Logan Martha

Professor Bishop

CS – 330 Computer Graphics and Visualization

16 October 2024

**Reflection:**

1. **Justify development choices for your 3D scene. Think about why you chose your selected objects. Also consider how you were able to program for the required functionality.**
2. When creating my 3D scene I wanted to make sure it was easy to depict the actual picture. I wanted someone to be able to look at the image and easily see that it is a computer desk with specific items on it. The desk was a no brainer to choose, as it is the base of the entire image. As for the items on the desk, I wanted to pick items that weren’t just one shape objects and had the ability to add good depth to it. When creating the object, there are a variety of shapes that can be used. By changing the sizes of the shapes and orientation, you can really create an accurate depiction of the object at hand. For example, the lamp used a combination of cylinders, tapered cylinders and spheres to create its entirety.
3. **Explain how a user can navigate your 3D scene. Explain how you set up to control the virtual camera for your 3D scene using different input devices.**
   * 1. To maneuver around the scene, we take keyboard inputs for the user to be able to easily navigate different angles and viewpoints. The “global” inputs used to move are WASD. Using these four buttons alone allows the user to move left, right, forward, and backward of the scene. We can then speed up the ability to move using the mouse wheel. Simply spinning the wheel will allow the user to look around the scene faster. We also created a way to change the perspective of the scene. Clicking O or P would allow you to switch between an orthographic (3D) or a perspective (2D) scene. Lastly, by pressing Q and E, you can move up and down the scene.
4. **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?**
   * 1. The main functions that are used very often are when you are initializing the textures and object materials of the objects. When it comes to setting the textures in the scene, being able to take the title of the texture you wish to use and input it into the SetShaderTexture function saves redundant code of constantly having manual re-code to set the texture. In addition to the texture of an object, every texture reacts differently to the light sources within the scene. When creating each shape, you could create a new object material for each one, but being able to call the SetObjectMaterial function allows you to grab a specific material from basically a storage of materials that you created, instead of re-coding multiple lines of code again and again. This is super important because each object has multiple shapes to create it, and each shape is assigned a shader texture and object material. Being able to call functions to complete these tasks allows for clean code and to avoid redundancy.