

Course 3D_XNA: 3D-Computer Graphics with XNA

Chapter C4: XBox 360 Controller

Copyright © by V. Miszalok, last update: 11-08-2008

- ↓ [XBox 360 Controller](#)
- ↓ [Project Controller1](#)
- ↓ [The complete Code of Game1.cs](#)
- ↓ [The complete Code of Form1.cs](#)

XBox 360 Controller



There is just one generic gamepad support by XNA: the wired USB Xbox 360 Controller from Microsoft. Price: ca. 30 €
The wired version carrying an USB-cable works fine with both a PC and an Xbox 360 console. Price: ca. 30 €

Plug it into an USB-port and check it: Start → Control Panel → Game Controllers → XBOX 360 For Windows (Controller) → Properties → Test.

This chapter is an extended version of the second 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 2: Making Your Model Move Using Input.

Project controller1

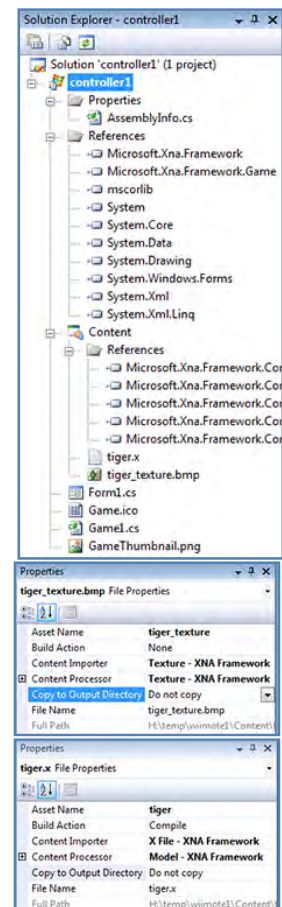
1. Main Menu after starting VS 2008: File → New Project... → Project types: XNA GameStudio 3.0 → Templates: Windows Game (3.0) → Name: controller1 → Location: C:\temp → Create directory for solution: **switch it off** → OK.
Solution Explorer - controller1:
Delete the file Program.cs and the default code of Game1.cs.

2. Right click this link: [tiger_texture.bmp](#) and store the texture into the project directory C:\temp\controller1\Content.
Right click this link: [tiger.x](#) and store the mesh into the project directory C:\temp\controller1\Content.

2.1 We have to add the texture image and the mesh file to project controller1:
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-References-branch.

2.2 Click the [tiger_texture.bmp](#) branch. 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.

3. Create a new class file: controller1 → Add → Windows Form... → Name: Form1.cs → Add.
Click the plus-sign in front of the Form1.cs-branch. Delete the file Form1.Designer.cs underneath Form1.cs and the default code of Form1.cs.



The complete code of Game1.cs

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

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

static class Program
{ 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 positionX = 0.0f, positionY = 0.0f, positionZ = 0.0f;
  private float scaleX = 1.0f, scaleY = 1.0f, scaleZ = 1.0f;
  private float rotationX = 0.0f, rotationY = 0.0f, rotationZ = 0.0f;
  public Form1 form;
  public Game1() { g = new GraphicsDeviceManager( this ); }

  protected override void Initialize()
  { g.PreferredBackBufferWidth = 600;
    g.PreferredBackBufferHeight = 600;
    g.ApplyChanges();
    g.IsFullScreen = false;
    Window.AllowUserResizing = true;
    Window.Title = "Moving the Tiger via an Xbox 360 Controller";
    base.Initialize();
    form = new Form1();
    form.Location = new System.Drawing.Point(Window.ClientBounds.Right+5, Window.ClientBounds.Top);
    form.Size = new System.Drawing.Size ( 100, Window.ClientBounds.Height );
    form.Show();
  }
  protected override void LoadContent()
  { Content.RootDirectory = "Content";
    model = Content.Load<Model>( "tiger" );
    effect = (BasicEffect)model.Meshes[0].Effects[0];
    effect.View = Matrix.CreateLookAt( new Vector3(0.0f, 0.0f, 4.0f), Vector3.Zero, Vector3.Up );
    effect.Projection = Matrix.CreatePerspectiveFieldOfView( MathHelper.Pi/4, 1f, 0.1f, 1000.0f );
  }

  protected override void Update( GameTime gameTime )
  { GamePadState s = GamePad.GetState( 0 );
    if ( !s.IsConnected )
      { System.Windows.Forms.MessageBox.Show( "Can't find a Xbox 360 Controller" ); Exit(); }
    ButtonState bp = ButtonState.Pressed;
    if ( s.Buttons.A != bp) form.checkbox[ 0].Checked = false;
    else { scaleY *= 0.99f; form.checkbox[ 0].Checked = true; }
    if ( s.Buttons.B != bp) form.checkbox[ 1].Checked = false;
    else { scaleX *= 1.01f; form.checkbox[ 1].Checked = true; }
    if ( s.Buttons.X != bp) form.checkbox[ 2].Checked = false;
    else { scaleX *= 0.99f; form.checkbox[ 2].Checked = true; }
    if ( s.Buttons.Y != bp) form.checkbox[ 3].Checked = false;
    else { scaleY *= 1.01f; form.checkbox[ 3].Checked = true; }
    if ( s.Buttons.LeftShoulder != bp) form.checkbox[ 4].Checked = false;
    else { scaleZ *= 0.99f; form.checkbox[ 4].Checked = true; }
    if ( s.Buttons.RightShoulder != bp) form.checkbox[ 5].Checked = false;
    else { scaleZ *= 1.01f; form.checkbox[ 5].Checked = true; }
    if ( s.Buttons.Start != bp) form.checkbox[ 6].Checked = false;
    else { reset(); form.checkbox[ 6].Checked = true; }
    if ( s.Buttons.Back != bp) form.checkbox[ 7].Checked = false;
    else { reset(); form.checkbox[ 7].Checked = true; }
    if ( s.Buttons.LeftStick != bp) form.checkbox[ 8].Checked = false;
    else { form.checkbox[ 8].Checked = true; }
    if ( s.Buttons.RightStick != bp) form.checkbox[ 9].Checked = false;
    else { form.checkbox[ 9].Checked = true; }
  }
}
```

```

        if (s.DPad.Left           != bp) form.checkbox[10].Checked = false;
        else { positionX -= 0.01f;    form.checkbox[10].Checked = true; }
        if (s.DPad.Right          != bp) form.checkbox[11].Checked = false;
        else { positionX += 0.01f;    form.checkbox[11].Checked = true; }
        if (s.DPad.Up             != bp) form.checkbox[12].Checked = false;
        else { positionY += 0.01f;    form.checkbox[12].Checked = true; }
        if (s.DPad.Down           != bp) form.checkbox[13].Checked = false;
        else { positionY -= 0.01f;    form.checkbox[13].Checked = true; }
        Vector2 ts1 = s.ThumbSticks.Left;
        Vector2 ts2 = s.ThumbSticks.Right;
        rotationZ -= 0.1f * ts1.X;    form.trackbar[0].Value=(int)( 50f*(ts1.X +1f) );
        rotationX -= 0.1f * ts1.Y;    form.trackbar[1].Value=(int)( 50f*(ts1.Y +1f) );
        rotationZ -= 0.1f * ts2.X;    form.trackbar[2].Value=(int)( 50f*(ts2.X +1f) );
        rotationY -= 0.1f * ts2.Y;    form.trackbar[3].Value=(int)( 50f*(ts2.Y +1f) );
        float tr1 = s.Triggers.Left ; form.trackbar[4].Value=(int)( 100f*tr1 );
        GamePad.SetVibration(0,tr1,0f);
        float tr2 = s.Triggers.Right; form.trackbar[5].Value=(int)( 100f*tr2 );
        GamePad.SetVibration(0,0f,tr2);
        base.Update( gameTime );
    }

    private void reset()
    { positionX = positionY = positionZ = rotationX = rotationY = rotationZ = 0f;
      scaleX = scaleY = scaleZ = 1f;
    }

    protected override void Draw( gameTime )
    { g.GraphicsDevice.Clear( Color.DarkBlue );
      effect.EnableDefaultLighting();
      effect.World = Matrix.CreateScale( scaleX, scaleY, scaleZ ) *
                    Matrix.CreateRotationX( rotationX ) *
                    Matrix.CreateRotationY( rotationY ) *
                    Matrix.CreateRotationZ( rotationZ ) *
                    Matrix.CreateTranslation( positionX, positionY, positionZ );
      model.Meshes[0].Draw();
    }
} // end of class Game1
} // end of class Program

```

The complete code of Form1.cs

Replace the existing lines of Form1.cs by the following code:

```

using System;
using System.Drawing;
using System.Windows.Forms;

public class Form1 : System.Windows.Forms.Form
{ public const Int32 nCheckBoxes = 14, nTrackBars = 6;
  public CheckBox[] checkbox = new CheckBox[nCheckBoxes];
  public TrackBar[] trackbar = new TrackBar[nTrackBars];
  public Label [] label = new Label[nTrackBars];

```

```

public Form1()
{
    BackColor = Color.White;
    Text = "Xbox 360 Controller Buttons";
    Int32 i;
    for ( i=0; i < nCheckBoxes; i++ )
    {
        checkbox[i] = new CheckBox(); Controls.Add( checkbox[i] );
        checkbox[i].TextAlign = ContentAlignment.MiddleCenter;
    }
    checkbox[ 0].Text = "A";
    checkbox[ 1].Text = "B";
    checkbox[ 2].Text = "X";
    checkbox[ 3].Text = "Y";
    checkbox[ 4].Text = "Left Shoulder";
    checkbox[ 5].Text = "Right Shoulder";
    checkbox[ 6].Text = "Start";
    checkbox[ 7].Text = "Back";
    checkbox[ 8].Text = "Left Stick";
    checkbox[ 9].Text = "Right Stick";
    checkbox[10].Text = "Left";
    checkbox[11].Text = "Right";
    checkbox[12].Text = "Up";
    checkbox[13].Text = "Down";
    for ( i=0; i < nTrackBars; i++ )
    {
        trackbar[i] = new TrackBar(); Controls.Add( trackbar[i] );
        label [i] = new Label(); Controls.Add( label[i] );
        trackbar[i].AutoSize = false;
        trackbar[i].TickStyle = TickStyle.None;
        trackbar[i].Minimum = 0;
        trackbar[i].Maximum = 100;
        label [i].TextAlign = ContentAlignment.TopCenter;
    }
    label[0].Text = "X-Axis 1";
    label[1].Text = "Y-Axis 1";
    label[2].Text = "X-Axis 2";
    label[3].Text = "Y-Axis 2";
    label[4].Text = "Left Trigger";
    label[5].Text = "Right Trigger";
    foreach ( Control c in Controls ) c.BackColor = Color.Gray;
    StartPosition = FormStartPosition.Manual;
}

protected override void OnResize( EventArgs e )
{
    Int32 w = ClientRectangle.Width;
    Int32 h = ClientRectangle.Height / Controls.Count;
    Int32 i, top = 1;
    for ( i=0; i < Controls.Count; i++ )
    {
        Controls[i].Top = top;
        Controls[i].Left = 2;
        Controls[i].Width = w;
        Controls[i].Height = h - 2;
        top += h;
    }
    for ( i=0; i < nTrackBars; i++ ) trackbar[i].Height = h;
}
}

```

Click Debug → Start Without Debugging Ctrl F5.

Remarks: I couldn't detect any "sticks" on the XBox 360 Controller which switch the `s.Buttons.LeftStick-` and `s.Buttons.RightStick-` properties of the `GamePadButtons-` structure of the `Microsoft.XNA.Framework.Input-`library.

These properties seem to be inaccessible via the normal XBox 360 Controller and reserved for future use. This is the reason why the `left Stick-` and `Right Stick-` check boxes of program controller1 are probably useless.