**Project Title:**  
*3D Grid Pathfinding and Visualization Using A*\* *Algorithm*

**Name:** M. Ammar Riaz  
**Reg. No.:** 2023377  
**Course:** AI Lab Project (AI201L)

**Abstract/Problem Statement**

The primary goal of this project is to develop an interactive 3D pathfinding system that uses the A\* algorithm to determine the shortest path between two points in a grid populated with obstacles. The project addresses challenges in navigating 3D spaces with constraints such as limited movement options and the presence of obstacles. This simulation provides a foundation for applications in robotics, UAV navigation, and game development.

**Introduction**

**Black Box Running**

* **Inputs:**
  + Grid size (e.g., 10x10x10)
  + Number of obstacles
  + Start and end coordinates (e.g., (0,0,0) to (9,9,9))
* **Outputs:**
  + The shortest path between the start and end points (if one exists).
  + Visualization of the grid, including obstacles, path, and adjustable obstacle opacity.

**Scope**  
The system simulates a 3D environment and provides visual feedback. However, it is limited to:

* A cubic grid structure (NxNxN).
* Static obstacles that cannot be moved or dynamically added during execution.
* Basic visualization and cannot handle large-scale environments efficiently due to computational limits.

**Methodology**

**Tools and Technologies Used**

* **Programming Language:** Python
* **Libraries:**
  + matplotlib: For 3D visualization and slider interface.
  + queue.PriorityQueue: For managing open nodes in the A\* algorithm.
  + tkinter: For displaying messages to the user.

**Design Approach**

* **Algorithm:**
  + A\* algorithm, leveraging Manhattan distance as the heuristic to calculate the shortest path.
  + Efficient neighbor exploration with obstacle avoidance.
* **Structure:**
  + **Classes:**
    - Grid3D: Encapsulates the 3D grid's properties, obstacles, and methods for pathfinding and visualization.
  + **Functions:**
    - generate\_obstacles: Randomly places obstacles while avoiding start and end points.
    - a\_star\_search: Implements the A\* algorithm to find the shortest path.
    - visualize: Visualizes the grid, obstacles, and path using a 3D plot with adjustable obstacle opacity.
  + **Data Structures:**
    - Priority Queue for managing the A\* algorithm’s open set.
    - Dictionaries for tracking scores (g\_score, f\_score) and reconstructing paths.

**Results**

**Findings**

* Successfully implemented a 3D grid with random obstacles and the ability to specify start and end points.
* A\* algorithm efficiently calculates the shortest path between the two points, providing meaningful results even with varying grid sizes and obstacle densities.
* Visualization offers an intuitive representation of the 3D grid and pathfinding process.

**Data Representation**

* The grid and path are visualized in 3D, where:
  + **Green:** Start point
  + **Red:** End point
  + **Gray:** Obstacles
  + **Blue Line:** Path from start to end
* An opacity slider dynamically adjusts the visibility of obstacles for better clarity.

**Example Path Visualization**

* **Grid Size:** 10x10x10
* **Obstacles:** 200
* **Path Length:** 17 steps (shortest path in the example scenario).

**Conclusion**

This project demonstrates the practical application of the A\* algorithm for navigating 3D grids. It offers a platform for exploring pathfinding techniques in constrained environments. Future improvements could include dynamic obstacle addition, performance optimization for larger grids, and support for more complex movement patterns.