

Enhanced Tic-Tac-Toe with AI Opponent and Customizable Board

Team Members

- Harsimran Singh - 2401030148
- Arayan Sharma - 2401030159

Abstract

This project presents an enhanced version of the classic Tic-Tac-Toe game, incorporating an AI opponent and customizable board size. The AI utilizes the Minimax algorithm to provide a challenging and adaptive gameplay experience, significantly increasing the game's complexity and replayability beyond a simple, fixed-size board. Developed using Python and the Pygame library, the game features a visually appealing graphical user interface (GUI). The project employs object-oriented programming (OOP) principles to achieve a clean, modular, and maintainable code structure. The core functionality is encapsulated within classes representing the game board (`Board`), players (`Player`), and the AI (`AI`). The `Board` class manages the game state, including board size, move validation, win condition checks, and board rendering. The `Player` class represents both human and AI players, abstracting their move selection methods. The `AI` class implements the Minimax algorithm to determine the optimal move for the AI opponent. The main game loop handles player turns, input processing, and win condition checks, ensuring a seamless and engaging user experience. The use of Pygame facilitates the creation of an interactive and visually appealing game, making it an engaging and educational project that demonstrates key software engineering principles and AI concepts.

Topics of SDF Used

- Object-Oriented Programming (OOP)
- Algorithms (Minimax algorithm)
- Data Structures (Lists, potentially arrays for board representation)
- Control Flow (Loops for game iteration, conditional statements for win/loss checks)
- Functions and Modules
- Recursion (within the Minimax algorithm)

Header Files and Libraries

- Python 3.x
- Pygame library