Cody Faircloth

Dr. Kurt Diesch

CS 330

June 18, 2023

Project Submission: Reflection

The scene I chose had objects that could be modeled accurately with rectangular prisms and with a variety of triangles. I made several changes to the scene to create a more realistic representation of the desk. The desk was supported on one side by a built-in filing cabinet, but the left side had no visible support. I decided to create a side using the same texture as the desk to create more perceived structural integrity for the desk. There were two monitors and two speakers. These seemed like relatively redundant objects, so I reduced the number of these objects down to one each and placed them in the scene according to the image.

Navigation in my scene is handled by polling input from the mouse and keyboard, and utilizing the camera.h header file to properly process the input. Forward, backward, left, and right movement are possible by using the W, S, A, and D keys respectively. Similarly, I created vertical movement by creating UP and DOWN functions in the camera.h header file and programmed the Q and E keys to move the camera up and down respectively. Additionally, the mouse is used to move the camera laterally and vertically from a static position. This is done by utilizing the UProcessMouseInput() function in the code. This function handles mouse movement events and updates the camera depending on the magnitude and the direction of the motion of the mouse.

The custom functions in my project each performs unique tasks while calling upon each other to perfom these tasks. For example, the URender function creates regions for each object to be

meshed and rendered. Each region assigns a separate vertex array object with a specific scale, rotation, and translation value. URender also binds specific textures to each object. URender then calls UCreateMesh to draw the arrays assigned in URender. The use of these separate functions allows for additional objects to be created in the future and added simply by creating new regions and new vertex array. Additionally, code to function can be modified while maintaining the integrity of the other function. This allows for modular, well commented code, to be modified by other developers later easily and efficiently.