Maze Solver - Project Proposal

Group Members:

- Syed Sharjeel Ahmad (22K-4646)
- Muhammad Abdullah Shariq (22K-4497)
- Muhammad Taha (22K-4458)

1. Introduction

The Maze Solver project will implement a dynamic maze-solving environment where an agent must navigate through various maze configurations to reach a goal state. Our implementation will feature the following unique elements:

- **Dynamic Maze Generation**: Unlike standard mazes, our system will generate randomized mazes with varying complexity levels.
- **Multiple Agent Types**: The system will support different agents implementing various AI search algorithms.
- **Obstacle Integration**: The maze will include moving obstacles that the agent must avoid, adding complexity to the pathfinding problem.
- **Resource Collection**: Agents can collect resources (keys, power-ups) that modify their capabilities or score.
- **Time Constraints**: Certain paths may only be available for limited periods, requiring strategic planning.

2. Implementation Strategy

Our maze solver will implement and compare multiple AI algorithms:

Uninformed Search Algorithms:

- **Breadth-First Search (BFS)**: For finding the shortest path in terms of steps.
- **Depth-First Search (DFS)**: For exploring maze branches deeply before backtracking.

Informed Search Algorithms:

- *A Search**: Using Manhattan distance heuristic to find optimal paths efficiently.
- **Greedy Best-First Search**: Using direct distance to goal as the heuristic.

Implementation Details:

- The maze will be represented as a grid where cells can be paths, walls, or special elements.
- We will implement state representation for each agent position, with transitions representing valid moves.
- Heuristic functions will calculate estimated distances to the goal state.
- Performance metrics will track solution steps, time efficiency, and memory usage.

3. Deliverables

Goal State:

- The agent successfully reached the exit point of the maze.

State Representation:

- Each state will consist of:
    - Current agent position (x,y coordinates)
    - Collected resources (keys, power-ups)
    - Remaining time (for time-constrained scenarios)
    - Visited the cells history

Scoring System:

- Points awarded based on:
    - Time to completion
    - Number of moves taken
    - Resources collected
    - Obstacles avoided
    - Efficiency compared to optimal path

Lifelines:

- Limited "reveal path" hints showing optimal paths for a few steps
- "Freeze obstacle" ability to temporarily stop moving obstacles
- "Wall break" ability to remove a single wall section

User Interface:

- Visual representation of the maze with agent movement
- Algorithm selection options
- Performance statistics display
- Step-by-step solution visualization