Project Title: **Connect 4 Game**

Submitted By:

Sania Ushaa 22k-4382

Eesha Fatima 22k-4226

Shermeen Ziauddin 22k-4206

Course: Artificial Intelligence

Instructor:Talha Shahid

Submission Date: 11-05-2025

1. Project Overview

- **Project Topic:**

  Our project implements ‘Connect 4’, a classic two player board game. We are enhancing it by introducing variations such as variable board sizes, obstacles (blocked cells), and time limited moves. These changes will increase the complexity and strategic depth of the game.

- **Objective:**

  Our goal is to develop an intelligent AI opponent using the **Minimax algorithm** (with Alpha-Beta Pruning), capable of playing Connect 4 with the introduced variations. The AI should adapt its strategy to different board sizes, avoid blocked cells, and make decisions efficiently within time constraints.

2. Game Description

- **Original Game Background:**

  Connect 4 is a two player board game where players take turns dropping discs into a vertical 6x7 grid. The objective is to form a horizontal, vertical, or diagonal line of four of one's own discs.

- **Innovations Introduced:**
  - **Variable Board Size**: The board size will be customizable (e.g., 5x5, 8x8), and the winning condition will adjust accordingly (e.g., 5-in-a-row on larger boards).
  - **Obstacles / Blocked Cells**: Random cells on the board will be blocked and cannot be played in, requiring smarter planning.

- **Time-Based Moves**: Players (including the AI) must make a move within a fixed time (e.g., 5 seconds), adding time pressure to the decision-making process.

These three variations together make the game more unpredictable and challenging for the AI. The AI must now adapt to the changing board state, predict future moves, and make decisions in real-time, while keeping track of the added constraints (blocked cells, time limits). As a result, the AI's decision-making will be more complex.

3. AI Approach and Methodology

- **AI Techniques to be Used:**
  - **Minimax Algorithm**: Used for optimal decision-making by evaluating future game states.
  - **Alpha-Beta Pruning**: Applied to improve the efficiency of Minimax by pruning unnecessary branches.

- **Heuristic Design:**

  Board evaluation will consider:

  - Number of potential connect-4 lines formed: The AI will evaluate how many winning lines it can complete in future moves thus prioritizing moves that increases its chances of winning
  - Blocks to opponent's threats: The AI assesses whether the opponent is close to winning and thus blocking these moves to prevent defeat.
  - Proximity to center columns to maximize winning opportunities: The AI value moves in the center columns more, as they offer greater potential for forming winning lines in multiple directions.
  - Avoidance of blocked cells: The AI avoids placing pieces in blocked cells, ensuring all moves are effective and the game state remains optimal.

- **Complexity Analysis:**
  - Time complexity of Minimax grows exponentially with depth (O(b^d)).
  - Introducing variable board sizes and blocked cells increases the branching factor.
  - Alpha-Beta pruning will help in reducing the effective number of nodes.
  - The timer constraint ensures bounded decision-making time.

4. Game Rules and Mechanics

- **Modified Rules:**
  - Players can choose board size before starting the game.
  - Certain cells are blocked and cannot be played in.
  - Players must make a move within a fixed time or lose their turn.

- **Winning Conditions:**
  - Form N tokens in a row (based on board size) horizontally, vertically, or diagonally.

- **Turn Sequence:**
  - Player 1 → Player 2 → Repeat.
  - If time expires, the turn is passed to another player.

5. Implementation Plan

- **Programming Language:** Python
- **Libraries and Tools:**
  - **Pygame**: For graphical user interface.
  - **NumPy**: For board data manipulation.
  - **Time module**: For implementing move timers.
- **Milestones and Timeline:**
  - **Week 1-2**: Game design and rule modifications.
  - **Week 3-4**: Implement Minimax and AI heuristics.
  - **Week 5-6**: Add variations (variable board, obstacles, timer).
  - **Week 7**: Integration and testing.
  - **Week 8**: Final review and report writing.

6. References

- SarowarAlam/Connect-4-Using-AI
- GeeksforGeeks (Minimax Algorithm)