

Course IPCis: Image Processing with C#

Chapter C1: The Complete Code of the Bitmap Project

Copyright © by V. Miszalok, last update: 12-01-2008

Copy all this code into an empty Form1.cs of a new Windows Application C#-project bitmap1 and clear Form1.Designer.cs and Program.cs.

```
using System;
using System.Drawing;
using System.Drawing.Imaging;
using System.Windows.Forms;

public class Form1 : Form
{
    static void Main() { Application.Run( new Form1() ); }
    Brush bbrush = SystemBrushes.ControlText;
    Brush rbrush = new SolidBrush( Color.Red );
    Bitmap bmp;
    int nClicks;

    public Form1()
    {
        MenuItem miRead = new MenuItem( "&Read", new EventHandler( MenuFileRead ) );
        MenuItem miExit = new MenuItem( "&Exit", new EventHandler( MenuFileExit ) );
        MenuItem miFile = new MenuItem( "&File", new MenuItem[] { miRead, miExit } );
        Menu = new System.Windows.Forms.MainMenu( new MenuItem[] { miFile } );
        Text = "Bitmap1";
        SetStyle( ControlStyles.ResizeRedraw, true );
        Width = 1024;
        Height = 800;
        try { //Delete this and the following 6 lines if you have no Internet connection running.
            System.Net.WebRequest webreq = System.Net.WebRequest.Create(
                "http://www.miszalok.de/Images/Madonna.bmp" );
            System.Net.WebResponse webres = webreq.GetResponse();
            System.IO.Stream stream = webres.GetResponseStream();
            bmp = (Bitmap)Image.FromStream( stream );
            Invalidate();
        } catch {};
    }

    void MenuFileRead( object obj, EventArgs ea )
    {
        OpenFileDialog dlg = new OpenFileDialog();
        if ( dlg.ShowDialog() != DialogResult.OK ) return;
        try { bmp = (Bitmap)Image.FromFile( dlg.FileName ); } catch { return, }
        nClicks = 0;
        Invalidate();
    }

    void MenuFileExit( object obj, EventArgs ea )
    {
        Application.Exit(); }

    protected override void OnMouseDown( MouseEventArgs e )
    {
        nClicks++;
        Invalidate();
    }
}
```

```

protected override void OnPaint( PaintEventArgs e )
{ Graphics g = e. Graphics;
  if ( bmp == null ) { g.DrawString( "Open an Image File !", Font, bbrush, 0, 0 ); return; }
  Rectangle cr = ClientRectangle;
  int line = 0;
  switch ( nClicks % 9 )
  { case 0: //Information
    g.DrawString("RawFormat = " + bmp.RawFormat.ToString(), Font, bbrush, 0, line+=Font.Height);
    if ( bmp.RawFormat.Guid == ImageFormat.Bmp.Guid )
      g.DrawString( "BMP", Font, bbrush, 0, line+=Font.Height );
    if ( bmp.RawFormat.Guid == ImageFormat.Jpeg.Guid )
      g.DrawString( "JPG", Font, bbrush, 0, line+=Font.Height );
    g.DrawString("Width      = " + bmp.Width.ToString()      ,Font, bbrush, 0, line+=Font.Height);
    g.DrawString("Height     = " + bmp.Height.ToString()     ,Font, bbrush, 0, line+=Font.Height);
    g.DrawString("PixelFormat = " + bmp.PixelFormat.ToString(),Font, bbrush, 0, line+=Font.Height);
    g.DrawString("Click on left mouse button !", Font, rbrush, 0, line+=Font.Height);
    break;
  case 1: //Raw display
    g.DrawImage( bmp, 0, Font.Height );
    g.DrawString( "Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
  case 2: //Center
    Int32 x = (cr.Width - bmp.Width ) / 2;
    Int32 y = (cr.Height - bmp.Height) / 2;
    g.DrawImage(bmp, x, y, bmp.Width, bmp.Height);
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
  case 3: //Horizontal stretch
    x = 0;
    y = ( cr.Height - bmp.Height / 2 ) / 2;
    g.DrawImage( bmp, x, y, cr.Width, bmp.Height / 2 ); //full form width, half bmp height
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
  case 4: //Vertical stretch
    x = ( cr.Width - bmp.Width / 2 ) / 2;
    y = 0;
    g.DrawImage( bmp, x, y, bmp.Width / 2, cr.Height ); //half bmp width, full form height
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
  case 5: //Full size
    g.DrawImage( bmp, cr );
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
  case 6: //Mirror
    g.DrawImage( bmp, cr.Width/2, cr.Height/2, cr.Width/2, cr.Height/2 );
    g.DrawImage( bmp, cr.Width/2, cr.Height/2, -cr.Width/2, cr.Height/2 );
    g.DrawImage( bmp, cr.Width/2, cr.Height/2, cr.Width/2, -cr.Height/2 );
    g.DrawImage( bmp, cr.Width/2, cr.Height/2, -cr.Width/2, -cr.Height/2 );
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
  case 7: //Zoom animation
    x = cr.Width / 20;
    y = cr.Height / 20;
    for ( Int32 i = 0; i < 20; i++ )
      g.DrawImage( bmp, 0, 0, cr.Width - x*i, cr.Height - y*i );
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
  case 8: //Rotation animation
    Single fx = cr.Width / 100;
    Single fy = cr.Height / 100;
    PointF[] p = new PointF[3];
    p[1].X = cr.Width;
    p[2].Y = cr.Height;
    do
    { p[0].X += fx;
      p[1].Y += fy;
      p[2].Y -= fy;
      g.DrawImage( bmp, p );
    } while ( p[2].Y > 0 );
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
  }
}
}
}

```