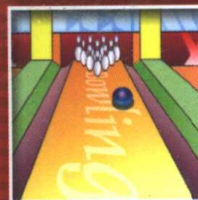


Flash MX

网络游戏解码书

——写Game其实很Easy

MX



朱志山 王大任
飞思科技产品研发中心

著
改编

本书所附光
盘内容为书
中所有范例
的完成文件



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

为设计师量身定制

飞思 数码设计院

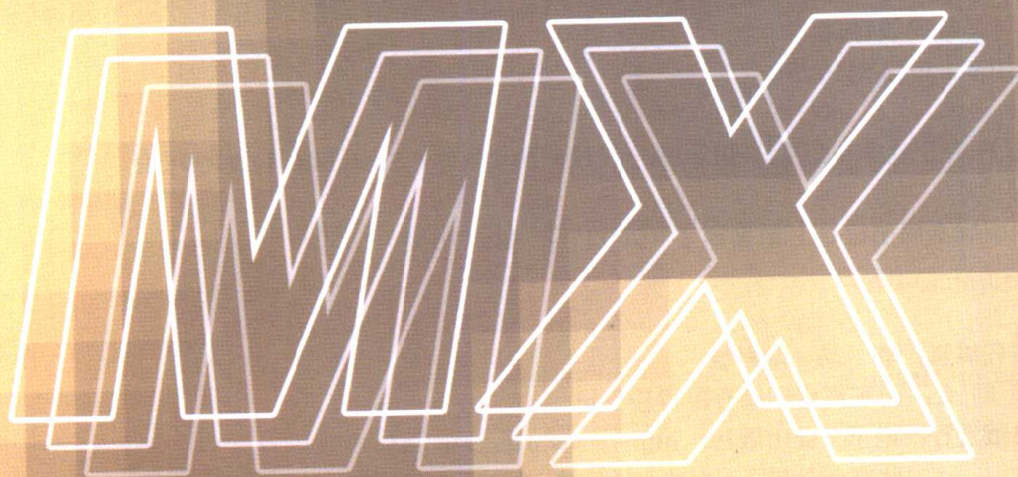
Flash MX

网络游戏解码书

——写Game其实很Easy

朱志山 王大任
飞思科技产品研发中心

著
改编



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是作者根据多年从事网站设计、电子游戏制作的经验，结合具体实例写成的。书中用最快的方式让读者熟悉 ActionScript，并从制作最简单的小游戏开始，讲解创作思路，解析该类型游戏制作的关键环节，再用 Step by Step 的方式示范整个游戏的制作流程。不管您以往对 Flash 了解有多少，都能轻松进入 Flash Game 的世界中。

本书由浅入深地介绍了多种类型的游戏，所有步骤都会一一呈现给您。您会发现大家最头痛的 ActionScript，原来是我们创作过程中最大的帮手。本书完全以实例为引导，绝不是呆板的语法书。当您完成本书所有范例时，您将不再视写程序为畏途，并将拥有绝对足够的技术，帮您实现完成任何 Flash Game 的梦想！本书所附光盘内容为书中所有范例的完成文件。

本书繁体版图书原名为《Flash MX 网络游戏解码书——写 Game 其实很 Easy》，由上奇（中国）有限公司授权出版，版权归上奇（中国）有限公司所有。本书简体中文版授权电子工业出版社出版。专有出版权属于电子工业出版社所有，未经本书原出版者和本书出版者书面许可，任何单位和个人均不得以任何手段复制或抄袭本书的部分或全部内容。

版权贸易合同登记号：01-2002-5474

图书在版编目（CIP）数据

Flash MX 网络游戏解码书——写 Game 其实很 Easy / 朱志山，王大任著. —北京：电子工业出版社，2003.1

（飞思数码设计院）

ISBN 7-5053-8110-5

I.F... II.①朱...②王... III.动画—设计—图形软件, Flash MX IV.TP391.41

中国版本图书馆 CIP 数据核字（2002）第 086176 号

责任编辑：郭 晶 陆舒敏

印 刷 者：山东高唐印刷有限责任公司

出版发行：电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编：100036

经 销：各地新华书店

开 本：787 × 1092 1/16 印张：17 字数：435.2 千字 附光盘 1 张

版 次：2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

印 数：5000 册 定价：39.00 元（含光盘）

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077

前言

关于本丛书

21 世纪是数字多媒体的时代，其中备受瞩目的应该就是计算机图形设计技术。当前的文化艺术领域也在广泛利用计算机图形设计功能，最典型的例子就是曾经风靡全球的电影《泰坦尼克号》。现在，计算机图形设计已经是很多领域中必不可少的一项技术。而每当面对国外著名设计师的精彩设计时，我们只能惊叹其构思的巧妙和设计的精美。那么什么时候我们能够赶上世界一流的水平？这一点需要我们认真地反省！而出版高品质的电脑图形设计书籍是培养出世界级著名设计师的基础。

综观当前整个电脑图形技术书籍市场，我们会发现这片领域已经是相当成熟，各类图形设计技术的书籍琳琅满目。不过同时我们也发现大部分书籍只是停留在基础的层次上，对于有一定基础知识的读者来说，要想提高到更高的层次，就会遇到一个难以突破的瓶颈。

“飞思数码设计院”是电子工业出版社计算机研发部全新规划的、面向国内电脑图形设计爱好者量身定做的一套中高级电脑图形设计精品图书。本系列图书精心组织了国内优秀设计人员编写，力求保证图书的高质量，同时还将选择一些在电脑图形设计领域具有领先技术的日本、韩国和中国台湾等国家和地区的最畅销的精品图书进行改编。“飞思数码设计院”将兼顾电脑图形设计技术的深度和广度，让本系列图书成为电脑图形技术领域的精品，使其拥有更加长久的生命力，以开创高、精、尖电脑图形设计的全新理念。

我们真诚希望“飞思数码设计院”系列丛书可以为更多读者带来广阔的学习空间，并希望我们的努力能够为国内的设计师队伍建设做出一些贡献。我们期待着您能为我们的努力提出宝贵的意见，同时，我们也在等待着您的加入。

关于本书

Flash 是一个令人惊奇的创作工具！您可以利用它制作出各种动画，或是利用它让您的网页充满绚丽而动听的声光效果。另外，利用 Flash 作品的高度流通性，加上简单但可塑性极高的 ActionScript 程序，我们还可以创作出各种各样的游戏。

只要您有 Flash MX 的基础，做一个 Flash Game 不会太难，但是也许往往不知如何开始。书中先用最快的方式让您熟悉 ActionScript，并通过简单的小游戏，讲解创作的理念，解析该类型游戏的制作关键。然后再用 Step by Step 的方式，示范游戏的制作全程，不管

您以往对 Flash 了解多少，都能轻易进入 Flash Game 的世界中。

本书由浅入深介绍了多种类型游戏的所有制作步骤。您会发现大家最头痛的 ActionScript，原来是我们创作过程中最大的帮手。本书完全以实例为导向，绝不是呆板的语法书，让您不再视写程序为畏途。当完成本书所有范例时，您将拥有丰富的技术，进一步实现完成任何 Flash Game 的梦想！本书所附光盘内容为所有范例的完成文件。

本书具有以下特色：

- 全书以范例做引导，学习和务实并重，绝对不是呆板的语法书，让您不再视写程序为畏途。
- 所有范例皆为专业的 Flash Game，让读者直达专家殿堂。
- 书中范例涵盖各种类型游戏，让您成为全方位的高手。
- 内容循序渐进，按难易度讲解范例，让您的学习一路顺风。
- 以有趣的 Flash Game 为范例，轻松学习 ActionScript。
- 由基础的 ActionScript 讲起，连初学者都能够轻松上手。
- 以 Step by Step 的方式示范整个游戏的制作流程，使初学者也能轻松进入 Flash Game 的世界中。
- 读者可至作者网站，和作者进行最直接的讨论。

如有任何不足之处，欢迎读者批评指正。我们的联系方式为：

电 话：(010) 68134545 68131648

电子邮件：support@fecit.com.cn

飞思在线：<http://www.fecit.com.cn> <http://www.fecit.net>

答疑网址：<http://www.fecit.com.cn/question.htm>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

飞思科技产品研发中心

目 录

第一章 ActionScript 详解

1-1 认识 ActionScript	2
1-2 ActionScript 的保留字与变量	4
1-3 条件式与循环	8
1-4 数组	12
1-5 影片片段对象	17
1-6 音效对象	24
1-7 数学函数表达式	29
1-8 事件	31

第二章 猜拳游戏

2-1 规划游戏流程	34
2-2 准备素材	35
2-3 角色的制作	37
2-4 建立场景	46
2-5 程序编写	53
2-6 游戏试玩	58

第三章 九宫拼图

3-1 进行准备动作	61
3-2 影片、影片片段与实体	66
3-3 加入贴齐网格线功能	71
3-4 难度再提升	79

第四章 猜数字

4-1 准备工作及设计重点	86
---------------------	----

4-2 产生一组不重复的四位数字	87
4-3 数字影片的动作处理	91
4-4 输出对比结果	95
4-5 显示成功或失败画面	99

第五章 ICQ PONG

5-1 准备场景	104
5-2 游戏流程	108
5-3 球影片片段的设计	110
5-4 ICQ 地鼠	116
5-5 主时间轴程序设计	118

第六章 猫狗大战

6-1 准备场景及游戏组件	124
6-2 游戏流程	127
6-3 罐头程序设计剖析	128
6-4 其他影片程序剖析	133
6-5 起始画面与测试	135

第七章 乐透模拟器

7-1 规划游戏流程	138
7-2 准备素材	139
7-3 角色的制作	140
7-4 建立场景	145
7-5 程序编写	151
7-6 测试游戏	163

第八章 打击魔鬼

8-1 规划游戏流程	166
8-2 准备素材	167
8-3 角色的制作	172

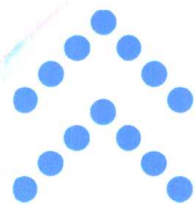
8-4	场景安排	182
8-5	程序编写	185
8-6	测试游戏	195

第九章 爆破砖块

9-1	规划游戏流程	198
9-2	游戏素材与程序结构	200
9-3	产生起始砖块	203
9-4	产生新砖块	204
9-5	砖块的自我意识	211
9-6	寻找同色相连的砖块	213
9-7	炸弹执行程序	218
9-8	结论与注意事项	219

第十章 保龄球 Online

10-1	规划游戏流程	222
10-2	准备素材	223
10-3	角色的制作	225



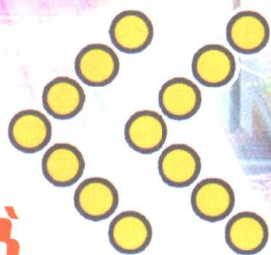
第一章

ActionScript 详解

Flash MX

网络游戏解码书

——写Game其实很Easy



ActionScript 在 Flash 5 推出时,就已完全符合面向对象编程的要求,大大提升了程序编写的能力。在 Flash MX 版本中, ActionScript 扩展了更多的功能。本章,将介绍编写程序时常用的 ActionScript。(至于详细的 ActionScript 的语法列表,请参阅光盘中的附录。)

1-1

认识 ActionScript

ActionScript 是什么

Script (描述程序语言,简称“程序”),在形式上及语法上是程序语言的架构,必须在特定的环境下才能执行,例如 JavaScript 可以在浏览器中执行,而 ActionScript 则必须在 Flash 播放程序中执行。

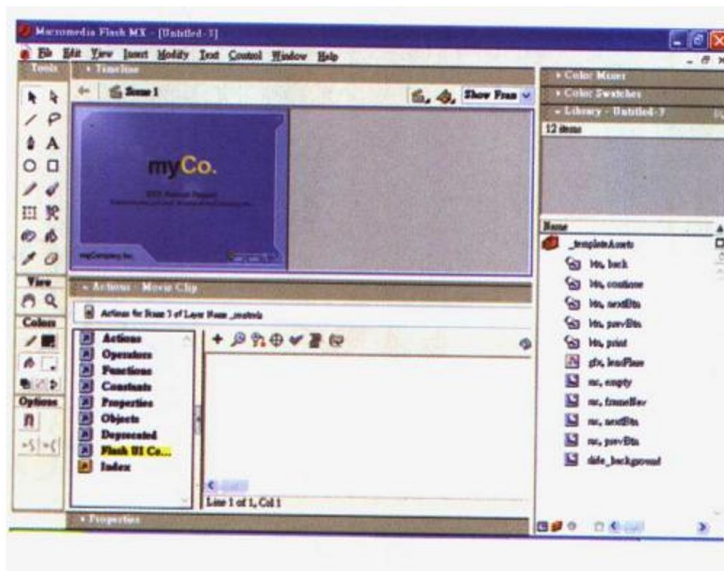
执行 Script 时,不需要 Compile (编译)及 Link (连接),如同前述,在特定环境中执行即可。ActionScript 则是 Flash 中使用的描述程序语言,如果熟悉 JavaScript 的读者,看到 ActionScript 后一定倍感亲切,因为由 Flash 5 版本起, ActionScript 就依照由欧洲计算机制造商协会 (European Computer Manufacturers Association) 制定的 ECMA-262 准则,也正是 JavaScript 依循的准则,例如 Dot Syntax (点语法) 的编写方式即是一例。

由于 ActionScript 指令相当多,要详细论述足以写一本书。本书将在编写游戏时所需的指令、对象、属性、方法及函数中加以说明,这些都是在编写程序时常用的功能。有几个对象是在编写游戏程序时,几乎是必备的项目,像影片片段对象等,以下将分别做说明。

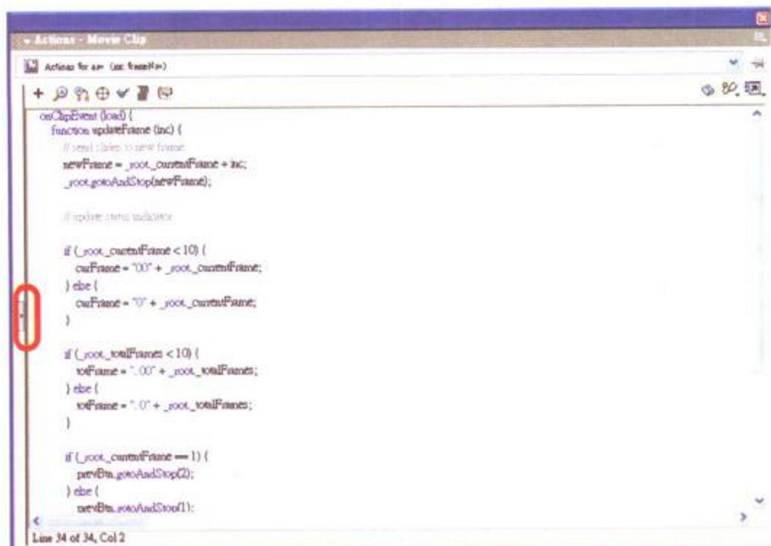
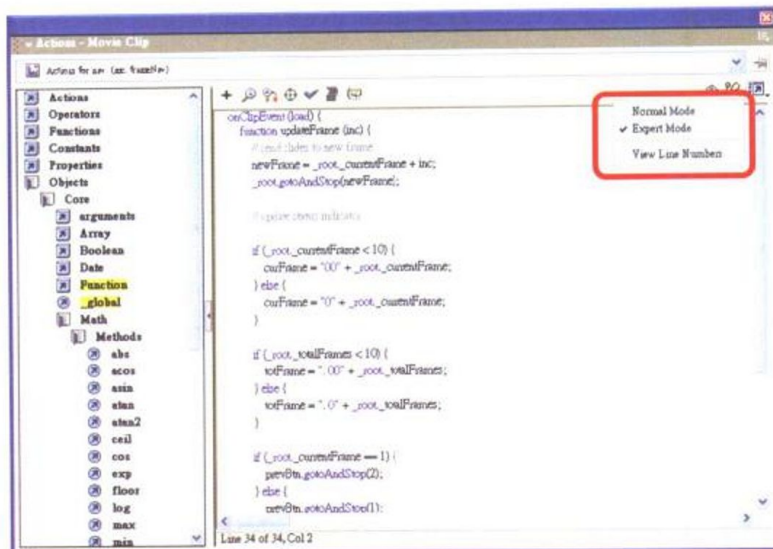
动作面板

与 Flash 5 相比较,Flash MX 的操作界面有大幅度改变,如右图所示。

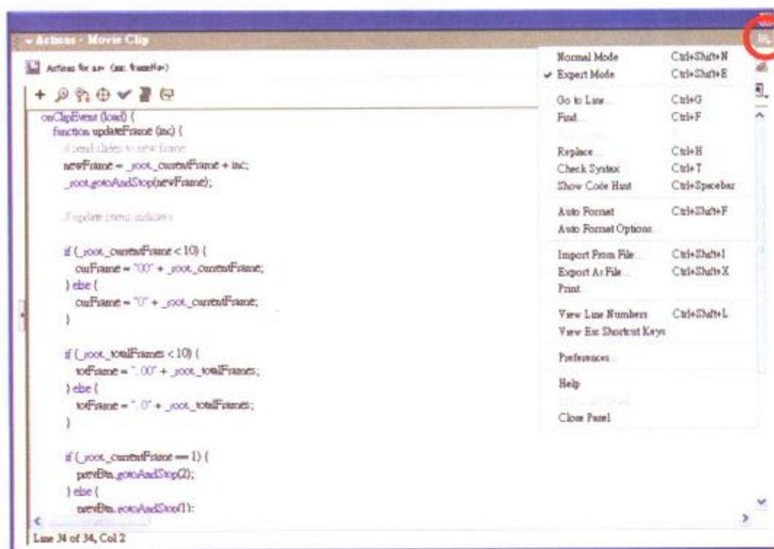
不同以往,Flash MX 的所有面板都有固定位置。



当然，同样可以切换成一般模式与专家模式，也可以自动编排零乱的程序代码，并且在重新编排时做基本语法检查。如右图所示。

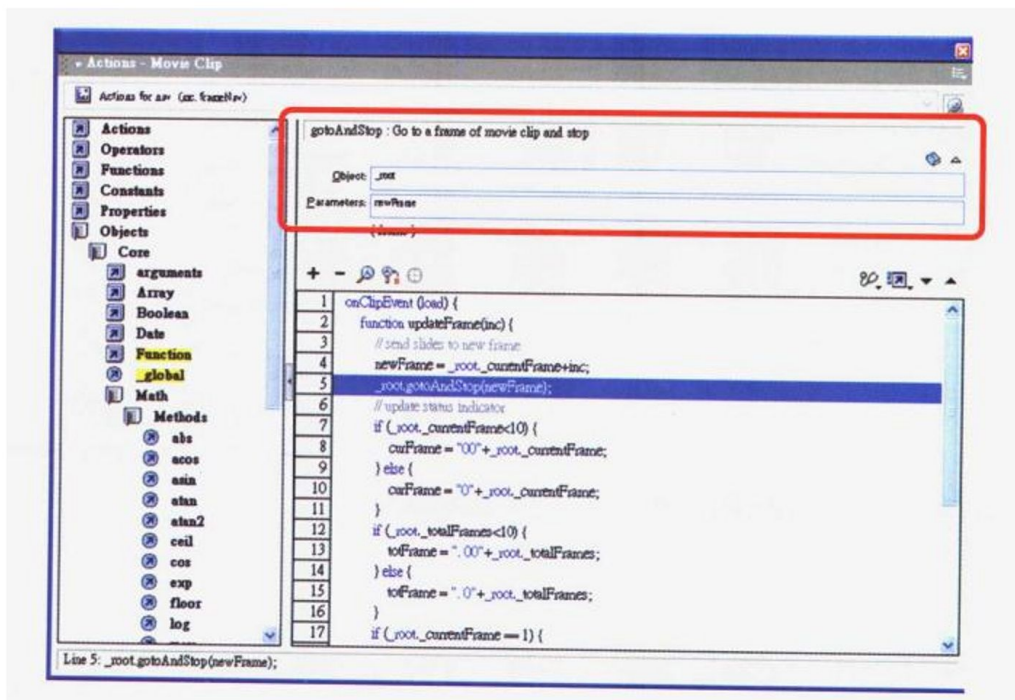


如果你用不到左边的语法集合，也可以将它们收合起来喔！



若想使用更多的功能，可以单击右图右上角的功能按钮查看。

在一般模式下，有更多的提示说明，不过本书用到不少进阶的语法，建议各位使用专家模式。如下图所示。



1-2

ActionScript 的保留字与变量

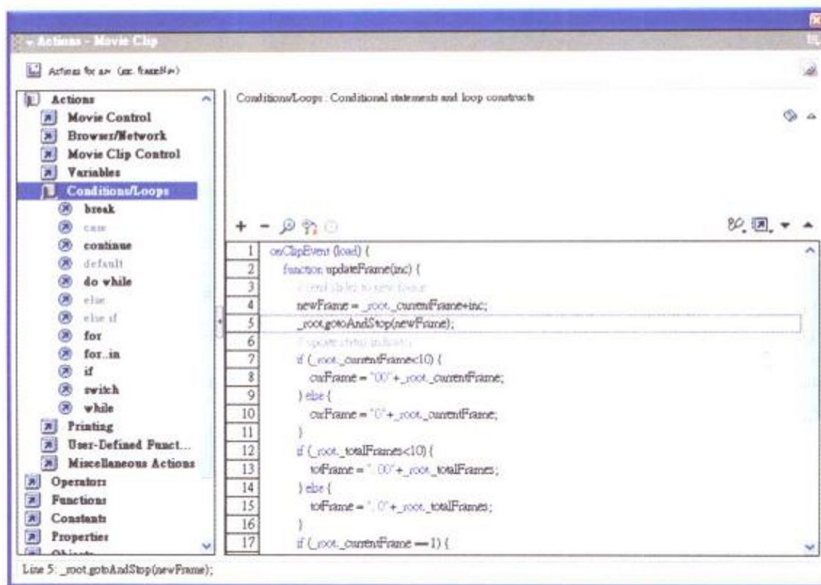
保留字

保留字是指 ActionScript 语法中，必须使用到的文字，这些文字不能当做变量来使用，故称为保留字。这些保留字有：

break	for	new	var
continue	function	return	void
delete	if	this	while
else	in	typeof	with

大多数的保留字都是程序语法中的指令。当然，所有内部函数的名称也算保留字的一种，这里就不再列举。万一您使用到这些函数名称当做变量，在执行的时候，就会出现错误信息。

大多数的保留字都是用于程序控制指令。如下图所示。



变量设定的基本原则

变量，在 ActionScript 程序设计中，不仅指程序中所用到暂时储存变值的变量，还包括像影片片段的实体名称，或是动态文字、窗体文字字段名称等，都算是变量。

在指定变量时，有些规则是必须遵循的，例如：

- ❶ 不可以使用数字 0~9，以及 \,.;/?*-=!@#\$\$%^&<>[]{}|\ 这些特殊字符作为变量名称的起始，但下划线 “_” 则允许。
- ❷ 变量名称不可含有空白。
- ❸ 不可以使用保留字或内部函数名称为变量名称。
- ❹ 在 ActionScript 中，变量名称是不分大小写的。
- ❺ 在同一段程序区段中，变量名称必须是惟一的，不可重复。

运算符

运算符（operator）是用于程序编写中，作为表达式与表达式之间的关联，例如，在 myString=“Flash MX” 中，myString 是变量名称，Flash MX 是字符串值，而中间的等号（=）就是运算符，用途是“指定”。运算符分为几类，用途也有些差异，列表如下所示。

数学运算符

+	正号	-	负号	~	位 1 补码	++	后递增	--	后递减
*	乘法	/	除法	%	余数	+	加法	-	减法

位运算符

<<	位向左位移
>>>	位向右位移（无正负号）
>>	位向右位移
&	位 AND
^	位 XOR
	位 OR

条件式运算符

<	小于	<=	小于等于	>	大于	>=	大于等于
==	相等	!=	不等于	?:	条件	=	指定

对于条件式运算符，要特别注意使用的时机，这些全是用于条件式，如 if、while 等，以等号来说，单纯等号代表“指定”，例如：

```
A=130;
```

此表达式表示把变量 A 的值设为 130，但若是要判断 A 是否等于 130 时，则要用“相等”运算符判断，也就是两个等号：

```
if (A==130)
```

逻辑运算符

!	逻辑 NOT
&&	逻辑 AND
	逻辑 OR

逻辑运算符，大多用在条件式判断，例如：

若<条件式 1 成立>则<条件式 2 成立>，语法为：

```
if (aug1 && aug2)
```

aug1 和 aug2 即是条件式 1 和条件式 2。例如，要判断变量 A 是否介于 -100~+100 之间，可以有如下写法：

```
if ((A) > -100) && (A < 100))
```

单一条件式，以括号括起，若是多个条件式要同时成立，则如上例， $A > -100$ 以括号括起， $A < 100$ 也以括号括起，中间以 $\&\&$ 相连后，再全部以括号括起来。

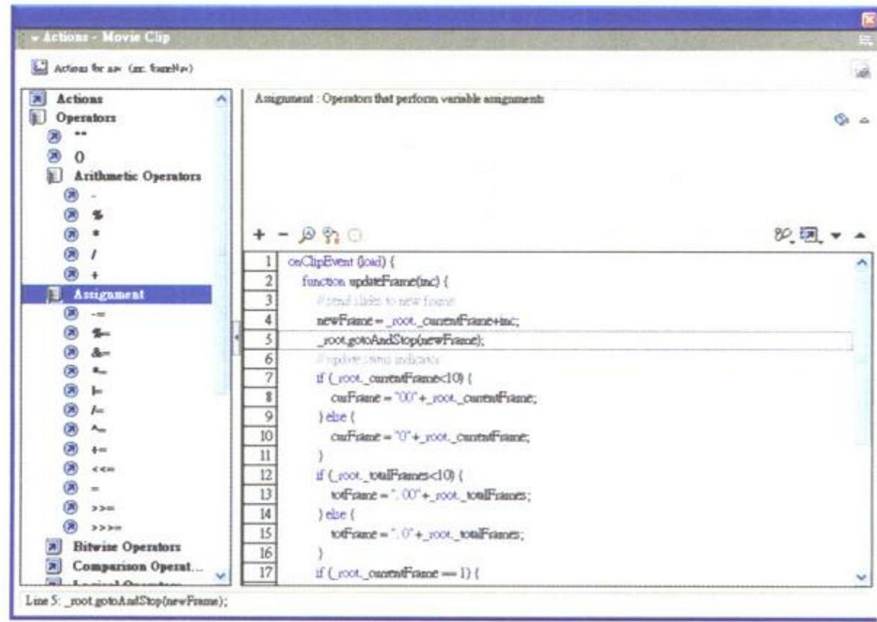
复合指定运算符

$\ast=$	$A \ast= B$ 代表 $A = A \ast B$
$/=$	$A /= B$ 代表 $A = A / B$
$\% =$	$A \% = B$ 代表 $A = A \% B$
$+=$	$A += B$ 代表 $A = A + B$
$-=$	$A -= B$ 代表 $A = A - B$
$\&=$	$A \&= B$ 代表 $A = A \& B$
$ =$	$A = B$ 代表 $A = A B$
$\wedge=$	$A \wedge= B$ 代表 $A = A \wedge B$
$\ll=$	$A \ll= B$ 代表 $A = A \ll B$
$\gg=$	$A \gg= B$ 代表 $A = A \gg B$
$\gg\gg=$	$A \gg\gg= B$ 代表 $A = A \gg\gg B$

复合指定运算符

<code>loopCount++</code>	loopCount 递增 1
<code>loopCount += 3</code>	相当于 loopCount 增加 3
<code>loopCount %= 5</code>	等于 <code>loopCount=loopCount % 5</code> 取得 loopCount 除以 5 的余数

复合指定运算符范例如下图所示。



1-3

条件式与循环

if 表达式

if 是最常用的条件判断式，语法为：

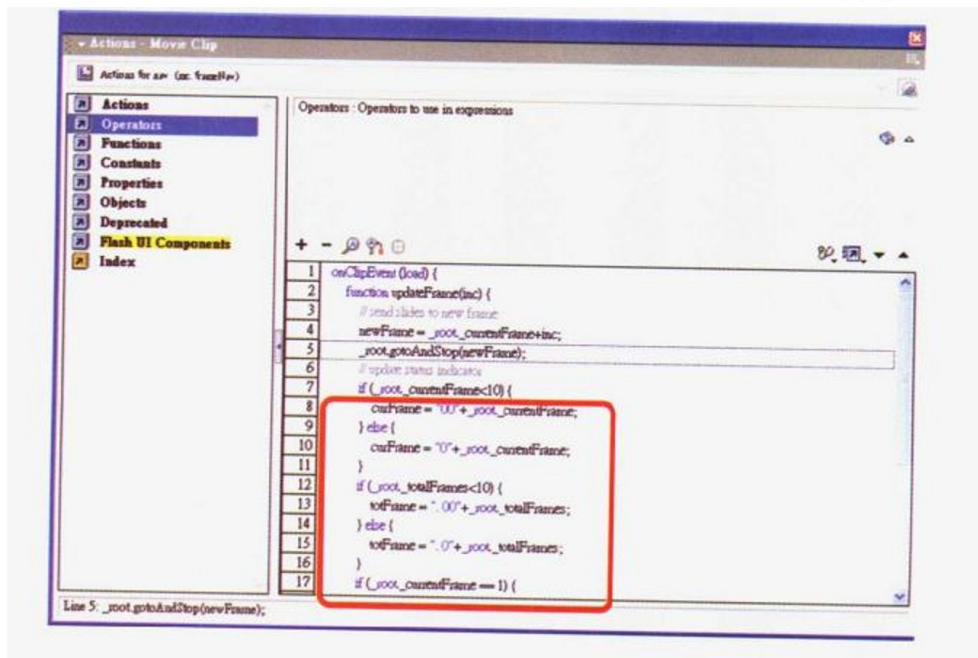
```
if (条件式){  
    条件成立时执行的内容  
}  
else{  
    条件不成立时执行的内容  
}
```

条件式可以有多个条件，如果两个以上条件必须同时成立时，以“&&”符号连接，代表“and”或是“且”的意思；反之，如果两个以上条件中，有一个成立即可时，以“||”符号连接，代表“or”或是“或”的意思，例如：

```
if ((quantity) =200) && (uniPrice) =100))
```

指定了一个“且”的表达式。

请注意，在 if 表达式中，若是使用到等于号时，必须使用“==”两个等号符号串接，才是代表真正的左边的表达式等于右边的表达式，单独一个等号的含义是指把等号右边的值“指定”到等号左边的变量或组件中。如下图红框中使用的例子。



for 循环

for 表达式，是常用的一种循环方式，指定重复执行 for 表达式中设定的次数，语法如下：

```
for (var 计数器变量 = 起始值 ; 计数器变量结束值 ; 计数器增减的方式){
    要重复执行的动作;
}
```

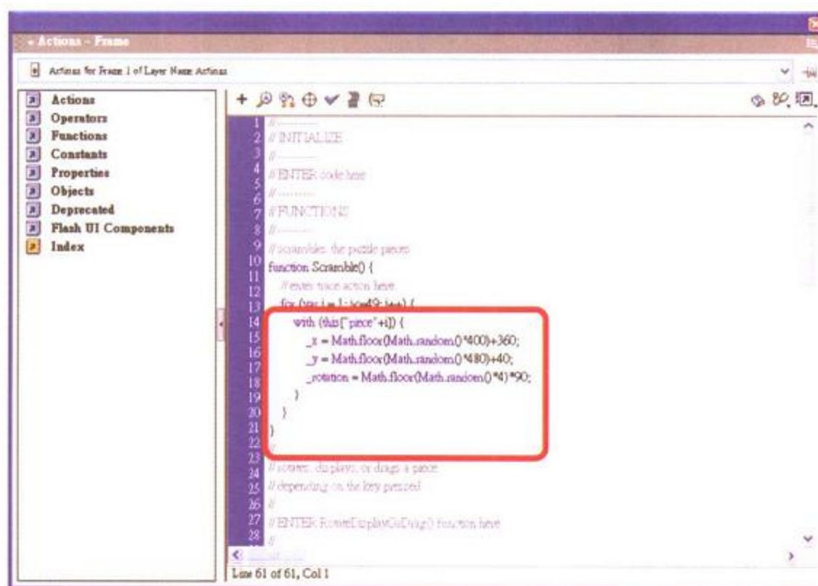
我们来一个一个拆解说明。先说明 for 循环的含义，整个 for 循环是这么指定的：

当计数器由起始值开始，以指定的增减方式，达到计数器变量结束值时，就停止这个循环，例如：

```
for (var i = 1 ; i <= 100 ; i++)
```

一个计数器名称为 i，由 1 起算 (i=1)，每次加 1 (i++)，直到计数器是 100 为止 (i <= 100)。

for 循环可以用在许多地方，例如数组的初始化便是一例。如下图所示。



do...while 循环

do...while 循环则是不指定重复执行的次数，而是由条件来判断，语法为：

```
do {
    要重复执行的动作;
}while(条件式)
```