Introduction

There are 2 objectives in this project. The first one is to implement the Minimax and Alpabeta pruning alghoritms. The second part is to try different heuristics and see how they behave against some other players. The aim of this document is to describe this heuristics.

For testing the performace of the heuristics I used the code from tournament.py. Since there are some randomness involved in the games played I increased the number of matches played to 1,000 for each heuristic.

Heuristics

For testing the 3 different heuristics 4 bots have been created. One as benchmark and 3 other using each heuristic. Those bots have competed against 7 simpler bots. The number of wins for every bot can be seen in Table 1.

	AB_Improved	**AB_Custom**	**AB_Custom_2**	**AB_Custom_3**
Random	179	179	172	178
MM_Open	146	167	158	136
MM_Center	167	168	175	172
MM_Improved	143	145	133	139
AB_Open	118	120	117	106
AB_Center	108	112	123	108
AB_Improved	113	95	103	95
Total	**69.43%**	**70.43%**	**70.07%**	**66.71%**

Table 1: Number of wins for each heuristic

With these stats we can see that bot 1 and 2 are slightly better than the benchmark bot. So the heuristics they are using are better (Details in Figure 1).

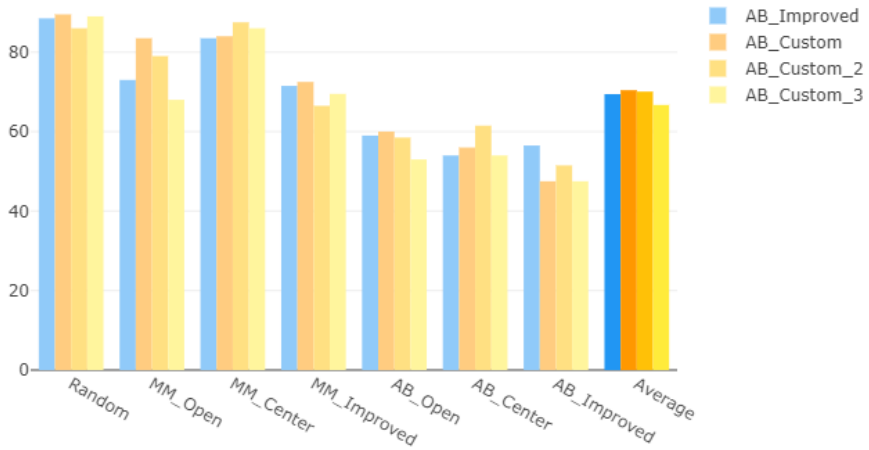

Figure 1: Percentage of wins for each heuristic

Common aspects of each heuristic

The 3 heuristics work differently depending on how many blank space are left in the board. By using that they can be aware of some problems that can happen at the end of the game.

Heuristic 1 (AB_Custom)

If less than 75% of the squares have been occupied the bot will only count the number of own moves vs the opponent ones.

For the last part of the game the bot will be more focused on surviving rather than trapping the opponent.

Heuristic 2 (AB_Custom_2)

If less than 50% of the squares have been occupied the bot will take into account 2 things:

- Number of moves - number of opponent's moves
- A factor that stats how far the player is going from the center

And for the other part of the game the bot will only count the number of moves since it does not matter any more if it is centered.

Heuristic 3 (AB_Custom_3)

For the first part it will behave like the first bot. In the second part it will also be penalized if it tries to go to the corners.

Results

After trying a lot of different heuristics there are some things that can be deduced:

1. Simplicity matters. Difficult heuristics means less computing time and the bot is not able to dive very deep
2. The most important factor is the number of own moves vs opponent moves
3. The other ideas tested doesn't seem to affect positively

Given all of the above the first bot (Heuristic 1) is the best one.