Procedural Terrain Painter

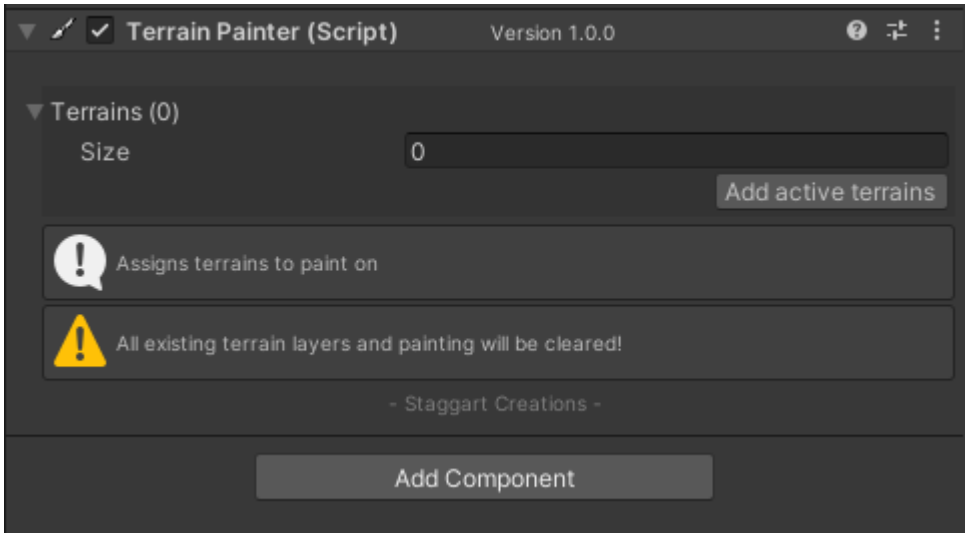
Thank you for downloading this package, it is something I created for personal use, so hope it also proves useful to you! If you are familiar with it, please consider leaving a review!

Table of Contents

Getting started

You should already have one or more terrains setup at this point, created manually or with tools such as Gaia, MapMagic, World Creator or Terra. If you're already familiar with tools like Photoshop or the Substance suite, using this tool will feel like a second nature.

To start modifying the terrain layer distribution create an empty GameObject and go to **Add component->Terrain Painter**

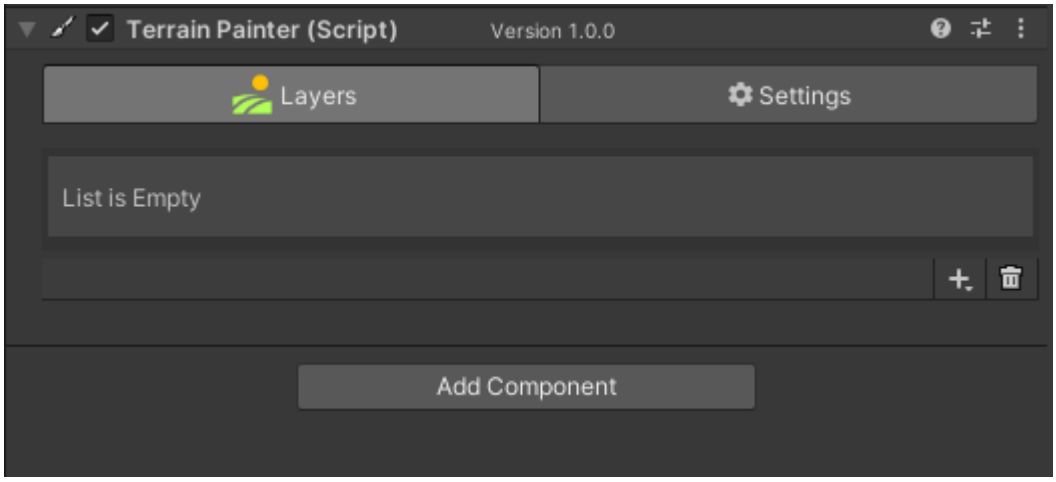


After adding the component, assign your terrain objects to the "Terrains" list.

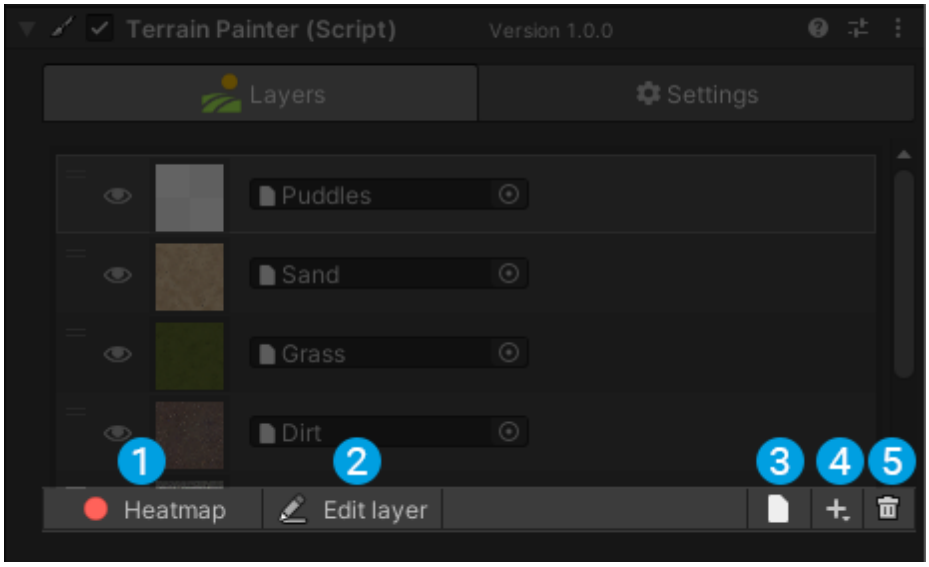
This component is now going to control all terrain painting, so any existing painted materials are going to be cleared. The terrain system can't differentiate between manually painted textures, and those procedurally applied. So, it's always one or the other.

Adding layers

Switch over to the "Layers" tab to start adding terrain layers. A terrain layer is an asset that holds a diffuse, normal and mask map texture. For all intents and purposes, you can consider is a regular material, but one without a shader.



Layer toolbar



1 Toggles the heatmap visualization. This'll cover all the pixels on the terrain red where the layer is applied.

2 Expands the layer's inspector, here you can modify the layer's textures and tiling properties as if it were selected in the inspector.

3 Create a new layer asset from the texture asset you select in the popup window.

4 Adds a new layer based on one that already exists in your project.

5 Removes the selected layer (Note: cannot be undone).

Add a terrain layer now, using either option 3 or 4.

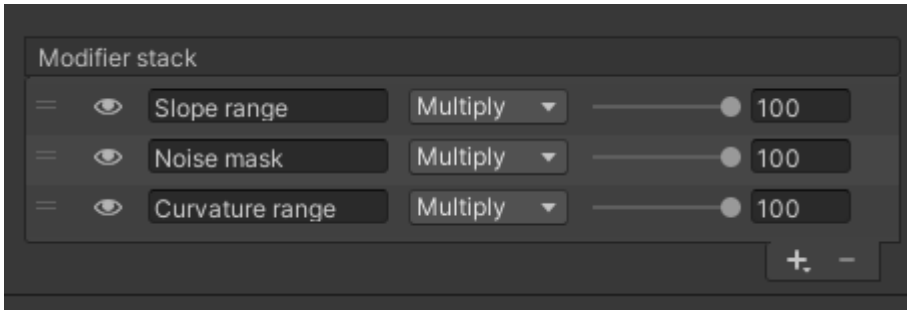
Layers can be reordered by dragging one to a different position. They're processed from bottom to top, in the same way layers work in Photoshop or Substance Painter.

The base layer at the bottom of the list always fills the entire terrain, so it has no options when you select it.

Modifiers

After adding a base layer, and at least one other layer. You will notice this once again fills the entire terrain, through the modifier stack you can start to add painting rules.

Press the “+” button to add a new modifier.



Modifiers essentially output a mask (values between back and white, 0 and 1), this determines how strongly a layer is painted on the terrain, and where. Like layers, they are processed from bottom to top.

The dropdown menu determines the blending mode, between itself and the modifier below it. This is analogous to how blending modes work with layers in other software like Photoshop and GIMP.

The slider represents the modifier’s opacity.

Modifier	**Description**
Curvature	Masks between concave and convex areas on the terrain. Concave meaning crevices and dents, whereas convex means ridges and spikes. Limiting a material to concave areas can create the natural appearance of sediment (small, loose rocks pushed down the terrain by water)
Height Range	Creates a mask based on a minimum and maximum height. The height is relative to the lowest point on the terrain. The falloff values create a smooth gradient, starting from the min/max values you set.
Noise	Generates simplex or gradient noise. This can be used to break up a material
Slope	Slope represents the 0-90 degrees angle of a point on the terrain. For example, you can have grass only paint on 0-45 degrees areas. Note that the result of this modifier depends greatly on the heightmap resolution. If this is changed at some point, you’ll likely have to fine tune this modifier again
Texture Mask	Aligns a texture with the terrain (or all of them if the “Span Terrains” option is enabled). You can specify which color channel is being used. This is useful to utilize an externally created mask texture (or even a splatmap) to create area specific painting.

Settings tab

Splatmap resolution

A splatmap is an RGBA texture, where each channel describes how strongly a certain terrain layer is applied to the terrain. For every 4 terrain layers, the terrain creates 1 splatmap.

A higher splatmap resolution means the transitions between different terrain layers is more detailed, but at the cost of more memory.

You can at any point change the resolution without loosing any painting settings. So you'll have to choose the resolution that strikes a balance between appearance and memory cost.

Note: The resolution is capped at 1024px. A 4k splatmap results in storing a ~800mb uncompressed texture in the terrain. If you need higher resolution (specifically texels per world-unit), make your terrain tiles smaller.

Colormap resolution

The terrain system creates a precomputed texture that represents the terrain color, with all terrain layers combined and blends this in after the "Base Map Dist." value you set in the terrain settings.

You can generally get away with a low resolution for this.

Recalculate Bounds

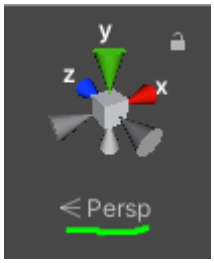
When the settings tab is active, you will notice a white box around the terrain. This box should encompass all terrains exactly, if you have resized your terrain, or moved them somehow, the bounds must be recalculated. This information is used by both the height, noise and texture mask modifiers.

Creating biomes

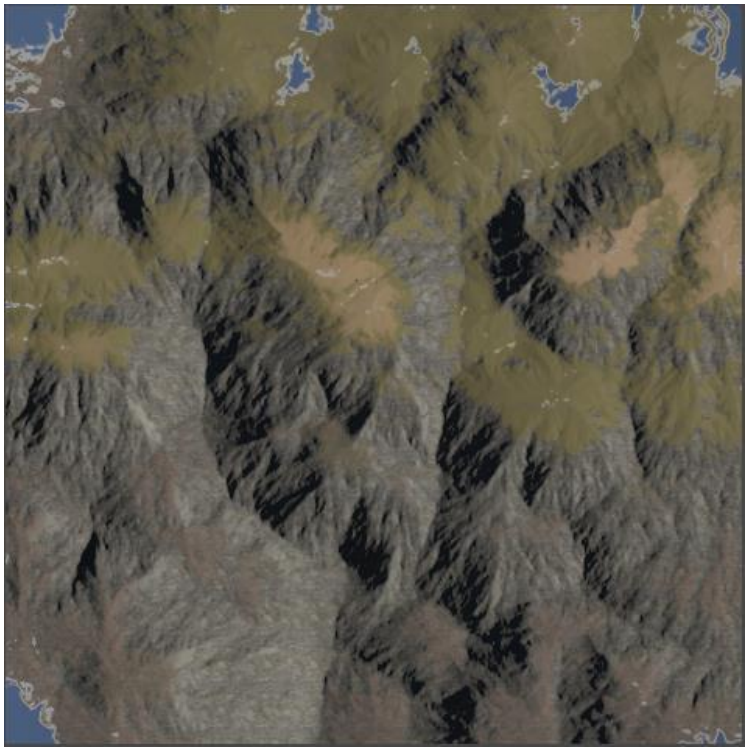
Biomes are distinct regions that represent a different climate. In classic game world terms this means: tropical, volcanic, icy, temperate, etc.

The Texture Mask modifier can be used to apply texture to specific regions on the terrain.

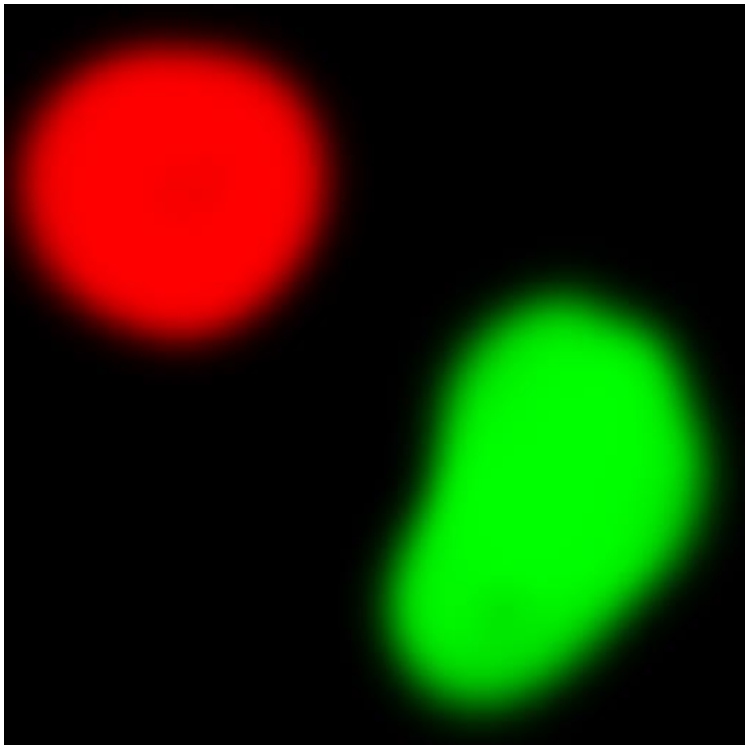
To create one, click the "Persp" text on the scene view gizmo, then click the Green arrow. This will align the scene-view camera to a top-down orthographic perspective.



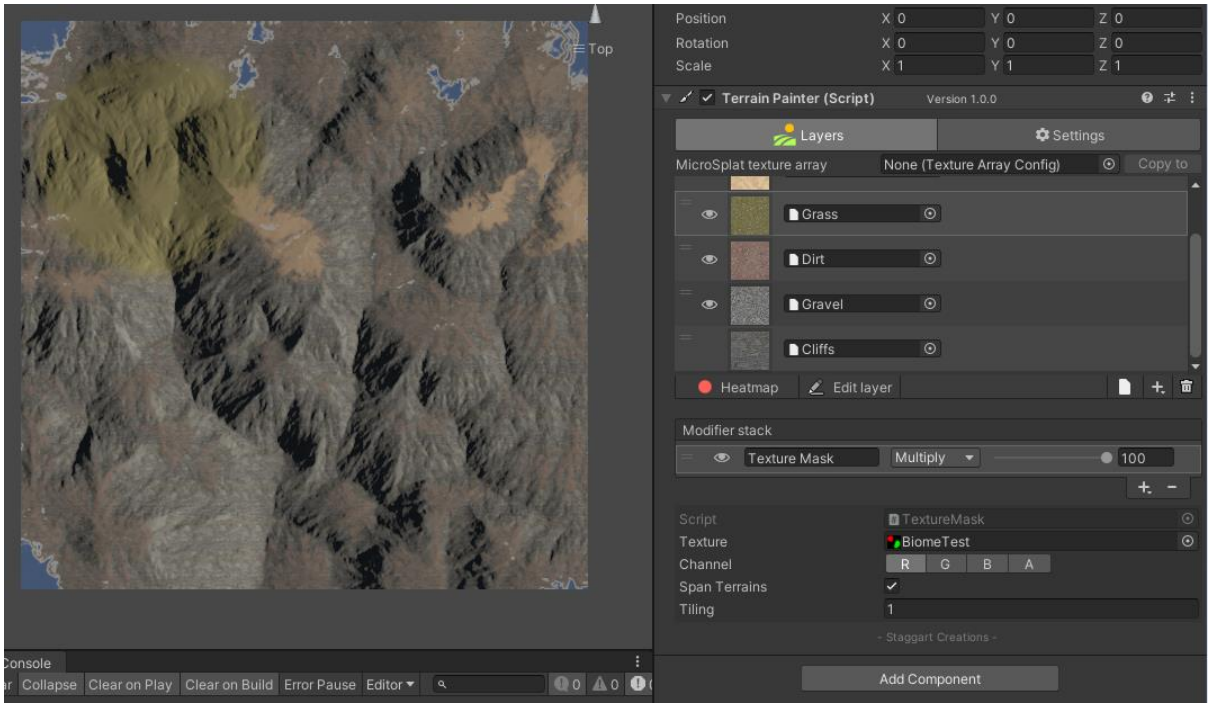
Then bring your entire terrain into view and capture a screenshot. This will serve as a reference image for painting later on.



In your favorite image editing software, create a black background, and paint a pure red color on the region you want a specific terrain layer to show.



Next, add a “Texture Mask” modifier to the stack. And check the “Span terrains” checkbox.

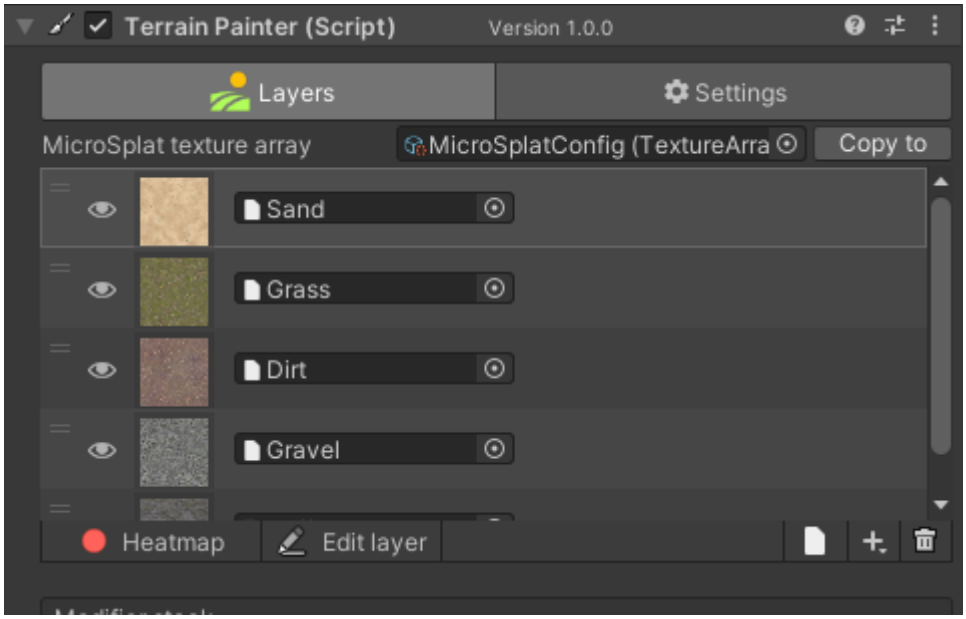


You'll notice the layer is only applied where the red area is painted in the texture.

You can reuse the same texture for other layers, and simply use a different color channel (such as the green area). This also means you can use splatmaps created in external applications such as World Machine, or masks exported from World Creator.

MicroSplat integration

If MicroSplat is installed in the project, a field will appear to assign a related *TextureArrayConfig* asset:



If it is assigned, when you add, remove or re-order a layer the same operation is performed on the texture array.

The "Copy to" button will copy the current list of layers into the array. This can be used to force the texture array to match the layer configuration in the Terrain Painter. Working with MicroSplat will only work if the Terrain Painter component takes full control, copying from the texture array is not possible, since it has no actual terrain layers.

Note: switching the terrain layer asset will swap out the diffuse and normal map textures in the array accordingly. But whatever Height/Smoothness/AO map that was there, will be left unaltered.

Re-ordering a terrain layer will not re-order the MicroSplat configuration file. Meaning if Layer A has any override options (such as tiling, or interpolation contrast) and it is swapped with Layer B, Layer B will end up using Layer A's settings.

It will be worthwhile to check the array after doing heavy modifications, to ensure that no textures are assigned where they should not be. Ideally, you'd set up MicroSplat after having largely finished all terrain painting.

Known issues

Console error about PaintContext.ApplyDelayedActions()

Safe to consider this a bug (API being used is experimental after all), the function should do a null check on the terrain it is trying to process. It is essentially calling on a terrain object, that no longer exists. This can happen if:

- You close the scene, without saving, then save the project
- Reverting a scene with terrains in it through source control, then saving