ANALYSIS OF RESULTS

Two simulations were performed, using the tournament.py script, to evaluate the performance of the custom heuristics, the results are as follows:

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	8	2	9	1	8	2
2	MM_Open	7	3	5	5	6	4	6	4
3	MM_Center	8	2	9	1	6	4	9	1
4	MM_Improved	7	3	6	4	5	5	6	4
5	AB_Open	5	5	4	6	6	4	7	3
6	AB_Center	6	4	6	4	7	3	6	4
7	AB_Improved	6	4	4	6	5	5	1	9
	Win Rate:	68.6%		60.0%		62.9%		61.4%	

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	10	0	9	1	8	2
2	MM_Open	3	7	7	3	5	5	6	4
3	MM_Center	7	3	6	4	9	1	10	0
4	MM_Improved	5	5	8	2	7	3	6	4
5	AB_Open	5	5	5	5	7	3	3	7
6	AB_Center	7	3	6	4	7	3	3	7
7	AB_Improved	5	5	5	5	7	3	4	6
	Win Rate:	60.0%		67.1%		72.9%		57.1%	

As can be seen from the results obtained, in most cases, the 3 AB_Custom agents performed well using the evaluation heuristics. In both cases, the AB_Custom_2 agent performed better, reaching 62.9% of the time in the first simulation, and 72.9% in the Second, since it is not possible to know the heuristics used by each agent, it is difficult to evaluate the performance of each heuristic individually, in the first simulation the agent AB_Custom presented the worst performance with only 60% of the hits, and the agent AB_Custom_3 presented the worst performance in In all cases, when the opponent is Random, the agents presented the best result, in the first simulation when the opponent is AB_Improved, and in the second when the opponent is AB_Open, although modest the results are oils.

The first evaluation heuristic calculates the number of moves that the player has available at least 3 times the number of moves that the opposing player has available in the state of the current game, returning less infinity if the player loses, more infinite if the player wins, otherwise returns the number of moves that the player has less than 3 times the number of moves that the opponent has in the state of the current game.
the idea is that this evaluation function can help the agent to choose moves in which he has more possibilities of movement, thus helping him to win.
The second heuristic is a less aggressive version of the first, whose only difference is that it calculates the number of moves that the player has minus 2 times the number of

moves that the opposing player has in the current game state.
The third heuristic, check if there is a partition, if it exists, returns the amount of available squares for the current player minus the amount of squares available to the opposing player in the current game state, it returns 0.0 if there is no partition, for that uses an auxiliary function called checks_partition. the idea is that if there is a partition the player that is on one side of the partition with the largest number of squares available will win because there are more options of movements.