Curve Edition

Edit a FredoSpline curve: move / add / delete control points, change parameters globally and locally

▶ Editing a curve - Principles

With Preselection

- Launch the Edition tool with a <u>single</u> FredoSpline curve preselected
- The curve will be editable in **focused mode**

Interactive (no preselection)

- Launch the Edition tool <u>with no curve selected</u>
- Hover over a curve to highlight it **-** The curve is in **Hover mode**

Focused mode and Hover mode

- **When a curve is in focused mode**, the interactive curve selection is locked on it
  - It is shown in blue
  - It remains selected even if the mouse is not on the curve
- **When a curve is in Hover mode**, you can select another curve by moving the mouse over it
  - It is shown in lightblue
  - Click on the curve to enter the Focused mode
  - or move the mouse over another curve
- **When already in focused mode**, you can select another curve
  - Hover over it (highlighted in gray) and click
- You can edit a curve in Focused mode or in Hover mode, but Focused mode is more comfortable for some operations

Exit the Edition tool

- **ClickinVoid** to exit (as shown by cursor)

▶ Editing Control Points

Most operations are also available from the contextual menu with **Right-Click**

They can be performed in Hover mode and in Focused mode

Move a Control point

- **Click-Move** or **Click-Drag** on a control point to move it around
- Inferences and plane constraints can be set

Delete a Control point

- **Double-Click** on a control point to remove it from the curve definition

Insert a Control point - No move

- **Double-Click** **on a control segment** to insert a control point

Insert and Move a Control point

- **Click-Move** or **Click-Drag** **on a control segment** to insert a control point and move it
- This is possible in Focused mode only

Move a Control segment

- **Ctrl-Click** or **Ctrl-Click** **on a control segment** to move it

▶ Smooving Control Polygon

***Smoove** is a variant of the Move operation on control points and segments*

*It moves all control points depending on their **distance** to the control point or segment moved*

Smoove modifier

- **Shift** is the modifier when you perform a move of control point or segment
- **Shift** works permanently (**toggle**) or temporarily (**press-down**)

- Press or Toggle **Shift** <u>before</u> moving (otherwise the Shift modifier is used for inferencing)
- The curve is highlighted in light yellow when in Smoove mode

▶ Loop curve and continuity

A loop curve has its last control point equal to the first one, which then behave as a single control point

*Most types of curve support loop natively with continuity*

Making a loop from an open curve
- Move the **first** control point over the **last** control point (or vice-versa)
- The mark becomes ⊙ , indicating that a Loop is formed

Breaking a loop
- Keep **Alt** pressed when **moving the loop point**
- The point (first or last) will separate and make the curve **open**

Modifying the Curve continuity at Loop point
- **Toggle continuity** at loop with the palette button ◁ / ◁
- The option is also available in the **contextual menu**

▶ Extending a curve

Click on the extension handles ⬇ , at begin or end of the curve
- You will switch to the **Creation tool** where you can add new control points

▶ Local Edition of parameters

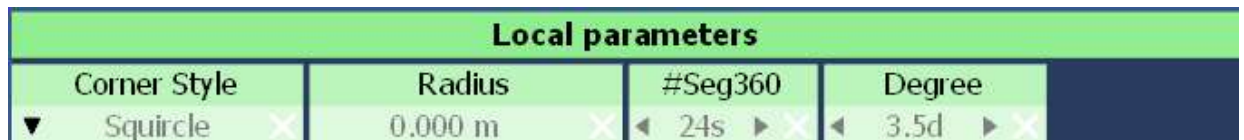*Several curves allows **modifying parameters** <u>individually</u> at control points*

*When this is possible, you will see a **small pin** 📍 when hovering the control point*

Local Parameters
*When you hover a control point, Local parameters are show in a **green button palette***

Local parameters
Tension
◄ 50% ► ✗

*There may be **several** local parameters*

Local parameters			
Corner Style	Radius	#Seg360	Degree
▼ Squircle ✗	0.000 m ✗	◄ 24s ► ✗	◄ 3.5d ► ✗

*Check the tooltip to see which one, their **modifier key** and **VCB convention***
- 📍 The pin is shown in green when local parameters have their **global value**
- 📍 Otherwise, the pin is shown in **lime green**

To change Local Parameters
- **VCB**: type the value with an optional suffix
- **Variator**: **Click-Drag** in the pin to make the variator appear (with **modifier key**)
- **MouseWheel**: gently mouse-wheel to increase / decrease the parameter (with **modifier key**)
- Use the button Palette if the pin is selected

To Select control points
*Selection allows to change parameters for **one or several** control points*
- **Click** on the pin to **select / unselect** - The pin becomes bigger 📍
- **Ctrl-Click** to **add to** or **remove from** the selection
- When pins are selected, you can freely access the button palette
- Any parameter change will **apply to ALL selected** control points
- **Esc** or **Click in empty space** to **exit** the local selection mode

▶Local Tangent

Local Spline allows changing the tangent at control points

Tangents are shown in green when they have their **default orientation**

Tangents are shown in **lime green** when they have been **modified**

To change the tangent orientation

- Click-Drag to rotate the tangent
- Ctrl-Click-Drag to only rotate the half-tangent under mouse

To reset the tangent orientation

- Double-Click on the tangent line to reset it to its default value
- Ctrl-Double-Click will reset the half-tangent under mouse

▶Inferences / direction lock

Lock direction

- Shift (toggle) to lock / unlock the *current* direction
- When a direction is locked you can **type a distance** in the VCB

Lock axis

- Use Arrows for Model axis
- **Repeat** arrow for Local axis

Unlock direction

- Press Arrow-Down

NO snapping

- Alt (toggle) to place a control point <u>without</u> snapping

Clear inferences

- BackSpace to clear the visual information related to inferences

▶Plane constraints

Force an Axis plane

- Ctrl-Arrow-Left → **Vertical** Plane YZ (blue-green) (X normal)
- Ctrl-Arrow-Right → **Vertical** Plane XZ (blue-red) (Y normal)
- Ctrl-Arrow-Up → **Horizontal** Plane XY (red-green) (Z normal)
- **Repeat the arrow key** to toggle between **local** and **model** axis (if applicable)
- You can also use the palette buttons
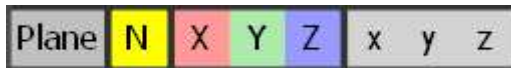
Force a Custom Plane

- Toggle Ctrl **while hovering an element** in the model (do <u>not</u> click)
  - **Face** → Plane of the face
  - **Edge** → Plane orthogonal to the edge
  - **Axis** → Plane orthogonal to the axis

Remove Plane constraint

- Ctrl-Arrow-Down → Remove any plane constraint

Plane from first 3 points

You can force the curve to stay on the plane defined by the first 3 points when NO other plane constraints are active

- Ctrl-3 → Toggle this default constraint
- You can also use the toggle button

▶Cusp (or Break)

*Some curves (Bezier, Local Spline] have the option to force a **Cusp** or **Break** at control points*

This feature is available in Creation and Edition mode

Cusp when Editing a curve

- The Cusp marks are shown when you **hover the mouse over control points**
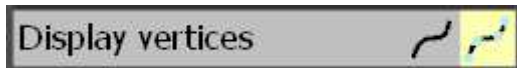- Click on the mark to enable ◀ / disable ◁ the Cusp at point

▶Show / Hide curve vertices

You can display or hide the vertices of the curves

This option is available for all tools

To show / hide vertices...

- From the option bar

  Display vertices

- From the toggle buttons

- From the **VCB**: type `*` to toggle display
- From the contextual menu

▶Picking Style

You have the choice between two picking styles

Sketchup *style is the native one*

Extended *provides additional inferences*

To toggle the Picking Style...

- From the option bar

  Picking style

- From the toggle buttons

- From the contextual menu

▶Flow Management

Pause

- **TAB** to pause: this interrupts the input or current operation

  ⊕Paused
  Paused
  Click to resume
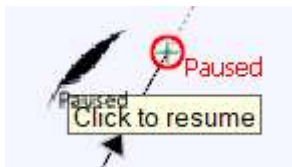
- This is useful to change the curve parameters in the palette
- **TAB** again to resume

Undo

- **Esc** or **Ctrl-Z** or ↩ to **undo** the last operation

Redo

- **Esc** or **Ctrl-Y** or ↪ to **redo** the last operation

Finish and Exit

- Click on ⏏ to exit
- or **Click in empty space** in most situations (as shown by the cursor)
- For Creation tool, you have **additional finish options** (check the contextual menu)

Default Parameters

- Click on ✕ to access the **Default Parameters dialog**

Palette buttons

- You can also use the **palette buttons**

▶Variators

At any time, you can modify the numeric parameters of the curve via **Click-Drag**

Variators are available for all FredoSpline tools

In Palette button

- **Click-Drag** in the button → Variator will appear and can be moved

In empty space

- **Click-Drag** in the button → Variator will appear

For Local Parameters

- **Click-Drag** in the small circle → Variator will apply to local parameter

Modifiers

- When there are **several parameters**, press-down a modifier key (**Ctrl**, **Shift**, **Alt**) while dragging
- The applicable modifier for a parameter is indicated in the **tooltip**

Corner Style: Squircle
Radius: 1.0 m   [Drag]
#Seg360: 24s   [Shift-Drag]
Degree: 3.5d   [Ctrl-Drag]
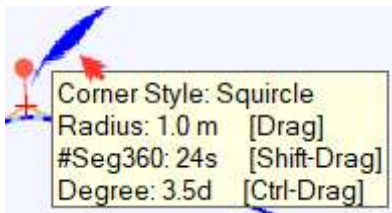
▶Custom curves

FredoSpline comes with a set of curves with predefined parameters

You can however create Custom curves with **your own parameters**

Creating a Custom curve

- At any point in time press the button 🖫 to create a custom curve with the current type and parameters
    - Note that the parameters must be **different** from the default parameter of the curve type
    - This is indicated by a **small** * appended to the curve name
- You will be prompted for a **name** and the **scope**

Create Custom Curve
Custom Curve Name   Bezier Classic (1)
Scope   SketchUp
OK   Cancel
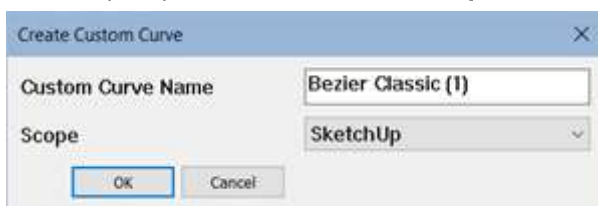
- **SketchUp**: in All models and All SketchUp versions
- **Model**: **only in the model** and its copies

Managing Custom curves

- Press the button ⚙ (next to the curve name) to manage all your Custom curves

- A dialog box appears; you can **rename**, **delete** or **change the scope** of your Custom curves
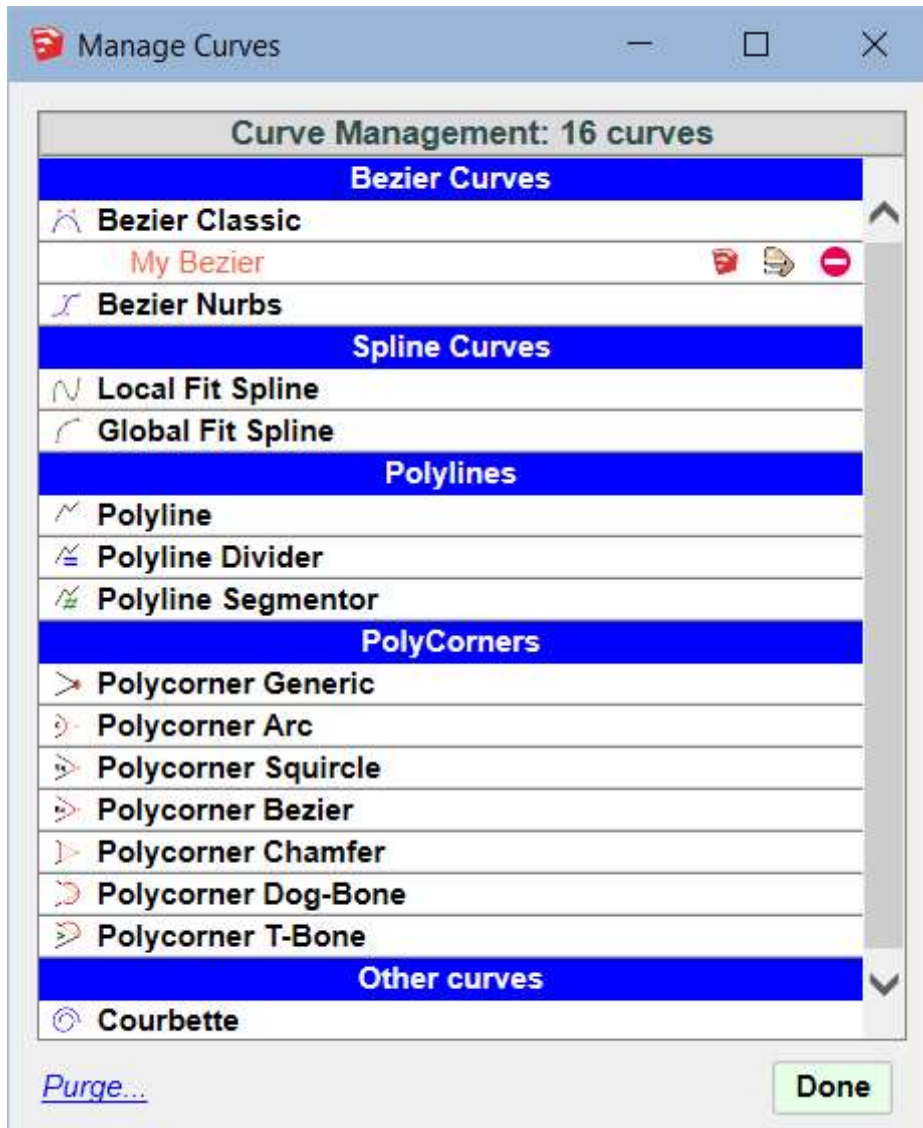
▶Curve Family: BEZIER

Smooth curves NOT passing by the control points

⋀ *Classic Bezier is of degree 3 and tangent to the control polygon*

ʃ *Bezier Nurbs can have a higher degree (**tension**) giving it more rigidity*

*All Bezier curves support **Cusp** (discontinuity at control point) and **native loop***

Global Parameters (Classic and Nurbs)
- **Number of segments**: (0 for automatic)

Global Parameters (Nurbs only)
- **Tension** [0..100%]: increase the degree to gives more rigidity

Local Parameters
- **Weight** [0..1] and [1..10]: proximity to the control point (Default 1)

▶Curve Family: SPLINE

Smooth curves passing BY the control points

∿ *Local Fit Spline is close to the control polygon and has many local tweaking*

ſ *Global Fit Spline passes very close to control points and keeps a global shape*

*All Spline curves support **Cusp** (discontinuity at control point) and **native loop***

Parameters for Local Fit Spline
- **Smooth Angle**: deviation angle for smoothing the curve (in degree)

- **Tension** [0..100%]: Kind of rigidity making the curving closer or farther to the control polygon
- The tension can be adjusted locally at each control point
- **Tangent** orientation at control points, symmetrically and asymmetrically

Parameters for Global Fit Spline
- **Number of segments**: (0 for automatic)
- There are NO local parameters

▶ Curve Family: POLYLINE

Polylines based on the control points with division options

Polyline Basic: Simplest polyline just joining the control points

Polyline Divider: Polyline divided in segments of specified length

Polyline Segmentor: Polyline divided in a specified number of segments

Polyline Path: Polyline divided for camera path frames

When divided, polylines may not pass through all control points

Polylines do not have local parameters

Polylines are usually used to divide a path formed by an edge sequence

Parameters for Divider
- **Interval**: length of segments
- **Method**: Specify how to handle the division

Parameters for Segmentor
- **Number of segments**: Number of equal segments
- **Method**: Specify how to handle the division and the respect of the control points

▶ Curve Family: POLYCORNER

*Polylines with **corner shaping at vertices**, configurable **globally** and **locally***

Polycorner Generic: Fully configurable Polyline, with any corner type

Polycorner Arc: Corners are based on **Arc of Circle**

Polycorner Squircle: Corners are based on **Squircle**

Polycorner Bezier: Corners are based on a **Bezier junction**

Polycorner Chamfer: Corners are based on a **straight Chamfer**

Polycorner Dog-Bone: Corners are based on a **Dog-Bone**

Polycorner T-Bone: Corners are based on a **T-Bone** on the longest side

You can mix corner types for any Polycorners, and remove corner shaping at any vertice

All Polycorner curves support native loop

Each type of corner has its own parameters, avaibale globally and locally