

Course 3D_WPF: 3D-Computer Graphics with C# + WPF

Chapter C1: Moving Triangles Complete Code



Copyright © by V. Miszalok, last update: 2010-01-07

Guidance for Visual C# 2010 Express:

- 1) Main Menu after start of Visual C# 2010 Express: Tools → Options → check lower left checkbox: Show all Settings → Projects and Solutions → Visual Studio projects location: → C:\temp
- 2) Main Menu after start of Visual C# 2010 Express: File → New Project... → WPF Application Name: triangle1 → OK.
- 3) Main menu of Visual C# 2010 Express → Tools → Options... → An Options-window appears. **Double-click** the branch Text Editor. **Double-click** C#. **Double-click** Formatting. Click General. Uncheck all three check boxes. Click Indentation. Uncheck all four check boxes. Click New Lines. Uncheck all thirteen check boxes. Click Spacing. Uncheck all twenty three check boxes. Click IntelliSense. Uncheck all six check boxes. Quit the Options- window with the OK-button.

Replace the default code of MainWindow.xaml and of MainWindow.xaml.cs by the following codes:

MainWindow.xaml

```
<Window x:Class="triangle1.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="triangle1">
  <Window.Resources>
    <MeshGeometry3D x:Key="coordinates"
      Positions      ="-1 -1 0, 1 -1 0, 0 1 0"
      Normals        =" 0 0 1, 0 0 1, 0 0 1"
      TextureCoordinates=" 0 1 , 1 1 , 0.5 0" />
    <ImageBrush      x:Key="brush" ImageSource="http://www.miszalok.de/Images/Foto.jpg" />
    <DiffuseMaterial x:Key="front" Brush="{StaticResource brush}" />
    <DiffuseMaterial x:Key="back" Brush="Gray" />
  </Window.Resources>

  <Viewport3D>
    <ModelVisual3D>
      <ModelVisual3D.Content>
        <Model3DGroup x:Name="group">
          <GeometryModel3D x:Name="triangle_prototype"
            Geometry=      "{StaticResource coordinates}"
            Material=      "{StaticResource front}"
            BackMaterial=" {StaticResource back}" />
          <AmbientLight Color="#404040" />
          <DirectionalLight Color="#ffffff" Direction="0 0 -1" />
        </Model3DGroup>
      </ModelVisual3D.Content>
    </ModelVisual3D>
    <Viewport3D.Camera>
      <PerspectiveCamera
        Position      = "0 0 5"
        LookDirection="0 0 -1"
        UpDirection   ="0 1 0" />
    </Viewport3D.Camera>
  </Viewport3D>
</Window>
```

MainWindow.xaml.cs:

```

using System;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Threading;
using System.Windows.Media.Media3D;
using System.Windows.Markup;
using System.IO;
using System.Xml;

namespace triangle1
{
    public partial class MainWindow: Window
    {
        const Int32 nTriangles = 50;
        Matrix3D[] matrix = new Matrix3D [nTriangles];
        GeometryModel3D[] triangle = new GeometryModel3D[nTriangles];
        float[] dx = new float [nTriangles];
        float[] dy = new float [nTriangles];
        float[] dz = new float [nTriangles];
        //This quaternion defines a rotation of 1 degree around axis (1,1,1)
        //It will be used in void TimerOnTick(...) to animate all triangles
        Quaternion quaternion = new Quaternion( new Vector3D(1,1,1), 1 );
        DispatcherTimer timer = new DispatcherTimer();
        Random r = new Random();

        public MainWindow()
        {
            InitializeComponent();
            //Remember triangle_prototype as XAML-string
            string triangle_prototype_string = XamlWriter.Save( triangle_prototype );
            group.Children.Remove( triangle_prototype ); //Kick the prototype out
            for ( int i = 0; i < nTriangles; i++ )
            {
                dx[i] = 2*(float)r.NextDouble() - 1; //random permanent translation dx
                dy[i] = 2*(float)r.NextDouble() - 1; //random permanent translation dy
                dz[i] = 2*(float)r.NextDouble() - 1; //random permanent translation dz
                matrix[i].Rotate ( new Quaternion( new Vector3D(1,1,1),
                                                    (float)r.NextDouble() * 45 ) );
                matrix[i].Scale ( new Vector3D( 0.5 ,0.5 ,0.5 ) ); //smaller
                matrix[i].Translate( new Vector3D( dx[i],dy[i],dz[i] ) ); //random shifts
                //Clone the triangle_prototype from its XAML-string
                StringReader sr = new StringReader( triangle_prototype_string );
                XmlTextReader xr = new XmlTextReader( sr );
                triangle[i] = (GeometryModel3D)XamlReader.Load( xr );
                group.Children.Add( triangle[i] );
            }
            timer.Interval = TimeSpan.FromMilliseconds( 1 );
            timer.Tick += TimerOnTick;
            timer.Start();
        }

        void TimerOnTick( Object sender, EventArgs args )
        {
            for ( int i = 0; i < nTriangles; i++ )
            {
                //Insert this rotation in front of previous transforms
                matrix[i].RotatePrepend( quaternion );
                triangle[i].Transform = new MatrixTransform3D( matrix[i] );
            }
        }
    }
}

```