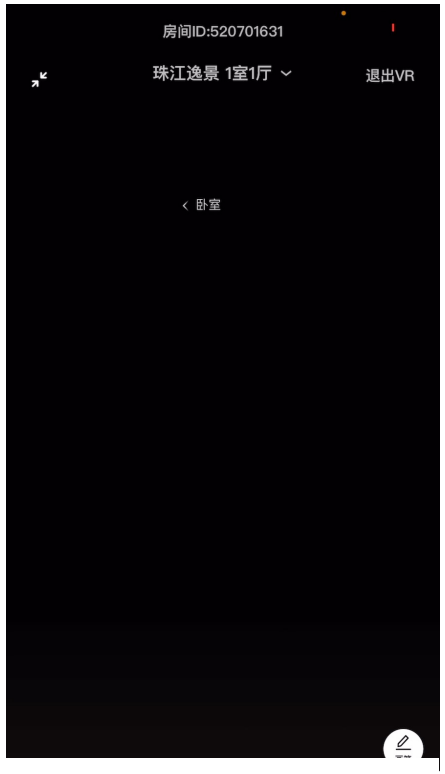
如视VR性能优化的那些事儿

李 阳 | 贝壳·如视

大家对如视 **VR** 有哪些坏印象？

① 加载慢 -- "卡"

② 耗电 + "发热"

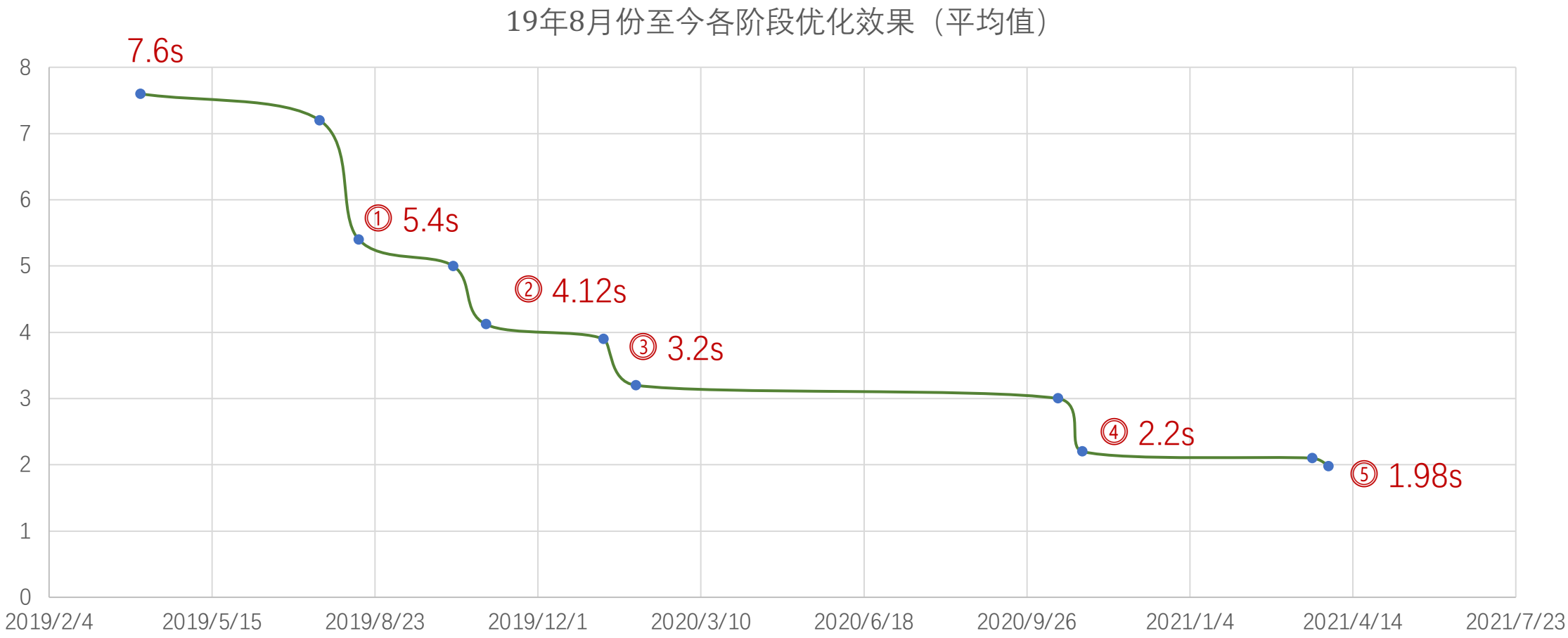


③ "黑屏"

目录

一：首屏加载耗时优化

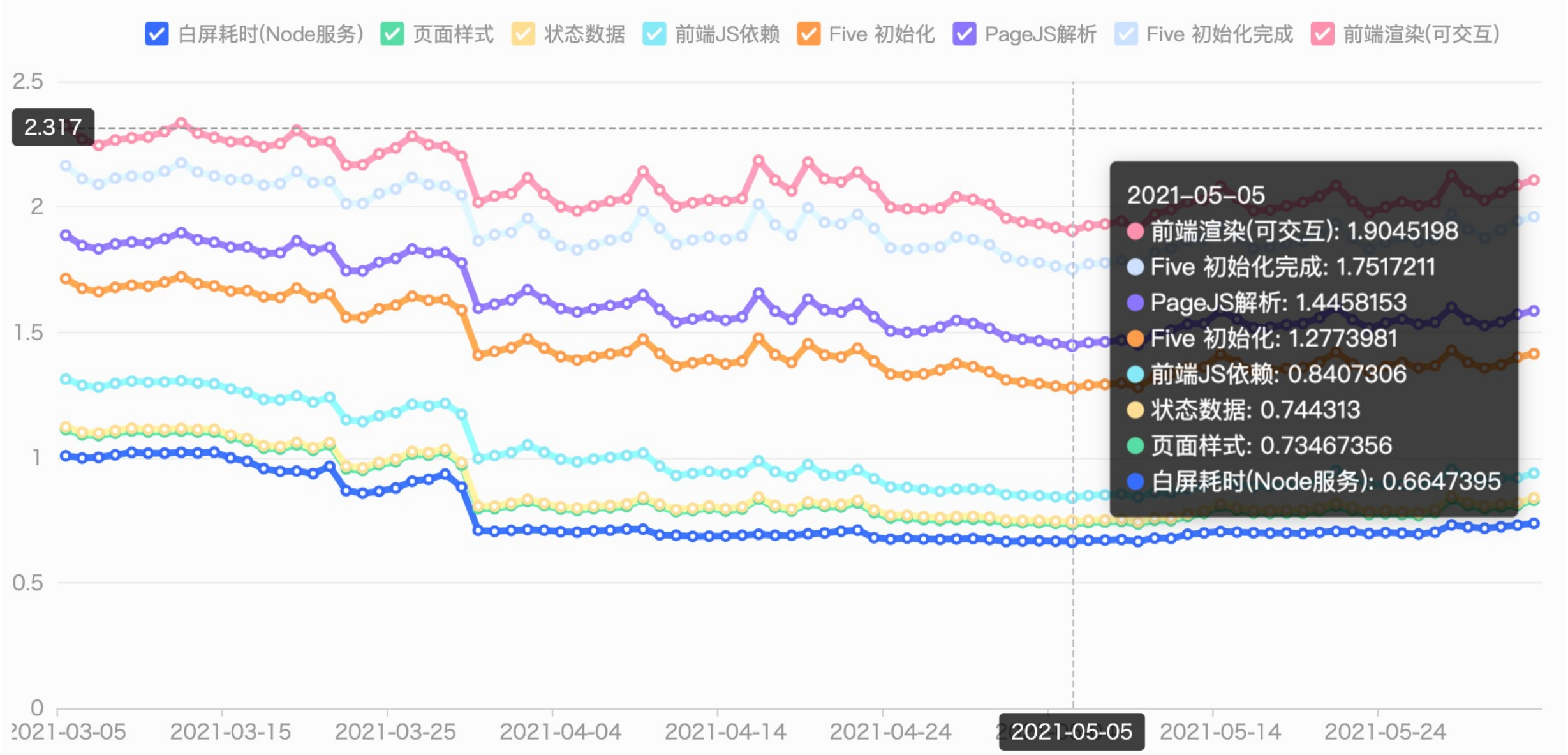
首屏加载优化


19年8月份至今各阶段优化效果（平均值）
7.6s
① 5.4s
② 4.12s
③ 3.2s
④ 2.2s
⑤ 1.98s
8
7
6
5
4
3
2
1
0
2019/2/4
2019/5/15
2019/8/23
2019/12/1
2020/3/10
2020/6/18
2020/9/26
2021/1/4
2021/4/14
2021/7/23

首屏加载优化 —— 关键节点

- ① 5.4s 降低首屏阶段 HTTP 请求数量
- ② 4.12s "懒"加载: 非首屏内容异步加载或 触发加载
- ③ 3.2s 客户端 请求代理 + 缓存；HTTP2
- ④ 2.2s VR 3.0 架构升级，将 ①、② 做得更加彻底
- ⑤ 1.98s 后端 API HTTP 请求由1s 优化至 0.67s

- ✓ **减少 HTTP 请求**
- ✓ **提升 HTTP 请求效率**
- ✓ **在合适的时机做正确的事情**


白屏耗时(Node服务)
页面样式
状态数据
前端JS依赖
Five 初始化
PageJS解析
Five 初始化完成
前端渲染(可交互)
2.5
2.317
2
1.5
1
0.5
0
2021-03-05
2021-03-15
2021-03-25
2021-04-04
2021-04-14
2021-04-24
2021-05-05
2021-05-14
2021-05-24
2021-05-05
前端渲染(可交互): 1.9045198
Five 初始化完成: 1.7517211
PageJS解析: 1.4458153
Five 初始化: 1.2773981
前端JS依赖: 0.8407306
状态数据: 0.744313
页面样式: 0.73467356
白屏耗时(Node服务): 0.6647395

二：耗电 & 发热

耗电 + "发热" —— WHY

打开VR后，**掉电**很快，手机**发烫**

都是JS动画惹得祸：每一帧都在 计算、重绘

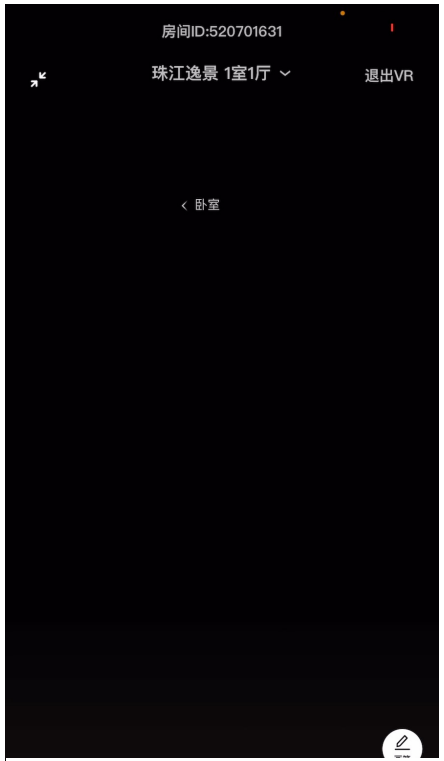
耗电 + "发热" —— FIX

```
requestAnimationFrame(() => {
  /*
    animate()
    - 计算
    - 重绘
   */
})
```

JavaScript 动画：每一帧都在计算、重绘


```
requestAnimationFrame(() => {

  mesh.needsRender = true
})
```

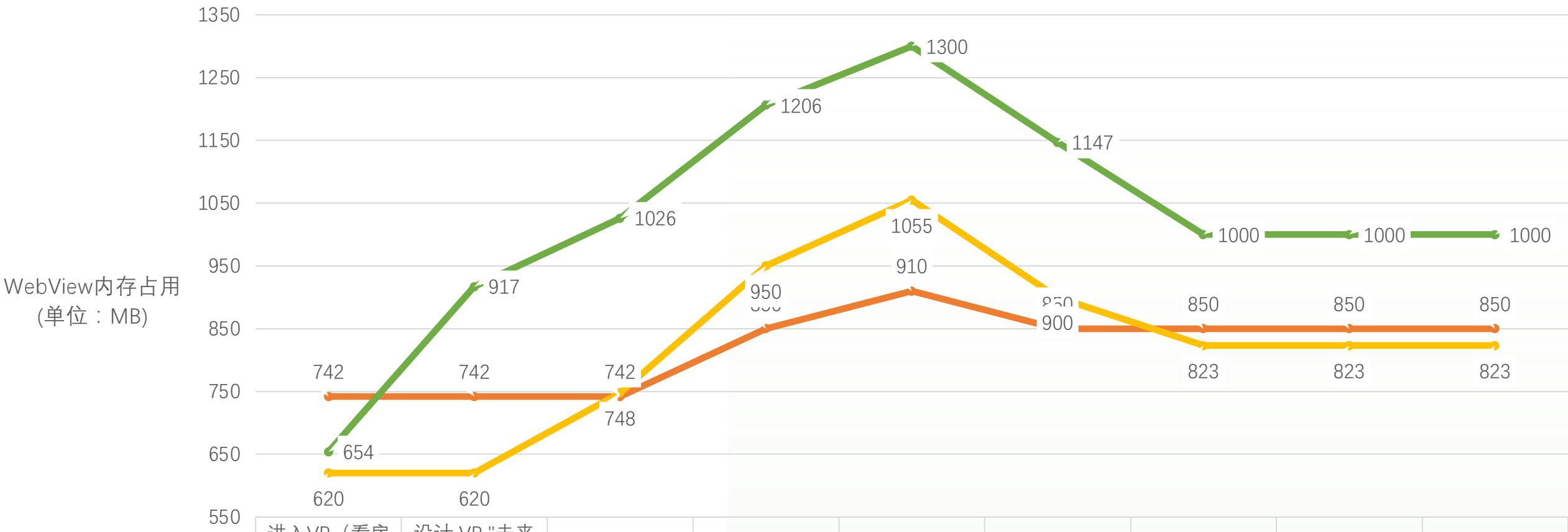
- ✓ 局部绘制: 规避全局重绘,
- ✓ 降低动画频率 或 及时终止 "适可而止"



"黑屏"

三：内存溢出

内存溢出 —— 黑屏、白屏及闪退

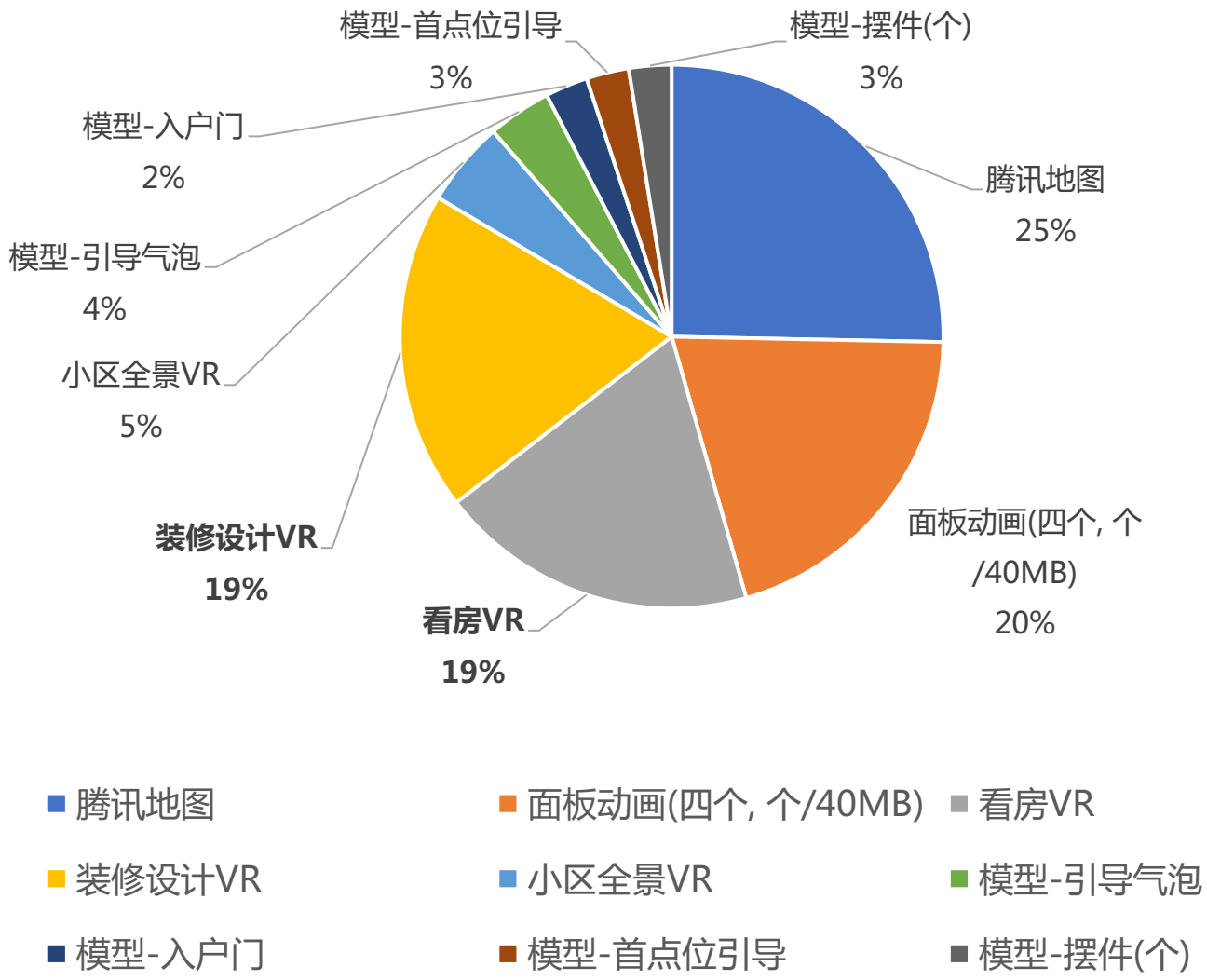


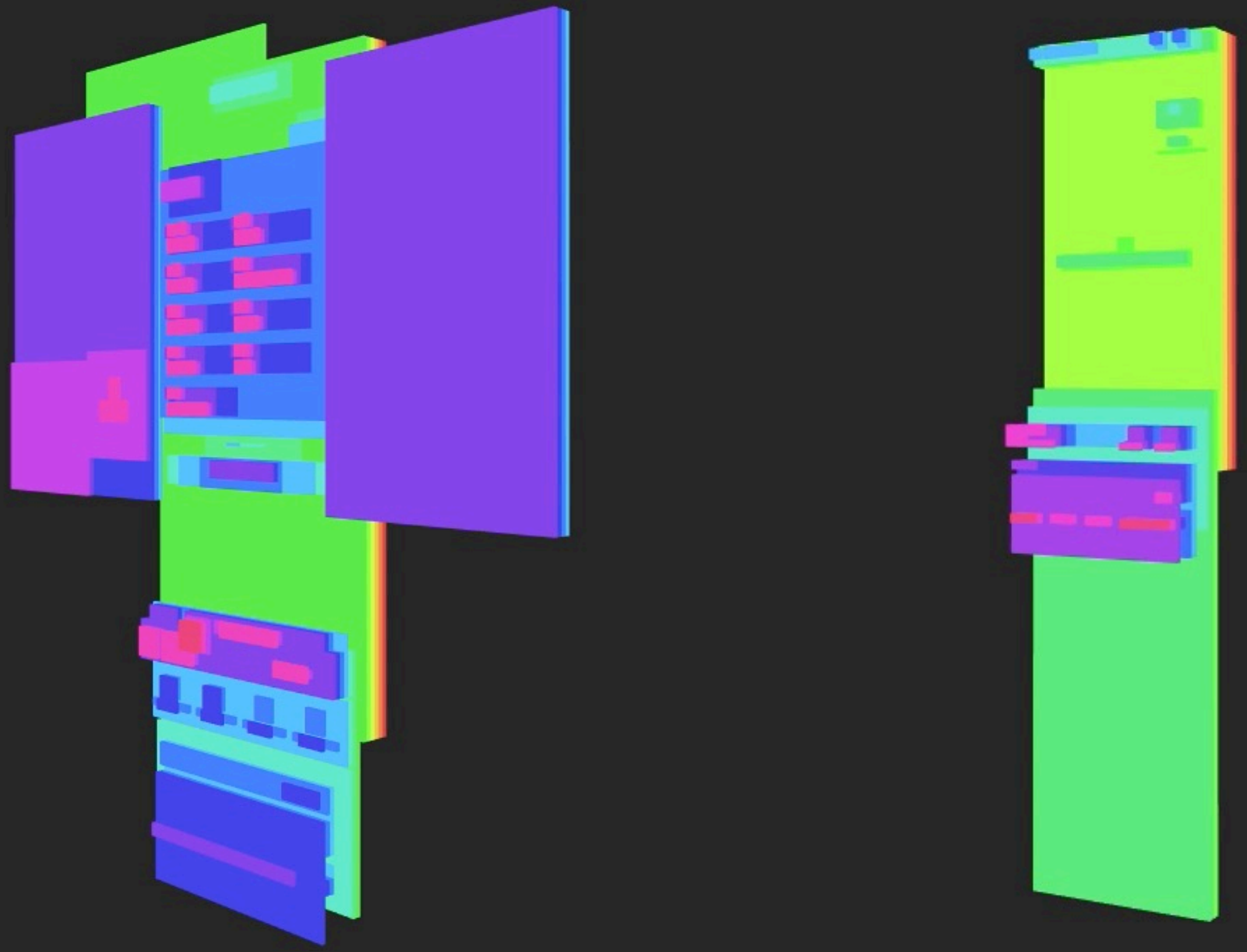
	进入VR（看房VR）	设计 VR "未来家"	切换面板	开启地图	地图交互(缩放)	关闭地图	等待内存回收	无交互	无交互
2.0 看房VR	742	742	742	850	910	850	850	850	850
3.0 看房VR	620	620	748	950	1055	900	823	823	823
3.0 看房VR + "未来家"	654	917	1026	1206	1300	1147	1000	1000	1000

以 iPhone 12 Mini 测试数据，

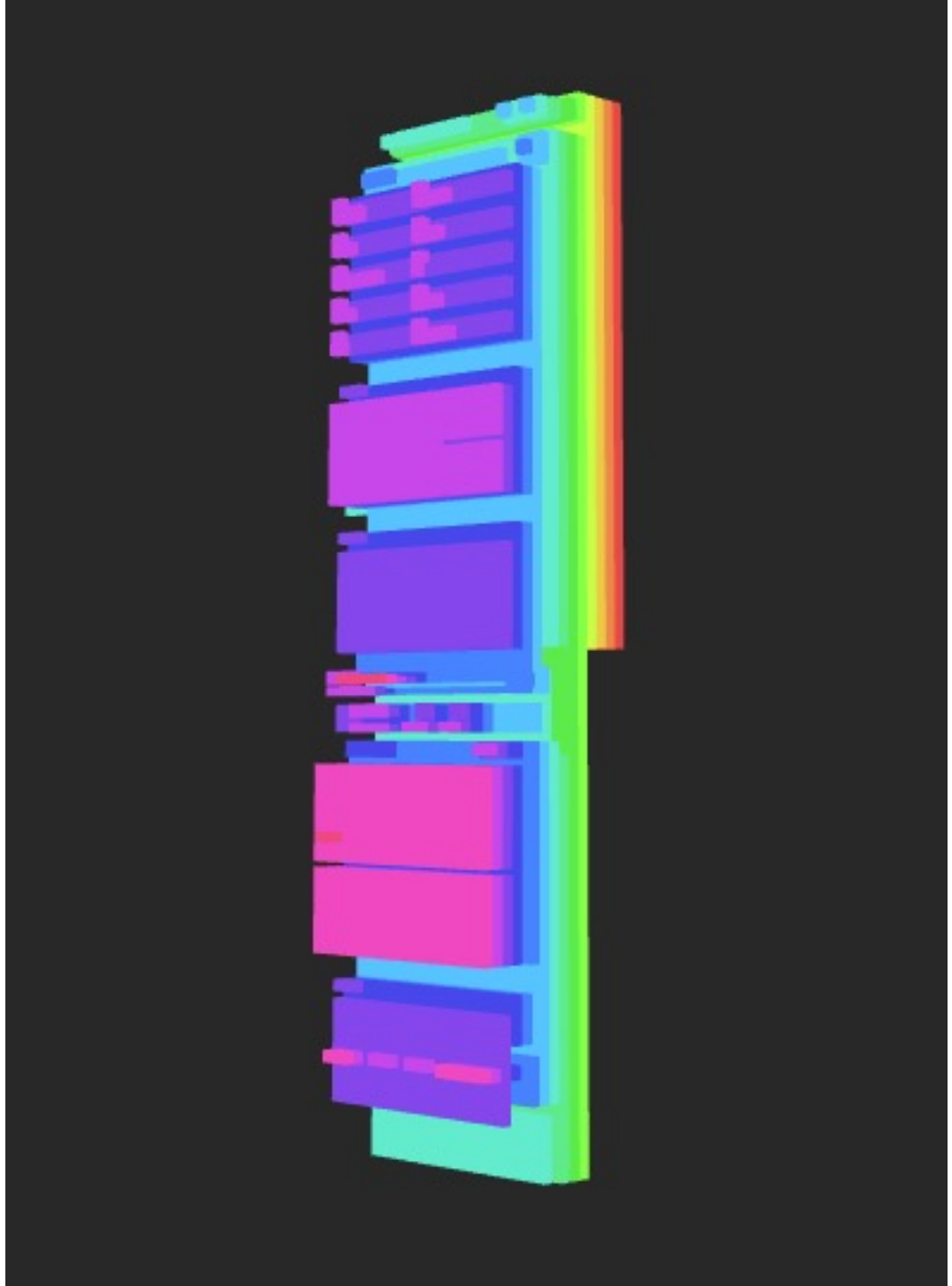
维持在 1000MB左右，

iOS 崩溃阈值 1.5G






12:32
奥体玫瑰园B区 3室2厅
房源信息
售价
约150万
单价
11573元/㎡
户型
3室2厅
面积
129.62㎡
挂牌
2020年10月01日
朝向
南
年代
2015年
楼型
暂无数据
城市
盐城
楼层
高楼层/26层
小区图集
查看更多
周边配套



一个 VR 3D 模型 在 iPhone 上居然占用 300+ MB 内存？

一个 VR 3.0 带装修的页面居然 占用 920+ MB 内存？？？

四：图片内存占用计算

图片内存占用 —— 计算方式

ARGB_8888

A - alpha 透明 8bit(位)

R - Red 8bit(位)

G - Green 8bit(位)

B - Blue 8bit(位)

2048*2048

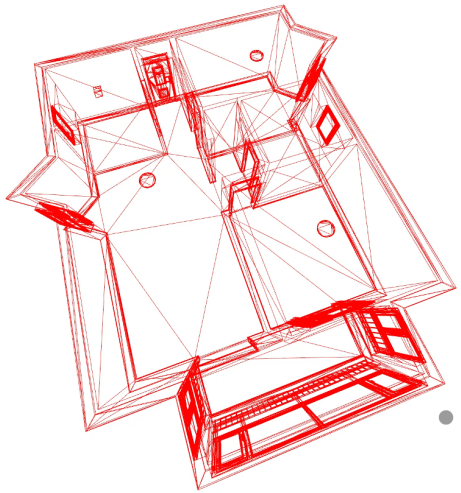
WebView 渲染图片采用格式

(2048*2048*4) / (1024*1024) = 16 MB

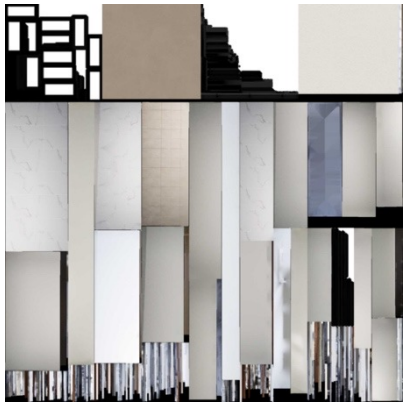
立方体全景：6 张 2048*2048 图片

UV 贴图：12 张 512*512 图片

=



+



一个 VR 实例图片理论上计算的内存占用：

$$
\begin{aligned}
&(\\
&\quad (2048*2048*4) *6 + \quad \text{立方体全景} \\
&\quad (512*512*4)*12 \quad \text{uv 贴图} \\
&) / (1024*1024) = 108 \text{ MB}
\end{aligned}
$$

一个 VR 3D 模型 在 iPhone 上居然占用 300+ MB 内存，

貌似还差很多？？？

iPhone 设备显示屏像素信息表

机型	逻辑像素	渲染像素	物理像素	DPR	一像素对应几个字节	2048*2048图片内存占用
iPhone Xs / 11 Pro Max	414*896	1242*2688	1242*2688	3	4 * 3个字节	48MB
iPhone XR/11	414*896	828*1792	828*1792	2	4 * 2个字节	32MB
iPhone X/Xs/11 Pro	375*812	1125*2436	1125*2436	3	4 * 3个字节	48MB
iPhone 12/12 Pro	390*844	1170*2532	1170*2532	3	4 * 3个字节	48MB
iPhone 12 Pro Max	428*926	1284*2778	1284*2778	3	4 * 3个字节	48MB
iPhone 12 mini	375*812	1125*2436	1080*2340	3	4 * 2.88个字节	46.08MB
iPhone 6/6s/7/8/SE2	375*667	750*1334	750*1334	2	4 * 2个字节	32MB
iPhone 6/6s/7/8/ Plus	414*736	1242*2208	1080*1920	3	4 * 2.61个字节	41.76MB
iPhone 5/5C/5s/SE	320*568	640*1136	640*1136	2	4 * 2个字节	32MB
iPhone 4/4s	320*480	640*960	640*960	2	4 * 2个字节	32MB
iPhone 3G/3Gs	320*480	320*480	320*480	1	4个字节	16MB

- ✓ 物理像素: 硬件真实的像素
- ✓ 渲染像素: 操作系统抽象的像素
- ✓ 逻辑像素: 前端/UI 使用的像素

结论 ⇨

从iPhone 4代开始，iPhone 屏幕的物理分辨率是很高的，除了 "iPhone 6/6s/7/8/ Plus" 和 "iPhone 12 mini" 设备之外，**iOS 系统基本是把2个或3个物理像素当作1个逻辑像素来使用的（放大到2~3倍数）**。

PC

((2048*2048*4) *6 +

(512*512*4)*12)

/ (1024*1024)

= 108 MB

iPhone 6s (2015)

((2048*2048*4)*6 +

(512*512*4)*12) * **2**

/ (1024*1024)

= 216 MB

iPhone 12 Pro Max (2020)

((2048*2048*4)*6 +

(512*512*4)*12) * **3**

/ (1024*1024)

= 324 MB

一个 VR 实例 (看房 VR)

PC

((2048*2048*4) *6 +

(512*512*4)*12) * **2 * 2**

/ (1024*1024)

= 216 MB

iPhone 6s (2015)

((2048*2048*4)*6 +

(512*512*4)*12) * **2 * 2**

/ (1024*1024)

= 432 MB

iPhone 12 Pro Max (2020)

((2048*2048*4)*6 +

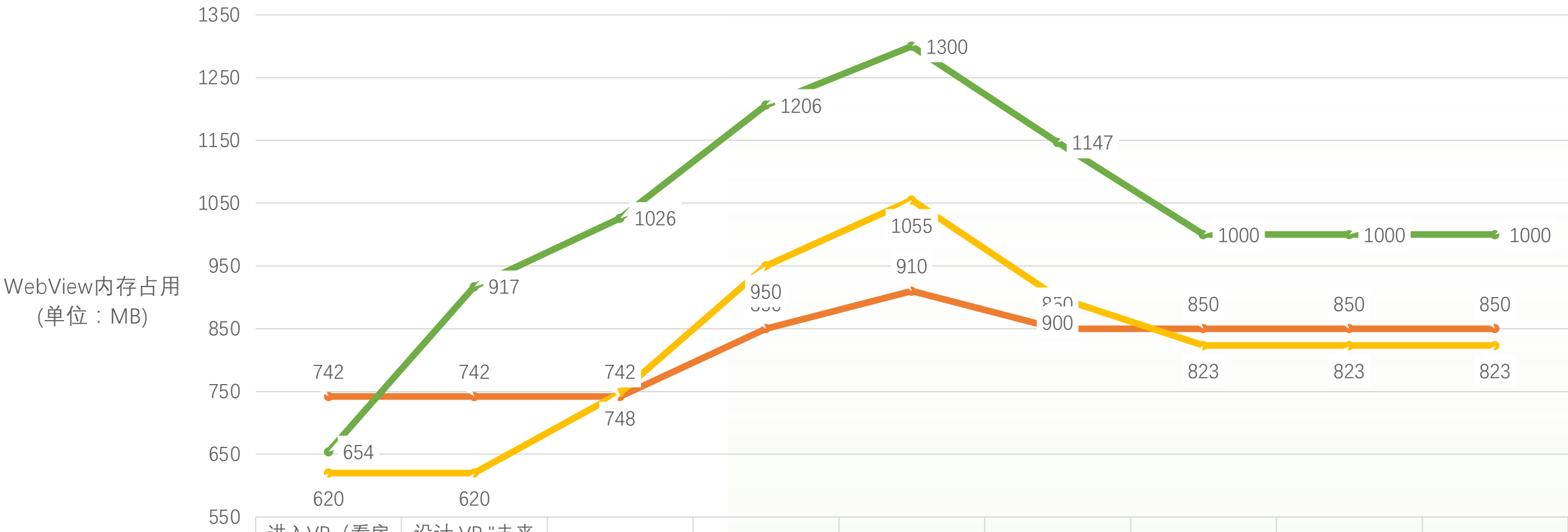
(512*512*4)*12) * **3 * 2**

/ (1024*1024)

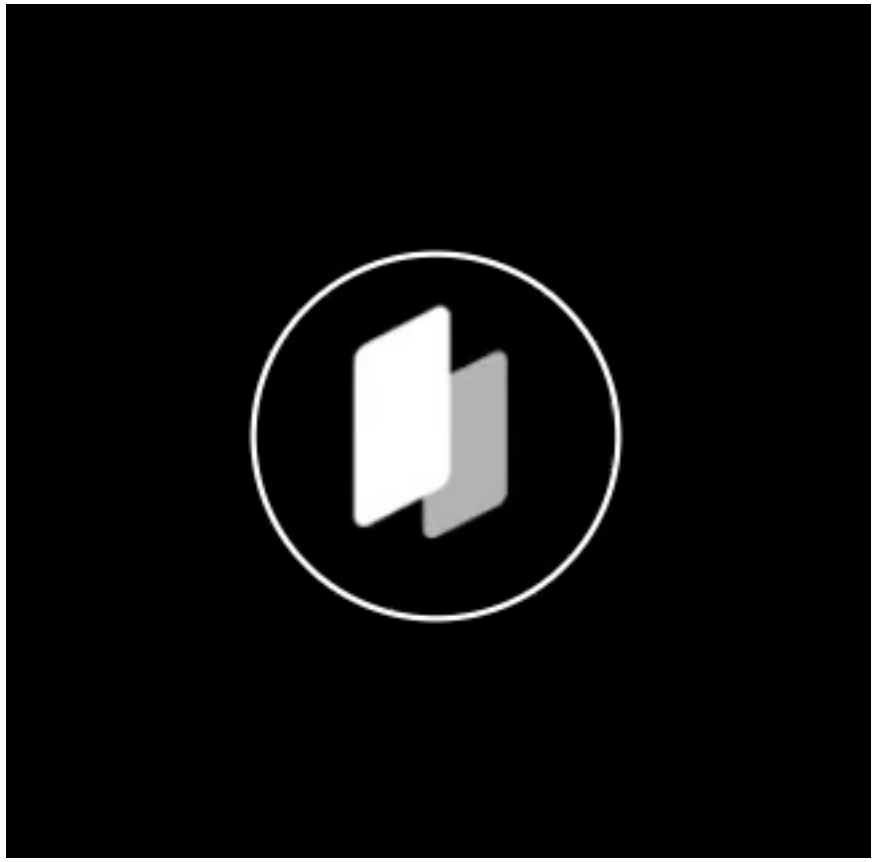
= 648 MB

两个 VR 实例 (看房 VR + "未来家" 装修设计 VR)

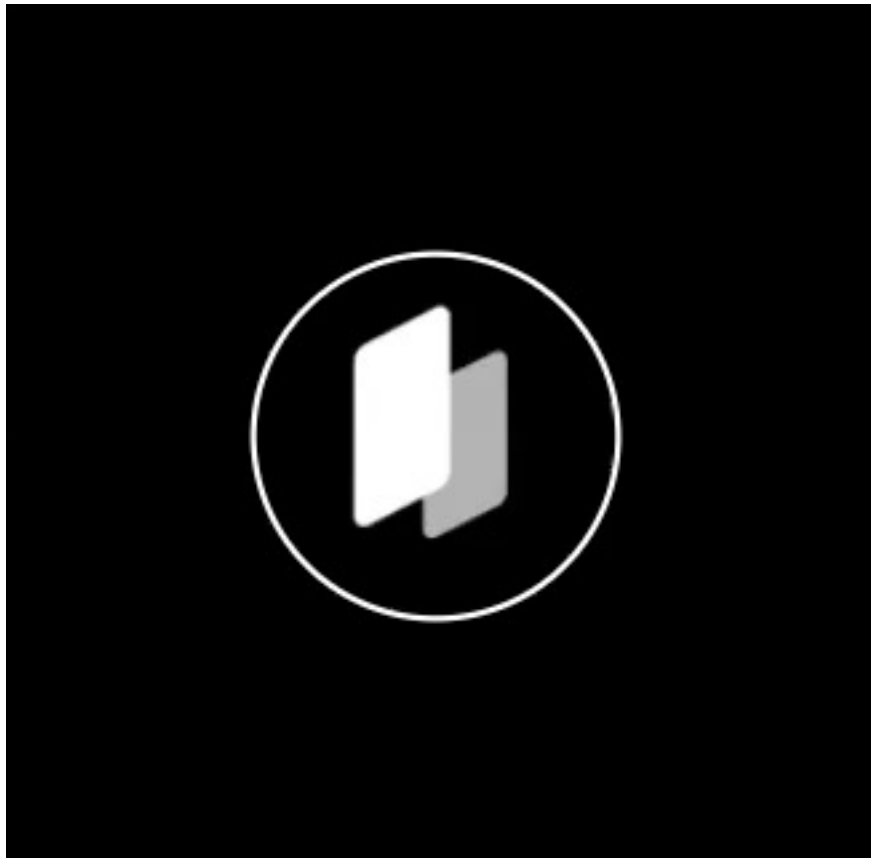
内存溢出 —— 黑屏、白屏及闪退



	进入VR（看房VR）	设计 VR "未来家"	切换面板	开启地图	地图交互(缩放)	关闭地图	等待内存回收	无交互	无交互
2.0 看房VR	742	742	742	850	910	850	850	850	850
3.0 看房VR	620	620	748	950	1055	900	823	823	823
3.0 看房VR + "未来家"	654	917	1026	1206	1300	1147	1000	1000	1000



五：CSS3 序列帧内存



五：CSS3 序列帧内存

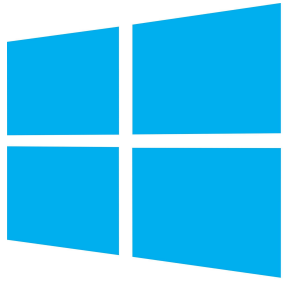
```
@keyframes logo-sprites {

  0% {
    background-position: 0 0;
  }

  100% {
    background-position: 13800px 0;
  }

}

animation: logo-sprites 2.208s 0s
steps(53) infinite normal;
```

序列帧（53帧）雪碧图分辨率 14065*265：

(14065*265*4)

/ (1024*1024) = 14.22 MB

理论上计算的内存占用



PC

iPhone 6s (2015)

iPhone 12 Pro Max (2020)

14.22 *** 3** = 42.66 MB

14.22 * 2 *** 3** = 85.32 MB

14.22 * 3 *** 3** = 127.98 MB

PerfDog 统计总是计算的 **3** 倍值。

目前的猜测是CSS 3 逐帧动画本质上是个补间动画，用在帧动画中，需要上一帧、当前帧、下一帧 来计算补间动画，同时需要三张图片，所以可能会同时存在三张图片实例。

逐帧动画慎用，帧数最好限制在24帧以内，占用内存不要超过20MB。

谢谢、