**Project Summary: Fire Rescue Grid Simulation**

**Goal:**  
Simulate a fire rescue operation where a robot navigates a grid to rescue civilians trapped in a burning building and take them to the exit safely.

**💡 Core Features Implemented**

1. **ROS-Based Architecture**
   * Modular nodes:
     + environment\_node.py: Manages the grid environment and updates state (fire spread, civilians, etc.)
     + sensor\_node.py: Feeds sensor data (3x3 view) to the planner.
     + state\_generator.py: Broadcasts the full grid state.
     + planner\_agent.py: Plans the robot’s next move using AI logic (A\* or greedy).
     + executor\_agent.py: Executes planner’s decisions.
     + validator\_agent.py: Validates movement (boundaries, collisions).
     + movement\_controller.py: Moves the robot physically in grid.
     + grid\_gui\_node.py: GUI node for visualizing the grid (now with Pygame!)
2. **Grid Environment Logic**
   * 10x10 grid with symbols:
     + 'R' = Robot
     + 'C' = Civilian
     + 'X' = Rescued civilian
     + 'F' = Fire
     + 'E' = Exit
     + '\_' = Empty
   * Civilians turn to 'X' when picked up.
   * Robot drops civilians at exit.
3. **Fire Spread Simulation**
   * Fire dynamically spreads every few seconds.
   * Civilians caught in fire are marked.
4. **Mission Control Logic**
   * Robot picks up civilians and brings them to exit.
   * MISSION COMPLETE ✅ is triggered only after all civilians are dropped off at exit.
5. **Visual Grid Display (Tkinter ➝ Pygame)**
   * Initially used Tkinter — later migrated to Pygame for better performance and color control.
   * Real-time display:
     + Colored grid tiles with proper mapping.
     + Centered "Mission Complete" banner.

**🛠️ Technologies Used**

* **ROS (Robot Operating System)** – for inter-node communication
* **Python 3** – core programming
* **Pygame** – grid GUI
* **Custom Planner** – logic to select actions based on robot and civilian positions

**🎯 Final Output**

* ✅ Working real-time robot navigation.
* ✅ Dynamic fire spread.
* ✅ Real-time visual display.
* ✅ Accurate civilian pickup/drop-off logic.
* ✅ Mission completion tracking and visual confirmation.