

Documentation for the program:

The I/O for sample games in action are in a notepad file included in this folder. The reason why is because the transcript is longer and would be inconvenient to look at on a pdf.

The heuristic explanation is also in a separate pdf file in the file since it was already written up for part A of the homework.

The program consists of a minimax function within the player class that allows the user to look ahead two moves for every call of the function. It can look deeper than 2 moves but having more than a 2 move lookahead is extremely lengthy.

After the lookahead is complete, it compiles a list of the best 5 moves and then does a Monte-Carlo search of these 5 best moves and plays 150 games with itself (the moves chosen after the determined best moves are random to save time). And then it looks at the record of the games: wins are worth 2 points, draws are worth 1 point, and losses are worth 0 points. Then the minimax formula returns the move with the highest point value out of the five moves.

**Extra Credit:**

It also incorporates alpha-beta pruning where it breaks the for loop that the pruning is a part of if it finds that it can prune the node. Depths 3 and 4 without alpha pruning could not finish due to very long search time.

**Alpha-Beta results:**

Lookahead Depth	Time to Search (seconds)	Nodes Visited	Alpha Pruning?
2	18.50039	76440	N
2	3.36348	12684	Y