



[< Return to Classroom](#)

Build a Game Playing Agent

REVIEW

CODE REVIEW 3

HISTORY

Meets Specifications

Congratulations

Great submission!

You have implemented Option 1 "Advanced Heuristic".

All functions were implemented correctly, and the final algorithm seems to work quite well.

You presented your experimental results and analysis of performance in a brief, but clearly written report.

By the way, I suggest you take a look at these papers:

- [Search Heuristics for Isolation](#) in which the results of the application of 7 different heuristics were presented using alpha-beta pruning with iterative depth against different opponents who use different algorithms in the Isolation game.
- [A Survey of Monte Carlo Tree Search Methods](#). Monte Carlo Tree Search (MCTS) is a search method that combines the precision of tree search with the generality of random sampling.

Take also a look at this paper:

- [Improving Opening Book Performance Through Modeling of Chess Opponents.](#)

Good luck, and enjoy the rest of the course!

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.)

Good work

You have implemented the Advanced Heuristic choice, you have passed all unit tests and Project Assistant tests.

For more information on different heuristics that could be used, follow the links provided below;

- [This document](#) for example talks on strategy game programming. In it we see principal variation search and killer moves.
- More on killer heuristic and other heuristics can be found [here](#)
- For the algorithm of principal variation search, this [link to wikipedia](#) provides the pseudocode.
- In this [document](#) we see some details on Monte Carlo Tree Search.

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)

Your report gives a helpful introduction and identifies the values used for key parameters.

Your table of experimental results is simple but effective.

OPPONENT	WINNIG MATCHES (%)
MINIMAX	20
SELF	60
GREEDY	80
RANDOM	95

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?

You've given brief but effective answers to both questions.

Advanced Heuristic

- What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during the search?
→ Depth 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?
→ If there is a time limit, time becomes important in order to select the best move.

More information is given in the links below on the relationship between speed and accuracy.

- [In this article](#), you will see whether you should go for speed or accuracy.
- Finally, in this [document](#) you will see how and when to prioritize between speed and accuracy.
- [This document](#) provide more info on the relationship between the time and depth and how they influence the result.

 [DOWNLOAD PROJECT](#)

RETURN TO PATH

Rate this review

START