ARTIFICAL INTELLIGENCE PROJECT PROPOSAL

Project Title: Multiplayer Tic Tac Toe Game
Submitted By: Sharjeel Safdar (22K4210) Tulaib Touseef (22K4437)
Course: Artifical Intelligence
Instructor: Alina Arshad (Theory) Ayesha Almas Ansari (Lab)
Submission Date: 22/03/2025

1. Project Overview

Project Topic:

Developing an AI-powered multiplayer Tic-Tac-Toe variant for 2 to 5 players with an adaptive strategy.

Objective:

- Create an AI capable of playing Tic-Tac-Toe in various multiplayer settings using Minimax with multi-agent adaptation.
- Modify traditional rules to accommodate up to 5 players while maintaining balanced gameplay.
- Implement an interactive user-friendly interface using Streamlit.

2. Game Description

Original Game Background:

Tic-Tac-Toe is a simple turn-based strategy game where two players take turns placing their marks (X or O) on a 3x3 grid. The goal is to align three marks in a row, column, or diagonal.

Innovations Introduced:

- Expansion to support 2 to 5 players.
- Variable board sizes (e.g., 3x3 for 2 players, 5x5 for 3 players, and larger for 4-5 players).
- New winning conditions: Players must align a set number of marks depending on board size and player count.
- AI that adapts to multiple opponents rather than a single adversary.

3. AI Approach and Methodology

AI Techniques to be Used:

- **Minimax Algorithm:** Modified for multi-player settings, evaluating multiple potential moves per turn.
- **Alpha-Beta Pruning:** To optimize Minimax search and reduce computational overhead.
- **Reinforcement Learning (Optional):** Train AI to recognize strategic positions over time.
- **Monte Carlo Tree Search (MCTS) (Optional):** Enhance decision-making for higher board complexity.

Heuristic Design:

- Assign weights to game states based on potential winning paths, opponent blockages, and board control.
- Prioritize moves that create multiple winning opportunities simultaneously.

Complexity Analysis:

- The complexity increases with board size and player count (O(b^d), where b is the branching factor and d is depth).
- Multi-player scenarios introduce additional state evaluation challenges.

4. Game Rules and Mechanics

Modified Rules:

- Each player is assigned a unique symbol.
- The board size scales with player count (3x3 for 2 players, 5x5 for 3 players, etc.).
- Players win by aligning a set number of marks (3 in 3x3, 4 in 5x5, etc.).
- If all spaces are filled without a winner, the game ends in a draw.

Winning Conditions:

- A player wins by forming a straight line (row, column, diagonal) with the required marks.
- If multiple players achieve the condition simultaneously, the game declares a tie or follows priority rules.

Turn Sequence:

- Players take turns sequentially.
- AI opponents will analyze board states and predict best moves before playing.

5. Implementation Plan

Programming Language:

- **Python** (for AI logic and backend)

Libraries and Tools:

- **Streamlit** (for UI development)
- **NumPy** (for game board matrix manipulation)

- **Pygame** (optional, for alternative GUI implementation)
- **Scikit-learn / TensorFlow** (if ML-based AI is used)

Milestones and Timeline:

- **Week 1-2:** Finalize rules and game mechanics.
- **Week 3-4:** Implement AI strategies (Minimax, heuristics, optional ML-based approaches).
- **Week 5-6:** Develop game logic and test fundamental mechanics.
- **Week 7:** Integrate AI with game logic and UI.
- **Week 8:** Conduct testing, optimize performance, and finalize the project.

6. References

- Research papers on multi-player Minimax applications.
- Online resources for AI-based board game strategies.
- Python documentation for NumPy, Streamlit, and Pygame libraries.