

21世纪高等教育计算机规划教材

COMPUTER

ActionScript 3.0 语言基础与应用

ActionScript 3.0 Programming

俞淑燕 王海庆 编著

— 阐述素材制作的技巧

— 讲解面向对象的编程思维

— 分析数据视图设计模式



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

ActionScript 3.0 语言基础与应用

ActionScript 3.0 Programming

■ 俞淑燕 王海庆 编著



人民邮电出版社

北京

图书在版编目 (CIP) 数据

ActionScript 3.0 语言基础与应用 / 俞淑燕, 王海庆
编著. — 北京: 人民邮电出版社, 2014.12
21世纪高等教育计算机规划教材
ISBN 978-7-115-37630-5

I. ①A… II. ①俞… ②王… III. ①动画制作软件—
高等学校—教材 IV. ①TP391.41

中国版本图书馆CIP数据核字 (2014) 第307254号

内 容 提 要

本书依据 ActionScript 3.0 语言的知识结构与开发特点, 将内容分为基础篇、提高篇、实战篇三部分, 共 10 章, 主要包括 ActionScript 3.0 语言的基础知识、面向对象与类、对象交互与事件处理、视觉编程、外部数据处理、第三方类库的使用、网站应用、游戏开发与移动应用开发等内容。

本书适合有一定计算机语言基础、从事交互设计的数字媒体开发人员或软件开发的专业人员使用, 也适用于动画爱好者提升动画编程能力。

-
- ◆ 编 著 俞淑燕 王海庆
责任编辑 戴思俊
责任印制 沈 蓉 彭志环
 - ◆ 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>
大厂聚鑫印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 19.5
字数: 512 千字
-

2014 年 12 月第 1 版
2014 年 12 月河北第 1 次印刷

定价: 46.00 元

读者服务热线: (010) 81055256 印装质量热线: (010) 81055316
反盗版热线: (010) 81055315

前言

ActionScript 3.0 语言是一种基于 ECMAScript 的编程语言，其设计意图是创建一种适合快速构建效果丰富的互联网应用程序的语言。事实表明，ActionScript 3.0 因其高度的互动性、良好的用户体验以及强大的功能，已经成为构建 Rich Internet Application 的理想语言。在开发用户互动类产品、触摸系统、网络游戏、手机游戏等方面，ActionScript 3.0 都具有很好的表现。eBay AIR 拍卖程序、Google Analytics 以及微软的 Wallop 社区，都是基于 ActionScript 开发的典型案例。

本书作者从 2009 年开始接触 ActionScript 3.0，开发了很多应用项目。正是在边学边用的过程中，发现了 ActionScript 3.0 在互动开发中无法比拟的优越性，进而把它引入到教学中，培养了不少从事游戏开发和前端开发的工程人员。随着目前用户对体验要求的提升以及游戏行业的发展，越来越需要 AS3 开发人员，AS3 学习者也日渐增多，为此，我们编写了此书，目的是想把我们在项目开发以及教学过程中积累的心得和案例分享给 AS3 的学习者。

本书以面向对象思想为主轴，讲述 ActionScript 3.0 的基本语法和设计思想，并以恰当的案例去解释这些知识点，使基本理论与实践结合。本书根据 AS3 语言的知识结构，结合 AS3 开发形成的特点，将内容分为基础篇、提高篇和实战篇。为便于学习和使用，本书所有案例都有配套源代码，本书中加“*”部分内容稍难，如果初学不理解可以先跳过。读者在使用过程中如果有任何疑问可以通过邮件联系：ysy@zptc.cn。

感谢曾经一起参与过 AS3 开发和教学的同事，这本书中也有你们的智慧。由于作者水平有限，书中如有疏漏和不足，敬请广大读者朋友批评指正。

作 者
2014 年 12 月

目 录

第 1 篇 基础篇

第 1 章 ActionScript 3.0 概述.....1

1.1 Flash 应用现状及前景.....1

1.2 ActionScript 3.0 简介.....4

1.3 第一次与 ActionScript 3.0 亲密接触.....5

1.3.1 在 Flash CS6 中创建.....5

1.3.2 在 Flash Builder4.7 中创建.....8

1.4 模拟小球运动.....11

1.4.1 模拟小球运动（一）.....11

1.4.2 模拟小球运动（二）.....12

1.5 本章小结.....12

第 2 章 ActionScript 3.0 语言基本元素.....13

2.1 基础语法.....13

2.1.1 点语法.....13

2.1.2 分号.....13

2.1.3 括号.....14

2.1.4 注释.....15

2.2 数据表现及运算.....16

2.2.1 变量和常量.....16

2.2.2 数据类型.....18

2.2.3 运算符和表达式.....27

2.2.4 图片三维展示.....30

2.3 流程控制.....32

2.3.1 条件语句.....32

2.3.2 循环语句.....34

2.3.3 立体字.....36

2.4 应用函数.....37

2.4.1 函数的声明.....37

2.4.2 函数的参数.....38

2.4.3 函数的妙用.....40

2.4.4 粒子喷泉.....42

2.5 常用核心类.....43

2.5.1 数组.....44

2.5.2 抽奖.....47

2.5.3 日期和时间.....49

2.5.4 模拟时钟.....51

2.5.5 文本.....51

2.5.6 计算器.....55

2.6 本章小结.....58

第 3 章 面向对象与类.....59

3.1 面向对象和面向过程.....59

3.2 创建和使用类.....60

3.2.1 类和对象.....60

3.2.2 创建类.....60

3.2.3 使用类.....61

3.3 类和成员的访问控制.....62

3.3.1 构造函数.....62

3.3.2 this 关键字.....62

3.3.3 属性.....63

3.3.4 方法.....64

3.3.5 访问控制.....64

3.4 使用类模拟小球运动.....64

3.4.1 创建单个小球并运动.....64

3.4.2 创建多个小球并运动.....66

3.5 复合与继承.....69

3.5.1 复合.....69

3.5.2 继承.....72

3.6 *多态.....75

3.7 模拟打怪.....76

3.7.1 怪物角色.....76

3.7.2 模拟打怪游戏.....79

3.8 本章小结.....82

第 4 章 对象交互与事件处理.....83

4.1 ActionScript 3.0 事件模型.....83

4.2 使用事件侦听器	84	4.8.1 小车运动 (一)	92
4.3 事件 (Event 类)	85	4.8.2 小车运动 (二)	94
4.4 单击画圈	86	4.9 控制小球缩放	95
4.5 鼠标事件	87	4.10 *自定义事件	97
4.6 小球拖曳	90	4.11 *升级打怪游戏	102
4.7 键盘事件	91	4.12 *消息输入与输出	106
4.8 模拟小车运动	92	4.13 本章小结	110

第 2 篇 提高篇

第 5 章 视觉编程	111	第 6 章 外部数据处理	159
5.1 显示对象模型	111	6.1 使用 Loader 加载外部素材	159
5.2 核心显示对象类继承关系	112	6.1.1 使用 Loader 加载图片文件	159
5.3 显示对象类	113	6.1.2 使用 Loader 加载 SWF 文件	162
5.4 显示对象容器类	114	6.2 制作图册	164
5.5 显示对象切换	117	6.2.1 制作图册 (一)	164
5.6 矢量图	118	6.2.2 制作图册 (二)	165
5.7 绘制矢量图	121	6.3 使用 URLLoader 加载外部数据	166
5.7.1 绘制矩形	121	6.3.1 以文本格式接收数据	166
5.7.2 能绘制的毛笔	122	6.3.2 以二进制格式接收数据	169
5.8 位图	123	6.4 使用 navigateToURL 打开网址	170
5.8.1 Bitmap 和 BitmapData	123	6.5 单击网址打开网页	172
5.8.2 外部图片的使用	125	6.6 使用 XML 数据	173
5.8.3 位图操作	126	6.6.1 重要概念和术语	174
5.9 应用位图操作	129	6.6.2 创建 XML 对象	175
5.9.1 切割位图	129	6.6.3 操作 XML 数据	177
5.9.2 踢足球	132	6.7 绘制星星	182
5.10 遮罩	134	6.8* 异常捕获与处理	184
5.11 动态遮罩	137	6.8.1 同步异常处理	184
5.12 事件流机制	140	6.8.2 异步异常处理	187
5.12.1 事件流机制的三个阶段	141	6.9* 模拟计算器	188
5.12.2 事件流机制案例	142	6.10* 安全沙箱	191
5.12.3 案例思考	145	6.10.1 跨域文件: crossdomain.xml	193
5.13 动画编程	147	6.10.2 授予脚本访问权限	194
5.13.1 利用 Event.ENTER_FRAME 事件 实现动画	147	6.11 图片轮播	194
5.13.2 利用 Timer 类实现动画	148	6.12 本章小结	197
5.14 下雪效果	150	第 7 章 第三方类库的使用	198
5.15 人物走动	153	7.1 Tween 类	198
5.16 本章小结	158	7.1.1 Flash Builder 导入 fl.transitions 包	198
		7.1.2 Tween 类用法	198

7.1.3 TransitionManager 类的用法	201	7.4.2 创建一个有刚体的 Box2D 世界	213
7.2 TweenLite 类	202	7.4.3 继续添加刚体	220
7.2.1 下载并安装 greensock 包	202	7.4.4 EasyBox2D 使用	221
7.2.2 TweenLite 类使用	203	7.4.5 创建有皮肤的刚体	225
7.3 应用 TweenLite 制作特效	207	7.4.6 操作刚体	227
7.3.1 鼠标跟随效果制作	207	7.4.7 不规则刚体的创建	229
7.3.2 图册制作	210	7.5 发射小鸟	238
7.4* Box2D 物理引擎	212	7.6 本章小结	242
7.4.1 下载并安装 Box2D	213		
第 3 篇 实战篇			
第 8 章 网站应用	243	9.4 举一反三	271
8.1 作品分析	243	9.5 本章小结	271
8.2 制作思路	245	第 10 章 移动应用	272
8.2.1 制作素材	245	10.1 作品分析	272
8.2.2 代码部分思路	245	10.2 制作思路	273
8.3 代码解析	246	10.2.1 制作播放器界面影片剪辑	273
8.4 举一反三	249	10.2.2 创建 ActionScript 手机 AIR 项目	274
8.5 本章小结	249	10.2.3 初步的视图操作	275
第 9 章 游戏开发	250	10.2.4 数据操作	276
9.1 作品分析	250	10.2.5 进一步的视图操作	277
9.2 制作思路	251	10.2.6 Android 应用调试配置	278
9.2.1 制作游戏素材	251	10.2.7 发布 apk	280
9.2.2 代码部分整体思路	254	10.3 代码解析	284
9.2.3 角色类处理	254	10.3.1 GameView 类	284
9.2.4 游戏管理和控制	255	10.3.2 PlayerData 类	292
9.3 代码解析	255	10.3.3 SoundManager 类	293
9.3.1 Bullet 类	255	10.3.4 MyTimeTransform 类	296
9.3.2 Enemy 类	256	10.3.5 Main 类	296
9.3.3 Hero 类	258	10.4 举一反三	297
9.3.4 BulletManager 类	262	10.5 本章小结	297
9.3.5 EnemyManager 类	263	写在最后	298
9.3.6 GameView 类	265	附录	299
9.3.7 GameMap 类	266	参考文献	304
9.3.8 Game 类	270		
9.3.9 Main 类	270		

第 1 篇 基础篇

第 1 章

ActionScript 3.0 概述

Flash 发展到现在,已经远不是 Flash 8.0 时代的那个只做动画的 Flash,有了 ActionScript 3.0 语言的支撑,Flash 的应用领域更加广阔,小到课件,大到网游、企业的 ERP 前端等等。本章将让您初步领略一下 AS3 的世界。

1.1 Flash 应用现状及前景

Flash 和 HTML5 之争始于 2009 年,各界众说纷纭,争论得如火如荼。很多人认为 HTML5 的出现给 Flash 一个致命的打击,HTML5 完全可以达到 FlashPlayer10 的绚丽效果。就目前应用现状来说,HTML5 确实对 Flash 有一定的冲击,但是 Flash 在富媒体应用和网页游戏方面还是表现不凡的。Flash Player 已经不再是 Flash 内容的唯一载体。对于桌面浏览器,Flash Player 仍然占据统治地位,且特性不断增强;对于移动设备(iOS,Android,BlackBerry 等等),虽然没有 Flash Player 支持,但有 AIR 同样可以让我们的 Flash 内容进入到移动设备。Android 系统用户可以安装 AIR 运行时(Runtime);开发者也可以将 AIR 运行时(Runtime)和 APK 捆绑进行打包(此时用户可以在设备上不安装 AIR)。对于 iOS 系统,因为不允许运行时的存在,我们的 Flash 内容会被 ADT 编译为 iOS 本地代码来执行。

多媒体课件、小游戏、虚拟现实和电子杂志等,Flash 在视觉以及体验上有着无可比拟的优越性,而在网页游戏市场,更是几乎被 Flash 占据。多点触摸、人脸识别、Socket 通信同步、AR 增强现实、实时视频等功能在 Flash 平台上的实现,让越来越多的互动多媒体项目采用 Flash 方案。

来看看 Flash 应用的案例。

1. 网站元素

一些网页中的小广告、展示类动画、企业的宣传片头等,很多都使用 Flash 制作,如图 1-1、图 1-2 所示。



图 1-1 网页小广告

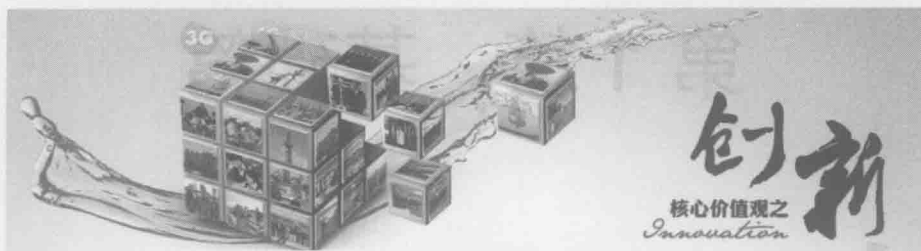


图 1-2 企业网站片头

2. 动画 MTV、MV

这一类型的作品很多，像一些娱乐类节目的片头，比如《爸爸去哪儿》的片头；歌手的专辑，比如比较早之前雪村、周惠等的歌；很多的儿歌和动画片，比如《喜羊羊与灰太狼》《大头儿子小头爸爸》《水漫金山》等。

图 1-3 是酷迷网的部分截图，从中可以看到 Flash 的很多应用，如动画片、大电影、动画 MV、游戏动画、儿歌和小游戏等。



图 1-3 酷迷网截图

3. Flash 游戏

小游戏自不必说，诸如“7k7k”“3366”，还有腾讯的小游戏，QQ 空间的一些应用，很多都应用 Flash 制作，比如说“掏金子”是一款比较早的 Flash 游戏，再比如“2048”、“围住神经猫”、“切水果”等。不少网页游戏、网络游戏的客户端也是由 Flash 来完成，比如“神仙道”。图 1-4 和图 1-5 分别为一些小游戏和网页游戏。



图 1-4 Flash 应用于小游戏开发



图 1-5 Flash 应用于页游开发

4. 宣传片

Flash 也用来制作一些公益广告宣传片，企业、产品展示宣传片，如图 1-6 和图 1-7 所示。



图 1-6 廉政公益广告



图 1-7 楼盘展示广告

5. 多媒体课件、儿童益智类学习材料

Flash 因其较好的表现形式，在课件中应用很广，尤其是低幼阶段的动画产品特别多。图 1-8 所示的是宝宝吧中部分动画产品截图。



图 1-8 宝宝吧部分截图

Flash 的优点在于如下几个方面。

(1) 交互效果、视觉效果好，且易于开发。

(2) 普及率高。Adobe 公司声称全世界 97% 的网络浏览器都内置 Flash 播放器 (Flash Player)。

(3) Flash 的开发群体也很庞大，十多年的积累，有众多的开发人员和设计人员在 Flash 平台上做互动媒体开发，累积了众多的类库和引擎，如 Box2D、Away3D 等。

(4) 对浏览器的依赖性小，在 PC 上的可移植性强。Flash Player 的更新只需要升级插件。

当然，Flash 也有其自身的缺点，如不开源、收费等。无论作为开发者也好，用户也好，我们欢迎各大技术之间的争议，只有保持这样的争议，各种技术才会有不断的努力和更新，我们才能够享受技术进步带来的高效、便捷，以及更好的用户体验。

1.2 ActionScript 3.0 简介

ActionScript 语言版本经历了 1.0 和 2.0，目前最新版本是 3.0，其与早期版本有着显著的不同，ActionScript 3.0 (AS3) 是一种完全面向对象的编程语言，功能强大，类库丰富，语法与 JavaScript 类似，多用于 Flash 交互性应用开发、网页制作和 RIA 应用程序开发。ActionScript 3.0 使用新型虚拟机 AVM2，使得性能得到显著改善，ActionScript 3.0 代码的执行速度可以比旧式 ActionScript 代码快 10 倍。

ActionScript 语言编写的程序，最终可以编译成 SWF、SWC 格式。SWF 是我们常说的 Flash 动画，它是 Flash 平台的最终表现形式。但是现在 SWF 已经不仅仅是动画，而是 RIA 的载体。SWC 是一种库文件，通常用来发布非开源的类。程序员们将源码编译成 SWC 文件，并发布 API 文档，我们就可以使用这些 SWC 提供的类库进行自己的开发。SWC 并不是运行时共享的，而是在编译的时候直接加入了 SWF 程序中。最终的程序 SWF 可以脱离 SWC 运行。

AS3 的开发环境有 Flash Professional 和 Flash Builder。目前最新的零售版本为 Adobe Flash Professional CC (2013 年发布)，从 CS3 版本开始支持 AS3。

Flash Professional 可以认为是一个创作环境，适合设计人员进行操作。用 Flash Professional 可以生产 SWF，可以在 FLA 文件内部写 AS3 代码，也可以在 FLA 外部写 AS3 代码。

而 Flash Builder 是一个针对程序员来说的开发环境 (IDE)，它由 Flex Builder 升级而来，Flex Builder 从 4.0 开始，被 Adobe 改名为 Flash Builder。在 Flash Builder 中包涵了更多、效果更好、功能更完善的组件。使用 Flash Builder 可以快速高效地创作出 RIA 程序。

Flash Player 和 AIR 是 Flash 的运行环境 (Runtime，部分文献资料也译成“运行时”)。

Flash Player 是 SWF 基于网页的运行环境。Flash Player 9 是能支持 AS3 的最低版本 (其实前面还有一个 Flash Player 8.5，只是没普及就升级到 9 了)，后面的 Flash Player 都能支持 AS3 了。

AIR 是基于桌面的运行环境。有了 AIR，就可以用 AS3 开发桌面软件，而且开发出的软件可跨平台运行 (这和 JAVA 的一次编译，随处运行如出一辙)。目前国内的 AIR 应用主要集中在网站的客户端方面，例如新浪微博 AIR 客户端等。

1.3 第一次与 ActionScript 3.0 亲密接触

我们可以选择 Flash Professional 或者 Flash Builder 作为 AS3 的开发平台。本节中分 Flash CS6 和 Flash Builder 4.7 两部分来介绍。读者可以根据实际，选择其中之一作为开发平台。本书中，AS3 的语言基础部分使用 Flash CS6，在初步认识和熟悉 ActionScript 3.0 语言以及面向对象之后过渡到 Flash Builder 4.7 开发环境。关于平台的安装以及操作界面介绍，我们在这里不再赘述。下面仅以一个小球运动来简单接触一下 AS3。

1.3.1 在 Flash CS6 中创建

在 Flash CS6 中有两种写入代码的方式：一种是在时间轴中写入；另一种是在新建的 ActionScript 文件中写入，并与相关的库元件绑定，或者直接与 FLA 文件绑定。

1. 在时间轴中写入代码

打开 Flash CS6 之后，选择“新建 ActionScript 3.0”，进入编辑界面，这就是 fla 文件，如图 1-9 所示。

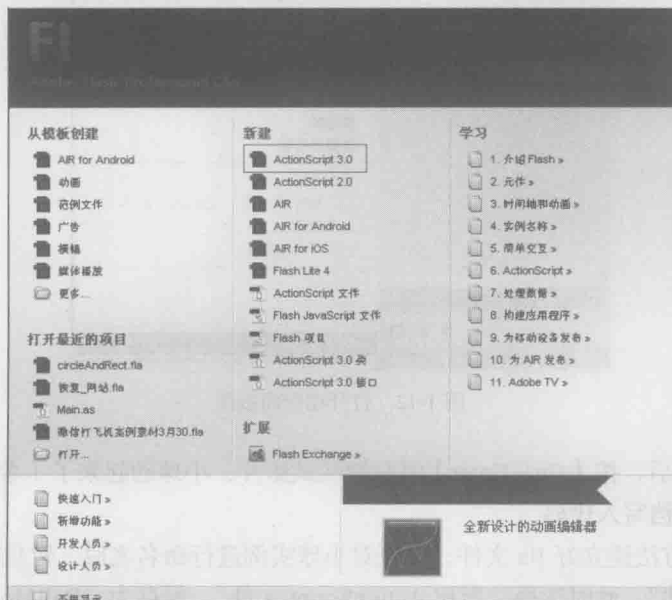


图 1-9 新建基于 ActionScript 3.0 的 fla 文件

建好 fla 文件后，新建一个椭圆作为小球，并将其转为元件，取名“Ball”，如图 1-10 所示。确定之后，对舞台上的小球实例进行命名，取名“ball”，在小球的属性面板上设置，如图 1-11 所示。

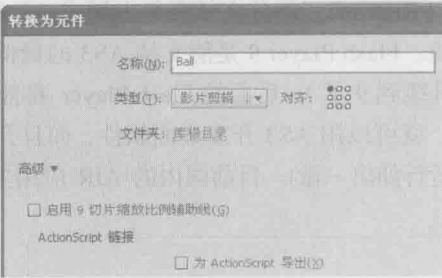


图 1-10 将小球转换为元件

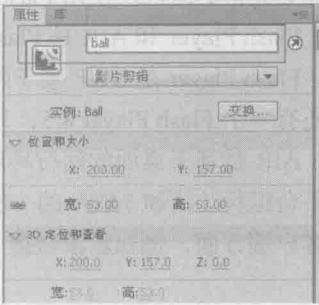


图 1-11 为实例命名

关于元件和实例的概念，熟悉 Flash 工具的同学应该清楚，涉及与 AS3 相关的，我们将在第 3 章“面向对象和类”中进行具体阐述。单击主时间轴的第一帧，按【F9】键弹出动作面板，如图 1-12 所示，在其中写入代码：

```
ball.addEventListener(Event.ENTER_FRAME,onLoop);
function onLoop(e:Event):void{
    ball.x += 5;
}
```

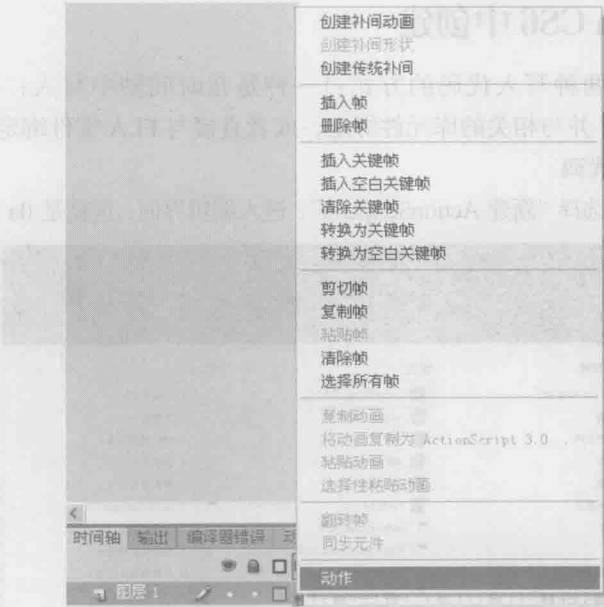


图 1-12 打开动作面板图

保存好 fla 文件后，按【Ctrl+Enter】组合键测试影片，小球动起来了（参见案例 1.1）！

2. 新建 AS 文档写入代码

按上一小节的方法建立好 fla 文件，并且对小球实例进行命名之后，暂且将 FLA 文件放置一边。再次单击“新建”，此时选择“新建 ActionScript 文件”，保存为“BallMoving.as”，和刚建立的 fla 文件放在同一个目录中，如图 1-13 所示。

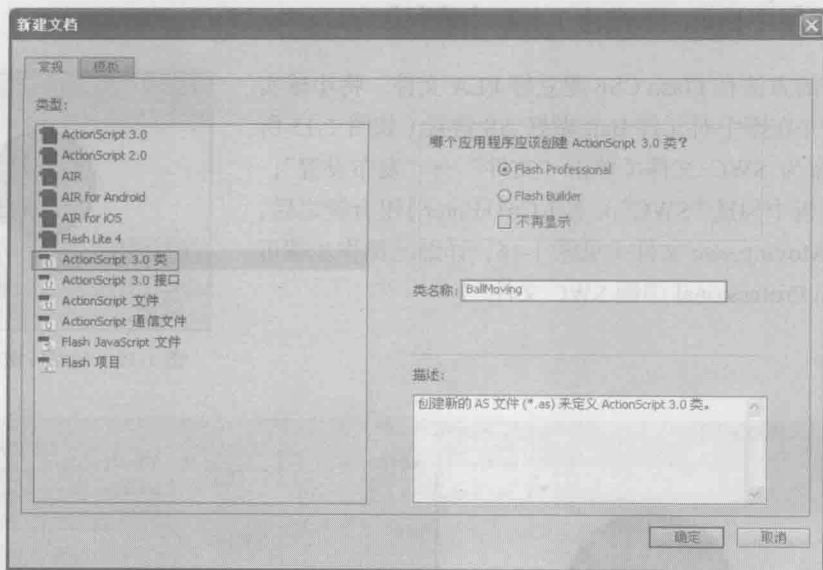


图 1-13 创建 as 文件

在“BallMoving.as”中写入代码：

```
package
{
    import flash.display.MovieClip;
    import flash.events.Event;
    public class BallMoving extends MovieClip
    {
        public function BallMoving()
        {
            ball.addEventListener(Event.ENTER_FRAME, onLoop);
        }
        function onLoop(e:Event):void
        {
            ball.x += 5;
        }
    }
}
```

保存之后，切换到 FLA 文件，在舞台属性面板中，选择文档类输入框，写入“BallMoving”，如图 1-14 所示，单击右侧的“编辑类定义”，如果文件正确，会直接链接到 as 文档“BallMoving”。

按【Ctrl+Enter】组合键测试影片，小球也动起来了！（参见案例 1.2）

在这个作品中，我们就是将 fla 文档和 as 文档通过类绑定的方式使代码发生作用，其中的 as 文档称为主文档类。绑定就是通过“文档类输入框”中输入外部类文件的相对路径加文件名。而实际执行的入口是主文档类中 function 中 BallMoving 的内容，这个 function 称为构造函数。

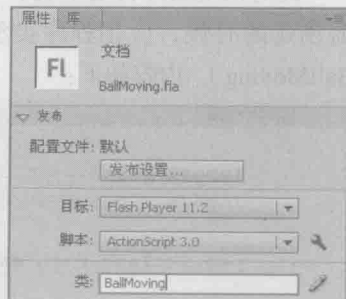


图 1-14 文档类设定

1.3.2 在 Flash Builder4.7 中创建

按上一节的方法在 Flash CS6 建立好 FLA 文件，将小球实例做好命名，并在库中对元件 Ball 做好 AS 链接（如图 1-15 所示），将其发布为 SWC 文件（单击“文件”→“发布设置”，在发布设置面板中勾选“SWC”），按【Ctrl+Enter】组合键之后，你会发现 ballMoving.swc 文件（见图 1-16，详细的操作步骤可见附录“Flash Professional 中制 SWC 文件”）。



图 1-15 对元件做好 AS 链接

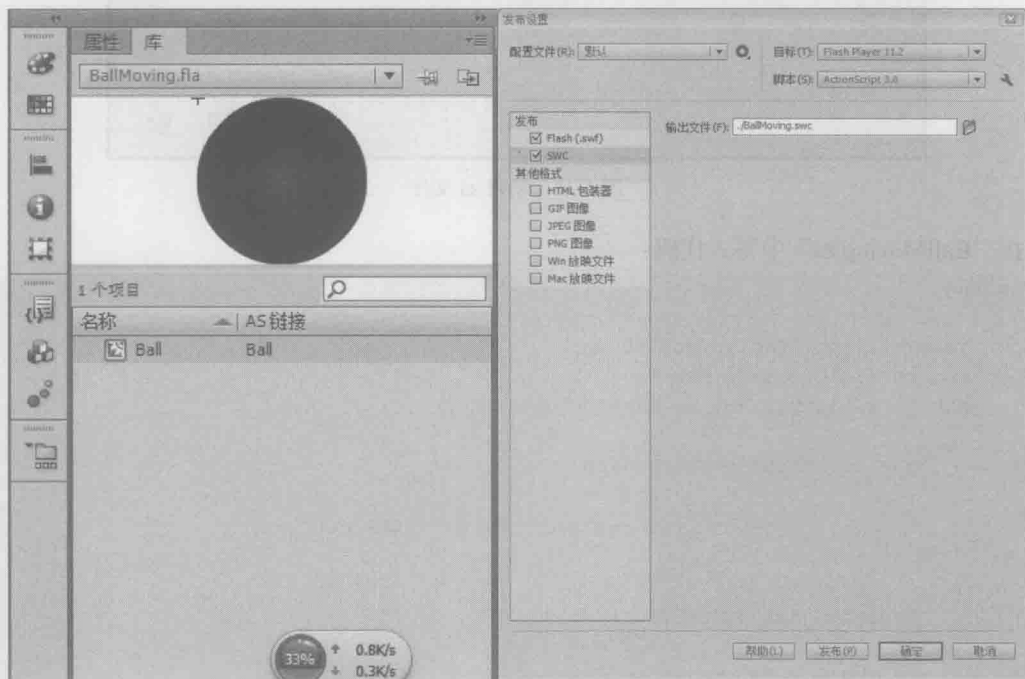


图 1-16 将 fla 文件发布为 SWC 文件

打开 Flash Builder 4.7，新建 ActionScript 项目（如图 1-17 所示），在新建对话框中输入项目名称“BallMoving”（如图 1-18 所示），指定项目存放位置，应用程序类型选择“桌面”，单击“完成”。

此时系统已经自动生成了一个 Main.as 文件，这个文件就是默认应用程序类文件（如果在之前创建的时候，应用程序类型选择“默认为 Web 方式”，则默认的应用程序类文件名称为 BallMoving）。内容如下：

```
package
{
    import flash.display.Sprite;
    import flash.events.Event;

    publicclass Main extends Sprite
    {
        publicfunction Main()
        {
```

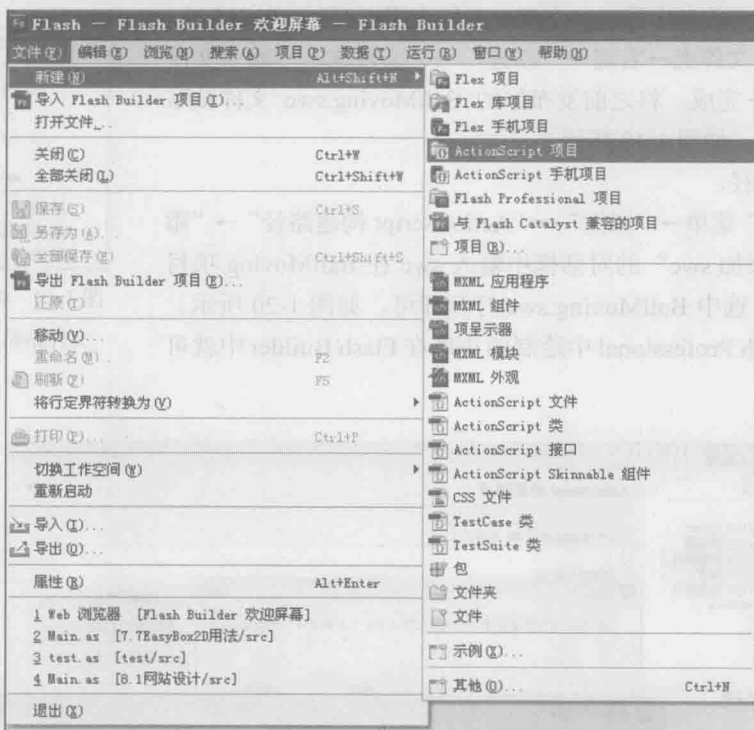


图 1-17 新建 ActionScript 项目

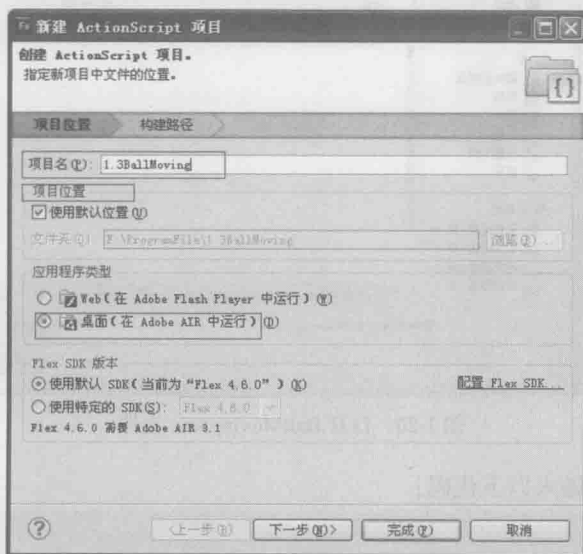


图 1-18 创建 ActionScript 项目

这个文档与上一节中用在 as 文档中写入代码时的文档类是类似的。实际上，Flash Builder 中使用 AS 项目时，默认的应用程序类和 Flash CS6 中的文档类是类似的，在 Flash Builder 中不需要

去绑定 fla 文件。

左侧的包资源管理中，以树状结构的视图来显示工程下面的所有包、文件夹、各类文件等。接下来我们把之前创建好的 SWC 文件导入到 Flash Builder 中来使用，方法如下。

首先在项目文件夹中建立一个 libs 文件夹用以存放 SWC 文件。

选中“src”文件夹→右键→“新建”→“文件夹”→输入文件夹名称“libs”→完成。将之前发布好的 BallMoving.swc 文件粘贴到 libs 文件夹中，如图 1-19 所示。

其次构建路径。

选中“项目”菜单→“属性”→“ActionScript 构建路径”→“添加 swc”。在“添加 swc”的对话框中输入 swc 在 BallMoving 项目中存放的位置，选中 BallMoving.swc 打开即可，如图 1-20 所示。这样我们在 Flash Professional 中绘制的小球在 Flash Builder 中就可以用了。

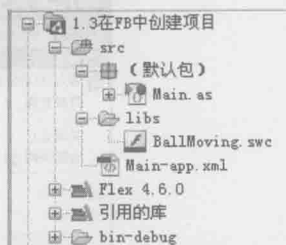


图 1-19 将 BallMoving.swc 文件粘贴到 libs 文件夹中

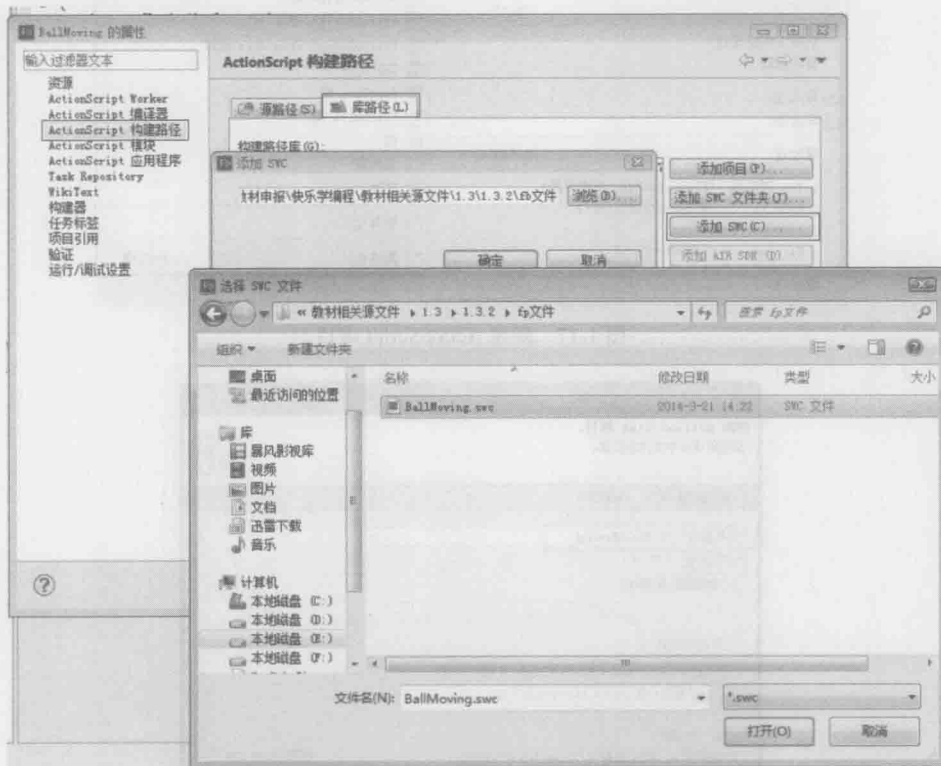


图 1-20 打开 BallMoving.swc 文件

在代码编辑窗口中插入以下代码：

```
package
{
    import flash.display.Sprite;
    import flash.events.Event;

    public class Main extends Sprite
    {
```