



[< Back to Deep Reinforcement Learning Nanodegree](#)

Collaboration and Competition

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear Student,

Going through your code was a pleasure as it is really well written in a good and modular style. You also seem to have a good hold on python and Deep Reinforcement learning.

You have handled all the edge cases elegantly, I am quite impressed!

You might also be interested in [this](#) for further experimentation.

[This](#) and [this](#) helped me personally to gain deep insights in Deep Reinforcement learning.

Congratulations on successfully completing the project.
Cheers!

Training Code

The repository includes functional, well-documented, and organized code for training the agent.

Good Job! You might me further interested in reading [style guide](#) for python. Also, you should never miss writing [docstrings](#) for your functions. It makes your code much more readable and understandable.

The code is written in PyTorch and Python 3.

The code is written in pyTorch and Python 3. Well Done.

You might be interested in [this](#) and [this](#) to understand differences between pytorch and tensorflow.

The submission includes the saved model weights of the successful agent.

README

The GitHub submission includes a `README.md` file in the root of the repository.

The README describes the the project environment details (i.e., the state and action spaces, and when the environment is considered solved).

Good Job! The state and action spaces has been described well in the Readme file. In addition, You have also provided a youtube link of the trained agent. The agent has been trained really well and I appreciate your extra effort here. Well Done!

The README has instructions for installing dependencies or downloading needed files.

Good job here! However, you should also include instructions for installing all the other dependencies of your project. Infact, you should try exporting your whole project as a packager. You may find the setup.py script to be very helpful for managing your dependencies. Have a look at [this](#) to know more.

The README describes how to run the code in the repository, to train the agent. For additional resources on creating READMEs or using Markdown, see [here](#) and [here](#).

Report

The submission includes a file in the root of the GitHub repository (one of `Report.md`, `Report.ipynb`, or `Report.pdf`) that provides a description of the implementation.

The report clearly describes the learning algorithm, along with the chosen hyperparameters. It also describes the model architectures for any neural networks.

Well Done! The learning algorithm, the hyper-parameters and the model architecture are clearly explained in the report.

A plot of rewards per episode is included to illustrate that the agents get an average score of +0.5 (over 100 consecutive episodes, after taking the maximum over both agents).

The submission reports the number of episodes needed to solve the environment.

Good Job here, the agent was able to solve the environment in a decent number of episodes!

The submission has concrete future ideas for improving the agent's performance.

Good Job here! You have provided very good future ideas on how to improve the agent's performance. It would be really interesting to see how the agent performs on the suggested implementations. As a additional challenge, you should try them out and see which one performs better and maybe report them in your repo!

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

[Rate this review](#)