

Course 3D_XNA: 3D-Computer Graphics with XNA

Chapter C4: NoName GamePad

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

- ↓ [NoName-GamePad](#)
- ↓ [Project gamepad1](#)
- ↓ [The complete Code of Game1.cs](#)
- ↓ [The complete Code of Form1.cs](#)

NoName GamePads



There is just one generic gamepad support by XNA: the wired USB Xbox 360 Controller from Microsoft. Price: ca. 30 €. A talented hacker "SoopahMan" from Germany wrote a `Soopah.XNA.Input.dll` based on `DirectX.DirectInput` that allows the use of nearly any other USB-GamePad. Plug it into an USB-Port and check it: `Start → Control Panel → Game Controllers → Generic USB Joystick → Properties → Test`. If more than one controller is listed, the following project `gamepad1` will just recognize the first of them. Problem: The controllers can be very different: arrangement of buttons, no. of analog thumb sticks. Our program arranges the buttons in a technical, not in a logical order.

Project gamepad1

1. Main Menu after starting VS 2008: `File → New Project... → Project types: XNA GameStudio 3.0 Templates: Windows Game (3.0) → Name: gamepad1 → 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`.

2.1 If You find no Solution Explorer-window, open it via the main menu: `View → Solution Explorer Ctrl+W, S`.

Inside the Solution Explorer-window click + in front of `gamepad1`. A tree opens. Look for the branch "References". Click the + in front of References. Right click References → `Add Reference → .NET → scroll down until you find Microsoft.DirectX.DirectInput Version 1.0.2902.0 → select it → OK`. Check if the reference: `Microsoft.DirectX.DirectInput` is listed among the default references as `Microsoft.XNA.Framework` etc.

2.2 Download SoopahMans component lib [Soopah.XNA.Input.dll](#) and store it to `C:\temp\gamepad1` and check whether it arrived.

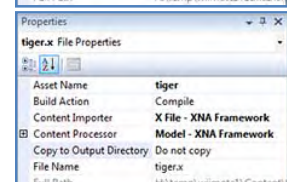
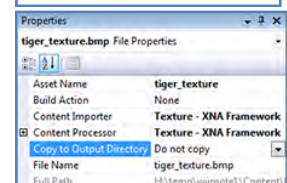
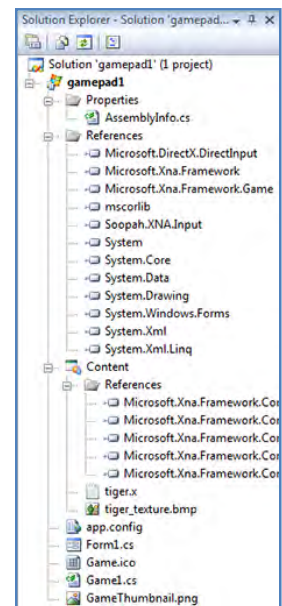
Alternate source: <http://sourceforge.net/projects/xnadirectinput>

2.3 Solution Explorer → right click main branch References → Add a reference to `Soopah.XNA.Input.dll` using the Browse tab of the Add Reference-Dialog Box.

2.4 Change a debug switch of VS 2008: Main Menu of VS 2008: `Debug → Exceptions... → Managed Debugging Assistants → LoaderLock → uncheck the first check box → OK`.

3. Right click this link: [tiger_texture.bmp](#) and store the texture into the project directory `C:\temp\gamepad1\Content`.

Right click this link: [tiger.x](#) and store the mesh into the project directory `C:\temp\gamepad1\Content` and check whether both arrived.



3.1 We have to add the texture image and the mesh file to project `gamepad1`:

Solution Explorer → right click Content → Add → Existing Item... → 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 Change the Build Action-property of `tiger_texture.bmp` from Compile to None.

Check if all properties of `tiger_texture.bmp` and `tiger.x` correspond to the screenshots on the right.

4. Create a new class file: Solution Explorer → `gamepad1` → Add → Windows Form... → Name: `Form1.cs` → Add.

Click the plus-sign in front of branch `Form1.cs`. Delete the file `Form1.Designer.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 ButtonState[] bs = new ButtonState[12]; //8 Buttons + 4 DPads
  private Vector2      [] ts = new Vector2[3];    //3 Thumbsticks with x/y
  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 = "XNA Using NoName GamePads";
    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 )
{
    if ( Soopah.Xna.Input.DirectInputGamepad.Gamepads.Count < 1 )
    {
        System.Windows.Forms.MessageBox.Show( "Error: Can't find a GamePad" ); Exit(); }
    for ( int i=0; i < 8; i++ ) //check all possible 8 Buttons, 4 DPads, 3 ThumbSticks
        bs[i] = Soopah.Xna.Input.DirectInputGamepad.Gamepads[0].Buttons.List[i];
    bs [ 8] = Soopah.Xna.Input.DirectInputGamepad.Gamepads[0].DPad.Down;
    bs [ 9] = Soopah.Xna.Input.DirectInputGamepad.Gamepads[0].DPad.Up;
    bs [10] = Soopah.Xna.Input.DirectInputGamepad.Gamepads[0].DPad.Left;
    bs [11] = Soopah.Xna.Input.DirectInputGamepad.Gamepads[0].DPad.Right;
    ts [ 0] = Soopah.Xna.Input.DirectInputGamepad.Gamepads[0].ThumbSticks.Left;
    ts [ 1] = Soopah.Xna.Input.DirectInputGamepad.Gamepads[0].ThumbSticks.Right;
    ts [ 2] = Soopah.Xna.Input.DirectInputGamepad.Gamepads[0].ThumbSticks.Third;
    ButtonState bp = ButtonState.Pressed;
    for ( int i=0; i < 12; i++ )
        if ( bs[i] == bp ) form.checkbox[i].Checked = true;
        else form.checkbox[i].Checked = false;
    if ( bs[ 0] == bp ) scaleY *= 1.01f; //Button No. 1
    if ( bs[ 1] == bp ) scaleX *= 1.01f; //Button No. 2
    if ( bs[ 2] == bp ) scaleY *= 0.99f; //Button No. 3
    if ( bs[ 3] == bp ) scaleX *= 0.99f; //Button No. 4
    if ( bs[ 4] == bp ) scaleZ *= 0.99f; //Button No. 5
    if ( bs[ 5] == bp ) scaleZ *= 1.01f; //Button No. 6
    if ( bs[ 6] == bp ) rotationZ += 0.01f; //Button No. 7
    if ( bs[ 7] == bp ) rotationZ -= 0.01f; //Button No. 8
    if ( bs[ 8] == bp ) positionY -= 0.01f; //DPad.Down
    if ( bs[ 9] == bp ) positionY += 0.01f; //DPad.Up
    if ( bs[10] == bp ) positionX -= 0.01f; //DPad.Left
    if ( bs[11] == bp ) positionX += 0.01f; //DPad.Right
    for ( int i=0, j=0; i < 3; i++, j+=2 ) //3 ThumbSticks mapped on 6 TrackBars
    {
        if ( Math.Abs(ts[i].X) < 0.2f ) ts[i].X = 0f; //suppress small movements
        if ( Math.Abs(ts[i].Y) < 0.2f ) ts[i].Y = 0f; //suppress small movements
        form.trackbar[j ].Value = (int)( 50f * (ts[i].X+1f) ); //map to range 0..100
        form.trackbar[j+1].Value = (int)( 50f * (ts[i].Y+1f) ); //map to range 0..100
    }
    rotationX -= ts[0].X / 50f;
    rotationY -= ts[1].Y / 50f;
    rotationZ -= ts[2].X / 50f;
    base.Update( gameTime );
}

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 = 12, nTrackBars = 6;
    public CheckBox[] checkbox = new CheckBox[nCheckBoxes];
    public TrackBar[] trackbar = new TrackBar[nTrackBars];
    Label [] label = new Label[nTrackBars];
}

```

```

public Form1()
{ BackColor = Color.White;
  Text      = "GamePad 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 = "1";
  checkbox[ 1].Text = "2";
  checkbox[ 2].Text = "3";
  checkbox[ 3].Text = "4";
  checkbox[ 4].Text = "5";
  checkbox[ 5].Text = "6";
  checkbox[ 6].Text = "7";
  checkbox[ 7].Text = "8";
  checkbox[ 8].Text = "Down";
  checkbox[ 9].Text = "Up";
  checkbox[10].Text = "Left";
  checkbox[11].Text = "Right";
  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 = "X-Axis 3";
  label[5].Text = "Y-Axis 3";
  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 Debugging F5.

If an exception occurs in `game.Run()`; check the Error List.

Probably there will be this warning:

Found conflicts between different versions of the same dependent assembly.

Double click the warning. A Microsoft Visual Studio message box appears and asks:

Do you want to fix these conflicts by adding binding redirect records in the `app.config` file ?.

Quit by pressing Yes.

An `app.config` item appears within the Solution Explorer-branch `gamepad1` and solves the problem.