‹ Return to "Artificial Intelligence Nanodegree" in the classroom

DISCUSS ON STUDENT HUB ›

Build a Game Playing Agent

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Beautiful project. Delightful to read and review, with a very to-the-point writing.

I hope you have learned a lot during this project! And hope that you can learn some small thing with my review. Small improvements are the way to go into master skills.

Have a great learning journey onwards!

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.)**

Awesome. You've chosen the custom heuristic, and you overperformed the baseline, which is very hard!

The code inside the report is great. You organized very clearly and your report is very stylised.

✓ **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)**

You added a table and a chart, which convey the necessary information in a very concise fashion.

You asked about the fair matches in the search techniques:

> If fair_matches is true, then the agents repeat every game they played, but the agents switch initiative and use their opponent's opening move. In some games, picking a winning move for the opening guarantees the player a victory. Playing "fair" matches this way will balance out the advantage of picking perfect openings (the player would win the first time, and then lose when their opponent uses that move against them).

You in the advanced search, one could either use fair matches or not, but the results will be somewhat different in each case. For example, if the search technique is very deep and is able to select a "perfect opening", but not so powerful in the middle of the game, then choosing "fair matches" will make the technique underperform. To choose whether to use "fair matches" is dependent on how much the opening matters for your algorithm.

It is essentially a choice about what you want to see and how would you discuss it. You could even run both fair matches and unfair matches, and compare the performance of your algorithm to understand better how much it is dependant on the opening moves!

✓ **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?**

Great! You provided good answers to each of the questions. You did a somewhat deep dive on the comparison of different time limits.

It would be **very nice** to see a chart comparing the performance with respect to different time limits, don't you think?!

DOWNLOAD PROJECT

RETURN TO PATH