CS330 Final Project

Rocky Blasier

Southern New Hampshire University

CS330

Sherin Aly

10/19/25

### **Design Decisions for 3D Scene**

The last project for CS 330 was to make an easy castle-like structure in 3D. The base is a long rectangular cube and on top of it, there are two cylindrical towers, each one topped with a cone. The main objective was to show the skill of making low-polygon 3D objects, using textures, and having basic camera control. The selected objects were meant to be simple but unique enough to represent a common structure and keep triangle count under 1,000 per object at the same time.

Among the scene's main components are the long cube base, two cylinders, and two cones. The cube is the base and has a tiled pattern that looks like stone or floor material. The gold material gives the two cylinders, which stand for towers, a different look from the base and makes them a visual focus. The two cones, one above each cylinder, are also abstractly textured, and this adds to the stylistic appearance of the towers. Each object was formed from several primitive shapes per the project's rules. The use of cubes, cylinders, and cone primitives made the design modular and thus the scene could be rendered in an efficient manner.

The application of textures was done with the help of royalty-free images that had a resolution of at least 1024 by 1024 pixels or higher. The cube’s tile texture was mapped in such a way that there was a uniform spacing and alignment across its whole surface. The cylinder textures were mapped around the height and the circumference to give a metallic appearance. The cone textures were applied with the help of simple UV coordinates to produce an abstract pattern. A lot of care was taken so that texture coordinates were applied consistently throughout, which would prevent stretching or misalignment, and thus also improve the overall visual quality of the scene.

In this project, lighting was not done but proper placing of the objects made it possible to see everything clearly from every camera angle. Next time, use of point and directional lights may be made to give more depth and realism. Even without lighting, the combination of texturing and the positioning of objects led to the making of a clear and distinct 3D model that was acceptable for demonstration and early visualization.

The navigation of the camera was done through the standard input controls. The camera can be moved forward, left, backward, and right using the W, A, S, and D keys, respectively. The user was able to change the camera's distance from the objects by using the scroll wheel for zooming in and out. These controls provided a very intuitive way to change the scene and also maintain the orientation, thus offering an immersive experience that was compatible with the project's learning objectives.

Custom functions were developed for the purpose of code modularization and providing assistance in object creation. As an example, separate functions were developed that would create cubes, cylinders, and cones, and each one of these functions would return vertex and index information that was ready for uploading to the GPU buffers. These functions are not only ready for reuse but can also be employed in future projects for the efficient production of additional objects. Shader management was handled using a dedicated Shader class, which was responsible for loading and compiling GLSL vertex and fragment shaders, thus allowing for rendering textures. This method of dividing the process into smaller parts not only improved code clarity and maintenance but was also in line with the practice used in the graphics programming industry for better organization of such work.

To summarize it all, the CS 330 final project explicitly shows the skill to produce low-poly 3D models, to map the textures correctly, and to control the camera. The whole environment formed by a tiled base cube and golden cylindrical towers with abstract cone-shaped tops is suitable for the project requirements while being simple and uniform in computation. Besides, with the use of modular functions, organized code, and texture mapping, the project is poised to be a steppingstone for more intricate 3D graphics applications down the line.

**References**

American Psychological Association. (2020). *Publication manual of the American Psychological Association* (7th ed.). Washington, DC: Author.

OpenGL. (n.d.). *The OpenGL graphics system*. <https://www.opengl.org>

GLM. (n.d.). *OpenGL Mathematics library documentation*. <https://glm.g-truc.net>