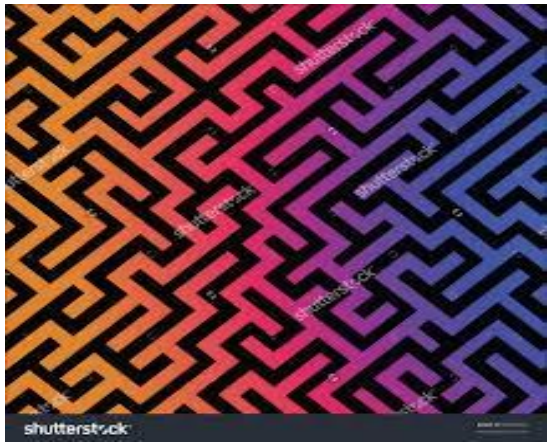
IMPLEMENTING MAZE USING DIFFERENT AI ALGORITHMS



BY

Jeevana Kala Bommannagari

Sri Ramya Nimmagadda

MAZE

- A maze is a path or collection of paths typically from an entrance to a goal.
- It defines the state space has to solve in a variety of ways.

RULES OF THE PROBLEM

1. This game can be defined in a finite space so that a space is used for the main board.
2. we can consider the N * N array of boards.
3. The array of rows and columns to rows and columns around the board to be considered.
4. The board can only have a unique path from the beginning to the end boards.
5. Just move forward and down is allowed.
6. A starting point has to be considered.
7. A point-to-end path is considered.
8. In each case, only one of two moves are allowed.
9. The output rows and columns are allowed to be displayed in the path traveled.

PATH FINDING SEARCH ALGORITHMS

- A*
  - Informed search algorithm or a Breadth-first search
  - It solves problems by searching among all possible paths to the goal for the one that incurs the smallest cost and among these paths it first considers the ones that appear to lead most quickly to the solution.
  - Works better on mazes with non-uniform tiles and obstacles within the paths.

$$f(n) = g(n) + h(n)$$

A* Algorithm

- Requires knowing the exit point.
- It uses this data in its heuristic calculation.
  - **Manhattan Distance**

```
this.manhattan = function(pos0, pos1) {
var d1 = Math.abs(pos1.x - pos0.x);
var d2 = Math.abs(pos1.y - pos0.y);
return d1 + d2;
}
```

PATH FINDING SEARCH ALGORITHMS

- Tremaux
  - Depth First Search
  - Works better with the maze that is well-defined with narrow passages and on complex “classic” style mazes .
  - Tremaux does not require knowing the exit point when beginning the search through the maze and needs to know only when to stop

RESULTS

- A* search takes longer, exploring almost all of the map to locate a solution.
- Tremaux performs a more direct search to locate the solution.

THANK YOU