CS 4200 Final Project Proposal: Connect Four

Corey Lang and Jenna Mauch

Problem Statement:

Game-playing agents can be built using a number of different algorithms. We want to investigate what algorithm can produce the best performance in a game of Connect Four. Best performance in this case is defined as the highest win-rate. Solving this problem will help give us firsthand experience with determining what algorithm can be superior over the other for a single case.

Literature Review:

In the development of an AI Connect Four game, Python code can be used to implement a game board using a matrix. This approach allows for efficient tracking of moves and easy evaluation of the current game state, which is the base environment for this project. Each state in the game will have some value set to true if the game recognizes a winning move. Alternatively, informed and uninformed searches can be used to explore the game tree and identify the best move. In an uninformed search, such as breadth-first search or uniform cost search, the search algorithm does not use any prior knowledge of the game state to guide the search. In contrast, informed searches, such as A* or greedy search, use heuristics or evaluation functions to estimate the value of each move and prioritize the search accordingly. In addition, reinforcement learning can be applied to improve the game's performance by allowing the AI to learn from its mistakes and successes. This technique involves updating the weights of the neural network based on the rewards received from the game, with the goal of maximizing the total reward over time. Reinforcement learning and informed/uninformed searches could be compared in terms of their ability to learn and adapt to different game scenarios when competing against each other. Further research is needed to explore the strengths and limitations of these approaches in the context of Connect Four and other similar games with turn-based decision making.

Approach:

To tackle the problem, we will be using search algorithms. The reason for this is because a search algorithm can be useful in determining the agent's next action based upon the current environment state. However, we are also considering using reinforcement learning. We will be using two informed search algorithms (Greedy search and A* search) and two uninformed search algorithms (Uniform-cost search and Breadth-first search). Upon completing our algorithms, we plan to determine the algorithm with the highest performance by making the AI play against each other in a tournament-like style, with each matchup determining the winner with a best of 5 games. We also plan to use the winner of the "tournament" to play against a human player, using the winrate of the agent as a measure for performance.

Our approach relates to our problem because it factors in comparing the different algorithms with each other, while also comparing the overall performance when in a game with a human agent. An algorithm can perform very well amongst other game-playing AI agents, but could perform poorly when against a human player, which helps to put the algorithm into a better perspective of how successful it is.

Resources:

For this project, we will be using several python libraries, including:

NumPy: a Python library for numerical computing
Random: a Python library for generating random numbers and data
PyGame: a Python library for game development
Math: a Python library for mathematical operations
Sys: a Python library for system-specific functionally

We will also be using GitHub for version control and collaboration between members. Additionally, we will be using a YouTube video by Keith Galli, *How to to Program a Connect 4 AI (implementing the minimax algorithm)* [1], for guidance in implementing certain features and functionality in our project. Our game-playing agents will be using Corey's pre-existing code of a matrix-based game of Connect Four. Finally, our algorithms will be based on the examples in *Artificial Intelligence: A Modern Approach* [2].

Team Members:

Corey Lang - Major in Computer Science, Undergrad
Jenna Mauch - Major in Computer Science, Undergrad

Work Breakdown Structure:

Due dates are subjected to change

Task	Due Date	Assignment	Notes
Greedy Search	March 10th	Jenna	
Breadth-First Search	March 10th	Corey	
A* Search	March 17th	Jenna	
Minimax Algorithm	March 17th	Corey	
AI Tournament	March 24th	Jenna & Corey	
AI vs. Human Agent	March 29th	Jenna & Corey	
Presentation Creation	TBD	Jenna & Corey	

References:

[1] K. Galli. How to Program a Connect 4 AI (Implementing the Minimax Algorithm). (Jan. 2, 2019). Accessed: Feb. 24, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=MMLtza3CZFM

[2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach 4th ed.* Upper Saddle River, New Jersey: Pearson, 2020.