The touch events

Peter-Paul Koch (ppk)
http://quirksmode.org
http://twitter.com/ppk
DIBI, 28 April 2010
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
Before we start

please open the following link on your iPhone or Android:

http://quirksmode.org/touchevents

It gives links to the test files.
Mouse
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.
keydown
keypress
keyup
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.

Key events and mouse events steer a mouse cursor.

But...

The "arrow keys"...
Today we'll concentrate on the touch events, though.
Touch !=== mouse

- Area
- Pressure
- Temperature
- more than one touch
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).
dropdown.onmouseover = function (e) {
    // open dropdown
    dropdown.onmouseout = function (e) {
        // close dropdown
        // if appropriate
        dropdown.onmouseout = null
    }
}
}
http://quirksmode.org/touchevents

Now open the second dropdown example.

Task: Click on option 3.2

Doesn't work.
Not an entirely fair comparison.
Not an entirely fair comparison.

Touchstart and touchend are not the equivalents ofmouseover and mouseout.

Still, there is no better option. Besides, it shows how different touch interaction is from mouse interaction.
## 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</td>
</tr>
</tbody>
</table>
There is no true hover on a touchscreen.

No way of saying “I might be interested in this element but I'm not sure yet.”

Instead, the mobile browsers fake mouseover/out and :hover. (We'll see how later.)
## 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.
# 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.*
## 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, submit, resize, zoom, change etc. etc.</td>
<td></td>
<td></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, change</td>
</tr>
<tr>
<td></td>
<td>submit, resize,</td>
<td>etc. etc.</td>
</tr>
</tbody>
</table>
In theory a touchscreen device should fire only the touch events, and not the mouse events.

However, too many websites depend on the mouse events, so touch browser vendors are forced to support them, too.
Therefore, 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
touchover
touchevent (only one!)
touchedown
touchedup
touchend
:hover styles applied
When the user touches another element mouseout :hover styles removed

On the iPhone this element must listen to events. If it doesn't, it's not clickable and events do not fire.
touchstart
mouseover
mousemove
mousedown
mouseup
:hover styles applied
If a DOM change occurs onmouseover or onmousemove, the rest of the events is cancelled.
(iPhone and Symbian)
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.
element.onmousedown = function (e) {
    // initialise
}

document.onmousemove = function (e) {
    // move
}

document.onmouseup = function (e) {
    document.onmousemove = null;
    document.onmouseup = null;
}

element.onmousedown = function (e) {
  // initialise
  document.onmousemove = function (e) {
    // move
  }
}

Set mousemove and mouseup handlers only when mousedown has taken place. May save some processing time; especially on mobile.
element.onmousedown = function (e) {
    // initialise
    document.onmousemove = function (e) {
        // move
    }
}

document.onmouseup = function (e) {
    document.onmousemove = null;
    document.onmouseup = null;
}

Setmousemove and mouseup handlers on the document.
Helps when user moves too fast and “overshoots”: the script remains functional.
element.onmousedown = function (e) {
  // initialise
  document.onmousemove = function (e) {
    // move
  }
  document.onmouseup = function (e) {
    document.onmousemove = null;
    document.onmouseup = null;
  }
}
element.ontouchstart = function (e) {
  // initialise
  document.ontouchmove = function (e) {
    // move
  };
  document.ontouchend = function (e) {
    document.ontouchmove = null;
    document.ontouchend = null;
  };
}

But: how do we know which events to use? How do we know whether a user uses a mouse or a touchscreen?
element.onmousedown = function (e) {
    document.onmousemove = etc.
    document.onmouseup = etc.
}

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

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

Remove the mousedown event handler when a touchstart takes place: now you're certain that the user uses a touchscreen.
http://quirksmode.org/touchevents

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 Android: the browser doesn't (yet) support true multitouch.
http://quirksmode.org/touchevents

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