Bézier Extrude Tool for Unity

Made by Álvaro Chuan Díaz-Maroto (Holowblink)

Documentation vers. 1.0

Unity vers 2022.3

Table of contents

Tool description

The Bézier Extrude Tool for Unity is a tool that allows the end user to create a Bézier curves inside the Unity's editor and extrude 2D shapes along them, while being able to manipulate the position, orientation and rotation of each individual segment of the curve.

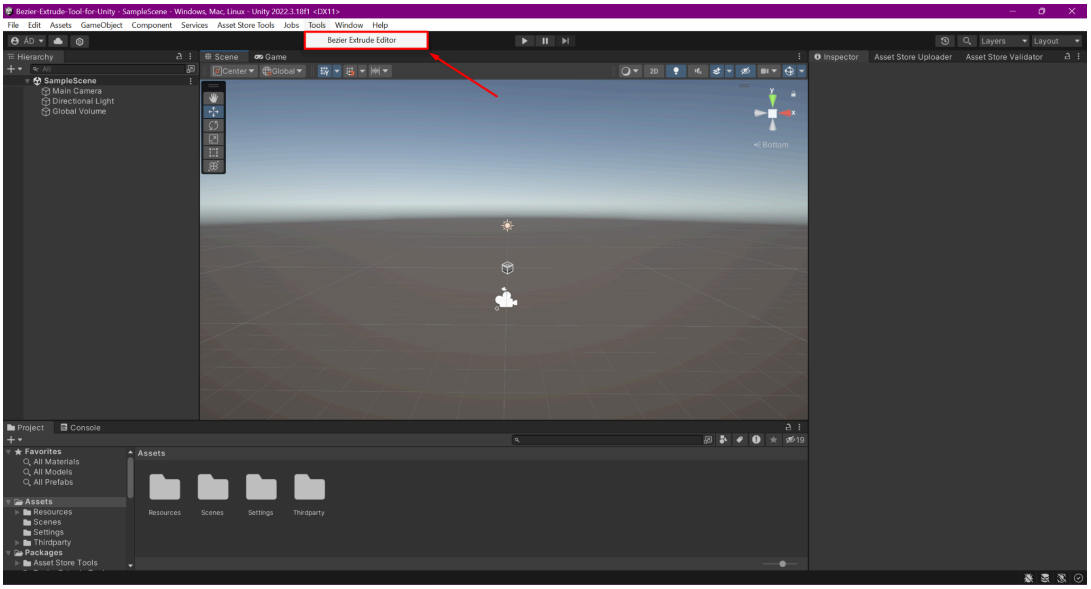
The goal of this tool is to speed-up part of the process of creating environments by creating and editing meshes on the fly without having to leaving Unity's editor.

Apart from the specs said in the first paragraph, users can modify the subdivisions of the meshes generated with the tool, save those meshes as assets, create their own shapes and modify the way textures are applied to the model.

How to use

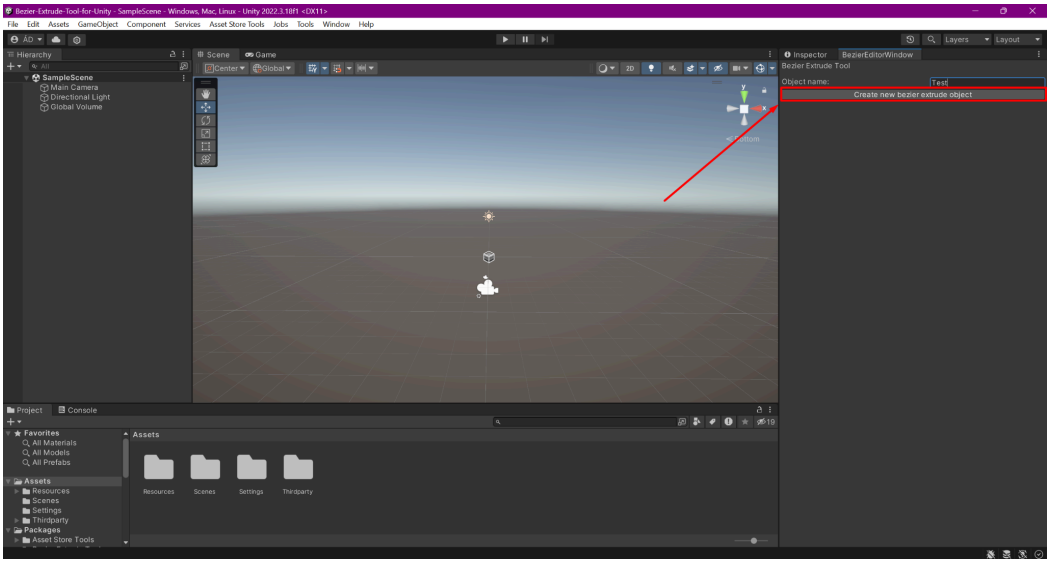
Getting started

Once you have installed the package to your Unity project, a new window will appear in your top bar called “Tools”. To access the tool, you’ll need to open the main window by clicking on “Tools” → “Bézier Extrude Tool”



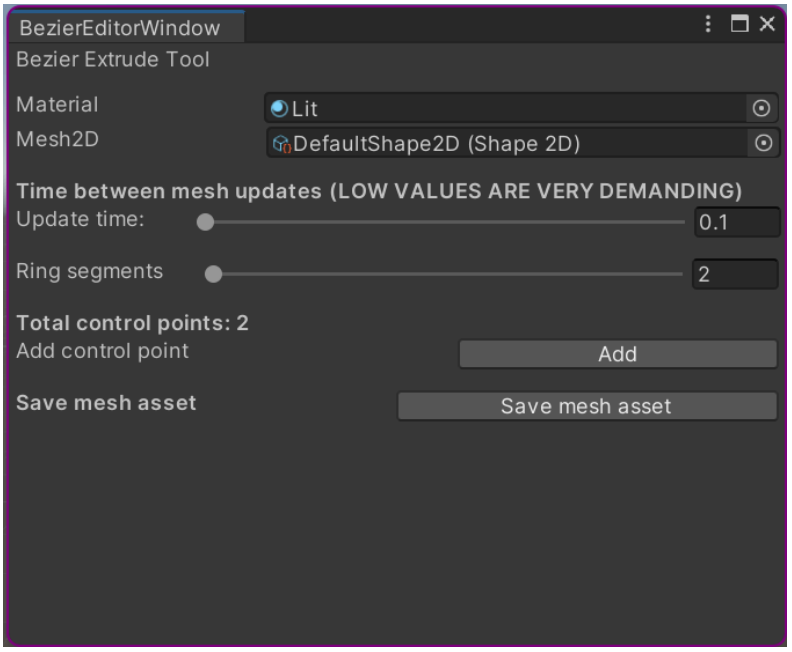
Creating your first Bézier Curve object

Now that you have the tool’s window opened, you’ll see that the only option that appears is to create a new Bézier Curve Object, in order to do that, fill the name text input with the name you want for your object and click “Create new Bézier extrude object”



Editing the object properties

Now that you have created your first object, select it inside your scene's hierarchy for the properties of the object to show on your tool window



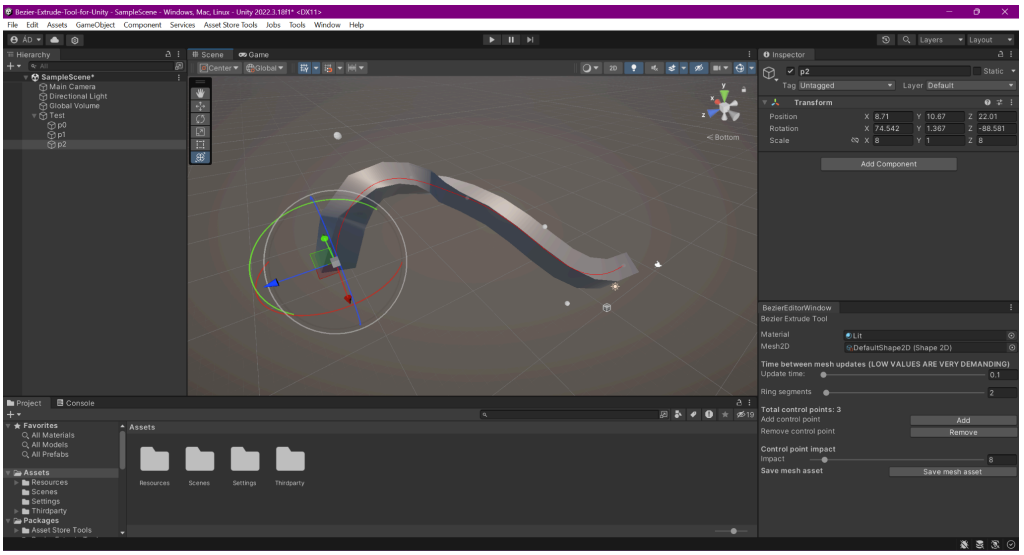
Object Properties

- **Material:** the material that your mesh will have
- **Mesh2D:** the shape2D that will be extruded along the Bézier curve
- **Update time:** the time that will pass between each mesh regeneration, modify this value based on the performance gotten while editing the meshes, higher poly meshes will consume more resources in your system, so longer wait times may be needed
- **Ring segments:** the amount of divisions of the meshes between each pair of control points (this amount is shared for all the segments of your object)
- **Control points:** Each segment of your curve is controlled by 2 control points, add or remove them to fit the number of segments that you need (the option for removing control points will only appear when the total number of control points is higher than 2)
- **Save mesh asset:** save the mesh generated by the tool into your game files. This must be done before building the game, as the tool only works inside the editor.

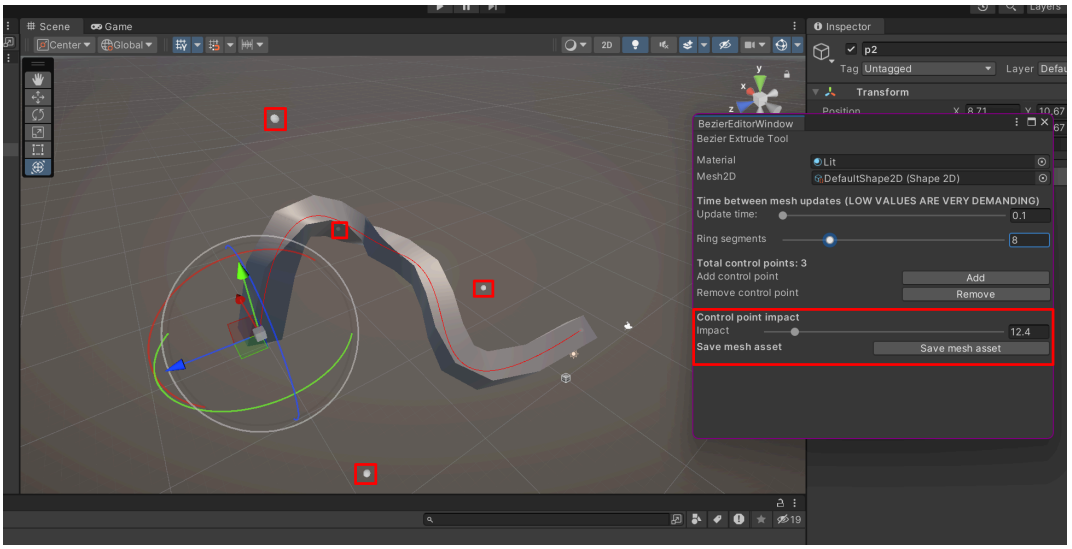
Editing your control points

In order to control the path that your mesh will be extruded through, you need to modify the position and rotation of your control points.

Control points are children of your main Bézier curve game object, you must modify their position and rotation using unity's integrated control, the same way you would move and rotate any other game object. But, you should never delete or unparent these objects from the main object, as the tool need the to work properly. If you need to get rid of a control point, use the integrated controls inside the tools window.



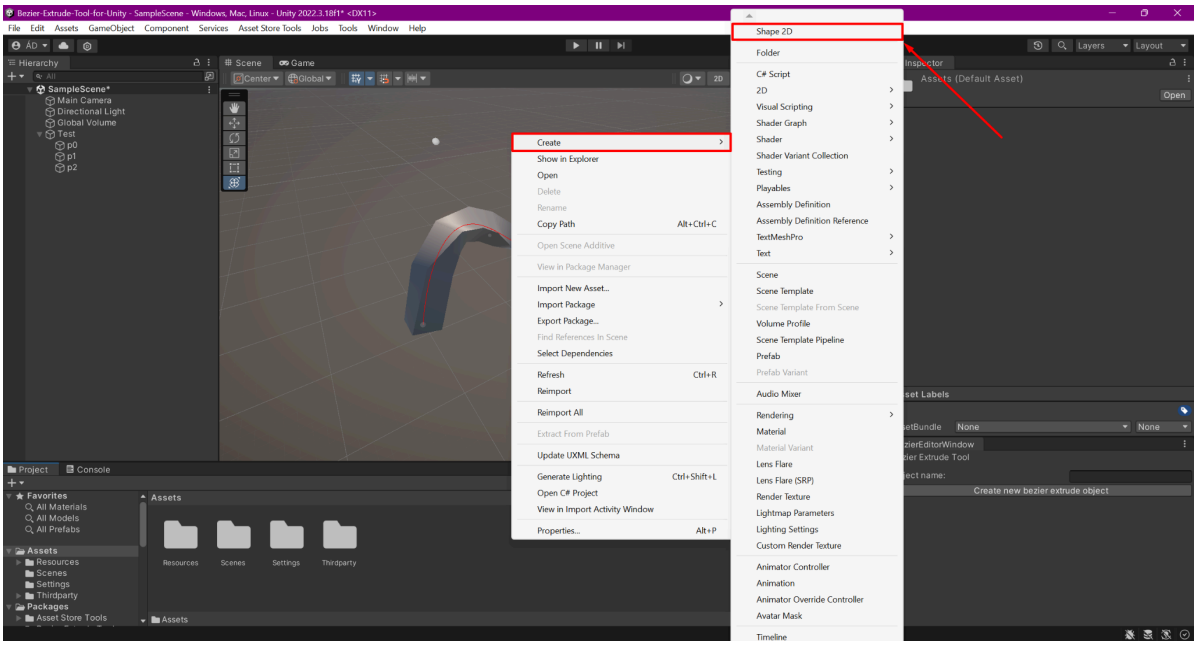
When editing a control point, you'll notice that other points appear near your control points, and that a new option is revealed inside the tool's window called "Impact". This tool uses 4 control points to generate the Bézier curve, these new 2 points are attached to the main ones and alter the curve by attracting it to them. To modify the position of these points, use the slider inside the tool's window.



Creating your own 2D shapes

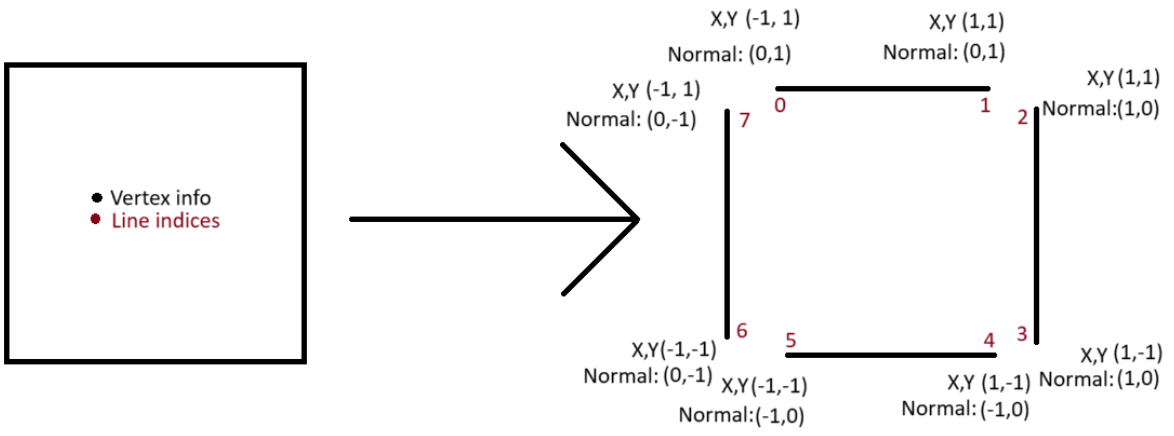
Generating a Shape2D asset

To generate a new Shape2D is as simple as right-clicking your project window and selecting “Shape2D” under the “Create” tab. “Create” → “Shape2D”



Designing your 2D shape

The way the Shape2D asset works is that it has 2 lists of vertices and lines. The vertex list contains all the vertices that will form the shape. Each vertex must have its X and Y coordinates, along as their normal Vector and U coordinate for the textures. For corner each vertex is duplicated as each pair of the forms a line. (Must be in clockwise order).



Default Shape2D example