For my 3D scene, I decided to create a garden because it gave me the chance to include several detailed objects that met the project requirements. I picked a few main objects like trees, a bench, a flowerbed, and a walkway. These choices allowed me to show that I could build complex shapes using basic 3D shapes such as cones, cylinders, and cubes. I used cones and cylinders for trees because the cylinder worked well for the tree trunk, and the cone looked like the leafy top. For the bench, I used rectangular prisms to form the seat and the legs. The flowerbed was made with a base box and small cones to look like flowers. The walkway was created using textured rectangles to help guide the user through the garden. I picked these objects because they look good together and give the user something nice to explore. I also made sure these objects showed my ability to transform and combine basic shapes in OpenGL.

To program the scene, I used the provided Visual Studio project with the OpenGL libraries already set up. I created each object by writing individual functions to build and place them in the 3D space. For example, I used transformation functions like glTranslatef() and glRotatef() to move and rotate objects. I also added textures and lighting to make the scene feel more real. I used glEnable(GL\_LIGHTING) and glEnable(GL\_LIGHT0) to turn on lighting, and I changed the color and position of the light to make sure nothing was in full shadow. I used ambient, diffuse, and specular lighting to give the objects depth and brightness. This helped fix some of the early lighting issues I had where some objects were too dark.

For navigation, I made it so users can move around the garden using both the keyboard and the mouse. This gives them control over both movement and view direction. I set it up so the W key moves forward, S moves backward, A moves left, and D moves right. This is similar to first-person games, which makes it familiar and easy to use. I also added mouse movement to let the user turn the camera and look around. When the mouse moves, the camera angle updates to follow. This makes the scene feel more interactive and gives the user a way to explore every part of the garden. I used variables to store the camera’s position and direction, and I updated them in my input functions whenever a key or mouse input is detected. This part took some adjusting, but once it was working, it made the scene much more fun to use.

To keep my program clean and easy to understand, I created custom functions for the different objects and repeated tasks. For example, I made a drawTree() function that builds a tree from a cone and a cylinder. I could call this function many times with different position values to place trees all over the garden. I also made a drawBench() function for the bench, and a drawFlower() function for the flowers in the flowerbed. These functions take in location values, so I can reuse them anywhere in the scene without copying all the code again. Another useful function I created was setLighting(), which sets up the lighting in the scene in one place. That way, if I needed to change the light color or brightness, I only had to change it once in that function instead of all over the code.

Creating these functions helped make my code more modular and organized. Each part of the scene has its own code section, which makes it easier to read and fix if needed. If I want to add new objects later or change how something looks, I just go into that one function instead of searching through all the code. This saves time and helps avoid errors. These functions are also reusable for future projects. For example, I could use the same drawTree() function in a different 3D scene, like a forest or park, just by copying the code and changing the locations.

In the end, my development choices were focused on meeting the project goals while making the scene look good and run well. I used simple objects to build a more complex space and organized the code using custom functions. I added keyboard and mouse controls for navigation and used lighting and textures to improve the visuals. I feel that my scene is both interactive and complete, and I’ve shown how to build and organize a 3D scene using OpenGL in a way that is both creative and functional.