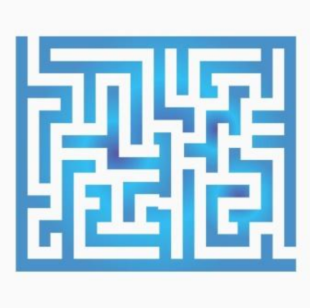
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

A PROJECT REPORT

ON

MAZE PATHFINDING VISUALIZATION



SUBMITTED BY :

1.Anika Tasnim - C223294

2. Afroja Akter Ruma - C223290

3. Hujaifa Tarannum – C223278

SECTION : 6CF

SEMESTER : 6TH

DATE OF SUBMISSION : 9 JULY,2025

SUBMITTED TO :

Sara Karim

Adjunct Faculty

Dept of CSE , IIUC

Maze Pathfinding Visualization

1. Objective

This project demonstrates the implementation of the A* (A-Star) algorithm for pathfinding in a maze. The goal is to find the shortest and most efficient path from the starting point to the goal in a 2D grid maze. The maze is visualized in a graphical format using Python.

2. Why This Project?

2.(a)To understand how A* works in real-time applications like robotics and game AI.
2.(b)To visualize the decision-making of the algorithm at every step.
2.(c)To gain hands-on experience with search algorithms and Python GUI.

3. Tools & Technologies Used

1.Frontend (Visualization Layer)-

3.1.(a) Python Tkinter (via pyamaze library): Used for rendering the maze grid and agent movement.
3.1.(b)Color-coded agents: Different colored agents represent search path, parent mapping, and final path.

2.Backend (Logic Layer)-

3.2.(a)Python 3.x
3.2.(b) A* Algorithm: Uses a priority queue to explore paths with the lowest cost.
3.2.(c)Heuristic: Manhattan Distance to estimate the cost between two cells.
3.2.(d) Maze Data: Loaded from astardemo.csv file, which contains the maze layout.

4. Project Features

4.(a)Maze Generation & Loading: A predefined maze layout is loaded from a CSV file.
4.(b)A* Implementation:

1. Calculates g (cost from start) and h (heuristic to goal).
2. Uses a priority queue for node selection.
3. Builds a path by tracking the best parent nodes.

4.(c)Multiple Agents:

1. Blue: Search path.
2. Yellow: Parent path.
3. Red: Final optimal path.

4.(d)Live Visualization: The steps of the algorithm are shown in real-time on the maze grid.
4.(e) Path Statistics: Displayed using an on-screen text label.

5. Screenshots & Visual Demonstrations

Figure 1: Maze Creation Interface

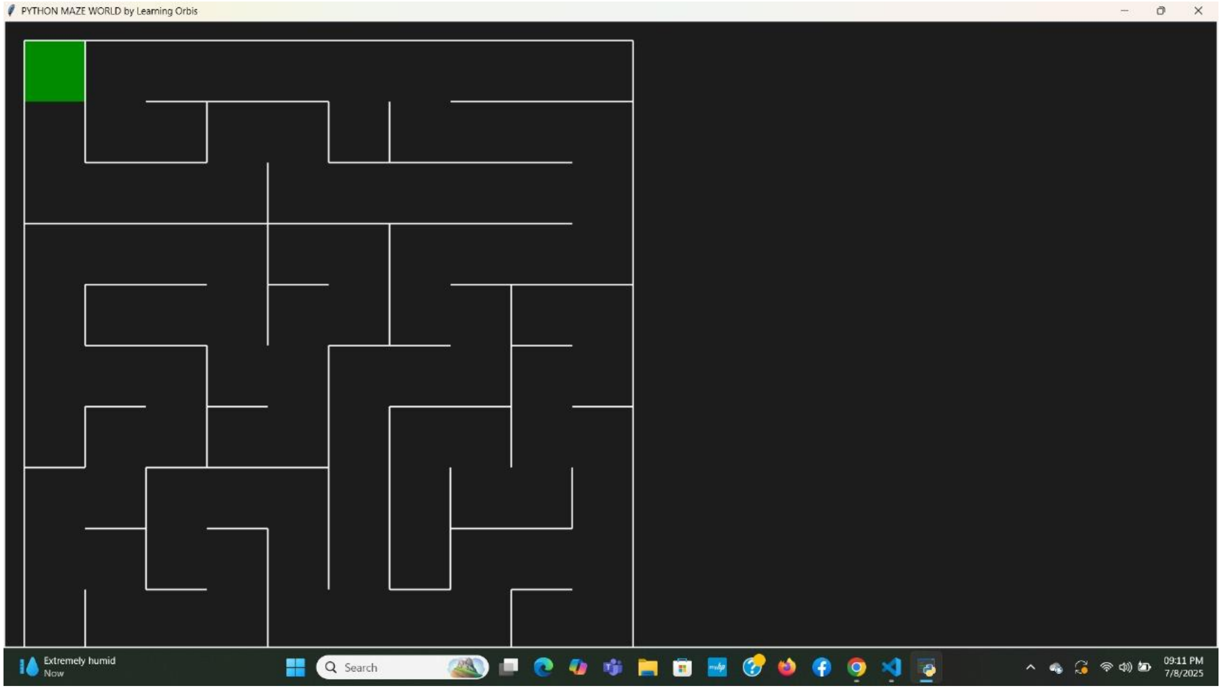
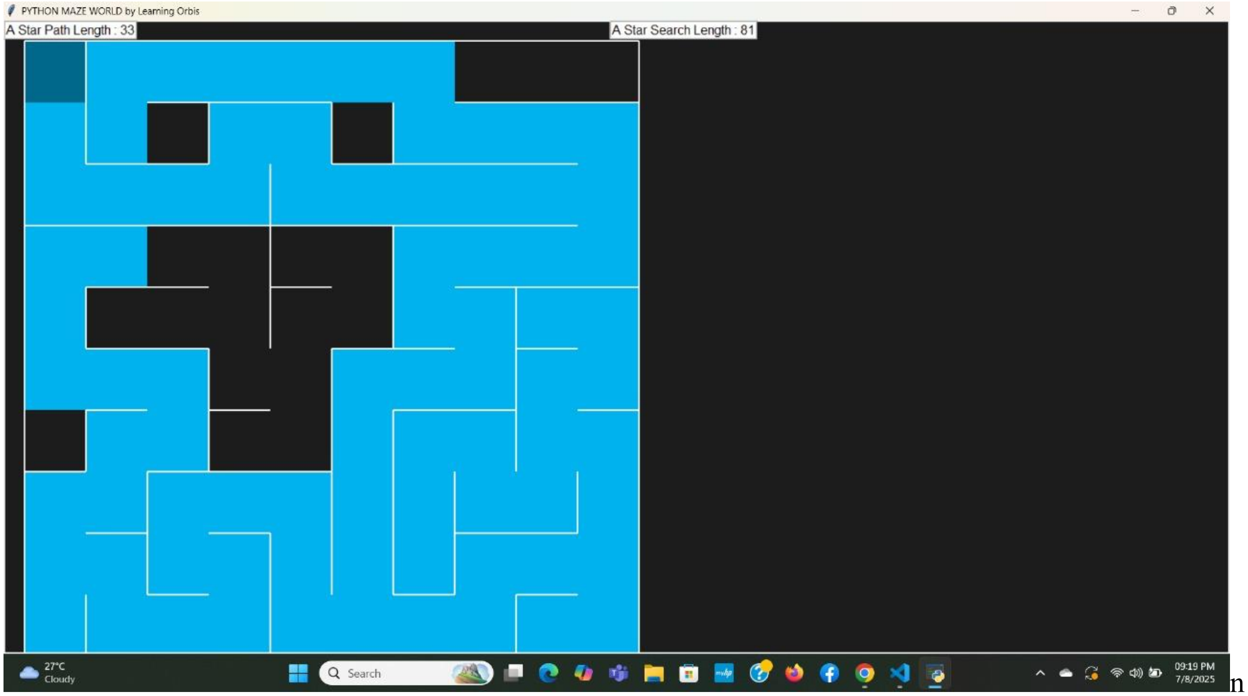


Figure 2: A* Pathfinding Visualizatio



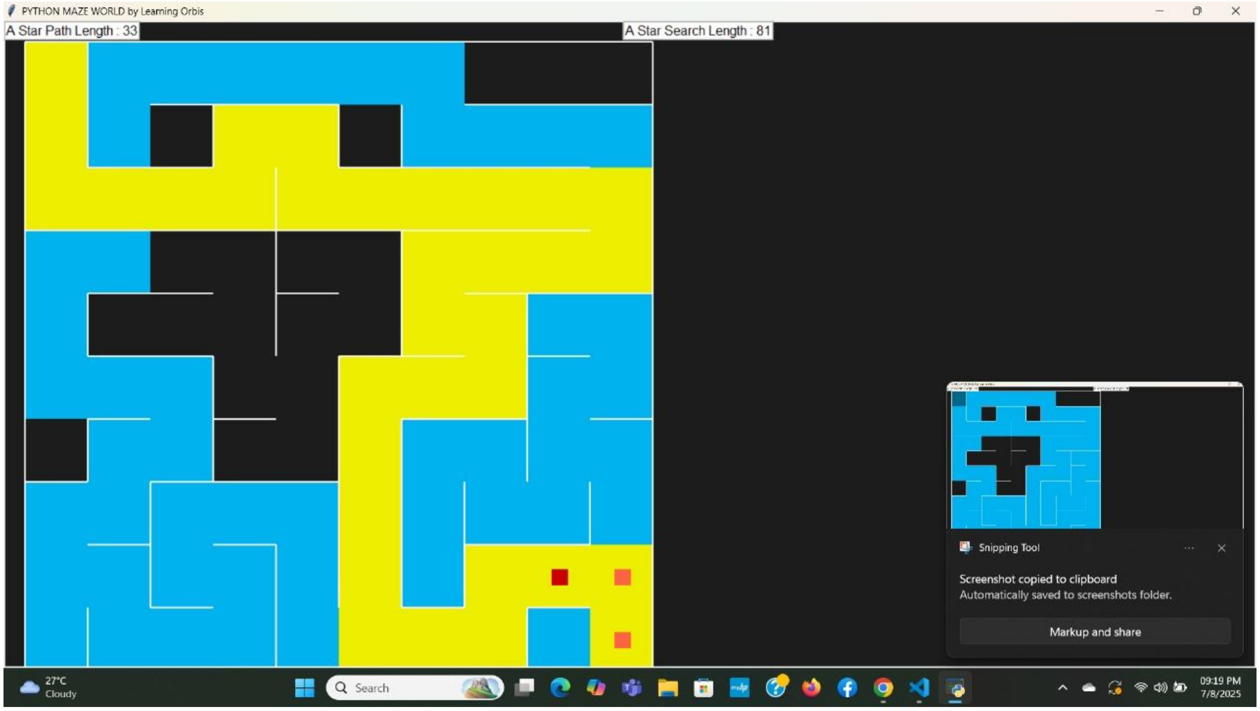
PYTHON MAZE WORLD by Learning Orbis

A Star Path Length : 33

A Star Search Length : 81

n

Figure 3: Backtracking the Shortest Path



PYTHON MAZE WORLD by Learning Orbis

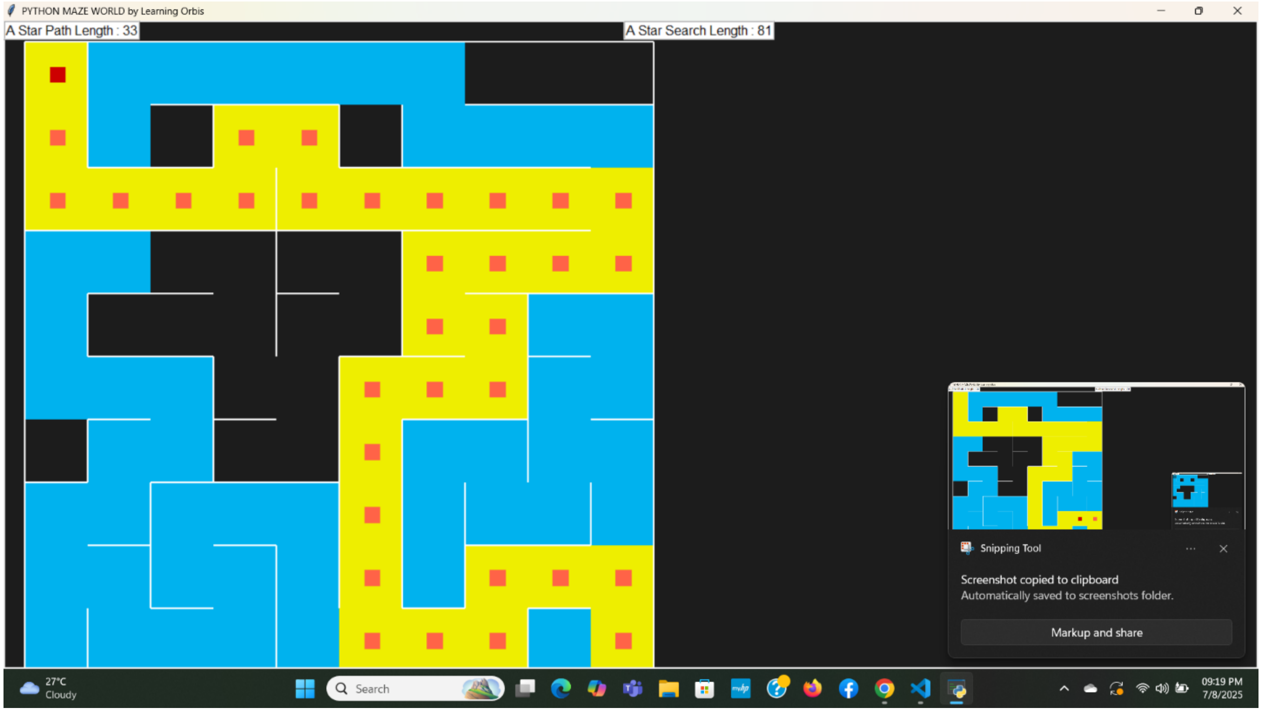
A Star Path Length : 33

A Star Search Length : 81

Snipping Tool

Screenshot copied to clipboard
Automatically saved to screenshots folder.

Figure 4: Final Result Showing Path Length and Search Count



A Star Path Length : 33

A Star Search Length : 81

Screenshot copied to clipboard
Automatically saved to screenshots folder.

6. Conclusion

This project serves as a great educational tool to understand the mechanics of the A* search algorithm in a visual and interactive way. It brings together algorithmic logic and GUI development, offering both backend challenge and frontend clarity.