



第8章

ActionScript 的应用

ActionScript 代码通常被编译器编译成“字节码格式”(一种由计算机编写且能够为计算机所理解的编程语言)。ActionScript 脚本撰写语言允许向应用程序添加复杂的交互性、播放控制和数据显示。可以使用动作面板、“脚本”窗口或外部编辑器在创作环境内添加 ActionScript。ActionScript 遵循自身的语法规则和保留关键字,并且允许使用变量存储和检索信息。ActionScript 含有一个很大的内置类库,使用户可以通过创建对象来执行许多有用的任务。

可使用“动作”面板来编写放在 Animate 文档中的脚本(即嵌入 FLA 文件中的脚本)。“动作”面板提供了一些功能(如“动作”工具箱),让用户能够快速访问核心 ActionScript 语言元素。用户会收到创建脚本所需元素的提示。

本章将学习 Animate 软件中 Action Script 的作用、Action Script 3.0 的特点、Action Script 语法规则、“代码片段”面板的使用方法和动作脚本的添加。



学习目标

- 了解 Action Script 的作用。
- 理解 Action Script 3.0 的特点。
- 掌握 Action Script 语法规则。
- 掌握“代码片段”面板的使用方法。
- 掌握动作脚本的添加。
- 通过制作绿水青山相册,让大家通过学习动画制作的同时欣赏到了山青水秀的风景;通过制作刘三姐画册制作,让大家通过学习动画制作的同时欣赏到了动听的壮族民歌和美丽的山水画卷,培养学生的空间想象能力和创新意识,形成正确、规范的思维方式和分析方法。



● 视频



交互电子相册制作

案例 8-1 图片选择代码应用——交互电子相册制作



情境导入

在当今社会，电子相册影像已成为人们生活和工作中日益追求的物质和精神需求，随着数码摄影时代的到来，无论是专业摄影师建立图片档案或是向他人展示自己的摄影作品，还是家庭生活摄影，都需要电子相册来保管摄影作品。



案例说明

电子相册动画制作主要图片选择代码应用来实现。



相关知识

一、认识交互式动画

交互动画是指在动画作品播放时支持事件响应和交互功能的一种动画。就是说，动画播放时可以接受某种控制，这种控制可以是动画播放者的某种操作，也可以是在动画制作时预先准备的操作。这种交互性提供了观众参与和控制动画播放内容的手段，使观众由被动接受变为主动选择。

最典型的交互式动画就是Animate动画。观看者可以用鼠标或键盘对动画的播放进行控制。

Animate动画交互性就是用户通过菜单、按钮、键盘和文字输入等方式，来控制动画的播放。交互性是为了用户和计算机之间产生互动性，使计算机对指示做出相应的反应。交互式动画就是动画在播放时支持事件响应和交互功能的一种动画，动画在播放时不是从头播到尾，而是可以接受用户的控制。

二、ActionScript 3.0的新增功能

Action Script 3.0是Animate的编程语言，与之前的版本有着本质上的不同，它是一门功能强大、符合业界标准的面向对象的编程语言。ActionScript 3.0新增了很多独有的功能，非常适合因特网应用程序开发。

核心语言定义编程语言的基本构成块，如语句、条件、表达式、循环和类型。

ActionScript 3.0实现了ECMAScript for XML (E4X)，最后被标准化为ECMA-357。E4X提供一组用于操作XML的自然流畅的语言构造。

ActionScript 3.0编辑器借助内置ActionScript 3.0编辑器提供的自定义类代码提示和代码完成功能，简化开发作业，可有效地参考本地或外部的代码库。

三、ActionScript 3.0常用术语

Actions “动作”：用于控制影片播放的语句。

Classes “类”：用于定义新的对象类型。

Constants “常量”：是不变的元素。

Constructors “构造函数”：用于定义一个类的属性和方法。

Data types “数据类型”：用于描述变量或动作脚本元素可以包含的信息种类。

Events “事件”：是在动画播放时发生的动作。

Expressions “表达式”：具有确定值的数据类型的任意合法组合，由运算符和操作数组成。

Functions “函数”：是可重复使用的代码块，它可接受参数并返回结果。

Identifiers “标识符”：用于标识一个变量、属性、对象、函数或方法。

Instances “实例”：是一个类初始化的对象。每一个类的实例都包含这个类中所有属性和方法。

Instance names “实例名”：脚本中用于表示影片剪辑实例和按钮实例的唯一名称。可以通过【属性】面板为舞台上的实例指定实例名称。

Keywords “关键字”：是有特殊意义的保留字。

Methods “方法”：是与类关联的函数。

Objects “对象”：是一些属性的集合。每一个对象都有自己的名称，并且都是特定类的实例。

Operators “运算符”：通过一个或多个值计算新值。

Parameters “参数”：用于向函数传递值的占位符。

Properties “属性”：用于定义对象的特性。

Target paths “目标路径”：动画文件中，影片剪辑实例、变量和对象的分层结构地址。

Variables “变量”：用于存放任何一种数据类型的标示符，可以定义、改变和更新变量，也可在脚本中引用变量的值。

四、ActionScript 3.0常用语法规则

1. 区分大小写

在动作脚本中的语句除了关键字区分大小写外，其他ActionScript 3.0语句大小写可以混用，但根据书写规范进行输入，可以使ActionScript 3.0语句更容易阅读。

对于关键字、类名、变量、方法名等，要严格区分大小写。如果关键字的大小写出现错误，在编写程序时就会有错误信息提示。如果采用了彩色语法模式，那么正确的关键字将以蓝色显示。

2. 点运算符

动作脚本中的语句，点“.”用于指示与对象相关的属性或方法。通过点语法可以引用类的属性或方法。例如：

```
var Company:Object = {};           //新建一个空对象，将其引用赋值给变量Company
Company.name = "企鹅";             //新增一个属性name，将字符串“企鹅”赋值给它
Trace(Company.name);               //输出“企鹅”
```

3. 界定符

- (1) 大括号。动作脚本中的语句可被大括号包括起来组成语句块，用于将代码分成不同的块。
- (2) 小括号。通常用于放置使用动作时的参数，在定义或调用函数时都要使用小括号。
- (3) 分号。在动作脚本中的语句的结束处添加分号，表示该语句结束。虽然不添加分号也可以正常运行语句，但使用分号可以使语句更易于阅读。

4. 注释

在语句后面添加注释有助于用户理解动作脚本的含义，以及向其他开发人员提供信息。添加注释的方法是先输入两个斜杠“//”，然后输入注释的内容即可。注释以灰色显示，长度不受限制，也不会影响语句的执行。例如：



```
Public Function myDate () ( //创建新的Date对象
Var myDate:Date = new Date ();
CurrentMonth = myDate.getMonth (); //将月份数转换为月份名称
monthName= calcMonth (currentMonth);
year = myDate.getFullYear ();
currentDate = myDate.getDate ();
}
```

5. 关键字和标识符

现实生活中，所有事物都有自己的名字，从而与其他事物区分开。在程序设计中，也常常用一个记号对变量、方法和类等进行标示，这个记号就称为标识符。动作脚本保留一些单词用于表示该语言的特定用途，因此不能将它们用作变量、函数或标签的名称。如何在编写程序的过程中使用关键字，动作编辑框中的关键字会以蓝色显示。为了避免冲突，在命名时可以展开动作工具箱中的Index域，检查是否使用了已定义的名称。

标识符的命名必需符合一定的规范，在语言中，标识符的第一个字符必须为字母、下划线或美元符号，后面的字符可以是数字、字母、下划线或美元符号。

五、数据与运算

1. 常量

“常量”是程序运行过程中数值恒定不变的量。在ActionScript 3.0中可以使用const关键字进行声明，并且“常量”只能在声明时直接赋值。一旦赋值，就不再改变。使用ActionScript 3.0编程时，建议能使用“常量”的就尽量使用“常量”。

“常量”声明格式如下：

```
const 常量名:数据类型 = 值
```

2. 变量

(1) “变量”的定义：

“变量”是为了存储数据而创建的。“变量”就像一个容器，用于容纳各种不同类型的数据。当然对变量进行操作，“变量”的数据就会发生改变。

“变量”必须先声明后使用，否则编译器就会报错。例如，现在要去喝水，首先要有一个杯子，否则怎么样去装水呢？要声明“变量”的原因与此相同。

(2) “变量”的命名规则：

①它必须是一个标识符。第一个字符必须是字母、下划线(_)或美元符号(\$)。其后的字符必须是字母、数字、下划线或美元符号。不能使用数字作为变量名称的第一个字符。

②它不能是关键字或动作脚本文本，如true、false、null或undefined。特别不能使用ActionScript 3.0的保留字，否则编译器会报错。它在其范围内必须是唯一的，不能重复定义。

(3)“变量”类型。在使用“变量”之前，应先制定存储“数据”的类型，“数值类型”将对变量产生影响。

在Animate CC 中，系统会在给“变量”赋值时自动确定“变量”的“数据类型”。

①“字符串变量”：该变量主要用于保存特定的文本信息，如姓名。

②“对象性变量”：用于存储对象型的数据。

③“逻辑变量”：用于判定指定的条件是否成立。其值有两种，true和false。true即真，表示条件成

立；false即假，表示条件不成立。

④ “数值型变量”：一般用于存储特定的数值，如日期、年龄。

⑤ “电影片段变量”：用于存储电影片段类型的数据。

⑥ “未定义型变量”：当一个变量没有赋予任何值时，即为未定义型变量。

(4) 变量的作用域。“变量”的作用域是指变量能被识别和应用的区域。根据“变量”的作用可将其分为全局变量和局部变量。

① 全局变量。全局变量是指在代码的所有区域中定义的变量。全局变量在函数定义的内部和外部均可使用。

② 局部变量。局部变量是指仅在代码的某部分定义的变量。在函数内部声明的局部变量仅存在于该函数中。

3. 数据类型

(1) 布尔类型。布尔类型（Boolean）包含两个值：true和false。对于Boolean类型的变量，其他任何值都是无效的。已经声明但尚未初始化的布尔变量的默认值是false。

(2) 字符串类型。字符串类型可以使用单引号和双引号来声明字符串，也可以使用String的构造函数来生成。

(3) Number数据类型。Number数据类型是双精度浮点数。数字对象的最小值约为5E-324，最大值约为1.79E+308。

(4) Null数据类型。Null数据类型只有一个值，即null，此值意味着没有值，即没有数据。在很多情况下，可以指定null值，以指示某个属性或变量尚未赋值。

六、事件

1. 鼠标事件

单击：MouseEvent.click。

MouseEvent.couple_click。

按键状态：

MouseEvent.mouse_down。

MouseEvent.mouse_up。

鼠标悬停或移开：

MouseEvent.mouse_over。

MouseEvent.mouse_out。

MouseEvent.roll_over。

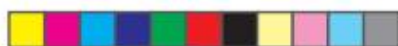
MouseEvent.roll_out。

鼠标移动：MouseEvent.mouse_move。

鼠标滚轮：MouseEvent.mouse_wheel。

当前鼠标的坐标：相对坐标local X、local Y；舞台坐标stage X、stage Y。

相关按键是否按下，Boolean类型；alt key、ctrl key、shift key、button down鼠标主键，一般情况为左键。



2. 关键帧事件

将动作脚本添加到关键帧上时，只需选中关键帧，然后在【动作】面板中输入相关动作脚本即可，添加动作脚本后的关键帧上会出现一个“α”符号，如图8-1所示。

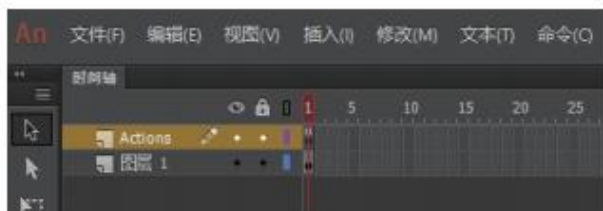


图8-1 动作脚本添加到关键帧上

3. 影片剪辑事件

(1) 实例名称。这里所指的实例包括影片剪辑实例、按钮元件实例、视频剪辑实例、动态文本实例和输入文本实例，它们是Animate CC 动作脚本面板的对象。

(2) 绝对路径。使用绝对路径时，不论在哪个影片剪辑中进行操作，都是从场景的时间轴出发，到影片剪辑实例，再到下一级的影片剪辑实例，一层一层地往下寻找，每个影片剪辑实例之间用“.”分开。

(3) 相对路径。“相对路径”是以当前实例为出发点，来确定其他实例的位置。

4. 使用函数

(1) 调用内置函数。内置函数是执行特定任务的函数，可用于用户访问特定的信息。

(2) 向函数传递参数。参数是某些函数执行其代码所需要的元素。

(3) 从函数返回值。使用return语句可以从函数中返回值。return语句将停止函数运行并使用return语句的值替换它。

5. 自定义函数

(1) 用户可以把执行自定义功能的一系列语句定义为一个函数。该函数可以有返回值，也可以从任意一个时间轴中调用它。

(2) 用户可以使用目标路径从任意时间轴中调用任意时间轴内的函数。如果函数是使用_global标识符声明的，则无须使用目标路径即可调用它。

七、动画的跳转

1. 循环语句的使用

(1) while循环。如果用户要在条件成立时重复动作，可使用while语句。

while循环语句可以获得一个表达式的值，如果表达式的值为true，则执行循环体中的代码。在主体重的所有语句都执行之后，表达式将再次被取值。

(2) do...while语句。使用do...while语句可以创建与while循环相同类型的循环。在do...while循环中，表达式在代码块的最后，这意味着程序将在执行代码块之后才会检查条件，所以无论条件是否满足循环都至少会执行一次。

(3) for语句。如果用户要使用内置计数器重复动作，可使用for语句。多数循环都会使计数器以控

制循环执行的次数。每执行一次循环就称为一次“迭代”，用户可以声明一个变量并编写一条语句，每执行一次循环，该变量都会增加或减小。在for动作中，计数器和递增计数器的语句都是该动作的一部分。

(4) for...in语句。使用for...in循环可以循环访问对象属性或者数组元素（不按任何特定的顺序来保存对象的属性，因此属性可能以看似随机的顺序出现）。

(5) for each...in语句。for each...in循环用于循环访问集中的项目，它可以是对象中的标签、对象属性保存的值或数组元素。

2. 条件语句的使用

(1) if...else控制语句。if...else控制语句是一个判断语句。该语句的调用格式有如下3种。

```
if(condition1){statement(s1);} //格式1
if(condition1){statement(s1);}else{statement(s2);} //格式2
if(condition1){statement(s1);}else if(condition2){statement(s2);} //格式3
```

(2) if...else if控制语句。if...else if控制语句可以用来测试多个条件。

(3) switch...case控制语句。switch...case控制语句是多条件判断语句，也是创建ActionScript语句的分支结构。像if动作一样，switch动作测试一个条件，并在条件返回true值时执行语句。

八、动作脚本的添加

ActionScript 3.0发生了重大变化，代码只能写在帧和AS类文件中。在实际开发过程中，如果把代码写在帧上会导致代码难以管理，因此建议用AS类文件来组织代码，这样可以使设计与开发分离，利于协同工作。

1. 给关键帧添加代码

打开【动作】面板或按【F9】键，直接在控制面板中输入代码。

2. AS类文件

选择【文件】|【新建】命令，弹出【新建文档】对话框，选择“ActionScript 3.0类”选项，即可创建一个外部类文件。

案例实施

- (1) 运行Animate CC 软件，选择【新建】|【ActionScript 3.0】选项，新建一个文件。
- (2) 在【属性】面板的【位置和大小】区域舞台大小为宽700像素，高400像素。
- (3) 选择【文件】|【导入】|【导入到舞台】命令（快捷键【Ctrl+R】），导入图片。
- (4) 选择【插入】|【新建元件】命令（快捷键【Ctrl+F8】）。
- (5) 在【创建新元件】对话框中，名称输入“1”，类型选择“按钮”，单击【确定】按钮，完成元件的创建。
- (6) 把“风景1”图片拖到舞台中，图片大小设置为宽100像素，高72像素，完成元件1的创建，如图8-2所示。
- (7) 依此类推，给所有图片分别创建按钮元件。
- (8) 把“图层1”重命名为“风景”，把“风景1”图片拖到舞台，在【位置和大小】区域设置X为“150”、Y为“0”，如图8-3所示。



图8-2 新建按钮元件



图8-3 图片位置属性

(9) 在第2帧处按【F7】键插入空白关键帧，把“风景2”图片拖到舞台中，在【位置和大小】区域设置X为“150”、Y为“0”。

(10) 依此类推，在第3、4、5帧处把“风景3”“风景4”“风景5”图片拖到舞台中，在【位置和大小】区域设置X为“150”、Y为“0”。

(11) 新建图层，命名为“按钮”。绘制一个宽150像素，高400像素的矩形。绘制黑色的圆，制作胶卷效果，如图8-4所示。



图8-4 绘制矩形

(12) 选择【视图】|【标尺】命令，显示标尺，拖出一条辅助线到舞台。

(13) 把元件“1”拖到舞台中，实例名称定义为“a”，如图8-5所示。

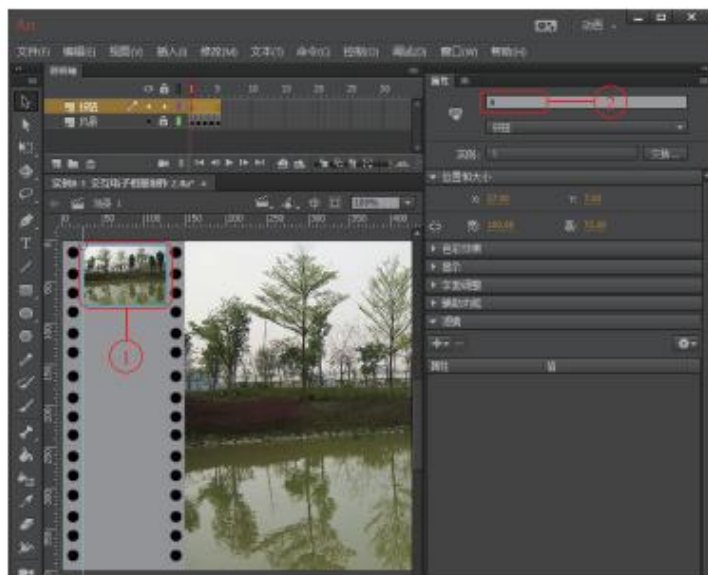


图8-5 定义实例名称 (1)

(14) 依此类推，把元件“2”拖到舞台中，实例名称定义为“b”；把元件“3”拖到舞台中，实例名称定义为“c”；把元件“4”拖到舞台中，实例名称定义为“d”；把元件“5”拖到舞台中，实例名称定义为“e”，如图8-6所示。



图8-6 定义实例名称 (2)

(15) 新建图层，命名为“代码”，选择第1帧，选择【窗口】|【动作】命令（或按【F9】键），打开【动作】面板，如图8-7所示。

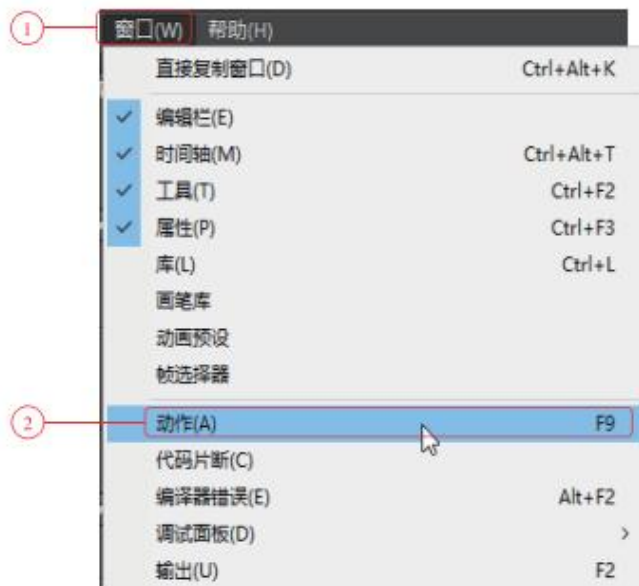


图8-7 打开“动作”面板

(16) 在【动作】面板中输入图8-8所示代码。

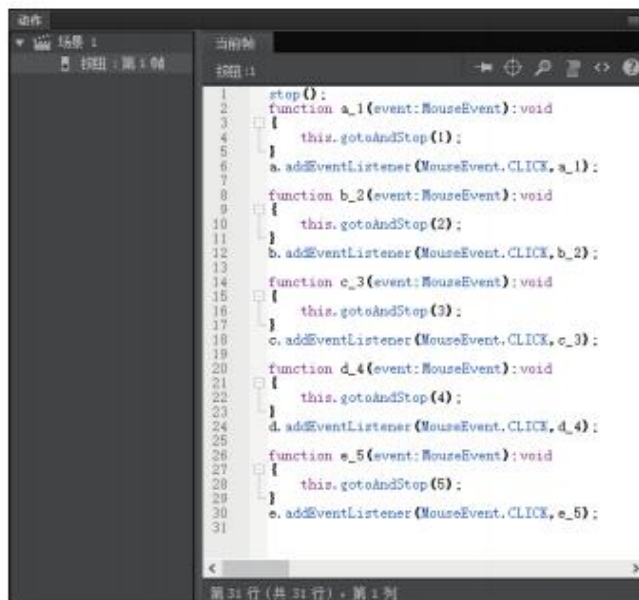


图8-8 点击转换图片动作代码

(17) 保存文件，按【Ctrl+Enter】组合键进行影片测试。

案例 8-2 播放与重播代码应用—刘三姐画册制作

情境导入

刘三姐的传说

相传唐代，在广西罗城与宜州交界处的天洞之滨，有个美丽的小山村（现罗城仫佬族自治县四把镇蓝靛村）。村中有一位叫刘三姐的壮族姑娘，她自幼父母双亡，靠哥哥刘二抚养，兄妹俩二人以打柴、捕鱼为生，相依为命。三姐不但勤劳聪明，更是纺纱织布的好手，而且长得宛如出水芙蓉一般，容貌绝伦。她尤其擅长唱山歌，她的山歌遐迩闻名，故远近歌手经常聚集其村，争相与她对歌、学歌。

刘三姐常用山歌唱出穷人的心声和不平，故而触犯了土豪劣绅的利益。当地财主莫怀仁贪其美貌，欲占为妾，遭到她的拒绝和奚落，便怀恨在心。莫怀仁企图禁歌，被刘三姐用山歌驳得理屈词穷。又请来三个秀才与刘三姐对歌，又被刘三姐等弄得丑态百出，大败而归。莫怀仁恼羞成怒，不惜耗费家财去勾结官府，咬牙切齿把刘三姐置于死地而后快。为免遭毒手，三姐偕同哥哥在众乡亲的帮助下，趁天黑乘竹筏，顺流沿天河直下龙江后入柳江，辗转来到柳州，在小龙潭村边的立鱼峰东麓小岩洞居住。

据说来到柳州以后，三姐那忠厚老实的哥哥刘二心有余悸，怕三姐又唱歌再招惹是非，便想方设法来阻止。一天，他终于想出了个办法，从河边捡回一块又圆又厚的鹅卵石丢给三姐，说：“三妹，用你的手帕角在石头中间钻个洞，把手帕穿过去！若穿不过去，就不准你出去唱歌！”接着铁青着脸一字一顿地补充道：“为兄说不二，决无戏言。”

先还是甜甜微笑的三姐，看着哥哥的满脸愠色，不敢像往常那样据理争辩，拾起丢在面前的石头，暗忖道：“我又不是神仙，手帕角怎能穿得过去？”她下意识地试穿，并唱道：哥发癫，拿块石头给妹穿，软布穿石怎得过？除非凡妹变神仙！

“管你是凡人也好，神仙也好，为兄一言既出，绝不更改！”哥哥像是吃了秤砣——铁了心。心想：这一招够绝了吧，还难不倒你？

谁料三姐凄切婉转的歌声直上霄汉，传到了天宫七仙女的耳里。七仙女非常感动，恐三姐从此歌断失传，于是施展法术，从发上取下一根头发簪甩袖向凡间刘三姐手中的石块射去，不偏不歪，把石头穿了一个圆圆的洞。三姐无意中见手帕穿过石头，心中暗喜，张开甜润的嗓子：

哎……穿呀穿，柔能克刚好心欢，歌似滔滔柳江水，源远流长永不断！

从此，刘三姐的歌声又萦回鱼峰山顶、树梢，慕名来学歌的、对歌的人络绎不绝。后来，三姐在柳州的踪迹被莫怀仁侦知。他又用重金买通官府，派出众多官兵将立鱼峰团团围住，来势汹汹，要捉杀三姐。小龙潭村及附近的乡亲闻讯，手执锄头棍棒纷纷赶来，为救三姐而与官兵搏斗。三姐不忍心使乡亲流血和受牵连，毅然从山上跳入小龙潭中……

正当刘三姐纵身一跳的时候，顿时狂风大作，天昏地暗。随着一道红光，一条金色的大鲤鱼从小龙潭中冲出，把三姐驮住，飞上云霄。刘三姐就这样骑着鱼上天，到天宫成了歌仙。而她的山歌，人们仍世代传唱着。为纪念她在柳州传唱的功绩，人们在立鱼峰的三姐岩里，塑了一尊她的石像，

视频



刘三姐画册制作

•



二维动画制作案例教程

一直供奉。

“三月三”，是壮族地区最大的歌圩日，又称“歌仙节”，相传是为纪念刘三姐而形成的民间纪念性节日。1984年农历三月三日，广西壮族自治区人民政府正式将这一天定为壮族的全民性节日——“三月三”歌节。每年的这一天，南宁市及其他各地都要举行盛大的歌节。歌节期间，除传统的歌圩活动外，还要举办抢花炮、抛绣球、碰彩蛋及演壮戏、舞彩龙、擂台赛诗、放映电影、表演武术和杂技等丰富多彩的文体娱乐活动。另外，各种商业贸易、投资洽谈等活动亦逐渐增加，形成“文化搭台，经济唱戏”的新风尚。届时，岭南壮乡四海宾朋云集，歌如海、人如潮。那不绝于耳的嘹亮歌声，寄托着人们对歌仙刘三姐的思念和对丰收、对爱情、对幸福美好生活的憧憬和向往。



案例说明

刘三姐画册制作主要应用播放与重播、停止等代码来实现。



相关知识

一、停止代码

```
Stop();
```

二、播放代码

```
/* 单击以转到下一场景并播放  
单击指定的元件实例会将播放头移动到时间轴中的下一场景并在此场景中继续回放  
*/  
p.addEventListener(MouseEvent.CLICK, fl_ClickToGoToNextScene);  
function fl_ClickToGoToNextScene(event:MouseEvent):void  
{  
    MovieClip(this.root).nextScene();  
}
```

三、重播代码

```
/* 单击以转到前一场景并播放  
单击指定的元件实例会将播放头移动到时间轴中的前一场景并在此场景中继续回放  
*/  
rp.addEventListener(MouseEvent.CLICK, fl_ClickToGoToPreviousScene);  
function fl_ClickToGoToPreviousScene(event:MouseEvent):void  
{  
    MovieClip(this.root).prevScene();  
}
```



案例实施

一、导入素材

(1) 运行Animate CC 软件，选择【新建】|【ActionScript 3.0】选项，新建一个文件。把舞台大小设置为宽800像素，高600像素。

- (2) 选择【文件】|【导入】|【导入到库】命令。
- (3) 找到“第8章 刘三姐画册制作素材”文件夹，选择文件夹中所有文件，如图8-9所示。



图8-9 选择素材

- (4) 单击【打开】按钮，导入素材，在【库】面板中可以看到所有的图片和音乐素材。

二、创建元件

- (1) 选择【插入】|【新建元件】命令（快捷键【Ctrl+F8】）。
- (2) 在【创建新元件】对话框中，输入名称“播放”，类型选择“按钮”，单击【确定】按钮，完成元件的创建。
- (3) 绘制一个绿蓝渐变色矩形，输入“播放”，如图8-10所示。
- (4) 分别在“指针”“按下”“点击”处插入关键帧，在“指针”关键帧处把矩形颜色改成黄紫渐变色，如图8-11所示。



图8-10 按钮元件弹起状态



图8-11 按钮元件指针状态

- (5) 依此类推，完成“重播”按钮元件制作。
- (6) 选择【插入】|【新建元件】命令（快捷键【Ctrl+F8】）。



(7) 在【创建新元件】对话框中,输入名称“刘三姐画册”,类型选择“图形”,单击【确定】按钮,完成元件的创建。

(8) 输入“刘三姐画册”,文本属性设置如图8-12所示。

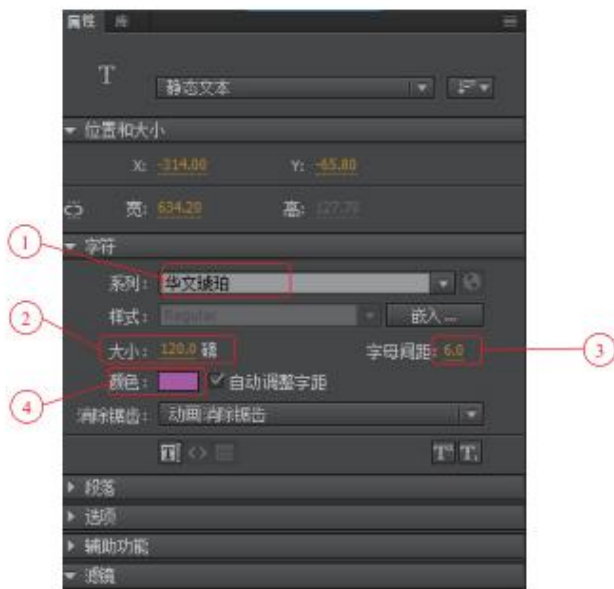
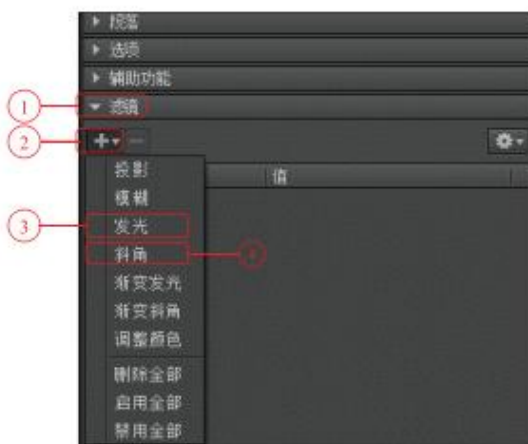


图8-12 文本属性设置

(9) 给文本添加“斜角”和“发光”滤镜,如图8-13所示。



刘三姐画册

图8-13 添加文本滤镜

三、动画制作

(1) 把“图层1”重命名为“背景”，从库中把“背景”拖到场景，在【属性】面板的【位置 and 大小】区域设置X为“0”，Y为“0”，宽为800像素，高为600像素。

(2) 在第100帧处插入帧，并锁定图层。

(3) 新建图层，命名为“文本”，从【库】面板中把元件“刘三姐画册”拖到场景，在第40帧处按【F6】键插入关键帧，并创建传统补间动画，在第1帧处把文字缩小。

(4) 新建图层，命名为“按钮”，在第40帧处按【F7】键插入空白关键帧，从【库】中把元件“播放”拖到场景右下角，如图8-14所示。



图8-14 添加“播放”按钮

(5) 新建图层，命名为“stop”，在第100帧处按【F7】键插入空白关键帧；选择【窗口】|【动作】命令或按【F9】键，打开【动作】面板，在【动作】面板中输入“stop();”，如图8-15所示。



图8-15 设置停止播放

(6) 在第40帧处单击【播放】按钮，在【动作】面板中单击【代码片段】按钮打开【代码片段】对话框，选择“ActionScript”，如图8-16所示。

(7) 展开【时间轴导航】选项，双击【单击以转到下一场景并播放】选项，如图8-17所示。

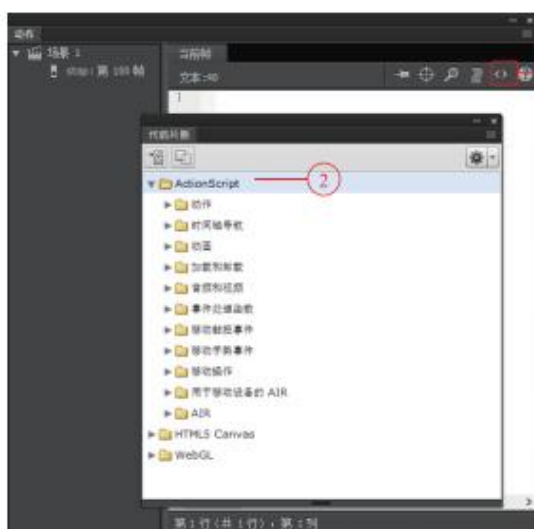


图8-16 【代码片段】对话框

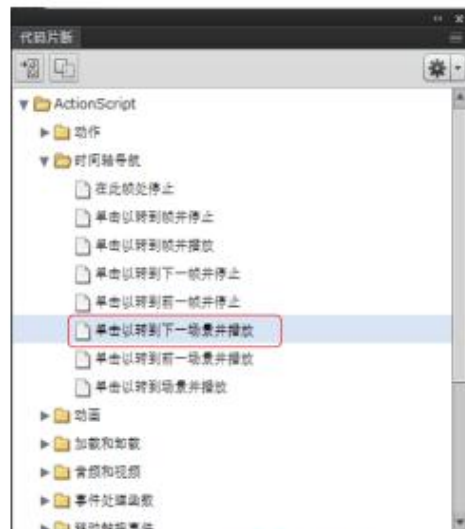


图8-17 设置播放代码

(8) 在弹出的对话框中单击“确定”按钮，如图8-18所示。



图8-18 确定实例名称

(9) 在【动作】面板中，自动生成播放代码，如图8-19所示。



图8-19 播放代码

(10) 在【时间轴】面板中,自动生成动作代码图层“Actions”,如图8-20所示。



图8-20 时间轴上自动生成动作代码图层

(11) 新建图层,命名为“音乐”,把音乐“只有山歌敬亲人.mp3”从【库】面板中拖到场景中,如图8-21所示。



图8-21 添加音乐

(12) 选择【窗口】|【场景】命令(快捷键【Shift】+【F2】)新建场景,如图8-22所示。

(13) 在【场景】面板中单击【添加场景】按钮,创建场景2,如图8-23所示。

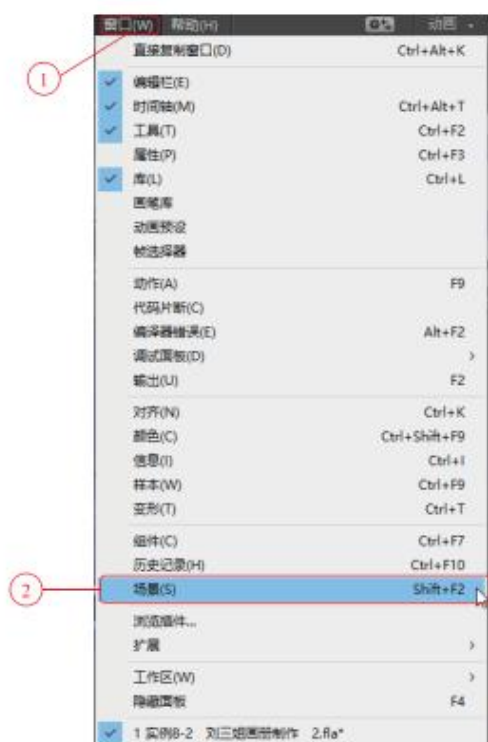


图8-22 新建场景



图8-23 新建场景

(14) 把“图层1”重命名为“背景”，从【库】面板中把“画框”拖到场景，在【属性】面板的【位置和大小】区域设置X为“0”，Y为“0”，宽为800像素，高为600像素，如图8-24所示。

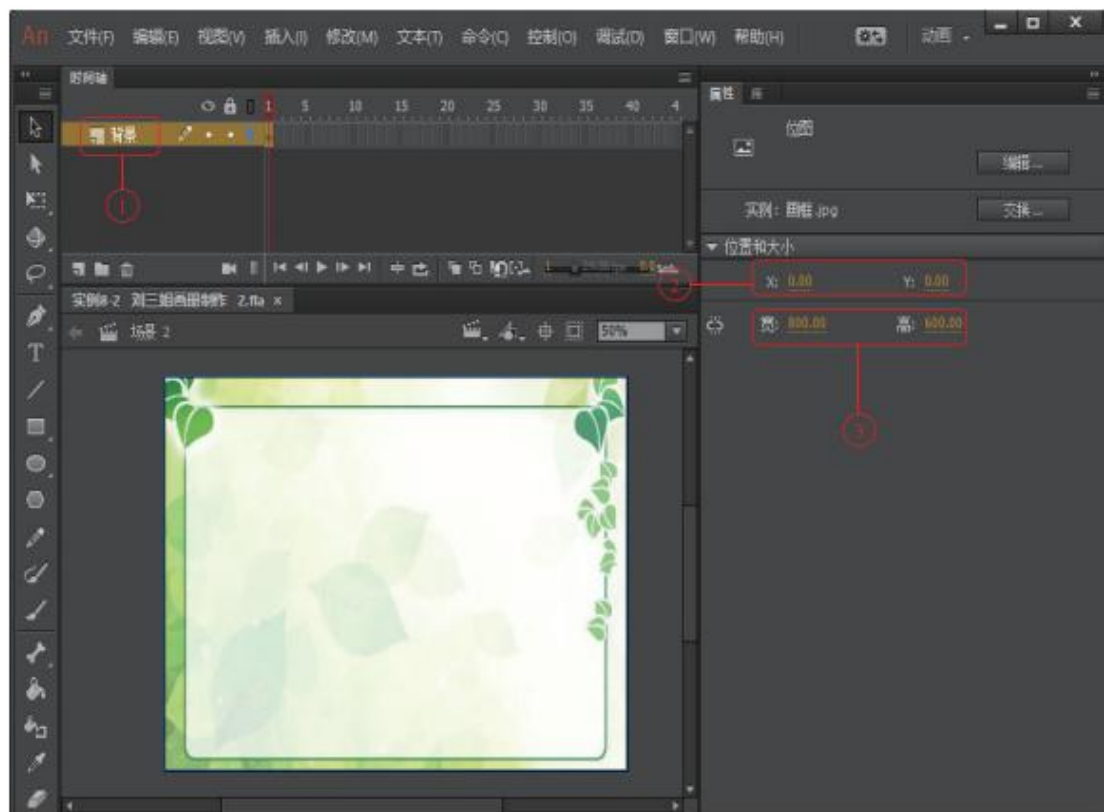


图8-24 设置场景2背景

(15) 在第350帧处插入帧，并锁定图层，如图8-25所示。

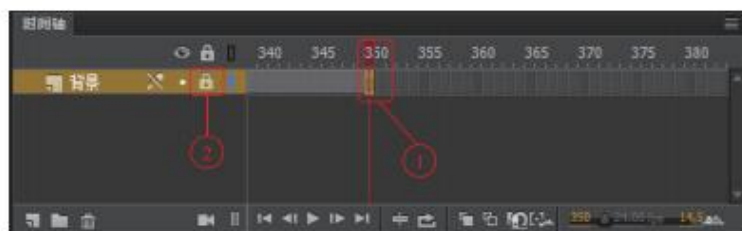


图8-25 插入帧并锁定图层

(16) 新建图层，命名为“画册1”，把图片“画册1”拖到舞台合适位置，并调整大小，如图8-26所示。按【F8】快捷键，将图片转换为元件，输入元件名为“画册1”，类型选择“图形”，单击【确定】按钮。

(17) 在第30帧处按【F6】键插入关键帧，并创建传统补间，在第1帧处把图片透明度设置为“1%”，如图8-27所示。



图8-26 摆放好画册1

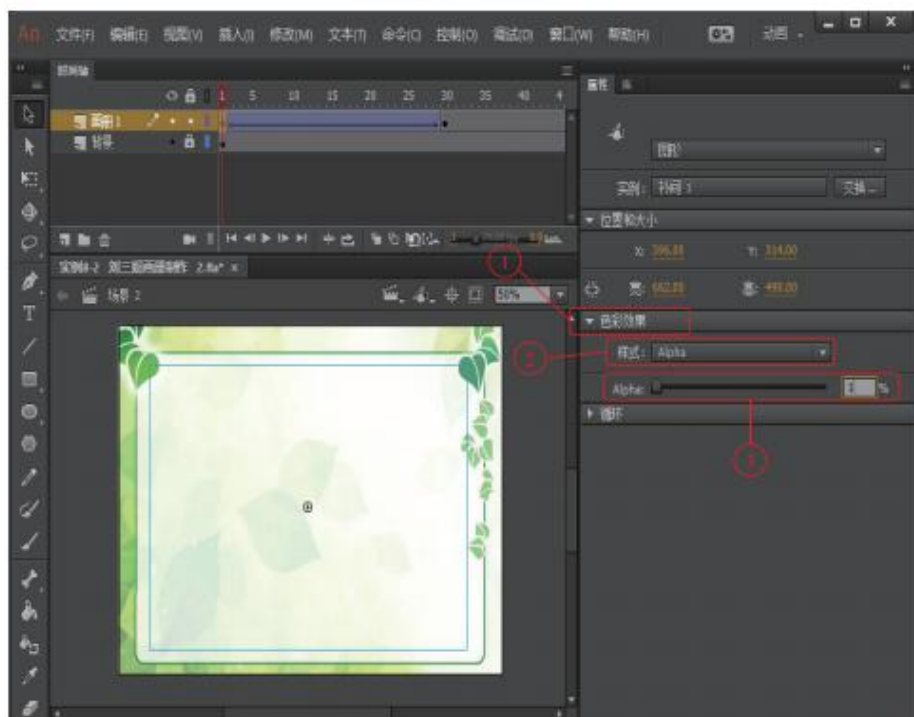
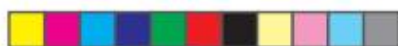


图8-27 把图片透明度设置为“1%”



(18) 新建图层, 命名为“画册2”, 在第60帧处按【F7】键插入空白关键帧, 把图片“画册2”拖到舞台合适位置, 并调整大小, 如图8-28所示。按【F8】键, 将图片转换为元件, 输入元件名为“画册2”, 类型选择“图形”, 单击【确定】按钮。



图8-28 摆放好画册2

(19) 在第90帧处按【F6】键插入关键帧, 并创建传统补间, 在第60帧处把图片缩小并设置透明度为“1%”, 如图8-29所示。

(20) 新建图层, 命名为“画册3”, 在第120帧处按【F7】键插入空白关键帧, 把图片“画册3”拖到舞台合适位置, 并调整大小, 如图8-30所示。按【F8】键, 将图片转换为元件, 输入元件名为“画册2”, 类型选择“图形”, 单击【确定】按钮。

(21) 在第150帧处按【F6】键插入关键帧, 并创建传统补间, 在第120帧处把图片缩小并设置透明度为“1%”。在【属性】面板把【补间】区域将【旋转】设置为“顺时针”, 如图8-31所示。

(22) 新建图层, 命名为“画册4”, 在第180帧处按【F7】键插入空白关键帧, 把图片“画册4”拖到舞台合适位置, 并调整大小, 如图8-32所示。按【F8】键, 将图片转换为元件, 输入元件名为“画册2”, 类型选择“图形”, 单击【确定】按钮。

(23) 在第210帧处按【F6】键插入关键帧, 并创建传统补间, 在第180帧处把图片缩小并设置透明度为“1%”, 如图8-33所示。

(24) 新建图层, 命名为“画册5”, 在第210帧处按【F7】键插入空白关键帧, 把图片“画册5”拖到舞台合适位置, 并调整大小, 如图8-34所示。按【F8】键, 将图片转换为元件, 输入元件名为“画册2”, 类型选择“图形”, 单击【确定】按钮。

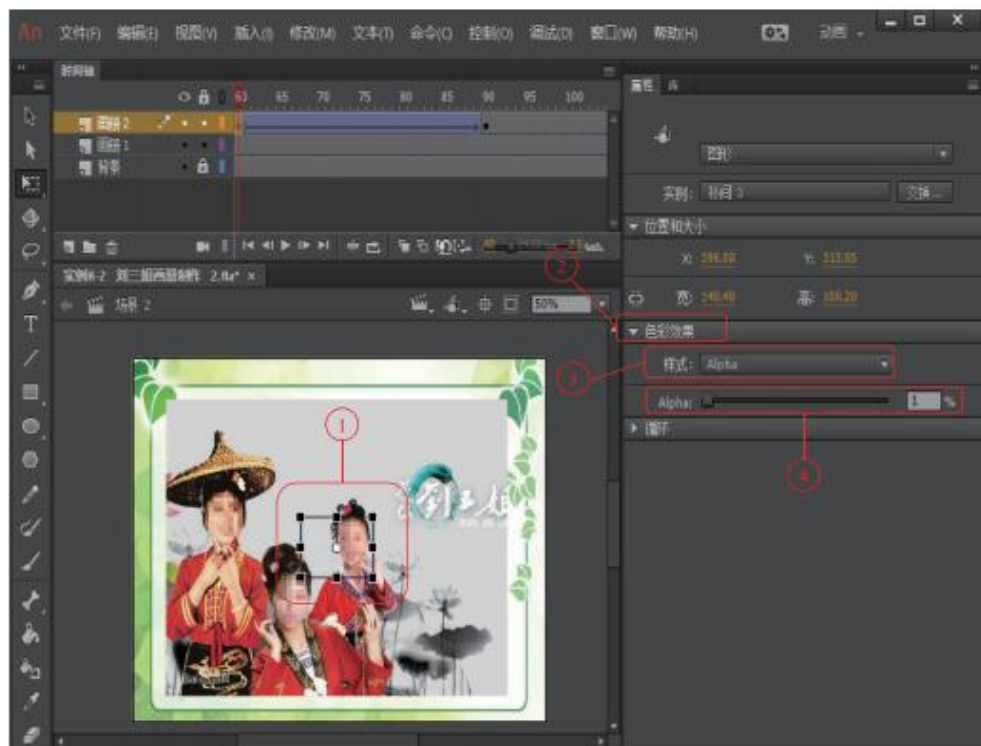


图8-29 把图片缩小并设置透明度为“1%”



图8-30 摆放好画册3

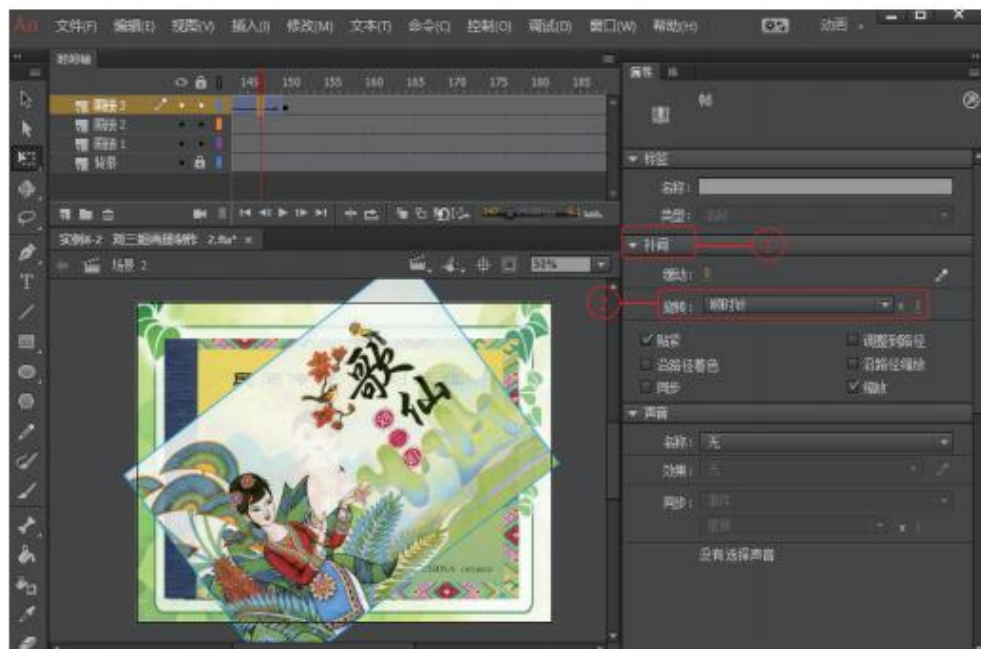


图8-31 把图片设置成旋转效果



图8-32 摆放好画册4

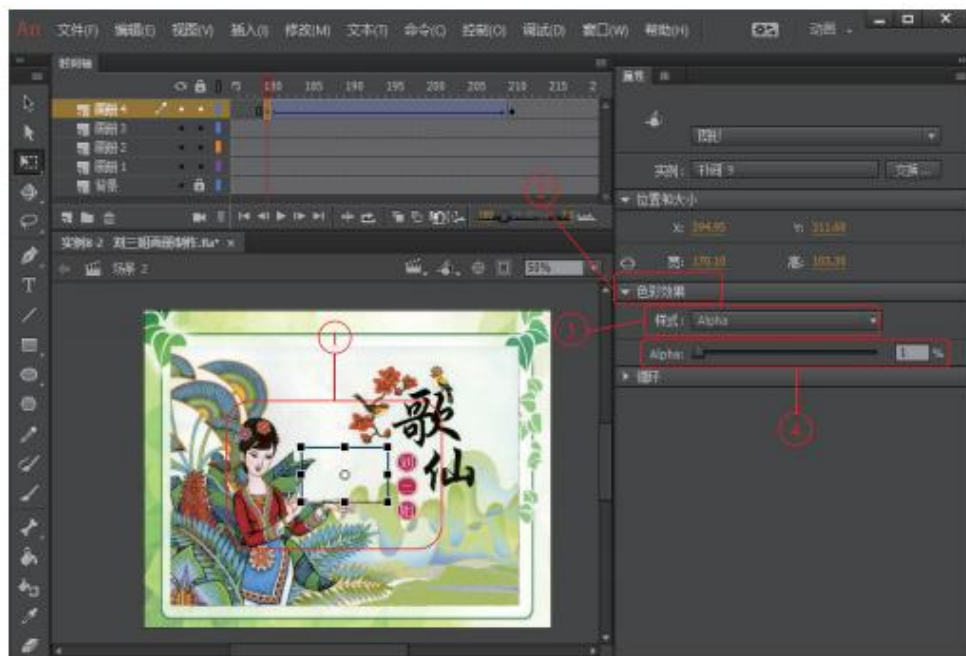


图8-33 把图片缩小并设置透明效果



图8-34 摆放好画册5



二维动画制作案例教程

(25) 在第270帧处按【F6】键插入关键帧，并创建传统补间，在第240帧处把图片放大并设置透明度为“1%”，如图8-35所示。

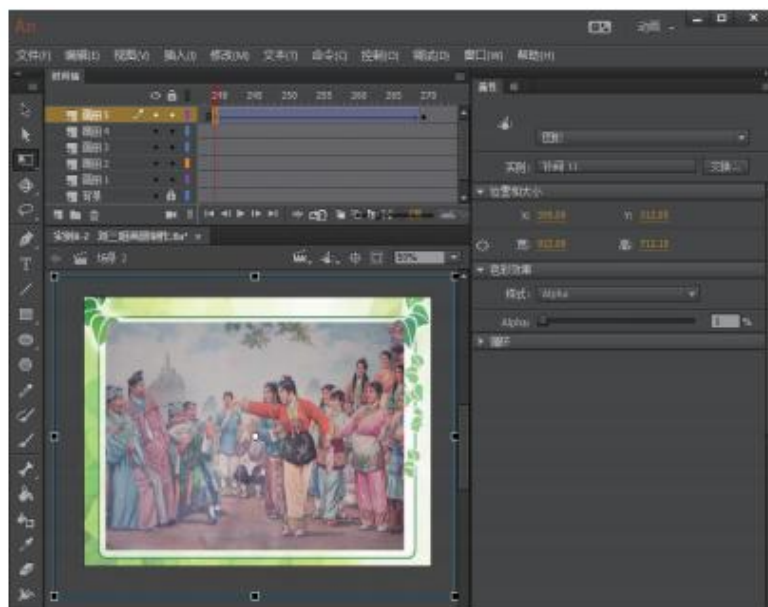


图8-35 把图片放大设置透明效果

(26) 新建图层，命名为“画册6”，在第300帧处按【F7】键插入空白关键帧，把图片“画册6”拖到舞台合适位置，并调整大小，如图8-36所示。按【F8】键，将图片转换为元件，输入元件名为“画册6”，类型选择“图形”，单击【确定】按钮。



图8-36 摆放好画册6

(27) 在第330帧处按【F6】键插入关键帧，并创建传统补间，在第300帧处把图片缩小并设置成旋转效果，如图8-37所示。



图8-37 把图片缩小并设置成旋转效果

(28) 新建图层，命名为“按钮”，在第330帧处按【F7】键插入空白关键帧，从【库】面板中把元件“重播”拖到场景右下角，如图8-38所示。



图8-38 添加重播按钮



(29) 新建图层, 命名为“stop”, 在第350帧处按【F7】键插入空白关键帧。选择【窗口】|【动作】命令或按【F9】键, 打开【动作】面板, 在【动作】面板中输入“stop();”。

(30) 在第330帧处单击【重播】按钮, 在【动作】面板中单击【代码片断】按钮, 打开【代码片断】面板。

(31) 在【代码片断】面板中, 展开【时间轴导航】选项, 双击【单击以转到下一场景并播放】选项。

(32) 弹出创建实例名称提醒, 单击【确定】按钮, 如图8-39所示。



图8-39 创建实例名称提醒

(33) 在【动作】面板中, 自动生成播放代码, 如图8-40所示。



图8-40 重播代码

(34) 在【时间轴】面板中, 自动生成动作代码图层“Actions”, 如图8-41所示。



图8-41 时间轴上自动生成动作代码图层

(35) 保存文件,按【Ctrl+Enter】组合键进行影片测试。

小 结

本章主要介绍了ActionScript 3.0的基础知识,【代码片段】面板的使用。在本章的学习中还应注意以下几点:

- (1) 要使ActionScript语句能够正常运行,必须按照正确的语法规则进行编写。
- (2) 利用【代码片段】面板可以使初学ActionScript 3.0语言的用户快速上手,还可以帮助用户了解不同语句的用途和使用方法。
- (3) 在【动作】面板中可以查看、添加和编辑ActionScript代码。
- (4) 利用ActionScript 3.0制作的交互效果,有很多都是利用鼠标事件触发的,因此应熟练掌握各种鼠标事件的代码。

练习与思考

一、填空题

1. 按照组件文件的发布格式,ActionScript 3.0中的组件可分为:_____和_____。
2. ComboBox 组件就是下拉列表框组件,应用于需要从列表中选择一项的表单或应用程序中,该组件由三个子组件组成_____组件、TextInput组件和_____组件。
3. Label组件就是标签组件,一个标签组件就是一行文本,该组件_____,不能具有焦点,并且_____。
4. Slider组件的当前值由滑块与端点之间的相对位置确定,其中_____和_____两个属性的值分别对应滑块与最左端和最右端的组件值。
5. 在任何需要单行文本字段的地方,都可以使用单行文本(TextInput)组件,该组件可以_____,或_____。

二、选择题(1-5单选)

1. RadioButton组件有一个独特的参数用于设置组名称()。
 - A. label
 - B. groupName
 - C. text
 - D. name
2. ComboBox 组件属性rowCount设置在不使用滚动条的情况下一次最多可以显示的项目数,默认值为()。
 - A. 1
 - B. 7
 - C. 3
 - D. 5
3. TextInput组件的()参数可以设置用户在文本字段输入的最大字符数。
 - A. editable
 - B. restrict
 - C. maxChars
 - D. text
4. TextArea组件的()参数指明文本是否自动换行,默认值为true。
 - A. text
 - B. wordWrap
 - C. html
 - D. editable