University of Asia Pacific

Department of Computer Science & Engineering

Course Code: CSE 404

Course Title: Artificial Intelligence and Expert Systems Lab

Assignment No: 02

Submitted to

Dr. Nasima Begum

Associate Professor

Department of CSE

Submitted by

Name: Anjina Islam Anni

ID: 21101010

Name: Raita Sikder Fareha

ID: 21101023

Name: Ayon Ghosh

ID: 21101032

Problem Title

Maze Navigation using A* Algorithm

Problem Description

Our task is to navigate a 10x10 maze grid by moving from a specified start point to a designated goal while avoiding obstacles. The maze configuration, including start, goal, and obstacles, is defined in an input file that shapes our strategy. Together, we aim to develop a solution that efficiently finds and displays a viable path to the goal or provides feedback if the path is blocked. This assignment strengthens our skills in collaborative problem-solving, pathfinding, and strategic thinking in a structured environment.

Tools and Languages Used

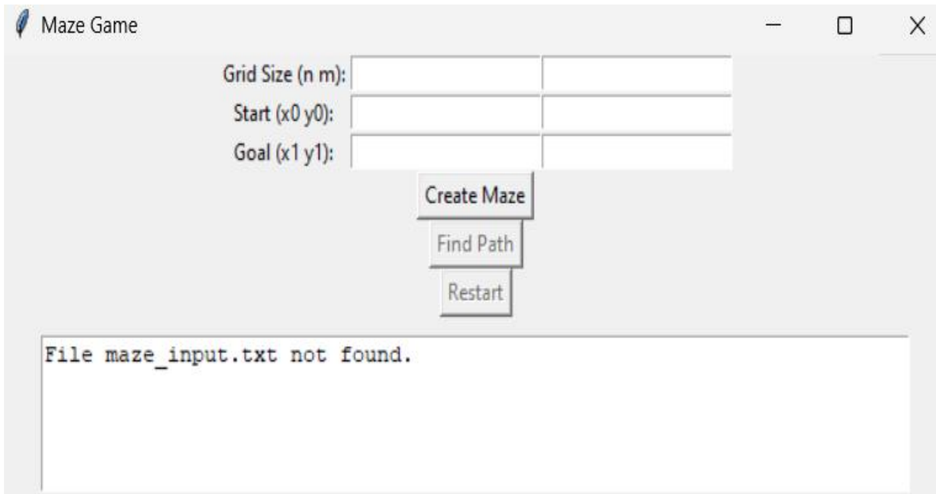
- **Language** – Python
- **IDEs** – VS Code
- **Visualization Libraries** –
  - tkinter: For GUI creation
  - math: For mathematical functions
  - heapq: For implementing the priority queue in A*
  - time: For creating delays in animations
  - os: For file handling (checking for file existence)

Sample Input/Output

Input

- **Maze Specifications:**
  - Grid size: (n, m)

- Start position: (x0, y0)
- Goal position: (x1, y1)
- Obstacles: Click to place on the grid or specify in a file.



- **Input File (maze_input.txt):**

  n m

  x0 y0

  x1 y1

  obstacle1_x obstacle1_y

  obstacle2_x obstacle2_y

Output

1. **Maze Visualization:** Displays the maze with start, goal, and obstacles.
2. **Cost Grid:** Shows g, h, and f values for each cell.

3. **Path Visualization:** Animates the path (if found) from start to goal in dark orange.

4. **Messages:** Provides feedback on pathfinding status and obstacles.

Challenges

- **Maze Configuration**: Ensuring the maze was configured correctly with a valid starting point, goal, and obstacles required careful planning. Incorrect placements could lead to no available path.

- **Heuristic Selection:** Choosing the right heuristic function is crucial for the efficiency of the A* algorithm. An inappropriate heuristic could lead to suboptimal performance or longer search times.

- **Handling Edge Cases:** Scenarios where the start and goal points are the same, or where the path is completely blocked, required additional logic to handle and provide appropriate feedback.

- **Performance Optimization:** As the size of the maze increases or becomes more complex, managing the open and closed sets efficiently is essential to ensure the algorithm runs in a reasonable time frame.

- **Path Reconstruction:** Implementing the logic to backtrack and reconstruct the path from the goal to the start point added complexity, particularly when ensuring no nodes are skipped.

- **Visualization:** If visualizing the algorithm's progress, creating an effective graphical representation that accurately reflects the maze state at each step could be challenging.

- **Debugging:** Identifying and fixing bugs in the pathfinding logic or maze configuration could be time-consuming, especially in a complex implementation.

- **Collaboration:** Working collaboratively on the assignment required clear communication and division of tasks, which can sometimes lead to misunderstandings or conflicts in approach.

Conclusion

In this maze assignment, we implemented the A* pathfinding algorithm to navigate a 10x10 grid from a specified starting point to a goal while avoiding obstacles. By defining essential components such as node structures and heuristic functions, we efficiently found the shortest path.

The A* algorithm's combination of cost analysis and heuristic estimation proved effective in minimizing evaluations. This project enhanced our collaborative problem-solving skills and deepened our understanding of algorithmic implementation. Overall, it solidified our knowledge of pathfinding algorithms and prepared us for future computational challenges.