

# Course 3D\_XNA: 3D-Computer Graphics with XNA

## Chapter C3: Drunken Tiger

Copyright © by V. Miszalok, last update: 10-01-2010

- ↓ [Project TigerRot1](#)
- ↓ [Version 1: Minimum](#)
- ↓ [Version 2: In a Quadratic Resizable Window](#)
- ↓ [Version 3: Rotate the Tiger Around the Y-Axis](#)
- ↓ [Version 4: Translation and Scaling](#)
- ↓ [Version 5: Changing the Mesh](#)
- ↓ [Version 6: The Complete Code](#)
- ↓ [Version 7: Space Ship](#)

This chapter is a subcompact summary of the first XNA-Tutorial from Microsoft.

You find the tutorial here: VS 2008 → Main Menu Help → Contents → XNA Game Studio 3.0 → Getting Started with XNA Game Studio → Going Beyond: XNA Game Studio in 3D → Tutorial 1: Displaying a 3D Model On The Screen.

## Project TigerRot1

1. Main Menu after starting VS 2008: File → New Project... → Project types: XNA GameStudio 3.1 → Templates: Windows Game (3.1) → Name: TigerRot1 → Location: C:\temp → Create directory for solution: switch it off → OK. Solution Explorer - TigerRot1: Delete the file Program.cs and the default code of Game1.cs.

2. If You find no Solution Explorer-window, open it via the main menu: View → Solution Explorer. Inside the Solution Explorer-window click the plus-sign in front of TigerRot1. A tree opens. Look for the branch "References". Check if there are (among others) the references: **Microsoft.XNA.Framework** and **Microsoft.XNA.Framework.Game** and **mscorlib** and **System**.

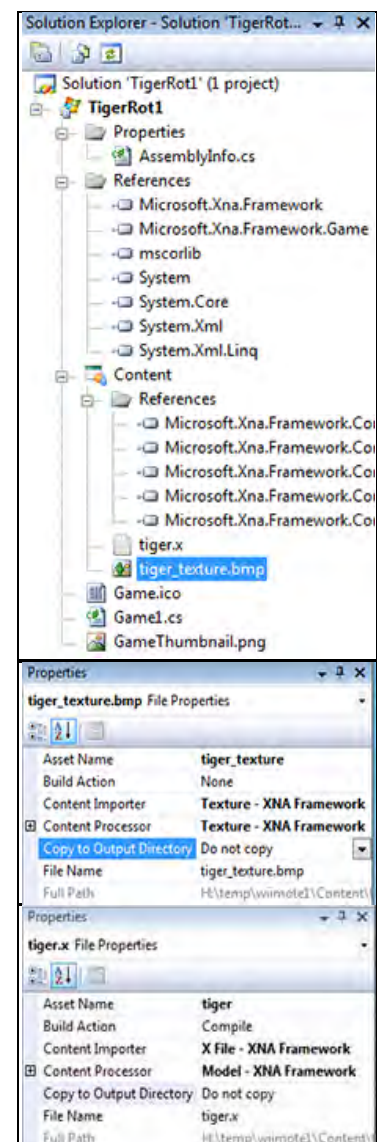
3. Right click this link: [tiger\\_texture.bmp](#) and store the texture into the project directory C:\temp\TigerRot1\Content.

Right click this link: [tiger.x](#) and store the mesh into the project directory C:\temp\TigerRot1\Content.

3.1 We have to add the texture image and the mesh file to project TigerRot1: Solution Explorer → Right click the branch Content → Add → Existing Item... → Directory: Content: All Files (\*.\*). Select both [tiger\\_texture.bmp](#) and [tiger.x](#) and quit by clicking the Add-button and check whether both file names arrived underneath the Content-branch.

3.2 Solution Explorer → Click [tiger\\_texture.bmp](#) and in its Properties-window change the Build Action-property from Compile to None.

Check if all properties of [tiger\\_texture.bmp](#) and [tiger.x](#) correspond to the screenshots on the right.



## Version 1: Minimum

Write the following code into the empty code window of `Game1.cs`:

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

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

public class Game1 : Microsoft.Xna.Framework.Game
{ private GraphicsDeviceManager g;
  private Model model;
  private BasicEffect effect;
  public Game1()
  { g = new GraphicsDeviceManager( this );
  }
  protected override void LoadContent()
  { model = Content.Load< Model >("Content\\tiger");
    effect = (BasicEffect)model.Meshes[0].Effects[0];
    effect.View = Matrix.CreateLookAt( new Vector3( 0f, 0f, 4f ), //camera position
                                     Vector3.Zero, Vector3.Up );
    effect.Projection = Matrix.CreatePerspectiveFieldOfView
      ( MathHelper.Pi/4, 1f, 1f, 100000f );
  }
  protected override void Draw( GameTime gameTime )
  { model.Meshes[0].Draw();
  }
} // end of class Game1
} // end of class Program
```

Click Debug → Start Without Debugging Ctrl F5.

Experiments	with camera position
initial	<code>new Vector3( 0f, 0f, 4f )</code>
nearer	<code>new Vector3( 0f, 0f, 3f )</code>
more distant	<code>new Vector3( 0f, 0f, 6f )</code>
to the rear	<code>new Vector3( 0f, 0f, -4f )</code>
to the right	<code>new Vector3( 3f, 0f, 4f )</code>
to the left	<code>new Vector3( -3f, 0f, 4f )</code>
upwards	<code>new Vector3( 0f, 3f, 4f )</code>
downwards	<code>new Vector3( 0f, -3f, 4f )</code>

## Version 2: In a Quadratic Resizable Window

Insert the following event handler below the constructor:

```
public Game1() { g = new GraphicsDeviceManager( this ); }

protected override void Initialize()
{ g.PreferredBackBufferWidth = 600;
  g.PreferredBackBufferHeight = 600;
  g.ApplyChanges();
  Window.Title = "Tiger Rotation";
  Window.AllowUserResizing = true;
  base.Initialize();
}
```

Click Debug → Start Without Debugging Ctrl F5.

## Version 3: Rotate the Tiger Around the Y-Axis

Insert the following global variable definitions below the constructor: `private BasicEffect effect;`

```
private float trans      = 0,
            transIncrement = 0.02f,
            zoom         = 1,
            zoomIncrement = 0.005f,
            rotation     = 0,
            rotationIncrement = 0.01f;
```

Change the protected override void Draw( gameTime gameTime ) event handler until it looks like this:

```
protected override void Draw( gameTime gameTime )
{ //g.GraphicsDevice.RenderState.FillMode = FillMode.WireFrame;
  g.GraphicsDevice.Clear( Color.DarkBlue );
  effect.World = Matrix.CreateRotationY ( modelRotation += 0.01f );
  model.Meshes[0].Draw();
}
```

Click Debug → Start Without Debugging Ctrl F5.

Experiments	with the camera	Results
1.	In the effect.World-line replace the letter "Y" in CreateRotationY to "X".	looping around the X-axis
2.	In the effect.World-line replace the letter "Y" in CreateRotationY to "Z"	looping around the Z-axis
3.	In the effect.World-line replace "0.01f" by "0.05f".	5 times faster ≈ 3°/Draw
4.	In the effect.World-line replace "0.01f" by "0.001f".	10 times slower ≈ 0.06°/Draw
5.	In the effect.Projection-line replace "MathHelper.Pi/4" by "MathHelper.Pi/6".	camera angle shrinks from 45° to 30°
6.	In the effect.Projection-line replace "MathHelper.Pi/4" by "MathHelper.Pi/2".	camera obtains a wide angle of 90°
7.	In the effect.Projection-line replace the 2. parameter "1f" by "0.5f".	distorted image: width = 2*height
8.	In the effect.Projection-line replace the 2. parameter "1f" by "2f".	distorted image: height = 2*width
9.	In the effect.Projection-line replace the 3. parameter "1f" by "3f".	Head an tail closer than 3f are lost.
Experiments	with the GraphicsDeviceManager g	
1.	Remove the comment slashes "//" in front of the first line of the new Draw-function.	triangles without skin
2.	Insert comment slashes "//" in front of the second line of the new Draw-function.	rotational body

## Version 4: Translation and Scaling

Replace protected override void Draw( gameTime gameTime ) { ... } by:

```
protected override void Draw( gameTime gameTime )
{ //g.GraphicsDevice.RenderState.FillMode = FillMode.WireFrame;
  g.GraphicsDevice.Clear( Color.DarkBlue );
  if ( trans < -0.8f | trans > 0.8f ) transIncrement *= -1f;
  if ( zoom < 0.1f | zoom > 1.2f ) zoomIncrement *= -1f;
  trans += transIncrement;
  zoom += zoomIncrement;
  rotation += rotationIncrement;
  effect.World = Matrix.CreateScale ( zoom );
  effect.World *= Matrix.CreateRotationX ( rotation );
  effect.World *= Matrix.CreateRotationY ( rotation );
  effect.World *= Matrix.CreateRotationZ ( rotation );
  effect.World *= Matrix.CreateTranslation( trans, 0, 0 );
  model.Meshes[0].Draw();
}
```

There are two faster alternatives to code the cascading movements by `effect.World`-lines:

```

/*1*/effect.World = Matrix.CreateScale      ( zoom      ) *
                    Matrix.CreateRotationX ( rotation ) *
                    Matrix.CreateRotationY ( rotation ) *
                    Matrix.CreateRotationZ ( rotation ) *
                    Matrix.CreateTranslation( trans, 0, 0 );
/*2*/effect.World = Matrix.CreateScale      ( zoom      ) *
                    Matrix.CreateFromYawPitchRoll( rotation, rotation, rotation ) *
                    Matrix.CreateTranslation ( 0, trans, 0 );

```

Experiments:

1. Change the values of `transIncrement`, `zoomIncrement` and `rotationIncrement`.
2. Replace `(trans, 0, 0)` by `(0, trans, 0)`.
3. Remove single `Matrix.xxx`-lines by comment-slashes `///  
//`.

## Version 5: Changing the Mesh

In the Solution Explorer `TigerRot1`-window open the `Content`-branch. → Double click `tiger.x`. →

A text file `tiger.x` opens. It contains 3 important parts:

1. 303 lines defining 303 3D-vertices = Vertex Buffer
2. 599 lines defining 599 triangles = Index Buffer and
3. 303 lines defining 303 2D-coordinates = Texture Coordinates Buffer.

This file has been produced by a diligent external designer using 3D modeling software such as [Autodesk 3ds Max](#) or [Autodesk Maya](#) (complete list of [3D modeling programs](#)). He exported his result to this DirectX format file `tiger.x`.

Let us edit `tiger.x` in order to see how x-files work.

**1. Experiment:** Comment out the last vertex (vertex no. 303) of the Vertex Buffer:

```
-0.213423;-0.066057;0.311063;; and write a new line: -0.6;-0.066057;0.6;;
```

The end of the vertex Buffer should look like this:

```
//-0.213423;-0.066057;0.311063;;
-0.6;-0.066057;0.6;;
```

Store `tiger.x` and run the program. The tiger's right hind leg looks strange with a big tapered tumor.

**2. Experiment:** Comment out the first 5 lines (no. of triangles plus the triangle numbers 1, 2, 3, 4) of the Index Buffer and adjust the no. of triangles until these lines look like this.

```
// 599;
//3;300,301,302;,
//3;299,300,302;,
//3;298,300,299;,
//3;295,296,297;,
595;
3;294,295,297;, .....
```

Store `tiger.x` and run the program.

The tiger's right hind leg looks even stranger with a big defect in and near the tumor.

## Version 6: The Complete code

```

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

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

public class Game1 : Microsoft.Xna.Framework.Game
{ private GraphicsDeviceManager g;
  private Model model;
  private BasicEffect effect;
  private float trans          = 0,
                transIncrement = 0.02f,
                zoom           = 1,
                zoomIncrement  = 0.005f,
                rotation       = 0,
                rotationIncrement = 0.01f;

```

```

public Game1()
{ g = new GraphicsDeviceManager( this );
}
protected override void Initialize()
{ g.PreferredBackBufferWidth = 600;
  g.PreferredBackBufferHeight = 600;
  g.ApplyChanges();
  Window.Title = "Tiger Rotation";
  Window.AllowUserResizing = true;
  base.Initialize();
}
protected override void LoadContent()
{ model = Content.Load< Model >("Content\\tiger");
  effect = (BasicEffect)model.Meshes[0].Effects[0];
  effect.View = Matrix.CreateLookAt( new Vector3( 0f, 0f, 4f ),
                                     Vector3.Zero, Vector3.Up );
  effect.Projection = Matrix.CreatePerspectiveFieldOfView
    ( MathHelper.Pi/4, 1f, 1f, 100000f );
}
protected override void Draw( gameTime )
{ //g.GraphicsDevice.RenderState.FillMode = FillMode.WireFrame;
  g.GraphicsDevice.Clear( Color.DarkBlue );
  if ( trans < -0.8f | trans > 0.8f ) transIncrement *= -1f;
  if ( zoom < 0.1f | zoom > 1.2f ) zoomIncrement *= -1f;
  trans += transIncrement;
  zoom += zoomIncrement;
  rotation += rotationIncrement;
  effect.World = Matrix.CreateScale ( zoom );
  effect.World *= Matrix.CreateRotationX ( rotation );
  effect.World *= Matrix.CreateRotationY ( rotation );
  effect.World *= Matrix.CreateRotationZ ( rotation );
  effect.World *= Matrix.CreateTranslation( trans, 0, 0 );
  model.Meshes[0].Draw();
}
} // end of class Game1
} // end of class Program

```

## Version 7: Space Ship

Let us replace the tiger by another 3D-model.

1. Right click this link: [ship\\_texture.tga](#) and store the texture into the project directory C:\temp\TigerRot1\Content.

2. Right click this link: [ship.fbx](#) and store the mesh into the project directory C:\temp\TigerRot1\Content.

3. We have to add the texture image and the mesh file to project TigerRot1:  
 Solution Explorer → Right click the branch Content → Add → Existing Item... →  
 Directory: Content: All Files (\*.\*)  
 Select both [ship\\_texture.tga](#) and [ship.fbx](#), and quit by clicking the Add-button and check whether both file names arrived underneath the Content-branch.

4. Solution Explorer → Click [ship\\_texture.tga](#) and in its Properties-window and change the Build Action-property from Compile to None.

5. We have to call the ship-model instead of the tiger by changing the Content.Load-line to:  
 model = Content.Load< Model >("Content\\ship");

6. The ship is much bigger than the tiger. Thus we have to retract the camera  
 from Matrix.CreateLookAt( new Vector3( 0f, 0f, 4f ), Vector3.Zero, Vector3.Up )  
 to Matrix.CreateLookAt( new Vector3( 0f, 0f, 4000f ), Vector3.Zero, Vector3.Up ).

7. Click Debug → Start Without Debugging Ctrl F5.