**Image:**

**A bread and orange on a table

Description automatically generated**

**Scene:**

**A computer screen shot of food on a table

Description automatically generated**

**Shape/Object Description:**

I selected my four objects based on their unique shapes and textures. My scene includes a shiny plastic bottle. The bottle object consists of two cylinders, a tapered cylinder, and a torus. The base cylinder needed to be significant to imitate the bottle; the bottle is simply transparent, so I opted not to use a texture for this shape. I merely made the color almost white with medium transparency. The material uses low diffuse, with high specular and shininess, to imitate the plastic sheen of the bottle. The top of the bottle is made of a tapered cylinder using the same color and material. The strap is a torus elongated and is placed to connect the lid to the bottle. The color is an orangish yellow, and a flat plastic material is recreated with medium diffuse, low specular, and low shininess to create a matte look. The lid is made of a shortened cylinder with flat plastic material and a texture that creates the look of blue vertical ridges.

The loaf of bread is created using a box and a cylinder. The cylinder is stretched along the z-axis to give it an oval shape. The texture used resembles the top of a loaf of bread. The base is a box. The texture is dirt, but it achieves the coarse brown look I desired. The material for both objects uses a brown diffuse light pattern, with low specular and shininess, to create a natural look.

The butter container uses two cylinders and two tapered cylinders. The cylinders make up the base and top. The lid cylinder is short and slightly wider than the base to hangover. The tapered cylinders are stacked to create an hourglass-shaped handle on the lid. The material is used to resemble glass with high specularity and shininess. A texture on the base labels the container “butter,” and the other shapes are an off-white color to resemble the base.

The orange is a sphere and a cylinder. The cylinder is the stem and uses a brown color with the same material as the bread. The sphere uses an orange ridged texture and an image of pale, cracked skin to recreate the surface of the orange. The material on the orange uses low specularity and shininess with an orange diffuse to make the orange seem natural.

**Lighting:**

There are three light sources in the scene: two point lights and one spotlight. The first point light is positioned to the left of the scene, creating a warm glow with red highlights. The second is placed to the right and uses a cool blue setting. These lights create the scene's ambiance and ensure the lighting is accurate to the image. The spotlight imitates the scene’s actual light source, ensuring the proper specular highlights in the brightest areas of the image.

**Camera/Movement:**

The camera can be controlled using keyboard and mouse inputs. The program leverages OpenGL callbacks to capture mouse and keyboard inputs while the OpenGL window is open. The keyboard controls the camera position via the W, A, S, D, Q, and E keys. When a user presses the key, the camera adjusts forward, left, backward, right, up, or down respectively. The view matrix is adjusted appropriately, creating movement around the scene. Additionally, the O and P keys adjust the projection type. O can be pressed anytime in the window to switch to an orthogonal projection of the scene, and P switches back to the perspective-based projection.

The mouse wheel controls movement speed and camera pitch/yaw. Its y-offset adjusts how much each keypress moves the camera. Scrolling down allows the camera to move less; scrolling up increases the distance. The cursor controls pitch and yaw, or how much the camera looks up/down and left/right. The mouse's x-offset controls the yaw, while the y-offset controls the pitch.

**Custom Functions:**

The program we were given to start the assignment was well-designed and modular; however, I included several custom functions to achieve the desired scene.

**ViewManager:**

* LoadGLTextures() – Used to abstract the creation of the textures. Utilizes function to create and bind textures that will be present in the scene. It provides file path and texture tag to helper function in making the textures for the scene.
* SetTwoTextures(): This method replaces the SetShaderTexture() method. It allows the developer to utilize two textures in the same shape. The method uses the texture tags created to locate the index of the desired textures and activate them before passing them to the shader. The fragment shader had to be significantly edited to achieve this as well. When calculating the Phong result, I edited the program to mix the textures instead of using the single texture sampler. Then, I calculated the Phong result based on the matrix created by combining the textures. Then, I also had to consider this in the fragment color process.
* CreateMirroredTexture: This uses the same pattern as CreateGLTexture but sets the wrapping method to GL\_MIRRORED instead of GL\_REPEAT.
* DefineObjectMaterials: Abstracts the creation of shader materials determining how objects interact with light sources.
* SetUpSceneLights(): Abstracts the creation of light sources.

**ViewManager:**

* The keyboard callback function was edited to include up/down (Q/E) movement and the projection change (O/P).
* Mouse\_Scroll\_Wheel\_Callback(): Captures mouse scroll wheel events to adjust movement speed or the amount the camera adjusts with each key press.