

地理信息系统实验 · ②：

基于高德地图 JS API 的 在线地图设计与搭建



学院：智能工程学院

专业：17 级交通工程

姓名：吴 敏 毅

学号：17313030

指导老师：胡 继 华

2019.10.7

目录

一、实验目的:	1
二、知识储备	1
2.1 网页结构知识储备	1
2.2 HTML、CSS 、JavaScript 的概念	2
2.2.1 HTML	2
2.2.2 CSS	2
2.2.3 Javascript	2
2.2.4 三者之间的关系	2
2.3 Access	2
2.3.1 软件介绍	2
2.3.2 软件安装	3
2.4 Visual Studio	3
2.4.1 软件介绍	3
2.3.2 软件安装	4
2.5 高德开发平台	4
2.5.1 高德开发平台介绍	4
2.5.2 地图开发工具: API/SDK	4
2.5.3 地图 JS API 概述	5
三、实验准备	5
3.1 注册账号并申请 Key	5
3.2 创建项目环境	8
3.3 学习高德 JS API 的使用	10
四、功能开发	16
4.1 界面设计	16
4.2 显示广州地图并添加放缩工具条	18
4.3 实现 POI 查询功能	19
4.3 实现特色查询功能	23
五、实验总结	31
参考文献	32
附录	33

一、实验目的：

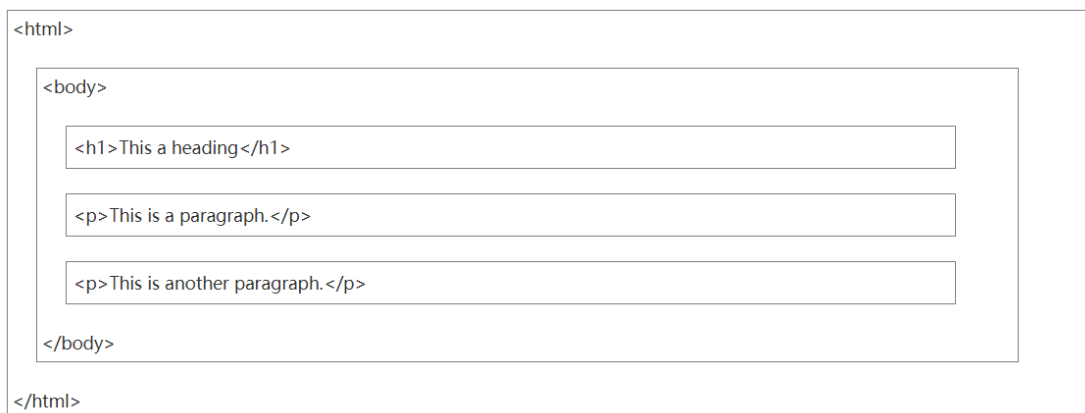
1. 了解网页结构和基本组成，并了解现今主流网页脚本语言；
2. 利用高德开放平台的学习资源，学习在线地图的开发；
3. 学习数据库的基本知识，并掌握 access 的使用；
4. 掌握 JavaScript 语言，并用其进行在线地图的设计和成品开发，要求该在线地图所含的功能有：
 - a) 显示广州市地图；
 - b) 实现 POI 查询功能，输入地名，可并将其查询结果显示在 html 界面，并后台将查询数据存储事先建好的数据库中；
 - c) 进一步开发出周边查询、沿线搜索、多边形搜索、矩形搜索、圆形搜索等高级示范功能，保存搜索到的数据。
5. 合理布局 and 美化 html 界面。

二、知识储备

2.1 网页结构知识储备

html 页面是静态的，从头到尾没有程序的执行，是纯的 html 语言，是不经过服务器处理就直接送出给浏览器呈现给浏览者，而其他如 php、asp、aspx、jsp 等后缀结尾的网页相对来说就是动态网页页面。动态页面是经过服务器对各自的程序翻译处理、数据库操作等处理后才由浏览器把服务器处理完的数据程序给用户，而网页内容数据可以随后台数据改变而改变。

下面是一个可视化的 HTML 页面结构（只有<body>标签中的内容会显示在浏览器中）：



更多 HTML 网页结构了解，请参考：

HTML 简介-W3Cschool: <https://www.w3cschool.cn/html/html-intro.html>

2.2 HTML、CSS 、JAVAscript 的概念

2.2.1 HTML

- HTML 是用来描述网页的一种语言。
- HTML 指的是超文本标记语言 (Hyper Text Markup Language)
- HTML 不是一种编程语言, 而是一种**标记语言** (markup language)
- 标记语言是一套标记标签 (markup tag)
- HTML **使用标记标签来描述网页**

2.2.2 CSS

层叠样式表(英文全称: Cascading Style Sheets)是一种用来表现 HTML (标准通用标记语言的一个应用) 或 XML (标准通用标记语言的一个子集) 等**文件样式的计算机语言**。CSS 不仅可以**静态地修饰网页**, 还可以**配合各种脚本语言动态地对网页各元素进行格式化**。

2.2.3 Javascript

JavaScript 是一种属于**网络的脚本语言**, 已经被广泛用于 Web 应用开发, 常用来**为网页添加各式各样的动态功能**, 为用户提供更**流畅美观的浏览效果**。通常 JavaScript 脚本是通过嵌入在 HTML 中来实现自身的功能的。

2.2.4 三者之间的关系

对于一个网页, **HTML 定义网页的结构**, **CSS 描述网页的样子**, **JavaScript 则实现人机交互**。如果说 HTML 是肉身、CSS 就是皮相、Javascript 就是灵魂。没有 Javascript, HTML+CSS 是植物人, 没有 Javascript、CSS 是个毁容的植物人。

2.3 Access

2.3.1 软件介绍

Microsoft Office Access (前名 Microsoft Access) 是由微软发布的**关系数据库管理系统**。它结合了 Microsoft Jet Database Engine 和图形用户界面两项特点, 是 Microsoft Office 的系统程序之一。Microsoft Access 在很多地方得到广泛使用, 例如小型企业, 大公司的部门。

Access 能够访问 Access/Jet、Microsoft SQL Server、Oracle 数据库, 或者任何 ODBC 兼容数据库内的数据。熟练的软件设计师和数据分析师利用它来**开发应用软件**, 而一些不熟

练的程序员和非程序员的高级用户则能使用它来开发简单的应用软件。喜爱编程的开发人员亦利用它来制作处理数据的桌面系统。它也常被用来开发简单的 WEB 应用程序。[1]

2.3.2 软件安装

可在中山大学微软企业服务门户中下载 Office2013 中文版套件光盘映像，然后将*.iso 文件中将安装包导出，即可安装安装教程进行 access 的安装操作。

Office 中文版套件光盘映像

名称	广州校区	珠海校区
Office 2016 SP1 32位 (SHA1: 0218F50774AAB63AF7755B0986CDB9972B853E44)	下载	下载
Office 2016 SP1 64位 (SHA1: AEB58DE1BC97685F8BC6BF0A614A8EF6903E318)	下载	下载
Office 2013 SP1 32位 (SHA1: 8E24A3A50003A633921F45AC02496F8C90122AEC)	下载	下载
Office 2013 SP1 64位 (SHA1: 70C450F78A3D48E7BF1086052C6F46981A205052)	下载	下载

文件下载：

<https://ms.sysu.edu.cn/office#Office%20%E4%B8%93%E4%B8%9A%E5%A2%9E%E5%BC%BA%E7%89%88%202016%E7%AD%89%E7%89%88%E6%9C%AC%E5%85%89%E7%9B%98%E6%98%A0%E5%83%8F%E4%B8%8B%E8%BD>

安装教程：<https://www.jb51.net/access/450734.html>

2.4 Visual Studio

2.4.1 软件介绍

Microsoft Visual Studio (简称 VS 或 MSVS) 是微软公司的开发工具包系列产品。VS 是一个基本完整的开发工具集，它包括了整个软件生命周期所需要的大部分工具，如 UML 工具、代码管控工具、集成开发环境 (IDE) 等等。所写的目标代码适用于微软支持的所有平台，包括 Microsoft Windows、Windows Phone、Windows CE、.NET Framework、.NET Compact Framework 和 Microsoft Silverlight。

而 Visual Studio .NET 是用于快速生成企业级 ASP.NET Web 应用程序和高性能桌面应用程序的工具。Visual Studio 包含基于组件的开发工具 (如 Visual C#、Visual J#、Visual Basic 和 Visual C++)，以及许多用于简化基于小组的解决方案的设计、开发和部署的其他技术。

本次实验便基于 VS2019 进行在线地图的开发。

2.3.2 软件安装

VS2019 Community 版本完全免费，可到官网进行下载安装，安装步骤也极其简单，正常安装即可。

下载地址：

<https://visualstudio.microsoft.com/zh-hans/downloads/?rr=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fvisualstudio%2Fide%2Fwhats-new-visual-studio-2019%3Fview%3Dvs-2019>

2.5 高德开发平台

2.5.1 高德开发平台介绍



高德开发平台的成果

高德开放平台是国内技术领先的 **LBS 服务提供商**，拥有先进的数据融合技术和海量的数据处理能力。高德开放平台向广大开发者**提供覆盖移动端和 Web 端的开发工具**，开发者通过**调用开发包或接口即可在应用或网页中实现地图显示、标注、位置检索等功能**。使得 LBS 应用的开发过程更加容易。

高德开放平台为开发者提供了三项主要的能力：

- 1、专业、易用的地图开发工具：API/SDK
- 2、快捷的位置云计算：云图
- 3、权威的位置大数据：高德位智

2.5.2 地图开发工具：API/SDK



高德地图的地图开发工具

高德地图 API/SDK 是一套为开发者提供的地图应用程序接口，包括 JavaScript、iOS、Android、Windows、静态地图、Web 服务等多种版本。提供了定位、地图、导航（公交\驾车\步行）、位置搜索、周边检索、地理编码及逆地理编码、实时路况等丰富功能。

因为高德已经将包括：定位、地图、导航等的功能以 API/SDK 的形式开放给开发者，通过调用高德地图的 API 及 SDK，APP 就能轻松拥有和高德一样专业的地图功能！

本次的实验成果便是基于高德地图 JS API 进行开发的。

2.5.3 地图 JS API 概述

高德地图 JS API 是一套 JavaScript 语言开发的的地图应用编程接口，移动端、PC 端一体化设计，一套 API 兼容众多系统平台。目前 JS API 免费开放使用。

JS API 提供了 2D、3D 地图模式，满足绝大多数开发者对地图展示、地图自定义、图层加载、点标记添加、矢量图形绘制的需求，同时也提供了 POI 搜索、路线规划、地理编码、行政区查询、定位等众多开放服务接口。

同时高德开发平台还开放开发教程供开发者学习，便于快速掌握和应用。



高德开发平台-示例中心

三、实验准备

3.1 注册账号并申请 Key

1. 首先，注册开发者账号，成为高德开放平台开发者
2. 登陆之后，在进入「应用管理」 页面「创建新应用」
3. 为应用添加 Key，「服务平台」一项请选择「 Web 端（ JSAPI ）」

① 打开高德 API 开发平台，点击页面右上角“注册”按钮；

网址：<https://lbs.amap.com/>



② 填写信息进行开发者注册；



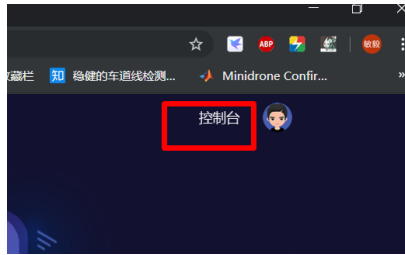
③ 注册成功；



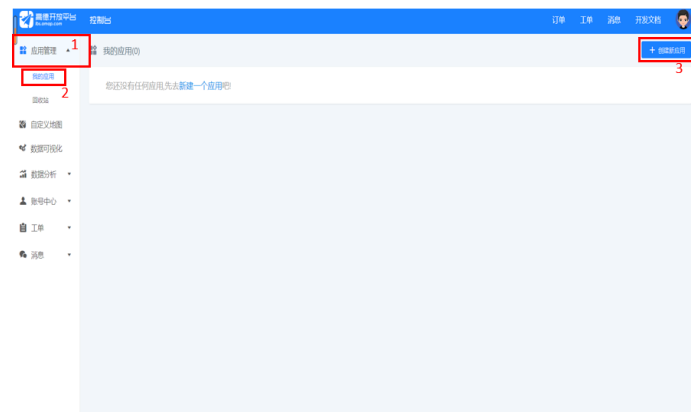
④ 登录开发者账号；



⑤ 登陆成功后，打开控制台；



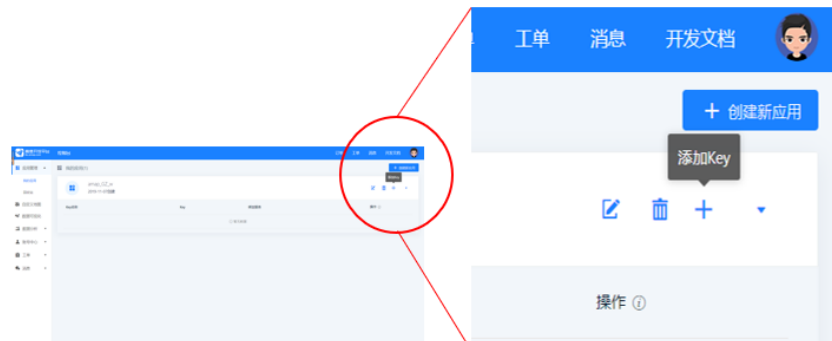
⑥ 点击应用管理，再点击我的应用，最后点击“创建新应用”；



⑦ 输入应用名称和应用类型；



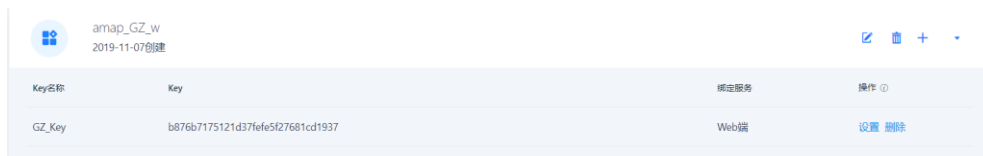
⑧ 点击添加新 key；



⑨ 输入 key 的名称和设定属性；

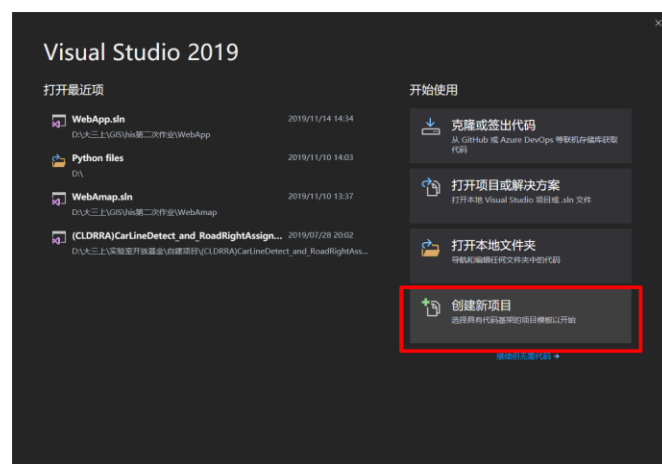


⑩ 输出完成点击提交，即可生成 key。



3.2 创建项目环境

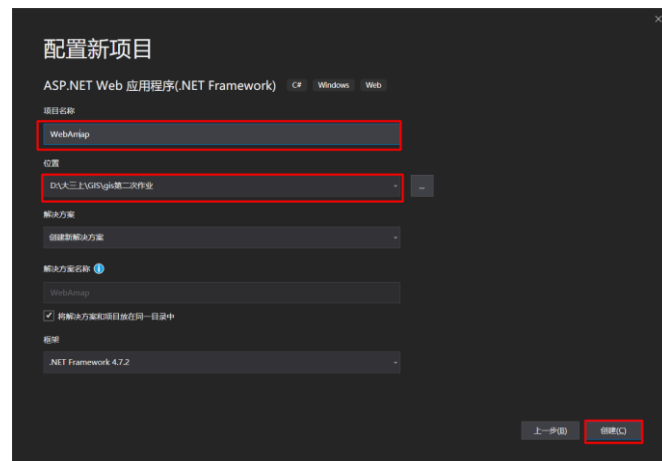
① 打开 VS2019，创建新项目；



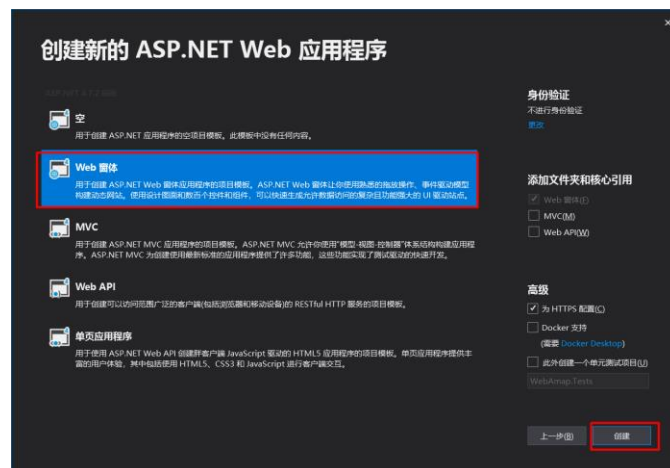
② 选择 ASP.NET Web 应用程序 (.NET Framework)；



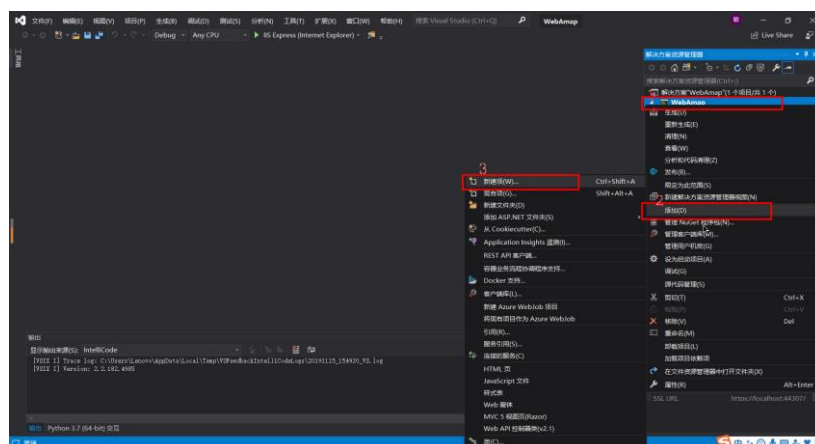
③ 输入项目名称，修改项目存放位置，最后点击创建：



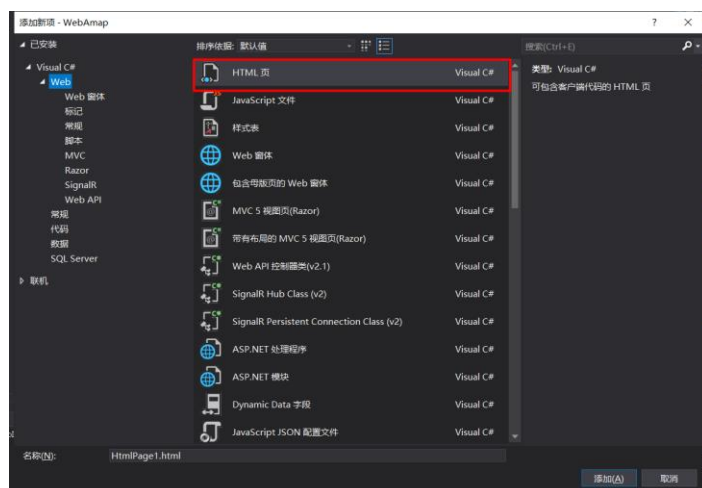
④ 新的页面选择 “Web 窗体” 后，点击创建：



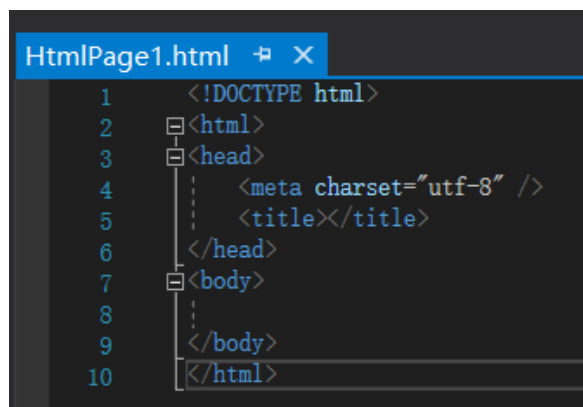
⑤ 锁定“解决方案资源管理器”->右键“Web_Online_Amap” (项目名)->点击“添加项”->点击“加载项”；



⑥ 在弹出页面找到 “HTML 页”，单击选中进行添加；



⑦ 编辑器中出现初始代码：



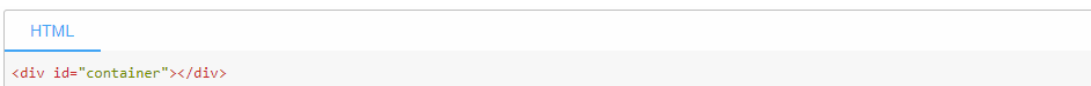
3.3 学习高德 JS API 的使用

① 准备页面

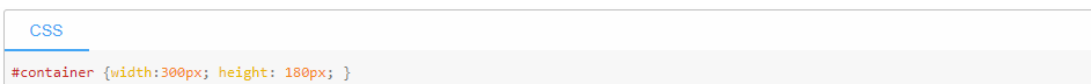
1. 在页面添加 JS API 的入口脚本标签，并将其中「您申请的 key 值」替换为您刚刚申请的 key；



2. 添加 div 标签作为地图容器，同时为该 div 指定 id 属性；



3. 为地图容器指定高度、宽度；



4. 进行移动端开发时，请在 head 内添加 viewport 设置，以达到最佳的绘制性能；

HTML

```
<meta name="viewport" content="initial-scale=1.0, user-scalable=no">
```

5. 在完成如上准备工作之后便可以开始进行开发工作了。

② JS API 的加载

1. 顺序同步加载方式

HTML

```
<script src="https://webapi.amap.com/maps?v=1.4.15&key=您申请的key值"></script>
<script type="text/javascript">
    var map = new AMap.Map('container', {
        center:[117.000923,36.675807],
        zoom:11
    });
</script>
```

2. 异步加载

异步加载指的是为 JS API 指定加载的回调函数，在 JS API 的主体资源加载完毕之后，将自动调用该回调函数。回调函数应该声明在 JS API 入口文件引用之前，异步加载可以减少对其他脚本执行的阻塞，HTTPS 下建议使用异步方式：

HTML

```
<script type="text/javascript">
    window.init = function(){
        var map = new AMap.Map('container', {
            center:[117.000923,36.675807],
            zoom:11
        });
    }
</script>
<script src="https://webapi.amap.com/maps?v=1.4.15&key=您申请的key值&callback=window.init"></script>
```

或者

JavaScript

```
window.onload = function(){
    var map = new AMap.Map('container');
}
var url = 'https://webapi.amap.com/maps?v=1.4.15&key=您申请的key值&callback=onLoad';
var jsapi = document.createElement('script');
jsapi.charset = 'utf-8';
jsapi.src = url;
document.head.appendChild(jsapi);
```

如果需要异步加载，比如通过 `appendChild`，或者通过 `require` 等异步方式来加载，这时需要先准备一个全局的回调函数作为 JS API 异步加载的回调函数，并将其函数名作为 `callback` 参数添加在 JS API 的引用地址后面，此时要注意，回调函数应该在脚本请求发出之前进行声明。异步加载方式只有在回调之后，才能开始调用 JSAPI 的相关接口。比如

JavaScript

```
window.onload = function(){
    var map = new AMap.Map('container');
}
var url = 'https://webapi.amap.com/maps?v=1.4.15&key=您申请的key值&callback=onLoad';
var jsapi = doc.createElement('script');
jsapi.charset = 'utf-8';
jsapi.src = url;
document.head.appendChild(jsapi);
```

③ 创建地图

简单创建一个地图只需要一行代码，构造参数中的 `container` 为准备阶段添加的地图容器的 `id`：

JavaScript

```
var map = new AMap.Map('container');
```

创建的同时可以给地图设置中心点、级别、显示模式、自定义样式等属性：

JavaScript

```
var map = new AMap.Map('container', {
    zoom: 11, // 级别
    center: [116.397428, 39.90923], // 中心点坐标
    viewMode: '3D' // 使用 3D 视图
});
```

④ 图层

默认情况下，地图只显示标准底图，如需要叠加别的图层，可以通过 `map.add` 方法添加图层：

JavaScript

```
var map = new AMap.Map('container', {
    resizeEnable: true,
    center: [116.397428, 39.90923],
    zoom: 13
});
// 实时路况图层
var trafficLayer = new AMap.TileLayer.Traffic({
    zIndex: 10
});
map.add(trafficLayer); // 添加图层到地图
```

也可以在地图初始化的时候通过 layers 属性为地图设置多个图层：

JavaScript

```
var map = new AMap.Map('container', {
  center: [116.397428, 39.90923],
  layers: [//使用多个图层
    new AMap.TileLayer.Satellite(),
    new AMap.TileLayer.RoadNet()
  ],
  zooms: [4,18],//设置地图级别范围
  zoom: 13
});
```

⑤ 点标记与矢量图形

JS API 提供了在地图之上绘制覆盖物的能力，比如点标记 Marker、文本标记 Text、圆点标记 CircleMarker。添加点标记的方法非常简单，比如添加一个默认样式的 Marker：

JavaScript

```
var marker = new AMap.Marker({
  position:[116.39, 39.9]//位置
})
map.add(marker);//添加到地图
```

移除的命令为 map.remove(marker)。

高德 js api 也提供了绘制圆 Circle、折线 Polyline、多边形 Polygon、椭圆 Ellipse、矩形 Rectangle、贝瑟尔曲线 BBezierCurve 等矢量图形的能力，比如添加折线：

JavaScript

```
var lineArr = [
  [116.368904, 39.913423],
  [116.382122, 39.901176],
  [116.387271, 39.912501],
  [116.398258, 39.904600]
];
var polyline = new AMap.Polyline({
  path: lineArr, //设置线覆盖物路径
  strokeColor: "#3366FF", //线颜色
  strokeWeight: 5, //线宽
  strokeStyle: "solid", //线样式
});
map.add(polyline);
```

⑥ 事件功能与信息窗体

JS API 提供的 Map、点标记、矢量图形的实例均支持事件，鼠标或者触摸操作均会触发相应的事件。我们通过给点标记绑定 click 事件来简单了解事件系统和信息窗体的基本使用：

JavaScript

```
var infoWindow = new AMap.InfoWindow({ //创建信息窗体
  isCustom: true, //使用自定义窗体
  content: '<div>信息窗体</div>', //信息窗体的内容可以是任意html片段
  offset: new AMap.Pixel(16, -45)
});
var onMarkerClick = function(e) {
  infoWindow.open(map, e.target.getPosition());//打开信息窗体
  //e.target就是被点击的Marker
}
var marker = new AMap.Marker({
  position: [116.481181, 39.989792]
})
map.add(marker);
marker.on('click',onMarkerClick);//绑定click事件
```

⑦ 插件的引入与使用

1. 异步加载插件

异步加载指的是在 JS API 加载完成之后，在需要使用到某个插件的时候，通过 AMap.plugin 方法按需引入插件，在 plugin 回调之后使用插件功能。

HTML

```
<script type="text/javascript" src="https://webapi.amap.com/maps?v=1.4.15&key=您申请的key值"></script>
<script type="text/javascript" >
  var map = new AMap.Map('container',{
    zoom:12,
    center:[116.39,39.9]
  });
  AMap.plugin('AMap.ToolBar',function(){//异步加载插件
    var toolbar = new AMap.ToolBar();
    map.addControl(toolbar);
  });
</script>
```

2. 异步加载多个插件

AMap.plugin 的第一个参数使用数组即可同时加载引入多个插件。

JavaScript

```
AMap.plugin(['AMap.ToolBar','AMap.Driving'],function(){//异步同时加载多个插件
  var toolbar = new AMap.ToolBar();
  map.addControl(toolbar);
  var driving = new AMap.Driving();//驾车路线规划
  driving.search(/*参数*/)
});
```

3. 同步加载插件

如果希望和 JS API 的主体同步加载某些插件，而不是异步加载，可以在 JS API 的

入口地址中添加 plugin 参数，将需要使用的一个或者多个插件的名称作为参数即可，这种用法在 JS API 加载完毕之后便可直接使用插件功能了。

```
HTML

<script type="text/javascript" src="https://webapi.amap.com/maps?v=1.4.15&key=您申请的key值&plugin=AMap.ToolBar"></script>
<script type="text/javascript" >
    var map = new AMap.Map('mapContainer',{
        zoom:12,
        center:[116.39,39.9]
    });
    var toolbar = new AMap.ToolBar();
    map.plugin(toolbar);
</script>
```

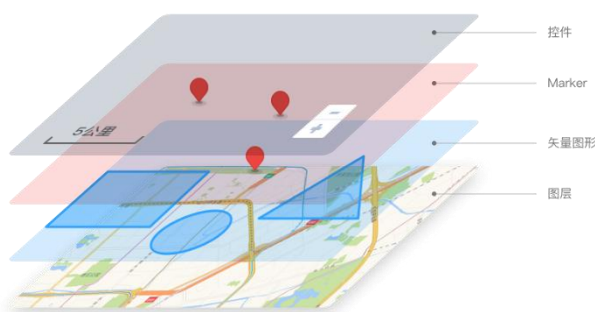
4. 同步加载多个插件

需要加载多个插件时，plugin 参数中的插件名称之间以逗号分割。

```
HTML

<script type="text/javascript" src="https://webapi.amap.com/maps?v=1.4.15&key=您申请的key值&plugin=AMap.ToolBar,AMap.Driving"></script>
<script type="text/javascript" >
    var map = new AMap.Map('mapContainer',{
        zoom:12,
        center:[116.39,39.9]
    });
    var toolbar = new AMap.ToolBar();
    map.plugin(toolbar);
    var driving = new AMap.Driving();
    driving.search(/*参数*/)
</script>
```

⑧ JS API 地图组成



使用高德地图 JS API 创建的地图通常由这几部分组成：

1. 地图容器 Container

即在准备阶段所创建的指定了 id 的 div 对象，这个 div 将作为承载所有图层、点标记、矢量图形、控件的容器。

2. 图层 Layers

图层是指能够在视觉上覆盖一定地图范围，用来描述全部或者部分现实世界区域内的地图要素的抽象概念，一幅地图通常由一个或者多个图层组成。如上图中处于整个地图容器最

下方的二维矢量图层和实施交通图层。

高德地图 JS API 即提供了标准图层、卫星图层、路网图层、路况图层等官方标准图层接口，又提供了加载其他厂商发布的标准地图服务的图层接口，还提供了加载开发者自己的图片、Canvas、视频、热力数据的图层接口。同时也深度开放了可以自定义绘制方法的自定义图层接口。

3. 矢量图形 Vector Overlays

矢量图形，一般覆盖于底图图层之上，通过矢量的方式(路径或者实际大小)来描述其形状，用几何的方式来展示真实的地图要素，会随着地图缩放而发生视觉大小的变化，但其代表的实际路径或范围不变，如上图中红框内的折线、圆、多边形等。

除了图中的折线、圆、多边形之外，JS API 还提供了矩形、椭圆、贝塞尔曲线等常用的矢量图形。3D 视图下的 Mesh、Prism 棱柱、MeshLine 也可以看做是一些特殊的矢量图形。

4. 点标记 Markers

点标记是用来标示某个位置点信息的一种地图要素，覆盖于图层之上。如图中蓝色方框中的两个点状要素。其在屏幕上的位置会随着地图的缩放和中心变化而发生改变，但是会与图层内的要素保持相对静止。

普通的点标记（即 Marker），支持将自定义图标和 DOM 元素作为展示内容。除了 Marker 之外为了满足不同的场景，JS API 另外提供了丰富的点标记类型，比如圆点标记（CircleMarker）、文本标记（Text）、灵活点标记（ElasticMarker）。同时我们提供了海量点（MassMarks）、点聚合（MarkerCluster）来满足大量点标记的展示需求。

5. 地图控件 Map Controls

控件浮在所有图层和地图要素之上，用于满足一定的交互或提示功能。一般相对于地图容器静止，不随着地图缩放和中心变化而发生位置的变化。如上图中绿色方框中的比例尺和级别控件。

四、功能开发

4.1 界面设计

① 编写网页框架：

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
  <link rel="stylesheet" href="https://a.amap.com/jsapi_demos/static/demo-center/css/demo-center.css" />
  <title>17313030 吴敏毅 AMap_GZ</title>
</head>
<body>
</body>
</html>
```

② 将<style>加入<head>;

```

<style type="text/css">
  html,
  body,
  #container {
    width: 100%;
    height: 100%;
  }
  .input-item {
    height: 2rem;
  }
  .btn {
    width: 6rem;
    margin: 0 1rem 0 2rem;
  }
  .input-text {
    width: 4rem;
    margin-right: 1rem;
  }
  #panel {
    position: fixed;
    background-color: white;
    max-height: 80%;
    overflow-y: auto;
    top: 10px;
    right: 10px;
    width: 280px;
    border-bottom: solid 1px silver;
  }
</style>

```

Container 用于承载地图

Input-item 即“周边/沿线/多边形/矩形/圆形搜索”选项

Btn 即“清除标注/重置搜索”按钮

Input-text 即“搜索框”组件

Panel 用于放搜索或查询结果

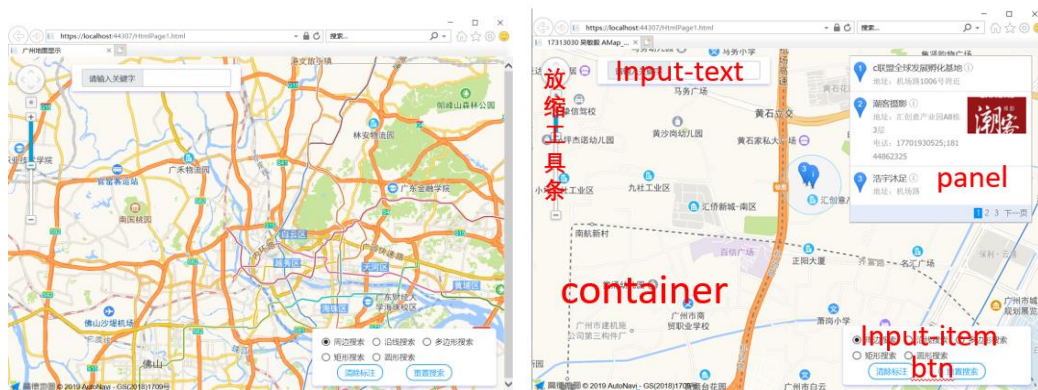
③ 对<body>进行细化，添加组件，具体设置部件的类别和使用函数：

```

<body>
  <div id="container"></div>
  <div id="panel"></div>
  <div class="info" style="width: 10rem; left: 100px;">
    <div class="input-item">
      <div class="input-item-prepend">
        <span class="input-item-text" style="width: 8rem;">请输入关键字</span>
      </div>
      <input id="tipinput" type="text">
    </div>
  </div>
  <div class="input-card" style="width: 24rem;">
    <div class="input-item">
      <input type="radio" name="func" checked="" value="SearchAround"><span class="input-text">周边搜索</span>
      <input type="radio" name="func" value="SearchByLine"><span class="input-text">沿线搜索</span>
      <input type="radio" name="func" value="SearchByPol"><span class="input-text" style="width: 5rem;">多边形搜索</span>
    </div>
    <div class="input-item">
      <input type="radio" name="func" value="SearchByRec"><span class="input-text">矩形搜索</span>
      <input type="radio" name="func" value="SearchByCircle"><span class="input-text">圆形搜索</span>
    </div>
    <div class="input-item">
      <input id="clear" type="button" class="btn" value="清除标注" />
      <input id="close" type="button" class="btn" value="重置搜索" />
    </div>
  </div>
</body>

```

④ 界面分为四个部分：地图底图、放缩工具条、搜索框、范围搜索工具箱。最终的展示界面如图：



接下来详细介绍本次实验各个任务的实现步骤。

4.2 显示广州地图并添加放缩工具条

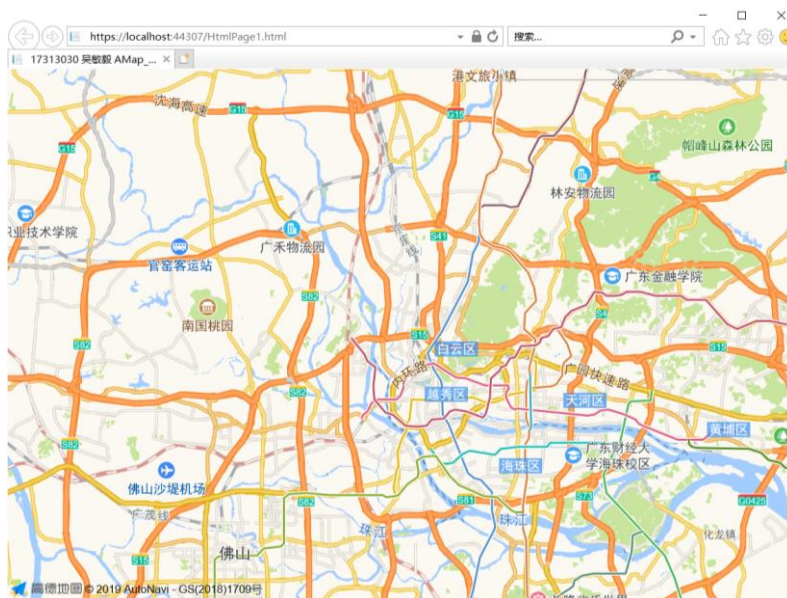
① 在<body>中加载 JSAPI 脚本；

```
<script src="https://webapi.amap.com/maps?v=1.4.15&key=b876b7175121d37fefe5f27681cd1937"></script>
```

② 创建新地图，定义地图中心在[113.23333, 23.16667]，即可将城市定位在广州；

```
var map = new AMap.Map('container', {  
  resizeEnable: true, //是否监控地图容器尺寸变化  
  zoom: 11, //初始化地图层级  
  center: [113.23333, 23.16667] //初始化地图中心点在广州  
});
```

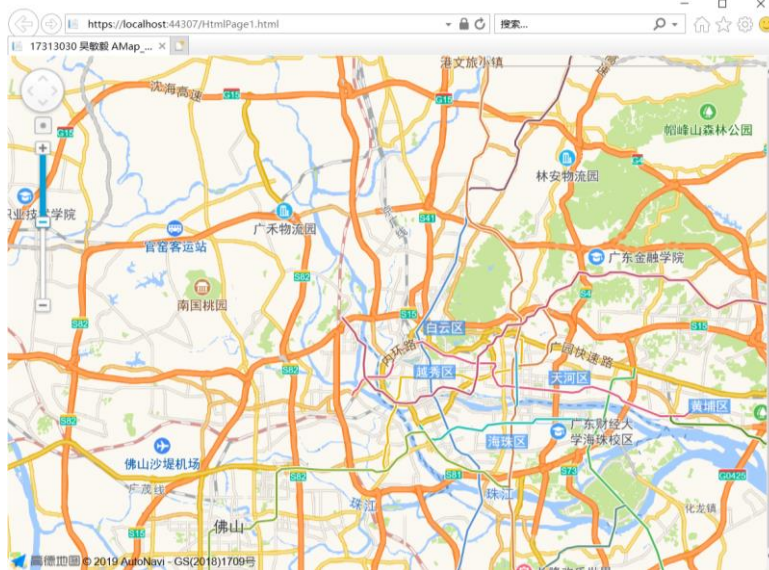
效果展示：



③ 在<script>中加入代码，异步加载 AMap.ToolBar 工具，导入放缩工具条

```
AMap.plugin('AMap.ToolBar', function () { //异步加载ToolBar插件：放缩工具条  
  var toolbar = new AMap.ToolBar();  
  map.addControl(toolbar);  
});
```

效果展示：



4.3 实现 POI 查询功能

① 引入 `AMap.Autocomplete`, `AMap.PlaceSearch` 插件;

```
<script src="https://webapi.amap.com/maps?v=1.4.15&key=b876b717512d37fe5f27681cd1937&plugin=AMap.Autocomplete,AMap.PlaceSearch"></script>
```

② 定义搜索框;

```
<div class="info" style="width: 10rem;left:100px;">
  <div class="input-item">
    <div class="input-item-prepend">
      <span class="input-item-text" style="width:8rem;">请输入关键字</span>
    </div>
    <input id='tipinput' type="text">
  </div>
</div>
```

③ 在<script>中编写 poi 搜索模块;

```
var auto = new AMap.Autocomplete({//定义搜索框监听变量auto
  input: "tipinput"
});

var placeSearch = new AMap.PlaceSearch({//定义搜索变量pS
  map: map
});

AMap.event.addListener(auto, "select", select) //注册监听, 当选中某条记录时会触发

function select(e) { //针对选中的poi实现“标记”和“导入数据库”功能
  placeSearch.setCity(e.poi.adcode);
  placeSearch.search(e.poi.name, function (status, result) {
    // 查询成功时, result即对应匹配的POI信息
    console.log(result);
    var pois = result.poiList.pois;
    for (var i = 0; i < pois.length; i++) {
      var poi = pois[i];
      var marker = [];
      marker[i] = new AMap.Marker({
        position: poi.location, //经纬度对象, 也可以是一维数组[116.39, 39.9]
        title: poi.name
      });
      // 将创建的点标记添加到已有的地图实例:
      map.add(marker[i]);
      addPOIData(poi.name, poi.id, poi.location.lng, poi.location.lat)
      //aPOI函数的实参为poi类的变量名
    }
    map.setFitView();
  });
};
```


④ 编写 addPOIData 函数：

```
function addPOIData(strName, strId, strLongitude, strLatitude) { //全局函数, 数据导入mdb
    //Insert data into the database
    var conn = new ActiveXObject("ADODB.Connection");
    conn.Open("DBQ=D:\\大三上\\GIS\\gis第二次作业\\gdpoi.mdb;DRIVER={Microsoft Access Driver (*.mdb)}");
    var sql = "insert into poi(Name,poiID,Lng,Lat) values('"+strName+"','"+strId+"','"+strLongitude+"','"+strLatitude+"')";
    try {
        conn.execute(sql);
    }
    catch (e) {
        document.write(e.description);
    }
}
```

⑤ 和数据库建立联系

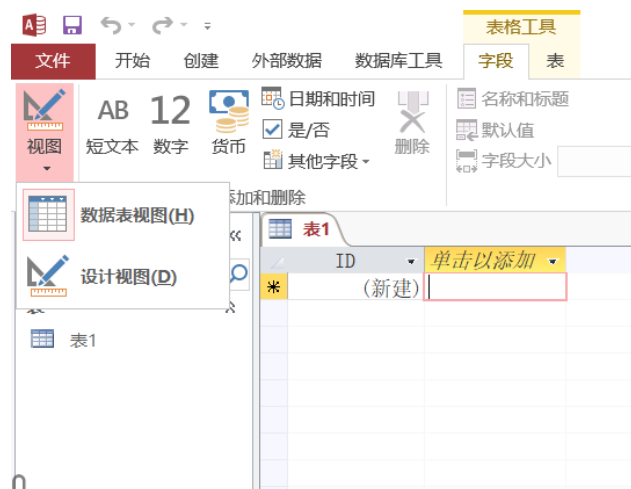
1. 创建空数据库



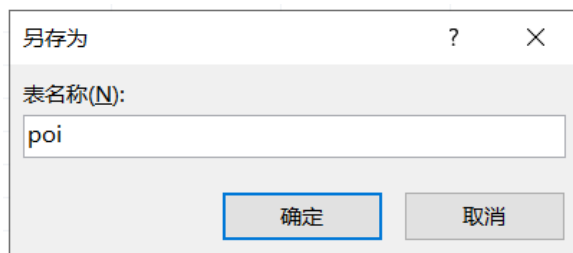
2. 填入数据库名称，更改路径



3. 打开文件，并选中设计视图：



4. 新建表，名称设置为 poi



5. 键入数据表各数据项名称

poi	
字段名称	数据类型
ID	自动编号
Name	长文本
poiID	长文本
Lng	长文本
Lat	长文本

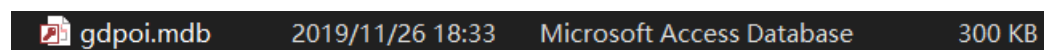
6. 返回主界面，点击另存为



7. 选择 access2002-2003 数据库 (*.mdb) 版本

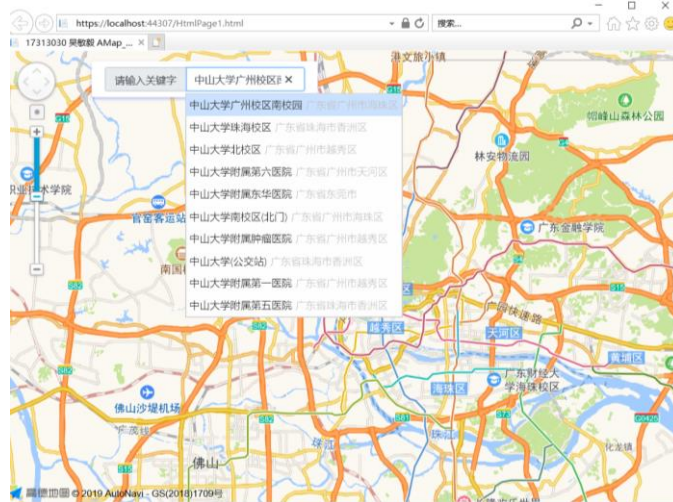


8. Ctrl+s 保存为 gd.mdb 文件后退出

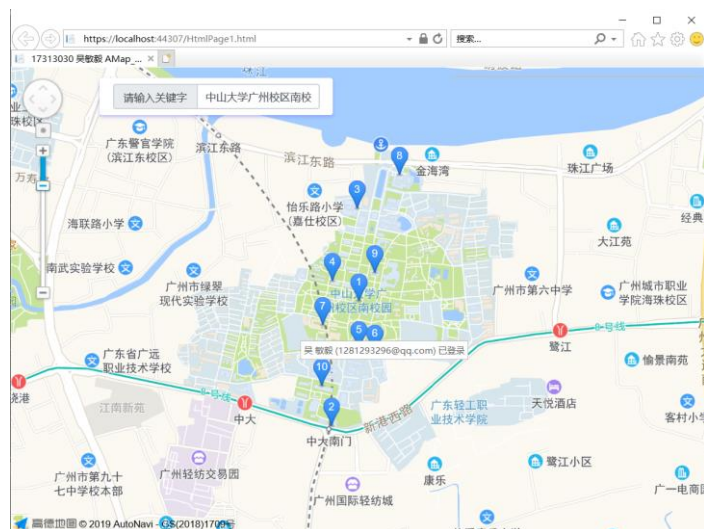
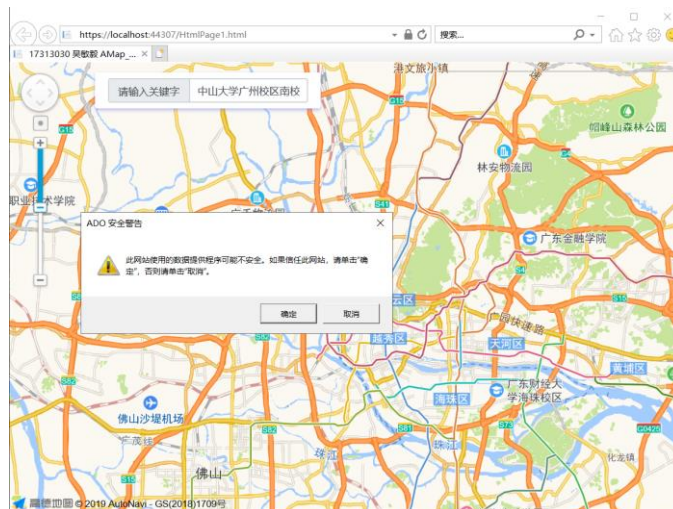


效果展示：

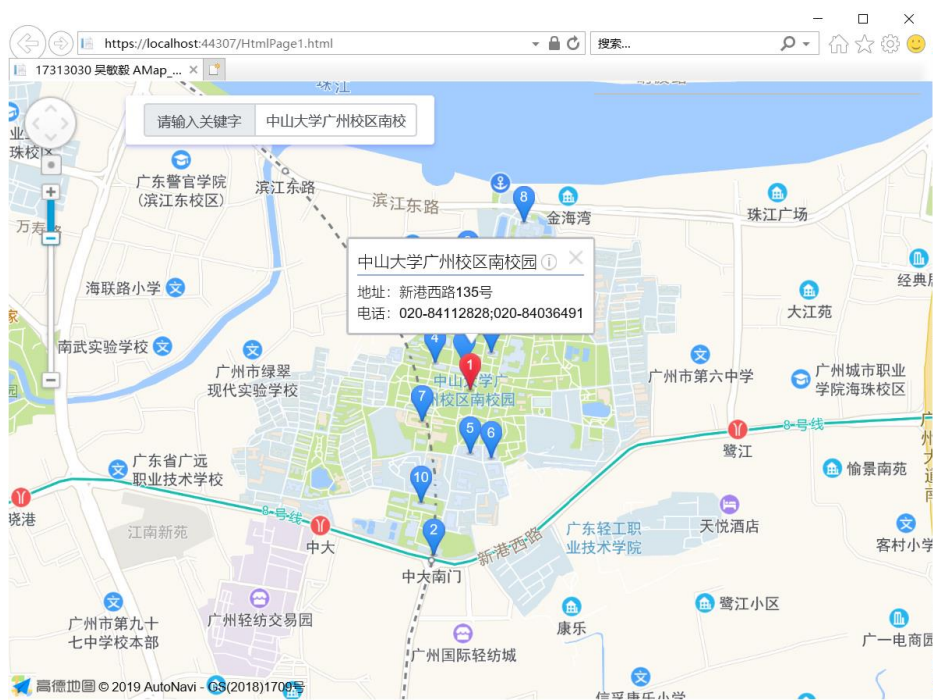
1. 使用搜索功能，在搜索栏键入“中山大学南校园”字样，弹出搜索结果



2. 连续点击弹窗确认按钮后呈现搜索结果



3. 点击标志时，可以出现地点详情



4. 打开 gdpoi.mdb 文件可看到导入的搜索记录

ID	Name	poiID	Lng	Lat
269	中山大学广州校区南校园	B00141IHRZ	113.298379	23.096705
270	中山大学广州校区南校园(南门)	B00140TVAB	113.29705	23.091124
271	中山大学南校区(北门)	B00140TAV2	113.300039	23.102687
272	中山大学南校区管理学院	B0FFFDQNL0	113.29948	23.100002
273	中山大学广州南校区震寰堂	B0FFFDDBVQ	113.292372	23.093256
274	中山大学广州南校区法学院	B00140TVAP	113.296932	23.096685
275	中山大学南校区附属小学	B00140TFUF	113.290511	23.094417
276	中山大学南校区网络教育学院	B0FFH6M6RR	113.304185	23.095392
277	中山大学南校区公共实验教学平台	B0FF9YJU5	113.304094	23.095864
278	中山大学南校区幼儿园瑞康园区	B0FFJHVQC9	113.296974	23.089407
279	中山大学广州校区南校园	B00141IHRZ	113.298379	23.096705
280	中山大学广州校区南校园(南门)	B00140TVAB	113.29705	23.091124
281	中山大学广州校区南校园丰盛堂	B0FFK4B6NF	113.298302	23.10083
282	中山大学广州校区南校园十友堂	B0FFK4B6NG	113.297079	23.097588
283	中山大学广州校区南校园文科大楼	B00141I5JW	113.298375	23.094558
284	中山大学广州校区南校园中文堂	B0FFK4GAFY	113.299123	23.094426
285	中山大学广州校区南校园-逸夫楼	B0FFGKE0IC	113.296629	23.095624
286	中山大学广州校区南校园岭南MBA教学楼	B0FFK4IH8T	113.300357	23.102348
287	中山大学广州校区南校园教学楼	B0FFG4FL8I	113.299157	23.097958
288	中山大学广州校区南校园生物楼	B0FFK4LKE6	113.296554	23.092891

4.3 实现特色查询功能

① <body>中引入 AMap.MouseTool 插件；

```
<script src="https://webapi.amap.com/maps?v=1.4.15&key=b876b7175121d37fefe5f27681cd1937&plugin=AMap.MouseTool,AMap.Autocomplete,AMap.PlaceSearch"></script>
```

② 定义操作面板：

```


<div class="input-item">
    <input type="radio" name="func" checked="" value="SearchAround"><span class="input-text">周边搜索</span>
    <input type="radio" name="func" value="SearchByLine"><span class="input-text">沿线搜索</span>
    <input type="radio" name="func" value="SearchByPol"><span class="input-text" style="width:5rem;">多边形搜索</span>
  </div>
  <div class="input-item">
    <input type="radio" name="func" value="SearchByRec"><span class="input-text">矩形搜索</span>
    <input type="radio" name="func" value="SearchByCircle"><span class="input-text">圆形搜索</span>
  </div>
  <div class="input-item">
    <input id="clear" type="button" class="btn" value="清除标注" />
    <input id="close" type="button" class="btn" value="重置搜索" />
  </div>
</div>


```

③ 在<script>中编写特色搜索框架：

```

var mouseTool = new AMap.MouseTool(map); //定义鼠标操作工具变量

var overlays = []; //定义画图图层

mouseTool.on('draw', function (e) {}); //监听鼠标选择的函数

var radios = document.getElementsByName('func'); //操作数组，用以针对选择进行相应的draw操作
for (var i = 0; i < radios.length; i += 1) {
  radios[i].onchange = function (e) {
    draw(e.target.value);
  }
}

function draw(type) {} //画图函数，点/线/多边形/矩形/圆
draw('SearchAround'); //默认draw函数为SearchAround为起始

document.getElementById('clear').onclick = function () {}; //清除标记

document.getElementById('close').onclick = function () {}; //重置搜索

function SearchAround(e) {} //周边搜索
function SearchByLine(e) {} //沿线搜索
function SearchByPol(e) {} //多边形搜索
function SearchByRec(e) {} //矩形搜索
function SearchByCircle(e) {} //圆形搜索

```

④ 编写监听鼠标选择的函数：

```

mouseTool.on('draw', function (e) { //监听draw事件可获取画好的覆盖物
  overlays.push(e.obj);
  switch (e.obj.CLASS_NAME) {
    case 'AMap.Marker': {
      SearchAround(e); break;
    }
    case 'AMap.Polyline': {
      SearchByLine(e); break;
    }
    case 'AMap.Polygon': {
      SearchByPol(e); break;
    }
    case 'AMap.Rectangle': {
      SearchByRec(e); break;
    }
    case 'AMap.Circle': {
      SearchByCircle(e); break;
    }
  }
}); //监听鼠标选择的函数

```

⑤编写画图函数，可以画点/线/多边形/矩形/圆；

```
function draw(type) {
    switch (type) {
        case 'SearchAround': {
            mouseTool.marker(); break;
        }
        case 'SearchByLine': {
            mouseTool.polyline({ strokeColor: '#80d8ff' });
            break;
        }
        case 'SearchByPol': {
            mouseTool.polygon({
                fillColor: '#00b0ff',
                strokeColor: '#80d8ff'
                //同Polygon的Option设置
            });
            break;
        }
        case 'SearchByRec': {
            mouseTool.rectangle({
                fillColor: '#00b0ff',
                strokeColor: '#80d8ff'
                //同Polygon的Option设置
            });
            break;
        }
        case 'SearchByCircle': {
            mouseTool.circle({
                fillColor: '#00b0ff',
                strokeColor: '#80d8ff' //同Circle的Option设置
            });
            break;
        }
    }
}
//画图函数，点/线/多边形/矩形/圆
```

⑥ 编写清除标记和重置搜索函数；

```
document.getElementById('clear').onclick = function () {
    mouseTool.close(true) //关闭，并清除覆盖物
    var btContent = '';
    var resLine = '';
    map.clearMap();
    document.getElementById("panel").innerHTML = '&nbsp;'; //清空panel内的标记
}; //清除标记

document.getElementById('close').onclick = function () {
    mouseTool.close(true) //关闭，并清除覆盖物
    for (var i = 0; i < radios.length; i += 1) {
        radios[i].checked = false;
    }
}; //重置搜索
```

⑦ 编写特色搜索函数：

1. 周边搜索

```
function SearchAround(e) {
    var placeSearch = new AMap.PlaceSearch({
        pageSize: 3, // 单页显示结果条数
        pageIndex: 1, // 页码
        city: '020', // 兴趣点城市
        citylimit: true, //是否强制限制在设置的城市内搜索
        map: map, // 展现结果的地图实例
        panel: "panel", // 结果列表将在panel中进行展示。
        autoFitView: true // 是否自动调整地图视野使绘制的 Marker点都处于视口的可见范围
    });
    var cpoint = e.obj.getPosition();
    placeSearch.searchNearBy('', cpoint, 200, function (status, result) {
        var pois = result.poiList.pois;
        for (var i = 0; i < pois.length; i++) {
            var poi = pois[i];
            var marker = [];
            marker[i] = new AMap.Marker({
                position: poi.location, //经纬度对象，也可以是一维数组[116.39, 39.9]
                title: poi.name
            });
            // 将创建的点标记添加到已有的地图实例：
            map.add(marker[i]);
            addPOIData(poi.name, poi.id, poi.location.lng, poi.location.lat);
        }
    });
} //周边搜索
```

2. 沿线搜索

```
function SearchByLine(e) {
    var placeSearch = new AMap.PlaceSearch({
        pageSize: 3, // 单页显示结果条数
        pageIndex: 1, // 页码
        city: '020', // 兴趣点城市
        citylimit: true, //是否强制限制在设置的城市内搜索
        map: map, // 展现结果的地图实例
        panel: "panel", // 结果列表将在panel中进行展示。
        autoFitView: true // 是否自动调整地图视野使绘制的 Marker点都处于视口的可见范围
    });
    var polylineArr = e.obj.getPath();
    var polylineArray = [];
    for (i = 0; i < polylineArr.length; i++)
        polylineArray[i] = [polylineArr[i].lng, polylineArr[i].lat];
    var polyline = new AMap.Polygon({
        path: polylineArray, //设置多边形边界路径
        strokeColor: "#FF33FF", //线颜色
        strokeOpacity: 0.2, //线透明度
        strokeWeight: 3, //线宽
        fillColor: "#1791fc", //填充色
        fillOpacity: 0.35 //填充透明度
    });
    placeSearch.searchInBounds('', polyline, function (status, result) {
        var pois = result.poiList.pois;
        for (var i = 0; i < pois.length; i++) {
            var poi = pois[i];
            var marker = [];
            marker[i] = new AMap.Marker({
                position: poi.location, //经纬度对象，也可以是一维数组[116.39, 39.9]
                title: poi.name
            });
            // 将创建的点标记添加到已有的地图实例：
            map.add(marker[i]);
            addPOIData(poi.name, poi.id, poi.location.lng, poi.location.lat);
        }
    });
} //沿线搜索
```

3. 多边形搜索

```
function SearchByPol(e) {
    console.log(e.obj.getPath()); // 获取路径/范围
    var placeSearch = new AMap.PlaceSearch({
        pageSize: 3, // 单页显示结果条数
        pageIndex: 1, // 页码
        city: '020', // 兴趣点城市
        citylimit: true, // 是否强制限制在设置的城市内搜索
        map: map, // 展现结果的地图实例
        panel: "panel", // 结果列表将在panel中进行展示。
        autoFitView: true // 是否自动调整地图视野使绘制的 Marker点都处于视口的可见范围
    });
    var polygonArr = e.obj.getPath();
    var polygonArray = [];
    for (i = 0; i < polygonArr.length; i++)
        polygonArray[i] = [polygonArr[i].lng, polygonArr[i].lat];
    var polygon = new AMap.Polygon({
        path: polygonArray, // 设置多边形边界路径
        strokeColor: "#FF33FF", // 线颜色
        strokeOpacity: 0.2, // 线透明度
        strokeWeight: 3, // 线宽
        fillColor: "#1791fc", // 填充色
        fillOpacity: 0.35 // 填充透明度
    });
    placeSearch.searchInBounds(' ', polygon, function (status, result) {
        var pois = result.poiList.pois;
        for (var i = 0; i < pois.length; i++) {
            var poi = pois[i];
            var marker = [];
            marker[i] = new AMap.Marker({
                position: poi.location, // 经纬度对象，也可以是一维数组[116.39, 39.9]
                title: poi.name
            });
            // 将创建的点标记添加到已有的地图实例：
            map.add(marker[i]);
            addPOIData(poi.name, poi.id, poi.location.lng, poi.location.lat);
        }
    });
} // 多边形搜索
```

4. 矩形搜索

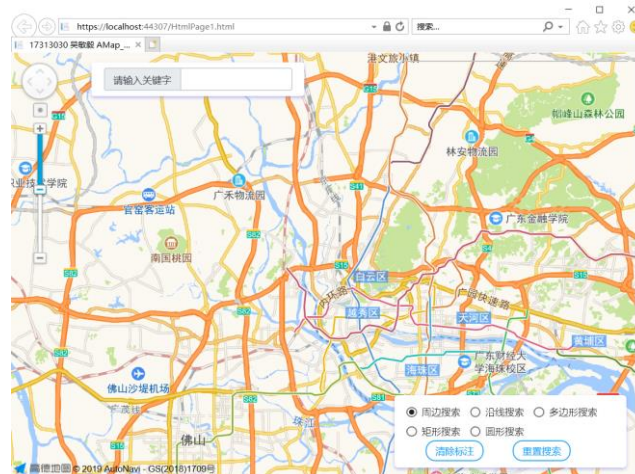
```
function SearchByRec(e) {
    console.log(e.obj.getBounds()); // 获取路径/范围
    var placeSearch = new AMap.PlaceSearch({
        pageSize: 3, // 单页显示结果条数
        pageIndex: 1, // 页码
        city: '020', // 兴趣点城市
        citylimit: true, // 是否强制限制在设置的城市内搜索
        map: map, // 展现结果的地图实例
        panel: "panel", // 结果列表将在panel中进行展示。
        autoFitView: true // 是否自动调整地图视野使绘制的 Marker点都处于视口的可见范围
    });
    placeSearch.searchInBounds(' ', e.obj.getBounds(), function (status, result) {
        var pois = result.poiList.pois;
        for (var i = 0; i < pois.length; i++) {
            var poi = pois[i];
            var marker = [];
            marker[i] = new AMap.Marker({
                position: poi.location, // 经纬度对象，也可以是一维数组[116.39, 39.9]
                title: poi.name
            });
            // 将创建的点标记添加到已有的地图实例：
            map.add(marker[i]);
            addPOIData(poi.name, poi.id, poi.location.lng, poi.location.lat);
        }
    });
} // 矩形搜索
```


5. 圆形搜索

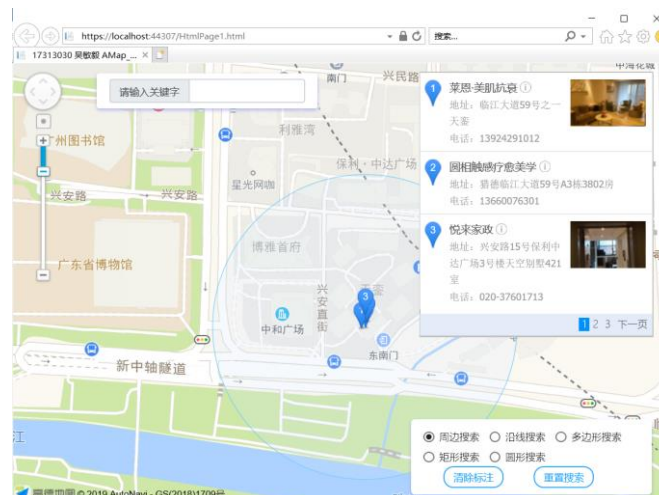
```
function SearchByCircle(e) {  
    var placeSearch = new AMap.PlaceSearch({  
        pageSize: 3, // 单页显示结果条数  
        pageIndex: 1, // 页码  
        city: '020', // 兴趣点城市  
        citylimit: true, //是否强制限制在设置的城市内搜索  
        map: map, // 展现结果的地图实例  
        panel: "panel", // 结果列表将在panel中进行展示。  
        autoFitView: true // 是否自动调整地图视野使绘制的 Marker点都处于视口的可见范围  
    });  
    var cpoint = e.obj.getCenter();  
    var r = e.obj.getRadius();  
    placeSearch.searchNearBy(' ', cpoint, r, function (status, result) {  
        var pois = result.poiList.pois;  
        for (var i = 0; i < pois.length; i++) {  
            var poi = pois[i];  
            var marker = [];  
            marker[i] = new AMap.Marker({  
                position: poi.location, //经纬度对象, 也可以是一维数组[116.39, 39.9]  
                title: poi.name  
            });  
            // 将创建的点标记添加到已有的地图实例:  
            map.add(marker[i]);  
            addPOIData(poi.name, poi.id, poi.location.lng, poi.location.lat);  
        }  
    });  
} //圆形搜索
```

效果展示:

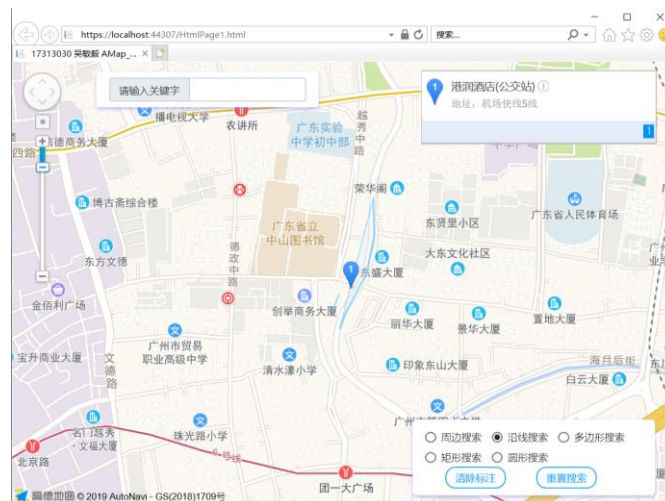
1. 展示界面:



2. 周边搜索功能示例:



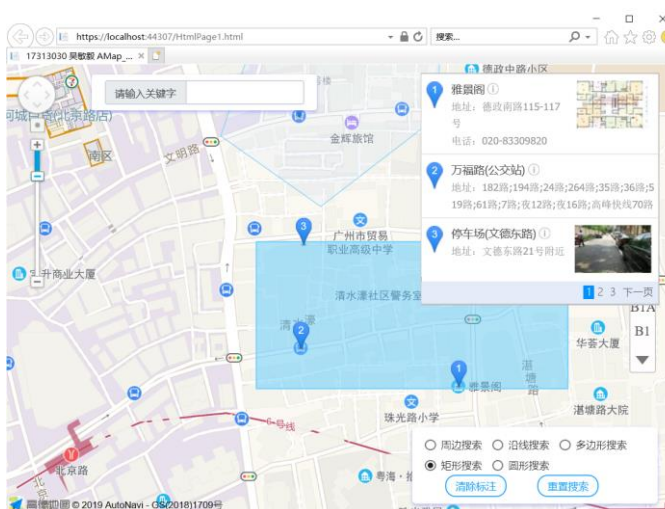
3. 沿线搜索功能示例：



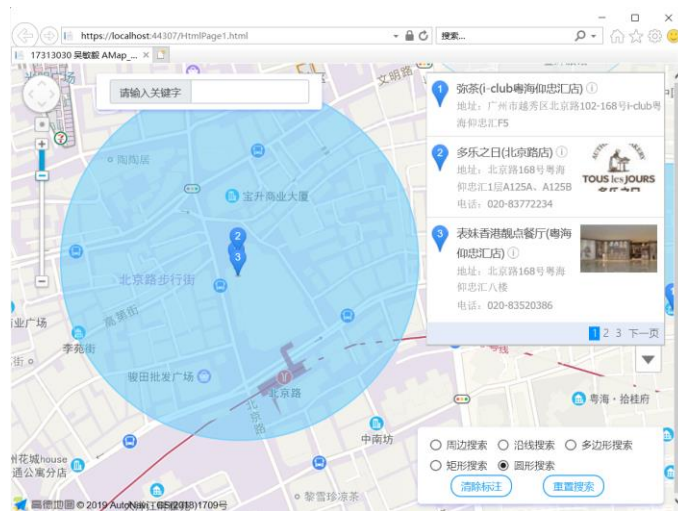
4. 多边形搜索功能示例：



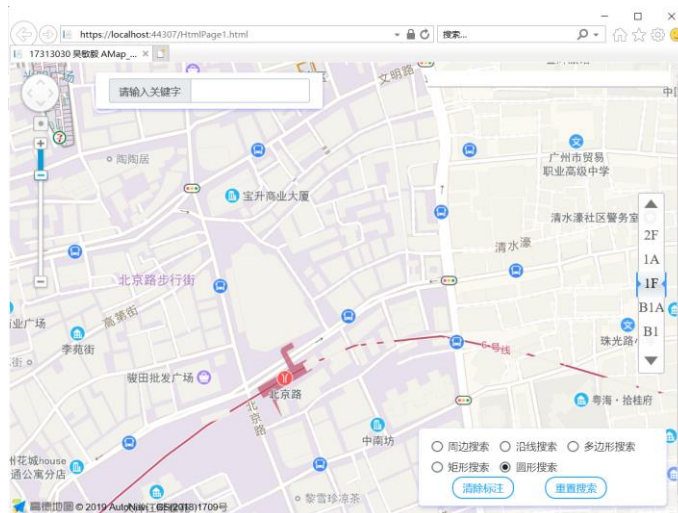
5. 矩形搜索功能示例：



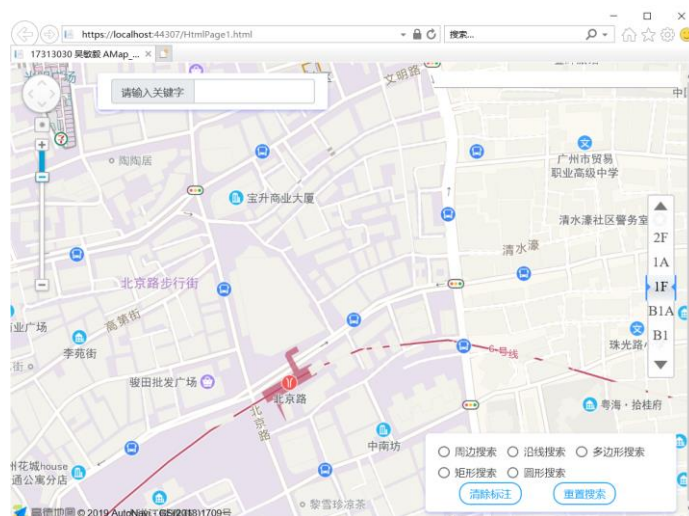
6. 圆形搜索功能示例：



7. 清除标注：（将圆形搜索结果清除）



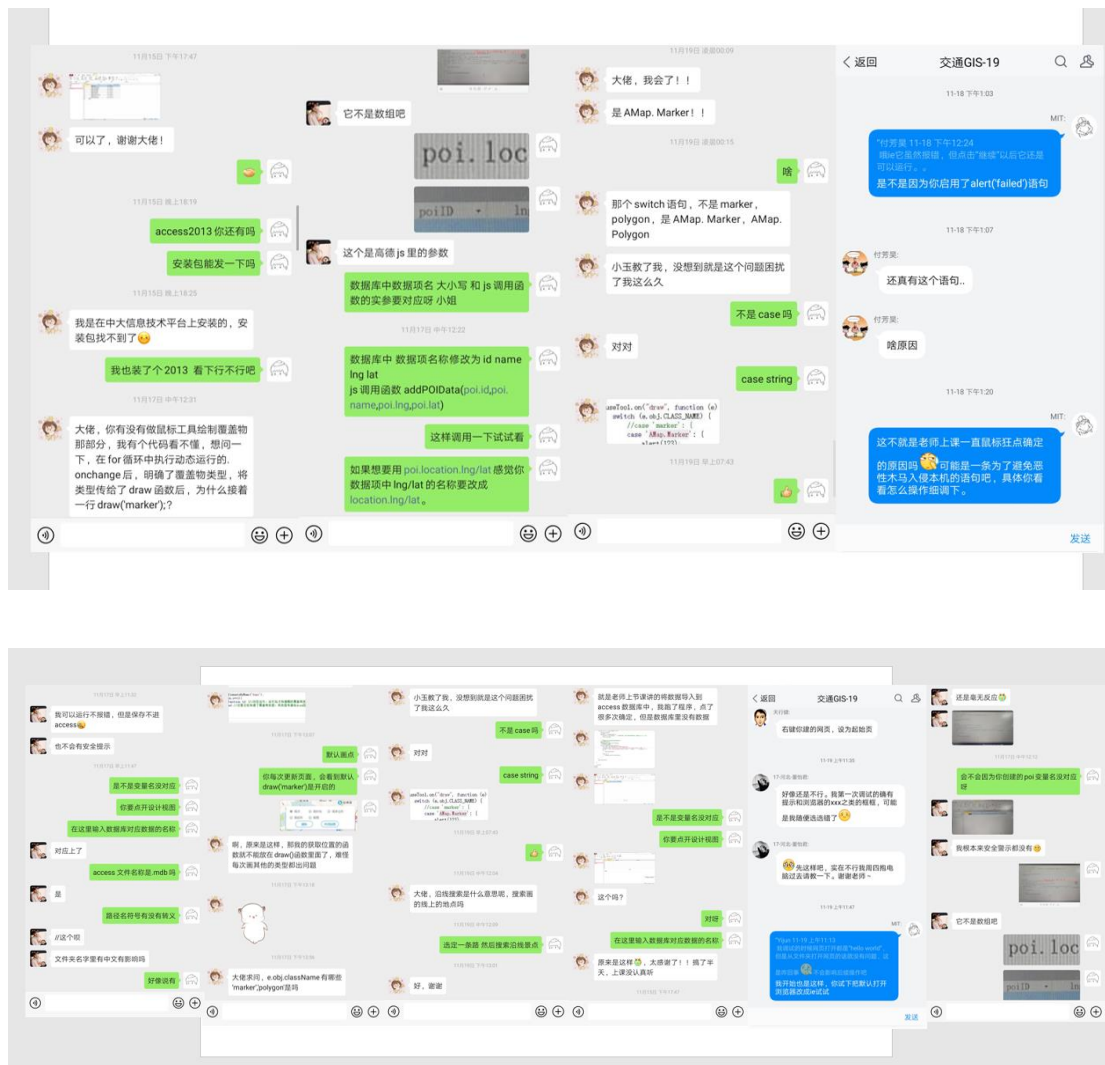
8. 重置搜索（闲置搜索框）



-

- 31 / 44

7. 我和其他同学互帮互助，积极探讨问题和解决方法，这在以后的生活和工作中有很大的好处。不仅帮助了其他同学，而且在问答过程中，我的技能也得到了巩固和加强。



参考文献

[1]Microsoft Access-wikipedia: https://zh.wikipedia.org/wiki/Microsoft_Access

[2]中国各主要大城市经纬度数据-CSDN:
<https://blog.csdn.net/linraise/article/details/50612371>

[3]基于高德地图的空间数据采集实验设计[A], 胡继华, 李珈宜.

附录：

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="initial-scale=1.0, user-
scalable=no, width=device-width">
  <link rel="stylesheet"
href="https://a.amap.com/jsapi_demos/static/demo-center/css/demo-
center.css" />
  <title>17313030 吴敏毅 AMap_GZ</title>
  <style type="text/css">
    html,
    body,
    #container {
      width: 100%;
      height: 100%;
    }

    .input-item {
      height: 2rem;
    }

    .btn {
      width: 6rem;
      margin: 0 1rem 0 2rem;
    }

    .input-text {
      width: 4rem;
      margin-right: 1rem;
    }

    #panel {
      position: fixed;
      background-color: white;
      max-height: 80%;
      overflow-y: auto;
      top: 10px;
      right: 10px;
      width: 280px;
```

```

        border-bottom: solid 1px silver;
    }
</style>
</head>
<body>
    <div id="container"></div>
    <div id="panel"></div>
    <div class="info" style="width: 10rem;left:100px;">
        <div class="input-item">
            <div class="input-item-prepend">
                <span class="input-item-text" style="width:8rem;">请输入关键字</span>
            </div>
            <input id='tipinput' type="text">
        </div>
        <div class="input-card" style='width: 24rem;'>
            <div class="input-item">
                <input type="radio" name='func' checked=""
value='SearchAround'><span class="input-text">周边搜索</span>
                <input type="radio" name='func'
value='SearchByLine'><span class="input-text">沿线搜索</span>
                <input type="radio" name='func' value='SearchByPol'><span
class="input-text" style='width:5rem;'>多边形搜索</span>
            </div>
            <div class="input-item">
                <input type="radio" name='func' value='SearchByRec'><span
class="input-text">矩形搜索</span>
                <input type="radio" name='func'
value='SearchByCircle'><span class="input-text">圆形搜索</span>
            </div>
            <div class="input-item">
                <input id="clear" type="button" class="btn" value="清除标
注" />
                <input id="close" type="button" class="btn" value="重置搜
索" />
            </div>
        </div>
        <!-- 加载地图 JSAPI 脚本 -->
        <script
src="https://webapi.amap.com/maps?v=1.4.15&key=b876b7175121d37fefe5
f27681cd1937&plugin=AMap.MouseTool,AMap.AutoComplete,AMap.PlaceSear
ch"></script>
        <script type="text/javascript">

```

```
var map = new AMap.Map('container', {
    resizeEnable: true, //是否监控地图容器尺寸变化
    zoom: 11, //初始化地图层级
    center: [113.23333, 23.16667] //初始化地图中心点在广州
});

AMap.plugin('AMap.ToolBar', function () { //异步加载 ToolBar 插件：放缩工具条
    var toolbar = new AMap.ToolBar();
    map.addControl(toolbar);
});

var auto = new AMap.Autocomplete({ //定义搜索框监听变量 auto
    input: "tipinput"
});

var placeSearch = new AMap.PlaceSearch({ //定义搜索变量 pS
    map: map
});

AMap.event.addListener(auto, "select", select) //注册监听，当选中某条记录时会触发

function select(e) { //针对选中的 poi 实现“标记”和“导入数据库”功能
    placeSearch.setCity(e.poi.adcode);
    placeSearch.search(e.poi.name, function (status, result)
    {
        // 查询成功时，result 即对应匹配的 POI 信息
        console.log(result);
        var pois = result.poiList.pois;
        for (var i = 0; i < pois.length; i++) {
            var poi = pois[i];
            var marker = [];
            marker[i] = new AMap.Marker({
                position: poi.location, //经纬度对象，也可以是一维数组[116.39,39.9]
                title: poi.name
            });
            // 将创建的点标记添加到已有的地图实例：
            map.add(marker[i]);
            addPOIData(poi.name, poi.id, poi.location.lng, poi.location.lat)
        }
        //aPD 函数的实参为 poi 类的变量名
    });
}
```

```
    }  
    map.setFitView();  
  });  
};  
  
var mouseTool = new AMap.MouseTool(map); //定义鼠标操作工具变量  
  
var overlays = []; //定义画图图层  
  
mouseTool.on('draw', function (e) { //监听 draw 事件可获取画好的  
覆盖物  
    overlays.push(e.obj);  
    switch (e.obj.CLASS_NAME) {  
        case 'AMap.Marker': {  
            SearchAround(e); break;  
        }  
        case 'AMap.Polyline': {  
            SearchByLine(e); break;  
        }  
        case 'AMap.Polygon': {  
            SearchByPol(e); break;  
        }  
        case 'AMap.Rectangle': {  
            SearchByRec(e); break;  
        }  
        case 'AMap.Circle': {  
            SearchByCircle(e); break;  
        }  
    }  
}); //监听鼠标选择的函数  
  
var radios = document.getElementsByName('func'); //操作数组，用  
以针对选择进行相应的 draw 操作  
for (var i = 0; i < radios.length; i += 1) {  
    radios[i].onchange = function (e) {  
        draw(e.target.value);  
    }  
}  
  
function draw(type) {  
    switch (type) {  
        case 'SearchAround': {  
            mouseTool.marker(); break;  
        }  
    }  
}
```

```
        case 'SearchByLine': {
            mouseTool.polyline({ strokeColor: '#80d8ff' });
            break;
        }
        case 'SearchByPol': {
            mouseTool.polygon({
                fillColor: '#00b0ff',
                strokeColor: '#80d8ff'
                //同 Polygon 的 Option 设置
            });
            break;
        }
        case 'SearchByRec': {
            mouseTool.rectangle({
                fillColor: '#00b0ff',
                strokeColor: '#80d8ff'
                //同 Polygon 的 Option 设置
            });
            break;
        }
        case 'SearchByCircle': {
            mouseTool.circle({
                fillColor: '#00b0ff',
                strokeColor: '#80d8ff' //同 Circle 的 Option 设置
            });
            break;
        }
    }
} //画图函数，点/线/多边形/矩形/圆

draw('SearchAround');//默认 draw 函数为 SearchAround 为起始

document.getElementById('clear').onclick = function () {
    mouseTool.close(true) //关闭，并清除覆盖物
    var btContent = '';
    var resLine = '';
    map.clearMap();
    document.getElementById("panel").innerHTML = '&nbsp;'; //清
空 panel 内的标记
}; //清除标记

document.getElementById('close').onclick = function () {
    mouseTool.close(true) //关闭，并清除覆盖物
```

```
        for (var i = 0; i < radios.length; i += 1) {  
            radios[i].checked = false;  
        }  
    }; //重置搜索  
  
    function SearchAround(e) {  
        var placeSearch = new AMap.PlaceSearch({  
            pageSize: 3, // 单页显示结果条数  
            pageIndex: 1, // 页码  
            city: '020', // 兴趣点城市  
            citylimit: true, //是否强制限制在设置的城市内搜索  
            map: map, // 展现结果的地图实例  
            panel: "panel", // 结果列表将在 panel 中进行展示。  
            autoFitView: true // 是否自动调整地图视野使绘制的 Marker 点  
            都处于视口的可见范围  
        });  
        var cpoint = e.obj.getPosition();  
        placeSearch.searchNearBy('', cpoint, 200, function  
(status, result) {  
            var pois = result.poiList.pois;  
            for (var i = 0; i < pois.length; i++) {  
                var poi = pois[i];  
                var marker = [];  
                marker[i] = new AMap.Marker({  
                    position: poi.location, //经纬度对象, 也可以是一  
                    维数组[116.39,39.9]  
                    title: poi.name  
                });  
                // 将创建的点标记添加到已有的地图实例:  
                map.add(marker[i]);  
                addPOIData(poi.name, poi.id, poi.location.lng,  
                poi.location.lat);  
            }  
        });  
    }; //周边搜索  
  
    function SearchByLine(e) {  
        var placeSearch = new AMap.PlaceSearch({  
            pageSize: 3, // 单页显示结果条数  
            pageIndex: 1, // 页码  
            city: '020', // 兴趣点城市  
            citylimit: true, //是否强制限制在设置的城市内搜索  
            map: map, // 展现结果的地图实例  
            panel: "panel", // 结果列表将在 panel 中进行展示。
```



```

        autoFitView: true // 是否自动调整地图视野使绘制的 Marker 点
        都处于视口的可见范围
    });
    var polylineArr = e.obj.getPath();
    var polylineArray = [];
    for (i = 0; i < polylineArr.length; i++)
        polylineArray[i] = [polylineArr[i].lng,
polylineArr[i].lat];
    var polyline = new AMap.Polygon({
        path: polylineArray, //设置多边形边界路径
        strokeColor: "#FF33FF", //线颜色
        strokeOpacity: 0.2, //线透明度
        strokeWeight: 3, //线宽
        fillColor: "#1791fc", //填充色
        fillOpacity: 0.35 //填充透明度
    });
    placeSearch.searchInBounds('', polyline, function
(status, result) {
        var pois = result.poiList.pois;
        for (var i = 0; i < pois.length; i++) {
            var poi = pois[i];
            var marker = [];
            marker[i] = new AMap.Marker({
                position: poi.location, //经纬度对象，也可以是一
维数组 [116.39, 39.9]
                title: poi.name
            });
            // 将创建的点标记添加到已有的地图实例：
            map.add(marker[i]);
            addPOIData(poi.name, poi.id, poi.location.lng,
poi.location.lat);
        }
    });
} //沿线搜索

```

```

function SearchByPol(e) {
    console.log(e.obj.getPath()); //获取路径/范围
    var placeSearch = new AMap.PlaceSearch({
        pageSize: 3, // 单页显示结果条数
        pageIndex: 1, // 页码
        city: '020', // 兴趣点城市
        citylimit: true, //是否强制限制在设置的城市内搜索
        map: map, // 展现结果的地图实例
        panel: "panel", // 结果列表将在 panel 中进行展示。
    });

```

```
        autoFitView: true // 是否自动调整地图视野使绘制的 Marker 点
        都处于视口的可见范围
    });
    var polygonArr = e.obj.getPath();
    var polygonArray = [];
    for (i = 0; i < polygonArr.length; i++)
        polygonArray[i] = [polygonArr[i].lng,
        polygonArr[i].lat];
    var polygon = new AMap.Polygon({
        path: polygonArray, //设置多边形边界路径
        strokeColor: "#FF33FF", //线颜色
        strokeOpacity: 0.2, //线透明度
        strokeWeight: 3, //线宽
        fillColor: "#1791fc", //填充色
        fillOpacity: 0.35 //填充透明度
    });
    placeSearch.searchInBounds('', polygon, function (status,
    result) {
        var pois = result.poiList.pois;
        for (var i = 0; i < pois.length; i++) {
            var poi = pois[i];
            var marker = [];
            marker[i] = new AMap.Marker({
                position: poi.location, //经纬度对象，也可以是一
                维数组[116.39,39.9]
                title: poi.name
            });
            // 将创建的点标记添加到已有的地图实例：
            map.add(marker[i]);
            addPOIData(poi.name, poi.id, poi.location.lng,
            poi.location.lat);
        }
    });
} //多边形搜索

function SearchByRec(e) {
    console.log(e.obj.getBounds()); //获取路径/范围
    var placeSearch = new AMap.PlaceSearch({
        pageSize: 3, // 单页显示结果条数
        pageIndex: 1, // 页码
        city: '020', // 兴趣点城市
        citylimit: true, //是否强制限制在设置的城市内搜索
        map: map, // 展现结果的地图实例
        panel: "panel", // 结果列表将在 panel 中进行展示。
```

```

        autoFitView: true // 是否自动调整地图视野使绘制的 Marker 点
        都处于视口的可见范围
    });
    placeSearch.searchInBounds('', e.obj.getBounds(),
function (status, result) {
    var pois = result.poiList.pois;
    for (var i = 0; i < pois.length; i++) {
        var poi = pois[i];
        var marker = [];
        marker[i] = new AMap.Marker({
            position: poi.location, //经纬度对象, 也可以是一
            维数组[116.39,39.9]
            title: poi.name
        });
        // 将创建的点标记添加到已有的地图实例:
        map.add(marker[i]);
        addPOIData(poi.name, poi.id, poi.location.lng,
poi.location.lat);
    }
});
} //矩形搜索

function SearchByCircle(e) {
    var placeSearch = new AMap.PlaceSearch({
        pageSize: 3, // 单页显示结果条数
        pageIndex: 1, // 页码
        city: '020', // 兴趣点城市
        citylimit: true, //是否强制限制在设置的城市内搜索
        map: map, // 展现结果的地图实例
        panel: "panel", // 结果列表将在 panel 中进行展示。
        autoFitView: true // 是否自动调整地图视野使绘制的 Marker 点
        都处于视口的可见范围
    });
    var cpoint = e.obj.getCenter();
    var r = e.obj.getRadius();
    placeSearch.searchNearBy('', cpoint, r, function (status,
result) {
        var pois = result.poiList.pois;
        for (var i = 0; i < pois.length; i++) {
            var poi = pois[i];
            var marker = [];
            marker[i] = new AMap.Marker({
                position: poi.location, //经纬度对象, 也可以是一
                维数组[116.39,39.9]
            });
        }
    });
}

```

```
        title: poi.name
    });
    // 将创建的点标记添加到已有的地图实例:
    map.add(marker[i]);
    addPOIData(poi.name, poi.id, poi.location.lng,
poi.location.lat);
    }
    });
} // 圆形搜索

function addPOIData(strName, strId, strLongitude,
strLatitude) { // 全局函数, 数据导入 mdb
    // Insert data into the database
    var conn = new ActiveXObject("ADODB.Connection");
    conn.Open("DBQ=D:\\大三上\\GIS\\gis 第二次作业
\\gdpoi.mdb;DRIVER={Microsoft Access Driver (*.mdb)};");
    var sql = "insert into poi(Name,poiID,Lng,Lat) values('"
+ strName + "','" + strId + "','" + strLongitude + "','" +
strLatitude + "')";
    try {
        conn.execute(sql);
    }
    catch (e) {
        document.write(e.description);
    }
}

</script>

</body>
</html>
```