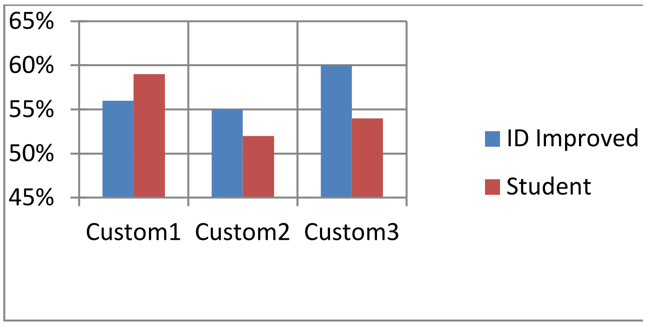
Heuristic Analysis

By Akash Chauhan



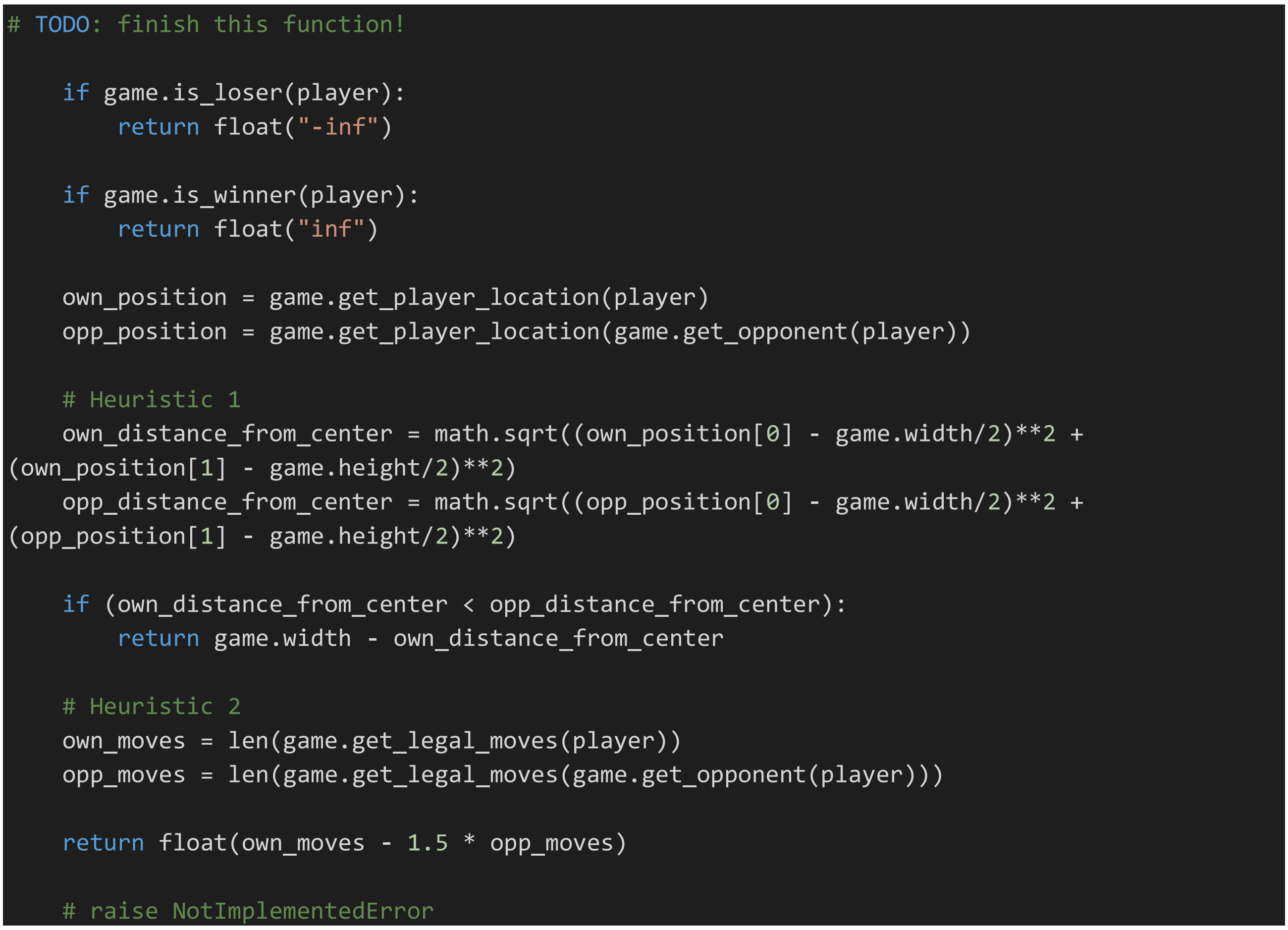
We have tested 3 types of heuristic functions and find the custom1 to be working best among the heuristics.

1. custom_score:
This heuristic is a combination of two heuristic function

a. center heuristic the player which is closer to the center can do better comparing to the case when he is far from the center

b. left legal moves: that states the condition of player comparing to the opponents 50% more moves.

Heuristic change point: when player is closer to the center than the opponent, if the player is far then the b heuristic comes into play.



```
# TODO: finish this function!

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

    if game.is_winner(player):
        return float("inf")

    own_position = game.get_player_location(player)
    opp_position = game.get_player_location(game.get_opponent(player))

    # Heuristic 1
    own_distance_from_center = math.sqrt((own_position[0] - game.width/2)**2 +
(own_position[1] - game.height/2)**2)
    opp_distance_from_center = math.sqrt((opp_position[0] - game.width/2)**2 +
(opp_position[1] - game.height/2)**2)

    if (own_distance_from_center < opp_distance_from_center):
        return game.width - own_distance_from_center

    # Heuristic 2
    own_moves = len(game.get_legal_moves(player))
    opp_moves = len(game.get_legal_moves(game.get_opponent(player)))

    return float(own_moves - 1.5 * opp_moves)

    # raise NotImplementedError
```

****
Evaluating: ID_Improved
****

```
Playing Matches:
----------
  Match 1: ID_Improved vs   Random        Result: 20 to 0
  Match 2: ID_Improved vs   MM_Null       Result: 16 to 4
  Match 3: ID_Improved vs   MM_Open       Result: 6 to 14
  Match 4: ID_Improved vs MM_Improved Result: 10 to 10
  Match 5: ID_Improved vs   AB_Null       Result: 13 to 7
  Match 6: ID_Improved vs   AB_Open       Result: 7 to 13
  Match 7: ID_Improved vs AB_Improved   Result: 7 to 13


Results:
----------
ID_Improved         56.43%

*************************
   Evaluating: Student
*************************

Playing Matches:
----------
  Match 1:   Student   vs   Random     Result: 19 to 1
  Match 2:   Student   vs   MM_Null           Result: 14 to 6
  Match 3:   Student   vs   MM_Open           Result: 13 to 7
  Match 4:   Student   vs MM_Improved       Result: 9 to 11
  Match 5:   Student   vs   AB_Null    Result: 11 to 9
  Match 6:   Student   vs   AB_Open           Result: 10 to 10
  Match 7:   Student   vs AB_Improved       Result: 7 to 13


Results:
----------
Student             59.29%
```

2. custom_score_2:
Center Heuristic: it states that the player near to the center has better chances to win the game.

```
def custom_score_2(game, player):
    if game.is_loser(player):
        return float("-inf")
    if game.is_winner(player):
        return float("inf")

    own_position = game.get_player_location(player)
    opp_position = game.get_player_location(game.get_opponent(player))

    # Center Heuristic
    own_distance_from_center = math.sqrt((own_position[0] - game.width/2)**2 +
(own_position[1] - game.height/2)**2)
    opp_distance_from_center = math.sqrt((opp_position[0] - game.width/2)**2 +
(opp_position[1] - game.height/2)**2)

    return game.width - own_distance_from_center
```

```
*************************
 Evaluating: ID_Improved
*************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random         Result: 19 to 1
  Match 2: ID_Improved vs   MM_Null        Result: 15 to 5
  Match 3: ID_Improved vs   MM_Open        Result: 8 to 12
  Match 4: ID_Improved vs MM_Improved Result: 10 to 10
  Match 5: ID_Improved vs   AB_Null        Result: 11 to 9
  Match 6: ID_Improved vs   AB_Open        Result: 8 to 12
  Match 7: ID_Improved vs AB_Improved   Result: 7 to 13


Results:
----------
ID_Improved        55.71%

*************************
   Evaluating: Student
*************************

Playing Matches:
----------
  Match 1:   Student   vs   Random     Result: 20 to 0
  Match 2:   Student   vs   MM_Null            Result: 16 to 4
  Match 3:   Student   vs   MM_Open            Result: 10 to 10
  Match 4:   Student   vs MM_Improved       Result: 9 to 11
  Match 5:   Student   vs   AB_Null    Result: 8 to 12
  Match 6:   Student   vs   AB_Open            Result: 6 to 14
  Match 7:   Student   vs AB_Improved        Result: 5 to 15


Results:
----------
Student          52.86%
```

custom_score_3:

Legal moves left heuristic: this one is intuitive, one with the more moves available has more chances to win.

```
def custom_score_3(game, player):
    if game.is_loser(player):
        return float("-inf")

    if game.is_winner(player):
        return float("inf")

    #  2
    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)
```

```
*************************
 Evaluating: ID_Improved
*************************

Playing Matches:
----------
  Match 1: ID_Improved vs   Random        Result: 20 to 0
  Match 2: ID_Improved vs   MM_Null       Result: 20 to 0
  Match 3: ID_Improved vs   MM_Open       Result: 8 to 12
  Match 4: ID_Improved vs MM_Improved     Result: 10 to 10
  Match 5: ID_Improved vs   AB_Null       Result: 10 to 10
  Match 6: ID_Improved vs   AB_Open       Result: 9 to 11
  Match 7: ID_Improved vs AB_Improved     Result: 7 to 13


Results:
----------
ID_Improved        60.00%

*************************
   Evaluating: Student
*************************

Playing Matches:
----------
  Match 1:   Student   vs   Random    Result: 19 to 1
  Match 2:   Student   vs   MM_Null           Result: 14 to 6
  Match 3:   Student   vs   MM_Open           Result: 11 to 9
  Match 4:   Student   vs MM_Improved         Result: 8 to 12
  Match 5:   Student   vs   AB_Null   Result: 11 to 9
  Match 6:   Student   vs   AB_Open           Result: 9 to 11
  Match 7:   Student   vs AB_Improved         Result: 4 to 16


Results:
----------
Student          54.29%
```