Project One

Michael Hunsinger

February 8, 2015

Contents

Project

The first project for Linguistic Geometry, Spring semester of 2015.

Installation

The project was done using Node.js and requires Node.js along with npm to be installed (usually comes with Node.js). Node.js can be downloaded from the Node.Js website. It can also be install through a package manager if running Linux.

Build

Once Node.js is installed, navigate to the directory containing the project. The node modules were not bundled and will have to be installed via the following command

```
npm install
```

Run

After the modules are installed, run the following command to run the project.

```
npm run project res/chess.json
```

Input

The `chess.json` file provided contains all chess pieces. Below is a small snippet of the file.

```
{
    "board": {
"xMax": 8,
"yMax": 8,
"zMax": null,
"obstacles": [ ]
    },
    "pieces": [
{
    "piece": "King",
    "xCoordinate": 2,
    "yCoordinate": 7,
    "reachability": [
{
    "condition-1": "| y1 - x1 | <= 1"
},
{
    "condition-1": "| y2 - x2 | <= 1"
}
```

```
    ]
}
    ]
}
```

As you can see, it contains information about the board and any number of pieces.

Board

Below are some notes regarding the fields for the board.

- The board is 1-based and assumes the min is '1' and thus not required.
- Obstacles are an array of points that cannot be reached by any piece. Below is a sample of an obstacle object that would cause the point (3,2) to be unreachable..

```
{
    "x": 3,
    "y": 2
}
```

- The board information stays constant when reading each piece.

Pieces

Below are some notes regarding the fields for pieces.

- `x` and `y` are the starting coordinates for the given piece.
- Reachability is an array of conditions that are used to determine if a given cell is reachable. For each reachability, all conditions must be met in order for it to be reachable. However, not all reachabilities must be hold true for the cell to be reachable.

Output

Below is sample output for a King.

```
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
7 6 6 6 6 6 6 6 6 6 6 6 6 6 7
7 6 5 5 5 5 5 5 5 5 5 5 5 6 7
7 6 5 4 4 4 4 4 4 4 4 4 5 6 7
7 6 5 4 3 3 3 3 3 3 3 4 5 6 7
7 6 5 4 3 2 2 2 2 2 3 4 5 6 7
7 6 5 4 3 2 1 1 1 2 3 4 5 6 7
7 6 5 4 3 2 1 0 1 2 3 4 5 6 7
7 6 5 4 3 2 1 1 1 2 3 4 5 6 7
7 6 5 4 3 2 2 2 2 2 3 4 5 6 7
7 6 5 4 3 3 3 3 3 3 3 4 5 6 7
7 6 5 4 4 4 4 4 4 4 4 4 5 6 7
7 6 5 5 5 5 5 5 5 5 5 5 5 6 7
7 6 6 6 6 6 6 6 6 6 6 6 6 6 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
```