Project Title:

"Dynamic Multi-Agent Pathfinding in an Evolving Maze with Swarm Intelligence"

Overview:

In this project, you will develop a Python-based simulation of multi-agent pathfinding in a **dynamic maze**. The agents must navigate from their starting points to unique target destinations in a **perfect maze** that evolves over time, with walls opening and closing during the simulation. The project will utilize **Swarm Intelligence** concepts, such as Ant Colony Optimization (ACO) or Boids, for coordination.

Enhancements:

1. **Perfect Maze Generation**:
   Use an algorithm (e.g., Prim’s or Recursive Backtracking) to generate a **perfect maze** (a maze with no loops and exactly one path between any two points).
2. **Dynamic Obstacles**:
   Maze walls can open or close at runtime, requiring agents to adapt their paths dynamically.
3. **Real-Time Swarm Intelligence**:
   Agents will use Swarm Intelligence strategies to adjust their behavior in real-time, maintaining efficiency and avoiding collisions.

Project Objectives:

1. **Perfect Maze Generation**:
   - Implement a maze generation algorithm to create the grid world.
   - Visualize the generated maze for validation.
2. **Dynamic Maze Updates**:
   - Implement a mechanism to modify the maze during runtime: Open paths can close.
   - Ensure agents can handle sudden changes in the environment:changing end goal position.
3. **Multi-Agent Pathfinding**:
   - Use Swarm Intelligence to allow agents to navigate collaboratively.
   - Include components like alignment, separation, and cohesion for path optimization.
4. **Optimization & Adaptability**:
   - Minimize the total path cost while handling maze changes.
   - Evaluate performance under various scenarios (e.g., static vs. dynamic environments).

Deliverables:

1. **Codebase**:
   - Fully functional simulation with agents navigating an evolving maze.
   - Modular, well-documented Python code.
2. **Visualization**:
   - Real-time visualization of the maze and agent paths using **Pygame**.
   - Dynamic updates to the maze structure during runtime.
3. **Report**:
   - Documentation of the maze generation, dynamic updates, and Swarm Intelligence implementation.
   - Analysis of results, including metrics like path cost, time, and adaptability.
4. **Optional Extensions**:
   - Agents can learn optimal strategies using reinforcement learning.
   - Introduce random events, such as temporary goals or agent-specific constraints.

Suggested Workflow:

1. **Week 1: Research & Maze Generation**:
   - Study algorithms for perfect maze generation (e.g., Recursive Backtracking).
   - Implement the maze generation and visualize the result.
2. **Week 2: Dynamic Maze Updates**:
   - Add a mechanism for walls to open and close randomly or based on predefined rules.
   - Test agent adaptability in the dynamic maze.
3. **Week 3: Multi-Agent Pathfinding**:
   - Implement Swarm Intelligence for agent navigation.
   - Integrate maze updates with agent pathfinding.
4. **Week 4: Finalization**:
   - Optimize the algorithm and test under various conditions.
   - Prepare the report and finalize the project.

Steps Required:

- Research & Design
- Algorithm design (e.g., maze generation)
- AI and optimization techniques (e.g., Swarm Intelligence)
- Visualization using **Pygame**

Evaluation Criteria:

1. Quality of the maze generation and its adherence to the "perfect maze" criteria. (10%)
2. Robustness of agent navigation in both static and dynamic mazes. (10%)
3. Efficiency and adaptability of the Swarm Intelligence algorithm.(10%)
4. Clarity and functionality of the visualization. (10%)
5. Extensive documentation of the whole process. (20%)
6. Class Presentation with working Demo covering all requirements(40%)