# 实

# 验

# 报

# 告

# 学院：电子信息与通信学院

# 班级：信卓2101班

# 姓名：吴永浩

# 学号：U202114208

# 实验时间：2024年4月25日

目录

[一、 实验目的 1](#_Toc164977220)

[二、 实验原理 1](#_Toc164977221)

[三、 编程环境介绍 1](#_Toc164977222)

[四、 程序结构及流程 1](#_Toc164977223)

[五、 实验结果 7](#_Toc164977224)

# 实验目的

1. 解码GIF文件并显示;
2. 有完整的实现代码;
3. 软件有图形界面。

# 实验原理

Gif播放器原理：

Gif使用LZW编码，LZW算法采用动态的建立字典的方法，依次读入原文件的字符序列，每次碰到新的连续的字符串。就在字典中增加标示。当下次再次遇到这样的字符串时，就能够用字典索引序号直接取代字符串，写入压缩文件里。

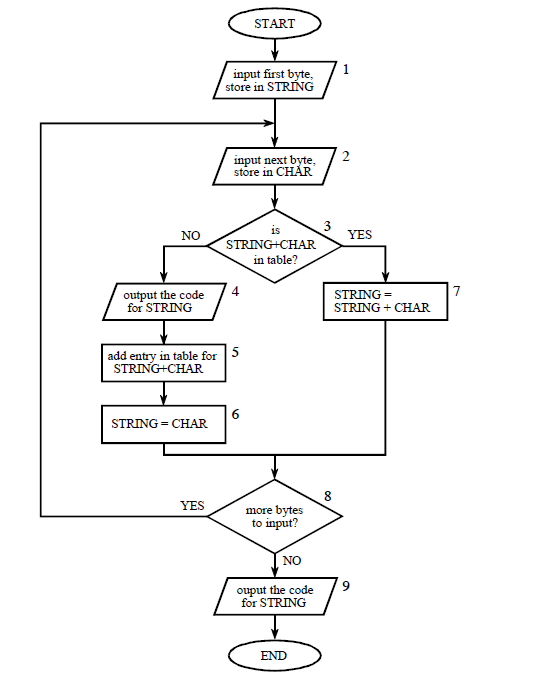
这里引入两个名词-- "string"，"char"：string表示前缀，char 表示新读入的字符，每一个字典索引相应一对(string,char）；

**编码**过程可概括为：

1."前缀string+字符char"在字典中不存在，增加字典。输出前缀，更新前缀=char,读入新字符char;

2."前缀string+字符char"在字典中存在，更新前缀=索引号，读入新字符char;

可用流程图表示为：



再引入两个名词--OCODE:表示已经读入的12位数据; NCODE:表示新读入的12位数据。

**解码**过程可概括为：

1. 开始时,初始化字典(0~255),读入第一个数据OCODE = A,输出字典索引OCODE相应的字符串A。读下一个数据NCODE = B 。

2. NCODE在字典中存在,输出table[NCODE]=B,CHAR=B;在字典索引256增加(OCODE,CHAR)=(A,B),更新OCODE=NCODE,读取数据NCODE=C；

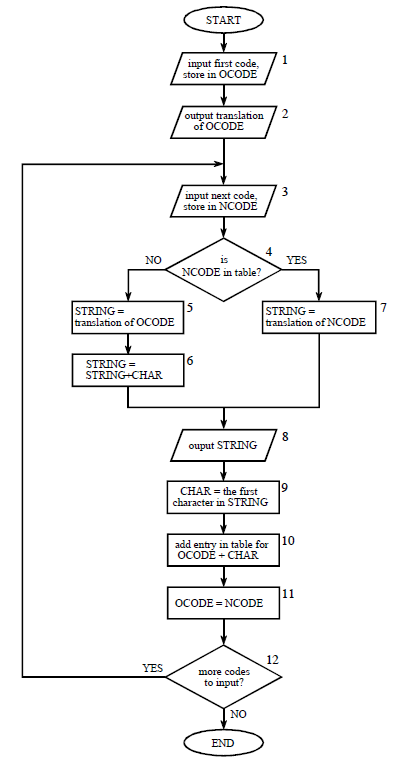
3. NCODE在字典中存在,输出table[NCODE]=C,CHAR=C;在字典索引257增加(OCODE,CHAR)=(B,C),更新OCODE=NCODE,读取数据NCODE=256。

4. NCODE在字典中已建立,table[256]=AB，并记录CHAR=A;在字典索引258增加(OCODE,CHAR)=(C,A)，更新OCODE=NCODE,读取数据NCODE=C；

5. NCODE在字典中已建立,table[NCODE]=C，CHAR=C。在字典索引259增加(OCODE,CHAR)=(256,C)，更新OCODE=NCODE读取数据NCODE=空。

6. 文件结束，跳出循环。

可用流程表示为：



本实验使用.net了自身携带的类ImageAnimator，这个类类似于控制动画的时间轴，使用ImageAnimator.CanAnimate可以判断一个图片是否为动画，调用ImageAnimator.Animate可以开始播放动画，即每经过一帧的时间触发一次OnFrameChanged委托。由于gif是由帧构成的，只需要在该委托中将Image的活动帧选至下一帧，再迫使界面重绘就可以实现播放效果了。

# 编程环境介绍

本实验使用visual studio IDE的Windows窗体编程环境。

WinForm 是 Windows Form 的简称，是基于 .NET Framework 平台的客户端（PC软件）开发技术，一般使用C#编程。C# WinForm 编程需要创建「Windows窗体应用程序」项目。

.NET 提供了大量 Windows 风格的控件和事件，我们可以直接拿来使用，上手简单，开发快速

Windows 窗体应用程序是 C# 语言中的一个重要应用，也是 C# 语言最常见的应用。

本实验主要使用c#语言进行编程。

# 程序结构及流程

实验程序主要由两部分组成：

classES.cs,包含一个AnimateImage类的定义，该类提供了CanAnimate、FrameCount、CurrentFrame等属性，以及Play()、Stop()、Reset()等播放gif必要的方法；

Form1.cs，包含了图形化界面、按钮及读取gif文件并通过AnimateImage类的调用来解码并显示它。

以下为代码：

AnimateImage**类定义代码**

namespace DIP\_03

{

/// <summary>

/// 表示一类带动画功能的图像。

/// </summary>

public class AnimateImage

{

Image image;

FrameDimension frameDimension;

/// <summary>

/// 动画当前帧发生改变时触发。

/// </summary>

public event EventHandler<EventArgs> OnFrameChanged;

/// <summary>

/// 实例化一个AnimateImage。

/// </summary>

/// <param name="img">动画图片。</param>

public AnimateImage(Image img)

{

image = img;

lock (image)

{

mCanAnimate = ImageAnimator.CanAnimate(image);

if (mCanAnimate)

{

Guid[] guid = image.FrameDimensionsList;

frameDimension = new FrameDimension(guid[0]);

mFrameCount = image.GetFrameCount(frameDimension);//

}

}

}

bool mCanAnimate;

int mFrameCount = 1, mCurrentFrame = 0;

/// <summary>

/// 图片。

/// </summary>

public Image Image//这个Image是System.Drawing.Image，将其重写

{

get { return image; }

}

/// <summary>

/// 是否动画。

/// </summary>

public bool CanAnimate

{

get { return mCanAnimate; }

}

/// <summary>

/// 总帧数。

/// </summary>

public int FrameCount

{

get { return mFrameCount; }

}

/// <summary>

/// 播放的当前帧。

/// </summary>

public int CurrentFrame

{

get { return mCurrentFrame; }

}

/// <summary>

/// 播放这个动画。

/// </summary>

public void Play()

{

if (mCanAnimate)

{

lock (image)

{

ImageAnimator.Animate(image, new EventHandler(FrameChanged));

}

}

}

/// <summary>

/// 停止播放。

/// </summary>

public void Stop()

{

if (mCanAnimate)

{

lock (image)

{

ImageAnimator.StopAnimate(image, new EventHandler(FrameChanged));

}

}

}

/// <summary>

/// 重置动画，使之停止在第0帧位置上。

/// </summary>

public void Reset()

{

if (mCanAnimate)

{

ImageAnimator.StopAnimate(image, new EventHandler(FrameChanged));

lock (image)

{

image.SelectActiveFrame(frameDimension, 0);//将当前帧设置为第0帧

mCurrentFrame = 0;

}

}

}

private void FrameChanged(object sender, EventArgs e)

{

mCurrentFrame = mCurrentFrame + 1 >= mFrameCount ? 0 : mCurrentFrame + 1;//如果当前帧+1后超出总帧数，则设置为0

lock (image)

{

image.SelectActiveFrame(frameDimension, mCurrentFrame);

}

if (OnFrameChanged != null)

{

OnFrameChanged(image, e);

}

}

}

**主窗体及调用、显示代码**

namespace DIP\_03

{

public partial class Form1 : Form

{

private string curFileName;

//图像对象

AnimateImage image;

public Form1()

{

InitializeComponent();

//image = new AnimateImage(Image.FromFile(@"F:\UnivercityLessons\DIP\DIP\_03\_testPIC.gif"));//--

//image = new AnimateImage(Image.FromFile(curFileName));//--

//image.OnFrameChanged += new EventHandler<EventArgs>(image\_OnFrameChanged);//--

//SetStyle(ControlStyles.OptimizedDoubleBuffer | ControlStyles.AllPaintingInWmPaint | ControlStyles.UserPaint, true);//--

}

private void Form1\_Load(object sender, EventArgs e)

{

//image.Play();//--

}

private void Form1\_Paint(object sender, PaintEventArgs e)

{

if(image == null)

return;

lock (image.Image)

{

e.Graphics.DrawImage(image.Image, new Point(0, 0));

}

}

void image\_OnFrameChanged(object sender, EventArgs e)

{

Invalidate();

}

private void open\_gif\_Click(object sender, EventArgs e)

{

if (open\_gif.Text.Equals("Stop"))

{

image.Stop();

open\_gif.Text = "Play";

}

else

{

image.Play();

open\_gif.Text = "Stop";

}

Invalidate();

}

private void reset\_Click(object sender, EventArgs e)

{

image.Reset();

open\_gif.Text = "Play";

Invalidate();

}

private void open\_choose\_Click(object sender, EventArgs e)

{

//创建OpenFileDialog

OpenFileDialog opnDlg = new OpenFileDialog();

//为图像选择一个筛选器

opnDlg.Filter = "所有图像文件|\*.bmp;\*.pcx;\*.png;\*.jpg;\*.gif;" +

"\*.tif;\*.ico;\*.dxf;\*.cgm;\*.cdr;\*.wmf;\*.eps;\*.emf|" +

"位图(\*.bmp;\*.jpg;\*.png;...)|\*.bmp;\*.pcx;\*.png;\*.jpg;\*.gif;\*.tif;\*.ico|" +

"矢量图(\*.wmf;\*.eps;\*.emf;...)|\*.dxf;\*.cgm;\*.cdr;\*.wmf;\*.eps;\*.emf";

//设置对话框标题

opnDlg.Title = "打开图像文件";

//启用“帮助”按钮

opnDlg.ShowHelp = true;

//如果结果为“打开”，选定文件

if (opnDlg.ShowDialog() == DialogResult.OK)

{

//读取当前选中的文件名

curFileName = opnDlg.FileName;

//使用Image.FromFile创建图像对象

try

{

image = new AnimateImage(Image.FromFile(curFileName));

image.OnFrameChanged += new EventHandler<EventArgs>(image\_OnFrameChanged);

SetStyle(ControlStyles.OptimizedDoubleBuffer | ControlStyles.AllPaintingInWmPaint | ControlStyles.UserPaint, true);

}

catch (Exception exp)

{

MessageBox.Show(exp.Message);

}

}

image.Play();

//对窗体进行重新绘制，这将强制执行paint事件处理程序

Invalidate();

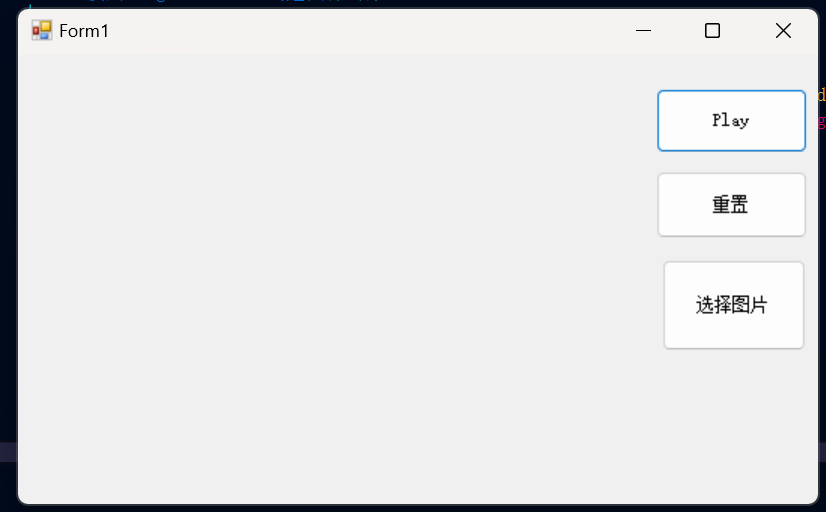
}

}

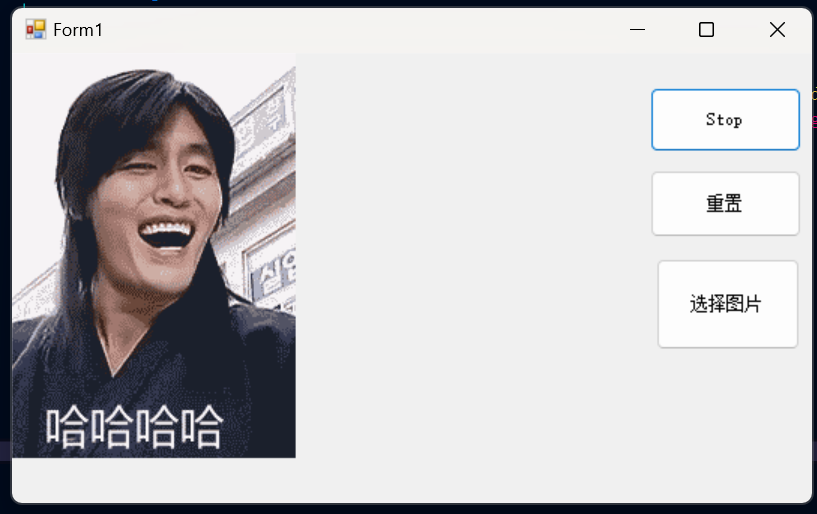
}

# 实验结果

在最后的窗体程序中，可以选择电脑中的gif文件打开，还可以通过两个按钮，分别控制gif图片播放/暂停和重置播放状态。



*图形界面*



*Gif图片播放状态*