

**School of Computer Science and Engineering**

Artificial Intelligence Assignment 2 Report

|  |  |
| --- | --- |
| **Name:** | Munirah Binte Mohamad (U1822418F) |
| **Module Code:** | CZ3005 |
| **Tut/Lab Group:** | TSP2 |

## Description of algorithm

The method of reinforcement learning that I have chosen is the single-step Q-learning algorithm. This algorithm learns from the actions that are outside the current policy, such as taking random actions. It uses Q-values (action value) to iteratively improve the behaviour of the learning agent.

The Q-learning algorithm that I have coded finds the optimal greedy policy while improving itself using an epsilon greedy policy.

Software used: IDLE (Python 3.7 64-bit)

Outside Module: Plotting.py in ‘lib’ folder (Inside Zip file)

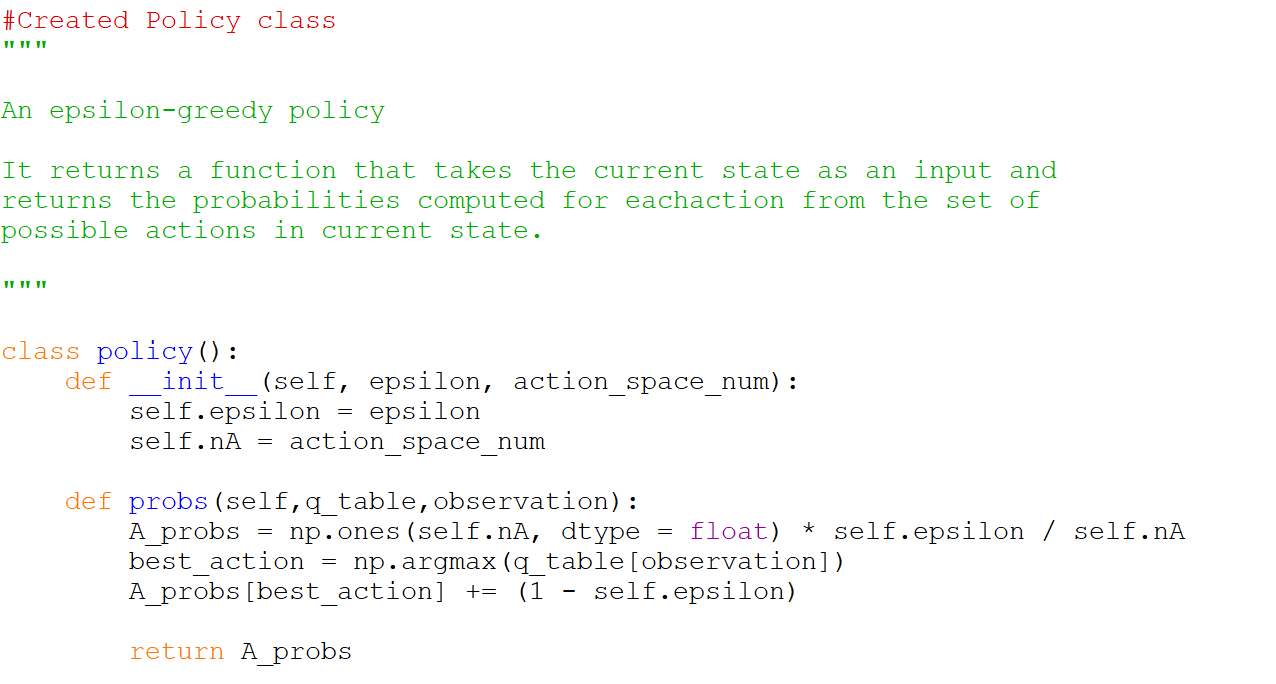


Figure 1. Policy Code

In figure 1, the greedy policy helps to choose actions using the current Q-value estimations. By computing the probability using ( 1 - epsilon), it chooses the action which has the highest Q-value.

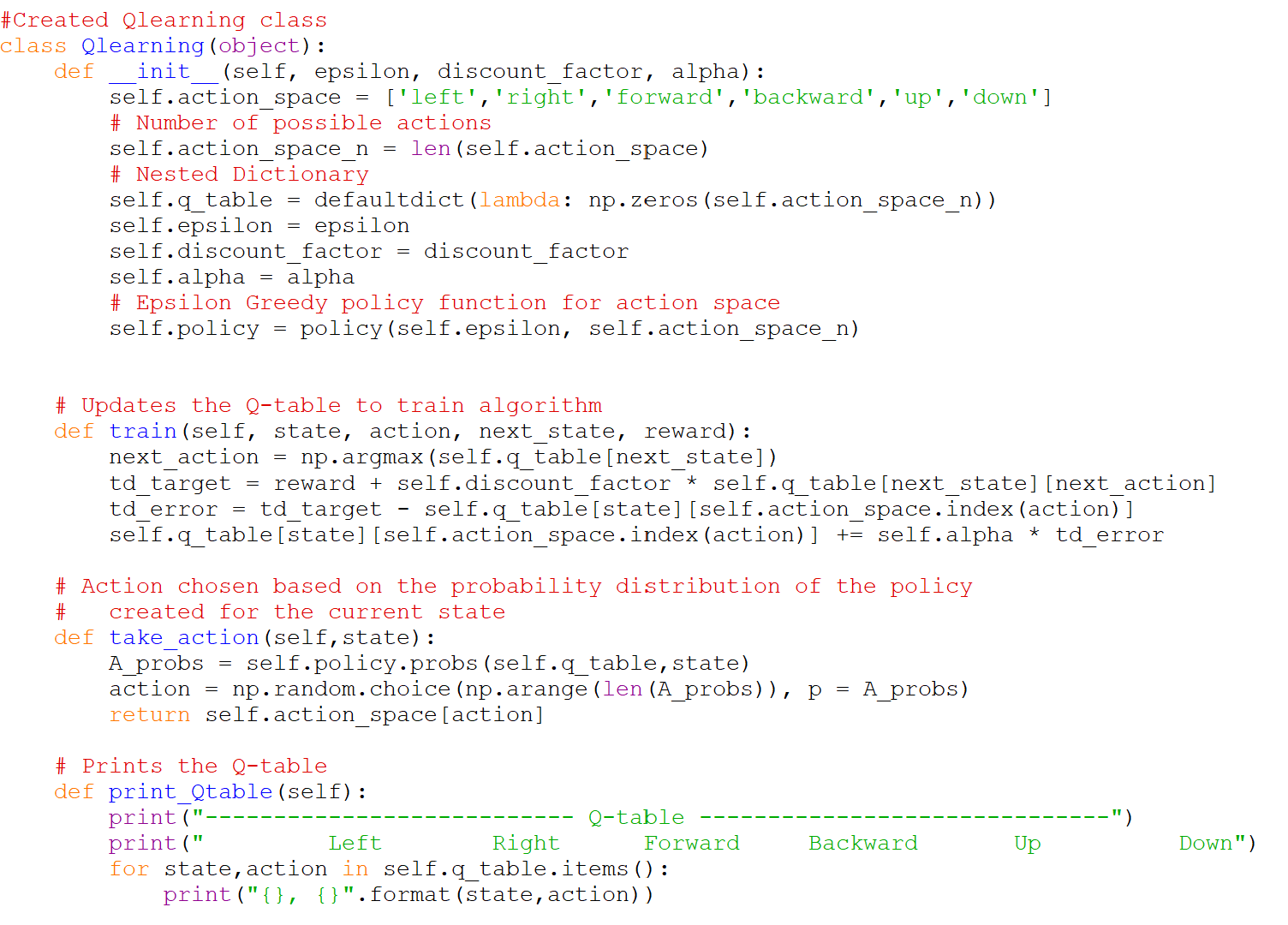
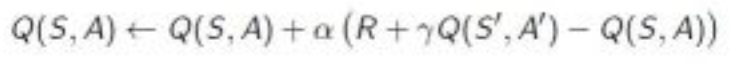


Figure 2. Q-Learning class

In figure 2, I used the following equation to perform a Temporal Difference update to train the algorithm:



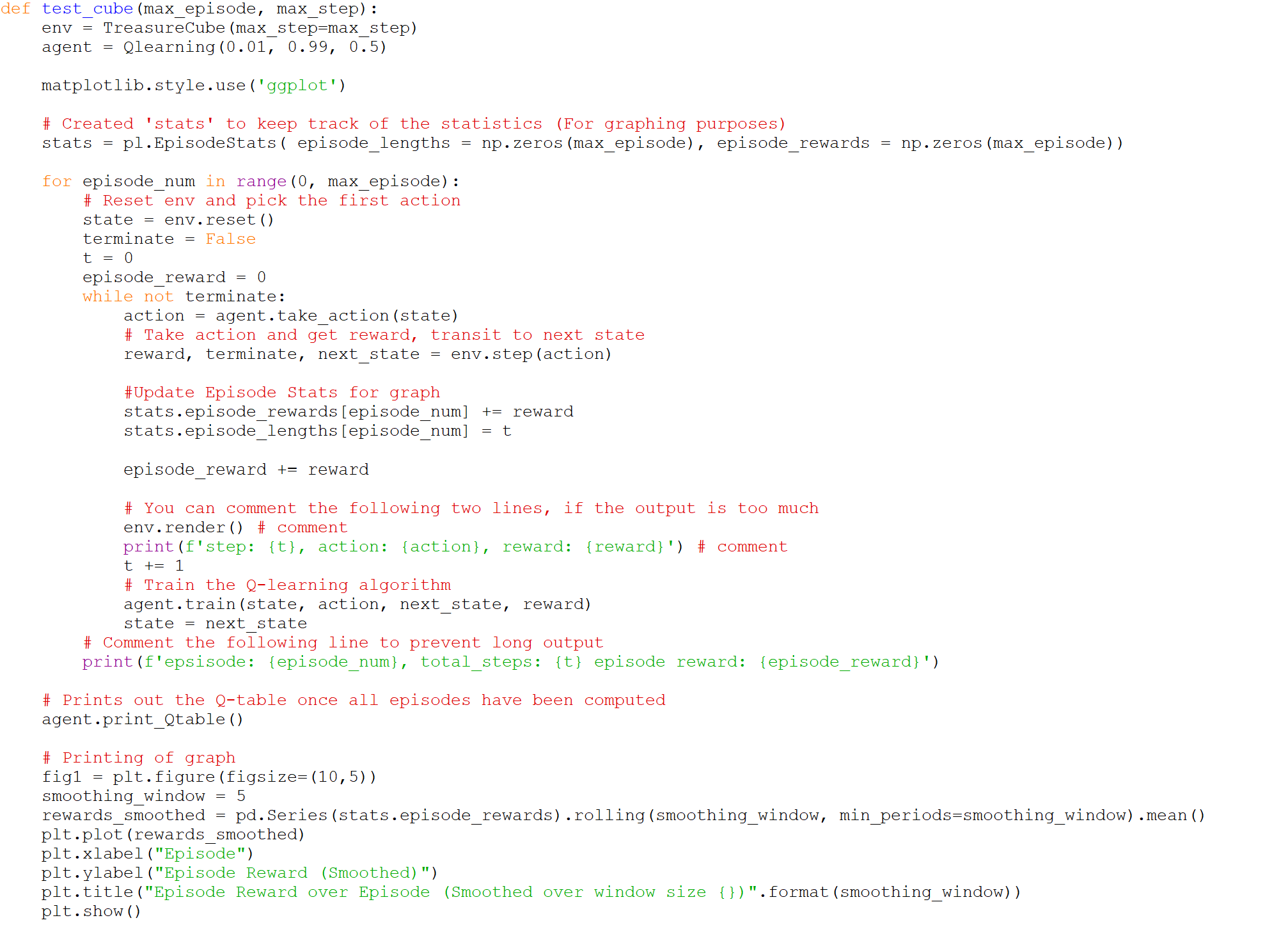


Figure 3. Test Cube Method

## The Learning Progress (Rewards in each episode)

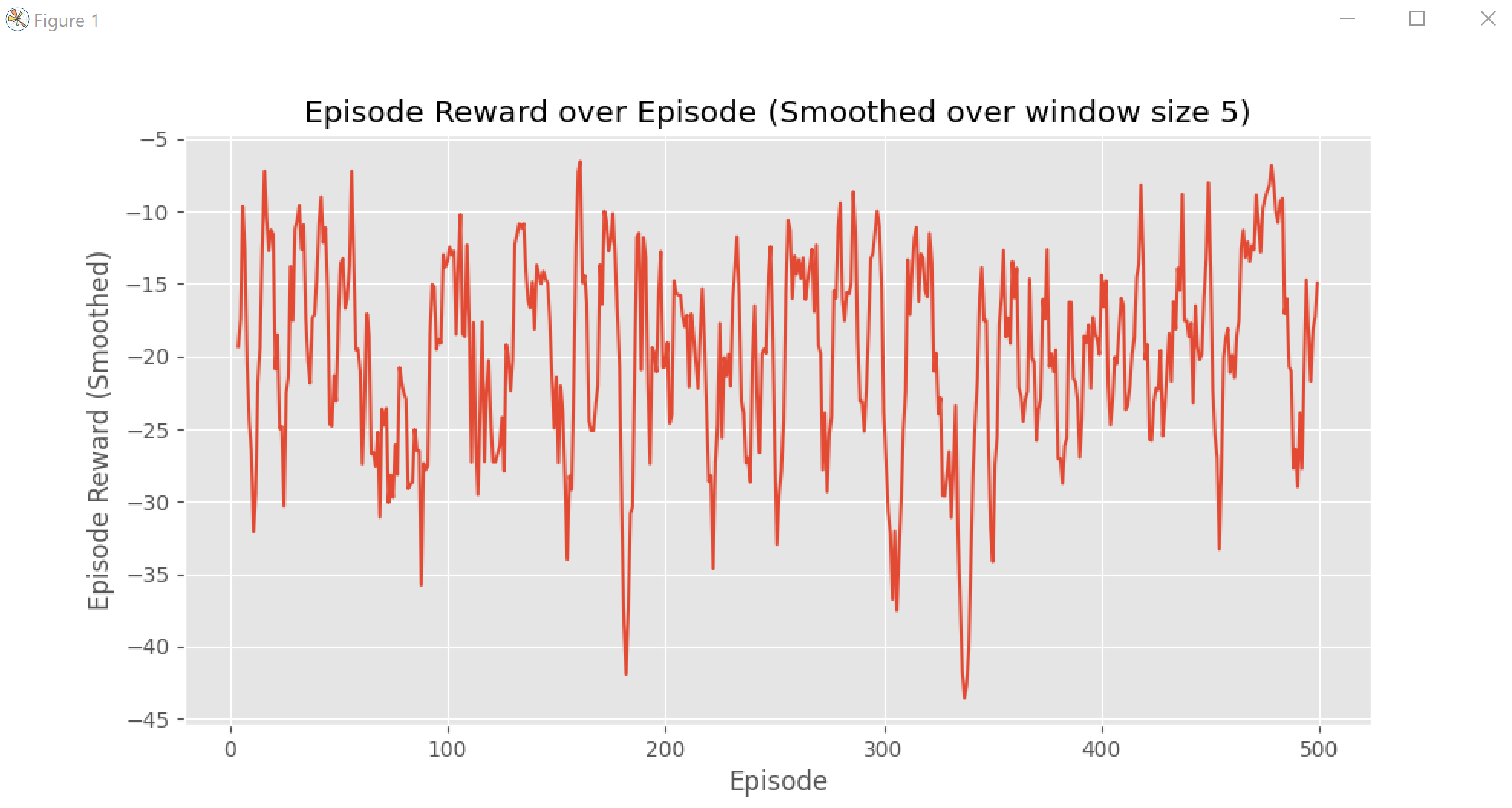


Figure 4. Graph for RandomAgent()

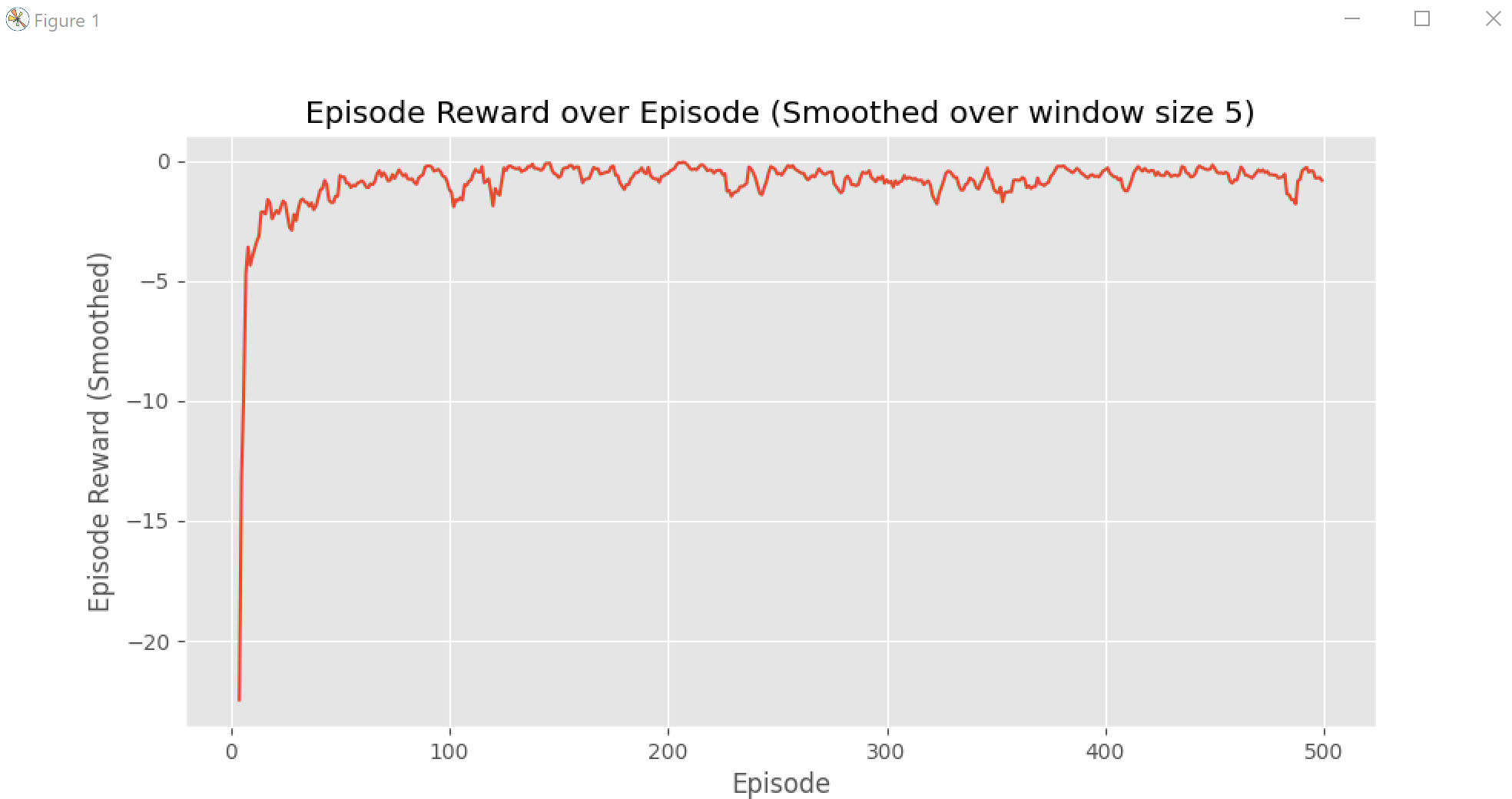


Figure 5. Graph for Qlearning()

Figure 4 and 5 shows the Episode Reward over Episode for RandomAgent and Qlearning classes respectively.

In Figure 4, the graph for RandomAgent shows a random number of highs and lows. This indicates that the episode reward using RandomAgent is not consistent and does not always provide high rewards throughout the episodes.

On the other hand, in Figure 5, the graph for Qlearning shows a significant increase during the fiesr few episodes and maintained at the top level throughout the remaining episodes. This indicates that the episode rewards using Qlearning class progressively increases over time and it maintain at a high reward level per episode. This also shows that the Qlearning agent has learnt to maximise its total reward earned in an episode by behaving optimally at every state.

## Q-table

