Tutorial 4: D3Js in React (Part2)

Nicolas Médoc Luxembourg Inst. of Science and Technology

Outline

1. Recall of React concepts

React principles: components

- What is a component ?

React principles: components

- What is a component ?
  - A React **component** is a function responsible to render a part of the user interface
  - the function returns **JSX code**, a combination of HTML and Javascript generating the final HTML page displayed in the browser
  - the **component tree** is the tree structure resulting from the recursive declaration of all components from the root component (App)
  - **pure components:** try to build pure components, i.e. the same input (props + state) render the same output (HTML)

React principles: props

- Why using **props** ?
  <MyChildComponent myProps={myPropsValue}>

React principles: props

- Why using **props** ?
  <MyChildComponent myProps={myPropsValue}>
  - to declare a piece of data sent by the parent component and used by the child component to render its DOM structure
  - to declare the child component to be an observer of any update of this data

React principles: props example

- Declaring props in a component:

```
function MyChildComponent({myProps}){
    // do something with myProps
    return (
        // return JSX component
    )
}
```

- Sending props data to the child component

```
<MyChildComponent myProps={anyValue}>
```

React principles: states

- Why using **states**?
  const [state,setState] = useState(initialState);

React principles: states

- Why using **states**?
  const [state,setState] = useState(initialState);
  - to memorize the data between the rendering cycles
  - to notify the child components to re-render when the state is updated with the setter method
  - to store data which is reactive (updated by external events)
  - avoid storing in a state any data that can be computed with a pure logic (same input => same output) from another state
  - declare a state as deeply as possible in the component tree

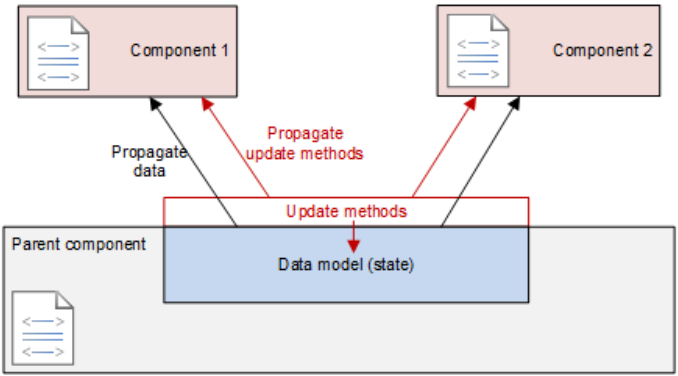
React principles: synchronizing components

- How to synchronize multiple components?

React principles: synchronizing components

- How to synchronize multiple components?
  - declare a state in the closest common ancestor
  - declare an event handler function to set the data state in the closest common parent
  - send the handler function to the child components as props
  - the child component that triggers the event call the event handler function to update the state
  - when the state setter function is called in the common ancestor, all the child components are notified to re-render (with updated props values according to the new parent states)
- **Examples:** See exercises 3 and 4 of the Tuto3-D3js-React (part1).

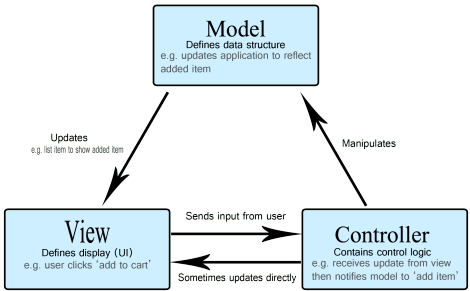
React principles: synchronizing components



Synchronization of multiple components

Model View Controller (MVC) design pattern

Does React props and states allows to implement MVC design pattern?



https://developer.mozilla.org/en-US/docs/Glossary/MVC