

AssetBuddle 优点：减小压缩包、资源更新、分开安装包和数据包、
AssetBuddle加密不容易被反编译

缺点：读取麻烦 组装麻烦 、各平台打的包互相不兼容、耗内存

a

内存：

- 1, 静态区 只实例化一次
- 2, 堆区 new出来的 引用类型
- 3, 栈区 参数 声明的 值类型的
- 4, 代码区

优化

程序堆：

1, 程序员写的代码

1, string 尽量 stringbuilder string每加一次就会重新分配一块内存 释放之前那块内存，产生垃圾 造成内存浪费|||字符串相加减stringBulider相当于一个list 进行字符串相加减

字符串加减法的话就用stringbuilder

2, foreach for

每一次foreach的时候就会new一个指针出来 指向下一个 foreach的遍历的时候会释放很多的垃圾

1, D11 选.NET 2.0 Subset

ripping Level” 表示从build的库中剥离的力度 每一个剥离选项都将从打包好的库中去掉一部分内容你需要保证你的代码没有用到这部分被剥离的功能，选为“Use micro msclorlib”的话将使用最小的库

本机堆：

Unity 所占的内存。

1, 尽量减少在Hierarchy对资源的直接引用，而是使用Resource.Load的方法，在需要的时候从硬盘中读取资源，在使用后用Resource.UnloadAsset()和Resources.UnloadUnusedAssets()尽快将其卸载掉

2, 尽量减少代码的耦合和对其他脚本的依赖是十分有必要的, 耦合性是指单例, 脚本依赖啦, 或者是两张照片 用同一张贴图

托管堆:

.NET 平台 托管对象 。 最后引用计数 没有引用了 GC 释放 。

```
GameObject tmpObj = new GameObject();
```

```
tmpObj = null;
```

优化:

1, 减少对GameObject的Instantiate()和Destroy()调用

(提及预加载 切换场景的时候一般要预加载)

2, 在合适的时候, 也可以手动地调用System.GC.Collect()来建议系统进行一次垃圾回收
五秒钟调用一次

3, 多用 缓存池

Fbx:

动画 模型分离 节约内存

物理:

1, 真实的物理 (刚体) 很消耗, 不要轻易使用, 尽量使用自己的代码模仿假的物理

2.对于投射物不要使用真实物理的碰撞和刚体, 用自己的代码处理

3.不要使用mesh collider

4.在edit->project setting->time中调大FixedTimestep (真实物理的帧率) 来减少cpu
损耗

FixedUpdate()

图片:

我们发现过很多次正常不应该出现的情况，就是有些不需要Alpha通道的纹理在导入时保留了Alpha通道。我们建议大家都检查一下，把这些不必要的Alpha通道都关闭掉。

2, 格式：

Android:

Android设备就没有那么好的运气了。由于硬件平台不统一，每个厂商的GPU可能使用不同的纹理压缩格式。所以还是老老实实用PNG比较好。

Etc 1: 没有透明通道的 图片 。兼容所有的机器。

ETC2 : 带有透明通道。 高端机 。

IOS:

当然是用pvr格式。

pvr是iOS设备的图形芯片 [PowerVR 图形](#) 支持的专用压缩纹理格式。它在PowerVR图形芯片中效率极高，占用显存也小。

性能对比可以看这里: [In Depth iOS & Cocos2D Performance Analysis with Test Project.](#)

a.对于有的函数，则可以每个几帧 执行一次

b.通过使用InvokeRepeating函数实现定时重复调用，比如，启动0.5s后每隔1s执行一次

DoSomeThing

c.尽量少使用临时变量，特别是在Update() OnGUI()等实时调用的函数中

d.定时进行垃圾回收

e.优化数学运算，尽量避免使用float，而用int，特别是在手机游戏中，尽量少用复杂的数学函数，比如sin, cos等函数。改除法为乘法，比如 $0.5f * a$ 而不是 $a / 2.0f$

Resources :

现在Unity官方不再推荐大家使用Resources文件夹，我们未来可能把这个功能关闭掉也是有可能的，因为你把资源放在这里面的话，它是有几个缺点的，第一个缺点你游戏启动的时候，比如说我们手机上面启动这个游戏，它第一步会把Resources文件夹内的文件构建一个索引，这样你后面再动态加载资源的时候，他可以在这里面查找这个文件，这样做导致游戏启动比较耗时，会发现启动黑屏的时间很长，这样对玩家来说这样的体验不太好。

另外，构建索引会占用你更多系统的内存资源。所以说我们建议以后大家不要使用Resources文件夹

Audio:

1, 格式:

对于音频设置方面，我们建议iOS使用**mp3**，安卓采用**vorbis**

游戏标准：

1. 首先是安卓2g低端机型要流畅运行在20到25fps，内存不超过350mb
2. IOS1g低端机型流畅运行20到25fps，内存不超过300mb
3. 国内游很多1g安卓。还有场景加载速度，启动过场场景速度尽量控制在5秒以内。还有优化是对性能和效果的权衡，这点是非常重要的，不能只考虑性能而不讲究美术效果，或者反过来只追求美术效果，玩都不能玩的话，游戏也是不成功的。