Joseph Roberts

CS-330

Design Decisions

6/16/2025

**Reflection:**

For my 3D scene I took a picture of my coffee bar:



**Justify Development Choice:**

In this picture I have arranged the items to be clearly visible, I believe each of these items utilize basic shapes as well as multiple shapes to replicate this picture, I chose these objects to focus on everyday realism and material interactions, such as reflections, transparency, and different textures like ceramic and metal.

For functionality, I ensured objects have proper scale, positioning, and material properties by systematically applying transformations and shader values. My approach to modular design—with reusable functions for texture loading, transformation setting, and material assignment—ensures efficiency and scalability in scene development.

**User Navigation:**

The camera control is used by selecting keys along with using a mouse to speed up or slow down as well as rotate the camera. (Q,E) moves the camera up and down, (W,S) moves forward and back and (A,D) is left and right. This allows a user complete 360 movement and control.

**Custom Functions:**

I attempted to keep a structured approach while ensuring scalability and readability. Several key functions contribute to this efficiency. The SetTransformations() function standardizes object transformations, including scaling, rotation, and translation, preventing redundant code. SetShaderTexture() abstracts the logic behind texture assignment, allowing materials to be applied dynamically with ease. FindMaterial() streamlines object material retrieval, removing the need for manually setting shader values. CreateGLTexture() simplifies image loading and OpenGL texture configuration by encapsulating complex setup procedures into a single function. Lastly, DefineObjectMaterials() centralizes material definitions, improving organization and maintaining scene consistency. Together, these functions enhance code reusability across multiple objects, minimizing redundancy and enabling efficient scene scaling.