

Course 2D_WPF: 2D-Computer Graphics with C# + WPF

Chapter C1a: The Intro Project Written in XAML and C#

The Complete Code

Copyright © by V. Miszalok, last update: 2011-02-08

Guidance for **Visual C# 2010 Express**, see [Introduction into all Courses](#).

- 1) Main Menu after start of VC# 2010: Tools → Options →
check lower left checkbox: Show all Settings → Projects and Solutions →
Visual Studio projects location: → C:\temp
- 2) Main Menu after start of VC# 2010: File → New Project... →
Installed templates: WPF Application → Name: intro1_XAML_CS → OK.

MainWindow.xaml

```
<Window x:Class="intro1_XAML_CS.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="A first WPF-Program written in XAML and C#" Height="300" Width="300"
    Top="50" Left="50" SizeChanged="Window_SizeChanged">
  <Canvas Name="myCanvas">
    <Canvas.Background>
      <LinearGradientBrush>
        <GradientStop Color="Blue" Offset="0" />
        <GradientStop Color="Green" Offset="0.5"/>
        <GradientStop Color="Red" Offset="1" />
      </LinearGradientBrush>
    </Canvas.Background>
    <Line Name="line1" Stroke="Black" StrokeThickness="3"/>
    <Line Name="line2" Stroke="Black" StrokeThickness="3"/>
    <Rectangle Name="rect" Stroke="Black" StrokeThickness="3" Fill="White"/>
    <Ellipse Name="elli" Stroke="Black" StrokeThickness="3">
      <Ellipse.Fill>
        <RadialGradientBrush x:Name="elliBrush" RadiusX="0.1" RadiusY="0.1" SpreadMethod="Repeat">
          <GradientStop x:Name="stop1" Offset="0" />
          <GradientStop x:Name="stop2" Offset="0.5"/>
          <GradientStop x:Name="stop3" Offset="1" />
        </RadialGradientBrush>
      </Ellipse.Fill>
    </Ellipse>
    <DockPanel Name="myPanel">
      <TextBox Name="left" Text="left" VerticalAlignment="Center" DockPanel.Dock="Left" />
      <TextBox Name="right" Text="right" VerticalAlignment="Center" DockPanel.Dock="Right" />
      <TextBox Name="top" Text="top" HorizontalAlignment="Center" DockPanel.Dock="Top" />
      <TextBox Name="bottom" Text="bottom" HorizontalAlignment="Center" DockPanel.Dock="Bottom" />
      <TextBox Name="centerText" HorizontalAlignment="Center" VerticalAlignment="Center"
        TextAlignment="Center"/>
    </DockPanel>
  </Canvas>
</Window>
```

MainWindow.xaml.cs

```

using System;
using System.Windows;
using System.Windows.Media;
using System.Windows.Controls;
using System.Windows.Threading;

namespace intro1_XAML_CS
{
    public partial class MainWindow:Window
    {
        double zoom = 1.1;
        double angle = 0;
        Random r = new Random();
        Byte r1, g1, b1, r2, g2, b2, r3, g3, b3;
        Point radialGradientBrushOrigin = new Point( 0.5, 0.5 );
        public MainWindow()
        {
            InitializeComponent();
            //myTimer is a clock intended to animate the window size
            DispatcherTimer myTimer = new DispatcherTimer();
            myTimer.Interval = TimeSpan.FromMilliseconds( 1 );
            myTimer.Tick += TimerOnTick;
            myTimer.Start();
            //initial random colors of RadialGradientBrush "elliBrush"
            r1 = (Byte)r.Next(255); g1 = (Byte)r.Next(255); b1 = (Byte)r.Next(255);
            r2 = (Byte)r.Next(255); g2 = (Byte)r.Next(255); b2 = (Byte)r.Next(255);
            r3 = (Byte)r.Next(255); g3 = (Byte)r.Next(255); b3 = (Byte)r.Next(255);
        }
        private void TimerOnTick( Object sender, EventArgs args )
        {
            if ( this.ActualWidth < 200 ) zoom = 1.1; //fast enlargement
            if ( this.ActualWidth > 800 ) zoom = 0.99; //slow shrinking
            this.Width *= zoom;
            this.Height *= zoom;
            centerText.FontSize *= zoom;
        }
        private void Window_SizeChanged( object sender, SizeChangedEventArgs e )
        {
            //compose the text of the "centerText"-TextBox
            String s1 = "Hello World " + DateTime.Now.ToString() + "\n";
            int width = Convert.ToInt32( this.Width );
            int height = Convert.ToInt32( this.Height );
            String s2 = "Window Size = " + width.ToString() + " x " + height.ToString() + "\n";
            width = Convert.ToInt32( myCanvas.ActualWidth );
            height = Convert.ToInt32( myCanvas.ActualHeight );
            String s3 = "Client Size = " + width.ToString() + " x " + height.ToString() + "\n";
            String s4 = String.Format( "Font Size = {0,2:F1}", centerText.FontSize );
            centerText.Text = s1 + s2 + s3 + s4;
            //adjust all contents of "MainWindow" to its new window size
            line1.X1 = 0; line1.Y1 = 0; line1.X2 = myCanvas.ActualWidth; line1.Y2 = myCanvas.ActualHeight;
            line2.X1 = myCanvas.ActualWidth; line2.Y1 = 0; line2.X2 = 0; line2.Y2 = myCanvas.ActualHeight;
            Canvas.SetLeft( rect, myCanvas.ActualWidth / 5 );
            Canvas.SetLeft( elli, myCanvas.ActualWidth / 5 );
            Canvas.SetTop ( rect, myCanvas.ActualHeight/5 );
            Canvas.SetTop ( elli, myCanvas.ActualHeight/5 );
            rect .Width = elli.Width = 3 * myCanvas.ActualWidth / 5;
            rect .Height = elli.Height = 3 * myCanvas.ActualHeight / 5;
            myPanel.Width = myCanvas.ActualWidth;
            myPanel.Height = myCanvas.ActualHeight;
            //Increment colors of RadialGradientBrush "elliBrush".
            //Whenever a color byte exceeds 255, it automatically resets to 0.
            stop1.Color = Color.FromRgb( r1++, g1++, b1++ );
            stop2.Color = Color.FromRgb( r2++, g2++, b2++ );
            stop3.Color = Color.FromRgb( r3++, g3++, b3++ );
            //Move the center of RadialGradientBrush "elliBrush" slightly around point (0.5, 0.5).
            radialGradientBrushOrigin.X = 0.5 + 0.05 * Math.Cos(angle);
            radialGradientBrushOrigin.Y = 0.5 + 0.05 * Math.Sin(angle);
            elliBrush.GradientOrigin = radialGradientBrushOrigin;
            angle += Math.PI / 32;
        }
    }
}

```