

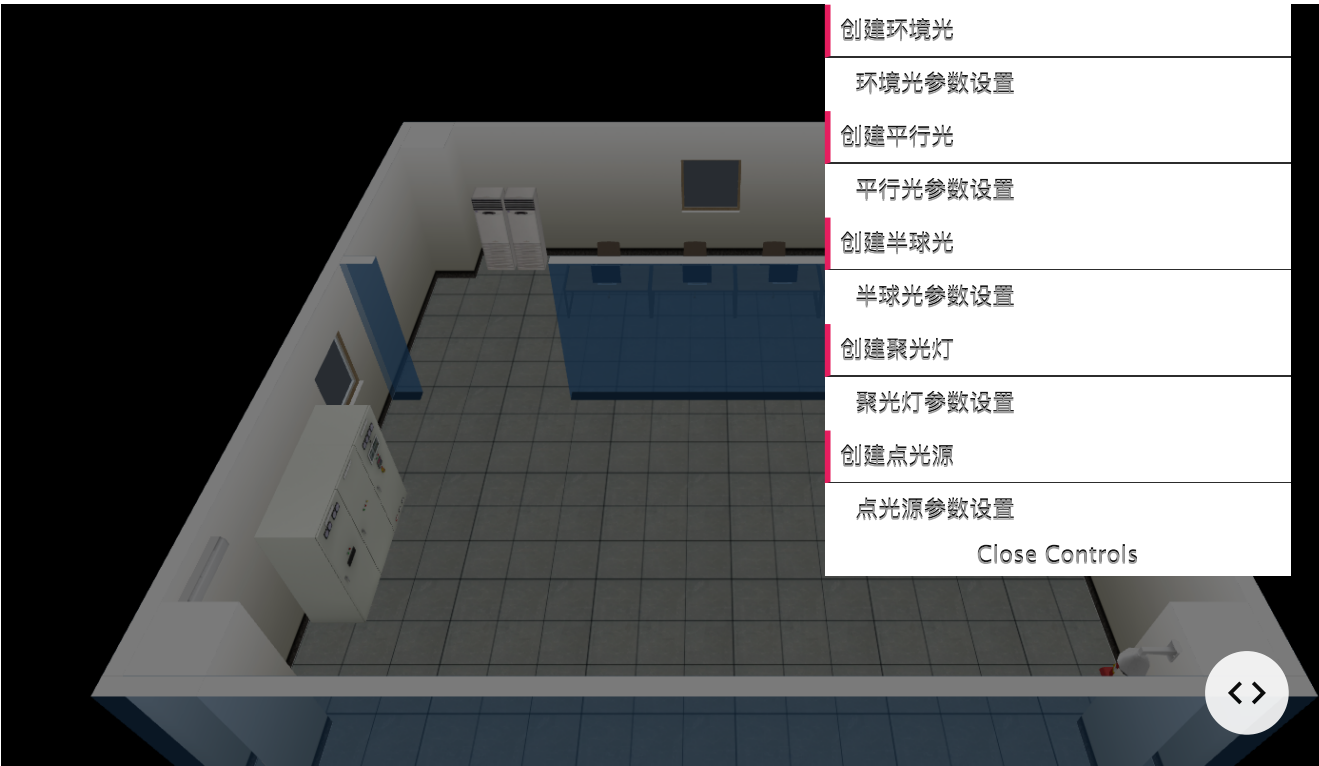
灯光

提示

`soonspacejs` 内部会在初始化时分别创建一个环境光、平行光、半球光。

`id` 依次是 `defaultAmbientLight`、`defaultDirectionalLight`、`defaultHemiLight`。

样例：



createAmbientLight

创建环境光

定义：

ts

```
interface BaseLightInfo {  
  id: string | number;  
  name?: string;  
  color?: IColor;  
  intensity?: number;  
}  
  
interface AmbientLightOptions extends BaseLightInfo {}  
  
function createAmbientLight(options: AmbientLightOptions): THREE.AmbientLight;
```

用法:

js

```
ssp.createAmbientLight({  
  id: 'ambientLight',  
  name: 'ambientLight',  
});
```

参数:

options

- 描述: 环境光配置项
- 必填: ✓
- 类型: AmbientLightOptions

AmbientLightOptions

属性	描述	类型	必填	默认值
id	唯一ID	string number	✓	
name	名称	string	✗	
color	颜色	IColor	✗	0x9a9a9a
intensity	光照强度	number	✗	1

setAmbientLight

设置环境光

定义:

```
function setAmbientLight(options: AmbientLightOptions): boolean;
```

ts

用法:

```
const isUpdated = ssp.setAmbientLight({  
  id: 'ambientLight',  
  color: 0x8a8a8a,  
  intensity: 0.5,  
});  
if (isUpdated) {  
  console.log('环境光配置更新成功');  
}
```

js

提示

`setAmbientLight` 与 `createAmbientLight` 的 `options` 完全一致。

`setAmbientLight` 用于更新场景已存在的光的配置, `createAmbientLight` 用于创建一个光。

createDirectionalLight

创建平行光

定义:

```
interface DirectionalLightOptions extends BaseLightInfo {  
  position?: Position;  
  target?: Position;  
  openShadow?: boolean;  
}  
  
function createDirectionalLight(  

```

ts

```
options: DirectionalLightOptions
): THREE.DirectionalLight;
```

用法:

```
ssp.createDirectionalLight({
  id: 'directionalLight',
  name: 'directionalLight',
});
```

js

参数:

options

- 描述: 平行光配置项
- 必填: ✓
- 类型: DirectionalLightOptions

DirectionalLightOptions

属性	描述	类型	必填	默认值
id	唯一ID	string number	✓	
name	名称	string	✗	
color	颜色	IColor	✗	0xffffffff
intensity	光照强度	number	✗	1
position	光源的位置	Position	✗	{ x: 0, y: 1000, z: 0 }
target	光照向的位置	Position	✗	{ x: 0, y: -100, z: 0 }
openShadow	是否开启阴影	boolean	✗	false

setDirectionalLight

设置平行光

定义:

```
function setDirectionalLight(options: DirectionalLightOptions): boolean;
```

ts

用法:

```
const isUpdated = ssp.setDirectionalLight({
  id: 'directionalLight',
  color: 0x8a8a8a,
  intensity: 0.5,
});
if (isUpdated) {
  console.log('平行光配置更新成功');
}
```

js

createHemisphereLight

创建半球光

定义:

```
interface HemisphereLightOptions extends BaseLightInfo {
  skyColor?: IColor;
  groundColor?: IColor;
  position?: Position;
}

function createHemisphereLight(
  options: HemisphereLightOptions
): THREE.HemisphereLight;
```

ts

用法:

js

```
ssp.createHemisphereLight({
  id: 'hemiLight',
  name: 'hemiLight',
  intensity: 0.1,
});
```

参数:

options

- 描述: 半球光配置项
- 必填: 
- 类型: HemisphereLightOptions

HemisphereLightOptions

属性	描述	类型	必填	默认值
id	唯一ID	string number		
name	名称	string		
intensity	光照强度	number		1
skyColor	天空颜色	IColor		0xffffffff
groundColor	地面颜色	IColor		0xdddddd
position	光的朝向位置	Position		{ x: 0, y: 0, z: 0 }

setHemisphereLight

设置半球光

定义:

ts

```
function setHemisphereLight(options: HemisphereLightOptions): boolean;
```

用法:

```
const isUpdated = ssp.setHemisphereLight({
  id: 'hemiLight',
  color: 0x8a8a8a,
  intensity: 0.5,
});
if (isUpdated) {
  console.log('半球光配置更新成功');
}
```

js

createSpotLight

创建聚光灯

定义:

```
interface SpotLightOptions extends BaseLightInfo {
  angle?: number;
  position?: Position;
  target?: Position;
  openShadow?: boolean;
}

function createSpotLight(options: SpotLightOptions): THREE.SpotLight;
```

ts

用法:

```
ssp.createSpotLight({
  id: 'spotLight',
  name: 'spotLight',
});
```

js

参数:

options

- 描述: 聚光灯配置项
- 必填: ✓

- 类型: SpotLightOptions

SpotLightOptions

属性	描述	类型	必填	默认值
id	唯一ID	string number	✓	
name	名称	string	✗	
color	颜色	IColor	✗	0xffffffff
intensity	光照强度	number	✗	1
angle	光照方向扩散的角度（最大值为90）	number	✗	45
position	光源的位置	Position	✗	{ x: 0, y: 500, z: 0 }
target	光照向的位置	Position	✗	{ x: 0, y: 0, z: 0 }
openShadow	是否开启阴影	boolean	✗	false

setSpotLight

设置半球光

定义:

```
function setSpotLight(options: SpotLightOptions): boolean;
```

用法:

```
const isUpdated = ssp.setSpotLight({
  id: 'spotLight',
  color: 0x8a8a8a,
  intensity: 0.5,
});
if (isUpdated) {
```



```
console.log('聚光灯配置更新成功');
}
```

createPointLight

创建点光源

定义:

```
interface PointLightOptions extends BaseLightInfo {
  position?: Position;
}

function createPointLight(options: PointLightOptions): THREE.PointLight;
```

用法:

```
ssp.createPointLight({ id: 'pointLight', name: 'pointLight' });
```

参数:

options

- 描述: 点光源配置项
- 必填:
- 类型: PointLightOptions

PointLightOptions

属性	描述	类型	必填	默认值
id	唯一ID	string number		
name	名称	string		
color	颜色	IColor		0xffffffff
intensity	光照强度	number		1

属性	描述	类型	必填	默认值
<code>position</code>	光源的位置	<code>Position</code>	✗	<code>{ x: 0, y: 500, z: 0 }</code>

setPoinntLight

设置点光源

定义：

```
function setPoinntLight(options: PointLightOptions): boolean;
```

ts

用法：

```
const isUpdated = ssp.setPoinntLight({
  id: 'pointLight',
  color: 0x8a8a8a,
  intensity: 0.5,
});
if (isUpdated) {
  console.log('点光源配置更新成功');
}
```

js

getLightById

根据 id 查询 Light 对象

定义：

```
function getLightById<T extends Light>(id: BaseObject3DInfo['id']): T | null;
```

ts

用法：

```
const pointLight = ssp.getLightById('pointLight');
```

js

removeLightById

根据 id 移除 Light 对象

定义:

```
function removeLightById(id: BaseObject3DInfo['id']): boolean;
```

ts

用法:

```
const isRemoved = ssp.removeLightById('pointLight');
if (isRemoved) {
  console.log('灯光移除成功!!!');
}
```

js

clearLight

清空 Light 对象

定义:

```
function clearLight(): void;
```

ts

用法:

```
ssp.clearLight();
```

js

showAllLight

显示所有光

定义:

```
function showAllLight(): void;
```

ts

用法:

```
ssp.showAllLight();
```

js

hideAllLight

隐藏所有光

定义:

```
function hideAllLight(): void;
```

ts

用法:

```
ssp.hideAllLight();
```

js

