

更改缩放效果

平移和缩放效果是与用户导航体验相关的非常重要的操作。在第1章 (/book/web_development/9781785287756/1)，**Web制图基础**，在**周围的地图视图搬家**配方展示了如何可以控制和动画的地图可以平移，缩放，旋转和方式。

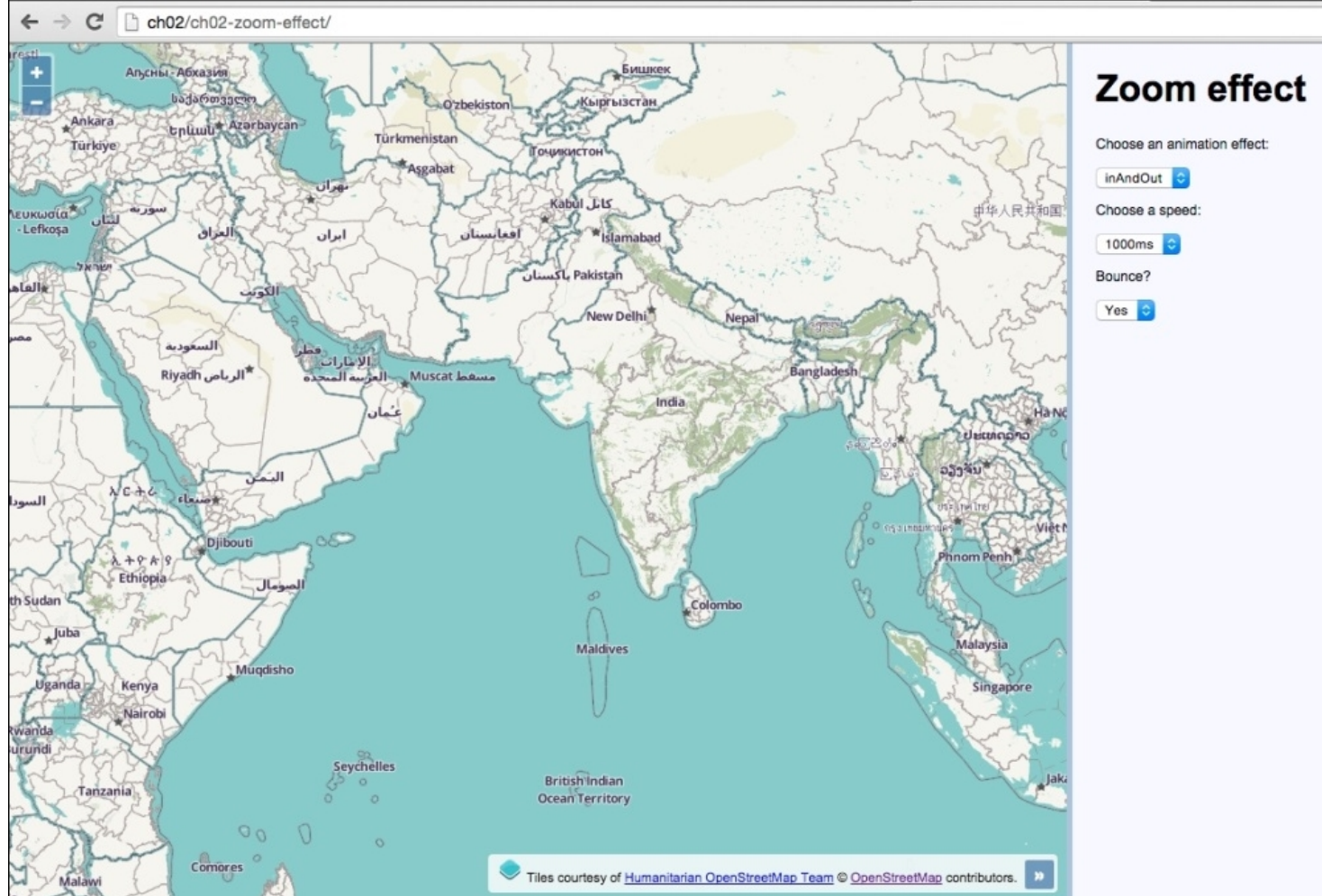
在本食谱中，我们将探讨 通过演示可自定义图层两个缩放级别之间的过渡效果的不同方式来进行进一步制作动画。

默认情况下，OpenLayers为缩放过渡设置动画。为了自己进行自定义，OpenLayers附带了 `ol.animation` 对象可以使用的一系列不同的动画方法。对于本食谱，我们将使用 `ol.animation.bounce` 和 `ol.animation.zoom` 方法来自定义缩放效果。

我们将在右侧提供一个面板，以便您选择动画缓和算法，持续时间以及是否使用反弹效果。

这是我们将要得到的结果的屏幕截图（可在中找到源代码 `ch02/ch02-zoom-effect` ）：





怎么做...

在本节中，我们将看到 更改缩放效果。请遵循以下步骤：

- 1 创建一个HTML文件，并包含必需的OpenLayers依赖项。特别是，创建表单元素以动态更新效果：

复制

```
<p>Choose an animation effect:</p>
<select id="js-zoom-effect">
  <option value="easeIn">easeIn</option>
  <option value="easeOut">easeOut</option>
  <option value="inAndOut" selected>inAndOut</option>
  <option value="linear">linear</option>
</select>

<p>Choose a speed:</p>
<select id="js-zoom-speed">
  <option value="100">100ms</option>
  <option value="250">250ms</option>
  <option value="500">500ms</option>
  <option value="1000" selected>1000ms</option>
</select>

<p>Bounce?</p>
<select id="js-bounce">
  <option value="true">Yes</option>
```

2 创建您的自定义 JavaScript 文件并 map 使用 Humanitarian OpenStreetMap 层构建实例：

[复制](#)

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 5,
    center: [9686000, 1707000]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM({
        attributions: [
          new ol.Attribution({
            html: 'Tiles courtesy of ' +
              '<a href="http://hot.openstreetmap.org">' +
              'Humanitarian OpenStreetMap Team</a>'
          })
        ],
        ol.source.OSM.ATTRIBUTION
      })
    },
    url: 'http://{a-c}.tile.openstreetmap.fr/' +
      'hot/{z}/{x}/{y}.png'
```

3 将表单DOM元素缓存到变量中，如下所示：

[复制](#)

```
var easingSelect = document.getElementById('js-zoom-effect');
var durationSelect = document.getElementById('js-zoom-speed');
var bounceSelect = document.getElementById('js-bounce');
```

4 将事件侦听器添加到地图视图以更改缩放比例，并基于当前表单输入值创建动画效果：

[复制](#)

```
map.getView().on('change:resolution', function(event) {
  if (bounceSelect.value === 'true') {
    map.beforeRender(ol.animation.bounce({
      resolution: map.getView().getResolution() * 1.5,
      duration: parseInt(durationSelect.value, 10)
    }));
  } else {
    map.beforeRender(ol.animation.zoom({
      resolution: event.oldValue,
      duration: parseInt(durationSelect.value, 10),
      easing: ol.easing[easingSelect.value]
    }));
  }
});
```

怎么运行的...

HTML 表单元素提供动画配置的实时反映。每个表单元素都有一个ID属性，我们可以用来在JavaScript中选择和缓存DOM元素，因为我们可能会多次访问这些元素的值。

我们的JavaScript设置了地图和相关属性，包括OpenStreetMap Humanitarian层。这一切对您都应该看起来很熟悉，因此，不用费劲，我们将研究缩放过渡背后的逻辑：

复制

```
map.getView().on('change:resolution', function(event) {
```

我们订阅地图视图上的所有分辨率（缩放）更改，并将匿名函数注册为事件处理程序。

复制

```
if (bounceSelect.value === 'true') {
  map.beforeRender(ol.animation.bounce({
    resolution: map.getView().getResolution() * 1.5,
    duration: parseInt(durationSelect.value, 10)
  }));
}
```

我们进行条件检查，以查看退回选择菜单的值是 `'true'` 还是 `'false'` 。如果是真的，那么我们将注册一个自定义 `ol.animation.bounce` 方法以添加到 `beforeRender` 队列中。我们将 `1.5` 原始分辨率乘以所需的分辨率，以使其具有合适的反射效果。持续时间量是从 `durationSelect` 菜单值得出的。我们通过JavaScript `parseInt` 方法运行持续时间值，以确保它是适用于OpenLayers的类型。持续时间以毫秒为单位。



注意

用类比来描述弹跳效果，是想象网球从某个高度撞击地面。当它撞击表面时，它会跳回去，直到最终失去动力并停在表面上。例如，当您放大地图时，就好像您变成了下降球一样，因此您会弹回最终分辨率。当您更改缓动值（使用的算法）（例如）时 `easeOut` ，它会更改动画帧之间的间距，从而导致事件发生不同的运动。

如我们所见，地图实例具有一个 `beforeRender` 期望使用type函数的方法

`ol.PreRenderFunction` 。这些 `ol.animation` 方法返回此预期类型的函数。此函数类型在呈现视图的最终状态之前对视图执行增量操作，这非常适合动画。

该 `ol.PreRenderFunction` 方法接受两个参数：`map` 实例和框架状态。后者是OpenLayers中的共享对象，代表当前渲染帧状态。它提供信息，例如视图的当前状态（中心位置，分辨率等）和当前时间戳。这些是动画算法（主要由应用缓动技术的数学计算组成）用来执行其视觉效果指标。在函数内，返回true以便在下一帧运行此函数；否则，将返回false进行处理（换句话说，您的动画已完成）。

`ol.animation.bounce` 此函数类型的变体执行增量操作，这些操作会产生缩放请求，并将其“反弹”到位置。

复制

```
} else {
  map.beforeRender(ol.animation.zoom({
    resolution: event.oldValue,
    duration: parseInt(durationSelect.value, 10),
    easing: ol.easing[easingSelect.value]
  }));
}
```

如果禁用了弹跳效果，则将带有 `ol.animation.zoom` 方法的标准缩放动画添加到 `beforeRender` `map` 方法中。

对于动画的起点，将分辨率设置为旧的分辨率值，该值可从事件对象（`event.oldValue`）获得。持续时间再次从持续时间选择菜单中直接拔出，并传递到JavaScript `parseInt` 方法中以进行类型转换。

我们可以自定义用于缩放动画效果的缓动算法。该 `easing` 属性期望适当的缓动功能。OpenLayers带有多重效果供您选择，我们从缓动选择菜单中提供了其中的大多数效果。

例如，OpenLayers的宽松方法是 `ol.easing.easeIn` 。为了从选择菜单中的值动态使用这些缓动功能之一，我们通过数组符号执行对象查找。因此，继续此示例，如果您 `easeIn` 从选择菜单中选择选项，则 `ol.easing[easingSelect.value]` 变为 `ol.easing['easeIn']` 。

还有更多...

尽管OpenLayers提供了一些开箱即用的缓动方法，但您也可以扩展列表并自己提供任何缓动算法。您可能已经使用了其他库中的一些缓动算法，希望将其移植到OpenLayers中。

这些 `ol.animation` 方法都有一个 `start` 属性。这意味着您可以延迟动画的开始。默认开始时间是立即开始。

也可以看看

- 所述**添加WMS层**配方
- 所述**改变不透明度层**配方
- 在**周围的地图视图**搬家食谱

◀ [上一节 \(/book/web_development/9781785287756/2/ch02lvl1sec19/adding-wms-layers\)](/book/web_development/9781785287756/2/ch02lvl1sec19/adding-wms-layers)

[下一节 \(/book/web_development/9781785287756/2/ch02lvl1sec21/changing-layer-opacity\)](/book/web_development/9781785287756/2/ch02lvl1sec21/changing-layer-opacity)

