TIC-TAC-TOE AI

Mini Project Report

Name: SHAZIA TABASSUM
Branch: AIML
College: Kakatiya University College of Engineering and Technology for Women

Introduction

Tic-Tac-Toe is a classic 2-player strategy game. This project aims to develop an AI that plays Tic-Tac-Toe against a human player using the Minimax algorithm. The AI is designed to be unbeatable and always makes the optimal move.

Objective

- Build an intelligent game agent using Minimax.
- Understand decision-making in turn-based games.
- Learn the implementation of game theory and recursive algorithms.

Technology Stack

- **Language:** Python
- **Editor:** Visual Studio Code
- **Algorithm Used:** Minimax (without Alpha-Beta Pruning)

System Requirements

- Python 3.x
- Visual Studio Code
- Command-line terminal

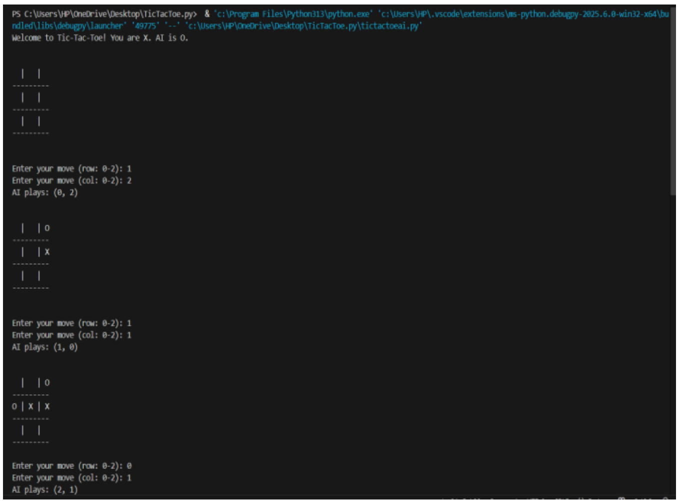
Project Architecture

- **Game Engine:** Handles board, player input, and display
- **Minimax Function:** Recursively evaluates all possible outcomes
- **AI Decision-Maker:** Selects the move with the highest score

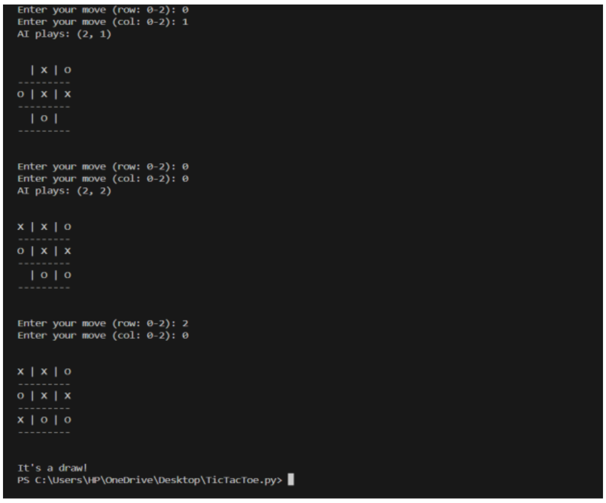
Working Explanation

1. The board is a 3x3 matrix.
2. The human plays as 'X', and the AI plays as 'O'.
3. The AI evaluates all possible future states of the game using Minimax.
4. The optimal move is selected to either win or force a draw.
5. The game continues until a win/draw condition is met.

Screenshots



```
PS C:\Users\HP\OneDrive\Desktop\TicTacToe.py>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\HP\.vscode\extensions\ms-python.debugpy-2025.6.0-win32-x64\bundled\libs\debugpy\launcher' '49775' '--' 'c:\Users\HP\OneDrive\Desktop\TicTacToe.py\tictactoeai.py'
Welcome to Tic-Tac-Toe! You are X. AI is O.


  |   |
---------
  |   |
---------
  |   |
---------


Enter your move (row: 0-2): 1
Enter your move (col: 0-2): 2
AI plays: (0, 2)


  |   | O
---------
  |   | X
---------
  |   |
---------


Enter your move (row: 0-2): 1
Enter your move (col: 0-2): 1
AI plays: (1, 0)


  |   | O
---------
O | X | X
---------
  |   |
---------

Enter your move (row: 0-2): 0
Enter your move (col: 0-2): 1
AI plays: (2, 1)
```



```
Enter your move (row: 0-2): 0
Enter your move (col: 0-2): 1
AI plays: (2, 1)


  | x | o
---------
o | x | x
---------
  | o |
---------


Enter your move (row: 0-2): 0
Enter your move (col: 0-2): 0
AI plays: (2, 2)


x | x | o
---------
o | x | x
---------
  | o | o
---------


Enter your move (row: 0-2): 2
Enter your move (col: 0-2): 0


x | x | o
---------
o | x | x
---------
x | o | o
---------


It's a draw!
PS C:\Users\HP\OneDrive\Desktop\TicTacToe.py> █
```

Conclusion

This project gave valuable insights into basic AI, recursive problem solving, and game theory. The Minimax algorithm ensures that the AI never loses, offering a practical application of AI in turn-based games.

Future Scope

- Add Alpha-Beta pruning for performance optimization.
- Build a GUI using Tkinter or Pygame.
- Implement multiplayer or network play.

References

- GeeksforGeeks (Minimax Algorithm)

- Python Official Documentation
- YouTube tutorials on AI game development