Intelligent Tic-Tac-Toe: A Python-based Game with AI Opponents

Team Members

Harsimran - 2401030148

Abstract

This project implements a single-player Tic-Tac-Toe game using Python and object-oriented programming principles. The core functionality involves the creation of a console-based game where a human player competes against an AI opponent. The AI's intelligence is adjustable, offering varying difficulty levels. At its most challenging, the AI employs the Minimax algorithm, a well-known decision-making algorithm in game theory, to select optimal moves. The project emphasizes a clean, modular design using several classes. The `Board` class manages the game state, updating after each move and checking for win/draw conditions. An abstract `Player` class provides a foundation for both human and AI players, with subclasses (`HumanPlayer` and `AIPlayer`) implementing specific behaviors. The `Game` class orchestrates the gameplay, handling turns and player interactions. A separate `Difficulty` class (optional) allows for flexible difficulty adjustments. Functions such as `display_board()`, `get_player_move()`, `make_move()`, `check_win()`, `check_draw()`, `minimax()`, and `get_ai_move()` handle the game logic and AI decision-making. The project showcases fundamental AI concepts, game development principles, and object-oriented programming practices, while also providing a clear illustration of the Minimax algorithm's application in game AI. The console-based interface maintains simplicity, focusing on core functionality and code clarity.

Topics of SDF Used

- **Object-Oriented Programming (OOP) principles:** classes, inheritance, polymorphism.
- **Algorithm design and implementation:** Minimax algorithm.
- **Data structures:** lists/arrays for representing the game board.
- **Control flow:** loops for iterating through game turns and checking game conditions.
- **Functions:** modularization of code into reusable components.

Header Files and Libraries

- Python 3.x (standard library)
- `pytest` (optional, for unit testing)