

Flowmap Generator Sand Dunes Tutorial



Table of Contents

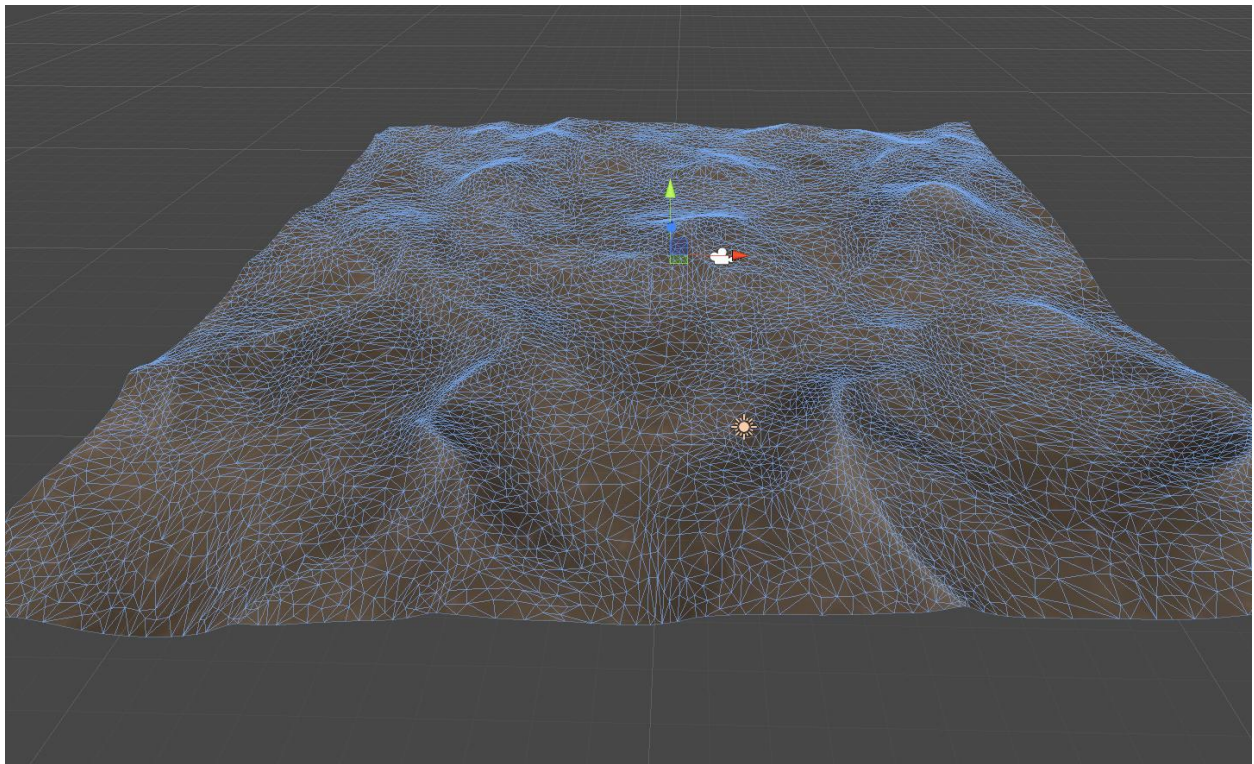
Getting started	3
Render Heightmap Setup	5
Add Fluid	5
Add Force Fields	7
Bake to texture	7
Material Setup	8

Getting started

A good example of the uses for flowmaps beyond water is sand blowing over dunes. The principle is the same, using a flowmap to give the illusion of motion, but in this case we use a diffuse texture instead of normals.

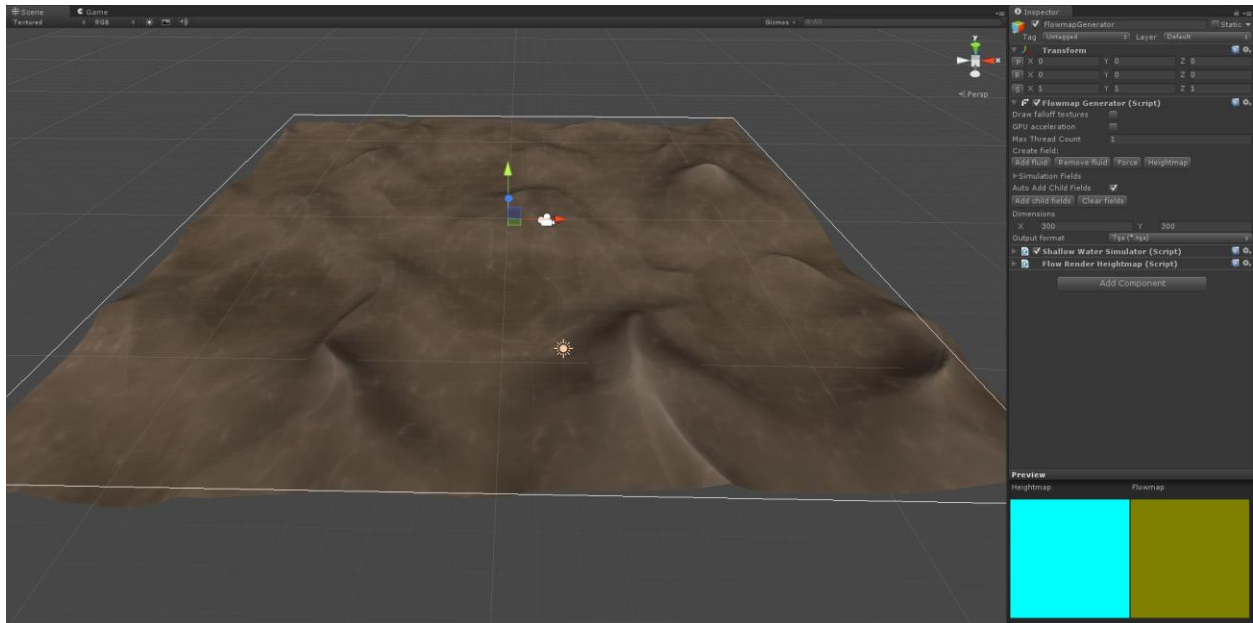
For this scene I'm using a feature that requires Unity Pro. That feature is rendering a heightmap from scene geometry. If you don't have Unity Pro the best approach is to generate a heightmap from your geometry using another application. You can then import that heightmap into Unity and use it as the heightmap for a Flow Texture Heightmap component. The result will be the same, it just requires more work outside of Unity.

The shader I'm using for this scene is a bit complex. In most cases with surface flow you will probably need to write your own shader. Most use cases are quite unique, so making a shader that works well for all use cases isn't really possible. The Blowing Sand shader I've used for this example uses world projected Uvs that assume the terrain is centered in the world and is 300x300m.



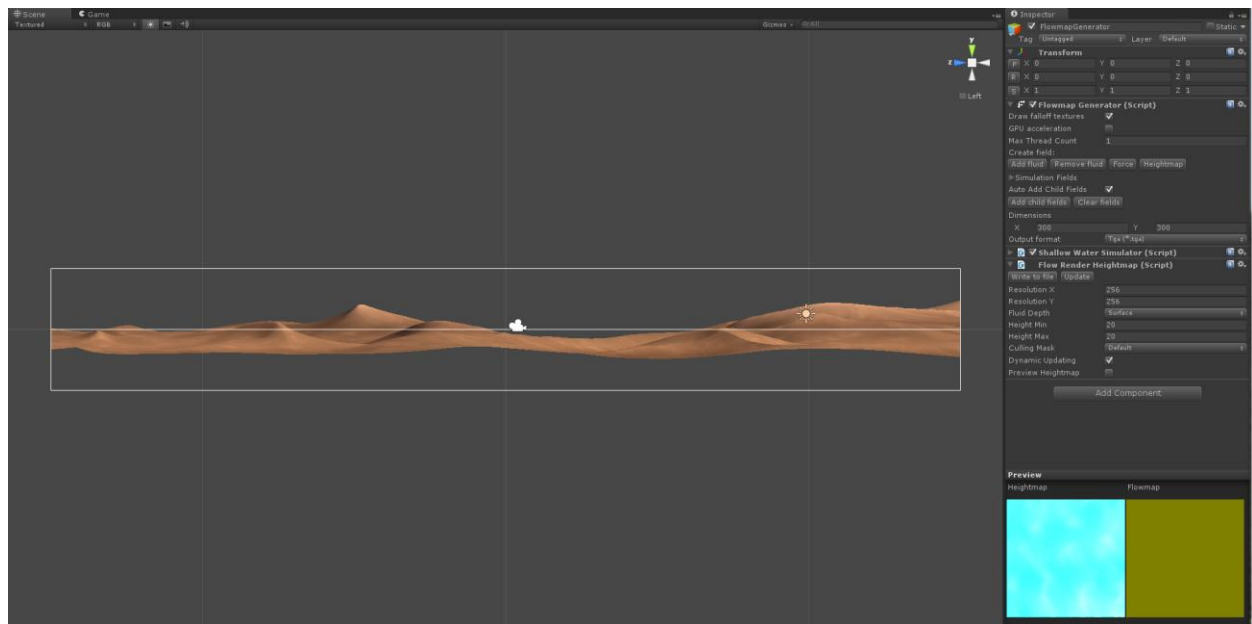
Create a Generator

To get started I first created a Generator from the Game Object->Create Other->Flowmap Generator menu. I selected the “Render” heightmap style. The next step is to position the generator and set it's Dimensions to match my terrain.



Render Heightmap Setup

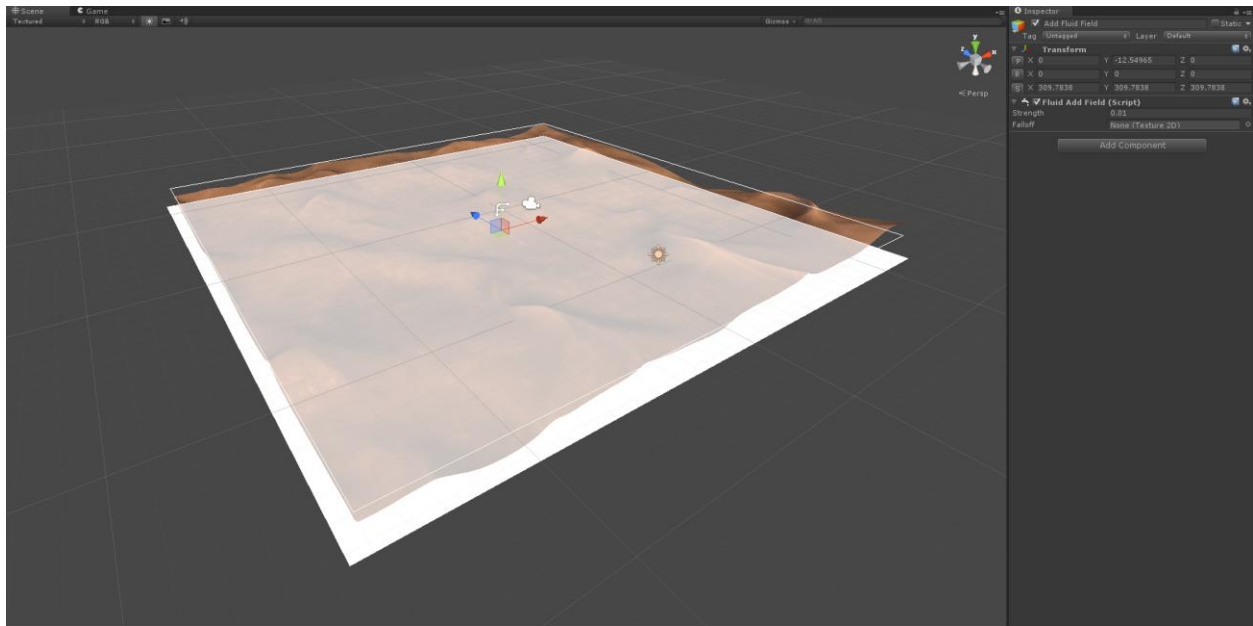
As I'm using a Flow Render Heightmap to render my scene geometry to a heightmap, I need to set the Height Min and Height Max to contain my terrain mesh. Height Min extends below the Generator's position and Height Max extends above. The heightmap will have a value of black at the Height Min and a value of white at the Height Max. I've also set the Fluid Depth to Surface, this will render out a standard heightmap. The DeepWater setting renders out a heightmap that acts more like an intersection mask.



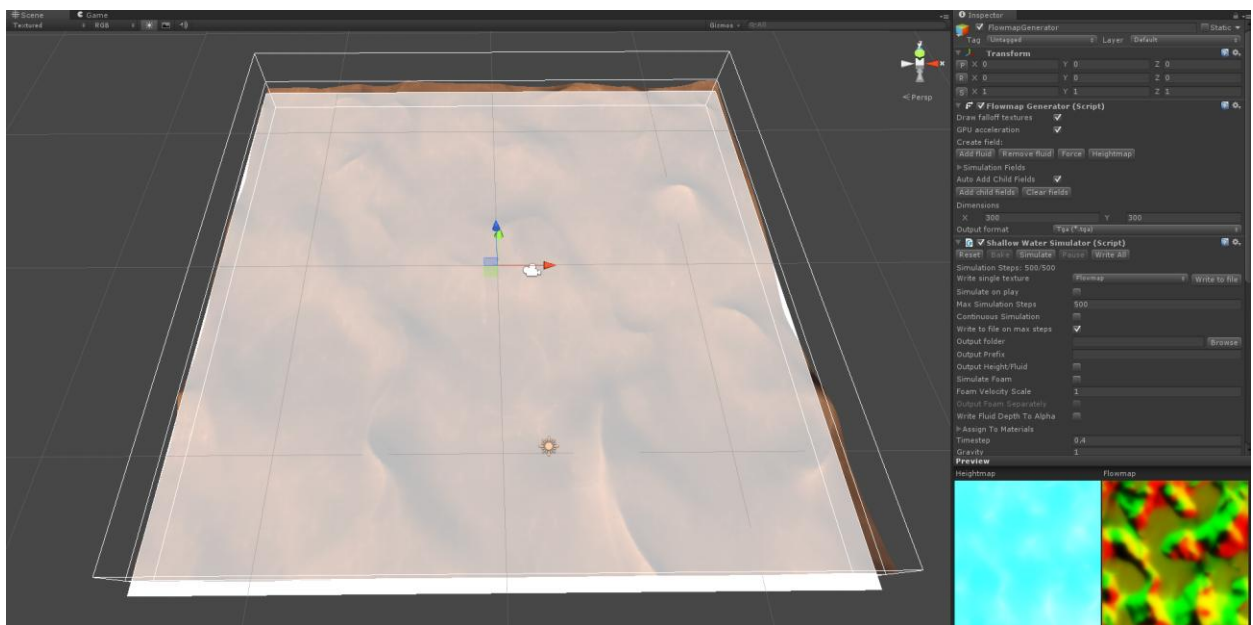
Add Fluid

I want the flowmap to have an effect over the whole terrain, so I will need to add fluid that can be pushed around by forces and react to the heightmap. The simplest way of doing this is creating an Add Fluid field, found in the Generator component. Then I scaled the field so it covers the whole terrain. I also removed the falloff texture, as I want the same effect over the whole terrain. I've also lowered the strength to 0.01. Too much fluid in the simulation will result in a flowmap without much detail.

The position of the field in the Y axis has no affect on the field, so I've placed it a bit below the terrain so the scene is a bit easier to work with.

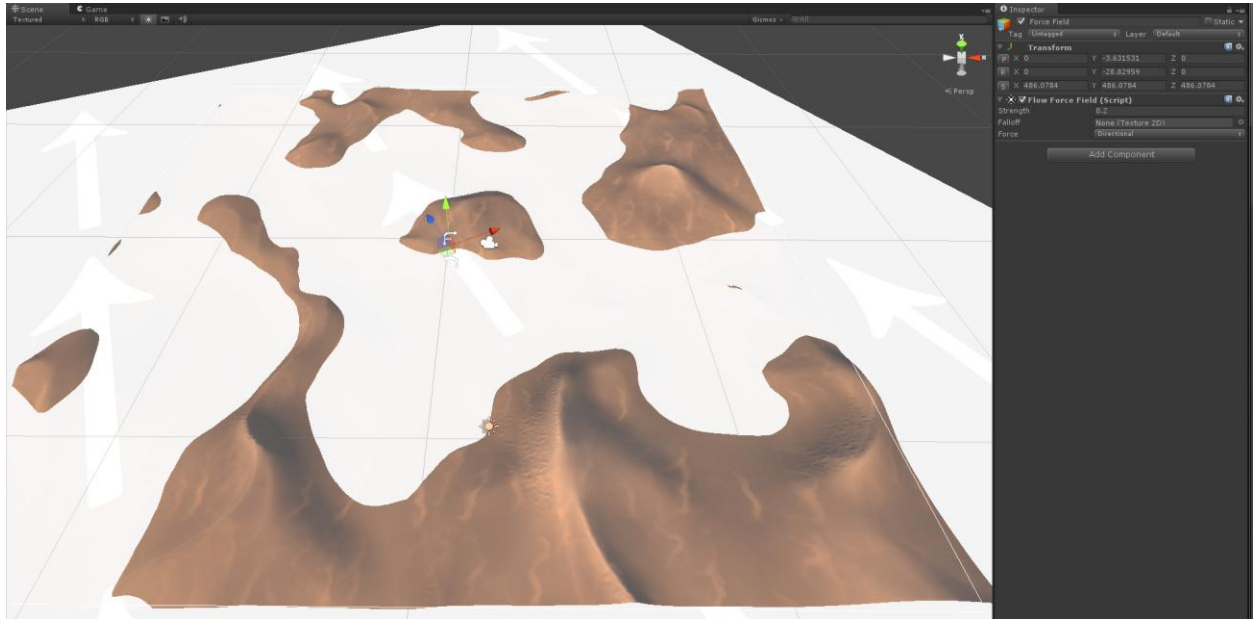


On the Shallow Water Simulator component I set Border Collision to PassThrough to let the fluid run out of the borders of the generator, instead of bouncing back. I also set Fluid Depth to Surface. Running the simulation now should result in a flowmap with the flow vectors pointing downhill.



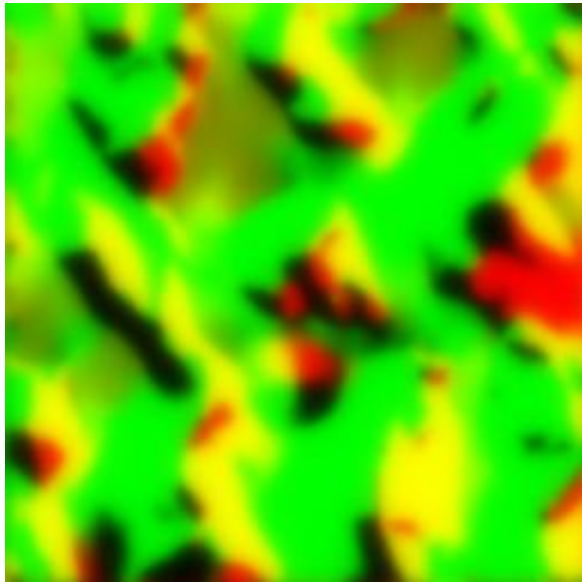
Add Force Fields

The next step is placing Force fields to control the wind direction. I've placed a large Force field with the Force type set to Directional. I set the Strength to 0.2, so that it doesn't completely overpower the heightmap's effect on the fluid.



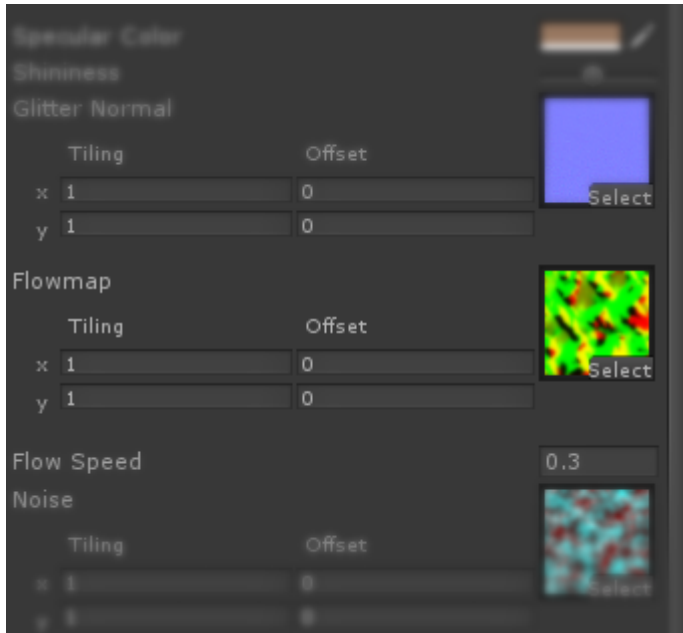
Bake to texture

After setting my output folder to a folder in my Unity project I baked the simulation for 500 simulation steps. Here's the resulting flowmap.

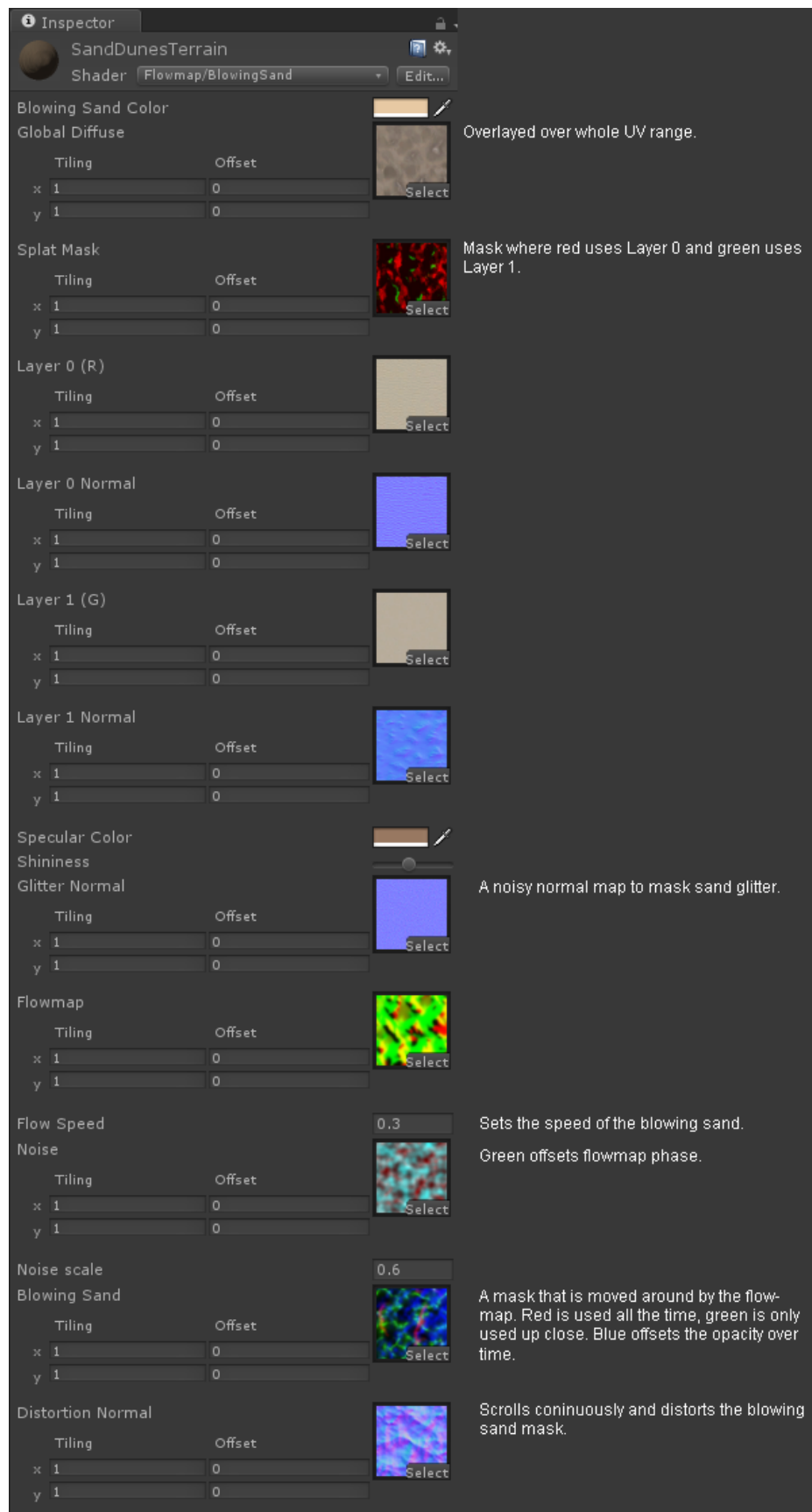


Material Setup

The next step is to set this texture as the Flowmap texture for the material I have applied to my terrain mesh.



The shader I'm using has quite a few texture inputs that need to be set. Below is a bit of information about some of the inputs.



The shader has a few comments that should help for creating your own shaders. Some inspiration for this shader came from Journey. An interesting write up about their sand technology can be found here:

http://advances.realtimerendering.com/s2012/thatgamecompany/SandRenderingInJourney_thatgamecompany.pptx