PROJECT SPECIFICATION

Navigation

Training Code

CRITERIA	MEETS SPECIFICATIONS
Training Code	The repository (or zip file) includes functional, well-documented, and organized code for training the agent.
Framework	The code is written in PyTorch and Python 3.
Saved Model Weights	The submission includes the saved model weights of the successful agent.

README

CRITERIA	MEETS SPECIFICATIONS
`README.md`	The GitHub (or zip file) submission includes a `README.md` file in the root of the repository.
Project Details	The README describes the the project environment details (i.e., the state and action spaces, and when the environment is considered solved).
Getting Started	The README has instructions for installing dependencies or downloading needed files.
Instructions	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

CRITERIA	MEETS SPECIFICATIONS
Report	The submission includes a file in the root of the GitHub repository or zip file (one of `Report.md`, `Report.ipynb`, or `Report.pdf`) that provides a description of the implementation.
Learning Algorithm	The report clearly describes the learning algorithm, along with the chosen hyperparameters. It also describes the model architectures for any neural networks.
Plot of Rewards	A plot of rewards per episode is included to illustrate that the agent is able to receive an average reward (over 100 episodes) of at least +13. The submission reports the number of episodes needed to solve the environment.
Ideas for Future Work	The submission has concrete future ideas for improving the agent's performance.

Suggestions to Make Your Project Stand Out!

- Include a GIF and/or link to a YouTube video of your trained agent!
- Solve the environment in fewer than 1800 episodes!
- Write a blog post explaining the project and your implementation!
- Implement a double DQN, a dueling DQN, and/or prioritized experience replay!
- For an extra challenge ***after passing this project***, try to train an agent from raw pixels! Check out `(Optional) Challenge: Learning from Pixels` in the classroom for more details.