

## Assignment 3 Report

### Q1. Discuss how you speed up your grid world environment. (10 pts.)

1. First, I prevented any `self.reset()` calls in `self.step()`
2. I stored all the valid initial position in `self.valid_states` as an array rather than calling all the "is\_each\_state" to determine whether it is valid in `self.reset()`
3. I also prevented any redundancy using `if` and `elif` rather than all `ifs`. Also, I return as soon as the reward, observations, ... are determined.
4. I browsed for information and imported the `lru_cache` from `functools` package to hasten the speed of those functions that are called frequently and have same results.

### Q2. What's your best result in 2048? (5 pts.)

1. This is not the best average performance but it reaches 512

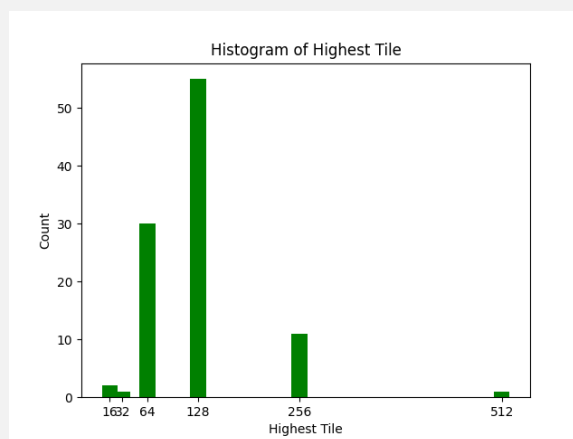


Figure 1: 512 histogram

2. This is the best performance

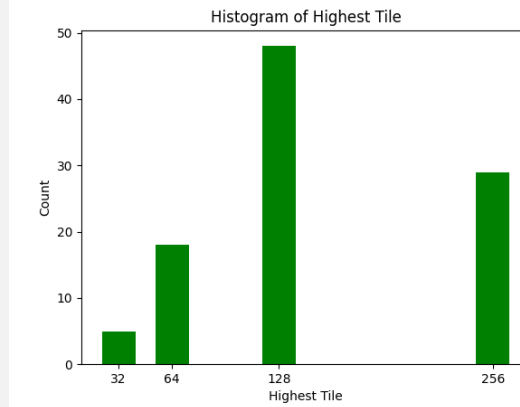


Figure 2: Best histogram

**Q3. Describe what you have done to train your best model. (15 pts.)**

I've tried many methods to try to achieve better performance

1. I implemented a linear learning rate scheduler that linearly decays the learning rate as time step goes.
2. I adjusted the total training epochs and the timesteps\_per\_epoch to make the agent explore longer and learn longer
3. I've tried several patterns and implemented it in the weight matrix

**Q4. Choose an environment from the Gymnasium library and train an agent. Show your results or share anything you like. (10 pts.)**

1. I chose lunar lander as the environment to train
2. This is the learning curve which shows the final reward as the time step goes

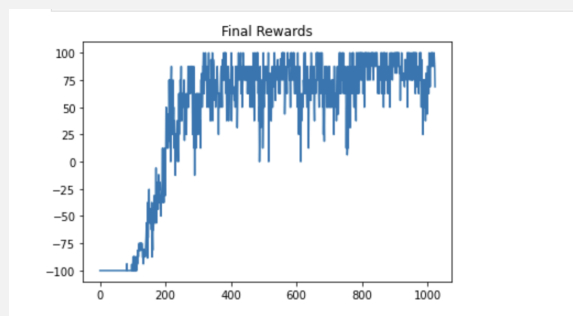


Figure 3: learning curve

3. Below is the visualization of the lunar lander environment

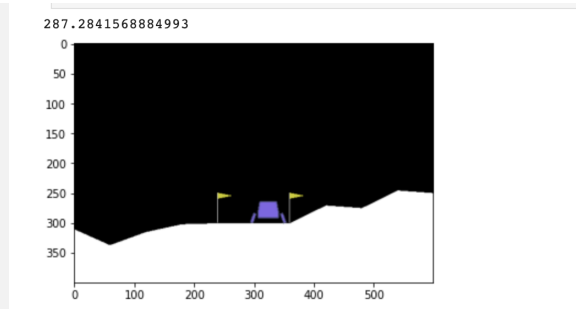


Figure 4: Lunar Lander Environment

4. I, personally think the most difficult part is to set up reasonable reward, e.g. cumulative decaying reward rather than immediate reward.
5. The interesting part is that the landing process often seems to be wobbly.