

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

### Chapter C2: The Complete Code of the Draw Project

Copyright © by V. Miszalok, last update: 16-10-2009

Copy all this code into an empty Form1.cs of a new Windows Application C#-project draw1 and clear Form1.Designer.cs and Program.cs.

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

public class Form1 : Form
{ [STAThread] static void Main() { Application.Run( new Form1() ); }
  Graphics g;
  Int16 i, n;
  const Int16 nMax = 100;
  Point[] polygon = new Point[nMax];
  Point p0, p1, mid_of_p, mid_of_r;
  Rectangle minmax;
  Double perimeter, area;
  Brush redbrush = new SolidBrush( Color.Red );
  Brush graybrush = SystemBrushes.Control;
  Brush blackbrush = SystemBrushes.ControlText;
  Pen blackpen = SystemPens.ControlText;
  Pen redpen = new Pen( Color.Red );
  Pen greenpen = new Pen( Color.Green );
  String myline;

  public Form1()
  { Text = "Draw1: A Scribble Program";
    Width = 800;
    Height = 600;
    g = this.CreateGraphics();
  }

  protected override void OnMouseDown( MouseEventArgs e )
  { polygon[0] = p0 = e.Location;
    n = 1;
    Invalidate();
  }

  protected override void OnMouseMove( MouseEventArgs e )
  { if ( e.Button == MouseButtons.None ) return;
    p1 = e.Location;
    Int32 dx = p1.X - p0.X;
    Int32 dy = p1.Y - p0.Y;
    if ( dx*dx + dy*dy < 100 ) return;
    if ( n >= nMax-1 ) return;
    g.DrawLine( blackpen, p0, p1 );
    polygon[n++] = p0 = p1;
    g.DrawString( myline, Font, graybrush, 0, Font.Height );
    myline = String.Format( "{0}, {1}", p1.X, p1.Y );
    g.DrawString( myline, Font, blackbrush, 0, Font.Height );
    g.DrawRectangle( blackpen, p1.X-3, p1.Y-3, 7, 7 );
  }
}
```

```

protected override void OnMouseUp( MouseEventArgs e )
{
    if ( n < 2 ) return;
    p0 = polygon[n++] = polygon[0];
    perimeter = area = 0;
    mid_of_p.X = mid_of_p.Y = 0;
    Int32 xmin, xmax, ymin, ymax;
    xmin = xmax = p0.X;
    ymin = ymax = p0.Y;
    for ( i=1; i < n; i++ )
    {
        p1 = polygon[i];
        Double dx = p1.X - p0.X;
        Double dy = p1.Y - p0.Y;
        Double my = (p0.Y + p1.Y) / 2.0;
        perimeter += Math.Sqrt( dx*dx + dy*dy );
        area      += dx * my;
        mid_of_p.X += p1.X;
        mid_of_p.Y += p1.Y;
        if ( p1.X < xmin ) xmin = p1.X;
        if ( p1.X > xmax ) xmax = p1.X;
        if ( p1.Y < ymin ) ymin = p1.Y;
        if ( p1.Y > ymax ) ymax = p1.Y;
        p0 = p1;
    }
    mid_of_r.X = ( xmax + xmin ) / 2;
    mid_of_r.Y = ( ymax + ymin ) / 2;
    mid_of_p.X /= n-1;
    mid_of_p.Y /= n-1;
    minmax.X = xmin-1; minmax.Width = xmax - xmin + 2;
    minmax.Y = ymin-1; minmax.Height = ymax - ymin + 2;
    Invalidate();
}

protected override void OnPaint( PaintEventArgs e )
{
    g.DrawString( "Press the left mouse button and move!", Font, redbrush, 0, 0 );
    if ( n < 2 ) return;
    myline = String.Format( "Perimeter= {0}, Area= {1}", (Int32)perimeter, (Int32)area );
    g.DrawString( myline, Font, blackbrush, 0, 2*Font.Height );
    for ( i=0; i < n-1; i++ ) g.DrawLine( blackpen, polygon[i], polygon[i+1] );
    for ( i=0; i < n-3; i+=3 )
        g.DrawBezier( redpen, polygon[i], polygon[i+1], polygon[i+2], polygon[i+3] );
    g.DrawRectangle( greenpen, minmax );
    g.DrawLine( greenpen, mid_of_r.X - 4, mid_of_r.Y, mid_of_r.X + 4, mid_of_r.Y );
    g.DrawLine( greenpen, mid_of_r.X, mid_of_r.Y - 4, mid_of_r.X, mid_of_r.Y + 4 );
    g.FillEllipse( blackbrush, mid_of_p.X-5, mid_of_p.Y-5, 11, 11 );
}
}

```