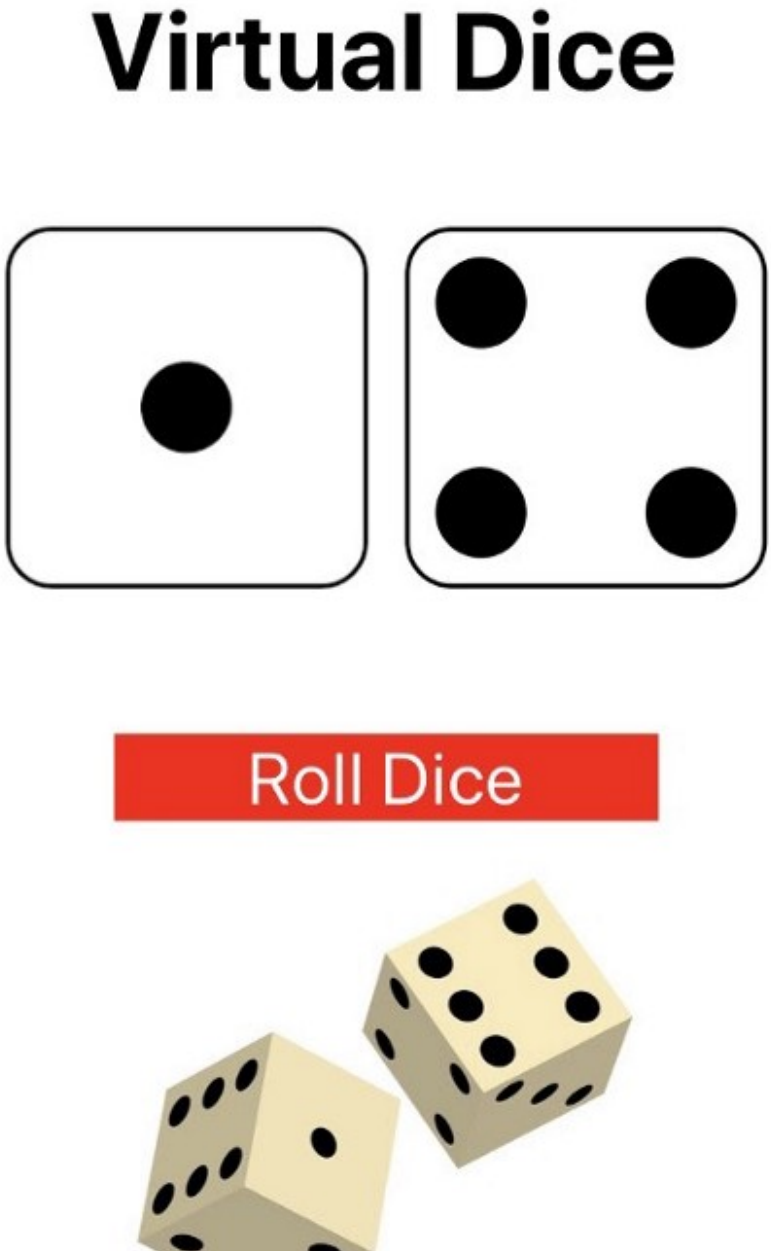
Create Your First App in React Native with Expo

Daniel Longo Follow
May 26 · 6 min read



In this article you will learn how to create a simple virtual dice App in React Native using Expo. Let's get started.

First we must create a new expo app.

If you do not have expo installed, the easiest way to get it is by running

```
$ npm install — global expo-cli
```

Expo is a framework for universal React applications. It allows you to develop apps for multiple platforms, including Android and IOS, from the same codebase.

In your terminal and in a desired directory create a new app called "virtualDice." When prompted select expo-template-blank.

```
$ expo init virtualDice
```

Now start your app.

```
$ cd virtualDice
$ yarn start
```

To view the app on your smartphone first download the expo app (https://apps.apple.com/us/app/expo-client/id982107779). Then scan the QR code outputted in your computer terminal with your phone.

Once you scan the QR code your phone should load your app (screenshot below).

Now it's time for us to start coding. Open the app code in your editor of choice (I prefer Webstorm). App.js contains the app code. If you have used React before, the React Native structure should appear familiar.

Our virtual dice app requires a few basic elements: TouchableOpacity , Image, and Text.

First, we will convert our App function to a React Component. This conversion increases the functionality available to us. Changes are noted in bold.

```
import React from 'react';
import { StyleSheet, Text, View} from 'react-native';

class App extends React.Component {
    render() {
        return (
            <View style={styles.container}>
```

```
                <Text>Open up App.js to start working on your app!
</Text>
            </View>
        );
    }
}

// This a method of making cleaner code. It acts similarly to a CSS
file would in HTML. But we won't use it much in this tutorial
const styles = StyleSheet.create({
    container: {
        flex: 1,
        backgroundColor: '#fff',
        alignItems: 'center',
        justifyContent: 'center',
    },
});

export default App
```

Next, let's create our dice.

```
import React from 'react';
import { StyleSheet, Text, View} from 'react-native';

class App extends React.Component {

    state = {
        dieA : 0, // 0 since we haven't rolled yet
        dieB : 0,
    };

    render() {
        return (
            <View style={styles.container}>
                <Text>Die 1: {this.state.dieA}</Text>
                <Text>Die 2: {this.state.dieB}</Text>
            </View>
        );
    }


}

// This a method of making cleaner code. It acts similarly to a CSS
file would in HTML. But we won't use it much in this tutorial.
const styles = StyleSheet.create({
    container: {
        flex: 1,
        backgroundColor: '#fff',
```

```
        alignItems: 'center',
        justifyContent: 'center',
    },
});

export default App
```

What is happening above?

We have a render function that will be called by React to figure out what should be displayed. The tags inside the return statement are very analogous to HTML tags; however, there are many different kinds with different properties very different from standard HTML.

Additionally, you can create (by creating another class that extends React.Component and exporting it just like we did with App) and/or import tags (just like we did with StyleSheet, Text, and View above).

If you want to insert variables or code to execute inside of a tag, you can do so by surrounding it in brackets; Above, we reference the variable this.state.dieA in this manner.

Now we will create a TouchableOpacity. A TouchableOpacity is very similar to a button. The tag makes an element clickable.

```
import React from 'react';
import { StyleSheet, Text, View, TouchableOpacity, Image} from
'react-native';

class App extends React.Component {

    state = {
        dieA : 0,
        dieB : 0,
    };

    render() {
        return (
            <View style={styles.container}>
                <Text>Die 1: {this.state.dieA}</Text>
                <Text>Die 2: {this.state.dieB}</Text>

                <View
```

```
                    style={{ marginTop: 40}}
                > // The reason there are double brackets is because
the inner brackets create an object while the outer brackets tell
React that the inside text should be interpreted as javascript
                    <TouchableOpacity
                        style={{backgroundColor: 'red'}}
                    >
                        <Text style={{ fontSize: 20, color: '#fff',
textAlign:'center' }}>Roll Dice</Text>
                        <Image source={{uri:
'https://cdn.pixabay.com/photo/2018/09/30/15/14/dice-
3713718_960_720.jpg'}}
                                style={{width: 150, height: 150}} />
                    </TouchableOpacity>
                </View>
            </View>
        );
    }
}
```

At this point, your app should look like the screenshot below/

If we click the dice, however, nothing happens. Let's make the Die numbers change.

```
class App extends React.Component {

    state = {
        dieA : 0,
        dieB : 0,
    };

    rollDie = () => {
        this.setState({
            dieA : Math.floor(Math.random() * 6) + 1,
            dieB : Math.floor(Math.random() * 6) + 1
        })
    }
    render() {
        return (
            <View style={styles.container}>
                <Text>Die 1: {this.state.dieA}</Text>
                <Text>Die 2: {this.state.dieB}</Text>

                <View
                    style={{ marginTop: 40}}
                >
                    <TouchableOpacity
                        style={{backgroundColor: 'red'}}
                        onPress={this.rollDie}
                    >
                        <Text style={{ fontSize: 20, color: '#fff',
textAlign:'center' }}>Roll Dice</Text>
                        <Image source={{uri:
'https://cdn.pixabay.com/photo/2018/09/30/15/14/dice-
3713718_960_720.jpg'}}
                                style={{width: 150, height: 150}} />
                    </TouchableOpacity>
                </View>
            </View>
        );
    }
}
```

Now the die should change number when we press the Roll Die image. This doesn't look to nice though. Let's replace the die values text with images and also add a title to the app page.

```
import React from 'react';
import { StyleSheet, Text, View, TouchableOpacity, Image} from
```

```
'react-native';

const dieImages = [
    "https://upload.wikimedia.org/wikipedia/commons/thumb/9/99/Dice-
0.svg/768px-Dice-0.svg.png",
    "https://encrypted-tbn0.gstatic.com/images?
q=tbn%3AANd9GcRCoZ9S_uqsrgbR5T0j8RxzY_QqLA4MFQxxOZ7ikDB9orVwjPT6&usqp
=CAU",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/5/5f/Dice-
2-b.svg/557px-Dice-2-b.svg.png",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/b/b1/Dice-
3-b.svg/557px-Dice-3-b.svg.png",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/f/fd/Dice-
4-b.svg/557px-Dice-4-b.svg.png",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/0/08/Dice-
5-b.svg/557px-Dice-5-b.svg.png",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/2/26/Dice-
6-b.svg/557px-Dice-6-b.svg.png"
];



class App extends React.Component {

    state = {
        dieA : 0,
        dieB : 0,
    };

    rollDie = () => {
        this.setState({
            dieA : Math.floor(Math.random() * 6) + 1,
            dieB : Math.floor(Math.random() * 6) + 1
        })

    }
    render() {
        return (
            <View style={styles.container}>
                <Text style={"fontSize":30, "fontWeight" : "bold",
marginBottom: 30}}>Virtual Dice</Text>
            <View style={{flexDirection: 'row'}}>
    <Image source={{uri: dieImages[this.state.dieA]}}
        style={{width: 100, height: 100, marginRight: 5}} />
        <Image source={{uri: dieImages[this.state.dieB]}}
               style={{width: 100, height: 100, marginLeft: 5}} />
        </View>


        <View
            style={{ marginTop: 40}}
        >
            <TouchableOpacity
```

```
                style={{backgroundColor: 'red'}}
                onPress={this.rollDie}
            >
                <Text style={{ fontSize: 20, color: '#fff',
textAlign:'center' }}>Roll Dice</Text>
                <Image source={{uri:
'https://cdn.pixabay.com/photo/2018/09/30/15/14/dice-
3713718_960_720.jpg'}}
                       style={{width: 150, height: 150}} />
            </TouchableOpacity>
        </View>
        </View>
    );
    }
}

const styles = StyleSheet.create({
    container: {
        flex: 1,
        backgroundColor: '#fff',
        alignItems: 'center',
        justifyContent: 'center',
    },
});

export default App
```

Finally, let's clean this up a bit. We can decompose some of the render code to make it a bit more readable.

```
import React from 'react';
import { StyleSheet, Text, View, TouchableOpacity, Image} from
'react-native';

const dieImages = [
    "https://upload.wikimedia.org/wikipedia/commons/thumb/9/99/Dice-
0.svg/768px-Dice-0.svg.png",
    "https://encrypted-tbn0.gstatic.com/images?
q=tbn%3AANd9GcRCoZ9S_uqsrgbR5T0j8RxzY_QqLA4MFQxxOZ7ikDB9orVwjPT6&usqp
=CAU",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/5/5f/Dice-
2-b.svg/557px-Dice-2-b.svg.png",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/b/b1/Dice-
3-b.svg/557px-Dice-3-b.svg.png",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/f/fd/Dice-
4-b.svg/557px-Dice-4-b.svg.png",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/0/08/Dice-
5-b.svg/557px-Dice-5-b.svg.png",
    "https://upload.wikimedia.org/wikipedia/commons/thumb/2/26/Dice-
6-b.svg/557px-Dice-6-b.svg.png"
];



class App extends React.Component {

    state = {
        dieA : 0,
        dieB : 0,
    };

    rollDie = () => {
        this.setState({
            dieA : Math.floor(Math.random() * 6) + 1,
            dieB : Math.floor(Math.random() * 6) + 1
        })

    }

    renderRolledDie() {
        let imageA = dieImages[this.state.dieA];
        let imageB = dieImages[this.state.dieB];
        return (
```

```
            <View style={{flexDirection: 'row'}}>
                <Image source={{uri: imageA}}
                       style={{width: 100, height: 100, marginRight:
5}} />
                <Image source={{uri: imageB}}
                       style={{width: 100, height: 100, marginLeft:
5}} />
            </View>
        )
    }

    renderRollButton() {
        return (
            <View
                style={{ marginTop: 40}}
            >
                <TouchableOpacity
                    style={{backgroundColor: 'red'}}
                    onPress={this.rollDie}
                >
                    <Text style={{ fontSize: 20, color: '#fff',
textAlign:'center' }}>Roll Dice</Text>
                    <Image source={{uri:
'https://cdn.pixabay.com/photo/2018/09/30/15/14/dice-
3713718_960_720.jpg'}}
                           style={{width: 150, height: 150}} />
                </TouchableOpacity>
            </View>
        )
    }

    render() {
        return (
            <View style={styles.container}>
                <Text style={{"fontSize":30, "fontWeight" : "bold",
marginBottom: 30}}> Virtual Dice</Text>
        {this.renderRolledDie()}
        {this.renderRollButton()}
    </View>
    );
    }
    }

    const styles = StyleSheet.create({
    container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
    },
    });

export default App
```

Congratulations! You have made your first React Native App with Expo. Leave any questions or comments below.

Special thanks to Kai Fronsdal who wrote part of this article

React Native Mobile App Development Tutorial Beginners Guide Expo