**Solution 2**

For this Solution, I am using the same model like I have used in Solution 1 but with different way of implementation. Previously I implemented DQN model with the help of TensorFlow but this time I am using stable baseline library to implement the DQN model. As there are so many libraries are present in the market like TF-Agent, stable-baselines, open gym. So, I thought, it would be great to explore these libraries as well.

In this task, there are 3 states So, observation space has increased. So, it will take more time in training as compare to the previous task. To decide for how much episode I should train my model, I did training 2 times, one with huge number of training step and with the help of that I got the idea that certain training step my model is not learning anything, the running reward is not increasing So, with this logic, I decided the number of training step for my final model.

Training Reward plot is given below:



I have tested my model for 50 episodes with 0 epsilon value.

Testing-average = 623.23

Testing-std = 337.52

For me, the main challenge for this task was how to shape reward function so that my model will start learning and for this first I analyzed that if I want to achieve this goal then in which direction I should go. How will I know whether I am going in right direction or not.

So, for this, I compared the state before action and state after performing that action. And with the help of this, I gave the reward to the agent So that it can get the idea that it is going in the right direction.