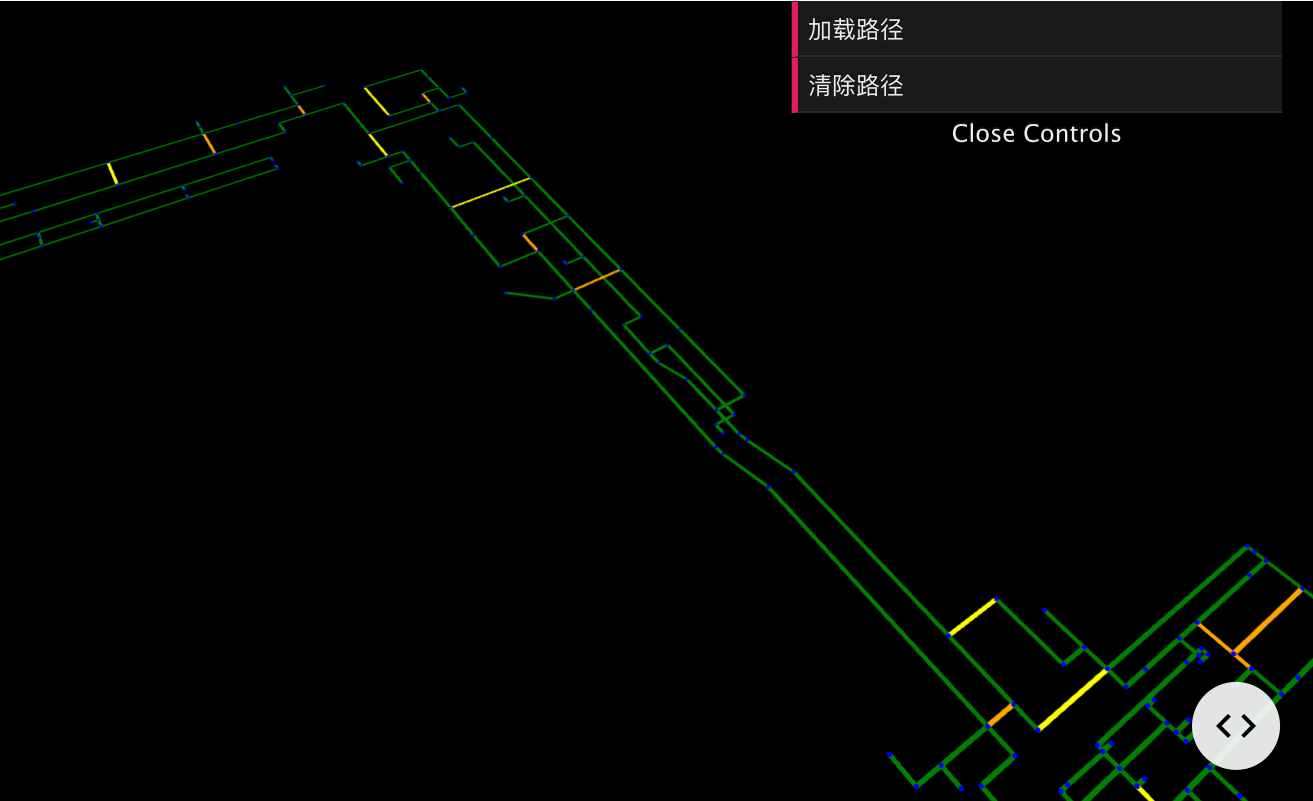


拓扑路径

样例：



createTopology

创建 `topology` 对象。

定义：

```
interface TopologyEffectInfo {  
  linkWidth?: number;  
  renderNode?: boolean;  
  nodeColor?: IColor;  
  renderLink?: boolean;  
  linkColor?: IColor | IColor[];  
  imgUrl?: LinkInfo['imgUrl'];  
  animation?: LinkInfo['animation'];  
}
```

```
interface TopologyNodeInfo {
  id: BaseObject3DInfo['id'];
  name?: BaseObject3DInfo['name'];
  position: Position;
  graphs?: TopologyNodeGraph[];
}

interface TopologyNodeGraph {
  targetNodeId: string
  linkInfo: {
    id: string
    name: string
  }
  passable: number
  length?: number
}

interface TopologyInfo extends BaseObject3DInfo, TopologyEffectInfo {
  type: TopologyType;
  nodes: TopologyNodeInfo[];
}

function createTopology(topologyInfo: TopologyInfo): Topology;
```

用法:

```
const topology = ssp.createTopology({
  id: 'topology_1',
  name: 'topology_1_name',
  type: 'line',
  nodes: [
    {
      id: 'node1',
      position: { x: 0, y: 1, z: 0 },
    },
    {
      id: 'node2',
      position: { x: 0, y: 1, z: 100 },
    },
    {
      id: 'node1',
```

js

```
    id: 'node1',
    position: { x: 100, y: 1, z: -100 },
  },
],
renderNode: true,
});
```

参数:

topologyInfo

- 描述: 实例路径对象所需信息
- 类型: TopologyInfo
- 必填: ✓

TopologyInfo

属性	描述	类型	必填	默认值
id	唯一ID	string number	✓	
name	名称	string	✗	
nodes	节点坐标集合	TopologyNodeInfo[]	✓	
type	路径类型	line network	✓	
linkWidth	线宽	number	✗	20
renderLink	是否渲染连接线	boolean	✗	true
linkColor	连接线颜色	IColor IColor[]	✗	0x00ff00
renderNode	是否渲染节点	boolean	✗	true
nodeColor	节点颜色	IColor	✗	0x0000ff
imgUrl	非纯色线时使用的图片资源路径	string	✗	null
animation	非纯色线时的流动动画	boolean AnimationOptions	✗	false

level	显示层级范围	Level	✗	{ max: null, min: null }
visible	是否可见	boolean	✗	true
position	位置坐标	Position	✗	{ x: 0, y: 0, z: 0 }
rotation	旋转弧度	Rotation	✗	{ x: 0, y: 0, z: 0 }
scale	缩放比例	Scale	✗	{ x: 1, y: 1, z: 1 }
userData	用户数据	any	✗	{}

`linkColor` 为数组时，有效长度是 4 个，分别对应 `passable` 的四个状态时路径颜色；为单个颜色时，表示所有路径颜色。

TopologyNodeInfo

属性	描述	类型	必填	默认值
id	节点唯一ID	string number	✓	
name	节点名称	string	✗	
position	节点坐标	Position	✓	
graphs	网结构信息	TopologyNodeGraph[]	✗	

TopologyNodeGraph

属性	描述	类型	必填	默认值
targetNodeId	目标 node ID	string number	✓	
linkInfo	路径信息	{ id: string, name: string }	✓	
passable	路径通行许可	0 1 2 3	✓	
length	起始点到目标点距离	number	✗	

getShortestPath

获取最短路径

定义:

```
interface ShortestPathInfo extends BaseObject3DInfo, TopologyEffectInfo {ts
  start: Position;
  end: Position;
}

function getShortestPath(
  topology: Topology,
  info: ShortestPathInfo
): Topology | null;
```

用法:

```
const shortestTopology = ssp.getShortestPath(topologyFromOther, {js
  start: { x: 0, y: 0, z: 0 },
  end: { x: 100, y: 0, z: 300 },
  id: 'shortestPath',
  linkColor: 'red',
  nodeColor: 'orange',
  imgUrl: '../..asstes/img/topology/arrow.png',
  animation: true,
});
```

参数:

topology

- **描述:** 拓扑路径对象，一般是从 gml 文件加载的拓扑路径图
- **类型:** `Topology`

info

- 描述: 最短路径信息
- 类型: `ShortestPathInfo`
- 必填: ✓

属性	描述	类型	必填	默认值
<code>start</code>	路径的起始点	<code>Position</code>	✓	
<code>end</code>	路径的结束点	<code>Position</code>	✓	

部分配置参考 `TopologyInfo`

getTopologyById

通过 `id` 查找

定义:

```
function getTopologyById(id: TopologyInfo['id']): Topology | null;
```

ts

用法:

```
const topology = ssp.getTopologyById('xxx');
```

js

getTopologyByName

通过 `name` 查找

定义:

用法:

```
const topologyList = ssp.getTopologyByName('xxx');
```

js

getAllTopology

获取所有 `Topology` 对象

定义:

```
function getAllTopology(): Topology[];
```

ts

用法:

```
const allTopologyList = ssp.getAllTopology();
```

js

getTopologyByUserDataProperty

通过 `userData` 属性查找

定义:

```
function getTopologyByUserDataProperty(  
  propNameOrFindFunc: string | UserDataPropertyFindFunc,  
  value?: any  
): Topology[];
```

ts

用法:

```
const topologyList = ssp.getTopologyByUserDataProperty('propKey', 'propVal')  
// or  
const topologyList = ssp.getTopologyByUserDataProperty(item => item['itemProp'])
```

参数:

propNameOrFindFunc

- 描述: `userData` 内属性名 或 `find` 函数
- 类型: string | function
- 必填: ✓

propValue

- 描述: `userData` 内属性值。
- 类型: any
- 必填: ✗

find 函数使用场景

```
topology.userData = {  
  people: {  
    name: 'xiaoming',  
    age: 18,  
  },  
};  
  
const topologyList = ssp.getTopologyByUserDataProperty(  
  (userData) => userData?.people?.name === 'xiaoming'  
);
```

removeTopologyById

定义:

```
function removeTopologyById(id: TopologyInfo['id']): boolean;
```

ts

用法:

```
ssp.removeTopologyById('xxx');
```

js

createTopologyToGroup

创建 `topology` 到一个组内。

定义:

```
function createTopologyToGroup(  
  groupInfo: GroupInfo,  
  topologyInfoList: TopologyInfo[]  
): Group;
```

ts

用法:

```
ssp.createTopologyToGroup(  
  // groupInfo  
  {  
    id: 'firstTopologyGroup',  
    name: 'name_firstTopologyGroup',  
    // ...  
  },  
  // topologyInfoList
```

js

参数

groupInfo

- 描述: 实例组对象所需信息
- 类型: **GroupInfo**
- 必填: ✓

topologyInfoList

- 描述: `topologyInfo` 集合
- 类型: **TopologyInfo[]**
- 必填: ✓

createTopologyFromGml

创建 Topology 组，从 gml 文件资源。

定义:

```
interface TopologyInfoForGml {  
  url: string;  
  id: BaseObject3DInfo['id'];  
  name?: BaseObject3DInfo['name'];  
  linkWidth?: number;  
  linkColor?: IColor;  
  renderNode?: boolean;  
  nodeColor?: IColor;  
}  
  
function createTopologyFromGml(  
  topologyInfo: TopologyInfoForGml  
): Promise<Topology>;
```

ts

用法:

```
.createTopologyFromGml({
  url: './tuobutujinzui.gml',
  id: 'gml_for_topology',
  name: 'gml_for_topology_name',
  linkWidth: 100,
  linkColor: 'blue',
  renderNode: true,
  nodeColor: 'green',
})
.then((topology) => {
  console.log(topology);
});
```

参数

topologyInfo

- 描述: `topologyInfo` 对象
- 类型: `TopologyInfoForGml`
- 必填: ✓

TopologyInfoForGml

属性	描述	类型	必填	默认值
<code>url</code>	gml 资源路径	string	✗	
<code>id</code>	路径对象唯一ID	string	✗	
<code>name</code>	路径对象名称	string	✗	
<code>linkWidth</code>	路径线宽	number	✗	20
<code>linkColor</code>	路径线颜色	<code>IColor</code>	✗	0x00ff00
<code>renderNode</code>	是否渲染路径节点	boolean	✗	true
<code>nodeColor</code>	节点颜色	<code>IColor</code>	✗	0x0000ff

createGroupForTopology

使用场景

与 `createTopologyToGroup` 不同，有些时候可能你还没有具体的 `topologyInfo` 数据，但你想提前创建一个批量管理的空组，当有数据时再使用 `addTopologyForGroup` 插入。

定义：

```
function createGroupForTopology(groupInfo: GroupInfo): Group;
```

ts

用法：

```
ssp.createGroupForTopology({  
  id: 'firstTopologyGroup',  
  name: 'name_firstTopologyGroup',  
  // ...  
});
```

js

参数

groupInfo

- 描述: 实例组对象所需信息
- 类型: `GroupInfo`
- 必填: ✓

addTopologyForGroup

向一个已经存在的组内添加 `topology` 对象。

定义：

```
groupedTopologyInfoList: TopologyInfo[]  
) : Group | null;
```

用法:

```
ssp.addTopologyForGroup(  
  // groupId  
  'firstTopologyGroup',  
  // topologyInfoList  
  [topologyInfo4, topologyInfo5],  
  // onProgress  
  (progress) => console.log('进度信息: ', progress)  
);
```

js

参数

groupId

- 描述: 组 `id`
- 类型: `groupId['id']`
- 必填: ✓

topologyInfoList

- 描述: `topologyInfo` 集合
- 类型: `topologyinfo[]`
- 必填: ✓

getTopologyGroupById

通过 `id` 查找 `topology` 组

定义:

用法:

```
const group = ssp.getTopologyGroupById('firstTopologyGroup');
```

js

getTopologyGroupName

通过 `name` 查找 `topology` 组

定义:

```
function getTopologyGroupName(name: string): Group[];
```

ts

用法:

```
const groupList = ssp.getTopologyGroupName('name_firstTopologyGroup');
```

js

removeTopologyGroupById

通过 `id` 移除 `topology` 组

定义:

```
function removeTopologyGroupById(id: GroupInfo['id']): boolean;
```

ts

用法:

clearTopology

清除当前场景内所有 `topology` 对象。

定义：

```
function clearTopology(): void;
```

ts

用法：

```
ssp.clearTopology();
```

js

showAllTopology

显示当前场景内所有 `topology` 对象。

定义：

```
function showAllTopology(): void;
```

ts

用法：

```
ssp.showAllTopology();
```

js

hideAllTopology

定义:

```
function hideAllTopology(): void;
```

ts

用法:

```
ssp.hideAllTopology();
```

js

[← 空间画布对象](#)[控制器 →](#)