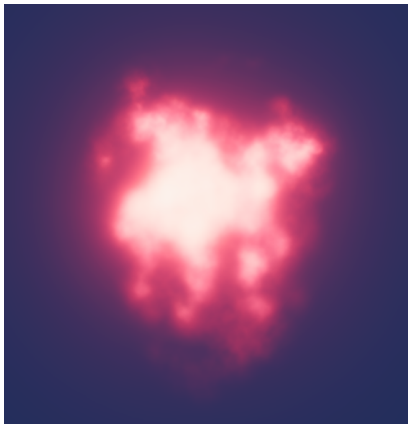
Unity Shader Graph Workshop
Part 2: Diablo 3 Two-Texture Shader

Dr. Micah Schuster

Summer 2021

School of Computing and Data Science

Wentworth Institute of Technology



Wentworth INSTITUTE OF TECHNOLOGY

VFX of Diablo 3

I've based this part of the workshop on creating the two-texture shader from Julian Love's GDC presentation:

https://www.youtube.com/watch?v=YPy2hytwDLM

The overall shader will be very general, which makes it a bit more complicated.

For example, we'll be using two textures in our shader, but each textures has a RGB component and Alpha component. Each Texture will be able to scroll at different speeds and tile at different scales.

This variation allows us to create very different effects with slight changes to the base RGBA textures.

Wentworth INSTITUTE OF TECHNOLOGY

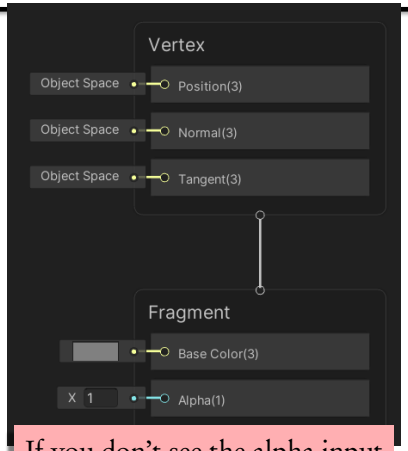
Initial Shader Graph

We'll use the same basic unlit shader as Part 1.

Again, our focus will be on the fragment shader.

Reminder: Each of the little circles on the left are inputs, and the boxes on the left are "default" values. We can either change them or hook other nodes into those values.

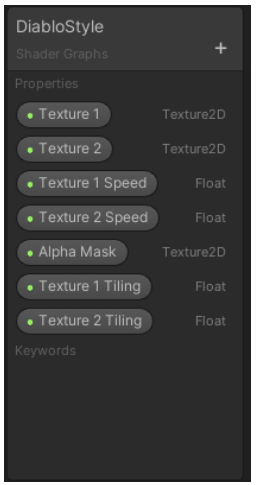
Start by creating two Sample Texture 2D nodes (and add a noise texture to each one).



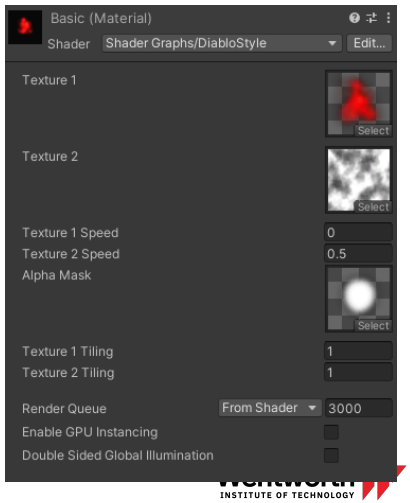
If you don't see the alpha input in the fragment shader, you'll need to set the shader to "Transparent" in the global properties.

Shader Graph: Properties

We're starting to get a larger number of parameters that can be adjusted and it's not ideal to open the shader to change them.



Creating a property adds the value to the blackboard (left) so that the value can be changed in the material (right)

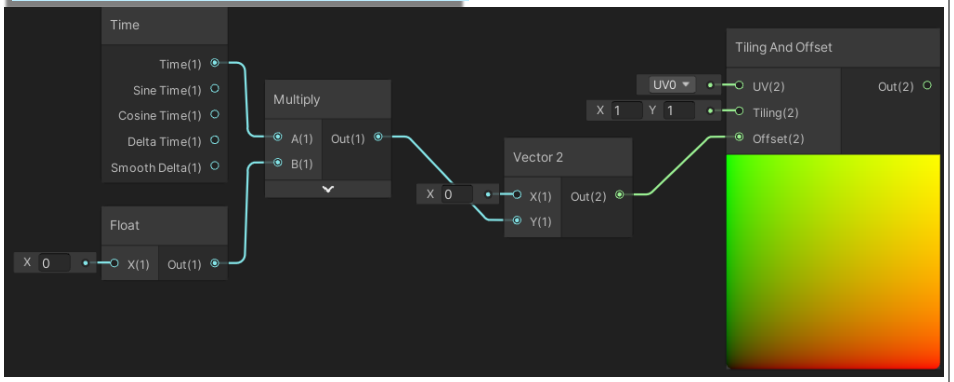


Wentworth INSTITUTE OF TECHNOLOGY

Scrolling Two Textures

Using a Time node with a Tiling and Offset node allows us to "scroll"our UV coordinates, thus making our texture appear to scroll over time.

We're only going to scrolling up (negative Y), so the float must be a negative value.

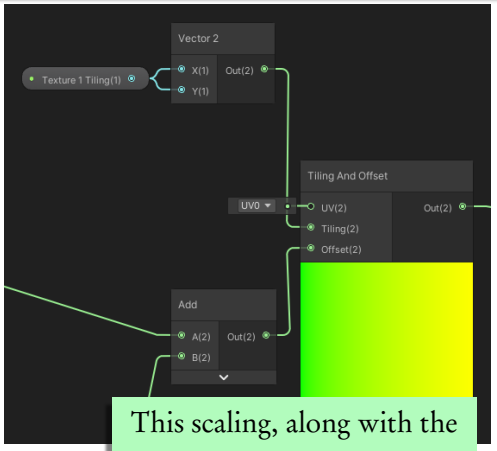


Scaling Two Textures

The Tiling and Offset node also includes a Tiling input.

If your textures are tillable (like the provided noise textures) you can easily "zoom in or out", changing the scaling of the texture.

Just like adjusting the speed of the scrolling, create a float or vector2 and plug it into the Tiling input.



This scaling, along with the scrolling creates a lot of variation with only a few textures.

Basic Shader Outline

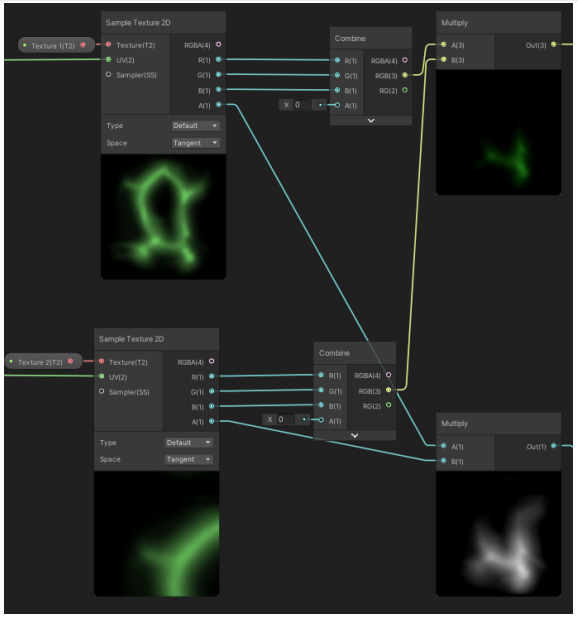
Alpha Channel

$$\left(\text{Alpha 1} \times \text{Alpha 2} \right) \times 2$$

Color Channel

$$\left(\text{RGB 1} \times \text{RGB 2} \right) \times 2$$

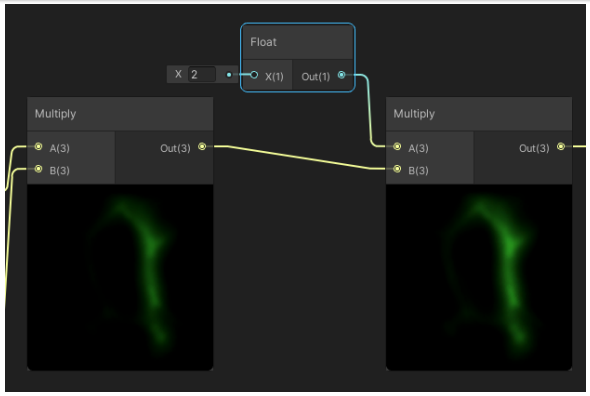
RGB and Alpha Setup



Note that each texture uses the individual components. The RGB are rejoined and multiplied (at the top) and the alpha channels are multiplied separately at the bottom.

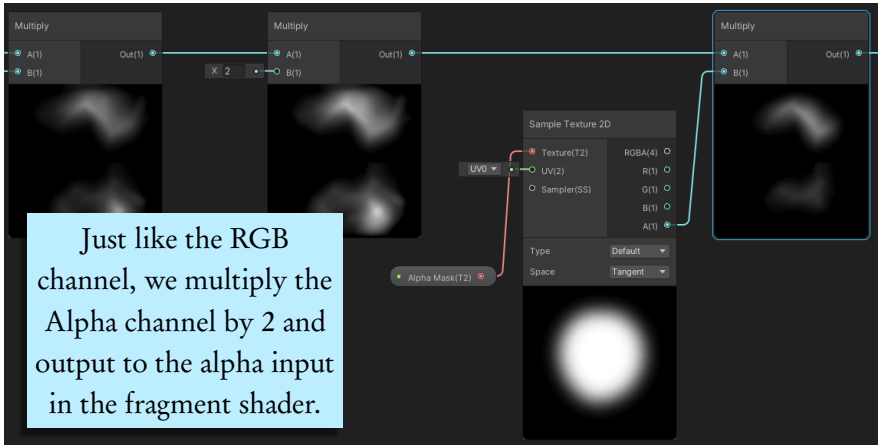
You're free to use whatever textures you want here. Creating four textures, 2 alpha and 2 RGB, can allow for even more variation.

RGB Multiply



Now we simply multiply the RGB channel by 2 and output to the color input of the fragment shader.

Alpha Multiply and Mask



Just like the RGB channel, we multiply the Alpha channel by 2 and output to the alpha input in the fragment shader.

I've included another texture, a mask. This softly fades out the edges to transparent so we don't see any edges of the texture. Masks like this can be very important for this kind of shader.

Coming up:

Part 3:

- Adjust our shaders to be useful in particle systems