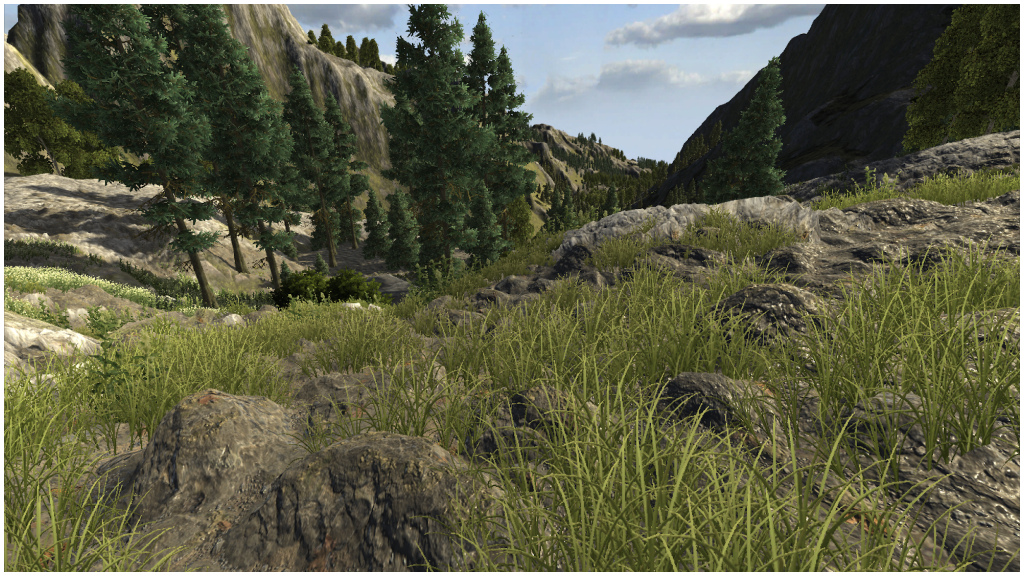
MicroSplat

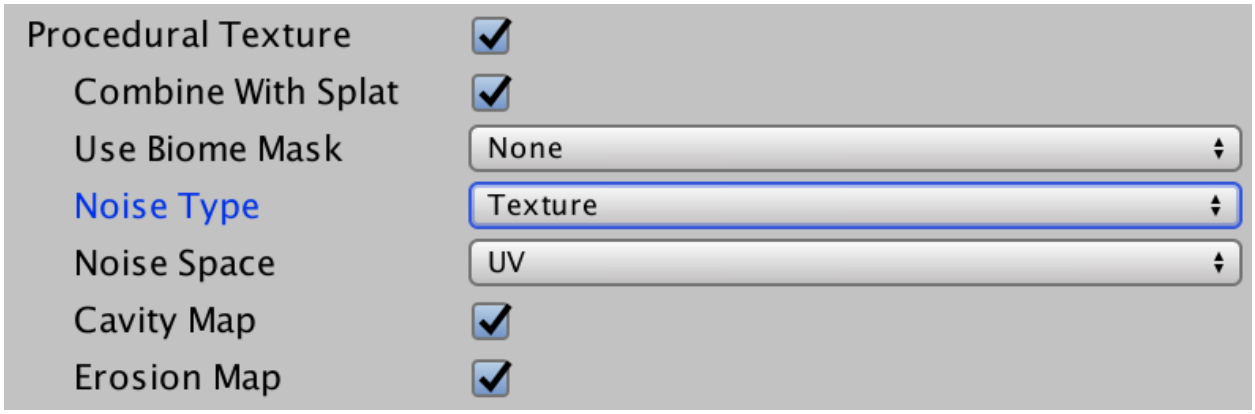
Procedural Texturing Documentation



Overview

The procedural texturing module of MicroSplat allows you to texture your terrain based on rules. The texturing is executed in the actual terrain shader, so no splat map textures or painting is needed.

Shader Features



Enable the Procedural Texture option in the Shader Generation section of the material. The options are:

Combine With Splat

When this option is enabled, both traditional splat maps and procedural texturing are used. There are two “Alpha Modes” available:

Replace Texture Index

A single texture in your array is designated as the mask; wherever that texture is painted, the procedural texturing will be used- and if any other texture is painted, it will override the procedural texturing. This allows you to procedurally texture your terrain, but then paint roads or other custom features in where you want them. The texture index used as the procedural mask is chosen in the material section, described below. Note that you can still use the chosen texture in your procedural texturing, but painting that texture on the terrain will effectively mean “Use Procedural Texturing here”.

Preserve Texture Indexes

In this mode, you can choose up to 4 texture indexes to preserve. The use case here is that you’ve used some 3rd party tool to paint roads and paths around your scene,

but now you'd like to retexture the terrain procedurally. With this mode, you can select the road and path textures as the preserved textures, and everywhere else will get procedurally textured.

Use Biome Mask

When this is enabled, an RGBA texture can be used to mask out procedural layer rules in given areas. The RGBA channels of the texture correspond to weights of four biomes, and a mask value is used to weight each of the rules. For instance, when the mask is set to 1, 0, 0, 0, that ruleset will only be applied in areas where the biome mask has red.

Noise Type

Noise type lets you select between texture based noise and procedural noise. This is computed per layer. In texture mode, a small noise texture is used, which means tiling can occur- but this is faster to lookup than using procedural noise. Each channel of the texture can hold a different noise, so the signature of noise applied to height can be different than noise applied to slope. The data is stored as noises for Height (R), Slope (G), Cavity (B) and Erosion (A).

In procedural mode, a 3 octave FBM based on world position is used. This gives a noise which is similar in looks to the cloud filter in photoshop, and the same noise is used for all the channels.

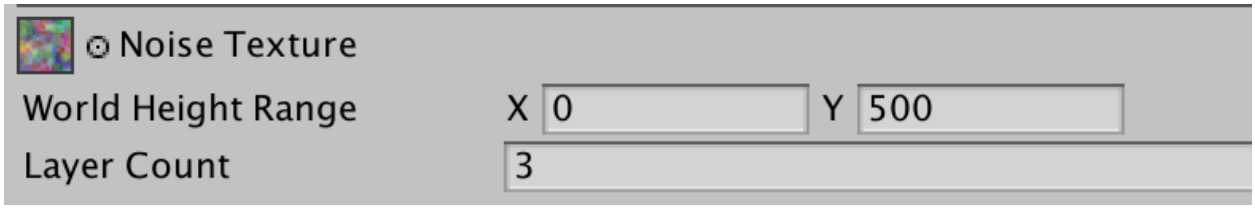
Noise Space

Noise Space is only available when textured noise is used. Noise Space lets you use either UV space, world space, or triplanar to compute noise values. Triplanar noise requires 3 samples of the noise texture instead of one.

Cavity and Erosion Map

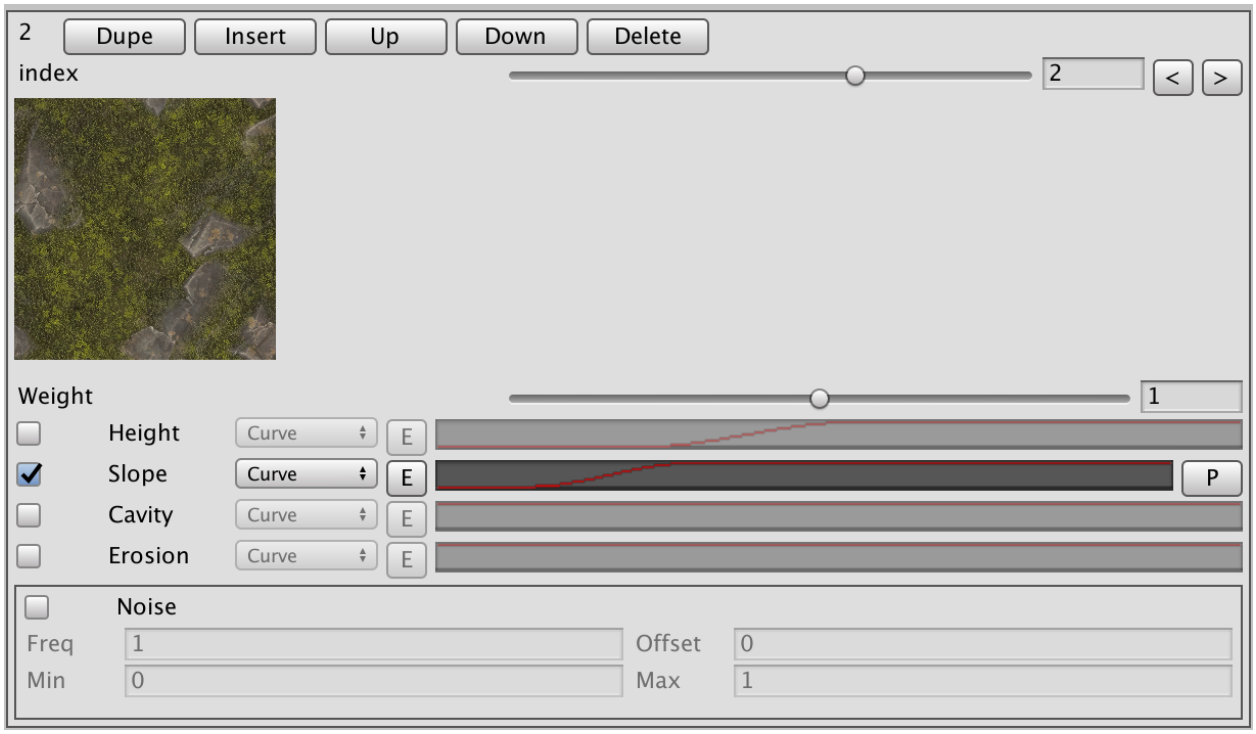
Cavity and Erosion Map allow you to pregenerate a special map containing cavity and erosion information, which can be used to filter terrain texture weights. These two options require a precomputed map, which can be computed by MicroSplat, or supplied by a third party program.

Properties



Once enabled, a Procedural Texturing section will be added to the material settings. At the top are the following options:

- Procedural Index. This appears if Combine With Splat is on, and allows you to determine which texture means “Show Procedural”.
- Noise Texture. The noise texture allows you to rough up your ruleset with noise. One noise texture is used across the whole system for speed.
- World Height Range. Height based rules need to know the range of the terrain, so you can enter your minimum and maximum world height for your terrain(s) here
- Layer Count. This controls how many rules you have in your system. Each rule is processed a bit like a photoshop layer, where layers on top have priority over layers below them.



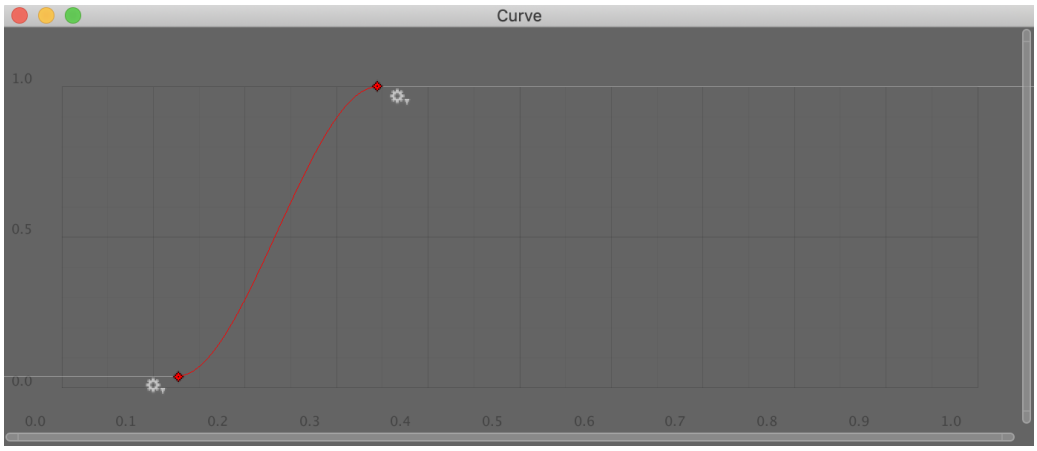
Each layer has an interface like the one pictured above. At the top, you can choose the index of the texture you are going to apply. Remember, layers are ordered like photoshop layers, so it is suggested to have the last layer be the “background” layer, with higher layers deciding where they should have weight. You can move a layer up or down with the button

The layers properties include:

- **Weight**. This is the overall opacity of the layer.
- **Height**. When enabled, you can use a curve to determine how much weight the layer should have at what heights. Remember to set the height range. Weight is defined on the vertical axis, height is on the horizontal.

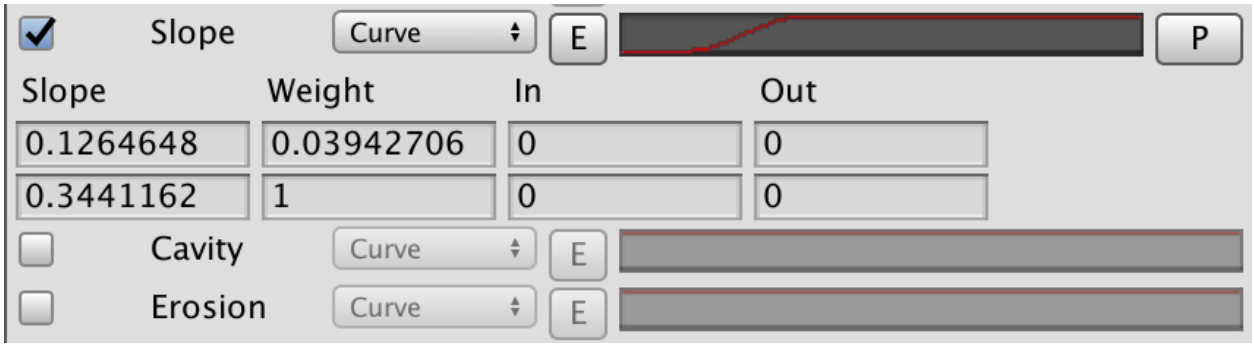
- **Slope**. When enabled, lets you define a curve to determine which slopes the layer should appear on. Weight is defined on the vertical axis, while angle is defined on the horizontal axis. The angle goes from 0-1, with 0 being down and 1 being up.
- **Cavity**. When enabled, lets you define a curve to apply weight based off of the cavity map.
- **Erosion**. When enabled, lets you define a curve to apply weight based off erosion data.
- **Noise**. Noise modifies the weight of the layer. You can use the frequency and offset to scale or move the noise, while the min and max parameters remap the noise output. You can use the remapping to make the noise faint (0.0 to 0.2), increase contrast (0 to 5), or even be subtractive (-1 to 0).

Each of the filtering controls can be set to several modes. The curve mode gives you a curve to define where the texture appears.



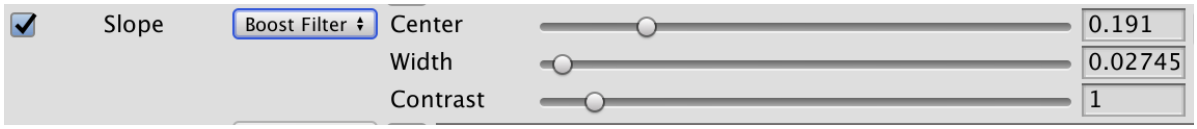
The Y axis on the curve represents the weight. And the X axis represents whatever value we are adjusting (height, slope, etc). Note that all values are mapped in a 0-1 range. So if this curve was applied to height, it would allow texturing on the upper portions of the world, and filter it out on the bottom.

The E button will open a curve spreadsheet editor, allowing you to type in precise values for the keyframes.



The curve sheet editor allows you to adjust values with higher precision than the curve editor would allow. Note that you can easily do math right in the editor controls, which can be useful if you are trying to get an exact height value converted into a 0-1 range.

You can also change the filter mode from Curve to one of several other controls:



All of the non-curve filter modes work similar to an audio EQ or pass filter. You select a center point, with, and contrast for the filter. The four modes control how these are applied:

- Boost Filter
  - In this mode, the weight will start and end at 0, and be one at the center point.
- High Pass
  - A high pass filter starts at 0, then ramps up to 1 by the center position, then stays at 1.

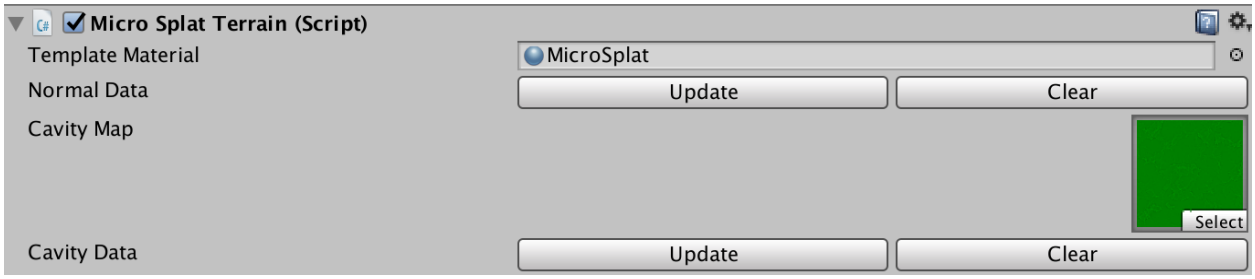
- Low Pass
    - A low pass filter starts at 1 and stays at 1 until the center position, then ramps down to 0.
- Cut Filter
    - The cut filter is the inverse of the boost filter, it starts and ends at 1, and interpolates to 0 at the center position.

Biome Mask

When the biome mask is enabled, you can supply a texture to filter each rule by biome. In Splat Map mode the texture is an RGBA texture, allowing for four biomes areas. Each rule has 4 values which determine the weight for that rule set on that biome. For instance, if you set the value for a rule to 1, 0, 0, 0, then the rule's weight will only be applied where there is red in the biome mask. Essentially, these values are multiplied together to produce a weight.

In exclusion mode, values like 1,1,0,0 can be used allowing for 16 possible biomes.

Cavity Map



Procedural texturing based on height and slope can be performed without any additional data. However, when Cavity Map or Erosion Map are enabled, a precomputed map is required as generating this data in the shader can be costly. Many landscape programs can output these extra maps, but MicroSplat has a generation system included. Once enabled, simply select the

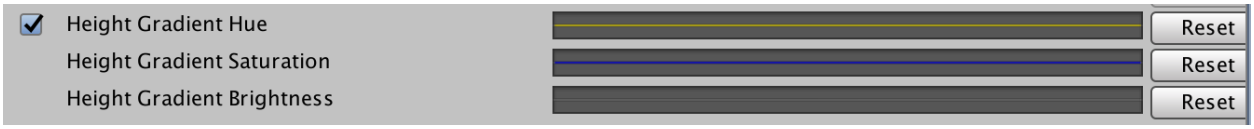
MicroSplatTerrain component and press ‘Generate’ on the Cavity Data line. Once a map is generated, should the terrain change, you can press update to regenerate the map.

You may also provide your own map. Cavity is read from the G channel of the texture, and Erosion is read from the A channel. Note you do not have to use a cavity and flow map, you could use whatever map you want and filter weights based on that texture data.

Gradient Heights

This module also adds a special per texture property for Gradient Heights. You can find this in the per-texture property section. Once enabled, each texture can have a gradient tint based on the height of the terrain. In this case, the grass is being made more lush at the bottom, and darker at the top. Note that this feature does not require you to procedurally texture your terrain, and works fine with traditional splat mapping.

Gradient HSB



The gradient HSB controls are similar to the gradient tint, but allow you to adjust the Hue, Saturation and Brightness of a texture over the height of the terrain. This can allow for a great deal of control over your terrain’s coloring, for instance making grass get brown and desaturated as you get to the top of the hills, etc. Like the gradient height tint, this is also available when using traditional splat mapping.

Workflow Advice

If you are familiar with Photoshop layers, then the way weights work in this system will make a lot of sense. Create several layers, set their weights to 0, and set the last ones weight to 1. Select something universal for this last texture.

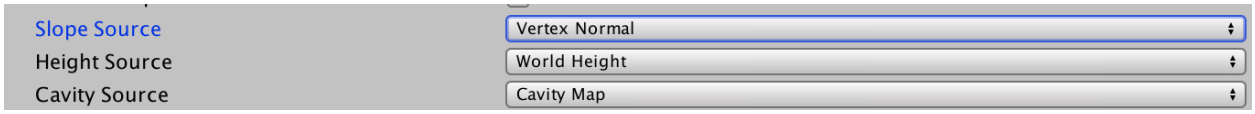
Then go through the other layers, choosing textures for them, and raising their weights to 1. Then enable noise or a filter curve and adjust the parameters to get the texture to appear where you want it.

Baking

Computing splat maps at runtime adds significant cost to the shader, so if you don't need the terrain to dynamically be textured at runtime, you can bake out the result to traditional splat maps. To do so, select the MicroSplatTerrain component and press "Bake Procedural To Terrain" under the "Procedural Baking" rollout. You can then turn off procedural texturing. Note that the texturing will not be exactly the same, because previously it was being computed on each pixel being drawn, and now it is being drawn at the splat map resolution.

Special Features

If you are working with the Mesh module, and using the Combined shader type, several new options are available in the Procedural Texturing module. While the procedurally texture module can already be used to texture meshes, these options allow you to do it with a smaller scale context than something like a terrain, working off your normal, height and AO maps like you would a terrains large scale structure.



Slope Source

This can be either Vertex Normal or Combined Normal. When in Combined Normal mode, the normal map from the combined map normal is used to compute the slope instead of the Vertex Normal.

Height Source

This can be either World Height or Combined Height. When in Combined Height mode, the height map from the combined map shader is used as the height value, instead of the world position.

Cavity Source

This can be either from Cavity Map or Combined AO Channel. When set to combined AO channel, the ao map from the combined map shader is used to evaluate cavity rather than an explicit cavity map.