

场景操作

setBackgroundColor

设置背景色

样例：



定义：

```
function setBackgroundColor(color: IColor): void;
```

ts

用法：

```
ssp.setBackgroundColor(0xff0000);
```

js

参数:

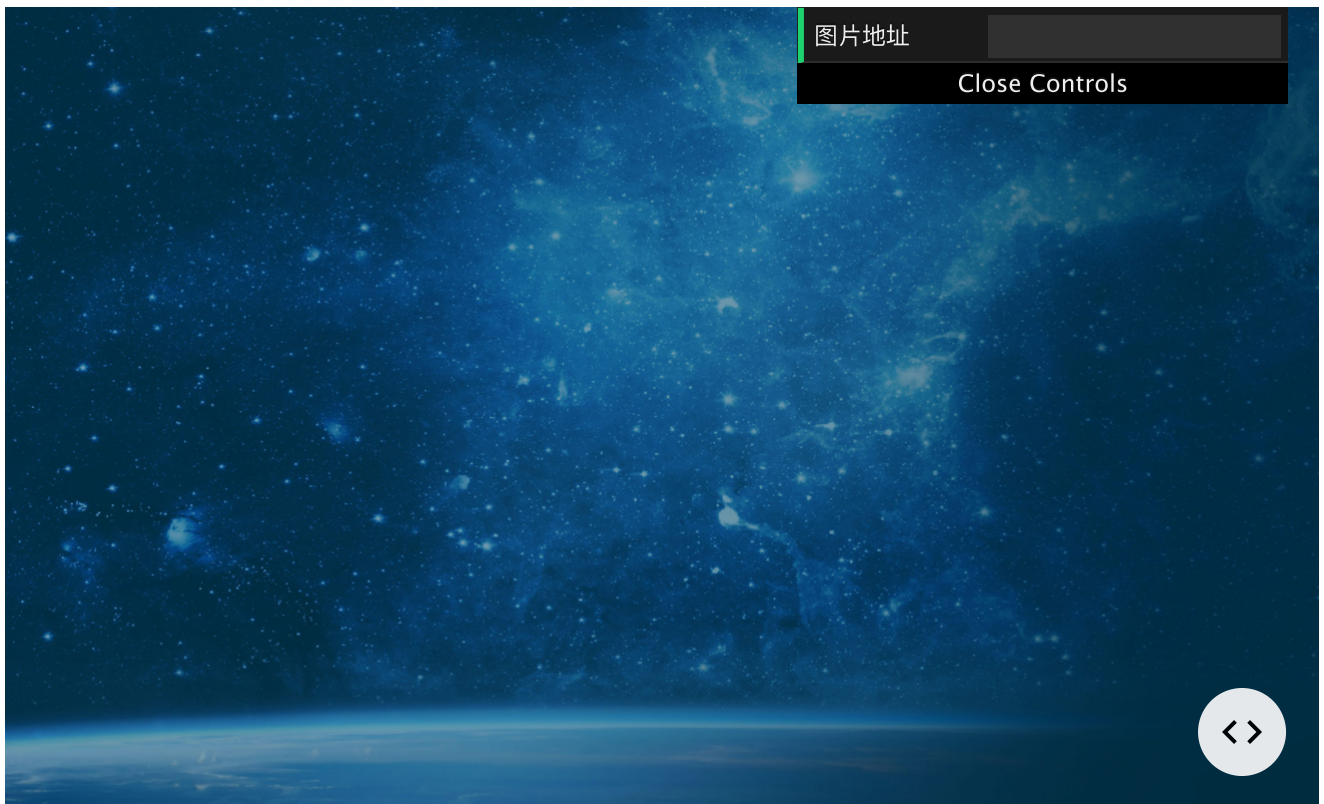
color

- 类型: **IColor**
- 描述: 颜色值
- 必填: **✓**

setBackgroundImage

设置背景图

样例:



定义:

```
function setBackgroundImage(imgUrl: string): void;
```

ts

用法:

```
ssp.setBackgroundImage('http://xx.com/xx.png');
```

js

参数:

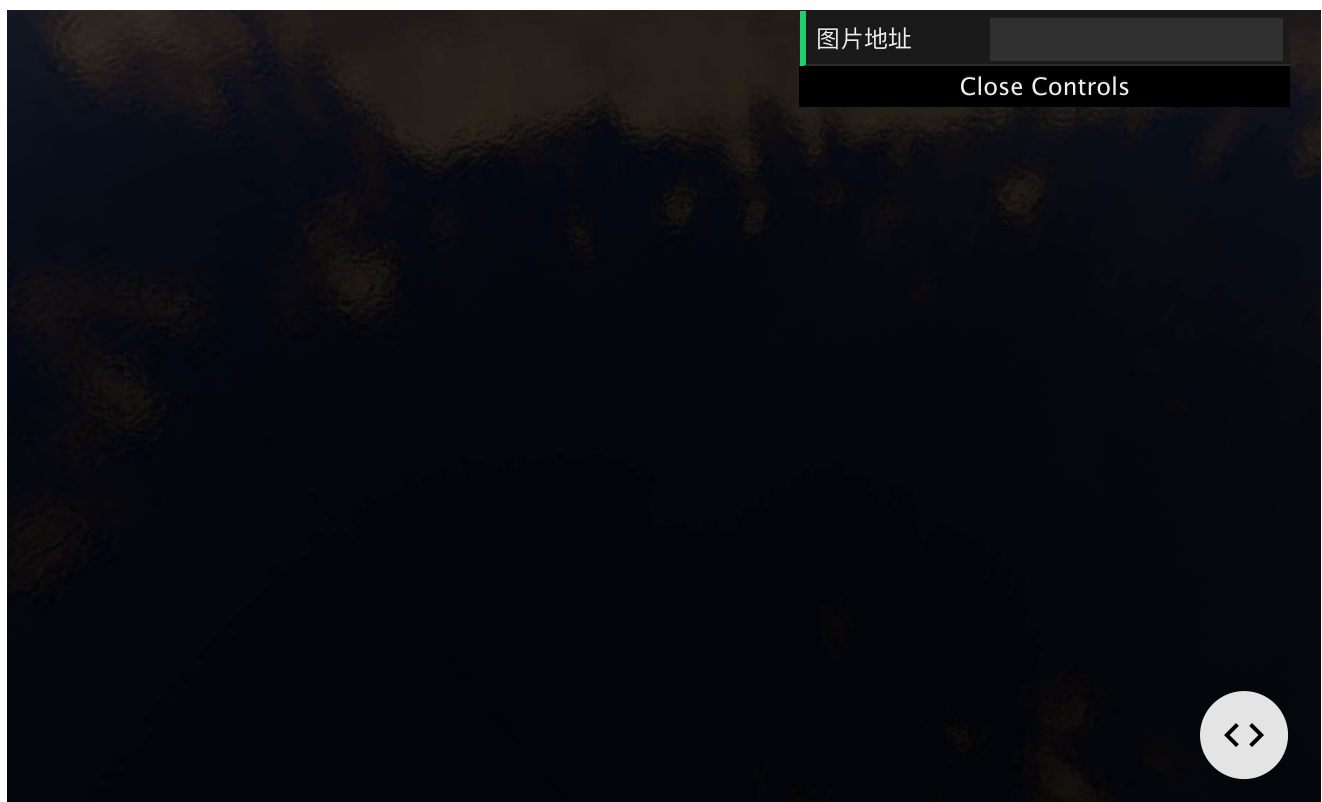
imgUrl

- 类型: string
- 描述: 图片路径
- 必填: ✓

setSphereSkyBackground

设置球体天空盒，球体天空盒的图片资源是一张全景图。

样例:



用法:

```
ssp.setSphereSkyBackground('http://xx.com/xx.png');
```

js

定义:

```
function setSphereSkyBackground(imgUrl: string): void;
```

ts

参数:

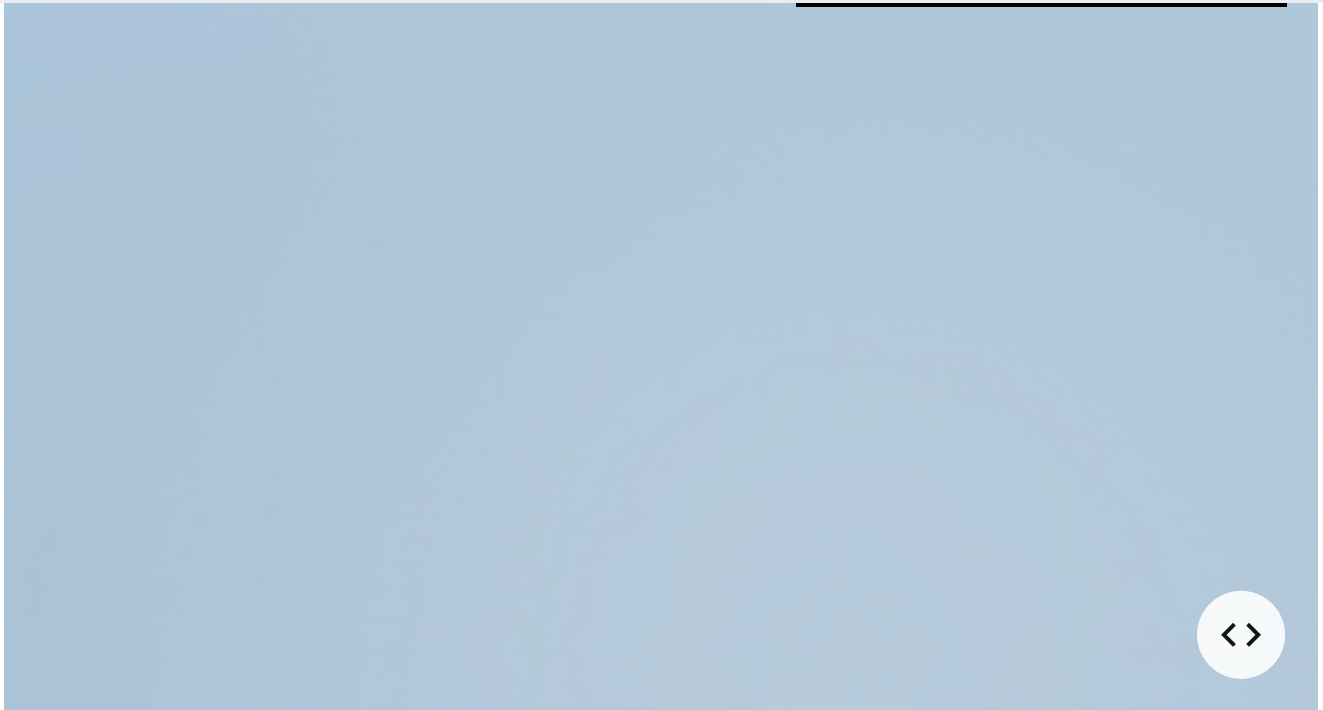
imgUrl

- 类型: string
- 描述: 图片路径。
- 必填: ✓

setSkyBackground

设置天空盒

样例:



定义:

```
function setSkyBackground(dirPath: string, fileNames: string[]): void;
```

ts

用法:

```
ssp.setSkyBackground(  
  // 文件夹路径  
  'http://www.xwbuilders.com:9018/soonspacejs/skybox/sunny/',  
  // 有序的图片名集合  
  ['px.jpg', 'nx.jpg', 'py.jpg', 'ny.jpg', 'pz.jpg', 'nz.jpg']  
);
```

js

参数:

dirPath

- 类型: string
- 描述: 图片文件夹路径。
- 必填: ✓

fileNames

- 类型: string[]
- 描述: 六张图片的名称集合。
- 必填: ✓

提示

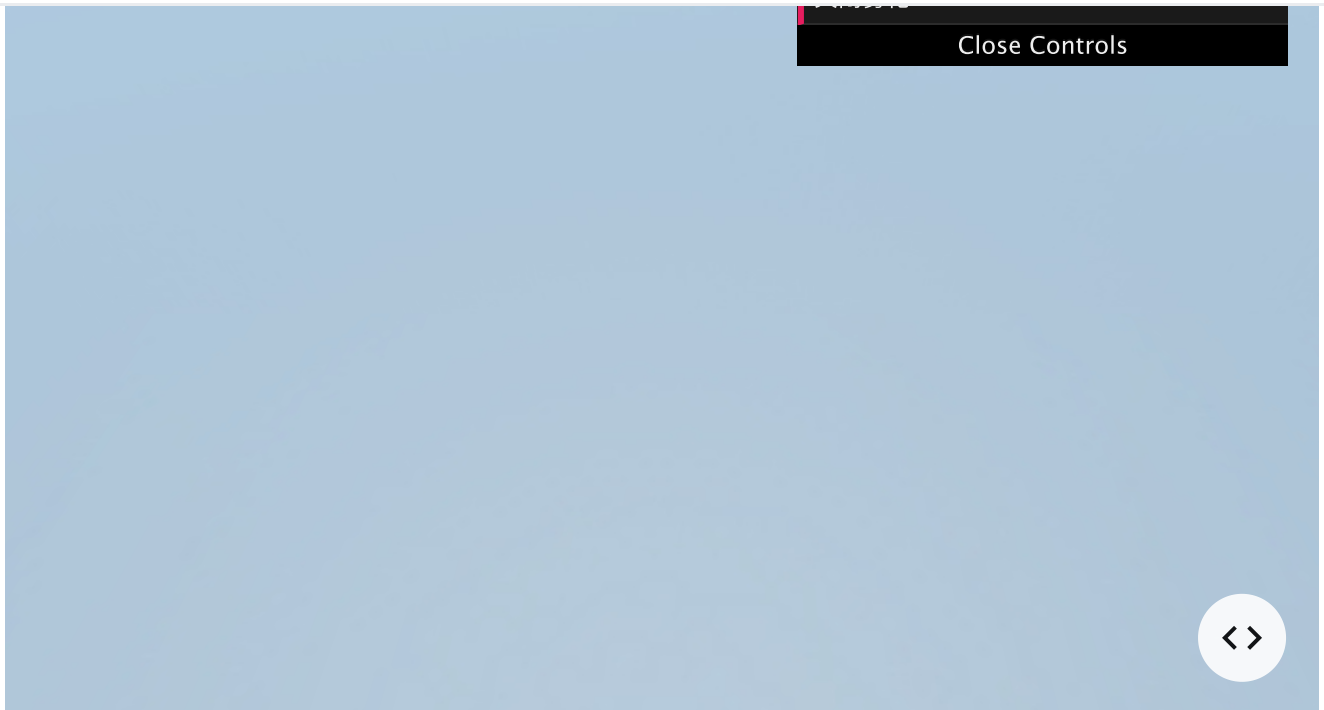
参数 `fileNames` 内的图片是有严格顺序的。依次为：

- 以空间坐标系为参考
x 轴正向 (px)、x 轴负向 (nx)、y 轴正向 (py)、y 轴负向 (ny)、z 轴正向 (pz)、z 轴负向 (nz)。
p: prev、n: next
- 以立方体盒子的六个面为参考
右 (right)、左 (left)、上 (top)、下 (bottom)、前 (front)、后 (back)。

openSceneFog

开启场景雾化

样例：



定义:

```
interface FogOptions {  
  color?: IColor;  
  near?: number;  
  far?: number;  
}  
  
function openSceneFog(options?: FogOptions): void;
```

ts

用法:

```
ssp.openSceneFog({ color: '0xcce0ff', near: 1, far: 1000 });
```

js

参数:

options

- 类型: **FogOptions**
- 描述: 配置选项

FogOptions

属性	描述	类型	必填	默认值
color	颜色	IColor	✗	0xcce0ff
near	起始位置 (距离相机)	number	✗	500
far	结束位置 (距离相机)	number	✗	50000

closeSceneFog

关闭场景雾化

定义:

```
function closeSceneFog(): void;
```

ts

用法:

```
ssp.closeSceneFog();
```

js

getObjectLabelPos

获取对象的标签位置坐标

定义:

```
interface LabelOptions {  
  mode?: 'scene' | 'screen';  
  viewpoint?: FlyToViewpoint;
```

ts


```
function getObjectLabelPos(  
  object: BaseObject3D,  
  options?: LabelOptions  
): Position;
```

用法:

```
ssp.getObjectLabelPos(object, { viewpoint: 'front', extendScale: 1.6 });
```

js

参数:

object

- 类型: BaseObject3D
- 描述: 模型对象
- 必填: ✓

options

- 类型: LabelOptions
- 描述: 配置选项
- 必填: ✗

LabelOptions

属性	描述	类型	必填	默认值
mode	相对于谁	scene screen	✗	scene
viewpoint	视角枚举	FlyToViewpoint	✗	frontTop
extendScale	延伸比例	number	✗	1.6

getOffsetByPosition

通过空间坐标点获取屏幕坐标点

定义:

```
function getOffsetByPosition(position: Position): { left: number; top: number }ts
```

用法:

```
ssp.getOffsetByPosition({ x: 100, y: 200, z: 300 });js
```

参数:

position

- 类型: **Position**
- 描述: 空间坐标
- 必填: ✓

getPositionByOffset

通过屏幕坐标点获取空间坐标点

定义:

```
function getPositionByOffset(offset: OffsetPoint, z?: number): Vector3;ts
```

用法:

```
ssp.getPositionByOffset({ offsetX: 10, offsetY: 10 });js
```

参数:

offset

- 类型: **OffsetPoint**
- 描述: 偏移量
- 必填: **✓**

z

- 类型: number
- 描述: z 轴, 默认值 0.1
- 必填: **✗**

addObject

添加对象

定义:

```
function addObject(object: Object3D, parent?: Object3D): void;
```

ts

用法:

```
ssp.addObject(object);
```

js

参数:

object

- 类型: Object3D
- 描述: 要添加的模型对象

parent

- 类型: Object3D
- 描述: 被添加对象的父级, 默认为整个场景
- 必填: **✗**

removeObject

移除对象

定义:

```
function removeObject(object: Object3D): void;
```

ts

用法:

```
ssp.removeObject(object);
```

js

参数:

object

- 类型: Object3D
- 描述: 要移除的模型对象
- 必填: **✓**

setHoverEnabled

设置开启鼠标悬浮

定义:

用法:

```
ssp.setHoverEnabled(true);
```

js

参数:

object

- 类型: boolean
- 描述: 是否开启
- 必填: ✓

render

手动渲染一次场景

定义:

```
function render(fn: Function): Promise<void>;
```

ts

用法:

```
// 同步
ssp
  .render(() => {
    model.visible = false;
  })
  .then(() => {
    console.log('场景渲染完成');
  });
```

js

```
ssp
    .render(() => {
      return new Promise((resolve) => {
        setTimeout(() => {
          model.position.x = 1000;
          resolve();
        }, 3000);
      });
    })
    .then(() => {
      console.log('场景渲染完成');
    });
```

参数:

fn

- 类型: Function
- 描述: 在渲染之前执行
- 必填: ✓

提示

fn 函数可以返回一个 Promise, 场景会在 fn 返回结果之后渲染

clearObject

清除除灯光外所有对象

定义:

```
function clearObject(): void;
```

ts

用法:

clearSignals

清除事件信号监听

定义:

```
function clearSignals(): void;
```

ts

用法:

```
ssp.clearSignals();
```

js

clear

清除所有对象

定义:

```
function clear(): void;
```

ts

用法:

```
ssp.clear();
```

js

dispose

定义:

```
function dispose(): void;
```

ts

用法:

```
ssp.dispose();
```

js

[← 模型操作](#)[相机 →](#)