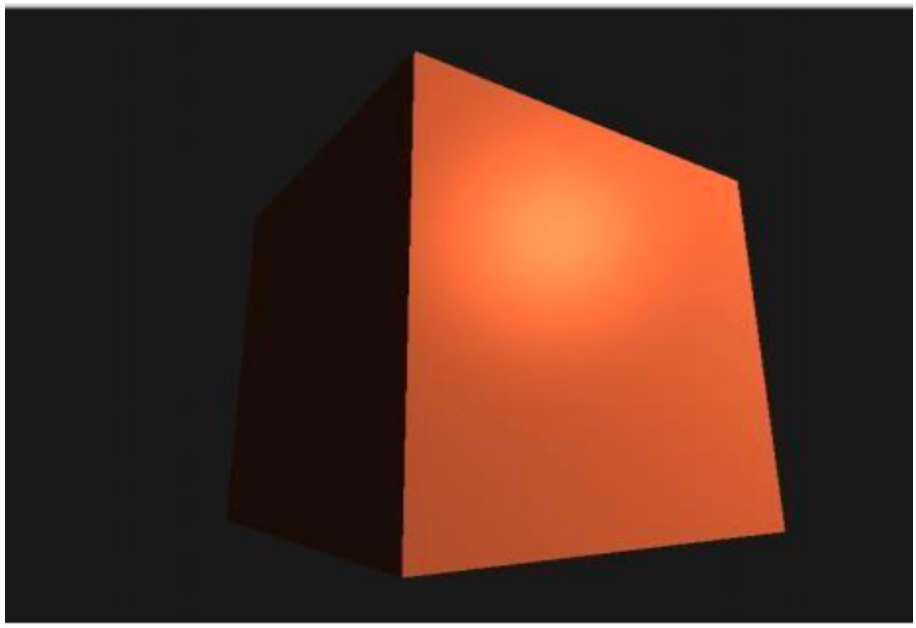
Sham Hintolay
1618608

Homework 3 Report

1. Problem

In this Homework, we are tasked with recreating the following images. We need to recreate the following image using 3D viewing and the phong shading model.

I used the resources available at learnopengl.com to assist in this homework, specifically the camera tutorial and the basic lighting tutorial



Sham Hintolay
1618608

2. Method

In order to succeed in the assignment, we only made changes in the GetViewMatrix function within the Camera.h file, Initialized a projection matrix within the main.cpp file, and filled the phong.frag and phong.vs files for the fragment shaders and vertex shaders.

In order to fill the GetViewMatrix() function, we initialized a view matrix of mat4 data type and used the glm:: LookAt function to get the matrix, then we returned it.

The projection matrix was initialized using the glm:: perspective function.

In order to create the necessary lighting effects, we combined ambient, diffusive, and specular lighting to correctly alter the color of the fragment in such a away that mimics natural lighting.

3. Implementation Details

Sham Hintolay
1618608

In order to complete the GetViewMatrix() function we need to initialize our view matrix with the LookAt function by using the correct parameters. The paramters needed are the position vector, target vector, and the up vector. The Camera system that's already outlined already contains the variables we need to plug in. For the position parameter we simply plug in the camera's position, which is expressed in a vec3 datatype. The target parameter can be represented by the front relative to the cameras position, so we use position + front for target, then we simply plug in the camera's up vector for the up parameter. After doing all this we return the view matrix that was created.

Next in main.cpp we initialized the projection matrix. We used the perspective() function and filled the given parameters of field of view, aspect, near, and far, all of which are supposed to be float values. The field of view represents the width of the perspective frustum, and changing this value gives a view that zooms in and out, 45 is the default value so I plugged in 45. Aspect simply is the aspect ratio of the view, I used a default value of 800/600. Near and far are used to specify the near and far planes of the perspective, we plug in a minimal value for near and a arbitrary large number for far so that all the coordinates in between are drawn.

Now we will examine the light effects made within the Phong.frag file. To add ambient light, we take the light color

and multiply it by a small ambience constant. We now have a ambient vector that we combine later.
To get the diffusive light, we need to use the fragments position and the normal, both of which are calculated and gathered from the vertex shader. Then, using the lights position and the fragments position, we are able to figure out the direction the light is coming from relative to the fragment position. By taking this light direction and multiplying it with the normal using dot product, we are able to figure out the diffusive impact of the light on the fragment. If the value returned is less than zero then we just use zero as the diffusive impact on the fragment. We then take this number, multiply it by our light color, and we have out diffuse vector.

Finally, to get the specular light, we need to figure out the view direction and the reflect direction. In order to get the view direction we need to find the normalized difference between the view position and the fragments position. Then we take this and do dot product again between the view direction and the reflection direction to find if theres light reflection at this positon, if the value is less than zero then we just use zero. The output of this is put to the power of 32, and this number represents a constant of shininess. Nowe we have a spec value, and we multiply it by the specular strength (a constant for specular intensity) and the lights color.

We now have our final values from the ambience, diffusive, and specular light. We add all these numbers up, and multiply them by our object color. Then we take this result and set it as the

fragments color. Every fragment position will have different results.

4. Results

This is the result :