**材质、着色器和纹理**

Unity 中的渲染采用**材质**、**着色器**和**纹理**。所有这三者之间有着密切的关系。

* **材质**通过包含对所用纹理的引用、平铺信息、颜色色调等来定义表面应使用的渲染方式。材质的可用选项取决于材质使用的着色器。
* **着色器**是一些包含数学计算和算法的小脚本，根据光照输入和材质配置来计算每个像素渲染的颜色。
* **纹理**是位图图像。材质可包含对纹理的引用，因此材质的着色器可在计算游戏对象的表面颜色时使用纹理。除了游戏对象表面的基本颜色（反照率）之外，纹理还可表示材质表面的许多其他方面，例如其反射率或粗糙度。

材质指定了要使用的一种特定着色器，而使用的着色器确定材质中可用的选项。着色器指定期望使用的一个或多个纹理变量，而 Unity 中的材质检视面板 (Material Inspector) 允许您将自己的纹理资源分配给这些纹理变量。

对于大多数正常渲染（例如渲染角色、景物、环境、实体和透明游戏对象、硬表面和软表面），[标准着色器](file:///E:\\UnityDocumentation\\Manual\\shader-StandardShader.html)通常是最佳选择。这是一种高度可定制的着色器，能够以高度逼真的方式渲染多种类型的表面。

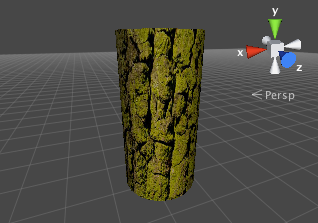
在其他情况下，可能适合采用另外的内置着色器，甚至自己编写的着色器（例如液体、树叶、折射玻璃、粒子效果、卡通、演示性效果或其他艺术效果或者其他特殊效果，如夜视、热视或 X 射线透视）。

有关更多信息，请参阅以下页面：

* [创建和使用材质](file:///E:\UnityDocumentation\Manual\Materials.html)
* [内置标准着色器](file:///E:\UnityDocumentation\Manual\shader-StandardShader.html)
* [其他内置着色器](file:///E:\UnityDocumentation\Manual\Built-inShaderGuide.html)
* [编写您自己的着色器](file:///E:\UnityDocumentation\Manual\ShadersOverview.html)

# 纹理

通常情况下，对象的网格几何形状仅给出粗略的近似形状，而大多数精细的细节由\_\_纹理\_\_提供。纹理就是应用于网格表面上的标准位图图像。您可以看作纹理图像好像是打印在橡胶板上，然后将橡胶板拉伸并固定在网格上的适当位置。纹理的定位是通过用于创建网格的 3D 建模软件完成的。

具有树皮的圆柱体

Unity 可从最常见的图像文件格式导入纹理。

## 用于 3D 模型的纹理

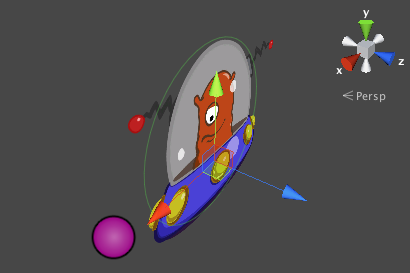
必须使用[材质](file:///E:\UnityDocumentation\Manual\Materials.html)将纹理应用于对象。材质使用称为[着色器](file:///E:\UnityDocumentation\Manual\Shaders.html)的专用图形程序在网格表面上渲染纹理。着色器可实现光照和着色效果，从而模拟许多其他事物的闪亮或凹凸表面。此外，它们还可一次使用两个或更多纹理，将这些纹理组合起来以获得更大的灵活性。

您应该使纹理的尺寸达到 2 的幂次方（例如 32x32、64x64、128x128、256x256 等）。只需将纹理放在项目的 Assets 文件夹中就足够了，它们将出现在 Project 视图中。

导入纹理后，应将其分配给[材质](file:///E:\UnityDocumentation\Manual\class-Material.html)。随后，可将材质应用到网格、**粒子系统\_\_或** GUI 纹理**。通过使用\_\_导入设置 (Import Settings)**，还可将其转换为\_\_立方体贴图 (Cubemap)\_\_ 或\_\_法线贴图 (Normalmap)\_\_，以便用于游戏中不同类型的应用。有关导入纹理的更多信息，请阅读[纹理组件页面](file:///E:\UnityDocumentation\Manual\class-TextureImporter.html)。

## 2D 图形

在 2D 游戏中，为了实现\_\_精灵 (Sprite)\_\_，必须将纹理应用到近似于对象形状的扁平网格。

从 3D 视点观看的精灵

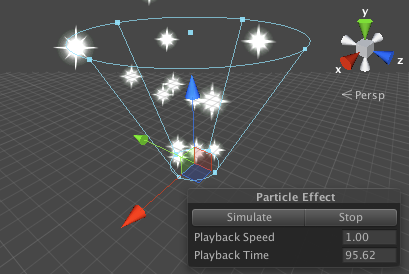
2D 游戏中的对象可能需要一组相关的图形图像来表示动画帧或角色的不同状态。可使用特殊技术来高效设计和渲染这些图像集。请参阅有关[精灵编辑器 (Sprite Editor)](file:///E:\\UnityDocumentation\\Manual\\SpriteEditor.html) 的手册页以了解更多信息。

## GUI

游戏的\_图形用户界面\_ (GUI) 包含了一些不直接在游戏场景中使用但允许玩家进行选择和查看信息的图形。例如，分数显示和选项菜单便是游戏 GUI 的典型示例。这些图形与用于表现网格表面细节的图形明显不同，但它们仍然使用标准 Unity 纹理进行处理。请参阅 [GUI 脚本指南](file:///E:\UnityDocumentation\Manual\GUIScriptingGuide.html)手册章节以了解有关 Unity GUI 系统的更多详细信息。

## 粒子

网格非常适合表示固体对象，但不太适合表现魔法留下的火焰、烟雾和火花等对象。\_粒子系统\_\_可以更好地处理这种类型的效果。粒子\_是一个小型 2D 图形，用于表示一小部分基本为流体或气体的东西，例如烟幕云团。当许多这样的粒子一次性创建完成并开始运动（还可能有随机变化）时，它们可产生非常令人信服的效果。例如，可通过从中心点以极快的速度发出带有火焰纹理的粒子来表现爆炸。此外，可通过从场景中的高处加速落下水粒子来模拟瀑布。

星形粒子系统

Unity 的粒子系统有很多选项可用于创建各种流体效果。请参阅有关该主题的[手册章节](file:///E:\\UnityDocumentation\\Manual\\ParticleSystems.html)以了解更多信息。

## 地形高度贴图

纹理甚至可用于根本不会查看图像（至少不能直接查看图像）的情况。在灰度图像中，每个像素值仅仅是对应于图像中该点处的灰度的数字（比如说，此数字可以是 0..1 范围内的值，其中 0 表示黑色，而 1 表示白色）。虽然可以查看这样的图像，但其实数字像素值还可用于其他用途，这正是\_\_地形高度贴图\_\_的作用。

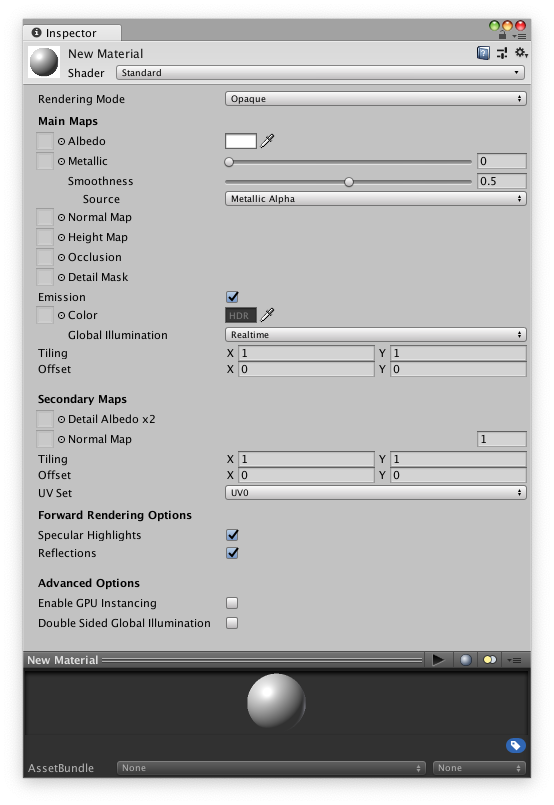
\_地形\_是表示地面区域的网格，而地面上的每个点距离基线都有特定的高度。地形的\_高度贴图\_以规则的间隔将数字高度样本存储为图像中的灰度值，其中的每个像素对应于地面上的网格坐标。这些值不会在场景中显示为图像，但会转换为用于生成地形网格的坐标。

有趣的是，即使不将高度贴图直接视为图像，仍然有一些常用的图像处理技术在应用于高度数据时非常有用。例如，向高度贴图添加噪点将产生岩石地形的效果，而模糊技术将使其平滑以便产生更柔和、绵延的景观。

# 创建和使用材质

要创建新材质，请从主菜单或 **Project 视图\_\_上下文菜单中选择** Assets > Create > Material\_\_。

默认情况下会为新材质指定标准着色器，并且所有贴图属性都为空，如下所示：



创建材质后，可将其应用于对象并在 **Inspector** 中调整其所有属性。要将材质应用于对象，只需将其从 **Project 视图\_\_拖到** Scene\_\_ 或 **Hierarchy** 视图中的任何对象上。

## 设置材质属性

您可以选择希望特定材质使用的着色器。只需在 Inspector 中展开 **Shader** 下拉选单，然后选择新的着色器。所选的着色器将决定可更改的属性。属性可能包括颜色、滑动条、纹理、数字或矢量。如果已将材质应用于 **Scene** 视图中的活动对象，您将看到属性更改会实时应用于对象。

将\_\_纹理\_\_应用于属性时可采用两种方法。

1.将其从 Project 视图拖到纹理方块上 1.单击 **Select** 按钮，然后从显示的下拉列表中选择纹理

## 内置着色器

除了[标准着色器](file:///E:\\UnityDocumentation\\Manual\\shader-StandardShader.html)，还有许多其他类别的内置着色器可用于特殊目的：

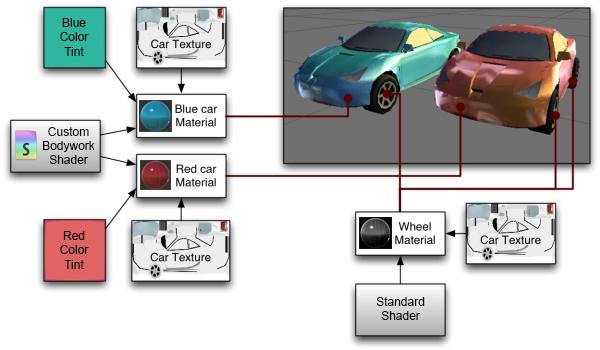
* **FX**：光照和玻璃特效。
* **GUI** 和 **UI**：用于显示用户界面图形。
* **移动端 (Mobile)**：适用于移动设备的简化高性能着色器。
* **大自然 (Nature)**：用于树木和地形。
* **粒子 (Particles)**：粒子系统特效。
* **天空盒 (Skybox)**：用于渲染所有几何体后面的背景环境
* **精灵 (Sprites)**：与 2D 精灵系统结合使用
* **卡通 (Toon)**：卡通风格的渲染。
* **无光照 (Unlit)**：用于完全绕过所有光照和阴影的渲染
* **旧版 (Legacy)**：已被标准着色器取代的大量旧着色器

## 着色器技术细节

一个着色器就是一个脚本，其中包含数学计算和算法，用于定义像素在模型表面上的显示效果。标准着色器执行复杂和逼真的光照计算。其他着色器可能使用更简单或不同的计算来显示不同的结果。在给定的着色器中有许多属性，可由使用该着色器的材质为它们赋值。这些属性可以是数字、颜色定义或纹理，在查看材质时会显示在检视面板中。然后，附加到游戏对象的渲染器组件将使用材质来渲染每个游戏对象的网格。

允许并且通常希望多种不同的材质引用相同的纹理。根据要求，这些材质还可使用相同或不同的着色器。

下面的示例展示了一种可能的设置组合，其中使用三种材质、两种着色器和一种纹理。



图中有一辆红色汽车和一辆蓝色汽车。两个模型对车身使用不同的材质，分别为“Red car material”（红色汽车材质）和“Blue car material”（蓝色汽车材质）。

这两种车身材质使用同一个自定义着色器“Carbody Shader”（车身着色器）。使用自定义着色器的原因是该着色器为专门汽车添加了额外的特性（例如金属性闪光渲染），或者可能具有自定义损坏遮罩功能。

每种车身材质均引用“Car Texture”（汽车纹理），这是包含车身所有细节但没有具体油漆颜色的纹理贴图。

车身着色器还接受一种色调颜色；红色和蓝色汽车设置为不同的颜色，使每辆汽车的外观不同，但两者使用同一种纹理。

车轮模型也使用单独的材质，但这次两辆汽车共用相同的车轮材质，因为每辆汽车的车轮没有差异。车轮材质使用标准着色器，并同样引用汽车纹理。

注意汽车纹理包含的车身和车轮细节：这是一个纹理图集，意味着纹理图像的不同部分被显式映射到模型的不同部分。

即使车身材质使用的纹理也包含车轮图像，车轮细节也不会出现在车身上，因为该纹理的那部分未映射到车身几何体。

同样，车轮材质使用相同的纹理，其中也有车身细节。车轮上不会出现车身细节，因为只有显示车轮细节的纹理部分才映射到了车轮几何体。

此映射由 3D 美术师在外部 3D 应用程序中完成，称为“UV 贴图”。

更具体地说，着色器定义了：

* 对象渲染方法。此方法包括代码和数学计算，其中可能包括光源的角度、视角以及任何其他相关计算。着色器还可根据最终用户的图形硬件指定不同的方法。
* 可在材质检视面板中自定义的参数，例如纹理贴图、颜色和数值。

材质定义了：

* 用于渲染此材质的着色器。
* 着色器参数（例如要使用的纹理贴图、颜色和数值）的具体值。

自定义着色器应由图形工程师构建。为创建自定义着色器，需要使用一种十分简单的语言：\_\_ShaderLab\_\_。然而，让着色器在各种显卡上正常工作是一项复杂的工作，需要相当全面的显卡工作原理知识。

Unity 中直接内置了许多着色器，还有一些着色器位于[标准资源 (Standard Assets)](file:///E:\UnityDocumentation\Manual\AssetTypes.html#Standard) 库中。