**Armando Gomez**

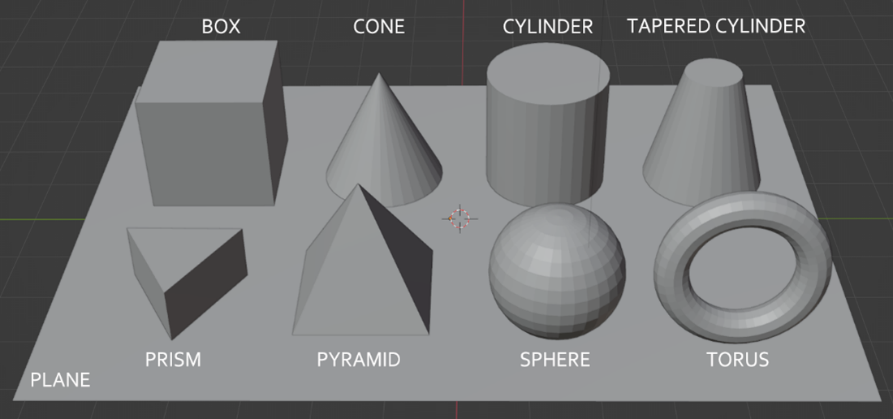
**CS-330-R4856 Comp Graphic and Visualization 24EW4**

**7-1 Final Project : Design Decisions**

**Professor Kurt Diesch**

**April 20, 2024**

****

 A group of objects on a table

Description automatically generated

During the creation of the environment for the client at Triangle and Cube Studios, we carefully balanced the selection of objects and how we added functionality to ensure both a visually appealing design and optimal performance. We decided to use low-polygon models to make sure that the rendering process was efficient while still maintaining detail to bring the scene to life. This choice was essential for creating a real-time 3D display, which we confirmed by comparing high-polygon models with low-polygon ones. The reduced polygon count significantly boosted our application's responsiveness for making interactive elements in the scene work smoothly.

Adding textures to these models presented some challenges because of their simplified shapes. I experimented with both generation and manual UV mapping in the initial stages but soon realized that procedural techniques had limitations in achieving the level of detail we wanted. By using UV mapping through the "applyTextures()" function, we gained precise control over how textures were placed on each object. This method effectively created realistic textures on basic geometric shapes like boxes and spheres in our scene, ensuring perfect alignment of high-resolution images without any stretching or misalignment issues.

The lighting played a role in shaping the overall look of the scene. While we initially considered ambient lighting, we eventually moved towards a more sophisticated shading model that included ambient, diffuse, and specular components.The function "SceneManager::render()" demonstrated the model's implementation, focusing on handling light sources to create a realistic lighting setup. This method was crucial for showcasing the textures and physical features of models, ensuring they were well balanced in terms of exposure and detail under various lighting conditions.

Improving camera navigation was part of enhancing the user experience. We found that initial trials with a fixed camera layout limited user engagement, leading us to adopt a flexible camera control system. The "updateCamera()" function managed this system, which users controlled with a keyboard (WASD for movement, QE for elevation), and a mouse (for orientation) to ensure effortless usability and comprehensive scene exploration. With this setup, users could explore scenes from viewpoints and distances for an enhanced immersive experience.

The code structure played a role in maintaining an organized and scalable project. Functions like "updateCamera()," "renderScene()," and "applyTextures()" were purposefully designed to serve functions while remaining versatile for use in various scenarios. This methodical approach allowed for adjustments and improvements to be made smoothly, showcasing project management and innovative design thinking.

The creation of this 3D environment was marked by decision-making and thorough evaluations of different technical methods. These choices, evident in the structure and features of the code, guaranteed that the final outcome was not just strong and effective but also visually striking. By blending accuracy with a creative flair, the project fulfilled the client’s needs in a clear, perceptive, and refined manner, ensuring that all components harmonized to produce an aesthetically pleasing visualization.