SuperMap\_iClient3D\_VUE\_10i(2021)

（内测版）

北京超图软件股份有限公司

2021年04月

1

# 开发指南

通过本章节的学习，可以快速了解如何安装和使用SuperMap\_iClient3D\_VUE\_10i(2020)。

* **快速安装教程**
* **快速使用教程**

## 安装

### Npm安装

1. 开始之前请确保电脑已经安装了node环境。
2. 安装vue3并新建Vue项目。对不熟悉Vue新建项目的请参考[Vue官网](https://www.vue3js.cn/docs/zh/guide/installation.html" \l "%E5%91%BD%E4%BB%A4%E8%A1%8C%E5%B7%A5%E5%85%B7-cli)
3. 进入新建的vue项目安装vue组件，推荐使用 npm 的方式安装，它能更好地和 webpack 打包工具配合使用。
4. 命令如下：npm install @supermap/iclient3d-vue-for-webgl --save-D

### CDN

1. 引入完整带界面组件：引用自定义界面的组件请参考后面自定义组件

<!-- 引入样式 -->

<link rel="stylesheet" href="https://www.supermapol.com/earth/vue-iEarth/examples/dist/components.css">

<!-- 引入组件库 -->

<script src="https://www.supermapol.com/earth/vue-iEarth/examples/dist/components.js"></script>

1. 页面编写方法：通过 CDN 的方式我们可以快速地使用 webgl3d写出范例——量算

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">

<!-- vue + element-plus-->

<script src="https://www.supermapol.com/earth/vue-iEarth/examples/public/js/vue.global.prod.js"></script>

<link rel="stylesheet" href="https://www.supermapol.com/earth/vue-iEarth/examples/public/element-plus/index.css">

<script src="https://www.supermapol.com/earth/vue-iEarth/examples/public/element-plus/index.full.js"></script>

<!-- cesium -->

<link href="https://www.supermapol.com/earth/vue-iEarth/examples/public/Cesium/Widgets/widgets.css"

rel="stylesheet">

<script src="https://www.supermapol.com/earth/vue-iEarth/examples/public/Cesium/Cesium.js"></script>

<!-- 组件包 -->

<link href="https://www.supermapol.com/earth/vue-iEarth/examples/dist/components.css" rel="stylesheet">

<script src="https://www.supermapol.com/earth/vue-iEarth/examples/dist/components.js"></script>

<title>完整组件-CDN引入-demo</title>

</head>

<body>

<div id="app">

<sm3d-viewer scene-url="http://www.supermapol.com/realspace/services/3D-ZF\_normal/rest/realspace">

<sm3d-measure></sm3d-measure>

</sm3d-viewer>

</div>

<script>

const app = Vue.createApp({});

app.use(webgl3d);

app.mount("#app");

</script>

</body>

<!-- 根据使用具体组件的需要引入其他第三方依赖 -->

<!-- <script src="https://cdn.jsdelivr.net/npm/echarts@5.0.2/dist/echarts.min.js" async></script>

<script src="https://www.supermapol.com/earth/vue-iEarth/examples/public/js/axios.min.js" async></script> -->

</html>

## 快速使用

### 第三方依赖安装

1. 可以根据需要选择安装，也可以使用CDN全局引入：npm install element-plus --save-d

### 引入 iclient3d-vue-for-webgl

1. 方法如下：
2. 在index.html里面引入组件所依赖的cesium包。
3. 在node\_module里找到此组件的安装包。
4. 复制public里需要的资源到工程目录public文件里。
5. 引入cesium等资源文件，例如：

index.html：

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- 此处注意引入路径，vue-cli可能需要把public改为. -->

<link href="public/Cesium/Widgets/widgets.css" rel="stylesheet">

<script src="public/Cesium/Cesium.js" ></script>

<title>webgl3d</title>

</head>

<body>

<div id="app"></div>

<script type="module" src="/src/main.js"></script>

<!-- 以上main.js引入是vite项目才需要，cli项目需删除 -->

<script src="public/js/echarts.min.js" async></script>

<script src="public/js/axios.min.js" ></script>

</body>

</html>

注意：

1. 可以引入整个 iclient3d-vue-for-webgl（后面都简称webgl3d），或是根据需要引入部分组件。
2. 如何引入完整的 webgl3d

在 main.js 中写入以下内容：

import { createApp } from 'vue'

import App from './App.vue'

const app = createApp(App);

// 完整引入第三方库，部分组件需要

import ElementPlus from 'element-plus';

import 'element-plus/lib/theme-chalk/index.css';

app.use(ElementPlus)

//import \* as echarts from 'echarts';

//window.echarts = echarts //挂载到window上，最好在html全局引入

// 引入webgl3d组件包

import '@supermap/iclient3d-vue-for-webgl/lib/theme/index.css'

import webgl3d from "@supermap/iclient3d-vue-for-webgl"

app.use(webgl3d)

app.mount('#app')

以上代码便完成了webgl3d 的引入。

在App.vue里测试代码：

<template>

<sm3d-viewer scene-url="http://www.supermapol.com/realspace/services/3D-ZF\_normal/rest/realspace">

<sm3d-measure></sm3d-measure>

</sm3d-viewer>

</template>

<script>

注意：

1. 样式文件需要单独引入。
2. 引入css遇到未知字符错误难以解决时，可以把css放到index.html里引入使用。即复制lib下theme到public。
3. 按需引入：借助 [babel-plugin-import](https://github.com/ant-design/babel-plugin-import)，我们可以只引入需要的组件，以达到减小项目体积的目的。

// 按需加载

// 引入组件前需要先全局配置

import locale from '@supermap/iclient3d-vue-for-webgl/lib/lang/resoueceCN.js' //引入语言中文包

import initDrag from "@supermap/iclient3d-vue-for-webgl/lib/initDrag.js"; //拖拽功能默认使用，需要添加

app.config.globalProperties.Resource = locale; //中文语言配置

initDrag(app);

// 引入需要的组件

import '@supermap/iclient3d-vue-for-webgl/lib/theme/index.css'

import terrainSlope from "@supermap/iclient3d-vue-for-webgl/lib/terrain-slope.js"

app.use(terrainSlope)

注意：

1. 按需引入也要引入全局css样式，若是引入报错，建议从入口index.html引入，复制组件包的css资源到工程目录下。
2. 按需引入第三方依赖：具体操作请参考第三方教程。
3. 第三方使用到的组件：

element-plus：Slider，DatePicker，Message

echarts ：LIneChart

### 全局配置

1. 语言默认为中文（除中文外目前还支持英文和日文）。需要其他语言配置操作如下

在main.js里写入以下内容：

import webgl3d from "@supermap/iclient3d-vue-for-webgl"

import locale from '@supermap/iclient3d-vue-for-webgl/lib/lang/resoueceEN.js' //英文

//import locale from '@supermap/iclient3d-vue-for-webgl/lib/lang/resoueceJA.js' //日文

app.use(webgl3d,locale)

### 自定义主题

1. 在App.vue的mounted里或其他全局位置写如下格式：

document.querySelector(':root').setAttribute('style', '--panel-bg-color: '+'#ffffff')

1. 目前支持修改的主题属性：

// 自定义主题

  --theme-bg-color: #3499e5; //主题色

  --panel-bg-color: rgb(249, 249, 249); //界面背景色

  --font-color: #333333; //字体颜色

  --font-family: "Microsoft YaHei", "Arial", "黑体", "宋体", sans-serif; //字体风格

  --border-color: #999999;  //边框颜色

  --content-font-size:14px;  //内容部分字体大小

  --title-font-size:16px;   //标贴部分字体大小

  --tip-font-size:12px;   //提示部分字体大小

  --panel-width:320px;   //界面宽度

  --panel-max-height:520px;   //界面最大高度

  --shadow-color:rgba(128, 128, 128, 0.5); //阴影

### 完整组件

1. 地球类：viewer
2. 地形类：Sm3dTerrainOperation Sm3dTerrainSlope Sm3dTerrainFlood Sm3dTerrainIsoline
3. 分析类：Sm3dMeasure Sm3dSkyline Sm3dProfile Sm3dSightline Sm3dShadowquery Sm3dViewshed Sm3dOpennessAnalysis Sm3dSpatialQuery3d
4. 裁剪类：Sm3dClipBoxByeditor Sm3dClipCross Sm3dClipPlane Sm3dClipPolygon

## 自定义组件

所谓自定义组件界面，即完全是用户自己写ui界面，然后使用组件逻辑的js部分。用户可以更加灵活自由的实现业务的需求。具体组件功能和接口请参考组件部分，下面介绍具体使用方式。

1. CDN使用方式：如实现地形开挖功能。

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="Content-Security-Policy" content="upgrade-insecure-requests">

<script src="https://unpkg.com/vue@next"></script>

<!-- cesium -->

<link href="https://www.supermapol.com/earth/vue-iEarth/examples/public/Cesium/Widgets/widgets.css"

rel="stylesheet">

<script src="https://www.supermapol.com/earth/vue-iEarth/examples/public/Cesium/Cesium.js"></script>

<!-- 组件包 -->

<script src="https://www.supermapol.com/earth/vue-iEarth/examples/dist/components\_basic.js"></script>

<title>自定义组件-CDN引入-demo</title></head>

 <style>

html,body, #app, #cesiumContainer {

width: 100%;

height: 100%;

padding: 0;

margin: 0;

}

.demo {

position: absolute;

top: 50px;

right: 100px;

background-color: rgb(253, 252, 252);

padding: 5px;

}

</style>

<body>

<div id="app">

<div id="cesiumContainer"></div>

<div class="demo">

<h3>自定义界面</h3>

<button @click="digTerrain">开挖</button>

<button @click="clearDig" style="margin-left: 8px;">清除</button>

</div>

</div>

<script>

const app = Vue.createApp({

setup(props) {

//初始化地球和加载测试地形场景

Vue.onMounted(() => {

viewer = new Cesium.Viewer("cesiumContainer");

viewer.scene.open('http://www.supermapol.com/realspace/services/3D-ZF\_normal/rest/realspace');

});

// 引入组件功能函数

let { digTerrain, clearDig } = webgl3d.terrainOperation(props)

return { digTerrain, clearDig }

}

});

app.mount("#app");

</script>

</body>

</html>

1. Npm使用方式，例如加载地球：

main.js里引入以下内容:

import webgl3d\_basic from "@supermap/iclient3d-vue-for-webgl/lib/index\_basic\_es.js";

在App.vue里：

<template>

<div id="cesiumContainer" ref="viewer">

</template>

<script>

import { onMounted } from "vue";

export default {

name: "Viewer",

setup(props) {

onMounted(() => {

webgl3d\_basis.initViewer(props);

});

}

};

</script>

<style >

html,body, #app, #cesiumContainer {

width: 100%;

height: 100%;

padding: 0;

margin: 0;

}

</style>

1. 拖拽设置

在main.js里使用以下内容：

import initDrag from "@supermap/iclient3d-vue-for-webgl/lib/initDrag.js";

initDrag(app)

使用指令：v-drag:启动拖动，v-stopdrag：停止拖动

1. 包含的所有接口（组件类的传入值props与返回值参考具体组件）
2. 工具类：

initDrag：拖拽指令

tool：封装的工具方法

layerManagement：图层管理方法，图层的增删等

Camera：相机的一些操作，飞行定位等

······

1. 组件类：

initViewer：初始化地球

terrainOperation：地形操作，包含地形开挖与地形修改

terrainSlope：地形坡度坡向

terrainIsoline：地形等值线

terrainFlood：地形淹没分析

clipBoxEditor：裁剪box交互

clipPolygon：裁剪多边形

clipPlane：裁剪平面

clipCross：裁剪Cross

clipBox:裁剪box

measure：量算

skyLine：天际线分析

shadowquery：阴影分析

viewshed：可视域分析

profile：剖面分析

sightLine：通视分析

openness：开敞度分析

spatialQuery3d：三维（GPU）查询分析

1. 工具类接口具体说明与使用：

······

注：Vite使用ui框架注意，工程内外文件名不能有中文，否则引入css会报错。

2

组件

通过本章节的学习，可以快速了解组件的全部内容。

* 介绍组件的使用举例和使用标签：

全局使用组件时，直接调用标签即可。

* 介绍组件的属性props值：

通过传递props可以更改组件的默认参数，也可以动态传入。

* 介绍组件函数的返回值：

当需要自定义组件时，可以调用组件的方法来返回需要的参数，实现数据的双向绑定。

* 在线范例地址：

<https://www.supermapol.com/earth/vue-iEarth/examples/index.html>

## 使用说明

使用组件的方式有很多种，下面对部分使用方式进行简要说明。

### 使用完整带界面组件

使用完整的组件方式最简单，可以参考第一章安装和快速使用的例子，引入后只需要使用标签即可。关于组件的切换或功能组合等，请自行参考vue项目学习。

改变默认参数方式（props）如viewer组件的sceneUrl属性，驼峰式写法。

<sm3d-viewer :scene-url="http://www.supermapol.com/realspace/services/3D-ZF\_normal/rest/realspace">

</sm3d-viewer>

### 使用自定义不带界面组件

自定义组件即用户自己编写UI界面后调用该功能的接口函数，这样可以满足不同组件的风格和功能需求。实现不同组件功能的函数封装，用户可以按需引入方法或参数。详细方法如下（以自定义天际线组件npm方式为例）：

<template>

<h1>自定义组件界面······</h1>

</template>

<script>

import skyLine from "./sky-line.js"; //引入组件功能实现接口函数，注意路径

export default {

name: "Sm3dSkyline", //组件名：标签调用

props: { //组件属性设置，可以设置默认值改变组件内部的默认参数，下面是部分举例

//天际线分析半径

skylineRadius: {

type: Number,

default: 12000 //设置默认值举例

},

//天际线宽度

lineWidth: {

type: Number

},

//显示高亮障碍物

highlightBarrier: {

type: Boolean,

default: false

},

},

setup(props) {

//按需引入需要的接口和参数，请参考具体组件解释说明

let {

getSkyline2d,

skyLineAnalysis,

setLimitBody,

clear,

echarts\_box,

skylineMode

} = skyLine (props);

return {

getSkyline2d,

skyLineAnalysis,

setLimitBody,

clear,

echarts\_box,

skylineMode

};

}

};

</script>

### 完整组件和自定义组件结合使用

符合需求的组件可以完整引用，如果需要个性化设计，则可以自定义组件。

### 组合完整带界面组件

例如iEarth产品，将各种类型的组件组合在一起形成一个大的组合式组件。这样可以在上层来设计和组合组件的界面，然后调用需要的组件。切换组件功能可以参考vue的is函数和keep-alive，可以更加方便的实现各种功能的组合使用。

注意：本产品组件使用的是基于vue3版本，低于vue3或者其他非vue项目，可以参阅GitHub组件功能实现的源码。

## 组件部分

### Viewer

标签: <sm-viewer></sm-viewer>

界面及介绍：无

Props:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| sceneUrl | 加载场景数据，由supermap的iserver发布提供的场景 | String | 参考SuperMap官网[webgl范例](http://support.supermap.com.cn:8090/webgl/examples/examples.html" \l "analysis" \t "http://support.supermap.com.cn:8090/webgl/examples/component/dist/zh/api/Viewer/_blank) | 无 |
| s3mScps | 加载s3m切片 | Array | 参考SuperMap官网[webgl范例](http://support.supermap.com.cn:8090/webgl/examples/examples.html" \l "layer-S3M" \t "http://support.supermap.com.cn:8090/webgl/examples/component/dist/zh/api/Viewer/_blank) | 无 |
| afterInitviewer | 初始化viewer后回调函数 | Function | / | / |
| openingAnimation | 开场动画 | Boolean | true/false | False |
|  |  |  |  |  |

Return 无

### 三维分析

* 量测

标签: <sm3d-measure></sm3d-measure>

功能：包括空间与贴地的距离，面积及高度的测量。

特点：半透明线绘制，多种贴地类型支持，实时等高线显示，精确度高。

第三方依赖：无。

Props：无。

measure返回值：

Return {

      measureMode,  //测量模式

      clampMode,  //贴地模式

      Ellipsoid,  //椭球选择

      isShowDVH,  //显示勾选界面

      isShowLine,  //显示等高线

      distanceClk,   //点击测距函数

      areaClk,   //点击测面

      heightClk,   //点击测高

      clear    //清除

    }

* 可视域

标签: <sm3d-viewshed ></sm3d-viewshed>

功能：进行可视域分析。

特点：支持可视域体和不可视体分析，支持动态可视域分析。

第三方依赖：axios。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| viewshedSpatialUrl | 可视域体数据服务 | String | - | 如下 |
| observerInformation | 初始化观察者信息 | Object | - | null |
| direction | 方向角 | Number | - | 1.0 |
| pitch | 俯仰角 | Number | - | 1.0 |
| addHeight | 附加高度 | Number | - | 1.8 |
| distance | 可视域距离 | Number | - | 200 |
| verticalFov | 水平视角 | Number | - | 60 |
| horizontalFov | 垂直视角 | Number | - | 90 |
| hintLineColor | 可视线颜色 | String | - | rgb(212,202,45) |
| visibleAreaColor | 可视区域颜色 | String | - | rgba(9,199,112,0.5) |
| hiddenAreaColor | 不可视域颜色 | String | - | rgba(238,114,22,0.5) |
| visibleBodyColor | 可视域体颜色 | String | - | rgba(9,199,112,0.7) |
| invisibleBodyColor | 不可视域体颜色 | String | - | rgba(238,114,22,0.7) |
| visibleBody | 显示可视域体 | Boolean | true/false | false |
| invisibleBody | 显示不可视域体 | Boolean | true/false | false |
| viewshedAnimation | 动态可视域 | Boolean | true/false | false |
| DynamicLine | 动态可视域路线点 | Array | - | [ ] |
| DynamicSpeed | 动态分析行进速度 | Number | - | 10 |

viewshedSpatialUrl默认值：<http://www.supermapol.com/realspace/services/spatialAnalysis-data_all/restjsr/spatialanalyst/geometry/3d/viewshedbody.json>

viewshed返回值：

return {

        ...toRefs(state), //包含上面的所有props

        analysis, //分析函数

        clear //清除函数

    };

* 剖面分析

标签：<sm3d-profile></sm3d-profile>

功能：对地形及模型进行剖面分析。

特点：支持折线，支持交互，精度高。

第三方依赖：echarts。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| profile2d | 默认显示剖面分析结果 | Boolean | true/false | false |
| polylineColor | 贴线颜色 | String | - | rgb(250, 213, 6) |
| polylineWidth | 贴线宽度 | Number | - | 5.0 |
| initEchartsOption | 初始化自定义echarts配置对象 | Object | - | - |
| updateEchartsOption | 自定义更新echarts配置对象 | Object | - | - |

profile返回值：

  return {

        ...toRefs(state),

        echarts\_box, //echarts节点对象，初始化echarts使用

        myChart, //导出echarts对象

        Entypositions, //entity对象位置

        LatAndLons,  //所有点的经纬度坐标

        Cartesians,  //所有点的笛卡尔坐标

        analysis, //分析函数

        clear //清除函数

    };

注：返回的节点对象要用ref

<div ref="echarts\_box" id="echarts\_box" ></div>

* 阴影分析

标签：<sm3d-shadowquery></sm3d-shadowquery>

功能：可以查看24小时、1年或其他任何时刻的日照情况，也可以分析指定区域的采光率。

特点：动画播放日照情况，可以分析具体区域时间段的采光率，以及采光率的体显示效果，过滤显示。

第三方依赖：无。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| timeValue | 开始结束时间 | Array | [0-96] | [24,64] |
| currentDate | 当前日期 | Object | - | new Date() |
| shadowShow | 显示阴影 | Boolean | true/false | true |
| timeInterval | 时间间隔 | Number | - | 60 |
| spacing | 间距（米） | Number | - | 10 |
| bottomHeight | 底部高程（米） | Number | - | 20 |
| extrudeHeight | 拉伸高度（米） | Number | - | 20 |
| shadowQueryRegion | 分析区域 | Array | - | [ ] |
| layerShadowType | 图层上阴影类型 | Number | - | Cesium.ShadowType.ALL |
| visibleAreaColor | 确定地形是否投射或接受来自太阳的阴影 | Number | - | Cesium.ShadowMode.RECEIVE\_ONLY |
| showStartImgForTime | 显示时间轴开始图标 | Boolean | - | true |
| showStartImgForDate | 显示日期开始图标 | Boolean | - | true |
| dockFontShow | 停靠图标显示 | Boolean | - | true |
| legendShow | 图例显示 | Boolean | - | false |
| shadowBodyShow | 阴影率体显示 | Boolean | - | false |
| shadowBodyFilter | 过滤区间 | Array | - | [0,100] |

shadowquery返回值：

 return {

        ...toRefs(state), //返回props

        timeChanged, //时间轴改变后设置分析起始时间函数（非实时）

        filterChanged, //过滤体显示函数

        formatTooltip, //格式化时间轴提示函数

        sunLightForTime, //播放或暂停时间段内阳光和阴影动画

        sunLightForDate, //播放一年的阳光和阴影动画

        analysis, //分析

        clear, //清除

        bubble, //气泡dom节点

        closeBubble, //关闭气泡函数

        dockBubble //悬停气泡函数

    };

* 天际线分析

标签: <sm3d-skyline></sm3d-skyline>

功能：进行天际线分析。

特点：支持天际线面和体分析，支持echarts二维展示，支持限高体绘制，可配合gpu空间查询天际线体。

第三方依赖：axios，echarts。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| spatialAnalysisUrl | 天际线体数据服务 | String | - | 如下 |
| observerInformation | 初始化观察者信息 | Object | - | null |
| skylineRadius | 天际线分析半径 | Number | - | 10000 |
| lineWidth | 天际线宽 | Number | - | 3 |
| skylineColor | 天际线颜色 | String | - | rgb(200, 0, 0) |
| skyBodyColor | 天际体颜色 | String | - | rgba(44,149,197,0.6) |
| highlightBarrierColor | 高亮障碍物颜色 | String | - | rgba(255, 186, 1, 1) |
| highlightBarrier | 显示高亮障碍物 | Boolean | - | true |
| skylineMode | 天际线分析模式线，面，体 | Number | 0，1，2 | 0 |
| getSkyline2d | 显示二维分析结果 | Boolean | - | true |

spatialAnalysisUrl默认值：<http://www.supermapol.com/realspace/services/spatialAnalysis-data_all/restjsr/spatialanalyst/geometry/3d/skylinesectorbody.json>

skyLine返回值：

return {

...toRefs(state), //返回所有props

skyLineAnalysis, //天际线分析

setLimitBody, //设置限高体

echarts\_box, //返回echarts节点元素，用ref接收

Clear //清除

};

* **通视分析**

标签: <sm3d-sightline></sm3d-sightline>

功能：进行实时可编辑的通视分析。

特点：通视分析是实时可编辑的，可一次进行多点分析，可返回障碍物高亮。

第三方依赖：无。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| viewPosition | 设置或获取视点位置 | Array | - | null |
| visibleColor | 可见线颜色 | String | - | rgb(0, 200, 0) |
| hiddenColor | 不可见线颜色 | String | - | rgb(200, 0, 0) |
| highlightBarrierColor | 高亮障碍物颜色 | String | - | rgba(255, 186, 1, 1) |
| highlightBarrier | 是否高亮障碍物 | Boolean | - | false |
| lineWidth | 分析线宽 | Number | - | 3 |

sightline返回值：

return {

        ...toRefs(state), //包含上面的所有props

        analysis, //分析函数

        clear //清除函数

    };

* 开敞度分析

标签: <sm3d-openness-analysis></sm3d-openness-analysis>

功能：进行开敞度分析。

特点：支持开敞度半径的设置和显示模式切换

第三方依赖：无。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| addHeight | 添加附加高度 | Number | - | 1 |
| viewPosition | 初始化分析位置 | Array | - | null |
| viewDomeRadius | 分析半径 | Number | - | 100 |
| domeType | 分析类型 | String | ALLDOME/  VISIBLEDOME/  HIDDENDOME | ALLDOME |
| isClosed | 是否封口 | Boolean | - | False |
| visibleAreaColor | 可是部分颜色 | String | - | rgba(9,199,112,0.5) |
| hiddenAreaColor | 不可视部分颜色 | String | - | rgba(238,114,22,0.5) |
| startAngle | 开始角度 | Number | - | 0 |
| endAngle | 终止角度 | Number | - | 360 |

openness返回值：

return {

        ...toRefs(state), //包含上面的所有props

        analysis, //分析函数

        clear //清除函数

    };

* 三维空间查询

标签: <sm3d-spatial-query3d></sm3d-spatial-query3d>

功能：进行可视域分析。

特点：支持可视域体和不可视体分析，支持动态可视域分析。

第三方依赖：axios。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| layerNames | 返回当前所有存在的图层 | Array | - | [ ] |
| selectedLayerName | 当前选择查询的图层 | String | - | ‘’ |
| Xpitch | x旋转 | Number | - | 0 |
| Yroll | y旋转 | Number | - | 0 |
| Zheading | z旋转 | Number | - | 0 |
| scale | 缩放 | Number | - | 3 |
| positionMode | 查询位置模式 | String | intersects/disjoint/contains | intersects |
| geometryType | 选择模型类型 | String | box/sphere/cone/cylinder/ellicpse | box |
| drawType | 模型显示类型 | String | Fill/WireFrame/Fill\_And\_WireFrame | Fill\_And\_WireFrame |
| FillColor | 模型填充颜色 | String | - | rgba(192,211,25,0.5) |
| WireFrameColor | 模型线框颜色 | String | - | rgba(89,129,228,0.8) |
| searchColor | 查询结果颜色 | String | - | rgba(255, 186, 1, 1) |
| GeometryBodyNames | 当前存在的体对象 | Array | - | [] |
| boxParameters | 模型Box参数设置 | Array | - | [100, 100, 100] |
| sphereParameters | 模型球体参数设置 | Array | - | [100] |
| coneParameters | 模型圆锥参数设置 | Array | - | [100, 200] |
| cylinderParameters | 模型圆柱参数设置 | Array | - | [100, 100, 200] |
| ellicpseParameters | 模型椭圆参数设置 | Array | - | [100, 50, 50] |
| rotateOrigin | 模型圆锥绕点旋转方式 | String | APEX/CENTER | APEX |

spatialQuery3d返回值：

return {

...toRefs(state),

setPosition, //设置或获取查询点位置

getQueryIDs, //获取查询结果

analysis, //分析

clear //清除

};

### 地形分析

* 地形操作

标签: <sm3d-terrain-operation></sm3d-terrain-operation>

功能：地形的挖掘与修改。

特点：支持编辑挖掘、修改区域。

第三方依赖：无。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| digDepth | 挖掘深度 | Number | - | 500 |
| digPositions | 初始化传入挖掘区域 | Array | - | [ ] |
| modifyPositions | 初始化传入修改区域 | Array | - | [ ] |
| isEdit | 是否编辑 | Boolean |  | false |
| isEditZ | 是否编辑Z轴 | Boolean | - | false |
| lineVisible | 是否显示绘制后的线 | Boolean | - | true |

terrainAnalysis返回值：

    return {

        ...toRefs(state),

        digTerrain, //地形开挖函数

        clearDig, //清除开挖

        modifyTerrain, //地形修改函数

        clearModify, //清除地形修改

        digPosition,  //导出开挖区域，便于用户需要保存当前开挖区域数据方案

        modifyPosition  //同上导出修改区域

    };

* 坡度坡向

标签: <sm3d-terrain-slope></sm3d-terrain-slope>

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| analysisArea | 分析区域 | String | ARM\_REGION/ARM\_ALL/ARM\_NONE | ARM\_REGION |
| displayMode | 显示模式 | String | FACE/ARROW/FACE\_AND\_ARROW | FACE |
| wideMaxr | 最大坡度 | Number | - | 90 |
| wideMinr | 最小坡度 | Number | - | 0 |
| slopeInterval | 最大最小坡度数组 | Array |  | [0,90] |
| trans | 透明度 | Number | - | 0.5 |
| slopePositions | 初始化传入分析区域 | Array | - | [ ] |
| lineVisible | 是否显示绘制后的线 | Boolean | - | true |

terrainSlope返回值：

return {

...toRefs(slopData),

startSlope, //分析函数

clearSlope, //清除

slopePosition //获取当前分析区域

};

* 淹没分析

标签: <sm3d-terrain-flood></sm3d-terrain-flood>

功能：地形淹没分析。

特点：支持地形淹没分析颜色选择，速度控制。

第三方依赖：无。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| maxHeight | 最大可见高层 | Number | - | 9000 |
| minHeight | 最小可见高程 | Number | - | 1000 |
| floodHeight | 最大最小可见高层数组使用方式 | Array |  | [1000,9000] |
| cheackedBand | 选择颜色 | String | - | band1 |
| floodSpeed | 淹没速度 | Number |  | 800 |
| floodTrans | 透明度 | Number | - | 0.8 |
| floodPositions | 初始化传入分析区域 | Array |  | [ ] |
| lineVisible | 是否显示绘制后的线 | Boolean | - | true |

terrainFlood返回值：

    return {

...toRefs(state),

floodBegin, //执行分析

floodClear, //清除

changeColor, //改变颜色函数

floodPosition //获取分析区域

};

* 地形等值线

标签: <sm3d-terrain-isoline></sm3d-terrain-isoline>

功能：地形等值线。

特点：支持等高线面等多种分析效果。

第三方依赖：无。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| fillMaxHeight | 最大可见高层 | Number | - | 9000 |
| fillMinHeight | 最小可见高程 | Number | - | 0 |
| fillHeight | 最大最小可见高程数组形式使用 | Array |  | [0,9000] |
| equivalentIsoline | 等值距 | Number | - | 100 |
| lineColor | 颜色 | String |  | Line |
| fillOptionsSelected | 显示模式 | String | - | #FF8040 |
| isEdit | 是否编辑 | Boolean |  | false |
| isolinePositions | 初始化传入分析区域 | Array |  | [ ] |
| lineVisible | 是否显示绘制后的线 | Boolean | - | true |

terrainIsoline返回值：

   return {

...toRefs(state),

isoLineAnalysis, //执行等值线分析

clearIsoLine, //清除

isolinePosition //获取分析当前区域

};

### 裁剪分析

* Box交互裁剪

标签: <sm3d-clip-box-byeditor></sm3d-clip-box-byeditor>

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| clipModel | 裁剪模式 | String | ClipInside/ClipOutside | ClipInside |

clipBoxByeditor返回值：

return {

...toRefs(state), //

BoxClipByEitor, //开始裁剪

clearBoxClipByEitor //清除裁剪

};

* cross裁剪

标签: <sm3d-clip-cross></sm3d-clip-cross>

功能：cross裁剪。

第三方依赖：无。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| clipWidth | 宽度 | Number | - | 5 |
| clipHeight | 高度 | Number | - | 5 |
| heading | 绕x旋转 | Number | - | 0 |
| pitch | 绕y旋转 | Number | - | 0 |
| roll | 绕z旋转 | Number | - | 0 |
| extrude | 拉伸 | Number | - | 1 |

terrainSlope返回值：

return {

...toRefs(state),

startCross, //开始裁剪

clearCross //清除裁剪

};

* 平面裁剪

标签: <sm3d-clip-plane></sm3d-clip-plane>

功能：绘制一个裁剪平面裁剪。

第三方依赖：无。

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| isEdit | 是否编辑 | Boolean | - | false |
| isEditZ | 是否编辑Z轴 | Boolean | - | false |
| PlanePositions | 初始化传入分析区域 | Array | - | [] |
| lineVisible | 是否显示绘制后的线 | Boolean | - | true |

clipPlaneAnalysis返回值：

return {

...toRefs(state),

clipPlaneStart, //开始裁剪

clearClipPlane, //清除

planePosition //获取裁剪当前区域

};

* 多边形裁剪

标签: <sm3d-clip-polygon></sm3d-clip-polygon>

Props：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 参数 | 说明 | 类型 | 可选值 | 默认值 |
| clipModelPolygon | 裁剪模式 | String | ClipInside/ClipOutside | ClipInside |
| isEdit | 是否编辑 | Boolean | - | false |
| isEditZ | 是否编辑Z轴 | Boolean | - | false |
| polygonPositions | 初始化传入分析区域 | Array | - | [] |
| lineVisible | 是否显示绘制后的线 | Boolean | - | true |

clipPolygonAnalysis返回值：

return {

...toRefs(state),

clipPolygon, //开始裁剪

clearClipPolygon, //清除

polygonPosition //获取裁剪当前区域

};