Tutorial 1 - Report AI

Mohammed Aadil[#1]

Information Technology,
Indian Institute Of Information Technology,
Allahabad, India.
[1]iit2018179@iiita.ac.in

***Abstract*— Artificial intelligence (AI) is a relatively new branch of computer science. A tremendous amount of effort has been put into research associated with understanding systems and abstracting key principles of intelligent behaviour. In this paper we will look into a few such features.**

***Keywords*— (U D L R) - Up, Down, Left, Right**

I. Introduction

In this paper I will walk you through the basics of AI Assignment we were given. The whole concept revolves around the process of an/many Agent(s) finding its/its' way around an Environment, to get to a Single common destination.

II. Context

There are 3 classes that I will be referring to throughout this paper. These classes containing the following features and methods besides the getters and setters :

I. Agent Class :
   - A. Variables:
     1. name (string)
     2. path(ArrayList)
   - B. Methods:
     1. Sensor()
     2. Travel()

II. Environment Class :
   - A. Variables:
     1. location (HashMap)
     2. destination (Pair)
   - B. Methods:
     1. BoundCheck()

III. Main Class : This class runs the GUI and the other classes

III. Part A

In this part we were supposed to spawn only 1 Agent and this agent is supposed to find its way to the destination, and in this part the agent has full knowledge of his position and the destination position.

This is crucial for the logic. As the Agent knows where he has to go exactly, we can use a slope method to decide the best course of steps to take to get to the destination in the least number of steps.

A. Algorithm :

The idea behind this approach is, as we know the destination and the current location coordinates.
We will calculate the slope between these 2 points and according to the slope of the line we will choose between 4 options, (U, D, L, R).

Let's assume the agent is currently at (0,0) and the destination is at (6,3). So the slope is 0.5, now this can be visually shown as the following (Fig 1).

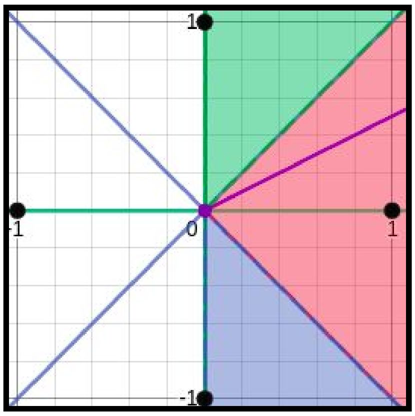

Fig 1

As we can see there are 3 main options the Agent has (UDR) but according to the slope the line lies in the RED region that gives a unanimous decision of taking a right.

The Agent will happily follow this rule base till he reaches the destination, and there will be no corner cases.

IV. Part B

In this part the only change is that the Agent is unaware of the location of the destination point and also it will never get to know his current coordinates.

The sensor function that is in the Agent class will simply return a double value that is the distance from the agent to the destination.

B. Algorithm :

The thought process in this part was that if we don't have the coordinates and are now limited to the distance between the 2 points. We can make 4 if conditions to go (U D L R) but this would be very inefficient as it has to go through weak if statements.

So the next best thing is to choose the 4 points randomly. After selecting a point at random we make the move to that point and then call the sensor function to get the new distance if this distance is less than the initial one, we consider this movement to be valid, else we select another random point.

C. Optimization :

If you follow this algorithm for a while, you will come to the conclusion that this approach is also quite in-efficient. As the moves are random and the same move could come up multiple times (ie. if the 1st move was up and up was a bad move the random number generator may generate the same move again …)

A simple fix to this would be if the random number is generated before just skip it and choose the other random ones. Although this is a small change it reduces the number of steps to half the original steps.

V. Part C

In this part there are quite a bit of changes as we now need to accommodate not just one Agent but 10 Agents and we should make sure that the agents don't collide with each other. This adds a flavour of multi-threading in these 10 agents.

Then all of this should come together and be displayed on a GUI. So for this purpose alone I will be using an ArrayList 'path' that will be in all the Agents, to store the path they took.

D. Algorithm :

So for this part the ArrayList plays a crucial part in deciding if the agents collide or not. I am considering a perimeter of 1 unit in the possible directions that the agent can take.

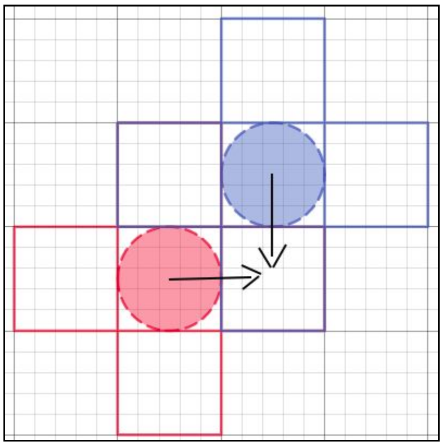

Fig 2.

Now we come to the challenging part of this assignment: how to avoid collision between the agents.

As shown in Fig 2, the red agent chooses to go right whereas the blue agent decides to go down. We can see that this will lead into a collision. So how do we avoid this from happening. A simple approach would be to add a perimeter, if another agent tries to move into the perimeter of a different agent it will not move there and it will act as a border for the agent. making the agent halt for 1 unit of time.

We repeat this till all the agents reach the common destination. As they reach the destination they are removed from the screen.

VI. Conclusions

Over all this assignment/project revolved around the basics of Agent and Environment setup in a generic AI model. Also how multiple Agents interact with each other and maintain their constraints.

Acknowledgment

I would like to acknowledge Prof. Rahul Kala for his undivided attention to the projects, and his distribution of knowledge at all levels.