



[< Back to Artificial Intelligence Nanodegree](#)

Build a Game Playing Agent


REVIEW

CODE REVIEW 7

HISTORY

Meets Specifications

Hi Udacity Learner,

Brilliant work, the project has all the requirements. Congratulations, you have built an adversarial game playing agent! I also want to commend the heuristic analysis for the reason that it was well-written and shows that you have a great understanding of the project. Keep it up dear student, stay awesome! 

Game Agent Implementation

(AUTOGRADED) Game playing agent can return an action.

- `.get_action()` method calls `self.queue.put()` at least once before the time limit expires

Correct! (Note: this rubric item was graded automatically.)

(AUTOGRADED) Game playing agent can play a full game.

- `CustomPlayer` successfully plays as both player 1 and player 2 in a full game to a terminal state (i.e., the agent does not deadlock during search, return an invalid action, or raise an exception during a game)

Correct! (Note: this rubric item was graded automatically.)

Experimental Results & Report

`CustomAgent` class implements at least one of the following:

- Custom heuristic (must not be one of the heuristics from lectures, and cannot *only* be a combination of the number of liberties available to each agent)
- Opening book (must be at least 4 plies deep)
- Implements an advanced technique not covered in lecture (e.g., killer heuristic, principle variation search, Monte Carlo tree search, etc.)

Great! The heuristic was implemented correctly. There is not a guaranteed way to find a good heuristic; it usually involves humans developing insight into the characteristics that matter in the game through experience so good job on this one!

For further learning, I encourage you to take a look at heuristics for other games. 😊

- [Playing Checkers with Minimax](#)
- [Alpha-Beta Pruning and Checkers](#)
- [The Heuristic Search and the Game of Chess](#)
- [Improve Minimax Heuristic Function for Connect4](#)

Submission includes a table or chart with data from an experiment to evaluate the performance of their agent. The experiment should include an appropriate performance baseline. (Suggested baselines shown below.)

Advanced Heuristic

- Baseline: `#my_moves - #opponent_moves` heuristic from lecture (should use `fair_matches` flag in `run_match.py`)
Opening book
- Baseline: randomly choosing an opening move (should *not* use `fair_matches` flag in `run_match.py`)
Advanced Search Techniques
- Baseline: student must specify an appropriate baseline for comparison (student must decide whether or not `fair_matches` flag should be used)

Nice work on including tables that evaluate the performance of the algorithm, i.e.:

- `Minimax`
- `Random`
- `Greedy`

The report also includes appropriate discussion regarding the results and the heuristics used; it was very clear and comprehensive.

Notes

- Comparing performance against multiple opponents produces better analysis. It's possible for increasing performance against one opponent to decrease performance against other agents.
- More advanced analysis would include analyzing the search depth of their heuristic. The game grows exponentially, so slower heuristics can be more effective if they are significantly more accurate.

Submission includes a short answer to the applicable questions below. (A short answer should be at least 1-2 sentences at most a small paragraph.)

NOTE: students only need to answer the questions relevant to the techniques they implemented. They may choose *one* set of questions if their agent incorporates multiple techniques.

Advanced Heuristic

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

Opening book

- 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?
- 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?

Advanced Search Techniques

- Choose a baseline search algorithm for comparison (for example, alpha-beta search with iterative deepening, etc.). How much performance difference does your agent show compared to the baseline?
- Why do you think the technique you chose was more (or less) effective than the baseline?

The analysis provided was very thorough and the answers perfectly address the rubric points. Excellent work! 🙌 Good job on sharing the results and your observation on accuracy being more important, it is correct that increasing the depth (until we reach a certain limit) will further increase the win rate.

In some heuristics and results, speed matters more than accuracy for the reason that if there are no moves returned within the time limit, your turn might be forfeited that could result in losing the game.

 [DOWNLOAD PROJECT](#)

7

[CODE REVIEW COMMENTS](#)



RETURN TO PATH

Rate this review
