

---

# HEURISTIC ANALYSIS

---

For an Adversarial Game Playing Agent for Isolation

Submitted by

Azraf Kabir

(17129042/1 | WID170711)

as part of WID3009: Artificial Intelligence Game Programming

## TABLE OF CONTENTS

---

Synopsis	2
Custom Heuristics	2
Performance	3
Conclusion	5
Appendices	6

# SYNOPSIS

---

Isolation is a deterministic, two-player game of perfect information in which the players alternate turns moving a single piece from one cell to another on a board. Whenever either player occupies a cell, that cell becomes blocked for the remainder of the game. The first player with no remaining legal moves loses, and the opponent is declared the winner.

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.

## CUSTOM HEURISTICS

---

Modifying the code in the 'game\_agent.py' file, specifically the 'custom\_score()' function.

The Code for the heuristics is as follows:

```
if game.is_loser(player):
    return float('-inf')

if game.is_winner(player):
    return float('inf')

my_moves = game.get_legal_moves(player)
opp_moves = game.get_legal_moves(game.get_opponent(player))
my_moves_n = len(my_moves)
opp_moves_n = len(opp_moves)

w, h = game.width / 2., game.height / 2.
y, x = game.get_player_location(player)
# While considering available moves, try to stay towards the middle.
return float(my_moves_n - opp_moves_n) - float(math.sqrt((h - y)**2 + (w - x)*
*2))*0.25
```

This custom heuristic implementation is run under the agent name of "WID170711 Implementation"

# PERFORMANCE

---

The performance 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

## Against Standard Opponents

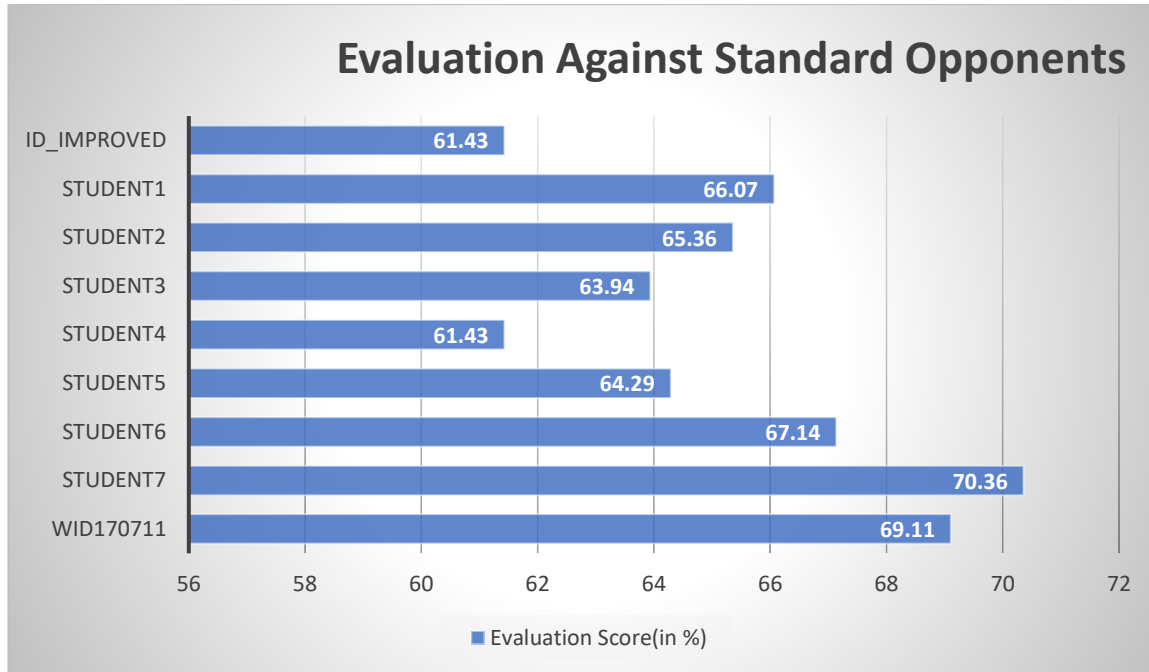
Here, we run the custom heuristics along with other heuristic components already provided in the project.

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
WIDI 70715_Implementation	custom_score()

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



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

## Against Teammates

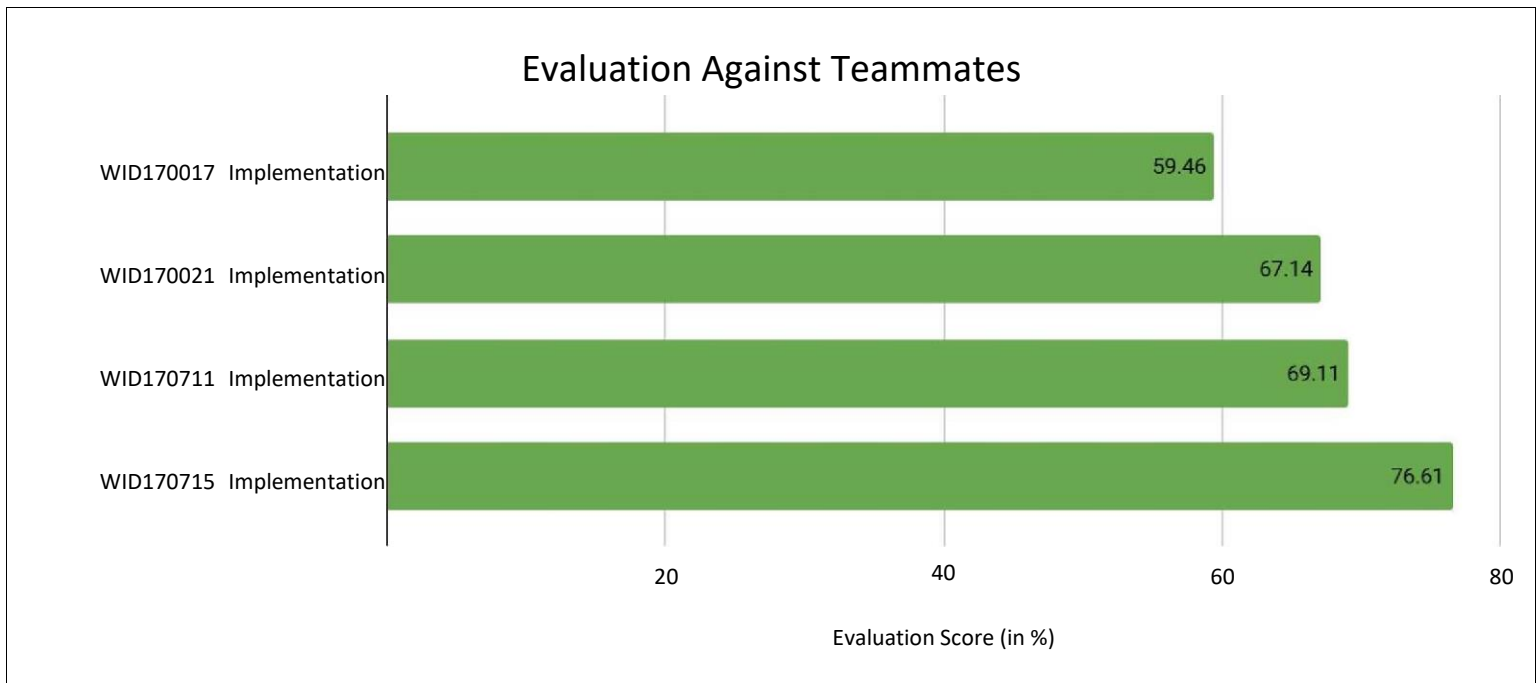
Here, we run the custom heuristic implementation against my teammates' heuristic implementations.

To minimize the number of variables affecting the performance, the heuristics are run on a single machine (WIDI 70715, 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	WIDI 70021 _Implementation
17129042   WID170711	Azraf Kabir	WIDI 70711 _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 B: 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. When run with the included standard opponent agents, at 10 matches, the custom implementation got a final score of 69.11%. When run along with the implementations of teammates, at 20 matches, the custom implementation got a score of 69.11%.

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 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.

-----

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:

-----

WID170711\_Implementation 69.11%

## APPENDIX B 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 v s	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%