Kevin Schroeder

**Project Design Decisions**

**Justify development choices for your 3D scene**

When selecting my objects for the 3D scene, I wanted to add a variety of different shapes that could challenge my understanding of the content. I thought that the book would be a great starter piece because it’s just a cube. I knew that the book would give me good experience with modifying the height, width, and depth of a cube (or other shapes) to represent something else. I ended up gaining experience with combining shapes when developing the book because I used three planes to represent the pages. I thought that the paperclip would be another beginner object and it gave me some experience with building pyramids. Although my result wasn’t exactly a true pyramid, the concepts that I had learned in our weekly assignments on how to build pyramids were very helpful during this build. I saw the pen (which I later swapped out for a pencil) as a good starter piece for me to practice combining multiple shapes. I used a cylinder for the base, a pyramid for the top of the pencil, and then another pyramid for the black tip. Not only did the pencil give me experience with combining objects, but it also challenged me on how to use different textures to make a more realistic image. I saw the coffee cup as a good way to learn how to create a cylinder and how to combine multiple objects to represent the cup and the handle.

**Explain how a user can navigate your 3D scene**

My 3D scene can be navigated by using a combination of the keys and the mouse. The A key allows users to move to the left in the scene. The D key allows users to move right. The W key zooms into the scene. The S key zooms out of the scene. The E key moves the camera up in the scene. And the Q key moves the camera down. The mouse allows the user to angle the camera to see their surroundings. The mouse doesn’t move the camera’s position but rotates it to see the scene from different angles. The scroll function on a mouse or trackpad speeds up the velocity in which the movements happen on the screen. Scrolling up will increase the velocity, and scrolling down will decrease the velocity.

My project has a function called processInput that is in the main while loop and is constantly checking for user input. When the user types on the keyboard or uses their mouse, the processInput function delegates the input information to a callback function, which utilizes functions in the camera.h file. The camera.h file is responsible for processing the input from the user and moving the camera accordingly. This feedback is sent back to the main Source.cpp file and the camera's position is updated.

**Explain the custom functions in your program that you are using to make your code more modular and organized**

I tried to keep abstract as much logic outside of the main while loop as I could. I had different functions that set up the VBAs and VBOs of most of the objects and I tried to keep these function names generic when I could to promote reuse in the future. Having these object-specific functions was a huge help in making my code more readable. In the while loop, the process of creating an object that uses the OurShader shader is the same: define the texture, bind the VAO, manipulate the object (translate, scale, rotate, etc.), and draw the object. This pattern gets repeated and I put comments on which object is being created to guide a reader. I also had the logic for setting up a cylinder and handling camera movements in different files. There are a lot of functions required for those two functionalities, so it was helpful to abstract that code into a different file to prevent my Source.cpp from getting too long. I used a combination of inline and full-line comments to keep track of the objects that were being drawn and the purpose behind the code.