COSC4370 HW3 – 3D Viewing and Shaders

March 18, 2022

Created by Yusra Shaikh

1. Problem

This assignment requires the implementation of OpenGL's Phong shader model. This technique allows for 3D viewing.

2. Method

The method used to obtain the results were through the completion of the get view matrix function in Camera.h, and the projection matrix in main.cpp. After, in the main.cpp, the vertex as well as the fragment shaders for the Phong model were coded to create a cube. Lastly, the stubs for the shaders were created in the phong.vs and phong.frag files. This resulted in the formation of a cube.

3. Implementation

Part 1.

In the camera class, the GetViewMatrix() function was implemented through the lookAt function. This allowed for the view matrix to be created. And since the front and up vector of the camera object was already known, there was only a need to implement them into the lookAt function. Furthermore, the camera was placed at a certain position. Then the position of the camera was subtracted by the vector of the object allowing for the camera to point to the object. Then the vector was needed to become perpendicular.And lastly, it was important to take the product between the perpendicular vector and the position the object the camera was pointing towards.

Part 2.

Next, the projection matrix in the perspective function was used, which allowed for the camera to get the projection of the perspective. The purpose of the perspective function is so that each of the vertex coordinates can be divided by the width, which makes the object change its appearance depending on how close or far away it is when the user looks at it.

Part 3.

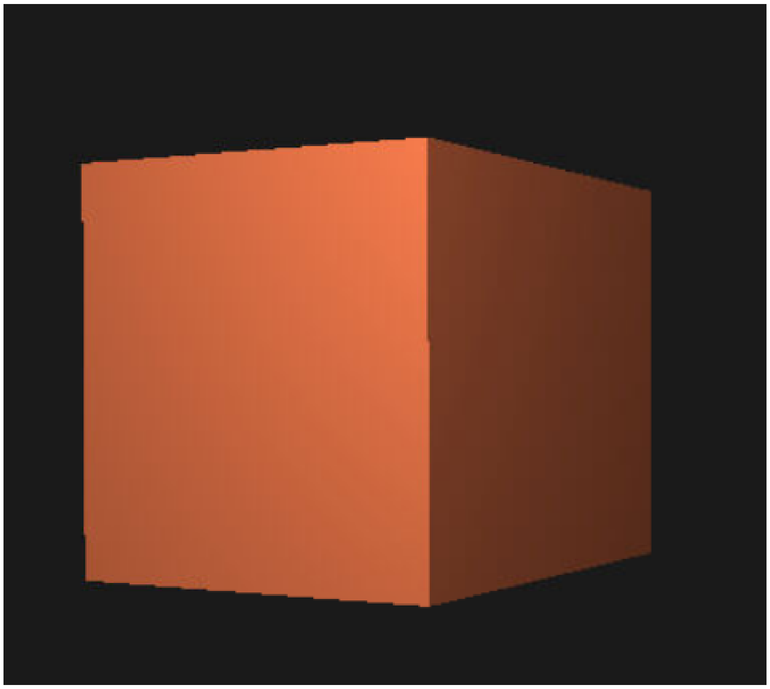
Continuing on, the phong.frag file needed to be edited. And the most important part of this file was to create the color of the object. This was done by creating a variable called str, and multiplying it by lightColor. In doing so, this was able to result in the creation of the color of the object.

Part 4.

Adding on, the file phong.vs was edited to calculate the lighting, as well as the direction of the vector that was between the light and the position. There was also a need to apply the functions transpose and inverse to result in a vector. Basically, we needed to normalize the vector as well as the light, which allowed us to lighten the model. Because initially, the cube was given color, but the rest of the canvas was a bit too dark, so it was important to apply these calculations to lighten it, and create the results.

4. Results

The results from the calculation created a cube through the use of the Phong shader model. This assignment allowed for learning OpenGL more deeply, and allowed for the creation of the replica of the initial picture that was given in the assignment.



5. Sources

-https://learnopengl.com/Lighting/Colors
-https://learnopengl.com/Lighting/Basic-Lighting
-https://learnopengl.com/Getting-started/Coordinate-Systems
-https://learnopengl.com/Getting-started/Shaders
-https://learnopengl.com/Getting-started/Transformations
-https://learnopengl.com/Lighting/Review