ME 557 Assignment 4

Team: Nigel Lee, Chloe McPherson, Anupa Prudhivi

**Problem 1**

In the first problem, we were asked to combine three textures (a landscape, a color gradient and an animal) in such a way that three images appear together in a single image. We were also required to blend these three textures with different operations.

The following are the three textures used in problem 1:

Landscape Color Gradient

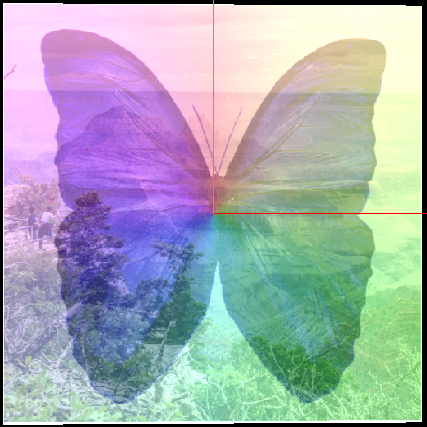


Butterfly

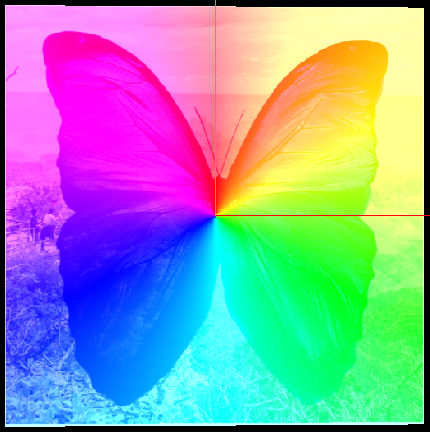
The first blend we applied was:

color = 0.1 \* pass\_Color + 0.3\*tex\_landscape + 0.3\*tex\_color\_gradient + 0.6\*tex\_butterfly;

The result of this blend mode is shown in the following figure.

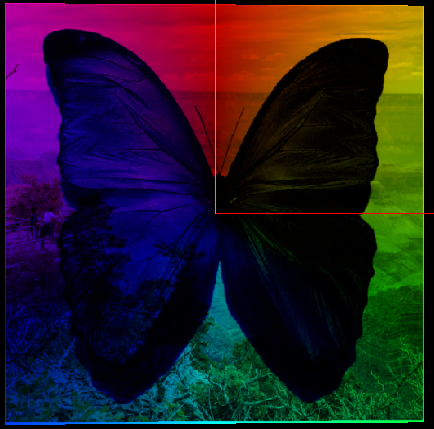


The second blend we tried was color = tex\_landscaper + tex\_color\_gradient \* 0.8\*tex\_butterfly; and the result of this is as follows.



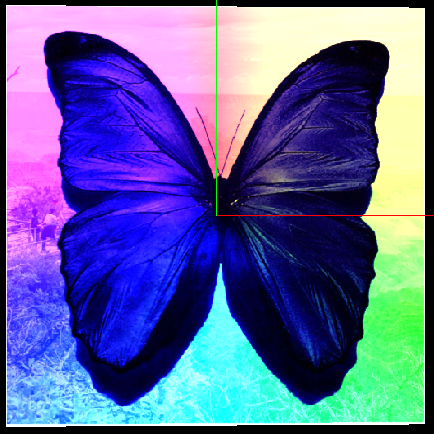
The third blend we tried was: color = tex\_landscape \* tex\_color\_gradient\*tex\_butterfly

The result of it we obtained is shown below.



Also we tested the blend mode by giving

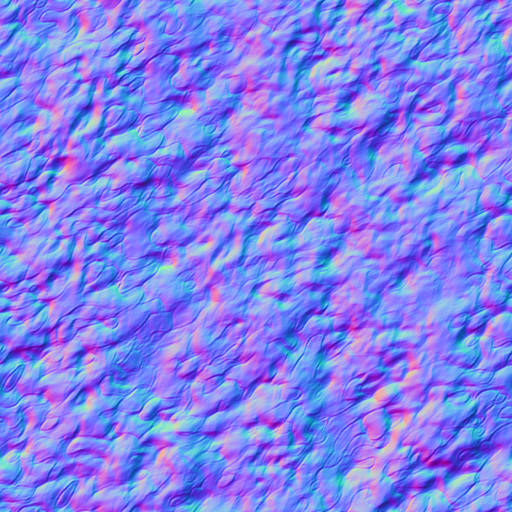
color = (tex\_color\_gradient) \* (tex\_butterfly + tex\_landscape); and the result is as shown in the below figure.



**Problem 2**

For the second problem, were asked to combine a landscape and noise map together through mapping every pixel on the noise map as a vector displacement.

Below are the landscape and noise map that we combined.

Landscape Noise Map of the Sun Texture

In order to consider every pixel value of the noise map as a vector displacement, the noise map adheres to the following rule:

Red from (0-255) receives X (-1.0, 1.0)

Green from (0-255) receives Y (-1.0, 1.0)

Blue from (0-255) receives Z (0.0, 1.0)

To combine the landscape and noise map we used the following code:

void main(void)

{

// This function finds the color component for each texture coordinate.

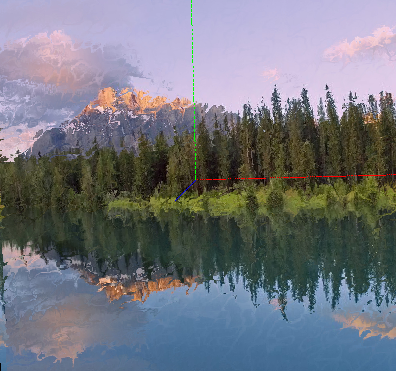
vec2 noiseVec;

noiseVec = normalize(texture(texture\_foreground,pass\_TexCoord).xy);

noiseVec = (noiseVec \* 2.0 - 0) \* 0.050;

color = texture(texture\_background, (pass\_TexCoord + noiseVec));

When we scale the noisevec = (noisevec \* 2.0 - 1.0) \* 0.02 below is the result



When we changed the scaling factor to 0.05, this is the result:

