

- 深受读者赞誉的Unity游戏开发原创经典，资深Unity游戏开发工程师十年经验分享
- 根据Unity 2018改版升级，多款热门游戏实战教学，Unity3D/2D游戏开发核心技术揭秘
- 新增VR/AR游戏开发、Shade图形编程、Lua脚本交互与热更新等，内容更精彩



Unity 3D/2D

手机游戏开发从学习到产品 (第4版)

金玺曾 著

企业级游戏开发必备



清华大学出版社



Unity 3D\2D

手机游戏开发 (第4版) 从学习到产品

金玺曾 著

清华大学出版社
北京

内 容 简 介

本书以实例教学为主线,循序渐进地介绍了 Unity 2018 在手机游戏开发方面的不同功能。第 1 章,由零开始,引导读者熟悉 Unity 编辑器的各个功能模块和特性。第 2~4 章是 3 个不同特色的 3D 游戏实例,使读者对 Unity 游戏开发有一个较全面的认识。第 5 章是一个 2D 游戏实例,全方位地介绍了 Unity 在 2D 游戏方面的应用。第 6 章和第 7 章重点介绍了 Unity 在网络方面的应用。第 8~10 章介绍了如何将 Unity 游戏移植到网页、iOS 和 Android 平台。第 11 章详细介绍了 Unity 的新 GUI 系统。第 12 章是关于创建 Unity 游戏美术资源的工作流程。第 13 章和第 14 章分别介绍了行为树和 Play Maker 两款插件,第 15 章介绍了使用 HTC Vive 创建 VR 应用,第 16 章介绍了 Unity 结合 Vuforia 在 AR 方面的应用,第 17 和第 18 章介绍了 Shader 图形编程和 Lua 脚本在 Unity 中的应用。本书最后还附有 C# 语言的快速教程,帮助缺乏程序开发基础的读者快速入门,同时也包括 Unity 编辑器菜单栏的中英文对照表供读者查阅。

本书提供了所有实例的源代码与素材文件,供读者上机练习使用,读者可从网上下载本书资源文件。

本书适合广大游戏开发人员、游戏开发爱好者、软件培训机构以及计算机专业的学生等使用。

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 3D\2D 手机游戏开发:从学习到产品/金玺曾著. —4 版. —北京:
清华大学出版社, 2019

ISBN 978-7-302-52581-3

I. ①U… II. ①金… III. ①手机软件—游戏程序—程序设计 IV. ①TP317.67

中国版本图书馆 CIP 数据核字(2019)第 043804 号

责任编辑:王金柱

封面设计:王翔

责任校对:闫秀华

责任印制:李红英

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:三河市铭诚印务有限公司

经 销:全国新华书店

开 本:190mm×260mm

印 张:31

字 数:794 千字

版 次:2013 年 8 月第 1 版

2019 年 4 月第 4 版

印 次:2019 年 4 月第 1 次印刷

定 价:99.00 元

产品编号:078061-01

前 言

编写本书的目的

Unity, 也称 Unity 3D, 是近几年非常流行的一款 3D 游戏开发引擎, 它的特点是跨平台能力强, 支持 PC、Mac、Linux、网页、iOS、Android 等平台, 移植便捷, 3D 图形性能出众, 同时也支持 2D 功能, 为众多游戏开发者所喜爱。在手机平台, Unity 几乎成为 3D、2D 游戏开发的标准工具。

游戏开发是一项复杂的工作, 本书在编写过程中十分注重与实际开发相结合, 全书以实例为基础, 使读者在较短的时间内能快速掌握 Unity 2018 的各种工具和开发技巧, 并应用于实践中。

本书主要内容

本书为第 4 版, 总体上更新了大部分代码和截图, 改进了细节, 确保与 Unity 的新版本保持一致, 下面是各章的内容概要。

第 1 章介绍 Unity 编辑器的各个功能模块, 突出介绍了 Unity 的一些特点。

第 2 章是一个太空射击游戏教程, 这是一个入门级的教程, 从如何创建一个脚本, 到一个完整的游戏有较为细致的介绍, 最后介绍了使用对象池插件 Pool Manager 缓存游戏对象。

第 3 章是一个第一人称射击游戏教程, 涉及人工智能寻路、动画、摄像机控制等内容。

第 4 章是一个塔防游戏教程, 介绍了创建更为复杂的关卡, 详细地介绍了如何自定义 Unity 编辑器, 灵活运用协程实现相对复杂的逻辑, 配置和生成敌人。

第 5 章介绍 Unity 在 2D 游戏方面的应用, 包括创建 Sprite、动画的播放和一个较为完整的 2D 捕鱼游戏实例。

第 6 章介绍 Unity 在 HTTP 网络通信方面的应用, 还涉及 PHP 和 MySQL 的基础应用, 使 Unity 游戏可以与 Web 服务器进行通信, 上传下载得分记录等。

第 7 章是一个完整的、基于 TCP/IP 协议的聊天实例, 介绍使用 Unity 创建聊天客户端, 并使用 .NET 开发环境创建聊天服务器端。与前一版相比, 本章的内容编排改动较大, 示例更加简练。

第 8 章介绍如何将 Unity 游戏运行在 HTML5 网页上。

第 9 章介绍如何将 Unity 游戏移植到 iOS 平台, 从如何申请开发资格到测试、发布 iOS 游戏都有详细的介绍, 与前一版相比, 本章增加了使用 iOS 命令行编译工程的示范。

第 10 章介绍如何将 Unity 游戏移植到 Android 平台, 并详细介绍了几种为 Unity 开发 Android 插件的方法, 结合百度地图 SDK 完成完整的应用实例。本章最后还更新了 AssetBundle 的内容。与前一版相比, 本章对使用 Android Studio 进行开发的一些细节做了补充。

第 11 章全面介绍 Unity 新 GUI 的大部分功能和细节，并附有大量示例，最后还介绍了 DOTween Pro 和 EnhancedScroller 两款常用插件的使用。

第 12 章主要是对创建 Unity 游戏美术资源的介绍，包括光照系统、Lightmap、PBR Shader 和两足动画系统等，同时还结合了一些 3D 动画软件的介绍，如 3ds Max 和 Maya。和前一版相比，本章增加了 Unity 动画工具 Timeline 的教程示例。

第 13 章介绍行为树 AI 插件 Behavior Designer（行为设计师），它主要应用在 AI 方面，无论是程序员还是游戏开发爱好者都能找到使用它的乐趣。

第 14 章介绍 Unity 社区中最有名的插件 Play Maker，它和 Behavior Designer 都属于可视化编程产品，Behavior Designer 的设计模式是基于行为树，Player Maker 是基于状态机，后者有更广泛的用户群。和前一版相比，本章的截图更新到 PlayMaker 版本 1.9，并修正了一些描述上的问题。

第 15 章介绍了使用 HTC Vive 创建 VR 应用的基本流程，实现包括拾取、投掷等很多基本功能。

第 16 章介绍了 Unity 结合 Vuforia 在 AR 方面的应用。

第 17 章介绍了 Shader 图形编程的基本概念，Shader Lab 和 CG 语言的运用及大量示例，最后还讲述了如何创建全屏特效 Shader 实现后期效果。

第 18 章介绍了编写 Lua 脚本的基本概念，如何在 Unity 中与 Lua 脚本进行交互及实现 Lua 脚本的热更新等。

读者对象

本书的读者主要是游戏开发程序员和 Unity 爱好者，部分内容也适合游戏策划和游戏艺术家作为参考。

对于本书的完成，要特别感谢王金柱编辑给予的帮助和指导，感谢我的妻子在深夜帮助我校对书稿，还要感谢我的儿子给我莫大的精神支持。

代码下载

本书案例代码均在 Unity 2018.2 下调试通过，案例源代码及素材文件请扫描下列二维码获取。若下载有问题，请发送电子邮件到 booksaga@126.com，邮件主题为“Unity 3D\2D 手机游戏开发从学习到产品”。



金玺曾
2019年1月

目 录

第 1 章 Unity 基础	1	第 2 章 太空射击游戏	23
1.1 初识游戏引擎和 Unity.....	2	2.1 游戏介绍.....	24
1.2 运行 Unity.....	3	2.1.1 游戏操作.....	24
1.2.1 Unity 的版本.....	3	2.1.2 主角和敌人.....	24
1.2.2 下载并安装 Unity.....	4	2.1.3 游戏 UI.....	24
1.2.3 在线激活 Unity.....	4	2.2 导入美术资源.....	24
1.2.4 运行示例工程.....	4	2.3 创建场景.....	25
1.3 创建一个 Hello World 程序.....	7	2.3.1 创建火星背景和星空动画.....	26
1.3.1 安装 Visual Studio.....	8	2.3.2 设置摄像机和灯光.....	30
1.3.2 编写脚本.....	8	2.4 创建主角.....	31
1.3.3 编译输出.....	10	2.4.1 创建脚本.....	32
1.4 调试程序.....	11	2.4.2 控制飞船移动.....	33
1.4.1 显示 Log.....	11	2.4.3 创建子弹.....	35
1.4.2 在 Visual Studio 中设置断点.....	11	2.4.4 创建子弹 Prefab.....	36
1.5 Unity 脚本基础.....	12	2.4.5 发射子弹.....	37
1.5.1 Script (脚本) 组件.....	12	2.5 创建敌人.....	38
1.5.2 脚本的执行顺序.....	13	2.6 物理碰撞.....	39
1.5.3 脚本的序列化.....	14	2.6.1 添加碰撞体.....	39
1.5.4 组件式编程.....	15	2.6.2 触发碰撞.....	40
1.5.5 协程编程.....	16	2.7 高级敌人.....	43
1.6 预置文件 Prefab 和资源管理.....	17	2.7.1 创建敌人.....	43
1.6.1 创建 Prefab.....	17	2.7.2 发射子弹.....	44
1.6.2 Unity 资源包.....	18	2.8 声音与特效.....	46
1.6.3 管理 Unity 插件.....	18	2.9 敌人生成器.....	48
1.7 读取资源和实例化.....	19	2.10 游戏 UI 和战斗管理.....	50
1.7.1 在脚本中创建一个 Game Object.....	19	2.10.1 创建显示得分的 UI 界面.....	50
1.7.2 通过序列化方式引用资源.....	19	2.10.2 创建显示游戏失败的 UI 界面.....	51
1.7.3 通过 IO 方式读取资源.....	20	2.10.3 编写脚本.....	52
1.8 保存工程.....	20	2.11 关卡跳转.....	54
1.9 时间和动画.....	21	2.12 用鼠标控制主角.....	56
1.10 小结.....	21	2.13 精确的碰撞检测.....	58

2.14	自动创建 Prefab	60	4.2.1	“格子”数据	96
2.15	发布游戏	62	4.2.2	在 Inspector 窗口添加自定义 UI 控件	98
2.16	使用 PoolManager 创建对象池	64	4.2.3	创建一个自定义窗口	100
2.17	小结	66	4.3	游戏场景	102
第 3 章	第一人称射击游戏	68	4.4	制作 UI	103
3.1	策划	69	4.5	创建游戏管理器	105
3.1.1	游戏介绍	69	4.6	摄像机	109
3.1.2	UI 界面	69	4.7	路点	111
3.1.3	主角	69	4.8	敌人	115
3.1.4	敌人	69	4.9	敌人生成器	118
3.2	游戏场景	69	4.9.1	创建敌人生成器	118
3.3	主角	70	4.9.2	遍历敌人	121
3.3.1	角色控制器	70	4.10	防守单位	121
3.3.2	摄像机	72	4.11	生命条	130
3.3.3	武器	73	4.12	小结	132
3.4	敌人	74	第 5 章	2D 游戏	133
3.4.1	寻路	74	5.1	Unity 2D 系统简介	134
3.4.2	设置动画	76	5.2	创建 Sprite	134
3.4.3	行为	77	5.2.1	使用 SpriteEditor 创建 Sprite	134
3.5	UI 界面	80	5.2.2	使用 SpritePacker 创建 Sprite	136
3.6	交互	84	5.2.3	图层排序	137
3.6.1	主角的射击	84	5.2.4	Sprite 边框和重复显示	138
3.6.2	敌人的进攻与死亡	86	5.3	动画制作	138
3.7	出生点	88	5.3.1	序列帧动画	138
3.8	小地图	90	5.3.2	使用脚本实现序列帧动画	140
3.9	小结	93	5.3.3	骨骼动画	141
第 4 章	塔防游戏	94	5.4	2D 物理	142
4.1	策划	95	5.5	捕鱼游戏	146
4.1.1	场景	95	5.5.1	游戏玩法	146
4.1.2	摄像机	95	5.5.2	准备 2D 资源	146
4.1.3	胜负判定	95	5.5.3	创建鱼	148
4.1.4	敌人	95	5.5.4	创建鱼群生成器	150
4.1.5	防守单位	95	5.5.5	创建子弹和大炮	151
4.1.6	UI 界面	95	5.5.6	物理碰撞	153
4.2	地图编辑器	96	5.6	2D 材质	154

5.6.1	修改 Sprite 颜色	154	7.3.3	核心 TCP 网络功能	192
5.6.2	自定义的黑白效果材质	155	7.3.4	创建聊天协议	196
5.7	小结	155	7.4	Unity 聊天客户端	197
第 6 章	与 Web 服务器的交互	156	7.5	聊天服务器端	200
6.1	Web 服务器简介	157	7.6	JSON.NET 简介	205
6.2	在 Windows 上安装部署 Apache	158	7.7	小结	206
6.2.1	安装 Apache	158	第 8 章	HTML5 (WebGL) 游戏移植	207
6.2.2	Apache 常用命令	159	8.1	关于 HTML5 和 WebGL	208
6.2.3	安装 MySQL	159	8.2	导出 Unity 游戏到 WebGL 平台	208
6.2.4	安装 PHP	161	8.2.1	导出 WebGL 游戏	208
6.2.5	安装 Redis	162	8.2.2	设置 WebGL 模板	210
6.3	PHP 开发环境	163	8.2.3	默认 index.html	210
6.3.1	第一个 PHP 程序	163	8.2.4	文件访问	211
6.3.2	调试 PHP 代码	164	8.3	自定义 Loading 页面	212
6.3.3	PHP 基本语法	165	8.4	编写 WebGL 游戏插件	215
6.4	WWW 基本应用	167	8.4.1	访问 JavaScript 示例一	215
6.4.1	创建简单的 UI 界面	167	8.4.2	访问 JavaScript 示例二	216
6.4.2	GET 请求	168	8.5	在网页上保存游戏记录	217
6.4.3	POST 请求	169	8.6	小结	217
6.4.4	上传下载图片	170	第 9 章	iOS 游戏移植	218
6.4.5	下载声音文件	171	9.1	iOS 简介	219
6.5	分数排行榜	172	9.2	软件安装	219
6.5.1	创建数据库	172	9.3	申请开发权限	219
6.5.2	创建 PHP 脚本	173	9.4	设置 iOS 开发环境	219
6.5.3	上传下载分数	175	9.5	测试 iOS 游戏	222
6.5.4	使用 Redis 缓存数据	177	9.6	发布 iOS 游戏	223
6.6	MD5 验证	179	9.7	对接 iOS 原生语言	223
6.7	小结	180	9.7.1	在 Xcode 中创建 Objective-C 代码	223
第 7 章	TCP UDP 实例	181	9.7.2	在 Unity 中引用 Objective-C 代码	226
7.1	网络开发简介	182	9.8	内消费	227
7.2	简单的网络通信程序	184	9.9	本地存储位置	227
7.2.1	简单的 TCP 程序	184	9.10	使用命令行编译 Xcode 工程	228
7.2.2	简单的 UDP 程序	186	9.10.1	编译 Xcode 工程	228
7.3	异步 TCP 网络通信	188			
7.3.1	创建数据包对象	188			
7.3.2	逻辑处理	190			

9.10.2 打包.ipa 文件	228	10.13.1 创建 AssetBundle 资源	261
9.11 小结	228	10.13.2 下载、实例化 AssetBundle 资源	263
第 10 章 Android 游戏移植	229	10.13.3 批量创建 AssetBundle	264
10.1 Android 简介	230	10.14 小结	265
10.2 安装 Android SDK	230	第 11 章 Unity 新 GUI 完全攻略	266
10.3 运行 Android 游戏	231	11.1 Unity 的 GUI 系统	267
10.3.1 准备测试环境	231	11.2 Canvas (画布)	267
10.3.2 设置 Android 游戏工程	232	11.2.1 创建 Canvas	268
10.3.3 测试 Android 游戏	234	11.2.2 设置 Canvas	268
10.3.4 发布 Android 游戏	234	11.2.3 Canvas 的屏幕适应模式	271
10.3.5 Obb 数据包	236	11.2.4 Canvas 层级内 UI 控件的排序	271
10.4 使用 Android Studio	236	11.2.5 Canvas 的隐藏和显示	272
10.4.1 Android Studio 简介	237	11.2.6 查找 UI 层级下的子控件	272
10.4.2 配置 Android Studio	237	11.3 UI 坐标对齐方式	272
10.4.3 创建 Hello world 程序	238	11.4 Text (文字)	273
10.5 从 Unity 到 Android Studio	239	11.5 自定义字体	274
10.5.1 创建 Unity 工程	240	11.5.1 创建字体贴图	274
10.5.2 创建 Android Studio 工程	241	11.5.2 创建 Shader	276
10.5.3 查看 Log	243	11.6 Image (图像)	278
10.5.4 发布程序	244	11.6.1 创建 Image 控件	278
10.6 使用脚本编译游戏	245	11.6.2 设置 Alpha	279
10.6.1 在 Unity 中使用脚本导出 Android 工程	245	11.6.3 设置 Raycast Target	279
10.6.2 使用 Python 脚本访问 Unity 命令行	247	11.6.4 设置 UI 图像为 Sliced 类型	280
10.6.3 使用命令行编译 Android Studio 工程	248	11.6.5 设置 UI 图像为 Tiled 类型	281
10.7 获得签名证书的 sha1 值	249	11.6.6 设置 UI 图像为 Filled 类型	281
10.8 导入库文件	250	11.6.7 Mask (蒙版)	282
10.9 Plugins 目录	251	11.7 Button (按钮) 控件	283
10.10 代码混淆	251	11.7.1 创建按钮	283
10.11 百度地图实例	252	11.7.2 设置按钮状态变化	283
10.11.1 导入百度地图 SDK	252	11.7.3 在编辑器中设置按钮触发事件	284
10.11.2 实现百度地图控件	253	11.7.4 在脚本中定义按钮单击事件	285
10.12 触屏操作	256	11.7.5 判断按压按钮事件	286
10.13 AssetBundle	261	11.8 Toggle (开关) 控件	286
		11.8.1 创建 Toggle 控件	286

11.8.2	Toggle 组	287	12.3.6	Fog (雾)	313
11.8.3	Toggle 控件的脚本	288	12.3.7	直接照明与间接照明	314
11.9	Raw Image 控件	288	12.3.8	Lightmapping (光照贴图)	316
11.10	Slider (滑块) 控件	290	12.3.9	环境反射采样	321
11.10.1	创建 Slider 控件	290	12.3.10	光照采样	322
11.10.2	改变滑块大小	290	12.4	色彩空间	324
11.10.3	使用脚本控制滑块	291	12.5	物理材质	326
11.11	Input Field (文本输入) 控件	291	12.6	摄像机	329
11.12	Scroll View (卷轴视图) 控件	292	12.7	地形	331
11.13	Dropdown (下拉列表) 控件	294	12.8	粒子特效	334
11.14	实用 UI 功能	295	12.9	物理设置	339
11.14.1	判断是否单击了 UI	295	12.10	游戏资源	341
11.14.2	拖动 UI 控件方法 1	295	12.10.1	贴图	341
11.14.3	拖动 UI 控件方法 2	297	12.10.2	在 3ds Max 中创建 法线贴图	342
11.14.4	在 UI 前面显示粒子特效	298	12.10.3	3ds Max 静态模型导出	343
11.15	使用插件 DOTween 制作动画	299	12.10.4	3ds Max 动画导出	345
11.15.1	在编辑器中设置 DOTween 动画	299	12.10.5	Maya 模型导出	345
11.15.2	在脚本中设置 DOTween 动画	300	12.11	Unity 两足动画系统	346
11.16	使用插件 EnhancedScroller 优化 卷轴视图	301	12.11.1	修改 3ds Max 两足动画 骨骼	347
11.16.1	创建数据模型	302	12.11.2	导入 3ds Max 两足角色到 Unity	348
11.16.2	创建 UI 视图控制脚本	302	12.11.3	导入角色动画到 Unity	348
11.16.3	创建控制器	302	12.11.4	表情动画	349
11.16.4	创建卷轴视图 UI	303	12.11.5	动画控制器	350
11.17	小结	305	12.12	动画插件 Skele	351
第 12 章	游戏开发中的美术工作 流程	306	12.13	电影片段制作	355
12.1	Unity 和艺术家	307	12.14	美术资源的优化	357
12.2	视图操作捕捉	307	12.15	小结	358
12.3	光照和渲染系统	308	第 13 章	Behavior Designer——行为 树 AI	359
12.3.1	光源类型	308	13.1	行为树和 AI	360
12.3.2	渲染管道	309	13.1.1	行为树插件的安装	360
12.3.3	质量设置	310	13.1.2	简单的 AI 处理方式	360
12.3.4	实时阴影	310	13.1.3	状态机 AI 处理方式	361
12.3.5	环境光	312			

13.1.4 Behavior Designer 的行为树 AI 处理方式	362	15.2 设置 HTC Vive 开发环境	393
13.2 行为树任务	363	15.3 VR 操作示例	394
13.2.1 任务返回值	363	15.3.1 创建 VR 场景	394
13.2.2 Action (动作)	363	15.3.2 手柄控制器	395
13.2.3 Composite (组合)	363	15.4 小结	397
13.2.4 Conditional (条件)	365	第 16 章 AR 增强现实开发	398
13.2.5 Decorator (修饰)	366	16.1 AR 和 Vuforia	399
13.3 行为树实例	366	16.2 创建 Vuforia 工程	399
13.3.1 主角的行为树	367	16.3 准备 AR 参考目标	401
13.3.2 游荡的敌人	372	16.4 设置 AR 参考目标	402
13.3.3 行为树的交互	373	16.5 小结	404
13.4 行为树系统扩展	376	第 17 章 Shader 编程与后期特效	405
13.4.1 外部行为树	376	17.1 Shader 简介	406
13.4.2 自定义任务	377	17.2 渲染管线	406
13.4.3 自定义共享数据类型	378	17.3 ShaderLab 语言	407
13.4.4 在脚本中设置行为树	380	17.4 CG 语言简介	409
13.4.5 在脚本中发送事件	380	17.5 Vertex and Fragment Shader 示例	409
13.5 优化	380	17.5.1 显示纹理和色彩	409
13.6 小结	381	17.5.2 使用通道表现多个纹理	412
第 14 章 玩转 PlayMaker	382	17.5.3 添加基本光照	414
14.1 关于 PlayMaker	383	17.5.4 UV 动画效果	415
14.1.1 优点和缺点	383	17.5.5 黑白效果	416
14.1.2 安装	383	17.5.6 简单描边效果	417
14.2 PlayMaker 的模块和工作机制	384	17.5.7 Alpha 透明效果	419
14.2.1 有限状态机	384	17.6 屏幕特效示例	421
14.2.2 创建 PlayMaker 状态机	384	17.7 后期特效	423
14.2.3 State (状态)	385	17.8 小结	424
14.2.4 Event (事件)	385	第 18 章 动态编程和 Lua	425
14.2.5 Action (动作)	386	18.1 动态语言简介	426
14.2.6 变量	387	18.2 动态读取程序集实现热更新	426
14.2.7 发送事件	388	18.2.1 创建.dll 文件	426
14.3 自定义 Action	388	18.2.2 测试热更新.dll 文件	427
14.4 小结	391	18.2.3 通过反射访问.dll 文件	428
第 15 章 VR 虚拟现实开发	392	18.3 Lua 语言快速教程	429
15.1 VR 简介	393		

18.3.1	MoonSharp 和 Lua 开发环境的安装	429	A.1.18	预处理	448
18.3.2	Lua 提供的类型	429	A.2	面向对象编程	449
18.3.3	Lua 的判断和循环语句	430	A.2.1	类	449
18.3.4	Lua 的函数	431	A.2.2	this 关键字	450
18.3.5	Lua 的 IO 操作	431	A.2.3	封装	451
18.3.6	Lua 的数组操作	432	A.2.4	继承与多态	452
18.3.7	Lua 的字符串操作	433	A.2.5	静态成员	454
18.3.8	Lua 的协程	433	A.2.6	异常处理	455
18.4	Lua 与 Unity 的交互	435	A.3	字符串	456
18.4.1	在 Unity 中读取 Lua 代码	435	A.4	数组	458
18.4.2	将 Unity 功能导入到 Lua 中	436	A.4.1	创建数组	458
18.5	Lua 脚本的热更新	437	A.4.2	遍历数组和迭代器	458
18.6	小结	438	A.5	基本 I/O 操作	460
附录 A	C#语言	439	A.5.1	写文件	460
A.1	C#基础	440	A.5.2	读文件	461
A.1.1	C#简介	440	A.5.3	删除文件	462
A.1.2	运行控制台程序	440	A.5.4	二进制读写	462
A.1.3	类型	441	A.6	泛型	463
A.1.4	内置类型	441	A.6.1	实现泛型的基本语法	463
A.1.5	标识符	442	A.6.2	List 的常见操作	464
A.1.6	语句和表达式	442	A.6.3	List 的排序	465
A.1.7	变量和常量	443	A.7	委托	468
A.1.8	枚举	443	A.7.1	委托	468
A.1.9	数学操作符	443	A.7.2	Lambda 表达式	469
A.1.10	关系操作符	444	A.7.3	泛型委托	470
A.1.11	逻辑操作符	444	A.7.4	Action 和 Func	470
A.1.12	操作符优先级	445	A.8	反射和特性	471
A.1.13	方法	445	A.8.1	反射	471
A.1.14	递归	446	A.8.2	特性	471
A.1.15	条件分支语句	446	A.9	小结	472
A.1.16	循环语句	447	附录 B	特殊文件夹	473
A.1.17	三元操作符	448	附录 C	Unity 编辑器菜单中英文对照	475



第 1 章

Unity 基础

本章主要介绍什么是 Unity，Unity 的安装及其基本使用，编写最简单的脚本，Unity 的功能特点等内容。

1.1 初识游戏引擎和 Unity

随着计算机软硬件技术的发展，对游戏画面和音效的要求越来越高，开发难度也变得越来越大，一些实力雄厚的公司将自己的技术商业化，作为游戏引擎供其他开发者使用，使开发者可以很大程度地忽略底层技术的复杂性，集中精力在游戏的逻辑和设计上，从而提高生产效率。

一些比较知名的商业化游戏引擎包括 Unreal、CryEngine、Quake、Source、Renderware、Game Byro、Torque Game Engine、Ogre 3D（仅是一个图形引擎）等，这些引擎都曾经非常活跃，有些也很昂贵。随着市场的变化，一些缺乏竞争力的引擎已经逐渐退出了历史舞台，有兴趣的读者可以通过网络了解一下这些引擎的历史，几乎也是一部 3D 游戏发展史。

Unity（也称 Unity 3D）是一套包括图形、声音、物理等功能的游戏引擎，提供了一个强大的图形界面编辑器，支持大部分主流 3D 软件格式，对 2D 游戏也有全面的支持，主要使用 C# 语言进行开发，使开发者无须了解底层复杂的技术，即可快速开发出高性能、高品质的游戏产品。实际上，在游戏引擎的家族中，Unity 是“后起之秀”，但其发展迅速，目前已经成为世界上最活跃的游戏引擎。

在 Unity 的早期版本，使用 Unity 开发的知名游戏仅限于一些手机平台上的休闲游戏，如《Battle heart》等，随着 Unity 的不断升级和普及，现如今很多国内外的游戏大作都是由 Unity 开发的，比如暴雪公司的《炉石传说》和国内的《王者荣耀》等。

Unity 是跨平台的游戏引擎，支持包括 Windows、Mac、Linux、Web、iOS、Android、Windows Phone、Xbox、Play Station 等大部分主流游戏发布平台，还包括各种 VR（虚拟现实）平台，如图 1-1 所示。

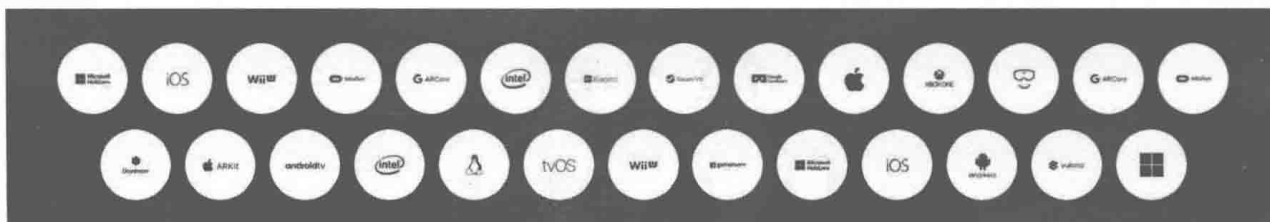


图 1-1 Unity 支持的主流平台

在 Unity 早期的版本，可以将游戏导出为 Flash 或 Unity 自己的网页格式放到网页上，但随着 HTML5 的发展，Unity 在网页游戏领域已经主要转移到 Web GL 平台上。笔者曾经开发的一些游戏，除了在移动平台上发布，也发布到了网页游戏平台 KONGREGATE 上，有兴趣的读者，可访问 <http://www.kongregate.com/>，然后搜索游戏的英文名 Wild Defense，就可以玩到笔者过去完成的一个塔防游戏，如图 1-2 所示。

Unity 的主要开发环境是在 Windows 或 Mac 上面，因为在 Windows 上开发有很多优势，可以在 Windows 平台开发和测试，然后将游戏移植到其他平台。本书中的大部分示例是在 Windows 上完成的。

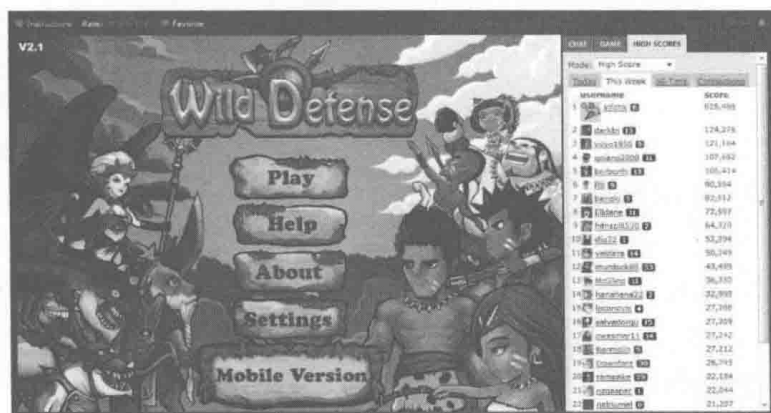


图 1-2 网页版野人大作战

在开始使用 Unity 之前不得不提一下 Asset Store (<https://www.assetstore.unity3d.com/>)，它是 Unity 官方的在线商店，如图 1-3 所示。Asset Store 里面主要出售 Unity 的插件或美术资源，它已经成为 Unity 的一个重要组成部分，很多插件在 Unity 开发中已经是必不可少的，在本书的示例中，也将使用到很多插件和美术资源，有一部分是免费的。

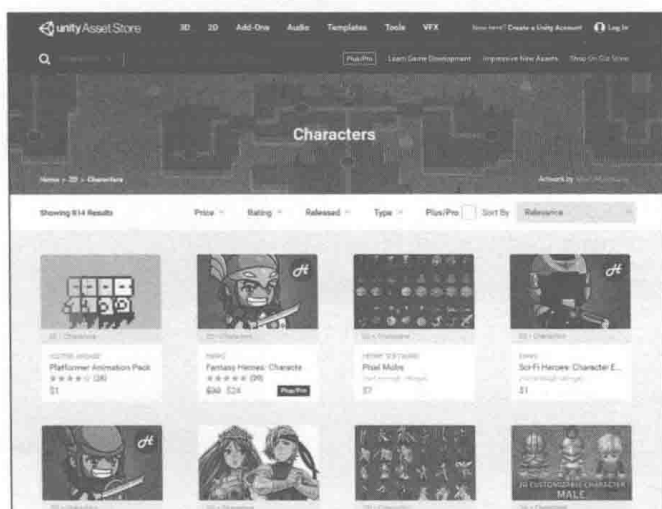


图 1-3 Asset Store 的页面截屏

1.2 运行 Unity

本节主要介绍如何安装和运行 Unity 示例工程以及安装 Visual Studio，创建一个 Unity 工程。

1.2.1 Unity 的版本

Unity 提供了专业版和个人版，个人版是免费的。在功能上，不同版本并没有太大区别，专业版会提供一些额外的云端服务，不过如果公司的收入超过一定额度，则必须购买专业版。对于大部分学生或个人开发者来说，选择个人版即可，使用个人版同样可以发布商业化的游戏，这一点并不受版本的限制。

1.2.2 下载并安装 Unity

在 Unity 的官方网站 <https://store.unity.com/> 中可以免费下载 PC 版和 Mac 版的 Unity，这是完整的安装包，包括针对所有平台的全部功能。下载 Unity 后，运行安装程序，可以仅选择需要的内容安装，其中 Unity 的编辑器是必须安装的组件，Microsoft Visual Studio 是推荐使用的 C# 编辑器。

有一点需要注意，通过 Unity 下载主页面链接下载的版本为 Unity 的“主要版本”，页面上还有一个链接可以下载“补丁版本”（Patch releases），补丁版本通常是在主要版本的基础上修复了各种 Bug，在下载之前，建议阅读版本发布说明，选择适合自己的版本。

1.2.3 在线激活 Unity

第一次运行 Unity 会提示选择版本，如果没有购买过专业版，选择个人版（Personal）即可，但无论选择哪个版本，都需要注册一个 Unity 账号进行登录，这个账号非常有用，除了用来登录 Unity，也可以用来在 Asset Store 中购买或下载资源，同时还可以使用这个账号在 Asset Store 中销售自己开发的插件或美术素材供别人使用。

1.2.4 运行示例工程

启动 Unity，打开 Unity 的工程对话框，选择【New】新建一个工程，输入工程名称和保存路径，如图 1-4 所示。标准的 Unity 工程主要包括几个部分：Assets 文件夹内包括所有的工程文件，这里是主要工作的地方；Library 文件夹内是工程的数据库文件（可以删除，重新打开工程后会自动重建）；ProjectSettings 文件夹内保存工程的配置文件。

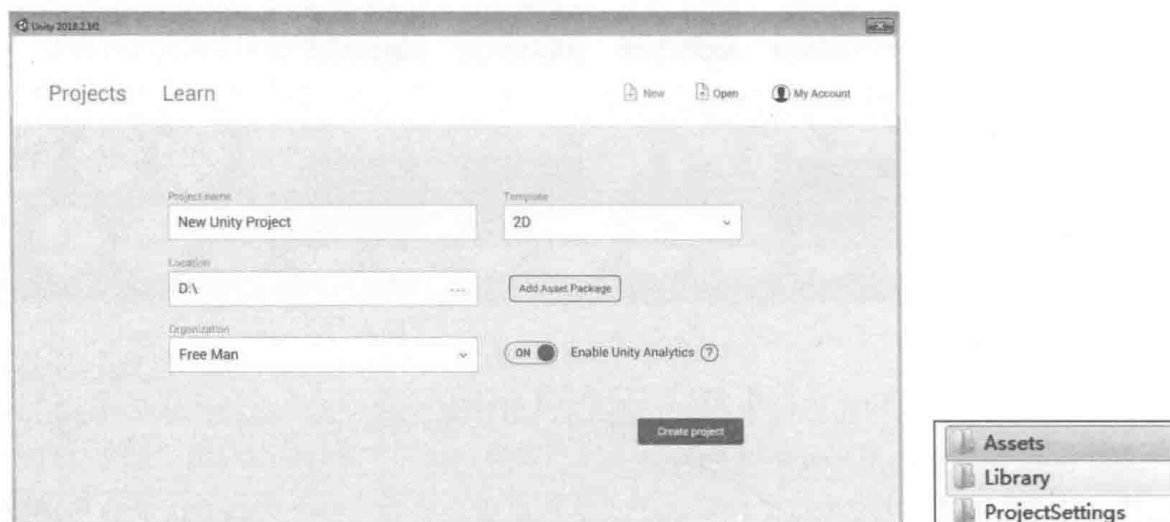


图 1-4 创建新工程

启动 Unity 编辑器后，我们先来尝试下 Unity 的 Asset Store，在菜单栏中选择【Window】→【General】→【Asset Store】打开 Asset Store 窗口（也可以通过网址 <https://assetstore.unity.com> 直接访问），选择 Unity Essentials，如图 1-5 所示，这里的资源都是 Unity 官方免费提供的，有游戏示例、教程、美术或音效等各种资源，方便我们学习和开发。在本书的教程中，也将经常使用到这里的资源。

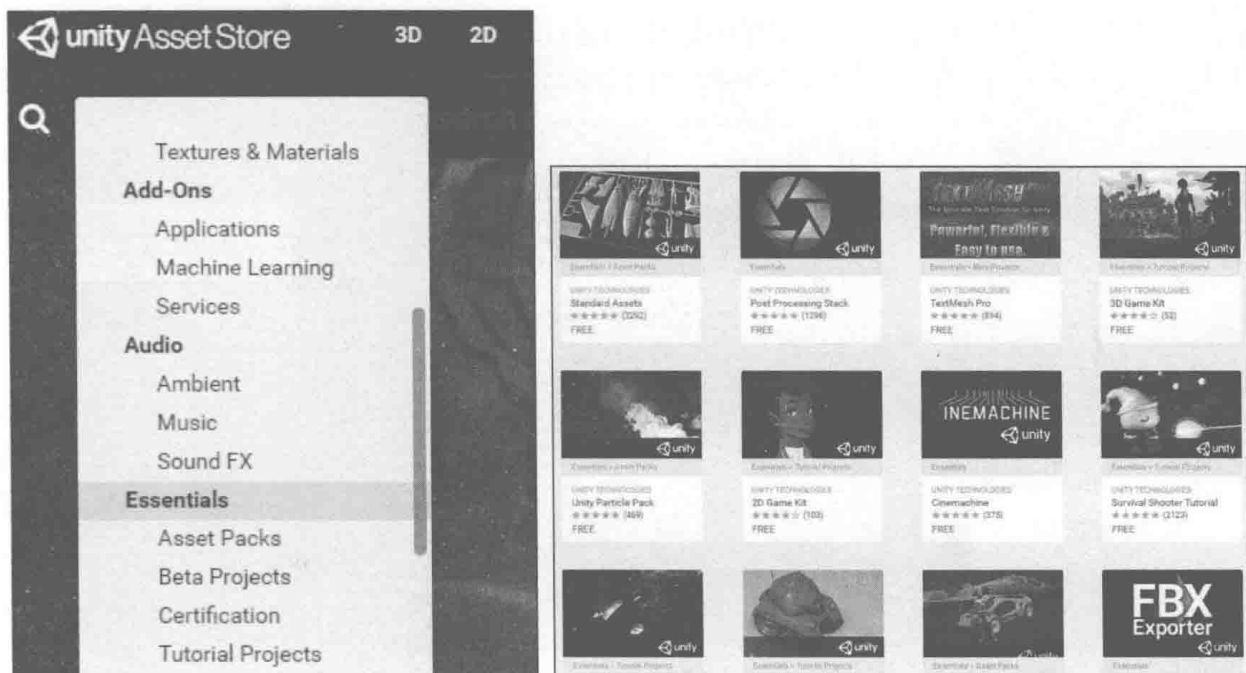


图 1-5 Unity 官方提供的免费资源

我们在这里选择下载 Survival Shooter，如图 1-6 所示，这是一个射击类游戏的示例工程。

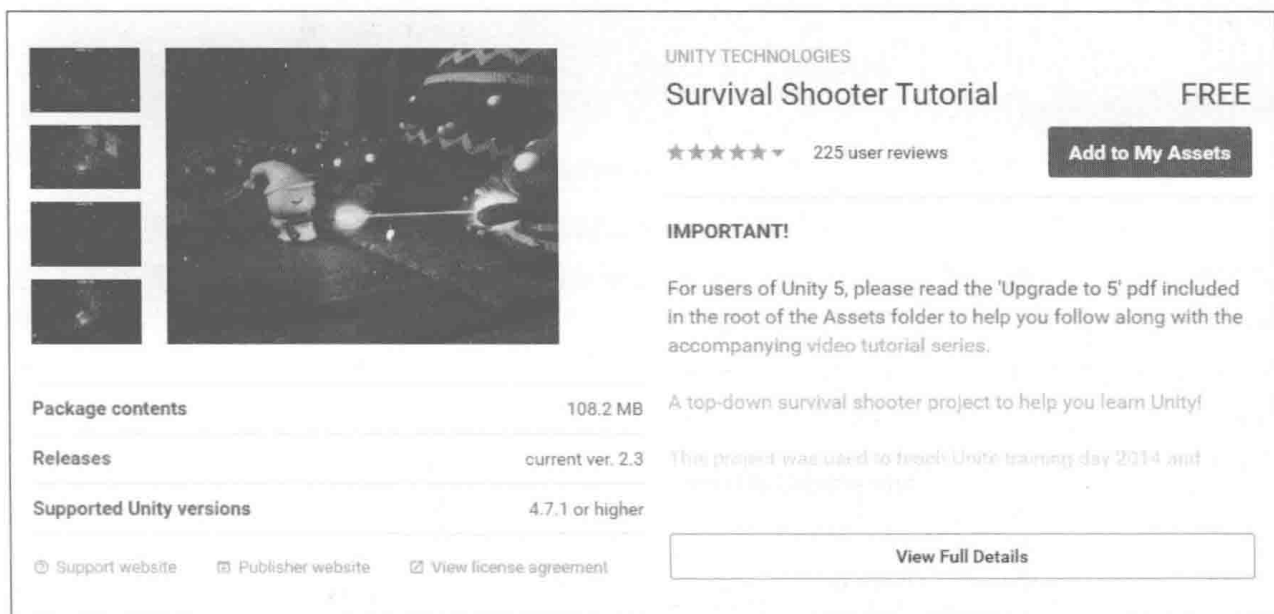


图 1-6 免费的射击游戏示例

下载完成后，选择 Import 将游戏示例导入到当前 Unity 工程中。在编辑器的 Project 窗口中浏览到 Assets/_CompletedAssets/Scenes 目录，双击后缀名为.unity 的场景文件打开场景，然后单击编辑器上方的播放按钮，即可运行游戏，如果再次单击播放按钮则会退出游戏，如图 1-7 所示。