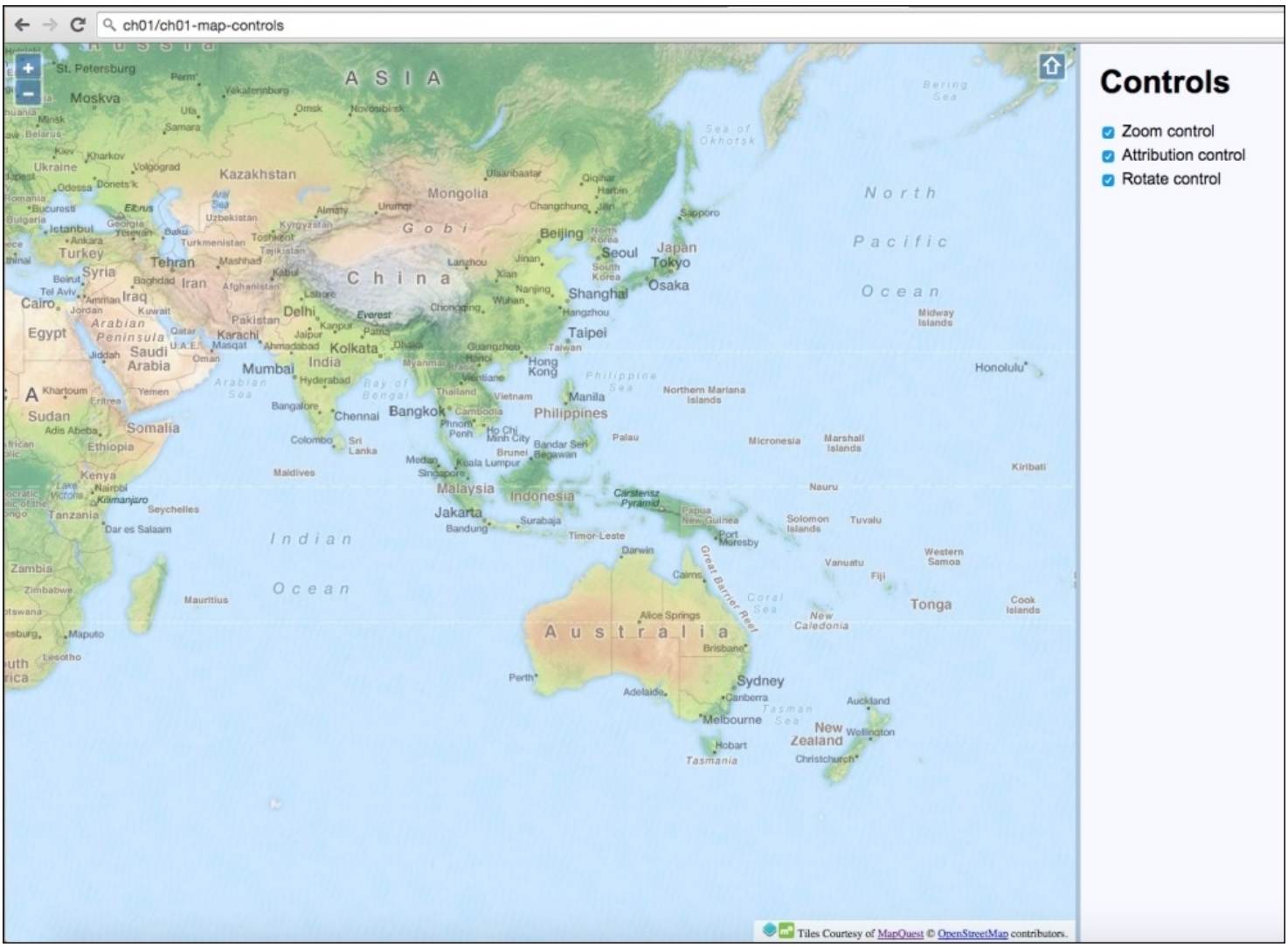
管理地图的控件

OpenLayers附带了很多 与地图互动的控件，例如平移，缩放，显示总览图，编辑要素等。

与一样 `layers` ， `ol.Map` 该类具有管理附加到地图的控件的方法。

我们将创建一种方法来打开或关闭地图控件。可以在中找到源代码 `ch01/ch01-map-controls/` 。这就是我们的最终结果：



怎么做...

1. 创建一个新的HTML文件，并添加OpenLayers依赖项以及jQuery库。特别是，将以下标记添加到正文中：

复制

```
<div id="js-map" class="map"></div>
<div class="pane">
  <h1>Controls</h1>
  <ul id="js-controls">
    <li>
      <label>
        <input type="checkbox" checked value="zoomControl">
        <span>Zoom control</span>
      </label>
    </li>
    <li>
      <label>
        <input type="checkbox" checked value="attributionControl">
        <span>Attribution control</span>
      </label>
    </li>
    <li>
      <label>
        <input type="checkbox" checked value="rotateControl">
```

2 创建一个新的 CSS文件并添加以下内容：

复制

```
.map {
  position: absolute;
  top: 0;
  bottom: 0;
  left: 0;
  right: 20%;
}

.pane {
  position: absolute;
  top: 0;
  bottom: 0;
  right: 0;
  width: 20%;
  background: ghostwhite;
  border-left: 5px solid lightsteelblue;
  box-sizing: border-box;
  padding: 0 20px;
}
```

3 创建一个新的脚本文件并创建映射，如下所示：

复制

```
var map = new ol.Map({
  layers: [
    new ol.layer.Tile({
      source: new ol.source.MapQuest({
        layer: 'osm'
      })
    })
  ],
  view: new ol.View({
    center: [12930000, -78000],
    zoom: 3
  }),
  target: 'js-map',
  controls: []
});
```

4 创建一些控件 并将它们添加到地图中，如下所示：

复制

```
var zoomControl = new ol.control.Zoom({
  zoomInTipLabel: 'Zoom closer in',
  zoomOutTipLabel: 'Zoom further out',
  className: 'ol-zoom custom-zoom-control'
});

var attributionControl = new ol.control.Attribution({
  collapsible: false,
  collapsed: false
});

var rotateControl = new ol.control.Rotate({
  autoHide: false
});

map.addControl(zoomControl);
map.addControl(attributionControl);
map.addControl(rotateControl);
```

5 最后，启用控制切换逻辑：

复制

```
$('#js-controls').on('change', function(event) {
  var target = $(event.target);
  var control = target.val();

  if (target.prop('checked')) {
    map.addControl(window[control]);
  } else {
    map.removeControl(window[control]);
  }
});
```

怎么运行的...

我们的HTML和CSS鸿沟 页面上，使其包含地图和控制面板。在此面板中有三个复选框，分别对应于将添加到地图的三个控件。切换复选框将依次添加或删除所选控件。

请务必注意，复选框的值与JavaScript中控件的变量名匹配。例如， `value="zoomControl"` 将链接到名为的地图控件变量 `zoomControl` 。

让我们拆开OpenLayers代码以了解其工作原理：

复制

```
var map = new ol.Map({

// ...
  controls: []
});
```

该地图实例化代码在以前的食谱中会很熟悉，但是请注意，由于我们不希望OpenLayers在地图上设置任何默认控件，因此我们将一个空数组显式传递给该 `controls` 属性。

复制

```
var zoomControl = new ol.control.Zoom({
  zoomInTipLabel: 'Zoom closer in',
  zoomOutTipLabel: 'Zoom further out',
  className: 'ol-zoom custom-zoom-control'
});
```

我们将对缩放控件的引用存储在 `zoomControl` 变量内。我们已决定自定义出现在加号和减号按钮上的工具提示。还对该 `className` 属性进行了修改，以包括zoom控件的默认类名称（ `ol-zoom` ），以便继承默认的OpenLayers样式和一个自定义类 `custom-zoom-control` 。对于任何覆盖默认样式的样式，我们都可以将此自定义类名称用作CSS钩子。

复制

```
var attributionControl = new ol.control.Attribution({
  collapsible: false,
  collapsed: false
});
```

我们在 `attributionControl` 变量内部存储了对归因控件的引用。此控件通常允许用户折叠属性，默认情况下，其初始状态为折叠。通过指定这两个属性，我们反转了默认值。

复制

```
var rotateControl = new ol.control.Rotate({
  autoHide: false
});
```

我们在 `rotateControl` 变量内存储对旋转控件的引用。通常，仅当地图旋转度不是0时，才显示此控件。我们明确将此控件设置为not自动隐藏自己。

复制

```
map.addControl(zoomControl);
map.addControl(attributionControl);
map.addControl(rotateControl);
```

所有三个控件都添加到 `map` 实例中。

复制

```
$('#js-controls').on('change', function(event) {
  var target = $(event.target);
  var control = target.val();

  if (target.prop('checked')) {
    map.addControl(window[control]);
  } else {
    map.removeControl(window[control]);
  }
});
```

我们利用JavaScript中的事件冒泡功能，并将单个更改事件侦听器附加到包含层列表的HTML；这比将事件侦听器附加到每个输入元素更有效。

切换复选框后，将执行此事件处理程序。事件目标（复选框）被缓存在 `target` 变量内部，因为它已被多次使用。复选框的值（也是映射控件的名称）存储在 `control` 变量中。

此控件的复选框的新状态传递到该 `if` 语句中。如果启用此功能，我们将使用 `ol.Map` 方法将控件添加到地图中 `addControl` 。否则，我们将使用相反的 `ol.Map` 方法从地图中删除控件 `removeControl` 。

我们使用复选框值 `window` 使用数组表示法从对象中选择匹配的OpenLayers控件。控件的变量名（例如 `zoomControl` ）将与复选框值（例如 `zoomControl` ）相同，这是伪造此链接的方式。

注意

所有控件都是的子类 `ol.control.Control` 。这意味着扩展到此类之外的所有控件都将继承 `ol.Object` 方法（例如 `get` 和 `set` ）以及其他函数（例如），这些函数 `getMap` 将通知您此控件附加到的映射。本 `ol.control.Control` 类使创建自定义的控制要容易得多，一个食谱，后来涵盖在这本书。

也可以看看

- 该**管理图的栈层**食谱
- 在**周围的地图视图搬家**食谱

上一节 (/book/web_development/9781785287756/1/ch01lvl1sec12/managing-the-map-s-stack-

下一节 (/book/web_development/9781785287756/1/ch01lvl1sec14/moving-around-the-map-vi