Improved_score

From a heuristic: $h = \ own_{moves} - \ opp_{moves}$

Results: ID_Improved 75.71% Student 79.29 %

Calculate % of change with: $\frac{Student_h – ID\_Improved_h}{ID\_Improved_h}$

Student is 4.73% better than ID_Improved.

ID_Improved and Student use the same strategy. Student has 4.73% of score more than ID_improved. I don't think this 4.73% is a significant of difference.

```
if game.is_loser(player):
    return float("-inf")
if game.is_winner(player):
    return float("inf")
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
return float(own_moves - opp_moves)
```

Improved_score 2x weighted to opponent

From a heuristic: $h = own_{moves} - \ 2opp_{moves}$

Results: ID_Improved 77.86% Student 81.43%

Calculate % of change with: $\frac{Student – ID\_Improved}{ID\_Improved}$

Student is 4.59% better than ID_Improved.

I multiply Opponent's moves with 2; Make my agent more aggressive. The result is Student has 4.59% of score more than ID_improved. It's the same as before, no improvement.

```
If game.is_loser(player):
    return float(“-inf”)
if game.is_winner(player):
    return float(“inf”)
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)
```

Staying close to the centre

From a heuristic: $\frac{game_{width} + game_{height}}{2} - \sqrt[2]{(location_x - \frac{game_{width}}{2})^2 + (location_y - \frac{game_{height}}{2})^2}$

I try to calculate a different between the average of the size of the board and the distance from the centre of the board. (The closer to the middle, the more score)

Results: ID_Improved 80.00% Student 73.57%

Calculate % of change with: $\frac{Student – ID\_Improved}{ID\_Improved}$

Student is 8.04% worse than ID_Improved.

I think because the piece moves like a knight, a strategy to stick close to the centre is not fit as the game that the piece moves like a queen.

```
if game.is_loser(player):
    return float("-inf")
if game.is_winner(player):
    return float("inf")
centre_width = game.width / 2.0
```

```
centre_height = game.height / 2.0
player_location = game.get_player_location(player)

close_to_centre = (game.width + game.height) / 2.0 - pow(pow((current_location[0] - center_height), 2) +
                                                        pow((current_location[1] - center_width), 2), 0.5)

return close_to_centre
```

Diff percentage weight

From a heuristic: $100\ \frac{own_{moves}-opp_{moves}}{own_{moves}+opp_{moves}}$

Results: ID_Improved 76.43% Student 79.29%

Calculate % of change with: $\frac{Student\ –ID\_Improved}{ID\_Improved}$

Student is 3.61 % better than ID_Improved.

I try to calculate heuristic by weight the different between the own moves and the opponent's moves. The result is Student has 3.61% of score more than ID_improved. It's the same as Improved_score heuristic, no significant improvement or degrades.

```
if game.is_loser(player):
    return float("-inf")
if game.is_winner(player):
    return float("inf")
 own_moves = len(game.get_legal_moves(player))
 opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
 return 100.0 * (own_moves - opp_moves) / (own_moves + opp_moves)
```

Diff percentage weight with 2x opponent weight

From a heuristic: $100\ \frac{own_{moves}-2opp_{moves}}{own_{moves}+opp_{moves}}$

Results: ID_Improved 75.71% Student 83.57%

Calculate % of change with: $\frac{Student\ –ID\_Improved}{ID\_Improved}$

Student is 10.38% better than ID_Improved.

I multiply the opponent's moves of the previous heuristic to make my agent more aggresive. The result is satisfied. Student has 10.38% better than ID_Improved.

```
if game.is_loser(player):
    return float("-inf")
if game.is_winner(player):
    return float("inf")
own_moves = len(game.get_legal_moves(player))
opp_moves = len(game.get_legal_moves(game.get_opponent(player)))
 return 100.0 * (own_moves - 2 * opp_moves) / (own_moves + opp_moves)
```

Visualisation

The table below shows winning rates of ID_Improved which always uses Improved_score heuristics, used for a base line; winning rates of my agent with each heuristics; and improvements between each heuristics and the baseline.

	Improved_score	Improved_score 2x weighted to opponent	Staying close to the centre	Diff percentage weight	Diff percentage weight with 2x opponent weight
ID_Improved (Base line)	75.71%	77.86%	80.00%	76.43%	75.71%
Student	79.29%	81.43%	73.57%	79.29%	83.57%
Improvement	4.73%	4.59%	-8.04%	3.61%	10.38%

Conclusion

The best result is Diff percentage weight with 2x opponent weight. Because it has got 10.38% of improvement over the base line, whilst each of other heuristics has less than 5% of improvement.

One odd thing I usually found is my agent always looses to the random agent one time (Win 19/20), whilst the baseline rarely gets the score (19/20 over the random), but I don’t see any time-out message.

The weight is like a magic number. I’ve tried to adjust the value of the opponent weight of Diff percentage weight heuristic to 2.5, but the result is the same as the weight of 2. The appropriate value needs optimising further.