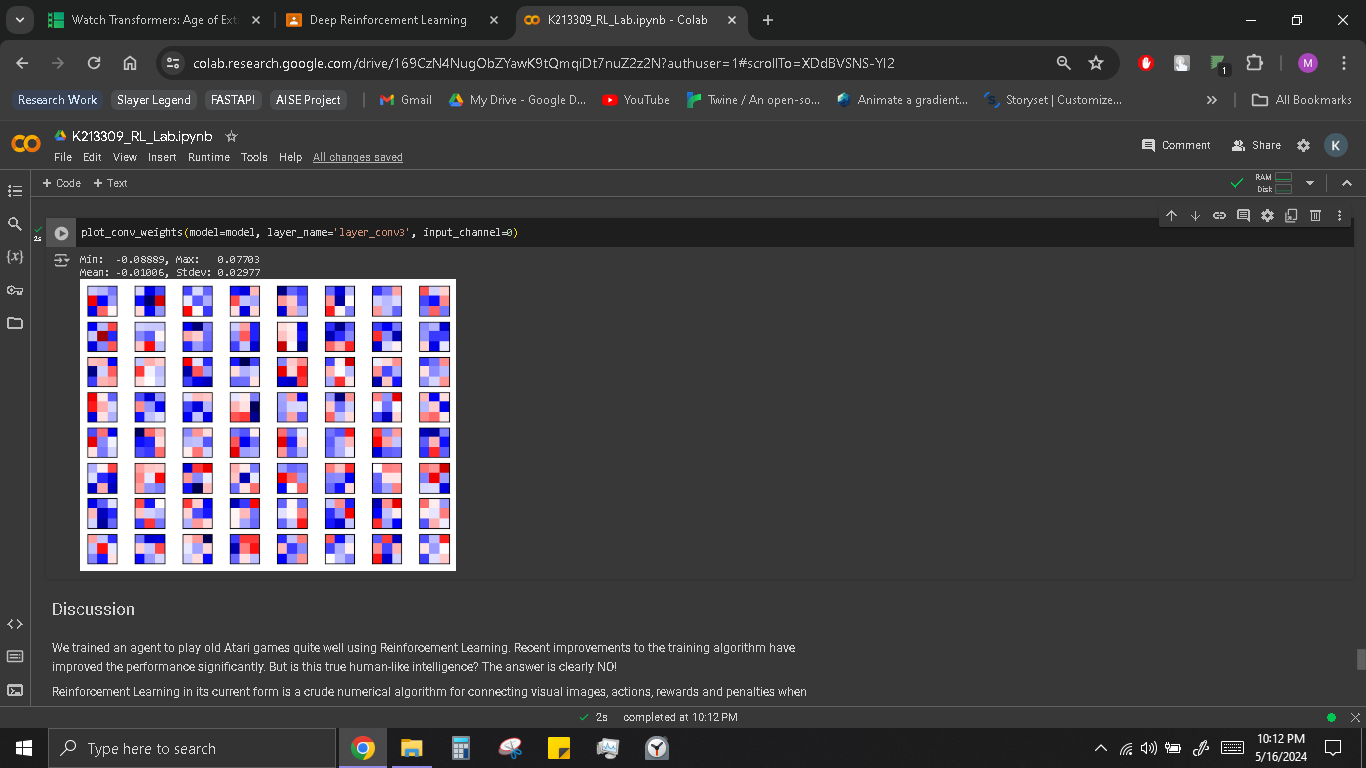
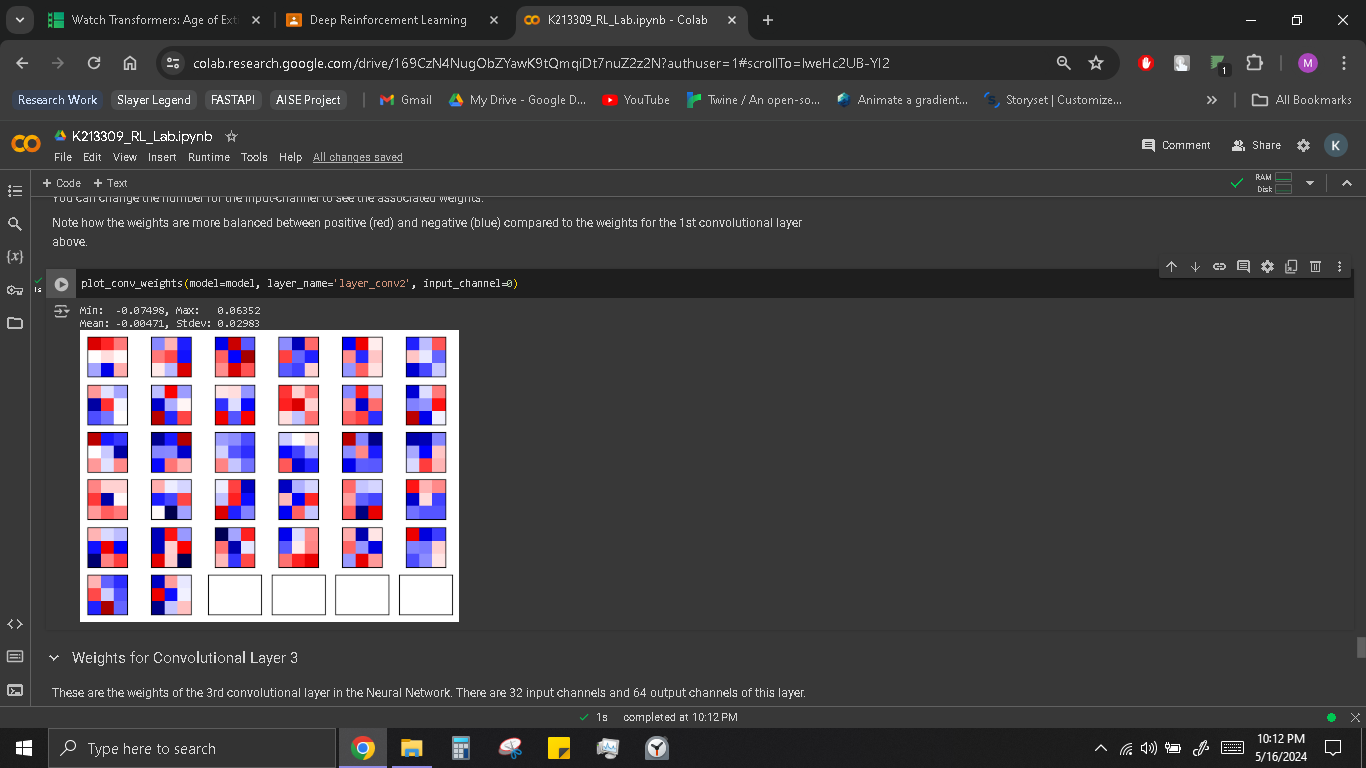
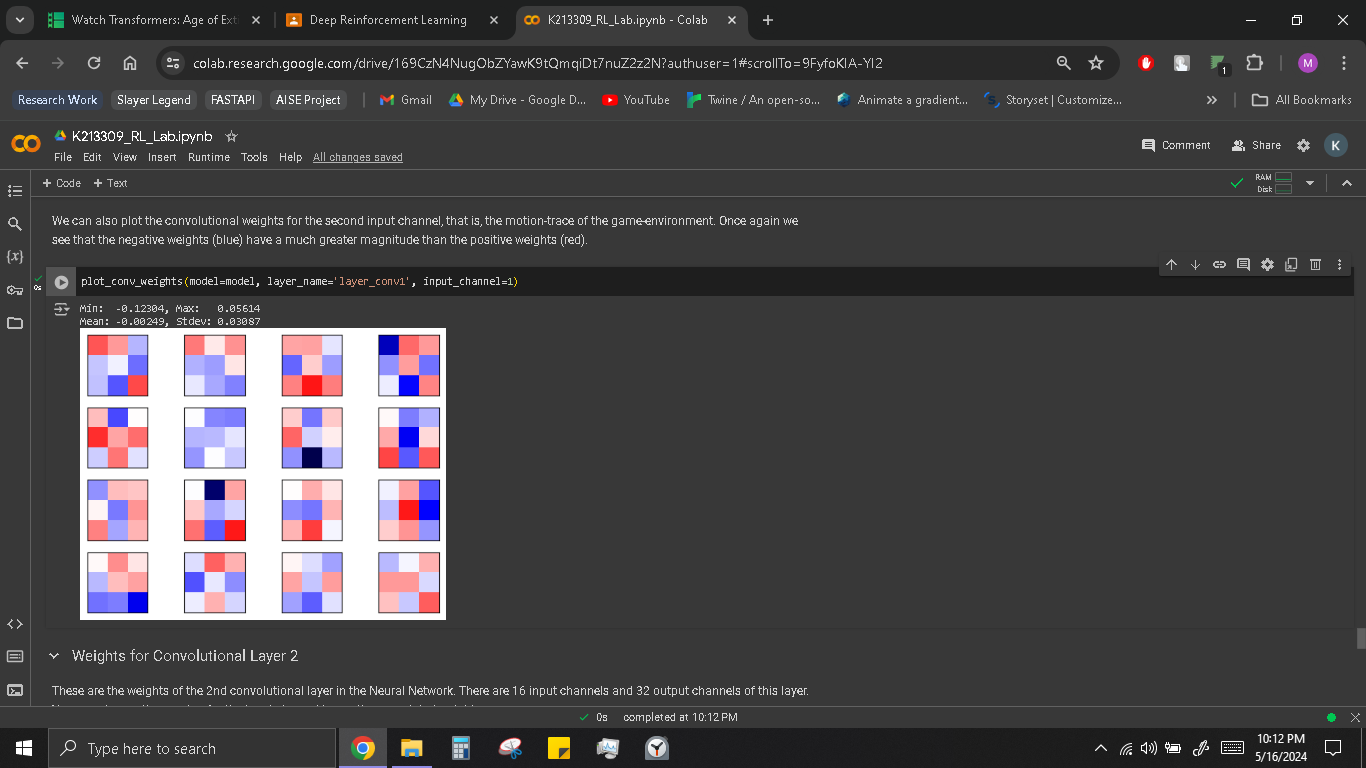
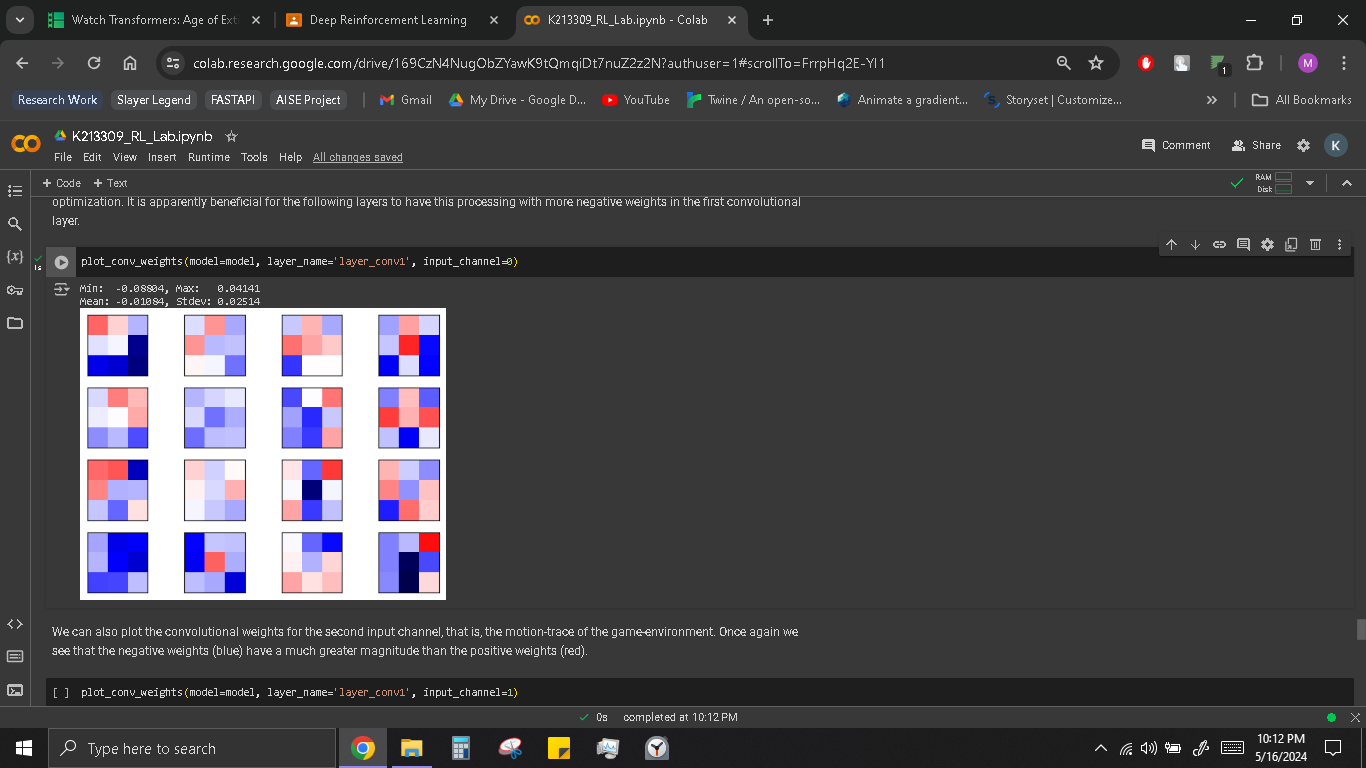
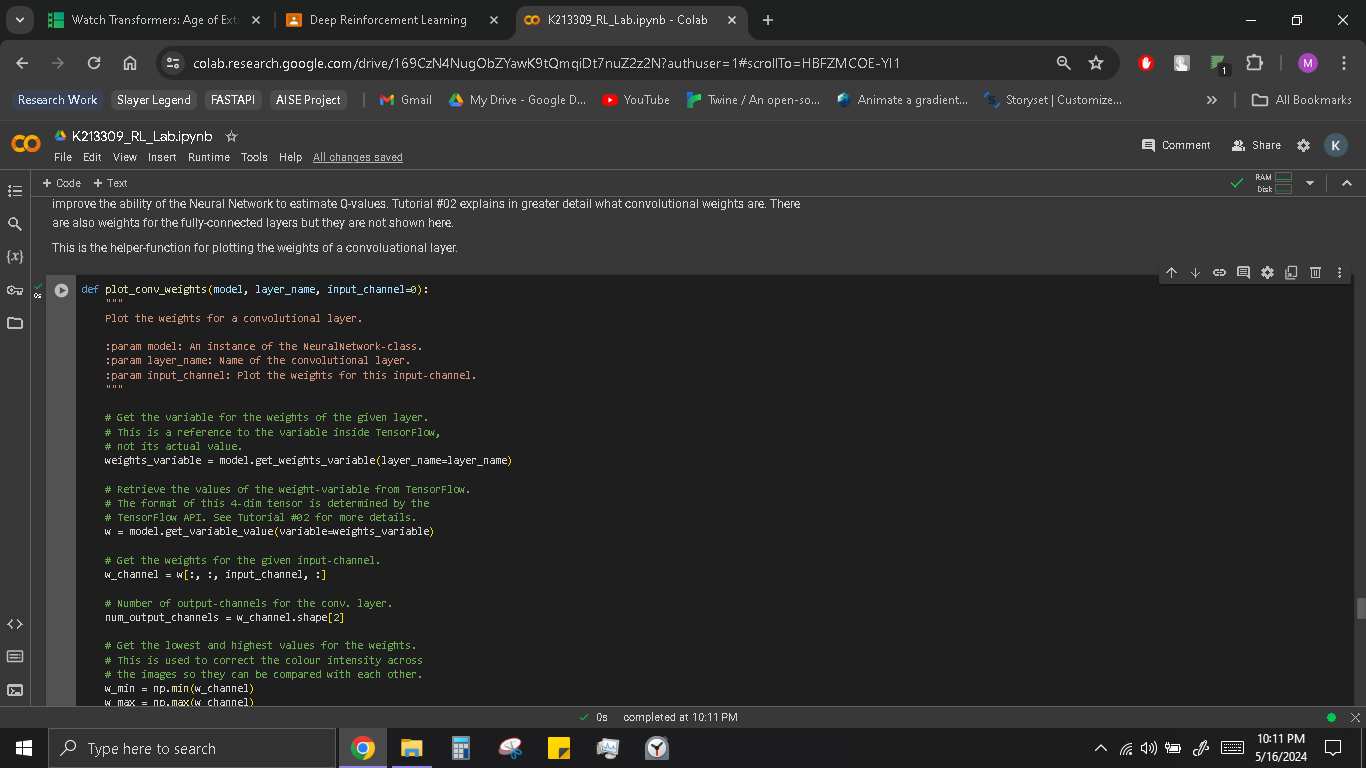
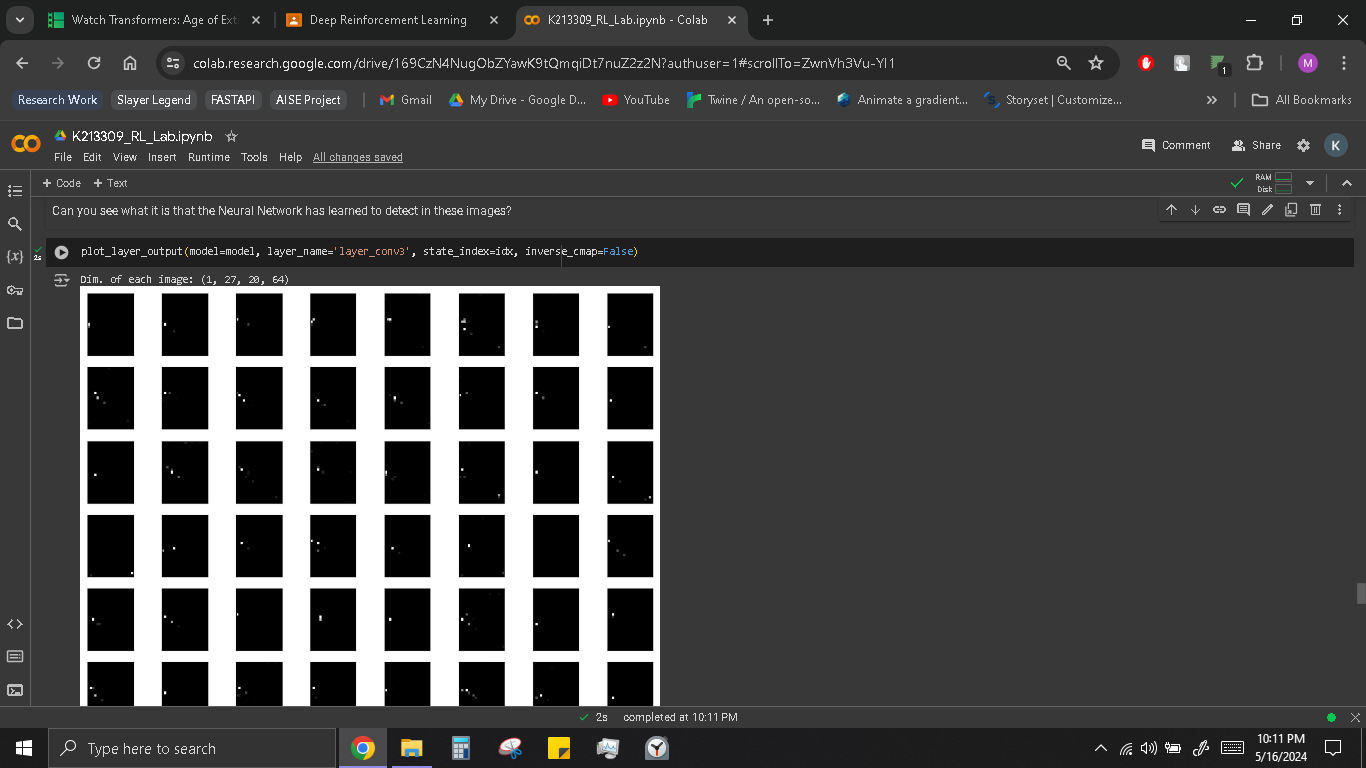
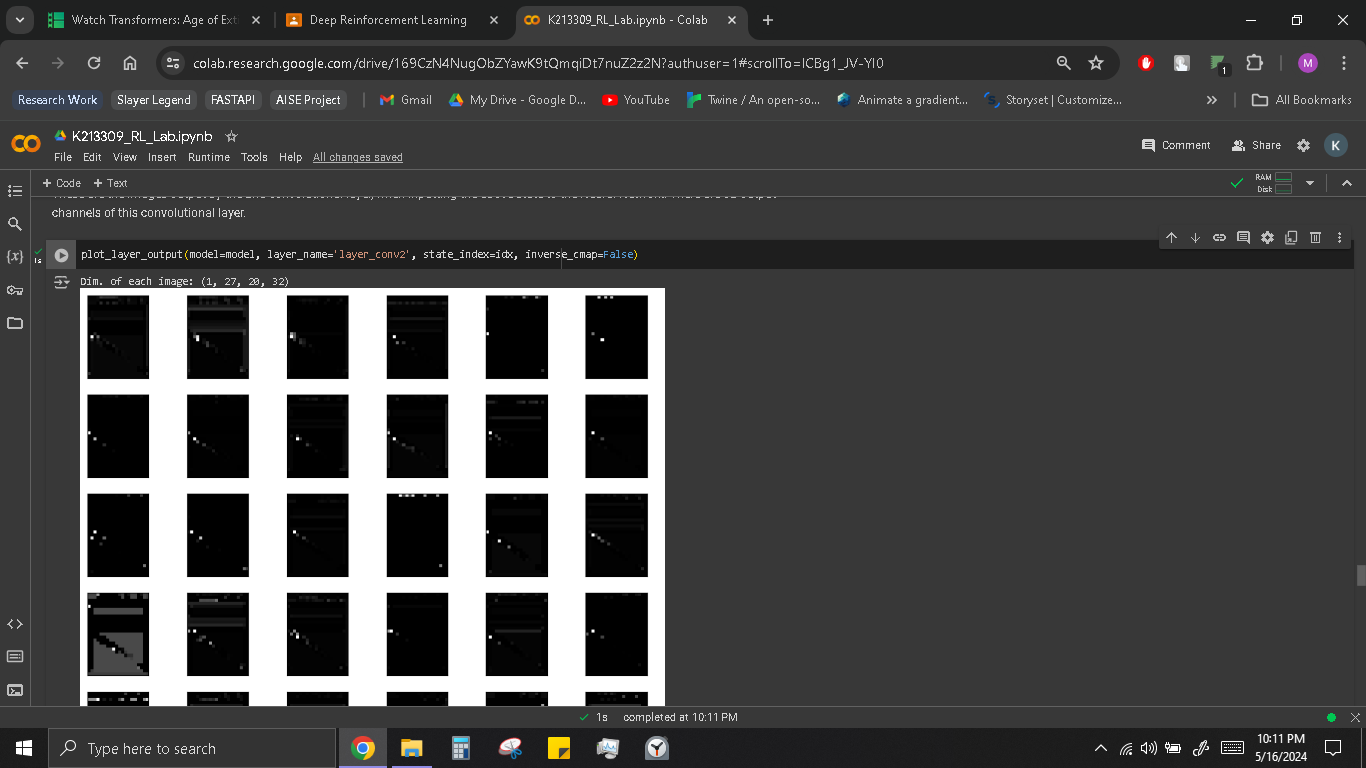
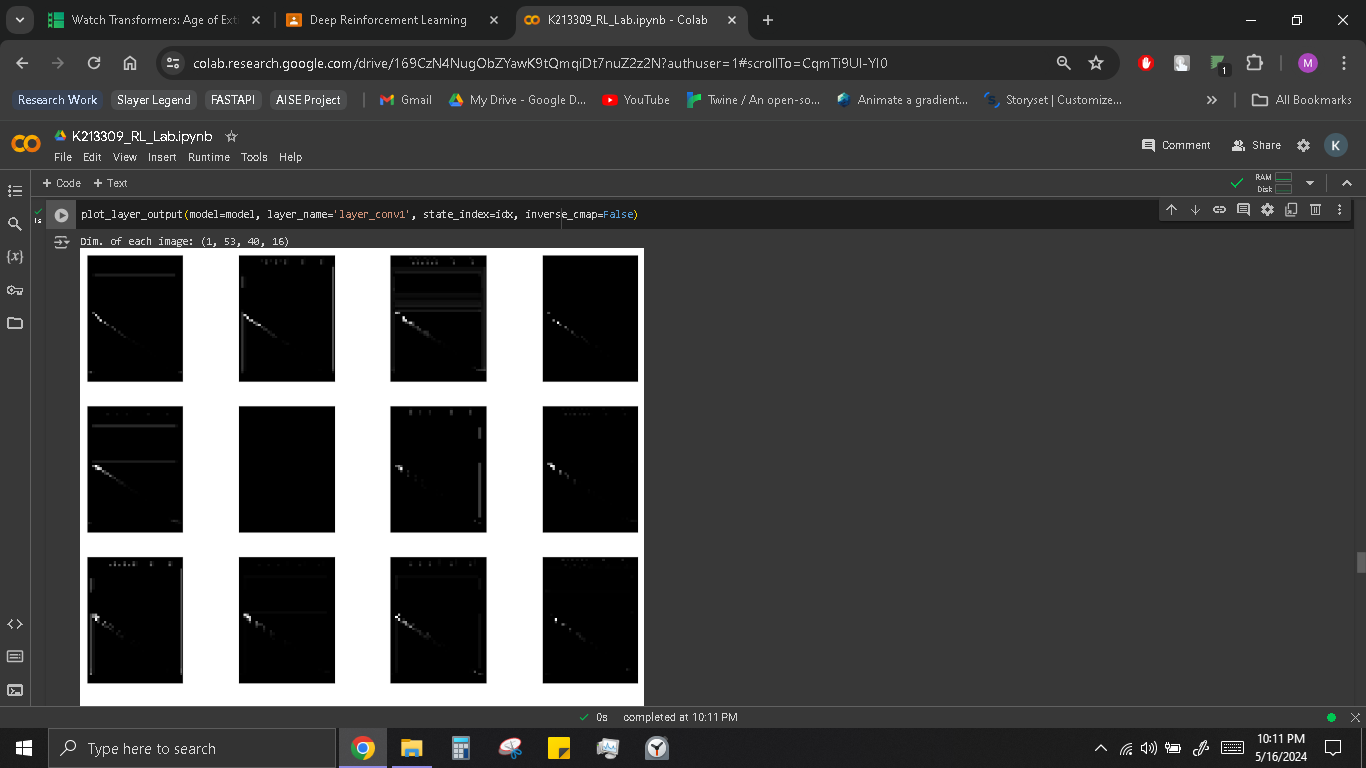
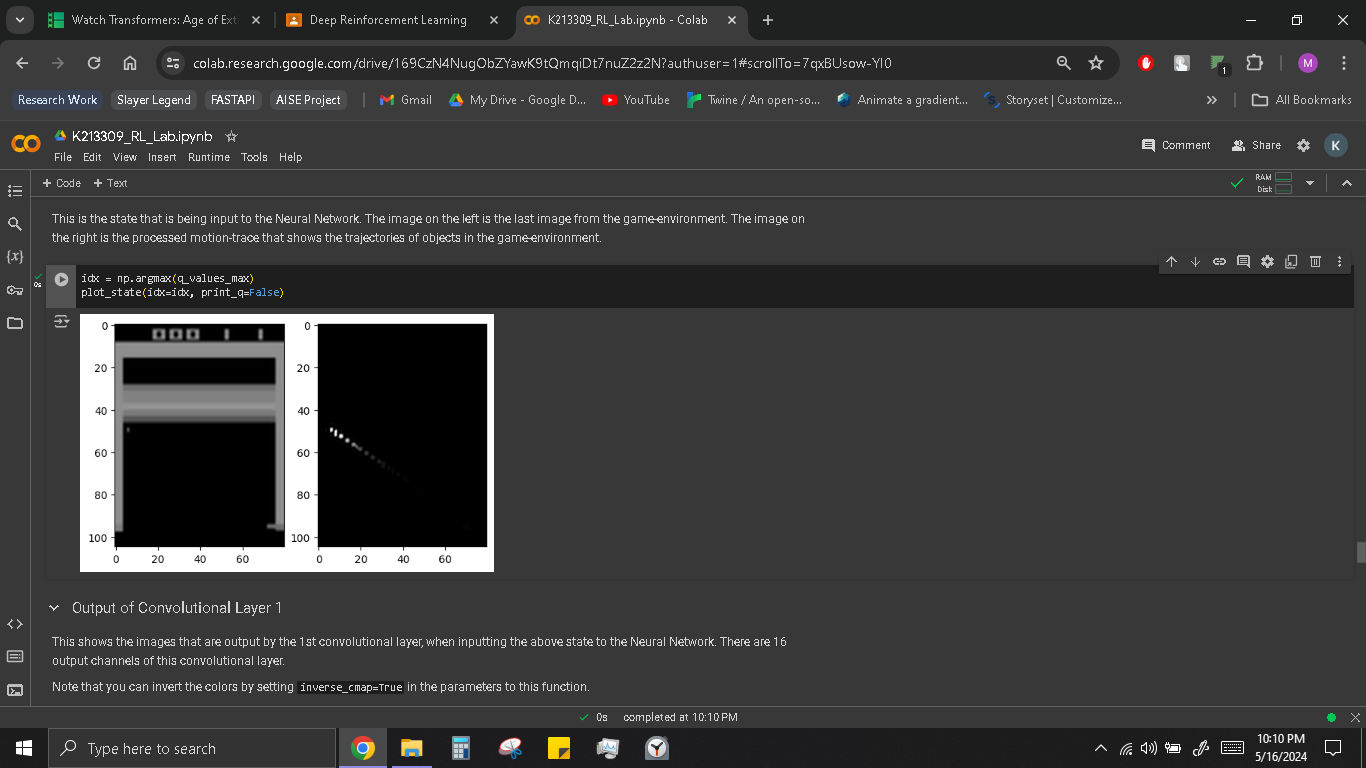
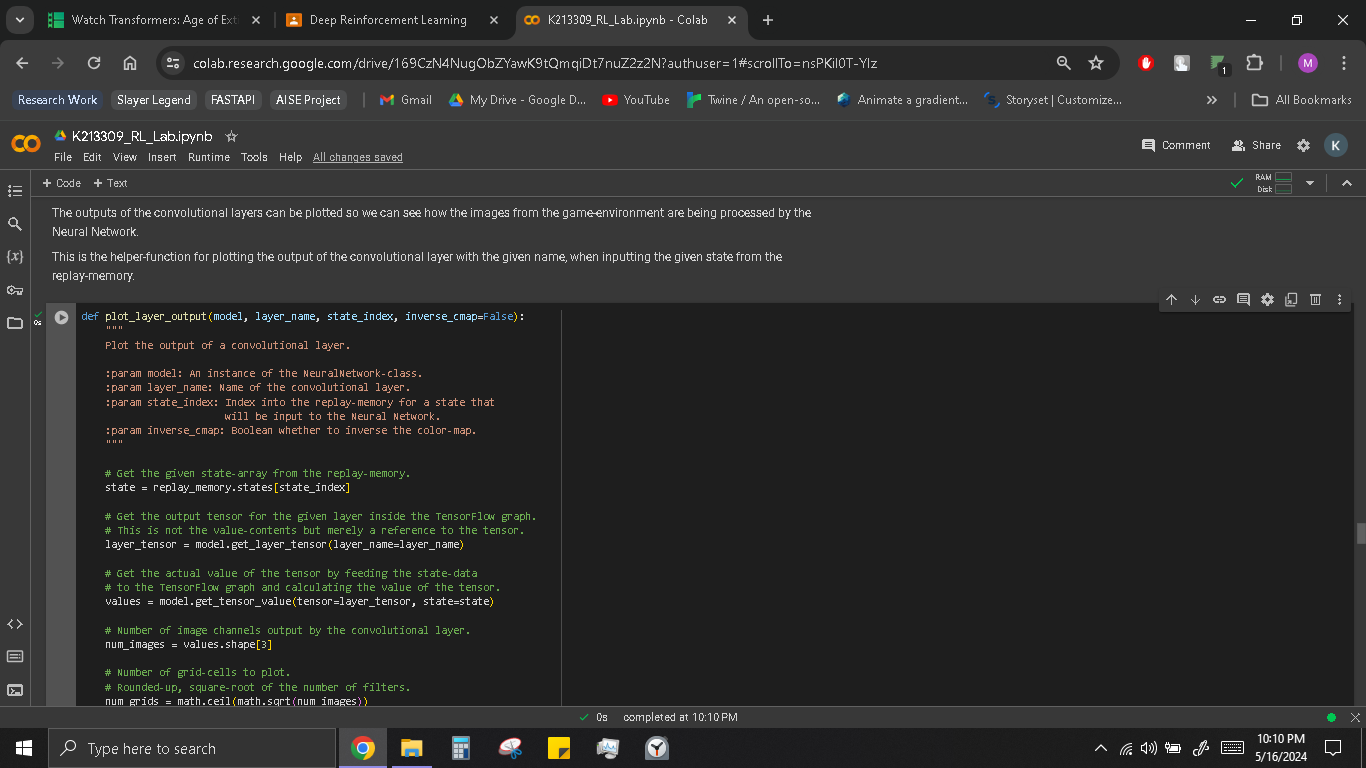
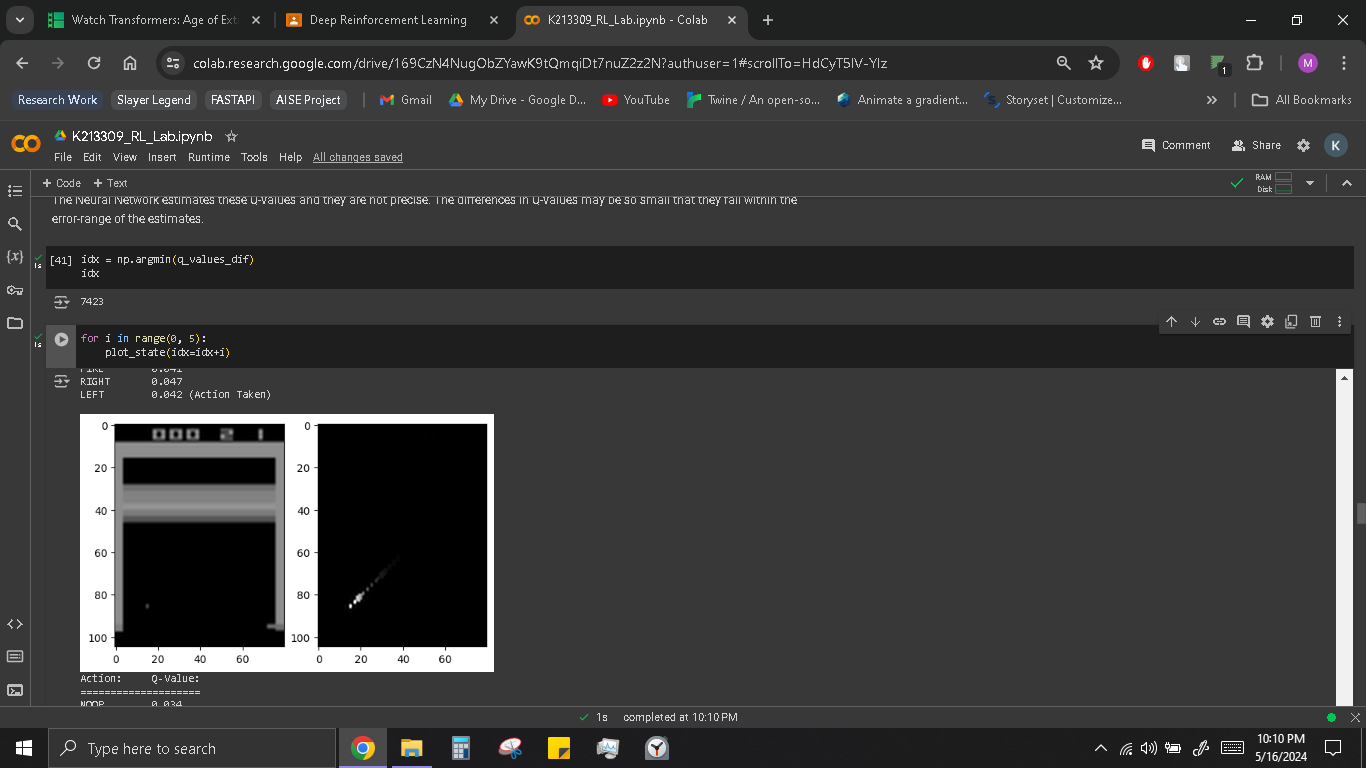
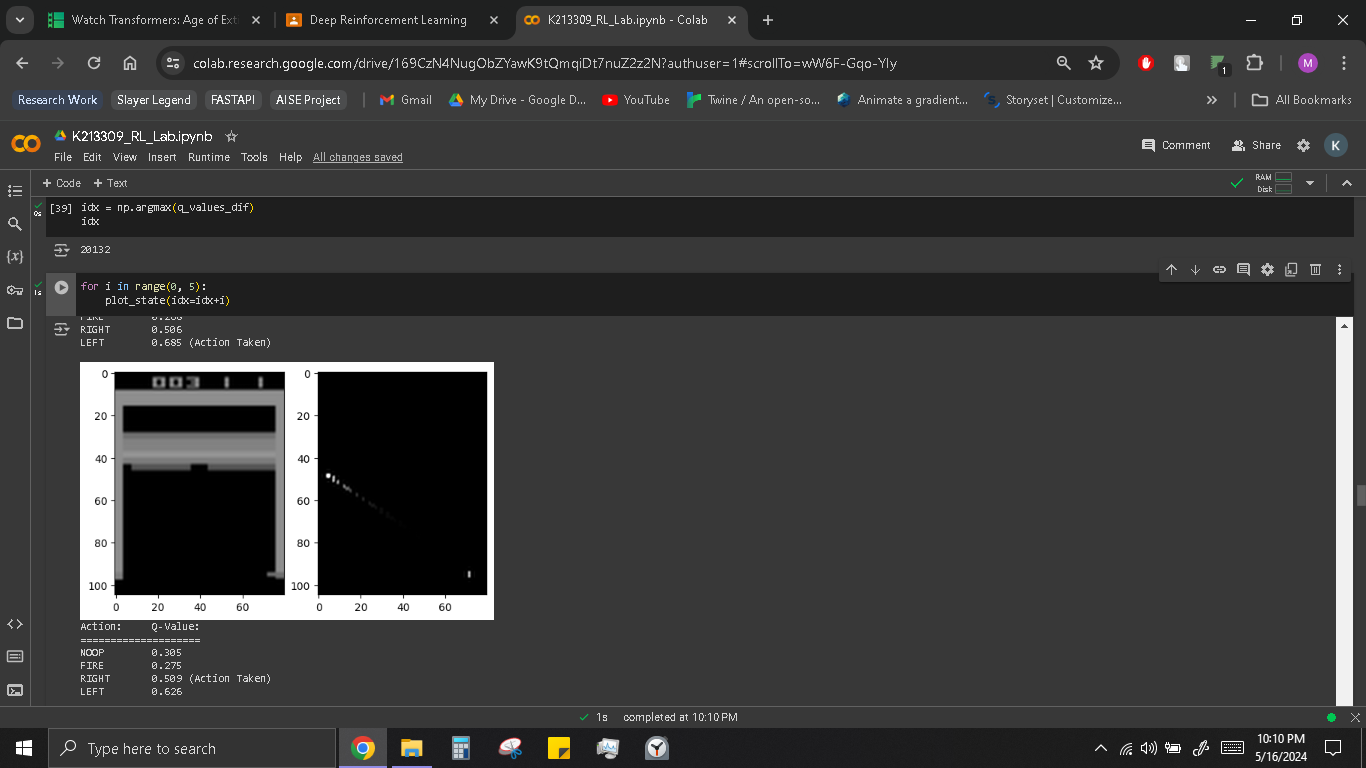
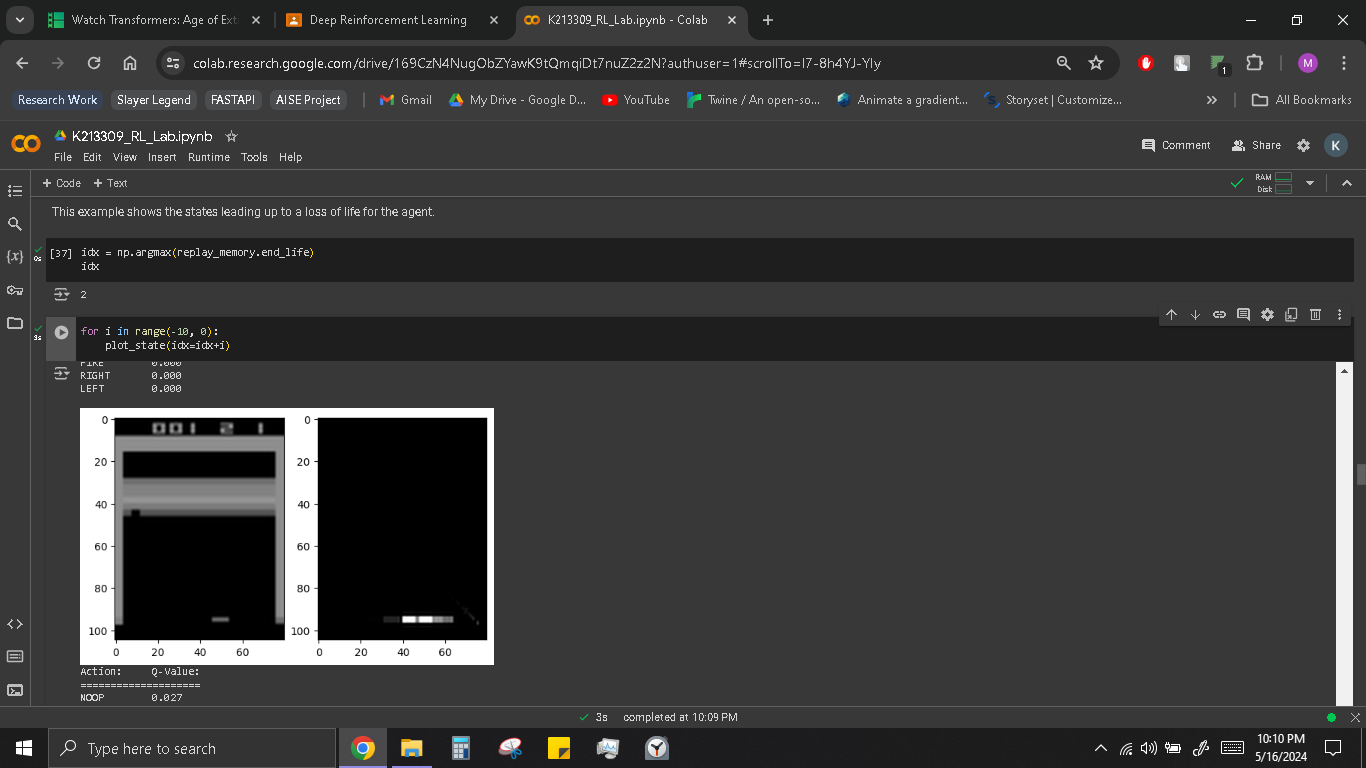
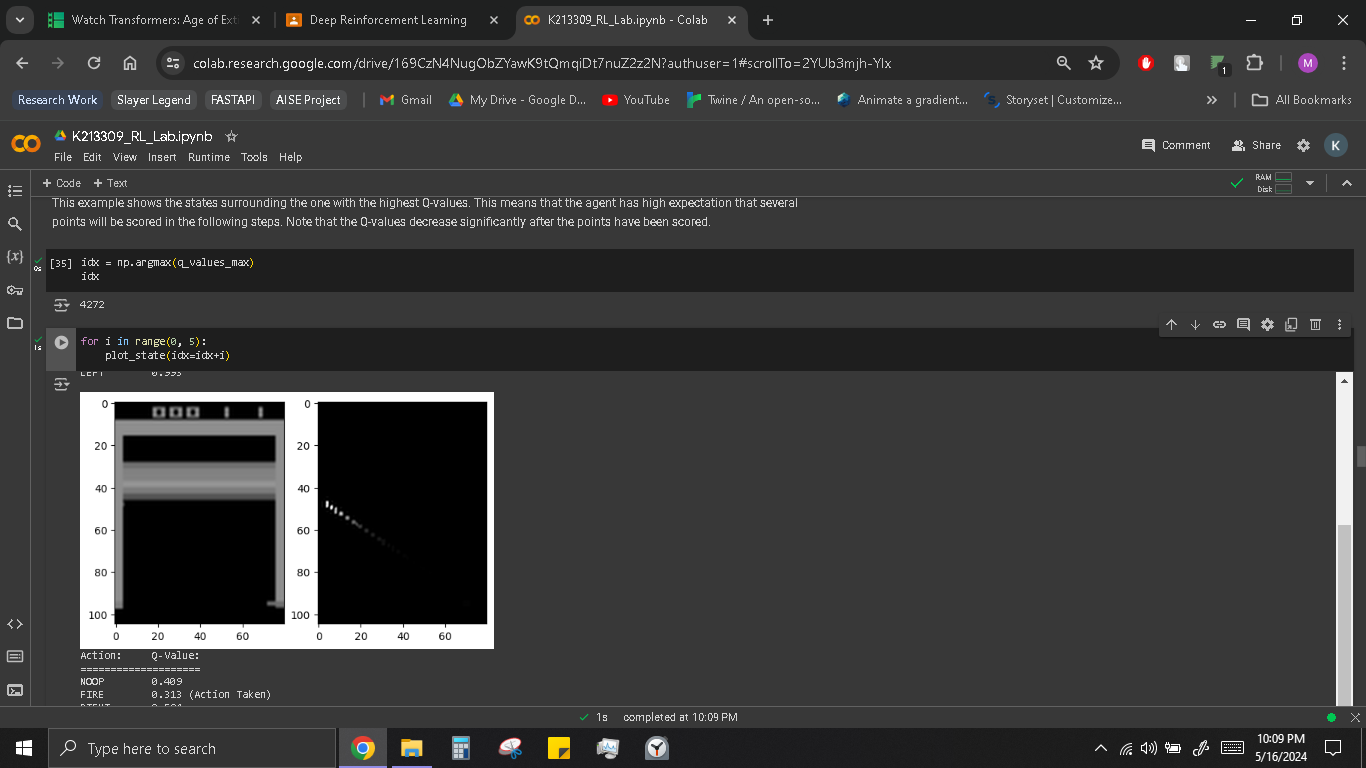
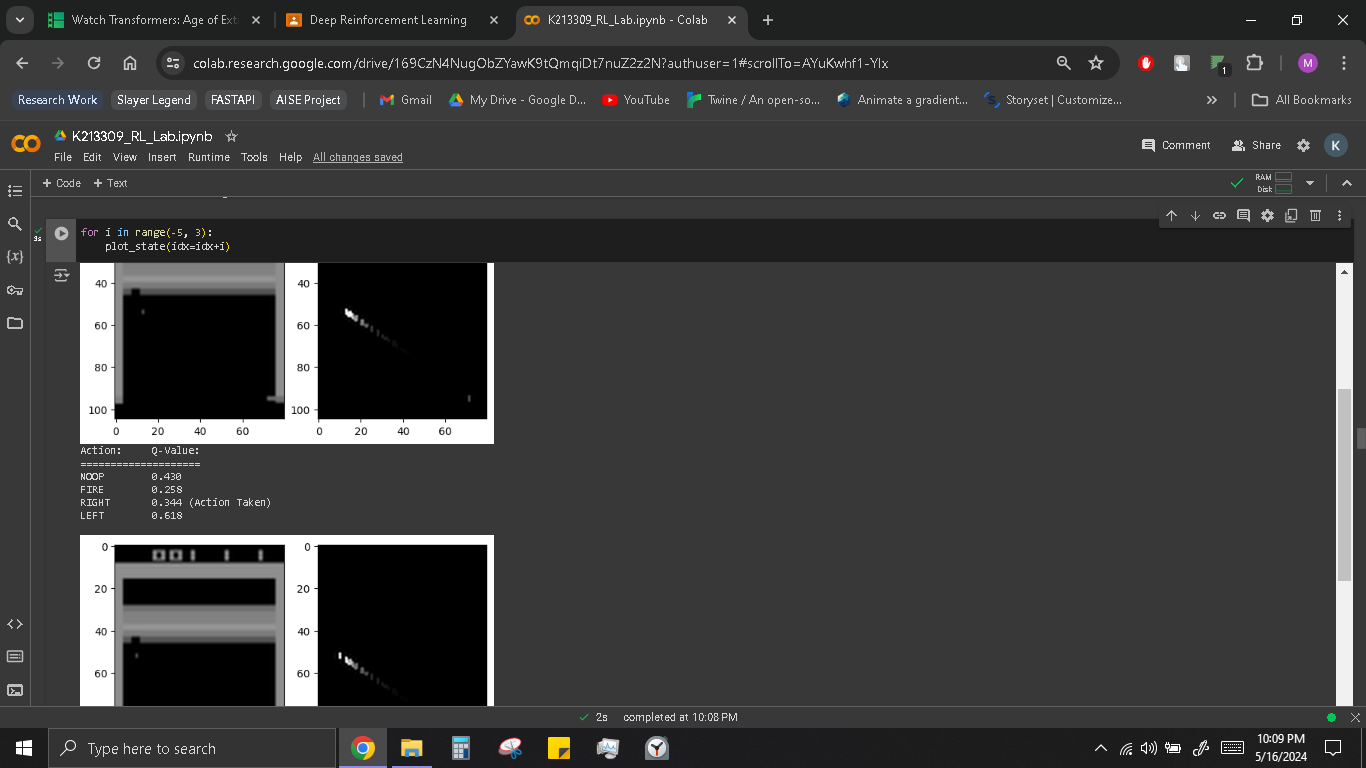
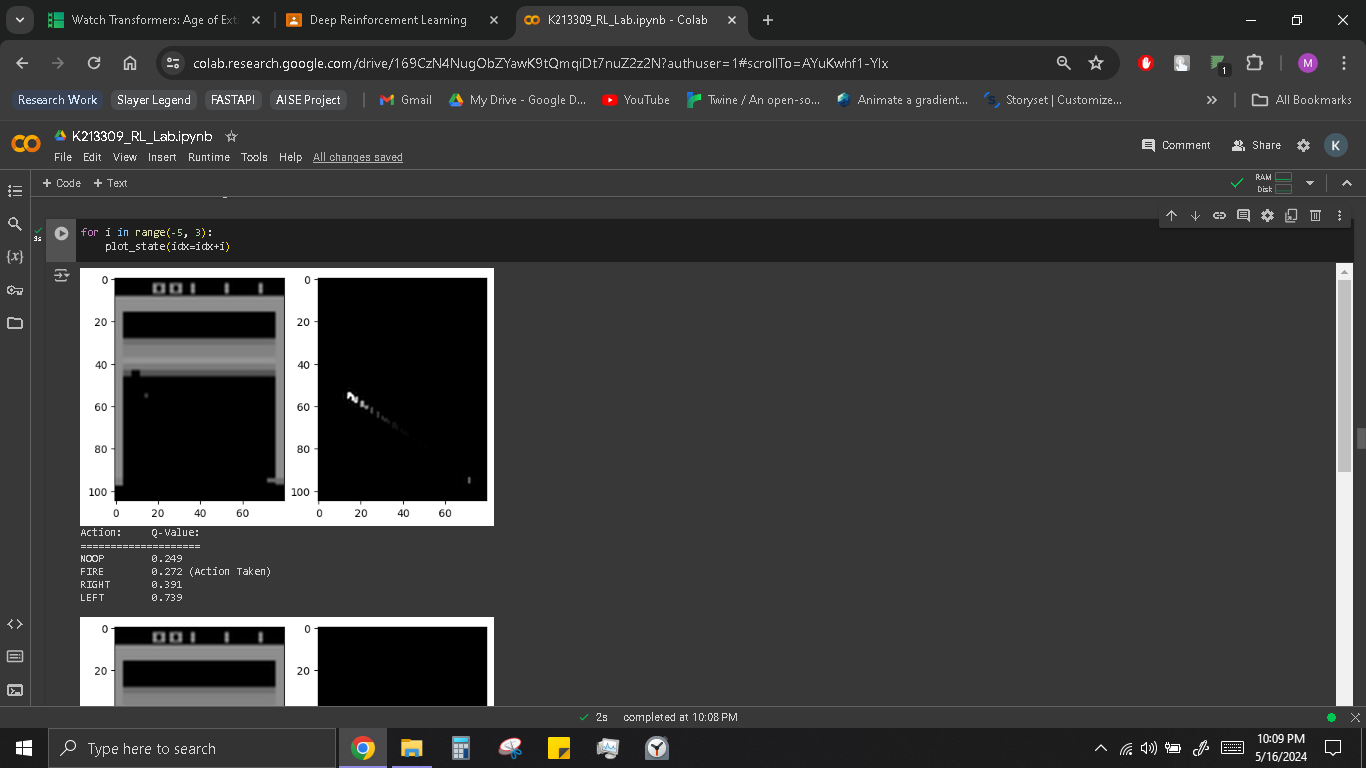
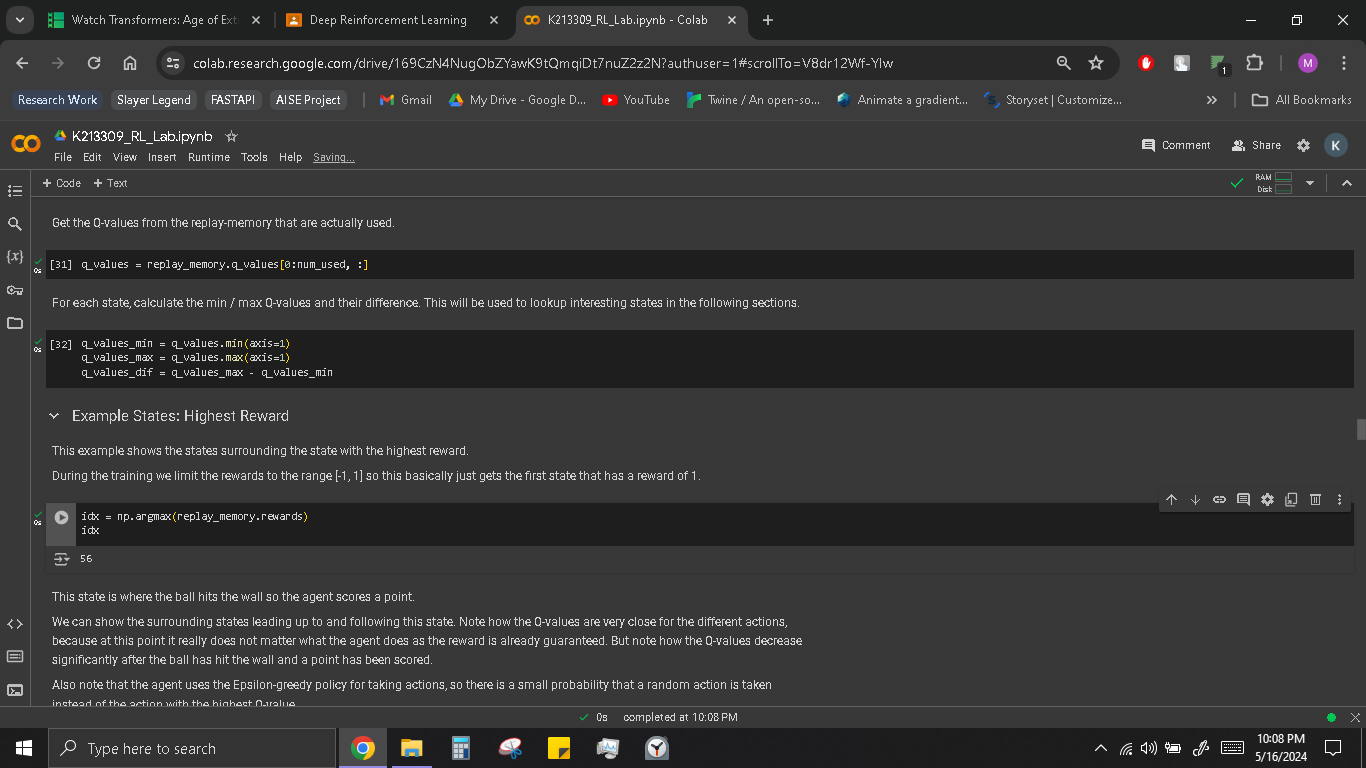
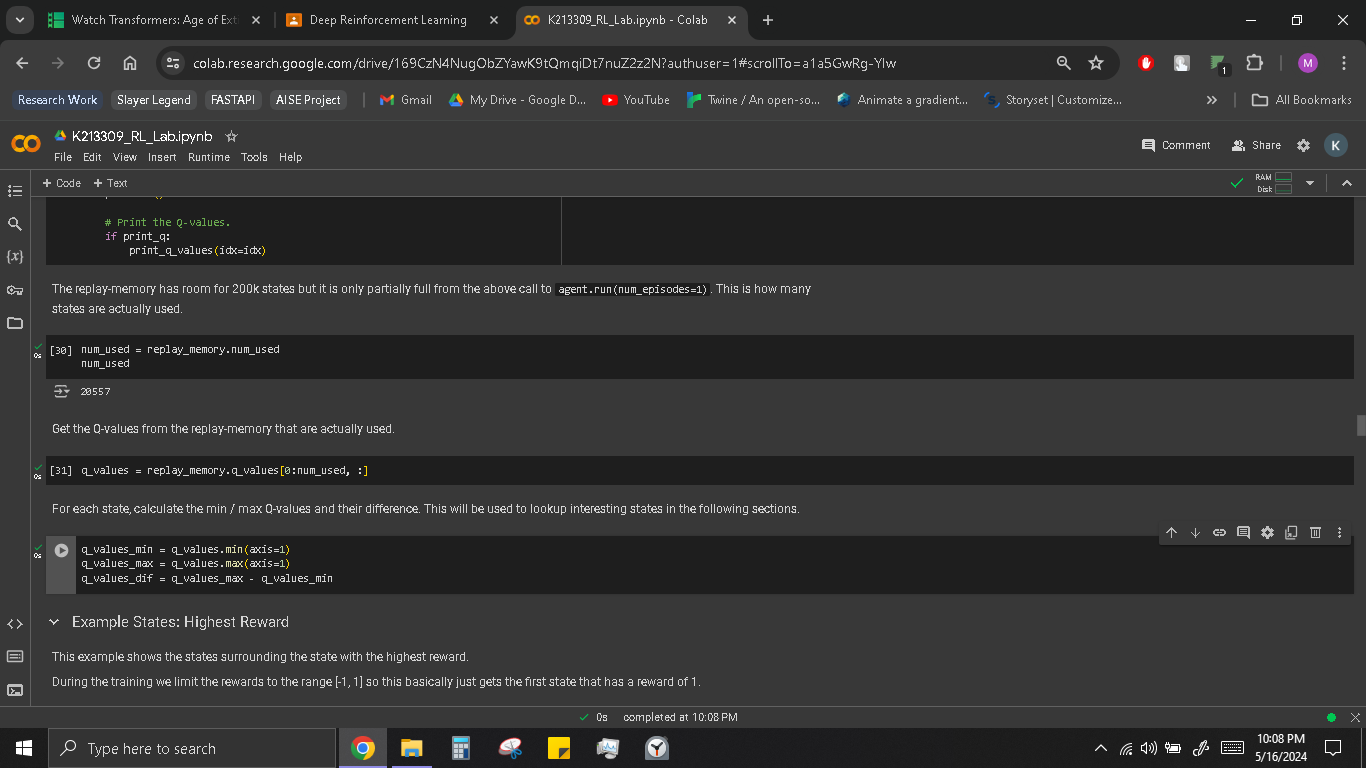
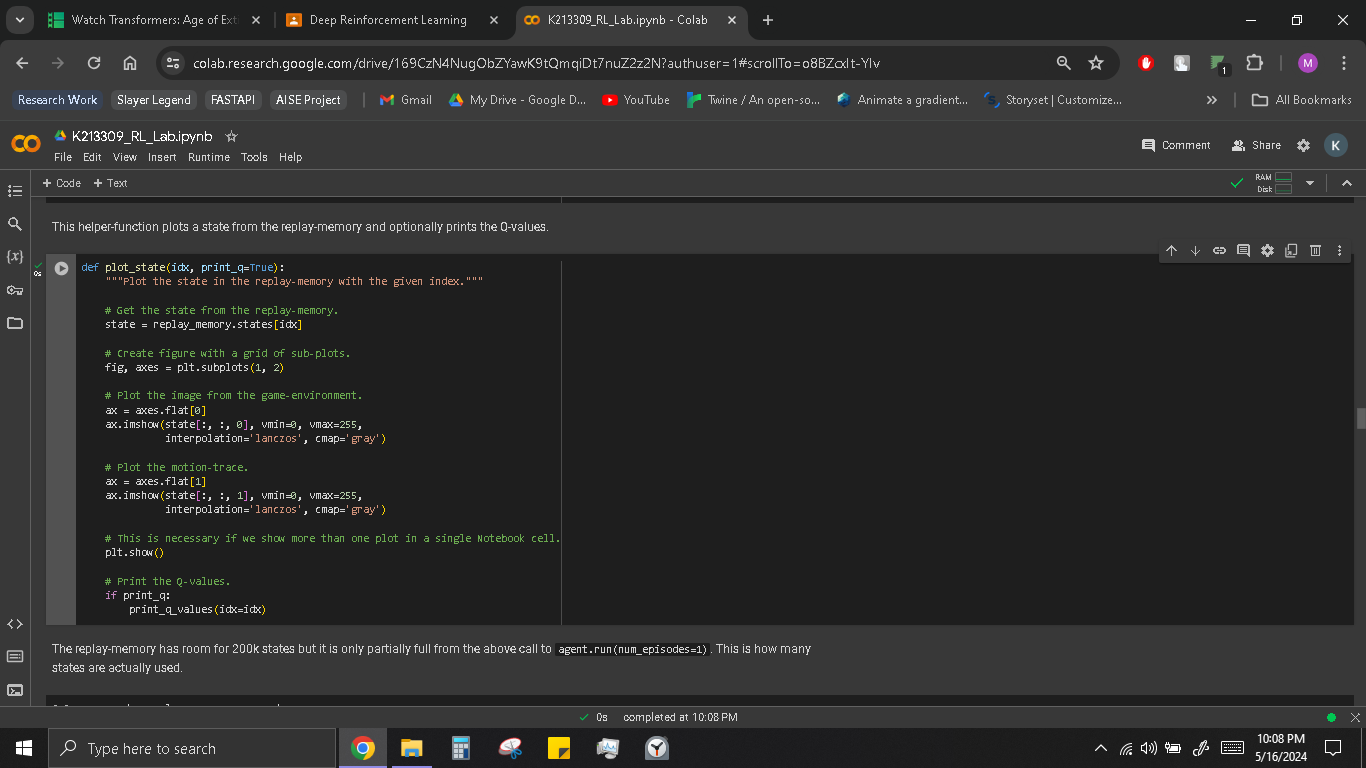
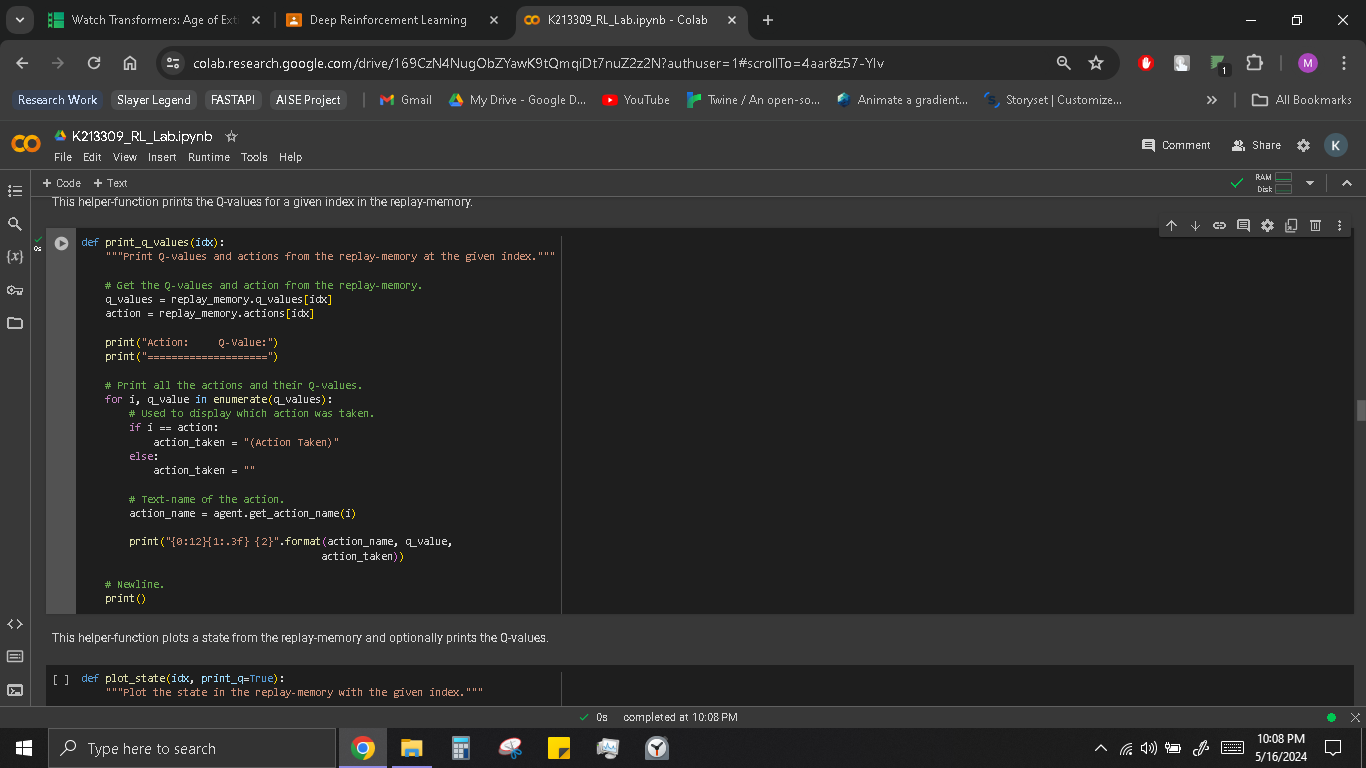
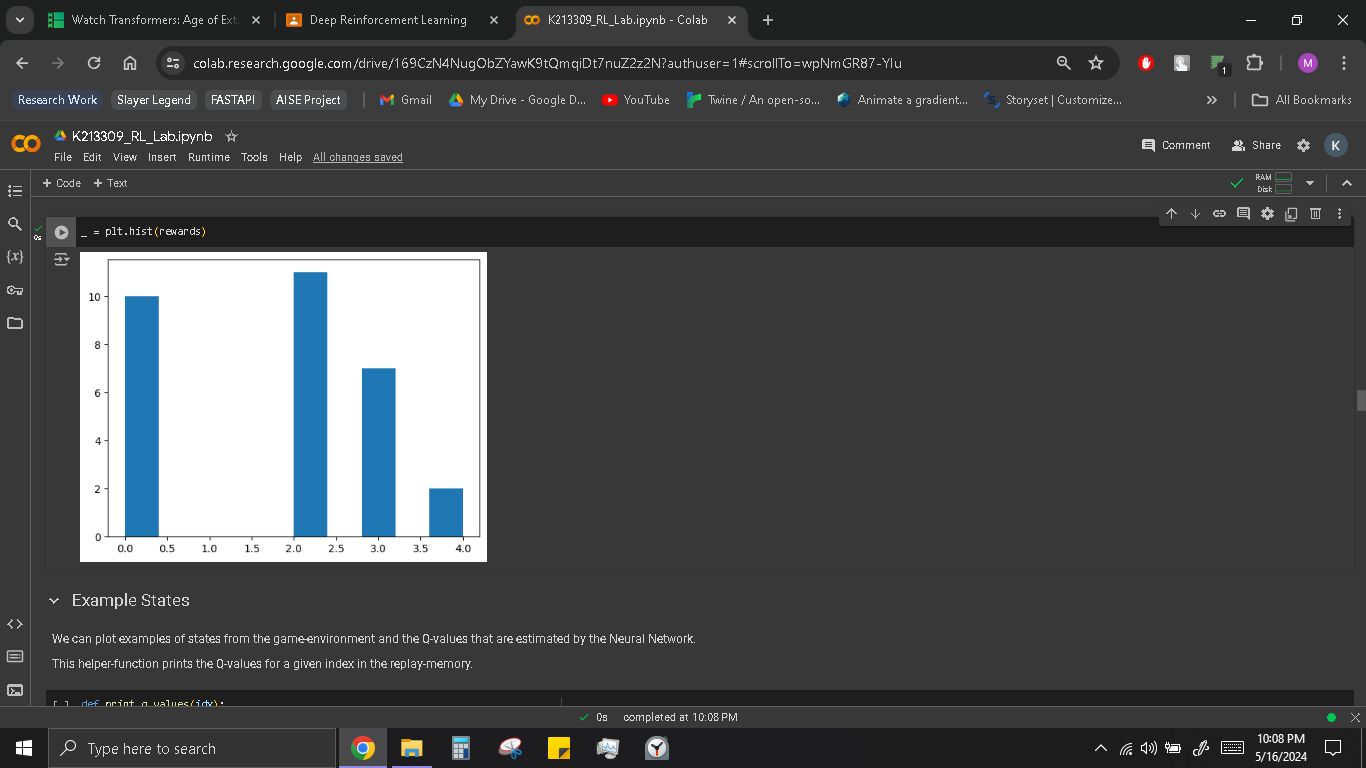
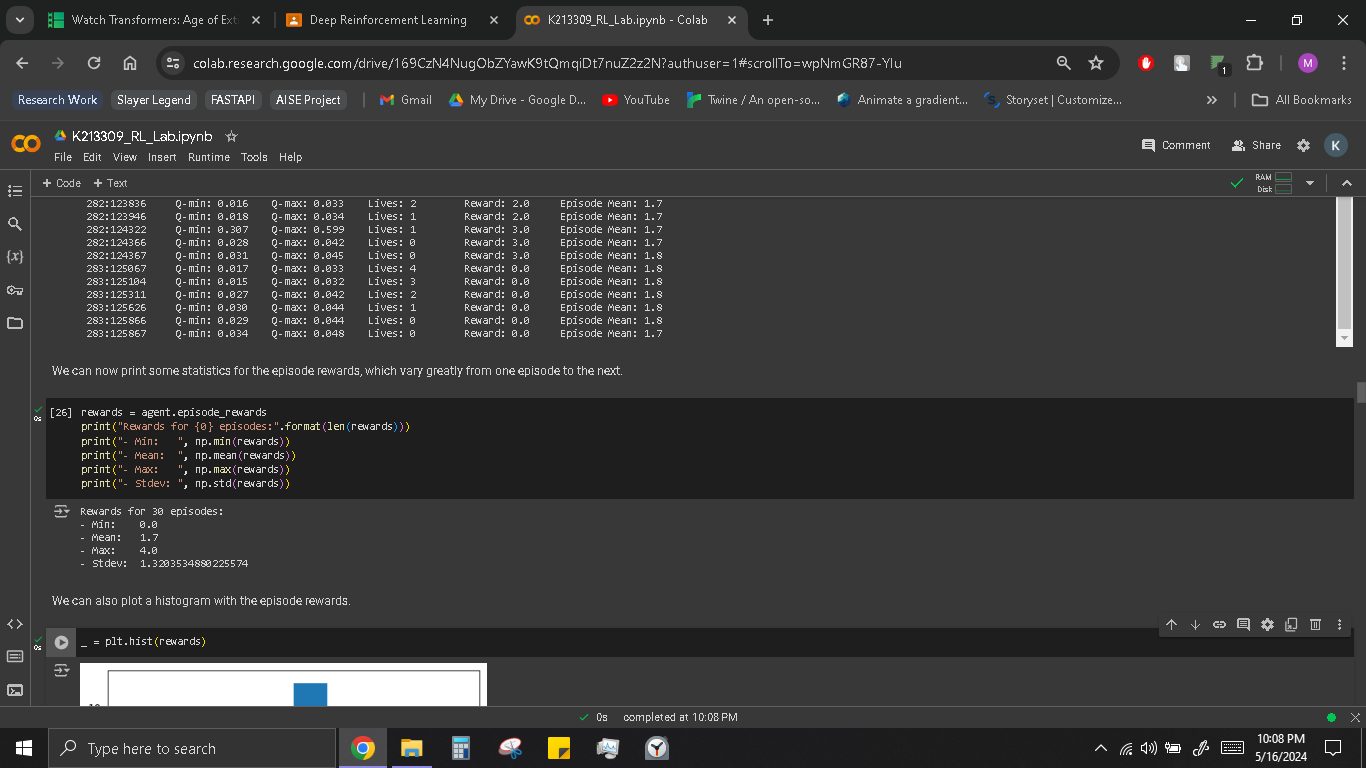
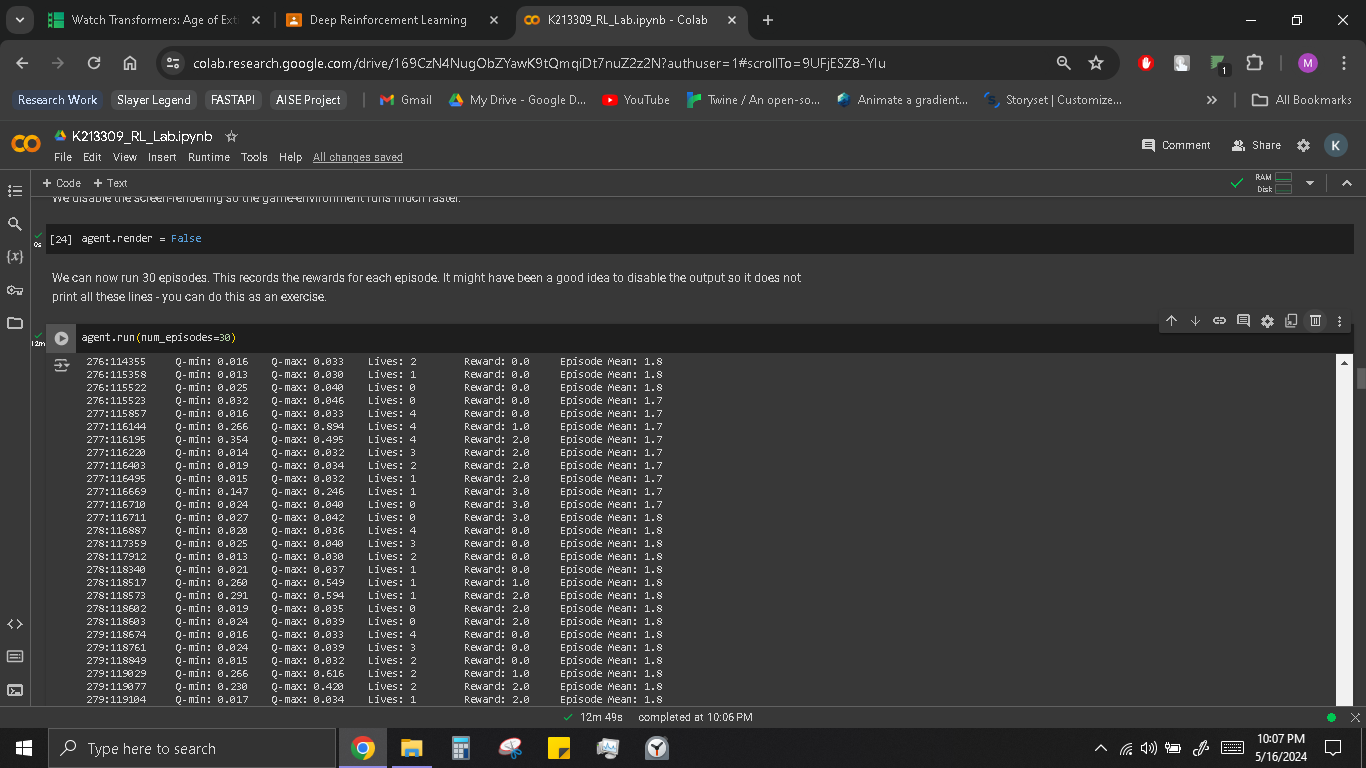
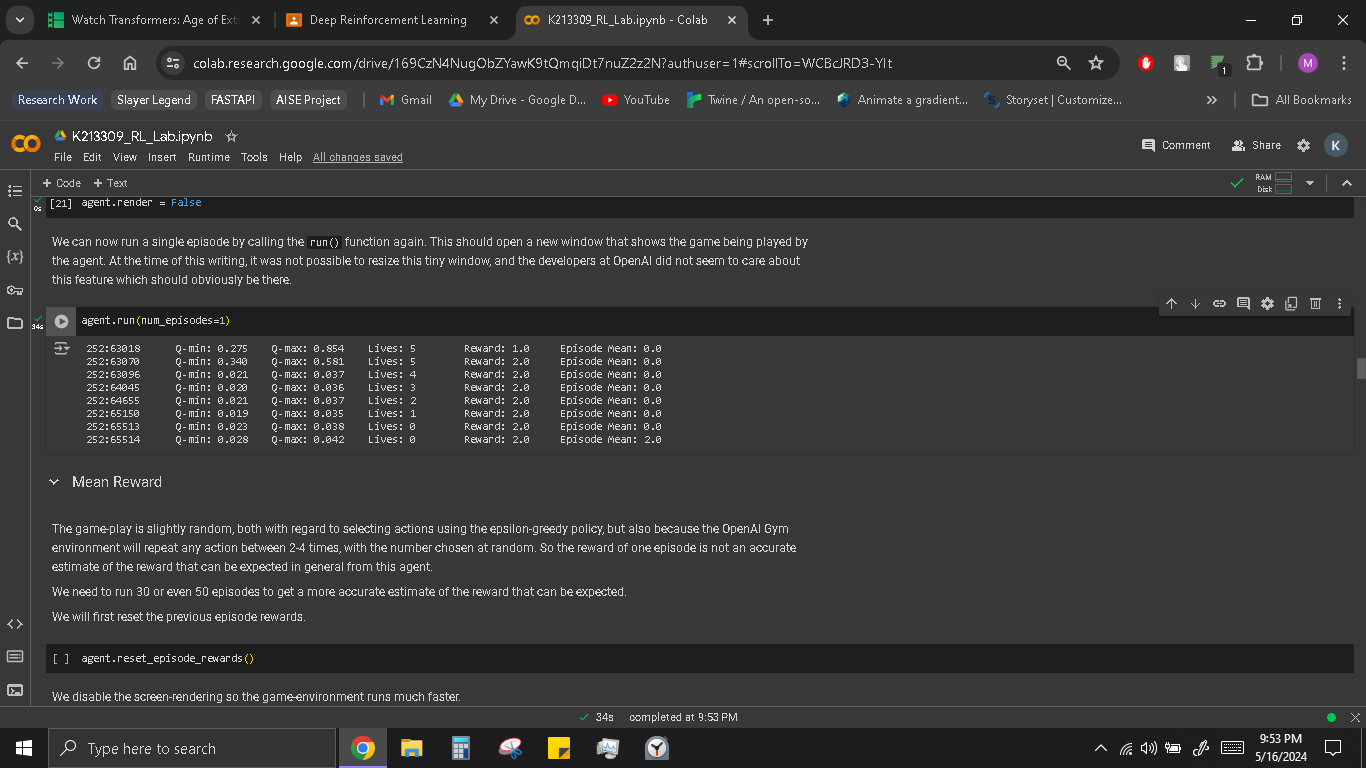
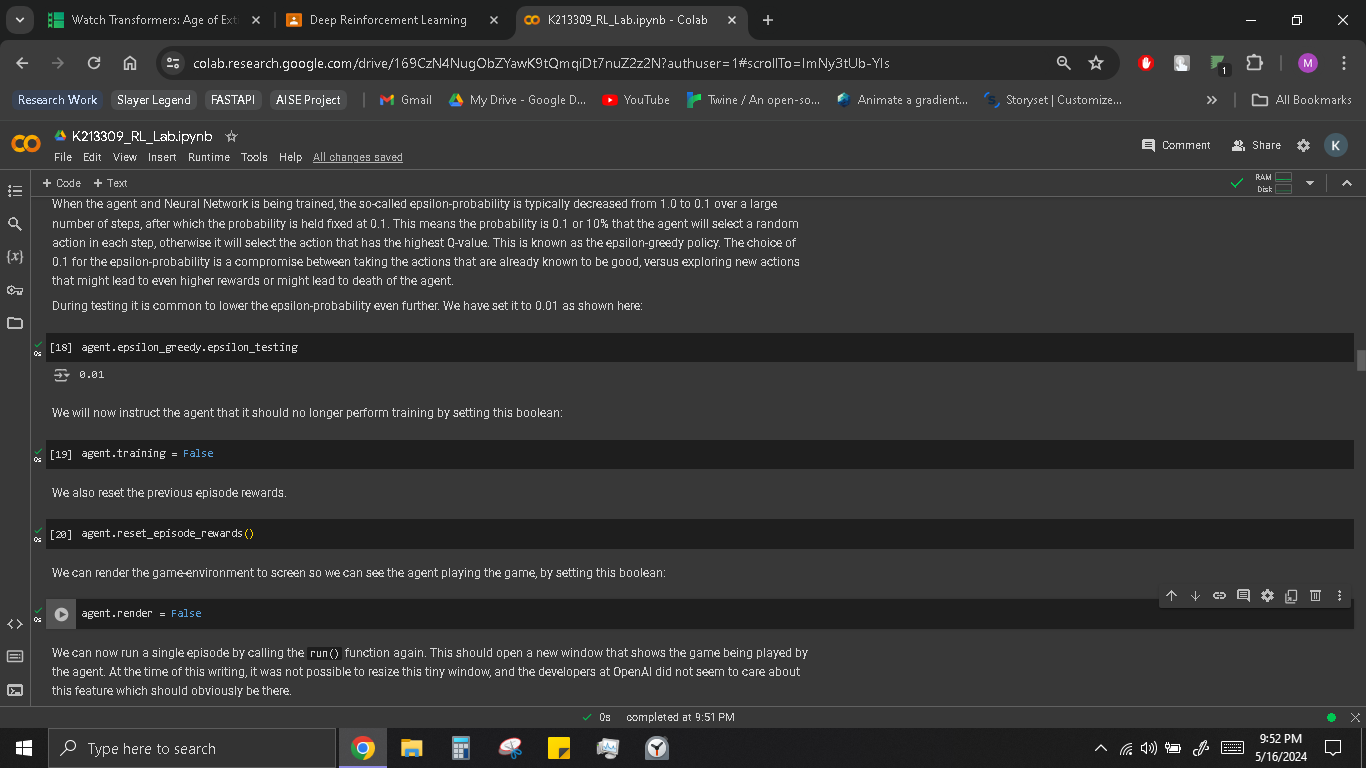
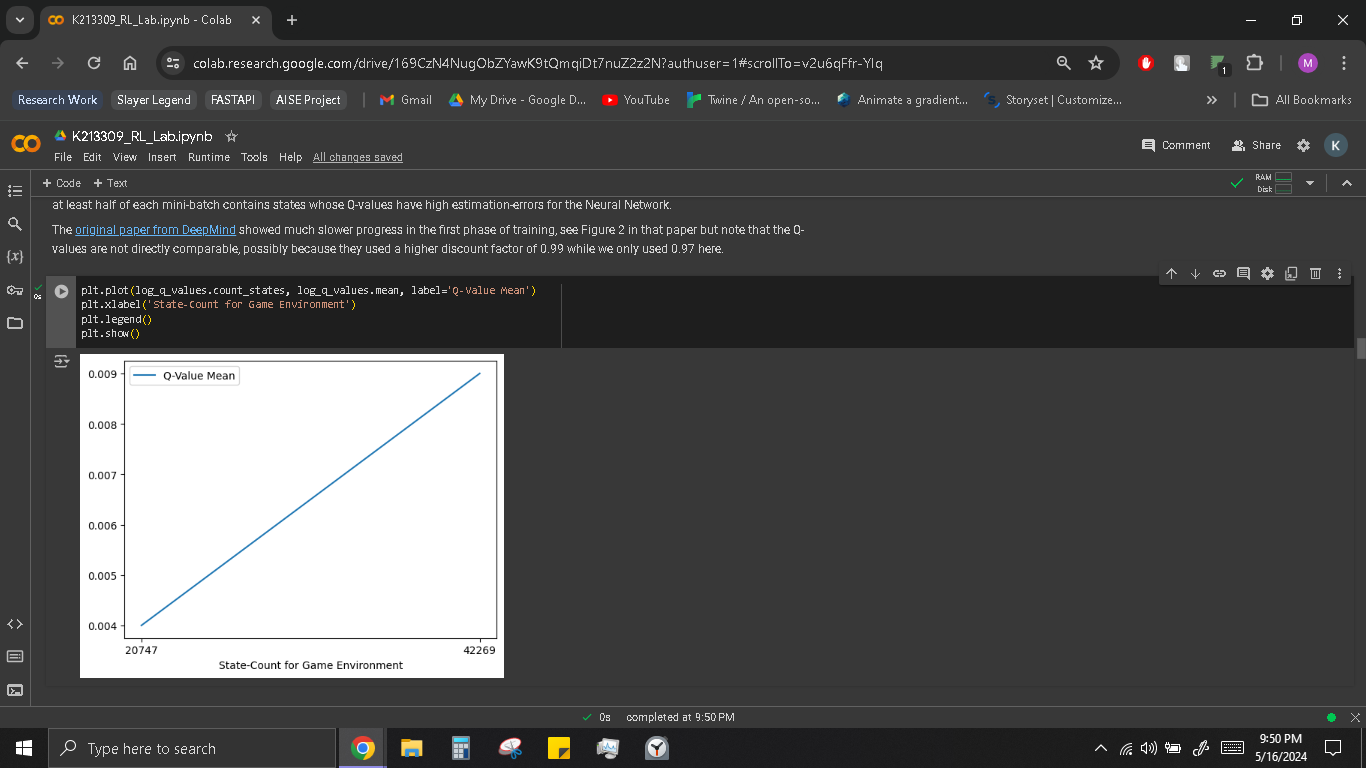
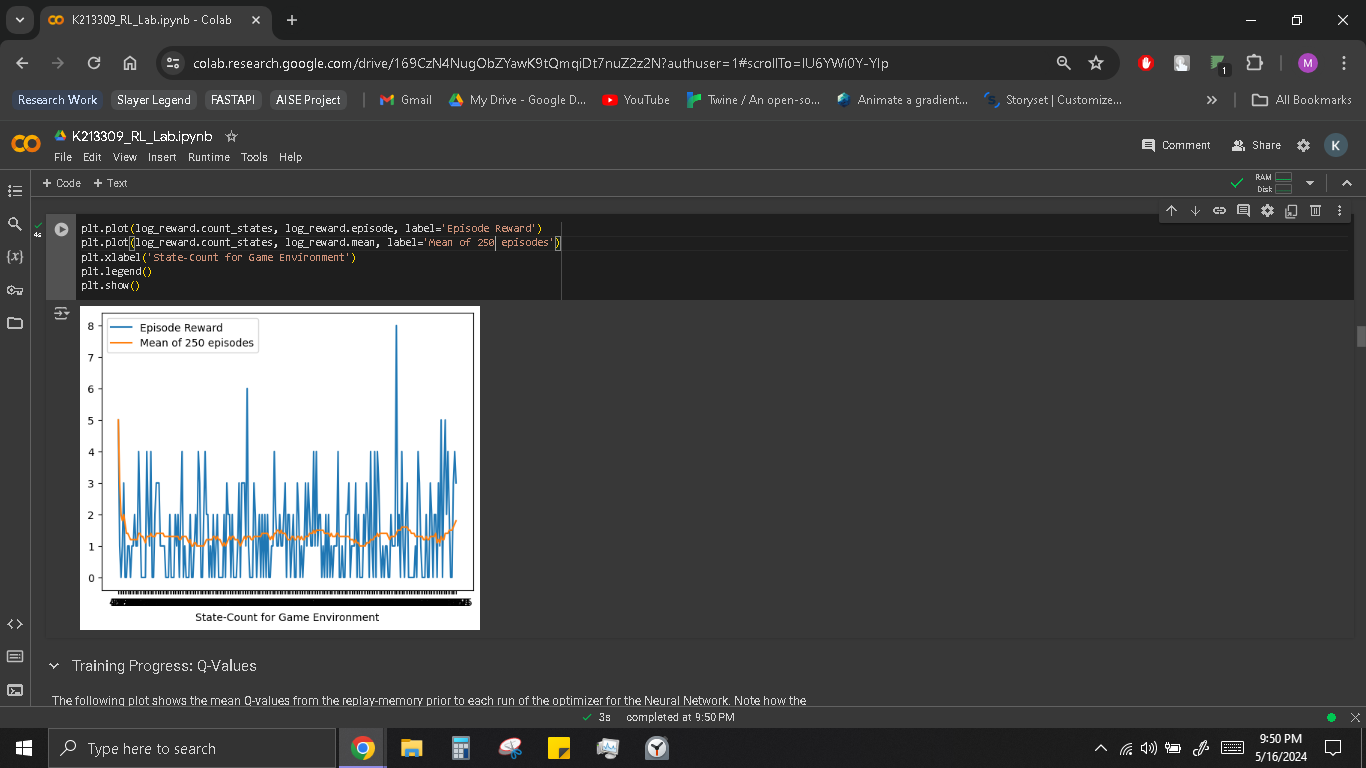
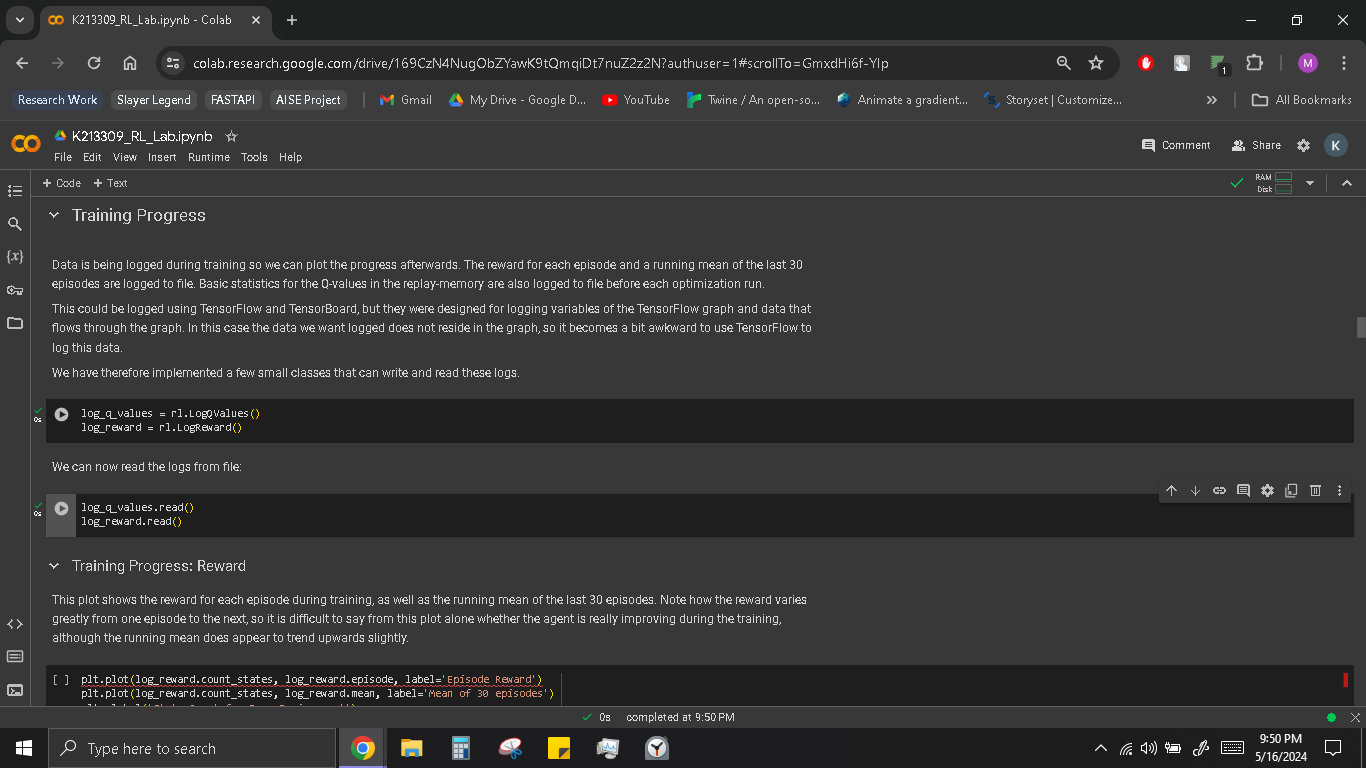
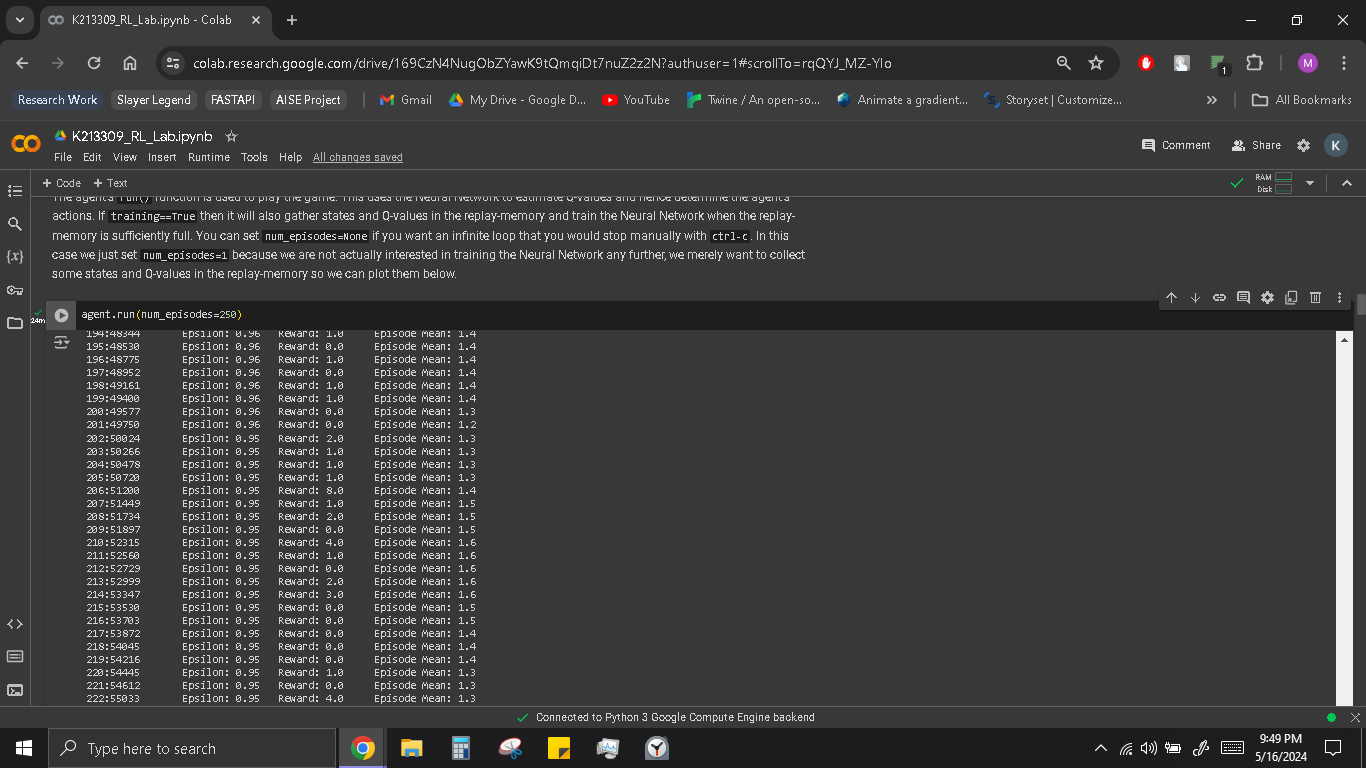
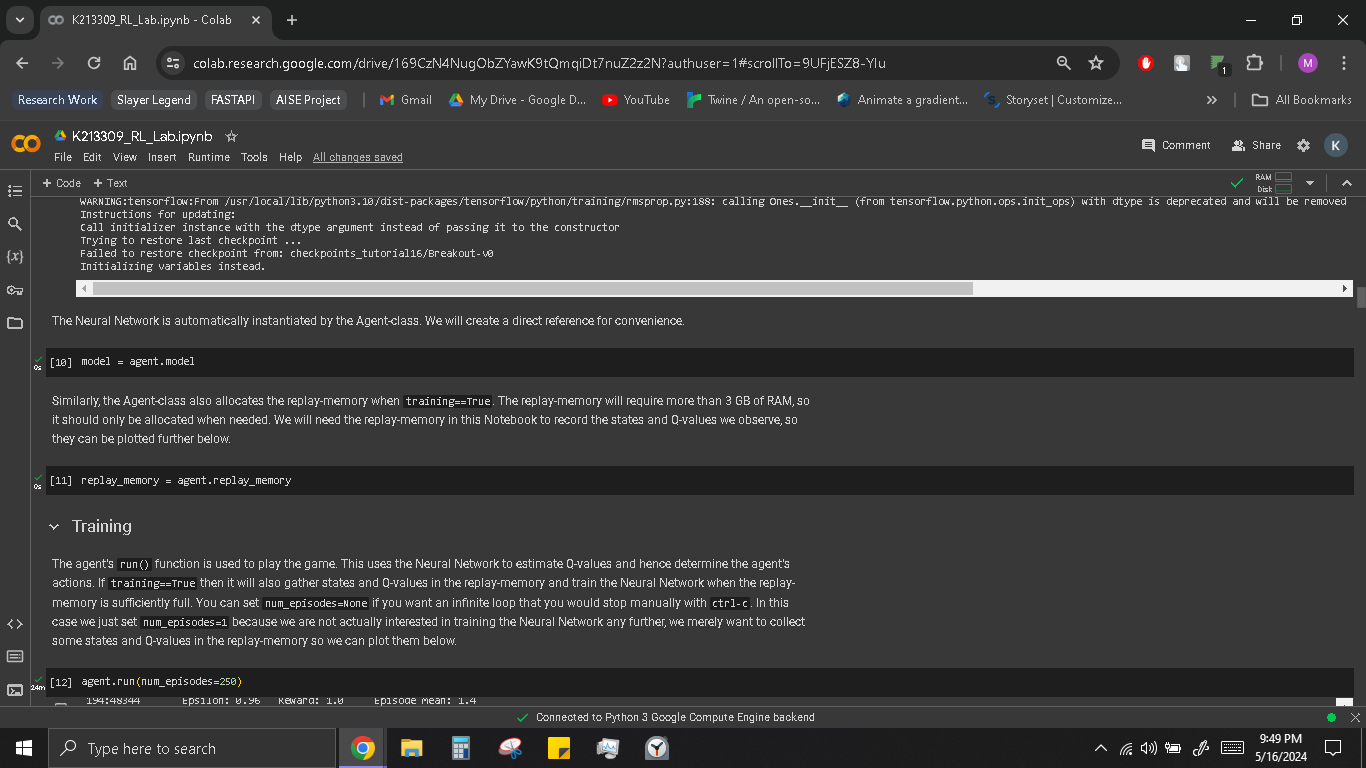
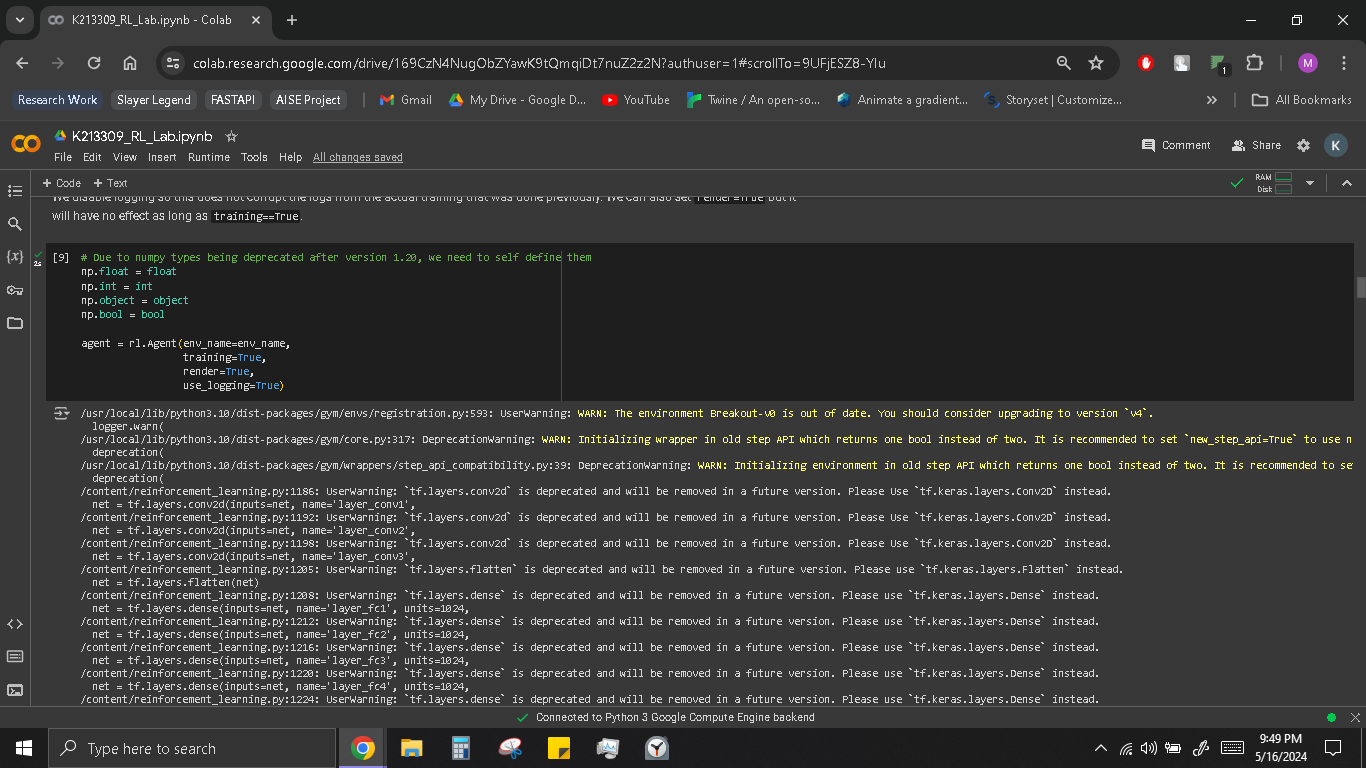
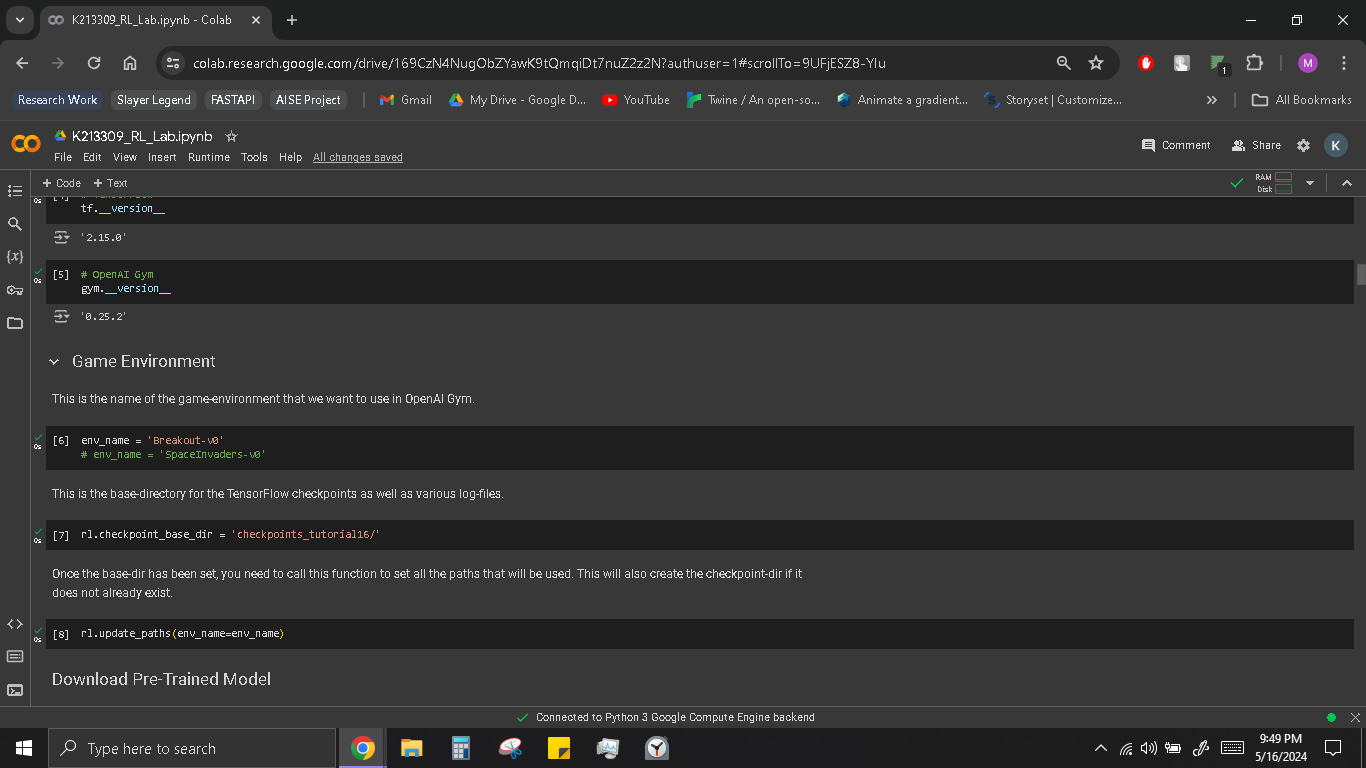
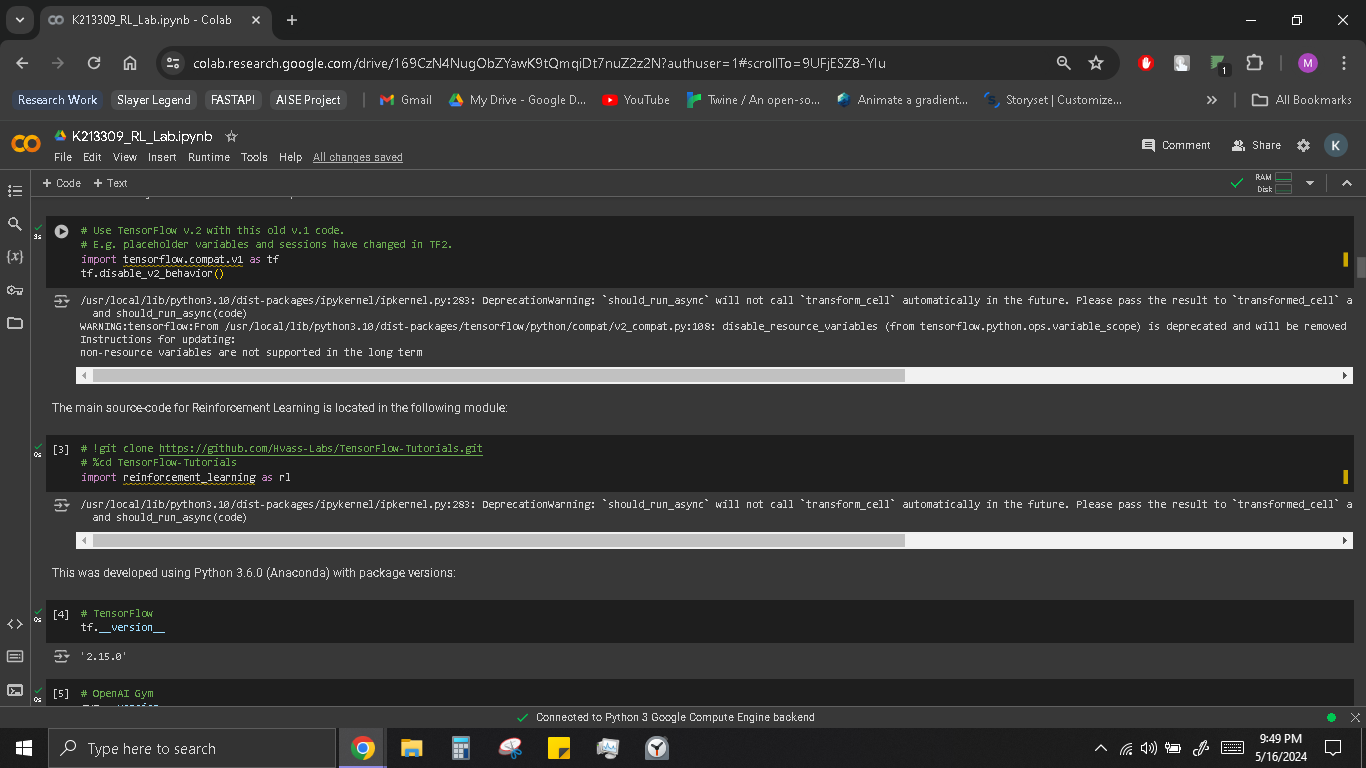
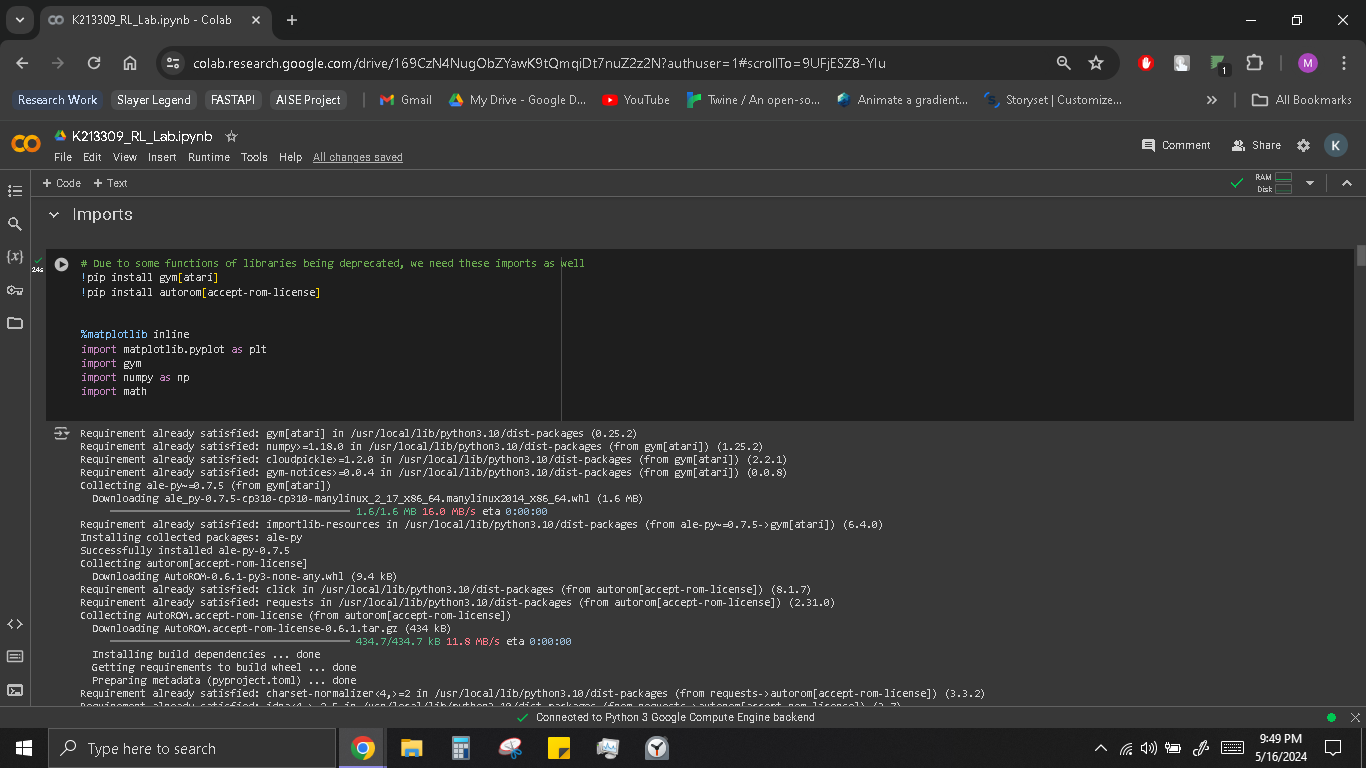
# Question#1

Take snapshots every time you run the code on colab and paste the snapshots on the file.

Due to the notebook being old (7 years to be exact), I had to change some lines of code to properly execute the code. Also remember to upload the reinforcement\_learning.py file OR git clone the repository and cd into it.



# Question#2

How many iterations did you make for your model to estimate Q-values?

250

# Question#3

How would you ensure that features are extracted from images correctly?

Some ways of ensuring correct feature extraction are: Normalizing dataset, using pre-trained dataset, or use different regularization techniques (like dropout).

# Question#4

What is the role of activation function here?

There are 2 activation functions used in the notebook. The first one being ReLU. It is most probably used to handle the vanishing gradient problem and also allows the network to converge faster. The second one being Softmax. It is used to convert raw output into probability.