# CS 330 Computer Graphics & Visualization 7-1 Project - Reflection

**Professor Philip Enkema (MSwE)**

**Mubeen Ahmed Khan**

**mubeenahmed.khan@snhu.edu**

**Southern New Hampshire University**

**23rd February, 2025**

## Justification of Development Choices

The development of the 3D scene was driven by the need to create an immersive and functional environment that incorporates various elements such as custom shapes, textures, lighting, and camera movement. The selection of objects, particularly the HalfTorus and SemiSphere, was influenced by the requirement to construct complex geometries that enhance realism and functionality. The modification of ExtraTorusMesh1 for use as the thermos handle demonstrates adaptability and the ability to repurpose assets to serve specific design goals.

Custom-designed shapes played a crucial role in achieving the desired level of detail in the scene. The **HalfTorus** was specifically designed to function as a handle for a cup, allowing for a smooth curvature that provides an ergonomic grip appearance. Its construction involved careful consideration of vertex distribution to ensure smooth shading and realistic light reflections. Similarly, the **SemiSphere** was introduced as a cap element, offering a rounded and seamless transition for cylindrical objects. These redesigned shapes added structural variation and improved the overall composition of the 3D models, demonstrating the flexibility of custom geometry creation.

Textures were integrated into the scene to provide realistic surface properties and enhance visual appeal. By leveraging shader management techniques, lighting and reflections were meticulously configured to ensure accurate representation of materials and illumination effects. The choice to attach light sources to the camera further enhances scene navigation by dynamically adapting to changes in the viewer's perspective.

Orthographic and perspective views were incorporated to offer flexibility in visualization, catering to different user preferences and interaction needs.

## Navigation and Virtual Camera Control

Users can navigate the 3D scene through a virtual camera system that responds to various input controls. Camera movement is achieved through keyboard inputs, allowing translation along different axes. Mouse inputs are utilized for adjusting the camera’s orientation, providing an intuitive way to look around the scene.

The system supports toggling between orthographic and perspective views using dedicated key bindings ('O' for orthographic and 'P' for perspective). Additionally, the camera’s zoom level and movement speed can be modified dynamically, ensuring a customizable and user-friendly navigation experience. A spotlight attached to the camera further enhances scene visibility by illuminating objects as the user moves through the environment.

## Custom Functions and Code Modularity

To maintain an organized and modular code structure, several custom functions were implemented. These include:

* **LoadSceneTextures()**: Facilitates the loading and binding of textures to enhance the realism of 3D objects, up to 16 textures can be loaded.
* **SetupSceneLights()**: Defines multiple light sources, including point lights, directional lights, and spotlights, to improve scene illumination via the Phong Lighting Model.
* **PrepareSceneView()**: Manages the configuration of view matrices, ensuring seamless transitions between different camera perspectives.
* **DrawSemiSphereMesh(), DrawSphereMesh(), DrawHalfTorusMesh(), DrawPyramid3Mesh(), DrawTaperedCylinderMesh(), DrawCylinderMesh()**: Modular functions dedicated to rendering distinct 3D shapes, making the code reusable and scalable.

These functions contribute to code reusability by encapsulating key functionalities and promoting a structured approach to scene management. The use of encapsulation ensures that modifications can be made without affecting the overall integrity of the codebase, making future enhancements more manageable.

## Conclusion

In summary, the development of the 3D scene adhered to best practices in graphics programming, ensuring that the environment is visually appealing, interactive, and efficiently managed through modular programming principles.