Willem Kroeger

CS 330: Computer Graphics and Visualization

Southern New Hampshire University

A mirror on a wall

Description automatically generated with medium confidenceJune 13, 2023

A picture containing candle, screenshot

Description automatically generated

**Development of the Scene**

For my scene I chose the mantel piece in the above picture to recreate. I chose to recreate the potted plants, the green labeled candle and it’s lid, and a cubed representation of the cork coasters. Each of these objects are located at different points in the above photo, and the scale of the recreated scene can be better visualized with these objects present.

Creating the pots required a base cylinder, and two torus shapes superimposed over the cylinder. I removed the code to draw the top of the cylinder by commenting it out to give the pot a hole in the top. The torus objects required edits to the source code to increase the size of the “tube radius” to make the tube bigger, while still maintaining a radius that would overlap with the cylinder. This gave the pots the “bulge” look that was present in the picture.

For the candles, I used two cylinders over-lapped to create the jar and added a cone to the top of the jar to represent the candle wick. The candle wicks are also a point light source, as seen by the shadows cast on the pots. The candle lid is shaped accurately to fit on top of the lid. The second candle uses a blue light to showcase the different light sources easier.

The cork coasters are represented by cubes in the bottom right of the scene. The coasters are rotated slightly from each other to give the stacked appearance and show the 4 separate objects.

The mantel itself is represented by a flat plane with a dark wood texture applied. It is slightly lighter than the black matte finish in the photo to be more visible in the 3d representation. The scene is lit with an ambient light and a point light from the candle wick.

**Navigating the Scene**

By usings the WASD keys, the user can move forward, back, left, and right. The Q & E keys allow the user to move the camera vertically up and down, respectively. The mouse scroll wheel controls the camera navigation speed, and the mouse movement controls pointing the camera.

While developing the camera controls, user-accessibility was a large objective. By navigating the camera into a dark corner or loosing track of the scene, the user could struggle to reorient themselves within the scene. To address this, two functions were added to the camera controls. By pressing the F key, the camera reorients itself to the 0,0,0 position in the center of the world space. The scene itself is positioned slightly below zero, so that the scene is immediately visible. Additionally, orthographic and projection view modes are selectable. By pressing the O key, the user switches to Orthographic View mode and the camera is positioned level with the scene. By pressing the P key, the user switches to Projection View mode and sees the 3d representation again.

**Custom Functions**

All of the camera controls are modular and reusable by importing the learnOpengl camera.h header file.

The UProcessInput, UMousePositionCallback, UMouseScrollCallback, and UMouseButton Callback functions are all reusable by importing the glfw3.h header file.

All the shapes are reusable by importing the meshes.cpp file that was provided and calling each shape as needed.

The main function that contains the texture loading code is reusable by repeatedly calling the UCreateTexture & UDestroyTexture functions and applies the corresponding textures to the correct objects.