**Justify development choices for your 3D scene. As you write, think about why you chose your selected objects. Also consider how you were able to program for the required functionality.**

For my project, I chose to make a shelf with three objects on it. Those three objects were a tennis ball, a brick, and a roll of toilet paper. I selected those objects because I wanted to meet the requirements for complex and simple shapes that make up a scene that I think I could make in the time provided. The shelf itself is made of simple rectangle faces but there are a lot of rectangles needed to create the full shelf, so it was an interesting prospect for a challenge. The shelf is made to hold toilet paper.

To program for the required functionality, I thought that it would be simple enough to make a 2d object and repeat it for all my modeling needs, however I found out that many shapes require more than one type and size of shapes. I needed to make 3 different rectangles for one shelf and 3 more for a brick. After I made those 3 rectangles per object, then I could repeat them for the sides that applied. That is how I happened upon the top, bottom, left right, front, back where the smallest sides were called the ‘caps’. Each side and its opposite were a pair that required the same rectangle to replicate. Once I made one of the 3d models, if applicable I would use a loop to replicate that model and translate it to the place it is needed. Take for example, the shelves/pillars. That is only one long shelf (in this case, the bottom shelf), looped to make 4 copies, 3 of which are translated upward to make all 4 shelves. After that there is another loop that generates 4 more copies but rotated to fit a vertical version of the shelf and translated on the x axis to move it from leftmost pillar to rightmost pillar.

**Explain how a user can navigate your 3D scene. As you compose your thoughts, discuss how you set up to control the virtual camera for your 3D scene using different input devices.**

For this 3D scene, I used the mouse and keyboard as input devices capable of navigating the 3D scene. The keyboard is in charge of controlling the direction the camera is moving in. W moves forward, A is left, S is back, D is right, Q is up, E is down. The keyboard also controls what view we look at the scene with. P swaps between orthographic and projection view. The mouse controls where the camera is looking as well as the camera’s movement speed. Moving the mouse around controls the view, while moving the scroll wheel up or down increases or decreases camera speed respectively. The combination of the two allows for finite camera movements to navigate the 3D scene comfortably. Setting up these controls proved to be difficult because it took a lot more math than one might expect. There are a lot of matrices involved in transforming the camera’s position as well as the view. I would have expected simple modifications of (x, y, z) values.

**Explain the custom functions in your program that you are using to make your code more modular and organized. Ask yourself, what does the function you developed do and how is it reusable?**

My custom function within the code was “targetFollowsCursor()”. It is a simple function made to modify the target variable in real time in an organized place outside of the main loop. I made the call to the function in the main loop so that it updates constantly as if it were in the main loop even when its not. This makes the code more readable because it sections the camera view following cursor functionality from the main code.