CS330 Final Project Reflection

Paul Mashett

The choices that were made for my 3D scene were based on accuracy in depicting the 2D scene that it was based on and time constraints for completing the project. The objects in the initial 2D picture were chosen based on variety of shapes and feasibility. The book contained basic shapes, namely rectangular prisms, that were combined to form the whole book object. This seemed like a good place to start since the tutorials used throughout the course were based on creating and manipulating a cube. Similarly, the door stopper, shortened to just “stopper” in the project’s code, was comprised of a rectangular prism and a triangular prism. Lastly, I chose the cylinder for my 3D scene because it is a complex shape that demonstrated the use of several simple chapes including a cylinder for the body, a cone for the neck, and another cylinder for the cap.

The user can navigate forward (positive z-axis), left (negative x-axis), back (negative z-axis), and right (positive x-axis) using the W, A, S, and D keys, respectively. The UP, DOWN, LEFT, and RIGHT directional keyboard keys can also be used to navigate in their respective directions. Navigating up (positive y-axis) or down (negative y-axis) uses the Q and E keys, respectively. The mouse cursor can be used to change the orientation of the camera to look up, down, left, or right. A slight modification was made that requires the user to hold down the left mouse button when using the mouse cursor to look around after discovering that this was preferable to the original method used. The mouse scroll wheel can be used to adjust the movement speed faster or slower depending on the user’s preference. Lastly, the escape key can be used to close the rendering window. These navigation components are made possible through the ‘processInput’ function in my program. This function relies heavily on the GLFW library for the ‘glfwGetKey’, ‘glfwGetMouseButton’, GLFW\_PRESS, and many other functions and parameters.

My code is modular by design, making the functions easily transferrable and usable in another program. There are functions to create the meshes for the various shapes, functions to create and draw shapes, set lighting, load textures, and process input. As mentioned previously, the GLFW library is used extensively for processing input from the mouse and the keyboard. These commands can easily be transferred to another program to be reused in the same manner. The ’setLighting’ function is used to set camera position, set parameters for directional light, spotlight, 4-point light, and material (diffuse, specular, shininess parameters). These could be reused and easily modified to fit the needs of another program or another 3D scene. The functions to create the meshes for the cube, prism, and cone are modular and can be used to draw complex shapes that are comprised of those simple shapes. The modular functions used to draw the actual shapes in the scene – the door stopper, bottle, book, and floor – incorporate the meshes so that, within the ‘main’ function, the functions can be called without any arguments needed. Overall, the functions in my program make the code more organized, modular, and reusable. These three aspects would allow another programmer to look at the code and easily decipher what each function does and how they fit together. It also makes it easier for someone else to take pieces of the code and incorporate them into a new program.