Geekbang 极客邦科技　InfoQ

QCon 全球软件开发大会【北京站】2016

青瓷引擎 - 打造 HTML5 游戏引擎开发工具之路

QICI H5

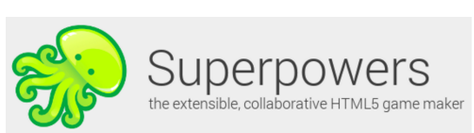
林意炜

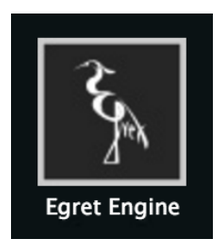
Intemational Software Development Conference

CONTENTS

An Overview of Games and HTML5

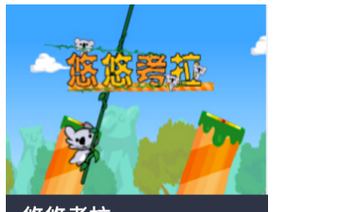
- **HTML5 Game Market**
- **HTML5 Game Development**
- **Domestic and International Ecosystem**





On Whose Shoulders?

- **Pixi.js Rendering Engine**
- **Phaser Game Framework**
- **Unity User Constructs**
- **JavaScript/Node.js**
- **Web Browsers**



悠悠考拉

一款测试预判和反应能力的小游戏，总想完美落地，却一次又一次的摔死！

小黄鸭跳跳跳

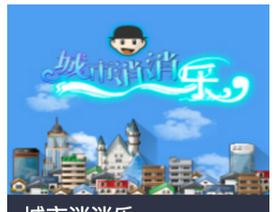
点击屏幕的红绿按钮控制小黄鸭在对应颜色的方块跳跃，你的速度有多快呢？

圣诞危机

不要小看它只是一个五子连消的游戏，玩不好一分钟之内就game over

进击的方块

轻点右侧跳跃，左侧降落，连击两下可连续跳跃，前方高能，小心手指抽筋



城市消消乐

你想成为店长？村长？镇长?市长?洲长？还是地球统治者呢？

航空指挥官

点击屏幕选中飞机，为飞机规划线路至轨道入口安全降落，注意飞机碰撞哦

粉碎瓷砖

点击任意颜色即可消除，颜色相同三块自动消除，每次点击方块都会上升一层

蛇精病

史无前例的超级魔性小游戏，不是简单的贪吃蛇，晕车指数玩了才知道哦！

神奇的六边形

玩过俄罗斯方块，挑战下更高难度的神奇六边形吧，任意直线填满即可消除哦

2187

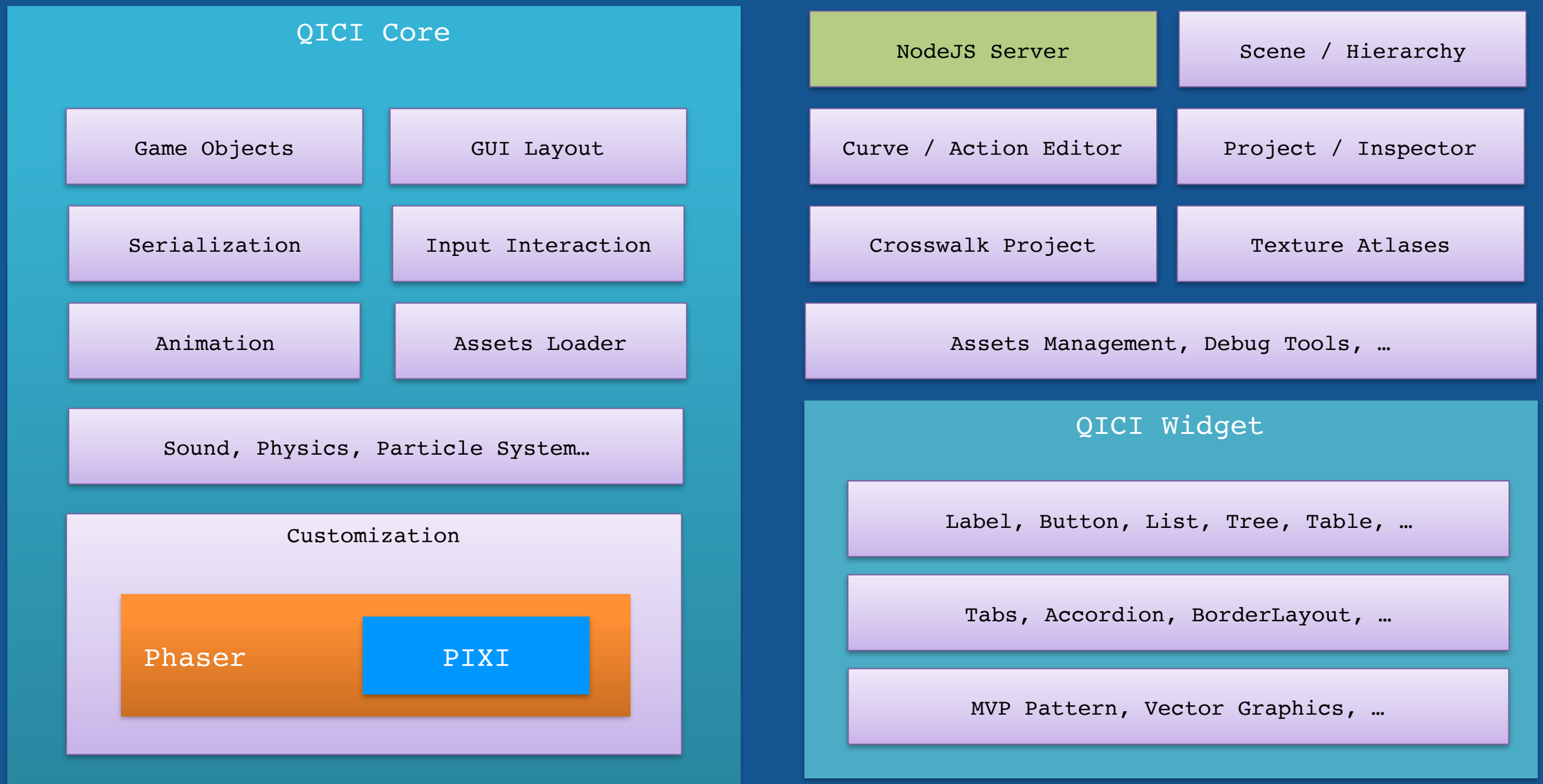
别看它只是个数字，别以为2187就是终点，考验大家智商的时刻来啦！

跳跃的方块

跳跃的方块，轻点屏幕，方块即可跳跃，来看看你可以跳过几层吧！

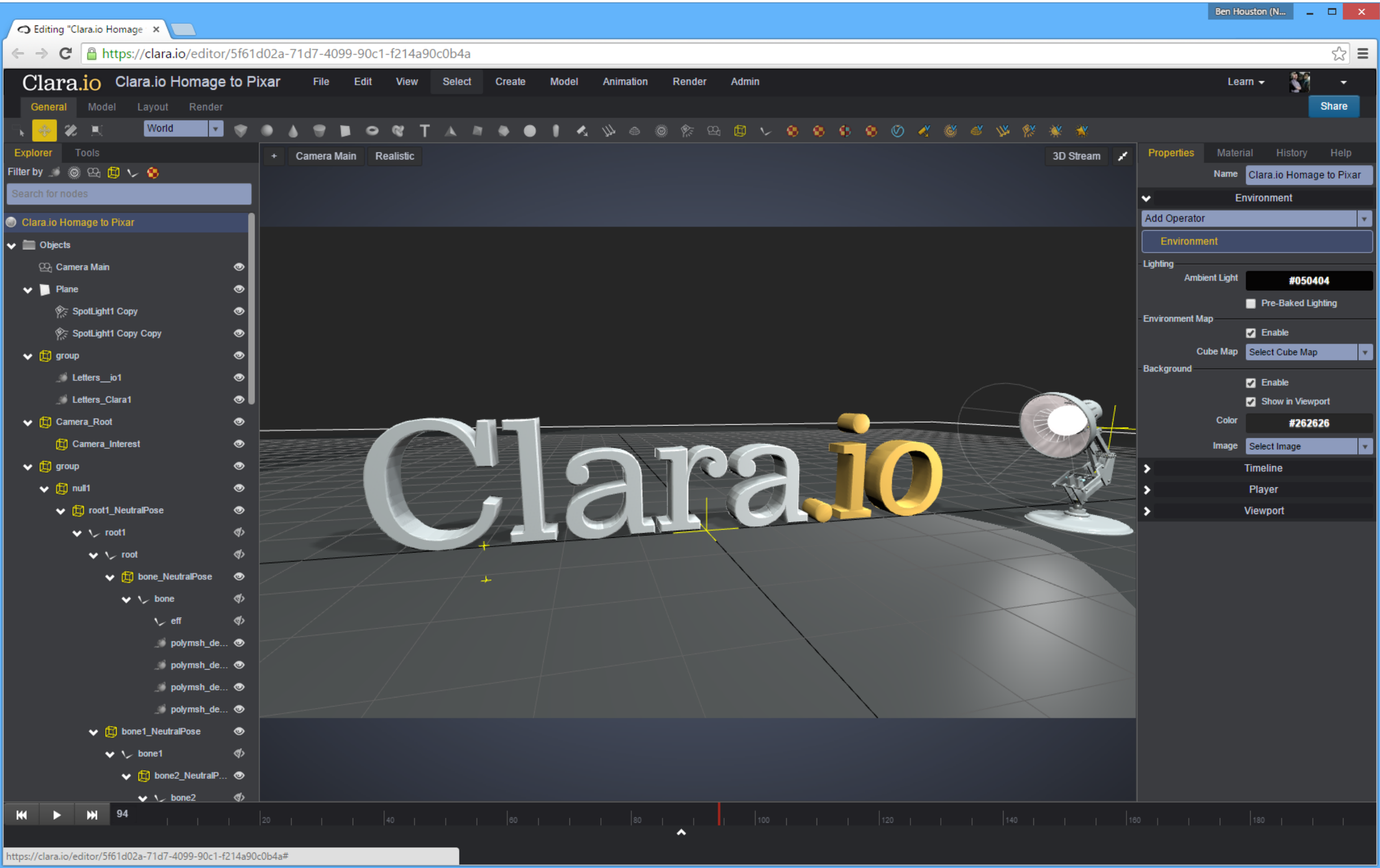
The Shoulders We Stand Upon

QICI Engine Technology Stack

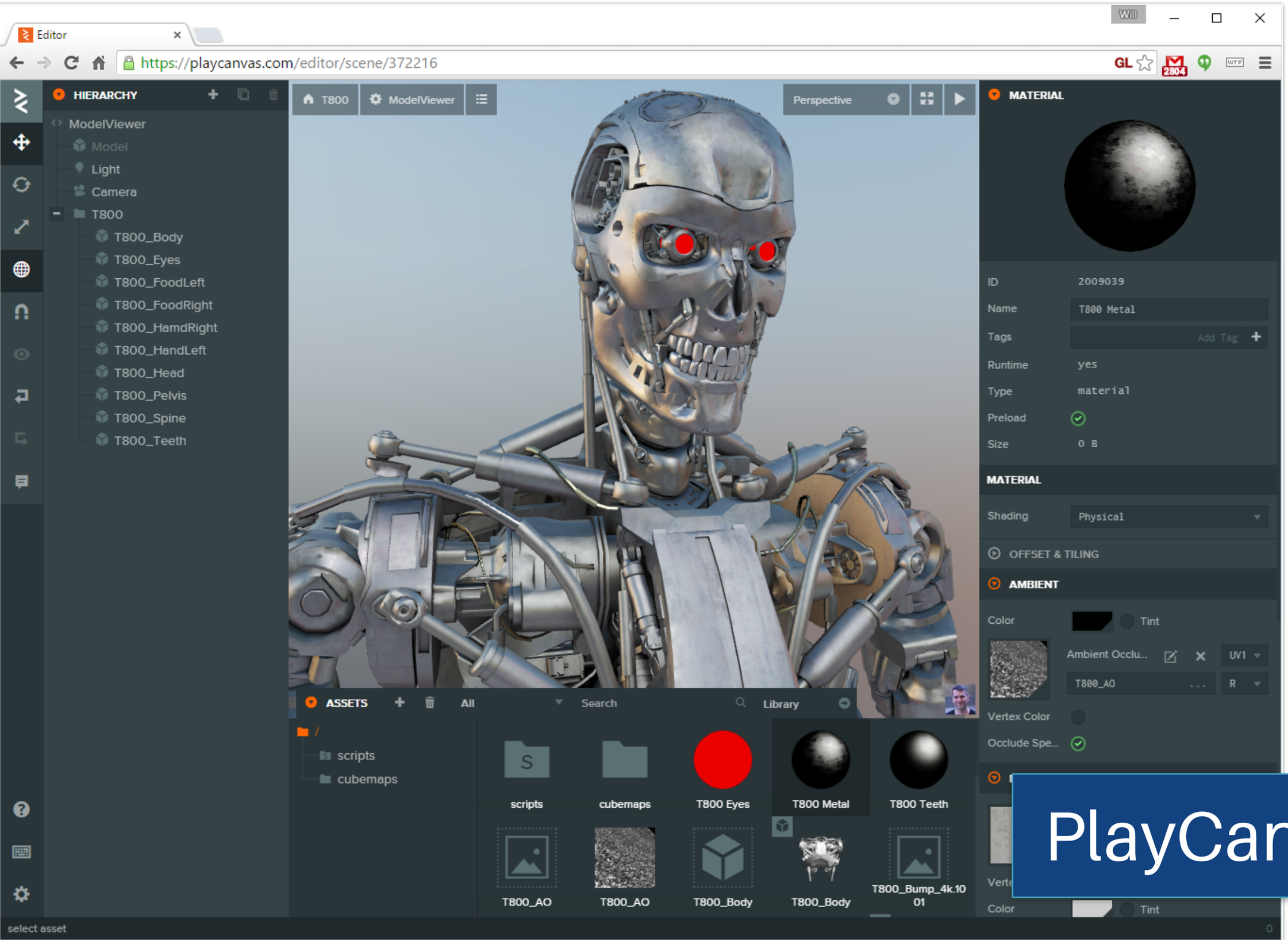


Why Web based Technologies?

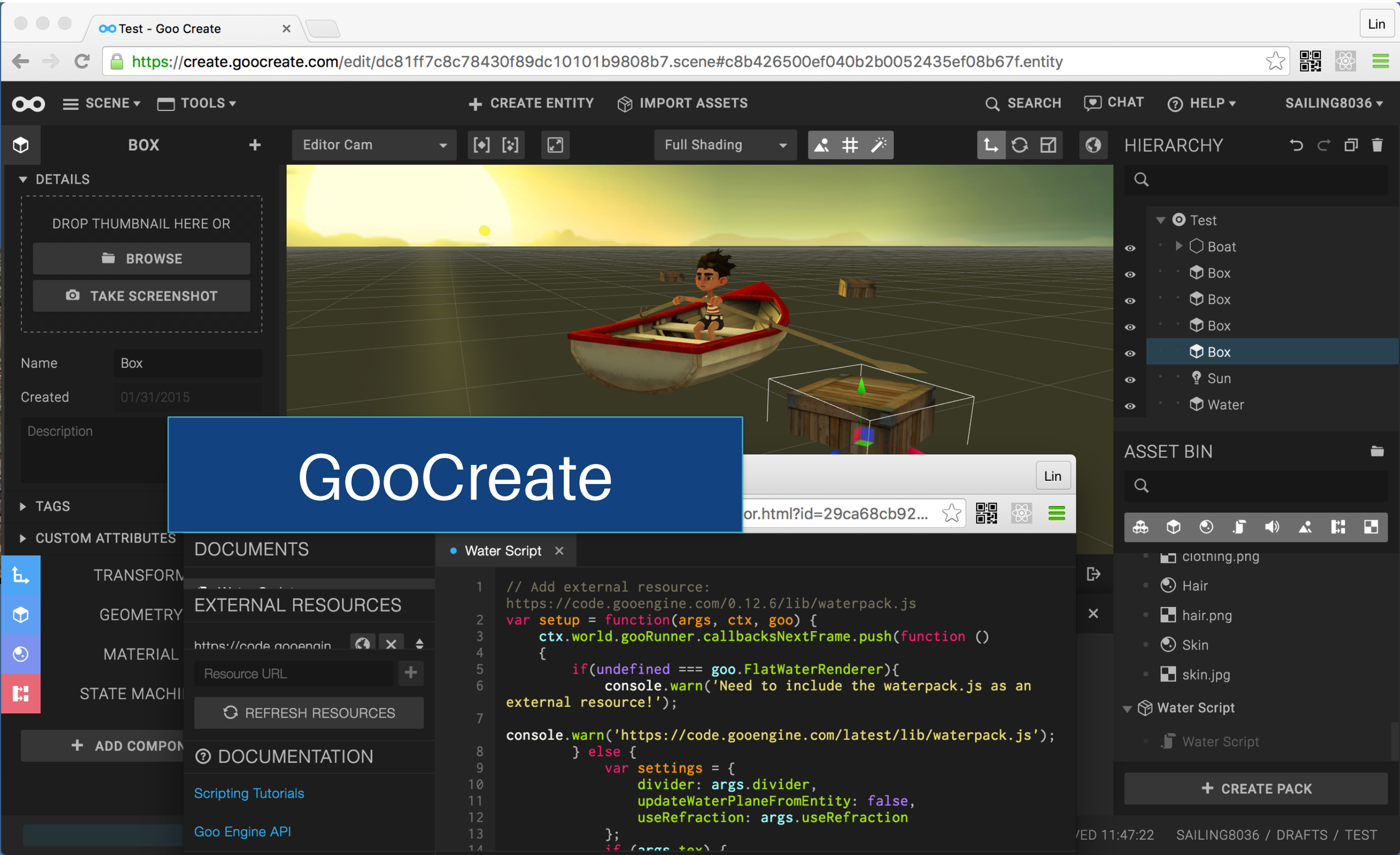
Why Web based Technologies?



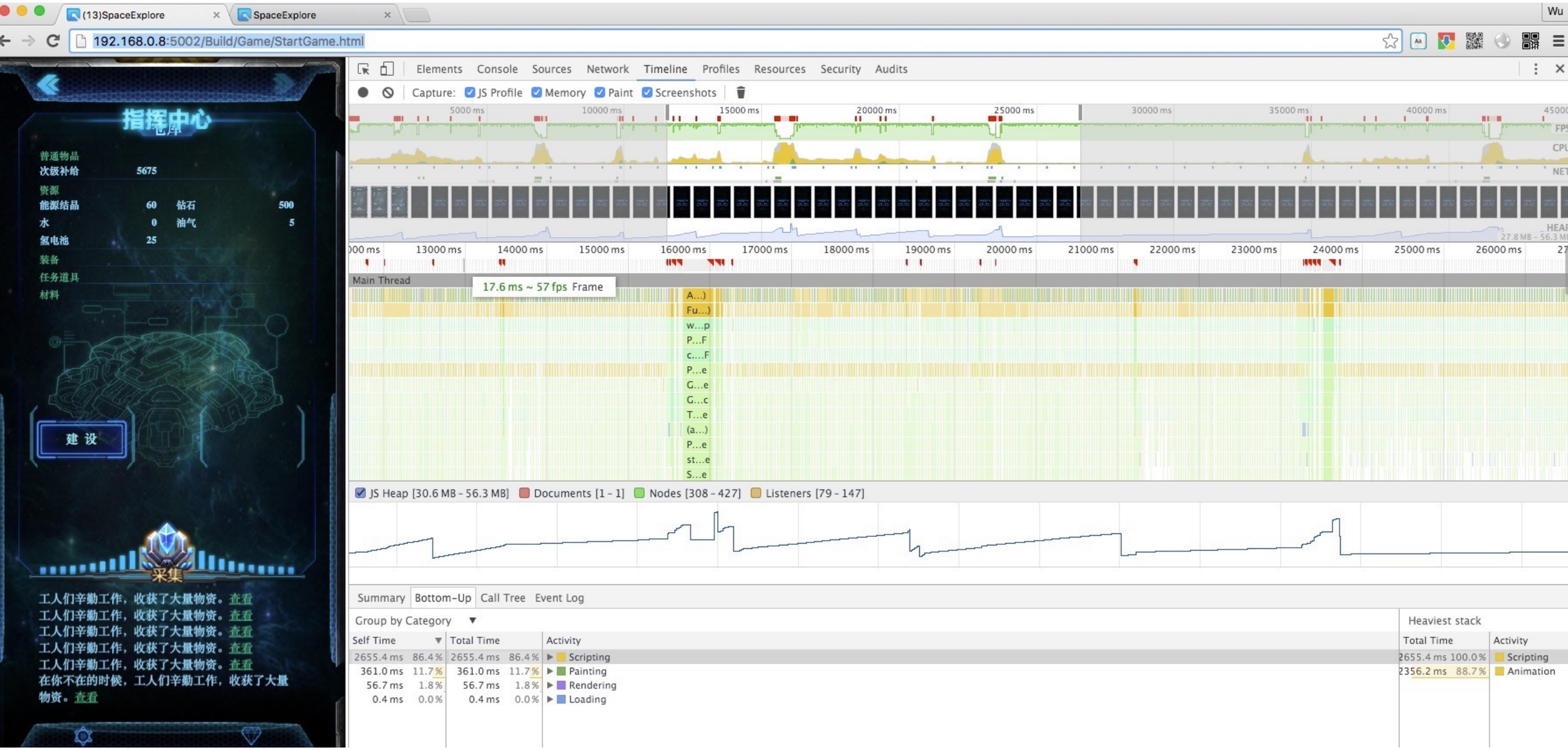
Why Web based Technologies?



Why Web based Technologies?



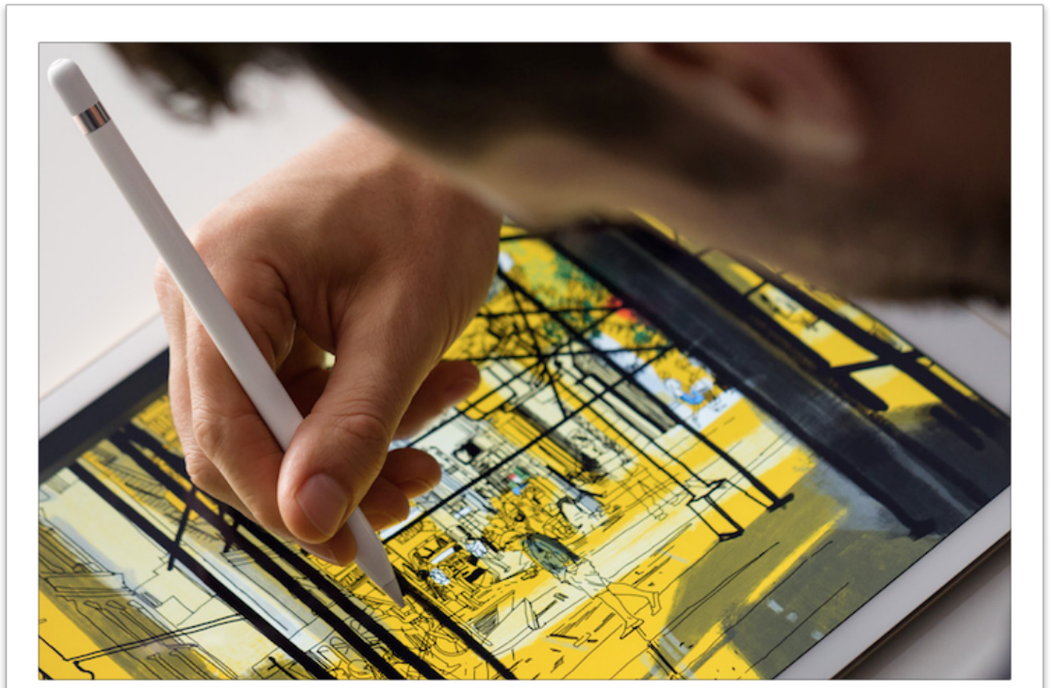
Why Web based Technologies?

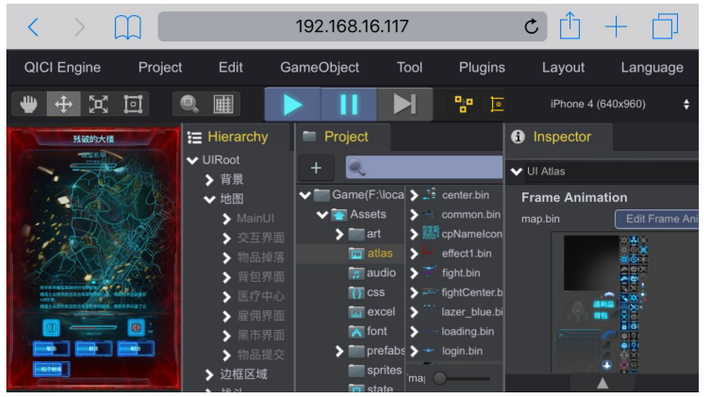


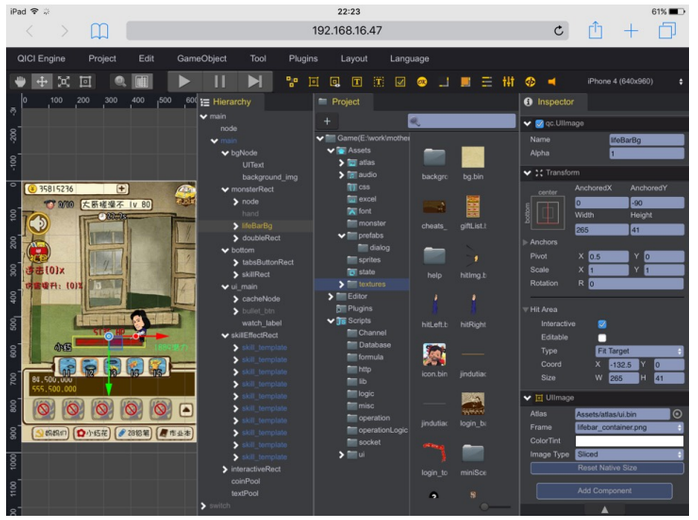
Leverage Browser Debugging Tools

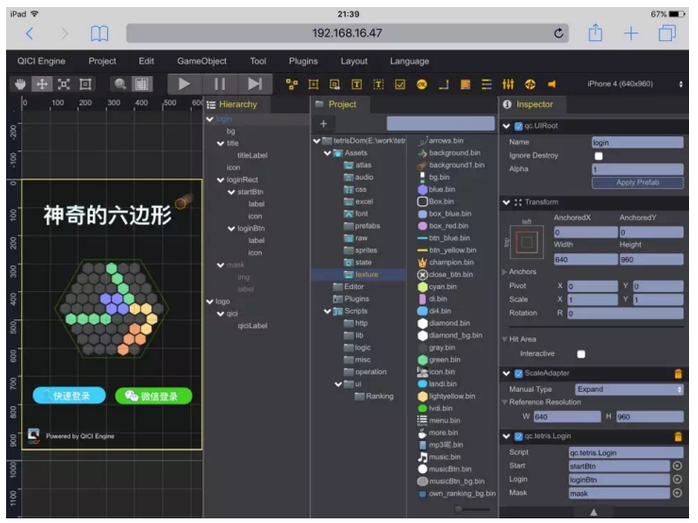
Why Web Technologies?

Design
Build
Play
on
Tablets





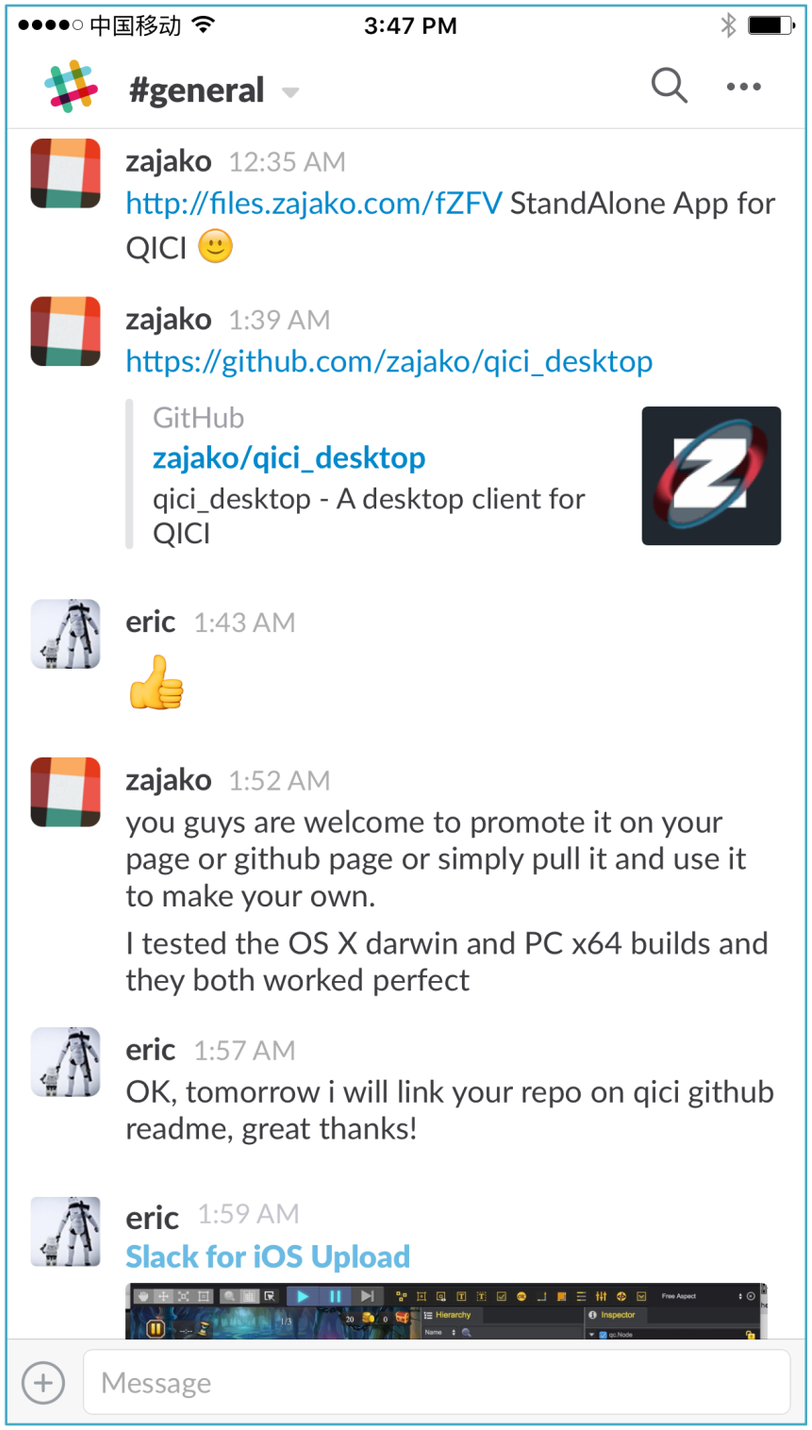


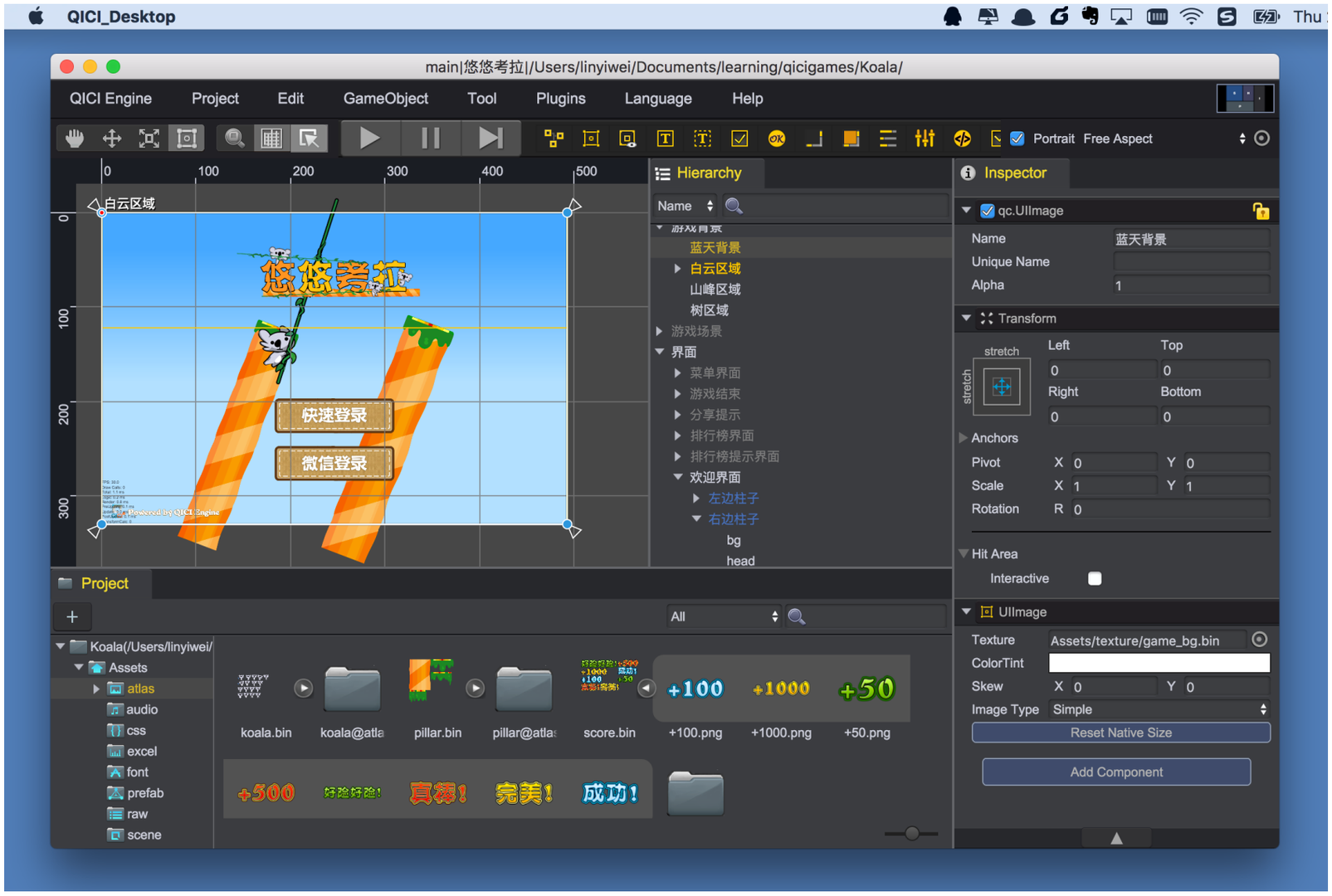


ELECTRON

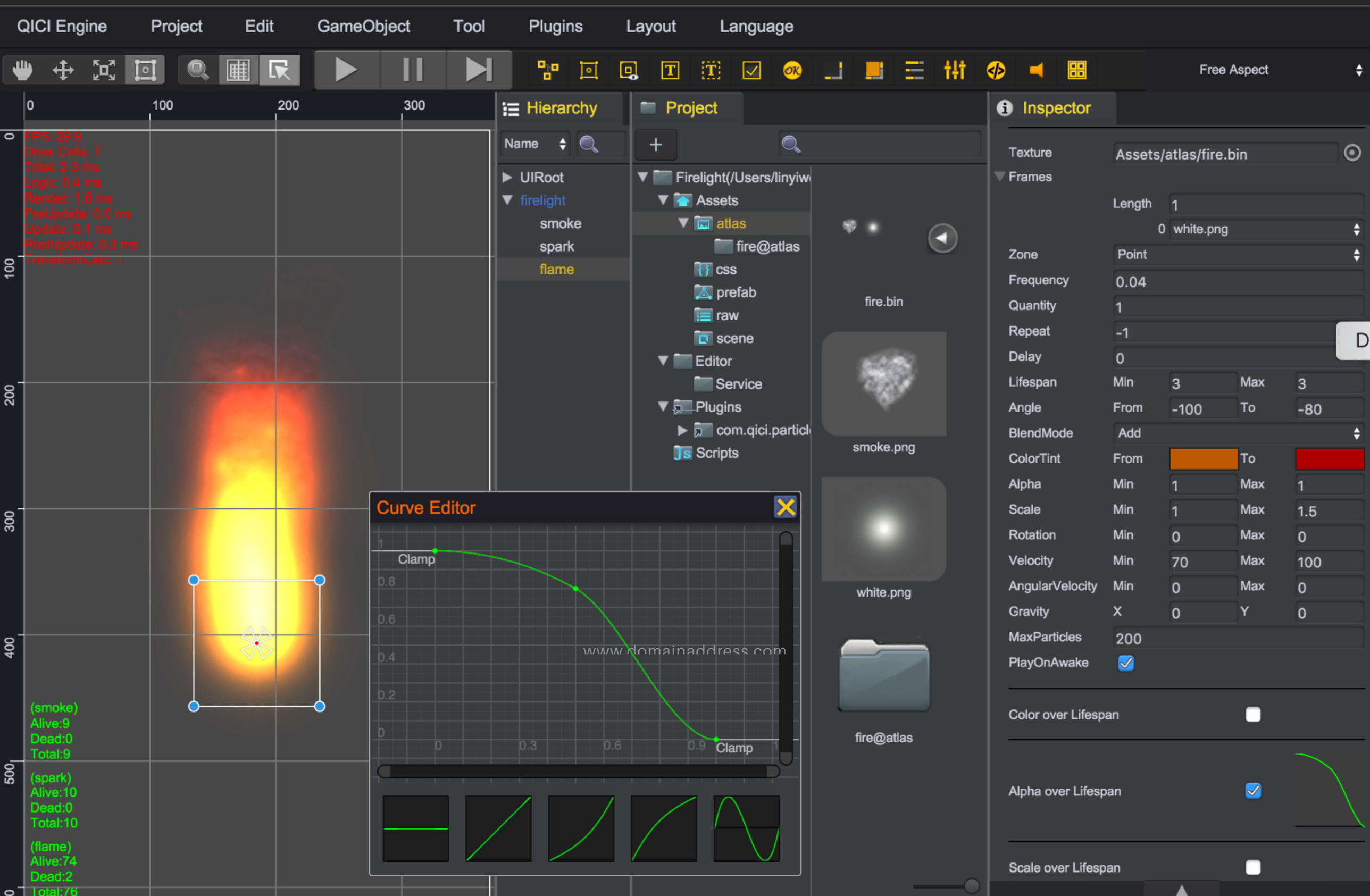
Build cross platform desktop apps with web technologies

Why Web Technologies?

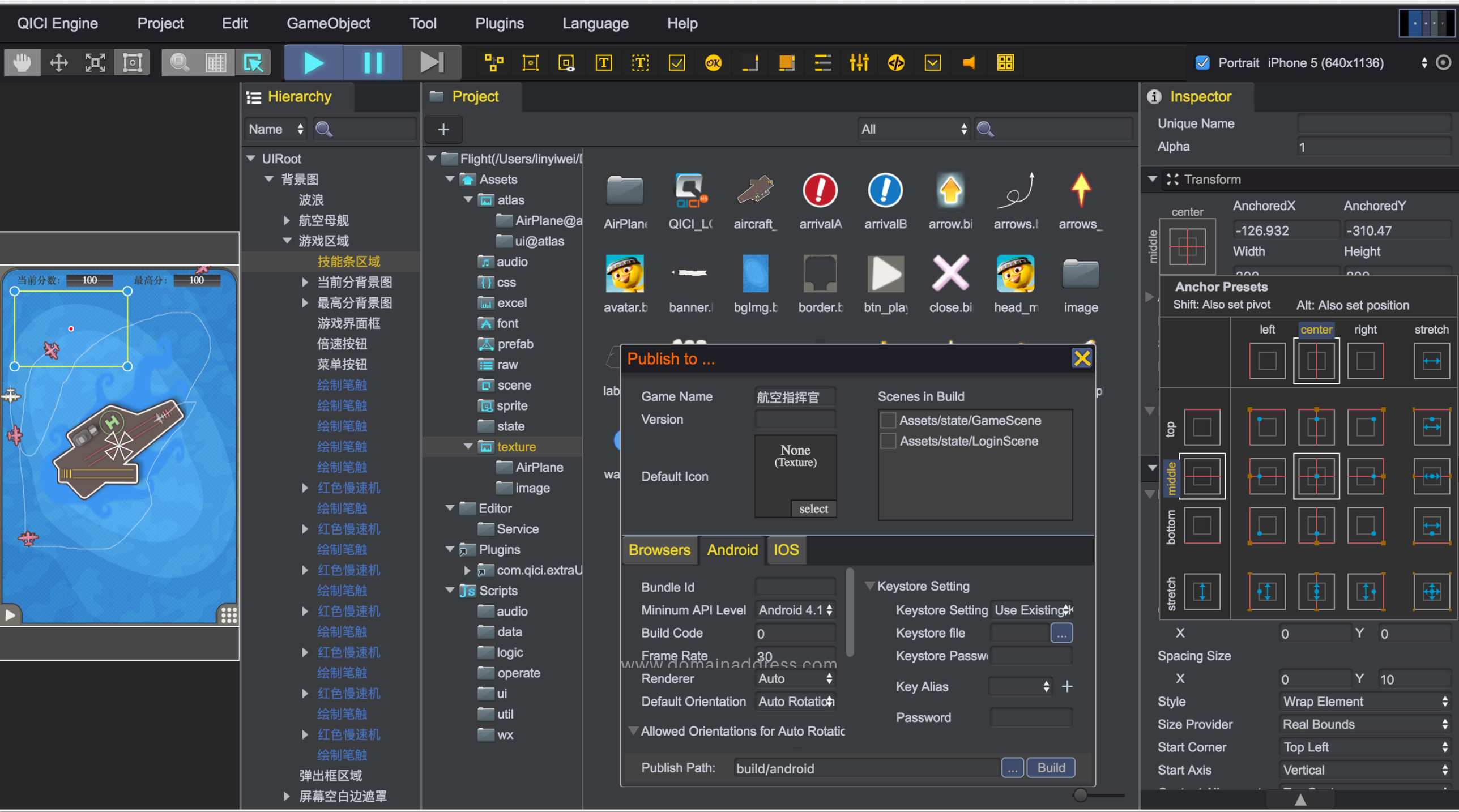




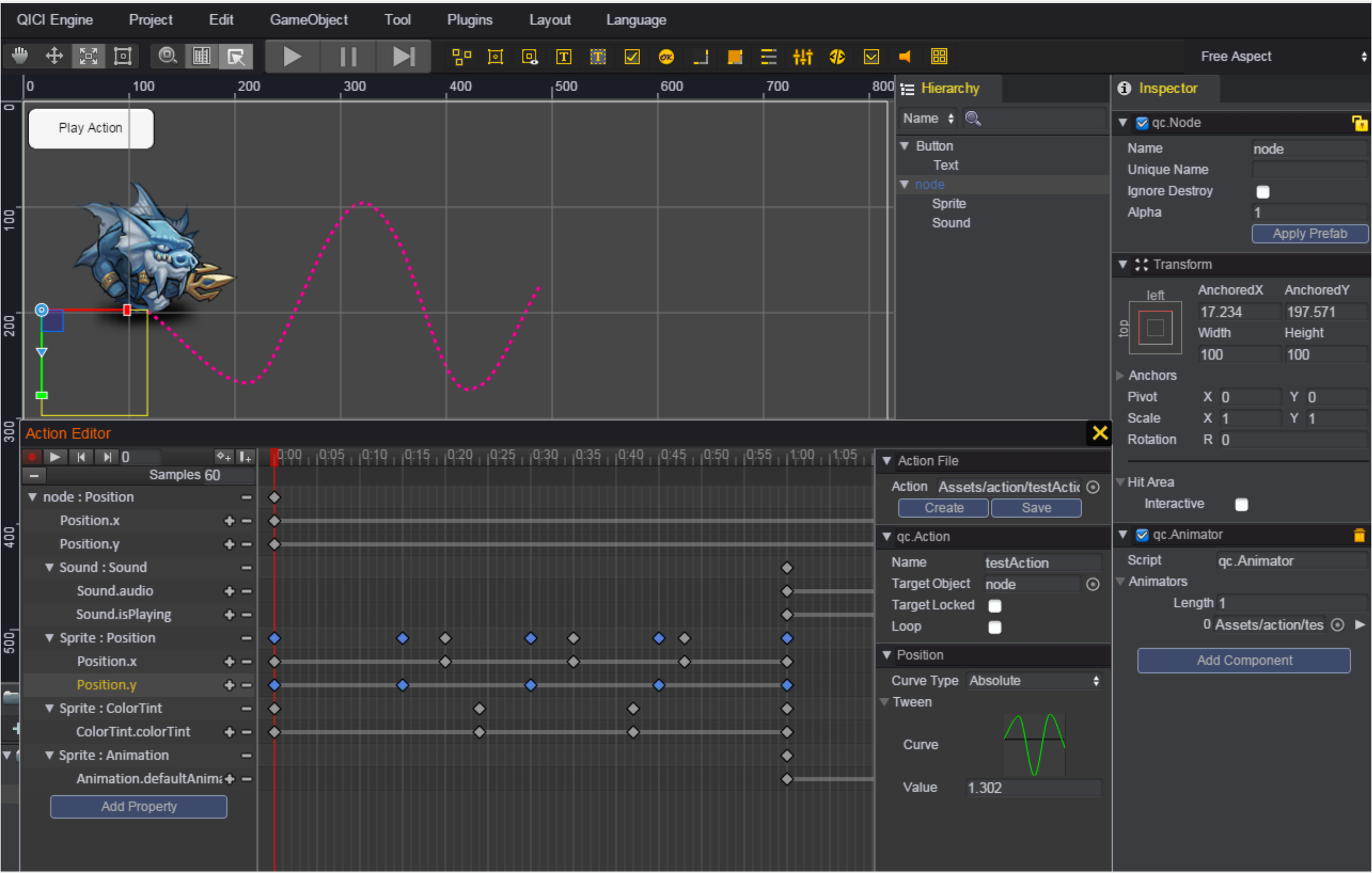
Why Reinvent the GUI Wheel?



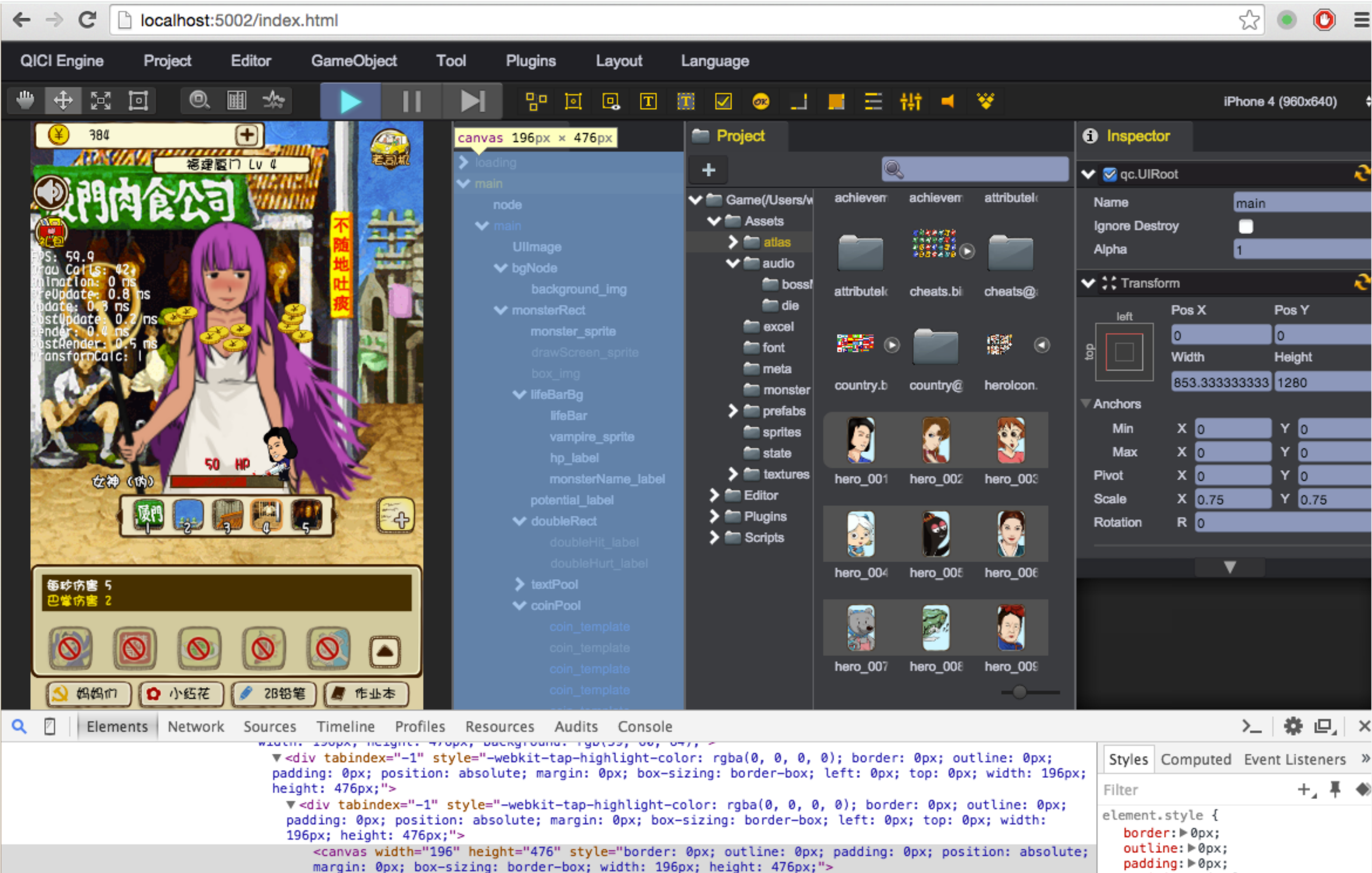
Why Reinvent the GUI Wheel?



Why Reinvent the GUI Wheel?

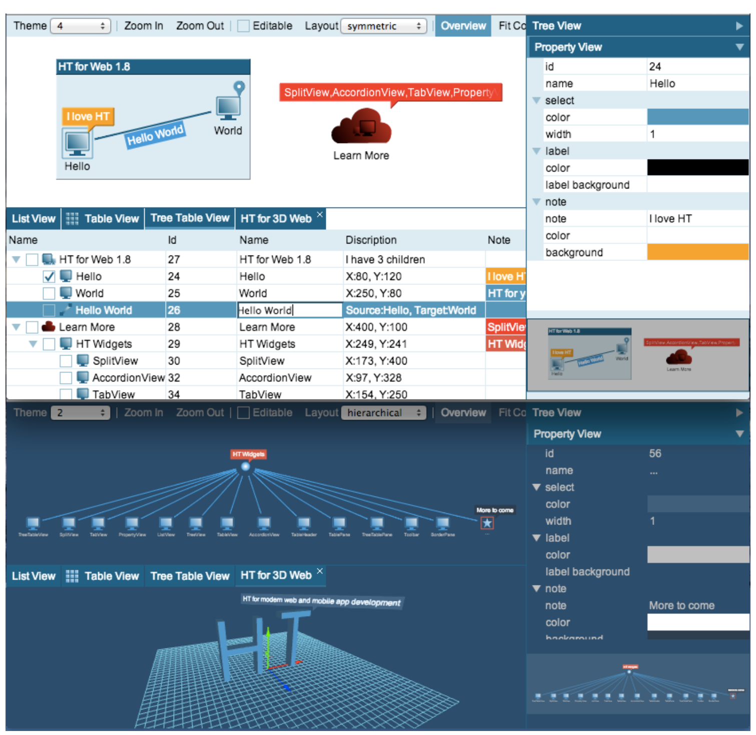
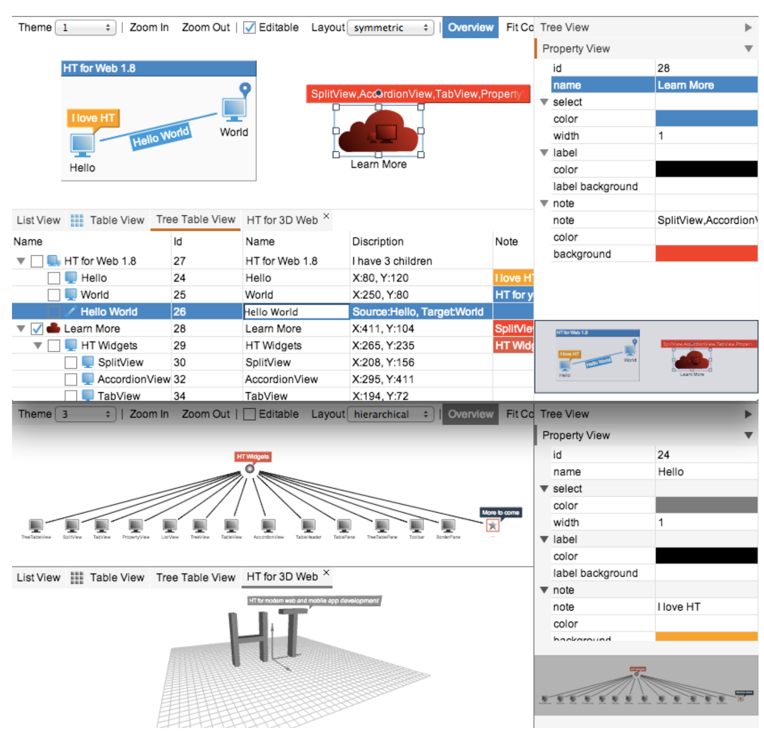
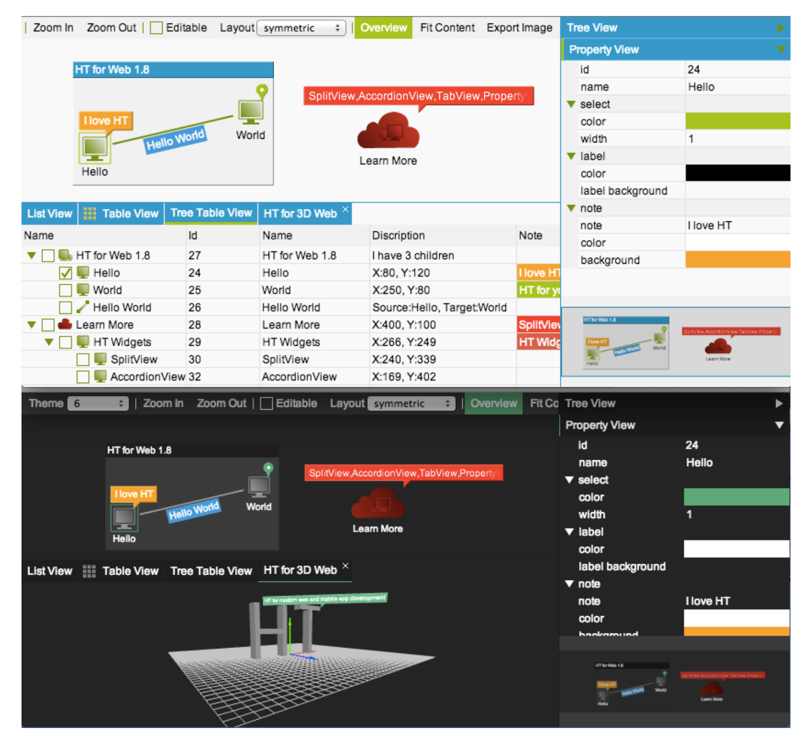


Why Reinvent the GUI Wheel?



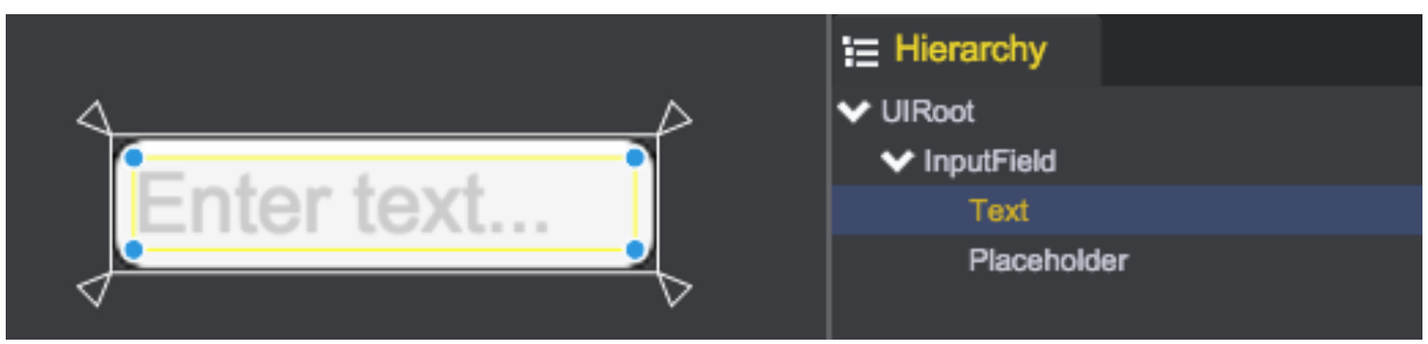
Why Reinvent the GUI Wheel?

- **Vector Graphics with Data Binding**
- **Heavily Canvas Based for Ultimate Performance**
- **Unify Mobile and Desktop Experience**
- **Model-View-Presenter (MVP) Pattern**
- **Strive for Few Dependencies**

Canvas or DOM?

Mix Canvas and DOM to Leverage HTML/CSS

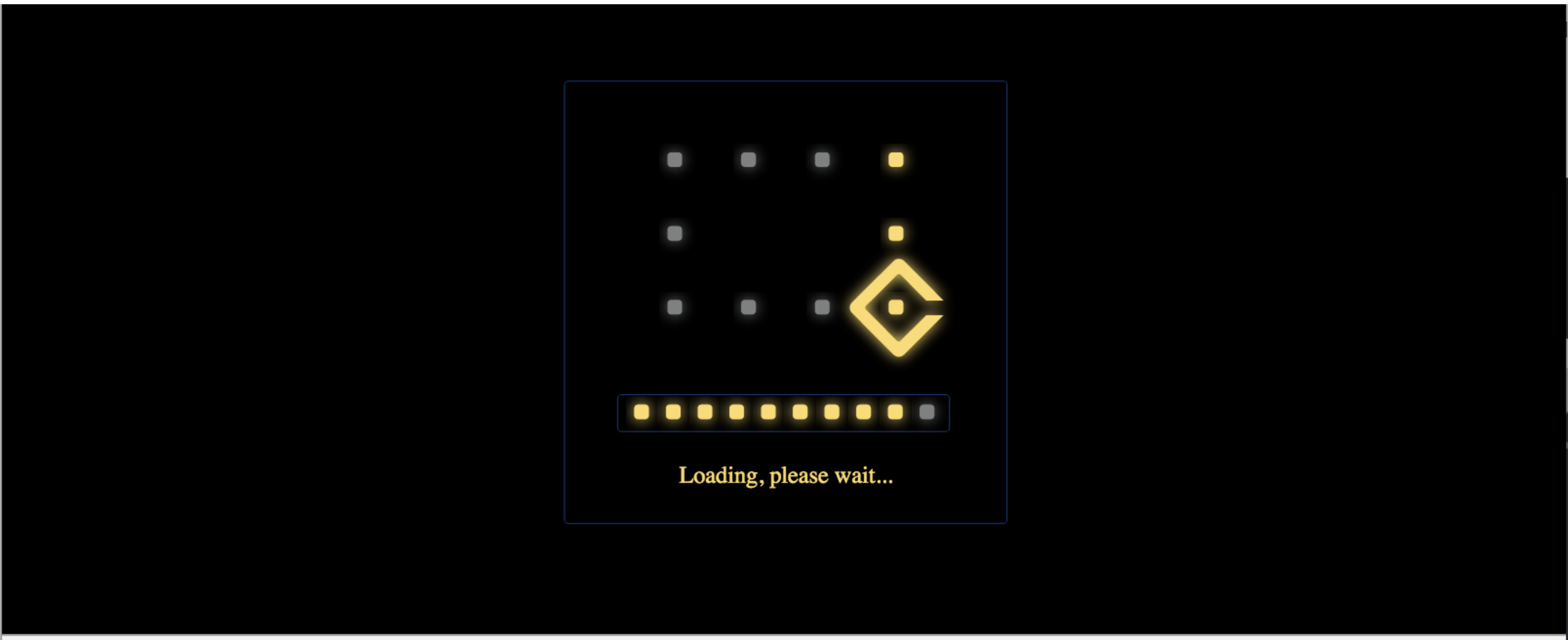


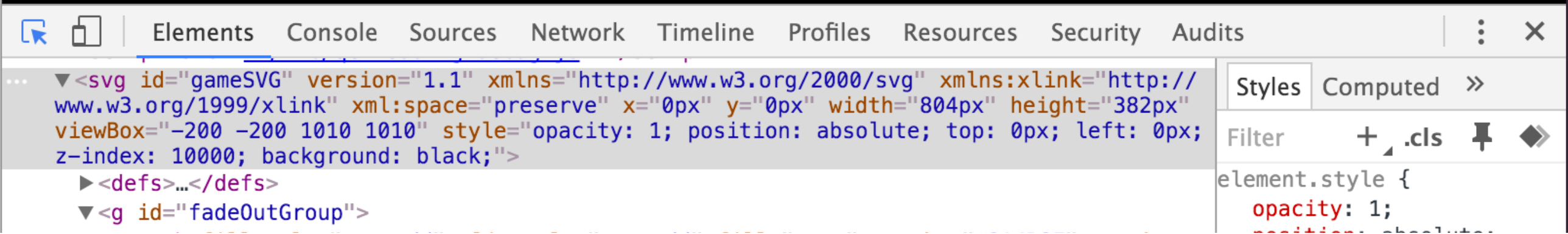
Canvas or DOM?

Avoid Cross-Domain WebGL Textures Security Issue

Canvas or DOM?

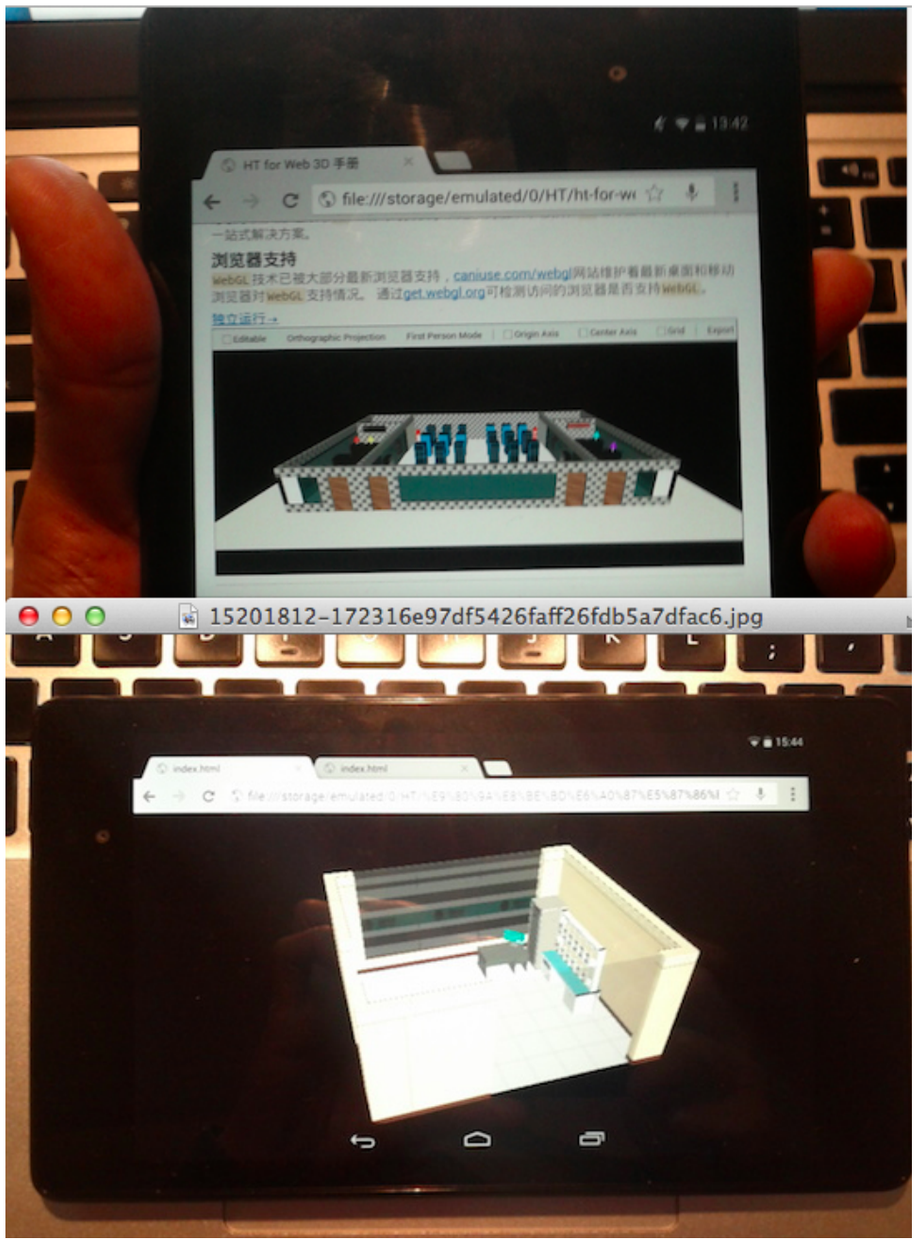
SVG Loading Indicators for resolution independence





Is WebGL Ready?

2013 - WebGL on Android



https://www.zhihu.com/question/19905844/answer/21015535

1、2013年应该感谢Google和MS，前者提供了Android Chrome浏览器的WebGL支持，后者终于让IE11支持了WebGL并且降低到win7也能运行。

2、还得感谢Firefox和Opera这些厂家推动了移动终端浏览器对WebGL的支持，并且性能还都不差，早期Chrome的Android版还不支持WebGL时测试只能靠他们。

3、感谢UCWeb浏览器提供了唯一（如果不是唯一还请高人指点）官方iPad的浏览器的WebGL支持，虽然有很多bug我们提交了问题也没人理，还好找到了workaround规避了些坑。

4、IE11很给力，我在mac下的VirtualBox虚拟机里面IE11是唯一能跑WebGL的（其他浏览器不知道是否有开关设置可以在虚拟器可以跑的，如果有还请高人指点），但还得努力啊，一堆基础的API都不支持，例如TRIANGLE_FAN类型都不支持，例如lineWidth都不让设置，例如Shading Language都不支持return语句，还有好多坑，但还好我们都找到方法绕开了（如果你也遇到了IE11的坑可以帮我继续列举）。

5、Nexus 7二代很给力，作为WebGL的移动测试机，很轻且性能不错，终于找到了让我满意的愿意随身带着用的Android板。

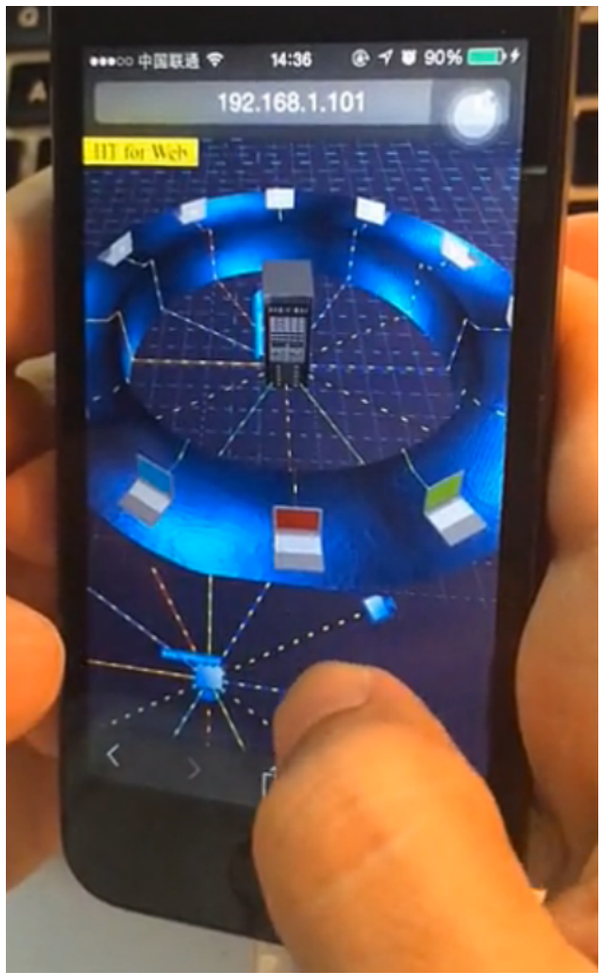
6、打包android的WebView控件方式还很难搞，默认的WebView核与Android Chrome不一样，搞了个独立的Chrome 28的核还开启不了WebGL（要是有高人搞定了，还请指点）。

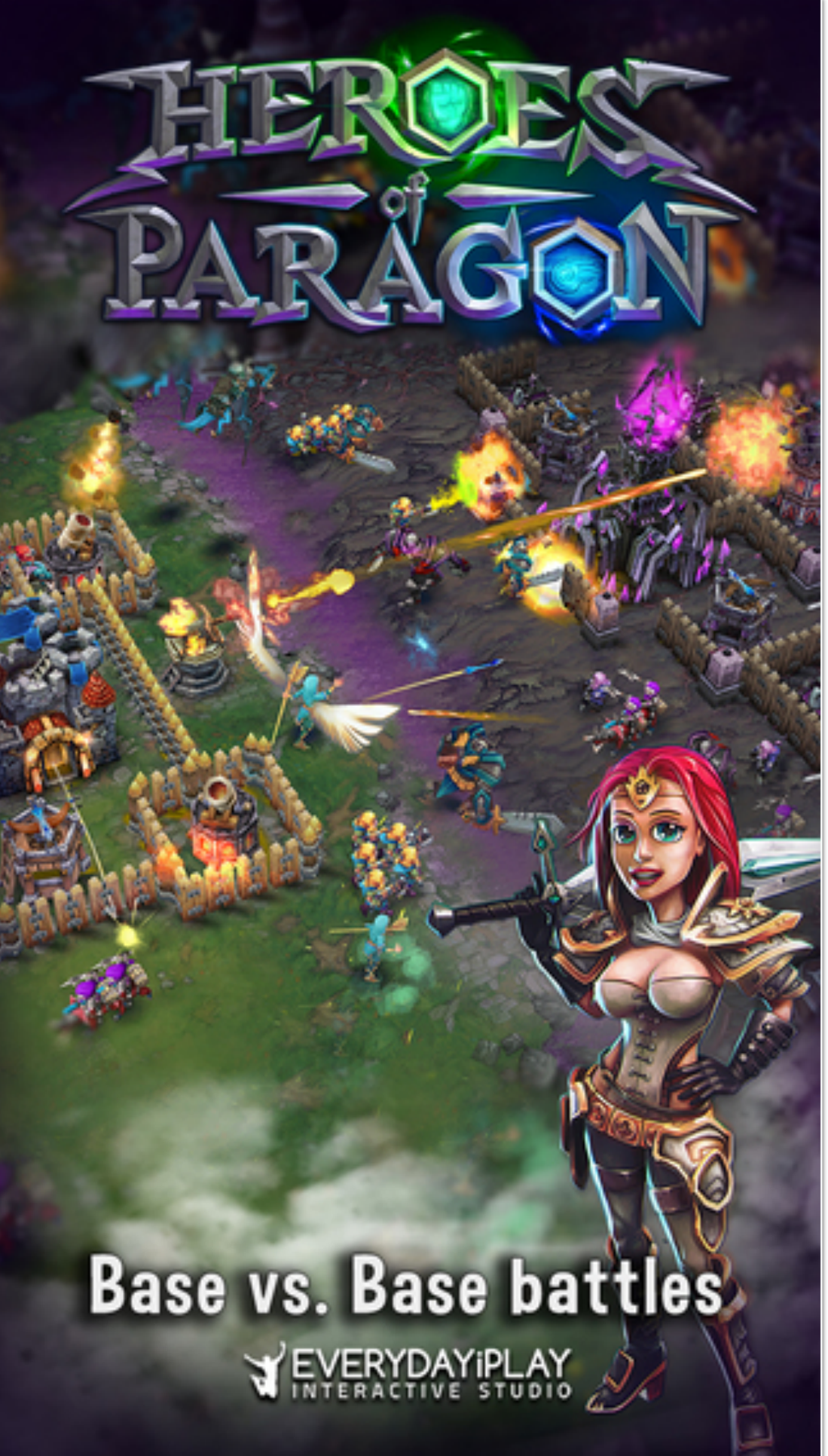
7、上点代表性的项目抓图，一个是IE11下的，一个是Android下的，总体说我感觉2013年经过各个厂家的努力，对于大多数的企业应用应该可以说：WebGL is Ready!

Is WebGL Ready?

2014 - WebGL on iOS 8

iOS 8





Is WebGL Ready?

Chrome Officially Disables NPAPI Plugin

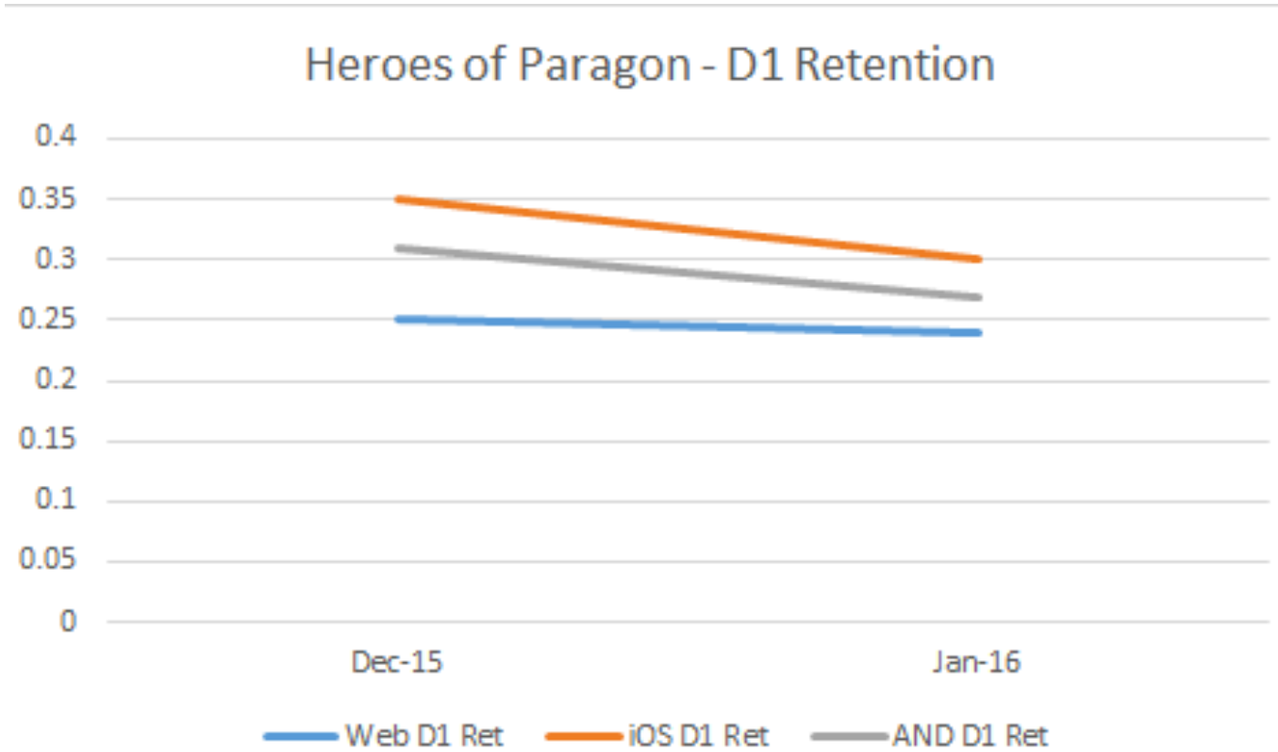
2015/2016 - WebGL meets Game Industry

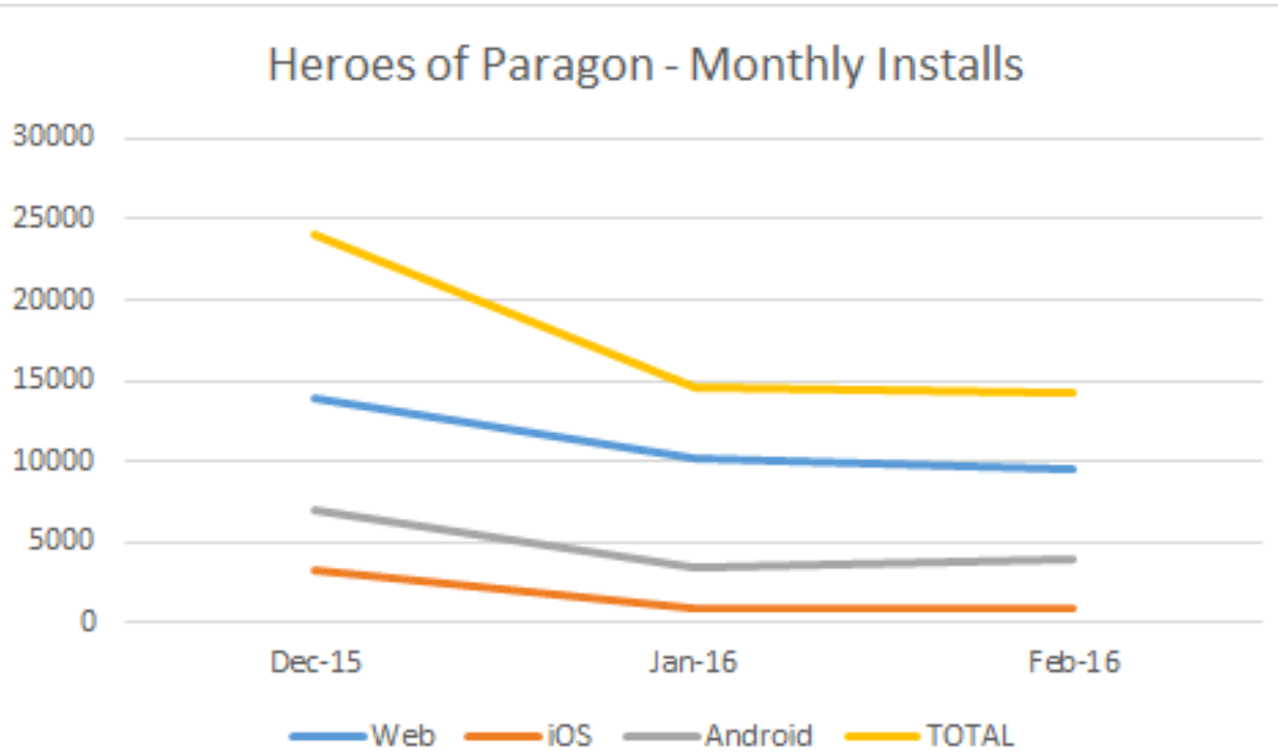
HEROES of PARAGON

Launched December 3rd 2015 on Facebook with WebGL.

Won Best Social Game at GCD Awards.

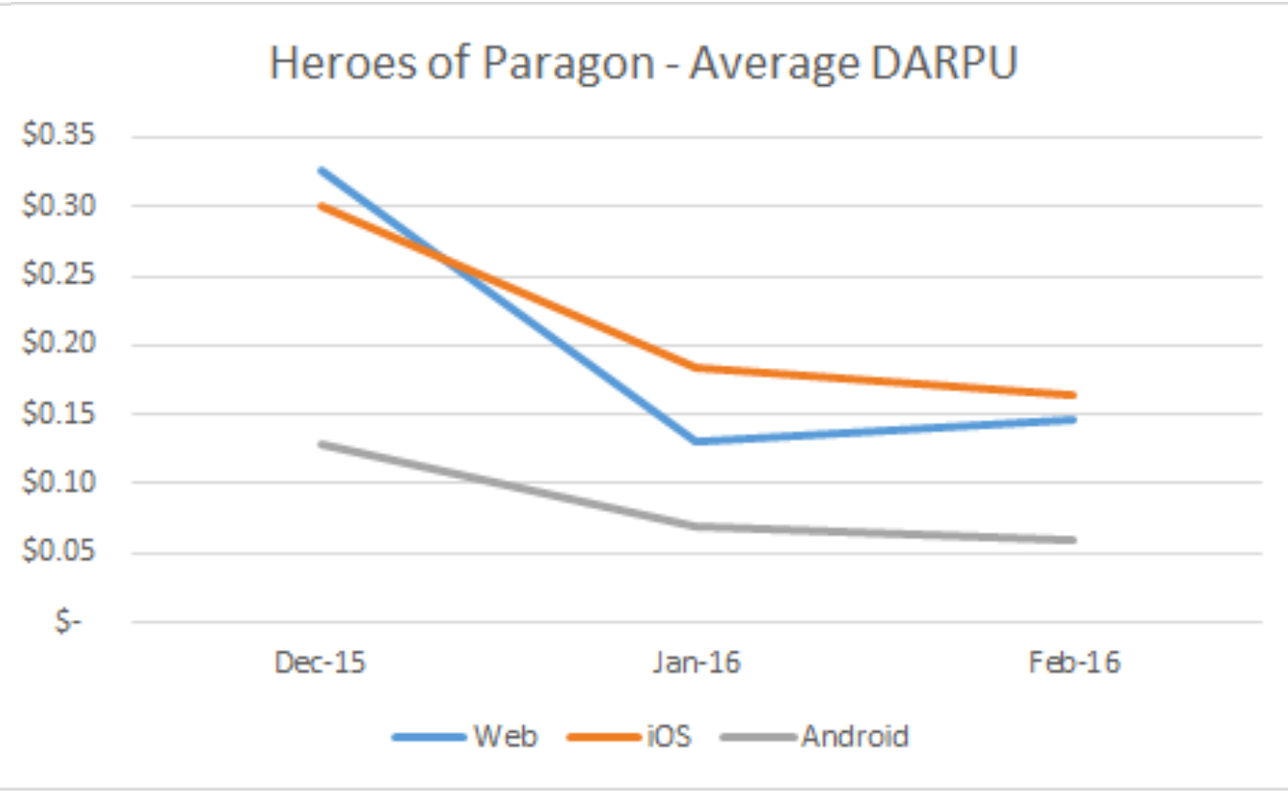
https://medium.com/@vvergon/webgl-matures-into-a-commercial-ready-technology-ba71e254cae6#.361f9csgu





WebGL Readiness:

HEROES of PARAGON



https://medium.com/@vvergon/webgl-matures-into-a-commercial-ready-technology-ba71e254cae6#.361f9csgu

WebGL Readiness

WebGL 2.0 is well on its way...

Brandon Jones
@Tojiro

@brianblakely WebGL2 should be shipping "pretty soon" and WebVR "not quite as soon but still soon." :D sorry, dates are hard to nail down.

2/25/16, 12:35 AM

Ben Houston
@BenAtExocortex

A great big thanks to @mattdesl for his awesome work adding WebGL 2 support to #ThreeJS: github.com/mrdoob/three.j...

Mobile Browsers: HTML5 Compatibility

iOS issues...

- Backgrounding host crashes WebGL rendering
- Web Audio and Audio Tag both need to be unlocked on touch event (Use ‘touchend' for iOS 9, ‘touchstart' for previous versions)
- FPS drops after application re-activated from background on iOS 9, fixed in iOS 9.3
- Use ‘shrink-to-fit’ meta-tag or ‘documentElement.clientHeight/Width’ to workaround ‘window.innerHeight/Width’ iOS issue on iOS 9

Mobile Browsers: HTML5 Compatibility

Android Audio and Sound FX issues...

- Stalled audio does not report errors consistently and has binary use of stopped or error report. Need monitor both
- Sometime Audio Tag showing wrong duration. Save the duration value in meta file when editing as alternative
- Web Audio is still not well supported
- Concurrent audio is very limited and problematic

Mobile Browsers: HTML5 Compatibility

Third-Party Browsers issues...

- UC browser has no gradient fill for text, requires workaround: gradient rect on text - globalCompositeOperation as ‘source-in’

- Tencent's X5 hardware acceleration only on first 5 canvases; ensure game’s rendering is within these

- UC browser returns incorrect height value after keyboard disappears

- If ES6 features used, Emscripten transcompiles are incompatible with any browser not ES6 compliant.
  eg: Math.fround(x) & Math.imul(x, y)

Optimising Mobile Browser Performance

- Use AppCache for speed and offline browsing
- Use DOM and dirty rectangles to reduce power consumption
- keep your JS/HTML/CSS payload under 2MB
- Reduce resolution/Canvas size to reduce memory usage
- Avoid using ‘Stencil’ for some browsers in Android

Optimising Mobile Browser Performance

- Sample and cache computation for skeleton animation
- Avoid using ‘LINE_LOOP’ and ‘TRIANGLE_FAN’
- Limit batch size, and balance mobile and desktop
- Getters and setters call overhead is still significant on mobile
- Is WebGL rendering always faster then Canvas?

Optimising Mobile Browser Performance

Using Web Workers

ArrayBuffers can be transferred to a worker thread without copying by list buffers in an array in the second argument of:

`Worker#postMessage(message, transferArray)`

```
1  var ab = new ArrayBuffer(1024);
2  var uInt8Array = new Uint8Array(ab);
3  for (var i = 0; i < uInt8Array.length; ++i) {
4    uInt8Array[i] = i;
5  }
6
7  var worker = new Worker("worker.js");
8
9  // before transferring
10 console.log(uInt8Array.byteLength); // 1024
11
12 worker.postMessage(uInt8Array.buffer, [uInt8Array.buffer]);
13
14 // after transferring
15 console.log(uInt8Array.byteLength); // 0
```

Optimising Mobile Browser Performance

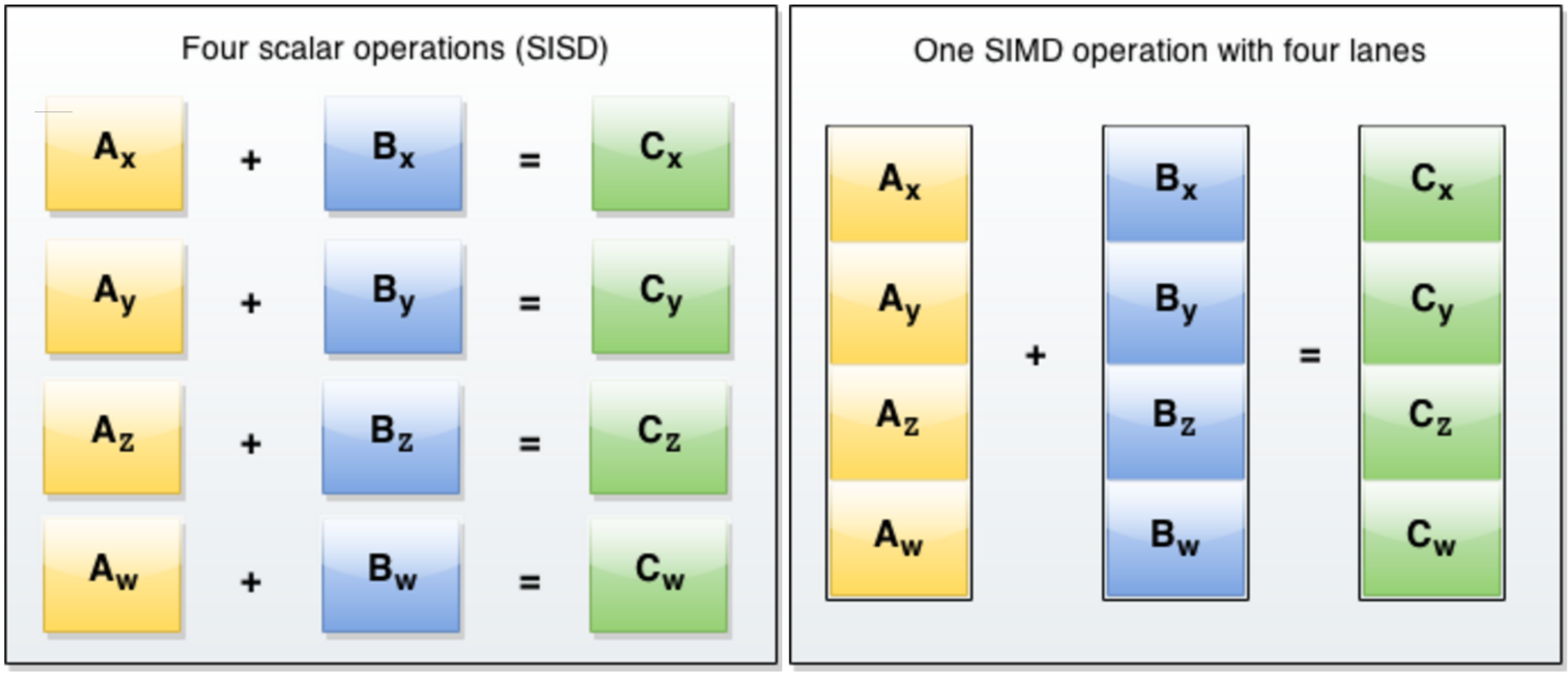
SharedArrayBuffer will bring threading support to JavaScript

```
var sab = new SharedArrayBuffer(1024);

// before transferring
console.log(sab.byteLength); // 1024

worker.postMessage(sab, [sab]);

// after transferring
console.log(sab.byteLength); // 1024
```

Using Web Workers

Optimising Mobile Browser Performance

Single instruction, multiple data (SIMD)



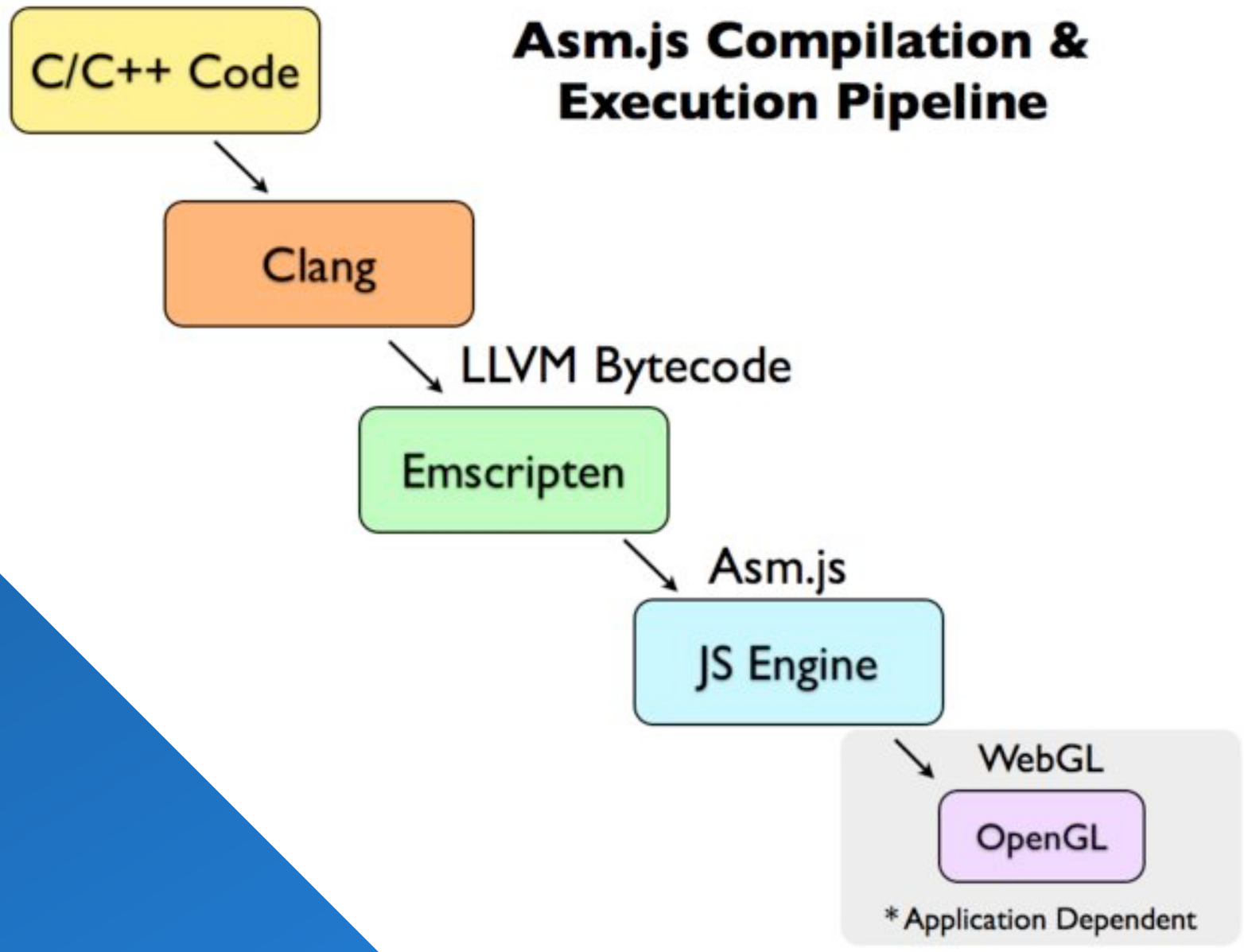

```
1 | var a = SIMD.Float32x4(1, 2, 3, 4);
2 | var b = SIMD.Float32x4(5, 6, 7, 8);
3 | var c = SIMD.Float32x4.add(a,b); // Float32x4[6,8,10,12]
```

https://github.com/tc39/ecmascript_simd/blob/master/src/ecmascript_simd.js

Optimising Mobile Performance

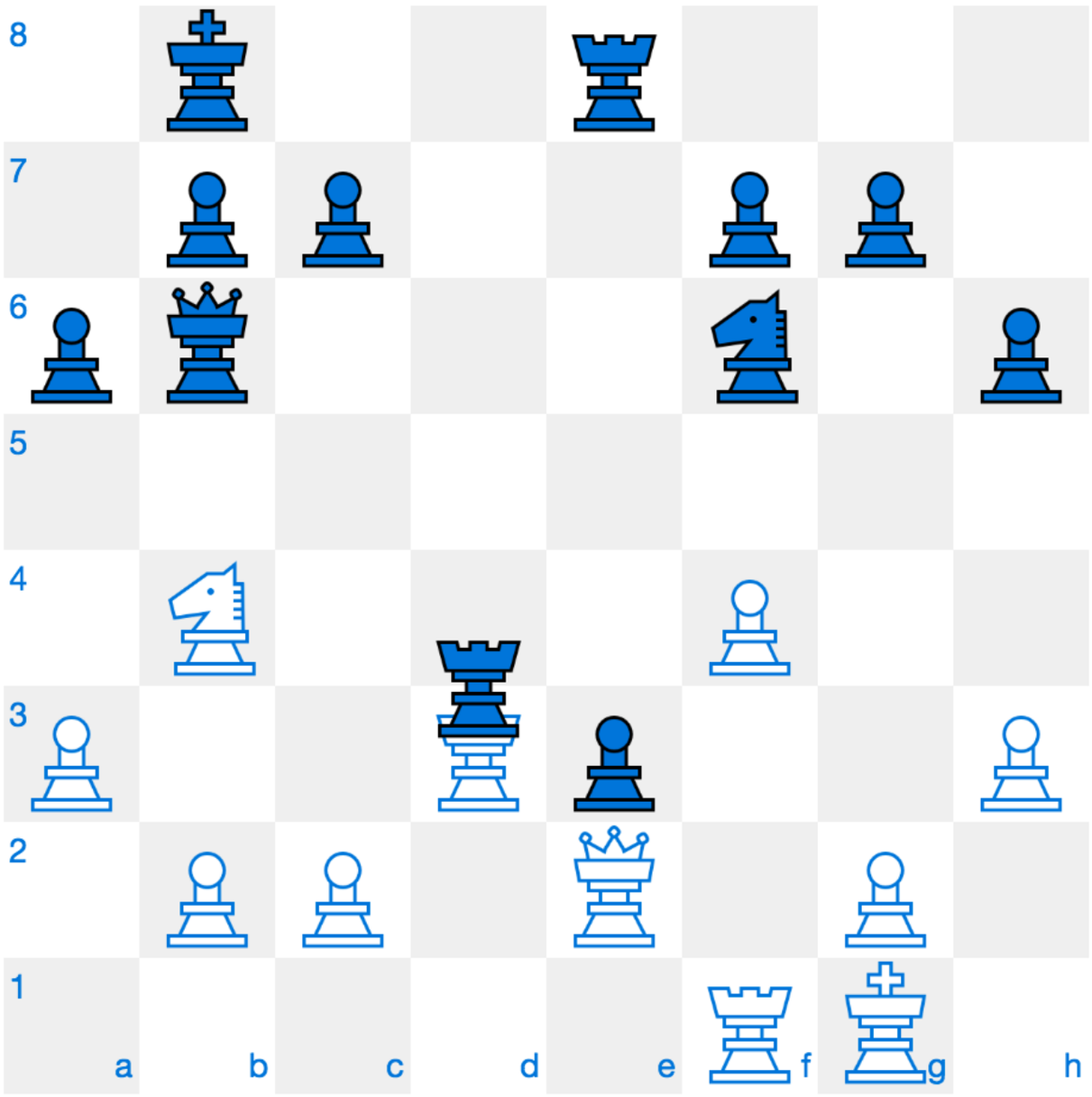
An extraordinarily optimizable, low-level subset of JavaScript, fast enough to be close to native speeds, even on mobile!

Asm.js



Asm.js Compilation & Execution Pipeline

Asm.js Mobile Performance



Winner Prediction

Asm.js

70.8%

PREDICTED CHANCE OF WIN

154

SCORE (IN CENTIPAWNS)

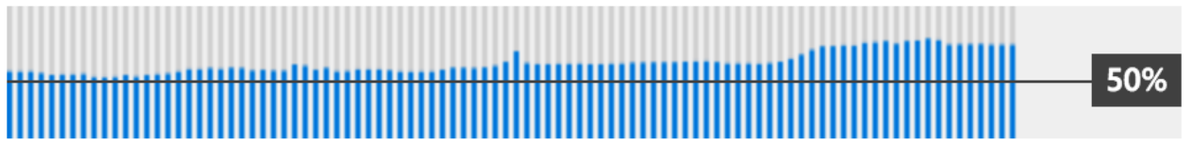
Not Optimized

30.5%

PREDICTED CHANCE OF WIN

-143

SCORE (IN CENTIPAWNS)



Captured Pieces

Captured Pieces

https://blogs.windows.com/msedgedev/2015/11/10/supercharging-javascript-performance-with-asm-js/

Optimising Mobile Performance

WebAssembly

The next step for asm.js

Demo	asm.js	binary	`gzip` asm.js	`gzip` binary
AngryBots	19MiB	6.3MiB	4.1MiB	3.0MiB
PlatformerGame	49MiB	18MiB	11MiB	7.3MiB

Eric Elliott
@_ericelliott

Follow

JS is safe. The ecosystem will thrive for many years. Wasm is about performance, diversity and innovation, not fragmentation. #JavaScript

Optimising Mobile Performance

Angry Bots Demo

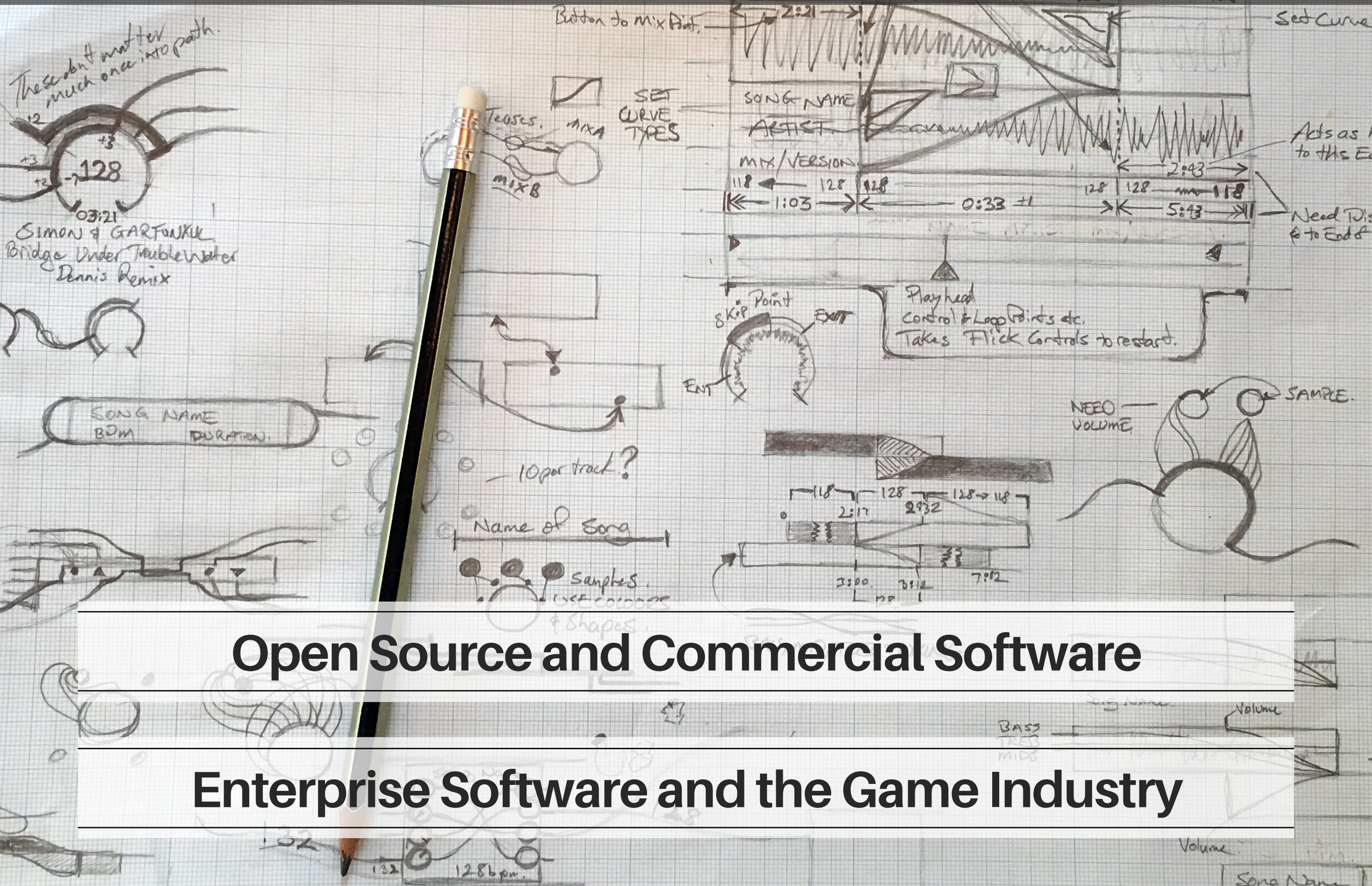
Uh-oh! WebAssembly isn't enabled in this browser. To get an early preview of this experimental technology, at your own risk:

- on Chrome Canary, open `chrome://flags/#enable-webassembly` and enable the switch.
- on Firefox Nightly, open `about:config` and set `javascript.options.wasm` to `true`.

See a preview of Microsoft Edge support and follow Safari support on WebKit's feature status page.

Play WebAssembly | Play asm.js fallback

ONE YEAR OPEN SOURCE EXPERIENCE



THANKS!

Please enjoy
your lunch