WireTerrain

v.1.1

Wire terrain package allows you procedurally create wireframe meshes from terrain objects. Generated meshes are native Unity meshes that are created with flag "**`MeshTopology.Lines`**". These meshes are created in 3d space.

Important!

Due to a specific rendering of meshes with **`MeshTopology.Lines`** type there is no way to change thickness (stroke) of generated lines – line always renders with the thickness of one pixel.

Getting started

There are two types of wire representation that you can create from terrain: **Contour** (`WireTerrainContours.cs`) and **Grid** (`WireTerrainGrid.cs`)

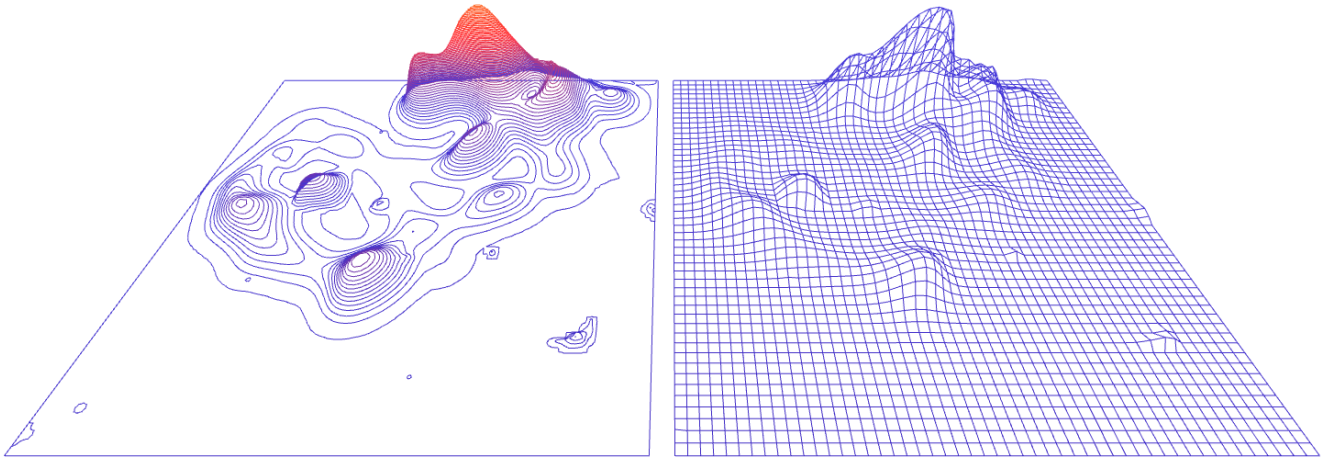

Fig.1 Contours and grid wire terrains

To create a wire representation of terrain, you can select a GameObject on which you want to place a generated wire mesh and add one of the WireTerrain scripts (`WireTerrainContours.cs` or `WireTerrainGrid.cs`) either by selecting from menu "`Component/WireTerrain`" or dragging script from "Scripts" folder in package folder. In order to see a generated mesh for `WireTerrainGrid.cs`, you must add MeshRenderer and MeshFilter components to this or other GameObject and set some parameters in WireTerrain scripts to non-zero values. `WireTerrainContours.cs` script generates children to the transform to which it attached with MeshRenderer and MeshFilter components.

You can save generated mesh using script "`SaveMeshButton.cs`" – just add this script to GameObject with generated mesh and press button "Save Mesh" in script inspector.

There are experimental scripts (`WireMeshContours.cs and WireMeshGrid.cs`) which can generate grid and contours not from Unity terrain object, but from mesh objects. They have similar options as they corresponded terrain scripts.

Texturing

For generated meshes, you can generate UV coordinates for texturing along height (Y direction). UV coordinates generated only for “U” direction, “V” coordinate always have value zero. The distribution of texture can be changed through material texture property “`Tiling X`”.

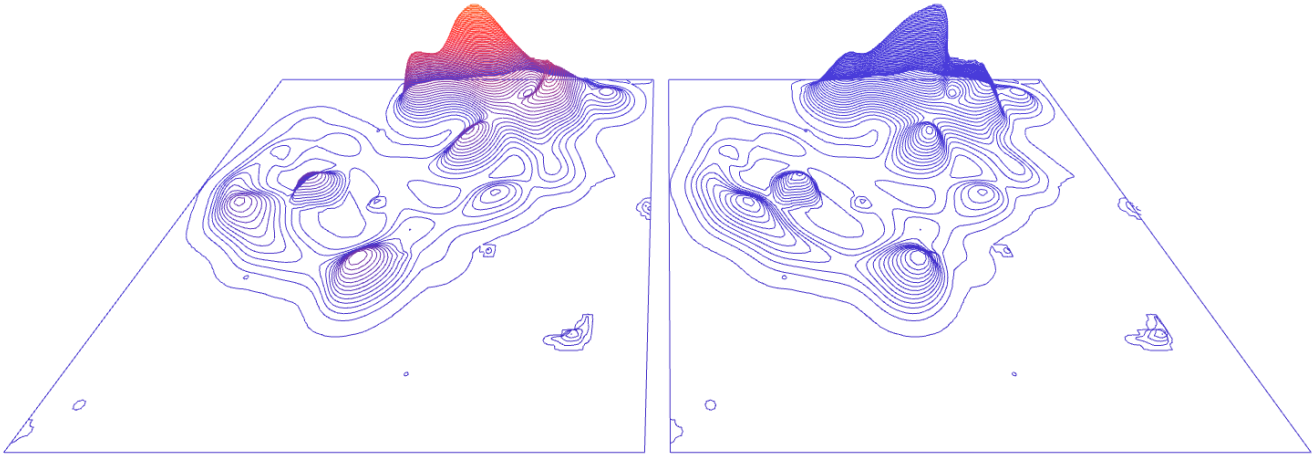

Fig.2 Left mesh with UV coordinates and material with gradient texture, right rectangle without UV coordinates.

Scripts

1 Wire Terrain Contours

parameters:

sourceTerrain – Terrain from which we creating wire representation
targetMesh – MeshFilter in which we store generated wire mesh
heightStart – Start Y coordinate value for contours (first contours would lie on this height)
heightStep – Distance between planes that dissect terrain to produce contours
cellCountX – Number of cells in X direction for the grid with terrain heights
cellCountZ – Number of cells in Z direction for the grid with terrain heights
flatten – If "true" all contours belong to one plane on zero height
drawBound – If "true" draws a rectangular frame with the size of Terrain on zero height
generateUV – If "true" generates UV coordinates for wire mesh vertices. U coordinate mapped to vertex height, V coordinate is zero
optimize – If "true" merges coincident vertices, it has heavy computation, so uncheck it if you intend to generate the mesh in runtime

cellCountX and **cellCountZ** - corresponds to the coarseness of produced contours: higher values produce smooth contours with more vertices, and lower values mesh with fewer vertices, but not so smoothed contours.

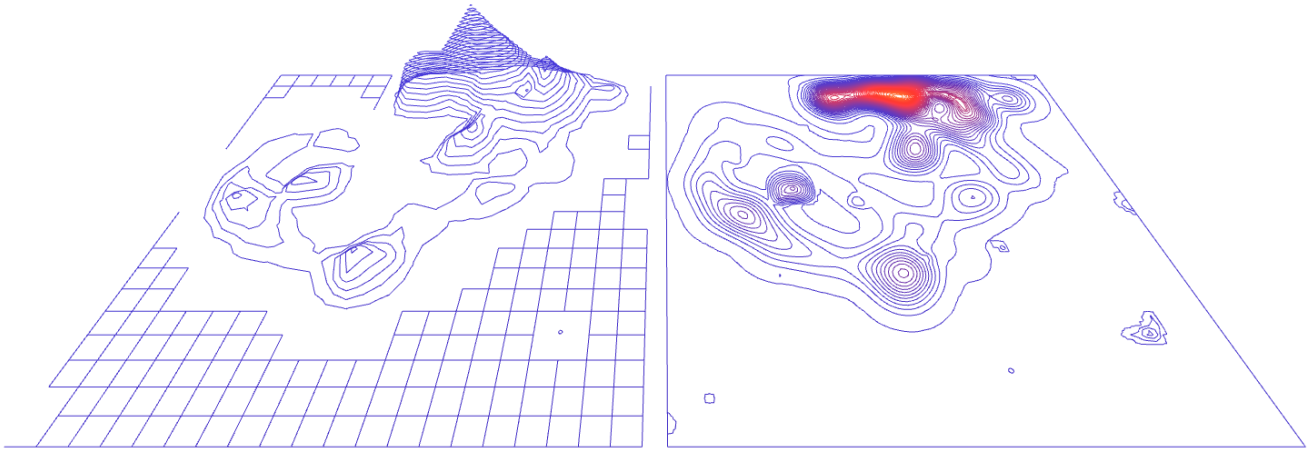

Fig.3 Contours wire meshes with different parameters.

HeightStart: **0**
HeightStep: **5**
CellCountX: **20**
CellCountZ: **20**
Flatten : **False**
DrawBound : **False**
GenerateUV: **False**

HeightStart: **0.3**
HeightStep: **3**
CellCountX: **100**
CellCountZ: **100**
Flatten : **True**
DrawBound : **True**
GenerateUV: **True**

2 Wire Terrain Grid

parameters:

sourceTerrain – Terrain from which we creating wire representation
targetMesh – MeshFilter in which we store generated wire mesh
cellCountX – Number of cells in X direction for grid with terrain heights
cellCountZ – Number of cells in Z direction for grid with terrain heights
addPolygonMesh – If "true" generates two submeshes: one with triangle mesh and other with line mesh. You must provide two materials in materials field in MeshRenderer component in order to see both submeshes.
generateUV – If "true" generates UV coordinates for wire mesh vertices. U coordinate mapped to vertex height, V coordinate is zero

cellCountX and **cellCountZ** - corresponds to the coarseness of produced grid: higher values produce smoother lines with more vertices, and lower values produce mesh with fewer vertices, but with the more jagged view.

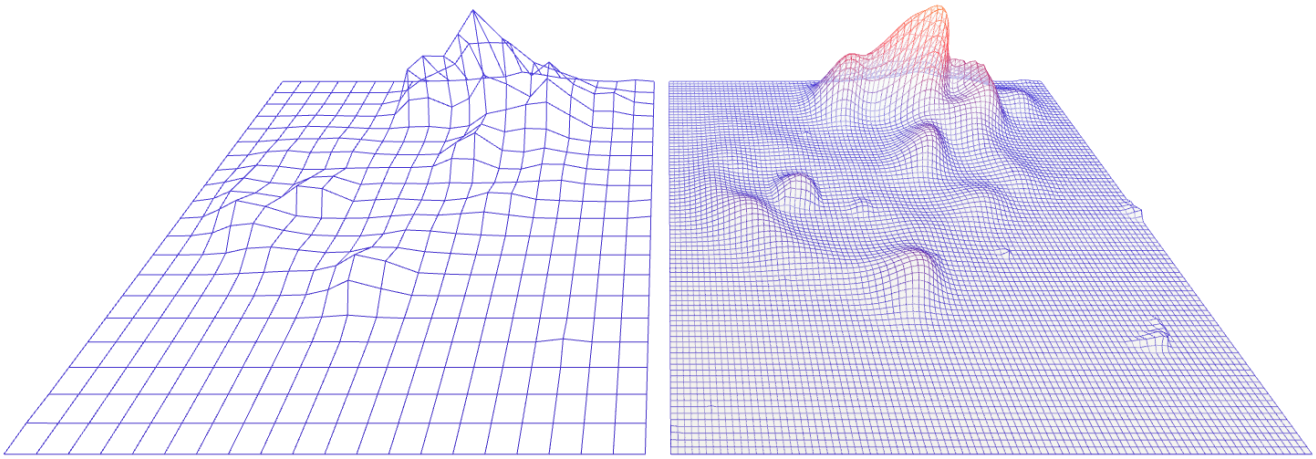

Fig.4 Grid wire meshes with different parameters.

CellCountX: **20**
CellCountZ: **20**
AddPolygonMesh: **False**
GenerateUV: **False**

CellCountX: **100**
CellCountZ: **100**
AddPolygonMesh: **True**
GenerateUV: **True**

WireTerrain: http://u3d.as/Tnr
Thanks for purchasing my asset.
doubtfulpixel@gmail.com