7-1 Final Project

Dhiraj Gurung

Southern New Hampshire University

CS330

Omar Aaziz

October 13th, 2025

**7-1 Final Project**

For the 3D scene, I chose the objects based on what was available in and around my desk space. I wanted to have a mix of both simple and complex shapes, so that I could learn to render progressively more complex shapes. This is because this was my first time working with 3D modeling and rendering objects into a scene. As such, I wanted to include shapes that were relatively simple and easy to implement. The roll of tape was chosen for this reason. I did not need to worry about adjusting multiple shapes to make the tape. I just had to render the torus shape and worry about positioning the torus in the right spot. The next items I chose were the perfume bottle and the Nintendo Switch Dock. I chose these shapes because they were a little more complex than the tape. The perfume bottle only required two shapes: a cylinder and a box. I had to create a rectangular-shaped bottle, while the cap was a cylinder stacked on top of the rectangular shape. The Switch Dock was a combination of 3 boxes that were placed side by side, with the middle box being shorter than the other two. Working on these shapes was trickier than just a single torus because I had to figure out not only where to put the objects, but also how to shift each individual shape in relation to the other. Shifting the whole object became easier once I had the shapes aligned in the proper places first. The last object I chose was the pill bottle, which utilizes a cylinder for the base and cap, and a tapered cylinder for the neck of the bottle. This was by far the most complex shape I had to work with. This is because of the neck of the bottle. It took me some time to figure out how I was going to create the neck shape, and I ultimately decided to use the tapered cylinder to show the size difference between the base of the bottle and the neck of the bottle.

The user can navigate the scene using the WASD keys to move forward, backward, left, and right, respectively. They can also use the Q key to move upward and the E key to move downward. The user can use the O button to change to orthographic view and the P button to change to perspective view. By default, the view is set to perspective view. The mouse can be moved to look up, down, right, and left. The scroll wheel can be used to adjust the camera movement speed. Scrolling up makes the camera move slower, and scrolling down makes the camera move faster. The scrolling setting may vary depending on the user’s personal scroll wheel settings. For all the keyboard button inputs, I used the existing code in the ProcessKeyboardEvents function as a template. I wrote an if statement for each button input, such that an input would trigger a specific camera movement. I mapped the different camera inputs to the following keys: W, A, S, D, Q, E, O, and P.  The code for the orthographic and perspective views was similar in that I used an if statement to check if the buttons were pressed. Then, within the if statement, if the button pressed was O, I would set the bOrthographicProjection variable to true, and false if the button pressed was P. This allowed me to map the different viewing modes to the O and P buttons. Additionally, I set the value to false to begin with, so that the scene starts in perspective view.

There were a lot of functions already given to us, which allowed us to create the majority of the project. However, I still had to include a few functions on my own. For example, for the texture wrapping for the perfume bottle, I had to create a function to help me overlay the texture with the words on the front side of the bottle. This function was named DrawBoxSideMesh, and it is located in the ShapeMeshes file. This function allowed me to draw a specific side of a box, instead of the whole box. This was important because it allowed me to overlay 2 different textures on the perfume bottle. This same function was also used to add two different textures to the Nintendo Switch Dock. This function can be used for any box in the scene and allows me to better control the textures of each side of a given box. Using different custom functions allowed me to keep my code modular because instead of creating one big function with a lot of functionality, I had a function focused on a specific functionality. This has several different benefits. For example, debugging the code is now easier because if I have an issue with a specific functionality, I can isolate the problem to a specific function rather than specific lines of one function, which is harder to do. Additionally, the function can be invoked anywhere within the file multiple times, which allows the code to be reusable. This saves me time from having to rewrite the code again each time I need the specific functionality.