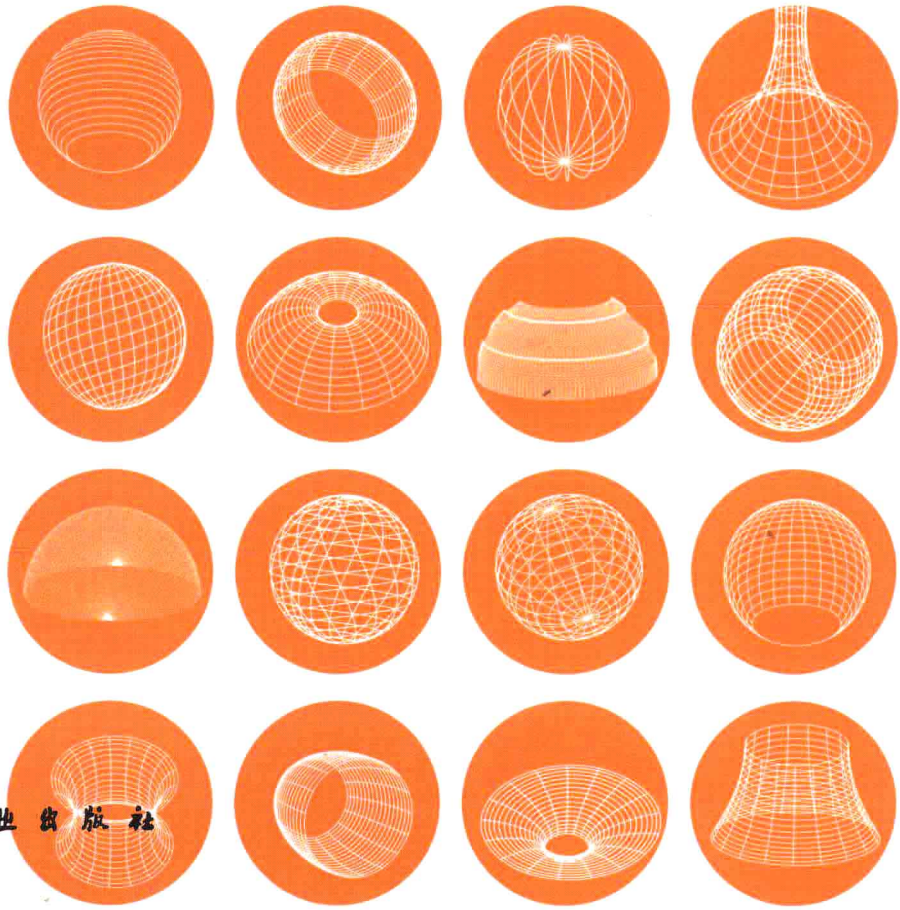
Visual Basic 2010 & OpenGL 的可视化设计

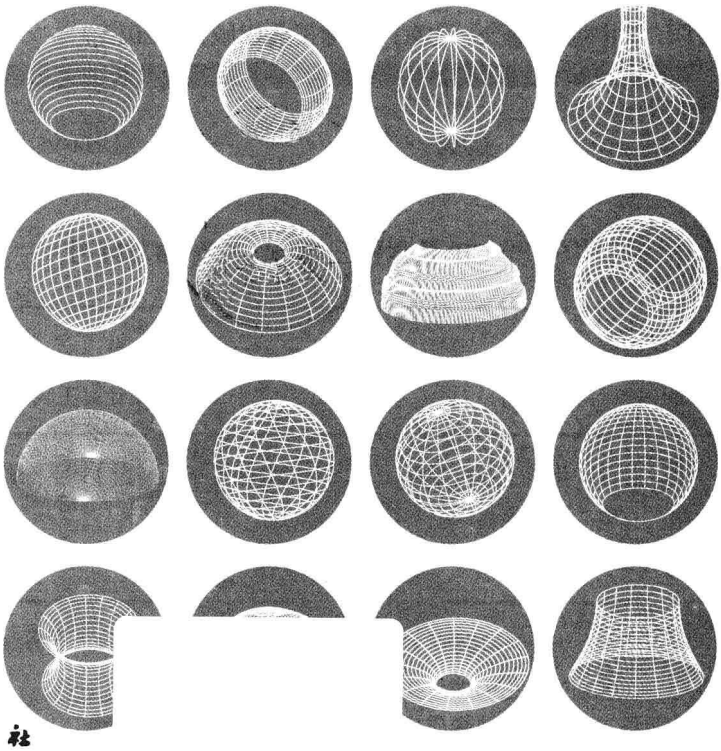
杨 亮　曹 阳　刘钟馨　编著



学 工 业 出 版 社

基于 Visual Basic 2010 & OpenGL 的可视化设计

杨 亮　曹 阳　刘钟馨　编著



化学工业出版社

·北京·

本书全面地介绍了利用Visual Basic 2010结合OpenGL函数库进行三维仿真等可视化设计的基本知识。内容包括Visual Basic 2010环境下使用OpenGL函数库的方法；利用OpenGL函数库进行三维开发的基础知识及函数使用方法；OpenGL程序开发的基本步骤及工作原理；颜色、光照及材质的基础知识；纹理的使用方法及融合技术；摄像机的使用方法及漫游、反馈和拾取的基本操作；贝塞尔曲线、曲面及Nurbs曲面的基础知识和实现方法；不同三维文件格式的加载方法。根据以上基础知识，进行了真空镀膜机三维仿真的演示及薄膜生长过程仿真研究。

本书通俗易懂，内容实用，实例丰富，可作为本科和高职院校进行实验仿真的专业教材，也可供利用OpenGL进行可视化设计的开发人员学习参考。

图书在版编目(CIP)数据

基于Visual Basic 2010&OpenGL的可视化设计 / 杨亮，曹阳，刘钟馨编著. —北京：化学工业出版社，2012.7

ISBN 978-7-122-14220-7

Ⅰ. ①基… Ⅱ. ①杨… ②曹… ③刘… Ⅲ. ①Basic语言-程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2012）第087524号

责任编辑：傅聪智　　文字编辑：云　雷

责任校对：宋　玮　　装帧设计：王晓宇

出版发行：化学工业出版社（北京市东城区青年湖南街13号　邮政编码100011）

印　　装：大厂聚鑫印刷有限责任公司

710mm×1000mm 1/16　印张13　彩插3　字数262千字　2012年10月北京第1版第1次印刷

购书咨询：010-64518888（传真：010-64519686）　售后服务：010-64518899

网　　址：http: // www. cip. com. cn

凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：49.00元

前　言

随着科学技术的不断进步，计算机硬件制造工艺及软件开发技术方面均蓬勃发展，计算机仿真技术及虚拟实现技术也取得长足进步，成为与实验科学和理论分析并列的认识客观世界的工具之一。计算机仿真技术在系统仿真过程和辅助决策方面几乎接近完美，但在图形输出方面尤其在三维仿真结果输出方面略显不足。目前，国内对系统仿真的原理、技术、理论及应用方面的著作、论文很多，但对仿真结果的输出方面的资料却相对较少。

本书主要针对非计算机专业教学科研人员进行模拟仿真实验进行过程、虚拟仪器开发及三维课件设计进行编写，本书未涉及计算机图形学的基本理论知识，而是立足于简单化、实用性，以大量实例进行仿真技术的直观阐述。如需深入学习OpenGL可参看计算机图形学方面的相关书籍。

本书以Visual Basic 2010学习版为开发环境，以OpenGL API函数库为开发工具进行虚拟实验仪器及仿真实验过程的开发研究。目前，国内外与OpenGL函数库相关的开发工作几乎都是在C、C＃或C++语言下进行，很少有涉及Visual Basic开发环境的文章和相关实例，在Visual Basic 2010开发环境下利用OpenGL进行开发的相关书籍更是凤毛麟角；而目前Visual Basic是世界上使用人数最多的开发语言之一，为使Visual Basic初级开发者能利用OpenGL函数进行三维仿真及虚拟实验仪器的开发，本书本着简单、易学和模仿学习的原则进行了大量实例演示，对利用Visual Basic进行三维实验过程研究或虚拟实验仪器开发感兴趣的人员应该会有一定帮助。

全书共分十章进行编写，前八章主要介绍常用的OpenGL库函数及其函数的用法。在前八章知识的基础上，第九章给出了一个有关真空镀膜机模型的读取及控制实例，第十章在此基础上利用蒙特卡罗方法对薄膜生长过程进行仿真，以解决真空镀膜实验中薄膜成长机理的直观表达难题。

由于作者水平有限，书中难免存在疏漏和欠妥之处，希望广大读者批评指正。书中相关代码的索取、书中相关问题的讨论及有关建议可与作者联系：YL5923@sina.com.cn。

杨　亮

2012年3月

目　录

第 1 章　OpenGL 开发环境

经过半个多世纪的发展，个人计算机无论从硬件配置到软件应用均有迅猛发展，现在已经渗入人们生活、工作、娱乐和学习等领域。计算机可方便地实现管理数码照片、存储生活 DV 片段、记录生活信息、方便家庭成员间异地联系等；在工作方面，离开计算机人们根本无法工作，公司将会处于瘫痪状态；在空闲时间网上冲浪、在线音乐、点播视频、在线游戏及跟同学、同事或朋友聊聊天也是一种十分惬意的个人消遣选择。总之，个人计算机已经进入当今社会的各个领域，在社会发展中占有举足轻重的地位。

近年来，3D 电影《怪物史来克》、《阿凡达》等相继上映及 3D 大型游戏、在线游戏相继发布，通用软件、专业软件在三维图像输出方面均有所增强，达到了相当的高度，这些均标志着计算机已经进入到 3D 时代。20 世纪是计算机互联网的时代，而 21 世纪计算机将跨入三维时代。而在教育教学方面，大部分课件或视频软件还停留在二维输出图像或传统幻灯片播放的计算机初级多媒体应用阶段，在某种意义上讲，其发展水平仍停留在 20 世纪水平并没有太大的进步。与其他行业相比，计算机在教育教学方面并没有能充分发挥其内在的巨大潜力。OpenGL 函数库在这方面可大有作为，它不但能输出三维图像图形，还能与三维模型进行交互，该函数库的应用可将教育教学软件提升到新的高度，制作出可与目前流行游戏相媲美的视觉效果。

1.1　什么是 OpenGL

OpenGL 函数库是 SGI 公司开发的一种硬件和图形软件接口，可独立于操作系统和硬件环境的三维图形库，可在不同的平台、不同的计算机上进行应用，甚至可以应用到其他智能设备（如 PalmOS、Symbian、Windows mobile、Linux、Android、iPhoneOS 和黑莓）。由于其强大的图形处理功能、三维图形效果表现突出且易于使用和跨平台的能力，事实上已成为业界的图形标准。目前，包括 ATI 公司 UNIX 软件实验室、IBM 公司、DEC 公司、SUN 公司、HP 公司、Microsoft 公司和 SGI 公司在内的多家在计算机市场占领导地位的大公司都采用了 OpenGL 图形标准。许多软件厂商也纷纷以 OpenGL 为基础推出自己的产品，其中主要有：Soft Image、3D Studio MAX、Open Inventor、World ToolKit 等。OpenGL 在科学技术领域也颇受许多专业技术人员的喜爱，已经设计出大量专业软件，应用于建筑、产品设计、医学、地球科学、流体力学、材料计算等领域。Microsoft 公司在 Windows NT 中首先提供 OpenGL 图形标准动态库原生支持，并在 Win98、Win2000、XP、Vista、Win7 等操作系统都

进行了更好的原生支持，使 OpenGL 在个人计算机中得到广泛应用。尤其是 OpenGL 三维图形加速卡的高速发展，使得人们可以在个人计算机甚至手机等智能设备上实现三维图形应用，如 CAD 设计、仿真模拟、三维游戏、广告设计、虚拟现实及手机游戏等。

OpenGL 是一个 API（Application Programming Interface），它本身是一个与硬件无关的编程接口，因此 OpenGL 中没有包括处理窗口和用户输入的命令或函数，它只能提供基本的几何图元（点、线和面），这也使编程开发更加灵活。

1.2 OpenGL 的主要功能

OpenGL 作为三维图形接口 API，通过调用此 API 可以绘制三维图形，只要你有决心编写代码，它就可以实现极其复杂的模型构建，同时也能方便地实现三维图形的交互操作。一般情况下，复杂的模型可通过三维应用软件（MAX、Maya 等）来构建，然后再通过 OpenGL 函数进行读取并利用软件接口来实现三维模型与用户的交互。OpenGL 主要实现的功能如下。

（1）几何建模　在 OpenGL 中提供了绘制点、线、多边形等基本的形体函数，还提供了绘制复杂三维曲线、曲面（如 Bezier、Nurbs 等）和三维形体（如球、锥体和多面体等）的函数。由于 OpenGL 是以顶点为图元，由点构成线，由线及其拓扑结构构成多边形。所以应用这些建模函数可构造出几乎所有的三维模型。

（2）坐标变换　包括基本变换和投影变换。基本变换有平移、旋转、变换、镜像四种模型变换，投影变换有正投影（Orthographic Projection）和透视投影（Perspective Projection）。其中，正投影是指投影后物体的大小与视点的远近无关，通常用于 CAD 设计；而透视投影则更符合人们的心理习惯，离视点近的物体大些，离视点远的物体小些。此外，在 OpenGL 中还要定义投影范围，只有在该范围中的物体才会被投射到计算机屏幕上，投影范围外的物体将被裁减掉。

（3）颜色模式设置　OpenGL 颜色模式有两种，即 RGBA 模式和颜色索引模式（Color Index）。在 RGBA 模式下所有颜色的定义用 RGB 三个分量值来表示，有时也会加上 Alpha 值（表示透明度）。RGB 三个分量值代表最终颜色中所占的比例。如：（1，1，0）表示黄色，（0，0，1）表示蓝色。颜色索引模式下每个像素的颜色是用颜色索引表中的某个颜色索引值表示（类似于 Windows 下从调色板中选取颜色）。由于三维图形处理中要求颜色灵活，而且在阴影、光照、雾化、融合等效果处理中 RGBA 的效果要比颜色索引模式好。所以，在 OpenGL 实现某些应用时大多采用 RGBA 模式而很少采用颜色索引模式。

（4）光照和材质设置　OpenGL 可设置四种光，即辐射光、环境光、镜面光和漫反射光。材质也是用模型表面的反射特性表示。OpenGL 的光源大体分为三种：环境光（Ambient light），即来自于周围环境没有固定方向的光；漫射光（Diffuse light）来自同一个方向，照射到物体表面时在物体表面的各个方向上均匀发散；镜面光

（Specular light）则是来自于同一方向，也沿同一个方向反射。辐射光是一种特殊的环境光，它不来自某种特定光源，通常被用作场景的自然光源。在 OpenGL 中，用材质对光的三原色（红绿蓝）的反射率大小来定义材质的颜色。与光源相对应，材质的颜色也分为环境色、漫反射色和镜面反射色，由此决定该材料对应不同的光呈现出不同的反射率。由于人所看到的物体的颜色是光源发出的光经物体反射后进入眼睛的颜色。所以，物体的颜色是光源的环境光、漫反射光、镜面反射光、材料的环境色、漫反射色和镜面反射色的综合。OpenGL 可提供 8 个光源。

（5）图像功能　提供像素拷贝和读写操作的函数，还提供了反走样、融合和雾化等功能以增强图像效果。融合（blending）、雾化（fog）与反走样（Antialiasing）是 OpenGL 中的三种特殊效果处理方法。融合提供了一种透明或半透明显示的技术；雾化处理则根据物体距离视点的远近对其进行恰当的模糊处理；反走样则可减少在绘制离散化的图形时所产生的误差走样，使图像更加圆润。

（6）纹理映射　OpenGL 的纹理映射功能可十分逼真地再现物体表面的细节。纹理映射（Texture Mapping）的方法运用得很广，尤其在描述具有真实感的物体方面。比如绘制一面砖墙，就可以用一幅真实的砖墙图像或照片作为纹理贴到一个矩形上，这样，一面逼真的砖墙就画好了。如果不用纹理映射的方法，则墙上的每一块砖都必须作为一个独立的多边形来画。另外，纹理映射能够保证在变换多边形时，它上面的纹理图案也随之变化。例如，以透视投影方式观察墙面时，离视点远的砖块的尺寸就会缩小，而离视点较近的就会大些。此外，纹理映射也常常运用在其他一些领域，如飞行仿真中常把一大片植被的图像映射到一些大多边形上用以表示地面，或用大理石、木材、布匹等自然物质的图像作为纹理映射到多边形上表示相应的物体。

（7）实时动画　利用 OpenGL 的双缓存（Double Buffer）技术可获得平滑逼真的动画效果。OpenGL 提供了双缓存，可以用来制作动画。也就是说，在显示前台缓存内容中的一帧画面时，后台缓存正在绘制下一帧画面，当绘制完毕，则后台缓存内容便在屏幕上显示出来，这样循环反复，屏幕上显示的总是已经画好的图形，于是看起来所有的画面都是连续的。如果没有启用双缓存功能，Visual Basic 2010 制作的动画运行时会有一种闪烁感。

（8）交互技术　方便的三维图形交互接口，选择（Select）、拾取（Pick）和反馈（Feedback），可十分方便地实现人机交互操作。选择模式为用户提供了一种拾取物体的机制，反馈模式将绘图信息加以组织并返回给应用程序，成为程序设计的重要资料和用户所需的信息。选择物体是响应 OpenGL 应用程序的一个拾取事件，该事件通常是由鼠标来触发的。为此，应用程序必须建立物体的名称，并将名称加以组织，然后当发生拾取事件时，就对其进行响应，这样就能开发出具有高交互功能的程序，提高了程序的互动性。

1.3 开发环境

1.3.1 Visual Basic 6.0 下使用 OpenGL 的方法

由于计算机软件及硬件技术的发展突飞猛进，OpenGL 已经可以在大多数个人计算机上很好地运行，使原来只能在图形工作站才能实现的效果现在在个人计算机中也同样能实现，这主要归功于 Microsoft 和 SGI 等公司将 OpenGL 在 Windows 平台上予以实现，Microsoft 也在其开发环境 Visual Studio 各版本给以很好的支持。例如，在 Visual Basic 6.0 开发环境中需下载 Vbogl.TLB 文件，这是一个可免费使用的第三方库，它封装了大量的底层 OpenGL 库函数，很大程度上简化了开发工作。该库在 Visual Basic 6.0 下使用也很方便，只需在 Visual Basic 6.0 开发环境中由“工程”菜单→“引用”对话框下选择“VB OpenGL API 1.2 (ANSI)”，引用界面如图 1-1 所示，这样即可在 Visual Basic 6.0 中进行 OpenGL 的开发啦！

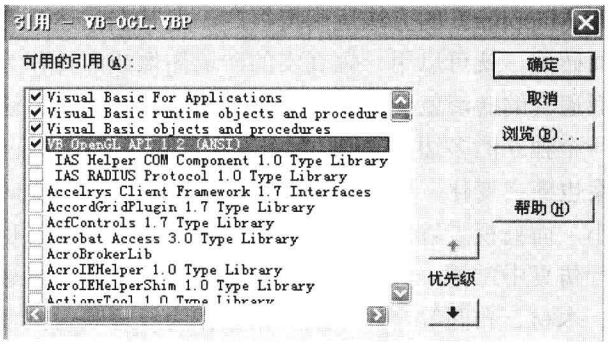


图 1-1 Visual Basic 6.0 中引用 VB OpenGL API 1.2 对话框

由于在 Visual Basic 6.0 下没有现成的 OpenGL 控件，必须将窗体或图片框等图像设备的像素格式重新进行定义，使其能在 OpenGL 下进行图形绘制。下面是 Visual Basic 6.0 在引用 VB OpenGL API 时定义的设备部分像素描述结构：

```
    Dim Pfd As PIXELFORMATDESCRIPTOR                    '设置像素描述结构
      '设置上下文的 PFD 结构
      Pfd.nSize = Len(Pfd)                              '像素描述结构的大小
    Pfd.nVersion = 1                                    '版本号
      Pfd.dwFlags = PFD_SUPPORT_OPENGL Or PFD_DRAW_TO_WINDOW Or PFD_
DOUBLEBUFFER Or PFD_TYPE_RGBA                           '缓存区支持 OpenGL 绘图、缓
存区的输出显示在一个窗口中、颜色缓存区是双缓存、使用 RGBA 颜色格式
      Pfd.iPixelType = PFD_TYPE_RGBA                    '使用 RGBA 颜色格式
      Pfd.cColorBits = 32                               '位深度缓存
```

```
Pfd.cDepthBits = 16                                    位深度缓存
Result = ChoosePixelFormat(hDC1, Pfd)                  '选择设备
If Result = 0 Then                                     '判断是否应用成功
    MsgBox "OpenGL Initialization Failed!", vbCritical
    InitGL = False
    Exit Function
End If
SetPixelFormat hDC1, Result, Pfd                       '设置设备的像素结构
hglRC1 = wglCreateContext(hDC1)                        '创建上下文
```

1.3.2 Visual Basic 2010 下使用 OpenGL 的方法

本书以 Microsoft 公司于 2010 年发布的 Visual Basic 2010 学习版为开发环境，由于 Visual Basic 2010 是基于.NET 4.0 的开发环境，所以不能再利用 Visual Basic 6.0 的 OpenGL 开发库，在.NET 2.0 发布时，麻省理工学院就发布了 Tao FrameWork 工具包，是主要针对.NET 下进行 OpenGL 开发的库函数，并集成了 Cg、Glut、OpenAL 等大量函数库，这样极大程度地提高了应用程序的开发效率。Tao FrameWork 主要针对可视化开发的 Visual Sudio.net 研发的工具包，内部封装了一个可视化的控件——SimpleOpenGLControl，该控件简单易用，省去了大量定义设置像素描述结构的工作。

由于 Tao FrameWork 工具包内置的 SimpleOpenGLControl 控件已经内部封装了这些代码，只需初始化操作就能直接使用，Visual Basci 2010 结合 OpenGL 函数库进行程序开发过程如下。

① 启动 Visual Basic 2010 学习版开发环境，并选择“新建项目”，如图 1-2 所示。

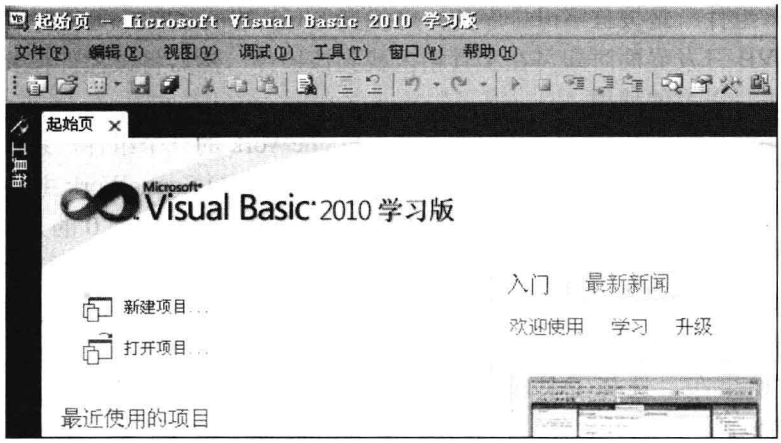


图 1-2 Visual Basica 2010 学习版起始页

② 打开新建项目对话框，并在名称栏内写入新建项目的名字（中英文均可），

创建类型选择“Windows 窗体应用程序”即可（如图 1-3 所示）。

③ 进入 Visual Basic 2010 开发环境页，并在工具箱的任一位置单击鼠标右键，弹出如图 1-4 所示快捷菜单并选择“选择项（I）...”。

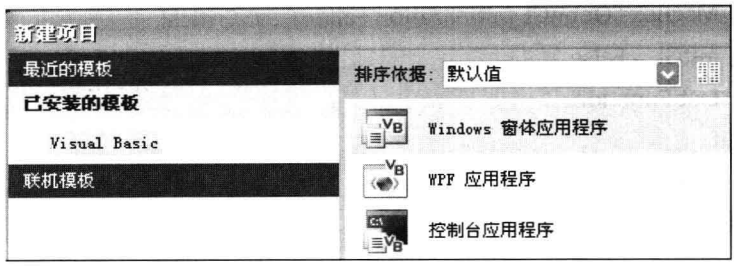


图 1-3　Visual Basic 2010 学习版新建项目页

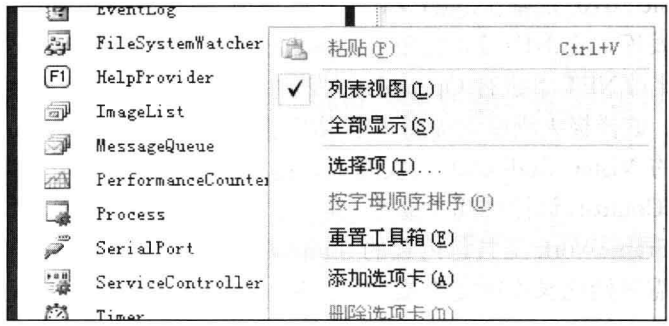


图 1-4　Visual Basic 2010 打开“选择项”对话框

④ 打开如图 1-5 所示的“选择工具箱项”，该对话框有 4 个选项卡，其中“.NET Framework 组件”是按首字母排序的，可找到 SimpleOpenGLControl 控件，并选择它。回到 VB 开发界面时即可在工具箱内找到该控件，然后将该控件拖到工具箱相应的分类下，以方便使用。

✧ SimpleOpenGLControl 控件是 Tao FrameWork 的一个组件，系统必须已经安装 TAO FrameWork 才能找到相应的控件。由于 TAO FrameWork 主要支持.NET Framework 2.0 开发，而 Visual Basic 2010 是.NET FrameWork 4.0 的开发环境，安装 TAO FrameWork 前系统先安装.NET Framework 2.0 才能顺利安装 TAO Framework，在 Windows 7 操作系统不必要安装.NET Framework 2.0 也可顺利安装 TAO Framework 工具包。

✧ 如果在“选择工具箱项”的“.NET Framework 组件”页内不能找到 SimpleOpenGLControl 控件，可利用图 1-5 页面底端的“浏览”按钮找到“X:\Program Files\TaoFramework\bin”（X 指 TaoFramework 程序安装的驱动器名称）目录下的“Tao.Platform.Windows.dll”文件即可将 SimpleOpenGLControl 引入到“选择工具箱项”并能进行正常操作。

⑤ 在“工具箱”中找到 SimpleOpenGLControl 控件，现在就可以像操作其他 Visual Basic 2010 控件一样操作它，但是由于 Visual Basic 2010 的绘图模式是基于 GUI+模式，而 SimpleOpenGLControl 是基于 OpenGL 的模式，所以 SimpleOpenGLControl 在绘图方法与 Visual Basic 2010 本身的控件的绘图方法有所不同。如果直接运行带有 SimpleOpenGLCtrol 控件的窗体会出现如图 1-6 所示的错误提示。

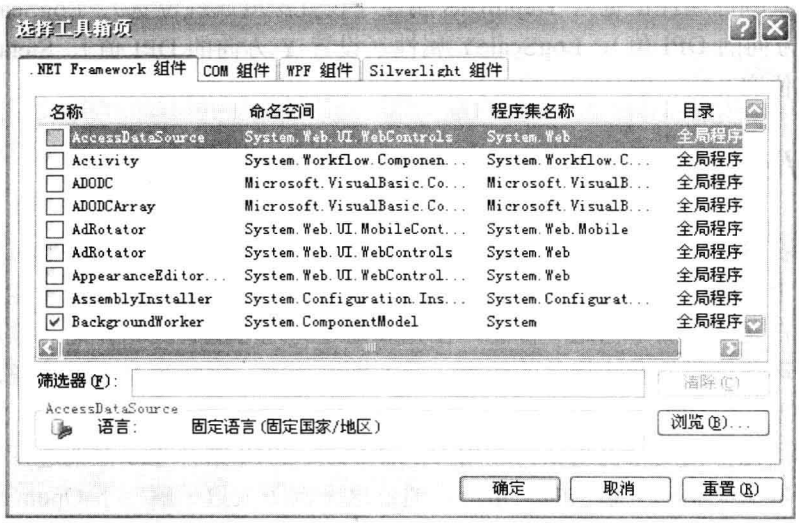


图 1-5　Visual Basic 2010 中的“选择工具箱项”操作界面

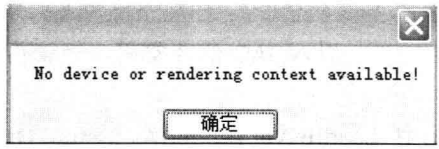


图 1-6　由于没有进行初始化操作，系统报 SimpleOpenGLControl 运行错误

现在，只要在 Form1_Load 事件中加入下列语句即可消除错误，但是运行后 SimpleOpenGLControl 控件上会出现一些杂乱的图形，这是由于没有清空该控件的缓冲区，控件将缓冲区内的内容直接绘制到窗口所致。

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MyBase.Load
        SimpleOpenGLControl1.InitializeContexts()'初始化控件的上下文
End Sub
```

⑥ SimpleOpenGLControl 控件的属性。SimpleOpenGLControl 控件的属性有几十个，但是绝大部分属性都是由.NET 控件的属性继承而来，由于 SimpleOpenGLControl 控件与.NET 控件绘画机制的不同使得继承的控件属性并不能真正起作用，

例如：BackColor 属性、FontColor 属性和 BackgroundImage 属性等均无法正常使用，但是像 Name 属性、Dock 属性和 BorderStyle 属性就能运行良好。SimpleOpenGL-Control 控件也拥有特殊的属性，如：AccumBits 属性（设置累积缓存区深度）、AutoCheckErrors 属性（自动检查错误）、AutoFinish 属性（自动完成）、AutoMakeCurrent 属性、AutoSwapBuffers 属性（自动交换缓存区数据）、ColorBit 属性（设置颜色属性位宽）、DepthBits 属性（设置深度属性位宽）、LogScaleX 属性（设置 X 方向的 DPI 值）、LogScaleY 属性（设置 Y 方向的 DPI 值）、StencilBits 属性（剪切位宽）。

1.4 Visual Basic 2010&OpenGL 开发示例

下面在 SimpleOpenGLControl 控件上先绘制一个彩色的二维三角形，为什么不在该控件上直接绘制三维图形呢？这是因为三维图形需要有光照存在才能显示出三维效果，在现实生活中也是一样的，在光线充足的地方能看到物体的三维形貌，但在黑暗的环境里只能看清物体的轮廓，看不到物体的三维特征及颜色特征，这里也是这个道理。

下面用几行代码来说明如何利用 Visual Basic 2010 结合 Tao 框架进行图形的绘制，在屏幕上绘制一个彩色的三角形，通过这个例子大致了解一下 OpenGL 绘制图形的过程：首先初始化 OpenGL 环境，包括清空缓存区、定义光照、定义纹理、定义雾，设置视口、视景体等操作；然后进行三维模型构建、移动、旋转或变换模型，其中基本函数的用法在后续章节会提到，基本代码如下：

```
imports Tao.OpenGl          '引入 TAO 命名空间，为使用 OpenGL 做好准备工作
Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.
EventArgs) Handles MyBase.Load
        SimpleOpenGlControl1.InitializeContexts()  '初始化控件的上下文
        Gl.glClearColor(0.0F, 0.0F, 0.0F, 0.0F)    '以全黑色来消除颜色缓存
区，当然也可以用其他颜色来消除，它使用的是 RGBA 格式，最大值不能超过 1，
最小值为 0
        Gl.glViewport(0, 0, SimpleOpenGlControl1.Width, SimpleOpenGlControl1.
Height)                                    '定义视口的大小
        Gl.glMatrixMode(Gl.GL_PROJECTION)  '设置视图投影变换矩阵
        Gl.glLoadIdentity()                '加载单位矩阵
        Gl.glOrtho(-10, 10, -10, 10, 0, -SimpleOpenGlControl1.Width / 2) '创建
平行投影空间，在平行投影空间中，远处的物体和近处的物体大小是一致的，这
主要用于科学计算、CAD 等方面
```

```
            Gl.glMatrixMode(Gl.GL_MODELVIEW)        '将矩阵变换对象切换为模
型视图变换
            Gl.glLoadIdentity()                     '加载单位矩阵
        End Sub
        Private Sub SimpleOpenGlControl1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles SimpleOpenGlControl1.Paint
            Gl.glClear(Gl.GL_COLOR_BUFFER_BIT Or Gl.GL_DEPTH_BUFFER_BIT)
                                                    '清除颜色缓存区及深度缓存区
            Gl.glBegin(Gl.GL_TRIANGLES)             '告诉 OpenGL 将要绘制三角形
            Gl.glColor3f(1, 0, 0)                   '定义颜色为红色
            Gl.glVertex2f(-5, 5)                '传输三角形的一个顶点坐标给 OpenGL
            Gl.glColor3f(0, 1, 0)               '定义颜色为绿色
            Gl.glVertex2f(5, 5)                 '传输三角形的一个顶点坐标给 OpenGL
            Gl.glColor3f(0, 0, 1)               '定义颜色为蓝色
            Gl.glVertex2f(0, -5)                '传输三角形的一个顶点坐标给 OpenGL
            Gl.glEnd()                          '结束图元的绘制
        End Sub
    End Class
```

程序的运行结果如图 1-7 所示，该图为 OpenGL 渲染的简单二维图形，但是它开启了使用 OpenGL 的大门，后续的 OpenGL 开发的实例均以此例框架为基础。

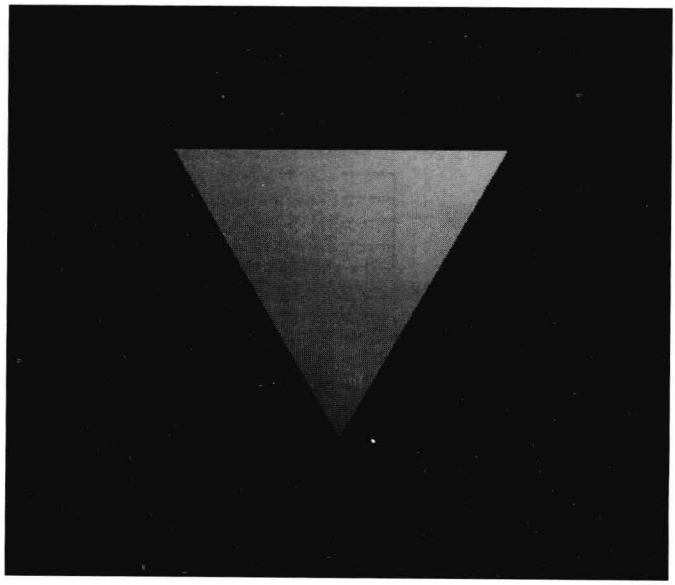


图 1-7　OpenGL 绘制的彩色三角形（彩图见封二）