Curve Creation

Create a variety of curves defined by their control points

▶Creating a curve

A curve is created by a sequence of control points

Entering Control Points

- Click to enter the **first** control point
- Click-Drag or Click-Move for the **subsequent** control points

Finish Curve creation

- Double-Click or Enter to finish and switch to Edition
- Alt-Double-Click or Alt-Enter to finish and create another curve
- Ctrl-Double-Click or Ctrl-Enter to finish and exit

▶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

Loop when Creating a curve

- When entering a control point, just move it on the first point
- The mark becomes  , indicating that a Loop is formed
- Double-Click when entering the first point to create a loop curve *directly* and then just enter the intermediate points

Curve continuity at Loop point

*For some curves (Bezier, Spline, Polycorner) continuity is the **default** but can be disabled*

- To disable continuity, keep Alt pressed (or Alt-Double-Click when moving the last point over the first point.
- You can also **toggle continuity** at loop with the palette button  / 

▶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 without 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 not 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 Creating a curve

- The Cusp mark is only shown for the **last point** entered
- Click on it to enable / disable the Cusp at point
- In **Pause mode** all Cusp marks are shown and can be set on or off

▶Generate in container

*You can create a curve in a **Group** or **Component***

This allows removing interference with the model geometry

Set the generation context...

- From the option bar

- From the toggle buttons

- From the contextual menu

▶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

- From the toggle buttons

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

▶Start-End / Sequential creation

*You can create a curve by **entering the first and last point** (Start-End)*

*or enter the point **in sequence***

***Classic Bezier** uses Start-End by default*

To toggle Start-End mode...

- From the option bar

- From the toggle buttons

- From the VCB: type / to toggle mode
- 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

- From the toggle buttons

- From the contextual menu

▶Flow Management

Pause

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

- 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

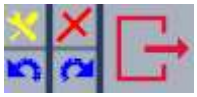
- Click on to exit
- or Click in empty space in most situations (as shown by the cursor)
- For Creation tool, you have **additional finish options** (check the contextual menu)

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

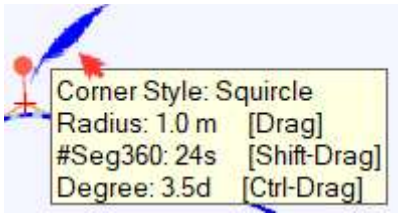
- **Click-Drag** in the button → Variator will appear and can be moved

In empty space

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

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**



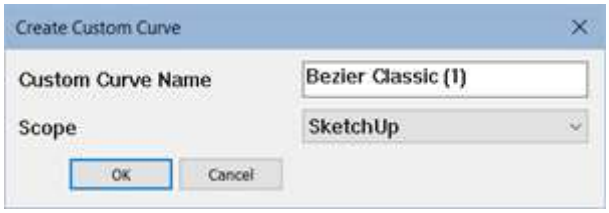
▶Custom curves

FredoSpline comes with a set of curves with predefined parameters

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

Creating a Custom curve

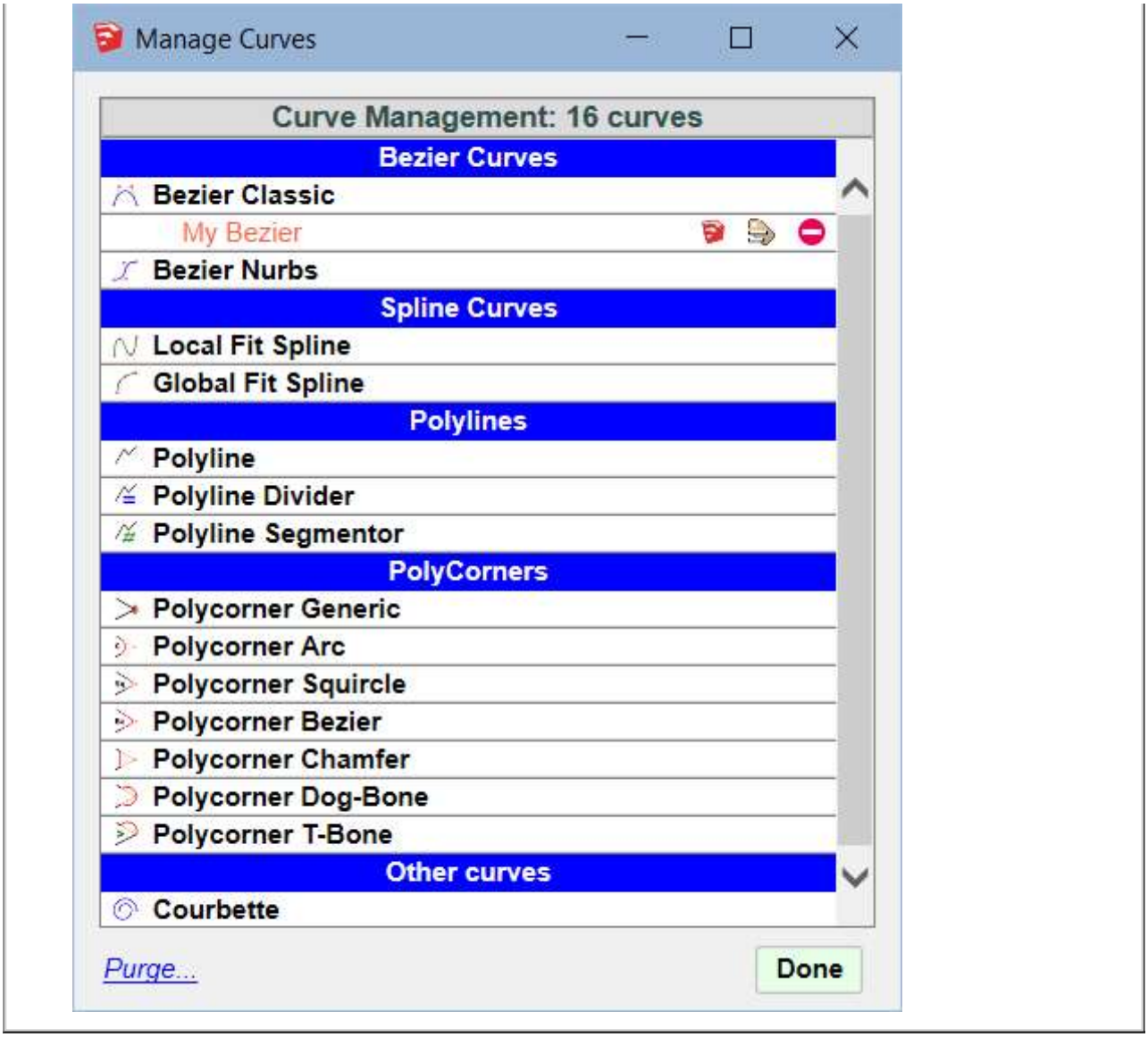
- At any point in time press the button  to create a custom curve with the current type and parameters
  - Note that the parameters must be **different** from the default parameter of the curve type
  - This is indicated by a **small** * appended to the curve name
- You will be prompted for a **name**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



  -  **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*

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*

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