9tech 开发者社区技术支持

ActionScript 3.0 图像处理

基础教程

〔英〕Todd Yard 著

陈文登 译 | 方志超 审校

深入学习 ActionScript 3.0 BitmapData 和运行时图像处理

探究 Flash Player 10 中 ActionScript 的绘图 API 和原生 3D 新特性

全面学习 Pixel Bender 和 Flash Player 中 Shader 的创建方法

科学出版社

ActionScript 3.0图像处理基础教程

〔英〕Todd Yard 著
陈文登 译
方志超 审校

科学出版社
北京

图字：01-2012-8844号

内 容 简 介

图像处理是Flash应用的一个重要领域，如网络上常见的Flash画板、在线Photoshop、涂鸦等应用，都是基于Flash图像处理技术实现的。本书介绍ActionScript 3.0中图像、动画、视频和音频等大量视觉效果处理的相关知识。这些基本知识不仅可以应用在上述绘图应用中，同时也可以应用到游戏中，替代以往通过加载位图实现的效果，一方面可以减小游戏的体积，另一方面可以大大提升游戏性能。另外，作者将书中介绍的图像处理功能集成到一起，开发了aeon和aether类库，开发者通过这两个类库，可以轻松快速制作出各种绚丽的效果。

本书适合程序设计、游戏开发、网页设计等相关领域的读者，也可作为相关高等院校及培训机构的教材。

图书在版编目（CIP）数据

ActionScript 3.0图像处理基础教程/（英）Todd Yard 著;陈文登译.—北京：科学出版社，2013.8

书名原文：Foundation ActionScript 3.0 Image Effects

ISBN 978-7-03-037536-0

Ⅰ.A… Ⅱ.①T… ②陈… Ⅲ.动画制作软件-教材 Ⅳ.TP391.41

中国版本图书馆CIP数据核字（2013）第107662号

责任编辑：喻永光 杨 凯 / 责任制作：魏 谨

责任印制：魏 谨 / 封面制作：段淮沱

北京东方科龙图文有限公司 制作

http://www.okbook.com.cn

科 学 出 版 社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

北京东海印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

2013年8月第 一 版 开本：787×960 1/16

2013年8月第一次印刷 印张：37 1/2

印数：1—3 500 字数：900 000

定价：98.00元

（如有印装质量问题，我社负责调换）

Foundation ActionScript 3.0 Image Effects
By Todd Yard, ISBN: 978-1-4302-1871-5
Original English language edition published by Apress Media.

Cover Image by Corné van Dooren.

前　言

我觉得没有比“Flash”更好的名字了。虽然现在很多平台都用到“Flash”这个词，而不仅仅是Flash Player和IDE，但是不管是用Flash或Flex Builder开发，在线或通过AIR传播Flash影片，最终输出结果都是一个通过Flash Player播放的SWF文件，和当年只用Flash做动画是一样的。所以，不管你做的是什么——Flash影片、Flash应用或生成艺术——总归是“Flash”。朋友们，是这样的。从最开始，人们对Flash影片的反应通常是“哇哦”，如果是像你一样懂开发的人，反应可能是“这是怎么做的？”

我从2000年开始从事Flash工作，不久Flash 5就发布了。如果你是一位Flash的老用户，应该可以回想起那时Flash 4还只是一个基于时间轴的动画工具，只有屈指可数的几个ActionScript命令（即便是没有对象和数组，不过在Photoshop的图层概念出现之前，我更喜欢用ActionScript）。而且那时候，虽然就这么几个命令，Flash做的大量新颖的效果同样也是非常惊人的。

惊人之处在于，Flash制作只需要了解少许的编程知识，因此也吸引了各行各业的不同人才，从动画师到程序员，再到插画师和音乐人，等等。现在的哪一款软件拥有同样的功能？

多元化的主题成就了社区这样一个可以让我们一起学习和研究各种不同新兴技术的组织，社区是开放、乐于分享的一个地方，这些分享的资源、乐趣和经验也是非常值得一看的。

不管是一段动画、一个游戏、一个应用或是一些值得一看的酷“东东”，Flash给人的第一印象是漂亮的视觉效果。第一次接触Flash我就兴奋不已(那大约是在十年前了)，同时也进行了大量的关注。Flash平台的成长是非常迅速的，ActionScript已经是第三次更新了，它是一个复杂的、强大的语言，开发者可以用ActionScript做出与桌面应用相抗衡的Web应用。随之而来的是，Flash对图形元素的控制能力也有了大幅的提升和强化。

本书中，我们会学习大量在SWF中通过ActionScript创建和控制这些视觉元素的方法。无论你是游戏、应用开发者或动画人员，或者只是玩一玩，本书都包含了所有你需要的像素处理知识。

书中的第一部分，我们会逐一学习ActionScript中主要的图像创建和处理的内容。包括绘图API、滤镜和图层模式，掌握BitmapData和它的方法、属性等内容，然后深入学

习Flash Player 10的3D和Pixel Bender等新增特性。完成这些课题之后，介绍了一些特效类库，通过这些类库可以轻松完成剩余章节里的示例，创建大量的效果。

第二部分列出了一系列的教程，展示如何将第一部分学到的知识应用到实际中去。真正有意思的才刚刚从这里开始。这很重要，因为有意思才把那么多人吸引到Flash上来（包括我），也是因为有意思才驱使着人们去玩、去研究，做出一些真正“赞”的、有创意的应用。你也可以创建一些商业的应用，让Flash不只是取悦人们的眼睛，而是强大应用的实用性，创建更吸引人的用户体验，在Web中呈现出卓越超群的作品。来吧，让本书告诉你怎么做。

目标读者

本书不是介绍面向对象编程或ActionScript 3.0基础的，也不是讲解Flash或Flex Builder编译环境的书（实际上，我努力让本书不会局限于任何一个编译环境）。所以你可能要先熟悉一下如何编译SWF文件，或者至少了解基本的ActionScript 3.0知识。

希望本书中图像编程、像素处理的内容能给你带来快乐。我不是什么计算机专家，也没有编程或数学背景，我只是喜欢Flash，它能给我带来无限的欢乐，让我无法自拔。如果你熟悉ActionScript 3.0，也对这个语言的图像处理能力饶有兴趣，不怕碰到数学算法，那么本书正合你的口味。

开发环境

我是一个ActionScript开发人员。刚开始时用Flash，现在天天在Flex Builder中用mxmlc编译器编译SWF文件。本书中，我尽量用纯ActionScript代码，这样可以在任何编译环境下编译示例，而不单纯地依赖于Flash或Flex框架中的时间轴或框架。

在本书中，你可能会慢慢发现，这些纯ActionScript 3.0代码的示例，在编译SWF或测试影片时通常会有一个配置说明，因为每个环境下的配置方式不同，实际没有做出说明。不过在本书的附录中，我介绍了如何在Flash和Flex Builder中编译书中的示例。如果对这些编译环境不熟悉，可以随时查看。

如果你用的是Flex Builder 3，需要多做几步，配置一下SDK，用最新的包含Flash Player 10新增ActionScript类的SDK，并设置Flex Builder用Flash Player 10来编译SWF。如果对这些类（如Vector和Shader）不熟悉，现在就翻到附录，看看如何配置吧。

前几章，我会提醒你到附录查看配置方法，但是如果你是跳着章节看书，最好先看一下附录的内容。

代码注释

代码应该要有注释，我觉得这没有什么好争论的。但是你可能注意到，书中示例的代码都是没有注释的。原因有三个。首先，不加注释可以节省很大的篇幅，如果加上注释，某些教程可能会被整个去掉，示例也会很少。我想用尽可能少的章节介绍更多的示例。其次，这是我的个人习惯，在看一些技术文章时，我更喜欢在文章中看到代码注解，而不是把注释分散到代码中，因为注释会增加代码的长度，让代码更难凸显重点。最后，我在书中用了很多篇幅对代码的内容进行详细的解释，加上注释会把完整的说明打散，而且会产生冗余重复的信息。

不过，正如我所说的，代码应该是要有注释的。在本书附属的源代码中，打开对应的代码文件，可以看到完整的代码和注释。只是书中的代码没有注释而已。

本书约定

为了让内容尽可能地清晰、容易阅读，整本书都遵循了下面的文字规范：

- 代码用等宽字体。
- 新增或变更的代码用加粗等宽字体。
- 菜单命令的写法是“Menu→Submenu→Submenu”。
- 当有需要引起注意的内容时，用下面的方式强调：

> 别说我没提醒你啊！

- 如果代码过长，无法一行显示出来，则用箭头➥表示：

```
This is a very, very long section of code that should be written all➥
on the same line without a break.
```

目　录

第1章 绘图API

回到遥远的以前，地球还是冰冷的，所有的生命都是由海洋生物演变而来，世界大陆还只是一个整体的大陆块，iPod的存储量还不到1GB，那时候的Flash无法动态地绘制图形。如果还是这样子的话，那这本书就没有什么好写的了。庆幸的是，Flash MX推出了开发人员期盼已久的绘图API（Application Programming Interface），它的强大功能引起了众人的广泛关注。

1.1 绘图API历史回顾

原始的ActionScript 1.0绘图API只有8个简单命令，在运行时只能绘制和填充简单线条。在如今这个飞行汽车和喷气式飞机都司空见惯的年代，绘制直线和曲线，用纯色或渐变颜色填充线条，简直不值一提，而在当时只能先定义好向量和库中的图像来绘制。绘图API提供无数新的可能——从液态图形和图表页面，到复杂的3D引擎，再到动态的视觉效果。

到了ActionScript 2.0，绘图API又多了两个新的命令：一个用来绘制渐变填充线条，另一个用位图填充图形。另外，还加强了一些原有的方法和一些其他功能，如定义渐变扩展的方法，定义线条的节点和拐点样式。

到了ActionScript 3.0，最新的 Flash Player升级到9.0版，所有的绘制方法都集成到了专门的`Graphics`类中，不再属于`MovieClip`；新增了一些新的绘图图形，如矩形和椭圆形。这些变化很有用，但没有太大的变化。开发者仍然在等待激动人心的时刻。

现在，Flash Player升级到了10.0，ActionScript 3.0的绘图API有了翻天覆地的变化。可以把一个对象的图形数据直接复制到另外一个对象，可以在任意时刻保存任何对象一连串的绘图命令，并在必要时返回，笔触可以用图形填充。最后，ActionScript新增的`Shader`可以用来绘制笔触和填充，也可以作为自定义的位图渐变或填充图案。

`Graphics`类及其新特性是一个庞大的话题，可以专门用一章来介绍，也是非常重要的一个题目。书中讲到的大多数图像处理技术，都需要牢固地掌握ActionScript绘图API所提供的功能。所以，我们先从8个基本命令开始学起，据说这也是最好的入门知识。

1.2　早期的绘图功能

首先，我们来看一下在早期的Flash版本中就有的几个绘图API方法。如果你是一个ActionScript老手，熟悉ActionScript 2.0的绘图API，可以跳过本节，直接学习新增的特性。放心，你不会落下什么知识的。

如果你从来没有用过绘制API，或者是一个新手，本节会带着你快速浏览一下早期版本的ActionScript和Flash就有的一些绘图方法。本节内容不是很多，也不是很详细，因为大部分内容都是很古老的知识，本章重点放在现在流行的新增功能上。不过，这些知识可以让你快速进入状态。

绘图API就像一支虚拟的钢笔，可以用来绘制直线或曲线到指定的坐标，还可以给这些线条填充颜色或位图。所有的绘制动作都通过一个Graphics对象和它的方法实现，只有Sprite或Shape对象中才包含Graphics对象，Graphics对象不需要实例化（不过实际上是可以实例化的），可以通过Sprite或Shape的Graphics属性获取。也就是说要使用绘图API，首先要有一个Sprite或Shape对象，而且这个对象要添加到显示列表中。然后通过这个Sprite或Shape对象的Graphics属性进行绘制。

```
var sprite:Sprite = new Sprite();
sprite.graphics.moveTo(50,50);
```

这个Sprite（可以包含其他显示对象）中用绘图API绘制的内容，渲染在所有子对象的下面。

Flash影片、游戏或者应用中的任何可视对象都必须添加到显示列表中。显示列表是一个包含所有可渲染对象的层次分明的列表，最底部的一层叫做显示对象的root。在Flash或ActionScript项目，用Sprite或MovieClip作为文档类的SWF，它的root和主文档类的root是一样的，会自动出现在显示列表中（如果是Flex框架，主文档类也会自动添加到显示列表中，通过root属性页可以访问到最底部的显示对象）。

显示列表可以保存任何的DisplayObject对象。DisplayObject是一个虚拟类，它的具体子类有Bitmap、TextField、Video、Shape和Sprite（也是MovieClip的父类）。实例化这些类对象后，就可以添加到显示列表中了，否则Flash不会渲染。具体的操作是调用DisplayObjectContainer（本身继承DisplayObject）的addChild()或addChildAt()方法，把这些对象添加到DisplayObjectContainer中，作为它的子对象，而且这个DisplayObjectContainer必须在显示列表中。

因为本书涉及很多生成和处理图像的内容，所以这里跳过具体的ActionScript代码实现过程。为了包含更多的内容，一些基本的ActionScript和Flash Player特性（如使用显示列表）没有包含在本书中。更多深入介绍ActionScript的内容请参考Adobe的ActionScript 3.0文档，或者像由friends of ED出版的*Function ActionScript 3.0 with Flash CS3 and Flex*（ISBN:978-1-59059-815-3）（我也参与了这本书的编写）这样的书。

开始绘制后，虚拟的钢笔会出现在Sprite或Shape指定在坐标位置。初始时在显示对象的(0,0)位置，然后可以用绘图命令移动。接下来，我们看一些基本的绘图命令，学习一下如何用这些命令操作虚拟的钢笔。

基本命令回顾

首先从八个简单的绘图命令开始，下面我们逐一进行学习。后面没有课后习题，不过假设有一个也是不错的。

1. 绘制直线

下面的moveTo()方法将虚拟钢笔移动到指定的位置，但不绘制任何线条：

```
moveTo(x:Number,y:Number):void
```

就像钢笔被某种魔力（或者是你充满魔力的手）拿起离开纸面，然后在一个新的位置放下一样。如果要绘制不连贯的线段，或者移出初始的(0,0)坐标，都需要用到这个方法。

```
lineTo(x:Number,y:Number):void
```

lineTo()方法会从钢笔当前的位置，向参数中设定的位置绘制一条点到点的直线。

lineStyle()方法用来设置线条的虚拟属性，如粗细和颜色。

```
lineStyle(
  thickness:Number=NaN,
  color:uint=0,
  alpha:Number=1.0,
  pixelHinting:Boolean=false,
  scaleMode:String="normal",
  caps:String=null,
  joints:String=null,
  miterLimit:Number=3
):void
```

这个方法有很多参数，它们的功能分别如下。

- thickness：设置绘制线条的粗细。参数值为0表示极细的笔触，且不会随Graphics对象缩放而缩放。
- color：指定绘制线条的颜色。
- alpha：指定绘制线条的透明度。
- pixelHinting：参数指定线条和端点绘制时是否使用完整像素和完整的宽度（true），或者使用部分像素，如线条的粗细设置为0.5，而锚点在(1.5, 1.5)的位置。
- scaleMode：这个参数设置笔触的粗细是否随着绘制该笔触的显示对象的缩放而缩放。它的取值可以从LineScaleMode类的NORMAL、NONE、VERTICAL或HORIZONTAL属性中选择：NORMAL表示总是随着显示对象缩放粗细，NONE表示从不缩放，HORIZONTAL和VERTICAL分别表

示只在水平或垂直方向上随显示对象缩放。

● caps：这个参数指定了线条端点的类型。它的取值可以是CapsStyle类的NONE（端点无任何样式）、ROUND（端点为圆形）和SQUARE（端点为方形）静态属性的其中之一。

● joints：这个参数指定了线条弯曲时拐角处的样式。它的有效取值是JointStyle类的静态属性——MITER（如果端点的尺寸在尖角限制以内，则创建一个端点，否则用斜角连接）、ROUND（圆角连接）或BEVEL（斜角连接）。关节和端点的图示请参考图1.1。

● miterLimit：这个参数和关节的MITER结合着使用，用来指定线条拐角处的向外延伸的两边相交形成的结合点的长度。图1.2展示了相同角度拐角的三种不同的斜接限制。

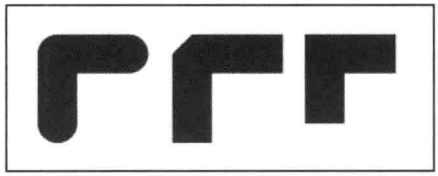

图1.1 线条的端点和关节分别设置为ROUND（左）、SQUARE与BEVEL（中）、NONE与MITER（右）

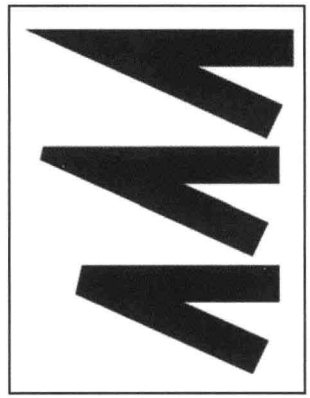

图1.2 相同的线条三种不同的斜接限制（自上而下的斜接限制分别为5、3、0）

在绘制线条之前，要设置好线条的样式，这正是lineStyle()要做的。lineStyle()方法要在lineTo()或curveTo()之前调用，否则绘制的线条粗细为0，也就是不可见。这里介绍的lineStyle()只需要前三个参数，分别是以像素为单位的线条粗细、颜色和透明度；其他参数是在Flash 8时代添加进来的，用来控制线条的端点是否使用完整像素、在父级容器缩放时如何缩放线条，以及如何渲染线条的端点和拐角。

```
clear():void
```

如果要清除绘制的图形，clear()方法可以清除所有绘制的线条和填充，还会把虚拟钢笔的坐标重置为(0,0)，并清除线条样式。

要测试这四个方法，可以用下面的代码，或者是本章源文件DrawingStraightLines.as类中带有完整注释的代码。可以直接运行源码中编译好的SWF文件，也可以编译下面的代码，在Flash或Flex Builder中设置和编译代码的具体方法请参考附录。运行SWF文件，在舞台上点击并拖动几