

1.10 输入事件

输入方式

我们的目标是让编写跨平台的交互更容易些，输入事件在不同的设备上是不一样的，所以我们提

供了 `event handlers` 能让最常见的几个输入方式(鼠标、键盘、手柄)标准化。

桌面上，最简单的交互方式就是用鼠标了，但在 VR 头戴设备上

是眼球追踪，你看到的目标就是输入的 `events`。简单来说，依赖不同的平台，选中一个组件可以用键盘、鼠标点击、点击屏

幕、眼球注视，我们提供了事件处理器来处理不同设备的交互，但是我们也为不同的平台提供了公

用的输入事件。

如果你要用简单有点击事件的按键，可以用 `VrButton`。

可见光标

当你带上头戴设备后，很难知道哪个组件当前是焦点，所以我们提供了一个应用层面的可见光标

标，就是在可见区域的中心。

`cursorVisibility` 在初始化的时间设置，值可以为以下几个值：

- **hidden** 光标不可见，默认是不可见的；
- **visible** 光标可见
- **auto** 这个可见性是由 React

VR的每一帧计算出来的，有输入事件的组件光标是可见，如绑定了

事件的`<VrButton>`或者`<View>`

这些设置是在初始化的时间就定义好的，第一次定义是在 `index.htm`

`l` 里面， `ReactVR.init()` 有第

三个可选的参数：

```
<!-- vr/index.html -->
<script>
ReactVR.init(
  '../index.vr.bundle?platform=vr',
  document.body,
  // third argument is a map of initialization options
  {
    cursorVisibility: 'visible',
  }
);
</script>
```

如果你想让这些配置也编译到你的应用里面，你可以在 `vr/client.js` 里面设置，这些配置就是

`VRInstance` 的最后一个参数：

```
// vr/client.js
```

```
const vr = new VRInstance(bundle, 'YourProject', parent, {  
  // Custom initialization options go here  
  cursorVisibility: 'visible',  
  
  ...options,  
});
```

已支持的输入事件

onEnter

这是当光标开始和组件交叉时触发的事件，在桌面上，发生在鼠标光标移到组件身上时；在 VR 头戴设备上，发生在用户视觉的中心移到组件上时

onExit

这是当光标离开组件时触发的事件，在桌面上，发生在鼠标光标离开组件时；在 VR 头戴设备上，发生在用户视觉的中心离开组件时

onGazeEnter

只有视觉的中心开始移到组件上时发生的事件

onGazeExit

只有视觉的中心离开组件时发生的事件

onMouseEnter

只有鼠标移到组件上时才发生的事件

onMouseExit

只有鼠标离开组件上时才发生的事件

onInput

这个事件在所有的设备上都能捕获所有的交互，提供了一个 **events** 的稳定的流，但是也可以过滤一

些动作，例如：按键 **handler** 或者 **VrButton** 过滤器就是为了每一个输入渠道捕获主要的按键

On its own, it provides a steady stream of events, but they can be filtered down to capture specific actions. For instance, the **onClick** handler of **VrButton** filters events to capture the primary buttons for each input channel.

Example

下面的代码就是当用户注视或者鼠标移到这个组件时，组件就会改变颜色

```
class ColorChange extends React.Component {  
  constructor() {  
    super();  
  }  
}
```

```
this.state = {textColor: 'white'};
}

render() {
  return (
    <Text
      style={{color: this.state.textColor}}
      onEnter={() => this.setState({textColor: 'red'})}
      onExit={() => this.setState({textColor: 'white'})}>
      This text will turn red when you look at it.
    </Text>
  );
}
}
```