

菜鸟与诸神

讲述 **ActionScript 3** 与 **Web Game** 的故事

邱鑫 编著

从零起步，阶梯平缓 重视思维，牢固基础
悉心引导，禅悟模式 谈笑风生，轻松入境

菜鸟与诸神：

讲述 ActionScript 3 与 Web Game 的故事

電子工業出版社·

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书采用故事模式讲述一门编程语言，以一个无知的菜鸟成长过程为背景，全文用菜鸟与大神的对话贯通，如同阅读一本探险小说，同时也在感受一名程序员在成长过程中的思想变革。

本书以准 C++、Java 程序员的阶梯模式构建内容。因为写作时注重编程思想的培养，其价值已经远远超过 ActionScript 3 这门语言，内容涵盖：面向过程、面向对象、开发流程、架构设计、数据结构、游戏引擎、算法等知识框架，对学习任何语言都是有帮助的，这也是“一通百通”的道理。

本书实战案例围绕当今热门的 Web Game，内容源自作者多年积累的开发笔记、项目和教学经验。本书还配备了案例代码以及相关课件，方便读者进一步学习。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

菜鸟与诸神：讲述 ActionScript 3 与 Web Game 的故事/邱鑫编著. —北京：电子工业出版社，2014.1
ISBN 978-7-121-21720-3

I. ①菜… II. ①邱… 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 (2013) 第 250604 号

责任编辑：张月萍

印 刷：三河市鑫金马印装有限公司

装 订：三河市鑫金马印装有限公司

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：787×980 1/16

印张：32 字数：699千字

印 次：2014年1月第1次印刷

定 价：79.00元（含光盘1张）



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。

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

服务热线：(010) 88258888。

目 录

第 1 篇 基础知识		第 3 章 面向过程设计	54
第 1 章 从零起步	2	3.1 运算符	55
1.1 计算机和程序语言	4	3.1.1 算术运算符	55
1.1.1 计算机硬件和二进制	4	3.1.2 关系运算符	56
1.1.2 语言的发展	5	3.1.3 逻辑运算符	57
1.1.3 操作系统与虚拟机	8	3.1.4 类型检查运算符	58
1.2 牵手 ActionScript 3	9	3.1.5 成员运算符	59
1.2.1 使用 Flash CS6 开发第 一个程序	10	3.1.6 位运算符	60
1.2.2 使用 Flash Builder 开发 第一个程序	16	3.1.7 赋值运算符	65
菜鸟总结	26	3.1.8 条件运算符	66
第 2 章 数据类型	27	3.1.9 逗号运算符	67
2.1 基本数据类型	27	3.1.10 运算符的优先级	67
2.1.1 变量与常量	28	3.2 语句	69
2.1.2 基本数据类型之间的转换 ..	29	3.2.1 顺序结构	69
2.1.3 认识数组	33	3.2.2 选择结构	70
2.1.4 处理字符串	40	3.2.3 循环结构	72
2.2 复杂数据类型与对象	47	3.3 函数	77
2.2.1 对象的创建和引用	48	3.3.1 定义函数	77
2.2.2 对象的销毁	51	3.3.2 函数参数	79
2.3 别具一致的强对象语言	52	3.3.3 变量在函数中的存活周期 ...	82
菜鸟总结	53	3.3.4 递归函数	82
		菜鸟总结	84
		第 4 章 游戏——汉诺塔	85
		4.1.1 游戏制作	85
		4.1.2 使用递归求解	92



第 2 篇 面向对象

第 5 章 面向对象概述	98
5.1 面向对象的特征	99
5.2 使用 Flash Builder 进行面向对象开发	102
5.2.1 创建 ActionScript 项目	103
5.2.2 创建包	105
5.2.3 类的全限定名	106
5.3 使用 Flash CS6 进行面向对象开发	108
5.3.1 时间轴	108
5.3.2 元件库	109
5.3.3 对象和自动引用声明	110
5.4 类的导入和发布	112
5.4.1 Flash Builder 中的库项目 ..	113
5.4.2 在 Flash CS6 中发布和导入 swc	117
5.5 开始程序员生涯中的积累	119
菜鸟总结	121
第 6 章 游戏——打地鼠	122
第 7 章 面向对象编程	131
7.1 as3 中的封装层次	131
7.1.1 访问控制符	132
7.1.2 包外类	135
7.2 类的成员	138
7.2.1 属性初始化	138
7.2.2 get/set 属性	140
7.2.3 构造函数	144
7.2.4 静态成员	145
7.3 继承和多态	149

7.3.1 this 和 super 关键字	149
7.3.2 实现多态	150
7.3.3 静态成员不参与继承	156
7.4 里氏代换原则	158
7.4.1 向上转型	158
7.4.2 向下转型	158
7.4.3 转型时常见错误	159
7.5 动态类	159
第 8 章 面向对象开发过程	163
8.1 OOAD	163
8.1.1 OOA	164
8.1.2 OOD	164
8.2 面向对象开发过程	165
8.2.1 瀑布模型	165
8.2.2 统一软件开发过程 (USDP)	166
8.3 统一建模语言 UML	168
8.3.1 诞生背景	169
8.3.2 UML 构成	170
8.3.3 UML 中的公共机制	174
8.3.4 UML 绘制工具	176
8.4 其他工具	183
8.4.1 Visio	183
8.4.2 FlexFormatter	184
8.4.3 MindManager	186
第 9 章 游戏——敲砖块	188
9.1 开发过程	188
9.2 OOA 部分	189
9.3 OOD 部分	191
9.4 OOP 部分	194
9.5 同步代码	205
9.6 反向工程	207



9.7 生成文档	208	11.7.2 抽象工厂模式	296
第 10 章 架构与设计	213	11.8 交互中的耦合	301
10.1 抽象类	214	11.8.1 观察者模式	301
10.1.1 定义抽象类	215	11.8.2 命令模式	311
10.1.2 抽象方法	217	11.9 模式合作	324
10.2 接口	217	11.9.1 复合模式	326
10.2.1 UML 中的接口	222	11.9.2 MVC 模式	327
10.2.2 使用接口设计架构	224	11.10 模式夜话	345
10.2.3 抽象类与接口	226	11.10.1 平衡的世界	345
10.3 命名空间	231	11.10.2 模式分类	348
10.3.1 定义和使用命名空间	231	11.10.3 发掘自己的模式	349
10.3.2 命名空间的特性	234	第 12 章 游戏——大鱼吃小鱼	352
10.4 对图形化语言的探索	236	12.1 游戏介绍	352
第 11 章 设计模式	239	12.2 游戏开发	352
11.1 OO 原则	240	12.2.1 初始阶段	353
11.2 认识设计模式	242	12.2.2 细化阶段	355
11.3 创建全局访问点	243	12.2.3 构造阶段：首次迭代	356
11.3.1 静态库	244	12.2.4 构造阶段：第 1 次迭代 ...	357
11.3.2 单例模式	246	12.2.5 构造阶段：第 2 次迭代 ...	363
11.4 封装变化的部分	248	12.2.6 构造阶段：第 3 次迭代 ...	374
11.4.1 策略模式	248	12.2.7 构造阶段：第 4 次迭代 ...	388
11.4.2 状态模式	255	12.2.8 交付阶段	392
11.5 巩固不变的部分	272	12.3 项目总结	392
11.5.1 适配器模式	273	第 3 篇 显示编程	
11.5.2 装饰者模式	275	第 13 章 as3 中的事件架构	395
11.5.3 外观模式	279	13.1 事件和信号	395
11.6 减少依赖	284	13.2 同步调用和异步调用	397
11.6.1 模板方法模式	284	13.3 事件需要一个架构	398
11.6.2 最少知识	287	13.4 架设自己的广播台	402
11.6.3 简单工厂	289	13.5 内置信号	409
11.7 依赖倒置	290		
11.7.1 工厂方法模式	291		



13.6 自定义信号	411	14.1.5 帧事件.....	442
13.7 冒泡机制	415	14.1.6 混合与滤镜.....	443
13.7.1 事件响应	416	14.1.7 渲染与重绘.....	454
13.7.2 截断事件流	418	14.2 InteractiveObject 的交互能力 ...	461
菜鸟总结	421	14.2.1 鼠标事件.....	461
第 14 章 as3 中的显示架构	423	14.2.2 键盘与焦点	472
14.1 DisplayObject 的显示能力	424	14.2.3 菜单.....	485
14.1.1 位置与大小	424	14.3 DisplayObjectContainer 的管理	
14.1.2 名字和引用	432	能力	497
14.1.3 可见性与透明度	435	14.3.1 深度管理.....	497
14.1.4 剪裁与遮罩	436	14.3.2 管理子对象.....	504

第 1 篇 基础知识



我是菜鸟，但我有梦想，我梦想有一天成为大神，我现在“菜”不是因为我不聪明，而是没有自信，当我翻开这本书时仿佛信心开始回归，因为高手都是从菜鸟成长起来的。

第 1 章

从零起步

诸神论坛

菜鸟：我是一只菜鸟，热爱游戏开发并梦想成为 Web Game 领域的专家，但我没有经验，也没有基础，还在长夜中摸索，看不清前面的道路。

大神：如今的菜鸟，应该比当年只会在网吧中上网却不会装系统的我强，因为那时我还买不起一台电脑，但是现在人手一台笔记本不成问题，所以你比我的基础好。

菜鸟：听起来真让人安慰，但我应该从哪里起步呢？由于没有基础，我去书店翻书时发现大部分知识都看不懂。

大神：从前有个阿拉伯商人，他起早贪黑，非常勤奋，却发现自己一直在亏损，后来他找到了原因，原来他算数不过关，总是算错账。基础不牢固造成的灾难最为严重，想成为一个程序员，首先要牢固基础知识，否则如漂泊在大海上的轻舟，经不起风吹雨打。

菜鸟：如果从头讲解计算机的基本原理，再加上软件的使用，怕是花掉本书的全部篇幅也不够吧。

大神：基础知识介绍用一章就可以了，有了清晰的轮廓，便可洞察自己缺乏之处，从而去寻找相关资料查阅，以求不对后面章节的理解造成障碍。月亮虽然没有太阳明亮，但足以让人看清前面的道路。

菜鸟：你应该知道，像我这样的菜鸟迷惑肯定不止一个，我会在学习过程中随时提出问题，希望你能替我解惑。

大神：书中有很多领域的专家，因此解答你问题的不止我一个，这里为了方便，所以统称为“大神”，我希望你在初学阶段做好笔记，学会总结，一步一个脚印。当你成长到一定阶段时，我们还会为你准备一些思考题或任务等待你去完成。

菜鸟：太好了！那我们现在就开始吧。

如果我们要从零起步，那么应该先弄清计算机的运行原理，然后为开发 Web Game 选择一门语言，并掌握这门语言的开发环境，最后一起做个简单的例子，一同度过黎明前的黑暗，



去海边观看壮丽的日出。当然，在讲解之前请先下载安装 Adobe Flash Professional CS6 和 Adobe Flash Builder 4.7 这两个软件，如图 1-1 和图 1-2 所示。你可以到 <http://www.adobe.com/downloads> 下载官方最新版本，等基础知识讲解完后你的软件就可以运行了。



建议不要下载绿色或简化版本，这些版本会删除一些我们需要的功能，例如 Flash 自带的组件或与 Flash Builder 协作的能力。或许你下载的版本比书中的版本要高，但对阅读本书影响不大，安装软件非常简单，这里不作讲解。安装后的软件界面如图 1-1 和图 1-2 所示。

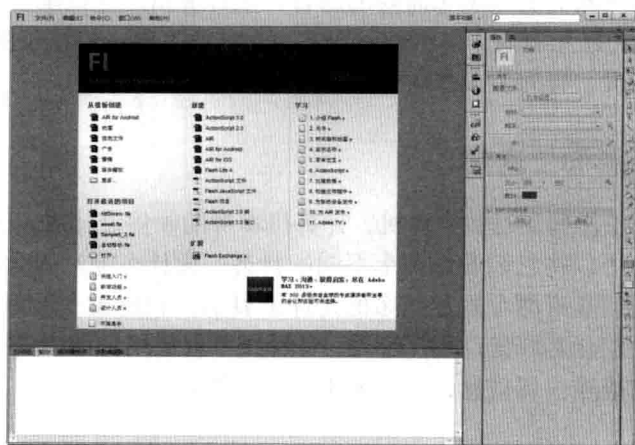


图 1-1 Adobe Flash Professional CS6

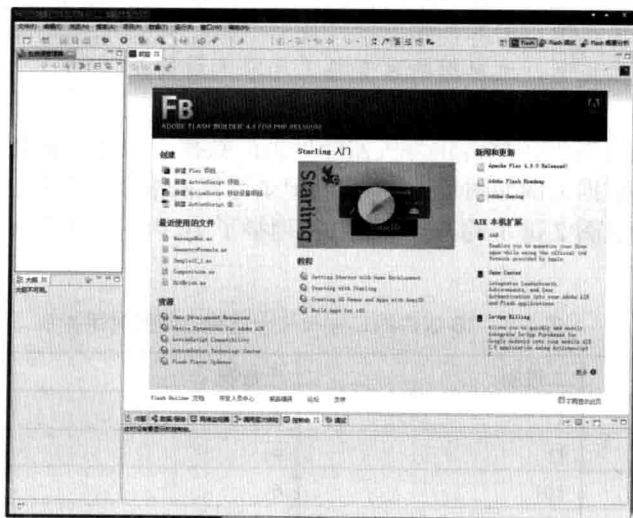


图 1-2 Adobe Flash Builder 4.7

在软件安装时我们便可以迈出属于菜鸟的第一步：了解计算机基本工作原理并认识



as3 语言。

1.1 计算机和程序语言

无论我们使用什么语言，都要依赖运行的硬件平台，那就是计算机，“电脑”这个称呼很容易给人误导，联想到拥有人类智商的“变形金刚”。实际上计算机是非生命体，它不会自己思考，只会执行预先输入的指令，这些集中的指令称为**计算机程序**。计算机程序是逻辑思维的表达，如同我们交流思想需要的语言，从计算机诞生的那一刻起，无数编程语言也随之诞生，到现在已经有两千多种编程语言，这些语言有各自擅长的领域并且相互竞争，有的已被时代所淘汰，有的历久弥新并且不断有新语言出现。

1.1.1 计算机硬件和二进制

回到我们赖以工作和学习的计算机，计算机本身的硬件设计决定了它只能识别 0 和 1 这两个数字，你也可以将它们看作两个不同的状态，为什么呢？因为电路实现最为简单，每个数都可以用任何具有两个稳定状态的元件来表示，制造成本最为低廉，譬如我们日常生活中的电灯开关，合上灯亮，断开熄灭。计算机不仅想用 0 和 1 描述电路是断开还是接通的，还尝试用它描述一切信息，于是诞生了二进制。



如果仿造人脑结构来制造计算机，那么其复杂程度将难以想象，即便最精密的元件也无法代替脑细胞。现代科学尚未将人脑研究透彻，何况仿制，大自然的鬼斧神工远非人所能及。

二进制研究将自然界信息转化为机器能处理的数据，因此二进制可以用于数学计算，也可以描述非数字数据（例如图片），还可以记录计算机指令。由于二进制只有 0 和 1 两个数字，因此只能采取满 2 进 1 的方式，表 1-1 列举了 10 以内二进制和十进制之间的对照关系。

表 1-1 10 以内的二进制和十进制之间的对照关系

十进制	二进制	十进制	二进制
1	1	2	10
3	11	4	100
5	101	6	110
7	111	8	1000
9	1001	10	1010



二进制可以描述小数和负数，拥有一系列计算法则，除了二进制，还有便于描述内存的八进制和十六进制，不管是十进制、八进制还是十六进制，最后都需要转化为二进制，交给计算机处理。对进制转化感兴趣的同学可以找本计算机一级考试的教材查阅，如果不喜欢手动计算，那么可以调用 Windows 自带的计算器帮我们快速转化，如图 1-3 所示。

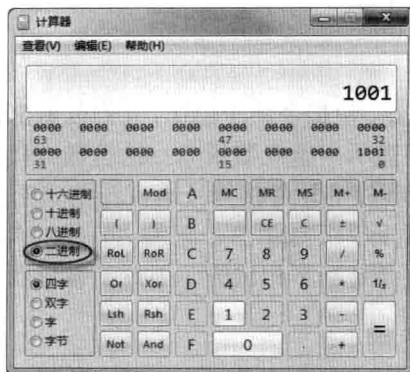


图 1-3 Windows 计算器

使用二进制进行数学运算在计算机中只是冰山一角，非数值领域远远超过数学计算，计算机中有文字、图片、音乐和影视等，它们都将信息转化为二进制存储和处理，这就是当今的“数码科技”，如图 1-4 所示。

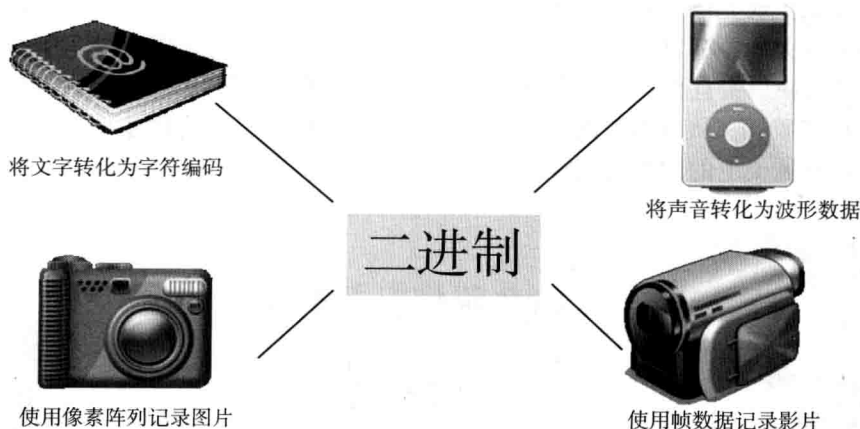


图 1-4 数码科技

1.1.2 语言的发展

计算机诞生后，计算机语言也经历了三个阶段。



● 机器语言

计算机刚刚诞生时还没有人性化的语言，由于计算机只能识别 0 和 1，所以编写的指令是长度为 16 的二进制编码，称为**机器指令**（machine instruction）。机器指令最直接，但比我们老祖宗发明的甲骨文还难理解和记忆，更谈不上开发效率。现在能使用二进制编程的人寥寥无几，这些人是稀有人才。

● 符号语言

为了克服机器指令的缺点，使指令便于记忆，人们创造出**符号语言**（symbolic language），符号语言用英文单词和数字表示一条机器指令，这是最直接的创新。显然计算机不认识这些符号，需要一种称为**汇编程序**的软件将符号语言转换成机器指令，转换过程称为“代真”或“汇编”，因此符号语言也称为汇编语言。汇编语言仍然与硬件打交道，为不同型号的计算机编写的汇编语言代码彼此是互不相认的，因此汇编仍然是**低级语言**，主要用于给设备编写驱动程序或烧录芯片。

● 高级语言

低级语言要求开发者非常了解计算机内部结构和工作原理，使用低级语言开发的程序不能在其他型号的计算机上运行，为了摆脱硬件束缚，将精力集中于逻辑自身，人们又研发了与硬件无关的高级语言。高级语言也需要“翻译”成二进制给机器识别，翻译高级语言的程序称为**编译程序**。与汇编不同的是，高级语言在编译时一条语句对应多条机器指令，这个过程就实现了将人类大脑思维转化为机器的逻辑指令，如图 1-5 所示。使用高级语言编写程序可以无须了解计算机内部结构，写出的程序还可以在不同型号的机器上运行，高级语言符合人脑思维且将“吃力不讨好”的部分交给了编译程序，因此效率得到了极大提升。如同算盘将数学家从奴隶般的计算中解脱出来，高级语言也将程序员从硬件中解脱，被称作计算机发展史上的“惊人成就”。



图 1-5 编译程序

当前项目开发大多使用的是高级语言，高级语言在成长中也经历了两个重要的阶段：



面向过程设计和面向对象开发阶段。这是因为计算机发明初期程序并不复杂，如编程解一些趣味数学题等，我们容易陶醉于代码执行的过程。像 Basic 这样的语言编程风格比较随意，程序流程可以任意跳转，这种语言称为**非结构化语言**。而后随着程序复杂性增加，我们意识到随意跳转很容易让人迷失，毕竟有着国际象棋九段逻辑能力的人千万人之中难得找到一个。为了增强阅读性，我们规定程序必须由上到下，使用良好的结构（循环、分支）构成，不允许随意跳转，这种语言称为**结构化语言**。

当程序规模进一步扩大时，结构化语言也变得吃力，面对几万条代码不能分割保存，修改较艰难，实现多人共同开发也没有较好的规则可循，我们还是不能从环环相扣的逻辑中解脱，整个开发流程需要变革。随着时代的进步，一种称为面向对象的思维模式诞生了，面向对象使用对象描述逻辑世界，这种思维符合大脑对自然界的认知方式，代码也以对象的方式分割保存，大大降低了程序的复杂性，并且适合多人合作开发。面向对象开发不仅分割对象，还有继承、多态等高级特性，这些特性我们会在后面的章节逐一讲述。

本书讲述的语言是 ActionScript 3（简称 as3），在某些章节还会涉及一点点 Java 语言，它们都是高级语言，as3 用于客户端程序开发，Java 用于编写服务器代码。TIOBE 是一个专业化的软件评估公司，每月都会对各语言的市场占有率进行统计并且公布结果，登录 <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>，可以查看最新的语言排名，如图 1-6 所示，本书编写时 as3 和 Java 分别排名 37 和第 1 名。

Position Mar 2013	Position Mar 2012	Delta in Position	Programming Language	Ratings Mar 2013	Delta Mar 2012	Status
1	1	=	Java	33	Erlang	0
2	2	=	C	34	Prolog	0
3	5	↑↑	Objective-C	35	Scratch	0
4	4	=	C++	36	Logo	0
5	3	↓↓	C#	37	ActionScript	0

图 1-6 编程语言排名

Java 出现很早，几乎遍及整个计算机领域，本书仅用于网络通信，因此涉及较少；as3 出现较晚，但随着 Web Game 的兴起迅猛发展，国内 as3 程序员更是供不应求，以至待遇比 Java 程序员还高。现在 Adobe 决定在 5 年内将 Flash Player 重写，使之成为 Web Game 中的霸主，那时我们在网页中也能玩“魔兽”这样的大型 3D 游戏了，as3 程序员前途非常光明。



从非结构化语言到结构化语言，从面向过程到面向对象，人们总是希望事情变得简单有条理，日常生活中何尝不是这样呢？自己的衣物分类叠放，拿取时方便寻找；写文章时将语言拆分为句子和单词，还要加上标点符号以便于理解，编程语言的进化同样是大脑习惯的回归。



1.1.3 操作系统与虚拟机

刚生产的机器没有安装任何软件，称为**裸机**，对于不会写程序的用户而言，它是一堆废品。为了日常操作和硬件管理，业界巨头开发出各自操作系统供广大群众使用。操作系统是可以独立运行于硬件上的软件，能够直接与硬件交流，用户操作和运行的应用程序都要由它管理。不是任何程序都可以在此操作系统上运行的，只有为此操作系统开发的应用程序才能运行于其上。程序启动时操作系统为它分配资源，程序关闭时操作系统回收该资源，在某时刻运行的应用程序，有多个是操作系统负责协调这些应用程序（如进程管理）。被开发成某个操作系统的应用程序的好处是减少开发量，因为操作系统提供通用库给应用程序调用，例如应用程序只需调用操作系统提供的保存文件的命令，向硬盘写数据、管理文件便交给操作系统了。



操作系统和应用程序一样也是高级语言编写的软件，Windows 和 Linux 都是用 C 语言编写的。它们也会编译为二进制代码执行，区别是操作系统生成的二进制程序以硬件支持的体系运行，其指令能够被硬件识别，而应用程序编译的二进制需要运行于操作系统之上，不能直接被硬件识别，只能算一个“半成品”。

被开发成某个操作系统的应用程序的缺点是对操作系统有依赖，换一个操作系统软件就不能运行了，例如 ps3 上的游戏不能在电脑上玩，要在电脑上玩有两种途径：一种是在 ps3 上安装 Windows 操作系统，另一种是进行**程序移植**——将游戏编译成其他操作系统能够执行的程序，其代价是很大的，可能需要重写。令人烦恼的是，如今硬件产品众多，硬件厂商往往只允许安装自己的操作系统，对于可以安装多个操作系统的个人计算机，不同操作系统之间的程序也是不能互通的，只能选择移植。这就使得开发者非常郁闷，被迫做吃力不讨好的事情。能不能寻找途径开发出与操作系统无关的应用程序呢？出于这个目的，有人提出了虚拟机的概念。虚拟机是一个软件，它在某个操作系统中虚构出一台计算机，虚构的计算机能仿真真实计算机的功能。虚拟机运行于各类操作系统之上，并努力虚拟出一个相同的运行环境，为这些虚拟机使用的高级语言能够在各种硬件平台和操作系统中畅通无阻（也可以说该虚拟机支持的语言与操作系统无关或跨平台语言），如图 1-7 所示。

不同国家的领导人谋求思想沟通时，迅速且廉价的方式是雇佣翻译，他们的重点不在于学习对方的语言。虚拟机将跨平台语言翻译给不同的操作系统，以操作系统支持的形式运行，操作系统再翻译为机器语言，这算不算又一个惊人成就？

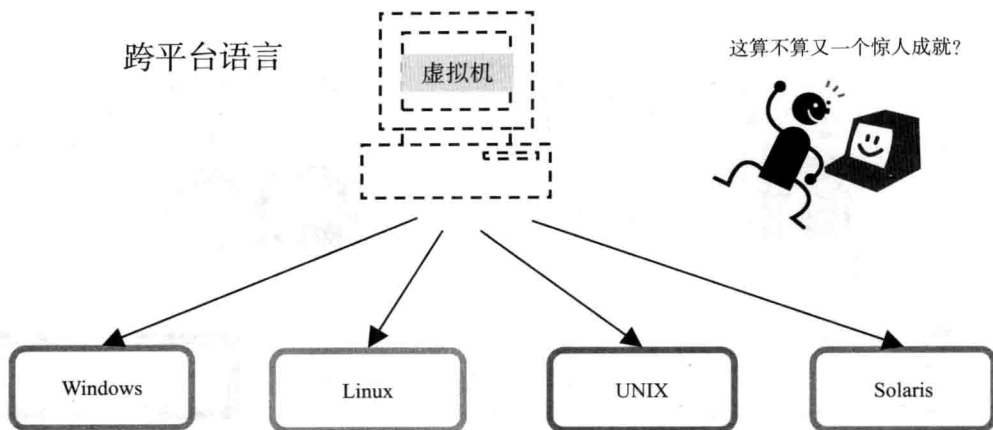


图 1-7 虚拟机



通过虚拟机再来看看语言的发展,不同地域的人创造了不同的文明,遍地开花后又需要谋求共识,程序语言的发展何尝不是这样?初期因为需求不同而创建了各种设备和编程语言,后因差异增加了开发成本又发明了虚拟机。以虚拟机技术为核心的跨平台语言是未来趋势,as3 和 Java 都是基于虚拟机的跨平台语言。

1.2 牵手 ActionScript 3

as3 是近几年诞生的语言,新诞生的语言必定会吸收以往优秀语言的特点,总体来说 as3 是一种面向对象、跨平台、功能强大、开发效率极高且易学易用的语言,as3 的设计参照了 Java 和 C#, 特点如下。

- 符合 ECMAScript 4 标准(具有上述优点的语言标准),与 Java 相比,as3 开始就没有打算做一个所有领域都擅长的选手,而是瞄准了 Web 市场,它规避了 Java 语言中一些繁冗的概念,将之简化又不失其强大,一旦你接触 as3, 马上就会喜欢上它。
- as3 是一门跨平台语言,运行 as3 的虚拟机有两个:一个是在浏览器中安装的 Flash Player 插件,我们在网页中欣赏 Flash 动画,玩游戏、看视频都是 Flash Player 在表演。在桌面上执行的 Flash 程序运行在 AIR (Adobe Integrated Runtime) 虚拟机中,手机、平板电脑上的程序都以这个方式运行,如图 1-8 所示。

不必担心客户机器上没有安装虚拟机,如今 Flash Player 市场占有率已高达 98%,即使没装也会自动连网安装。

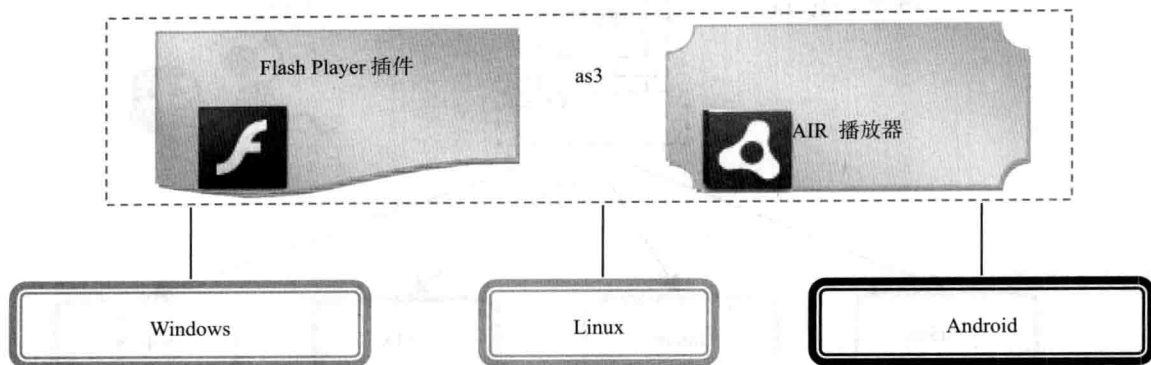


图 1-8 as3 的虚拟机

- as3 可以独立编译，也可以作为动画脚本运行，使用的开发工具有擅长动画制作的 Flash CS6 和擅长程序开发的 Flash Builder，将美术和程序结合开发的语言不止 as3 一种，但 as3 是其中的佼佼者。

1.2.1 使用 Flash CS6 开发第一个程序

在学习 as3 语言之前，我们必须掌握开发工具的使用，as3 和支持它的原生工具 Flash CS6 是密不可分的，Flash 软件起初是 Macromedia 的产品，以一款矢量动画制作软件出名，它制作的矢量动画生动且体积小，因此迅速在 Web 上走红，几乎垄断了 Web 动画市场。Flash 动画还拥有互动脚本，Adobe 看到这个巨大的市场潜力后便收购了 Macromedia 公司，将 Flash 打造成具有交互性的多媒体平台，这个过程涵盖了 ActionScript 语言的几次重大变革，as 从辅助动画的次要角色摇身变成能和主流语言平起平坐的独立语言。今天 Flash CS6 仍然定位于矢量动画制作，但现在的 Flash CS6 功能比之前更加丰富，脚本也更加强大，让我们打开 Flash CS6 新建一个 ActionScript 3.0 文档，如图 1-9 所示。

在新建文档面板中可以制定舞台的宽、高及背景颜色，这里保持默认即可，单击“确定”按钮后，我们进入 Flash CS6 制作界面，如图 1-10 所示。

舞台是创作的平台，使用工具栏可以在舞台上绘制各种图形，时间轴可用来制作动画。按 F9 键可以打开动作面板，动作面板是编写脚本代码的地方，现在我们通过一个简单的例子来学习 Flash CS6 的使用，虽然这个过程更像在做一件美术人员的工作。

step 1 使用椭圆工具在舞台上画一个圆表示太阳，选中太阳，在“修改”菜单中找到“形状→柔化填充边缘”，如图 1-11 所示。在“柔化填充边缘”对话框中距离和步长都设置为 10，单击“确定”按钮后可以看到太阳发光的效果，如图 1-12 所示。