makes a new window out of a graphics path that has been traced on this forms space

```
/*
GDI+ Programming in C# and VB .NET
by Nick Symmonds

Publisher: Apress
ISBN: 159059035X
*/
using System;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace CustomWindow
{
///

/// This project makes a new window out of a graphics path that has been traced on this forms space.
/// Features include: Smoothing capability via turning sub paths into curves, etc
/// Use point for stretching curve.
///
/// Have some kind of window that allows the user to set or view smoothing factor.
/// The user can then see the shape in the smoother form side-by-side with the original.
```

```
/// The user can then choose which one to make the new window with.
/// Show the smoothing factor as percentage reduction in number of points = 1-(p1/p2)
/// Have user choose some basic aspects of the form.
/// min and max buttons
/// border style
/// inflation size
/// have another form that has a slider so the user can graphically size the shape
/// opacity
/// background color
/// background image
/// Quit button on form
///
public class CustomWindow : System.Windows.Forms.Form
{
#region Class Local Storage
private GraphicsPath m_WindowPath;
private GraphicsPath m_path;
private Point m_StartPoint;
private Point m_LastPoint;
private bool m_Drawing;
private Rectangle InvalidRect;
private Cursor DrawCursor;
#endregion

private System.Windows.Forms.MainMenu mainMenu1;
private System.Windows.Forms.MenuItem menuItem1;
private System.Windows.Forms.MenuItem mnuExit;
private System.Windows.Forms.MenuItem menuItem3;
private System.Windows.Forms.MenuItem menuItem4;
private System.Windows.Forms.MenuItem menuItem5;
private System.Windows.Forms.MenuItem menuItem2;
private System.Windows.Forms.MenuItem menuItem6;
private System.Windows.Forms.MenuItem menuItem7;
private System.Windows.Forms.MenuItem menuItem8;
private System.Windows.Forms.MenuItem menuItem9;
private System.Windows.Forms.TreeView PathTree;
```

```
private System.Windows.Forms.Panel P1;
private System.Windows.Forms.Splitter TreeSplitter;
private System.Windows.Forms.MenuItem menuItem10;
private System.Windows.Forms.Splitter PanelSplitter;
private System.Windows.Forms.RichTextBox DebugWindow;

///

/// Required designer variable.
///
private System.ComponentModel.Container components = null;

public CustomWindow()
{
InitializeComponent();

this.Icon = new Icon("Icon.ico");
DrawCursor = new Cursor("Pen.cur");

this.Size = new Size(800, 600);
this.SetStyle(ControlStyles.AllPaintingInWmPaint,true);
this.SetStyle(ControlStyles.DoubleBuffer,true);

// this.MouseDown += new MouseEventHandler(this.M_Down);
// this.MouseUp += new MouseEventHandler(this.M_Up);
// this.MouseMove += new MouseEventHandler(this.M_Move);

P1.Paint += new PaintEventHandler(this.PanelPaint);
P1.MouseDown += new MouseEventHandler(this.M_Down);
P1.MouseUp += new MouseEventHandler(this.M_Up);
P1.MouseMove += new MouseEventHandler(this.M_Move);

m_WindowPath = new GraphicsPath();
m_path = new GraphicsPath();
InvalidRect = Rectangle.Empty;

//Set RichTextBox properties
DebugWindow.Height = this.Height /8;
DebugWindow.Dock = DockStyle.Bottom;
```

```
// Set properties of TreeView control.
PathTree.Dock = DockStyle.Left;
PathTree.Width = this.ClientSize.Width / 6;
PathTree.TabIndex = 0;
PathTree.Nodes.Add(“Shapes”);

//Set Panel Properties
P1.Dock = DockStyle.Fill;
P1.BackColor = Color.Bisque;

//Set up the splitters
TreeSplitter.Location = new Point(PathTree.Width, 0);
TreeSplitter.Size = new Size(2, this.Height);
TreeSplitter.BackColor = Color.Black;

PanelSplitter.Dock = DockStyle.Bottom;
PanelSplitter.Height = 2;
PanelSplitter.BackColor = Color.Black;

}

///

/// Clean up any resources being used.
///
protected override void Dispose( bool disposing )
{
if( disposing )
{
if (components != null)
{
components.Dispose();
}
}
base.Dispose( disposing );
}

#region Windows Form Designer generated code
///
```

```
/// Required method for Designer support – do not modify
/// the contents of this method with the code editor.
///
private void InitializeComponent()
{
this.mainMenu1 = new System.Windows.Forms.MainMenu();
this.menuItem1 = new System.Windows.Forms.MenuItem();
this.mnuExit = new System.Windows.Forms.MenuItem();
this.menuItem3 = new System.Windows.Forms.MenuItem();
this.menuItem4 = new System.Windows.Forms.MenuItem();
this.menuItem5 = new System.Windows.Forms.MenuItem();
this.menuItem2 = new System.Windows.Forms.MenuItem();
this.menuItem6 = new System.Windows.Forms.MenuItem();
this.menuItem7 = new System.Windows.Forms.MenuItem();
this.menuItem8 = new System.Windows.Forms.MenuItem();
this.menuItem9 = new System.Windows.Forms.MenuItem();
this.PathTree = new System.Windows.Forms.TreeView();
this.P1 = new System.Windows.Forms.Panel();
this.TreeSplitter = new System.Windows.Forms.Splitter();
this.menuItem10 = new System.Windows.Forms.MenuItem();
this.DebugWindow = new System.Windows.Forms.RichTextBox();
this.PanelSplitter = new System.Windows.Forms.Splitter();
this.SuspendLayout();
//
// mainMenu1
//
this.mainMenu1.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {
this.menuItem1,
this.menuItem3,
this.menuItem7});
//
// menuItem1
//
this.menuItem1.Index = 0;
this.menuItem1.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {
```

```
this.mnuExit});
this.menuItem1.Text = “File”;
//
// mnuExit
//
this.mnuExit.Index = 0;
this.mnuExit.Text = “&Exit”;
this.mnuExit.Click                    +=                    new
System.EventHandler(this.mnuExit_Click);
//
// menuItem3
//
this.menuItem3.Index = 1;
this.menuItem3.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {
this.menuItem4,
this.menuItem10,
this.menuItem5,
this.menuItem2,
this.menuItem6,
this.menuItem9});
this.menuItem3.Text = “Shape”;
//
// menuItem4
//
this.menuItem4.Index = 0;
this.menuItem4.Text = “Create”;
//
// menuItem5
//
this.menuItem5.Index = 2;
this.menuItem5.Text = “Resize”;
//
// menuItem2
//
this.menuItem2.Index = 3;
this.menuItem2.Text = “-“;
```

```
//
// menuItem6
//
this.menuItem6.Index = 4;
this.menuItem6.Text = “Load Shape”;
//
// menuItem7
//
this.menuItem7.Index = 2;
this.menuItem7.MenuItems.AddRange(new
System.Windows.Forms.MenuItem[] {
this.menuItem8});
this.menuItem7.Text = “Form”;
//
// menuItem8
//
this.menuItem8.Index = 0;
this.menuItem8.Text = “Spawn”;
//
// menuItem9
//
this.menuItem9.Index = 5;
this.menuItem9.Text = “Save Shape”;
//
// PathTree
//
this.PathTree.ImageIndex = -1;
this.PathTree.Name = “PathTree”;
this.PathTree.SelectedImageIndex = -1;
this.PathTree.Size = new System.Drawing.Size(88, 488);
this.PathTree.TabIndex = 0;
//
// P1
//
this.P1.Location = new System.Drawing.Point(112, 16);
this.P1.Name = “P1”;
this.P1.Size = new System.Drawing.Size(464, 472);
```

```
this.P1.TabIndex = 1;
//
// TreeSplitter
//
this.TreeSplitter.Name = “TreeSplitter”;
this.TreeSplitter.Size = new System.Drawing.Size(8, 573);
this.TreeSplitter.TabIndex = 2;
this.TreeSplitter.TabStop = false;
//
// menuItem10
//
this.menuItem10.Index = 1;
this.menuItem10.Text = “Smooth”;
//
// DebugWindow
//
this.DebugWindow.Location = new System.Drawing.Point(24, 520);
this.DebugWindow.Name = “DebugWindow”;
this.DebugWindow.Size = new System.Drawing.Size(536, 40);
this.DebugWindow.TabIndex = 3;
this.DebugWindow.Text = “richTextBox1”;
//
// PanelSplitter
//
this.PanelSplitter.Location = new System.Drawing.Point(8, 0);
this.PanelSplitter.Name = “PanelSplitter”;
this.PanelSplitter.Size = new System.Drawing.Size(8, 573);
this.PanelSplitter.TabIndex = 4;
this.PanelSplitter.TabStop = false;
//
// CustomWindow
//
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(592, 573);
this.Controls.AddRange(new System.Windows.Forms.Control[] {
this.PanelSplitter,
this.DebugWindow,
```

```
this.TreeSplitter,
this.P1,
this.PathTree});
this.Menu = this.mainMenu1;
this.Name = “CustomWindow”;
this.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = “CustomWindow”;
this.Load += new System.EventHandler(this.CustomWindow_Load);
this.ResumeLayout(false);

}
#endregion

#region Main Entry Point
///

/// The main entry point for the application.
///
[STAThread]
static void Main()
{
Application.Run(new CustomWindow());
}
#endregion

private void CustomWindow_Load(object sender, System.EventArgs
e)
{
}

// protected override void OnPaint(PaintEventArgs e)
// {
// Graphics G = e.Graphics;
// PaintMe(e.Graphics);
// }

private void PanelPaint(object sender, PaintEventArgs e)
{
```

```
PaintMe(e.Graphics);
}

private void PaintMe(Graphics G)
{
G.SmoothingMode = SmoothingMode.AntiAlias;

if (m_WindowPath.PointCount > 0)
G.DrawPath(Pens.Black, m_WindowPath);

if (m_path.PointCount > 0)
G.DrawPath(Pens.Red, m_path);
}

private void M_Down(object sender, MouseEventArgs m)
{
m_StartPoint = new Point(m.X, m.Y);
m_LastPoint = m_StartPoint;
m_WindowPath = new GraphicsPath();
m_Drawing = true;
P1.Invalidate();
}

private void M_Up(object sender, MouseEventArgs m)
{
m_WindowPath.CloseFigure();
m_Drawing = false;

//Smooth out the path by reducing lines that make up path
m_path = SmootherPath(m_WindowPath);
Matrix Q = new Matrix();
Q.Translate(50, 5);
m_path.Transform(Q);

int a = m_WindowPath.PointCount;
int b = m_path.PointCount;

//Draw the paths and make a window.
P1.Invalidate();
```

```
MakeWindow();
}

private void M_Move(object sender, MouseEventArgs m)
{
if(m_Drawing)
{
m_WindowPath.AddLine(m_LastPoint.X, m_LastPoint.Y, m.X, m.Y);
m_LastPoint.X = m.X;
m_LastPoint.Y = m.Y;

InvalidRect = Rectangle.Truncate(m_WindowPath.GetBounds());
InvalidRect.Inflate( new Size(2, 2) );
P1.Invalidate(InvalidRect);
}
}

///
/// Start first with reducing the number of lines in this
graphics path
/// 1. read first point
/// 2. if x value of next point = x value of last point then
skip
/// 3. if x value of next point != value of last point then…
/// 3a. make line based on these two points
/// 3b. add line to new path
/// 3c. first point = next point
/// 4. repeat 1-3c
/// 5. Repeat 1-4 for both X and Y
///
private GraphicsPath SmootherPath(GraphicsPath gp)
{
PathData pd = gp.PathData;
PointF pt1 = new Point(-1,-1);

//First do all values in the X range
GraphicsPath FixedPath_X = new GraphicsPath();
foreach (PointF p in pd.Points)
```

```
{
if (pt1.X == -1)
{
pt1 = p;
continue;
}
// If I introduced an error factor here I could smooth it out
even more
if (p.X != pt1.X)
{
FixedPath_X.AddLine(pt1, p);
pt1 = p;
}
}
FixedPath_X.CloseFigure();

//Second do all values in the Y range
pd = FixedPath_X.PathData;
pt1 = new Point(-1,-1);
GraphicsPath FixedPath_Y = new GraphicsPath();
foreach (PointF p in pd.Points)
{
if (pt1.Y == -1)
{
pt1 = p;
continue;
}
// If I introduced an error factor here I could smooth it out
even more
if (p.Y != pt1.Y)
{
FixedPath_Y.AddLine(pt1, p);
pt1 = p;
}
}
FixedPath_Y.CloseFigure();
```

```
return FixedPath_Y;
}

private void MakeWindow()
{
Form frm = new Form();
GraphicsPath path = (GraphicsPath)m_WindowPath.Clone();
Matrix Xlate = new Matrix();

//Find the lowest Y value and normalize all Y values to zero
//Find the lowest X value and normalize all X values to zero
PointF[] p = path.PathPoints;
int Xoffset = 9999;
int Yoffset = 9999;
foreach (PointF p2 in p)
{
if (p2.X < Xoffset) Xoffset = (int)p2.X; if (p2.Y < Yoffset) Yoffset = (int)p2.Y; } Xlate.Translate(-Xoffset, -Yoffset); path.Transform(Xlate); // Set the paractical viewing region of the form frm.Region = new Region(path); //Set the size of the form Rectangle frmRect = Rectangle.Truncate(path.GetBounds()); frm.Size = frmRect.Size; //Set some other parameters frm.StartPosition = FormStartPosition.CenterParent; frm.FormBorderStyle = FormBorderStyle.FixedSingle; frm.ShowDialog(); frm.Dispose(); Xlate.Dispose(); } private void mnuExit_Click(object sender, System.EventArgs e) { this.Close(); } } } CustomWindow.zip( 34 k)[/csharp]
```