

Tic Tac Toe

Tuesday, March 22, 2022 5:10 PM

- State should be managed in board
- Board should know
 - What each cell should display
 - What will be next value
 - Should current cell be updated on the click or not
- We need to upgrade the board component from function component to class component
- Cell doesn't need to maintain it's own state.
- It should display the state passed by the parent

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

```
export default Cell;
```

- Now this component doesn't have any state. It can be downgraded to a function component
 - You may keep it as class component
 - But function components are easier to maintain

```
const Cell=(props)=>{  
  return <button  
    className="cell">{props.value}</button>;  
}
```

```
export default Cell;
```

Child sending message to parent

- When cell is clicked, it should inform Board that it has been clicked.
- Board must know which cell is clicked.
 - Cell should pass it's id to Board

1. Board should pass a function to Child : onCellClick

```
<div className='board'>  
  <Cell value={this.state.cells[0]} id={0} onCellClick={this.handleClick}/>  
  <Cell value={this.state.cells[1]} id={1} onCellClick={this.handleClick}/>  
  <Cell value={this.state.cells[2]} id={2} onCellClick={this.handleClick}/>  
</div>
```

2. Child should call this function and pass the id

```
const Cell=(props)=>{  
  return <button  
    onClick={()=> props.onCellClick(props.id)}  
    className="cell">{props.value}</button>;  
}
```

- When button is clicked we pass a lambda function
- This function calls onCellClick with proper id

Dynamic Style

- Sometimes we want to apply some style dynamically and can't be written in CSS
- We can create a JavaScript object to represent the style
- Pass this object as style props in JSX

```
const Cell=(props)=>{  
  let value = props.value || '-'; //if props.value === falsy use '-'  
  
  let style={  
    color: props.value ? "black" : "transparent"  
  };  
  
  return <button  
    style={style}  
    onClick={ ()=> props.onCellClick(props.id)}  
    className="cell">{value}</button>;  
}
```

- No value in button is breaking the alignment
 - We use a dummy text instead null
- Since we don't really want to show anything, we change the style color to transparent which will make it invisible