

1.6 React VR 综述

React 是 FaceBook 出的开源 JS 库，而 React VR 更上一层楼，可以在虚拟现实创建 UIs 和 3D 场景

这一节讲解的是 React VR 的一些关键点和概念

JSX 和声明似的 UIs

React 的一个吸引人的特性就是它将声明似的元素和代码以一种优雅的方式结合起来，UI 元素通过

组件标签的形式描述，有点儿类似 HTML，举个例子，标签 `<Greeting name='Joe' />` 描述的就是一

个带有 name 参数的 greeting 组件，这些表亲通过 JSX 插入到 JavaScript 里面。

JSX 是 JavaScript 的一个语法扩展，它将在 React Native 包里面处理成 JavaScript，JSX 比直接用代码来展现 UI 更优雅些

```
<MyButton color="blue" shadowSize={2}>
  Click Me
</MyButton>
```

编译成：

```
React.createElement(
  MyButton,
```

```
{color: 'blue', shadowSize: 2},  
'Click Me'  
)
```

用 JSX 也并非强制要用的，但是能带代码可读性增强，React 官方网站有 [JAX 的详细说明](#)。

React 关键的几个概念

用 React 需要知道一些术语和一些关键的概念，下面就是几个重要的点：

- **Components**- Components(组件) 可以在标签中复用的UI元素，

如<Greeting/>. React Native

提供了**Text**和**Image**组件.

更多的组件可以重写**React.Component**来声明.每一个组件都有一个

render() 函数，这个函数反悔了一系列的子组件

- **Props**- Components(组件) 可以携带像在<Greeting

name='Rexxar' />里面的**name**一样的参数.

参数就是属性，可以通过**this.props**加上**name**获取到，上面的例子中就是得到的，下一节有

[组件、属性和状态](#)的详细介绍.

- **State** - Components(组件)

可维持状态，这个状态可以影响组件的显示，state数据变化了，组

件重新渲染，所有状态都放在 **this.state** 对象里，通过 **setState** 函数改变状态，举个例子：

```
this.setState({myStateVariableCounter : 10})
```

· Events-

UI的动作发生时, Components(组件)会产生事件, 比如**View**有两个事件: **onEnter**、

onExit, 也就是当你看向或者鼠标指针进入退出这个区域时发生的, 我们也就是用这些事

件来进行交互的。

```
<View onEnter={() => this.setState({gazeEnabled : true})}>
```

· Layout-

React是用flexbox算法和布局规则自动地把组件放置到2D平面内的, 这个布局是通过

测量计算组件的宽高, 还有其他的样式比如**alignItems**, 更多的布局样式请查看左侧的[组件](#)、

[属性和状态](#)模块

· Style样式控制着组件的布局和展示效果:

```
<View style={{width: 100,height: 100,backgroundColor: 'skyblue'}}/>
```

属性除了放到组件行内(前端称为内联属性), 属性还可以通过**StyleSheet.create**抽出来放到外

部来引用, 这样外部的style可以在其他组件中复用, 效果更好些。

React 生态系统

React 最开始是为 web 开发而生的, 现在已经扩展到其他几个领域, 形成了 React 生态系统:

· [React](#)- 最原始的库, 主要是为web服务的, 创建浏览器渲染的DOM。

- [React Native](#)-

这个是用来开发IOS和Android的移动应用，也是建立在React的一些内核之上。

- **React VR**- 这里讨论的新库，主要是在VR中的UIs开发，简历在React Native框架之上的。

尽管 React VR 试运行在浏览器上的，相比 React Native 和 React, React VR 的结构更像前者，因为

也有<View>和<Text>,除了 2D 布局，也引入了 3D 场景、变换、全景，可以把物体放置在 3D 空间

内，同时在 VR 中渲染。

React VR 用了 OVRUI 库，这个库是利用 [Three.js](#)JS 3D 引擎开发，Three.js 是在浏览器上开发的，通

过 WebGL 来渲染场景的。利用 Web VR API 可以接入到 VR 眼镜中，然后再 Rift。GearVR 或者其他设

备上显示出来。React VR 可以不用 VR 眼镜也能在浏览器或者移动设备上创建一个 360 度的场景。

React Native 的几个概念框架同样在 VR 中也适用，下面就是几个概念的介绍。

React Native 包管理器

React JavaScript 代码在浏览器上运行之前要预处理下，React Native 包管理器就是做这个预处理

的工作的，它与 [Browserify](#) 和 [Webpack](#) 的作用类似，都是提供 CommonJS 样的模块系统、JS 编

译(ES6, Flow, JSX)、打包、加载资源等

React VR 的两个常用命令：

- **bundle**：处理、转换、合并JS文件到一个js文件中
- **start**：加载react

native包管理，开启一个web服务，动态转换js文件为bundles文件

npm start 就是打开这个管理器，也是下面命令的简写，在 `package.json` 文件中有配置：

```
node node_modules/react-native/local-cli/cli.js start
```

包管理器表现的就恶化本地 web 服务一样，也是把 React 或者 JSX 的代码转换成 js 代码，你可以通过

命令 **npm start** 运行项目，在浏览器中查看 <http://localhost:8081/index.vr.bundle?platform=vr>

如果要放到网站上，需要打包后的文件，打包命令 **npm run bundle** 的作用和下面的命令类似：

```
node node_modules/react-native/local-cli/cli.js bundle --entry-file
index.vr.js --platform vr --dev false --bundle-output
index.vr.bundle
```

`index.vr.bundle` 里面包含有 js 代码，可以直接被 HTML `script` 获取并执行

这里我们只讨论项目用到的范围，更多的包管理命令可以查看 [React Native Packager](#)

联网操作

React VR 用的是 Fetch API 从网络获取资源的，Fetch 和 `XMLHttpRequest` 或者其他的 APIs。

更过 Fetch 的用法，要去 React Native 的官网查看了：[Networking](#)

Fetch 更全的资料可以去这儿看, [the Fetch Request documentation](#).

也可以去 [the Fetch API documentation](#) 查看.