Assignment: Enhanced Dynamic Robot Movement Simulation

Objective: Design and implement an advanced simulation environment for a robot navigating through a dynamically created grid. This project aims to deepen understanding of basic programming concepts, object-oriented programming (OOP), algorithms for navigation and pathfinding, task optimization, safety, and energy management strategies.

Overview: Develop a simulation for a robot moving through a grid-based environment considering task optimization strategies and safety to ensure efficient travel, collision avoidance, and effective energy management. The simulation should manage the robot's energy levels and battery status, incorporating the necessity of recharging.

Detailed Requirements:

1. **Environment Setup**
    - Implement a class **Environment** that generates a 10x10 grid. Dynamically place obstacles, a start position, and an end position within the grid.
2. **Robot**
    - Implement a class **Agent** with movement capabilities and tracking of its current position. Include methods to manage the robot's energy levels and battery status, incorporating task optimization and safety for efficient and safe navigation.
    - **Battery Management:** The robot starts with a battery level of 100%. For each move from one block to another, the battery level decreases by 10%. If the battery level reaches 0%, the robot must recharge to 100% before continuing.
3. **Simulation**
    - Simulate the robot's movement through the 10x10 grid with randomly placed obstacles, accounting for energy consumption and managing energy levels to complete tasks.
4. **Pathfinding Algorithms**
    - Students are required to implement two pathfinding algorithms: Uniform Cost Search (UCS) and A* (A Star). Evaluate these algorithms based on the number of times the robot needs to recharge its battery while traversing the path to the goal. This will determine the best algorithm for this environment.
5. **Visualization**

   - Use libraries like matplotlib to visualize the grid, obstacles, paths, and the robot's energy levels over time.

6. **Task Optimization and Safety**

   - Implement strategies to minimize travel time, energy consumption, and ensure robots can detect and avoid potential collisions, maintaining safety.

7. **Bonus (Optional)**

   - Explore advanced pathfinding algorithms and consider different terrain types affecting movement cost and energy consumption.

Deliverables:

- GitHub Project Submission with a structured repository containing the Jupyter notebook, a detailed report or presentation, and README.md files providing an overview and instructions for running the simulation.

Submission Guidelines:

- Ensure accessibility to instructors and compliance with the provided instructions. Organize files neatly and document your work comprehensively.

Evaluation Criteria:

- Correct implementation, functionality, efficiency of the robot's movements, energy management, and clarity of submitted code and documentation will be considered.

Additional Instructions:

- Students may use all source code available in the Google Classroom. A viva will be conducted for evaluation, focusing on the understanding and implementation of the algorithms, design choices, and functionality of the simulation.