I. Agent Structure (MADDGP - Multi-Agent Deep Deterministic Policy Gradient)

agents - list of all the DDPG agents (the number of agents is equal to the number of players in the game)

Details of DDPG is in the project 2 report

Parameters

num_agents	2	Number agents (players)
n_episodes	1000	Number of times the environment is run
BUFFER_SIZE	1e6	Replay (memory) buffer size
BATCH_SIZE	128	Minibatch size
GAMMA	0.99	Discount factor
TAU	1e-3	Soft update of target parameters
LR_ACTOR	1e-4	Learning rate of the actor
LR_CRITIC	1e-4	Learning rate of the critic
WEIGHT_DECAY	0	L2 weight decay
mu	0	Noise parameter
theta	0.05	Noise parameter
sigma	0.4	Noise parameter

Overview

For each episode:

1) Reset the environment
2) Get the state of the environment

While the environment is running:

1) Use the agent to **predict action** based on the environment's state
2) Pass the predict action into the environment
3) Retrieve the next state, reward, and done variable
4) **Agent step**. Record next state, reward, and done variable to **memory** in agent
5) Set the current state variable to next state
6) Add reward to score
7) If the current score is the highest, save the model
8) Break out of while loop if done is true

Predict action

1) Loop through the states that were passed in

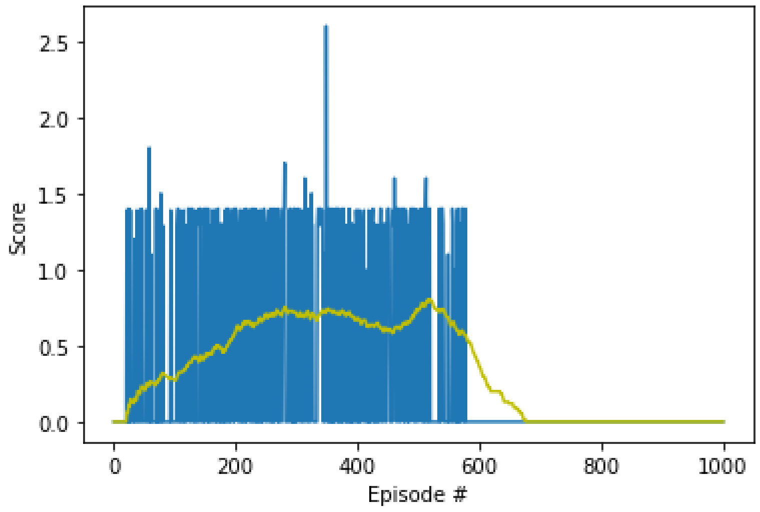
2) For each state, pass it to the corresponding agent
3) Collect the action and return the actions as an array

Agent step

1) Loop through the states that were passed in
2) For each state, actions, rewards, next_states, dones, pass it into the corresponding agent

II. Results

Here's a graph of the episodes vs scores. Most scores were between 1 and 1.5 except for a 2.5. The yellow line is the average score. The highest average score is around 0.75



III. Ideas for Future work

Other improvements to the algorithm I would like to pursue:

1) Prioritized Experience Replay
2) Experiment with sharing the replay buffer with all agents