AI PROJECT REPORT

Topic: Triple Threat: Tic Tac Toe
Sections: BCS-6F
Members: 22K - 4585 SAMRA SHAHID
Instructors: Alina Arshad & Syeda Ravia Ijaz

1. Project Synopsis

● Concept Overview:

Triple Threat: Tic Tac Toe brings a modern twist to the timeless 3x3 Tic Tac Toe game. By integrating an AI opponent that uses the Minimax algorithm enhanced with Alpha-Beta pruning, this project transforms a simple game into a challenging strategic battle. The modified gameplay introduces layered rules and added complexity, pushing players to think ahead while testing the AI's decision-making using a custom heuristic evaluation.

2. Project Foundation

● Game Origins:

The original Tic Tac Toe is a straightforward game played on a 3x3 grid where two players alternate marking X and O symbols in an attempt to align three in a row. Although fun, it lacks depth once mastered. To remedy this, **Triple Threat** uses an extended variant where each square of the main board is itself a mini Tic Tac Toe game, arranged in a 3x3 format. This adds a layer of strategy and unpredictability, providing an excellent platform to implement and test game-playing AI.

● Project Goals:

1. Create a competitive AI using the Minimax algorithm optimized with Alpha-Beta pruning.
2. Code the game logic, rules, and interface using Python.
3. Build and refine a heuristic function for evaluating game states and guiding AI behavior.
4. Enable human vs AI gameplay for demonstration and analysis.

3. Gameplay Structure

● Basic Mechanics:

In traditional Tic Tac Toe, players alternate moves aiming to form a line of three marks. A draw is declared if all spaces are filled without a winner.

● Enhancements and Changes:

- The main board is made up of 9 individual 3x3 mini boards.
- Winning a mini board claims the corresponding space on the main board.
- Winning three mini boards in a row secures overall victory.
- AI evaluation prioritizes board control, near-win situations, and center dominance.
- Capturing a mini board allows the player to take another consecutive turn.

4. Artificial Intelligence Strategy

● AI Method Employed:

The AI is powered by the Minimax algorithm, bolstered with Alpha-Beta pruning to optimize search efficiency by cutting out non-impactful move branches.

● Heuristic Development:

The custom evaluation function is fine-tuned using the following key criteria:

- **Mini Board Ownership**: +100/-100 points for securing a mini board.
- **Two-in-a-Row Patterns**: +10/-10 points for forming potential winning rows.
- **Main Board Threats**: +300/-300 points for almost-winning positions on the macro level.
- **Central Control**: Bonus points for claiming center cells in both mini and main boards.
  These factors are summed to evaluate and select optimal AI moves during gameplay.

5. Gameplay Dynamics and Rule Modifications

● Board Configuration:

- The complete board consists of 9 mini Tic Tac Toe boards arranged in a 3x3 grid.
- Capturing a mini board allows for an extra move.
- A drawn mini board becomes neutral and unclaimable.
- Victory is achieved by aligning three claimed mini boards.

● Turn Progression:

- Turns alternate unless a mini board is won, which grants a bonus move.
- The first player is chosen at random.
- Moves are generally restricted to the mini board corresponding to the opponent's last move—unless that board is already won or drawn.

● Victory and Draw Scenarios:

- A player wins by aligning three mini boards (row, column, or diagonal).
- If all mini boards are complete without such alignment, the game is a draw.

6. Technical Implementation

● Development Approach:

The game was built in Python. The logic for game states was established first, followed by the AI mechanics. The heuristic function underwent several refinements for better AI performance. For the user interface, libraries such as tkinter or pygame were used to make gameplay visually interactive.