The touch events

Peter-Paul Koch
http://quirksmode.org
http://twitter.com/ppk
London JS, 13 February 2014

The touch events

- touchstart: when the user touches the screen
- touchmove: when the user moves his touch
- touchend: when the user ends his touch
- touchcancel: vague

Support: all mobile browsers except for IE and the proxy browsers.
IE does it differently. We'll get back to that.

Examples

- Please open the following page on your phone:
- http://quirksmode.org/touchevents
- It gives links to the test files I'll refer to later
- General touch events test page is the one I used most
- If you have a Windows 8 device, be nice and show your neighbours what's going on.

Stick with

click

Stick with click

- click is not a mouse event
- click really means “activate”
- “I am now going to use this element for its intended purpose”
- Works everywhere.
- Will continue to work everywhere.
- But: slow. About 300ms delay between touch and the following of the link

The slowness of click

What does touching the screen mean?

- “I want to click on this element”
- “I want to scroll”
- “I want to zoom”
- “I want to hold my touch”
- “I want to double-tap”

Thus, a single touchstart event doesn’t give enough clues. The OS needs to wait a little while to figure out what you mean. Hence the delay.

The slowness of click

What does touching the screen mean?

- “I want to click on this element”
- “I want to scroll”
- “I want to zoom”
- “I want to hold my touch”
- “I want to double-tap”

Double-tap resembles a click most: the user’s finger leaves the screen before tapping again.

The slowness of click

The Chrome team is conducting an experiment:

If the page uses width=device-width,

the user doesn't have to double-tap to zoom,

so that interaction is suppressed

and thus the delay is not needed.

Let's see how this experiment turns out.

2 The event cascade

Event cascade

Type 1: Any action but a single tap

- touchstart
- touchmove
- touchend
- possibly an interaction-related event such as scroll or resize

Event cascade

Type 2: Single tap

- touchstart
- touchend
- mouseover
- mousemove (one!)
- mousedown
- mouseup
- click
- :hover styles applied

Event cascade

Type 2: Single tap; ctd.

When tapping another element

- mouseout
- :hover styles removed

Example

- http://quirksmode.org/touchevents
- Dropdown menu 1
- Task: click on option 3.2
- This is with mouse events. No touch events involved.
- Mouseover fires when you touch an element, mouseout when you touch something else.

Event cascade

SAFARI: if a content change occurs onmouseover or onmousemove, the rest of the cascade is cancelled.

- touchstart
- touchend
- mouseover
- mousemove (one!)
- … nothing

Event cascade

What is a content change?

It turns out that Apple means a DOM change

done through actual DOM methods such as appendChild()

innerHTML does NOT count.

Go figure…

3 Separate events?

Separate events

Keyboard	Mouse	Touch
keydown	mousedown	touchstart
(keydown/press)	mousemove	touchmove
keyup	mouseup	touchend
focus	mouseover	-
blur	mouseout	-

onremotewiggle

ondoorclose

onnomorebeer

Does every interaction mode need its own events?

So far the answer has been

YES

Separate events

Keyboard	Mouse	Touch
keydown	mousedown	touchstart
(keydown/press)	mousemove	touchmove
keyup	mouseup	touchend
focus	mouseover	-
blur	mouseout	-

Converging events

Keyboard	Mouse	Touch
keydown	pointerdown	
(keydown/press)	pointermove	
keyup	pointerup	
focus	mouseover	-
blur	mouseout	-

This is Microsoft's idea, and it merits careful consideration.

Example

- http://quirksmode.org/touchevents
- Drag and drop
- Works with mouse events
- Needs minimal changes for touch events
- But bigger ones for pointer events

Drag and drop

```
el.onmousedown = function () {
  el.onmousemove = function () {
    ...
  }
  el.onmouseup = function () {
    el.onmousemove = null;
  }
}
```

Drag and drop

```
el.onmousedown = el.ontouchstart = function () {
  el.onmousemove = el.ontouchmove = function () {
    ...
  }
  el.onmouseup = el.ontouchend = function () {
    el.onmousemove = el.ontouchmove = null;
  }
}
```

Drag and drop

```
el.ontouchstart = function () {
  el.ontouchmove = function () {
    ...
  }
  el.ontouchend = function () {
    el.ontouchmove = null;
  }
}
```

Touch events

```
el.ontouchstart = el.onmspointerdown = function () {
  el.ontouchmove = el.onmspointermove = function () {
    ...
  }
  el.ontouchend = el.onmspointerup = function () {
    el.ontouchmove = el.onmspointermove = null;
  }
}
```

Touch events

```
el.ontouchstart = el.onmspointerdown = function () {
  el.ontouchmove = el.onmspointermove = function () {
    ...
  }
  el.ontouchend = el.onmspointerup = function () {
    el.ontouchmove = el.onmspointermove = null;
  }
}
```

Doesn't work.

Why not?

4 Working with the Microsoft events

Microsoft's approach

It turns out that you have to add one line of CSS:

```
-ms-touch-action: none;
```

Microsoft's approach

It turns out that you have to add one line of CSS:

```
-ms-touch-action: none;
```

And no, I don't much like that.

Microsoft's approach

Still, in practice you're also canceling default actions when you use the touch events.

So Microsoft just made it more explicit.

One problem, though: your script cannot decide whether to handle an event or send it on to the browser. That decision must be taken in the CSS.

Not nice.

Microsoft's approach

These are the -ms-touch-action values. They tell the browser which gestures are allowed. Using an allowed gesture doesn't trigger events.

```
none                - nothing allowed
auto                - everything allowed
pan-x and pan-y     - scrolling x or y
pinch-zoom          - pinch-zooming
double-tap-zoom     - double-tap zooming
manipulation        - everything
                      except double-tap
```

Example

- http://quirksmode.org/touchevents
- -ms-touch-action
- Here are all the values. Play with them.

Microsoft's approach

I'm torn.

Philosophically, Microsoft has a point.

But couldn't they make the actual code nicer?

(Worse example coming up.)

5 Hover

Separate events

Keyboard	Mouse	Touch
keydown	mousedown	touchstart
(keydown/press)	mousemove	touchmove
keyup	mouseup	touchend
focus	mouseover	-
blur	mouseout	-

Example

- http://quirksmode.org/touchevents
- Dropdown menu 2
- Task: click on option 3.2
- Doesn’t work
- Touchstart and touchend are not the equivalents of mouseover and mouseout
- Touch events are discontinuous; while mouse events are continuous

No hover

- There is no hover on touchscreen devices
- No way of saying "I might be interested in this element, but I'm not sure yet."
- Depends on continuous events
- It's technically very hard: device must detect finger above the screen
- But even if it works your finger obscures the screen

No hover

- There is no hover on touchscreen devices
- All our hover-based interactions have become old-fashioned overnight
- Get used to it

6 Event properties

Event info

```
el.onwhatever = function (e) {
   e.type; // the event type
   e.target; // the event target
   e.clientX/Y; // the event coordinates
}
```

Works for the touch events.

Event info

```
el.onwhatever = function (e) {
   e.type; // the event type
   e.target; // the event target
   e.clientX/Y; // the event coordinates
}
```

Works for the touch events.

With one exception.

Event info

- touches array: contains all current touches
- changedTouches array: contains all touches that changed and thus caused an event to fire
- targetTouches array: contains all touches on the target element

Event info

- touches array: contains all current touches
- changedTouches array: contains all touches that changed and thus caused an event to fire
- targetTouches array: contains all touches on the target element

Event info

```
function getCoors(e) {
    var currentTouch = e.changedTouches[0];
    return [currentTouch.clientX,
        currentTouch.clientY];
}
```

Event info

- But not in the Microsoft model
- The touches arrays don't exist.
- Instead, we just read out the coordinates in the old-fashioned way.
- I think this is a good idea in 90% of the cases
- but in the remaining 10% we actually need a list of touch actions taking place, and there isn't any.

Event info

```
function getCoors(e) {
    var currentTouch;
    if (event.changedTouches) {
      currentTouch = e.changedTouches[0];
    } else {
      currentTouch = e;
    }
    return [currentTouch.clientX,
      currentTouch.clientY];
}
```

7 Moving

touchmove

- touchmove continues firing as long as the user's finger is on the screen
- even if it has left the element the event handler is defined on
- MSPointerMove, on the other hand, stops firing when the user's finger leaves the element

Example

- http://quirksmode.org/touchevents
- Traditional and Microsoft move events
- Be sure to move out of the test element.

The traditional events continue firing; the Microsoft ones don't.

Gestures

- The MSPointerMove event doesn't continue firing when you leave the element.
- But the MSGestureChange event does.
- Which brings us to ...

8 Gesture events

Gestures

- A coordinated action of two or more touches constitutes a gesture. Example: pinch-zoom
- Apple (and only Apple) offers the gesturestart, gesturechange, and gestureend events that fire when a gesture takes place.
- Still, I've never used them, and not just because of the lack of support in other browsers. Why do we need them?

Gestures

Microsoft offers a lot of extra events:

- gesturestart, gesturechange, and gestureend
- gesturetap
- gesturehold
- contentzoom
- and more

Gestures

I think I'm in love ...

- gesturestart, gesturechange, and gestureend
- gesturetap
- gesturehold
- contentzoom
- and more

Gestures

... except that they don't work ...

- gesturestart, gesturechange, and gestureend
- gesturetap
- gesturehold
- contentzoom
- and more

Gestures

Well, they do, but … it's complicated

- gesturestart, gesturechange, and gestureend
- gesturetap
- gesturehold
- contentzoom
- and more

MS gesture events

```
el.onmsgesturestart = function () {
  ...
}
```

MS gesture events

```
var MSGesture = new Gesture();
MSGesture.target = el;
```

```
el.onmsgesturestart = function () {
  ...
}
```

MS gesture events

```
var MSGesture = new Gesture();
MSGesture.target = el;
el.onmspointerdown = function (e) {
  MSGesture.addPointer(e.pointerId);
}
el.onmsgesturestart = function () {
  ...
}
```

MS gesture events

```
var MSGesture = new Gesture();
MSGesture.target = el;
el.onmspointerdown = function (e) {
  MSGesture.addPointer(e.pointerId);
  -ms-touch-action: none;
}
el.onmsgesturestart = function () {
  ...
}
```

And don’t forget our old friend:

Gestures

... and even then contentzoom doesn't work

- gesturestart, gesturechange, and gestureend
- gesturetap
- gesturehold
- ~~contentzoom~~
- and more

Example

- http://quirksmode.org/touchevents
- Apple and MS Gestures
- Be sure to move out of the test element.
  MSGestureChange continues firing.

9 Question time

Thank you

I'm going to put these slides online.

Questions?

Peter-Paul Koch
http://quirksmode.org
http://twitter.com/ppk
London JS, 13 February 2014