1 Introduction

The goal of the project is to create a DQN agent and define a reward system to teach a robotic arm to carry out two primary objectives:

- Have any part of the robot arm touch the object of interest, with at least a 90% accuracy.

- Have only the gripper base of the robot arm touch the object, with at least a 80% accuracy.

The project will be done using Gazebo simulated environment with a C++ plug-in to interface Gazebo Simulated Robot to the DQN agent. in order to achieve the objective few tasks are required to be completed:

- Subscribe to camera and collision topics published by Gazebo.
- Create the DQN Agent and pass all required parameters to it.
- Define position based control function for arm joints.
- Penalize robot gripper hitting the ground.
- Interim Reward/Penalize based on the arm distance to the object.
- Reward based on collision between the arm and the object.
- Tuning the hyperparameters.
- Reward based on collision between the arm's gripper base and the object.

Next sections will explain in details how the required tasks were achieved.

2 Reward Function

Deep Q-Network (DQN) output is usually mapped to a particular action, which, for this project, is the control of each joint for the simulated robotic arm. Control of the joint movements can be through velocity, position, or a mix of both. In case of this project position control approach was selected.

Reward system was designed to train the robot to have any part of the robot arm touch the object of interest in one attempt then have the gripper base of the robot arm touch the object in a second attempt. designed reward system is as described in below 1. Each episode is limited to a certain number of attempts, penalty will be issued if maximum length of the episode was reached without winning. a penalty will be issued and episode will be ended if robot arm touched ground. interim reward or interim penalty will be issued while robot arm is moving based on the distance from the object of interest. when collision happens with object a win reward is issued and episode is ended.

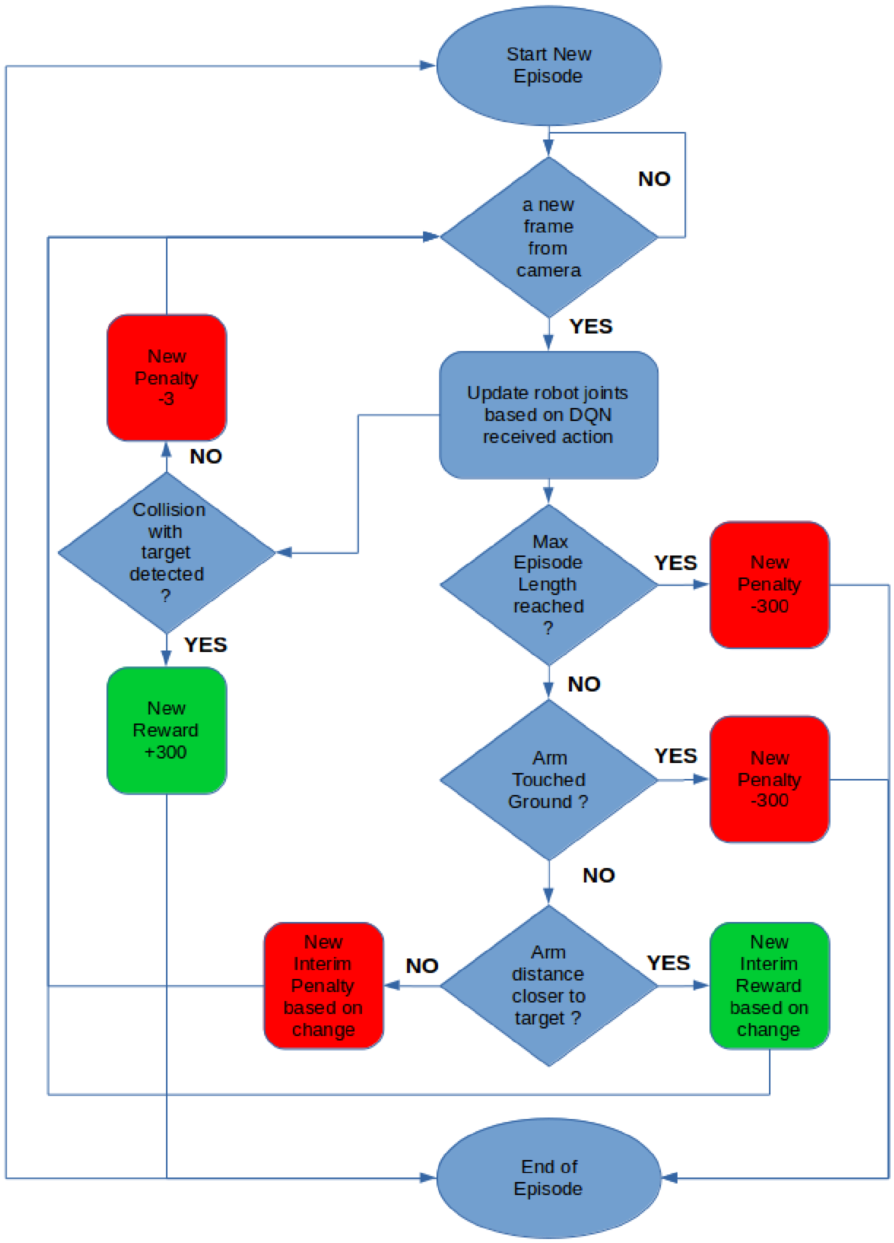

Figure 1: Reward System flowchart

Interim rewards are issued based on a smoothed moving average of the delta of the distance from the robot arm/gripper to the object of interest. It is calculated as following:

Distance Delta = last Distance to Goal - Current Distanced to Goal

Average Goal Delta = (Average Goal Delta * alpha) + (distance Delta * (1.0 - alpha))

Then interim reward is calculated based on the Average Goal Delta.

3 Hyper-parameters

Specify the hyper-parameters that you selected for each objective, and explain the reasoning behind the selection. Student should explain the choice of hyper-parameters for both objectives.

4 Results

Explain the results obtained for both objectives. Include discussion on the DQN agent's performance for both objectives. Include watermarked images, or videos of your results. Student should describe and briefly explain the results they achieved for both objectives. The discussion should also include their comments on the DQN agent's performance and if there were any shortcomings. Student should include either watermarked images of their results, or attach a video that displays the results and the arm in action.

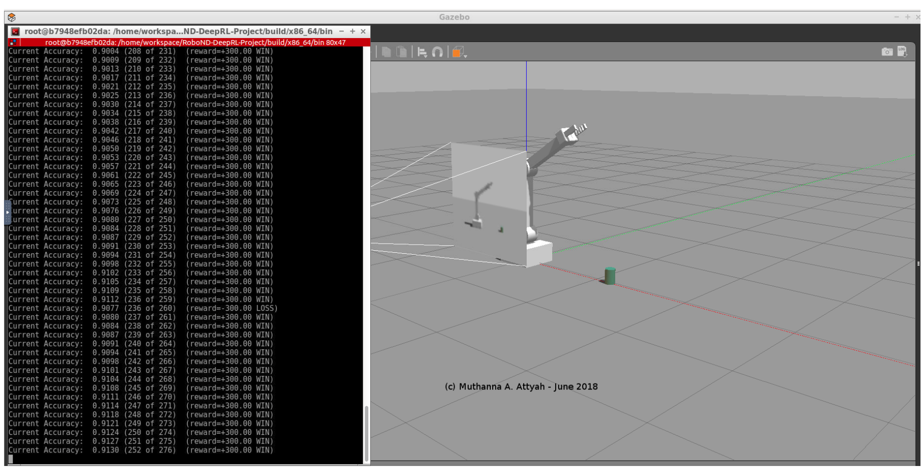

Figure 2: Results of Robot Arm touching the object

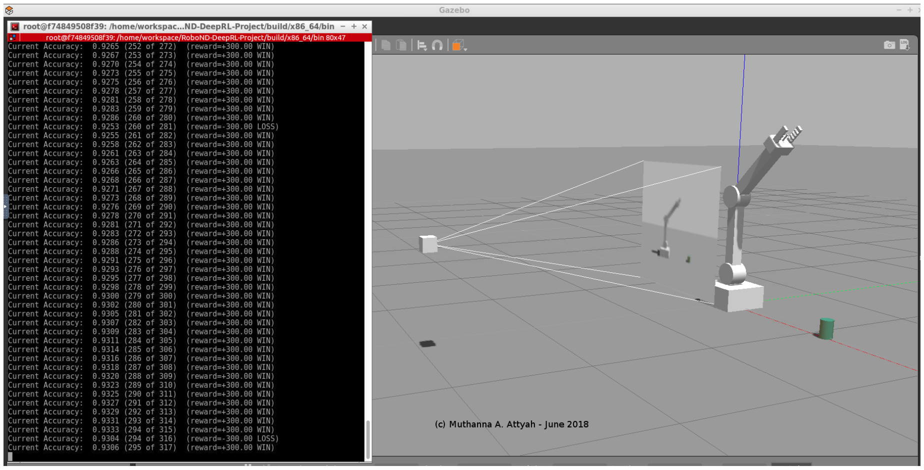

Figure 3: Results of Robot Gripper touching the object

5 Future Work

After fine tuning the hyper-parameters results was quite encouraging, however still there is a room for improvement. One suggest approach is to graph the accuracy progress for every change in parameters then identify the maximum accuracy that can be achieved and how many iterations (learning time) it will take to reach it.

DQN can be also used by itself to fine tune some of the parameters of another DQN where reward can be issued to the first DQN based on obtained accuracy from the second DQN.

Briefly discuss how you can improve your current results. Student should discuss on what approaches they could take to improve their results.