ConnectFour - AI Game

All the code can be accessed [here](#).

A demo for the game can be accessed [here](#).

Project Definition

This project implements an AI agent for the Connect Four game using the minimax algorithm with alpha-beta pruning. Connect Four is a classic two-player game where players take turns dropping coloured discs into a vertical grid, aiming to connect four of their discs consecutively in any direction (horizontal, vertical, or diagonal). The game features a moderately complex search space of approximately 4.5 trillion possible board configurations, making brute force approaches impractical. However, the game's perfect information nature and well-defined rules make it an excellent choice for adversarial search algorithms.

The challenge lies in creating an AI agent that can balance the search depth with reasonable computation times. Connect Four is well-suited for minimax with alpha-beta pruning because the game tree can be efficiently pruned, and position evaluation can be conducted through pattern recognition. The project explores how varying search depths affect the AI's playing strength and computational efficiency, giving practical insights about adversarial search algorithms in game-playing environments.

Solution Specification

Game Implementation:

I created a complete Connect Four game environment using Python/Pygame, employing a modular design that separates game logic from visualization. The implementation uses NumPy arrays for board representation, enabling efficient manipulation/ evaluation of game states. The UI features smooth animations, hovering piece indicators that enhance player experience, and difficulty selection options for adjusting the challenge level.

AI Agent Implementation:

The core of the solution is the minimax implementation with several optimizations:

1. **Minimax with Alpha-Beta Pruning**: The algorithm explores the game tree to a specified depth, maximizing the AI's chances while minimizing the player's opportunities. Alpha-beta pruning

significantly reduces the search space by avoiding branches that can't influence the final decision.

2. **Optimized Evaluation Function**: A position evaluation function that examines "windows" of four positions horizontally, vertically, and diagonally. It assigns higher scores to patterns containing more AI pieces, especially for three-in-a-row configurations. It applies penalties for opponents' threatening positions and gives preference to the center column due to its strategic importance.
3. **Early Termination Checks**: Before deep searches, the algorithm checks for immediate winning moves for the AI/ moves that block an immediate win for the opponent. This allows for making obvious tactical decisions without unnecessary computation, improving performance in time-critical situations.
4. **Depth Control**: Implemented search depths (ranging from 2 to 6) to adjust the difficulty.
5. **Randomization**: Added controlled randomization in move selection when multiple moves have similar evaluations to create variety in play.

Analysis of Solution

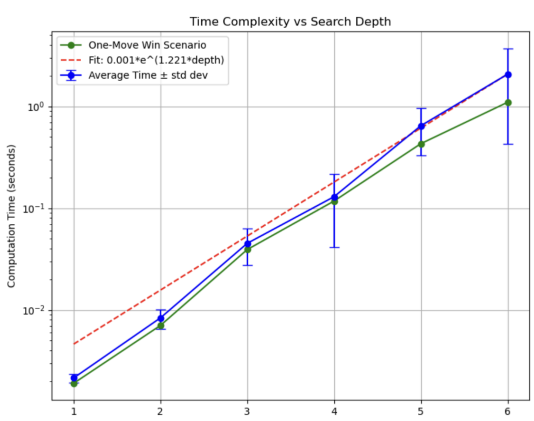

Fig 1.

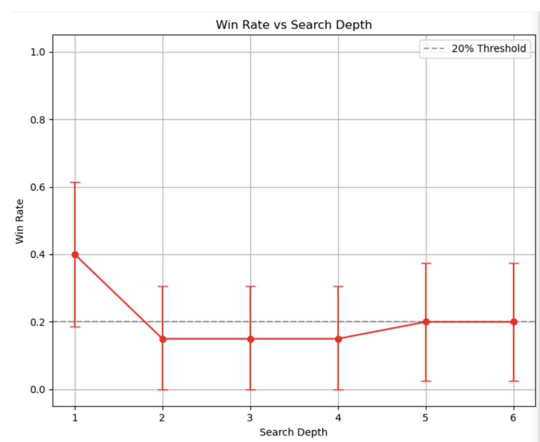

Fig 2.

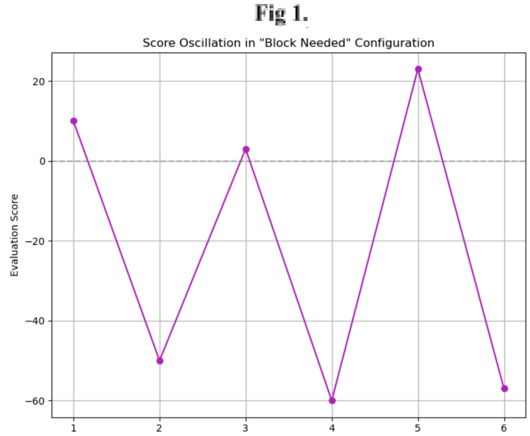

Fig 3.

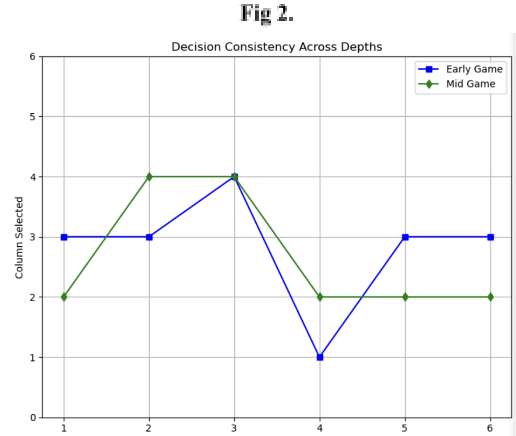

Fig 4.

I analyzed the AI's performance across search depths (1-6) using time complexity, win rates, and decision patterns. Fig. 1 shows **computation time** grows exponentially (0.001e^(1.221depth)), with depth 4 (0.129s) being the practical limit before slowdowns (2.069s at depth 6). The one-move win scenario runs 2-3 times faster, confirming alpha-beta pruning's efficiency for forced wins.

Win rates (Fig. 2) peak at 40% (depth 1) but drop to 15-20% (depths 2-6), with overlapping confidence intervals indicating no benefit from deeper search. The evaluation function over-prioritizes immediate wins (depth 1 success) but lacks positional nuance for higher depths. In defensive scenarios (Fig. 3), scores oscillate between positive (10-23, odd depths) and negative (-50 to -60, even depths), yet the AI always correctly blocks in column 0, revealing a disconnect between scores and optimal moves.

The Connect Four AI demonstrates perfect tactical execution (100% accuracy in forced wins/blocks) but reveals evaluation function flaws through score oscillations (10 to 60) tied to search depth parity**. Decision consistency** (Fig. 4) varies by phase: early-game favors columns 3-4 (except depth 4's column 1 anomaly), while mid-game stably alternates between columns 2 and 4. This suggests the AI adapts better to complex mid-game positions than simpler early-game ones.

Overall, the AI successfully identifies immediate winning moves at all depths, demonstrating basic tactical capability. The defensive capabilities of the AI show room for improvement, as evidenced by inconsistent blocking decisions. The AI reliably executes forced wins but struggles with positional evaluation. Depth 4 balances speed and performance, while deeper searches offer diminishing returns. Refining the evaluation function—not increasing depth—would resolve the observed inconsistencies in defensive scoring and early-game decisions. A more nuanced scoring system could better capture strategic patterns, as current results show minimax provides a strong foundation but requires refinement for optimal play.

AI Statement: *Only Grammarly was used to correct the report's sentence-level, spelling, and grammatical errors. No other AI was used.*

References

0xADADA, Ibsen, E., & Zhang, C. (2003, December 15). *A Connect Four Playing AI Agent: Algorithm and Creation Process*. https://0xadada.pub/2003/12/15/connect-four-playing-ai-agent/

Abdoul Wahab Touré. (2023). *Evaluation of the Use of Minimax Search in Connect-4—How Does the Minimax Search Algorithm Perform in Connect-4 with Increasing Grid Sizes? 14*(06), 419–427. https://doi.org/10.4236/am.2023.146025

Adversarial Search Algorithms. (2024, May 13). GeeksforGeeks. https://www.geeksforgeeks.org/adversarial-search-algorithms/

Artificial Intelligence: A Modern Approach. (2021). Pearson.com. https://www.pearson.com/en-us/subject-catalog/p/artificial-intelligence-a-modern-approach/P200000003500/9780137505135?srsltid=AfmBOooB2VuZPsrYHsjtCPMXc_M7sM44ZLJ81CSTsksN99Rk2s5bI5KL

Lim, B., San, G., Arabee, Z., Salam, A., Xin, Y., & Dhanoa, C. (2022). Connect-4 using Alpha-Beta pruning with minimax algorithm. *Journal of Applied Technology and Innovation*, *6*(1), 43. https://dif7uuh3zqcps.cloudfront.net/wp-content/uploads/sites/11/2022/01/15124632/Volume6_Issue1_Paper10_2022.pdf

Millington, I. (2019). *AI for Games* (3rd ed.). CRC Press. https://doi.org/10.1201/9781351053303

Nasa, R., Didwania, R., Maji, S., & Kumar, V. (2018). Alpha-Beta Pruning in Mini-Max Algorithm -An Optimized Approach for a Connect-4 Game. In *International Research Journal of Engineering and Technology*. https://www.irjet.net/archives/V5/i4/IRJET-V5I4366.pdf

Wikipedia Contributors. (2019, November 16). *Connect Four*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Connect_Four