Project Proposal

Pathfinding in Mazes using A* and BFS Algorithm

Project Description

The core of this project is to implement a framework for solving the **maze pathfinding problem** using two fundamental search algorithms: **Breadth-First Search (BFS)** and **A***.

We will design a maze environment and apply both algorithms to find a path from a given start point to a goal (exit).

The project requires a comparative analysis of the implemented algorithms, focusing on:

- **Time Complexity** (execution speed).
- **Space Complexity** (memory usage).
- **Solution Optimality** (whether the path found is the shortest).
- **Path Cost** (total steps or distance).

The final submission will consist of:

- Maze generator or loader
- Visualization of maze with solution path
- Implementation of all algorithms
- Performance comparison
- Analysis of heuristic effectiveness

Problem Domain

The chosen problem is **Maze Pathfinding**, which involves finding a valid path from a start position to a goal within a maze full of possible paths and dead ends.

The maze will be represented as a **grid (matrix)** where each cell is either:

- **Walkable (path)**
- **Blocked (wall)**

The algorithms will be applied to this representation to determine the path.

Algorithms to be Implemented

1. **Breadth-First Search (BFS)**
   a. Explores the maze level by level.
   b. Guarantees find the shortest path in terms of steps if all edges have equal cost.
2. **A* Search Algorithm**
   a. Uses a heuristic (e.g., Manhattan distance) to guide the search.
   b. More efficient than BFS in large mazes.
   c. Guarantees the optimal path if the heuristic is admissible.

Expected Outcomes

- A working implementation of BFS and A* applied to maze pathfinding.
- Comparative analysis shows differences in performance and efficiency.
- Visualization of the maze and the paths found by each algorithm.
-

Deliverables

- Source Code.
- Written Report (implementation + analysis).
- Demonstration (maze visualization with highlighted paths)