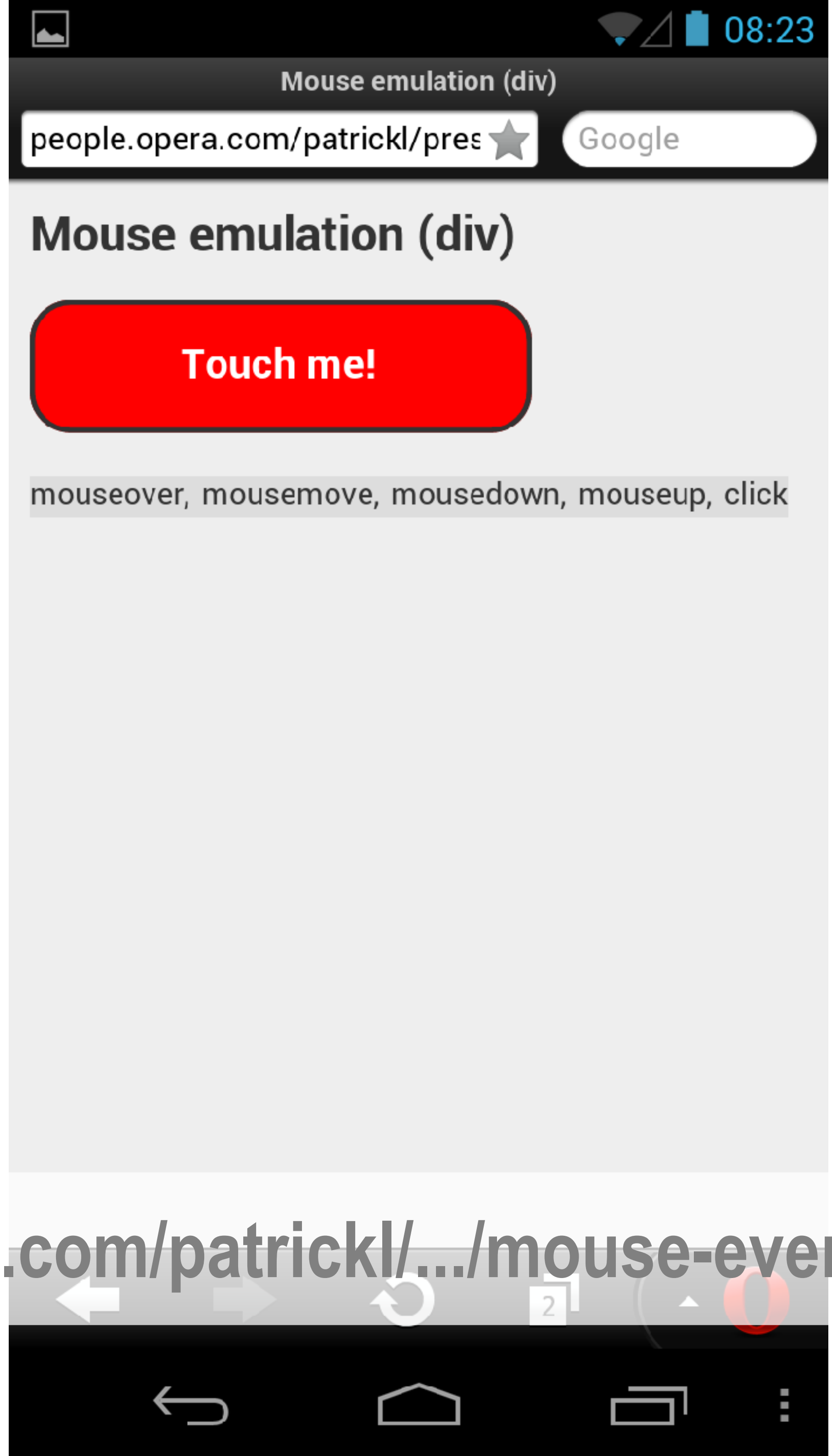
getting touchy

AN INTRODUCTION TO TOUCH EVENTS

Patrick H. Lauke / Web Standards Days / Москва/ 24 November 2012 you **don't** <u>need</u> **touch events**

browsers emulate regular mouse events



Mouse emulation (div)

mouseover, mousemove, mousedown, mouseup, click

people.opera.com/patrickl/.../mouse-event-simulation

`mouseover > mousemove > mousedown > (focus) > mouseup > click`

on first tap

`mouseover > mousemove > mousedown > (focus) > mouseup > click`

subsequent taps

`mousemove > mousedown > (focus) > mouseup > click`

tapping away

`mouseout > (blur)`

`focus/blur` only on focusable elements in **Opera Mobile** and **Firefox**
`mouseout` not on **iOS Safari** and embedded WebView (e.g. **iOS Chrome**)

emulation works but is
limiting/problematic

(more on that in a minute)

touch events

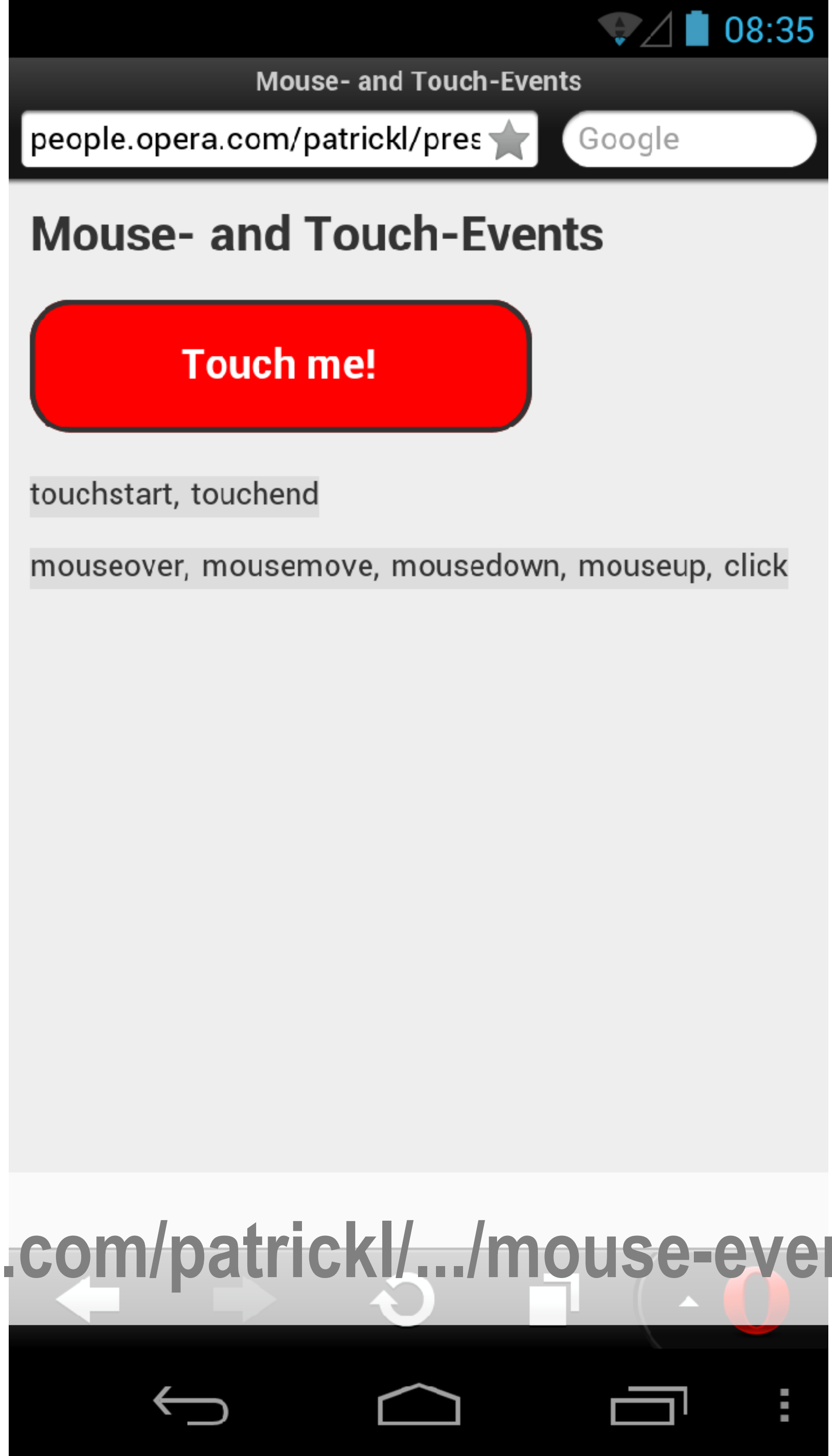
www.w3.org/TR/touch-events

touchstart

touchmove

touchend

touchcancel



Mouse- and Touch-Events

touchstart, touchend

mouseover, mousemove, mousedown, mouseup, click

people.opera.com/patrickl/.../mouse-event-simulation `touchstart > [touchmove]+ > touchend > mouseover > mousemove > mousedown > (focus) > mouseup > click`

on first tap

`touchstart > [touchmove]+ > touchend > mouseover > mousemove > mousedown > (focus) > mouseup > click`

subsequent taps

`touchstart > [touchmove]+ > touchend > mousemove > mousedown > (focus) > mouseup > click`

tapping away

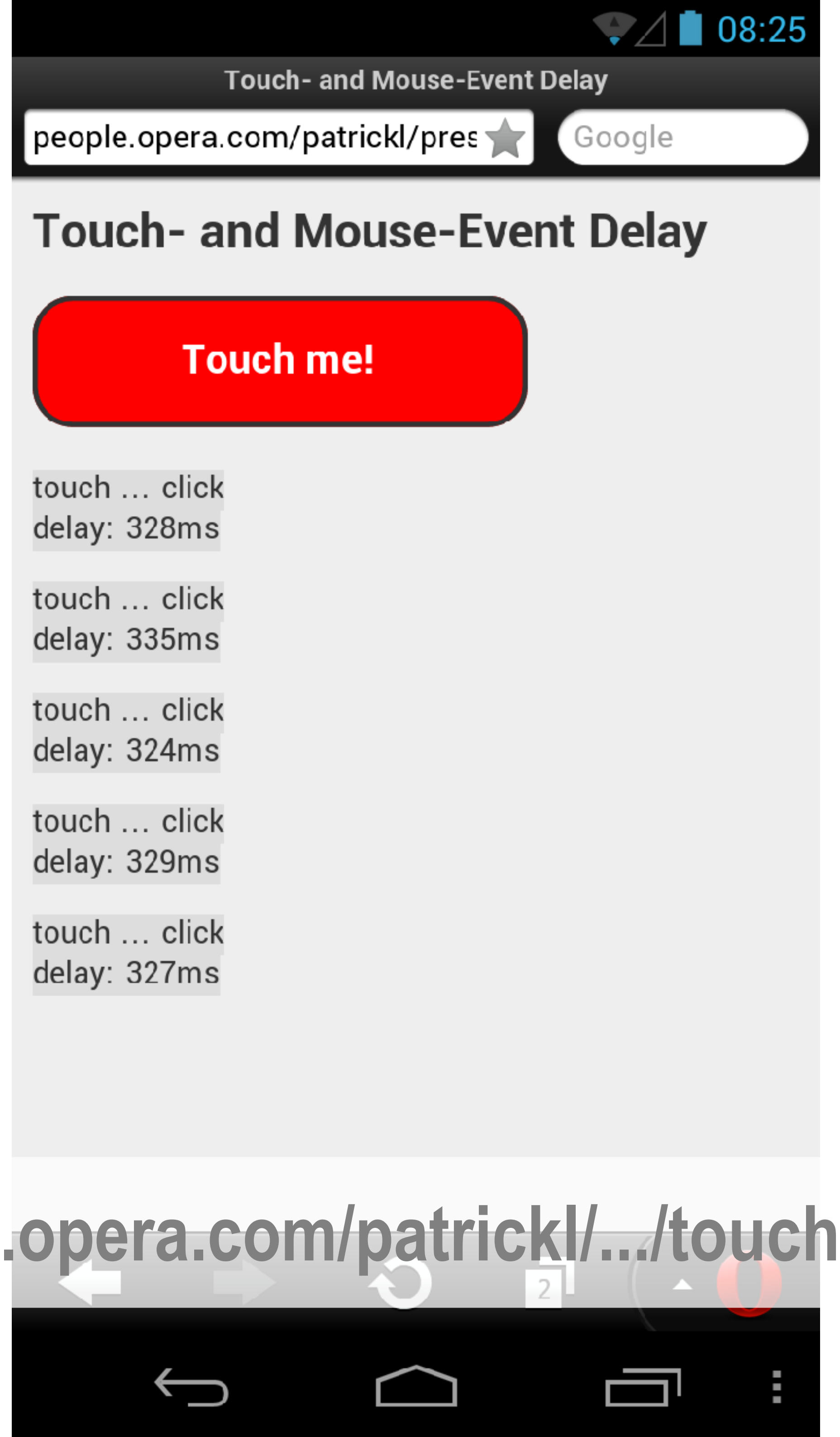
`mouseout > (blur)`

too many **`touchmove`** events abort the tap (after **`touchend`**)
not all browsers consistently send the **`touchmove`**

limitations/problems of mouse event emulation

1. delayed event dispatch
2. mouse-specific interfaces
3. mousemove doesn't track

1. **delayed event dispatch**
2. **mouse-specific interfaces**
3. **mousemove doesn't track**



Touch- and Mouse-Event Delay

touch ... click delay: 328ms

touch ... click delay: 335ms

touch ... click delay: 324ms

touch ... click delay: 329ms

touch ... click delay: 327ms

people.opera.com/patrickl/.../touch-delay

simple **feature detection** for touch events

```
if ('ontouchstart' in window) {
  /* some clever stuff here */
}
```

```
/* common performance “trick” */
var event = 'click';
if ('ontouchstart' in window) {
    event = 'touchstart';
}
foo.addEventListener(event,function(e)
{
    /* execute on click or touch */
}, false);
```

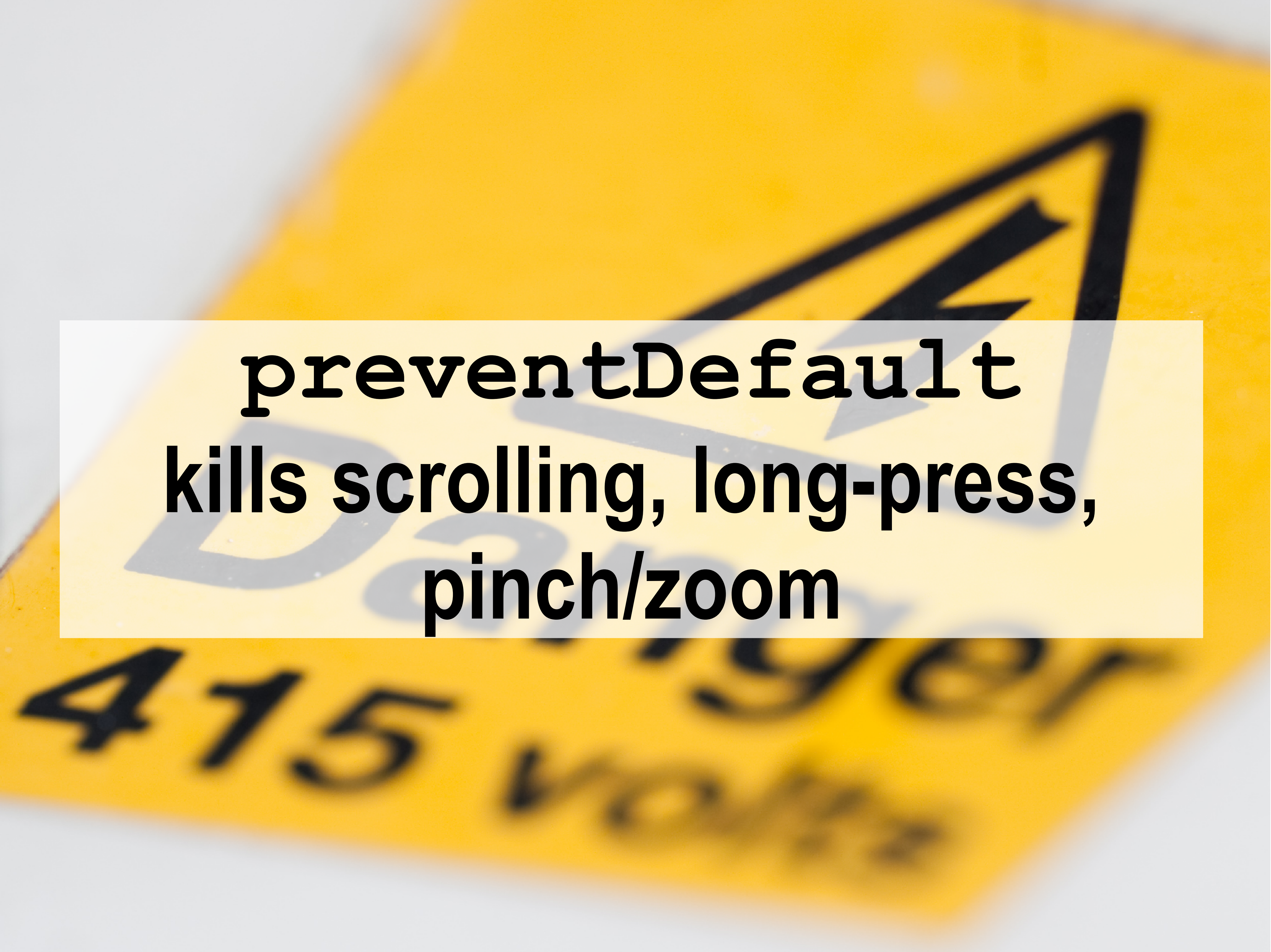
don't make it touch-exclusive

hybrid devices

touch + mouse + keyboard

```
/* doubled-up event listeners */

foo.addEventListener('click',
    function(e) {…}, false);

foo.addEventListener('touchstart',
    function(e) {…}, false);
```

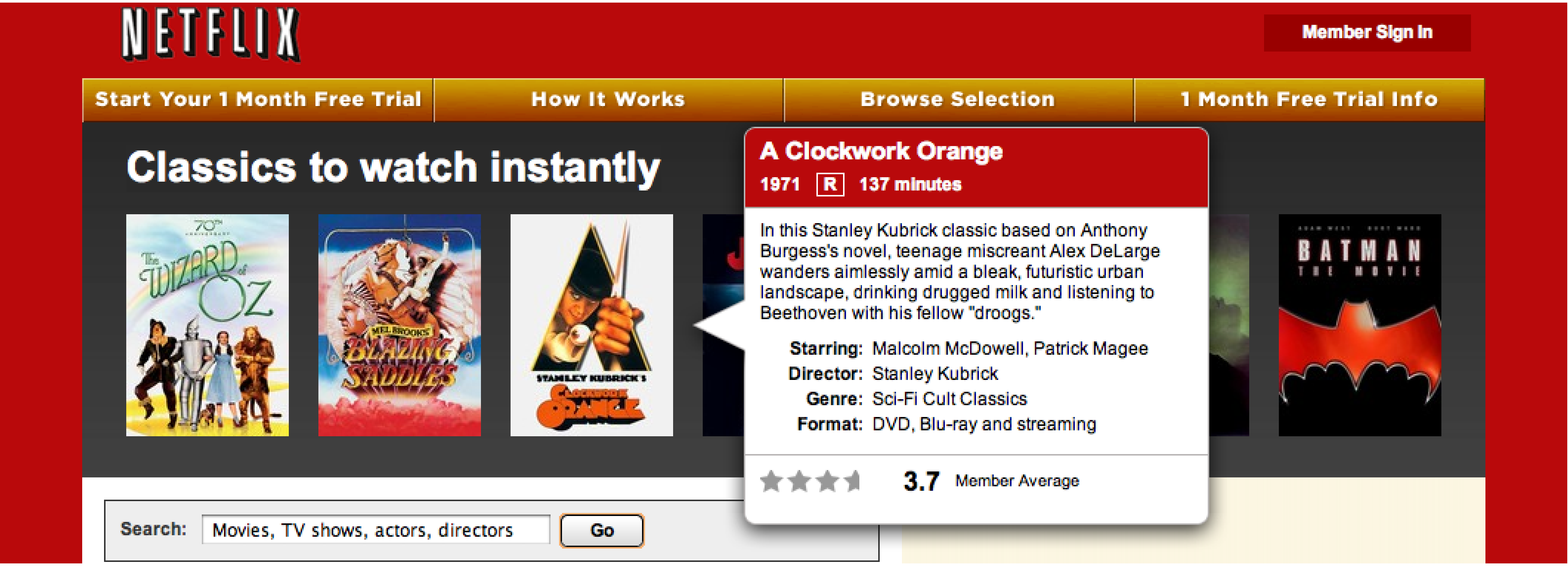
```
/* doubled-up event listeners */
foo.addEventListener('click',
    function(e) {…},false);
foo.addEventListener('touchstart',
    function(e) {

      …

      /* stop mouse event emulation */

      e.preventDefault();

},false);
```



`preventDefault`

kills scrolling, long-press, pinch/zoom

1. delayed event dispatch
2. mouse-specific interfaces
3. mousemove doesn't track



Classics to watch instantly

A Clockwork Orange

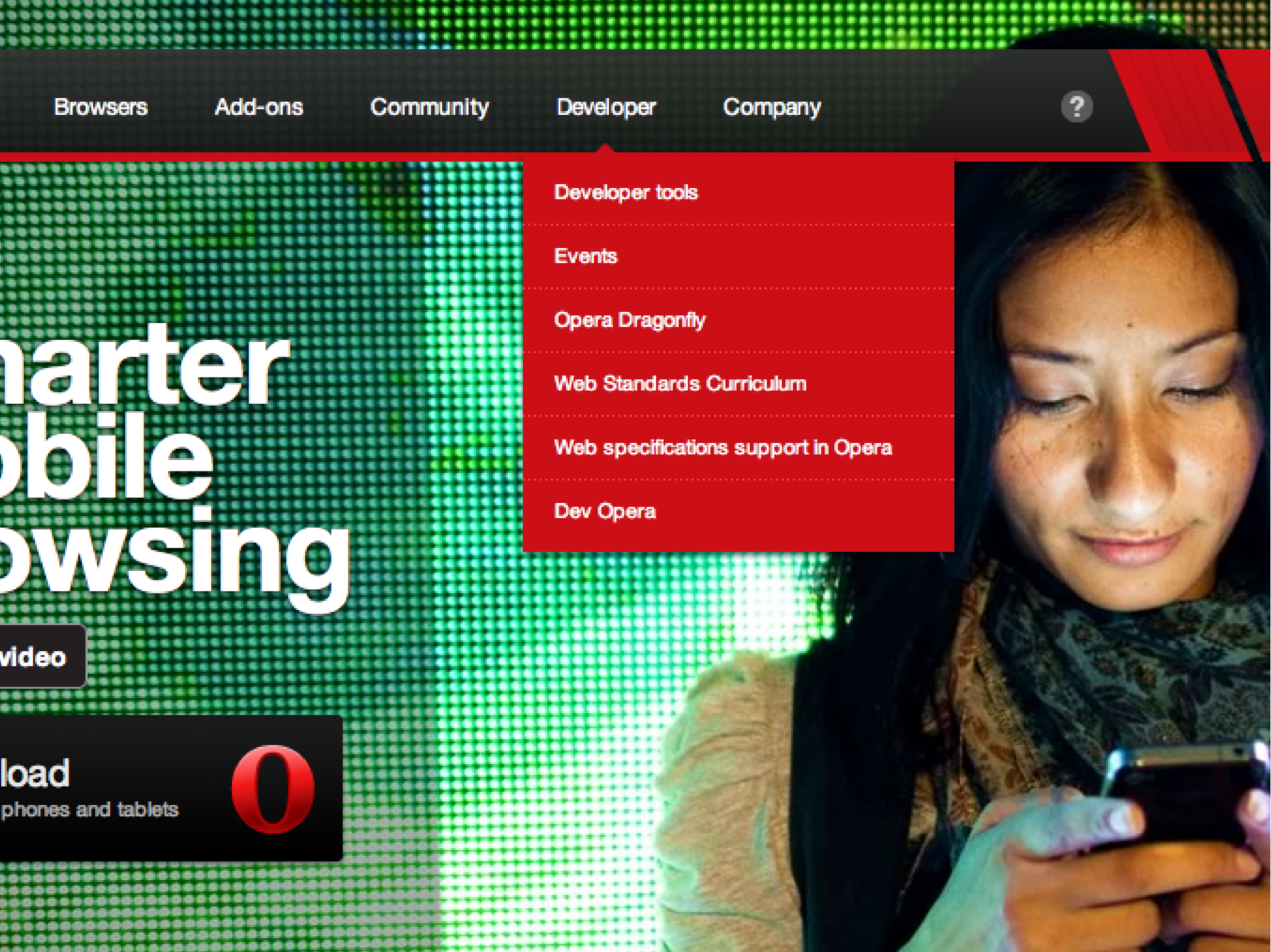
1971 R 137 minutes

In this Stanley Kubrick classic based on Anthony Burgess's novel, teenage miscreant Alex DeLarge wanders aimlessly amid a bleak, futuristic urban landscape, drinking drugged milk and listening to Beethoven with his fellow "droogs."

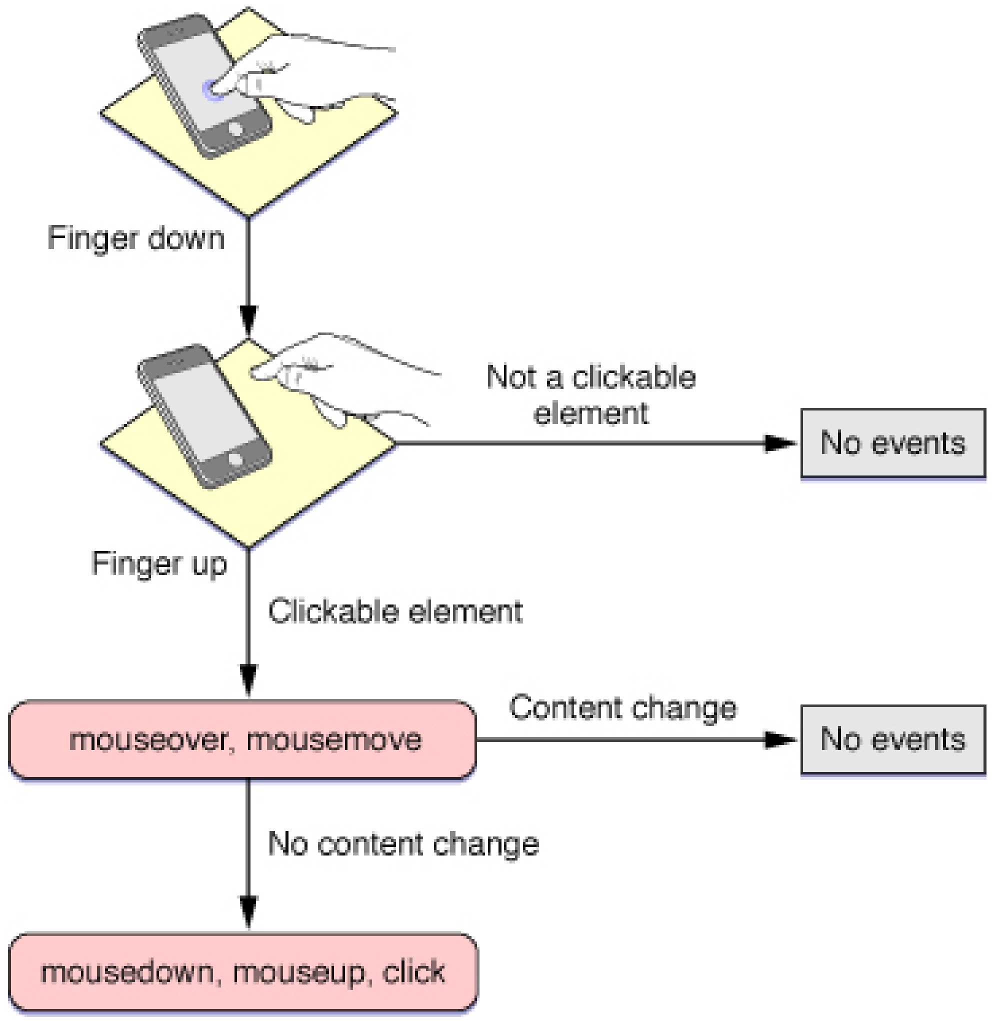
Starring:	Malcolm McDowell, Patrick Magee
Director:	Stanley Kubrick
Genre:	Sci-Fi Cult Classics
Format:	DVD, Blu-ray and streaming

3.7 Member Average

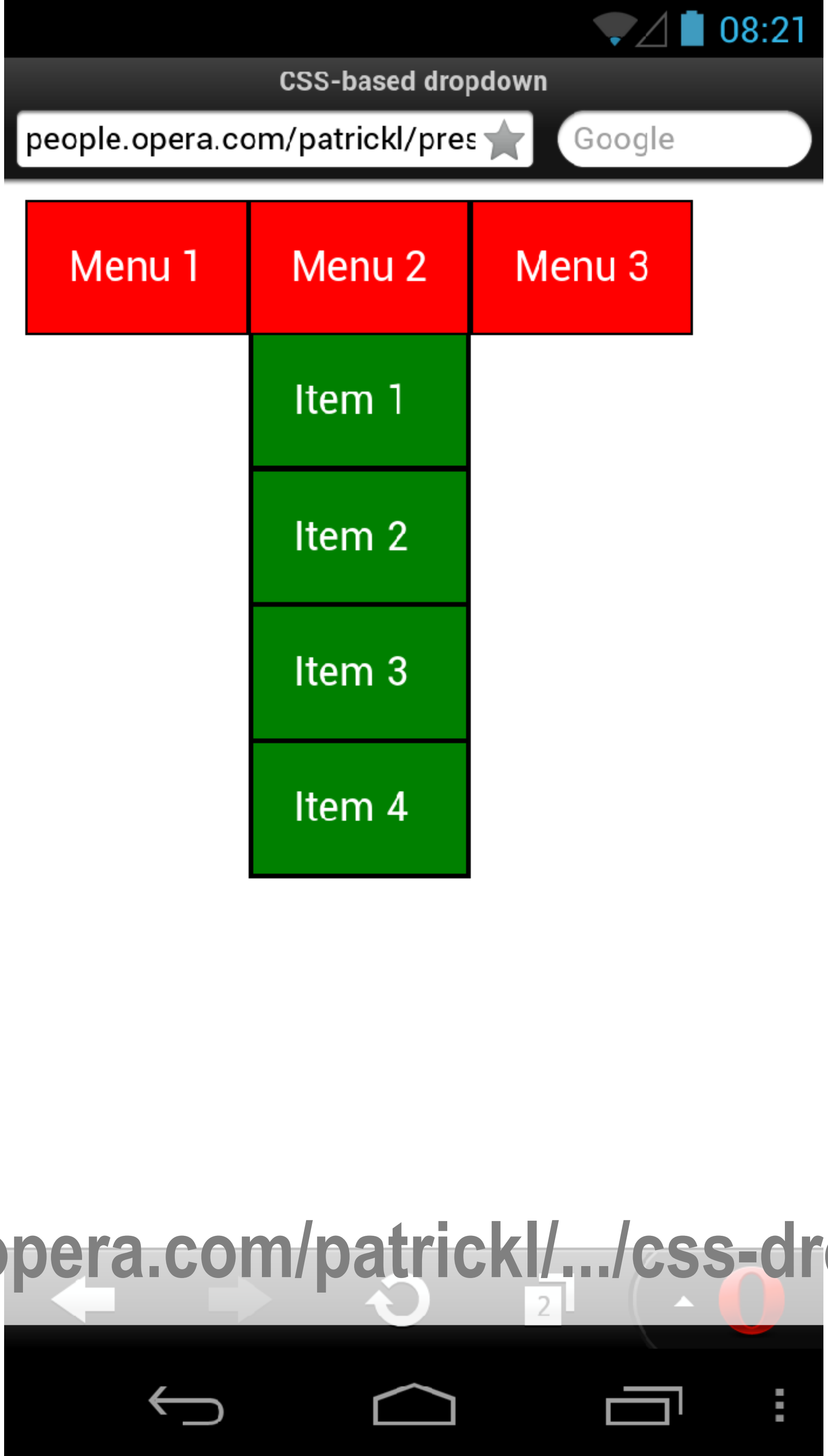
Search: Movies, TV shows, actors, directors Go


Browsers
Add-ons
Community
Developer
Company
?
Developer tools
Events
Opera Dragonfly
Web Standards Curriculum
Web specifications support in Opera
Dev Opera
narter
obile
owsing
video
load
phones and tablets

no isolated hover (or focus) on touch devices



http://developer.apple.com/library/IOS/...



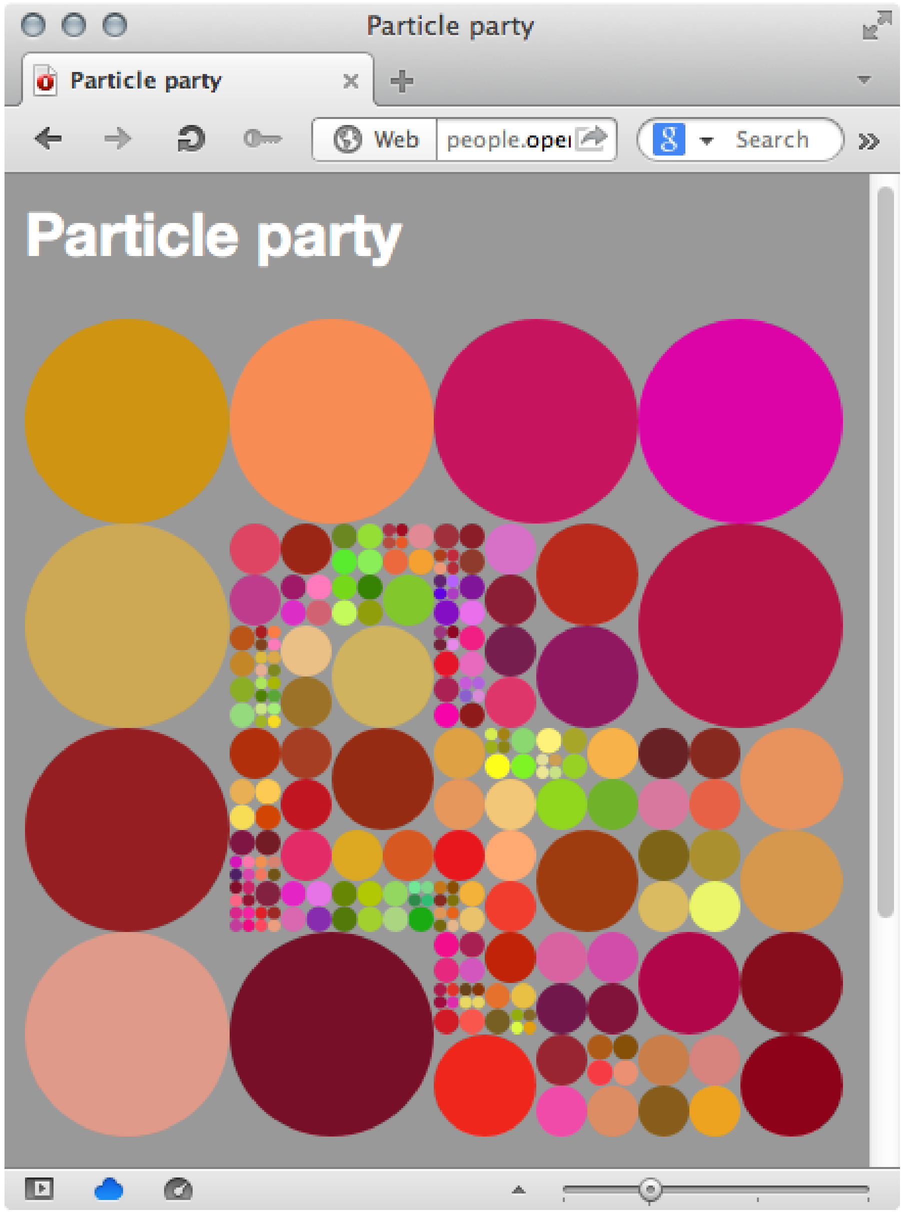
people.opera.com/patrickl/.../css-dropdown

```
/* pure CSS dropdown */
ul.menu li:hover ul { display: block; }


<ul class="menu">
  <li><a href="">Menu 1</a></li>
  <li class="submenu"><a href="">Menu 2</a>
    <ul>
      ...

    </ul>
  </li>

  ...

</ul>
```

```
/* CSS and JS dropdown */

ul.menu li:hover ul,
ul.menu li.open ul { display: block; }

/* make it keyboard accessible */

document.querySelectorAll('ul.menu li.submenu > a')
.addEventListener('focus', function(e) {

   ...

   this.parentNode.classList.add('open');

},false);
```

```
/* CSS and JS dropdown */

ul.menu li:hover ul,
ul.menu li.open ul { display: block; }

/* touch opens menu */

document.querySelectorAll('ul.menu li.submenu > a')
.addEventListener('touchstart', function(e) {
    ...
    if (!this.parentNode.classList.contains('open')) {
          ...
        this.parentNode.classList.add('open');
        e.preventDefault();
    }

},false);
```

1. delayed event dispatch
2. mouse-specific interfaces
3. mousemove doesn't track



people.opera.com/patrickl/experiments/canvas/particle/2

```
var posX, posY;
...
function positionHandler(e) {
  posX = e.clientX;
  posY = e.clientY;
}
...
canvas.addEventListener('mousemove',
    positionHandler, false);
```

```
interface MouseEvent : UIEvent {
   readonly attribute long                  screenX;
   readonly attribute long                  screenY;
   readonly attribute long                  clientX;
   readonly attribute long                  clientY;
   readonly attribute boolean               ctrlKey;
   readonly attribute boolean               shiftKey;
   readonly attribute boolean               altKey;
   readonly attribute boolean               metaKey;
   readonly attribute unsigned short        button;
   readonly attribute EventTarget           relatedTarget;
   void                   initMouseEvent(...);
};
```

www.w3.org/TR/DOM-Level-2-Events ...

```
var posX, posY;

...

function positionHandler(e) {
  /* handle both mouse and touch? */
}

...

canvas.addEventListener('mousemove',
  positionHandler, false);

canvas.addEventListener('touchmove',
  positionHandler, false);
```

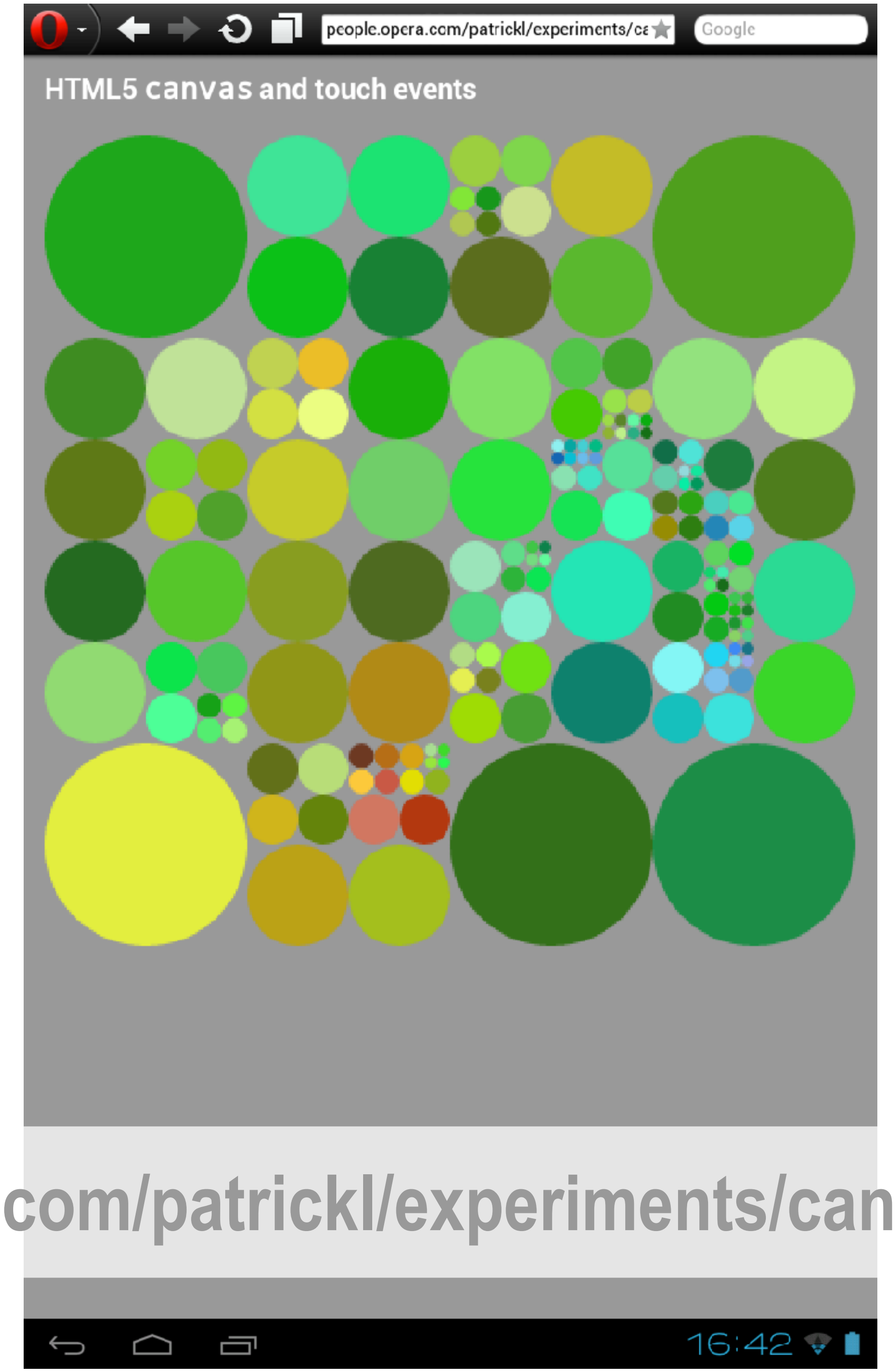
```
interface TouchEvent : UIEvent {
   readonly attribute TouchList touches;
   readonly attribute TouchList targetTouches;
   readonly attribute TouchList changedTouches;
   readonly attribute boolean   altKey;
   readonly attribute boolean   metaKey;
   readonly attribute boolean   ctrlKey;
   readonly attribute boolean   shiftKey;
};
```

www.w3.org/TR/touch-events

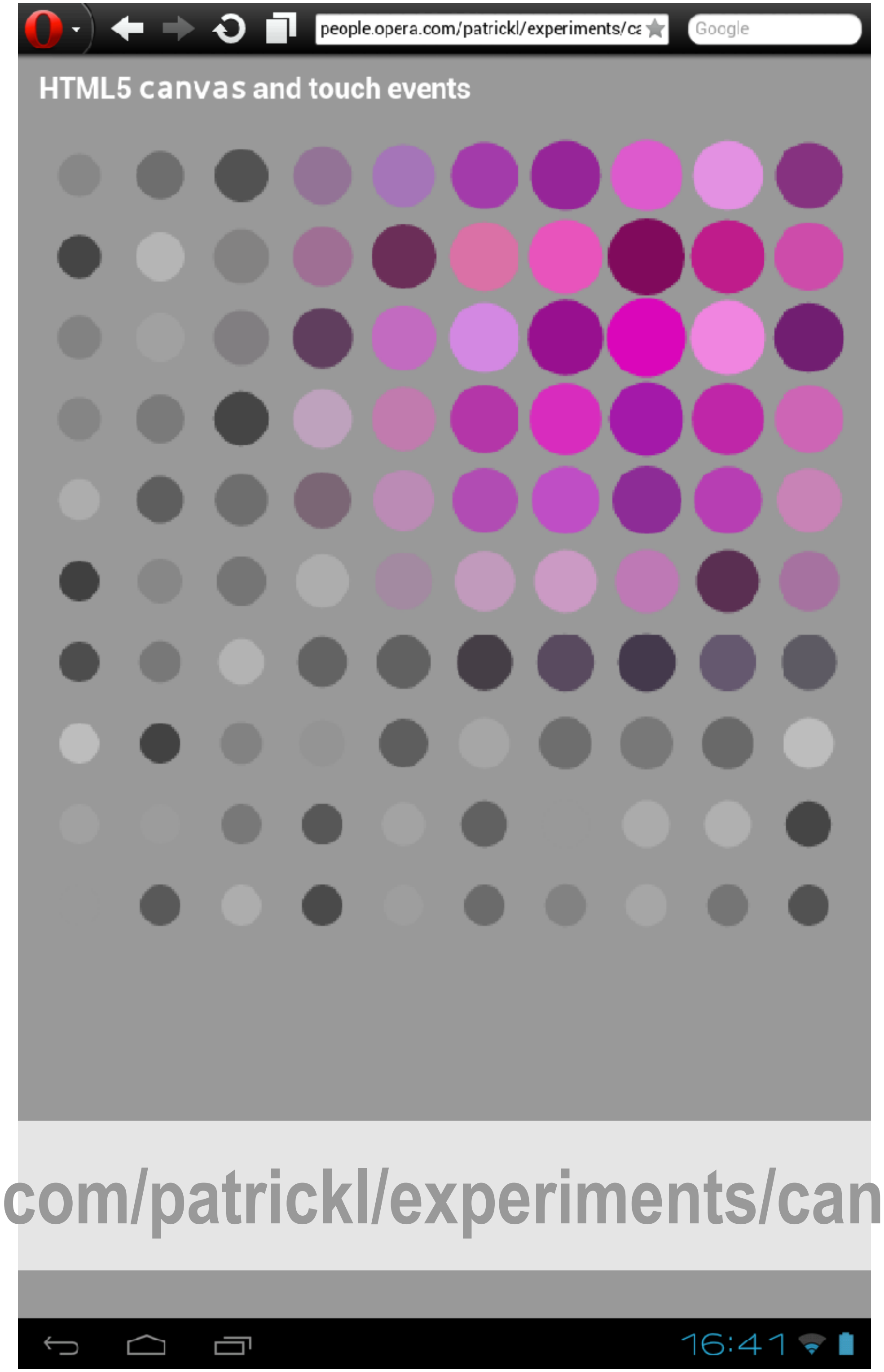
```
interface Touch {
   readonly attribute long        identifier;
   readonly attribute EventTarget target;
   readonly attribute long        screenX;
   readonly attribute long        screenY;
   readonly attribute long        clientX;
   readonly attribute long        clientY;
   readonly attribute long        pageX;
   readonly attribute long        pageY;
};
```

www.w3.org/TR/touch-events

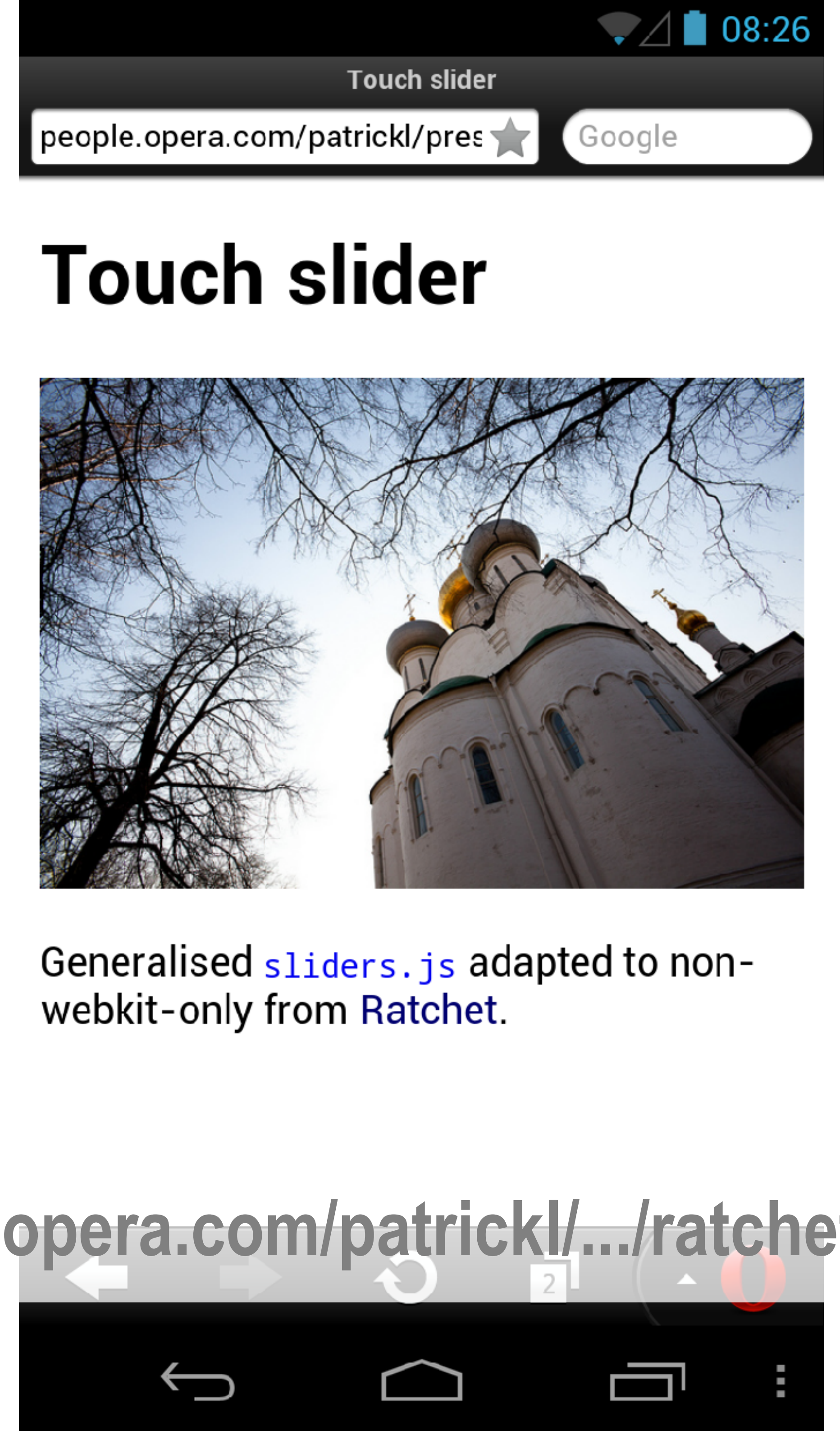
```
var posX, posY;

...

function positionHandler(e) {
   if ((e.clientX)&&(e.clientY)) {
      posX = e.clientX;
      posY = e.clientY;
   } else if (e.targetTouches) {
      posX = e.targetTouches[0].clientX;
      posY = e.targetTouches[0].clientY;
      e.preventDefault();
   }
}

...

canvas.addEventListener('mousemove',
   positionHandler, false );

canvas.addEventListener('touchmove',
   positionHandler, false );
```


people.opera.com/patrickl/experiments/ca
Google
HTML5 canvas and touch events
16:42

people.opera.com/patrickl/experiments/canvas/particle/3


people.opera.com/patrickl/experiments/ca
Google
HTML5 canvas and touch events
16:41

people.opera.com/patrickl/experiments/canvas/particle/4

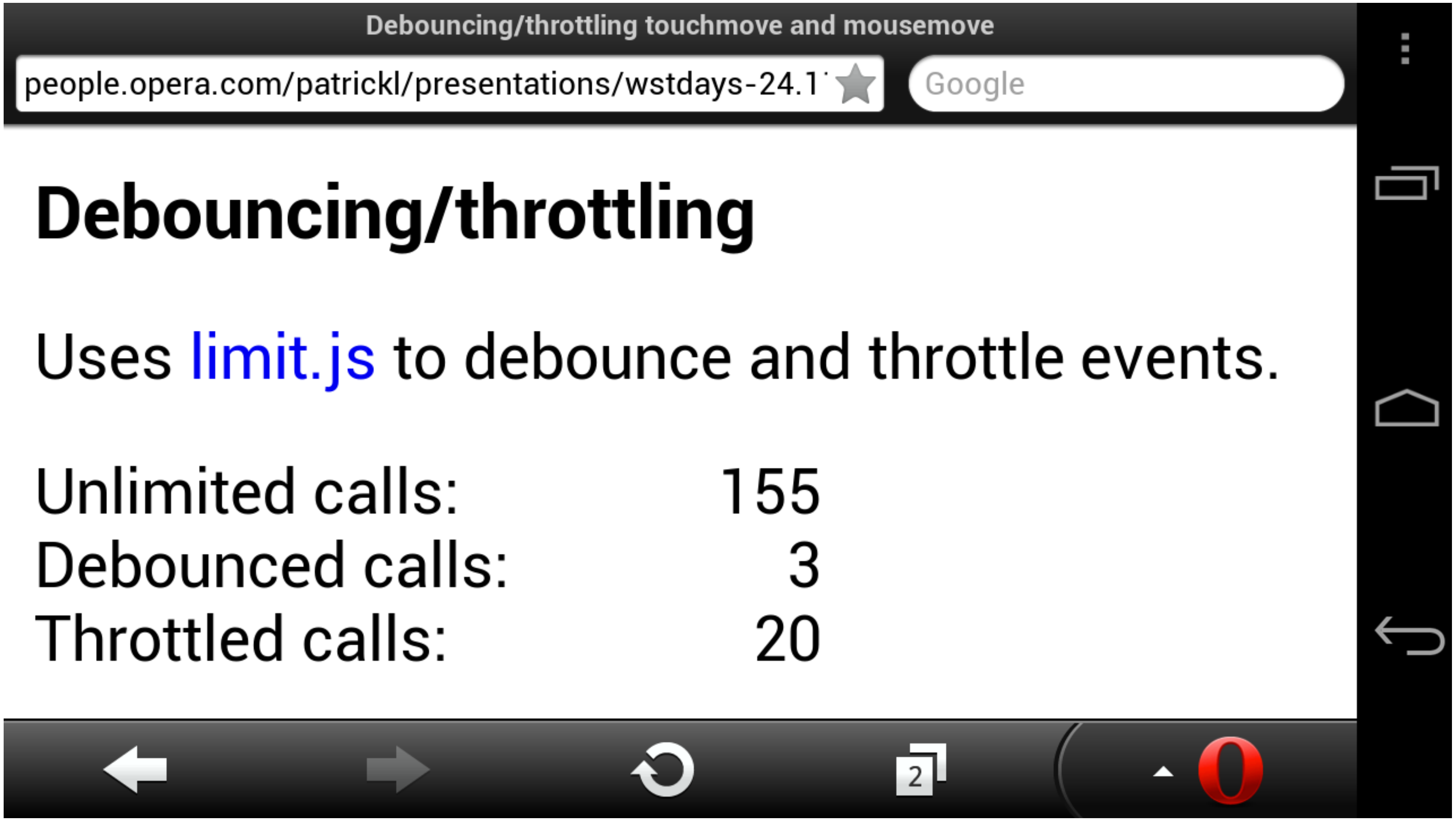


Touch slider

Generalised `sliders.js` adapted to non-webkit-only from Ratchet.

people.opera.com/patrickl/.../ratchet-slider

touchmove fires...a lot!


Debouncing/throttling touchmove and mousemove

people.opera.com/patrickl/presentations/wstdays-24.1

Debouncing/throttling

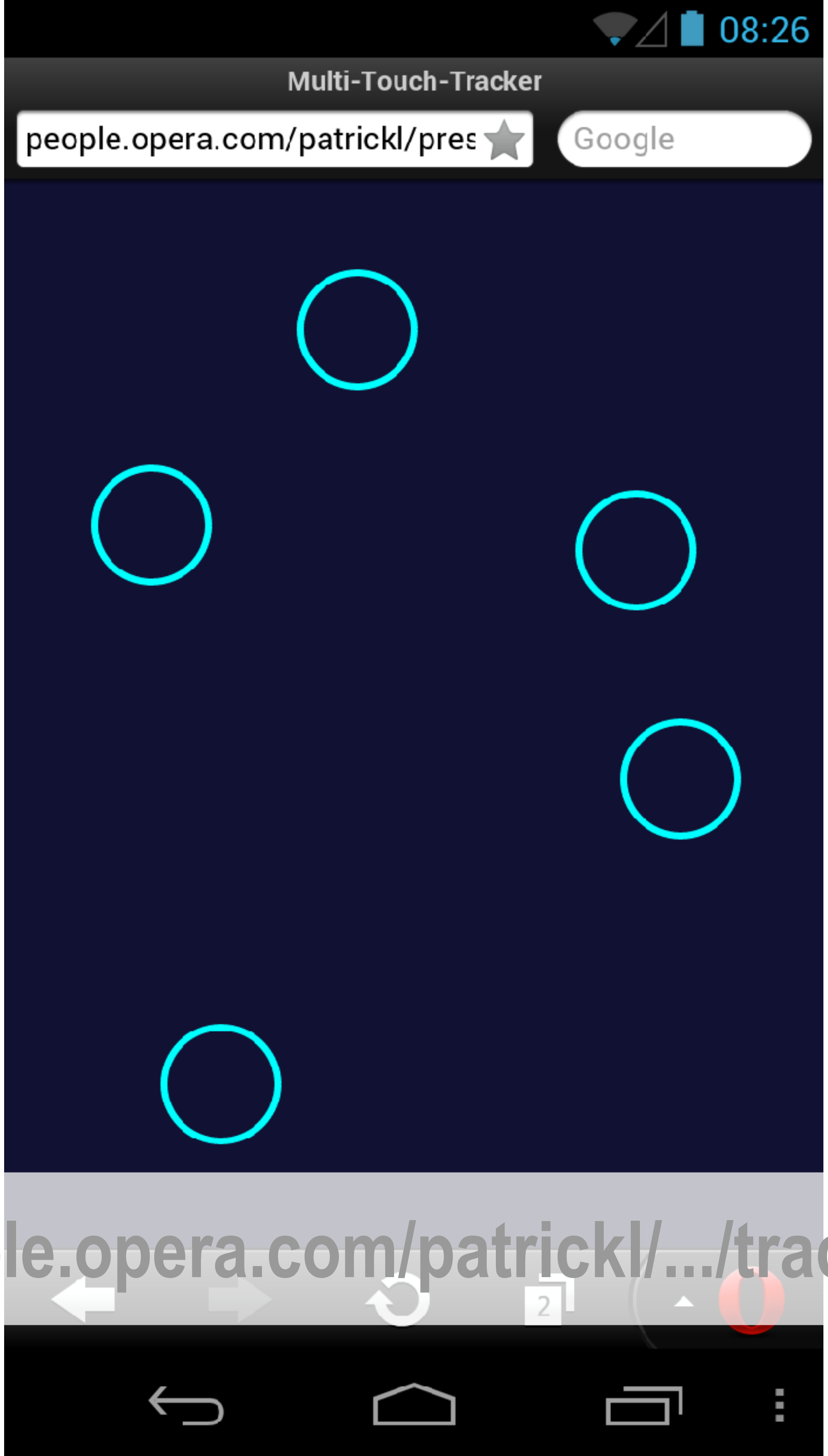
Uses limit.js to debounce and throttle events.

Unlimited calls:	155
Debounced calls:	3
Throttled calls:	20

people.opera.com/patrickl/.../touch-limit why stop at a single point?

multitouch support

```
for (i=0; i<e.targetTouches.length; i++) {
   ...
   posX = e.targetTouches[i].clientX;
   posY = e.targetTouches[i].clientY;
   /* do something clever */
}
```

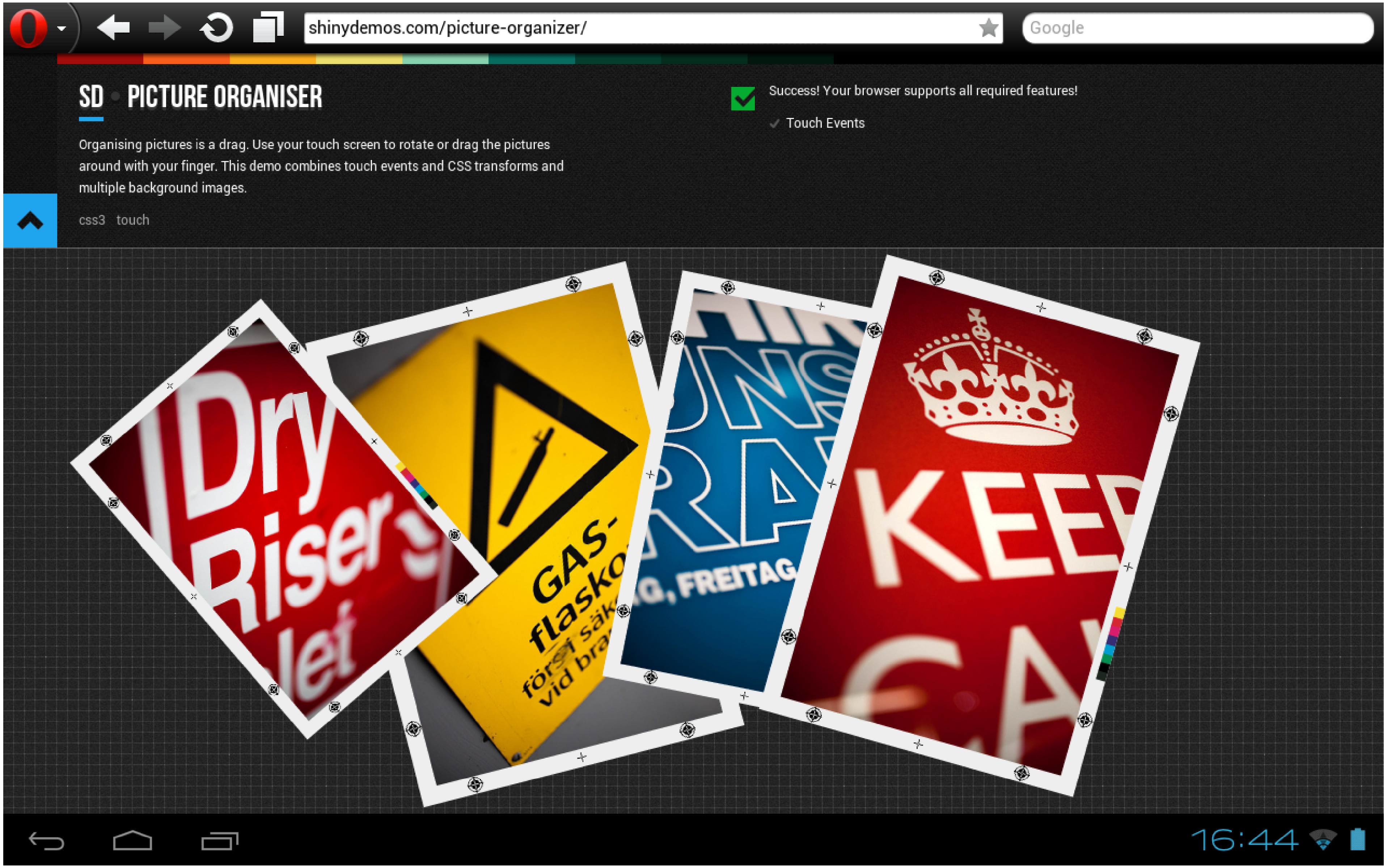


people.opera.com/patrickl/.../tracker/

multitouch gestures

```
/* iOS Safari has gesture events for size/rotation
   with some trigonometry we can make this x-browser */
var distance = Math.sqrt(Math.pow(...)+Math.pow(...));
var angle = Math.atan2(...);
```

people.opera.com/patrickl/.../pinch-zoom-rotate



Organising pictures is a drag. Use your touch screen to rotate or drag the pictures around with your finger. This demo combines touch events and CSS transforms and multiple background images.

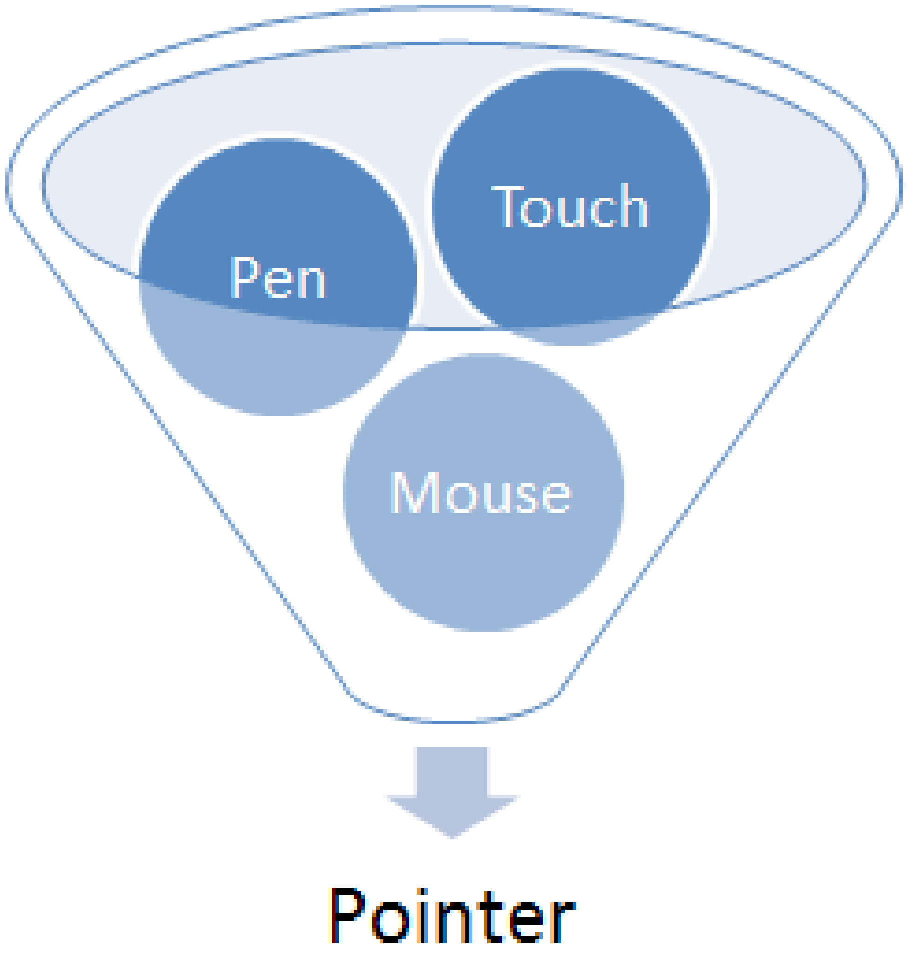
css3 touch

Success! Your browser supports all required features!

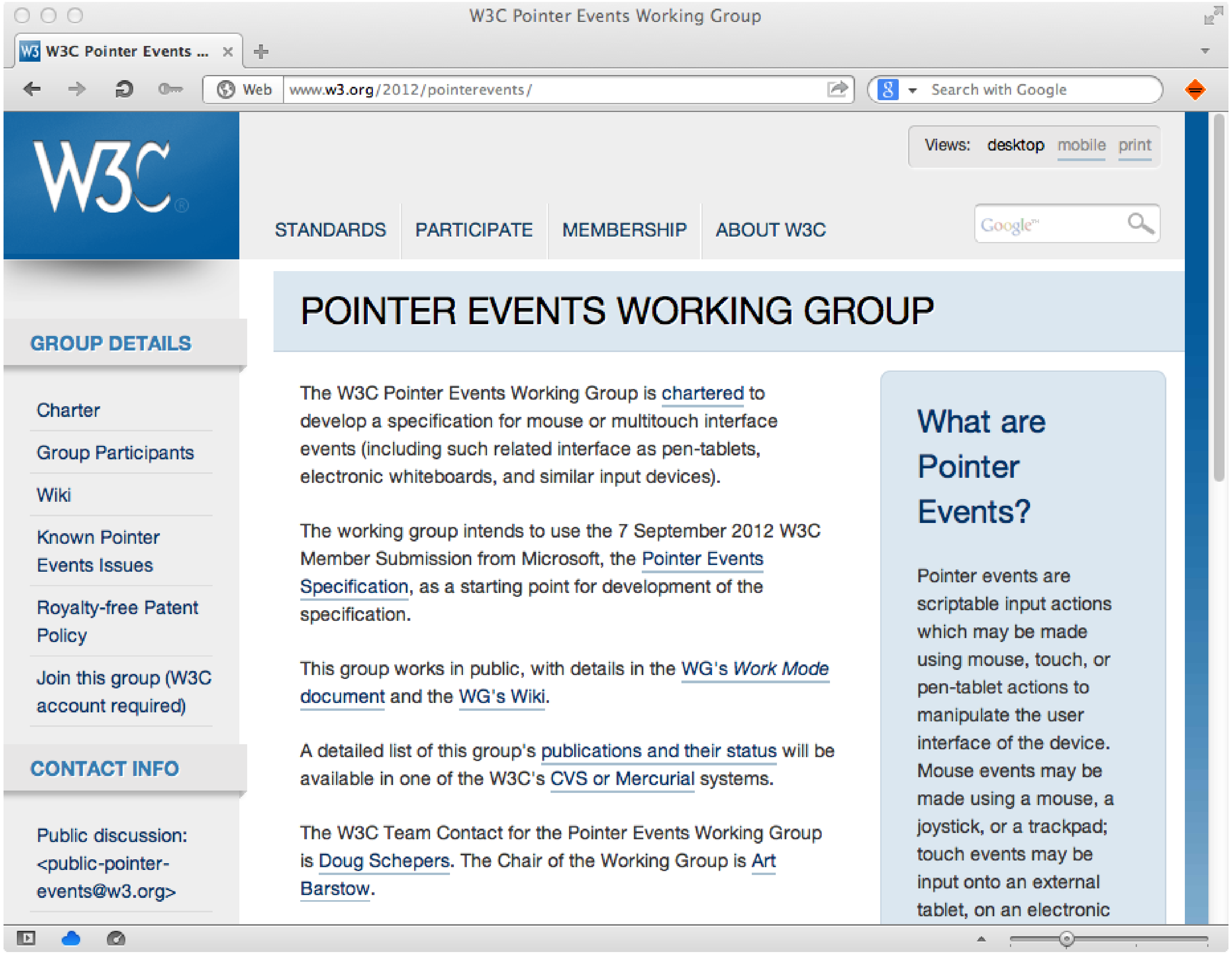
✓ Touch Events

shinydemos.com/picture-organizer

touch events and IE10



blogs.msdn.com/...


W3C Pointer Events Working Group
W3C Pointer Events ...
Web
www.w3.org/2012/pointerevents/
Search with Google
Views: desktop mobile print
STANDARDS
PARTICIPATE
MEMBERSHIP
ABOUT W3C
Google

POINTER EVENTS WORKING GROUP

GROUP DETAILS

- Charter
- Group Participants
- Wiki
- Known Pointer Events Issues
- Royalty-free Patent Policy
- Join this group (W3C account required)

CONTACT INFO

Public discussion: <public-pointer-events@w3.org>

The W3C Pointer Events Working Group is chartered to develop a specification for mouse or multitouch interface events (including such related interface as pen-tablets, electronic whiteboards, and similar input devices).

The working group intends to use the 7 September 2012 W3C Member Submission from Microsoft, the Pointer Events Specification, as a starting point for development of the specification.

This group works in public, with details in the WG's *Work Mode* document and the WG's Wiki.

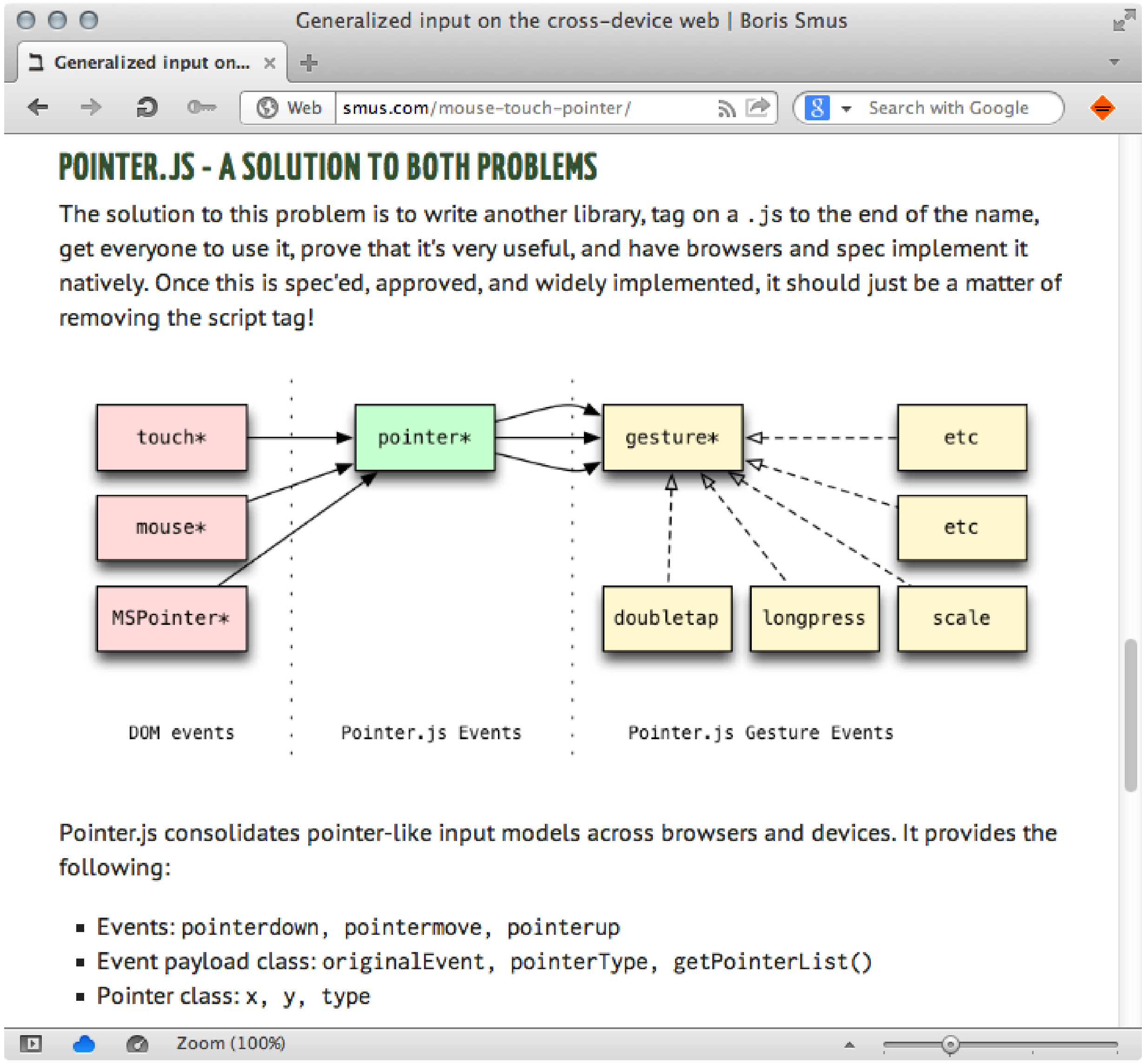
A detailed list of this group's publications and their status will be available in one of the W3C's CVS or Mercurial systems.

The W3C Team Contact for the Pointer Events Working Group is Doug Schepers. The Chair of the Working Group is Art Barstow.

What are Pointer Events?

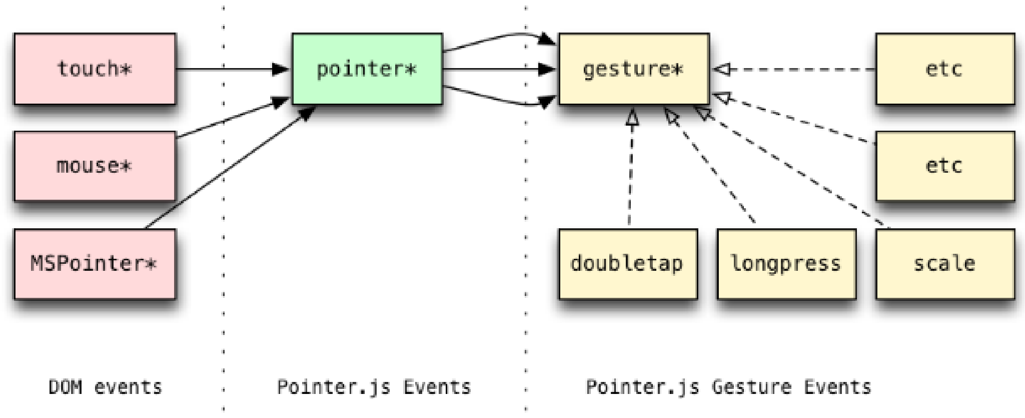
Pointer events are scriptable input actions which may be made using mouse, touch, or pen-tablet actions to manipulate the user interface of the device. Mouse events may be made using a mouse, a joystick, or a trackpad; touch events may be input onto an external tablet, on an electronic

www.w3.org/2012/pointerevents


Generalized input on the cross-device web | Boris Smus
Generalized input on...
Web
smus.com/mouse-touch-pointer/
Search with Google
Zoom (100%)

POINTER.JS - A SOLUTION TO BOTH PROBLEMS

The solution to this problem is to write another library, tag on a `.js` to the end of the name, get everyone to use it, prove that it's very useful, and have browsers and spec implement it natively. Once this is spec'ed, approved, and widely implemented, it should just be a matter of removing the script tag!



Pointer.js consolidates pointer-like input models across browsers and devices. It provides the following:

- Events: `pointerdown, pointermove, pointerup`
- Event payload class: `originalEvent, pointerType, getPointerList()`
- Pointer class: `x, y, type`

smus.com/mouse-touch-pointer

youtube.com/watch?v=AZKpByV5764

www.opera.com/developer

patrick.lauke@opera.com