Ken Rodriguez

Professor Diesch

Comp Graphic and Visualization

April 20, 2022

Project Submission

For my 3D scene, I decided to choose objects that would scale with the project’s development as I continued to learn and grow throughout the course. While I did face many challenges during development, I feel that my object choices were sufficient to match this project’s requirements and facilitate my own growth. I feel that the first object was a good starting point to build a familiarity with OpenGL: a copy of the book *Flour Water Salt Yeast*. The object created to reflect this book was simple as a result; two cubes representing a cover sandwiching a single white cube textured to act as pages and bound with a plane representing the spine. This starting point allowed for simple vertex positioning, texturing, and lighting; after creating the mesh of the shape, I was positioned to experiment with different texturing and positioning methods before settling on a method that would allow for easy encapsulation and modifying of object meshes.

The approach I took to generating meshes involved separating object pieces into different methods based on the texture each object would need. Given that each piece (or pieces) utilized different textures, this made identifying specific sets of vertexes and triangles easier to identify in the render, subsequently allowing for simple mesh modification. This approach also served to reduce visual confusion; rather than sifting through hundreds of lines of vertex coordinates to find a specific vertex, each object was broken up into fifty or fewer lines, reducing confusion and increasing readability.

The virtual camera was added early in development, using both the client’s stipulated controls and a popular first-person game control scheme. In both contexts, the user uses the keyboard to traverse a flat plane where the “W” key allows users to move forward relative to the camera’s position, the “S” key allows users to move backwards relative to the camera, the “A” key allows the user to move left relative to the camera, the “D” key allowing the user to position the camera right, and the “Q”, “E”, spacebar and left control keys allowing the user to move the camera up or down relative to its position. A computer mouse can be used to allow the user to pan the camera in its stationary position; by combining these features, users can fully explore the 3D scene and observe details, a functionality that allowed for faster development and simpler mesh modification. Camera functionality was stored in its own header file, further contributing the code’s general readability.

In addition to writing methods to generate specific meshes and modify camera movement, methods were implemented to allow for easy texture, shader, and mesh generation and destruction. Creating a single method to generate textures provides a single place where images can be assigned to slots in the OpenGL texture map where they can be called for objects in the scene. The shader generation method allows for multiple different shaders to be created at the project’s runtime with a single reusable function, then destroyed with its related destruction method. Similarly, the generated can be passed through a function that handles projecting them to the shader, successfully rendering the scene, then destroyed to allow the program to release memory.

Altogether, these mesh generation methods, the camera movement method, and the creation and destruction methods create a program that is organized and relatively easy to modify to best meet the project’s requirements.