

简介

*hammer.js*是一款开源的移动端脚本框架，他可以完美的实现在移动端开发的大多数事件，如：点击、滑动、拖动、多点触控等事件。*Hammer.js*不需要依赖任何其他的js框架，并且整个框架非常小，v2.0.4版本只有3.96kb。

*hammer.js*在使用时非常简单，代码示例如下：

```
<script type="text/javascript">
  //创建一个新的hammer对象并且在初始化时指定要处理的dom元素
  var hammertime = new Hammer(document.getElementById("test"));
  //添加事件
  hammertime.on("pan", function (e) {
    document.getElementById("result").innerHTML += "X偏移量：【" + e.deltaX
+ "】，Y偏移量：【" + e.deltaY + "】<br />";
    //控制台输出
    console.log(e);
  });
</script>
```

简介

基础事件

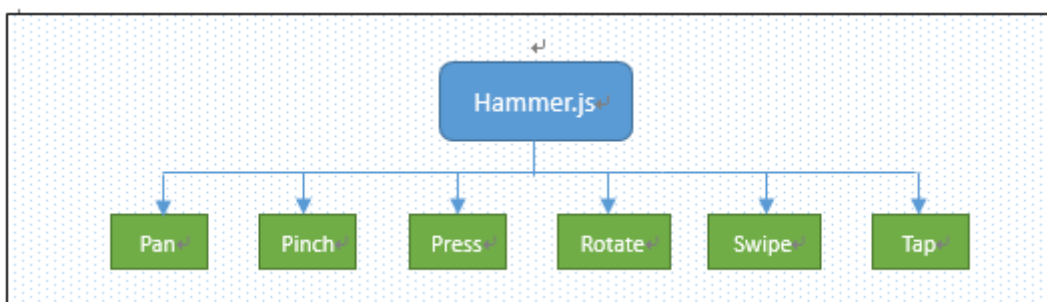
事件对象属性

更高级事件

实例

在触屏中我们使用Click事件不也可以吗？这个Tap事件与Click事件有什么区别呢？

基础事件



1、 Pan事件：在指定的dom区域内，一个手指放下并移动事件，即触屏中的拖动事件。这个事件在屏触开发中比较常用，如：左拖动、右拖动等，如手机上使用QQ时向右滑动出现功能菜单的效果。该事件还可以分别对以下事件进行监听并处理：

Panstart：拖动开始、Panmove：拖动过程、Panend：拖动结束、Pancancel：拖动取消、Panleft：向左拖动、Panright：向右拖动、Panup：向上拖动、Pandown：向下拖动

2、 Pinch事件：在指定的dom区域内，两个手指（默认为两个手指，多指触控需要单独设置）或多个手指相对（越来越近）移动或相向（越来越远）移动时事件。该事件可以分别对以下事件进行监听并处理：

Pinchstart：多点触控开始、Pinchmove：多点触控过程、Pinchend：多点触控结束、Pinchcancel：多点触控取消、Pinchin：多点触控时两手指距离越来越近、Pinchout：多点触控时两手指距离越来越远

3、 Press事件：在指定的dom区域内触屏版本的点击事件，这个事件相当于PC端的Click事件，该不能包含任何的移动，最小按压时间为500毫秒，常用于我们在手机上用的“复制、粘贴”等功能。该事件分别对以下事件进行监听并处理：

Pressup：点击事件离开时触发

4、 Rotate事件：在指定的dom区域内，当两个手指或更多手指成圆型旋转时触发（就像两个手指拧螺丝一样）。该事件分别对以下事件进行监听并处理：

Rotatestart：旋转开始、Rotatemove：旋转过程、Rotateend：旋转结束、Rotatecancel：旋转取消

5、 Swipe事件：在指定的dom区域内，一个手指快速的在触屏上滑动。即我们平时用到最多的滑动事件。

Swipeleft：向左滑动、Swiperight：向右滑动、Swipeup：向上滑动、Swipedown：向下滑动

6、 Tap事件：在指定的dom区域内，一个手指轻拍或点击时触发该事件(类似PC端的click)。该事件最大点击时间为250毫秒，如果超过250毫秒则按Press事件进行处理。

事件对象属性

Name	Value
type	事件的名称，比如 <code>panstart</code> 。
deltaX	X坐标轴上的移动距离
deltaY	Y坐标轴上的移动距离
deltaTime	交互过程的总时长（毫秒）
distance	移动距离
angle	移动角度
velocityX	在X坐标轴上的移动速率，单位为 <code>px/ms</code> 。
velocityY	在Y坐标轴上的移动速率，单位为 <code>px/ms</code> 。
velocity	X/Y上最高的速率值
direction	移动方位，匹配 <code>DIRECTION</code> 常量
offsetDirection	从起始点算起的移动方位。匹配 <code>DIRECTION</code> 常量
scale	多点触摸结束时的缩放比例，若为单点触摸则为1
rotation	多点触摸结束时的旋转数值，若为单点触摸则为0
center	多点触摸的中心位置，或者单点的坐标
srcEvent	源事件对象，类型为 <code>TouchEvent</code> ， <code>MouseEvent</code> 或 <code>PointerEvent</code> 。
target	接收事件的目标
pointerType	基本的指针类型，可以为 <code>touch</code> ， <code>mouse</code> ， <code>pen</code> 或 <code>kinect</code> 。
eventType	事件类型，匹配 <code>INPUT</code> 常量
isFirst	<code>true</code> 表示当前交互为首次交互
isFinal	<code>true</code> 表示当前交互为最后一次交互
pointers	由指针类型组成的数组，包含了最终指针类型 (<code>touchend</code> ， <code>mouseup</code>)。
changedPointers	由 <code>new/moved/lost</code> 指针类型组成的数组
preventDefault	参考 <code>srcEvent.preventDefault()</code> 方法。主要给熟手使用的

更高级事件

上网查资料

实例

看demo

实例中还是有所区别的。在Pinch事件和Rotate事件中，我们用了`hammer.add(new Hammer.Pinch());`和`hammer.add(new Hammer.Rotate());`；而其他四个事件没有用，而是直接添加了事件的监听程序。原因在于，我们在`new Hammer(htmlElement)`的时候，Hammer.js默认对Pan、Press、Swipe和Tap事件进行了监听。但没有对Pinch和Rotate事件进行监听。

在触屏中我们使用Click事件不也可以吗？这个Tap事件与Click事件有什么区别呢？

经过网上查阅后得知，在安卓触屏上，Tap事件和click事件可以同时触发，但click事件会有几百毫秒的延迟，即先触发Tap事件，过一段时间再触发click事件。

所以，移动端还是要用移动端的事件，pc端的事件毕竟不是为移动端设计的。