Tammy Hartline

08/11/2023 – 23EW6

Final Project Review and Justification

CS-330 Computer Graphics and Visualization

**Justification for My Development Choices:**

* ***My Selected Objects and Overall Scene:***

For my scene I originally chose to draw my own, of little Kawaii ice-cream characters. I did this thinking it would be easy to create them… which I was incredibly wrong about, lol. By the time it was time to begin development of my project, I narrowed down my scene to a single ice-cream in a cone, sitting atop a table in the moonlight. The scene consists of a plane for the tabletop, a sphere for the scoop of ice-cream, a cone/cylinder for the ice-cream cone and another sphere for the moon. The objects I chose provide a visually engaging and recognizable scene that does look like an ice-cream cone sitting atop a table at night in the moonlight. Combining the sphere and cone added to the overall complexity of the scene, while still adhering to the low-polygon requirement.

* ***My Programming Choices:***

I completed the entire program in Visual Studio 2019 IDE Enterprise edition, as I use it for work also. The program uses C++, Modern OpenGL, and GLSL for rendering, which are very common choices for creating real-time 3D graphics. My choice to use shaders allowed me to create more advanced lighting and material effects. The use of three textures and color overlays, along with shadow, light and reflection creation enhances the realism of my scene. While I struggled with the complexity of my chosen scene, I am very proud of the final result achieved from nothing more than code and mathematics. I also added camera navigation through use of key commands (WASD and QE), which offers users a familiar and intuitive way to control the scene and change the view. I also enabled mouse control which helps with the view of my scene.

* ***User Navigation:***

My program allows users to navigate my 3D scene with their mouse movements, as well as the WASD key controls of W->forward, A->left, S->backward, and D->right, while the Q and E controls upward and downward movement, and the P->changes perspective. My code is implemented throughout the program, allowing the camera’s position to be adjusted based on the key selection. This allows my users to traverse the scene along the X, Y, and Z axes. It was the simplest and most basic way to set up an explorable environment, (3D world/scene) for a user.

* ***Custom Functions and Modularity:***

While I believe I could’ve done much better by grouping each section of my code into functions, I did have several included. Some of these include my load shaders functions, as I used more than one shader program, each of which called and read in a separate file, where I had stored my shader code as glsl file. Another function I included was one to calculate and create each of my objects. Due to having different objects, and different ways of calculating the vertices, which was dependent on the shape type, I created one for each object shape, such as createCone(), createPlane(), and createSphere(). This did allow me to create more than one sphere, by calling the createSphere() function and passing in my variables, which saved me quite a bit of time when I decided my light source would be a sun or moon. I also created a processInput() function, that was used for the WASD, QE, and P key command functionality, which controls the camera movement based off of the users entry. Given more time and had it not taken me so many attempts for each section, I would like to have restructured my program and added functions that would have included cleanup and object rendering processes for each object, including the ability to set uniforms for model, view, and projection matrices. Had I included this in my original program structure, it would have allowed me to better sort through the program, and likely would have also cut my troubleshooting and debug time in half. While the functions I created do make my code easier to manage and maintain, as it allowed me to reuse most of the sections by making a call to the function and passing in new variables, more of them would’ve made my program that much more comprehensive without having to include over 1,000 lines of code.

A red and white sphere on a black surface

Description automatically generated­