Cecily Boucher

CS 330 – Final Project

10/17/2025

This project focused on transforming a simple Halloween image into a lively three-dimensional OpenGL scene. I aimed to blend straightforward shapes with a cohesive, eerie atmosphere. Building on my initial proposal, I created five key objects: a ghost, a jack-o’-lantern, a haunted castle, a glowing moon, and a ground plane. Each object was designed using basic geometric forms and was enhanced with lighting, texturing, and shading techniques to illustrate a complete 3D graphics workflow. The final result is a low-polygon scene that effectively captures the spooky essence of Halloween.

I kept the polygon count low to ensure efficient rendering. The ghost's body is a cone, with a sphere for the head, and small spheres form a scalloped edge at the bottom. For the eyes and mouth, I used simple black box meshes. The pumpkin is modeled as a textured sphere with a short cylinder for its stem and three small boxes as its carved face. The haunted castle has several box shapes for walls and towers, topped with cone roofs for an eerie silhouette. The moon is a flat cylinder with an unlit texture, simulating a glowing backdrop, while the ground is a large half-sphere providing a natural base for the scene.

I applied textures to increase realism without sacrificing performance. The castle features a “pavers” stone texture, and the pumpkin has a high-resolution skin texture applied. Each texture was mapped with careful UV scaling to prevent stretching or distortion. The materials were defined using the DefineObjectMaterials() function, setting the ambient, diffuse, and specular values according to the Phong shading model. This ensured a realistic interaction between light and surfaces, creating contrast between solid matte surfaces and translucent glowing elements.

Lighting was vital to the scene's final feel. Set up in the SetupSceneLights() function, the lighting system includes three sources: a cool moonlight as the leading key light, a warm fill light simulating lanterns and the pumpkin glow, and a subtle rim light to accentuate the object silhouettes. Each light source was configured for ambient, diffuse, and specular intensities, producing a balanced blend of warm and cool tones and natural shadows.

Camera navigation was designed to be user-friendly, working along all three axes with straightforward keyboard and mouse controls. The WASD keys maneuver the camera horizontally, forward, or backward, while Q and E allow vertical movement. The mouse controls pitch and yaw, enabling a full-range view of the scene. Scrolling the mouse wheel adjusts movement speed for precise exploration. A single key switch toggles perspective and orthographic projection modes, maintaining the same camera orientation for versatile viewing options.

The code follows best practices for formatting and structure. Each significant scene component is encapsulated in its function, like RenderPumpkin() or RenderCastle(), while shared tasks such as lighting and texturing are handled centrally. This modular design enhances readability and debugging efficiency. I also included descriptive comments throughout the code for clarity, and ensured that each primitive mesh is loaded into memory only once, showcasing effective memory management.

The final 3D composition reflects the original vision of a "fun and spooky Halloween scene" through primitive geometry. The scene captures the Halloween aesthetic by layering color, texture, and lighting while keeping shapes simple. The castle's translucent ghost and glowing windows draw the eye and add depth. Users can navigate the scene, examine the ghost's movement around the pumpkin, and enjoy the dynamic interplay of light and shadow from various perspectives. The smooth camera controls and adjustable speed enhance the immersive experience.

Custom reusable functions contribute to the project's modularity and flexibility. For instance, the DefineObjectMaterials() function is central to the scene's overall coherence and efficiency.