Heuristic Analysis

by Egor Ermilov

Heuristic 1

```
    if game.is_winner(player):
        return float("inf")
    elif game.is_loser(player):
        return float("-inf")

    return float(len(game.get_legal_moves(player)) -
len(game.get_legal_moves(game.get_opponent(player))))
```

This heuristic assumes the more available moves the player has, the better.

Heuristic 2

```
     if game.is_winner(player):
        return float("inf")
    elif game.is_loser(player):
        return float("-inf")

    return float(relative_distance_from_center(game,
game.get_player_location(player)) -
                 relative_distance_from_center(game,
game.get_player_location(game.get_opponent(player))))
```

This heuristic assumes the closer the player to the board center is, the better.

Heuristic 3

```
  if game.is_winner(player):
        return float("inf")
    elif game.is_loser(player):
        return float("-inf")

    player_moves = game.get_legal_moves(player)
    opponent_moves =
game.get_legal_moves(game.get_opponent(player))

    available_centrality_player =
sum(relative_distance_from_center(game, i_move) for i_move in
player_moves)
```

```
    available_centrality_opponent =
sum(relative_distance_from_center(game, i_move) for i_move in
opponent_moves)

    return float(available_centrality_player -
available_centrality_opponent)
```

This heuristic is a new measure – it sums the centrality measures for all the available moves. It combines the heuristics 1 and 2. The higher this new measure is, the better.

Performance

```
                    *************************
                         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     9  |   1     9  |   1    10  |   0
    2       MM_Open      4  |   6     5  |   5     5  |   5     8  |   2
    3      MM_Center     7  |   3     8  |   2     9  |   1    10  |   0
    4     MM_Improved    5  |   5     6  |   4     3  |   7     7  |   3
    5       AB_Open      5  |   5     5  |   5     5  |   5     6  |   4
    6      AB_Center     5  |   5     5  |   5     7  |   3     7  |   3
    7     AB_Improved    5  |   5     5  |   5     5  |   5     7  |   3
----------------------------------------------------------------------------
           Win Rate:      57.1%        61.4%        61.4%        78.6%

Process finished with exit code 0
```

We can see that the heuristic 3 has the best win rate, therefore it has been chosen for the final .score() function.