Move stuff. More faster.

Towards as-fast-as-native animations

Current state of the art?

Accelerated CSS animations.

Cons:

- Not expressive enough for some effects.
- Only opacity and transform.

What about rAF?

Pro: very expressive.

Con: subject to main thread jank.

What's the ideal?

Drive animations from a clean thread.
JavaScript control would be nice, too.

…but which thread?

Main Thread?

This is ideal, but we'll have to de-jank.

Ideas being thrown around:

- *Find and kill stupid inefficiencies!*
- overflow:clip
- manual mode
- partial layout
- and many more…

What about a Web Worker?

Would be great, but you can’t touch style.

…but what if you could?

Animation Proxy

Bundles up ‘layout free’ properties:

- transform
- opacity
- scroll offset
- hopefully more soon…

Can be used on a web worker if you’d like.

Surprisingly Expressive

With this kernel we could implement

- Accelerated CSS animations.
- Smooth scrolling.
- Parallax.
- and lots more…

Example: Parallax

```
function raf callback(timestamp) {
  overflow_scroll_proxy.getScrollOffset().then(do_parallax, error);
}

function do parallax(offset) {
  offset.scale(parallax scroll scale);
  parallax scroll proxy.setScrollOffset(offset)
    .then(function(result) { self.requestAnimationFrame(raf_callback); },
          error);
}
```

How will this really look?

In practice, probably more like this:

```
fancyLibrary.parallax(scroller, 'bg.png', 0.9);
```

That is, direct consumers of `AnimationProxy` will most likely be library authors.

But how will it work?

(Wave hands)

Concerns

More threads == more pain?

Still useful on few cores:

- *Gives preemptibility if not parallelism.*
- *Provides an API with no performance cliffs.*

Concerns

Committing to implementation details?

Don't think so.

- *Hooks into accelerated CSS animation machinery.*
- *Have to support this anyhow.*

What about events?

Could let proxies listen to events.
As with proxies, proxy events would be limited.

Gesture recognition in a worker.

Further reading

https://github.com/ianvollick/animation-proxy

Other idea: CanvasInWorkers

Would allow,

- Rendering to a canvas from a worker.
- Render multiple canvases with a single rendering context.

More info:

http://wiki.whatwg.org/wiki/CanvasInWorkers

...but back to main thread.

What's stopping us from making it jank-free?