LP1-AIR Mini project

Problem Statement:-

To find a way out of the given maze using the starting and ending point using A* algorithm .

Team members:-

41250 - Sahil Patil
41254 - Pralhad Kulkarni
41258 - Srushti Raybhoge

Abstract:-

A * algorithm is a searching algorithm that searches for the shortest path between the *initial and the final state.* It is used in various applications, such as *maps*.

In *maps* the A* algorithm is used to calculate the shortest distance between the source (initial state) and the destination (final state).In *maps* the A* algorithm is used to calculate the shortest distance between the source (initial state) and the destination (final state).

H/W & S/W requirements:-

Python , Jupyter notebook.

Introduction:-

A* algorithm has 3 parameters:

- **g :** the cost of moving from the initial cell to the current cell. Basically, it is the sum of all the cells that have been visited since leaving the first cell.
- **h :** also known as the *heuristic value,* it is the **estimated** cost of moving from the current cell to the final cell. The actual cost cannot be calculated until the final cell is reached. Hence, h is the estimated cost. We **must** make sure that there is **never** an over estimation of the cost.
- **f :** it is the sum of g and h. So, **f = g + h**

The way that the algorithm makes its decisions is by taking the f-value into account. The algorithm selects the *smallest f-valued cell* and moves to that cell. This process continues until the algorithm reaches its goal cell.

Objective:-

Objective of this project to use appropriate smartf algorithms to find a way out of the maze . Other algorithms like Djikstras can be used but they are not smart enough . This is because they mostly decide the next step using the current state of the problem . In A * algorithm the distance from initial as well as distance from final state is considered while deriving the next step.

Scope:-

In the given problem a text file with the original state of the maze . There are a total of 3 symbols here .

@ - Starting point
$ - Ending Point
# - The boundaries of the maze
. - The open path

In the output we have an extra symbol '+' which defines the path taken by the algorithm to reach the end state from the starting state.The algorithm uses distances to find the difference between current state and final state as well as starting state.

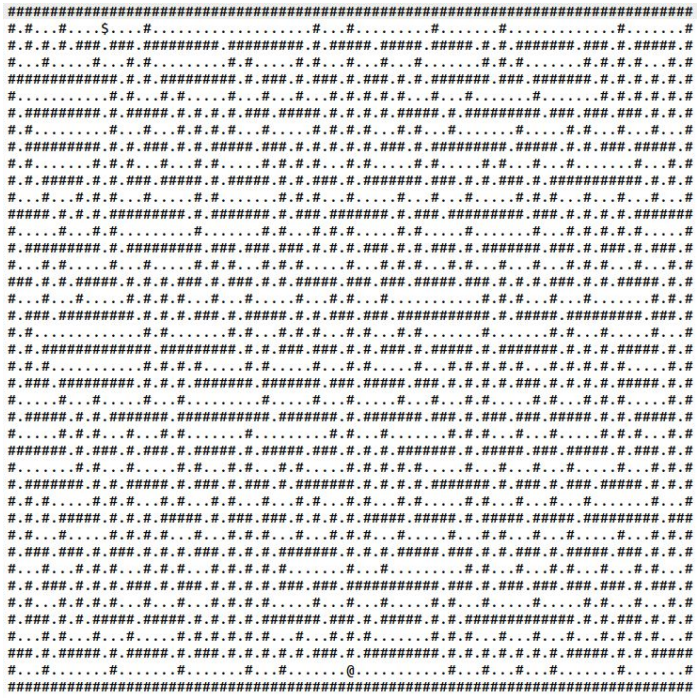
Algorithm/System Architecture:-

1. Initialize the open list
2. Initialize the closed list
   put the starting node on the open
   list (you can leave its **f** at zero)
3. while the open list is not empty
   a) find the node with the least **f** on
   the open list, call it "q"
   b) pop q off the open list

c) generate q's 8 successors and set their
parents to q
d) for each successor
i) if successor is the goal, stop search
successor.**g** = q.**g** + distance between
successor and q
successor.**h** = distance from goal to
successor (This can be done using many
ways, we will discuss three heuristics-
Manhattan, Diagonal and Euclidean
Heuristics
successor.**f** = successor.**g** + successor.**h**
ii) if a node with the same position as
successor is in the OPEN list which has a
lower **f** than successor, skip this successor
iii) if a node with the same position as
successor is in the CLOSED list which has
a lower **f** than successor, skip this successor
otherwise, add the node to the open list
end (for loop)
e) push q on the closed list
end (while loop)

Test Cases:-

Input:-



Output :-

```
Steps to goal: 340
```

Results:-

Appropriate path is found out if any by using A * algorithm . The path is marked with ‘+’ sign.

Conclusion:-

Thus we studied A* algorithm and used it to find out a path through a given maze along with the initial point and finishing point.