# Moving an agent source to Destination in given Environment

***Abstract: In this paper, we have devised an algorithm for following situations:***
***a) Assume a grid world of size m*n with no obstacles. Make an environment and agent class. The agent has a position sensor that gives the (x,y) coordinates of the agent at any time. The agent is moved by the actions left, right, up and down. The goal coordinates are known in advance.***
***b) Suppose the goal coordinates are not known but there is a sensor that detects the distance to the goal. As an example, the goal can be a sound source and the sensor is a microphone. Simulate under this setting.***

## I. INTRODUCTION

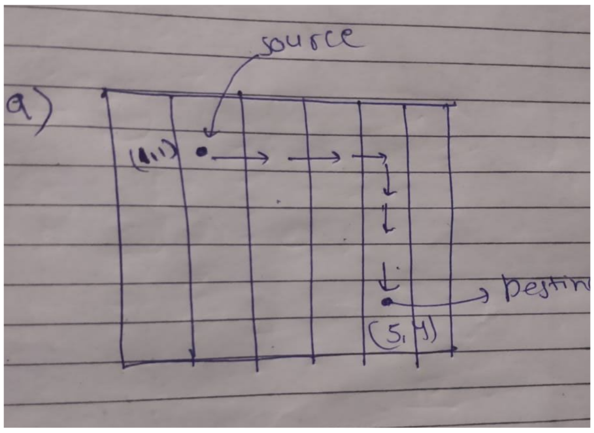
we made An environment and Agent class to simulate the above situation:
environment have dimensions of boundary i.e. max width and max height
Agent have an environment source destination
i) In the first part we made A function of mynextMove by which made a move of Agent in the valid direction till it reached his destination.
ii)in the second part we made A function of mynextMove by which made a move of Agent by calculating its distance from the goal position. move that Agent in direction where distance becomes minimum.

## II. ALGORITHM DESCRIPTION

This algorithm simulates in two steps
Stage 1: is a valid direction then it goes in that direction and change the current position
Stage2: record each move into an Array list of positions

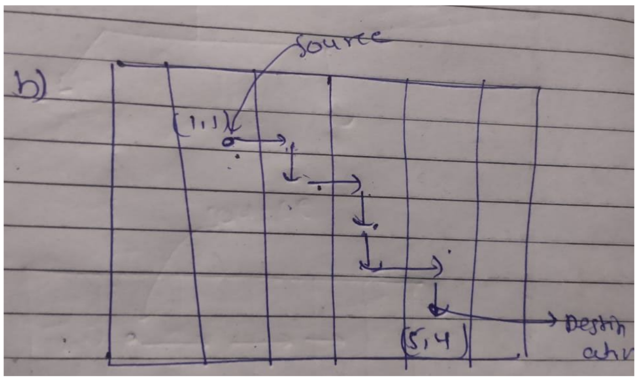
```
a)  myNextMove(cura,end){
  if(cura.X<end.X){
      return new (cura.X+1,cura.Y);
  }
   if(cura.X>end.X){
      return new P(cura.X-1,cura.Y);
   }
  if(cura.Y<end.Y){
      return new (cura.X,cura.Y+1);
   }
   if(cura.Y>end.Y){
      return new (cura.X,cura.Y-1);
   }
   return cura;
 }

}
```

this rules makes the move to Agent in the valid and available direction



```
.b) myNextMove( cura, end){
      if(dis(cura,end)>dis((cura.X+1,Y),end)){
         return new (cura.X+1,cura.Y);
      }
      if(dis(cura,end)>dis((cura.X-1,Y),end))){
         return new (cura.X-1,cura.Y);
      }
      if(dis(cura,end)>dis((cura.X,Y+1),end)){
         return new (cura.X,cura.Y+1);
      }
      if(dis(cura,end)>dis((cura.X,Y-1),end)){
         return new (cura.X,cura.Y-1);
      }
      return cura;
   }
```

this rules makes the move to Agent in the valid and towards the direction where distance from goal position going to be minimum .



## III. ALGORITHM AND ANALYSIS

```
isValid(point){
if(point.X<0||point.X>=end.X||point.Y<0||point.
Y>=end.Y) return false;
     return true;
  }
```

this part validate the move of Agent.

```
Analyze()
 {
  if(cura.Y==this.desa.Y   &&.cura.X==desa.X)
setAstate(true);
     else setAstate(false);
   }

  public Pair <Integer,Integer> AgentNext()
   {
     setCura(env.myNextMove(cura,desa));
     Analyze();
     return getCura();
   }
```

Analyse function analyse the state of Agent it reaches the goal destination or not.

## CONCLUSION

In this we make an environment and agent. The agent is moved by the actions left, right, up and down According to its sensor and given Environment.

## REFERENCES

[2]https://en.wikipedia.org/wiki/Intelligent_agent