Heuristic Analysis

----Heuristics----

1. Hybrid Strategy
2. Keep-Away-Or-Stick-To Strategy
3. Toward Freedom Strategy
4. Stick-To Strategy

----Brief Summary----

Best Heuristic: Keep-Away-Or-Stick-To Strategy

Reason:

- According to search depth, the difference between the four strategies is not that significant. Keep-Away-Or-Stick-To Strategy (2.) searches 1 depth deeper than Toward Freedom Strategy (3.) and also searches to the similar depth with Hybrid Strategy (1.). Yet, it searches 1 depth lesser than Stick-To Strategy. From search depths of the agent with each heuristics, it seems the complexities of the heuristics are similar. In terms of the performance, however, Keep-Away-Or-Stick-To Strategy often performs better than the agent with other 3 heuristics and also the agent with improved score. As a result, under the similar depth the agents searched, I would recommend to choose Keep-Away-Or-Stick-To Strategy to get higher chances of winning!
- Another point is that Keep-Away-Or-Stick-To Strategy is built on improved_score, which calculate the difference legal moves between the agent and the opponent. And, this strategy is derived from the notion that if the agent get more legal moves, choose the step away from the opponent which avoids the opponent's stepping into the agent's legal

moves. On the contrary, if the agent gets lesser legal moves, stick to the opponent in order to get more legal moves. From the results shown below (in 2.), the strategy showed a steadily higher chance of winning than the agent with improved_score. This would support my assumption above.

- Although the performance and the depth between the Hybrid Strategy and Keep-Away-Or-Stick-To Strategy are similar, the Hybrid Strategy is a bit of complex. Unlike Keep-Away-Or-Stick-To Strategy, which just calculate difference between legal moves and distance between players, it needs to do one more operation, that is, calculating the distance between the center and the agent. In terms of the code complexity, I would recommend to use Keep-Away-Or-Stick-To Strategy rather than Hybrid Strategy.

1. Heuristic: Hybrid strategy
    - Concept:
        - ✧ Choose the move which fits following constraints
            - ✓ Get around along the contour of the circle whose center is at the middle (3,3) and radius is 1.5
            - ✓ Get closer to the opponent
            - ✓ Get the move with more possible legal moves
    - Code Implementation:

```
    # Get the difference of legal moves between player and its opponent
    player_moves = game.get_legal_moves(player = player)
    opponent_moves = game.get_legal_moves(player = game.get_opponent(player))
    move_difference = len(player_moves) - len(opponent_moves)

    # get player's and opponent's locations
    player_x, player_y = game.get_player_location(player = player)
    opponent_x, opponent_y = game.get_player_location(player = game.get_opponent(player))

    # center of the game
    center_x, center_y = game.width / 2., game.height/2.


    return -1*float(abs(abs(player_x-center_x) + abs(player_y-center_y) - 1.5)) + \
    -1*float(abs(player_x-opponent_x) + abs(player_y-opponent_y)) + (move_difference)
```

- Result:
  - The chance of winning is 70% around.

		First Run				Second Run				Third Run			
Match #	Opponent	AB_Improved		AB_Custom		AB_Improved		AB_Custom		AB_Improved		AB_Custom	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	9	1	10	0	10	0	10	0	10	0
2	MM_Open	6	4	8	2	7	3	9	1	7	3	8	2
3	MM_Center	9	1	9	1	9	1	9	1	8	2	9	1
4	MM_Improved	9	1	7	3	7	3	6	4	9	1	9	1
5	AB_Open	7	3	5	5	6	4	4	6	4	6	6	4
6	AB_Center	6	4	7	3	7	3	7	3	5	5	3	7
7	AB_Improved	5	5	4	6	5	5	4	6	3	7	6	4
	Win Rate:	72.9%		70.0%		72.9%		70.0%		65.7%		72.9%	

- Discussion:
  - Implementing this heuristic, the agent beats the minimax opponents; however, it just only beats the opponent with alpha-beta pruning in 50% chance around.

2. Heuristic: Keep-Away-Or-Stick-To Strategy
   - Concept:
     - If the agent has more legal moves than the opponent, keep away from the opponent; otherwise, stick to the opponent.
     - As the agent has more legal moves, it should keep away from the opponent, which avoiding the opponent from stepping into the agent's legal moves. As the agent has less legal moves, it should follow the opponent in order to get more legal moves (because we know that the opponent has more legal moves here)

➢ Code Implementation:

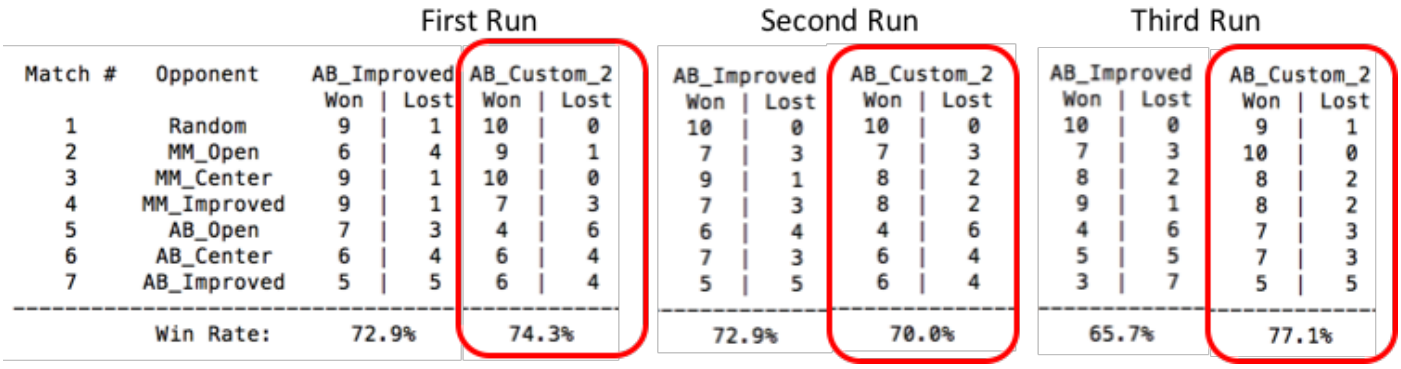
```
# step differences
player_moves = game.get_legal_moves(player = player)
opponent_moves = game.get_legal_moves(player = game.get_opponent(player))
move_difference = len(player_moves) - len(opponent_moves)

# location
player_x, player_y = game.get_player_location(player = player)
opponent_x, opponent_y = game.get_player_location(player = game.get_opponent(player))

if move_difference > 0:
    return (move_difference)*(float(abs(player_x-opponent_x) + abs(player_y-opponent_y)))
else:
    return abs(move_difference)*-1*(float(abs(player_x-opponent_x) + abs(player_y-opponent_y)))
```

➢ Result:

✧ The chance of winning is around 70~77%.



First Run

Match #	Opponent	AB_Improved Won	AB_Improved Lost	AB_Custom_2 Won	AB_Custom_2 Lost
1	Random	9	1	10	0
2	MM_Open	6	4	9	1
3	MM_Center	9	1	10	0
4	MM_Improved	9	1	7	3
5	AB_Open	7	3	4	6
6	AB_Center	6	4	6	4
7	AB_Improved	5	5	6	4
	Win Rate:	72.9%		74.3%	

Second Run

AB_Improved Won	AB_Improved Lost	AB_Custom_2 Won	AB_Custom_2 Lost
10	0	10	0
7	3	7	3
9	1	8	2
7	3	8	2
6	4	4	6
7	3	6	4
5	5	6	4
72.9%		70.0%	

Third Run

AB_Improved Won	AB_Improved Lost	AB_Custom_2 Won	AB_Custom_2 Lost
10	0	9	1
7	3	10	0
8	2	8	2
9	1	8	2
4	6	7	3
5	5	7	3
3	7	5	5
65.7%		77.1%	

➢ Discussion:

✧ With this heuristic, it seems that my agent could beat the opponent with alpha-beta pruning in higher chances than 50% (around 60~70%), which is much better than the hybrid heuristic (the one firstly shown in this analysis).

3. Heuristic: Toward Freedom Strategy

➢ Concept:

✧ Choose a future step that moves closer to the region with more empty spaces, which may indicate there will be legal moves at most of the time after choosing the move.

➢ Code implementation:

```
# get empty moves (exclude player's legal moves)
empty_spaces = np.array(list(set(game.get_blank_spaces()) - set(game.get_legal_moves(player = player))))
empty_space_center = empty_spaces.mean(axis = 0)

# get player's location
player_x, player_y = game.get_player_location(player = player)

return -1*float(abs(empty_space_center[0]-player_x) + abs(empty_space_center[1]-player_y))
```

✧ Get the blank spaces excluding current player's legal moves

✓ Calculate their center

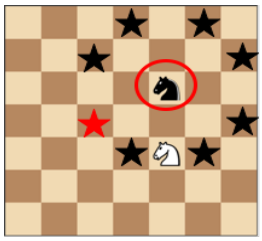
✧ Get the player's location (x, y)

✧ Return the distance between them (use absolute value to simplify calculation)

✧ Note: Use negative value here to let agent choose the move which is closer to the free space

➢ Visualization:

✧



Next Move	Empty Center	Distance	No. legal moves
(0,3)	(3.18, 2.97)	-3.2	3
(0,5)	(3.13, 2.84)	-5.29	2
(1,2)	(3.21, 3.14)	-3.36	5
(1,6)	(3.11, 2.81)	-5.29	2
(3,2)	**(3.00, 3.22)**	**-1.22**	**6**
(3,6)	(3.00, 2.79)	-3.2	2
(4,3)	(2.74, 2.97)	-1.28	7
(4,5)	(2.80, 2.73)	-3.46	5

Next Move	Empty Center	Distance	No. legal moves
(4,3)	**(2.62, 3.06)**	**1.43**	**5**
(6,3)	(2.77, 3.05)	3.27	1

➢ Result:

✧

****
Playing Matches
****

Match #	Opponent	AB_Improved Won	AB_Improved Lost	AB_Custom Won	AB_Custom Lost	AB_Custom_2 Won	AB_Custom_2 Lost	AB_Custom_3 Won	AB_Custom_3 Lost
1	Random	9	1	10	0	9	1	10	0
2	MM_Open	7	3	8	2	8	2	8	2
3	MM_Center	8	2	8	2	10	0	6	4
4	MM_Improved	5	5	7	3	8	2	7	3
5	AB_Open	4	6	6	4	4	6	3	7
6	AB_Center	8	2	6	4	5	5	5	5
7	AB_Improved	6	4	3	7	4	6	5	5
	Win Rate:	67.1%		68.6%		68.6%		62.9%	

- ✧ winning rate is around 68%

➢ Discussion:

- ✧ Compared to “Improved” heuristic, it seems this heuristic doesn’t choose a better move. This may indicate that although the agent makes a step toward the region with more free spaces, but sometimes there will be no legal moves around that center (empty space center). Choosing that move will trap itself.
- ✧ From the visualized board and char shown above, it can be said that this heuristic also chooses the move with more possible legal moves but may not the greatest possible legal moves. So, this heuristic won’t that powerful if face to the opponent with alpha-beta pruning.

5. Heuristic: Stick-To Strategy

➢ Concept:

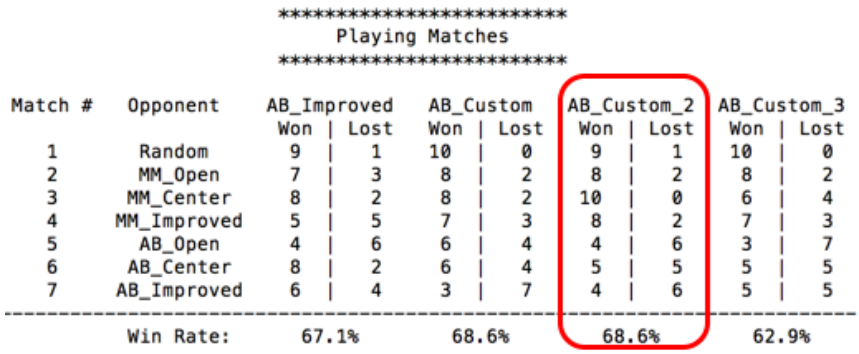
- ✧ Choose a move closer to the opponent, which may somehow step into opponent’s next legal moves or try to mimic opponent’s behavior

➢ Code Implementation:

```
player_x, player_y = game.get_player_location(player = player)
opponent_x, opponent_y = game.get_player_location(player = game.get_opponent(player))

return -1*float(abs(player_x-opponent_x) + abs(player_y-opponent_y))
```

➢ Result:



```
**************************
    Playing Matches
**************************
```

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	10	0	9	1	10	0
2	MM_Open	7	3	8	2	8	2	8	2
3	MM_Center	8	2	8	2	10	0	6	4
4	MM_Improved	5	5	7	3	8	2	7	3
5	AB_Open	4	6	6	4	4	6	3	7
6	AB_Center	8	2	6	4	5	5	5	5
7	AB_Improved	6	4	3	7	4	6	5	5
	Win Rate:	67.1%		68.6%		68.6%		62.9%	

✧ Wining rate is around 68%

➢ Discussion:

✧ Compared to the “Improved” heuristic, this heuristic seems to have equal chance of winning. It may suggest that sticking to the opponent will be a good strategy if the opponent does well under the same search strategy (alpha-beta pruning). However, it also indicates that if the opponent doesn’t do well, the agent implementing this heuristic doesn’t, either (shown by the competition versus AB_Open & AB_Center).