Objects: The chosen objects for the 3D scene are a coffee mug, a pitcher, and a plan. These objects were selected based on the theme of a coffee scene and to create a visually appealing composition. The coffee mug and pitcher represent key elements of a coffee set, while the plane adds an additional decorative element.

Color and Texture: Pink was chosen as the color for the objects to create a vibrant and eye-catching scene. The brick texture was applied to the objects to add depth and visual interest, as it provides a contrast to the smooth surfaces of the coffee mug and pitcher. The texture choice also adds a tactile quality to the scene, giving the objects a more realistic appearance.

To program the functionality of the 3D scene, the following steps were taken:

Object Placement: The coffee mug, pitcher, and plane, the coffee mug had a right side handle, and the Pitcher is far enough away from the cup to show placement correct placement.

Texturing: The pink color and brick texture were applied to the objects . The UV mapping technique was used to ensure the texture fits properly on the objects' surfaces.

Lighting: Lighting was set up in the scene to provide appropriate illumination and enhance the visual appeal. Different light sources, such as directional lights or point lights, were placed strategically to achieve the desired lighting effects.

Users can navigate the 3D scene using different input devices, such as a keyboard and mouse. The virtual camera can be controlled using the following methods:

Keyboard and Mouse: The user can control the camera's movement by using the WASD keys to move forward, backward, and strafe left or right. The QE keys can be used to move up or downwards in the camera view. The mouse cursor is used to change the orientation of the camera to look up, down, and sideways. The mouse scroll is used to increase the speed of the movement by zooming in and out.

To make the code more modular and organized, custom functions can be created for specific tasks. The provided code in the function UCreatePitcherMesh creates a mesh for a pitcher object with a cup and plane in OpenGL. It defines the vertex positions, normal, texture coordinates, and flags for each vertex of the mesh. The mesh is created using a series of triangles that form a cup shape for the pitcher and cup along with the handle and plane. Each face of the two cup-like shapes is divided into two triangles (bottom and top). The vertices are specified in counterclockwise order when looking at the face from the outside.

Here's a breakdown of the code:

The function begins by declaring some variables, including the height of the pitcher, deg (initially 0), rad (the conversion of deg to radians), and x and z (the x and z coordinates of a point on a circle of radius 1). The variable midPos is set to a fixed value of 0.70710678f, which represents the middle position of the cup-like shapes. The variable nozzle base is set to 80% of the height of the cup. The next part of the code defines an array of verts that contains the position, normal, texture coordinate, and flag data for each vertex of the mesh. The vertices for each face are specified in groups of three, forming triangles. For each vertex, the position (x, y, z) is given, followed by the normal vector (nx, ny, nz), the texture coordinates (u, v), and the flag value. The code continues to define the vertices for each face of the octagon, including the front, front-left, front-left-side, back-left, back, and back-right faces, as well as the base of the octagon.

Overall, this code generates the vertex data for a pitcher mesh with an octagonal shape and assigns positions, normal, texture coordinates, and flags to each vertex. The mesh can then be rendered in an OpenGL application using this vertex data.

This project presents an exciting opportunity to showcase skills learned throughout this course as practice being a 3D graphics developer. By recreating a 3D version of a 2D image, applying textures, implementing lighting, and incorporating interactive controls. creating a visually appealing and engaging 3D scene. Through careful consideration of design choices, effective navigation controls, and modular code organization, learning to deliver a project that meets the client's requirements.