Project Proposal

Our Team

We are working in a team of three. The members are Yichen Sun, Miles Vollner, and Andrew Yuan. We plan on splitting the work evenly over the three of us. The exact method, splitting by interface, class, algorithm, etc. will be decided when we have a better idea of how much code needs to be written and implemented (for example, do we need to build a basic 4-player chess program?).

Problem Formulation and Algorithm

The problem we are trying to address is The 4-Player Chess (team version 2 vs 2). This means teams has 2 agents on each side of the game board and try to play rationally to achieve a winning outcome. A possible formulation of The 4-Player Chess (team version) will be creating a 4-Player Chess board with Python library and utilize Mini-max algorithm learned in class which is a recursive strategy used in decision-making and n-players gaming. It provides an optimal move or action for the agents assuming all other opponents is also playing optimally. For the algorithm, the input will be each tree node as a game state, a minimizer and a maximizer, and the agent in the same team will act optimally to maximize own team and minimize oppnonents. Then, the desired output will be a terminal node that provides the terminal values and backtracks the tree until the initial state occurs.

Idea Outcome

The ideal outcome of the project would be to create an AI that beats a randomly moving AI 100% of the time, can beat other 4 player chess AIs(The 4-Comfuter), and when the 3 of us play with the one AI on a team we'd all want the AI on our team. Our minimum viable product would be to have a working AI that preforms strategy specific to 4-Player chess and can beat a randomly playing agent more than 50% of the time. What we expect to show through achieving this is that our AI has a solid understanding of 4-Player chess strategy and can significantly outperform agents without strategy and maybe even other 4-Player chess AIs and humans.

Expected Algorithm

The project will likely revolve heavily around minimax and creating an evaluation function that performs well in the 4-player chess environment.

Topic, Libraries, and Platforms

Provide references where applicable
Some things that we may need to look into:

- Javascript (integration with web)
- PyGame
- PyTorch
- Source code of chess engines (if available)

Machine Learning Component and Dataset

Our project most likely will not have a machine learning component but if we do decide to implement it we would probably create a dataset going thro many games with our AI

to use as reinforcement learning to improve our AI. I don't know if we would have the resources to do so as the amount of possible gamestates is huge which is one of the main reasons we will probably avoid using a machine learning component.

Milestones

Milestone 1:

By milestone 1, we want to have done research into the following:

- Techniques used in vanilla chess AI (non-ML related) and figure out if there is existing code/techniques we can adapt to our model
- Does a viable 4-player gameboard exist / can we interface our games with Chess.com. If so, we have code that allows us to interact with the website. If not, we will have the gameboard and pieces properly configured in such a way that allows us to play the game reasonably well (maybe not UI, but a text visual that allows us to play).

We also want to have at least some ideas to modify the base minimax so that the state explosion is minimized.

Milestone 2:

Have a first implementation of a 4-player chess algorithm that can play games on the board and have started implementing improvements to the algorithm.

Presentation:

We will demonstrate the goals and outcomes of each minestone. The problem we are trying to achieve is the implementation of AI game playing on The 4-Player Chess (2 vs 2 team version). We concludes how to use Mini-max algorithm precisely in the project, showing a demo playing with agents in the game. Furthermore, we will prospect potential extensions on 4-player chess game, like applying different search algorithms.

Final report:

The final report will be a complete written report explaining our algorithms we are using (minimax algorithm) on 4-player chess game AIs. Some problems occurred during the implementation of The 4-Player Chess game. In addition, some possible extensions and considerations for this game playing project in the future.

Week by week Plan

- Week 1
    - Know wether we can use an existing gameboard or if we need to build our own
    - If the latter have made progress on making the board, have at least the physical board and some pieces by the end of the week
    - Have picked out techniques that we think will be useful
- Week 2
    - Have a working gameboard
    - Have started to test out techniques we have found
- Week 3
    - Create minmax function

    - Test at least one evaluation function
    - Create some useful heuristics
- Week 4
    - Start testing evaluation functions and heuristics
    - Begin to finalize our strategy and streamline our algorithm
- Week 5 onwards
    - Continue to optimize our algorithm
    - Start testing our algorithm against The 4-Comfuter and against us
    - Streamline space usage
    - Make sure our minimum viable product has been achieved
    - Push to achieve our ideal goals