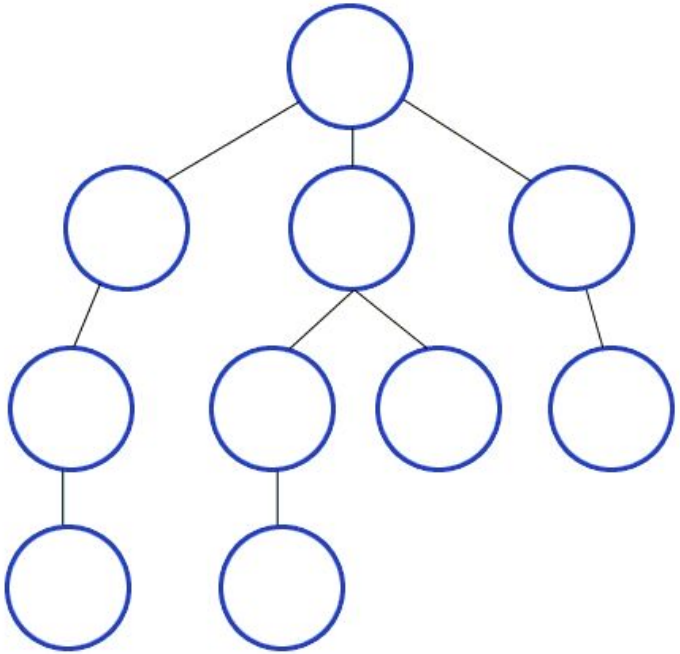
DFS Maze Solver

Franz-Aurel Huber

The Algorithm



https://commons.wikimedia.org/wiki/File:Depth-First-Search.gif

The Problem

- Solving a maze using the DFS Algorithm
- Showing the user the correct path
- Dynamic start and end detection

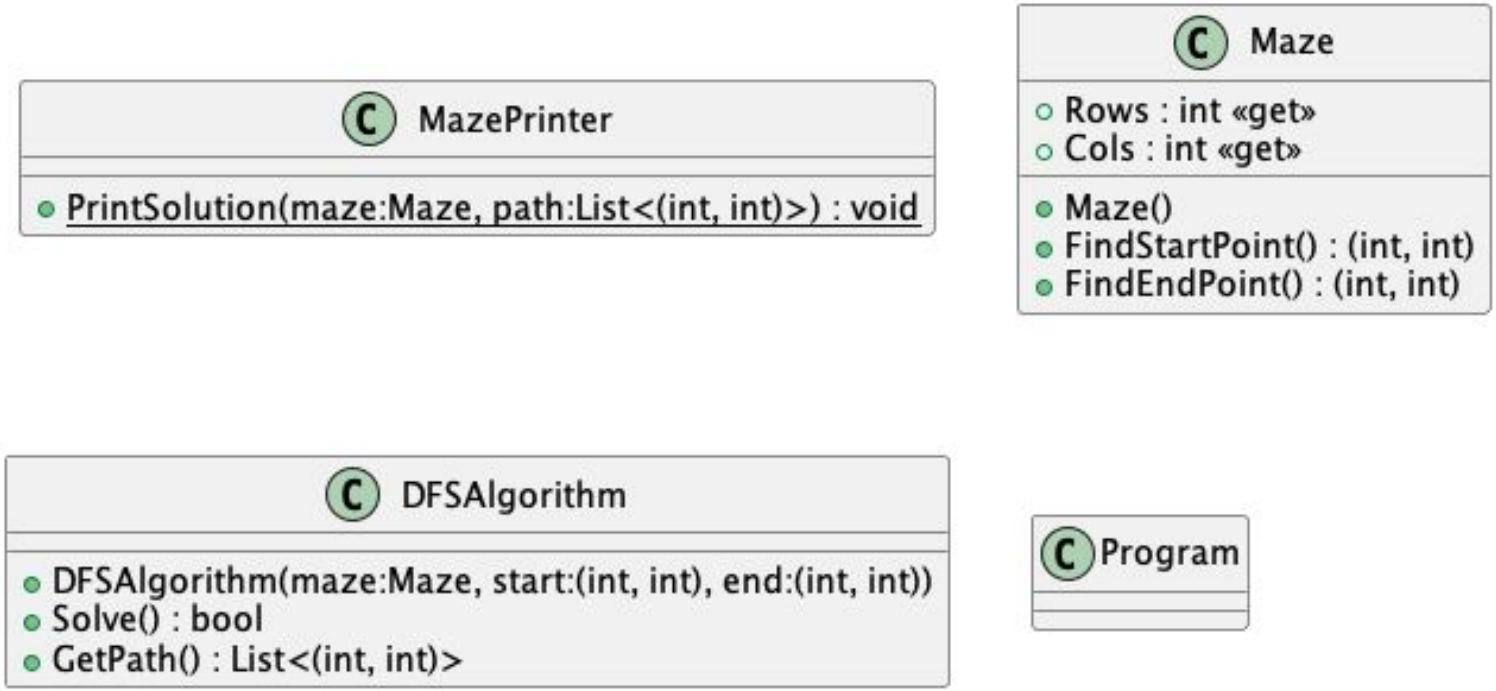
Implementation

- Maze has to be translated into a 2D-integer array (0=path, 1=wall)
- DFS is used to find a continuous path

  -> Dead ends are handled by backtracking. All paths are explored recursively until a solution is found

- The correct maze path gets output to the console. This path is highlighted in green and changed from 0 to 2

Class Diagram



Challenges

- Understanding the DFS Algorithm
- The dynamic start and end detection
- The path visualization strictly through the console

Reflection

- DFS was not too easy to understand
- Maze solving is a good use-case for the DFS Algorithm

Possible Improvements:

- Adding a GUI
- Supporting mazes as file imports
- Inheritance (BaseMaze class and let Maze inherit from it)