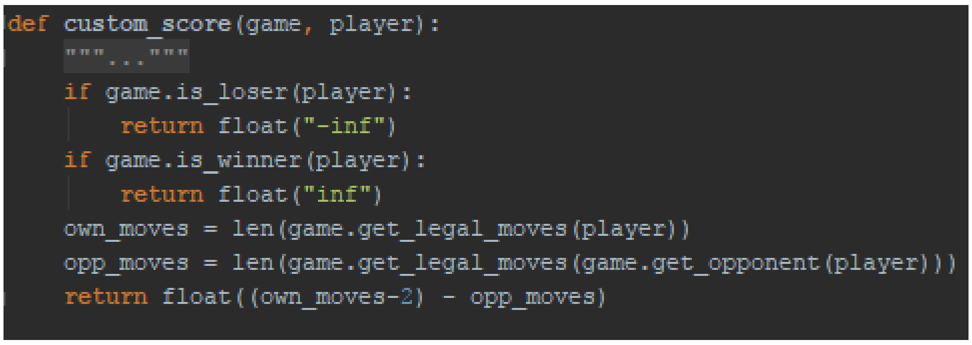
Heuristic Analysis By Ng Fang Kiang

Udacity – Artificial Intelligence Nanodegree (Class of 2017)

Heuristic 1:

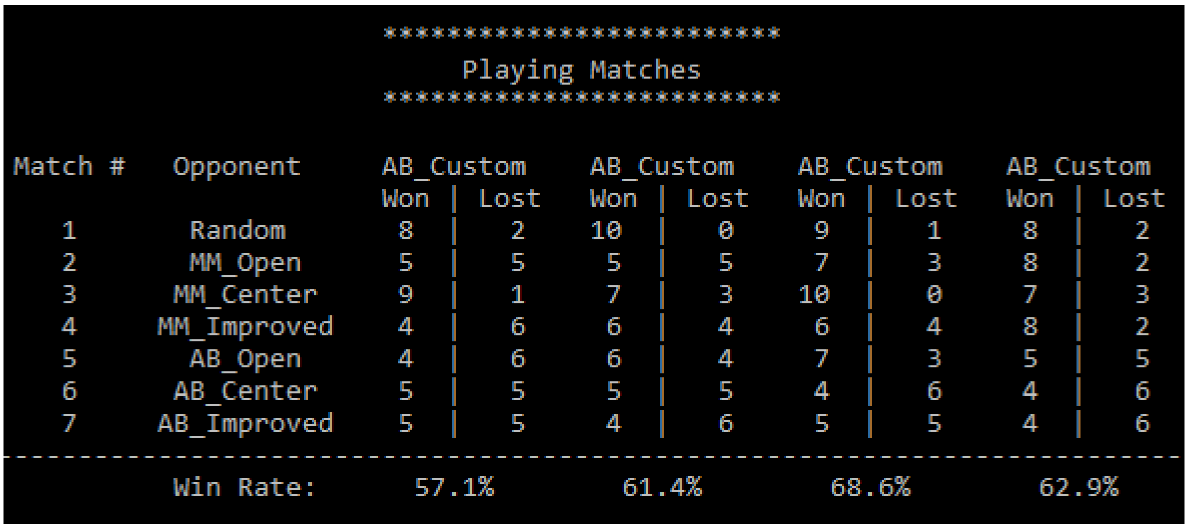
Left Heuristic: Remove two move from the difference in the number of moves available to the player and opponent.



```
def custom_score(game, player):
    """..."""
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    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)
```

Result 1:

This heuristic achieved highest win rate 68.6% compared to others.

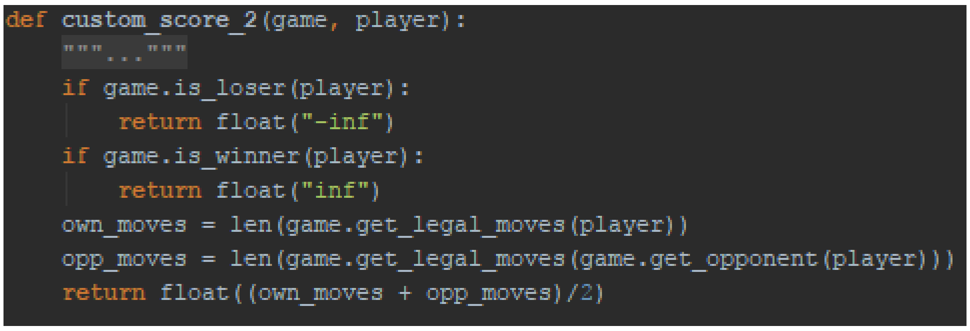


```
                         *************************
                              Playing Matches
                         *************************

Match #    Opponent      AB_Custom     AB_Custom     AB_Custom     AB_Custom
                         Won | Lost    Won | Lost    Won | Lost    Won | Lost
   1         Random       8  |   2     10  |   0      9  |   1      8  |   2
   2        MM_Open       5  |   5      5  |   5      7  |   3      8  |   2
   3       MM_Center      9  |   1      7  |   3     10  |   0      7  |   3
   4      MM_Improved     4  |   6      6  |   4      6  |   4      8  |   2
   5        AB_Open       4  |   6      6  |   4      7  |   3      5  |   5
   6       AB_Center      5  |   5      5  |   5      4  |   6      4  |   6
   7      AB_Improved     5  |   5      4  |   6      5  |   5      4  |   6
--------------------------------------------------------------------------
           Win Rate:        57.1%         61.4%         68.6%         62.9%
```

Heuristic 2:

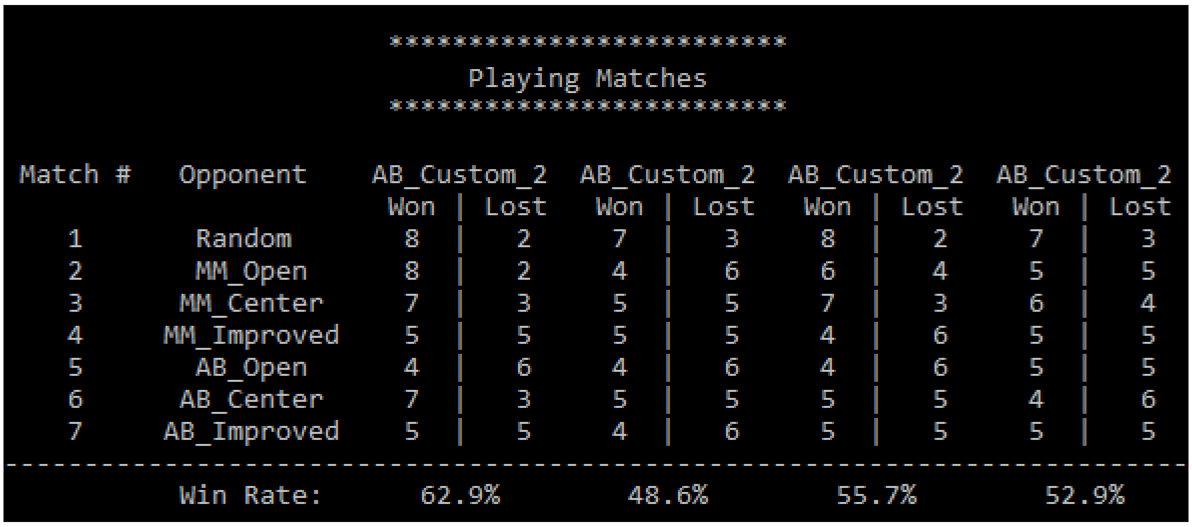
Average Heuristic: It state the average between numbers of own move and opponent move.



```
def custom_score_2(game, player):
    """..."""
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    own_moves = len(game.get_legal_moves(player))
    opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
    return float((own_moves + opp_moves)/2)
```

Result 2:

The result are not stable compare with other heuristic.

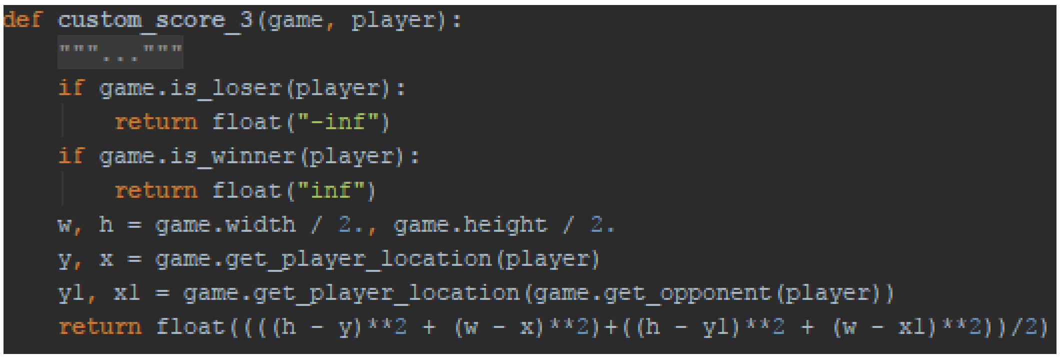


```
                        *************************
                             Playing Matches
                        *************************

 Match #     Opponent     AB_Custom_2   AB_Custom_2   AB_Custom_2   AB_Custom_2
                           Won | Lost    Won | Lost    Won | Lost    Won | Lost
    1         Random        8  |  2       7  |  3       8  |  2       7  |  3
    2         MM_Open       8  |  2       4  |  6       6  |  4       5  |  5
    3        MM_Center      7  |  3       5  |  5       7  |  3       6  |  4
    4       MM_Improved     5  |  5       5  |  5       4  |  6       5  |  5
    5         AB_Open       4  |  6       4  |  6       4  |  6       5  |  5
    6        AB_Center      7  |  3       5  |  5       5  |  5       4  |  6
    7       AB_Improved     5  |  5       4  |  6       5  |  5       5  |  5
--------------------------------------------------------------------------
           Win Rate:        62.9%         48.6%         55.7%         52.9%
```

Heuristic 3:

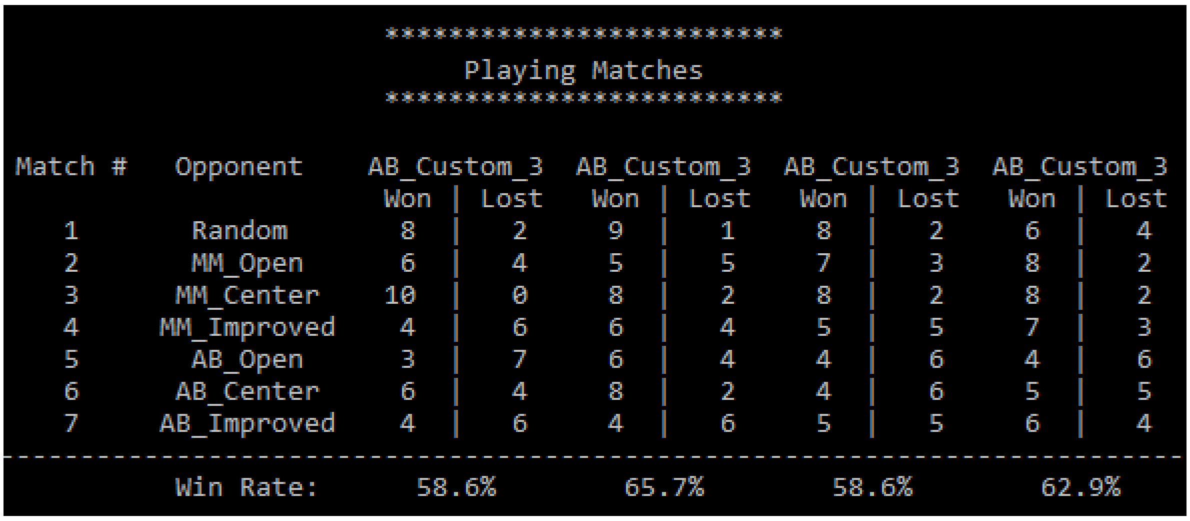
Average Center Heuristic: Outputs a score equal to square of the distance from the center of the board to the position of the player and opponent. It state the average between own location and opponent location.



```
def custom_score_3(game, player):
    """..."""
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    w, h = game.width / 2., game.height / 2.
    y, x = game.get_player_location(player)
    yl, xl = game.get_player_location(game.get_opponent(player))
    return float((((h - y)**2 + (w - x)**2)+((h - yl)**2 + (w - xl)**2))/2)
```

Result 3:

This produce the most stable win rate compare to others heuristic.



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

 Match #   Opponent    AB_Custom_3  AB_Custom_3  AB_Custom_3  AB_Custom_3
                        Won | Lost   Won | Lost   Won | Lost   Won | Lost
    1       Random       8  |  2      9  |  1      8  |  2      6  |  4
    2       MM_Open      6  |  4      5  |  5      7  |  3      8  |  2
    3      MM_Center    10  |  0      8  |  2      8  |  2      8  |  2
    4     MM_Improved    4  |  6      6  |  4      5  |  5      7  |  3
    5       AB_Open      3  |  7      6  |  4      4  |  6      4  |  6
    6      AB_Center     6  |  4      8  |  2      4  |  6      5  |  5
    7     AB_Improved    4  |  6      4  |  6      5  |  5      6  |  4
--------------------------------------------------------------------------
           Win Rate:      58.6%        65.7%        58.6%        62.9%
```