现象：卡顿、发热、耗电、闪退、延迟

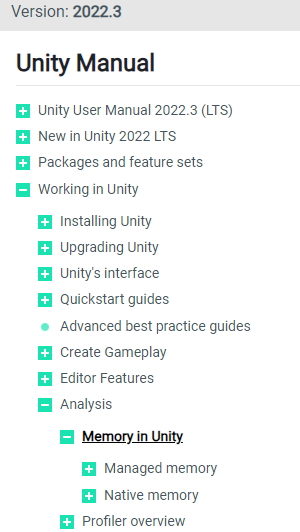
一个app进程能够使用的内存是有一定限制的，内存过高、内存泄漏

性能优化做了什么内容？

手机游戏运行时占用内存较大带来的主要问题是什么？

Unity 手册，内存占用详解

[file:///H:/Software/Unity/2022.3.26f1/Editor/Data/Documentation/en/Manual/performance-memory-overview.html](file://H:\\Software\\Unity\\2022.3.26f1\\Editor\\Data\\Documentation\\en\\Manual\\performance-memory-overview.html)



1.对CPU和GPU产生的影响分别是什么？

2.MipMap 有什么作用，如何使用，何时使用？

1. 什么是DrawCall，如何减少DrawCall？降低 DrawCall：24种无法合批的场景 <https://github.com/Unity-Technologies>

Canvas 绘制时主要的性能瓶颈是什么，如何优化

4.静态合批与动态合批的原理是什么？有什么限制条件？为什么

shader 优化：减少pass以及复杂计算

动、静态合批的要求？

UGUI 合批要求？？

5.LOD层次细节技术？

LodGroup，根据物体距离摄像机的距离切换不同的模型，近处的渲染精模，远处的渲染低模

6.Occlusion Culling 遮挡剔除技术，模型背面摄像机看不到的面尽量不渲染

7.Lightmapping 光照贴图技术，对游戏场景进行烘焙

8.Mesh合并

9.资源池的利用：当大量物体需要频繁创建和销毁的时候使用资源池，可以节约性能

10.内存溢出、内存泄漏、如何优化内存,CPU,GPU开销

11.资源优化，在压缩不影响效果的情况下尽量选择压缩

用不到的资源不要放在项目中，特别是 Resources目录，因为该目录下的资源都会打包到安装包中，

不要相同的资源复制多份放到不同的目录

12.关于切换场景加载耗时的优化问题

在切场景时，为什么调整Application.backgroundLoadingPriority为High可以减少加载耗时呢?

A：Application.backgroundLoadingPriority这个API会限制主线程的集成的时间，默认设置是

ThreadPriority.BelowNormal，也就是每一帧执行主线程集成的操作耗时不能超过4毫秒，这将

会导致每一帧剩余的时间用于等待或者渲染，而这些等待和渲染的时间都是无意义的耗时。如

果把这个设置改为ThreadPriority.High，那么每一帧执行主线程集成的操作耗时可以增加到

50毫秒，大大减少了所需要的帧数。这里主线程的集成时间体现在函数Application.Integrate Assets in Background

上面，它会受到加载Texture、Mesh的Texture.AwakeFromLoad、Mesh.AwakeFromLoad的影响，

另外对于比较复杂的Prefab（父子节点多，层数多）需要进行拼装，耗时也会体现在这个函数当中。

13.SpriteAtlas打包AssetBundle后运行时内存大小和什么有关系

明明只有2张Atlas，2个纹理只有6MB，最终AssetBundle的内存却达到11.7MB

A：运行时AssetBundle大小会受到打包Sprite数量的影响，打包了1700个Sprite的情况下AssetBundle内存11.7MB

测试了同样大小的SpriteAtlas，打包了6个1024\*1024大小的Sprite，AssetBundle大小却只有4.9KB。

14.手机上使用MRT的限制和兼容性

A：在支持OpenGL ES 3.0的手机上就支持MRT，可参考：

https://www.informit.com/articles/article.aspx?p=2181697&seqNum=2

https://registry.khronos.org/OpenGL/specs/es/3.0/es\_spec\_3.0.pdf

另外实测过，在红米4X这样非常低端的手机上，MRT也是可以正常运行的。红米4X的GPU是Adreno 505。

15.如何控制粒子系统组件数量的上限

项目Overdraw有点高，我们想做一个与MMO游戏里同屏角色数量上限类似的，粒子系统上限的功能。

也就是低端分档下，当粒子数量达到某个值以后，就不再往画面里画新的粒子了。但只对特效Prefab

数量做限制还是比较模糊，毕竟不同特效的粒子数量不同，所以想精确到播放状态的粒子组件数量。

A1：用Find肯定很耗时，尽量用逻辑做。比如，粒子系统一般都做缓存池管理，可以直接把相关逻辑

写在这里：做一个计数器C，某个特效E要实例化或出池时，同时传入它的子粒子数量A（提前配置好），

加给C；入池再减掉A；对C作监控看是否在某个值内，否则停止所有实例化和出池操作。同理，给每个

特效E挂一个触发器脚本，激活或隐藏的时候把子粒子数量A传给写在某处逻辑的计数器C即可。

另外实际做的时候，还可以给不同的粒子特效加上不同的计算影响系数，这样可以让某些特效

（比如多人游戏中玩家角色的技能）出现的概率更高一些，而其他特效（其他玩家的技能）显示得少一些

A2：提个思路，对于每一个特效Prefab，可以离线给这个Prefab做一个评分。对于场上存在的特效Prefab，

记录总的特效评分，可以设置一个总的上限评分，超过就不显示

这样问题就抛到了离线分析每个特戏Prefab评分上，评分参考维度可以参考：特效粒子发射器数量、

特效overdraw、特效Batches、特效内存等等

16.Crash异常崩溃后如何快速定位

1. 各类烘焙（光照贴图烘焙，遮挡剔除烘焙，NavMesh烘焙）的原理与用法
2. 资源生命周期如何管理？如何加载与释放资源？资源打包颗粒是怎么设计的？资源压缩格式是什么？
3. GPU instancing 是什么，何时使用，有什么限制，什么情况下使用 static batching？
4. 如果程序运行时会有 GC 带来的较大性能波动，你认为主要问题在哪里以及优化方向

21.内存碎片是如何产生的？其产生的后果是什么？如何降低产生内存碎片？