**地理信息网络服务第一次课后作业**

2020302131119 殷雨薇

**一、基于OpenLayers的多个WMS叠加显示与交互**

创建了一个WMS的图层组，利用USGS网站提供的图层的URL来对这些开源数据进行调用，实现了美国岩矿资源分布的展示。在此基础上，实现了移动图像、缩放图像及隐藏图层等功能。

**1、调用WMS图层**

调用WMS图层的主要思路为：

1. 确定可用的WMS地址。本次实习使用美国地质勘探局（USGS）官网提供的可用的OGC WMS。
2. 获取WMS的URL。USGS为每个OGC WMS提供了GetCapabilities接口，点击该接口会访问相应的WMS元数据文档。
3. 获取layer参数。获取到图层的元数据文档之后，查看该WMS支持的图层。
4. 写入HTML文档。将获取到的URL和相应的图层参数按照java的规则写入到HTML文档中。
5. var layers = [
6. new ol.layer.Tile({
7. source: new ol.source.OSM()
8. }),
10. new ol.layer.Tile({
11. source: new ol.source.TileWMS({
12. url: 'https://mrdata.usgs.gov/services/ds801',
13. params: {
14. 'LAYERS': 'ds801','TILED': true
15. },
16. serverType: 'unknown'
17. })
18. }),
20. new ol.layer.Tile({
21. source: new ol.source.TileWMS({
22. url:  'https://mrdata.usgs.gov/services/ngdbrock',
23. params: {'LAYERS': 'Rock\_Geochemistry\_NGDB', 'TILED': true},
24. serverType: 'unknown'
25. })
26. }),
28. new ol.layer.Tile({
29. source: new ol.source.TileWMS({
30. url:  'https://mrdata.usgs.gov/services/ngdbsed',
31. params: {'LAYERS': 'Sediment\_Geochemistry\_NGDB', 'TILED': true},
32. serverType: 'unknown'
33. })
34. }),
36. new ol.layer.Tile({
37. source: new ol.source.TileWMS({
38. url:  'https://mrdata.usgs.gov/services/ngdbconc',
39. params: {'LAYERS': 'Concentrate\_Geochemistry\_NGDB', 'TILED': true},
40. serverType: 'unknown'
41. })
42. }),
43. ];
44. **WMS图层叠加显示**

1、首先设置了一个控件容器，用于存放调整不同图层显示情况的控件。

1. <div class="button-group" id='LayerCbGrp' style="width:100%">
2. <h4>显示/隐藏图层：</h4>
3. <label>
4. <input type="checkbox" checked="checked" onclick="checkOsm(this);"> 底图
5. </label>
6. <label>
7. <input type="checkbox" checked="checked" onclick="checkSoil(this);"> 土壤图层
8. </label>
9. <label>
10. <input type="checkbox" checked="checked" onclick="checkRock(this);"> 岩石图层
11. </label>
12. <label>
13. <input type="checkbox" checked="checked" onclick="checkSediment(this);"> 沉积物图层
14. </label>
15. <label>
16. <input type="checkbox" checked="checked" onclick="checkConcentrate(this);"> 精矿图层
17. </label>
18. </div>

2、以OSM底图为例，当鼠标单击OSM底图图层控件时，触发checkOsm()函数，继而调用该图层对应的setVisible()函数，图层可见性改变。

1. function checkOsm(elem) {
2. layers[0].setVisible(elem.checked);
3. }

其它图层同理，即可实现多图层的叠加显示。

**3、添加地图交互控件**

1. 首先添加了openlayers框架自带的控制地图缩放的相关控件，包括滑块缩放、全屏、鹰眼等。
2. controls: ol.control.defaults().extend([    *// 往地图增加控件*
3. new ol.control.FullScreen(),
4. new ol.control.OverviewMap(),
5. new ol.control.ZoomSlider(),
6. new ol.control.ZoomToExtent()
7. 为了满足更加丰富的交互功能，增加了自定义的放大、缩小以及平移控件，并对其展示样式进行了编辑处理。

以向上移动地图为例：

1. *// 向上移动地图*
2. function moveToUp() {
3. var view = map.getView();
4. var mapCenter = view.getCenter();
5. var pan=ol.animation.pan({
6. duration:200,
7. source:mapCenter
8. });
9. *// 让地图中心的y值增大，即可使得地图向上移动*
10. mapCenter[1] += 300000;
11. view.setCenter(mapCenter);
12. map.beforeRender(pan);
13. map.render();
14. }

设置控件样式：

1. #CtrlBtnGrp button {
2. background-color: #5F81D1;
3. border: none;
4. color: white;
5. padding: 10px 22px;
6. text-align: center;
7. text-decoration: none;
8. display: inline-block;
9. font-size: 16px;
10. }

**二、基于OpenLayers的WFS网页可视化**

首先建立一个底图，由必应地图提供切片数据。再通过WFS协议对时空数据的要求，加载 Geojson 格式数据，由函数ol.layer.Vector获取，函数ol.source.Vector提供矢量图层数据，继承自ol.source.Source。

1. var raster = new ol.layer.Tile({
2. source: new ol.source.BingMaps({
3. imagerySet: 'Aerial',
4. key:'AnodOYdNpyr8LXnJcpIOrFlsFV5LQ25Z2gg-UOTWrn9TCdA5yvx8J2Qq1NOyVrfL'
5. })
6. });
7. var vector = new ol.layer.Vector({
8. source: new ol.source.Vector({
9. format: new ol.format.GeoJSON({ignoreExtraDims: true}),
10. url: 'https://earthquake.usgs.gov/fdsnws/event/1/query?starttime=2016-07-01&endtime=2016-07-15&format=geojson',
11. strategy: ol.loadingstrategy.all
12. })
13. });
14. **整体美化**

将标题和姓名学号居中

1. var divHead=document.getElementById('HeadOfPage');
2. divHead.style.cssText+='text-align:center';

在两个网站最上方均添加了菜单栏，方便跳转浏览不同网站

1. <div class="top">
2. <center>
3. <ul>
4. <li><a href="./WMS.html">WMS</a></li>
5. <li><a href="./WFS.html">WFS</a></li>
6. </ul>
7. </center>
8. </div>

设置菜单栏样式：

1. .top{
2. */\* 设置宽度高度背景颜色 \*/*
3. height: auto; */\*高度改为自动高度\*/*
4. width:100%;
5. margin-left: 0;
7. background:rgb(38, 26, 107);
8. position: relative; */\*固定在顶部\*/*
9. top: 0cm;*/\*离顶部的距离为0\*/*
10. margin-bottom: 3px;
11. }
12. .top ul{
13. */\* 清除ul标签的默认样式 \*/*
14. width: auto;*/\*宽度也改为自动\*/*
15. list-style-type: none;
16. white-space:nowrap;
17. overflow: hidden;
18. margin-left: 5%;
19. */\* margin-top: 0;          \*/*
20. padding: 0;
21. }
22. .top li {
23. float:left; */\* 使li内容横向浮动，即横向排列  \*/*
24. margin-right:2%;  */\* 两个li之间的距离\*/*
25. position: relative;
26. overflow: hidden;
27. padding-top: 0;
28. }
29. .top li a{
30. */\* 设置链接内容显示的格式\*/*
31. display: block; */\* 把链接显示为块元素可使整个链接区域可点击 \*/*
32. color:white;
33. text-align: center;
34. position: relative;
35. padding: 10px;
36. overflow: hidden;
37. text-decoration: none; */\* 去除下划线 \*/*
39. }
40. .top li a:hover{
41. */\* 鼠标选中时背景变为深色 \*/*
42. background-color: #1D1450;
43. }
44. **效果截图**







