



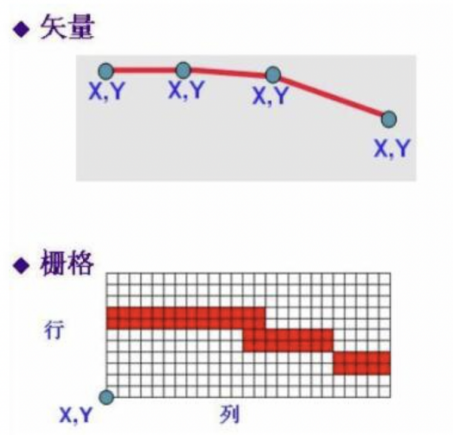
WebGIS

WebGIS介绍

- 什么是WebGIS
 - GIS全称网络地理信息系统，WebGIS是Web技术和GIS技术相结合的产物，是利用Web技术来扩展和完善地理信息系统的一种技术。
 - 将GIS的对空间数据的采集、存储、管理、分析与应用的功能在Web端实现。
- WebGIS能做什么
 - 实现空间数据的检索、查询、制图输出、编辑等。
 - 通过互联网对地理空间数据进行发布和应用，以实现空间数据的共享和操作。
 - 前端结合后端进行数据可视化交互。
- 应用方向
 - 标绘。如绘制点、线、面、扇形、圆弧、圆、多边形、平尾箭头等。
 - 业务数据可视化。将已知数据、实时数据展示在地图上。
 - 制作专题图。如热力图、散点图、蜂窝图、迁徙图等。
 - 空间分析。如测量、通视分析、淹没分析、视域分析、缓冲区分析、填挖方分析等。
 - 数字城市、智慧校园等。
- 相关技术
 - JavaScript、Vue、React、Webpack、Vite、脚手架
 - GIS前端开源库，如CesiumJS、Open layers等
 - Java、GeoServer、Linux、图形学
 - 地理坐标转换、空间测量、批量渲染、视角方位等

空间数据

- 矢量数据
 - 基于要素的，由点、线、面组成的复合对象，可携带更多空间信息
- 栅格数据
 - 由像素组成，并按照一定的方式排列



比较	矢量数据	栅格数据
优点	数据精确度高、数据量小、容易操作、不失真、有利于网络和检索分析	结构简单、数据模拟方便、易开发
缺点	结构复杂、显示及绘制要求高、技术复杂	数据量大、输出不美观、投影转换比较复杂
格式	SHP、GeoJSON、GML、KML、OSM	GeoTIFF、瓦片、png/jpg

• SHP文件组成

- **shp** 存储空间信息
- **shx** 形状位置索引
- **dbf** 属性数据存储
- **prj** 坐标数据
- **xml** 关联元数据存储
- **sbn** 索引文件，与sbx一起保存
- **sbx** 索引文件，与sbn一起保存
- **cpg** 指定编码格式

坐标系

- WGS84：大地坐标系，目前广泛使用的GPS全球卫星定位系统
- GCJ02：火星坐标系，由中国国家测绘局制订的地理信息系统的坐标系统，由WGS84坐标系经加密后的坐标系
- BD09：百度坐标系，在GCJ02坐标系基础上再次加密。其中bd09ll表示百度经纬度坐标，bd09mc表示百度墨卡托米制坐标
- EPSG:4326 → WGS84(大地)
- EPSG:3857 → 墨卡托投影

百度地图API

- 轻量、使用简单、可做简单的交互
- 离线地图项目集成性差，适合静态页面使用

- 对外接口的坐标系为BD09坐标系；非中国地区地图，服务坐标统一使用WGS84坐标
- 在线地图集成简单，并有对应的React库，可在项目直接使用
 - [百度地图JavaScript API](#)
 - [React-BMapGL](#)
- 可实现功能
 - 点、线、面、标注、定位
 - 自定义控件、轨迹绘制
 - 鼠标单击、右键事件、搜索

前端GIS开源库

Web GIS	Openlayers	Cesium	Mapbox	ArcGIS API	Leaflet
是否开源	开源	开源	开源	商用	开源
是否支持2D	支持	支持	支持	支持	支持
是否支持3D	不支持	支持	支持	支持	支持
功能多样性	一般	全面	一般	全面	一般
坐标系支持	均支持	均支持	墨卡托投影	均支持	均支持
兼容性	IE、现代浏览器	现代浏览器	现代浏览器	现代浏览器	现代浏览器、跨平台、移动设备
框架侧重点	IE浏览器的兼容	三维建模与时态模拟	视觉的呈现	可视化	轻量、传统GIS
其它		WebGL、硬件加速			

Cesium集成方案(GIS集成)

- 通用方案
 - 静态源码引入。将Cesium源码引入到项目公共目录，页面引入即可。
- Vue 2.x

```
// 1. node包安装，将Cesium包作为项目依赖安装，并在配置文件进行相关配置
npm i cesium
// 使用
import Cesium, { Viewer } from 'cesium'
// do something

// 2. vue-cesium插件。可通过vue-cesium集成cesium环境
npm i vue-cesium
// main.js
import Vue from 'vue'
import VueCesium from 'vue-cesium'
import lang from 'vue-cesium/lang/zh-hans'
// import lang from 'vue-cesium/lang/en-us'
// 注册
Vue.use(VueCesium, {
  cesiumPath: '', // cesium库路径,本地和cdn均可,
  lang: lang, // 显示语言
})
```

- Vue 3.x

```
// 1. 通过vue-cesium插件,与Vue2.x相似
npm i vue-cesium

// 2. 基于vite打包工具
npm i vite-plugin-cesium
// vite.config.js
import { defineConfig } from 'vite';
import cesium from 'vite-plugin-cesium';
export default defineConfig({
  plugins: [cesium()]
});
```

- React

```
// 1. 可采用通用方案,全局引入
// 2. 将Cesium包作为项目依赖安装,并通过Webpack进行相应配置
```

CesiumJS

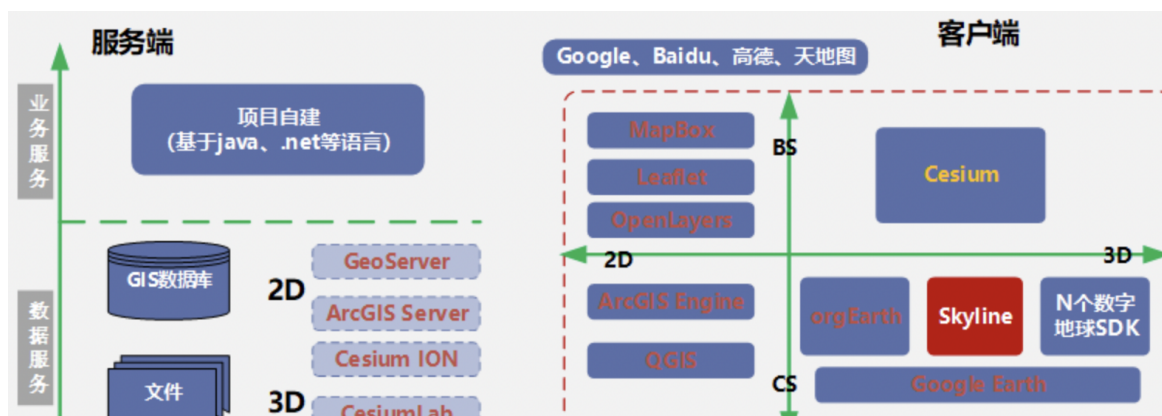
- 简介

- CesiumJS是一个开源 JavaScript 库，用于展示具有最佳性能、精度、视觉质量和易用性的世界级 3D 地球仪和地图。
- CesiumJS主要用于创建交互式 Web 应用程序来共享动态地理空间数据，在没有插件的 Web 浏览器中创建 3D 地球和 2D 地图。
- CesiumJS渲染上使用 WebGL 进行硬件加速图形，且是跨平台、跨浏览器的，可针对动态数据可视化进行了调整。

- 环境要求

- Cesium的运行环境需浏览器支持WebGL、Canvas等Web API。
- 推荐使用Google Chrome、Microsoft Edge、Apple Safari等浏览器。

- 项目基本架构



- CesiumJS API示例

◦ 基本搭建

```
// 创建基本视图
import Cesium from 'cesium';
const viewer = new Cesium.Viewer(container, options);
```

◦ 加载图层

- Cesium针对不同的数据格式提供不同的加载接口
- `Cesium.UrlTemplateImageryProvider` 加载URL模板（常用）
- `WebMapServiceImageryProvider` 加载 `wms`、`tiff` 格式数据
- 具体API根据实际情况调用

◦ 实体操作

- 实体是Cesium绘制图形的基本单位
- 如地图上点、扇形、栅格其实都是一个实体
- 实体有众多的实体属性，可根据实际情况进行配置

```
import Cesium from 'cesium';
const viewer = new Cesium.Viewer(container, options)

// 增
viewer.entities.add({
  id: 'uniqueId'
});

// 删
viewer.entities.removeById('uniqueId')
viewer.entities.removeAll()

// 查
const entity = viewer.entities.getById('uniqueId');

// 改 先查到某个实体，修改实体的属性以达到修改效果
const entity = viewer.entities.getById('uniqueId')
entity.polygon.material = Cesium.Color.fromCssColorString('#fff')
```

◦ 相机系统

- Cesium通过相机来控制场景中的视域
- 旋转、缩放、平移等都是通过操作相机移动来实现的
- 相机位置通过确定 `(经度, 纬度, 大地高度) -> (x, y, z)` 来确定

```
// setView通过定义相机飞行目的点的三维坐标（经纬度和大地高）和视线方向，将视角直接切换到所设定的视域范围内
// setView没有空中飞行的过程，适合快速切换视角
// flyTo有相机空中飞行的过程。可以设置飞行时间。

const position = Cesium.Cartesian3.fromDegrees(86.889, 27.991, 4000);
viewer.camera.flyTo({
  destination: position,
  orientation: {
    heading: Cesium.Math.toRadians(0.0),
    pitch: Cesium.Math.toDegrees(0.0),
```

```

        roll: 0.0
    },
    duration: 5, // 相机飞行持续的时间，默认值会根据距离进行计算
});

```

- 地理坐标
 - Cesium默认使用 **WGS84** 即谷歌地图坐标系
 - 若使用如百度等地图，需利用对应地图Cesium集成API进行转化
- 支持数据格式及json格式
 - WMS、TMS、WMT、ArcGIS、Tiff、离线瓦片、在线地图（天地图、谷歌、百度等）
 - GeoJson
- 常用类
 - Viewer
 - Camera
 - Entity
 - UrlTemplateImageryProvider
 - WebMapServiceImageryProvider
 - Color
 - Billboard
 - Label
 - ScreenSpaceEventHandler
 - ScreenSpaceEventType

实现功能

- 地图配置(控件、初始化)
 - 在初始化创建地图时可进行控件的配置

```

const viewer = new Cesium.Viewer(container, {
    // 以下均为可配置项
    baseLayerPicker: false, // 初始图层
    timeline: false, // 时间线控件
    animation: false, // 动画轮盘控件
    fullscreenButton: false, // 是否全屏按钮
    homeButton: false, // 主页按钮
    navigationHelpButton: false, // 导航栏帮助按钮
    geocoder: false,
    sceneModePicker: false, // 场景模式选择器
    infoBox: false, // 信息框
    selectionIndicator: false, // 指示框
    sceneMode: Cesium.SceneMode.SCENE2D, // 开启2D场景模式
    requestRenderMode: true,
    // 图层提供者
    imageryProvider: new Cesium.BaiduImageryProvider({
        url: 'tiles url',
        minimumLevel: 5,
        maximumLevel: 18,
    })
});

```

```

        style: 'normal',
        crs: 'WGS84'
    })
})

```

- 放大缩小按钮控件
 - Cesium本身没有提供放大缩小按钮的控件
 - 可通过 `cesium-navigation-es6` 库引入该功能
 - 可作为依赖集成，也可下载源码打包静态引入
 - 自定义实现放大缩小功能

- 绘制点

```

// 根据一组坐标绘制
const entity = {
  position: Cesium.Cartesian3.fromDegrees(...pos),
  point: {
    // some attributes for a point
  }
}

```

- 绘制线、面

```

// 线、面，主要通过实体中 polygon(多边形) 属性指定
const positions = Cesium.Cartesian3.fromDegreesArray(data),
const entity = {
  polygon: {
    hierarchy: {
      positions: positions
    },
    // some attributes for a polygon
  }
}

```

- 绘制扇形

```

// 扇形也属于一种特殊的多边形，也可通过polygon属性指定
// 需引入 turf.js 进行转换
const point = turf.point(center)
const data = turf.sector(point, radius, startAngle, endAngle, {steps: 512}).geometry.coordinates[0];
const entity = {
  polygon: {
    hierarchy: {
      positions: data
    },
    // some attributes for a polygon
  }
}

```

- 事件注册/移除（单击、双击）
 - Cesium内置单击、双击事件，使用前需进行注册

```
// 注册事件
const handlerLeftClick = new Cesium.ScreenSpaceEventHandler(viewer.scene.canvas)
const handlerLeftDoubleClick = new Cesium.ScreenSpaceEventHandler(viewer.scene.canvas)

handlerLeftClick.setInputAction(e => {
    // do something
}, Cesium.ScreenSpaceEventType.LEFT_CLICK)

// 移除事件
handlerLeftClick.removeInputAction(Cesium.ScreenSpaceEventType.LEFT_CLICK)
```

- 监听地图加载

```
const helper = new Cesium.EventHelper()
helper.add(viewer.scene.globe.tileLoadProgressEvent, e => {
    // do something
})
```

- 地图切换

- 基于不同区域的地图进行切换
- 在线与离线(手动导入数据)进行切换

- 绘制标记

- 定位标记
- Label文本
- 广告牌

- 实体操作

- 各种渲染及查找
- 条件清除、全部清除
- 高亮、恢复颜色

- 检索定位

- 目前检索定位主要在现有的实体上进行搜索
- 搜索基站后跳转到相应位置并进行标注
- 道路数据基于地图服务，由后端汇总
- 支持基站、道路模糊搜索
- 支持跳转过渡效果

- 地图管理设计

- 用于对地图进行集中管理，目前支持新增、编辑、删除、是否启用、导入离线地图等功能
- 地图使用方式可选在线或离线，目前离线方式已验证
- 地图类型可选谷歌、百度地图等类型
- 在线地图目前基于GeoServer实现

- 实时显示信息
 - 实时显示小区名称
 - 实时显示栅格详情信息
 - 显示容器可基于组件
- 地图服务
 - 基于GeoServer，目前可加载WMS格式矢量数据
 - 发布矢量数据单图层、多图层
 - 地图自定义样式
 - 可基于Cesium或手动查询接口获取矢量要素

GeoServer

- 介绍
 - GeoServer 是 OpenGIS Web 服务器规范的 J2EE 实现，利用 GeoServer 可以方便的发布地图数据，允许用户对特征数据进行更新、删除、插入操作。
 - 通过 GeoServer 可以比较容易的在用户之间迅速共享空间地理信息，GeoServer是一个开源软件
 - 支持矢量数据和栅格数据，支持各种投影，可自定义样式
 - 支持OGC规范，如WMS、WFS、WMTS等
 - 支持PostGIS、Shapefile、ArcSDE、Oracle、VPF、MySQL、MapInfo
 - 能够将网络地图输出为JPEG、GIF、PNG、SVG、KML等格式
 - 提供多种服务及API接口
- 环境要求
 - Java8或Java11
- 功能
 - Restful接口: 客户端可以通过该接口检索有关实例的信息并进行配置更改
 - 服务：WMS(图层数据)、WFS(要素查询)
 - 样式、筛选

参考文档

- <https://www.cesium.com/>
- <https://geoserver.org/>