Kerrian Offermann

CS-330: Comp Graphic and Visualization

Final Project: Design Decisions

October 13, 2022

When I chose the objects for my 3D scene, my goal was to pick objects with different shapes and materials to avoid redundancy. I wanted something transparent like a water bottle; something soft like my daughter’s pink neck pillow; something plastic like her red ball; and something shiny like the leather three-sided D3 dice. Programming these different objects honestly ended up becoming more difficult than I expected. In my mind, I wanted to do a perfect recreation of the scene down to the last detail. I even wanted to find a way to program the couch and blanket in the background! As time went along, though, I realized that working with OpenGL was not like drawing in a software like Photoshop or GIMP, and it was not like using HTML and CSS which offers a lot of design flexibility. In other words, it was not as simple as creating the object by hand or easily importing shapes with a few lines of code. There was a lot more math than I expected. In order to create the objects in my scene, I had to essentially compile an entire folder worth of bookmarked tutorials as well as dig into some of my old C++ and math notebooks to brush up on old terminologies that slipped my mind. The guidance received from the various announcements in this course also cleared up a lot of confusion I had in regards to new terms like VAOs, VBOs, EBOs, etc. and what they all meant. All in all, the entire project was a learning process. Until the very end, I was learning new techniques and new things that I hope will assist me in the future.

For the user to navigate my scene, they have several options: W to move the camera forward, S to move the camera backwards, A to move the camera left, D to move the camera right, Q to move the camera up, and E to move the camera down. The scene can also be navigated by camera, pressing the alt key and panning with the left button or orbiting with the right button. Another option for looking at the scene is the usage of pressing P to offer a 2D orthographic view of the scene. Releasing the P will return the scene back to its original 3D perspective. These camera options were set up by establishing what happens when specific buttons were pressed and/or released. Once these buttons are pressed then the camera is programmed to move and rotate as needed.

When it came to custom functions, one of the things that I did was create drawings with functions, vertices, indices, etc. that could be reused. In fact, there were certain functions that I ultimately ended up using more than once for things such as the lighting (a torus for my disc-shaped ceiling light and a sphere for the secondary light source like a lamp). Each object is sectioned off from other objects in every part of the main function, so it is easy to add, remove, or replace different objects in the future when necessary. I also sorted each function in what I saw as sequential order so when they need to be used in the future, adding new objects starts at the drawing functions and then continues through changing vertices, indices, setting up VAOs and VBOs, setting up textures, possible shader changes, and then drawing the object in the area designated for creating different loops and model matrix transformations. At the very end, there is a section to release vertex arrays and buffers. There are also codes at the start and end of each drawing function that adds and removes shader programs in the event that the new object needs to use something the different. The same goes for releasing textures so it does not carry on to the next drawing.