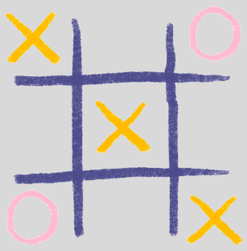
EGYPT UNIVERSITY OF INFORMATICS
FACULTY OF COMPUTING & INFORMATION SCIENCES

جامعة مصر للمعلوماتية
EGYPT UNIVERSITY OF INFORMATICS

TIC TAC TOE



Introduction

The Tic Tac Toe game features a 3x3 grid and a tkinter interface with two AI modes: Normal Mode, using simple heuristics, and Unbeatable Mode, which utilizes a Minimax algorithm. The game highlights winning sequences and provides a brief post-game analysis, demonstrating basic GUI and AI concepts in Python.

Objectives

1-Create two AI modes: a Normal mode with adjustable difficulty levels and an Unbeatable mode powered by the Minimax algorithm.

2-Integrate post-game analysis, including a review of possible moves and an evaluation of the algorithm Used, to enhance the overall user experience.

3- Understand and implement AI search algorithms in game development

Search Algorithms

1.Heuristic (Greedy) Search — Used in Normal Mode:

AI employs a heuristic (greedy) approach, focusing on immediate checks for winning and blocking moves without considering future moves. To adjust difficulty levels, the AI incorporates varying degrees of randomness: in Easy mode, it is 70% random and 30% strategic; in Medium mode, it balances with 50% randomness and 50% strategy; and in Hard mode, it becomes more calculated with 20% randomness and 80% strategic play.

2.Minimax Search (with Memoization) — Used in Unbeatable Mode:

Our AI employs the Minimax algorithm to deliver an unbeatable experience by evaluating all possible moves and selecting the optimal outcome to win or force a tie. It maximizes its advantage during its turn and minimizes the player's advantage during theirs. To enhance performance, the AI uses memoization, storing unique board configurations to avoid redundent calculations, ensuring fast and seamless gameplay.

- **Memoization:**
  Memoization improves efficiency by storing the outcomes of previously analyzed board configurations. When a repeated layout occurs, it retrieves the stored result instead of recalculating, enabling faster decision-making and maintaining seamless gameplay.

Team Members

Basmala Yasser (22-101098)
Abeer Sherif (22-101136)
Shahd Ayman (22-101045)
Renad Sameh (22-10225)

Obstacles

- Implementing the Minimax algorithm efficiently.
- Managing the GUI state transitions smoothly.
- Ensuring the AI's Unbeatable Mode performs optimally without delays.

Solution

- Used memoization to optimize Minimax.
- Structured the code with clear function categorization for better manageability.
- Tested thoroughly to balance AI performance and responsiveness.

Conclusion

The Tic Tac Toe project successfully combined intuitive GUI design with intelligent AI strategies. Featuring a heuristic-driven Normal Mode, and an Unbeatable Mode using the Minimax algorithm, it demonstrated effective integration of Python programming concepts and delivered an engaging user experience.

Supervised by

Dr. Mohamed Taher
Dr. Farid Zaki
Eng. Tukka mohamed