The project implements a CaveExplorer class that finds a path to the mirror pool in a cave represented by a 2D grid of characters. The class has two constructors, one without parameters that hardcodes a specific cave layout, and one that takes a filename and reads the cave layout from a file. The solve method returns a boolean indicating whether a path was found, and the getPath method returns a string representing the path taken. The toString method returns a string representation of the cave layout. The project works as expected.

Testing:

I tested the program using the hardcoded layout in the zero-parameter constructor and a separate layout stored in the cave\_layout.txt file. The tests included layouts with and without paths to the mirror pool. These tests helped verify that the implemented logic works correctly and can handle different scenarios.

What I learned:

I learned how to implement a backtracking algorithm to find a path in a 2D grid and how to read a cave layout from a file.

What I liked:

I liked that this project involved problem-solving using a backtracking algorithm and working with file input.

What was confusing or could be done differently:

The project could provide clearer instructions on the expected output format and how to handle cases where there's no path to the mirror pool.

If I had more time, I would add a feature to generate random cave layouts and ensure they have unique paths to the mirror pool. Additionally, I would consider optimizing the backtracking algorithm to improve its performance.