National University
Of Computer and Emerging Sciences

Project Title: Mind-Maze

Team Members: Ahmed Abdullah 22K-4449
Daniyal Ahmed 22K-4601

Course: Artificial Intelligence

Instructor: Miss. Alishba Subhani

Submission date: 15th April, 2025

1. Project Overview

- Project Topic:
  The project is based on an original 2-player strategy-based board game called Mind-Maze. It combines elements of memory, bluffing, and strategy. Players discover hidden tiles on a 6x6 grid which can also be increased for more longer game. There can be traps, scorers, advantages, or neutral tiles on the board. The game emphasizes incomplete information, forcing players and AI to make decisions based on partial knowledge and remembered positions.

- **Objective:**
  To develop an AI agent for the game using an enhanced version of the Minimax algorithm that handles partial observability and bluff prediction. The goal is to simulate human-like decision-making using memory-based heuristics and strategic evaluation of hidden risks.

2. Game Description

- **Original Game Background:**
  The game is an original concept with inspiration from Minesweeper and card game called bluff. It involves a dynamic tile-based board where information is gradually revealed as players discover tiles and try to remember or trick their opponent into thinking what the tile actually was. Combined rules of minesweeper and bluff encourages a player to think their strategy and also trick other players be bluffing about different tiles.

- **Innovations Introduced:**
  1. Players can bluff and hide their past moves by moving, which re-hides tiles so where a player memory is used to remember past tiles.
  2. AI must not only calculate possible outcomes but also simulate memory and opponent deception.
  3. Boosters allow players to peek hidden tiles or gain extra moves, adding non-linearity.

  Impact on Gameplay Complexity:

  1. Requires players to weigh risk vs reward on every turn.
  2. Increases unpredictability and long-term memory planning.
  3. Makes the game perfect for applying advanced AI strategies, particularly those that incorporate uncertainty handling and opponent modeling.

3. AI Approach and Methodology

- ### AI Techniques to be Used:

1. Minimax Algorithm with enhancements for partial observable environment.
2. Alpha-Beta Pruning to optimize decision tree exploration based on available tiles.
3. Use a simple, rule-based system that looks at the opponent's previous moves to guess if they might be bluffing for example, if they avoid good tiles on purpose or act strangely when revealing points. This helps the AI figure out if the opponent is trying to trick it or playing normally.

- ### Heuristic Design:

1. Score potential of revealed tiles: This means figuring out how valuable the tiles that can be seen are, based on their position and what they might lead to in the game.
2. Memory of known tile types and locations**:** Keeping track of which tiles you've already seen and where they are, so you can make smarter decisions.
3. Risk factor of unexplored tiles**:** Considering the danger of tiles you haven't seen yet, based on how they're placed and what might be hidden there.
4. Tracking opponent's last move**:** Watching what your opponent did last to try to guess if they are bluffing or trying to trick you.

- ### Complexity Analysis:

1. Time complexity increases due to unknowns and extra depth from memory-based logic.
2. There are numerous challenges to encounter like handling hidden information and integrating AI memory.
3. Estimated complexity: O(b^d) where branching factor (b) varies due to uncertainty.

4). Game Rules and Mechanics

- ### Modified Rules:

1. 6x6 board where each player places 3 scorers, 3 traps, 3 advantage tiles while rest of the tiles are neutral
2. Unrevealed tiles hide their functionality.
3. When a tile is flipped, players can either move to it or leave it.
4. Moving causes the previous tile to re-hide.
5. Boosters may allow bonus moves or tile peeking based on what booster the tile is of.

- **Winning Conditions:**

1. First player to reach 10 points, or player with the most points when all tiles are revealed.

- **Turn Sequences**

1). A person flips a tile and decides based on the value of the tile and his strategy whether he wants to take it or not.
2. If he moves to it last tile is hidded and this tile becomes visible to other player.
3. Now next player makes his turn and this sequence continues until someones point are 10 or game is drawn.

5. Implementation Plan

● Programming Language: Python

● Libraries and Tools:

1. Pygame for graphical board and interactions of tiles.
2. NumPy for board state tracking
3. Tkinter can be used so simplifying UI.
4. Custom Minimax implementation, possibly random or copy modules for simulations

● Milestones and Timeline:

1. Week 11–12: Game rule finalization, board setup, basic interactions
2. Week 12–13: Implement AI strategy. Minimax with memory simulation
3. Week 13–14: Coding advantages, traps, scorer effects
4. Week 14-15: AI integration with turn handling, testing strategy
5. Week 8: Final playtesting, improving code, and preparation of project report

6. References

1. Russell & Norvig – Artificial Intelligence: A Modern Approach
2. TutorialsPoint – Minimax & Alpha-Beta Pruning
3. Game theory articles on partial observability
4. Pygame documentation.