Diego Bez Zambiazzi

April 17, 2021

Design Decisions

CS-330

The first choice I made when creating my scene was using a cuboid and a cylinder to represent the wooden chest. I later created custom vertices to depict the legs of the chest. I used cuboids to recreate the marble box, the perfume, and the altar in my scene. I also used cylinders to recreate the perfume's top, the candle, and the wick. Additionally, I defined custom vertices to create a glass shape. Lastly, I used planes to represent the floor, the window, and the door in my scene. Each object and detail has textures, such as wood, marble, and wax, as well as normal values to reflect light, which makes them more realistic. There are two natural sources of light in my scene. One is the window, on the left of the altar, and the other is the door, on the right of the altar. I wanted the scene to be illuminated mainly by these sources, therefore I chose large diffuse values for their lighting qualities.

Users can navigate the 3D scene using the keyboard and the mouse. Users can move the mouse to control the direction of the camera and scroll to increase or decrease its speed. The user can walk through the scene using the keys W, S, A, D, E, and Q. The key W moves forward, S moves backward, A moves to the left, D moves to the right, E moves upward, and Q moves downward. The user can also press P to transform the view from 3D to a 2D orthographic world. Aside from the lights emanating from the window and from the door, I included a spotlight that is emanated from the camera and follows the mouse’s movements.

One of the most important custom functions in this program was *loadTexture*. It contains all the logic necessary to retrieve data from image files and load them into OpenGL as textures. Various factors, such as taking paths as argument and handling different formats, make this function reusable. I used this function numerous times in my project to load all the different textures needed. This made my code much cleaner and easier to read. I only had an issue with it once, but that was because the dimensions of the image I was trying to use were not multiples of 4. After I fixed its dimensions, the image was loaded as expected.