Lab 3
Component based Game

Software Studio

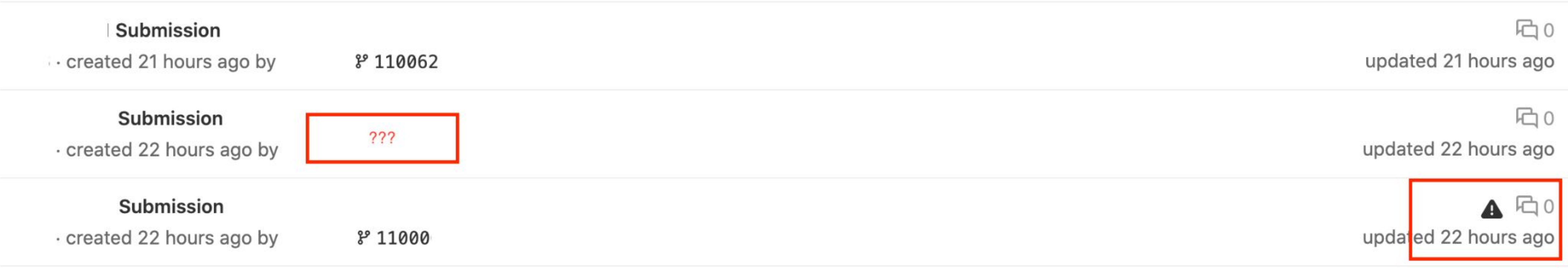
DataLab, CS, NTHU

2023 spring

HW

Fork from master and Clone from your branch

Don’t merge into master branch



Submission

· created 21 hours ago by 110062

0

updated 21 hours ago

Submission

· created 22 hours ago by

???

0

updated 22 hours ago

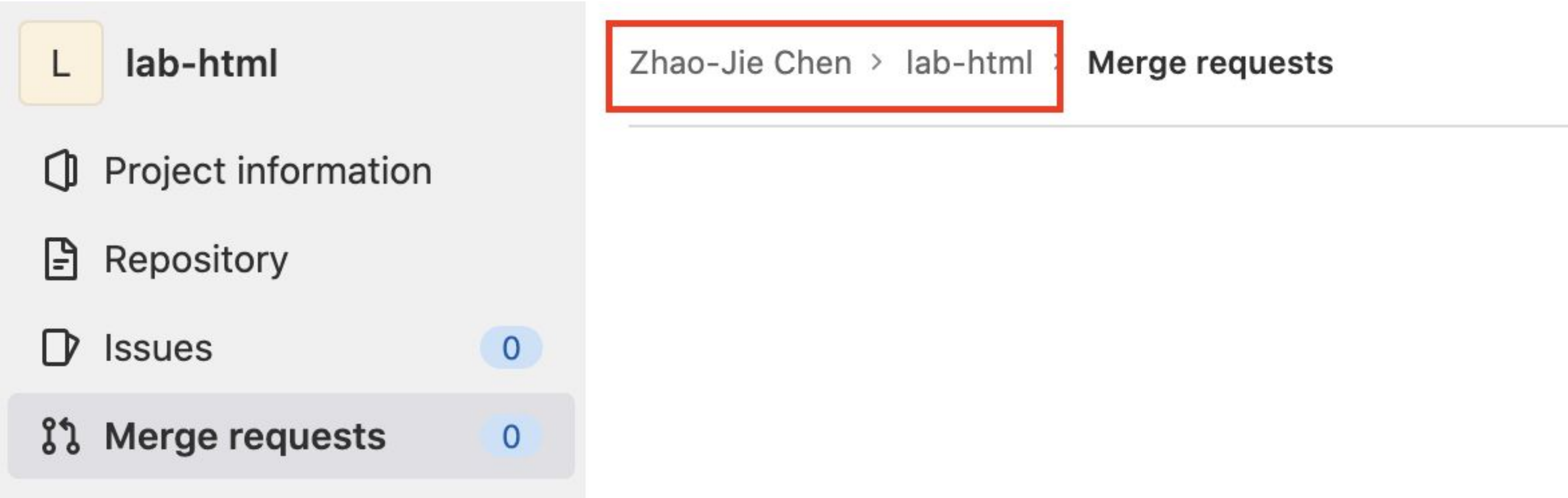
Submission

· created 22 hours ago by 11000

⚠ 0

updated 22 hours ago

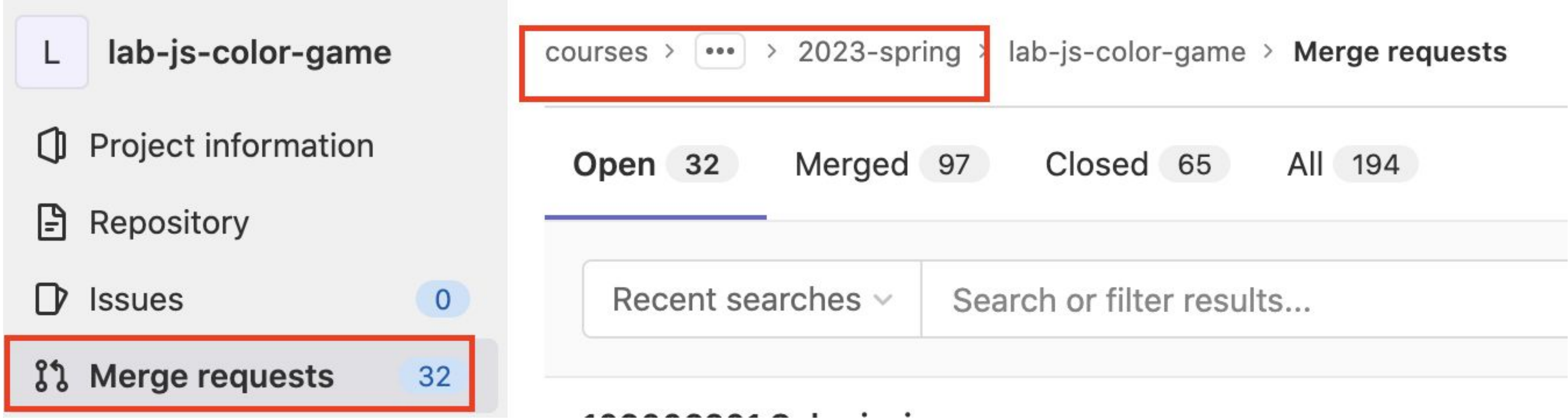
Hand in successfully



L lab-html

- Project information
- Repository
- Issues 0
- Merge requests 0

Zhao-Jie Chen › lab-html › Merge requests



L lab-js-color-game

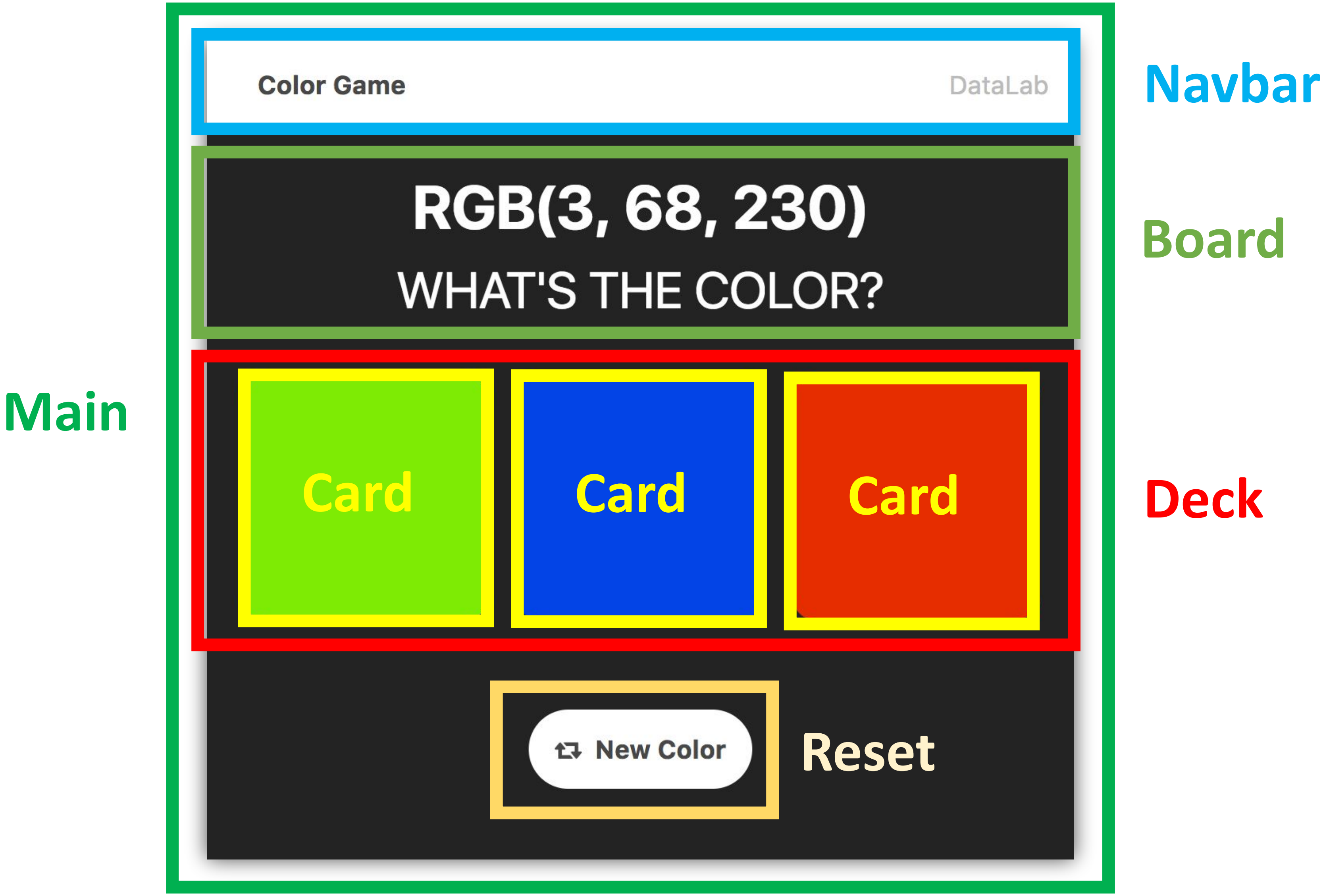
- Project information
- Repository
- Issues 0
- Merge requests 32

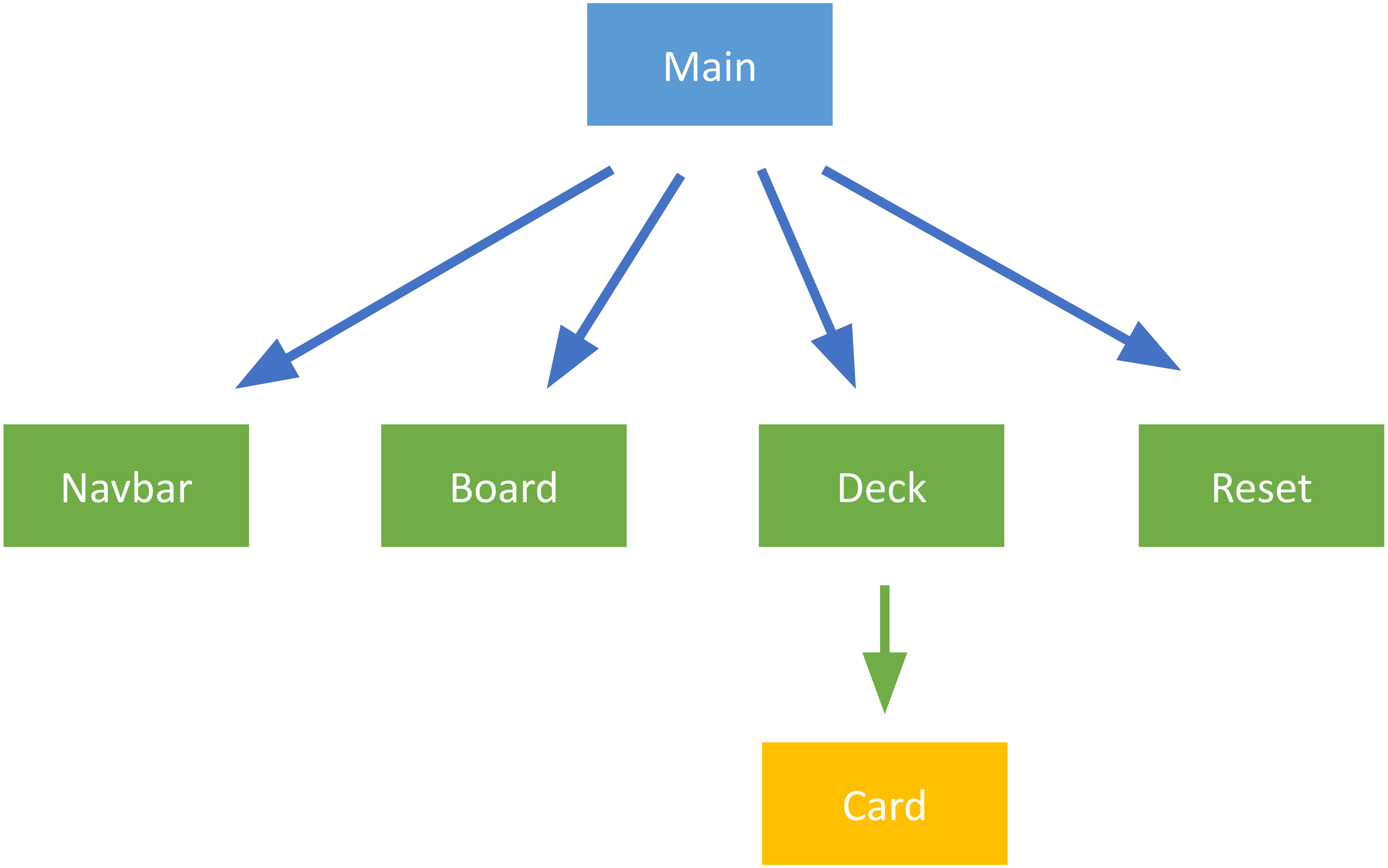
courses › ••• › 2023-spring › lab-js-color-game › Merge requests

Open 32 Merged 97 Closed 65 All 194

Recent searches ⌄ Search or filter results...

Color Game component





Main

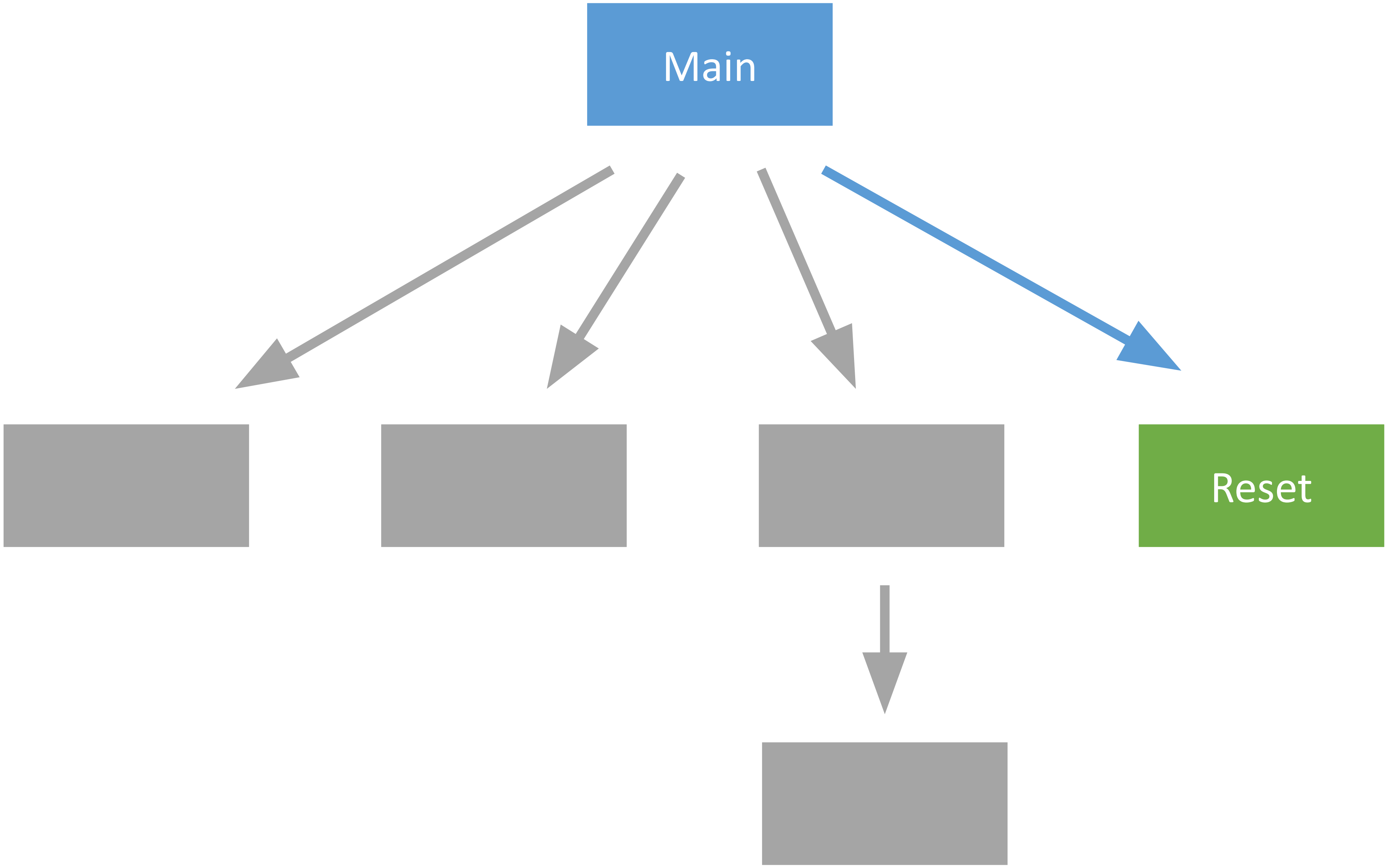
Navbar

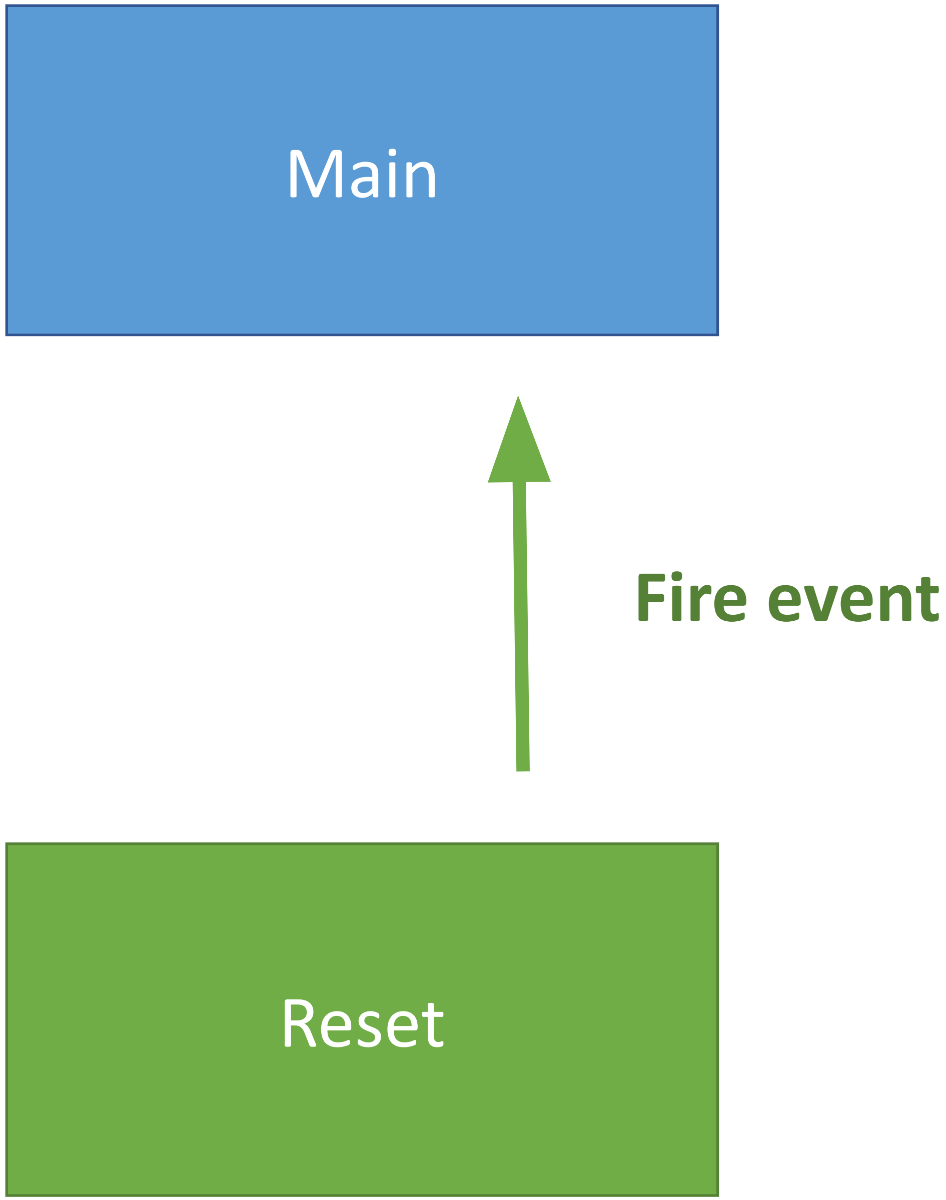
Board

Deck

Reset

Card


Main
Reset

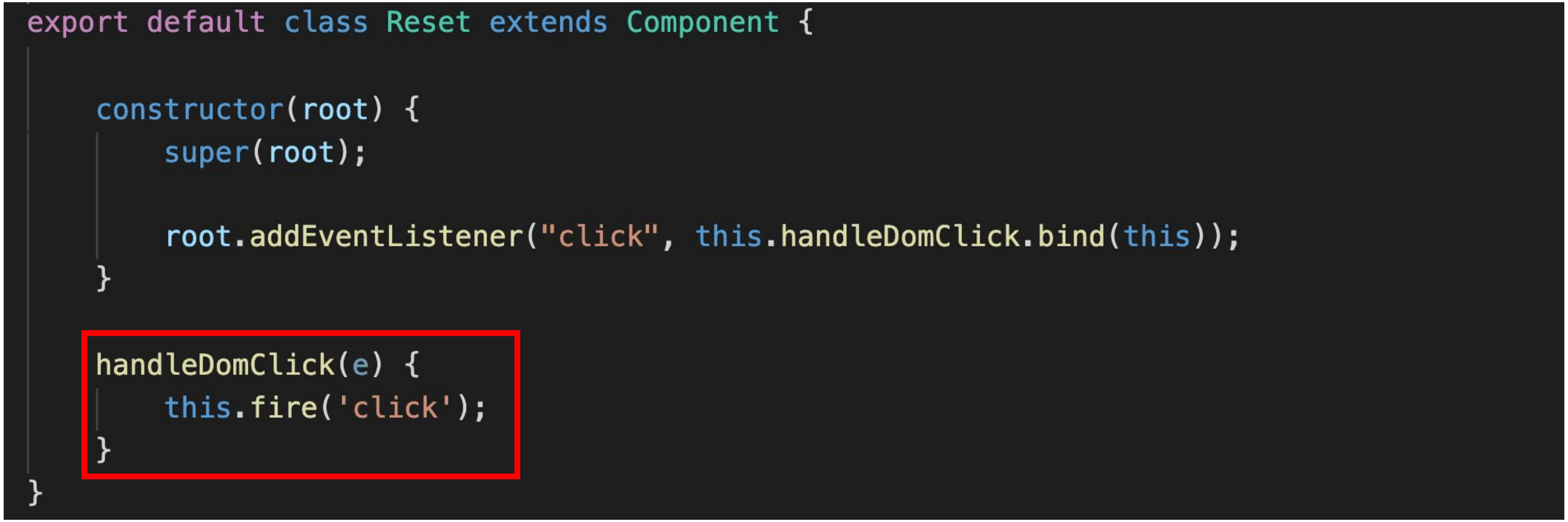


Main

Fire event

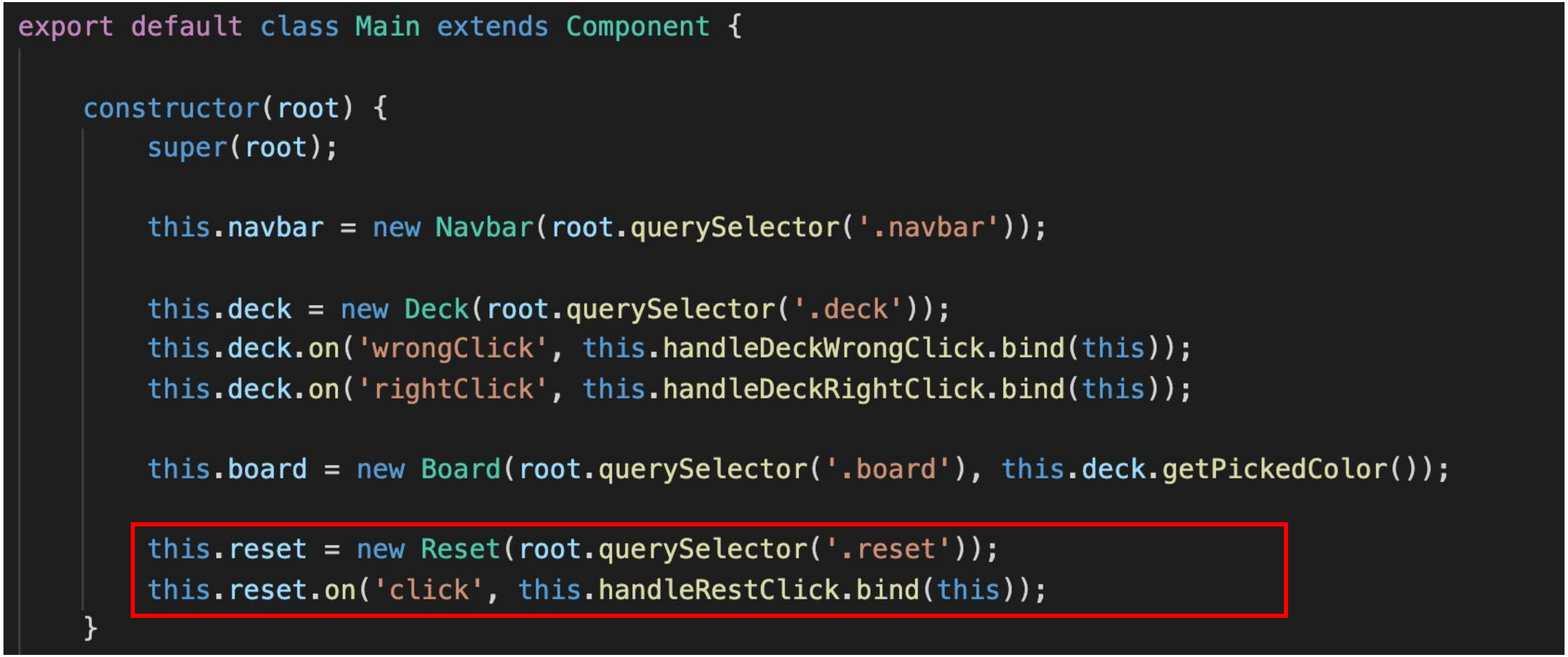
Reset

Reset

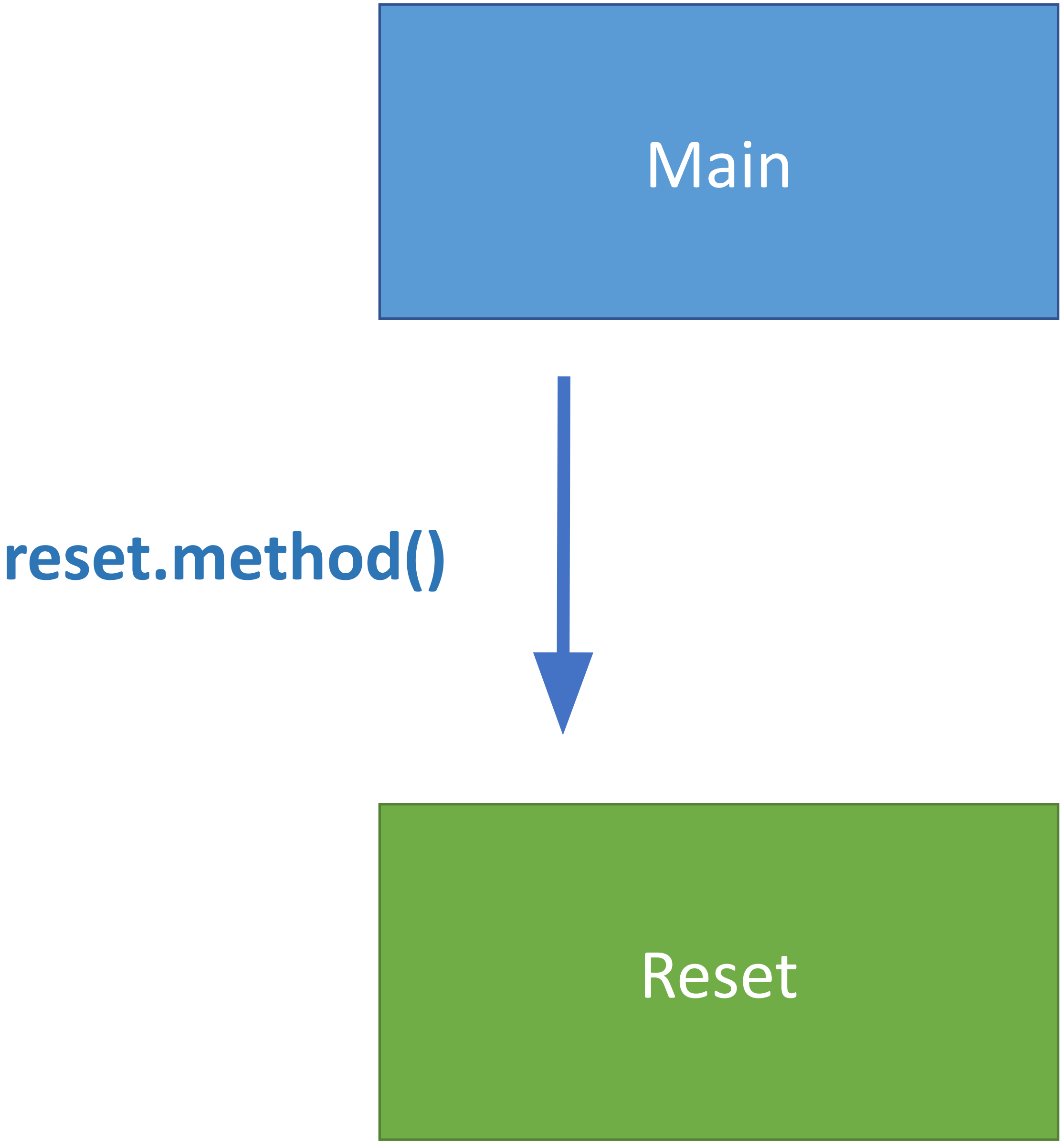


```
export default class Reset extends Component {

    constructor(root) {
        super(root);

        root.addEventListener("click", this.handleDomClick.bind(this));
    }

    handleDomClick(e) {
        this.fire('click');
    }
}
```

Main



```
export default class Main extends Component {

    constructor(root) {
        super(root);

        this.navbar = new Navbar(root.querySelector('.navbar'));

        this.deck = new Deck(root.querySelector('.deck'));
        this.deck.on('wrongClick', this.handleDeckWrongClick.bind(this));
        this.deck.on('rightClick', this.handleDeckRightClick.bind(this));

        this.board = new Board(root.querySelector('.board'), this.deck.getPickedColor());

        this.reset = new Reset(root.querySelector('.reset'));
        this.reset.on('click', this.handleRestClick.bind(this));
    }
```

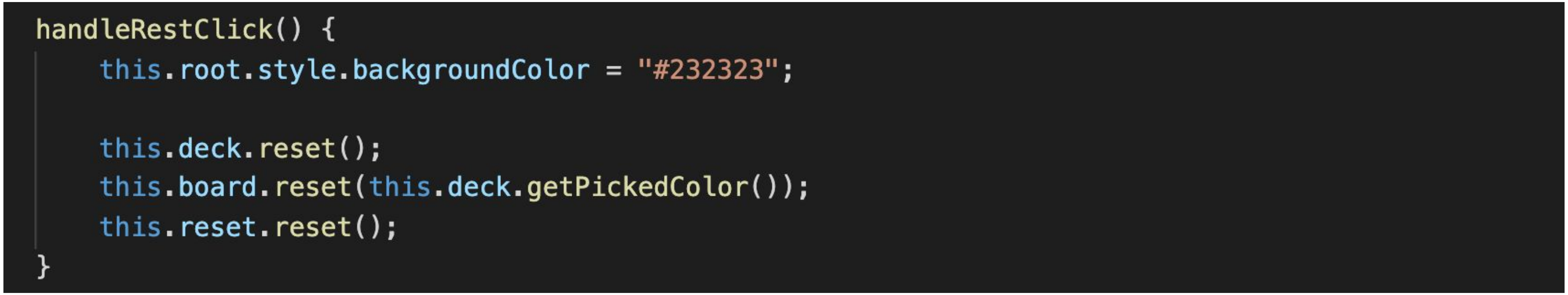


Main

reset.method()

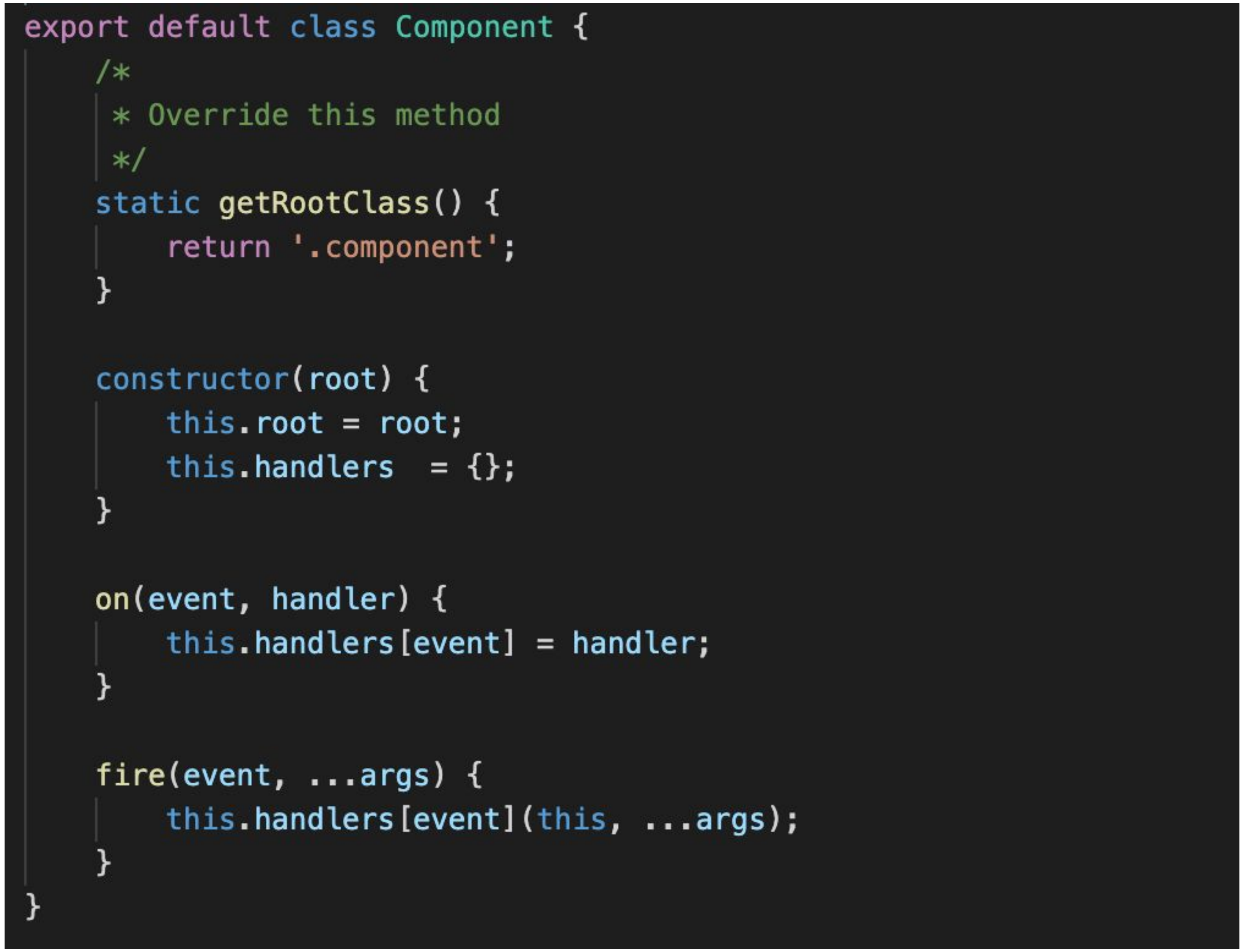
Reset

Main



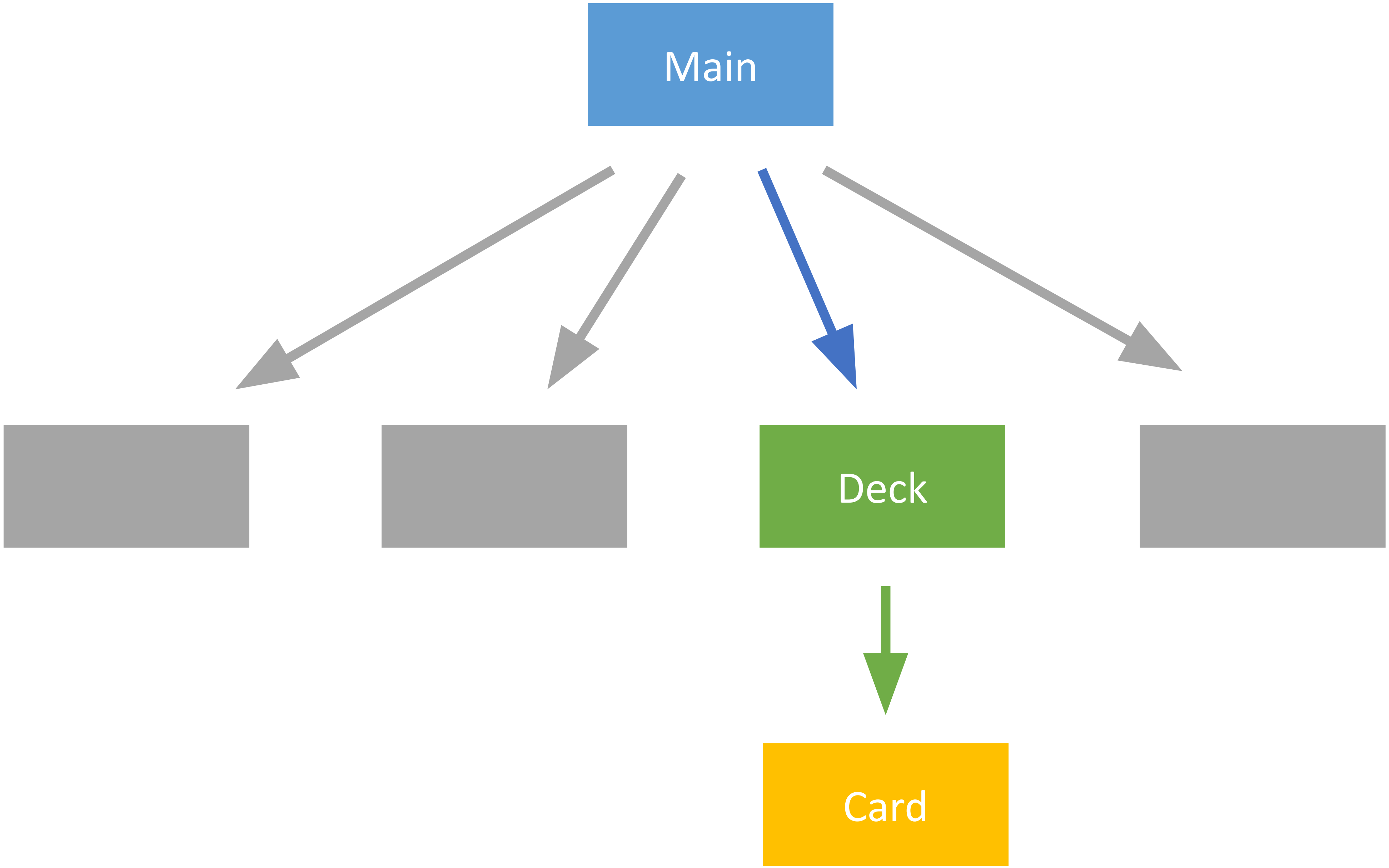
```
handleRestClick() {
    this.root.style.backgroundColor = "#232323";

    this.deck.reset();
    this.board.reset(this.deck.getPickedColor());
    this.reset.reset();
}
```

Component

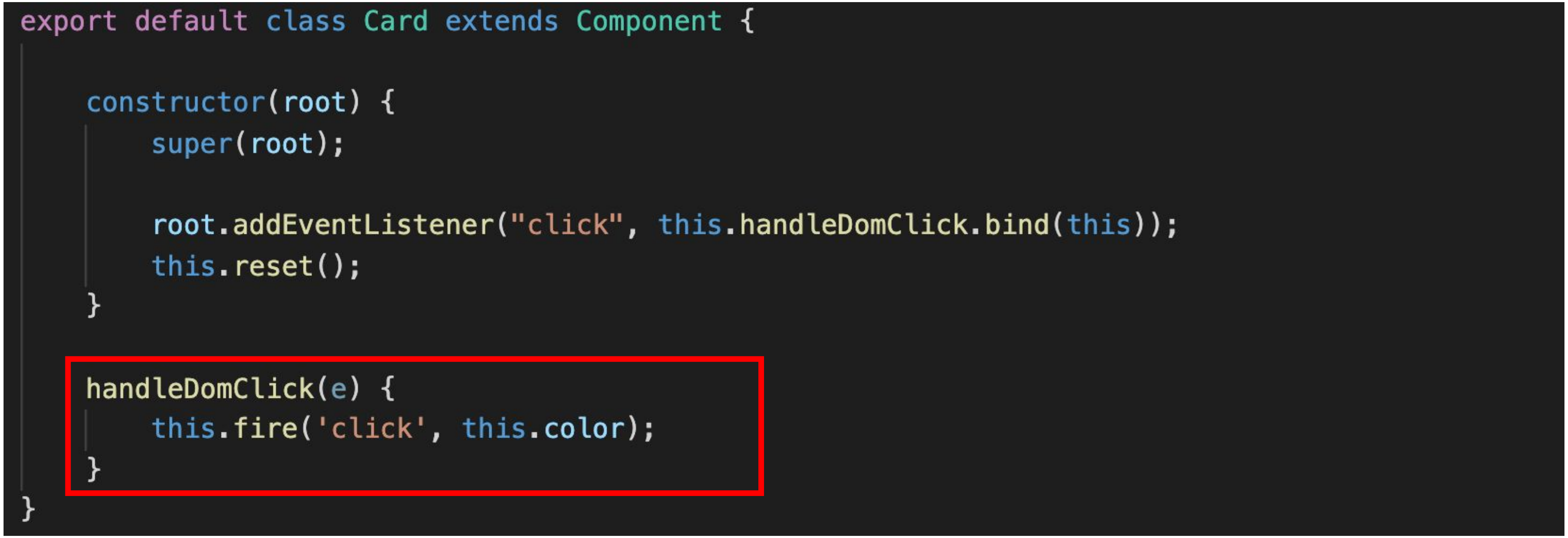


```
export default class Component {
    /*
     * Override this method
     */
    static getRootClass() {
        return '.component';
    }

    constructor(root) {
        this.root = root;
        this.handlers  = {};
    }

    on(event, handler) {
        this.handlers[event] = handler;
    }

    fire(event, ...args) {
        this.handlers[event](this, ...args);
    }
}
```

How about click card?

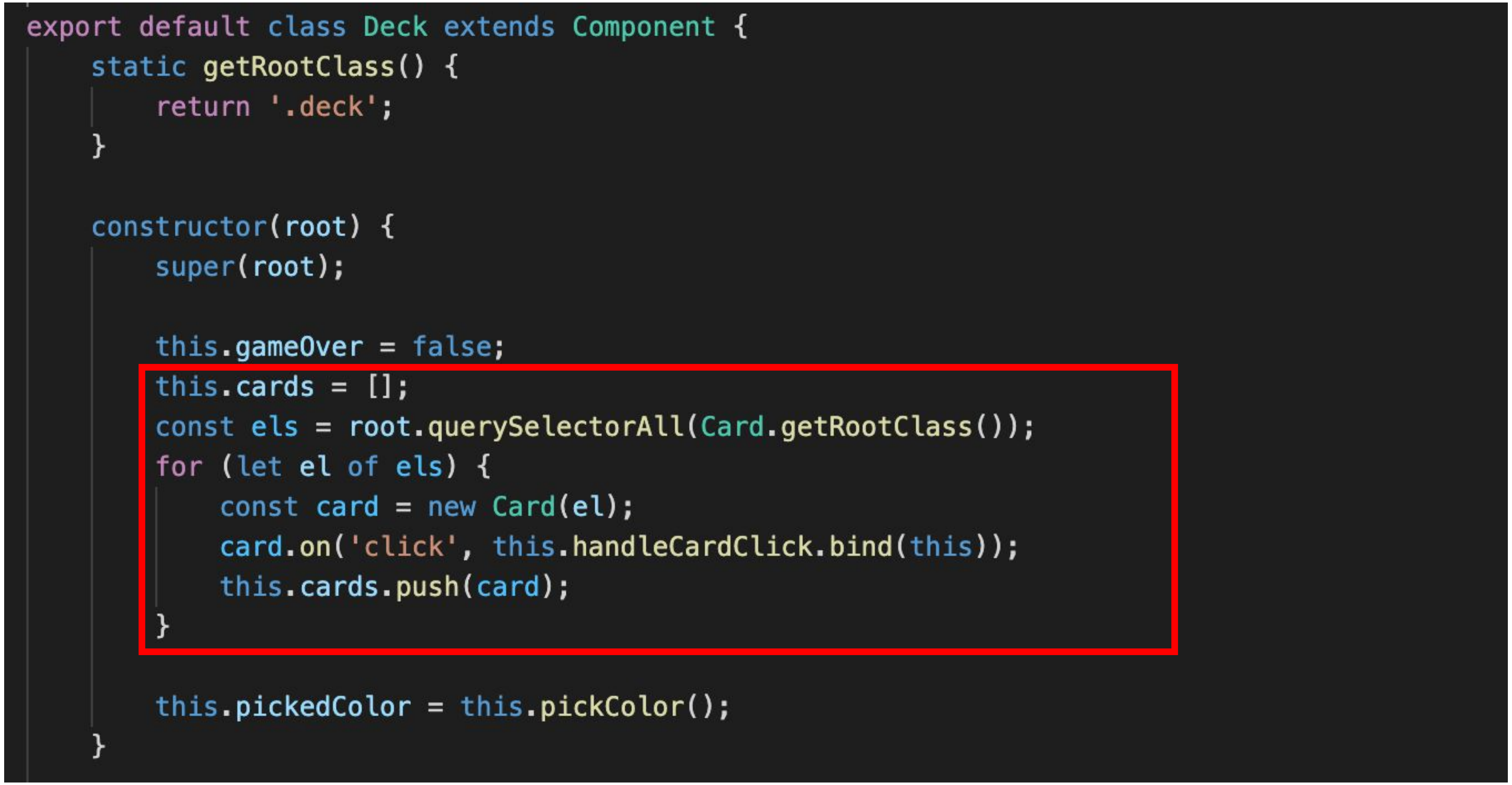

Main
Deck
Card

Card



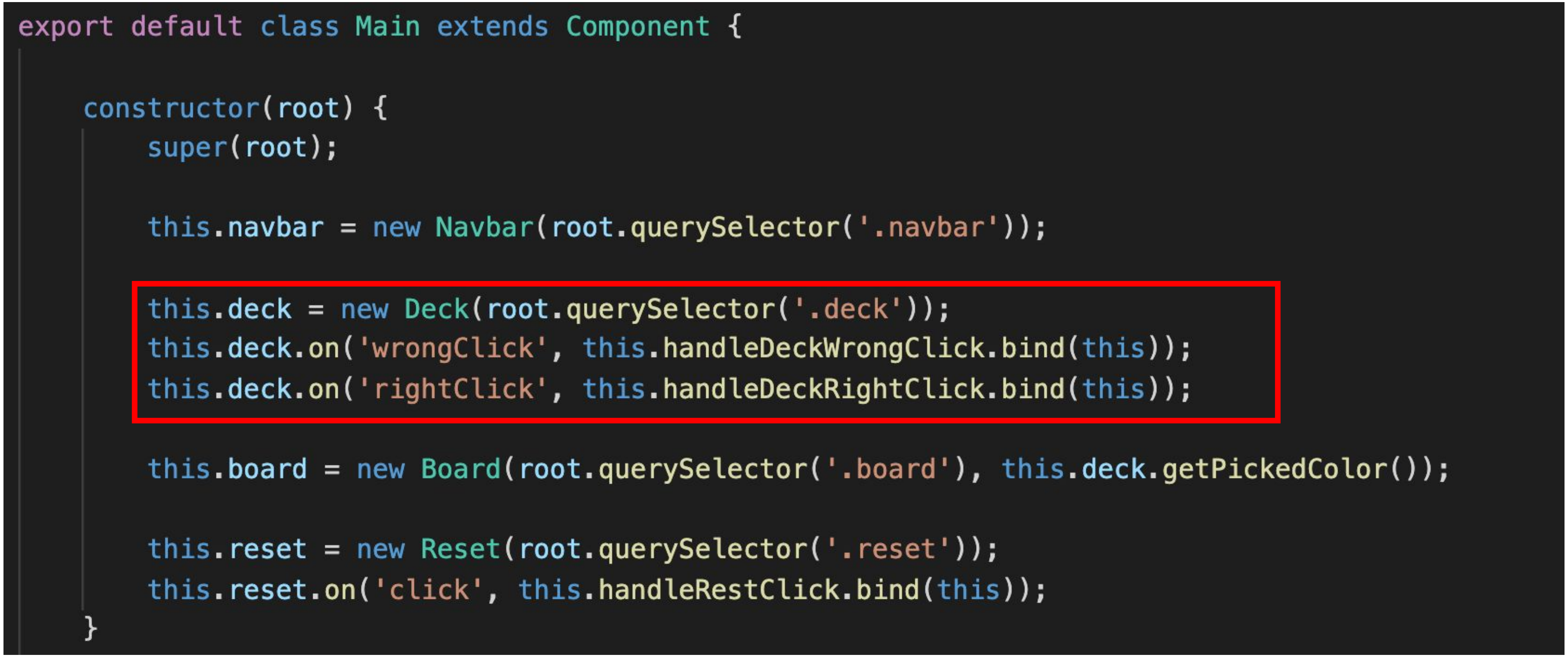
```
export default class Card extends Component {

    constructor(root) {
        super(root);

        root.addEventListener("click", this.handleDomClick.bind(this));
        this.reset();
    }

    handleDomClick(e) {
        this.fire('click', this.color);
    }
}
```

Deck



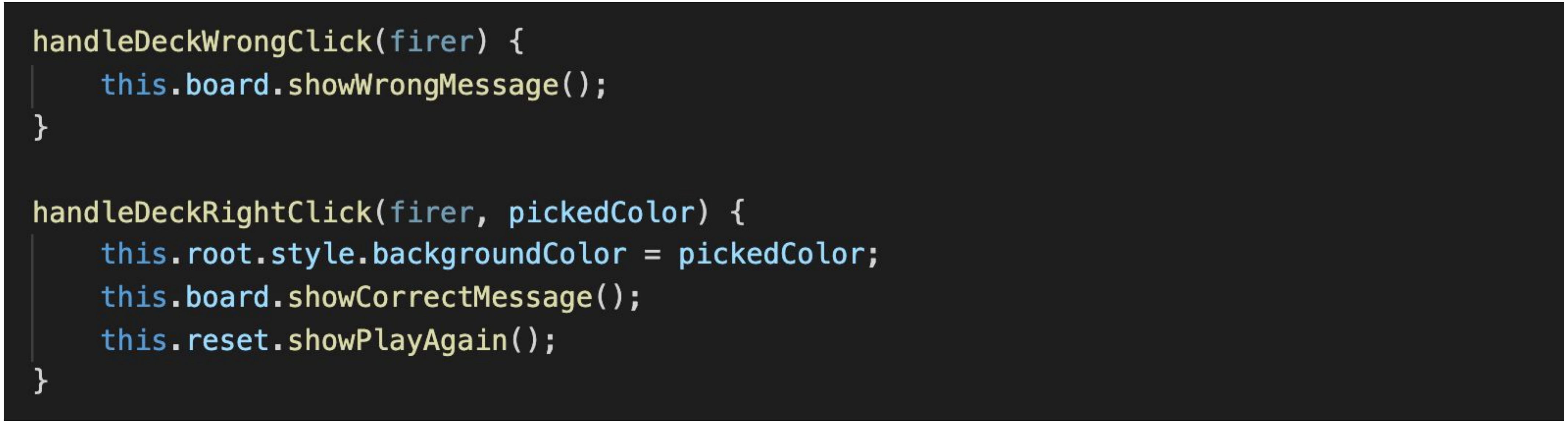
```
export default class Deck extends Component {
    static getRootClass() {
        return '.deck';
    }

    constructor(root) {
        super(root);

        this.gameOver = false;
        this.cards = [];
        const els = root.querySelectorAll(Card.getRootClass());
        for (let el of els) {
            const card = new Card(el);
            card.on('click', this.handleCardClick.bind(this));
            this.cards.push(card);
        }

        this.pickedColor = this.pickColor();
    }
```

Main



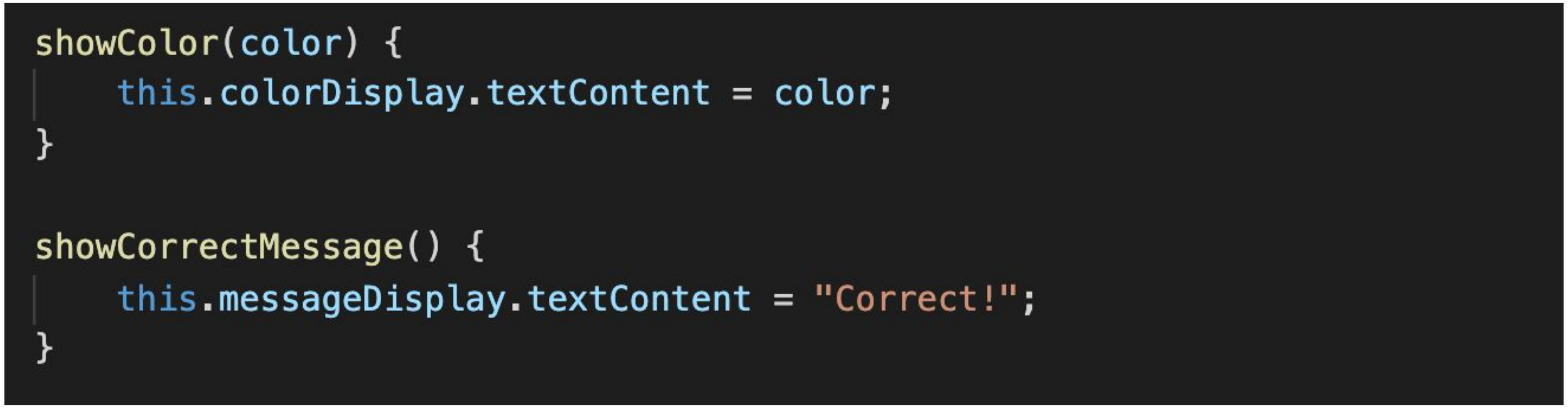
```
export default class Main extends Component {

    constructor(root) {
        super(root);

        this.navbar = new Navbar(root.querySelector('.navbar'));

        this.deck = new Deck(root.querySelector('.deck'));
        this.deck.on('wrongClick', this.handleDeckWrongClick.bind(this));
        this.deck.on('rightClick', this.handleDeckRightClick.bind(this));

        this.board = new Board(root.querySelector('.board'), this.deck.getPickedColor());

        this.reset = new Reset(root.querySelector('.reset'));
        this.reset.on('click', this.handleRestClick.bind(this));
    }
```

Main



```
handleDeckWrongClick(firer) {
    this.board.showWrongMessage();
}

handleDeckRightClick(firer, pickedColor) {
    this.root.style.backgroundColor = pickedColor;
    this.board.showCorrectMessage();
    this.reset.showPlayAgain();
}
```

Board



```
showColor(color) {
    this.colorDisplay.textContent = color;
}

showCorrectMessage() {
    this.messageDisplay.textContent = "Correct!";
}
```

You can change:

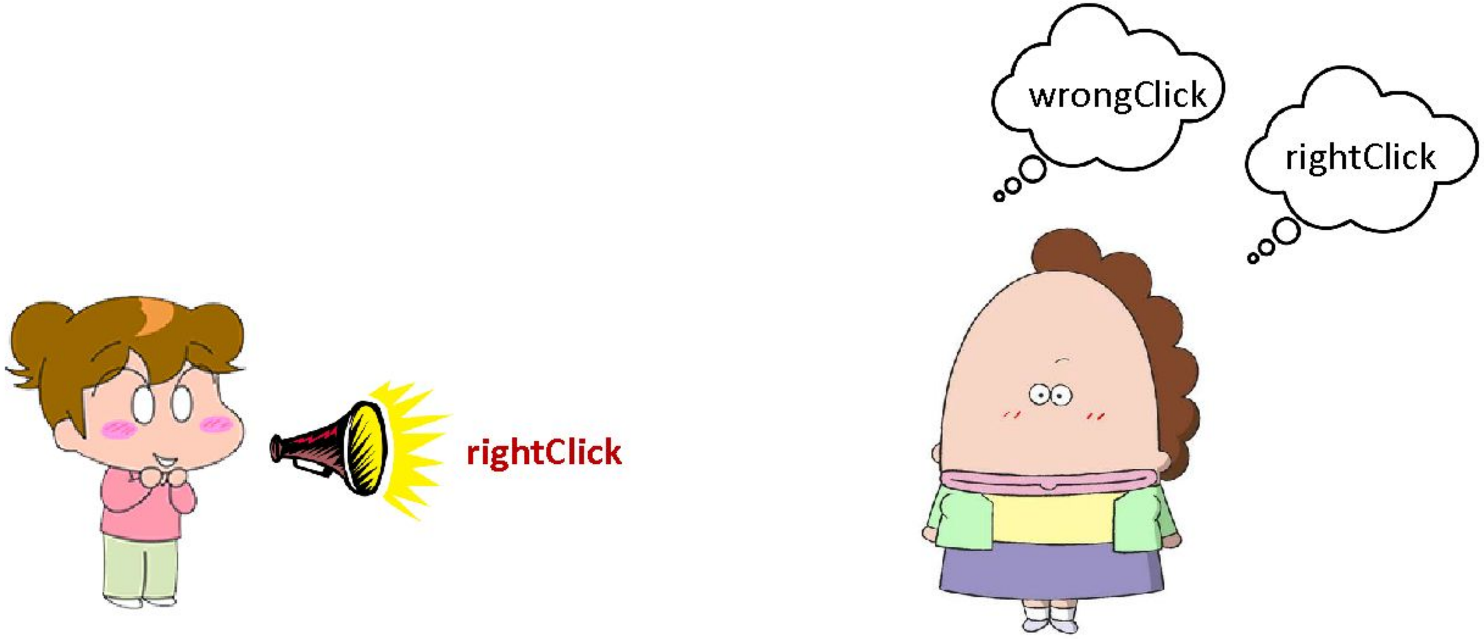
index.html (card & mode) in dist folder

all .js file except component.js in src folder

navbar.css in src folder

new file for mode.css & mode.js

You should keep its structure.



Deck

Main

rightClick

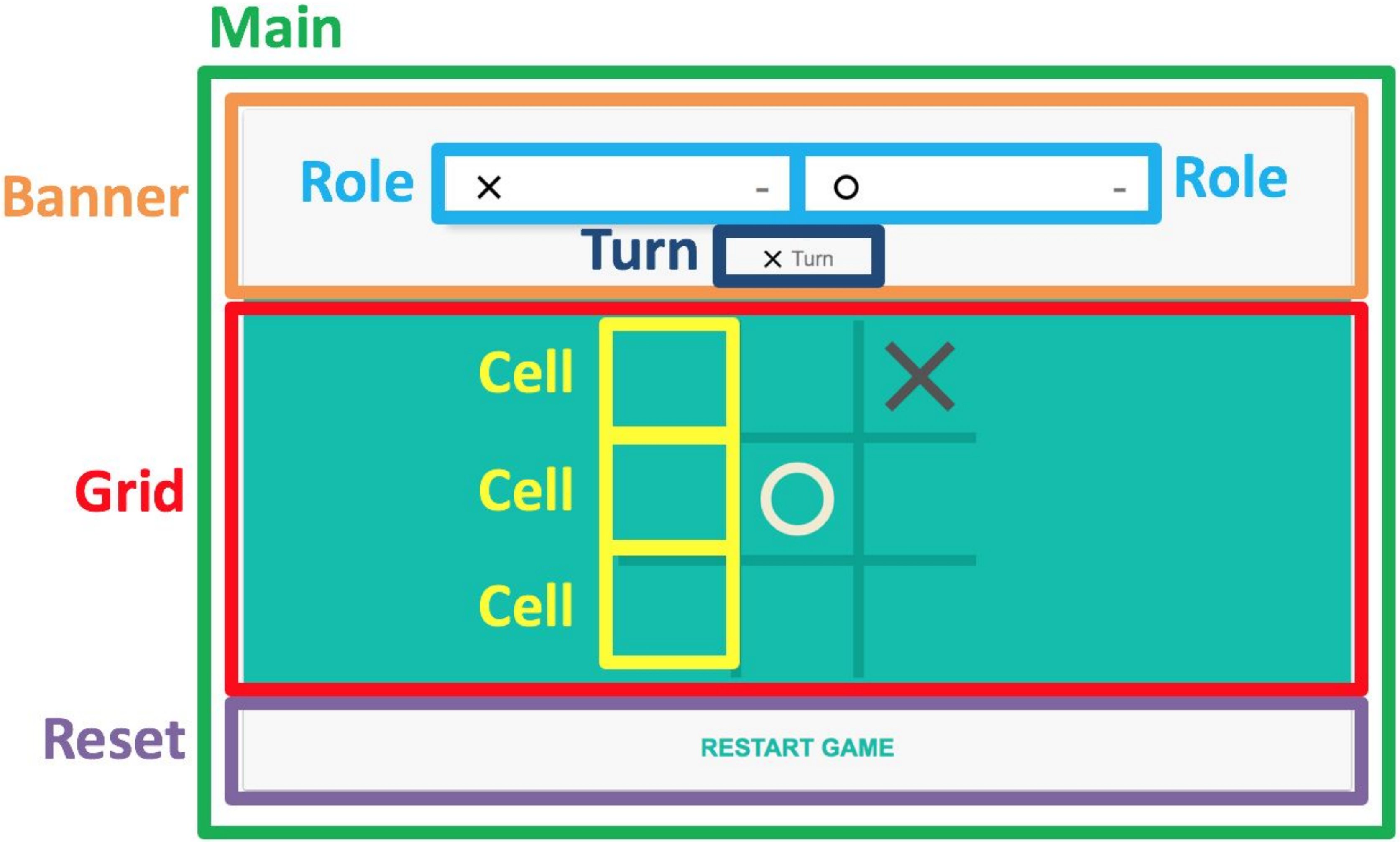
rightClick

Deck

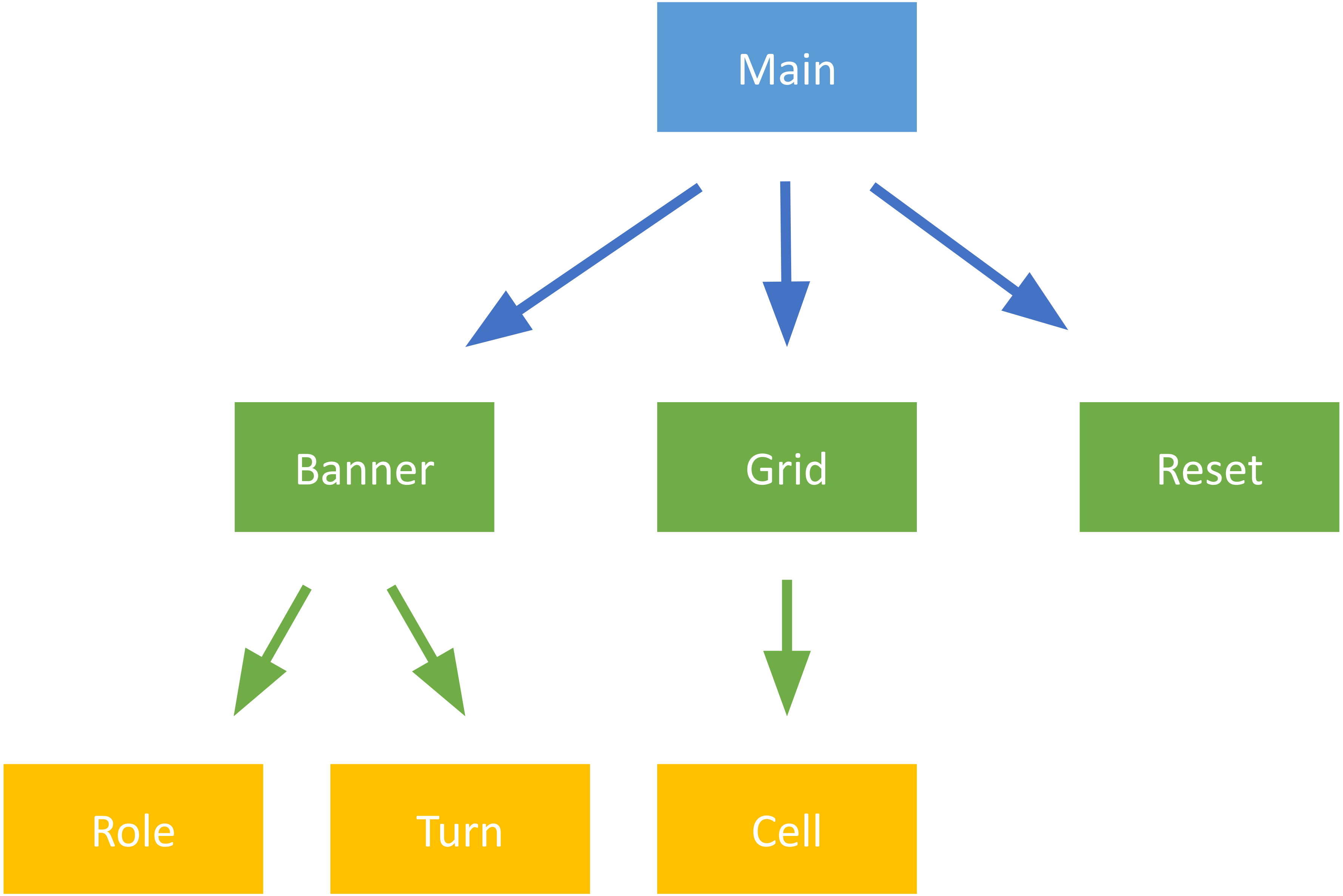
Main

Homework!

Tic-Tac-Toe

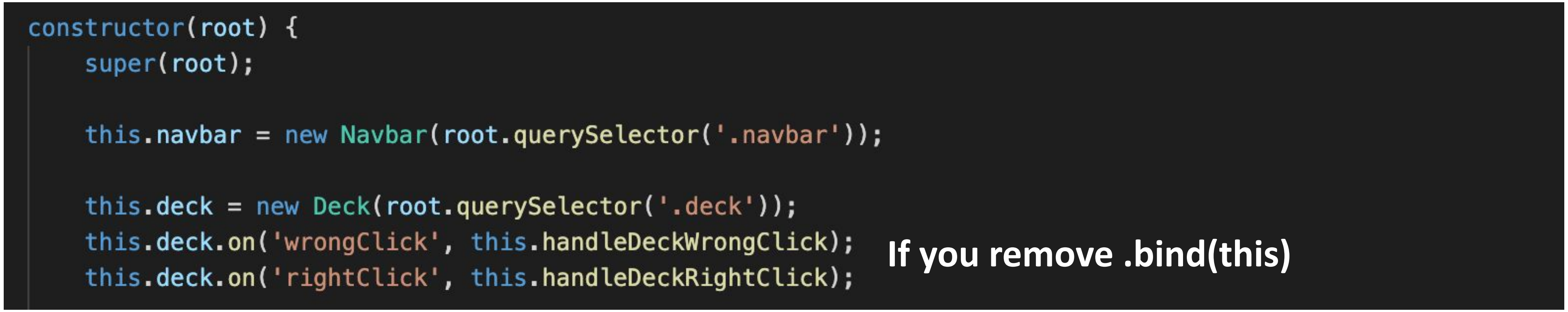


Main
Banner
Role
×
-
○
-
Role
Turn
× Turn
Grid
Cell
Cell
Cell
Reset
RESTART GAME


Main
Banner
Grid
Reset
Role
Turn
Cell

Precautions

- Remember **.bind(this)**



```
constructor(root) {
    super(root);

    this.navbar = new Navbar(root.querySelector('.navbar'));

    this.deck = new Deck(root.querySelector('.deck'));
    this.deck.on('wrongClick', this.handleDeckWrongClick);
    this.deck.on('rightClick', this.handleDeckRightClick);
```

If you remove .bind(this)

The window calls the function!

Precautions

- Remember **.bind(this)**
- Find your bug by printing the value!