CS330 7-1 Final Project

Michael Ross

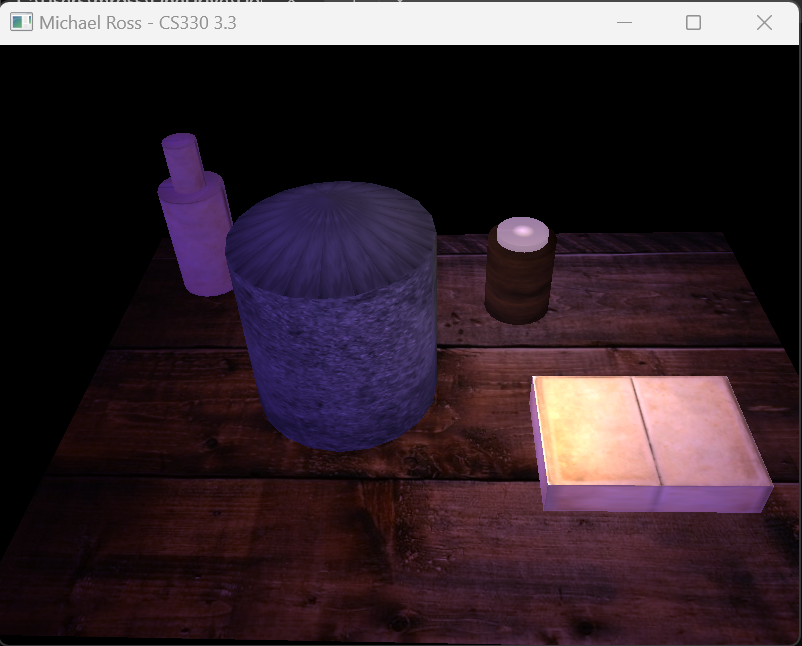
Department of Computer Science, Southern New Hampshire University

CS330: Computer Graphics and Visualization

Prof Diesch

February 19th, 2023





For reference, the original image and the simplified version.

# Justification

As part of creating the final 3D scene, I chose to create the helmet, the table, the book, the candle, and the bottle. In this section, I will detail the decisions on why these objects were chosen and some of the programming and OpenGL constructs used to make them work.

The helmet was composed of a cylinder with a very shallow cone on top. This cone is used to represent the top of the helmet. I chose these two shapes because I thought that they would represent the helmet best when combined. I stacked the cone on top of the cylinder so there were no seams and then used an iron texture to apply the metallic sheen to the completed object. The programming for the cylinder involved specifying how many vertical slices to utilize to represent the cylinder. This allowed the creation class to loop through the required number of slices and create the Vertex Array.

The candle was composed of two stacked cylinders. Unlike the helmet, this required two different textures to complete. Because two textures were required, the shader needed to be modified between drawing the bottom and the top so that the textures drawn could be different. The bottom utilizes a wood grain texture, and the top utilizes a white wax texture. In creating this object, I had to be extra careful because part of this object would represent one of the sources of light for the scene. I ended up creating a custom function to allow the light source on top of the candle to have a flickering effect by alternating the linear and quadratic values for the point light source by a random amount.

For the book, I only needed a longer and thinner cube shape to represent the book with a texture that allowed the middle to look like it was indented. This simplified the generation of the object. This was possible because I added the ability to set all three dimensions of my cube primitive when creating it through the custom Shapes class.

The bottle involved creating two stacked cylinders both with the green glass texture applied. It could be refined by enabling transparency in the material, but that is not something we covered as part of this experiment. The result looks like a bottle of green glass.

Finally, the table is represented by a plane with a texture that looks like rough wooden planks. The result is a nice-looking object that appears to the viewer like a rough wooden table.

The lighting is constructed of two different light sources. The main directional light source is a blue glow coming from above and to the left of the table. This casts an eerie blue light on the scene. I had to try several different combinations of ambient, diffuse, and specular light values to get this blue glow exactly right. The second source is the candlelight coming from just above the candle object. This also took several tries to get perfect as it needed to be a point light, but the linear and quadratic values needed to be realistic while at the same time appropriately lighting the scene. In the end I opted to add an additional effect that made the point light act like candlelight with random fluctuations of the amount of light being put out. It gives the scene an additional level of sophistication.

# Navigation

To navigate the scene, the user has several options at their disposal. To pan left and right, the ‘A’ and ‘D’ keys can be utilized. To pan up and down, the ‘Q’ and ‘E’ keys are used. To zoom closer to the object, the user can use the ‘W’ key and to get farther away, the user can use the ‘S’ key. The amount of movement is fixed initially to a pre-determined value, but the user can change this by leveraging the scroll wheel. The scroll wheel makes the movement controls move faster or slower. The mouse can also be utilized to change the look direction. Finally, the user can press the ‘P’ key to switch between perspective and orthographic viewpoints.

All these movement commands serve to move the virtual camera in various directions to allow the user to view the scene from different angles. This increases immersion and allows the user to see the light source effects and textures from an unfamiliar perspective.

# Custom Functions

Various custom functions and classes were utilized to make creating a functional scene easier. The first class that I created was the shader class. This allowed a shader to be loaded and utilized and to set Uniform variables inside to change the way it behaves. This simplified the code in allowing common functions to be abstracted in the class. It also simplifies the main loop in the code by keeping the instructions simple such as “shader.use()”

The other class I used kept long and complex array calculations for the various primitives out of the main function. This was the Shapes class. The Shapes class includes various sub classes for the primitives used in this project. It allows creation of the vertex arrays, colors, normals, and texture coordinates for the primitives. Creation of the shape can be done by instantiating an instance of the needed class and specifying the right parameters to the constructor. Drawing the created primitive involves simply a “.draw()” method to be called.

Other functions created were used by the mouse and keyboard input to modularize that functionality. The use of several callback functions enabled these functions to be called when certain input conditions existed. This allowed the view portion of the matrix to be modified to show the accurate position in space that the keyboard and mouse were pointing at.

# Conclusion

The creation of a realistic looking scene including lights, proper positioning, model creation, texture creation and the backend work of creating the classes and functions that go into the final product was a lot of work. Seeing the final product that was well lit and represented the initial image accurately was a great feeling. Even better, classes and functions were created as part of this effort that can be utilized in future projects to make them proceed even faster.