Unity3D PlayMaker 游戏设计与实现

周 頔 孙辛欣 盛歆漪 著

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Unity3D PlayMaker
游戏设计与实现

周頓　孙辛欣　盛歆漪　著

電子工業出版社
Publishing House of Electronics Industry
北京·BEIJING

内 容 简 介

与传统软件操作类图书不同的是，本书并没有以PlayMaker的各种组件为顺序进行介绍，而是从如何制作一个完整电子游戏的角度出发，详细介绍了如何使用PlayMaker在Unity3D的环境中设计并开发游戏的各个重要组成部分，包括制作游戏的玩家控制角色，非玩家控制角色，地形、天空、关卡、声音，以及图形用户界面等。本书内容深入浅出，层层递进。通过对本书内容的学习，读者就可以制作出完整的3D游戏。

全书分为7章。第1章主要介绍了游戏设计与开发、Unity游戏引擎、PlayMaker可视化编程插件等相关知识。第2章介绍了Unity的获取与使用、Unity的基本操作、PlayMaker的获取与导入、PlayMaker的基本操作。第3章至第7章介绍了如何使用PlayMaker和Unity3D开发游戏中的玩家控制角色、战斗型非玩家控制角色、服务型非玩家控制角色、游戏环境的设计、图形用户界面设计以及如何发布游戏。本书附有配套的数字资源，包括书中案例的完整资料，方便读者对照学习。

本书适合有志于独立游戏开发的艺术设计师、游戏设计人员，以及所有对游戏开发感兴趣的读者参考。

图书在版编目（CIP）数据

Unity3D PlayMaker游戏设计与实现 / 周頔, 孙辛欣, 盛歆漪著. — 北京 : 电子工业出版社, 2019.4

ISBN 978-7-121-35561-5

Ⅰ. ①U… Ⅱ. ①周… ②孙… ③盛… 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260306号

责任编辑：赵玉山
印　　刷：天津嘉恒印务有限公司
装　　订：天津嘉恒印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路173信箱　　邮编：100036
开　　本：787×1092　1/16　印张：14.25　字数：462千字
版　　次：2019年4月第1版
印　　次：2019年4月第1次印刷
定　　价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

本书咨询联系方式：（010）88254556，zhaoys@phei.com.cn。

前 言

为什么要写这样一本书

电子游戏的设计与开发，是个既需要艺术，也需要技术的复杂工作。过去，游戏开发领域基本都以程序员为主导，而负责游戏中各种艺术设计工作的模型设计师、UI 设计师、动画设计师等基本都处于辅助地位。随着《纪念碑谷》这类游戏的出现，艺术设计在游戏设计与开发中的作用得到了显著提高，而且很多艺术背景的设计师也开始转做独立游戏开发。借助 Unity3D 这类引擎，游戏开发的难度已经大大降低。但是阻碍艺术背景的设计师独立开发游戏的，仍旧是如何用程序把设计好的各种游戏素材组装起来。Unity3D 中的代码基本都是用 C# 来完成的，要想在短时间内掌握这门编程语言，并且流畅地编写游戏脚本并不是一件简单的事情。而且对于程序员来讲，在开发中也经常会遇到快速制作出游戏原型，或者快速实现某种游戏功能的需求。为了让更多的人能参与到游戏开发中，更方便快捷地开发电子游戏，出现了一些可视化编程工具。而 PlayMaker 无疑是其中最受欢迎的一款。

虽然 PlayMaker 得到越来越多游戏开发人员的关注，甚至像《炉石传说》、《INSIDE》这类经典游戏的开发中都有 PlayMaker 的参与，但遗憾的是，国内缺乏关于 PlayMaker 的图书。为了方便大家学习如何使用 PlayMaker 高效地制作游戏，特此推出这样一本既适合设计师、也适合程序员阅读的专业书。

本书特点

版本最新

本书以最新版本的 Unity3D 和 PlayMaker 为对象进行讲解，系统介绍了如何在 Unity3D 环境中使用 PlayMaker 设计开发游戏，内容新颖。

实用的章节安排，面向实战

与传统软件操作类书不同的是，本书并没有以 PlayMaker 的各种组件为顺序进行介绍，而是从如何制作一个完整电子游戏的角度出发，详细介绍了如何使用 PlayMaker 在 Unity3D 的环境中设计并开发游戏的各个重要组成部分，包括制作游戏的玩家控制角色，非玩家控制角色，地形、天空、关卡、声音，以及图形用户界面等。顺序学完本书内容，即可制作出完整的 3D 游戏，而且对 PlayMaker 也有了全面的理解，非常实用。

左右双栏

本书所有章节的排版都采用左右双栏的布局，包括正文栏和注释栏。建议将左右栏结合在一起阅读。

既适合设计师，也适合程序员

无论是有志于独立游戏开发的艺术设计师，还是需要快速制作游戏原型和功能模块的程序员，本书对他们都有自己独特的价值。本书既可以作为教材，也可以作为对游戏开发感兴趣读者的参考书。

实用的配套资源

本书示例中用到的素材，既有从 Asset Store 中下载的免费资源，也有专门为此制作的模型、脚本等。所有这些书中用到的素材，以及完整的游戏项目，都在本书配套的数字资源中有提供，以方便读者对照学习。

内容导读

本书分为 7 章，以制作游戏中的不同组成部分为顺序，进行了系统性的讲解。

第 1 章： 主要介绍了游戏设计与开发中的相关内容，为后续章节使用 Unity3D 和 PlayMaker 进行游戏开发做好准备。具体包括：游戏与电子游戏的概念、电子游戏的分类、游戏设计与开发的过程、游戏引擎的概念、Unity3D 简介、PlayMaker 简介、使用 Unity3D 和 PlayMaker 开发的游戏简介。

第 2 章： 介绍 Unity3D 和 PlayMaker 的获取及安装，并通过实例介绍它们各自的使用方法，以及使用 PlayMaker 在 Unity3D 中控制游戏对象的方法，包括平铺直叙式 FSM、多 FSM 协同式。

第 3 章： 详细介绍了游戏中玩家控制角色的设计与实现。分析了玩家控制角色一般必须要具备的五种功能：前 / 后移动、转向、攻击、跳跃、收集，并通过多 FSM 协同式实现了这些功能。具体包括如何把 3ds Max 做的模型导入游戏，如何处理动画，如何使用角色控制器，如何对键盘输入做出响应，如何使用 C# 脚本，如何使用 PlayMaker 控制 C# 脚本，如何使用 Tag，如何销毁游戏对象。

第 4 章： 详细介绍了战斗型 NPC 的设计与实现方法，着重分析了战斗型 NPC 的行为逻辑。使用上下两层式结构，完成了战斗型 NPC 的 FSM 构建。介绍了游戏中角色（包括玩家控制角色与非玩家控制角色）生命系统的设计与实现方法。并在此基础上实现了玩家控制角色与非玩家控制角色之间的交互。具体包括如何从 Asset Store 中导入素材，如何使用数组，如何用武器进行攻击，如何徒手进行攻击，如何在一个 FSM 中访问另一个 FSM 中的变量，如何禁用或使用一个 FSM，什么情况下需要设置子物体与父物体，碰撞体 Collider 与刚体 Rigidbody，Is Trigger 以及碰撞检测。

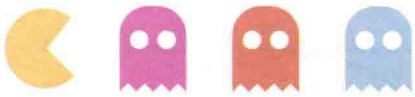
第 5 章： 重点介绍了服务型 NPC 的设计与实现方法。具体包括大多数游戏中角色对话的实现途径，游戏中的视角切换问题，服务型 NPC 的行为逻辑，使用 UGUI 来实现对话框（包括 Canvas、Panel、Text 和 Button 的使用），如何用 PlayMaker 来控制 UGUI，预制件与实例。

第 6 章： 介绍了如何设计与实现游戏世界中的四种重要元素：地形、天空、关卡和声音。具体包括如何创建地形（山脉、河流），Paint Height 工具，Raise/Lower Terrain 工具，Smooth Height 工具，地形的纹理，如何在地形上植树，LOD 技术，Paint Trees 工具，如何在地形上种草，Paint Details 工具，如何制作波动的水面，游戏场景的边界，天空盒技术，关卡的实现，存档点与位置保存，玩家控制角色的死亡与复活机制，PlayMaker 中对预制件与实例的操作，Unity 游戏中播放声音的原理，音频监听器，音源，如何给游戏加背景音乐，如何用 PlayMaker 控制游戏音效的播放，3D 音效的使用。

第 7 章： 主要以 HUD 和主菜单为例，详细介绍了游戏中图形用户界面的设计与制作方法。具体包括 UGUI 的容器与控件，Canvas 的三种渲染模式，如何让 UI 自适应屏幕，如何搭建图形用户界面，如何用 PlayMaker 来控制图形用户界面，如何制作游戏中的血条，如何制作小地图，如何实现不同坐标系之间的变换，如何搭建游戏的主菜单，如何进行场景的切换，如何制作弹出式图形用户界面，如何退出游戏，全局变量和局部变量的概念，如何设置及使用全局变量，如何控制背景音乐的音量，如何发布游戏。

配套资源说明

在本书的配套资源中，既有需要导入游戏项目的素材，也有完整的游戏项目，读者可在华信教育资源网（www.hxedu.com.cn）下载。在正文中相应的位置会提示读者此处该使用哪个配套资源，读者只需按照正文的提示使用即可。



目 录

游戏设计概论

CHAPTER 01

1.1 游戏设计与开发

游戏是连接幻想与现实的通道。游戏产业作为一个新的经济增长点，已经成为新文化创意产业中最有活力的一部分。《2017 年中国游戏行业发展报告》显示，中国游戏市场 2017 年的实际销售收入达到了 20136.1 亿元，同比增长 23.0%，是同期电影票房的四倍。这给广大的游戏设计师和游戏开发者带来了前所未有的机遇。

1.1.1 游戏与电子游戏

所谓游戏，其实是一个很宽泛的概念。无论发生在真实环境中，还是虚拟环境里，所有由参与者按照规则行动，去实现至少一个既定目标任务的娱乐性活动，都可以被称为游戏。一个完整的游戏，在理论上应该包含四种必备的组成要素：游戏目标、游戏规则、可玩性以及假想性。

① **游戏目标：**也就是游戏中预设要完成的任务。这种任务既可以是以竞争为目的的，也可以是以创造为目的的。比如足球，就是一种现实生活中的大型竞争游戏。它把参与人员分成两队，以争取比对方进更多的球为游戏目标。再比如《模拟城市》这样的电子游戏（见图 1.1），它的游戏目标就不是竞争，而是建立并管理城市，不让其毁掉。只要城市不毁，游戏就会无限发展下去。

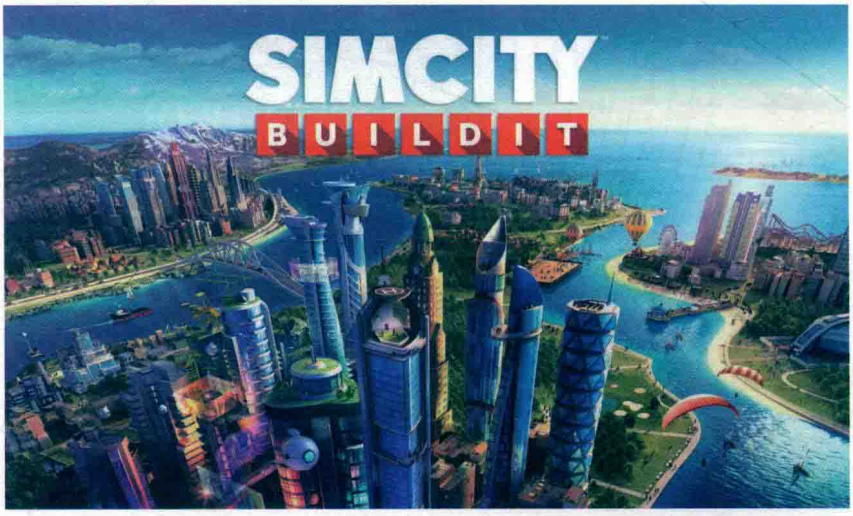


图 1.1 《模拟城市》——以创造为游戏目标

② **游戏规则：**指导和要求我们如何去玩游戏的这些内容，就是游戏规则。对于玩游戏的人来说，有些游戏的规则是清晰的、成文的，而有些游戏，在玩之前其游戏规则是隐藏而不明确的。比如很多电子游戏，在开始时玩家并不确切地知道规则是什么，而是随着游戏的进行，根据游戏给出的反馈才逐步了解游戏的规则到底是什么。

③ **可玩性：**没有可玩性，就谈不上游戏。本质上，游戏的可玩性是由游戏中的各种挑战和动作来体现的。一个可玩性高的游戏，往往既设置有令人感兴趣的挑战，又配备有可以克服这些挑战的动作。在让玩家

选择不同的动作方案去尝试挑战的过程中，让玩家体验到紧张、成功等复杂的感觉。

④ **假想性：**游戏往往会创造一个假想世界。而且事实上，游戏能让玩家产生想象这一特点，正是游戏能吸引玩家的重要原因之一。有时玩家需要想象自己身处一个虚幻的世界中，或者把自己想象成游戏中的某一个角色。这种代入感越强，就能给玩家越好的游戏体验。

作为游戏中的一个子集，电子游戏是指那些通过计算机运行的游戏。从钥匙扣上的电子宠物，到游乐场中的大型电玩游戏，都属于电子游戏的范畴。通过计算机屏幕和扬声器，电子游戏能将游戏中的假想世界更直观逼真地展现给玩家，带来更强烈的感官刺激。现代的电子游戏充满了传统游戏所不具备的图片、动画、光影、声音和对话。近年来，更是出现了增强现实（Augmented Reality, AR）以及虚拟现实（Virtual Reality, VR）类型的电子游戏，将现实世界与假想世界进一步融合，或者直接混淆这两者的边界，让玩家得到更身临其境的奇幻体验。

1.1.2 电子游戏的分类

对于现代的电子游戏来说，经常在同一个游戏中兼具多种特征，所以很难明确地把一个电子游戏划分到某一类中。从内容来看，现代的电子游戏主要有以下几种：

① **射击游戏：**典型的就是Microsoft开发的《Halo》系列（见图1.2）、Valve开发的《Counter-Strike》系列。这类游戏基本采用第一人称或者第三人称视角。玩家使用某种形式的武器，在一定距离以外对游戏中的敌人发起进攻。这类游戏考验的是玩家的反应速度和精确度。

图1.2　射击游戏《Halo》

② **格斗游戏：**这类游戏中较少涉及探索、射击或者解密，而是集中在模拟近身格斗上。这种游戏一般有竞技场模式（也就是玩家之间单对单的比赛）以及混战模式（一到两个玩家合作去挑战一大群敌人），考验的主要是玩家的反应速度和时机，典型的是日本SNK开发的《KOF》，如图1.3所示。

图 1.3　格斗游戏《KOF》

③ **策略游戏：**在这类游戏中，需要玩家依据在游戏中收集到的各种信息，制定出游戏计划，与一个或多个敌人进行对抗。将策略游戏分成回合制和实时制两种。策略游戏经常把减少敌人数量作为游戏的主要目标，所以大部分的策略游戏都或多或少会有一些战争的影子。这类游戏主要考验玩家处理信息、制定计划的能力，典型的是 Blizzard Entertainment 开发的《World of Warcraft》系列，如图 1.4 所示。

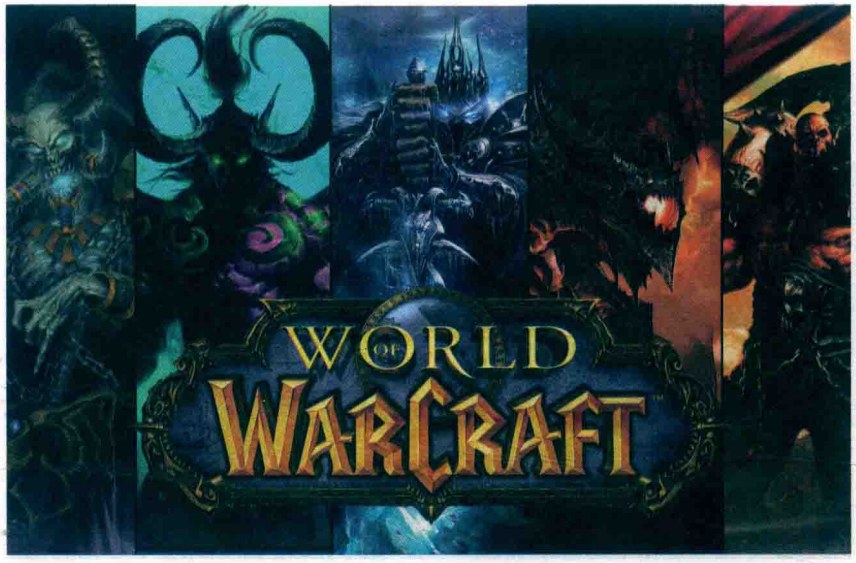


图 1.4　策略游戏《World of Warcraft》

④ **RPG 游戏：**也就是角色扮演类游戏。在这类游戏中，玩家负责扮演主角，在某个假想世界中活动。通过培养自身拥有游戏中的某种技能，让玩家获得从一个普通人到一个超级英雄的成长体验。大多数的 RPG 游戏综合了战术、后勤和探索的挑战，典型的是 Ubisoft Entertainment 开发的《Assassin's Creed》系列，如图 1.5 所示。

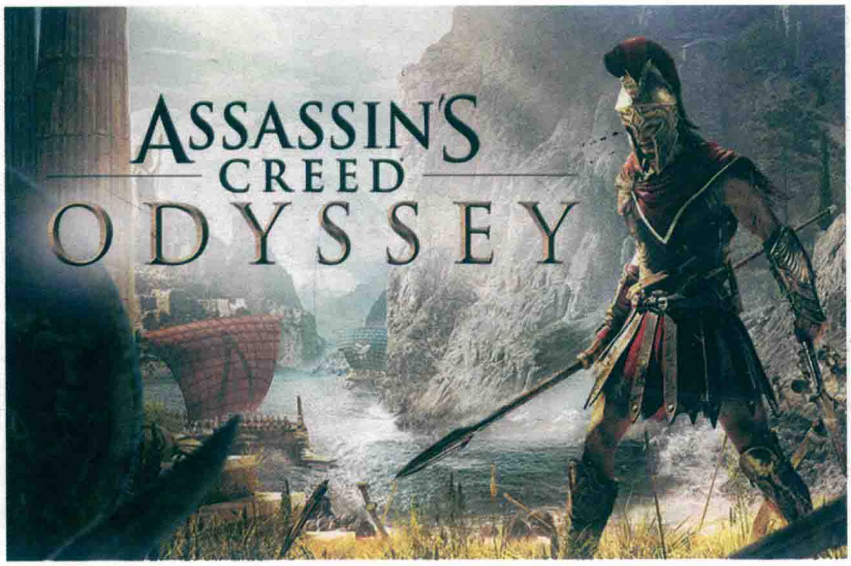


图 1.5　RPG 游戏《Assassin's Creed》

⑤ **体育游戏：**与其他游戏不同的是，体育游戏中的世界是玩家本来就熟悉的。很多体育游戏致力于模拟真正的运动项目，并需要在游戏中管理一个团队或者某一个运动员的运动生涯。这类游戏重点考验玩家对现实世界中某项体育运动规则的熟悉程度、反应力以及规划能力，典型的是 Konami 开发的《Pro Evolution Soccer》系列，如图 1.6 所示。

图 1.6　体育游戏《Pro Evolution Soccer》

⑥ **驾驶游戏：**这类游戏模拟了现实世界中的某种交通工具，旨在提供一种不会让玩家人身受到伤害，但又能体会到各种极限驾驶的逼真体验。所以这类游戏会尽可能让玩家在视觉上跟真正操控一个交通工具相同，典型的是 Microsoft 开发的《Forza Motorsport》系列，如图 1.7 所示。这类游戏考验的主要是玩家的反应速度和驾驶感。

图 1.7　驾驶游戏《Forza Motorsport》

⑦ **解谜游戏：**这类游戏一般体量都比其他类型的游戏要小。在这类游戏中，经常随着游戏的推进会出现难度递增的谜题。谜题的类型包括模式识别、逻辑推理、理解过程、寻找物品等。在游戏中通常还会加入时间压力。典型的是 Zeptolab 开发的《Cut the Rope》系列，如图 1.8 所示。

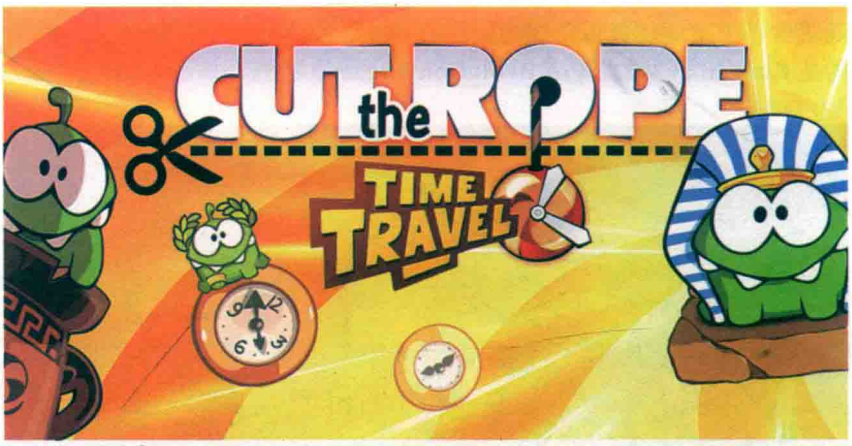


图 1.8　解谜游戏《Cut the Rope》

1.1.3 游戏设计与开发的过程

游戏设计与开发，是一个典型的需要艺术与技术互相结合的工作。对于电子游戏来说，它必须给人视觉上的愉悦，同时还要用各种技术手段把精心设计的游戏元素呈现出来并流畅地运行。所以将两者割裂，单独强调游戏中的艺术，或者技术，是不完整的。

对于游戏的设计来说，我们也可以借鉴产品设计中的理念，以玩家为中心来进行游戏设计（Player-centric Game Design）。设计者把自己想象成玩家，揣摩玩家的心理和喜好，理解玩家想从游戏中获得的到底是娱乐、教育、研究，或者其他什么体验，考虑玩家对游戏中的美感、用户界面、可玩性等方面会有什么样的反应，找到现有同类游戏中的痛点，有针对性地进行游戏设计。

对于游戏开发来说，一般都会经过三个阶段，如图 1.9 所示。

① **概念设计阶段：**在这个阶段，首先需要进行市场调研，确定目标玩家是哪些人。然后要有一个游戏创意，也就是打算如何通过游戏来吸引目标玩家的想法。将这个游戏创意演变成一个具体的故事情节之后，就需要设计一个游戏模式，包括视角、交互方式、挑战方式、单机或联网、表现风格等。然后设计一个或多个角色，包括他们的形象、性格、技能等。并对游戏中的假想世界、关卡、进度和任务进行创意。

概念设计阶段做出的决定，一般在游戏开发的整个过程中都会产生影响。所以该阶段的设计，要特别谨慎。因为一旦出现问题，往往会导致后续开发阶段中的所有结果都要改动，影响面很大。

② **详细设计阶段：**在这个环节中，开发工作将由理论演变为实际。其重点是先做出游戏原型，然后进行原型测试，不断迭代。在艺术部分，需要依照前期策划的视觉风格，将人物原画、建模、材质贴图、人物动作、场景动画、音效等制作出来。而在技术部分，用程序把所有这些素材组合起来，实现策划阶段提出的各种需求，包括搭建游戏主程序、客户端、服务器端、数据库等，并在整个游戏制作完毕之后，进行多轮测试。所以，这个阶段的工作是多次重复、迭代进行的。

图 1.9　游戏设计与开发中的三个阶段

③ **调整阶段：**这个阶段的工作主要是微调游戏的平衡性，也就是让游戏既不能太简单，也不能太难，而且还应该让所有玩家都觉得公平，移除所有可能无意识产生的统治性策略或者困难度突变。目的就是在游戏交付的截止日期之前，将游戏调整至最佳状态。

另外，在游戏的设计与开发过程中还有一个不可忽视的内容，就是运营。游戏的运营通常都是贯穿游戏开发整个过程的，它主要通过一系列的营销手段让市场认可游戏，提高游戏安装数量和在线玩家数量，刺激消费，增加利润，并收集市场反馈，指导游戏的版本迭代。

1.2 Unity 游戏引擎

1.2.1 游戏引擎

在游戏引擎还没有出现的年代里，电子游戏基本都是横版的 2D 游

戏。这些目前看起来很简单粗糙的游戏，当时的开发周期平均达到 8 个月左右。一方面是因为受当时技术与设备的限制，而另一方面却是因为每次开发游戏都需要重新编写代码。每款游戏中涉及底层技术的代码，其实有很多是重复的。所以如果每次开发都重新编写，就会造成开发中大量的重复性劳动。因此后来的开发人员就尝试将一些经常要用到的代码编写到一起，形成一个框架。每次需要开发新游戏时，就在这些现成的框架基础上进行修改和添加，从而节约大量的时间和成本，提高了游戏开发的效率。而这些框架，就是现代游戏引擎的雏形。

因此，所谓游戏引擎，其实就是一种特殊的软件，用来给游戏开发者提供各种开发工具，以便让他们能集中精力于游戏的逻辑和设计上，可以更容易、更快速地制作出游戏，而无须花太多时间关注底层技术。一般来说，一个游戏引擎内部都会包含多个功能模块，常见的有 2D/3D 图像渲染、物理系统、动画系统、音效处理、资源管理、输入 / 输出系统、网络互联等。

① **2D/3D 图像渲染：**主要负责把游戏中的模型、动画、光影、特效等各种视觉效果实时计算出来，并显示在屏幕上。渲染一直都是游戏引擎中最重要的功能之一，其性能直接影响游戏最后的输出质量和运行速度。

② **物理系统：**其实就是一套力学规则，包括物体之间发生碰撞时的力学模拟、物体与自然环境之间的力学模拟、角色骨骼运动的力学模拟等，目的是让游戏中物体的运动可以更加符合现实中的规律，比如碰撞、重力、摩擦力、飞行轨迹的计算等。

③ **动画系统：**包含骨骼动画和模型动画两部分内容。通过动画系统，可以让游戏中的角色拥有更丰富的动作表现。

④ **音效处理：**负责游戏中各种声音的播放，以及不同的播放效果，比如混响、立体、声音的障碍等。

⑤ **资源管理：**负责载入和管理游戏所需的各种资源，包括离线资源管理和运行资源管理。

⑥ **输入 / 输出系统：**主要负责处理玩家与游戏之间的交互，包括来自键盘、鼠标、触屏、摇杆、陀螺仪、手柄等各种设备的信号。

⑦ **网络互联：**主要处理网络发包和延迟、异步通信、同步通信、服务器端软件配置管理、服务器程序的最优化等。

经过多年的竞争和发展，目前主流的游戏开发引擎包括 Unreal Engine、Godot Engine、Cry Engine、Hero Engine、Unity、AppGameKit 等，如图 1.10 所示。其中 Unity 因其学习门槛低、易使用、兼容几乎所有的游戏平台、具有强大的开发者社区，以及最具竞争力的授权条款等优势，近年来在游戏开发领域占据越来越重要的地位。截至 2018 年 5 月的统计数据表明，在移动端游戏领域，全球 50% 的移动游戏是采用 Unity 开发的。这些游戏下载量达到了 240 亿次，运行在 60 亿台独立设备上。在 VR 开发方面，69% 的 Oculus Rift 平台内容、74% 的 HTC Vive 平台内容、87% 的 Gear 平台内容，以及 91% 的 HoloLens 平台内容都是采用 Unity 开发的。