Matt Jackson

7-1 Final Project

CS-330

Enkema

SNHU

> **Justify development choices for your 3D scene**. Think about why you chose your selected objects. Also consider how you were able to program for the required functionality.

I chose my objects based on how their simple shapes could build into complex objects. As it was my first graphics project, I didn't want to get too ambitious, but upon breaking down the chosen scene, it became obvious how the simple shapes went together to create more complex objects. A box and a prism became my coffee bag, several cylinders together with an inverted pyramid for a spout became my coffee press. It wasn't long before I had met the required minimum functionality of four basic shapes. In total I had six basic shapes: planes, boxes, cylinders, a prism, a pyramid, and a torus.

I selected textures for my objects to make the scene more realistic. Wallpaper for the side planes, a tiled floor, wood paneling for the table, a logo for the coffee mug and print for the handle, brown paper with a logo for the coffee bag and stainless steel for the coffee press. I also added materials to make the lighting bounce or absorb based on the reflectiveness. The coffee press reflects light, while the coffee bag does not, for example.

I used a white specular light to get some glints off the glass mug and stainless-steel coffee press and added a bit of blue to an ambient light with some warmer diffuse colors to keep the lighting balanced.

Explain how a user can navigate your 3D scene. Explain how you set up to control the virtual camera for your 3D scene using different input devices.

In addition to the given aswd keys which move the camera left, out, in and right, respectively, the q and e keys move the camera up and down. Moving the mouse around changes the angle of view, and scrolling the mouse wheel increases / decreases the speed of movement around the scene. The calculations for these movements are defined in the camera.h header and implemented in the ViewManager.cpp file. The main function repeatedly renders the scene and polls for events like mouse movement/clicks or keyboard entries.

Explain the custom functions in your program that you are using to make your code more modular and organized. Ask yourself, what does the function you developed do and how is it reusable?

In the ViewManager class, Mouse_Position_Callback function, mouse events are measured and the camera is moved based on the x/y position of the mouse movement. Similarly, in the Mouse_Wheel_Callback function, the scroll distance of the wheel is used to increase/decrease the speed at which the camera moves. Process_Keyboard_events is where the awsdqe key actions are defined. It also allows the user to switch between orthographic projection and perspective with the O and P keys by setting a Boolean variable.

In the SceneManager class, DefineObjectMaterials allows object materials to take on their own ambient, diffuse and specular colors and strength along with shininess which defines their interaction with light. Each material is pushed into a queue that can be accessed before drawing the meshes. SetupSceneLights creates light sources by applying different positions, ambient colors, diffuse colors, specular colors, focal strength and specular intensity. LoadSceneTextures allows different graphics to be loaded for application to various shape meshes.