Ryan LeChien

Professor Enkema

CS-330-R1885

15 October 2023

**Final Project Design Decision**



Figure 1. The original image

**Two-Dimensional Objects to Be Rendered in Three Dimensions:** The image above exhibits a simple geometry in a two-dimensional plane. The objects that will be rendered in three dimensions are the floor, the area rug, the painting, the yellow credenza, the flowerpot on the yellow credenza, and the blue chair. The primitive shapes to be used are a plane, a cube, and a pyramid.



Figure 2. The rendered image

**Development Choices Justification:** This image above (Figure 1.) was selected because of the simple geometry it exhibits. The depicted objects possess detail while offering ease of rendition. Note the yellow credenza (Figure 2.): it is composed of three simple objects. Furthermore, on top of this credenza and composing a plant (a pyramid) in a pot are two more simple objects.

The painting seen in the background of the original picture (Figure 1.) is of a llama. However, the decision was made to render this object (a plane) with a different texture—specifically, the original image (Figure 1.) was used. This homage is believed to add humor to the scene.

Additionally, the epic face was used to texture the flowerpot atop the credenza. The foliage within this flowerpot is rendered as a textured pyramid to capture the growth of real plants.

Multiple light sources were used to accurately capture the lighting in Figure 1.

**Scene Navigation:** The virtual camera offers traversal of this scene via the WASD keys. Additionally, the user can toggle between perspective and orthographic displays of the 3D world by pressing P. Adjusting camera speed is also supported by the mouse scroll wheel.

**Modularization of Code:** Much of this project required repetitive operations. For example, for each primitive object, its texture had to be loaded, the vertex array object had to be bound, the vertices had to be drawn, et cetera. While this entire process was not reduced to a repeatable function, it was aided by code modularization. Take the function *UCreateTexture()*, for example.

A screen shot of a computer program

Description automatically generated

Figure 3. Example code segment in which textures are loaded

Within this section of code (specifically lines 281–343), the function *UCreateTexture()* was used multiple times to load each texture. This reduced the clustering of the same code, achieving a much more organized code body.