



# Unity

## 着色器和屏幕特效

[美] 杰米·迪恩 (Jamie Dean) 著 周翀 张薇 译

---

### Mastering Unity Shaders and Effects

---

- 以实际操作为主，按照从易到难的顺序，循序渐进讲解Unity着色器的编写和各种特效的创建
- 通过一个典型的科幻题材恐怖游戏，展示Unity 5中的所有主要着色器和效果，包含大量场景示例



机械工业出版社  
China Machine Press

# Unity

## 着色器和屏幕特效

[美] 杰米·迪恩 (Jamie Dean) 著 周翀 张薇 译

Mastering Unity Shaders and Effects



机械工业出版社  
China Machine Press

## 图书在版编目 (CIP) 数据

Unity 着色器和屏幕特效 / (美) 杰米·迪恩 (Jamie Dean) 著; 周翀, 张薇译. —北京: 机械工业出版社, 2017.6

(游戏开发与设计技术丛书)

书名原文: Mastering Unity Shaders and Effects

ISBN 978-7-111-57041-7

I. U… II. ①杰… ②周… ③张… III. 游戏程序—程序设计 IV. TP317.61

中国版本图书馆 CIP 数据核字 (2017) 第 118184 号

本书版权登记号: 图字: 01-2016-8653

Jamie Dean: *Mastering Unity Shaders and Effects* (ISBN: 9781783553679).

Copyright © 2016 Packt Publishing. First published in the English language under the title “Mastering Unity Shaders and Effects”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2017 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

## Unity 着色器和屏幕特效

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 和 静

责任校对: 殷 虹

印 刷: 三河市宏图印务有限公司

版 次: 2017 年 6 月第 1 版第 1 次印刷

开 本: 186mm×240mm 1/16

印 张: 12.25

书 号: ISBN 978-7-111-57041-7

定 价: 49.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

## *The Translator's Words* 译者序

Unity 是近两年中最流行且应用最广泛的游戏引擎之一。在国内，Unity 的使用已经成为游戏研发从业者的一项必备技能。

这种局面的形成有两个重要原因：其一在于 Unity 的高度完善性。它不但拥有一个游戏引擎所需的全部功能，并且其渲染效果与主流 3A 游戏产品相比毫不逊色。其二是因为 Unity 的开放性。所有人都可以从其官方网站免费获得一份近乎全功能的 Unity 开发工具包，以便用于学习或者功能评估。不仅如此，从帮助新手入门和提高方面来看，Unity 还拥有完善的使用文档和活跃的开发者社区。对于 Unity 使用者来说，翔实的学习和研发资料唾手可得。

本书是一本面向 Unity 初学者的“手册”式书籍。以实际操作为主线，按照从易到难的顺序，本书循序渐进地带领读者学习 Unity 着色器的编写和各种特效的创建。读者将在阅读的过程中边学边做，边动手边思考。既可以避免纸上谈兵，又可以方便读者将学习过程中偶尔产生的灵感及时付诸实践，检验其有效性。这是迅速开启游戏引擎应用大门的最佳途径。

游戏研发是一个持续高速发展的领域。一方面，玩家对新游戏的要求越来越高；另一方面，新理论、新算法也在不断地提出，照片级别的场景渲染和真实可信的物理模拟不断地出现在近些年的游戏新作当中。因此，每一个学习游戏研发技术的人都希望自己所学能够跟上时代的步伐、契合市场的需求。虽然本书面向初学者，但其内容不但涉及了一些先进的游戏图形渲染技术，比如用于表现皮肤的次表面散射效果，以及用于绘制毛发的各向异性高光效果等，而且也将近两年逐渐流行起来的移动设备平台涵盖其中。

总之，Unity 是一座桥梁，一端连接了游戏研发领域中跃跃欲试的入门者群体，另一端则通向真正的次世代游戏大作。而本书则向 Unity 初学者提供了一条以动手实践为基础的最佳入门途径，有助于读者在成为 Unity 应用专家的道路上快步前进。

## 前言 Preface

Unity 提供了一系列工具来帮助开发者创建和运行游戏项目。Unity 5 用全新的标准着色器替换了大部分旧版着色器，为我们带来了功能上的重大升级。新着色器通过简单设置便可模拟真实物体表面上的很多物理特性。

本书将探索这一系列新着色器为我们带来的全新选择，以及学习它们如何与光照和特效相互作用，产生高质量的游戏画面。

为了实现当今游戏画面中的真实感，人们走过了一条漫长的道路。在过去相当长的时间里，实时光照所带来的额外运算量为游戏研发制造了很多困扰。有时候，实现真实感的重担直接落在了纹理美术设计师的肩上。他们不得不将许多静态的表面效果直接烘焙到游戏角色和材质的纹理当中。

这种静态烘焙的方法有时在一些动态的游戏角色身上难以达到理想效果——特别是当一个角色移动到较阴暗或者较明亮的空间中时，静态烘焙难以实现良好的亮度等级变化，使得游戏画面显得粗糙而不真实。

计算机图形理论的进化，特别是显卡设备的升级换代，使很多基于实时光照的技术能够更加高效地运作。同时这也使人们对现代游戏报以更高的期望——我们希望在充满动态光照的场景中看到更逼真的物体表面。

在本书中，读者将通过一个典型的科幻题材恐怖游戏来学习这些新的着色器和光照理论。基于此，我们将能够探索各式各样的光照和物体表面效果。

## 本书主要内容

在本书的不同章节中，我们将学习 Unity 5 中的所有主要着色器和效果。在每一章中我们会研究一个不同的场景文件。

第 1 章介绍基本概念，并通过为恐怖科幻游戏中的场景设置材质属性，展示 Unity 标准

着色器的能力和局限性。

第2章继续设计飞船维修场景——通过创建自定义着色器优化游戏场景的外观。我们将从无到有创建自定义着色器，然后基于它，为头盔的透明外观和星球的大气创建更加复杂的效果。

第3章深入学习场景光照和自发光表面的关系，并在飞船驾驶舱场景中为一个星球模型实现全息影像效果。

第4章研究通过不同的技术在着色器中实现动画。在展示更复杂的基于自定义着色器的顶点动画之前，我们将演示如何通过滚动 UV 坐标和在 C# 中遍历纹理数组的方法来显示动画的全息影像。

第5章介绍在星球表面上所应用的各种透明效果，并且为多种材质指定现有的以及特殊创建的着色器。

第6章重点展示在 Unity 5 的标准着色器中，高光工作流和金属工作流的不同之处。

第7章展示在一个室内场景中为角色创建统一的皮肤和毛发着色器。本章中的第一个着色器为一些特殊的半透明材质（例如，宇航员的皮肤）实现次表面散射效果。紧接着我们将继续创建使用“基于物理渲染”方法的毛发和眼睛着色器。

第8章讲述如何创建实现烟雾、蒸汽和火焰的着色器。与前面章节类似，我们将不会满足于 Unity 的默认着色器，而是要为恐怖科幻游戏创建更高级别的电影效果。本章的场景将是飞船的走廊。

第9章研究如何在安卓和 iOS 等移动设备平台上测试自定义着色器。

## 阅读准备

完成本书中的各个章节需要读者具备基本的 3D 图形概念。此外还需要：

- 在计算机中安装 Unity 5 (macOS 或者 Windows 均可)。免费个人版即可满足本书所有章节中的实例程序。该版本可以从 Unity 官方网站下载。
- 虽然书中所含代码均会被清晰地讲解，但是具备 Unity 脚本和 C# 基础知识会对理解本书内容有益。

## 读者人群

本书面向具有一定 Unity 和 C# 基础知识并希望为下一代游戏开发着色器和效果的中等水平的游戏开发者。我们假设读者对游戏开发和游戏美术设计的基础背景知识有一定了解。

## 本书约定

在本书中，不同类型的内容使用多种不同的字体风格。这里列举一些示例并阐述它们的含义。

下面是一个代码段的示例：

```
Properties
{
    _Color ("Color", Color) = (0,0,0,0)
    _EdgeColor ("Edge Color", Color) = (0,1,0,1)
    _Width ("Width", float) = 0.1
}
```



表示警告和重要提醒。



表示提示和技巧。

## 下载示例代码

读者可以从 <http://www.packtpub.com> 通过个人账号下载本书的示例代码文件。如果本书是通过其他途径购买的，可以访问 <http://www.packtpub.com/support> 并且注册账户以便通过电子邮件接收示例代码文件。

读者也可以访问华章图书官网 [www.hzbook.com](http://www.hzbook.com)，通过注册并登录个人账号，下载本书的源代码。

## 下载书中彩图

我们也将本书所使用的彩色屏幕截图和图制作成了一个 PDF 文件供读者参考。彩图可以帮助读者更直观地理解某一项技术或方法为程序的输出带来的改变。该 PDF 文件可以从[这里](http://www.packtpub.com/sites/default/files/downloads/MasteringUnityShadersAndEffects_ColorImages.pdf)下载：[http://www.packtpub.com/sites/default/files/downloads/MasteringUnityShadersAndEffects\\_ColorImages.pdf](http://www.packtpub.com/sites/default/files/downloads/MasteringUnityShadersAndEffects_ColorImages.pdf)。

译者序

前 言

|                           |    |
|---------------------------|----|
| 第1章 掌握标准着色器 .....         | 1  |
| 1.1 创建程序项目 .....          | 2  |
| 1.2 导入项目文件 .....          | 3  |
| 1.3 加载飞船维修场景并在场景中漫游 ..... | 4  |
| 1.3.1 为宇航员创建材质 .....      | 5  |
| 1.3.2 为宇航员的装备创建材质 .....   | 9  |
| 1.3.3 为飞船创建材质 .....       | 16 |
| 1.3.4 为星球创建材质 .....       | 20 |
| 1.3.5 设置天空盒 .....         | 23 |
| 1.3.6 调节场景光照并添加效果 .....   | 25 |
| 1.4 总结 .....              | 28 |
| 第2章 创建自定义着色器 .....        | 29 |
| 2.1 打开程序项目 .....          | 29 |
| 2.1.1 打开场景 .....          | 30 |
| 2.1.2 创建第一个自定义着色器 .....   | 30 |
| 2.2 进阶的透明效果 .....         | 39 |
| 2.2.1 创建自定义透明着色器 .....    | 39 |
| 2.2.2 编辑透明着色器 .....       | 40 |



|                                   |           |
|-----------------------------------|-----------|
| 2.3 进阶的星球大气效果 .....               | 45        |
| 2.3.1 创建自定义大气着色器 .....            | 45        |
| 2.3.2 应用大气着色器 .....               | 46        |
| 2.3.3 编辑大气着色器 .....               | 46        |
| 2.4 总结 .....                      | 49        |
| <b>第3章 使用光照和自发光表面 .....</b>       | <b>50</b> |
| 3.1 查看场景的光照设置 .....               | 50        |
| 3.1.1 为材质添加自发光属性 .....            | 53        |
| 3.1.2 为画面添加 Bloom 特效 .....        | 55        |
| 3.2 为星球表面扫描器创建线框自发光材质 .....       | 57        |
| 3.2.1 实际观察线框自发光着色器的表现 .....       | 61        |
| 3.2.2 为线框着色器添加第二遍渲染 .....         | 62        |
| 3.3 总结 .....                      | 66        |
| <b>第4章 通过代码和着色器实现物体表面动画 .....</b> | <b>67</b> |
| 4.1 打开场景 .....                    | 67        |
| 4.2 创建警示灯的闪烁效果 .....              | 68        |
| 4.3 创建显示屏的动态亮度 .....              | 71        |
| 4.4 创建星球全息影像的动画效果 .....           | 78        |
| 4.5 总结 .....                      | 86        |
| <b>第5章 研究透明表面和效果 .....</b>        | <b>87</b> |
| 5.1 打开场景 .....                    | 87        |
| 5.1.1 创建尘埃云团材质 .....              | 88        |
| 5.1.2 添加雾效果 .....                 | 90        |
| 5.1.3 让尘埃云团动起来 .....              | 92        |
| 5.2 创建更好的透明玻璃材质 .....             | 97        |
| 5.3 创建旋风特效 .....                  | 100       |
| 5.4 总结 .....                      | 103       |

|                                 |     |
|---------------------------------|-----|
| <b>第 6 章 使用高光和金属表面</b>          | 104 |
| 6.1 打开场景                        | 105 |
| 6.1.1 运行时修改货箱的第二材质              | 105 |
| 6.1.2 添加第二反光纹理贴图                | 106 |
| 6.1.3 为货箱创建自定义贴花着色器             | 108 |
| 6.1.4 运行时切换贴花纹理                 | 113 |
| 6.2 找到并修改渲染器光照模型                | 115 |
| 6.2.1 创建自定义光照模型                 | 118 |
| 6.2.2 为自定义光照模型添加高光              | 121 |
| 6.3 总结                          | 124 |
| <b>第 7 章 有机表面着色器</b>            | 125 |
| 7.1 打开场景                        | 125 |
| 7.2 理解皮肤表面的复杂性                  | 126 |
| 7.3 创建皮肤着色器                     | 126 |
| 7.3.1 为皮肤着色器添加贴图                | 129 |
| 7.3.2 为皮肤着色器添加自定义光照模型           | 132 |
| 7.4 为眼球创建材质                     | 136 |
| 7.5 为头发创建材质                     | 140 |
| 7.6 总结                          | 145 |
| <b>第 8 章 自定义粒子着色器——烟雾、蒸汽和流体</b> | 147 |
| 8.1 打开场景                        | 148 |
| 8.2 添加粒子系统                      | 148 |
| 8.2.1 调整粒子系统参数                  | 149 |
| 8.2.2 为粒子系统创建材质                 | 152 |
| 8.3 创建自定义粒子着色器                  | 153 |
| 8.3.1 为自定义粒子着色器添加颜色支持           | 156 |
| 8.3.2 添加蒸汽效果                    | 156 |
| 8.3.3 创建黏液滴落效果                  | 162 |
| 8.4 总结                          | 169 |

|                         |     |
|-------------------------|-----|
| 第9章 为移动设备优化着色器 .....    | 170 |
| 9.1 打开场景 .....          | 170 |
| 9.2 为移动设备创建简化的着色器 ..... | 175 |
| 9.3 为移动设备创建高级着色器 .....  | 181 |
| 9.4 总结 .....            | 183 |



## 掌握标准着色器

在本章中，我们将学习 Unity 的标准着色器，并为第一个场景创建一系列材质。

Unity 5 的这套全新的标准着色器，采用了时下流行的基于物理渲染（Physically-Based Rendering, PBR）的方法。

在标准着色器出现之前，开发者必须为了创建透明、镜面反射和金属光泽材质而生成特定的着色器。这会包含众多不同表面的游戏带来问题——每一个着色器都是一段需要运行的小程序，而每添加一段这样的小程序都有可能对整体运行效率带来一定的负面影响。

标准着色器提供了一种统一方法——它们既可以实现镜面反射，也可以实现透明、金属光泽，以及所有其他效果。未使用的贴图通道会在运行时从计算中剔除，从而缩短计算耗时。

标准着色器与 Unity 的万能光照方案 Enlighten 结合在一起，增强了游戏场景中物体表面的真实感，以及对环境变化的响应。

我们会利用各种机会将金属工作流和高光工作流同时应用到游戏场景中，以便读者对它们的各方面特性进行观察和对比。

本章涉及的内容如下。

- 讨论如何配置程序项目以及导入自定义资源包（Asset Package）。

- 为各种物体表面创建复杂的材质。
- 使用图形界面上的滑动条和纹理贴图来设定高光和金属光泽参数。
- 将材质组应用到导入的模型上。
- 使物体表面透明。
- 将多个材质叠加。
- 讨论金属工作流和高光工作流的关键区别。
- 讨论如何创建天空盒 (skybox) 资源。
- 讨论如何调整场景的基本亮度。

通过学习本书中的科幻恐怖游戏的设计, 读者将深刻理解着色器和效果。

在游戏中, 一个孤独的宇航员为了与一个神秘失联的科研小队取得联系, 驾驶飞船飞向一个遥远的星球 “Ridley VI”。这个游戏内容将为我们提供更多的机会去探索如何在游戏中应用着色器和效果。

开始工作吧!

## 1.1 创建程序项目

Unity 用程序项目将场景、模型、材质、着色器、纹理以及其他资源整合在一起。

在这一阶段, 我们先创建程序项目, 然后导入本书示例程序所需的各种项目文件。

1. 启动 Unity。

2. 当提示打开现有程序项目或创建新程序项目时, 选择创建新项目 (Create New Project)。

3. 给程序项目起一个恰当的名称, 例如 “Shaders And Effect” (着色器和效果), 并且在硬盘中为其指定一个方便访问的位置。

4. 保持默认的 3D 选项。

5. 选中 Effects (效果) 资源包<sup>⊖</sup>。

6. 单击创建项目 (Create project) 按钮。

Unity 将花一些时间向程序项目内复制默认的文件。当新程序项目创建完成时, 在

---

⊖ 请单击添加资源包 (Add Asset Package) 按钮, 在弹出的资源包选择窗口里, 单击左下部的选择所有 (select all) 按钮并单击完成 (Done) 按钮。这一操作将 Unity 自带的默认资源包添加到程序项目里。后续章节中将有部分设置需要使用默认效果资源。——译者注

场景 (Scene) 视图将出现一个空场景空间。

接下来向程序项目导入本书所需的项目文件。



### 下载示例代码

在本书前言中有下载示例代码的详细步骤，请参考。

示例代码也可以在 GitHub 上找到，请访问下面链接：<https://github.com/PacktPublishing/Mastering-Unity-Shaders-And-Effects>。同时读者在 <https://github.com/PacktPublishing/> 上也可以找到与其他书籍或视频所对应的示例代码。

## 1.2 导入项目文件

本书示例代码所需的项目文件已经被制作作为 Unity 资源包，内有各章节内容所需的模型和纹理素材。请通过如下步骤导入项目文件。

1. 在 Unity 的主界面菜单栏上单击资源 (Assets) 菜单。

2. 在弹出的菜单中单击导入包 (Import Package) | 自定义包 (Custom Package...) 命令。

3. 在弹出的文件对话框中，找到用于保存 PACKT\_SAE 资源包的文件夹。

在该文件夹内，有一个名为 PACKT\_SAE.unitypackage 的文件。

4. 选中并打开该文件。

Unity 会花些时间解压缩该资源包。资源包的内容将以列表的形式显示在导入 Unity 包 (Import Unity Package) 对话框上。

默认情况下全部资源被选中。

5. 单击导入 (Import) 按钮。

当导入过程结束，在项目 (Project) 面板上应当显示出多个项目文件夹。接下来加载第一个场景。

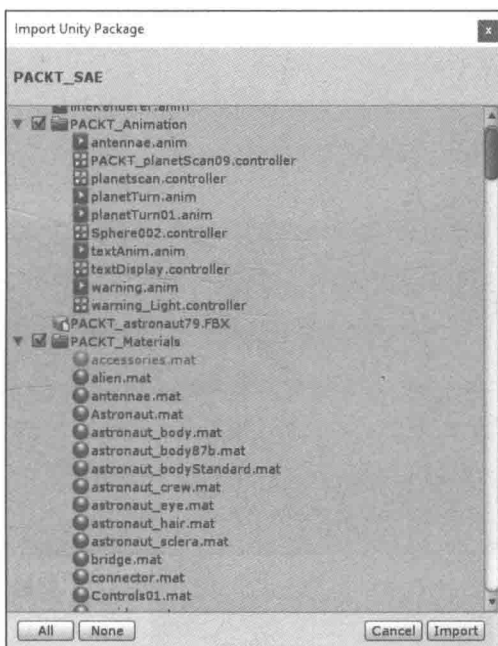


图 1-1 导入 Unity 包

## 1.3 加载飞船维修场景并在场景中漫游

第一个游戏场景位于星球“Ridley VI”附近。宇航员停下飞船进行一些基础维修，为登陆星球做好准备。

在这里我们先简单地基于 Unity 的标准着色器来创建不同的材质，从而使得不同的物体表面拥有相近的显示质量。

1. 在项目（Project）面板上找到 PACKT\_Scenes 文件夹，并单击文件夹的图标，以便使它的内容显示在资源（Assets）面板上。

2. 在资源（Assets）面板上找到名为 Chapter1\_Start 的资源。

3. 双击该资源，以便将它包含的场景加载到项目中。

在场景（Scene）视图中会看到如下的场景画面。



图 1-2 初始状态的场景画面

初始状态的场景看起来有些简陋，其中只有默认外观的飞船、宇航员，以及远处的星球和它的卫星。

场景已经由 Unity 的默认光源照亮，因此当我们对标准着色器进行设置时，可以同时观察和对比不同材质在光源下的外观。

接下来将从宇航员的材质开始设置。

### 1.3.1 为宇航员创建材质

宇航员的模型由许多不同的表面构成，比如，她的脸、玻璃头盔以及由金属和布料制成的太空服等。Unity 的标准着色器可以表现所有这些不同之处，开发者只需通过图形界面设定一些参数及纹理即可定义出具有金属光泽或者高光效果的表面。

下面从创建材质资源开始做起。

1. 在项目 (Project) 面板上找到 PACKT\_Scenes 文件夹，并单击文件夹的图标，以便使它的内容显示在资源 (Assets) 面板上。

2. 右键单击资源 (Assets) 面板上的任一空白位置，在弹出的菜单上单击创建 (Create)| 材质 (Material) 命令。

新材质资源将会在 PACKT\_Materials 文件夹下面产生。

3. 将新产生的资源重命名为“Astronaut”。

Unity 5 为新材质默认指定标准着色器，并使用金属光泽工作流。这是一种能够适用于多种不同物体表面的万能着色器。

4. 在资源 (Assets) 面板上单击“Astronaut”材质，将它的属性显示在检视 (Inspector) 面板上<sup>①</sup>。

在材质初始的空白状态下，“Astronaut”材质具有白色的单色反光以及一定量的平滑度，使得该材质的外观略微显得闪亮。

5. 单击反光 (Albedo) 右侧的白色方框并选择其他颜色。

在为一种材质设定更多纹理素材之前，临时为其设定一种比较显眼的单色反光颜色是一个比较实用的技巧。因为这样便于观察该种材质正在应用于场景中的哪些物体表面。

具体的做法是，为材质设定某一种单色反光颜色后 (如正红色)，在 Unity 界面的顶端中部单击场景 (Scene)

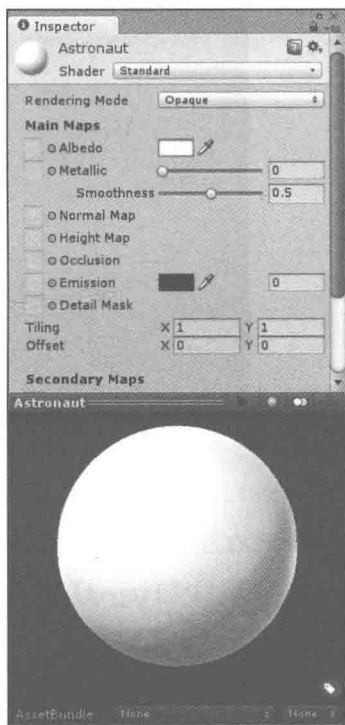


图 1-3 材质的默认设置

① 如果检视 (Inspector) 面板目前不可见，请在 Unity 的主界面菜单栏上单击窗口 (Window) 菜单，并在弹出的菜单里单击检视 (Inspector) 命令，以便使检视 (Inspector) 面板可见。——译者注



卡片将显示切换到场景视图。通过在场景中观察并寻找特定颜色（如寻找正红色），便可以方便地知道我们的修改正在影响场景的哪些部分。

6. 用摄像机控制功能将镜头向宇航员拉近一些：“Alt+ 鼠标左键”为旋转摄像机，“Alt+ 鼠标中键”为平移摄像机，“Alt+ 鼠标右键”为拉近或拉远摄像机。

7. 从资源（Assets）面板上将“Astronaut”材质拖放到场景视图中的宇航员模型上。模型的外观将产生相应变化，这表明“Astronaut”材质已经应用到模型上。

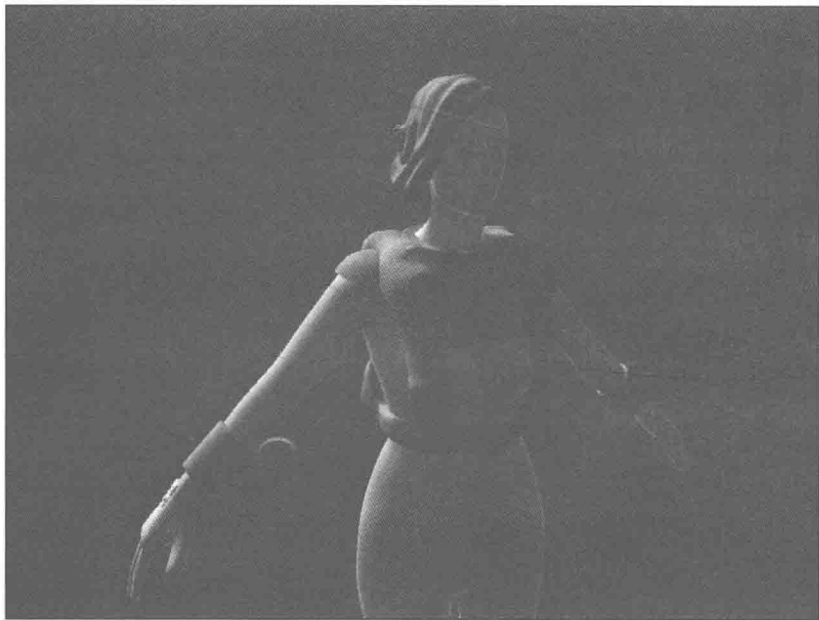


图 1-4 “Astronaut”材质应用到场景视图中

从现在起对该材质所做的修改将会直接在场景视图中体现出来。

8. 在项目（Project）面板上单击 PACKT\_Textures 文件夹左侧的灰色小三角形，使子文件夹显示出来。

9. 找到并单击“astronaut\_suit”子文件夹，将它的内容显示在资源（Assets）面板上。

10. 在资源（Assets）面板上找到“astronaut\_albedo”纹理。

11. 将该纹理拖放到检视（Inspector）面板上的反光（Albedo）槽上<sup>①</sup>。

---

① “槽”翻译自原书用语“Slot”。这里指界面上“Albedo”文字左侧的白色小方框。后面提及的所有其他参数的“槽”均为类似位置上的小方框。——译者注