Multi-Texture Shader

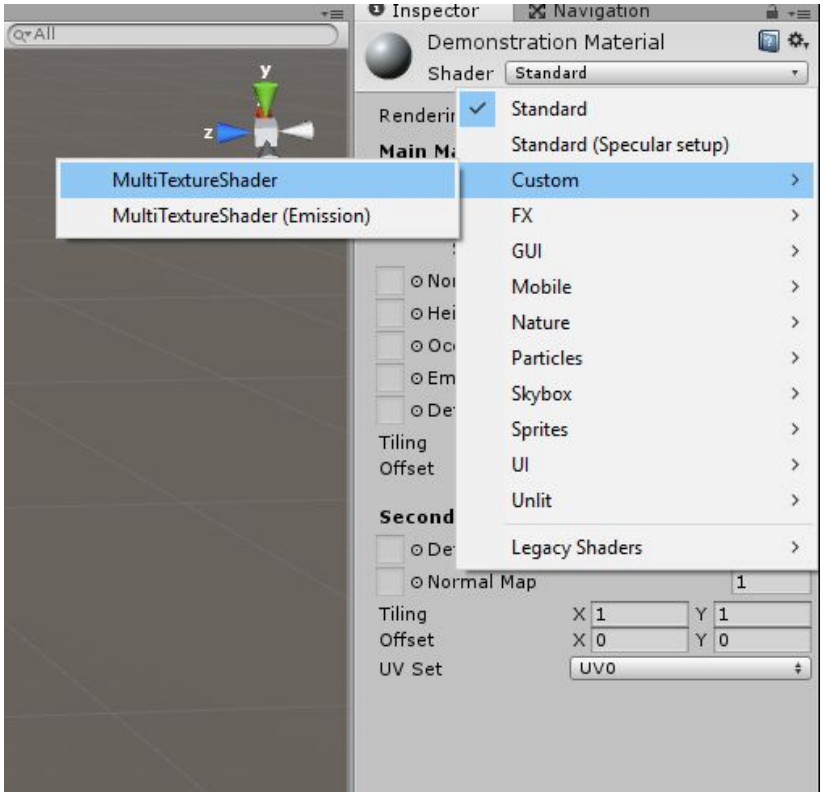
Contents:

Overview:

This shader gives you the ability to apply multiple diffuse, normal, and emissive maps, as well as multiple layers of specular metallic and smoothness sliders, and apply these via bitmasks to a model. This enables the reuse of textures throughout the production of a game or application, and makes the art style very consistent with a lower memory footprint. The entirety of this functionality is done through a single surface shader. This entails a higher computing cost at runtime, but allows you to more easily use multiple materials across entire projects, instead of baking diffuse, normal, emissive, and specular maps beforehand.

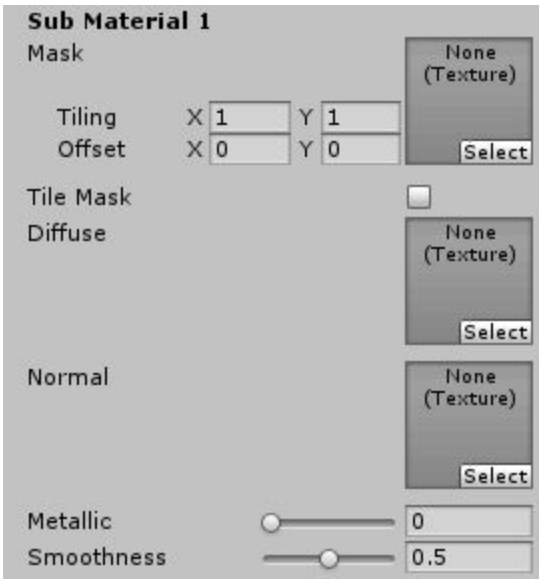
Tutorial:

1. Start by creating a new material. Then, in the inspector, set the shader from “Standard” to “Custom>MultiTextureShader”

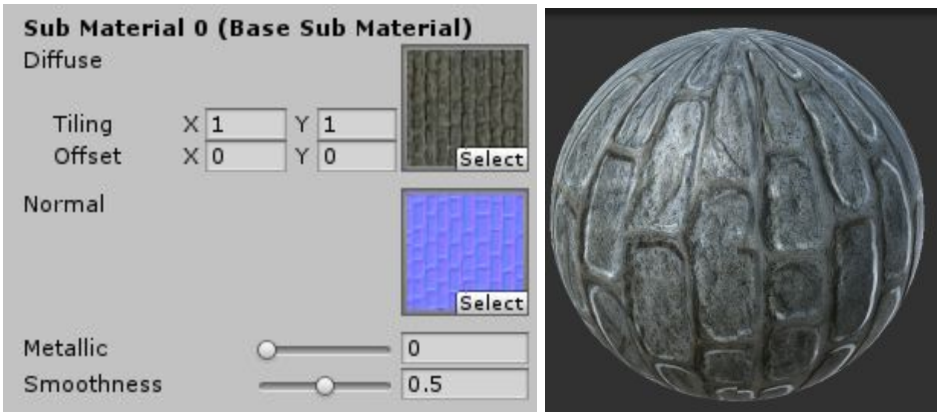


2. You will then see the default shader. Along the left are the inputs for sub-materials. Each of these has 3 respective slots for the following:
    - Mask - determines where (in UV coordinates) the following diffuse and normal maps, as well as the specularity setting, are applied. Note, the first sub-material does not have a mask component. This field also controls the tiling and offset of the entire sub material.
    - Diffuse - determines the color of the sub-material. Identical to “albedo” component of the standard shader.
    - Normal - determines the appearance of the roughness or smoothness of the surface.

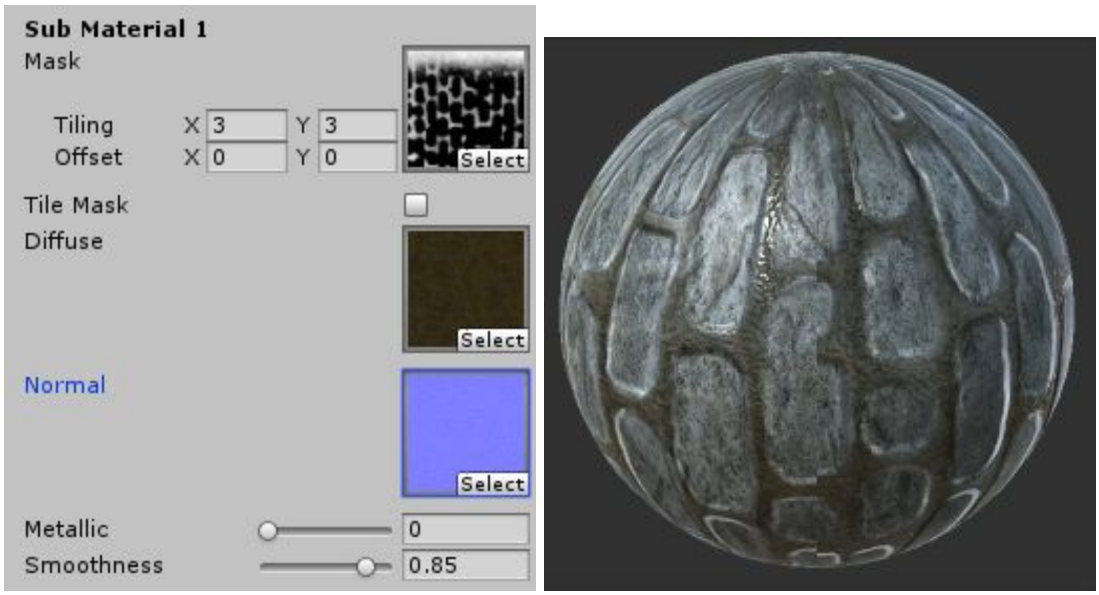
    Additionally, there are two float input boxes for the values for metallic and smoothness.



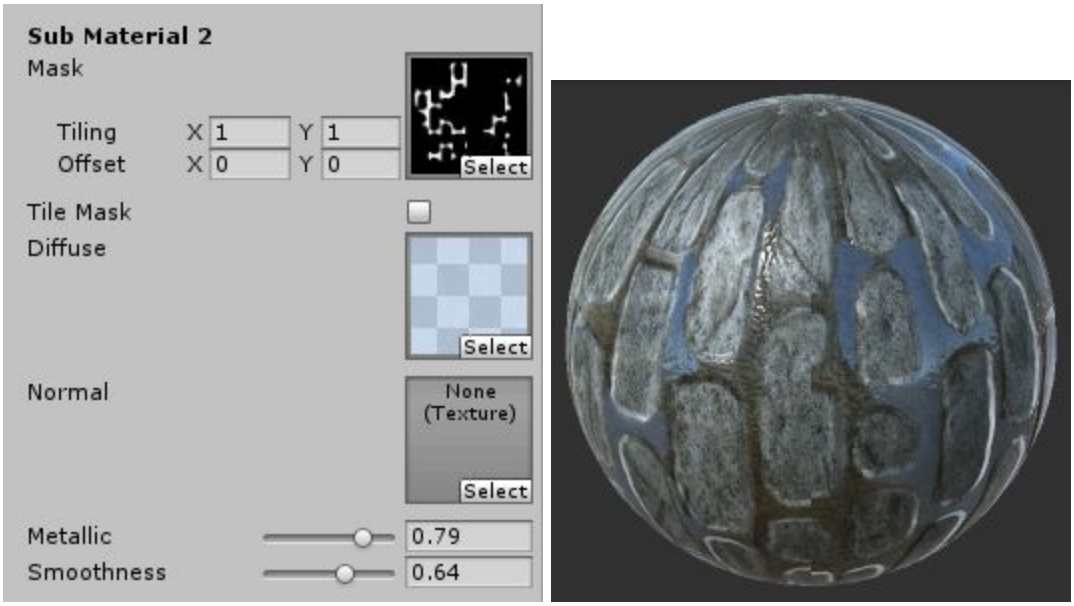
3. First, we will apply our base sub-material in the **Sub Material 0 (Base Sub Material)** section. This will be the underlying texture for our entire model, so be sure to plan around that. Here is the appearance of the material and shader after applying an initial diffuse and normal (found in the cobblestone texture group, in the materials folder). Leave the metallic and smoothness settings at their defaults for now.



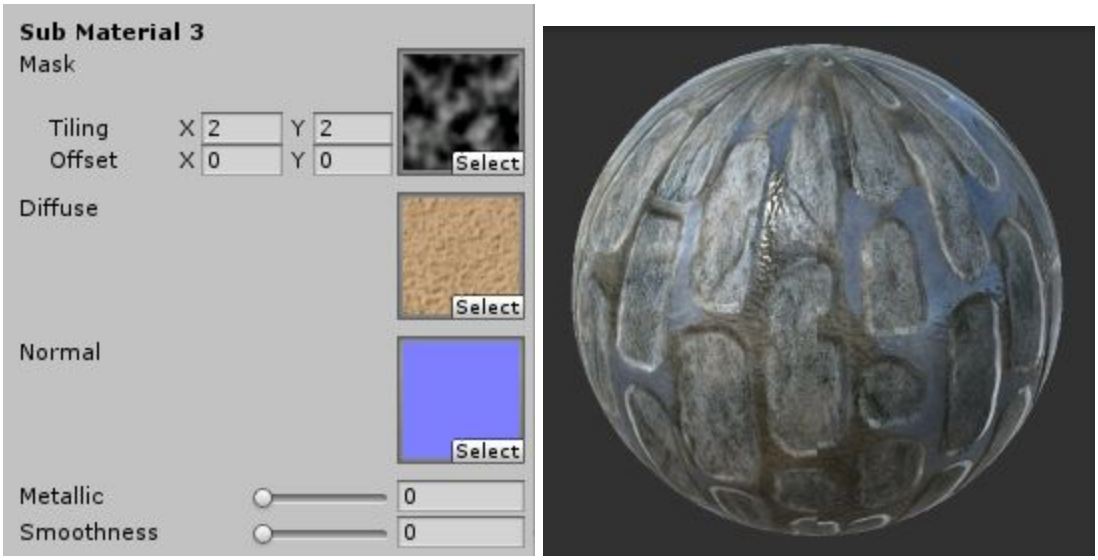
4. Next, we will overlay mud onto the cobblestone. We will do this by applying the mask, before adding the diffuse and the normal maps to the corresponding slots (found in the “Mud” texture group). We will also tweak the mud’s smoothness - we want it to have a dull shine. Set this value to 0.85, emulating the look of wet mud. This gives us the shader and the material viewer as such:



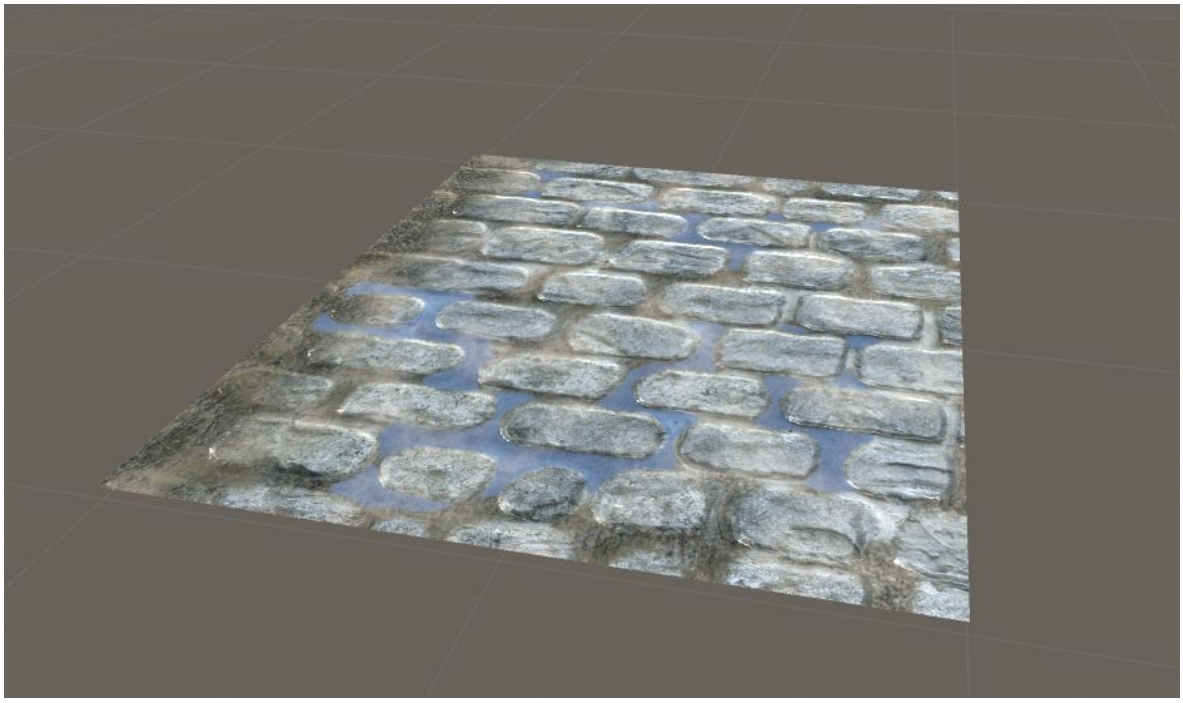
5. Finally, we will apply the water to the model. Begin by applying the diffuse and mask.Note that there is no corresponding normal for the water - and that’s fine. Additionally, you can see the water is a transparent diffuse texture - this is supported by the shader. For a diffuse-only setup, simply leave the corresponding normal slot blank. You should have the material should look like this:



6. Finally, we will add some repeating tiles. Sand will be used to make the road look more dusty and weathered overall. We can do this by using the “tiling” field in the sub material. This affects the entire sub material, resulting in the final material:

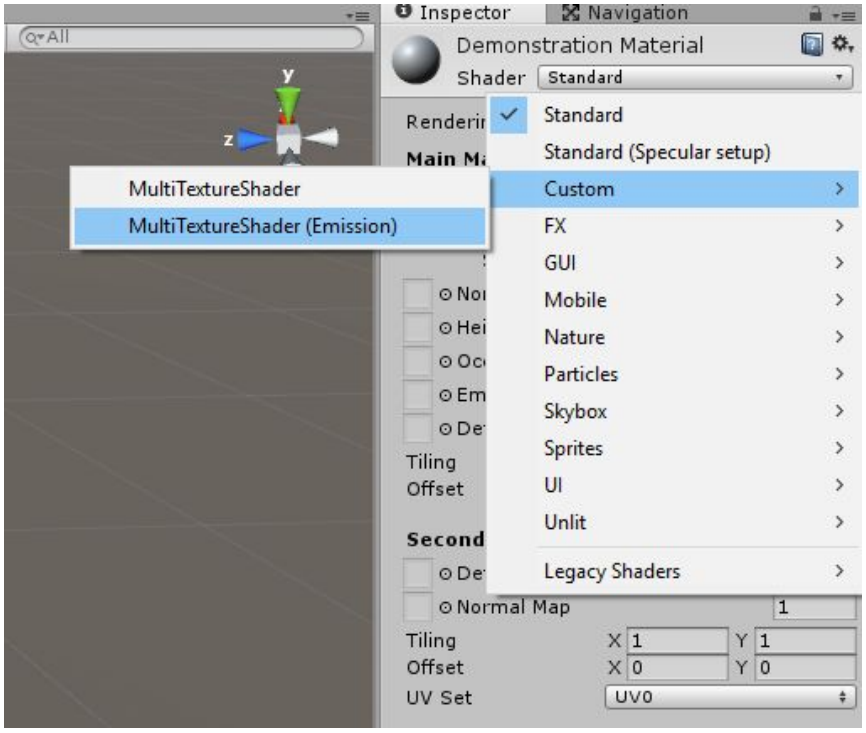


7. This concludes the material tutorial! All four of the sub material slots have been used. The resulting material, applied to a flat plane, appears as such.

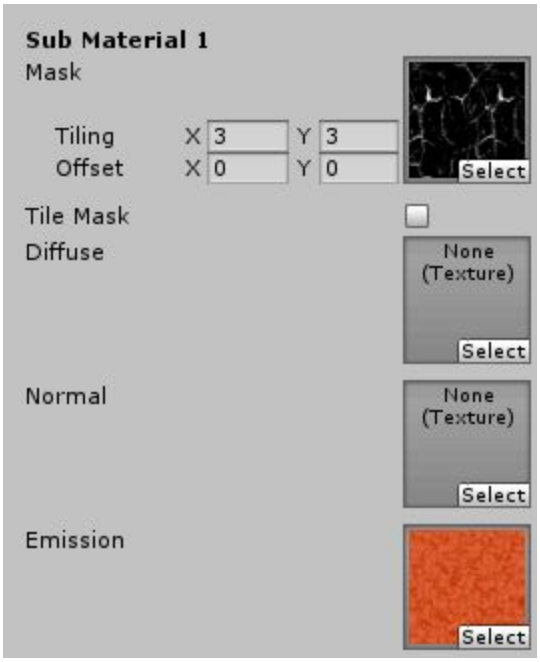


Emission Tutorial

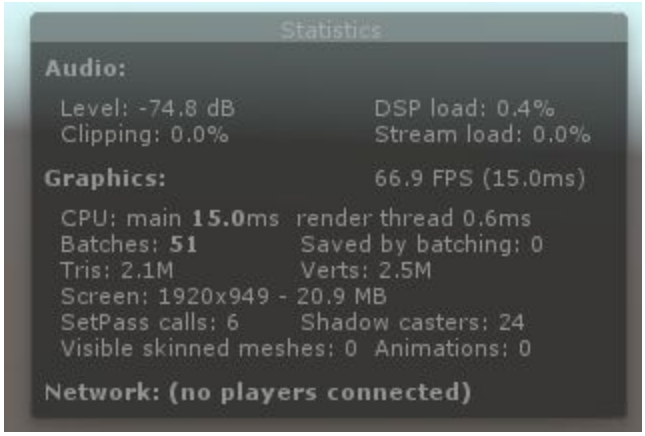
1. Emission textures work very similarly to the procedure above. Simply select the “Emission” shader variant from the drop-down menu to begin.



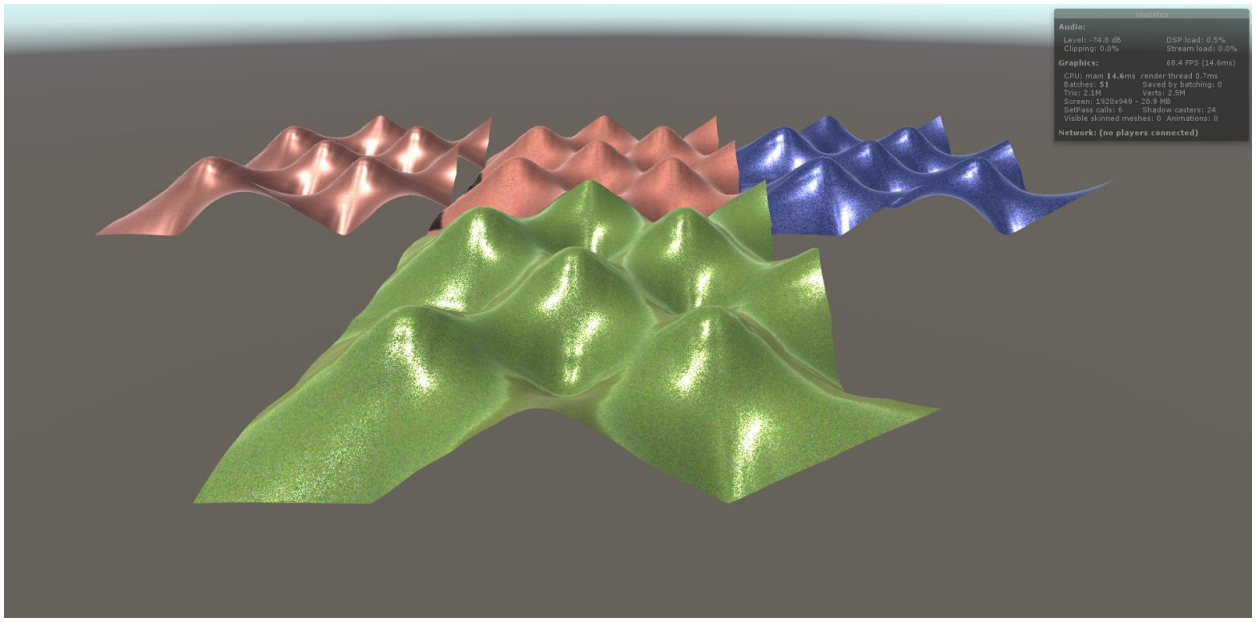
2. Note that this shader variant only has 3 sub material slots, unlike the 4 slots the diffuse contains. The use of this shader, however, is fundamentally the same as the above tutorial. Simply apply the emissive textures to the sub material used, and utilize the masks to control what renders to the final texture.



Performance



Tested on a GTI 970, Intel i7, 4 GB Ram. Testing with 4 meshes (~5,000,000 Tris each) totalling 2.1 Milling tris. Each mesh had four 2048^2 diffuse and mask sets applied to them. Each material was different, disabling batching by material completely. Framerate never fell below 60, and would flucatuate from 63 to 90 frames per second depending on lighting conditions.


Statistics
Audio:
Level: -74.8 dB
DSP load: 0.5%
Clipping: 0.0%
Stream load: 0.0%
Graphics:
68.4 FPS (14.6ms)
CPU: main 14.6ms render thread 0.7ms
Batches: 51
Saved by batching: 0
Tris: 2.1M
Verts: 2.5M
Screen: 1920x949 - 29.9 MB
SetPass calls: 6
Shadow casters: 24
Visible skinned meshes: 0
Animations: 0
Network: (no players connected)