Color-Picker

https://github.com/alienbrains/Background-Color-Changer

Src

App.js

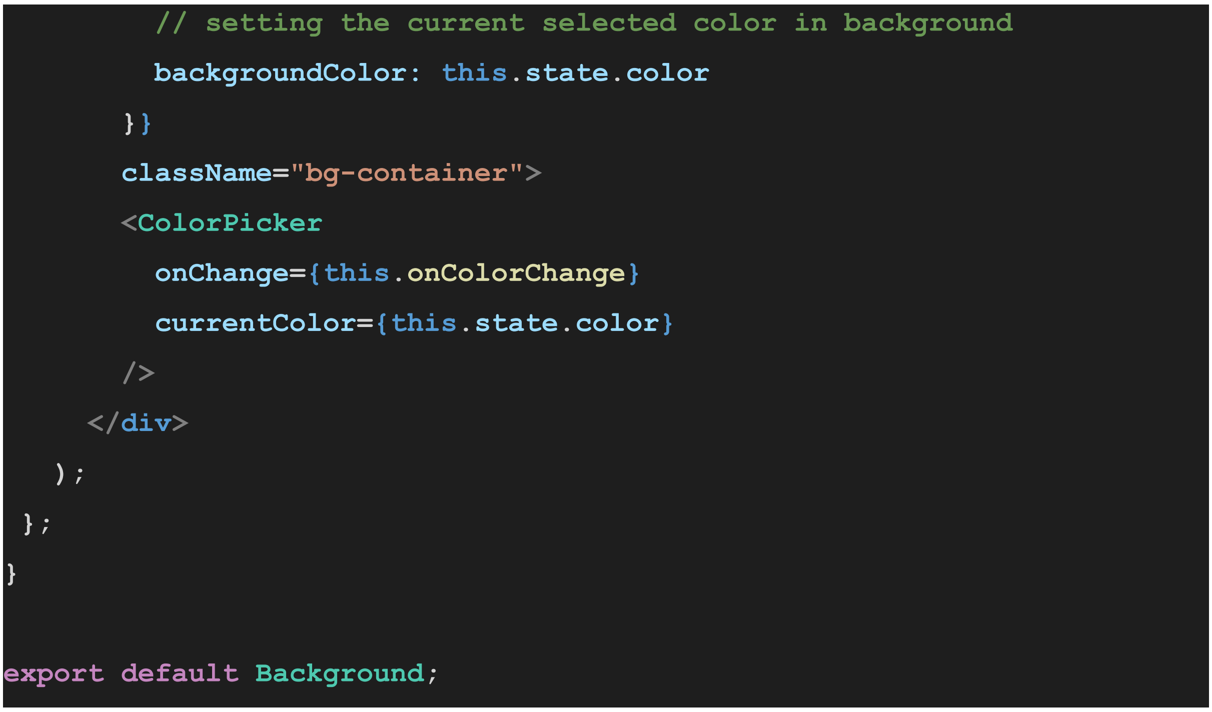
```
import React from "react";
import Background from "./Components/Background/index";

import "./App.css";

function App() {
 return <Background />;
}

export default App;
```

This is the initial App.js Component which is a functional component.

Components

Components/Background/index.js

```
import React from "react";

import ColorPicker from "../ColorPicker/index";

import "./style.css";

class Background extends React.Component {
 constructor(props) {
   super(props);
   this.state = {
     color: "white"
   };
 }

 //  executed when a color is selected in color picker
 onColorChange = (e) => {
   //  here e refers to an event which has been occured when a color is
selected

   //  whenever we update the color in state then the UI also
re-renders to show the color
   this.setState({
     color: e.target.value // the current color value selected is
stored in e.target.value
   });
 };

 render = () => {
   return (
     <div
       style={{
```



```
         // setting the current selected color in background
         backgroundColor: this.state.color
       }}
       className="bg-container">
       <ColorPicker
         onChange={this.onColorChange}
         currentColor={this.state.color}
       />
     </div>
   );
 };
}

export default Background;
```

Initially in the state we declare the background color as white.
This initial color stored in **this.state.color** is passed down to the child ColorPicker component from the parent Background component as **currentColor** . We also pass the function **this.onColorChange** function using the prop **onChange,** this function is used to change the current color.

The <div/> tag accepts an inline style property.
Now, the in-line style accepts a style object, here the style object is
{
        backgroundColor: this.state.color
}

We apply this inside the style attribute. Hence, two brackets.
style = { { backgroundColor: this.state.color } }
When the state of the color changes we can see that the background color also changes.

Components/Brick/index.js

```
import React from "react";

import "./style.css";

const ColorPicker = (props) => {
 return (
   <div className="picker-box">
     <span>Choose a color</span>
     <input type="color" onChange={props.onChange} />
     <span>{props.currentColor}</span>
   </div>
 );
};

export default ColorPicker;
```

The **ColorPicker** component accepts props from its parent component **Background.**
The <input /> tag is of type “color” hence it will display a color picker.
The <span /> tag displays the current color name.
When we change the particular color in the color picker. The **onChange** function is called which in turn calls the parent function **this.onColorChange,** as it is passed down as props from the parent to the child component.

The **onColorChange** function accepts an **event** as an argument (this occurs when we change the color using the color picker). This changed color is stored in the **event.target.value** key of the **event** object.

We update the state of the color using **this.setState,** which updates **this.state.color .** On calling **this.setState** the component re-renders and we see the changes in the UI. The updated name of the particular color is also displayed in the <p/> tag.