The main implementation of all three custom 1,2 and 3 area as follows:

```
min_open_move_score = 0.0
min_improved_score  = -7.0
min_center_distance = 0.5
max_open_move_score = 7.0
max_improved_score = 7.0
max_center_distance = 40.5

_open_move_score = open_move_score(game, player)
_improved_score = improved_score(game, player)
_center_distance = center_score(game, player)

if _open_move_score < 0:
    _open_move_score
elif _open_move_score > 7:
    _open_move_score = 7

if _improved_score < -7:
    _improved_score = -7
elif _improved_score > 7:
    _improved_score = 7

if _center_distance < 0.5:
    _center_distance = 0.5
elif _center_distance > 40.5:
    _center_distance = 40.5

_open_move_score = get_equal_scale(_open_move_score, min_open_move_score,
max_open_move_score)
_improved_score = get_equal_scale(_improved_score, min_improved_score,
max_improved_score)
_center_distance = get_equal_scale(_center_distance, min_center_distance,
max_center_distance)
```

then for custom 1 I got as follows:

```
score = ((_open_move_score) ** 2 + (_improved_score) ** 2 +
(_center_distance * 0.5) ** 2) ** (0.5)
```

for custom 2 I got as follows:

```
score = ((_open_move_score) ** 2 + (_improved_score) ** 2 +
(_center_distance) ** 2) ** (0.5)
```

and for custom 3 I got as follows:

```
score = (_open_move_score + _improved_score + _center_distance )
```

the results are as folows:

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	10	0	10	0
2	MM_Open	7	3	8	2	8	2	7	3
3	MM_Center	10	0	10	0	10	0	10	0
4	MM_Improved	7	3	9	1	6	4	8	2
5	AB_Open	4	6	7	3	5	5	4	6
6	AB_Center	5	5	9	1	7	3	7	3
7	AB_Improved	6	4	5	5	2	8	2	8
	Win Rate:	68.6%		82.9%		68.6%		68.6%	

The process for getting the best score was iterative. I got the max and min of :

```
_open_move_score
_improved_score
_center_distance
```

then I evaluate a vector normalization. I adjust a variable as 0.5 that gave me the best score I could got