Name: Sen Lin Perm:6959399

CS165A Machine Problem 1 Report

Architecture:

In my CS165AMP2.cpp file, the main class is to read the input of messages like “-l” and boardsize “-n<int>” to setup the board size and the color of chess my program is gonna use during the game. After that, the real game begins at the while loop, it keeps reading and reacting to the coming input by place the chess from opponent and place its own chess by evaluate the value of each empty place on chessboard. It does 2 layers of estimation, 1st is to find the one with high enough score for us, second is to find the next step opponent most likely to do if we do step 1, then minus the score of step 1 with score of step 2 to find out the highest value over these 2 layers of estimation then make choice of which step 1 to make. The program also contains class to find if the location is on board, transfer message from string to two integers for 2d vector as position on board. And a class to act moving from one position to another, and finally, scoring class.

1. The program setup setting according to the first line of input to decide if the ai hold black or white chess and the size of chess board. Then go to the while loop of keep reading movement made by opponent, print out, change value of 2d vector chessboard.
2. The program run into massive for loops to evaluate every empty spots on the chess board. It first evaluate if we are doing attack first by finding the greatest score of attacking minus the greatest score of step opponent might do next. It has a pruning to cut down the runtime might caused by running some useless loops.
3. Once the program searched for all empty spots and return the highest scored result from them, print out the move made by the AI.

Searching:

The program iterate through all empty spots, and for every empty spot, it gives out the score of placing my chess on it by using the function score which recorded all kind of situation around this empty spot and how many score it worth, then it creates a temporary chessboard to estimate what if I put my chess here and what's the best choice opponent might make. After getting 2 scores (1 for my own chess, 1 for opponent), use the score of my AI to subtract the score of opponent's best choice based on the move I made. Store the highest score and the position to place chess which achieve that highest score, the position would be my next step.

Challenges:

It tooks me a lot of time to implement all different situation a empty position on chess board might facing, and I was also stuck in doing the second layer search for estimate opponent's move. Furthermore, I think the correct value we give to different situation is also important, I had a few times when my AI choose a place which that can form two 3-in-a-row instead of the place of forming 5-in-a-row for myself or enemy.

Weakness:

The search could be deeper to 3rd step, 4th step, even 5th step. But it might runs to a timeout because It takes much more time than I think to do that. And I might missed some situation for scoring I never know because that part is too messy to do further change. And scoring might have a better way to do so.