Katelyn Perry

CS-330

October 2022

3D Scene: Design Decisions

Planning out the development of my 3D scene began with choosing which objects I would include to represent. When picking these objects, I tried to make a conscious decision to make sure the objects varied in shape, size, and material. One object is a book. This is made of two kinds of shapes – one cube to represent the insert of book pages and three planes to represent the binding of the book. The next object is a pad of sticky notes which is made up as a plane for the sticky tabs to sit on and multiple cube shapes to represent the sticky tabs. Next, I recreated a highlighter I own. To do so I created a cylinder shape and rendered this shape multiple times, transforming it each time to make the various sections of the highlighter pen. This pen also had a section to clip itself onto your shirt, which I recreated using transformed cubes. Lastly, I had intended to draw a sphere to represent a Jupiter globe I own. However, I was unable to get a sphere to work. Due to this, I made the decision to switch out this object and recreated a silver paperweight in its place.

There are a few different camera controls for the user to navigate the 3D scene with. The first is the use of the WASD keys, which when pressed enable to user to move the camera forward, backward, left, and right. The user can also change the height of the camera by using the Q and E keys and switch from a perspective and orthographic view of the scene through the P key. This was achieved by creating an object for the camera class, coming from a header file implemented in my project. With this file and the use of GLFW library functions, I was able to create a function for processing certain keyboard events to reflect the desired changes live. Using this same object and library, I was able to use the users’ mouse as an input device for the camera as well. The user can move the mouse around the screen the change the orientation of the camera, so it can face any direction. Another accepted input is the mouse scroll button, which will increase or decrease the speed of the camera. Both functions were achieved by using separate call back functions.

To make sure that my code is more organized and in a modular fashion, I used functions to separate different areas of the code. This made projects main function much simpler, and overall makes testing and editing easier. One example of this is the processInput function. This function handles all input coming in from the keyboard. If you ever needed to update the input options, it could be done quickly within this function without compromising the integrity of the rest of the code. Another function that provides a lot of reusability is the createTexture function. This enables you to call to this function each time you load in an image to make it into a texture, rather than rewriting the code for each image needed. With functions everything is sectioned off, and you can minimize areas you aren’t working on to prevent any confusion or clutter on your screen. To further organize the project, I also used a namespace. This namespace held the structure of the mesh, all the associated vertex objects, and information pertaining to each object.