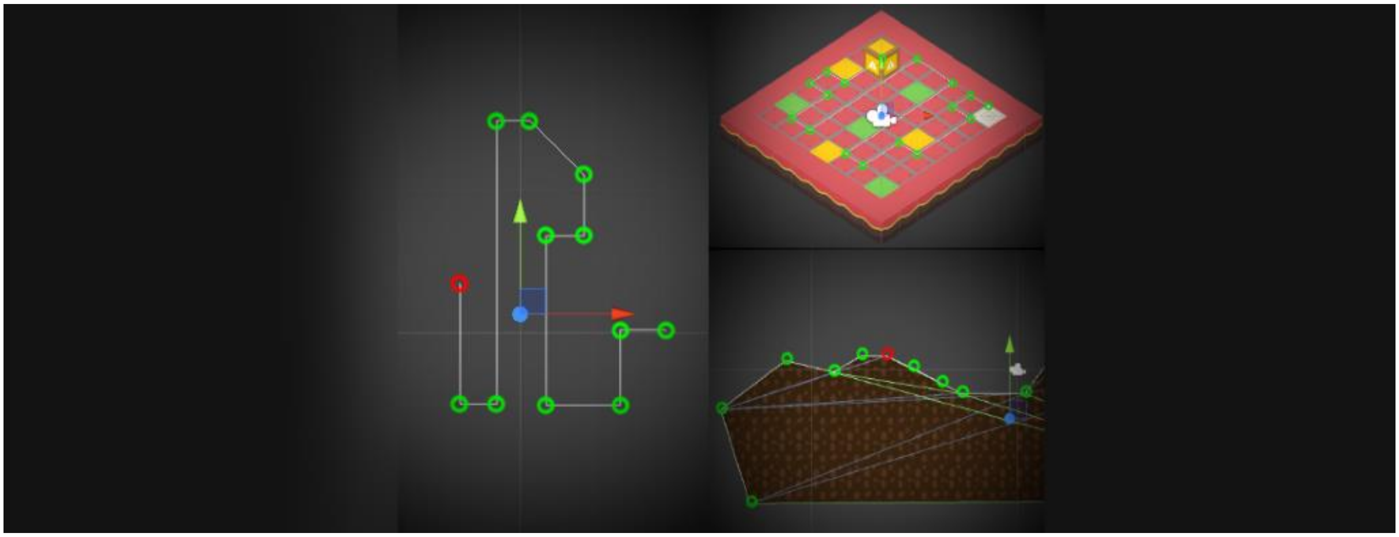
Welcome to the Unity3D Polyline Editor Wiki!

The "Unity3D Polyline Editor" is a unity editor that is used to create a polyline/path that can be used in any 2D project.



Getting Started

How to Use:

- Add the Polyline script to any game object, the polyline will initialized by two points ([-3,0,0] and [3,0,0]).
- Begin dragging your nodes in the scene view to the desired position (you may need to switch to the 2D tool instead of the default move tool to be able to drag the nodes).
- To add a new node press and hold on Shift key then move the mouse to desired location, the editor will show you the two nodes that your new node will inserted between them, click the mouse left button to add the node.
- To delete node press and hold the Ctrl key and move the mouse to the node that you want to delete, then click the left mouse button.

- Getting the edit nodes by code is very easy, you need to get the `Polyline` component of the object and it contains a public `List<Vector3>` called `Node` that contains your nodes (check the Example1_Path).

Understanding the Examples:

Example1_Path This is a very basic example of `Polyline` usage in which the `Polyline` editor used to create a path and the Cube Object follow this path. The code behind the path follow is written in the `PathMovement` script, this script has a reference to a `Polyline`script called path and we set it in the inspector. In `Update`function the `PathMovement` script check the next target point form the path and alter the cube `velocity`, then move it with this velocity.

Example2_Simple2DTerrainEditor This is more complex example in which the `Polyline` editor customized and the code changed. You will find a new version of `PolylineEditor` script called `Simple2DTerrainEditor` and a different version of `Polyline` script called Simple2DTerrain. All the changes are in Simple2DTerrainEditor in which the there is a Function called `UpdateTerrain` this function called whenever the polyline/terrain vertices changed (that happened when any node moved, added or deleted). The `UpdateTerrain` function will use a `Traingulator` to calculate the new mesh triangles and update the `Mesh` of the `MeshFilter` component assigned to the Terrain.