

## CS747: Programming Assignment 3

### Task 1:

#### Approach 1:

Tried to apply Q-learning control algorithm to the problem using discretized/approximated and not approximated states.

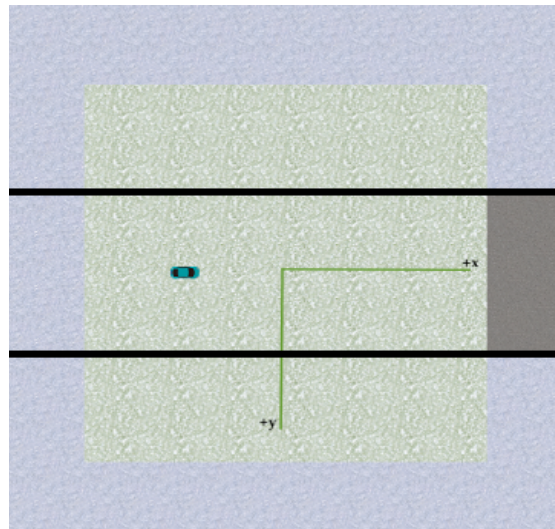
#### Observation:

State space was observed to be very large even after discretization due to which no learning was happening as Q values were not getting updated. Hence this algorithm was not used.

#### Final Approach:

I coded a controller based on if-else logic using which car can travel to the road.

First, the grid was divided into three parts: above the road, below the road, and along the road.



Following is the logic if the initial state of the car is in along the road or the y coordinate lies between -100 and 100(length of road).

The inner loops first rotate the car such that it is approximately zero and upon reaching that we travel in the x direction to reach the road.

```
if(state[1]>=-100 and state[1]<=100):  
    if(state[3]>=180):  
        action_steer = 2  
        action_acc = 3  
    elif(state[3]<180):  
        action_steer = 0  
        action_acc = 3  
    else:  
        action_steer = 1  
        action_acc = 0
```

Now if the car lies in the above portion of the grid then we first rotate it so that it makes a 90 angle with the x-axis upon which it is moved in the vertical direction till it reaches the middle section.

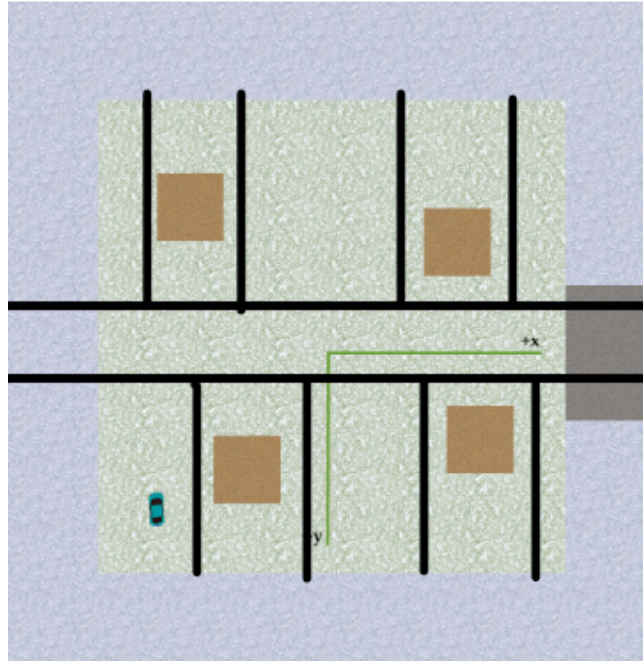
```
elif(state[1]<-100):  
    if(state[3]>270 or state[3]<90):  
        action_steer = 2  
        action_acc = 3  
    elif(state[3]<=270 or state[3]>90):  
        action_steer = 0  
        action_acc = 3  
    else:  
        action_steer = 1  
        action_acc = 3
```

Lastly, we have the code for the bottom section which similar to the top is first rotated to make an angle of 270 and then moved vertically upwards till it reaches the middle section.

```
else:  
    if(state[3]>270 or state[3]<90):  
        action_steer = 0  
        action_acc = 3  
    elif(state[3]<=270 or state[3]>90):  
        action_steer = 2  
        action_acc = 3  
    else:  
        action_steer = 1  
        action_acc = 3
```

## Task 2:

In this task also a controller was coded taking into account the center locations of the pits which were given and applying if-else logic. The grid was divided into the following sort of configuration



If the car lies in the middle portion then it is rotated to zero degrees and then moved forward. The line is drawn at y coordinates of -70 and 70 as that is the maximum reach of a pit.

```
if(state[1]>=-70 and state[1]<=70):
    if(state[3]>=180 and state[3]<=357):
        action_steer = 2
        action_acc = 2
    elif(state[3]>=3 and state[3]<180):
        action_steer = 0
        action_acc = 2
    else:
        action_steer = 1
        action_acc = 4
```

Now if it lies in the upper or lower portion and above or below a box then first the car is rotated to zero degrees and moved horizontally such that it's x coordinate is not in the range of the box. After that, it is rotated 90/270 degrees and moved vertically till it reaches the middle portion. From there it is again rotated to 0 degrees and accelerated to reach the road.

```

elif(state[1]<-70):
    if((state[0]>= (self.list[3][0]-70) and state[0]<=(self.list[3][0]+70))or
       (state[0]>= (self.list[1][0]-70) and state[0]<=(self.list[1][0]+70))):
        if(state[3]>=180 and state[3]<=358):
            action_steer = 2
            action_acc = 2
        elif(state[3]>=3 and state[3]<180):
            action_steer = 0
            action_acc = 2
        else:
            action_steer = 1
            action_acc = 4

```

If the car is not in range of the pit then simply Task 1 is repeated which means rotate till 90/270, move vertically till the middle section, then rotate to 0 degrees, and move horizontally.

```

else:
    if(state[3]>=268 or state[3]<=88):
        action_steer = 2
        action_acc = 2
    elif(state[3]<=272 and state[3]>=92):
        action_steer = 0
        action_acc = 2
    else:
        action_steer = 1
        action_acc = 4

```