The touch events

Peter-Paul Koch (ppk)

http://quirksmode.org

http://twitter.com/ppk

WebExpo, 24 September 2010

Hell is other browsers - Sartre
The desktop web

Boring!

- Only five browsers
- with only one viewport each
- that support nearly everything
- Even IE? Yes, even IE.
The Mobile Web

Exciting!

- Fifteen browsers and counting
- ranging from great to lousy
- Interesting new bugs
- About five times as many users as the desktop web (eventually)
- New interaction modes
The Mobile Web

Exciting!

- Fifteen browsers and counting
- ranging from great to lousy
- Interesting new bugs
- About five times as many users as the desktop web (eventually)
- New interaction modes
Mobile browsers

- Android WebKit
- Opera Mobile
- Dolfin (bada)
- Safari
- MicroB
- BlackBerry WebKit
- Symbian WebKit
- IE Mobile
- Palm WebKit
- NetFront
- Bolt
- UCWeb
- Obigo
- Fennec
- Opera Mini

You may groan now.
Mobile browsers

- Android WebKit
- Opera Mobile
- Dolfin (bada)
- Safari
- MicroB
- BlackBerry WebKit
- Symbian WebKit
- IE Mobile
- Palm WebKit

- NetFront
- Bolt
- UCWeb
- Obigo
- Fennec
- Opera Mini

These are all WebKit-based.
There is no WebKit on Mobile.

There's iPhone Safari (3 and 4), and Android (1.6 and 2.1 and 2.2) and Symbian WebKit (1 and 2) and Blackberry WebKit and Palm WebKit and Dolfin for bada

These WebKits are *all different*. 
There is no WebKit on Mobile.

http://quirksmode.org/webkit.html

<table>
<thead>
<tr>
<th>Symbiote 1</th>
<th>Symbiote 2</th>
<th>iPhone 2.2</th>
<th>iPhone 3.1</th>
<th>And 1.0</th>
<th>And 1.6</th>
<th>And 2</th>
<th>Dolphin</th>
<th>BB6</th>
<th>Palm Pre 1.2.1</th>
<th>Bolt 1.5</th>
<th>Ozone 0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>incorrect</td>
<td>yes</td>
<td>untestable</td>
<td>incorrect</td>
<td></td>
<td></td>
<td></td>
<td>yes</td>
<td>untestable</td>
<td>incorrect</td>
<td></td>
</tr>
</tbody>
</table>

need `clientX/Y`, so it's only a matter of time before that becomes the standard.
Mobile web dev

Focus on (in roughly this order)
- iPhone
- Android
- BlackBerry (WebKit and old)
- Symbian WebKit
- Opera Mini
- Opera Mobile
- Dolfin (bada)
Mobile web dev

Focus on (in roughly this order)
- iPhone
- Android
- BlackBerry (WebKit and old)
- Symbian WebKit
- Opera Mini
- Opera Mobile
- Dolfin (bada)
Opera Mobile and Mini

Opera Mobile is a normal browser. Request HTML, CSS, and JavaScript, interpret and execute them.

Opera Mini, though, sends your request to a special server, which requests the HTML, CSS, and JS then interprets it and sends back basically a bitmap
Opera Mini

Opera Mini advantages:
- Little memory necessary, works on old phones, too.
- Less network traffic: you receive a highly compressed bitmap
Opera Mini disadvantage:
- No client-side interactivity. If a click event fires, Opera Mini goes back to the server to ask for instructions.

Tradeoff: lose client-side interactivity, but spend less money on devices and network connections.
Mobile web dev

Focus on (in roughly this order)
- iPhone
- Android
- BlackBerry (WebKit and old)
- Symbian WebKit
- Opera Mini
- Opera Mobile
- Dolfin (bada)
Mobile web dev

Get real devices to test on.
- iPhone or Android
- Nokia S60
- BlackBerry (older)
- Windows Mobile (useful secondary test platform)

Yes, that's expensive.
It's also necessary.
Mobile web dev

Get real devices to test on.

- Unlocked! You should test with multiple networks
- or with wifi
Mobile web dev

Use testing services:

http://www.deviceanywhere.com/
http://perfectomobile.com/

Or use an emulator. Most mobile phone vendors offer one.
Drawback: you can't hold them in your hand, and that's crucial for mobile.
The Mobile Web

Exciting!

- Fifteen browsers and counting
- ranging from great to lousy
- Interesting new bugs
- About five times as many users as the desktop web (eventually)
- New interaction modes
Before we start

please open the following link on your phone:

http://quirksmode.org/touchevents

It gives links to the test files. The touch tests only work on iPhone, Android, bada, and BlackBerry Torch.
Mouse
Mouse

Keyboard
Keyboard users need different interaction than mouse users need different interactions than touch users.

Your script accommodates all three modes, right?

It's all a question of events.
touchstart
touchmove
touchend
touchcancel
It's not an either-or proposition.
It's not an either-or proposition.

The Nokia E71 has a four-way navigation. Works like the arrow keys (including keycodes).

But...
It's not an either-or proposition.

But... the “arrow keys” steer a mouse cursor.

Key events and mouse events
Today we'll concentrate on the touch events, though.
Open the first dropdown example.

Task: Click on option 3.2

This is with traditional mouseover and mouseout; no touch-specific code. Works (a bit oddly, but works).
In theory a touchscreen device should fire only the touch events, and not the mouse events.

A touchscreen device doesn't have a mouse, after all.

However, too many websites depend on the mouse events, so touch browser vendors are forced to support them, too.
Touch !== mouse

- Area
- Pressure
- Temperature
- more than one touch
http://quirksmode.org/touchevents

Now open the second dropdown example.

Task: Click on option 3.2

Doesn't work.
Not an entirely fair comparison.

Touchstart and touchend are not the equivalents of mouseover and mouseout.

In fact, true hovering is impossible on touchscreens. There is no way of saying “I might be interested in this element but I'm not sure yet.”
## Interaction modes

<table>
<thead>
<tr>
<th>Mouse</th>
<th>Keyboard</th>
<th>Touch</th>
</tr>
</thead>
<tbody>
<tr>
<td>mousedown</td>
<td>keydown</td>
<td>touchstart</td>
</tr>
<tr>
<td>mousemove</td>
<td>keypress</td>
<td>touchmove</td>
</tr>
<tr>
<td>mouseup</td>
<td>keyup</td>
<td>touchend</td>
</tr>
<tr>
<td>mouseover</td>
<td>focus</td>
<td>-</td>
</tr>
<tr>
<td>mouseout</td>
<td>blur</td>
<td>-</td>
</tr>
<tr>
<td>All</td>
<td>All</td>
<td>iPhone, Android, bada, BlackBerry</td>
</tr>
</tbody>
</table>
## Interaction modes

<table>
<thead>
<tr>
<th>Mouse</th>
<th>Keyboard</th>
<th>Touch</th>
</tr>
</thead>
<tbody>
<tr>
<td>mousedown</td>
<td>keydown</td>
<td>touchstart</td>
</tr>
<tr>
<td>mousemove</td>
<td>keypress</td>
<td>touchmove</td>
</tr>
<tr>
<td>mouseup</td>
<td>keyup</td>
<td>touchend</td>
</tr>
<tr>
<td>mouseover</td>
<td>focus</td>
<td>-</td>
</tr>
<tr>
<td>mouseout</td>
<td>blur</td>
<td>-</td>
</tr>
<tr>
<td>load, unload,</td>
<td>click, submit,</td>
<td>resize, zoom,</td>
</tr>
<tr>
<td>zoom, change etc.</td>
<td>etc. etc.</td>
<td>etc. etc.</td>
</tr>
</tbody>
</table>
## Interaction modes

<table>
<thead>
<tr>
<th>Mouse</th>
<th>Keyboard</th>
<th>Touch</th>
</tr>
</thead>
<tbody>
<tr>
<td>mousedown</td>
<td>keydown</td>
<td>touchstart</td>
</tr>
<tr>
<td>mousemove</td>
<td>keypress</td>
<td>touchmove</td>
</tr>
<tr>
<td>mouseup</td>
<td>keyup</td>
<td>touchend</td>
</tr>
<tr>
<td>mouseover</td>
<td>focus</td>
<td></td>
</tr>
<tr>
<td>mouseout</td>
<td>blur</td>
<td></td>
</tr>
<tr>
<td>load, unload, click, <strong>submit</strong>, resize, zoom, change etc. etc.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Interaction modes

Mouse
mousedown
mousemove
mouseup
mouseover
mouseout

Keyboard
keydown
keypress
keyup
focus
blur

Touch
touchstart
touchmove
touchend

load, unload, click, submit, resize, zoom, change etc. etc.
# Interaction modes

<table>
<thead>
<tr>
<th>Mouse</th>
<th>Keyboard</th>
<th>Touch</th>
</tr>
</thead>
<tbody>
<tr>
<td>mousedown</td>
<td>keydown</td>
<td>touchstart</td>
</tr>
<tr>
<td>mousemove</td>
<td>keypress</td>
<td>touchmove</td>
</tr>
<tr>
<td>mouseup</td>
<td>keyup</td>
<td>touchend</td>
</tr>
<tr>
<td>mouseover</td>
<td>focus</td>
<td></td>
</tr>
<tr>
<td>mouseout</td>
<td>blur</td>
<td></td>
</tr>
</tbody>
</table>

*load, unload, **click**, submit, resize, zoom, change etc. etc.*
When you touch the screen of a touchscreen, both touch and mouse events fire.

But the mouse events are a bit special. They all fire at the same time.

http://quirksmode.org/touchevents

You can test the events for yourself at the touch action test page.
touchstart
mouseover
mousemove (only one!)
mousedown
mouseup
click
:hover styles applied
Once started, the touch events keep firing regardless of where your finger is.

element.ontouchmove = function () {
  // do stuff
}

This event handler continues firing even if your finger leaves the element.

We need touchenter and touchleave events for when finger enters or leaves element.
touchstart  If a DOM change occurs
mouseover  onmouseover or
mousemove  onmousemove, the rest
mousedown  of the events is cancelled.
mousedown  (iPhone and Symbian)
mouseup

:hover styles applied
When the user touches another element, mouseout :hover styles removed
http://quirksmode.org/touchevents

Try the Event Delegation page. Touch the bordered div.

document.onclick = function () {
  // change divs
}

You click on a div, not on the document. Still, the event bubbles up. Except on the iPhone.
The iPhone does not register a click event if the element you click on doesn't have a click event handler.

document.onclick = function () {
   // change divs
}
div.onclick = function () {}

Now the div itself has a click event handler, and event delegation works fine.
http://quirksmode.org/touchevents

Now open the first drag-and-drop example.

Should work fine; both on touch devices and with a mouse.

This is very simple.
Mouse and touch events:

```javascript
element.ontouchstart = function (e) {
  element.ontouchmove = function (e) {etc}
  element.ontouchend = function (e) {etc}
}

element.onmousedown = function (e) {
  document.onmousemove = function (e) {etc}
  document.onmouseup = function (e) {etc}
}
```
Mouse and touch events:

```javascript
element.ontouchstart = function (e) {
    element.ontouchmove = function (e) {etc}
    element.ontouchend = function (e) {etc}
    element.onmousedown = null;
}
element.onmousedown = function (e) {
    document.onmousemove = function (e) {etc}
    document.onmouseup = function (e) {etc}
}
```

But how do you know whether to use the mouse or the touch events?
Now open the second drag-and-drop example.

iPhone only.
Try dragging two or all three layers simultaneously.
(A bit stilted, but you get the point.)
This is impossible on a desktop computer. Two mice?

Useful for games, maybe (especially on the iPad).

Does not work on other browsers: they don't (yet) support true multitouch.
Now open the scrolling layer example.

Works fine – on mobile.
But how do we port this to the other interaction modes?
- keys: use arrow keys
- mouse: ???
Interaction modes

- mouse
- keyboard
- touch
- and a fourth....
Interaction modes

- mouse
- keyboard
- touch
- trackball

Generally fires a mousemove event
Thank you!
Questions?

http://quirksmode.org
http://twitter.com/ppk

I'll post these slides on my site.