MicroSplat

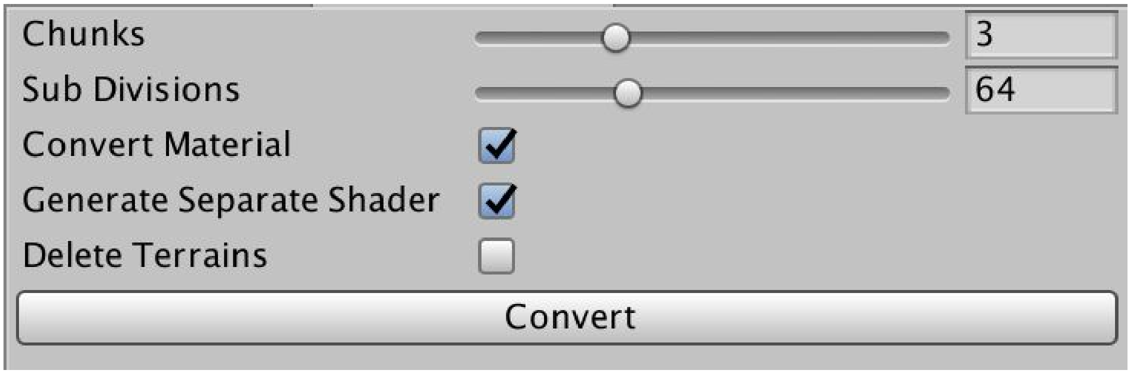
Mesh Terrain Documentation

Overview

The MicroSplat Mesh Terrain module allows you to convert Unity Terrains to meshes, using MicroSplat on the resulting meshes. This is commonly used for low detail distant rendering where Unity Terrains are too heavy, but can also be used if your generating procedural meshes and avoiding the Unity Terrain system.

Quick Start

To begin, make sure all the terrains you want to work with are converted to MicroSplat and have unique names, as those names will be used to generate textures and mesh files. Select all the terrains that you want to convert in the scene, and open the terrain conversion tool (Windows->MicroSplat->Terrain To Mesh). Select your options and press the Convert button. The data (meshes and splat maps) will be written into the MicroSplatData directory with the rest of the terrains data.

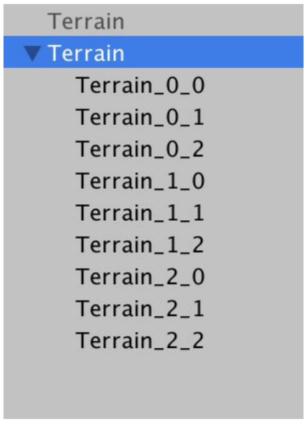


Chunks 3
Sub Divisions 64
Convert Material ☑
Generate Separate Shader ☑
Delete Terrains □

- Chunks
    - How many times to subdivide the terrain into unique meshes. A value of 2 will split each terrain into a 2x2 set of meshes
- SubDivisions
    - How detailed the resulting mesh should be. A value of 64 will create each chunk as a 64 by 64 grid of mesh data.
- Convert Material

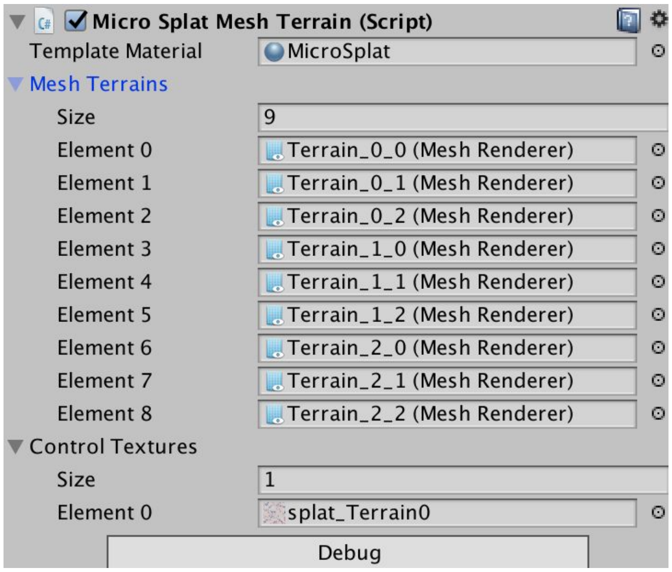
    - When checked, this will convert your existing terrain material into a Mesh based workflow, as well as export all the Splat Map textures from the original terrain. You usually want this on, otherwise you'll just have an untextured mesh.
- Generate Separate Shader
    - Enable this if you plan to have your mesh terrain use a different shader setup than the original terrain- perhaps if you are swapping them based on distance. This will create a MeshTerrain directory inside your MicroSplatData directory with another MicroSplatData directory inside of it, where it stores the meshes and textures for the mesh terrain. It will also create a copy of your material, shader, and keywords object and assign them for use on your new mesh terrains, allowing you to change options without affecting the original terrain. Your original textures and Per Texture Properties will still be used, so that if you update those things they will be represented here as well.
- Delete Terrain
    - Deletes the terrain object when done

Once done, you may hide the original terrain and see the results. In this example I set my chunks to 3, giving me a 3x3 grid of meshes for my terrain. A parent object is created with the same name as the original terrain, and mesh objects are added below it for each chunk.



Terrain
▼ Terrain
Terrain_0_0
Terrain_0_1
Terrain_0_2
Terrain_1_0
Terrain_1_1
Terrain_1_2
Terrain_2_0
Terrain_2_1
Terrain_2_2

Note that the MicroSplatMeshTerrain component is on the top level "Terrain" object, not the sub objects. This is so one material can be shared across all the sub objects.



☑ Micro Splat Mesh Terrain (Script)

Template Material	MicroSplat
Mesh Terrains	
Size	9
Element 0	Terrain_0_0 (Mesh Renderer)
Element 1	Terrain_0_1 (Mesh Renderer)
Element 2	Terrain_0_2 (Mesh Renderer)
Element 3	Terrain_1_0 (Mesh Renderer)
Element 4	Terrain_1_1 (Mesh Renderer)
Element 5	Terrain_1_2 (Mesh Renderer)
Element 6	Terrain_2_0 (Mesh Renderer)
Element 7	Terrain_2_1 (Mesh Renderer)
Element 8	Terrain_2_2 (Mesh Renderer)
Control Textures	
Size	1
Element 0	splat_Terrain0

The MicroSplatMeshTerrain component should be very familiar to you if you've worked with the MicroSplatTerrain component. However it contains a list of all the terrain objects that render using that material, as well as the splat maps used for them.

Modules

All module features are supported on Mesh Terrains. However, it must be noted that these modules often rely on the underlying topology being a height field, so modifying the mesh may break certain features which rely on this height field data.

Additionally, some of the tools which generate data, like per-pixel normal maps, cavity maps, the various painting tools, etc, are only available when working on a Unity Terrain. As such, it's best to keep the terrain around to generate this information, only removing it before builds if it's no longer needed.

Notes

The default splat maps output by the converter are compressed, while Unity uses uncompressed splat maps. This saves significant memory but might ever so slightly alter the texturing. You can set them back to uncompressed if you'd prefer.

Using the Per-Pixel Normal option on your terrain can be a massive quality difference when creating low resolution mesh terrains, because they will be lit like high resolution terrains.

Material LODs

When rendering on Unity terrains, MicroSplat outputs two separate shaders. The first has things like tessellation and other close up effects enabled, while the second does not. When those effects are no longer noticeable, the shader is swapped. This can be a massive win for performance. As meshes don’t have such a system built in, it’s often useful to generate a unique shader for your meshes if you know they are only going to appear in the distance. Use the Generate Separate Shader option to do this for you automatically when converting terrains.

Once generated, you can disable any expensive effects you don’t want to pay for in the distance, such as tessellation, detail noise, etc.

If you are using the anti-tiling module and using distance resampling, and your mesh terrains will only be used in the distance, you can disable this effect on your mesh terrains and multiply the UV scale to match, saving you significant performance.