Assignment 02 (Due: Friday, March 8, 2019, 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 playing of the variation of Connect Four.

1.1 Data File Specification

An example of properly formatted file is shown in Figure 1. The first file encodes a game and the second file encodes the moves to be made.

test02.game.cnf

```
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,4,-,-,-,-,1
-,2,3,4,2,3,1
```

test02.moves.cnf

```
6,3,5,4,7,6,5,1,2,7,7,1,1,5,1,4,2,3,3,4,6,3,2,2,7,3,5,4,7,1,4,2,4,6,1,1,6,5,3,6,5
```

Figure 1: A properly formatted encoding

2 One Player, One Move

The first part (`onePlayerOneMove` in the file `csce322assignment02part01.js`) will take in one (1) argument (a game) and return a function that takes in one (1) argument (a move), and returns the game that is the result of Player 1 dropping a `1` into the column of the game denoted by the move. The `1` should fall (move down through the `-`'s in the column) until it reaches a space that is not `-` or reaches the last row in the game. An example is provided below

test02.game.cnf

```
-,-,-,-,-,-,-
```

```
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,4,-,-,-,-,1
-,2,3,4,2,3,1
```

6,3,5,4,7,6,5,1,2,7,7,1,1,5,1,4,2,3,3,4,6,3,2,2,7,3,5,4,7,1,4,2,4,6,1,1,6,5,3,6,5

test02.onePlayerOneMove.solution

```
game
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '4', '-', '-', '-', '1', '1' ]
[ '-', '2', '3', '4', '2', '3', '1' ]
```

3   One Player, Many Moves

The second part (`onePlayerManyMoves` in the file `csce322assignment02part02.js`) will take in one (1) argument (a game) and return a function that takes in one (1) argument (an arry of moves), and returns the game that is the result of Player 1 playing each move in succession (following the rules of onePlayerOneMove) until all of the moves in the array have been played, or Player 1 has placed four (4) 1's in a row horizontally, vertically, or diagonally.

test02.game.cnf

```
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,4,-,-,-,-,1
-,2,3,4,2,3,1
```

test02.moves.cnf

6,3,5,4,7,6,5,1,2,7,7,1,1,5,1,4,2,3,3,4,6,3,2,2,7,3,5,4,7,1,4,2,4,6,1,1,6,5,3,6,5

test02.onePlayerManyMoves.solution

```
game
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
```

```
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '4', '1', '1', '1', '1', '1' ]
[ '-', '2', '3', '4', '2', '3', '1' ]
```

4   Many Players , One Move

The third part (`manyPlayersOneMove` in the file `csce322assignment02part03.js`) will take in one
(1) argument (a game) and return a function that takes in one (1) argument (an arry of moves), and
returns the game that is the result of each player in the game making exactly one move until each
player has completed a move or a player has won the game (placed four of their pieces vertically,
horizontally, or diagonally). Player 1 places 1s, Player 2 places 2s, etc. (you may assume the
highest number in the provided game is the number of players in the game). The moves are made
in the order they appear in the moves array.

<div align="center">test02.game.cnf</div>

```
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,4,-,-,-,-,1
-,2,3,4,2,3,1
```

<div align="center">test02.moves.cnf</div>

```
6,3,5,4,7,6,5,1,2,7,7,1,1,5,1,4,2,3,3,4,6,3,2,2,7,3,5,4,7,1,4,2,4,6,1,1,6,5,3,6,5
```

<div align="center">test02.manyPlayersOneMove.solution</div>

```
game
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '4', '2', '4', '3', '1', '1' ]
[ '-', '2', '3', '4', '2', '3', '1' ]
```

5   Many Players , Many Moves

The fourth part (`manyPlayersManyMoves` in the file `csce322assignment02part04.js`) will take
in one (1) argument (a game) and return a function that takes in one (1) argument (an arry of
moves), and returns the game that is the result of each player in the game taking turns making a
move until all of the moves in the array have been exhausted or a player has won the game (placed
four of their pieces vertically, horizontally, or diagonally). Player 1 places 1s, Player 2 places 2s,
etc. (you may assume the highest number in the provided game is the number of players in the
game).

<div align="center">test02.game.cnf</div>

```
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,-,-,-,-,-,-
-,4,-,-,-,-,1
-,2,3,4,2,3,1
```

<div align="center">test02.moves.cnf</div>

```
6,3,5,4,7,6,5,1,2,7,7,1,1,5,1,4,2,3,3,4,6,3,2,2,7,3,5,4,7,1,4,2,4,6,1,1,6,5,3,6,5
```

<div align="center">test02.manyPlayersManyMoves.solution</div>

```
game
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '-' ]
[ '-', '-', '-', '-', '-', '-', '3' ]
[ '3', '1', '3', '4', '2', '-', '2' ]
[ '1', '1', '2', '4', '3', '2', '1' ]
[ '4', '4', '2', '4', '3', '1', '1' ]
[ '4', '2', '3', '4', '2', '3', '1' ]
```

6 Extra Credit (10%)

Games will contain an arbitrary number of players.

7 Naming Conventions

Your files should follow the naming convention of
`csce322assignment02part01.js`, `csce322assignment02part02.js`, `csce322assignment02part03.js`,
and `csce322assignment02part04.js`.

7.1 `helpers.js`

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

8 webgrader Note

Submissions will be tested with `node.js`, not the browser. `cse.unl.edu` is currently running
version 8.11.4 of `node`.

9 Point Allocation

Component	Points
`csce322assignment02part01.js`	
Test Cases	1×20
Total	20
`csce322assignment02part02.js`	
Test Cases	1×20
Total	20
`csce322assignment02part03.js`	
Test Cases	1×30
Total	30
`csce322assignment02part04.js`	
Test Cases	1×30
Total	30
Total	100

10 External Resources

JavaScript Tutorial