Notre Dame University Bangladesh

Department of Computer Science and Engineering

Subject: Artificial Intelligence and Expert System

Code: Lab CSE 4104

<u>Team Members</u>

Sumaiya Islam

Id: 0692210005101021

Md.Fatin Ishrak Mahi

Id:0692210005101012

<u>Date of Submission</u>

26/05/2025

Lab Report: AI-Based Tic Tac Toe Game

Abstract- This report presents the development and implementation of an artificial intelligence (AI) system to play the Tic Tac Toe game. The project employs the Minimax algorithm to enable optimal decision-making by the AI, ensuring a competitive and challenging gameplay experience. The report outlines the motivation, methodology, and features of the implemented AI system and compares it with existing Tic Tac Toe AI models. Additionally, the report discusses potential improvements and broader applications of the Minimax algorithm in other AI-driven strategy games. The study further explores the theoretical background of decision trees and evaluates alternative AI techniques that could enhance performance and efficiency.

1. Introduction Tic Tac Toe is a well-known two-player strategy game played on a 3x3 grid, where players alternate turns marking X or O. The goal is to align three symbols in a row, column, or diagonal. While the game is simple in design, it serves as an excellent testbed for artificial intelligence techniques. The project aims to develop an intelligent AI system that utilizes game theory principles, particularly the Minimax algorithm, to play optimally against human opponents[1].

The significance of this project lies in its ability to demonstrate fundamental AI concepts, such as decision trees, heuristics, and recursion, in a manageable and easily interpretable format. Tic Tac Toe is an ideal choice because of its deterministic nature and limited state space, making it suitable for illustrating AI methodologies without excessive computational complexity. Furthermore, implementing AI in a familiar game environment allows for an intuitive understanding of computational decision-making processes and their real-world applications.

2. Objective

- To develop an AI system capable of playing Tic Tac Toe efficiently with optimal strategies.
- To implement the Minimax algorithm to enable AI-driven decision-making.
- Demonstrate the application of AI principles in game-playing scenarios.

3. Motivation : The motivation for this project stems from the increasing role of AI in strategic decision-making and game theory. Tic Tac Toe provides a controlled environment where fundamental AI principles can be applied effectively. The Minimax algorithm, which is widely used in board games and real-world applications such as robotics, economic modeling, and cybersecurity, is demonstrated in this simple yet powerful project.

4. Related Work:

The game of Tic Tac Toe has long been used as a foundational example in artificial intelligence (AI) and machine learning due to its simplicity and well-defined rules. Many previous projects and studies have explored various algorithms to simulate intelligent gameplay. Traditional approaches use minimax algorithms, often enhanced with alpha-beta pruning, to evaluate optimal moves by simulating all possible future game states.

More recent implementations have incorporated reinforcement learning methods, such as Q-learning, where the AI learns optimal strategies through trial-and-error interactions. Projects like DeepMind's AlphaZero also demonstrate how advanced techniques, including deep neural networks and Monte Carlo Tree Search (MCTS), can be applied to more complex games but have been scaled down effectively for simpler games like Tic Tac Toe.

Other studies have looked into AI player behavior based on human mimicry using supervised learning, where models are trained on datasets of human-played games to predict the best move.

Our work builds on these foundational techniques by selecting appropriate AI strategies suited to the game's simplicity while ensuring that the AI demonstrates logical and intelligent decision-making.

Methodology:

The methodology for creating the Tic-Tac-Toe game with AI involves:

1. Setting up the game board and player assignments.
2. Implementing the Minimax algorithm for decision-making.
3. Checking for terminal states after each move.
4. Adding optional features like difficulty levels and optimizing with Alpha-Beta pruning.

By following this methodology, we ensure that the AI plays optimally and the game provides a seamless experience for the player.

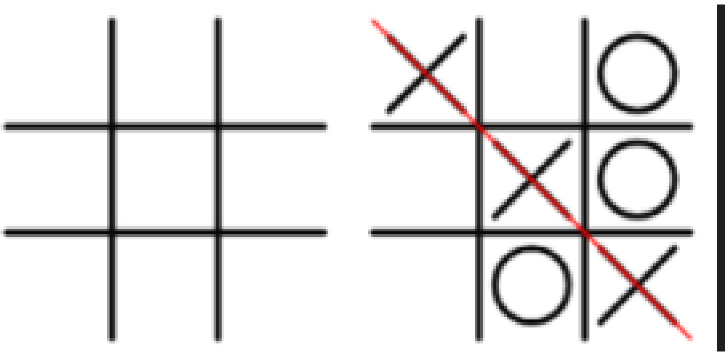

Fig 1: Tic Tac Toe Game representation

4.1 Game Components

- **Board Representation**: The Tic-Tac-Toe board will be represented as a 3x3 grid.
  - A list of lists (2D array) or a 1D list with 9 elements is commonly used to represent the board.
- **Player vs AI**: The player will play as 'X', and the AI will play as 'O'. ● **Turn Logic**: The game alternates between the player and the AI. The AI makes its move based on an evaluation function (e.g., Minimax).

4.2 Game Logic

- **Displaying the Board**: The board will be displayed after every move so the player can visualize the game state.
- **Winning Condition**: Checking will be done for winning conditions after each move. There are 8 possible ways to win (3 horizontal, 3 vertical, and 2 diagonal).
- **Draw Condition**: If all cells are filled and no one wins, the game ends in a draw.

4.3 AI Implementation (Minimax Algorithm)

The AI’s decision-making will be done using the **Minimax Algorithm**. Here’s how it works:

- **Minimax**: This is a backtracking algorithm used for decision-making in two-player games. In the Minimax algorithm, the two players are referred to as the Maximizer and the Minimizer. The Maximizer aims to achieve the highest possible score, while the Minimizer attempts to minimize the score, effectively trying to secure the lowest possible value.

  Each board state is assigned a numerical value that reflects the advantage of one player over the other.

❖ If the Maximizer has the upper hand, the score will lean toward a positive value.

❖ If the Minimizer is in a stronger position, the score will tend toward a negative value.

- **Steps**:
  1. Evaluating all possible game states.
  2. Assigning a score to each game state.
  3. Choosing the move that maximizes the AI's score while minimizing the player's possible score.
  4. Lastly,the AI will "think" ahead by simulating future moves.

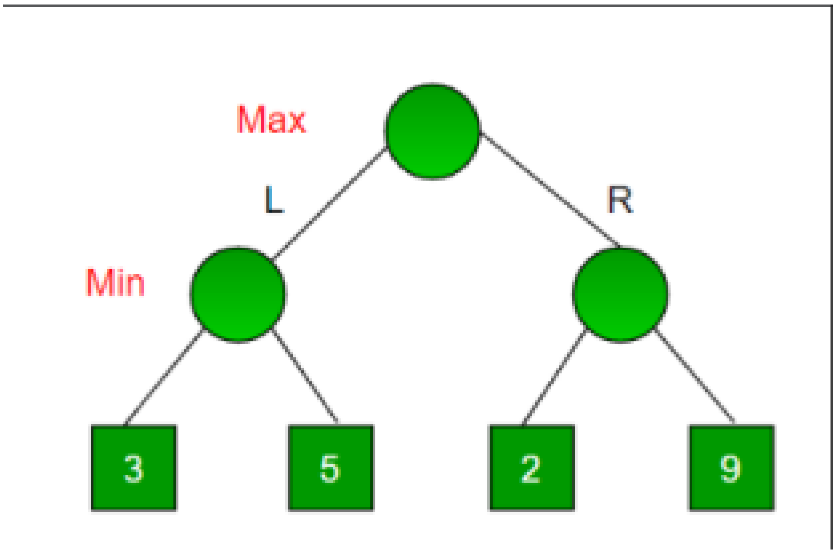

Fig:2 - Min Max algorithm representation.

5. Workflow

Problem Definition:

Objective: Create an AI opponent capable of playing unbeatable Tic Tac Toe.

Scope: Two-player game with AI acting as one player.

Design and Planning:

- Game board structure using a 3x3 matrix.
- Rules of the game are implemented using conditional logic.
- Selection of the AI algorithm (Minimax).

Implementation:

- **Game logic:** Developed core mechanics for player turns, checking wins/draws, and board updates.
- **AI algorithm:** Implemented the minimax function with depth-first recursion to simulate all possible moves.
- **User Interface:** Built a simple UI (text-based or graphical) for interaction.

Testing and Evaluation:

- Played multiple rounds to ensure AI behaves correctly.
- Verified that the AI never loses, even against optimal human moves.
- Adjusted performance and response time.

Result and Finalization:

- Successfully implemented an AI that always plays optimally.
- Verified that the project meets AI criteria through intelligent move prediction and adaptability.

Why Our Project is AI

- **Simulates Intelligent Behavior:**
  The system makes automated decisions that reflect intelligent gameplay without human intervention.

- **Game State Evaluation:**
  The AI assesses the current board configuration to determine available moves and potential threats.

- **Optimal Move Selection:**
  Based on its analysis, the AI selects the best possible move to win or prevent the opponent from winning.

- **Predictive Capability:**
  Through algorithms like **minimax**, the AI anticipates future moves by simulating possible outcomes several steps ahead.

- **Human-like Reasoning:**
  The AI mimics how a human would:

    - Analyze the game situation

    - Predict the opponent's next move

    - Choose the most strategic response

- **Use of AI Concepts:**
  The implementation involves core AI techniques such as:
  - **Search trees** to explore move sequences
  - **Heuristic evaluation** to score positions
  - **Decision-making under constraints** (limited moves, opponent behavior)
- **Not Just Rule-Based Logic:**
  Unlike simple if-else logic, this AI goes deeper into strategy, planning, and counterplay.
- **Dynamic & Adversarial Environment:**
  The system operates in a changing environment with an opponent actively trying to win—mirroring real-world decision-making scenarios.
- **No Learning Required:**
  Even without learning (e.g., reinforcement learning), the system qualifies as AI because it replicates intelligent decision-making.

Application :

Applications of the AI-based Tic Tac Toe model extend beyond gaming. The decision-making processes employed in this project can be adapted for AI-driven decision support systems, automated problem-solving, and strategic forecasting in business and military applications. By understanding how AI evaluates potential moves and outcomes, we can apply similar logic to financial markets, medical diagnostics, and automated negotiation systems[2].

6. Comparison with Existing Models :

Traditional Tic Tac Toe AI implementations range from basic rule-based strategies to more sophisticated approaches such as Minimax with Alpha-Beta pruning. Rule-based approaches rely on predefined heuristics to make decisions, whereas the Minimax approach examines all possible game states to determine the best move.

Compared to simple rule-based AI models, the Minimax-based AI in this project ensures optimal move selection, making it a formidable opponent. In contrast to more advanced AI techniques like reinforcement learning, the Minimax approach does not require training data and performs well in fully deterministic environments like Tic Tac Toe. Furthermore, alternative AI strategies such as Monte Carlo Tree Search (MCTS) offer a probabilistic approach to decision-making, providing potential areas for future exploration.

Here's a short comparison table for creating an AI Tic Tac Toe game using Python with the Minimax algorithm and comparing it with the Alpha-Beta pruning enhancement[3][4]:

Aspect	Minimax Algorithm	Alpha-Beta Pruning
Performance	Computationally intensive, explores entire game tree	More efficient, prunes unnecessary branches
Optimality	Guarantees optimal moves	Guarantees optimal moves
Efficiency	Less efficient, slower for larger game trees	Faster decision-making, fewer nodes evaluated
Complexity	Easier to implement and understand	Slightly more complex due to pruning logic
Resource Usage	Higher resource usage	Lower resource usage

Both Minimax and Alpha-Beta Pruning ensure optimal move selection, but Minimax stands out for its simplicity and educational value. It is easier to implement and understand, making it an excellent choice for beginners and for a project focused on foundational AI concepts. In the context of a simple game like Tic Tac Toe, the performance gains from Alpha-Beta Pruning may not justify the additional complexity, making Minimax a better fit for this project.

7. Challenges and Limitations

Despite its strengths, the project has inherent limitations:

- **Computational Cost:** Without Alpha-Beta pruning, Minimax can be computationally expensive as the game tree expands.
- **Predictability:** Since Tic Tac Toe has a finite number of states, a Minimax AI will always play optimally, making games against an experienced human player predictable.
- **Limited Scalability:** While effective for Tic Tac Toe, project AI alone is not suitable for more complex games with vast state spaces, such as Chess or Go, without significant optimizations.
- **Alternative Strategies:** Project assumes rational play; however, real-world applications often involve human-like randomness, which is not well accounted for in deterministic models.

8. Future Improvements

- Adaptive AI Strategies: Implementing learning-based AI models to adapt to different playing styles.
- Reinforcement Learning: Exploring AI models that can improve performance through self-play and experience.
- Online Multiplayer Integration: Allowing real-time gameplay between human players and AI opponents over the internet.
- Voice and Gesture Controls: Enhancing user experience by integrating AI-driven voice or gesture-based input methods.

9. Conclusion

The AI-based Tic Tac Toe game successfully demonstrates the application of the Minimax algorithm in a low-complexity setting. The system ensures optimal gameplay and highlights fundamental AI principles such as decision trees, heuristics, and recursion. Future enhancements could include the integration of Alpha-Beta pruning for efficiency, adaptive difficulty levels, and exploration of machine learning techniques to create a more dynamic opponent. Additionally, testing alternative AI models, such as Monte Carlo Tree Search, could provide insights into improving performance and adaptability in other strategic games. The insights gained from this project can be extended to more complex decision-making scenarios, reinforcing the importance of AI in game theory, computational strategy, and interactive applications.

10. References

1. *GitHub Repository: Cledersonbc/tic-tac-toe-minimax [Accessed on 31 January 2025].*
2. *Wright, P., & Atkyns, R. (1986). Game Theory and AI Strategies. [Accessed on 31 January 2025].*
3. *McPhail, D. (1971). Minimax and its Applications in Zero-Sum Games. [Accessed on 31 January 2025].*
4. *Knuth, D. E. (1975). Alpha-Beta Pruning and Search Trees. [Accessed on 31 January 2025.]*