

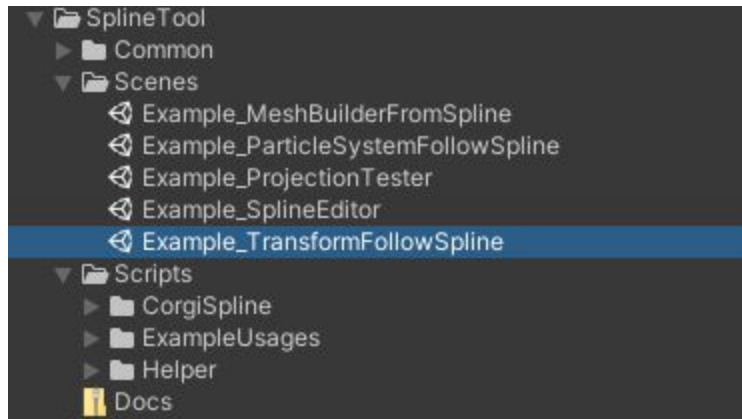


Corgi Splines Documentation

Thanks for purchasing Corgi Splines.
You can find script reference by unzipping the Docs.zip folder.
For help getting started, please keep reading.

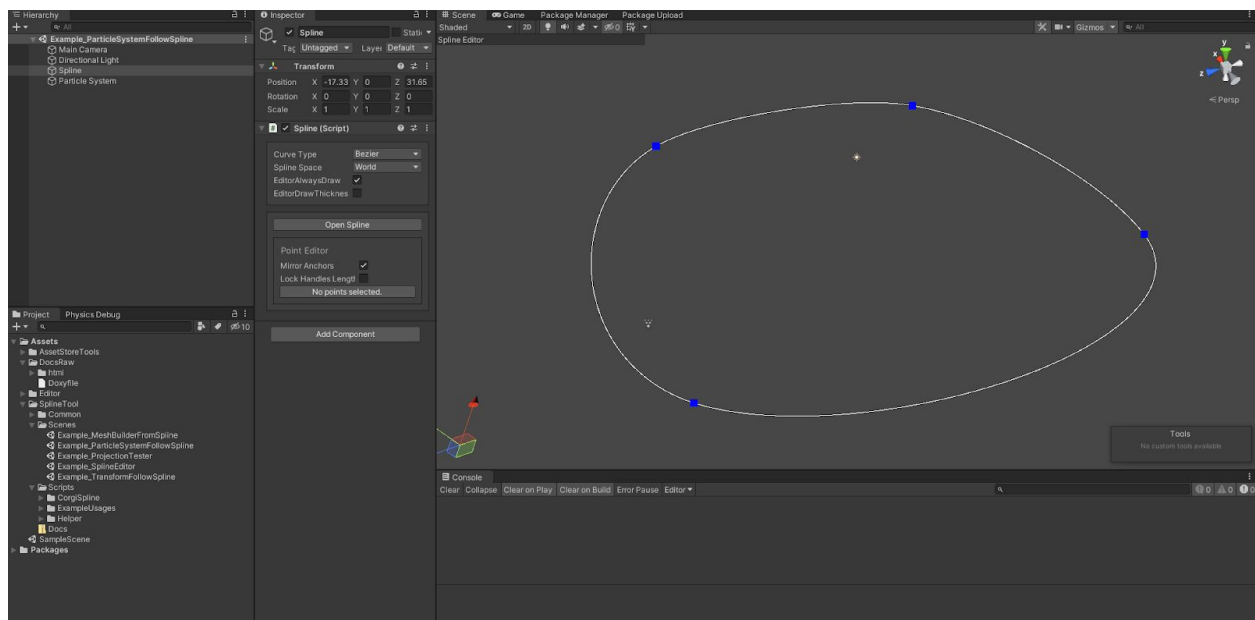
Getting started

When you import the project, you'll find the following folder structure. I've included a handful of example scenes, showing some uses of this spline tool.

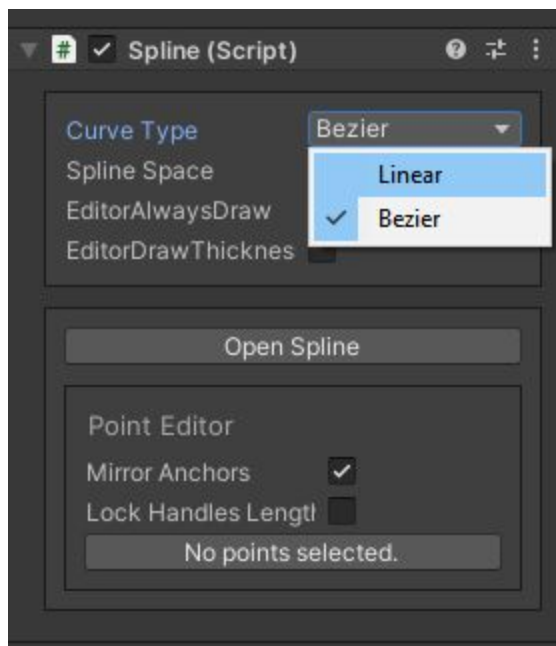


Example_SplineEditor.scene

This is probably the best place to start. Here you'll find the editor in a very simple scene, to see how everything works. To start, click on the **Spline** GameObject.



Spline Editor

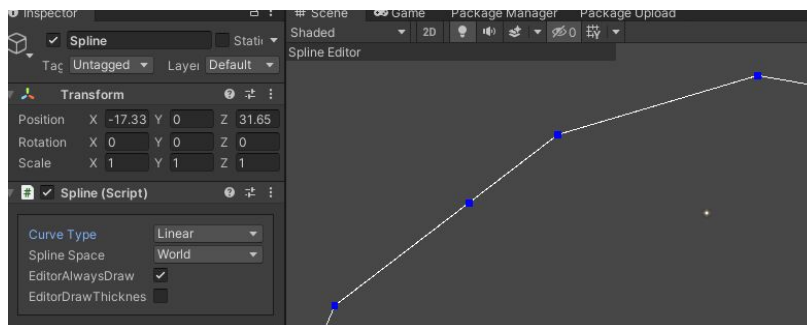


The Spline editor has two parts. The first box includes settings that are per-Spline. From top to bottom, we have Curve Type, Spline Space, EditorAlwaysDraw, and EditorDrawThickness.

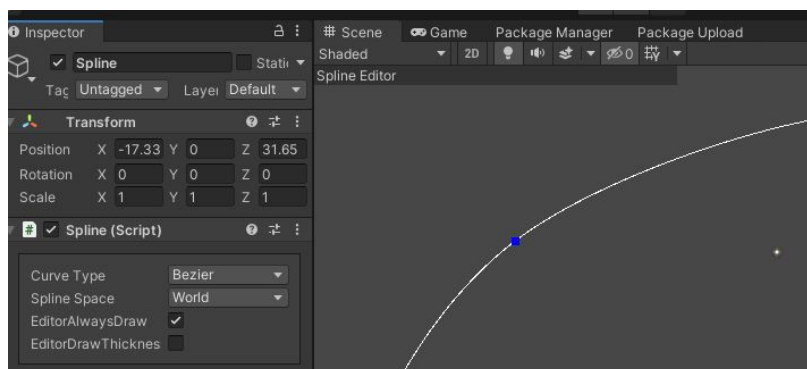
Curve Type

Corgi Splines currently supports two types of curves: Linear, and Bezier. Linear curves are what they sound like, when sampling from point to point, you'll draw a straight line. If you're looking for smoother curves, I recommend trying the Bezier curve type.

Linear spline

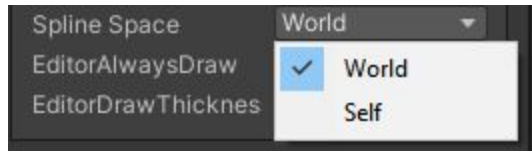


Bezier spline



Spline Space

Splines can be treated as World Space or Local (Self) space. If a Spline is set to World Space, moving around its GameObject will not move the points of the Spline. If it is set to Local (Self) Space, points WILL move and rotate with the GameObject. Try it out!



EditorAlwaysDraw

When enabled, the debug view for the spline will continue to be drawn, even while the GameObject is not selected.

EditorDrawThickness

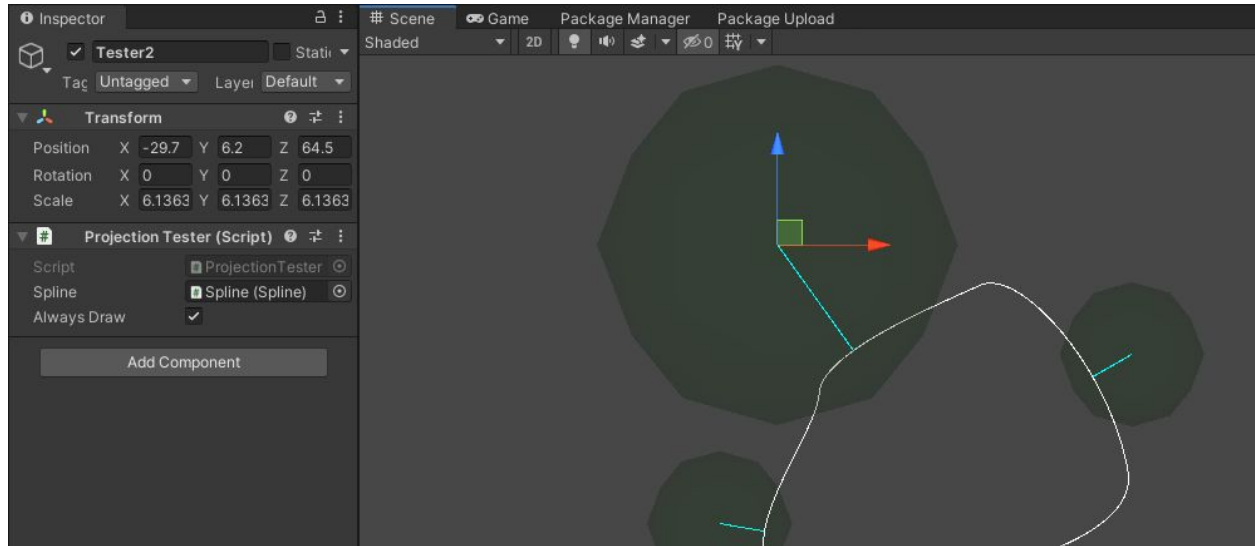
When enabled, the debug view for the Spline will also draw thickness, which is assumed to be the X axis scale between points. This is purely for debug, you can use the scale in any way you feel is useful for your project.

This is also a useful view for viewing the rotation of the spline.



Example_ProjectionTester.scene

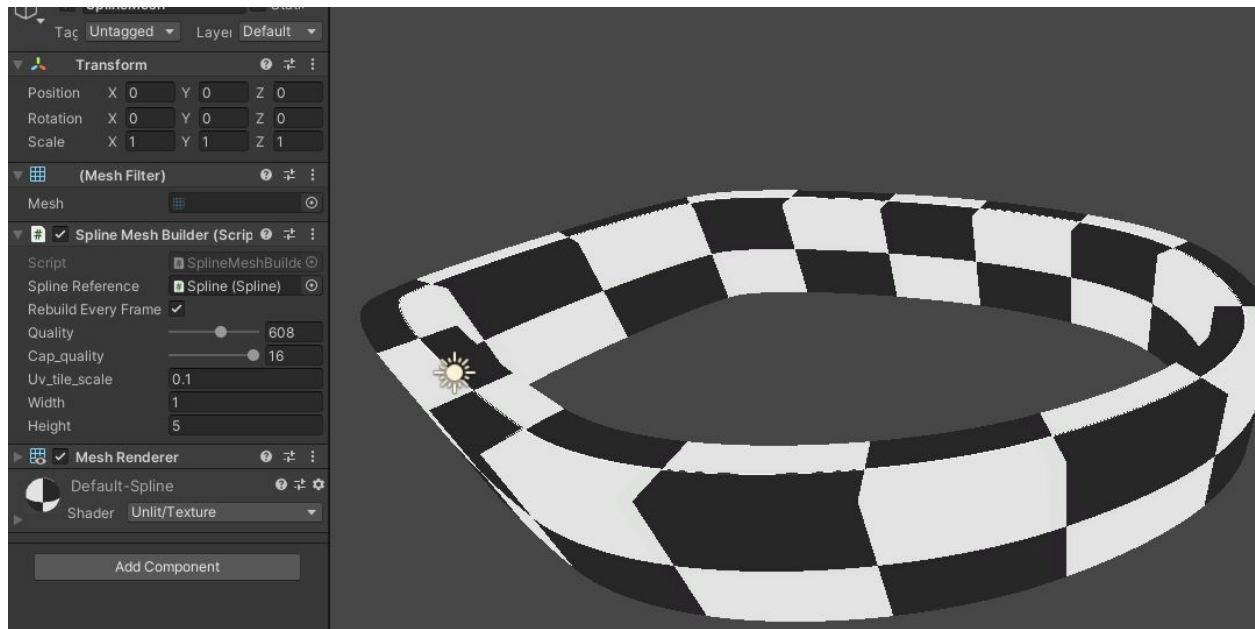
This scene includes the example helper script ProjectionTester, which shows a gizmo view as an example of how to use the Projection API in the Spline class.



You can move around the projection testers and get a feel for how it works.

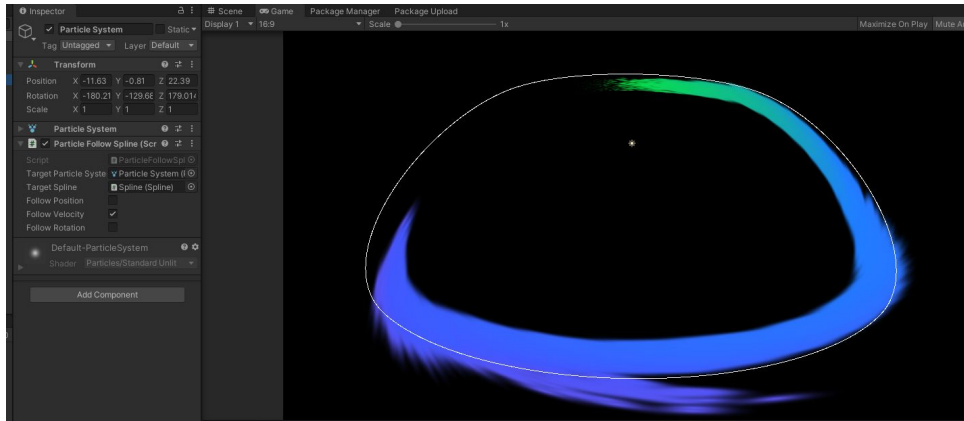
Example_MeshBuilderFromSpline.scene

This scene shows a very simple mesh builder example class. If you enter Play mode in Unity, the mesh builder will work in real time with Spline edits. It's not recommended to use this example class directly, but to look at the code and see how it works, to build off of it.

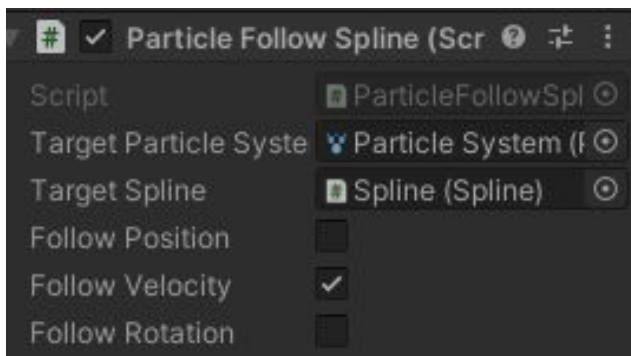


Example_ParticleSystemFollowSpline.scene

This scene shows a production-ready example script for having Particles in a ParticleSystem automatically follow a Spline.



To use the script ParticleFollowSpline.cs in your project, you'll have to manually assign the TargetParticleSystem and TargetSpline by dragging in the Unity Inspector, or by assigning it via code.



FollowPosition

This will project the position of every particle directly onto the Spline.

FollowVelocity

This will project the velocity of every particle onto the forward direction of the Spline, given the particle's current position.

FollowRotation

This will force the rotation of every particle to follow the rotation of the Spline, given the particle's current position.