Assignment 02 (Due: Wednesday, October 22, 2014)

CSCE 322

1 Instructions

In this assignment, you will be required to write JavaScript functions that determine where a Chutes and Ladders piece will move given a certain roll of the die.

1.1 Data File Specification

An example of a properly formatted file is shown in Figure 1. The first row encodes the positions of the pieces on the board. The second row encodes the locations of chutes on the board. The third row encodes the locations of ladders on the board. Finally, the remaining rows encode the board (each location holds the board position that you would travel to from that location, 0 denotes the end of the board).

```
1,1,1
4,24,27
9,16,28
0,30,30,10,27
22,23,24,6,26
21,20,19,18,27
12,13,14,15,16
11,13,9,8,7
2,3,4,3,6
```

Figure 1: A properly formatted Chutes and Ladders encoding

2 Moving without Chutes or Ladders

The first part (`withoutCaL` in the file `csce322as02pt01.js`) will assume movement on the board without the effect of chutes and ladders. `withoutCaL` should take a matrix (the game state) as an argument and return a function that, given the position of a piece and a value of a roll, returns the position that the given piece should move to. This function will be provided with a game state that only contains player positions and the board.

3 Moving with Chutes

The second part (`withoutL` in the file `csce322as02pt02.js`) will assume movement on the board without the effect of ladders. `withoutL` should take a matrix (the game state) as an argument and return a function that, given the position of a piece and a value of a roll, returns the position that the given piece should move to. This function will be provided with a game state that only contains player positions, chute positions, and the board.

4 Moving with Chutes and Ladders

The third part (withCaL in the file csce322as02pt03.js) will assume movement on the board with the effect of chutes **and** ladders. withCaL should take a matrix (the game state) as an argument and return a function that, given the position of a piece and a value of a roll, returns the position that the given piece should move to. This function will be provided with a game state that contains player positions, chute positions, ladder positions, and the board.

5 Naming Conventions

Your files should follow the naming convention of csce322as02pt01.js, csce322as02pt02.js, and csce322as02pt03.js

5.1 helpers.js

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

6 webgrader Note

Submissions will be tested with node.js, not the browser. cse.unl.edu is currently running version 0.8.12 of node.

7 Point Allocation

Component	Points
csce322as02pt01.js	20%
csce322as02pt02.js	30%
csce322as02pt03.js	40%
README.txt	10%
Total	100%

8 External Resources

JavaScript Tutorial