



# 游戏中的性能优化技巧

## UWA 优化日 - 上海站

侑虎科技（上海）有限公司

2016.7.31

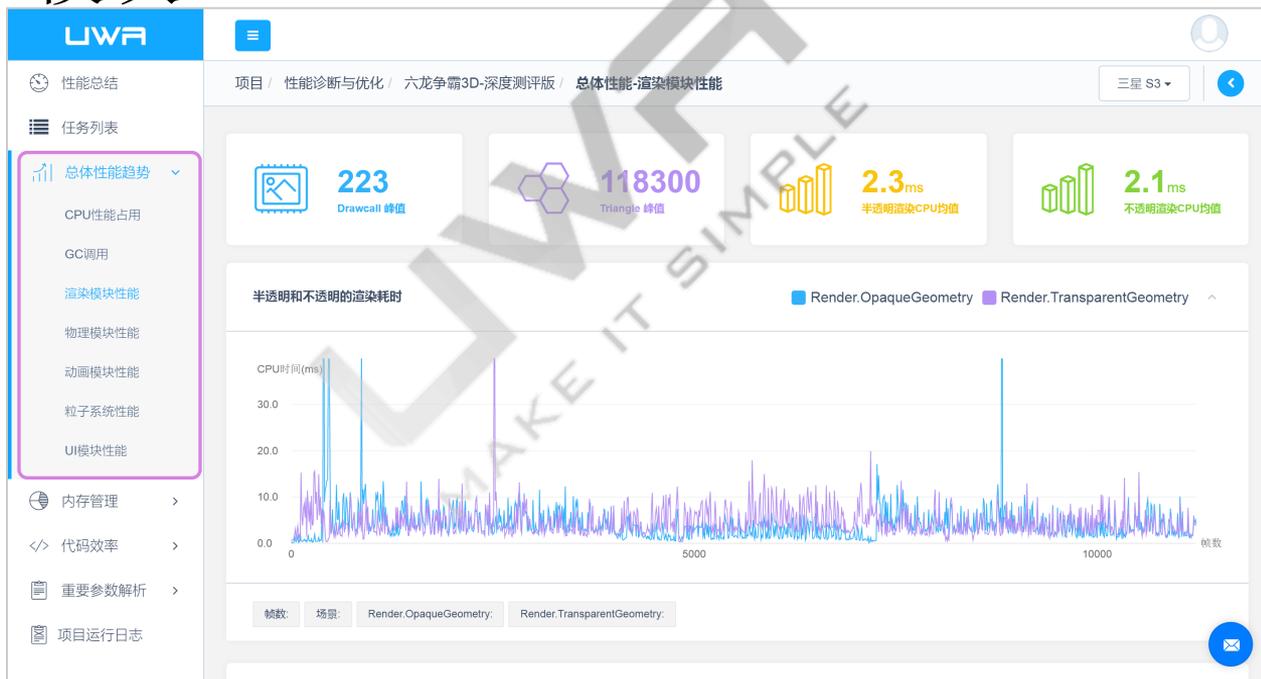
# 优化是一个永恒不变的话题

- 优化是游戏从平凡到优秀的必经之路
  - 无论硬件设备多么先进、无论开发团队经验多么丰富
  - 玩家的需求和项目的要求永远在不停增长
- **VR应用（游戏）更是如此**

因此，我们开发了**UWA**！

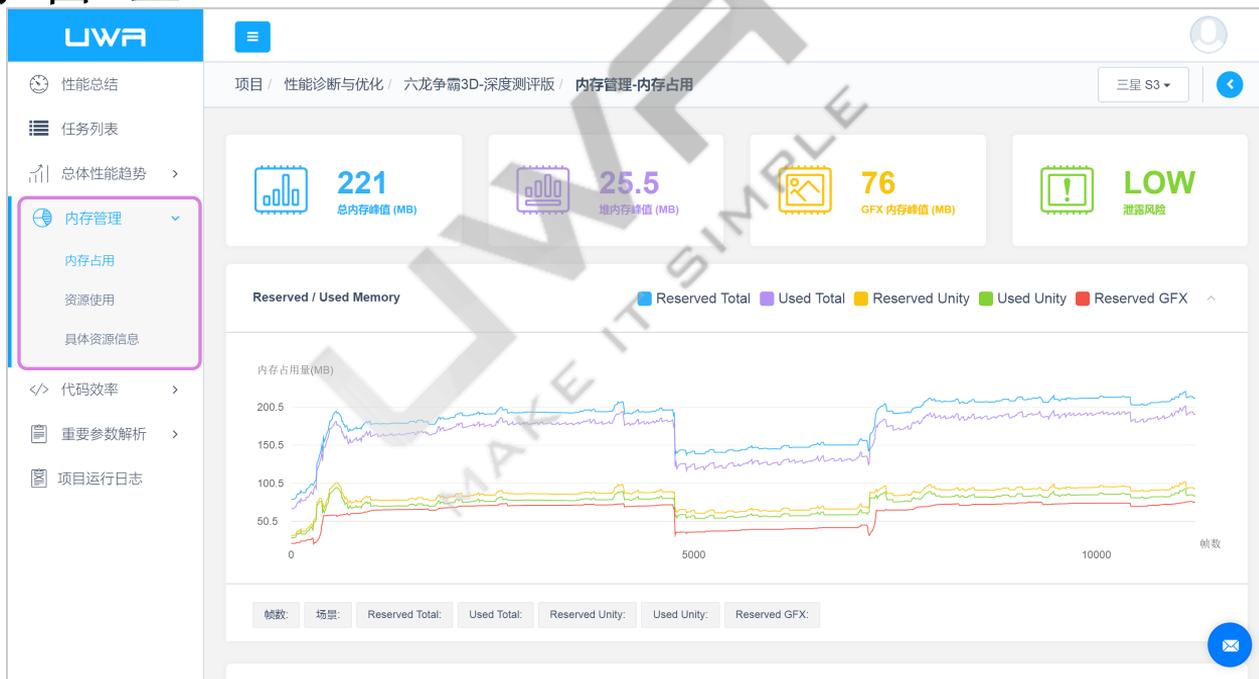
# UWA 优化

- 引擎模块



# UWA 优化

- 内存管理



# UWA 优化

- 代码效率



# Overview

---

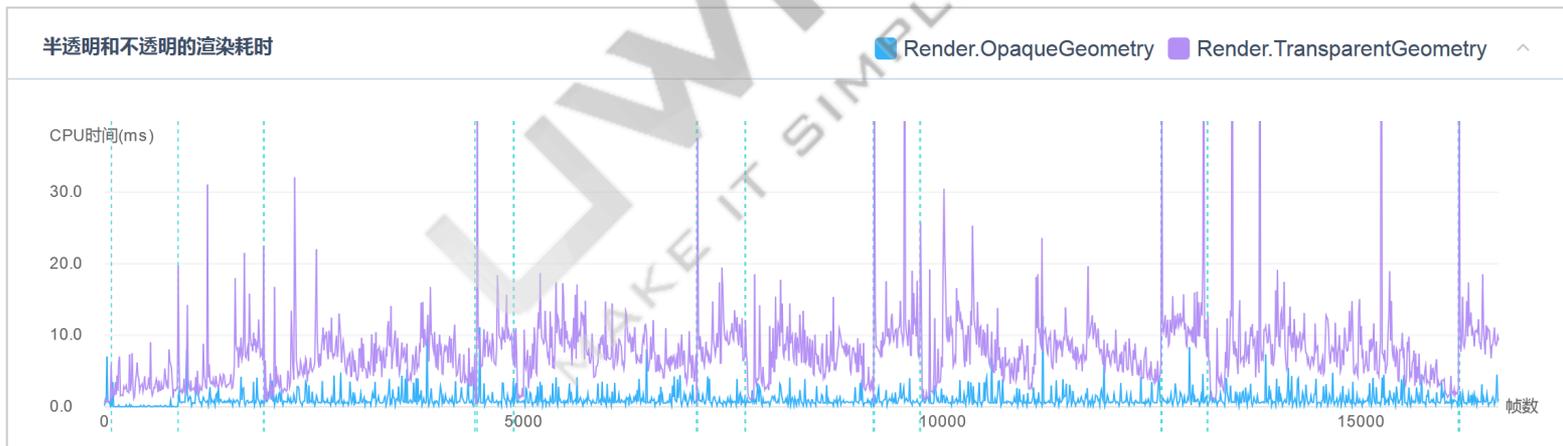
- CPU优化
  - 渲染模块
  - 加载模块
  - 物理模块
  - UI模块
  - 逻辑代码

LWFA  
MAKE IT SIMPLE

# CPU优化

- 渲染模块

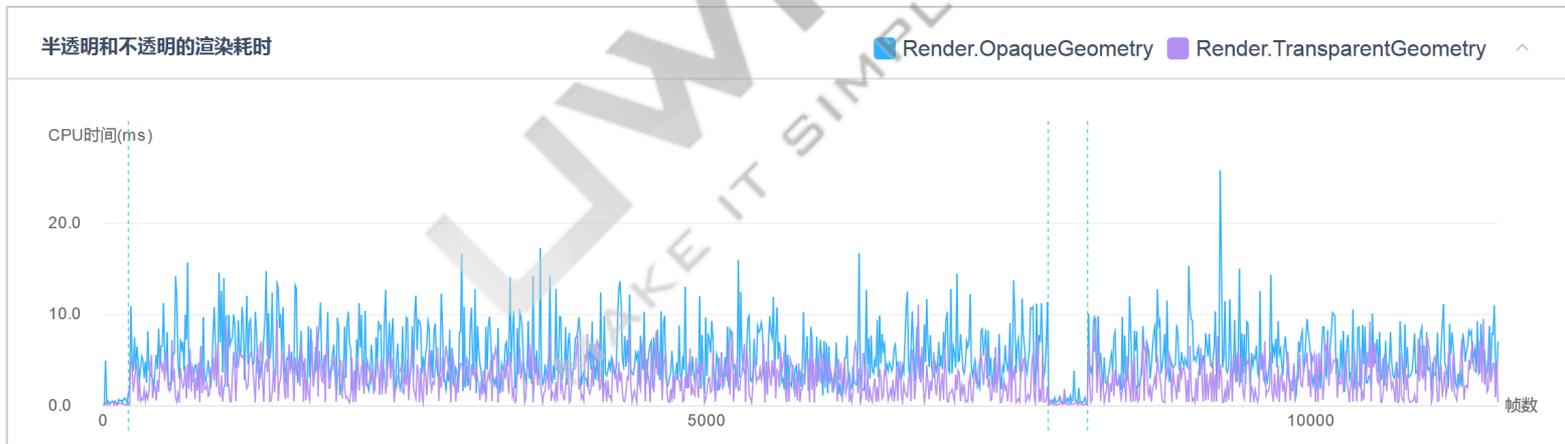
- 半透明物体渲染 & 不透明物体渲染



# CPU优化

- 渲染模块

- 半透明物体渲染 & 不透明物体渲染



# CPU优化

- 渲染模块

- Draw Call & Triangle

**Mobile:**

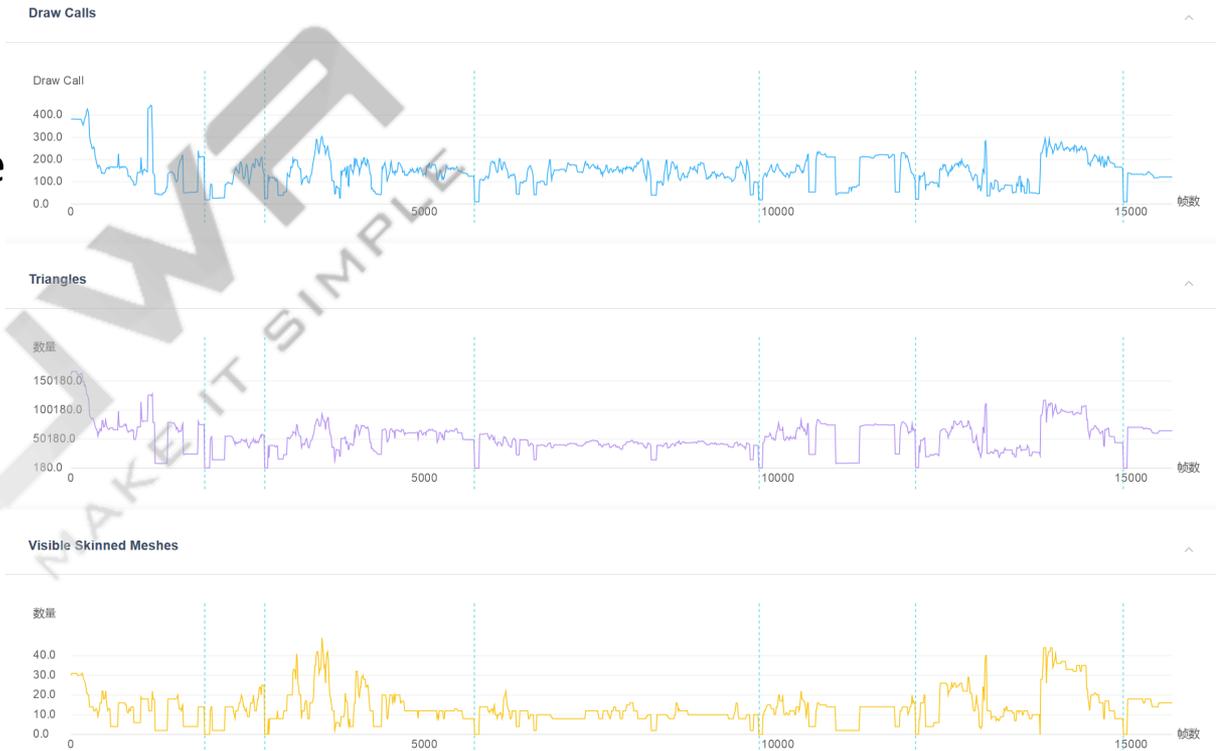
Draw Call < 200

三角面片 < 10w

**Gear VR:**

Draw Call < 200

三角面片 < 20w



# CPU优化

- 渲染模块

- Draw Call & Triangle

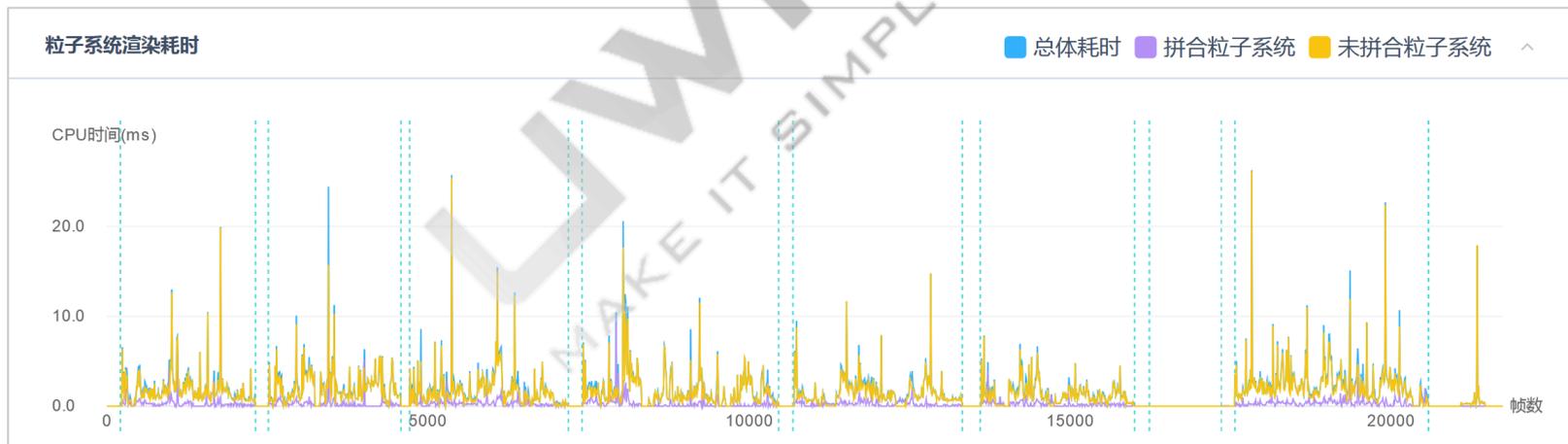
Draw Call Batching、Texture Packing

Asset Store上的模型简化工具（SimpleLOD等）

# CPU优化

- 渲染模块

- 粒子系统渲染



# CPU优化

- 渲染模块

- 粒子系统渲染

简化粒子系统、减小屏幕的覆盖面积

尽可能将相同材质的粒子系统放在比较接近的深度下

尝试升级到Unity 5.3版本以后

# CPU优化

- 渲染模块

- 蒙皮网格渲染

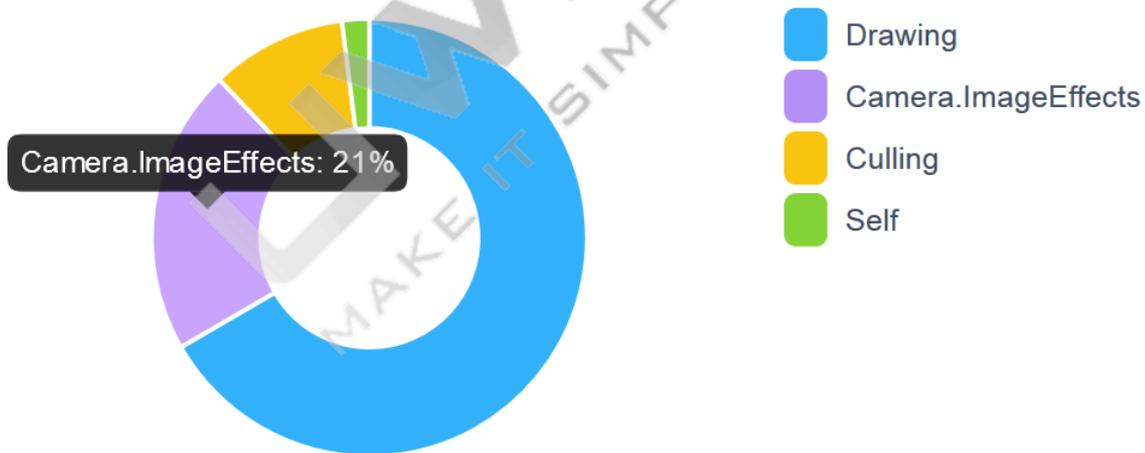
减少Skinned Mesh面片数

简化Shader

资源名称	生命周期(场景数)	内存占用	数量峰值	Vertex数量	Triangles数量	Normal数量	Colors数量	Tangents数量	BoneWeights
<input type="checkbox"/> mesh	4	1.2 MB	1	8933	12344	8933	0	8933	8933
<input type="checkbox"/> mesh	4	920.2 KB	1	6706	6896	6706	0	6706	6706
<input type="checkbox"/> mesh	6	874.5 KB	1	5337	6824	5337	5337	5337	5337
<input type="checkbox"/> mesh	7	830.3 KB	1	5330	6704	5330	0	5330	5330
<input type="checkbox"/> mesh	7	746.1 KB	1	5295	6992	5295	0	5295	5295
<input type="checkbox"/> mesh	6	735.0 KB	1	4906	3651	4906	0	4906	4906
<input checked="" type="checkbox"/> mesh	7	729.6 KB	1	4637	6408	4637	0	4637	4637
<input type="checkbox"/> mesh	2	691.3 KB	1	4449	5314	4449	0	4449	4449
<input type="checkbox"/> mesh	7	459.1 KB	1	4087	6244	4087	0	0	4087
<input type="checkbox"/> mesh	6	624.8 KB	1	4086	4016	4086	0	4086	4086

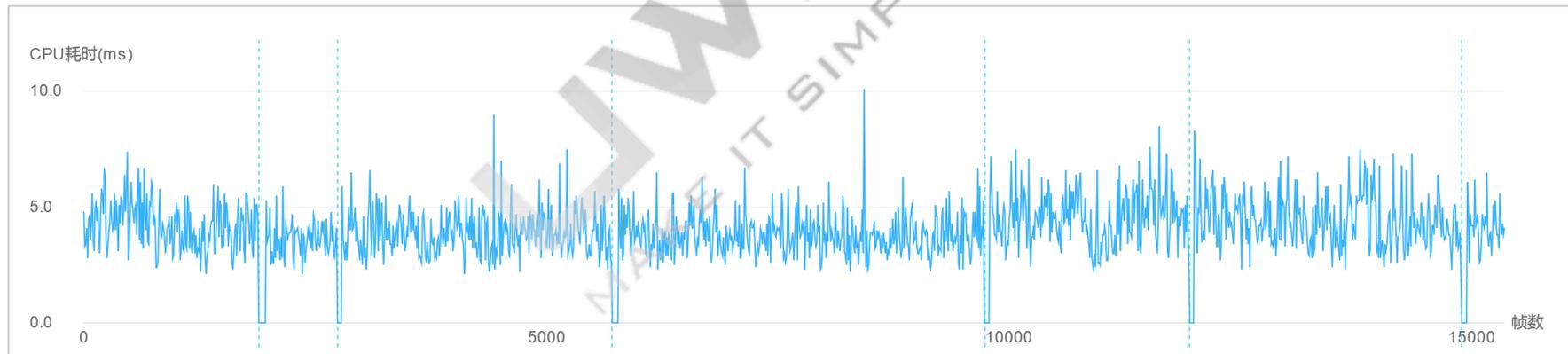
# CPU优化

- 渲染模块
  - 相机屏幕后处理特效



# CPU优化

- 渲染模块
  - 相机屏幕后处理特效



# CPU优化

- 渲染模块

- 相机屏幕后处理特效

- 尝试开启多线程渲染

- 需要关注渲染模块性能的真实变化

LWA  
MAKE IT SIMPLE

# CPU优化

- 加载模块

- Loading.UpdatePreloading

负责资源的加载（Texture、Mesh、Shader等）、代码序列化等



# CPU优化

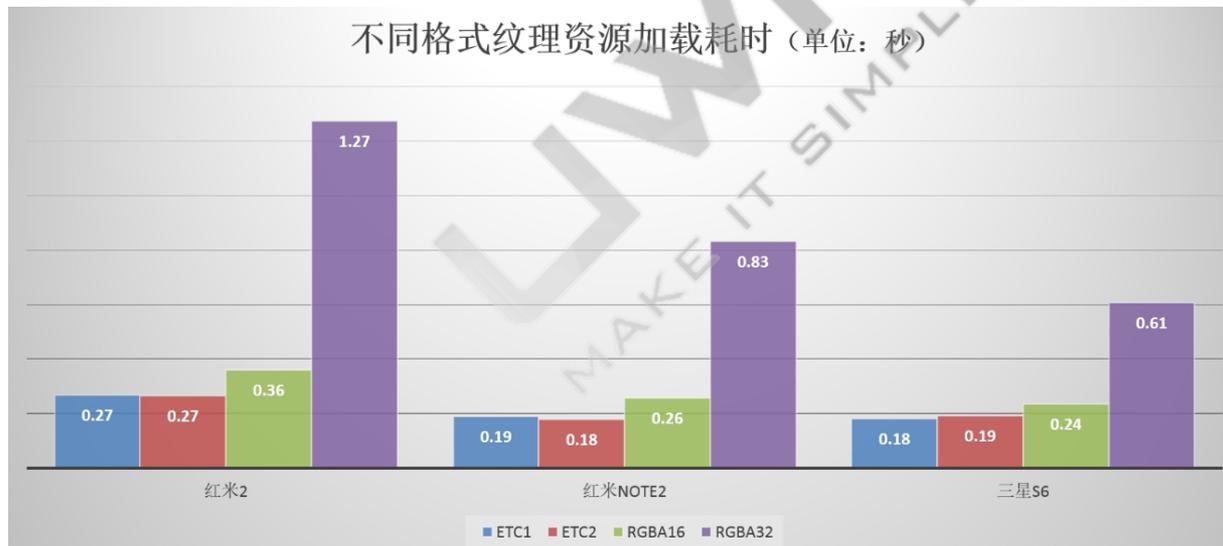
- 加载模块

- Loading.UpdatePreloading

- 资源加载量（资源大小、AssetBundle大小）
    - 关注纹理格式（ETC1:Android, PVRTC:iOS, DXT:Windows, ASTC:Gear VR）
    - Shader解析（Shader.Parse）
    - 序列化信息（SerializedField）
    - 资源卸载（Resources.UnloadUnusedAsset）

# CPU优化

- 加载模块
  - 资源使用



UWA Blog:  
[blog.uwa4d.com](http://blog.uwa4d.com)

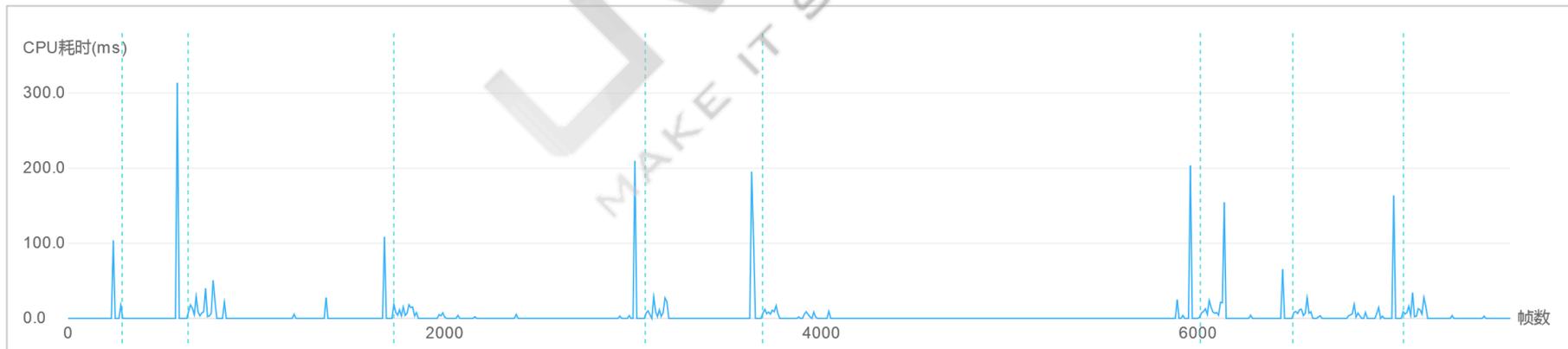
# CPU优化

- 加载模块

- 冗余资源加载：Shader解析

- Shader加载时的解析开销

- AssetBundle依赖、独立打包



# CPU优化

- 加载模块

- 冗余资源加载：AssetBundle打包问题

在我们检测的所有使用AB加载资源的项目中，存在资源冗余问题的项目占比为：

**100%**

总 AB 数

2673

有外部依赖的 AB 数

1063

含冗余资源的 AB 数

470

缺失 AB 数

0

## AssetBundle文件的资源使用情况

输入AssetBundle文件名

Search:

AssetBundle 名称	TextAsset	Sprite	Material	AudioClip	Shader	Mesh	Texture2D	Font	AnimationClip
font_wqy.unity3d							1	1	
prefabs_effect_10001.unity3d			3				3		
prefabs_effect_10004.unity3d			2			1	1		1
prefabs_effect_10005.unity3d			2				2		1
prefabs_effect_10008.unity3d			1				1		
prefabs_effect_10010.unity3d			2			1	2		1
prefabs_effect_10012.unity3d			5			2	5		1
prefabs_effect_10013.unity3d			2			1	2		
prefabs_effect_10018.unity3d			1			1	1		1
prefabs_effect_10028.unity3d			1			1	1		

## GroundObjects.assetbundle 所包含的具体资源



Mesh (3)						
polySurface1566	冗余	polySurface1576	冗余	polySurface1570	冗余	

Material (2)					
Atlas_Alpha1	冗余	Atlas_Nature1	冗余		

Texture2D (9)						
Atlas_Nature1	冗余	Atlas_Alpha1	冗余	Atlas_Nature1_nmp	冗余	grasstxt1
Ground5		grasstxt2		Ground6_B		Ground1_B
ground3						

哪些AssetBundle文件依赖它？

- Terrain\_Data\_1\_1.assetbundle
- Terrain\_Data\_2\_8.assetbundle
- Terrain\_Data\_6\_3.assetbundle

它依赖哪些AssetBundle文件？

无

全部 (210)

Texture2D (85)

Mesh (113)

Material (9)

Shader (3)

输入资源名称

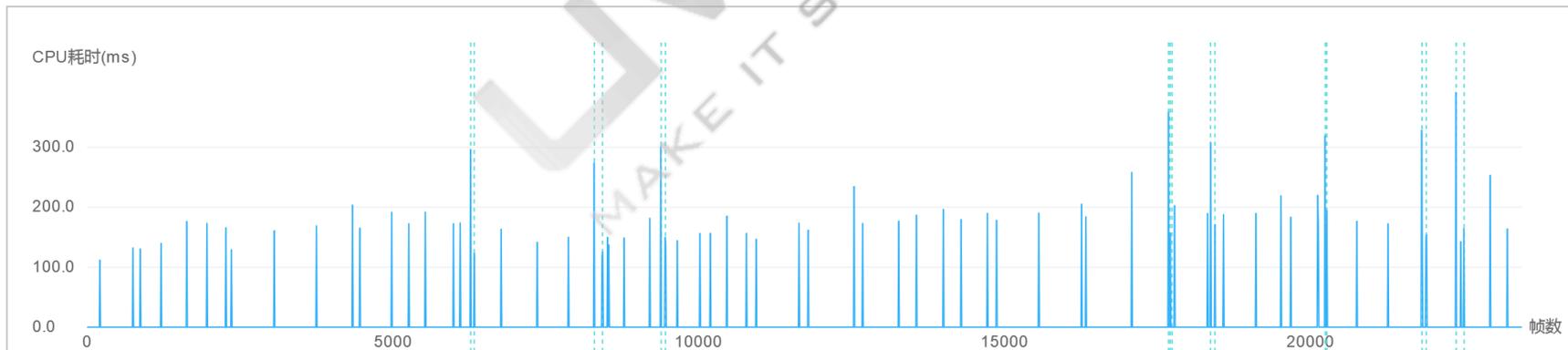
 AssetBundle文件
  宽度
  高度
  内存占用
  格式
  MipMap功能
  Read/Write

资源名称	资源类型	AB文件数量	相应的AB文件	宽度	高度	内存占用	格式	MipMap功能	Read/Write
Atlas_Nature1_nmp	Texture2D	2	topdownmobile_nature_source_materials.assetbundle GroundObjects.assetbundle	1024	1024	682.7 KB	ETC_RGBA4	True	False
Atlas_Alpha1	Texture2D	2	topdownmobile_nature_source_materials.assetbundle GroundObjects.assetbundle	1024	1024	2.7 MB	RGBA4444	True	False
Atlas_Nature1	Texture2D	2	topdownmobile_nature_source_materials.assetbundle GroundObjects.assetbundle	1024	1024	682.7 KB	ETC_RGBA4	True	False
SplatAlpha 0	Texture2D	1	Terrain_Data_1_7.assetbundle	128	128	85.3 KB	ARGB32	True	True
SplatAlpha 0	Texture2D	1	Terrain_Data_7_2.assetbundle	128	128	85.3 KB	ARGB32	True	True

# CPU优化

- 加载模块
  - Resources.UnloadUnusedAssets

性能开销很大，建议仅在场景切换后调用



# CPU优化

- 加载模块

- Loading.LoadFileHeaders

▼ Loading.ReadObject	57.1%	1.3%	64	145.8 KB	1398.37	34.06
▼ Loading.AwakeFromLoad	53.7%	0.8%	63	145.8 KB	1315.12	20.90
▼ Loading.ReadObject	52.9%	3.9%	167	145.8 KB	1294.22	96.78
Loading.LoadFileHeaders	21.9%	21.9%	103	0 B	537.83	537.83
▼ Loading.ReadObject	16.2%	0.2%	53	0 B	398.21	7.32
Loading.LoadFileHeaders	14.8%	14.8%	53	0 B	364.23	364.23
Loading.AwakeFromLoad	1.0%	1.0%	53	0 B	26.58	26.58
Loading.FindInThreadedActivationQueue	0.0%	0.0%	53	0 B	0.06	0.06
▶ Loading.AwakeFromLoad	10.0%	0.0%	167	116.3 KB	244.85	1.38
Loading.IDRemapping	0.6%	0.6%	15172	0 B	14.84	14.84
AudioItem..ctor()	0.0%	0.0%	54	0 B	0.63	0.63
MecanimEvents..ctor()	0.0%	0.0%	12	10.1 KB	0.58	0.58
Loading.FindInThreadedActivationQueue	0.0%	0.0%	167	0 B	0.22	0.22
AudioSubItem..ctor()	0.0%	0.0%	58	0 B	0.15	0.15
Event..ctor()	0.0%	0.0%	11	176 B	0.03	0.03
AudioController..ctor()	0.0%	0.0%	1	24 B	0.03	0.03
UIPanel..ctor()	0.0%	0.0%	1	168 B	0.02	0.02
HKBathHolyLightEnterEvent..ctor()	0.0%	0.0%	1	0 B	0.01	0.01
Loading.LoadFileHeaders	1.9%	1.9%	49	0 B	47.42	47.42
Loading.IDRemapping	0.0%	0.0%	230	0 B	1.68	1.68
Loading.FindInThreadedActivationQueue	0.0%	0.0%	64	0 B	0.08	0.08

# CPU优化

- 加载模块

- 加载方式

1024 x 1024 ETC1纹理，10份

逐一进行加载： 0.3s

一次性加载： 2s

切换场景时

减少协程yield return 次数；  
同步加载

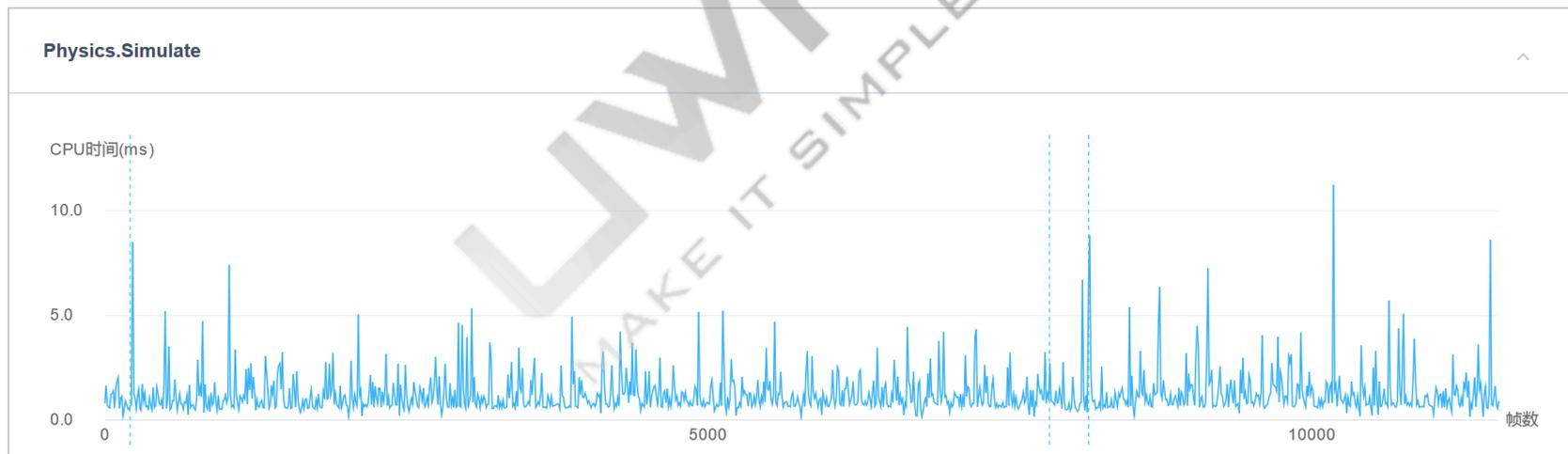
战斗运行时

异步加载

LWA  
MAKE IT SIMPLE

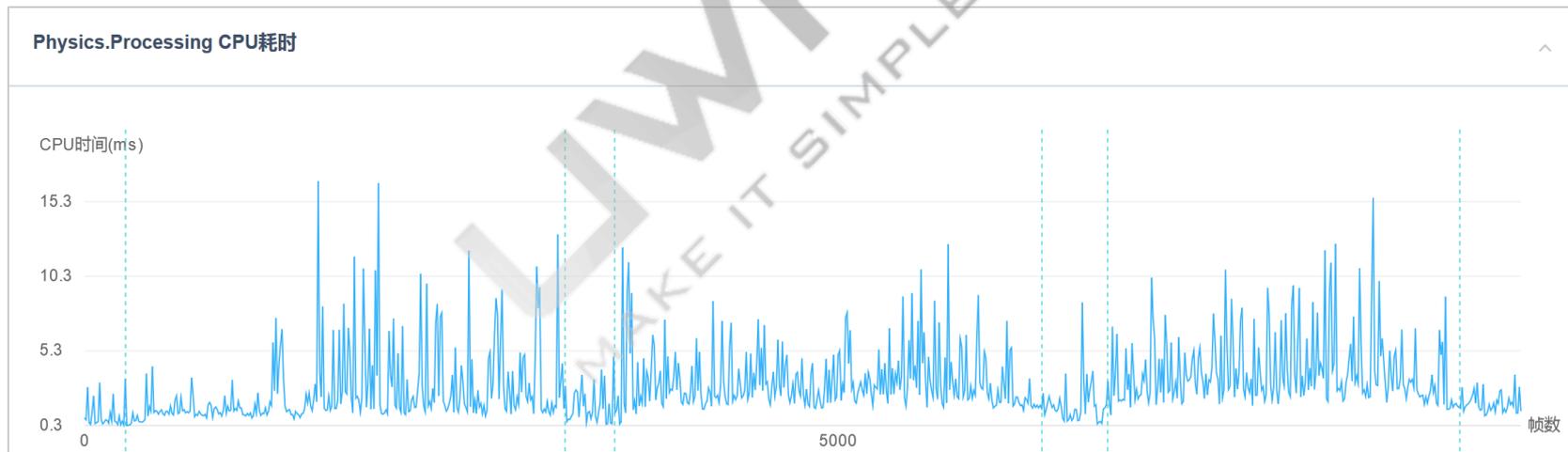
# CPU优化

- 物理模块
  - Physics.Simulate



# CPU优化

- 物理模块
  - Physics.Processing



# CPU优化

- 物理模块
  - Contacts碰撞对

LWA  
MAKE IT SIMPLE

# CPU优化

- 物理模块
  - OnTriggerXXX  
需要大家自行检测逻辑代码

LWFA  
MAKE IT SIMPLE

# CPU优化

- 物理模块

- “有趣”的现象

你可能什么都没做，但物理模块就是莫名其妙的高

LWA  
MAKE IT SIMPLE

# CPU优化

- UI模块
  - NGUI是按Panel进行重建的、UGUI是按Canvas进行重建的

LWA  
MAKE IT SIMPLE

# CPU优化

- UI模块
  - 动态元素和静态元素分离

LWFA  
MAKE IT SIMPLE

# CPU优化

- UI模块

- 合理配置UI Canvas/Panel

- 一个Canvas下的Widget不宜过多

LWA  
MAKE IT SIMPLE

# CPU优化

- UI模块
  - 善用UIPanel的Static属性

LWFA  
MAKE IT SIMPLE

# CPU优化

- UI模块
  - 不推荐通过Instantiate/Destroy或Active/Deactive来频繁切换UI界面

LWA  
MAKE IT SIMPLE

# CPU优化

- UI模块
  - 不推荐将Anchor的执行模式设置为OnUpdate

LWA  
MAKE IT SIMPLE

# CPU优化

- UI模块

- 控制UI Draw Call

- 通过UISprite来代替UITexture

- UI界面避免重叠

LWVA  
MAKE IT SIMPLE

# 代码优化

- Top10函数（CPU开销和堆内存分配）
  - 宏观筛选、微观定位



基于整体时间

基于特定场景

基于特定帧数

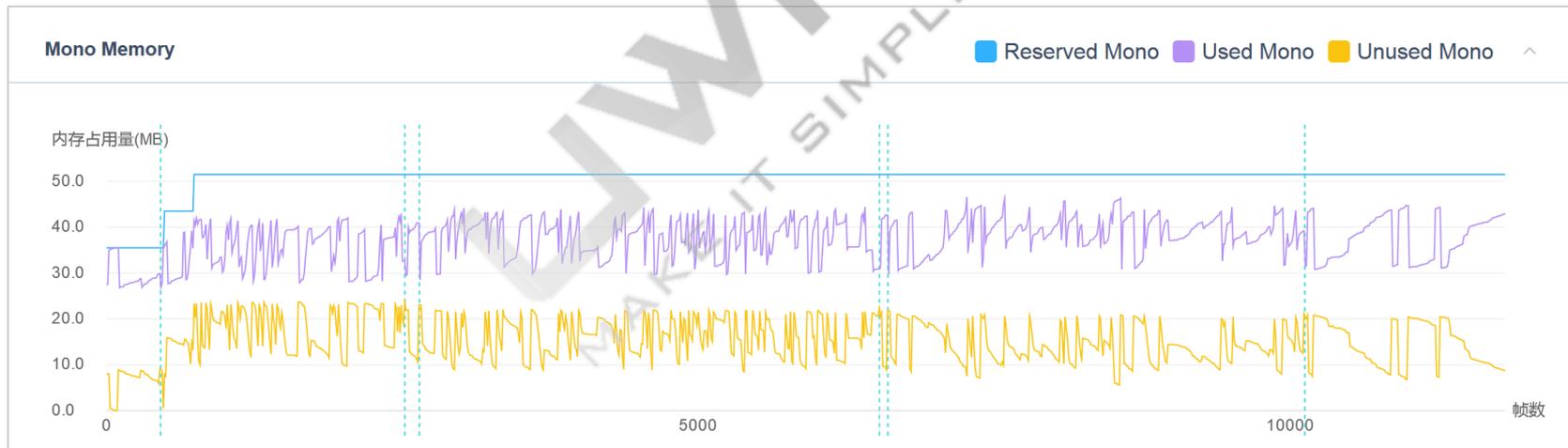
# 代码优化

- Top10函数（CPU开销和堆内存分配）
  - 很多性能问题从技术上来说都没问题，需要从策略性上进行解决

LWW  
MAKE IT SIMPLE

# 代码优化

- Top10函数（CPU开销和堆内存分配）
  - Mono堆内存



# 代码优化

- Top10函数（CPU开销和堆内存分配）
  - Mono堆内存  
配置文件过大



# 代码优化

- Top10函数（CPU开销和堆内存分配）

- Mono堆内存

- 避免不必要的堆内存分配

- 避免频繁New Class/Constainer
      - 控制Log输出
      - For代替Foreach
      - String连接
      - 避免大量装箱操作
      - Lambda表达式、LINQ等合理的使用

- ...

# 代码优化

- GC调用
  - 建议调用频率达到1000帧/次

LWA  
MAKE IT SIMPLE

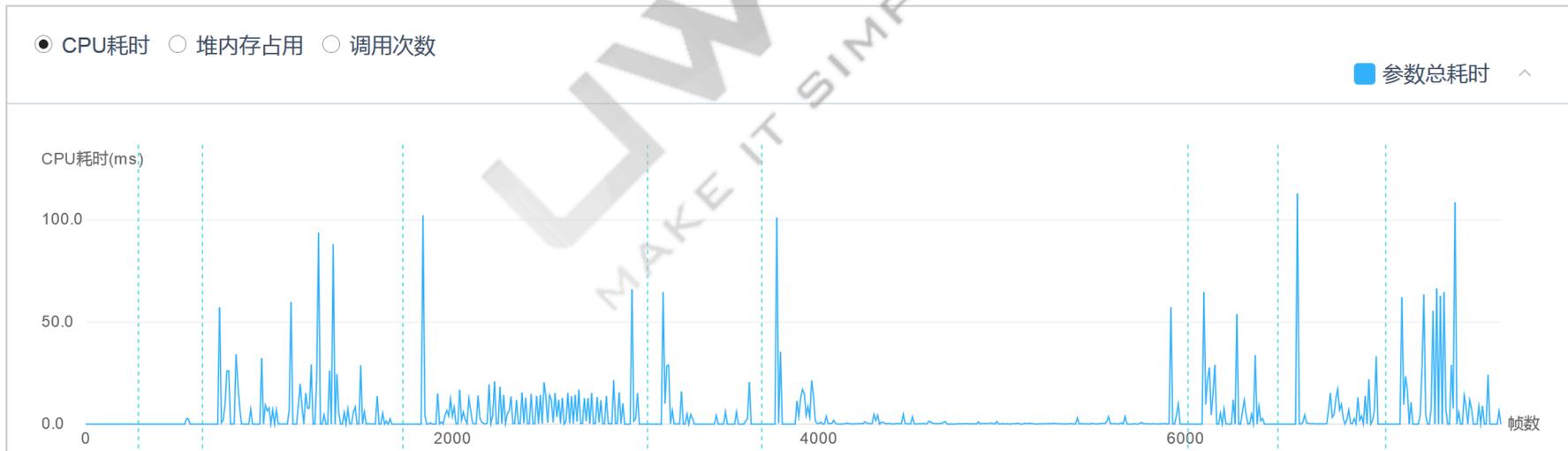
# 代码优化

- Instantiate调用过于频繁
  - 技能释放、角色加载、UI切换等
  - 产生内存碎片，加速GC到来
  - 使用Object Pool（缓存池）

LWA  
MAKE IT SIMPLE

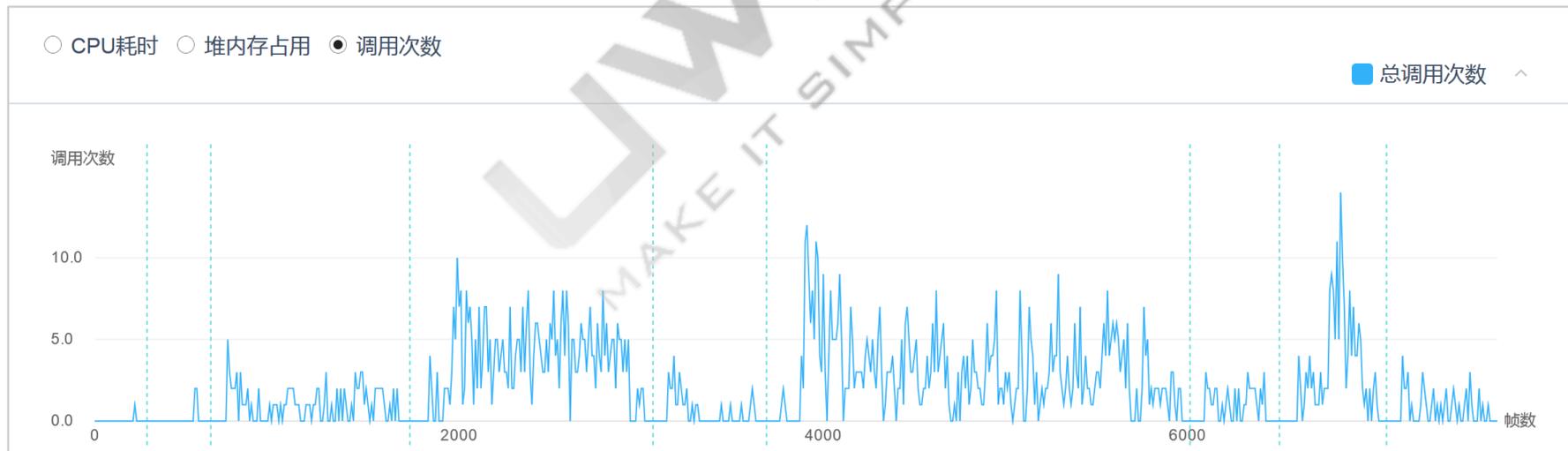
# 代码优化

- Instantiate调用过于频繁
  - 技能释放、角色加载、UI切换等
  - 产生内存碎片，加速GC到来
  - 使用Object Pool（缓存池）



# 代码优化

- Instantiate调用过于频繁
  - 技能释放、角色加载、UI切换等
  - 产生内存碎片，加速GC到来
  - 使用Object Pool（缓存池）



# 代码优化

- Instantiate调用过于频繁
  - 技能释放、角色加载、UI切换等
  - 产生内存碎片，加速GC到来
  - 使用Object Pool（缓存池）

调用路径	总调用次数 ↕	总CPU占用(ms) ▲	总堆内存占用 ↕
BehaviourUpdate / App3.Update()	1200	1877.2	2.2 MB
AssertBundleLoader.StartLoadAsset() [Coroutine: MoveNext]	265	1631.2	5.0 MB
BehaviourUpdate / TimerManager.Update()	43	529.1	426.9 KB
BehaviourUpdate / UICamera.Update() / UIButton.OnClick()	19	269.1	1.1 MB
BehaviourUpdate / UICamera.Update() / UIEventListener.OnPress()	8	14.5	192 B

# 总结

- 引擎模块、代码效率等等...
- 具体的优化经验其实并不重要，两年前的经验不适合于今天，今天的经验也不适合于两年后...
- 真正想让大家看到的是研究问题的方式和厚积薄发的态度，这才是以后能帮到大家的东西！
- 这是我们做UWA的真正原因。

[www.uwa4d.com](http://www.uwa4d.com)

