# COSC 4370 Homework 3

By: Ivan Madrid

**Assignment Problem:**

Implement the 3D viewing and Phong shading model to shade a simple cube by completing the GetViewMatrix() function in Camera.h, the projection matrix in main.cpp, and write the vertex and fragment shaders by completing phong.vs and phong.frag.

**Assignment Implementation:**

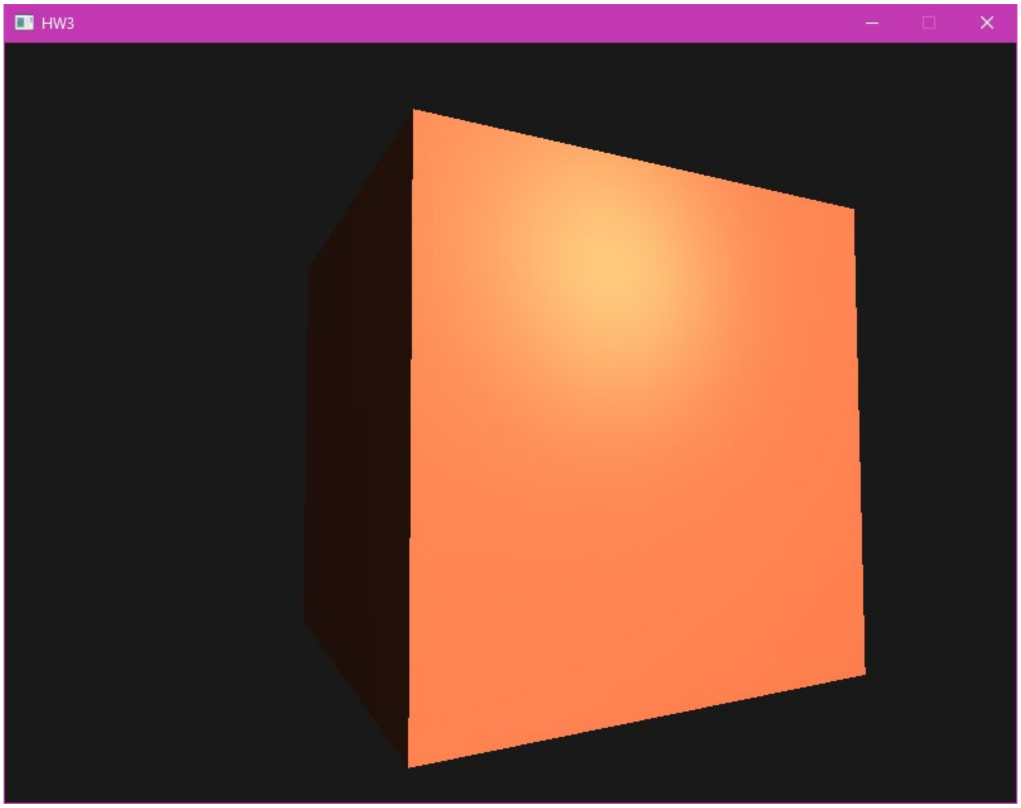
For implementing both the camera and Phong shading model, I referenced the camera and basic lighting tutorials from LearnOpenGL. To implement the GetViewMatrix function for the camera, I converted the yaw and pitch Eular angles to a 3D direction vector. I had the view matrix first be the lookAt matrix with the position being the 3D direction vector. Then the function will return the lookAt matrix with the position being the camera position, the target being the camera position plus the camera front, and up being the camera up.

In main.cpp, the only implementation was having the projection matrix be the perspective projection matrix from the glm perspective function with the parameters being camera.Zoom, 800 / 600, 0.1, and 100.

For the implementation of the Phong shading model, we must create the vertex and fragment shaders. For the vertex shader, we correctly set the position matrix, the fragment's position, and the fragment's normal. Then in the fragment shader, we divide the Phong shading model into its three components, ambient, diffuse, and specular. The ambient component is the ambient strength times the light color, with the ambient strength set to 0.1. The diffuse component is used by determining the max of the dot product of the normal and the light direction and 0, and then multiplying it by the light color. The specular component is the max of the dot product of the view direction and reflect direction and 0 times the specular strength and the light color where the specular strength is equal to 0.5. The last step is to add the ambient, diffuse, and specular together times the object color and then set it as the object's color.

**Results:**

The resulting application is a camera that we can control to view a simple cube with the Phong shading model applied to it, with the camera acting as the light source. The application was able to reproduce the image given in the homework instructions.



Sources:

https://learnopengl.com/Getting-started/Camera

https://learnopengl.com/Lighting/Basic-Lighting