Tic-Tac-Toe Game with AI Opponent

The Tic-Tac-Toe game with an AI opponent is a project designed to provide an engaging and interactive gaming experience for users. This report will discuss the important features of the project and their functions in detail.

1. Minimax Algorithm with Alpha-Beta Pruning:

The minimax algorithm is a fundamental component of this project that enables the AI opponent to make strategic decisions. The algorithm works by simulating possible game outcomes, minimizing the possible loss (for the AI) or maximizing the possible win (for the player). Here's how it functions:

minimax(board, depth, is_maximizing, alpha, beta): This recursive function evaluates possible game states by exploring different moves. It returns a score based on the game outcome, where -1 represents a win for the player, 1 for the AI, and 0 for a draw. The alpha-beta pruning technique is used to optimize the search by eliminating unnecessary branches, making it efficient even for deeper levels of exploration.

2. Different Difficulty Levels:

The project offers two difficulty levels for the AI opponent, catering to varying user preferences:

EASY: In this mode, the AI makes random moves, offering a less challenging experience.

HARD: In the hard mode, the AI employs the minimax algorithm with alpha-beta pruning to make more strategic and challenging moves, aiming for victory.

3. Move History List:

To enhance user experience and provide an undo feature, a move history list is maintained. This list keeps track of each move made during the game. The undo_last_moves(board, move_history, num_moves) function allows users to undo the last 'num_moves' moves.

4. Game Statistics:

The project maintains comprehensive game statistics to track the performance of both the player and the AI. The statistics include:

Player Wins: Count of games won by the human player.

AI Wins: Count of games won by the AI opponent.
Draws: Count of games that ended in a draw.
These statistics are displayed after each game, allowing players to track their progress and performance.

5. User Input Validation:

The project incorporates robust user input validation to ensure that the game operates smoothly and without errors. The get_user_input(prompt, valid_inputs) function verifies user input, ensuring it matches the expected options.

6. Undo Last Move Feature:

One of the unique features of this Tic-Tac-Toe game is the ability to undo the last move. The undo_last_moves(board, move_history, num_moves) function checks if there are enough moves in the history to undo 'num_moves' moves. It then updates the game board accordingly.

Conclusion:

In conclusion, the Tic-Tac-Toe game with an AI opponent is a well-designed project that offers a variety of features to enhance the gaming experience. These features include the implementation of the minimax algorithm with alpha-beta pruning, different difficulty levels, a move history list for undoing moves, game statistics for tracking wins and draws, user input validation, and an undo last move feature. These functionalities make the game engaging and enjoyable for players of all skill levels while providing a challenging AI opponent for those seeking a tougher match.