

استفاده از OpenGL در VB.NET

نویسنده: Jacek Artymiak

04/28/2003

اگرچه VB.NET اولین زبان برنامه نویسی نیست که وقتی شما به API های OpenGL می اندیشید به فکرتان بیاید اما در صورتی که شما مایل باشید که پنجه هایتان را در آبهای داغ Open Source غوطه ور کنید، اینکار امکان پذیر است. OpenGL در اصل بوسیله SGI توسعه داده شد و حالا انواع تجاری و مجانی آن در دسترس است. OpenGL یک API چند پلتفرمی استاندارد صنعت برای گرافیکهای سه بعدی پشتیبانی شده بوسیله اغلب توسعه دهندگان این صنعت است که بطور وسیعی برای گرافیکهای کامپیوتری، انیمیشن کامپیوتری، چندرسانه ای و بازیهای کامپیوتری استفاده می شود. در ویندوز 2000/XP شما باید یک کپی از آن در مسیر C:\WinNT\System\System32 با نام OPENG32.DLL داشته باشید. در این مقاله به شما نشان خواهیم داد که چگونه به سرعت برنامه های مبتنی بر OpenGL را در VB.NET با استفاده از استعداد Open Source بسازید.

ابزار مورد نیاز

من فرض می کنم که شما از VB.NET یا یک نسخه کامل از VS.NET و SDK مربوط به .NET Framework استفاده می کنید.

سپس شما به یک کپی از کتابخانه C# که توسط تیم توسعه Lioyd Dupont نوشته شده نیاز خواهید داشت. C# یک پیاده سازی دیگر از OpenGL نیست اما یک مجموعه از لایه های پوششی برای OpenGL DLL است که برنامه های VB.NET یا C# شما را بطور مناسبی به OpenGL مرتبط می کند. آن را برای برنامه نویسان C# در نظر گرفته اند اما به همان خوبی با VB.NET نیز کار می کند. این کتابخانه بر اساس لیسانس BSD است که به این معنی است که شما می توانید در صورت موافقت با گزینه های آن، از آن در برنامه های تجاری نیز استفاده کنید.

نصب C# خیلی آسان است. فقط کافی است توزیع باینری آن را دانلود کرده و با ابزاری مانند WinZip از حالت فشرده خارج کنید و اسکریپت نصب را که با C# می آید اجرا کنید. همین!

اولین پروژه شما

Visual Studio را اجرا کنید و از منوی File گزینه New\Project را انتخاب کنید. در پنجره پروژه جدید Visual Basic و همچنین Windows Application را انتخاب کنید. روی OK کلیک کنید. شما حالا باید فرم مستطیلی برنامه را ببینید. روی فرم کلیک راست کرده و گزینه View Code را انتخاب کنید. شما حالا باید این قطعه کد را ببینید:

```
Public Class Form1 Inherits System.Windows.Forms.Form
```

```
Windows Form Designer generated code
```

```
End Class
```

روی References در پنجره مرورگر Solution کلیک راست کرده و گزینه Add Reference را انتخاب کنید. در پنجره اضافه کردن رفرنس روی کلیک Browse کلیک کنید و محل آن را مشخص کنید. در همان محلی که OPENG32.DLL قرار دارد می توانید به دنبال آن بگردید. آن را انتخاب کنید و روی کلیک Open کلیک کنید. سپس کلیک OK را کلیک کنید. می بینید که ارجاعی به C# به پروژه شما اضافه شده است.

حالا اجازه دهید به کد برنامه برگردیم. روی علامت + بعد از Windows Form Designer Generated code کلیک کنید تا ساختار کد توسعه داده شود. چیزی که شما می بینید باید اینگونه باشد:

```
Public Class Form1
```

```
Inherits System.Windows.Forms.Form
```

```
#Region " Windows Form Designer generated code "
```

```
Public Sub New()
```

```
MyBase.New()
```

```
'This call is required by the Windows Form Designer.
```

```
InitializeComponent()
```

```
'Add any initialization after the
'InitializeComponent() call

End Sub

'Form overrides dispose to clean up the component list.
Protected Overrides Sub Dispose _
    (ByVal disposing As Boolean)
    If disposing Then
        If Not (components Is Nothing) Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(disposing)
End Sub

'Required by the Windows Form Designer
Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the
'Windows Form Designer It can be modified using the
'Windows Form Designer. Do not modify it using the
'code editor.
<System.Diagnostics.DebuggerStepThrough()>
Private Sub InitializeComponent()
    components = New System.ComponentModel.Container()
    Me.Text = "Form1"
End Sub

#End Region

End Class
```

شما حالا باید این ساختار را جهت کارهای زیر ویرایش کنید:

- وارد کردن CSDL
- تعریف دو صفت: View (viewport) و thrOpenGL (یک ریسمان OpenGL)

این کار بدین صورت انجام می شود:

```
Imports CSDL.OpenGL
Public Class Form1
    Inherits System.Windows.Forms.Form
    Private view As myOpenGL.myView
    Private thrOpenGL
```

سپس شما باید مقداردهی اولیه view و thrOpenGL را انجام دهید:

```
#Region " Windows Form Designer generated code "

Public Sub New()
    MyBase.New()

    'This call is required by the Windows Form Designer.
    InitializeComponent()

    'Add any initialization after the InitializeComponent() call

    Me.view = New myOpenGL.myView()
```

```
Me.view.Parent = Me
Me.view.Dock = DockStyle.Fill
Me.thrOpenGL = New Threading.Thread(AddressOf OpenGL_Start)
Me.thrOpenGL.Start()
```

```
End Sub
```

```
#End Region
```

وقتی thrOpenGL مقداردهی شد، یک ارجاع از متد OpenGL_Start را معین می کند که انجام عملیات بازسازی viewport را بر عهده دارد.

```
Private Sub OpenGL_Start()
    While 1 = 1
        Me.view.Refresh()
    End While
End Sub
```

یک بار ما شروع ریسمان OpenGL را تعیین کردیم، مرحله بعد اضافه کردن کدی است که آن را خاتمه می دهد. این کار یک نسخه تغییر یافته متد پیشفرض Dispose() است.

```
'Form overrides dispose to clean up the component list.
Protected Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
        If Not (components Is Nothing) Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(disposing)
    Me.thrOpenGL.Abort()
End Sub
```

همانطور که می بینید، همه آن چیزی که مورد نیاز بود، یک فراخوانی اضافی به Me.thrOpenGL.Abort() بود. سپس ما باید یک قطعه کد بزرگ را که کار واقعی را می کند، اضافه کنیم. در فضای نامی myOpenGL، کلاس myView را تعریف کنید که از CsGL.OpenGL.OpenGLControl به ارث می برد و تعدادی از متدهای آن را Override و بعضی از آنها را Overload می کنیم و یک گرداننده صفحه کلید را تعریف می کنیم. تعریف فضای نامی و کلاس، و ارث بری از OpenGLControl با استفاده از این خطوط کد انجام می شود:

```
Namespace myOpenGL
    Public Class myView
        Inherits OpenGLControl
```

مرحله بعد کمی غیر معمول است، زیرا ثابتهایی را که بطور معمول در فایلهای سرآیند gl.h و glu.h یافت می شوند، تعریف می کند. به بعضی دلایل من نتوانستم به آنها از همان روشی که برای متدها و کلاسهای CsGL استفاده میشود دسترسی پیدا کنم اما یک راه دیگر برای آن یافتم. من این دستورات را برای ساختن کدی که می تواند بصورت مستقیم در داخل بلاک Private Enum GLConstant...End Enum قرار داشته باشد، اجرا می کنم:

```
$ awk '/#define/ { print $2 " = &H" substr($3, 3) }' gl.h > glconst
$ awk '/#define/ { print $2 " = &H" substr($3, 3) }' glu.h > gluconst
```

در این مثال، لیست ثابتها مانند این است:

```
Private Enum GLConstants
    GL_COLOR_BUFFER_BIT = &H4000
```

```

GL_DEPTH_BUFFER_BIT = &H100
GL_SMOOTH = &H1D01
GL_DEPTH_TEST = &HB71
GL_LEQUAL = &H203
GL_PERSPECTIVE_CORRECTION_HINT = &HC50
GL_NICEST = &H1102
GL_PROJECTION = &H1701
GL_MODELVIEW = &H1701
GL_POLYGON = &H9
End Enum

```

سپس ما `glDraw()` را Override می کنیم. این جایی است که ما کدی که واقعا چیزی رسم می کند را اضافه می کنیم. توجه کنید که چگونه ثابتها بکار گرفته شده اند و چطور هنگام استفاده به `UInt32` تبدیل شده اند.

```

Public Overrides Sub glDraw()
    GL.glClearColor(0.0, 0.0, 0.0, 0.0)
    GL.glClear( _
        Convert.ToUInt32( _
            GLConstants.GL_COLOR_BUFFER_BIT Or _
            GLConstants.GL_DEPTH_BUFFER_BIT) )
    GL.glLoadIdentity()
    '
    ' Fun begins
    '
    GL.glColor3f(1.0, 1.0, 1.0)
    GL.glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0)
    GL.glBegin(Convert.ToUInt32(GLConstants.GL_POLYGON))
    GL.glVertex3f(0.25, 0.25, 0.0)
    GL.glVertex3f(0.75, 0.25, 0.0)
    GL.glVertex3f(0.75, 0.75, 0.0)
    GL.glVertex3f(0.25, 0.75, 0.0)
    GL.glEnd()
    GL.glFlush()
    '
End Sub

```

سپس `InitGLContext()` را Override می کنیم.

```

Protected Overrides Sub InitGLContext()
    GL.glShadeModel(Convert.ToUInt32(GLConstants.GL_SMOOTH))
    GL.glClearColor(0.0, 0.0, 0.0, 0.5)
    GL.glClearDepth(1.0)
    GL.glEnable(Convert.ToUInt32(GLConstants.GL_DEPTH_TEST))
    GL.glDepthFunc(Convert.ToUInt32(GLConstants.GL_LEQUAL))
    GL.glHint( _
        Convert.ToUInt32( _
            GLConstants.GL_PERSPECTIVE_CORRECTION_HINT), _
        Convert.ToUInt32(GLConstants.GL_NICEST))
End Sub

```

اگر می خواهیم بطور خودکار viewport پنجره OpenGL تغییر اندازه داده شود، باید `OnSizeChanged()` همپوشانی شود.

```

Protected Overloads Sub OnSizeChanged(ByVal e)
    Me.OnSizeChanged(e)
    Dim s As Size
    Dim aspect_ratio As Double
    s = Me.Size
    aspect_ratio = s.Width / s.Height
    GL.glMatrixMode(Convert.ToUInt32(GLConstants.GL_PROJECTION))

```

```

GL.glLoadIdentity()
GL.gluPerspective(45.0, aspect_ratio, 0.1, 100.0)
GL.glMatrixMode(Convert.ToInt32(GLConstants.GL_MODELVIEW))
GL.glLoadIdentity()
End Sub

```

سرانجام، ما باید گرداننده رویداد صفحه کلید را اضافه کنیم. این گرداننده رویداد فقط کلید ESC را بررسی می کند اما شما می توانید به سادگی آن را برای گرفتن کلیدهای دیگر نیز توسعه دهید.

```

Protected Sub myView_OnKeyDown(ByVal Sender As Object, _
    ByVal kea As System.Windows.Forms.KeyEventArgs) _
    Handles MyBase.KeyDown
    If (kea.KeyCode = Keys.Escape) Then
        Application.Exit()
    End If
End Sub

```

خوب است! اولین برنامه OpenGL شما با VB.NET اینچنین است:

```

Imports CsGL.OpenGL
Public Class Form1
    Inherits System.Windows.Forms.Form
    Private view As myOpenGL.myView
    Private thrOpenGL

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

        Me.view = New myOpenGL.myView()
        Me.view.Parent = Me
        Me.view.Dock = DockStyle.Fill
        Me.thrOpenGL = New Threading.Thread(AddressOf OpenGL_Start)
        Me.thrOpenGL.Start()

    End Sub

    Private Sub OpenGL_Start()
        While 1 = 1
            Me.view.Refresh()
        End While
    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overrides Sub Dispose(_
        ByVal disposing As Boolean)

        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
        Me.thrOpenGL.Abort()
    End Sub

```

```
'Required by the Windows Form Designer
Private components As _
    System.ComponentModel.IContainer

'NOTE: The following procedure is required
'by the Windows Form Designer
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.
<System.Diagnostics.DebuggerStepThrough()> _
Private Sub InitializeComponent()
    components = New System.ComponentModel.Container()
    Me.Text = "Form1"
End Sub

#End Region
End Class
```

```
Namespace myOpenGL
Public Class myView
    Inherits OpenGLControl
    Private Enum GLConstants
        GL_COLOR_BUFFER_BIT = &H4000
        GL_DEPTH_BUFFER_BIT = &H100
        GL_SMOOTH = &H1D01
        GL_DEPTH_TEST = &HB71
        GL_LEQUAL = &H203
        GL_PERSPECTIVE_CORRECTION_HINT = &HC50
        GL_NICEST = &H1102
        GL_PROJECTION = &H1701
        GL_MODELVIEW = &H1701
        GL_POLYGON = &H9
    End Enum

    Public Overrides Sub glDraw()
        GL.glClearColor(0.0, 0.0, 0.0, 0.0)
        GL.glClear( _
            Convert.ToUInt32( _
                GLConstants.GL_COLOR_BUFFER_BIT Or _
                GLConstants.GL_DEPTH_BUFFER_BIT))
        GL.glLoadIdentity()
        '
        ' Fun begins
        '
        GL.glColor3f(1.0, 1.0, 1.0)
        GL.glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0)
        GL.glBegin(Convert.ToUInt32(GLConstants.GL_POLYGON))
        GL.glVertex3f(0.25, 0.25, 0.0)
        GL.glVertex3f(0.75, 0.25, 0.0)
        GL.glVertex3f(0.75, 0.75, 0.0)
        GL.glVertex3f(0.25, 0.75, 0.0)
        GL.glEnd()
        GL.glFlush()
    End Sub

    Protected Overrides Sub InitGLContext()
        GL.glShadeModel(Convert.ToUInt32(GLConstants.GL_SMOOTH))
        GL.glClearColor(0.0, 0.0, 0.0, 0.5)
        GL.glClearDepth(1.0)
        GL.glEnable(Convert.ToUInt32(GLConstants.GL_DEPTH_TEST))
        GL.glDepthFunc(Convert.ToUInt32(GLConstants.GL_LEQUAL))
        GL.glHint( _
            Convert.ToUInt32( _
                GLConstants.GL_PERSPECTIVE_CORRECTION_HINT), _
```

```

        Convert.ToInt32(GLConstants.GL_NICEST))
End Sub

Protected Overloads Sub OnSizeChanged(ByVal e)
    Me.OnSizeChanged(e)
    Dim s As Size
    Dim aspect_ratio As Double
    s = Me.Size
    aspect_ratio = s.Width / s.Height
    GL.glMatrixMode(Convert.ToInt32(GLConstants.GL_PROJECTION))
    GL.glLoadIdentity()
    GL.gluPerspective(45.0, aspect_ratio, 0.1, 100.0)
    GL.glMatrixMode(Convert.ToInt32(GLConstants.GL_MODELVIEW))
    GL.glLoadIdentity()
End Sub

Protected Sub myView_OnKeyDown(ByVal Sender As Object, _
    ByVal kea As System.Windows.Forms.KeyEventArgs) _
    Handles MyBase.KeyDown
    If (kea.KeyCode = Keys.Escape) Then
        Application.Exit()
    End If
End Sub
End Class
End Namespace

```

حالا همه کاری که باید بکنید ساختن کد (Ctrl+Shift+B) و اجرای آن با کلید F5 است. چیزی که شما می بینید یک مستطیل سفید روی یک زمینه سیاه است که در کتاب Red Book شرح داده شده است. برای اطلاعات بیشتر درباره متدهای CSGL می توانید سورس آن را بررسی کنید که در سایت آن موجود است.