

H5 游戏调研

一、实现方案

远古时代：DOM+CSS+JS

之前做过一些小游戏。事实证明，效果一旦稍微复杂或者炫酷，在安卓机上都是会出现卡顿的。因为要频繁的去重排，重绘。

近现代：

1. CSS3

使用纯原生 CSS3 的 key-frames、translate、transition 高性能 CSS3 属性来进行游戏动画的实现。

优点：

可以轻松开启 3d GPU 硬件加速，直接跳过 JavaScript 解释器的性能瓶颈。移动平台上最流畅的动画实现方案。

缺点：

CSS3 的 transition 动画 ease 类型有限，这样使得模拟物理世界多变的动画变得困难，但是可以通过这个方案进行解决。

因为 CSS3 动画不通过 JavaScript 控制，无法对动画过程进行检测和逻辑操作，这使得物理世界的碰撞检测变得不可能。直接成为这个方案最大的硬伤，不排除有方案可以实现这种不可能，有待探索。

2. JavaScript + Canvas

使用 JavaScript 和 Canvas 进行动画的实现，由于移动平台的限制，基本不考虑 WebGL。使用

requestAnimationFrame + 逐帧渲染 Canvas 进行动画的实现。基本上游戏框架内部就采取这种实现，所

以这里不讨论游戏框架的使用，直接讨论底层性能优化。

优点：

所有动画都是 JS 控制，可以对动画进行最精细的调控。

基本上可以做到任意能够想到的效果。

缺点：

因为 Canvas 是像素级的渲染，每次动画的实现其实是对画布的“擦除 + 重新渲染”。对于移动平台来说会

有很大的计算压力，可能造成较大的性能瓶颈。无法启动 3d 底层 GPU 加速，不能压榨渲染性能。

3. JavaScript + CSS3

描述：

不使用 Canvas 进行游戏元素的渲染，而是使用 DOM 进行元素的实现。动画的实现采用这么一种方式。

用 CSS3 开启 3D 加速，使用 translate 进行元素位置的控制。使用 JS 控制 CSS3 的 translate 属性进行动画的实现。

优点：

既能使用 JavaScript 对动画进行精细控制，又能使用 3D 硬件加速

缺点：

无法像 Canvas 那样做出非常绚丽的效果。因为 JS 控制 CSS3 的方式实际操作的是 DOM，一点元素多

起来了，就会可能遇到一个非常大的性能瓶颈：DOM 天生就非常慢。游戏复杂程度高了，这种方案开发

效率会比较低。

综上，第一种方案虽然效果最好，因为遇到较大的实现困难，直接排除；方案 2 和方案 3 值得探索,理论

上来说 JS+CSS3 的模式是性能最好的，但是它满足不了只有 canvas 才能实现的效果

二、性能检测

根据目前流行的游戏框架来看。基本都是采用 JS+Canvas。故找了一个测试用例

<http://livoras.github.io/blog/ca-move/>;

1.测试数据：

移动端 HTML5 游戏和手机操作系统的关系：经过一系列在不同手机上同一浏览器的测试，同一手机上不

同浏览器的测试。得出的现象是：在低级操作系统（2.3）下用好的浏览器也可以达到比较好的 FPS，而在

高级操作系统（4.4）中用低级浏览器可以获取到一个比较低 FPS

同一台手机不同浏览器下

在 webkit build 为 537.36 的一个浏览器上获取到性能数据：FPS：100 左右

在 webkit build 为 604.5.6 的一个浏览器上获取到性能数据：FPS：60 左右

如果 canvas 中存在对象数量过多或者重绘 canvas 频率过大。在一些低版本浏览器中会出现卡顿现象。

FPS 明显降低。

综上所述，一个 H5 游戏的效果其实和设备操作系统的关系并不是特别大，而是和浏览器的版本息息相关

FPS 降低导致的结果：

1. 动画卡顿。用户体验变的极为糟糕
2. 在对战类小游戏中可能会导致两个玩家的画面不同步（FPS 导致运动相同的位置所花费的时间不同）

优化方法：

1. 利用 time-based 算法代替原始的 frame-based 算法，解决两个玩家画面不同步的问题，下面地址是一片关于解决方案的介绍 <http://blog.sklambert.com/using-time-based-animation-implement/>
2. 利用离屏渲染的方式去优化 canvas 部分重复绘制，清除工作所带来的性能损耗

三、游戏引擎

1. 以上说了很多，都是基于原生的 JS+canvas 模式的一些方案和优化。目前大多数的 H5 游戏都是基于游戏框架的。很多成熟的 H5 游戏框架已经把该做的优化都做在了框架内部。甚至有 H5 游戏引擎的性能测试已经超过了 native。

Engine	2D Render	3D Render	JavaScript	TypeScript
Three.js	NO	YES	YES	NO
Phaser	YES	NO	YES	YES
Pixi.js	YES	NO	YES	YES
Egret	YES	YES	YES	YES
enchant.js	YES	NO	YES	NO
craftyJS	YES	NO	YES	NO
Turbulenz	YES	YES	YES	YES
cocos2d-js	YES	NO	YES	NO
PlayCanvas	NO	YES	YES	NO
melonJS	YES	NO	YES	NO
Quintus	YES	NO	YES	NO
Hilo	YES	NO	YES	NO

四、总结

对于节奏大师这款游戏来说，H5 实现是完全可以的，性能方面主要取决于用户浏览器的选取。