

**Title:** Report for achieving optimal walking path for Crawler Bot

**Professor:** Ju Shen

**Course:** CPS 580\_Z1

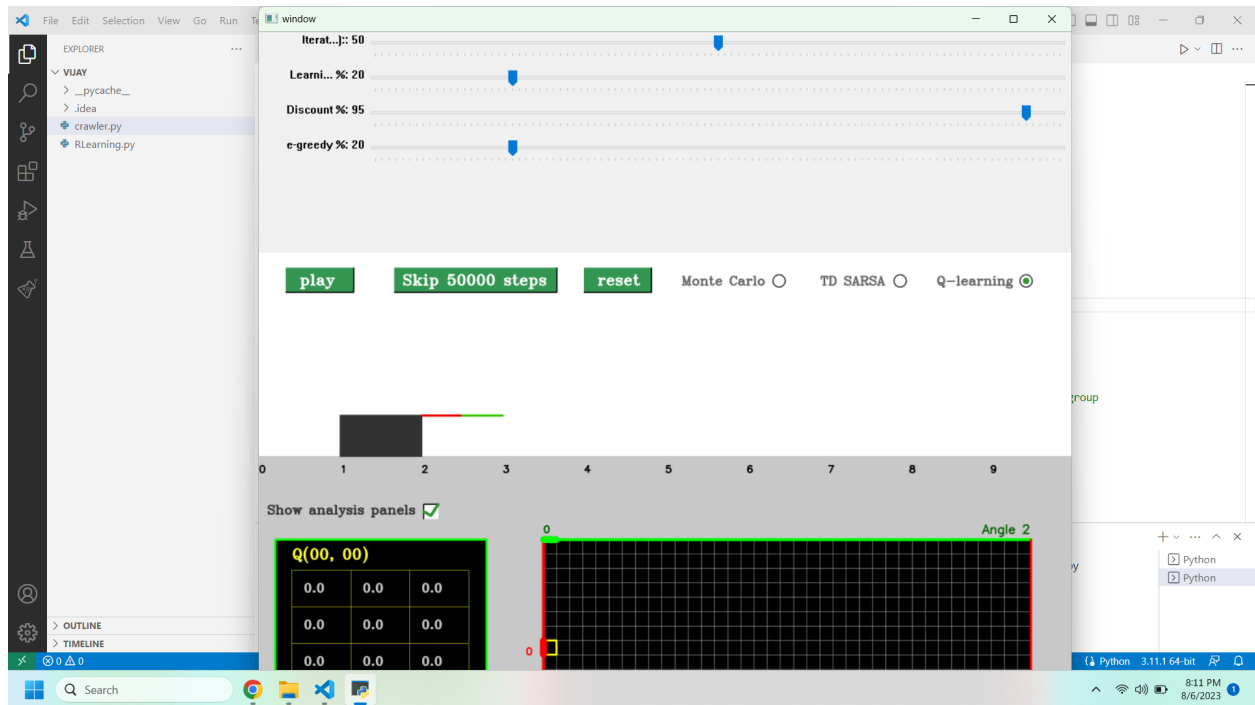
**Subject:** Artificial Intelligence

**Name:** Vijayendra Kosigi

**ID:** 101746057

\*\*\*\*\*

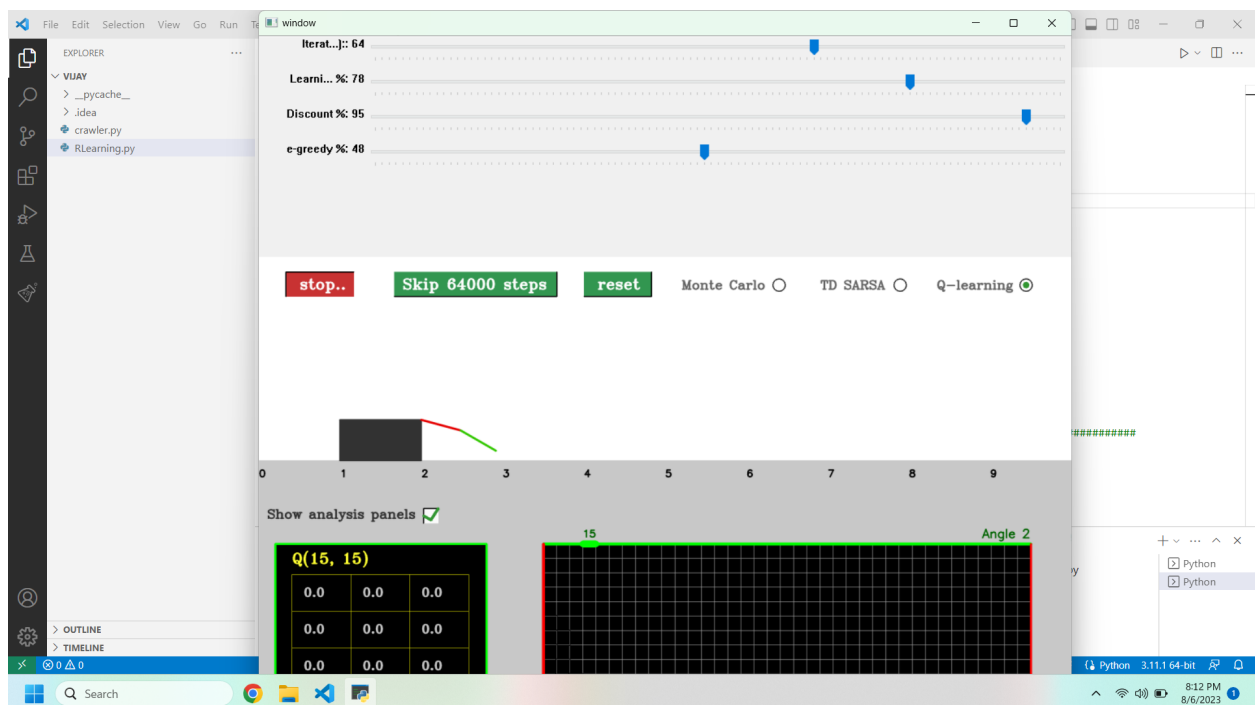
## Report:



The above figure is what we get upon executing the crawler.py file.



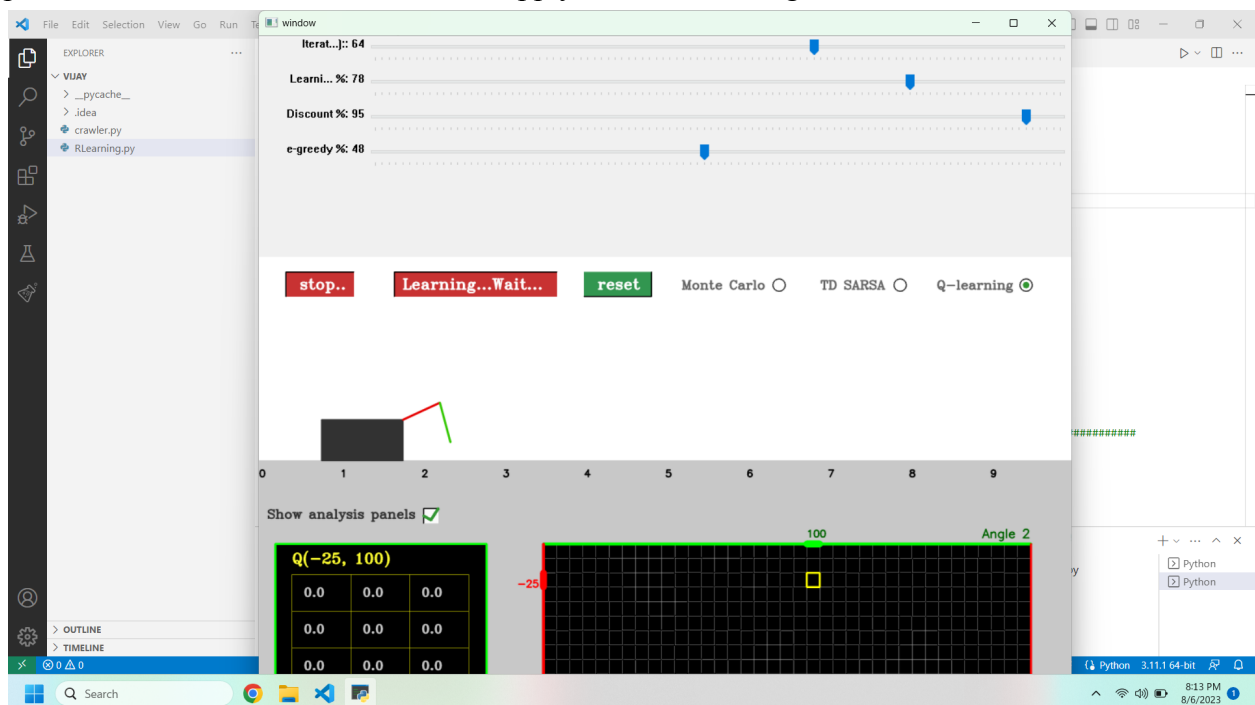
I've chosen the Q-Learning radio button and adjusted the values of Iterations, Learnings, Discount and e-greedy respectively. I've maintained a moderate value for e-greedy in the trial to observe the changes in the crawler bot when the values are too high or low. I've also maintained a high learning rate and have increased the number of iterations in the project to 64,000 steps. The same has been illustrated in the above screenshot.



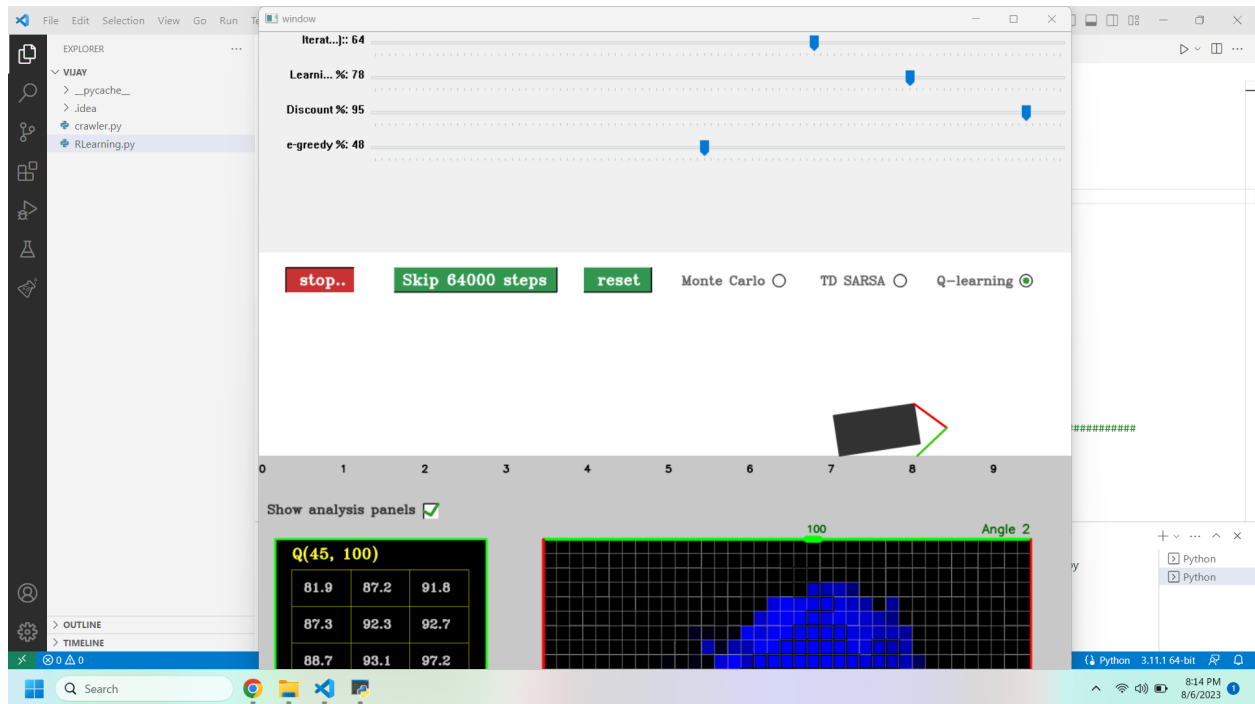
In the above screenshot, we can see that I have selected the play button [at 8:12 PM EDT] and this is the first look of the crawler bot before Artificial Intelligence is applied.



Now, a minute later, we can observe in the above screenshot that there is minimal change in the position of the crawler bot. It's time to apply Artificial Intelligence on the crawler bot.



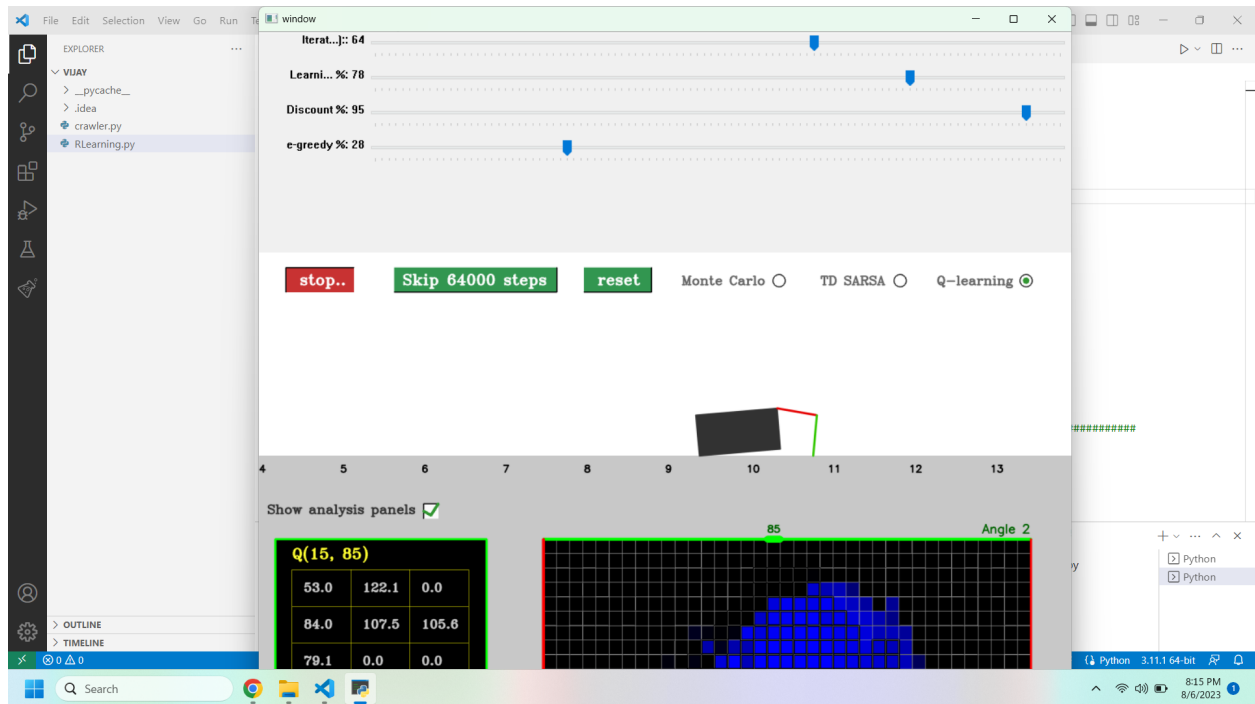
Upon selecting skip 64,000 steps, the crawler bot undergoes learning and it has been illustrated in the above screenshot. This is right at the start of the applied learning process and we will observe its position a minute later.



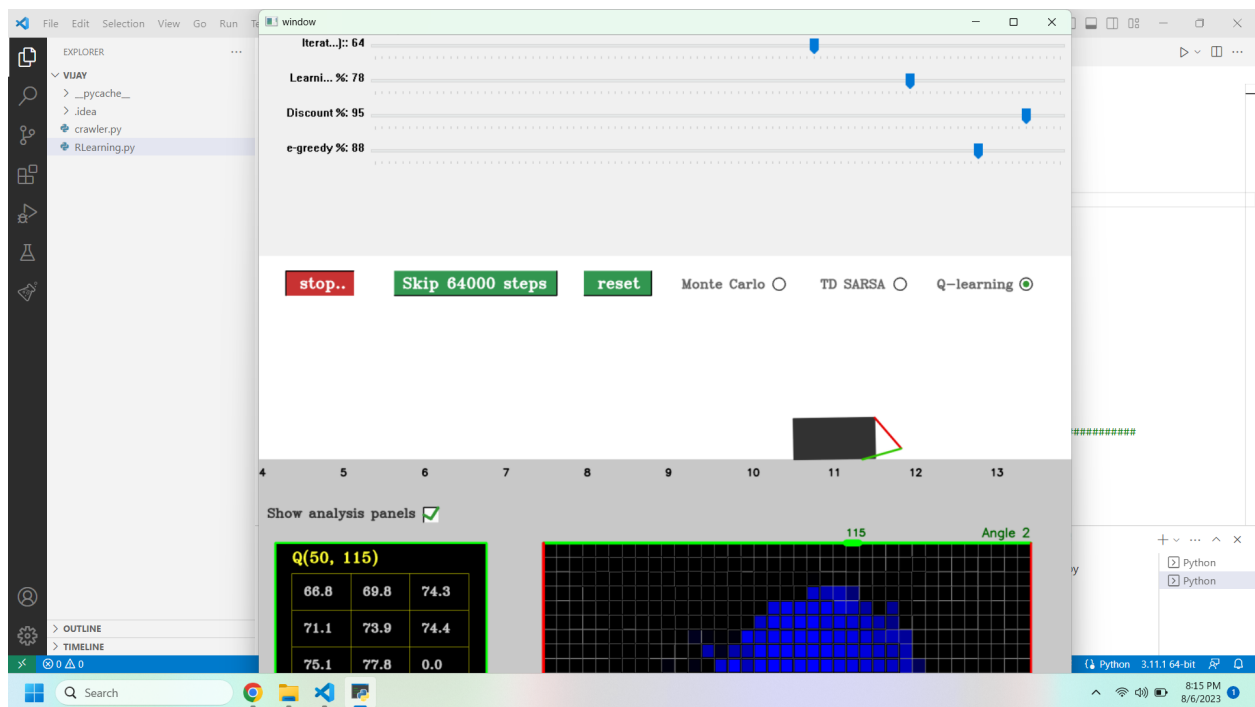
A minute later, the crawler bot has made significant progress in its relative position due to the application of Q-Learning. The relative position of the crawler bot changed by a fair margin.



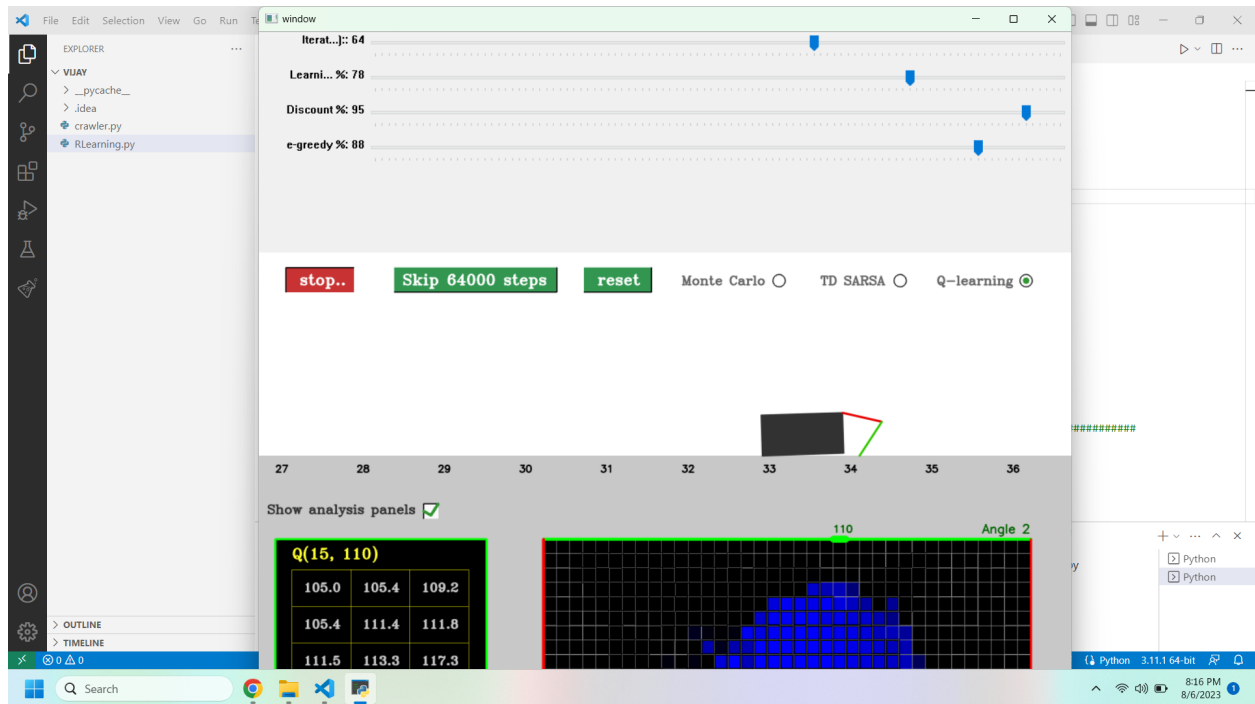
Now, we have decreased the e-greedy to 28% from 48% and will notice the changes in the position of the bot a minute later.



A minute later, we can observe that the change in position is very small. This is because of the decrease in the e-greedy%. As the e-greedy% decreases, the bot undergoes many learnings and there are more states that'll be explored in the process. In short, more learnings are taking place and the bot takes several moments in deciding the next step.



Now, we increase the e-greedy% to 88% from 28% and will notice its positioning, a moment later. The above screenshot captures the start position of the bot when we increased the e-greedy%.



In a matter of a minute, with e-greedy% being higher than ever, the bot has made a drastic progress in its positioning. This is because it'll take learnings from its previous learnings and choose a path that will move forward in a short time. As the e-greedy% is high, the angle2 [shows 110 in the screenshot] also doesn't oscillate much and has only a fixed set of values ranging from 105 to 120 where it moves forward when the ideal value is set.

I hereby conclude that all the requirements are met for QLearning and the project is implemented successfully with no errors.