Project Proposal: Adaptive Maze Battle Game

Group members:-

SAHIL	22K-4689
HADI	22K-4693
ALI	22K-4681

1. Project Title

Adaptive Maze Battle Game: A Dynamic AI-Powered Strategy Game with Real-Time Maze Alteration

2. Introduction

This project aims to develop a Python-based interactive game that combines strategic gameplay with artificial intelligence. The core idea is to build a battle game where players navigate and combat in a dynamically changing maze. The maze structure is intelligently modified in realtime using rule-based AI mechanisms, while an A* pathfinding algorithm enables efficient player movement. This project serves as an engaging platform to explore the fusion of adaptive AI and pathfinding in game design.

3. Objectives

- Implement an interactive 2D maze battle game with dynamic obstacles.
- Design and integrate an AI engine that modifies the maze layout in real-time based on player positions and actions.
- Apply A* pathfinding for player or enemy navigation through complex mazes.
- Develop multiple game modes (e.g., player vs AI, survival, etc.).
- Promote strategic decision-making through an evolving game environment.

4. Technologies Used

- **Programming Language**: Python
- **Game Logic and Rendering**: Pygame (or custom rendering)
- **Pathfinding Algorithm**: A* (implemented in `pathfinding.py`)

- **AI System**: Rule-based adaptive maze generator (in `ai_engine.py`)
- **Architecture**: Modular Python scripts

5. Key Modules

- `main.py`: Initializes and runs the game loop.
- `maze.py`: Manages the creation and transformation of the maze.
- `pathfinding.py`: Implements the A* algorithm for player and AI movement.
- `ai_engine.py`: Contains the logic for modifying the maze using rule-based AI.
- `player.py`: Represents the player entity with movement, state, and interaction methods.
- `game_modes.py`: Supports different styles of gameplay.
- `config.py`: Stores global constants and configuration settings.

6. Methodology

- **Step 1**: Design the static structure of the maze and player controls.
- **Step 2**: Implement A* pathfinding for navigation in static mazes.
- **Step 3**: Develop a rule-based AI to modify maze walls and paths dynamically.
- **Step 4**: Integrate adaptive AI with the game loop to trigger real-time maze updates.
- **Step 5**: Test various gameplay scenarios to ensure smooth interaction between AI, pathfinding, and user inputs.

7. Expected Outcomes

- A functional battle game with real-time strategic AI.
- Demonstration of adaptive gameplay environments using rule-based systems.
- Enhanced understanding of AI integration in game development.
- A modular, extensible codebase suitable for further enhancements or academic presentation.

8. Potential Extensions

- Integration of reinforcement learning to replace or enhance the rule-based AI.
- Online multiplayer support.
- Enhanced graphics and animations using advanced libraries.
- Difficulty scaling and player progression systems.

9. Conclusion

The Adaptive Maze Battle Game offers a novel approach to blending AI and game mechanics. It is a promising academic and practical project demonstrating problem-solving, game design, and AI adaptation in real-time environments.