ME 557

Homework 4: Problem 1

Group Members: Adam R. Kohl, Jiazhong Zhou

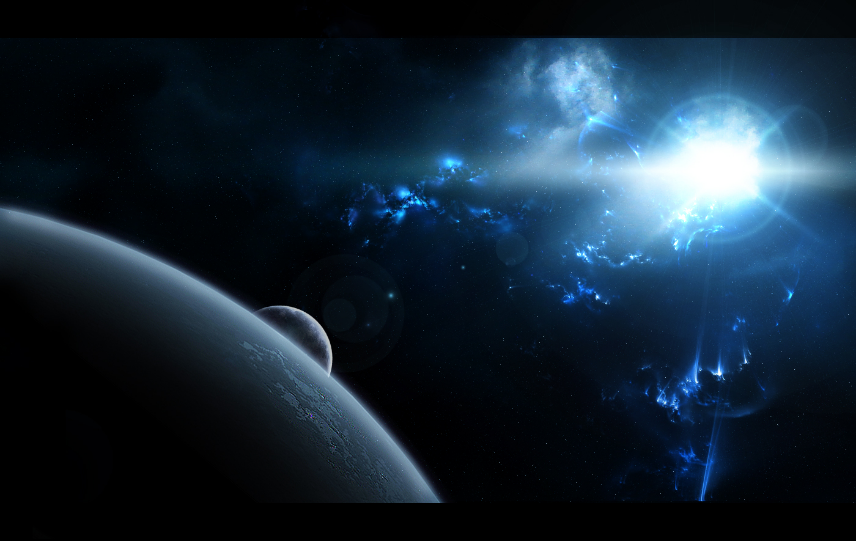
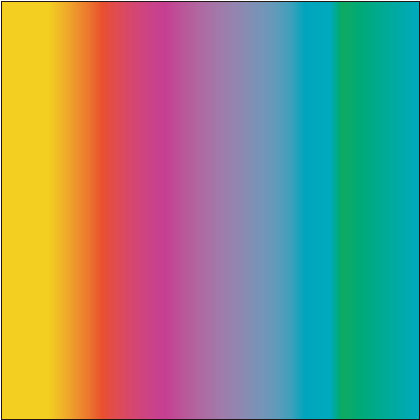
The code has been modified are: Main\_multi\_texture.cpp

Texture.cpp Texture.h

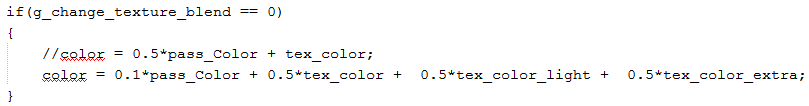
GLApperance.cpp GLApperance.h

The code has been created are: Triple\_texture.fs Triple\_texture.vs

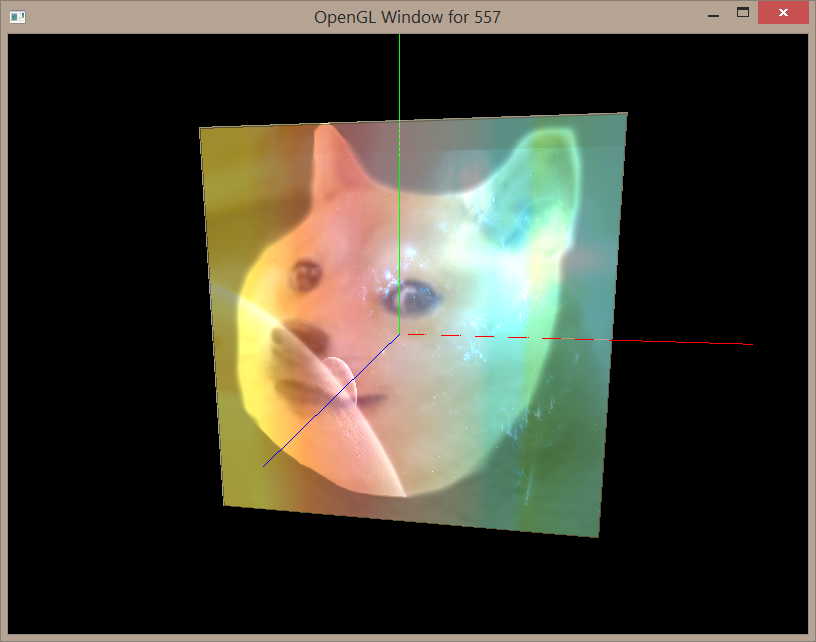
The three images used for problem 1 are doge, rainbow color gradient, and a universal landscape as shown in the below figures.



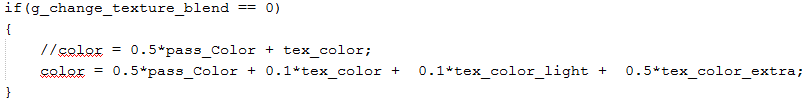
The first blend model has a highlight on those three images, therefore the blend function is defined as:



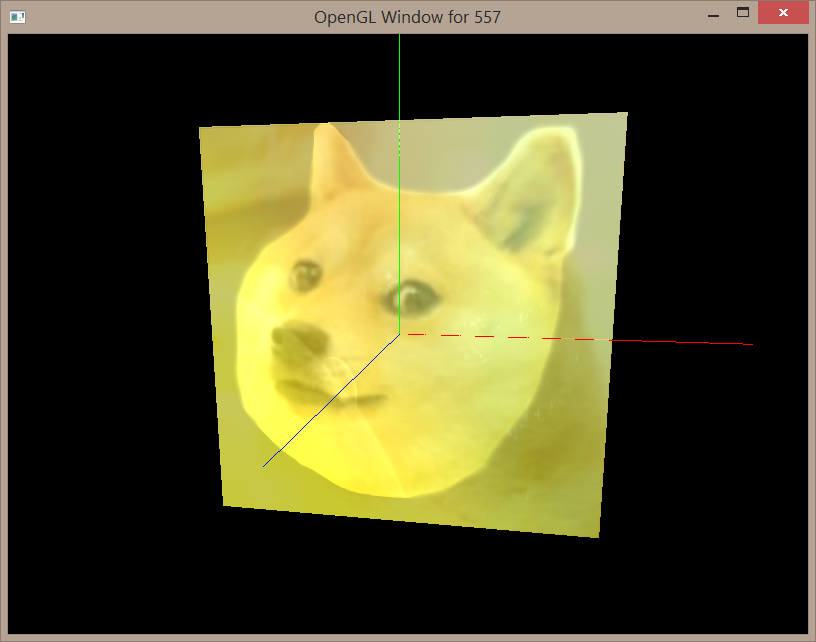
The image generated by the program is:



The second blend mode has a highlight on the background color, therefore the blend function is defined as:



The image generated from the program is:



ME 557

Homework 4: Problem 2

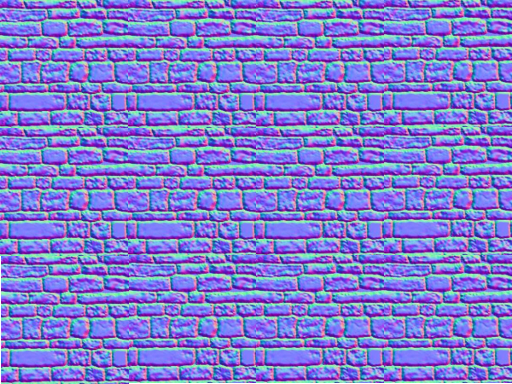
Group Members: Adam R. Kohl, Jiazhong Zhou

**Background**

There were two texture images implemented in the rendering. The first image was a landscape photo, which was overlaid by normal map image. The normal map image pictured in Fig. 2 resembled that of a wall of bricks.

**

*Fig. 1 Texture Background Image*

**

*Fig. 2 Texture Foreground Image*

**Vector Manipulation**

Every pixel value was considered as a vector the displacement, therefore the shifting distance for vertex normal vectors were stored in the normal map. Variable noiseVec in Fig. 3 uses the color information from the noise map to retrieve the vector displacement. In more detail, the texture coordinates are (0,1), therefore when both x and y components are inputted into the fragment shader code the following are the result:

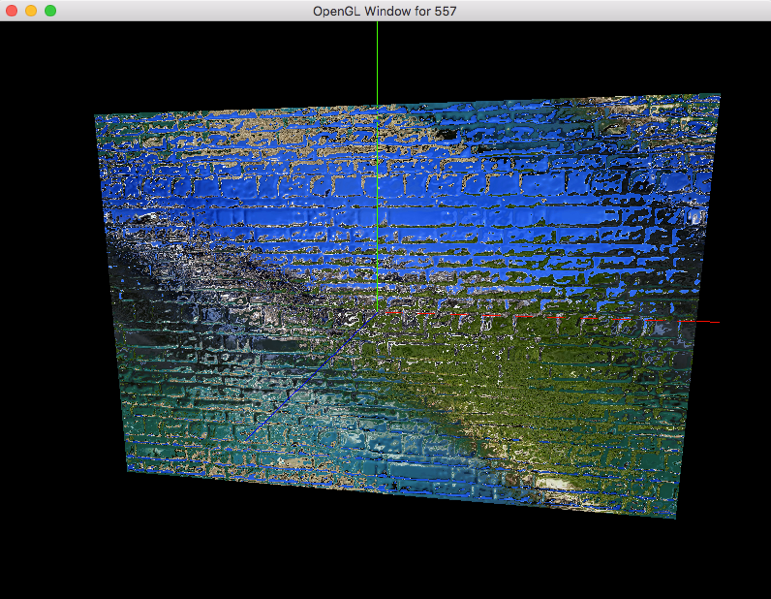
* Red from (0-255) 🡪 X (-1.0, 1.0)
* Green from (0-255) 🡪 X (-1.0, 1.0)
* Blue from (0-255) 🡪 X (0.0, 1.0).



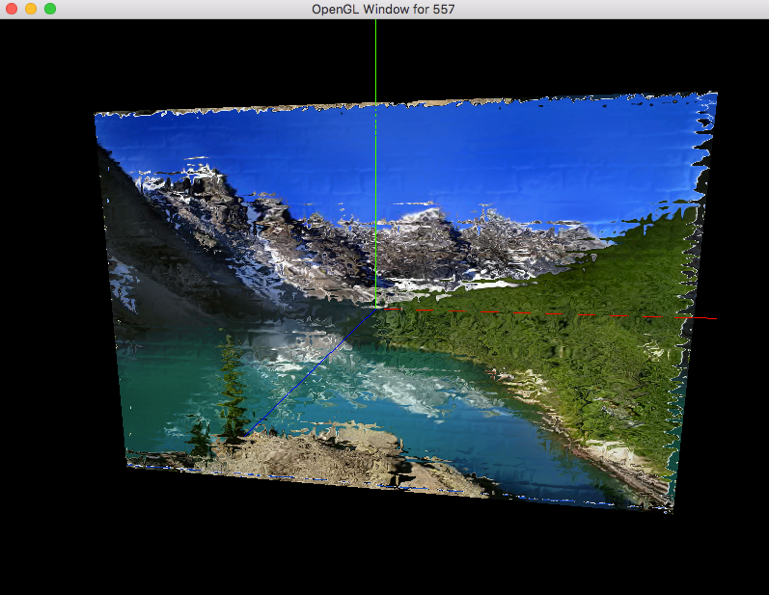
*Figure 3. Screenshot of Fragment Shader Code*

**Observations**

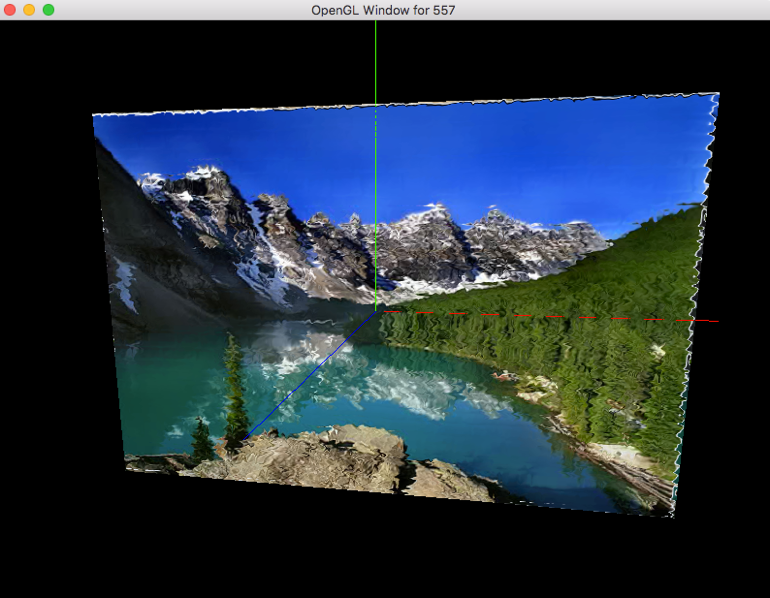
The following three figures illustrate how the image’s final result will vary when the scaling factor is manipulated. The scaling factor can be seen in Fig. 3 as the value of 0.0085. Figure 4 can be seen to have the pattern of the noise map with a high scaling factor. This is because the vector displacement is increased, therefore leading to more distortion in the landscape image. Once the scaling factor is brought down to a smaller value as shown in Fig. 6 the original landscape image becomes more clear.



*Fig. 4 Scaling factor of 0.25*



*Fig. 5 Scaling factor of 0.025*



*Fig. 6 Scaling factor of 0.0085*