SYNTHETIX 3.0

PROGRAMMING PROBLEM STATEMENTS

Each participant must select a single challenge and develop a solution for it.

CHALLENGE 1 Autonomous Rescue Rover for Disaster Zones

Problem Statement

Disaster response requires autonomous rovers to navigate unstructured terrain, locate survivors, and deliver aid under difficult conditions.

Scope

Develop a web app for a single autonomous rover that can:

- Navigate disaster zones **without GPS**.
- Detect & identify survivors using **ultrasonic, IR, RFID, or accelerometer data**.
- Deliver aid & return safely with **minimal human intervention**.
- Operate with constrained power & communication.

Data & API Usage

Start a new session:
POST https://roverdata2-production.up.railway.app/api/session/start

All API calls must include the generated session ID.
Example: Check fleet status:

Get fleet status:
GET https://roverdata2-production.up.railway.app/api/fleet/status?session_id=<SessionId>

API: Disaster Rover Data

- The rover stops moving when recharging.
- Recharging starts at 5% and stops at 80%.
- If the battery is below 10%, communication is lost. Communication is regained only after rover is recharged above 10%.

Bonus Points

- Simulate rover behavior entirely in Webots with realistic sensor data.

Constraints (Choose at least one)

- **No GPS/External Positioning**: Self-localization required.

- **Intermittent Communication**: Data transmission at fixed intervals only.
- **Limited Sensor Inputs**: Use only ultrasonic, IR, RFID, or accelerometer data.

Tech Stack (Recommended, Not Mandatory)

Languages: Python, C++| **Simulation**: ROS, Webots, Gazebo
Navigation: PID, state machines, path-planning logic
Sensors: Simulated ultrasonic, accelerometer, RFID, IR
Networking: LoRa, MQTT, serial data transfer

CHALLENGE 2 Autonomous SLAM-Based Fleet for Smart Agricultural Supervision

Problem Statement

A fleet of SLAM-enabled agricultural robots (Slambots) can enhance farm supervision through distributed monitoring and autonomous navigation.

Scope

Develop a web app for a fleet of Slambots that can:

- **Navigate**: Use SLAM for large farmlands.
- **Distributed Monitoring**: Collect & analyze soil moisture, temperature, and crop health.
- **Adaptive Collaboration**: Share data for optimized task distribution.
- **Fail-Safe Operations**: Handle sensor failures, communication delays, and power constraints.
- **Remote Supervision**: Enable farmer monitoring via a web/mobile interface.

Data & API Usage

Session-based isolation required. Start a session with:
POST https://fleetbots-production.up.railway.app/api/session/start

All API calls must include the generated session ID.
Example: Check fleet status:
GET https://fleetbots-production.up.railway.app/api/fleet/status?session_id=<SessionId>

(Refer to user guide for detailed API documentation.)

Bonus Points

- Simulate fleet behavior in Webots with sensor-based data generation.

Constraints (Choose at Least One)

- **Limited Communication** – Predictive decision-making required.
- **Low-Power Optimization** – Minimize energy consumption.
- **Sensor Failures & Redundancy** – Handle dropouts/inaccuracies.
- **Localization Without GPS** – Alternative mapping techniques.
- **Multi-Robot Coordination** – Optimize fleet-wide task execution.