Documentation v.1.0

Introduction

Thank you for purchasing Swiss Army Spline. It is a simple yet powerful tool which allows you to generate all kinds of spline based meshes and animate objects along paths. The goal of this tool is to give you as much control as possible while keeping workflow complexity and learning curve to a minimum. You are not expected to deeply study this manual. The implemented tooltips will help to eliminate possible confusion while you are working and the TUTORIAL VIDEO (link) will provide you with most, if not all, necessary knowledge.

There are 5 main aspects to Swiss Army Spline:

1. Spline Editor

   Create a path based on control points.

2. Profile Editor

   Design 2d shapes (profiles).

3. Mesh Generation
   - **Extruders** Extrude profiles along a path.
   - **Distributors** Distribute 3d meshes along a path.
   - **Pillars** Extrude a profile orthogonal to a path.

4. Instantiation

   Distribute game objects along a path.

5. Animators

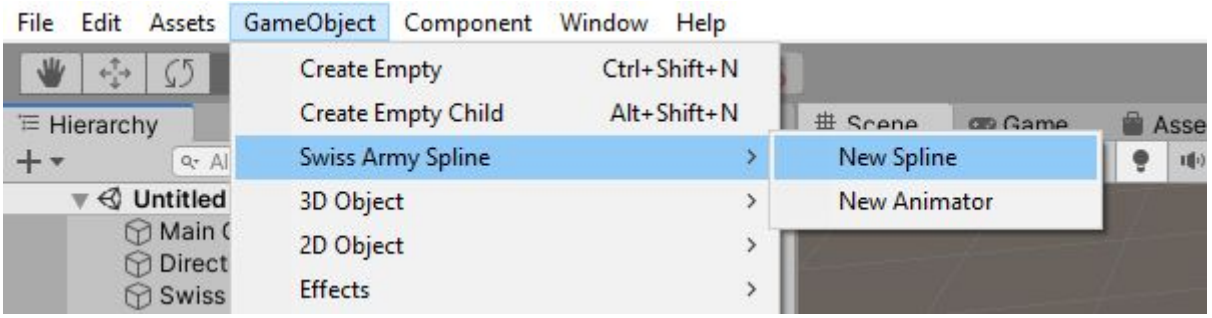
   Animate game game objects along a path

On the following pages each of these aspects will be explained.
Note: The words “Spline”, “Path”, and “Curve” are used interchangeable in this document.

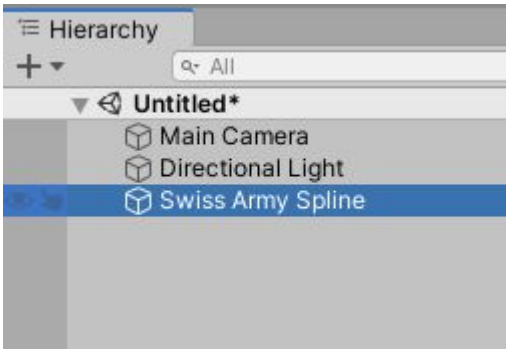
The Spline Editor

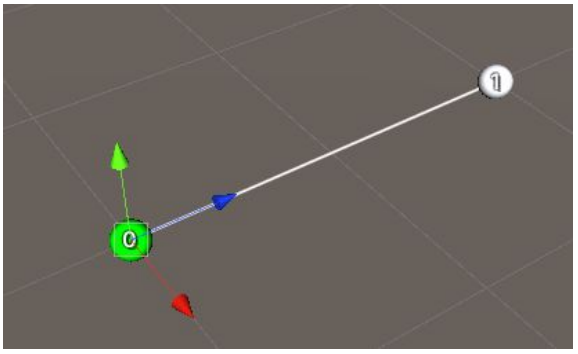
Creating a new spline object:

In the gameobject menu, select “Swiss Army Spline” and click on new spline. Alternatively you can right click in the hierarchy to get the same menu in the hierarchy context.



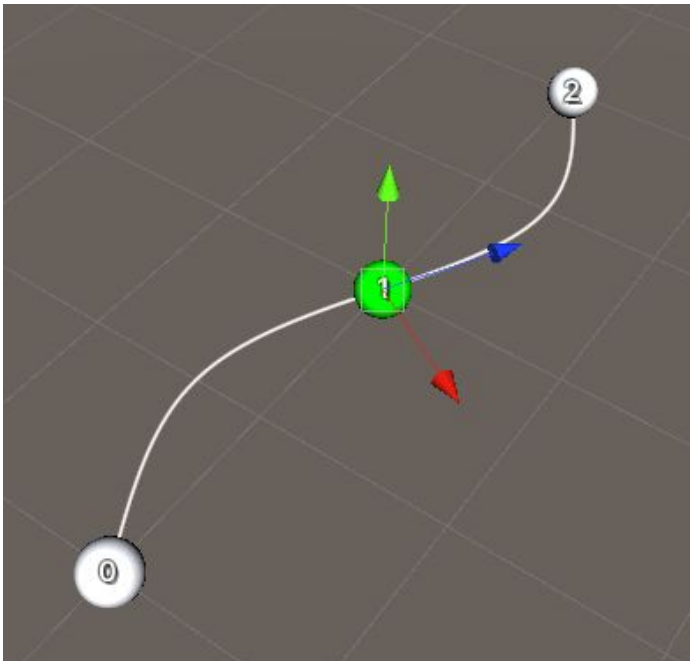
The hierarchy will be populated with a new game object called “Swiss Army Spline” and Scene View will show the spline gizmos:

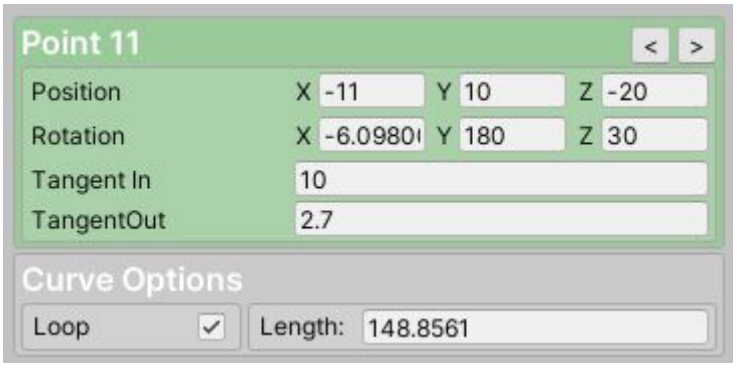


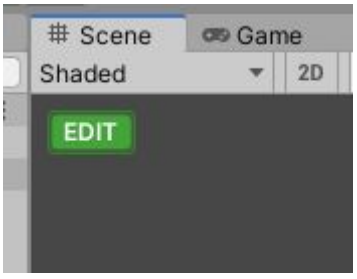


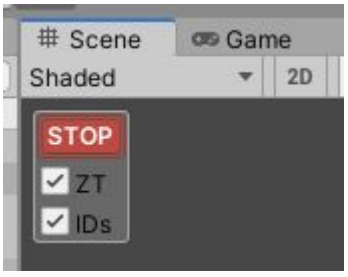
Note: Only selected Swiss Army Spline objects will have visible gizmos in the scene view
With that knowledge we can focus on the first aspect of Swiss Army Spline.

The spline editor lives in the scene view and in the inspector.



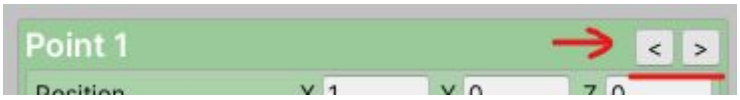






Selecting points

Points can be selected by clicking on them in the scene view. alternatively you can cycle through points in the inspector.



Adding new points

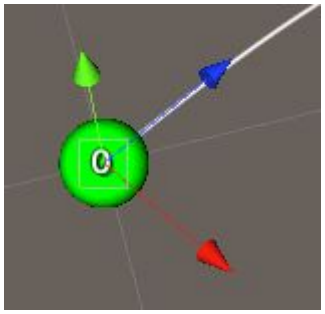
To add a new point hold shift and click on the path to insert a point or outside of the path to append a point

Deleting points

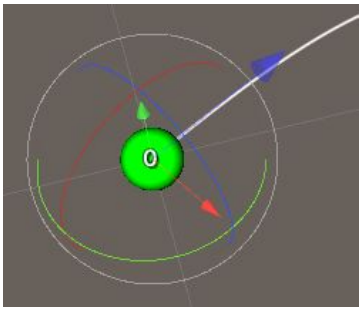
To delete a point simply hold control on the keyboard and click the point you want to delete.

Manipulating points

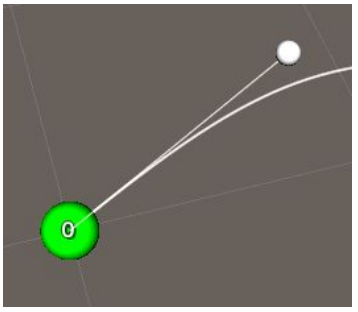
Depending on which tool you have selected you can move, move, rotate, or scale a point.



Move Tool

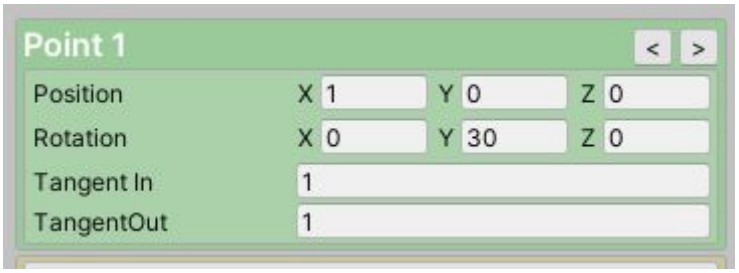


Rotate Tool



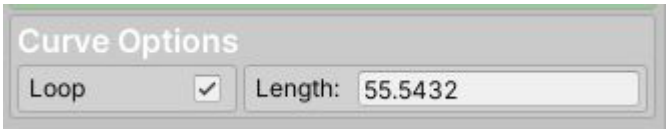
Scale Tool

In addition you have numerical control over position, rotation, and scale (length of the tangents) in the inspector:

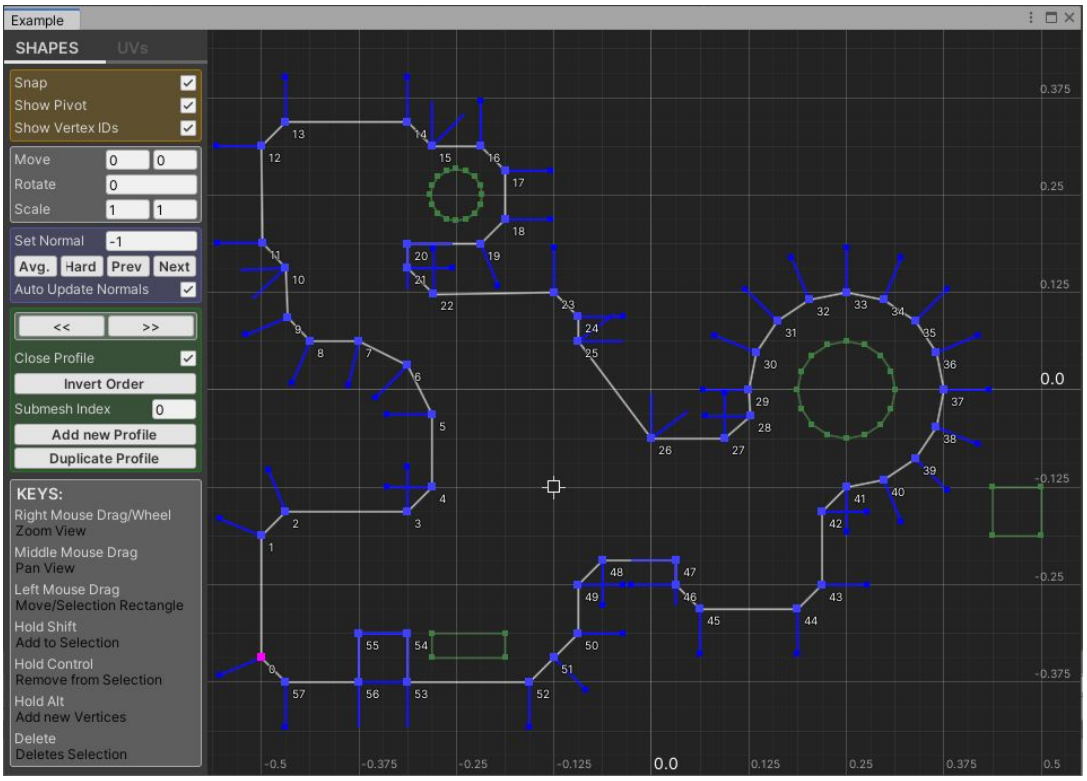


Making a spline loop

If you want your spline to be a closed loop, simply enable the checkbox in the curve options of the inspector.



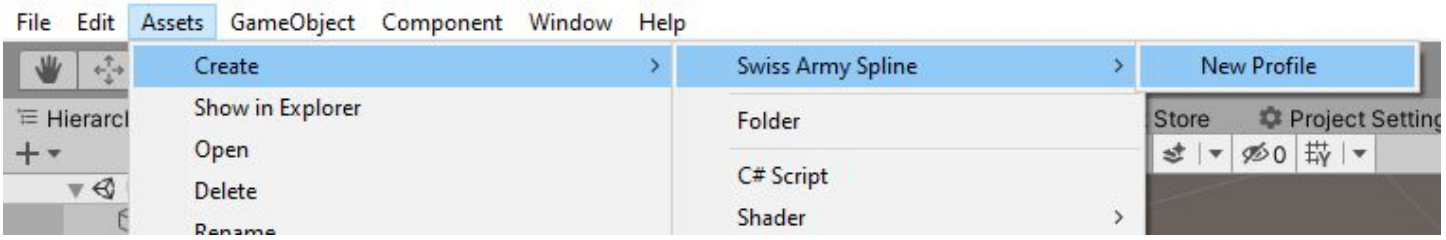
The Profile Editor



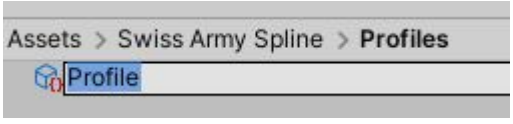
The Profile Editor allows you to create complex 2d shapes which then can be used to extrude a 3d mesh along a path.

Creating a new profile

A new profile can be created by accessing “Swiss Army Spline” in the Assets Menu. Alternatively you can right click in the project view to get the same menu in project context.

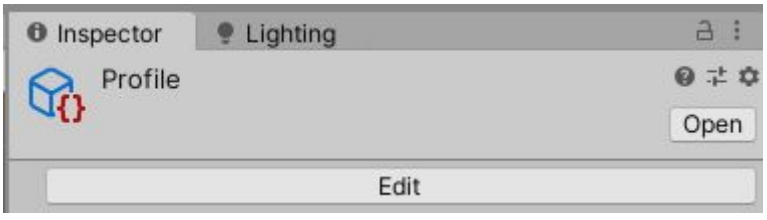


A new file will appear in the project, ready to be named.



Opening the profile editor

With a profile selected, just click on the “Edit” button in the inspector and the profile editor window will be opened.

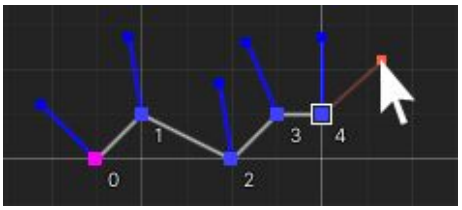


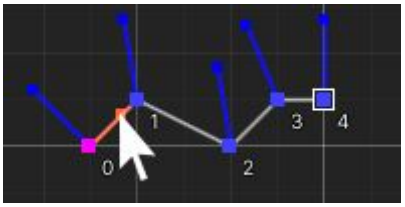
Manipulating the view

Hold the middle mouse button and move the mouse to pan the view. Hold the Right mouse button and move the mouse or rotate the mouse scroll wheel to zoom in or out.

Adding points

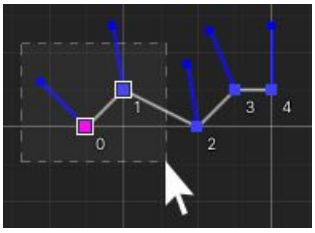
To add points simply hold the "**Alt**" button on your keyboard and left click in the editor. Depending on the to the distance of the mouse cursor to an already existing line the point will either be inserted on the line or appended to the end.





Selecting points

To select a point simply click on it with the left mouse button. You can also select multiple points by dragging a selection rectangle with the left mouse button pressed. To add points to an existing selection you can do the same while holding the "**Shift**" button on your keyboard. To remove points from an existing selection the same principle applies but with the "**Control**" button on your keyboard.



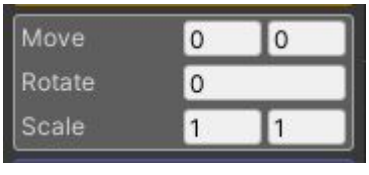
Deleting points

To delete one or more points simply select them and press "**Delete**" on your keyboard.

Note: All mouse/keyboard commands are also shown in the left pane of the editor itself (see the image on the right side of this page). So you don't need to memorize the commands.

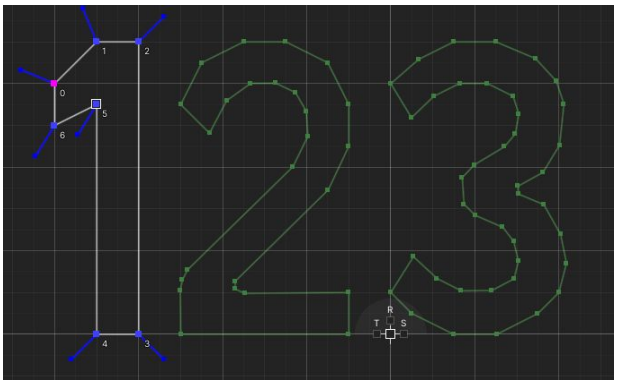
The pivot

The pivot lets you perform translation, rotation, and scaling to a selection relative to the pivots position. This is e.g. useful if you need to create a perfect circle or modify a carefully created profile. "T", "R", "S", stand for Translate, Rotate, and Scale. The handle in the center is used to reposition the pivot. Clicking and dragging in the according handle allows for modifications "by hand".



For numerical transformations you can use the The "Move", "Rotate", and "Scale" input fields on the left editor panel. These work in relation to the pivot position, So if you want, for example, to rotate a selection of points (also known as vertices) by an exact amount you can simply enter the number in degrees and the rotation will be performed clockwise. Moving works as an offset to the current position of your selection and scaling works multiplicative.

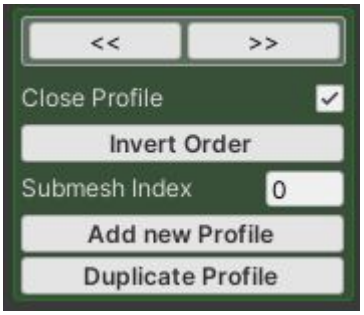
Sub-profiles and profile related options



A profile file can contain more than one profile. Unselected sub-profiles are shown in green. You can select them simply by clicking on them or by using the arrow buttons in the green section on the left panel of the profile editor.

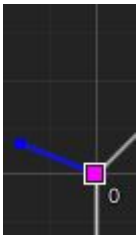
Sub-profiles are useful if you need to have disconnected parts or if you need to apply a different material on that portion of the generated mesh (mesh generation will be explained further down below).

Additional options exist per sub-profile:

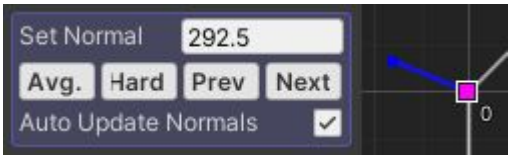


Close Profile: Connects first and last point.
Invert order: The order defines if a surface points inward or outward. The general rule is: Clockwise ordering results in outward facing polygons and counter clockwise ordering results in inward facing polygons. If you want to create a pipe with an inside and an outside you may want to create the outside first then duplicate the profile, scale it down a bit and then invert the order to make the polygons face the correct way.
Submesh Index: If you want to apply a different material to a sub-profile you need to specify a unique submesh index. The submesh index is used by unity's mesh renderer component. Make sure that you use sequential indices, e.g. 0, 1, 2, 3, 4, and so on. Mesh renderers don't like unnecessary or empty material slots and can throw errors if submesh indices are not sequential (a bad example would be: 0, 1, 4, 5, ...).
To add a **new profile** or **duplicate** existing ones simply use the according buttons.

Normals



Normals are a topic on its own. Normals are extremely important to get correctly lit objects in any 3d software, Unity being no exception. In the profile editor normals are shown as blue lines with a handle at the tip. To manipulate a normal just click and drag the handle and orient it in the desired direction. This can be useful in some cases but usually you want a more automated way of setting normals accurately. If you drag a point around its normal will automatically update based on the average directions to its neighbours (given that "Auto Update Normals" is enabled).



For precise control you have the following options:
Set Normal, Average, Hard, Previous, Next

Set Normal - sets the angle in degrees, starting at the top in clockwise direction.

Average - (fig.1) sets the normal based on the directions to its neighbour points.
Hard - (fig.2) splits the normal in 2 parts, one for the direction to the previous neighbouring point and one for the direction to the next neighbouring point.
Previous and **Next** - (fig.3) are very useful if you want to have smooth looking corners

fig.1

fig.2

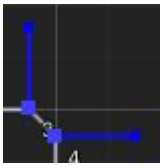
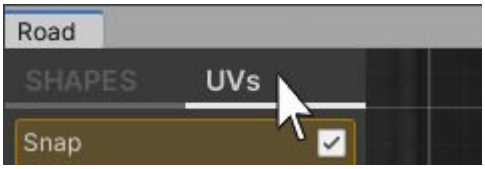


fig.3

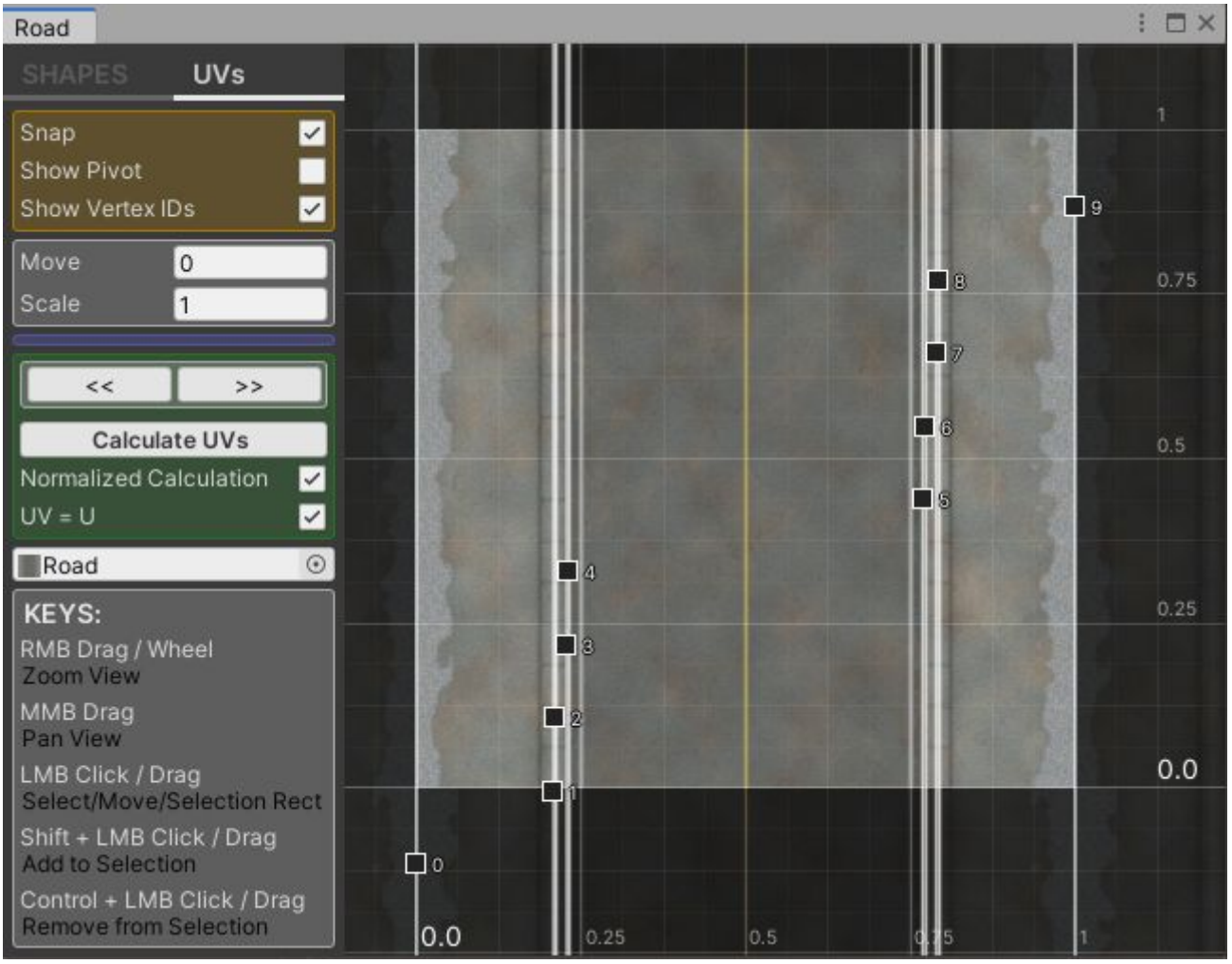
UVs

The built in profile UV editor allows you to setup and manipulate uvs in various ways.

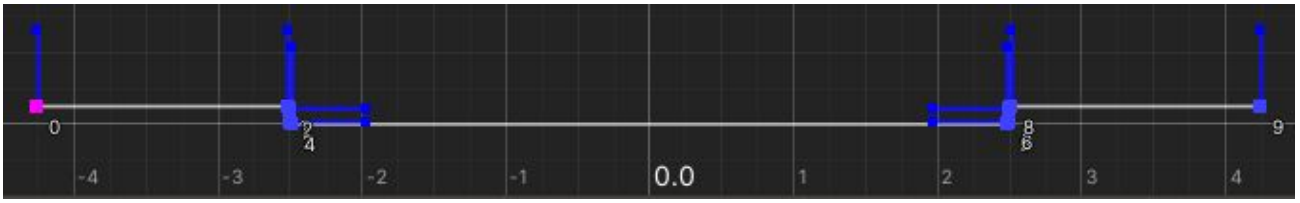


To switch to the UV editing mode simply click the “UVs” tab in the top of the left editor panel.

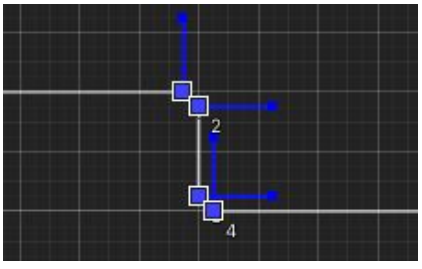
Here a road texture applied as preview:



The according profile:



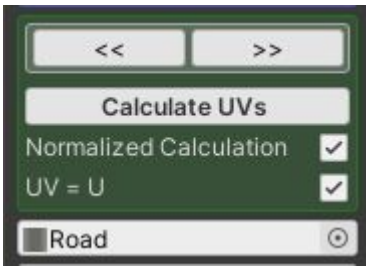
Each point in a profile has its according position in uv space. Since a profile is 2 dimensional we are actually only setting either U or V and the extrusion will handle the other axis. This zoomed view illustrates the way how uv space and actual poin positions correlate.



The basic uv workflow after a profile has been created is as following:

- Apply a desired preview texture
- Decide on the UV axis - either U (horizontal) or V (vertical).
- Move the points in uv space until they are at the desired positions.

The Profile Options in UV editing mode:

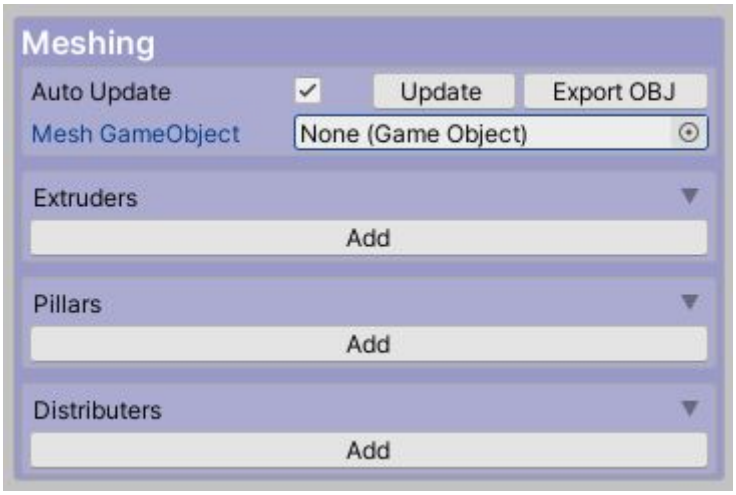


Arrow Buttons allow you to switch sub-profiles without leaving the UV editor.
Calculate UVs will automatically set the UVs for you based on the distances between the points in the profile.
Normalized Calculation - if this is enabled the result will be remapped into a 1 to 1 range. If this is disabled the calculated values will not be changed and the actual distances will be kept (this is useful to get perfectly square tiling extrusions without further uv manipulation).
UV = U - if this is enabled the uv space will tile horizontally, otherwise vertically.
The **Texture Picker** can be used to select the preview texture.

In case you want to mirror or fold uvs using the pivot can be of great help. Please refer to the tutorial video for tips and tricks regarding UV editing.

Mesh Generation

The meshing section in the inspector of a Swiss Army Spline Object looks like this.

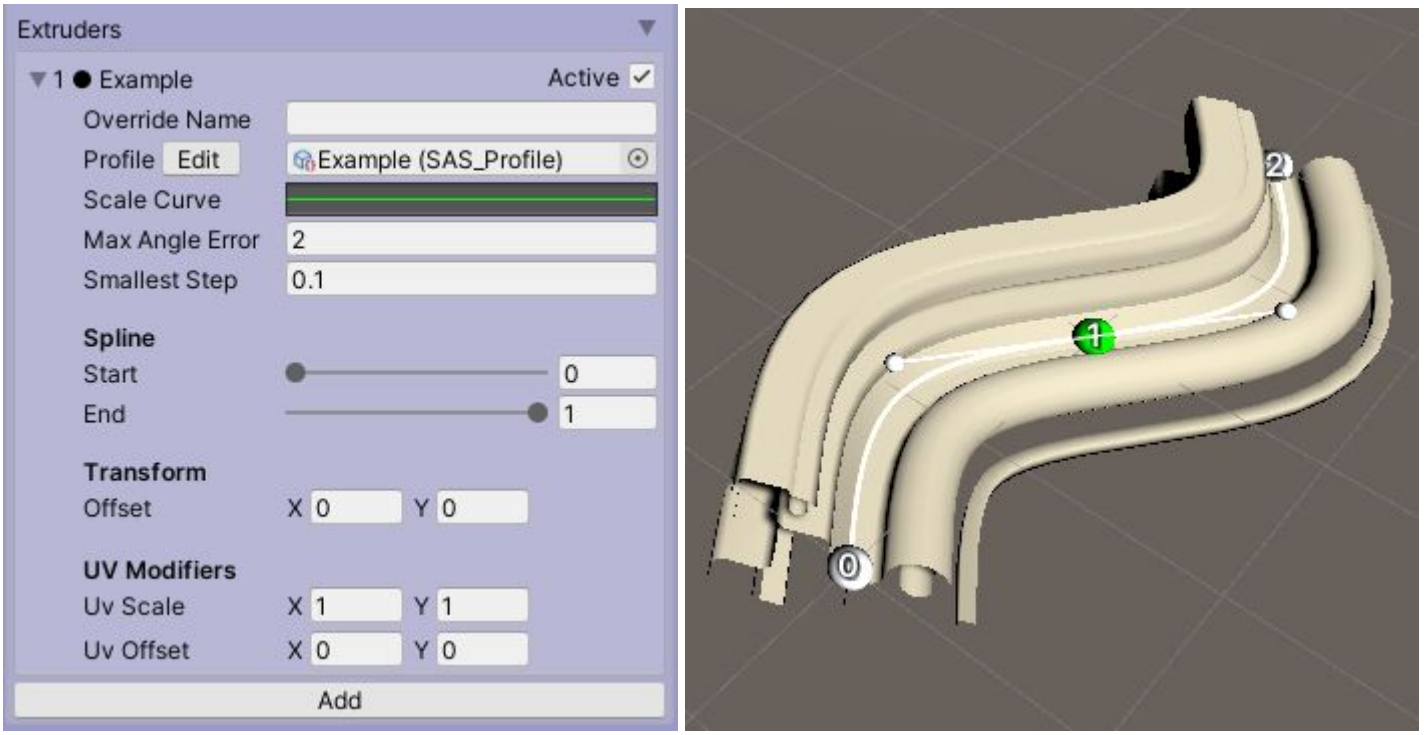


Auto Update - If this is enabled the mesh will be automatically updated upon changes.
Update - Press this to manually update the mesh.
Export OBJ - this will save the generated mesh in a specified location.
A **Mesh GameObject** needs to be specified to hold the generated mesh (Note: This object can not be the spline object itself).
Extruders - Extrude profiles along a path.
Pillars - Extrude profiles orthogonal to a path.
Distributors - Distribute 3d meshes along a path.

There are tons of different possibilities to use the Extruders, Pillars, and Distributors, this manual can only explain the basic concept and functionality, coming up with individual use cases is up to the user. Also every knob, slider, input field, etc. has tooltips implemented and using them is more helpful than a bloated manual. For this reason some things will be skipped in the manual.

Extruders

An extruder in action.



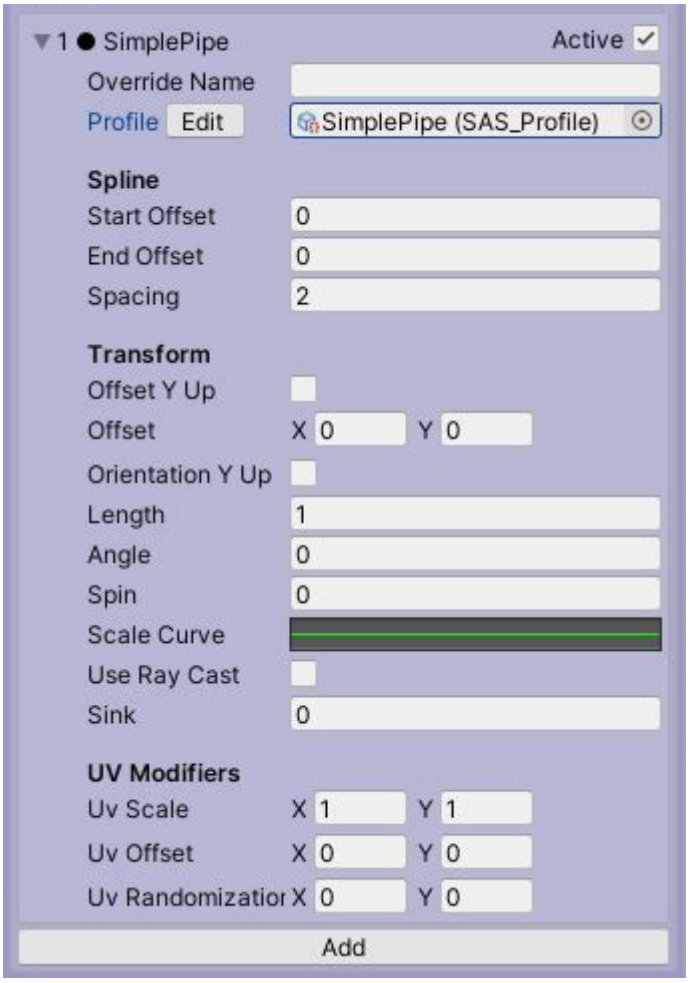
Extruders are simple, a 2d profile will be extruded along a 3d path, that generates a 3d object.
There are only a few options available which are self explanatory:

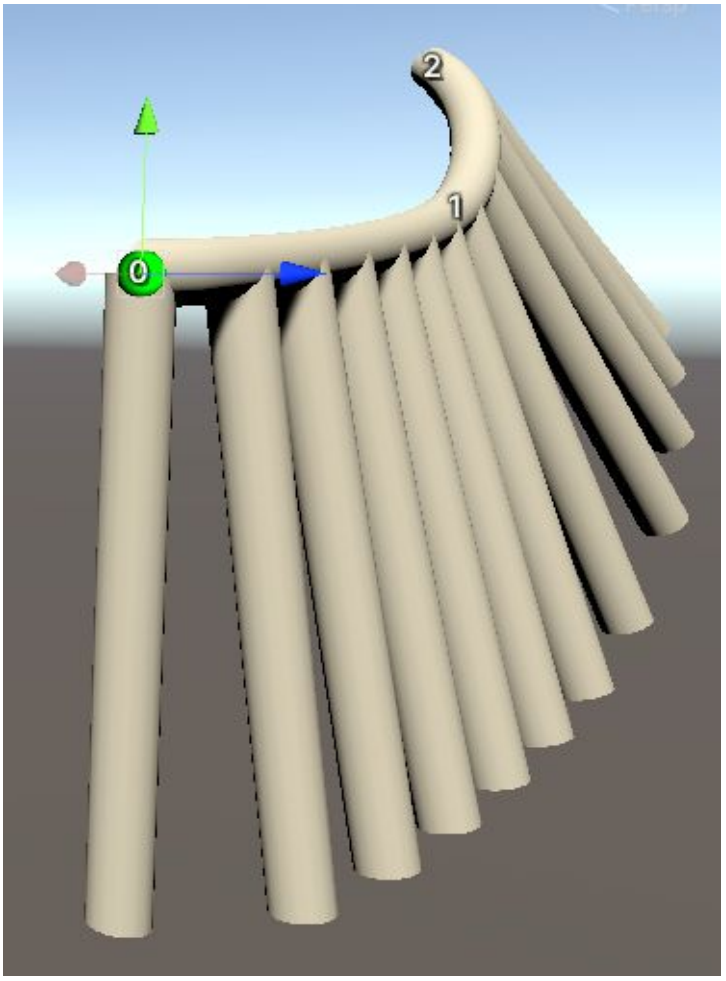
- You can modify the thickness of the profile along the path percentage with a Scale Curve.
- You can define the quality of the generated mesh by changing the “Max Angle Error” and the “Smallest Step”. You may use this to generate LOD versions of your meshes.
- You can define a start and end offset of the path length in which the mesh will be generated.
- You can offset the profile orthogonal to the path - useful if you want to reuse the same profile offsetted multiple times on a path.
- You can modify the UVs.

Example below: The rails and the gravel are extruded profiles.

Pillars

Pillars in action.

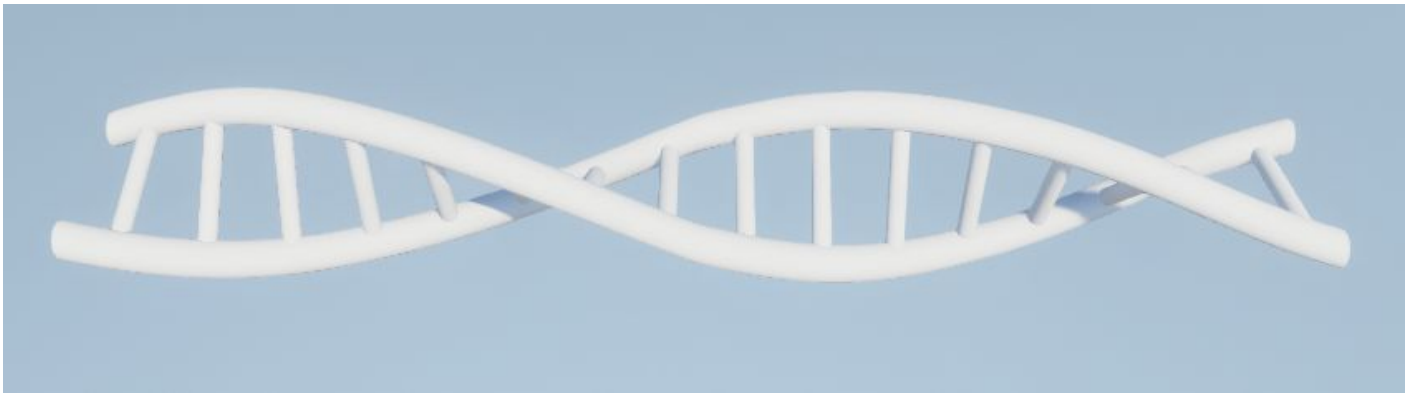




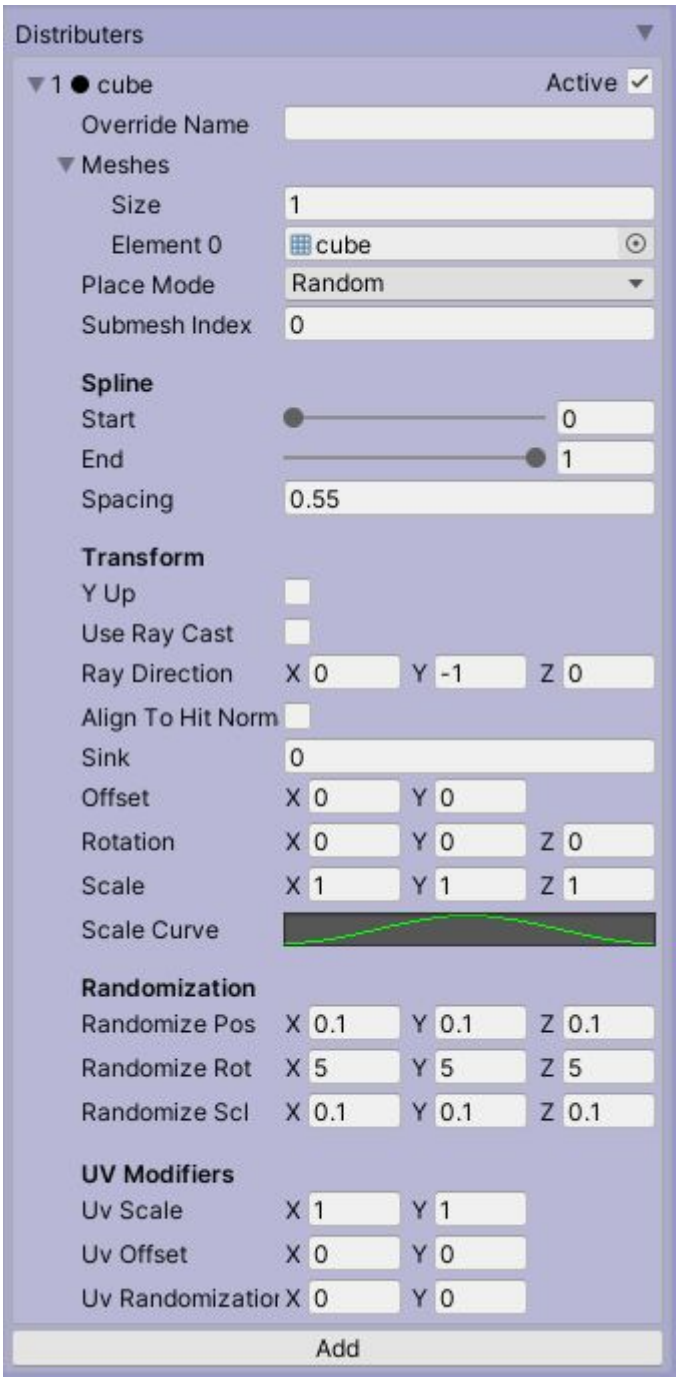
Pillars are very similar to extruders. They also work by extruding a profile. The differences to extruders are that they are not extruded along the path but ortogonal to it. Pillars are very useful as pillars for a bridge(hence the name) or as connecting strings of a rope bridge, they even could act as feathers on the wing of a bird as the image above suggests. Pillar options are similar to the options of extruders but extend them by some:

You can use a raycast to limit the length of the pillars.
You can limit their orientation to Y-Up.
You can define the direction of extrusion by the angle around the path.
You can define a spin angle to rotate the pillars around themselves.
You can sink their ends into the surface when using raycasting, useful with uneven surfaces.
You can randomize their uvs, useful to hide obvious repetitions with texturing.

In this example 2 extruders on a twisted path are connected by a pillar layer to form a double helix.

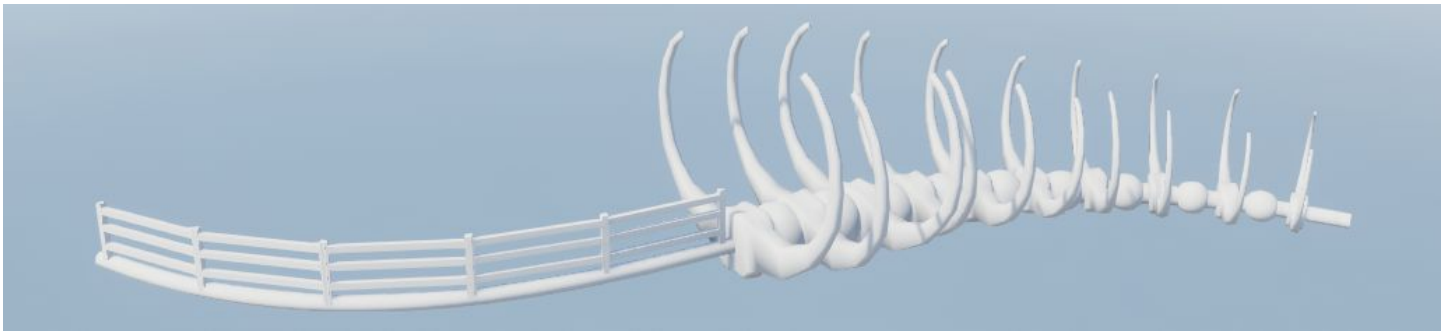


Distributors

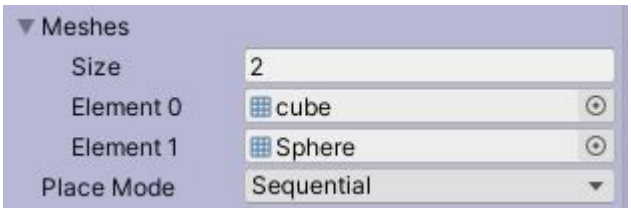


Distributors are very powerful. They distribute existing meshes along the path. They are useful whenever something needs to repeat very often. For example this dragon rib cage, only one spine segment has been modeled, the distributer repeats it according to the specified rules. The individual repeated meshes will be combined so that there will be only 1 resulting mesh.

This path uses two distributors, one with multiple meshes (Sphere and Spine Segment), one with a fence.

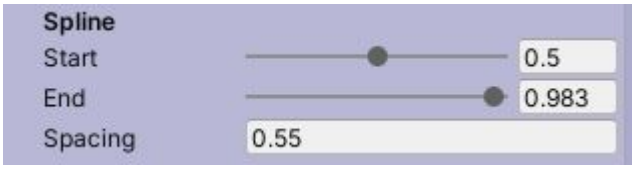


Meshes



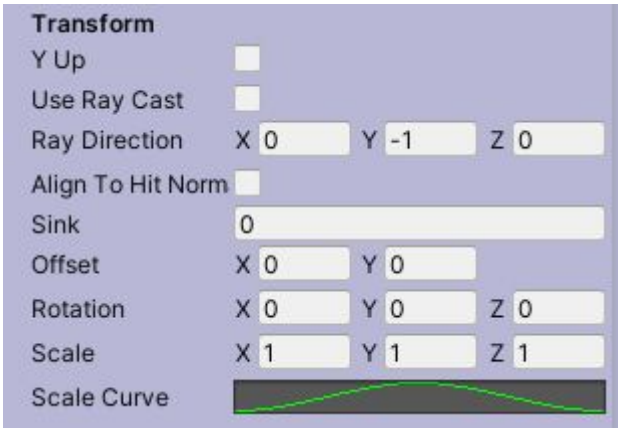
Multiple meshes can be assigned, the “Place Mode” defines how they will be ordered, the modes are:
Random - the meshes will be chosen in random order.
Sequential - the meshes will be chosen in sequential order.
Ping Pong - the meshes will be chosen sequentially but with their indices counting up and down. E.g. 1 2 3 2 1 2 3 2 1.

Spline



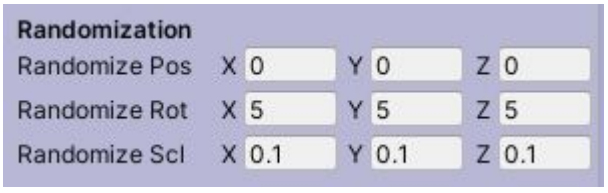
Start / End Offset - defines the start and end distance along the spline in which the distributor will be active.
Spacing - defines how far apart the meshes will be distributed, this number is rounded for even distribution.

Transform



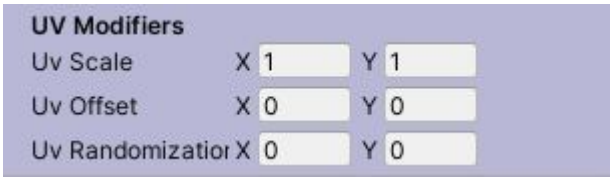
Y Up - If enabled the distributed meshes will stay upright.
Use Raycast - if enabled a raycast will be fired to determine where a mesh should be positioned.
Ray Direction - specifies in which direction (relative to the current point on the spline) the ray will be fired.
Align To Hit Normal - If enabled the mesh will be oriented along the normal of the surface the ray has hit.
Sink - how far into the surface the mesh will be sunken.
Offset, Rotation, Scale - relative to the point on the spline.
Scale Curve - Controls how big the meshes will be relative to the percentual distance along the spline.

Randomization



Randomize Pos - Randomizes the position of the mesh.
Randomize Rot - Randomizes the rotation of the mesh.
Randomize Scl - Randomizes the scale of the mesh.
All randomizations are relative to the current point on the spline.

UV Modifiers

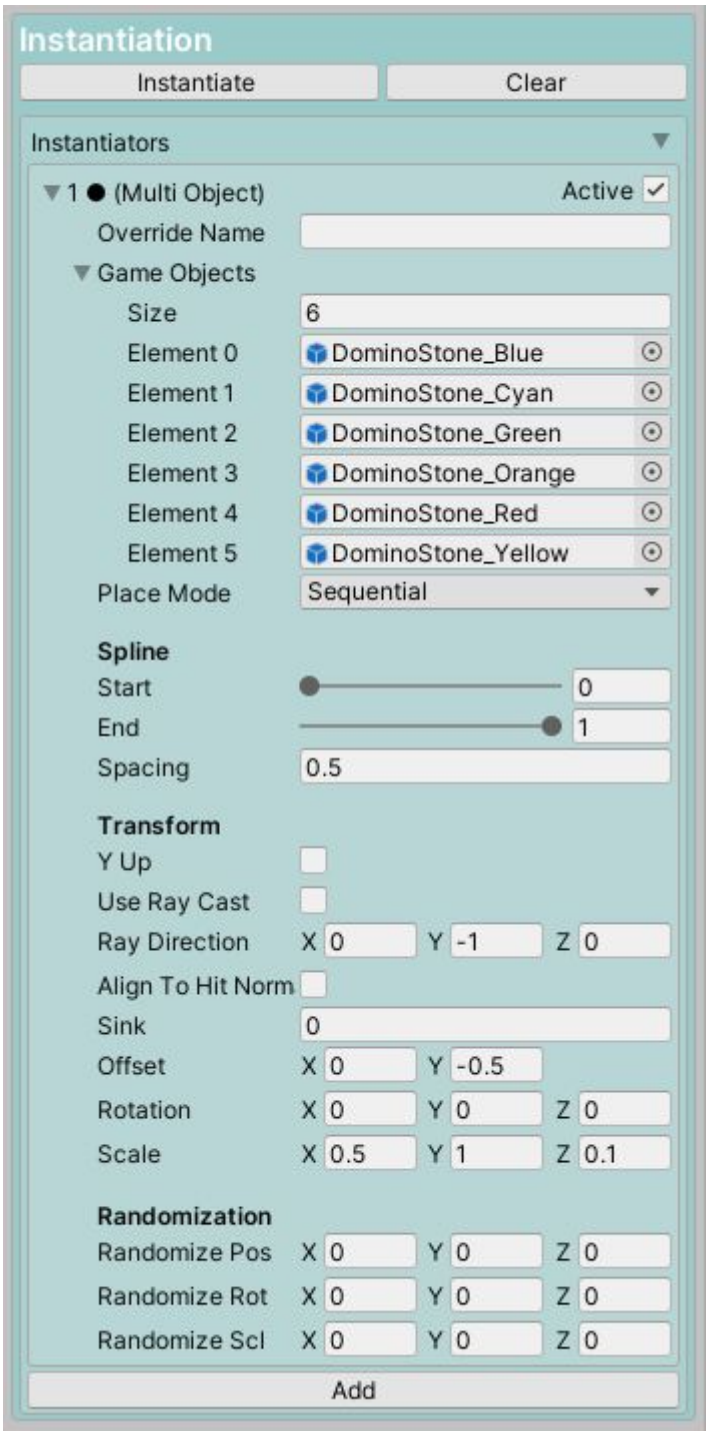


UV Scale - Scales the UVs, very useful for example if you want to repurpose a unity primitive (like cube or cylinder).
UV Offset - Offsets the UVs
UV Randomization - Adds a random offset to the UVs.

Notes:
Multiple Distributors can work together on the same curve. A good example would be lamp posts on a road combined with other side objects e.g. crash barriers.
Everything in the meshing category will combine the results into one single mesh. If you need to keep them as separate gameobjects you need to use instantiators.

Instantiation

Instantiators are very similar to the distributors from the mesh generation section. They even share the same options. The main difference is that instantiators keep the game objects intact instead of combining everything into a single mesh. For this reason instantiators are very useful for objects that rely on some kind of logic, e.g. unity lights, particle systems, or, like in the example below, domino stones.

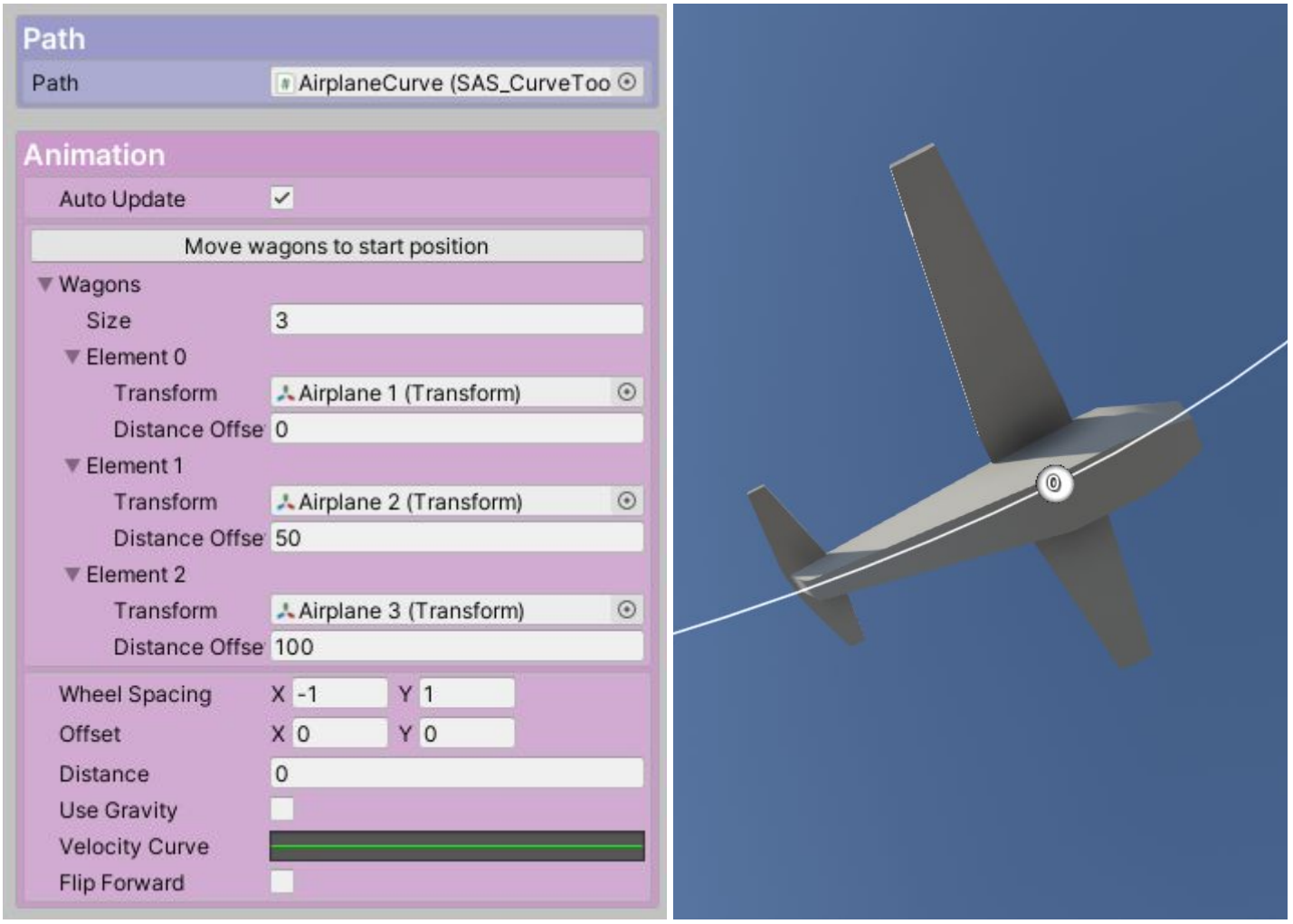


Instantiators unlike distributors don't update automatically so if you have made changes to the spline itself or the instantiator options you have to click the "Instantiate" button. Likewise if you want to remove instantiated objects click on the "Clear" button. Objects will be instantiated as children of the Mesh Game Object specified in the meshing section.

Like previously mentioned, intantiator options match those of distributors, please refer to the according documentation.

Animators

Animators animate one or more objects along a specified path.



Path - A reference to the path the object(s) should move along.
Auto Update - If enabled the object(s) will move automatically based on the specified settings. If disabled you need to take care of the movement by yourself. See the API section.
Move wagons to start position - Press this button if you want to move the object(s) to the position(s) they will have when pressing play in unity.
Wagons - You can specify multiple objects that should be animated.

- **Transform** - A reference the transform of the object that should be animated.
- **Distance Offset** - Offsets the transform from the main distance value.

Wheel Spacing - Specifies the axle distance relative to the object pivot. This is important for example if you want to have train wagon wheels stick well to the rails.
Offset - A positional offset relative to the current point on the path.
Distance - The distance along the path, this value grows or shrinks based on the speed of the object.
Use Gravity - If enabled the object will move on the path based on gravity. Useful for roller coasters.
Velocity Curve - specifies the speed of the object along the length of the path in %.
Flip Forward - If enabled the object will face the other way when moving along the path.

Notes:

- When an object reaches the end of the path it will teleport to the beginning of the path and continue from there.
- If you need special behaviours the included animator will be a good template for custom scripting.

API

Swiss Army Spline uses the namespace SwissArmySpline.

SAS_Animator

- public void **ManualUpdateSpeed**(float speed)
  Increments the distance by speed * Time.deltaTime.

- public void **SetDistance**(float distance);
  Sets the distance along the path directly.

SAS_CurveTools

- public float **totalLength**
  The total length of the entire path.

- public void **CalculateLength**()
  Calculates the total length of the entire path and writes to totalLength.

- public struct **Point**
  Holds position, rotation, forward, up, and right vectors. This is what the following function returns.

- public Point **GetPoint**(float distance, Vector2 sampleSeparation, bool posIsBetweenSamples = false)
  Returns a Point struct based on:
  - float **distance** - The distance along the path.

  - Vector2 **sampleSeparation**
    In order to calculate the vectors in the Point struct 2 samples are needed.
    sampleSeparation.x is a negative offset along the distance.
    sampleSeparation.y is a positive offset along the distance.
    Imagine the centers of the rear(x) and front(y) axles of a train wagon.

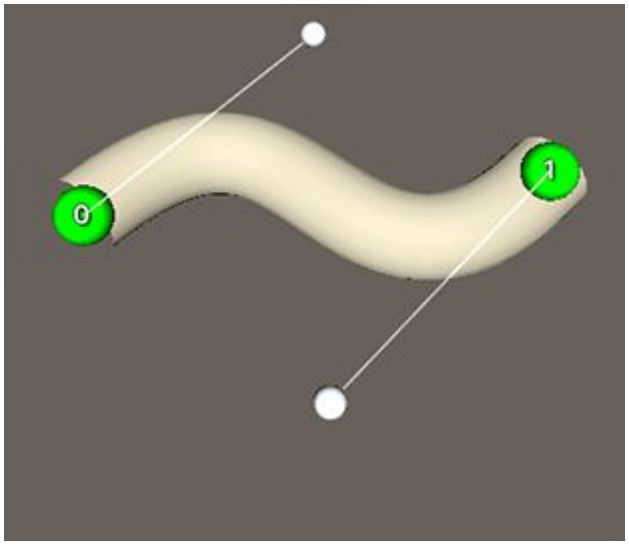
  - bool **posIsBetweenSamples**
    If true Point.position will be the average position between the 2 samples defined by the components of the Vector2 sampleSeparation. Imagine the centers of the rear and front axles of a train wagon. The train wagon itself is placed not on the rail but on a position between the axles.

- public Vector3 **GetPointSimple**(float distance)
  Returns a 3D position.
  This function is lower level than the GetPoint() function described above.

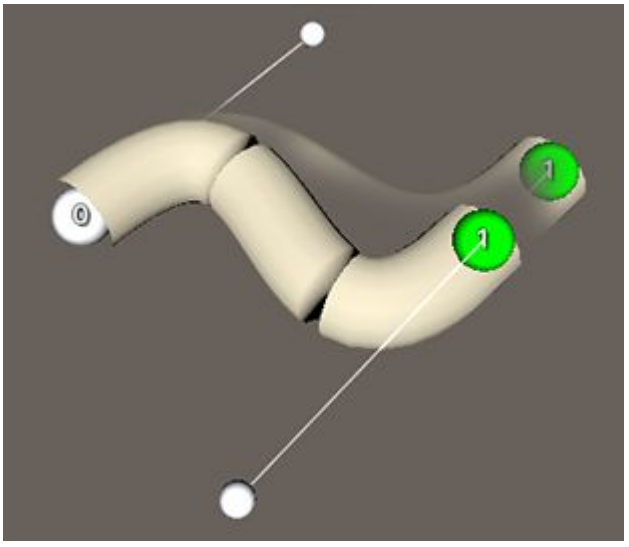
- public Vector3 **GetDerivative**(float distance)
  Returns a 3D Vector.
  This function is lower level than the GetPoint() function described above.

Known Issues & Limitations

Angles bigger than 90 degrees between points leads to twisting:



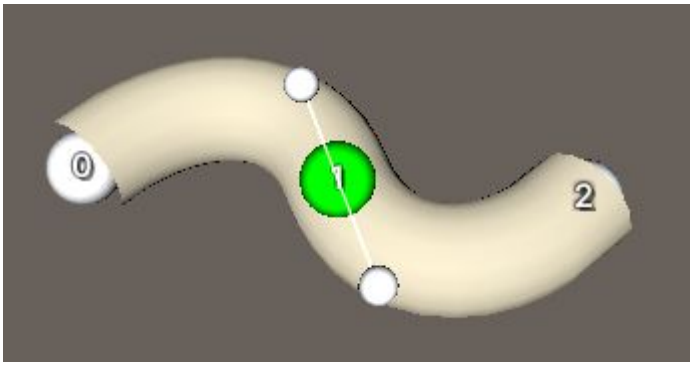
Still good.



Bad!

This happens because the method of calculating the up and right vectors along the path does not support bigger angles than 90°. It's a result of a tradeoff in order to support loopings with correct up and right vectors.

Workaround:
An in-between point has to be placed:



Fixed!

Compiling with the Profile Editor window open can open another instance of the Profile Editor window.
Workaround: Close the instances and reopen the Profile Editor.

Support

For support requests please email to: peter@becoming.at