Heuristics Evaluation Review

Game Isolation

Heuristic Functions

`moves_ratio` : positively weighted scores if the player has more moves that of opponent and Negatively weighted scores if the opponent has more moves than that of the player.

`weighted_available_moves` : The legal moves available for the current player are weighted based on their position in the board. The moves are weighted proportionally with scores for each move decreasing the farther they are from the center of the board.

i.e. Suppose a move is made at the center of the board it is given a score of 4, a move made at a unit distance away from center of the board it is given a score of 3 and vice-versa.

`weighted_center_board_moves` : This is a slight variation of `weighted_available_moves` . Here, the moves at the center of the board are given a lot more weightage than the moves that are far away from the center of the board.

If the move is at the center of the board it is given a score of 8. And other consecutive moves that are away from the center are given a scores of 5 and downwards.

`improved_score_([0-9]+)_heuristic` : This is a variation of `improved_score` function. Here the no.of available opponent moves are given more weightage (say alpha) than the player's no.of available legal moves.

`improved_score_2_heuristic` : Here I used the same heuristics logic as mentioned above. With the value of `alpha = 1.5`

`improved_score_3_heuristic` : Here I used the same heuristics logic as mentioned above. With the value of `alpha = 2.0`

`improved_score_4_heuristic` : Here I used the same heuristics logic as mentioned above. With the value of `alpha = 2.5` .

TOURNAMENT - I

- Custom => `improved_score_heuristic` with alpha = 1.5
- Custom 2 => `improved_score_heuristic` with alpha = 2.0

- Custom 3 => `improved_score_heuristic` with alpha = 2.5

```
This script evaluates the performance of the custom_score evaluation
function 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      8  |   2     10  |   0     9  |   1     10  |   0
    2        MM_Open     8  |   2      5  |   5     8  |   2      7  |   3
    3       MM_Center    7  |   3      9  |   1     8  |   2      9  |   1
    4     MM_Improved    7  |   3      7  |   3     9  |   1      6  |   4
    5        AB_Open     6  |   4      4  |   6     7  |   3      4  |   6
    6       AB_Center    4  |   6      6  |   4     7  |   3      7  |   3
    7     AB_Improved    4  |   6      5  |   5     7  |   3      5  |   5
--------------------------------------------------------------------------
            Win Rate:      62.9%        65.7%        78.6%        68.6%
```

From the above tournament I found that `improved_score_heuristic` with `alpha = 2.0` has more win % than 1.5 and 2.5 alpha values.

Hence, I chose to use `improved_score_3_heuristic` as a heuristic function with alpha = 2.0

TOURNAMENT - II

- Custom => `moves_ratio`
- Custom 2 => `weighted_available_moves`
- Custom 3 => `improved_score_3_heuristic`

```
This script evaluates the performance of the custom_score evaluation
function 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     10  |   0     10  |   0      9  |   1
    2        MM_Open     7  |   3      9  |   1      6  |   4      9  |   1
    3       MM_Center    9  |   1      9  |   1      8  |   2      8  |   2
    4     MM_Improved    7  |   3      7  |   3      8  |   2      9  |   1
    5        AB_Open     4  |   6      5  |   5      5  |   5      6  |   4
    6       AB_Center    6  |   4      5  |   5      5  |   5      6  |   4
    7     AB_Improved    4  |   6      4  |   6      3  |   7      5  |   5
--------------------------------------------------------------------------
            Win Rate:      67.1%        70.0%        64.3%        74.3%
```

TOURNAMENT - III

- Custom => `moves_ratio`
- Custom 2 => `weighted_available_moves`
- Custom 3 => `weighted_center_board_moves`

```
This script evaluates the performance of the custom_score evaluation
function 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       9  |   1      9  |   1     10  |   0     10  |   0
    2       MM_Open      6  |   4      7  |   3      7  |   3      6  |   4
    3      MM_Center    10  |   0     10  |   0      7  |   3     10  |   0
    4     MM_Improved    5  |   5      8  |   2      6  |   4      7  |   3
    5       AB_Open      5  |   5      5  |   5      4  |   6      5  |   5
    6      AB_Center     7  |   3      5  |   5      4  |   6      5  |   5
    7     AB_Improved    6  |   4      5  |   5      4  |   6      4  |   6
----------------------------------------------------------------------------
          Win Rate:      68.6%        70.0%        60.0%        67.1%
```

TOURNAMENT - IV

- Custom => `moves_ratio`
- Custom 2 => `improved_score_3_heuristic`
- Custom 3 => `weighted_center_board_moves`

```
This script evaluates the performance of the custom_score evaluation
function 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     10  |   0      8  |   2      8  |   2
    2       MM_Open      7  |   3      8  |   2      8  |   2      7  |   3
    3      MM_Center    10  |   0     10  |   0      9  |   1      7  |   3
    4     MM_Improved    5  |   5      7  |   3      7  |   3      8  |   2
    5       AB_Open      7  |   3      6  |   4      5  |   5      3  |   7
    6      AB_Center     5  |   5      6  |   4      7  |   3      5  |   5
    7     AB_Improved    4  |   6      7  |   3      5  |   5      4  |   6
----------------------------------------------------------------------------
          Win Rate:      68.6%        77.1%        70.0%        60.0%
```

From the above 3 tournaments we can observe that `moves_ratio` has been outperforming all the other 2 heuristic functions consistently in the last 2 tournaments.

Hence, I chose my best heuristic function as `moves_ratio` in my final project submission