**Scene:** Cutting board with vegetables

**Objects:**

* Cutting board (box)
* knife (thin prism for blade, cylinder for handle)
* Tomato (spheres)
* Pepper (cylinder)

A tomato and a knife on a cutting board

Description automatically generated

For my 3D scene, I chose to create a simple kitchen setup featuring a cutting board, knife, tomato, and pepper. These objects were selected because they can be easily modeled using basic geometric shapes, which keeps the development process straightforward while still allowing for a recognizable and realistic look. The cutting board, represented by a box, serves as the base for the other items. The knife uses a thin prism for the blade and a cylinder for the handle, making it easy to construct while still capturing the essence of a kitchen utensil. For the tomato, a sphere was used to create a smooth, round shape, while the pepper is represented as a cylinder to mimic its natural elongated form. A flat plane acts as the surface of the kitchen island, providing context and grounding the objects. This selection of simple shapes allows for a focus on the textures and lighting, resulting in a cohesive scene without overwhelming complexity.

To navigate the scene, a control system combined with keyboard and mouse inputs were implemented. Users can move around the kitchen using the **W** and **S** keys to go forward and backward, **A** and **D** to navigate left and right, and **Q** and **E** for adjusting the camera’s height. The mouse controls the camera’s rotation, letting users look around the scene. This setup involves adjusting translation and rotation matrices based on the input, ensuring a smooth and responsive navigation experience.

Custom functions were utilized to maintain a modular and organized code structure. Functions such as drawCuttingBoard(), drawKnife(), drawTomato(), and drawPepper() handle the rendering logic for each object. This encapsulation makes it easy to adjust or reuse these functions individually without impacting the rest of the scene. Additionally, a function like handleInput() processes user inputs to update the camera's position and orientation based on keyboard and mouse controls. This approach simplifies input handling, resulting in cleaner and more readable code. Overall, these design choices contribute to a more efficient and manageable coding process.