

The results listed below were generated using the provided python script titled “tournament.py”. The performance of the *custom\_score* was tested against a baseline agent using alpha-beta search and iterative deepening (ID) called `AB\_Improved`. The three `AB\_Custom` agents use ID and alpha-beta search with the *custom\_score* functions defined in *game\_agent.py*.

***** Playing Matches *****									
Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	9	1	9	1	9	1
2	MM_Open	6	4	6	4	7	3	4	6
3	MM_Center	7	3	6	4	9	1	8	2
4	MM_Improved	7	3	5	5	6	4	4	6
5	AB_Open	4	6	4	6	6	4	4	6
6	AB_Center	5	5	5	5	6	4	9	1
7	AB_Improved	5	5	6	4	1	9	5	5
-----									
Win Rate:		62.9%		58.6%		62.9%		61.4%	

### Custom Score – Main heuristic I (AB\_Custom)

This heuristic uses the two data points on the agent and the opponent on every turn to weight the next move, we are utilizing the following equation to generate weights:

$$5 * ( \text{count} * \text{my\_distance} ) ** 2 - ( \text{count\_opponent} * \text{opponent\_distance} ) ** 2$$

- Count = number of legal moves that our agent has
- my\_distance = how far is the agent from the centre of the board
- count\_opponent = number of legal moves that our opponent has
- opponent\_distance = how far is the opponent from the centre of the board

### Custom Score 2 I (AB\_Custom\_2)

This heuristic is a variation on the method used in the first custom score

function, the way we calculate the weights is calculated by dividing the opponents moves as opposed to deduction

```
return 1*5* (count * my_distance) / (count_opponent * opponent_distance)
```

- Count = number of legal moves that our agent has
- my\_distance = how far is the agent from the centre of the board
- count\_opponent = number of legal moves that our opponent has
- opponent\_distance = how far is the opponent from the centre of the board

### **Custom Score 3 I (AB\_Custom\_3)**

This heuristic utilizes the number of legal moves for each player, we deduct the number of legal moves of opponent from the agent's move and return that value as the weight of the next step.

## **Result**

Custom heuristic one yielded the best results over testing with a win percentage of 62.9% - I think the reasoning behind that is that it takes into account 3 inputs that are dynamically readjusted on every move.

The following chart compares the overall



Giving all of the above, **I choose the Custom One**

1- It takes into account four factors when calculating the weight of the next move which quite substantial in terms of making a better decision on the agent side.

2- strikes a good balance between simplicity (allowing for deeper search) and accuracy (better predicting the outcomes of a position)

3- although second heuristics had the same overall score, the first one did better against stronger agents while the second one's performance was deteriorating as the opponent agent got stronger.