

Apress®

• 游戏开发经典丛书 •

[美] Sue Blackman 著  
罗岱 译

# Unity 4

## 3D游戏开发入门经典

(第2版): 多平台游戏开发全攻略

Beginning 3D Game Development with Unity 4:  
All-in-One, Multi-Platform Game Development



清华大学出版社

游戏开发经典丛书

# Unity 4 3D 游戏开发入门 经典(第 2 版):

多平台游戏开发全攻略

[美] Sue Blackman

罗 岱

著  
译



清华大学出版社

北 京

Sue Blackman

Beginning 3D Game Development with Unity 4: All-in-One, Multi-Platform Game Development

EISBN: 978-1-4302-4899-6

Original English language edition published by Apress, 2855 Telegraph Avenue, #600, Berkeley, CA 94705 USA. Copyright © 2013 by Sue Blackman. Simplified Chinese-Language edition copyright © 2015 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Apress 出版公司授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字：01-2014-2520

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

#### 图书在版编目(CIP)数据

Unity 4 3D 游戏开发入门经典(第2版): 多平台游戏开发全攻略 / (美) 布莱克曼(Blackman,S.) 著; 罗岱译. —北京: 清华大学出版社, 2015

(游戏开发经典丛书)

书名原文: Beginning 3D Game Development with Unity 4: All-in-One, Multi-Platform Game Development  
ISBN 978-7-302-38741-1

I. ①U… II. ①布… ②罗… III. ①游戏程序—程序设计 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2014)第 284285 号

责任编辑: 王 军 于 平

装帧设计: 牛艳敏

责任校对: 曹 阳

责任印制: 王静怡

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](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈: 010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者: 清华大学印刷厂

装 订 者: 三河市少明印务有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 42 字 数: 1130 千字

版 次: 2015 年 2 月第 1 版 印 次: 2015 年 2 月第 1 次印刷

印 数: 1~3000

定 价: 98.00 元

产品编号: 056940-01

# 译者序

本书结合实际游戏设计与开发，系统地深入论述了 Unity 3D 各个子模块的基本概念、使用方法和开发过程，是一本不可多得的 Unity 3D 游戏开发书籍。本书的作者 Sue Blackman 女士从事游戏开发与教学多年，对游戏设计、艺术资源创作和功能实现有深入理解，写作团队中也包括了技术专家，从而使本书在设计上和技术上都达到了较高的水平。

作者首先论述游戏开发的概况，按照游戏开发技术的演进过程，介绍了游戏开发的常见方法、基本概念和典型实现方案。接着作者分章节由浅入深地介绍了 Unity 3D 游戏引擎的各个功能模块，通过与实际游戏开发过程结合，作者细致地论述了游戏中不同的功能是如何利用这些 Unity 3D 功能模块实现的。

本书对于 Unity 3D 技术部分的论述系统而全面。作者并没有仅仅在性能和功能层面介绍 Unity 3D，除此之外，通过迭代过程，作者展示了游戏功能实现的不同方案。更重要的是，由于作者具有多年的游戏开发背景，因此书中所使用的功能实现方法都比较典型，而且作者还在书中介绍了很多一般的游戏功能实现策略，这有助于读者在自己的游戏开发过程中选择正确的技术路线。

本书中所使用的游戏案例的开发过程，每一步，作者都紧紧围绕需求，从技术方案选择、初步功能实现、功能完善、用户界面打磨及使用新功能替代旧实现的方法等方面来进行论述，读者可以在这个过程中体验游戏开发的实现过程并实际提高开发能力。

本书注重游戏设计过程，作者并不是直接给出功能实现的代码并解释代码为什么是这样的，相反地，对于功能实现，作者按照实际开发过程，先实现功能的原型，再反复迭代，不断完善功能的代码部分和用户界面部分，真实地反映了游戏开发的实际过程，能够有效帮助读者在学习过程中形成利用 Unity 3D 引擎进行游戏开发的实际能力。

本书并不是一本单纯讨论 Unity 3D 引擎使用的技术书籍，也不是一本单纯论述游戏设计的书籍。本书的作者是 3D 艺术家和交互应用设计师，同时参与游戏开发和教学多年，这使得本书对于游戏开发的论述，注重游戏的设计过程、用户体验和用户界面设计，有利于读者把握游戏开发的本质。

本书由北京林业大学艺术设计学院数字艺术系副教授罗岱翻译，参与本次翻译活动的还有北航世纪学院数字媒体技术教研室教师熊庆，以及北京林业大学动画艺术学研究生李誉、蒙少青和叶明胜，感谢你们在本书翻译过程中付出的辛苦和努力。此外，还要特别感谢清华大学出版社的编辑们对翻译提出的宝贵、细致的意见。由于时间仓促，译者水平有限，在翻译中的错误和疏漏之处在所难免，恳请各位读者和同行批评指正。

译者

# 作者简介



Sue Blackman 是南加利福尼亚的 3D 艺术家和交互应用设计师，同时也是一名教师。她在包括加利福尼亚艺术学院在内的一流社区大学和私立学校教授 3ds Max 和游戏课程已经超过 10 年时间。在 Activision 公司(美国动视公司，著名游戏开发商，其著名游戏系列包括使命召唤系列等——译者注)的分支机构，她曾经领导 3D 艺术家从事多款游戏的设计。Blackman 在业界工作多年，帮助财富 1000 榜单内的公司进行培训，例如波音(Boeing)、雷神(Raytheon)、诺格(Northrop Grumman)等公司，从而使这些公司能够开发严肃游戏和游戏格式的训练应用程序。她已经从事了 10 多年的实时 3D 引擎相关的商业开发，并在 Lynda.com 上为 Unity 引擎制作了多个视频教程。她是一名活跃的作者，多年以来，她以特约作者的身份不仅参与了多本 3ds Max 书籍的素材创作工作，同时也为多款严肃游戏 3D 著作应用程序编写培训手册。她也为 ACM Siggraph(美国计算机协会计算机图形专业组)编写严肃游戏方面的内容，这是她喜爱的领域之一。可以通过网站 [www.3dadventurous.com](http://www.3dadventurous.com) 或 Unity 的论坛来联系她。

# 技术编辑简介



Robert Reed 现居亚利桑那州凤凰城。在那里他获得了科林斯大学(Collins College)的学士学位。在 2010 年，他寻找合适的工具完成毕业设计时，第一次接触到了 Unity3D 游戏引擎，并一直使用至今。在 Unity3D 游戏引擎中，他主要使用的是 JavaScript。但是除了 JavaScript 之外，他也对其他的一些编程语言和脚本语言较为熟悉，这些语言包括：Java、C、C++、C#、ActionScript 3.0 和 Lua。

# 致 谢

感谢 Gabriel Acosta 和 Jenny Wang，他们对本书的游戏设计给了很多意见，参与了头脑风暴。特别感谢 Jenny 在开始菜单、石碑创意和 3DAventurous 网站的设计方面的贡献。

感谢 Binary Sonata 工作室的 Bryan Taylor 和 Sean McCurry，他们创作了神庙音乐和摄像机匹配代码。

还要依次感谢 Evalyn Hernandez、Zach Sarachman、Rosendo Rojas、Mario Im 和 Pablo Ortega，他们参与了一些章节的对话录制。特别感谢 Erik Toraason 专门为角色 Gimbok 所做的贡献。

感谢你们所有的人从自己的项目里挤出时间参与我的项目。对于你们提供的资源、创意和热情，我深怀感谢。

# 前 言

## 0.1 为什么写这本书

实时 3D 游戏已经出现 10 年多了。我们都玩这类游戏，按自己的喜好为这类游戏创作相关资源，甚至可能 Mod(游戏模组，指通过修改游戏原作创建新游戏——译者注)其中的一些。然而，Unity 游戏引擎突破性地提供了一种免费的或者是低成本的桌面游戏、手机游戏和单机游戏创作方式，创作你自己的游戏的唯一障碍是你的决心以及你愿意或能够投入到游戏创作中的时间。

时代变了。3D 技术已经普及，在影视领域我们可以看到很多以计算机图形(Computer Graphics, CG)为特色的公司，在游戏业界也是一样，很多休闲游戏正从 2D 转向 3D。Unity 迈出了大胆的一步，通过提供免费、强大的引擎，改变了高端引擎的价格模式，推动了业界的发展。引擎的价格不再是将你的创意变成游戏原型甚至是产品的障碍。

无论你的兴趣是休闲游戏还是其他更高的目标，如果你之前没有脚本经验，但是又很想把你的艺术创作资源和故事变成游戏，本书都能帮助你实现愿望。在现代的游戏引擎中，繁重的程序由引擎自己控制，而逻辑和游戏玩法(指游戏的目标、规则、挑战、交互等设定——译者注)可以由那些在创意方面比在程序方面更具特长的人通过脚本实现。

在本书中，我将从 3D 艺术家的角度来探讨游戏创作和设计决策，将逻辑和脚本分散在各个小的部分，同时向艺术家、初级游戏设计人员以及编程新手介绍实时游戏引擎概念。

本书以项目导向的方式撰写，所以你最终不但可以完成一个可以赏玩的游戏，并且能够获得一些可以复用到其他游戏中的脚本资源。同时，你也会体验到典型的设计决策，这些决策在整个游戏设计过程中都不得完成。你将从创建游戏的基础开始，通过增加功能，不断地优化它；在这个过程中，我将会解释特定选择背后的逻辑以及实时游戏的相关概念。

本书的项目基于第一人称点击式冒险游戏，功能包括装备包、状态管理、载入/保存功能，特别强调游戏创作中的视觉要素。即使你不单是第一人称射击游戏的爱好者，你也能学到如何通过引入娱乐性的和有趣的人物和功能，来优化你偏爱的游戏。

本书项目旨在向你介绍一种符合逻辑的设计决策和问题解决过程，这个过程对冒险游戏之外的其他游戏设计也有助益。它为创建并(更重要的是)最终完成你的游戏提供了一个框架和方法。本章内容的介绍将超出基本的 Unity 功能和标准资源(Unity 引擎自带的标准资源包，含游戏设计常用的基本功能和素材——译者注)的使用，还会介绍一些在在线教程和其他书籍里很少讨论的内容。项目需要的所有艺术资源都已提供。

### 作者的话

几年以前，在带领我的班级完成一个竞速游戏、一个第一人称的射击游戏和一个平台跳跃游戏之后，我决定最后一个小项目是经典的冒险游戏。令我惊讶的是班里的同学被这个游



戏的设计决策和逻辑所激发的热情。因为在当时没有这类游戏的现成教程，我们不得不从头设计游戏，从之前的项目中借鉴代码，创建自己的脚本，从 Unity 社区汲取大量的知识。这本书的创作就源于那段经历，既是为了需要，也是为了与他人分享探索的过程。

## 0.2 现代冒险游戏

在这个项目中，你将要创建一个经典的第一人称点击式冒险游戏的变种。冒险游戏有很多变种，对于没有或者只有很少脚本经验的艺术家或一般爱好者来说，这是一个很好的起点。故事、艺术资源、动画是这类游戏的主要组成部分，较慢的节奏允许更多的脚本优化方法，因为争分夺秒的响应时间是不需要的。如果你的目标是创建第一人称射击类游戏，本书可能不适合你。相反，如果你的口味特别一些，你也许会找到各种各样的信息，它们都可以用到你当前正在创作的游戏类型中。

在 Unity 4.0 中，加入了 Mecanim，它提供了理想的方法，用来实现探险游戏的一个常规特性——角色和对话树。即使你并非角色动画专家，也能够这样做，这非常有趣。更重要的改进在于更好地结合了物理特性以及各种特殊效果，使得本版书籍中的项目从“经典的”变成了“现代的”。

冒险游戏的一个最有意思的部分是可以搜集和使用奇怪的对象组合。因为装备包和状态管理的重要性，有几章专门介绍它们的设计和实现。Unity 游戏开发新手经常在社区论坛提问如何实现这些特性，但是他们很少得到回答，因为话题涉及的范围超出了 Unity 本身。在本书的末尾，你将获得额外的一些脚本、概念和经验，它们能够使你学到的知识应用到其他类型的游戏中。

交互冒险游戏对于独立开发人员也非常理想，因为这类游戏吸引了大批玩家。FireProof Games 的 *The Room* 游戏在本年度手机游戏排行中名列前茅，它是使用 Unity 开发的，这说明 Unity 已经在休闲游戏和手机游戏社区中占有重要的地位。

## 0.3 关于 Unity 游戏引擎

Unity 是小型工作室、独立游戏开发人员和那些希望开发自己游戏的开发人员的理想选择。它有广泛的用户群(在 2013 年夏季大约有 120 万左右)和特别活跃的用户社区，使得从新手到资深高手都能够及时地获得答案和分享信息。

Unity 是游戏开发的一个很好的切入点，它在价格与特性和功能间获得平衡。免费版的 Unity 允许人们在付出辛苦赚来的资金前试验、测试、开发和销售游戏。Unity 很便宜，功能齐备的 Pro 版是免版税的，允许人们以非常低的附加成本制作和销售游戏，这对休闲游戏市场非常重要。

多平台游戏市场——特别是 iPhone 和 Android 休闲游戏——在当前是非常流行的，Unity 的跨平台能力已经经过充分证明。最早基于 Mac 系统创作的应用程序可以发布到 Mac 和 Windows，2009 年 Unity 发布了 Windows 版，正如期望的那样，它为 Windows 平台的开发人员和艺术家提供了机会。从那时开始，Unity 开始不断地增加对 iPhone、Android、iPad 和 Wii 的支持，

并开始为支持 Xbox 360 和 PS3 进行开发。在 2013 年春季，Unity 推出了更大的惊喜：对于 Unity 的免费用户，iOS、Android、Windows 8 Mobile 和 Blackberry 的许可证现在或者很快也将免费。

最早的 Unity 引擎适配器试图从 Flash 或者 Director 进行迁移，这可以使得脚本环境更容易应用。毕竟很多 Unity 用户都有用 Flash 制作游戏并使用 ActionScript 的背景，这无论如何是一个先决条件。有很多关于在 Unity 中创建游戏的资料，包括第一人称射击游戏、竞速游戏、平台跳跃游戏等。即使你的兴趣在别的地方，也有很多提示和技巧可以从一些不太可能的地方收集到。应积极利用 Unity 网站([www.Unity3D.com](http://www.Unity3D.com))、Unity Forum([forum.unity3d.com](http://forum.unity3d.com))、维基百科([www.unifycommunity.com/wiki](http://www.unifycommunity.com/wiki))、Unity Answers([answers.unity3d.com](http://answers.unity3d.com))提供的资源，也包括数量巨大的与 Unity 引擎相关的其他私有网站。

Unity 文档也包含了非常丰富而有价值的信息，但是，正如任何技术都有特别的词汇一样，有时候很难找到你想要的东西。在你查询 Unity 或者 Unity3D 之前，先搜索互联网通常是找到难以捉摸的概念或功能的最简单的方法。可以通过 Help 菜单使用 Unity 帮助，但是通常使用在线版本更快。

## 0.4 我必须学习脚本吗

用 Unity 来设计游戏，你不必是一个程序员，但是你必须具备如下能力：识别脚本完成什么功能、知道如何调整脚本以符合你的需要、判断一个脚本是否符合你的需要。

大多数游戏玩法都要通过脚本实现，但是有数以百计的现成脚本都可以重用。Unity 自带了一些最常用的脚本。可以访问 Unity Forum、Wiki 和 Unity Answers 来搜索查询更多的脚本。很多论坛成员甚至会为较不擅长脚本的用户编写一些代码。在论坛的合作组中，你甚至能找到用脚本换艺术资源的脚本编写者。在本书的最后，你将看到 Unity 社区的丰富资源。

游戏最初的定义是一种交互；即使在基本由物理规律控制、由逻辑驱动、符合因果规律的游戏里，交互性也是使这些游戏区别于线性的、线索驱动的被动媒体的重要特征。即使是最“对艺术家友好”的游戏引擎也需要通过脚本来进行移动，更不要说简单的环境导航了。本书旨在教你每次熟悉几行脚本代码，并在这个过程中尽可能地提供视觉反馈。本书假设你还没有决定要系统学习脚本，但是乐于以稍被动的方式参与脚本的创作。

### 0.4.1 除了语法，脚本更重要的是逻辑

尽管很多人因学习一门新的语言而感到困难和胆怯，但可以这样想：大多数人在 35 岁前已经书写流利，书面语言是我们母语的一个子集。在某种程度上，书面语言有着自己的词汇和语法，并且它与脚本语言非常相似，因此，脚本更像我们母语的一个子集，而不是一门新的语言(对英语为母语的人而言——译者注)。

区别在于掌握这门语言的方法。对于书面语言，由于是我们的母语，因此我们并没有特意准备去学习它。然而，脚本需要我们学习、实验并最终掌握它，这主要是通过反复使用、测试排错以及一些被动的方法来达到，但绝不是死记硬背和紧张的考试。本书主要关于游戏设计与创作背后的逻辑，因此把脚本作为身临其境的体验，即游戏开发过程的一条副线。你

可以自由地选择在脚本上投入的精力。无论你怎么选择,本书开发的主要脚本将帮助你使用自己的想法和艺术资源来创建和扩展本书的游戏。

尽管如此,要明白脚本究竟在做什么,只有有限的概念和为数不多的关键字需要掌握。幸运的是,大多数脚本编写者都很友好地将注释包括在脚本中,解释脚本的作用,这使得复杂的脚本变得更易于理解。

本书的脚本以章为单位列出,脚本背后的逻辑在各章中解释,但是希望你可以发挥探索者的好奇心,从脚本的解释中获益并自己做一些实验。

## 0.4.2 数学是个问题吗

在 3D 业界,我常听到的一句话是“如果我知道数学这么有用,我当时真应该在学校好好学”。即使如此,大多数艺术家和设计师还是不想花时间提高他们的数学技巧。别担心!我的主要目标是帮助你创建游戏。一些游戏必须实现的功能非常容易使用,即使你不知道它的工作原理。我们大多数人都喜欢驾驶,但是对于汽车内部的发动机却没有广泛的知识。不要介意用同样的方法来对待脚本。

## 0.5 假设与前提

本书假设你已经至少熟悉 3D 资源和 3D 空间,但是书中也有相关概念的一个简短的综述,全书也包括了提示和技巧。

本书假设你没有太多的脚本经验(如果有的话),但是你愿意使用它将你的故事变为现实。

本书假设你现在有很少或者没有 Unity 游戏引擎的经验。

本书也假设你对游戏制作过程非常陌生,但是有非常强烈的愿望创建自己的实时 3D 游戏。

最后,本书假设如果你希望探索除经典点击式探险游戏之外的其他游戏类型,你会阅读本书,并思考如何将书中的各种技巧应用到其他游戏类型中来得到想要的效果。在休闲游戏市场,综合探险游戏、第一人称射击游戏和其他游戏类型的元素不仅能被认可,而且可以获得一些非常具有娱乐性的效果。

## 0.6 本书未涵盖的内容

本书不是介绍常规游戏设计的书籍;它更像是一本先导书籍,让你更习惯于分析需求和权衡选择。本书也不是在你需要自己实现游戏所有细节时,用来创作一个详尽的设计文档的书籍,毕竟在对引擎和自己的能力没有充分了解时,这是不大可能的。你将一点点地构建自己的游戏,在此过程中将逐步原型化创意和功能。

本书不是帮助你成为程序员的书籍,它还欠缺一些程序设计的最佳练习。书中的脚本尽可能地提供即时的视觉反馈,以帮助非程序员轻松进入脚本编写过程。尽管同一个目标通常都有多种实现方法,但在本书中,脚本采用艺术家或者设计师最容易理解的方式编写。在本

书中，脚本的介绍方式采用人们学习母语的方式。读者被脚本包围，沉浸在脚本中，允许修补脚本，从而可以慢慢地获得关于脚本的基本知识。根本不要担心如何记忆脚本。一些内容你将在整个项目中反复使用，同时鼓励你将另一些内容记录下来以备之后参考。

## 0.7 平台

本书使用 Unity 4.x 并在 Windows 7 和 Windows 8 环境下编写和完成。在 Mac 系统下，快捷键和操作系统文件存储方面的差异将在本书中进行标注，在帮助文件中也可以查到。

# 目 录

第 1 章 游戏开发简介	1	2.5.3 Assets 菜单	29
1.1 冒险游戏类型	1	2.5.4 GameObject	30
1.1.1 文字冒险游戏	1	2.5.5 Component 菜单	30
1.1.2 图形冒险游戏	2	2.5.6 Window 菜单	30
1.1.3 实时 3D 冒险游戏	3	2.5.7 Help 菜单	30
1.2 设计你的游戏	6	2.6 创建简单的对象	30
1.2.1 定义风格	7	2.7 选择和聚焦	35
1.2.2 划分场景	8	2.8 变换对象	35
1.2.3 第一人称还是第三人称	9	2.9 捕捉	37
1.2.4 动画	9	2.10 场景手柄工具	39
1.2.5 内容	9	2.11 灯光	41
1.3 管理项目	10	2.12 3D 对象	46
1.3.1 多角色	10	2.12.1 网格	46
1.3.2 选择游戏引擎	11	2.12.2 网格的子对象	46
1.3.3 需求	11	2.12.3 贴图	47
1.3.4 完成第一个游戏的提示	12	2.13 材质	50
1.4 实时渲染相对于预渲染的 新概念	14	2.14 本章小结	51
1.5 本章小结	19	第 3 章 脚本：通过实践学习	53
第 2 章 Unity UI 基础——入门	21	3.1 什么是脚本	53
2.1 安装和启动 Unity	21	3.2 脚本的成分	54
2.2 加载或创建一个新项目或场景	21	3.2.1 函数	54
2.3 布局	24	3.2.2 变量	60
2.3.1 Scene 视图	25	3.2.3 在游戏中拾取对象	64
2.3.2 Game 窗口	25	3.3 条件和状态	68
2.3.3 Hierarchy 视图	26	3.4 运算顺序	70
2.3.4 Project 视图	26	3.5 本章小结	70
2.3.5 Inspector 视图	27	第 4 章 地形生成：创建一个测试环境	73
2.4 工具栏	27	4.1 创建环境	73
2.5 菜单	28	4.1.1 创建一个地形对象	74
2.5.1 File 菜单	28	4.1.2 场景漫游导航	77
2.5.2 Edit 菜单	28	4.2 Terrain Engine	77
		4.2.1 地形结构	78

4.2.2	绘制纹理	81	7.3.2	箱子	208
4.2.3	树木	84	7.3.3	鲜花	210
4.2.4	地形的细节对象	100	7.4	阴影	212
4.2.5	创建你自己的地形资源	104	7.5	本章小结	216
4.2.6	Terrain Settings	107	<b>第 8 章</b>	<b>动作对象</b>	<b>217</b>
4.3	阴影	108	8.1	碰撞器	217
4.4	雾效	112	8.2	触发动画	221
4.5	本章小结	113	8.3	添加声音特效	224
<b>第 5 章</b>	<b>导航和功能</b>	<b>115</b>	8.4	设置一个双状态动画	229
5.1	导航	115	8.5	使用 Unity 的动画视图	231
5.1.1	方向键导航和输入	121	8.6	触发另一个对象的动画	244
5.1.2	玩转平台	132	8.7	局限性	246
5.1.3	碰撞墙	139	8.8	本章小结	246
5.1.4	首次编译	150	<b>第 9 章</b>	<b>管理状态</b>	<b>249</b>
5.2	本章小结	154	9.1	识别动作对象	249
<b>第 6 章</b>	<b>光标控制</b>	<b>157</b>	9.1.1	测试情节介绍	250
6.1	光标可视化	157	9.1.2	排练	250
6.2	自定义光标	159	9.1.3	动作对象	251
6.3	GUI 纹理光标	160	9.1.4	替身	251
6.3.1	Texture Importer	161	9.2	开发一个状态机	251
6.3.2	色卡	168	9.2.1	状态机的组件	251
6.3.3	光标位置	168	9.2.2	选取事件	252
6.4	硬件光标	169	9.2.3	分离状态元数据和转变动作	253
6.5	UnityGUI 光标	170	9.2.4	定义状态和转变	254
6.6	对象与对象的通信	173	9.2.5	可视化转变	254
6.7	鼠标悬停光标更改	174	9.2.6	组织信息	256
6.8	快速发布	182	9.3	查找表	257
6.9	对象对鼠标悬停的反应	183	9.3.1	评估结果	259
6.10	本章小结	183	9.3.2	解析字符串	259
<b>第 7 章</b>	<b>导入的资源</b>	<b>185</b>	9.3.3	游戏笔记	260
7.1	3D 艺术资源	185	9.4	ObjectLookup 脚本	261
7.2	导入设置	189	9.5	动作相关的消息	268
7.2.1	导入设置: Model	189	9.6	对象元数据	270
7.2.2	导入设置: Rig	194	9.6.1	状态相关的元数据	270
7.2.3	父对象	196	9.6.2	处理选取事件	273
7.2.4	导入设置: Animations	198	9.6.3	添加元数据	275
7.3	设置材质	204	9.6.4	激活状态引擎	277
7.3.1	钥匙	204	9.6.5	优化选取	284
			9.6.6	隐藏光标	285

9.6.7 创建定时器 .....	285	<b>第 12 章 信息文本</b> .....	379
9.7 本章小结 .....	286	12.1 GUI Skin .....	379
<b>第 10 章 探索转变</b> .....	289	12.2 文本可见性 .....	388
10.1 处理辅助对象 .....	289	12.3 使用对象元数据 .....	392
10.2 处理对象的可见性 .....	292	12.3.1 调整文本 .....	394
10.2.1 使用标签 .....	292	12.3.2 激活动作消息 .....	397
10.2.2 可见性 .....	296	12.3.3 最后的优化 .....	404
10.2.3 处理不透明度变化 .....	298	12.4 本章小结 .....	405
10.2.4 开发非关键帧渐变 .....	301	<b>第 13 章 装备包逻辑</b> .....	407
10.2.5 完成可见性变化 .....	303	13.1 使用层 .....	407
10.3 处理特殊情况 .....	307	13.2 创建装备包屏幕 .....	411
10.4 确保获得玩家焦点 .....	312	13.3 添加装备包图标 .....	417
10.4.1 阻断玩家输入 .....	312	13.3.1 2D 对象控制 .....	421
10.4.2 引导玩家焦点 .....	314	13.3.2 基于光标的响应 .....	426
10.4.3 探索 Lerp .....	314	13.4 在 3D 场景中删除光标 .....	427
10.4.4 摄像机匹配元数据 .....	320	13.5 本章小结 .....	430
10.4.5 改善摄像机匹配 .....	322	<b>第 14 章 管理装备包</b> .....	431
10.5 本章小结 .....	328	14.1 管理装备包对象 .....	431
<b>第 11 章 物理和特效</b> .....	329	14.2 装备包布局 .....	441
11.1 添加新资源 .....	329	14.3 装备包溢出 .....	444
11.2 物理 .....	333	14.4 设置限制 .....	448
11.3 结合物理和关键帧动画 .....	336	14.4.1 装备包网格限制 .....	449
11.3.1 布料 .....	336	14.4.2 箭头限制 .....	452
11.3.2 落石 .....	344	14.5 从装备包添加和移除对象 .....	454
11.3.3 引入随机性 .....	347	14.5.1 移除装备包对象 .....	454
11.3.4 找到水晶 .....	350	14.5.2 将对象添加到装备包 .....	456
11.4 粒子系统 .....	352	14.5.3 组合条目 .....	457
11.4.1 Shuriken 基础 .....	354	14.5.4 错误的拾取 .....	460
11.4.2 粒子类型 .....	354	14.5.5 再次访问动作对象 .....	463
11.4.3 倾泻粒子 .....	357	14.6 本章小结 .....	464
11.4.4 粒子组 .....	361	<b>第 15 章 对话树</b> .....	465
11.5 其他特效 .....	369	15.1 游戏剧情 .....	465
11.5.1 创建激光柱 .....	369	15.2 开始一个对话 .....	466
11.5.2 建造激光设备 .....	370	15.2.1 筛选话题 .....	469
11.5.3 激活激光 .....	371	15.2.2 获得回答 .....	470
11.5.4 创建 UV 动画脚本 .....	373	15.2.3 分支对话 .....	478
11.5.5 应用 UV 动画脚本 .....	374	15.3 本章小结 .....	492
11.6 本章小结 .....	378		

<b>第 16 章 Mecanim 和角色</b> .....	493	17.4 本章小结	576
16.1 添加角色	493	<b>第 18 章 建立游戏</b>	579
16.1.1 导入角色	493	18.1 完成主关卡	579
16.1.2 设置替身	495	18.1.1 阅读材料	579
16.1.3 定义动画剪辑	498	18.1.2 找到 Golden Sleeve	584
16.2 控制角色	501	18.1.3 搭乘木筏	586
16.2.1 Animator Controller	501	18.1.4 定位水晶	590
16.2.2 作一些改变	503	18.1.5 进入神庙	591
16.2.3 转变	503	18.1.6 遇见 Gimbok	596
16.2.4 脚本化动作	507	18.1.7 遇见 Kahmi	602
16.2.5 混合树	509	18.1.8 隧道地图	606
16.2.6 图层和图层遮罩	512	18.1.9 获取光源	608
16.2.7 物理特性	514	18.1.10 隧道	610
16.3 NPC 控制器	516	18.2 终极关卡	615
16.3.1 调整游戏	517	18.3 本章小结	617
16.3.2 同步音频和动画状态	519	<b>第 19 章 菜单与关卡</b>	619
16.4 组合角色和对话	522	19.1 游戏菜单	619
16.5 与 Dialogue Manager 进行通信	535	19.1.1 MenuManager 脚本	619
16.6 打包测试场景	537	19.1.2 主菜单	622
16.6.1 准备 ObjectLookup	537	19.1.3 确认对话框	629
16.6.2 向游戏中添加 NPC	538	19.1.4 制作人员	632
16.7 本章小结	541	19.1.5 追踪菜单模式	633
<b>第 17 章 游戏环境</b>	543	19.1.6 暂停导航和场景交互	634
17.1 添加最后的资源	543	19.2 音频	636
17.1.1 准备新的资源	543	19.2.1 声音特效	636
17.1.2 重论材质和光照贴图	545	19.2.2 环境声音	637
17.1.3 在 Beast 中使用导入的光照贴图	550	19.2.3 音乐	637
17.1.4 进入神庙的低层平台	551	19.2.4 声音音量	638
17.2 设置神龛	558	19.2.5 最终关卡菜单	638
17.2.1 保持水神殿湿润	559	19.3 关卡保存与加载	639
17.2.2 照亮光神龛	560	19.3.1 用 PlayerPrefs 存储	639
17.3 设置迷宫	560	19.3.2 保存到文本文件	640
17.3.1 检查迷宫几何体	561	19.3.3 保存游戏数据	642
17.3.2 管理迷宫功能	561	19.3.4 加载保存的数据	645
17.3.3 检查落点访问	563	19.4 在关卡之间漫游	651
17.3.4 调整几率	570	19.5 开始菜单	652
17.3.5 创建一个“复活节彩蛋”	575	19.6 最终任务	653
		19.7 本章小结	655



# 第 1 章

## 游戏开发简介

在本书的第 1 版中，我们介绍了经典的点击式冒险游戏类型，它不仅提供了一种宽容的游戏创作环境，还允许读者探索提示、技巧和各种技术，通常这些内容在其他的 Unity 书籍中被忽略。本版次将经典冒险游戏更新为它的现代形式，即现代冒险游戏。通过 3D 的世界、物理属性、实时特效和 3D 角色来提升玩家的体验，当玩家在这个世界中探索时，他搜集可能被使用到的物件，破解谜题，这些都基于物理和逻辑的控制。

另外，这类游戏吸引各种类型的玩家，包括那些从未玩过第一人称射击游戏的玩家，它也成为休闲游戏市场的一个理想工具。考虑到你可能希望冒险中包括免费的武器使用，随着内容的深入，这一点也将被介绍。任何项目在开始时总需要一些探索。在开始时进行批判性思考和调查从长远来看是会节省时间的。

### 1.1 冒险游戏类型

如果你年龄足够大，那么可能已经经历了冒险游戏的发展史，基于那些最成功的、经典的游戏的特性，你可能已经有了自己游戏的设计思路。如果你还年轻，可以通过各种网站获得相关的信息。无论如何，总可以找到很多有益的想法，特别是在技术变化为游戏内容提供了新的道路的今天。

#### 1.1.1 文字冒险游戏

“你站在一条道路的尽头……”

所有冒险游戏的鼻祖可以勉强说是冒险，因为它们是基于文字的游戏，最早在 20 世纪 70 年代中期由 Will Crowther 设计。在 *Pong* 游戏(乒乓游戏)和 *PacMan* 游戏(吃豆人游戏)风靡的年代，文字游戏迎合了那些具有灵活的头脑而不是灵活的手指的人们，这是一个启示。在最早的也被叫做 *Colossal Cave*(巨大洞穴)的版本中，Crowther 设定了一个场景，胆大的冒险者将探索一个巨大的地下洞穴系统，搜集钱财并对付不时出现的怪物。据报道，在 Crowther 根据自己的探索和已有的调查建立了肯塔基州猛犸洞洞穴系统并开发了矢量地图后，该游戏变得流行起来。

冒险于是成为一种游戏类型，在游戏里，叙述很优美并充满乐趣，这部分是归因于词汇和对用户输入的解析。

Infocom 公司的 *Zork*(魔域帝国)系列呈现了一个巨大的地下王国——Flathead 王朝以及王国的钱币 zorkmid。这个游戏系列留下了一些难忘的语句，例如“你的灯快熄灭了”、“你不能从那儿到这儿”，此外还有很多笑话。那些知道恐惧的人会告诉你，当你的灯快熄灭时，你就可能被吃掉。