Anica Wang

305-532-153

Professor Eggert

CS 35L

Tic Tac Toe

setting up the game:

* download react
* npx create-react-app tic-tac-toe
* remove all files in src folder
* create index.css and index.js
* run npm start

adding components in React:

* need parameters or props
* returns hierarchy of views to display via render method
  + render returns React element, describes what to render
* ex.

return React.createElement('div', {className: 'shopping-list'},

React.createElement('h1', /\* ... h1 children ... \*/),

React.createElement('ul', /\* ... ul children ... \*/)

);

index.js

* 3 components: square, board, game
* pass props
  + board passes prop to square using this.props.value
* arrow syntax avoids confusing behavior of this
* class Square extends React.Component {

render() {

return (

<button className="square" onClick={() => alert('click')}> {this.props.value}

</button>

);

}

}

* components use state to remember things
  + setting this.state in constructor is private to React component it is defined in
  + constructors initialize state
  + constructor(props) {

super(props);

this.state = {

value: null,

};

}

* in JavaScript classes, need to always call super when defining the constructor of a subclass. all React component classes that have a constructor should start with a super(props) call
* To collect data from multiple children, or to have two child components communicate with each other, you need to declare the shared state in their parent component instead. The parent component can pass the state back down to the children by using props; this keeps the child components in sync with each other and with the parent component.
* write constructor for Board to set Board’s initial state to contain array of 9 nulls corresponding to 9 squares
* modify the Board’s renderSquare method to read from sqares array:

renderSquare(i) {

return <Square value={this.state.squares[i]} />; }

* + Each Square will now receive a value prop that will either be 'X', 'O', or null for empty squares.
* need to create a way for square to update board state but since state is considered to be private to component that defines it, cannot update board’s state directly from square
  + instead pass a function from the board to the square and have square call that function when a square is clicked

renderSquare(i) {

return (

<Square

value={this.state.squares[i]}

onClick={() => this.handleClick(i)} />

);

}

* split returned element into multiple lines for readability and add parentheses so JavaScript doesn’t insert a semicolon after return to break the code
* The DOM <button> element’s onClick attribute has a special meaning to React because it is a built-in component. For custom components like Square, the naming is up to you. We could give any name to the Square’s onClick prop or Board’s handleClick method, and the code would work the same. In React, it’s conventional to use on[Event] names for props which represent events and handle[Event] for the methods which handle the events.
* components that no longer maintain state and receive values from other components are controlled components
* Immutability makes complex features much easier to implement.
* Detecting changes in mutable objects is difficult because they are modified directly. This detection requires the mutable object to be compared to previous copies of itself and the entire object tree to be traversed.
* In React, function components are a simpler way to write components that only contain a render method and don’t have their own state.
* In JavaScript, arrays have a [map() method](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map) that is commonly used for mapping data to other data, for example:
  + const numbers = [1, 2, 3];
  + const doubled = numbers.map(x => x \* 2); // [2, 4, 6]

npm pack –-dry-run

anicawang@Anicas-MacBook-Pro tic-tac-toe % npm pack --dry-run

npm notice

npm notice 📦 tic-tac-toe@0.1.0

npm notice === Tarball Contents ===

npm notice 3.4kB README.md

npm notice 814B package.json

npm notice 3.9kB public/favicon.ico

npm notice 1.7kB public/index.html

npm notice 5.3kB public/logo192.png

npm notice 9.7kB public/logo512.png

npm notice 492B public/manifest.json

npm notice 67B public/robots.txt

npm notice 641B src/index.css

npm notice 5.7kB src/index.js

npm notice === Tarball Details ===

npm notice name: tic-tac-toe

npm notice version: 0.1.0

npm notice filename: tic-tac-toe-0.1.0.tgz

npm notice package size: 23.6 kB

npm notice unpacked size: 31.7 kB

npm notice shasum: 1ec0b8c1cf514f1e0da6b718a833d898ea96bfec

npm notice integrity: sha512-DP7p4ckz33wx7[...]dOkv938oRLyaA==

npm notice total files: 10

npm notice

tic-tac-toe-0.1.0.tgz