NATIONAL UNIVERSITY
of Computer & Emerging Sciences

Project Title: Ultimate Tic-tac-toe

Submitted By: Ghulam Hussain(22K-4280), Hamdan Vohra(22K-4318),

Hamza Hussain (22K-4317)

Course: AI

Instructor: Ms. Alishba Subhani

Submission Date: 11 May 2025

1. Executive Summary

Project Overview:
This project involved designing and developing an enhanced version of the traditional Tic-Tac-Toe game, named Ultimate Tic-Tac-Toe. In this version, the game board consists of a 3x3 grid of 9 smaller Tic-Tac-Toe boards A AI opponent was implemented using the Minimax algorithm with Alpha-Beta pruning. The goal was to challenge human players by integrating intelligent move evaluation heuristics and a user friendly graphical user interface for users.

2. Introduction

Background:
Tic-Tac-Toe is a two-player game that offers limited depth, making it ideal for introducing concepts in game AI. To increase complexity and provide a better AI testing ground, Ultimate Tic-Tac-Toe was chosen. This variation requires players to consider both local and global board states, enhancing strategic planning.

Objectives of the Project:

- To develop an Ultimate Tic-Tac-Toe game with interactive GUI.
- To implement an AI using the Minimax algorithm with Alpha-Beta pruning.
- To design heuristics that improve the AI’s strategic play.

- To evaluate the AI’s performance against human opponents.

3. Game Description

Original Game Rules:

- Tic-Tac-Toe is played on a 3x3 grid by two players (X and O), who alternate placing their marks. The first to form a line (row, column, or diagonal) of three wins.

Innovations and Modifications:

- The main board is a 3x3 grid of sub-boards.
- The location of a player's move dictates the sub-board their opponent must play in.
- Sub-board victories are recorded on the main board.
- Players win by claiming three sub-boards in a row (row, column, or diagonal) on the main board.

4. AI Approach and Methodology

AI Techniques Used:

- Minimax search algorithm.
- Alpha-Beta pruning to reduce unnecessary search paths.

Algorithm and Heuristic Design:

- The AI simulates game states up to a depth of 4.
- Heuristics include:
  - Sub-board win score: +20 for AI, -20 for human.
  - Two-in-a-row evaluation: +5 or -5 depending on chances of win/block.
  - Center cell control bonus: +2 or -2.
- The evaluation function guides decision-making in non-terminal states.

AI Performance Evaluation:

- Evaluated by playing multiple games against human players.
- Win rate: approx. 70% against beginners, approx. 50% against intermediate players.

5. Game Mechanics and Rules

Modified Game Rules:

- The active sub-board is determined by the last cell played by the opponent.
- If that board is full or won, the player may choose any available board.

Turn-based Mechanics:

- Turns alternate between human and AI.
- After every move, the system checks for a win in the sub-board and the main board.

Winning Conditions:

- Win by controlling 3 sub-boards in a line (row, column, diagonal).
- The game also ends in a draw if all sub-boards are completed without a main board win.

6. Implementation and Development

Development Process:

- Designed game rules and board structure.
- Implemented core game logic and AI decision engine.
- Developed GUI with Pygame for interactive play.
- Integrated evaluation function and game-over conditions.

Programming Languages and Tools:

- Programming Language: Python
- Libraries: Pygame, sys, copy
- Tools: GitHub for version control and backup

Challenges Encountered:

- Ensuring move validation with active sub-board logic.
- Avoiding excessive computation with deep Minimax trees.
- Handling draws conditions and visualizing AI's last move.

7. Team Contributions

Ghulam Hussain: Responsible for AI algorithm development (Minimax, Alpha-Beta Pruning).

Hamdan Vohra: Focused on implementing the user interface and

integrating AI with gameplay.

Hamza Hussain: Handled game rule modifications and board design. Conducted performance testing and evaluation of the AI's decisions.

8. Results and Discussion

AI Performance:

- The AI successfully blocked potential winning moves and created opportunities to win.
- The center control and two-in-a-row heuristic improved decision-making.
- The game offered a challenging experience to players and demonstrated intelligent AI behavior.
- Visualization of the AI's last move helped players understand its strategy.

9. References

- Pygame Documentation: https://www.pygame.org/docs/
- Minimax Algorithm: https://en.wikipedia.org/wiki/Minimax
- Alpha-Beta Pruning: https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning
- https://en.wikipedia.org/wiki/Ultimate_tic-tac-toe