# 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

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

```
function reward=GetReward(S,cur_row,cur_col,rot_idx,action)
    % Computed the reward resulting from a selected action
    %
    % Input:
    % - S: n x m matrix that stores a current state
    % - cur_row: the row position of a robot's centroid
    % - cur_col: the column position of a robot's centroid
    % - rot_idx: a scalar indicating which rotation
    % - action: a scalar indicating a selected action
    %      (1-going down, 2-going right, 3-going up, 4-going left
    %        5-rotating to rotation position 1, 6-rotating to rotation position 2
    %        7-rotating to rotation position 3, 8-rotating to rotation position 4).
    % Output:
    % - reward: a scalar indicating the received reward

   MIN_VAL=-99;
   MAX_VAL=99;

   % Getting next state
   [next_S,~,~,~]=MakeNextState(S,cur_row,cur_col,rot_idx,action);

   % Getting a reward

   if max(next_S(:))<=0 % CASE 1: the robot either hit the wall or stepped outside of the grid
       reward= MIN_VAL;
   else %the robot performed a valid action
      if isGoal(next_S) % CASE 2: case where the robot reached the goal
          reward= MAX_VAL;
      elseif next_S == S % CASE 3: case where the next state is the same as the current state
          reward= MIN_VAL;
      else % CASE 4: every other case
          reward= 0;
      end
   end

end

```

### Code to call your function

Reset

```

```

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

▶ Run Function

## Previous Assessment: All Tests Passed

Submit

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