

# Course 2DCis: 2D-Computer Graphics with C#

## Chapter C4: The Complete Code of the Animation Project

Copyright © by V. Miszalok, last update: 04-12-2005

Copy all this code into an empty Form1.cs of a new Windows Application C#-project anim1 and (in case of VS 2005) clear Form1.Designer.cs and Program.cs.

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

public class Form1 : Form
{ public static void Main() { Application.Run( new Form1() ); }
  static Graphics g, bitmap_g;
  static Bitmap bitmap;
  static Single zoom      = 1.01f;
  static Single cosinus   = (Single)Math.Cos( Math.PI/180.0 );
  static Single sinus     = (Single)Math.Sin( Math.PI/180.0 );
  static Brush  redbrush  = new SolidBrush( Color.Red );
  static Pen    blackpen  = SystemPens.ControlText;
  static Font   arial10   = new Font( "Arial", 10 );
  static Int32  myWidth, myHeight;
  static PointF[] pf;
  ArrayList polygon = new ArrayList();
  Point p0 = new Point();
  Point p1 = new Point();
  Timer myTimer = new Timer();

  public Form1()
  { Text = "Anim1: Draw an Endless Animation";
    Width = 800;
    Height = 600;
    g = this.CreateGraphics();
    SetStyle(ControlStyles.ResizeRedraw,true);
    myTimer.Tick += new EventHandler( OnTimer );
    myTimer.Interval = 1;
  }
  protected override void OnMouseDown( MouseEventArgs e )
  { myTimer.Stop();
    polygon.Clear(); Invalidate();
    p0.X = e.X;
    p0.Y = e.Y;
    polygon.Add( p0 );
  }
  protected override void OnMouseMove( MouseEventArgs e )
  { if ( e.Button == MouseButton.None ) return;
    p1.X = e.X;
    p1.Y = e.Y;
    Int32 dx = p1.X - p0.X;
    Int32 dy = p1.Y - p0.Y;
    if ( dx*dx + dy*dy < 100 ) return;
    g.DrawLine( blackpen, p0, p1 );
    polygon.Add( p1 );
    p0 = p1;
  }
  protected override void OnMouseUp( MouseEventArgs e )
  { if ( polygon.Count < 2 ) return;
    pf = new PointF[polygon.Count];
    for ( Int32 i=0; i < polygon.Count; i++ ) pf[i] = (Point)polygon[i];
    myTimer.Start();
  }
}
```

```

protected override void OnPaint( PaintEventArgs e )
{ e.Graphics.DrawString( "Press the left mouse button and move!", Font,
    redbrush, Width/2-50, 0 );
}

protected override void OnResize( System.EventArgs e )
{ g = this.CreateGraphics();
  g.Clear( SystemColors.Control );
  myWidth = ClientRectangle.Width;
  myHeight = ClientRectangle.Height;
  if ( bitmap != null ) bitmap.Dispose();
  bitmap = new Bitmap( myWidth, myHeight );
  if ( bitmap_g != null ) bitmap_g.Dispose();
  bitmap_g = Graphics.FromImage( bitmap );
}

protected static void OnTimer( Object myObject, EventArgs myEventArgs )
{ Single x, y, xmin, ymin, xmax, ymax, xmid, ymid;
  xmin = xmax = pf[0].X;
  ymin = ymax = pf[0].Y;
  for ( Int32 i=0; i < pf.Length; i++ )
  { x = pf[i].X;
    y = pf[i].Y;
    if ( x < xmin ) xmin = x;
    if ( x > xmax ) xmax = x;
    if ( y < ymin ) ymin = y;
    if ( y > ymax ) ymax = y;
  }
  xmid = (xmin+xmax) / 2;
  ymid = (ymin+ymax) / 2;
  if ( xmin < 0 || ymin < 0 || xmax > myWidth || ymax > myHeight )
  { //g.Clear( SystemColors.Control );
    g.DrawString( "Press the left mouse button and move!",
      arial10, redbrush, 320, 0 );
    zoom = 0.99f;
  }
  if ( xmax - xmin < 50 || ymax - ymin < 50 )
  { //g.Clear( SystemColors.Control );
    zoom = 1.01f;
  }
  for ( Int32 i=0; i < pf.Length; i++ )
  { x = pf[i].X - xmid;
    y = pf[i].Y - ymid;
    x *= zoom;
    y *= zoom;
    Single xx = x*cosinus - y*sinus;
    Single yy = x*sinus + y*cosinus;
    pf[i].X = xx + xmid;
    pf[i].Y = yy + ymid;
  }
  bitmap_g.FillRectangle( SystemBrushes.Control, 0, 0, myWidth, myHeight );
  bitmap_g.DrawLines( blackpen, pf );
  g.DrawImage( bitmap, 0, 0 ); //en bloc transfer
}
}

```