# HEURISTIC ANALYSIS

For an Adversarial Game Playing Agent for Isolation

Submitted by
Govardhan Padmanabhan
(17069496/1 | WID170715)

as part of WID3009: Artificial Intelligence Game Programming

# TABLE OF CONTENTS

# SYNOPSIS

Isolation is a two-player game in which alternate players move a single piece from one cell to another on a board. Whenever a piece occupies a cell, the said cell becomes unavailable to use for the rest of the game. The goal of the game is to leave the opponent with no remaining legal moves. The first player with no remaining cells loses, and the opponent wins.

In this project, each piece is similar to a knight in chess, i.e., restricted to L-shape movement, can jump over blocked or occupied cells, and can only move to open cells that are 2-rows & 1-column away or 2-columns &1-row away from their current cell.

Additionally, the agents will have a fixed time limit for searching the move and responding for each turn. If the time limit is expired, the player forfeits and the opponent wins the match.

The aim of this project is to develop an adversarial search agent to play the game "Isolation". This project report focuses on the heuristics to be used in A* Search for minimax and alpha-beta pruning.

The code in the `game_agent.py` file, specifically the `custom_score()` function is modified, and the agent is evaluated by modifying and running the `tournament.py` file

# CUSTOM HEURISTICS

A basic analysis of the agent implementation is that it returns the difference between the total moves in the second game state of the current player and that of the opponent. Additionally, an “aggressiveness” factor is assumed, in the form of a 1:2 ratio (player : opponent)

The pseudocode for the heuristics is as follows:

```
For each legal moves in the current state of player do:
    Get the next game state
    For each legal moves in the next game state of player do:
        Get the next, next game state
        Add length of all legal moves to player score
    End for
End for

For each legal moves in the current state of opponent do:
    Get the next game state
    For each legal moves in the next game state of opponent do:
        Get the next, next game state
        Add length of all legal moves to opponent score
    End for
End for

Return player score -  2 * opponent score
```

This custom heuristic implementation is run under the agent name of “WID170715_Implementation”

# PERFORMANCE

The `tournament.py` script is used to evaluate the effectiveness of heuristic. The script measures the relative performance of players in a round-robin tournament against several other predefined agents.

Note that the performance also depends on the hardware, as faster hardware can search deeper in a short amount of time. So, when run another time, run a different time, or run on different hardware, the evaluation score can change.

Three sets of experiments have been conducted: standalone, against standard opponents, and against the custom heuristics from the rest of my teammates

## Standalone

Here, we just run the custom heuristics to get a good picture of the performance.

To get more samples, the number of matches were increased to 500. The time limit remains the same at 150ms.

The resulting evaluation score of the implementation is 70.15%

The raw evaluation results can be found in Appendix A: Standalone Evaluation.

## Against Standard Opponents

Here, we run the custom heuristics along with other heuristic components already provided in the project.

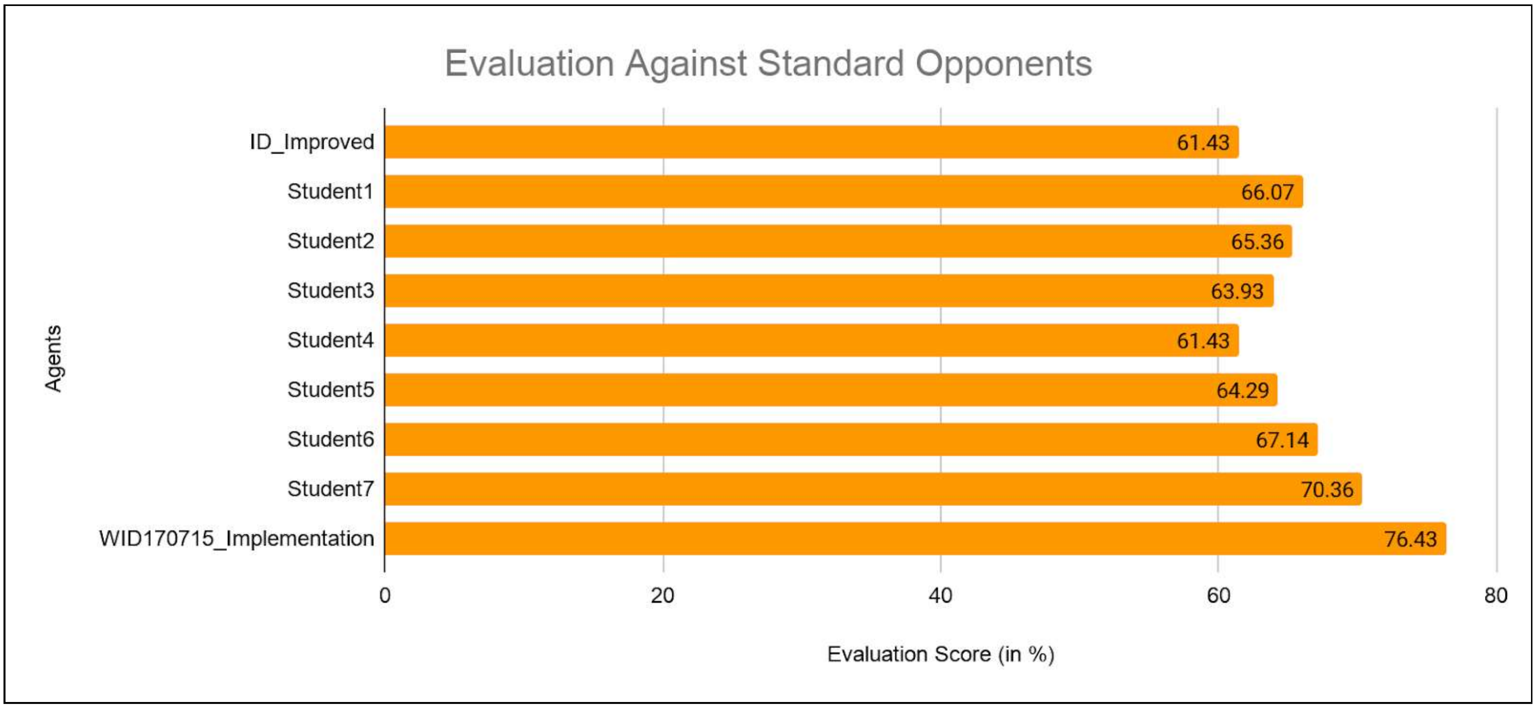
For faster evaluation (due to hardware constraints), the number of trials is set 10. The time limit remains the same, at 150 ms.

The heuristics and agent names used in this evaluation are:

| *Agent Name* | *Heuristic Function* |
|---|---|
| ID_Improved | improved_score() |
| Student1 | aggressive_heuristic() |
| Student2 | defensive_heuristic() |

| | |
|---|---|
| Student3 | maximizing_win_chances_heuristic() |
| Student4 | minimizing_losing_chances_heuristic() |
| Student5 | chances_heuristic() |
| Student6 | weighted_chances_heuristic() |
| Student7 | weighted_chances_heuristic_2 |
| WID170715_Implementation | custom_score() |

The evaluation comparison between the aforementioned agents and the custom heuristic discussed in this report is shown in this chart:



The raw evaluation results can be found in Appendix B: Evaluation Against Standard Opponents

## Against Teammates

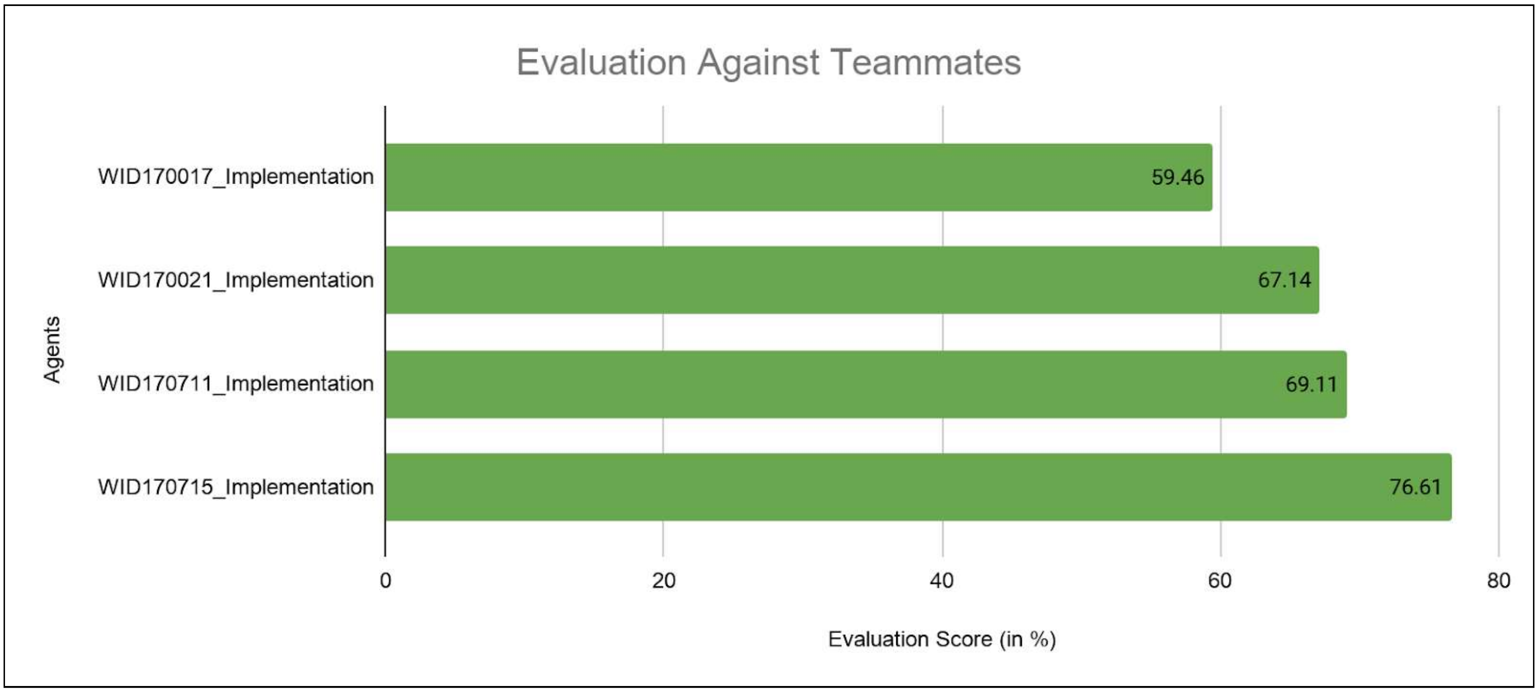
Here, we run the custom heuristic implementation against my teammates' heuristic implementations.

To minimize the number of variables affecting the performance, the heuristics are run on a single machine (WID170715, Govardhan's system). The number of trials are 20, and the time limit is 150 ms.

The teammates and agent names are:

| *Matric Number* | *Name* | *Agent Name* |
|---|---|---|
| 17195727 \| WID170017 | Jasherr Ravindran | WID170017_Implementation |
| 17107141 \| WID170021 | Kuganeswaran Letshimanan | WID170021_Implementation |
| 17129042 \| WID170711 | Azraf Kabir | WID170711_Implementation |
| 17069496 \| WID170715 | Govardhan Padmanabhan | WID170715_Implementation |

The evaluation comparisons between the heuristics are shown in this chart:



The raw evaluation results can be found in Appendix C: Evaluation Against Teammates.

# CONCLUSION

In this report, the custom heuristic implementation is explained. Discussed is the brief description of the method and the pseudocode. The custom implementation is a two-ply lookahead score, where it returns the difference between the total number of moves in the second ply of the current player, and that of the opponent. Additionally, an aggressive factor is included, with a 2:1 ratio applied to the opponent's moves.

To evaluate the custom implementation, three sets of experiments were conducted. When run alone with 500 matches, it got a result of 70%. When run along with the included standard opponent agents, at 10 matches, the custom implementation got a final score of 76.43%. When run along with the implementations of teammates, at 20 matches, the custom implementation got a score of 76.61%.

Since the custom heuristic implementation scored high and better in all scenarios, this method is highly recommended. Additionally, these experiments were conducted on a moderately powerful system. However, if run on a more powerful hardware platform, the results could possibly easily reach high 70s or even low to mid 80s.

# APPENDICES

## APPENDIX A
## Standalone Evaluation

This script evaluates the performance of the custom heuristic function by comparing the strength of an agent using iterative deepening (ID) search with alpha-beta pruning against the strength rating of agents using other heuristic functions. The `ID_Improved` agent provides a baseline by measuring the performance of a basic agent using Iterative Deepening and the "improved" heuristic (from lecture) on your hardware. The `Student` agent then measures the performance of Iterative Deepening and the custom heuristic against the same opponents.

```
*************************
Evaluating: WID170715_Implementation
*************************

Playing Matches:
----------
  Match 1:      WID170715_Implementation  vs     Random         Result: 1824 to 176
  Match 2:      WID170715_Implementation  vs     MM_Null        Result: 1752 to 248
  Match 3:      WID170715_Implementation  vs     MM_Open        Result: 1029 to 971
  Match 4:      WID170715_Implementation  vs  MM_Improved       Result: 964 to 1036
  Match 5:      WID170715_Implementation  vs     AB_Null        Result: 1595 to 405
  Match 6:      WID170715_Implementation  vs     AB_Open        Result: 1452 to 548
  Match 7:      WID170715_Implementation  vs  AB_Improved       Result: 1205 to 795


Results:
----------
WID170715_Implementation     70.15%
```

# APPENDIX B

## Evaluation Against Standard Opponents

This script evaluates the performance of the custom heuristic function by comparing the strength of an agent using iterative deepening (ID) search with alpha-beta pruning against the strength rating of agents using other heuristic functions. The `ID_Improved` agent provides a baseline by measuring the performance of a basic agent using Iterative Deepening and the "improved" heuristic (from lecture) on your hardware. The `Student` agent then measures the performance of Iterative Deepening and the custom heuristic against the same opponents.

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

Playing Matches:
----------
  Match 1:        ID_Improved  vs     Random          Result: 35 to 5
  Match 2:        ID_Improved  vs     MM_Null         Result: 28 to 12
  Match 3:        ID_Improved  vs     MM_Open         Result: 22 to 18
  Match 4:        ID_Improved  vs  MM_Improved        Result: 20 to 20
  Match 5:        ID_Improved  vs     AB_Null         Result: 22 to 18
  Match 6:        ID_Improved  vs     AB_Open         Result: 24 to 16
  Match 7:        ID_Improved  vs  AB_Improved        Result: 21 to 19


Results:
----------
ID_Improved       61.43%

*************************
  Evaluating: Student1
*************************

Playing Matches:
----------
  Match 1:        Student1  vs     Random             Result: 35 to 5
  Match 2:        Student1  vs     MM_Null            Result: 32 to 8
  Match 3:        Student1  vs     MM_Open            Result: 21 to 19
  Match 4:        Student1  vs  MM_Improved           Result: 20 to 20
  Match 5:        Student1  vs     AB_Null            Result: 31 to 9
  Match 6:        Student1  vs     AB_Open            Result: 24 to 16
  Match 7:        Student1  vs  AB_Improved           Result: 22 to 18


Results:
----------
```

```
Student1        66.07%

*************************
  Evaluating: Student2
*************************

Playing Matches:
----------
  Match 1:        Student2  vs     Random         Result: 38 to 2
  Match 2:        Student2  vs     MM_Null        Result: 33 to 7
  Match 3:        Student2  vs     MM_Open        Result: 18 to 22
  Match 4:        Student2  vs  MM_Improved       Result: 19 to 21
  Match 5:        Student2  vs     AB_Null        Result: 26 to 14
  Match 6:        Student2  vs     AB_Open        Result: 22 to 18
  Match 7:        Student2  vs  AB_Improved       Result: 27 to 13


Results:
----------
Student2        65.36%

*************************
  Evaluating: Student3
*************************

Playing Matches:
----------
  Match 1:        Student3  vs     Random         Result: 33 to 7
  Match 2:        Student3  vs     MM_Null        Result: 25 to 15
  Match 3:        Student3  vs     MM_Open        Result: 21 to 19
  Match 4:        Student3  vs  MM_Improved       Result: 24 to 16
  Match 5:        Student3  vs     AB_Null        Result: 29 to 11
  Match 6:        Student3  vs     AB_Open        Result: 24 to 16
  Match 7:        Student3  vs  AB_Improved       Result: 23 to 17


Results:
----------
Student3        63.93%

*************************
  Evaluating: Student4
*************************

Playing Matches:
```

```
----------
  Match 1:       Student4  vs     Random         Result: 35 to 5
  Match 2:       Student4  vs     MM_Null        Result: 26 to 14
  Match 3:       Student4  vs     MM_Open        Result: 21 to 19
  Match 4:       Student4  vs  MM_Improved       Result: 19 to 21
  Match 5:       Student4  vs     AB_Null        Result: 26 to 14
  Match 6:       Student4  vs     AB_Open        Result: 24 to 16
  Match 7:       Student4  vs  AB_Improved       Result: 21 to 19


Results:
----------
Student4        61.43%

*************************
  Evaluating: Student5
*************************

Playing Matches:
----------
  Match 1:       Student5  vs     Random         Result: 36 to 4
  Match 2:       Student5  vs     MM_Null        Result: 27 to 13
  Match 3:       Student5  vs     MM_Open        Result: 26 to 14
  Match 4:       Student5  vs  MM_Improved       Result: 19 to 21
  Match 5:       Student5  vs     AB_Null        Result: 27 to 13
  Match 6:       Student5  vs     AB_Open        Result: 21 to 19
  Match 7:       Student5  vs  AB_Improved       Result: 24 to 16


Results:
----------
Student5        64.29%

*************************
  Evaluating: Student6
*************************

Playing Matches:
----------
  Match 1:       Student6  vs     Random         Result: 38 to 2
  Match 2:       Student6  vs     MM_Null        Result: 29 to 11
  Match 3:       Student6  vs     MM_Open        Result: 21 to 19
  Match 4:       Student6  vs  MM_Improved       Result: 21 to 19
  Match 5:       Student6  vs     AB_Null        Result: 28 to 12
  Match 6:       Student6  vs     AB_Open        Result: 29 to 11
```

```
 Match 7:         Student6  vs  AB_Improved           Result: 22 to 18


Results:
----------
Student6        67.14%

*************************
 Evaluating: Student7
*************************

Playing Matches:
----------
 Match 1:         Student7  vs     Random             Result: 36 to 4
 Match 2:         Student7  vs     MM_Null            Result: 32 to 8
 Match 3:         Student7  vs     MM_Open            Result: 27 to 13
 Match 4:         Student7  vs  MM_Improved           Result: 23 to 17
 Match 5:         Student7  vs     AB_Null            Result: 31 to 9
 Match 6:         Student7  vs     AB_Open            Result: 24 to 16
 Match 7:         Student7  vs  AB_Improved           Result: 24 to 16


Results:
----------
Student7        70.36%

*************************
Evaluating: WID170715_Implementation
*************************

Playing Matches:
----------
 Match 1:         WID170715_Implementation  vs     Random           Result: 38 to 2
 Match 2:         WID170715_Implementation  vs     MM_Null          Result: 37 to 3
 Match 3:         WID170715_Implementation  vs     MM_Open          Result: 25 to 15
 Match 4:         WID170715_Implementation  vs  MM_Improved         Result: 26 to 14
 Match 5:         WID170715_Implementation  vs     AB_Null          Result: 32 to 8
 Match 6:         WID170715_Implementation  vs     AB_Open          Result: 28 to 12
 Match 7:         WID170715_Implementation  vs  AB_Improved         Result: 28 to 12


Results:
----------
WID170715_Implementation    76.43%
```

# APPENDIX C
# Evaluation Against Teammates

This script evaluates the performance of the custom heuristic function by comparing the strength of an agent using iterative deepening (ID) search with alpha-beta pruning against the strength rating of agents using other heuristic functions. The `ID_Improved` agent provides a baseline by measuring the performance of a basic agent using Iterative Deepening and the "improved" heuristic (from lecture) on your hardware. The `Student` agent then measures the performance of Iterative Deepening and the custom heuristic against the same opponents.

```
*************************
Evaluating: WID170017_Implementation
*************************

Playing Matches:
----------
  Match 1:       WID170017_Implementation  vs     Random          Result: 65 to 15
  Match 2:       WID170017_Implementation  vs     MM_Null         Result: 59 to 21
  Match 3:       WID170017_Implementation  vs     MM_Open         Result: 42 to 38
  Match 4:       WID170017_Implementation  vs  MM_Improved        Result: 34 to 46
  Match 5:       WID170017_Implementation  vs     AB_Null         Result: 50 to 30
  Match 6:       WID170017_Implementation  vs     AB_Open         Result: 46 to 34
  Match 7:       WID170017_Implementation  vs  AB_Improved        Result: 37 to 43


Results:
----------
WID170017_Implementation     59.46%

*************************
  Evaluating: WID170021_Implementation
*************************

Playing Matches:
----------
  Match 1:       WID170021_Implementation  vs     Random          Result: 68 to 12
  Match 2:       WID170021_Implementation  vs     MM_Null         Result: 63 to 17
  Match 3:       WID170021_Implementation  vs     MM_Open         Result: 49 to 31
  Match 4:       WID170021_Implementation  vs  MM_Improved        Result: 43 to 37
  Match 5:       WID170021_Implementation  vs     AB_Null         Result: 54 to 26
  Match 6:       WID170021_Implementation  vs     AB_Open         Result: 50 to 30
  Match 7:       WID170021_Implementation  vs  AB_Improved        Result: 49 to 31


Results:
----------
```

```
WID170021_Implementation     67.14%

*************************
  Evaluating: WID170711_Implementation
*************************

Playing Matches:
----------
  Match 1:    WID170711_Implementation  vs     Random       Result: 69 to 11
  Match 2:    WID170711_Implementation  vs     MM_Null      Result: 60 to 20
  Match 3:    WID170711_Implementation  vs     MM_Open      Result: 43 to 37
  Match 4:    WID170711_Implementation  vs  MM_Improved     Result: 45 to 35
  Match 5:    WID170711_Implementation  vs     AB_Null      Result: 56 to 24
  Match 6:    WID170711_Implementation  vs     AB_Open      Result: 60 to 20
  Match 7:    WID170711_Implementation  vs  AB_Improved     Result: 54 to 26


Results:
----------
WID170711_Implementation     69.11%

*************************
  Evaluating: WID170715_Implementation
*************************

Playing Matches:
----------
  Match 1:    WID170715_Implementation  vs     Random       Result: 76 to 4
  Match 2:    WID170715_Implementation  vs     MM_Null      Result: 73 to 7
  Match 3:    WID170715_Implementation  vs     MM_Open      Result: 50 to 30
  Match 4:    WID170715_Implementation  vs  MM_Improved     Result: 43 to 37
  Match 5:    WID170715_Implementation  vs     AB_Null      Result: 70 to 10
  Match 6:    WID170715_Implementation  vs     AB_Open      Result: 61 to 19
  Match 7:    WID170715_Implementation  vs  AB_Improved     Result: 56 to 24


Results:
----------
WID170715_Implementation     76.61%
```