ARXR Project - Solving mazes

The basic idea of the project is to simulate the solving of a maze with different algorithms. The solving algorithms are a CA (Cellular Automaton), BFS (breadth first search) and DFS (depth first search). The time to solve the maze was taken for each algorithm and compared.

Maze generation  
To generate the mazes that should be solved later, the Kruskal algorithm was used. The Kruskal algorithm generates mazes that have no circles, no rooms/dungeons and it only has one path from start to the goal.  
To bring in some variation the “Prims” algorithm is also used. Prim mazes also have no circles, no rooms/dungeons and it only has one path from start to the goal.  
Both algorithms create perfect mazes, which means that every point is reachable and there is only one path a point to another point.

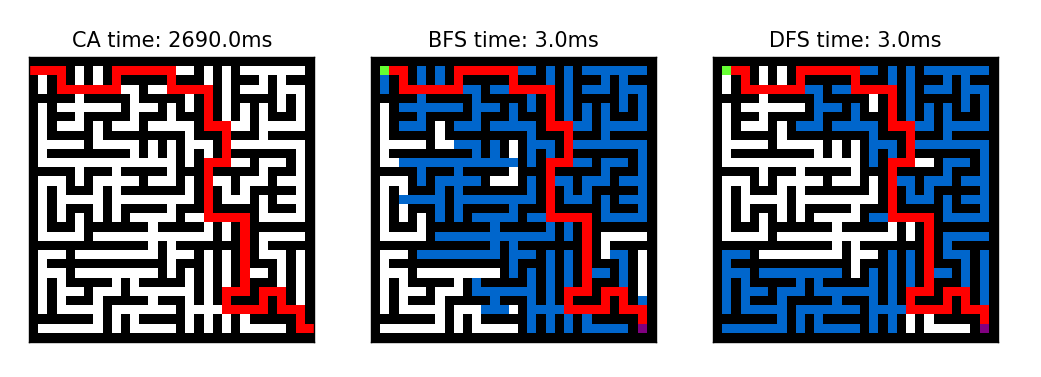


Figure 1: Depiction of the three algorithms with the path (red) found. Tiles that are visited by BFS/DFS are marked in blue.

Visualization  
To visualize how the algorithms solve the mazes, a GUI was created that shows each step the algorithm takes.  
Please note, that depiction of the mazes differs by algorithm. That’s because of the different ways the algorithms work. For example, the CA needs a “whole” in the wall at the start and the goal, whereas the BFS and DFS simply needs a marker where to start and end (green and purple). The final path the algorithm finds is marked in red. The tiles that are visited by DFS and BFS on the process of finding the best path are marked in blue.

Time measurement  
The time to solve the maze has been taken for every algorithm with different mazes ten times and then averaged.

|  |  |  |  |
| --- | --- | --- | --- |
|  | CA | BFS | DFS |
| Kruskal 30x30 |  |  |  |
| Prim 30x30 |  |  |  |
| Kruskal 100x100 |  |  |  |