

鸿城InfraFuser开发教程

2022.04





1

产品简介

2

使用入门

3

基础知识点

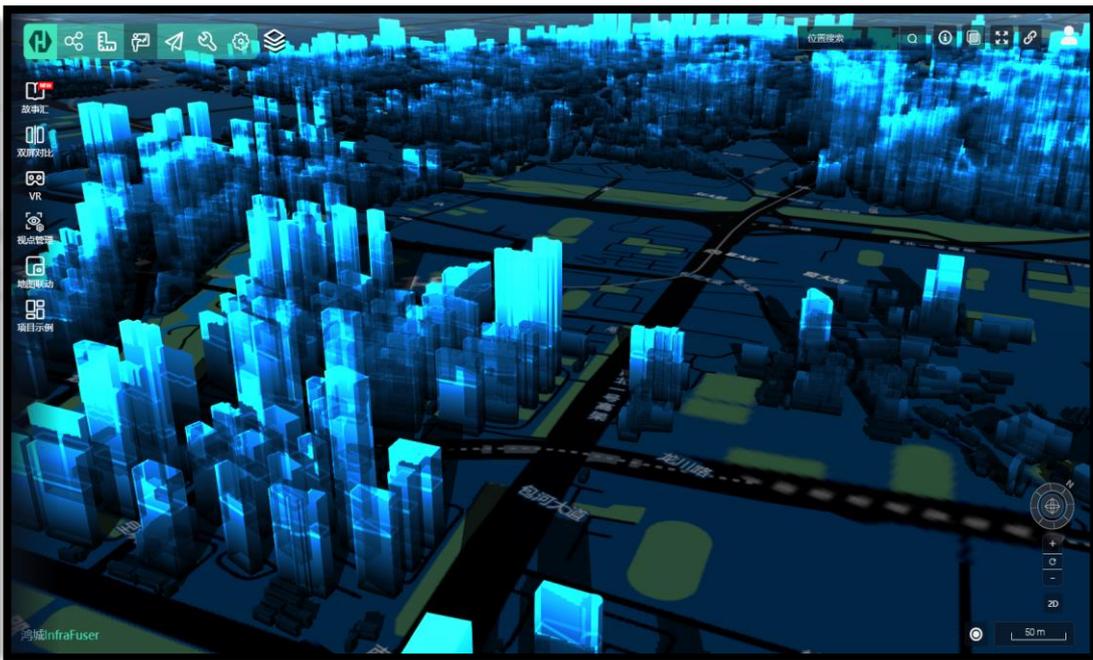
4

常见问题

1

产品简介

鸿城InfraFuser是一款基于 WebGL 技术实现的三维客户端开发平台，基于Cesium优化提升与B/S架构设计，支持多行业扩展的轻量级高效能GIS开发平台，能够无插件地在浏览器中高效运行，并可快速接入与使用多种GIS数据和三维模型，呈现三维空间的可视化，完成平台在不同行业的灵活应用。



■ 各大厂商的地图管理

支持高德、ArcGIS、天地图、Google 等厂商的地图接入，对于国内部分地图可进行偏移处理。

■ 各类数据的统一管理

提供覆盖物对应的图层管理操作，通过图层可对覆盖物进行针对性的筛选和整体的控制。

■ 各类数据动画和特效

提供线、面、圆、体等标绘的材质效果和细节动画，以更好的形式辅助展示三维场景。

■ 三维场景后期的特效

提供云、雨、雪、雾等天气功能和包括抗锯齿、高视觉质量等三维场景展示优化功能。

■ 路径漫游和视点漫游

提供根据设定的路线进行自动漫游功能，方便重点区域定位查看。

■ 空间分析能力

提供测距、面积、高度等各种量算功能以及可视域、通视、开挖分析等三维场景分析功能。

◆ Cesium

Cesium 是一款开源的基于JavaScript的3D地图框架。它是一款用于创建虚拟场景的3D地理信息平台。目标是用于创建以基于Web的地图动态数据可视化。目前尽力提升平台的性能、准确率、虚拟化能力、易用性以及平台的各种支持。

◆ 鸿城InfraFuser与Cesium的关系

Cesium库总体来说是很难用、API接口不友好的一个SDK，所以鸿城InfraFuser对Cesium的类的使用做了一致性的封装，统一的API接口风格、并对一些事件、调度等做统一内部处理，外部使用更加简单易用。

可以理解为鸿城InfraFuser对Cesium做了扩展和提升，并重新封装、并增加了很多新功能。

开发时如果需要 Cesium 的内部接口可以直接使用Cesium.*来调用即可。

◆ 运行环境

鸿城InfraFuser是依赖于WebGL运行的一套开发平台，需要开发或运行终端配置 独立显卡 和安装支持 WebGL 的浏览器，推荐使用 Chrome(谷歌)。

◆ 硬件配置

建议使用如下的硬件配置进行三维场景的浏览和开发

项目	标准配置	推荐配置
CPU	2.6GHz 64位 处理器	3.0GHz 64位 处理器
内存	8GB	32GB
显卡型号	NVIDIA GTX 1660 (或同级别)	NVIDIA RTX 2080 (或同级别)
显存	4GB	16G
最优分辨率	FHD - 1920 * 1080	4K QFHD - 3840 * 2160
操作系统	64 位 Windows 8/10/11 或 macOS 10.12.1以上版本	
浏览器	Google Chrome 或 Firefox 当前最新版本	

◆ 开发工具清单 (推荐)

分类	名称
IDE	VS Code
Web服务器	IIS
浏览器	Chrome

◆ VS Code

这里主要推荐下常用的IDE工具， Visual Studio Code简称VS Code，是一款免费开源的现代化轻量级代码编辑器，支持几乎所有主流的开发语言的语法高亮、智能代码补全、自定义热键、括号匹配、代码片段、代码对比 Diff、GIT 等特性。有大量的插件可选安装，使用方便易用，建议使用。

2

使用入门

◆ 获取WebGL及Cesium

开发时需要引入鸿城InfraFuser的WebGL.js及cesium相关文件。

这些文件可以通过鸿城二次开发示例官网获得。

◆ 文件引入

新建examples文件夹，将上方所述文件放入其中，在examples文件夹下新建HTML文件，在文件中通过<script> 标签将Cesium.js和WebGIS.js文件引入。

```
编辑器 重置 运行  
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4   <meta charset="utf-8">  
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6   <meta http-equiv="Access-Control-Allow-Origin" content="*" />  
7   <title>示例</title>  
8   <link rel="stylesheet" href="../Build/Cesium/Widgets/widgets.css">  
9   <script src="../Build/Cesium/Cesium.js"></script>  
10  <script src="../Build/WebGIS.js"></script>  
11 </head>  
12  
13 <body>  
14   <div id="cesiumContainer"></div>  
15
```

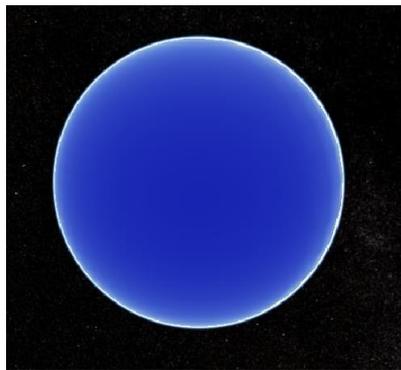
◆ 绘制地球

在绘三维地球前我们需要先准备一个定义了宽高的 DOM 容器。在刚才的例子 `</head>` 之后添加

```
14 <body>
15   <div id="cesiumContainer"></div>
16
```

然后就可以通过 `new WebGIS.Viewer3D`方法初始化一个实例并通过传入 `viewer3DConfig` 参数生成一个简单的三维地球，下面是完整代码

```
18 <body>
19   <div id="cesiumContainer"></div>
20   <script>
21     function onload() {
22       let self = this;
23       // 初始化地球
24       let viewer3D = new WebGIS.Viewer3D('cesiumContainer', viewer3DConfig);
25     }
26     //执行程序
27     onload();
28   </script>
29 </body>
```



◆ 加载项目

在绘三维地球后我们可以先给地球加载一个在线底图：

```
28     // 初始化地球
29     let viewer3D = new WebGIS.Viewer3D('cesiumContainer', viewer3DConfig);
30
31     this.baseMapSetting = new WebGIS.BaseMapSetting(viewer3D);
32     // 设置影像图层
33     this.baseMapSetting.loadArcGIS();
```

然后就可以通过图层管理接口 layerManagerService加载一个项目。

```
36     this.baseMapSetting.loadArcGIS();
37     // 获取图层管理的service
38     this.layerManagerService = new WebGIS.Service.LayerManagerService();
39
40     // 用getLayersByProjectInfo属性获取图层树json并加载图层
41     this.layerManagerService.getLayersByProjectInfo(viewer3DConfig.projectCode
42         .test1, undefined, true).then(function (layers) {
43         //layers --- 获取到的渲染数据
44         viewer3D.layerManager.loadLayers(layers); //调用layerManager
45         .loadLayers加载项目数据
46         // 首先查找第一个节点是否有lookAt点
47         let firstNode = viewer3D.layerManager.layerRoot.children[0];
48
49         // 如果firstnode存在且有Lookat属性 则飞到该默认视点
50         if (firstNode && firstNode.lookAt) {
51             viewer3D.layerManager.flyTo(firstNode);
52             return;
53         }
54     });
```

3

基础知识点

我们使用DIV渲染后看到的三维地球对象，我们统称为三维场景，在鸿城InfraFuser中对应是WebGIS.Viewer3D类，这是一切的开始，所有相关控制的起点，需要着重掌握。

◆ 三维场景初始化

在需要对地球进行相关设置时，你可以根据需要对默认参数进行配置，如果只是想得到默认的效果，仅需要写下面一行代码即可。

```
23 // 初始化地球
24 let viewer3D = new WebGIS.Viewer3D('cesiumContainer', viewer3DConfig);
```

当你需要对地图进行配置的时候，这里提供了详细的参数配置方案：

```
27 let viewerOpts = {};
28 // viewerOpts = {
29 //   requestRenderMode: true, // 增强模式
30 //   timeline: false,
31 //   animation: false,
32 //   fullscreenButton: false,
33 //   navigationHelpButton: false,
34 //   baseLayerPicker: false, // 图层控件
35 //   scene3DOnly: false,
36 //   geocoder: false, // 默认搜索框
37 //   homeButton: false,
38 //   selectionIndicator: false, // 选中元素显示,默认true(选中元素的绿框)
39 //   sceneModePicker: false, // 2d,3d模式切换
40 //   infoBox: false,
41 //   vrButton: false,
42 //   shouldAnimate: true,
43 //   shadows: false,
44 //   imageryProvider: false,
45 //   globeVectorEnable: true,
46 //   globeVectorHeight: 0
47 // }
48 // 初始化地球
49 let viewer3D = new WebGIS.Viewer3D('cesiumContainer', viewer3DConfig, viewerOpts);
50
```

Cesium支持多种服务来源的高精度影像地图数据的加载和渲染。鸿城InfraFuser对这块整体做了封装，可方便快捷的进行调用。

◆ 调用底图

WebGIS.BaseMapSetting管控各种底图的调用：

```
23 <script>
24   function onload() {
25     let self = this;
26
27
28     // 初始化地球
29     let viewer3D = new WebGIS.Viewer3D('cesiumContainer', viewer3DConfig);
30
31     this.baseMapSetting = new WebGIS.BaseMapSetting(viewer3D);
32     // 天地图
33     this.baseMapSetting.loadTianMap();
34     // 静态图片
35     this.baseMapSetting.loadLocalImage('../Build/hy_webgis/static/earth.jpg');
36     // ArcGIS
37     this.baseMapSetting.loadArcGIS();
38     // 高德影像
39     this.baseMapSetting.loadGaodeMapByType(6, false);
40     // 深蓝色影像
41     this.baseMapSetting.loadArcGISDeepBlue();
42
43   }
44
45   //执行程序
46   onload();
47
48 </script>
```

Camera相机控制了三维场景的视图。有很多方法可以操作Camera，鸿城InfraFuser封装在了一个视点工具ViewTool中。

◆ 调用视点工具

在WebGIS.Viewer3D.viewTool中进行视点的相关操作

方法名	说明
viewer3D.viewTool.setView (cameraInfo)	直接设置视点，没有过渡
viewer3D.viewTool.flyTo (cameraInfo, opts)	直接设置视点，有过渡
viewer3D.viewTool.getCurrentView (shoot, name, w, h, q)	获取当前视角信息
viewer3D.viewTool.flyToExtentCenter (extent, opts)	定位到四至范围的终点
viewer3D.viewTool.lookAt (lookAt, isFly)	设置视点，相机看的目标点

鸿城InfraFuser的图层管理均在LayerManager下，可以进行图层的相关交互操作。

◆ 调用图层管理工具

在WebGIS. Viewer3D.layerManager中进行图层的相关操作

方法名	说明
viewer3D.layerManager.loadLayers (layers)	卸载之前的图层树，加载新的图层树
viewer3D.layerManager.addLayer (node)	加载图层
viewer3D.layerManager.removeLayer (node)	卸载图层
viewer3D.layerManager.clean ()	清理当前图层
viewer3D.layerManager.flyTo (node)	定位当前图层

Tileset中Feature管理类, FeatureManager。

◆ 调用feature管理工具

在WebGIS. Viewer3D.featureManager中进行feature的相关操作

方法名	说明
viewer3D.featureManager.highlight (id, color)	高亮Feature
viewer3D.featureManager.setHighlightColor (color)	设置默认高亮颜色
viewer3D.featureManager.cancelHighlight ()	取消高亮
viewer3D.featureManager.onlyShow (ids)	只显示指定Feature, 其余Feature隐藏
viewer3D.featureManager.twinkled (id)	闪烁指定Feature

平台支持一些场景特效，包括 雾天气，雨天气，雪天气等。

```
57  
58 // 天气管理  
59 this.weatherManager = new WebGIS.WeatherManager(viewer3D.viewer);  
60
```

◆ 调用场景特效工具

在WebGIS.WeatherManager中进行场景特效的相关操作

方法名	说明
viewer3D.weatherManager.postProcess (type)	场景特效（雨、雪、雾）
viewer3D.weatherManager.close ()	关闭场景特效

涉及到地图开发，需要进行频繁的坐标系转换，鸿城InfraFuser封装了较完善的计算工具接口ComputeTool。

◆ 调用计算工具

在WebGIS.Viewer3D.computeTool中进行坐标系转换的相关操作

方法名	说明
viewer3D.computeTool.computeDistance (postions)	计算距离
viewer3D.computeTool.getCurrentCameraHeight ()	获取当前镜头的高度
viewer3D.computeTool.getPolygonCenter (entity)	获取多边形中心点
viewer3D.computeTool.computeAngle (positions)	计算角度
viewer3D.computeTool.cartesianToCartographic (catesian3)	地心转地理坐标

4

常见问题

■ 如何高亮构件

高亮feature可以通过viewer3D.featureManager.Highlight (id, color) 来进行高亮。

■ 如何给不同的模型显示不同的颜色

可以通过Highlight中传递color参数来控制显示不同的颜色。

■ 浏览器不支持WebGL

请选择支持WebGL的浏览器进行二次开发，比如：Chrome、火狐等浏览器。