

CSC4130 Assignment 4

Start: Nov 15 at 10:00 PM

End: Nov 29 at 11:59 PM

1 Description

This assignment aims to help students get familiar with React (i.e., a user interface framework) and how to utilize D3 and React to build a visual interface with multiple views.

Multiple-View Interaction

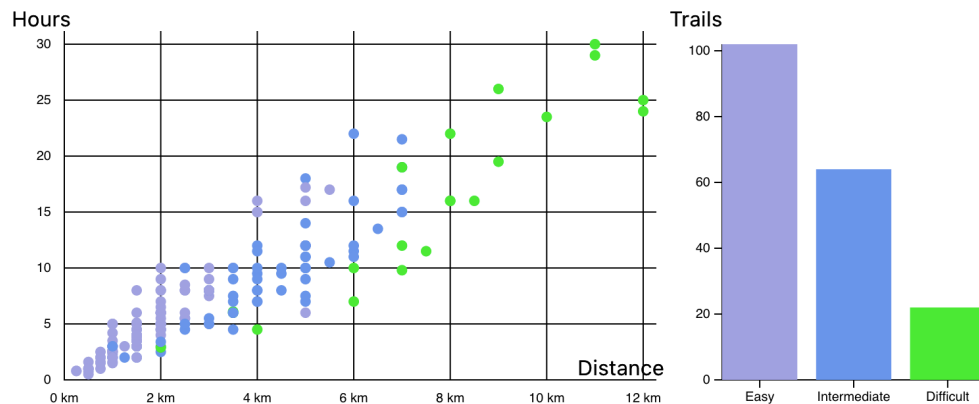


Figure 1: The result of assignment 4.

2 Install React and D3

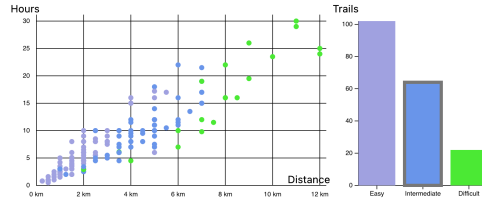
Open the terminal and go to the Assignment 4 folder. Then use the following terminal commands to install React and D3 on your computer.

```
npm install  
npm i d3
```

After that, type the following terminal command to run the program.

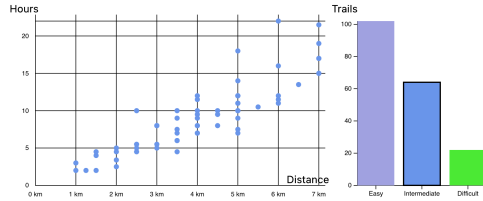
```
npm start
```

Multiple-View Interaction



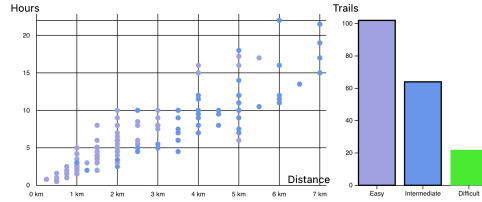
(a)

Multiple-View Interaction



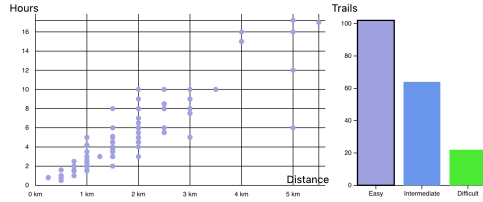
(b)

Multiple-View Interaction



(c)

Multiple-View Interaction



(d)

Figure 2: Multiple-view interaction in the designed interface. (a) When a mouse hovers to a bar in the bar chart, the hovered bar will be highlighted by adding a grey bounding box, (b) users click a bar in the bar chart, the clicked bar will be highlighted by adding a black bounding box, and the corresponding points in the scatter plot will be shown, (c) users click multiple bars in the bar chart and the corresponding points in the scatter plot will be shown, and (d) users click the clicked bar again, the points belong to the bar will not be visualized.

3 Requirement

- Given a .csv file stored in the public folder, you need to visualize the data into two forms (i.e., scatter chart and bar chart), as shown in Figure 1.
- When users use a mouse to hover a bar in the bar chart, the hovered one will be highlighted by adding a grey bounding box, as shown in Figure 2 (a). Hover means that the mouse moves to a bar and there are no mouse press and release actions.
- When users use a mouse to click a bar, the bar will be highlighted by adding a black bounding box. In addition, in the scatter plot, we only show the points belonging to the clicked bar, as shown in Figure 2 (b).
- Users can click multiple bars, the bars will be highlighted by adding black bounding boxes to each. In addition, in the scatter plot, we only show the points belonging to the clicked bars, as shown in Figure 2 (c).
- After clicking, users can click the clicked bar again, then the highlight will be removed, and the corresponding points in the scatter plot will not be visualized, as shown in Figure 2 (d).

Implementation Hints

- Following assignment 3, the Barchart and ScatterPlot.js still have the functions called `initVis` and `updateVis`, which are inline functions in a functional component. The `renderVis` is embedded into `updateVis`.
- In this template, we use React functional component, so you need to think about when and how to call `initVis` and `updateVis` in the two functional components, i.e., `BarChart` and `ScatterPlot`.
- Use `React.useState()` to bind and filter data.
- Use `React.useEffect()` to render data when dependency/dependencies change.
- Under different rendering conditions, consider which dependency/dependencies should be passed into `useEffect()`.

- You are allowed to use React class component to complete this assignment.

Online Resources

- [useState](#)
- [useState Guide](#)
- [useEffect](#)
- [useEffect Explanation](#)
- [React and D3](#)
- [D3 with React](#)

4 Evaluation

In total, there are 100 points in this assignment. A detailed evaluation is provided here.

1) In ScatterPlot.js, initialize the scatter plot and update rendering result (10pts).

- Initialization (5pts).
- Update (5pts).

2) In BarChart.js, initialize the bar chart and update rendering result (10pts).

- Initialization (5pts).
- Update (5pts).

3) In BarChart.js, complete the click function (20pts).

- Check current category/categories (5pts).
- Obtain filtered category/categories (10pts).
- Change React state using the selected category/categories (5pts).

4) In App.js, create three states, i.e., data, selectedData, and filterCategory (15pts).

- Each state is for 5pts.

5) In App.js, render and update visual results when dependency/dependencies change (30pts).

- Each one is for 15pts.

6) In App.css, stylize the hover and click actions (10pts).

- Hover is for 5pts.
- Click is for 5pts.

7) Submission (5pts). Please compress your code and a readme file (optional) into a zip file and submit the zip file to OneDrive. The readme file can include descriptions that can help the instructor run the interface successfully.

Note that a penalty of 10 pts will be given to those students who submit the assignment between Nov 30 at 12:00 AM and Nov 30 at 11:59 PM. A penalty of 20 pts will be given to those students who submit the assignment between Dec 1 at 12:00 AM and Dec 1 at 11:59 PM. Submissions after Dec 2 at 12:00 AM will not be graded.