Amrita Vadhera
REACT Tutorial Notes
February 8, 2018
Software Dev. Team 31

What is react? React is a declarative, efficient, and flexible JavaScript library for building user interfaces.

React component class or React component type
example: class ShoppingList { }

A component takes in parameters and returns a value to the render method
example: Shopping List for {this.props.name}
in this example, “props” is the parameter

The render( ) method returns a description of what you want to render, and then React takes that description and renders it to the screen. Render returns a react element, which is a lightweight description of what to render.

React components can have state by setting “ this.state ”.

How to set the initial state of the button and use “ onClick ”:

```
render() {
    return (
      <button className="square" onClick={() => this.setState({value:
'X'})}>
        {this.state.value}
      </button>
    );
  }
```

What the following code above is doing is- When the component rerenders, this.state.value will be 'X' so you’ll see an X in the grid.

When you want to aggregate data from multiple children or to have two child components communicate with each other, move the state upwards so that it lives in the parent component. The parent can then pass the state back down to the children via props, so that the child components are always in sync with each other and with the parent. In this example of “Tic Tac Toe”, we do this to keep track of who is winning. Therefore, we need to have the value of all 9 squares in one place rather than thinking of each square component as individual. To do this, we need to replace a few things in the constructor:

Amrita Vadhera
REACT Tutorial Notes
February 8, 2018
Software Dev. Team 31

```
class Square extends React.Component {
  render() {
    return (
      <button className="square" onClick={() => this.props.onClick()}>
        {this.props.value}
      </button>
    );
  }
}
```

Now when the square is clicked, it calls the onClick function that was passed by Board. Let's recap what happens here:

1. The onClick prop on the built-in DOM <button>component tells React to set up a click event listener.
2. When the button is clicked, React will call the onClick event handler defined in Square's render() method.
3. This event handler calls this.props.onClick(). Square's props were specified by the Board.
4. Board passed onClick={() => this.handleClick(i)}to Square, so, when called, it runs this.handleClick(i)on the Board.
We have not defined the handleClick() method on the Board yet, so the code crashes.

Why is immutability important?

Immutability is important because it can help us increase component and overall application performance. It also makes it easier to undo/redo changes as well as track changes made to our code.

Functional Components:

In this example, it is used for a component like Square.

How to allow for "Taking turns" in the game:

Create a Boolean "xIsNext" and default it to true.
Then set in the constructor; xIsNext as !this.state.xIsNext
When this Boolean is true, player X will go and when false player Y will go.
To allow Y to go next, we also have to set the status as:

const status = 'Next player: ' + (this.state.xIsNext ? 'X' : 'O');

Show that the game is over and a certain player has won:

```
if (calculateWinner(squares) || squares[i]) {
      return;
    }
```

Amrita Vadhera
REACT Tutorial Notes
February 8, 2018
Software Dev. Team 31

Storing a History:

```
history = [
  {
    squares: [
      null, null, null,
      null, null, null,
      null, null, null,
    ]
  },
  {
    squares: [
      null, null, null,
      null, 'X', null,
      null, null, null,
    ]
  },
  // ...
]
```

Showing the moves:

```
const moves = history.map((step, move) => {
      const desc = move ?
        'Go to move #' + move :
        'Go to game start';
      return (
        <li>
          <button onClick={() => this.jumpTo(move)}>{desc}</button>
        </li>
      );
    });
```

Keys:

Key is a special property that's reserved by React (along with ref, a more advanced feature). When an element is created, React pulls off the key property and stores the key directly on the returned element. Even though it may look like it is part of props, it cannot be referenced with this.props.key. React uses the key automatically while deciding which children to update; there is no way for a component to inquire about its own key.

It's strongly recommended that you assign proper keys whenever you build dynamic lists.

Amrita Vadhera
REACT Tutorial Notes
February 8, 2018
Software Dev. Team 31
In conclusion:

Now, you've made a tic-tac-toe game that:

- lets you play tic-tac-toe,
- indicates when one player has won the game,
- stores the history of moves during the game,
- allows players to jump back in time to see older versions of the game board.

**if time: set up react and spring boot connection*
*spring boot project -> if the UI can display a hello world**