

## Custom heuristics analysis

### Custom score functions

We implemented three different custom score functions:

1. 'Easy win' (cust\_score): This heuristics is the similar than the AB\_Improved agent uses. It is enhanced with a feature that checks if our own move would block the only choice our opponent has left. Moving there gives us an easy win. The winning move gets a score that is bigger than any other score.
2. 'Static' (custom\_score2): This score function contains a static array with number of possible moves on an empty board (Answer to the question: How many moves do I have left if I move onto that cell from some other cell? Keeping in mind, that the cell where I move from 'eats' one move). The rationale behind this is that when the game is played, score of every cell is going down approximately equally. While this in reality is not true, it is, as said an approximation. As an enhancement, the cells around the center cell get an increased score, because from there the player gets a little bit more freedom to move. Also, during the opening game, we try to keep the player in the two outer rings of the board, where are fewer possible legal moves (going there is the later phase can be lethal). Then, during the mid-game, utilise more the center of the board. The idea of using the array is to have a fast table lookup instead of search over all possible legal moves. The speed hopefully would allow us to search a little bit further during the given time slot.
3. 'Chasing' (custom\_score3): This heuristics tries to make opponent's life as miserable as possible. The score is the better the less our opponent has moves left. In practice, we can block one cell with one of our own moves. We choose one of those move if possible. However, a move cannot happen without ensuring that we still have own moves left. If our own available moves would go to zero (and this would not be the winning move, i.e. opponent has moves left), the score will be zero. Implicitly this heuristic includes also the 'easy win' check.

### The results

50 rounds (25 as first player) against each test agent gave the following results:

***** Playing Matches *****									
Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	44	6	45	5	47	3	46	4
2	MM_Open	33	17	43	7	36	14	33	17
3	MM_Center	44	6	42	8	42	8	40	10
4	MM_Improved	29	21	36	14	37	13	35	15
5	AB_Open	24	26	24	26	23	27	27	23
6	AB_Center	30	20	29	21	25	25	30	20
7	AB_Improved	28	22	25	25	24	26	23	27
-----									
Win Rate:		66.3%		69.7%		66.9%		66.9%	

Looks like 'Easy win' heuristics outperforms all others, when played against minimax agents. 'AB\_Improved' is the best against alpha-beta agents, but 'Chasing' heuristics is only two defeats (out of 150 possible) worse. More data would be needed to analyse if this has any statistical significance.