1. **Object Selection:**
   * **Cubes for Sofa:** I chose cubes to represent the sofa because they are simple geometric shapes that are easy to create and manipulate. This choice aligns with the simplicity of the scene and allows for straightforward rendering and collision detection if needed.
   * **Planes for Walls, Floor, and Ceiling:** Using planes for the walls, floor, and ceiling also adds simplicity to the scene. Planes are versatile and provide a clean and minimalistic look. The use of textures on the planes enhances visual appeal without increasing the complexity of the geometry.
2. **Lighting:**
   * **Ceiling Light Fixture:** I implemented a simple light source to simulate a ceiling light fixture. This enhances the overall ambiance of the scene and demonstrates the application of lighting techniques in 3D graphics.
3. **User Navigation:**
   * **Camera Control:** I implemented a custom camera class with functions for camera movement and orientation. The camera can be controlled using input from the mouse and keyboard, allowing users to explore the 3D scene interactively. This choice provides a familiar and intuitive way for users to navigate the environment.
4. **Input Devices:**
   * **Mouse and Keyboard:** I chose to use input from the mouse and keyboard for camera control. The mouse allows for intuitive rotation, while the keyboard facilitates movement along the camera's forward, backward, left, and right directions. This combination provides a flexible and user-friendly navigation experience.
5. **Modularity and Organization:**
   * **Shader Class:** I implemented a shader class to encapsulate shader-related functionalities. This promotes code modularity, making it easier to manage and reuse shader code for different objects in the scene.
   * **Camera Class:** The custom camera class encapsulates camera-related operations, enhancing code organization. It allows for easy adjustments to the camera's position, orientation, and field of view. This modular approach makes the code more readable and maintainable.
   * **Texture Loading Function:** I created a function to load textures, promoting reusability. This function can be used to load textures for various objects in the scene, enhancing efficiency and reducing redundancy.

In summary, the development choices aimed at simplicity, visual appeal, and user-friendly interaction. The use of modular functions and classes contributes to code organization and reusability, fostering a maintainable and extensible codebase. The selection of input devices and camera control mechanisms enhances the user experience, allowing for seamless exploration of the 3D environment.