REPORT OF IMPLEMENTATION

In this project was used DQN method. DQN means Deep Q-Networks and extend value-based reinforcement learning methods to complex problems using deep neural networks.
The union of value-based reinforcement learning methods and deep neural networks helps the agent's to learn faster and better.
This method was chosen because DQN method perform well in discrete actions spaces.
Also, was used Experience Replay and Fixed Q-Targets methods to improve the agent's performance.

The implementation begins in step 4. It's Your Turn.

1- Was created the QNetwork class that build the deep neural network with three fully connected layers, receiving the state as input and actions as output.
Model architecture of the neural network:

    37(Input-state_size) x 64(hidden_layer) x 64(hidden_layer) x 4(output - action_size)

The neural network was created using PyTorch.

2- Was created the Agent class.
Looking for learning and improving the results, the agent implements the Experience Replay and Fixed Q-Targets methods.

3- Was created the ReplayBuffer class to implement the Experience Replay method

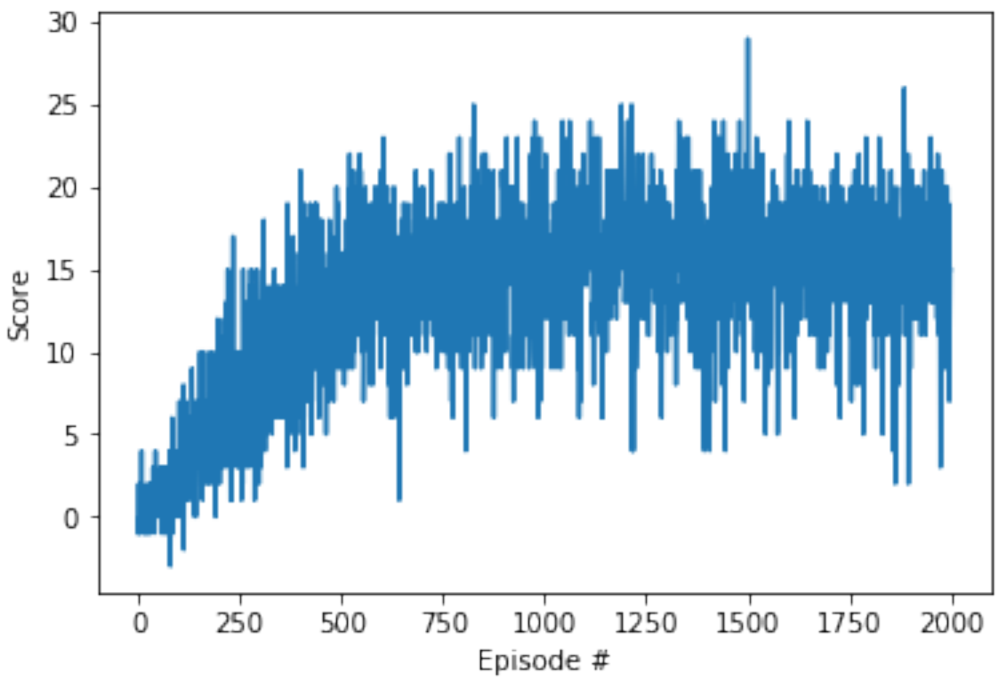
4- Training the agent
The agent was trained in 2.000 episodes and epsilon decay = 0.995
The environment is considered solved when the agent achieve the mean score >= 13.0, but the train continues until finish the 2.000 episodes.

Hyperparameters used to train the agent:

```
BUFFER_SIZE = int(1e5)  # replay buffer size
BATCH_SIZE = 64         # minibatch size
GAMMA = 0.99            # discount factor
TAU = 1e-3              # for soft update of target parameters
LR = 5e-4               # learning rate
UPDATE_EVERY = 4        # how often to update the network
N_EPISODES = 2.000    # number of episodes
EPS_START = 1.0  # epsilon start
EPS_end = 0.01  # epsilon end
EPS_START = 0.995  # epsilon decay
```

5- RESULTS:

- The agente achieve score 13.01 in episode 535
- The biggest average score was 16.26 in episode 1.500



- Running the agent after the training it achieve score = 16.0

6- Future ideas for improving agent's performance
Looking for improving the agent's performance we suggest that be implemented the method Prioritized Experience Replay.