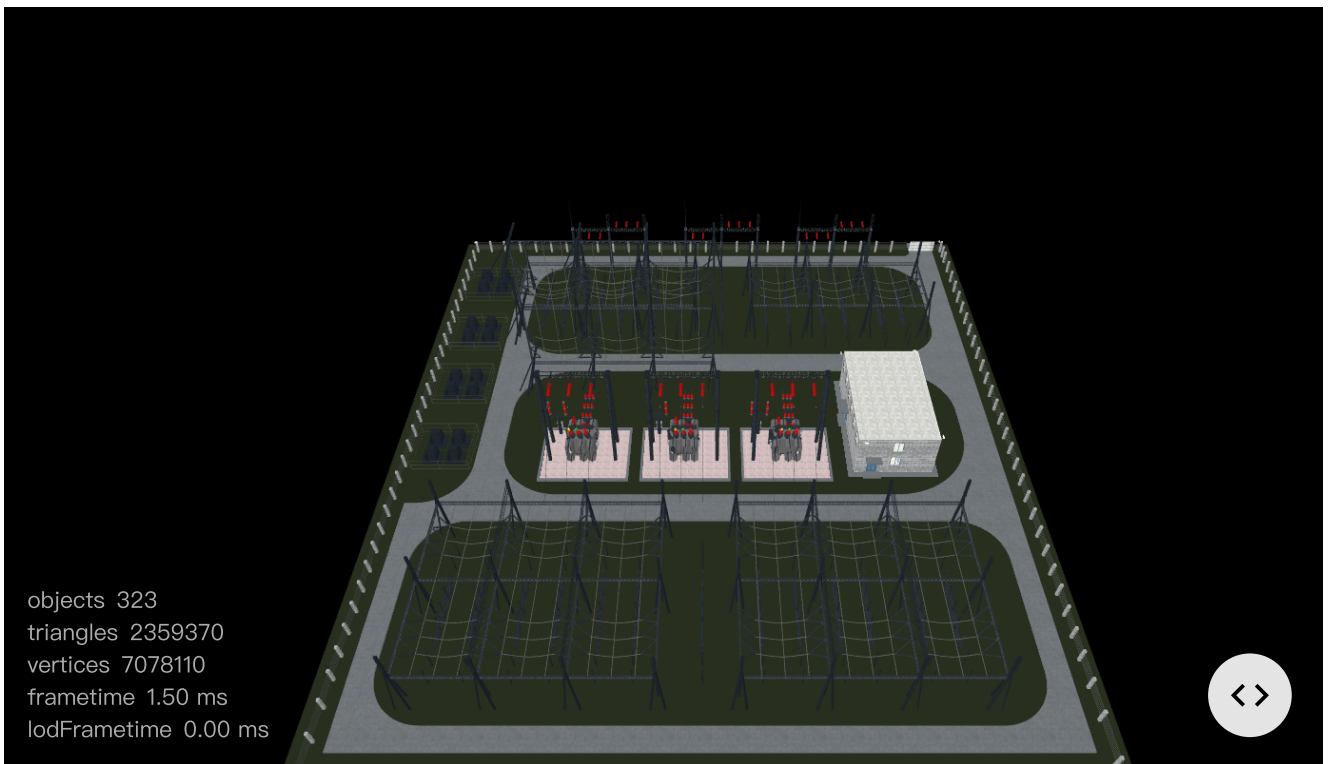


soonmanager-sync

npm@latest v2.2.0

空间平台 [🔗](#) 生产的场景加载及数据读取。

样例



安装

```
npm install @soonspacejs/plugin-soonmanager-sync -S  
# or  
yarn add @soonspacejs/plugin-soonmanager-sync -S
```

sh

使用方法

```
import SoonSpace from 'soonspacejs';
import SoonmanagerSyncPlugin from '@soonspacejs/plugin-soonmanager-sync';

const ssp = new SoonSpace({
  el: '#view',
  options: {},
  events: {},
});

const soonmanagerSync = ssp.registerPlugin(
  SoonmanagerSyncPlugin,
  'soonmanagerSync'
);
console.log(soonmanagerSync);
```

方法

setBaseUrl

设置基础路径

定义

```
function setBaseUrl(url: string): void;
```

用法

```
soonmanagerSync.setBaseUrl('http://xxx.com/back-resource/');
```

注意

必须要先调用 `setBaseUrl` , 才能加载效果、模型资源...

由于内部将 `url` 与 资源路径进行拼接获取资源, 所以参数的结尾需要加上斜线防止链接出错.

setGlobalSetting

同步背景、环境光、平行光、雾化效果

定义

```
interface SMYFogOptions extends FogOptions {
  visible: boolean;
}

interface GlobalSetting {
  color: IColor;
  ambientLight: AmbientLightOptions;
  hemisphereLight: HemisphereLightOptions;
  directionalLight: DirectionalLightOptions;
  fog: SMYFogOptions;
}

function setGlobalSetting(setting?: GlobalSetting): void;
```

ts

用法

```
soonmanagerSync.setGlobalSetting({
  color: '#f65',
  ambientLight: {
    id: 'sm_ambientLight',
    color: '#9a9a9a',
    intensity: 1,
  },
  hemisphereLight: {
    id: 'hemisphereLight',
    skyColor: '#ffffff',
    groundColor: '#dddddd',
    intensity: 1,
    position: {
      x: 0,
      y: 0,
      z: 0,
    },
  },
  directionalLight: {
    id: 'sm_directionalLight',
    color: '#ffffff',
    intensity: 1,
    position: {
      x: 0,
      y: 500,
      z: 0,
    },
  },
});
```

js

```
    },
    target: {
      x: 0,
      y: -100,
      z: 0,
    },
  },
  fog: {
    color: 0xcce0ff,
    near: 500,
    far: 50000,
    visible: true,
  },
});
```

提示

如果未传入 setting, 则获取 SoonManager 平台上保存的设置

参数

setting

- 描述: 全局效果的对象集合
- 必填: ✗
- 类型: `GlobalSetting`

GlobalSetting

属性	描述	类型	必填	默认值
color	背景颜色	IColor	✓	
ambientLight	环境光	AmbientLightOptions	✓	
hemisphereLight	半球光	HemisphereLightOptions	✓	
directionalLight	平行光	DirectionalLightOptions	✓	
fog	雾化效果	SMYFogOptions	✓	

loadScene

加载场景模型

定义

```
export interface LoadSceneOptions {  
  targetId?: number | number[];  
  targetLevel?: number;  
  isIdleRest?: boolean;  
  loadSceneAllSuccess?: () => void;  
}  
  
function loadScene(options?: LoadSceneOptions): Promise<void>;
```

ts

用法

```
/**  
 * 指定开始加载的树节点 ID  
 * 未指定或指定为空，从根节点开始加载场景  
 * 指定的 ID 无法命中节点，不会加载任何模型  
 * targetId: xxxxx | [xxxxx],  
 */  
  
/**  
 * 从指定节点开始，向下加载的层级深度。  
 * 等于 1 时只加载根节点  
 * 不设置时加载全部层级子集  
 * targetLevel: 2,  
 */  
  
/**  
 * 是否利用页面交互空闲时间去加载剩余模型，当 targetLevel 大于1或未指定时生效  
 * 可以提升初始化时，用户的交互等待时间  
 * isIdleRest: true  
 */  
  
/**  
 * 当前 loadScene 加载模型全部完成的回调  
 * loadSceneAllSuccess: () => {}  
 */  
soonmanagerSync  
  .loadScene({  
    isIdleRest: true,  
    loadSceneAllSuccess: () => {  
      console.log('全部模型加载完成');  
    },  
  },  
  )
```

js

```
.then(() => {  
  console.log('主层级加载完成');  
});
```

参数

options

- 描述: 场景加载选项
- 必填: ✗
- 类型: `LoadSceneOptions`

LoadSceneOptions

属性	描述	类型	必填	默认值
targetId	开始加载的目标 id	number number[]	✗	
targetLevel	从指定节点开始，向下加载的的层级深度	number	✗	
isIdleRest	否利用页面交互空闲时间去加载剩余模型	boolean	✗	false
loadSceneAllSuccess	所有模型加载完成的回调	Function	✗	function(){}