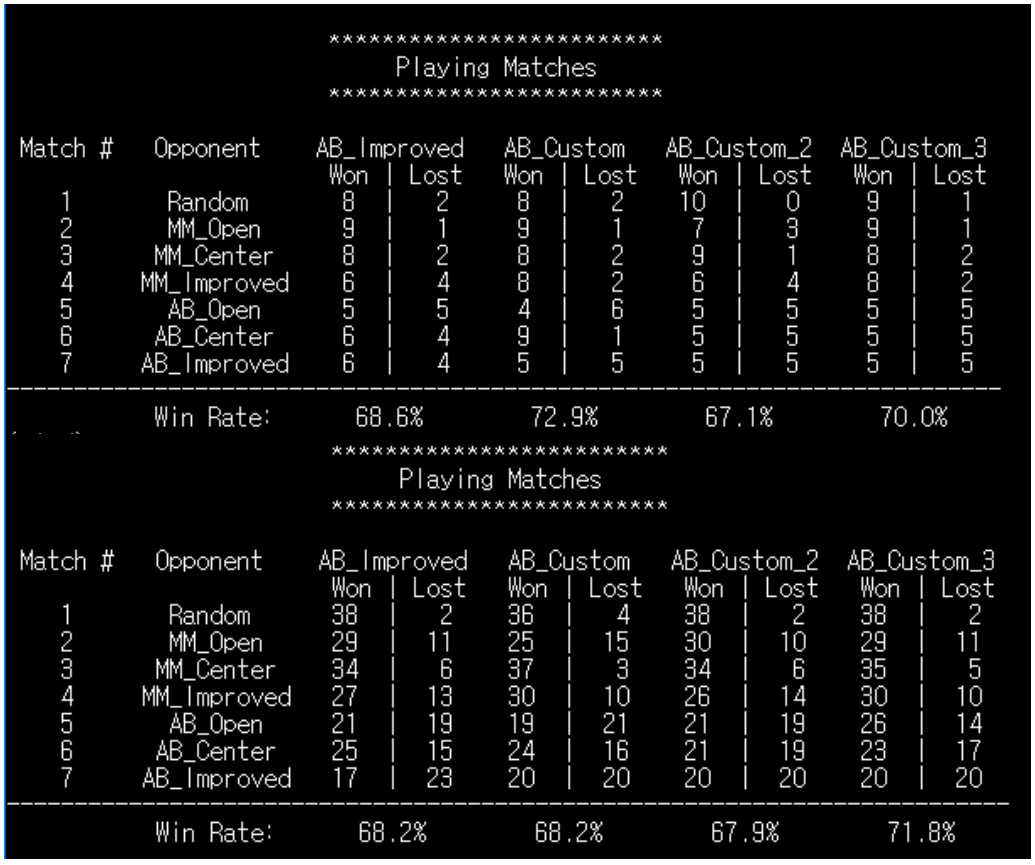
Udacity – AIND (Build a Gameplay Agent) Project Analysis

I implemented total 3 heuristic evaluation functions. For the evaluation function 3, I counted the players' legal moves and opponents' legal moves. The return value is the difference between the numbers of players' legal moves and the numbers of opponents' legal moves. When it comes to evaluation function 2, the basic concept is similar to the first one (players' legal moves – opponents' legal moves). However, I used the different weight based on the available (remained) moves on the game. In other words, if the available moves are smaller than about half of the total number of moves on the game, I thought that it will be more reasonable to limit the opponent moves in order to increase the winning rate within the limited available spaces. Therefore, in that case, I used the strategy, which is players' legal moves – 2*opponents' legal moves. Lastly, for my evaluation function 1, I made use of the heuristic function integrating legal moves and center score for both player and opponent. The more players' legal moves and opponents' distance to center is, the more winning rate is. Therefore, the final return value will be as follow: players' legal moves – opponents' legal moves + opponents' distance to center – players' distance to center. Below picture is the final tournament result for all 3 heuristic evaluation functions. (10 games (Top), 40 games (Bottom))



****
Playing Matches
****

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	8	2	10	0	9	1
2	MM_Open	9	1	9	1	7	3	9	1
3	MM_Center	8	2	8	2	9	1	8	2
4	MM_Improved	6	4	8	2	6	4	8	2
5	AB_Open	5	5	4	6	5	5	5	5
6	AB_Center	6	4	9	1	5	5	5	5
7	AB_Improved	6	4	5	5	5	5	5	5
	Win Rate:	68.6%		72.9%		67.1%		70.0%	

****
Playing Matches
****

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	38	2	36	4	38	2	38	2
2	MM_Open	29	11	25	15	30	10	29	11
3	MM_Center	34	6	37	3	34	6	35	5
4	MM_Improved	27	13	30	10	26	14	30	10
5	AB_Open	21	19	19	21	21	19	26	14
6	AB_Center	25	15	24	16	21	19	23	17
7	AB_Improved	17	23	20	20	20	20	20	20
	Win Rate:	68.2%		68.2%		67.9%		71.8%	

As you can see, the average performance (winning rate) of 1st evaluation function is 70.5% and 3rd evaluation function is 70.8%, both better than that of 'AB_Improved'. The take way in this result is that 3rd evaluation function is the best one even though I initially expected that 1st heuristic function would be the best one since the 1st heuristic logic covers both the center distances and legal moves. The reason of relatively low performance of 1st heuristic evaluation is that the simple linear combination of center distance and legal moves evaluation function didn't give the synergy. In this case, 3rd evaluation is the best heuristic evaluation. First of all, the test result proved the best performance. The overall winning rate is 70.8%, and it is obvious to pick it up as the best one. Secondly, 3rd heuristic evaluation is overall robust/strong to all of 7 opponents. For example, 1st evaluation is too weak toward specific opponent, such as the one using 'Open' score algorithm (MM_Open, AB_Open). Therefore, 1st heuristic cannot be the best one even though the average performance is almost close to 3rd one. Finally, the 3rd heuristic function is the simplest, and might have lowest computational complexity. The function

just calculates the difference between players' legal moves and opponents' legal moves. However, 1st evaluation function calculates the additional information, distance to center for both parties. 2nd evaluation function also requires additional information such as currently available moves, and has conditional statement.

However, in this project, my conclusion is that "The optimal heuristic function is difficult to implement". In order to organize optimal heuristic, we might analyze our algorithm in more detail, and it could be the future homework for Udacity student.