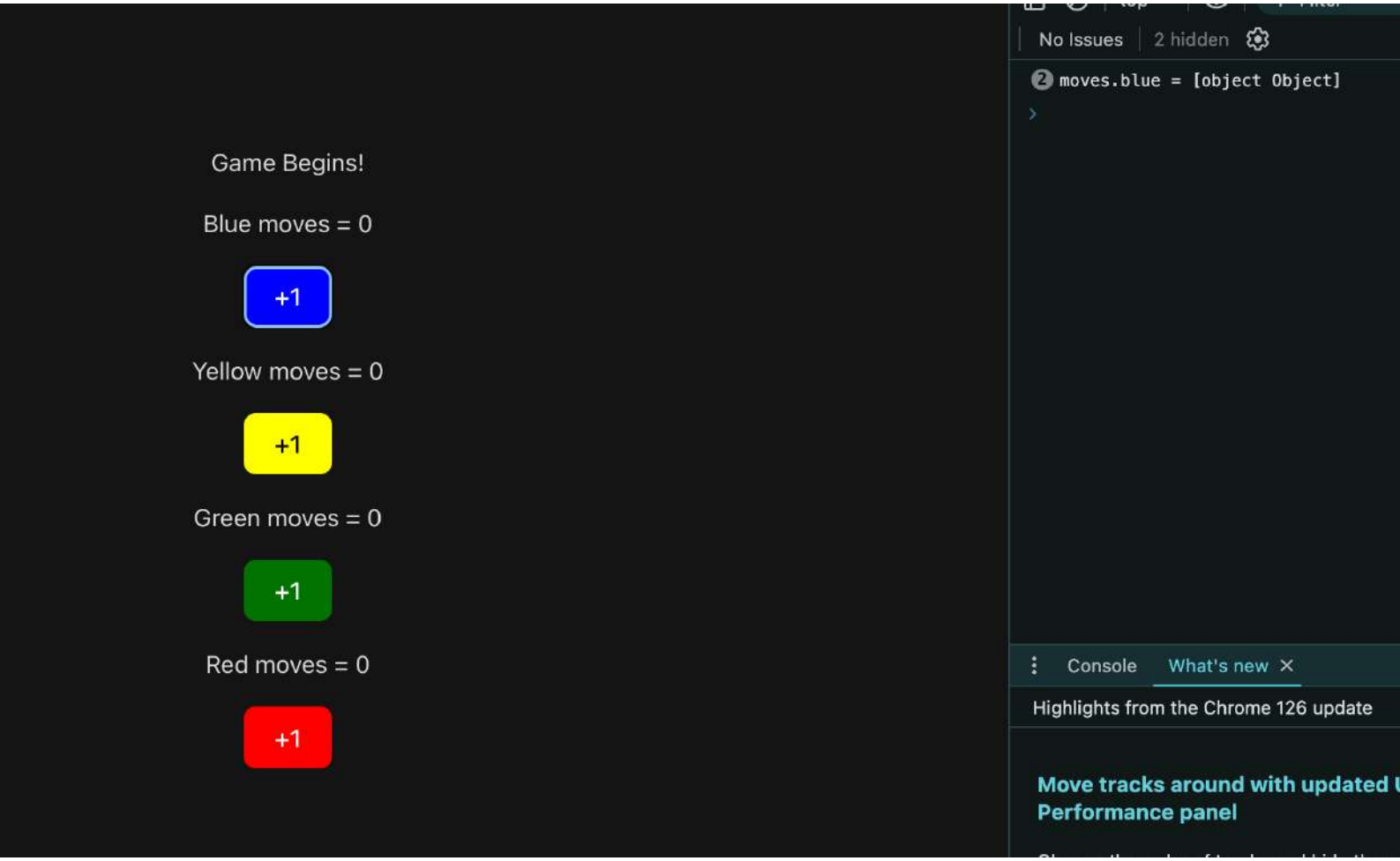
Objects & State

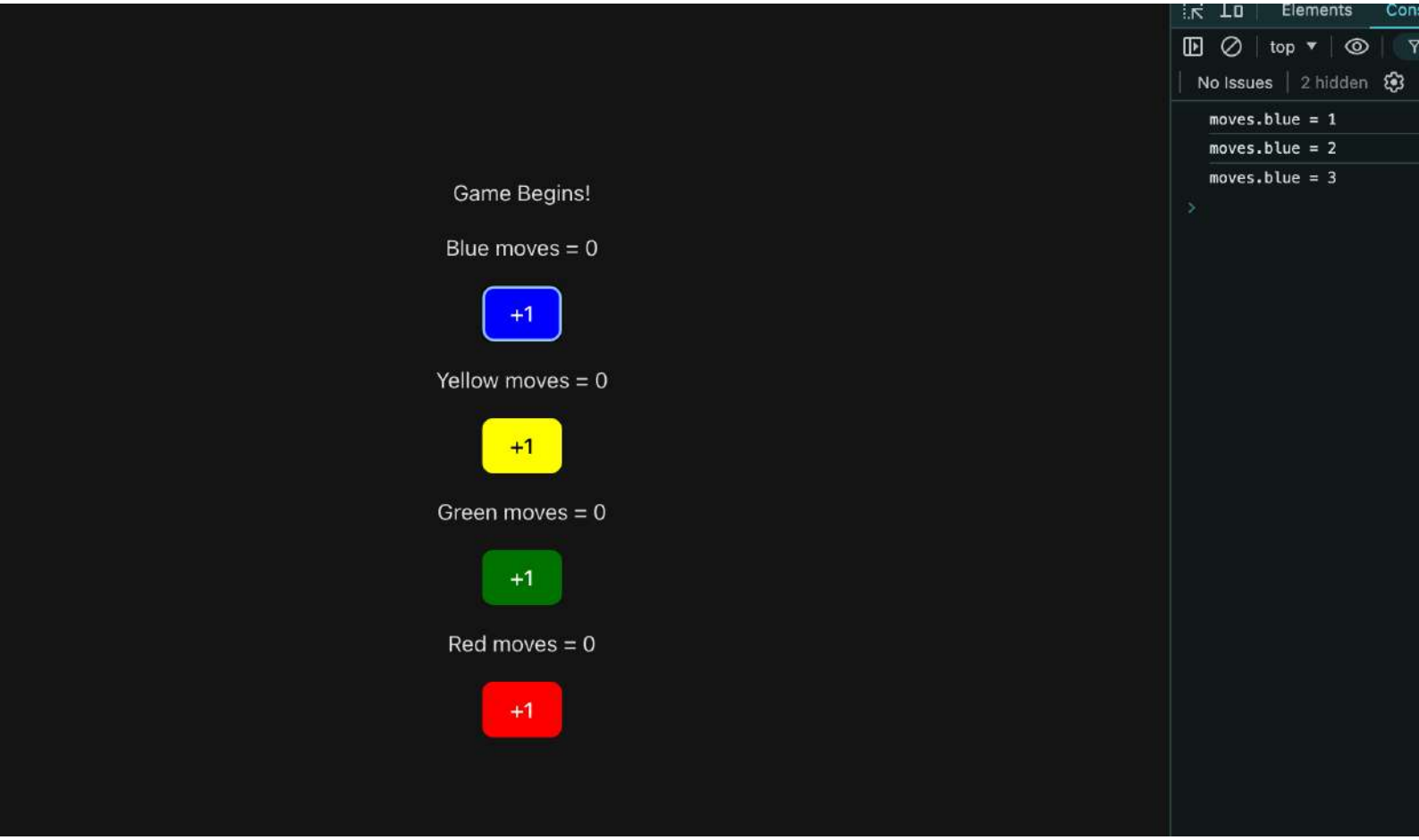
Ludo Board

```jsx
import {useState} from "react";
Complexity is 14 You must be kidding
export default function LudoBoard(){
    let [moves,setmoves]=useState({blue:0,yellow:0,green:0,red:0})

    let updateBlue =()=>{
        moves.blue++;
        console.log(`moves.blue = ${moves}`);
        setmoves(moves);
    }
    return(
        <div>
            <p>Game Begins!</p>
            <div className="Board">
                <p>Blue moves = {moves.blue}</p>
                <button style={{backgroundColor:"Blue"}} onClick={updateBlue}>+1</button>
                <p>Yellow moves = {moves.yellow}</p>
                <button style={{backgroundColor:"Yellow",color:"black"}}>+1</button>
                <p>Green moves = {moves.green}</p>
                <button style={{backgroundColor:"Green"}}>+1</button>
                <p>Red moves = {moves.red}</p>
                <button style={{backgroundColor:"Red"}}>+1</button>
            </div>
        </div>
    );
}
```


Game Begins!
Blue moves = 0
+1
Yellow moves = 0
+1
Green moves = 0
+1
Red moves = 0
+1
No Issues
2 hidden
2 moves.blue = [object Object]
Console
What's new
Highlights from the Chrome 126 update
Move tracks around with updated
Performance panel

FRONTEND > Apna College > React > React Lecture-4 > Objects-Array-State > src > LudoBoard.jsx > LudoBoard

```jsx
import {useState} from "react";
Complexity is 14 You must be kidding
export default function LudoBoard(){
    let [moves,setmoves]=useState({blue:0,yellow:0,green:0,red:0})

    let updateBlue =()=>{
        moves.blue++;
        console.log(`moves.blue = ${moves.blue}`);
        setmoves(moves);
    }
    return(
        <div>
            <p>Game Begins!</p>
            <div className="Board">
                <p>Blue moves = {moves.blue}</p>
                <button style={{backgroundColor:"Blue"}} onClick={updateBlue}>+1</button>
                <p>Yellow moves = {moves.yellow}</p>
                <button style={{backgroundColor:"Yellow",color:"black"}}>+1</button>
                <p>Green moves = {moves.green}</p>
                <button style={{backgroundColor:"Green"}}>+1</button>
                <p>Red moves = {moves.red}</p>
                <button style={{backgroundColor:"Red"}}>+1</button>
            </div>
        </div>
    );
}
```


Game Begins!
Blue moves = 0
+1
Yellow moves = 0
+1
Green moves = 0
+1
Red moves = 0
+1
Elements
top
No Issues
2 hidden
moves.blue = 1
moves.blue = 2
moves.blue = 3

Objects & State

Ludo Board

Objects { a: 1, b: 2 / 4 }

Arrays are [1, 2, 3 / 4]

Blue moves : 0

+1

Yellow moves : 0

+1

Green moves : 0

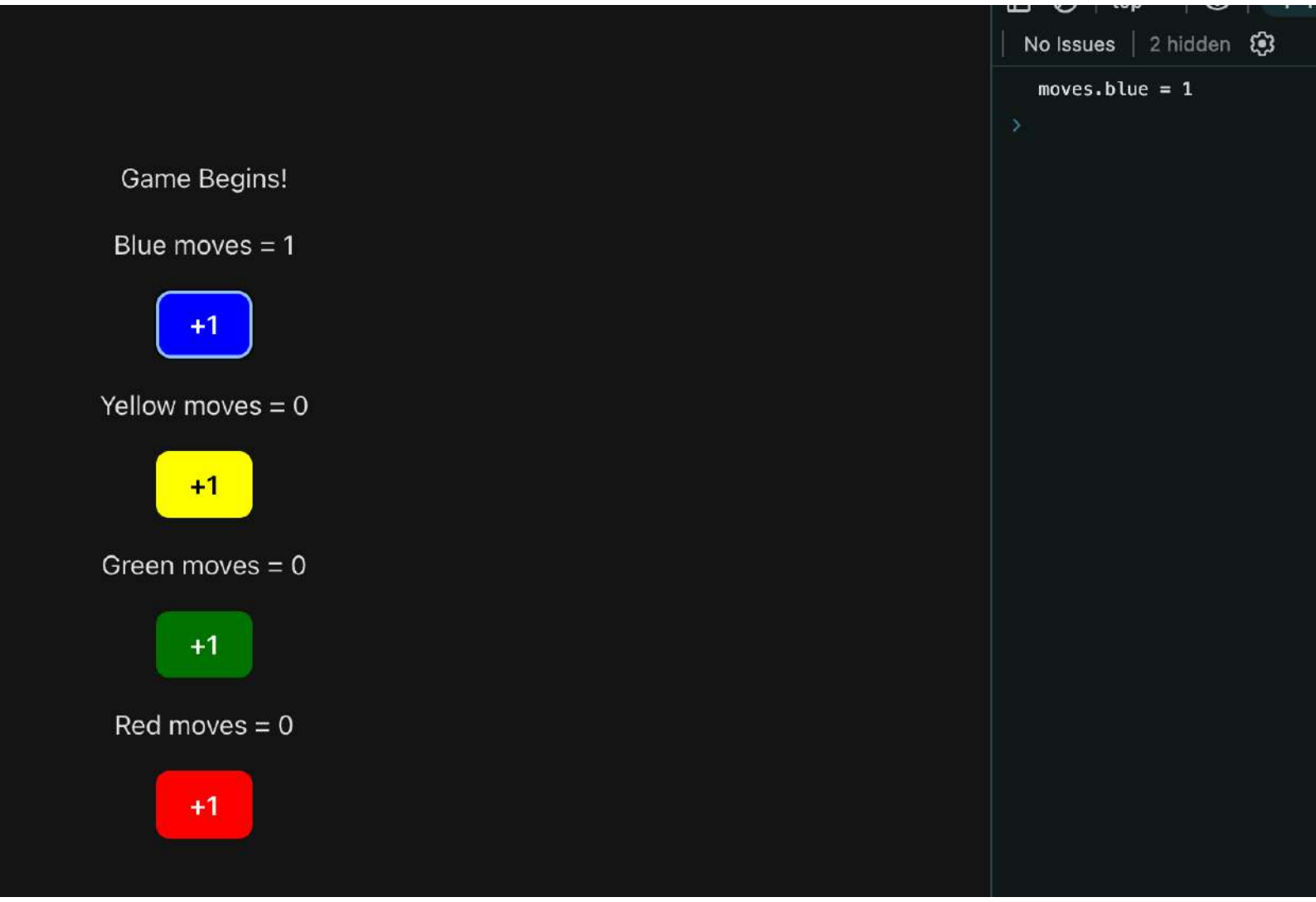
+1

Red moves : 0

+1

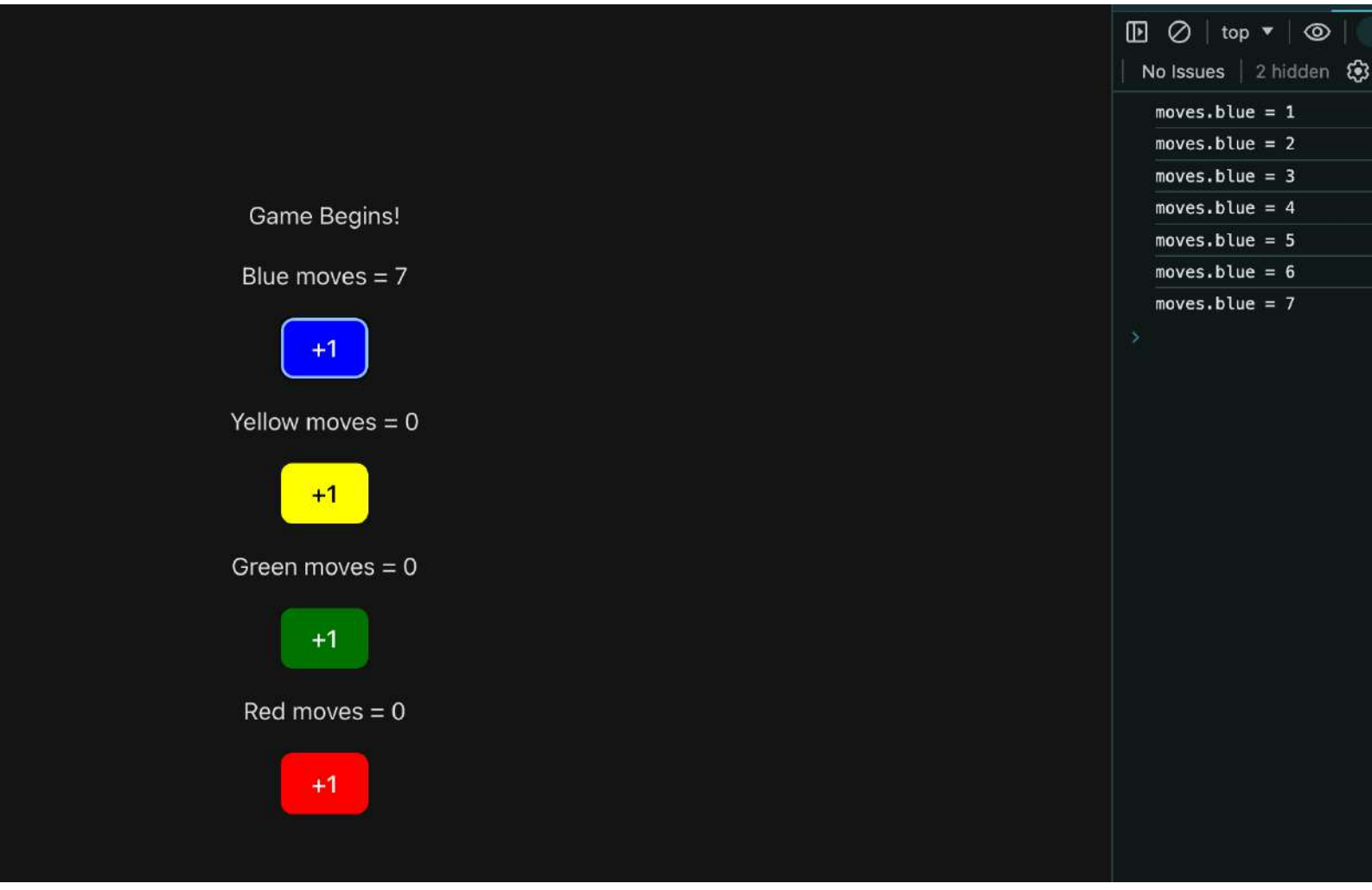
> Apna College > React > React Lecture-4 > Objects-Array-State > src > LudoBoard.jsx > LudoBoard > upda

```jsx
import {useState} from "react";
Complexity is 14 You must be kidding
export default function LudoBoard(){
    let [moves,setmoves]=useState({blue:0,yellow:0,green:0,red:0})

    let updateBlue =()=>{
        moves.blue++;
        console.log(`moves.blue = ${moves.blue}`);
        setmoves({...moves});
    }
    return(
        <div>
            <p>Game Begins!</p>
            <div className="Board">
                <p>Blue moves = {moves.blue}</p>
                <button style={{backgroundColor:"Blue"}} onClick={updateBlue}>+1</button>
                <p>Yellow moves = {moves.yellow}</p>
                <button style={{backgroundColor:"Yellow",color:"black"}}>+1</button>
                <p>Green moves = {moves.green}</p>
                <button style={{backgroundColor:"Green"}}>+1</button>
                <p>Red moves = {moves.red}</p>
                <button style={{backgroundColor:"Red"}}>+1</button>
            </div>
        </div>
    );
}
```


Game Begins!
Blue moves = 1
+1
Yellow moves = 0
+1
Green moves = 0
+1
Red moves = 0
+1
No Issues
2 hidden
moves.blue = 1

Learning > FRONTEND > Apna College > React > React Lecture-4 > Objects-Array-State > src > LudoBoard.jsx

```jsx
import {useState} from "react";
Complexity is 14 You must be kidding
export default function LudoBoard(){
    let [moves,setmoves]=useState({blue:0,yellow:0,green:0,red:0})

    let updateBlue =()=>{
        setmoves({...moves,blue:moves.blue+=1});
        console.log(`moves.blue = ${moves.blue}`);
    }
    return(
        <div>
            <p>Game Begins!</p>
            <div className="Board">
                <p>Blue moves = {moves.blue}</p>
                <button style={{backgroundColor:"Blue"}} onClick={updateBlue}>+1</button>
                <p>Yellow moves = {moves.yellow}</p>
                <button style={{backgroundColor:"Yellow",color:"black"}}>+1</button>
                <p>Green moves = {moves.green}</p>
                <button style={{backgroundColor:"Green"}}>+1</button>
                <p>Red moves = {moves.red}</p>
                <button style={{backgroundColor:"Red"}}>+1</button>
            </div>
        </div>
    );
}
```


Game Begins!
Blue moves = 7
+1
Yellow moves = 0
+1
Green moves = 0
+1
Red moves = 0
+1
top
No Issues
2 hidden
moves.blue = 1
moves.blue = 2
moves.blue = 3
moves.blue = 4
moves.blue = 5
moves.blue = 6
moves.blue = 7

```
import {useState} from "react";
Complexity is 25 You must be kidding
export default function LudoBoard(){
    let [moves,setmoves]=useState({blue:0,yellow:0,green:0,red:0})

    Complexity is 3 Everything is cool!
    let updateBlue =()=>{
        setmoves((prevMove)=>{
            return {...prevMove,blue:prevMove.blue+1};
        });
        // console.log(`moves.blue = ${moves.blue}`);
    }
    Complexity is 3 Everything is cool!
    let updateYellow =()=>{
        setmoves((prevMove)=>{
            return {...prevMove,yellow:prevMove.yellow+1};
        });
        // console.log(`moves.yellow = ${moves.yellow}`);
    }
    Complexity is 3 Everything is cool!
    let updateGreen =()=>{
        setmoves((prevMove)=>{
            return {...prevMove,green:prevMove.green+1};
        });
        // console.log(`moves.green = ${moves.green}`);
    }
    Complexity is 3 Everything is cool!
    let updateRed =()=>{
        setmoves((prevMove)=>{
            return {...prevMove,red:prevMove.red+1};
        });
        // console.log(`moves.red = ${moves.red}`);
    }
```

```
    return(
        <div>
            <p>Game Begins!</p>
            <div className="Board">
                <p>Blue moves = {moves.blue}</p>
                <button style={{backgroundColor:"Blue"}} onClick={updateBlue}>+1</button>
                <p>Yellow moves = {moves.yellow}</p>
                <button style={{backgroundColor:"Yellow",color:"black"}} onClick={updateYellow}>+1</button>
                <p>Green moves = {moves.green}</p>
                <button style={{backgroundColor:"Green"}} onClick={updateGreen}>+1</button>
                <p>Red moves = {moves.red}</p>
                <button style={{backgroundColor:"Red"}} onClick={updateRed}>+1</button>
            </div>
        </div>
    );
}
```

Game Begins!

Blue moves = 6

+1

Yellow moves = 3

+1

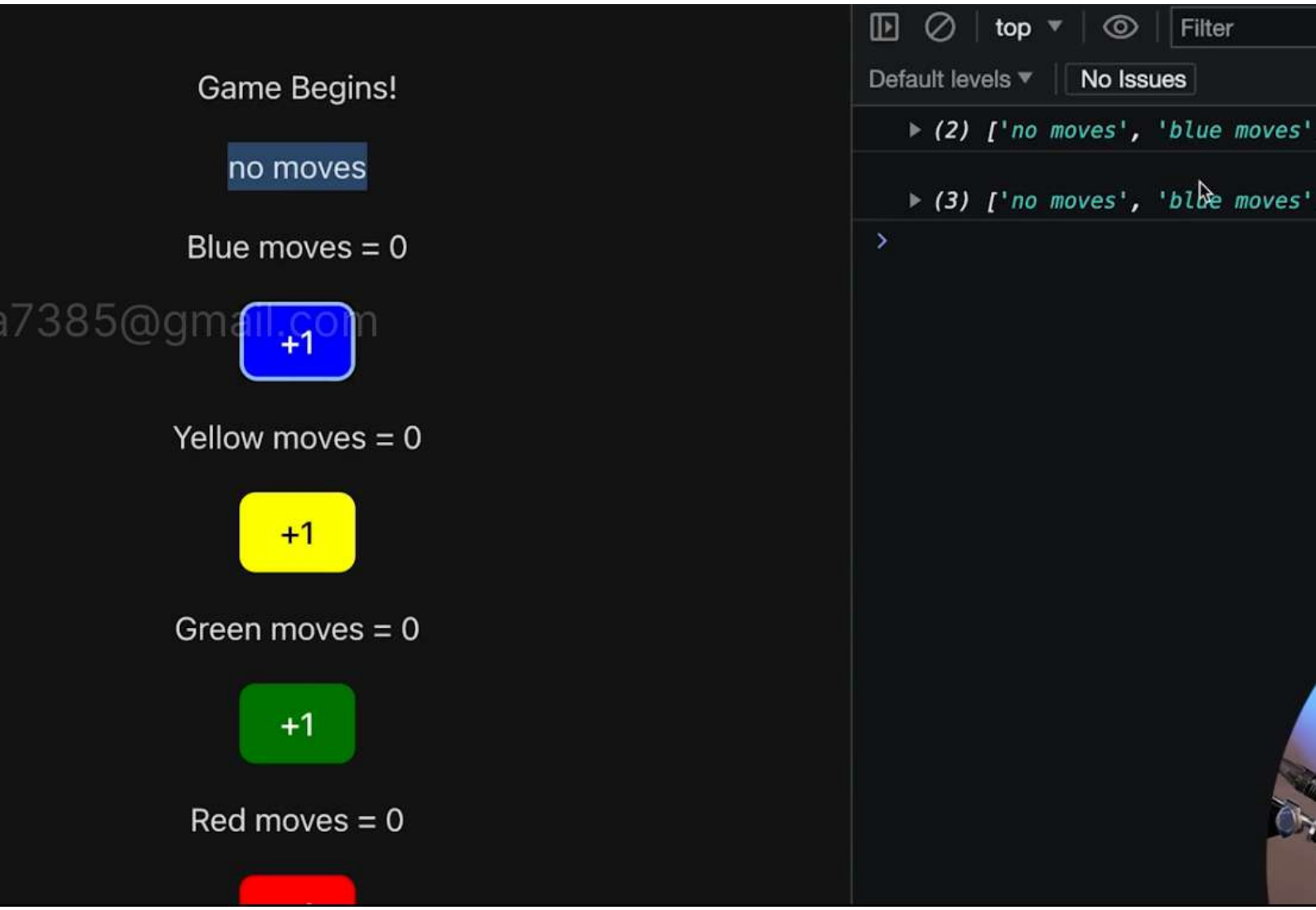
Green moves = 4

+1

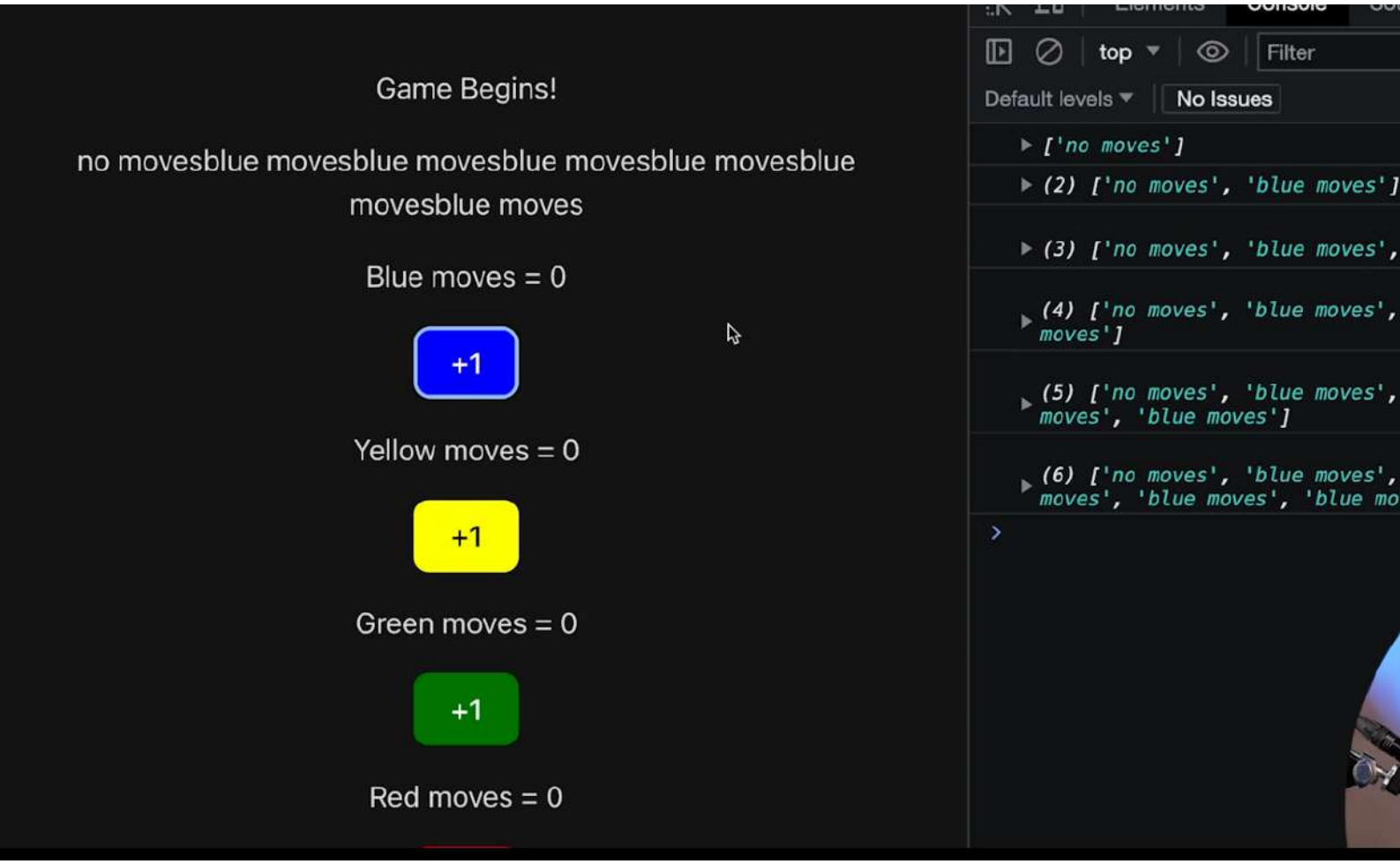
Red moves = 5

+1


src > LudoBoard.jsx > LudoBoard > updateBlue
4 let [moves, setMoves] = useState({ blue: 0, red: 0, yellow: 0, green: 0 });
5 let [arr, SetArr] = useState(["no moves"]);
6
7 let updateBlue = () => {
8 // setMoves((prevMoves) => {
9 // return { ...prevMoves, blue: prevMoves.blue + 1 };
10 // });
11
12 arr.push("blue moves");
13 SetArr(arr);
14 console.log(arr);
15 };
16
17 let updateYellow = () => {
18 setMoves((prevMoves) => {
19 return { ...prevMoves, yellow: prevMoves.yellow + 1 };
20 });
21 };


Game Begins!
no moves
Blue moves = 0
+1
Yellow moves = 0
+1
Green moves = 0
+1
Red moves = 0
top
Filter
Default levels
No Issues
(2) ['no moves', 'blue moves'
(3) ['no moves', 'blue moves'

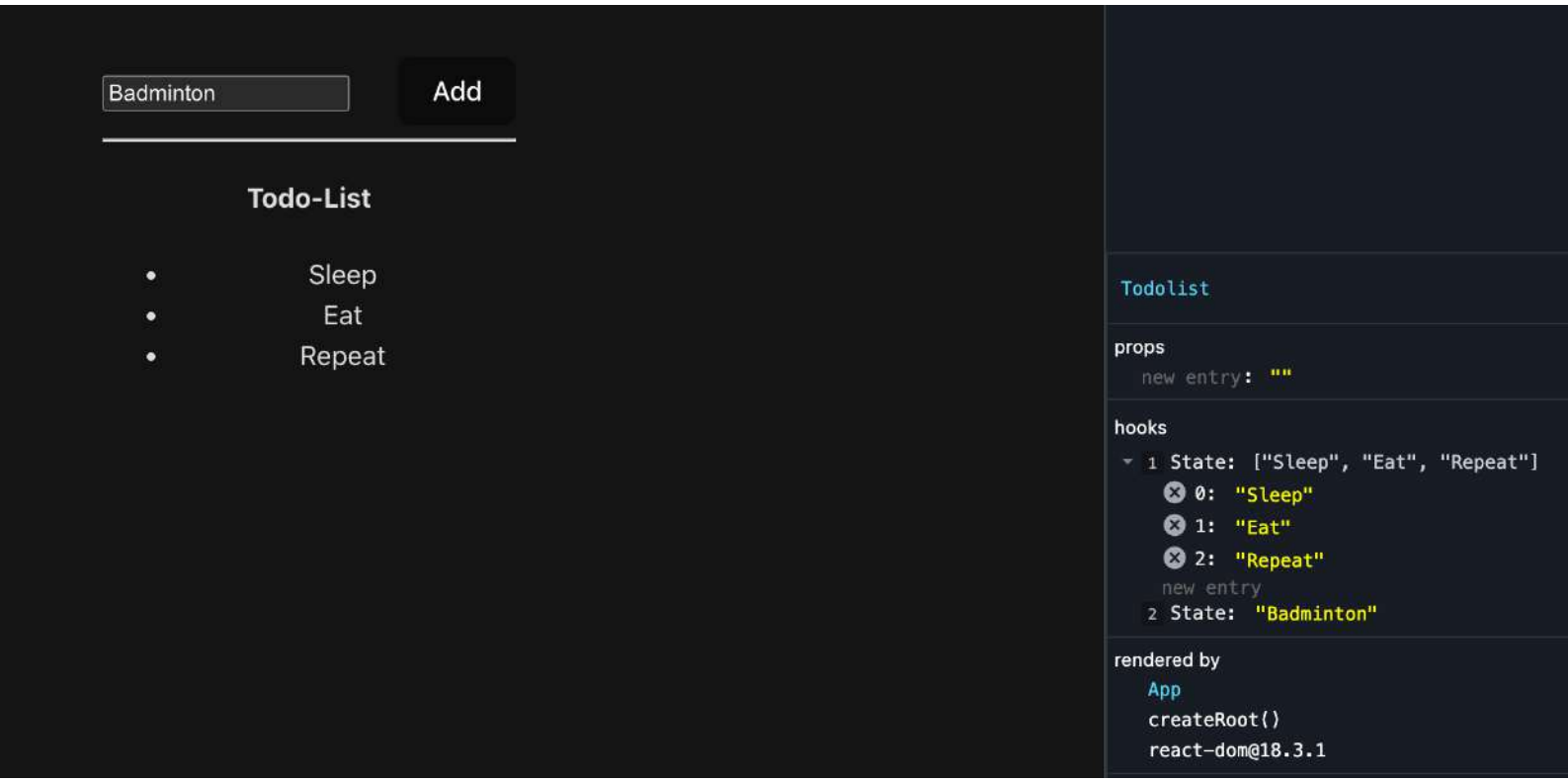
```
let updateBlue = () => {
  // setMoves((prevMoves) => {
  //   return { ...prevMoves, blue: prevMoves.blue + 1 };
  // });

  SetArr((prevArr) => {
    return [...prevArr, "blue moves"];
  });
  console.log(arr);
};

let updateYellow = () => {
  setMoves((prevMoves) => {
    return { ...prevMoves, yellow: prevMoves.yellow + 1 };
  });
};
```


Game Begins!
no movesblue movesblue movesblue movesblue movesblue movesblue moves
Blue moves = 0
+1
Yellow moves = 0
+1
Green moves = 0
+1
Red moves = 0
top
Filter
Default levels
No Issues
['no moves']
(2) ['no moves', 'blue moves']
(3) ['no moves', 'blue moves',
(4) ['no moves', 'blue moves', moves']
(5) ['no moves', 'blue moves', moves', 'blue moves']
(6) ['no moves', 'blue moves', moves', 'blue moves', 'blue mo

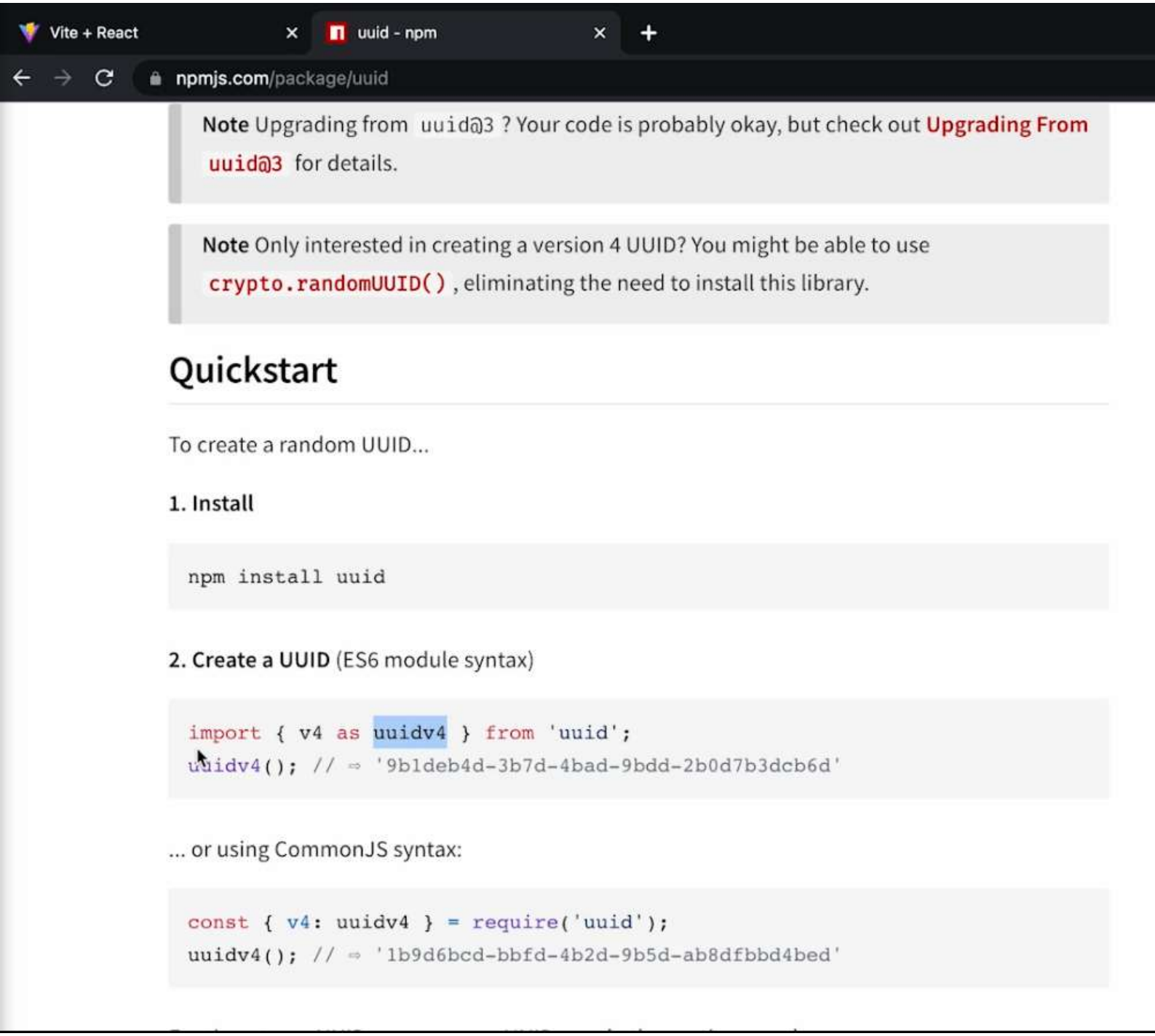
App.jsx U Todolist.jsx U

Learning > FRONTEND > Apna College > React > React Lecture-4 > Objects-Array-State > src > Todolist.jsx > Todolist

```jsx
import { useState } from "react"

Complexity is 13 You must be kidding
export default function Todolist(){
    let [list,setlist]=useState(["Eat"]);
    let [newlist,setnewlist]=useState("");

    let addNewTask = () =>{
        setlist([...list,newlist]);
        setnewlist("");
    }
    let updatelistvalue = (event)=>{
        setnewlist(event.target.value);
    }

    return(
    <div>
        <input type="text" placeholder="Enter the task" value={newlist} onChange={updatelistvalue}/>
               
        <button onClick={addNewTask}>Add</button>
        <hr />
        <h4>Todo-List</h4>
        <ul>
            {
                Complexity is 3 Everything is cool!
                list.map((list)=>{return(<li>{list}</li>)})
            }
        </ul>
    </div>
    );
}
```


Badminton
Add
Todo-List
Sleep
Eat
Repeat
Todolist
props
new entry: ""
hooks
1 State: ["Sleep", "Eat", "Repeat"]
0: "Sleep"
1: "Eat"
2: "Repeat"
new entry
2 State: "Badminton"
rendered by
App
createRoot()
react-dom@18.3.1

Unique Key for List Items


Vite + React
uuid - npm
npmjs.com/package/uuid
Note Upgrading from uuid@3 ? Your code is probably okay, but check out Upgrading From uuid@3 for details.
Note Only interested in creating a version 4 UUID? You might be able to use crypto.randomUUID() , eliminating the need to install this library.
Quickstart
To create a random UUID...
1. Install
npm install uuid
2. Create a UUID (ES6 module syntax)
import { v4 as uuidv4 } from 'uuid';
uuidv4(); // ⇨ '9b1deb4d-3b7d-4bad-9bdd-2b0d7b3dcb6d'
... or using CommonJS syntax:
const { v4: uuidv4 } = require('uuid');
uuidv4(); // ⇨ '1b9d6bcd-bbfd-4b2d-9b5d-ab8dfbbd4bed'

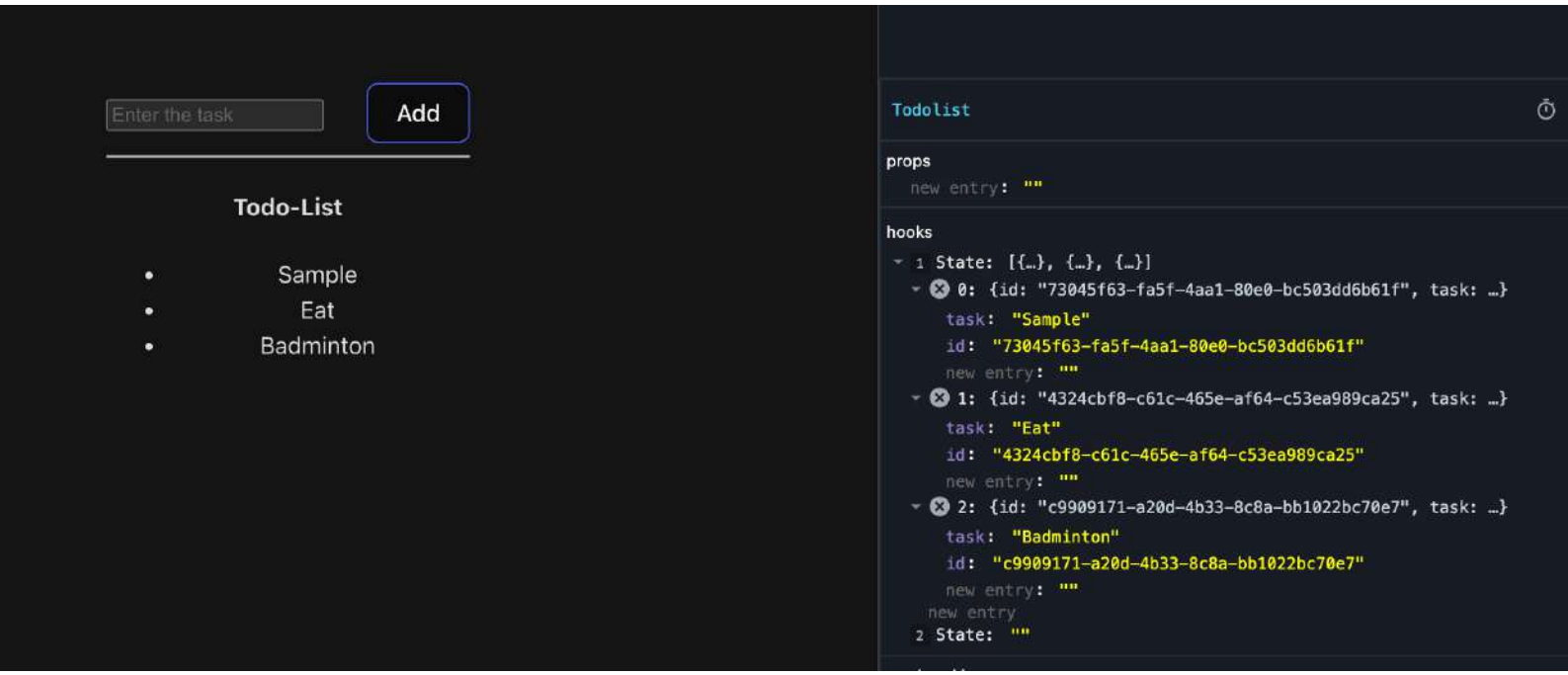
Unique Key for List Items


export default function TodoList() {
let [todos, setTodos] = useState({ task: "sample-task", id: uuidv4() });
let [newTodo, setNewTodo] = useState("");
let addNewTask = () => {
setTodos([...todos, newTodo]);
setNewTodo("");
};
let updateTodoValue = (event) => {
setNewTodo(event.target.value);
};

App.jsx U Todolist.jsx U

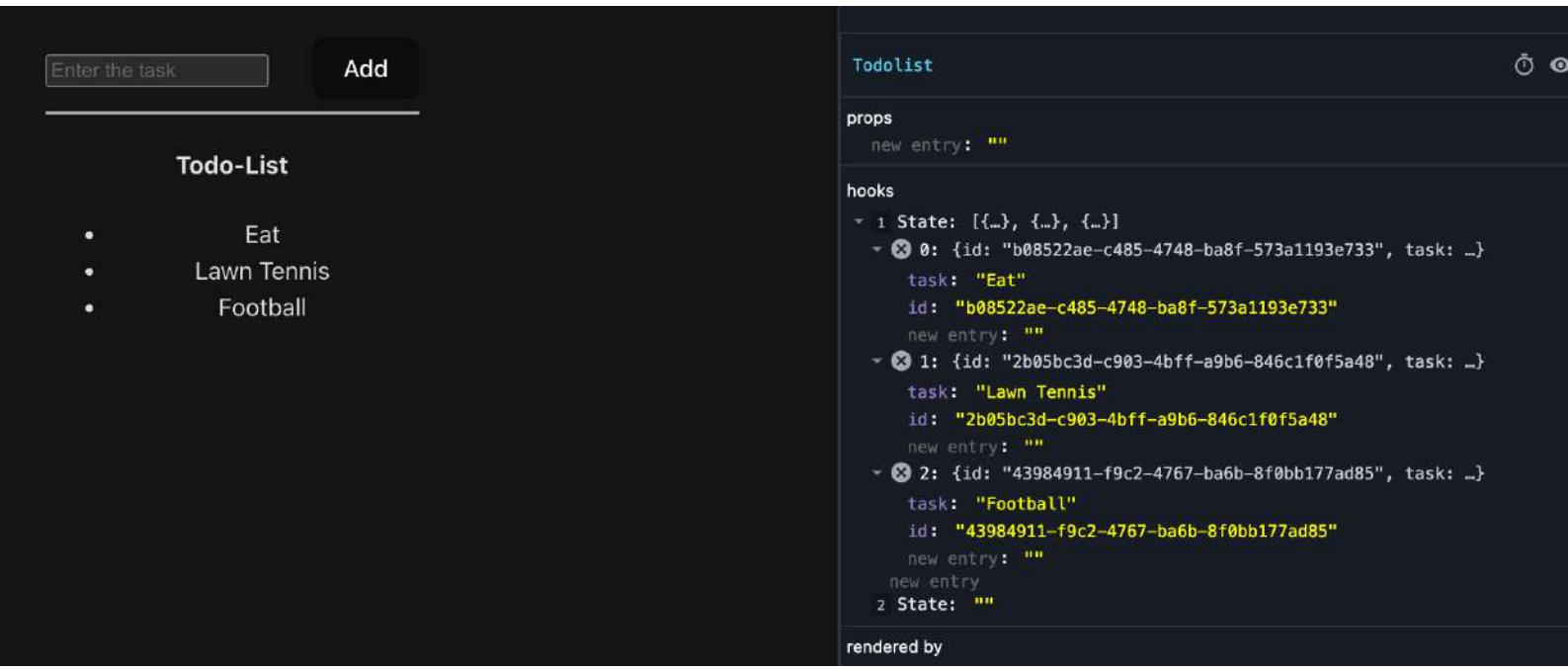
Learning > FRONTEND > Apna College > React > React Lecture-4 > Objects-Array-State > src > Todolist.jsx > Todolist > addNewT

```jsx
import { useState } from "react"
import { v4 as uuidv4} from 'uuid';

Complexity is 12 You must be kidding
export default function Todolist(){
    let [list,setlist]=useState([{task:"Sample",id:uuidv4()}]);
    let [newlist,setnewlist]=useState("");

    let addNewTask = () =>{
        setlist([...list,{task:newlist,id:uuidv4()}]);
        setnewlist("");
    }
    let updatelistvalue = (event)=>{
        setnewlist(event.target.value);
    }

    return(
    <div>
        <input type="text" placeholder="Enter the task" value={newlist} onChange={updatelistvalue}/>
               
        <button onClick={addNewTask}>Add</button>
        <hr />
        <h4>Todo-List</h4>
        <ul>
            {
                list.map((list)=>((<li key={list.id}>{list.task}</li>)))
            }
        </ul>
    </div>
    );
}
```


Enter the task
Add
Todo-List
Sample
Eat
Badminton
Todolist
props
new entry: ""
hooks
1 State: [{…}, {…}, {…}]
0: {id: "73045f63-fa5f-4aa1-80e0-bc503dd6b61f", task: …}
task: "Sample"
id: "73045f63-fa5f-4aa1-80e0-bc503dd6b61f"
new entry: ""
1: {id: "4324cbf8-c61c-465e-af64-c53ea989ca25", task: …}
task: "Eat"
id: "4324cbf8-c61c-465e-af64-c53ea989ca25"
new entry: ""
2: {id: "c9909171-a20d-4b33-8c8a-bb1022bc70e7", task: …}
task: "Badminton"
id: "c9909171-a20d-4b33-8c8a-bb1022bc70e7"
new entry: ""
new entry
2 State: ""

Learning > FRONTEND > Apna College > React > React Lecture-4 > Objects-Array-State > src > Todolist.jsx > Todolist

```jsx
import { useState } from "react"
import { v4 as uuidv4} from 'uuid';

Complexity is 12 You must be kidding
export default function Todolist(){
    let [list,setlist]=useState([]);
    let [newlist,setnewlist]=useState("");

    let addNewTask = () =>{
        setlist([...list,{task:newlist,id:uuidv4()}]);
        setnewlist("");
    }
    let updatelistvalue = (event)=>{
        setnewlist(event.target.value);
    }

    return(
    <div>
        <input type="text" placeholder="Enter the task" value={newlist} onChange={updatelistvalue}/>
               
        <button onClick={addNewTask}>Add</button>
        <hr />
        <h4>Todo-List</h4>
        <ul>
            {
                list.map((list)=>((<li key={list.id}>{list.task}</li>)))
            }
        </ul>
    </div>
    );
}
```


Enter the task
Add
Todo-List
Eat
Lawn Tennis
Football
Todolist
props
new entry: ""
hooks
1 State: [{…}, {…}, {…}]
0: {id: "b08522ae-c485-4748-ba8f-573a1193e733", task: …}
task: "Eat"
id: "b08522ae-c485-4748-ba8f-573a1193e733"
new entry: ""
1: {id: "2b05bc3d-c903-4bff-a9b6-846c1f0f5a48", task: …}
task: "Lawn Tennis"
id: "2b05bc3d-c903-4bff-a9b6-846c1f0f5a48"
new entry: ""
2: {id: "43984911-f9c2-4767-ba6b-8f0bb177ad85", task: …}
task: "Football"
id: "43984911-f9c2-4767-ba6b-8f0bb177ad85"
new entry: ""
new entry
2 State: ""
rendered by

	avoid (mutates the array)	prefer (returns a new array)
adding	`push`, `unshift`	`concat`, `[...arr]` spread syntax (example)
removing	`pop`, `shift`, `splice`	`filter`, `slice` (example)
replacing	`splice`, `arr[i] = ...` assignment	`map` (example)
sorting	`reverse`, `sort`	copy the array first (example)

Updating in Arrays

Updating All Elements in Array

```
let UpperCaseAll = () => {
  setTasks((prevTasks) =>
    prevTasks.map((todo) => {
      return {
        ...todo,
        task: todo.task.toUpperCase(),
      };
    })
  );
};
```

Updating One Element in Array

```
let upperCaseOne = (id) => {
  setTasks((prevTasks) =>
    prevTasks.map((todo) => {
      if (todo.id === id) {
        return {
          ...todo,
          task: todo.task.toUpperCase(),
        };
      } else {
        return todo;
      }
    })
  );
};
```

Updating in Arrays

Updating All Elements in Array

```
let UpperCaseAll = () => {
  setTasks((prevTasks) =>
    prevTasks.map((todo) => {
      return {
        ...todo,
        task: todo.task.toUpperCase(),
      };
    })
  );
};
```

Updating One Element in Array

```
let upperCaseOne = (id) => {
  setTasks((prevTasks) =>
    prevTasks.map((todo) => {
      if (todo.id === id) {
        return {
          ...todo,
          task: todo.task.toUpperCase(),
        };
      } else {
        return todo;
      }
    })
  );
};
```

Changing Todo

Implement "Done Task" feature to the todo

todo : { task :
id :
isDone :
}

Learning > FRONTEND > Apna College > React > React Lecture-4 > Objects-Array-State > src >

```jsx
import { useState } from "react"
import { v4 as uuidv4} from 'uuid';

Complexity is 34 Bloody hell...
export default function Todolist(){
    let [list,setlist]=useState([]);
    let [newlist,setnewlist]=useState("");

    Complexity is 3 Everything is cool!
    let addNewTask = () =>{
        setlist((prevlist)=>{
            return([...prevlist,{task:newlist,id:uuidv4()}])
        });
        setnewlist("");
    }
    let updatelistvalue = (event)=>{
        setnewlist(event.target.value);
    }

    Complexity is 3 Everything is cool!
    let deletelistitem =(id)=>{
        setlist((prevlist)=>list.filter((prevlist)=> prevlist.id!=id))

    }

```

Complexity is 4 Everything is cool!

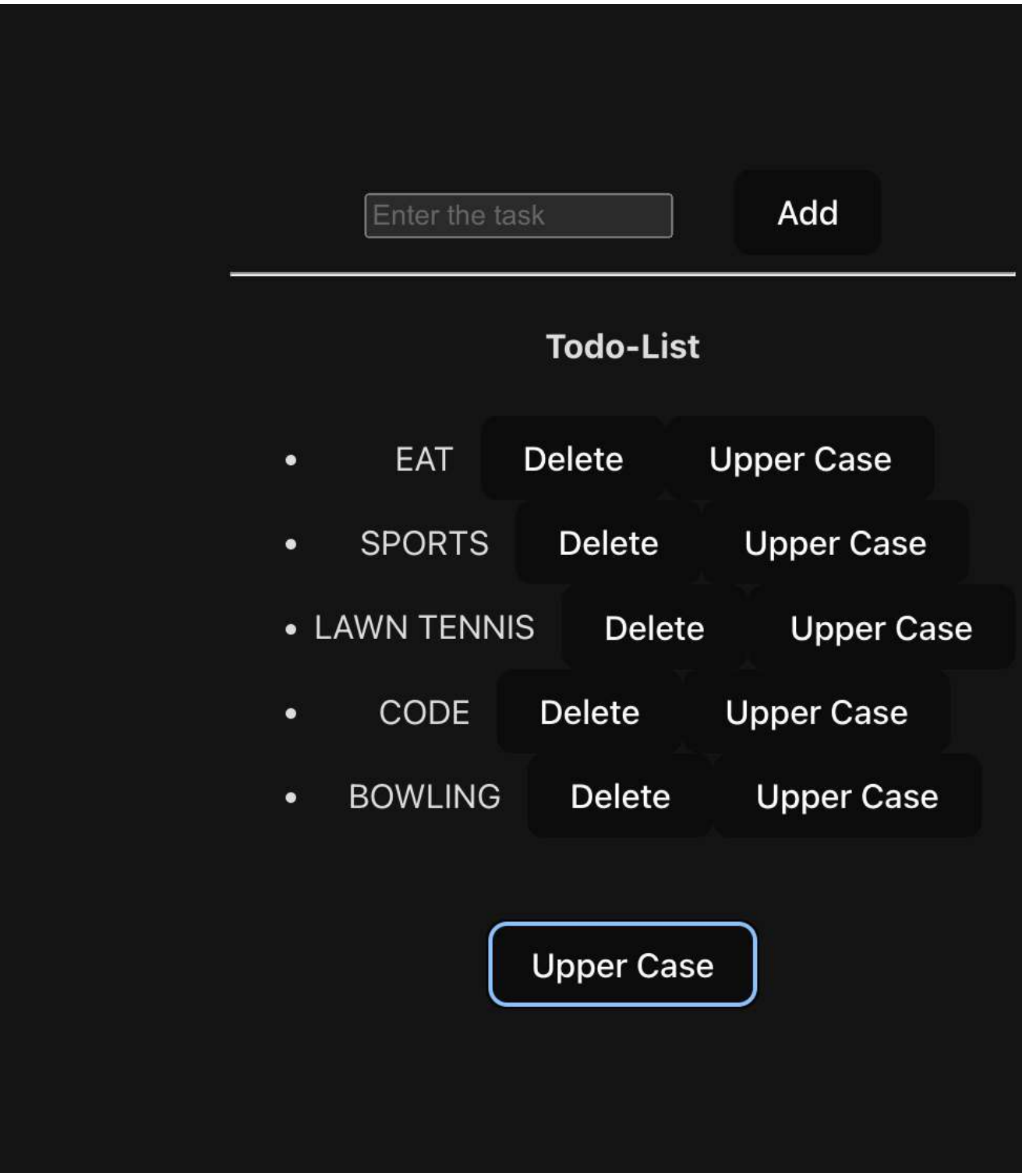
```
let uppercase = () => {
    Complexity is 3 Everything is cool!
    setlist((prevlist) =>
        prevlist.map((list)=>{
        return {
            ...list,
            task: list.task.toUpperCase()
        };
    })
  );
};
```

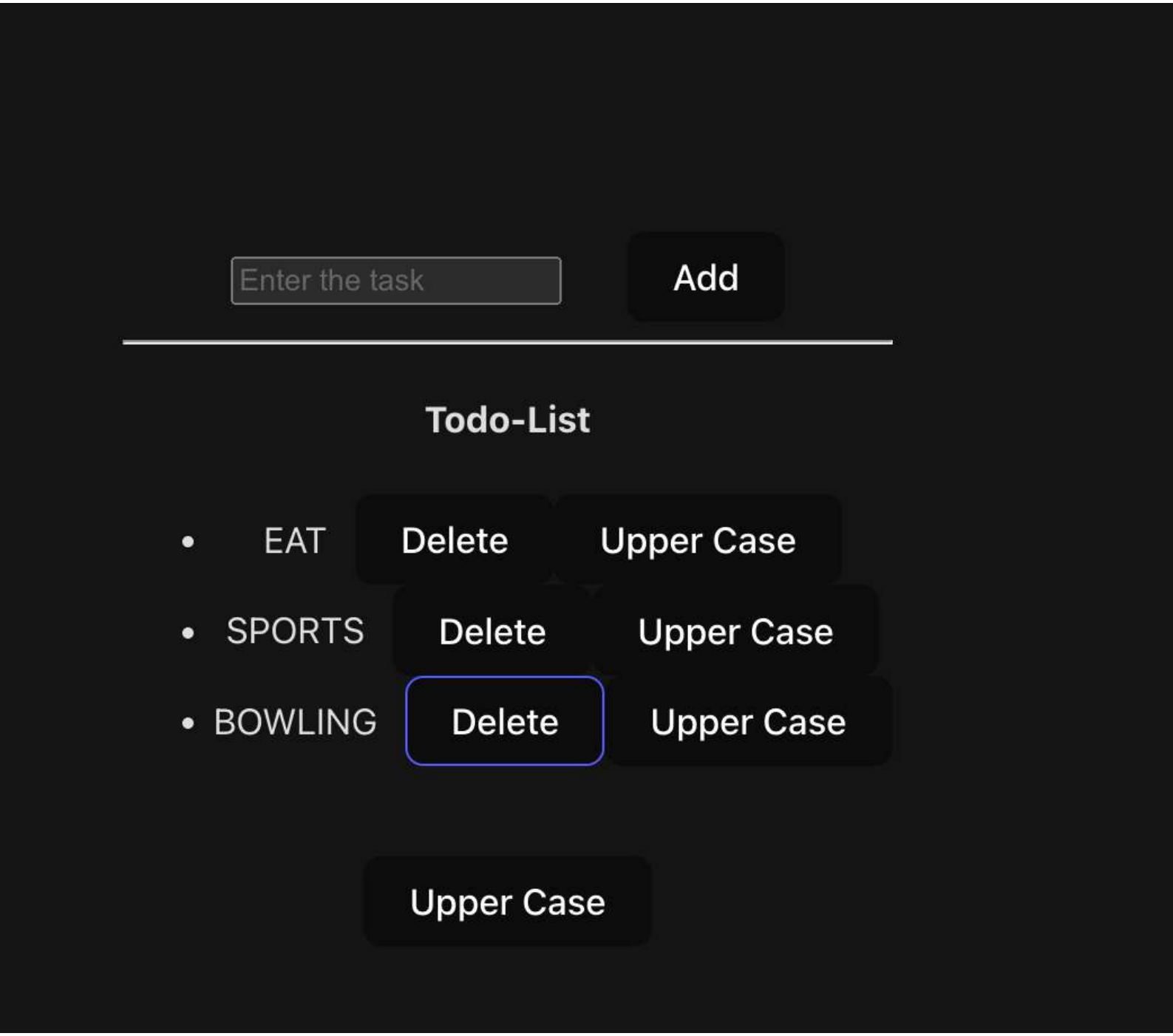
Complexity is 6 It's time to do something...

```
let uppercaseOne = (id) => {
    Complexity is 5 Everything is cool!
    setlist((prevlist) =>
        Complexity is 4 Everything is cool!
        prevlist.map((list)=>{
        if(list.id==id){
            return {
                ...list,
                task: list.task.toUpperCase()
            };
        } else {
            return list;
        }
    })
  );
};
```

```
return(
<div>
    <input type="text" placeholder="Enter the task" value={newlist} onChange={updatelistvalue}/>
           
    <button onClick={addNewTask}>Add</button>
    <hr />
    <h4>Todo-List</h4>
    <ul>
        {
            Complexity is 7 It's time to do something...
            list.map((list)=>(( 
            <li key={list.id}>
                <span>{list.task}</span>
                   
                <button onClick={()=> deletelistitem(list.id)}>Delete</button>
                <button onClick={()=> uppercaseOne(list.id)}>Upper Case</button>
            </li>
            ))
        }

    </ul>
    <br />
    <button onClick={uppercase}>Upper Case</button>
</div>
);
```


Enter the task
Add
Todo-List
eat
Delete
Upper Case
SPORTS
Delete
Upper Case
lawn tennis
Delete
Upper Case
CODE
Delete
Upper Case
bowling
Delete
Upper Case
Upper Case


Enter the task
Add
Todo-List
EAT
Delete
Upper Case
SPORTS
Delete
Upper Case
LAWN TENNIS
Delete
Upper Case
CODE
Delete
Upper Case
BOWLING
Delete
Upper Case
Upper Case


Enter the task
Add
Todo-List
EAT
Delete
Upper Case
SPORTS
Delete
Upper Case
BOWLING
Delete
Upper Case
Upper Case