

Course IPCis: Image Processing with C#

Chapter C1: The Bitmap Project

Copyright © by V. Miszalok, last update: 08-01-2008

- ↓ [Projekt bitmap1 mit leerem Fenster](#)
- ↓ [Bild lesen und anzeigen](#)
- ↓ [Zentrieren](#)
- ↓ [Horizontal strecken](#)
- ↓ [Vertikal strecken](#)
- ↓ [Maximale Größe](#)
- ↓ [Spiegeln](#)
- ↓ [Zoom-Animation](#)
- ↓ [Drehanimation](#)
- ↓ [Beispielbilder](#)
- ↓ [Weitere Aufgaben](#)

Projekt bitmap1 mit leerem Fenster

Anleitung für **Visual Studio 2008**:

1) Main Menu nach dem Start von VS 2008: `File -> New Project... -> Visual Studio installed templates: Windows Forms Application`
 Name: `bitmap1` -> Location: `C:\temp` -> Create directory for solution: ausschalten -> OK
 Es meldet sich `Form1.cs[Design]`.

2) Sie müssen zwei überflüssige Files löschen: `Form1.Designer.cs` und `Program.cs`.
 Sie erreichen diese Files über das `Solution Explorer` - `bitmap1`-Window: Klicken Sie das Pluszeichen vor `bitmap1` und dann das Pluszeichen vor `Form1.cs`.
 Klicken Sie mit der **rechten** Maustaste auf den Ast `Program.cs`. Es öffnet sich ein Kontextmenu. Sie Klicken auf `Delete`. Eine Message Box erscheint: `'Program.cs' will be deleted permanently`. Sie quittieren mit OK.
 Klicken Sie mit der **rechten** Maustaste auf den Ast `Form1.Designer.cs` und löschen auch dieses File.

3) Klicken Sie mit der **rechten** Maustaste auf das graue Fenster `Form1`. Es öffnet sich ein kleines Kontextmenü. Klicken Sie auf `View Code`.
 Sie sehen jetzt den vorprogrammierten Code von `Form1.cs`. Löschen Sie den gesamten Code vollständig.

4) Schreiben Sie in das vollständig leere File `Form1.cs` folgende 3 Zeilen:

```
public class Form1 : System.Windows.Forms.Form
{ static void Main() { System.Windows.Forms.Application.Run( new Form1() ); }
}
```

5) Klicken Sie `Debug` im Main Menu oberhalb des Hauptfensters.

Es öffnet sich ein Untermenü. Klicken Sie auf `Start Without Debugging Ctrl F5`.

Wichtig: Immer zuerst alle Instanzen von `bitmap1` beenden, bevor neuer Code eingegeben und übersetzt wird !

Im Zweifel Task Manager mit `Ctrl+Alt+Del` starten und kontrollieren, ob noch ein `bitmap1.exe`-Prozess läuft und diesen töten.

Wenn Sie nicht Visual C# Express sondern Visual Studio 2008 Professional benutzen sollten Sie die Nerven tötende Formatier- und Einrückautomatik des Code-Editors ausschalten, bevor Sie den unten vorgegebenen Code durch kopieren nach `Form1.cs` transferieren:

1. Hauptmenu von Visual Studio 2005 Professional: Klick auf Menüpunkt "Tools".
2. Es erscheint ein DropDown-Menu. Klick auf "Options...".
3. Es erscheint eine Options Dialog Box.
4. Klick auf den Ast "Projects and Solutions". Klick auf "General". Alle drei Pfade auf `C:\temp` stellen.
5. Klick auf den Ast "Text Editor", dann auf "C#".
6. Es erscheint ein Unterbaum mit den Ästen "General, Tabs, Advanced, Formatting, IntelliSense".
7. Klick auf "Tabs". Stellen Sie "Indenting" auf None, "Tab size" und "Indent size" auf 1 und schalten Sie die Option "Insert spaces" ein.
8. Klicken sie im Unterbaum "C#" auf das Pluszeichen vor "Formatting" und ändern Sie alle "Formatting"-Äste: "General": alle CheckBoxes ausschalten, "Indentation": alle CheckBoxes ausschalten, "New Lines": alle CheckBoxes ausschalten, "Spacing": alle CheckBoxes ausschalten, "Wrapping": beide CheckBoxes einschalten.
9. Verlassen Sie den Dialog mit Button "OK".

Bild lesen und anzeigen

Falls Sie den Code nicht mehr sehen, klicken Sie in Visual Studio auf den Karteireiter **Form1.cs**, das bringt den Code zum Vorschein.

Sie löschen alles und schreiben in das leere Codefenster **Form1.cs** folgenden Code:

```
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
                g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
                break;
            case 3: //Horizontal stretch
                g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
                break;
        }
    }
}
```

```

case 4: //Vertical stretch
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
case 5: //Full size
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
case 6: //Mirror
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
case 7: //Zoom animation
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
case 8: //Rotation animation
    g.DrawString( "Change window size ! Click on left mouse button !", Font, rbrush, 0, 0 );
    break;
}
}
}

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie das Programm. Lesen Sie Bilder der Formate BMP, ICO, GIF, JPG, PNG, TIFF.

Zentrieren

Version2: Beenden Sie Ihr Programm bitmap1.

Schreiben Sie in der Funktion `protected override void OnPaint(PaintEventArgs e)` unterhalb der Zeile `case 2: //Center` folgende weitere Zeilen:

```

Int32 x = (cr.Width - bmp.Width) / 2;
Int32 y = (cr.Height - bmp.Height) / 2;
g.DrawImage(bmp, x, y, bmp.Width, bmp.Height);

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie die Zentrierung.

Horizontal strecken

Version3: Beenden Sie Ihr Programm bitmap1.

Schreiben Sie in der Funktion `protected override void OnPaint(PaintEventArgs e)` unterhalb der Zeile `case 3: //Horizontal stretch` folgende weitere Zeilen:

```

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

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie die Streckung.

Vertikal strecken

Version4: Beenden Sie Ihr Programm bitmap1.

Schreiben Sie in der Funktion `protected override void OnPaint(PaintEventArgs e)` unterhalb der Zeile `case 4: //Vertical stretch` folgende weitere Zeilen:

```

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

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie die Streckung.

Maximale Größe

Version5: Beenden Sie Ihr Programm bitmap1.

Schreiben Sie in der Funktion `protected override void OnPaint(PaintEventArgs e)` unterhalb der Zeile `case 5: //Full size` folgende weitere Zeilen:

```

g.DrawImage( bmp, cr );

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Ziehen Sie am Fensterrand, um die Vergrößerung zu erproben.

Spiegeln

Version6: Beenden Sie Ihr Programm bitmap1.

Schreiben Sie in der Funktion `protected override void OnPaint(PaintEventArgs e)` unterhalb der Zeile `case 6: //Mirror` folgende weitere Zeilen:

```

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 );

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5. Erproben Sie die Spiegelungen.

Zoom-Animation

Version7: Beenden Sie Ihr Programm `bitmap1`.

Schreiben Sie in der Funktion `protected override void OnPaint(PaintEventArgs e)` unterhalb der Zeile `case 7: //Zoom animation` folgende weitere Zeilen:

```
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 );
```

Klicken Sie `Debug -> Start Without Debugging Ctrl F5`. Erproben Sie die Zoom-Animation.

Drehanimation

Version8: Beenden Sie Ihr Programm `bitmap1`.

Schreiben Sie in der Funktion `protected override void OnPaint(PaintEventArgs e)` unterhalb der Zeile `case 8: //Rotation animation` folgende weitere Zeilen:

```
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 );
```

Klicken Sie `Debug -> Start Without Debugging Ctrl F5`. Erproben Sie die Drehung um 90 Grad.

Beispielbilder

Im Prinzip sollte das Programm alle Bildformate `BMP`, `ICO`, `GIF`, `JPG`, `PNG`, `TIFF` lesen und anzeigen. Falls Sie eine alte Graphikkarte mit 8 Bit benutzen und/oder falls Sie Ihren Desktop auf 256 Farben eingestellt haben, kann es sein, dass die Farben ganz schlecht aussehen.

Falls Sie keine `*.bmp` - Dateien auf Ihrer Harddisk finden, benutzen Sie folgende Beispielbilder:

Download: [Butterfly.bmp 217 kB 24Bit-TrueColor-Bild](#)

Download: [Madonna.bmp 18 kB 8Bit-Grauwert-Bild](#)

Download: [Lena256.bmp 66 kB 8Bit-Grauwert-Bild](#)

Download: [Lena512.bmp 258 kB 8Bit-Grauwert-Bild](#)

Download: [Angiography.bmp 66 kB 8Bit-Grauwert-Bild](#)

Weitere Aufgaben

Klicken Sie auf `Help` in der Menüleiste von Visual Studio. Klicken Sie auf das Untermenü `Index`.

Gehen Sie in das Feld `Filtered by:` und wählen Sie dort `.NET Framework`. Dann geben Sie im Feld `Look for:` folgende Schlüsselwörter ein:

`Image.FromFile`, `Bitmap.RawFormat`, `Bitmap.RawFormat.Guid`, `Bitmap.PixelFormat`, `Graphics.DrawImage`. Lesen Sie die Hilfetexte. Die Hilfetexte überdecken Ihren Code. Wenn Sie mit dem Lesen fertig sind, entfernen Sie die Hilfetexte mit deren X-Button in deren rechter oberer Fensterecke.

Verändern Sie die Bildposition in `case 2: //Center`.

Verändern Sie die Streckungen in `case 3` und `case 4`.

Verändern Sie die Schrittweite und die Zoomecke in `case 5: //Zoom animation`.

Verändern Sie die Schrittweite und die Drehrichtung in `case 6: //Rotation animation`.

Beenden Sie Visual Studio, starten Sie den Explorer, löschen Sie die gesamte Directory `C:\temp\bitmap1`

Starten Sie Visual Studio wieder und erzeugen dasselbe Programm so oft, bis Sie das Programm ohne Anleitung und schnell von Null an erstellen, verändern und bedienen können.

Erfinden und erproben Sie neue Varianten des Programms in Form von neuen Projekten: `bitmap2`, `bitmap3` usw. nach obigem Muster.