Deep RL Arm Manipulation

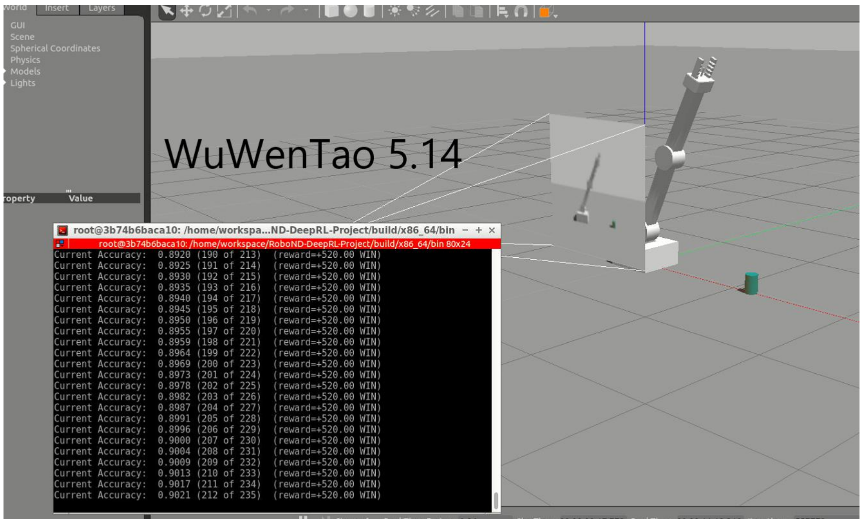
Two primary objectives:

1、Have any part of the robot arm touch the object of interest, with at least a 90% accuracy for a minimum of 100 runs.

2、Have only the gripper base of the robot arm touch the object, with at least a 80% accuracy for a minimum of 100 runs.

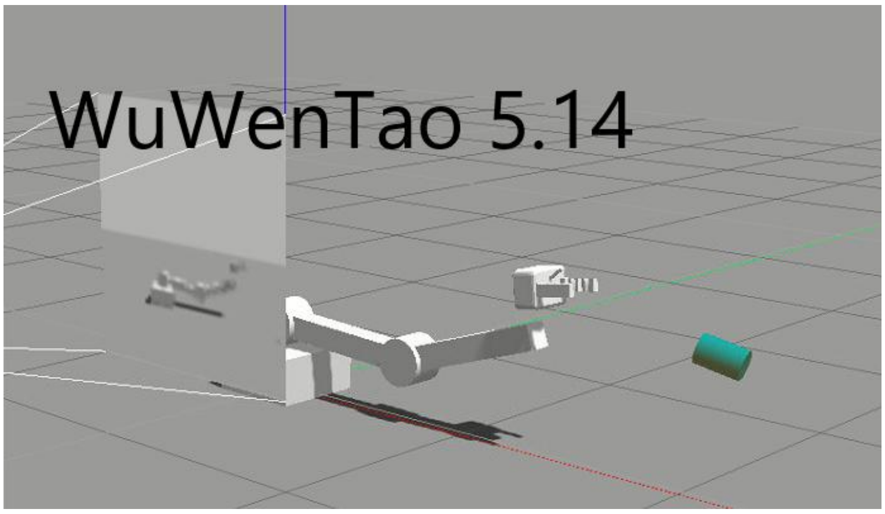
For the first project:

(1) Result and some interesting pic



```
root@3b74b6baca10: /home/workspa...ND-DeepRL-Project/build/x86_64/bin
root@3b74b6baca10: /home/workspace/RoboND-DeepRL-Project/build/x86_64/bin 80x24
Current Accuracy:  0.8920 (190 of 213)  (reward=+520.00 WIN)
Current Accuracy:  0.8925 (191 of 214)  (reward=+520.00 WIN)
Current Accuracy:  0.8930 (192 of 215)  (reward=+520.00 WIN)
Current Accuracy:  0.8935 (193 of 216)  (reward=+520.00 WIN)
Current Accuracy:  0.8940 (194 of 217)  (reward=+520.00 WIN)
Current Accuracy:  0.8945 (195 of 218)  (reward=+520.00 WIN)
Current Accuracy:  0.8950 (196 of 219)  (reward=+520.00 WIN)
Current Accuracy:  0.8955 (197 of 220)  (reward=+520.00 WIN)
Current Accuracy:  0.8959 (198 of 221)  (reward=+520.00 WIN)
Current Accuracy:  0.8964 (199 of 222)  (reward=+520.00 WIN)
Current Accuracy:  0.8969 (200 of 223)  (reward=+520.00 WIN)
Current Accuracy:  0.8973 (201 of 224)  (reward=+520.00 WIN)
Current Accuracy:  0.8978 (202 of 225)  (reward=+520.00 WIN)
Current Accuracy:  0.8982 (203 of 226)  (reward=+520.00 WIN)
Current Accuracy:  0.8987 (204 of 227)  (reward=+520.00 WIN)
Current Accuracy:  0.8991 (205 of 228)  (reward=+520.00 WIN)
Current Accuracy:  0.8996 (206 of 229)  (reward=+520.00 WIN)
Current Accuracy:  0.9000 (207 of 230)  (reward=+520.00 WIN)
Current Accuracy:  0.9004 (208 of 231)  (reward=+520.00 WIN)
Current Accuracy:  0.9009 (209 of 232)  (reward=+520.00 WIN)
Current Accuracy:  0.9013 (210 of 233)  (reward=+520.00 WIN)
Current Accuracy:  0.9017 (211 of 234)  (reward=+520.00 WIN)
Current Accuracy:  0.9021 (212 of 235)  (reward=+520.00 WIN)
```

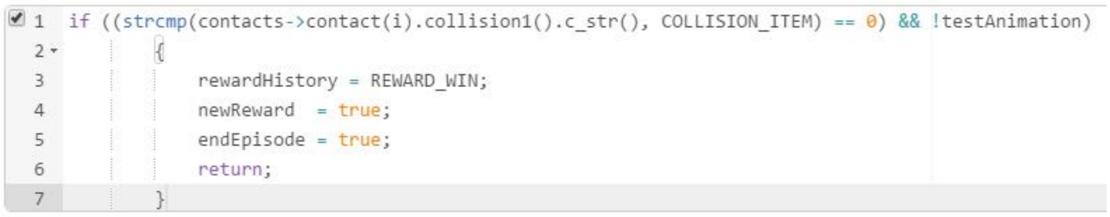
(1)The first project result



(2)The first project interesting pic

(2) Reward Functions

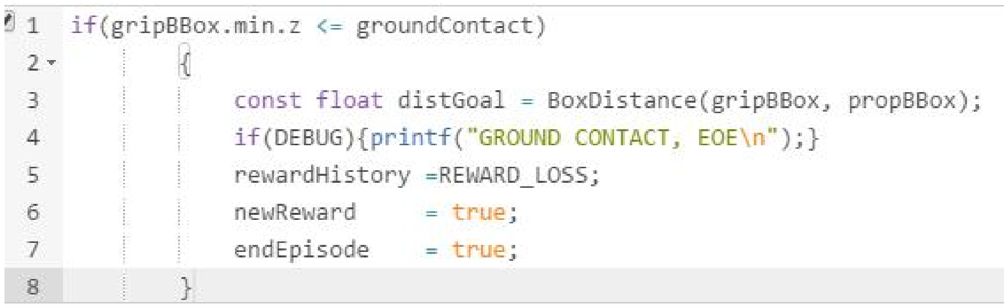
1) When any part of the robot arm touch the object of interest, I set reward function like this:



```
if ((strcmp(contacts->contact(i).collision1().c_str(), COLLISION_ITEM) == 0) && !testAnimation)
        {
            rewardHistory = REWARD_WIN;
            newReward  = true;
            endEpisode = true;
            return;
        }
```

When it touch the object, it is win, so I set rewardHistory=REWARD_WIN and end the episode.

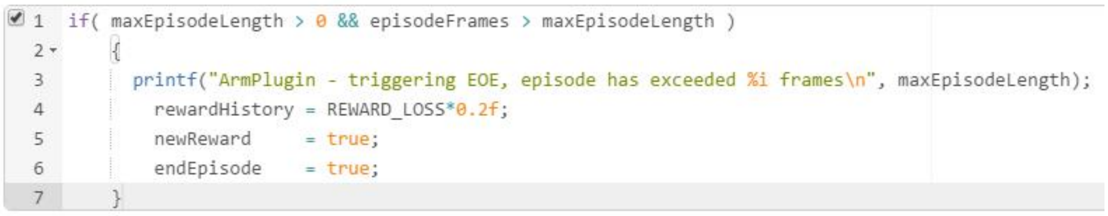
2) When have any part of the robot arm touch the ground, I set reward function like this:



```
if(gripBBox.min.z <= groundContact)
        {
            const float distGoal = BoxDistance(gripBBox, propBBox);
            if(DEBUG){printf("GROUND CONTACT, EOE\n");}
            rewardHistory =REWARD_LOSS;
            newReward     = true;
            endEpisode    = true;
        }
```

Touch the ground is ban, it is too danger to touch ground for robot so I set rewardHistory=REWARD_LOSS.

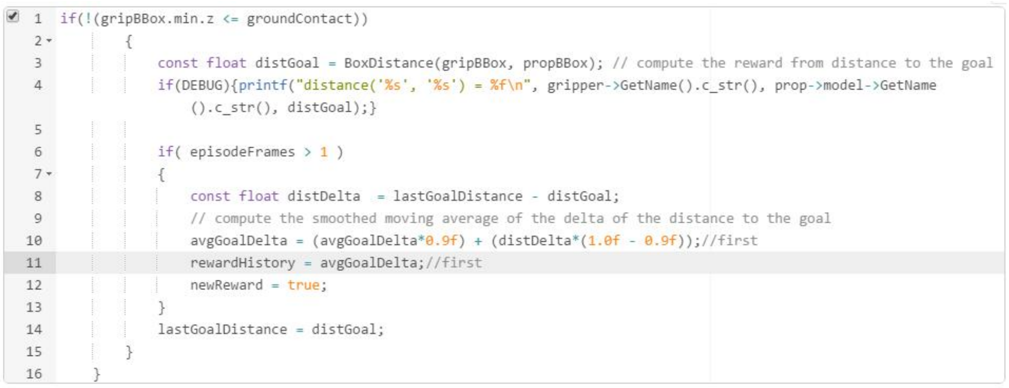
3) episodeFrames > maxEpisodeLength,I set reward function like this:



```
if( maxEpisodeLength > 0 && episodeFrames > maxEpisodeLength )
    {
      printf("ArmPlugin - triggering EOE, episode has exceeded %i frames\n", maxEpisodeLength);
        rewardHistory = REWARD_LOSS*0.2f;
        newReward     = true;
        endEpisode    = true;
    }
```

Arm not tough anything is bad, but I think it is better than touch ground,so I set rewardHistory=REWARD_LOSS*0.2f

4) Distance reward



```
if(!(gripBBox.min.z <= groundContact))
        {
            const float distGoal = BoxDistance(gripBBox, propBBox); // compute the reward from distance to the goal
            if(DEBUG){printf("distance('%s', '%s') = %f\n", gripper->GetName().c_str(), prop->model->GetName
                ().c_str(), distGoal);}

            if( episodeFrames > 1 )
            {
                const float distDelta  = lastGoalDistance - distGoal;
                // compute the smoothed moving average of the delta of the distance to the goal
                avgGoalDelta = (avgGoalDelta*0.9f) + (distDelta*(1.0f - 0.9f));//first
                rewardHistory = avgGoalDelta;//first
                newReward = true;
            }
            lastGoalDistance = distGoal;
        }
    }
```

I set distance reward like this just for reward the step to close goal, I want the arm more and more close the object.

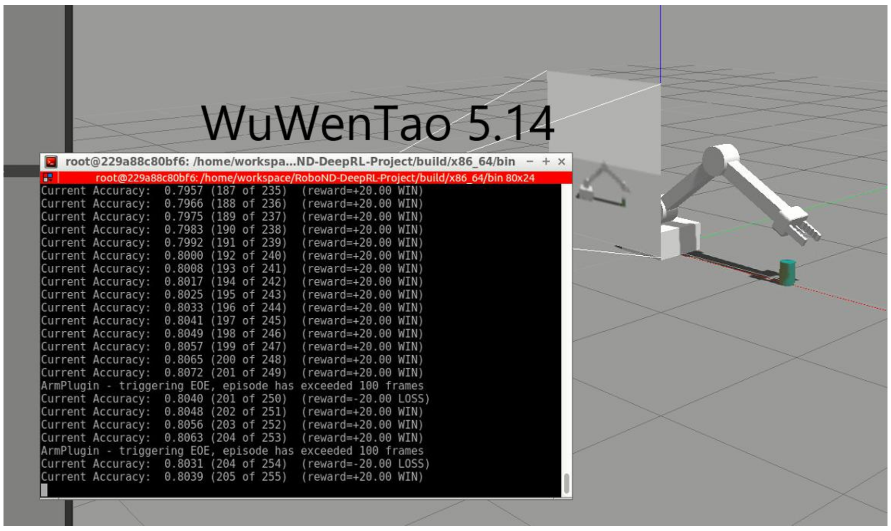
(3) Hyperparameters

#define VELOCITY_CONTROL true
#define INPUT_WIDTH 64
#define INPUT_HEIGHT 64
#define OPTIMIZER "RMSprop"
#define LEARNING_RATE 0.01f
#define REPLAY_MEMORY 10000
#define BATCH_SIZE 32
#define USE_LSTM true
#define LSTM_SIZE 256
#define REWARD_WIN   520.0f
#define REWARD_LOSS -520.0f

In the objectives 1 , I set input=64*64, LSTM_SIZE 256 and BATCH_SIZE 32, these parameters is important , but in my objectives 1 I believe VELOCITY_CONTROL=true is most interesting, we can see the pic-2,the target is hit and fly. It use joint or nor velocity. When I complete objectives 2 ,mabe, I can get better result though the code of objectives 2. But I do not want to modify it, because it is interesting.

For the second project:

(1) Result

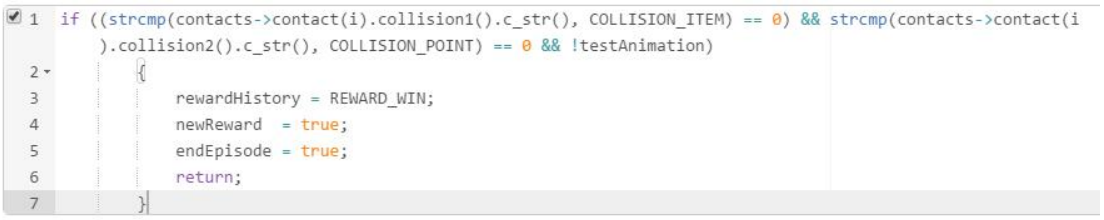

WuWenTao 5.14
root@229a88c80bf6: /home/workspa...ND-DeepRL-Project/build/x86_64/bin
root@229a88c80bf6: /home/workspace/RoboND-DeepRL-Project/build/x86_64/bin 80x24

```
Current Accuracy:  0.7957 (187 of 235)  (reward=+20.00 WIN)
Current Accuracy:  0.7966 (188 of 236)  (reward=+20.00 WIN)
Current Accuracy:  0.7975 (189 of 237)  (reward=+20.00 WIN)
Current Accuracy:  0.7983 (190 of 238)  (reward=+20.00 WIN)
Current Accuracy:  0.7992 (191 of 239)  (reward=+20.00 WIN)
Current Accuracy:  0.8000 (192 of 240)  (reward=+20.00 WIN)
Current Accuracy:  0.8008 (193 of 241)  (reward=+20.00 WIN)
Current Accuracy:  0.8017 (194 of 242)  (reward=+20.00 WIN)
Current Accuracy:  0.8025 (195 of 243)  (reward=+20.00 WIN)
Current Accuracy:  0.8033 (196 of 244)  (reward=+20.00 WIN)
Current Accuracy:  0.8041 (197 of 245)  (reward=+20.00 WIN)
Current Accuracy:  0.8049 (198 of 246)  (reward=+20.00 WIN)
Current Accuracy:  0.8057 (199 of 247)  (reward=+20.00 WIN)
Current Accuracy:  0.8065 (200 of 248)  (reward=+20.00 WIN)
Current Accuracy:  0.8072 (201 of 249)  (reward=+20.00 WIN)
ArmPlugin - triggering EOE, episode has exceeded 100 frames
Current Accuracy:  0.8040 (201 of 250)  (reward=-20.00 LOSS)
Current Accuracy:  0.8048 (202 of 251)  (reward=+20.00 WIN)
Current Accuracy:  0.8056 (203 of 252)  (reward=+20.00 WIN)
Current Accuracy:  0.8063 (204 of 253)  (reward=+20.00 WIN)
ArmPlugin - triggering EOE, episode has exceeded 100 frames
Current Accuracy:  0.8031 (204 of 254)  (reward=-20.00 LOSS)
Current Accuracy:  0.8039 (205 of 255)  (reward=+20.00 WIN)
```

(1)The second project result

(2) Reward Functions

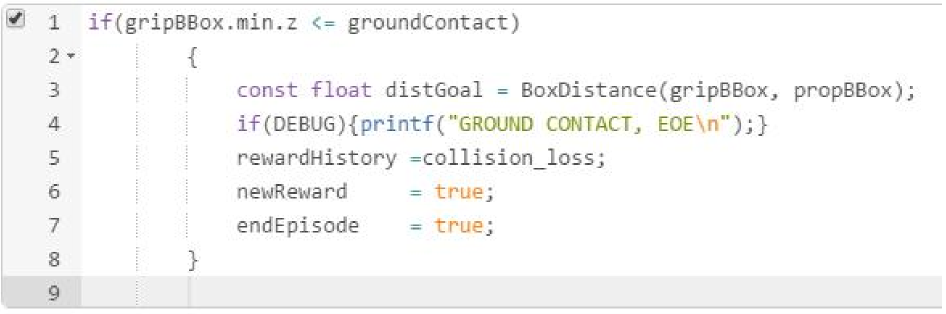
1) When only the gripper base of the robot arm touch the object, I set reward function like this:



```
if ((strcmp(contacts->contact(i).collision1().c_str(), COLLISION_ITEM) == 0) && strcmp(contacts->contact(i
    ).collision2().c_str(), COLLISION_POINT) == 0 && !testAnimation)
        {
            rewardHistory = REWARD_WIN;
            newReward  = true;
            endEpisode = true;
            return;
        }
```

Add new constraints to segment whether only the gripper base of the robot arm touch the object .

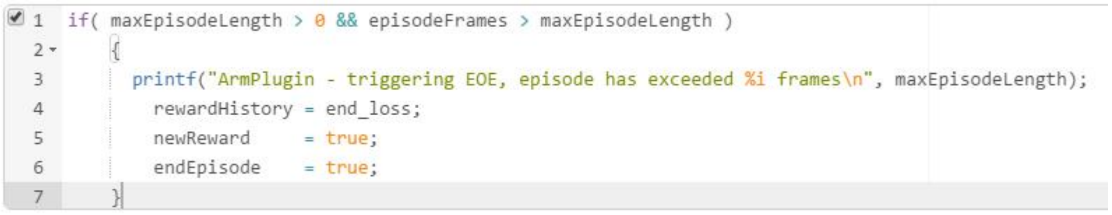
2) When have any part of the robot arm touch the ground, I set reward function like this:



```
if(gripBBox.min.z <= groundContact)
        {
            const float distGoal = BoxDistance(gripBBox, propBBox);
            if(DEBUG){printf("GROUND CONTACT, EOE\n");}
            rewardHistory =collision_loss;
            newReward     = true;
            endEpisode    = true;
        }
```

Collision_loss had been set in the code top, it is just for convenient adjustment parameters .In last, I set collision_loss=REWARD_LOSS

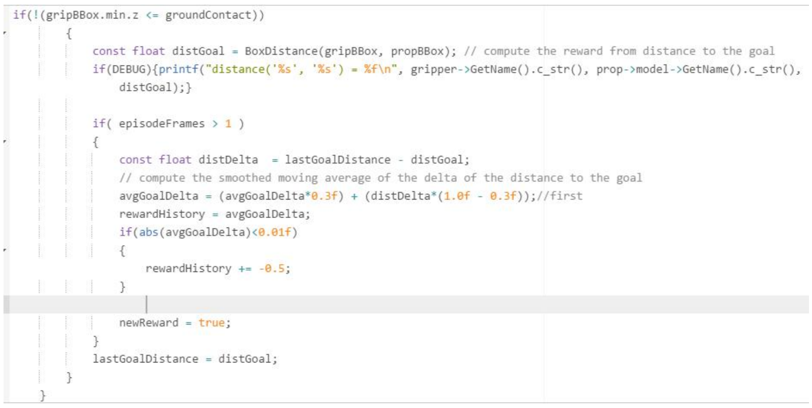
3) episodeFrames > maxEpisodeLength,I set reward function like this:



```
if( maxEpisodeLength > 0 && episodeFrames > maxEpisodeLength )
    {
      printf("ArmPlugin - triggering EOE, episode has exceeded %i frames\n", maxEpisodeLength);
        rewardHistory = end_loss;
        newReward     = true;
        endEpisode    = true;
    }
```

end_loss had been set in the code top, it is just for convenient adjustment parameters .In last, I set end_loss=REWARD_LOSS. It is not any special.

4) Distance reward



```
if(!(gripBBox.min.z <= groundContact))
        {
            const float distGoal = BoxDistance(gripBBox, propBBox); // compute the reward from distance to the goal
            if(DEBUG){printf("distance('%s', '%s') = %f\n", gripper->GetName().c_str(), prop->model->GetName().c_str(),
                distGoal);}

            if( episodeFrames > 1 )
            {
                const float distDelta  = lastGoalDistance - distGoal;
                // compute the smoothed moving average of the delta of the distance to the goal
                avgGoalDelta = (avgGoalDelta*0.3f) + (distDelta*(1.0f - 0.3f));//first
                rewardHistory = avgGoalDelta;
                if(abs(avgGoalDelta)<0.01f)
                {
                    rewardHistory += -0.5;
                }

                newReward = true;
            }
            lastGoalDistance = distGoal;
        }
    }
```

If the arm close the goal step and step it will be great, so I set

rewardHistoy=avaGoalDelta

But if the step too slow to close the goal ,I punish the step by add -0.5. It will Encourage arm to close the goal more quickly. At the same time, it can also reduce operation time.

(3) Hyperparameters

#define VELOCITY_CONTROL false
#define INPUT_WIDTH 64
#define INPUT_HEIGHT 64

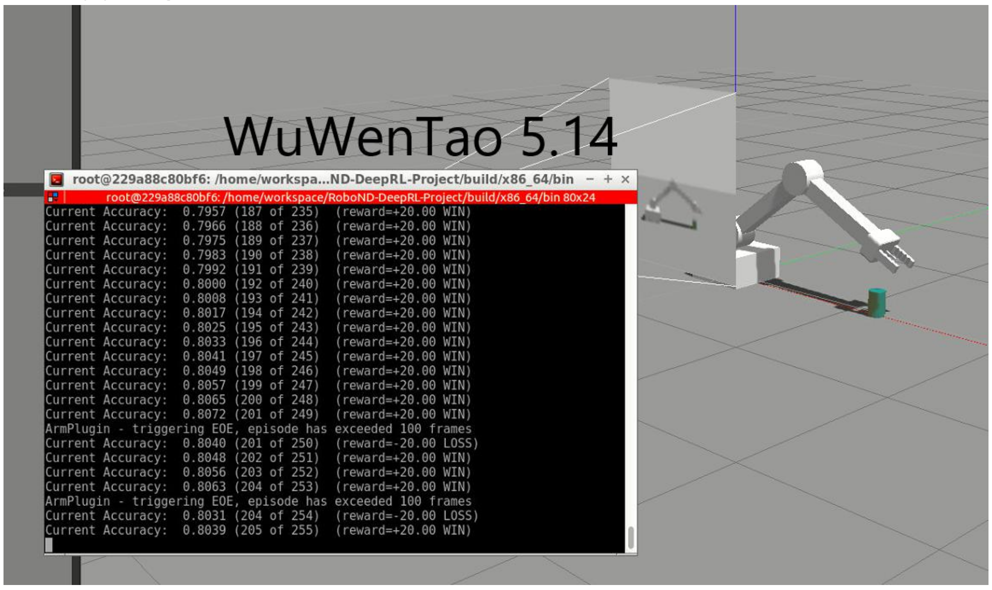
```
#define OPTIMIZER "Adam"
#define LEARNING_RATE 0.01f
#define REPLAY_MEMORY 20000
#define BATCH_SIZE 516
#define USE_LSTM true
#define LSTM_SIZE 256
#define REWARD_WIN    20.0f
#define REWARD_LOSS -20.0f
#define end_loss REWARD_LOSS
#define collision_loss REWARD_LOSS
```

In the objectives 2 , I set VELOCITY_CONTROL=false to avoid hit the goal again, enlarge BATCH_SIZE to 516 and LSTM_SIZE to 256 for add the LSTM learning ability and speed. Change OPTIMIZER="Adam" and LEARNING_RATE=0.01f is the result of constant tuning parameter.Through the experiment, define REWARD_WIN=20.0f and REWARD_LOSS=-20.0f is enough.

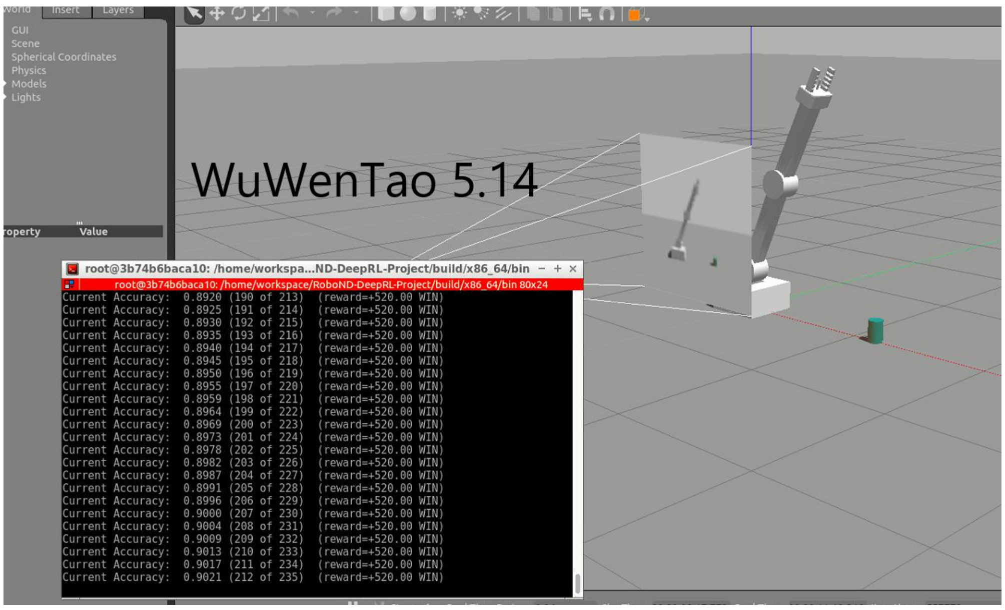
Result

From the pic above, we complete all requirment. As I tried to keep trying all the solutions, I found that the large LSTM structure greatly enhanced the ability to learn, but the oversized structure was unbearable in the workspace.For reward function, we just need set suitable parameter, to reward good behavior and punish bad behavior. When I over this project I found some defect in it: if the goal in the outside of the camera how arm to find where to go. After thinking, I got some future work.

Put my pic again:


WuWenTao 5.14
root@229a88c80bf6: /home/workspa...ND-DeepRL-Project/build/x86_64/bin
root@229a88c80bf6: /home/workspace/RoboND-DeepRL-Project/build/x86_64/bin 80x24

```
Current Accuracy:  0.7957 (187 of 235)  (reward=+20.00 WIN)
Current Accuracy:  0.7966 (188 of 236)  (reward=+20.00 WIN)
Current Accuracy:  0.7975 (189 of 237)  (reward=+20.00 WIN)
Current Accuracy:  0.7983 (190 of 238)  (reward=+20.00 WIN)
Current Accuracy:  0.7992 (191 of 239)  (reward=+20.00 WIN)
Current Accuracy:  0.8000 (192 of 240)  (reward=+20.00 WIN)
Current Accuracy:  0.8008 (193 of 241)  (reward=+20.00 WIN)
Current Accuracy:  0.8017 (194 of 242)  (reward=+20.00 WIN)
Current Accuracy:  0.8025 (195 of 243)  (reward=+20.00 WIN)
Current Accuracy:  0.8033 (196 of 244)  (reward=+20.00 WIN)
Current Accuracy:  0.8041 (197 of 245)  (reward=+20.00 WIN)
Current Accuracy:  0.8049 (198 of 246)  (reward=+20.00 WIN)
Current Accuracy:  0.8057 (199 of 247)  (reward=+20.00 WIN)
Current Accuracy:  0.8065 (200 of 248)  (reward=+20.00 WIN)
Current Accuracy:  0.8072 (201 of 249)  (reward=+20.00 WIN)
ArmPlugin - triggering EOE, episode has exceeded 100 frames
Current Accuracy:  0.8040 (201 of 250)  (reward=-20.00 LOSS)
Current Accuracy:  0.8048 (202 of 251)  (reward=+20.00 WIN)
Current Accuracy:  0.8056 (203 of 252)  (reward=+20.00 WIN)
Current Accuracy:  0.8063 (204 of 253)  (reward=+20.00 WIN)
ArmPlugin - triggering EOE, episode has exceeded 100 frames
Current Accuracy:  0.8031 (204 of 254)  (reward=-20.00 LOSS)
Current Accuracy:  0.8039 (205 of 255)  (reward=+20.00 WIN)
```


World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Models
Lights
Property
Value
WuWenTao 5.14
root@3b74b6baca10: /home/workspa...ND-DeepRL-Project/build/x86_64/bin
root@3b74b6baca10: /home/workspace/RoboND-DeepRL-Project/build/x86_64/bin 80x24

```
Current Accuracy:  0.8920 (190 of 213)  (reward=+520.00 WIN)
Current Accuracy:  0.8925 (191 of 214)  (reward=+520.00 WIN)
Current Accuracy:  0.8930 (192 of 215)  (reward=+520.00 WIN)
Current Accuracy:  0.8935 (193 of 216)  (reward=+520.00 WIN)
Current Accuracy:  0.8940 (194 of 217)  (reward=+520.00 WIN)
Current Accuracy:  0.8945 (195 of 218)  (reward=+520.00 WIN)
Current Accuracy:  0.8950 (196 of 219)  (reward=+520.00 WIN)
Current Accuracy:  0.8955 (197 of 220)  (reward=+520.00 WIN)
Current Accuracy:  0.8959 (198 of 221)  (reward=+520.00 WIN)
Current Accuracy:  0.8964 (199 of 222)  (reward=+520.00 WIN)
Current Accuracy:  0.8969 (200 of 223)  (reward=+520.00 WIN)
Current Accuracy:  0.8973 (201 of 224)  (reward=+520.00 WIN)
Current Accuracy:  0.8978 (202 of 225)  (reward=+520.00 WIN)
Current Accuracy:  0.8982 (203 of 226)  (reward=+520.00 WIN)
Current Accuracy:  0.8987 (204 of 227)  (reward=+520.00 WIN)
Current Accuracy:  0.8991 (205 of 228)  (reward=+520.00 WIN)
Current Accuracy:  0.8996 (206 of 229)  (reward=+520.00 WIN)
Current Accuracy:  0.9000 (207 of 230)  (reward=+520.00 WIN)
Current Accuracy:  0.9004 (208 of 231)  (reward=+520.00 WIN)
Current Accuracy:  0.9009 (209 of 232)  (reward=+520.00 WIN)
Current Accuracy:  0.9013 (210 of 233)  (reward=+520.00 WIN)
Current Accuracy:  0.9017 (211 of 234)  (reward=+520.00 WIN)
Current Accuracy:  0.9021 (212 of 235)  (reward=+520.00 WIN)
```

Future work

For improve the results, I think we need more larger LSTM structure and time to train the model. Add the imput image’s size is important too, it representative how much information will be learning for LSTM. And lastly, I think we can add new cameras in difference place to collect more information and input all camera’s pic into LSTM at a time, in these method LSTM could get more information at the same time. In order to catch the goal surely, we can add camera in the arm’s grip, RGB-D camera may be useful to it for get distance information.That is all my think about this project.