Alex Casanova

CS330 Final Project – Reflection

* 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.
  + Developing the application for this project posed some interesting challenges. I initially tried developing the scene one mesh at a time, and it was a challenge to get it to work. However, after watching some tutorials and finding a way to compartmentalize the code for reusability, it was much easier for me to add shapes to the project, and I was done with it relatively quickly. Using a Class to build objects, rather than writing out the vertices manually, allows for more reusable code, as well as improved ease of development. The SceneBuilder Class allows developers to specify scale and position data for shapes, which is then passed to the ShapeBuilder class to generate the mesh used in rendering the shape. This completely replaces the manual process I used before, and I don’t think I’d go back to creating shapes manually given this option.
* 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.
  + This application relies on standard WASD controls.
    - W moves the camera forward
    - S moves the camera backward
    - A moves the camera to the left
    - D moves the camera to the right
    - Q moves the camera up
    - E moves the camera down
  + In addition, some other keyboard controls allow for extra functionality.
    - L toggles the movement of the light
    - P changes the perspective from Orthographic to Projection.
  + The app also allows for mouse control of the camera, in order to provide camera rotation. This control scheme is relatively common, and applies to many 3D scenes, such as video games.
  + Since GLFW supports controllers, I implemented controller support as well.
    - Left Analog – Movement
    - Right Analog – Camera
    - LB – Camera Speed Down
    - RB – Camera Speed Up
  + This additional functionality allows for those without keyboard and mouse access to use this application.
  + The camera and movement classes are platform-agnostic, so mapping them to controller inputs, or other input devices with sensor input, is relatively simple.
* 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?
  + I used the provided SceneBuilder and ShapeBuilder classes, which allow for simple addition of shapes by inputting a few parameters. Previous module assignments saw me inputting the vertices and attribute values manually, costing me a lot of time and headaches. By implementing these classes, I was able to easily generate new shapes as meshes and apply textures to them. These classes reduced hours of work into a few minutes of implementation. This is the main benefit of reusable code. By developing a block of code once (or having it provided, thanks prof!), developers are able to reuse large blocks of code with a single call. In this case, the ShapeBuilder class converts parameters into a mesh to be rendered by the SceneBuilder class.