3D Visualization & Command

Brief: Develop a program and GUI to visualize and command multiple vehicles in a 3D environment and update a terrain map based on their sensor measurements.

Objective

The goal of this challenge is to develop a **real-time GUI application** that visualizes a **3D scene** using **LiDAR data streams** from five autonomous "rovers." These rovers would be construction machines such as excavators, bulldozers, and roller-compactors. The application should provide both **a dynamic visualization** of the environment and **a control interface** for interacting with the rovers.

Core Feature Requirements

1. **3D Visualization of Rover & LiDAR Data**
    - The application should **render and update the environment in real-time** using the data streams available from the 5 rovers.
    - Each rover's **position and orientation** should be represented visually
    - The visualization should **update dynamically** as new LiDAR data arrives.
    - The **terrain should be mapped and updated** from incoming LiDAR points as the rovers explore. The result should be a persistent surface of where the rovers covered
2. **Camera & Scene Navigation**
    - The user should be able to **freely reposition the camera** in the scene to inspect different parts of the environment
    - There should be an easy way to **focus on specific rovers** and follow their movements
3. **Rover Control & Status Interface**
    - The application should provide an interface to **select and monitor** a rover
    - The interface should display:
        - The rover’s **current position and orientation**.
        - The **status of its command buttons** (4 total, on/off).
    - Users should be able to **send commands** to control a rover's buttons 0 through 3
    - **Seamless switching** between rovers should be supported.
4. **Performance & Usability**
    - The solution must **run with minimal latency**, sub 50ms end-to-end.
    - The application should run on a **Linux system** and should be as lightweight as possible, not requiring overly specialized hardware to run without lag (sample computer would be a Lenovo ThinkPad T16 Gen 3 running Ubuntu 22.04)
    - The UI should be **intuitive** for users to switch between viewing rovers and issuing command

All data exchange should occur via **UDP on localhost**.

This challenge tests your ability to build a **real-time, interactive 3D visualization system**, efficiently process incoming data, and design an intuitive **GUI for controlling autonomous vehicles**. Code for emulating the rovers, along with a detailed README is included in the attached zip file.

You may use any resources of your choosing (including textbooks, internet, AI tools, etc.); however, you may **not** share the problem statement or materials with others.

Send source code with compilation instructions that can run on a linux system. You can also optionally include an executable or playable demo in any format.