

Southern New Hampshire University

7-1 Project: Reflection

Preston Burkhardt

Malcolm Wabara, M.S

CS-330-T5621 Computer Graphics and Visualization

June 19, 2022

When trying to decide what objects to choose for my scene, I wanted to pick a few that would be simple, but also choose one of two that would pose a slight challenge. I didn't want to choose the easiest objects I could find. This is what lead me to choosing a speaker (cube), a tennis ball (sphere), a roll of tape (torus), and a camera (combination of cube and cylinder). In hindsight, I did not know just how difficult a sphere, cylinder, and torus would be to render programmatically. As for camera movement, I felt that using the WASD keys for movement and mouse movement for panning the camera, as the assignment required, is the most logical way to control the camera. I must admit that I may be biased due to playing my fair share of computer games where this is how you control your character within the game. One requirement for the project was to add at least one colored light. I felt that this would make my recreation a bit inaccurate, so I opted instead to create lighting toggles. The "1" key toggles the default lights, or the lights that I felt were most accurate to my original image. The number keys 2 through 4 switch the lights to red, green, or blue respectively.

For movement within the scene, like I stated above, a user can move the camera around using the WASD keys. W goes forward, A to the left, S goes backward, and D goes to the right. The user can also move the camera up and down on the Y-axis by using Q (up) and E (down). For panning the camera, the user simply moves their mouse. Finally, the user can use they scroll wheel on the mouse to adjust movement speed of the camera. I feel that this is the most logical way of controlling a camera with mouse and keyboard. Imagine if movement and panning were swapped. Panning the camera with the WASD keys while moving with the mouse doesn't feel "natural".

In order to make my code more modular and organized, I used classes for my shapes and lights. I noticed early on that all shapes have lines of code relating to "building" said shapes and

lines of code related to “rendering” the shapes. Furthermore, all shapes would need a mesh structure. Because of this, I created a Mesh parent class and the shapes were all child classes of that Mesh class. The Mesh class contained the code related to building the shapes while the shape classes had their respective vertices vector. In hindsight, if I had more time, I would have also rolled the code related to rendering the shapes into its own render() function within the Mesh class.

Overall, this class/project was a very good (and hard) test of my ability to write code, understand code presented to me, and implement ideas/practices into my code in order to achieve what I wanted. It also helped me practice implementing object-oriented programming as you can see with my shape and light classes.