<http://www.jianshu.com/p/fd24db27818d>

# 一 内存优化

一 大概标准

1. 纹理             40M

2. Mono          30M

3. Animation  20M

4. Mesh          10M

6. Font            10M

7. Audio          5M

8. GfxDriver    25M

9. ResourceManager 视情况而定（跟 你ResourceManager里放了多少文件）

（该标准是我做 MMO 和 FPS时的标准值 不同项目会有不同程度的降低）

内存安全线 android unity在170M左右 pss在260M  （1G内存手机上不会因为内存不足crash）

ios的在280M左右（均指峰值）

二 工具

1. unity Profile

2. Mono提供的内存查看工具 heapshot 需要替换unity自带的dll  查看当前内存对象情况

3. 网易 emmagee性能评测工具 开源 可做自定义 加入当前场景名字

4. IOS 工具不好用

5. 根据unity提供的API 自定义性能Profile工具  带来的损耗较小。

三 注意点

1. 内存基本存在3种问题 1 内存泄露 2 资源冗余 3 Mono无效内存

一 内存泄露 主要是资源管理 注意AssetBundle Asset对象的卸载与缓存。小心处理挂在DonotDestroy的节点下面的东西。小心static变量持有资源问题。所有场景均有一个空场景进入，空场景负责进行堆内存的清空。

保证每次进入空场景的内存是一样的。

二 资源冗余 保证资源不会被带入多个场景，可勤看Profiler即可。注意不要在内存中出现同一份的多份复制。例如 纹理开启了 read/write enable. 调用 material mesh属性而不是 shareMaterial shareMesh属性。动画进行分割。例如场景只需要idle动画，则可以做一个prefrab只有idle动画，无需加载其他动画。

三 Mono无效内存 保证mono峰值不要太高。主要是配置文件加载，序列化。

四 降低内存 纹理是最好的方式。 更改纹理格式，降低纹理尺寸。检查是否有冗余纹理。

# 2 CPU优化

一 标准

渲染 15ms 左右

脚本 10ms 左右

其他（主要是物理，动画） 8ms   左右

二 评判瓶颈

查看主要通过unity的profile工具。如果瓶颈在GPU上，profile上会有 waitforpresent 占比 或者通过 Adreno屏蔽掉openGl 来查看帧率是否有提升。

然后在看占比最高的一项是什么引起的。进行相对应的优化。

三 常用手段

1. 开启多线程渲染。

2. 减少GC。NGUI是贡献GC大户，可以通过profile进行逐一排查，优化。减少GetComponent AddComponent次数。对象多进行复用。能用struct就用struct(项目推广比较困难)

3. 去掉无用log。

4. 多进行缓存， 减少文件IO次数

5. update函数 最好只在基础层面出现。外围逻辑不要有update 函数，需要也用Invoke 或者自己实现的timer。

6. 减少 skinmesh的数量 尽量用meshRenderer 同屏数过多 通过骨骼序列帧代替skinmesh实现

7. 合并drawcall StaticBatch DymiticBatch

8. 粒子尽量不用 用的话 场景效果大的粒子加入 离屏不渲染

9. 对 粒子加入 lod

# 3 GPU优化

一。标准

1. DrawCall 控制在 200以下

2. 同屏面数控制在4W面以下

二。评判瓶颈

1. Unity Profile 出现 waitforPresent

2. IOS的 FrameDebugger 可以看到CPU跟GPU分别耗时 同时也能看到某步的具体耗时

3.Adreno 也可给出每次pass消耗的clock数以及传输数量数。

三。常用手段

1. 纹理采用GPU支持格式 尽量减少纹理大小

2. camera的远裁剪面设置近一些

3. 看情况开启遮挡剔除

4. 减少透明片 overdraw在低端机会有很严重的性能问题

5. 减少shader复杂度。利用好 mask通道

6. 降低分辨率

7. 善用欺骗 很多效果可以通过巧妙的办法进行仿制。需要多跟美术沟通<https://simonschreibt.de/gat/fallout-3-edges/>

8. GPU基本三大瓶颈 1.带宽 2.Vertex  3. Pixel 找出瓶颈分别优化

四。常用工具

1.unity profile 包括5的frameDebugger

2.ios frameDebugger

3. adreno

（PS unity5 android 以后会默认有一次blit，很耗。曾经的想法是通过unity提供的RenderPlugIn 重新绑定原定的fbo 然后在新的fbo替换掉他的shader 移除摄像机，通过剔除给剔除掉解决。但是重新绑定fbo后，unity会检测到更改，在改回来。所以只能hook住opengl，让绑定函数失效。但是没成功.有相同的问题的各位可以一起聊聊。与4.6相比，大概有5~10帧的消耗）

# 4 包大小优化

未完待续。。。