



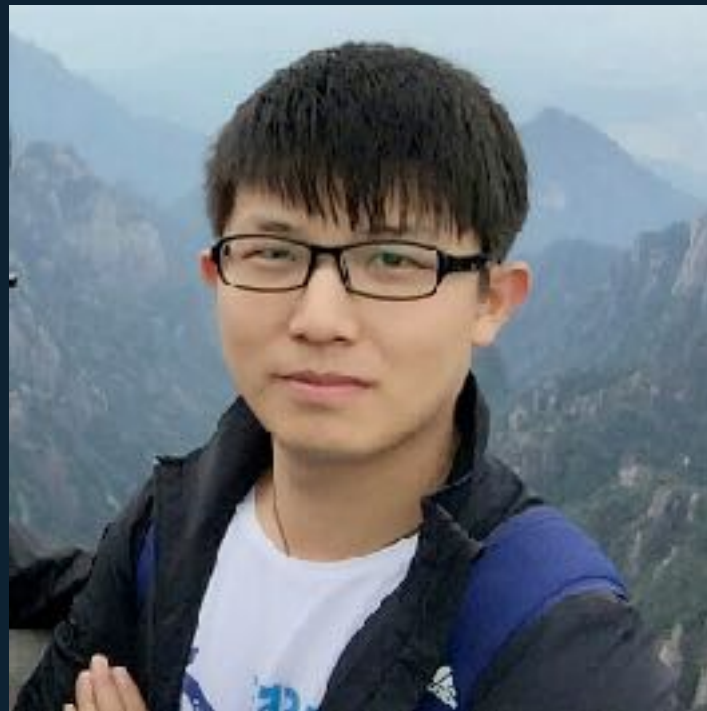
高效H5动画设计与性能优化

廖泽恺

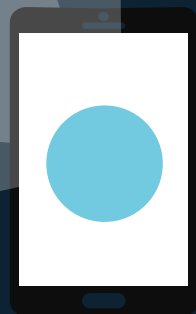


自我介绍

- 2015年加入AlloyTeam
- 参与过兴趣部落、群话题、Now直播
相关生态开发
- 专注移动前端性能优化，前端项目工
程化、组件化和自动化
- 喜欢旅游，乐于分享



目录



1

常用的动画方案

2

移动端屏幕适配

3

性能监控与优化

4

自动化与可配置



兴趣部落送礼



水边的阿拔拉


等级 23 积分 1000000

关注

大揭秘！C罗从娃娃变成男人，得感谢他的狼性教育！

2017-9-5 17:52 · 体育精英

刚出道时年轻的C罗除了极佳的天赋和刻苦的训练之外，一样非常优秀的是他比较叛逆。尤上是在对抗中，一旦对方比较强悍侵犯自己，C罗会像个布娃娃一样跌倒在地，向裁判抱怨投诉。



面对这个问题，C罗的父亲非常严厉地对他进行训练，因此从娃娃起，C罗就养成了这种强悍的意志力和不屈不挠的精神。正是这种精神，让C罗在足球场上成为了一个真正的战士。

点赞 68 评论 12



晨至

等级 23 积分 1000000

关注

著名物理学家发现“上帝”是存在的，而我们生活在矩阵之中

2017-6-5 21:33 · 科幻世界



一个世界著名的理论物理学家声称，他发现了证据表明“上帝”真的存在。他说：“我的结论是，我们生活在一个有智能生命创造的世界中。”

世界著名的理论物理学家Michio Kaku，他提出的弦理论，在科学界获得了广泛的认可，也提出了一个令人震惊的结论，证明了上帝是真实存在的。这是一个令人震惊的结论，证明了上帝是真实存在的。这是一个令人震惊的结论，证明了上帝是真实存在的。

点赞 12 评论 25

发现 热门 推荐



健身

积分 1000000

关注



NBA

积分 1000000

关注



科技

积分 1000000

关注

NBA新赛季开始了！快来看看你的主队表现吧！

4



QQ运动

积分 1000000

关注

点赞

5



肌肉男

积分 1000000

关注

点赞

6



减肥瘦身

积分 1000000

关注

点赞

7



詹姆斯

积分 1000000

关注

点赞



林俊杰

关注 95万 积分 94189

关注

发现 热门 推荐



周杰伦

积分 1000000

关注



周杰伦

积分 1000000

关注



周杰伦

积分 1000000

关注



周杰伦

积分 1000000

关注

周杰伦新歌《说好不哭》上线，全网热议！

4



周杰伦

积分 1000000

关注

点赞

5



周杰伦

积分 1000000

关注

点赞

6



周杰伦

积分 1000000

关注

点赞

7



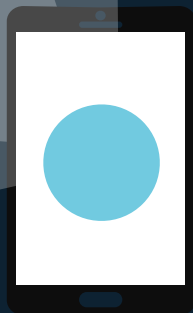
周杰伦

积分 1000000

关注

点赞

目录



1

常用的动画方案

2

移动端屏幕适配

3

性能监控与优化

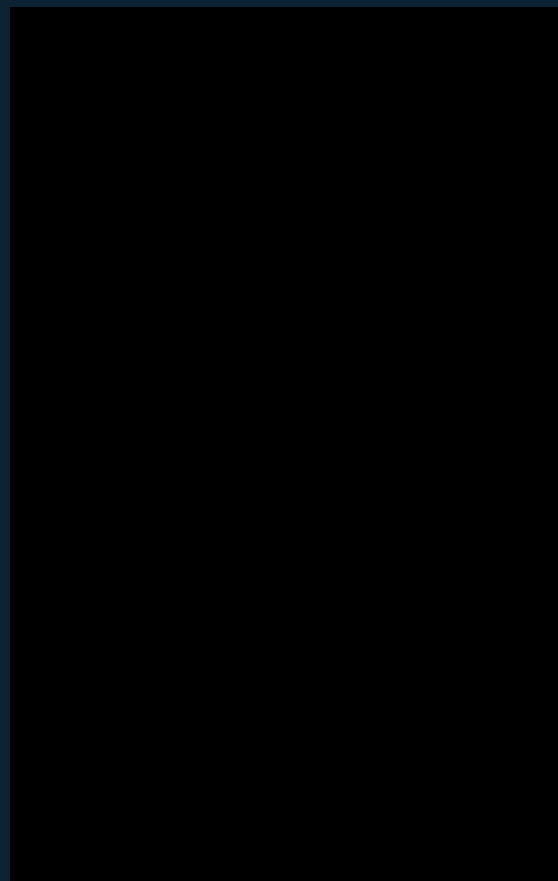
4

自动化与可配置



动画实现方式—GIF

- 一张图片资源
- 最多256种颜色
- 支持全透明，不支持半透明
- 不适用于“真彩色”— RGB
- 动画质量与资源体积成正比





动画实现方式—APNG

- Animated PNG一张图片
- 非官方认可
- 第一帧是PNG，后续是拓展块
- 支持度低

Animated PNG (APNG) - UNOFFICIAL

Like animated GIFs, but allowing 24-bit colors and alpha transparency

Current aligned Image relative Data relative Default

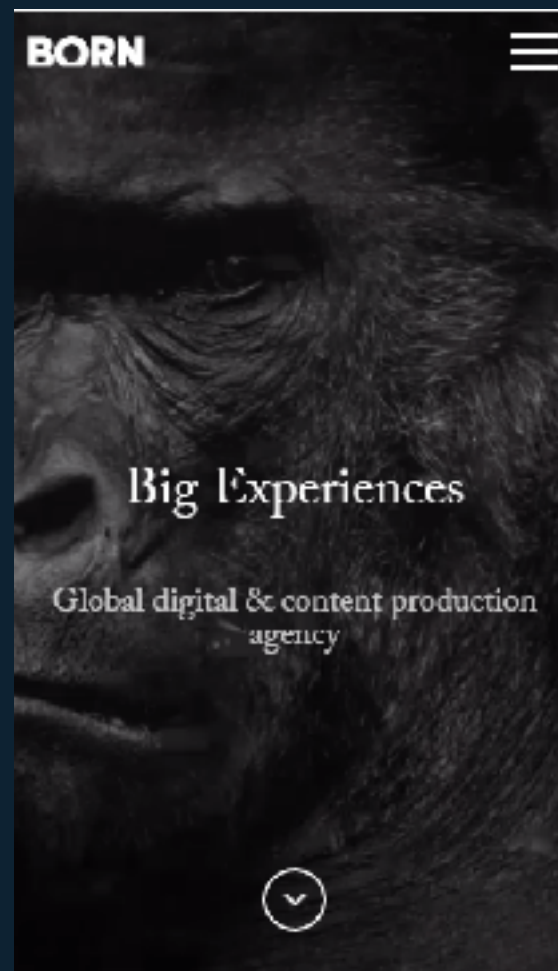
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
			49					4.4	
	14	54	59			10.2		4.4.4	
11	15	55	60	10.1	47	10.3	all	5.6	59
	16	56	61	11	48	11			
		57	62	12	49				
		58	63						



动画实现方式—Video

- 设置autoplay属性自动播放
- 设置loop属性控制循环
- 视频长度很重要（12s - 30s）
- 带宽是个问题

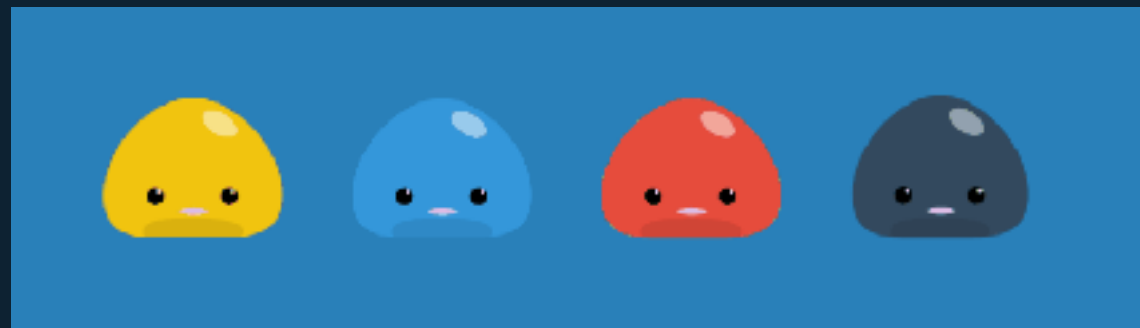
```
1 <video autoplay loop poster="demo.jpg" id="bg">
2   <source src="demo.webm" type="video/webm">
3   <source src="demo.mp4" type="video/mp4">
4   
5 </video>
```





动画实现方式—SVG

- 矢量无失真
- 难以呈现复杂位图效果
- 复杂动画具有渲染问题





动画实现方式—Canvas

- 可编程的画布
- 2D Context API
- 单个DOM元素，无状态
- WebGL





动画实现方式—JavaScript

- 使用JS控制DOM、CSS
- 自身维护时间流和动画速度

```
const box = document.querySelector('.box');  
box.style.width = 200 + 'px';  
box.style.opacity = 0.6;
```



动画实现方式—CSS3

- 过渡动画 (transition)
- 关键帧动画 (keyframes)
- 渲染引擎会使用跳帧保证动画的流畅性
- 渲染引擎会停止或降低不可见元素动画的刷新频率

```
webkit transition: webkit transform 200ms ease in out, opacity 200ms ease in out;  
transition: transform .2s ease-in-out, opacity 200ms ease-in-out;  
-webkit-transform: translate3d(0, 20px, 0);  
transform: translate4d(0, 20px, 0);
```

```
@-webkit-keyframes easeBigSmall {  
  0% {  
    transform: scale(1);  
  }  
  55% {  
    transform: scale(1.3);  
  }  
  83% {  
    transform: scale(.86);  
  }  
  100% {  
    transform: scale(1);  
  }  
}
```



总结


过渡动画

- Gif
- APNG
- Video
- Canvas
- JS
- SVG
- CSS


DOM元素

关键帧动画






兴趣部落送礼








林俊杰
歌手
 关注 95万 粉丝 94109
 歌手



关注


网页链接 >


跟上AIPL公司小分队，经验值绝对1001

点击在右侧APP中搜索

- 聊天室：13人正在讨论林俊杰
- 【直播】关于最近网络热议的警魂封号事件
- 「招募」林俊杰粉丝招募大家来应援吧
- 「和自己对话」林俊杰的深夜日记视频
- 本夜直播剪辑的规则



林俊杰
 歌手



「和自己对话」林俊杰 - 星球

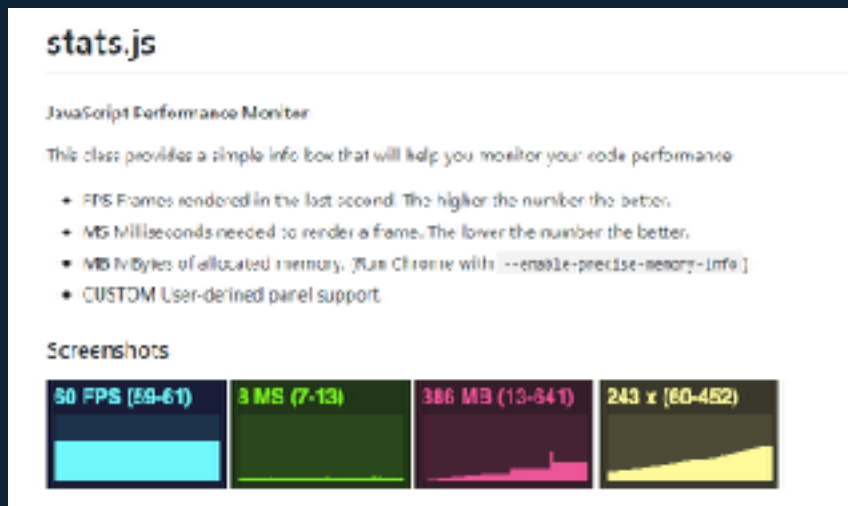


选择方案的标准—性能

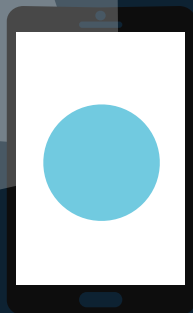


FPS工具选择

- stats.js: <https://github.com/mrdoob/stats.js/>



目录



1

常用的动画方案

2

移动端屏幕适配

3

性能监控与优化

4

自动化与可配置



常规页面与动画页面





常规页面与动画页面

- 屏幕越大，内容越多
- 滚动加载更多
- 内容和布局有条理、可重复
- 边距和控件大小不变



- 无论屏幕大小，同一场景在一块屏幕显示内容基本相同
- 所有内容全部显示无遮挡
- 页面元素以大量图片为主
- 元素之间具有关联性
- 不同尺寸，内容进行了缩放



适配方案一背景

1. 背景为纯色或渐变色，直接铺满
2. 背景为单张大图
3. 背景为可重复性纹理

```
.bg {  
  background-image: url('bg.png');  
  background-size: cover;  
  background-position: center;  
}
```

```
.bg {  
  background-image: url('texture.png');  
  background-repeat: repeat;  
}
```



适配方案一主体

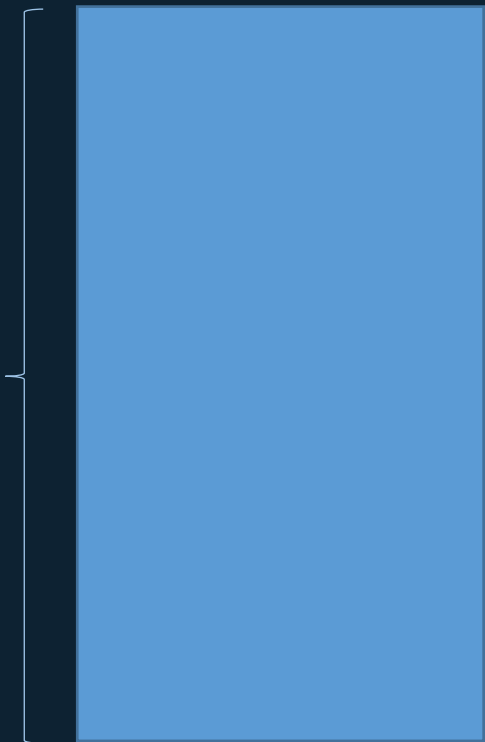
- 主体区域默认iPhone6基准尺寸
- 基于主体区定位的元素
- 只有缩放
- 默认屏幕居中缩放
- rem
- 主体区域如何界定？





适配方案一主体

标准尺寸



屏幕尺寸



屏幕尺寸



主体高度



适配方案一主体

```
const screenWidth = document.documentElement.clientWidth;
const screenHeight = document.documentElement.clientHeight;

const currentRatio = screenWidth / screenHeight;
const originRatio = 750 / 1200;

const wScale = screenWidth / 375;
const hScale = screenHeight / 600;

let eleScale, bgScale;

//如果屏幕宽高比大于标准宽高比
if (currentRatio > originRatio) {

    eleScale = hScale;
    bgScale = wScale >= 1 ? wScale : 1 / hScale;

} else { //如果屏幕宽高比小于标准宽高比

    eleScale = wScale;
    bgScale = hScale >= 1 ? hScale : 1 / wScale;

}
```

```
function scaleEle(selector, position) {
    $(selector).css({
        "webkit transform origin": position,
        "transform-origin": position,
        "-webkit-transform": "scale(" + eleScale + ")",
        "transform": "scale(" + eleScale + ")",
    });
}

scaleEle(".screen__content", "center");
```



适配方案一边缘元素

- 音乐图标、操作指引等主体无关元素
- 处于屏幕固定位置，基于屏幕定位

```
scaleEle(".logo","right top");  
  
scaleEle(".control-next","center bottom");
```



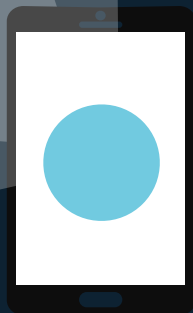


总结

- 背景、主体、边缘元素的适配
- 使用 transform 的 scale 缩放
- 使用 rem 全适配
- 方案仅供参考，根据场景、需求灵活运用



目录



1

常用的动画方案

2

移动端屏幕适配

3

性能监控与优化

4

自动化与可配置



性能监控与优化

考虑因素

- 内存消耗
- 帧率

推荐工具

- Stats.js
- Chrome developTool



硬件加速—GPU

- 避免频繁重绘和重排
- 避免 width 和 height, 使用 transform: scale(x)
- 避免 margin、top / left, 使用 transform: translate(x, y)
- 元素创建后, 立即设置 transform: translateZ(0) 或 translate3d(0,0,0)
- 元素会使用独立的 layer 参与渲染



- TextureMerger:





墨守成规并不是万能—性能

	LG G3 (Android 5.0.1)	MI 2S (Android 5.0.1)	iPhone 6P (iOS 9.3.2)
CSS (background)	7 FPS (4-9) 	32 FPS (3-42) 	60 FPS (48-61)
CSS (translate)	27 FPS (4-43) 	61 FPS (38-65) 	59 FPS (48-60)
JS+CSS (background)	17 FPS (3-19) 	39 FPS (8-43) 	59 FPS (46-60)
JS+CSS (translate)	28 FPS (1-48) 	46 FPS (45-65) 	60 FPS (40-60)
canvas	42 FPS (3-49) 	64 FPS (63-65) 	60 FPS (46-60)

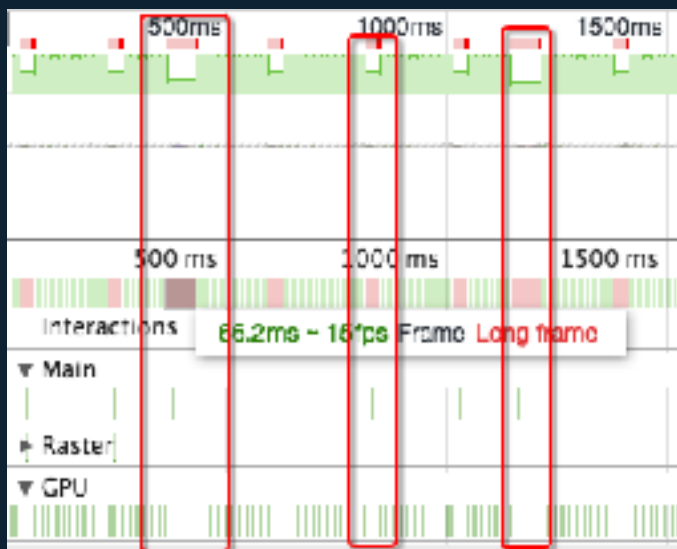
- translate 应该明显优于 background
- 锯齿 → 卡顿明显
- 图片太大?

	大图 30080*460	小图 1000*100
CSS (background)	7 FPS (4-9) 	41 FPS (29-48)
CSS (translate)	27 FPS (4-43) 	57 FPS (43-58)



性能调试—**timeline**

锯齿是如何产生的？



- FPS降低时，GPU罢工
- 大图 → 丢帧
- 大图烧显卡！



帧动画图片—性能

帧动画图片组到底要不要合图？

- 无须合图：HTTP2.0
- 雪碧图：空白区域

一个动画大概40~50帧图片，单张图片要
500px，

合并图：4000px * 3000px

是否有更好的方案？



px *



帧动画优化方案—zip压缩包

zip压缩包可行吗？

- 资源打包，减少请求数
- 避免单个大图渲染瓶颈
- 前端解压？ jszip <https://stuk.github.io/jszip/>
- jszip解析出ArrayBuffer, base64 ? Blob
- URL.createObjectURL(blob)



帧动画优化方案—zip压缩包

```
export async function downloadZip(url) {
  let data = await new Promise(function(rs, rj) {
    JSZipUtils.getBinaryContent(url, function(err, result) {
      if (err) {
        throw err;
      }
      rs(result);
    });
  });

  let zip = await JSZip.loadAsync(data);

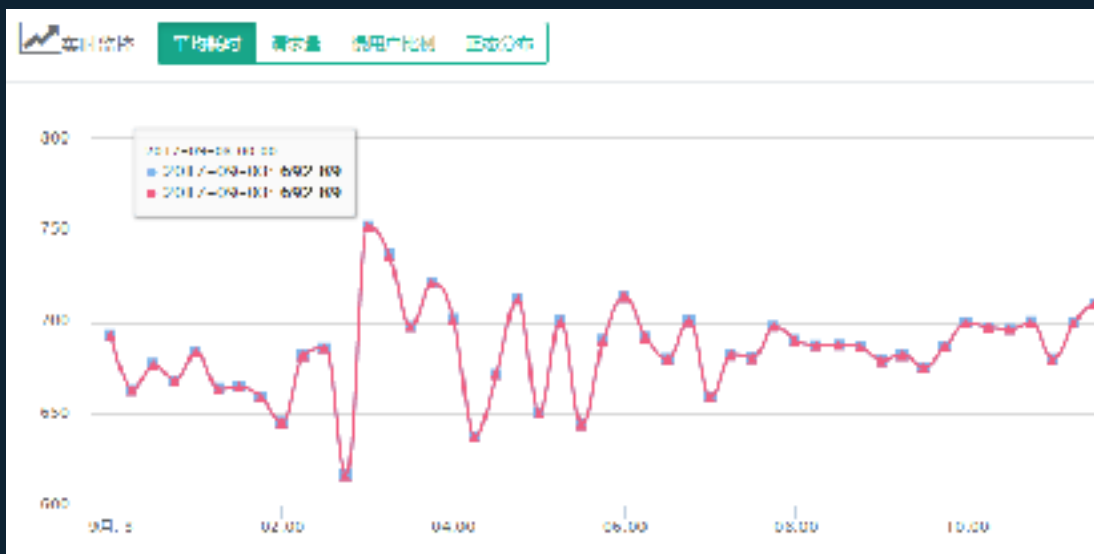
  let re = /\.(jpg|.png|.gif|.ps|.jpeg)$/;
  let result = await Promise.all(
    Object.keys(zip.files)
      .filter(fileName => re.test(fileName.toLowerCase()))
      .map(async function (fileName) {
        let file = zip.files[fileName];
        let blob = await file.async('blob');

        return [
          fileName,
          blob // create an url. img.src = URL.createObjectURL(blob)
        ];
      })
  );
};
```



帧动画优化方案—zip压缩包

zip压缩包拉取并解析耗时



还能更快点吗?



帧动画优化方案—zip压缩包

- zip动画资源CDN部署，max-age
- 前端缓存

	LocalStorage	webSQL	indexedDB	H5Data
iOS	5MB	60MB	磁盘剩余空间 50%	~
Android	10MB	60MB	磁盘剩余空间 50%	~



帧动画优化方案—indexedDB

IndexedDB R-C

Method of storing data client-side, allows indexed database queries.

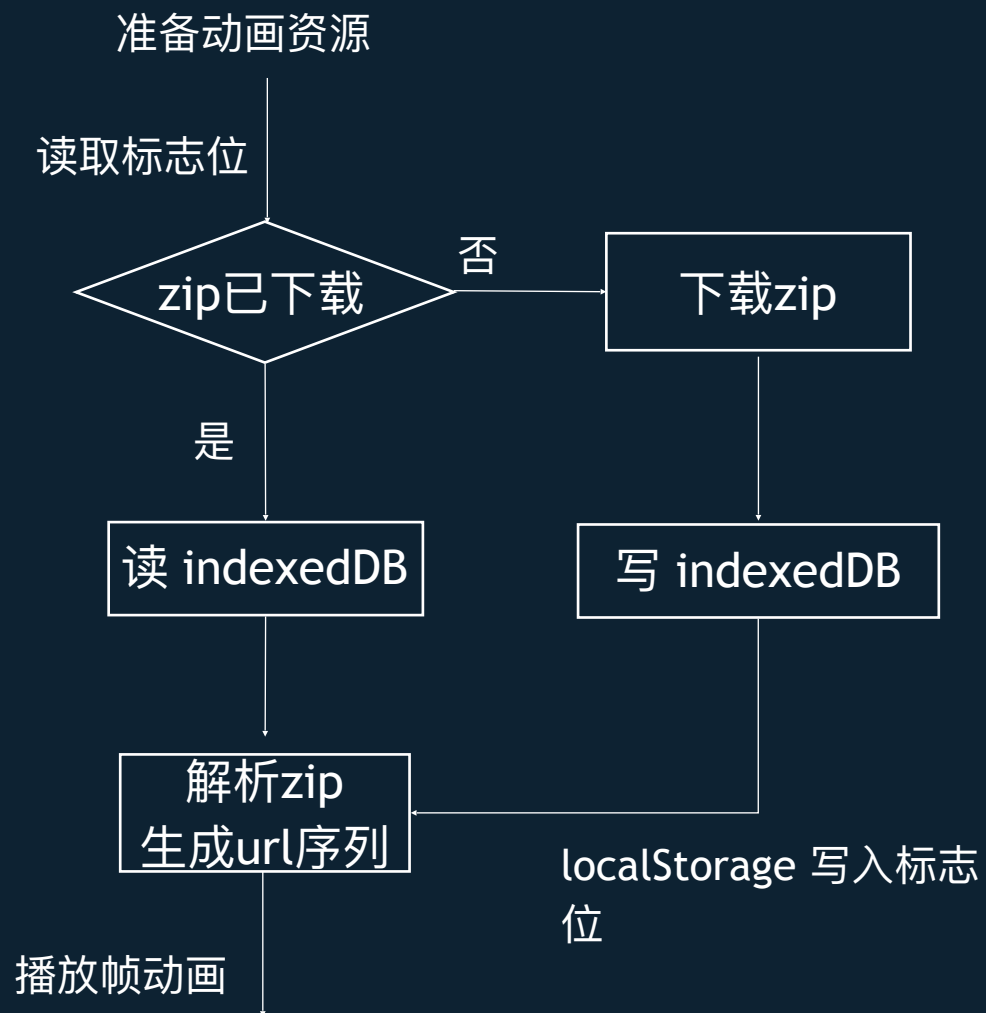
Global 87.28% + 5.82% = 93.1%
unprefixed 87.24% + 5.66% = 92.91%

Current | Aligned | Usage relative | Data relative | Show all

IF	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
			49					4.4	
	14	54	59			10.2		4.4.4	
11	15	55	60	10.1	47	10.3	al	50	59
	16	56	61	11	48	11			
		57	62	1P	49				
		58	63						



帧动画优化方案—indexedDB





帧动画优化方案—indexedDB






帧动画优化方案—zip压缩包

还有可提升空间吗？



帧动画优化方案—indexedDB

Web Workers  - is Global 94.48%

Method of running scripts in the background, isolated from the web page

IE	edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49			10.2			
		55	60	10.1	47	10.3		4.4	
11	15	56	61	11	48	11	all	56	61
	16	57	62	TP	49				
		58	63		50				
		59	64						



硬件加速—will-change

- 提前通知浏览器元素哪些属性会发生动画
- 浏览器预优化
- 不能过多滥用

```
.hand {  
    will-change: transform;  
}  
  
.hand {  
    transform: scale(2);  
}
```



兼容性问题—CSS3动画控制

- 使用setTimeout做CSS3动画结束后回调？误差大！
- CSS3 Animation提供了动画事件 `animationStart` & `animationEnd`
- 部分机器不支持
- 降级使用JavaScript实现精准控制

```
this.num.addEventListener('webkitAnimationEnd', () => {  
  console.log('num animation end');  
  this.group.classList.add('animating');  
  setTimeout(() => {  
    this.props.hide();  
  }, GIFTLASTTIME);  
});
```



兼容性问题—FPS控制

- setTimeout / setInterval, 执行时不精确
- 使用 requestAnimationFrame
- 与浏览器刷新频率保持一致
- iOS全部支持
- Android支持率约 89%

如果限定帧率怎么办？



RAF—限定帧率

```
draw(timestamp) {  
  !this.start && (this.start = timestamp);  
  
  // 如果没有限定帧率, 使用 RAF 自有发挥  
  let raf = this.fps === 60 ? true : (timestamp - this.start > 1000 / this.fps);  
  
  if (raf) {  
    this.ctx.clearRect(0, 0, this.width, this.height);  
    const f = this.frameArray[this.currentFrame];  
    const {x, y} = f;  
    if (this.singleMode) {  
      this.ctx.drawImage(this._img, x, y, this.width, this.height, (this.scr
```

- 与设计师沟通保证60FPS
- 关注资源包体积



RAF不支持—polyfill

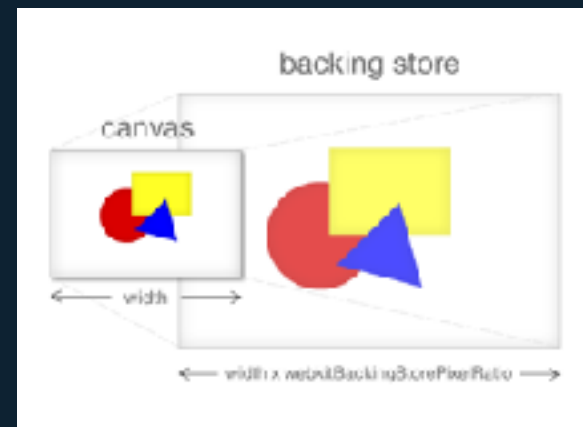
- 手Q X5 全支持
- Android4.4及其他

```
(function() {  
  var lastTime = 0;  
  var vendors = ['webkit', 'moz'];  
  for(var x = 0; x < vendors.length && !window.requestAnimationFrame; ++x) {  
    window.requestAnimationFrame = window[vendors[x]+'RequestAnimationFrame'];  
    window.cancelAnimationFrame =  
      window[vendors[x]+'CancelAnimationFrame'] || window[vendors[x]+'CancelRequestAnimationFrame'];  
  }  
  
  if (!window.requestAnimationFrame)  
    window.requestAnimationFrame = function(callback, element) {  
      var currTime = new Date().getTime();  
      var timeToCall = Math.max(0, 16 - (currTime - lastTime));  
      var id = window.setTimeout(function() { callback(currTime + timeToCall); },  
        timeToCall);  
      lastTime = currTime + timeToCall;  
      return id;  
    };  
  
  if (!window.cancelAnimationFrame)  
    window.cancelAnimationFrame = function(id) {  
      clearTimeout(id);  
    };  
})();
```



兼容性问题—canvas Retina适配

- 500px * 500px 的原图画到 250px * 250px 的画布上，模糊了！
- 罪魁祸首：设备像素比 `window.devicePixelRatio`
- 逻辑像素
- 画在 500px * 500px 的 Canvas上再 `scale(0.5)` ？





兼容性问题—canvas Retina适配

```
const canvas = opts.canvas,  
      context = canvas.getContext('2d'),  
      oldWidth = canvas.width;  
      oldHeight = canvas.height;  
      ratio = window.devicePixelRatio || 1;  
  
canvas.width = oldWidth * ratio;  
canvas.height = oldHeight * ratio;  
  
canvas.style.width = oldWidth + 'px';  
canvas.style.height = oldHeight + 'px';  
  
context.scale(ratio, ratio);  
  
context.drawImage();
```



性能评分

- 原因：

机器性能低，内存紧张，帧率低

- 原理：

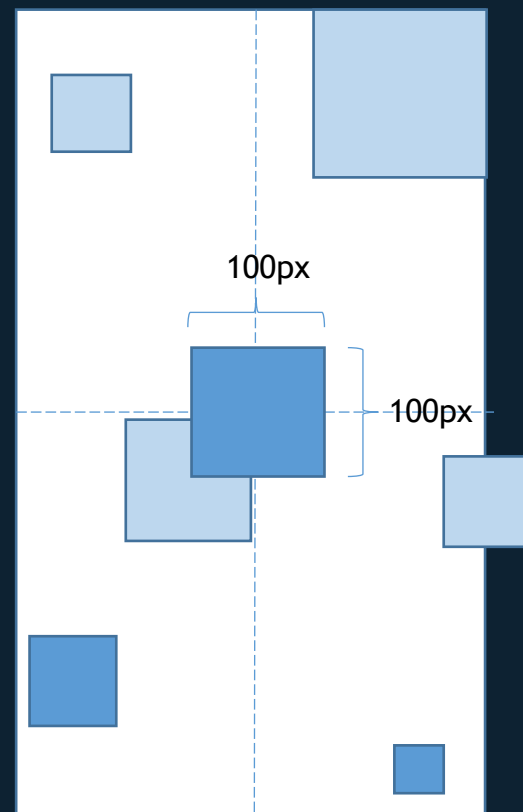
页面中创建一定数量的DOM，并添加transform和opacity变化。

根据帧率衡量机器性能



性能评分一方法

- 在页面中创建200个正方形DIV元素。
- 每个DIV宽度和高度均为100px，DIV初始位置的中心点与页面的中心点重合
- 设置时间点 t_1 和 t_2
- 在每一帧周期内对页面中每个DIV进行随机位移、缩放和透明度变换





性能评分—方法

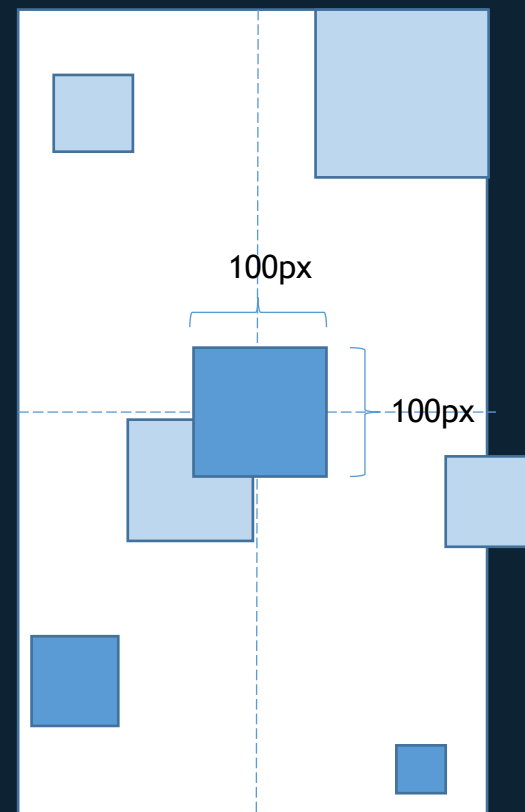
- 开始时间 t_1 ，结束时间 t_2 ，帧数为 n

$$\text{fps} = 1000 * n / t$$

- 取帧率的10倍为性能评分

$$\text{score} = \text{floor}(10000 * n / t)$$

- 分级



iPhone 7 / vivo X7 / 小米6	iPhone5s	小米4	红米1
400+	300 - 400	200 - 300	0 - 200



性能评分—措施

- 减少帧数（图片数）
- 减少装饰及动画复杂度
- 取消特效
- 使用 @1x 图片

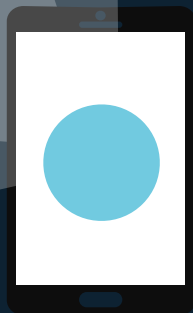




性能优化总结

- GPU硬件加速
- CSS: will-change
- 图片压缩、图片合并
- zip压缩包 + indexedDB + WebWorker
- requestAnimationFrame
- 性能评分
- FPS调试
- Timeline调试

目录



1

常用的动画方案

2

移动端屏幕适配

3

性能监控与优化

4

自动化与可配置



自动化与可配置

- 产品频繁新增 / 更换礼物
- 动画效果可配置?
- 定义统一配置文件
- 动画资源由 packageID 唯一标志
- 礼物配置与动画资源分离

```
▼ {date: "2017-02-09 13:26:10",...}
  date: "2017-02-09 13:26:10"
  ▼ items_coinGift: [{goodsWord: "没毛病", gifUrl:
    ▼ 0: {goodsWord: "没毛病", gifUrl: "http://pub.:
      animateType: 2
      gifUrl: "http://pub.idqqimg.com/pc/misc/gro
      giftPrice: 8
      goodsWord: "没毛病"
      label: "弹幕"
      likeCount: 16
      packageID: "91002"
      playRule: 4
      pngUrl: "http://pub.idqqimg.com/pc/misc/gro
    ► 1: {goodsWord: "什么鬼", gifUrl: "http://pub.:
    ► 2: {goodsWord: "我觉得ok", gifUrl: "http://pul
    ► 3: {goodsWord: "这波很强", gifUrl: "http://put
```



自动化与可配置



产品经理





“常规操作”—五连绝世

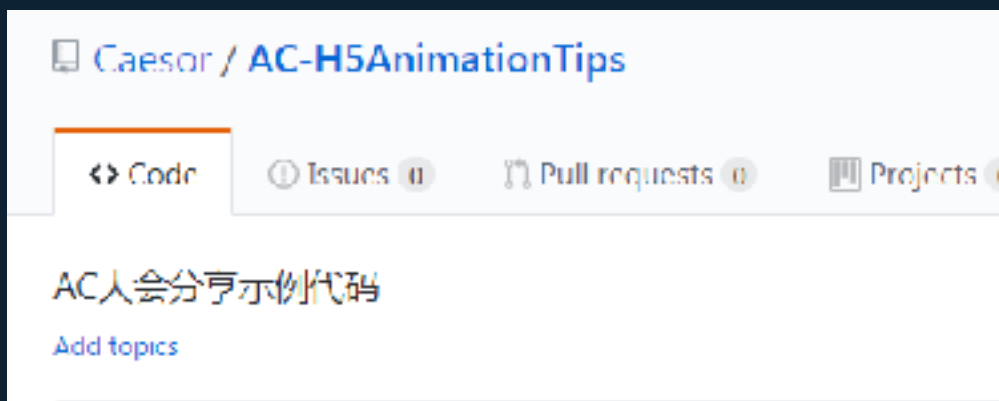


- 礼物“面板” 懒加载
- 礼物“配置” 预加载
- 礼物面板“图片” 预加载
- 面板呼起时送礼“动画资源”预加载
- 送礼“动画资源” 缓存indexedDB





课程资源



- <https://github.com/Caesor/AC-H5AnimationTips>



Thank You

Q & A