Project Report: Memory Maze Game

---

1. Project Overview

Project Topic:
Memory Maze is a strategic 2D maze game where players must navigate a grid, avoid traps, collect bonuses, and reach the goal. The unique twist is the inclusion of memory tokens, which players can use to temporarily reveal parts of the maze to help them avoid obstacles and plan their path more effectively.

Objective:
The main objective of this project is to develop a maze game that tests both the memory and strategy of players. The game uses a memory token system that allows players to reveal tiles temporarily, assisting them in making strategic moves. The project also includes an AI opponent that can compete with the player using advanced algorithms like Minimax.

---

2. Game Description

Original Game Background:
Memory Maze takes inspiration from classic maze navigation games but adds a memory element. In traditional maze games, players navigate from start to goal while avoiding obstacles. This game adds the complexity of traps, bonuses, and memory tokens. Memory tokens allow players to reveal surrounding tiles temporarily, giving them critical information to avoid traps and find the goal faster.

Innovations Introduced:

- **Memory Tokens**: Players have a limited number of tokens that can reveal surrounding tiles temporarily. This introduces a strategic element, as players must decide when to use their tokens and whether it's worth risking moving without full knowledge of the maze.
- **Traps**: These are randomly placed tiles that harm players when stepped on. If a player steps on three traps, they are eliminated from the game.
- **Bonuses**: These tiles provide advantages, such as additional memory tokens, allowing players to use their memory more frequently.

- **Game Progression**: The game progresses with players trying to reach the goal, but they must carefully manage memory tokens, avoid traps, and make decisions about their path forward.

These innovations increase the game's complexity by requiring players to use their memory, manage limited resources (memory tokens), and make strategic decisions about when and where to move.

---

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm**: This algorithm is central to the AI's decision-making. It will evaluate possible moves and choose the one that maximizes the AI's chances of winning while minimizing the human player's opportunities.
- **Alpha-Beta Pruning**: This optimization technique is used to reduce the number of nodes evaluated by the Minimax algorithm, improving performance without sacrificing optimality.
- **Reinforcement Learning (optional)**: If time permits, reinforcement learning may be introduced to allow the AI to improve its strategy through self-play.

Heuristic Design:

- **Proximity to the Goal**: The AI evaluates the current position relative to the goal. The closer it is to the goal, the higher the value.
- **Trap Avoidance**: The AI will prioritize avoiding traps by taking paths that have been revealed to be safe.
- **Memory Token Usage**: The AI will consider when it is beneficial to use memory tokens, potentially revealing a larger portion of the maze to plan its path.

Complexity Analysis:

- The Minimax algorithm has a time complexity of O(bd)O(b^d)O(bd), where bbb is the branching factor (number of moves available at each step) and ddd is the depth of the game tree (number of turns). Given the complexity of the maze and the presence of traps and bonuses, the branching factor can be large, requiring efficient pruning methods to keep the algorithm computationally feasible.

---

4. Game Rules and Mechanics

Modified Rules:

- **Memory Tokens**: Players start with 2 memory tokens. These tokens can be used to temporarily reveal surrounding tiles on the maze. Once used, the token count decreases by 1.
- **Traps**: If a player steps on a trap, they incur a penalty. After hitting 3 traps, the player is eliminated.
- **Goal**: The goal is to reach the center of the maze. The first player to reach the goal wins. If a player is eliminated due to hitting too many traps, the other player wins by default.

Winning Conditions:

- A player wins by reaching the goal at the center of the maze. However, a player can also lose if they hit 3 traps, which eliminates them from the game.

Turn Sequence:

- Players take turns moving in the grid, with each move involving either using a memory token to reveal surrounding tiles or moving in one of the four cardinal directions (up, down, left, or right). If a player lands on a trap, they are sent back to their checkpoint.
- The game alternates between players until a winner is determined.

---

5. Implementation Plan

Programming Language: Python

Libraries and Tools:

- **Pygame**: Used for rendering the graphical interface and handling user input. Pygame allows for easy creation of game windows, drawing of grid tiles, and handling player movement.
- **NumPy**: Used for data handling, especially for managing the grid and tracking tile states (whether a tile is revealed, a trap, or a bonus).
- **AI Algorithms**: Minimax and Alpha-Beta pruning will be implemented for the AI opponent's decision-making process.

Milestones and Timeline:

- **Week 1-2**: Finalize game design and rules. Implement the game board and basic mechanics such as player movement, trap placement, and goal setting.
- **Week 3-4**: Develop the AI strategy using Minimax and heuristics. Implement the game's AI opponent.
- **Week 5-6**: Integrate memory tokens and bonuses into the game, allowing players to interact with the board and use memory tokens.
- **Week 7**: Test AI behavior and integrate all features, ensuring smooth interaction between players and the AI.
- **Week 8**: Perform final testing, debug the game, and prepare the final report.

---

6. Visual Representation

The images provided above illustrate the current gameplay:

1. **First Image**: Shows the game where Player 1 wins with tokens and traps counted.
2. **Second Image**: Shows the state where Player 1 is one step away from winning but has encountered a trap.
3. **Third Image**: Displays Player 1's position near the goal and close to Player 2. The AI's behavior and traps make the game complex, enhancing strategic thinking.

The game's grid, player movements, traps, bonuses, and memory tokens are clearly represented in these screenshots.

Conclusion

The *Memory Maze* project successfully integrates the mechanics of a maze game with strategic elements involving memory and decision-making. By utilizing advanced algorithms like Minimax, the project aims to create a challenging environment for human players, as well as provide an AI that can compete with or against players in an engaging way. The implementation plan, milestones, and AI techniques outlined provide a structured approach to developing the game, ensuring that it meets both educational and entertainment objectives.