Course 3D_MDX: 3D-Graphics with Managed DirectX 9.0
Chapter 6: Mesh Viewer

Project mesh_viewer1

Main Menu after starting VS 2005: `File → New Project... → Templates: Windows Application Name: mesh_viewer1 → 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_viewer1`. 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`.

If You want to use DirectX Libs = References older than `Version 1.0.2902.0` You'll obtain 3 error messages. The solution is simple: You have to delete the first parameter of the function calls in these lines:

```
mesh0 = Mesh.Clean( CleanType.Optimization, mesh0, adjacency, adjacency );
myfont.DrawText( null, "This mesh has " ...
myfont.DrawText( null, "divided into " ...
```

i.e. You have to delete the parameters `CleanType.Optimization,` `null,` and `null,` and the program will run.

This chapter requires that `C:\DXSDK\Samples\Media\Tiger\tiger.x` and `C:\DXSDK\Samples\Media\Tiger\tiger.bmp` and other 13 mesh- and texture-files are located in specific paths.. In case these files are somewhere else, You have to change the string `static String s = @"C:\DXSDK\Samples\";` in the head of `Form1`.

Complete program

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;
  static float xAngle, yAngle, zAngle;
  static Mesh mesh0, mesh1;
  static ExtendedMaterial[] materials;
  static Microsoft.DirectX.Direct3D.Font myfont;
  struct m{ public String title; public String mesh; public String texture; public Int32 eyedistance;
              public m(String a, String b, String c, Int32 d)  //Constructor
                     { title = a; mesh = b; texture = c; eyedistance = d; }
           };
  static String s = @"C:\DXSDK\Samples\";
  static m[] meshes =
  { new m("tiger"  ,s+@"Media\Tiger\tiger.x"             ,s+@"Media\Tiger\tiger.bmp",   5),
    new m("bigship",s+@"Media\misc\bigship1.x"          ,s+@"Media\Tiger\tiger.bmp",  50),
    new m("knot"   ,s+@"Media\misc\knot.x"              ,s+@"Media\Tiger\tiger.bmp",   5),
    new m("shapes" ,s+@"Media\misc\shapes1.x"           ,s+@"Media\Earth\earth.bmp",  50),
    new m("scull"  ,s+@"Media\misc\skullocc.x"          ,s+@"Media\Tiger\tiger.bmp",  50),
    new m("shark"  ,s+@"Media\Prt Demo\LandShark.x"     ,s+@"Media\Tiger\tiger.bmp",1000),
    new m("car"    ,s+@"C++\Direct3D\EffectParam\car2.x",s+@"C++\Direct3D\EffectParam\EffectParam.jpg",2000),
    new m("tiny"   ,s+@"Media\Tiny\tiny.x"              ,s+@"Media\Tiger\tiger.bmp",2000),
    new m("dwarf"  ,s+@"Media\Dwarf\dwarf.x"            ,s+@"Media\Tiger\tiger.bmp",  10),
    new m("airplan",s+@"Media\Airplane\airplane 2.x"     ,s+@"Media\Airplane\bihull.bmp",50),
    new m("headsad",s+@"Media\Prt Demo\Head_Sad.x"      ,s+@"Media\Tiger\tiger.bmp",1000),
    new m("virus"  ,s+@"C++\Direct3D\EffectParam\cytovirus.x",s+@"Media\Tiger\tiger.bmp",2000)
  };
  String myMeshFile    =  meshes[0].mesh;
  String myTextureFile =  meshes[0].texture;
  Bitmap         myBitmap  = null;
  BaseTexture    myTexture = null;
  GraphicsStream adjacency = null;
  GroupBox group      = new GroupBox();
  RadioButton[] radio = new RadioButton[meshes.Length];
  TrackBar[] track    = new TrackBar[2];
  Label[] label       = new Label[2];
  TextBox[] text      = new TextBox[2];
  CheckBox check      = new CheckBox();
  Panel panel         = new Panel();
  Timer myTimer       = new Timer();

  public Form1()
  { Text = "Mesh Viewer";
    for ( int i = 0; i < track.Length; i++ )
    { label[i] = new Label();    Controls.Add( label[i] );
      track[i] = new TrackBar(); Controls.Add( track[i] );
      text [i] = new TextBox();  Controls.Add( text [i] );
      label[i].BackColor = track[i].BackColor = Color.Gray;
      track[i].Minimum   = 1;
      track[i].TickStyle = TickStyle.None;
      label[i].TextAlign = ContentAlignment.MiddleCenter;
      text [i].TextAlign = HorizontalAlignment.Center;
    }
    Controls.Add( group );
    for ( int i = 0; i < meshes.Length; i++ )
    { radio[i] = new RadioButton(); Controls.Add( radio[i] );
      radio[i].Parent = group;
      radio[i].Text = meshes[i].title;
      radio[i].Location = new Point( 5, Convert.ToInt32((0.6 + i*1.2)*FontHeight) );
      radio[i].Size = new Size( 60, Convert.ToInt32(1.2*Font.Height) );
      radio[i].TextAlign = ContentAlignment.MiddleCenter;
      radio[i].CheckedChanged += new EventHandler( radio_changed );
    }
    label[0].Text = "Reduce Vertices"; label[1].Text = "Eye Distance";
    Controls.Add( check );
    Controls.Add( panel );
    track[0].MouseUp        += new MouseEventHandler( track0_MouseUp );
    track[1].MouseUp        += new MouseEventHandler( track1_MouseUp );
    check   .CheckedChanged += new EventHandler( check_changed );
    myTimer.Tick            += new EventHandler( OnTimer );
    myTimer.Interval = 1;
    myBitmap = (Bitmap)Image.FromFile( meshes[0].texture );
    track[1].Value = track[1].Maximum = meshes[0].eyedistance;
    ClientSize = new Size( 1024, 800 ); //calls OnResize( ... )
  }
```

```
protected override void OnResize( System.EventArgs e )
//Whenever the window changes we have to initialize Direct3D from scratch
{ myTimer.Stop();// stop the timer during initialization
  for ( int i = 0; i < track.Length; i++ )
    label[i].Width = track[i].Width = text[i].Width = ClientSize.Width / 10;
  text[0].Text = "vertices = " + track[0].Value.ToString();
  text[1].Text = "eye = - "    + track[1].Value.ToString();
  check  .Text = "Wire Frame";
  group.Size = new Size( ClientSize.Width / 10, meshes.Length*radio[0].Height + 6 );
  check.Size = new Size( ClientSize.Width / 10, radio[0].Height );
  panel.Size = new Size( ClientSize.Width - label[0].Width - 2, ClientSize.Height - 2 );
  group   .Location = new Point( 2, 1 );
  label[0].Location = new Point( 2, group   .Location.Y + group   .Height + 20 );
  track[0].Location = new Point( 2, label[0].Location.Y + label[0].Height +  1 );
  text [0].Location = new Point( 2, track[0].Location.Y + track[0].Height +  2 );
  label[1].Location = new Point( 2, text[ 0].Location.Y + text [0].Height + 20 );
  track[1].Location = new Point( 2, label[1].Location.Y + label[1].Height +  1 );
  text [1].Location = new Point( 2, track[1].Location.Y + track[1].Height +  2 );
  check   .Location = new Point( 2, text [1].Location.Y + text [1].Height + 20 );
  panel   .Location = new Point( 2 + group.Width + 2, group.Location.Y );
  try
  { PresentParameters presentParams = new PresentParameters();
    presentParams.Windowed = true;
    presentParams.SwapEffect = SwapEffect.Discard;
    presentParams.EnableAutoDepthStencil = true;
    presentParams.AutoDepthStencilFormat = DepthFormat.D16;
    if ( device != null ) device.Dispose();
    device = new Device( 0, DeviceType.Hardware, panel,
                         CreateFlags.MixedVertexProcessing, presentParams );
    //turn on some white directional light from left to right
    device.Lights[0].Type = LightType.Directional;
    device.Lights[0].Diffuse = Color.White;
    device.Lights[0].Direction = new Vector3( 1, 0, 0 );
    device.Lights[0].Enabled = true;
    Material myMaterial = new Material();
    myMaterial.Diffuse = myMaterial.Ambient = Color.White;
    device.Material = myMaterial;
    device.RenderState.Ambient = Color.FromArgb( 0x00303030 );
    device.Transform.Projection = Matrix.PerspectiveFovLH( (float)Math.PI/4, 1f, 1f, 5000f );
    device.Transform.View = Matrix.LookAtLH(
      new Vector3( 0f, 0f, -track[1].Value ),
      new Vector3( 0f,0f,0f ),
      new Vector3( 0f,1f,0f ) );
    device.RenderState.Lighting = true;
    if ( check.Checked ) device.RenderState.FillMode = FillMode.WireFrame;
    else                 device.RenderState.FillMode = FillMode.Solid;
    myfont = new Microsoft.DirectX.Direct3D.Font(
             device, new System.Drawing.Font( "Arial", 12, FontStyle.Bold ) );
    SetUpMesh();
    SetUpTexture();
    myTimer.Start();
  }
  catch (DirectXException) { MessageBox.Show("Could not initialize Direct3D."); return; }
}

private void SetUpMesh()
{ Cursor.Current = Cursors.WaitCursor; //change mouse pointer to hour glass
  if ( mesh0 != null ) mesh0 .Dispose(); //free the old mesh if any
  mesh0 = Mesh.FromFile( myMeshFile, MeshFlags.Managed, device, out adjacency, out materials );
  mesh0 = Mesh.Clean( CleanType.Optimization, mesh0, adjacency, adjacency );
  if ( mesh1  != null ) mesh1.Dispose(); //free the old mesh if any
  //make a copy mesh1 from mesh0
  mesh1 = mesh0.Clone( mesh0.Options.Value, mesh0.VertexFormat | VertexFormats.Normal, device );
  //if mesh0 has no normals, compute them within mesh1 and copy them back to mesh0
  if ( (mesh0.VertexFormat & VertexFormats.Normal) == 0 )
  { mesh1.ComputeNormals();
    mesh0.Dispose();
    mesh0 = mesh1.Clone( mesh1.Options.Value, mesh1.VertexFormat, device );
  }
  track[0].Value = track[0].Maximum = mesh0.NumberVertices;
  text[0].Text = "vertices = " + track[0].Value.ToString();
  Cursor.Current = Cursors.Arrow; //change mouse pointer back to normal arrow
}
```

```
  private void SetUpTexture()
  { if ( myTexture != null ) myTexture.Dispose(); //free the old texture if any
    myTexture = new Texture( device, myBitmap, 0, Pool.Managed );
    device.SetTexture( 0, myTexture );
  }

  protected static void OnTimer( Object myObject, EventArgs myEventArgs )
  { device.Clear(ClearFlags.Target | ClearFlags.ZBuffer, Color.Gray, 1f, 0);
    device.Transform.World = Matrix.RotationYawPitchRoll( yAngle += 0.02f,
                                                          xAngle += 0.02f,
                                                          zAngle += 0.02f );
    device.BeginScene();
      for ( int i=0; i < materials.Length; i++ ) mesh1.DrawSubset( i );
      myfont.DrawText( null, "This mesh has " + mesh1.NumberVertices.ToString() + " vertices",
                       new Rectangle( 0,  0, 100, 20 ), DrawTextFormat.NoClip, Color.Red );
      myfont.DrawText( null, "divided into " + materials.Length.ToString() + " subsets",
                       new Rectangle( 0, 20, 100, 20 ), DrawTextFormat.NoClip, Color.Red );
    device.EndScene();
    device.Present(); //show the canvas
  }

  private void radio_changed( Object sender, EventArgs e )
  { RadioButton radio = (RadioButton)sender;
    Int32 i;
    for ( i = 0; i < meshes.Length; i++ )
      if ( meshes[i].title == radio.Text ) break;
    if ( myMeshFile != meshes[i].mesh )
    { myMeshFile = meshes[i].mesh;
      SetUpMesh();
    }
    if ( myTextureFile != meshes[i].texture )
    { myBitmap = (Bitmap)Image.FromFile( myTextureFile = meshes[i].texture );
      SetUpTexture();
    }
    track[1].Value = track[1].Maximum = meshes[i].eyedistance;
    text [1].Text = "eye = - " + track[1].Value.ToString();
    device.Transform.View = Matrix.LookAtLH(
      new Vector3( 0f, 0f, -meshes[i].eyedistance ),
      new Vector3( 0f,0f,0f ),
      new Vector3( 0f,1f,0f ) );
  }

  private void track0_MouseUp(object sender, System.EventArgs e) //Reduce no. of vertices
  { if ( materials.Length > 1 )
    { MessageBox.Show( "This mesh has more than one subset. It can't be simplified !" );
      track[0].Value = track[0].Maximum;
      return;
    }
    Cursor.Current = Cursors.WaitCursor;
    if ( mesh1 != null ) mesh1.Dispose();
    try { mesh1 = Mesh.Simplify( mesh0, adjacency, null, track[0].Value, MeshFlags.SimplifyVertex ); }
    catch { mesh1 = mesh0.Clone( mesh0.Options.Value, mesh0.VertexFormat, device );
            MessageBox.Show( "This mesh cannot be simplified !" );
          }
    track[0].Value = mesh1.NumberVertices;
    text[0].Text = "vertices = " + mesh1.NumberVertices.ToString();
    Cursor.Current = Cursors.Default;
  }

  private void track1_MouseUp(object sender, System.EventArgs e)//Eye Distance
  { device.Transform.View = Matrix.LookAtLH(
        new Vector3( 0f, 0f, -track[1].Value ),   //eye point 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
    text[1].Text = "eye = - " + track[1].Value.ToString();
  }

  private void check_changed( Object sender, EventArgs e ) //Wire Frame on/off
  { if ( check.Checked ) device.RenderState.FillMode = FillMode.WireFrame;
    else                 device.RenderState.FillMode = FillMode.Solid;
  }
}
```

Click `Debug` → `Start Without Debugging Ctrl F5`.

If You used DirectX Libs = References older than `Version 1.0.2902.0` You obtain 3 error messages. The solution is simple: You have to delete the first parameter of the function calls in these lines:

```
mesh0 = Mesh.Clean( CleanType.Optimization, mesh0, adjacency, adjacency );
myfont.DrawText( null, "This mesh has " ...
myfont.DrawText( null, "divided into " ...
```

i.e. You have to delete the parameters `CleanType.Optimization,` `null,` and `null,` and the program will run.

Exercises

1. Add another trackbar which shifts the x-axis rotation increment between 0 and 0.02.
2. Add another two trackbars which shift the y-axis and z-axis rotation increment between 0 and 0.02.
3. Add fourth trackbar which shifts the eye point in z-direction from -2000 to +2000.
4. Reduce automatically the no. of vertices, when the eye point moves away and the level of details can be reduced (LOD based tesselation). See **OpenGL & Direct3D Pipeline**