

相机

getCameraViewpoint

获取当前相机视角数据

定义：

```
function getCameraViewpoint(): CameraViewpointData;
```

ts

用法：

```
const cameraViewpointData = ssp.getCameraViewpoint();  
console.log('cameraViewpointData', cameraViewpointData);
```

js

setCameraViewpoint

设置当前相机视角数据，数据由 `getCameraViewpoint` 获取。

定义：

```
function setCameraViewpoint(viewpointData: CameraViewpointData): void;
```

ts

用法

```
ssp.setCameraViewpoint(cameraViewpointData);
```

js

参数:

viewpointData

- 描述: 由 `getCameraViewpoint` 获取到的相机视角数据。
- 必填: ✓
- 类型: CameraViewpointData

CameraViewpointData

属性	描述	类型	必填	默认值
<code>position</code>	相机位置	<code>Position</code>	✓	
<code>rotation</code>	相机旋转弧度	<code>Rotation</code>	✓	

flyMainViewpoint

相机飞向主场景视角

定义:

```
function flyMainViewpoint(  
  viewpoint: FlyToViewpoint = 'frontTop',  
  options: FlyToObjOptions = {}  
): Promise<void>;
```

ts

用法:

```
ssp  
  .flyMainViewpoint('top')  
  .then((object) => console.log('flyMainViewpoint done', object))  
  .catch((err) => console.error(err));
```

js

参数:

viewpoint

- 描述: 相机朝向主场景的视角面
- 必填: **X**
- 类型: **FlyToViewpoint**
- 默认值: `frontTop`

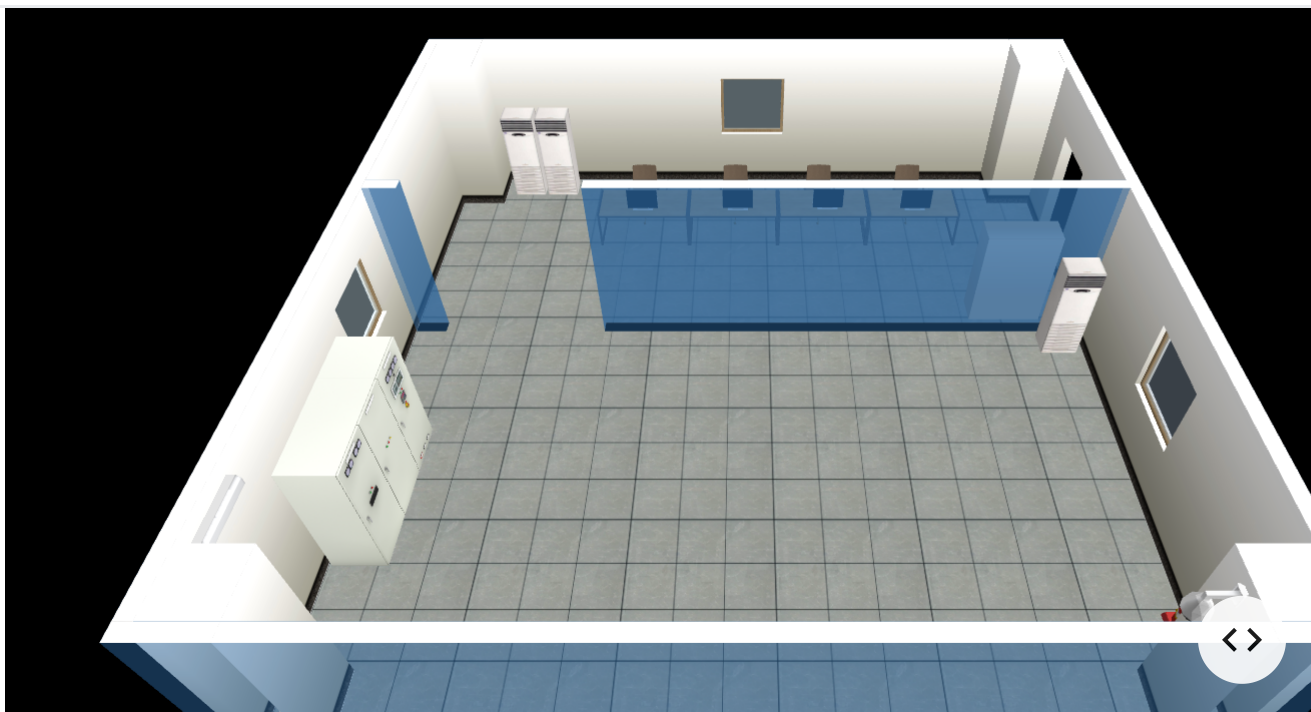
options

- 描述: 可配置参数
- 必填: **X**
- 类型: **FlyToObjOptions**
- 默认值: `frontTop`

flyToObj

相机飞向对象

样例:



定义:

```
interface FlyToObjOptions extends AnimationOptions {  
  padding?: number;  
}  
  
function flyToObj(  
  object: BaseObject3D | BaseMesh,  
  viewpoint: FlyToViewpoint = 'frontTop',  
  options: FlyToObjOptions = {}  
): Promise<void>;
```

ts

用法:

```
ssp  
  .flyToObj(  
    // object  
    sbm,  
    // viewpoint  
    'top',  
    // option  
    {  
      padding: 100,  
    }  
  )
```

js

```
.then((object) => console.log('flyToObj done', object))
.catch((err) => console.error(err));
```

参数:

object

- 描述: 相机飞向的空间对象
- 必填: ✓
- 类型: **BaseObject3D** | **BaseMesh**

viewpoint

- 描述: 相机朝向物体对象的视角面, **可选枚举**。
- 必填: ✗
- 类型: **FlyToViewpoint**
- 默认值: `frontTop`

option

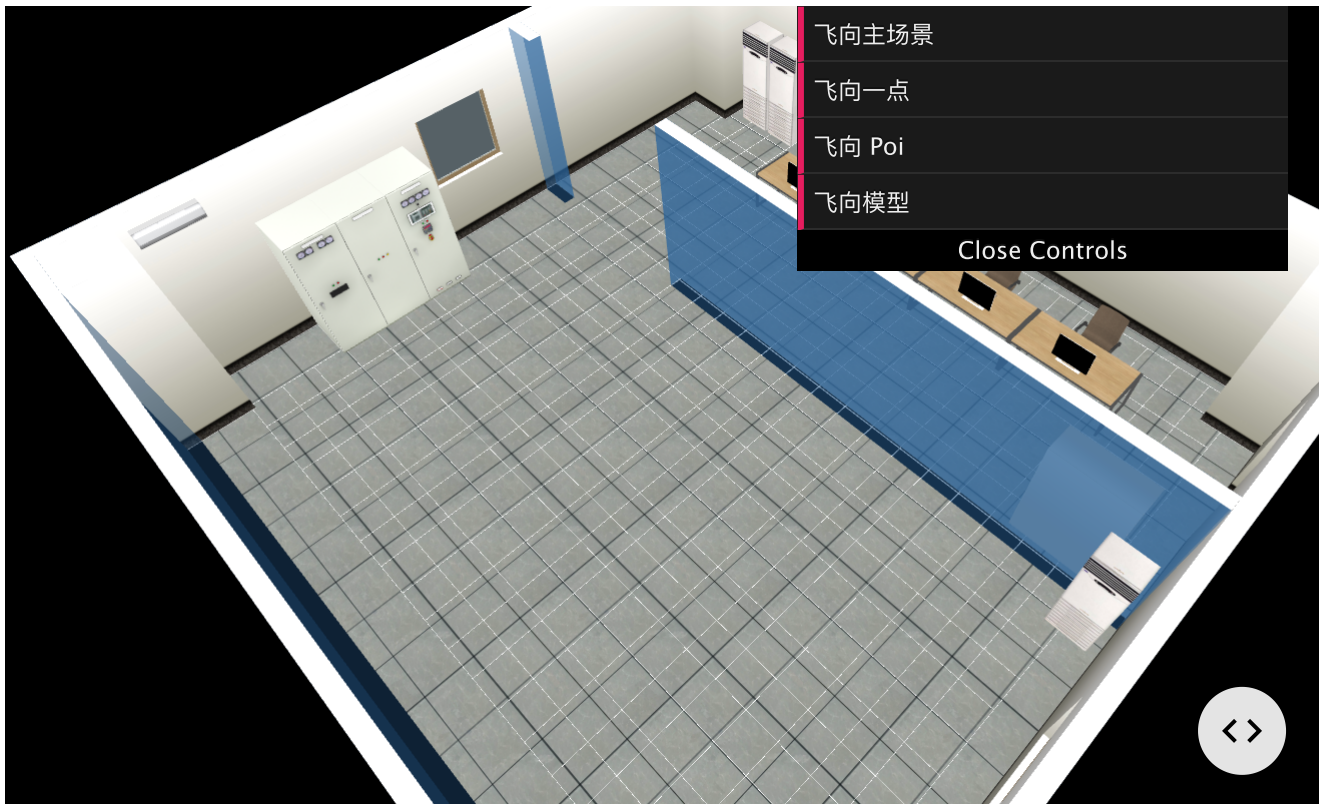
- 描述: 可配置项
- 必填: ✗
- 类型: **FlyToObjOptions**

FlyToObjOptions

属性	描述	类型	必填	默认值
<code>padding</code>	视角后飞向的偏移量	number	✗	0
<code>duration</code>	补间执行时长 (ms)	number	✗	1000
<code>delay</code>	补间开始前延时 (ms)	number	✗	0
<code>repeat</code>	循环	number boolean	✗	false

flyTo

样例:



定义:

```
function flyTo(  
  position: Position,  
  rotation: FlyToViewpoint | Rotation | Euler = 'frontTop',  
  options?: AnimationOptions  
): Promise<void>;
```

ts

用法:

```
ssp.flyTo(  
  // position  
  {  
    x: 0,  
    y: 1000,  
    z: 1000  
  },
```

js

```
    x: 0,  
    y: 0,  
    z: 0  
  },  
  // option  
  {  
    duration: 1000  
  }  
)  
  .then(res => console.log(res))  
  .catch(err => console.error(err))
```

参数:

position

- 描述: 相机飞向的坐标点
- 必填: ✓
- 类型: **Position**

rotation

- 描述: 相机的旋转弧度
- 必填: ✗
- 类型: **Viewpoint | Rotation**
- 默认值: `frontTop`

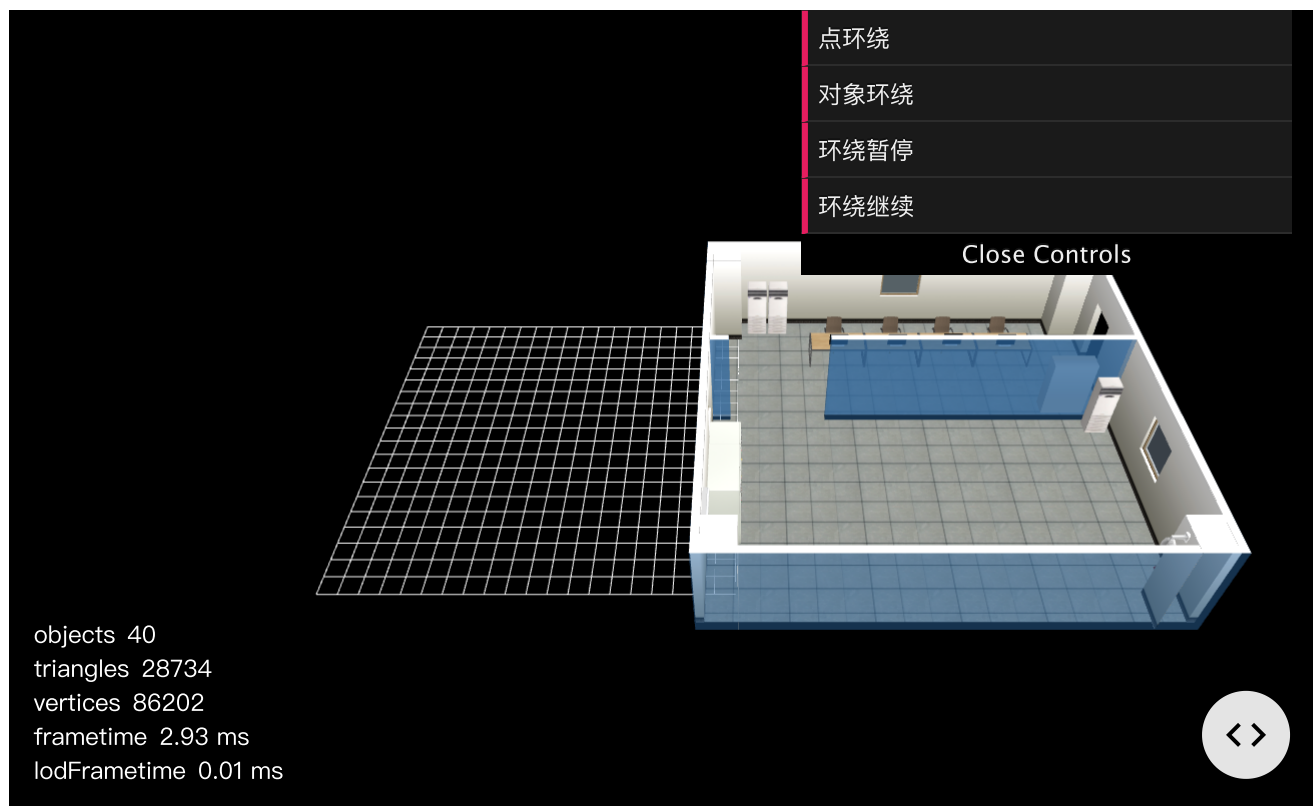
option

- 描述: 可配置项
- 类型: **AnimationOptions**
- 必填: ✗
- 默认值: `{}`

surroundOnTarget

围绕一个目标点旋转

样例:



定义:

```
interface SurroundOptions {  
  speed?: number  
  angle?: number  
  onStart?: <TweenType>(tween: TweenType) => {}  
}  
  
function surroundOnTarget(  
  target: Position,  
  options: SurroundOptions = {}  
): Promise<void>;
```

ts

用法:

```
ssp  
  .surroundOnTarget(  
    // ...
```

js


```
    x: 0,
    y: 0,
    z: 0,
  },
  // option
  {
    angle: 360,
    speed: 2,
    onStart: tween => {
      console.log('旋转动画对象', tween)
    }
  }
)
.then(() => console.log('surroundOnTarget done'))
.catch((err) => console.error(err));
```

参数:

target

- 描述: 目标点
- 类型: **Position**
- 必填: **✓**

options

- 描述: 可配置项
- 必填: **✗**
- 类型: SurroundOptions
- 默认值: `{}`

SurroundOptions

属性	描述	类型	必填	默认值
angle	旋转角度	number	✗	360
speed	旋转速度	number	✗	1
onStart	旋转开始回调函数	function(tween){}	✗	

surroundOnObject

围绕一个目标对象旋转

定义:

```
function surroundOnObject(  
  object: BaseObject3D | BaseMesh,  
  options: SurroundOptions = {}  
) : Promise<void>;
```

ts

用法:

```
ssp  
  .surroundOnTarget(  
    // object  
    ssp.getSbmById('xxx'),  
    // option  
    {  
      angle: 360,  
      speed: 2,  
      onStart: tween => {  
        console.log('旋转动画对象', tween)  
      }  
    }  
  )  
  .then(() => console.log('surroundOnTarget done'))  
  .catch((err) => console.error(err));
```

js

参数:

object

- 描述: 目标对象



options

- 描述: 可配置项
- 必填: **✗**
- 类型: **SurroundOptions**
- 默认值: `{}`

← **场景操作**

灯光 →