**HTML**

<html>

<body>

<div id="container"></div>

</body>

</html>

**CSS**

@import 'https://fonts.googleapis.com/css?family=Open+Sans';

body {

font-family: 'Open Sans', sans-serif;

}

**Javascript**

function toHex(number) {

let num = parseInt(number).toString(16);

let result = num.length != 2 ? "0" + num : num;

return result;

}

function hexColor(r, g, b) {

return toHex(r) + toHex(g) + toHex(b);

}

const Client = React.createClass({

getInitialState: function() {

return {

color: '000000'

}

},

componentDidMount: function() {

this.update();

},

update: function() {

this.setState({

color: hexColor(

this.refs.red.value,

this.refs.green.value,

this.refs.blue.value

)

});

if (this.props.updater !== undefined) {

this.props.updater({

r: this.refs.red.value,

g: this.refs.green.value,

b: this.refs.blue.value,

});

}

},

render: function() {

return ( <

table style = {

{

border: '.5em solid #F9F7ED',

padding: '2em',

margin: 'auto',

width: '100%'

}

} >

<thead>

<th colSpan="4">{this.props.title}</th>

</thead> <

tr >

<td>

R: <input type="number" min="0" max="255" defaultValue={this.props.color.r} ref="red" onChange={this.update} />

</td> <

td >

G: <input type="number" min="0" max="255" defaultValue={this.props.color.g} ref="green" onChange={this.update} /> <

/td> <

td >

B: <input type="number" min="0" max="255" defaultValue={this.props.color.b} ref="blue" onChange={this.update} /> <

/td> <

td >

<Bucket width="100px" height="100px" color={'#' + this.state.color}/> <

/td> < /

tr > <

/table>

);

}

});

const StaticClient = React.createClass({

render: function() {

return (

<table style={{ border: '.5em solid #F9F7ED', padding: '2em', width: '100%' }}>

<thead>

<th colSpan="4">{this.props.title}</th>

</thead>

<tr>

<td>

R: <input type="number" min="0" max="255" value={this.props.color.r} ref="red" readOnly />

</td>

<td>

G: <input type="number" min="0" max="255" defaultValue={this.props.color.g} ref="green" readOnly />

</td>

<td>

B: <input type="number" min="0" max="255" defaultValue={this.props.color.b} ref="blue" readOnly />

</td>

<td>

<Bucket width="100px" height="100px" color={'#' + hexColor(this.props.color.r,this.props.color.g,this.props.color.b)}/>

</td>

</tr>

</table>

);

}

});

const Diffie = React.createClass({

randomColor: function() {

return {

r: this.randomNumber(),

g: this.randomNumber(),

b: this.randomNumber()

};

},

getInitialState: function() {

return {

agreedPublicColor: this.randomColor(),

bobSecretColor: this.randomColor(),

aliceSecretColor: this.randomColor(),

}

},

randomNumber: function() {

return Math.floor(Math.random() \* (255 - 0 + 1)) + 0;

},

addColors: function(color1, color2) {

return {

r: (parseInt(color1.r) + parseInt(color2.r)) % 255,

g: (parseInt(color1.g) + parseInt(color2.g)) % 255,

b: (parseInt(color1.b) + parseInt(color2.b)) % 255

}

},

render: function() {

return (

<table style={{ textAlign: 'center' }}>

<tr>

<td colSpan="2">

<Client

title="Agreed Public Colour"

color={this.state.agreedPublicColor}

updater={(color) => this.setState({

agreedPublicColor: color

})}

<Client

title="Bob's Secret Colour"

color={this.state.bobSecretColor}

updater={(color) => this.setState({

bobSecretColor: color

})}

<Client

title="Alice's Secret Colour"

color={this.state.aliceSecretColor}

updater={(color) => this.setState({

aliceSecretColor: color

})}

<StaticClient

title="Bob's Mixed Colours"

color={

this.addColors(

this.state.bobSecretColor,

this.state.agreedPublicColor

<StaticClient

title="Alice's Mixed Colours"

color={

this.addColors(

this.state.aliceSecretColor,

this.state.agreedPublicColor

)

} <td colSpan="2">

<TwistedArrows />

<StaticClient

title="Bob's Secret Colour + Alice's Mixed Colours"

color={

this.addColors(

this.addColors(

this.state.aliceSecretColor,

this.state.agreedPublicColor

),

this.state.bobSecretColor

)

}

/>

</td>

<td>

<StaticClient

title="Alice's Secret Colour + Bob's Mixed Colours"

color={

this.addColors(

this.addColors(

this.state.bobSecretColor,

this.state.agreedPublicColor),

this.state.aliceSecretColor)

}

/>

</td>

</tr>

</table>

);

}

});

const TwistedArrows = React.createClass({

render: function() {

return ( < svg style = {

{

'transform': 'rotate(90deg)',

'transform-origin': '50% 50%',

width: 250,

height: 300,

'marginLeft': 100

}

} >

<defs id="genericDefs" /> <

g >

<g>

<path d="M247.3594 159.1875 L195.0469 205.5938 L195.0469 177.3281 Q193.9219 177.6094 192.375 177.6094 Q191.3906 177.6094 189.6328 177.4688 Q187.875 177.3281 186.8906 177.3281 Q182.8125 177.3281 176.3438 179.4375 Q169.875 181.5469 167.0625 184.0781 Q165.2344 185.625 150.3281 201.5156 Q132.1875 220.6406 117.8438 236.3906 Q110.1094 245.5312 93.5156 262.5469 Q90.1406 265.5 80.0156 270.2812 Q69.3281 275.2031 65.5312 275.2031 L11.5312 275.2031 L11.5312 237.7969 L61.5938 237.7969 Q71.2969 237.7969 87.1875 221.4844 Q98.1562 209.3906 108.9844 197.1562 Q123.1875 181.125 152.8594 150.75 Q163.9688 140.4844 178.4531 140.4844 Q179.5781 140.4844 181.8984 140.6953 Q184.2188 140.9062 185.3438 140.9062 Q187.875 140.9062 194.9062 140.625 L194.9062 112.2188 L247.3594 159.1875 ZM247.3594 256.5 L194.9062 302.7656 L194.9062 274.5 L172.8281 274.5 Q173.3906 274.5 167.4844 272.8125 Q156.9375 269.8594 146.25 259.3125 Q137.6719 249.6094 129.0938 240.0469 L151.1719 216.1406 Q170.5781 238.3594 186.3281 238.3594 Q187.7344 238.3594 190.4062 238.0781 Q193.0781 237.7969 194.3438 237.7969 Q194.9062 236.6719 194.9062 233.4375 Q194.9062 229.5 194.4844 209.5312 L247.3594 256.5 ZM113.3438 175.6406 L92.6719 199.6875 Q88.1719 195.3281 79.0312 186.6094 Q69.8906 177.8906 61.0312 177.8906 L11.5312 177.8906 L11.5312 140.4844 L66.2344 140.4844 Q66.2344 140.4844 72.9844 142.3125 Q84.5156 145.4062 95.4844 155.8125 Q99.9844 160.0312 113.3438 175.6406 Z" stroke="none" />

</g> <

/g> < /

svg > )

}

});

const Bucket = React.createClass({

render: function() {

return ( <

div style = {

{

width: this.props.width,

textAlign: 'center'

}

} >

<svg xmlns="http://www.w3.org/2000/svg" width={this.props.width} height={this.props.height} viewBox="0 0 24 24"><path fill={this.props.color} d="M22 4v-2h-20v2h-2c0 5.057 3.755 10.5 12 10.5 7.878 0 12-5.282 12-10.5h-2zm-20.941 1h.941v3.167c-.513-1.008-.822-2.086-.941-3.167zm17.941 6.495c-.602.417-1.267.783-2 1.082v-8.577h2v7.495zm3-6.495h.939c-.123 1.064-.437 2.114-.939 3.094v-3.094zm0 6.565v10.435h-20v-10.343c2.127 2.273 5.422 3.843 10 3.843 1.863 0 3.53-.278 5-.77v5.27h2v-6.135c1.168-.628 2.17-1.408 3-2.3z"/></svg> <

span style = {

{

'margin-top': '-10px',

'display': 'block'

}

}

{

this.props.color

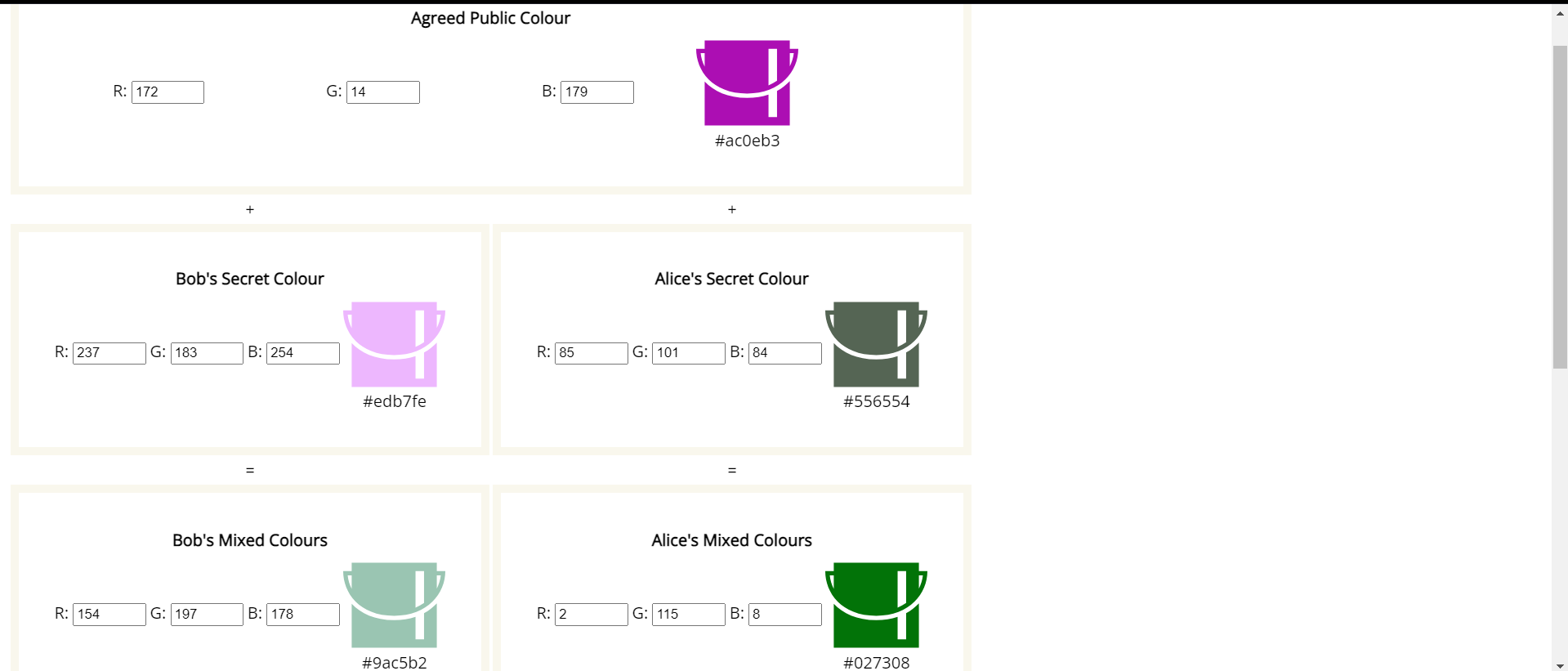
);

}

});

ReactDOM.render(<Diffie />, document.getElementById('container'));

**Output:**

****

****