Emily Domoracki

Final Project Reflection

CS 330 | December 2023

Professor Brian Holbert

My 3D scene was of a coffee mug, a small handful of blueberries, a quart of cream, and a french-press with some coffee still in it, all on top of a wood table. The scene is lit with natural light coming from behind the starting position of the camera. I started the project with a plane for the table and the blueberries. At first, I thought the blueberries would be spheres, but they look more realistic when rendered as tori with a very small inner radius. I duplicated them almost a dozen times over with slightly different sizes, placement and angles until they matched the original photo. The cream quart was the first complex object I made. I chose to use a pyramid atop a cube for the carton and a cylinder for the cap. This object is the one that I am least satisfied with, as the top of the carton in the original photo is slightly more complex than just a pyramid, but I left it as such. The next complex object I programmed was the coffee mug. I used a cylinder for the base, a torus for the handle, stacked two more for the lip of the mug, and finally rendered the coffee as a two-sector sphere. When building the french-press, The main container was constructed of 4 different cylinders. I used a pyramid for the spout, two spheres at the top of the press, a much smaller cylinder for the internal rod, and 3 cylinders for the handle.

Most of my trouble rendering these objects came from applying texture and lighting. I could not figure out how to augment the sphere and torus functions to add normals to their data matrices, so these objects were rendered with a shader file that did not take normals as an input. I was able to render the empty glass portion of the french-press with the shader files which implemented phong lighting that I used for the rest of the program – but I had to adjust the shader itself to make it transparent, so that became a new shader class object. Lastly, the light source got its own shader, so in total, I ended up using 4 different shader files.

The user can navigate the scene with the use of the keyboard and mouse. Keyboard input allows movement along the x, y, or z axis relative to the camera position. Movements include forward (press W), backward (press S), left (press A), right (press D), up (press E), and down (press Q). The mouse scroll can also move the camera forward and backward. Mouse movement itself controls the pitch and yaw, which correlates to the angle at which the camera looks up, down, left, and right. Roll is the third Euler angle, but is not included in this setup, as we don’t need to traverse the scene upside down.

There is a lot of opportunity for modularization in this program beyond what’s been done. It is certainly helpful that different shapes have their own classes which also generate and bind VAOs and VBOs each time it’s rendered through the function. The shader class is also helpful when creating and implementing multiple shaders. I would have liked to modularize the function that loads and generates textures, but all my attempts were unsuccessful which led me to sacrifice efficiency in code for the sake of completing the project. The other piece I wish I had time to figure out how to modularize are the blueberries. Rendering the blueberries created a lot of redundant code that can certainly be cleaned up with an iterating function. While my code is formatted, commented, and employs functional logic – I am not satisfied with the redundancies and lack of modularization. However, I did the best I could with the time I had.