Winston G. Grant

CS-330

Southern New Hampshire University

The project recreated the 2D image of a house into a 3D object. Objectives associated with creating this 3d object also included using simple primitives and keeping the use of polygons to a minimum. The primitives that made their way into the project are the cube, pyramid, and plane. They were used to build the house, the fence, the ground, and the grass. Starting with the house, I chose to use the cube because it was easy to manipulate to build each section of the house. These sections were the different floors, the garage and the two chimneys. Th cube was also able to be manipulated not only in size and location but also in rotation on its axis to make the different sections of the fence. The final object that utilized the cube was the grass areas. Next was the pyramid which was used to make the tops of the fence. This was the best choice because the tops could be changed in size to match the style needed without affecting the overall shape of the fencepost themselves. Lastly was a plane that was used to make the ground my objects are sitting on. A plane was used because all that was needed was a flat surface and it used the least polygons, in comparison to using a cube that would include unneeded sides. The texture for each of these objects was royalty free and was applied with the aid of the stb implementation and OpenGL’s uniform operations to evenly distribute the image on objects as needed. To better show the developed objects, the Phong lighting model was used to generate a combination of ambient, diffuse, and specular lighting. This allowed the creation of a vivid display of the 3D scene.

Accomplishing all these tasks would be quite cumbersome without an effective way to navigate the scene while objects were being placed. This called for the implementation of camera controls that allowed the user to effectively navigate the scene and view objects from different angles. The camera itself was locked on to the scene location. The movement of the world in relation to the camera was bound to the WASD and QE keys. This allowed the up, left, down, right, forward, and backward motions. The movement of the camera itself was bound to the user’s mouse. Together this allowed easy and effective navigation of the world space.

The overall modularity of this project can use some improvement. This is because the functions were not abstracted to their own classes. However due to extensive commenting of the code, an individual can easily extract what part of the code accomplishes a specific function a move from the point. My next task from this point would be to go back through and modularize my work.