

## REPORT

Agent	Adversary	fair_matches	Num. Rounds	Win Rate
MINIMAX ( $\alpha$ , $\beta$ deepening.)	MINIMAX	used	100	77.6% (BASELINE)
MCTS	MINIMAX	used	100	80.0%
MCTS	GREEDY	used	100	93.8%
MCTS	RANDOM	used	100	98.8%

Table for the following CRITERIA 1.

### 1. CRITERIA

Report includes a table or chart documenting an experiment to evaluate the performance of their agent

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)

### 2. CRITERIA

Report answers all required questions

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 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?
  - ➔ The baseline search algorithm for comparison in this report is the minimax search algorithm with alpha-beta iterative deepening. Even if the algorithm itself is very slightly different from the original minimax algorithm, the performance of the alpha-beta search is way better than the minimax algorithm. With a simple heuristics, the performance can greatly be improved.
- Why do you think the technique you chose was more (or less) effective than the baseline?
  - ➔ As MCTS concentrates on the more promising subtrees, the game tree in MCTS asymmetrically grows, so that it can achieve better results in games than classical algorithms with a high branching factor. However, this does not imply that MCTS is always better than the minimax search (generally better with using the same computational resource). While running MCTS, some branches leading to a loss can be chosen (some difficult solutions not easily searched with random approaches). The AlphaGo's loss in the fourth game against Lee Sedol may be related to this issue. The performance of MCTS can greatly improve, for example, with an opening book.