SMART TIC-TAC-TOE AI ENGINE (PYTHON)

Author: Routh Kiran Babu

TABLE OF CONTENTS

----

1. INTRODUCTION

Tic-Tac-Toe is a classic two-player board game played on a 3x3 grid. Although simple in design, it provides an excellent foundation to implement game logic, validation rules, artificial intelligence, and memory-based learning.

This project implements a Smart Tic-Tac-Toe AI Engine in Python that not only allows human vs computer gameplay but also enables the computer to learn from previous games using persistent memory storage.

----

2. GAME OVERVIEW

The game supports:

- Board validation
- Turn-based move execution
- Win and draw detection
- Score tracking
- Multi-round gameplay
- Memory-based AI move selection

The board is represented as a 3x3 matrix with numeric positions initially and replaced by 'x' or 'o' after each move.

----

3. CORE LOGIC EXPLANATION

Key functions used:

checkBoard():

Validates board dimensions and ensures only valid symbols exist.

checkFilled():

Checks whether the board is completely filled with player symbols.

winCondition():

Detects winning combinations horizontally, vertically, and diagonally.

update_board():

Safely updates board position after validating inputs.

run_step():

Controls one logical game step including validations.

run_round_odd() and run_round_even():

Handle game flow based on round parity.

----

4. AI STRATEGY AND LEARNING

The AI uses a file-based memory system (memory.txt) to store winning and losing move sequences. During future games, the AI attempts to reuse winning paths and avoid losing paths.

If no matching strategy is found, the AI selects a random available move.

This creates a simple reinforcement learning behavior.

----

5. CHALLENGES AND SOLUTIONS

Challenge: Board validation errors

Solution: Implemented strict type and dimension checking.

Challenge: AI move repetition

Solution: Introduced memory file storage.

Challenge: Handling invalid user input

Solution: Try-except validation and retry loops.

Challenge: Maintaining turn order

Solution: Step counter logic and round parity handling.

----

6. IMPORTANCE OF THE PROJECT

This project demonstrates:

- Algorithmic thinking
- AI fundamentals

- Data persistence
- Input validation
- Game logic design
- Python programming best practices

It is suitable for learning:

- Game development
- AI fundamentals
- Software testing
- Automation logic

----

7. CONCLUSION

The Smart Tic-Tac-Toe AI Engine is not just a game but a complete learning model for Python programming and artificial intelligence concepts. It highlights how even simple games can evolve into intelligent systems using memory and logical decision-making.

This project strengthens problem-solving skills and builds a foundation for advanced AI development.

AUTHOR

Routh Kiran Babu

Python Developer | AI Enthusiast | Software Learner
