REPORT

Option 1: Advanced Heuristic

Custom Player is implemented to search for the next best move using : Alpha - Beta pruning with iterative deepening

Two Heuristics implemented

- Standard Heuristic
- Custom Heuristic to check for states away from the edges of the board

Scenario 1(a): Custom Agent with Standard Heuristic, against Random Agent

In [33]:

```
%run run_match.py -f -r 100 -o RANDOM -p 4
```

```
Running 200 games:
+++++-++++--+++++++++-+++++-+++-+++++-+++++-+++-++++++++++++++++++
+++++++-++++++++++++++++++-++++++++--++-++++++++--++++++++++++++
+++++++++++++-++--++++++++++++++++++++++++++++++++-+++++++++--++
+++------++++++++++++++++++++++++++-++++
Outcome of Fair Matches:
Running 200 games:
+++++++++---++++++++++++++++++++++++++-++++++++++++++--++++++++-
++++++++++++++++++++++--+++++++++-++-+++++--++++-++++++++++++++
+++++++-+-+++--++++++++-++++++++++++++++++---+++++++++++++-+++++
+-++++++++++++--+-++++++++-+-++++++++-
Your agent won 86.8% of matches against Random Agent
```

Scenario 1(b): Custom Agent with Custom Heuristic, against Random Agent

In [28]:

```
%run run_match.py -f -r 100 -o RANDOM -p 4
```

```
Running 200 games:
++++++++++++-+-++++++++++++++-+-++++++++++++++++++++++++++-++--++
+++++-++++++++++++++++--+---++++++++++++++-++++++-+++++--++++++-++
+++-+++++---+++--++++++++++++++++-+-++--++++++++-+++++-+--+++++++++
++++++++++++--++++++++++++++++-+++++++++-
Outcome of Fair Matches:
Running 200 games:
+++-++-++++++++---+++++++++-++++++++++++++++++-++++++++++--++-++++
++++++++++++++++++-++-+++-++++++--++++++++-++++++++++++++++++++++
++++++++++++-++++++++++++++++++-+++++-+++++--+++++-+++++++-+++++++
+++-++-++++++++++++++++++++++++++++-++++++++++++-++
Your agent won 86.5% of matches against Random Agent
```

Scenario 2(a): Custom Agent with Standard Heuristic, against Greedy Agent

In [26]:

```
%run run_match.py -f -r 100 -o GREEDY -p 4
```

```
Running 200 games:
+-+++++--++--+-+-+--++++---+-++++++--++---+-+--++++-+++++--+------
-++-+--++---+-++++++++++-++---++++++---++++-++-+++-+++-+++-+++--
--++++-++----++++++--+++--------+++++--+++-+++-++--++++++++++-+++
++--+-+-++-++-+++++++--+-+--++-++++++++---+
Outcome of Fair Matches:
Running 200 games:
++-++-+++-+-+++--+++-++++--+-++++---+--+++++-+-++++++++-++--+--+--
-+--+-++--+++++--+---+++---+---++++-+-+++++-++++++++++++++-----++-
--++++--+---++++--++++++----++++++++++++-+-----++++-+--++++++++-+-
-+++++++-+--+----++++--++++++--+--++++++-++--+
Your agent won 62.0% of matches against Greedy Agent
```

Scenario 2(b): Custom Agent with Custom Heuristic, against Greedy Agent

In [29]:

```
%run run_match.py -f -r 100 -o GREEDY -p 4
```

```
Running 200 games:
+-++++--+---++++++-++-++++++++++-+++++---++-+++++-+++++++-++---+++
---++++++++---+---++++--++-+++++++++-+++++--++---++-++--++++-++-+--
--+-+---++++++++++-+++++++++--++++++----+++-+----+--+--+++---++++
++++++-+-++++-+++--+++++++-+++++++----+++---++-+++++++++++++++
Outcome of Fair Matches:
Running 200 games:
----++-+++--+----+-+++--+--++++-+++++++++-+++--+-++++-++---++++++
--+++-++++-++-+---+-++--++-+++++++++--+---++++---++++++++-++++++--+
---+++-++++++---+++--++-----+-++++++--+-++---+-+++++---+++++++-+++
++++++--+---+++++-+-+++--+-+-+++--+--+----
Your agent won 65.0% of matches against Greedy Agent
```

Scenario 3(a): Custom Agent with Standard Heuristic, against Minimax Agent

In [27]:

```
%run run_match.py -f -r 100 -o MINIMAX -p 4
```

```
Running 200 games:
+--+--+-+--+---------------++--+-+-++--++-------+------+++----+--
+--------+--++---++---+++++--++-+--+--+--++-+---+--+-+-+---+++-++
+++--++---+-----+--+----------+++++--------+--+++-------+++++++-++
--+--++++++-+++-+-+-++++---+++----++
Outcome of Fair Matches:
Running 200 games:
-+---++--++-------+--------++---+++------++------------++---+-+-+-
--+++-+---------------+--+-++---++------+++----+++---+-+++-+-+-+-
-------+--++--++-++--+----+----+--++---++---+++--++-+-+++----++--+
---+-++--++---+--+-++----+-++--+--+----++-
Your agent won 38.5% of matches against Minimax Agent
```

Scenario 3(b): Custom Agent with Custom Heuristic, against Minimax Agent

In [31]:

```
%run run_match.py -f -r 100 -o MINIMAX -p 4
```

```
Running 200 games:
--++++---+--+++----+++-+++------+++-+--++------+-+++-++------++---
--++--+-+++-++++---++-+--+---+-++-+--+----+--++++++-+-++++----+-++
+-+++----+-++-+----+--++-++++-+--+--++-++-----+--+--++--++--+-+--
-++-+---+-++----++-----++---+----++---+++----+--
Outcome of Fair Matches:
Running 200 games:
-----+--++-+++------+-+++--++--+--+---------+--+---------+--++----
++---++---+-++--++-+-+---+-+---+++--++-+-+++---------++---++-+--+-
+--+---++--++-------+++--------++------+-++--++----+------++--++
---------+++----------+--+---+---++++++-+++-+
Your agent won 39.2% of matches against Minimax Agent
```

Advanced Heuristic Analysis

In [19]:

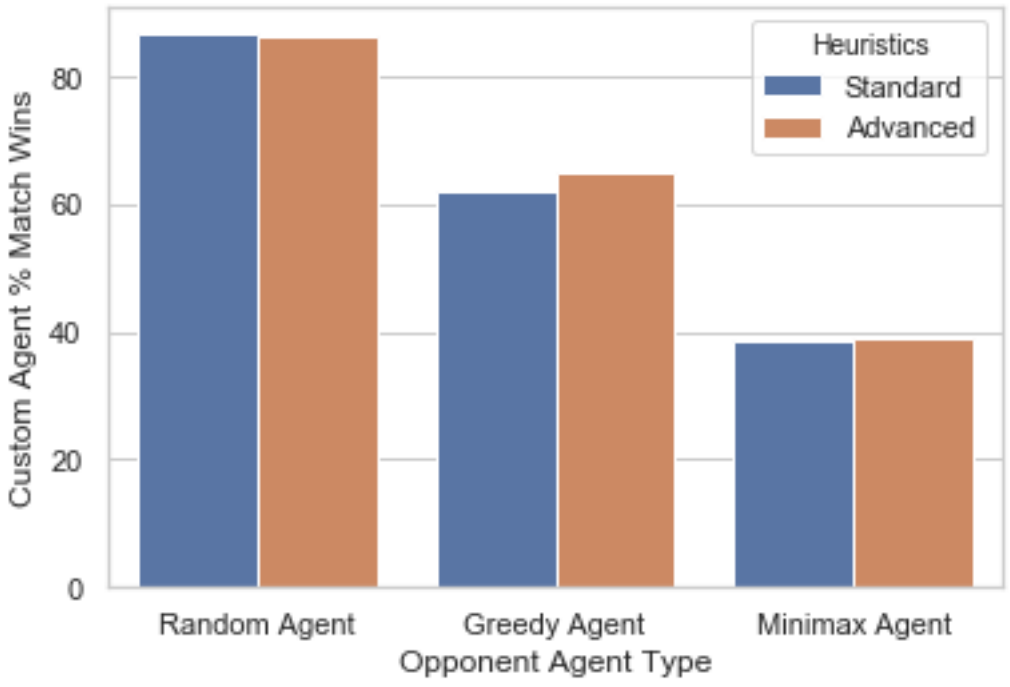
```
# Generating report chart using seaborn

import pandas as pd
import seaborn as sns

# Baseline: Standard Heuristic with fair_matches flag enabled


match_wins = pd.DataFrame({
    'Opponent Agent Type': ['Random Agent', 'Random Agent', 'Greedy Agent', 'G
reedy Agent', 'Minimax Agent', 'Minimax Agent'],
    'Custom Agent % Match Wins' : [86.8, 86.5, 62.0, 65.0, 38.5, 39.2], # Data
from outcome of Scenarios 1, 2, 3 above
    'Heuristics' : ['Standard', 'Advanced', 'Standard', 'Advanced', 'Standard'
, 'Advanced']
})

sns.set(style="whitegrid")
ax = sns.barplot(x="Opponent Agent Type", y="Custom Agent % Match Wins", hue =
'Heuristics', data=match_wins)
```



- Above graph shows my custom agent 'Percent Match Wins' against different opponent agent types.
- The games were played first with Standard Heuristic and then with the Advanced Heuristic
- For each of the scenarios, a total of 200 games were played.

- What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during search?

My custom heuristic incorporates choosing locations on the board away from the borders. Since, the more the knight remains in the centre board , the greater would be chances for more number of liberties at each state. Having more liberties than the opponent is one of the key factors to make sure that my player remains in the game.

- Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

My custom agent achieves a search depth of 5. Search depth limit greater than 5, runs the risks of taking longer than the default time constraint placed on each move. Heuristic accuracy is more important than search speed when there is a time constraint for making the move.

Option 2: Opening book

- Open book data built using MiniMax Search with alpha-beta pruning and Custom Heuristic, created as part of Advance Heuristic Analysis above
- Depth search limit of 5

Code Output: open_book.py

In [30]:

```
%run open_book.py
```

```
Enter open_book __main__
Round : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 2
3 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 4
5 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 6
7 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 8
9 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 10
8 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124
125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 14
1 142 143 144 145 146 147 148 149 150
Open Book Data for data.pickle file
Book State: Isolation(board=350296210042700169735167700547604 47, p
ly_count=4, locs=(101, 99))
Action: -15
Book State: Isolation(board=350302548295701310882175184063631 35, p
ly_count=3, locs=(101, 110))
Action: -11
Book State: Isolation(board=350327901307705875470205118127738 87, p
ly_count=2, locs=(112, 110))
Action: -11
Book State: Isolation(board=415231612039391220826836322242990 07, p
ly_count=0, locs=(None, None))
Action: 112
Book State: Isolation(board=363308643454042944541531358950789 11, p
ly_count=1, locs=(112, None))
Action: 110
```

Scenario 4(a) Custom Agent using data from opening book, against Random Agent. Without Fair Matches

In [14]:

```
%run run_match.py -r 80 -o RANDOM -p 4
```

```
Running 160 games:
++++++++++++++++++++++++++++++++++++++++++++++++++---++++++++-++
++++++++++++++++++-++++++++++++++++++++++++++++++-++++++++++++
++++++++++++++++-++++++++++++++++++++++++++++++++--++++++-++
+++++++++++++++
Your agent won 95.6% of matches against Random Agent
```

Scenario 4(b) Custom Agent using data from opening book, against Random Agent. With Fair Matches

In [16]:

```
%run run_match.py -f -r 80 -o RANDOM -p 4
```

```
Running 160 games:
++++++++++++++++++++++++++++++++++++++++++++++--++++++++++++++++++
++++++++++++++++++++++++--++++++++++++-++++++++--++++++++++++++
++++++++++++++++++++++--++++++++++-++++++++++++-+++++++-++++++-+++
++++-++-++++
Outcome of Fair Matches:
Running 160 games:
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++-+-++++++++++++++++-++++++++++++++++-+-+++--++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++--+++
++++++++++++--++++++-+++
Your agent won 92.8% of matches against Random Agent
```

Scenario 5(a) Custom Agent using data from opening book, against Greedy Agent. Without Fair Matches

In [20]:

```
%run run_match.py -r 80 -o GREEDY -p 4
```

```
Running 160 games:
++-+++++-++++++++++++--++++++++++-++-++++++++++++++++-+++-++++
+++++-++++++++++++++++++++++--++++-+-+-++++++++++-++-+++++++++
+++++++-++-++-++++++++++++++++--++++++++++++++++++--++++++++++-+
+-+++++++
Your agent won 85.6% of matches against Greedy Agent
```

Scenario 5(b) Custom Agent using data from opening book, against Greedy Agent. With Fair Matches

In [24]:

```
%run run_match.py -f -r 80 -o GREEDY -p 4
```

```
Running 160 games:
++++++++++++++-+++++++-+++++++-+++++++-+++-+-+-++++++++++++++--+
++++---++++---++++++++++++++++++++++++-++++++++++-+++++-++++++++++
--+++++-++++++++-+++++++-+++++++++-+++++++-++++++++++++++--++-++
+
Outcome of Fair Matches:
Running 160 games:
+++--++++-----++++++---+-++++++++++--++---+++++++++--++++++++--++-
-+-+--+++++-++++-++++--+++-++++++-+++++-++++++++++++---++-+++++-++++
++++++++--+++++++-++++++++++++++++-+--++++--++-++++++++++++++++++
+++-+-+++-+++
Your agent won 80.3% of matches against Greedy Agent
```

Scenario 6(a) Custom Agent using data from opening book, against Minimax Agent. Without Fair Matches

In [25]:

```
# Disable fair_matches to create initial baseline

%run run_match.py -r 80 -o MINIMAX
```

```
Running 160 games:
--++--+++-----+--++-+--+--+---+++++-++++++++--------++-++++++--++
+++++---------++-++++-++++++--++----+++----++------+++++---++++--
++++----+---++-+------------++--++-+-+---++++--++++++++++++++++-
---++--+++-++++++--+--++---++-++++
Your agent won 50.0% of matches against Minimax Agent
```

Scenario 6(b) Custom Agent using data from opening book, against Minimax Agent. With Fair Matches

In [26]:

```
#opening moves selected from data.pickle. fair_matches enabled.

%run run_match.py -f -r 80 -o MINIMAX
```

```
Running 160 games:
--++++++++--+----++------++++--++-----++--++--+--++++++---++-+--+-
---++---++--+-----+-+++-++---++++---+++-++++++++---+++---+------
+-------++++++-++------++--+-++++++++--+----++--++-+++----+-++----
+++++--+++------+--++--------+++-++-
Outcome of Fair Matches:
Running 160 games:
-+++--++--++--+----++--++++--++----------++---++-+--+-++++++-+-+-
-++++-+++++-+++--+++--+++++++-+--------++------++-----++-----++++
++++++------+-+---++++++-++++++++++-----++++---++---++++------+
+++-+--------+-----++----+-----+
Your agent won 47.5% of matches against Minimax Agent
```

Opening Book Analysis

In [27]:

```
# Generating report chart using seaborn

import pandas as pd
import seaborn as sns

# Baseline: Matches played against other agents not using opening book


match_wins = pd.DataFrame({
    'Opponent Agent Type': ['Random Agent', 'Random Agent', 'Greedy Agent', 'G
reedy Agent', 'Minimax Agent', 'Minimax Agent'],
    'Custom Agent % Match Wins' : [95.6, 92.8, 85.6, 80.3, 50.0, 47.5], # Data
from outcome of Scenarios 4, 5, 6 above
    'Match Style' : ['No Fair Matches', 'Fair Matches', 'No Fair Matches', 'Fa
ir Matches', 'No Fair Matches', 'Fair Matches']
})

sns.set(style="whitegrid")
ax = sns.barplot(x="Opponent Agent Type", y="Custom Agent % Match Wins", hue =
'Match Style', data=match_wins)
```

- Above graph shows 'Percent Match Wins'for my custom agent, that selects opening moves from the opening book, against different opponent agent types that do not use opening book.
- My custom agent uses the advanced heuristic for all the matches.
- The games were played both with and without Fair Matches.
- For each of the scenarios, a total of 160 games were played.

- Describe your process for collecting statistics to build your opening book. How did you choose states to sample? And how did you perform rollouts to determine a winner?

The data is built using Alpa-Beta pruning with Iterative Deepening. States which have led to the most win counts, for the number of rounds, in which the search ran were selected as winners.

- What opening moves does your book suggest are most effective on an empty board for player 1 and what is player 2's best reply?

From the above data it is clear that the close Player 1 is to the centre board, the higher are it's chances of winning.
