.NET 开发经典名著

C#实践入门

快捷学习 C#编程和 Unity 游戏开发

(第 4 版)

[美] 哈里森 • 费隆(Harrison Ferrone) 著
冯俊儒 许瑞灌 梅 晶 译

清华大学出版社

北 京

北京市版权局著作权合同登记号　图字：01-2020-6981

图书在版编目(CIP)数据

C#实践入门：快捷学习 C#编程和 Unity 游戏开发：第 4 版/(美)哈里森 • 费隆(Harrison Ferrone)著；冯俊儒，许瑞灌，梅晶 译. —北京：清华大学出版社，2021.5
(.NET 开发经典名著)
书名原文：Learning C# by Developing Games with Unity 2019: Code in C# and build 3D games with Unity, 4th Edition
ISBN 978-7-302-57585-6

Ⅰ.①C… Ⅱ.①哈… ②冯… ③许… ④梅… Ⅲ.①C 语言－程序设计 Ⅳ.①TP312.8

中国版本图书馆 CIP 数据核字(2021)第 031282 号

责任编辑：王　军
装帧设计：孔祥峰
责任校对：成凤进
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：大厂回族自治县彩虹印刷有限公司
经　　销：全国新华书店
开　　本：170mm×240mm　　印　　张：17　　字　　数：352 千字
版　　次：2021 年 5 月第 1 版　　印　　次：2021 年 5 月第 1 次印刷
定　　价：69.80 元

产品编号：088760-01

译 者 序

C#是一种简洁精炼、类型安全的面向对象编程语言，因易用、高效、稳定而得到广泛使用。通常，了解 C、C++、Java 中任何一种编程语言的开发人员都能在短时间内上手 C#。C#还可作为脚本语言用于 Unity3D 开发。

Unity3D 是目前市面上最热门的跨平台游戏引擎。由于十分易用、便利，Unity3D 极大提高了游戏开发效率。经过多年的发展，Unity3D 有了良好的生态圈，AssetStore 插件丰富，网络上的技术分享也有很多。

本书将带领你从零开始学习 C#语言，同时在熟悉 Unity3D 基本操作的前提下，开发一个有趣的游戏。本书内容深入浅出、循序渐进，不但授人以鱼，还授人以渔，各个核心要点都有代码示例。对于入门 C#或 Unity3D 游戏开发来说，本书简单易学，可操作性强，是最佳阅读选择。

本书共分 12 章，其中第 1~4 章由许瑞灌翻译，第 5~8 章由冯俊儒翻译，第 9~12 章由梅晶翻译，全书由冯俊儒统稿。

非常感谢清华大学出版社为我们提供了这次宝贵的翻译机会，在此要感谢本书的编辑，他们为本书的翻译和出版投入了巨大的热情并付出了很多心血。没有他们的帮助和鼓励，本书不可能顺利付梓。

对于这本 C#实用之作，译者在翻译过程中力求忠于原文，将作者的本意呈现给读者，但是由于水平有限，书中肯定存在不当或遗漏之处，敬请各位读者不吝赐教，我们将不胜感激。

作者简介

Harrison Ferrone 是土生土长的芝加哥人，他经常为 LinkedIn 和 Pluralsight 创建教学内容，此外还是 Ray Wenderlich 网站的技术编辑。

在科罗拉多大学博尔德分校和芝加哥哥伦比亚学院求学时，Harrison Ferrone 撰写过许多有趣的论文。作为 iOS 开发人员，在为一家初创公司和另一家名列《财富》500 强的公司工作数年后，Harrison Ferrone 选择从事教育事业。期间，他购买了许多书籍，养了几只猫，还思索了为何论文集《神经漫游者》不会出现在更多的教学大纲里。

审校者简介

Luiz Henrique Bueno 是一位拥有 Certified ScrumMaster (CSM) 和 Unity 认证的开发人员，他在软件开发领域有超过 29 年的经验。2002 年，当 Visual Studio 发布时，Luiz Henrique Bueno 就撰写了 *Web Applications with Visual Studio.NET, ASP.NET, and C#*一书。他曾担任智能家居杂志 *Casa Conectada* 的主编长达 6 年时间，期间，他使用 Crestron 和 Control4 开发了许多项目。此外，Luiz Henrique Bueno 还是 *Unity 2017 Game Optimization，Second Edition* 一书的审稿人。他的座右铭是“不要为质量检查编写代码，而要为创建产品编写代码”。

前　言

Unity 已成为全球最受欢迎的游戏引擎之一，可满足业余爱好者、专业 3A 工作室和电影工作室人员的需求。虽然主要被视为 3D 工具，但 Unity 还包含 2D 游戏、虚拟现实乃至产品后期处理及跨平台发布等诸多专业功能。

尽管 Unity 提供的即拖即用接口与内置功能已广受开发者欢迎，但真正使得 Unity 得到进一步发展的原因在于 Unity 支持使用 C#来编写行为与游戏机制。编写 C#代码对于有其他编程语言经验的程序员来说虽然不会是什么太大的障碍，但是会让没有任何编程经验的人望而生畏。这正是本书将要发挥的作用，因为本书将会带领你从零开始学习 C#语言与编程的基本要素，同时在 Unity 中开发一些有趣的好玩游戏。

目标读者

本书主要是为没有任何编程经验或不了解 C#语言的读者编写的。如果有相关知识抑或是一位经验丰富的程序员，但想要尝试游戏开发，那么本书也适合阅读和参考。

本书内容

第 1 章“了解环境”将介绍 Unity 的安装过程、编辑器的主要功能以及如何查找并学习与 C#和 Unity 主题有关的文档。该章同时还会介绍如何在 Unity 中创建 C#脚本并使用 Visual Studio 编写代码。

第 2 章“编程的构成要素”将列出编程中最基本的概念，尝试将变量、方法、类型与日常生活中的情境联系起来。你还将学习简单的调试技巧、合适的格式与注释并了解

Unity 是如何将 C#脚本转换成组件的。

第 3 章“深入研究变量、类型和方法”将深入介绍变量相关的知识，包括 C#的数据类型、命名规范、访问修饰符以及其他一切编程基础知识。之后你将学习如何编写方法，使用参数与返回类型。该章最后将概述属于 MonoBehaviour 类的标准 Unity 方法。

第 4 章“控制流程与集合类型”将介绍用来进行决策的通用方式，包含 if-else 和 switch 语句。然后讨论数组、列表和字典，并利用迭代语句遍历以上集合类型。该章最后将讨论条件循环语句和一种特殊的 C#数据类型——枚举。

第 5 章“使用类、结构体和 OOP”将详细介绍如何在代码中构造并实例化类与结构体，包括创建构造函数、添加类及结构体的变量和方法的基本步骤以及有关子类和继承的知识。该章将以介绍面向对象编程以及如何将其应用于 C#结束。

第 6 章“亲自上手使用 Unity”将脱离 C#语法，开始对游戏设计、关卡构建和 Unity 特色工具进行学习。该章将从一份基础的游戏设计文档开始，然后摆放好关卡几何体并添加光照和简单的粒子系统。

第 7 章“移动、相机控制与碰撞”将解释移动玩家对象和设置第三人称相机的不同方式，还会讨论如何利用 Unity 的物理系统来达到更真实的运动效果，并使用碰撞体组件在场景中进行交互。

第 8 章“编写游戏机制”将介绍游戏机制的概念以及如何高效地加以实现。该章从简单的跳跃行为开始，接着创建射击机制，并添加用来处理物品收集机制的代码。

第 9 章“人工智能基础和敌人行为”将简要介绍游戏中的人工智能以及将要应用到 Hero Born 中的概念，内容涵盖在 Unity 中使用关卡几何体和导航网格进行寻路、智能代理以及实现敌军的自动移动。

第 10 章“回顾类型、方法和类”将更深入地讨论数据类型、函数特性以及可应用于更复杂类中的其他行为。该章将使你对 C#语言的广度与深度有更深刻的理解。

第 11 章“探索泛型、委托等”将详细讲解 C#语言的更多中级特性以及如何将它们应用于实际场景中。你将学习泛型编程并了解诸如委托、事件和异常处理的概念。该章将以对一些通用设计模式的讨论而结束，使你为未来的学习做好准备。

第 12 章“旅行继续”将回顾你在本书中学习的主题，并提供一些可用于后续学习 C#和 Unity 的资源，包括网络材料、认证信息以及许多受欢迎的视频教程。

附录 A“完整的游戏代码文件”中包含组成 Hero Born 的必需的 C#脚本。

附录 B“辅助类”中包含一些来自第 10 章和第 11 章的中级代码，这些代码不会直接影响 Hero Born 的游戏机制。

附录 C“小测验答案”提供本书每一章末尾出现的小测验的答案。

如何有效利用本书

在即将到来的学习 C#和 Unity 之旅中，你唯一需要做的就是保持好奇心。为了巩固所学的知识，你需要完成所有的“实践”与“试验”以及各章末尾的小测验。最后，在继续前进之前，回顾你从每一章学到的知识是绝佳的主意。勿在浮沙上筑高台！

下载示例代码和书中截图

本书的示例代码及截图可通过扫描封底的二维码获得。

目　录

第 I 部分

编程基础与 C#

本书第 I 部分将从头开始介绍 Unity 开发环境、编程基础和 C#语言，你将了解如何使用 Unity 并掌握创建简单游戏所需的基础知识。

内容包括：

- 第 1 章“了解环境”。
- 第 2 章“编程的构成要素”。
- 第 3 章“深入研究变量、类型和方法”。
- 第 4 章“流程控制与集合类型”。
- 第 5 章“使用类、结构体和 OOP”。

第 1 章

了解环境

大部分人认为，计算机程序员不合群，是独行侠或是在算法思想方面有非凡心智和天赋的极客，他们几乎没有社交智商且骨子里有种古怪的不守规矩的天性。然而事实并非如此，实际上，有些观点则认为学习编程从根本上改变了人们看待世界的方式。值得庆幸的是，人们天生的好奇心可以快速适应这种新的思维方式，甚至可能会慢慢喜欢上。

在日常生活中，我们已经在使用可以转换为编程的分析技能——只是缺少将这些技能映射到代码中的正确语言和语法。你知道自己的年龄，对吧？年龄就是变量。过马路时，你会向左右看，然后像其他人一样再次向左看，此时就是在评估不同的条件或控制流程。当看到一瓶汽水时，你会本能地识别出一些属性，比如形状、重量和所装之物，这瓶汽水就是类对象。

本章将通过以下主题开启你的 C#编程和 Unity 之旅：

- 一些基本前提
- 从 Unity 2019 开始
- 在 Unity 中使用 C#
- 使用 Visual Studio 编辑器
- 访问文档和资源

1.1 一些基本前提

有时候，说出某个事物不是什么相比说出是什么更容易。本书的主要目标不是学习 Unity 或所有游戏开发的来龙去脉。本书将简单介绍这些主题，并在第 6 章进行较详细的讨论，目的是提供一种有趣且易懂的方式，使读者能够从头开始学习 C#编程语言。

因为本书针对编程新手，所以如果没有使用过 C#或 Unity，那么来对地方了！如果对 Unity 编辑器有一些经验，但没有编程经验，那么仍然来对地方了！即使已经涉足 C#和 Unity，但仍想探索一些中级或高级主题，本书也可以提供所需的内容。

注意：

对于具备其他编程语言经验的程序员，可以跳过入门理论，直接阅读自己感兴趣的章节。

1.2 从 Unity 2019 开始

访问 https://store.unity.com/网站，你会看到几个选项，如图 1-1 所示。不要不知所措——你可以通过选择右边的 Personal 选项免费获得 Unity。其他付费选项提供了更高级的功能和服务，后面可自行查看这些选项。

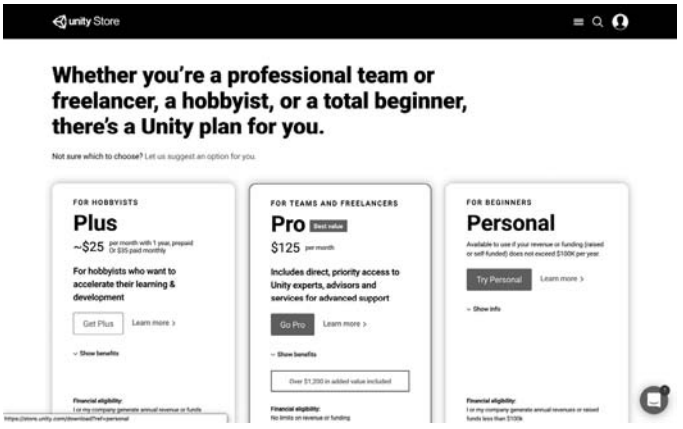

图 1-1 访问 https://store.unity.com/网站

选择个人版(有别于专业版)后，将停留在下载界面。请接受条款，然后单击 Download Installer for Mac OS X 按钮，如图 1-2 所示。

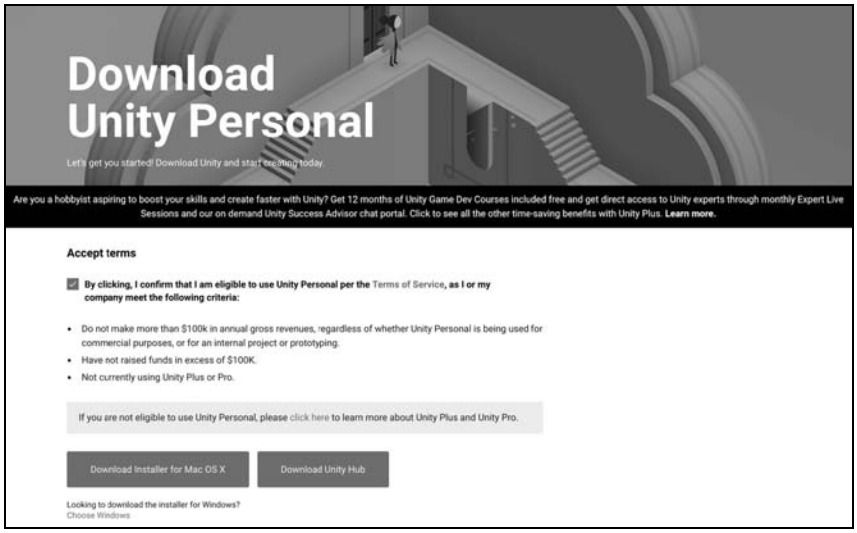

图 1-2　下载 Unity 个人版

如果使用的是 Windows 操作系统，请单击 Download Installer for Mac OS X 按钮下方的 Choose Windows 链接，然后接受条款就可以开始下载 Unity 个人版了！

注意：

通过图 1-2 所示界面还可以下载 Unity Hub，通过 Unity Hub 可以下载和管理不同版本的 Unity。

下载完之后，双击安装文件并按照安装说明进行操作。当获得许可证时(需要激活的有效 Unity 证书)，即可继续并启动 Unity！

注意：

本书使用 Unity 2018.3.8f1 创建示例并截图，这些示例已在 Unity 2019.2.0a7 上运行通过。如果你使用的是较新版本，那么编辑器中的内容则可能会略有不同，但后续操作不会有问题。

1.2.1　创建新项目

打开 Unity 后，你首先看到的将是仪表板(Unity 启动界面)。如果有 Unity 账户，请登录。如果没有，则需要创建 Unity 账户或单击界面底部的 Skip 按钮。

如图 1-3 所示，选择右上方的 New 选项卡并设置如下字段以创建新项目。

- Project name：这里设置为 Hero Born。
- Location：输入项目的存储路径。

- Template：确保设置为 3D。

设置完项目之后，单击 Create project 按钮。

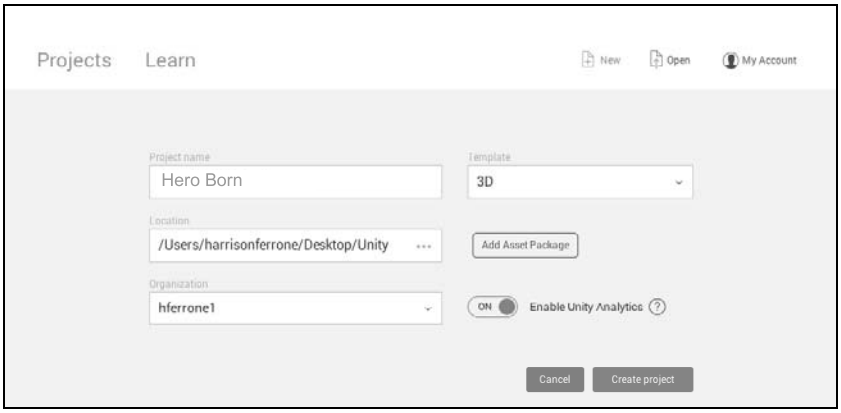

图 1-3　在 New 选项卡中设置项目

1.2.2　浏览编辑器

在完成对新项目的初始化之后，你就会看到壮观的 Unity 编辑器！图 1-4 已对重要的选项卡(或面板)做了标记。

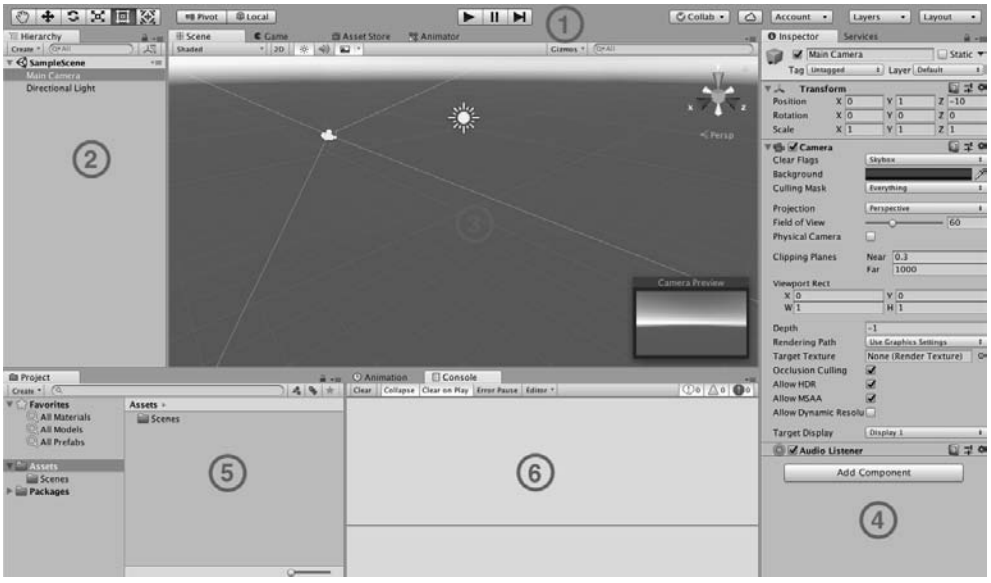

图 1-4　Unity 编辑器

- 工具栏位于 Unity 编辑器的顶部。其中，最左边的一组按钮可以操纵对象，中间的一组按钮可以播放或暂停游戏，最右边的一组按钮则涉及 Unity 服务、Layer Masks(层级菜单)以及本书从不使用的 Layout 菜单。
- Hierarchy 面板展示了当前场景中的每个游戏对象。项目最初只有默认的相机和平行光，当创建了原型环境时，Hierarchy 窗口开始被填充。
- 游戏中最直观的是 Game 视图和 Scene 视图。Scene 视图可以当作舞台，设计者在 Scene 视图中移动并布置 2D 和 3D 对象。当单击 Play 按钮时，Game 视图将接管并渲染 Sence 视图以及任何已编程的交互。
- Inspector 面板提供了查看和编辑对象属性的一站式服务。在图 1-4 中，可以看到 Main Camera 显示的几个部分(Unity 称之为组件)都可以通过 Inspector 面板来访问。
- Project 面板显示了当前项目中的所有资产(Asset)，它们可被看成项目的文件夹和文件。
- Console 面板用于显示脚本的输出结果。从现在开始，我们谈到的所有控制台输出或调试输出都会显示在 Console 面板中。

注意：
通过访问 https://docs.unity3d.com/Manual/UsingTheEditor.html，就可以在 Unity 文档中找到有关每个面板的功能的更深入介绍。

如果不熟悉 Unity，那么需要处理很多事情。但请放心，任何指示总是能够指导你执行必要的步骤。问题不会悬而不决，让你不知道该怎么处理。把上述事情放在一边，就可以创建一些实际的 C#脚本了。

1.3　在 Unity 中使用 C#

展望未来，把 Unity 和 C#看作共生的实体是很重要的。Unity 是创建脚本并最终运行脚本的引擎，但实际的编程工作是在另一个名为 Visual Studio 的编辑器中进行的。不用担心——稍后就会讲到。

1.3.1　使用 C#脚本

即使还未介绍任何基础的编程概念，你也仍然需要知道如何在 Unity 中创建 C#脚本。

在编辑器中创建 C#脚本的方法有如下几种：

- 在菜单栏中选择 Assets | Create | C# Script。
- 在 Project 选项卡中选择 Create | C# Script。
- 右击 Project 选项卡，从弹出的菜单中选择 Create | C# Script。

注意：
当需要创建 C#脚本时，可使用自己喜欢的任何方法。

提示：
除 C#脚本外的所有资源和对象都可以使用前面介绍的方法在编辑器中创建出来。本书不会在每次创建新的资源时都提到这些选项，所以请记住它们。

实践——创建 C#脚本

为了便于组织，我们会把各种资源和脚本存储在明确指定的文件夹中。这不仅是与 Unity 相关的任务，也是我们经常要做的事情。

(1) 在 Project 选项卡中选择 Create | Folder，创建一个文件夹并命名为 Scripts。

(2) 双击进入 Scripts 文件夹，创建一个新的 C#脚本。这个脚本默认被命名为 NewBehaviourScript，这里需要改为 LearningCurve。

刚刚发生了什么

我们刚创建了一个名为 Scripts 的文件夹，如图 1-5 所示。在 Scripts 文件夹中，我们又创建了一个名为 LearningCurve 的 C#脚本，并将其保存为 Hero Born 项目的部分资源。

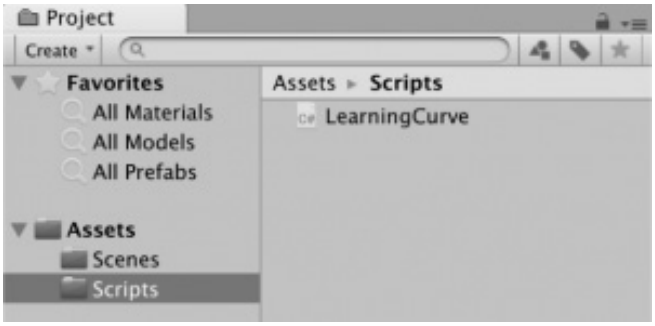

图 1-5　创建的 Scripts 文件夹

1.3.2　Visual Studio 编辑器

虽然 Unity 可以用来创建和保存 C#脚本，但我们需要使用 Visual Studio 来编辑它们。

Visual Studio 已预安装在 Unity 中，在编辑器中双击任何 C#脚本，就会自动打开 Visual Studio。

1. 实践——打开 C#脚本

Unity 在首次打开 C#脚本时，会把它们同步到 Visual Studio 中。最简单的同步方式就是在 Project 选项卡中选择脚本，比如 LearningCurve 脚本，双击后就可以看到脚本已在 Visual Studio 中打开，如图 1-6 所示。

图 1-6　在 Visual Studio 中打开 LearningCurve 脚本

在图 1-6 中，界面的左侧展示了项目的文件夹结构，可以像访问普通文件夹一样进行访问；右侧是实际的代码编辑器。当然，Visual Studio 还有更多的功能，但掌握这些是我们保持继续前进所必需的。

2. 当心命名不匹配

新手程序员容易遭遇的常见陷阱就是文件的命名，具体而言，就是命名不匹配。以图 1-6 中的第 5 行代码为例：

```
public class LearningCurve : MonoBehaviour
```

LearningCurve 类名与 LearningCurve.cs 文件名相同，这是必需的，也是要求之一。即便还不知道类是什么，也没关系，只需要记住脚本文件名和类名必须相同即可。

当在 Unity 的 Project 选项卡中创建 C#脚本时，脚本文件名处于编辑模式，可以进行重命名。选择此时对脚本文件进行重命名是个好习惯。如果选择之后再命名，那么

脚本文件名和类名有可能不匹配。因为文件名虽然改了，但脚本中没有更新。例如，图 1-6 中的第 5 行代码很可能是下面这个样子：

```
public class NewBehaviourScript : MonoBehaviour
```

如果不小心这样做了，情况还不是很糟糕。只需要打开 Visual Studio，将 NewBehaviourScript 改为 C#脚本的名称即可。

1.3.3 同步 C#文件

作为共生关系的一部分，Unity 和 Visual Studio 保持相互联系并同步内容，这意味着如果在其中一个应用程序中新增、删除或修改脚本文件，那么另一个应用程序也会自动同步发生的变化。

修复中断同步

那么，当同步似乎无法正常工作时，该怎么办呢？如果遇到这种情况，请深呼吸并执行以下操作：右击 Unity 中的 Project 面板，从弹出的菜单中选择 Sync Visual Studio Project。

1.4 文档

我们要谈的最后一个主题是文档。这很重要，当我们接触新的编程语言或开发环境时，早点养成好习惯十分重要。

1.4.1 访问 Unity 文档

一旦开始认真编写脚本，就可能经常使用 Unity 文档。因此，了解如何访问 Unity 文档对你是很有帮助的。参考手册会对组件或主题进行概述，特定的编程示例则可在参考脚本中找到。

1. 实践——打开参考手册

场景中的每个游戏对象(GameObject，显示在 Hierarchy 面板中)都有 Transform 组件用于控制游戏对象的位置、旋转和缩放。简而言之，我们可以在参考手册中查找相机的 Transform 组件。

(1) 在 Hierarchy 面板中选中 Main Camera 对象。

(2) 单击 Transform 组件右侧的书本图标，如图 1-7 所示。

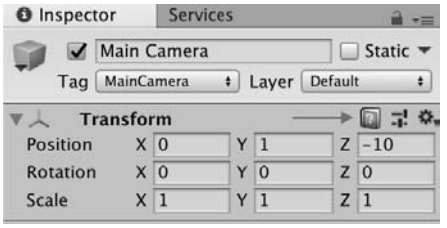

图 1-7　Inspector 面板中的 Transform 组件

刚刚发生了什么

此时，Web 浏览器将打开参考手册的 Transform 页面，如图 1-8 所示。Unity 中的所有组件都具有此功能。因此，如果想知道关于组件工作原理的更多信息，你应该知道该怎么做了。

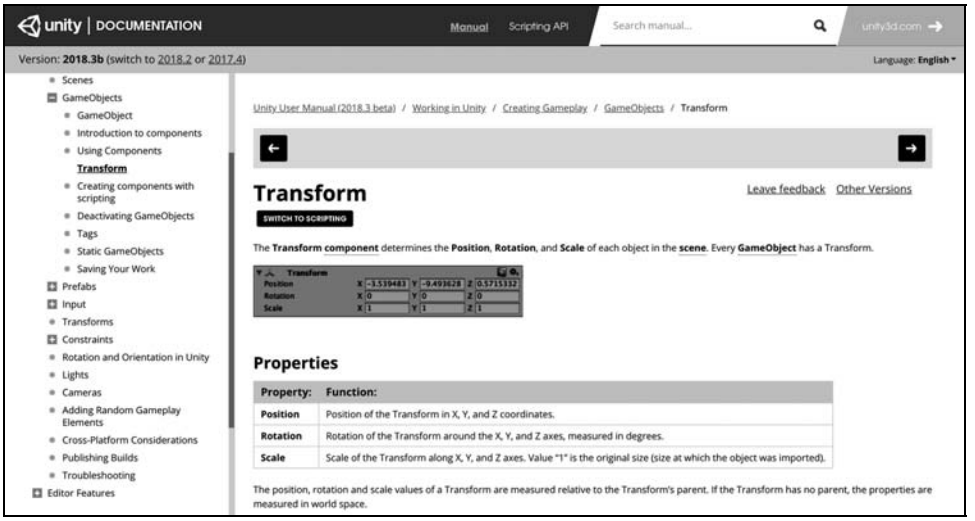

图 1-8　参考手册的 Transform 页面

2. 实践——使用参考脚本

现在，我们已经打开了参考手册。但是，如果想要得到与 Transform 组件相关的具体编码示例，该怎么办呢？这很简单——只需要访问参考脚本即可。单击组件或类名下方的 SWITCH TO SCRIPTING 链接。

刚刚发生了什么

参考手册将自动切换到 Transform 组件的参考脚本，如图 1-9 所示。

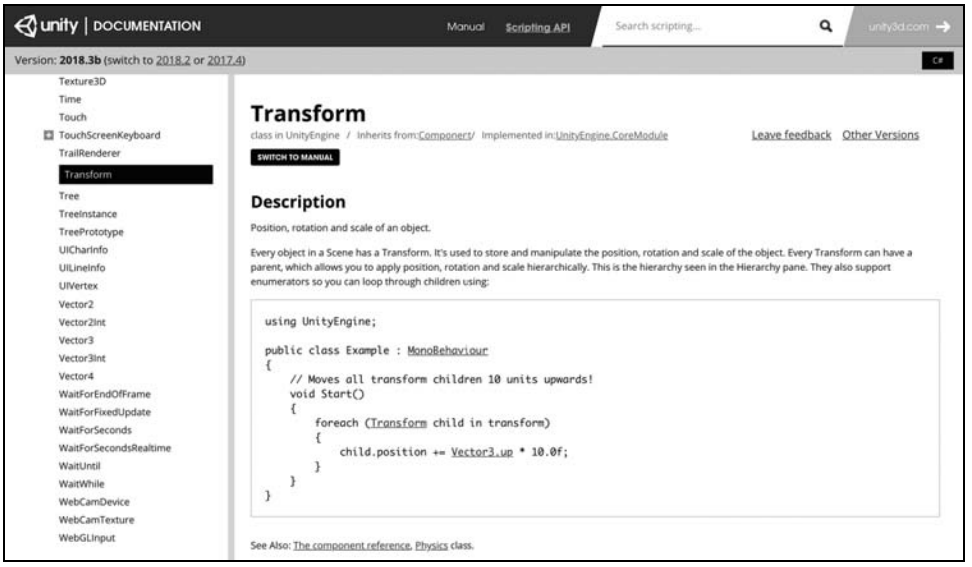

图 1-9　参考脚本

注意：
参考脚本一定是大型文档。然而，这并不意味着必须记住甚至熟悉参考脚本的所有信息才可以开始编写脚本。顾名思义，参考脚本仅供参考。

1.4.2　查找 C#资源

现在，我们需要关心一些 Unity 资源。通过访问链接 https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/index，可以查看微软的一些 C#文档。

注意：
还有其他许多 C#资源，从教程、快速入门到版本规范，各方面都有。如果感兴趣，可以访问 https://docs.microsoft.com/en-us/dotnet/csharp/。

实践——查找 C#类

加载编程指南链接，查找 C#的 String 类，如图 1-10 所示。

(1) 在左上角的搜索框中输入 Strings 并回车，也可向下滚动到语言部分(Language Sections)，然后直接单击 Strings 链接。

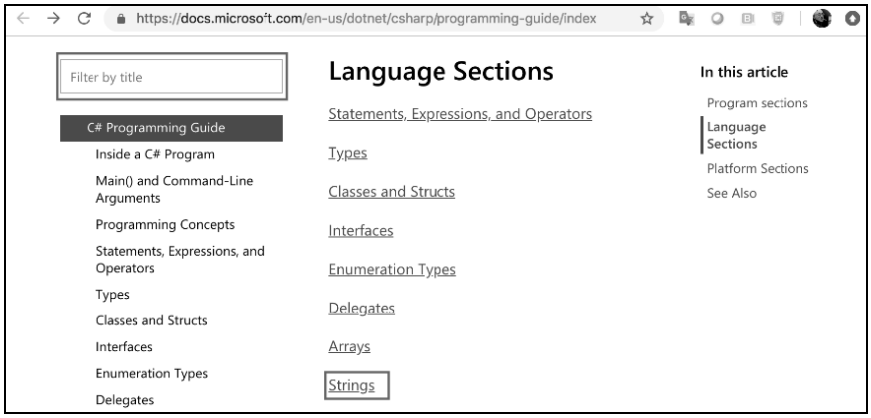

图 1-10　单击 Strings 链接

刚刚发生了什么

可以看到如图 1-11 所示的类描述页面。与 Unity 文档不同，C#参考和脚本信息全部混合在一起，这里唯一的可取之处就是右边的子主题列表，请充分利用好。

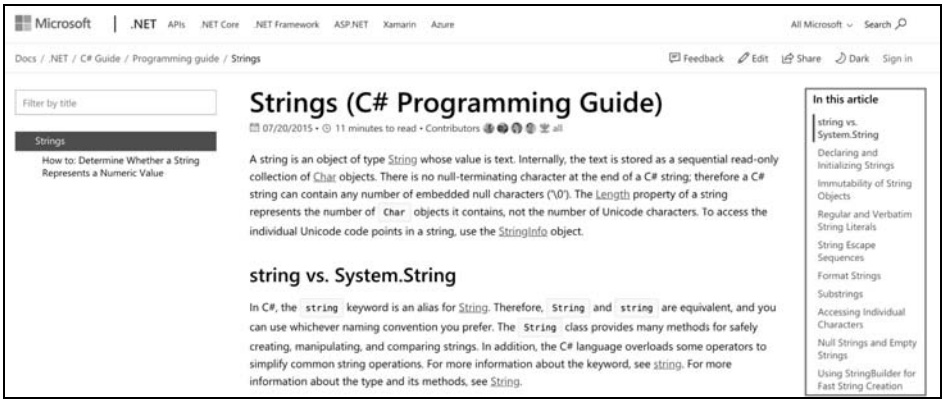

图 1-11　Strings 类的描述页面

1.5　小测验——处理脚本

1. Unity 和 Visual Studio 共享哪种类型的关系？
2. 参考脚本提供了有关使用特定 Unity 组件或功能的示例代码。从哪里可以找到有关 Unity 组件的与代码无关的更多详情？
3. 参考脚本是大型文档。在尝试编写脚本前，你记住了多少？
4. 命名 C#脚本的最佳时间是什么时候？

1.6 本章小结

本章介绍了很多预备信息，你如果此时就渴望编写一些代码，我们对你的急切之情表示理解。启动新项目、创建文件夹和脚本以及访问文档是 C#编程中容易让人忘记的主题。

请记住，本章介绍了很多后续章节中可能需要的资源。因此，不用害怕回来再次访问它们。可以将编程比喻为健身：锻炼越多，身体就越强壮。

第 2 章将开始介绍编程所需的理论、术语和主要概念。即使这些知识都是概念性的，我们也仍然会在 LearningCurve 脚本中编写第一行代码。请做好准备！