

Unity3D 内置了很多 Shader, 文档很详细, 自己翻一下. 便于加深印象.

首先先解释下 Unity3D 的 Shader. Unity 里面的 Shaders 是使用一种叫 ShaderLab 的语言编写的, 它同微软 的 .FX 文件或者 NVIDIA 的 CgFX 有些类似. 传统意义上的 vertex shader 和 pixel shader 还是使用标准的 Cg/HLSL 编程语言编写的。(因此 Unity 文档里面的 Shader, 都是指用 ShaderLab 编写的代码)

然后我们来看下 Unity3D 自带的 60 多个 Shader。这些 Shader 被分为五个大类: Normal, Transparent, Transparent Cutout, Self-illuminated, Reflective。由于数量比较多, 将分几个篇幅一一介绍。

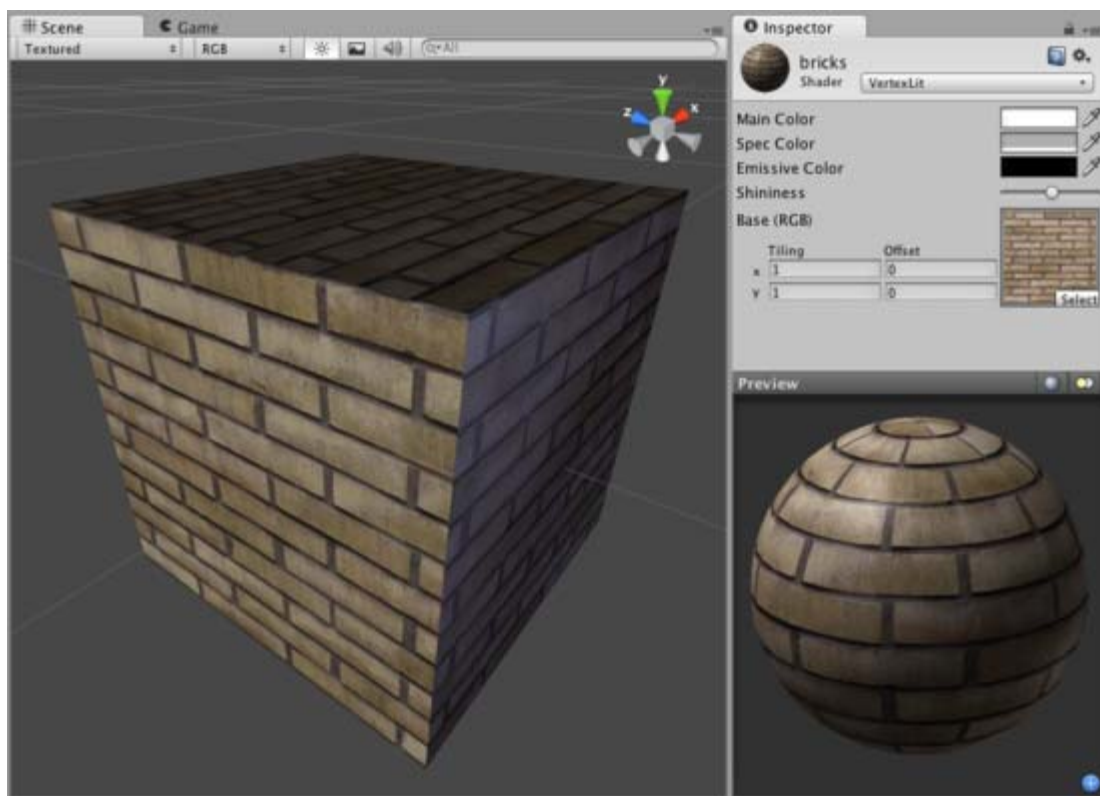
(一) Normal Shader Family

这个家族一共 9 个 Shader, 都是针对不透明的对象的。

(1) Vertex-Lit:

最简单的一种 Shader 之一, 所有照射在该物体上的光在一个 Pass 里面渲染完, 光源只在顶点计算。所以不会有任何基于像素渲染得效果, 比如 说: normal mapping, light cookies 和 shadows. 这个 shader 对模型的剖分 (将一个物体从几何描述变为多边形表示的过程) 非常敏感, 如果你将一个点光源放在很靠近一个立方体的一个顶点那里, 并且对立方体使用这个 shader, 光源只会在角落计算。基于像素光照的 shader 对剖分没有要求, 在表现圆形高光上效果也很好。如果上述情况时你想要的效果, 你可以考虑使用一个基于像素光照的 shader 或者增加模型的剖分。(增加顶点数)

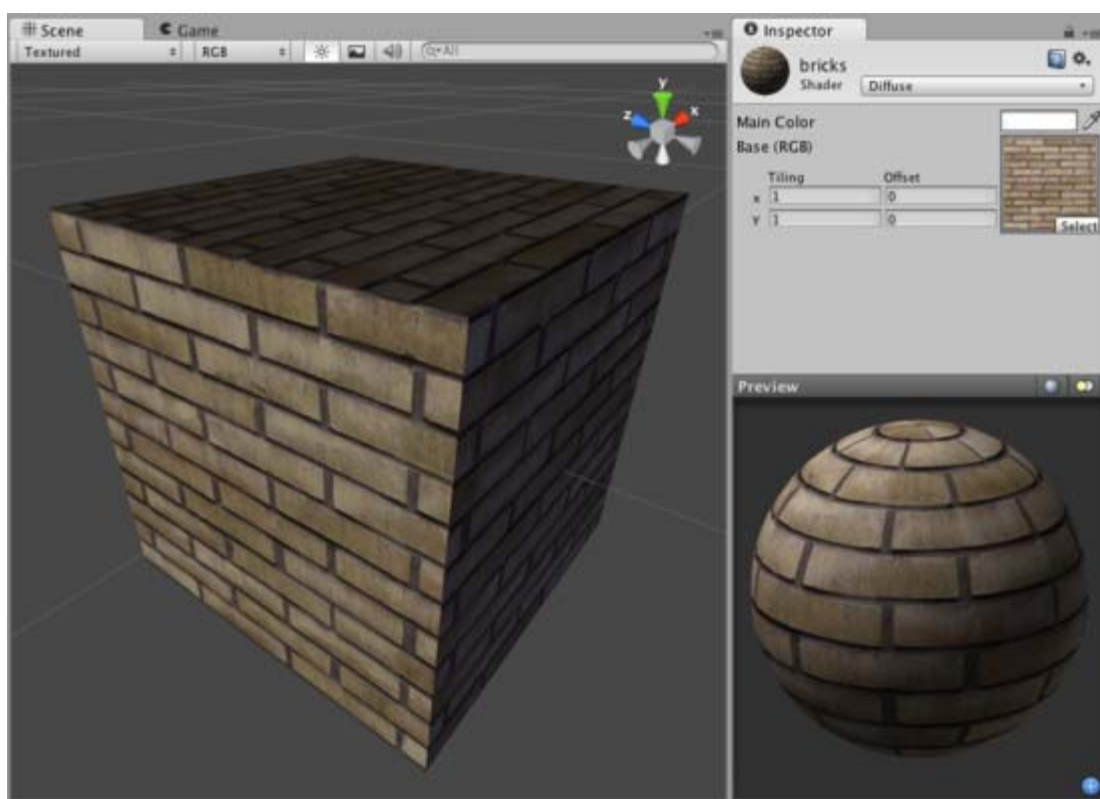
总的来说, 这个 shader 的渲染代价比较小。这个 shader 包含了 2 个 subshader, 分别对应可编程管线和固定管线。是所有 硬件都支持的一个最基本的 shader。如果设备支持可编程管线, 那使用可编程管线的 subshader, 否则使用固定管线的。



(2) Diffuse:

Diffuse 基于一个简单的光照模型-Lambertian, 光照强度随着物体表面和光入射角夹角的减小而减小(即光垂直于表面时强度最大)。光照的强度只和该角度有关系, 和摄像机无关。由于这是一个基于像素光照的 shader, 因此他有这类 shader 的优势, 同时他需要设备支持可编程管线, 如果设备不支持, 则自动使用 Vertex-Lit 这个 Shader。

总的来说, 这个 shader 渲染代价比较小。

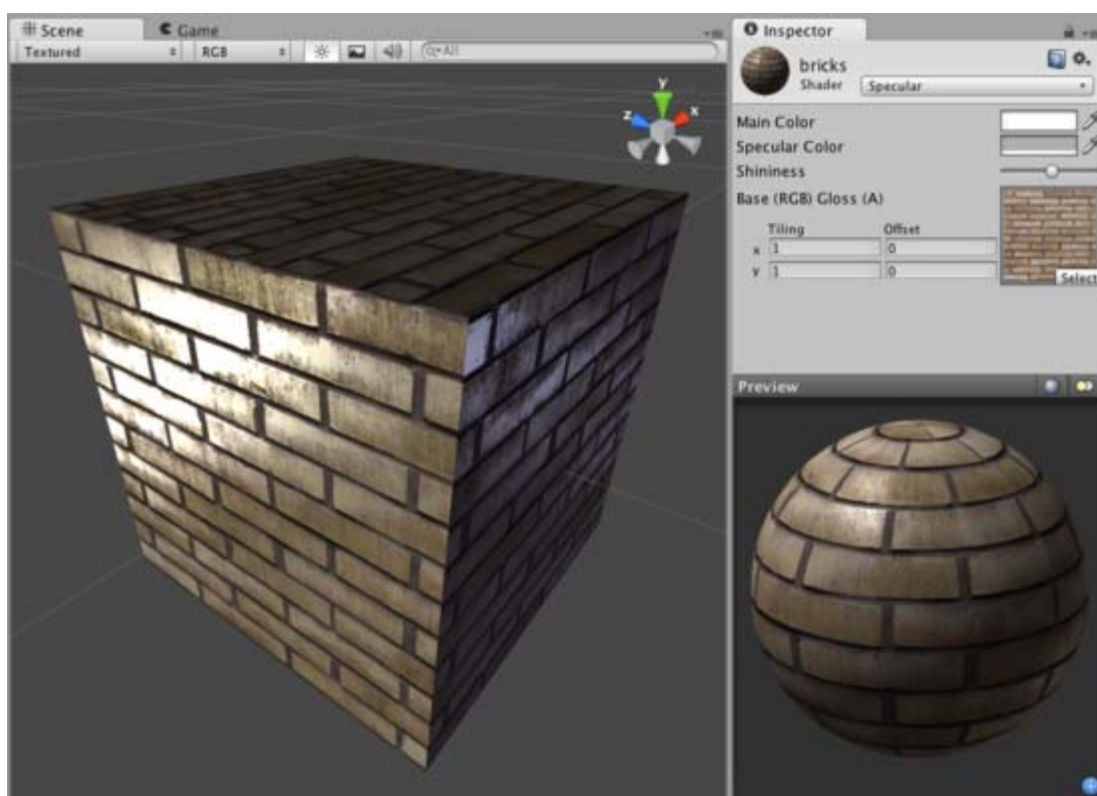


(3) Specular

Specular 使用和 Diffuse 相同的光照模型，但是添加了一个和观察角度相关的反射高光。这个被称为 Blinn-Phong 光照模型。他包含了一个反射强光，这个 反射高光物体表面角度，光的入射角度以及观察者角度都有关系。这种高光计算方法实际上是对实时光源模糊反射的一种具有可行性的模拟。模糊的等级通过 inspector 里面 Shininess 这个变量控制。

主纹理的 alpha 通道被用来当作 Specular Map 来使用(有时候也称 gloss map)，它定义了物体的反光率。纹理中 alpha 里面全黑 的部分将完全不反光(即反光率为 0%)。而全白的的部分反光率为 100%。这个 Map 在你的物体在不同的部分有不同的反光率的时候非常有用。比如说，锈迹 斑斑的金属会反光率低，而磨光的金属反光率比较高。口红的反光率比皮肤高，而皮肤的反光率比棉质衣服高。一个精心制作的 Specular Map 将会让玩家身临其境。如果目标设备不支持可编程管线，则自动使用 Vertex-Lit 这个 Shader。

这个 shader 的渲染代价会比较大。



(4) Bumped Diffuse

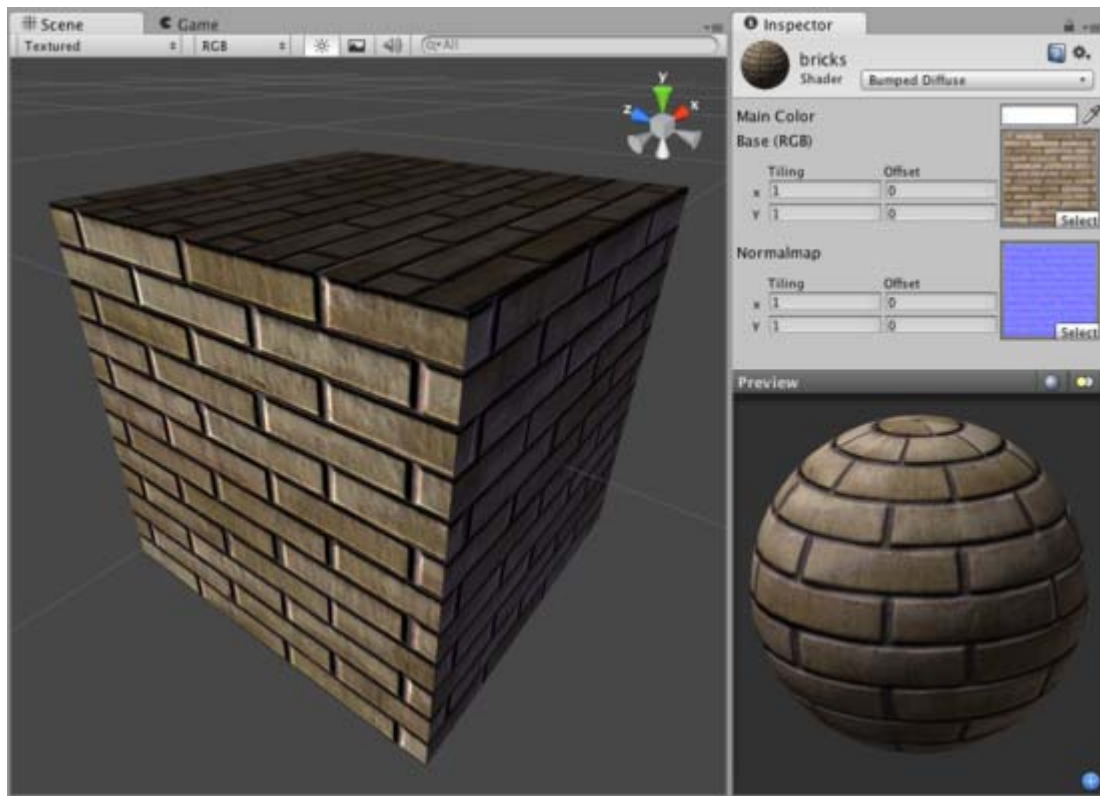
同 Diffuse Shader 一样，这个 Shader 基于 Lambertian 光照模型，同时使用了 normal mapping 技术来增加物体表面细节。相对于通过增加剖分来表现物体表面细节的方式，normal mapping 并不改变物体的形状，而是使用一张称为 Normal Map 的特殊纹理来达到这种效果。在 normal map 中，每个像素的颜色代表了该像素所在物体表面的法线，然后通过这个法线（而不是通过物体模型计算而来的法线）来计算光照。可以说 Normal Map 在计算光照的过程中“高效地修改”了整个模型。

如何创建 Normal maps：你可以通过导入一张普通的灰度图（白色表示凸起，黑色表示凹进）Unity 会自动将它转换为 Normal Map。

技术细节：这里使用 Normal map 是一种“Tangent space Normal Map”，Tangent space（正切空间）是一个跟随模型物体表面的空间。在这个空间中，z 轴始终从表面指向外面。Tangent space Normal Map 相对于另一种被称为“Object space Normal Map”来说有点复杂，但是他有一些优势。1）可以使用在各种奇形怪状的表面 2）便于在同一物体不同区域或者不同物体间复用

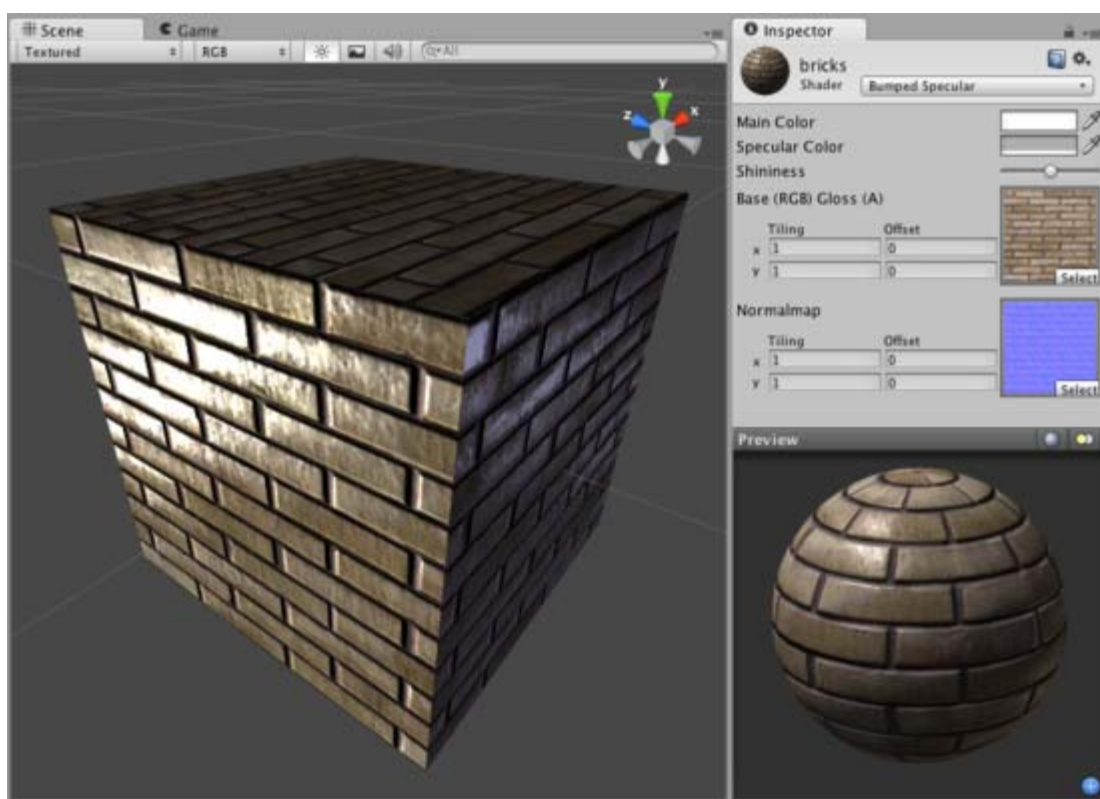
具体关于 Tangent Space Normal Map 和 Object Space Normal Map 的区别，下次再讲。

如果调用这个 shader 失败,则会调用 Diffuse 这个 shader,一般而言,该 shader 渲染代价低。



(5) Bumped Specular

和 Specular 一样的光照模型,相比 Specular 而言,它使用一张 Tangent Space Normal Map 来描述物体表面法向量的变化,来增加物体细节。(关于 normal map 具体见 Bumped Diffuse) 如果调用失败,则使用 Specular 这个 shader,一般而言,这个 shader 的渲染代价会比较大。

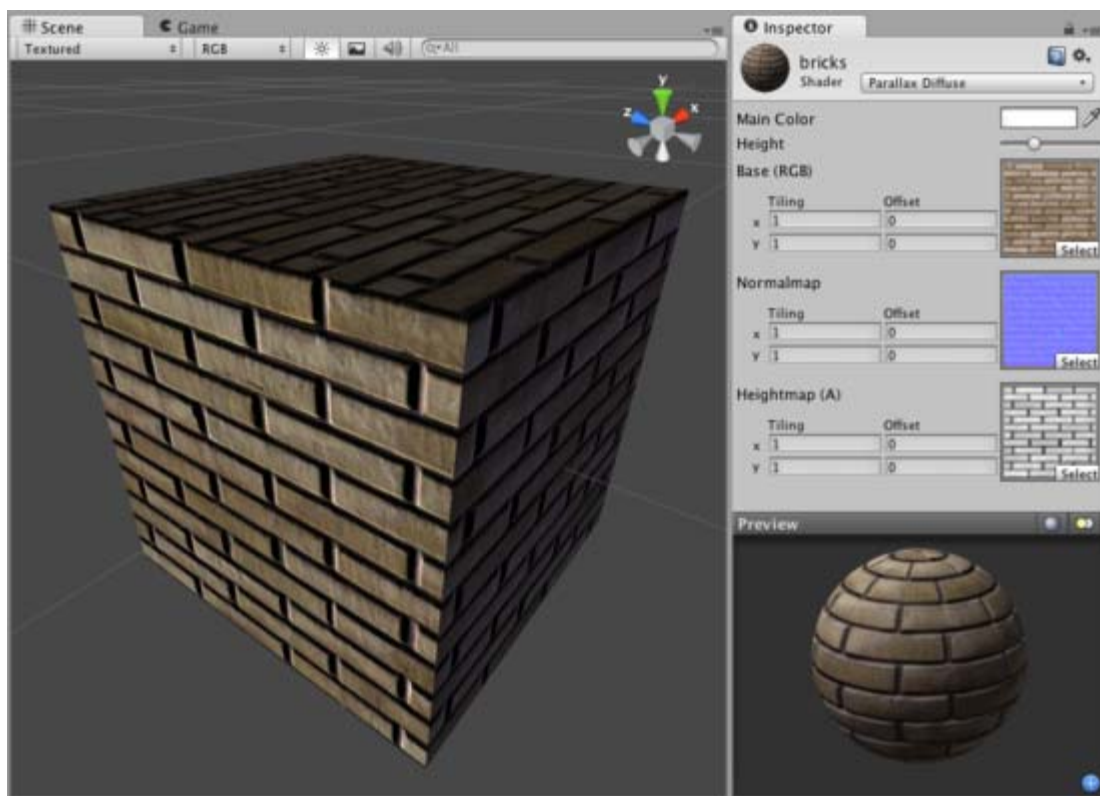


(6) Parallax Diffuse

Parallax Normal mapped 与传统的 normal mapped 一样，但是对“深度”的模拟更佳。额外的深度效果是通过 Height Map (高度图) 来实现的。Height Map 在 Normal map 的 alpha 通道里面保存。全黑表示没有高度，而白色表示有高度。通常这用来表现石头或者砖块间的裂缝。

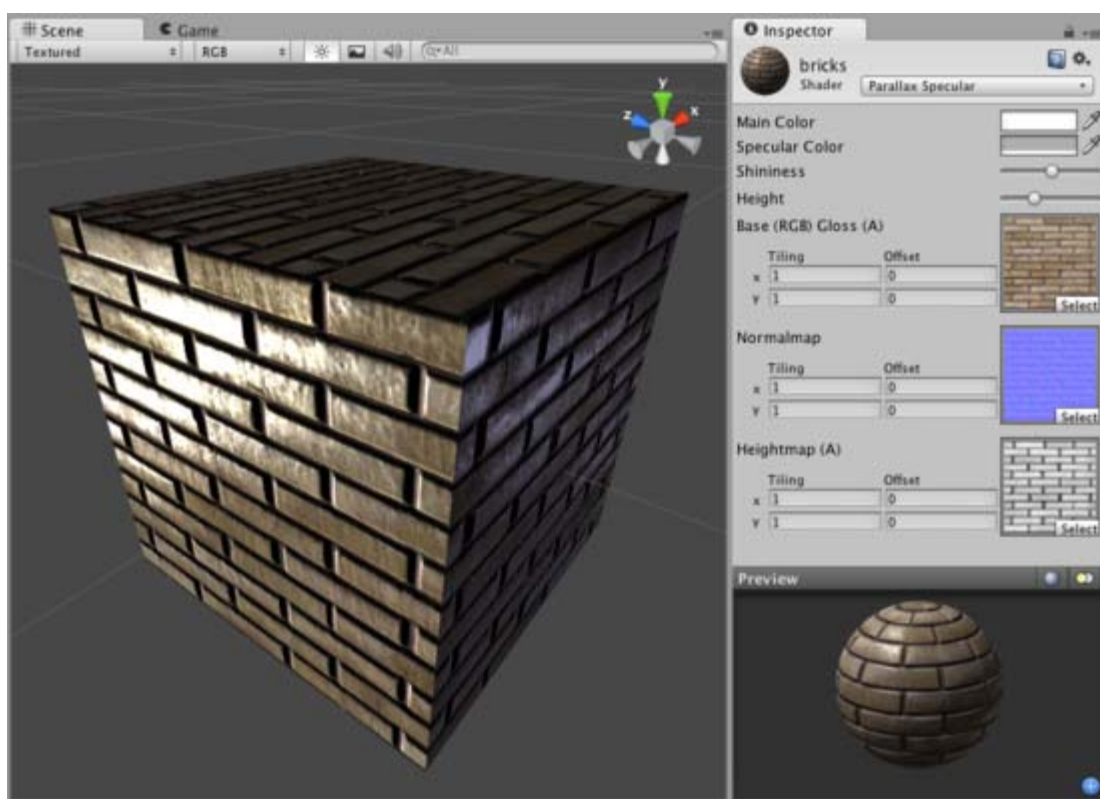
Parallax mapping 的技术很简单，因此会有些人工痕迹或者不太正常的效果出现。尤其是，急剧陡峭的高低转换在高度图里面应该避免。在 inspector 里面调整 height 数值来调整高度的范围，有时候会造成物体不真实，凌乱的情况。因此建议使用高度变化平缓的高度图或者将高度数值设置比较低，让表面看起来比较平缓。

这个 shader 的渲染代价相比 bumped diffuse 而言更大。如果调用这个 Shader 失败，则自动使用 Bumped Diffuse。



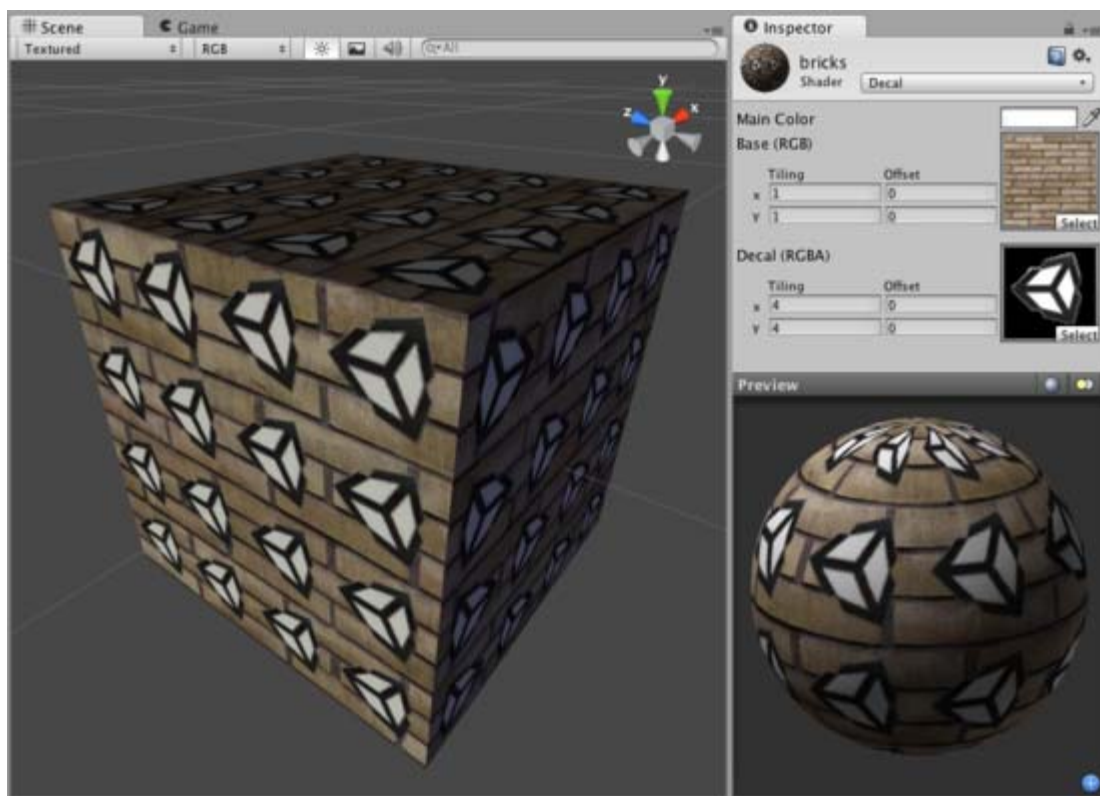
(7) Parallax Specular

与 Bumped Specular 相比，增加了一张 Height Map 来描述深度细节。关于 Height Map，见 parallax Diffuse。如果调用失败，则调用 Bumped Specular 这个 Shader。



(8) Decal

这个 Shader 在 unity 文档里面的描述和 Unity3.0 有明显的实现区别，文档由于比较老，07 年写的，官网下的 Built-in Shader 里面，decal 是使用可编程管线实现的，就是说，如果你的机器不支持可编程管线，会使用 diffuse，因为 diffuse 也不需要可编程 管线，所以只能使用 vertex-lit。这个 Shader 除了主纹理之外，这个 Shader 还是用了第二张纹理用来描述细节。第二张用来 Decal (贴 花) 的纹理使用 alpha 通道来确定是否覆盖主纹理。贴花用的纹理只是对主纹理的补充。比如说你有一个砖砌的墙壁，你可以使用一个砖块的纹理作为主纹理， 然后使用带有 alpha 通道的 Decal 纹理在墙壁的不同地方涂鸦。



(9) Diffuse Detail

这是一个普通的 diffuse shader 加上一些额外数据的 shader。它允许你定义第二张纹理,称为 Detail Texture。当 camera 靠近的时候,Detail Texture 逐渐显示出来,一般用于地形。比如说,当你使用一张低分辨率的纹理拉升到整个地形上的时候。随着 camera 逐渐拉近,低分辨率的纹理开始模糊,这不是我们想要的效果。为了避免这个效果,创建一张 Detail 纹理会将地形 tile 化。在这种模式下,随着 camera 逐渐拉近,额外的细节将会出现以避免出现模糊的效果。

Detail 纹理是覆盖在主纹理上面的。Detail 纹理中深色的部分将会使得主纹理变深,而淡色的部分将会使主纹理变亮,Detail 纹理通常是浅灰色。(与 Decal 里面 Decal 纹理不同的是,Decal 纹理是 RGBA,通过 alpha 控制 Decal Texture 与 Main Texture 的融合,而 Detail 的纹理是 RGB,直接是两张纹理的 rgb 通道分别相乘再*2,就是说,Detail 纹理中颜色数值 = 0.5 不会改变主纹理颜色,>0.5 会变亮,<0.5 加深)

