LMLT Technical Test – Front End Developer

Introduction

This test is intended to give LMLT a good idea of your technical skills in Front End Development and React / Redux and to review your code style, structure and techniques. The results may be used as a discussion point in any face-to-face interview that follows.

The test can take as long as you like, there is no set time limit.

Requirements React/Redux Project

See the supporting files included with this Test.

Please use the React and Redux JavaScript libraries to create a UI consisting of:

- An input labelled 'Rows' which only allows integers and max value is 20
- An input labelled 'Column' which only allows integers and max value is 20
- A button labelled 'Generate'
- An X by Y grid of squares each of which is large enough to be clicked with the mouse.
- The look of the UI should match the one supplied to you in the zip file attached.

Additional requirements:

- The initial dimensions of the grid should be 10 by 10.
- When the grid is first initialised or resized exactly one square on the left is chosen at random and marked as a 'start' square and coloured appropriately.
- When the grid is first initialised or resized exactly one square on the right is chosen at random and marked as an 'end' square and coloured appropriately.
- The initial state of each square in the grid except the 'start' and 'end' squares is 'filled' and it should be coloured appropriately.
- Clicking on any 'filled' square which is not the 'start' or 'end' square changes its state to 'clear', and it should be re-coloured appropriately.
- Clicking on any 'clear' square which is not the 'start' or 'end' square changes its state to 'filled'
- If the value of either input is changed and the generate button is clicked, the grid should be resized appropriately, the 'start' and 'end' squares should be re-set as described above and all squares other squares should be re-set to 'filled'
- If at any point there is a connected path of 'clear' squares all the way from the 'start' square to the 'end' square then the path should be coloured appropriately.
- 'clear' squares are only connected by their sides. They do not connect diagonally.
- If there is more than one path from the 'start' square to the 'end' square then only the shortest path should be shown.

Depending on your approach you may find that you need a min or max priority queue. Please feel free to use one of the priority queue libraries on NPM.

Feel free to use es5/es6 as you prefer. We use TypeScript but unless you are already familiar with it I recommend you stick to JavaScript for the test.

Results

Please provide your results in a Zip file, or access to a code repository, containing the following :

- Copy of the source code
- Any notes or comments that you want to make about :
  - your approach to implementation
  - issues faced

    - design decisions

Evaluation

We will be looking for the following in evaluating the App

- App performance – does it work ?
- Coding standards and good coding practice
- Knowledge of React / Redux
- Good use of Libraries
- Good architecture & design
- UI elements & usability