I chose my objects for this 3D project because they are some of my favorite fitness items at home. The combination of a core ball, dumbbell weights, a massage stick, and a yoga block reflects tools I regularly use in my personal fitness routine. I felt these objects would be ideal for this project as they possess simple geometric shapes, making them perfect candidates for still modeling. Their basic forms—spheres, cylinders, and cuboids—lend themselves well to 3D rendering while maintaining a personal connection to my fitness journey.

Programming these objects into the project involved defining their shapes using standard 3D primitives, such as spheres for the core ball, toruses for the weights, cylinders for the connecting bar and massage stick, and a cuboid for the yoga block. I used OpenGL's built-in functions to create and manipulate these shapes, ensuring they were scaled, rotated, and positioned accurately within the scene. By adjusting material properties and lighting effects, I was able to achieve a realistic representation of each object’s surface, from the smooth texture of the core ball to the reflective metal of the weights. This combination of simple shapes and material programming brought the scene to life, matching my vision for this 3D project.

In my 3D scene, users can navigate the environment using a combination of keyboard and mouse controls. The \*\*WASD\*\* keys allow for forward, backward, and lateral movements, enabling users to explore the scene horizontally. The \*\*Q\*\* and \*\*E\*\* keys control vertical movement, allowing users to move up and down within the 3D space. Additionally, the \*\*mouse scroll\*\* can be used to zoom in and out, offering more precise control over the view. Pressing the \*\*P\*\* key switches the camera to a \*\*perspective view\*\*, providing a more realistic 3D experience with depth perception, while pressing the \*\*O\*\* key changes it to an \*\*orthographic view\*\*, which eliminates perspective distortion for better inspection of object proportions. I set up these navigation controls by integrating input handling functions that map keyboard and mouse inputs to camera transformations, ensuring a smooth and intuitive exploration of the 3D scene using standard input devices.

I employ functional coding logic best practices to ensure that my program runs as expected. By using clear, organized, and efficient code, I make it easier to understand, maintain, and debug. Each function is designed with a specific purpose, adhering to the principle of single responsibility, which helps prevent code duplication and reduces complexity. Additionally, I use meaningful variable and function names, making the code more readable and self-explanatory. Consistent indentation and spacing further enhance clarity, allowing me to easily follow the flow of logic throughout the program. I also implement error handling and checks to ensure the stability of the program, particularly when managing input devices or rendering objects in the 3D scene. This approach ensures that the program is not only functional but also reliable and adaptable for future modifications or expansions.