Audrey Fuller, Thalia Kennedy

Professor Homan

Introduction to Artificial Intelligence

11 February 2024

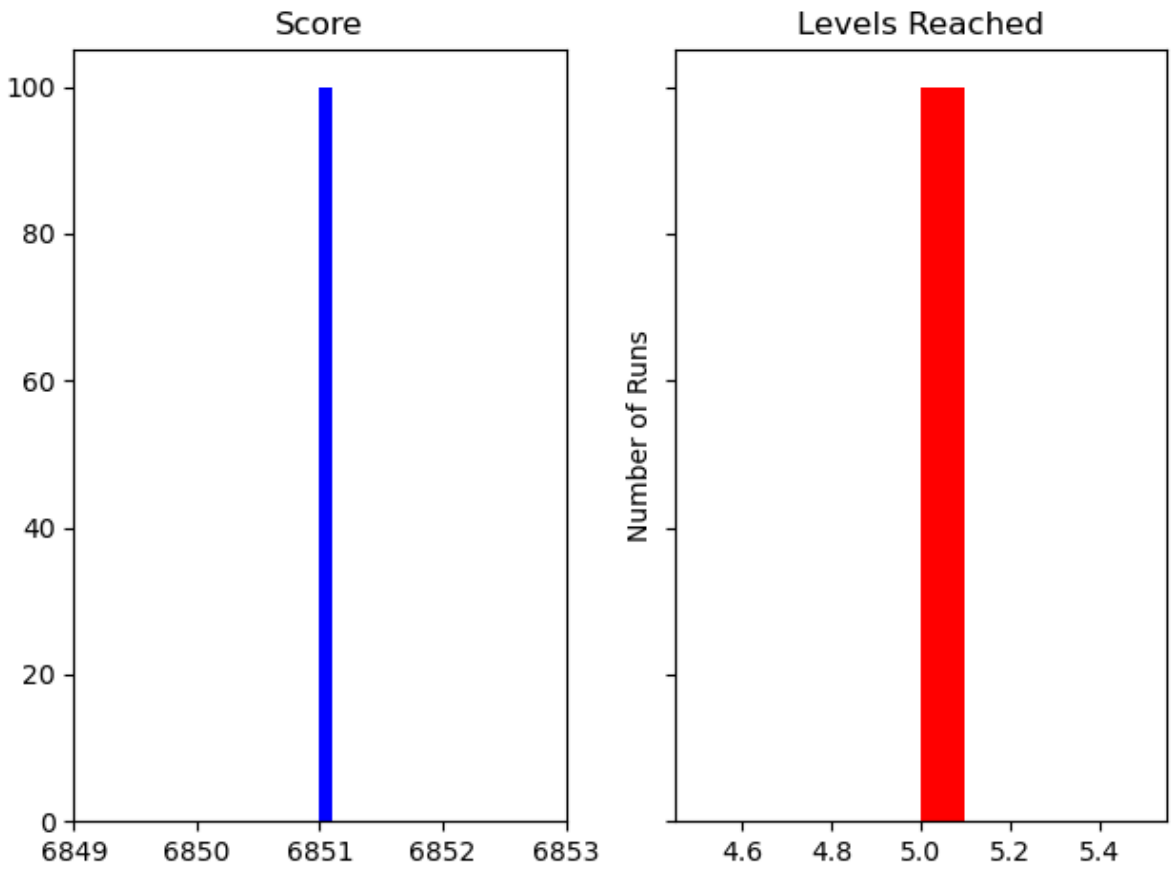
CSCI 331 Project 1 Report

The centipede environment was modeled using the observation, specifically by looking for the colors of the mushrooms/player, centipede and spider for each level. While the level colors seemed random at initial observation, we eventually discovered that there are only 8 unique color combinations. These colors then repeat in the same order every time, meaning they could be stored in an array and accessed with the index of the level number (it could be % 8 once the first 8 levels are cleared to support more than that number). In order to determine the current level, we use the bar color at the bottom, as the location is static. The current bar color is compared with an array of possible bar colors, and the index of the current color is used to determine the level. When debugging, we found that the observation was duplicating the color x and y positions, so we remove the duplicates. In order to find the locations of the important information of the environment, we search the locations where the observation is the color of the current mushroom, centipede or spider color. To differentiate the mushrooms from the player, we look at the region around every mushroom colored point, to see if the number of mushroom pixels is large enough to be the player sprite. In terms of ignored information, we did not search for any of the enemy types such as the scorpion or flea due to their rarity and lack of spawning until later levels. We also ignored all of the observation that fell below the bar, as that just

recorded score and lives. In addition, while the agent searches for the centipede so it can fire at it if it is not busy avoiding enemies, it does not seek out the spider.

There was a bug that we ran into frequently that hindered the environment being properly modeled in later boards. Throughout programming, we would use MatPlotLib to plot the game environment we had modeled to make sure it was being processed properly. In later boards, these graphs would show that some elements of the environment were being processed incorrectly (e.g. the bottom bar being perceived as a spider). While we were able to find work-arounds for the first few iterations of this bug, once we reached board 5, it was much harder to avoid, as the mushrooms were being perceived as spiders in the environment model. While we were not able to solve this bug, we still successfully created an agent that could pass multiple boards of the game. We also only included the first five levels in the list of colors, as the agent is not currently able to clear any past that.

Ideally, we would have liked to have the agent be able to anticipate the centipede's movements, knowing which direction the centipede was headed, to more accurately fire at the enemy. This, however, was a bit out of the scope of this project, and the set of rules that we did give the agent to fire at the centipede worked well enough to pass multiple boards. In addition, we would have liked to add rules for the agent to detect when it was at the edge of the screen so that, instead of moving left or right to avoid an enemy, which would have been ineffective, it would move up or down to avoid the enemy. Again, this implementation was a bit out of scope, and the instructions that the agent did have were successful enough without this addition.



The histograms above showcase the agent's score for 100 runs, as well as the board/level reached for 100 runs. The agent had no randomized movements and the centipede environment is completely deterministic, so the results for both histograms are unanimous. The mean and median for the score are both 6851 as the agent's score was the same every time and the standard error is 0. The same is true for the levels reached by the agent. For every run, the agent would reach level 5 before dying , making the mean and median both 5 and the standard error 0.