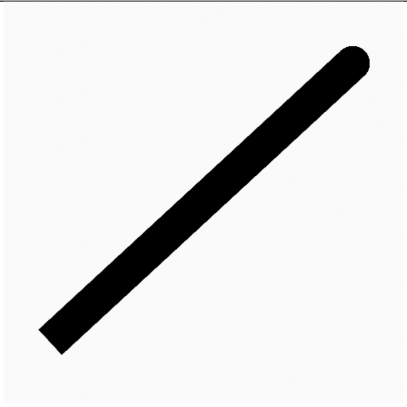
Toggle ending cap styles of the stroke.

Default Sets stroke start and end points to rounded (default).

Both Toggle stroke start and end points caps to flat or rounded.

Start Toggle stroke start point cap to flat or rounded.

End Toggle stroke end point cap to flat or rounded.

 Fig. 1554: Stroke ending with rounded caps.	 Fig. 1555: Stroke ending with flat caps.	 Fig. 1556: Stroke ending with combined caps.

Switch Direction

Reference

Mode Edit Mode

Menu *Stroke → Switch Direction*

Reverse the direction of the points in the selected strokes (i.e. the start point will become the end one, and vice versa).

Scale Stroke Thickness

Reference

Mode Edit Mode

Menu *Stroke → Scale Stroke Thickness*

When enabled, scales the stroke thickness during scale transformations.

Reset Fill Transform

Reference

Mode Edit Mode

Menu *Stroke → Reset Fill Transform*

Reset all fill translation, scaling and rotations in the selected strokes.

Point Menu

Extrude Points

Reference

Mode Edit Mode

Menu *Point → Extrude Points*

Tool *Toolbar → Extrude*

Hotkey `E`

Extrudes points by duplicating the selected points, which then can be moved. The new points stay connected with the original points of the edit line.

Note: Since Grease Pencil strokes can only have one start an end point, a new stroke will be created when extrude intermediate points in the strokes.

Smooth Points

Reference

Mode Edit Mode

Menu *Point → Smooth Points*

Softens strokes by reducing the differences in the locations of the points along the line, while trying to maintain similar values that make the line fluid and smoother.

Repeat The number of times to repeat the procedure.

Factor The amount of the smoothness to apply.

Selected Points When enabled, limits the effect to only the selected points within the stroke.

Position When enabled, the operator affect the points location.

Thickness When enabled, the operator affect the points thickness.

Strength When enabled, the operator affect the points strength (alpha).

UVs When enabled, the operator affect the UV rotation on the points.

Merge Points

Reference

Mode Edit Mode

Menu *Point → Merge Points*

Combine all selected points into a unique stroke. All the selected points will be connected by new edit lines when needed to create the new stroke.

Vertex Paint Mode

Introduction

Vertex Painting is a simple way of painting color onto a Grease Pencil object, by directly manipulating the color of points/vertices, rather than use only the materials base color.

When a point is painted, the color of the points is mixing with the base material color according to the settings of the brush.

Note: A vertex in Grease Pencil is called point. Point and vertex names are equivalent.

Vertex Paint Mode

Vertex Paint Mode is selected from the *Mode* menu in the 3D Viewport header. Once Vertex Paint Mode is activated, the Toolbar of the 3D Viewport will change to Vertex Paint Mode specific panels.

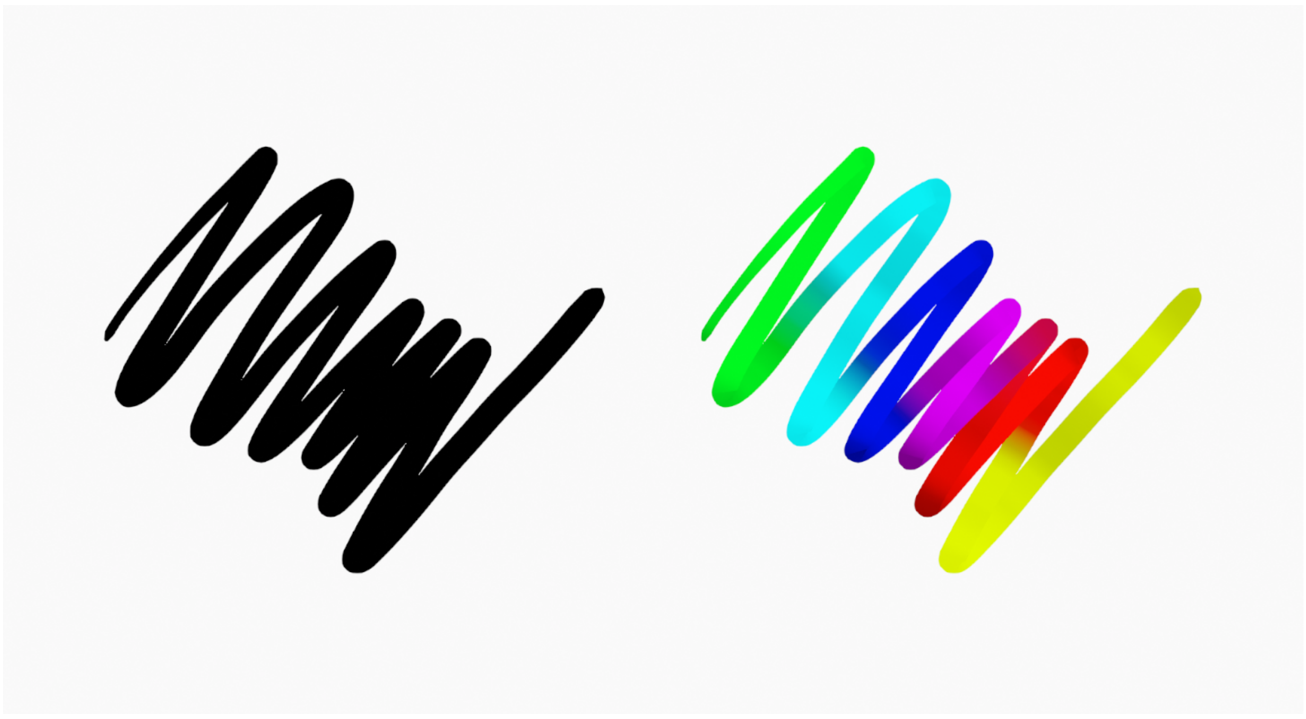


Fig. 1557: Stroke with original base material color (left) and with vertex painting (right).

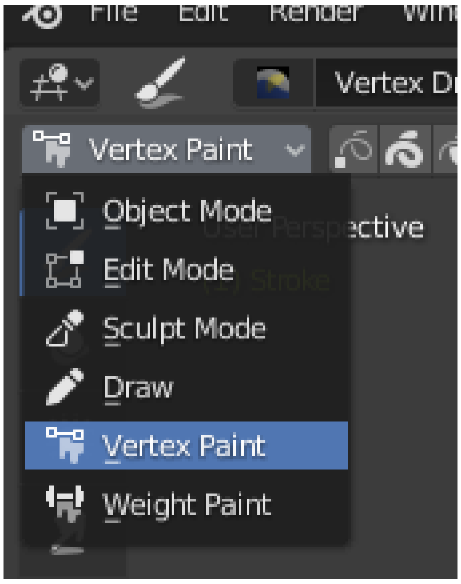


Fig. 1558: 3D Viewport Mode selector set to Vertex Paint Mode.

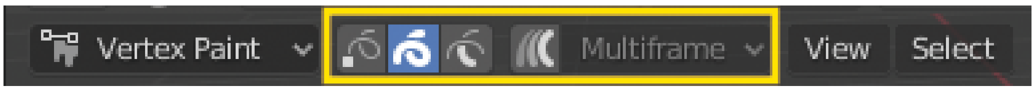


Fig. 1559: General Vertex Paint options.

Vertex Paint Options

Selection Mask Vertex Paint Mode in Grease Pencil allows you to select points or strokes to restrict the effect of the painting tools to only a certain areas of your drawing.

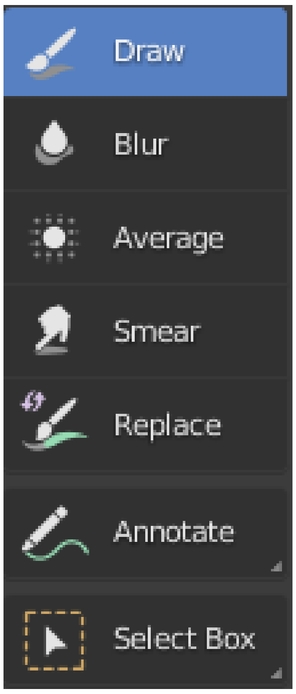
You can use the selection tools in the Toolbar for a quick selections.

You can restrict painting only on the selected points or strokes with the Selection mode toggle.

Multiframe Sometimes you may need to modify several frames at the same time with the painting tools.

You can activate multiframe edition with the Multiframe button next to the modes selector (faded lines icon). See *Multiframe* for more information.

Vertex Paint Tools



Draw Paints a specified color over the object.

Blur Smooths out the colors of adjacent vertices. In this mode the Color Value is ignored. The strength defines how much the colors are blurred.

Average Smooths color by painting the average resulting color from all colors under the brush.

Smear Smudges colors by grabbing the colors under the brush and "dragging" them. This can be imagined as a finger painting tool.

Replace Change the color only to the stroke points that already have vertex color applied.

Annotate Draw free-hand annotation.

Annotate Line Draw straight line annotation.

Annotate Polygon Draw a polygon annotation.

Annotate Eraser Erase previous drawn annotations.

Brush Settings

Painting needs paint brushes and Blender provides a Brush panel within the Toolbar when in *Weight Paint Mode*.

Brush In the *Data-Block menu* you find predefined Brush presets. And you can create your own custom presets as needed.

Radius This option controls the radius of the brush, measured in pixels. `F` allows you to change the brush size interactively by dragging the mouse and then `LMB` (the texture of the brush should be visible inside the circle). Typing a number then enter while using `F` allows you to enter the size numerically.

> **Pressure** Brush size can be affected by enabling the pressure sensitivity icon, if you are using a *Graphics Tablet*.

Strength How powerful the brush is when applied.

> **Pressure** Strength can be affected by enabling the pressure sensitivity icon, if you are using a *Graphics Tablet*.

Mode

> **Stroke** Only paint over strokes.
>
> **Fill** Only paint over fill areas.
>
> **Stroke and Fill** Paint over strokes and fill areas.

Cursor See the global brush settings for *Cursor* settings.

Color Picker

The color of the brush. See *Color Picker*.

Note: Note that Vertex Paint works in sRGB *space* and the RGB representation of the same colors will be different between the paint tools and the materials that are in linear space.

Color Palette

The active Color Palette. See *Color Palette*.

Falloff

See the global brush settings for *Falloff* settings.

Weight Paint Mode

Introduction

Assigning weight to the points is primarily used for rigging strokes in cut-out animation, where the vertex groups are used to define the relative bone influences on the strokes. See *Using Vertex Group* for more information.

Note: A vertex in Grease Pencil is called point. Point and vertex names are equivalent.

Weight Painting is a method to maintain large amounts of weight information in an intuitive way. The selected Grease Pencil object is displayed slightly shaded with a rainbow color spectrum. The color visualizes the weights associated to each point in the active vertex group. By default blue means unweighted and red means fully weighted.

You assign weights to the points of the object by painting on it with weight brushes. Starting to paint on a strokes automatically adds weights to the active vertex group (a new vertex group is created if needed).

Weight Paint

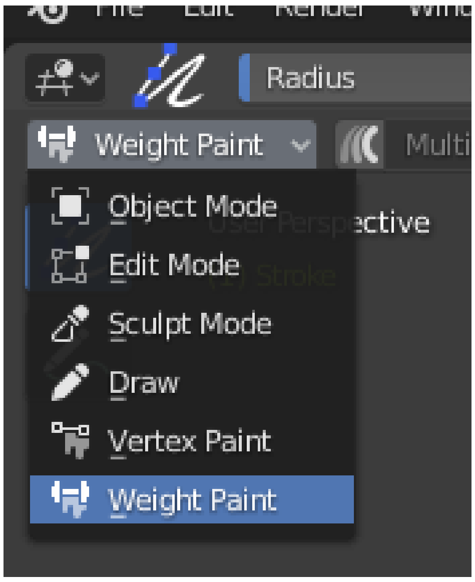


Fig. 1560: 3D Viewport Mode selector: Weight Paint Mode.

Weight Paint Mode is selected from the *Mode* menu in the 3D Viewport header. Once Weight Paint Mode is activated, the Toolbar of the 3D Viewport will change to Weight Paint Mode specific panels. A red circle will appear and follow the location of the cursor in the 3D Viewport.

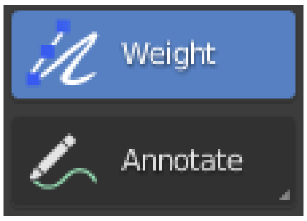
Weight Options

Multiframe Sometimes you may need to assign weight to several frames at the same time with the Weight Paint tools.

You can activate multiframe edition with the Multiframe button next to the modes selector (faded lines icon). See *Multiframe* for more information.

Weight Paint Tools

For Grease Pencil Weight Paint modes each brush type is exposed as a tool, the brush can be changed in the Tool Settings. See *Brush* for more information.



Draw Paints a specified weight over the strokes.

Annotate Draw free-hand annotation.

> **Annotate Line** Draw straight line annotation.
>
> **Annotate Polygon** Draw a polygon annotation.
>
> **Annotate Eraser** Erase previous drawn annotations.

Brush

Painting needs paint brushes and Blender provides a Brush panel within the Toolbar when in *Weight Paint Mode*.

Brush In the *Data-Block menu* you find predefined Brush presets. And you can create your own custom presets as needed.

Radius The radius defines the area of influence of the brush.

Strength This is the amount of paint to be applied per brush stroke.

Use Falloff When enabled, use Strength falloff for the brush. Brush Strength decays with the distance from the center of the brush.

Weight The weight (visualized as a color) to be used by the brush.

Weights Menu

Reference

> **Mode** Edit Mode
>
> **Menu** *Weights*

This page covers many of the tools in the *Weights* menu.

Normalize All

Reference

> **Mode** Edit Mode
>
> **Menu** *Weights → Normalize All*

For each point, this tool makes sure that the sum of the weights across all vertex groups is equal to 1. It normalizes all of the vertex groups, except for locked groups, which keep their weight values untouched.

Lock Active Keep the values of the active group while normalizing all the others.

Normalize

Reference

> **Mode** Edit Mode
>
> **Menu** *Weights → Normalize*

This tool only works on the active vertex group. All points keep their relative weights, but the entire set of weights is scaled up such that the highest weight value is 1.0.

Invert

Reference

Mode Edit Mode

Menu *Weights → Invert*

Replaces each weight of the selected vertex group by × -1.0 weight.

Examples:

- Original 1.0 converts to 0.0
- Original 0.5 remains 0.5
- Original 0.0 converts to 1.0

Subset Restrict the tool to a subset. See *The Subset Option* about how subsets are defined.

Add Weights Add vertices that have no weight before inverting (these weights will all be set to 1.0).

Remove Weights Remove vertices from the vertex group if they are 0.0 after inverting.

Smooth

Reference

Mode Edit Mode

Menu *Weights → Smooth*

Smooths the weights of the active vertex group.

Generate Weights

Reference

Mode Edit Mode

Menu *Weights → Generate Weights*

Generate automatic weight for armatures (requires an Armature modifier).

With Empty Group When parenting it will create an empty vertex groups on the child objects (if they do not exist already) for and named after each deforming bone in the armature.

With Automatic Weights Works similar to *With Empty Groups*, but it will not leave the vertex groups empty. It calculates how much influence a particular bone would have on points based on the distance from those points to a particular bone ("bone heat" algorithm). This influence will be assigned as weights in the vertex groups.

Object Mode

Convert to Geometry

Reference

Mode Object Mode

Menu *Object → Convert To → Path, Bézier Curve, Polygon Curve*

In the 3D Viewport, sketches on the active layer can be converted to geometry, based on the current view settings, by transforming the points recorded when drawing (which make up the strokes) into 3D space. Currently, all points will be used, so it may be necessary to simplify or subdivide parts of the created geometry for standard use. Sketches can currently be converted into curves in Object Mode.

Options

Type The type of object to convert to.

Path Create NURBS 3D curves of order 2 (i.e. behaving like polylines).

Bézier Curve Create Bézier curves, with free "aligned" handles (i.e. also behaving like polylines).

Polygon Curve Bézier curve with straight line segments (auto handles).

Note: Converting to Mesh

If you want to convert your sketch to a mesh, simply choose *NURBS* first, and then convert the created curve to a mesh.

Bevel Depth The *Bevel Depth* to use for the converted curve object.

Bevel Resolution The *Bevel Resolution* to use for the converted curve object.

Normalize Weight Will scale weights value so that they fit into the (0.0 to 1.0) range.

Radius Factor Multiplier for the points' radii (set from the stroke's width).

Link Strokes Will create a single spline, i.e. curve element, from all strokes in active Grease Pencil layer. This is especially useful if you want to use the curve as a path. All the strokes are linked in the curve by "zero weights/radii" sections.

Timing

Grease Pencil stores "dynamic" data, i.e. how fast strokes are drawn. When converting to curve, this data can be used to create an *Evaluate Time* F-curve (in other words, a path animation), that can be used e.g. to control another object's position along that curve (*Follow Path* constraint, or, through a driver, *Curve* modifier). So this allows you to reproduce your drawing movements.

Link Strokes has to be enabled for all timing options.

Timing Mode This control lets you choose how timing data is used.

No Timing Just create the curve, without any animation data (hence all following options will be hidden).

Linear The path animation will be a linear one.

Original The path animation will reflect to original timing, including for the "gaps" (i.e. time between strokes drawing).

Custom Gaps The path animation will reflect to original timing, but the "gaps" will get custom values. This is especially useful if you want to shorten large pauses between some strokes.

Frame Range The "length" of the created path animation, in frames. In other words, the highest value of *Evaluation Time*.

Start Frame The starting frame of the path animation.

Realtime When enabled, the path animation will last exactly the same duration it has taken you to draw the strokes.

End Frame When *Realtime* is disabled, this defines the end frame of the path animation. This means that the drawing timing will be adjusted to fit into the specified range.

Gap Duration *Custom Gaps* only. The average duration (in frames) of each gap between strokes. Please note that, the value will only be exact if *Realtime* is enabled, otherwise it will be scaled, exactly as the strokes' timing is.

Example

Here is a simple "hand writing" video created with curves converted from sketch data:

A video can be found at https://www.youtube.com/watch?v=VwWEXrnQAFI

The blend-file from the above example can be found here.

Trace Images to Grease Pencil

Reference

Mode Object Mode

Menu *Object → Trace Images to Grease Pencil*

The *Trace Images to Grease Pencil* tool traces a black and white image and generates Grease Pencil strokes. If the image is not black and white, it will be internally converted. For better results, convert the images manually to black and white. Also try to keep the resolution of the image small; high resolutions can produce very dense strokes.

Usage

1. Add an *Image Empty* to the scene.
2. Run *Trace Images to Grease Pencil*.

Options

Target Object The Grease Pencil object name. If left empty, a new Grease Pencil object will be created.

Thickness The thickness of the generated Grease Pencil strokes.

Resolution Resolution of the generated Grease Pencil strokes i.e. how many control points the stroke will have. A higher resolution will preserve more detail from the original image.

Scale The object scale applied to the generated Grease Pencil object.

Sample Recreates the stroke geometry with a predefined length between points. Smaller values will require more points to recreate the stroke, while larger values will result in fewer points needed to recreate the curve. A value of 0 disables stroke sampling.

Color Threshold Determines what is considered white and what black.

Turn Policy Determines how to resolve ambiguities during decomposition of an image into paths.

> **Black** Prefers to connect black (foreground) components.
>
> **White** Prefers to connect white (background) components.
>
> **Left** Always take a left turn.
>
> **Right** Always take a right turn.
>
> **Minority** Prefers to connect the color (black or white) that occurs least frequently in the local neighborhood of the current position.
>
> **Majority** Prefers to connect the color (black or white) that occurs most frequently in the local neighborhood of the current position.
>
> **Random** Choose pseudo-randomly.

Mode Determines if the image being traced is an single image or image sequence.

> **Single** The image empty is a single image or the current frame of an image sequence.
>
> **Sequence** The image empty is an *Image Sequence*.

2.8 Animation & Rigging

2.8.1 Introduction

Animation

Animation is making an object move or change shape over time. Objects can be animated in many ways:

Moving as a whole object Changing their position, orientation or size in time;

Deforming them Animating their vertices or control points;

Inherited animation Causing the object to move based on the movement of another object (e.g. its parent, hook, armature, etc.).

In this chapter, we will cover the first two, but the basics given here are actually vital for understanding the following chapters as well.

Animation is typically achieved with the use of *keyframes*.

See also:

Related Sections

- *Physical Simulation*
- *Motion Tracking*

State Colors

Properties have different colors and menu items for different states.

Gray	Not animated
Yellow	Keyframed on the current frame
Green	Keyframed on a different frame
Orange	Changed from the keyframed value
Purple	Controlled by a driver

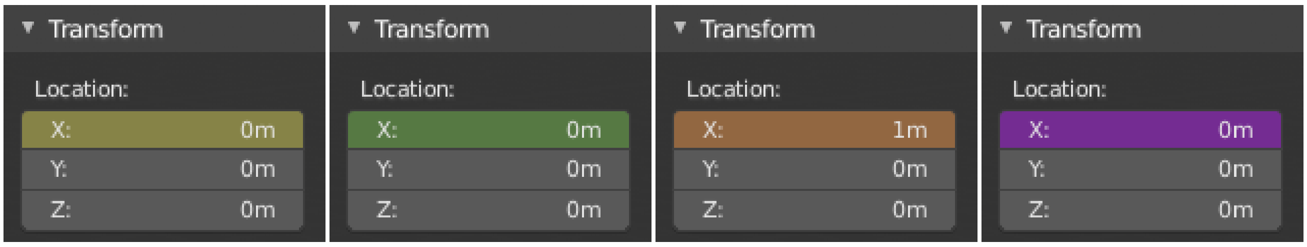


Fig. 1561: State colors of properties.

The changed value highlight currently doesn't work with *NLA*.

Rigging

Rigging is a general term used for adding controls to objects, typically for the purpose of animation.

Rigging often involves using one or more of the following features:

Armatures This allows mesh objects to have flexible joints and is often used for skeletal animation.

Constraints To control the kinds of motions that make sense and add functionality to the rig.

Object Modifiers Mesh deformation can be quite involved, there are multiple modifiers that help control this.

Shape Keys To support different target shapes *(such as facial expressions)* to be controlled.

Drivers So your rig can control many different values at once, as well as making some properties automatically update based on changes elsewhere.

Rigging can be as advanced as your project requires, rigs are effectively defining own user interface for the animator to use, without having to be concerned the underlying mechanisms.

Examples

- An armature is often used with a modifier to deform a mesh for character animation.
- A camera rig can be used instead of animating the camera object directly to simulate real-world camera rigs (with a boom arm, mounted on a rotating pedestal for example, effects such as camera jitter can be added too).

See also:

The content of this chapter is simply a reference to how rigging is accomplished in Blender. It should be paired with additional resources such as Nathan Vegdahl's excellent introduction to the fundamental concepts of character rigging, Humane Rigging.

2.8.2 Keyframes

Introduction

A *Keyframe* is simply a marker of time which stores the value of a property.

For example, a Keyframe might define that the horizontal position of a cube is at 3 m on frame 1.

The purpose of a Keyframe is to allow for interpolated animation, meaning, for example, that the user could then add another key on frame 10, specifying the cube's horizontal position at 20 m, and Blender will automatically determine the correct position of the cube for all the frames between frame 1 and 10 depending on the chosen interpolation method (e.g. Linear, Bézier, Quadratic, etc.).

An overview of existing keyframes can be seen via the *Dope Sheet* editor.

Visualization

There are some important visualization features in the 3D Views that can help animation.

When the current frame is a keyframe for the current active object, the name of this object (shown in the upper left corner of the 3D Views) turns yellow.

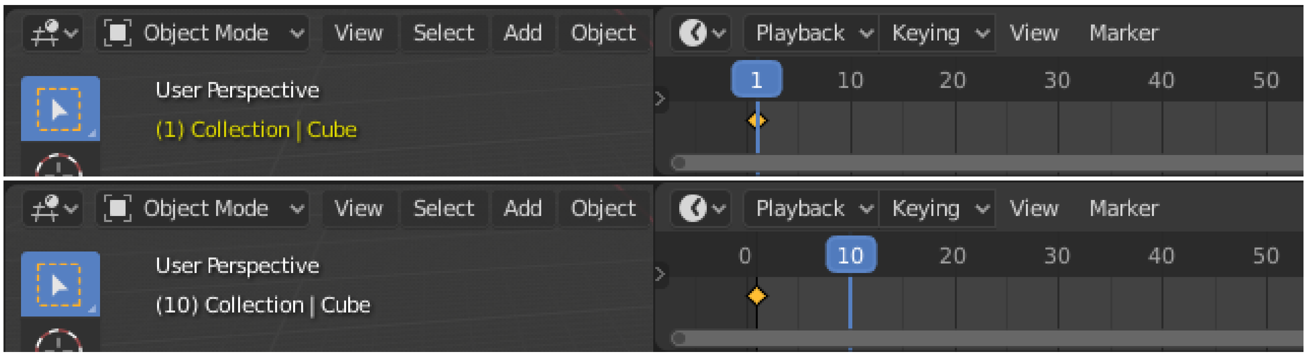


Fig. 1562: Top: Current frame is a keyframe for Cube. Bottom: Current frame isn't a keyframe.

Interpolation

Keyframe interpolation is represented and controlled by *animation curves*, also known as *F-Curves*. These curves can be viewed and modified via the *Graph Editor*.

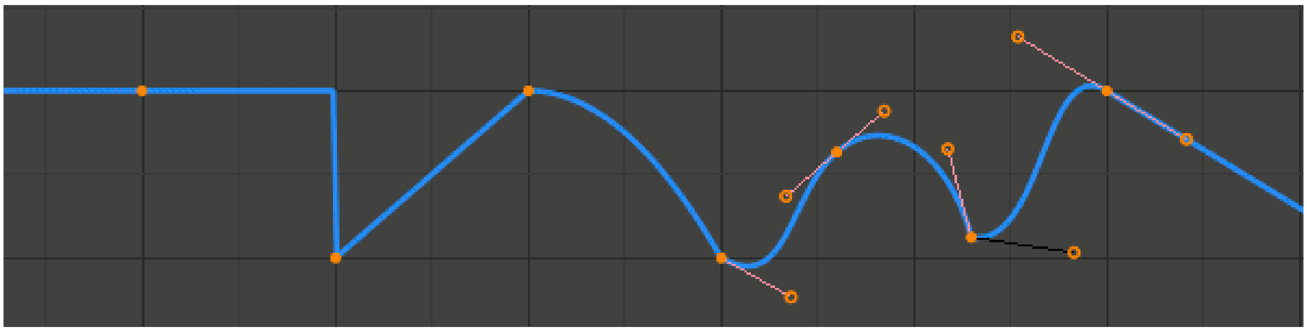


Fig. 1563: Constant, Linear, Quadratic and Bézier interpolation, with Linear extrapolation.

The X axis of the curve corresponds to time, while Y represents the value of the property. Keyframes themselves define points of the curve, while interpolation is controlled by additional parameters.

The *Interpolation Mode* is the main setting that specifies for each keyframe how the curve is interpolated from that key to the next one. There are a number of modes with fixed shapes, e.g. *Constant*, *Linear*, *Quadratic* etc, and a free form *Bézier* mode.

Extrapolation specifies how the curve extends before the first, and after the last keyframe. The main available choices are *Constant* and *Linear*; it is also possible to configure the curve to loop.

Bézier interpolation is controlled by handles, which have a *handle type* and position. The position of *Free* and *Aligned* handles must be set manually from the Graph editor, while *Vector*, *Automatic* and *Auto Clamped* handles are computed automatically from keyframe values.

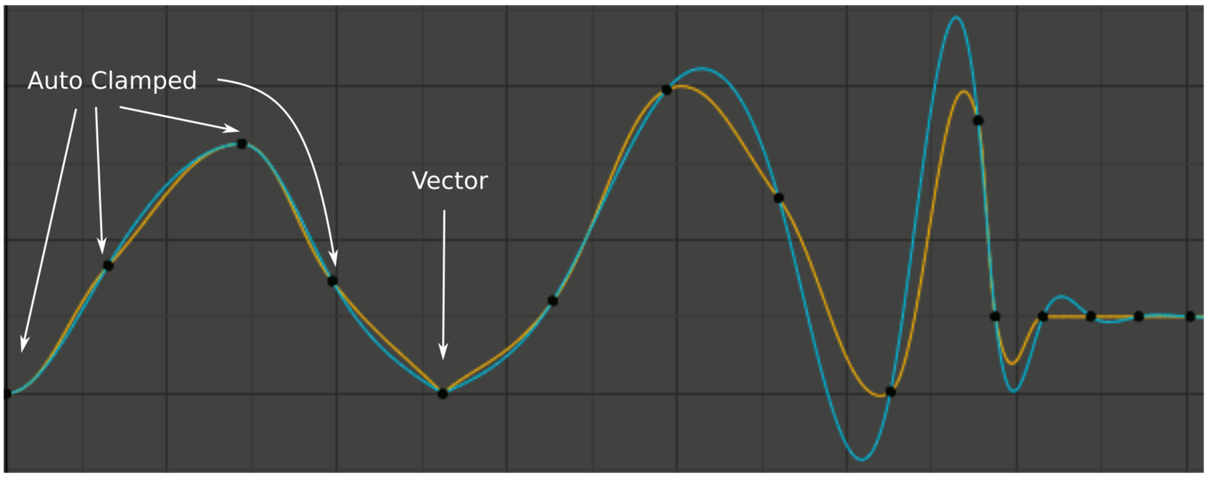


Fig. 1564: Handle smoothing modes. Yellow: *None*, Cyan: *Continuous Acceleration*.

Interpolation, Extrapolation and Handle Type can also be changed from the *Dope Sheet* editor.

The method how the three automatic handle types are computed is controlled by the per-curve *Auto Handle Smoothing* setting. The *None* mode resembles how most other software works and only considers the values of the immediately adjacent keys. The *Continuous Acceleration* mode considers the shape of the whole curve, which produces smoother results out of the box, but means that changes in one key affect interpolation over a larger section of the curve; it also tends to overshoot more with *Automatic* handles.

Keyframe Types

For visually distinguishing regular keyframes from different animation events or states (extremes, breakdowns, or other in-betweens) there is the possibility of applying different colors on them for visualization.

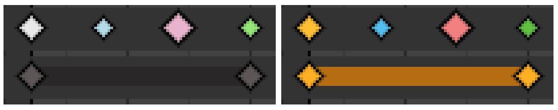


Fig. 1565: Left: not selected; Right: selected.

Keyframe (white / yellow diamond) Normal keyframe.

Breakdown (small cyan diamond) Breakdown state. e.g. for transitions between key poses.

Moving Hold (dark gray / orange diamond) A keyframe that adds a small amount of motion around a holding pose. In the Dope Sheet it will also display a bar between them.

Extreme (big pink diamond) An 'extreme' state, or some other purpose as needed.

Jitter (tiny green diamond) A filler or baked keyframe for keying on ones, or some other purpose as needed.

Handles & Interpolation Mode Display

Dope Sheet can display the Bézier handle type associated with the keyframe, and mark segments with non-Bézier interpolation. This facilitates basic editing of interpolation without the use of the Graph Editor.

The icon shape represents the type of the *Bézier Handles* belonging to the keyframe.

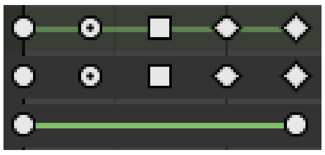


Fig. 1566: From top: summary, Bézier, linear.

Circle	Auto Clamped (default)
Circle With Dot	Automatic
Square	Vector
Clipped Diamond	Aligned
Diamond	Free

If the handles of a keyframe have different types, or in case of summary rows representing multiple curves, out of the available choices the icon that is furthest down the list is used. This means that if a grouped row uses a circle icon, it is guaranteed that none of the grouped channels have a non-auto key.

Horizontal green lines mark the use of non-Bézier *Interpolation*. The line is dimmed in summary rows if not all grouped channels have the same interpolation.

Display of this information can be disabled via the *Show Handles and Interpolation* option of the Dope Sheet's *View Menu*.

Editing

Insert Keyframe

Reference

Mode Object Mode

Menu *Object → Animation → Insert Keyframe…*

Hotkey I

There are several methods of adding new keys. Namely:

- In the 3D Viewport, pressing I will bring up a menu to choose what to add a keyframe to.
- Hovering over a property and pressing I or with the context menu by RMB a property and choose *Insert Keyframe* from the menu.

Auto Keyframe

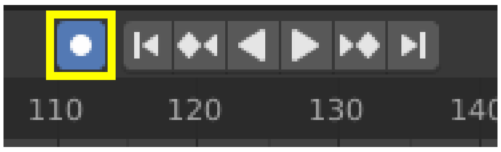


Fig. 1567: Timeline Auto Keyframe.

Auto Keyframe is the record button in the *Timeline* header. Auto Keyframe adds keyframes automatically to the set frame if the value for transform type properties changes.

See *Timeline Keyframe Control* for more info.

Delete Keyframes

Reference

Mode Object Mode

Menu *Object → Animation → Delete Keyframes…*

Hotkey `Alt-I`

There are several methods of removing keyframes:

- In the 3D Viewport press `Alt-I` to remove keys from selected objects on the current frame.
- When the mouse is over a value, press `Alt-I`.
- `RMB` a value and choose *Delete Keyframe* from the menu.

Clear Keyframes

Reference

Mode Object Mode

Menu *Object → Animation → Clear Keyframes…*

Removes all keyframes from the selected object.

Editing Keyframes

Keyframes can be edited in two editors. To do so go to either the *Graph Editor* or the *Dope Sheet*.

Examples

Keyframe Animation

This example shows you how to animate a cube's location, rotation, and scale.

1. First, in the *Timeline*, or other animation editors, set the frame to 1.
2. With the *Cube* selected in *Object Mode*, press `I` in the 3D Viewport.
3. From the *Insert Keyframe Menu* select *LocRotScale*. This will record the location, rotation, and scale, for the *Cube* on frame 1.
4. Set the frame to 100.
5. Use Move `G`, Rotate `R`, Scale `S`, to transform the cube.
6. Press `I` in the 3D Viewport. From the *Insert Keyframe Menu*, select *LocRotScale*.

To test the animation, press `Spacebar` to play.

Keying Sets

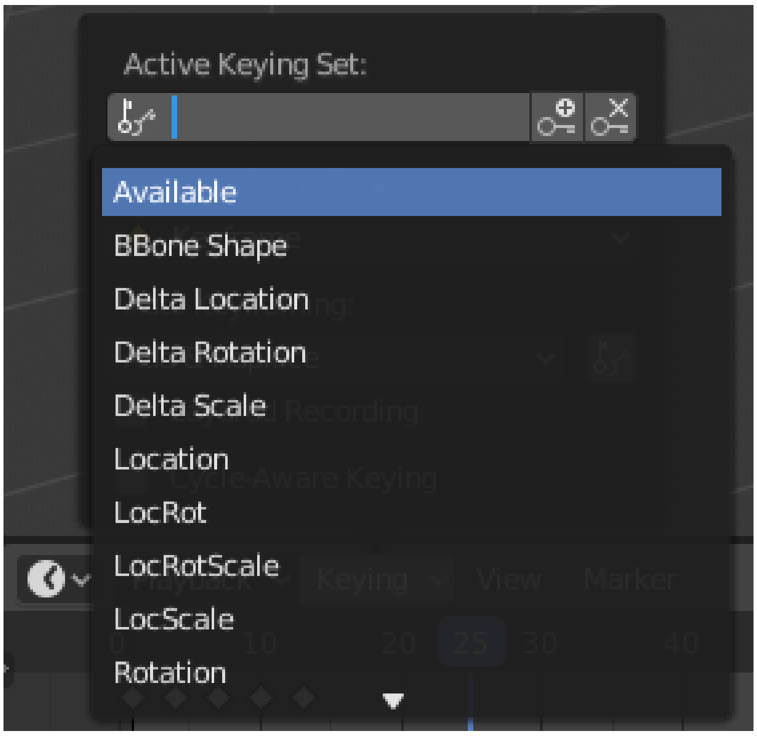


Fig. 1568: The Active Keying Sets data ID in the Timeline.

Keying sets are a collection of animated properties that are used to animate and keyframe multiple properties at the same time. For example, using keying sets you can press `I` in the 3D Viewport, Blender will add keyframes for all the properties in the active keying set. There are some built-in keying sets and, also custom keying sets called *Absolute Keying Sets*. To select and use a keying set, set the *Active Keying Set* in the *Keying popover* in the Timeline header, or the Keying Set panel, or press `Shift-Ctrl-Alt-I` in the 3D Viewport.

Keying Set Panel

Reference

Editor Properties

Panel *Scene* → *Keying Set*

This panel is used to add, select, manage *Absolute Keying Sets*.

Active Keying Set The *List View* of keying sets in the active scene.

Add + Adds an empty keying set.

Description A short description of the keying set.

Export to File Export keying set to a Python script `File.py`. To re-add the keying set from the `File.py`, open then run the `File.py` from the Text Editor.

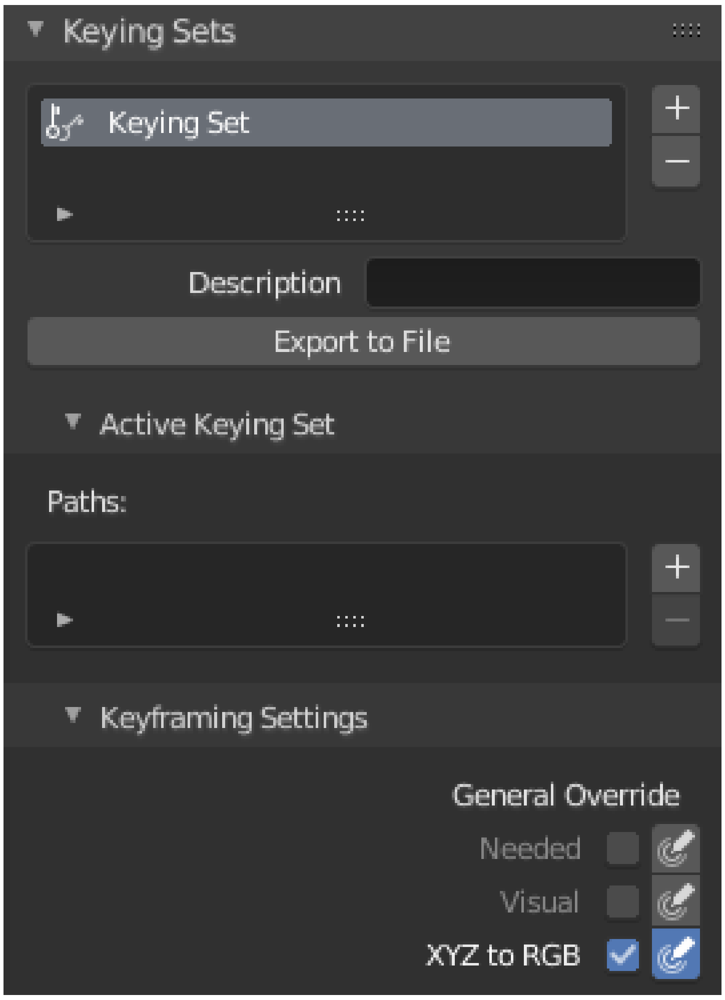


Fig. 1569: The Keying Set panel.

Keyframing Settings

General Override These options control all properties in the keying set. Note that the same settings in *Preferences* override these settings if enabled.

Active Set Override These options control individual properties in the keying set.

Common Settings

Only Needed Only insert keyframes where they are needed in the relevant F-curves.

Visual Keying Insert keyframes based on the visual transformation.

XYZ to RGB For new F-curves, set the colors to RGB for the property set, Location XYZ for example.

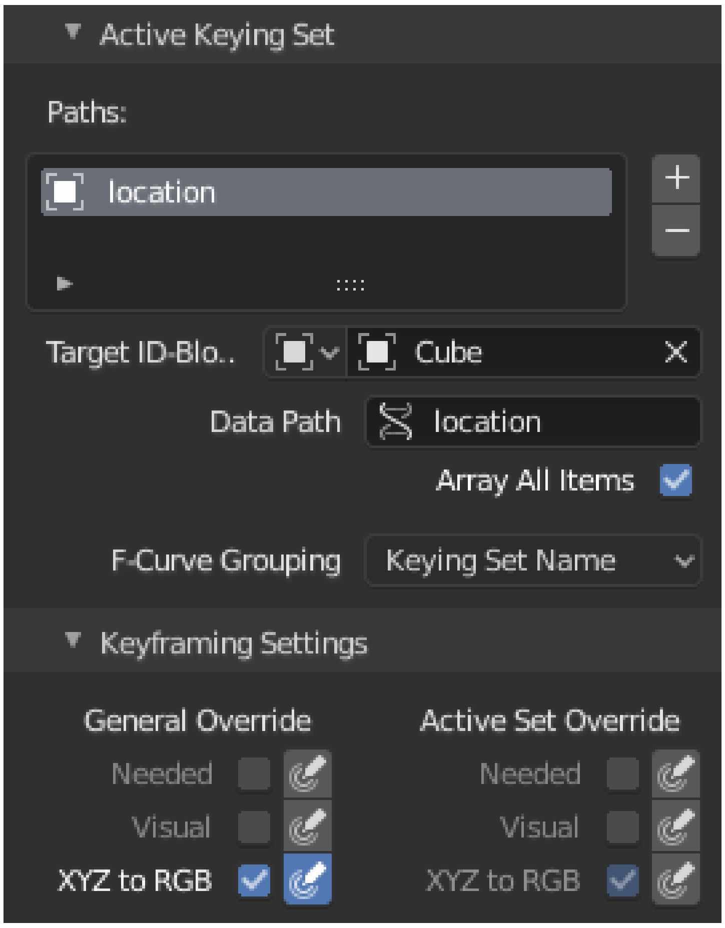
Active Keying Set Panel

Reference

Editor Properties

Panel *Scene → Active Keying Set*

This panel is used to add properties to the active keying set.



Target ID-Blo.. Cube
Data Path location
Array All Items ✓
F-Curve Grouping Keying Set Name

Fig. 1570: The Active Keying Set panel.

Paths A collection of paths in a *List View* each with a *Data Path* to a property to add to the active keying set.

Add + Adds an empty path.

Target ID-Block Set the ID Type and the *Object IDs* data path for the property.

Data Path Set the rest of the Data Path for the property.

Array All Items Use *All Items* from the Data Path or select the array index for a specific property.

F-Curve Grouping This controls what group to add the channels to.

Keying Set Name, None, Named Group

Adding Properties

Reference

Menu *Context menu → Add All/Single to Keying Set*

Hotkey `K`

Some ways to add properties to keying sets.

`RMB` the property in the *User Interface*, then select *Add Single to Keying Set* or *Add All to Keying Set*. This will add the properties to the active keying set, or to a new keying set if none exist.

Hover the mouse over the properties, then press `K`, to add *Add All to Keying Set*.

Whole Character Keying Set

The built-in *Whole Character* keying set is made to keyframe all properties that are likely to get animated in a character rig. It is also implicitly used by the *Pose Library system*.

In order to determine which bones to add keys for, and which bones to skip, the keying set uses the bone names. The following bone name prefixes will be skipped:

“COR”, “DEF”, “GEO”, “MCH”, “ORG”, “VIS”

2.8.3 Armatures

Introduction

An armature in Blender can be thought of as similar to the armature of a real skeleton, and just like a real skeleton an armature can consist of many bones. These bones can be moved around and anything that they are attached to or associated with will move and deform in a similar way.

An “armature” is a type of object used for rigging. A rig is the controls and strings that move a marionette (puppet). Armature object borrows many ideas from real-world skeletons.

Your First Armature

In order to see what we are talking about, let us try to add the default armature in Blender.

(Note that armature editing details are explained in the *armatures editing section*.)

Open a default scene, then:

1. Delete all objects in the scene.
2. Make sure the cursor is in the world origin with `Shift-C`.
3. Press `Numpad1` to see the world in Front view.
4. Add a *Single Bone* (*Add → Armature*).

5. Press `NumpadDelete` to see the armature at maximum zoom.

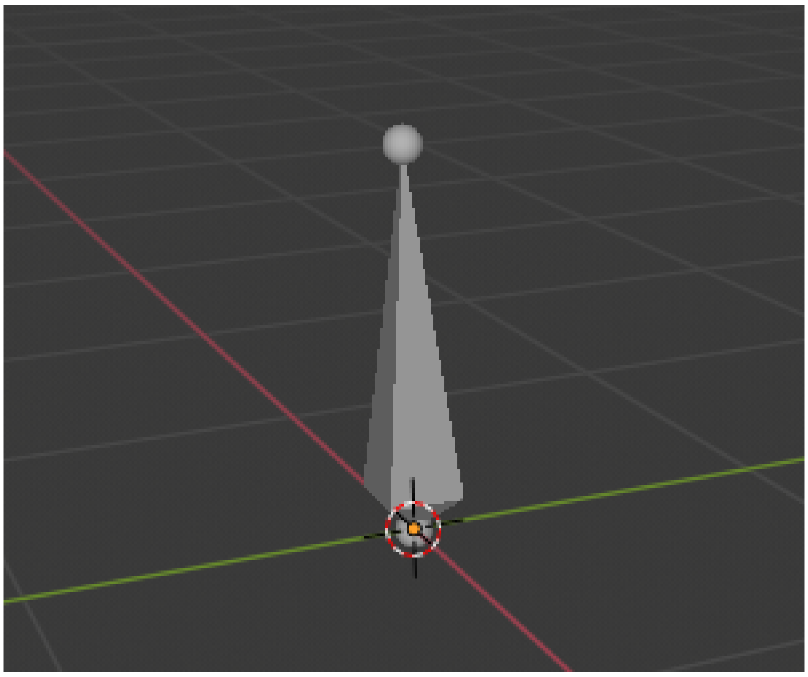


Fig. 1571: The default armature.

The Armature Object

As you can see, an armature is like any other object type in Blender:

- It has an origin, a position, a rotation and a scale factor.
- It has an Object Data data-block, that can be edited in *Edit Mode*.
- It can be linked to other scenes, and the same armature data can be reused on multiple objects.
- All animation you do in *Object Mode* is only working on the whole object, not the armature's bones (use the *Pose Mode* to do this).

As armatures are designed to be posed, either for a static or animated scene, they have a specific state, called "rest position". This is the armature's default "shape", the default position/rotation/scale of its bones, as set in *Edit Mode*.

In *Edit Mode*, you will always see your armature in rest position, whereas in *Object Mode* and *Pose Mode*, you usually get the current "pose" of the armature (unless you enable the *Rest Position* button of the *Armature* panel).

Bones

Introduction

Bones are the base elements of armatures. The visualization of bones can be set in the Armatures *Viewport Display Panel*.

Classification

Bones in an Armature can be generally classified into two different types:

1. Deforming Bones
2. Control Bones

Deforming Bones

Are bones which when transformed will result in vertices associated with them also transforming in a similar way. Deforming Bones are directly involved in altering the positions of vertices associated with their bones.

Control Bones

Are Bones which act in a similar way to switches, in that, they control how other bones or objects react when they are transformed. A Control Bone could for example act as a sliding switch control when the bone is in one position to the left, it could indicate to other bones that they react in a particular way when transformed, and when the Control Bone is positioned to the right, transforming other bones or objects could do something completely different. Control Bones are not directly used to alter the positions of vertices; in fact, Control Bones often have no vertices directly associated with themselves.

Structure

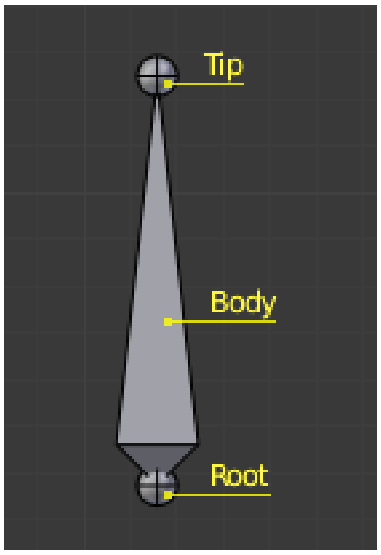


Fig. 1572: The elements of a bone.

They have three elements:

1. The "start joint" named *root* or *head*.
2. The "body" itself.
3. And the "end joint" named *tip* or *tail*.

With the default armature in Edit Mode, you can select the root and the tip, and move them as you do with mesh vertices.

Both root and tip (the "joints") define the bone by their respective position.

They also have a radius property, only useful for the envelope deformation method (see below).

Roll

Activating *Axes* checkbox on the *Armature tab* → *Display panel*, will show local axes for each bone's tip. The Y axis is always aligned along the bone, oriented from root to tip. So, this is the "roll" axis of the bones.

Bones Influence

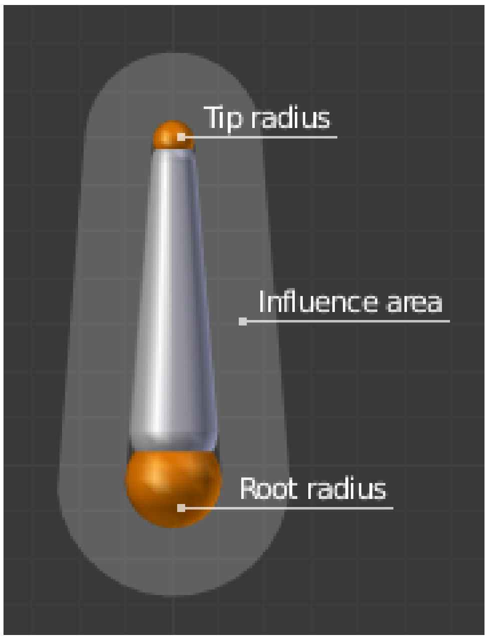


Fig. 1573: A bone in Envelope visualization, in Edit Mode.

Basically, a bone controls a geometry when vertices "follow" the bone. This is like how the muscles and skin of your finger follow your finger-bone when you move a finger.

To do this, you have to define the strength of *influences* a bone has on a certain vertex.

The simplest way is to have each bone affecting those parts of the geometry that are within a given range from it. This is called the *envelope technique*, because each bone can control only the geometry "enveloped" by its own influence area.

If a bone is visualized as *Envelope*, in *Edit Mode* and in *Pose Mode* you can see the area of influence, which depends on:

- The *distance* property and
- the root's radius and the tip's radius.

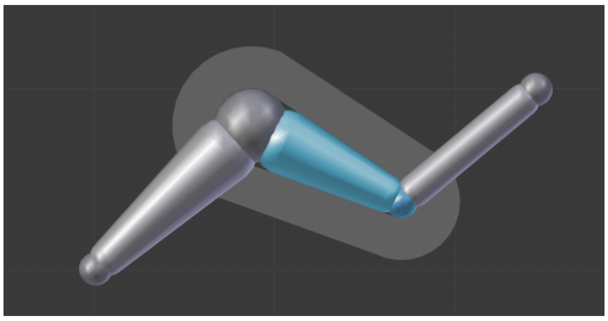


Fig. 1574: Our armature in Envelope visualization, in Pose Mode.

All these influence parameters are further detailed in the *skinning pages*.

Tools

Toolbar

Mesh Edit Mode tools:

Select Select or move.

> ***Select Box*** Select geometry by dragging a box.
>
> ***Select Circle*** Select geometry by dragging a circle.
>
> ***Select Lasso*** Select geometry by drawing a lasso.

Cursor Change the location of the 3D Cursor.

Move Translation tool.

Rotate Rotation tool.

Scale Scale tool.

> ***Scale Cage*** Change the scale of an object by controlling its cage.

Transform Tool to adjust the objects translation, rotations and scale.

Annotate Draw free-hand annotation.

> **Annotate Line** Draw straight line annotation.
>
> **Annotate Polygon** Draw a polygon annotation.
>
> **Annotate Eraser** Erase previous drawn annotations.

Measure Measure distances in the scene.

Roll Rotates a bone around its local Y axis.

Bone Size Todo.

> **Bone Envelope** Todo.

Extrude Creates a new bone connected to the last selected joint.

> **Extrude to Cursor** Creates a new bone between the last selected joint and the mouse position.

Shear Todo.

Tool Settings

Options

X-Axis Mirror

See *X-Axis Mirror Pose Mode*.

Selecting

You can select and edit bones of armatures in *Edit Mode* and in *Pose Mode*. Here, we will see how to select bones in *Edit Mode*. Selecting bones in *Pose Mode* is similar to selecting in *Edit Mode* with a few specific differences that will be detailed in the *posing part*.

Similar to *vertex/edge selection* in meshes, there are two ways to select whole bones in *Edit Mode*:

1. Directly, by selecting the bone's body.
2. Selecting both of its joints (root and tip).

This is an important point to understand, because selecting bones' joints only might lead to non-obvious behavior, with respect to which bone you actually select.

Note that unlike the mesh display type, the armature display type has no effect on selection behavior. In other words, you can select a bone's joint or body the same way regardless of the bone visualization chosen.

Selecting Bone Joints

To select bones' joints you have the *standard selection* methods.

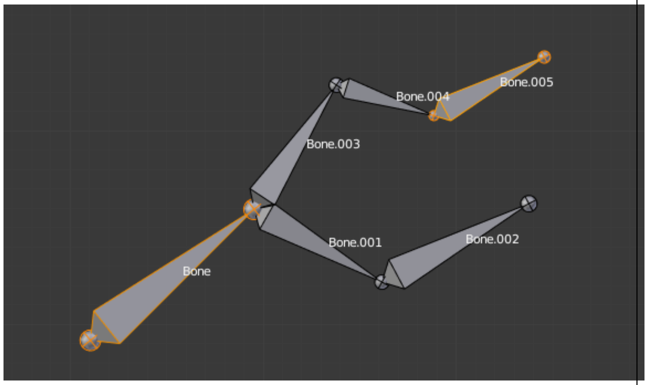
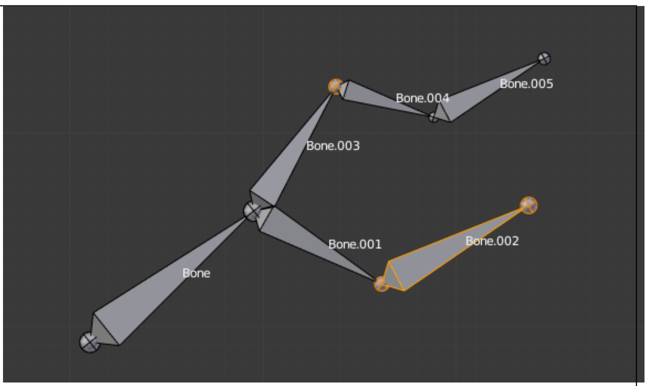
Inverse Selection

As stated above, you have to remember that these selection tools are for bones' joints only, not the bones' bodies.

For example, the *Inverse* selection option `Ctrl-I` inverts the selection of bones' joints, not of bones (see *Inverse selection*).

Remember that a bone is selected only if both its joints are selected. So, when the selection status of bones' joints is inverted, a new set of bones is selected.

Table 98: Inverse selection.

 Fig. 1575: Two bones selected.	 Fig. 1576: The result of the inverse selection `Ctrl-I`: The bones joints selection has been inverted, and not the bones selection.

Selecting Connected Bone Joints

Another example is: when you select the root of a bone connected to its parent, you also implicitly select the tip of its parent (and vice versa).

Note: Remember that when selecting bones' joints, the tip of the parent bone is the "same thing" as the root of its children bones.

Selecting Bones

By clicking on a bone's body, you will select it (and hence you will implicitly select its root and tip).

Using `Shift`-click, you can add to/remove from the selection.

You also have some *advanced selection* options, based on their relations.

Pick Shortest Path `Ctrl-click` Selects the path from the active bone to the bone under the mouse.

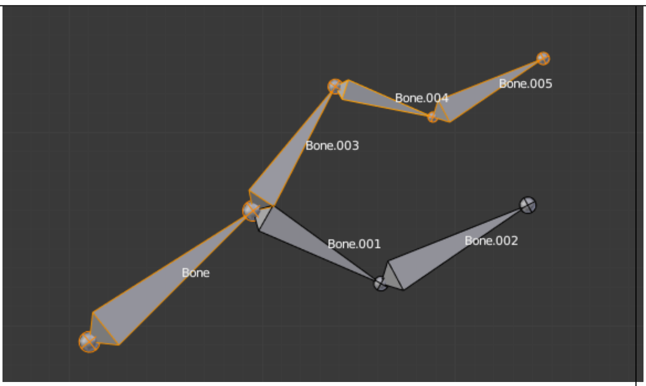
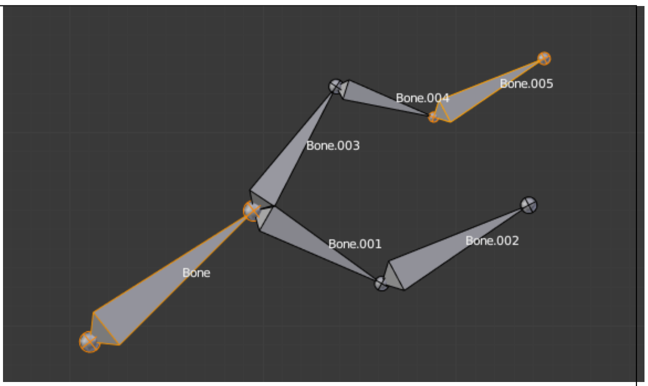
Deselecting Connected Bones

There is a subtlety regarding connected bones.

When you have several connected bones selected, if you deselect one bone, its tip will be deselected, but not its root, if it is also the tip of another selected bone.

To understand this, look at Fig. *Bone deselection in a selected chain.*.

Table 99: Bone deselection in a selected chain.

 Fig. 1577: A selected chain.	 Fig. 1578: Two selected bones.

After `Shift`-clicking "Bone.003":

- "Bone.003" 's tip (which is same as "Bone.004" 's root) is deselected.
- "Bone" is "Bone.003" 's parent. Therefore "Bone.003" 's root is same as the tip of "Bone". Since "Bone" is still selected, its tip is selected. Thus the root of "Bone.003" remains selected.

Mirror

Reference

Mode Edit Mode

Menu *Select → Mirror*

Hotkey `Shift-Ctrl-M`

Flip the selection from one side to another.

More/Less

Reference

Mode Edit Mode

Menu *Select → More/Less*

More `Ctrl-NumpadPlus` Expand the current selection to the connected bones.

Less `Ctrl-NumpadMinus` Contrast the selection, deselect bones at the boundaries of each selection region.

Linked

Reference

Mode Edit Mode

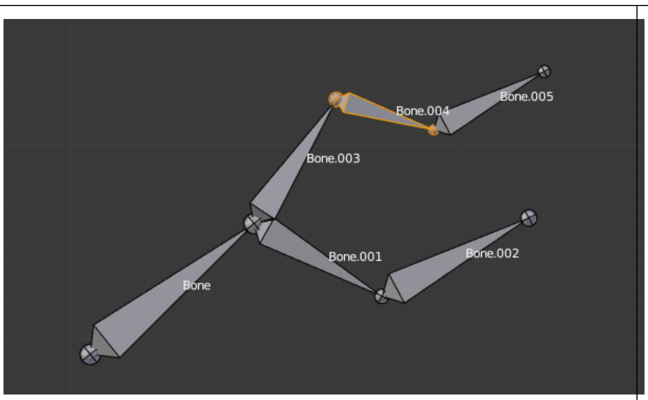
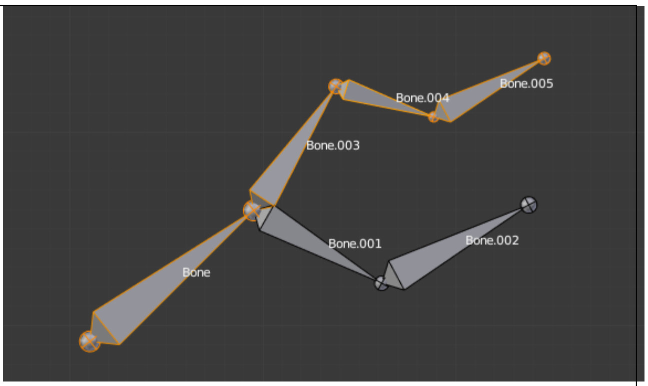
Menu *Select → Linked*

Hotkey `Ctrl-L`

Selects all the bones in the chain which the active (last selected) bone belongs to.

All Forks Selects all bones connected to the active bone even if the branch off from the current bone.

Table 100: Linked bones selection.

 Fig. 1579: A single selected bone.	 Fig. 1580: Its whole chain selected with Linked.

Parent/Child

Parent `[`**, Child** `]` You can deselect the active bone and select its immediate parent or one of its children.

Extend Parent/Child

Extend Parent `Shift-[`**, Extend Child** `Shift-]` Similar to *Parent/Child* but it keeps the active bone in the selection.

Similar

Reference

Mode Edit Mode

Menu *Select → Similar*

Hotkey `Shift-G`

Children Extends the selection to all hierarchical descendant bones.

Immediate Children Extends the selection to all direct child bones.

Siblings Selects bones that have the same parent as the active bone.

Length Selects bones with a similar bone length (between tip and tail) under the specified *Threshold*.

Direction (Y axis) Select bones aligned on the Y axis (along the bone's length).

Prefix Select bones with matching name prefix (separated by `.`).

Suffix Select bones with matching name suffix (separated by `.`).

Layer Select bones on the same layer.

Group Select bones in the same group.

Shape Select bones using the same shape object (in Pose Mode).

Select Pattern

Reference

Mode Object Mode

Menu *Select → Select Pattern…*

Selects all bones whose name matches a given pattern. Supported wild-cards: * matches everything, ? matches any single character, [abc] matches characters in "abc", and [!abc] match any character not in "abc". As an example *house* matches any name that contains "house", while floor* matches any name starting with "floor".

Case Sensitive The matching can be chosen to be case sensitive or not.

Extend When *Extend* checkbox is checked the selection is extended instead of generating a new one.

Editing

Introduction

As with any other object, you edit your armature in *Edit Mode* `Tab`.

The set of bone editing tools is quite similar to the one for *mesh* editing.

Important: One important thing to understand about armature editing is that you edit the *rest position* of your armature, i.e. its "default state". An armature in its *rest position* has all bones with *no* rotation and scaled to 1.0 in their own local space.

The different *poses* you might create afterwards are based on this rest position. So if you modify it in *Edit Mode*, all the poses already existing will also be modified. Thus you should in general be sure that your armature is definitive before starting to *skin* and *pose* it!

Note: Please note that some tools work on bones' joints, while others work on bones themselves. Be careful not to get confused.

Add Menu

Reference

Mode Edit Mode

Menu *Add*

Hotkey `Shift-A`

In the 3D Viewport, `Shift-A` to add a new bone to your armature.

This bone will be:

- Of one unit of length.
- Oriented towards the global Z axis.
- With its root placed at the 3D cursor position.
- With no relationship with any other bone of the armature.

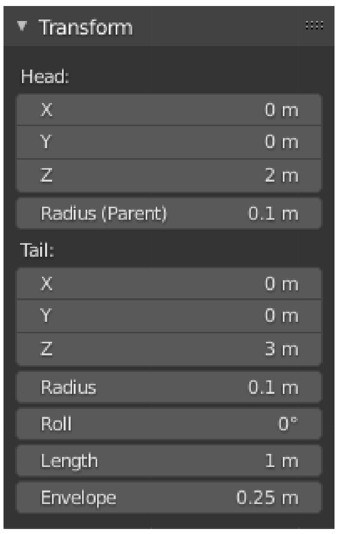
Locking Bones

You can prevent a bone from being transformed in *Edit Mode* in several ways:

- All bones can be locked clicking on the *Lock* checkbox of their Transform panel in the *Bones* tab;
- Press `Shift-W` *Toggle Bone Options* → *Locked*
- Select *Armature* → *Bone Settings* → *Toggle a Setting*.

If the root of a locked bone is connected to the tip of an unlocked bone, it will not be locked, i.e. you will be able to move it to your liking. This means that in a chain of connected bones, when you lock one bone, you only really lock its tip. With unconnected bones, the locking is effective on both joints of the bone.

Transform



Transform

Head:

X	0 m
Y	0 m
Z	2 m
Radius (Parent)	0.1 m

Tail:

X	0 m
Y	0 m
Z	3 m
Radius	0.1 m
Roll	0°
Length	1 m
Envelope	0.25 m

Fig. 1581: The Transform panel for armatures in Edit Mode.

We will not detail here the various transformations of bones, nor things like axis locking, pivot points, and so on, as they are common to most object editing, and already described in the *mesh section*. The same goes for mirroring, as it is nearly the same as with *mesh editing*. Just keep in mind that bones' roots and tips behave more or less like meshes' vertices, and bones themselves act like edges in a mesh.

As you know, bones can have two types of relationships: They can be parented, and in addition connected. Parented bones behave in *Edit Mode* exactly as if they had no relations. They can be moved, rotated, scaled, etc. without affecting their descendants. However, connected bones must always have parent's tips connected to child's roots, so by transforming a bone, you will affect all its connected parent/children/siblings.

While with other transform tools, the "local axes" means the object's axes, here they are the bone's own axes (when you lock to a local axis, by pressing the relevant key twice, the constraint is applied along the selected bone's local axis, not the armature object's axis).

Finally, you can edit in the *Transform* panel in the Sidebar region the positions and radius of both joints of the active selected bone, as well as its *roll rotation*.

Scale Radius

Reference

Mode Edit Mode

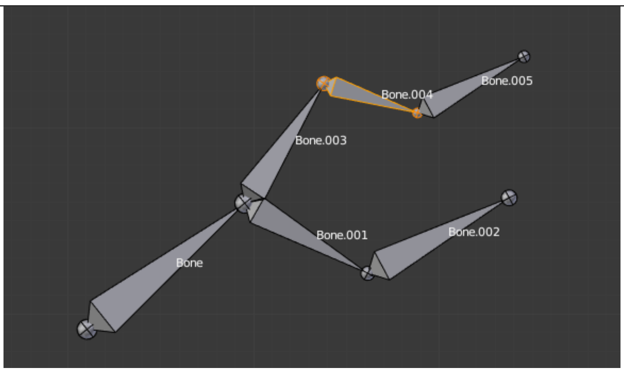
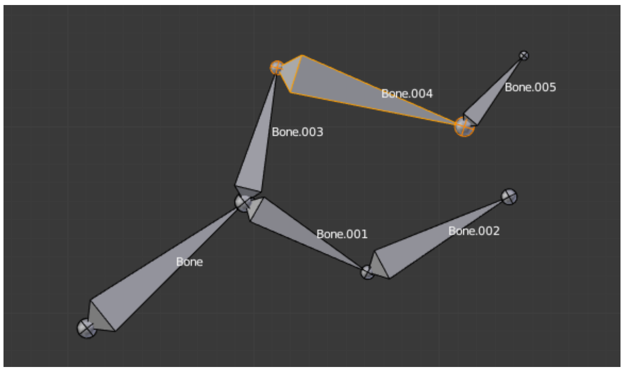
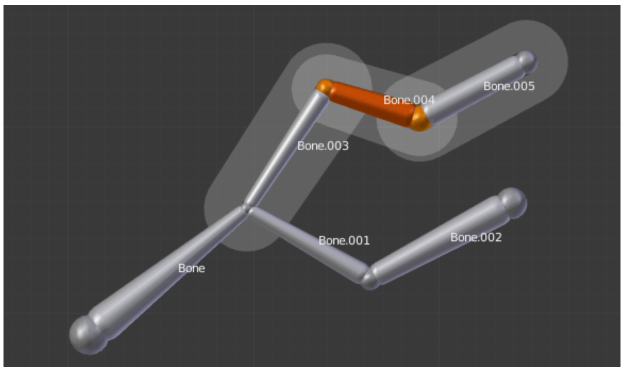
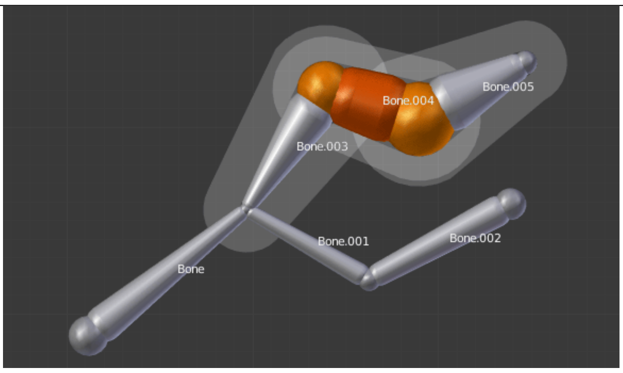
Menu *Armature → Transform → Scale Radius*

Hotkey `Alt-S`

You can alter the radius that a bone has by selecting the head, body or tail of a bone, and then press `Alt-S` and move the mouse left or right. If the body is selected the mean radius will be scaled. And as usual, with connected bones, you scale at the same time the radius of the parent's tip and of the children's roots.

You can also alter the bone radius by selecting the tail or head of the bone you wish to alter, then navigate to *Properties → Bone → Deform → Radius Section* and entering new values for the *Tail* and *Head* number fields.

Table 101: Bone Scale and Scale Radius comparison.

 Bone.004 Bone.005 Bone.003 Bone.001 Bone.002 Bone Fig. 1582: A single selected bone in Octahedron visualization.	 Bone.004 Bone.005 Bone.003 Bone.001 Bone.002 Bone Fig. 1583: After normal scale.
 Bone.004 Bone.005 Bone.003 Bone.001 Bone.002 Bone Fig. 1584: A single selected bone in Envelope visualization.	 Bone.004 Bone.005 Bone.003 Bone.001 Bone.002 Bone Fig. 1585: After Scaled Radius. Its length remains the same, but its joints' radius are bigger.

Note that, when you resize a bone (either by directly scaling it, or by moving one of its joints), Blender automatically adjusts the end-radii of its envelope proportionally to the size of the modification. Therefore, it is advisable to place all the bones first, and only then edit their properties.

Scale Envelope Distance

Reference

Mode Edit Mode and Pose Mode

Menu *Armature* → *Transform* → *Scale Envelope Distance*

Hotkey `Ctrl-Alt-S`

You can alter the size of the Bone Envelope volume by clicking on the body of the bone you want to alter, `Ctrl-Alt-S` then drag your mouse left or right and the Bone Envelope volume will alter accordingly.

You can also alter the Bone Envelope volume by selecting the Bone you wish to alter and then navigate to *Properties* → *Bone* → *Deform* → *Envelope* → *Distance* then enter a new value into it.

Altering the Bone Envelope volume does not alter the size of the bone just the range within which it can influence vertices of child objects.

Table 102: Envelope scaling example.

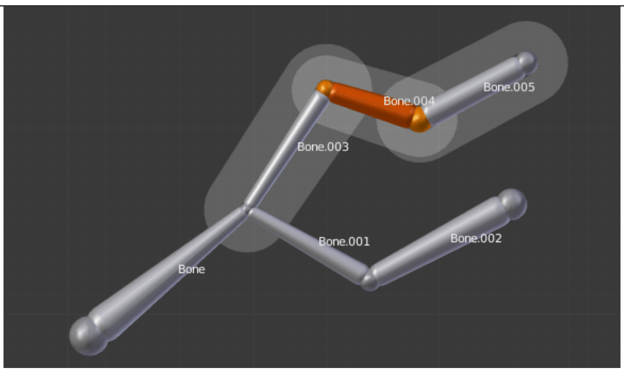
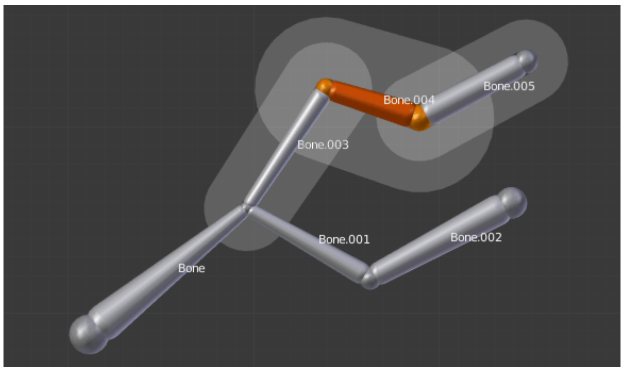
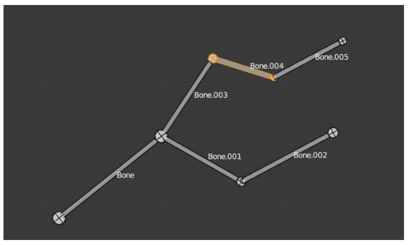
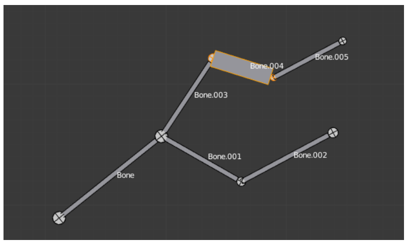
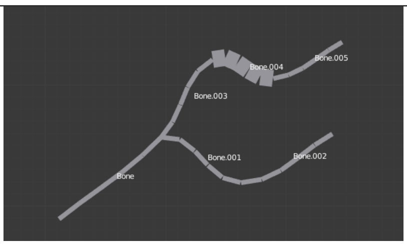
 Fig. 1586: A single bone selected in Envelope visualization.	 Fig. 1587: Its envelope distance scaled.

Table 103: “Bone size” scaling example.

 Fig. 1588: A single “default size” bone selected in B-Bone visualization.	 Fig. 1589: Its envelope distance scaled.	 Fig. 1590: The same armature in Object Mode and B-Bone visualization, with Bone.004’s size scaled up.

Align Bones

Reference

> **Mode** Edit Mode
>
> **Menu** *Armature → Transform → Align Bones*
>
> **Hotkey** `Ctrl-Alt-A`

Rotates the selected bones to achieve the same orientation as the active one.

Bone Roll

In *Edit Mode*, you can control the bone roll (i.e. the rotation around the Y axis of the bone).

However, after editing the armature, or when using *Euler Rotation*, you may want to set the bone roll.

Recalculate

Reference

Mode Edit Mode

Menu *Armature → Bone Roll → Recalculate*

Hotkey `Ctrl-N`

Axis Orientation

Local Tangent Align roll relative to the axis defined by the bone and its parent.

X, Z

Global Axis Align roll to global X, Y, Z axis.

X, Y, Z

Active Bone Follow the rotation of the active bone.

View Axis Set the roll to align with the viewport.

Cursor Set the roll towards the 3D cursor.

Flip Axis Reverse the axis direction.

Shortest Rotation Avoids rolling the bone over 90 degrees from its current value.

Set

Reference

Mode Edit Mode

Menu *Armature → Bone Roll → Set*

Hotkey `Ctrl-R`

This is a transform mode where you can edit the roll of all selected bones.

Extrude

Reference

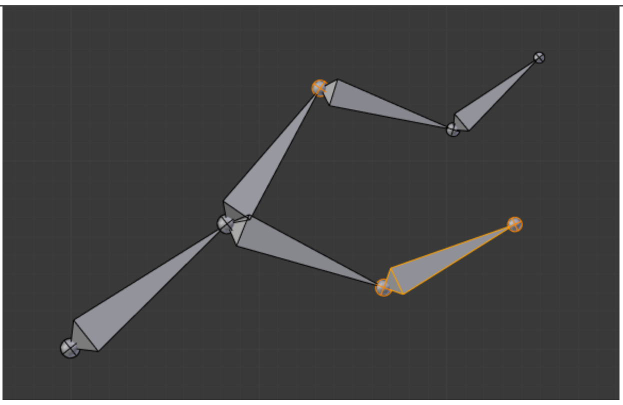
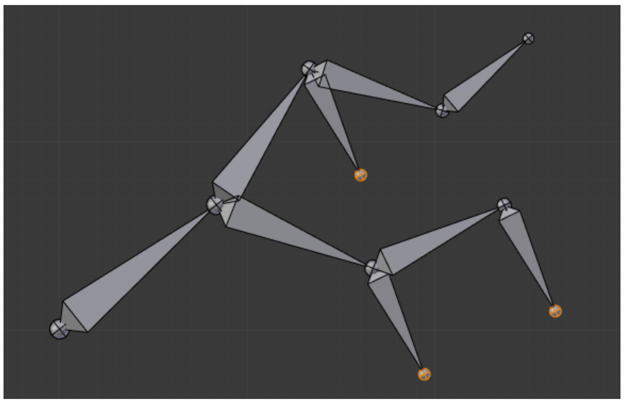
Mode Edit Mode

Menu *Armature → Extrude*

Hotkey `E`, `Shift-E`

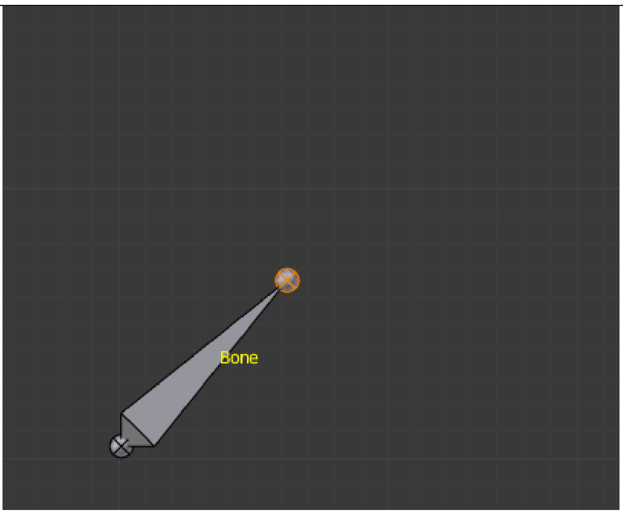
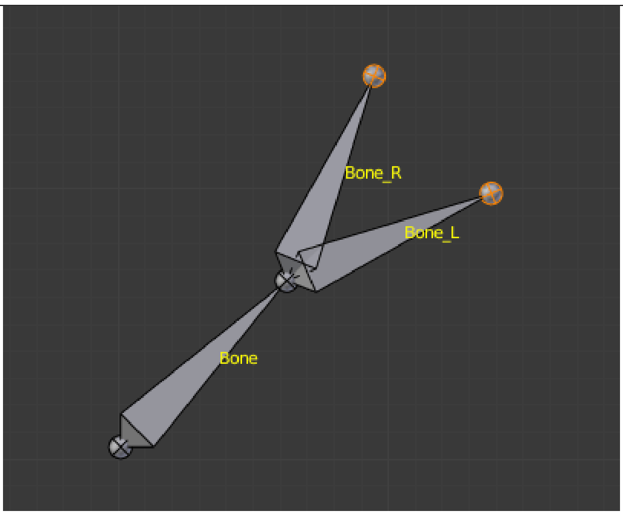
When you press `E`, for each selected tip (either explicitly or implicitly), a new bone is created. This bone will be the child of "its" tip owner, and connected to it. As usual, once extrusion is done, only the new bones' tips are selected, and in select mode, so you can place them to your liking. See Fig. *Extrusion example.*.

Table 104: Extrusion example.

 Fig. 1591: An armature with three selected tips.	 Fig. 1592: The three extruded bones.

You also can use the rotating/scaling extrusions, as with meshes, by pressing respectively `E` `R` and `E` `S` – as well as *locked* extrusion along a global or local axis.

Table 105: Mirror extrusion example.

 Fig. 1593: A single selected bone's tip.	 Fig. 1594: The two mirror-extruded bones.

Bones have an extra "mirror extruding" tool, called by pressing `Shift-E`. By default, it behaves exactly like the standard extrusion. But once you have enabled *X-Axis Mirror* editing option, each extruded tip will produce *two new bones*, having the same name except for the "_L"/ "_R" suffix (for left/right, see the *naming conventions*). The "_L" bone behaves like the single one produced by the default extrusion – you can move, rotate or scale it exactly the same way. The "_R" bone is its mirror counterpart (along the armature's local X axis), see Fig. *Mirror extrusion example*..

Important: Canceling the extrude action causes the newly created bones to snap back to the source position, (creating zero length bones). These will be removed when exiting Edit Mode, however, they can cause confusion and it's unlikely you want to keep them. If you realize the problem immediately undo the extrude action.

In case you are wondering, you cannot just press `X` to solve this as you would in mesh editing, because extrusion selects the newly created tips, and as explained below the Delete tool ignores bones' joints. To get rid of these extruded bones without undoing, you would have to move the tips, then select the bones and *delete* them.

Mouse Clicks

Reference

Mode Edit Mode

Hotkey `Ctrl-RMB`

If at least one bone is selected, `Ctrl-RMB`-clicking adds a new bone.

About the new bone's tip:

After you `Ctrl-RMB`-clicked it becomes the active element in the armature, it appears to be right where you clicked, but (as in mesh editing) it will be on the plane parallel to the view and passing through the 3D cursor.

The position of the root and the parenting of the new bone depends on the active element:

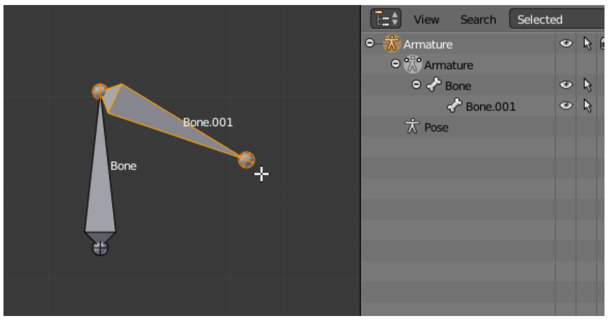


Fig. 1595: Ctrl-clicking when the active element is a bone.

If the active element is a *bone*:

- The new bone's root is placed on the active bone's tip.
- The new bone is parented and connected to the active bone (check the Outliner in Fig. *Ctrl-clicking when the active element is a tip.*).

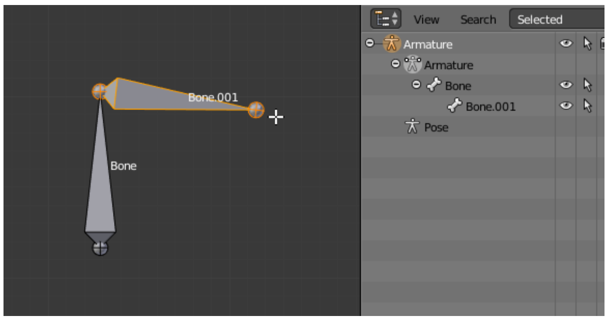


Fig. 1596: Ctrl-clicking when the active element is a tip.

If the active element is a *tip*:

- The new bone's root is placed on the active tip.
- The new bone is parented and connected to the bone owning the active tip (check the Outliner in Fig. *Ctrl-clicking when the active element is a tip.*).

If the active element is a *disconnected root*:

- The new bone's root is placed on the active root.
- The new bone is **not** parented to the bone owning the active root (check the Outliner in Fig. *Ctrl-clicking when the active element is a disconnected root.*).

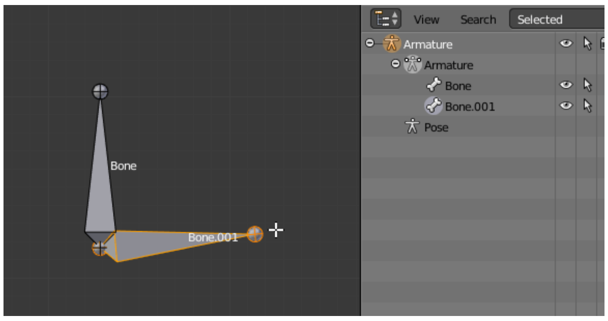


Fig. 1597: Ctrl-clicking when the active element is a disconnected root.

And hence the new bone will **not** be connected to any bone.

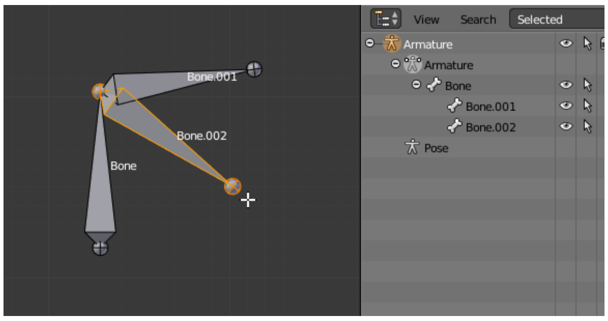


Fig. 1598: Ctrl-clicking when the active element is a connected root.

If the active element is a *connected root*:

- The new bone's root is placed on the active root.
- The new bone **is** parented and connected to the parent of the bone owning the active root (check the Outliner in Fig. *Ctrl-clicking when the active element is a connected root.*).

This should be obvious because if the active element is a connected root then the active element will be also the tip of the parent bone, so it is the same as the second case.

As the tip of the new bone becomes the active element, you can repeat these `Ctrl-RMB` clicks several times, to consecutively add several bones to the end of the same chain.

Duplicate

Reference

Mode Edit Mode

Menu *Armature → Duplicate*

Hotkey `Shift-D`

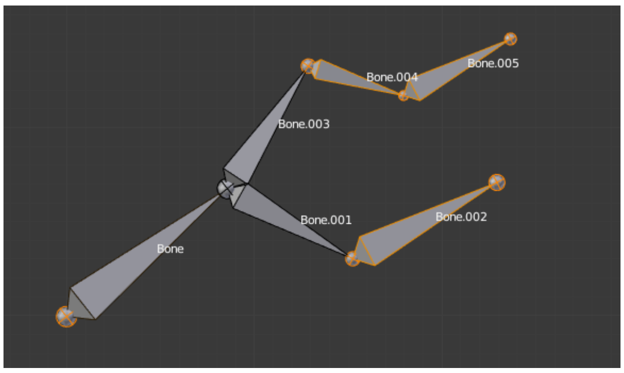
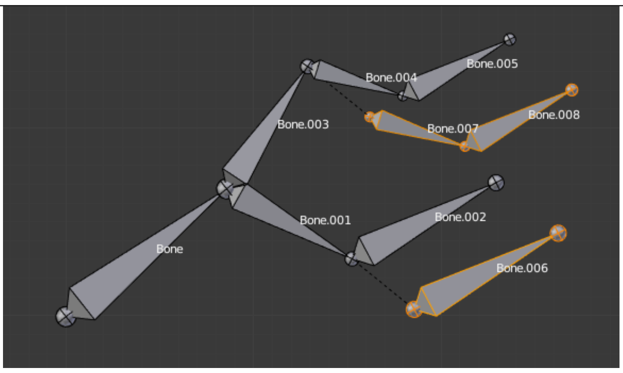
Note: This tool works on selected bones; selected joints are ignored.

As in mesh editing, by pressing `Shift-D` the selected bones will be duplicated. The duplicates become the selected elements and they are placed in select mode, so you can move them wherever you like.

If you select part of a chain, by duplicating it you will get a copy of the selected chain, so the copied bones are interconnected exactly like the original ones.

The duplicate of a bone which is parented to another bone will also be parented to the same bone, even if the root bone is not selected for the duplication. Be aware, though, that if a bone is parented **and** connected to an unselected bone, its copy will be parented, but **not** connected to the unselected bone (see Fig. *Duplication example.*).

Table 106: Duplication example.

 Bone.005 Bone.004 Bone.003 Bone.001 Bone.002 Bone Fig. 1599: An armature with three selected bones and a selected single root.	 Bone.005 Bone.004 Bone.003 Bone.007 Bone.008 Bone.001 Bone.002 Bone Bone.006 Fig. 1600: The three duplicated bones. Note that the selected chain is preserved in the copy, and that Bone.006 is parented but not connected to Bone.001, as indicated by the black dashed line. Similarly, Bone.007 is parented but not connected to Bone.003.

Fill Between Joints

Reference

Mode Edit Mode

Menu *Armature → Fill Between Joints*

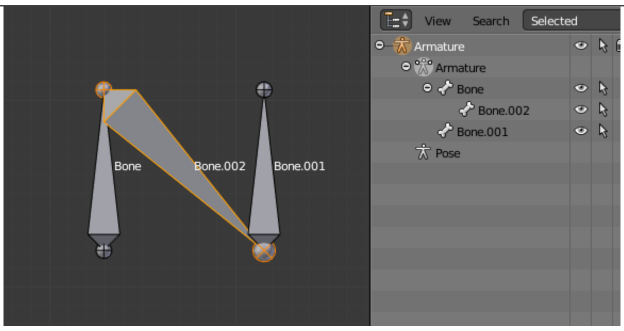
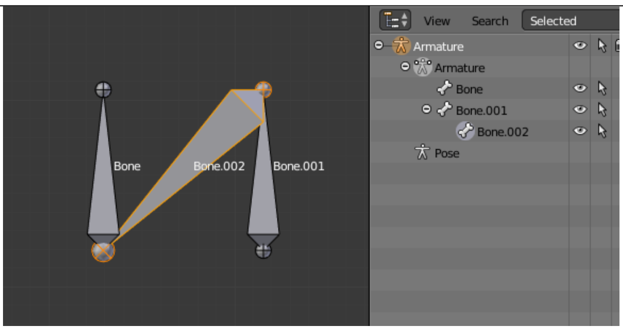
Hotkey F

The main use of this tool is to create one bone between two selected joints by pressing F, similar to how in mesh editing you can "create edges/faces".

If you have one root and one tip selected, the new bone:

- Will have the root placed on the selected tip.
- Will have the tip placed on the selected root.
- Will be parented and connected to the bone owning the selected tip.

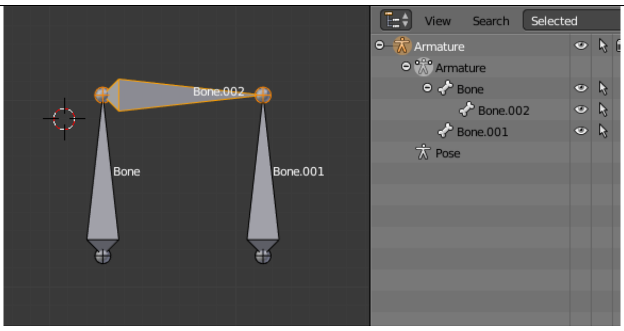
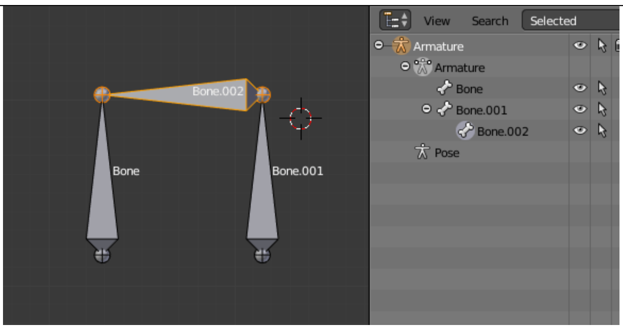
Table 107: Fill between a tip and a root.

 View Search Selected Armature Armature Bone Bone.002 Bone.001 Pose Bone Bone.002 Bone.001 Fig. 1601: Active tip on the left.	 View Search Selected Armature Armature Bone Bone.001 Bone.002 Pose Bone Bone.002 Bone.001 Fig. 1602: Active tip on the right.

If you have two tips selected, the new bone:

- Will have the root placed on the selected tip closest to the 3D cursor.
- Will have the tip placed on the other selected tip.
- Will be parented and connected to the bone owning the tip used as the new bone's root.

Table 108: Fill between tips.

 Bone.002 Bone Bone.001 View Search Selected Armature Armature Bone Bone.002 Bone.001 Pose Fig. 1603: 3D cursor on the left.	 Bone.002 Bone Bone.001 View Search Selected Armature Armature Bone Bone.001 Bone.002 Pose Fig. 1604: 3D cursor on the right.

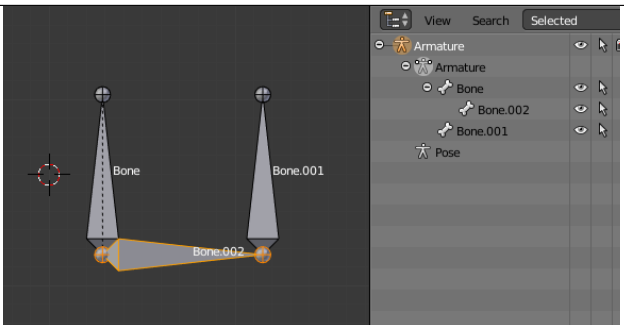
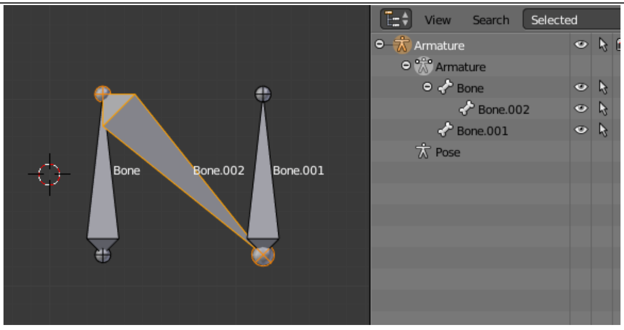
If you have two roots selected, you will face a small problem due to the event system in Blender not updating the interface in real-time.

When clicking F, similar to the previous case, you will see a new bone:

- With the root placed on the selected root closest to the 3D cursor.
- With the tip placed on the other selected root.
- Parented and connected to the bone owning the root used as the new bone's root.

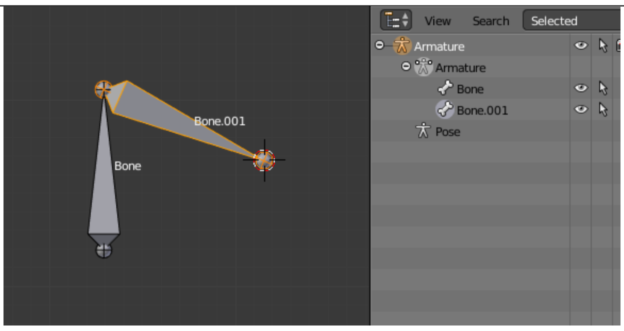
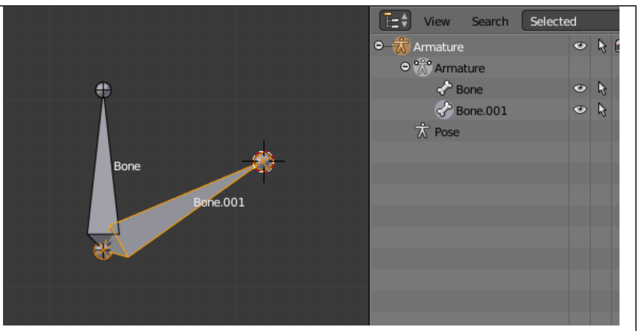
If you try to move the new bone, Blender will update the interface and you will see that the new bone's root moves to the tip of the parent bone.

Table 109: Fill between roots.

 Bone Bone.001 Bone.002 View Search Selected Armature Armature Bone Bone.002 Bone.001 Pose Fig. 1605: Before UI update (3D cursor on the left).	 Bone Bone.002 Bone.001 View Search Selected Armature Armature Bone Bone.002 Bone.001 Pose Fig. 1606: After UI update, correct visualization.

Clicking F with only one bone joint selected will create a bone from the selected joint to the 3D cursor position, and it will not parent it to any bone in the armature.

Table 110: Fill with only one bone joint selected.

 View Search Selected Armature Armature Bone Bone.001 Pose Bone.001 Bone Fig. 1607: Fill with only one tip selected.	 View Search Selected Armature Armature Bone Bone.001 Pose Bone Bone.001 Fig. 1608: Fill with only one root selected.

You will get an error when:

- Trying to fill two joints of the same bone.
- Trying to fill more than two bone joints.

Split

Reference

Mode Edit Mode

Menu *Armature → Split*

Hotkey `Y`

Disconnects the selection and clears the parent at the start and end. ToDo <2.8 add.

Separate Bones

Reference

Mode Edit Mode

Menu *Armature → Separate Bones*

Hotkey `P`

You can, as with meshes, separate the selected bones in a new armature object *Armature → Separate*, `Ctrl-Alt-P` and of course, in *Object Mode*, you can join all selected armatures in one *Object → Join Objects*, `Ctrl-J`.

Subdivide

Reference

Mode Edit Mode

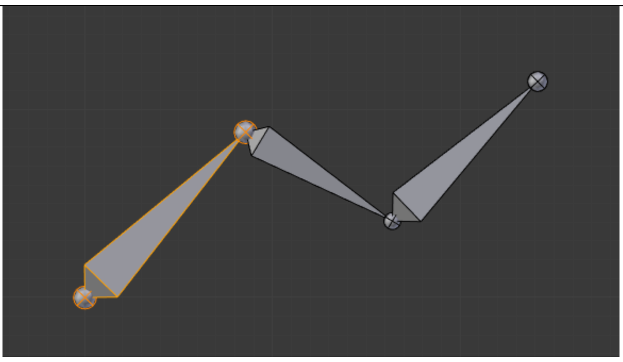
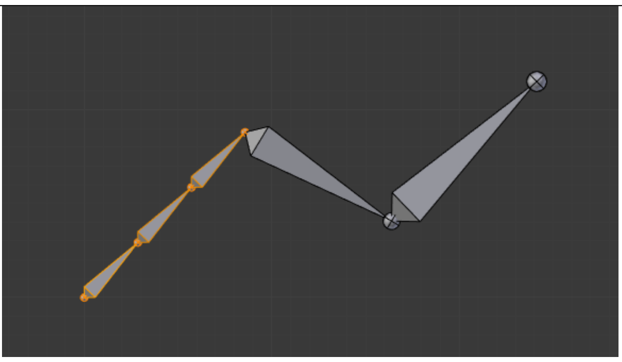
Menu *Armature → Subdivide*

You can subdivide bones, to get two or more bones where there was just one bone. The tool will subdivide all selected bones, preserving the existing relationships: the bones created from a subdivision always form a connected chain of bones.

To create an arbitrary number of bones from each selected bone in the Subdivide Multi *Adjust Last Operation* panel.

Number of Cuts Specifies the number of cuts. As in mesh editing, if you set *n* cuts, you will get *n* + 1 bones for each selected bone.

Table 111: Subdivision example.

 Fig. 1609: An armature with one selected bone, just before multi-subdivision.	 Fig. 1610: The selected bone has been "cut" two times, giving three sub-bones.

Switch Direction

Reference

Mode Edit Mode

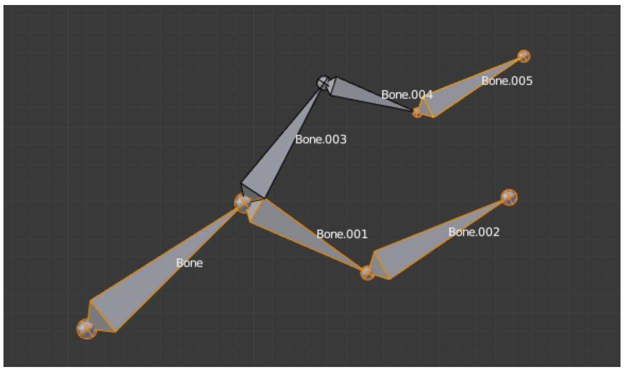
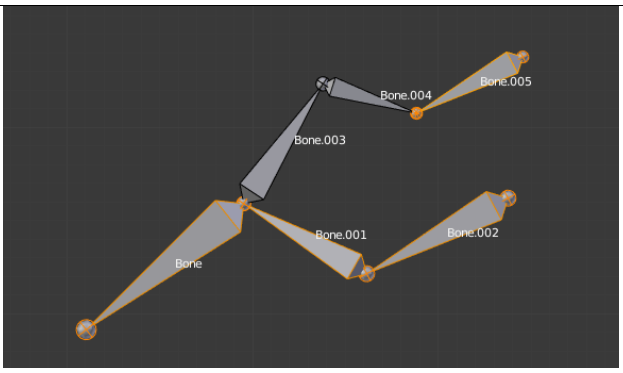
Menu *Armature* → *Switch Direction*

Hotkey `Alt-F`

This tool allows you to switch the direction of the selected bones (i.e. their root will become their tip, and vice versa).

Switching the direction of a bone will generally break the chain(s) it belongs to. However, if you switch a whole (part of a) chain, the switched bones will still be parented/connected, but in "reversed order". See the Fig. *Switching example.*.

Table 112: Switching example.

 Bone.004 Bone.005 Bone.003 Bone.001 Bone.002 Bone Fig. 1611: An armature with one selected bone, and one selected chain of three bones, just before switching.	 Bone.004 Bone.005 Bone.003 Bone.001 Bone.002 Bone Fig. 1612: The selected bones have been switched. Bone.005 is no more connected nor parented to anything. The chain of switched bones still exists, but reversed (now Bone.002 is its root, and Bone is its tip). Bone.003 is now a free bone.

Symmetrize

Reference

Mode Edit Mode

Menu *Armature → Symmetrize*

Todo.

Naming

Reference

Mode All Modes

Panel *Properties → Bone Properties*

You can rename your bones, either using the *Name* field in the *Bones Properties*. It is also possible to rename by double-clicking bones in the Outliner.

Blender also provides you some tools that take advantage of bones named in a left/right symmetry fashion, and others that automatically name the bones of an armature.

Naming Conventions

Naming conventions in Blender are not only useful for you in finding the right bone, but also to tell Blender when any two of them are counterparts.

In case your armature can be mirrored in half (i.e. it is bilaterally symmetrical), it is worthwhile to stick to a left/right naming convention. This will enable you to use some tools that will probably save you time and effort (like the *X-Axis Mirror* editing tool).

1. First you should give your bones meaningful base-names, like "leg", "arm", "finger", "back", "foot", etc.

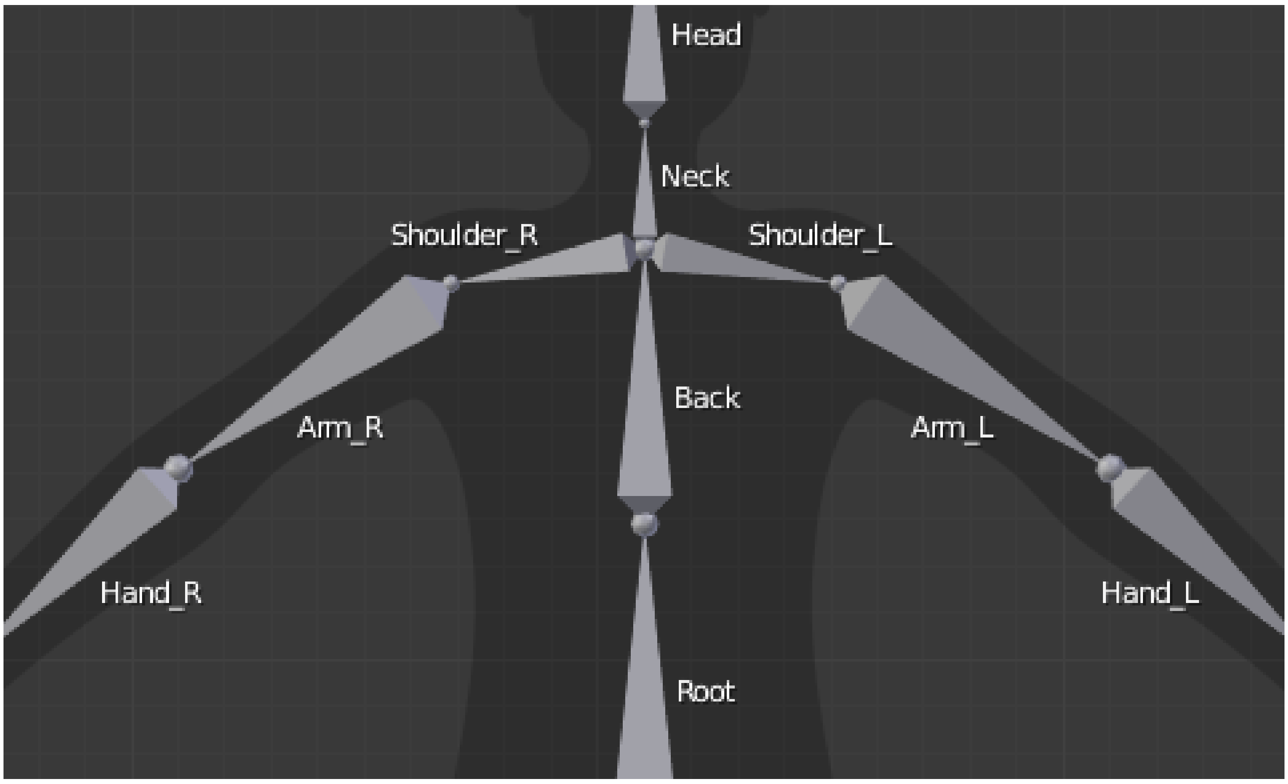


Fig. 1613: An example of left/right bone naming in a simple rig.

2. If you have a bone that has a copy on the other side (a pair), like an arm, give it one of the following separators:

   - Left/right separators can be either the second position "L_calfbone" or last-but-one "calfbone**.**R".
   - If there is a lower or upper case "L", "R", "left" or "right", Blender handles the counterpart correctly. See below for a list of valid separators. Pick one and stick to it as close as possible when rigging; it will pay off.

   Examples of valid separators:

   - (nothing): handLeft –> handRight
   - "_" (underscore): hand**_**L –> hand**_**R
   - "." (dot): hand**.**l –> hand**.**r
   - "-" (dash): hand**-**l –> hand**-**r
   - " " (space): hand LEFT –> hand RIGHT

   Note: Note that all examples above are also valid with the left/right part placed before the name. You can only use the short "L"/ "R" code if you use a separator (e.g. "handL"/ "handR" will not work!).

3. Before Blender handles an armature for mirroring or flipping, it first removes the number extension, e.g. ".001".
4. You can copy a bone named "bla.L" and flip it over using *Flip Names*. Blender will name the copy "bla.L.001" and flipping the name will give you "bla.R".

AutoName

Reference

Mode Edit Mode

Menu *Armature → Names → AutoName Left/Right, Front/Back, Top/Bottom*

The three *AutoName* entries of the *Armature → Names* menu allow you to automatically add a suffix to all selected bones, based on the position of their root relative to the armature's origin and its local coordinates:

AutoName Left/Right Will add the ".L" suffix to all bones with a *positive* X coordinate root, and the ".R" suffix to all bones with a *negative* X coordinate root. If the root is exactly at 0.0 on the X axis, the X coordinate of the tip is used. If both joints are at 0.0 on the X axis, the bone will just get a period suffix, with no "L"/ "R" (as Blender cannot decide whether it is a left or right bone…).

AutoName Front/Back Will add the ".Bk" suffix to all bones with a *positive* Y coordinate root, and the ".Fr" suffix to all bones with a *negative* Y coordinate root. The same as with *AutoName Left-Right* goes for 0.0 Y coordinate bones…

AutoName Top/Bottom Will add the ".Top" suffix to all bones with a *positive* Z coordinate root, and the ".Bot" suffix to all bones with a *negative* Z coordinate root. The same as with *AutoName Left-Right* goes for 0.0 Z coordinate bones…

Flip Names

Reference

Mode Edit Mode

Menu *Armature → Names → Flip Names*

You can flip left/right markers (see above) in selected bone names. This can be useful if you have constructed half of a symmetrical rig (marked for a left or right side) and duplicated and mirrored it, and want to update the names for the new side. Blender will swap text in bone names according to the above naming conventions, and remove number extensions if possible.

Change Layers

Change Armature Layers

Reference

Mode Edit Mode

Menu *Armature → Change Armature Layers*

Each armature has 32 *Layers* to organize armatures by "regrouping" them into sets of bones. Only bones in active layers will be visible/editable, but they will always be effective (i.e. move objects or deform geometry), whether in an active layer or not. This tool changes which layers are visible in the 3D Viewport. To show several layers at once, `Shift-LMB` on the desired layers to view. To move bones to a given layer, use *Change Bone Layers*.

Change Bone Layers

Reference

Mode Edit Mode

Menu *Armature → Change Bone Layers*

Hotkey `M`

Todo.

Parenting

Reference

Mode Edit Mode

Menu *Armature → Parent*

Panel *Properties → Bones → Relations*

Hotkey `Ctrl-P`, `Alt-P`

You can edit the relationships between bones (and hence create/modify the chains of bones) both from the 3D Views and the Properties. Whatever method you prefer, it's always a matter of deciding, for each bone, if it has to be parented to another one, and if so, if it should be connected to it.

To parent and/or connect bones, you can:

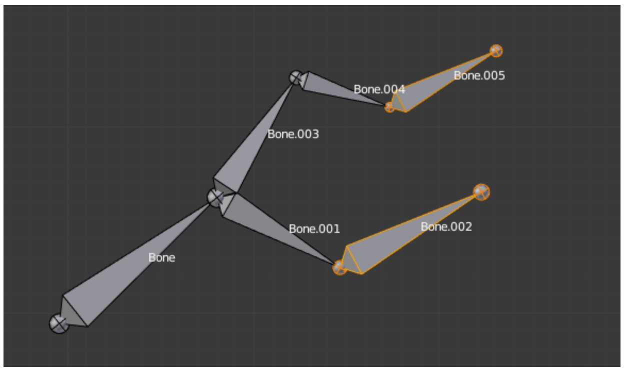
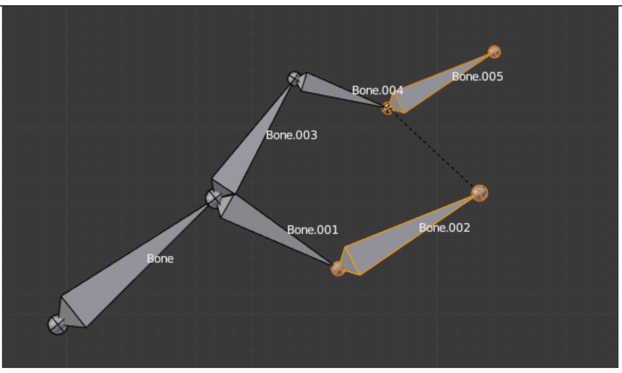
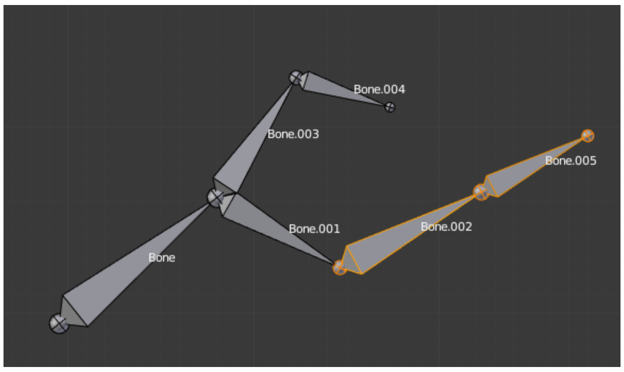
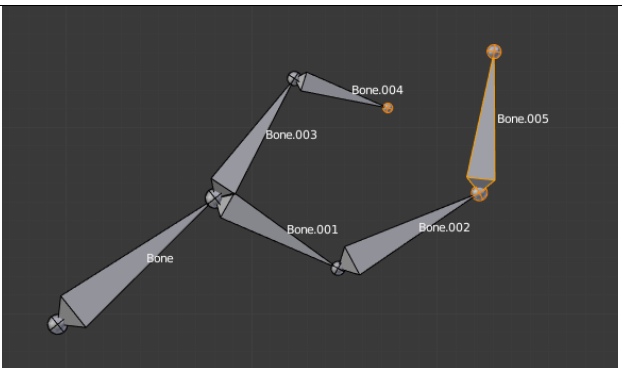
In a *3D View*, select the bone and *then* its future parent, and press `Ctrl-P` (or *Armature → Parent → Make Parent…*). In the small *Make Parent* menu that pops up, choose *Connected* if you want the child to be connected to its parent, else click on *Keep Offset*. If you have selected more than two bones, they will all be parented to the last selected one. If you only select one already-parented bone, or all selected bones are already parented to the last selected one, your only choice is to connect them, if not already done. If you select only one non-parented bone, you will get the *Need selected bone(s)* error message…

Note: With this method, the newly-children bones will not be scaled nor rotated – they will just be moved if you choose to connect them to their parent's tip.

In the *Properties*, Bones tab, for each selected bone, you can select its parent in the *Parent* data ID to the upper right corner of its Relations panel. If you want them to be connected, just enable the checkbox to the right of the list.

Note: With this method, the tip of the child bone will never be moved – so if *Connected* is enabled, the child bone will be completely transformed by the operation.

Table 113: Parenting example.

 Bone.005 Bone.004 Bone.003 Bone.002 Bone.001 Bone Fig. 1614: The starting armature, with Bone.005 parented and connected to Bone.004.	 Bone.005 Bone.004 Bone.003 Bone.002 Bone.001 Bone Fig. 1615: Bone.005 re-parented to Bone.002, but not connected to it (same result, using either `Ctrl-P 2` in 3D Viewport, or the Bones tab settings).
 Bone.004 Bone.003 Bone.005 Bone.002 Bone.001 Bone Fig. 1616: Bone.005 parented and connected to Bone.002, using `Ctrl-P 1` in 3D Viewport.	 Bone.004 Bone.005 Bone.003 Bone.002 Bone.001 Bone Fig. 1617: Bone.005 parented and connected to Bone.002. Using the Parent data ID of Bone.005 Relations panel.

To disconnect and/or free bones, you can:

- In a 3D Viewport, select the desired bones, and press `Alt-P` (or *Armature* → *Parent* → *Clear Parent…*). In the small *Clear Parent* menu that pops up, choose *Clear Parent* to completely free all selected bones, or *Disconnect Bone* if you just want to break their connections.
- In the Properties, *Bones* tab, for each selected bone, you can select no parent in the *Parent* data ID of its Relations panel, to free it completely. If you just want to disconnect it from its parent, disable the *Connected* checkbox.

Note that relationships with non-selected children are never modified.

Properties

Reference

Mode Edit Mode

Menu *Armature* → *Bone Settings* → *…*

Hotkey `Shift-W`, `Shift-Ctrl-W`, `Alt-W`

Most bones' properties (except the transform ones) are regrouped in each bone's panels, in the *Bones* tab in *Edit Mode*. Let us detail them.

Note that some of them are also available in the 3D Views, through the three pop-up menus within the same entry:

- *Toggle Setting*: `Shift-W` or *Armature* → *Bone Settings* → *Toggle a Setting*
- *Enable Setting*: `Shift-Ctrl-W` or *Armature* → *Bone Settings* → *Enable a Setting*
- *Disable Setting*: `Alt-W` or *Armature* → *Bone Settings* → *Disable a Setting*

Display Wire Always display the bone as wireframe.

Deform (also `Shift-W` → *(Deform, …)*).

Multiply Vertex Group by Envelope (also `Shift-W` → *(Multiply Vertex Group by Envelope, …)*).

These settings control how the bone influences its geometry, along with the bones' joints radius. This will be detailed in the *skinning part*.

Inherit Rotation The bone automatically rotates together with its parent in *Pose Mode*. For more details, see the *relations page*.

Lock (also `Shift-W` → *(Locked, …)*) This will prevent all editing of the bone in *Edit Mode*; see *bone locking*.

Delete

Bones

Reference

Mode Edit Mode

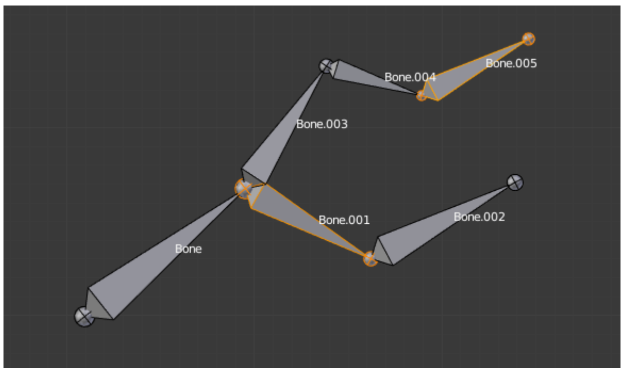
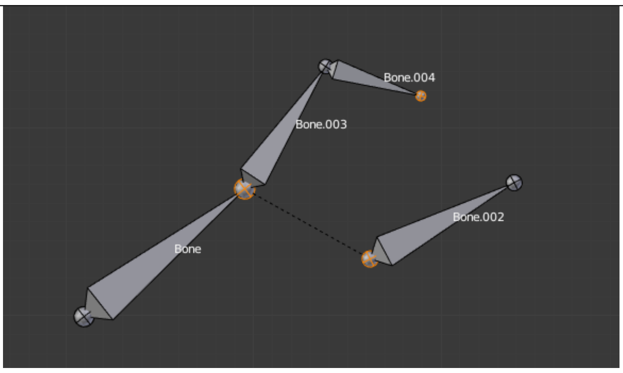
Menu *Armature* → *Delete* → *Bones*

Hotkey `X`

This tool delete selected bones, selected *joints* are ignored.

If you delete a bone in a chain, its child(ren) will be automatically re-parented to its own parent, but **not** connected, to avoid deforming the whole armature.

Table 114: Deletion example.

 Fig. 1618: An armature with two selected bones, just before deletion.	 Fig. 1619: The two bones have been deleted. Note that Bone.002, previously connected to the deleted Bone.001, is now parented but not connected to Bone.

Dissolve

Reference

Mode Edit Mode

Menu *Armature → Delete → Dissolve*

Hotkey `Ctrl-X`

Todo 2.76.

Properties

Introduction

Reference

Mode Object Mode, Edit Mode and Pose Mode

Panel *Properties → Bone*

When bones are selected (hence in *Edit Mode* and *Pose Mode*), their properties are shown in the *Bone* tab of the Properties. This shows different panels used to control features of each selected bone; the panels change depending on which mode you are working in.

Transform

Reference

Mode Edit Mode and Pose Mode

Panel *Bone → Transform*

When in *Edit Mode* you can use this panel to control position and roll of individual bones. Whereas in *Pose Mode* you can only set location for the main bone, and you can now set rotation and scale.

In addition, in *Pose Mode* it is possible to restrict changes in position, rotation and scale by axis on each bone in the armature.

Bendy Bones

Reference

Mode All Modes

Panel *Bone → Bendy Bones*

Bendy Bones (B-Bones) are an easy way to replace long chains of many small rigid bones. A common use case for curved bones is to model spine columns or facial bones.

Technical Details

Blender treats the bone as a section of a Bézier curve passing through the bones' joints. Each of the *Segments* will bend and roll to follow this invisible curve representing a tessellated point of the Bézier curve. The control points at each end of the curve are the endpoints of the bone. The shape of the B-Bones can be controlled using a series of properties or indirectly through the neighboring bones (i.e. first child and parent). The properties construct handles on either end of the bone to control the curvature.

When using the B-bone as a constraint target *Data ID* offers an option to follow the curvature.

Note: However, if the bone is used as a target rather than to deform geometry, only *Armature* and *Copy Transforms* constraints will use the full transformation including roll and scale.

Display

You can see these segments only if bones are visualized as *B-bones*.

When not visualized as *B-Bone*s, bones are always shown as rigid sticks, even though the bone segments are still present and effective. This means that even in e.g. *Octahedron* visualization, if some bones in a chain have several segments, they will nonetheless smoothly deform their geometry.

Rest Pose

The initial shape of a B-Bone can be defined in Edit Mode as a rest pose of that bone. This is useful for curved facial features like curved eyebrows or mouths.

B-Bones have two sets of the Bendy Bone properties – one for Edit Mode (i.e. the Rest Pose/Base Rig) and another for Pose Mode – adding together their values to get the final transforms.

Example

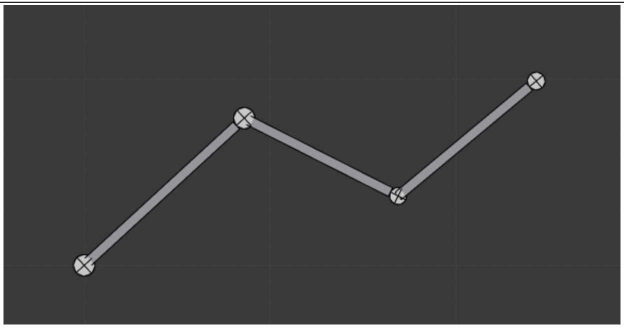


Fig. 1620: Bones with just one segment in Edit Mode.

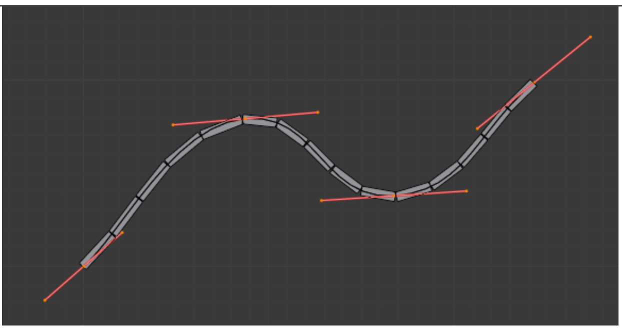


Fig. 1621: The Bézier curve superposed to the chain, with its handles placed at bones' joints.

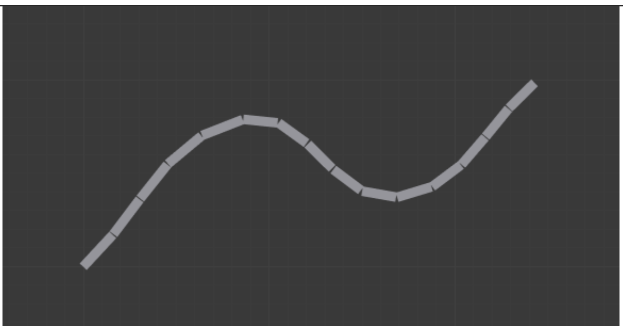


Fig. 1622: The same armature in Object Mode.

In Fig. *Bones with just one segment in Edit Mode.* we connected three bones, each one made of five segments.

Look at Fig. *The same armature in Object Mode.*, we can see how the bones' segments smoothly "blend" into each other, even for roll.

Options

Segments

The *Segments* number field allows you to set the number of segments, which the given bone is subdivided into. Segments are small, rigid linked child bones that interpolate between the root and the tip. The higher this setting, the smoother "bends" the bone, but the heavier the pose calculations.

Curve XY Offsets

Applies offsets to the curve handle positions on the plane perpendicular to the bone's primary (Y) axis. As a result, the handle moves per axis (XY) further from its original location, causing the curve to bend.

Roll

Roll In, Out The roll value (or twisting around the main Y axis of the bone) is interpolated per segment, between the start and end roll values. It is applied as a rotational offset on top of the previous rotation.

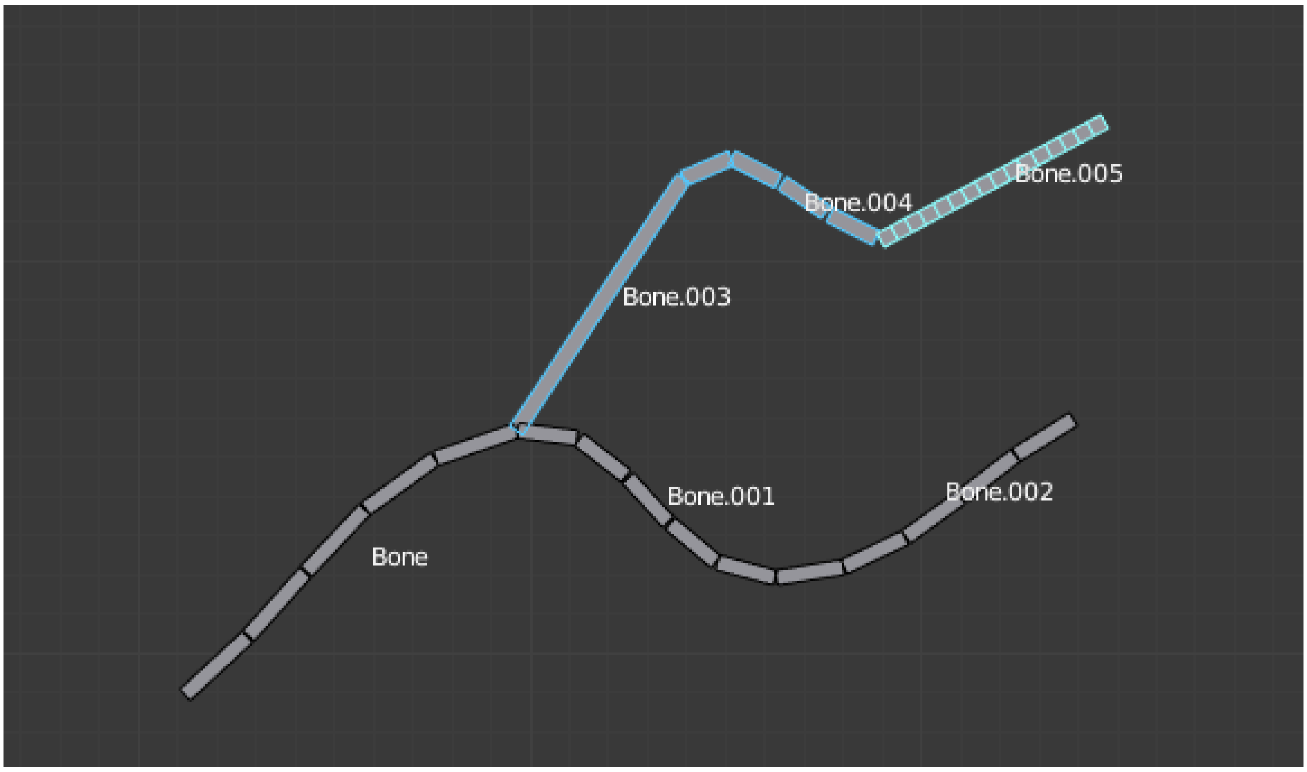


Fig. 1623: An armature in Pose Mode, B-Bone visualization: Bone.003 has one segment, Bone.004 has four, and Bone.005 has sixteen.

Inherit End Roll If enabled, the *Roll Out* value of the *Start Handle* bone (connected parent by default) will be implicitly added to the *Roll In* setting of the current bone.

Scale

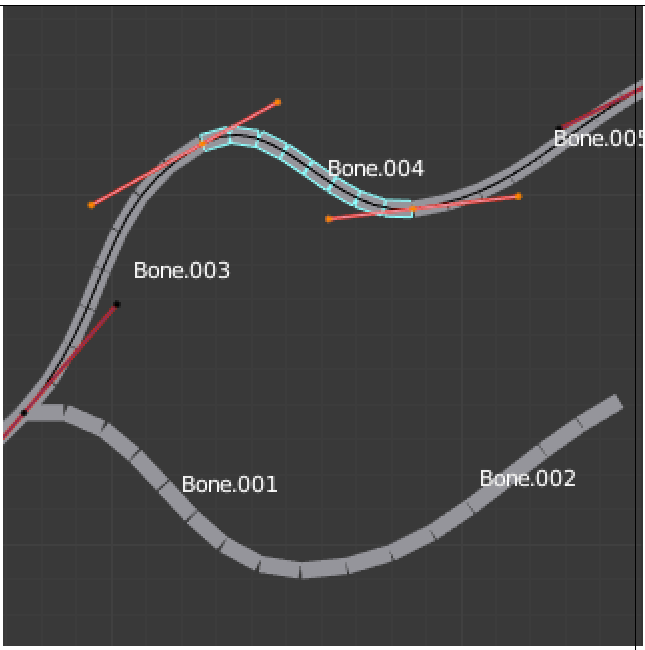
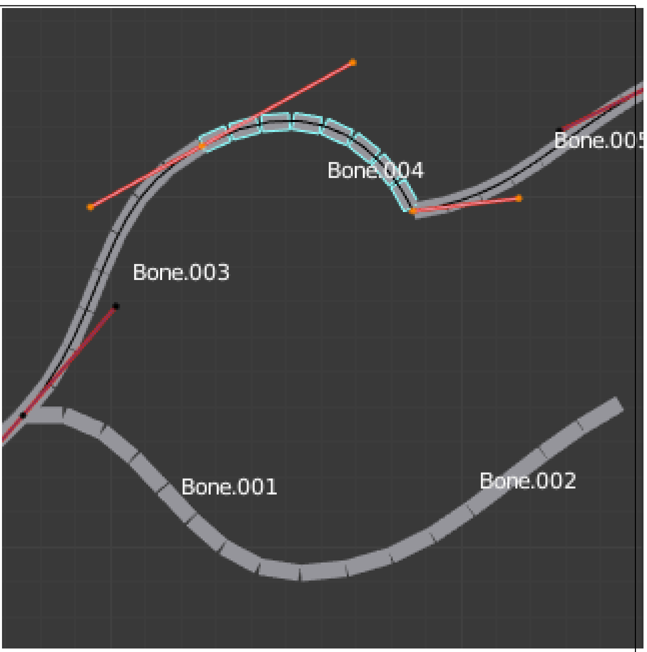
Scale In X/Y, Scale Out X/Y Scaling factor that adjusts the thickness of each segment for the X and Y axes only, i.e. length (Z axis) is not affected. Similar to *Roll* it is interpolated per segment.

Easing

Ease In, Out The *Ease In/Out* number fields, change the "length" of the *"auto"* Bézier handle to control the "root handle" and "tip handle" of the bone, respectively.

These values are proportional to the default length, which of course automatically varies depending on bone length, angle with the reference handle, and so on.

Table 115: Ease In/Out settings example, with a materialized Bézier curve.

 Fig. 1624: Bone.004 with default In and Out (1.0).	 Fig. 1625: Bone.004 with In at 2.0, and Out at 0.0.

Custom Handles

B-Bones can use custom bones as their reference bone handles, instead of only using the connected parent/child bones.

Start, End Handle Type Specifies the type of the handle from the following choices:

> **Automatic** The connected parent (or first connected child) of the bone is chosen as the handle. Calculations are done according to the *Absolute* handle type below.
>
> **Absolute** The Bézier handle is controlled by the **position** of the head (tail) of the handle bone relative to the head (tail) of the current bone. Note that for this to work, there must be a non-zero distance between these bones. If the handle is also a B-Bone, additional processing is applied to further smooth the transition, assuming that the bones in effect form a chain.
>
> **Relative** The Bézier handle is controlled by the **offset** of the head (tail) of the handle bone from its rest pose. The use of this type is not recommended due to numerical stability issues near zero offset.
>
> **Tangent** The Bézier handle is controlled by the **orientation** of the handle bone, independent of its location.

Custom Handle For types other than *Automatic*, a bone to use as handle has to be manually selected. Switching to a custom handle type without selecting a bone can be used to effectively disable the handle.

It is valid for two bones to refer to each other as handles – this correlation is applied in connected chains with *Automatic* handles.

Tip: Keying Set

The "BBone Shape" Keying Set includes all Bendy Bones properties.

Example

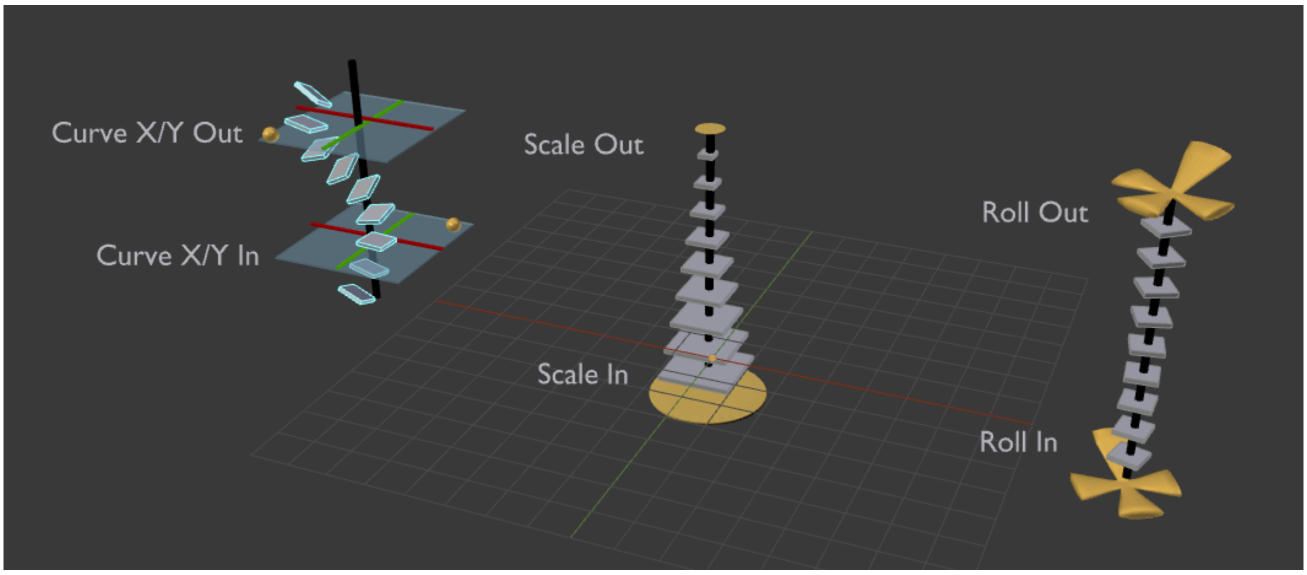


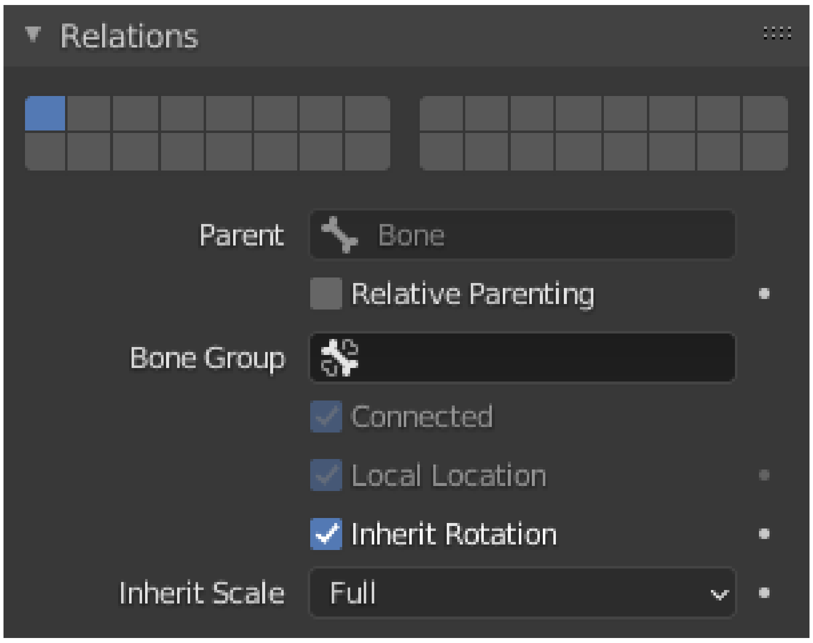
Fig. 1626: Visualization of the Bendy Bones properties.
From Left: 1) Curve X/Y offsets, 2) Scale In/Out, 3) Roll In/Out

Relations

Reference

Mode All Modes

Panel *Bone* → *Relations*


Relations

Parent: Bone

☐ Relative Parenting

Bone Group:

✓ Connected

✓ Local Location

✓ Inherit Rotation

Inherit Scale: Full

Fig. 1627: Relations panel.

In this panel you can arrange sets of bones in different layers for easier manipulation.

Bone Layers

Moving Bones between Layers

Obviously, you have to be in *Edit Mode* or *Pose Mode* to move bones between layers. Note that as with objects, bones can lay in several layers at once, just use the usual `Shift-LMB` clicks… First of all, you have to select the chosen bone(s)!

- In the Properties, use the "layer buttons" of each selected bone Relations panel (*Bones* tab) to control in which layer(s) it lays.
- In the *3D View* editor, use the menu *Armature* → *Move Bone To Layer* or *Pose* → *Move Bone To Layer* or press `M` to show the usual pop-up layers menu. Note that this way, you assign the same layers to all selected bones.

Bone Group

Reference

Mode Pose Mode

To assign a selected bone to a given bone group use the *Bone Group* data ID.

Object Children

Reference

Mode Pose Mode

Relative Parenting Changes how transformation of the bone is applied to its child Objects.

Parenting

Parent A *Data ID* to select the bone to set as a parent.

Connected The *Connected* checkbox set the head of the bone to be connected with its parent root.

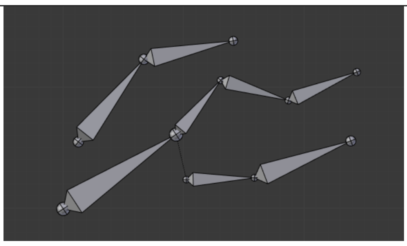
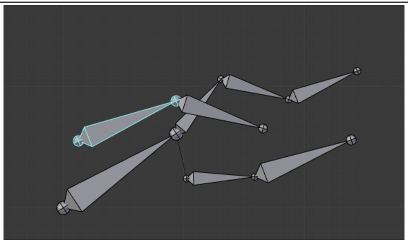
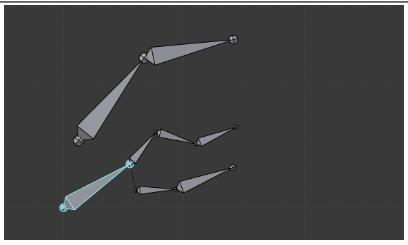
Transformations

Bones relationships have effects on transformations behavior.

By default, children bones inherit:

- Their parent position, with their own offset of course.
- Their parent rotation (i.e. they keep a constant rotation relatively to their parent).
- Their parent scale, here again with their own offset.

Table 116: Examples of transforming parented/connected bones.

 Fig. 1628: The armature in its rest position.	 Fig. 1629: Rotation of a root bone.	 Fig. 1630: Scaling of a root bone.

Exactly like standard children objects. You can modify this behavior on a per-bone basis, using the Relations panel in the *Bones* tab:

Local Location When disabled, the location transform property is evaluated in the parent bone's local space, rather than using the bone's own *rest pose* local space orientation.

Inherit Rotation When disabled, this will "break" the rotation relationship to the bone's parent. This means that the child will keep its rotation in the armature object space when its parent is rotated.

Inherit Scale Specifies which effects of parent scaling the bone inherits:

Full The bone inherits all effects of parent scaling and shear.

Fix Shear Full parent effects are applied to the rest state of the child, after which any shear is removed in a way that preserves the bone direction, length and volume, and minimally affects roll on average. The result is combined with the local transformation of the child.

If the inherited scale is non-uniform, this does not prevent shear from reappearing due to local rotation of the child bone, or of its children.

Aligned Parent scaling is inherited as if the child was oriented the same as the parent, always applying parent X scale over child X scale, and so on.

Average Inherits a uniform scaling factor that is the total change in the volume of the parent.

None Ignores all scaling and shear of the parent.

None (Legacy) Ignores all scaling, provided the parent is not sheared. If it is, there are no guarantees.

This choice replicates the behavior of the old Inherit Scale checkbox, and may be removed in a future release.

These inheriting behaviors propagate along the bones' hierarchy. So when you scale down a bone, all its descendants are by default scaled down accordingly. However, if you disable one bone's *Inherit Scale* or *Inherit Rotation* property in this "family", this will break the scaling propagation, i.e. this bone *and all its descendants* will no longer be affected when you scale one of its ancestors.

Tip: The various *Inherit Scale* options are provided as tools in avoiding shear that is caused by non-uniform scaling combined with parenting and rotation. There is no obvious best way to achieve that, so different options are useful for different situations.

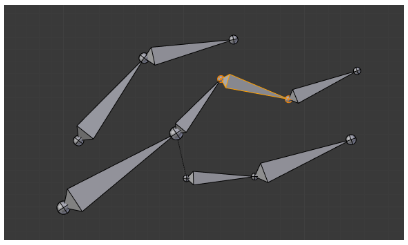
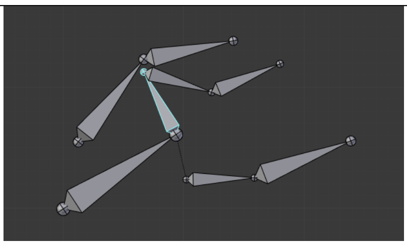
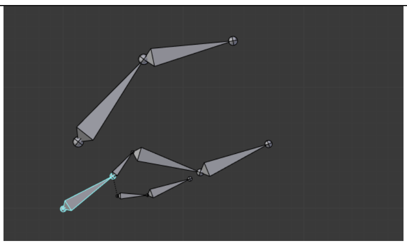
None Useful for gaining full control over the scaling of the child in order to e.g. manually overwrite it with constraints.

Average Useful to block squash and stretch propagation between sub-rigs, while allowing uniform changes in the size and volume to pass through.

Aligned Can be used within bone chains, e.g. tentacles, in order to propagate lengthwise scaling as lengthwise, and sideways as sideways, no matter how the tentacle bends. Similar to using *None* with *Copy Scale* from parent.

Fix Shear May be useful at the base of an appendage in order to reallocate squash and stretch between axes based on the difference in rest pose orientations of the parent and child. It behaves closest to *Full* while suppressing shear.

Table 117: Examples of transforming parented/connected bones with Inherit Rotation disabled.

 Fig. 1631: The yellow outlined Inherit Rotation disabled bone in the armature.	 Fig. 1632: Rotation of a bone with an Inherit Rotation disabled bone among its descendants.	 Fig. 1633: Scaling of a bone with an Inherit Rotation disabled bone among its descendants.

Connected bones have another specificity: they cannot be moved. Indeed, as their root must be at their parent's tip, if you do not move the parent, you cannot move the child's root, but only its tip, which leads to a child rotation. This is exactly what happens, when you press `G` with a connected bone selected, Blender automatically switches to rotation operation.

Bones relationships also have important consequences on how selections of multiple bones behave when transformed. There are many different situations which may not be included on this list, however, this should give a good idea of the problem:

- Non-related selected bones are transformed independently, as usual.
- When several bones of the same "family" are selected, *only* the "most parent" ones are really transformed – the descendants are just handled through the parent relationship process, as if they were not selected (see Fig. *Scaling bones, some of them related.* the third tip bone, outlined in yellow, was only scaled down through the parent relationship, exactly as the unselected ones, even though it is selected and active. Otherwise, it should have been twice smaller!)

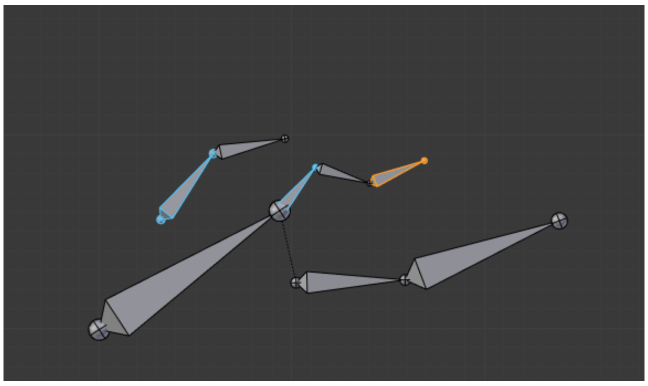


Fig. 1634: Scaling bones, some of them related.

- When connected and unconnected bones are selected, and you start a move operation, only the unconnected bones are affected.
- When a child connected hinge bone is in the selection, and the "most parent" selected one is connected, when you press `G`, nothing happens, because Blender remains in move operation, which of course has no effect on a connected bone.

So, when posing a chain of bones, you should always edit its elements from the root bone to the tip bone. This process is known as *Forward Kinematics* (FK). We will see in a *later page* that Blender features another pose method, called *Inverse Kinematics* (IK), which allows you to pose a whole chain just by moving its tip.

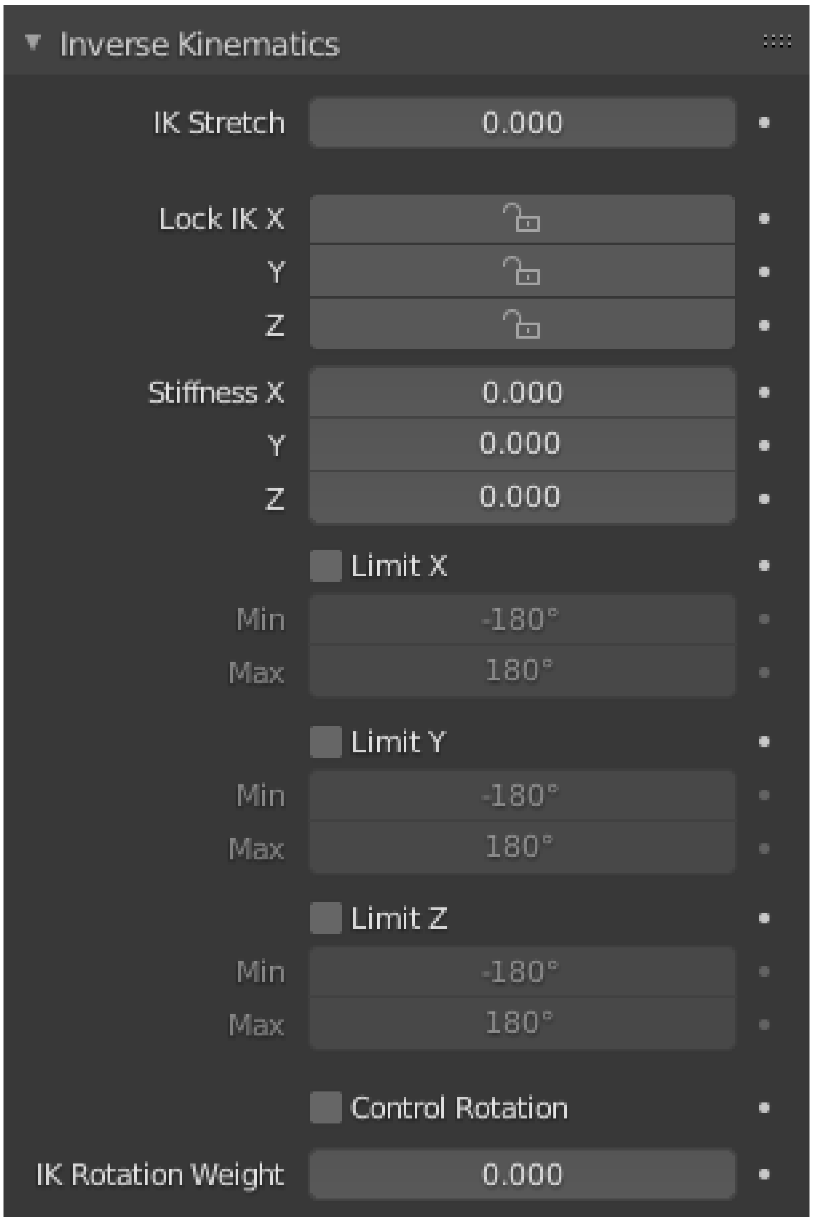
Note: This feature is somewhat extended/completed by the *pose library* tool.

Inverse Kinematics

Reference

Mode Pose Mode

Panel *Bone → Inverse Kinematics*



Inverse Kinematics

IK Stretch 0.000

Lock IK X

Y

Z

Stiffness X 0.000

Y 0.000

Z 0.000

Limit X

Min -180°

Max 180°

Limit Y

Min -180°

Max 180°

Limit Z

Min -180°

Max 180°

Control Rotation

IK Rotation Weight 0.000

Fig. 1635: The Inverse Kinematics panel.

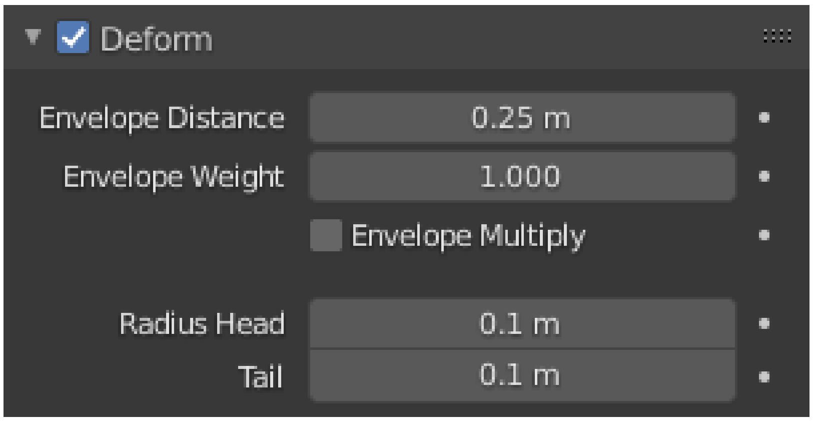
This panel controls the way a bone or set of bones behave when linked in an *inverse kinematic* chain.

Deform

Reference

Mode All Modes

Panel *Bone → Deform*


Deform

Envelope Distance: 0.25 m

Envelope Weight: 1.000

Envelope Multiply: ☐

Radius Head: 0.1 m

Tail: 0.1 m

Fig. 1636: The Deform panel.

In this panel you can set deformation options for each bone.

Toggling the checkbox in the panel header off, prevents the bone from deforming the geometry at all, overriding any weights that it might have been assigned before; it mutes its influence.

It also excludes the active bone in the automatic weight calculation when the mesh is parented to the armature using the *Armature Deform* tool with the *With Automatic Weights* option.

Envelope

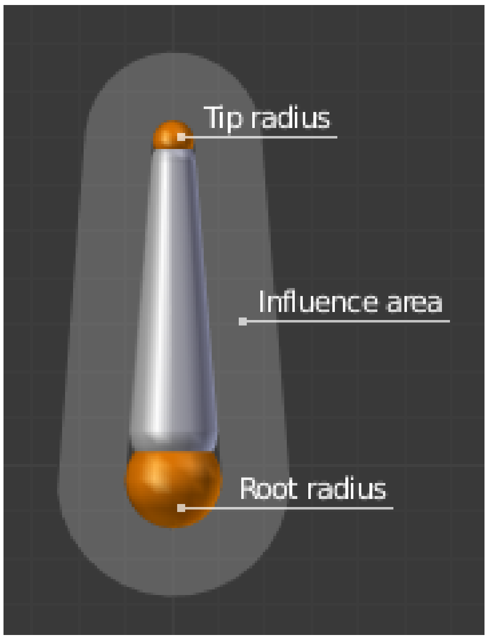


Fig. 1637: Bone influence areas for envelopes method.

Envelopes is the most general skinning method. It works with all available object types for skinning (meshes, lattices, curves, surfaces and texts). It is based on proximity between bones and their geometry, each bone having two different areas of influence, shown in the *Envelope* visualization:

- The inside area, materialized by the "solid" part of the bone, and controlled by both root and tip radius.
- The outside area, materialized by the lighter part around the bone, and controlled by the *Distance* setting.

See also:

The *editing pages* for how to edit these properties.

Envelope Distance The Distance defines a volume which is the range within the bone has an influence on vertices of the deformed object. The geometry is less and less affected by the bone as it goes away by following a quadratic decay.

Envelope Weight A bone property, that controls the global influence of the bone over the deformed object, when using the envelopes method.

It is only useful for the parts of geometry that are "shared", influenced by more than one bone (generally, at the joints…) – a bone with a high weight will have more influence on the result than one with a low weight… Note that when set to 0.0, it has the same effect as disabling the *Deform* option.

Radius Set the radius for the head and the tail of envelope bones. Inside this volume, the geometry if fully affected by the bone.

Envelope Multiply This option controls how the two deforming methods interact, when they are both enabled. By default, when they are both active, all vertices belonging to at least one vertex group are only deformed through the vertex groups method. The other "orphan" vertices being handled by the envelopes one. When you enable this option, the "deformation influence" that this bone would have on a vertex (based from its envelope settings) is multiplied with this vertex's weight in the corresponding vertex group. In other words, the vertex groups method is further "weighted" by the envelopes method.

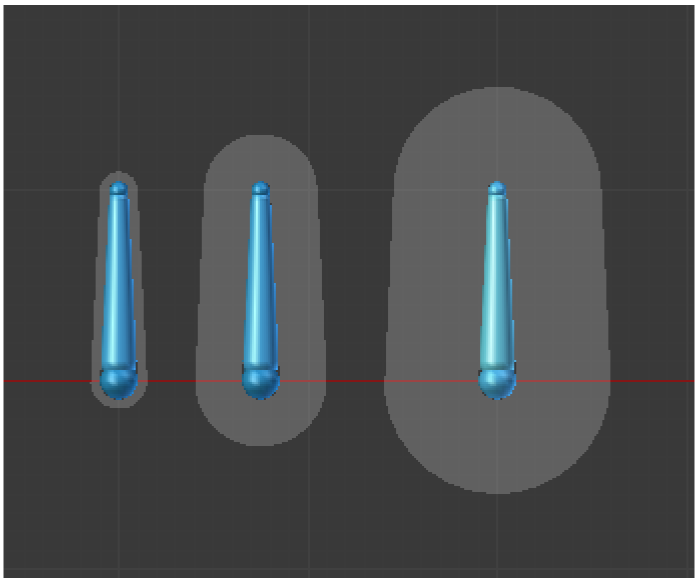


Fig. 1638: Single bone with various different envelope sizes.

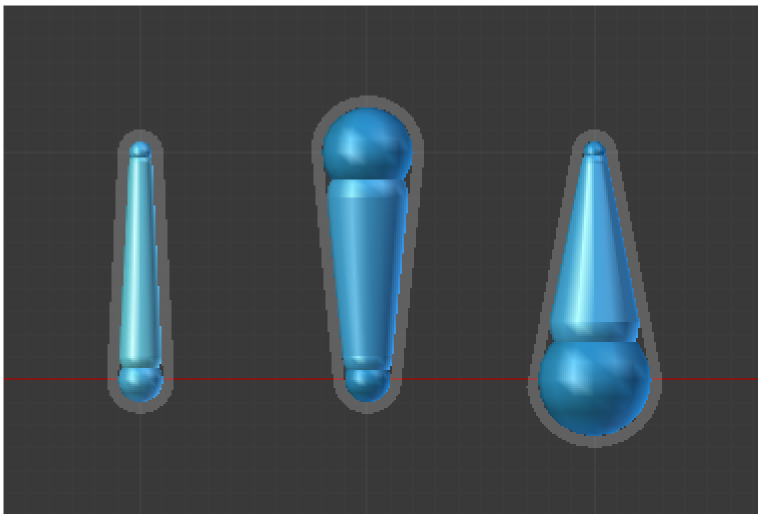


Fig. 1639: Three Armature Bones all using Envelope Weight.
The 1st with a default radius value, the two others with differing Tail and Head radius values.

Viewport Display

Reference

Mode Object and Pose Mode

Panel *Bone → Viewport Display*

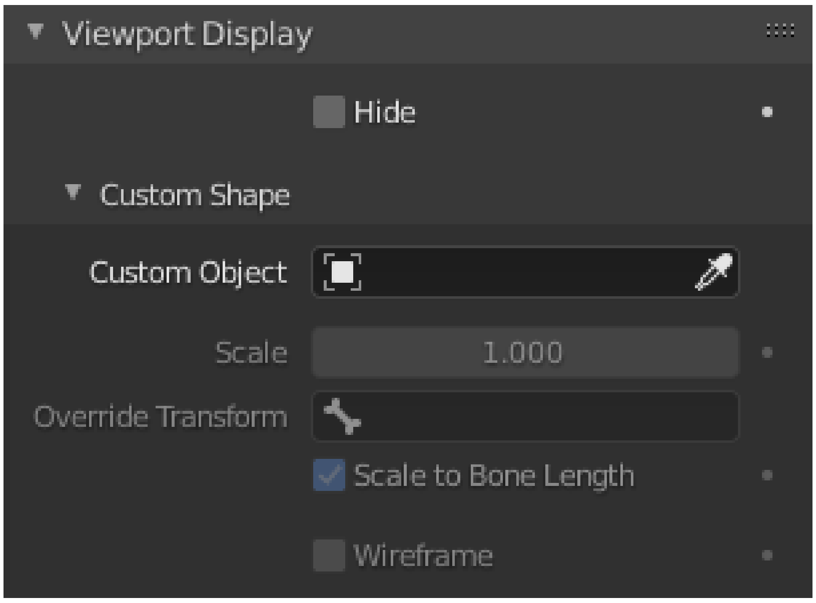


Fig. 1640: Viewport Display panel.

Display panel lets you customize the look of your bones taking the shape of another existing object.

Hide Hides the selected bone.

Custom Shape

Blender allows you to give to each bone of an armature a specific shape (in *Object Mode* and *Pose Mode*), using another object as "template". In order to be visible the *Shapes* checkbox has to be enabled (*Armature → Viewport Display* panel).

Custom Object Object that defines the custom shape of the selected bone.

Override Transform Bone that defines the display transform of the custom shape.

Scale Additional scaling factor to apply to the custom shape.

Scale to Bone Length Option not to use bones length, so that changes in Edit Mode don't resize the custom shape.

Wireframe When enabled, bone is displayed in wireframe mode regardless of the viewport display mode. Useful for non-obstructive custom bone chains.

Workflow

To assign a custom shape to a bone, you have to:

1. Switch to *Pose Mode* `Ctrl-Tab`.
2. Select the relevant bone by clicking on it.

3. Go to the *Display* panel *Custom Shape* field and select the 3D object previously created in the scene; in this example we are using a cube and a cone. You can optionally set the *At* field to another bone.

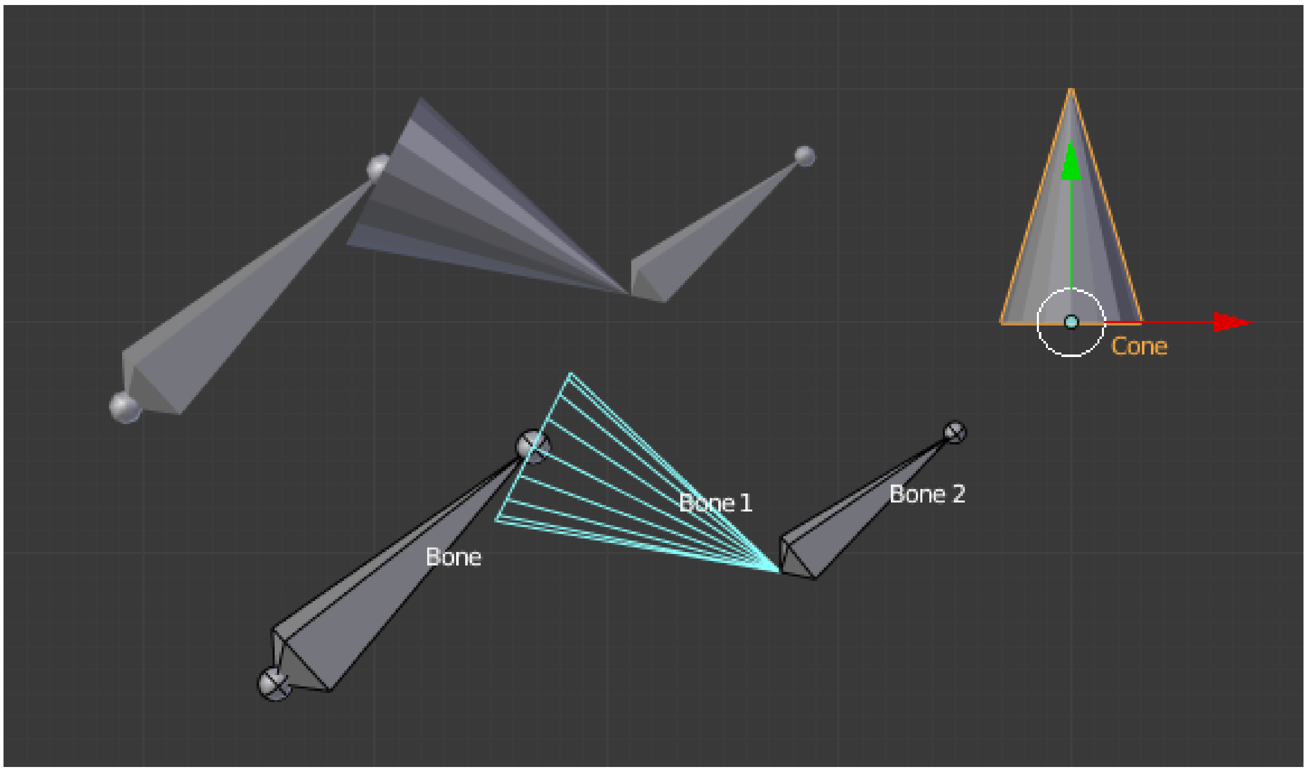


Fig. 1641: The armature with shape assigned to bone. Note the origin of the Cone object.

Note:

- These shapes will never be rendered, like any bone, they are only visible in 3D Views.
- Even if any type of object seems to be accepted by the *Object* field (meshes, curves, even metas...), only meshes really work. All other types just make the bone invisible.
- The origin of the shape object will be at the *root of the bone* (see the *bone page* for root/tip).
- The object properties of the shape are ignored (i.e. if you make a parallelepiped out of a cube by modifying its dimensions in *Object Mode*, you will still have a cube-shaped bone...).
- The "along bone" axis is the Y one, and the shape object is always scaled so that one unit stretches along the whole bone length.
- If you need to remove the custom shape of the bone, just right-click in the *Custom Shape* field and select *Reset to default value* in the pop-up menu.

So to summarize all this, you should use meshes as shape objects, with their center at their lower -Y end, and an overall Y length of 1.0 unit.

Custom Properties

Reference

Mode Pose Mode

Panel *Bone* → *Custom Properties*

See the *Custom Properties* page for more information.

Properties

Introduction

The *Armature* tab in Properties contains various panels gathering the armature settings.

Motion Paths

Reference

Mode All Modes

Panel *Armature* → *Motion Paths*

In the *Motion Paths* panel you can enable visualization of the motion path your skeleton leaves when animated.

Inverse Kinematics

Reference

Mode All Modes

Panel *Armature* → *Inverse Kinematics*

Defines the type of *IK solver* used in your animation.

Custom Properties

Reference

Mode All Modes

Panel *Armature* → *Custom Properties*

See the *Custom Properties* page for more information.

Skeleton

Reference

Mode All Modes

Panel *Armature* → *Skeleton*

In this panel you can arrange sets of bones into different layers for easier manipulation.

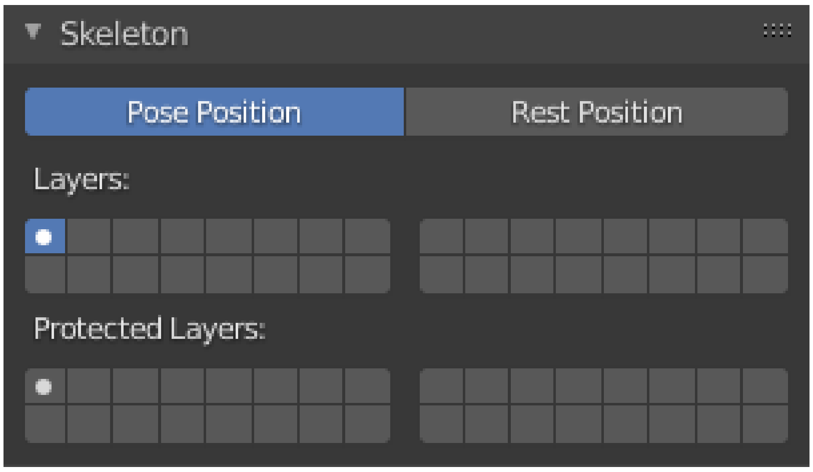


Fig. 1642: Skeleton panel.

Pose Position A radio button to switch between Pose Position and Rest Position.

In *Edit Mode*, you always see armatures in their rest position, in *Object Mode* and *Pose Mode*, by default, you see them in *Pose Position* (i.e. as it was transformed in the *Pose Mode*). If you want to see it in the rest position in all modes, select *Rest Position*.

Armature Layers Each armature has 32 layers to organize armatures by "regrouping" them into sets of bones. Only bones in active layers will be visible/editable, but they will always be effective (i.e. move objects or deform geometry), whether in an active layer or not. This property changes which layers are visible in the 3D Viewport. To show several layers at once, `Shift-LMB` on the desired layers to view. To move bones to a given layer, use *Change Bone Layers*.

Protected Layers You can lock a given bone layer for all *proxies* of your armature, i.e. all bones in this layer will not be editable.

Note: *Protected Layers* in proxy are restored to proxy settings on file reload and undo.

Viewport Display Panel

Reference

Mode All Modes

Panel *Armature → Viewport Display*

Display As This controls the way the bones appear in the 3D Viewport; you have four different visualizations you can select.

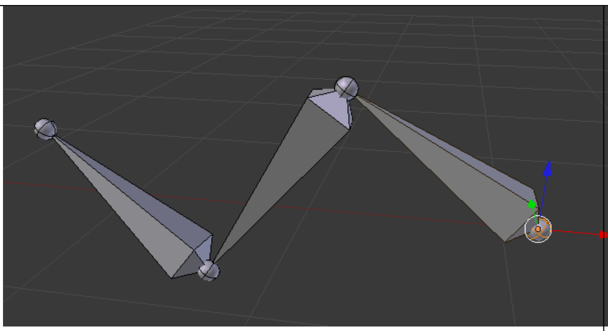


Fig. 1643: Octahedral bone display.

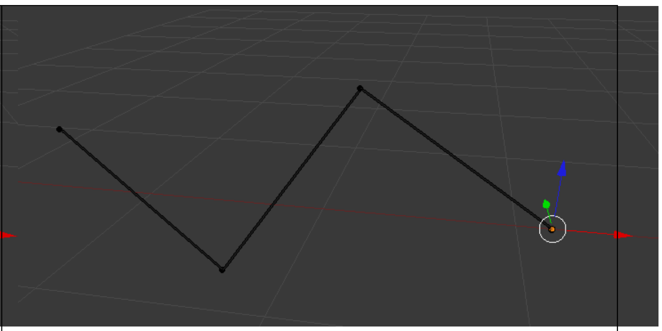


Fig. 1644: Stick bone display.

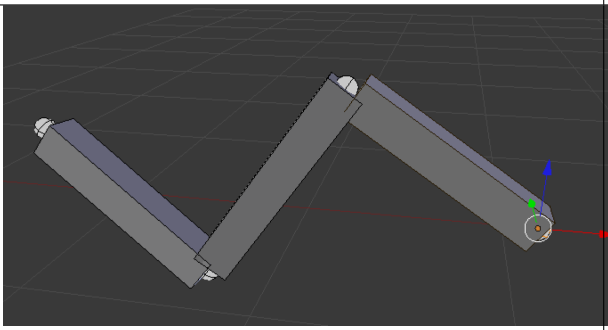


Fig. 1645: B-Bone bone display.

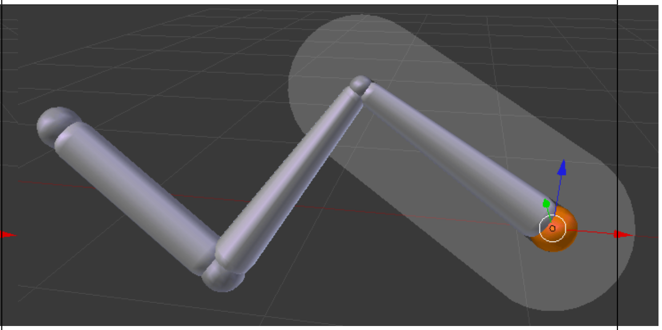


Fig. 1646: Envelope bone display.

Octahedral This is the default visualization, well suited for most of editing tasks. It materializes:

- The bone root ("big" joint) and tip ("small" joint).
- The bone "size" (its thickness is proportional to its length).
- The bone roll (as it has a square section).

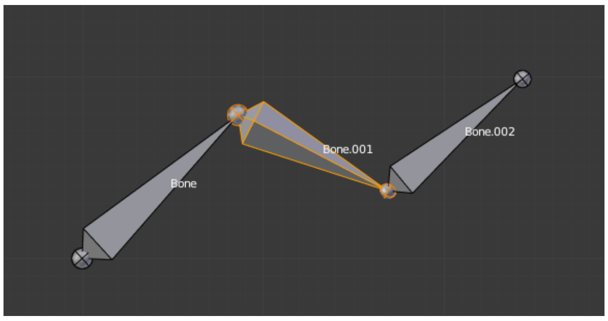


Fig. 1647: Note the 40° rolled Bone.001 bone.

Stick This is the simplest and most non-intrusive visualization. It just materializes bones by sticks of constant (and small) thickness, so it gives you no information about root and tip, nor bone size or roll angle.

B-Bone This visualization shows the curves of "smooth" multi-segmented bones; see the *Bendy Bones* for details.

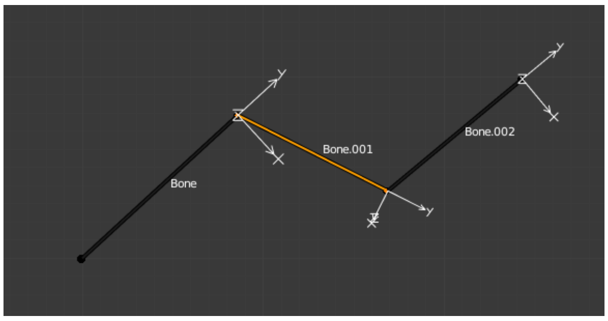


Fig. 1648: Note that Bone.001 roll angle is not visible (except by its XZ axes).

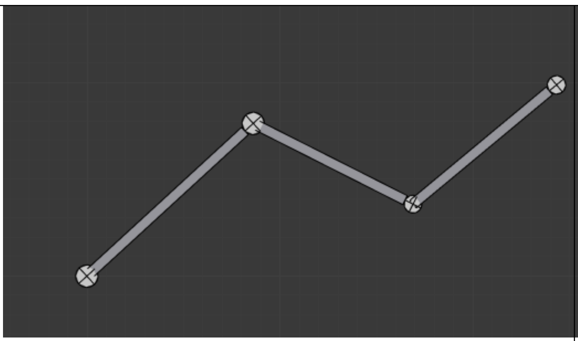


Fig. 1649: An armature of B-Bones, in Edit Mode.

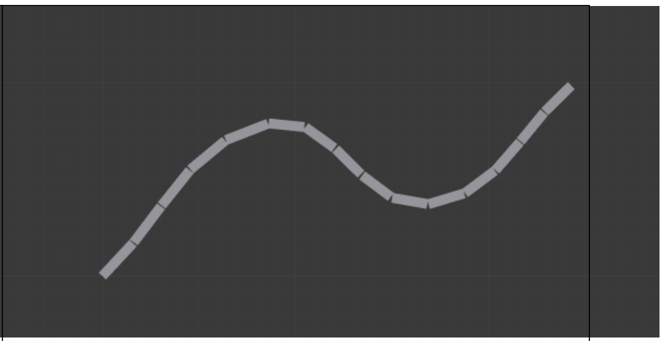
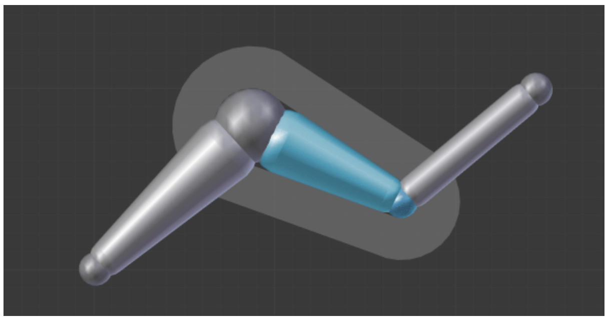


Fig. 1650: The same armature in Object Mode.

Envelope This visualization materializes the bone deformation influence. More on this in the *bone page*.



Wire This simplest visualization shows the curves of "smooth" multi-segmented bones.

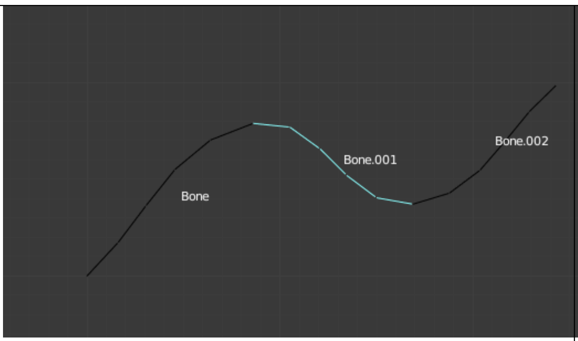


Fig. 1651: An armature of Wire, in Pose Mode.

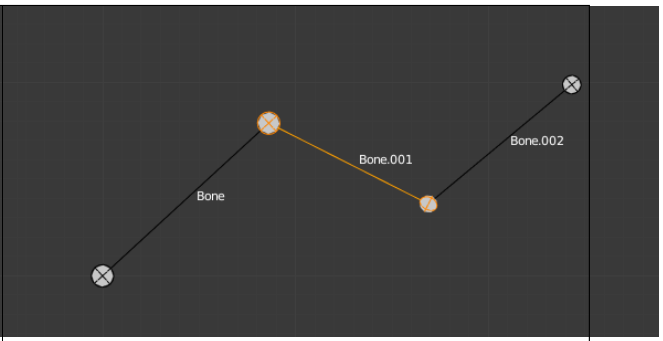


Fig. 1652: The same armature in Edit Mode.

Show

Names Displays the name of each bone.

Axes When enabled, the (local) axes of each bone are displayed (only relevant for *Edit Mode* and *Pose Mode*).

Shapes When enabled, the default standard bone shape is replaced, in *Object Mode* and *Pose Mode*, by the shape of a chosen object (see *Shaped Bones* for details).

Group Colors Use the Bone Group colors to color the bone. For more details see *Bone Groups*.

In Front When enabled, the bones of the armature will always be shown on top of the solid objects (meshes, surfaces, …). I.e. they will always be visible and selectable (this is the same option as the one found in the *Display* panel of the *Object data* tab). Very useful when not in *Wireframe* mode.

Bone Groups

Reference

Mode Pose Mode

Panel *Properties → Armature → Bone Groups*

Menu *Pose → Bone Groups → …*

This panel allows the creation, deletion and editing of Bone Groups. Bone Groups can be used for selection or to assign a color theme to a set of bones. In example to color the left parts of the rig as blue and right parts as red.

Active Bone Group The Bone Group *List view*.

Color Set

You can assign a "color theme" to a group (each bone will have these colors). Remember you have to enable the *Colors* checkbox (*Display* panel) to see these colors.

Bone Color Set A select menu.

- *Default Colors*: The default (gray) colors.
- *nn - Theme Color Set*: One of the twenty Blender presets by the theme.
- *Custom Set*: A custom set of colors, which is specific to each group.

Normal The first color field is the color of unselected bones.

Selected The second color field is the outline color of selected bones.

Active The third color field is the outline color of the active bone.

As soon as you alter one of the colors, it is switched to the *Custom Set* option.

Assign and Select

In the 3D Views, using the *Pose → Bone Groups* menu entries, and/or the *Bone Groups* pop-up menu `Ctrl-G`, you can:

Assign Assigns the selected bones to the active bone group. It is important to note that a bone can only belong to one group.

Remove Removes the selected bones from the active bone group.

Select Selects the bones in the active bone group.

Deselect Deselects the bones in the active bone group.

See also:

A single bone can be assigned to a group in the *Relations panel*.

See also:

Bones belonging to multiple groups is possible with the *Selection Sets* add-on.

Pose Library

Reference

Mode All Modes

Panel *Properties → Object Data Properties → Pose Library*

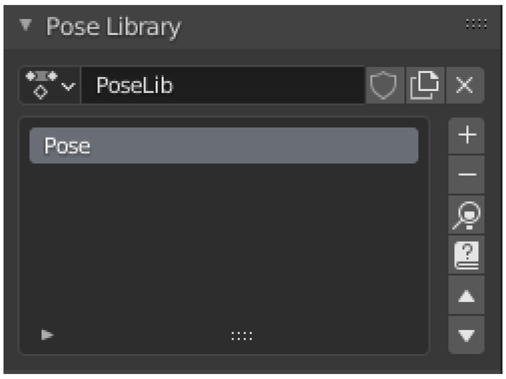


Fig. 1653: The Pose Library panel.

The pose library is a way to store pose positions so the same position can be used again later in the animation or to store different rest poses. Pose libraries are saved to *Actions*. They are not generally used as actions, but can be converted to and from. The *Pose Library* panel is used to save, apply, and manage armature poses.

See also:

Pose Library Editing.

Action A *Data-Block Menu* for Actions or pose libraries.

Pose Libraries A *List view* of poses for the active pose library.

Add + If a pose is added, a *pose marker* is created. The *Whole Character keying set* is used to determine which bones to key. If any bones are selected, only keyframes for those bones are added, otherwise all bones in the keying set are keyed. Bones that are ignored by the *Whole Character* keying set are always ignored, regardless of their selection state.

Add New Adds a new pose to the active pose library with the current pose of the armature.

Add New (Current Frame). Will add a pose to the pose library based on the current frame selected in the Timeline. In contrast to *Add New* and *Replace Existing* which automatically allocate a pose to an action frame.

Replace Existing Replace an existing pose in the active pose library with the current pose of the armature.

Apply Pose (magnifying glass icon) Applies the active pose to the selected pose bones.

Sanitize Action (book icon) Makes an action suitable for use as a pose library. This is used to convert an Action to a pose library. A pose is added to the pose library for each frame with keyframes.

Move (up/down arrow icon) Moves the pose up/down in the list.

Structure

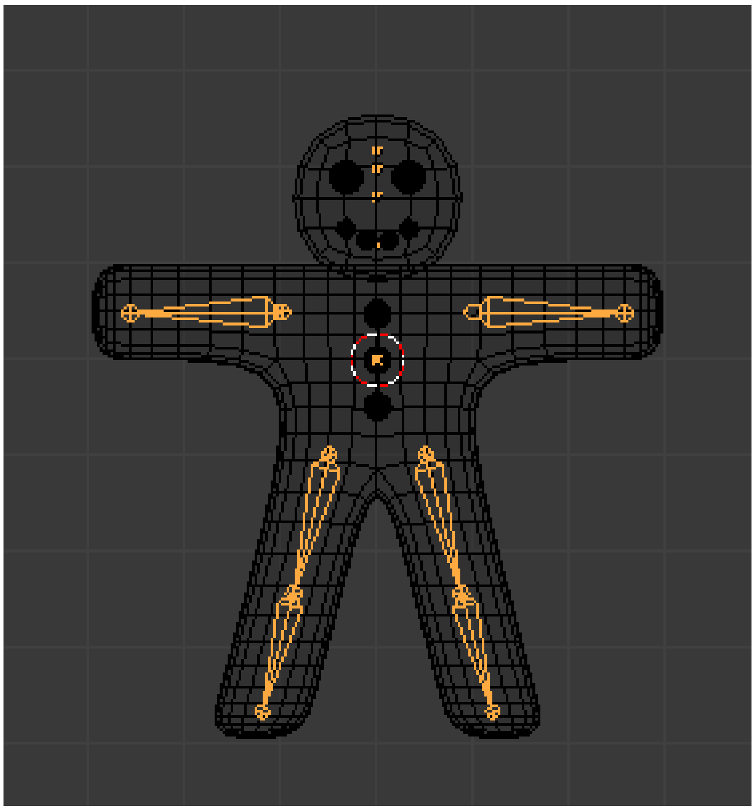


Fig. 1654: Example of a very basic armature.

Armatures mimic real skeletons. They are made out of bones, which are (by default) rigid elements. But you have more possibilities than with real skeletons: In addition to the "natural" rotation of bones, you can also move and even scale them! And your bones do not have to be connected to each other; they can be completely free if you want. However, the most natural and useful setups imply that some bones are related to others, forming so-called "chains of bones", which create some sort of "limbs" in your armature, as detailed in *Chains of Bones*.

Chains of Bones

The bones inside an armature can be completely independent from each other (i.e. the modification of one bone does not affect the others). But this is not often a useful set up: To create a leg, all bones "after" the thigh bone should move "with" it in a well-coordinated manner. This is exactly what happens in armatures by parenting a bone to the next one in the limb, you create a "chains of bones". These chains can be ramified. For example, five fingers attached to a single "hand" bone.

Bones are chained by linking the tip of the parent to the root of the child. Root and tip can be *connected*, i.e. they are always exactly at the same point; or they can be *free*, like in a standard parent-child object relationship.

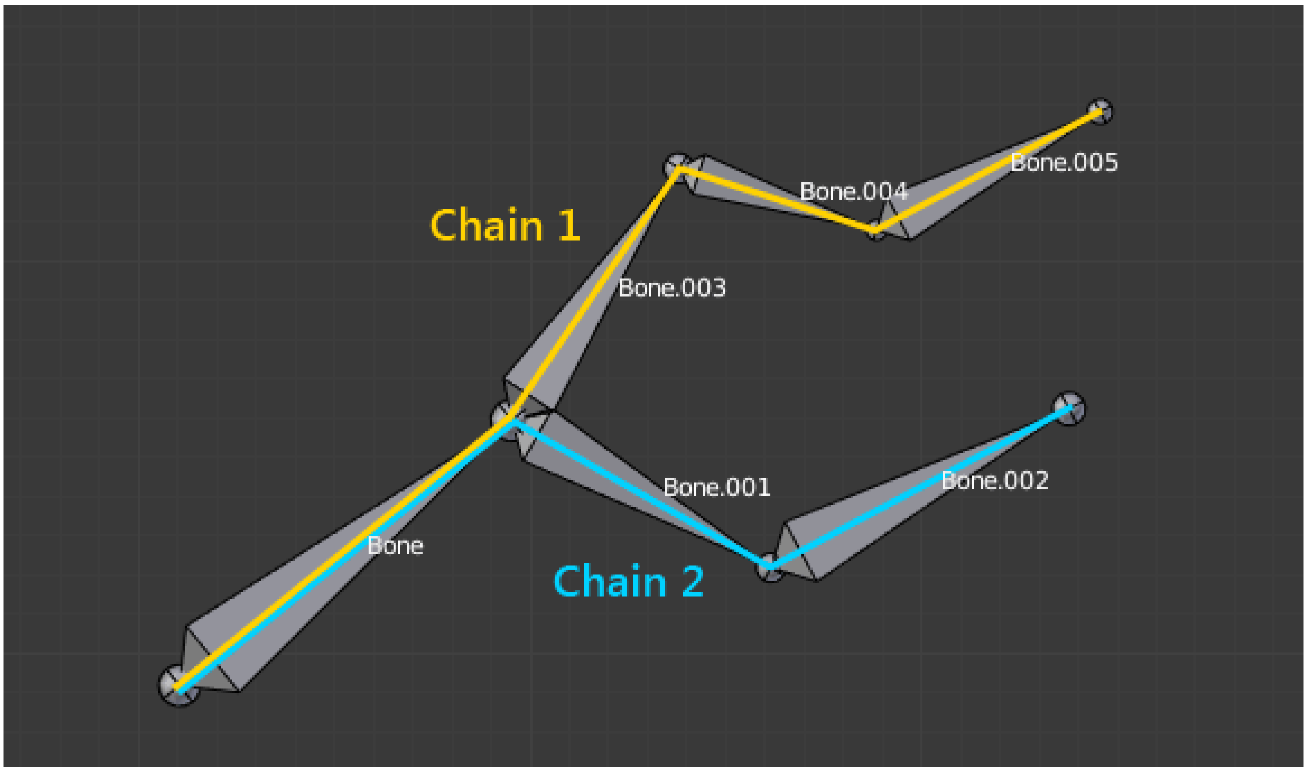


Fig. 1655: An armature with two chains of bones.

A given bone can be the parent of several children, and hence be part of several chains at the same time.

The bone at the beginning of a chain is called its *root bone*, and the last bone of a chain is the *tip bone* (do not confuse them with similar names of bones' joints!).

Chains of bones are a particularly important topic in *posing* (especially with the standard *forward kinematics* versus "automatic" *inverse kinematics* posing techniques). You create/edit them in *Edit Mode*, but except in case of connected bones, their relationships have no effect on bone transformations in this mode (i.e. transforming a parent bone will not affect its children).

The easiest way to manage bones relationships is to use the *Relations panel* in the *Bone* tab.

Skinning

Introduction

We have seen in *previous pages* how to design an armature, create chains of bones, etc. Now, having a good rig is not the final goal, unless you want to produce a "Dance Macabre" animation, you will likely want to put some flesh on your skeletons! Surprisingly, "linking" an armature to the object(s) it should transform and/or deform is called the "skinning" process…

In Blender, you have two main skinning types:

1. You can *Parent/Constrain Objects to Bones* – then, when you transform the bones in *Pose Mode*, their "children" objects are also transformed, exactly as with a standard parent/children relationship… The "children" are **never** deformed when using this method.
2. You can *Use the Armature Modifier on entire Mesh*, and then, some parts of this object to some bones inside this armature. This is the more complex and powerful method, and the only way to really deform the geometry of the object, i.e. to modify its vertices/control points relative positions.

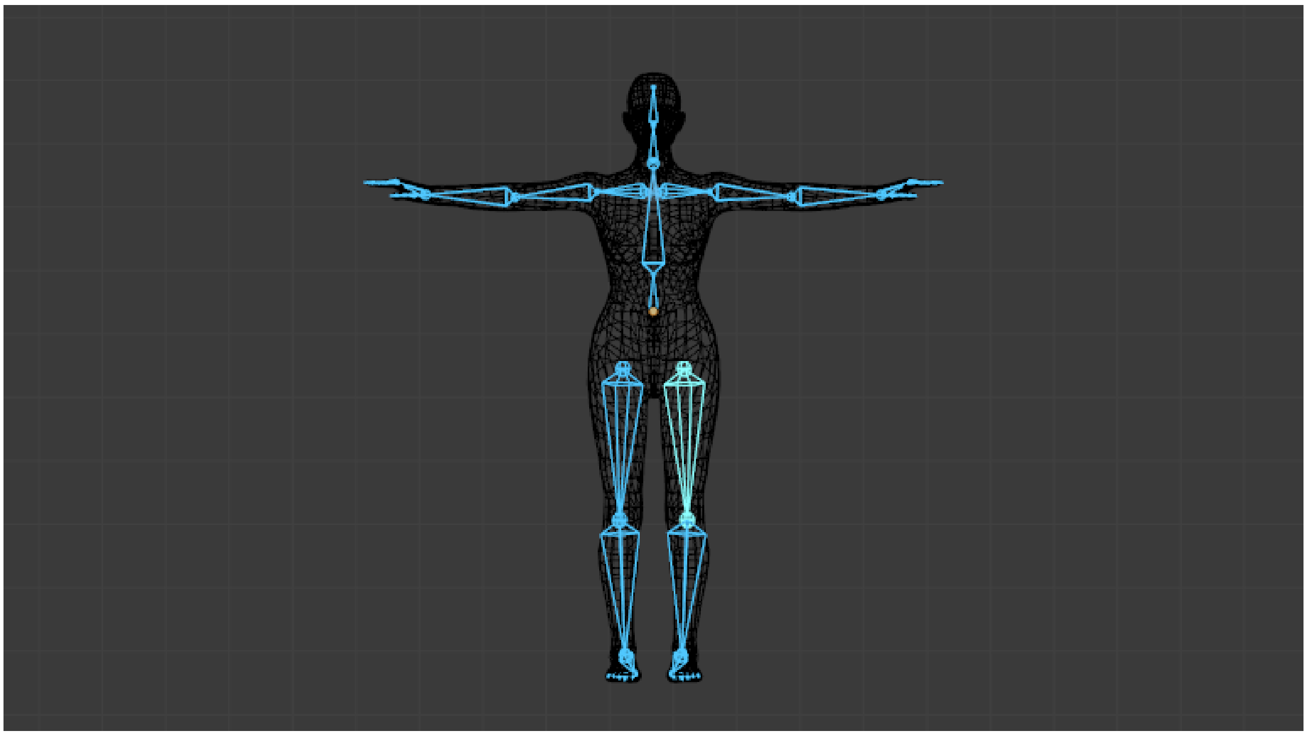


Fig. 1656: The human mesh skinned on its armature.

Hint: Retargeting

Retargeting, which is a way to apply motion-capture data (acquired from real world) to a rig, is available through add-ons and importers.

Armature Deform Parent

Reference

Mode Object Mode and Pose Mode

Menu *Object/Pose* → *Parent* → *Armature Deform*

Hotkey `Ctrl-P`

Armature Deform Parenting is a way of creating and setting up an *Armature Modifier*.

To use *Armature Deform Parenting* you must first select all the child objects that will be influenced by the armature and then lastly, select the armature object itself. Once all the child objects and the armature are selected, press `Ctrl-P` and select *Armature Deform* in the *Set Parent To* pop-up menu.

The armature will be the parent object of all the other child objects and each child object will have an Armature Modifier with the armature associated (*Object* field).

With Empty Groups

When parenting it will create empty *vertex groups* on the child objects (if they do not already exist) for and named after each deforming bone in the armature. The newly created vertex groups will

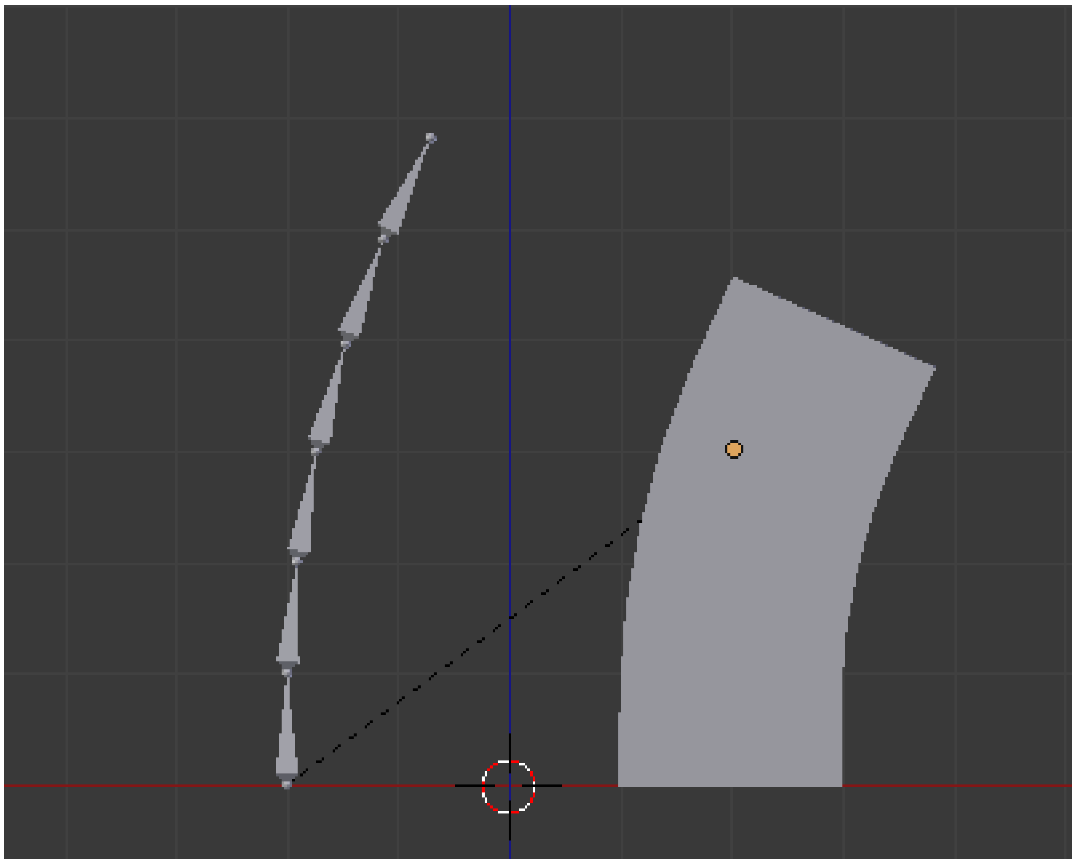


Fig. 1657: Bone associated with Mesh Object.

be empty. This means they will not have any weights assigned. Vertex groups will only be created for bones which are setup as deforming (*Properties* → *Bone* → *Deform Panel*).

You can then manually select the vertices and assign them to a particular vertex group of your choosing to have bones in the armature influence them.

Choose this option if you have already created (and weighted) all the vertex groups the mesh requires.

Example

For example, if you have an armature which consists of three bones named "BoneA", "BoneB" and "BoneC" and cube mesh called "Cube". If you parent the cube to the armature, the cube will get three new vertex groups created on it called "BoneA", "BoneB" and "BoneC". Notice that each vertex group is empty.

With Automatic Weights

With Automatic Weights parenting works similar to *With Empty Groups*, but it will not leave the vertex groups empty. It calculates how much influence a particular bone would have on vertices based on the distance from those vertices to a particular bone ("bone heat" algorithm). This influence will be assigned as weights in the vertex groups.

This method of parenting is certainly easier to setup, but it can often lead to armatures which do not deform child objects in ways you would want. Overlaps can occur when it comes to determining which bones should influence certain vertices when calculating influences for more complex armatures and child objects. Symptoms of this confusion are that when transforming

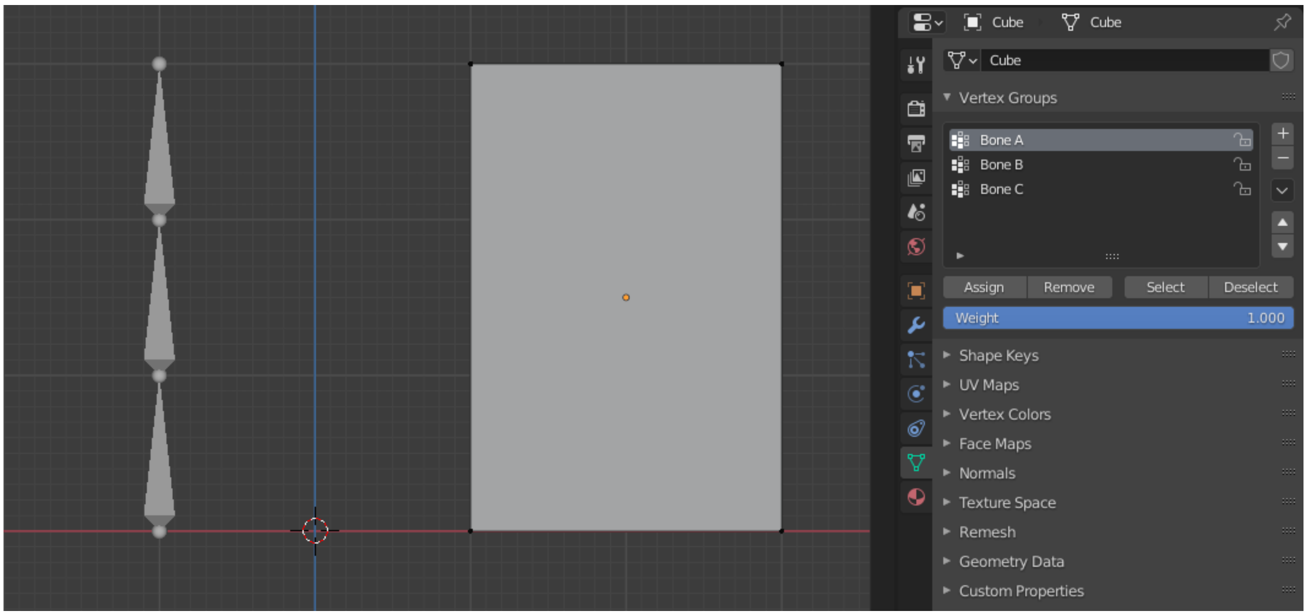


Fig. 1658: Cube in Edit Mode using Armature Deform with empty groups.

the armature in *Pose Mode*, parts of the child objects do not deform as you expect; If Blender does not give you the results you require, you will have to manually alter the weights of vertices in relation to the vertex groups they belong to and have influence in.

With Envelope Weights

Works in a similar way to *With Automatic Weights*. The difference is that the influences are calculated based on the *Bone Envelopes* settings. It will assign a weight to each vertex group the vertices that is inside its bone's influence volume, depending on their distance to this bone.

This means newly included/excluded vertices or new envelope settings will not be taken into account. You will have to apply Armature Deform With Envelope Weights parenting again.

Tip: If you want the envelope setting to be used instantly, bind the Armature Modifier to *Bone Envelopes*.

Warning: If you had defined vertex groups using same names as skinned bones, their content will be completely overridden by both *Automatic* and *Envelope Weights*. In this case *With Empty Groups* could be used instead.

See also:

Vertex Groups for Bones.

Posing

Introduction

Once an armature is *skinned* by the needed object(s), you need a way to configure the armature into positions known as poses. Basically, by transforming the bones, you deform or transform the skinned object(s). However, you will notice that you cannot do this in *Edit Mode* – remember that

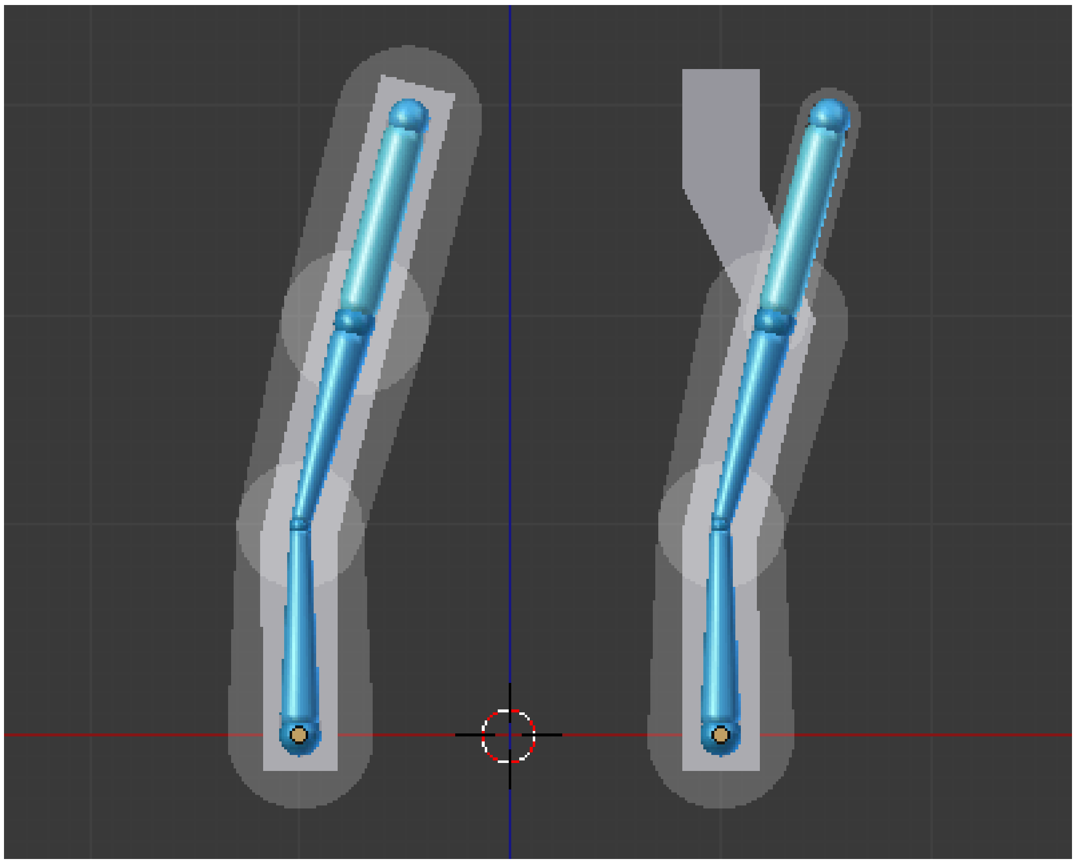


Fig. 1659: Two sets of armatures, each with three bones.

Edit Mode is used to edit the default, base, or "rest" position of an armature. You may also notice that you cannot use *Object Mode* either, as here you can only transform whole objects.

So, armatures have a third mode dedicated to the process of posing known as *Pose Mode*. In rest position (as edited in *Edit Mode*), each bone has its own position/rotation/scale to neutral values (i.e. 0.0 for position and rotation, and 1.0 for scale). Hence, when you edit a bone in *Pose Mode*, you create an offset in the transform properties, from its rest position. This may seem quite similar if you have worked with *relative shape keys* or *Delta Transformations*.

Even though it might be used for completely static purposes, posing is heavily connected with animation features and techniques. So if you are not familiar at all with animation in Blender, it might be a good idea to read the *animation chapter* first, and then come back here.

Visualization

Bone State Colors

The color of the bones are based on their state. There are six different color codes, ordered here by precedence (i.e. the bone will be of the color of the bottommost valid state):

- Gray: Default.
- Blue wireframe: in Pose Mode.
- Green: with Constraint.
- Yellow: with *IK Solver constraint*.
- Orange: with Targetless Solver constraint.

Note: When *Bone Groups* colors are enabled, the state colors will be overridden.

Selecting

Selection in *Pose Mode* is very similar to the one in *Edit Mode*, with a few deviations: You can only select *whole bones* in *Pose Mode*, not roots/tips…

Flip Active

Reference

Mode Edit Mode

Menu *Select → Flip Active*

Hotkey `Shift-Ctrl-M`

Flip the selection from one side to another.

Constraint Target

Todo.

Linked

Reference

Mode Edit Mode

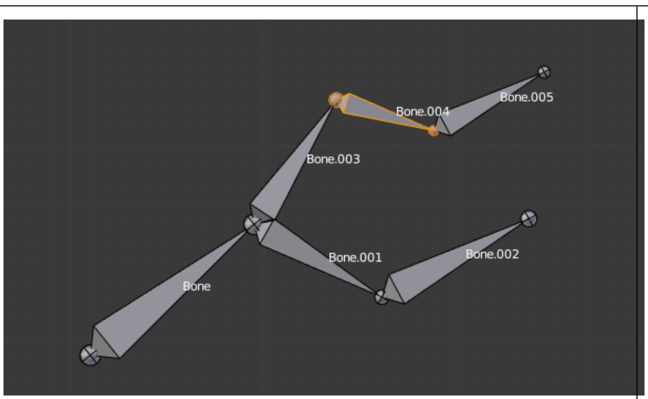
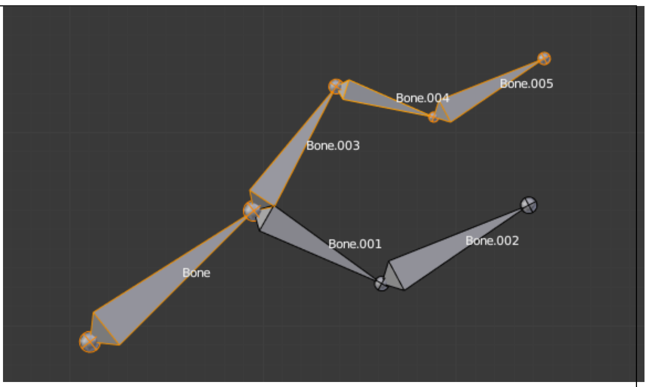
Menu *Select → Linked*

Hotkey `L`

Selects all the bones in the chain which the active (last selected) bone belongs to.

All Forks Selects all bones connected to the active bone even if the branch off from the current bone.

Table 118: Linked bones selection.

 Bone.005 Bone.004 Bone.003 Bone.001 Bone.002 Bone Fig. 1660: A single selected bone.	 Bone.005 Bone.004 Bone.003 Bone.001 Bone.002 Bone Fig. 1661: Its whole chain selected with Linked.

Select More/Less

Reference

Mode Edit Mode

Menu *Select → Select More/Less*

Parent [, Child] You can deselect the active bone and select its immediate parent or one of its children.

Extend Parent `Shift-[`, Extend Child `Shift-]` Similar to *Parent/Child* but it keeps the active bone in the selection.

Grouped

Reference

Mode Pose Mode

Menu *Select → Grouped*

Hotkey `Shift-G`

You can select bones based on their group and/or layer, through the *Select Grouped* pop-up menu `Shift-G`:

Layer To select all bones belonging to the same layer(s) as the selected ones, use the *In Same Layer* entry `Shift-G 1`.

Group To select all bones belonging to the same group(s) as the selected ones, use the *In Same Group* entry `Shift-G 2`.

Keying Set ToDo.

Select Pattern

Reference

Mode Object Mode

Menu *Select → Select Pattern…*

Selects all bones whose name matches a given pattern. Supported wild-cards: * matches everything, ? matches any single character, [abc] matches characters in "abc", and [!abc] match any character not in "abc". As an example *house* matches any name that contains "house", while floor* matches any name starting with "floor".

Case Sensitive The matching can be chosen to be case sensitive or not.

Extend When *Extend* checkbox is checked the selection is extended instead of generating a new one.

Editing

Introduction

In *Pose Mode*, bones behave like objects. So the transform actions (move, rotate, scale, etc.) are very similar to the same ones in Object Mode (all available ones are regrouped in the *Pose* → *Transform* submenu). However, there are some important specificities:

- Bones' relationships are crucial (see *Parenting*).
- The "transform center" of a given bone (i.e. its default pivot point, when it is the only selected one) is *its root*. Note by the way that some pivot point options seem to not work properly. In fact, except for the *3D Cursor* one, all others appear to always use the median point of the selection (and not e.g. the active bone's root when *Active Object* is selected, etc.).

Basic Posing

As previously noted, bones' transformations are performed based on the *Rest Position* of the armature, which is its state as defined in *Edit Mode*. This means that in rest position, in *Pose Mode*, each bone has a scale of 1.0, and null rotation and position (as you can see it in the *Transform* panel, in the 3D Views).

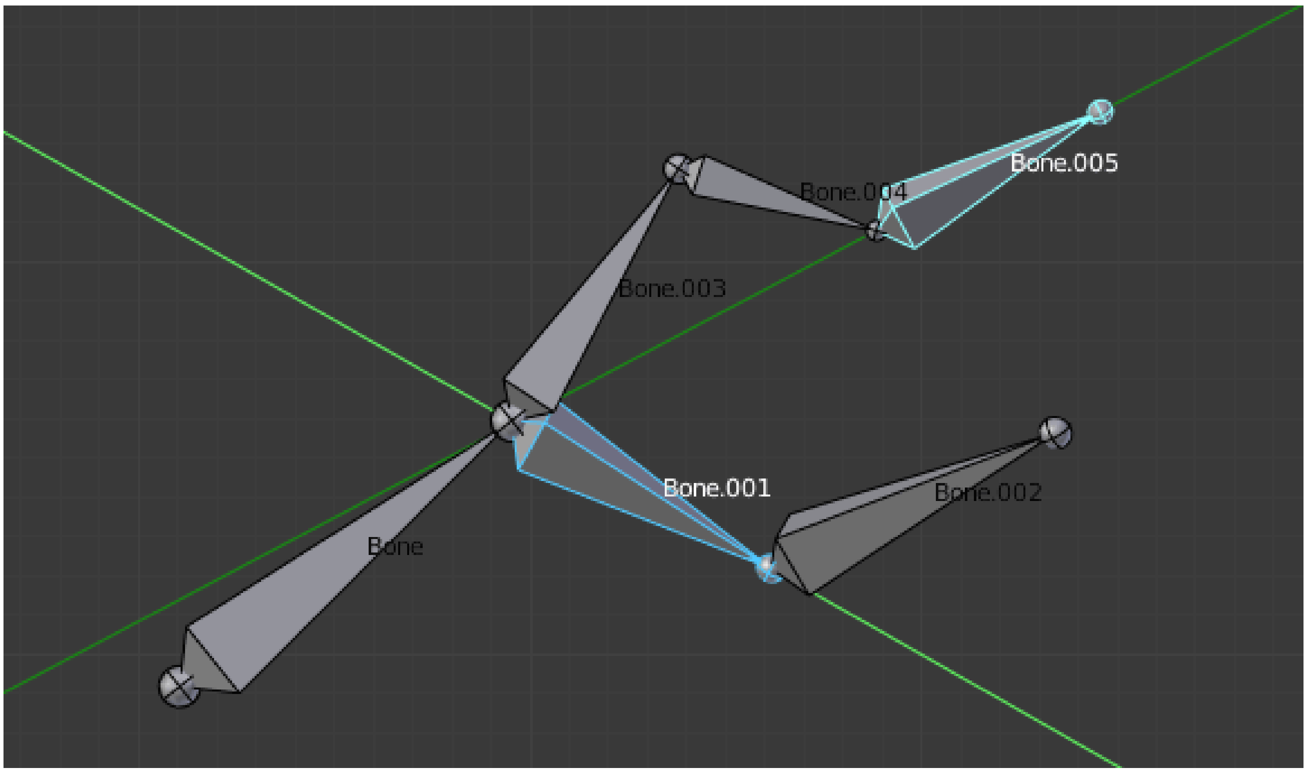


Fig. 1662: An example of a rotation locked to the local Y axis, with two bones selected.
Note that the two green lines materializing the axes are centered on the armature's center, and not each bone's root...

Moreover, the local space for these actions is the bone's own one (visible when you enable the *Axes* option of the *Armature* panel). This is especially important when using axis locking, for example, there is no specific "bone roll" tool in *Pose Mode*, as you can rotate around the bone's main axis just by locking on the local Y axis R Y Y... This also works with several bones selected; each one is locked to its own local axis!

When you pose your armature, you are supposed to have one or more objects skinned on it! And obviously, when you transform a bone in *Pose Mode*, its related objects or object's shape is moved/deformed accordingly, in real-time. Unfortunately, if you have a complex rig set-up and/or a heavy skin object, this might produce lag during interactive editing. If you experience such troubles, try enabling the *Delay Deform* button in the *Armature* panel the skin objects will only be updated once you confirm the transform operation.

Clear Transform

Reference

Mode Pose Mode

Menu *Pose → Clear Transform*

Hotkey `Ctrl-A`

Once you have transformed some bones, if you want to return to their rest position, just clear their transformations.

All Resets location, rotation, and scaling of selected bones to their default values.

Location, Rotation, Scale `Alt-G`, `Alt-R`, `Alt-S` Clears individual transforms.

Note that in *Envelope* visualization, `Alt-S` does not clear the scale, but rather scales the *Distance* influence area of the selected bones. (This is also available through the *Pose → Scale Envelope Distance* menu entry, which is only effective in *Envelope* visualization, even though it is always available…)

Reset Unkeyed Clears the transforms to their keyframe state.

Only Selected Operate on just the selected or all bones.

Apply

Reference

Mode Pose Mode

Menu *Pose → Apply*

Hotkey `Ctrl-A`

Pose as Rest Pose Conversely, you may define the current pose as the new rest pose (i.e. "apply" current transformations to the *Edit Mode*). When you do so, the skinned objects/geometry is **also** reset to its default, undeformed state, which generally means you will have to skin it again.

Pose Selected as Rest Pose Same as *Pose as Rest Pose* but only applies to selected bones.

Visual Transform to Pose Applies the position of the bone after *Constraints*; allowing the constraints to be deleted and the bones will remain in their constrained positions.

Assign Custom Property Values as Default Assign the current values of custom properties as their defaults, for use as part of the rest pose state in NLA track mixing.

In-Betweens

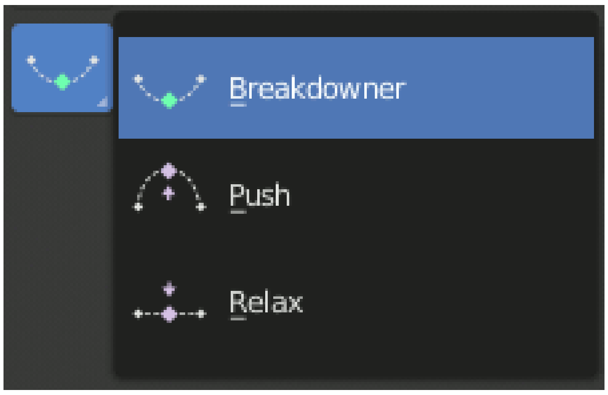


Fig. 1663: In-Betweens Tools.

There are several tools for editing poses in an animation.

There are also in *Pose Mode* a bunch of armature-specific editing options/tools, like *auto-bones naming*, *properties switching/enabling/disabling*, etc., that were already described in the armature editing pages. See the links above…

Push Pose from Rest Pose

Reference

Mode Pose Mode

Menu *Pose → In-Betweens → Push Pose from Rest Pose*

Similar to *Push Pose from Breakdown* but interpolates the pose to the rest position instead. Only one keyframe is needed for this tool unlike two for the other.

Relax Pose to Rest Pose

Reference

Mode Pose Mode

Menu *Pose → In-Betweens → Relax Pose to Rest Pose*

Similar to *Relax Pose to Breakdown* but works to bring the pose back to the rest position instead. Only one keyframe is needed for this tool unlike two for the other.

Push Pose from Breakdown

Reference

Mode Pose Mode

Tool *Toolbar → In-Betweens Tools → Push*

Menu *Pose → In-Betweens → Push Pose from Breakdown*

Hotkey `Ctrl-E`

Push Pose interpolates the current pose by making it closer to the next keyframed position.

Relax Pose to Breakdown

Reference

Mode Pose Mode

Tool *Toolbar → In-Betweens Tools → Relax*

Menu *Pose → In-Betweens → Relax Pose to Breakdown*

Hotkey `Alt-E`

Relax pose is somewhat related to the above topic, but it is only useful with keyframed bones. When you edit such a bone (and hence take it "away" from its "keyed position"), using this tool will progressively "bring it back" to its "keyed position", with smaller and smaller steps as it comes near it.

Pose Breakdowner

Reference

Mode Pose Mode

Tool *Toolbar region → In-Betweens Tools → Breakdowner*

Menu *Pose → In-Betweens → Pose Breakdowner*

Hotkey `LMB`-drag

Creates a suitable breakdown pose on the current frame.

The Breakdowner tool can be constrained to work on specific transforms and axes, by pressing the following keys while the tool is active:

- `G`, `R`, `S`: move, rotate, scale
- `B`: Bendy bones
- `C`: custom properties
- `X`, `Y`, `Z`: to the corresponding axes

Propagate

Reference

Mode Pose Mode

Menu *Pose → Propagate*

Hotkey `Alt-P`

The Propagate tool copies the pose of the selected bones on the current frame over to the keyframes delimited by the *Termination Mode*. It automates the process of copying and pasting.

Termination Mode Modes which determine how it decides when to stop overwriting keyframes.

> **While Held** The most complicated of the modes available, as it tries to guess when to stop propagating by examining the pauses in the animation curves per control (i.e. all F-curves for a bone, instead of per F-curve).
>
> **To Next Keyframe** Simply copies the pose to the first keyframe after (but not including any keyframe on) the current frame.
>
> **To Last Keyframe** Will simply replace the last keyframe (i.e. making action cyclic).
>
> **Before Frame** To all keyframes between current frame and the *End frame* option. This option is best suited for use from scripts due to the difficulties in setting this frame value, though it is possible to set this manually via the *Adjust Last Operation* panel if necessary.
>
> **Before Last Keyframe** To all keyframes from current frame until no more are found.
>
> **On Selected Keyframes** Will apply the pose of the selected bones to all selected keyframes.
>
> **On Selected Markers** To all keyframes occurring on frames with Scene Markers after the current frame.

End Frame Defines the upper-bound for the frame range within which keyframes will be affected (with the lower bound being the current frame).

Copy/Paste Pose

Reference

> **Mode** Pose Mode
>
> **Menu** *Pose* → *Copy Pose*, *Pose* → *Paste Pose*, *Pose* → *Paste Pose Flipped*
>
> **Hotkey** `Ctrl-C`, `Ctrl-V`, `Shift-Ctrl-V`

Blender allows you to copy and paste a pose, either through the *Pose* menu, or by using hotkeys.

Copy Pose Copy the current pose of selected bones into the pose buffer.

Paste Pose Paste the buffered pose to the currently posed armature.

Paste Pose Flipped Paste the *X axis mirrored* buffered pose to the currently posed armature.

Here are important points:

- This tool works at the Blender session level, which means you can use it across armatures, scenes, and even files. However, the pose buffer is not saved, so you lose it when you close Blender.
- There is only one pose buffer.
- Only the selected bones are taken into account during copying (i.e. you copy only selected bones' pose).
- During pasting, on the other hand, bone selection has no importance. The copied pose is applied on a per-name basis (i.e. if you had a `forearm` bone selected when you copied the pose, the `forearm` bone of the current posed armature will get its pose when you paste it – and if there is no such named bone, nothing will happen…).
- What is copied and pasted is in fact the position, rotation or scale of each bone, in its own space. This means that the resulting pasted pose might be very different from the originally copied one, depending on:
  - The rest position of the bones.
  - And the current pose of their parents.

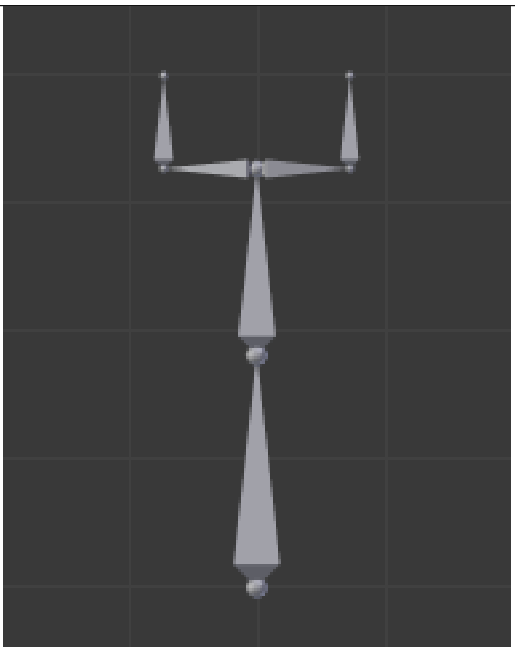
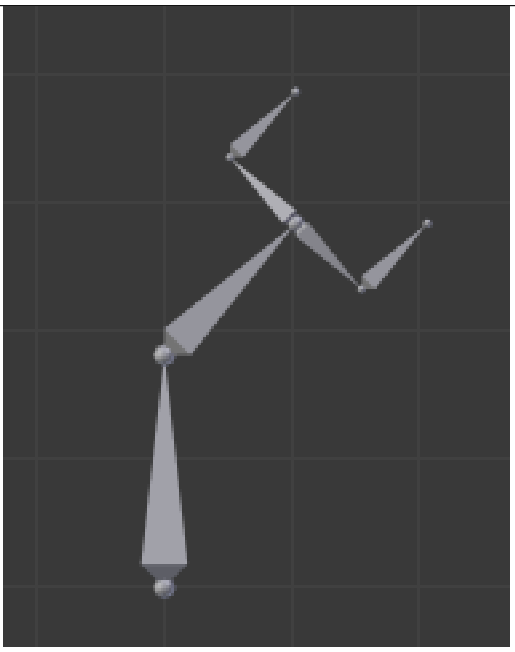
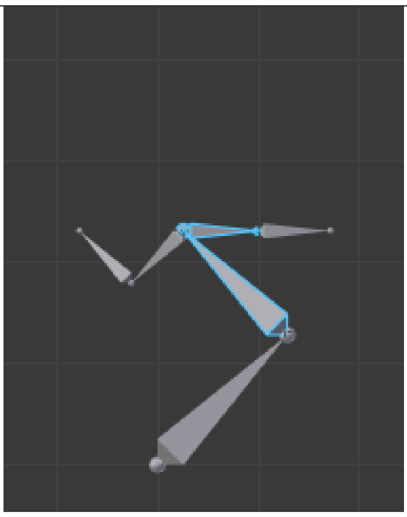
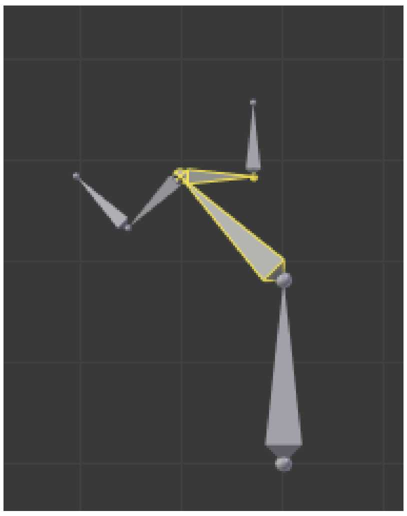
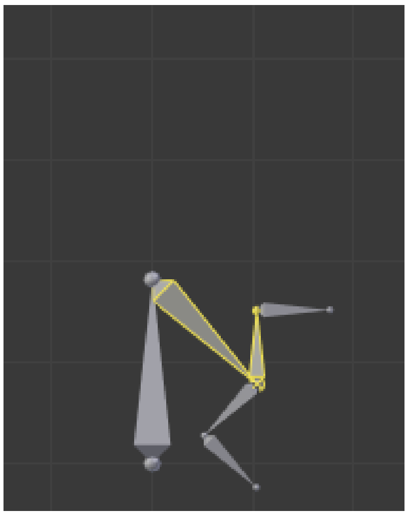
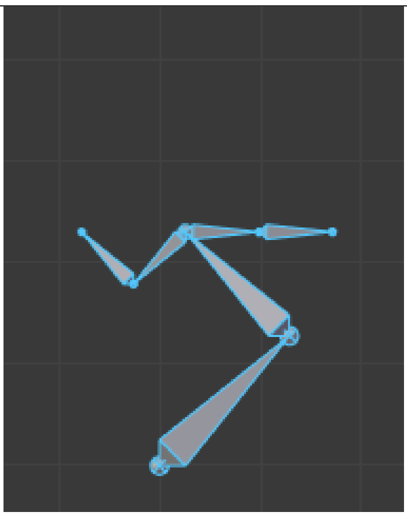
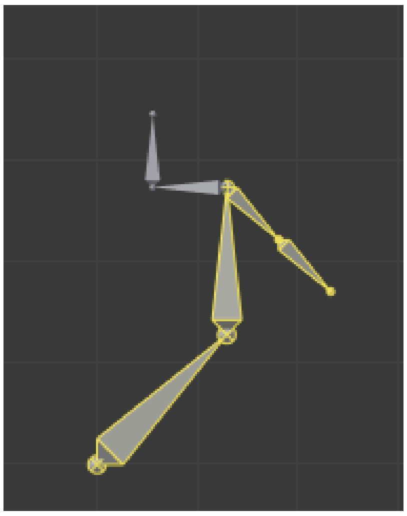
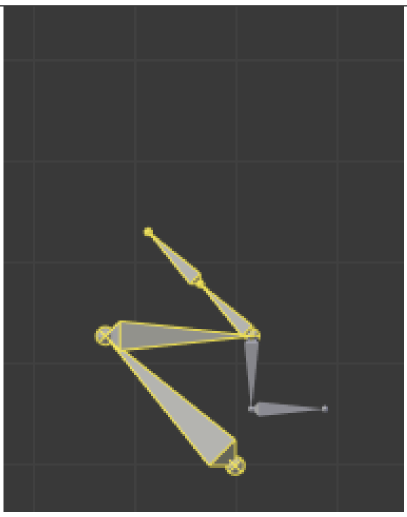
 Fig. 1664: The rest position of the original armature.	 Fig. 1665: The rest position of the destination armature.

Table 119: Examples of pose copy/paste.

 Fig. 1666: The first copied pose (note that only two bones are selected and hence copied).	 Fig. 1667: The pose pasted on the destination armature.	 Fig. 1668: The pose mirror-pasted on the destination armature.
 Fig. 1669: The same pose as above is copied, but this time with all bones selected.	 Fig. 1670: The pose pasted on the destination armature.	 Fig. 1671: The pose mirror-pasted on the destination armature.

Pose Library

Reference

Mode Pose Mode

Menu *Pose* → *Pose Library*

The pose library is a way to store pose positions so the same position can be used again later in the animation or to store different rest poses. *Pose libraries* are saved to *Actions*. They are not generally used as actions, but can be converted to and from.

See also:

Pose Library Properties.

Browse Poses

Reference

Mode Edit Mode

Menu *Pose → Pose Library → Browse Poses*

Hotkey `Alt-L`

Interactively browse poses in the 3D Viewport. After running the operator, cycle through poses using the `Left` and `Right` arrow keys. The name of the pose being previewed is displayed in the header region. After the desired pose is selected using `Return` or `LMB` to make it the active pose; to cancel browsing, use `Esc` or `RMB`.

Pose Index of the pose to apply (-2 for no change, -1 to use the active pose).

Add Pose

Reference

Mode Edit Mode

Menu *Pose → Pose Library → Add Pose*

Hotkey `Shift-L`

If a pose is added, a *pose marker* is created. The *Whole Character keying set* is used to determine which bones to key. If any bones are selected, only keyframes for those bones are added, otherwise all bones in the keying set are keyed. Bones that are ignored by the *Whole Character* keying set are always ignored, regardless of their selection state.

Add New Adds a new pose to the active pose library with the current pose of the armature.

Add New (Current Frame). Will add a pose to the pose library based on the current frame selected in the Timeline. In contrast to *Add New* and *Replace Existing* which automatically allocate a pose to an action frame.

Replace Existing Replace an existing pose in the active pose library with the current pose of the armature.

Rename Pose

Reference

Mode Edit Mode

Menu *Pose → Pose Library → Rename Pose*

Hotkey `Shift-Ctrl-L`

Changes the name of the specified pose from the active pose library.

New Pose Name The new name for the pose.

Pose The pose action to rename.

Remove Pose

Reference

Mode Edit Mode

Menu *Pose → Pose Library → Remove Pose*

Hotkey `Shift-Alt-L`

Deletes the specified pose from the active pose library.

Flip Quats

Reference

Mode Pose Mode

Menu *Pose → Flip Quats*

Flip quaternion values to achieve desired rotations, while maintaining the same orientations.

Show/Hide

Reference

Mode All Modes

Panel *Properties → Bone → Viewport Display*

Menu *… → Show/Hide*

You do not have to use bone layers to show/hide some bones. As with objects, vertices or control points, you can use `H`:

- `H` will hide the selected bone(s).
- `Shift-H` will hide all bones *but the selected one(s)*.
- `Alt-H` will show all hidden bones.

You can also use the *Hide* checkbox of the *Bone tab → Viewport Display panel*.

Note that hidden bones are specific to a mode, i.e. you can hide some bones in *Edit Mode*, they will still be visible in *Pose Mode*, and vice versa. Hidden bones in *Pose Mode* are also invisible in *Object Mode*. And in *Edit Mode*, the bone to hide must be fully selected, not just its root or tip.

Tool Settings

Pose Options

Auto IK

Reference

Mode Pose Mode

Panel *Sidebar → Tool → Pose Options → Auto IK*

The auto IK option in the Sidebar enables a temporary IK constraint when posing bones. The chain acts from the tip of the selected bone to root of the upper-most parent bone. Note that this mode lacks options, and only works by applying the resulting transform to the bones in the chain.

X-Axis Mirror

Reference

Mode Edit and Pose Mode

Panel *Sidebar → Tool → Options → X-Axis Mirror*

This option enables automatic mirroring of editing actions along the X axis. You can enable this option in the *Tool tab → Options panel*, while the armature is selected in *Edit Mode*. When you have pairs of bones of the same name with just a different "side suffix" (e.g. ".R"/".L", or "_right"/"_left" ...), once this option is enabled, each time you transform (move, rotate, scale...) a bone, its "other side" counterpart will be transformed accordingly, through a symmetry along the armature local X axis. As most rigs have at least one axis of symmetry (animals, humans, ...), it is an easy way to keep the model symmetrical.

Relative Mirror

Reference

Mode Edit and Pose Mode

Panel *Sidebar → Tool → Options → Relative Mirror*

Accounts for any relative transformations when using *X-Axis Mirror*.

See also:

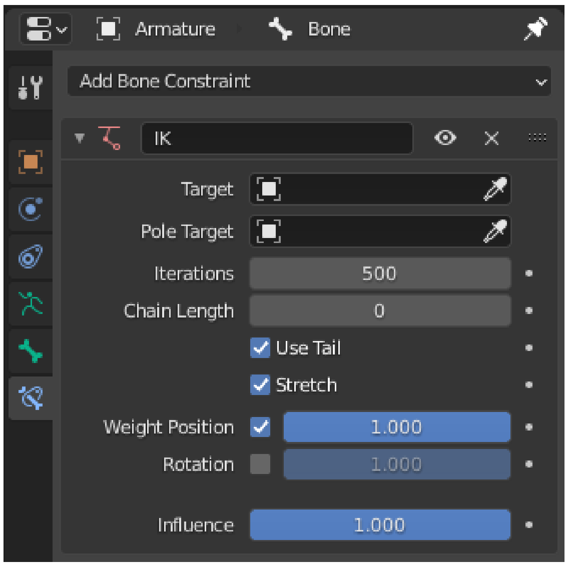
Naming bones.

Known Limitations

Relative Mirror is not supported with *Auto IK* enabled.

Bone Constraints

Introduction



Target
Pole Target
Iterations 500
Chain Length 0
✓ Use Tail
✓ Stretch
Weight Position ✓ 1.000
Rotation 1.000
Influence 1.000

Fig. 1672: The Bone Constraints Properties in Pose Mode, with an Inverse Kinematics constraint added to the active bone.

As bones behave like objects in *Pose Mode*, they can also be constrained. This is why the *Constraints* tab is shown in both *Object Mode* and *Edit Mode*. This panel contains the constraints *of the active bone* (its name is displayed at the top of the panel, in the *To Bone:...* static text field).

Constraining bones can be used to control their degree of freedom in their pose transformations, using e.g. the *Limit* constraints. You can also use constraints to make a bone track another object/bone (inside the same object, or in another armature), etc. And the *inverse kinematics feature* is also mainly available through the *IK Solver* constraint, which is specific to bones.

For example, a human elbow cannot rotate backward (unless the character has broken their arm), nor to the sides, and its forward and roll rotations are limited in a given range. (E.g. depending on the rest position of your elbow, it may be from (0 to 160) or from (-45 to 135).)

So you should apply a *Limit Rotation* constraint to the forearm bone (as the elbow movement is the result of rotating the forearm bone around its root).

Using bones in constraints, either as owners or as targets, is discussed in detail in the *constraints pages*.

Inverse Kinematics

Introduction

IK simplifies the animation process, and makes it possible to make more advanced animations with lesser effort.

IK allows you to position the last bone in a bone chain and the other bones are positioned automatically. This is like how moving someone's finger would cause their arm to follow it. By normal posing techniques, you would have to start from the root bone, and set bones sequentially until you reach the tip bone: When each parent bone is moved, its child bone would inherit its location and rotation. Thus making tiny precise changes in poses becomes harder farther down the chain, as you may have to adjust all the parent bones first.

This effort is effectively avoided by use of IK.

Automatic IK

Reference

Mode Pose Mode

Panel *Sidebar region → Tool → Pose Options*

Automatic IK is a tool for quick posing, it can be enabled in the Toolbar in the 3D Viewport, when in Pose Mode. When the Auto IK option is enabled, translating a bone will activate inverse kinematics and rotate the parent bone, and the parent's parent, and so on, to follow the selected bone. The IK chain can only extend from a child to a parent bone if the child is *connected* to it.

The length of the chain is increased (if there is a connected parent available to add to it) with `Ctrl-PageUp` or `Ctrl-WheelDown`, and decreased with `Ctrl-PageDown` or `Ctrl-WheelUp`. However, the initial chain length is 0, which effectively means follow the connections to parent bones as far as possible, with no length limit. So pressing `Ctrl-PageUp` the first time sets the chain length to 1 (move only the selected bone), and pressing `Ctrl-PageDown` at this point sets it back to 0 (unlimited) again. Thus, you have to press `Ctrl-PageUp` *more than once* from the initial state to set a finite chain length greater than 1.

This is a more limited feature than using an IK constraint, which can be configured, but it can be useful for quick posing.

IK Constraints

IK is mostly done with bone constraints. They work by the same method but offer more choices and settings. Please refer to these pages for detail about the settings for the constraints:

- *IK Solver*
- *Spline IK*

Armature IK Panel

Reference

Mode Pose Mode

Panel *Properties → Armature → Inverse Kinematics*

This panel is used to select the IK Solver type for the armature: *Standard* or *iTaSC*. Most the time people will use the *Standard* IK solver.

Standard

TODO.

iTaSC

iTaSC stands for instantaneous Task Specification using Constraints.

iTaSC uses a different method to compute the Jacobian, which makes it able to handle other constraints than just end effectors position and orientation: iTaSC is a generic multi-constraint IK solver. However, this capability is not yet fully exploited in the current implementation, only two other types of constraints can be handled: Distance in the Cartesian space, and Joint Rotation in the joint space. The first one allows maintaining an end effector inside, at, or outside a sphere centered on a target position, the second one is the capability to control directly the rotation of a bone relative to its parent. Those interested in the mathematics can find a short description of the method used to build the Jacobian here.

iTaSC accepts a mix of constraints, and multiple constraints per bone: the solver computes the optimal pose according to the respective weights of each constraint. This is a major improvement from the current constraint system where constraints are solved one by one in order of definition so that conflicting constraints overwrite each other.

Precision The maximum variation of the end effector between two successive iterations at which a pose is obtained that is stable enough and the solver should stop the iterations. Lower values means higher precision on the end effector position.

Iterations The upper bound for the number of iterations.

Solver Selects the inverse Jacobian solver that iTaSC will use.

SDLS (Selective Damped Least Square) Computes the damping automatically by estimating the level of 'cancellation' in the armature kinematics. This method works well with the Copy Pose constraint but has the drawback of damping more than necessary around the singular pose, which means slower movements. Of course, this is only noticeable in Simulation mode.

DLS (Damped Least Square) Computes the damping manually which can provide more reactivity and more precision.

Damping Max Maximum amount of damping. Smaller values means less damping, hence more velocity and better precision but also more risk of oscillation at singular pose. 0 means no damping at all.

Damping Epsilon Range of the damping zone around singular pose. Smaller values means a smaller zone of control and greater risk of passing over the singular pose, which means oscillation.

Note: *Damping* and *Epsilon* must be tuned for each armature. You should use the smallest values that preserve stability.

Note:

- The SDLS solver does not work together with a Distance constraint. You must use the DLS solver if you are going to have a singular pose in your animation with the Distance constraint.
- Both solvers perform well if you do not have a singular pose.

Animation

In Animation mode, iTaSC operates like an IK solver: it is stateless and uses the pose from F-curves interpolation as the start pose before the IK convergence. The target velocity is ignored and the solver converges until the given precision is obtained. Still the new solver is usually faster than the old one and provides features that are inherent to iTaSC: multiple targets per bone and multiple types of constraints.

Simulation

The Simulation mode is the stateful mode of the solver: it estimates the target's velocity, operates in a 'true time' context, ignores rotation from keyframes (except via a joint rotation constraint) and builds up a state cache automatically.

Reiteration

Never The solver starts from the rest pose and does not reiterate (converges) even for the first frame. This means that it will take a few frames to get to the target at the start of the animation.

Initial The solver starts from the rest pose and re-iterates until the given precision is achieved, but only on the first frame (i.e. a frame which doesn't have any previous frame in the cache). This option basically allows you to choose a different start pose than the rest pose and it is the default value. For the subsequent frames, the solver will track the target by integrating the joint velocity computed by the Jacobian solver over the time interval that the frame represents. The precision of the tracking depends on the feedback coefficient, number of substeps and velocity of the target.

Always The solver re-iterates on each frame until the given precision is achieved. This option omits most of the iTaSC dynamic behavior: the maximum joint velocity and the continuity between frames is not guaranteed anymore in compensation of better precision on the end effector positions. It is an intermediate mode between *Animation* and real-time *Simulation*.

Auto Step Use this option if you want to let the solver set how many substeps should be executed for each frame. A substep is a subdivision on the time between two frames for which the solver evaluates the IK equation and updates the joint position. More substeps means more processing but better precision on tracking the targets. The auto step algorithm estimates the optimal number of steps to get the best trade-off between processing and precision. It works by estimation of the nonlinearity of the pose and by limiting the amplitude of joint variation during a substep. It can be configured with next two parameters:

Min Proposed minimum substep duration (in second). The auto step algorithm may reduce the substep further based on joint velocity.

Max Maximum substep duration (in second). The auto step algorithm will not allow substep longer than this value.

Steps If Auto Step is disabled, you can choose a fixed number of substeps with this parameter. Substep should not be longer than 10 ms, which means the number of steps is 4 for a 25 fps animation. If the armature seems unstable (vibrates) between frames, you can improve the stability by increasing the number of steps.

Feedback Coefficient on end effector position error to set corrective joint velocity. The time constant of the error correction is the inverse of this value. However, this parameter has little effect on the dynamic of the armature since the algorithm evaluates the target velocity in any case. Setting this parameter to 0 means 'opening the loop': the solver tracks the velocity but not the position; the error will accumulate rapidly. Setting this value too high means an excessive amount of correction and risk of instability. The value should be in the range 20-100. Default value is 20, which means that tracking errors are corrected in a typical time of 100-200 ms. The feedback coefficient is the reason why the armature continues to move slightly in Simulation mode even if the target has stopped moving: the residual error is progressively suppressed frame after frame.

Max Velocity Indicative maximum joint velocity in radian per second. This parameter has an important effect on the armature dynamic. Smaller value will cause the armature to move slowly and lag behind if the targets are moving rapidly. You can simulate an inertia by setting this parameter to a low value.

Bone IK Panel

Reference

Mode Pose Mode

Panel *Properties → Bone → Inverse Kinematics*

This panel is used to control how the *Pose Bones* work in the IK chain.

IK Stretch Stretch influence to IK target.

Lock Disallow movement around the axis.

Stiffness Stiffness around the axis. Influence disabled if using *Lock*.

Limit Limit movement around the axis.

iTaSC Solver

If the *iTaSC IK Solver* is used, the bone IK panel changes to add these additional parameters.

Control Rotation Activates a joint rotation constraint on that bone. The pose rotation computed from Action or UI interaction will be converted into a joint value and passed to the solver as target for the joint. This will give you control over the joint while the solver still tracks the other IK targets. You can use this feature to give a preferred pose for joints (e.g. rest pose) or to animate a joint angle by playing an action on it.

Weight The importance of the joint rotation constraint based on the constraints weight in case all constraints cannot be achieved at the same time. For example, if you want to enforce strongly the joint rotation, set a high weight on the joint rotation constraint and a low weight on the IK constraints.

Arm Rig Example

This arm uses two bones to overcome the twist problem for the forearm. IK locking is used to stop the forearm from bending, but the forearm can still be twisted manually by pressing R Y Y in *Pose Mode*, or by using other constraints.

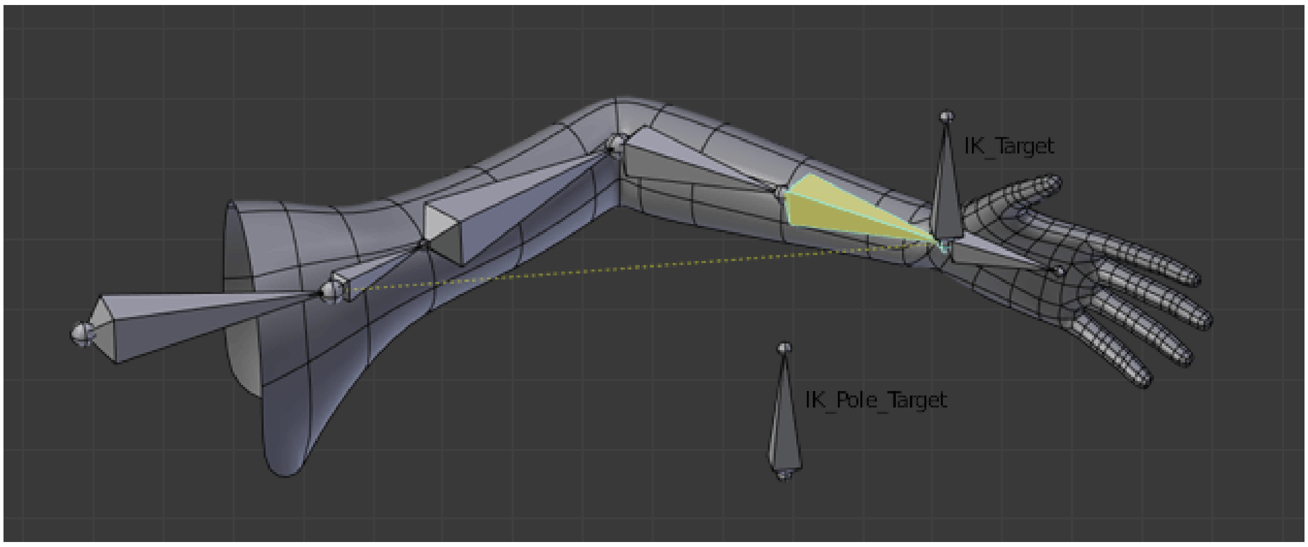


Fig. 1673: IK Arm Example.

Note that, if a *Pole Target* is used, IK locking will not work on the root boot.

Spline IK

Spline IK is a constraint which aligns a chain of bones along a curve. By leveraging the ease and flexibility of achieving aesthetically pleasing shapes offered by curves and the predictability and well integrated control offered by bones, Spline IK is an invaluable tool in the riggers' toolbox. It is particularly well suited for rigging flexible body parts such as tails, tentacles, and spines, as well as inorganic items such as ropes.

Full description of the settings for the spline IK can be found on the *Spline IK* page.

Basic Setup

The Spline IK Constraint is not strictly an *Inverse Kinematics* method (i.e. IK Constraint), but rather a *Forward Kinematics* method (i.e. normal bone posing). However, it still shares some characteristics of the IK Constraint, such as operating on multiple bones, not being usable for Objects, and being evaluated after all other constraints have been evaluated. It should be noted that if a Standard IK chain and a Spline IK chain both affect a bone at the same time the Standard IK chain takes priority. Such setups are best avoided though, since the results may be difficult to control.

To setup Spline IK, it is necessary to have a chain of connected bones and a curve to constrain these bones to:

1. With the last bone in the chain selected, add a *Spline IK* Constraint from the Bone Constraints tab in the Properties.
2. Set the *Chain Length* setting to the number of bones in the chain (starting from and including the selected bone) that should be influenced by the curve.
3. Finally, set the *Target* field to the curve that should control the curve.

Congratulations, the bone chain is now controlled by the curve.

Settings and Controls

For the precise list of options, see *Spline IK* constraint. This section is intended to introduce the workflow.

Roll Control

To control the *Roll* of the Spline IK chain, the standard methods of rotating the bones in the chain along their local Y axes still apply. For example, start at the farthest bone and simply rotate the bones in the chain around their local Y axes to adjust the roll of the chain from that point onward.

Applying Copy Rotation constraints on the bones also works.

Note: There are a couple of limitations to consider:

- Bones do not inherit a curve's *tilt* value to control their roll.
- There is no way of automatically creating a twisting effect where a dampened rotation is inherited up the chain. Consider using *Bendy Bones* instead.

Offset Controls

The entire bone chain can be made to follow the shape of the curve while still being able to be placed at an arbitrary point in 3D space when the *Chain Offset* option is enabled. By default, this option is not enabled, and the bones will be made to follow the curve in its untransformed position.

Length Control

The *Y Scale Mode* setting can be used to choose the way bones are scaled length-wise. The available options allow stretching the bone chain to fit the curve, using the pre-IK scaling, or doing neither. In addition, the scale of the curve Object affects the result.

Thickness Controls

The thickness of the bones in the chain is controlled using the constraint's *XZ Scale Mode* setting. This setting determines the method used for determining the scaling on the X and Z axes of each bone in the chain.

The available modes are:

None This option keeps the X and Z scaling factors as 1.0.

Volume Preserve The X and Z scaling factors are taken as the inverse of the Y scaling factor (length of the bone), maintaining the 'volume' of the bone.

Bone Original This options just uses the X and Z scaling factors the bone would have after being evaluated in the standard way.

In addition to these modes, there is an option, *Use Curve Radius*. When this option is enabled, the average radius of the radii of the points on the curve where the joints of each bone are placed, are used to derive X and Z scaling factors. This allows the scaling effects, determined using the modes above, to be tweaked as necessary for artistic control.

Tips for Nice Setups

- For optimal deformations, it is recommended that the bones are roughly the same length, and that they are not too long, to facilitate a better fit to the curve. Also, bones should ideally be created in a way that follows the shape of the curve in its 'rest pose' shape, to minimize the problems in areas where the curve has sharp bends which may be especially noticeable when stretching is disabled.
- For control of the curve, it is recommended that hooks (in particular, Bone Hooks) are used to control the control points of the curve, with one hook per control point. In general, only a few control points should be needed for the curve (e.g. one for every 3-5 bones offers decent control).
- The type of curve used does not really matter, as long as a path can be extracted from it that could also be used by the Follow Path Constraint. This really depends on the level of control required from the hooks.

2.8.4 Lattice

Lattice – or commonly called deformation cage outside of Blender. A lattice consists of a three-dimensional non-renderable grid of vertices. Its main use is to apply a deformation to the object it controls with a *Lattice Modifier*. If the object is parented with *Lattice Deform* a Lattice Modifier is automatically applied.

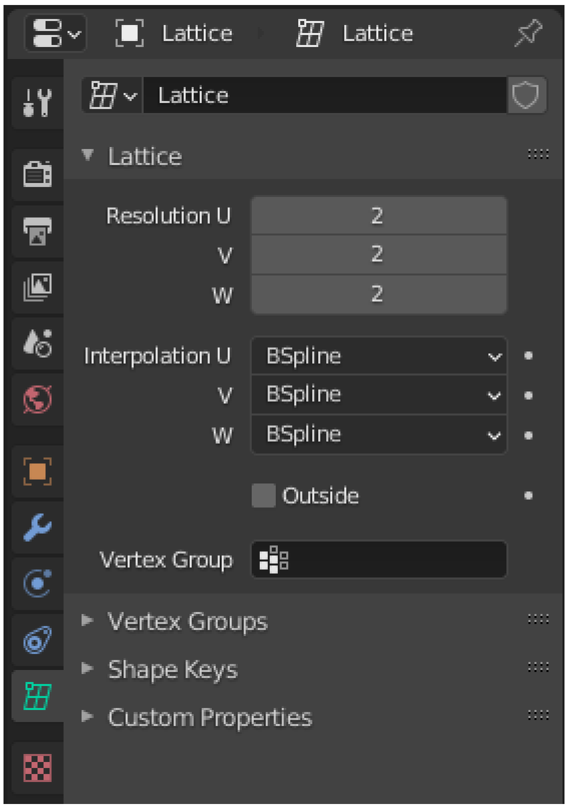
Editing

Flip (Distortion Free) Mirrors the vertices displacement from their base position.

U, V, W

Make Regular Resets the whole lattice to a regular grid, where the cells are scaled to one cubic unit.

Properties



Lattice > Lattice

Lattice

▼ Lattice

Resolution U 2
V 2
W 2

Interpolation U BSpline
V BSpline
W BSpline

☐ Outside

Vertex Group

▶ Vertex Groups
▶ Shape Keys
▶ Custom Properties

Fig. 1674: Lattice properties.

Lattice A *Data-Block Menu*.

Lattice

Points Rate of subdivision in the axes:

U, V, W

Interpolation Type Selector for each axis. See *Different types of interpolation*..

Linear, Cardinal, Catmull-Rom, B-Spline

Outside Takes only the vertices on the surface of the lattice into account.

Vertex Group The strength of the influence assigned as a weight to the individual vertices in the selected vertex group.

Usage

The lattice should be scaled and moved to fit around your object in Object Mode. Any scaling applied to the object in Edit Mode will result in the object deforming. This includes applying its scale with `Ctrl-A` as this will achieve the same result as scaling the lattice in Edit Mode, and therefore the object.

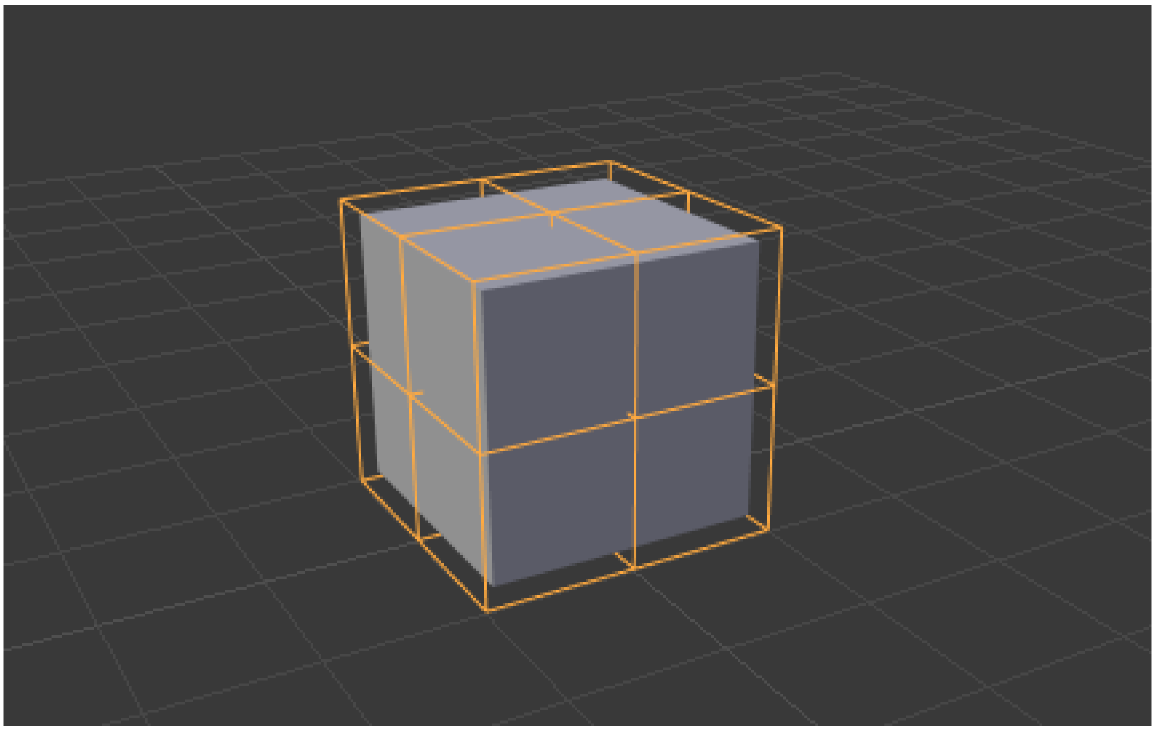


Fig. 1675: Lattice around the cube object in Object Mode.

2.8.5 Constraints

Introduction

Constraints are a way to control an object's properties (e.g. its location, rotation, scale), using either plain static values (like the *"limit" ones*), or another object, called "target" (like e.g. the *"copy" ones*).

Even though constraints are useful in static projects, their main usage is obviously in animation.

- You can control an object's animation through the targets used by its constraints (this is a form of indirect animation). Indeed, these targets can then control the constraint's owner's properties, and hence, animating the targets will indirectly animate the owner.
- You can animate constraints' settings. e.g. the *Influence* or when using an armature's bone as target, animate where along this bone (between root and tip) lays the real target point.

They can make the eyes of a tennis player track a tennis ball bouncing across the court, allow the wheels on a bus to all rotate together, help a dinosaur's legs bend at the knee automatically, and make it easy for a hand to grip the hilt of a sword and the sword to swing with the hand.

Constraints, in Blender, work with *Objects* and *Bones*. Read about using constraints in rigging in the *Armature chapter*.

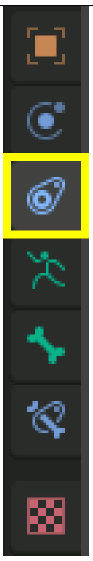


Fig. 1676: Object

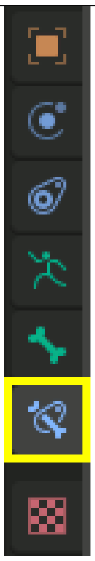


Fig. 1677: Bone

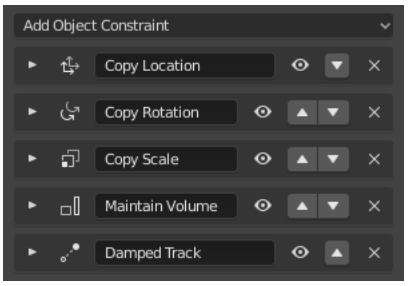


Fig. 1678: The Constraint Stack is evaluated from top to bottom.

Constraints work in combination with each other to form a Constraint Stack.

Tips

Constraints are a fantastic way to add sophistication and complexity to a rig.

But be careful not to rush in too quickly, piling up constraint upon constraint until you lose all sense of how they interact with each other.

Start simply. Get to know a single constraint inside and out. *Copy Location Constraint* is a good first constraint to explore it also has an animation example. Take the time to understand every fundamental concept behind it, and the other constraints will make far more sense.

Interface

Adding/Removing a Constraint

What is described on this page about Object Constraints can be also be applied on Bone Constraints.

Tab

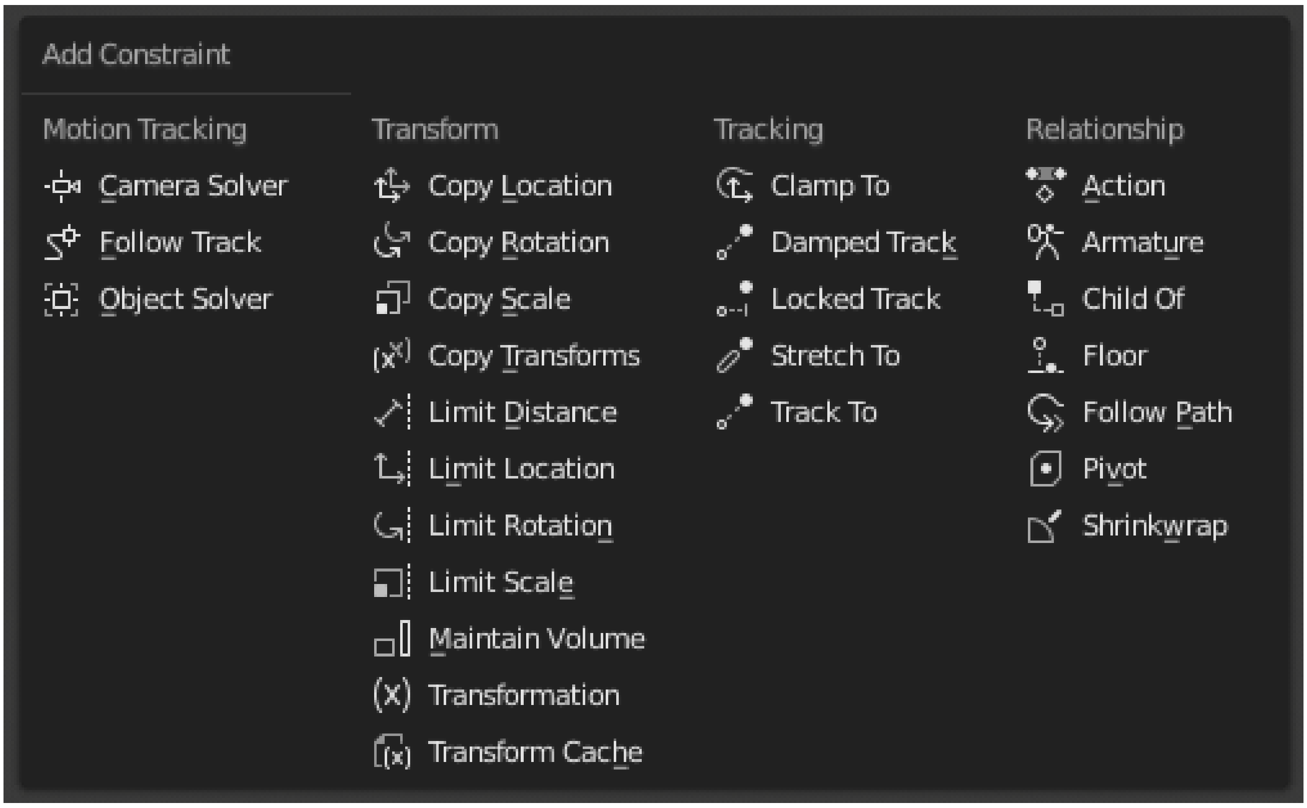
Reference

Mode Object Mode

Menu *Properties → Constraint tab*

To add a constraint click on the *Add Object Constraint* menu in the Constraints tab.

To remove a constraint click on the “X” button in the *header*.



Menu

Add Constraint (with Targets)

Reference

Mode Object Mode and Pose Mode

Menu *Object → Constraint → Add Constraint (with Targets)*

Adds a constraint to the active object. The type of constraint must be chosen from a pop-up menu, though it can be changed later from the *Add Constraint (with Targets) Adjust Last Operation* panel. If there is an other object selected besides the active one, that object will be the constraint target (if the chosen constraint accepts targets).

When using a bone from another armature as the target for a constraint, the tool will look inside the non-active armature and use its active bone, provided that armature is in Pose Mode.

Copy Constraints to Selected Objects

Reference

Mode Object Mode and Pose Mode

Menu *Object → Constraint → Copy Constraints to Selected Objects*

Copies the active object Constraints to the rest of the selected objects.

Clear Object Constraints

Reference

Mode Object Mode and Pose Mode

Panel *Object → Constraint → Clear Object Constraints*

Removes all Constraints of the selected object(s).

Track

Reference

Mode Object Mode

Panel *Object → Track*

These tools add a tracking constraint to the selected objects; the target object of the constraint will be the active object, which won't have a constraint added.

- *Damped Track Constraint*
- *Track To Constraint*
- *Lock Track Constraint*

Clear Track Removes all Damped Track, Track To and Lock Track Constraints from the selected objects.

Clear and Keep Transformation (Clear Track) Removes all Track Constraint from the selected objects, while keeping the final transform caused by them.

Header

Every constraint has a header. The interface elements of the header are explained below using a Copy Location constraint as an example.

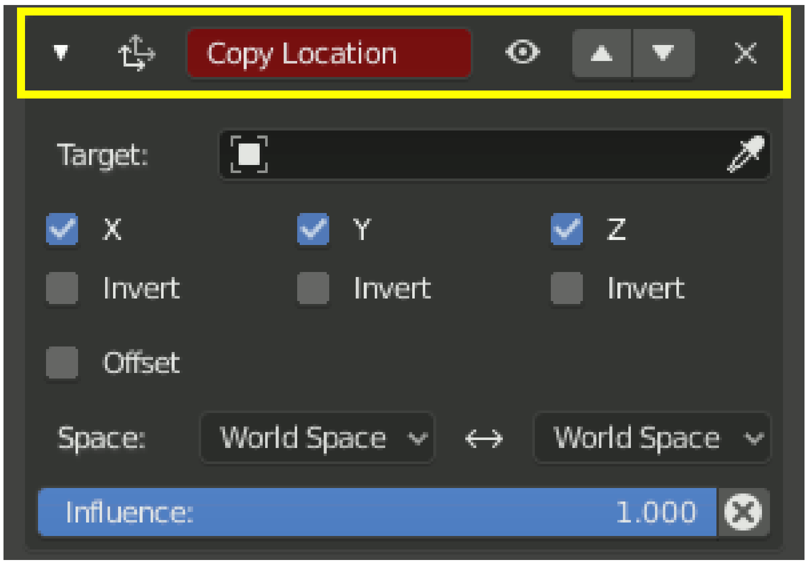


Fig. 1679: A Header sits at the top of every constraint.

Expand (down/right arrow icon) Show or Hide the settings of the constraint. Tidy up the *constraint stack* by hiding constraints that do not currently need attention. Constraints will continue to affect the scene even when hidden.

Icon The constraint type icon.

Name Give the constraint a meaningful name in this text field, which describes its purpose. Meaningful names help you and your team members understand what each constraint is supposed to do.

The *red* background is a warning that the constraint is not yet functional. The background will turn *gray* when the constraint is functioning. When this Copy Location constraint has a valid target in the *target field* it will turn gray and begin to function.

Mute (eye icon) Enable or Disable the constraint. Disabling a constraint will stop its affect on the scene.

Disabling a constraint is useful for turning off a constraint without losing all of its settings. Disabling means you can enable the constraint at a later time with the settings intact. Disabling is similar to setting the *Influence* to 0.0.

Delete `X` `X`, `Delete` Delete the constraint from the stack. The settings will be lost. The constraint will no longer affect the final outcome of the stack.

Move `::::` Move a constraint up or down in the *constraint stack*. Since the stack is evaluated from top to bottom, moving a constraint in the stack can significantly affect the final outcome of the stack.

- If there is only one constraint in the stack, the arrows will not be displayed.
- If the constraint is at the top of the stack, only the down arrow will be displayed.
- If the constraint is at the bottom of the stack, only the up arrow will be displayed.

Common

Target

The Target *Data ID* field lets you link the constraint to a Target object of your choosing. This link provides data to the constraint so that it can begin to function. For example, the Copy Location Constraint needs location data to function. Fill in the Target field, and the Copy Location constraint will begin to use location data from the Target object.

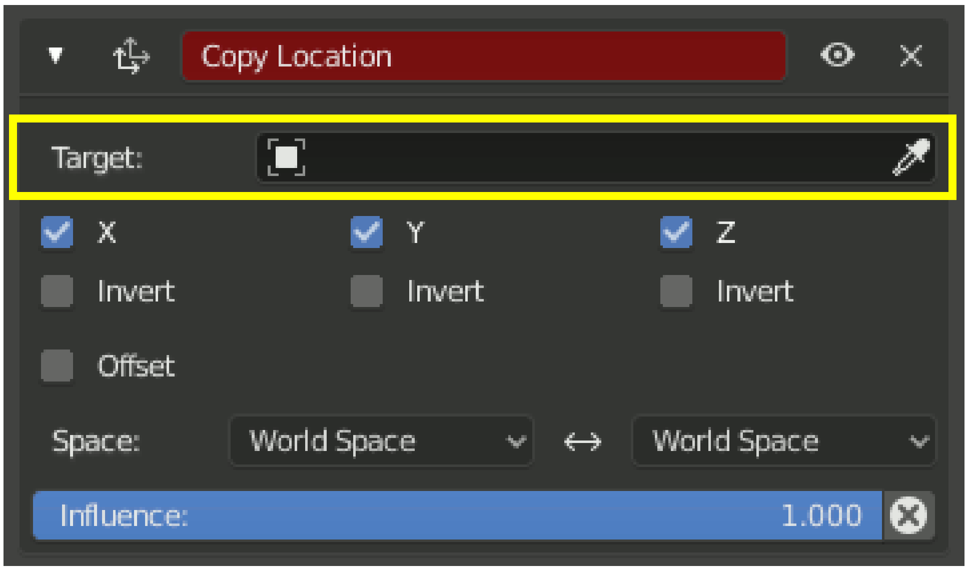
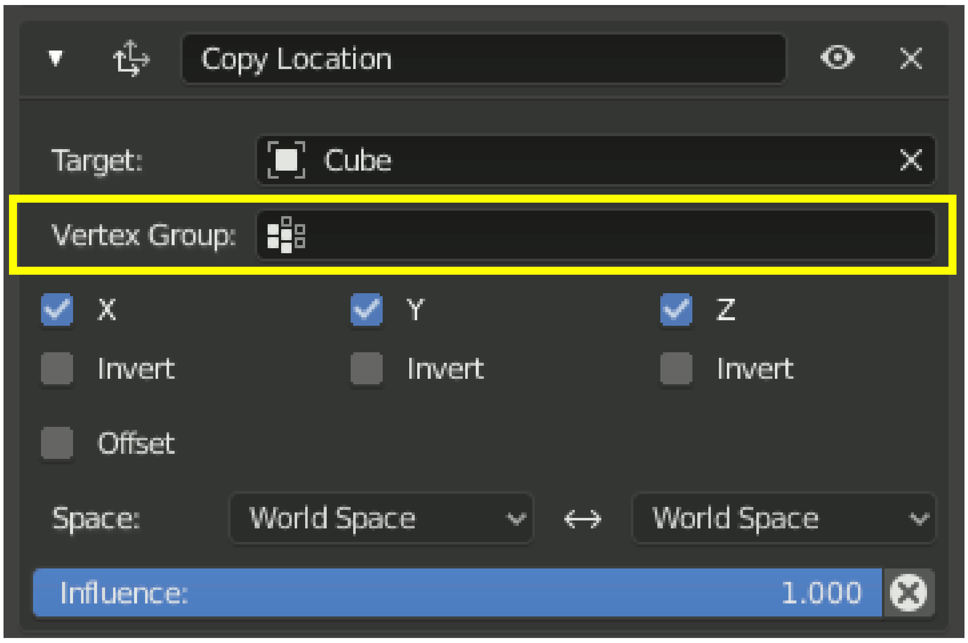


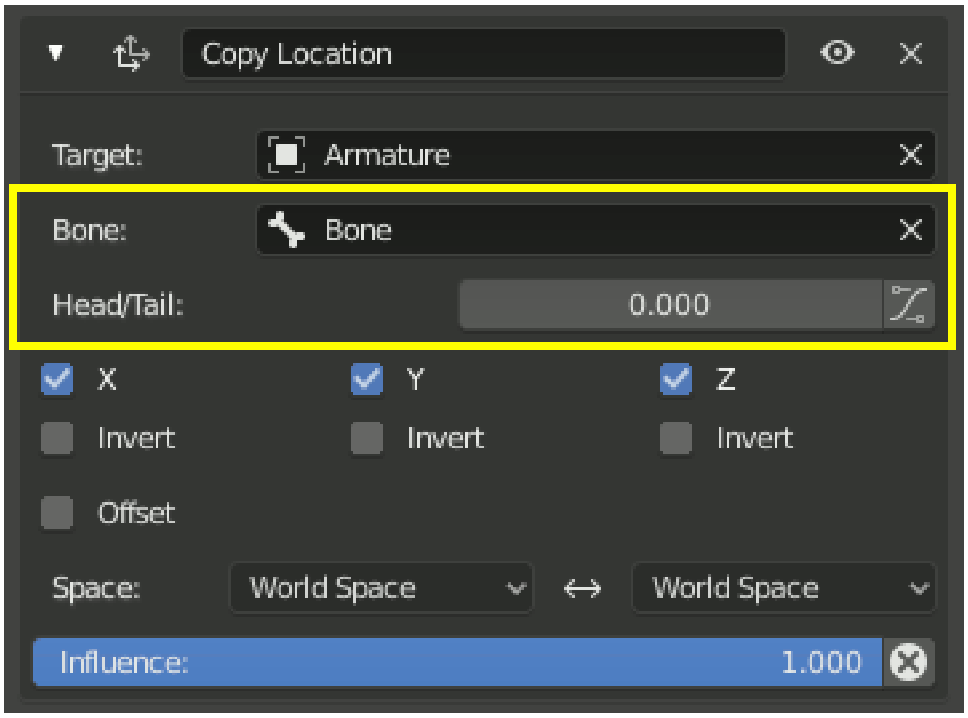
Fig. 1680: The Target field must be filled in for the constraint to function.

By default, the Target will use the *Object Origin* as the target point.

If the Target field links to a *Mesh* or *Lattice* object, a *Vertex Group* field will appear. Enter the name of a vertex group and the constraint will target the median point of this vertex group instead of the object's origin.



If the Target field links to an *Armature*, a *Bone* field will appear along with a *Head/Tail* slider. Enter the name of a bone and the constraint will target the bone instead of the entire armature object origin.



The slider moves the precise position of the target between the *Head* and *Tail* of the bone. Some constraints have a button next to the slider that enables using the curved shape of *Bendy Bones*.

Space

Constraints need a frame of reference in order to function. This frame of reference is called the "space" of the constraint. Choosing one space vs. another will change this frame of reference and substantially alter the behavior of a constraint.

To understand how changing the space will change the behavior of the constraint, consider experimenting with two empties. Make sure they display as arrows so that you can see the local axes for each empty. Make sure to size one empty a little larger than the other so that they are both always visible even if directly on top of each other. Then add a constraint to one empty that targets the other and experiment thoroughly by moving, rotating and scaling the target in many different ways.

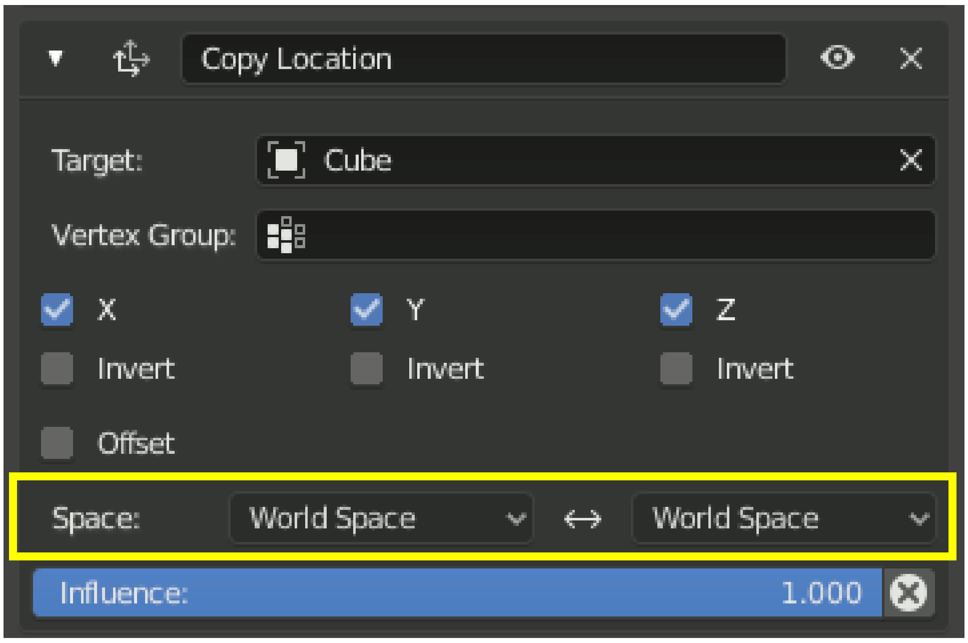

Copy Location
Target: Cube
Vertex Group:
X ✓ Y ✓ Z ✓
Invert Invert Invert
Offset
Space: World Space ↔ World Space
Influence: 1.000

Fig. 1681: This constraint is set to use World Space as the frame of reference for both its Target Space and its Owner Space.

Target Space & Owner Space

The space used to evaluate the target of the constraint is called the Target Space. The space used to evaluate the constrained object (the object that owns the constraint) is called the Owner Space. Hover over the space select menu(s) to learn whether it affects the space of the target or the space of the owner.

When the constraints use a Target and/or/nor an Owner space there will be no, one or two selector(s). The Copy Location constraint in example use both Target **and** Owner space.

When a constraint uses both Target and Owner space, the Target and Owner can be any combination of space types.

Space Types

World Space In this space type the world is the frame of reference for the object (or bone). Location is relative to the world origin. Rotation and Scale are oriented to the world axes. Transformations to the object, the object's parent and any other constraints higher up in the constraint stack are all taken into account.

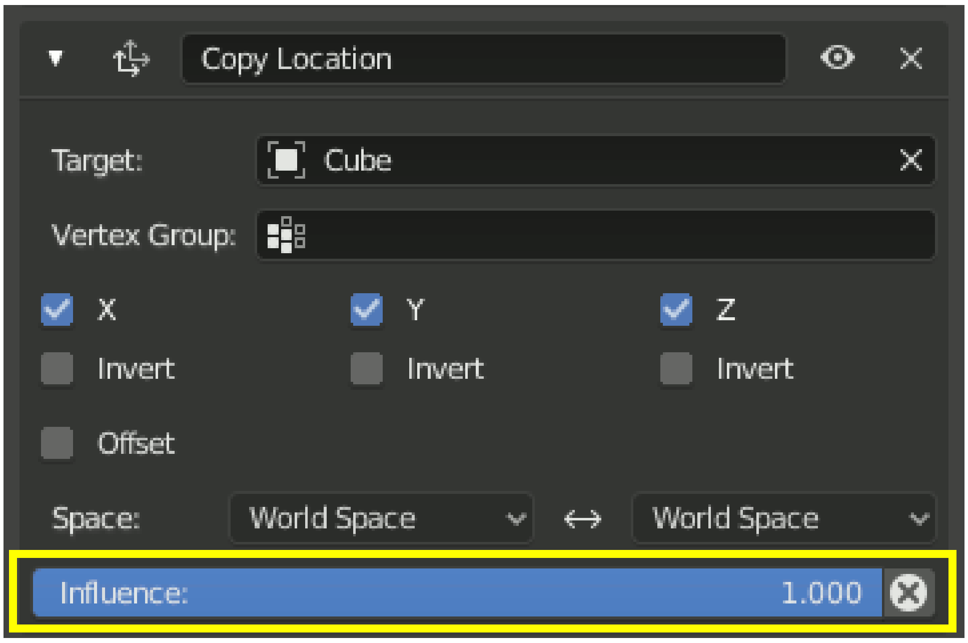
Local Space In this space type the parent of the object (or bone) is the frame of reference. Location is relative to the parent object origin. Rotation and Scale are oriented to the parent object axes. Only transformations to the object itself are taken into account. Transformations to the object's parent are **not** taken into account.

Local with Parent (bones only) The bone properties are evaluated relative to its rest pose location and orientation, thus including both its own transformations and those caused by a possible parent relationship (i.e. the chain's transformations above the bone).

Pose Space (bones only) The bone properties are evaluated in the armature object local space (i.e. independently from the armature transformations in *Object Mode*). Hence, if the armature object has null transformations, *Pose Space* will have the same effect as *World Space*.

Influence

The influence slider determines how much the constraint will affect the constrained object (target).



An influence of 0.0 will have no effect. An influence of 1.0 will have the full effect.

Values between (0.0 and 1.0) will have a partial effect, but be careful. These partial effects can be difficult to control, especially as the *constraint stack* grows in complexity.

The influence value is animatable, allowing constraints to be turned off, or partially on as needed.

The X button after the influence slider can be used to disable the constraint while trying to preserve the current object position. This may not work perfectly if other constraints remain active.

Stack

The combination of all the constraints affecting an object is called the Constraints Stack. The Stack is in the Constraints panel, below the Add Constraint menu.

Constraints in the stack are evaluated from top to bottom. The order of each constraint has a substantial impact on the final outcome of the stack. Changing the order of the constraints can change the behavior of the entire stack.

To change the order of a constraint use the up/down arrows in the *header*.

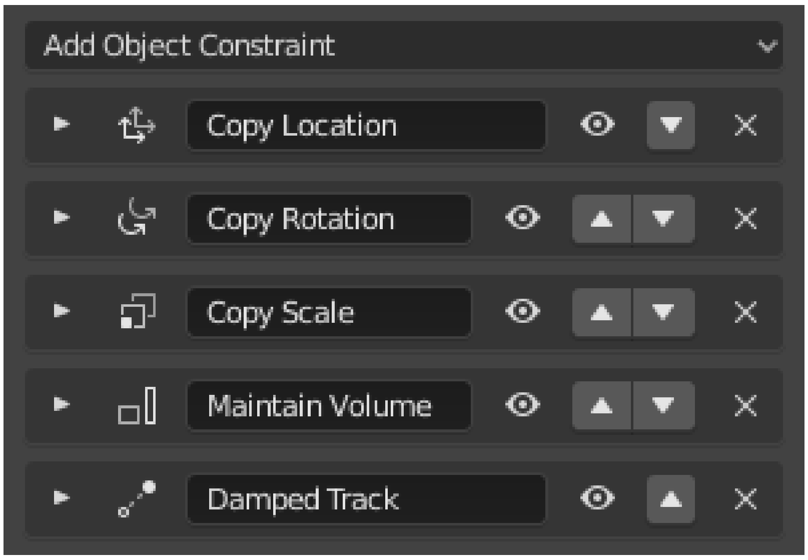


Fig. 1682: The constraints in this example stack are evaluated from top to bottom starting with the "Copy Location" constraint and ending with the final "Damped Track" constraint.

Motion Tracking

Camera Solver Constraint

The *Camera Solver* constraint gives the owner of this constraint, the location and rotation of the "solved camera motion".

The "solved camera motion" is where Blender reconstructs the position of the physical, real-world camera, when it filmed the video footage, relative to the thing being tracked.

Note: This constraint only works after you have set up a minimum of eight markers and pressed *Solve Camera Motion* (*Movie Clip Editor* → *Toolbar* → *Solve* → *Solve Camera Motion*).

Options

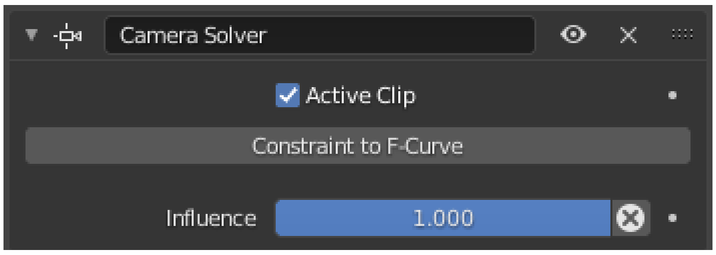


Fig. 1683: Camera Solver Constraint panel.

Active Clip Receive tracking data from the movie clip active in the Movie Clip editor. If unchecked, an option appears to choose from the other clips.

Constraint to F-Curve Applies the constraint, creating Keyframes for the transforms.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Object Solver Constraint

The *Object Solver* constraint gives the owner of this constraint, the location and rotation of the "solved object motion".

The "solved object motion" is where Blender thinks the physical, real-world (tracked) object was, relative to the camera that filmed it.

Can be used to add a mesh to video for example.

Note: This constraint only works after you have set up a minimum of eight markers and pressed *Solve object Motion*. Located at *Movie Clip Editor* → *Toolbar* → *Solve* → *Solve Camera Motion*.

If it says *Solve Camera Motion* instead of *Solve Object Motion* then go into the *Movie Clip Editor* → *Sidebar region* → *Objects* and switch it from the camera, to an object.

Options

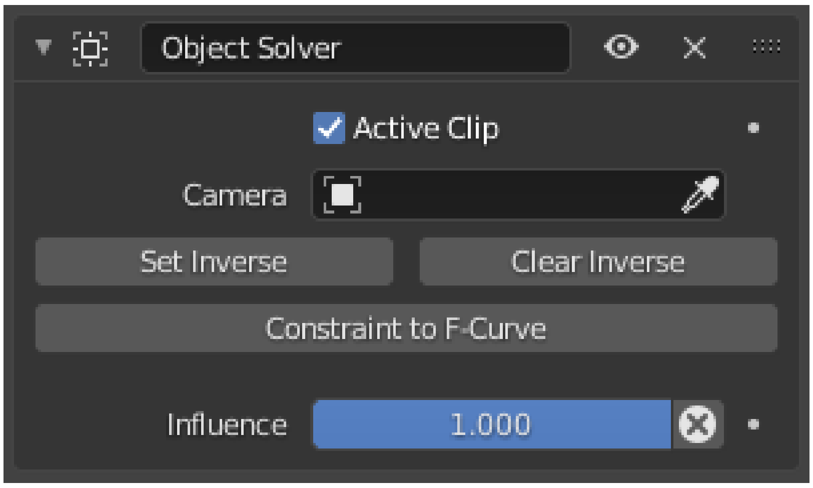


Fig. 1684: Object Solver Constraint panel.

Active Clip Receive tracking data from the active movie clip in the Movie Clip editor. If unchecked, an option appears to choose from the other clips.

Object Select a tracked object to receive transform data from.

Camera Select the camera to which the motion is parented to (if left empty the active scene camera is used).

Set Inverse Moves the origin of the object to the origin of the camera.

Clear Inverse Moves the origin of the object back to the spot set in the Movie Clip Editor *Toolbar* → *Solve* → *Orientation* → *Set Origin*.

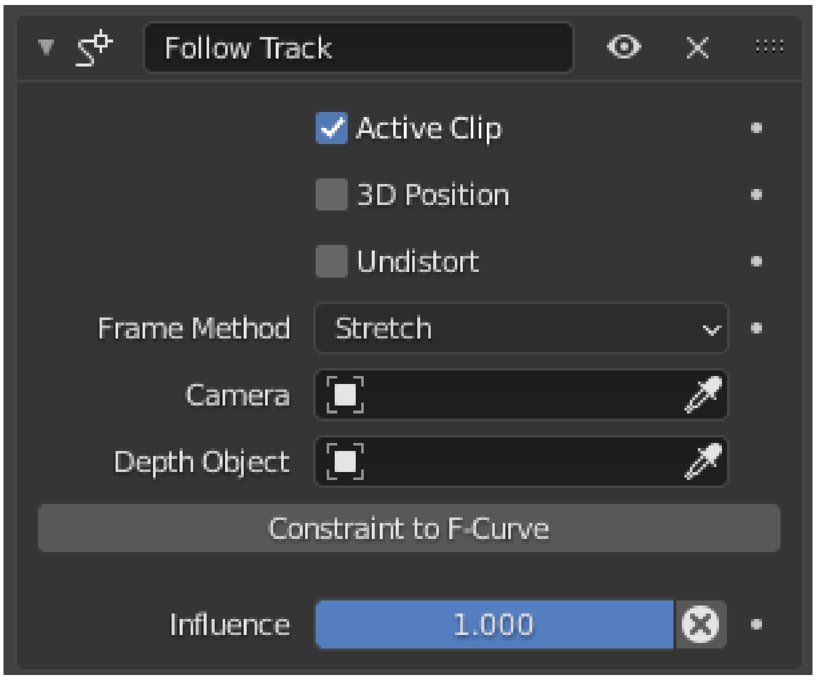
Constraint to F-Curve Applies the constraint, creating keyframes for the transforms.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Follow Track Constraint

By default the Follow Track constraint is making objects have the same position at a frame as the track has. The motion of this object happens on a single plane defined by the camera and the original position of the object.

Options



Follow Track

Active Clip: ✓

3D Position

Undistort

Frame Method: Stretch

Camera

Depth Object

Constraint to F-Curve

Influence: 1.000

Fig. 1685: Follow Track Constraint panel.

Active Clip Receive tracking data from the active movie clip in the Movie Clip editor. If unchecked, an option appears to choose from the other available clips.

3D Position Use the 3D position of the track to parent to.

Undistorted Parent to the undistorted position of the 2D track.

Frame Method Defines how the footage is fitted in the camera frame.

Camera Select the camera to which the motion is parented to (if active an empty scene camera is used).

Depth Object If this object is set, constrained objects will be projected onto the surface of this depth object which can be used to create facial makeup visual effects.

Constraint to F-Curve Creates F-curves for the object that copies the movement caused by the constraint.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://www.youtube.com/watch?v=KZalGrjGKSA

Transform

Copy Location Constraint

The *Copy Location* constraint forces its owner to have the same location as its target.

Important: Note that if you use such a constraint on a *connected* bone, it will have no effect, as it is the parent's tip which controls the position of your owner bone's root.

Options

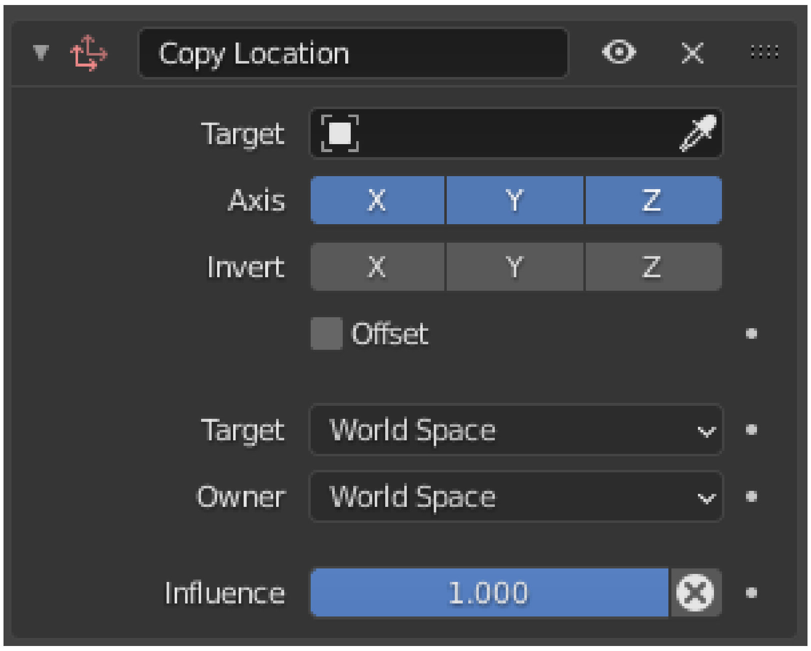


Fig. 1686: Copy Location panel.

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Axis These buttons control which axes are constrained.

Invert Invert their respective corresponding axis coordinates.

Offset When enabled, this control allows the owner to be moved (using its current transform properties), relative to its target's position.

Target/Owner Standard conversion between spaces. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Examples

A video can be found at https://vimeo.com/170198049

Animation

Let us animate the *Copy Location* constraint and its *Offset* button. For example, you can make your owner (let us call it "moon") describe perfect circles centered on the (0.0, 0.0, 0.0) point (using e.g. pydriven *LocX*/*LocY* animation curves, see *Drivers*), and then make it copy the location of a target (call it "earth", for example) with the *Offset* button enabled. Congratulation, you just modeled a satellite in a (simplified) orbit around its planet. Just do the same thing with its planet around its star (which you might call "sun", what do you think?), and why not, for the star around its galaxy.

Here is a small animation of a "solar" system created using (among a few others) the technique described above:

A video can be found at https://vimeo.com/15187945

Note that, this "solar" system is not realistic at all (the wrong scale, the "earth" is rotating in the wrong direction around the "sun", ...).

You can download the blend-file used to create this animation.

Furthermore you can also animate a few properties of each constraint using animation curves: e.g. you can animate the *Influence* of a constraint. It is used to first stick the camera to the "moon", then to the "earth", and finally to nothing, using two *Copy Location* constraints with *Offset* set.

Copy Rotation Constraint

The *Copy Rotation* constraint forces its owner to match the rotation of its target.

Options

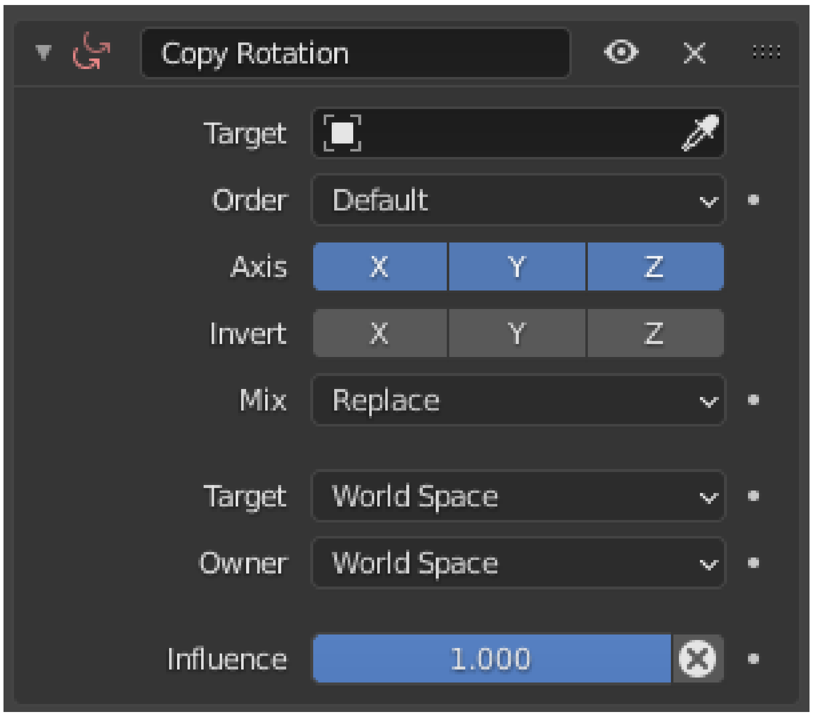

Copy Rotation
Target
Order: Default
Axis: X Y Z
Invert: X Y Z
Mix: Replace
Target: World Space
Owner: World Space
Influence: 1.000

Fig. 1687: Copy Rotation panel.

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Order Allows specifying which *Euler* order to use during the copy operation. Defaults to the order of the owner.

Axis These buttons control which axes are constrained.

Invert Invert their respective corresponding axis coordinates.

Mix Specifies how the new rotation is combined with the existing rotation.

> **Replace** The new axis values replace existing values.
>
> **Add** The new axis values are added to the existing values.
>
> **Before Original** The new rotation is added before the existing rotation, as if it was applied to a parent of the constraint owner.
>
> **After Original** The new rotation is added after the existing rotation, as if it was applied to a child of the constraint owner.
>
> **Offset (Legacy)** This replicates the behavior of the original Offset checkbox. It was intended to be similar to the *Before Original* behavior, but does not work correctly with multiple axis rotations, and is thus deprecated.

Target/Owner Standard conversion between spaces. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171073854

Copy Scale Constraint

The *Copy Scale* constraint forces its owner to have the same scale as its target.

Note: Here we talk of *scale*, not of *size*! Indeed, you can have two objects, one much bigger than the other, and yet both of them have the same scale. This is also true with bones: in *Pose Mode*, they all have a unitary scale when they are in rest position, represented by their visible length.

Options

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Axis These buttons control which axes of the target scale are copied.

Power Allows raising the copied scale to the specified arbitrary power.

Make Uniform Instead of copying scale for individual axes, apply a uniform scaling factor to all axes of the owner that achieves the same overall change in volume.

Offset When enabled, the constraint combines the copied scale with the owner's scale, instead of overwriting it.

Additive Uses addition instead of multiplication in the implementation of the *Offset* option.

Target/Owner Standard conversion between spaces. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Note: Since scale is a multiplicative quantity, it should be combined using multiplication, and split into fractions or inverted via power. Thus the use of *Power* is more mathematically correct than *Influence*, which uses linear interpolation. The use of the *Additive* option is also not recommended.

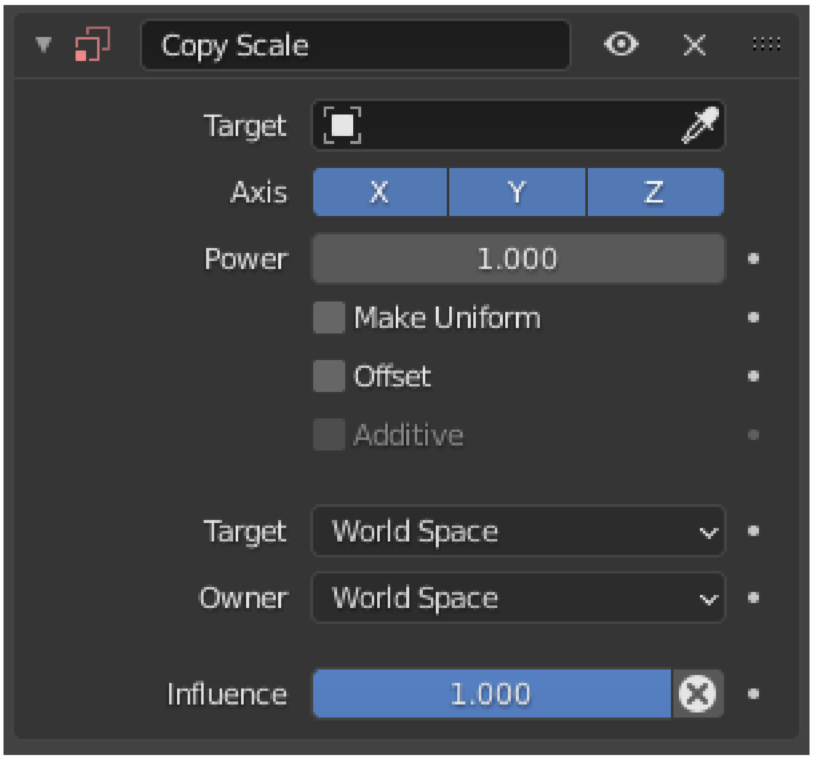


Fig. 1688: Copy Scale panel.

Tip: To copy scale from one axis of the target to all axes of the owner, disable other axes, enable *Make Uniform*, and set *Power* to 3.

Example

A video can be found at https://vimeo.com/171077617

Copy Transforms Constraint

The *Copy Transforms* constraint forces its owner to have the same transforms as its target.

Options

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Mix Specifies how the copied transformation is combined with the existing transformation.

Replace The new transformation replaces the existing transformation.

Before Original The new transformation is added before the existing transformation, as if it was applied to an imaginary parent of the constraint owner. Scale is handled like in the *Aligned Inherit Scale* mode of bones to avoid creating shear.

After Original The new transformation is added after the existing transformation, as if it was applied locally to an imaginary child of the constraint owner. Scale is handled like in the *Aligned Inherit Scale* mode of bones to avoid creating shear.

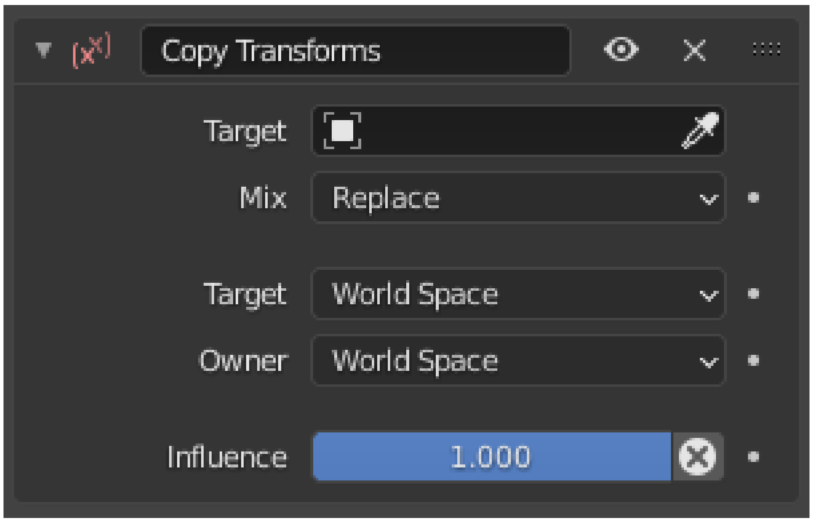


Fig. 1689: Copy Transforms panel.

Target/Owner Standard conversion between spaces. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171108888

Limit Distance Constraint

The *Limit Distance* constraint forces its owner to stay either further from, nearer to, or exactly at a given distance from its target. In other words, the owner's location is constrained either outside, inside, or at the surface of a sphere centered on its target.

When you specify a (new) target, the *Distance* value is automatically set to correspond to the distance between the owner and this target.

Important: Note that if you use such a constraint on a *connected* bone, it will have no effect, as it is the parent's tip which controls the position of your owner bone's root.

Options

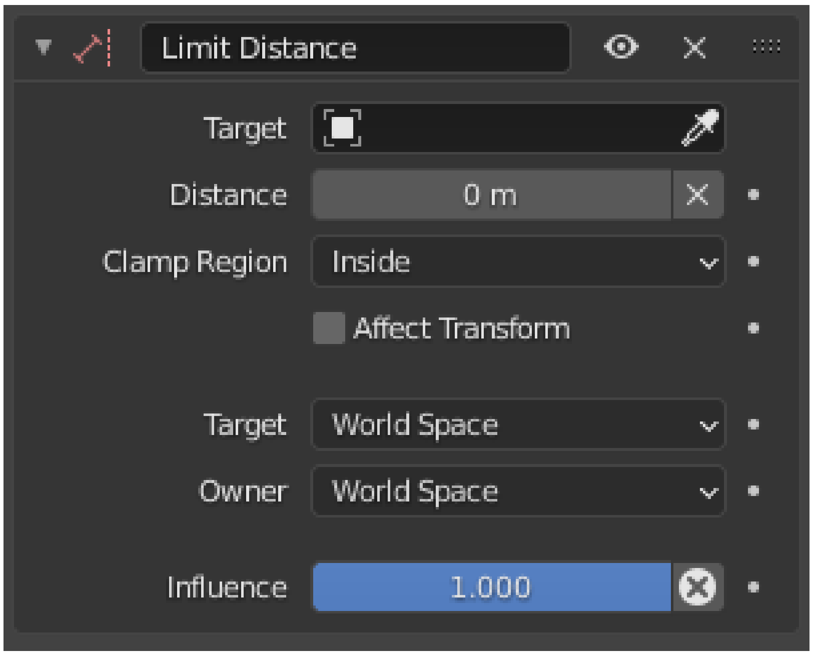
Target *Data ID* used to select the constraint's target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Distance This number field sets the limit distance, i.e. the radius of the constraining sphere.

Reset Distance X When clicked, this small button will reset the *Distance* value, so that it corresponds to the actual distance between the owner and its target (i.e. the distance before this constraint is applied).

Clamp Region The *Limit Mode* select menu allows you to choose how to use the sphere defined by the *Distance* setting and target's origin:

Inside The owner is constrained *inside* the sphere.



Limit Distance

Target

Distance 0 m

Clamp Region Inside

Affect Transform

Target World Space

Owner World Space

Influence 1.000

Fig. 1690: Limit Distance panel.

Outside The owner is constrained *outside* the sphere.

Surface The owner is constrained *on the surface* of the sphere.

Affect Transform Transform operators will take the constraint into account to immediately restrict the resulting transform property values.

Target/Owner Standard conversion between spaces. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171109014

Limit Location Constraint

An object or *unconnected* bone can be moved around the scene along the X, Y and Z axes. This constraint restricts the amount of allowed translations along each axis, through lower and upper bounds.

The limits for an object are calculated from its origin, and the limits of a bone, from its root.

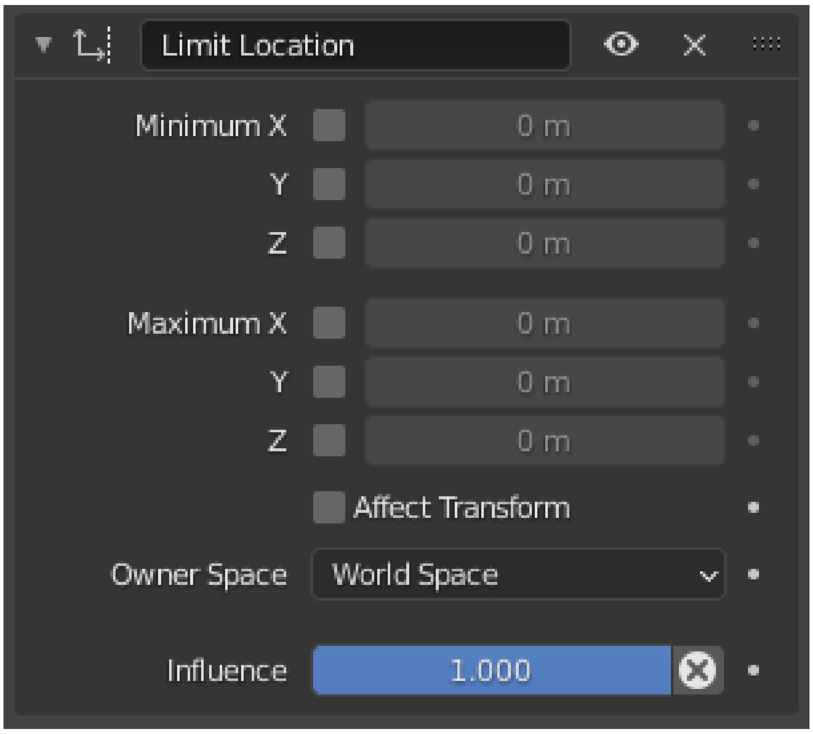
It is interesting to note that even though the constraint limits the visual and rendered location of its owner, its owner's data-block still allows (by default) the object or bone to have coordinates outside the minimum and maximum ranges. This can be seen in its *Transform* panel.

When an owner is selected and attempted to be moved outside the limit boundaries, it will be constrained to those boundaries visually and when rendered, but internally, its coordinates will still be changed beyond the limits. If the constraint is removed, its ex-owner will seem to jump to its internally specified location.

Similarly, if its owner has an internal location that is beyond the limits, dragging it back into the limit area will appear to do nothing until the internal coordinates are back within the limit threshold (unless you enabled the *For Transform* option, see below).

Setting equal the min and max values of an axis, locks the owner's movement along that axis... Although this is possible, using the *Transformation Properties* axis locking feature is probably easier!

Options


Limit Location

Minimum X 0 m
Y 0 m
Z 0 m

Maximum X 0 m
Y 0 m
Z 0 m

Affect Transform

Owner Space World Space

Influence 1.000

Fig. 1691: Limit Location panel.

Minimum X, Y, Z These buttons enable the lower boundary for the location of the owner's origin along, respectively, the X, Y and Z axes of the chosen *Space*. The number field below them controls the value of their limit. Note that if a min value is higher than its corresponding max value, the constraint behaves as if it had the same value as the max one.

Maximum X, Y, Z These buttons enable the upper boundary for the location of the owner's origin along, respectively, the X, Y and Z axes of the chosen *Space*. Same options as above.

Affect Transform As pointed out before by default, even though visually constrained, the owner can still have coordinates out of bounds (as shown by the *Transform* panel). Well, when you enable this checkbox, this is no longer possible – the owner's transform properties are also limited by the constraint. However note that, the constraint does not directly modify the coordinates: you have to select its owner one way or another for this to take effect...

Owner Space This constraint allows you to choose in which space to evaluate its owner's transform properties. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171115770

Limit Rotation Constraint

An object or bone can be rotated around the X, Y and Z axes. This constraint restricts the amount of allowed rotations around each axis, through lower and upper bounds.

It is interesting to note that even though the constraint limits the visual and rendered rotations of its owner, its owner's data-block still allows (by default) the object or bone to have rotation values outside the minimum and maximum ranges. This can be seen in the *Transform* panel. When an owner is rotated and attempted to be rotated outside the limit boundaries, it will be constrained to those boundaries visually and when rendered, but internally, its rotation values will still be changed beyond the limits. If the constraint is removed, its ex-owner will seem to jump to its internally specified rotation.

Similarly, if its owner has an internal rotation that is beyond the limit, rotating it back into the limit area will appear to do nothing until the internal rotation values are back within the limit threshold (unless you enabled the *For Transform* option, see below).

Setting equal the min and max values of an axis, locks the owner's rotation around that axis... Although this is possible, using the *Transformation Properties* axis locking feature is probably easier.

This transform does not constrain the bone if it is manipulated by the IK solver. For constraining the rotation of a bone for IK purposes, see the "Inverse Kinematics" section of Bone properties.

Options

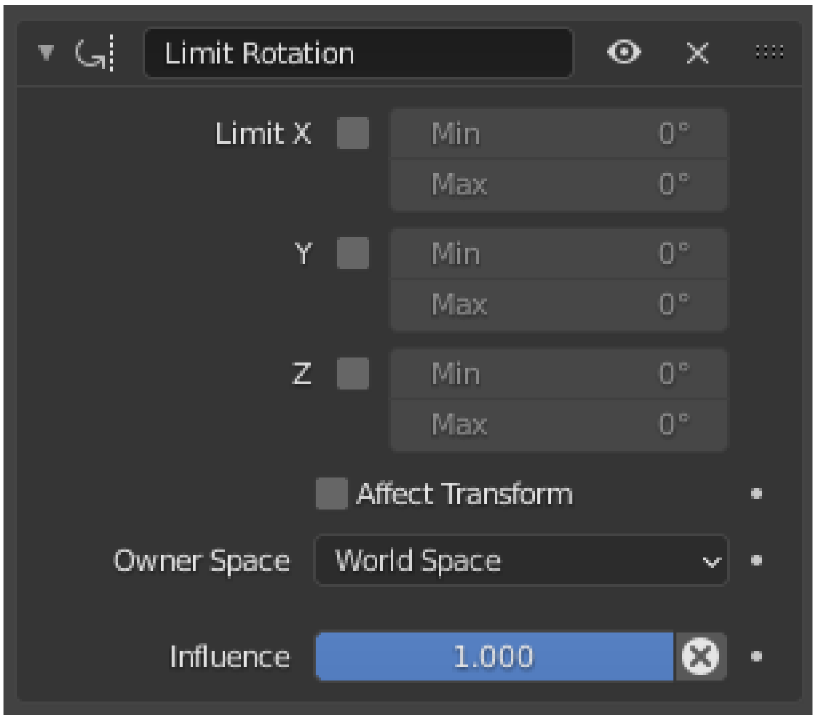

Limit Rotation
Limit X ☐ Min 0° Max 0°
Y ☐ Min 0° Max 0°
Z ☐ Min 0° Max 0°
☐ Affect Transform
Owner Space: World Space
Influence: 1.000

Fig. 1692: Limit Rotation panel.

Limit X, Y, Z These buttons enable the rotation limit around respectively the X, Y and Z axes of the owner, in the chosen *Owner Space*. The *Min* and *Max* number fields to their right control the value of their lower and upper boundaries, respectively.

Note:

- If a min value is higher than its corresponding max value, the constraint behaves as if it had the same value as the max one.

- Unlike the *Limit Location constraint*, you cannot enable separately lower or upper limits…

Affect Transform As pointed out before by default, even though visually constrained, the owner can still have rotations out of bounds (as shown by the *Transform* panel). When you enable this checkbox, this is no more possible – the owner transform properties are also limited by the constraint. However note that, the constraint does not directly modifies the rotation values: you have to rotate one way or the other its owner, for this to take effect…

Owner Space This constraint allows you to choose in which space evaluate its owner's transform properties. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171115852

Limit Scale Constraint

An object or bone can be scaled along the X, Y and Z axes. This constraint restricts the amount of allowed scaling along each axis, through lower and upper bounds.

Important: This constraint does not tolerate negative scale values (those you might use to mirror an object…): when you add it to an object or bone, even if no axis limit is enabled, nor the *For Transform* button, as soon as you scale your object, all negative scale values are instantaneously inverted to positive ones… And the boundary settings can only take strictly positive values.

It is interesting to note that even though the constraint limits the visual and rendered scale of its owner, its owner's data-block still allows (by default) the object or bone to have scale values outside the minimum and maximum ranges (as long as they remain positive!). This can be seen in its *Transform* panel. When an owner is scaled and attempted to be moved outside the limit boundaries, it will be constrained to those boundaries visually and when rendered, but internally, its coordinates will still be changed beyond the limits. If the constraint is removed, its ex-owner will seem to jump to its internally-specified scale.

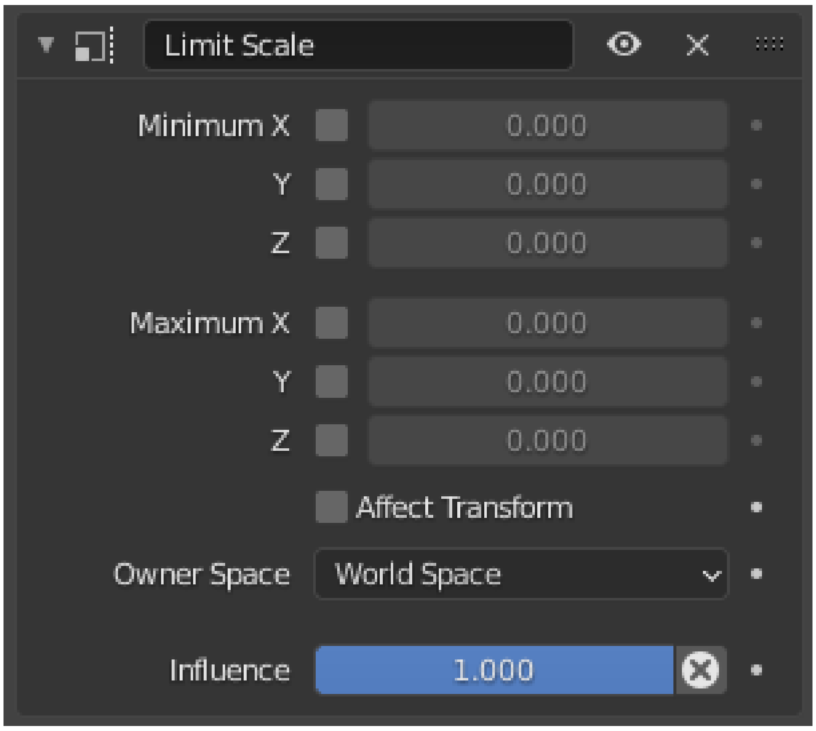
Similarly, if its owner has an internal scale that is beyond the limits, scaling it back into the limit area will appear to do nothing until the internal scale values are back within the limit threshold (unless you enabled the *For Transform* option, see below, or your owner has some negative scale values).

Setting equal the min and max values of an axis locks the owner's scaling along that axis. Although this is possible, using the *Transformation Properties* axis locking feature is probably easier.

Options

Minimum/Maximum X, Y, Z These buttons enable the lower boundary for the scale of the owner along respectively the X, Y and Z axes of the chosen *Space*. The *Min* and *Max* number fields to their right control the value of their lower and upper boundaries, respectively.

Note: If a min value is higher than its corresponding max value, the constraint behaves as if it had the same value as the max one.


Limit Scale

Minimum X 0.000
Y 0.000
Z 0.000

Maximum X 0.000
Y 0.000
Z 0.000

Affect Transform

Owner Space World Space

Influence 1.000

Fig. 1693: Limit Scale panel.

Affect Transform As pointed out before by default, even though visually constrained, and except for the negative values, the owner can still have scales out of bounds (as shown by the *Transform* panel). When you enable this checkbox, this is no longer possible, the owner transform properties are also limited by the constraint. However note that, the constraint does not directly modify the scale values: you have to scale its owner one way or another for this to take effect.

Owner Space This constraint allows you to choose in which space to evaluate its owner's transform properties. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171275278

Maintain Volume Constraint

The *Maintain Volume* constraint limits the volume of a mesh or a bone to a given ratio of its original volume.

See also:

Harkyman on the development of the Maintain Volume constraint.

Options

Mode Specifies how the constraint handles scaling of the non-free axes.

Strict This mode overrides non-free axis scaling to strictly maintain the specified volume. Only the ratio between the scale of the non-free axes is passed through.

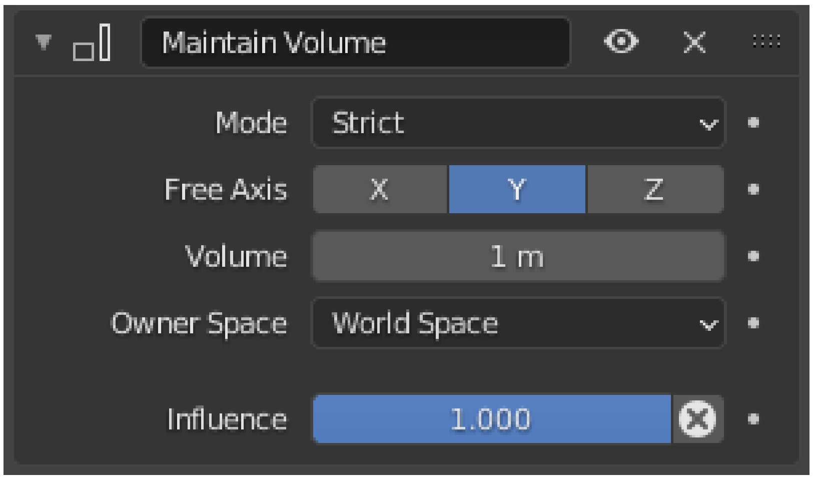


Fig. 1694: Maintain Volume Constraint.

Uniform This mode maintains the volume as specified only when the pre-constraint scaling is uniform. Deviations from uniform scaling on non-free axes are passed through.

Single Axis This mode maintains the volume only when the object is scaled just on its free axis. Any additional non-free axis scaling is passed through.

Free Axis The free-scaling axis of the object.

Volume The bone's rest volume.

Owner Space This constraint allows you to choose in which space to evaluate its owner's transform properties. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171275315

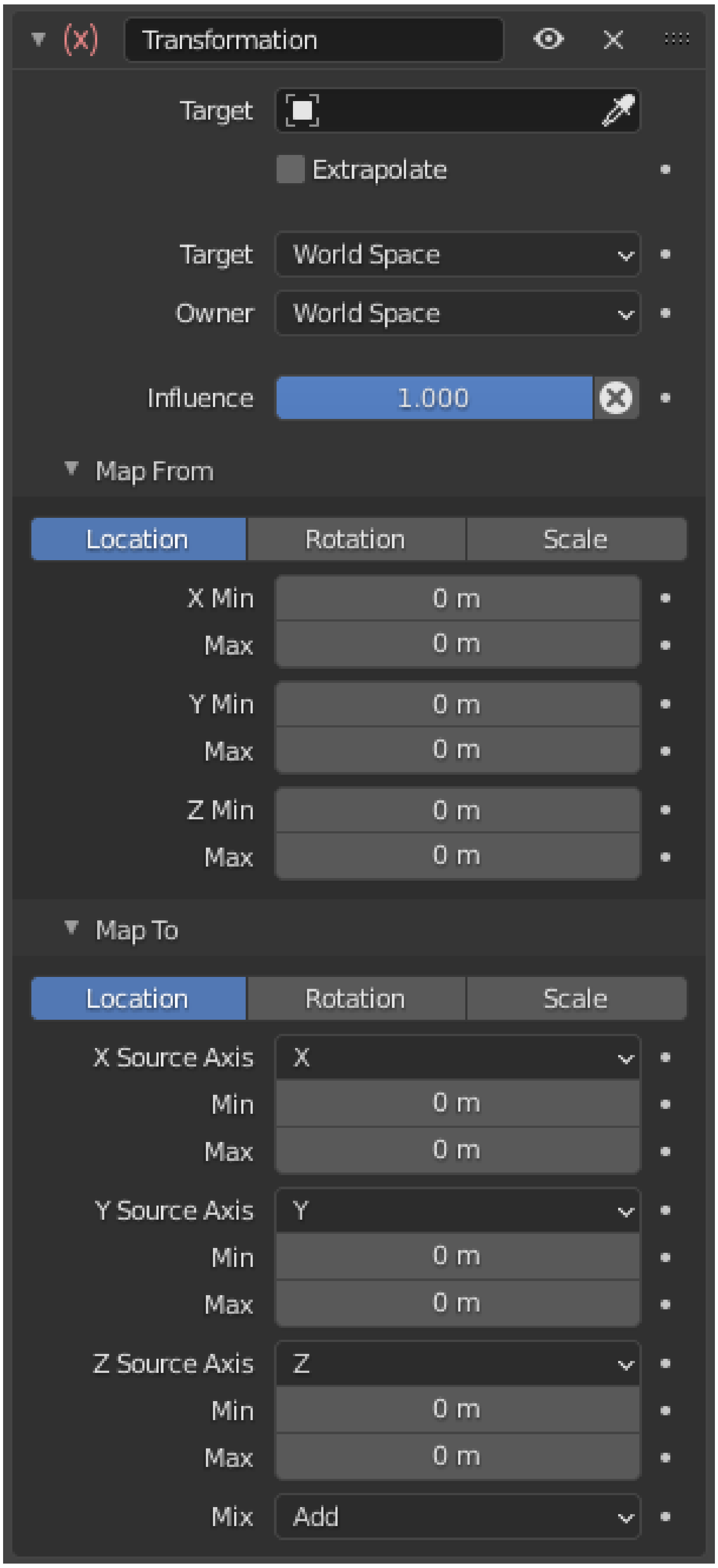
Transformation Constraint

This constraint is more complex and versatile than the other "transform" constraints. It allows you to map one type of transform properties (i.e. location, rotation or scale) of the target, to the same or another type of transform properties of the owner, within a given range of values (which might be different for each target and owner property). You can also switch between axes, and use the range values not as limits, but rather as "markers" to define a mapping between input (target) and output (owner) values.

So, e.g. you can use the position of the target along the X axis to control the rotation of the owner around the Z axis, stating that 1 unit along the target X axis corresponds to 10 units around the owner Z axis. Typical uses for this include gears (see note below), and rotation based on location setups.

Options

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.


Transformation

Target

Extrapolate

Target: World Space

Owner: World Space

Influence: 1.000

Map From

Location | Rotation | Scale

X Min: 0 m
Max: 0 m
Y Min: 0 m
Max: 0 m
Z Min: 0 m
Max: 0 m

Map To

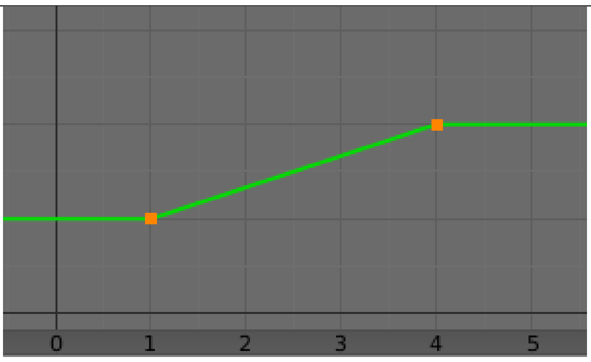
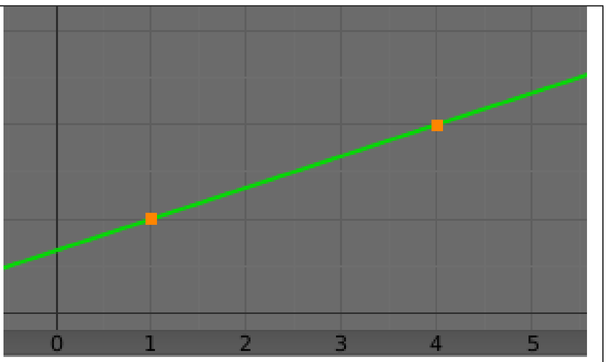
Location | Rotation | Scale

X Source Axis: X
Min: 0 m
Max: 0 m
Y Source Axis: Y
Min: 0 m
Max: 0 m
Z Source Axis: Z
Min: 0 m
Max: 0 m
Mix: Add

Fig. 1695: Transformation panel.

Extrapolate By default, the *Min* and *Max* values bound the input and output values; all values outside these ranges are clipped to them. When you enable this button, the *Min* and *Max* values are no longer strict limits, but rather "markers" defining a proportional (linear) mapping between input and corresponding output values. Let us illustrate that with two graphs Fig. *The Extrapolate principles.*. In these pictures, the input range (in abscissa) is set to (1.0 to 4.0), and its corresponding output range (in ordinate), to (1.0 to 2.0). The yellow curve represents the mapping between input and output.

Table 120: The Extrapolate principles.

 Fig. 1696: Extrapolate disabled: the output values are bounded inside the (1.0 to 2.0) range.	 Fig. 1697: Extrapolate enabled: the output values are "free" to proportionally follow the input ones.

Target/Owner Standard conversion between spaces. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Map From

It contains the input (from target) settings.

Location, Rotation, Scale The radio buttons allow you to select which type of property to use.

Mode (Rotation) Allows specifying the type of rotation input to use, including different *Euler* orders, *Quaternion*, and other *Rotation Channel Modes*. Defaults to using the *Euler* order of the constraint owner.

In the *Quaternion* mode the channels are converted to weighted angles in the same way as the swing angles of the *Swing and X/Y/Z Twist* modes.

X/Y/Z Min, Max Independently for each axis (X, Y, and Z) the min and max number fields control the lower and upper bounds of the input value range. Note that if a min value is higher than its corresponding max value, the constraint behaves as if it had the same value as the max one.

Map To

It contains the output (to owner) settings.

Location, Rotation, Scale The three radio buttons allow you to select which type of property to control.

Order (Rotation) For rotation, allows specifying which *Euler* order to use during evaluation of the constraint. Defaults to using the order of the constraint owner.

X/Y/Z Source Axis The three axis selectors allow you to select which input axis to map to, respectively (from top to bottom), the X, Y and Z output (owner) axes.

Min, Max The *Min* and *Max* number fields control the lower and upper bounds of the output value range, independently for each mapped axis. Note that if a min value is higher than its corresponding max value, the constraint behaves as if it had the same value as the max one.

Mix Specifies how the result of the constraint is combined with the existing transformation. The set of available choices varies based on the type of transformation.

Replace The result of the constraint replaces the existing transformation.

Multiply (Scale) The new values are multiplied with the existing axis values.

Add (Location, Rotation) The new values are added to the existing axis values.

Before Original (Rotation) The new rotation is added before the existing rotation, as if it was applied to a parent of the constraint owner.

After Original (Rotation) The new rotation is added after the existing rotation, as if it was applied to a child of the constraint owner.

Note:

- For historical reasons, the *Mix* mode defaults to *Add* for location and rotation, and *Replace* for scale.
- When using the rotation transform properties of the target as input, whatever the real values are, the constraint will always "take them back" into the (-180 to 180) range. E.g. if the target has a rotation of 420 degrees around its X axis, the values used as *X* input by the constraint will be:

  $((420 + 180) modulo 360) - 180 = 60 - ...$

  This is why this constraint is not really suited for gears!
- Similarly, when using the scale transform properties of the target as input, whatever the real values are, the constraint will always take their absolute values (i.e. invert negative ones).
- When a *min* value is higher than its corresponding *max* one, both are considered equal to the *max* one. This implies you cannot create "reversed" mappings…

Example

A video can be found at https://vimeo.com/171275353

Transform Cache Constraint

The *Transform Cache Constraint* is used to stream animations from *Alembic* made at the transformation matrix level (for example rigid bodies, or camera movements).

When *importing an Alembic file*, Transform Cache constraints are automatically added to objects with animated transforms. For time-varying meshes (so deforming animations), the *Mesh Sequence Cache modifier* is used.

Options

Cache File Data-block menu to select the Alembic file.

File Path Path to Alembic file.

Sequence Whether or not the cache is separated in a series of files.

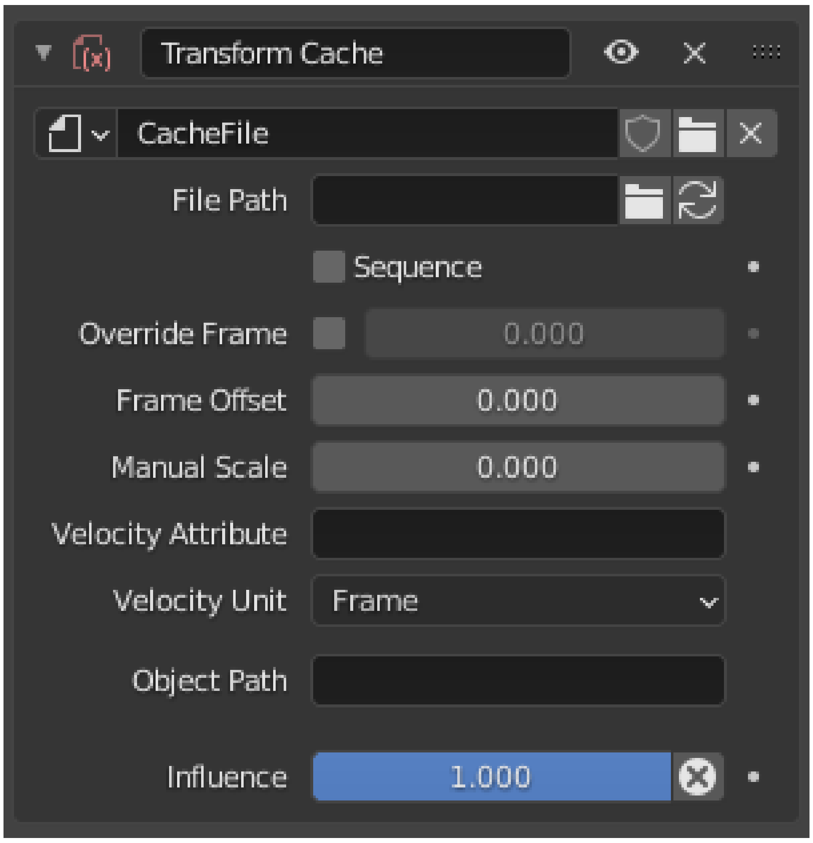

Transform Cache
CacheFile
File Path
Sequence
Override Frame 0.000
Frame Offset 0.000
Manual Scale 0.000
Velocity Attribute
Velocity Unit Frame
Object Path
Influence 1.000

Fig. 1698: Transform Cache Constraint.

Override Frame Whether to use a custom frame for looking up data in the cache file, instead of using the current scene frame.

The *Frame* value is the time to use for looking up the data in the cache file, or to determine which to use in a file sequence.

Frame Offset Subtracted from the current frame to use for looking up the data in the cache file, or to determine which file to use in a file sequence.

Manual Scale Value by which to enlarge or shrink the object with respect to the world's origin.

Velocity Attribute The name of the Alembic attribute used for generating motion blur data; by default, this is `.velocities` which is standard for most Alembic files.

Velocity Unit Defines how the velocity vectors are interpreted with regard to time.

Frame The velocity unit was encoded in frames and does not need to be scaled by scene FPS.

Second The velocity unit was encoded in seconds and needs to be scaled by the scene FPS (1 / FPS).

Object Path The path to the Alembic object inside the archive.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Tracking

Clamp To Constraint

The *Clamp To* constraint clamps an object to a curve. The *Clamp To* constraint is very similar to the *Follow Path* constraint, but instead of using the evaluation time of the target curve, *Clamp To* will get the actual location properties of its owner (those shown in the *Transform* panel), and judge where to put it by "mapping" this location along the target curve.

One benefit is that when you are working with *Clamp To*, it is easier to see what your owner will be doing; since you are working in the 3D Viewport, it will just be a lot more precise than sliding keys around on an F-curve and playing the animation over and over.

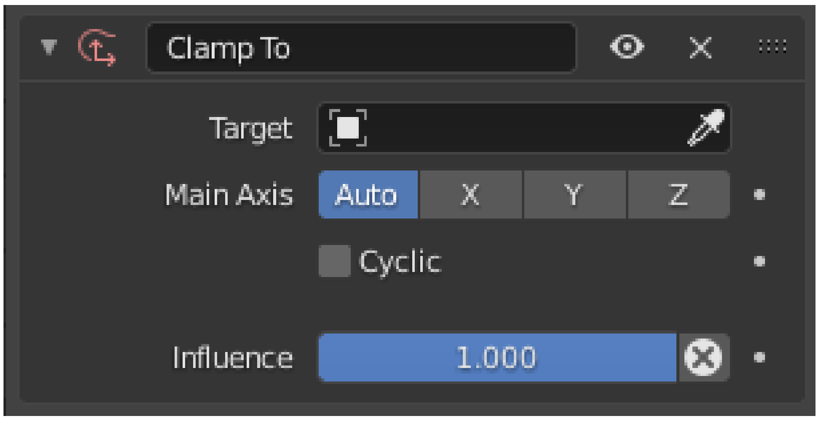
A downside is that unlike in the *Follow Path constraint*, *Clamp To* does not have any option to track your owner's rotation (pitch, roll, yaw) to the banking of the targeted curve, but you do not always need rotation on, so in cases like this it's usually a lot handier to fire up a *Clamp To*, and get the bits of rotation you do need some other way.

The mapping from the object's original position to its position on the curve is not perfect, but uses the following simplified algorithm:

- A "main axis" is chosen, either by the user, or as the longest axis of the curve's bounding box (the default).
- The position of the object is compared to the bounding box of the curve in the direction of the main axis. So for example if X is the main axis, and the object is aligned with the curve bounding box's left side, the result is 0; if it is aligned with the right side, the result is 1.
- If the cyclic option is unchecked, this value is clamped in the range 0-1.
- This number is used as the curve time, to find the final position along the curve that the object is clamped to.

This algorithm does not produce exactly the desired result because curve time does not map exactly to the main axis position. For example an object directly in the center of a curve will be clamped to a curve time of 0.5 regardless of the shape of the curve, because it is halfway along the curve's bounding box. However, the 0.5 curve time position can actually be anywhere within the bounding box!

Options


Clamp To

Target

Main Axis: Auto | X | Y | Z

Cyclic

Influence: 1.000

Fig. 1699: Clamp To panel.

Target The target *Data ID* indicates which curve object the Clamp To constraint will track along. The target must be a curve object type. See *common constraint properties* for more information.

Main Axis This button group controls which global axis (X, Y or Z) is the main direction of the path. When clamping the object to the target curve, it will not be moved significantly on this axis. It may move a small amount on that axis because of the inexact way this constraint functions.

For example if you are animating a rocket launch, it will be the Z axis because the main direction of the launch path is up. The default *Auto* option chooses the axis which the curve is longest in (or X if they are equal). This is usually the best option.

Cyclic By default, once the object has reached one end of its target curve, it will be constrained there. When the *Cyclic* option is enabled, as soon as it reaches one end of the curve, it is instantaneously moved to its other end. This is of course primarily designed for closed curves (e.g. circles), as this allows your owner to go around it over and over.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171276763

Damped Track Constraint

The *Damped Track* constraint constrains one local axis of the owner to always point towards *Target*. This constraint uses a pure *Swing* rotation, i.e. the shortest possible single axis rotation. In other 3D software you can find it with the name “Look at” constraint.

Although usually associated with bones, Damped Track can align objects to point to (and follow) other objects or bones. It is important to note that the constraint aligns the origin’s axes to point to the target’s origin point. This is illustrated in the following figure. In each case the objects are set as Damped Track to +X.

Options

Target *Data ID* used to select the constraint’s target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Track Axis Once the owner object has had a Damped Track constraint applied to it, you must then choose which axis of the object you want to point at the Target object. The negative axis direction cause the object to point away from the Target object along the selected axis direction.

-X, -Y, -Z, X, Y, Z

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171278084

Inverse Kinematics Constraint

The *Inverse Kinematics* constraint implements the *inverse kinematics* armature posing technique. Hence, it is only available for bones. To quickly create an IK constraint with a target, select a bone in Pose Mode, and press `Shift-I`.

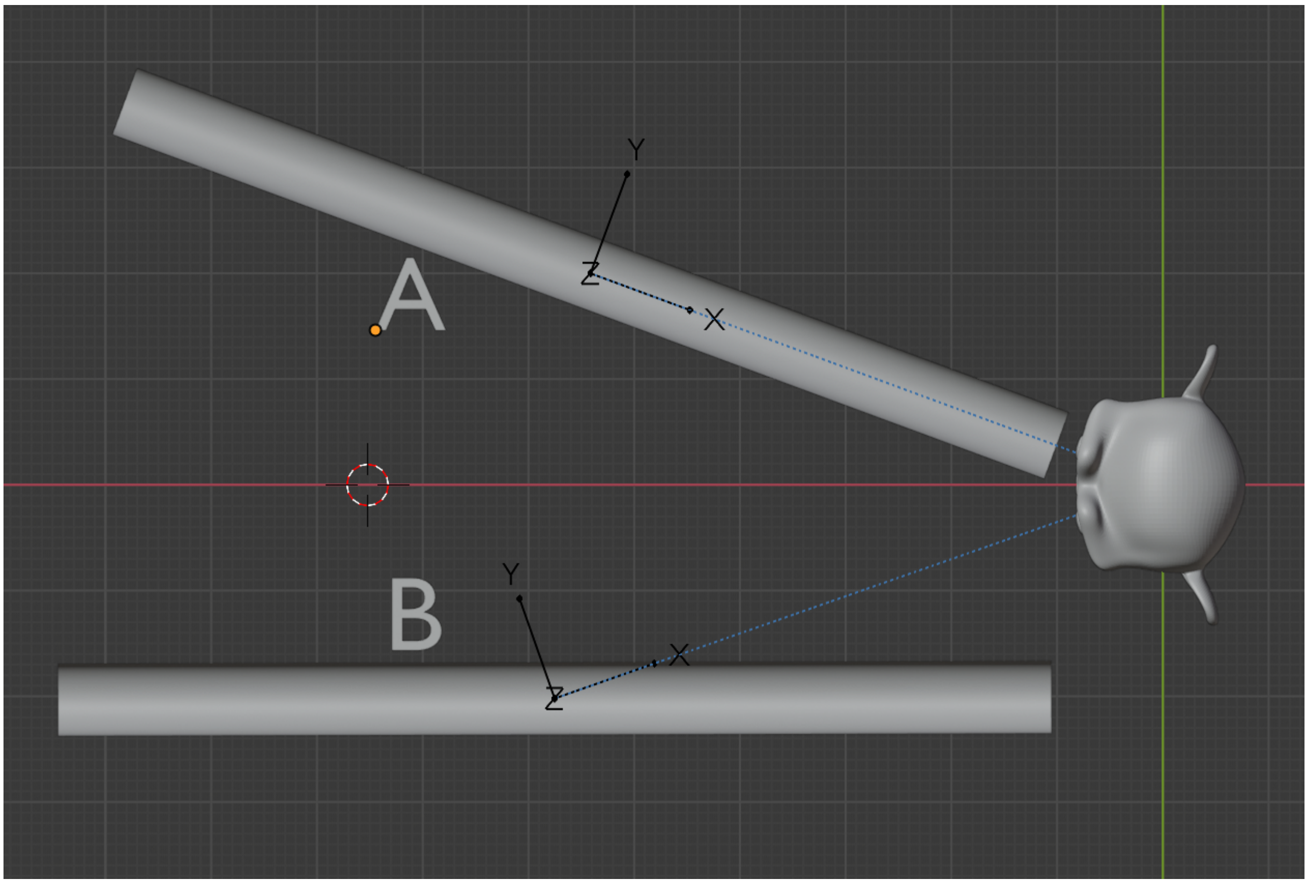
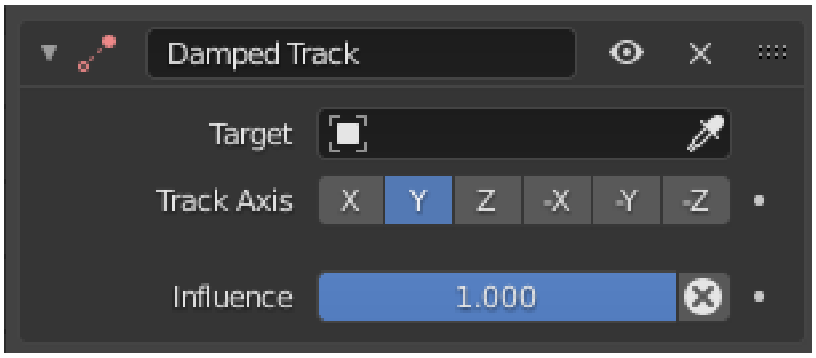


Fig. 1700: A: Object vertices aligned along axis origin. B: Object vertices aligned away from origin.



Target

Track Axis: X [✓ Y] Z -X -Y -Z

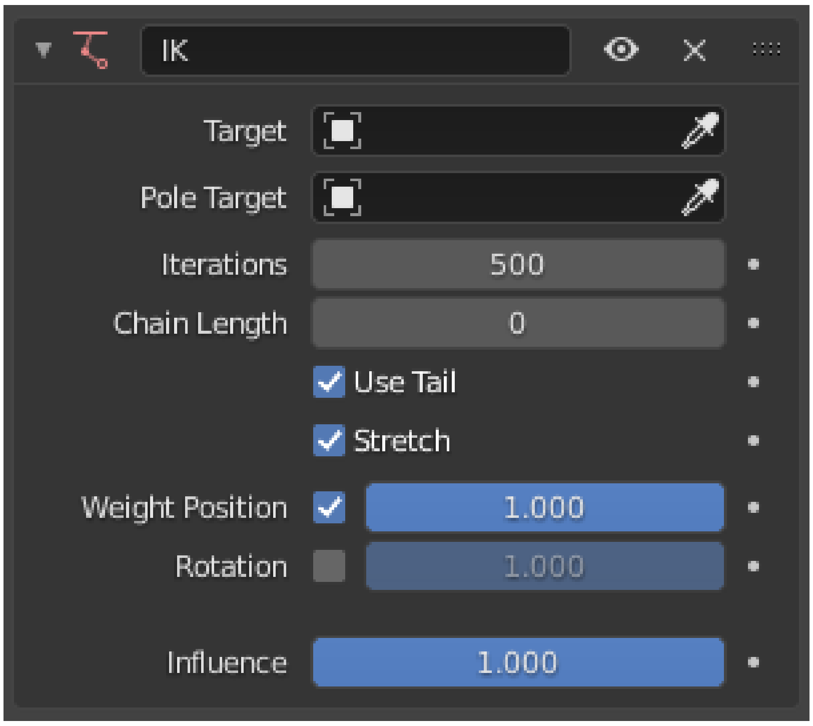
Influence 1.000

Fig. 1701: Damped Track panel.

This constraint is fully documented in the *Inverse Kinematics* page, part of the rigging chapter.

Note: The IK constraints are special in that they modify multiple bones. For this reason, they ignore their position in the stack and always run after all other constraints on the affected bones. To apply constraints after IK, it is necessary to first copy the final transformation to a new bone chain, e.g. using *Copy Transforms*.

Options



IK

Target

Pole Target

Iterations 500

Chain Length 0

✓ Use Tail

✓ Stretch

Weight Position ✓ 1.000

Rotation ☐ 1.000

Influence 1.000

Fig. 1702: Inverse Kinematics panel.

Target *Data ID* used to select the an armature. See *common constraint properties* for more information.

Pole Target Object for pole rotation.

Iterations Maximum number of solving iterations.

Chain Length How many bones are included in the IK effect. Set to 0 to include all bones.

Use Tail Include bone's tail as last element in chain.

Stretch Enable IK stretching.

Weight Position For Tree-IK: Weight of position control for this target.

Rotation Chain follows rotation of target.

Target Disable for targetless IK.

Rotation Chain follows rotation of target.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

iTaSC Solver

If the *iTaSC IK Solver* is used, the IK Solver Constraint changes to add these addition parameters.

IK Type

- **Copy Pose** Equivalent to the traditional end effector position and orientation constraint: the end effector is constrained to take the position, and optionally the orientation, of a given target, which is set in the target field.
  - **Position/Rotation Locking** Allows to obtain various effect by not constraining the coordinates along certain axis.
    - **Axis Reference** Specifies how to compute the axis coordinates.
      - **Bone** The coordinates are the position and orientation of the target relative to the bone.
      - **Target** The opposite of *Bone*, the coordinates are the position and orientation of the tip of the bone relative to the target.
- **Distance** Specify that the end effector will stay inside, at, or outside a sphere centered on the target object.
  - **Limit Mode**
    - **Inside** The end effector will stay inside of the distance from the target object.
    - **Outside** The end effector will stay outside of the distance from the target object.
    - **On Surface** The end effector will stay exactly at the distance from the target object.
  - **Distance** The radius from the target object.

Note: The *Influence* parameter is not implemented if *Pole Target* is used.

Example

A video can be found at https://vimeo.com/171279647

Locked Track Constraint

The *Locked Track* constraint is a bit tricky to explain, both graphically and textual. Basically, it is a *Track To constraint*, but with a locked axis, i.e. an axis that cannot rotate (change its orientation). Hence, the owner can only track its target by rotating around this axis, and unless the target is in the plane perpendicular to the locked axis, and crossing the owner, this owner cannot really point at its target.

Let us take the best real-world equivalent: a compass. It can rotate to point in the general direction of its target (the magnetic North, or a neighbor magnet), but it cannot point *directly at it*, because it spins like a wheel on an axle. If a compass is sitting on a table and there is a magnet directly above it, the compass cannot point to it. If we move the magnet more to one side of the compass, it still cannot point *at* the target, but it can point in the general direction of the target, and still obey its restrictions of the axle.

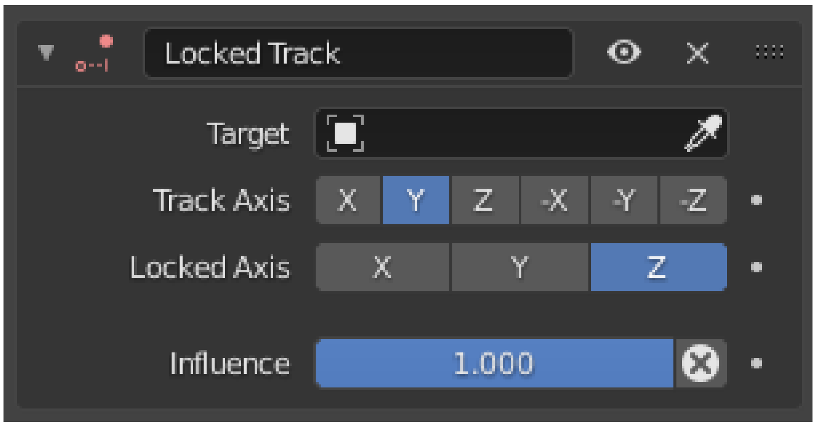
When using a *Locked Track* constraint, you can think of the target as a magnet, and the owner as a compass. The *Lock* axis will function as the axle around which the owner spins, and the *To* axis will function as the compass' needle. Which axis does what is up to you!

If you have trouble understanding the buttons of this constraint, read the tooltips, they are pretty good. If you do not know where your object's axes are, enable *Axis* in *Properties → Armature → Viewport Display*. Or, if you are working with bones, turn on the *Axes* in the armature's *Viewport Display* panel.

This constraint was designed to work cooperatively with the *Track To* constraint. If you set the axes buttons right for these two constraints, *Track To* can be used to point the axle at a primary target, and *Locked Track* can spin the owner around that axle to a secondary target.

This constraint also works very well for 2D billboarding.

Options


Locked Track

Target

Track Axis: X [✓ Y] Z -X -Y -Z

Locked Axis: X Y [✓ Z]

Influence: 1.000

Fig. 1703: Locked track panel.

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Track Axis The tracking local axis, i.e. the owner's axis to point at the target. The negative options force the relevant axis to point away from the target.

Locked Axis The locked local axis, i.e. the owner's axis which cannot be re-oriented to track the target.

Important: If you choose the same axis for *Tracking Axis* and *Locked Axis*, the constraint will no longer be functional (red state).

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171280773

Spline IK Constraint

The *Spline IK* constraint aligns a chain of bones along a curve. By leveraging the ease and flexibility of achieving aesthetically pleasing shapes offered by curves and the predictability and well-integrated control offered by bones, *Spline IK* is an invaluable tool in the riggers' toolbox. It is particularly well suited for rigging flexible body parts such as tails, tentacles, and spines, as well as inorganic items such as ropes.

To set up *Spline IK*, it is necessary to have a chain of connected bones and a curve to constrain these bones to:

- With the last bone in the chain selected, add a *Spline IK* constraint from the *Bone Constraints* tab in the *Properties*.

- Set the 'Chain Length' setting to the number of bones in the chain (starting from and including the selected bone) that should be influenced by the curve.
- Finally, set *Target* to the curve that should control the curve.

Note: The IK constraints are special in that they modify multiple bones. For this reason, they ignore their position in the stack and always run after all other constraints on the affected bones. To apply constraints after IK, it is necessary to first copy the final transformation to a new bone chain, e.g. using *Copy Transforms*.

Options

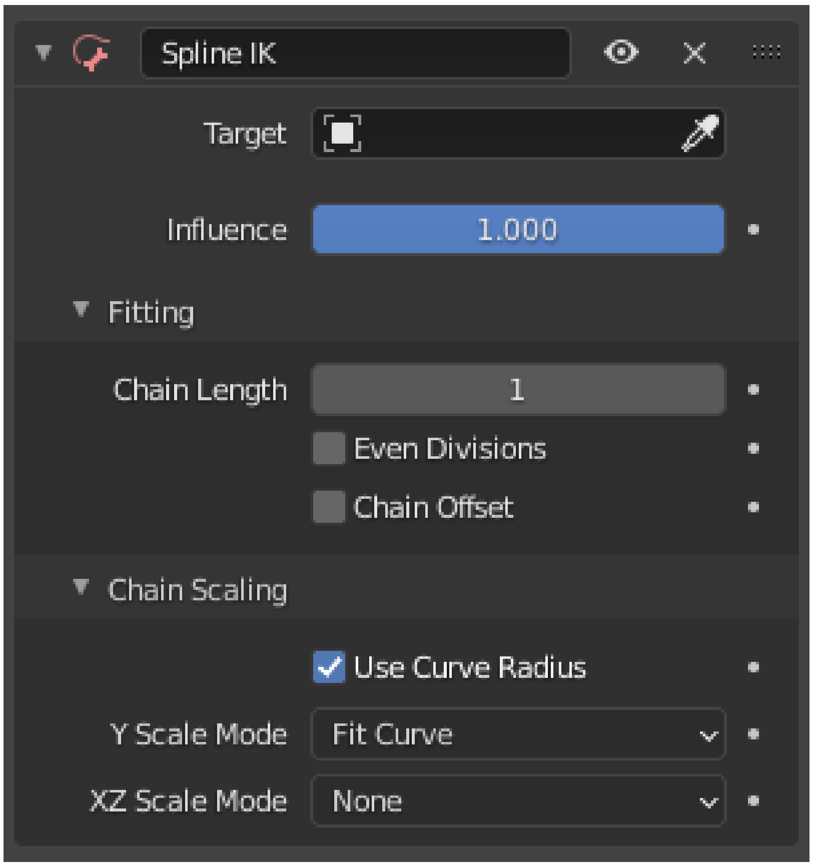

Spline IK
Target
Influence 1.000
Fitting
Chain Length 1
Even Divisions
Chain Offset
Chain Scaling
Use Curve Radius ✓
Y Scale Mode: Fit Curve
XZ Scale Mode: None

Fig. 1704: Spline IK panel.

Target *Data ID* used to select the target curve. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Fitting

Chain Length How many bones are included in the chain.

Even Division Ignore the relative length of the bones when fitting to the curve.

Chain Offset Retain the offset of the root joint from the start point of the curve.

Chain Scaling

Use Curve Radius Average radius of the endpoints is used to tweak the X and Z scaling of the bones, on top of the X and Z scale mode.

Y Scale Mode Specifies how the length of a bone is scaled when it is fitted to the curve, in addition to the effects of target curve object scale and curvature.

- **None** The bone is reset to its rest pose length.
- **Fit Curve** The bones are stretched to cover the entire length of the curve.
- **Bone Original** The original Y axis scale of the bone is used.

XZ Scale Mode Scaling that a bone undergoes in the other two directions.

- **None** Do not scale the X and Z axes.
- **Bone Original** Use the original scaling of the bones.
- **Inverse Scale** Scale of the X and Z axes is the inverse of the Y scale.
- **Volume Preservation** Similar to the *Stretch To* constraint.

Use Original Scale Specifies that *Inverse Scale* or *Volume Preservation* should be applied on top of the original scaling of the bones, like in the Stretch To constraint.

See also:

This subject is seen in-depth in the *Armature Posing section*.

Example

A video can be found at https://vimeo.com/171282278

Stretch To Constraint

The *Stretch To* constraint causes its owner to rotate and scale its Y axis towards its target. So it has the same tracking behavior as the *Track To constraint*. However, it assumes that the Y axis will be the tracking and stretching axis, and does not give you the option of using a different one.

It also optionally has some raw volumetric features, so the owner can squash down as the target moves closer, or thin out as the target moves farther away. Note however, that it is not the real volume of the owner which is thus preserved, but rather the virtual one defined by its scale values. Hence, this feature works even with non-volumetric objects, like empties, 2D meshes or surfaces, and curves.

With bones, the "volumetric" variation scales them along their own local axes (remember that the local Y axis of a bone is aligned with it, from root to tip).

Options

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Original Length This number field sets the rest distance between the owner and its target, i.e. the distance at which there is no deformation (stretching) of the owner.

- **Reset Original Length X** When clicked, this small button will recalculate the *Rest Length* value, so that it corresponds to the actual distance between the owner and its target (i.e. the distance before this constraint is applied).

Volume Variation This number field controls the amount of "volume" variation exponentially to the stretching amount. Note that the 0.0 value is not allowed, if you want to disable the volume feature, use the *None* button (see below).

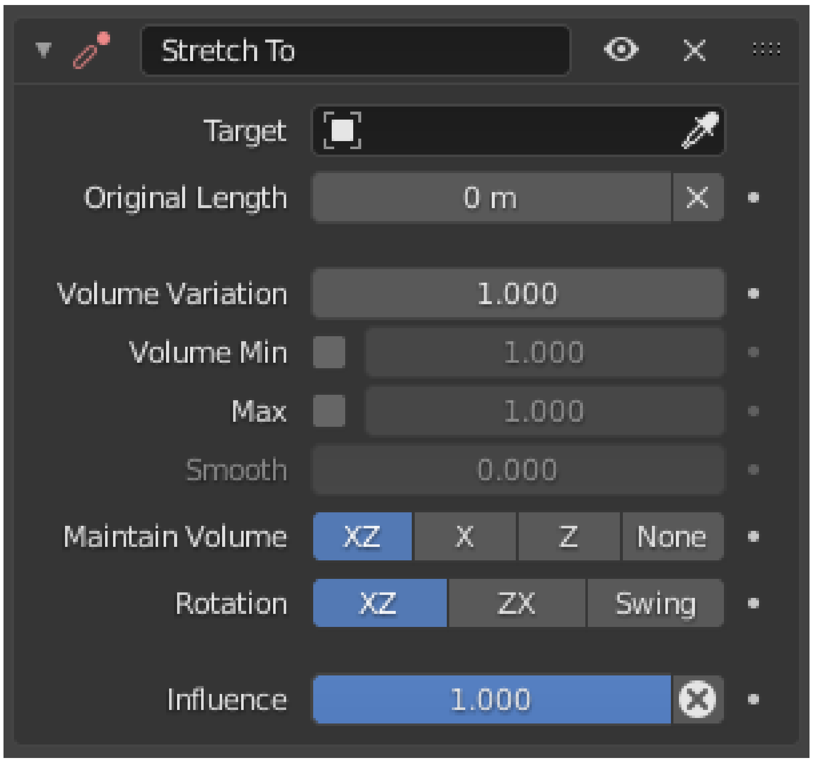


Fig. 1705: Stretch To panel.

Volume Min, Max Limits for the volume preservation to a minimum and maximum scaling each by a *Bulge* factor.

Smooth Smoothness factor to make limits less visible.

Maintain Volume These buttons control which of the X and/or Z axes should be affected (scaled up/down) to preserve the virtual volume while stretching along the Y axis. If you enable the *none* button, the volumetric features are disabled.

Rotation Specifies how the owner should be rotated to track the target with its Y axis.

XZ, ZX These buttons are equivalent to the *Up* ones of the *Track To constraint*: the owner is rotated around two local axes in the specified order.

Swing The constraint uses a single *Swing* rotation, equivalent to the *Damped Track constraint*.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171283118

Track To Constraint

The *Track To* constraint applies rotations to its owner, so that it always points a given "To" axis towards its target, with another "Up" axis permanently maintained as much aligned with the global Z axis (by default) as possible. This tracking is similar to the "billboard tracking" in 3D (see note below).

This is the preferred tracking constraint, as it has a more easily controlled constraining mechanism.

This constraint shares a close relationship to the *Inverse Kinematics constraint* in some ways.

Tip: Billboard Tracking

The term "billboard" has a specific meaning in real-time CG programming (i.e. video games!), where it is used for plane objects always facing the camera (they are indeed "trackers", the camera being their "target"). Their main usage is as support for tree or mist textures: if they were not permanently facing the camera, you would often see your trees squeezing to nothing, or your mist turning into a mille-feuille paste, which would be funny but not so credible.

Options

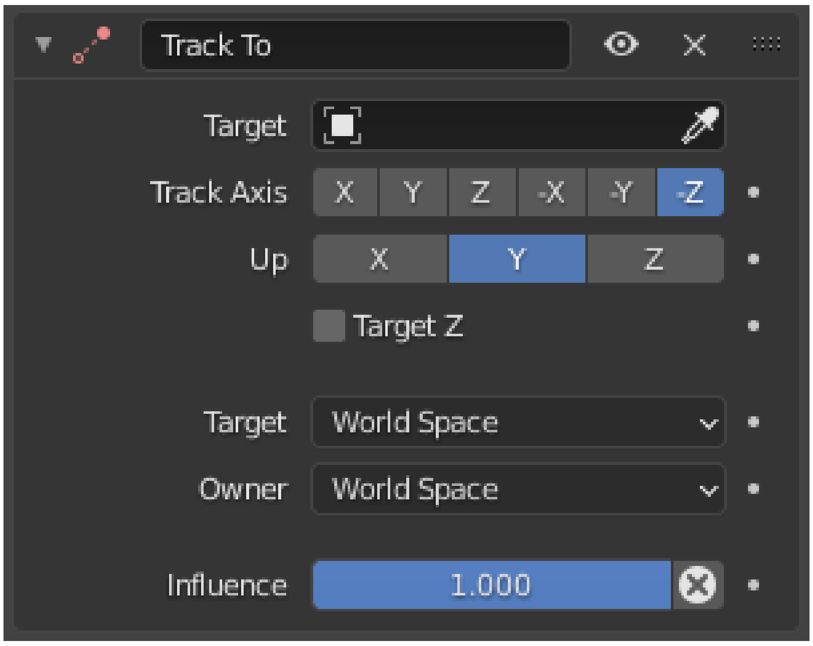

Track To
Target
Track Axis: X, Y, Z, -X, -Y, [✓ -Z]
Up: X, [✓ Y], Z
☐ Target Z
Target: World Space
Owner: World Space
Influence: 1.000

Fig. 1706: Track To panel.

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Track Axis The tracking local axis, i.e. the owner's axis to point at the target. The negative options force the relevant axis to point away from the target.

Up The "upward-most" local axis, i.e. the owner's axis to be aligned (as much as possible) with the global Z axis (or target Z axis, when the *Target* button is enabled).

Target Z By default, the owner's *Up* axis is (as much as possible) aligned with the global Z axis, during the tracking rotations. When this button is enabled, the *Up* axis will be (as much as possible) aligned with the target's local Z axis?

Target/Owner Standard conversion between spaces. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Warning: If you choose the same axis for *Tracking Axis* and *Up*, the constraint will not be functional anymore (red state).

Example

A video can be found at https://vimeo.com/171283522

Relationship

Action Constraint

The *Action* constraint is powerful. It allows you control an *Action* using the transformations of another object.

The underlying idea of the *Action* constraint is very similar to the one behind the *Drivers*, except that the former uses a whole action (i.e. multiple F-curves of the same type), while the latter controls a single F-curve of their "owner"…

Note that even if the constraint accepts the *Mesh* action type, only the *Object*, *Pose* and *Constraint* types are really working, as constraints can only affect objects' or bones' transform properties, and not meshes' shapes. Also note that only the object transformation (location, rotation, scale) is affected by the action, if the action contains keyframes for other properties they are ignored, as constraints do not influence those.

As an example, let us assume you have defined an *Object* action (it can be assigned to any object, or even no object at all), and have mapped it on your owner through an *Action* constraint, so that moving the target in the (0.0 to 2.0) range along its X axis maps the action content on the owner in the (0 to 100) frame range. This will mean that when the target's *X* property is 0.0 the owner will be as if in frame 0 of the linked action; with the target's *X* property at 1.0 the owner will be as if in frame 50 of the linked action, etc.

Options

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Evaluation Time This property allows objects to be driven without a constraint target by interpolating between the *Action Start* and *End* frames. The relative position between the start and end frame can be controlled using the value slider.

This is very helpful for more complex rigging and mechanical rigs, as it means the Action constraint can be controlled directly with a *Driver* or *Custom Property*.

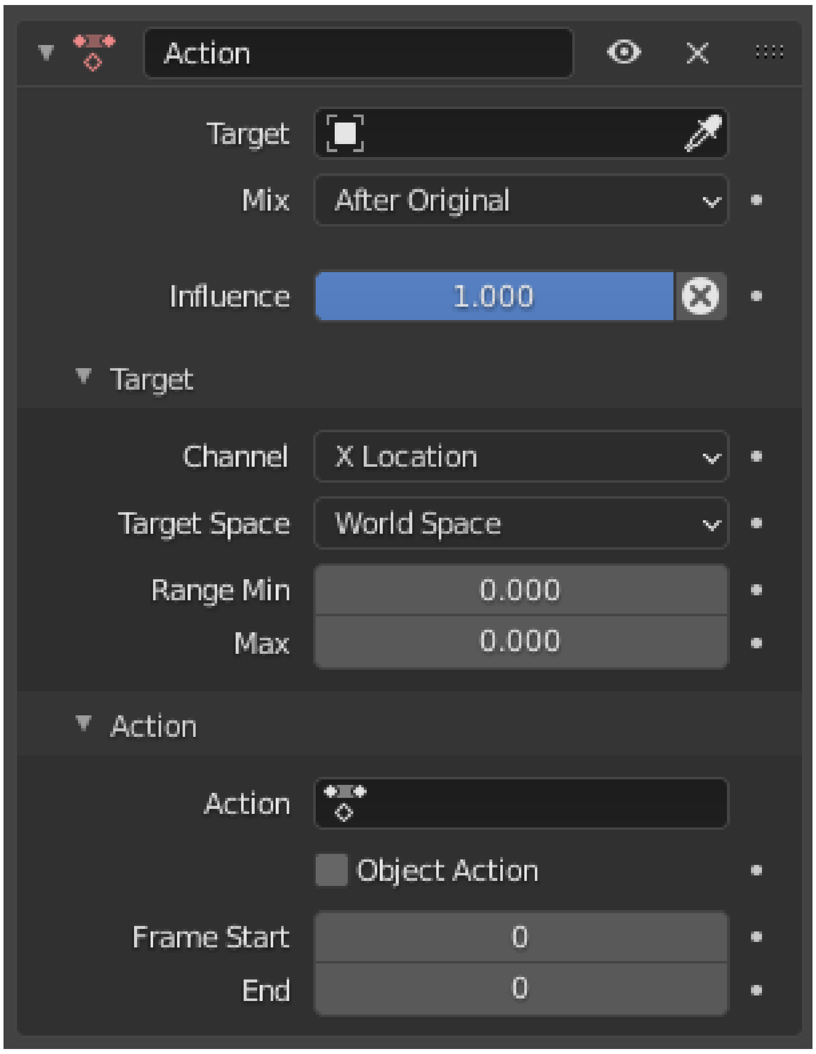
Mix Specifies how the keyframed transformation from the action is combined with the existing transformation.

Before Original The action transformation is added before the existing transformation, as if it was applied to an imaginary parent of the constraint owner. Scale is handled like in the *Aligned Inherit Scale* mode of bones to avoid creating shear.

After Original The action transformation is added after the existing transformation, as if it was applied locally to an imaginary child of the constraint owner. Scale is handled like in the *Aligned Inherit Scale* mode of bones to avoid creating shear.

After Original (Full Scale) The action transformation is added after the existing transformation, as if it was applied locally to an imaginary child of the constraint owner with ordinary parenting. This mode can create shear and is thus not recommended for use.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.


Action

Target

Mix: After Original

Influence: 1.000

Target

Channel: X Location

Target Space: World Space

Range Min: 0.000

Max: 0.000

Action

Action

Object Action

Frame Start: 0

End: 0

Fig. 1707: Action panel.

Target

Channel This selector controls which transform property (location, rotation or scale along/around one of its axes) from the target to use as "action driver".

Target Space This constraint allows you to choose in which space to evaluate its target's transform properties.

Range Min, Max The lower and upper bounds of the driving transform property value.

> **Warning:** Unfortunately, here again we find the constraint's limitations:
>
> - When using a rotation property as "driver", these values are "mapped back" to the (-180.0 to 180.0) range.
> - When using a scale property as "driver", these values are limited to null or positive values.

Action

Action Select the name of the action you want to use.

> **Warning:** Even though it might not be in red state (UI refresh problems...), this constraint is obviously not functional when this field does not contain a valid action.

Object Action Bones **only**, when enabled, this option will make the constrained bone use the "object" part of the linked action, instead of the "same-named pose" part. This allows you to apply the action of an object to a bone.

Frame Start, End The starting and ending frames of the action to be mapped.

> **Note:**
>
> - These values must be strictly positive.
> - By default, both values are set to 0, which disables the mapping (i.e. the owner just gets the properties defined at frame 0 of the linked action...).

Notes

- When the object or bone already has Action constraints, the next constraint using a newly keyframed action should be added before all others in order to get the same final combined transformation. This fact is not affected by the Mix mode.
- Unlike usual, you can have a *Start* value higher than the *End* one, or a *Min* one higher than a *Max* one: this will reverse the mapping of the action (i.e. it will be "played" reversed...), unless you have both sets reversed, obviously!
- When using a *Constraint* action, it is the constraint *channel's names* that are used to determine to which constraints of the owner apply the action. E.g. if you have a constraint channel named "trackto_empt1", its keyed *Influence* and/or *Head/Tail* values (the only ones you can key) will be mapped to the ones of the owner's constraint named "trackto_empt1".
- Similarly, when using a *Pose* action (which is obviously only meaningful and working when constraining a bone!), it is the bone's name that is used to determine which bone *channel's names* from the action to use (e.g. if the constrained bone is named "arm", it will use and only use the action's bone channel named "arm"...). Unfortunately, using a *Pose* action on a whole armature object (to affect all the keyed bones in the action at once) will not work...

- Note also that you can use the *pose library feature* to create/edit a *Pose* action data-block… just remember that in this situation, there is one pose per frame!

Example

A video can be found at https://vimeo.com/171554048

Armature Constraint

Armature is the constraint version of the *Armature Modifier*, exactly reproducing the weight-blended bone transformations and applying it to its owner orientation. It can be used like a variant of the *Child Of* constraint that can handle multiple parents at once, but requires all of them to be bones.

Note: Unlike the Armature modifier, the constraint does not take the *Deform* checkbox of bones into account, and can use any bone as target.

Options

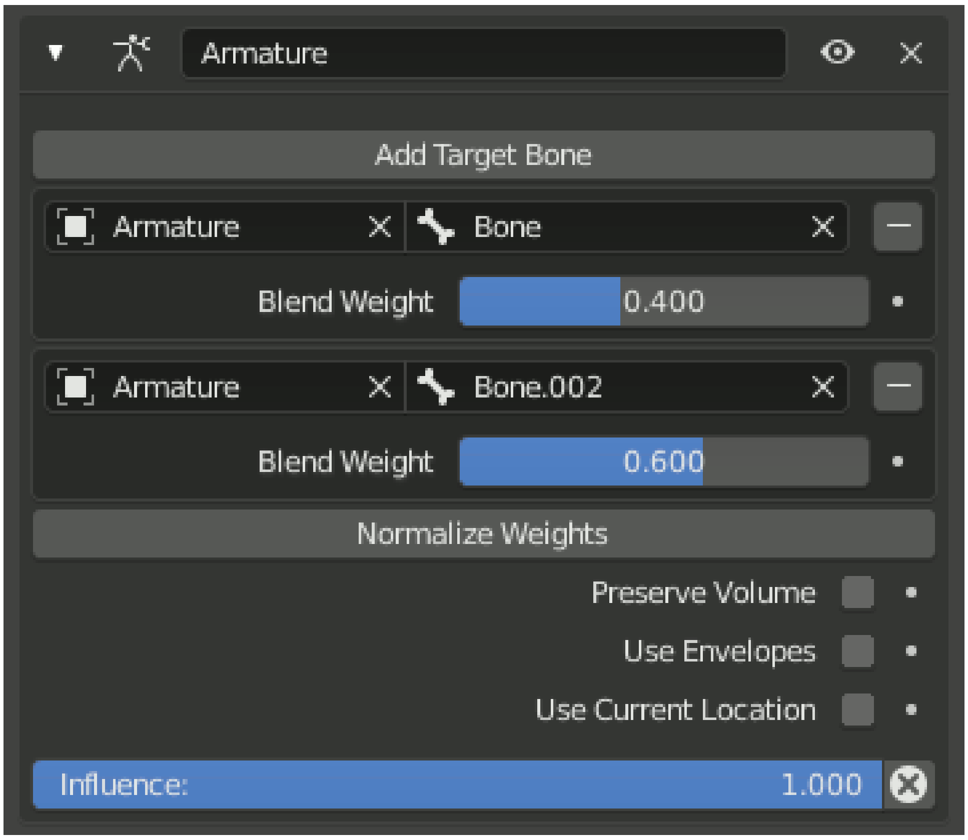

Armature
Add Target Bone
Armature | Bone
Blend Weight 0.400
Armature | Bone.002
Blend Weight 0.600
Normalize Weights
Preserve Volume
Use Envelopes
Use Current Location
Influence: 1.000

Fig. 1708: Armature constraint.

Preserve Volume Like the matching option of the modifier, it enables the use of quaternions for preserving the volume of the object during deformation.

Use Envelopes To approximate envelope-only behavior of the modifier, add all relevant bones with weight 1.0 and enable this option.

Note: Unlike the modifier, the constraint always requires explicitly listing all of its target bones with associated weights. This option merely enables envelopes for all bones, as if they had *Envelope Multiply* enabled.

Use Current Location Only for constraints on bones: Instead of using the rest location, use the current location of the owner bone to compute envelope weights or binding to B-Bone segments.

With envelope weights, this can be used to change the active "parent" bone of the owner bone dependent on its location. For non-bones this mode is always active, because they don't have a rest location.

Add Target Bone This button adds a new empty entry at the end of the target list.

Normalize Weights This button normalizes all weight values in the target list so that they add up to 1.0.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Bones

This specifies the list of bones used by the constraint to deform its owner. Each target bone has the following input fields and controls:

Target Unlike the modifier, the constraint can use bones coming from different armatures at the same time. See *common constraint properties* for more information.

Sub-target Name of the target bone.

Remove Button `X` Removes the entry from the target list.

Weight Weight associated with the bone, equivalent to vertex groups in the modifier.

Child Of Constraint

Child Of is the constraint version of the standard parent/children relationship between objects (the one established through the `Ctrl-P` shortcut, in the 3D Views).

Parenting with a constraint has several advantages and enhancements, compared to the traditional method:

- You can have several different parents for the same object (weighting their respective influence with the *Influence* slider).
- As with any constraint, you can key (i.e. animate) its Influence setting. This allows the object which has a Child Of constraint upon it to change over time which target object will be considered the parent, and therefore have influence over it.

Important: Do not confuse this "basic" object parenting with the one that defines the *chains of bones* inside of an armature. This constraint is used to parent an object to a bone (the so-called *object skinning*), or even bones to bones. But do not try to use it to define chains of bones.

Options

Target The target object that this object will act as a child of. *Data ID* used to select the constraint's target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

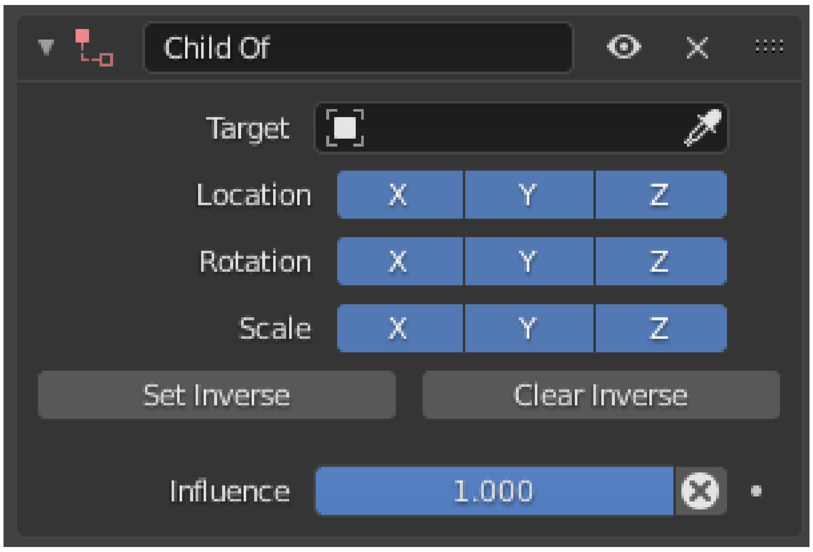


Fig. 1709: Child Of panel.

Location Each of these buttons will make the parent affect or not affect the location along the corresponding axis.

Rotation Each of these buttons will make the parent affect or not affect the rotation around the corresponding axis.

Scale Each of these buttons will make the parent affect or not affect the scale along the corresponding axis.

Set Inverse By default, when you parent your owner to your target, the target becomes the origin of the owner's space. This means that the location, rotation and scale of the owner are offset by the same properties of the target. In other words, the owner is transformed when you parent it to your target. This might not be desired! So, if you want to restore your owner to its before-parenting state, click on the *Set Inverse* button.

Clear Inverse This button reverses (cancels) the effects of the above one, restoring the owner/child to its default state regarding its target/parent.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Tips

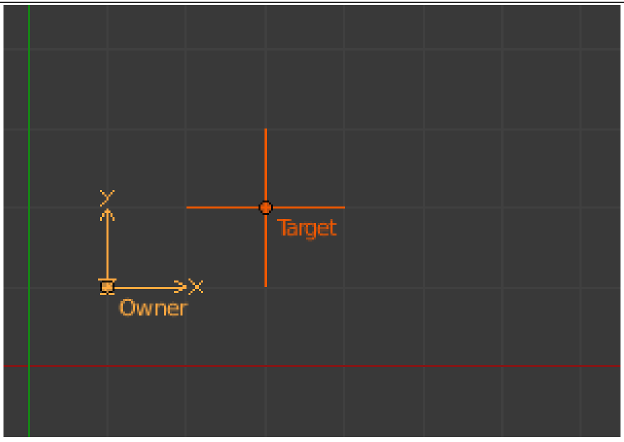
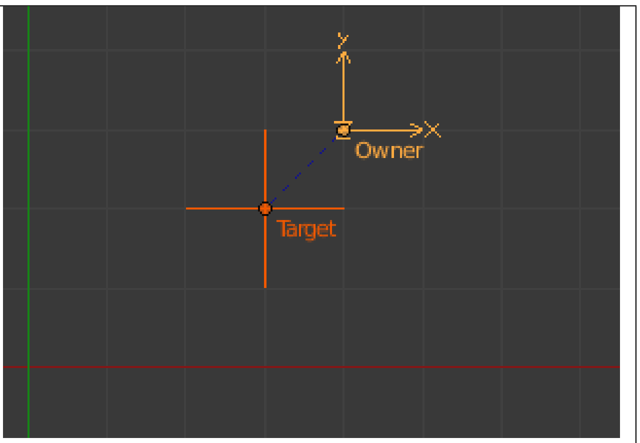
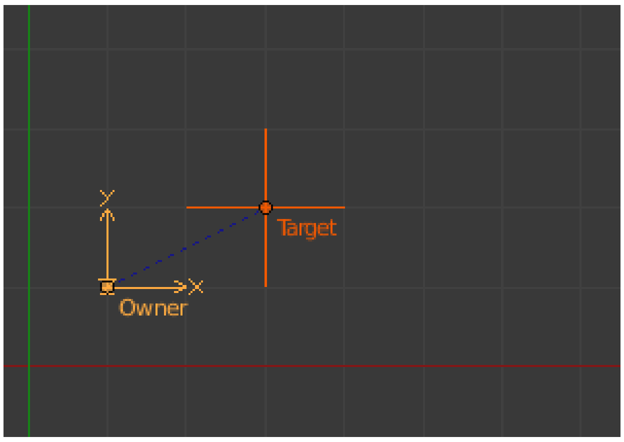
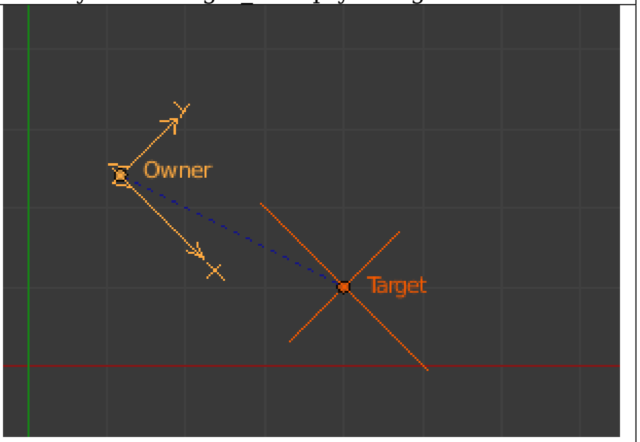
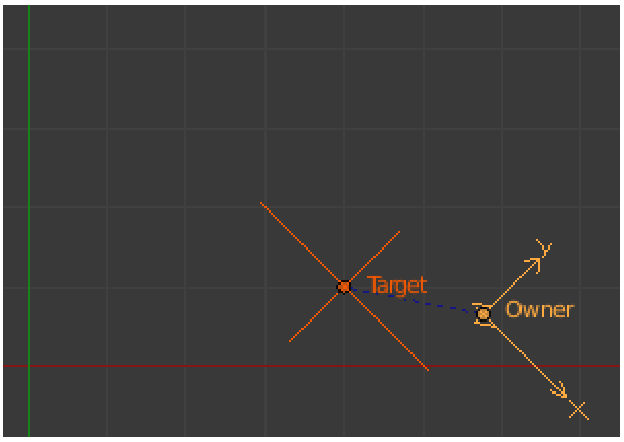
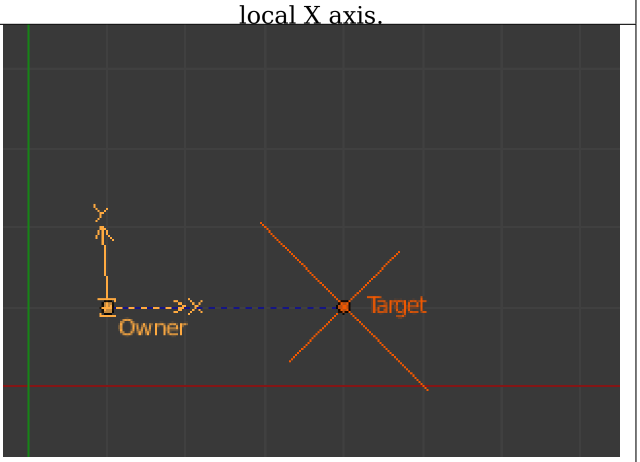
When creating a new parent relationship using this constraint, it is usually necessary to click on the *Set Inverse* button after assigning the parent. As noted above, this cancels out any unwanted transform from the parent, so that the owner returns to the location/rotation/scale it was in before the constraint was applied. Note that you should apply *Set Inverse* with all other constraints disabled (their *Influence* set to 0.0) for a particular *Child Of* constraint, and before transforming the target/parent (see example below).

About the toggle buttons that control which target's (i.e. parent's) individual transform properties affect the owner, it is usually best to leave them all enabled, or to disable all three of the given Location, Rotation and Scale transforms.

Technical Note

If you use this constraint with all channels on, it will use a straight matrix multiplication for the parent relationship, not decomposing the parent matrix into loc/rot/size. This ensures any transformation correctly gets applied, also for combinations of rotated and non-uniform scaled parents.

Examples

 Fig. 1710: No constraint. Note the position of Owner empty 1.0 unit along the X and Y axes.	 Fig. 1711: Child Of just added. Here you can see that Owner empty is now 1.0 unit away from Target_1 empty along X and Y axes.
 Fig. 1712: Offset set. Set Inverse has been clicked, and Owner is back to its original position.	 Fig. 1713: Target/parent transformed. Target_1 has been moved along the XY plane, rotated around the Z axis, and scaled along its local X axis.
 Fig. 1714: Offset cleared. Clear Inverse has been clicked. Owner is fully again controlled by Target_1.	 Fig. 1715: Offset set again. Set Offset has been clicked again. As you can see, it does not gives the same result as in (Target/parent transformed). As noted above, use Set Inverse only once, before transforming your target/parent.

A video can be found at https://vimeo.com/171554131

Floor Constraint

The *Floor* constraint allows you to use its target position (and optionally rotation) to specify a plane with a "forbidden side", where the owner cannot go. This plane can have any orientation you like. In other words, it creates a floor (or a ceiling, or a wall)! Note that it is only capable of simulating entirely flat planes, even if you use the *Vertex Group* option. It cannot be used for uneven floors or walls.

Options

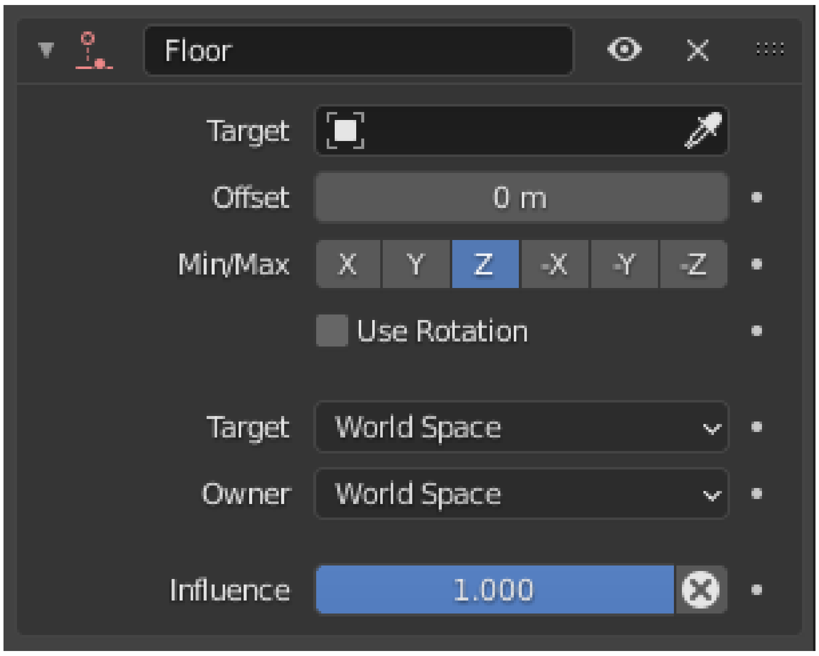

Floor
Target
Offset 0 m
Min/Max X Y Z -X -Y -Z
Use Rotation
Target World Space
Owner World Space
Influence 1.000

Fig. 1716: Floor panel.

Target *Data ID* used to select the constraints target, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Offset Allows you to offset the "floor" plane from the target's origin, by the given number of units. Use it e.g. to account for the distance from a foot bone to the surface of the foot's mesh.

Max/Min Controls which plane will be the "floor". The names of the buttons correspond, indeed, to the *normal* to this plane (e.g. enabling Z means "XY plane", etc.). By default, these normals are aligned with the *global* axes. However, if you enable *Use Rotation* (see above), they will be aligned with the *local target's axes*. As the constraint does not only define an uncrossable plane, but also a side of it which is forbidden to the owner, you can choose which side by enabling either the positive or negative normal axis… e.g. by default Z, the owner is stuck in the positive Z coordinates.

Use Rotation Forces the constraint to take the target's rotation into account. This allows you to have a "floor" plane of any orientation you like, not just the global XY, XZ and YZ ones…

Target/Owner Standard conversion between spaces. See *common constraint properties* for more information.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171554207

Follow Path Constraint

The *Follow Path* constraint places its owner onto a *curve* target object, and makes it move along this curve (or path). It can also affect its owner's rotation to follow the curve's bends, when the *Follow Curve* option is enabled.

It could be used for complex camera traveling, a train on its rails and most other vehicles can also use "invisible" tracks, the links of a bicycle chain, etc.

The owner is always evaluated in the global (world) space:

- Its location (as shown in the *Transform* panel) is used as an offset from its normal position on the path. E.g. if you have an owner with the (1.0, 1.0, 0.0) location, it will be one unit away from its normal position on the curve, along the X and Y axis. Hence, if you want your owner *on* its target path, clear its location `Alt-G`!
- This location offset is also proportionally affected by the scale of the target curve. Taking the same (1.0, 1.0, 0.0) offset as above, if the curve has a scale of (2.0, 1.0, 1.0), the owner will be offset two units along the X axis (and one along the Y one)…
- When the *Follow Curve* option is enabled, its rotation is also offset to the one given by the curve. E.g. if you want the Y axis of your object to be aligned with the curve's direction, it must be in rest, non-constrained state, aligned with the global Y axis. Here again, clearing your owner's rotation `Alt-R` might be useful…

The movement of the owner along the target curve/path may be controlled in two different ways:

- The most simple is to define the number of frames of the movement, in the *Path Animation panel* of the Curve tab, via the *Frames* number field, and its start frame via the constraint's Offset option (by default, start frame: 1 [= offset of 0], duration: 100).
- The second way, much more precise and powerful, is to define an *Evaluation Time* interpolation curve for the *Target* path (in the *Graph Editor*). See the *Graph Editor chapter* to learn more about F-curves.
- If you do not want your owner to move along the path, you can give to the target curve a flat *Speed* F-curve (its value will control the position of the owner along the path).

Follow Path is another constraint that works well with the *Locked Track one*. One example is a flying camera on a path. To control the camera's roll angle, you can use a *Locked Track* and a target object to specify the up direction, as the camera flies along the path.

Note: *Follow Path & Clamp To*

Do not confuse these two constraints. Both of them constraint the location of their owner along a curve, but *Follow Path* is an "animation-only" constraint, inasmuch as the position of the owner along the curve is determined by the time (i.e. current frame), whereas the *Clamp To constraint* determines the position of its owner along the curve using one of its location properties' values.

Note: Note that you also need to keyframe Evaluation Time for the Path. Select the path, go to the *Path Animation* panel in the curve properties, set the overall frame to the first frame of the path (e.g. frame 1), set the value of Evaluation time to the first frame of the path (e.g. 1), right-click on Evaluation time, select create keyframe, set the overall frame to the last frame of the path (e.g. frame 100), set the value of Evaluation time to the last frame of the path (e.g. 100), right-click on Evaluation time, select create keyframe.

Options

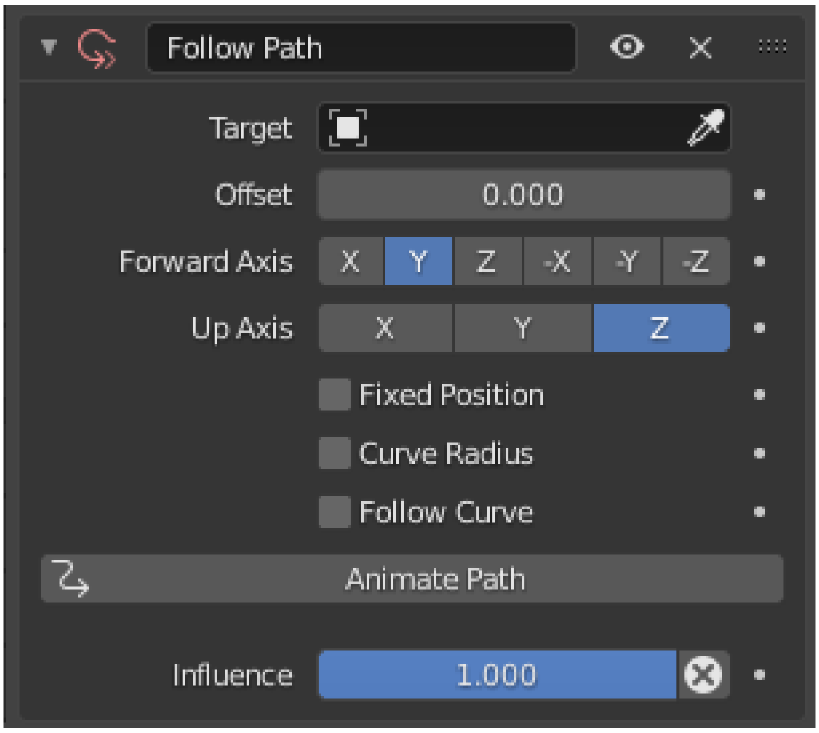


Fig. 1717: Follow Path panel.

Target *Data ID* used to select the constraint's target, which *must* be a curve object, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Offset The number of frames to offset from the "animation" defined by the path (by default, from frame 1).

Forward Axis The axis of the object that has to be aligned with the forward direction of the path (i.e. tangent to the curve at the owner's position).

Up Axis The axis of the object that has to be aligned (as much as possible) with the world Z axis. In fact, with this option activated, the behavior of the owner shares some properties with the one caused by a *Locked Track constraint*, with the path as "axle", and the world Z axis as "magnet".

Fixed Position Object will stay locked to a single point somewhere along the length of the curve regardless of time.

Curve Radius Objects scaled by the curve radius. See *Curve Editing*.

Follow Curve If this option is not activated, the owner's rotation is not modified by the curve; otherwise, it is affected depending on the Forward and Up Axes.

Animate Path Adds an F-curve with options for the start and end frame. ToDo: from above.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171554266

Pivot Constraint

The *Pivot* constraint allows the owner to rotate around a target object. It was originally intended for pivot joints found in humans e.g. fingers, feet, elbows, etc.

Options

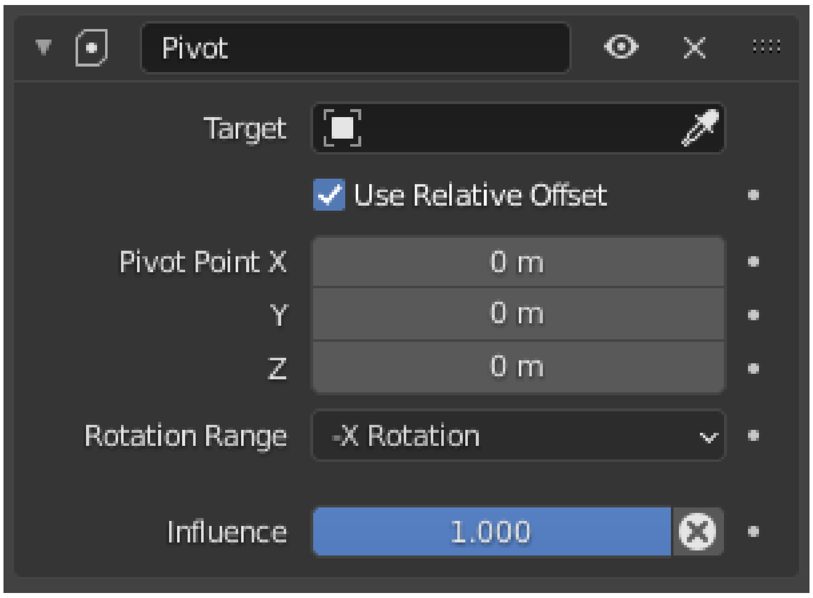

Pivot
Target
✓ Use Relative Offset
Pivot Point X 0 m
Y 0 m
Z 0 m
Rotation Range -X Rotation
Influence 1.000

Fig. 1718: Pivot panel.

Target *Data ID* for the selection of the object to be used as a pivot point. See *common constraint properties* for more information.

Use Relative Offset Offset will be an absolute point in space instead of relative to the target.

Pivot Point X, Y, Z Offset of pivot from target.

Rotation Range Rotation range on which pivoting should occur.

- **Always** Use the pivot point in every rotation.
- **-X/-Y/-Z/X/Y/Z Rotation** Use the pivot point in the corresponding direction around the corresponding axis.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Example

A video can be found at https://vimeo.com/171554353

Shrinkwrap Constraint

The *Shrinkwrap* constraint is the "object counterpart" of the *Shrinkwrap Modifier*. It moves the owner origin and therefore the owner object's location to the surface of its target.

This implies that the target *must* have a surface. In fact, the constraint is even more selective, as it can only use meshes as targets. Hence, the *Shrinkwrap* option is only shown in the *Add Constraint to Active Object* menu, `Ctrl-Alt-C`, (or its bone's equivalent), when the selected inactive object is a mesh.

Options

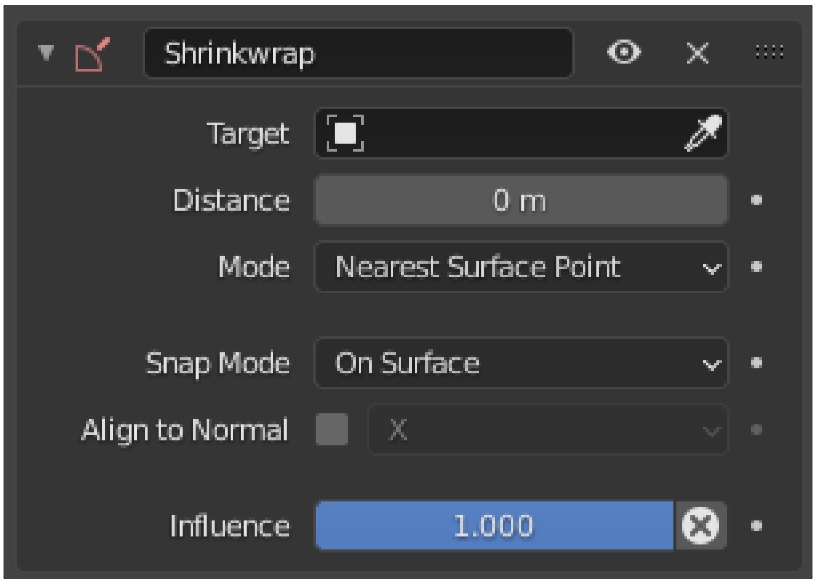

Shrinkwrap
Target
Distance 0 m
Mode Nearest Surface Point
Snap Mode On Surface
Align to Normal X
Influence 1.000

Fig. 1719: Shrinkwrap panel.

Target *Data ID* used to select the constraint's target, which *must* be a mesh object, and is not functional (red state) when it has none. See *common constraint properties* for more information.

Distance This number field controls the offset of the owner from the shrunk computed position on the target's surface.

Influence Controls the percentage of affect the constraint has on the object. See *common constraint properties* for more information.

Mode

This selector allows you to select which method to use to compute the point on the target's surface to which to move the owner's origin. You have these options:

Nearest Surface Point

The chosen target's surface's point will be the nearest one to the original owner's location. This is the default and most commonly useful option.

Projection

The target's surface point is determined by projecting the owner's origin along a given axis.

Project Axis This axis is controlled by the radio buttons that show up when you select this type. This mean the projection axis can only be aligned with one of the three axes, or their opposites. When the projection of the owner's origin along the selected direction does not hit the target's surface, the owner's location is left unchanged.

+X, +Y, +Z, -X, -Y, -Z

Space Coordinate space in which the axis direction is specified.

Distance Distance cutoff after which projection is assumed to have failed, leaving the location unchanged.

Project Opposite In addition to the selected projection axis, project in the opposite direction and choose the closest hit.

Face Cull This radio button allows you to prevent any projection over the "front side" (respectively the "back side") of the target's faces. The "side" of a face is determined by its normal (front being the side "from where" the normal "originates").

Off, Front, Back

Invert Cull When used with *Project Opposite* and *Face Culling*, it inverts the *Front* or *Back* cull choice for the opposite direction.

Nearest Vertex

This method is very similar to the *Nearest Surface Point* one, except that the owner's possible shrink locations are limited to the target's vertices.

This method doesn't support the *Snap Mode* setting described below.

Target Normal Projection

This method is similar to *Nearest Surface Point*, but produces a much smoother projection in return for being significantly slower.

Instead of finding the closest point, it searches for the nearest point that has its interpolated smooth normal pointing towards or away from the original owner position. Non-manifold boundary edges are specially handled as infinitely thin cylinders that emit normals in all perpendicular directions. Ignores flat shading and auto smooth settings.

Snap Mode

Most Shrinkwrap types support an additional setting to control how the owner is moved to the target point selected by the methods described above. Some of the choices only differ if *Distance* is not zero.

On Surface The owner location is always changed. The offset is applied along the projection line connecting the original owner location and selected target point towards the original position.

Outside Surface Like *On Surface*, but the offset is always applied towards the outside of the target.

Above Surface Like *On Surface*, but the offset is applied along the smooth normal of the target.

Inside The owner is not moved if it is already inside the target. Offset shrinks the allowed volume towards the inside along the projection line.

Outside The owner is not moved if it is already outside the target. Offset expands the exclusion volume towards the outside along the projection line.

The *Inside* and *Outside* options can be used for very crude collision detection. The inside vs outside determination is done based on the target normal and is not always stable near 90 degree and sharper angles in the target mesh.

Align To Normal

Whenever *Snap Mode* is available, it is also possible to align the specified local axis of the object to the smooth normal of the target at the selected point. The axis is selected via radio buttons.

The alignment is performed via smallest rotation, like in *Damped Track* constraint.

Example

A video can be found at https://vimeo.com/171554427

2.8.6 Actions

When animating objects and properties in Blender, Actions record and contain the data. As everything else in Blender, Actions are data-blocks.

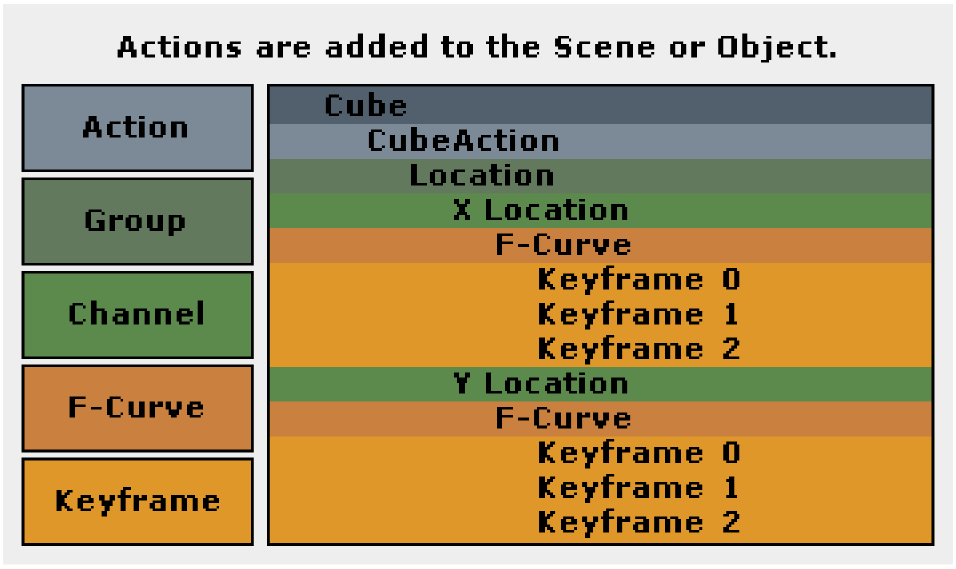


Fig. 1720: Actions.

So when you animate an object by changing its location with keyframes, the animation is saved to the Action.

Each property has a channel which it is recorded to, for example, `Cube.location.x` is recorded to Channel X Location. The *X location* and *Y location* properties can be shared across multiple objects, if all objects have *X location* and *Y location* properties beneath them.

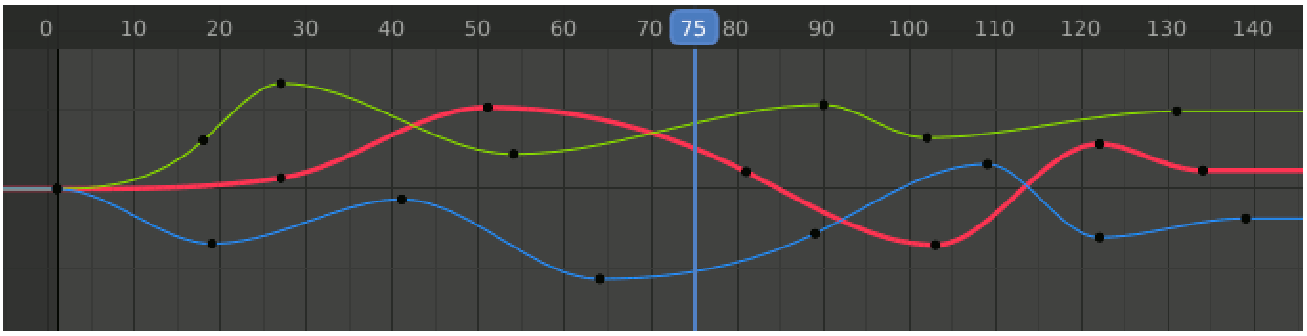


Fig. 1721: Graph Editor. Each channel has an F-curve represented by the lines between the keyframes.

Actions Record and contain animation data.

Groups Are groups of channels.

Channels Record properties.

F-Curves *F-Curves* are used to interpolate the difference between the keyframes.

Keyframes *Keyframes* are used to set the values of properties bound to a point in time.

Working with Actions

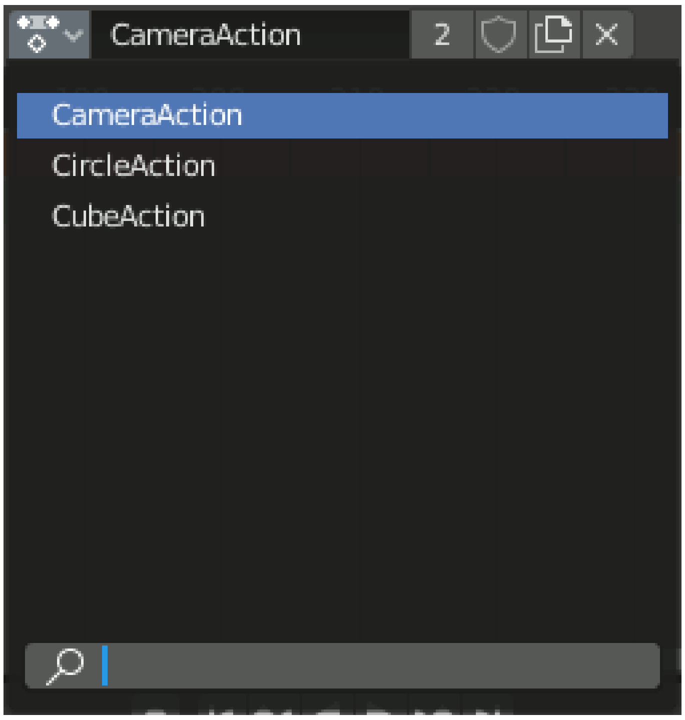


Fig. 1722: The Action data-block menu.

When you first animate an object by adding keyframes, Blender creates an *Action* to record the data.

Actions can be managed with the *Action data-block menu* in the Dope Sheet *Action Editor* header, or the Sidebar region of the *NLA Editor*.

If you are making multiple actions for the same object, press the shield button for each action. This will give the actions a *Fake User* and will make Blender save the unlinked actions.

Objects can only use one *Action* at a time for editing. The *NLA Editor* is used to blend multiple actions together.

Bake Action

Reference

Editor 3D View

Mode Object and Pose Modes

Menu *Object/Pose* → *Animation* → *Bake Action…*

The final motion of objects or bones depends not only on the keyframed animation, but also on any active F-curve modifiers, drivers, and constraints. On each frame of all the scene's frames, the *Bake Action* tool computes the final animation of the selected objects or bones with all those modifiers, drivers, and constraints applied, and keyframes the result.

This can be useful for adding deviation to a cyclic action like a *Walk Cycle*, or to create a keyframe animation created from drivers or constraints.

2.8.7 Drivers

Introduction

Drivers are a way to control values of properties by means of a function, or a mathematical expression.

Effectively, drivers consist of:

- A **driver configuration** that specifies zero, one, or more input values using other properties or object transformation channels, and combines them using a predefined mathematical function or a custom Python expression.
- An **animation** *F-Curve* that maps the output of the driver configuration to the final value to apply to the driven property.

As an example, the rotation of Object 1 can be controlled by the scale of Object 2. It is then said that the scale of Object 2 *drives* the rotation of Object 1.

Not only can drivers directly set the value of a property to the value of a different one, they can also combine multiple values using a fixed function or a Python expression and further modulate it with a manually defined curve and/or a modifier stack.

Drivers are an extremely powerful tool for building rigs and are typically used to drive bone transforms and the influence of shape keys, action constraints and modifiers, often using custom properties as inputs.

Graph View

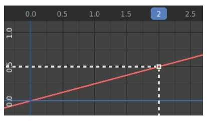


Fig. 1723: Driver curve in the Drivers editor.

The main area of the *Drivers editor* shows an *F-Curve* that represents the driver function.

The **X axis** maps to the output value of the driver configuration. The units depend on the setup.

The **Y axis** shows the value applied to the target property. The units depend on the property.

In the example image, if the driver value is 2.0 the property value will be 0.5.

The default F-curve is an identity map, i.e. the value produced by the driver configuration is applied to the driven property unchanged. If the driver output value is 2.0, the property will be 2.0.

The driver function can be defined artistically with Bézier curve handles or mathematically with trigonometric functions or polynomial expressions such as $y = a + bx$. Furthermore, the function can also be procedurally modulated with noise or cyclic repetitions. See *Modifiers* for more details.

Driver Configuration

The *Drivers panel* shows the setup for a driver.

A driver can have zero, one, or more **variables**. Variables specify which properties, object transformation channels, or relative distances between objects, are used as inputs by the driver.

The driver **type** determines how the variables are used. The type can be:

- a built-in function: for example, the sum of the variables' values, or

- a scripted expression: an arbitrary Python expression that refers to the variables by their names.

This driver configuration outputs a single value which changes when the variables change. This value is then evaluated through the driver function curve to produce the result to be applied to the driven property.

Notes on Scripted Expressions

When a driver uses a *Scripted Expression*, Blender can evaluate it without using the fully featured Python interpreter if it is simple enough. This means that drivers are fast to evaluate with simple divisions, additions and other "simple" expressions. The built-in functions are always evaluated natively.

See *Simple Expressions* for a comprehensive list of expressions that can be evaluated natively.

When the expression is not simple, it will be evaluated using Python. As a consequence, the driver will be slower and there is a security risk if the author of the Python code is unknown. This is an important thing to take into consideration for heavy scenes and when sharing files with other people. See also: *Auto run*.

Usage

Drivers can be added to properties via their context menu, a shortcut, copy-pasted, or by typing an expression directly into the property's value.

After adding drivers, they are usually modified in the *Drivers editor*, or via a simplified *Edit Driver* popover invoked from the property context menu.

Add Driver

Reference

Menu *Context menu → Add Driver*

Hotkey `Ctrl-D`

The usual way to add a driver to a property is to `RMB` click a property, then choose *Add Driver* in the context menu. Drivers can also be added by pressing `Ctrl-D` with the mouse over the property.

This operation adds a driver with a single variable (which needs to be filled in), and displays the *Edit Driver* popover.

Edit Driver

Reference

Menu *Context menu → Edit Driver*

Displays a popover window that allows editing the custom expression and input variables of the driver without opening the full *Drivers Editor*.

Many drivers don't use their *F-Curve* component, so this reduced interface is sufficient.

Open Drivers Editor

Reference

Menu *Context menu → Open Drivers Editor*

Opens a new window with the *Drivers Editor* and selects the driver associated with the property.

Copy & Paste

Reference

Menu *Context menu → Copy Driver*

Menu *Context menu → Paste Driver*

Drivers can be copied and pasted via the context menu. When adding drivers with the same settings, this can save time modifying settings.

Copy As New Driver

Reference

Menu *Context menu → Copy As New Driver*

A driver that sets the property value to the value of a different property can be quickly created by using the *Copy As New Driver* context menu option of the input property, and then pasting the result onto the output property via *Paste Driver*.

It is also possible to add the new driver variable to an existing driver using the *Paste Driver Variables* button in the editor panel.

Expression

This is a quick way to add drivers with a scripted expression. First click the property you want to add a driver to, then type a hash # and a scripted expression.

Some examples:

- `#frame`
- `#frame / 20.0`
- `#sin(frame)`
- `#cos(frame)`

Removing Drivers

Reference

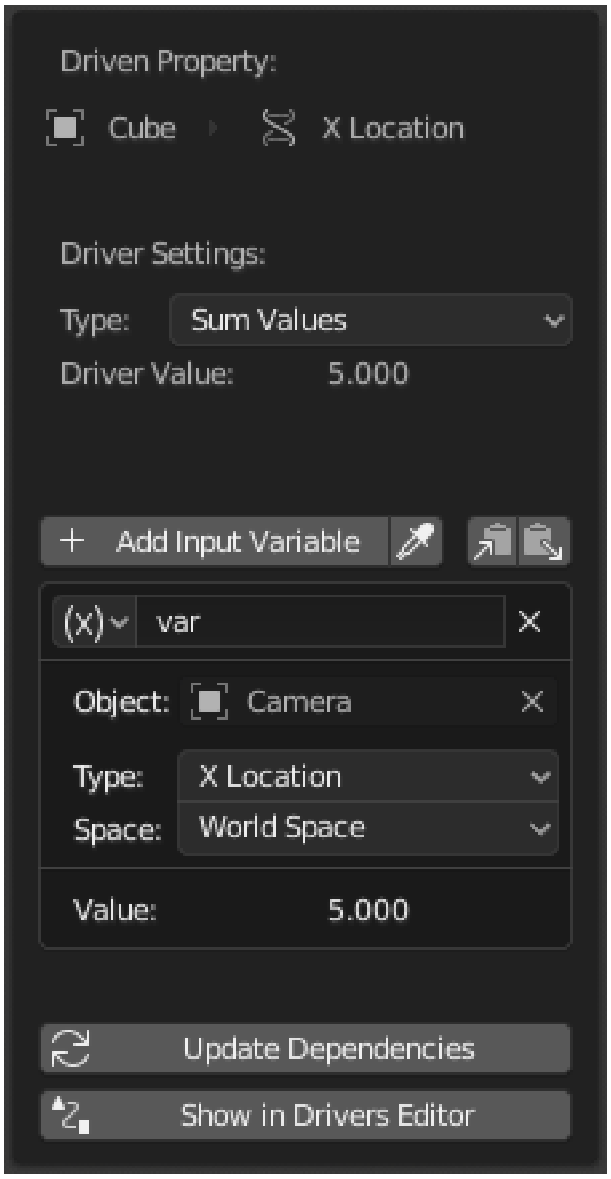
Menu *Context menu → Delete Driver(s)*

Menu *Context menu → Delete Single Driver*

Hotkey `Ctrl-Alt-D`

Removes driver(s) associated with the property, either for the single selected property or sub-channel, or all components of a vector.

Drivers Panel



Driven Property:

Cube ▸ X Location

Driver Settings:

Type: Sum Values

Driver Value: 5.000

var

Object: Camera

Type: X Location

Space: World Space

Value: 5.000

Fig. 1724: Edit Driver popover.

Reference

- **Editor** Graph editor
- **Mode** Drivers
- **Panel** *Sidebar region → Drivers*
- **Hotkey** `N`

Reference

Menu *Context menu → Edit Driver*

Hotkey `Ctrl-D`

This panel is visible in Sidebar of the *Drivers Editor* or as a popover when adding a driver to a property.

It shows the property that is being driven, followed by a series of settings that determine how the driver works.

Driver Settings

Type

There are two categories of drivers:

- **Built-in functions** (*Average*, *Sum*, *Min* and *Max*)

  The driven property will have the value of the average, sum, lowest or highest (respectively) of the values of the referenced *Driver Variables*. If there is only one driver variable, these functions will yield the same result.

- **Custom** (*Scripted Expression*).

  An arbitrary Python expression that can refer to the *Driver Variables* by name. See *Expressions*.

Driver Value

The current result of the driver setup. Useful for debug purposes.

Variables

See *Driver Variables*.

Update Dependencies

Forces an update for the Driver Value dependencies.

Show in Drivers Editor

Opens the fully featured *Drivers Editor*. This button only appears in the popover version of the Drivers panel.

Driver Variables

Variables are references to properties, transformation channels, or the result of a comparison between transformations of two objects.

Drivers should access object data via *Driver Variables*, rather than direct references in the Python expression, in order for dependencies to be correctly tracked.

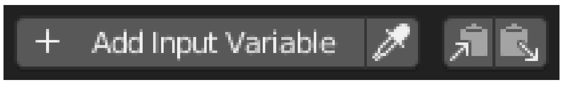
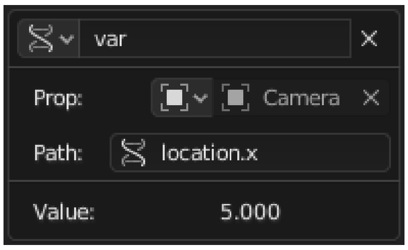
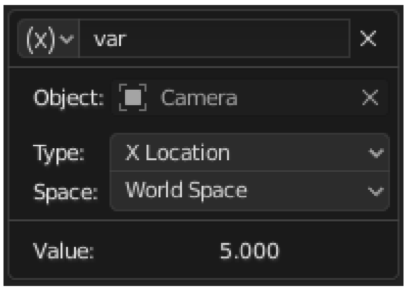
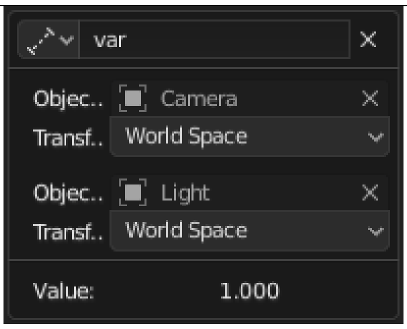


Fig. 1725: Add, Copy, Paste buttons.

Add Input Variable Adds a new Driver Variable.

Copy/Paste Variables Copies the current variable list so it can be pasted into another driver's variable list.

 var Prop: Camera Path: location.x Value: 5.000 Fig. 1726: Single property.	 (x) var Object: Camera Type: X Location Space: World Space Value: 5.000 Fig. 1727: Transform channel.	 var Objec.. Camera Transf.. World Space Objec.. Light Transf.. World Space Value: 1.000 Fig. 1728: Distance.

Name Name for use in scripted expressions. The name must start with a letter, and only contain letters, digits, or underscores.

Variable Type The type of variable to use.

Single Property Retrieves the value of a RNA property, specified by a data-block reference and a path string.

In case of transform properties, this will return the exact value of the UI property, while Transform Channel will take parenting and/or constraints into account as needed.

See also *Custom Properties*.

ID Type The ID-block type. For example: Key, Image, Object, Material.

ID The ID of the ID-block type. For example: "Material.001".

RNA Path The RNA name of the property, based on a subset of Python attribute access syntax. For example: `location.x` or `location[0]` for the raw X location value, or `["prop_name"]` for a custom property.

Tip: The easiest way to create a variable of this type is to use the *Copy As New Driver* context menu option of the input property, and paste the result into the driver via *Paste Driver Variables*.

Transform Channel Retrieves the value of a Transform channel from an object or bone.

ID ID of the object. For example: Cube, Armature, Camera.

Bone ID of the Armature bone. For example: "Bone", "Bone.002", "Arm.r". This option is for armatures.

Type For example, X Location, X Rotation, X Scale.

The *Average Scale* option retrieves the combined scale value, computed as the cubic root of the total change in volume. Unlike *X/Y/Z Scale*, this value can be negative if the object is flipped by negative scaling.

Mode (Rotation) For rotation channels, specifies the type of rotation data to use, including different explicit *Euler* orders. Defaults to using the Euler order of the target. See *Rotation Channel Modes*.

Space World Space, Transform Space, Local Space.

Rotational Difference Provides the value of the rotational difference between two objects or bones, in radians.

Distance Provides the value of the distance between two objects or bones.

Value Shows the value of the variable.

Rotation Channel Modes

Rotation Transform Channels support a number of operation modes, including:

Auto Euler Uses the *Euler* order of the target to decompose rotation into channels.

XYZ Euler, … Explicitly specifies the *Euler* rotation order to use.

Quaternion Provides the *Quaternion* representation of the rotation.

Swing and X/Y/Z Twist Decomposes the rotation into two parts: a *Swing* rotation that aims the specified axis in its final direction, followed by a *Twist* rotation around that axis. This is often necessary for driving corrective *Shape Keys* and bones for organic joint rotation.

This decomposition is often produced in rigs by using a helper bone with a *Damped Track Constraint* to extract the swing part, and its child with *Copy Transforms* to extract the twist component.

The channels values for *Swing and Y Twist* are:

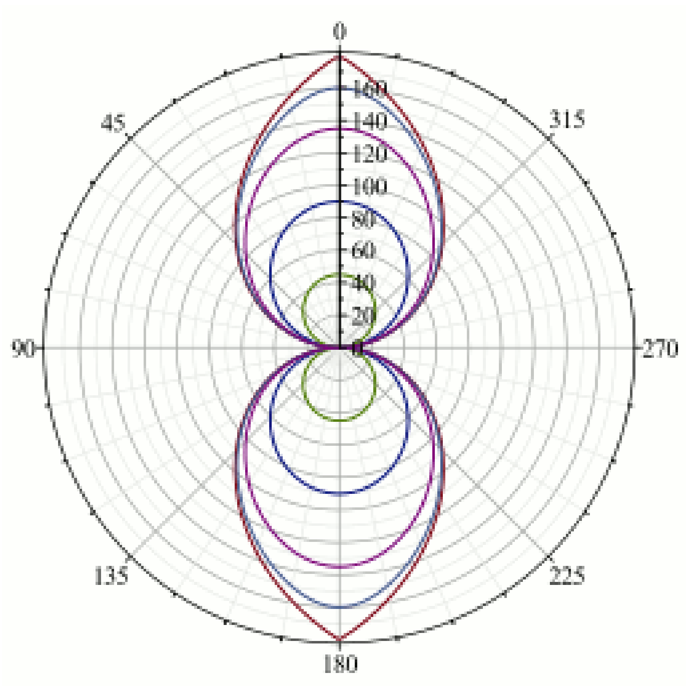


Fig. 1729: Falloff curves for weighted angles.

Y Rotation True angle of the twist rotation.

W Rotation True angle of the swing rotation, independent of its direction.

X Rotation, Z Rotation Weighted angles that represent the amount of swing around the X/Z axis.

The magnitude of the angle equals *W Rotation* when the rotation is purely around that axis, and fades out to zero as the direction changes toward the other axis, following the falloff curves from the graph on the right.

Mathematically, the swing angles are computed from quaternion components, using $2\arccos(w)$ for *W* and $2\arcsin(x)$ etc. for the others. The component of the swing rotation that corresponds to the twist axis is always 0, and is replaced by the twist angle.

Expressions

Expression A text field where you can enter an arbitrary Python expression that refers to *Driver Variables* by their names.

The expression has access to a set of standard constants and math functions from `math`, `bl_math` and other modules, provided in the *Driver Namespace*. For an example of adding a custom function to the namespace, see the *driver namespace example*.

For performance reasons it is best to use the *Simple Expressions* subset as much as possible.

Use Self If this option is enabled, the variable `self` can be used for drivers to reference their own data. Useful for objects and bones to avoid having creating a *Driver Variable* pointing to itself.

Example: `self.location.x` applied to the Y rotation property of the same object will make the object tumble when moving.

Note that dependencies for properties accessed via `self` may not be fully tracked.

Simple Expressions

Blender can evaluate a useful subset of Python driver expressions directly, which significantly improves performance, especially on multi-core systems. To take advantage of this, the driver expression must only use the following features:

Variable Names Use only ASCII characters.

Literals Floating point and decimal integer.

Globals `frame`

Constants `pi`, `True`, `False`

Operators `+`, `-`, `*`, `/`, `==`, `!=`, `<`, `<=`, `>`, `>=`, `and`, `or`, `not`, conditional operator/ ternary if

Standard Functions `min`, `max`, `radians`, `degrees`, `abs`, `fabs`, `floor`, `ceil`, `trunc`, `round`, `int`, `sin`, `cos`, `tan`, `asin`, `acos`, `atan`, `atan2`, `exp`, `log`, `sqrt`, `pow`, `fmod`

Blender Provided Functions `lerp`, `clamp`, `smoothstep`

Simple expressions are evaluated even when Python script execution is disabled.

When an expression outside of this subset is used, Blender displays a “Slow Python expression” warning. However, as long as the majority of drivers use simple expressions, using a complex expression in select few is OK.

See also:

- *Extending Blender with Python*.
- Python and its documentation.
- functions.wolfram.com.

Workflow & Examples

Simple Drivers can be configured from the pop-over that appears when adding a new Driver.

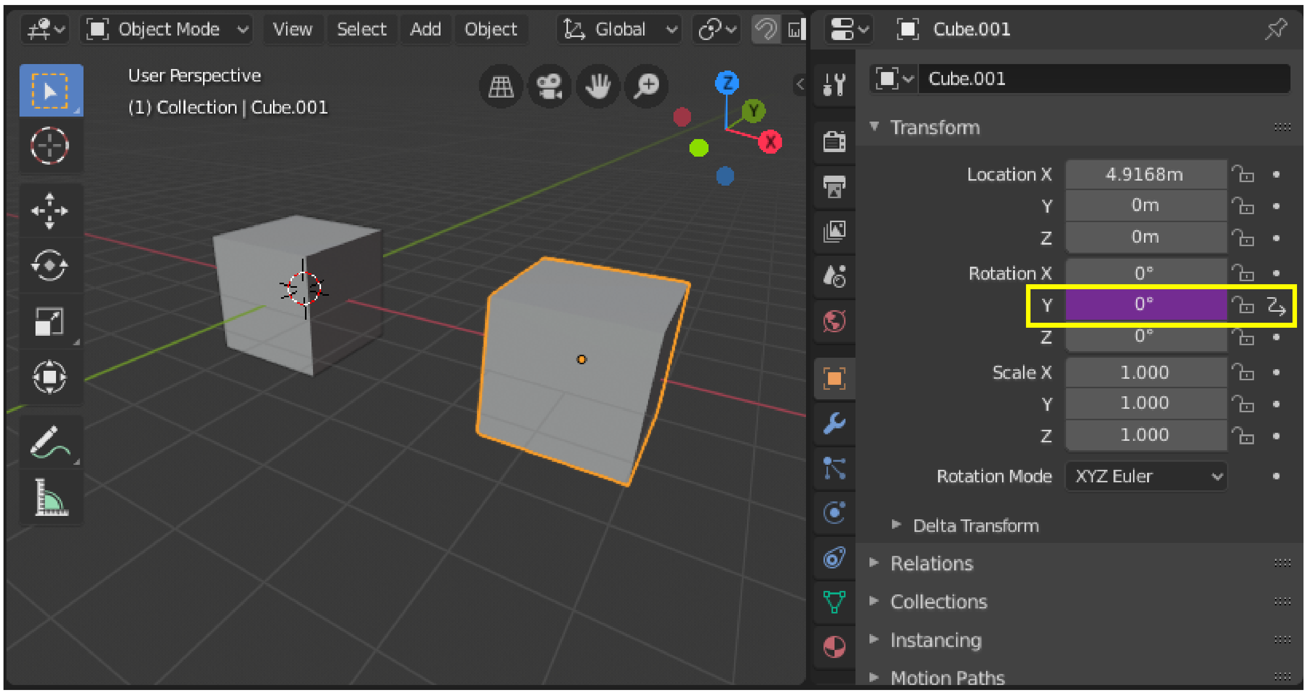
When adding multiple Drivers or for more advanced configurations, it is useful to have open the *Drivers Editor*.

Transform Driver

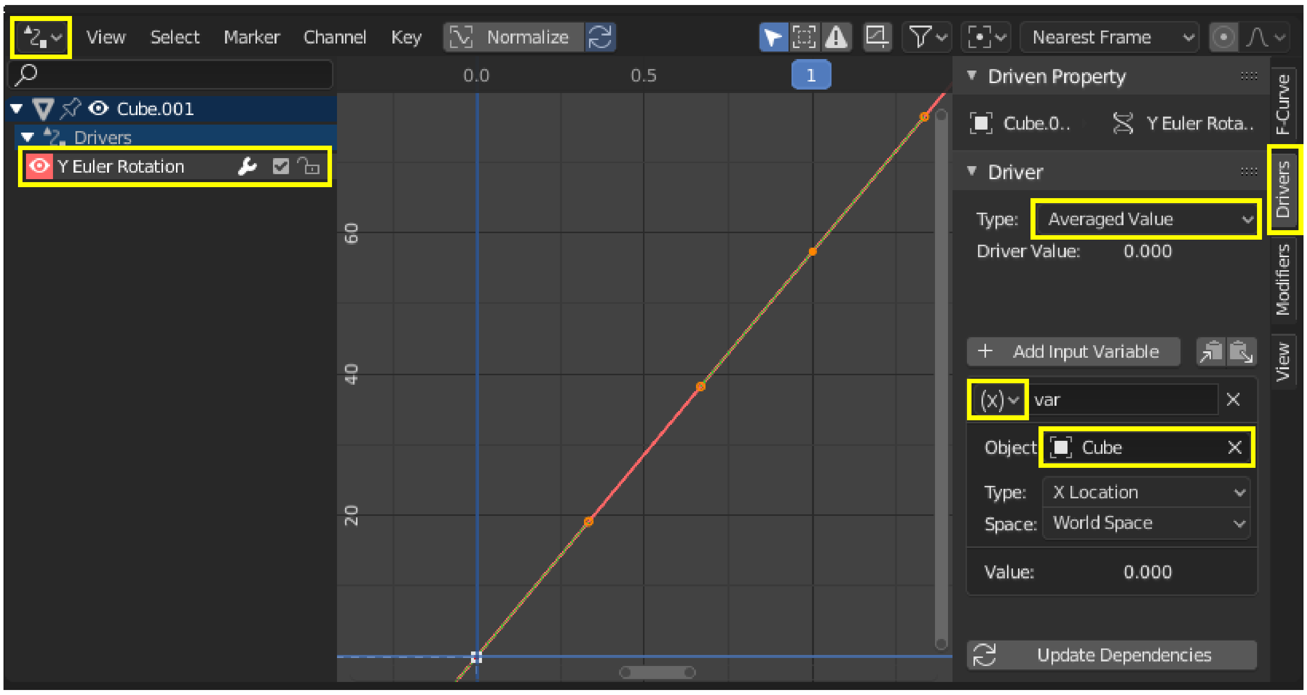
Control a property with an object's transform. In this example, the Y rotation of Object 2 will be driven by the X position of Object 1.

Starting from a simple setup with two objects:

1. Add a Driver to the *Rotation Y* property of the second object via the context menu or with `Ctrl-D`.



2. Open the *Drivers Editor* and select the *Y Euler Rotation* property in the channels region.
3. Open the Sidebar region and select the *Drivers* tab.
4. Configure the driver to be the *Averaged Value* of a *Transform Channel* of the first object.



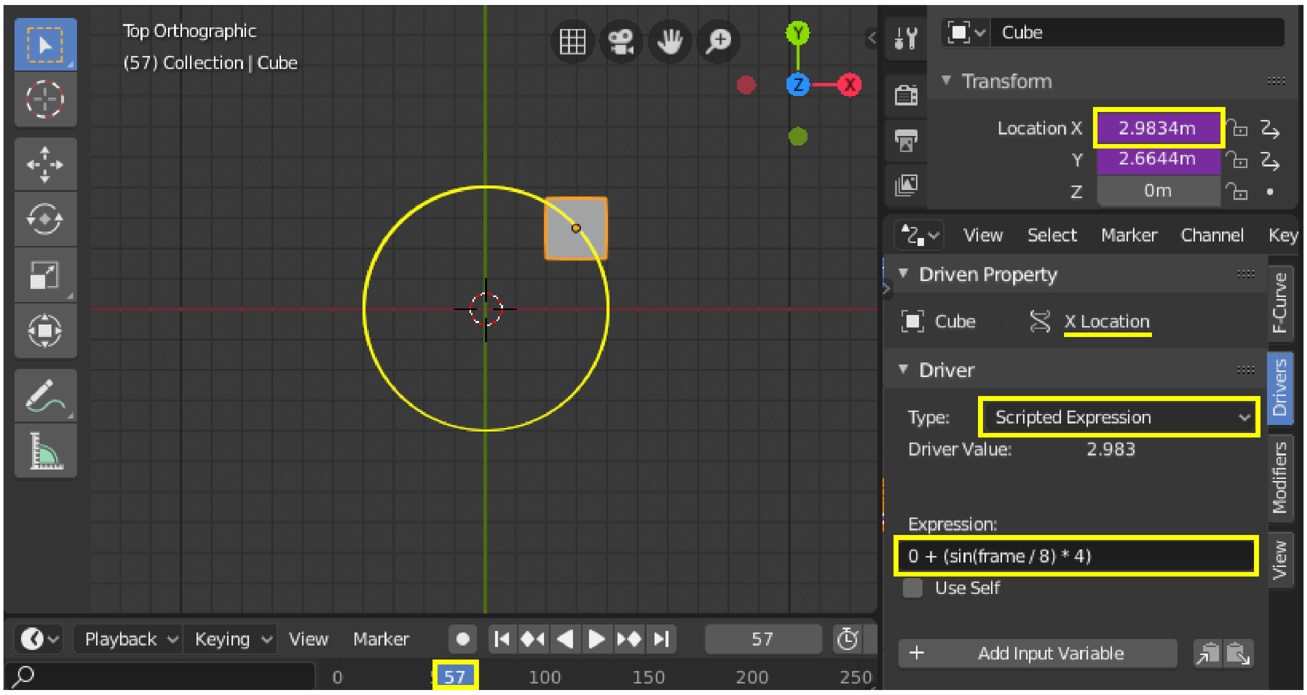
5. Experiment with moving the first object and notice how it affects the Y rotation of the second object.

Scripted Expression - Orbit a Point

Orbit an object's position around a point with a custom *Scripted Expression*. The object's position will change when scrubbing the timeline.

Using trigonometry, circular motion can be defined in 2D using the sinus and cosine functions. (See Unit Circle.)

In this example, the current frame is used as the variable that induces the motion. `frame` is a *Simple Expression* that corresponds to `bpy.context.scene.frame_current`.



Location X 2.9834m
Y 2.6644m
Z 0m

Cube — X Location

Type: Scripted Expression
Driver Value: 2.983
Expression:
0 + (sin(frame / 8) * 4)
Use Self

1. Add a driver to the X Location property.
   1. Set the *Driver Type* to *Scripted Expression*.
   2. Add the expression `0 + (sin(frame / 8) * 4)`, where:
      - `frame/8` : is the current frame of the animation, divided by 8 to slow the orbit down.
      - `(sin( )*4)` : multiplies the result of `sin(frame/8)` by 4 for a bigger circle.
      - `0 +` : is used to control the offset to the orbit center point.
2. Add a driver to the Y Location property with the expression `0 + (cos(frame / 8) * 4)`.
3. Scrub the timeline to see the effect. Experiment with the variables to control the size and center of the orbit.

Custom Function - Square Value

Create a custom function to get the square of a value (i.e. *value*^{2}). Adding the function to the *Driver Namespace* allows it to be used from driver expressions.

The *Driver Namespace* has a list of built-in functions for use in driver expressions, as well as constants such as π and e. These can be inspected via the Python Console:

```
>>> bpy.app.driver_namespace[' <tab>
                              acos']
                              acosh']
                              asin']
                              asinh']
                              atan']
                              ...
```

To add a new function to the *Driver Namespace*, the function itself needs to be implemented and then added to the `bpy.app.driver_namespace`.

1. Add the following to the Text Editor inside Blender and press *Run Script*.

   ```
   import bpy

   def square(val):
      """Returns the square of the given value"""
      return val * val

   # Add function to driver_namespace.
   bpy.app.driver_namespace['square'] = square
   ```

2. Add a driver with a *Scripted Expression* such as `square(frame)`.
3. Observe the effect when scrubbing the timeline.

There are more custom function examples available in Blender's Text Editor *Templates* → *Python* → *Driver Functions*.

Since *Simple Expressions* cannot access custom functions, using them only makes sense for complex computations.

Shape Key Drivers

Improved Mesh Deformation

Fix intersection problems that happen when using armatures and weight painting, especially at joints. Shape keys can also be used to tweak and refine a rig, for example to suggest muscle formations. In this example, a shape key is used to improve the deformation at the elbow of a rudimentary arm.

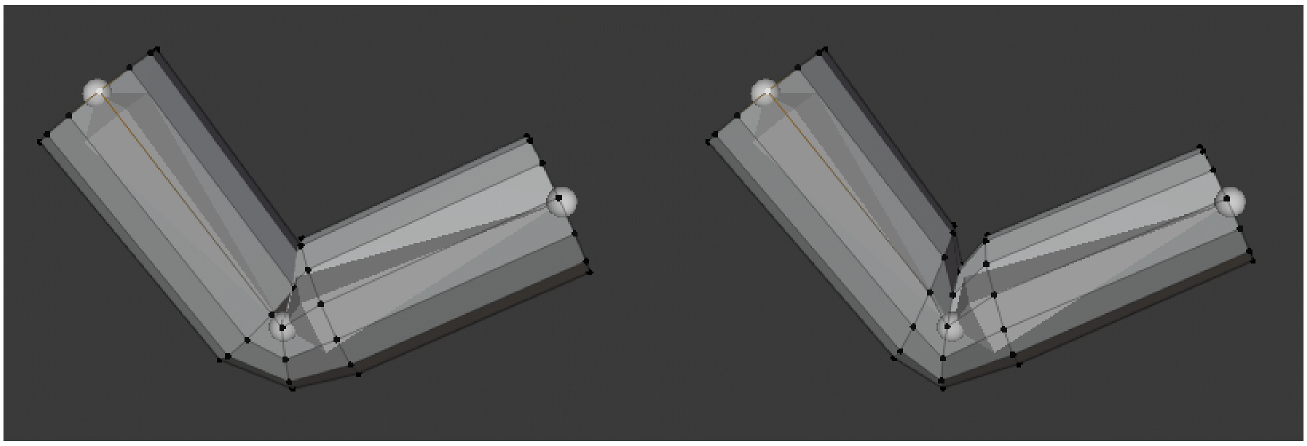


Fig. 1730: Left: Skeletal mesh deformation without correction. Right: Corrective shape key applied

Setup

1. Add a mesh (in this example, a cylinder with loop cuts).
2. Add an armature with a chain of bones.
3. Skin the mesh to the armature using weight painting.

(Note: to parent the mesh to the armature: select the mesh first, then the armature and use `Ctrl-P` to parent with auto weights.)

Experiment with posing the armature and observe the deformation at the joint. To fix intersection problems or angles that look unsatisfactory, you can associate a *Shape Key* with a pose.

Shape Key

1. Pose the armature such that the problems are visible. Be sure to cover the extreme poses that you want to support for the rig.
2. With the mesh selected, add a new *Shape Key* in addition to the *Basis* key. *Properties → Mesh tab → Shape Keys*
3. In order to author the shape key on top of the armature deformation, enable both *Edit Mode Display* and *Cage Editing* in the Armature modifier. *Properties → Modifiers tab → Armature Modifier → Header*
4. Enter Edit Mode and select the new shape key in the properties panel. Adjust the vertices as desired. Select the *Basis* key to toggle between the original mesh and your edits. (Note: be careful to apply edits only to your shape and not to the original mesh or other existing keys.)

Once you are satisfied with how the deformation looks for the problematic pose, you'll need to configure a driver to activate the shape smoothly when entering that position.

Driver

1. Add a driver to the *Value* of the shape key you've created.
2. Open the Drivers Editor and select the driver.

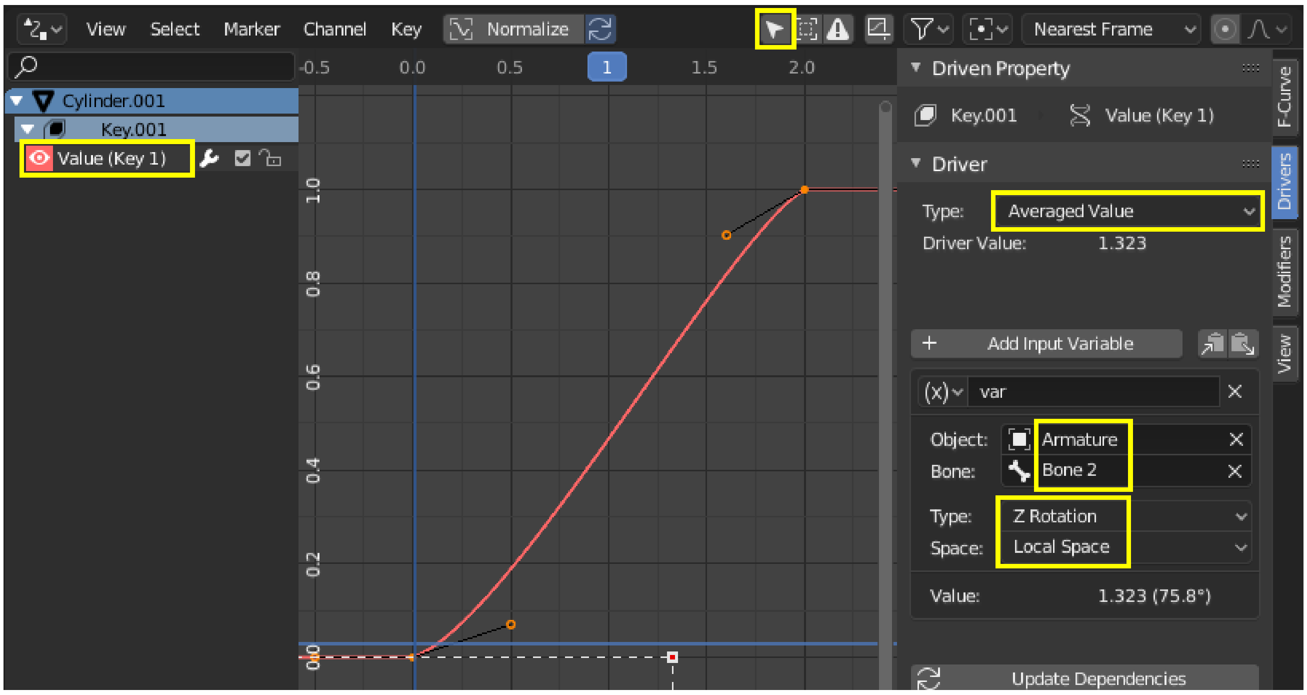
Method 1 – Direct mapping to a bone rotation value A simple way to configure the driver is with a direct correspondence of the value of a bone's rotation channel to the shape key activation *Value*. This method has the disadvantage of relying on a single channel of a bone's rotation which might be insufficient to precisely express the condition under which the shape key should be activated.

1. In the Drivers tab, select the *Averaged Value* of the rotation of the bone you're posing.

   Understand the rotation axis that you're interested in by enabling axes display in the armature or by observing the bone's transform values in the Properties.

   Select the rotation channel and set it to local, meaning, the bone's rotation value relative to its parent bone.
2. Manually set points in the driver curve by selecting a handle and dragging it or inserting values in the *F-Curve* tab. The Y axis represents the shape key *Value*, which should go from 0.0 to 1.0. The X axis is usually the frame, but for this driver it represents the rotation value in radians. You can have more than two points in the curve and tweak the transitions with the handles in the curve view (`G` to move).
3. To verify that the driver behaves correctly, deselect the option to only show drivers for selected objects. This way, you can pose the armature and keep an eye on the driver.

Method 2 – Rotational difference to a target bone This method requires an additional *target* or *corrective* bone, but it better expresses the spatial condition in 3D space of the bone that is causing the problem.

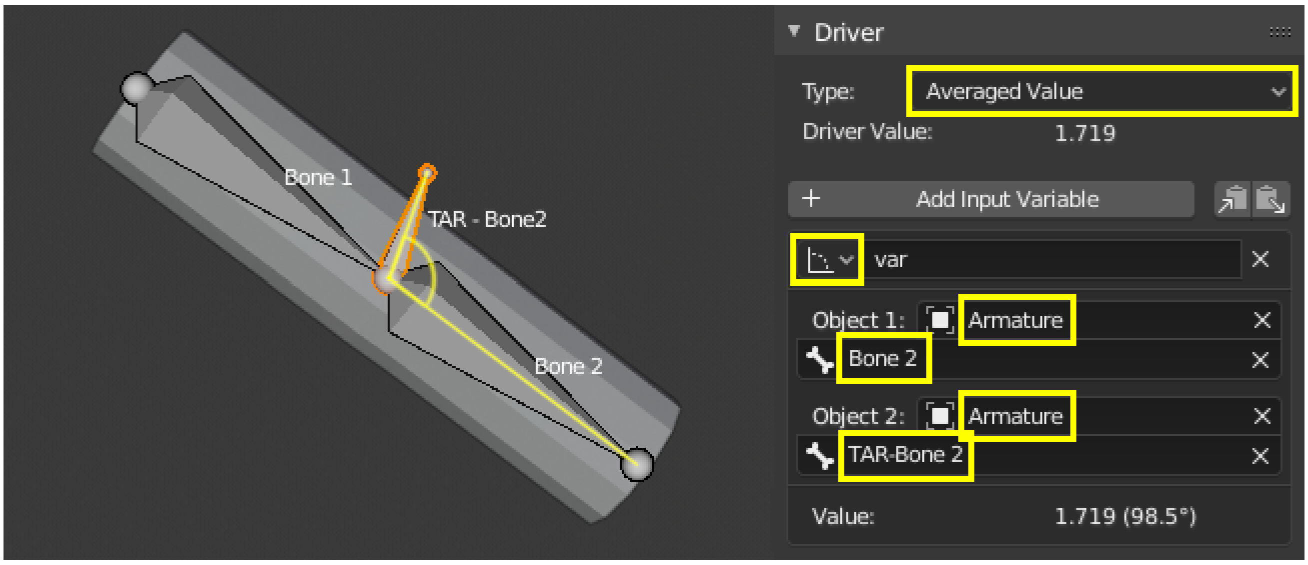


Key.001 Value (Key 1)

Type: Averaged Value
Driver Value: 1.323

var
Object: Armature
Bone: Bone 2
Type: Z Rotation
Space: Local Space
Value: 1.323 (75.8°)

1. In armature Edit Mode, add a new bone extruded from Bone 1, in the position at which Bone 2 should have the shape key active. This type of bones usually follow a naming convention such as "TAR-" (target) or "COR-" (corrective).
2. In the Drivers tab, select the *Averaged Value* of the rotational difference between the bone you're rotating and the target bone. A rotational difference is the minimum angle between two objects in World Space. It is therefore important that the bones have the same root, so that the only thing affecting the angle between the bones is the rotation of one of them. When the deformation bone (Bone 2) reaches the target rotation (TAR-Bone 2) the rotational difference will be 0°.



Type: Averaged Value
Driver Value: 1.719

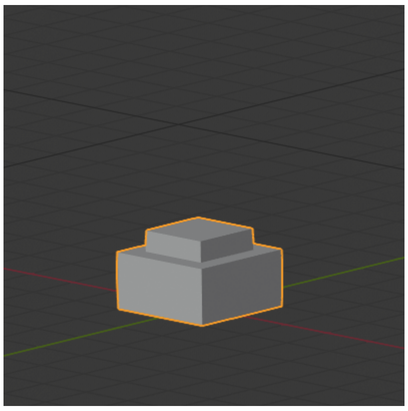
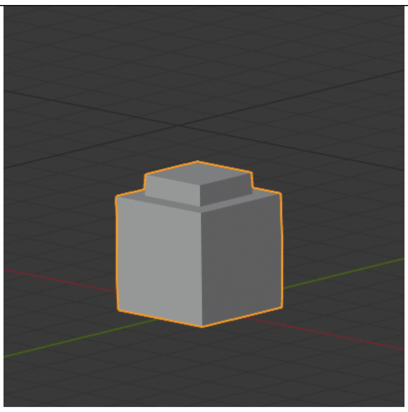
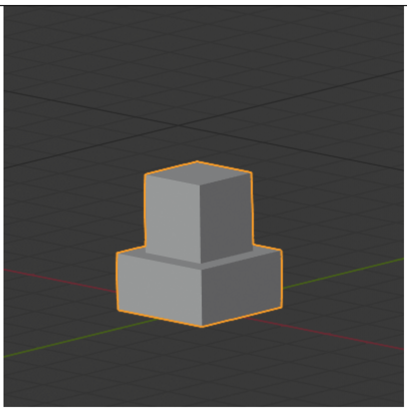
var
Object 1: Armature
Bone 2
Object 2: Armature
TAR-Bone 2
Value: 1.719 (98.5°)

3. Manually adjust the driver curve handles so that the shape key *Value* (Y axis) is 1.0 when the rotational difference (X axis) is 0°. The *Value* should be 0.0 when the arm is extended, at which point the rotational difference should be around 90° or more (in radians).
4. See the steps in Method 1 on how to adjust the curve handles and confirm that the functionality is working. Pose the armature to verify that the ranges are correct.

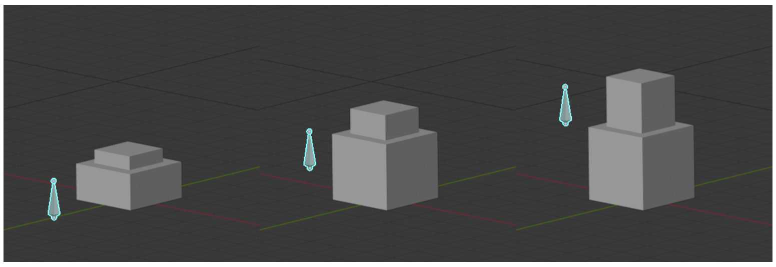
Chained Relative Shape Keys

Activate different shape keys in succession. In this example, moving a single bone will activate first *Key 1* and then *Key 2*. See also *relative shape keys mix additively*.

Shape Keys Add two shape keys to a mesh, besides the *Basis*.

 Fig. 1731: Basis.	 Fig. 1732: Key 1: top faces moved up by 1 m.	 Fig. 1733: Key 2: inner top moved up by 1 m.

Drivers Add an armature with a single bone to control the shape keys. The goal is to activate the keys in succession as this bone moves up.



As shown in the picture above, when the bone is halfway up, both *Key 1* and *Key 2* have an influence. It is a matter of preference if *Key 1* should be at its maximum *Value* before *Key 2* starts to become active, or how much they should overlap. This example shows a seamless blend.

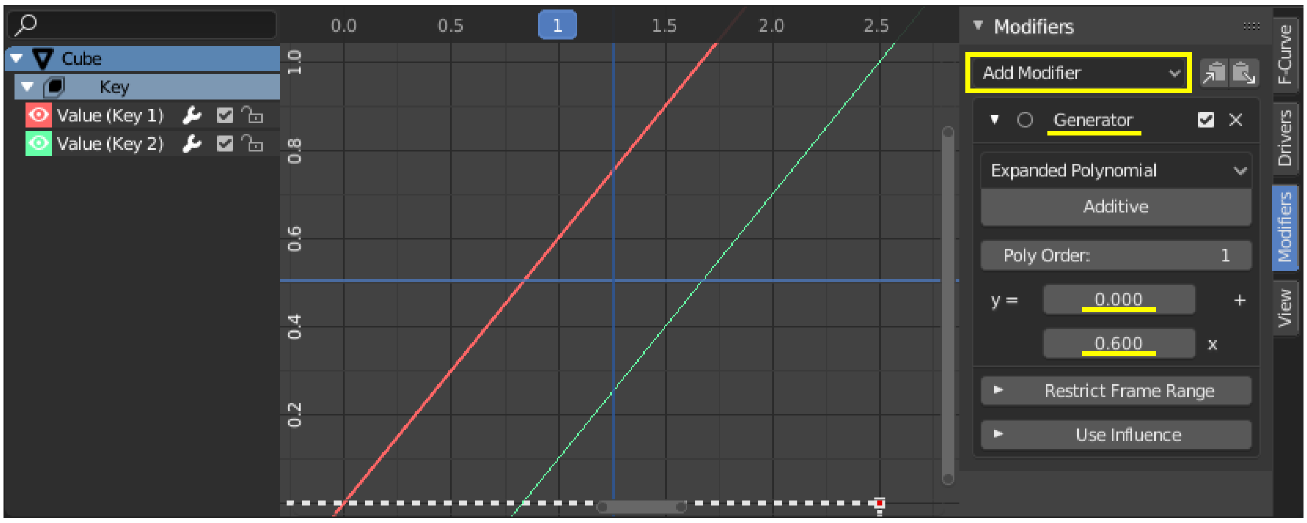
For a seamless blend where there is overlap, *Key 1* should have a *Value* of 0.0 when the bone is at the bottom and increase linearly to 1.0 until the bone is past the midpoint height. *Key 2* should have a value of 0.0 before the midpoint height and then increase at the same rate than *Key 1* until reaching *Value* 1.0 when the bone is at maximum height.

1. Add a driver to the *Value* of *Key 1* and *Key 2*. In the *Drivers* tab, configure both drivers to be the *Averaged Value* of a variable with the bone's Z location.
2. Determine the range of the bone's motion in the World Z axis by moving it up so that it is aligned with the top of the mesh when both keys are active. Here we will use [0.0 , 2.5].
3. Configure the driver functions so that the *Value* of the shape keys (Y axis) is as desired for the bone's height (X axis).

   The driver functions should be linear, therefore, they can be defined analytically with a function of type $y = a + bx$, where a is an offset in y and b is the slope.

   1. In the *Modifiers* tab, add a *Generator* of type *Extended Polynomial* for both drivers.
   2. Play with the values of a and b so that the curves go from [0.0 , 1.0] in the Y axis and from [0.0 , 2.5] in the X axis. The curves should overlap in the mid area of the X axis and they should have the same slope (b).

Possible values are *Key 1*: $y = 0.0 + 0.6x$ and *Key 2*: $y = -0.5 + 0.6x$.



Note that the functions go outside the range [0.0 , 1.0] for the shape keys' *Value*, but that has no effect because *Value* is clamped in a *Range* in the *Shape Keys* panel.

Troubleshooting

Some common problems people may run into when using drivers.

Scripted Expression

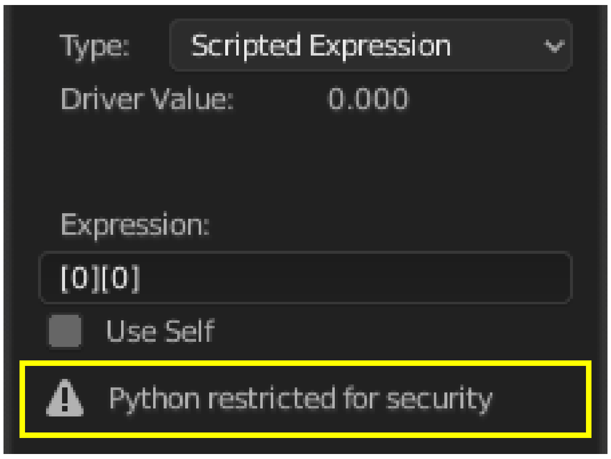

Type: Scripted Expression
Driver Value: 0.000
Expression:
[0][0]
Use Self
Python restricted for security

Fig. 1734: A security warning in the Drivers panel.

By default Blender will restrict execution of Python scripts.

If using a *Scripted Expression* Driver Type that doesn't follow the *Simple Expressions* subset, you will have to open the file as *Trusted Source*, or set *Auto Run Python Scripts* in *Preferences* → *Save & Load* → *Blender Files*.

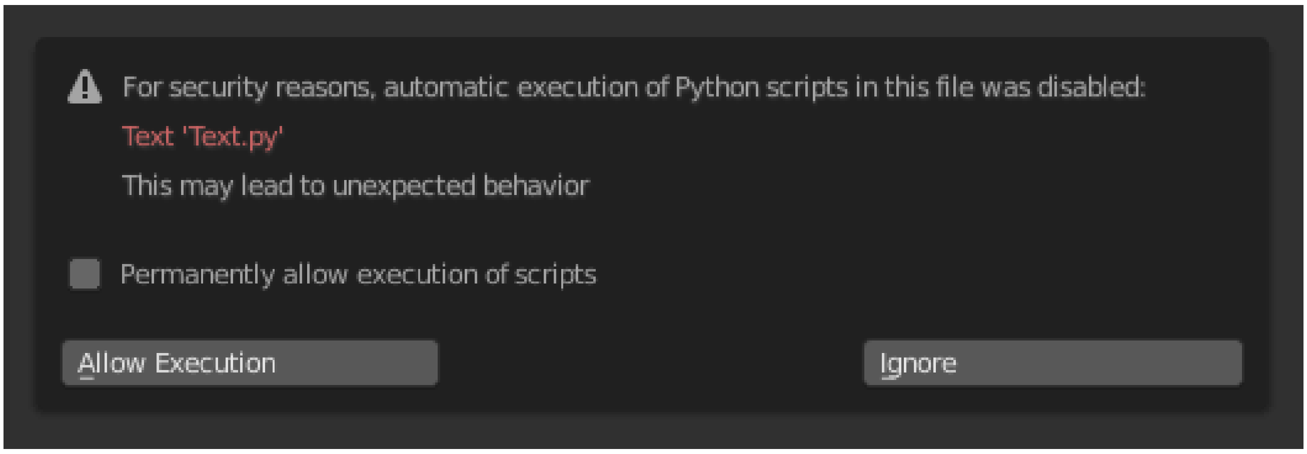

For security reasons, automatic execution of Python scripts in this file was disabled:
Text 'Text.py'
This may lead to unexpected behavior

Fig. 1735: An Auto-run warning in the Info editor's header.

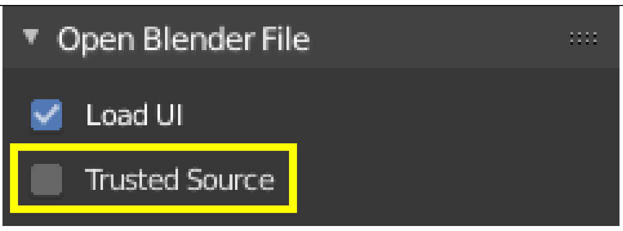


Fig. 1736: The Trusted Source checkbox in the File Browser.

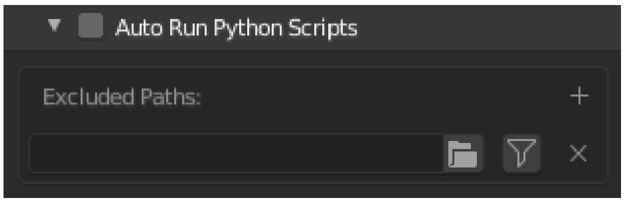


Fig. 1737: The Auto Run Python Scripts checkbox in the Preferences.

Rotational Properties are Radians

Parts of the User Interface may use different units of measurements for angles, rotation. In the Graph Editor, while working with Drivers, all angles are Radians.

2.8.8 Markers

Markers are used to denote frames with key points or significant events within an animation. E.g. it could be that a character's animation starts, the camera changes position, or a door opens. Markers can be given names to make them more meaningful at a quick glance. They are available in many of Blender's editors.

Note: Unlike keyframes, markers are always placed at a whole frame number, you cannot set a marker at frame 2.5.

Markers can be created and edited in the following editors:

- *Graph Editor*
- *Dope Sheet*
- *NLA Editor*
- *Video Sequence Editor*
- *Timeline*

Note: A marker created in one of these editors will also appear in all others that support them.

Types

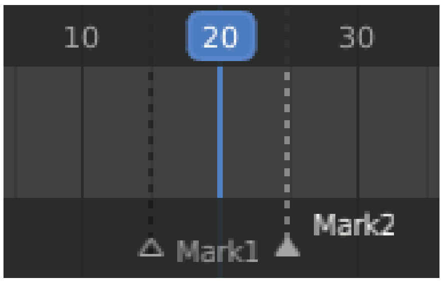
Besides standard markers, *pose markers* are another type of markers, which are specific to armatures and shape keys. They are used to denote poses in the Action Editor mode and Shape Keys Editor of Dope Sheet.

Visualization

In the supported editors, if at least one is created, markers are visualized in a separate row at their bottom. This area can be disabled per editor via the *View* → *Show Markers* menu option.

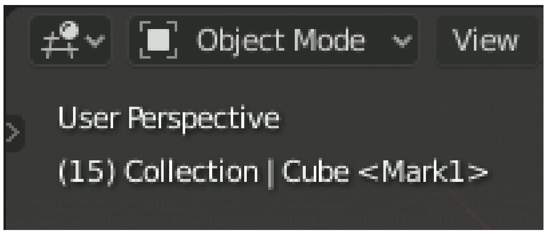
Note: While the markers area is disabled, markers operators are not available in that editor, and in the header the *Marker* menu is hidden.

Standard



Regular markers are shown as small white triangles, empty if unselected or filled if selected, and with a dashed line that covers the editor height at the corresponding frame. If they have a name, this is shown to their right in white.

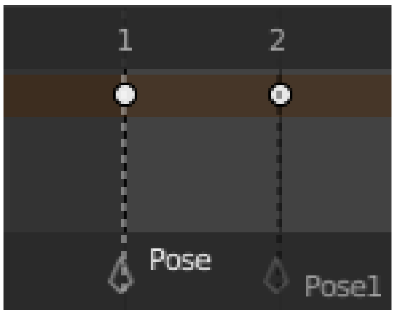
3D Viewport



User Perspective
(15) Collection | Cube <Mark1>

The 3D Viewport does not allow you to create, edit or remove markers, but it shows their name in the Object Info in the upper left corner, when on their frame.

Pose Markers



Pose markers show a diamond-shaped icon in the Dope Sheet. In the NLA editor pose markers are shown as a red dashed line inside the relative action strip.

Add Marker

Reference

Mode All modes

Menu *Marker → Add Marker*

Hotkey `M`

The simplest way to add a marker is to move to the frame where you would like it to appear, and press `M`.

Hint: Markers can also be added during playback.

Pose Markers

If *Show Pose Markers* is checked, a pose marker and a new pose in the *Pose Library* are added.

Selecting

Reference

Mode All modes

Hotkey `LMB`

Click `LMB` on the marker's triangle to select it. Use `Shift-LMB` to select multiple markers.

In the Graph Editor, Dope Sheet, NLA Editor, Timeline, and Video Sequence Editor, you can also select all markers with `A` while hovering the mouse over the marker row, and apply selection tools on them like Box Select, etc. (as usual, `LMB` to select, `RMB` to deselect). The corresponding options are found in the Select menu of these editors.

Editing

Duplicate Marker

Reference

Mode All modes

Menu *Marker → Duplicate Marker*

Hotkey `Shift-D`

You can duplicate the selected markers by pressing `Shift-D`. Once duplicated, the new ones are automatically placed in select mode, so you can move them to the desired location.

Note: Note that unlike most other duplications in Blender, the names of the duplicated markers are not altered at all (no `.001` numeric counter append).

Duplicate Marker to Scene

Reference

Mode All modes

Menu *Marker → Duplicate Marker to Scene…*

Duplicates the selected markers into another scene.

Delete Marker

Reference

Mode All modes

Menu *Marker → Delete Marker*

Hotkey `X`

To delete the selected markers simply press `X`, and confirm the pop-up message with `LMB`.

Rename Marker

Reference

Mode All modes

Menu *Marker → Rename Marker*

Hotkey `Ctrl-M`

Having dozens of markers scattered throughout your scene's time will not help you much unless you know what they stand for. You can name a marker by selecting it, pressing `Ctrl-M`, typing the name, and pressing the OK button.

Move Marker

Reference

Mode All modes

Menu *Marker → Move Marker*

Hotkey `G`

Once you have one or more markers selected, press `G`, while hovering with the mouse over the marker bar, to move them, and confirm the move with `LMB` or `Return` (as usual, cancel the move with `RMB`, or `Esc`). Or drag them with the `LMB`.

By default, you move the markers in one-frame steps, but if you hold `Ctrl`, the markers will move in steps corresponding to 1 second (according to the scene's *FPS*).

Show Pose Markers

Reference

Editor Dope Sheet

Mode Action Editor or Shape Keys Editor mode

Menu *Marker → Show Pose Markers*

Shows markers belonging to the active action instead of scene markers.

Make Markers Local

Reference

Mode All modes

Menu *Marker → Make Markers Local*

It is possible to convert standard markers into pose markers with *Marker → Make Markers Local*. Note that the original marker will be gone. If you want to keep it, make a duplicate before you convert.

Jump to Next/Previous Marker

Reference

Mode All modes

Menu *Marker → Jump to Next/Previous Marker*

Moves the playhead to the next/previous marker relative to the current playhead position.

Bind Camera to Markers

Reference

Editor Timeline

Menu *Marker → Bind Camera to Markers*

Hotkey `Ctrl-B`

Bind Camera to Markers is a special operator only available in the *Timeline*. The operator allows markers to be used to set the active object as the active camera.

To use this operator, select the object to become the active camera and select a marker to bind the active camera to. If no marker is selected when the operator is applied, a marker will be added. When an object is bound to a marker, the marker will be renamed to the name of the active object. These markers also have a camera icon next to the left of the name to easily distinguish them from other informative markers.

These markers can be moved to change the frame at which the active camera is changed to the object the marker is bound to.

2.8.9 Shape Keys

Introduction

Shape keys are used to deform objects into new shapes for animation. In other terminology, shape keys may be called "morph targets" or "blend shapes".

The most popular use cases for shape keys are in character facial animation and in tweaking and refining a skeletal rig. They are particularly useful for modeling organic soft parts and muscles where there is a need for more control over the resulting shape than what can be achieved with combination of rotation and scale.

Shape keys can be applied on object types with vertices like mesh, curve, surface and lattice.

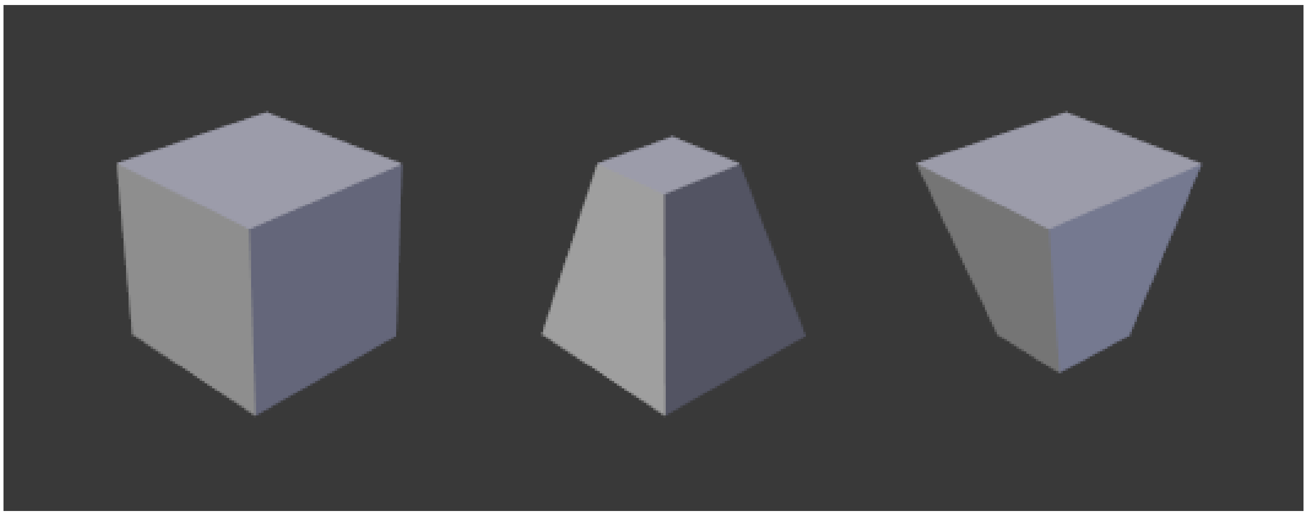


Fig. 1738: Example of a mesh with different shape keys applied.

Workflow

Shape keys are authored in the *Shape Keys panel* which is accessed in the Object Data tab of the Properties (e.g. the Mesh tab for mesh objects).

A shape key is modified by first selecting a shape key in the panel, and then moving the object's vertices to a new position in the 3D Viewport.

The panel has controls for affecting the current *Value* (influence, weight) of a shape. It is possible to see a shape in isolation or how it combines with others.

Adding and Removing Vertices

It is not possible to add or remove vertices in a shape key. The number of vertices and how they connect is specified by the mesh, curve, surface or lattice. A shape key merely records a position for each vertex and therefore shapes always contain all the object's vertices.

When adding a vertex, all shape keys will record it with the position in which it is created. Workflow-wise, adding and deleting vertices after creating shape keys is possible, but it is best to leave the creation of shape keys for when the mesh is finished or its topology is stable.

Adding Shape Keys

When adding a new shape key with the + button next to the list, the new shape will be a copy of the Basis shape, independently of the current result visible in the 3D Viewport.

When adding a new shape key from *Specials* → *New Shape from Mix*, the shape will start of with the vertex configuration that is visible at that moment.

When doing facial animation with relative shape keys, it can be useful to first create a shape key with a complex extreme pose (e.g. anger or surprise), and then break this complex shape into components by applying a temporary vertex group to the complex shape and creating a copy with *New Shape from Mix*. This technique helps reducing conflicts between different shape keys that would otherwise produce a double effect.

Relative or Absolute Shape Keys

A mesh (curve, surface or lattice) has a stack of shape keys. The stack may be of *Relative* or *Absolute* type.

Relative Mainly used for muscles, limb joints, and facial animation.

Each shape is defined relative to the Basis or to another specified shape key.

The resulting effect visible in the 3D Viewport, also called *Mix*, is the cumulative effect of each shape with its current value. Starting with the Basis shape, the result is obtained by **adding** each shape's weighted **relative** offset to its reference key.

Value Represents the weight of the blend between a shape key and its reference key.

A value of 0.0 denotes 100% influence of the reference key and 1.0 of the shape key. Blender can extrapolate the blend between the two shapes above 1.0 and below 0.0.

Basis Basis is the name given to the first (top-most) key in the stack.

The Basis shape represents the state of the object's vertices in their original position. It has no weight value and it is not keyable. This is the default *Reference Key* when creating other shapes.

Absolute Mainly used to deform the objects into different shapes over time.

Each shape defines how the object's shape will be at *Evaluation Time* specified in its *Value*.

The resulting shape, or *Mix*, is the interpolation of the previous and next shape given the current *Evaluation Time*.

Value Represents the *Evaluation Time* at which that shape key will be active.

Basis Basis is the name given to the first (topmost) key in the stack.

The Basis shape represents the state of the object's vertices in their original position.

Shape Keys Panel

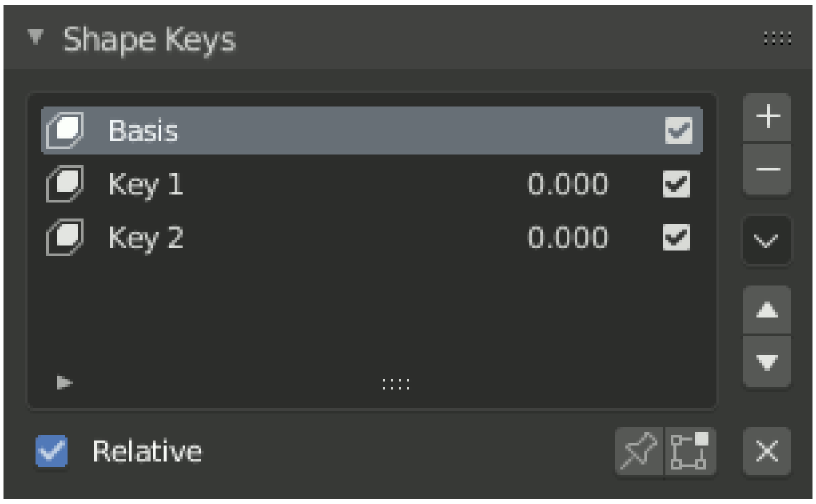


Fig. 1739: Shape Keys panel.

Reference

Editor Properties

Mode All modes

Panel *Object Data → Shape Keys*

The Shape Keys panel is used for authoring shape keys.

Settings

Active Shape Key Index A *List View*.

Value (number) In Relative mode: Value is the current influence of the shape key used for blending between the shape (value=1.0) and its reference key (value=0.0). The reference key is usually the Basis shape. The weight of the blend can be extrapolated above 1.0 and below 0.0.

In Absolute mode: Value is the *Evaluation Time* at which the shape will have maximum influence.

Mute (check mark) If unchecked, the shape key will not be taken into consideration when mixing the shape key stack into the result visible in the 3D Viewport.

Specials

New Shape from Mix Add a new shape key with the current deformed shape of the object. This differs from the + button of the list, as that one always copies the Basis shape independently of the current mix.

Mirror Shape Key If your mesh is symmetrical, in *Object Mode*, you can mirror the shape keys on the X axis. This will not work unless the mesh vertices are perfectly symmetrical. Use the *Mesh → Symmetrize* tool in *Edit Mode*.

Mirror Shape Key (Topology) Same as *Mirror Shape Key* though it detects the mirrored vertices based on the topology of the mesh. The mesh vertices do not have to be perfectly symmetrical for this action to work.

Join as Shapes (Transfer Mix) Transfer the current resulting shape from a different object.

Select the object to copy, then the object to copy into. Use this action and a new shape key will be added to the active object with the current mix of the first object.

Transfer Shape Key Transfer the active shape key from a different object regardless of its current influence.

Select the object to copy, then the object to copy into. Use this action and a new shape key will be added to the active object with the active shape of the first object.

Relative Set the shape keys to *Relative* or *Absolute*. See *Relative or Absolute Shape Keys*.

Shape Key Lock (pin icon) Show the active shape in the 3D Viewport without blending. *Shape Key Lock* gets automatically enabled while the object is in *Edit Mode*.

Shape Key Edit Mode (edit mode icon) If enabled, when entering *Edit Mode* the active shape key will **not** take maximum influence as is default. Instead, the current blend of shape keys will be visible and can be edited from that state.

Relative Shape Keys

See *Relative or Absolute Shape Keys*.

With absolute shape keys, the value shown for each shape in the list represents the current weight or influence of that shape in the current *Mix*.

Clear Shape Keys X Set all influence values, or weights, to zero. Useful to quickly guarantee that the result shown in the 3D Viewport is not affected by shapes.

Value The weight of the blend between the shape key and its reference key (usually the Basis shape).

A value of 0.0 denotes 100% influence of the reference key and 1.0 of the shape key.

Range Minimum and maximum range for the influence value of the active shape key. Blender can extrapolate results when the *Value* goes lower than 0.0 or above 1.0.

Vertex Group Limit the active shape key deformation to a vertex group. Useful to break down a complex shape into components by assigning temporary vertex groups to the complex shape and copying the result into new simpler shapes.

Relative To Select the shape key to deform from. This is called the *Reference Key* for that shape.

Absolute Shape Keys

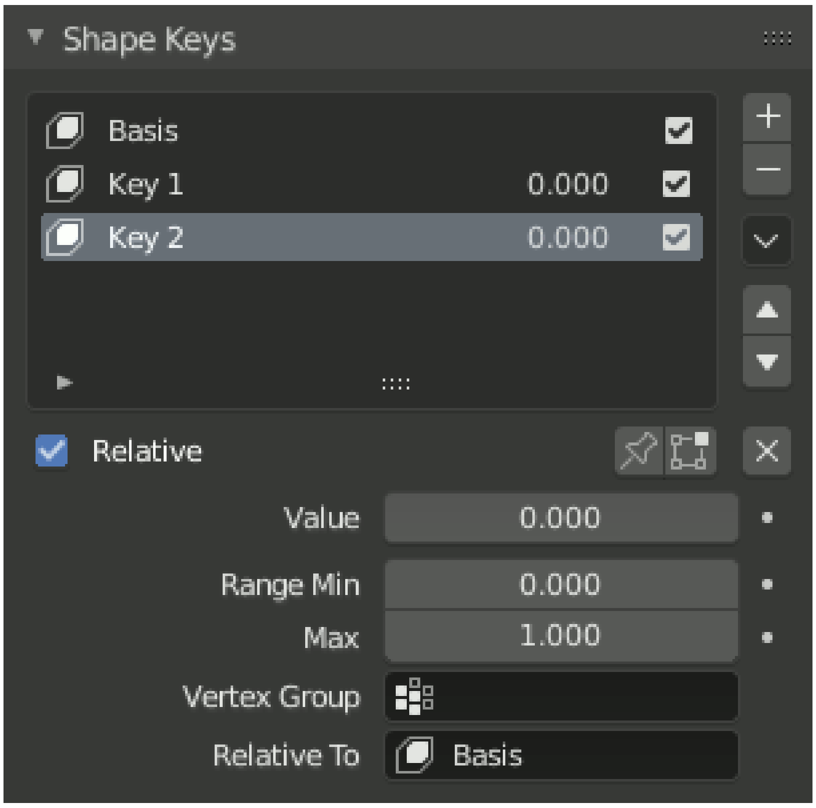
See *Relative or Absolute Shape Keys*.

With absolute shape keys, the value shown for each shape in the list represents the *Evaluation Time* at which that shape key will be active.

Re-Time Shape Keys (clock icon) Absolute shape keys are timed, by order in the list, at a constant interval. This button resets the timing for the keys. Useful if keys were removed or re-ordered.

Interpolation Controls the interpolation between shape keys.

Linear, Cardinal, Catmull-Rom, B-Spline


Shape Keys

Basis		✓
Key 1	0.000	✓
Key 2	0.000	✓

✓ Relative

Value 0.000

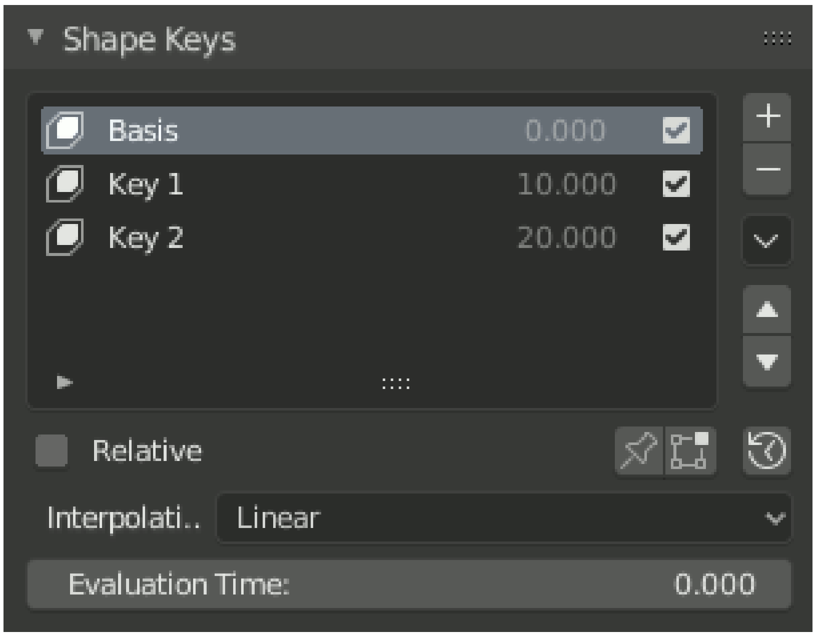
Range Min 0.000

Max 1.000

Vertex Group

Relative To Basis

Fig. 1740: Relative Shape Keys options.


Shape Keys

Basis	0.000	✓
Key 1	10.000	✓
Key 2	20.000	✓

Relative

Interpolati.. Linear

Evaluation Time: 0.000

Fig. 1741: Absolute Shape Keys options.

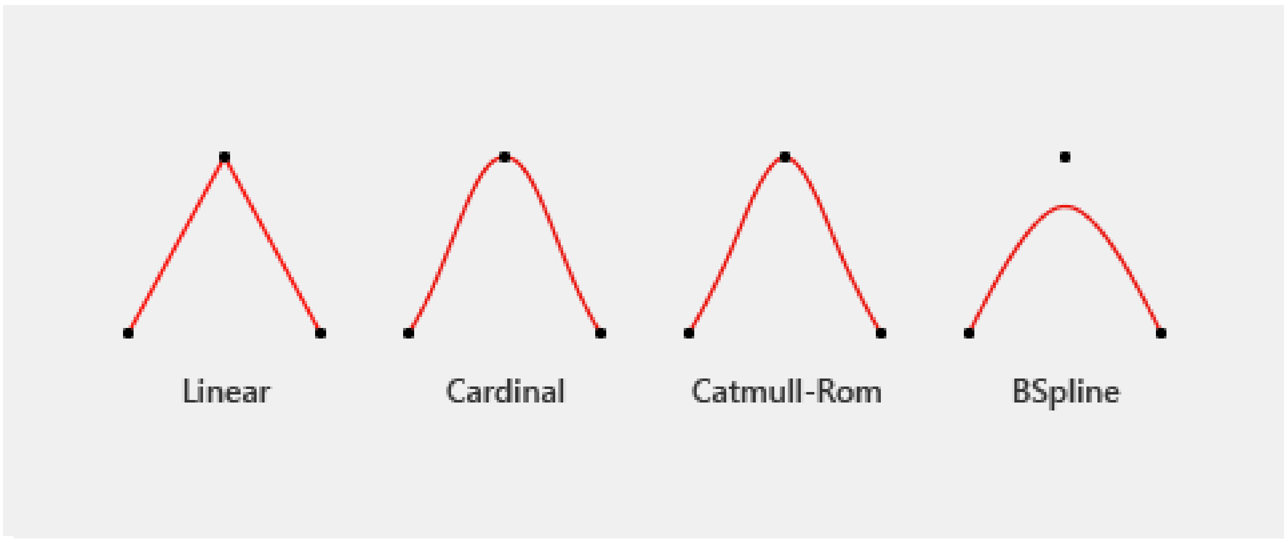


Fig. 1742: Different types of interpolation.
The red line represents interpolated values between keys (black dots).

Evaluation Time Controls the shape key influence. Scrub to see the effect of the current configuration. Typically, this property is keyed for animation or rigged with a driver.

Workflow

Relative Shape Keys

1. In *Object Mode*, add a new shape key via the *Shape Key* panel with the + button.
2. "Basis" is the rest shape. "Key 1", "Key 2", etc. will be the new shapes.
3. Switch to *Edit Mode*, select "Key 1" in the *Shape Key* panel.
4. Deform mesh as you want (do not remove or add vertices).
5. Select "Key 2", the mesh will be changed to the rest shape.
6. Transform "Key 2" and keep going for other shape keys.
7. Switch back to *Object Mode*.
8. Set the *Value* for "Key 1", "Key 2", etc. to see the transformation between the shape keys.

In the figure below, from left to right shows: "Basis", "Key 1", "Key 2" and mix ("Key 1" `1.0` and "Key 2" `0.8`) shape keys in Object Mode.

For more practical examples, see *how to combine shape keys and drivers*.

Absolute Shape Keys

1. Add sequence of shape keys as described above for relative shape keys.
2. Uncheck the *Relative* checkbox.
3. Click the *Reset Timing* button.
4. Switch to *Object Mode*.
5. Drag *Evaluation Time* to see how the shapes succeed one to the next.

By adding a *driver* or setting *keyframes* to *Evaluation Time* you can create an animation.

See also:

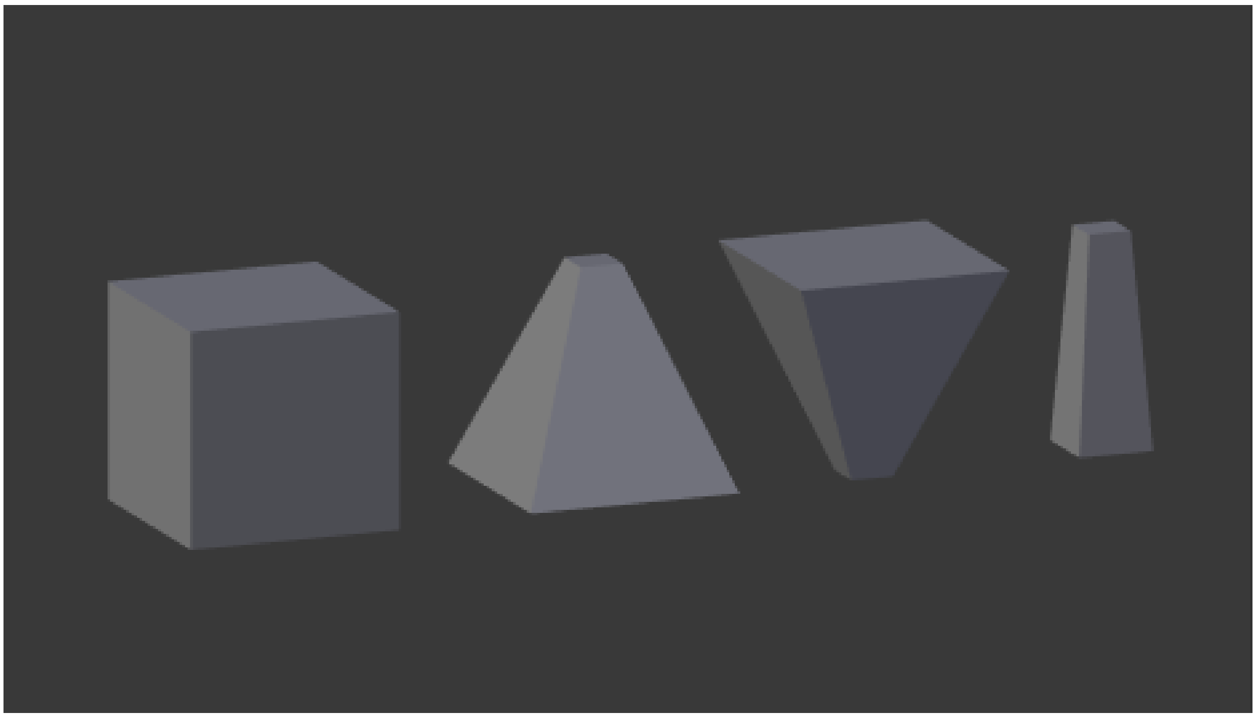
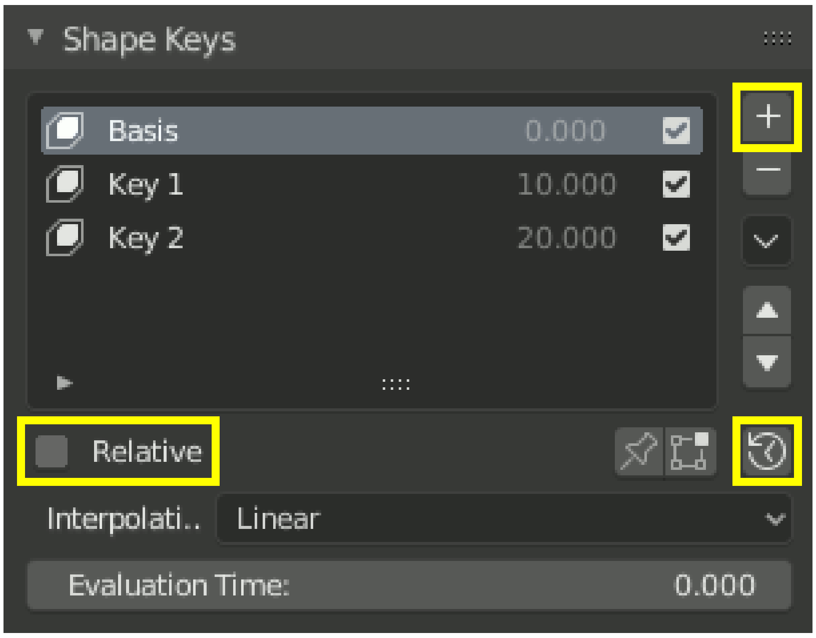


Fig. 1743: Relative shape keys example.



Shape Keys

Basis	0.000	✓
Key 1	10.000	✓
Key 2	20.000	✓

Relative

Interpolati.. Linear

Evaluation Time: 0.000

Fig. 1744: Absolute shape keys workflow.

Shape Key Operators

There are two modeling tools used to control shape keys and are found in *Edit Mode*.

2.8.10 Motion Paths

Reference

Editor 3D Viewport, Properties

Mode Object Mode

Panel *Properties → Object Properties → Motion Paths*

Reference

Editor 3D Viewport, Properties

Mode Pose Mode

Panel *Properties → Armature → Motion Paths*

Menu *Pose → Motion Paths*

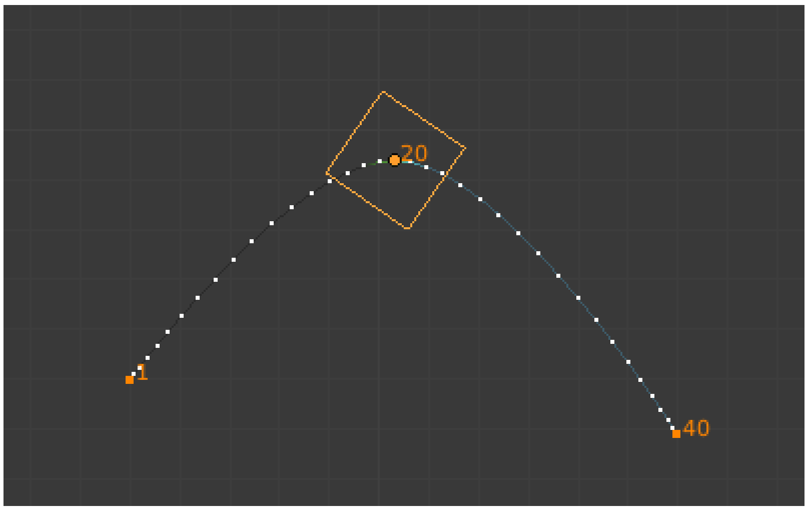


Fig. 1745: An animated cube with its motion path displayed.

The Motion Paths tool allows you to visualize the motion of points as paths over a series of frames. These points can be object origins and bone joints.

To create or remove motion paths, it is necessary to first select the bones. Then:

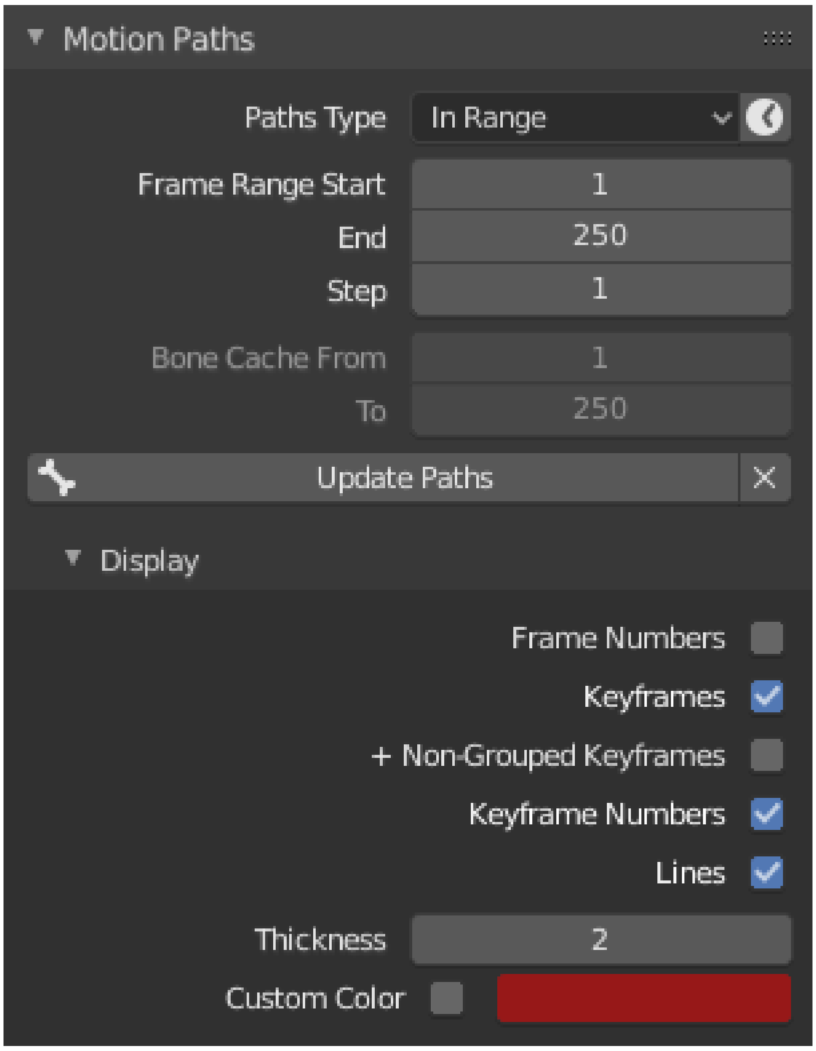
1. To show the paths (or update them, if needed), click on the *Calculate Path* button.
2. To hide the paths, click on the *Clear Paths* button.

Note: Remember that only selected bones and their paths are affected by these actions!

The paths are shown in a light shade of gray for unselected points, and a slightly blueish gray for selected ones. Around the current frame a glow indicate the direction of movement: blue towards future frames and green towards the past. Each frame is displayed by a small white dot on the paths.

The paths are automatically updated when you edit your poses/keyframes, and they are also active during animation playback. Playing the animation affects the paths only when the *Around Current Frame* option is enabled.

Options


Motion Paths

Paths Type: In Range

Frame Range Start: 1

End: 250

Step: 1

Bone Cache From: 1

To: 250

Update Paths

Display

Frame Numbers: ☐

Keyframes: ✓

+ Non-Grouped Keyframes: ☐

Keyframe Numbers: ✓

Lines: ✓

Thickness: 2

Custom Color: ☐ 🟥

Fig. 1746: The Motion Paths panel in the Armature tab.

Type

Around Frame Display paths of points within a fixed number of frames around the current frame. When you enable this button, you get paths for a given number of frames before and after the current one (again, as with ghosts).

In Range Display paths of points within specified range.

Clock Button Updates the display frame range from the scene frame range.

Frame Range

Before, After Number of frames to show before and after the current frame (only for *Around Current Frame* Onion-skinning method).

Start, End Starting and Ending frame of range of paths to display/calculate (not for *Around Current Frame* Onion-skinning method).