

CS 171 Connect K Report

Author: Allen Sallinger 81675129

Team: PossiblySkynet

Heuristic:

For the heuristic generation I created a function named `rateState`, which takes in the board state and determines the best approximate move for the player to make. This is done through a weighing of current board features for the player and their opponent. For each spot on the board we try to see if there is already a player piece there. If there is not then a move is made randomly. When a spot is found that has a player's piece or an opponent's piece then a search is done to see if there is a possible win path or a possible losing path for the player if an opponent's path is found to win. The opponent's pieces are counted and the player's pieces are counted. The maximum value is then taken from these values. If it is possible for the player's path to increase closer to k and the opponent's path is $< k-1$ then that is the maximum move chosen for that position. The groups of maximum moves for each of the possible 8 win paths is then saved and the maximum of all of the best moves for a spot are then judged for the overall best move. That is the move that is chosen. This is further enhanced with Alpha beta pruning.

Alpha-Beta Pruning:

With the Alpha-Beta pruning I created a cut off. This cutoff is for possible moves that would perform worse than the best moves for the opponent to make and better than moves the player could make if the opponent AI is playing optimally. With this done I was able to gain an extra level of depth to the minimax search tree.

IDS:

For IDS I was able to get the tree search to go to a depth of 3 with the current minimax with Alpha-Beta pruning in place. This is due to Python being my language of choice for this assignment. So surprisingly there was not much of a positive affect with this implementation. If I was using a more efficient language then there could possibly be a better effect or if the deadline was variable.

Selecting Children during IDS:

For IDS I just remember the best move of the last level that was found. This is the move that is then used/chosen by the AI for that turn.

Other:

For this assignment I did not go any further than the IDS as without reworking the code in another language, Java or C++, there would be not real benefit of implementing any of the other searches even though Monte Carlo can be better for timing. So for the last portion I just reran through the heuristic to make it as good as possible as that is what has the most effect on how my AI performs.