Project Title: The Moving Maze Game

Submitted By: Danial Ahmed(22K-4580), Abdul Aleem(22K-4555), Ali Noorani(22K-4343)

Course: AI

Instructor: Miss Syeda Ravia Ejaz

Submission Date: 15th May 2025

1. Executive Summary

Project Overview:
The Moving Maze Game is an interactive and dynamic maze game developed using Python and Pygame. The primary objective is to navigate a constantly shifting maze and reach the goal before the AI opponent. The game features adaptive AI using a hybrid of A* pathfinding and Minimax algorithms to strategically navigate the maze. The game also includes various game modes, such as Classic, Time Trial, and Survival, providing diverse challenges to players.

2. Introduction

Background:
The Moving Maze Game builds on the traditional maze navigation concept, introducing a dynamic maze that shifts periodically to enhance difficulty. This game was selected to explore the integration of AI pathfinding with adaptive decision-making strategies in a dynamic environment.

Objectives of the Project:

- Develop a game where players navigate a dynamic maze to reach the goal.
- Implement AI that adapts to shifting maze configurations.
- Integrate hybrid AI using A* for pathfinding and Minimax for strategic moves.
- Enhance gameplay with different modes and visual aesthetics.

3. Game Description

Original Game Rules:
Traditionally, maze games involve navigating from a start point to an end point without changing paths. The challenge lies in finding the shortest route through a static maze.

Innovations and Modifications:

- Dynamic maze that shifts every few moves, keeping the game unpredictable.
- Multiple game modes: Classic, Time Trial, and Survival.
- AI opponent with adaptive pathfinding and decision-making abilities.

4. AI Approach and Methodology

AI Techniques Used:
The game's AI combines A* pathfinding with the Minimax algorithm. The AI first uses A* to find the shortest path to the goal. Minimax then evaluates potential moves based on factors like distance to the goal, proximity to the human player, and recent move history to avoid stalling.

Algorithm and Heuristic Design:

- A* is used for calculating the shortest path.
- Minimax enhances decision-making by considering opponent moves.
- Recent move penalties are applied to reduce stalling and looped paths.

AI Performance Evaluation:
The AI was tested under various game modes and difficulty levels. Performance metrics included win rate, decision-making speed, and adaptability to dynamic maze changes.

5. Game Mechanics and Rules

Modified Game Rules:

- The maze shifts dynamically every few moves.
- Players must navigate around obstacles and traps while collecting power-ups.
- The AI recalculates its path after each maze shift.

Turn-based Mechanics:

- The player uses arrow keys to move, while the AI moves automatically.
- The game alternates between human and AI turns, with a countdown at the start.

Winning Conditions:

- The first player to reach their designated goal (top left for the AI, bottom right for the human) wins the game.

6. Implementation and Development

Development Process:

- The game was developed using Python, leveraging the Pygame library for rendering and NumPy for maze manipulation.
- The AI logic, including A* and Minimax, was implemented from scratch, focusing on efficiency and adaptability.
- The UI enhancements include a countdown timer, dynamic window title, and a dark-themed background.

Programming Languages and Tools:

- Programming Language: Python
- Libraries: Pygame, NumPy
- Tools: Visual Studio Code (VS Code) for development,

Challenges Encountered:

- Ensuring maze solvability during generation was a significant challenge, addressed by integrating A* validation before game start.
- Avoiding AI stalling required the introduction of penalties for revisiting recently explored positions.

7. Team Contributions

Team Members and Responsibilities:

- Abdul Aleem: Responsible for AI algorithm development (A*, Minimax), game mechanics, and UI design.
- Ali Noorani: Assisted in game logic and testing.
- Danial Ahmed: Handled user interface improvements and dynamic maze implementation.

8. Results and Discussion

AI Performance:
The AI was tested across all game modes with varying difficulty levels. In the final version, the AI was observed to successfully adapt to dynamic maze changes and make strategic moves without stalling. Performance testing showed a high win rate in Classic mode and good adaptability in Survival mode.

9. References

1. Dynamic Maze & AI in Board Games:

• Silver, D., et al. (2016). "Mastering the Game of Go with Deep Neural Networks and Tree Search." Nature.

2. Pathfinding Algorithms:

• A* Algorithm: Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." IEEE Transactions on Systems Science and Cybernetics.

• Dijkstra's Algorithm: Dijkstra, E. W. (1959). "A Note on Two Problems in Connexion with

Graphs." Numerische Mathematik.