Project Summary : Build a Game Playing Agent

A set of game playing algorithms were used to solve a 7x7 game of isolation where pieces moved like knights on a chessboard. More specifically, the MiniMax and AlphaBeta algorithms were explored in depth. Both these algorithms depend on the use of an evaluation function to make decisions about which path in the game tree to explore further and return a result. In this summary, I evaluate a set of custom heuristics developed to outperform those provided at the start of the project.

Heuristic #1 - Aggressive Player

In this approach, the algorithm makes the player act more aggressively prioritizing game boards in which the difference between player and opponent moves is the greatest.

$$H_1 = (2 * playerMoves) - opponentMoves$$

Heuristic #2 - Player Distance

Another method to evaluate game boards was to maximize the distance between players on the board. The thought being, that maintaining greater distance between players would maximize freedom of movement at all turns for our player.

$$H_2 = \sqrt{(x_{player} - x_{opponent})^2 + (y_{player} - y_{opponent})^2}$$

Heuristic #3 - Center Location

7	2	3	4	4	4	3	2
6	3	4	6	6	6	4	3
5	4	6	8	8	8	6	4
4	4	6	8	8	8	6	4
3	4	6	8	8	8	6	4
2	3	4	6	6	6	4	3
1	2	3	4	4	4	3	2
	1	2	3	4	5	6	7

This graphic shows the theoretical maximum number of moves available to the player at each position. The final heuristic tested gave bonus points to positions where the possible number of moves was greater than or equal to four.

If player in red square:

$$H_3 = playerMoves - opponentMoves$$

If player in green square:

$$H_3 = (playerMoves + 2) - opponentMoves$$

Results

Match #	Opponent	AB_Improved Won	AB_Improved Lost	AB_Custom Won	AB_Custom Lost	AB_Custom_2 Won	AB_Custom_2 Lost	AB_Custom_3 Won	AB_Custom_3 Lost
1	Random	186	14	188	12	188	12	182	18
2	MM_Open	153	47	158	42	150	50	142	58
3	MM_Center	174	26	171	29	176	24	169	31
4	MM_Improved	138	62	153	47	138	62	148	52
5	AB_Open	108	92	115	85	104	96	99	101
6	AB_Center	121	79	118	82	113	87	115	85
7	AB_Improved	112	88	102	98	94	106	99	101
	Win Rate:	70.9%		71.8%		68.8%		68.1%	

When I initially ran this study, the total game count was only 210. This did not yield consistent results across trials. Therefore, I increased the number of games played to get a more consistent result. A total of 5600 games were played between different agents to arrive at the results shown in the above table.

I was a little surprised that maximizing distance between players (H_2) did not yield a better result, as I expected this heuristic to keep player moves better maximized throughout the game. Similarly, I though keeping the player away from the corners (H_3) would yield in a better result. If I were to redo the analysis, I would increase the threshold for applying the bonus from 4 moves to 6 moves.

H_1 turned out to be the most effective game playing heuristic (it just so happened that it was the simplest to implement). This could be for a number of reasons :

- Compared to the AB_Improved heuristic, H_1 emphasizes boards where the absolute number of moves for our player is greater.
- Emphasis on aggressive moves at the start of the game likely resulted in more effective alpha-beta pruning. Pruning at the highest levels of the game tree can yield much better results.
- Given the simplicity of the algorithm, it’s likely that we could search more ply before the search timeout() function was triggered (there are fewer calculations per ply).