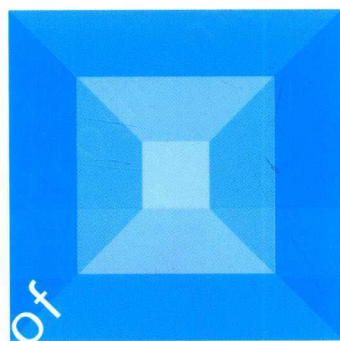
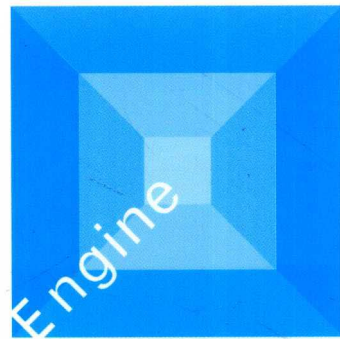
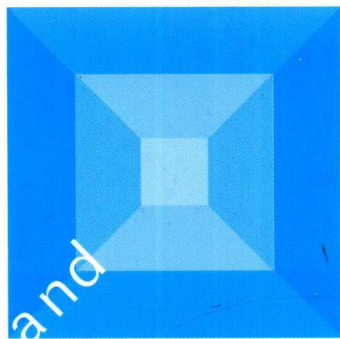
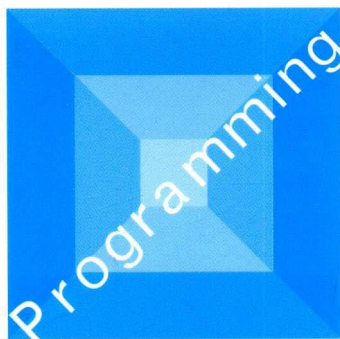




教育部大学计算机课程改革项目 | 计算机艺术设计规划教材

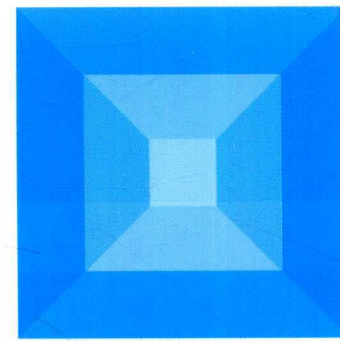
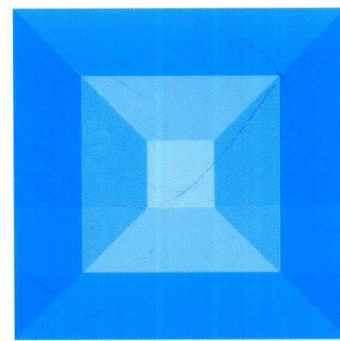
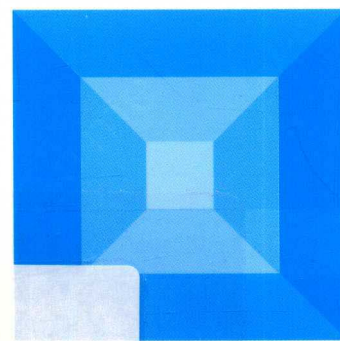
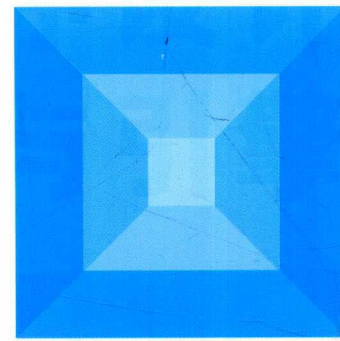
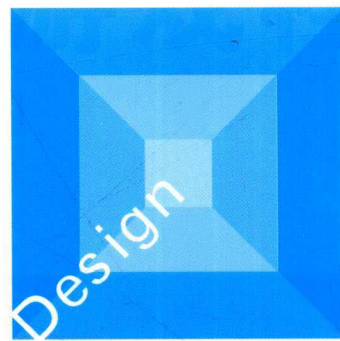
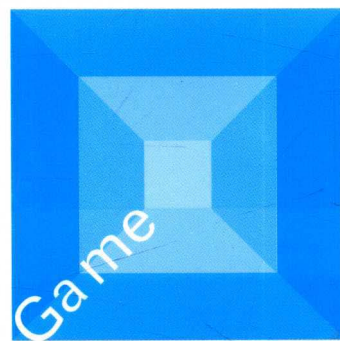
主编：何洁 杨静



南京艺术学院重点教材建设立项

# 游戏程序设计 与引擎

严宝平 编著



清华大学出版社

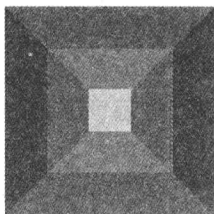


教育部大学计算机课程改革项目  
计算机艺术设计规划教材

主编：何洁 杨静

# 游戏程序设计与引擎

严宝平 编著



清华大学出版社  
北京

## 内 容 简 介

本书面向数字媒体艺术专业、游戏艺术设计专业的本科生及艺术类对游戏设计感兴趣的学生,旨在为其提供游戏开发相关的基础知识与必要技能。全书以4个具有代表性的案例为切入点,前半部分以Flash为工具分析游戏程序设计的必要知识;后半部分以Unity为工具分析游戏引擎的操控要点,与专门针对程序设计的教材不同的是,本教材涉及内容相对广泛,也符合艺术类学生学习游戏开发的特点,重点并非掌握游戏开发的核心技术,而是通过对技术的了解掌握在工具辅助下的开发流程,进而更好地与开发人员配合完成开发项目,再进一步在工具的帮助下独立完成开发项目。全书在上述内容讲解之余,亦投入相当篇幅分析具有前沿应用的增强现实与体感游戏,并就产业特点同样以一章的篇幅分析跨平台游戏开发与项目管理。

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

游戏程序设计与引擎/严宝平编著. —北京:清华大学出版社,2017

(教育部大学计算机课程改革项目·计算机艺术设计规划教材)

ISBN 978-7-302-44562-3

I. ①游… II. ①严… III. ①游戏程序—程序设计—高等学校—教材 IV. ①TP311.5

中国版本图书馆CIP数据核字(2016)第174861号

责任编辑:谢琛 赵晓宁

封面设计:孟梦

责任校对:胡伟民

责任印制:宋林

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)

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:11

字 数:270千字

版 次:2017年2月第1版

印 次:2017年2月第1次印刷

印 数:1~2000

定 价:35.00元

产品编号:066772-01

# 序

《计算机艺术设计规划教材》是在教育部高等学校文科计算机基础教学指导分委员会组织下完成的教育部“大学计算机课程改革项目·计算机艺术设计课程与教材创新研究”的成果,涵盖了大学计算机基础、信息与交互设计、互动媒体艺术、数字游戏设计、计算机网页设计、计算机动画应用与开发等内容。本套教材由清华大学美术学院牵头,国内多所在本领域具有广泛影响力的综合性院校和艺术院校的相关专业教师参与,目的是通过教材改革创新引导学生利用计算思维,发现并善于借助计算机的优势,科学运用计算机技术,培养学生基于计算思维优化创新、应用设计的综合能力,以适应当今时代的发展和需求。

在计算机技术应用目的、层次、范围不断扩延和提升的今天,其学科渗透与产业渗透越来越明显,计算机技术也正从一项或是一个系列性的技术技能,升蜕为一种思维模式,并进而深刻影响着人们分析与解决问题的角度和方式。换言之,计算思维已成为当今艺术设计领域从业人员不可或缺的素质和能力。尤其是其与艺术思维的互补和互动,彰显出鲜明的创新驱动性和广阔的发展前景。因此,在新的趋势下通过教材的改革创新将计算思维引入课程和训练环节,对推动计算机教学的改革与研究,具有积极的现实意义。

随着艺术设计人才培养改革的深入,如何构建以计算思维培养为导向的课程体系,探讨计算思维方式培养与应用的教学模式,提升学生多元化思维的能力,已成为时代的重要命题。本系列教材的编写,体现了各编著者多年来在此领域的努力和经验,反映了他们长期以来对计算机教学理论与实践探索总结的成果,具有较强的针对性和问题意识。教材内容力求在理论层面,从以计算思维为基础、与艺术思维相结合的角度,形成对计算机课程和知识结构、体系的务实探讨,从而有利于学生多元化思维的建立,以适应时代与社会、行业与职业的发展需求。

相信通过本系列教材的出版发行,将进一步引发艺术设计领域同仁对计算思维及计算思维模式与应用技术的关注和重视,推动计算机教学改革与课程建设的深层次尝试和探究。

是为序。

清华大学美术学院 何 洁 杨 静

2016年12月

# 前言

游戏是一门艺术,也是一门技术,正是这样的交叉学科所具有的独特魅力,使其不仅在商业上获得成功,也在当代艺术中倍显光彩。游戏及由游戏所衍生的模拟化数字互动形式,正以更快的速度进入大众视野,可以预见在不远的将来,以游戏形式呈现的娱乐、学习、社交、生活等将如同当前的电子设备一般牢牢地渗透我们的世界,并在其形态与功能上不断推陈出新,以更贴近我们需求的方式无所不在。

游戏产业的发展之迅速,是偶然更是必然。自2000年开始的网络游戏繁荣,是互联网普及与文化消费贫乏的必然;2006年网页游戏兴起,是社交媒体迅速发展、网络带宽提速与浏览器性能提升的必然;2008年开始的手机游戏大潮是以苹果为首的高配置智能手机普及的必然;手机游戏大有替代网络游戏的趋势也是手机的高使用频度及其显示、计算等性能提升的必然;而自2009年开始的网络游戏衰退,则是在网页游戏、手机游戏普及与文化消费观念回归的大环境下产生的必然。

在上述发展过程中,国内的产业形态也在发生着不断的变化。从代理到买入再到合作研发,从外包制作到自主研发以及定制开发,游戏产业及游戏从业者在摸索中前进。从最初就已被证明这是一条宽广的大道,在经历过几次大互通后,这早已不再仅仅是一条大道,而是广泛散开的路网。这个产业的人才向来都是缺乏的,前期外包制作的时候缺乏美术人才,当前自主研发的时候缺乏技术人才与策划人才。

本书作为游戏开发技术的入门教材,适合游戏专业二、三年级同学阅读,或是其他专业有兴趣的同学在具备基本程序基础后阅读。在网络如此发达的今天,书作为知识与经验分享的平台其存在价值仍是不容置疑的。书更多地承担的是知识的牵引与思维的发散,正基于此,我们将内容压缩,教材做薄,以期在简化阅读背后引出更多的思考。

受限于个人能力,在本教材成稿后仍存有很多遗憾,烦请各位读者指正。本教材的编写过程中,有幸得到我的两名学生包洋洋与谢亚男的倾力协助,在此表示诚挚的感谢。书中使用的案例由毕时、颜亚威与崔容峰制作,在此一并表示诚挚的谢意。

严宝平

2015年11月

# 目 录

|                                   |    |
|-----------------------------------|----|
| <b>第 1 章 游戏程序设计概述</b> .....       | 1  |
| 1.1 游戏开发的进化 .....                 | 1  |
| 1.1.1 数字游戏产生前.....                | 1  |
| 1.1.2 数字游戏的产生.....                | 2  |
| 1.1.3 复杂的硬件催生丰富的游戏内容.....         | 3  |
| 1.2 编程语言与脚本语言 .....               | 4  |
| 1.2.1 编程语言与脚本语言的区别.....           | 4  |
| 1.2.2 主要的编程语言与脚本语言.....           | 5  |
| 1.3 游戏基本类型 .....                  | 7  |
| 1.3.1 从数字游戏的介质中区分游戏类型.....        | 7  |
| 1.3.2 从数字游戏的内容中区分游戏类型.....        | 8  |
| 1.4 游戏中的常见功能.....                 | 13 |
| 1.4.1 场景的动态加载 .....               | 13 |
| 1.4.2 角色动作系统管理及驱动 .....           | 14 |
| 1.4.3 自动寻路系统 .....                | 15 |
| 1.4.4 资源的回收与优化 .....              | 15 |
| 1.4.5 回合制战斗系统 .....               | 16 |
| 1.4.6 塔防系统中的数据管理 .....            | 16 |
| 小结 .....                          | 18 |
| <b>第 2 章 Flash 游戏开发基础</b> .....   | 19 |
| 2.1 Flash 游戏开发历程 .....            | 19 |
| 2.1.1 Flash 的诞生与早期的发展 .....       | 19 |
| 2.1.2 Flash 在游戏设计领域的起步 .....      | 19 |
| 2.1.3 Flash 在社交网页游戏中的主导地位建立 ..... | 20 |
| 2.2 Flash 基本要素 .....              | 21 |
| 2.2.1 新版本 Flash CC 的特征内容.....     | 21 |
| 2.2.2 Flash 基本概念 .....            | 21 |
| 2.3 ActionScript 脚本 .....         | 25 |
| 2.4 程序设计的必要概念.....                | 26 |
| 2.4.1 变量、常量及数据类型.....             | 27 |

|            |                            |           |
|------------|----------------------------|-----------|
| 2.4.2      | 变量作用域 .....                | 31        |
| 2.4.3      | 函数、形参、实参 .....             | 32        |
| 2.5        | 程序的基本结构 .....              | 33        |
| 2.5.1      | 顺序结构 .....                 | 33        |
| 2.5.2      | 选择结构 .....                 | 33        |
| 2.5.3      | 循环结构 .....                 | 35        |
| 2.6        | 事件 .....                   | 36        |
| 2.6.1      | Event 事件类型 .....           | 37        |
| 2.6.2      | MouseEvent 鼠标事件类型 .....    | 37        |
| 2.6.3      | KeyboardEvent 键盘事件类型 ..... | 38        |
| 2.6.4      | TimerEvent 计时器事件类型 .....   | 38        |
| 2.7        | 游戏中的元素 .....               | 39        |
| 2.7.1      | 美术素材 .....                 | 39        |
| 2.7.2      | 音频素材 .....                 | 40        |
| 2.7.3      | 视频素材 .....                 | 40        |
| 小结         | .....                      | 41        |
| <b>第3章</b> | <b>记忆双消游戏实例 .....</b>      | <b>42</b> |
| 3.1        | 游戏策划中的功能设计 .....           | 42        |
| 3.1.1      | 消除类游戏简述 .....              | 42        |
| 3.1.2      | 本案例的功能计划 .....             | 43        |
| 3.2        | 场景元素的组织与动态生成 .....         | 44        |
| 3.2.1      | 游戏的开始界面 .....              | 44        |
| 3.2.2      | 游戏的主运行界面 .....             | 46        |
| 3.3        | 游戏控制的实现 .....              | 52        |
| 3.3.1      | 单击的控制 .....                | 52        |
| 3.3.2      | 消除的实现 .....                | 53        |
| 小结         | .....                      | 55        |
| <b>第4章</b> | <b>飞行射击类游戏实例 .....</b>     | <b>56</b> |
| 4.1        | 游戏策划中的平衡性设计 .....          | 56        |
| 4.2        | 飞行射击游戏功能分析 .....           | 56        |
| 4.2.1      | 飞行射击游戏的参考 .....            | 56        |
| 4.2.2      | 飞行射击游戏功能分析 .....           | 58        |
| 4.3        | 战斗场景的控制 .....              | 59        |
| 4.4        | 战斗系统与资源回收 .....            | 64        |
| 4.4.1      | 敌方飞机的不断出现与攻击 .....         | 64        |
| 4.4.2      | 敌我双方子弹的功能实现 .....          | 65        |
| 4.5        | 动态表现与游戏特效 .....            | 68        |
| 小结         | .....                      | 69        |

|                                   |     |
|-----------------------------------|-----|
| <b>第 5 章 游戏引擎概述</b> .....         | 70  |
| 5.1 游戏引擎的基本概念 .....               | 70  |
| 5.2 游戏引擎的功能模块 .....               | 72  |
| 5.3 主流游戏引擎分析 .....                | 73  |
| 5.3.1 Unreal 虚幻引擎 .....           | 74  |
| 5.3.2 CryEngine .....             | 76  |
| 小结 .....                          | 79  |
| <b>第 6 章 Unity 开发入门</b> .....     | 80  |
| 6.1 Unity 介绍 .....                | 80  |
| 6.1.1 使用 Unity 开发的游戏 .....        | 80  |
| 6.1.2 Unity 游戏引擎的发展历程 .....       | 83  |
| 6.1.3 Unity 5 及其技术优势 .....        | 83  |
| 6.2 安装与运行 .....                   | 84  |
| 6.2.1 Unity 的安装 .....             | 84  |
| 6.2.2 Unity 的激活 .....             | 87  |
| 6.2.3 样例工程分析 .....                | 90  |
| 6.2.4 工程文件的新建与维护 .....            | 93  |
| 6.3 Assets Store .....            | 94  |
| 6.4 C# 脚本 .....                   | 95  |
| 小结 .....                          | 105 |
| <b>第 7 章 打地鼠游戏实例</b> .....        | 106 |
| 7.1 游戏对象与地形 .....                 | 106 |
| 7.1.1 游戏对象的类型 .....               | 106 |
| 7.1.2 Terrain 地形 .....            | 109 |
| 7.2 游戏的视角 .....                   | 115 |
| 7.2.1 第一人称视角游戏 .....              | 115 |
| 7.2.2 第二人称视角游戏 .....              | 116 |
| 7.2.3 第三人称视角游戏 .....              | 116 |
| 7.3 使用 2D ToolKit 插件包构建游戏场景 ..... | 117 |
| 7.3.1 2D ToolKit 概述 .....         | 117 |
| 7.3.2 2D ToolKit 基本使用 .....       | 117 |
| 7.3.3 打地鼠场景构建 .....               | 119 |
| 7.4 地鼠的动态生成与受控 .....              | 121 |
| 小结 .....                          | 122 |
| <b>第 8 章 投射类游戏实例</b> .....        | 123 |
| 8.1 Unity 中的物理系统 .....            | 123 |



|               |                               |            |
|---------------|-------------------------------|------------|
| 8.1.1         | 重力与摩擦力系统                      | 123        |
| 8.1.2         | 物理材质系统                        | 125        |
| 8.1.3         | 碰撞检测                          | 128        |
| 8.2           | 游戏功能计划与开发周期预算                 | 130        |
| 8.2.1         | 游戏的功能计划                       | 131        |
| 8.2.2         | 游戏的功能点分析及周期预算                 | 131        |
| 8.2.3         | 开发流程控制                        | 133        |
| 8.3           | 主体功能设计                        | 135        |
| 8.3.1         | 飞镖的投射及控制                      | 135        |
| 8.3.2         | 飞镖的击中与稻草人控制                   | 137        |
|               | 小结                            | 138        |
| <b>第 9 章</b>  | <b>增强现实与独立游戏</b>              | <b>139</b> |
| 9.1           | 增强现实                          | 139        |
| 9.1.1         | 增强现实的产生                       | 139        |
| 9.1.2         | 增强现实应用的领域                     | 140        |
| 9.2           | 基于高通 Vuforia SDK 的 AR 实现      | 142        |
| 9.2.1         | 关于高通公司及其 Vuforia              | 142        |
| 9.2.2         | 与 Unity 配合的增强现实开发             | 142        |
| 9.3           | 基于微软 Kinect 的体感游戏开发           | 146        |
| 9.3.1         | 微软体感硬件 Kinect                 | 146        |
| 9.3.2         | Unity 和 Kinect 交互的环境配置        | 147        |
| 9.3.3         | 用于 Kinect 开发的包中包含的资源          | 147        |
| 9.3.4         | 在工程中使用 KinectWrapperPackage 包 | 148        |
| 9.4           | 独立游戏与实验游戏                     | 150        |
| 9.4.1         | 独立游戏代表作：陈星汉《旅程》               | 150        |
| 9.4.2         | 独立游戏代表作——《机械迷城》               | 151        |
|               | 小结                            | 153        |
| <b>第 10 章</b> | <b>跨平台游戏开发</b>                | <b>154</b> |
| 10.1          | 跨平台趋势                         | 154        |
| 10.1.1        | 跨平台前期的障碍                      | 154        |
| 10.1.2        | 跨平台发展的契机                      | 154        |
| 10.2          | 网页游戏平台及其特点                    | 156        |
| 10.3          | 手机游戏平台及其特点                    | 158        |
| 10.4          | 电视游戏平台及其特点                    | 160        |
|               | 小结                            | 161        |
|               | 参考文献                          | 162        |

# 第 1 章

## 游戏程序设计概述

游戏是一个极为广泛且普通的词汇,至今未有一个明确的定义来描述它具体是什么。柏拉图给的定义是:“游戏是一切幼子(动物的和人的)生活和能力跳跃需要而产生的有意识的模拟活动”。亚里士多德这样描述:“游戏是劳作后的休息和消遣,本身不带有任何目的性的一种行为活动”。亚里士多德强调游戏本身不带有目的性,我们可以理解为游戏的目的性比较松散或是不够明确,个人更倾向于柏拉图所描述的游戏是“有意识的模拟活动”。

甚至关于艺术的起源,也存在于游戏相关的“席勒—斯宾塞理论”。该理论最早由康德提出,后经席勒发展,其认为艺术活动是无功利无目的的自由的的游戏活动,是人与生俱来的本能,艺术就起源于人的游戏本能或冲动。当然该理论中所提及的游戏,是一种更为广泛存在的生活状态,其与工作相对应,可以理解为无太多约束力的自由活动。虽然这里的游戏与常识中理解的游戏是两个不同的概念,但其中仍有部分相通之处。可以理解为通识中的游戏是哲学家所表述之游戏的具象化。

古老的哲学家对游戏的定义,另一方面也在说明游戏不是现代文明才具有的事物。游戏与生俱来,人类在自身文化创造的过程中同时亦创造了大量的游戏文化。例如,棋局、积木、纸牌等,还有追逐、接力及利用球、棒、绳等器材进行的活动,有个体活动也有集体活动,并有情节和规则,具有竞赛性。这些游戏与游戏机无关,与计算机无关,但现代的游戏机、计算机大多吸取并继承了这些原始的游戏内容,用富有创意的方式重新表达了这些内容。

本书所讨论的游戏,是指狭义上的数字游戏,本书重点讨论数字游戏中的程序设计与游戏引擎的使用。作为一本入门教材,旨在让读者了解游戏程序设计的原理与方法,并能够独立使用游戏开发工具完成游戏的设计规划与功能实现。

### 1.1 游戏开发的进化

#### 1.1.1 数字游戏产生前

如前文所述,游戏的历史可以上溯到与人类同步的文明发展史,在本书所讨论的数字游戏产生之前,狭义中玩的游戏已经形成了一个很大的产业。例如,著名的日本任天堂游戏公司最早的历史可以追溯到 1889 年由其创始人建立的 Marufuku 公司,其核心业务为生产和销售一种名为 Hanafuda 的日本牌,直到 1951 年,该公司才更名为“任天堂”(Nintendo),取意“把好运留在天堂”,但彼时仍取名为任天堂牌业公司。随后进入 20 世纪 50 年代中后期,任天堂建立了独立的电子游戏部门,在后续的演变过程中,传统的牌业游戏日渐衰落,而电子游戏业务使这家公司成长为全球瞩目的行业巨头。

另一个日本的游戏开发巨头世嘉(Seга)与任天堂不同的是,该公司起步时即涉及与电子游戏较近的业务,1940年该公司创始人在夏威夷的火奴鲁鲁成立了一家名为“标准游戏公司”的实体,专门生产投币机,此时的投币机用于基于物理机械而运行的游戏类型。1952年在迁移东京一年后注册了名为“服务游戏公司”的实体,其英文名 Service Games 正是后来缩写成世嘉(Seга)的全称。在后续的发展过程中,通过并购 Rosen 公司拓展了电子游戏机的生产与销售业务。

在以上任天堂牌业与世嘉的投币机时代,可称为前数字游戏时代,从1958年开始,电子游戏的产生标志数字游戏时代的开始。

### 1.1.2 数字游戏的产生

关于世界第一款数字游戏软件的定义要根据电子和计算机的不同定义展开。

#### 1. 电子游戏的产生

如果以电子游戏为界定,则应该是1958年诞生的 Pong,国内将其翻译成“乒乓”。该游戏用示波器与实验室里的模拟计算机相连,毫无画面可言。即便到了1972年,历史上首家游戏机制造公司 Atari(雅达利)在其发布的第一款游戏机里置入了这一游戏,加入了在当时堪称美妙的操控模式,但画面仍是简陋至极(如图1.1所示,此图显示为1975年该公司更新发布的家居版)。无论是1958年版本的 Pong,还是14年后的新版本重生,这些都是需要堪称电气工程师与计算机科学家这样的角色通力合作才可以创造出来的作品。在软件技术特别是编程技术近乎荒蛮的年代,即便画面再简单,这样的开发难度也是难以想象的。



图 1.1 Atari 公司发布的 Pong 游戏

Pong 产生于 20 世纪 50 年代,它的产生并不是什么奇迹,而是技术发展与众人关注焦点集中后的产物。另一个值得回顾的关键点是,早在 1951 年,一个名叫 拉尔夫·巴尔(Ralph Bear)的美国人最早提出了家庭控制台的概念,但可惜的是当时的公司没有采纳他的建议,实际这样的做法是对的,好的产品出生在不正确的时间也是没有市场的。直到 15 年后,巴尔去了另一家公司并将他的思路最终形成了产品,这就是大名鼎鼎的游戏始祖: Magnavox Odyssey(图 1.2),不过虽然开始的时候确实也产生了相当大的轰动效应,但由于后续改进不足,以至到 1972 年很快这款机器便停产了。但他的影响是巨大的,随后产生的任天堂的 FC 游戏机、世嘉的 Play Station 系列游戏机以及微软的 XBox 游戏机等家用游戏机都沿用了这一模式。另外,值得一提的是,Odyssey 推向市场的时候微软公司还没有成立。

#### 2. 计算机游戏的产生

以计算机游戏为界定,第一款计算机游戏诞生于 1961 年。当时,麻省理工学院的“组装

机室”新进了一台 PDP-1(等离子显示器)微型计算机,这台计算机的大小和一台冰箱相当,在今天看来硕大无比,当时却是非常小巧新潮。学生斯蒂夫·罗素、阿兰·科托克与另外两位同学通力合作,用 40 页的源代码完成了双人射击游戏《太空大战》(图 1.3 所示)。

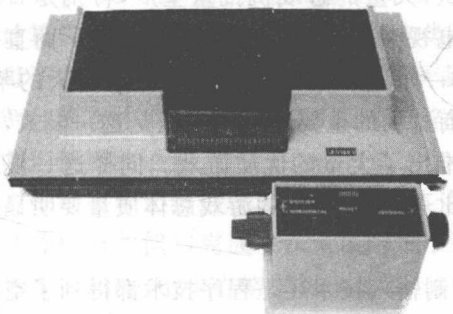


图 1.2 Odyssey 游戏机

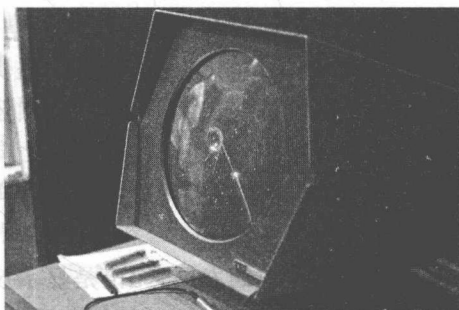


图 1.3 双人射击游戏《太空大战》

游戏中,玩家使用一对专用的控制器进行操作,控制飞船在屏幕上左右飞行,并使用导弹(不受引力影响,但射程短、威力小)和激光(会受引力影响发生偏转,但射程远、威力大)进行攻击,此外他们还在游戏中设计了用于脱逃的超高速空间“超空间”,不过其出现地点和概率都是随机的。

这样一个看似纯粹源自兴趣引导的研究活动,却在不经意间影响了计算机的发展方向。而后在 20 世纪 70 年代的个人计算机的发展高潮中,以乔布斯创立的公司引领的苹果风潮,将个人计算机的功能正式定位在了办公与游戏。

### 3. 早期电子游戏的开发

受制于硬件的水平、操作系统及编程语言,此时的游戏开发是极为复杂的,被视为超高智商的任务。如果拿现在的情况来类比,相当于我们给一个智能型的计算器外接一个显示器,然后用它来运行游戏。可能这样的说法有点夸张,不过当你看到 20 世纪 70 年代的堪称先进的控制台,就会认同这样的说法了。1977 年,前文所提及的著名游戏机及游戏软件提供商 Atari 公司推出了 Atari 2600,该机器使用的 6507 型号的 CPU,其运算速度只有 1.19MHz,装有 128B 的内存。CPU 运算量只有当前中低端手机的 1/3000(以相对已经落后的 1.5G 双核为参照),内存仅为当前同档手机的近 0.1%(以当前低端配置的 1GB 内存为参照)。但正是这样在现在看来无比落后的设备,创造了游戏机启蒙时代的辉煌。

早期的电子游戏,通常由一个人负责整个项目。绘图、然后将图形象素化、编写程序代码、采集或是制作声音等均由一个人完成。当然在此过程中也会需要艺术家提供帮助,但真正当艺术介入后发现能做的有限,因为在如此简陋的屏幕上也很难表现出多么复杂的画面,更糟糕的是功能也必须受限于内存的有限而放弃复杂的关卡。

#### 1.1.3 复杂的硬件催生丰富的游戏内容

也正是从前文提到的 1977 年开始,苹果公司推出了 Apple II 型个人计算机,虽然该机器用的 6502 型号的 CPU,与 Atari 2600 相差无几,但苹果创造性地为其配备了 16KB 的内存,这相比之前的 128B 内存,可以说是质的飞跃。苹果的工程师为其第二代机配备了完整

的 BASIC 解释器以便编译个人开发的复杂程序,并提供 QWERTY 键盘、磁带盒接口和游戏杆、游戏板等专用的游戏 I/O 接口,再配上一个 16 色的 280 像素×192 像素的显示器,这简直就是游戏的理想发布平台了。

到了 20 世纪 80 年代,游戏机产生了质的飞跃,大量的游戏机批量生产,特别是日本的世嘉、任天堂、索尼等公司推出的一系列掌上型、电视型游戏机。由于终端是由厂商直接控制,运行于终端的游戏自然也是由厂商提供,这样一个闭环的商业模式类似于今天的苹果,这样的模式使得硬件在售卖后,软件依然还是厂商盈利的重要阵地。也正是这一模式,在厂商之间的竞争过程中,促使以厂商为主力向市场推出了大量的优质游戏。同样这一模式也被苹果所继承,直到今天仍然可以看到这样的对比,苹果商店里的游戏总体质量要明显高于安卓市场。

随着计算机技术的发展,无论是显示技术、控制输入技术还是程序技术都得到了空前的进步。尤其是进入 20 世纪 90 年代中后期,工具的引入使得游戏开发变得更加容易。在当今 Flash 软件或是游戏引擎的支持下,当年 40 页的源代码量可以在只有 40 行脚本的情况下实现与原有一致的功能。甚至有的游戏引擎可以做到不需要一行代码同样可以实现诸多游戏的功能。本书在后续章节中将重点介绍如何使用 Flash 开发工具与 Unity 3D 游戏引擎方便、快捷地开发出不同类型的游戏。

## 1.2 编程语言与脚本语言

游戏设计主要可以划分为策划、美术设定、美术制作、程序设计、发布与测试等几个环节,游戏的重要特色在于其互动体验,程序设计正是实现这一互动体验的过程,因此其承担了非常重要的工作。也正是因为程序的作用,才使游戏与动画有了最为明显的区别。

程序的工作直观理解是让画面动起来,制定并实现一定的规则,让玩家与系统通过相互的交流动起来。它所负责的内容是相当丰富的:驱动美术元素、管理输入输出设备、管理游戏内部的数据等。用美术元素、设备与数据构造一个虚拟的世界。

1962 年,麻省理工学院(MIT)学生史蒂夫·拉塞尔(Steve Russell)设计出了 Spacewar! (太空大战),这是真正运行在计算机上的第一款交互式游戏。当时计算机技术还相当有限,空间大战必须使用新阴极射线管显示器来显示画面。史蒂夫·拉塞尔使用汇编语言编写了这个游戏,也开启了计算机游戏的时代。

到今天,数字游戏已经成为了大多数人生活的一部分,不仅在于娱乐,同时也成为了教育与商业宣传的工具之一。特别是进入 1996 年后,计算机经历了前所未有的发展,用户群及商业模式都创造了不少的奇迹。

游戏的画面发展早已超越了当年那个简单、朴素的画面,各种各样绚丽的画面层出不穷,同样,编写游戏的计算机程序也经历了很大的变化。当前使用的游戏程序比当年那个枯燥机械的汇编高级了许多,各种语言和脚本构建了缤纷复杂的程序世界。

### 1.2.1 编程语言与脚本语言的区别

脚本语言在从属上是编程语言的一种,编程语言包括汇编语言、脚本语言与高级语言。但习惯上,在非严密的情况下,人们经常将高级语言表述编程语言,与脚本语言相对独立,以

下本文的编程语言即指高级语言。

计算机语言是指计算机所使用的语言,受制于计算机本身严密化的二进制计算,在其语言的使用上极为强调逻辑性与严密度,所以其很难像人类语言那样可以用模糊化的语义表达,甚至同一词句在不同语境是表达完全不同的意思,计算机语言相对比较直接。另外,实际上计算机所能直接识别的语言不像人类这样不同人种拥有不同的母语,计算机世界只有两种母语,那就是机器语言以及在机器语言基础之上产生的汇编语言。这堪称世界上最难懂的两语言体系。早在计算机诞生甚至直到20世纪七八十年代,汇编语言一直是计算机工程师们使用的语种。

高级语言和汇编语言相比,它不但将许多相关的机器指令合成为单条指令,并且去掉了与具体操作有关但与完成工作无关的细节,如使用堆栈、寄存器等,这样就大大简化了程序中的指令。同时,由于省略了很多细节,编程者也就不需要有太多的专业化机器硬件相关的知识。

高级语言所编制的程序不能直接被计算机识别,必须经过转换才能被执行,按转换方式可将它们分为两类。

**解释类:**该类语言在执行时,计算机一边读一边执行,应用程序源代码被同步地由相应语言的解释器转换成计算机硬件能够识别的机器语言。因此效率比较低,而且不能生成可独立执行的可执行文件,应用程序不能脱离其解释器,但这种方式比较灵活,可以动态地调整、修改应用程序,如较早时期的QBASIC语言。

**编译类:**编译是指在应用源程序执行之前,就将程序源代码转换成机器语言,因此其目标程序可以脱离其语言环境独立执行,使用比较方便、效率较高。但相较解释类语言,其每次更改哪怕是很小的一些符号的更改都需要重新执行编译、连续流程,牺牲的是编码端的制作过程而提升使用端的执行效率。

两者各有千秋,不过相对复杂的高效能计算还是得使用编译类的语言,代码编写者运行程序的次数肯定会远远低于用户,毕竟用户的感受才是真正决定软件存亡的核心标准。

## 1.2.2 主要的编程语言与脚本语言

以下简单分析主要的游戏设计语言。

### 1. C++

C++是在C语言的基础上开发的一种集面向对象编程、泛型编程和过程化编程于一体的编程语言。应用较为广泛,是一种静态数据类型检查的、支持多重编程的通用程序设计语言。它支持过程化程序设计、数据抽象、面向对象设计等多种程序设计风格。

一开始C++是作为C语言的增强版出现的,从给C语言增加类开始,不断地增加新特性。虚函数(virtual function)、运算符重载(operator overloading)、多重继承(multiple inheritance)、模板(template)、异常(exception)、RTTI、命名空间(name space)逐渐被加入标准。

### 2. Java

Java平台由Java虚拟机(Java Virtual Machine, JVM)和Java应用编程接口

(Application Programming Interface, API)构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口,可分为基本部分和扩展部分。在硬件或操作系统平台上安装一个 Java 平台之后,Java 应用程序就可运行。现在 Java 平台已经嵌入了几乎所有的操作系统。这样 Java 程序只编译一次,就可以在各种系统中运行。

Java 分为 3 个体系即 JavaSE(Java 2 Platform Standard Edition,Java 平台标准版)、JavaEE(Java 2 Platform,Enterprise Edition,Java 平台企业版)及 JavaME(Java 2 Platform Micro Edition,Java 平台微型版)。

### 3. C

这个顺序似乎有点颠倒,前面分析了 C++,到这里才出现 C。主要在于 C 用于游戏编程不如 C++ 主流。C 基于流程,没有对象的概念,这样的方式多用于嵌入式设备的程序设计。且随着平台与技术的发展,越来越多的高级语言或开发工具介入到了比较低端的游戏开发环境,致使 C 的市场进一步缩小,但不可否认的是,由 C 系衍生出的开发语言越来越多地用在了游戏开发领域。

### 4. Objective-C

通常写作 ObjC 或 Obj-C,是扩充 C 的面向对象编程语言。它主要用于 Mac OS X 和 GNUstep 这两个使用 OpenStep 标准的系统,而在 NeXTSTEP 和 OpenStep 中它更是基本语言。Objective-C 可以在 gcc 运作的系统写和编译,因为 gcc 含 Objective-C 的编译器。随着 iPhone 系相关硬件的推广,Objective-C 的位置也在逐步爬升。早在 iPhone 之前,在 Mac 系统中,这已经是有着绝对优势的语言了。

### 5. C#

C# 是微软公司发布的一种面向对象的、运行于 .NET Framework 之上的高级程序设计语言。并定于在微软职业开发者论坛(PDC)上登台亮相。C# 看起来与 Java 有着惊人的相似。它包括了如单一继承、接口、与 Java 几乎同样的语法和编译成中间代码再运行的过程。但是 C# 与 Java 有着明显的不同,它借鉴了 Delphi 的一个特点,与 COM(组件对象模型)是直接集成的,而且它是微软公司 .NET Windows 网络框架的主角。

C# 是一种安全的、稳定的、简单的、优雅的,由 C 和 C++ 衍生出来的面向对象的编程语言。它在继承 C 和 C++ 强大功能的同时去掉了一些它们的复杂特性(如没有宏和模板、不允许多重继承)。C# 综合了 VB 简单的可视化操作和 C++ 的高运行效率,以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为 .NET 开发的首选语言。并且 C# 成为 ECMA 与 ISO 标准规范。C# 看似基于 C++ 写成,但又融入其他语言如 Pascal、Java、VB 等。

### 6. JavaScript

通常缩写成 JS,这是一个与 Java 没有半点关系却极让人误以为是 Java 简化版的语言,严格意义上讲,这不是语言,只是脚本而已。JS 用于多款游戏引擎的脚本编写,同时,在基于网页编程的游戏设计中,JS 是较为重要的一环用来构建基于网页的互动机制。

JavaScript 是一种由 Netscape 的 LiveScript 发展而来的原型化继承的面向对象的动态类型的区分大小写的客户端脚本语言,主要目的是为了解决服务器端语言,如 Perl,遗留的速度问题,为客户提供更流畅的浏览效果。当时服务端需要对数据进行验证,由于网络速度相当缓慢,只有 28.8Kb/s,验证步骤浪费的时间太多。于是 Netscape 的浏览器 Navigator 加入了 JavaScript,提供了数据验证的基本功能。

## 7. ActionScript 与 Lingo

这两个写在一起,主要是两者来源于同一家公司,但相互之间没有太多的相似性。ActionScript 脚本语言类似于 C++,特别是发展到当前的 AS3,其功能的复杂性与对象的丰富性变得越来越强大。而 Lingo 脚本语言则更像是 BASIC 的延续,其语法结构与保留关键词也同 BASIC 如出一辙。本书第 2 章将详细讲解在 Flash 中使用的 ActionScript 脚本。

## 8. 其他

其他的脚本还有很多,如 Unreal 游戏引擎使用的 UnrealScript 脚本,3DGameStudio 游戏引擎使用的 Lite-C 脚本。但大多都能在以上所列举的类型找到与之对应的相似性。

# 1.3 游戏基本类型

数字游戏的类型随着它自身的发展和计算机技术的革新而变得日益丰富。一些数字游戏在设计中产生了某些技术进步成为独特的形态,也有更多的游戏是按照自身内容中反映的文化因素来区分类型的。因此,确立数字游戏的类型也是从某个视角去看待身边已经存在的各种游戏,找寻它们的共性与差异性,从中了解它们的历史意义与价值。

## 1.3.1 从数字游戏的介质中区分游戏类型

从数字游戏类型赖以存在的介质形态来看,它们包括计算机游戏、电视游戏、街机游戏、掌上游戏、网络游戏等。

### 1. 计算机游戏

计算机游戏也叫 PC(Personal Computer)游戏,是指以个人计算机为游戏介质的游戏类型。目前计算机游戏主要指游戏玩家在单一的个人计算机上进行的游戏活动。

### 2. 电视游戏

电视游戏也称为 TV(Television)游戏,它代表以电视作为游戏平台的各类游戏,电视游戏在设计时有着严格的限制条件,包含各个电视游戏机生产厂商为之定制的设备和要求。

### 3. 街机游戏

街机游戏是人们对这种游戏形态的形象称谓,也称为投币游戏机。它的英文叫做 Arcade 游戏,Arcade 本义是建筑的拱廊,在此引申为放置在酒吧和饭店等场所进行演示和娱乐的游戏类型。由于这种游戏形式的流行,它成为公共娱乐场所的一种独立形态。街机



游戏通常是以独立的橱柜装置进行展示,需要游戏者投币进去才可进行游戏。

#### 4. 掌上游戏

掌上游戏是指便于携带的各类数字游戏的总称。在早期,它是指运行在各种专门性手持游戏机上的游戏类型,如任天堂 Game Boy 上出现的游戏。随着电子技术的发展,它现在也指各类多功能的手持电子设备上出现的游戏,如手机游戏、PDA 游戏等。

#### 5. 网络游戏

网络游戏又称为在线游戏(Online Game)。它是指通过各种网络协议进行游戏的所有形式,如在小型局域网内进行电子竞技比赛的各种游戏。通过因特网作为介质的多人连线游戏现在已经发展成为网络游戏中最流行的形态。这种游戏可以使相隔千里的人们在虚拟世界中聚在一起。一些网络游戏的同时在线的人数达到数万人以上,也被称为超大型网络游戏。

### 1.3.2 从数字游戏的内容中区分游戏类型

根据数字游戏所包含的内容特点与游戏方式上的差异,它们可以分为动作类游戏、冒险类游戏、策略类游戏、角色扮演类游戏、体育类游戏、模拟经营类游戏、军事训练类游戏、仿真设备模拟类游戏、益智类游戏等类型。这些内容处于不断的创造当中,数字游戏在涉及游戏内容的类型差异时,常常在主导内容类型确定之后,融合其他的内容形态,使游戏获得更加丰富的表现力。

#### 1. 动作类游戏

动作游戏在形式上常以游戏角色所展现的各类丰富的动作技巧为主要特色,游戏玩家不断进行尝试,游戏从中测试玩家使用各种动作技能的反应。早期的格斗游戏、单人射击游戏即属此类。代表作品有《STREET FIGHTER》、《TEKKEN》、《DEAD OR ALIVE》、《HALO》等系列游戏。动作游戏同时也在积极地与其他的游戏类型相结合,发展出不同的分支。

动作游戏向来是很受欢迎的游戏类型,丰富华丽的动作、灵活多变的操作感,是爱好者对这类游戏的诉求。如《街头霸王》系列已推了第 4 部(图 1.4),《街头霸王 4》已经不再是进行“连击”,而是更趋于类似真实拳击中的感觉——提前读出对手的动作并加以反击。在此基础上推出了两种全新的游戏模式:在线训练模式、小组战斗模式。在线训练模式中,玩家可以和自己的好友进行友谊训练赛,而在小队战斗模式中,玩家可以选择 3 名角色进行比赛。新一代的动作类游戏会在画面、用户操作体验这两方面进行更加深入的挖掘。

#### 2. 冒险类游戏

冒险游戏主要以探索未知环境或新世界作为游戏的主线。游戏中经常融合了情节、解谜等元素的特点,使之独树一帜。玩家在冒险游戏中常常需要通过不懈努力研究各种线索,使用各类道具,甚至与游戏对手战斗并奔波于迷宫式的地形环境中,直至解开谜底,完成奇幻世界的探索。