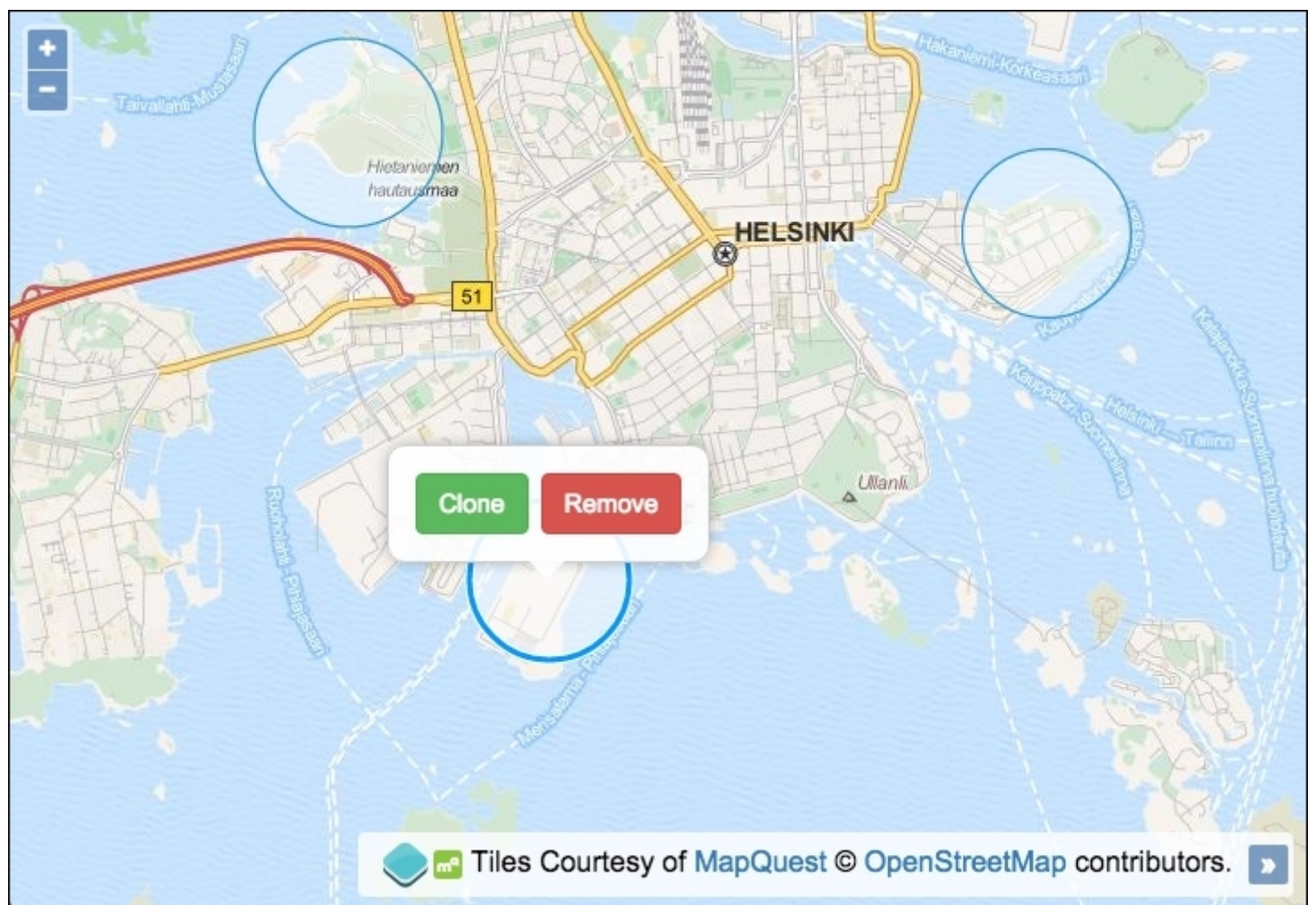


使用叠加层删除或克隆功能

普通的 Web 制图应用程序的特征 是他们显示信息或执行的能力 与功能有关的动作地图。所谓特征，是指我们可以用点，线，多边形等直观表示的任何真实现象或方面。

我们可以选择一项功能，检索其相关信息，然后选择将其显示在应用程序布局中的任何位置。一种常见的方法是使用**叠加层**。

在本食谱中，我们将能够从单击地图上附加的要素时显示的叠加气泡中克隆和删除要素。可以在中找到源代码 `ch03/ch03-removing-cloning-feature-overlay` 。这看起来像以下屏幕截图：



怎么做...

我们将执行一些特征使用以下说明：

- 1 创建具有OpenLayers依赖项的HTML文件，一个 `div` 用于保存地图的元素以及叠加层和内容，如下所示：

[复制](#)

```
<div id="js-map"></div>
<div id="js-overlay">
  <button id="js-clone">Clone</button>
  <button id="js-remove">Remove</button>
</div>
```

- 2 创建一个自定义JavaScript文件，并实例化具有三个圆形特征的新矢量层，如下所示：

[复制](#)

```
var vectorLayer = new ol.layer.Vector({
  source: new ol.source.Vector({
    features: [
      new ol.Feature(new ol.geom.Circle([2780119, 8437147], 900)),
      new ol.Feature(new ol.geom.Circle([2774826, 8433459], 850)),
      new ol.Feature(new ol.geom.Circle([2772686, 8438217], 999))
    ]
  })
});
```

- 3 使用 `view` 和实例化地图实例 `layers`，包括我们的自定义矢量层：

[复制](#)

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 13,
    center: [2775906, 8433717]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.MapQuest({layer: 'osm'})
    }),
    vectorLayer
  ]
});
```

- 4 创建一个 OpenLayers 覆盖从我们的自定义HTML并将其添加到中 `map`。另外，定义一个变量，我们将在整个代码中使用该变量来引用所选功能：

[复制](#)

```
var overlay = new ol.Overlay({
  element: document.getElementById('js-overlay')
});
map.addOverlay(overlay);
var selectedFeature;
```

- 5 创建一个OpenLayers交互以允许选择功能：

[复制](#)

```
var select = new ol.interaction.Select({
  condition: ol.events.condition.click,
  layers: [vectorLayer]
});
map.addInteraction(select);
```

6 订阅 select 互动事件并相应地切换 overlay :

复制

```
select.on('select', function(event) {
  selectedFeature = event.selected[0];
  if (selectedFeature) {
    overlay.setPosition(selectedFeature.getGeometry().getCenter())
  } else {
    overlay.setPosition(undefined);
  }
});
```

7 对于 clone 按钮, 附加 click 事件处理程序并克隆圆:

复制

```
document.getElementById('js-clone')
  .addEventListener('click', function() {
    var circle = selectedFeature.clone();
    var circleGeometry = circle.getGeometry();
    var circleCenter = circleGeometry.getCenter();

    circleGeometry.setCenter([
      circleCenter[0] + circleGeometry.getRadius() * 2,
      circleCenter[1]
    ]);
    vectorLayer.getSource().addFeature(circle);
    overlay.setPosition(undefined);
    select.getFeatures().clear();
  });
```

8 最后, 对于的 remove 按钮, 连接一个 click 事件处理程序 并删除该功能:

复制

```
document.getElementById('js-remove')
  .addEventListener('click', function() {
    vectorLayer.getSource().removeFeature(selectedFeature);
    overlay.setPosition(undefined);
    select.getFeatures().clear();
  });
```

怎么运行的...

我们已经省略了显示样式化叠加层的CSS, 如屏幕截图所示, 因此我们可以专注于OpenLayers代码。完整的代码可以在本书随附的源代码中找到。

地图使用MapQuest的栅格基础层，向量层包含三个大小可变的任意圆形几何体。我们将在本节的其余部分中介绍新引入的概念：

复制

```
var overlay = new ol.Overlay({
  element: document.getElementById('js-overlay')
});
map.addOverlay(overlay);
```

OpenLayers提供了一种机制，以便在指定坐标处显示地图上的叠加层或弹出窗口。这不同于静态位于视图上的控件（例如缩放按钮）。

该 `ol.Overlay` 构造函数用于创建我们自定义的 `overlay` 泡沫。通过传入用于叠加内容的DOM元素，我们提供了最低限度的配置。OpenLayers将此内容包装 `div` 在地图上的绝对位置。最后，使用 `addOverlay` `map`方法将此叠加层分配给地图。

覆盖可以是配置其他属性，例如作为 `autoPan` 和 `position` 。地图实例也包含删除方法 `重叠`，即 `removeOverlay` 。

复制

```
var select = new ol.interaction.Select({
  condition: ol.events.condition.click,
  layers: [vectorLayer]
});
map.addInteraction(select);
```

为了使功能（非）可选，我们使用称为的OpenLayers交互方法 `ol.interaction.Select` 。默认情况下，某些互动已在地图上启用，例如 `ol.interaction.MouseWheelZoom` ，等等。

交互类型需要用于触发事件的条件。我们在 `click` 这里使用的事件类型为 `ol.MapBrowserEvent` 。该 `ol.events.condition` 对象提供许多其他事件，例如 `ol.events.condition.singleClick` 事件，这是 `select` 交互的默认值。我们避免使用默认 `singleClick` 事件，因为它具有250 ms的延迟（以确保不是双击 `click` 事件），而有利于在没有任何延迟的情况下触发的`snapper` 事件。

我们通过 `layers` 属性将图层范围限制为仅包含圆圈的矢量层。我们首先通过 `addInteraction` `map`方法将交互添加到地图中。

使用 `select` 交互的好处是可以自动以不同的方式对选定的要素进行样式设置，以便很明显地选择了它们。您可以通过 `style` 属性来自定义在地图上呈现的样式。

复制



```
select.on('select', function(event) {
  selectedFeature = event.selected[0];
  if (selectedFeature) {
    overlay.setPosition(selectedFeature.getGeometry().getCenter());
  } else {
    overlay.setPosition(undefined);
  }
});
```

尽管已将选择交互添加到地图中，但是如果不先订阅一些已发布的事件，我们就无法对操作做出响应。我们使用该 `on` 方法为事件创建事件处理程序，从而伪造了订阅 `select` 。

发布此事件后，事件对象将在所选数组内包含所选功能（如果有）。如果适用，它还将在取消选择的数组中包含任何以前未选择的功能。我们在 `selectedFeature` 变量中存储第一个选定的功能（因为我们只希望一次为该配方选择一个功能）。

如果功能具有 `选择`，然后将叠加层放置在使用该 `setPosition` 方法的圆几何中心。如果没有选择功能，然后我们确保叠加层不可见。这可以通过传递 `undefined` 给来完成 `setPosition` 。

复制

```
var circle = selectedFeature.clone();
var circleGeometry = circle.getGeometry();
var circleCenter = circleGeometry.getCenter();

circleGeometry.setCenter([
  circleCenter[0] + circleGeometry.getRadius() * 2,
  circleCenter[1]
]);
vectorLayer.getSource().addFeature(circle);
overlay.setPosition(undefined);
select.getFeatures().clear();
```

`clone` 按钮的点击处理程序中发生了很多事情。您会看到我们使用了 `ol.Feature` 实例中可用的许多方法。

我们首先使用该 `clone` 方法克隆选定的圆。使用圆的副本，我们将几何（`getGeometry`）和中心坐标（`getCenter`）缓存到稍后引用的变量中。

通过该 `setCenter` 方法，新的圆几何将居于其他位置。该方法需要一个`type`类型的数组 `ol.Coordinate`。当我们填充此数组时，我们将修改`x`坐标（`circleCenter[0]`），使其与我们克隆的圆相距（原始圆的）整个直径宽度。所述`y`坐标保持不变，导致水平相邻于原始的新循环。

然后将此新圆添加到矢量层源中。为此，我们通过向量层中的 `getSource` 方法访问向量层源，然后链接 `addFeature` 通过圆的向量源方法。一旦将此功能添加到矢量源中，我们就可以通过将位置设置为来关闭叠加层 `undefined` 。



该 `select` 交互临时增加了圈子功能，它提供了“选择”样式的副本。要从地图上删除此地图项，我们使用 `getFeatures` 从 `select` 交互中检索选择的地图项（），这会返回 `ol.Collection of` 的地图项。我们使用 `clear` 方法从集合中删除所有这些功能。

复制

```
vectorLayer.getSource().removeFeature(selectedFeature);
overlay.setPosition(undefined);
select.getFeatures().clear();
```

内少的复杂 `click` 事件处理程序删除按钮，我们可以访问矢量层源（ `getSource` ）并删除当前选定功能（在事件处理程序中存储在 `selectedFeature` 其中 `select` ）。如在克隆处理程序中所见，我们重置了叠加位置以便将其从地图中隐藏，并且还清除了 `select` 交互集合中持有的所有临时要素。

在这里，我们通过介绍地图交互以及它们可以为地图应用程序提供的一些强大优势，介绍了一些重要的知识。随着本书的发展，我们将探索其他地图交互。

也可以看看

🔗 在使用点要素作为标记配方

◀ 上一节 (/book/web_development/9781785287756/3/ch03lvl1sec31/using-point-features-as-ma

下一节 ▶ (/book/web_development/9781785287756/3/ch03lvl1sec33/zooming-to-the-extent-of-a

