

react-soonspace

npm@latest v2.1.2

React 中快速使用 soonspacejs。

Breaking changes

2.0.0-rc.4 之后, 所有的事件以 `events` 形式传入, 与 `options` 保持一致.

安装

```
npm install react-soonspace soonspacejs -S
# or
yarn add react-soonspace soonspacejs -S
```

sh

提示

安装 `react-soonspace` 插件时, 要同时安装 `soonspacejs`, 但在使用 前者 组件时不必手动引入 后者, 内部自动引入。这样做是为了保证 后者 版本最新, 不受版本依赖限制。

使用方法

```
import React, { useMemo, useCallback } from 'react';
import ReactSoonspace from 'react-soonspace';

/**
 * 提取到组件外部保证 `options` 始终不变
 */
const options = {
  showInfo: false,
  background: {
```

```
    color: 0x333300,
  },
};

export default function SoonspaceTest() {
  /**
   * 或者将 events 提取到组件外部
   */
  const events = useMemo(
    () => ({
      selectPosition(position) {
        console.log('selectPosition', position);
      },
    }),
    []
  );

  /**
   * 或者将 sceneReady 提取到组件外部
   */
  const sceneReady = useCallback((ssp) => {
    console.log('sceneReady', ssp);
  }, []);

  return (
    <div className='App'>
      <ReactSoonspace
        className='soonspace-view'
        options={options}
        events={events}
        sceneReady={sceneReady}
      />
    </div>
  );
}
```

注意

自 `2.0.0-rc.3` 版本以后, 任何参与到 `Soonspace` 实例化的 `props` 的改变都会导致场景重新渲染.

例如 `options` , `events` 等.

这样的目的是使场景融合入 `react` 的生命周期中, 故在某些情况可能会导致场景渲染死循环.

建议使用 `useMemo` [↗](#) 、 `useCallback` [↗](#) 优化, 或者将渲染无关的 `props` 提取到组件外部.

配置属性

id

自定义 ID

- 类型: string
- 默认值: `SoonSpace_View_${random}`

className

自定义类名

- 类型: string
- 默认值: `undefined`

style

自定义样式

- 类型: [React.CSSProperties](#) 
- 默认值:

```
{  
  position: 'relative',  
  width: '100%',  
  height: '100%'  
}
```

js

options

`soonspace` 配置项

- 类型: object
- 默认值: `{}`

events

soonspace 场景事件

- 类型： object
- 默认值： `{}`

方法属性

sceneReady

场景准备完成时触发函数。

回调参数

ssp

soonspace 实例

← [vue-soonspace](#)

[soonmanager-sync](#) →