Reinforcement Learning: Putting it all Together

We are now ready to put the entire reinforcement learning pipeline together. Below we present a pseudo code of the learning procedure that you should implement in your function.

Algorithm 1 Full Reinforcement Learning Pipeline

```
Initialization: Initialize the parameters of a network
for 1 to N do
    Set the current_state to the start_state
    while current_state is not goal_state do
        action=PickAction()
        reward=GetReward()
        new_state=MakeNextState()
        nn=DeepQLearning()
        if reward>=0 then
            set current_state to new_state
        end if
    end while
end for
```

Note that a typical reinforcement learning procedure may take a long time. Therefore, we provide a network file that has been trained for 500 iterations but has not converged yet. You will be able to use this network file as an initialization network to speed up your training.

After running your reinforcement learning code for 100 iterations, the network should converge. Using such a network to select actions at every step, will allow the robot to reach the goal state in the minimum number of steps, which is 14 in this case.

Note that if you are working on this problem outside of the EdX environment, you should comment out or remove the line "rand('state',0);" from your `PickAction.m` function. If you are working inside the EdX environment, you don't need to worry about this.

Your Function

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

```
function nn=ReinforcementLearning
    % A full neural network based reinforcement learning pipeline
    %
    % Output:
    % - nn: a variable storing a neural network with the updated weights after full training.

    %% Initial Parameters
    epsilon=0.5;

    rows=7; cols=7;
    walls=[2 4; 3 4; 4 4; 5 4];
    cur_row=2; cur_col=1; rot_idx=1;

    %% Create A Start state
    start_S=MakeState(rows,cols,walls,cur_row,cur_col,rot_idx);

    %% Network Initialization
    load('RL_nn_500.mat');

    N=100;

    %% loop N times
    for i=1:N
        %% Resetting to a start state
```

Code to call your function

Reset

```
nn=ReinforcementLearning();
action_path=Prediction(nn);
number_of_actions_taken=size(action_path,1);
```

Run Function

Previous Assessment: All Tests Passed

Submit

Does the Network Produce a Correct Solution?