

Flowmap Generator Quickstart

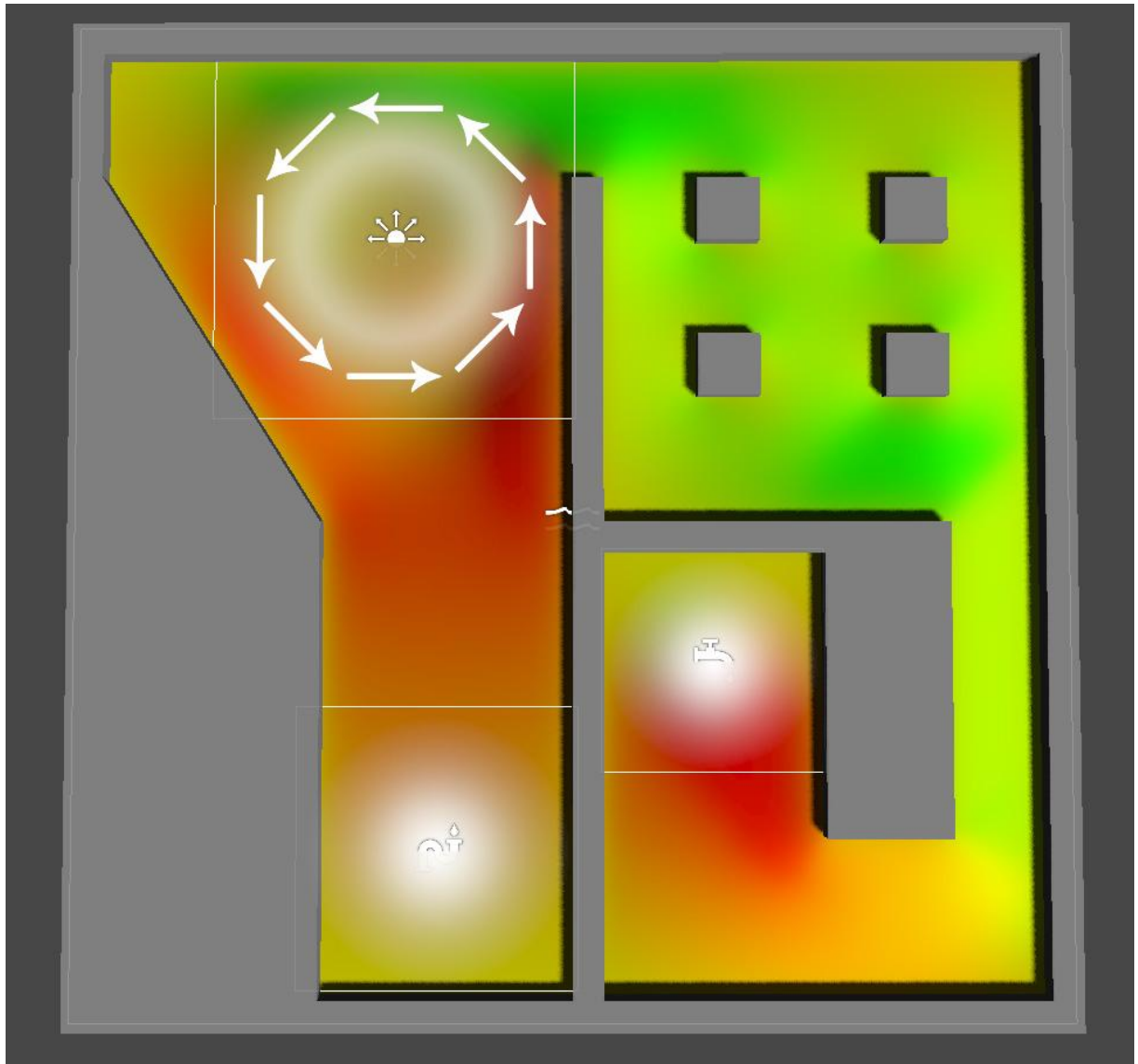


Table of Contents

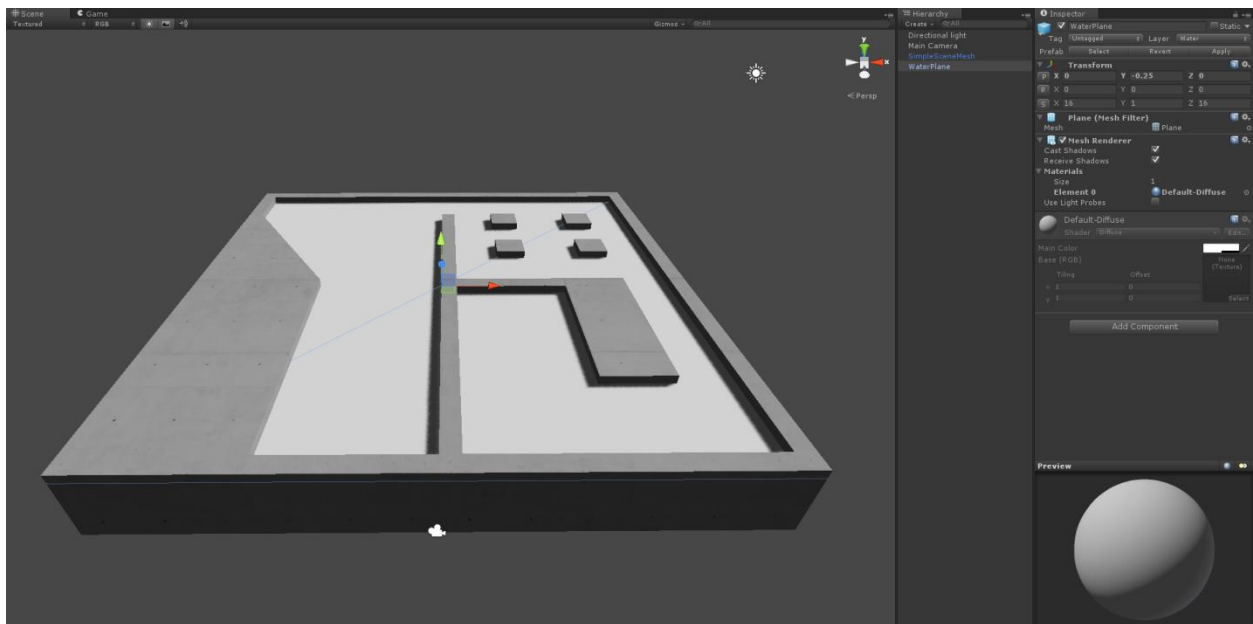
Quick start guide	3
First steps	3
Material Setup	3
Creating a Generator	6
Heightmap	8
Render	8
Texture	10
Fields	11
Simulating	13
Using the baked flowmap texture	15

Quick start guide

The purpose of this guide is to get you up and running with Flowmap Generator. I'll be using the Quickstart scene, found in FlowmapGenerator/Examples/Quickstart/Scenes. Also in that folder are finished scenes if you want to see how everything should look at the end of this guide. When you've gone through this quickstart guide check out the full reference for information about the other settings that aren't covered in this guide.

First steps

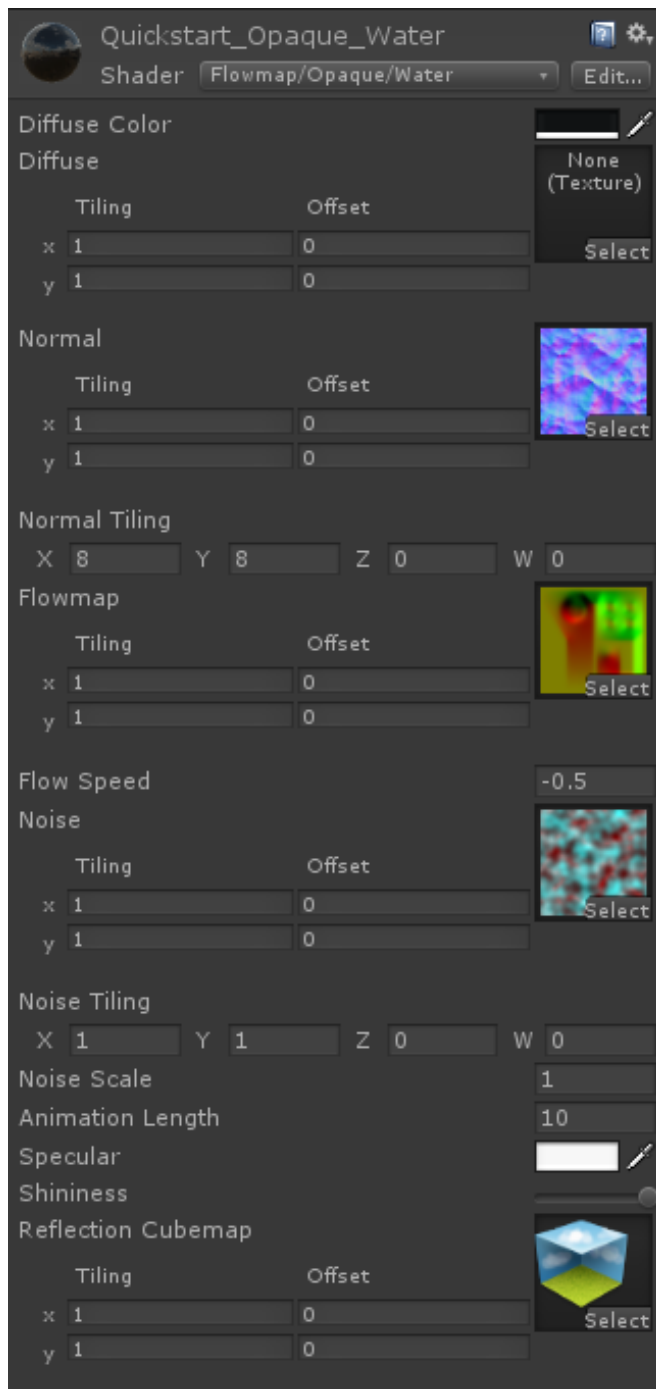
Open the scene called Quickstart, located in FlowmapGenerator/Examples/Quickstart/Scenes. You should see a simple scene with a water plane. We want to generate a flowmap to apply to the SimpleFlow material on the WaterPlane GameObject.



Material Setup

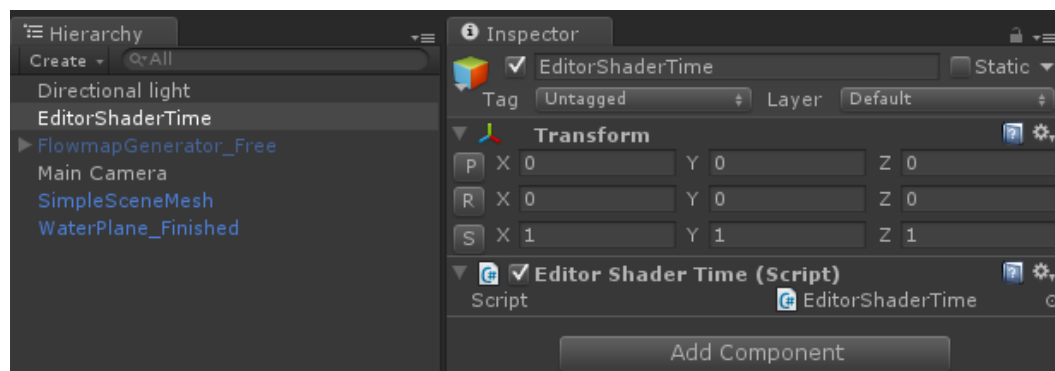
The shader I'm using is FlowmapGenerator/Opaque/Water. This is a basic flowmap shader where the normal map is affected by the flowmap. It has inputs for diffuse color and texture, along with

specular color and a reflection cubemap. The rest of the parameters and textures are for controlling the flowmap.



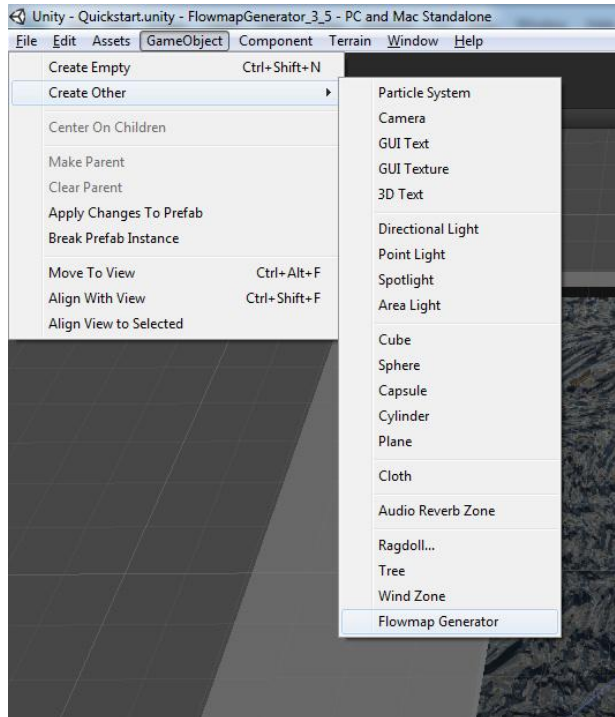
Parameter	Info
Diffuse Color	The surface diffuse color.
Diffuse	A texture multiplied with the diffuse color.
Normal	A normalmap that is affected by the flowmap.
Normal Tiling	Controls the UV tiling for the Normal texture.
Flowmap	The flowmap created by Flowmap Generator. Should be set to bypass sRGB sampling.
Flow Speed	Controls the speed of the flowmap. Negative values reverse the flow direction.
Noise	A noise texture used to offset the flowmap phase. The green channel is used. Should be set to bypass sRGB sampling.
Noise Tiling	Controls the UV tiling for the Noise texture.
Noise Scale	Controls the phase offset strength. Large values create extra distortion.
Animation Length	The length of time in seconds between phase changes. Use smaller values when using a faster Flow Speed.
Specular	Specular color.
Shininess	Specular highlight size.
Reflection Cubemap	A cubemap for reflections. Multiplied with specular color.

Flowmap shaders use the `_Time` shader variable to animate. In the Unity editor when not in play mode this variable doesn't change. To be able to see flowmap animation I've made a small script called `EditorShaderTime` that sets a global shader variable, `_EditorTime`. This variable is added to `_Time` in the shader so that the flowmap shader will always animate, even in the editor when not playing the game. If you are using a shader included with Flowmap Generator, just add the `EditorShaderTime` script to a `GameObject` in your scene and your flowmap shaders will animate properly.



Creating a Generator

Create a new Flowmap Generator object, found under GameObject->Create other->Flowmap Generator.

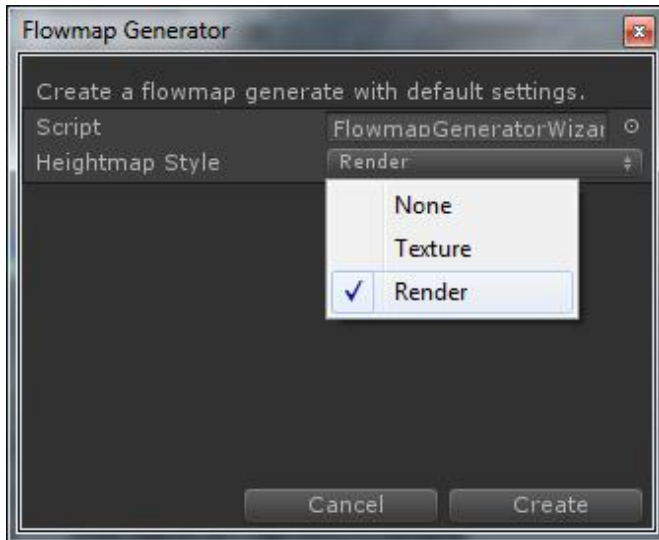


The new Flowmap Generator wizard will appear where you can change some settings. All of these settings can be changed later. Choose a heightmap style that is supported by your version of Unity. For Unity Pro select Render and for Unity Free select Texture.

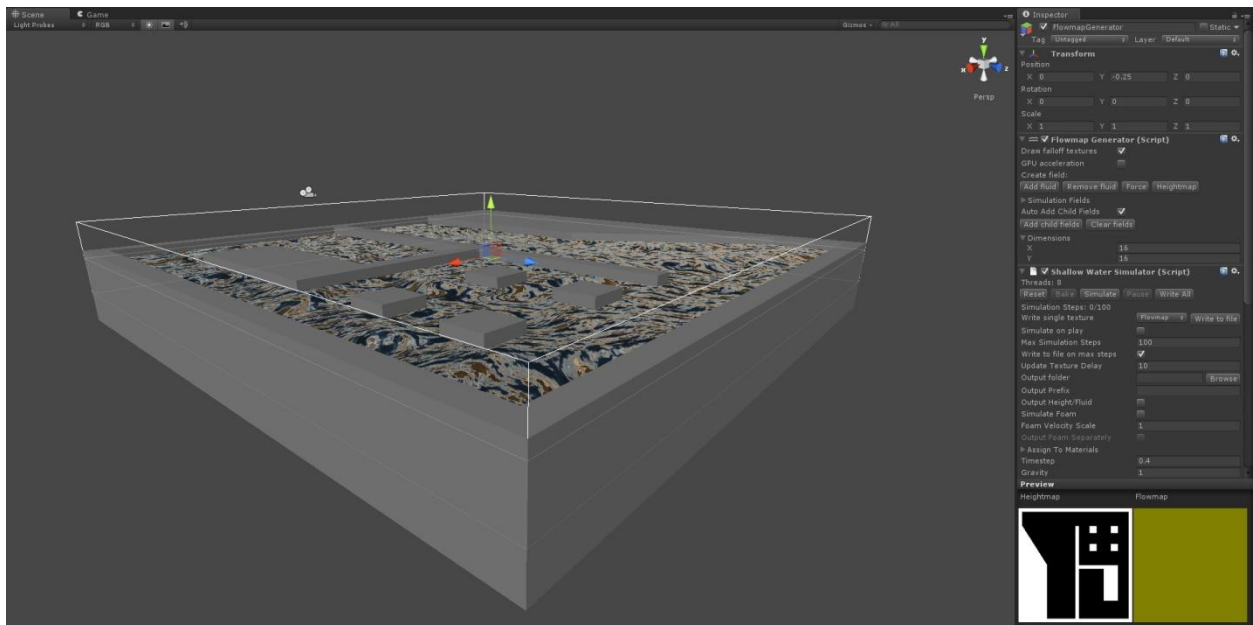
None - No heightmap will be used for this flowmap. This is useful for open water or if you want to use heightmap fields to create a heightmap for the generator.

Texture - Select this if you have created a heightmap using an external program.

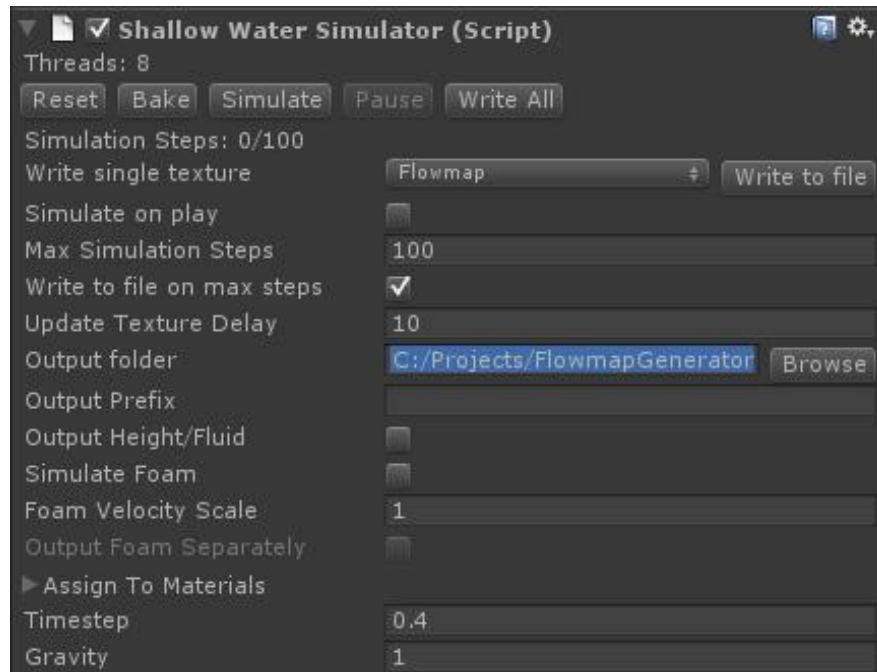
Render - (Unity Pro only) Render a heightmap from your scene geometry.



Now you can click create to create a new Flowmap Generator. Set the position to match the water plane's position and the dimensions to match the size of the water plane. This is especially important when rendering a heightmap from your scene geometry.



Set an output folder to save the flowmap texture. For this example I've chosen to save the flowmap to FlowmapGenerator/Examples/QuickStart/Textures.

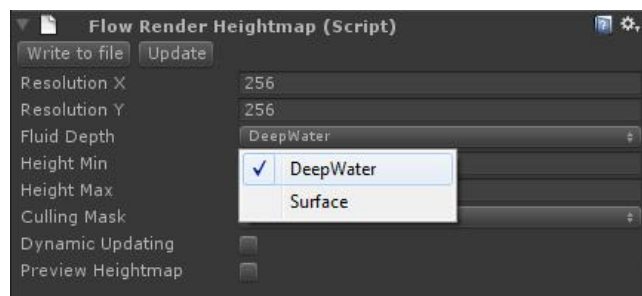


Heightmap

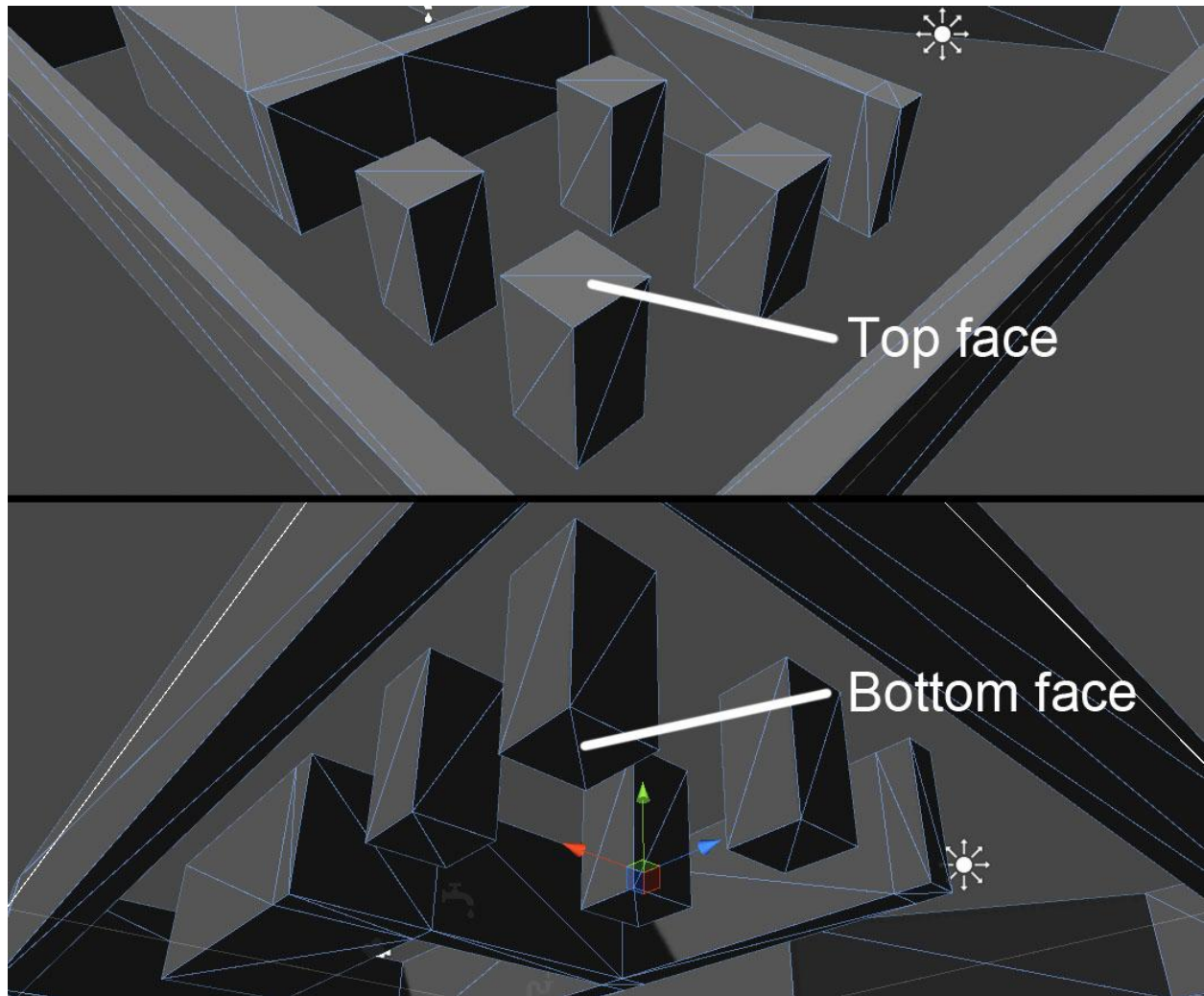
Render

If you are using Unity Pro and have chosen to use a Render heightmap it's time to set that up. The culling mask should be set to a layer that will include objects that intersect with the water plane. It is possible to use different geometry that is only used for creating flowmaps. For this example I have my scene geometry on the Default layer and set the Culling Mask on Render heightmap to the same layer (Default). I've assigned the Water layer to my water plane so it won't be rendered into the heightmap.

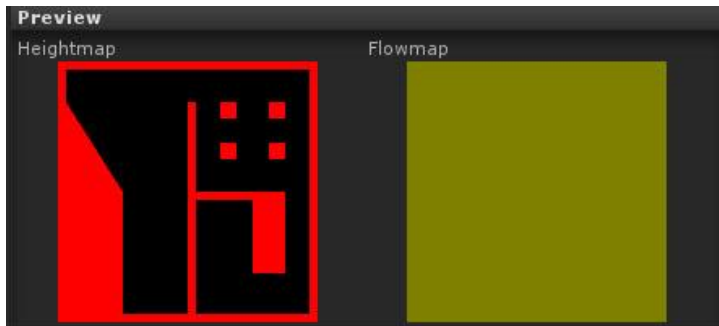
In this example we want a heightmap that is white where the geometry intersects with the water and black where it does not. To render a heightmap in this style, set the Fluid Depth to DeepWater.



There are certain requirements for the geometry to be rendered into a heightmap using the DeepWater setting. The Surface setting can be used when you want to generate a flowmap that simulates a fluid on the surface of an object. The geometry to be rendered should be watertight, when looking at the mesh from a top down view there should be faces pointing towards the camera above the water plane and faces pointing away from the camera below the water plane. For this reason it is often better to make a custom mesh that can be used for baking a heightmap.



When the Flowmap Generator is selected you will see a bounding box corresponding to the Height Min and Height Max values. Adjust the two values so that the top of the bounding box is above all geometry that intersects with the water plane. Height Min should be set so that the heightmap will only include the objects that intersect your water plane, this can often be a small value, less than 1 unit. If you click the Update button on the Flow Render Heightmap component you should see a preview of the heightmap. If not, double check the culling mask layer and the layer for your scene geometry. Try adjusting the Height Max and Depth Max. In Unity 4 the heightmap may be tinted red. This is because it will try using a render texture format that only contains a red channel, to save video memory.

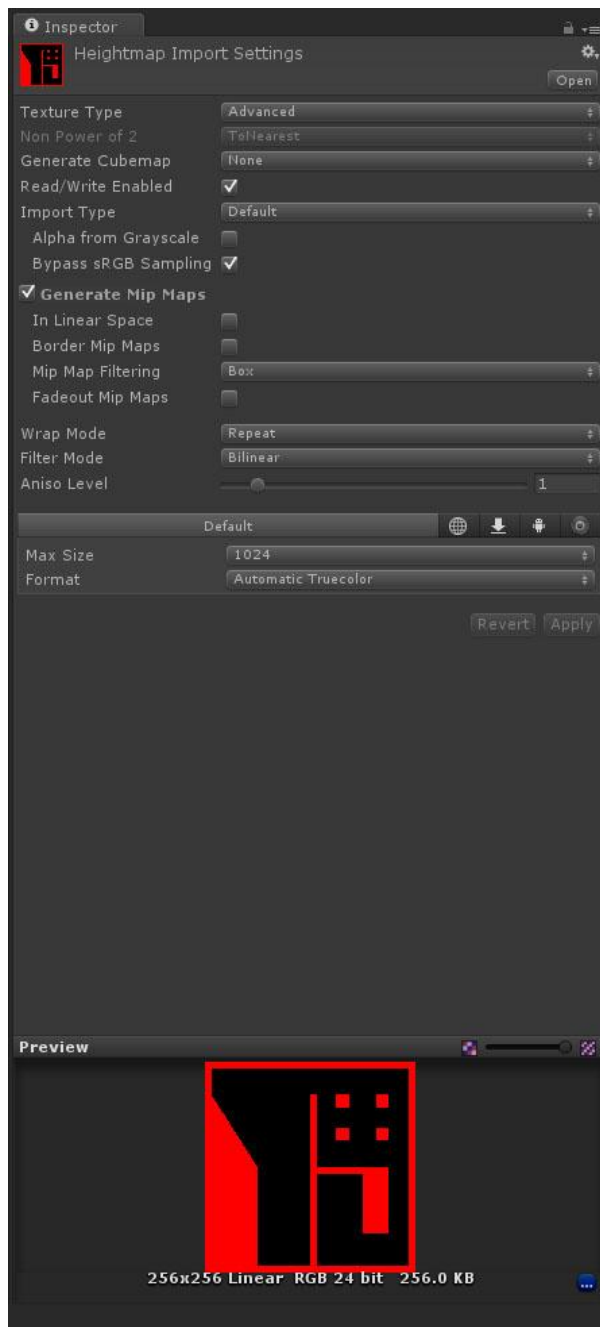


Texture

If you are using Unity Free and chose to use a texture heightmap when creating the Flowmap Generator you can now assign a heightmap texture to the Flow Texture Heightmap component.



Certain settings need to be set on the texture to be used as a heightmap. Set texture type to Advanced. Read/Write should be enabled and the texture should be set to Bypass sRGB sampling. It's also a good idea to set the format to Truecolor.

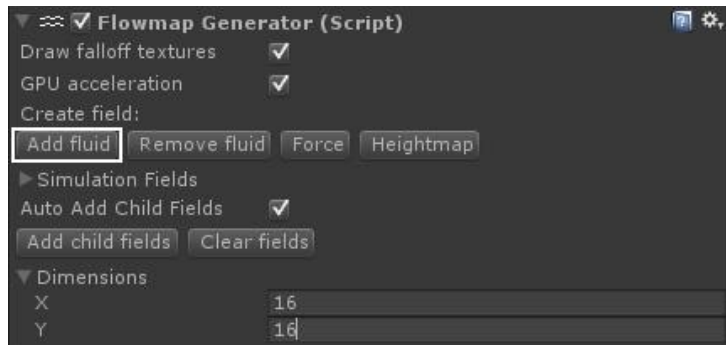


For this example you can use the Heightmap texture in FlowmapGenerator/Examples/QuickStart/Textures. This texture was rendered out using the Render heightmap for Unity Pro. For your own scenes you can render out a heightmap from your 3D application or create one by hand in an image editing application.

Fields

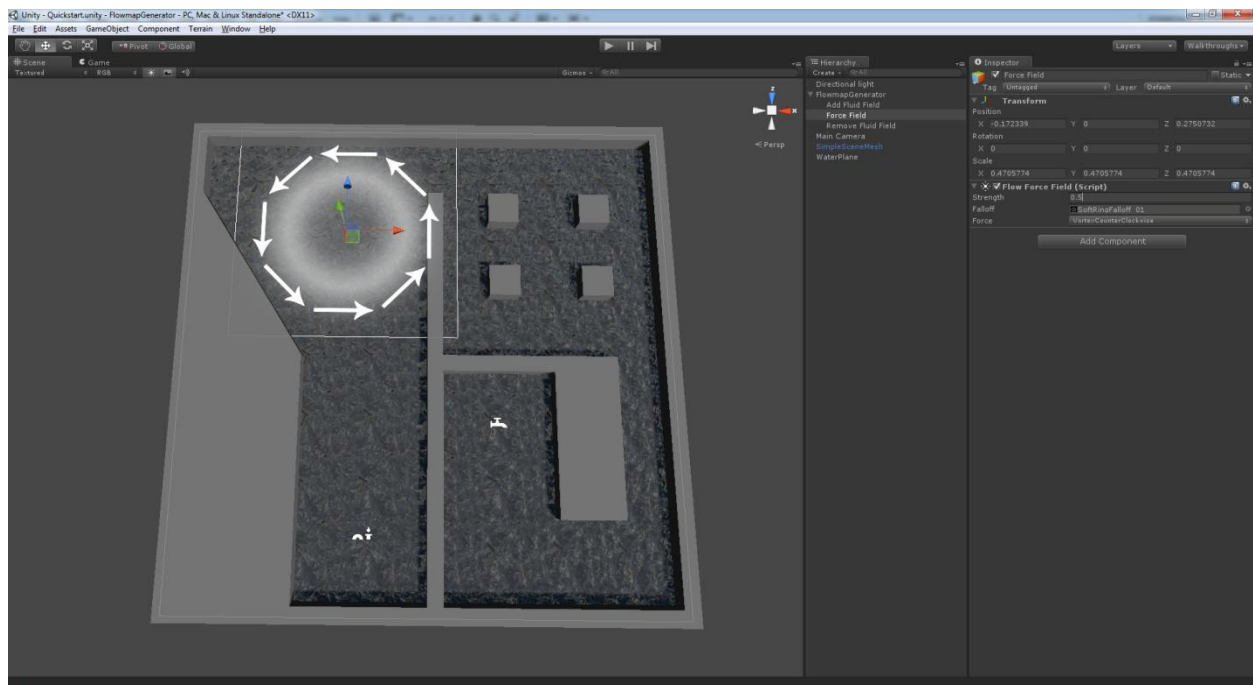
The next step is to add fields to your generator. In order to have any kind of fluid simulation at all we need to first add some fluid to the simulation. For this example we'll make an Add Fluid

Field to give the appearance of water bubbling up from below. In the Flowmap Generator component click the Add fluid button. This will create a new field and automatically add it to the generator. The new Add Fluid Field will be a child of the generator, but it's not necessary to do this. As long as the field is added to the Simulation Fields list and is enabled it will affect the simulation.



Fields can be moved and scaled and the strength property along with the falloff texture controls how much the field will affect the simulation.

In my scene I've also added a Remove Fluid Field and a Force Field. A Remove Fluid Field removes fluid from the simulation and pulls fluid towards it in the process. This can give the appearance of a drain. A Force Field applies a force to the fluid. I've added a Vortex force field to my scene.



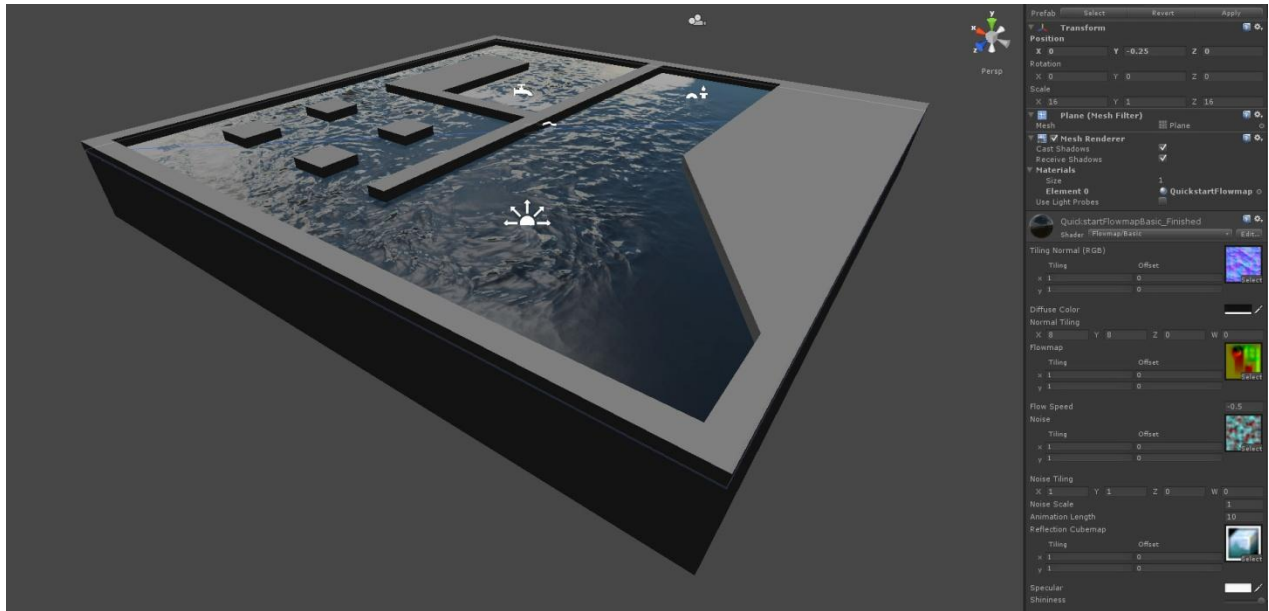
Simulating

Now we're ready to start simulating. If you're using Unity Pro you can enable GPU acceleration, it's much faster than simulating on the CPU. The CPU simulation uses multithreading, so the more processors you have the better.

With the default settings the simulation will run for 500 simulation steps. This most likely isn't enough for this example scene. If you set Max Simulation Steps to 0 and click Simulate the simulation will continue without stopping. This is a good way to find out just how many simulation steps are needed to get a good result. For my setup after around 2000 steps the simulation has distributed fluid across the scene and appears to be stable. I can now set the Max Simulation Steps to 2000 and click Bake to simulate 2000 steps, after which the simulation will pause and the flowmap will be saved to the Output folder.

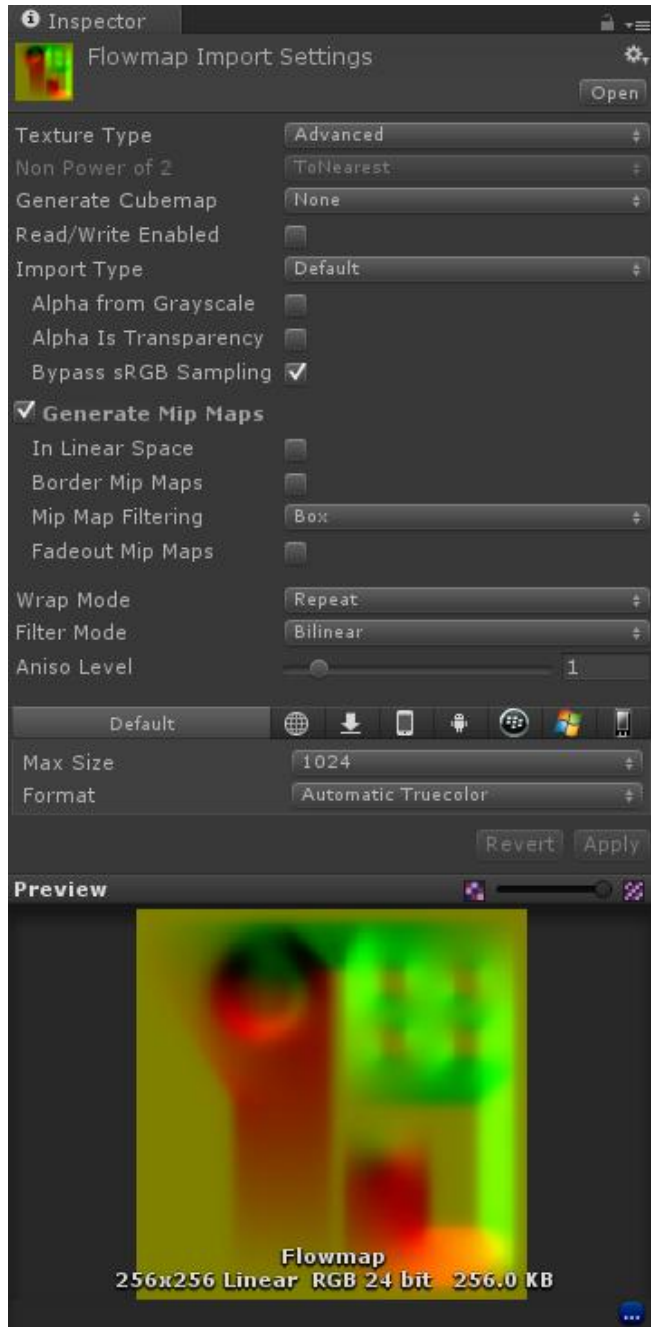


I decided to change the Force Multiplier to 0.02 as well as lower the strength of the vortex field to 0.1. I also decreased the strength of the Remove Fluid Field to 0.1. I increased the Fluid Add Multiplier to 0.05 in order to let the flow of the fluid be stronger than the forces acting upon it. I also lowered the Flow Speed property on my material to -0.5 so the flowmap has less effect. After re-simulating I got the look I wanted. In this example I needed to set the Flow Speed to a negative value. This is dependent on your object's UVs and how it is rotated in the world. Sometimes the X and Y axis will need to be swapped. This can be done either in the shader, or by editing the flowmap texture in an image editing program.



Using the baked flowmap texture

After baking your flowmap texture to a file in your Unity project you are ready to use that texture in your shader. There are a few settings that need to be set for the flowmap to work correctly.



To see the extra settings, set the Texture Type to Advanced. The most important setting is Bypass sRGB Sampling. This should be enabled. This tells Unity to interpret the texture values linearly, instead of applying gamma correction. If you can afford the extra video memory, set the

texture's Format to Truecolor. This reduces compression artifacts. You can now set this texture as the Flowmap texture for your material.

You can now disable (or delete) the Flowmap Generator object from your scene. All the work to animate the flowmap is now done in the shader.