Domonic Lee

20240629

CS352

UNIT-4 Responses

State of the Project

The project involving the `CaveExplorer` class is currently functional, capable of simulating cave exploration within a 2D grid. It initializes with either a default cave layout or user-specified configuration, employs a breadth-first search algorithm in `solve()` to navigate from 'S' to 'M', and accurately returns the path taken with `getPath()`. However, it assumes a single solution path and may need refinement for complex scenarios with multiple obstacles.

Testing the Program

Testing involved two main scenarios using hardcoded cave layouts. The first tested backtracking capabilities, while the second evaluated handling scenarios without a valid path from 'S' to 'M'. These tests validated functionality and highlighted areas for improvement in managing complex cave configurations and optimizing pathfinding algorithms.

Lessons Learned

This project provided insights into applying data structures and algorithms practically, emphasizing efficient state management and backtracking strategies. It reinforced algorithm design principles and Java's object-oriented features in solving pathfinding problems.

Liked About This Project

Designing algorithms for real-world problem-solving and refining pathfinding strategies were enjoyable aspects. The project facilitated the exploration of different approaches and optimization techniques tailored to specific scenarios.

Challenges and Areas for Improvement

Managing edge cases and ensuring robust handling of diverse cave layouts posed challenges. Enhancing clarity on managing multiple solution paths efficiently would improve versatility and usability.

Future Enhancements

Given more time, prioritizing advanced pathfinding algorithms like A\* search and developing a user-friendly interface would optimize performance and enhance user experience. These enhancements would refine functionality, improve usability, and visualize cave exploration processes effectively.