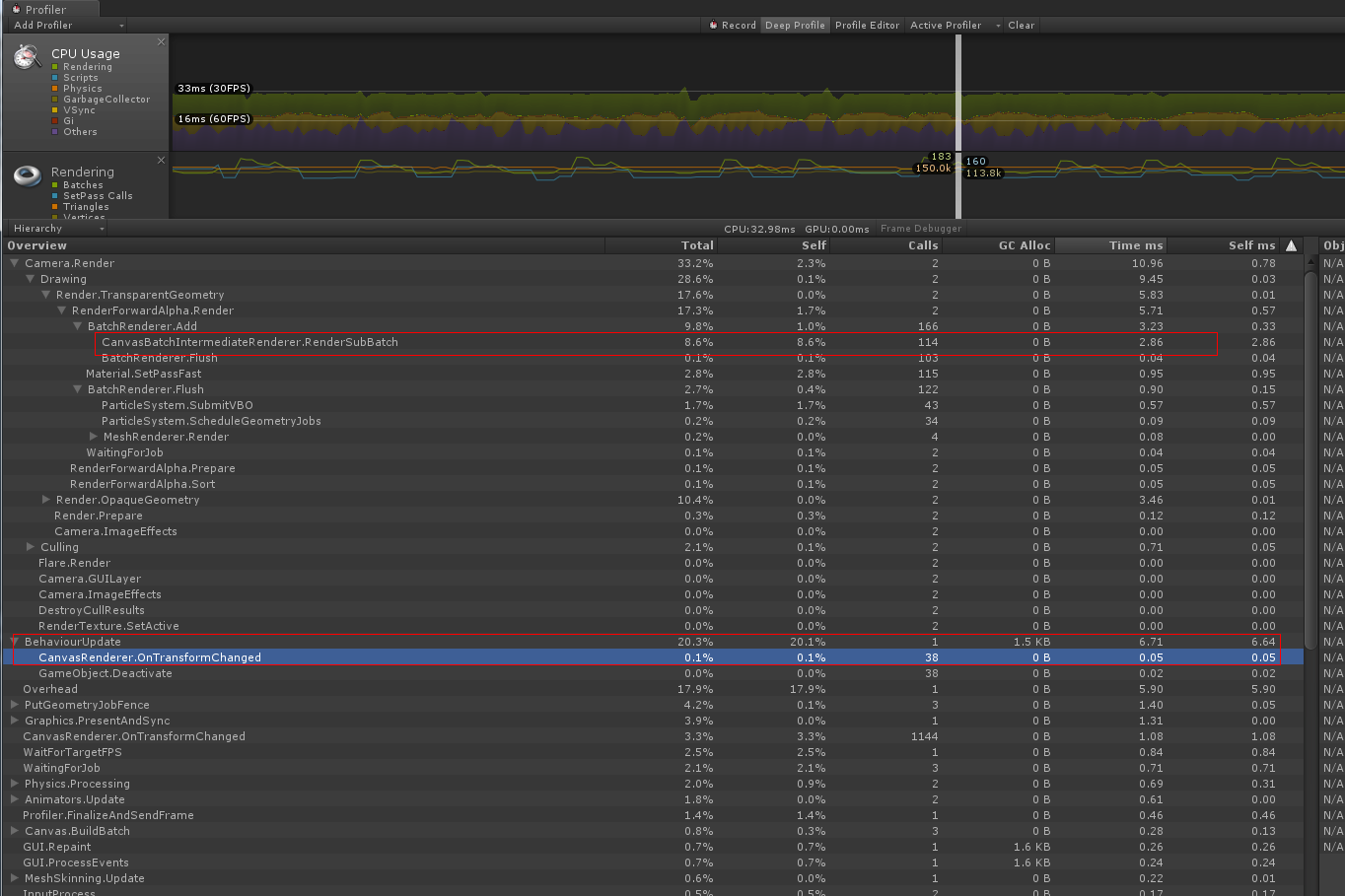
## Android 收集测试 Textmeshpro

使用TextMeshPro的伤害数字，20个npc，小悟空不断普攻



图一

不使用TextmeshPro的伤害数字，20个npc小悟空不断普攻



图二

说明：

Android测试把伤害数字换成Textmeshpro 和原来的伤害数字作对比，因为打出的伤害数字不一定是固定量的，所以存在一定误差，如图一的

CanvasBatchIntermediateRender.RenderSubBatch,的调用次数是114，而图二的时85次，图一的CanvasRender.OnTransformChanged时38次，而图二的是70次，正常来说，因为UGUI存在Bug，UGUI的控件在移动时，CanvasRender.OnTransformChanged接口会调用两次，所以同样的伤害数字数量，图二应该调用CanvasRender.OnTransformChanged接口76次。

结论：

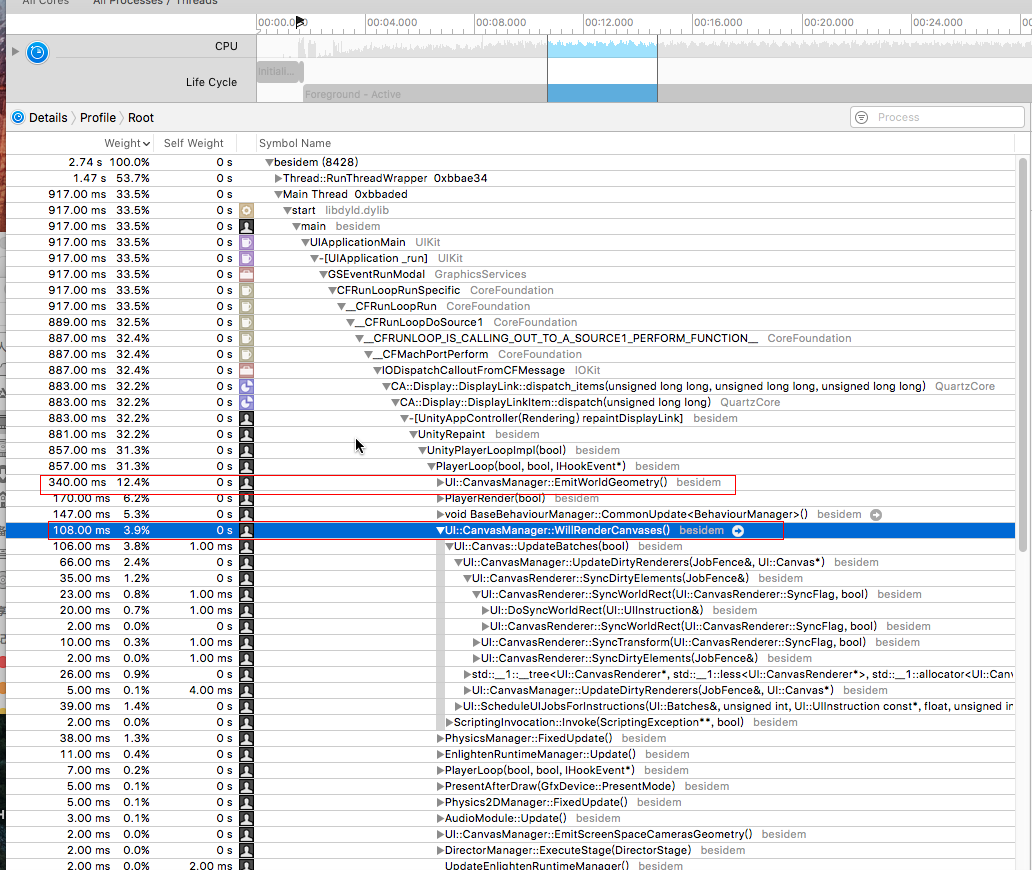
1. 手机上的此时数据上看，使用TextMeshPro在性能上并没有明显提高
2. 要使用TextmeshPro，显示美术做的图片数字，如显示25，需要给textMeshpro控件赋值为<sprite=”test” index=2><sprint=”test” index=5>,不仅需要拆分数字，还需要重新拼接字符串会造成额外开销。
3. Textmeshpro插件没有源码，内部代码有什么坑我们不好把控，如下图，TextmeshPro控件在IL2Cpp下编译时会报错：（应该是TextmeshPro.dll中使用了

UnityEngine.waitforsecondsrealtime,接口而此接口已经过时）。



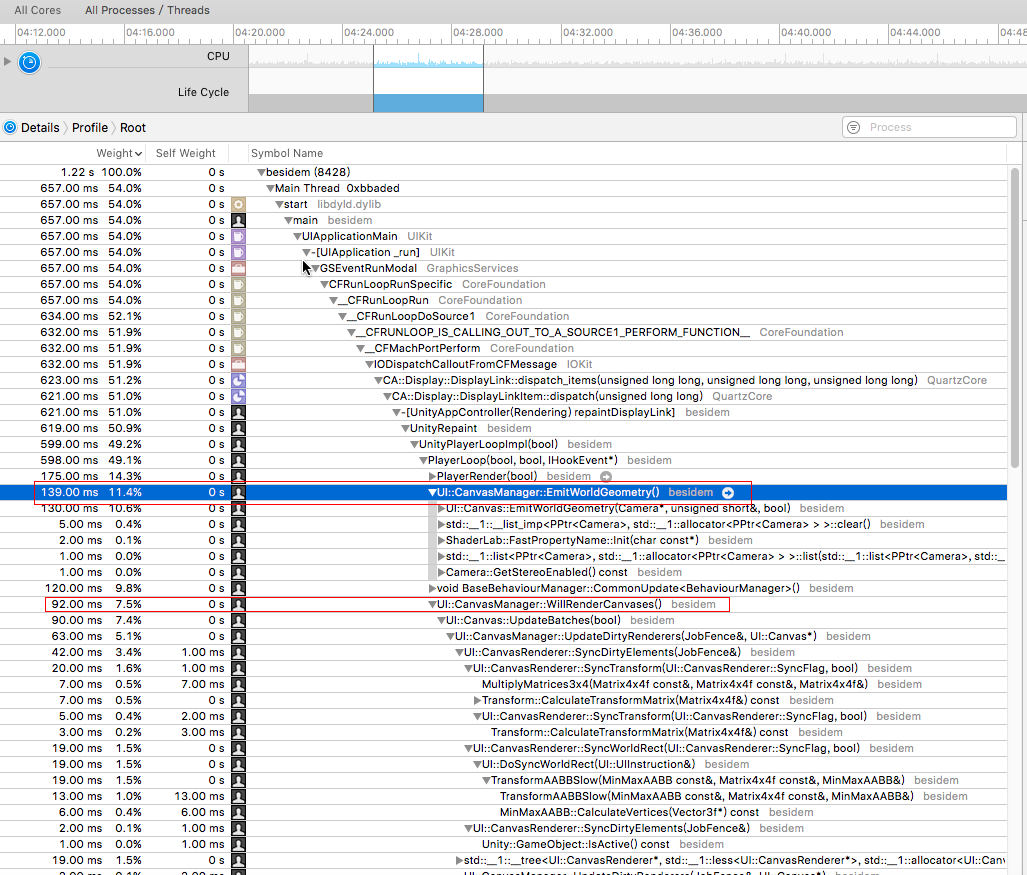
综合上面的测试，不推荐使用TextMeshpro插件做伤害数字。

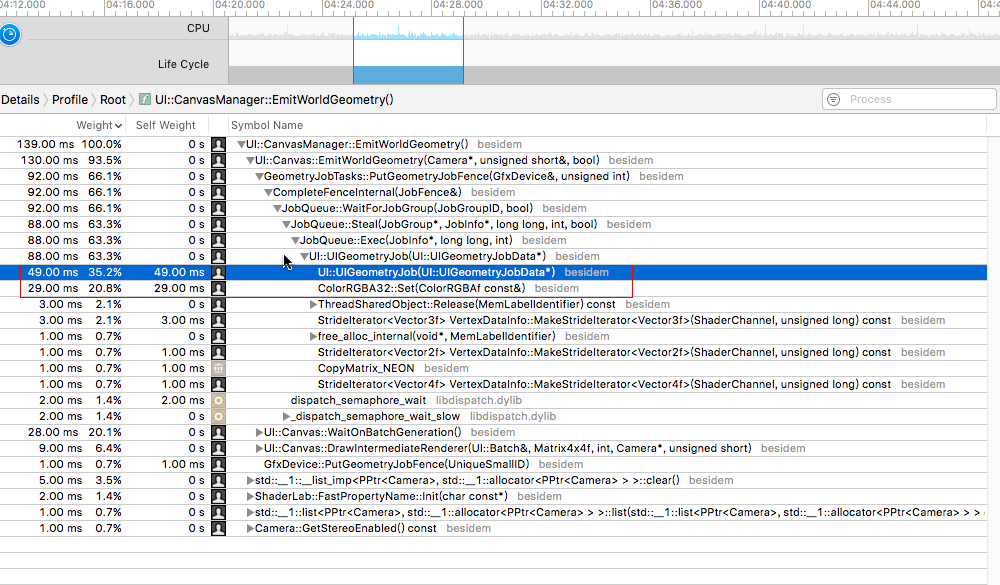
## IOS 的 text加Outline：





## 新的带OutLine的shader的测试对比





上面两组对比条件：1组200个text控件加OutLine控件，2组200个text控件，使用新的带有描边的shader。

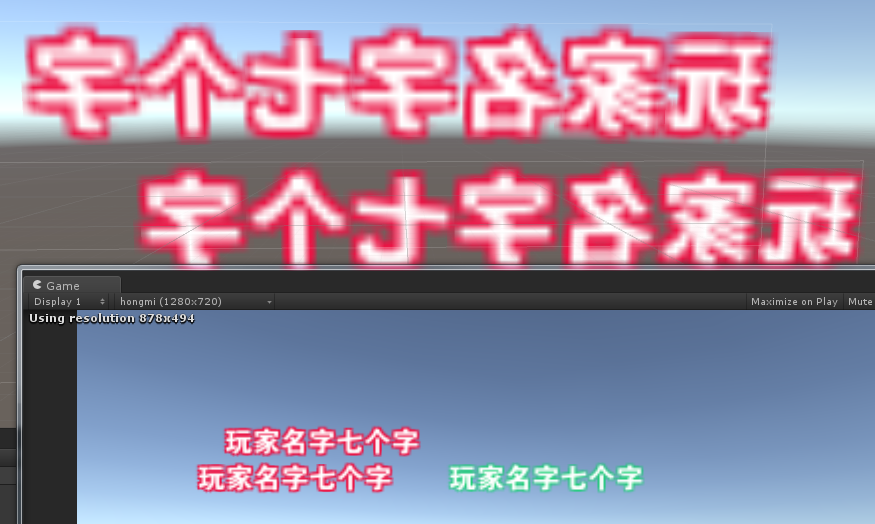
从两组对比中可以发现：消耗差别比价大的两个方法是：UIGeometryJob和ColorRGB32.

1组:176+92ms 2组：49+29ms

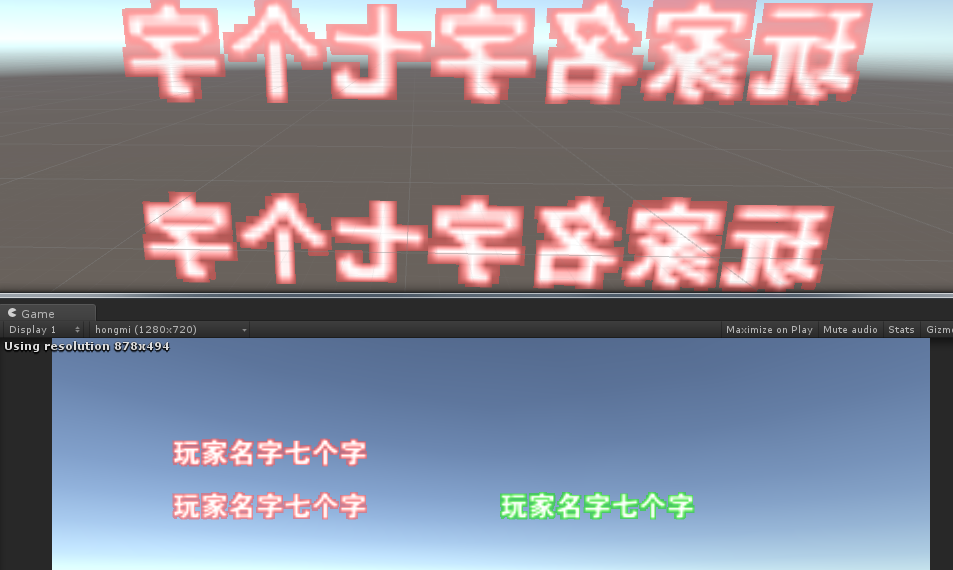
也就是说新的shader在UI渲染上有很大优势。

## 表现对比：

## Text加Outline组



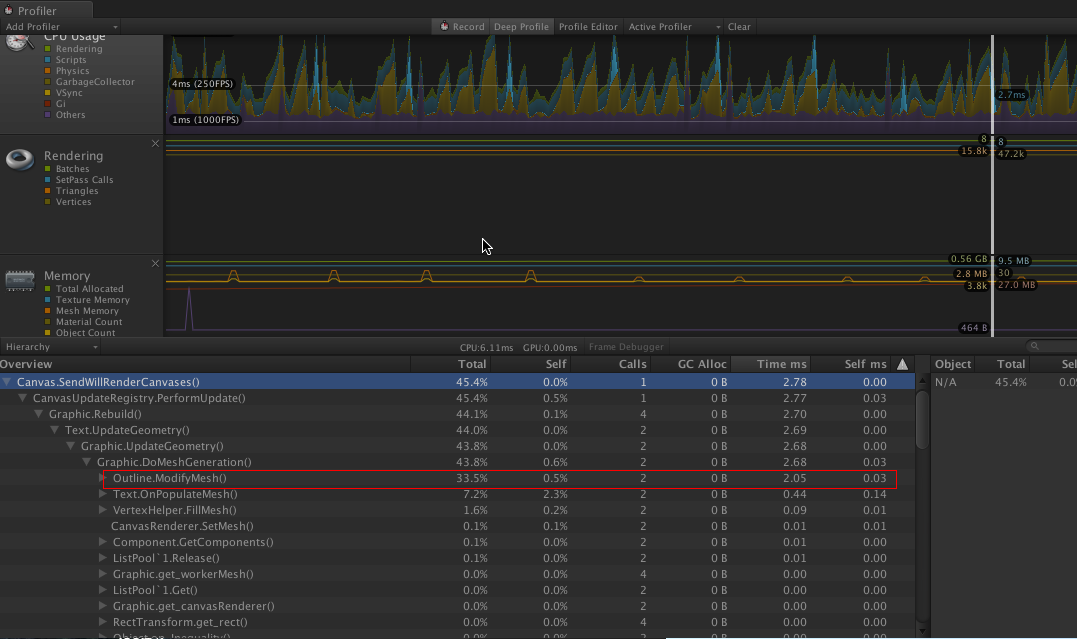
## 带有Outline的新shader组

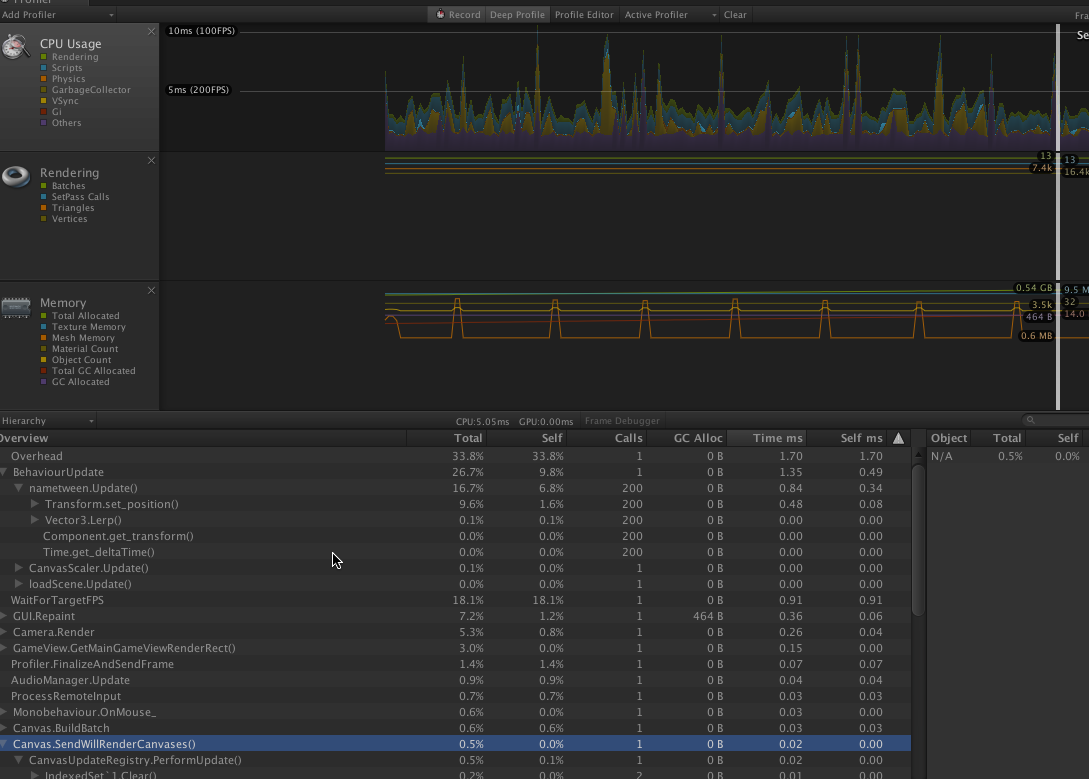


## 其他消耗对比

下面的两组对比条件：1组200个text控件加OutLine控件，每隔0.5s有两个text颜色发生变化，2组200个text控件，使用新的带有描边的shader，每隔0.5s有两个text颜色发生变化。模拟玩家名字发生颜色变化时差生的消耗。

如图可以发现，1组在颜色变化时会调用outline.modifyMesh方法，消耗为2ms，而2组却没有这个消耗。





## 总结

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 总耗时 | UIGeometryJob | ColorRGB32 | outline.modifyMesh |
| Text加Outline | 2.74s | 176ms | 92ms | Outline颜色变化时会触发一次， 2ms |
| 带有Outline的shader | 1.22s | 49ms | 29ms | 无 |

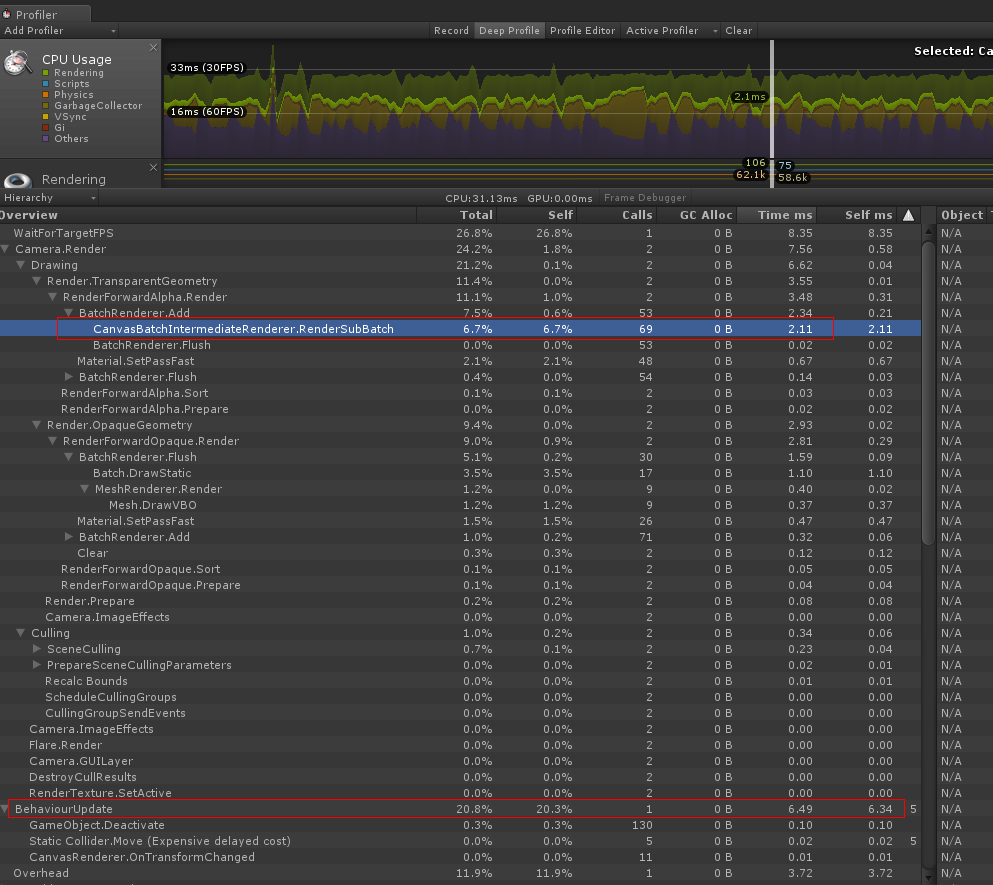
带有Outline的shader在性能上要比Text加Outline的模式有很大优势，但是表现效果不加，需要对shader做进一步优化，另外，带有Outline的shader的模式，Outline的颜色是跟着材质走的，每多一种颜色就要多一个材质球。

## 数据

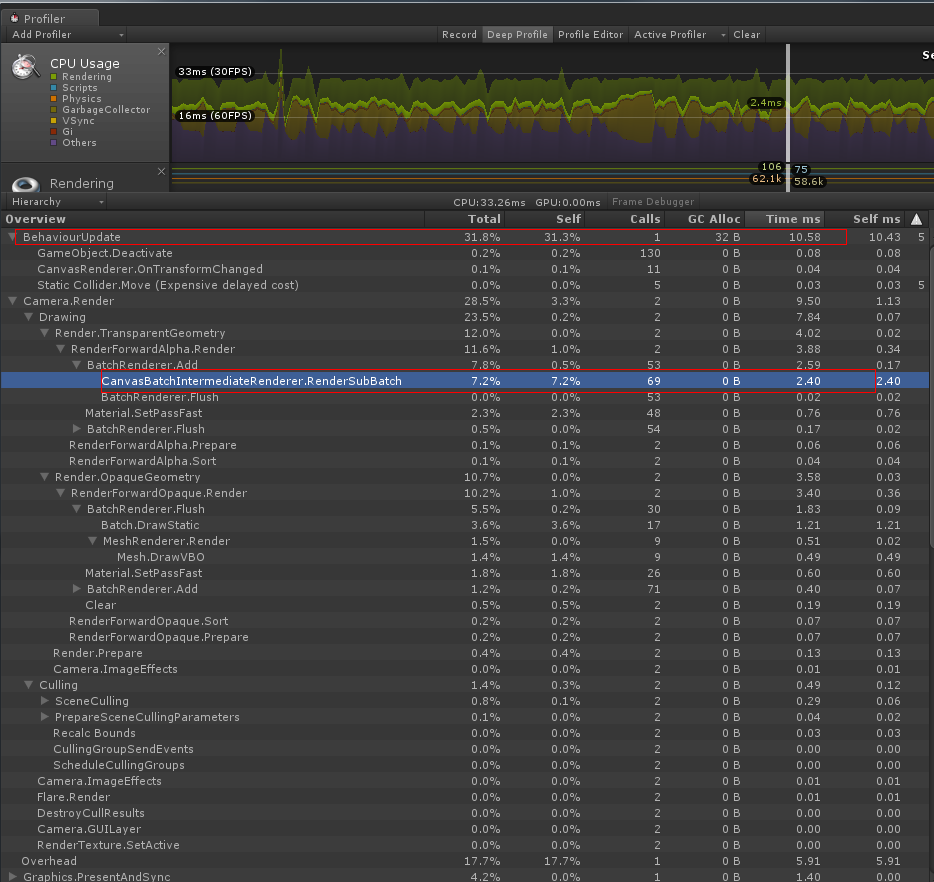
实际项目中的消耗：npc使用一种材质球，玩家使用另一种材质球

## Android（中配设置，高端机Mate8）：

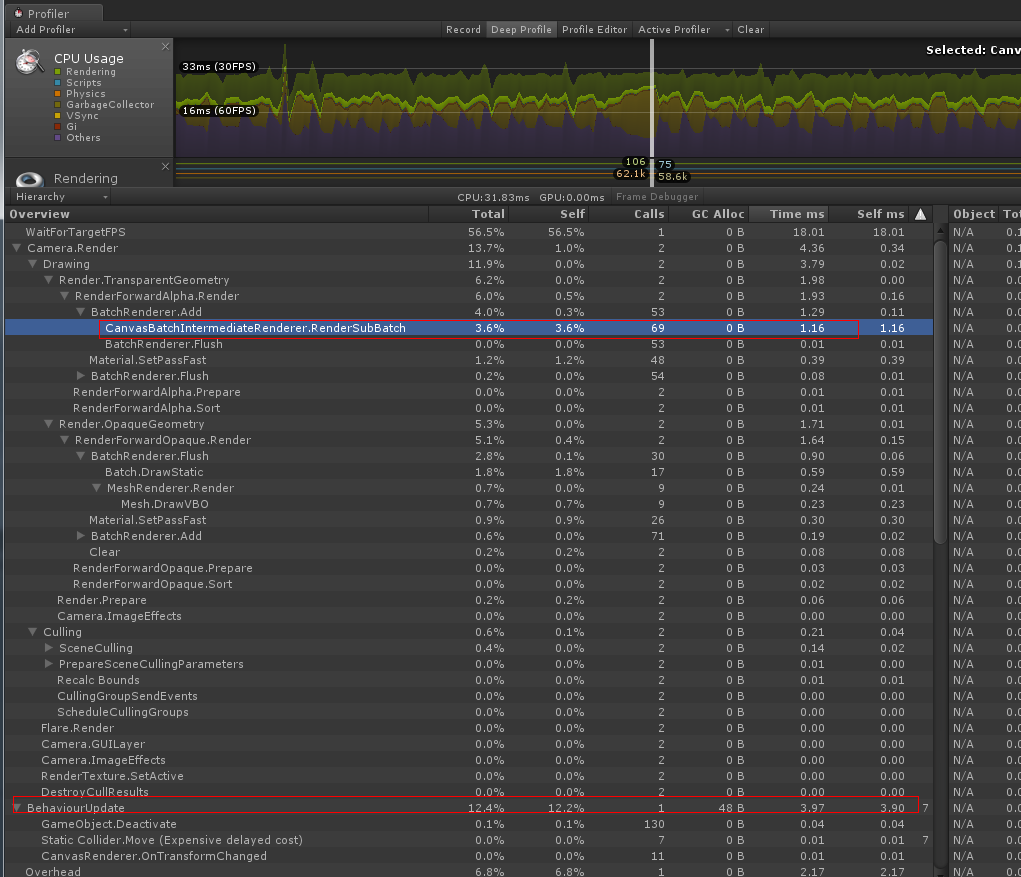
均值



峰值

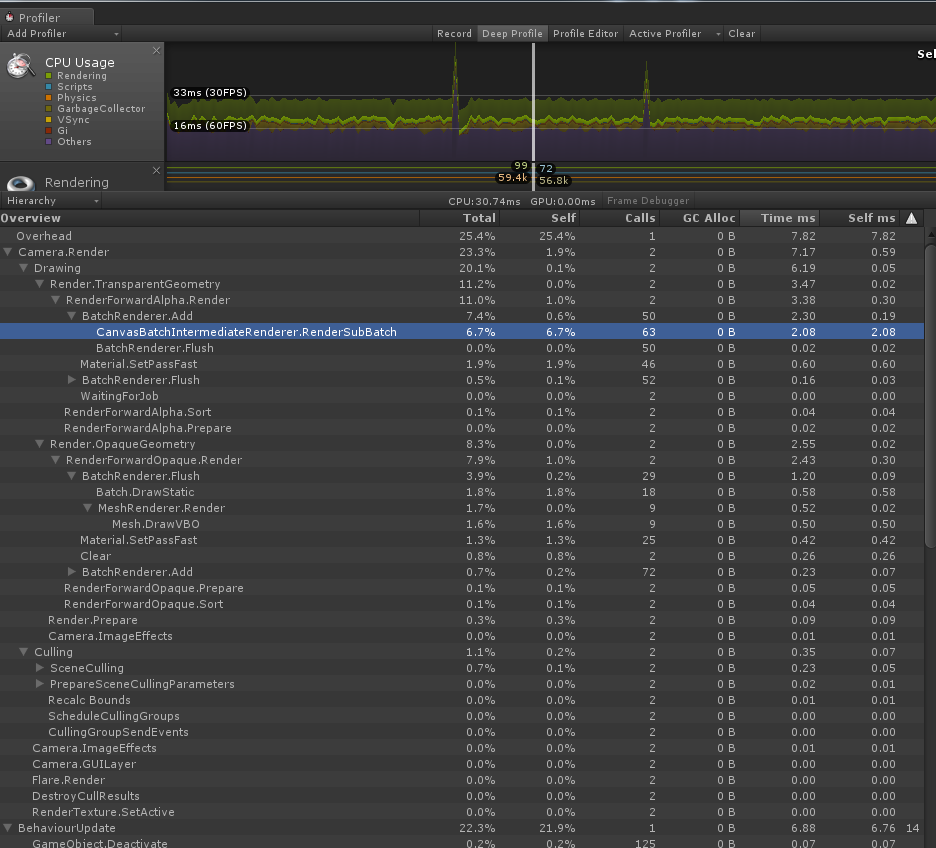


低值：

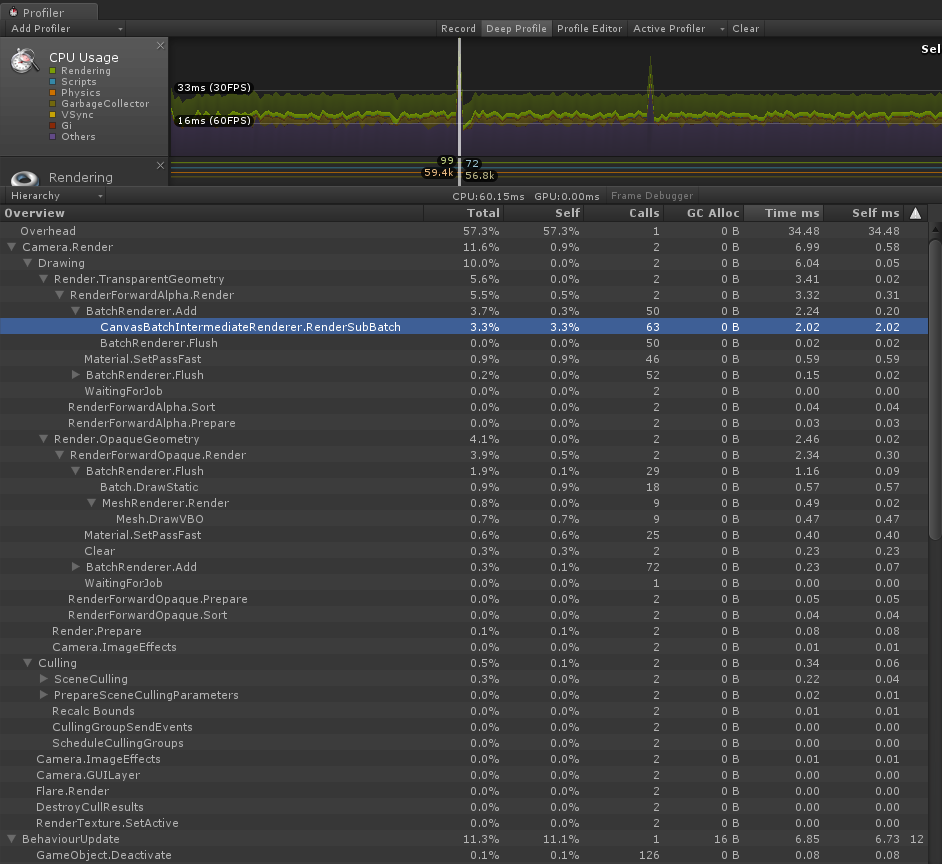


## Android（中配设置，中端机：红米note3）：

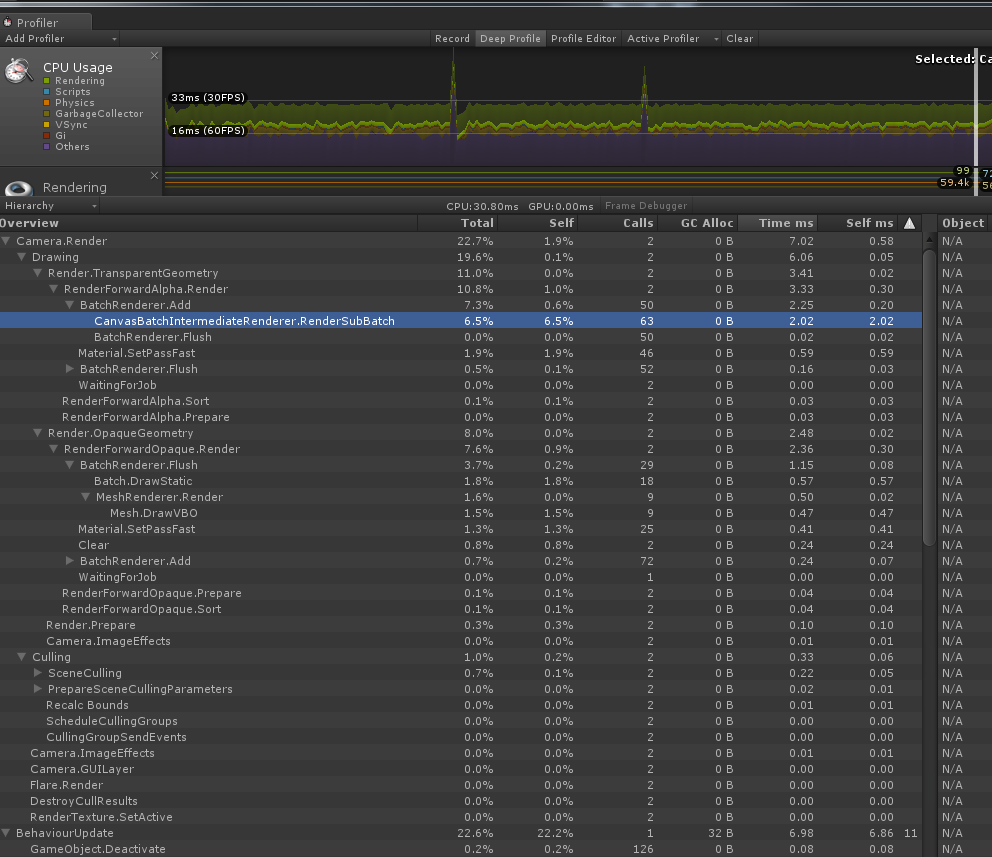
均值



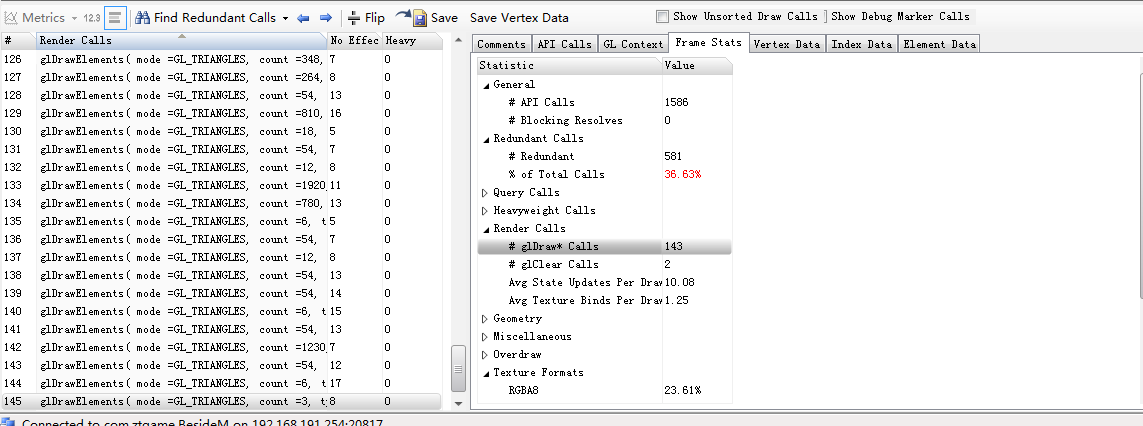
峰值：



低值：

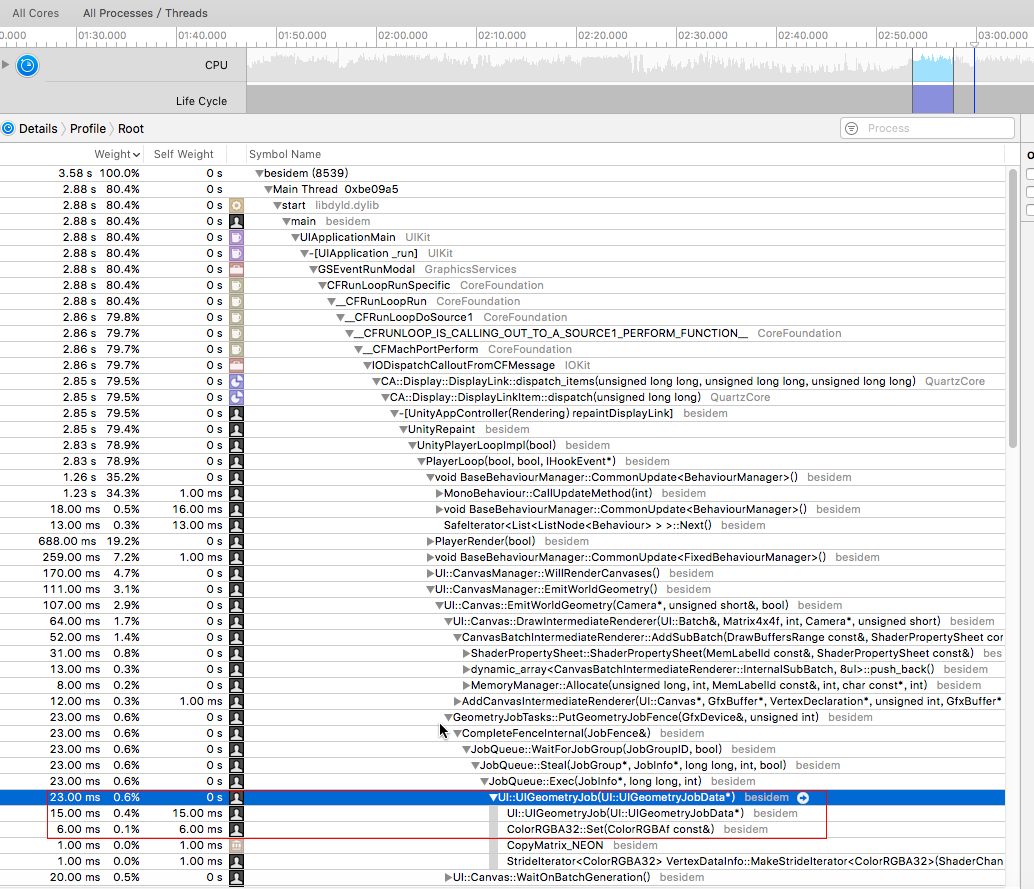


Android上的Drawcall总数：146：



## IOS（Iphone 6）：

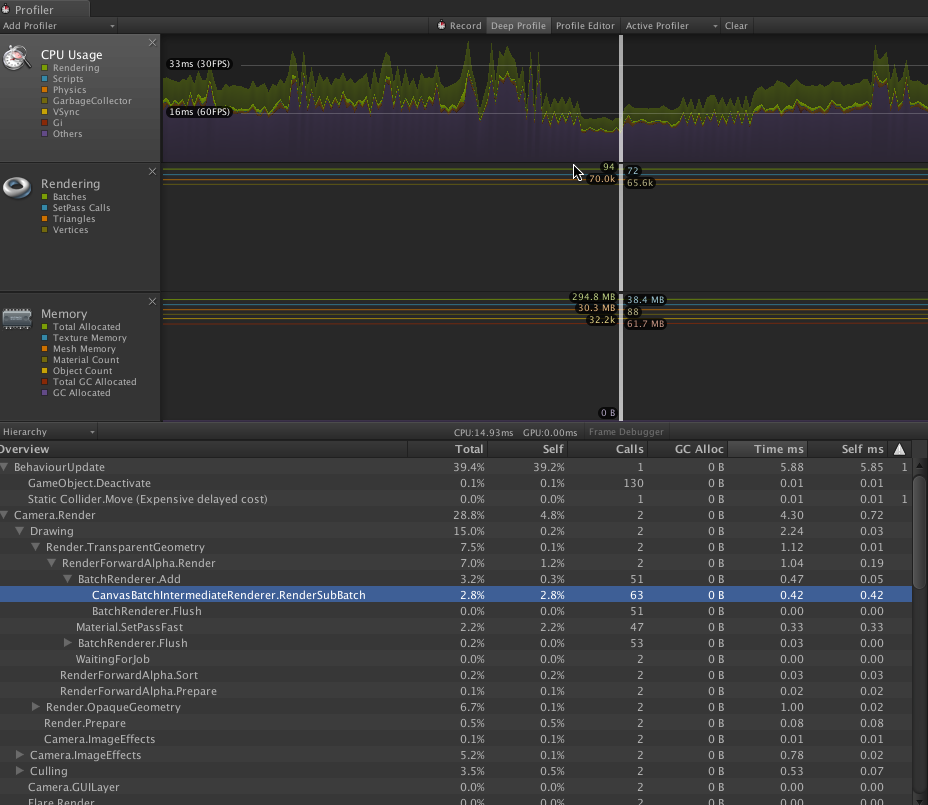
Instruments：



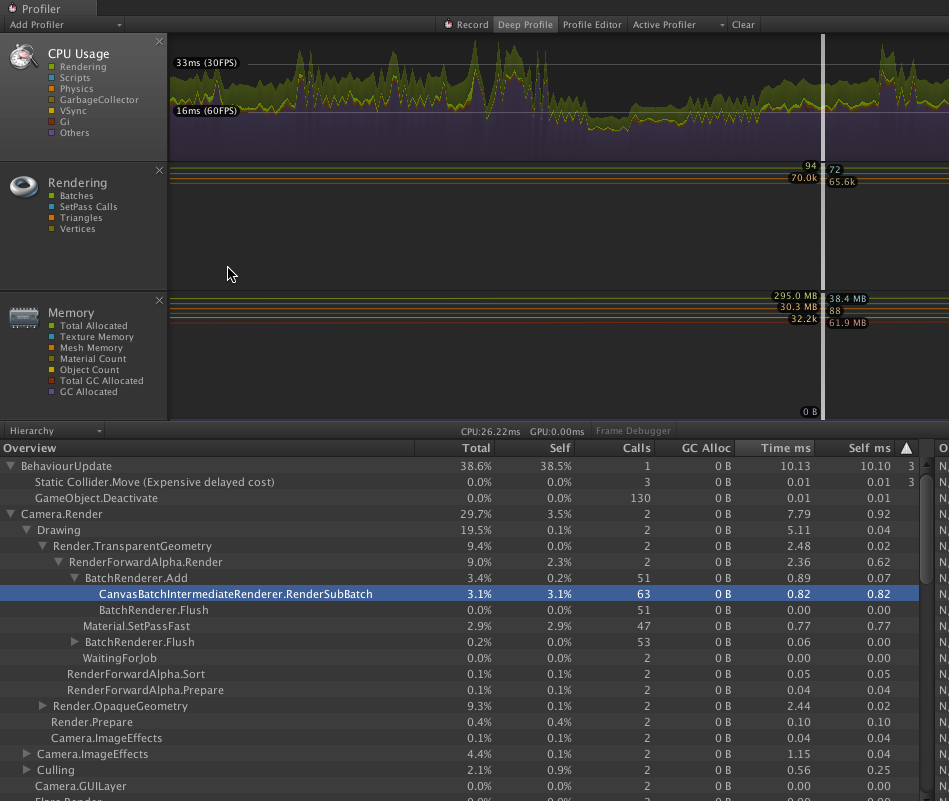
UIGeometryJob和ColorRGB32的消耗为15ms和6ms

Profiler：

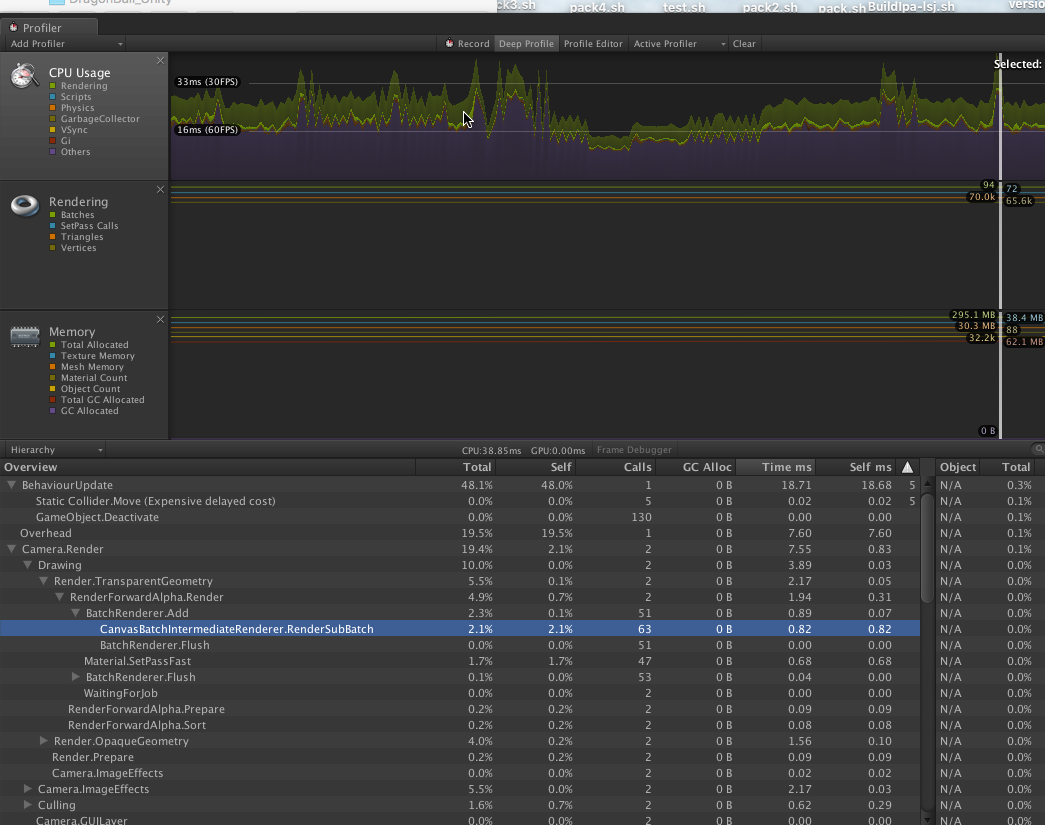
低值：



均值：



峰值：



## PC

因为发包使用上周的Android和Ios资源，内外服务器修改了Proto，所以导致玩家模型显示不出来，所以在内外截取Drawcall数据时，把玩家模型隐藏掉截取。

总Drawcall数：133



玩家和Npc名字占用的Drawcall：6

