

# Tic-Tac-Toe AI Implementation Report

Mehmet Ali GÜNEŞ

May 21, 2024

## 1 Abstarct

This study focuses on the development and implementation of a Tic Tac Toe game integrated with an Artificial Intelligence (AI) module using the Minimax algorithm. The game, developed using Python and the Pygame library, aims to provide an engaging experience for users by allowing them to compete against an AI that employs strategic decision-making. The main objective of this project is to explore the effectiveness of the Minimax algorithm in a classic game setting and to demonstrate the integration of AI in a simple yet challenging game environment.

**Keywords:** Artificial Intelligence (AI), Minimax Algorithm, Pygame, Game DevelopmentAI Opponent, Recursive Algorithm, User Interface (UI)

## 2 Technologies, Libraries and Methods

### 2.1 Pygame Library

Pygame is an open-source library designed for making video games. It provides functionalities to create game graphics, handle events, and manage game loops. In this project, Pygame is used to develop the user interface, handle user inputs, and render game elements such as the board and the player markers.

### 2.2 Minimax Algorithm

The Minimax algorithm is a recursive decision-making algorithm used in two-player games. It provides an optimal move for the player assuming that the opponent also plays optimally. The algorithm evaluates all possible moves and selects the one that maximizes the player's minimum gain. This project utilizes the Minimax algorithm with a depth parameter to control the difficulty level of the AI.

#### 2.2.1 Key Concepts:

1. Maximizing Player: The player who tries to maximize their score (often called "Max").
2. Minimizing Player: The opponent who tries to minimize the maximizing player's score (often called "Min").
3. Game Tree: A tree structure where each node represents a game state, and edges represent moves.
4. Depth: The level of the tree. Each level alternates between the maximizing player and the minimizing player.
5. Terminal State: A game state where the game ends (win, lose, or draw).

### 2.2.2 Algorithm Steps:

1. Generate the Game Tree: Start from the current board state and recursively generate all possible future moves.
2. Evaluate Terminal States: Assign scores to terminal states (win, lose, draw).
3. Backpropagate Scores: Move back up the tree, choosing the move that maximizes the score for the maximizing player and minimizes the score for the minimizing player.
4. Choose the Optimal Move: The root of the tree will contain the optimal move for the current player.

## 2.3 NumPy Library

NumPy (Numerical Python) is a fundamental package for scientific computing in Python. It provides support for arrays, matrices, and many mathematical functions to operate on these data structures efficiently.

Array Management: In my code, NumPy is used to create and manage the 2D array representing the Tic Tac Toe game board. This allows for efficient manipulation and checking of the board's state.

```
self.squares = np.zeros((ROWS, COLS))
```

Figure 1: This line initializes the game board as a 3x3 array filled with zeros, representing an empty board.

## 2.4 Copy Module

The copy module provides functions to create shallow and deep copies of objects in Python. A shallow copy creates a new object but inserts references into it to the objects found in the original, while a deep copy creates a new object and recursively adds copies of nested objects found in the original.

Deep Copy: In my code, The `copy.deepcopy` function is used to create independent copies of the game board. This is crucial for the Minimax algorithm to evaluate different possible game states without altering the actual game board.

```
temp_board = copy.deepcopy(board)
```

Figure 2: This line creates a deep copy of the current board state, allowing the AI to simulate moves and evaluate their outcomes without modifying the real game board.

## 2.5 Sys Module

The `sys` module provides access to some variables used or maintained by the Python interpreter and to functions that strongly interact with the interpreter. It includes functions for system-specific parameters and functions.

System Exit: In my code, The `sys.exit()` function is used to exit the game cleanly when the user decides to close the game window.

### **3 Innovations in the Tic Tac Toe AI Game**

The developed Tic Tac Toe AI game introduces several innovations and enhancements compared to a standard Tic Tac Toe game. These innovations aim to provide a more engaging, challenging, and enjoyable experience for players. Below are the key improvements:

#### **3.1 Advanced AI Using Minimax Algorithm**

##### **Description:**

- The game incorporates an AI opponent that uses the Minimax algorithm, which is a significant upgrade over simple random move selection.
- The Minimax algorithm ensures that the AI plays optimally, making it a challenging opponent for players of all skill levels.

##### **Benefits:**

- Provides a competitive and strategic gameplay experience.
- Helps players improve their skills by playing against a consistently strong opponent.

#### **3.2 Adjustable Difficulty Levels**

##### **Description:**

- The AI can operate at different difficulty levels. At lower levels, the AI may make random moves, while at higher levels, it uses the Minimax algorithm with a specified depth to determine the best moves.

##### **Benefits:**

- Allows players to choose a difficulty level that matches their skill.
- Makes the game accessible and enjoyable for both beginners and experienced players.

#### **3.3 Enhanced Game Graphics and User Interface**

##### **Description:**

- The game features a visually appealing user interface created using the Pygame library.
- Custom images for player markers (tridents and halos) and a background image enhance the visual experience.

##### **Benefits:**

- Provides a more immersive and engaging game environment.
- Improves the overall aesthetic appeal of the game.

### **3.4 Move History and Undo Functionality**

#### **Description:**

- The game keeps track of the move history, allowing players to see the sequence of moves.
- Includes an undo functionality that can revert moves, providing more flexibility during gameplay.

#### **Benefits:**

- Enables players to learn from their mistakes by reviewing past moves.
- Allows players to correct accidental moves, making the game more user-friendly.

### **3.5 Draw Condition Handling**

#### **Description:**

- The game includes a sophisticated mechanism to detect draw conditions, ensuring that the game recognizes when a draw occurs after a significant number of moves.

#### **Benefits:**

- Accurately reflects the outcome of the game, providing fair and clear results.
- Enhances the game's logic to handle more complex game states.

### **3.6 Dynamic Game Reset**

#### **Description:**

- Players can reset the game at any time using a designated key, allowing for quick restarts and new game sessions.

#### **Benefits:**

- Provides convenience and ease of use for players wanting to start over or play multiple rounds.
- Ensures a seamless and uninterrupted gaming experience.

### **3.7 Automatic Move Deletion After Seven Moves**

#### **Description:**

- After a total of seven moves, the game automatically deletes the oldest move to keep the game state manageable and to introduce a new level of strategy. Players need to consider the potential for their earlier moves to be removed from the board as they plan their next moves.

#### **Benefits:**

- Introduces a dynamic and evolving game state that requires continuous strategic thinking.
- Prevents the game board from becoming overly cluttered and ensures ongoing engagement.

These innovations collectively enhance the traditional Tic Tac Toe game by introducing advanced AI, improved graphics, user-friendly features, and robust game mechanics. The result is a modern and engaging version of Tic Tac Toe that appeals to a wide range of players.

## 4 GitHub Integration

You can find the complete source code for the Tic Tac Toe AI game on GitHub. The repository includes all the necessary files and instructions to run the game, as well as detailed documentation on the project's features and development process. Visit the following link to access the repository:

**GitHub Repository:** <https://github.com/aanomali/TICTACTOEWITHAI.git>

Feel free to clone the repository, explore the code, and contribute to the project. Whether you're interested in adding new features, fixing bugs, or simply playing the game, your contributions are welcome. Join the community and help improve this exciting AI-enhanced version of the classic Tic Tac Toe game!

## 5 User interface



Figure 3: **Crossed Tridents Image:** This image represents the 'X' character in the game, designed as crossed devil tridents to give a thematic feel.



Figure 4: **Halo Image:**This image represents the 'O' character in the game, designed as a halo to match the angelic theme.

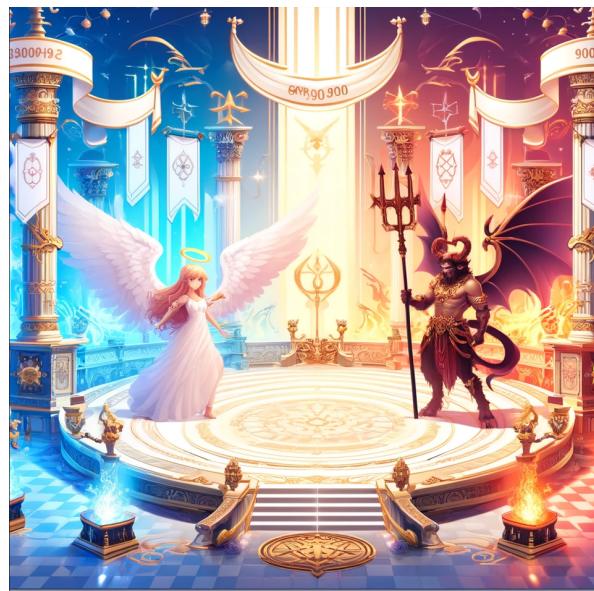


Figure 5: **Game Background Image:**This image is the thematic background for the Tic Tac Toe game, setting the scene for the angel versus devil showdown.



Figure 6: **Victorious Angel Image:** This image shows an angel victorious over a devil, symbolizing the AI's win in the game.

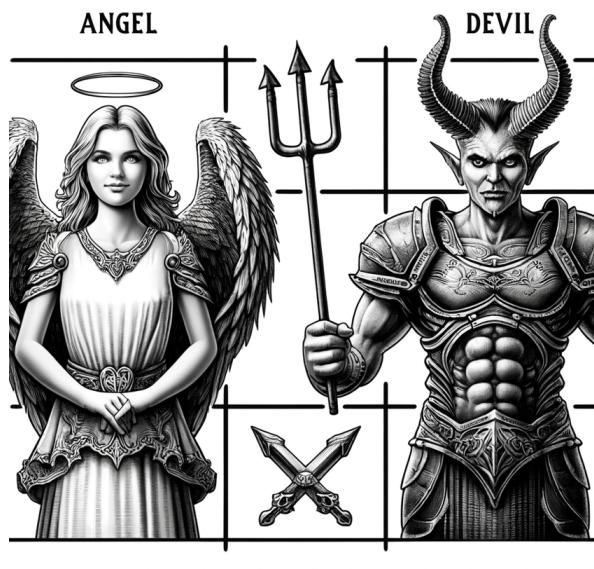


Figure 7: **Draw Screen Image:** This image represents the draw screen where neither the player nor the AI wins, showing an angel and a devil standing side by side.

## References

1. <https://pyga.me/>
2. <https://numpy.org/>
3. <https://www.pygame.org/docs/tut/MakeGames.html>