## **Computing a Reward**

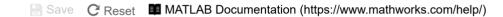
After every selected action, the robot needs to get a reward to learn whether that action was worth making or not. For instance, if a robot takes an action that leads to discovering the goal state then that action should be associated with a very large reward value so that the robot would learn to take that action at that particular state. Conversely, if a robot takes an action and hits the wall or steps out of a grid then the reward should be negative to discourage the robot from taking that action.

We will be considering four distinct reward cases:

- 1. When a robot hits the wall or steps out of the grid, the reward should be assigned to MIN VAL=-99.
- 2. When a robot reaches the goal state the reward should be assigned to MAX VAL=99.
- 3. When a robot selects an action that forces the next state to be the same as the current state the reward should be assigned to MIN VAL=-99.
- 4. Finally, in all the other cases, the reward should be assigned to zero value.

Note that for this part of the assignment, we provide you with a function MakeNextState that computes the next state given a selected action. In your code, you are also allowed to call a previously written function isGoal.

## **Your Function**



```
1 function reward=GetReward(S,cur row,cur col,rot idx,action)
 2
       % Computed the reward resulting from a selected action
      %
 3
 4
      % Input:
 5
      % - S: n x m matrix that stores a current state
 6
      % - cur_row: the row position of a robot's centroid
 7
       % - cur_col: the column position of a robot's centroid
 8
       % - rot_idx: a scalar indicating which rotation
 9
       % - action: a scalar indicating a selected action
10
              (1-going down, 2-going right, 3-going up, 4-going left
11
                5-rotating to rotation position 1, 6-rotating to rotation position 2
12
                7-rotating to rotation position 3, 8-rotating to rotation position 4).
13
      % Output:
      % - reward: a scalar indicating the received reward
14
15
16
      MIN VAL=-99;
17
     MAX_VAL=99;
18
19
      % Getting next state
20
      [next_S,~,~,~]=MakeNextState(S,cur_row,cur_col,rot_idx,action);
21
22
      % Getting a reward
23
24
      if max(next_S(:))<=0 % CASE 1: the robot either hit the wall or stepped outside of the grid
25
          reward= MIN_VAL;
26
      else %the robot performed a valid action
27
        if isGoal(next_S) % CASE 2: case where the robot reached the goal
28
             reward= MAX VAL;
29
         elseif next_S == S % CASE 3: case where the next state is the same as the current state
30
             reward= MIN_VAL;
31
         else % CASE 4: every other case
32
             reward= 0;
33
         end
34
      end
35
36 end
```

## Code to call your function

C Reset

```
2 % Current state
4 rows=7; cols=7;
5 walls=[2 4; 3 4; 4 4; 5 4];
6 cur_row=2; cur_col=1; rot_idx=1;
7 S=MakeState(rows,cols,walls,cur_row,cur_col,rot_idx);
 action=1;
```

8 ► Run Function

## **Previous Assessment: All Tests Passed**

Submit ?

- Is Case 1 Correct?
- Is Case 2 Correct?
- Is Case 3 Correct?
- Is Case 4 Correct?