About me

- Platform Architect - Mozilla
- Work on Web API team.
- We try to standardize Firefox OS API at W3C.
- Over the last year or so, been involved in W3C RICG.
- Lead editor of usecase and picture spec.
- Responsive images are important to Mozilla.
- Our target market is developing countries.
- People don't have much money and access to the internet is expensive.

Are we there yet?

Who cares?

- Question that comes up a lot
- Bandwidth and computing power
- New devices
- Range of screens just keeps growing

We do

- Turns out a lot of devs do
- Browsers: Microsoft, Apple, Google, Mozilla
- And CNDs
- So we formed...

responsiveimages.org

- Founded about 2 years ago by Mat Marquis
- Now the largest CG at W3C
- 300 members
- Grass roots effort - We do all our work in Github, and IRC
- Join us! (don't worry, will remind you at end of talk too!)

Why do we care?

- Turns out images are a huge performance and cost bottle neck on the Web
- I'm not going to ramble too long about this
- Search for Mat Marquis - who has spoken at length about this.

images - over 60%

- When compared to all other resources, images constitute, on average, +60%
- Average page size is still increasing.

72% same resources

- Turns out that despite people taking up the RWD cause, we are still serving the same amount of data to both desktop and mobile.
- Not always a bad thing! (probably a good thing!)
- This can sometimes be wasteful (too much data for a little screen).
- And can have a performance impact (e.g., sending all of jQuery).

First world problems

- These are my own pet peves.
- US trip - 5.15/MB
- EU trip - 0.55/MB
- Each website costs me €1
- I can buy Angry Birds for that!
- No joke - we often reach for apps when traveling. Not good for the platform.

First world 2

- This is what my phone bills looks like
- I’m paying 88 EU
- I only downloaded 268 MB
- Not happy...Vodafone.
- There are people far worst off than me.
- Every byte counts.

What are responsive images?

- Images that correspond to a particular environment.
- "Environment" is the the browsing context that the browser provides.
- We interact with some of that environment through CSS and JS.
- But other parts we cannot interface with - like user preferences.
- To understand - let's look at use cases.

Use cases

- Use cases is generally what we see people doing in the wild.
- A big part of what the RICG does is collect use cases and try to understand them.
- Having a good understanding of use cases allows us to formulate requirements.

Stretchy images

- The most common case is stretchy images.
- You know, images that will stretch to fit some column of content.
- This is usually done with the old "css-max-width" or "height".

Fitting different viewports

- Can be tricky, as different viewports can accommodate images in different ways.
- This image, by Paul Robert Lloyd, will serve a very purpose throughout this presentation.

Device pixel ratio

- What's the DPR of it iphone?
- How do we know that?
- DPR is hard.
- You need to understand the difference between a physical pixel, a device independent pixel, and a CSS pixel.
- Zooming also affects DPR.

Device pixel ratio 2

- Anyway, we know that having to upscale images can sometimes lead to crappy looking images.
- We need to remember that when we are talking about image sizes
- <img width and height> are in CSS pixels.
- How those images appear on the screen depends on DRP

Viewport matching

- Generally means matching on width of the device
- So, if your user’s resolution is 2560, you send something to match that size.
- On desktop, it’s easy.
- This gets funky when you add in DPR.
- Device width + DPR. 200, 2x = 400px wide.

Marching to portrait

- This is example from the BCC
- Generally, you target images to the way you assume your user’s prefer to consume content.
- But this changes from one device to another.

Switch to landscape

- When switching to landscape, this can have severe effects
- ... at least, you care about image quality.

Art direction * crop

- Art direction is the party-pooper use case.
  - It’s the elephant in the room.
- It comes in two forms: crop / swap
- Crop is easy to understand...
- But for a crop, you don’t want to download the big image
- Can potentially be done with CSS? Anyone?

Art Direction - Swap

- The more interesting case.
- Based on some constraint or environmental. Not just landscape (viewport is also very common).
- Used a lot of time with image + text placement.

Other use cases

- These are the lesser known
- Print - beautiful printed documents!
- File format support
- eInk - contrast control.

Usecases Doc

- If you want to know more...

Developer glass ceiling

- “Div-itis”: semantically neutral, annoying to write and maintain.
- We’ve been trying to hack around it.
- When we use JS, we bypass the preload scanner.
- Some solutions rely heavily on server side.
- This is not great. WE’VE HACKED THE HELL OUT OF THIS.
- We can’t go any further as devs....

microsoft.com

- A small responsive site
- Uses picturefill
- We are going to focus on the logo as a responsive image.

microsoft.com - code

- explain code...
- See replication of <img’s> attribute.
- noscript?!
- alt is repeated twice?
- note that logo-type-1x.png is repeated too.

Preload

- Prelaod scanner prioritizes img elements.
- Picturefill only executes on DOMContentLoaded.
- Problem of priority.

Without script

- We can see this without scripts running
- Order of requests is quite logical, follows the markup

Hacking around the problem

- Microsoft recognized this.
- Added this little snippet after each image.

Final code

- So, yeah, you have to repeat that for every image.
- That clearly sucks
- Can’t we do better?

What's a developer to do?

- We've go to extremes on this
- We've tried to use SVG
- Tried server-side solutions
- We keep smashing up against the glass ceiling
- So we go cry to ...

Standards bodies

- This is painful and time consuming
- Developers and browser vendors speak different languages
- We both see the world differently
  - Declarative vs imperative
- By the time you get here, you are probably going to be pretty pissed off.
- 2 years, we made progress...

Proposed solutions

- This is where we have gotten after 2 years.
- It’s been a pretty amazing journey
- It’s a story of people and perseverance.
- It’s a story of seeing the world differently.

The Good, bad, ugly

- There is rarely a perfect solution.
- There are always edge cases
- For each of the proposed solutions, I want to show the good, the bad, the ugly
- You would have seen some of these solutions before...
- let’s start with...

The picture element

- Dubious origins - Bruce Lawson, but we’ve found earlier references.
- It follows the <video> and <audio> element.
- Relies on source and its attributes

picture - example

- elements, attributes
- source element and attributes

picture- the good

- It covers the use cases:
  - art direction, dpr selection.
  - it does advanced use cases too (print, eInk, etc.)
- Dev’s get it. Bruce designed it, right? So it’s going to be genius.

The bad

- Multiple elements are hard for browsers.
- Mixes media queries and markup.
- Some media queries can’t be evaluated until after layout.
  - If the source changes the height/width then you might end up in an infinite loop.
- Doesn’t fully solve the problem... consider...

The ugly

- We know picture works ok with viewport selection, art direction...
- But can can picture be used with stretchy layouts and flex box?

Picture 95

- John Mellor from google tried to find out...
- Why he works at Google and we don't :)
- Some notable things here: repetition of sizes over and over again.
- I know what you are thinking...

How did it come to this?

the srcset attribte

- How did it come to this?
- Note that the media query part was not really the problem
- “srcset” doesn’t get the problem.
- Let’s take a look at it...

srcset - example

The good

The bad

The ugly

What if I told you...

<img>

- Sometimes we get so much into a problem that we don’t see if from every angle:
  - We are very used to markup, js, of c++
- This (<img src>) is what we really want! Not all that complex crap.
- What if we approach the problem at another layer.
- Think, the browser knows everything...

Client Hints

- HTTP-based solution.
- It was created proposed by Ilya Grigorik.
- It's been in the works for about a year.
- So how does it work?

Client hints example

- Using a HTTP-based opt-in
- Sends a request for things the server can negotiate on.
- Extensible.

The Good

The bad

The ugly

So yeah..

Drawing board

BOOM!

- Steve Jobs's it!
- Man, we are freeken clever.
- If only...
- Clever guys from Google: Tab Atkins and John Mellor.

RESPIMG SYNTAX

- Only proposed last week
- Subject to change
- Don’t get too excited - WHATWG not convinced.
- Let’s take a look...

Example

Respimg syntax

- Optional media query
- either a x-based url
- or a new viewport URL
- Yeah... what's that... saving that for the ugly bit.

The good

The bad

- Hard to polyfill because some browsers download empty <img>

The ugly

- The viewport URLs are not specifically ugly.
- Just a bit complicated to get at first.

viewport-ulr - simple

- for an image 100% of the available space, pic any of the following.
- The browser then choose the best one based on DPR, bandwidth, some other factors.
- Bit of magic...

viewport-urls fancy

viewport
100 30EM 400 70EM

- Explain code...
- I call this, the
- Spray and Pray!

Spray and Pray!

So, yeah... yeah

but wait

- The browser can pick the right image from a range of images
- The browser knows it’s own DPR
- The browser knows all the things.
- So...

Responsive Image Container

- Non-standard proposal...
- Proposed by Friend of the RICG, Yoav Weiss
- Or if we lovingly know it...

Magic Image Format

How does it work

- File based - no need to touch the markup!
- ..
- ...

Manifest

Resolution switching

- The image gets upscaled..

Art direction

Mind blown! Take my money!

The Good

The bad

- New file format - browser hates them
- users can’t share!

The ugly

Crystal ball

- Don’t know what’s going to happen.
- Probably want all of them.
- Client hints is nice for CDNs and those that can do con-neg
- src-n is nice as it covers most of the cases.
- but srcset is simple... Come talk to me!

science fiction

- ok, enough sci-fi

What can you do today?

Compress your images

- Lost art - back in the 96s
- Transition from dialup to broadband.
- Great tutorial on HTML5 rocks by Colt McAnlis

HTML5 Rocks compression

- Talks about cost to deliver images
- Has general practical advice

Lazy - like me

- What if you are lazy... I mean, a busy person?
- Like me? This presentation is 20+mb!
- Sorry if viewing on mobile!
- Use a CDN or get the server to handle it
- Pagespeed, Akamai, CDNConnect, free open source tools too (image chef).

Compressive images

- Originally research was done by Dann Jobsis
- Popularized by Filament group
- Considered an abhorrent abomination by some... but, it’s what we got.

Secret recipe

Side by side

Compressed image

- Yeah, I know you can't see the artifacts.
- Anyway, again... careful with this.
- Again, test it on mobile.

Picturefill

- If you need a client side solution.
- RICG recommended.

Other solutions

- Chris Coyier’s CSS tricks describes a great range of solutions.
- Discusses both client side and server side.

Join the RICG

- If you are not happy with any, want to do better, then join us!
- We want people who are passionate about the problem
- The RICG community is awesome! Community if friendly, helpful, diverse
- We do everything on GH.
- We have something for everyone to do!

Sizer Soze

- Latest project from Yoav Wiess.
- It tells you how many bytes you are wasting.
- Collaboration with many folks to give this tool a front end.
- Again, if you have ideas - it’s a good place to hang out.

x-picture (x-src-n?)

- We are also trying to work on a web components version.
- Working with the awesome guys from national geographic
- Going to bug Angelina Fabbro to help us! :)

Thanks!