万水计算机编程技术与应用系列

JavaApplet 实例编程

赤金　聂松　等编著

中国水利水电出版社
www.waterpub.com.cn

万水计算机编程技术与应用系列

JavaApplet 实例编程

赤金　聂松　等编著

中国水利水电出版社

内 容 提 要

Java 是一种新的编程语言，它的许多特点使其非常适合于 Internet 应用程序。Java 技术已被列为当今世界信息技术三大要点之一。

本书共分三大部分，七个章节。第一部分为 Applet 特效篇，分别介绍文字特效、图像特效、控件特效以及动画特效的实现方法及技巧；第二部分为 Applet 实用篇，介绍时间效果和其他效果的实现方法及技巧；第三部分为 Applet 游戏篇，介绍 Java 小游戏的实现方法及技巧。

本书通过几十个具体生动的实例讲解了使用 Java 进行编程的方法和技巧，是一本面向广大的 Java 爱好者和从事网络开发的院校学生及技术人员的参考书。

为方便读者学习，本书所有程序代码均可从中国水利水电出版社网站（www.waterpub.com.cn）下载。

图书在版编目（CIP）数据

JavaApplet 实例编程/赤金等编著. —北京：中国水利水电出版社，2002
（万水计算机编程技术与应用系列）
ISBN 7-5084-1320-2

Ⅰ. J… Ⅱ. 赤… Ⅲ. JAVA 语言—程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2002）第 101570 号

书　　名	JavaApplet 实例编程
作　　者	赤金　聂松　等编著
出版、发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@public3.bta.net.cn（万水） sale@waterpub.com.cn 电话：（010）68359286（万水）、63202266（总机）、68331835（发行部）
经　　售	全国各地新华书店
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787×1000 毫米　16 开本　14.25 印张　315 千字
版　　次	2003 年 1 月第一版　2003 年 1 月北京第一次印刷
印　　数	0001—5000 册
定　　价	26.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

版权所有·侵权必究

前　言

Java 是 Sun Microsystem 公司开发的一种新的编程语言。Java 一经推出就以其独有的开放性、跨平台性和面向网络的交互性席卷全球，并以其安全、易用和开发周期短等特点，迅速从最初的编程语言发展成为全球性软件开发平台。Java 的许多特点使得它非常适合于 Internet 应用程序，如 WWW，但 Java 并不仅限于 Internet。Java 技术已被列为当今世界信息技术三大要点之一。

根据权威部门的预测，2002 年我国对 Java 技术人员的需求量将高达 30 余万人。然而，到目前为止，实际人数和 30 万人的需求量相差甚远。因此，早日掌握 Java 技术，对每个有志在 IT 行业发展的人来说是尤为重要的。

近几年来，Java 在网络中的应用迅速发展和普及，利用 Java 进行研究开发的人员也越来越多。但是很多人仍然处在开始学习和摸索阶段，加之国内缺乏 Java 方面的中文资料，很需要这方面的书籍和教程。所以我们组织编写了这本书。

本书共分三大部分，七个章节。第一部分为 Applet 特效篇，第一章介绍文字特效的实现方法及技巧，第二章介绍图像特效的实现方法及技巧，第三章介绍控件特效的实现方法及技巧，第四章介绍动画特效的实现方法及技巧；第二部分为 Applet 实战篇，第五章介绍时间效果的实现方法及技巧，第六章介绍其他效果的实现方法及技巧；第三部分为 Applet 游戏篇，第七章介绍 Java 小游戏的实现方法及技巧。

在使用本书时，为了能感受正确的效果，希望读者对系统进行如下配置：

系统要求：Windows 98，IE 4.0/5.0。

系统设置：分辨率为 1024×768。

本书面向广大的 Java 爱好者和从事网络开发的院校学生及技术人员。

为了方便读者，所有程序实例的源代码均可在中国水利水电出版社网站 www.waterpub.com.cn 下载。

由于编写时间仓促，作者水平有限，书中难免有错漏之处，恳请读者批评指正。

作者

2002 年 11 月

目　　录

试读结束，需要全本PDF请购买 www.ertongbook.com

第一部分　JAVA 编程之特效篇

Java 的一项主要应用就是生成 Applet。简单地说，Applet 是使用 AppletViewer 或支持 Java 的 WWW 浏览器来运行的编译后的 Java 程序。Applet 可为网络提供丰富的功能，如在 WWW 页面显示动画、播放音乐、接受用户输入以及操作数据等，目前，它已成为 Internet 应用不可缺少的一部分。

当今互联网上各种生动活泼、漂亮迷人的精美网页、网站，确实令人赏心悦目，心驰神往，而评价一个网页的好坏，首先一点就是看它能不能以引人入胜的视觉效果和特效吸引住来访者。现在设计网页的工具有许多，但是真正具有美感和动感，能产生各种特效的网页大部分都是用 Java 语言来编写的。由于 Applet 程序比较小，可从网上很快下载并启动运行，这使得它们非常适合在 Internet 和 WWW 上使用。Applet 给 WWW 页面增加了新的功能和交互性，它的应用将使你的网页更加绚丽多彩。

本篇分类列举了二十多个典型的网页特效，具体地讲解了其实现方法和步骤，读者可以利用它们来为自己的个人网站增光添彩，同时可以掌握用 Java 设计网页的一些技巧。

第一章　文字特效

程序 1　波浪彩虹文字

这是一个对使用者的输入文字做处理的 Applet，使用者在 HTML 中输入字串，在 Applet 中就会以波动的轨迹及彩虹的色彩，依序展现。

HTML 中的内容如下：

```
<applet code="FancyText.class" width=400 height=100>
    <param name=word value="波动彩虹文字示例">
</applet>
```

HTML 所需设定的参数有一个，即欲显示的字串。

程序运行界面如图 1-1 所示。

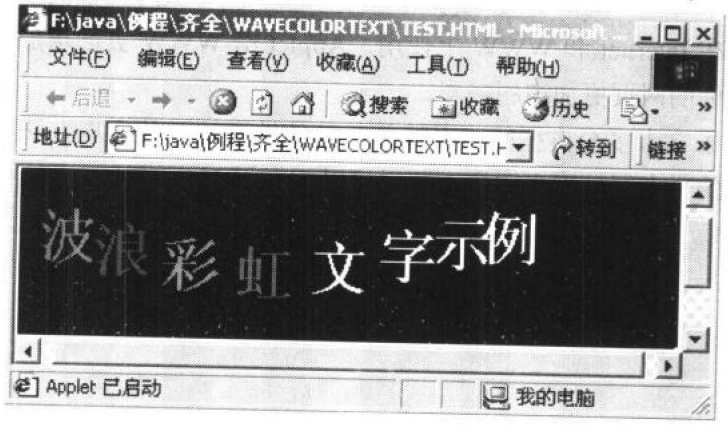

图 1-1　波浪彩虹文字

Java 程序分析如下：

```
//引入程序所需类库包
import java.awt.*;
import java.applet.*;

//定义基本类 FancyText，实现与线程的接口
public class FancyText extends Applet implements Runnable
{
String s = null;
    int direct = 1;
    int Hrad = 12;
    int Vrad = 12;
    Thread thread = null;
```

```
    char words[];
    int phase = 0;
    Image offI;
    Graphics offG;
    Color colors[];
    private Font f;
    private FontMetrics fm;

//定义init()方法
public void init()
{
    String param = null;
    //读取HTML文件中给定的欲显示字符
    s = getParameter("word");
    //设置背景色为黑色
    setBackground(Color.black);
    //将字符串的内容读入字符数组中
    words = new char [s.length()];
    s.getChars(0,s.length(),words,0);
    //创建一个离屏图像区域offI和图形环境offG
    offI = createImage(getSize().width,getSize().height);
    offG = offI.getGraphics();
    //设定字体及大小
    f = new Font("TimesRoman",Font.BOLD,36);
    fm=getFontMetrics(f);
    offG.setFont(f);
    //为每个字符设定不同的颜色
    float h;
    colors = new Color[s.length()];
    for (int i = 0; i < s.length(); i++)
    {
        h = ((float)i)/((float)s.length());
        colors[i] = new Color(Color.HSBtoRGB(h,1.0f,1.0f));
    }
}

//定义start()方法，启动线程
public void start()
{
    if(thread == null)
    {
        thread = new Thread(this);
        thread.start();
    }
```

```
}

//定义 stop()方法，终止线程
public void stop()
    {
    if (thread != null)
    {
        thread.stop();
        thread = null;
    }
}

//定义 run()方法，运行线程，每隔一定时间调用一次 repaint()方法，
//同时导致 update()方法的调用
public void run()
{
    while (thread != null)
    {
        try
        {
            Thread.sleep(200);
        }
        catch (InterruptedException e)
        {

        }
        repaint();
    }
}

//定义 update()方法
public void update(Graphics g)
{
    int x, y;
    double ang;
    //以黑色填充绘图区域，实现清屏功能
    offG.setColor(Color.black);
    offG.fillRect(0,0,getSize().width,getSize().height);
    phase+=direct;
    phase%=8;
    //在离屏环境中绘制出要在屏幕上显示的下一幅图像
    for(int i=0;i<s.length();i++)
    {
        //确定下一个字符的显示位置
```

```
            ang = ((phase-i*direct)%8)/4.0*Math.PI;
            x = 20+fm.getMaxAdvance()*i+(int) (Math.cos(ang)*Hrad);
            y = 60+ (int) (Math.sin(ang)*Vrad);
            //确定下一个字符的显示颜色
            offG.setColor(colors[(phase+i)%s.length()]);
            //画出一个字符
            offG.drawChars(words,i,1,x,y);
        }
          //将离屏图像发送到实际的显示屏幕上
         paint(g);
    }

    //定义paint()方法，向屏幕发送图像
    public void paint(Graphics g)
    {
      g.drawImage(offI,0,0,this);
    }
}
```

在该程序中，使用了双缓冲区方法，以减少令人生厌的闪烁效果，即先绘制出要在屏幕上显示的图像，然后再发送到屏幕上去。该过程如下：

（1）创建一个离屏图像和图形环境：

```
Image offI;
Graphics offG;
```

（2）在小应用程序的init()方法中，创建第一步中的类实例变量：

```
offI = createImage(getSize().width,getSize().height);
offG = offI.getGraphics();
```

（3）在离屏图形环境中绘制所有图形：

```
for(int i=0;i<s.length();i++)
{
    //确定下一个字符的显示位置
    ang = ((phase-i*direct)%8)/4.0*Math.PI;
    x = 20+fm.getMaxAdvance()*i+(int) (Math.cos(ang)*Hrad);
    y = 60+ (int) (Math.sin(ang)*Vrad);
    //确定下一个字符的显示颜色
    offG.setColor(colors[(phase+i)%s.length()]);
    //画出一个字符
    offG.drawChars(words,i,1,x,y);
}
```

（4）使用如下语句将离屏图像发送到实际的显示屏幕上：

```
g.drawImage(offI,0,0,this);
```

程序的源代码请参见“程序范例\第一章\程序 1”目录，可在www.waterpub.com.cn上下载。

试读结束，需要全本PDF请购买 www.ertongbook.com

程序 2 阴影走马灯

这是一个根据文字产生阴影而且文字会移动的走马灯，使用时可以指定文字颜色、大小、背景颜色以及阴影颜色。

以下是 HTML 的说明：

```
<applet code=ShadowText.class width=300 height=50>
    <param name="Text" value="恭贺新禧">
    <param Name="FontSize" value=40>
    <param Name="Back" value="0000ff">
    <param Name="Fore" value="ff0000">
    <param Name="Shadow" value="660066">
</applet>
```

上面 Text 代表走马灯的文字内容，FontSize 代表字形大小，Back 代表背景颜色，Fore 代表文字的颜色，Shadow 代表阴影的颜色。

文字显示效果如图 1-2 所示。

图 1-2 阴影走马灯

Java 程序分析如下：

```
//引入程序所需类库包
import java.awt.*;
import java.applet.*;

//定义基本类 ShadowText，实现与线程的 Runnable 接口
public class ShadowText extends Applet implements Runnable
{
    private Image img;
    private Image offI;
    private Graphics offG;
```

```
private Thread thread = null;
private MediaTracker imageTracker;
private int height,width;
private String text;
private int FontSize;
private Font font;
private int textcolor, backcolor, shadowcolor;

//定义 init()方法
public void init()
{
    width = this.size().width;
    height = this.size().height;
    //获取 HTML 中给出的各项参数
    text = new String("Hello");
    String s = new String(getParameter("Text"));
    if(s != null)   text = s;
    FontSize = 30;
    s = new String(getParameter("FontSize"));
    if(s != null)   FontSize = Integer.parseInt(s);
    s = getParameter("Fore");
    textcolor = (s==null) ? 0x000000 : Integer.parseInt(s, 16);
    s = getParameter("Back");
    backcolor = (s==null) ? 0x000000 : Integer.parseInt(s, 16);
    s = getParameter("shadow");
    shadowcolor = (s==null) ? 0x000000 : Integer.parseInt(s, 16);
    //设置绘图区域的背景色
    this.setBackground(new Color(backcolor));
    //创建一个离屏图像 img 和图形环境 temp
    img = createImage(width,height);
    Graphics temp = img.getGraphics();
    //以背景色填充离屏图像区域
    temp.setColor(new Color(backcolor));
    temp.fillRect(0,0,width,height);
    //画文字的阴影
    temp.setColor(new Color(shadowcolor));
    font = new Font("TimesRoman",Font.BOLD,FontSize);
    temp.setFont(font);
    temp.drawString(text,10,height*3/4);
    //画前景文字
    temp.setColor(new Color(textcolor));
    temp.drawString(text,10-3,height*3/4 - 3);
    //创建一个 MediaTracker 类的实体 imageTracker
```

```
        imageTracker = new MediaTracker(this);
        imageTracker.addImage(img,0);
        //等待图片加载完毕
        try
        {
        imageTracker.waitForID(0);
        }
            catch(InterruptedException e)
        {

        }
        //创建另一个离屏图像 offI 和图形环境 offG
        offI = createImage(width,height);
        offG = offI.getGraphics();
    }

    //定义 start()方法，启动线程
    public void start()
    {
        if(thread == null)
        {
            thread = new Thread(this);
            thread.start();
        }
    }

    //定义 run()方法，运行线程
    public void run()
    {
        int x=width;
        while(thread != null)
        {
            try
            {
                //将 img 中画好的图片调入离屏图形环境 offG 中
                offG.drawImage(img,x,0,this);
                //显示图片
                repaint();
                //延时
                thread.sleep(50);
            }
            catch(InterruptedException e)
            {
```

```
            }
            //改变图片显示位置
            x--3;
            if(x < -width)
            {
                x = width;
            }
        }
    }

    //定义update()方法
    public void update(Graphics g)
    {
        paint(g);
    }

    //定义paint()方法
    public void paint(Graphics g)
    {
        //在当前屏幕上显示图片
        g.drawImage(offI,0,0,this);
    }
}
```

本程序中用到了两个离屏图像，其中 img 中放的是画好的文字，offI 中放的是已变换位置的文字图片。使用这种方法，既可以节省内存，不用为每幅不同位置的图片分配地址，又可以减少花费在图形绘制上的时间，使得程序的运行看上去更加流畅自然。

程序的源代码请参见“程序范例\第一章\程序 2”目录，可在www.waterpub.com.cn上下载。

程序 3　3D 立体渐层文字

这是一个 3D 立体渐层的文字看板，字会由浅渐深慢慢浮出来，以达到立体的效果，文字的颜色随时间不停地更换，使用时不需用参数来控制。

HTML 中的内容如下：

```
<applet code=Text3D.class width=240 height=60>
    <param name="Text" value="3D立体渐层文字">
</applet>
```

在上面的 HTML 文件中，参数 Text 表示 Applet 显示出来的文字。

文字显示效果如图 1-3 所示。

图 1-3 3D 立体渐层文字

Java 程序分析如下：

```
//引入程序所需类库包
import java.awt.*;
import java.applet.*;

//定义基本类 Text3D，实现与线程的 Runnable 接口
public class Text3D extends Applet implements Runnable
{
    private Image img;
    private Image offI;
    private Graphics offG;
    private Thread thread = null;
    private MediaTracker imageTracker;
    private int height,width;
    private String text;
    private int FontSize;
    private Font font;

    //定义 init()方法
    public void init()
    {
        //获取小应用程序运行区域的宽和高
        width = this.size().width;
        height = this.size().height;
```

```
    //设置背景色为黑色
    this.setBackground(Color.black);
    //创建离屏图像 img 及其图形环境
    img = createImage(width,height);
    Graphics temp = img.getGraphics();
    //将 img "涂黑"，并设置画笔颜色为红色
    temp.setColor(Color.black);
    temp.fillRect(0,0,width,height);
    temp.setColor(Color.red);
    //缺省文字为 "Hello"
    text = new String("Hello");
    //获取 HTML 中的参数
    String s = new String(getParameter("Text"));
    if(s != null)
        text = s;
    //设置字符大小及字体
    FontSize = 30;
    font = new Font("TimesRoman",Font.BOLD,FontSize);
    temp.setFont(font);
    //在离屏图像 img 中绘制文字
    temp.drawString(text,10,height*3/4);
    //创建一个 MediaTracker 类，进行图像的存储与调入
    imageTracker = new MediaTracker(this);
    imageTracker.addImage(img,0);
    //等待图像调入完毕
    try
    {
        imageTracker.waitForID(0);
    }
    catch(InterruptedException e)
    {
    }
    //创建另一个离屏图像 offI 及其图形环境
    offI = createImage(width,height);
    offG = offI.getGraphics();
}

//定义 start()方法
public void start()
{
    if(thread == null)
    {
        thread = new Thread(this);
```

试读结束，需要全本PDF请购买 www.ertongbook.com