

# DirectX 9 mit Visual Basic.NET – DirectInput

## Vorwort

Nachdem mir ([HIER ZAHL EINFÜGEN] xD) 2 mal OpenOffice abgeschmiert ist, versuche ich nun das 4. Tutorial fertig zu schreiben...

Also, das Tutorial handelt von DirectInput wie es auch im Titel schon steht.

Warum sollte ich DirectInput benutzen? die Standardfunktionen sind doch auch ganz nett! Ganz einfach, DirectInput ermöglicht es z.B. mehrere Tastendrucke zu verarbeiten.

Im Grunde wollen wir das Dreieck was wir im letzten Tutorial zum Rotieren gebracht haben, heute per Benutzereingaben Rotieren lassen... wat'n Satz^^

## Verweis Festlegen

In diesem Tutorial brauchen wir einen Verweis auf die DirectInput Bibliothek:

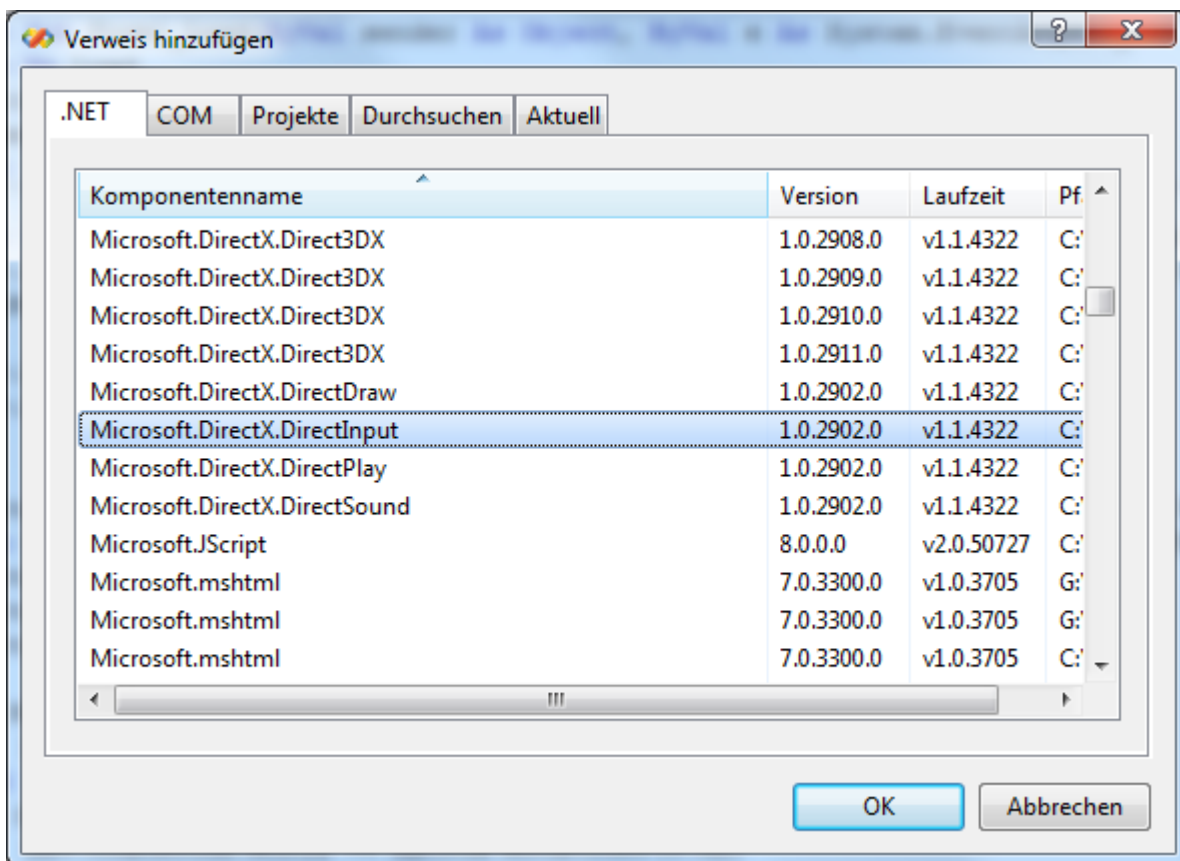


Bild 1: Verweis auf DirectInput festlegen

# Kapitel 1 – Den Code Aktualisieren

Der Code aus dem Letzten Tut:

Code 1:

```
Public Class Form1
    Public Dev As Device
    Public Parameter As PresentParameters
    Dim Camera As Camera
    Private VertexBuf As VertexBuffer

    Public Sub initialisieren()
        Parameter = New PresentParameters
        Parameter.Windowed = True
        Parameter.SwapEffect = SwapEffect.Discard
        Parameter.EnableAutoDepthStencil = True
        Parameter.AutoDepthStencilFormat = DepthFormat.D16
        Dev = New Device(0, DeviceType.Hardware, Me.Handle, _
            CreateFlags.SoftwareVertexProcessing, Parameter)
        VertexBuf = New VertexBuffer(GetType(CustomVertex.PositionColored), 3, _
            Dev, Usage.None, _
            CustomVertex.PositionColored.Format, Pool.Default)
        Camera = New Camera(Dev, Me, 1, 50, _
            New Vector3(0, 0, 5), _
            New Vector3(0, 0, 0), _
            New Vector3(0, 1, 0))
    End Sub

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) _
        Handles Me.Load
        initialisieren()
        Timer1.Start()
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) _
        Handles Timer1.Tick
        Dev.Clear(ClearFlags.Target Or ClearFlags.ZBuffer, Color.CornflowerBlue, 1, 0)

        Dev.RenderState.Lighting = False
        Dev.RenderState.CullMode = Cull.Clockwise & Cull.CounterClockwise

        Dim vertex As CustomVertex.PositionColored() = _
            CType(VertexBuf.Lock(0, LockFlags.None), CustomVertex.PositionColored())

        vertex(0) = New CustomVertex.PositionColored(0, 1, 0, Color.Blue.ToArgb)
        vertex(1) = New CustomVertex.PositionColored(1, -1, 0, Color.Red.ToArgb)
        vertex(2) = New CustomVertex.PositionColored(-1, -1, 0, Color.Green.ToArgb)
        Dev.Transform.World *= Matrix.RotationZ(0.02) * Matrix.RotationY(0.02) * _
            Matrix.RotationX(0.02)

        VertexBuf.Unlock()

        Dev.SetStreamSource(0, VertexBuf, 0)
        Dev.VertexFormat = CustomVertex.PositionColored.Format

        Dev.BeginScene()
        Dev.DrawPrimitives(PrimitiveType.TriangleList, 0, 1)
        Dev.EndScene()
        Dev.Present()
    End Sub
End Class
```

Nun erstellen wir 2 neue Klassenvariablen:

Code 2:

```
Public KeyBoard As DirectInput.Device  
Public KeyBoardState As DirectInput.KeyboardState
```

### **Erklärung**

KeyBoard	=	Erstellt ein DirectInput Device z.B. Tastatur oder Maus
KeyBoardState	=	Ruft den aktuellen zustand der Tastatur ab, z.B. welche Tasten gerade gedrückt werden.

## Kapitel 2 – Keyboard Initialisieren

Damit wir die Tastatur auch verwenden können müssen wir diese erst Initialisieren, dazu schreiben wir einfach diesen Code ganz unten in die Initialisierungs Sub:

Code 3:

```
KeyBoard = New DirectInput.Device(DirectInput.SystemGuid.Keyboard)
KeyBoard.SetCooperativeLevel(Me, DirectInput.CooperativeLevelFlags.Background Or _
    DirectInput.CooperativeLevelFlags.NonExclusive)

KeyBoard.Acquire()
```

### Erklärung

New DirectInput.Device(...) = Setzt das KeyBoard Device auf die Standardtastatur die am Computer angeschlossen ist.

SetCooperativeLevel(...) = Setzt die Parameter des Keyboard Devices( Ziel Control, Flags)

## Kapitel 3 – Let's rotate... again

Nun wollen wir unser Dreieck Rotieren lassen, damit unsere Tasten regelmäßig auf Tastendruck überprüft werden schreiben wir in die Timer1 Sub folgenden Code in die erste Zeile:

Code 4:

```
KeyBoardState = KeyBoard.GetCurrentKeyboardState
```

und dann löschen wir folgende Zeile in der Timer1 Sub:

Code 5:

```
Dev.Transform.World *= Matrix.RotationZ(0.02) * Matrix.RotationY(0.02) * _  
Matrix.RotationX(0.02)
```

und schreiben in die gleiche Zeile folgenden Code:

Code 6:

```
If KeyBoardState(DirectInput.Key.Left) Then  
    Dev.Transform.World *= Matrix.RotationY(-0.02)  
End If  
If (KeyBoardState(DirectInput.Key.Right)) Then  
    Dev.Transform.World *= Matrix.RotationY(0.02)  
End If  
If (KeyBoardState(DirectInput.Key.Up)) Then  
    Dev.Transform.World *= Matrix.RotationX(0.02)  
End If  
If (KeyBoardState(DirectInput.Key.Down)) Then  
    Dev.Transform.World *= Matrix.RotationX(-0.02)  
End If
```

### Erklärung

Im Grunde ist es das selbe wie vorher, nur dass die einzelnen Tasten überprüft werden und je nach Tastendruck Rotiert.

# Der Komplette Code

## Code 7:

```
Imports Microsoft
Imports Microsoft.DirectX
Imports Microsoft.DirectX.Direct3D

Public Class Form1

    Public Dev As Device
    Public KeyBoard As DirectInput.Device
    Public KeyBoardState As DirectInput.KeyboardState
    Public Parameter As PresentParameters
    Dim Camera As Camera
    Private VertexBuf As VertexBuffer

    Public Sub initialsieren()

        Parameter = New PresentParameters
        Parameter.Windowed = True
        Parameter.SwapEffect = SwapEffect.Discard
        Parameter.EnableAutoDepthStencil = True
        Parameter.AutoDepthStencilFormat = DepthFormat.D16
        Dev = New Device(0, DeviceType.Hardware, Me.Handle, CreateFlags.SoftwareVertexProcessing,
Parameter)
        VertexBuf = New VertexBuffer(GetType(CustomVertex.PositionColored), 3, Dev, Usage.None,
CustomVertex.PositionColored.Format, Pool.Default)
        Camera = New Camera(Dev, Me, 1, 50, New Vector3(0, 0, 5), New Vector3(0, 0, 0), New
Vector3(0, 1, 0))

        KeyBoard = New DirectInput.Device(DirectInput.SystemGuid.Keyboard)
        KeyBoard.SetCooperativeLevel(Me, DirectInput.CooperativeLevelFlags.Background Or _
DirectInput.CooperativeLevelFlags.NonExclusive)

        KeyBoard.Acquire()

    End Sub
    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) _
Handles Me.Load
        initialsieren()
        Timer1.Start()
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Timer1.Tick
        Dev.Clear(ClearFlags.Target Or ClearFlags.ZBuffer, Color.CornflowerBlue, 1, 0)
        KeyBoardState = KeyBoard.GetCurrentKeyboardState

        Dev.RenderState.Lighting = False
        Dev.RenderState.CullMode = Cull.Clockwise & Cull.CounterClockwise

        Dim vertex As CustomVertex.PositionColored() = CType(VertexBuf.Lock(0, LockFlags.None),
CustomVertex.PositionColored())
        vertex(0) = New CustomVertex.PositionColored(0, 1, 0, Color.Blue.ToArgb)
        vertex(1) = New CustomVertex.PositionColored(1, -1, 0, Color.Red.ToArgb)
        vertex(2) = New CustomVertex.PositionColored(-1, -1, 0, Color.Green.ToArgb)

        If KeyBoardState(DirectInput.Key.Left) Then
            Dev.Transform.World *= Matrix.RotationY(-0.02)
        End If
        If (KeyBoardState(DirectInput.Key.Right)) Then
            Dev.Transform.World *= Matrix.RotationY(0.02)
        End If
        If (KeyBoardState(DirectInput.Key.Up)) Then
            Dev.Transform.World *= Matrix.RotationX(0.02)
        End If
        If (KeyBoardState(DirectInput.Key.Down)) Then
            Dev.Transform.World *= Matrix.RotationX(-0.02)
        End If

        VertexBuf.Unlock()
        Dev.SetStreamSource(0, VertexBuf, 0)
        Dev.VertexFormat = CustomVertex.PositionColored.Format
        Dev.BeginScene()
        Dev.DrawPrimitives(PrimitiveType.TriangleList, 0, 1)
        Dev.EndScene()
        Dev.Present()
    End Sub
```

## **Build it!**

Nun können wir per F5 Taste das Projekt starten.

Bei fragen könnt ihr mich auf meiner Seite erreichen

<http://devtronic.bplaced.net>