

CMPM 146 Spring 2025

## Assignment 6: Monte Carlo Tree Search

Calvin Li & Luis Rocha

May 22, 2025

### Assignment Questions:

#### 1. Briefly describe how you implemented MCTS. Which selection strategy do you use?

We implemented the MCTS by utilizing the UCB-1 approach. We did this by having the `TreeNode` store a list of (`GameAction`, `TreeNode`) children, making sure that it also tracked visit counts and cumulative rewards. On each iteration, selection is performed by descending with the UCB-1 Formula. In order to fully expand each node we add one randomly unexplored action then roll out a random playout, then back propagate the results back up the tree.

#### 2. How does your agent perform in the various scenarios? How does it compare to the Sampling Bot?

Scenario	MCTS Win %	Sampling Win %
intro	98	99
giant	98	92
offerings	87	90
lowhp	74	56

#### 3. Which parameter values (number of iterations, value for the constant “c” if you are using UCB-1, $\epsilon$ if you use the $\epsilon$ -greedy strategy)

##### parameter values for intro, giant, and offerings:

Iterations: 20, UCB-1 constant (c): 0.5, Games: 100

##### parameter values for lowhp:

Iterations: 75, UCB-1 constant (c): 0.3, Games: 100

#### **4. Did you encounter any particular challenges during the assignment?**

We did not encounter any particular challenges in this assignment, the instructions were very thorough and since we only needed to edit the `TreeNode` it wasn't too complicated.

#### **Individual Retrospectives:**

**Both partners contributed to this document evenly.**

##### **Calvin Li:**

I helped with the rollout and backpropagate methods in the `TreeNode` class. I was in charge of testing various parameter changes in order to best reach the target win rates per scenario. I had to tune our parameters and I chose at least 100 games per test in order to get a good sense of an overall average. I wrote the logic that tracked node statistics and made sure the game state was cloning properly during rollouts and backpropagation. Through this assignment I learned how testing and changing parameters can affect the impact in simulation based decision making like this gameplay loop. I also learned about MCTS `TreeNodes` which is something I had never learned before.

##### **Luis Rocha:**

I focused on the implementation of the MCTS logic in the given `TreeNode` class. I worked on the `Select` and `expand` methods. We decided to go with the UCB-1 strategy for node selection. In the `select` method I made sure that the player favored nodes that had high average performance and also low visit counts. I wrote the `_apply` method in order to convert game actions into either `PlayCard` or `EndAgentTurn` actions based on the current hand we were in. I verified that each node correctly tracked visits and average scores, and that our tree structure grew as expected with each iteration. From this assignment I learned how to use a Monte Carlo Tree Search and how it can be used in video games.

#### **AI acknowledgement:**

We did not use any AI in the creation of this Assignment code or Report