NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

(KARACHI CAMPUS)

FAST School of Computing Spring 2025

A.I PROJECT PROPOSAL

MEMBERS:

- Mishkaat Yousuf (22K-4624)
- Ayesha Ansari (22K-4453)
- Jahantaab Kulsoom (22K-4214)

ADVISOR:

- Miss Almas Ayesha Ansari
- Miss Alina Arshad

Project Topic:

Reverse Ultimate Tic Tac Toe – AI-Powered Strategy Game with Inverted Win Conditions

1. Project Overview Project

Topic:

Reverse Ultimate Tic Tac Toe is a strategic twist on the classic Ultimate Tic Tac Toe, where players aim **not** to win traditional Tic Tac Toe boards but to **force the opponent** into completing them. The project integrates an AI opponent that uses **Minimax with Alpha-Beta Pruning** to make optimized decisions in real-time.

Objective:

The goal of this project is to develop a Python-based board game that challenges conventional gameplay mechanics by reversing the win condition. The AI will be designed to avoid winning mini-boards and instead strategize to force the opponent into victory—thereby losing the overall game.

---

2. Game Description

Original Game Background:

Ultimate Tic Tac Toe is played on a 9x9 grid made up of nine 3x3 Tic Tac Toe boards. The move a player makes in one mini-board determines where the opponent must play next. The objective is to win three mini-boards in a row, similar to regular Tic Tac Toe.

Innovations Introduced:

- **Reverse Win Condition:** Players *lose* by winning three mini-boards in a row, encouraging completely new strategic thinking.
- **Rule Inversion:** The game encourages defensive play and intentional misdirection.
- **AI Opponent:** The AI evaluates moves to avoid winning and manipulates the opponent into a forced victory.
- **Strategic Depth:** Decision-making involves multiple layers—preventing local wins while avoiding global loss.

These innovations introduce a unique challenge that subverts traditional game theory principles and rewards indirect play.

---

3. AI Approach and Methodology AI

AI Techniques to be Used:

- **Minimax Algorithm:** For simulating possible move sequences and outcomes.
- **Alpha-Beta Pruning:** To reduce the number of nodes evaluated by the Minimax algorithm.

- **Heuristic Evaluation:** Scoring board states based on the likelihood of *not* winning and pushing the opponent into traps.

Heuristic Design:

- **Mini-Board Danger Levels:** Assign high penalties to moves that bring the player closer to winning mini-boards.
- **Opponent Forcing Potential:** Reward moves that limit the opponent's safe options.
- **Global Pattern Analysis:** Evaluate risk of forming three-in-a-row mini-board wins.

Complexity Analysis:

- **Time Complexity:** O(b^d), where `b` is branching factor and `d` is depth.
- **Challenges:** Designing a heuristic that evaluates based on avoidance rather than progression.

---

4. Game Rules and Mechanics Rules:

- A player loses if they win three mini-boards in a row.
- Moves must still follow traditional Ultimate Tic Tac Toe constraints.
- The mini-board a player selects determines where the opponent must play next.

Winning Conditions:

A player **loses** if they complete three mini-boards in a row (horizontally, vertically, or diagonally). The game ends immediately upon this condition being met.

Turn Sequence:

- Players alternate turns, placing their mark (X or O) in a valid cell.
- The AI calculates an optimal move using Minimax and Alpha-Beta Pruning.
- The move chosen forces the opponent into an unfavorable position.

---

5. Implementation Plan

Programming Language: Python

Libraries and Tools:

- **Tkinter or Pygame:** For GUI rendering and user interaction.
- **Custom Game Logic Classes:** For move validation and game flow.
- NumPy: For optimized calculations in AI processing.

Milestones and Timeline:

- Week 1-2: Define reversed rules, board structure, and move mechanics.
- Week 3-4: Implement Minimax with Alpha-Beta Pruning for the AI.
- Week 5-6: Build and test the GUI; integrate the AI with the game.
- Week 7: Conduct playtesting and refine AI strategies.
- Week 8: Finalize documentation, polish GUI, and submit the project.

---

6. References

- Academic papers on Minimax and Alpha-Beta Pruning
- Research on non-standard Tic Tac Toe variants
- Python documentation and tutorials on GUI development using Pygame or Tkinter

---