Ryan Filgas

Program 3

AI

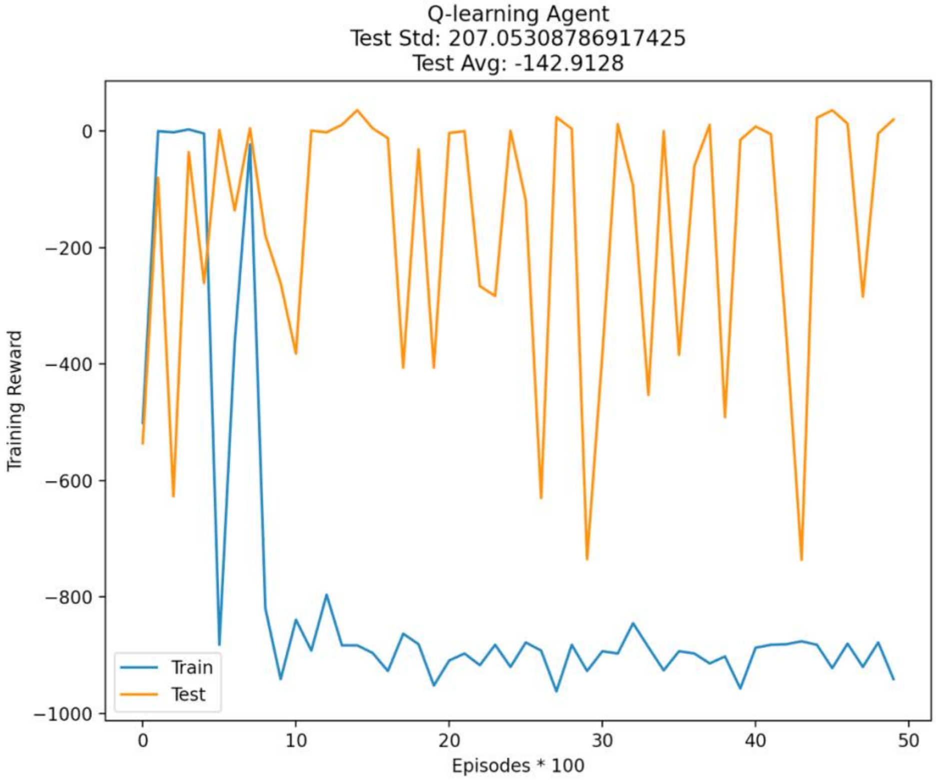
To create Robby the robot I made two separate classes, one for robot and another for the environment. As detailed in the assignment parameters it starts with a 10x10 grid of spaces where each space has a 50% likelihood of containing a can. If Robby picks up a can it gets 10 points, if it tries and the square is empty it gets -1 and if it attempts to go off the grid it gets -5. This is all taken care of by the environment class.

In Robby's inner mechanisms it stores its own 10x10 grid where each square is a node with a representation of its perceived reward for north, east, south, west. Every turn Robby takes an epsilon greedy action from its current square, it receives a reward and updates that action in the previous node according to the formula:

Update = epsilon_greedy_max(previous node) + learning_rate * (reward_from_action + (discount * epsilon_greedy_max(current_node)) – epsilon_greedy_max(previous_node)

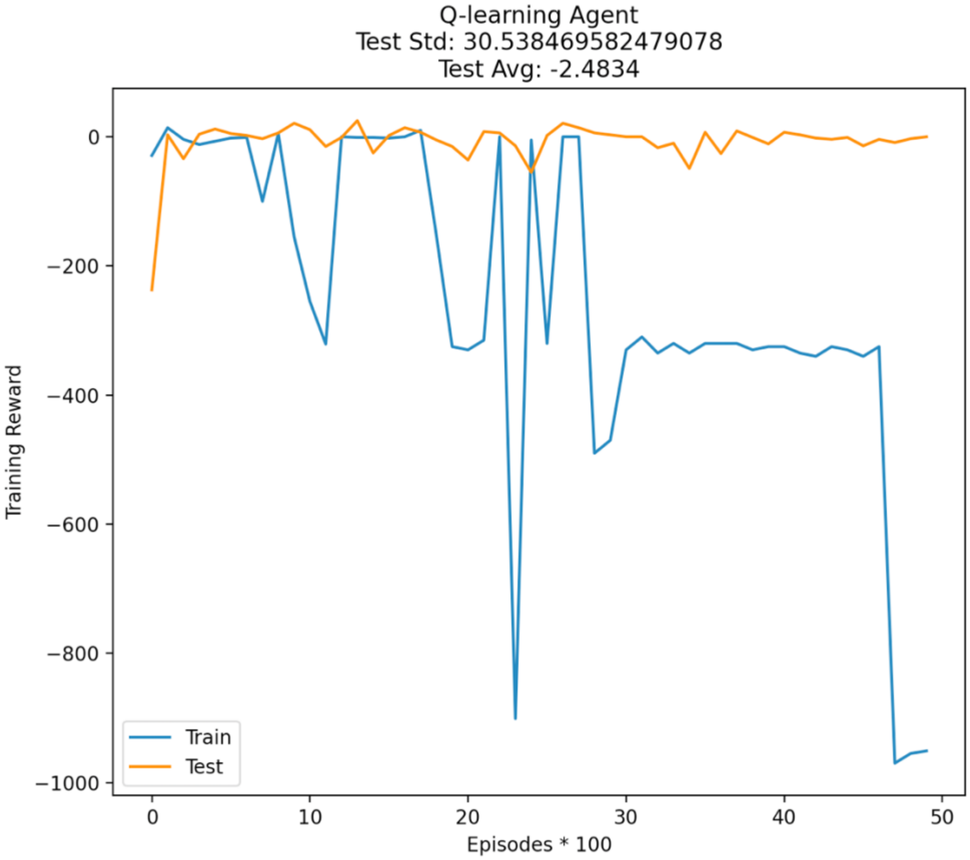
Previous_node_action_taken += Update.

After 100 episodes the environment resets itself to a new random state and Robby is placed in a random square to repeat the process. As a result of the reward mechanisms Robby is incentivized to pick up cans wherever it previously received a reward. Here are initial results:

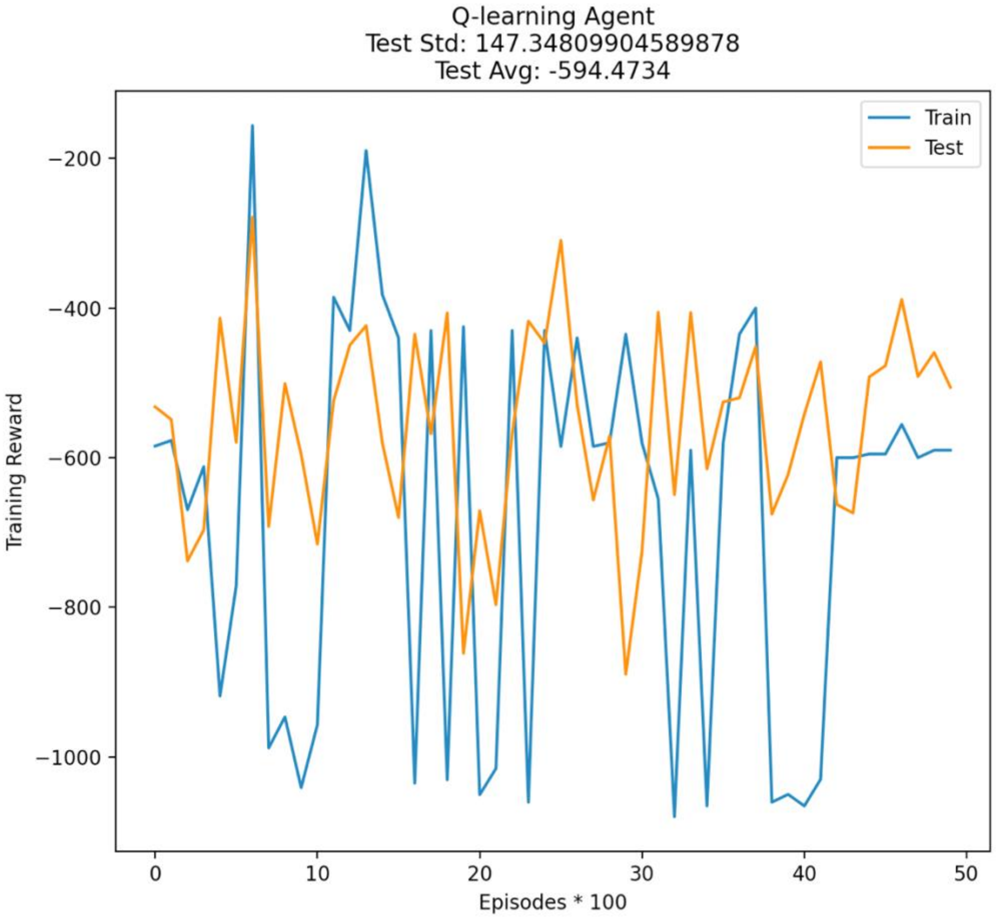


Initially Robby was incentivized to repetitively pick up cans after they were gone since picking up a can left robby in the same state and it was more likely to repeat that action. In order to combat this I added a rule that robby can’t choose the same action from the same square twice. Given a random environment with scattered rewards, exploring slightly more results in better performance as seen in the training and testing phases because Robby won’t try to pick up a can immediately after it picked one up before as seen in the next example below.

Given this, wherever there wasn't a can before robby still normally tried to pick it up once resulting in negative scores for a randomized environment. This is a logical result when the operation of moving is separated from the action of picking up a can. It makes rewards in the state space much more sparse as 4/5 possible actions are at best neutral for rewards and at worst -5. Here is a chart where picking up a can is not repeated for the same square twice. As epsilon approaches 0 for the training agent the rewards gathered progressively gets worse.



Adding a negative .5 reward on top of this for exploration of course results in less reward overall as Robby receives -.5 for every move, and for each position Robby is in 4 to 5 actions result in a negative reward depending on whether there's a can in the square.


Q-learning Agent
Test Std: 147.34809904589878
Test Avg: -594.4734
Train
Test
Training Reward
Episodes * 100