Solving Snake & Maze games using Artificial Intelligence

Ansh Bhatia(008), Bhanu Soni(013) and Dhruv Soni(016)

Department of CSE, IET, JK Lakshmipat University

Abstract

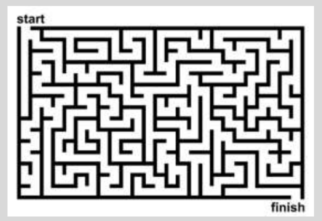
Snake game and **Mazes** have been really popular games of a time. The snake game has been one of the most popular game of its time while mazes had their charm in newspapers and magazines. The paper focuses on the implementation of the Artificial Intelligence concepts on certain games and using appropriate **algorithms** and **A*** attempts to solve the game.

In order to do the same, we decided to take two approaches into consideration, which are **Informed Search Techniques(A* algorithm)** and the other being **Uninformed Search Techniques(BFS, DFS Searching).** Based upon the requirement and conditions of the game we selected suitable algorithm and acted accordingly

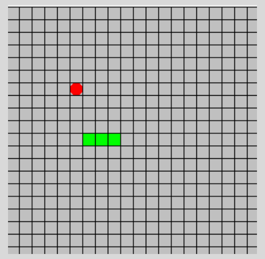
About the Games

The **Maze game** is basically moving from start point to the end or goal point, the walls are placed in form of puzzles that act as obstacles.

The **Snake game** works on the concept of playing the game until the snake dies, as the snake find and eats food, its lengths keeps on increasing. The snake dies if it tries to eat itself or comes in contact with the wall.



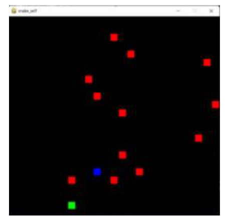
Maze Game



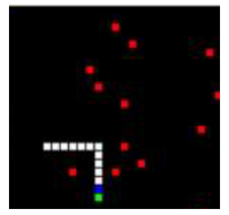
Snake Game

Analysis

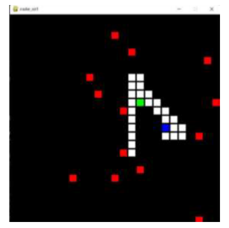
Snake Game



Initial State
The primary stage of game involves generation of a random solvable level.

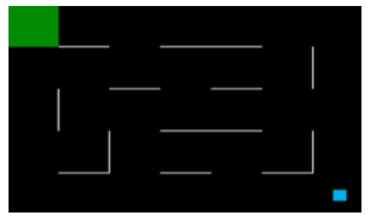


Middle State
The snake trying to consume as many fruits as it can without biting itself.

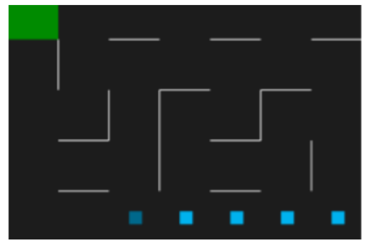


Final State
Snakes ending up biting itself and the game gets over.

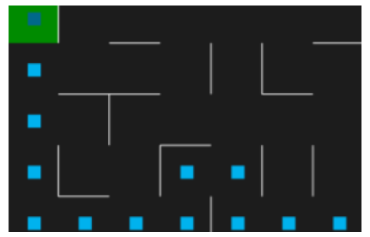
Maze Game



Initial State
Generation of maze games with the definite starting and ending point.



Middle State
Playing the game until the game is solved i.e. goal point is reached.



Final State
Reaching the goal position and hence solving the game.

Algorithms

Breadth First Search
This algorithm uses **FIFO** (First In First Out) queue. This involves visiting nodes in the order of their distance from the source node, where the distance is measured as the number of traversed edges.

BFS falls under the category of **Uninformed Search Technique**.

Depth First Search
DFS works on the LIFO (Last In First Out) and is a recursive algorithm. Basically, the algorithm gets deeper into search space, whenever possible. Even with its simple approach. DFS requires large computing power

DFS falls under the category of **Uninformed Search Technique**.

A* Algorithm
A* algorithm uses eucledian distance as heuristic value in case of path finding. Although, A* does give up on the optimality of the solution, it gives the much more efficient results

A* algorithm falls under the category of **Informed Search Technique**.

Discussion

The general crux of algorithms that involve **pathfinding** is just a small piece of puzzle in AI games. **A* algorithm** is one of the most popular and **efficient** path-finding algorithm. In this project, we have implemented A* algorithm to find the **shortest path** between the source and the destination

.The algorithm is tested by applying it on multiple maps of the snake and maze games. **Pathfinding** is a **crucial part** of various **strategy games** and other applications.

We studies across number of maps and scenarios of both snake and maze game. Overall, the system or **algorithm's performance** was **acceptable** and can be regarded as **efficient** to find the designated solution.

References

1. AAAI-21 Student Papers and Demostrations **https://bit.ly/3y0fKc1**
2. A* Algorithm **https://bit.ly/3rKb8FY**
3. Path-finding in Strategy Game and Maze Solving A* Search Algorithm **https://bit.ly/3ygvQOZ**
4. Solving the Classic Snake Game using AI **https://bit.ly/3pSBK55**