

[My Programs](#) ▸ ▸ [Navigation](#) ▸ [Submit Project](#)

Project: Navigation

Submission Results

Submission Date: February 19, 2020

 Submission Passed

[Download Submission](#)




Feedback Details

[Specification Review](#) Code Review

Reviewer Note

Great job getting familiar with the Deep Q Network and Dueling Deep Q Network to successfully complete the project. The next project is even more exciting and awaits you. All the best! 🌟 The recent achievement of the [Open AI group to play Dota 2](#) using Reinforcement Learning is a must read. 🍌

Training Code

-  Saved Model Weights
-  Framework
-  Training Code

Reviewer Note

Awesome work implementing a reinforcement learning agent to collect the "yellow bananas in a large square world".





- Very good decision to implement the DQN, and Duel DQN algorithms, very effective reinforcement learning algorithms.

Pros of the implementation

- Good implementation of Deep Q Network and Dueling Deep Q Network algorithms.
- Impressive work solving the Visual banana challenge also.
- Decoupling of the parameters being updated from the ones that are using a target network to produce target values has been done correctly.
- The Epsilon-greedy action selection to encourage exploratory behavior in the agent been perfectly implemented.
- Soft update helps to prevent variance into the process due to individual matches. Good use of tau parameter to perform soft-update.
- Good use of the replay memory to store and recall experience tuples.

The repository (or zip file) includes functional, well-documented, and organized code for training the agent

README

-  Getting Started
-  Instructions
-  [README.md](#)
-  Project Details

Reviewer Note

Awesome work providing the project environment details including the state and action spaces, the reward function and when the agent is considered solved. The description is very informative.

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

Report

-  Plot of Rewards

Reviewer Note

Performance of the agent is awesome.

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	▼
✓	Report	▼
✓	Learning Algorithm	▼