

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

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

Copyright © by V. Miszalok, last update: 24-05-2006

- ↓ [Projekt mesh\\_file1](#)
- ↓ [Form1, OnResize, OnTimer](#)
- ↓ [Weitere Aufgaben](#)

### Projekt mesh\_file1

Diese Übungsaufgabe ist eine kurze, übersichtliche Fassung eines Direct3D-Tutorials von Microsoft: Tutorial6. Sie finden es unter C:\DXSDK\Samples\Managed\Direct3D\Tutorials\Tutorial6.

Main Menu nach dem Start von VS 2005: File -> New Project... -> Templates: Windows Application

Name: mesh\_file1 -> Location: C:\temp -> Create directory for solution: ausschalten -> OK

Löschen Sie die Files Program.cs und Form1.Designer.cs und den Inhalt von Form1.cs, wie es in den Kapiteln 2DCisC1 bis 2DCisC4 beschrieben wurde.

Falls das Solution Explorer - Fenster nicht schon offen ist, öffnen Sie es über das Hauptmenü: View -> Solution Explorer.

Im Solution Explorer - Fenster klicken Sie auf das Pluszeichen vor mesh\_file1. Es öffnet sich ein Baum. Ein Ast heißt "References". Klicken Sie mit der **rechten** Maustaste auf References und dann mit der **linken** Maustaste auf Add Reference... Es öffnet sich eine Add Reference Dialog Box. Scrollen Sie abwärts, bis Sie den Component Name: Microsoft.DirectX Version 1.0.2902.0 sehen.

Markieren Sie durch Linksklick diese Referenz und (bei gedrückter der Strg-Taste) die beiden weiter unten stehende Referenzen

Microsoft.DirectX.Direct3D Version 1.0.2902.0 und

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

Verlassen Sie die Add Reference Dialog Box mit OK.

Kontrollieren Sie, ob jetzt im Solution Explorer Fenster unter mesh\_file1 -> References (unter anderen) die drei Referenzen

Microsoft.DirectX und

Microsoft.DirectX.Direct3D und

Microsoft.DirectX.Direct3DX stehen.

**Diese Übung setzt voraus dass C:\DXSDK\Samples\Media\Tiger\tiger.x (29 KB) und C:\DXSDK\Samples\Media\Tiger\tiger.bmp (66 KB) unter diesem Pfad existieren. Falls diese Files bei Ihnen irgendwo anders stehen, müssen Sie im Kopf von Form1 die Strings myMeshFile und myTextureFile entsprechend ändern.**

### Form1, OnResize, OnTimer

Schreiben in das leere Codefenster Form1.cs folgenden Code:

```
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 ( DirectException ) { 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();
}
}

```

Klicken Sie Debug -> Start Without Debugging Ctrl F5.

## Weitere Aufgaben

1. Laden Sie die Textur C:\DXSDK\Samples\Media\Earth\earth.bmp. Der Tiger hat nun die Erde als Fell mit Sibirien an der Brust und Australien am Hintern. Erproben Sie beliebige andere Bilder als Texturen. Geben Sie dem Tiger sein richtiges Fell zurück: C:\DXSDK\Samples\Media\Tiger\tiger.bmp.
2. Laden Sie das Mesh C:\DXSDK\Samples\Media\Misc\bigship1.x. Treten Sie mit dem Augenpunkt zurück von ( 0f, 0f, -4f ) nach ( 0f, 0f, -40f ). Geben Sie dem Raumschiff eine andere Haut als ein Tigerfell.
3. Erproben Sie die Bilder Madonna.bmp und Butterfly.bmp als Texturen.
4. Kopieren Sie auch C:\DXSDK\Samples\Media\Tiny\tiny.x und C:\DXSDK\Samples\Media\Tiny\Tiny\_skin.bmp in die Directory C:\temp\mesh\_tiger1. Anders als der Name suggeriert, ist das ein sehr großes Mesh. Um es zu sehen, müssen Sie mit der Kamera weit zurück: Ändern Sie deshalb in device.Transform.View = Matrix.LookAtLH den ersten Vektor von new Vector3( 0f, 0f, -5f ) auf new Vector3( 0f, 0f, 580f ). Außerdem müssen Sie die Far-Clipping-Plane weit wegschieben: Ändern Sie den letzten Parameter von Matrix.PerspectiveFovLH((float)Math.PI/4, 1f, 1f, 100f ); von 100f auf 10000f.  
**(Vorsicht: mit diesen beiden Einstellungen sehen Sie den Tiger nicht mehr, er ist jetzt schlicht zu klein. Wenn Sie den Tiger wieder sehen wollen, müssen Sie die beiden Änderungen rückgängig machen.)**
5. Lesen Sie die Kommentare und versuchen Sie, den Sinn der Befehle zu verstehen.
6. Öffnen Sie das File tiger.x mit Notepad=Editor oder Textpad und lesen Sie den Fileinhalt. Suchen Sie den Abschnitt, wo die Vertices als 3 x-, y-, z-Koordinaten definiert sind. Suchen Sie den Abschnitt wo die Dreiecksflächen als je 3 Indizes von Vertices definiert sind.
7. Lesen Sie die Vorlesung: [www.miszalok.de/Lectures/L06\\_3DVector/3D\\_Mesh/3DMesh\\_deutsch.htm](http://www.miszalok.de/Lectures/L06_3DVector/3D_Mesh/3DMesh_deutsch.htm)
8. Lesen Sie die Einführung in das x-File Format: [www.xbdev.net/3dformats/x/xfileformat.php](http://www.xbdev.net/3dformats/x/xfileformat.php).
9. Sie finden Erklärungen und Kommentare zu dieser Übung unter <http://msdn.microsoft.com/library/default.asp>.  
Verzweigen Sie in dem Baum auf der linken Seite auf: 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.  
**Vorsicht:** Mozilla Firefox zeigt den Baum auf der linken Seite nicht an.  
Empfohlener Browser für MSDN: Internet Explorer.