

PROJECT

Build a Game-Playing Agent

A part of the Artificial Intelligence Nanodegree and Specializations Program

PROJECT REVIEW

CODE REVIEW

NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT



Shiny Student

This work is an outstanding one. I enjoyed reviewing this work since It is straight, concise, well documented and a well-articulated material. A lot of knowledge in this area has been showcased in this project. Keep it up with this learning spirit and implementations in other projects will always be awesome!

The following resources can be of great interest:

- [Relevant Research, Papers, and Presentations](#)
- [General Game Playing in AI Research and Education](#)
- [Reinforcement Learning: Artificial Intelligence in Game Playing](#)
- [General Game Playing in AI Research and Education](#)
- [Artificial Intelligence for Computer Games](#)
- [Games, computers, and artificial intelligence](#)

Game Playing Agent

The minimax and alphabeta functions pass all test cases.

Correct!

Submission Includes All Files

All required file included.

Correct!

Heuristic Analysis

At least three evaluation functions are implemented and analyzed.

A great job was done in this section of the project by implementing and documenting three evaluation functions. Keep up with this wonderful work!

Pro Tips

The following resources may be of great interest:

- [Adversarial Search and Game-Playing;](#)
- [How to create a good evaluation function for a game;](#)
- [Heuristic Search When Time Matters;](#)

A brief report lists (using a table and any appropriate visualizations) and verbally describes the performance of agents using the implemented evaluation functions. Performance data includes results from tournament.py comparing (at a minimum) the best performing student heuristic against the ID_Improved agent.

A great job was done in verbally describing the performance of the agents and also visualizing the performance on a tables. Keep it up!

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	8	2	10	0	8	2	8	2
2	MM_Open	9	1	7	3	9	1	10	0
3	MM_Center	8	2	8	2	7	3	9	1
4	MM_Improved	6	4	6	4	8	2	7	3
5	AB_Open	5	5	5	5	6	4	6	4
6	AB_Center	7	3	5	5	5	5	4	6
7	AB_Improved	3	7	6	4	5	5	3	7
Win Rate:		65.7%		67.1%		68.6%		67.1%	

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	10	0	10	0	8	2
2	MM_Open	7	3	6	4	6	4	7	3
3	MM_Center	8	2	8	2	8	2	9	1
4	MM_Improved	8	2	9	1	7	3	6	4
5	AB_Open	4	6	5	5	6	4	5	5
6	AB_Center	6	4	5	5	5	5	3	7
7	AB_Improved	3	7	4	6	4	6	3	7
Win Rate:		64.3%		67.1%		65.7%		58.6%	

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	9	1	10	0	9	1	8	2
2	MM_Open	8	2	8	2	7	3	9	1
3	MM_Center	7	3	9	1	10	0	9	1
4	MM_Improved	7	3	8	2	5	5	4	6
5	AB_Open	6	4	6	4	5	5	3	7
6	AB_Center	5	5	6	4	6	4	4	6
7	AB_Improved	6	4	6	4	5	5	4	6

The report makes a recommendation about which evaluation function should be used and justifies the recommendation with at least three reasons supported by the data.

Nice job providing recommendations on which evaluation function should be used 🍌 The solution has also provided reasons to support the recommendation which is really cool. Keep it up!

Paper Summary

The write up is approximately 1 page (500 words) and includes a summary of the paper (including new techniques introduced), and the key results (if any) that were achieved.

A great job was done in providing a one page summary of the Mastering the game of Go with deep neural networks and tree search and the techniques used were also discussed. It was also a nice thing discussing the results achieved in the paper summary 🍌

[📄 DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)