TUTORIAL ASSIGNMENT 1
ARTIFICIAL INTILLIGENCE

Shubham Kumar.IIT2018146
V Semester B-Tech., Information Technology,

Abstract: In this paper, we have devised an algorithm that how an agent moved in the environment when the goal position is known and when there is a sound sensor is placed in the environment.
In both cases, different algorithm is used to reach the goal .

1. INTRODUCTION

A grid of size m*n is given which is considered as an environment .There is some agent in the environment which is somewhere on the grid .The agent has a position sensor which gives the (x,y) coordinates of the agent at any time.There are some restricted move which agent can move and the move are left or right or up or down, agent can only move in this direction.

Now in the first part , the goal position that is x coordinate and y coordinate of the goal is known in advance. The agent aim is to reach the goal by moving in four direction only,It is given that the grid doesn't have any obstacle so it will be an easy task for agent to reach the goal without having any collision with any obstacle.

In the second part, the goal position is not known but in this part there is a sound sensor which is situated somewhere in the environment.The sound sensor which is situated will always give the distance of the agent from the sound sensor. So, in this part each time an agent have to decide whether the agent have to move right or have to move left or up or down .The decision will

mainly depend on the distance which is given by the sound sensor so that agent make right decision in order to take the right step.

2. ENVIRONMENT CLASS

In the environment class, there is 2d grid which of size m*n .
Now in the environment class we have defined the initialise the grid size i.e max row and max columns.

Secondly we will take the goal position x coordinate and y coordinate in the constructor which is defined in the Environment class

After then ggetgoalposition_x() and getgoalposition_y() function is defined to get the x coordinate position of goal and y coordinate position of y.

3. AGENT CLASS

In the agent class we have taken the agent x coordinate and y coordinate position which is basically the sensor position of the agent .

Secondly i have made getpositionsensor_x
And getpositionsensor_y() in order to get the x and y position of the sensor.
After then I have made a valid_move() function to check if the given coordinate is valid or not.
Then some more function has been constructed like moveright(),moveleft() etc.. to move the position of agent to right or to left respectively/.
After then there is a function findpath() which is used to find the path from the initial agent position to the goal position.

4. ALGORITHM

For the first case:-

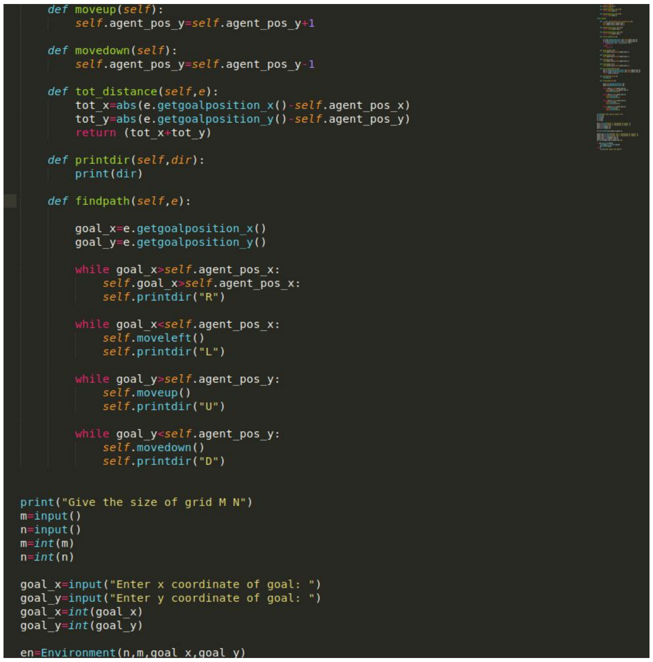
In the first part we simply check while goal_x>self.agent_pos_x then moveright() and print(“R)
Secondly we check while goal_x < self.agent_pos_x then moveleft()
And print(“L)
Thirdly, we check while goal_y > self.agent_pos_y:then moveup() and print(“U)
After then we check that while goal_y<self.agent_pos_y: the movcedown() and print(“D”)

For the second case:-

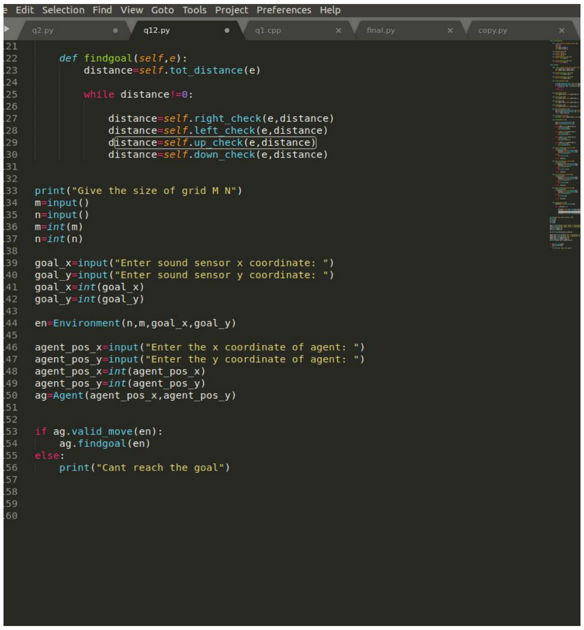
In this case we simply check if the distance is not equal to the zero then we try to move in all direction and check the distance if distance is less than the previous distance then we simply move in the that direction and print the position of their agent

while distance!=0:
distance=self.right_check(e,distance)
distance=self.left_check(e,distance)
distance=self.up_check(e,distance)
distance=self.down_check(e,distance)

5. CODE /SNIPPET



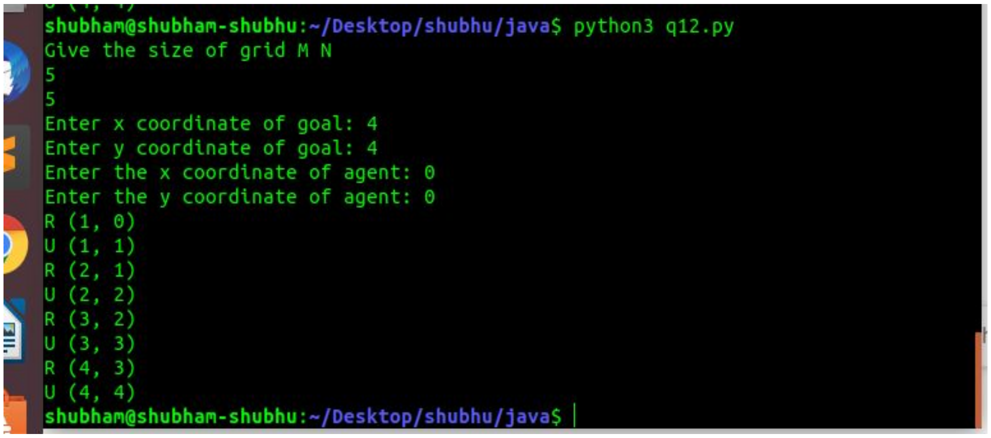
```
    def moveup(self):
        self.agent_pos_y=self.agent_pos_y+1

    def movedown(self):
        self.agent_pos_y=self.agent_pos_y-1

    def tot_distance(self,e):
        tot_x=abs(e.getgoalposition_x()-self.agent_pos_x)
        tot_y=abs(e.getgoalposition_y()-self.agent_pos_y)
        return (tot_x+tot_y)

    def printdir(self,dir):
        print(dir)

    def findpath(self,e):

        goal_x=e.getgoalposition_x()
        goal_y=e.getgoalposition_y()

        while goal_x>self.agent_pos_x:
            self.goal_x>self.agent_pos_x:
            self.printdir("R")

        while goal_x<self.agent_pos_x:
            self.moveleft()
            self.printdir("L")

        while goal_y>self.agent_pos_y:
            self.moveup()
            self.printdir("U")

        while goal_y<self.agent_pos_y:
            self.movedown()
            self.printdir("D")


print("Give the size of grid M N")
m=input()
n=input()
m=int(m)
n=int(n)

goal_x=input("Enter x coordinate of goal: ")
goal_y=input("Enter y coordinate of goal: ")
goal_x=int(goal_x)
goal_y=int(goal_y)

en=Environment(n,m,goal_x,goal_y)
```



e Edit Selection Find View Goto Tools Project Preferences Help
q2.py
q12.py
q1.cpp
final.py
copy.py

```
    def findgoal(self,e):
        distance=self.tot_distance(e)

        while distance!=0:

            distance=self.right_check(e,distance)
            distance=self.left_check(e,distance)
            distance=self.up_check(e,distance)
            distance=self.down_check(e,distance)


print("Give the size of grid M N")
m=input()
n=input()
m=int(m)
n=int(n)

goal_x=input("Enter sound sensor x coordinate: ")
goal_y=input("Enter sound sensor y coordinate: ")
goal_x=int(goal_x)
goal_y=int(goal_y)

en=Environment(n,m,goal_x,goal_y)

agent_pos_x=input("Enter the x coordinate of agent: ")
agent_pos_y=input("Enter the y coordinate of agent: ")
agent_pos_x=int(agent_pos_x)
agent_pos_y=int(agent_pos_y)
ag=Agent(agent_pos_x,agent_pos_y)


if ag.valid_move(en):
    ag.findgoal(en)
else:
    print("Cant reach the goal")
```

6. OUTPUT



```
shubham@shubham-shubhu:~/Desktop/shubhu/java$ python3 q12.py
Give the size of grid M N
5
5
Enter x coordinate of goal: 4
Enter y coordinate of goal: 4
Enter the x coordinate of agent: 0
Enter the y coordinate of agent: 0
R (1, 0)
U (1, 1)
R (2, 1)
U (2, 2)
R (3, 2)
U (3, 3)
R (4, 3)
U (4, 4)
shubham@shubham-shubhu:~/Desktop/shubhu/java$ 
```