

# Course 3D\_MDX: 3D-Graphics with Managed DirectX 9.0

## Chapter C5: Read a Mesh from File.x

Copyright © by V. Miszalok, last update: 29-08-2006

- ↓ [Project mesh\\_file1](#)
- ↓ [Form1, OnResize, OnTimer](#)
- ↓ [Exercises](#)

### Project mesh\_file1

This chapter is a short version of a Direct3D-tutorial from Microsoft: Tutorial6. See

C:\DXSDK\Samples\Managed\Direct3D\Tutorials\Tutorial6.

Main Menu after starting VS 2005: File → New Project... → Templates: Windows Application  
Name: mesh\_file1 → Location: C:\temp → Create directory for solution: switch it off → OK  
Delete the files Program.cs and Form1.Designer.cs and the content of Form1.cs, as described in the chapters 2DCisC1 to 2DCisC4.

If You can't find a Solution Explorer-window, open it via the main menu: View → Solution Explorer. Inside the Solution Explorer-window click the plus-sign in front of mesh\_file1. A tree opens. Look for the branch "References". **Right**-click References and **left**-click Add Reference... An Add Reference dialog box opens. Scroll down to the component name: Microsoft.DirectX Version 1.0.2902.0.

Highlight this reference by a left-click **and** (holding the Strg-key pressed) two more references:

Microsoft.DirectX.Direct3D Version 1.0.2902.0 and

Microsoft.DirectX.Direct3DX Version 1.0.2902.0 or 1.0.2903.0 or 1.0.2904.0.

Quit the Add Reference dialog box with OK.

Check if three references:

Microsoft.DirectX and

Microsoft.DirectX.Direct3D and

Microsoft.DirectX.Direct3DX are now visible inside the Solution Explorer window underneath mesh\_file1 → References.

**This exercise requires C:\DXSDK\Samples\Media\Tiger\tiger.x (29 KB) and**

**C:\DXSDK\Samples\Media\Tiger\tiger.bmp (66 KB) to be found in this path.** In case these files are somewhere else, You have to change the strings myMeshFile and myTextureFile in the head of Form1.

### Form1, OnResize, OnTimer

Write the following code into the empty Form1.cs:

```
using System;
using System.Drawing;
using System.Windows.Forms;
using Microsoft.DirectX;
using Microsoft.DirectX.Direct3D;

public class Form1 : Form
{
    static void Main() { Application.Run( new Form1() ); }
    static Device device = null;
    static float xAngle, yAngle, zAngle;
    static Mesh mesh = null;
    String myMeshFile = @"C:\DXSDK\Samples\Media\Tiger\tiger.x";
    String myTextureFile = @"C:\DXSDK\Samples\Media\Tiger\tiger.bmp";
    BaseTexture texture;
    Bitmap myTexture;
    Timer myTimer = new Timer();
    MenuItem miReadMesh;
}
```

```

public Form1()
{
    miReadMesh      = new MenuItem( "Read Mesh"      , new EventHandler( MenuFileRead ) );
    MenuItem miReadTexture = new MenuItem( "Read Texture", new EventHandler( MenuFileRead ) );
    MenuItem miExit    = new MenuItem( "Exit"        , new EventHandler( MenuFileExit ) );
    MenuItem miFile = new MenuItem( "File", new MenuItem[] { miReadMesh, miReadTexture, miExit } );
    Menu = new MainMenu( new MenuItem[] { miFile } );
    Text = "DirectX3DMesh";
    myTexture = (Bitmap)Image.FromFile( myTextureFile );
    //set up the timer
    myTimer.Tick += new EventHandler( OnTimer );
    myTimer.Interval = 1;
    ClientSize = new Size( 1024, 800 ); //calls OnResize( ... )
}

protected override void OnResize( System.EventArgs e )
//Whenever the window changes we have to initialize DirectX3D from scratch
{ myTimer.Stop();// stop the timer during initialization
  try
  { PresentParameters presentParams = new PresentParameters();
    presentParams.Windowed = true; //no full screen display
    presentParams.SwapEffect = SwapEffect.Discard; //no swap buffer
    presentParams.EnableAutoDepthStencil = true; //with depth buffer
    presentParams.AutoDepthStencilFormat = DepthFormat.D16; //16 bit depth
    //create a new D3D-device that serves as canvas
    if ( device != null ) device.Dispose(); //free the old canvas if any
    device = new Device( 0, DeviceType.Hardware, this,
        CreateFlags.SoftwareVertexProcessing, presentParams );
    if ( mesh != null ) mesh.Dispose(); //free the old mesh if any
    mesh = Mesh.FromFile( myMeshFile, MeshFlags.SystemMemory, device );
    if ( texture != null ) texture.Dispose(); //free the old texture if any
    texture = new Texture( device, myTexture, 0, Pool.Managed );
    device.SetTexture( 0, texture );
    Material myMaterial = new Material();
    myMaterial.Diffuse = myMaterial.Ambient = Color.White;
    device.Material = myMaterial;
    //turn on some ambient light that scatters and lights the object evenly
    device.RenderState.Ambient = Color.White;
    //set up the transformation of world coordinates into camera or view space
    device.Transform.View = Matrix.LookAtLH(
        new Vector3( 0f, 0f, -5f ), //eye point 5.0 in front of the canvas
        new Vector3( 0f, 0f, 0f ), //camera looks at point 0,0,0
        new Vector3( 0f, 1f, 0f ) ); //world's up direction is the y-axis
    //set up the projection transformation using 4 parameters:
    //1.: field of view = 45 degrees; 2.: aspect ratio = height/width=1 = square window;
    //3.: near clipping distance = 0; 4.: far clipping distance = 10;
    device.Transform.Projection = Matrix.PerspectiveFovLH( (float)Math.PI/4, 1f, 1f, 100f );
    device.RenderState.CullMode = Cull.None;
    device.RenderState.Lighting = true;
    xAngle = yAngle = zAngle = 0f; //start angles
    myTimer.Start(); //start the timer again
  }
  catch ( DirectXException ) { MessageBox.Show("Could not initialize DirectX3D."); return; }
}

protected static void OnTimer( Object myObject, EventArgs myEventArgs )
{ if ( device == null ) return;
  //throw the old image away
  device.Clear(ClearFlags.Target | ClearFlags.ZBuffer, Color.Gray, 1f, 0);
  //rotate with 3 angular velocities
  device.Transform.World = Matrix.RotationYawPitchRoll( yAngle += 0.02f,
                                                         xAngle += 0.02f,
                                                         zAngle += 0.02f );

  device.BeginScene();
    mesh.DrawSubset( 0 );
  device.EndScene();
  device.Present();
}

```

```

void MenuFileRead( object obj, EventArgs ea )
{
    OpenFileDialog dlg = new OpenFileDialog();
    dlg.InitialDirectory = @"C:\DXSDK\Samples\Media";
    if ( (MenuItem)obj == miReadMesh ) dlg.Filter = "meshes (*.x)|*.x| All files (*.*)|*.*";
    else
        dlg.Filter = "Bitmaps (*.bmp)|*.bmp| All files (*.*)|*.*";
    if ( dlg.ShowDialog() != DialogResult.OK ) return;
    if ( (MenuItem)obj == miReadMesh ) myMeshFile = dlg.FileName;
    else myTexture = (Bitmap)Image.FromFile( dlg.FileName );
    OnResize(null); //Initialize everything
}

void MenuFileExit( object obj, EventArgs ea )
{
    Application.Exit();
}
}

```

Click Debug → Start Without Debugging Ctrl F5.

## Exercises

1. Load the texture C:\DXSDK\Samples\Media\Earth\earth.bmp. The tiger now wears the earth with Siberia on his chest and Australia on his backside. Try out other arbitrary images how they look as fur of a tiger. Finally give him back its proper texture: C:\DXSDK\Samples\Media\Tiger\tiger.bmp.

2. Load the mesh C:\DXSDK\Samples\Media\Misc\bigship1.x. Step back with the eye-point from ( 0f, 0f, -4f ) to ( 0f, 0f, -40f ). Cover the spaceship with another skin.

3. Try out the images Madonna.bmp and Butterfly.bmp as textures.

4. Copy C:\DXSDK\Samples\Media\Tiny\tiny.x and C:\DXSDK\Samples\Media\Tiny\Tiny\_skin.bmp in the directory C:\temp\mesh\_tiger1. Its a very big mesh and not a tiny one. In order to see all of it, step back the camera by changing the first vector of device.Transform.View = Matrix.LookAtLH from new Vector3( 0f, 0f, -5f ) to new Vector3( 0f, 0f, 580f ).

Furthermore You have to move the Far-Clipping-Plane far away: Change the last parameter of Matrix.PerspectiveFovLH((float)Math.PI/4, 1f, 1f, 100f ); from 100f to 10000f.

**(Caution:** After these changes You'll not see the tiger anymore, he is now simply too small. If You want to see the tiger again, You have to rescind the changes..)

5. Read the [Comments](#) and try to understand the command lines.

6. Open the file tiger.x with Notepad=Editor or Textpad and read the file content. Look for the section where the vertices are defined in form of 3 x-, y-, z-coordinates . Look for the section where the triangles are defined as apiece 3 indices of 3 vertices.

7. Read the lecture: [www.miszalok.de/Lectures/L06\\_3DVector/3D\\_Mesh/3DMesh\\_deutsch.htm](http://www.miszalok.de/Lectures/L06_3DVector/3D_Mesh/3DMesh_deutsch.htm)

8. Read the introduction into the x-file format: [www.xbdev.net/3dformats/x/xfileformat.php](http://www.xbdev.net/3dformats/x/xfileformat.php).

9. You can find explanations and comments to this exercise here:

<http://msdn.microsoft.com/library/default.asp>.

In the tree on the left side branch to: Win32 and COM Development → Graphics and Multimedia → DirectX → SDK Documentation → DirectX SDK Managed → DirectX SDK → Introducing DirectX 9.0 → Direct3D Graphics → Getting Started with Direct3D → Direct3D Tutorials → Tutorial 6: Using Meshes.

**Caution:** Mozilla Firefox doesn't show the tree on the left side. Recommended browser for MSDN: Internet Explorer.