# 什么是图集？

在使用3D技术开发2D游戏或制作UI时（即使用GPU绘制），都会使用到图集 ，那么什么是图集呢？准确的说法图集是一张包含了多个小图的大图和一份记录了每个小图id、位置、尺寸等数据的数据文件 。

## 为什么要用图集？

在GPU已经成为PC、手机等设备的必备组件的现在，把所有显示的绘制操作交给专门处理图像的GPU显然比交给CPU更合适，这样空闲下来的CPU可以集中力量处理游戏的逻辑运算。

而GPU处理图像的做法和CPU是不一样的，在GPU中，我们要绘制一个图像需要提交图片（即纹理）到显存，然后在进行绘制（这个过程称为一次DrawCall），那么如果我们一帧要绘制100个就需要提交100次图片，如果使用包含了这100图片的图集，只需要一次提交即可，即一次DrawCall就搞定，处理效率上会有很大的提升。

另外使用图集也方便管理和归类各个模块的图片，可以通过一次加载和一次卸载完成多个图片的处理，同理，加载次数也下来了，可以提升运行效率。

## CPU与GPU的限制

GPU一般具有填充率(Fillrate)和内存带宽(Memory Bandwidth)的限制，如果你的游戏在低质量表现的情况下会快很多，那么，你很可能需要限制你在GPU的填充率。

CPU一般被所需要渲染物体的个数限制，CPU给GPU发送渲染物体命令叫做DrawCalls。一般来说DrawCalls数量是需要控制的，在能表现效果的前提下越少越好。通常来说，电脑平台上DrawCalls几千个之内，移动平台上DrawCalls几百个之内。这样就差不多了。当然以上并不是绝对的，仅作一个参考。

## 如何使用图集

在Unity中我们只要使用小图片即可，可以通过设置图片的Packing Tag来指定小图会被打包到的图集，比如2个小图的Packing Tag都叫“MyAtlas”，则Unity会将这两个小图打包到名为“MyAtlas”的图集中。

注意图片不能放在Resources文件夹下面，Resources文件夹下的资源将不会被打入图集。

是否打包图集的控制选项：Editor->Project Settings 下面有sprite packer的模式。Disabled表示不打包图集，Enabled For Builds 表示只有打包应用的时候才会打包图集，Always Enabled 表示始终会打包图集。

在Windows->Sprite Packer 里，点击packer 在这里你就可以预览到你的图集信息，图集文件被保存在和Assets文件夹同级的目录，Libary/AtlasCache里面。图集的大小还有图集的格式等等很多参数我们都是可以控制的，也可以通过脚本来设置。

通过设置我们就可以发现多个同一Packing Tag的小图放到场景中只会消耗一个DrawCall，表示我们的图集已经开始起作用了。

# Unity用到的基本图片格式

Unity3D引擎对纹理的处理是智能的：不论你放入的是PNG，[PSD](http://psd.knowsky.com/)还是TGA，它们都会被自动转换成Unity自己的Texture2D格式。

在Texture2D的设置选项中，你可以针对不同的平台，设置不同的压缩格式，如IOS设置成PVRTC4，Android平台设置成RGBA16等。嗯，非常的智能。

tga是无损支持透明的无压缩格式，dds有一点点压缩，png是无损压缩，对效率要求高的可以tga，dds，ui一般png，psd不可取

## 格式对比



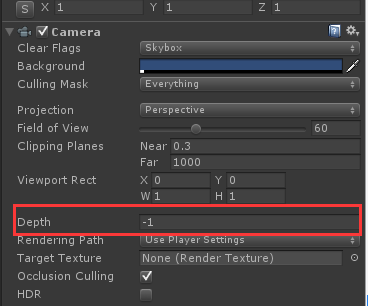
2的n次方幂天然被GPU接受

# 为什么要关心渲染顺序？

1.如果是2D游戏，渲染顺序关系着每个层次的显示先后，比如UI在游戏内容前面，游戏内容又有多层次。举一个简单的例子，在横版2D游戏中，经常会用到多层滚动的背景，把游戏物体分层管理起来，可以有效的减少出错几率，很好的控制显示效果。

2.对于3D游戏，游戏内容是3D的，UI一般是2D，有很多时候需要把某个模型啊，粒子特效啊，放在界面上，这样就有一个问题，3D物体和2D界面的先后关系，比如有些界面是在模型之上的，有些在下面，尝试过很多种办法，都能实现需求，但不是每种办法都是那么舒服的。

## unity中控制渲染顺序的方式

Camera是unity中最优先的渲染顺序控制。depth越大，渲染顺序越靠后。   


sortingLayer 和 sortingOrder   
对于这个属性，我也是云里雾里的，不是太明白。按照字面意思是层的排序，Canvas和Renderer都有这个属性，unity官方就一句话带过。优先级仅次于Camera的depth。

看网上的博客，一般都写着sorting layer是比Camera低一层级的控制渲染顺序的属性。然后我做了很多尝试，发现并不是所有Renderer组件这个属性作用，根据尝试的结果，做如下总结：

Canvas中的sorting layer可以控制Canvas的层级，这是ugui中的东西，下面会讨论的；

GameObject中的renderer属性就是挂在该GameObject的Renderer组件，Renderer是渲染组件的基类，下面有多个派生类，经过测试，目前知道的只有SpriteRenderer的sorting layer和sorting order能控制渲染顺序。所以我也觉得，sorting layer和 sorting order就是unity 对2D游戏所支持的吧；

renderer组件有如下几类： 

RenderQueue   
这是unity中的一个概念，（至少之前在opengl中没听过），大致意思就是渲染顺序，那无疑就是控制渲染顺序的嘛。

所以一般设置材质的renderQueue或直接在shader中设置。

在ShaderLab中，有4个提前定义好的render queue，你可以设置更多的在他们之间的值：

Background ：表示渲染在任何物体之前   
Geometry（default）：渲染大多数几何物体所用的render queue   
AlphaTest：用于alpha测试   
Transparent：用于渲染半透明物体   
Overlay：渲染所有物体之上

空间深度   
在摄像机坐标系下的Z轴，控制着该相机下的物体的深度，在fragment shader中进行深度测试，这样就控制了渲染到屏幕的顺序。（可以看看深度测试的具体原理，就明白怎么控制显示的先后顺序了）