Final Project Design Decisions

Rationale for 3D Scene Development Choices

For my 3D scene, I selected a padlock and a tennis ball as my primary objects and placed them on a table to create a realistic composition. I ensured that both objects were accurately represented while maintaining a low polygon count for computational efficiency. The padlock was constructed using a box for the lock body and a half torus for the shackle, capturing its essential shape while keeping the geometry simple. I applied a golden texture to the lock body to create a metallic effect and a steel texture for the shackle to resemble polished metal. The tennis ball was modeled using a textured sphere, with a fuzzy green texture to simulate the real felt covering.

To enhance the visual appeal and realism of the scene, I used multiple light sources, including a point light inside the padlock, a directional light from above, and a spotlight in front of the padlock. These lights ensure that the padlock's metallic surface reflects light realistically while also making the tennis ball's texture clearly visible. Additionally, I carefully positioned the padlock and tennis ball on the table, ensuring that the padlock appears to be resting securely while the tennis ball maintains a natural placement to match the reference image. The table itself was textured to provide a realistic surface, reinforcing the depth and spatial arrangement of the scene.

User Navigation and Camera Control in the 3D Scene

I implemented an intuitive camera navigation system that allows users to explore the padlock and tennis ball from multiple angles. The movement is controlled using the W, A, S, and D keys for forward, left, backward, and right movement, while the Q and E keys allow vertical movement. The mouse cursor controls the camera's orientation, enabling users to rotate their view and closely examine both objects. The scroll wheel adjusts movement speed, allowing for smooth exploration of finer details, such as the reflective surface of the padlock or the texture of the tennis ball.

I also implemented both perspective and orthographic views, enabling users to switch between a 3D perspective view and a flat, blueprint-like orthographic view with a simple key press. This functionality helps in better visualizing the spatial relationships between the objects, especially for the 3D printing requirements specified by the client. Additionally, the camera ensures that all objects are well-lit and visible from different viewpoints, preventing any dark or obscured areas in the scene.

Modular Code Structure and Custom Functions

To maintain a structured and modular codebase, I created several custom functions that handle different aspects of my 3D scene. The SceneManager class is responsible for creating and managing the 3D objects, ensuring that the padlock and tennis ball are loaded, textured, and rendered properly. Within this class, I wrote functions for setting transformations, which allow me to scale, rotate, and position objects accurately within the scene. For example, the padlock's

shackle transformation involves precise rotations and positioning using the SetTransformations function to ensure it aligns correctly with the lock body.

The lighting setup is modularized within SetupSceneLights, where I configured different light sources to properly illuminate the objects. The point light inside the padlock enhances its metallic reflections, while the spotlight in front of it ensures the scene has a polished and professional look. I also used the SetShaderMaterial() and SetShaderTexture() functions to assign materials and textures efficiently, ensuring that each object maintains a distinct appearance.

The camera controls are handled within the ViewManager class, allowing seamless navigation across the 3D space. I wrote functions to handle keyboard input for movement, mouse input for orientation, and scroll wheel adjustments for speed control. Additionally, I ensured that the camera could smoothly transition between perspective and orthographic views using a dedicated function, making it easier for users to examine the scene from different visual perspectives.

By keeping my code well-structured and modular, I made it easy to modify individual components without disrupting the overall functionality of the program. This approach ensures that the 3D scene remains efficient, interactive, and visually engaging.

References

Joey de Vries. (n.d.). *Lighting: Colors*. LearnOpenGL. Retrieved February 23, 2025, from https://learnopengl.com/Lighting/Colors

Southern New Hampshire University. (n.d.). *Applying textures to 3D shapes* [PDF]. Retrieved February 23, 2025, from https://learn.snhu.edu/content/enforced/1798873-CS-330-13647.202511-1/course_documents/CS%20330%20Applying%20Textures%20to%203D%20Shapes.pdf

Joey de Vries. (n.d.). *Getting started: Textures*. LearnOpenGL. Retrieved February 23, 2025, from https://learnopengl.com/Getting-started/Textures