

macromedia

FLASH MX 2004

完全实战经典百例

第一时间体验最新版

FLASH MX 2004的强大功能

- ▶ Flash MX 2004版新功能、新特性及技巧使用全接触
- ▶ Flash MX 2004经典百例创作过程全面教学
(赠送100个FLASH MX 2004 经典百例配套源文件)
- ▶ 精彩绝伦的UI Components应用全方位讲解
- ▶ 涵盖全面的Action Script2语言应用完全实例解析
- ▶ 采用音频、视频、动画等模式教学，学习更轻松
- ▶ 附送：功能强大的周边软件、动画设计工具大全、
Flash动画素材库

使用手册+2CD=25元

让大家期待已久由 Macromedia 公司最新出品的 Flash MX 2004 终于面世了,这让众多的 Flash 爱好者欣喜不已,更让大家感到满意的是 Macromedia 这次在新版的 Flash 里所做出的努力。Flash MX 2004 在众多方面都做了改进,其中包括新增加了时间线效果,加速和简化时间线上可重复使用的通用交互任务。无需了解代码,新增加的行为可以通过行为面板快速应用 ActionScript。支持 PDF&EPS 直接导入。支持 CSS 样式表,允许 HTML 和 Flash 内容设计一致。为显示清晰易读的小字体新增加锯齿文字支持。新增加了许多设计模板,大大方便了大家的简易设计。新增了视频向导和外部 PLV 支持,这就大大提高了视频文件在 Flash 作品中运用的空间。在改良软件性能的同时,还合理修改了操作模式,新增了开始页面,提供了全新的嵌入式帮助,更新了拼写检查/全局查找替换功能,支持改进 MSAA (Microsoft Active Accessibility) 标准。新的支持面向对象编程 ActionScript 2 语言规范,在这次新版中还改进了历史面板、命令面板和 Unicode& 字符串面板,拓展了新的可扩展架构,为第三方软件支持提供了一个可靠的平台,最后增强了报表功能等众多大家期待改观的方面。

针对 Flash MX 2004 版的诸多新增功能与特性,我们组织 Flash 高手经过 3 个多月的日夜奋战,终于完成并推出了《Flash MX 2004 完全实战经典百例》这个产品。这 100 个实例的创作,涵盖了 Flash MX 2004 各个方面的知识和使用技巧,为广大用户体验这个新版本的强大功能和快速掌握使用这个新版本提供了捷径。

为使大家快速上手,我们把每个实例分为实例分析、效果预览、设计步骤三大部分讲述,实例教程由浅入深。与此同时,我们还对 Flash MX 2004 版的 UI Components 应用配以实例做了重点的讲解,通过学习,相信你对 UI Components 的运用会有全新的认识。

FLASH MX 2004 精彩实例 1

条形简易菜单

一、实例分析

这是一个简易的菜单实例，我本人觉得这是一个比较好的 Flash 入门实例，所以把这个实例摆在了第一个，希望大家能从这个实例中学到点东西。

二、效果预览



图 1-1 效果预览

三、设计步骤

1、打开 Flash MX 2004，按快捷键 Ctrl+J，然后把场景设置成 450PX × 120PX，背景为 #003366，12FPS。如图 1-2 所示：

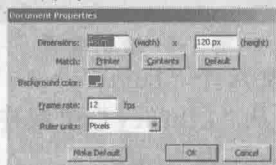


图 1-2 舞台大小设置

2、按 Ctrl+F8 新建一个名为“button_gfx”的 Graphic，进入编辑场景之后，我们绘制一个图 1-3 所示的图形。如图 1-3 所示：



图 1-3 绘制“button_gfx”Graphic

3、然后我们新建一个名为“button_movie”的 MC，进入编辑区之后，我们将刚才制作完成的“button_gfx”Graphic 拖拽到场景中，然后在第 15 帧按 F5，然后把默认的“Layer 1”层更名为“Ebene 1”。然后再新建一层名为“Ebene 2”的层，该层中也放置了一个“button_gfx”Graphic，而且和“Ebene 1”层中“button_gfx”Graphic 大小、位置都一样。

4、在“Ebene 2”的层的第 10 帧和第 15 帧添加关键帧，其中第 10 帧内的“button_gfx”Graphic 透明度为 0%，而且要比第 1 帧内的“button_gfx”Graphic 大点；第 15 帧内的“button_gfx”Graphic 透明度为 100%，要比第 1 帧内的“button_gfx”Graphic 小点。在对第 1 帧至第 10 帧以及第 10 帧至第 15 帧都执行“Create Motion Tween”命令之后的时间线如图 1-4 所示，为了获得一个更好的效果我们在“Ebene 2”的层第 1 帧放置了声音效果。



图 1-4 “button_movie”MC 的时间线

5、然后我们按 Ctrl+F8 新建一个名为“button_home”的 Button，在进入 But-

ton 编辑区之后我们在“Up”帧放“button_gfx”Graphic，在“Over”帧放“button_movie”MC（为了更美观你可以设置“Down”帧的效果），要求这些帧内的Graphic 或 MC 都必须重叠在一起。

6、我们新建一个名为“infotext”的 MC，该 MC 里只有一个 Var 为“text”的“Dynamic Text”文本框，文本框具体设置如图 1-5 所示：

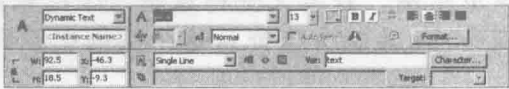


图 1-5 文本框设置

7、在做好这一切准备工作之后，我们按 Ctrl+E 回到主场景中，然后在主场景上增加两个图层，包括将默认的“Layer 1”层更名之后，从上到下依次为“Action”，“buttons”以及“background”层。我们在“background”层绘制一个背景，如图 1-6 所示，然后将该层锁定。



图 1-6 绘制背景

8、从“Library”面板中拖拽六个“button_home”Button 到“buttons”层中，并如图 1-7 所示一字排开，然后再从“Library”面板中拖拽出一个“infotext”，也放置在该层中，并设定 Instance Name 为“info”。



图 1-7 主场景上的各个 MC 排列

9、然后为“buttons”层中的“infotext”MC 以及六个“button_home”按钮分别添加如下所示的 ActionScript 代码：

“infotext”MC 添加的 ActionScript 代码：

```
onClipEvent (load) {
    anaus = 0;
}
onClipEvent (enterFrame) {
    _x = _x+_xmouse/5;
    if (anaus == 1) {
        if (_alpha<100) {
            _alpha = _alpha+10;
        }
        if (_alpha == 100) {
            _alpha = 90;
        }
    } else {
        if (_alpha>0) {
            _alpha = _alpha-10;
        }
    }
}
```

```

    }

```

左数第 1 个“button_home”按钮添加的 ActionScript 代码：

```

on (rollOver) {
    info.text = “Button 01”;
    info.anaus = 1;
}
on (rollOut, dragOut) {
    info.anaus = 0;
}

```

左数第 2 个“button_home”按钮添加的 ActionScript 代码：

```

on (rollOver) {
    info.text = “Button 02”;
    info.anaus = 1;
}
on (rollOut, dragOut) {
    info.anaus = 0;
}

```

左数第 3 个“button_home”按钮添加的 ActionScript 代码：

```

on (rollOver) {
    info.text = “Button 03”;
    info.anaus = 1;
}
on (rollOut, dragOut) {
    info.anaus = 0;
}

```

左数第 4 个“button_home”按钮添加的 ActionScript 代码：

```

on (rollOver) {
    info.text = “Button 04”;
    info.anaus = 1;
}
on (rollOut, dragOut) {
    info.anaus = 0;
}

```

左数第 5 个“button_home”按钮添加的 ActionScript 代码：

```

on (rollOver) {
    info.text = “Button 05”;
    info.anaus = 1;
}
on (rollOut, dragOut) {
    info.anaus = 0;
}

```

左数第 6 个“button_home”按钮添加的 ActionScript 代码：

```
on (rollOver) {  
    info.text = "Button 06";  
    info.anaus = 1;  
}  
on (rollOut, dragOut) {  
    info.anaus = 0;  
}
```

10、最后我们在主场景的“Action”层上的第 1 帧内添加 ActionScript “stop();”就完成了本例的制作,最后的预览效果如图 1-1 所示。

FLASH MX 2004 精彩实例 2

漂亮的 Flash 夜空繁星

一、实例分析

幽静的夜空,繁星点点,在空中一隐一显,多么美妙的一番景象啊,下面我们就使用功能强大的 Flash 来制作这一效果。

二、效果预览



图 2-1 效果预览

三、设计步骤

1、启动 Flash MX 2004,新建一个影片文件,设置影片舞台大小为 450PX × 150PX,设置影片背景色为黑色,12FPS。如图 2-2 所示:

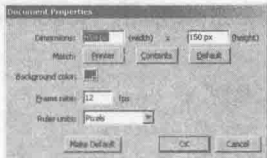


图 2-2 舞台大小设置

2、首先制作单个的星星。新建一个图形元件,命名为 Star,进入元件编辑区后,使用矩形工具在编辑区绘制一个无边框的白色矩形,并用鼠标拖动矩形左上角的顶点使之与右上角的顶点重合,形成一个长尖的三角形,如图 2-3 所示:



图 2-3 制作长三角形

3、选中工具箱中的颜料桶工具,然后再打开混色器面板,在填充类型下拉列表中选择线型渐变效果,然后设置左边的色块为黑色,右边的色块为白色。设置好后在三角形上点一下,给其填充上所设置的颜色,如图 2-4 所示:



图 2-4 给三角形填充渐变色

4、选中这个做好的三角形,复制并连续粘贴四个副本到编辑区中,使用旋转工具将其拼成一个星星图案,如下页图 2-5 所示。



图 2-5 拼接三角形为五角形

5、然后全选所有三角形,选择修改/组合菜单命令将其组合一个整体,并使用缩放工具将星星适当缩小,使其符合实际大小,如图 2-6 所示:



图 2-6 组合并缩小星星

6、接下来制作群星闪耀的效果。新建一个影片剪辑命名为“statMC”,进入元件编辑区后,从图库中将刚才做好的元件“star”拖到编辑区的中央,然后分别在时间轴的第 15 和第 30 帧处各插入一个关键帧。如图 2-7 所示:



图 2-7 设置关键帧

7、回到第 1 帧,选择该帧的星星,然后在属性面板中设置其透明度为 100%,最后选取第 15 帧,将该帧中星星的透明度设为 0%,最后分别在第 1 到第 15 帧,第 15 帧到第 30 帧之间设置运动渐变动画效果,从而实现了星星由有到无再到有的效果,时间轴如图 2-8 所示:



图 2-8 设置运动渐变动画

8、新建一个影片剪辑命名为“script”,进入元件的编辑区后,在图层 Layer1 中连续插入 3 个关键帧。

在第 1 帧中加入如下 Actions:

```
num=0;
```

在第 2 帧中加入 Actions 语句:

```
if (num>50){
    num=0;
}
```

```
_root.star.stop();
```

```
_root.star._x=random(550);
```

```
_root.star._y=random(400);
```

```
_root.star.duplicateMovieClip(“;star”;+num,num);
```

```
num=num+1;
```

在第 3 帧中加入 Actions 语句:

```
gotoAndplay(2);
```

9、最后回到主场景,把元件“starMC”和“script”分别拖到场景中,在属性面

板中设置元件“starMC”的实例名为“star”，这样一个星空的效果就完成了，最终效果请见图 2-1。

FLASH MX 2004 精彩实例 3

阴雨霏霏的场景

一、实例分析

春雨淅淅沥沥地下着，雨滴打到湖面上，泛起一个一个的圆圈。此情此景是借助复制、属性和随机函数应用的结果，具体实现步骤请查看——设计步骤。

二、效果预览

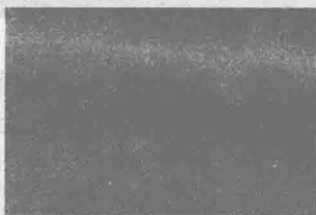


图 3-1 效果预览

三、设计步骤

1、启动 Flash MX 2004，新建一个影片文件，设置影片舞台大小为 300PX × 200PX，设置影片背景色为黑色，12FPS。如图 3-2 所示：

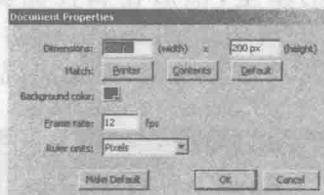


图 3-2 舞台大小设置

2、先创建一个小雨点 rain1 图形符号，再创建一个电影剪辑 rain2，将 rain1 拖入 rain2 窗口中，做成一个雨滴斜向下落的移动渐变动画如图 3-3 所示：

3、创建一个水波的图形符号 wave，它是一个用细线画成的椭圆，线的颜色选灰色较好，如图 3-4 所示：

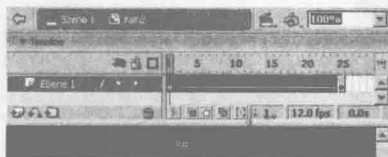


图 3-3 雨滴斜向下落的移动渐变动画



图 3-4 创建水波的图形符号

4、创建一个雨滴下落打到水面上泛起波纹的动画 rain_wave。这是一个两层的动画，第 1 至 25 帧为雨滴斜下落，第 20 至 25 帧为水波扩大，如图 3-5 所示：

5、进入主场景，在第 1 层画上背景，在第 2 层第 1 帧从图库中拖入电影剪辑 rain_wave，并命名为 drop，如下页图 3-6 所示。

6、在脚本层的第 1 帧加上脚本：

```
root.drop.visible=false;
```

在第 2 帧加上脚本：

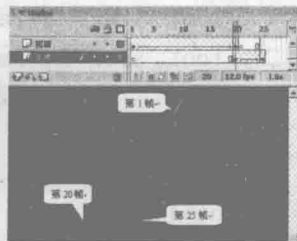


图 3-5 雨滴落到水面

```

if (i>60) {
i = 0;
}
duplicateMovieClip("drop", "drop" add i, i);
setProperty("drop" add i _x, random(300)+100);
setProperty("drop" add i _y, -(random(30)));
setProperty("drop" add i _alpha, random(100));
i++;

```

在第 3 帧加上脚本：

```
gotoAndplay(2);
```

提示：你可以试着改变第 2 帧脚本中的一些数字，看效果有何不同。

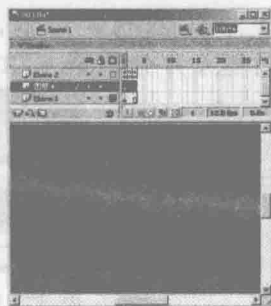


图 3-6 雨滴下落背景

FLASH MX 2004 精彩实例 4

旋转文字的制作

一、实例分析

旋转文字实例在实现上很简单，先是实现了单个字母的旋转动画，然后控制脚本语言实现整个动画效果，具体请查看——设计步骤。

二、效果预览



图 4-1 效果预览

三、设计步骤

1、启动 Flash MX 2004，新建一个影片文件，设置影片舞台大小为 200PX × 120PX，设置影片背景色为黑色，12FPS。如图 4-2 所示：

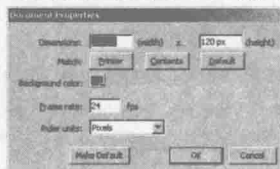


图 4-2 舞台大小设置

2、按下 Ctrl+F8 新建一个 MC，命名为 Text，在 layer1 中添加文本框，并在其中写入文字 M (Macromedia Flash MX)，并设置为动态文本框，变量名称设置为 char。



图 4-3 新建一个 Text MC

3、按下 Ctrl+F8 新建 MC，命名为 Play，将 Text 拖到 layer1，命名为 letter，制作一个文字绕椭圆运动轨迹，如下页图 4-4 所示。

4、回到主场景，将 Play 拖到 layer1 层的 1 到 8 帧，命名为 Play，同时新增一个层，并分别把 1、3、4、8 帧设为关键帧，并在 1、3、8 帧做如图 4-5 所示设定：

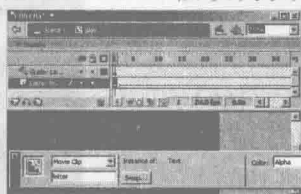


图 4-4 新建一个 Play MC

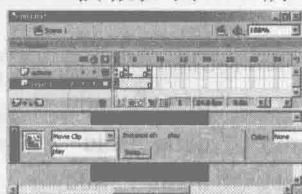


图 4-5 新建一个 Play MC

第一帧脚本：

```
fscommand("fullscreen", false);
fscommand("allowscale", false);
text = "Macromedia Flash MX";
xsize = 10;
ysize = 10;
max = length(text);
i = 1;
```

play._visible = 0;

第三帧脚本：

```
duplicateMovieClip ("play", "play" add i, i);
curchar=substring(text, i, 1);
set ("/play" add i add "/letter:char", curchar);
eval("play" add i)._xscale = xsize * 10;
eval("play" add i)._yscale = ysize * 10;
if(curchar eq " ") {
eval("play" add i)._visible=0;
}
```

i += 1;

第八帧脚本：

```
if (Number(i)>Number(max)) {
stop();
} else {
gotoAndPlay(3);
}
```

5、到此，该实例的全部工作都完成了，你可以按下 Ctrl+Enter 看看效果，最终效果请见图 4-1 所示。

FLASH MX 2004 精彩实例 5

动态添加图片的循环滚动效果

一、实例分析

网页上经常会用到无限循环滚动的图片这种效果，在 Flash 中来做，常规的方法是图片放在一个 MC 或 graphic，再把要展示的多幅图片依次相邻排好（假设它们

总长为 N, 然后按同样的顺序再放一组, 和前面的首尾相接), 然后用 AS 移动它们并且判断当它们移过的距离大于等于 N 时, 就将坐标向相反方向移动 N 个像素的距离, 从而实现无限循环滚动图片的效果。

但是这样做出来的源文件用过一次之后基本就没用了, 下次再用的时候得重新做, 而且一旦做好, 想改变一下图片的数目就不是那么方便, 有没有解决的办法呢? 本例就给出了一种解决办法, 我们按照上面的思路写如下 AS。

二、效果预览



图 5-1 效果预览

三、设计步骤

```
_global.i=6;
```

//6 是测试用的图片总数值, 完整的功能是用后台程序传递进来一个变量给 I。为提高速度, 图片按 1. jpg, 2. jpg 这样的规则依次命名, 图片大小为 100×40 像素

```
_global.leng = i*100;
```

//100 是图片的宽度, leng 是 i 张图片的总长度

```
_root.createEmptyMovieClip("snow", 0);
```

// 生成一个总的 MC snow

```
_root.snow.createEmptyMovieClip("a", 0);
```

```
_root.snow.createEmptyMovieClip("b", 1);
```

// 生成两个 snow 内部的 MC A 和 B

```
_root.snow.a._x = 0;
```

```
_root.snow.b._x = _root.snow.a._x+leng;
```

// 排列 MC A 和 B

```
for (j=1; j<=i; j++) {
```

```
    _root.snow.a.createEmptyMovieClip("mc"+j, j);
```

```
    _root.snow.b.createEmptyMovieClip("mc"+j, j);
```

```
    _root.snow.a["mc"+j].loadMovie("image/"+j+".jpg");
```

```
    _root.snow.b["mc"+j].loadMovie("image/"+j+".jpg");
```

// 在 MC A 和 B 内部产生数目为 i 的空 MC 并依次加载图片进去

```
    _root.snow.a["mc"+j]._x = (j-1)*100;
```

```
    _root.snow.b["mc"+j]._x = (j-1)*100;
```

// 设置 MC 的坐标, 让它们依次排开

```
} .
```

// 移动图像

```
_root.snow.onEnterFrame = function() {
```

```
    _root.snow._x -= 1;
```

// 向左移动总的 MC snow

```

if ((-_root.snow._x)>=leng) {
    _root.snow._x += leng;
    // 当 snow 向左移动的距离大于等于 leng 时, 将它向右移动 leng 个像素的长度
}
}

```

普通的无限循环图片动画中, 要想实现在某个图片上点击, 跳转到相应的地址非常简单, 但上面这种方法因为是通过 AS 生成的, 且使用了循环语句, 再想添加同样的效果似乎不那么容易。我们需要进行一点点修改, 放弃添加复杂的判断语句, 下面我们来看代码:

```

margin = _root._xmouse-_root.snow._x;
// 获得 MC snow 的 x 坐标
if (margin>=i*100) {
    margin-=leng;
}
id = Math.ceil(margin/100);
this.onMouseDown = function() {
    getURL("http://localhost/url.php?id="+id, "_blank");
};

```

上述代码是通过判断点击时鼠标和 MC snow 的 x 坐标的差来计算出鼠标现在停留在哪个图片上, 然后再赋予相应的网址。当然, 还可以修饰一下, 比如鼠标下面放一个按钮, 这样当鼠标移动到动画上方时指针会变形, 以便提示访客, 让他们知道这是可以点击的按钮。

提示: 其实这个例子最妙的用处还不是在这儿, 试着再进一步想, 假设我们把实现这个效果的图片放在一个单独的 image 目录下, 再用后台程序获取里面的图片数目传递给变量 i, 就可以任意改变 image 目录下的图片而不用再去制作新的动画了。

FLASH MX 2004 精彩实例 6

用 Transform 面板实现环形文字效果

一、实例分析

经常会有朋友问在 Flash 中怎么制作环形文字, 所以尝试了一些方法, 并总结了两种方法。下面用 Transform 面板实现环形文字效果为例, 来做一个说明。

二、效果预览

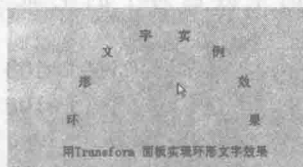


图 6-1 效果预览

三、设计步骤

1、启动 Flash MX 2004, 新建一个影片文件, 设置影片舞台大小为 450PX × 250PX, 设置影片背景色为 #CCCCCC, 12FPS。如下页图 6-2 所示。

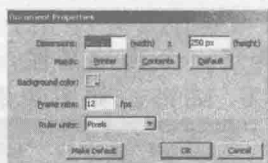


图 6-2 舞台大小设置

2、在主场景按 N，任意画一条线，并按 F8 转换为影片，把中心定为线条的右端，如图 6-3 所示。

3、返回主场景，选中直线，按 Ctrl+T 调出 Transform 面板，并在 Rotate 框中输入 36 (180/7)，角度的计算方法是 180 度除以边数。按面板中的 Copy and apply transform 按钮 5 次，就可复制出环形排列的七条直线。如图 6-4 所示：

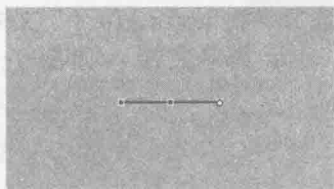


图 6-3 画一条直线

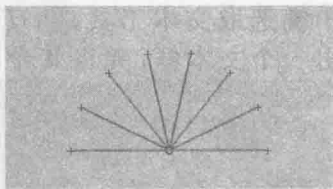


图 6-4 复制环形排列的七条直线

4、按 T 键，在场景中输入“环形文字实例效果”中的“环”字，并将其拖拽到第 1 条线的的最左端，这时 Flash 会自动把中心点放在线的左端，如图 6-5 所示。

5、最后如法炮制，制作出文字半圆排列的效果的其他几个文字，最后删除起辅助作用的线就可以了，如图 6-6 所示。

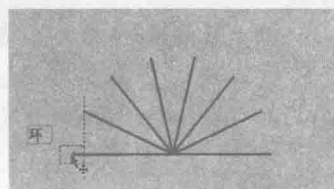


图 6-5 拖拽文字吸附到直线端

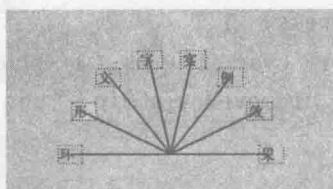


图 6-6 第一种方法效果图

终于搞定，Transform 面板真是好东东，用它制作了正多边形和文字的排列，它还能做什么呢？那要看你的想象力了。

FLASH MX 2004 精彩实例 7

用 AS 实现环形文字效果

一、实例分析

在实例六中我们已经讲述了用 Transform 面板实现环形文字效果的方法，那么在例七中我们将用 AS 来实现环形文字效果，具体操作请见——设计步骤。

二、效果预览

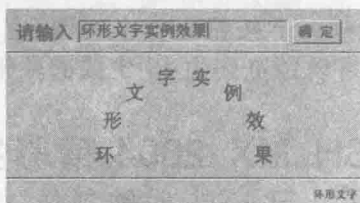


图 7-1 效果预览

三、设计步骤

1、启动 Flash MX 2004，新建一个影片文件，设置影片舞台大小为 450PX × 250PX，设置影片背景色为 #CCCCCC，12FPS。如图 7-2 所示：

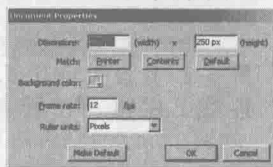


图 7-2 舞台大小设置

2、设置一个简单界面如图 7-3 所示，很简洁，有一个类型为 Input Text 的文本框和一个“确定”按钮。

3、按 T 键选取文本工具，并打开属性面板，选择“Dynamic Text”文本框，在场景中拖曳出一个文本框，并将其命名为 ZJS，以作显示文字之用。如图 7-4 所示：



图 7-3 初始化界面图

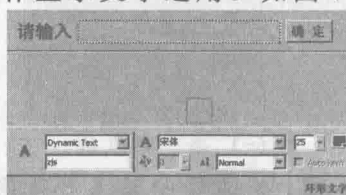


图 7-4 建立 ZJS 动态文本框

4、在场景中选择“确定”按钮，并为该按钮添加 AS，具体代码如下所示：

// 按下鼠标时移去 MC

```
on (press) {
    for (var i = 1; i<=shumu; i++) {
        removeMovieClip("zjs"+i);
    }
}
```

```
on (release) {
    shumu = zjs35.length; // 获取输入文本的长度
    jiaodu = 180/(shumu-1); // 计算角度
    for (var i = 1; i<=shumu;i++) {
        duplicateMovieClip("zjs", "zjs"+i, i); // 复制 MC
        _root["zjs"+i]._x = Math.sin((180-jiaodu*(i-1)+90)*math.pi/180)*100+210;
        _root["zjs"+i]._y = Math.cos((180-jiaodu*(i-1)+90)*math.pi/180)*100+170;
        // 利用圆的公式计算复制出的 MC 的坐标
        _root["zjs"+i].text = zjs35.substring(i-1, i); // 获取文字给动态文本
    }
}
```

5、现在你可以按下 Ctrl+Enter 按钮运行本实例，在文本录入框内写入要环形排列的文字，并按“确定”按钮就可以预览到由 AS 生成的环形文字效果了，效果见图 7-1。

FLASH MX 2004 精彩实例 8

FLASH 时钟制作

一、实例分析

这个例子为大家展示了 Flash 强大的 Action Script。这是个用 Flash 做的电子时钟,其中运用了 Flash 中提取系统时间的函数,并加以适当调整利用 Flash 的文字域显示。由于触及到的大部分是 Action,所以需要读者有一定的 Action 编程思想,对于初级 FLASH 用户来说,可能有点困难,可解析别人的作品是自己迅速成长的有效途径,下面我们就分析一下具体的制作过程。

二、效果预览



图 8-1 效果预览

三、设计步骤

1、启动 Flash MX 2004,新建一个影片文件,设置影片舞台大小为 450PX × 200PX,设置影片背景色为 #FFCC00,12FPS。如图 8-2 所示:

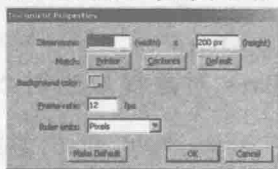


图 8-2 舞台大小设置

2、按快捷键 Ctrl+F8 新建一个 MC,命名为 Time。然后按快捷键 A 在 Time MC 添加三个文字域,并将文字域都设置为 Dynamic Text 类型。同样也是在 Text Options 里分别将三个文字域的名称设为:currenttime、currentdate、fulldate。如图 8-3 所示:

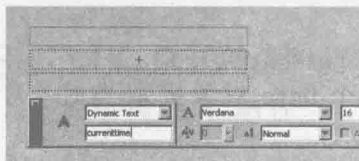


图 8-3 添加三个 Dynamic Text 文字域

3、现在我们对 Time 的操作就告一段落,回到主场景。从 Liabry 里拖出刚刚做好的 Time MC 放置在主场景的合适位置,并添加提示文字,最后如图 8-4 所示。



图 8-4 主场景最后界面

4、在做好这一切之后,我们应该对 Time MC 添加 Action 了,这里需要提示的是,我们所添加的 Action 不是针对 Time MC 所在的帧,而是添加在 Time MC 本身。

以下是程序清单,我将以注释方式解释:

```
onClipEvent (load) {
    days = new Array('周日', '周一', '周二', '周三', '周四', '周五',
        '周六', '周日');
    months = new Array('一月', '二月', '三月', '四月', '五月', '六月',
```

```

‘七月’，‘八月’，‘九月’，‘十月’，‘十一月’，‘十二月’);
//初始化数组,后面会用到 days、months 来帮助显示星期和月份

timedate = new Date();
// 声明一个 Date 对象用来提取系统时间
}

onClipEvent (enterFrame) {
    hour = (timedate.getHours());
    // 利用 timedate 这个 Date 对象提取当时“小时”

    minutes = timedate.getMinutes();
    // 利用 timedate 这个 Date 对象提取当时“分钟”

    seconds = timedate.getSeconds();
    // 利用 timedate 这个 Date 对象提取当时“秒钟”

    todaydate = timedate.getDate();
    // 利用 timedate 这个 Date 对象提取当时“日期”(也就是几号)

    day = timedate.getDay();
    // 利用 timedate 这个 Date 对象提取当时“星期数”(也就是星期几)

    dayname = days[day];
    // 利用 days 数组获得中文的星期数,放于 dayname 变量里

    month = (timedate.getMonth()+1);
    // 利用 timedate 这个 Date 对象提取当时“月数”(就是几月),加一是因为返回的值是从 0~11

    monthname = months[month-1];
    // 利用 months 数组获得中文的星期数,放于 monthname 变量里,在这里减一是因为中文的 months 顺序和系统返回的值相同

    year = timedate.getFullYear();
    // 利用 timedate 这个 Date 对象提取当时“年份”(比如 2003)

    if (Length(minutes)==1) {
        minutes = “0”+ minutes;
    }
    if (Length(seconds)==1) {
        seconds = “0”+ seconds;
    }
}

```

// 这两个 if 语句都是判断返回的分钟或秒钟数是不是一位数,如果是则在前面补零(为了美观)

```
currenttime = hour + ":" + minutes + ":" + seconds;
// 对 currenttime 赋值:小时:分钟:秒钟,同时这里就显示在我们定义好值为
currenttime 的那个文字域中了
```

```
currentdate = year + "/" + month + "/" + todaydate;
// 对 currentdate 赋值:年/月/日,同时这里就显示在我们定义好值为 cur-
rentdate 的那个文字域中了
```

```
fulldate = year + "年" + monthname + " " + todaydate + "日" + " " +
dayname;
```

```
// 对 fulldate 赋值:年,月,日,星期(中文),同时这里就显示在我们定义好
值为 fulldate 的那个文字域中了
```

```
delete timedate;
// 清除 timedate 这个 Date 对象
```

```
timedate = new Date();
// 再一次新建 timedate 这个 Date 对象,为了产生循环,可以不断侦测系统时间
```

```
}
```

好了,到此我们为 Time MC 添加的所有 Action 已经全部都讲完了,你会了吗? 现在看看是不是很简单呢? 按下 Ctrl+Enter 看看最终效果,见图 8-1。

FLASH MX 2004 精彩实例 9

星星文字制作

一、实例分析

星星文字的效果如果按照一帧一帧的来做也是可以成功的,但是那会是一件非常吃力的事情,如果使用 AS 来做的话,不仅可以节省劳动力,而且还可以让生成的 Flash 动画的体积变得很小。下面让我们来一步一步跟着做!

二、效果预览



图 9-1 效果预览

三、设计步骤

1、启动 Flash MX 2004,新建一个影片文件,设置影片舞台大小为 400PX × 200PX,设置影片背景色为黑色,12FPS。如下页图 9-2 所示。