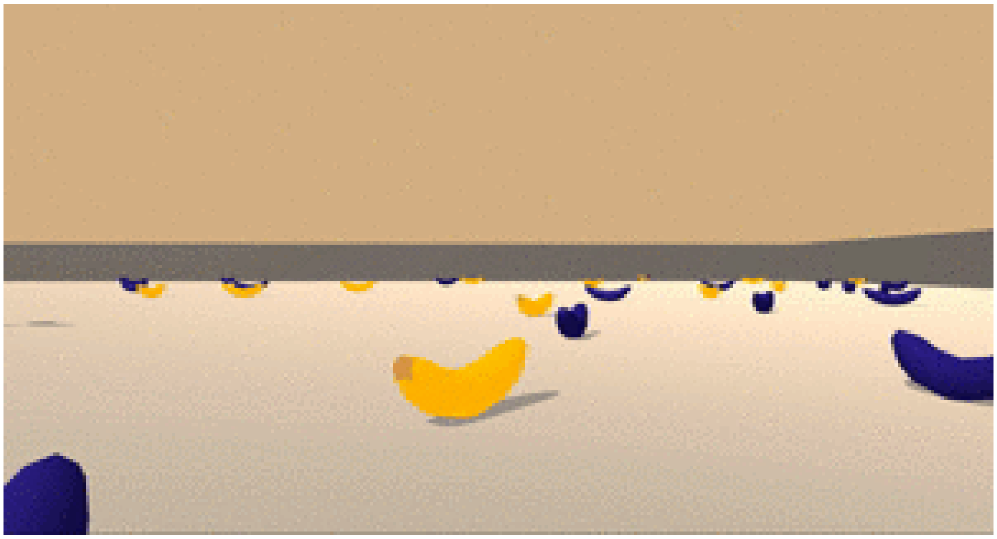
Project 1: Navigation Report



I: Project Summary

For this project, a Deep Q-Learning algorithm is implemented to train an agent to navigate a large, square world to collect yellow bananas, and avoid blue bananas.

II: Learning Algorithm

As mentioned in the summary, we'll use Deep Q-Learning (DQN) to teach an agent how to interact with the environment to learn how to achieve its goal of collecting yellow bananas and avoiding blue bananas.
The main idea and details behind DQN can be found in this paper here, but in general, in DQN, we use a neural network to replace the Q-table to estimate the optimal action-value function.

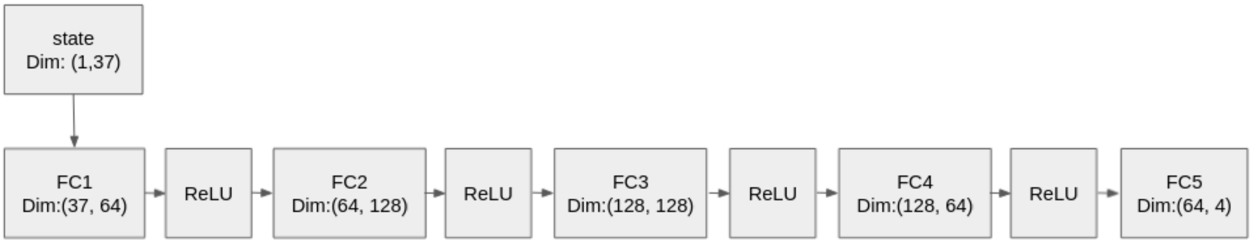
In addition to the vanilla DQN implementation, the experience relay technique is used to learn from individual experience tuples multiple times, recall rare occurrences, and make better use of our experience. For experience relay, a replay buffer of size 10,000 is used.

Moreover, in order to avoid chasing a moving target by having the parameters be a function of themselves causing unpredictable and unstable changes, the Fixed Q-Targets technique is used where we create a copy of the neural network parameters and use it to generate the targets while changing the original targets for a certain number of learning steps.

The table below shows the hyperparameters used to train the model:

BUFFER_SIZE (replay buffer size)	10,000
BATCH_SIZE (minibatch size)	64
GAMMA (discount factor)	0.99
TAU (for soft update of target parameters)	1e-3
LR (learning rate)	5e-4
UPDATE_EVERY (how often to update the network)	4

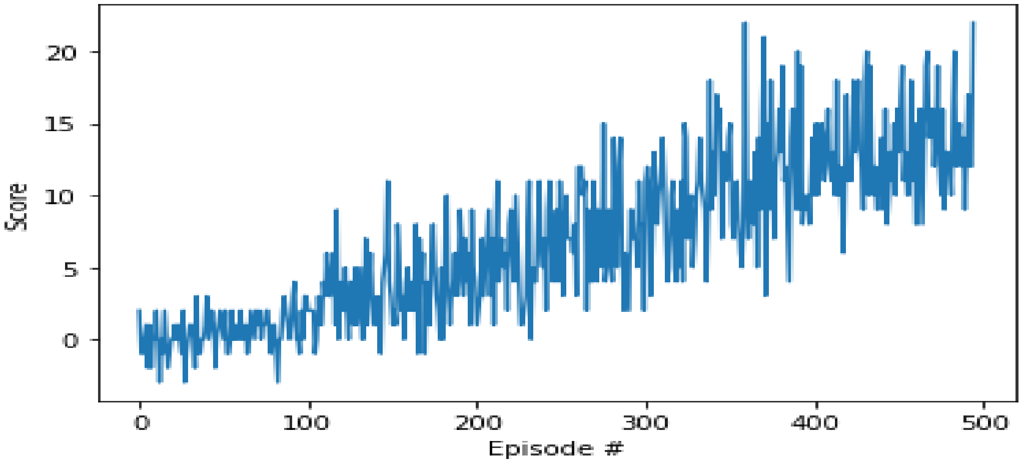
The figure below shows the deep neural network architecture used to train the agent, where:
FC: Fully Connected Layer with its dimensions
ReLU: Rectified-Linear Unit



III: Rewards Plot

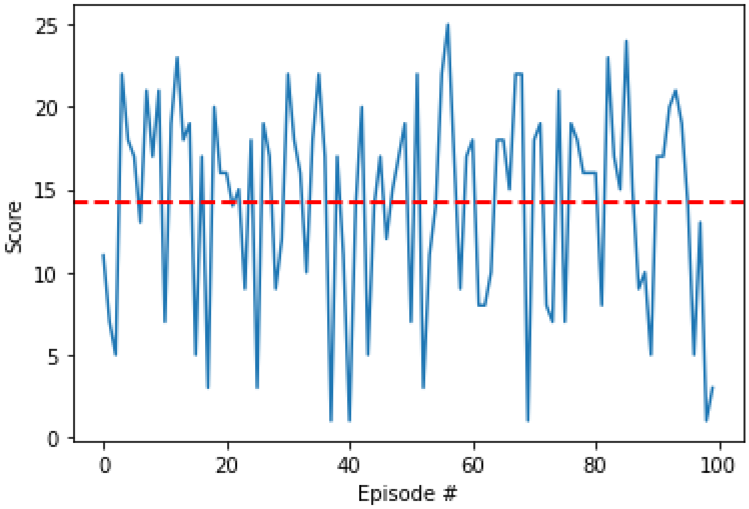
A: Training Plot

The plot below shows the rewards per episode that the agent received an average reward over training for 395 episodes to solve the environment.



B: Evaluation Plot

The plot below shows rewards per episode (over 100 episodes), it shows the agent has an average reward of at least 13 over the 100 episodes (the average reward over 100 episdoes is 14.25 to be exact)



IV: Future Work

Although the agent is able to navigate the environment to collect yellow bananas and avoid blue bananas, in some cases the agent gets stuck between a few actions and stays in place. One possible solution for this is to select an epsilon-greedy action when the agent is oscillating between actions over a specified period of time.
It would also be interesting to implement DQN improvements such as Double DQN, Prioritized Experience Replay, and Dueling DQN to see how they affect the agent's performance.