William Huang, Yun Lu
CS 5100
Spring 2025

# Progress Report on AI-Based Connect 4 Game

The goal of this project is to develop an AI-driven Connect 4 game that features both Player vs Player (PvP) and Player vs AI (PvAI) modes, with an AI opponent that adapts to different difficulty levels. The project integrates two advanced AI decision-making algorithms: Minimax and Monte Carlo Tree Search (MCTS), which together provide a challenging gaming experience. This progress report outlines the work completed to date and identifies areas that are still in progress or planned for future development.

## Completed Tasks

The basic game logic for Connect 4 has been implemented using Python and Pygame. The game board is represented as a 6x7 grid, with players aiming to align four pieces vertically, horizontally, or diagonally. Key functionality for the game, including the board setup, has been completed. The board ensures proper placement of pieces, and moves are validated according to the rules. Additionally, the game detects when a player has won by connecting four pieces in a row. Both Player vs Player (PvP) and Player vs AI (PvAI) gameplay modes have been integrated, allowing players to choose between playing with another player or against the AI on the terminal.

For the AI, two decision-making algorithms have been implemented to provide different levels of difficulty. The first, Minimax, is used in the easy mode. This algorithm recursively evaluates all possible moves and selects the one that maximizes the AI's advantage. To optimize performance, alpha-beta pruning has been implemented, allowing the AI to make decisions more efficiently. The second algorithm, Monte Carlo Tree Search (MCTS), is used in the hard mode. MCTS simulates multiple random gameplays to assess the potential outcomes of each move, selecting the move that provides the best chance of winning. This algorithm behaves more unpredictably and offers a greater challenge to players in the hard mode.

The core AI decision-making functionality has been integrated into the *AIAlgorithm* class, which includes methods for both Minimax and MCTS. The *minimax* function explores potential future moves and evaluates them based on the number of connected pieces in the center column, while also considering both offensive and defensive strategies. The MCTS method includes a *simulate_random_game* function, which runs simulations of random games to evaluate possible outcomes and selects the best move based on the simulations' results.

The graphical user interface (GUI) for the game has been developed using the Pygame library. The interface displays the game board as a 6x7 grid, with circles representing the pieces for each

player. Players interact with the game by clicking on columns to drop their pieces, and the game updates the board accordingly. The game alternates turns between players, and the AI automatically takes its turn when playing in PvAI mode. When a player wins, a message is displayed announcing the winner and halting further gameplay.

**Ongoing Tasks and Future Development**

Although both the Minimax and MCTS algorithms are functioning, there is still room for optimization for Minmax as the "Medium" mode of the game. Specifically, it could be further optimized to reduce computation time by increasing the depth of the search tree and improving pruning efficiency. As for "Easy" mode, an extra AI agent that moves purely random moves will be introduced to the game that will be suitable for beginners.

While the core functionality of the game has been implemented, the user interface could still be improved. Future versions of the game will include a main menu on the graphical user interface that will allow the player to choose the game mode ("Player vs.Player" or "Player vs. AI") and the difficulty mode when playing the AI ("Easy", "Medium", or "Hard").

Additional enhancements such as animations for win detections, and incorporating start/reset buttons and more detailed game status indicators, which will improve the overall user experience.

**Conclusion**

To date, significant progress has been made in the development of the AI-driven Connect 4 game. The foundational elements, including the game board setup, move validation, win detection, and basic user interaction, have been successfully implemented. The AI functionality, using Minimax and MCTS algorithms, offers varying levels of difficulty, and the game's user interface provides a smooth player experience. While the project is progressing well, further optimization of the AI, and user interface enhancements are planned for future versions.