PROJECT PROPOSAL

Project Title: Reverse Tic-Tac-Toe
Course: Artificial Intelligence
Instructor: Ms. Almas

Submitted By:

- Zehra Raza (22K-4197)
- Zoya Raza (22K-4194)
- Raheen Siddiqui (22k-4216)

Submitted On: 23th March 2025

Project Topic:

A variation of Tic-Tac-Toe where the goal is **to avoid making three in a row** instead of achieving it. Players take turns placing Xs and Os on a 3×3 board, but the first player to complete a line **loses** the game.

Objective:

Develop an AI that can strategically place pieces to **force the opponent into losing** while avoiding a losing move itself. The AI will use **Minimax with Alpha-Beta Pruning** for optimal strategy.

Game Description

Original Game Background:

Tic-Tac-Toe is a two-player game played on a 3x3 grid where players take turns marking Xs and Os. The first player to align three of their marks (horizontally, vertically, or diagonally) wins.

Innovations Introduced:

- **Objective Change:** Instead of aiming to win by forming a line, players now try to avoid forming one.
- **AI Strategy Complexity:** Unlike standard Tic-Tac-Toe, where forced wins are easy to compute, avoiding a loss requires deeper lookahead and opponent modeling.

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm:** Modified to minimize the chances of forming a three-in-a-row.

- **Alpha-Beta Pruning:** Optimizes the Minimax search by pruning unnecessary branches.
- **Reinforcement Learning (Optional):** AI can be trained via self-play to refine strategies over time.

Heuristic Design:

- Assign negative scores to potential three-in-a-row formations.
- Evaluate threats based on opponent's potential forced moves.
- Consider alternative board sizes where forced losses are harder to calculate.

Complexity Analysis:

- A standard 3x3 board results in **O(b^d) complexity**, where **b = branching factor** and **d = depth of search**.

4. Game Rules and Mechanics

Modified Rules:

- Players take turns placing marks (X or O) on a 3x3 grid.
- The first player to form three of their marks in a row (horizontally, vertically, or diagonally) loses.
- If the board fills up without a three-in-a-row, the game is a draw.

Winning Conditions:

- A player **loses** if they are forced to make a move that results in three marks in a row.
- If the board is filled and neither player has lost, the game is a draw.

Turn Sequence:

- Player 1 (X) moves first.
- Player 2 (O) moves next.
- Turns continue until a player forms a three-in-a-row and loses.

5. Implementation Plan

Programming Language:

- Python

Libraries and Tools:

- **Pygame:** For graphical user interface (GUI)
- **NumPy:** For efficient game state representation
- **Minimax Algorithm with Alpha-Beta Pruning** (Implemented in Python)

Milestones and Timeline:

- **Week 1-2:** Finalize game design, rules, and board variations.
- **Week 3-4:** Implement Minimax with Alpha-Beta Pruning.
- **Week 5-6:** Develop GUI and integrate AI.
- **Week 7:** Perform extensive testing and fine-tune AI heuristics.
- **Week 8:** Final testing and documentation preparation.

6. References

- Alpha-Beta Pruning