Maze Solver Documentation

Assumptions

1. Maze Specification will always stay the same. Expected symbols are hardcoded. This can be improved by specifying a list of symbols to denote walls, points, source and destination cells as input respectively.

Summary

The MazeSolver uses a modified A* algorithm (based on an implementation from https://www.geeksforgeeks.org/a-search-algorithm/) to find the shortest path. The A* was modified to only look for NSEW neighbours, which is based on Von Neumann neighbour hood approach. The next successor that will be added to the path is determined using heuristics based on the Euclidean distance.

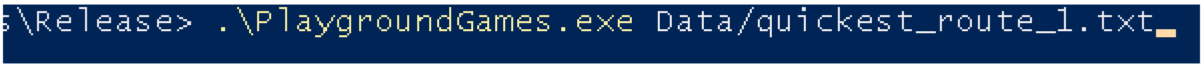
When tracing back the path, a stack was used to pop the next direction for the next cell. Cell struct was modified to contain a direction variable that was used to set the direction of the next successor from the current cell while solving the maze.

Testing

The code was tested by blocking the path to the destination, providing invalid input through command line and removing either start or end position within the file.

Executing Application

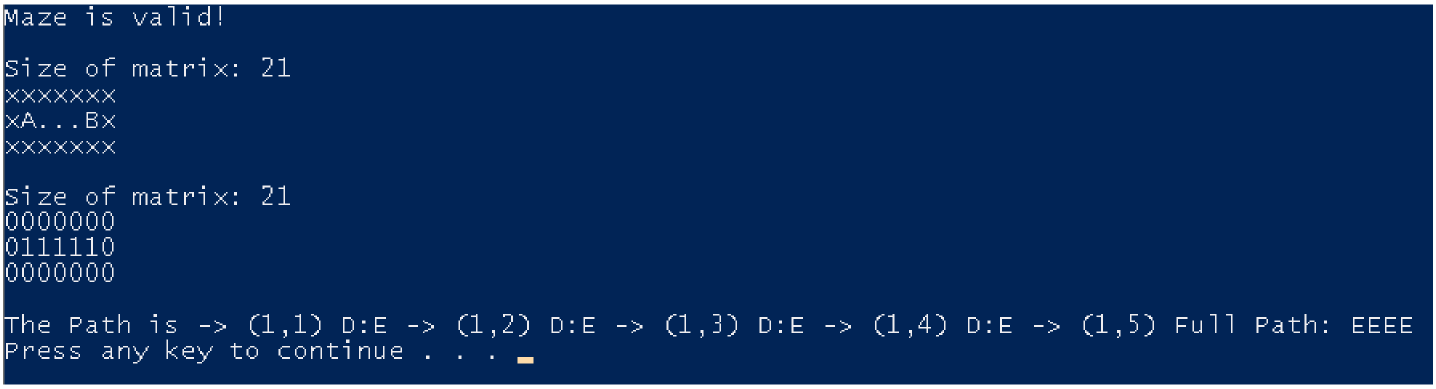
The application can be run either from command line or Powershell. The executable takes an argument as input file containing the maze.



```
s\Release> .\PlaygroundGames.exe Data/quickest_route_1.txt_
```

Output

Quickest_route_1



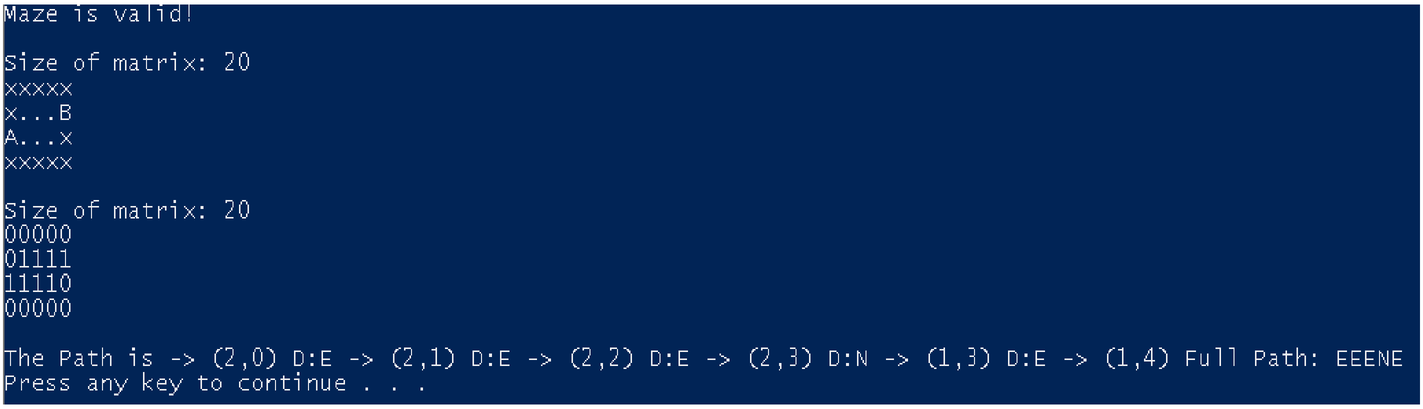
```
Maze is valid!

Size of matrix: 21
xxxxxxx
xA...Bx
xxxxxxx

Size of matrix: 21
0000000
0111110
0000000

The Path is -> (1,1) D:E -> (1,2) D:E -> (1,3) D:E -> (1,4) D:E -> (1,5) Full Path: EEEE
Press any key to continue . . .
```

Quickest_route_2



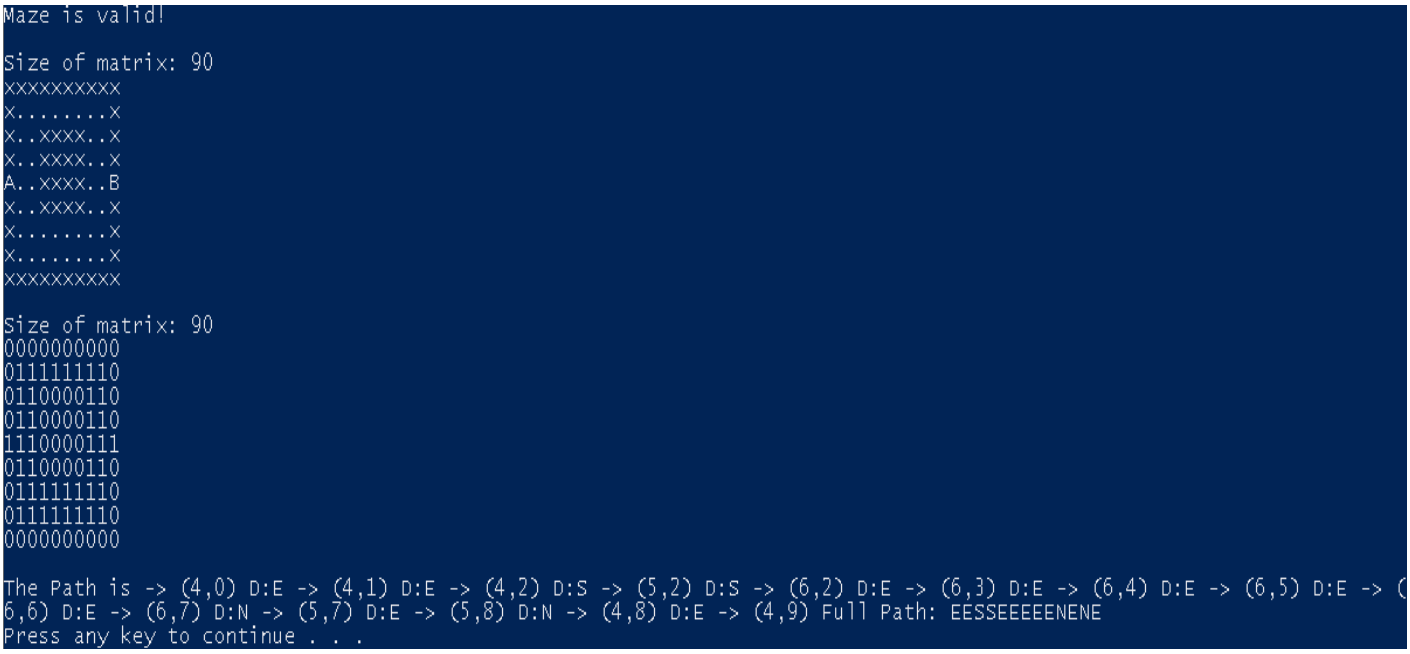
```
Maze is valid!

Size of matrix: 20
xxxxx
x...B
A...x
xxxxx

Size of matrix: 20
00000
01111
11110
00000

The Path is -> (2,0) D:E -> (2,1) D:E -> (2,2) D:E -> (2,3) D:N -> (1,3) D:E -> (1,4) Full Path: EEENE
Press any key to continue . . .
```

Quickest_route_3



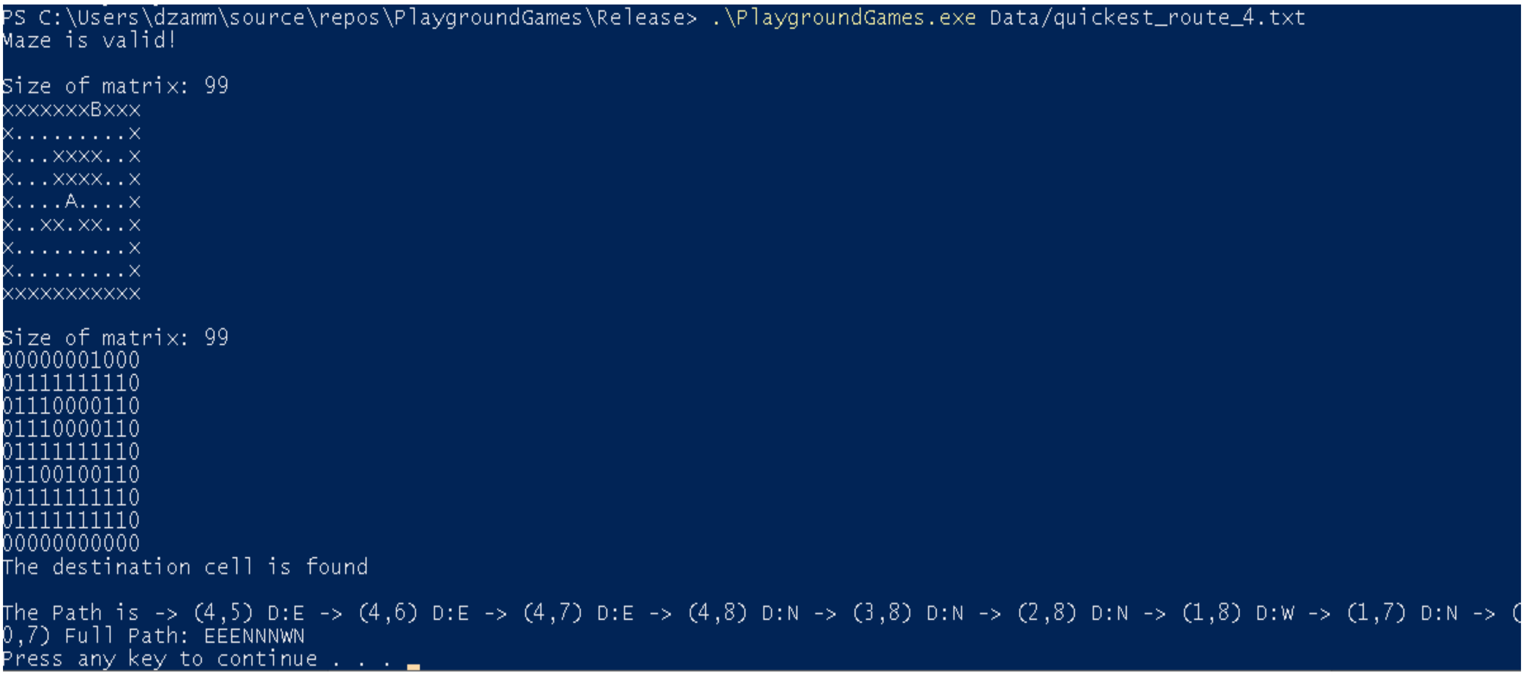
```
Maze is valid!

Size of matrix: 90
xxxxxxxxxx
x........x
x..xxxx..x
x..xxxx..x
A..xxxx..B
x..xxxx..x
x........x
x........x
xxxxxxxxxx

Size of matrix: 90
0000000000
0111111110
0110000110
0110000110
1110000111
0110000110
0111111110
0111111110
0000000000

The Path is -> (4,0) D:E -> (4,1) D:E -> (4,2) D:S -> (5,2) D:S -> (6,2) D:E -> (6,3) D:E -> (6,4) D:E -> (6,5) D:E -> (
6,6) D:E -> (6,7) D:N -> (5,7) D:E -> (5,8) D:N -> (4,8) D:E -> (4,9) Full Path: EESSEEEEENENE
Press any key to continue . . .
```

Quickest_route_4



```
PS C:\Users\dzamm\source\repos\PlaygroundGames\Release> .\PlaygroundGames.exe Data/quickest_route_4.txt
Maze is valid!

Size of matrix: 99
xxxxxxxBxxx
x.........x
x...xxxx..x
x...xxxx..x
x....A....x
x..xx.xx..x
x.........x
x.........x
xxxxxxxxxxx

Size of matrix: 99
00000001000
01111111110
01110000110
01110000110
01111111110
01100100110
01111111110
01111111110
00000000000
The destination cell is found

The Path is -> (4,5) D:E -> (4,6) D:E -> (4,7) D:E -> (4,8) D:N -> (3,8) D:N -> (2,8) D:N -> (1,8) D:W -> (1,7) D:N -> (
0,7) Full Path: EEENNNWN
Press any key to continue . . .
```