Connect 4

Author: Tamara Alhajj Student ID: 100948027

Overview

An informative explanation of the traditional game can be found at: https://en.wikipedia.org/wiki/Connect_Four

My implementation for this game allows for the conventional game play as well as additional features. These include:

```
- The option to remove your players piece from bottom peg
- Smart AI vs Human
- Smart AI vs Dumb AI
  - Dumb AI uses random moves
- Smart AI vs Average AI
  - Average AI uses minimax alpha-beta pruning with depth 1, thus finding
best moves at a glance
```

Play

To play run `python3 connect4-GUI.py` from the working directory. Then simply follow the instructions given by the CLI to start the GUI with your desired features.

AI

This is a two player game with minimax AI using alpha beta pruning. The state space is a 2D numpy matrix, with dimensions 6 by 7. These are the conventional dimensions of a connect 4 game.

The heuristics I used for my minimizing/maximizing functionality were an offensive scoring, and the other, a defensive scoring. Significantly, in both cases I use the static strategy of prioritizing the center column, as there are more possibilities there. Both mechanisms play effectively, however the defensive mechanism plays more timidly, and will occasionally miss a winning move to block their opponent. I removed the ability for the AIs to remove node, because against a human the AI will surely win regardless. It was a costly move check with removal capabilities, so I have them handle the *threat* of removal, which works well. For example, often the AI player will set up two winning possibilities with an intelligent play to defend against removal of the bottom peg. This strategy is done primarily by the defensive player.

The number of nodes searched were consistently in the thousands. This is due to the fact that node expansion is based on the production system and the depth of the alpha-beta search. Currently the depth cutoff is at 5. However, with a lower cut off, say 3, the number of nodes searched were in the hundreds. Moreover, when the Average AI is playing with the Smart AI, they *both* use the alpha-beta search so this can magnify the number of nodes searched to the tens of thousands! As for the heuristics the number of nodes searched seem to be the same, however it is hard to make a direct comparison with so many game play possibilities.

Improvements

Alpha-Beta pruning would work best if the best move was considered first, otherwise it is costly since it must search all trees one by one. Notably, the killer heuristic optimizes by checking the move that caused the beta-cut off at a similar level last time, and will be checked first. A similar heuristic to this is the history heuristic, which saves the best moves to reference using a dictionary of previous moves. Essentially, the history heuristic is the killer with memorization.

Acknowledgements

Thank you to Keith Galli for his video series of connect 4 using pygame.