

Unity 3D 游戏开发 (第2版)

宣雨松◎编著

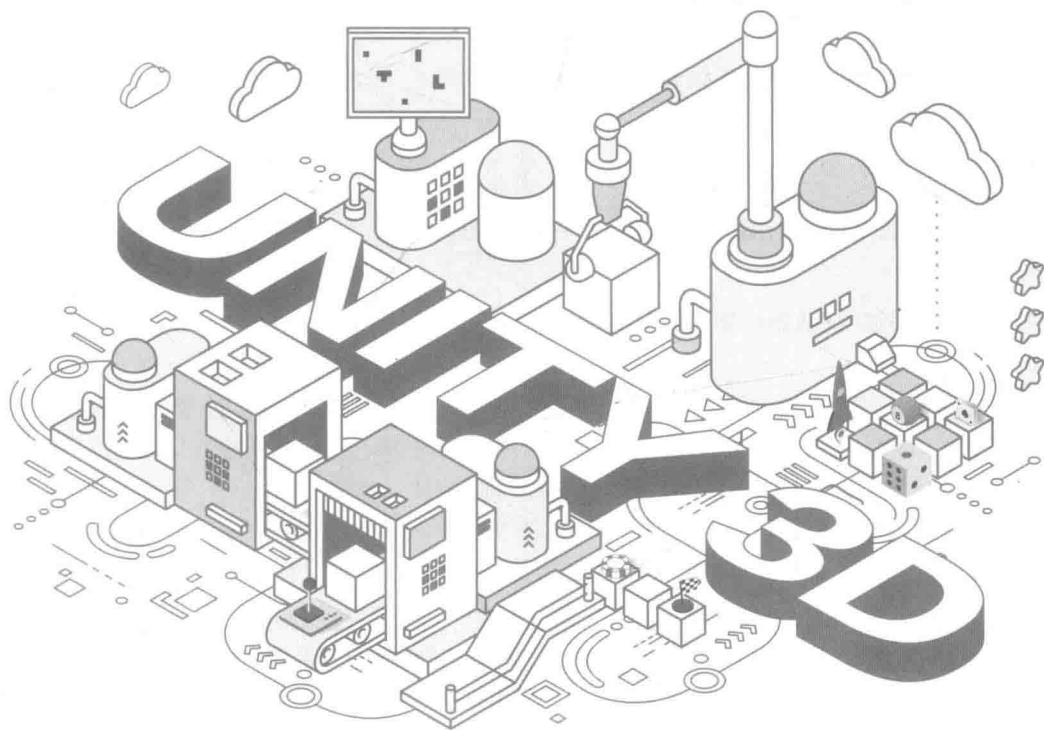


中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创



Unity 3D 游戏开发 (第2版)

宣雨松◎编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Unity 3D游戏开发 / 宣雨松编著. -- 2版. -- 北京:
人民邮电出版社, 2018. 9
(图灵原创)
ISBN 978-7-115-49294-4

I. ①U… II. ①宣… III. ①游戏程序—程序设计
IV. ①TP317.6

中国版本图书馆CIP数据核字(2018)第204995号

内 容 提 要

Unity 是一款市场占有率非常高的商业游戏引擎, 横跨 25 个主流游戏平台。本书基于 Unity 2018, 结合 2D 游戏开发和 3D 游戏开发的案例, 详细介绍了它的方方面面, 内容涉及编辑器、游戏脚本、UGUI 游戏界面、动画系统、持久化数据、静态对象、多媒体、资源加载与优化、自动化与打包等。

本书适合初学者或者有一定基础的开发者阅读。

◆ 编 著 宣雨松

责任编辑 王军花

责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京市艺辉印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 27.25

彩插: 2

字数: 644千字

2018年9月第2版

印数: 29 701 - 33 700册

2018年9月北京第1次印刷

定价: 89.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

序 一

第一次读雨松的书，也就是这本书的上一版，应该是五六年前的事情了。当时，我刚刚加入 Unity，而 Unity 在中国也只能算是刚刚起步，相关的技术书很少，雨松的书应该算是市面上最早的相关书之一了。在我个人来看，雨松的第一本书对于 Unity 引擎在中国的普及起到了很大的推动作用。可以毫不夸张地说，当时看雨松的书入门 Unity 的人，要远远多于看 Unity 用户手册的人。所以，当雨松问我是否可以他的第二本书作序时，我的回答是：非常荣幸！

在读这本书时，我已经离开 Unity 三年了，创办了自己的公司，每天解答着大量来自开发者的技术问题，其中有很大一部分是 Unity 相关的。原本以为这些烂熟于心的知识点，读起来不会再有什么味道，但是我发现我错了。原来哪怕是相同的问题，不同的人关注的点不同、所处的环境不同，那么解决方式也会完全不同。同时，相较于第一本书，这本书多了大量的内容锤炼和经验沉淀。书中关于 UGUI 的讲解、自动化打包的推荐以及 3D 游戏开发中的实战技巧，都是雨松这几年来的一线研发经验，同时也是目前大量游戏团队所需要了解和掌握的知识和技巧。我相信这本书的问世，可以帮助更多游戏团队解决更多棘手的问题。

我非常钦佩能够持续更新自己技术博客的人，因为这相当于在每天或每周进行大量工作的同时，还能将自己的经验和教训向外分享，从而去帮助更多的后来者解决问题。雨松就是这样一个人，他的博客是我闲暇时经常光顾的网站，通过阅读他的技术随笔来不断补充自己的知识点。我和雨松每次交谈时，都会非常欣喜地谈到，最近几年来游戏圈愿意分享经验和解答问题的人越来越多，分享的内容也越来越深入，而这些正在潜移默化地推动整个行业快速发展。如果你正在阅读此书，那么我希望你读完之后，不仅可以对 Unity 游戏开发有更为深入的了解，同时还可以将以往的经验进行总结、分享，并尝试去帮助身边的朋友解决问题。一个人的财富并不在于他拥有了多少金钱，而在于他影响了多少人。

张鑫，侑虎科技 CEO

序 二

移动互联网和手机智能化浪潮带来了全新的手机游戏模式。而随着手机游戏开发逐渐成熟，手机游戏开发门槛的降低，越来越多的开发者希望加入到这一行业中。Unity 作为一款优秀的游戏引擎，为广大游戏开发者提供了高效、简洁的开发流程，使游戏开发充满了乐趣。

Unity 最早诞生于丹麦，目前总部位于美国旧金山，是全球最大的游戏引擎开发商之一。

Unity 引擎的强大编辑器，使游戏开发过程简单、快捷。组件式开发和可视化开发，使过去艰涩难懂的开发情景变得清楚明了。

Unity 引擎具有优秀的默认渲染管线，即使是不了解计算机图形学的开发者，也可以渲染出炫丽的游戏画面。而最新的 Unity 版本也允许开发者自由定制渲染管线，给有丰富图形学经验的开发者更高的自由度，从而做出更加优秀的差异化产品。

Unity 引擎可以对接目前市面上几乎所有主流的开发工具，开发者可以根据自身需求，建立灵活的制作流水线。整个开发过程更加自由、可控。

总而言之，Unity 引擎是一款伟大的游戏引擎，它真正使游戏开发大众化，给更多有才华、有创意的开发者实现自己梦想的机会。

雨松是中国 Unity 技术普及的先行者。很多开发者（包括我）都是通过雨松的分享，开始学习和使用 Unity 的。可以说，雨松为 Unity 在中国的普及做了很大贡献，是 Unity 技术布道的先行者之一。

本书是雨松的第二本书，对 Unity 引擎进行了全面而翔实的讲解。书中的内容依然是雨松一贯的深入浅出风格。无论是对于初学者还是有经验的开发者，本书都会非常有帮助。

基于作者自己多年的经验和扎实的技术功底，雨松的这本书，可以说是 Unity 技术书中不可多得的佳作。

我与雨松相识近 10 年，能为他的这部大作写序，倍感荣幸。同时也希望通过此书，能有更多优秀的开发者参与到 Unity 开发中来。

祝你开卷有益。

高川，Unity 资深开发者

推 荐 语

雨松从 2012 年就开始写 Unity 相关的技术博客，而游戏蛮牛于 2013 年成立，我们一起见证了游戏行业的兴衰荣辱，也一起帮助很多游戏行业的从业者成长。雨松作为 Unity 领域的前辈，拥有丰富的知识沉淀和实战经验。这是他的第二本 Unity 图书，其内容系统、全面，对于想系统学习的开发者们，这无疑是一个不错的福利。

崇慕，游戏蛮牛创始人

Unity 在中国发展多年，以绝对的优势引领 3D 引擎类内容开发，俘获大量粉丝，目前仍是 3D 内容开发的首选引擎。本书不仅介绍了作者在工作中积累的经验，还涵盖了实际应用中的各类功能，很适合新手系统学习 Unity，并全方面了解其运行原理。

杨博，Unity 圣典社区创始人，威爱教育生态公司 CTO

本书条理清晰，覆盖了 Unity 游戏开发过程中所需的绝大部分知识点，稍有基础的新手按顺序阅读，即可快速了解游戏制作的整个过程。而本书的精华在于作者大量商业项目经验的详细呈现，即使是老司机，也会受益匪浅。

陈小飞，游族网络技术总监

本书比较全面地介绍了在 Unity 项目开发过程中经常用到的一些知识以及作者的一些经验总结。此书非常适合 Unity 初学者，同时书中的一些知识点对中高级开发人员也有一定的借鉴意义。

窦玉波，天马时空技术总监，游戏技术全能专家

本书是面向 Unity 游戏开发者的进阶指南，更宝贵的是作者结合过去几年的商业游戏实战经验，对游戏框架、 workflow、解决方案和性能调优等做了很详细的描述。我相信通过阅读本书，能让读者的游戏开发经验和技巧更上一层楼。

张宏亮，蓝港互动技术 VP

雨松结合自身经验撰写了此书。书中全方位覆盖 Unity 引擎技术，比较适合 Unity 初级开发者阅读。同时作为游戏开发从业人员，你可以通过阅读本书，点亮自身技能树，拓展眼界，作出更深入的思考。

郭智，腾讯游戏《穿越火线》手游主程

自从 2011 年正式在中国推广 Unity 不久，我就认识了雨松。作为一位游戏开发专家，雨松对技术（特别是对 Unity 技术）的执着让我印象极为深刻。雨松不仅自己专注探讨，而且愿意和他人分享，此书为证。书中既有雨松对 Unity 的深刻理解，又有他的丰富经验总结。本书必定是广大从事游戏开发朋友们的福音。

郭振平，前 Unity 布道师，华宸互动 CTO

雨松这本书条理清晰、循序渐进，内容描述非常详细、面面俱到，会让 Unity 初学者快速、扎实地掌握 Unity 开发技能。同时书中包含了大量实战中积累的技巧和知识点，即使对 Unity 有一定使用经验的人，也会受益匪浅。

胡志波，网元圣堂技术总监

Unity 的从业者或多或少都知道雨松，或是看过他的上一本书作为入门，或是关注他的博客来获取新的开发经验。而雨松的这本新书则兼具两者的特点，既适合初学者作为入门读物，快速掌握 Unity 引擎的基本概念和操作，同时也包含了他在 Unity 项目实战中积累的经验，可供有一定基础的开发者用来提升自己的能力。我相信通过阅读这本书，读者一定会对游戏开发和 Unity 的用法更加了解与娴熟。

陈嘉栋，Unity 中国 Field Engineer，《Unity 3D 脚本编程》作者

雨松多年来始终在 Unity 一线开发，积累和分享了大量经验和知识，非常值得大家学习。书中的内容有深有浅，不管是新手上路还是资深开发，都可以学到很多技巧。本书最大的亮点在于结合项目开发过程中遇到的各种情况，我相信作为 Unity 开发者的你一定能从中收获颇丰。

钱康来，北京 UUG 社区讲师，UVP 价值专家

宣雨松是中国 Unity 引擎发展的布道者。Unity 在国内发展的蛮荒时代里，是他的《Unity 3D 游戏开发》第一版以及技术站点“雨松 MOMO 程序研究院”引领广大工程师走入 Unity 的殿堂。今天，这本姗姗来迟的《Unity 3D 游戏开发（第 2 版）》更是结合了 2D 和 3D 游戏开发的实战案例，由浅入深地讲解了基础入门到引擎最新的进阶操作。无论是作为 Unity 基础开发者的入门书，还是作为熟练工程师的速查手册，你都应该拥有这本书。

王亮，Unity 资深专家，钛核互动联合创始人

初学 Unity 最好的学习资料无疑是官方的手册和最佳实践，之后需要去学习各种游戏开发的领域知识。雨松的这本书除了介绍 Unity 的知识外，还加入了很多实际开发过程中的经验。基于这种综合，推荐初学者通过此书快速建立对 Unity 游戏开发的认知。

程小山，完美世界游戏开发工程师，《武林外传》手游主程

时隔6年，终于盼到雨松的第二本书了。在新书中，系统阐述了Unity引擎的方方面面，如UGUI、Atlas、Shader、Timeline等。另外，更为难能可贵的是，雨松把自己最近几年实战一线的游戏经验技巧等也融会贯通到本书中，如游戏优化、团队源码管控、Excel持久化、AssetBundle动态加载等。我相信摆在你面前的这本书，一定会令你大快朵颐的。

王文刚，微软 MVP，UVP 价值专家

前 言

Unity 是一款市场占有率非常高的商业游戏引擎，横跨 25 个主流游戏平台，在市面上遥遥领先。尤其是在手机游戏平台，Unity 已经处于无可撼动的霸主地位。在 App Store 和 Google Market 上，已经有太多用 Unity 开发的游戏了。另外，Unity 不仅仅是一款游戏引擎，它已经渗透到了各个传统行业，在建筑、医疗、工业、娱乐、虚拟现实、动画、电影和艺术等行业中，都可以看到 Unity 的身影。为了兼容各平台特性以及复杂的需求，Unity 的版本更新也非常及时，最新最酷的功能会第一时间给到开发者手中。

近些年，硬件设备处于飞速发展，这促使游戏开发得到进一步发展。尤其是现在的移动设备，移动硬件几乎每年都会得到一次提升，这意味着将能做出效果更加酷炫的手机游戏来。这无疑也是对开发者的一种挑战，意味着每年都需要学习和应用新技术来使游戏效果更加逼真。

Unity 底层是由 C++ 编写的。为了让开发者上手更加容易，它利用 Mono 跨平台的特性使用 C# 作为游戏脚本语言。为了与 C++ 底层接口交互，引擎还封装了 UnityEngine.dll 和 UnityEditor.dll，这样开发者只需要用 C# 就可以开发自己的游戏了。另外，在编辑模式下，Unity 也非常灵活，开发者可以任意拓展专属自己的编辑器。此外，它还提供了资源商店，从中能找到全球顶尖开发者上传的插件，其中大多数都是源码形式的，这给学习和工作提供了非常大的便利。另外，Unity 还提供了完善的开发文档，甚至每一条 API 都有详细的解释，尤其是用户手册中还有详细的例子以及功能介绍。此外，官网中还有很多视频教程供开发者学习。

Unity 不仅仅可以做 3D 游戏，也全面支持 2D 游戏的开发，尤其是最新版本的 Unity 2018，它增加了 2D 的骨骼动画，对 2D 物理和地形也有更好的支持。在 UI 方面，UGUI 作为官方支持的界面工具，太多游戏都采用它来制作游戏界面，3D 方面就更不用说了，全新的动画系统以及时间线编辑工具让开发者可以更好地驾驭它。

本书主要面向初学者或者有一定基础的开发者，无论你是转行学习还是入门学习，都可以阅读本书。通过阅读本书，读者可以快速学习并入门 3D 游戏开发。此外，本书还包含了很多实战经验，其中丰富精彩的案例同样适用于晋级的读者。书中每一章都包含丰富的实例和源码，尤其是拓展编辑部分以及自动化部分，都是非常宝贵的实战经验，读者可以直接将其应用在实际开发中。

阅读本书

书中所有例子的源代码都可以在图灵社区（www.ituring.com.cn）本书主页免费注册下载，下载的源码按章编号，如图 0-1 所示。查看源码前，请确保 Unity 已经无误地安装在了本机当中。按照图中所示找到章节源码对应的游戏场景文件，双击该场景文件即可打开游戏工程，继而查看阅读。本书是在 macOS 下讲解 Unity 开发，若需在 Windows 下查看源码，查看方式与 macOS 下完全一致。

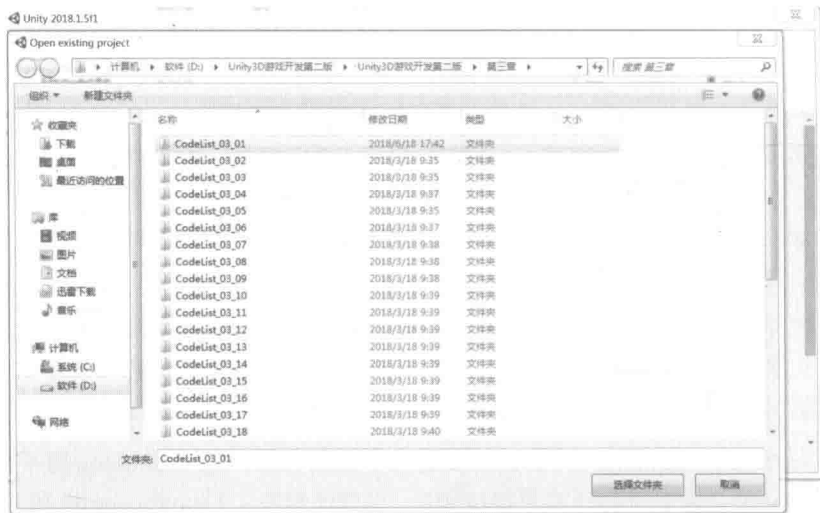


图 0-1 打开游戏工程

致谢

时至今日，我已经从事游戏开发十余年了。这本书是第 2 版，距离上一版出版已经 6 年了。其实这些年我也看到了上一版中的很多问题，包括一些读者给我的反馈，鉴于此，这本书我决定完全重写。这本书的写作也是一波三折，它本该在 2014 年就写完，可是一直拖到了 2018 年才完成。这几年我的身边发生了很多事情，我也做了好几款游戏，遗憾的是没有特别成功的作品。对于游戏开发，我的内心永远是狂热的，我将不忘初心，永远坚持钻研技术，励志走在研发的第一梯队中。

我最想感谢的人就是这些年努力的自己，我对技术非常痴迷，我非常享受解决一个复杂难题的瞬间。由于这份痴迷，我经常会上代码到凌晨，我最应该感谢的就是我的妻子，早在我创作上一版书的时候，她就在我身旁，并一直支持我到今天。我必须感谢图灵公司的编辑王军花，她教会我很多写作技巧并且审阅了本书的书稿，保证这本迟到了 4 年的书顺利出版。最后，祝图灵公司越做越好，为祖国的 IT 人才培养贡献出伟大的力量。

2018 年 7 月 19 日

宣雨松

目 录

第 1 章 基础知识	1	2.6.1 标题栏	22
1.1 Unity 简介	1	2.6.2 视图预览	23
1.2 跨平台与多工种协作	1	2.7 导航栏视图	23
1.3 Unity 版本	2	2.8 其他功能	25
1.4 Unity 内置资源或拓展资源	3	2.8.1 小锁头	25
1.5 示例项目打包与发布	5	2.8.2 窗口菜单	25
1.6 Unity 服务	7	2.8.3 保存组件参数	26
1.7 小结	8	2.8.4 Package Manager	26
第 2 章 编辑器的结构	9	2.9 小结	27
2.1 游戏项目	9	第 3 章 拓展编辑器	28
2.1.1 创建项目	9	3.1 拓展 Project 视图	28
2.1.2 打开项目	10	3.1.1 拓展右键菜单	28
2.1.3 版本管理	10	3.1.2 创建菜单	30
2.1.4 安装多个版本的 Unity	11	3.1.3 拓展布局	31
2.2 Project 视图	12	3.1.4 监听事件	32
2.2.1 创建资源	12	3.2 拓展 Hierarchy 视图	33
2.2.2 搜索资源	13	3.2.1 拓展菜单	33
2.2.3 搜索标签	14	3.2.2 拓展布局	34
2.3 Hierarchy 视图	16	3.2.3 重写菜单	35
2.3.1 创建游戏对象	16	3.3 拓展 Inspector 视图	37
2.3.2 搜索游戏对象	16	3.3.1 拓展源生组件	38
2.4 Inspector 视图	17	3.3.2 拓展继承组件	38
2.4.1 标题栏	17	3.3.3 组件不可编辑	40
2.4.2 组件栏	18	3.3.4 Context 菜单	42
2.5 Scene 视图	19	3.4 拓展 Scene 视图	44
2.5.1 导航栏	19	3.4.1 辅助元素	44
2.5.2 标题栏	20	3.4.2 辅助 UI	45
2.5.3 坐标系控制器	21	3.4.3 常驻辅助 UI	46
2.5.4 视图预览	21	3.4.4 禁用选中对象	47
2.6 Game 视图	22	3.5 拓展 Game 视图	48

3.6	MenuItem 菜单	49	4.4.3	serializedObject	90
3.6.1	覆盖系统菜单	50	4.4.4	监听部分元素修改事件	91
3.6.2	自定义菜单	50	4.4.5	序列化/反序列化监听	92
3.6.3	原生自定义菜单	51	4.4.6	ScriptableObject	95
3.6.4	拓展全局自定义快捷键	52	4.4.7	脚本的 Attributes 特性	97
3.7	面板拓展	53	4.4.8	单例脚本	99
3.7.1	Inspector 面板	53	4.4.9	定时器	100
3.7.2	EditorWindows 窗口	55	4.4.10	CustomYieldInstruction	101
3.7.3	EditorWindows 下拉菜单	56	4.4.11	工作线程	103
3.7.4	预览窗口	57	4.5	脚本编译	104
3.7.5	获取预览信息	59	4.5.1	编译规则	105
3.8	Unity 编辑器的源码	60	4.5.2	优化编译	105
3.8.1	查看 DLL	60	4.5.3	编译 DLL	105
3.8.2	清空控制台日志	61	4.5.4	脚本跨平台	106
3.8.3	获取 EditorStyles 样式	63	4.5.5	程序集定义	106
3.8.4	获取内置图标样式	64	4.5.6	日志	107
3.8.5	拓展默认面板	66	4.6	脚本调试	109
3.8.6	例子: 查找 ManagedStatic-References() 静态引用	67	4.7	小结	111
3.8.7	UIElements	69	第 5 章	UGUI 游戏界面	112
3.8.8	查询系统窗口	71	5.1	基础元素	112
3.8.9	自定义资源导入类型	73	5.1.1	Text	112
3.9	小结	74	5.1.2	描边和阴影	113
第 4 章	游戏脚本	75	5.1.3	动态字体	113
4.1	创建脚本	75	5.1.4	字体花屏	114
4.1.1	脚本模板	75	5.1.5	Image 组件	115
4.1.2	拓展脚本模板	76	5.1.6	Raw Image 组件	116
4.2	脚本的生命周期	78	5.1.7	Button 组件	116
4.2.1	脚本绑定事件	80	5.1.8	Toggle 组件	117
4.2.2	脚本初始化和销毁	80	5.1.9	Slider 组件	118
4.2.3	脚本更新与协程任务	82	5.1.10	Scrollbar & ScrollView 组件	119
4.2.4	停止协程任务	83	5.1.11	使用 ScrollRect 组件制作 游戏摇杆	120
4.2.5	使用 OnGUI 显示 FPS	84	5.2	事件系统	121
4.3	多脚本管理	85	5.2.1	UI 事件	122
4.3.1	脚本的执行顺序	85	5.2.2	UI 事件管理	124
4.3.2	多脚本优化	86	5.2.3	UnityAction 和 UnityEvent	126
4.4	脚本序列化	86	5.2.4	RaycastTarget 优化	128
4.4.1	查看数据	87	5.2.5	渗透 UI 事件	129
4.4.2	私有序列化数据	88	5.2.6	例子——新手引导聚合动画	131

5.3 Canvas 组件	135	6.3.5 拓展 Tile Palette	180
5.3.1 自适应屏幕	135	6.3.6 拓展 Tile	183
5.3.2 锚点对齐方式	137	6.3.7 更新 Tile	184
5.3.3 背景图全屏	138	6.4 2D 碰撞检测	186
5.3.4 布局组件	139	6.4.1 Collider 2D	186
5.3.5 Canvas 优化	139	6.4.2 Rigidbody 2D	187
5.4 Atlas	139	6.4.3 碰撞事件	189
5.4.1 创建 Atlas	139	6.4.4 碰撞方向	191
5.4.2 读取 Atlas	140	6.4.5 触发器监听	193
5.4.3 Variant	141	6.4.6 Effectors 2D	193
5.4.4 多图集管理	142	6.4.7 优化	194
5.5 UGUI 实例	142	6.4.8 计算区域	194
5.5.1 置灰	142	6.5 小结	195
5.5.2 粒子特效与 UI 的排序	145	第 7 章 动画系统	196
5.5.3 Mask & RectMask2D 裁切	147	7.1 模型	196
5.5.4 粒子的裁切	148	7.1.1 Mesh Filter	196
5.5.5 粒子自适应	152	7.1.2 Mesh Renderer	197
5.5.6 滑动列表嵌套	154	7.1.3 Prefab	197
5.5.7 UI 模板嵌套	156	7.2 动画编辑器	198
5.5.8 UI 特效与界面分离	157	7.2.1 编辑器面板	198
5.5.9 输入事件	161	7.2.2 在编辑器中添加事件	199
5.5.10 按钮不规则点击区域	162	7.3 导入类动画	200
5.5.11 更换默认 Shader	164	7.3.1 人形重定向动画	200
5.5.12 小地图优化	164	7.3.2 通用动画	202
5.5.13 查看 UGUI 源码	165	7.3.3 老版动画	203
5.6 小结	165	7.3.4 导入类动画事件	203
第 6 章 2D 游戏开发	166	7.4 动画控制器	203
6.1 Sprite	166	7.4.1 系统状态	204
6.1.1 2D 摄像机与分辨率自适应	167	7.4.2 切换条件	205
6.1.2 Sprite Renderer 排序	169	7.4.3 状态机脚本	206
6.1.3 裁切	170	7.4.4 IK 动画	207
6.2 Sprite 动画	171	7.4.5 Root Motion	209
6.2.1 创建 2D 动画	171	7.4.6 Avatar Mask	210
6.2.2 2D 动画控制器	173	7.4.7 层	211
6.3 Tile 地图	176	7.4.8 Blend Tree	212
6.3.1 创建 Tile	176	7.4.9 非运行播放动画	213
6.3.2 Tile Palette	178	7.4.10 Animator Override Controller	214
6.3.3 编辑 Tile	178	7.4.11 RuntimeAnimator- Controller	215
6.3.4 多 Tile 编辑与排序	179		

7.5	TimeLine 编辑器	215	8.4.1	YamlDotNet	252
7.5.1	创建 Timeline	216	8.4.2	序列化和反序列化	252
7.5.2	Activation Track	217	8.4.3	读取配置	253
7.5.3	Animation Track	217	8.5	小结	255
7.5.4	Audio Track	218	第9章 静态对象		256
7.5.5	Control Track	218	9.1	Lightmap	256
7.5.6	Playable Track	219	9.1.1	设置烘焙贴图	256
7.5.7	自定义 Track	221	9.1.2	实时光和烘焙光共存	258
7.6	Playables	222	9.1.3	灯光管理	258
7.6.1	播放动画	222	9.1.4	运行时更换烘焙贴图	258
7.6.2	动画混合	224	9.1.5	动态更换游戏对象	260
7.6.3	音频混合	225	9.1.6	复制游戏对象	262
7.6.4	自定义脚本	226	9.2	遮挡剔除	263
7.7	Constraint	227	9.2.1	遮挡与被遮挡	264
7.7.1	Aim Constraint	227	9.2.2	遮挡与被遮挡事件	265
7.7.2	Parent Constraint	228	9.2.3	动态剔除	267
7.7.3	脚本控制约束	229	9.2.4	自定义遮挡剔除	267
7.8	小结	229	9.3	Batching (静态合批)	268
第8章 持久化数据		230	9.3.1	设置静态合批	268
8.1	Excel	230	9.3.2	脚本静态合批	269
8.1.1	EPPlus	230	9.3.3	动态合批	269
8.1.2	读取 Excel	230	9.3.4	静态合批的隐患	270
8.1.3	写入 Excel	232	9.4	寻路网格	270
8.1.4	JSON	233	9.4.1	设置寻路	270
8.1.5	JSON 支持字典	235	9.4.2	连接两点	272
8.2	文件读取与写入	236	9.4.3	获取寻路路径	272
8.2.1	PlayerPrefs	237	9.4.4	动态阻挡	274
8.2.2	EditorPrefs	237	9.4.5	导出寻路网格信息	274
8.2.3	PlayerPrefs 保存复杂结构	238	9.5	反射探头	277
8.2.4	TextAsset	239	9.6	小结	278
8.2.5	读写文本	240	第10章 多媒体		279
8.2.6	运行期读写文本	240	10.1	音频	279
8.2.7	PersistentDataPath 目录	242	10.1.1	音频文件	279
8.2.8	游戏存档	243	10.1.2	Audio Source	280
8.3	XML	247	10.1.3	3D 音频	280
8.3.1	创建 XML	247	10.1.4	代码控制播放	281
8.3.2	读取与修改	248	10.1.5	混音区	283
8.3.3	XML 文件	249	10.1.6	Audio Mixer	284
8.4	YAML	251			

10.1.7	录音	284	11.4.7	加载场景	314
10.1.8	声音进度	285	11.4.8	卸载 AssetBundle	315
10.2	视频	287	11.5	游戏对象	315
10.2.1	视频文件	287	11.5.1	创建游戏对象	316
10.2.2	视频渲染模式	287	11.5.2	Transform 设置排序	316
10.2.3	视频自适应	288	11.5.3	删除节点	317
10.2.4	UI 盖在视频之上	289	11.5.4	获取游戏对象	317
10.2.5	视频渲染在材质上	290	11.5.5	管理游戏组件	318
10.2.6	视频 Render Texture	291	11.6	优化工具	319
10.2.7	播放工程外视频	291	11.6.1	重复无用资源	319
10.2.8	自定义视频显示	293	11.6.2	查看内存	321
10.2.9	视频进度	293	11.6.3	查看 CPU 效率	321
10.3	小结	294	11.6.4	自定义观察区间	322
11	第 11 章 资源加载与优化	295	11.6.5	Profiler 信息的导出与导入	323
11.1	编辑模式	295	11.6.6	Frame Debugger	324
11.1.1	加载资源	295	11.7	资源管理实例	325
11.1.2	实例化 Prefab	296	11.7.1	特殊内置的文件夹	325
11.1.3	创建 Prefab	297	11.7.2	隐藏文件夹	326
11.1.4	更新 Prefab	298	11.7.3	Resources 与 AssetBundle 无缝切换	326
11.1.5	卸载资源	299	11.7.4	资源加载策略	329
11.1.6	游戏对象与资源的关系	299	11.7.5	资源更新	329
11.2	版本管理	300	11.7.6	资源引用关系	329
11.2.1	.meta 文件	300	11.7.7	系统资源修改	332
11.2.2	多工程	301	11.7.8	AssetBundle 里的脚本	333
11.2.3	同步文件	302	11.7.9	热更新代码	334
11.2.4	SVN 外链	303	11.8	小结	335
11.3	运行模式	304	12	第 12 章 自动化与打包	336
11.3.1	引用资源	304	12.1	资源导入	336
11.3.2	Resources	305	12.1.1	监听导入事件	336
11.3.3	删除对象	306	12.1.2	自动设置贴图压缩格式	337
11.3.4	删除资源	306	12.1.3	自动设置模型	339
11.3.5	GC	307	12.1.4	禁止模型生成材质文件	340
11.4	AssetBundle	308	12.1.5	删除移动资源事件	341
11.4.1	设置 AssetBundle	308	12.1.6	选择性自动设置	342
11.4.2	依赖关系	310	12.1.7	主动设置	342
11.4.3	通过脚本设置依赖关系	311	12.1.8	待保存状态	343
11.4.4	压缩格式	311	12.1.9	自动执行 MenuItem	344
11.4.5	加载包体内的 AssetBundle	312	12.2	配置错误	345
11.4.6	下载 AssetBundle	313			

12.2.1	主动检查工具	345	13.2.3	主摄像机	387
12.2.2	被动检查工具	346	13.2.4	在 UI 上显示模型	387
12.2.3	导出类检查工具	346	13.2.5	Render Texture	389
12.2.4	导出 UI	346	13.2.6	在 Render Texture 上显示带特效的模型	390
12.2.5	生成 UI 代码	347	13.2.7	LOD Group	392
12.2.6	导出模型	351	13.3	场景管理	393
12.2.7	不参与保存对象	352	13.3.1	切换场景	393
12.2.8	导出场景	353	13.3.2	DontDestroyOnLoad	393
12.2.9	过滤无用场景	355	13.3.3	异步加载场景以及进度	393
12.3	自动打包	356	13.3.4	多场景	395
12.3.1	打包前后事件	356	13.3.5	多场景游戏对象管理	396
12.3.2	打包后自动压缩	357	13.3.6	场景切换事件	397
12.3.3	用 C# 调用 shell 脚本	359	13.3.7	多场景烘焙	398
12.3.4	等待 shell 结束	359	13.4	输入事件	398
12.3.5	shell 脚本调用 C#	360	13.4.1	全局事件	398
12.3.6	shell 脚本自动打包	362	13.4.2	射线	399
12.3.7	彻底解放程序员的双手	365	13.4.3	点选 3D 模型	400
12.4	自动构建图集与压缩	366	13.4.4	通过点击控制人物移动	402
12.4.1	UI 图集的压缩	366	13.5	物理碰撞	403
12.4.2	非 UI 图集压缩	370	13.5.1	碰撞器	404
12.5	小结	371	13.5.2	角色控制器	404
第 13 章	3D 游戏开发	372	13.5.3	碰撞区域	405
13.1	Shader	372	13.5.4	碰撞检测	406
13.1.1	固定渲染管线	372	13.5.5	碰撞触发器	407
13.1.2	可编程渲染管线	375	13.5.6	物理调试器	408
13.1.3	可编程渲染管线的表面着色器	378	13.6	实战技巧	409
13.1.4	深度排序	379	13.6.1	3D 头顶文字	409
13.1.5	透明	381	13.6.2	图文混排	410
13.1.6	裁切	382	13.6.3	图片数字	410
13.1.7	着色器变体采集	383	13.6.4	游戏对象缓存池	412
13.2	摄像机	384	13.6.5	游戏资源缓存池	414
13.2.1	3D 坐标转换屏幕坐标	384	13.6.6	运行时合并网格	414
13.2.2	3D 坐标转换 UI 坐标	385	13.6.7	运行时合并贴图	416
			13.7	小结	419

第 1 章

基础知识

Unity 是一款 3D 跨平台的次世代游戏引擎，“Unity”一词的释义为“团结”，好比集合所有人的力量一起来完成这件伟大的巨作。Unity 于 2005 年发布了第一个版本，至今已经 13 年了。起初，该公司致力于游戏引擎的研发，由于 Unity 具有强大的跨平台开发能力与绚丽的 3D 渲染效果，因而很快被广大游戏开发厂商以及开发者所信赖，尤其是在手游领域已经处于无可撼动的地位。近年来，该公司处于飞速发展中，它涉及的层面已经不仅仅局限于游戏引擎，还在 AI、VR、教育、医疗、工业、动画片等领域大放异彩。

1.1 Unity 简介

Unity 是一款标准的商业游戏引擎，商业引擎的主要特点就是：收费、封闭源码和功能强大。关于收费情况，Unity 目前分为 3 个版本：个人版、加强版和专业版。个人版是完全免费的，只能使用引擎核心的基础功能，适用于刚接触 Unity 的初学者或者学生。加强版和专业版可用于发布正式的游戏，当然也是收费的。专业版更加昂贵，但是它可使用的 Unity 服务是最全面的，主要包括 Unity 的 Analytics 分析服务等。如果你的公司或独立工作室的启动资金或年度收入大于 20 万美元的话，可以购买 Unity 专业版，反之可以购买 Unity 加强版。

1.2 跨平台与多工种协作

Unity 是一个跨平台游戏引擎，其中跨平台分为开发跨平台和发布跨平台。开发跨平台就是开发者可以在不同的操作系统下开发 Unity 游戏，目前可以跨 Windows 和 macOS 这两种操作系统。而发布跨平台则表示使用 Unity 开发出来的游戏能在多平台下运行。如图 1-1 所示，目前 Unity 已经可以横跨 25 个主流游戏平台，是不是很强大呢？不过每个平台可能有些自己特有的需求，详细的跨平台信息可以在如下网址中查阅到：<https://unity3d.com/cn/unity/features/multiplatform>。

传统的游戏引擎更多的是程序员在使用，而 Unity 则是多工种同时配合使用。Unity 提供了丰富的编辑工具，可以辅助美术人员以及策划人员在引擎中进行编辑工作，比如 TimeLine 时间线工具，程序员可以在不需要编写一行代码的情况下编辑各种复杂的剧情动画或者动画片。此外，Unity 还提供了拓展编辑器的接口，程序员可以开发自定义的辅助编辑工具供其他工种人员使用。