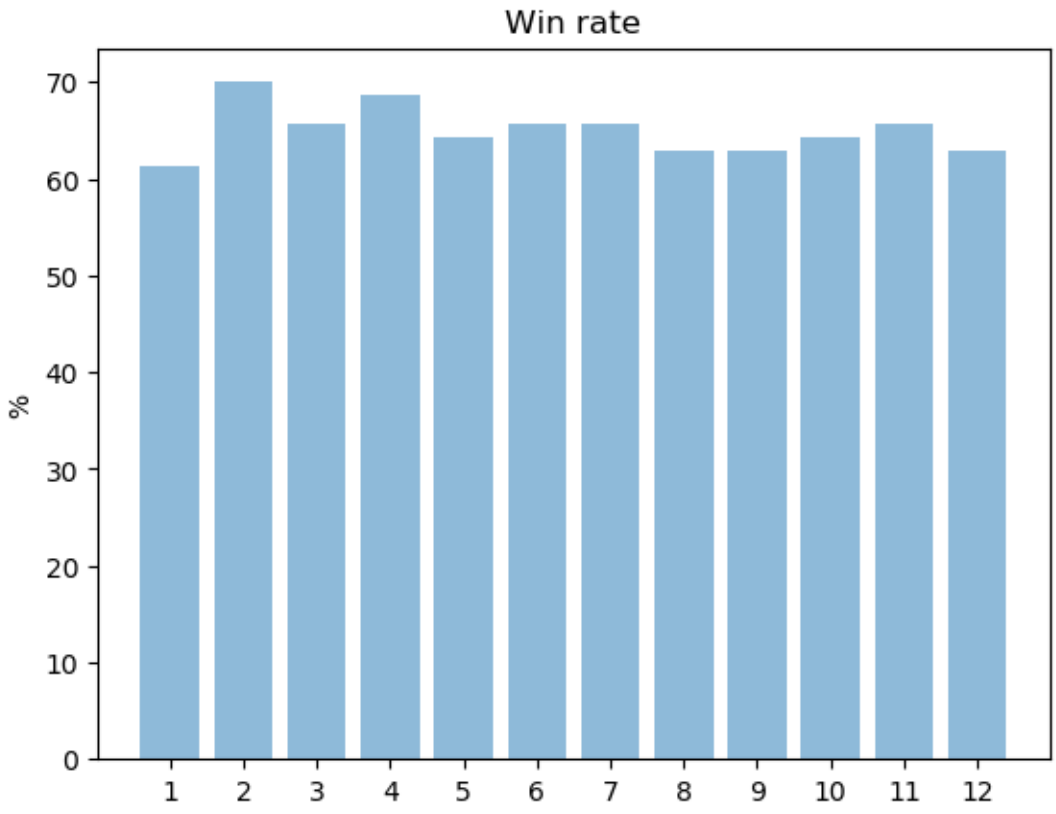
	Evaluation Function	Win Rate	Won Improved Win Rate
1	#My_moves – 2*#Oppenent_moves	61.4%	62.9% (NO)
2	#My_moves – #Oppenent_moves	70.0%	62.9% (YES)
3	#My_moves	65.7%	62.9% (YES)
4	#My_moves – 3*#Oppenent_moves	68.6%	64.3%(YES)
5	2*#My_moves – #Oppenent_moves	64.3%	64.3%(YES)
6	#My_moves^2 – 4*#Oppenent_moves	65.7%	64.3%(YES)
7	4*#My_moves – #Oppenent_moves^2	65.7%	64.3%(YES)
8	#My_moves^2 – #Oppenent_moves^2	62.9%	64.3%(NO)
9	#My_moves^2 – #Oppenent_moves^3	62.9%	64.3%(NO)
10	#My_moves – 10*#Oppenent_moves	64.3%	51.4%(YES)
11	4*#My_moves^4 – #Oppenent_moves^4	65.7%	51.4%(YES)
12	20*#My_moves – 10*#Oppenent_moves	62.9%	51.4%(YES)

- Evaluation function { #My_moves – 10*#Oppenent_mov } was the best evaluation function comparing it with ID_improved along with all The tournament opponents list (MM_Open, MM_Center, ..).

- Evaluation function { 20*#My_moves – 10*#Oppenent_mov} was the only evaluation function that won the ID_improved for all of The tournament opponents list (MM_Open, MM_Center, ..).

- Evaluation function { #My_moves – #Oppenent_moves} did make the highest win rate.


Win rate
%
0
10
20
30
40
50
60
70
1
2
3
4
5
6
7
8
9
10
11
12

```python
def custom_score(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    weight = 2
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    return my_moves - weight*opp_moves


def custom_score_2(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    return my_moves - opp_moves


def custom_score_3(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    return my_moves

```

```
                    *************************
                         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      8  |   2      9  |   1      6  |   4
    2        MM_Open      6  |   4      6  |   4      8  |   2      9  |   1
    3       MM_Center     5  |   5      7  |   3      8  |   2      8  |   2
    4     MM_Improved     8  |   2      6  |   4      7  |   3      8  |   2
    5        AB_Open      6  |   4      5  |   5      4  |   6      2  |   8
    6       AB_Center     7  |   3      5  |   5      7  |   3      7  |   3
    7     AB_Improved     3  |   7      6  |   4      6  |   4      6  |   4
----------------------------------------------------------------------------
           Win Rate:      62.9%        61.4%        70.0%        65.7%
```

```python
def custom_score(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    weight = 3
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    return my_moves - weight*opp_moves


def custom_score_2(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    weight = 2
    return weight*my_moves - opp_moves


def custom_score_3(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    exp = 2
    return my_moves**exp - 4*opp_moves
```

```
**************************
     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	8	2
2	MM_Open	8	2	7	3	5	5	9	1
3	MM_Center	7	3	9	1	10	0	8	2
4	MM_Improved	6	4	6	4	5	5	6	4
5	AB_Open	5	5	6	4	7	3	4	6
6	AB_Center	5	5	4	6	5	5	7	3
7	AB_Improved	5	5	6	4	4	6	4	6
	Win Rate:	64.3%		68.6%		64.3%		65.7%	

```python

def custom_score(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    exp = 2
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    return 4*my_moves - opp_moves**exp


def custom_score_2(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    exp = 2
    return my_moves**exp - opp_moves**exp


def custom_score_3(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    exp1 = 2
    exp2 = 3
    return my_moves**exp1 - opp_moves**exp2
```

```
                      *************************
                           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     9  |   1     8  |   2     8  |   2
    2       MM_Open      5  |   5     6  |   4     4  |   6     6  |   4
    3      MM_Center    10  |   0     7  |   3     8  |   2     7  |   3
    4     MM_Improved    7  |   3     7  |   3     8  |   2     9  |   1
    5       AB_Open      5  |   5     6  |   4     4  |   6     4  |   6
    6      AB_Center     6  |   4     7  |   3     8  |   2     6  |   4
    7     AB_Improved    4  |   6     4  |   6     4  |   6     4  |   6
----------------------------------------------------------------------------
           Win Rate:      64.3%        65.7%        62.9%        62.9%
```

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

    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    return my_moves - 10*opp_moves


def custom_score_2(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    exp = 4
    return 4*my_moves**exp - opp_moves**exp


def custom_score_3(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")
    my_moves = float(len(game.get_legal_moves(player)))
    opp_moves = float(len(game.get_legal_moves(game.get_opponent(player))))
    return 20*my_moves - 10*opp_moves
```

```
                    **************************
                         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      9   |   1     9   |   1    10   |   0
    2        MM_Open       5   |   5      6   |   4     4   |   6     5   |   5
    3       MM_Center      5   |   5      8   |   2     7   |   3     5   |   5
    4      MM_Improved     5   |   5      6   |   4     7   |   3     5   |   5
    5        AB_Open       5   |   5      7   |   3     8   |   2     8   |   2
    6       AB_Center      6   |   4      5   |   5     7   |   3     6   |   4
    7      AB_Improved     2   |   8      4   |   6     4   |   6     5   |   5
--------------------------------------------------------------------------------
           Win Rate:      51.4%        64.3%        65.7%        62.9%
```