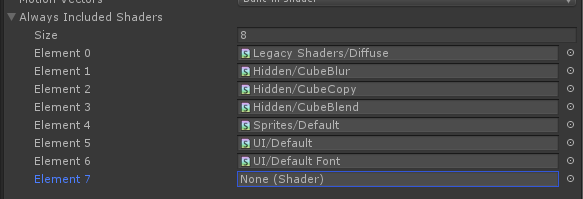
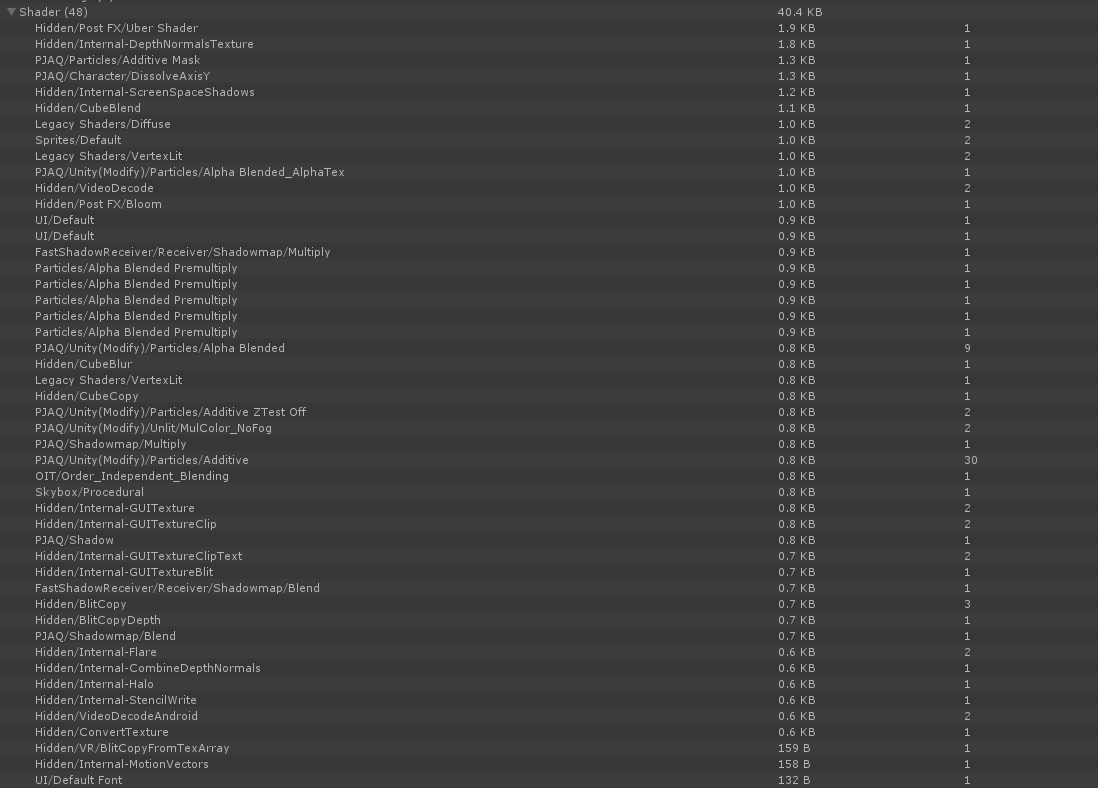
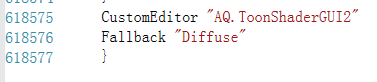
1、在进入游戏的时候，unity会把最基础的shader加载进来，根据graphicssetting里的下图，一共28个，为什么有那么多，是因为这些shader里面都有fallback，这个是一定会加的。

2、之后我们进入主城，并不是把所有shader一股脑加进来，而是用到什么shader加什么shader。

大概有50个（截图是48个，因为我把天空盒和任务shader删了）。

3、但是shaderlab就很大了，28个基本shader， shaderlab只有0.6M。而50个则到达26M。接着我试着把shader删了试试，首先想到的就是人物上的shader，测试过后人物的shader（toonV5）是占得较多的，有5.2M。

因为toonV5的变种完后有惊人的62W行（536个变体）

文件大小有20.18M

但是shaderlab仅占5.2M.....

4、其他的按照预测应该是（bloom后处理应该是它）占得最多，因为它有320W行（2592个变体）

那么大100.74M

吓尿！！

果不其然，我把这个shader删了，shaderlab只有0.8M了，也就是说这个shader，它占了shaderlab的19.5M！！！！



5、结论。

（1）实际上我们自用的shaderlab占用挺少的，最多的人物也就5M，看了下uber实在是太复杂了（用到多个自写cg库）。

（2）我们如何算出shader所占的内存？

先给个数据：

文件20M = shaderlab 5.2M

文件113M = shaderlab 19.5M

大概5倍。

首先先看shader是如何编译的，根据文章：

http://blog.csdn.net/qq\_14939027/article/details/52748015?locationNum=11

https://www.zhihu.com/question/30087487?sort=created

https://docs.unity3d.com/Manual/OptimizingShaderLoadTime.html

shader似乎真的是在运行时编译，而不是在打包时编译好（因为太多变体会使包体放大好多倍）。



也就是在运行时自会把shader加载到内存，使用时才创建对应的变体。而对shader进行编译的东西，就是shaderlab，shaderlab编译好后会把变体cache下来，所以才会这么大（那是否会释放？）。

这里又有另外一个疑问，我一台设备应该只会对应一个shader的其中一个变种啊？就算你一共100M的变种有3000个，那么一个也仅仅30KB啊，怎么会有20M。

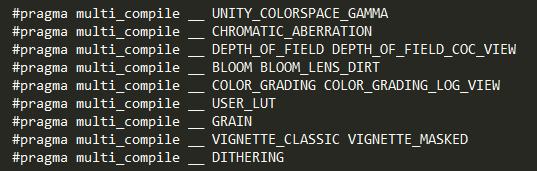
我找了下有下面这两篇文章。

https://www.cnblogs.com/Esfog/p/Shader\_Variant.html

https://www.cnblogs.com/cpxnet/p/6365446.html

意味着你只要使用了multi\_compile，unity就会生成所有keyword组合的变种。

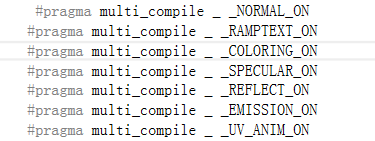
对于Uber



9个，C(1,9)+C(2,9)+C(3,9)+.....+C(9,9) = 761

761/2592\*100.74 = 29.57M，考虑到有压缩，可能可以压到19.5M。

而对于ToonV5



7个，C(1,7)+C(2,7)+C(3,7)+C(4,7)+C(5,7)+C(6,7)+C(7,7) = 127

127/569\*20.18 = 4.5M < 5M，唔.....如果加上压缩，就更小了......

虽然算出个大概，但是还是有误差，所以应该有哪里算错了（或者是观察错了）。

欢迎指正。