7-1 Final Project Reflection

Jacob Clara

Computer Science Department, Southern New Hampshire University

CS 330: Computational Graphics and Visualization

Philip Enkema, MSwE

December 15, 2024

The two-dimensional scene I chose to replicate in three dimensions for this project is a photo from a family visit to Hua Hin Beach in Thailand. The photo carries deep sentimental value because my wife and I discovered we were having our baby boy, now a year old, while on the trip, just before we went out to watch the sunset in the scene. Beyond the sentimental value, the scene has a variety of challenging objects to render, with the lantern being particularly captivating. The sky, water, and sand elements were all foundational to the beach view. I rendered them using planes with added texture and material attributes. The stone wall separating the sand and beach added depth to the scene and a contrasting texture. I utilized the basic mesh functionality to draw all my shapes, combining different meshes for my complex objects.

One complex object was the beach umbrella. The beach umbrella added verticality to the scene while all other objects had horizontal layouts. At first, I designed the beach umbrella stand using a flattened box with flattened cylinders to represent the hole for the umbrella pole to go through. As the project progressed, however, I sought a way to render the beach umbrella stand with an actual hole. My final design was based on a series of outwardly expanding horizontal tori, leaving a hole in the center that the user could see through, and the umbrella pole could go through. The final design was more appealing and realistic.

The lounge chairs were important to the scene because there were several of them in it. They also had unique aspects like the armrest areas that form triangle-like shapes. I proposed to render the lounge chair object using thin prisms for those armrest areas but opted for elongated boxes during the design process. My final design included those boxes rotated to meet at their edges and form a look as unique as in the original image.

My last complex object was the lantern, and my final design was quite different than the one I began with. As opposed to a box with a pyramid on top and legs made of cylinders, I designed a

lantern with grid-like sides. I achieved the grids by iterating through loops to draw crisscrossed bars, adjusting their position based on a spacing variable each time, and applying the necessary transformations before rendering. The lantern was the most enjoyable object to work on and render.

I added controls to the virtual camera in my scene for keyboard and mouse scroll wheel input devices. Using the GLFW keyboard callback method, I enabled the capability to move the camera up and down using specific keys. I also added the option for the user to switch between orthographic and perspective views of the scene with the press of a key. The keyboard input function retrieves the state of a specific key, determining whether it is pressed, released, or held down (GLFW, 2024a). I also enhanced camera control by allowing the user to control the speed of the camera movement via a mouse scroll callback. My mouse scroll callback method adjusts the speed of the camera at a small increment based on whether the user moves the mouse scroll wheel up or down (GLFW, 2024b). This camera control was vital to me as the designer, allowing me to slow down the camera speed to view specific, hard-to-reach sections of the scene in detail.

I added a separate *MyObjects* class to the project, to encapsulate my code for drawing custom shapes, as the *RenderScene()* method in the *SceneManager* class appeared cluttered after drawing just one complex object. The *MyObjects* class contains separate methods for drawing each of my shapes, with the lantern method calling additional internal methods to render specific lantern aspects. The lantern and beach chair methods specifically enhance modularization and reusability by rendering the objects with relative values, allowing you to draw these shapes in any program as many times as needed based on the XYZ position. This approach also allowed me to maintain the separation of concerns for the project's classes and the encapsulation of the *SceneManager* class.

References

GLFW. (2024a, February 21). *Keyboard input*. Retrieved from GLFW: https://www.glfw.org/docs/3.3/input_guide.html#input_keyboard

GLFW. (2024b, February 21). *Scroll input*. Retrieved from GLFW: https://www.glfw.org/docs/3.3/input_guide.html#scrolling