Heuristic Analysis

In [34]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from IPython.display import display, HTML

groups = 7
heuristic_agent_values = []
id_improved_values = []
labels = ('Random', 'MM_Null', 'MM_Open', 'MM_Improved', 'AB_Null',
'AB_Open', 'AB_Improved')

index = np.arange(groups)
bar_width = 0.35

def plot_graphs_tables(id_improved_values, versus_values, heursitic_values,
versus_values_h, label):
    data = {'ID_Imrpoved Agent Wins (Out of 20)': id_improved_values, 'Heur
istic Agent Wins (Out of 20)': heursitic_values}
    df = pd.DataFrame(data)
    display(df)
    id_improved_values = id_improved_values
    versus_values = versus_values
    plt.bar(index, id_improved_values, bar_width,
                     color='b',
                     label='ID_Improved Agent')

    plt.bar(index + bar_width, versus_values, bar_width,
                     color='r')

    plt.xlabel('Agents')
    plt.ylabel('Wins')
    plt.title('ID_Improved Agent Vs Other Agents')
    plt.xticks(index + bar_width / 2, labels)
    plt.rcParams["figure.figsize"] = [20, 5]
    plt.legend()
    plt.show()

    heursitic_values = heursitic_values
    versus_values = versus_values_h

    plt.bar(index, heursitic_values, bar_width,
                     color='b',
                     label='ID_Improved Agent')

    plt.bar(index + bar_width, versus_values, bar_width,
                     color='r')

    plt.xlabel('Agents')
    plt.ylabel('Wins')
    plt.title(label)
    plt.xticks(index + bar_width / 2, labels)
    plt.legend()
    plt.legend()
    plt.rcParams["figure.figsize"] = [20, 5]
    plt.show()
```

Heuristic I - Aggresively Chase Opponent

This Heuristic mainly focus on opponent moves aggressively. It calculates the difference between the number of moves for the player and twice the number of moves for the opponent.

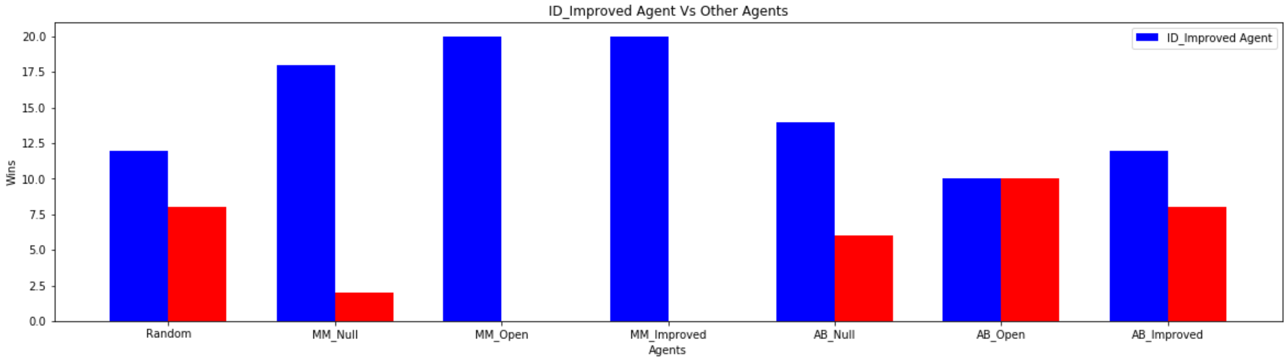
Strength Rating for ID_Improved Agent - 75.71%

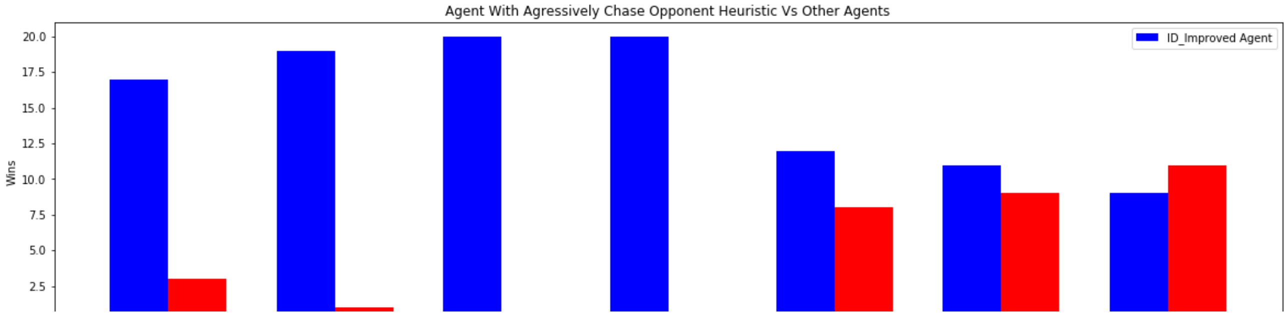
Strength Rating for Agent with Aggresively Chase Opponent Heuristic - 77.14%

In [35]:

```
id_improved_values = [12, 18, 20, 20, 14, 10, 12]
versus_values = [8, 2, 0, 0, 6, 10, 8]
heursitic_values = [17, 19, 20, 20, 12, 11, 9]
versus_values_h = [3, 1, 0, 0, 8, 9, 11]
label = 'Agent With Agressively Chase Opponent Heuristic Vs Other Agents'
plot_graphs_tables(id_improved_values, versus_values, heursitic_values, ver
sus_values_h, label)
```

	Heuristic Agent Wins (Out of 20)	ID_Imrpoved Agent Wins (Out of 20)
0	17	12
1	19	18
2	20	20
3	20	20
4	12	14
5	11	10
6	9	12





Heuristic II - Check Near Walls

The "Check Near Walls" evaluation function calculates a cumulative score based on the moves and their positions. A cumulative score is calculated for both the players. A positive score is added to the cumulative score for the player if the board is less than 50% occupied and the moves lie on the walls, in case the board is between 50% and 85% occupied a higher score is subtracted from the sum. If the occupancy is greater than 85% even more higher score is subtracted The process is negated in case of the opponent. The difference between both the player cumulative scores is added to the number difference of non-corners move for both players and the value returned

Strength Rating for ID_Improved Agent - 77.14%

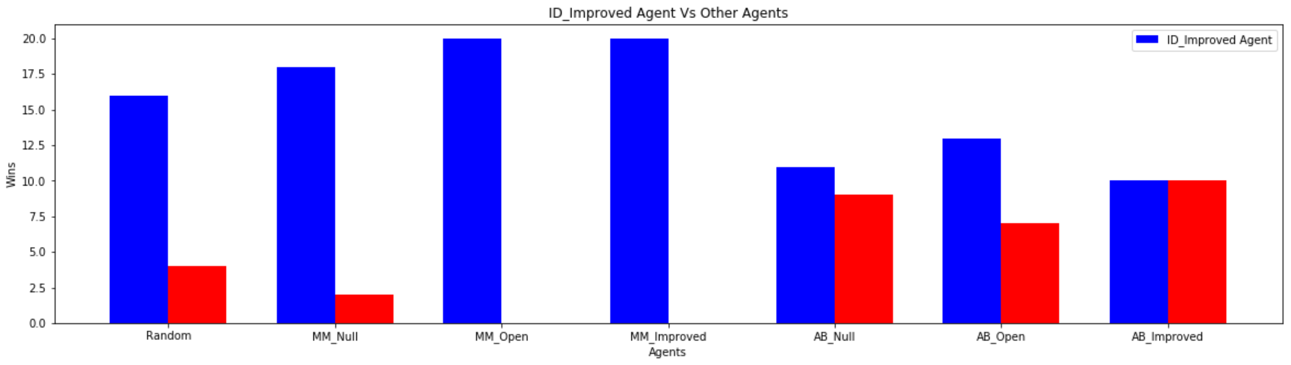
Strength Rating for Agent with Aggresively Chase Opponent Heuristic - 80.71%

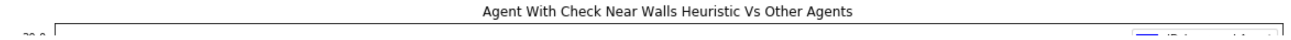
Looking at the results, it seems like avoiding moves near the walls when the board is close to getting full gets a better strength rating.

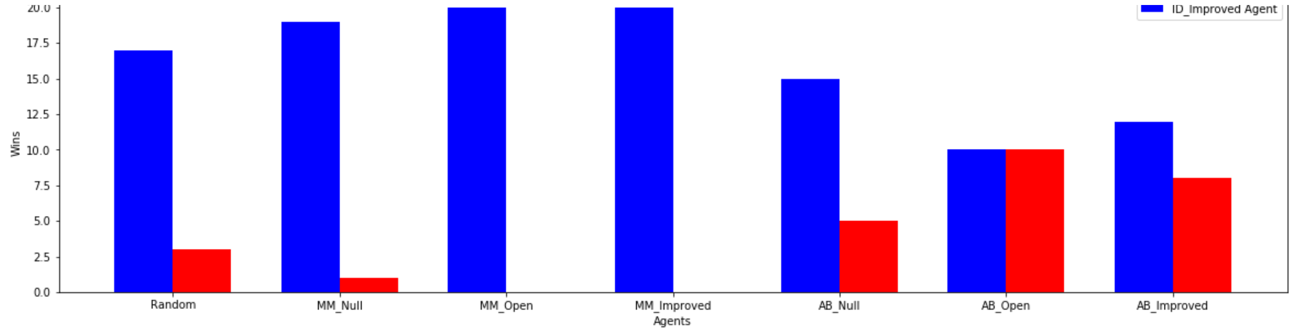
In [36]:

```
id_improved_values = [16, 18, 20, 20, 11, 13, 10]
versus_values = [4, 2, 0, 0, 9, 7, 10]
heursitic_values = [17, 19, 20, 20, 15, 10, 12]
versus_values_h = [3, 1, 0, 0, 5, 10, 8]
label = 'Agent With Check Near Walls Heuristic Vs Other Agents'
plot_graphs_tables(id_improved_values, versus_values, heursitic_values, ver
sus_values_h, label)
```

	Heuristic Agent Wins (Out of 20)	ID_Imrpoved Agent Wins (Out of 20)
0	17	16
1	19	18
2	20	20
3	20	20
4	15	11
5	10	13
6	12	10







Heuristic III - Check In Corners

The "Check In Corners" evaluation function calculates a cumulative score based on the moves and their positions. A cumulative score is calculated for both the players. A positive score is added to the cumulative score for the player if the board is less than 60% occupied and the moves lie in the corners, in case the board is more than 60% occupied a higher score is subtracted from the sum. The process is negated in case of the opponent. The difference between both the player cumulative scores is added to the number difference of non-corners move for both players and the value returned

Strength Rating for ID_Improved Agent - 75.00%

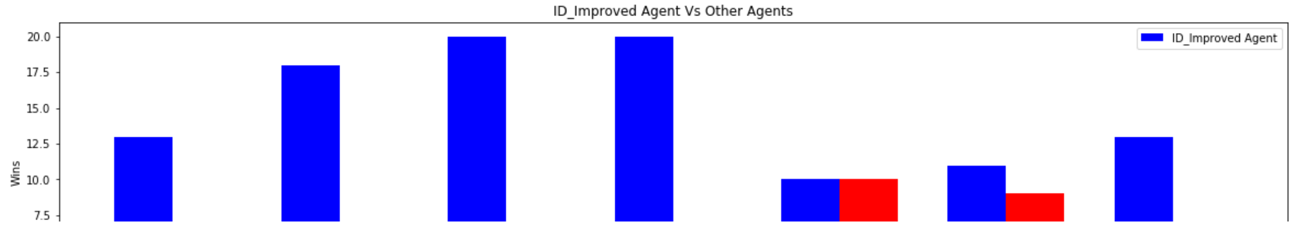
Strength Rating for Agent with Aggresively Chase Opponent Heuristic - 83.14%

Looking at the results, it seems like avoiding moves near the corners when the board is close to getting full gets a better strength rating.

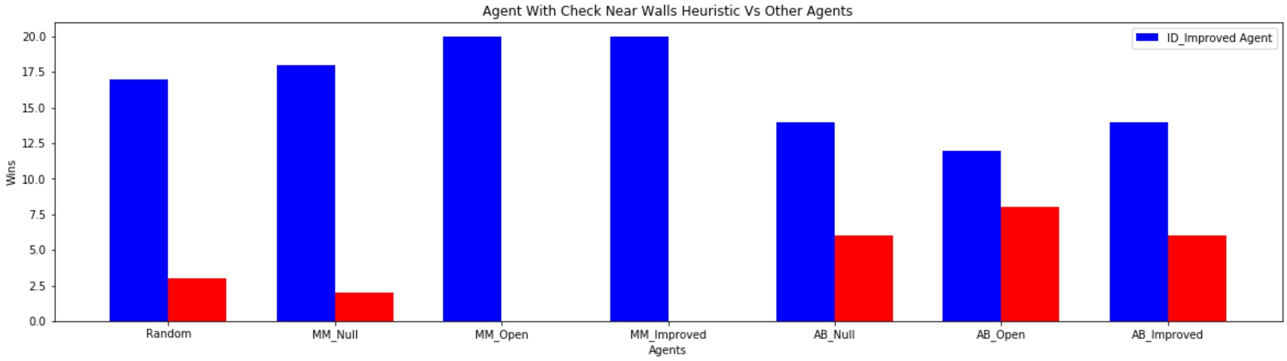
In [37]:

```
id_improved_values = [13, 18, 20, 20, 10, 11, 13]
versus_values = [7, 2, 0, 0, 10, 9, 7]
heursitic_values = [17, 18, 20, 20, 14, 12, 14]
versus_values_h = [3, 2, 0, 0, 6, 8, 6]
plot_graphs_tables(id_improved_values, versus_values, heursitic_values, ver
sus_values_h, label)
```

	Heuristic Agent Wins (Out of 20)	ID_Imrpoved Agent Wins (Out of 20)
0	17	13
1	18	18
2	20	20
3	20	20
4	14	10
5	12	11
6	14	13



5.0
2.5
0.0
Random
MM_Null
MM_Open
MM_Improved
AB_Null
AB_Open
AB_Improved
Agents


Agent With Check Near Walls Heuristic Vs Other Agents

Heuristic IV - Check Near Walls And Corners

The "Check Near Walls And Corners" evaluation function calculates a cumulative score based on the moves and their positions. A cumulative score is calculated for both the players. The method calculates the score for corners using 'evaluator_check_in_corners', and the score for walls using 'evaluator_check_near_walls'. The walls score is multiplied by 0.3 and corners score is multiplied y 0.7 and their sum is returned.

Strength Rating for ID_Improved Agent - 75.00%

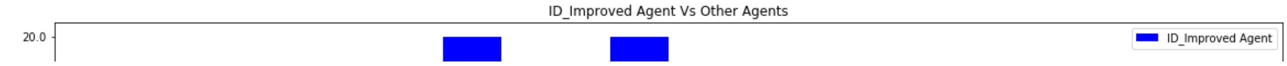
Strength Rating for Agent with Aggresively Chase Opponent Heuristic - 85.71%

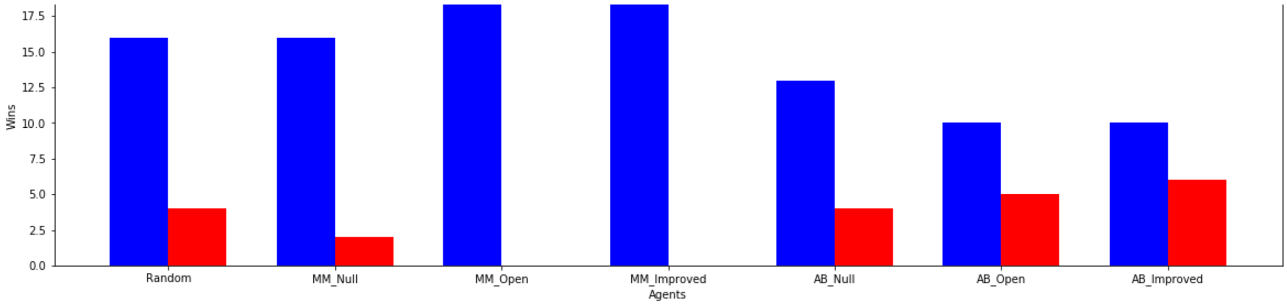
Looking at the results, it seems like avoiding moves near the walls as well as corners when the board is close to getting full gets a better strength rating.

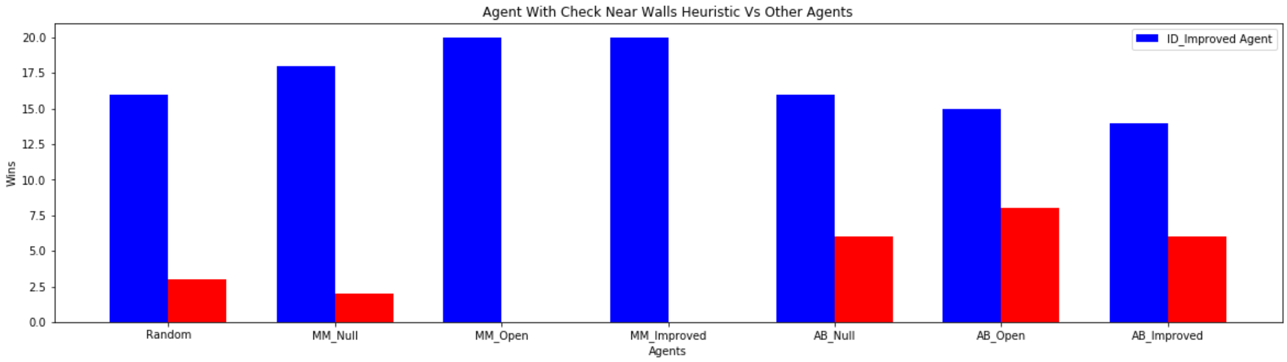
In [38]:

```
id_improved_values = [16, 16, 20, 20, 13, 10, 10]
versus_values = [4, 4, 0, 0, 7, 10, 10]
heursitic_values = [16, 18, 20, 20, 16, 15, 14]
versus_values = [4, 2, 0, 0, 4, 5, 6]
plot_graphs_tables(id_improved_values, versus_values, heursitic_values, ver
sus_values_h, label)
```

	Heuristic Agent Wins (Out of 20)	ID_Imrpoved Agent Wins (Out of 20)
0	16	16
1	18	16
2	20	20
3	20	20
4	16	13
5	15	10
6	14	10


ID_Improved Agent Vs Other Agents





Conclusion

Out of the four heuristics, The Check Near Walls And Corners seems to give the best strength rating. I chose this heuristic for the final use.

In the Check Near Walls Heuristic, I have observed that avoiding walls when board is getting full gave good performace, and In Check In Corners Heuristic, I have observed that avoiding corners when board is getting full gave good performance, so I have combined these two heuristics in order to yield even better performance.

The performance for this heursitic is also pretty consistent.

This is the reason for choosing 'Check Near Walls And Corners' as the final evaluation function.