INTRODUCTION TO DATA SCIENCE FINAL PROJECT

GROUP NUMBER：GROUP 18

GROUP MEMBER：余嘉俊(JACK)111522041、翁庭凱(KYLE)111522133

POSSIBLE FINAL PROJECT TOPICS ：

"TRAINING TO BECOME A REAL SONIC THE HEDGEHOG IN A CLASSIC SONIC GAME"

Presentation Outline

- **Simple introduction of our project**

- **Previously on our mid-term presentation & What problem we are facing during the mid-term**

- **The adjustment of the model**

- **After the adjustment**

- **Experiment**

- **Conclusion**

- **Review**

"Training to become a real Sonic The
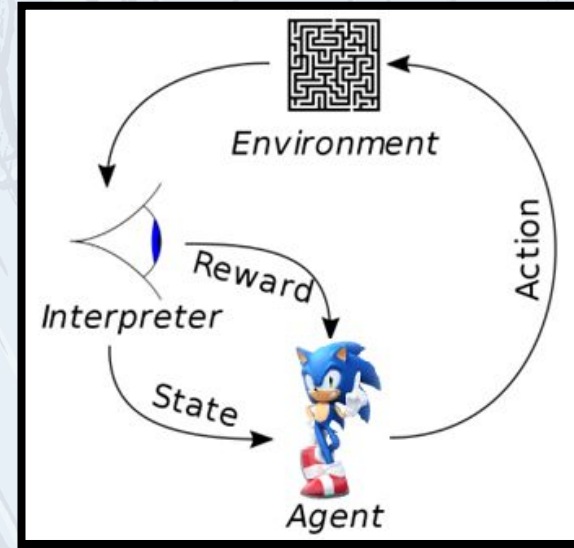
Hedgehog in a classic Sonic game"

Sonic The Hedgehog game:

The typical framing of reinforcement learning in this project:

Simple introduction of our project:

Agent: Sonic (or other characters to be our agent if we can)

Environment:
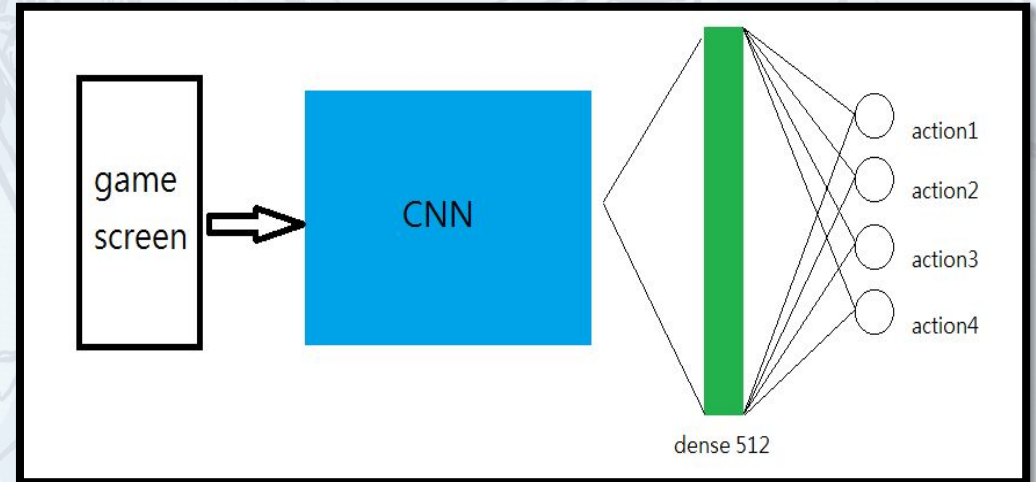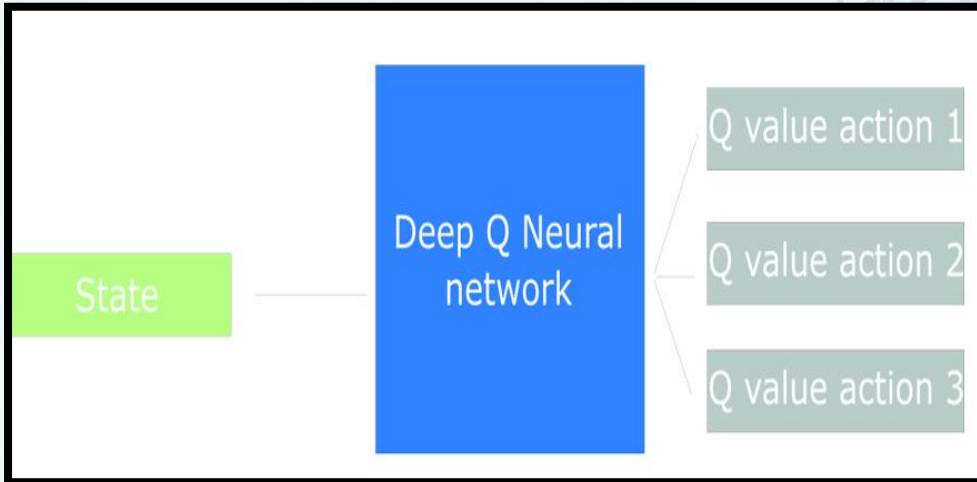

Model: D3QN, A2C

Rewards: written in scenario.json

Previously on our mid-term presentation:

We try to use DQN to build our first trainable agent :

For our Q-function, we want to input current game screen images as "state" and output each action value. We will choose the action which has the highest value as current state output , and this is how the network looks like:

And we use TD(Temporal Difference) to learn our Q-function.

What problem we are facing during the mid-term:

Our agent learn that it has to jump most of the time, and it is not the result we expected. Here is one of our guess:

The default reward of environment is not really the reward we have been seeking for. In this case, our agent will get the reward only if it kills an enemy (there is an enemy, flying robot, near the starting point), which is not necessary to complete the level.

The adjustment of the model

- DQN ➡ Dueling DQN ➡ D3QN

- Epsilon-greedy strategy ➡ Noisy network

- Reward function without penalty ➡ Reward function with penalty if sticking
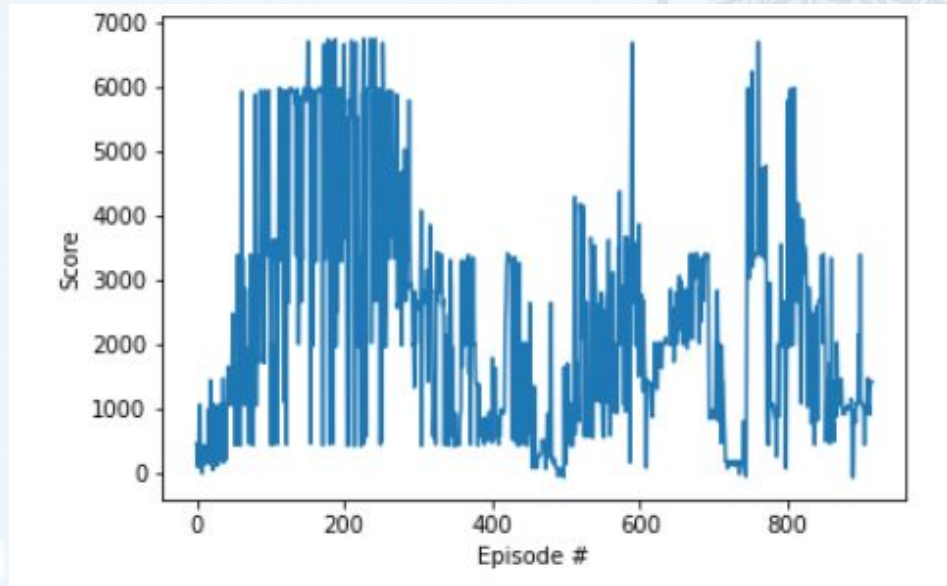
- Simple Replay buffer ➡ Prioritized Replay buffer

After the adjustment:

Our agent will not always try to kill an enemy.
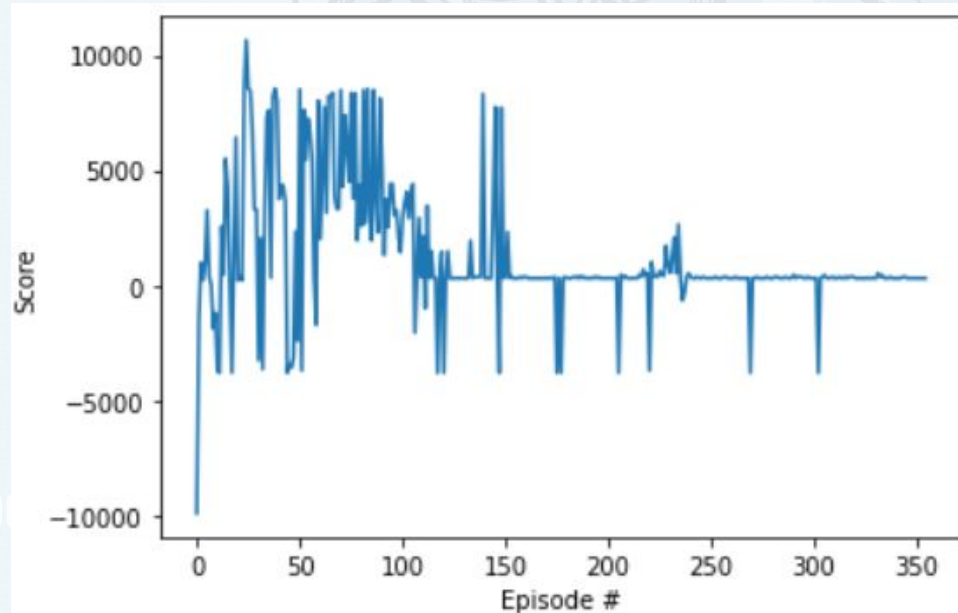
Experiment

We have been testing the model before the eventual update, but the results are not very good, as shown in the following figure. The result below is the return of a modified reward, no reward for passing to the next level and only penalty in no operation, using a D3QN model.
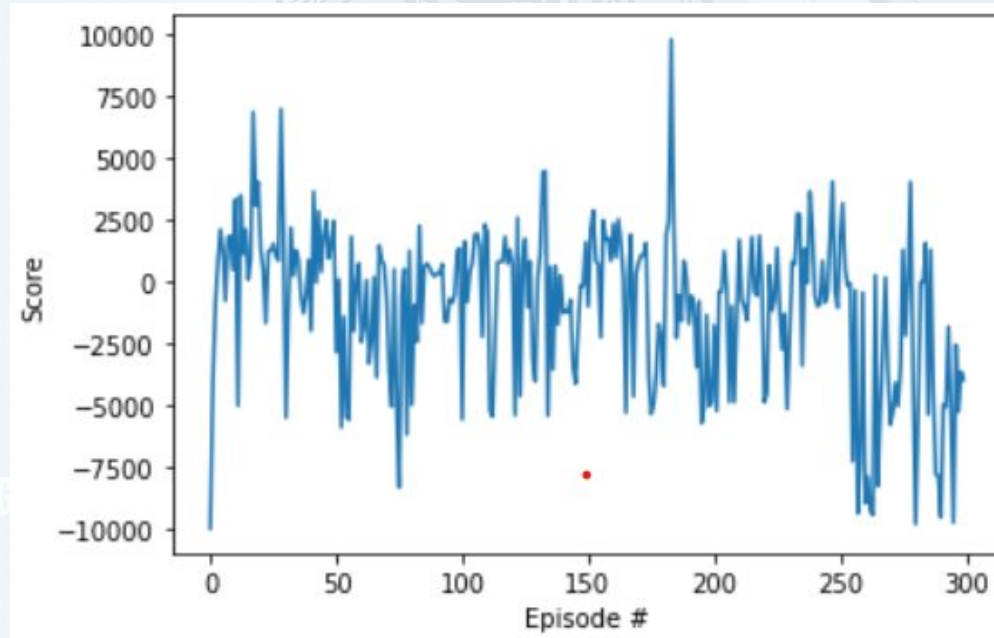
Conclusion

A couple of times, the agent passed the next level. There were two instances when the agent reached 7000 points, once when it was a perfect pass. However, the error occurs almost whenever the agent can pass to the next level. The later part of the agent will start to forget what it learned before.

Review

- **Sonic The Hedgehog** , which does not like Mario, is a non-linear game. Although the goal of the stage is always on the far right, the agent often has to turn back to run with the action, *'DOWN, B'*, i.e., acceleration, and touch the tool(e.g., a spring) to pass the dead-end. If we have more time, we will study reward shaping to do further research on this game.

THANK YOU
FOR YOUR LISTENING

余嘉俊(Jack)　　　　翁庭凱(Kyle)