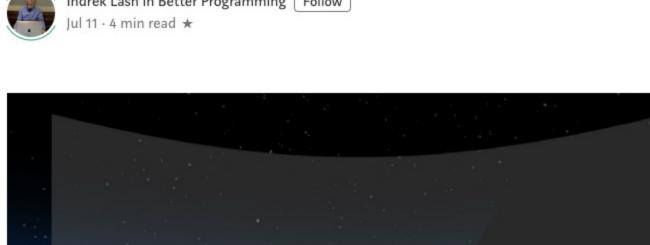
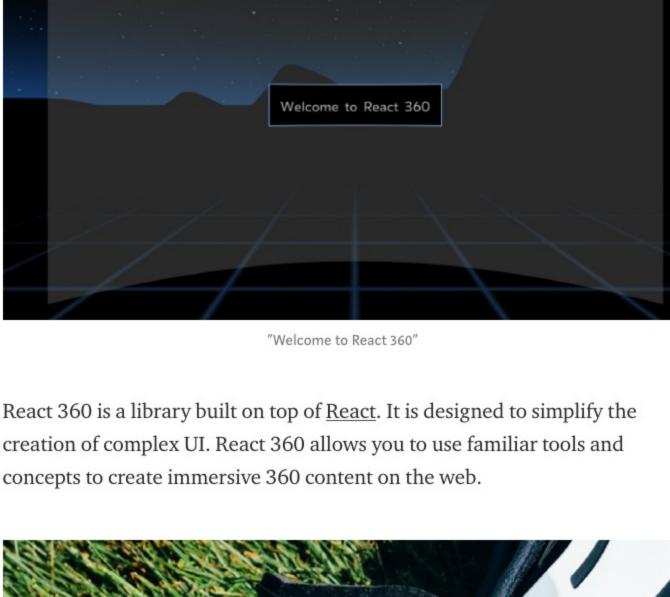
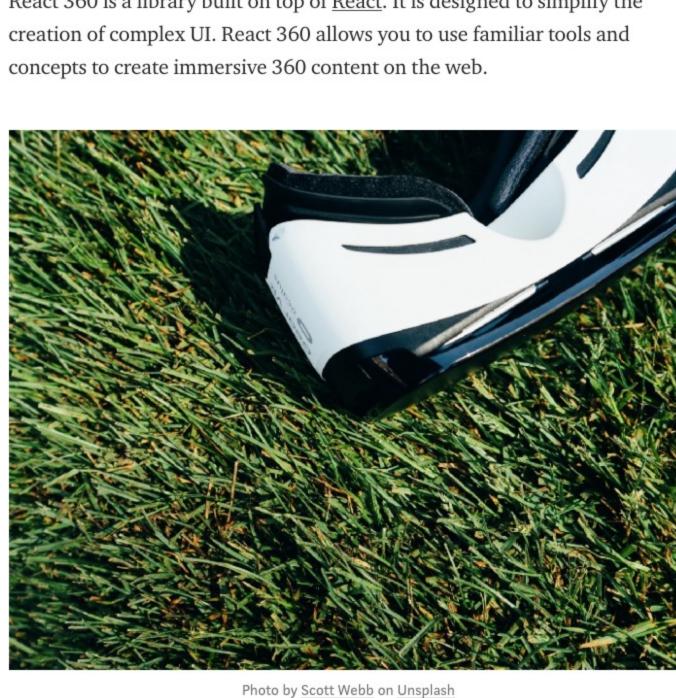
Exploring React 360 VR Library

React 360 is a framework for the creation of 3D and VR user interfaces

Indrek Lasn in Better Programming Follow







In early 2017, Facebook and Oculus released a JavaScript library called React VR, which was designed for the creation of 3D and VR experiences in the web browser. At the same time, Oculus began using a native C++ version of the framework to build its own first-party apps.

Over time, the APIs of the two projects diverged as we addressed the

can be consumed across PC, Mobile, and VR devices.

use cases. See full comparison here.

page.

0

npm install -g react-360-cli

yarn install v1.16.0

ncy "react@16.3.1".

success Saved lockfile.

☆ Done in 20.77s.

Done!

react-360`

0:

info No lockfile found.

different needs of the frameworks. To avoid confusion between the two

systems, the open source framework has been renamed React 360. This

If you previously used React VR, you should find React 360 to be very

similar. We've simplified a lot of the more common workflows, such as

We only need a browser and a code editor to get VR apps to work. Cool!

You'll need to install Node.js. While React 360 runs in your web browser,

Mac: On Mac, we recommend installing Node.js through <u>Homebrew</u>.

• Windows: Get the Windows installer from the nodejs.org download

• Linux: Go to the nodejs.org package manager page to find specific

Next, we're going to need the React 360 command line tool. This CLI tool

placing 2D UI elements in 3D space, and optimized performance for these

better reflects its use case: the creation of immersive 360 experiences that

How Is React 360 Different From React VR?

Getting Started

the build pipeline that bundles up your code relies on Node.

instructions for your Linux distribution.

lets you create and manage your React 360 projects.

Work react-360 init hello-react-360

Project directory created at hello-react-360

Creating new React 360 project...

[1/4] Resolving packages...

[2/4] Fetching packages...
[3/4] Linking dependencies...

Once we have the CLI, we can initialize our project.

react-360 init hello-react-360

longer maintained. Please, upgrade to core-js@3 or at least
to actual version of core-js@2.
warning react-native > metro > left-pad@1.3.0: use String.p
rototype.padStart()
warning react-devtools > electron > electron-download > nug

get > progress-stream > through2 > xtend > object-keys@0.4.

warning " > react-native@0.55.4" has incorrect peer depende

warning "react-native > eslint-plugin-react-native@3.7.0" h

Now enter the new project directory by running `cd hello-

Run `npm start` to initialize the development server

Open `index.js` to begin editing your app.

You can start your project now by running npm start

From there, browse to http://localhost:8081/index.html

initializing the React 360 project

warning react > fbjs > core-js@1.2.7: core-js@<2.6.8 is no</pre>

as unmet peer dependency "eslint@^3.17.0 || ^4 || ^5".
warning " > react-360@1.1.0" has incorrect peer dependency
"react@~16.0.0".
warning " > react-360@1.1.0" has incorrect peer dependency
"react-native@0.49.5".
[4/4] Building fresh packages...

https://github.com/facebook/react-native

/Users/indreklasn/Work/hello-react-360

[vr, dev] ./client.js

starting the React 360 project

If we open the browser at http://localhost:8081/index.html, we should be

Welcome to React 360

React 360 welcome message

90.4% (135/

Looking for JS files in

Metro Bundler ready.

BUNDLE

Loading dependency graph, done.

BUNDLE [vr, dev] ./client.js

able to see the welcome message.

React 360 Project Structure

BUNDLE [vr, dev] ./client.js

BUNDLE [vr, dev] ./client.js

| Price | Pric

RAP [wr, cov] _/client.js

application and determines how it looks.

Where all of your React code lives. This is the code that makes your

Any code imported by index.js will also become a part of your app,

This is the code that connects your browser to your React application — the

Runtime. This file does three things. First, it creates a new instance of React

DOM. This is also where your application can pass a variety of initialization

This is the web page you load when viewing your application. All it does is

functionality is left out of HTML, so that you can easily integrate your React

For the short and sweet demo, let's create a counter. The React 360 library

uses similar concepts as React Native. If you're interested in React Native

First, we need to import the View, Text, and VrButton elements from the

provide a point to mount your JavaScript code. This is intentional. Most

360 application into server-render pages or existing web apps.

360, loading your React code and attaching it to a specific place in the

allowing you to organize your app into many different files.

index.js

client.js

options.

index.html

Creating a Counter

react-360 library.

import {

AppRegistry,

StyleSheet,

2

3

4

14

15 16 17

18 19

21

2

3

4

6

8

14

15

20

};

index.js hosted with \ by GitHub

imported earlier.

render() {

return (

< VrButton

<Text>+</Text>

<Text>-</Text>

</VrButton>

</VrButton>

</Text>

index.js hosted with \ by GitHub

C Q localhost:8081/index.h

</View>

);

}

counter.

VR apps.

Where to Go From Here

Introduction to React 360

Thanks for reading! ♥

alligator.io

with React and uses Three.js for...

render() {

return ();

<View style={styles.panel}>

onClick={this.incrementCount}
style={styles.greetingBox}

onClick={this.decrementCount}
style={styles.greetingBox}

<Text style={styles.greeting}>
{`Count: \${this.state.count}`}

concepts, check out this tutorial.

import React from 'react';

100.0% (1/1), dom

```
Text,
 6
     View,
 7
       VrButton
     } from 'react-360';
                                                                                   view raw
 index.js hosted with \ by GitHub
Second, let's declare our initial state and create the decrement and
increment functions.
      export default class Hello360 extends React.Component {
  2
              state = {
                    count: 0,
  4
             };
  6
  7
              // This method increments our count, triggering a re-render
  8
             incrementCount = () => {
                      this.setState({count: this.state.count + 1});
  9
             };
 10
 11
              // This method decrements our count, triggering a re-render
              decrementCount = () => {
 13
```

this.setState({count: this.state.count - 1});

Pretty standard stuff. Next up, we're going to use the React 360 elements we

view raw

view raw

Roseth

", {..}, Array(4), Array(0), Array(0)]

, {_}, Array(5), Array(0), Array(0)]

", {_}, Array(1), Array(0), Array(0)]
Ross
sott", {_}, Array(0), Array(0), Array(0)
Ross
ross", (_), Array(1), Array(0), Array(0);

ReactMeliveContext.[3:73

ReactNativeContext.is:I3

* (5) ["VideoMedilo", (.), Array(8), Array(8), Array(8)]

Doministiller debug levels 0

Famining agglication "mello380" with appharams: ("initialProps": infelioo.isi35()
(), "rectfog":3), DEV_moment_level warning are 6N,
performance aptimizations are 6FF development_level warning are 6N,

Console

React 360 VR example

encourage you to take a deeper dive and explore all the possibilities to build

That's it for our super simple introduction to the React 360 library. I

Once we have the markup, we should be able to interact with our 360

I highly recommend reading through the official React 360 documentation for an in-depth dive.

React 360 · Create amazing 360 experiences

Create amazing 360 experiences

Create amazing 360 experiences

Create amazing 360 experiencesfacebook.github.io

WebVR provides support for exposing virtual reality devices - for example head-mounted displays like the Oculus Rift or...

developer.mozilla.org

VR Applications using React 360

If you've used React, you know how it can provide smart solutions to complex problems. And how exciting that can be...

egghead.io

React 360 (or React VR) is a library used to create virtual reality experiences