Course 3D_XNA: 3D-Computer Graphics with XNA
Chapter C1: The Complete Code of the Moving Triangles

This program requires:
1. Visual C# 2008 Express or Visual Studio 2008 Professional
3. XNA Game Studio 3.0
4. DirectX End-User Runtime June 2008

Main Menu after starting VS 2008: `File` → `New Project...` → `Templates: Windows Game (3.0)` → `Name: triangle1` → `Location: C:\temp` → `Create directory for solution:` switch it off → `OK`. `Solution Explorer - triangle1`: Delete the file `Program.cs` and the content of `Game1.cs`.

Copy the following code into `Game1.cs`:

```
using System;
using Microsoft.Xna.Framework;
using Microsoft.Xna.Framework.Graphics;

static class Program
{ static void Main() { Game1 game = new Game1(); game.Run(); }

public class Game1 : Microsoft.Xna.Framework.Game
{ GraphicsDeviceManager g;
  VertexBuffer vbuffer;
  BasicEffect  effect;
  EffectPass   pass;
  float rotAngle = 0f;
  Matrix rot        = Matrix.CreateRotationY( 0f );
  Matrix zoom       = Matrix.CreateScale( 0.5f );
  Matrix leftshift  = Matrix.CreateTranslation( new Vector3( -1f, 1f, 0f ) );
  Matrix rightshift = Matrix.CreateTranslation( new Vector3(  1f, 0f, 0f ) );
  const Int32 nTriangles = 100;
  float[] dx = new float[nTriangles];
  float[] dy = new float[nTriangles];
  float[] dz = new float[nTriangles];
  float[] ax = new float[nTriangles];
  float[] ay = new float[nTriangles];
  float[] az = new float[nTriangles];
  Random r = new Random();

  public Game1()
  { Window.Title = "3DTriangleAnimation";
    Window.AllowUserResizing = true;
    g = new GraphicsDeviceManager( this );
    g.IsFullScreen = false;
    g.PreferredBackBufferWidth  = 600;
    g.PreferredBackBufferHeight = 600;
  }
```

```
 protected override void Initialize()
  { VertexPositionColor[] v = new VertexPositionColor[3];
    v[0].Position.X=-1f; v[0].Position.Y=-1f; v[0].Position.Z=0f;
    v[1].Position.X= 0f; v[1].Position.Y= 1f; v[1].Position.Z=0f;
    v[2].Position.X= 1f; v[2].Position.Y=-1f; v[2].Position.Z=0f;
    v[0].Color = Color.DarkGoldenrod;
    v[1].Color = Color.Cornsilk;
    v[2].Color = Color.MediumOrchid;
    vbuffer = new VertexBuffer(g.GraphicsDevice,typeof(VertexPositionColor),3,BufferUsage.WriteOnly);
    vbuffer.SetData< VertexPositionColor >( v );
    effect = new BasicEffect( g.GraphicsDevice, null );
    effect.View = Matrix.CreateLookAt( new Vector3(0, 0, 3), Vector3.Zero, Vector3.Up );
    effect.Projection = Matrix.CreatePerspectiveFieldOfView(MathHelper.Pi/4, 1f, 1f, 10f );
    effect.VertexColorEnabled = true;
    pass = effect.CurrentTechnique.Passes[0];
    for ( int i = 0; i < nTriangles; i++ )
    { dx[i] = (float)r.NextDouble() - 0.5f;  //random permanent translation dx
      dy[i] = (float)r.NextDouble() - 0.5f;  //random permanent translation dy
      dz[i] = (float)r.NextDouble() - 0.5f;  //random permanent translation dz
      ax[i] = (float)r.NextDouble(); //random initial pitch rotation angle
      ay[i] = (float)r.NextDouble(); //random initial yaw   rotation angle
      az[i] = (float)r.NextDouble(); //random initial roll  rotation angle
    }
    //g.GraphicsDevice.RenderState.DepthBufferEnable = false;
    //g.GraphicsDevice.RenderState.DepthBufferWriteEnable = false;
  }

  protected override void Update( GameTime gameTime )
  { g.GraphicsDevice.Vertices[0].SetSource( vbuffer, 0, VertexPositionColor.SizeInBytes );
    g.GraphicsDevice.VertexDeclaration =
            new VertexDeclaration( g.GraphicsDevice, VertexPositionColor.VertexElements );
    g.GraphicsDevice.RenderState.CullMode = CullMode.None;
  }

  protected override void Draw( GameTime gameTime )
  { g.GraphicsDevice.Clear( Color.DarkBlue );
    //rot = Matrix.CreateRotationY( rotationAngle );
    //rot = Matrix.CreateFromYawPitchRoll( rotAngle += 0.01f, rotAngle, rotAngle );
    effect.Begin();
      for ( int i = 0; i < nTriangles; i++ )
      { pass.Begin();
          rot = Matrix.CreateFromYawPitchRoll( ax[i]+=0.01f, ay[i]+=0.01f, az[i]+=0.01f );
          leftshift = Matrix.CreateTranslation( dx[i], dy[i], dz[i] );
          effect.World = zoom * rot * leftshift;
          g.GraphicsDevice.DrawPrimitives( PrimitiveType.TriangleList, 0, 1 );
        pass.End();
      }
    effect.End();
  }
} // end of class Game1
} // end of class Program
```