CS747 - Assignment 3

Name - Utkarsh Ranjan
Roll No. - 200050147

Task 1

Out of the three approaches suggested in the assignment to solve the task, (i) coding it up yourself, (ii) learning with value function, (iii) using policy search, I used the first one i.e, **coding up the policy myself**

For the first task, because the reward for the each step was -1 , for reaching the road it was 100, and for the icy part of the road, it was -100. So one of the optimal way through which we could reach the road is simply first turning the car towards the road and then accelerating it to the road finally.

If we consider a new state S' $\in \mathfrak{R}^2$ derived from the state $S \in \mathfrak{R}^4$ given to us by taking the projection of S along the first two dimension, we obtain the new set of states as [x,y] without the angle and velocity component. If for this given state, we think of an action as moving in one of the 360 possible direction, our action would be a composition of the actions defined now, with initial steps involving rotating the car in that direction and then giving some acceleration without steering. Note that we can consider these composition of action, for our newly defined set of states S', these actions are valid because rotation of car along any direction without any motion still gives a reward of 0.

So now our new set of states are simply, (x,y) i.e, location of the car in the parking lot, while new set of actions are simply rotating and moving along a particular direction. With this derived environment if we think of our problem. We can conclude that for a given state s $\in$ S' , Q(s,a) is maximum for the action which moves the cars straight towards the direction of the road.

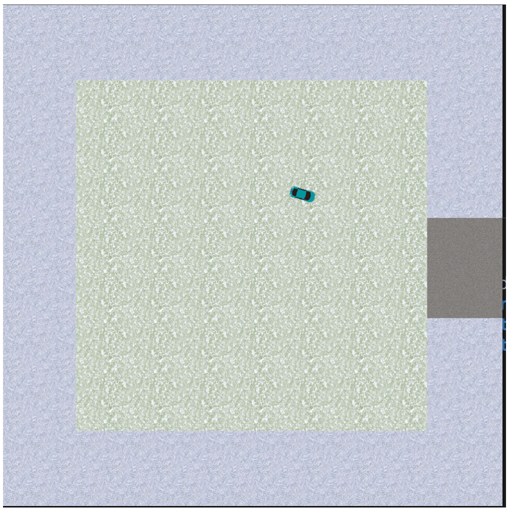
Here the important point to keep in mind is that, while making the turn car receive a reward of 0, which is better than the -1 reward which it is getting otherwise by taking some step (i.e, non-zero acceleration). And finally, moving towards the road would have the highest Q(S,a) value for that direction which takes the car straight towards the road.

Logic:

```
e = 10 // a threshold to compare the angle between direction in which we need to move and the head of the car.

// angle is the angle of the car with the road
// omega is the head of the car (obatined from the )
while(abs(angle - omega) > e):
    rotate()
accelerate()
```

Demo:



Task 2

For this task, the approach above doesn't give the optimal policy because falling into the dirt also gives, a strong negative reward of -100.

So the policy designed for this case was as follows:-

(i) The car first moves towards the horizontal central line (x-axis) of the parking lot. But before that it checks if there is a dirt in its way to the central line.
(ii) If there is a dirt in its way to the central line, it simply turn towards the central vertical line, reach the central vertical line, from there moves towards the central-x line , rotate towards the road and then finally heads to the road.

This technique works because there isn't exist any dirt near the axis of our coordinate plane. Also if there is a dirt, in its way to the central horizontal line, it is guaranteed that there won't be any dirt in the way to the central vertical line, because of the way dirt is initialized in the environment.

Logic:

```
Check if can go straight to the y=0 line:

if CanGo:
    policy1
else:
    policy2

policy1:
    Turn towards the y=0 line
    Move towards the y=0 line
    If within an e-range of y=0 line, i.e, abs(y) < e:
        Turn towards the road
        Move towards the x=0 line
policy2:
    Turn towards the x=0 line
    Move towards the x=0 line
    If within an e-range of x=0 line, i.e, abs(x) < e:
        Turn towards the y=0 line
        Move towards the y=0 line
        If within an e-range of y=0 and x=0 line, i.e, abs(y) < e and ans(x) < e:
        Turn towards the road
        Move towards the x=0 line
```

Demo: