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

---

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

KARACHI CAMPUS

FAST School of Computing – Spring 2025

PROJECT REPORT

Team Members:

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

Advisors:

- Miss Almas Ayesha Ansari
- Miss Alina Arshad

---

1. Introduction

This report presents the design and implementation of **Reverse Ultimate Tic Tac Toe**, a novel Python-based board game that inverts the traditional objective of Ultimate Tic Tac Toe. Instead of winning by aligning three mini-board victories in a row, players must **avoid** such alignments. The core innovation lies in the use of an **AI opponent powered by the Minimax algorithm with Alpha-Beta Pruning**, which strategically forces opponents into "winning" positions, thereby causing their loss. The game introduces an unconventional way of thinking about adversarial strategy and explores AI in a non-traditional game theory context.

---

2. Game Overview

2.1 Original Concept: Ultimate Tic Tac Toe

Ultimate Tic Tac Toe is played on a 9x9 grid comprising nine 3x3 mini-boards. A player's move determines the mini-board where the opponent must play next. The goal is to win three mini-boards in a row.

2.2 Reverse Concept: Strategic Inversion

In our version:

- **Objective:** Avoid winning three mini-boards in a line (horizontally, vertically, or diagonally).
- **Strategy:** Focus on misleading the opponent into completing a winning combination.
- **Innovation:** Encourages reverse logic and deceptive gameplay.

This inversion introduces new dimensions of planning, as players must now avoid dominating the game, often opting for suboptimal-looking moves.

---

3. AI Design and Methodology

3.1 Core Algorithms

- **Minimax Algorithm:** Explores all possible move sequences to choose an optimal path, assuming perfect play.
- **Alpha-Beta Pruning:** Speeds up Minimax by eliminating branches that do not influence the final decision.

3.2 Heuristic Evaluation

Custom heuristics evaluate board states from a "loss-avoidance" perspective:

- **Danger Level Mapping:** Assigns penalties to moves that bring the player closer to three mini-board wins.
- **Opponent Forcing Index:** Rewards moves that limit the opponent's safe options.
- **Pattern Detection:** Identifies risky formations that may lead to accidental global victories.

3.3 Complexity and Optimization

- **Time Complexity:** $O(b^d)$, where b is the branching factor and d is the depth of the game tree.
- To maintain performance, the depth is limited and heuristic evaluations are weighted and tuned through playtesting.

---

4. Game Rules and Mechanics

4.1 Core Rules

- A player **loses** if they win **three mini-boards in a row**.
- The opponent's next move is constrained by the location of the current move, as per traditional Ultimate Tic Tac Toe rules.
- If a mini-board is already won or filled, the player may choose any available mini-board.

4.2 Turn Sequence

1. Human or AI makes a move within the valid mini-board.
2. The selected cell dictates the mini-board for the opponent's next move.
3. The system checks for global loss (3-in-a-row mini-board victories).
4. If no global loss is detected, the game proceeds.

---

5. Implementation Details

5.1 Tools and Technologies

- **Language:** Python
- **GUI Libraries:** Tkinter or Pygame
- **Libraries Used:** NumPy for board manipulation and efficient calculations

5.2 Development Phases

Weeks	Tasks
1–2	Defined rules, created board logic, and designed game flow
3–4	Implemented Minimax and Alpha-Beta Pruning
5–6	Designed and integrated GUI with core game logic
7	Conducted playtesting and refined AI evaluation functions
8	Finalized codebase, completed documentation, and submitted project

---

6. Results and Evaluation

6.1 AI Behavior

The AI consistently demonstrated intelligent avoidance of direct wins and effective manipulation of the opponent into making forced mistakes. Its ability to recognize potential traps and indirect threats allowed it to win against both random and semi-strategic players.

6.2 Playtesting Outcomes

- AI won ~70% of test games against human players.
- Identified effective patterns for creating deceptive board states.
- Reinforced the value of reverse thinking in adversarial games.

---

7. Conclusion

Reverse Ultimate Tic Tac Toe transforms a classic game into a novel strategic experience by flipping the traditional objective. This required careful re-engineering of AI decision-making to prioritize **loss-avoidance** over direct success. The project not only demonstrated the adaptability of adversarial search algorithms but also provided a playful yet insightful application of reverse game theory.

Future work could involve reinforcement learning models, dynamic difficulty adjustment, or multiplayer online support to extend the gameplay and learning experience.

---

8. References

- Academic articles on Minimax and Alpha-Beta Pruning
- Research on reverse game strategies and adversarial search
- Python GUI development resources (Tkinter and Pygame documentation)
- Online communities and repositories of Ultimate Tic Tac Toe implementati