MDDPG (Learning Algorithm):

MDDPG is a variation of DDPG or Deep Deterministic Policy Gradient for multi-agent enviroments, MDDPG is the algorithm used in this project to solve the enviroment. DDPG is a different kind of actor-critic method, and it could be seen as an approximate DQN instead of an actual actor-critic, because the critic in DDPG is used to approximate the maximizer over the Q values of the next state and not as a baseline.
One of the limitations of the DQN agent is that it is not straightforward to use in continuous action spaces

In DDPG we use two deep neural networks, we call one the actor and the other the critic:

- Actor: takes the state S as input, and outputs the optimal policy deterministically, the output is the best believed action for any given state unlike the discrete case where the output is a probability distribution over all possible actions, the actor is basically learning the argmax Q(s,a)
- Critic: takes the input state S, then it tries to calculate the optimal action-value function by using the actor's best-believed action

Two interesting aspects of DDPG are:

- The use of a replay buffer
- Soft updates to the target networks:
    - In DQN you have two copies of the network weights: the regular network and the target network, where the target network gets a big update every n-step by simply copying the weights of the regular network into the target network
    - IN DDPG, you have two copies of the network weights for each network, a regular for the critic an irregular for the critic, a target for the actor, and a target for the critic, but in DDPG the target networks are updated using a soft updates strategy which is slowly blending the regular network weights with the target network weights, so every time step you mix in 0.01% of regular network weights with target network weights

DDPG Chosen Hyperparameters:

- Replay buffer size: 100000
- Minibatch size: 128
- Discount factor: 0.99
- TAU for soft update of target parameters: 0.001
- Learning rate of the actor: 0.001
- Learning rate of the critic: 0.001

Neural Networks Architecture:

Actor Network consists of:

- Fully connected layer which takes the state with the size 24, and outputs 256
- Fully connected layer which takes the ReLU of the output of the previous layer and outputs 2 logits, each representing an action
- The output from the previous layer is passed through a tanh to ensure the output actions are between -1 and 1

Critc Network consists of:

- Fully connected layer which takes the state with the size 24, and outputs 256
- Fully connected layer which takes the Leaky ReLU of the output of the previous layer along the actions chosen by the actor with size 260 and outputs 256
- Fully connected layer which takes the Leaky ReLU of the output of the previous layer and outputs 128
- Fully connected layer which takes the Leaky ReLU of the output of the previous layer and outputs 1 logit which resembles the State-Value