**7-1 Project One**

Celine K. Philachanh

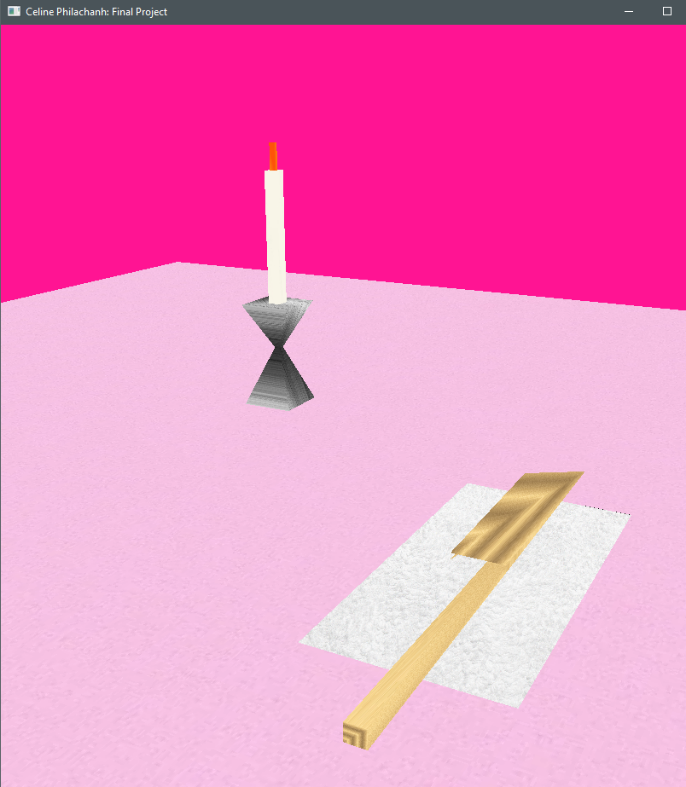
Department of Computer Science, Southern New Hampshire University

CS-330: Computer Graphic and Visualization

Professor Kurt Diesch

December 13, 2022

**7-1 Project One**

****

**Reflection: Development Choices**

The selected reference image I chose, not only included interesting objects to reconstruct in a 3D environment, but also provided a great learning experience. Originally, I intended to recreate five to six objects, but this changed due to my eagerness to achieve more than what I am capable of. At the end, I decided to recreate the candle, candlestick holder, the napkin, and the butterknife. I finalized my selection with these objects because they provided me with a challenge yet remained accurate and held the integrity of the original image without the objects not selected. I found it interesting how the objects in the original photo, in a way, don’t make sense. Such as, there being a butter knife paired with a coupe glass and cereal in a bowl. Recreating the butterknife with the candle and candlestick, and the napkin together brought humor to me. To add on, the chosen background color in the 3D scene remained pink. This was intentional so that it could take on as much resemblance to the reference photo. The same thought process applies to the table (plane). The texture for the table was found online and although it isn’t the exact color as the original table, it still holds value in the comparison.

I was able to program for the required functionality by using resources such as the website, learnopengl.com, and YouTube videos. Additionally, I emailed my peers for tips and advice when I ran into an issue. These resources helped me improve on my coding functionality because I implemented various functions and methods that I thought I could not improve on or provide in the final project. With the countless hours spent on creating the 3D scene, I consistently referred to my resources and this helped improve my understanding on computer graphic and visualization. Though, I finally reached the finish line, there were a few things that did not work out along the process. One of the issues that arose was deciding which primitive shapes to use to recreate certain objects. Since I originally planned to recreate the spoon, I spent most of my days researching how to create a half-indented sphere. I knew that I had to pick a different object since I was not getting anywhere. Another roadblock was the lighting. In my code, you can see the sections that contain the code for the phong shading model. After making changes and adjustments more times than I can count, the rendering would not display the lighting. I ended feeling discouraged and disappointed in myself but knew I attempted everything I could. Unfortunately, I truly did want to make it work but also knew that if I spent too much time focused on areas where I was not getting anywhere, I had to move on.

**Reflection: Navigation**

A user can navigate the 3D scene by pressing the keys, Esc, WASD, QE, PO, mouse scroll, and mouse cursor. The user can simply utilize the mouse cursor to adjust the orientation of the virtual camera. This in turn allows the user to look around (up, down, right, left). In contrast, using the mouse scroll lets the user adjust the speed of the movement. Such that, if the user wanted to decrease the movement, they scroll down, otherwise, they can scroll up to increase the speed of movement. As for the WASD keys, these keys allowed users to control moving around the scene whether horizontally, vertically, or depth wise. The keys were paired with the process inputs using the camera header and source.cpp files. For the user to exit the window, they would need to click the “Esc” key on their keyboard. The Q and E keys allows the user to navigate up or down the 3D scene. This makes the program more flexible in viewing the scene from different aspects. The keys, P and O were used in the code to let the user switch between a 2D (orthographic) and 3D (perspective) view of the 3D scene. By pressing on the O key, the user is in the orthographic view and if they decide to view the 3D scene in 3D, simply pressing the P key changes to a 3D perspective. Using a Boolean in the code helped with writing the input for this device. These input devices were vital in achieving a full view of the 3D scene.

**Reflection: Custom Functions**

To ensure that the code was modular and organized, various functions had to be implemented. Implementing a function for generating textures made it simple to insert textures all at once. This can be found in the “void generateTextures()” function of the code. Rather than going by the tutorial from the module 5 tutorial, this function made it easy to have the parameters and mipmap all together. This can be reusable because it is not only easy to understand but also minimally requires a texture file and texture definition. To add on, implementing the different object functions were challenging but also modular and organized. Rather than having each object function separated, they can be easily identifiable in the code because of the grouping which follows the “URender” function. This is reusable because it only requires vertex, color, and texture coordinates. To make it better, an individual can take out any vertex shader source or fragment source shaders if they do not want it included in the final rendering of their program. Whether it be the normal or texture, the individual has flexibility deciding what to include. Adding an additional light to the key light was added in the code. The code for this function could be reused to create additional lights in a 3D scene. By declaring the position, scaling, and determining a color, an individual can have unlimited possibilities with their lighting mechanics. The additional lightings in my scene took on the variable by an increase of one. For example, implementing more ambient lights would be named ambient and then ambient2, so on, so forth. There are no extra requirements for the lighting code because the code already has the requirements for lighting a 3D scene. Finally, it was important to me that the code was fully modular and organized by using best practices such as formatting and commenting. In the code, there are comments aligned with most of the lines. Additionally, logic was applied by reusing code from different sections to improve the functionality and reusability