

Flash 中利用 ActionScript3.0 实现图像的扭曲效果

陈银凤

(内蒙古财经大学 计算机信息管理学院, 内蒙古 呼和浩特 010051)

摘要: 主要在 Flash CS3 环境中利用 ActionScript3.0 脚本语言的 BitmapData 类和 DisplacementMapFilter 滤镜来实现图像的扭曲效果。

关键词: Flash; ActionScript; BitmapData; 扭曲

中图分类号: TP37

文献标识码: A

文章编号: 1674-7720(2012)16-0035-03

Using ActionScript 3.0 to realize distorted effect of images in Flash

Chen Yin Feng

(Department of Computer Information and Management, Inner Mongolia University of Finance and Economics, Hohhot 010051, China)

Abstract: This paper uses BitmapData class and the Displacement Map Filter filter of ActionScript3.0 scripting language which in Flash CS3 environment to design and implement image distorted effect.

Key words: Flash; ActionScript; bitmapdata; distort

Flash 与 Dreamweaver、Fireworks 一起被称作“网页制作三剑客”^[1]。随着网络技术的不断发展,Flash 已成为当今网络世界中的热点技术,其使用者可以充分发挥自己的想象力,设计和创作出具有图形、声音、视频和动画的各种交互内容,从而使广告、网站、游戏、贺卡、教学课件及音乐动画等真正独具特色。

与其他动画制作软件相比,Flash 制作的动画容量小、操作方便、绿色免安装,非常适合在网络中传播。借助 ActionScript 能够制作出更为细腻、逼真的动画。ActionScript 是 Flash 专用的脚本语言,采用面向对象的编程思想,通常以动画中的关键帧、按钮实例、影片剪辑实例作为对象来对 ActionScript 进行定义和编写,以帮助用户按照自己的意愿更加精确地创建动画,实现自己的意图。ActionScript 由 Flash Player 中的 ActionScript 虚拟机 (AVM) 来解释执行,其代码通常被编译器编译成“字节码格式”,字节码嵌入 SWF 文件中,SWF 文件由运行时环境 Flash Player 执行。ActionScript 的版本已从 1.0 发展到如今的 3.0,虚拟机也从 AVM 发展为支持 ActionScript 3.0 的新一代 AVM2,其执行速度是之前 ActionScript 版本的 10 倍。ActionScript 3.0 旨在方便创建拥有大型数据集和面向对象的 reusable 代码库的高度复杂应用程序。

本文主要阐述在 Flash CS3 环境中利用 ActionScript3.0 的 BitmapData 类和 DisplacementMapFilter 滤镜实

现图像的扭曲效果。Flash CS3 是 Adobe 公司收购 Macromedia 公司后将 Macromedia Flash 更名为 Adobe Flash 后的一款动画软件。

1 BitmapData 类和 DisplacementMapFilter 滤镜

1.1 BitmapData 类

使用 BitmapData 类可以处理 Bitmap 对象的数据 (像素)。可以使用 BitmapData 类的方法创建任意大小的透明或不透明位图图像,并在运行时采用多种方式操作这些图像。也可以访问使用 flash.display.Loader 类加载的位图图像的 BitmapData。此类允许将位图呈现操作与 Flash Player 的内部显示更新例程分隔开来。通过直接操作 BitmapData 对象,用户可以创建非常复杂的图像,不会因连续重新绘制矢量数据的内容而产生每帧开销。BitmapData 类的方法支持多种无法通过通用滤镜接口获得的效果。BitmapData 对象包含像素数据的数组。该数据可以表示完全不透明的位图,或表示包含 Alpha 通道数据的透明位图。以上任一类型的 BitmapData 对象都作为 32 bit 整数的缓冲区进行存储。每个 32 bit 整数确定位图中单个像素的属性。

每个 32 bit 整数都是 4 个 8 bit 通道值 (从 0~255) 的组合,这些值描述像素的 Alpha 及红色、绿色、蓝色的 (ARGB) 值。当将 4 个通道 (红色、绿色、蓝色和 Alpha) 与 BitmapData.copyChannel() 方法或 DisplacementMapFilter.

欢迎网上投稿 www.pcachina.com 37

图形、图像与多媒体

Image Processing and Multimedia Technology

componentX 属性和 DisplacementMapFilter.componentY 属性一起使用时,这些通道以数字形式表示,分别为:1(红色)、2(绿色)、4(蓝色)和 8(Alpha)。

(1)构造函数

BitmapData(width:Number, height:Number, [transparent:Boolean], [fillColor:Number])

(2)公共属性

height:Number[只读]:位图图像的高度,以像素为单位。

rectangle:Rectangle[只读]:定义位图图像大小和位置的矩形。

transparent:Boolean [只读]:定义位图图像是否支持每个像素具有不同的透明度。

width:Number[只读]:位图图像的宽度,以像素为单位。

1.2 DisplacementMapFilter 滤镜

DisplacementMapFilter 类使用指定的 BitmapData 对象(称为置换图图像)的像素值执行对象置换。可以使用此滤镜将扭曲或斑点效果应用于从 DisplayObject 类中继承的任何对象,例如 MovieClip、SimpleButton、TextField、Video 对象以及 BitmapData 对象。

DisplacementMapFilter 需要有一个灰度图像作为置换映射图像,并基于这个映射图像的像素对另外一个图像进行扭曲。映射图像中比中性灰亮的像素在目标图像中对应位置上向左上角移动,较暗的像素则向右下角移动。

(1)构造函数

DisplacementMapFilter (mapBitmap:BitmapData =null, mapPoint:Point =null, componentX:uint =0, componentY:uint =0, scaleX:Number=0.0, scaleY:Number=0.0, mode:String="wrap", color:uint=0, alpha:Number=0.0)

(2)公共属性

alpha: Number //指定对于超出范围的替换应用的 Alpha 透明度值

color:uint //指定对于超出范围的替换应用什么颜色

componentX:uint //说明在映射图像中使用哪个颜色通道来置换 x 结果

componentY:uint //说明在映射图像中使用哪个颜色通道来置换 y 结果

mapBitmap:BitmapData //包含置换映射数据的 BitmapData 对象

mapPoint:Point //一个值,它包含目标显示对象的左上角相对于映射图像左上角的偏移量

mode:String //滤镜模式

scaleX:Number //用于缩放映射计算的 x 置换结果的乘数

scaleY:Number //用于缩放映射计算的 y 置换结果的乘数。

2 应用置换映射滤镜创建图像扭曲效果

使用 BitmapData 对象^[1]和 DisplacementMapFilter 滤镜

可以实现图像的扭曲效果。DisplacementMapFilter 类使用 BitmapData 对象(称为置换图图像)中的像素值在新对象上执行置换效果。通常,置换图图像与将要应用滤镜的实际显示对象或 BitmapData 实例不同。置换效果包括置换过滤的图像中的像素,也就是说,将这些像素移开原始位置一定距离。

实现步骤:

(1)新建一个 Flash 文档,设置其舞台大小为 340×220 像素,背景为白色。

(2)选中图层 1,命名为“图像层”,导入准备好的一张图片到舞台,图片大小设置为 164×220 像素,选中该图片并转化为影片剪辑,元件名为“image”。选中舞台中的 image 影片剪辑实例,命名为“pic”,并在舞台中靠左放置。从库中再次拖入一个 image 影片剪辑实例,无需命名,舞台中靠右放置(为与扭曲效果作对比)。

(3)菜单栏中点击“插入/新建元件”,选择影片剪辑,元件命名为“displacer”,进入元件编辑环境。在图层 1 中,使用矩形工具绘制一个 164×220 的无边框灰色矩形,如图 1 所示。新建图层 2,在右下角绘制一个直径为 108 像素的正圆,颜色填充为放射状(3 个色块均匀分布,从左到右颜色值分别为:#956A89、#6B1452、#000000;Alpha 值分别为:100%、50%、0%),如图 2 所示。

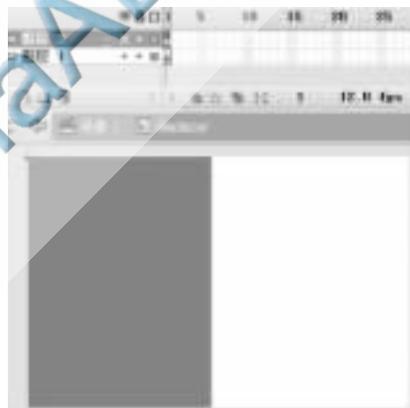


图 1 影片剪辑 displacer(1)

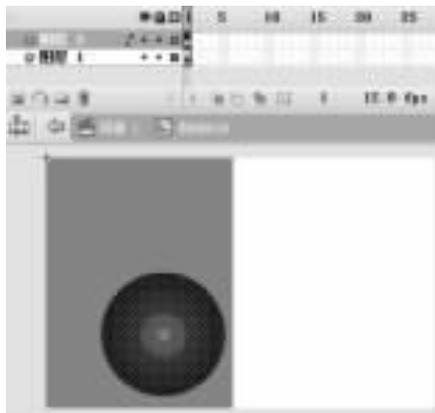


图 2 影片剪辑 displacer(2)

图形、图像与多媒体

(4)回到场景中,新建图层 2,命名为“displacer 层”,将 displacer 影片剪辑拖入舞台,影片剪辑实例命名为“dis”。其正好放置在“pic”实例的正上方。

(5)新建图层 3,命名为“AS”,选中第一帧,打开“动作”面板,输入如下代码:

```
import flash.display.BitmapData;
import flash.geom.Point;
import flash.filters.DisplacementMapFilter;
    //沿着 x 轴替换通道 1(红色)上的像素
var componentX:Number = 1;
    //沿着 y 轴替换通道 2(绿色)上的像素
var componentY:Number = 2;
    //沿着 x 轴替换的强度
var scale_X:Number = -128;
    //沿着 y 轴替换的强度
var scale_Y:Number = -128;
    //使用 ignore 模式可以显示下面扭曲的图像
var mode:String = "ignore";           //偏移量
var offset:Point = new Point(0, 0);
    //构建一个空的 BitmapData 实例,并将影片剪辑实例读进该实例
var myBitmapData:BitmapData=new BitmapData(pic.width,
    pic.height);
    //创建一个 DisplacementMapFilter 滤镜实例
var myFilter:DisplacementMapFilter = new Displacement
    MapFilter(myBitmapData,offset,componentX,componentY,
    scale_X, scale_Y, mode);
myBitmapData.draw(dis);//为影片剪辑实例 pic 应用滤镜
pic.filters = [myFilter];
dis.visible = false;
```

(4)保存文档,按 Ctrl+Enter 键进行执行,测试图像的扭曲效果。如图 3 左侧所示,影片剪辑实例 pic 出现扭曲效果。

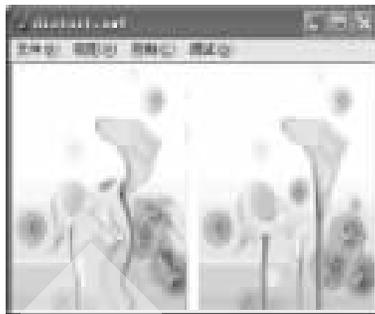


图 3 影片剪辑 pic 的扭曲效果

由于 Flash 制作出的动画体积小,非常适合在网络中传播,所以深受广大网民的喜爱。要想在 Flash 环境中制作出效果逼真的动画,ActionScript 是不可或缺的部分。ActionScript 是针对 Flash Player 运行时环境的编程语言,它用于处理各种人机交互、数据交互等功能,目前主要采用的版本是 ActionScript3.0。ActionScript3.0 与 Flash CS3(或以上版本)相结合,可以创建出各种不同的应用特效,实现丰富多彩的动画效果。本文中正是利用 ActionScript3.0 的 BitmapData 类和 DisplacementMapFilter 滤镜实现了图像的扭曲效果。Flash 中专业制作动画者可以借助 ActionScript3.0,更加充分地发挥想象力,随心所欲地制作出交互性极强的复杂动画,体现美仑美奂的意境。

参考文献

- [1] 张亚飞.Flash ActionScript 3.0 开发王[M].北京:电子工业出版社,2008.
- [2] 王传华.Flash MX 2004 实例教程[M].北京:清华大学出版社,2004.

(收稿日期:2012-05-23)

作者简介:

陈银凤,女,1978年生,硕士,讲师,主要研究方向:计算机网络、电子商务和数据库。