Tic Tac Toe AI Using Minimax Algorithm - Project Proposal

Submitted By: Muhammad Hassan Qureshi & Syed Wajih Mustafa

Course: AI

Submission Date: May 11, 2025

# 1. Project Overview

● Project Topic:  
This project implements an AI-based Tic Tac Toe game where the computer plays optimally using the Minimax algorithm. The focus is on perfect play and adversarial game planning.

● Objective:  
To develop an AI agent capable of playing Tic Tac Toe without ever losing, using the recursive Minimax algorithm to explore all possible game states and choose the best moves.

# 2. Game Description

● Original Game Background:  
Tic Tac Toe is a classic 3x3 turn-based game played by two players who alternately mark spaces with X and O. The goal is to align three marks in a row, column, or diagonal.

● Innovations Introduced:  
- Integration of a deterministic AI using the Minimax algorithm.  
- No rule-based shortcuts; the AI explores all future states recursively.  
- Interactive gameplay in Python with a text-based interface (optional extension: GUI via Tkinter or Pygame).

# 3. AI Approach and Methodology

● AI Techniques to be Used:  
- Minimax algorithm  
- Recursive decision trees  
- Heuristics: Win = +1, Loss = -1, Draw = 0

● Heuristic Design:  
The AI assumes the opponent plays optimally. Each game state is evaluated with a utility function and the best path is chosen using recursive backtracking.

● Complexity Analysis:  
The number of game states is limited (only 9! permutations), so the Minimax approach is computationally feasible. The time complexity is O(b^d), where b = 9 and d = depth.

# 4. Game Rules and Mechanics

● Game Rules:  
- Player vs. AI gameplay.  
- Players take turns (X goes first).  
- AI calculates optimal move based on current board state.  
- Game ends when a player wins or the board is full.

● Winning Conditions:  
Three of the same marks in a row (horizontal, vertical, or diagonal).

● Turn Sequence:  
Each turn, either the user or AI makes a move. After every move, the game checks for a winner or a draw before proceeding.

# 5. Implementation Plan

● Programming Language:  
Python 3.x

● Libraries and Tools:  
- math  
- Google Colab or Jupyter  
- Optional: tkinter or pygame

● Milestones and Timeline:  
- Week 1: Implement basic game logic and board  
- Week 2: Integrate and test the Minimax algorithm  
- Week 3: Build interface for player vs. AI  
- Week 4: Record demo and finalize report

# 6. References

- Minimax Algorithm (GeeksforGeeks, Brilliant.org)  
- Python documentation  
- Stack Overflow and GitHub examples  
- “Artificial Intelligence: A Modern Approach” by Russell & Norvig