

你还被局限在
2D平面空间里做
Flash动画吗？

 奶绿茶 著

- 最新流行的Flash 3D效果,看了之后心动却不知从何着手吗？
- 可随手套用的程序代码+最实用的动画范例,让你的Flash作品瞬间耀眼抢手！

FLASH

特效

Be ready for fighting!!!

商业范例随学随用

程序源代码
视频教学动画



- 迷你旋转木马MiniCarousel 3D • 超立体旋转九宫格 • 透明感十足的SpaceGallery
- 漫天飞舞的PaperDWord • 最经典的CoverFlow效果 • 气派十足的旋转木马Carousel3D效果
- 最杀FlatWall3D电视墙效果 • 立体方块跳个不停 • 舞动的牛奶盒特效
- 天旋地转TiltView效果 • 螺旋式SpiralCarousel



清华大学出版社



奶绿茶 著



商业范例随学随用

清华大学出版社
北 京

本书版权登记号：图字：01-2009-8004

本书为旗标出版股份有限公司授权出版发行的中文简体字版本。

内 容 简 介

本书是国内第一本介绍 Flash 3D 制作技术的专业图书。作者从基本的 ActionScript 知识与动画原理讲起，重点介绍了与 Flash 3D 技术相关的 Tween 类和 DocumentClass 类，以及开源 Flash 3D 渲染引擎 Papervision 3D。全书详尽的程序代码解说与丰富实用的动画范例相结合，即使你从来没有接触过编程，也能轻松学会。

相信按照本书的指引，任何对 Flash 技术有兴趣的读者，都会做出耀眼的 Flash 3D 动画，让你的网页与众不同。

本书封面贴有清华大学出版社、旗标出版股份有限公司防伪标签，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

Flash 3D 特效商业范例随学随用 / 奶绿茶著. —北京：清华大学出版社，2010.5

ISBN 978-7-302-22559-1

I. ①F… II. ①奶… III. ①动画—设计—图形软件，Flash ActionScript 3.0、PU3D IV. ①TP391.41
中国版本图书馆CIP数据核字 (2010) 第069744号

责任编辑：夏非彼 张 凝

装帧设计：图格新知

责任校对：张 楠

责任印制：孟凡玉

出版发行：清华大学出版社

地 址：北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者：北京彩和坊印刷有限公司

装 订 者：三河市新茂装订有限公司

经 销：全国新华书店

开 本：190×260 印 张：15.5 字 数：377 千字

附光盘 1 张

版 次：2010 年 5 月第 1 版 印 次：2010 年 5 月第 1 次印刷

印 数：1~4000

定 价：56.00 元

产品编号：036191-01

序

Preface

《Flash 3D特效商业范例随学随用》一书，是笔者所写的第一本关于Flash ActionScript 3.0的书籍，接触Flash 6年的时间，用玩乐的心态来研究Flash，没有想到一玩就玩出兴趣来，一开始也只会使用Flash做动画，编写ActionScript的简单程序代码，至今已经可以使用ActionScript 3.0来开发大型案例。感谢一路走来不断指导我的前辈们，还有许多不吝分享原始代码的网友们！

本书前几章笔者安排了ActionScript 3.0的介绍，让你了解如何从ActionScript 2.0转移到ActionScript 3.0。所以你至少需要熟悉Flash软件的使用，制作帧或转换元件等，也需要略懂什么是变量，什么是函数（function），这样读起来会比较不吃力。

另外笔者还在本书所附光盘中提供了两段自制的教学影片，好让不那么熟悉ActionScript程序开发的读者能一步一步地学习。这两段教学影片分别如下，你只须双击.html文件，便可以浏览器播放之。

■ 影片教学一：

_Videl_Tutorials\01_FlashCS4PV3D.html

对应本书的PV3D设置与安装。

■ 影片教学二：

_Videl_Tutorials\02_FlashDevelop.html

对应本书的第4、5章，使用Flash Develop编辑器来快速完成PV3D案例的开发，并熟悉软件提供的快捷键功能，以加快程序的编写。

奶绿茶



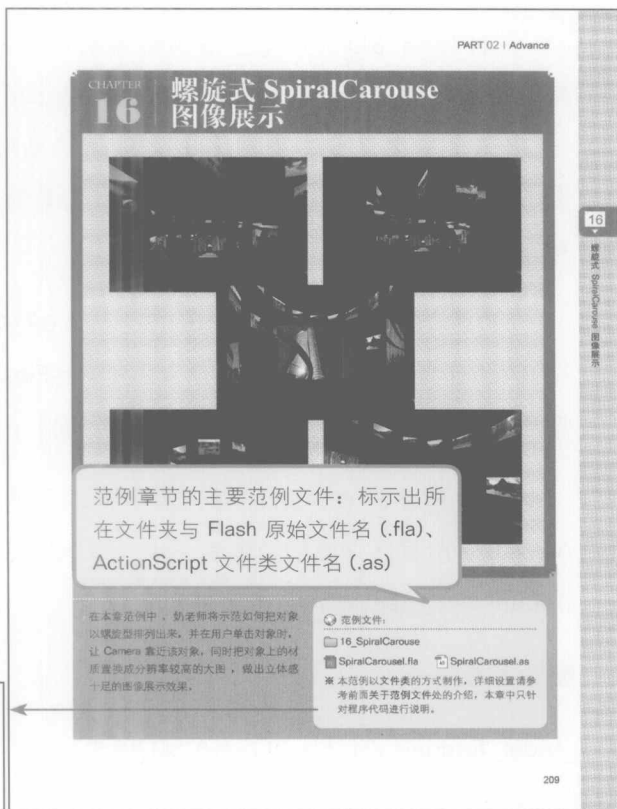
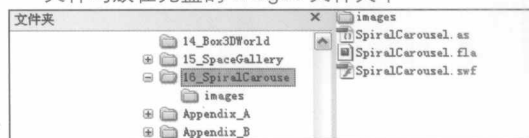
关于范例文件

About CD

范例文件标示

本书各章主要范例文件均标示在各章第 1 页下方，非范例章节中的简易范例文件也都以专门的格式标示出来。

▼ 若此范例有需要加载的图像文件，则图像文件均放在光盘的 images 文件夹中



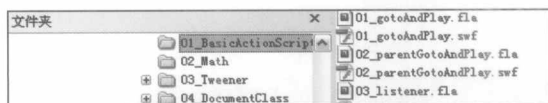
1-5 ActionScript 3.0 基本练习

接着让我们来实际练习一下 ActionScript 3.0 的程序写法，更好更清楚地了解它和 ActionScript 2.0 的语法差异。

单击对象，播放主时间轴动画

范例文件 01_BasicActionScript3 \ 01_gotoAndPlay.fla

▲ 一般章节的简易范例文件：标示出所在文件夹与 Flash 原始文件名 (.fla)、ActionScript 文件类别文件名 (.as)



文件版本、文件类别设置与程序代码确认

本书范例文件全都以文件类（详见第 4 章的说明）的方式制作。因此除了确认范例文件彼此间的位置关系外，你还须确认文件类别的设置是否正确，才能确保程序代码能正常运作。

01 本书使用Flash CS4版本，因此所有.flc 文件都为 Flash CS4 版本文件。

02 打开 .flc 范例文件，确认属性面板的文件类框中，填入的是对应的 .as 文件名（不含扩展名）。这也正是此 .flc 文件的类别。



填上对应的 .as 文件名 “SpiralCarousel”

03 用 Flash 或 FlashDevelop 等程序编写软件（详见 4-2 节）打开对应的 .as 文件（本例为 SpiralCarousel.as），确认其中的类名称和实例名称与文件名一致。

▼ ActionScript 3.0 : SpiralCarousel.as

```
package {
    import flash.display.*;
    ... (略) ...
    public class SpiralCarousel extends MovieClip {
        private var view : BasicView;
        private var loadedNumber : int = 0;
        ... (略) ...
        public function SpiralCarousel() {
            init3DEngine();
            init3DObject();
            initObject();
        }
        ... (略) ...
    }
}
```

类名称要和类别文件名一致

实例名称也要和类文件名一致

目 录

Contents

PART 01 "Basic

01	ActionScript 3.0 概述	1
1-1	Flash 3D 的基础 —— ActionScript 3.0	2
1-2	学习 ActionScript 3.0 的 12 个理由	2
1-3	ActionScript 3.0 显示对象类表	9
1-4	侦听与发送事件	11
1-5	ActionScript 3.0 基本练习	12
02	用程序代码制作动画	29
2-1	渐进移动	29
2-2	基本的数学公式与三角函数	32
2-3	简单的转盘效果 —— Carousel3D	38
03	ActionScript 3.0 制作动画的好伙伴 —— Tweeners	45
3-1	安装 Tweeners 类	45
3-2	Tweeners 的基本应用	47
3-3	转换 Tweeners 的运动方式	48
3-4	运用 Tweeners 的滤镜效果	50
3-5	使用 Tweeners 制作反复播放的动画	51
3-6	运用 Tweeners 制作出自动全屏效果	53
3-7	滑动切换的 Tweeners Slide 效果	56
3-8	Tweeners 的贝兹曲线式移动	58

04 DocumentClass 文件类	63
4-1 什么是类? 为什么要写类?	63
4-2 FlashDevelop 3 程序代码编辑器	63
4-3 以 FlashDevelop 编写并设置文件类	65
4-4 实际运用文件类制作动画	68
4-5 自定类程序代码	71
05 Flash 3D 秘密武器 —— Papervision3D	75
5-1 什么是 Papervision 3D	75
5-2 安装 PV3D类文件	76
5-3 了解 PV3D的坐标系	76
5-4 认识PV3D的类	78
5-5 建立PV3D世界所需的基本类与步骤	79
5-6 跟 PV3D 说 Hello —— 实例化 PV3D 基础世界	81
5-7 运用 PV3D 的基本对象与材质	84
5-8 让3D对象与鼠标互动	101
5-9 PV3D 与 Tweenener 的合作应用	103
5-10 使用 FreeCamera自由镜头	105

PART 02 "Advance

06 漫天飞舞的 Paper 3D World	108
--------------------------------------	------------



07 迷你旋转木马式秀图效果 —— MiniCarousel3D	115
---	------------



08 最经典！CoverFlow 图像展示效果.....125



09 天旋地转 TiltViewer 效果.....135



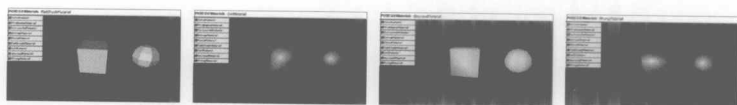
10 FlatWall3D 电视墙效果143



11 经典旋转木马Carousel3D 秀图效果153



12 光源材质展示与简易型的方块弹跳效果167



13 加载外部 3D 模型——舞动的牛奶盒179

13-1 PV3D 所支持的 3D 模型文件——DAE180

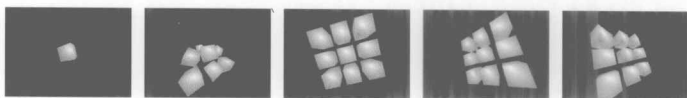
13-2 下载并安装 COLLADA 格式的输出外挂插件.....180

13-3 在 3D 绘图软件中建立模型并导出 DAE 文件182

13-4 用 PV3D 加载 DAE 模型	185
13-5 修正材质无法自动贴上的问题	187



14 超立体旋转九宫格 —— Box3Dworld	191
---------------------------------	-----



15 空间感与透明感兼具的SpaceGallery	203
---------------------------------	-----



16 螺旋式 SpiralCarouse 图像展示	209
---------------------------------	-----



PART 03 "Appendix

A 关于 FlashPlayer10 的 3D 功能	218
B 关于 Flash Player 性能的话题	227
C PV3D 延伸应用	235

01 *Basic*

CHAPTER

- 01 ActionScript 3.0 概述
- 02 用程序代码制作动画
- 03 ActionScript 3.0 制作动画的好伙伴——
Tweeners
- 04 DocumentClass 文件类
- 05 Flash 3D 秘密武器——Papervision3D

1-1 Flash 3D 的基础 —— ActionScript 3.0

随着 Flash 的发展，其版本从 Flash Player1 到了 Flash Player10，程序语言也从零散的动画指令，到现在已经具有对象导向精神的 ActionScript 3.0。以前所学的程序语法到了 ActionScript 3.0 版本，大部分都已取消或更改。不过 ActionScript 3.0 整体不仅性能大大提升，其更加严谨的语法加上功能强大的类，让 Flash 能玩的酷炫效果越来越多！

ActionScript 3.0除了可以使用 Adobe Flash CS3 以上版本开发外，也可选择使用 AdobeFlex。也就是说，一样的程序语言可以在不同的环境里编写、编译出来。截至笔者完稿日，虽然目前 Flash 的最新版本为 CS4，而 Flash Player 的最新版本为 10，但一样是使用 ActionScript 3.0，所以不需要因为软件更新而重新学习 ActionScript。



正因为有这样先进的 ActionScript 3.0 为基础，我们也才能利用 Papervision3D 包开发出酷炫的 3D 互动特效呢！

ActionScript 3.0 看起来像 ActionScript 2.0 的版本升级，但整个核心架构却是大大不同，语法也彻底翻新，对初学者来说，学习 ActionScript 3.0 的确门槛较高一些，但如果你已经有 ActionScript 2.0 的底子，只要再花些时间，一样可以快速上手；而对原本就熟悉 Java 或 C++ 的人来说，ActionScript 3.0 对象导向的观念则会让你感到非常熟悉。

笔者特别在本书的前 2 个章节中，先介绍 ActionScript 3.0 的架构与概念，好让你一步步了解到 ActionScript 3.0 的强大功能。而有别于一般书籍总是直接使用许多对象导向专用名词，为了让许多刚接触 ActionScript 3.0 的朋友轻松上手，本书中都会尽量采用口语化的说明方式，用最简单的文字，让你充分体会什么是 ActionScript 3.0。

1-2 学习 ActionScript 3.0 的 12 个理由

Reason01 类划分明确

在 ActionScript 2.0 以前的时代，都是使用 MovieClip 打天下，不论是加载图像、绘图、特效等等，它都能一手包办。但用了一堆的 MovieClip，却只使用其中部分功能，反而让性能都浪费掉。

ActionScript 3.0 则把功能面都分开，需要用哪个功能，就使用该功能的类，有用到的才导入，这样一来，功能分类就更加明确，也让整体的性能提升不少。

 关于 Flash 的性能问题，由于 Flash 本身是靠用户端的 CPU 做实时运算来呈现动画效果的，所以当 CPU 满负荷时，Flash 运算也会跟着慢下来，会导致本来很流畅的动画变得时常被卡住。

▼ ActionScript 2.0

```
movieClip.loadMovie("demo.swf");
//MovieClip 对象，加载外部的 demo.swf 文档
movieClip.moveTo(0, 2);
//MovieClip 对象绘图功能
movieClip.attachBitmap(myBitmapData);
//MovieClip 对象附加位图对象
```

▼ ActionScript 3.0

```
var ldr:Loader = new Loader();
//Loader对象，用来加载外部的 jpg、png、gif(不包含动画)或 SWF 文件
movieClip.graphics.moveTo(0, 2);
//MovieClip 对象的 graphics 属性，其属性值为 Graphic 类，可用来绘制图形
var bitmap:Bitmap = new Bitmap(myBitmapData);
//MovieClip 对象已经不能附加位图对象
//取而代之的是 Bitmap 类。
```



奶老师贴心语：关于 MovieClip 对象

所谓的 MovieClip 对象，指的是利用 new 这个关键词以程序来实例化出的 MovieClip 类别实例，或是直接在 Flash 编辑接口中放置一个制作好的 MovieClip 元件实例。而 ActionScript 2.0 无法使用 new MovieClip() 这样的程序来实例化 MovieClip 对象，而是使用 createEmptyMovieClip() 函数来达到目的。

Reason02 事件的统一

ActionScript 2.0 中有些类使用“on事件”的写法来赋予事件，有一些类则使用addListener，另一些又使用 addEventListener。明明是同一个程序语言，却有着不同的执行方法，很容易让人搞不清楚。

ActionScript 3.0 则全部统一使用 addEventListener() 函数来侦听特定的发送者所发出来的特定事件。

▼ ActionScript 2.0

```
ActionScript 2.0
movieClip.onRelease = function(){};
//MovieClip 对象的鼠标释放事件
Stage.addListener(listenerObject);
//listenerObject 对象侦听 Stage 类所发出的所有事件
component.addEventListener("click", onClick);
function onClick (e:Object){};
//onClick 侦听器函数侦听 component 元件发出的 "click"事件
```

▼ ActionScript 3.0

```
movieClip.addEventListener(MouseEvent.CLICK, onEventMouseUp);
function onEventMouseUp(e:MouseEvent){};
//统一采用 addEventListener 的方式侦听
//由 onEventMouseUp 侦听器函数侦听 movieClip 对象所发出的
// MouseEvent.CLICK 事件
```

Reason03 更有效率的内存工作方式

ActionScript 2.0 的显示对象一旦被实例化后，就会立即出现在 Flash 的场景里。而 ActionScript 3.0 的显示对象被实例化时，并不会马上出现，而是先存在内存中，等需要出现时，才通过 `DisplayObjectContainer` 或其子对象的 `addChild()` 方法将其加入，这样便可省去 Flash Player 多余的运算，而对 Flash Player 来说，更新 Player 的画面正是最消耗性能的动作，因此 ActionScript 3.0 的这种新做法，可以让动画执行更顺畅。



所谓的显示对象，简单来说，就是可以被显示化的对象，像是 `MovieClip`、`Button`、文字、位图、色块、影片等都是。而数组、变量、函数这些跟视觉没有直接关系的类，就是所谓的非显示对象。稍后于 1-3 节中，会再为你做更详细的说明。

▼ ActionScript 2.0

```
this.createEmptyMovieClip("movieClip", 1);
//建立一个实例名称为 movieClip 的空白 MovieClip 对象
this.attachMovieClip("linkID");
//从元件库调用指定的 MovieClip 元件，并加入在目前的时间轴里，
//一旦执行该函数，设置的对象会立即出现在场景上
```

▼ ActionScript 3.0

```
var movieClip:MovieClip = new MovieClip();
//所有类的实例化，都是通过 new 这个关键词
this.addChild(movieClip);
//将 movieClip 对象加入到目前的显示对象容器里
```

Reason04 对象的属性统一

ActionScript 2.0 中的对象属性，有些有下划线，有些则没有下划线，一样让人搞不清楚。在 ActionScript 3.0 中，则统一取消下划线。

▼ ActionScript 2.0

```
movieClip._x;
//MovieClip 对象的 x 轴属性
movieClip._alpha;
// MovieClip 对象的 alpha 属性
Stage.width;
//Stage 类的宽度属性
```

▼ ActionScript 3.0

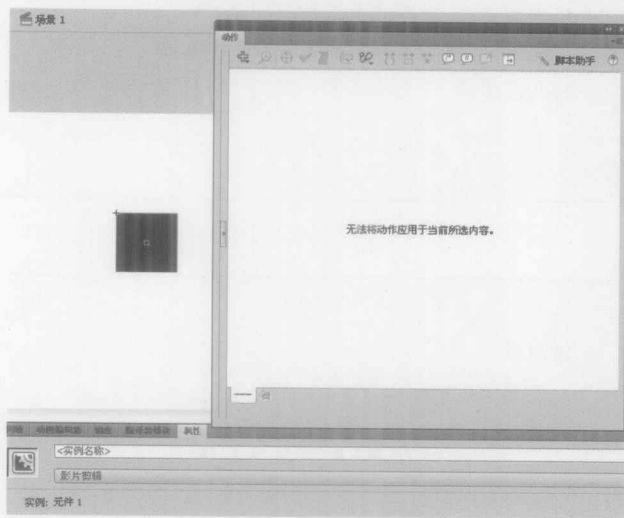
```
movieClip.x;
//MovieClip 对象的 x 轴属性
movieClip.alpha;
//MovieClip 对象的 alpha 属性
stage.stageWidth;
//Stage 类的宽度属性
```

ActionScript 2.0 V.S. ActionScript 3.0 显示对象属性比较表		
ActionScript 2.0	ActionScript 3.0	说明
_x	x	显示对象的 x 轴属性
_y	y	显示对象的 y 轴属性
_alpha (由0完全透明到100不透明)	alpha (由 0 完全透明到 1 不透明)	显示对象的透明度属性
_rotation	rotation	显示对象的旋转属性
_visible (可填入布尔值或是 1、0)	visible (只能填入布尔值，不可填入 1、0)	是否看得见该显示对象
_xmouse _ymouse	mouseX mouseY	显示对象目前鼠标的 x 坐标位置 显示对象目前鼠标的 y 坐标位置

ActionScript 2.0	ActionScript 3.0	说明
_xscale _yscale(100 表示不缩放)	scaleX scaleY(1 表示不缩放)	显示对象的宽度缩放比 显示对象的高度缩放比
_width _height	width height	显示对象的宽度 显示对象的高度
Stage 类属性		
Stage.width	stage.stageWidth	场景宽度, 只读属性
Stage.height	stage.stageHeight	场景高度, 只读属性

Reason05 统一程序代码的编写位置

ActionScript 2.0 程序代码可以写在时间轴上, 也可以写在对象上, 程序代码到处都是; ActionScript 3.0 则统一规定只能写在时间轴上, 或外部定义类 .as 文件。



ActionScript 3.0 只能在时间轴的帧中写程序代码, 无法在对象上编写

Reason06 变量的强制性声明

ActionScript 2.0 可以不用 var 这个关键词声明, 就直接指定变量, 很容易让人搞不清该变量是局部变量还是全局变量; ActionScript 3.0 则一定要通过 var 来声明变量。另外, ActionScript 3.0 还新增了 const 关键词, 用来声明常量。



所谓的常量, 就是当该值被定义后, 就无法再被重新写入。像 Math.PI, 就是 Math 类下的静态常量 PI, 它会回传圆周率 3.1415926。常量通常用来存储不需改变也不可改变的值。

▼ ActionScript 2.0

```
num = 3;
//可以不用 var 来声明变量
var str:String = "milkmidi";
```


▼ ActionScript 3.0

```
var num:Number = 3;
var str:String = "milkmidi";
num2 = 3;
//以上程序代码在发布时会产生“1093: 语法错误”的信息
//因为 num2 变量未定义
var test:int = 3;
test = 4;
//因为使用了 var 来声明成变量，所以可以重新指定新的数值
const TEST:int = 3;
TEST = 3;
//会发生“1049: 对指定为常数的变量进行的赋值非法。”的错误信息
//因为常量一经声明后，就无法被重新指定新的值
//一般声明常量时我们会使用全部大写的英文字母，以便与一般的变量区别
```

01

ActionScript 3.0 基础

Reason07 程序代码更加严谨

在对象还不存在，或未经声明、未定义时，ActionScript 2.0 依然可以调用该对象的方法或属性；ActionScript 3.0 在对象未定义或值为 null 的情况下，若去调用该对象的方法，则会产生错误信息。

▼ ActionScript 2.0

```
testFunction();
//调用一个未定义的函数，并不会产生错误
movieClip._x = 20;
//即使场景上没有 movieClip 对象，也可以指定该对象的属性，
//编译器也不会产生错误信息
```

▼ ActionScript 3.0

```
testFunction();
//编译时会输出，“调用可能未定义的方法 testFunction”的错误信息
movieClip.x = 20;
//编译时会输出，“1120: 访问未定义的属性 movieClip”
```

Reason08 可使用文件类

用 ActionScript 2.0 时，主时间轴本身虽然也是一个 MovieClip 类，但没有办法方便地将它扩充成自定义的类；ActionScript 3.0 则新增了文件类，让主时间轴可以方便地扩充成自定义类。