Homework 02 (Due: Friday, October 20, 2017, 11 : 59 : 00PM Central Time)

CSCE 322

THIS ASSIGNMENT IS ONLY WORTH 10% OF YOUR FINAL GRADE.

1 Instructions

In this assignment, you will be required to write JavaScript functions that simplify navigating an elaborate maze.

1.1 Data File Specification

An example of properly formatted file is shown in Figure 1. The first file encodes a maze, the second file encodes the directions inwhich to be moved.

part01test01.maze.emf

```
x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x
x,-,-,x,-,-,x,-,x,-,-,-,-,-,x,-,-,-,x,-,-,-,-,-,-,-,x,-,x
x,-,-,-,-,x,-,-,-,-,-,-,x,x,-,-,-,x,-,-,-,-,-,-,-,x,x,-,-,x
x,-,x,x,-,-,x,-,-,x,-,-,-,x,x,x,x,x,x,-,x,-,x,-,-,x,x,-,x
x,-,x,x,x,-,-,-,-,-,-,-,x,-,x,x,x,x,x,x,-,x,-,-,-,-,-,x,-,x
x,-,-,-,-,-,-,-,x,-,-,x,-,-,-,-,-,-,-,-,-,x,-,x,x,x,-,-,-,x
x,x,-,x,x,-,-,-,-,-,x,-,x,x,-,x,x,-,-,x,-,-,-,x,x,x,x,-,x,x
x,-,-,-,-,-,-,x,-,-,x,-,-,-,-,-,x,-,-,-,-,-,-,-,g,x,x,-,-,x
x,-,x,-,x,-,-,-,-,-,x,-,x,x,x,-,-,-,-,-,-,x,x,x,-,x,x,x,-,x
x,-,-,-,-,-,-,x,-,-,-,-,-,-,-,-,x,-,x,-,-,x,-,-,-,-,-,x,-,x
x,-,-,x,-,-,-,-,x,-,x,-,-,x,-,-,-,-,-,x,-,-,-,x,x,-,-,-,x
x,-,-,x,-,x,-,-,-,-,-,-,-,-,-,-,-,x,-,-,x,x,-,-,-,-,-,x,x
x,-,-,-,-,-,-,-,x,-,-,-,-,x,-,-,x,-,-,-,-,-,x,x,x,-,x,-,-,x
x,x,x,-,x,-,-,x,-,x,-,-,-,x,-,x,x,-,-,x,x,-,x,-,-,-,-,-,x
x,-,-,-,-,-,-,-,-,-,-,-,x,-,-,-,-,x,-,-,-,x,-,-,-,x,-,-,x,x
x,-,-,-,-,1,-,-,x,-,-,-,x,x,-,x,-,-,x,-,-,x,x,-,x,-,-,-,x
x,x,-,-,x,-,x,-,-,-,x,-,-,-,-,-,-,-,x,x,-,x,x,-,-,-,-,-,x
x,-,-,-,-,-,-,x,-,x,x,-,-,x,x,-,x,-,x,-,-,x,-,-,x,-,x,x,x
x,x,-,x,-,-,-,x,-,-,-,-,-,-,-,-,-,-,-,-,-,x,-,-,-,-,-,x,x,x
x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x
```

part01test01.moves.emf

```
l,r,l,d,r,r,l,u
```

Figure 1: A properly formatted encoding

2 One Player, One Move

The first part (`onePlayerOneMove` in the file `csce322homeWork02part01.js`) will take in one (1) argument (a maze) and return a function that takes in one (1) argument (a move), and returns the maze that is the result of moving Player 1 in the given direction.

The rules for moving are

- If a player is immediately blocked by a wall (`x`) or another player, they do not move
- A player continues to move in a given direction as long as "forward" and "backward" are their only options to move in

  "forward" is in the direction of travel and "backward" is opposite the direction of travel
- If a player encounters an obstacle (a wall or another player) while moving, they will attempt to change their direction (with priority given to up, down, left, and right in that order), but will not reverse their direction.
- A player will stop moving once they reach the goal (`g`), a dead-end where their only option is to reverse their direction, or when they reach a location where they have more options for travel than just "forward" and "backward".

```
x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x
x,-,-,x,-,-,x,-,x,-,-,-,-,-,x,-,-,-,x,-,-,-,-,-,-,x,-,x
x,-,-,-,-,x,-,-,-,-,-,x,x,-,-,-,x,-,-,-,-,-,-,-,x,x,-,-,x
x,-,x,x,-,-,x,-,-,x,-,-,-,x,x,x,x,x,x,-,x,-,x,-,-,x,x,-,x
x,-,x,x,x,-,-,-,-,-,-,-,x,-,x,x,x,x,x,x,-,x,-,-,-,-,-,x,-,x
x,-,-,-,-,-,-,-,x,-,-,x,-,-,-,-,-,-,-,-,-,-,-,x,-,x,x,x,-,-,-,x
x,x,-,x,x,-,-,-,-,-,x,-,x,x,-,x,x,-,-,x,-,-,-,x,x,x,x,-,x,x
x,-,-,-,-,-,-,x,-,-,x,-,-,-,-,-,-,x,-,-,-,-,-,-,-,g,x,x,-,-,x
x,-,x,-,x,-,-,-,-,-,x,-,x,x,x,-,-,-,-,-,-,-,x,x,x,-,x,x,x,-,x
x,-,-,-,-,-,-,x,-,-,-,-,-,-,-,-,x,-,x,-,-,x,-,-,-,-,-,x,-,x
x,-,-,x,-,-,-,-,-,x,-,x,-,-,x,-,-,-,-,-,-,x,-,-,-,x,x,-,-,-,x
x,-,-,x,-,x,-,-,-,-,-,-,-,-,-,-,-,-,x,-,-,x,x,-,-,-,-,-,-,x,x
x,-,-,-,-,-,-,-,x,-,-,-,-,x,-,-,x,-,-,-,-,-,x,x,x,-,x,-,-,x
x,x,x,-,x,-,-,x,-,x,-,-,-,x,-,x,x,-,-,x,x,-,x,-,-,-,-,-,-,x
x,-,-,-,-,-,-,-,-,-,-,-,x,-,-,-,-,x,-,-,-,x,-,-,-,x,-,-,x,x
x,-,-,-,-,1,-,-,x,-,-,-,x,x,-,x,-,-,x,-,-,x,x,-,x,-,-,-,x
x,x,-,-,x,-,x,-,-,-,x,-,-,-,-,-,-,-,x,x,-,x,x,-,-,-,-,-,x
x,-,-,-,-,-,-,x,-,x,x,-,-,x,x,-,x,-,x,-,-,x,-,-,x,-,x,x,x
x,x,-,x,-,-,-,x,-,-,-,-,-,-,-,-,-,-,-,-,x,-,-,-,-,-,x,x,x
x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x,x
```

```
l,r,l,d,r,r,l,u
```

Figure 2: Before `onePlayerOneMove`

```
xxxxxxxxxxxxxxxxxxxxxxxxxxx
x--x--x-x-----x---x-------x-x
x----x-----xx---x-------xx--x
x-xx--x--x---xxxxxx-x-x--xx-x
x-xxx------x-xxxxxx-x-----x-x
x------x--x---------x-xxx---x
xx-xx----x-xx-xx--x---xxxx-xx
x-----x--x-----x-------gxx--x
x-x-x----x-xxx------xxx-xxx-x
x-----x--------x-x--x-----x-x
x--x----x-x--x-----x---xx---x
x--x-x-----------x--xx-----xx
x------x----x--x-----xxx-x--x
xxx-x--x-x---x-xx--xx-x-----x
x----------x----x---x---x--xx
x---1---x---xx-x--x--xx-x---x
xx--x-x---x-------xx-xx-----x
x------x-xx--xx-x-x--x--x-xxx
xx-x---x-----------x-----xxx
xxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Figure 3: After `onePlayerOneMove`

3 One Player, Many Moves

The second part (`onePlayerManyMoves` in the file `csce322homeWork02part02.js`) will take in one (1) argument (a maze) and return a function that takes in one (1) argument (an array of moves), and returns the maze that is the result of Player 1 making all of the moves. If the maze is solved before the list of moves is completely processed, the resulting maze is returned.

```
x,x,x,x,x,x,x,x,x,x,x
x,-,x,-,-,-,-,-,x,-,x
x,-,-,-,-,x,-,-,x,-,x
x,-,-,-,-,-,-,-,-,-,x
x,x,x,-,-,x,-,x,x,-,x
x,-,-,-,-,-,-,x,-,-,x
x,x,-,-,-,-,-,-,x,-,x
x,-,x,-,x,-,-,g,-,-,x
x,-,x,-,-,-,x,x,-,x,x
x,-,-,-,x,-,-,x,-,-,x
x,-,x,-,x,x,-,x,x,-,x
x,-,x,x,-,-,-,-,-,-,x
x,-,-,-,-,-,x,-,x,-,x
x,-,x,-,x,-,-,-,-,x,x
x,-,-,-,x,-,-,x,-,x,x
x,-,-,x,x,x,-,x,-,-,x
x,-,-,-,-,-,-,-,1,-,x
x,-,-,x,-,-,-,x,-,-,x
x,-,-,-,-,-,x,-,-,-,x,x
x,-,x,x,-,-,x,-,-,-,x
x,-,-,-,-,-,-,x,-,-,-,x
x,x,x,x,x,x,x,x,x,x,x

r,d,d,l,u,l,r,l,r,r,r,u,r
```

Figure 4: Before `onePlayerManyMoves`

```
xxxxxxxxxx
x-x-----x-x
x----x--x-x
x---------x
xxx--x-xx-x
x-----x---x
xx-----x--x
x-x-x--g--x
x-x---xx-xx
x---x--x--x
x-x-xx-xx-x
x-xx------x
x-----x-x-x
x-x-x----xx
x---x--x-xx
x--xxx-x--x
x-------1-x
x--x---x--x
x----x---xx
x-xx--x---x
x-----x---x
xxxxxxxxxx
```

Figure 5: After `onePlayerManyMoves`

4 Many Players , One Move

The third part (`manyPlayersOneMove` in the file `csce322homeWork02part03.js`) will take in one (1) argument (a maze) and return a function that takes in one (1) argument (a move), and returns the maze that is the result of Player 1 making that move. This differs from the first part in that there will be more than one (1) player in the maze.

```
x,x,x,x,x,x,x,x,x,x,x,x,x,x
x,x,x,x,-,-,-,-,-,-,x,-,-,x
x,-,x,-,-,-,x,1,x,-,x,-,-,x
x,-,-,-,x,-,-,-,-,-,-,x,-,x
x,-,-,-,-,x,-,-,x,-,-,-,-,x
x,-,x,x,x,-,-,-,-,-,-,-,-,x
x,x,-,x,-,-,-,x,x,-,x,-,-,x
x,x,-,-,x,x,-,x,x,-,-,-,x,x
x,-,-,-,-,-,-,x,x,x,x,-,3,x
x,-,x,x,-,x,x,x,x,x,x,x,-,x
x,x,-,x,-,-,x,x,x,-,-,-,-,x
x,-,-,-,x,-,x,x,x,-,x,-,-,x
x,-,x,-,-,-,-,x,x,-,x,x,-,x
x,-,-,-,x,x,-,x,x,-,-,-,-,x
x,-,-,x,x,-,-,x,x,-,x,-,-,x
x,x,-,-,-,-,-,-,x,2,-,-,-,x
x,x,x,x,-,-,x,-,x,x,x,x,-,x
x,-,-,-,-,-,-,-,-,x,x,x,x,-,x
x,-,-,x,-,x,-,x,x,x,x,x,-,x
x,x,-,-,-,-,-,-,-,-,-,x,-,-,x
x,-,-,x,-,x,-,-,x,-,-,-,x,x
x,x,-,-,-,-,-,x,-,-,x,-,-,x
x,x,x,-,-,-,x,x,-,-,-,-,-,x
x,-,-,x,-,-,-,-,x,-,x,x,-,x
x,-,x,x,-,-,-,-,-,-,-,x,-,x
x,-,-,-,-,x,-,x,-,x,-,-,-,x
x,-,-,-,-,-,-,x,-,-,-,-,x,x
x,-,-,x,-,-,-,-,-,g,-,-,x,x
x,x,x,x,x,x,x,x,x,x,x,x,x,x

d,d,r,d,d,r,l,r,u,d,d,d,r,l,r
```

Figure 6: Before `manyPlayersOneMove`

```
xxxxxxxxxxxxx
xxxx------x--x
x-x---x-x-x--x
x---x--1---x-x
x----x--x----x
x-xxx--------x
xx-x---xx-x--x
xx--xx-xx---xx
x------xxxx-3x
x-xx-xxxxxxx-x
xx-x--xxx----x
x---x-xxx-x--x
x-x----xx-xx-x
x---xx-xx----x
x--xx--xx-x--x
xx------x2---x
xxxx--x-xxxx-x
x-------xxxx-x
x--x-x-xxxxx-x
xx--------x--x
x--x-x--x---xx
xx-----x--x--x
xxx---xx-----x
x--x----x-xx-x
x-xx-------x-x
x----x-x-x---x
x------x----xx
x--x-----g--xx
xxxxxxxxxxxxx
```

Figure 7: After `manyPlayersOneMove`

5 Many Players , Many Moves

The fourth part (`manyPlayersManyMoves` in the file `csce322homeWork02part04.js`) will take in one (1) argument (a maze) and return a function that takes in one (1) argument (a list of moves), and returns the maze that is the result of each player making the next move in the list in turn (starting with Player 1). If the maze has already been solved, the maze is returned in that state.

```
x,x,x,x,x,x,x,x,x,x,x,x
x,-,-,-,-,-,-,-,-,2,-,x
x,-,x,x,x,-,x,-,-,x,-,x
x,-,x,-,-,-,-,-,x,x,-,x
x,-,-,-,x,-,-,-,x,-,-,x
x,-,x,-,-,-,x,-,-,-,-,x
x,x,-,x,x,-,-,-,x,x,-,x
x,-,-,-,-,-,x,x,g,-,-,x
x,-,x,x,x,x,x,x,-,x,-,x
x,-,-,x,x,-,-,-,-,-,-,x
x,x,-,-,-,-,x,-,-,-,1,x
x,x,x,x,x,x,x,x,x,x,x,x

u,d,d,l,r,l,d,r,d,l,l
```

Figure 8: Before `manyPlayersManyMoves`

```
xxxxxxxxxxx
x------2---x
x-xxx-x--x-x
x-x-----xx-x
x---x---x--x
x-x---x----x
xx-xx---xx-x
x-----xxg--x
x-xxxxx-x-x
x--xx------x
xx----x-1--x
xxxxxxxxxxx
```

Figure 9: After `manyPlayersManyMoves`

6 Naming Conventions

Your files should follow the naming convention of `csce322homeWork02part01.js`, `csce322homeWork02part02.js`, `csce322homeWork02part03.js`, and `csce322homeWork02part04.js`.

6.1 `helpers.js`

A file named `helpers.js` has been provided with the functionality to read the `.emf` files into matrices. If a modified `helpers.js` file is not included with your submission, the default will be used in its place.

7 webgrader Note

Submissions will be tested with `node.js`, note the browser. `cse.unl.edu` is currently running version 4.8.4 of `node`.

8 Point Allocation

Component	Points
`csce322homeWork02part01.js`	
Test Cases	1×20
Total	20
`csce322homeWork02part02.js`	
Test Cases	1×20
Total	20
`csce322homeWork02part03.js`	
Test Cases	1×30
Total	30
`csce322homeWork02part04.js`	
Test Cases	1×30
Total	30
Total	100

9 External Resources

JavaScript Tutorial