Heuristics Analysis (AIND Isolation)

Heuristic functions implemented

1. custom_score – A basic variation of improved score. It calculates the difference between the number of own player moves and opponent player moves. It wights the number of opponent player moves with the factor 2.

```
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
return float(own_moves - 2*opp_moves)
```

2. custom_score_2 - This function is the improved center_score in sample_players. It calculates the difference between own player distance and opponent distance from center.

```
w, h = game.width / 2., game.height / 2.
y, x = game.get_player_location(player)
own_moves_from_center = float((h - y)**2 + (w - x)**2)
a, b = game.get_player_location(game.get_opponent(player))
opp_moves_from_center = float((h - a)**2 + (w - b)**2)
return float(opp_moves_from_center - own_moves_from_center)
```

3. custom_score_3 - This function calculate the distance between own player and opponent player.

```
own_moves = game.get_legal_moves(player)
opp_position = game.get_player_location(game.get_opponent(player))
mid_width = float(game.width / 2)
mid_height = float(game.height / 2)
score = 0
for move in own_moves :
    score = score + (move[0] - opp_position[0])**2 + (move[1] -
opp_position[1])**2
score = 1 / max(score,100)
return score
```

Agent Information

1. Random: Agent that randomly choose a move each turn
2. MM_Null: Agent using fixed-depth mini-max search and the null_score function
3. MM_Open: Agent using fixed-depth mini-max search and the open_move_score function
4. MM_Improved: Agent using fixed-depth mini-max search and the improved_score function
5. AB_Null: Agent using fixed-depth alpha-beta search and the null_score function
6. AB_Open: Agent using fixed-depth alpha-beta search and the open_move_score function
7. AB_Improved: Agent using fixed-depth alpha-beta search and the improved_score function

Results

Match	Opponent	AB_Improved		AB_Custom		AB_Custom2		AB_Custom3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	6	4	9	1	10	0
2	MM_Open	8	2	9	1	7	3	7	3
3	MM_Null	10	0	10	0	10	0	9	1
4	MM_Improved	8	2	9	1	9	1	8	2
5	AB_Null	5	5	5	5	4	6	4	6
6	AB_Open	5	5	7	3	6	4	5	5
7	AB_Improved	5	5	8	2	4	6	5	5
	Win Rate :	**72.9%**		**77.1%**		**70.0%**		**68.6%**	

Recommendations

custom_score function is recommended based on playing results because -

1. It is relatively simple to implement.
2. If we would like to improve this function, we could tune using grid search for this function:
   `A*own_moves - B*opp_moves`
3. This function has the highest Win Rate.