**Project Reflection**

When looking at the objects that were in the image I used as a reference for building my scene, it’s easy to see why I used the shapes and colors that I did. The red ball is easily represented by a red sphere. Same with the battery being represented by two cylinders, one for the main part, and one smaller one for the node at the top. For the deck of cards, I went with a large rectangle that shows the shape of everything, even if it doesn’t show the actual cards or labeling, etc. Everything was placed on a white plane that represents the paper that was used when placing the objects and taking the picture.

The most difficult object in this scene to build was the pin. It had multiple parts, each needed to be represented. I used many cylinders of varying sizes and different colors to try and represent the pin as best as possible. When looking at the two, it’s possible to see how the object was built and how similar they look. The base piece also had to be smaller than the top cylinder, with the shaft being a longer, thinner cylinder than both. The pin part of the object at the top needed to be silver and thin, as well as being quite long.

For creating all these shapes, I mainly used classes that could generate the vertices and indices of the objects that I needed. With the plane, the dimensions were passed into the class and the square plane was then generated. This is also how the cylinder and sphere were generated. The textures were then applied to them, and the object was drawn.

The rectangle was one of the more difficult things to implement, as it doesn’t have a quick implementation using a class like the other shapes. The vertices had to be manually input and the object had to be manually set up and drawn.

A user can navigate the screen with a few buttons and the mouse. The buttons here being W for forward, A for going left, S for backwards, and D for going right. There are also Q and E for going down and up, respectively. The mouse is used to navigate the screen, using it to look around and change the viewing angles and positions. It’s also very necessary to note that pressing the P button changes the view of the scene between orthogonal and perspective.

I used certain blocks of code to allow for better flow of the code. The top part, outside of the main function is where everything gets set up. It starts with the vertices and color positions, moving onto the VAO’s, VBO’s, etc. This also goes for setting up the lights and creation of all the plane and cylinder objects. These all get created and can then be loaded when we want to start running the code.

Once all this is done, the objects get loaded onto the screen. It’s easy to see in the code how this is done and broken down. For someone who wants to also use this code, it would be possible to make cylinders, cubes, and lights, wherever they want on this already drawn plane. The variables for each are changed here as well, so the code allows for reusability in the sense of changing all the variables directly where the objects get drawn. This makes everything more organized.