**纹理**

1. 纹理格式

它不仅影响着纹理的内存占用，同时还决定了纹理的加载效率

IOS下：

a.普通不透明：PVRTC

b.普通透明：PVRTC

Android下：

普通不透明：RGB ETC

普通透明：RGBA 16BIT （暂时不做透明通道的分离）

由于ETC、PVRTC等格式均为有损压缩，因此，当纹理色差范围跨度较大时，均不可避免地造成不同程度的“阶梯”状的色阶问题。所以尽量减少纹理的色差范围，使其尽可能使用硬件支持的压缩格式进行储存。

（2）纹理尺寸

尽可能降低纹理尺寸，最大1024\*1024

所有纹理的长宽最好是正方形，而且长度值最好是2的整数幂

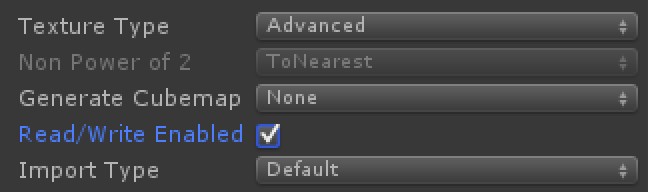
现在角色脸/头发的贴图大小要减少

（3） Mipmap功能

3D纹理开启Mipmap功能

（4）Read & Write

关闭纹理资源的“Read & Write”功能，否则内存消耗翻倍。



上图是错误的用法，不要勾选！！！！！

（5）纹理数量

使用纹理图集（一张大贴图里包含了很多子贴图）来代替一系列单独的小贴图。它们可以更快地被加载，具有很少的状态转换，而且批处理更友好

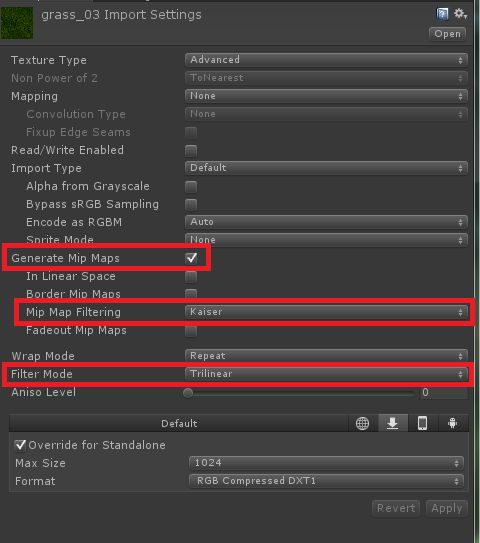
（6）地表贴图的特殊处理

为了解决地表贴图的锯齿问题，在贴图设置里需要设置合适的纹理过滤参数：

Generate Mip Maps 勾选

Mip Map Filtering Kaiser

Filter Mode Trilinear



（7）其他

在非地面和地板模型对象上，不要使用Anisotropic filtering

UV值范围尽量不要超过（0, 1）区间，尽量保证UV值不越界，这对于纹理拼合优化很有帮助

**网格**

（1） Normal、Color和Tangent

Mesh资源的文件体积和内存占用。其中，Color数据和Normal数据主要为3DMax、Maya等建模软件导出时设置所生成，而Tangent一般为导入引擎时生成。如果项目对Mesh进行Draw Call Batching操作的话，那么将很有可能进一步增大总体内存的占用。

这些数据主要为Shader所用。查看该模型的渲染Shader中是否需要这些数据进行渲染，如果不用删除这些选项。

（2）多余的组件

如果不需要，不要附加Animation / Animator

（3）关闭Read/Write功能

**粒子特效**

（1）每个粒子发射器发射的最大粒子数不超过50个。

（2）单个特效通常控制在5-10个发射器

（3）粒子大小    如果可以的话，粒子的size应该尽可能地小。对于非常小的粒子，建议粒子纹理去掉alpha通道。

（4）不要开启粒子的碰撞功能。非常耗时