

**Name(s):** Ziyi Tang

**Netid(s):** ziyet3

**Mean Reward Reached using DQN and DDQN:** (Add Here)

Mean Reward Reached using DQN: 7.42

Mean Reward Reached using DDQN: 14.71

**Uploaded Saved DQN/DDQN Model on Canvas (whichever performs better) :** (Yes if you uploaded, No if not)

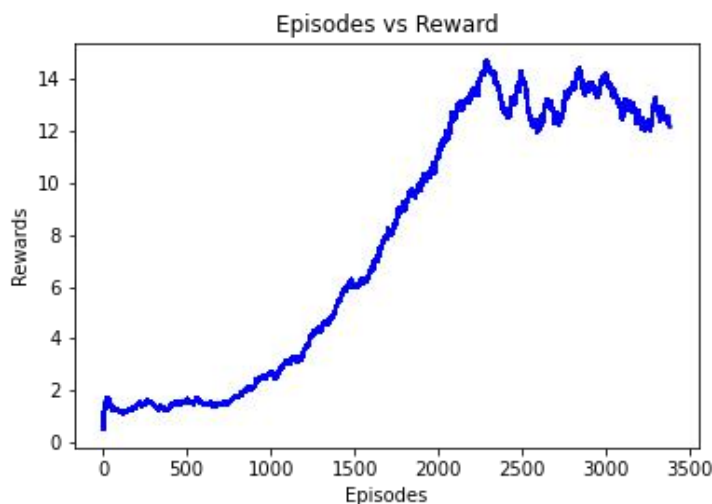
Yes

**Uploaded your Agent.py and Agent\_double.py file on Canvas :** (Yes if you uploaded, No if not)

Yes

**Plot of Mean Evaluation Reward for the model that reaches the target score (Either DQN or DDQN):** (Add Here)

Plot for DDQN:



**Provide a few sentences to analyze the training process and talk about some implementation details:** (Add Here)

The training process of this DQN agent is implemented using a replay memory and a target network, as well as double DQN modifications to address the overestimation problem in DQNs. The agent follows an epsilon-greedy policy to explore and exploit the environment, and uses the Adam optimizer with a Smooth L1 loss function to optimize the policy network. The implementation also includes a learning rate scheduler to adjust the learning rate over time. The `update_target_net()` method is called at a fixed interval to update the weights of the target

network to match those of the policy network. The training process includes clipping the gradients to a maximum value of 100 to prevent gradient explosion.

### **Extra Credit**

**Answer the questions accordingly if you did the corresponding part. The questions are just prompts. You should elaborate a bit more if you can.**

1. What games did you apply the extra credit to? How does it work?
2. What other algorithm did you use? Explain and cite all your sources. Any issues you got in training your new algorithm.