SHADER FORGE

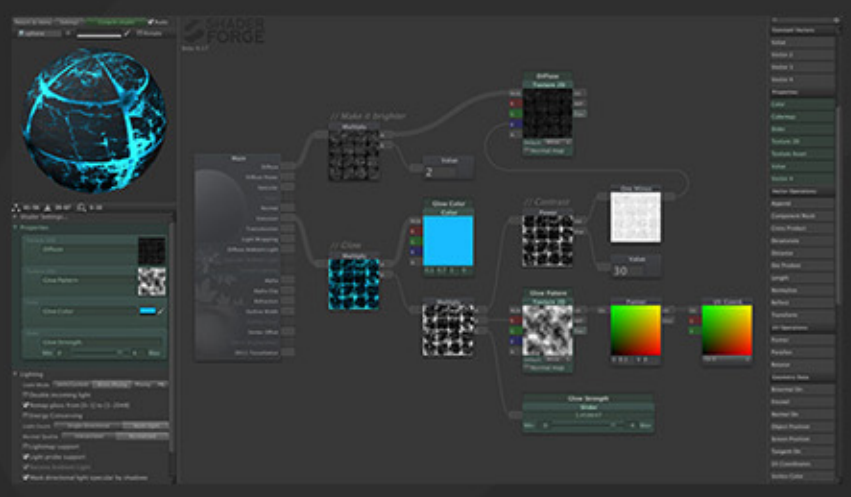
Quick Start GUIDE

Hello!

This PDF is meant as a little guide for getting started with Shader Forge.

Shader Forge is a node-based shader editor for Unity, making it much easier to create your own shaders, **no code required**!

I'll assume that you already have Shader Forge installed. If not, you can get it at **https://www.assetstore.unity3d.com/en/#!/content/14147**



Wait, what is a shader?

Shaders are used to draw your assets in the game. They are responsible for determining the color of the surface that is currently being rendered, using information such as textures, light sources, light color, mesh data, and so on, to calculate that color.

Unity comes with a few built-in shaders, some specialized ones for particles and effects, and some more general ones, like the Standard PBR shader that shipped with Unity 5.

Why Shader Forge?

While the default shaders in Unity certainly are good for what they are, you often want to get a more distinctive look on your game, or to achieve some very specialized effects. If you want that, you'll need to write your own shaders.

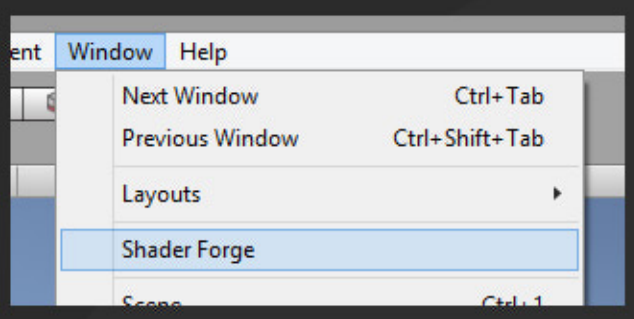
However, writing your own shaders is hard and very difficult to learn - especially for people who aren't programmers. This is where Shader Forge comes in.

Shader Forge allows you to create shaders, without knowing how to code!

© 2015 Neat Corporation

Creating a very basic diffuse shader

Let's jump right into the action!
First, open Shader Forge by going to Window -> Shader Forge

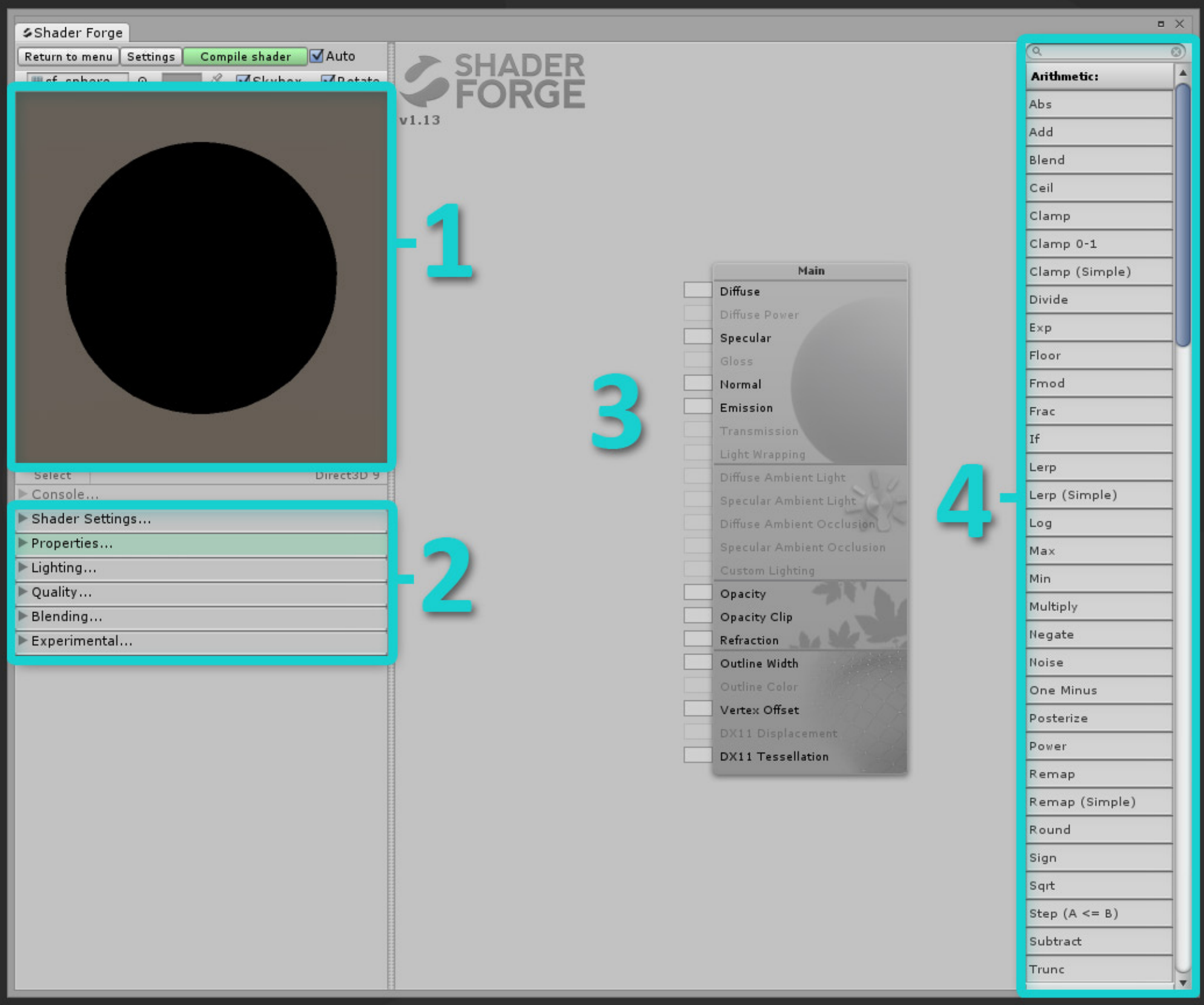


Press the New Shader button and give your shader a nice name, such as "MyBasicShader", then select the "Lit (Basic)" preset, and save it somewhere in your project!

If the window is too small, resize it by dragging the bottom right corner.

© 2015 Neat Corporation

You should now see the main interface of Shader Forge:



1. 3D preview, to see how your shader looks in action
2. Shader settings, to configure the way your shader renders
3. Node area, to create and manipulate nodes
4. Node browser, to find all the nodes you need

© 2015 Neat Corporation

At the moment, there's only one node placed, the Main node. This is the heart of all shaders - where all your nodes will end up connecting to.

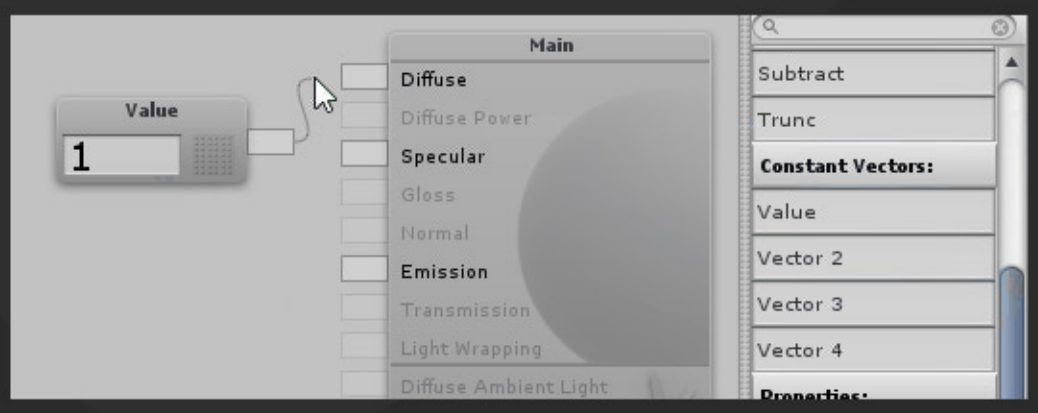
You can pan around the camera by clicking and dragging the left mouse button in the node view.

Now, let's create a super simple shader, that is simply white and affected by light sources. Create a Value node, either by dragging it from the panel on the right into the node view, or by holding the "1"-key on your keyboard and clicking in the node view.

On the node, make sure its number is set to 1, this will give you a white color

Drag the left mouse button from the output connector on the Value node, and release on the Diffuse input connector of the Main node.

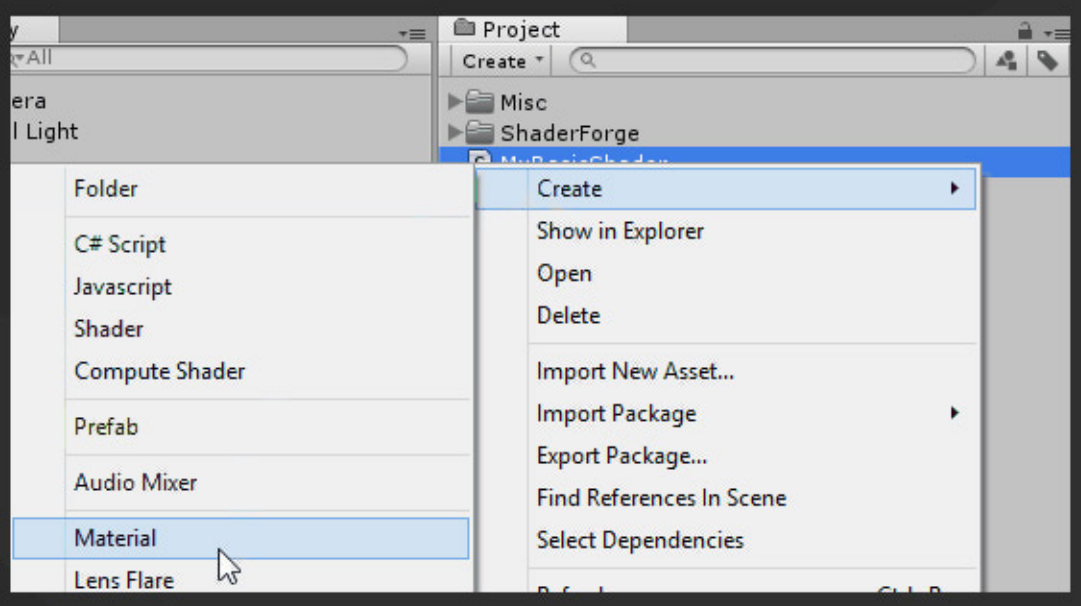
They should now be connected and your shader should compile!



The shader is now ready to be used!

However, you also need to create a material that is actually using the shader you just created.

To do that, right-click on the shader in your project window, and go to Create -> Material.



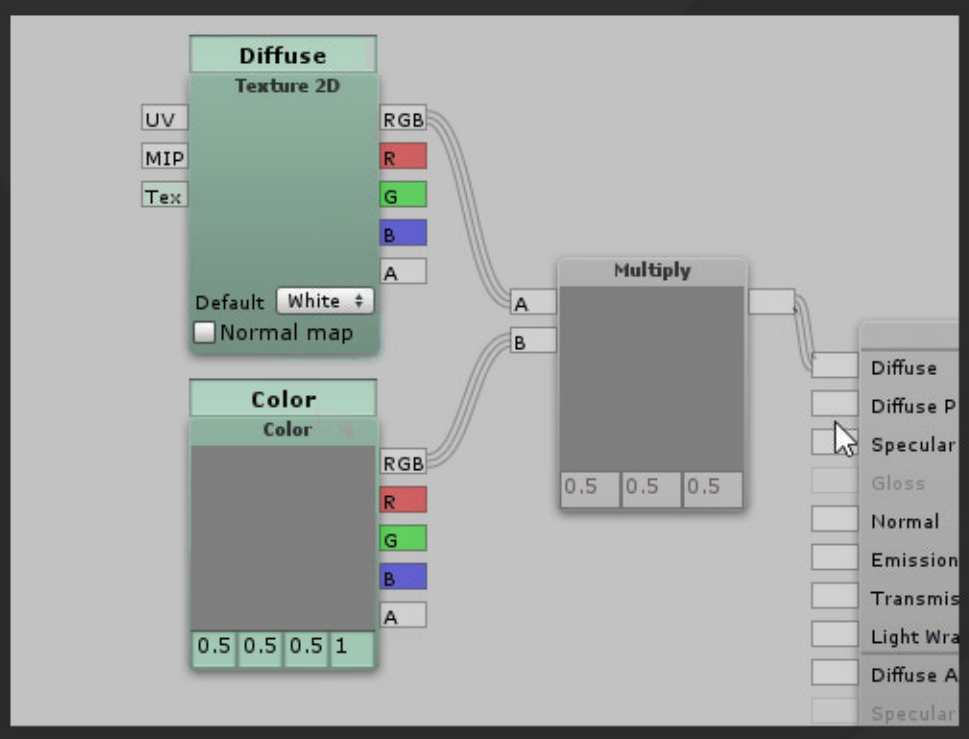
You now have a material, using your newly created shader, ready to be used in-game!

© 2015 Neat Corporation

Textures & properties

Now, let's add some properties to make our shader tweakable on a per-material basis!

1. Select the Value node and delete it by pressing delete or backspace
2. Create a Color node, and enter a nice name at the top of the node, such as "Color"
3. Create a Texture2D node, and enter a nice name at the top of the node, such as "Diffuse"
4. Create a Multiply node
5. Connect the RGB output connector of the Texture2D node into the "A" input of the Multiply node
6. Connect the RGB output connector of the Color node into the "B" input of the Multiply node
7. Connect the output connector of the Multiply node into the Diffuse input of the Main node



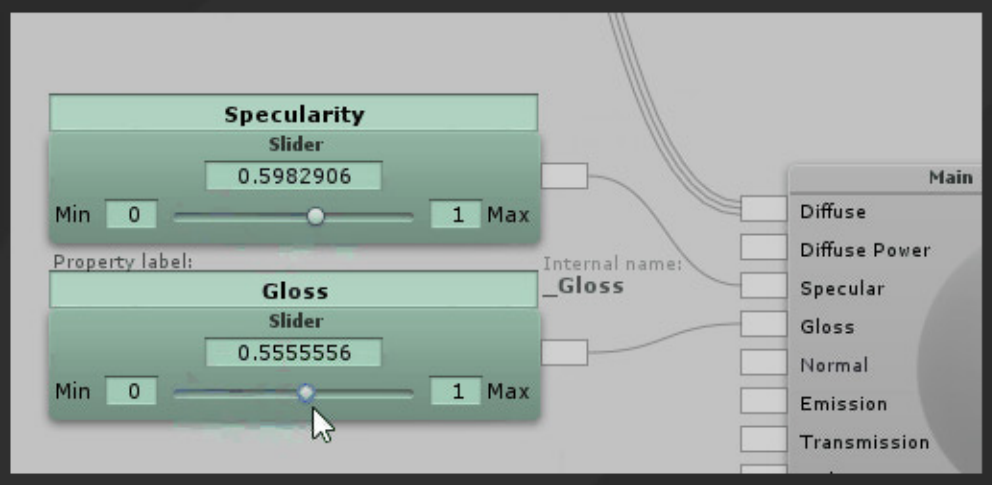
If you now go back to your material in the project view, you should be able to assign a texture to it, and to change the color of it. The color will tint your texture with the specified color. You can also duplicate the material, and have a different color on other objects, while still using the same shader!

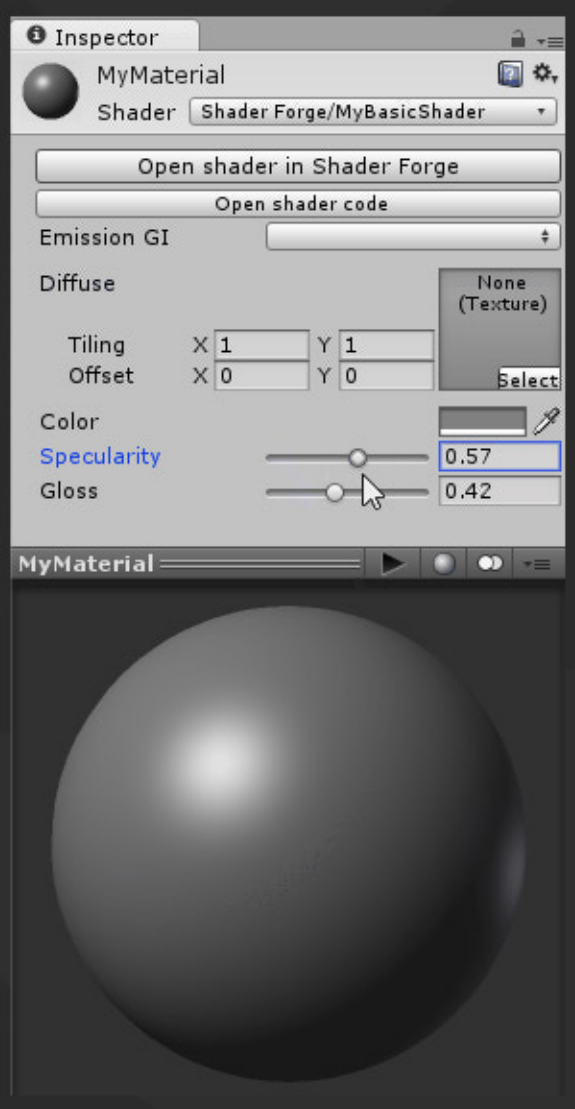
As you may have noticed, the Color node and the Texture2D node were both green. This means that they are properties, which in turn means that you can modify them and give them unique values, on a per-material basis.

© 2015 Neat Corporation

We might also want to add a slider to control the shininess of our object, which is another property.

1. Create a Slider node, and name it something like “Specularity”
2. Connect the output of the Slider node, into the Specular input of the main node
3. Create another Slider node, and name it something like “Gloss”
4. Connect the output of the Slider node, into the Gloss input of the main node





If you look at your material, you now have two additional sliders that you can use to tweak it.

One slider controls the specularity, which is the intensity of the specularly reflected light. The other slider controls gloss, the smoothness of the surface.

© 2015 Neat Corporation

This is just a tiny fraction of what you can do with Shader Forge!

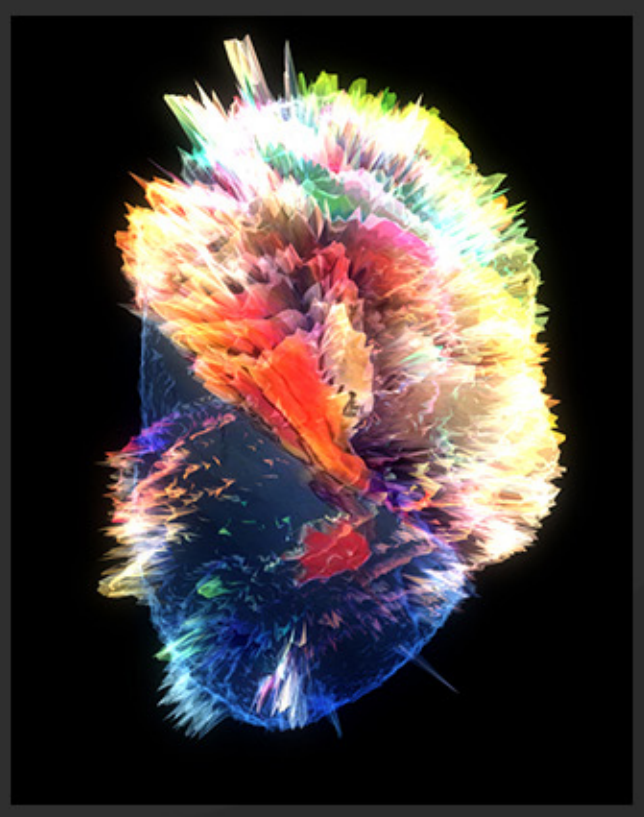
You can do pretty much whatever you like, from the most realistic PBR shaders, to the most super-customized stylized shaders you can think of!

If you want to dig deeper, here are a few links to get you started:

Node Documentation: http://acegikmo.com/shaderforge/nodes/
Wiki: http://acegikmo.com/shaderforge/wiki
Forum Thread: http://forum.unity3d.com/threads/shader-forge-a-visual-node-based-shader-editor.222049

Happy shading :)

// Joachim



© 2015 Neat Corporation