

第3章 OpenGL 编程环境介绍

3.1 VS.NET 2008 开发环境简介

Microsoft. NET(以下简称. NET)是微软公司推出的下一代计算计划,该计划使微软现 有的软件在 Web 时代不仅适用于传统的 PC 设备,而且也能满足新设备、新业务的要求。 如今,. NET 技术和相应的产品已经得到了广泛应用,从中小型的应用程序到大型的企业级 应用软件,从桌面应用程序、Web 应用程序、Mobile 应用程序、Web 服务到操作系统开发、 嵌入式设备开发等,都能看到. NET 的身影。

对. NET 可以从两个方面理解。首先,. NET 是一个开发平台。它对微软之前的各种 主要开发平台进行了集成,提供了一套全新的 Windows 平台实现。例如,在. NET 平台下 不仅可以从事 VB,C++ 程序的开发,而且还可以使用特别为. NET 平台开发的 C + 语言进 行编程。. NET 平台要做的就是消除互连环境中不同软硬件以及服务的差异,使得不同设备 和系统都可以相互通信,使得不同的程序和服务之间都可以相互调用。其次,. NET 是一组 规范。. NET 平台本身就基于一系列规范,其中有些规范是由微软以外的其他组织来维护 的。例如定义诸如 C + , VB. NET 和 IL 语言的规范,定义数据交换格式的规范,如 SOAP 等。. NET 平台的组成如图 3-1 所示。



图 3-1 Microsoft. NET 平台的组成

图 3-1 中的 Visual Studio. NET 是微软推出的全新的. NET 开发工具。Visual Studio. NET 2008 对微软之前的主要开发工具做了全新集成并有了质的飞跃,它内置支持 Visual Basic. NET, Visual C[‡]. NET, Visual C⁺⁺. NET 和 Visual J[‡]. NET 等多种语言,并且各种语言拥有统一的开发环境,可以进行跨语言调试和跨语言调用。需要说明的是 Visual Studio. NET 并不是对 Microsoft Visual C⁺⁺, Microsoft Visual Basic 等语言的简单集成, Visual C⁺⁺. NET, Visual Basic. NET 等语言是基于全新的设计思想开发的。对计算机图形学的主要算法,本书采用C语言加 OpenGL 的方式加以实现,开发环境为 Visual C⁺⁺. NET 2008 并且采用 Win32 控制台应用程序的形式,创建控制台应用程序的步骤参考 3.3 节。

3.2 OpenGL 图形软件包介绍

3.2.1 OpenGL 概述

开放图形库 OpenGL(Open Graphics Library) 是一个功能强大的图形库,它实现了各种二维和三维的高级图形处理技术(如光照应用、纹理映射、透明处理等)。OpenGL 的前身 是 SGI 公司为其图形工作站开发的 IRIS GL 库,由于其性能良好,该库在跨平台的移植过程中发展成为 OpenGL。由于 OpenGL 的广泛应用,它已经成为一个事实上的工业标准。

从程序开发人员的角度来看,OpenGL 是一组绘图命令的 API,利用这些 API 能够方便 地描述二维和三维图形。OpenGL 的 API 集提供了物体表述、平移、旋转、缩放、光照、纹 理、材质、像素、位图、文字、交互以及提高显示性能等多方面的功能,基本涵盖了开发二维、 三维图形程序所需的各个方面。与一般的图形开发工具相比,OpenGL 具有如下几个突出 的特点。

(1) 跨平台特性。OpenGL 独立于硬件和窗口系统,在运行各种操作系统的各种计算 机上都能使用,具有很高的可移植性。

(2)应用的广泛性。OpenGL是目前最主要的二维、三维交互式图形应用程序开发环境,已经成为业界最受推荐的图形应用编程接口。OpenGL已被广泛地应用于 CAD/ CAM、三维动画、数字图像处理以及虚拟现实等领域。

(3)网络透明性。建立在客户端/服务器模式上的网络透明性是 OpenGL 的固有特性, 它允许一个运行在工作站上的进程在本机或通过网络在远程工作站上显示图形。利用这个 特点能够均衡各图形工作站的工作负荷,共同承担图形应用任务。

(4)高质量和高性能。无论在哪个应用领域,无论是在 PC、工作站还是大型机上,OpenGL 的高质量和高效率的图形生成能力都能得到充分体现,绘制的二维、三维图形效果逼真。

(5) 出色的编程特性。OpenGL 在各种平台上已经有多年的应用实践,加之严格的规范控制,因此 OpenGL 具有良好的稳定性和易使用性。

3.2.2 OpenGL 的主要功能

作为性能优越的图形应用程序设计接口(API), OpenGL 具有如下 9 大功能。

(1)模型绘制。OpenGL能够绘制点、线和多边形。应用这些基本的形体,可以构造出 几乎所有的三维模型。

(2)模型观察。在建立了三维景物模型后,就需要用 OpenGL 描述如何观察所建立的 三维模型。观察三维模型是通过一系列的坐标变换进行的。模型的坐标变换使观察者能够 在视点位置观察与视点相适应的三维模型景观。在整个三维模型的观察过程中,投影变换 的类型决定观察三维模型的观察方式,不同的投影变换得到的三维模型的显示效果也是不 同的。最后的视窗变换则对模型的显示结果进行裁剪缩放,即决定整个三维模型在屏幕上 的图像。

23

(3)颜色模式的指定。OpenGL应用了一些专门的函数来指定三维模型的颜色。程序 开发者可以选择两个颜色模式,即RGBA模式和颜色表模式。在RGBA模式中,颜色直接 由RGB值来指定;在颜色表模式中,颜色值则由颜色表中的一个颜色索引值来指定。开发 者还可以选择平面着色和光滑着色两种着色方式对整个三维景观进行着色。

(4) 光照应用。用 OpenGL 绘制的三维模型必须加上光照才能更加与客观物体相似。 OpenGL 提供了管理 4 种光(辐射光、环境光、镜面光和漫反射光)的方法,另外还可以指定 模型表面的反射特性。

(5)图像效果增强。OpenGL提供了一系列的增强三维景观的图像效果的函数,这些 函数通过反走样、混合和雾化来增强图像的效果。反走样技术用于改善图像中线段图形的 锯齿状现象,从而使图像更加平滑,混合用于处理模型的半透明效果,雾化使得影像从视点 到远处逐渐褪色,从而使图形效果更接近于真实。

(6) 位图和图像处理。OpenGL 还提供了专门对位图和图像进行操作的函数。位图和 图像数据均采用像素矩阵表示。

(7) 纹理映射。三维景物因缺少景物的具体细节而显得不够真实,为了更加逼真地表现三维景物,OpenGL提供了纹理映射的功能。OpenGL提供的一系列纹理映射函数使得 开发者可以十分方便地把真实图像贴到景物的多边形上,从而可以更加逼真地绘制三维 景观。

(8) 实时动画。为了获得平滑的动画效果,需要先在内存中生成下一幅图像,然后把已 经生成的图像从内存拷贝到屏幕上,这就是 OpenGL 的双存技术(double buffer)。OpenGL 提供了双缓存技术的一系列函数。

(9) 交互技术。目前有许多图形应用需要人-机交互, OpenGL 提供了方便的三维图形 人-机交互接口, 用户可以选择修改三维景观中的物体。

3.2.3 OpenGL 的基本语法

1. 相关函数库介绍

OpenGL 主要由以下函数库组成。

(1) OpenGL 核心库。库中函数以"gl"为前缀,共计约 115 个最基本的命令函数。这些函数用于常规的、核心的图形处理,如建立各种几何模型,产生光照效果、进行反走样和投影变换等。

(2) OpenGL 实用程序库。库中函数以"glu"为前缀,共计约 43 个函数。这些函数在任何 OpenGL 平台上都可以使用,它们通过调用核心库函数,实现一些较为复杂的操作,如纹 理映射、坐标变换、网格化、曲线曲面的绘制等。

(3) OpenGL 编程辅助库。库中函数以"aux"为前缀,共计约 31 个函数。这些函数主要用于窗口管理、输入处理、删除处理以及绘制一些简单的三维图形,但它们不能在所有的 OpenGL 平台上使用。

(4) OpenGL 实用程序工具包(OpenGL Utility Toolkit, GLUT)。包中函数以"glut" 为前缀,共计约 30 多个函数。这些函数主要提供基于窗口的工具,如窗口系统的初始化、多 窗口管理、菜单管理、字体设置以及一些较为复杂图形的绘制。它们可以在所有的 OpenGL

第3章 OpenGL 编程环境介绍

平台上运行。

(5) Windows 专用库。库中函数以"wgl"开头,共6个函数。这些函数主要用于链接 OpenGL 和 Windows NT 操作系统。

2. 命名规则

OpenGL 函数都遵循一个命名约定,即:

<库前缀><根命令><可选参数个数><可选参数类型>

通过函数的命名规则,我们可以了解函数来自哪个库,需要多少个参数以及参数的数据类型 等信息。例如函数 glColor3f(),gl 表示该函数来自核心库,Color 是该函数的根命令,表示 函数用于颜色设定,3f 表示函数采用3个浮点数参数。这种命名规则便于程序开发人员记 住参数列表,非常方便。有些函数也使用一个或多个符号常量,这些符号常量均使用 GL 开 头,常量名中各组成词用"_"分开,且都采用大写形式,如 GL_RGB、GL_POLYGON 等。

3. 数据类型

由于 OpenGL 具有平台无关性,而不同机器的数据类型表示范围有可能不同,因此 OpenGL 定义了自己的数据类型。表 3-1 给出了 OpenGL 数据类型和 C 语言数据类型的对应关系。

OpenGL 数据类型	内部表示法	C 语言数据类型
GLbyte	8 位整数	signed char
GLshort	16 位整数	short
Glint, GLsizei	32 位整数	long
GLfloat, GLclampf	32 位浮点数	float
GLdouble, GLclampd	64 位浮点数	double
GLubyte, GLboolean	8位无符号整数	unsigned char
GLshort	16 位无符号整数	unsigned short
GLuint, GLenum, GLbitfield	32 位无符号整数	unsigned long

表 3-1 OpenGL 数据类型和 C 语言数据类型的对应关系

3.3 设置 OpenGL 编程环境

3.3.1 OpenGL 编程环境设置

在安装了 Microsoft Visual Studio. NET 之后,当第一次编写 OpenGL 程序时,需要在 VS. NET 环境下配置 OpenGL 的编程环境,具体步骤如下。

(1)选择编程环境。编程环境选择为 Visual C++. NET 2008,在安装 VS. NET 的时候进行安装,操作系统一般为 Windows 7 或 Windows XP。

(2) 下载 OpenGL 开发包。下载 OpenGL 开发库文件(例如可以在 www. csdn. net 上 搜索并下载),将下载的压缩包解压后如图 3-2 所示。



图 3-2 OpenGL 开发库文件

(3) OpenGL 开发包安装。具体分为如下三步。

① 把解压得到的扩展名为 h(*.h)的文件复制到 x:\Program Files\Microsoft\Visual Studio 9.0\VC\include\GL 文件夹中。这里的 x 是安装 VS 的盘符号,如果没有 GL 这个 文件夹则可以自己新建一个,Visual Studio. NET 2008 环境下是需要新建 GL 文件夹的。

②把解压得到的扩展名为 lib(*. lib)的文件复制到 x:\Program Files\Microsoft\ Visual Studio 9.0\VC\lib 文件夹中,即静态函数库所在的文件夹。

③把解压得到的扩展名为 dll(*.dll)的文件复制到操作系统目录下的 System32 文件 夹中,例如 C:\Windows\System32。

(4) 在 VC++. NET 2008 开发环境中创建工程。具体步骤如下。

① 创建一个空的 Win32 控制台应用程序,工程名称为 OpenGLE xample,如图 3-3 所示。

项目类型(P):		模板(T):	.NET Framework 3.5 💌	
数編集 Reporting WCF Workflow 週述 其他语言 Visual C++ ATL CIR 常規 概 に 者能設 調成 Win3之 其他项目类型 測成项目	•	Visual Studio 已杂装的模板 國 由至义向导 潮試项目 潮域支援 動Krc 应用程序 MFC 应用程序 MFC 应用程序 MFC 应用程序 MFC 应用程序 MFC 容能设备太市地区 支持 MFC 智能设备应用程序 通 Wrin32 智能设备页目 週 Wrin4dws 週 影库 契約模板 □ 搜索联机模板	 Windows 智体应用程序 CLR 控制台应用程序 工 项目 生成文件项目 工 広文件项目 CLR 空项目 MFC DLL MFC 智能设备 DLL Win32 项目 Windows 閣体控件库 空项目 	
用于创建 Win32 控	制台应用程序的项目			
名称(N):	OpenGLExample			
位置(L):	C:\Users\wong\Des	iktop	▼ 浏览(8)	
解决方案名称(M):	OpenGLExample		☑ 创建解决方案的目录(D)	
-			确定 取消	

图 3-3 创建 Win32 控制台应用程序

单击图 3-3 中的"确定"按钮后在弹出的界面中单击"下一步"按钮,出现如图 3-4 所示的界面,单击"完成"按钮创建空的 Win32 控制台应用程序。

② 链接 OpenGL 函数库。该操作主要使编译器能够找到 OpenGL 函数库编译程序。 具体步骤如下。

(a) 在 Win32 控制台应用程序中创建一个后缀名为 cpp 的文件(右键→"添加"→"新建 项"),如 OpenGLExample. cpp。

(b) 在 VC++. NET 2008 中右击项目, 再单击"属性", 如图 3-5 所示。

			解決方案 "OpenGLExample" (I 个项目)		
Win32 应用程序向导 - DD/	ALine 程序设置	-? ×	- ₩ Opercus - ₩ ₩ - ₩ - ₩ - ₩ - ₩	生成(U) 重新生成(E) 清理(N) (仅用于项目(J) → 技配置优化(P) → 自定义生成规则(B)	
観述 应用程序设置	広用程序类型: ● Yindows 広用程序(2) ● 控制自应用程序(2) ● 加L(2) ● 静态库(3) 附加透项: 図 空项目(2) □ 导出符号(3) □ 形将阐译头(2)	添加公共头文件以用于: □ ATL (点) □ MFC (加) (仅适用于 DLL 项目 : 指定 DLL 核导出符	ا م الم الم الم الم الم الم الم الم الم	工具生成顺序(L) 添加(D) → 引用(F) 添加 Web 引用(E) 查看美关系图(V) 设为启动项目(A) 调试(G) → 剪切(T) 私況(P) 終発(V)	
	〔〈上─歩〕 〔	下一步〉 完成 取消	C C	重命名(M) 卸载项目(L) 在 Windows 资源管理器中打开文件夹(X) 屬性(R)	

图 3-4 创建空项目

图 3-5 设置工程 OpenGLExample 的属性

(c) 在弹出的"属性"菜单中选择"链接器"→"输入",在"附加依赖项"中输入"glut32. lib Opengl32. lib Glu32. lib glaux. lib",如图 3-6 所示。

744 Tri-			
加中価性 配置属性 常規 御試 C/C++ 能設置 常規 電払 常規 電払 常規 電払 常規 電払 構見 電払 構用 電払 構見 電払 構用 一 電払 一 一 一 一 一 一 一 一 一 一 一 一 一	回改制则 都所有對以車 醫務方式 大定又文件 製快添加到現乎集 人尤若對選及(件) 则符号引用 Bho酸的 DLL 学集链接例源 学集链接例源	glut32.lib Opengl32.lib Glu32.lib glaux.lib 否	

图 3-6 属性设置

(d) 在如图 3-4 所示的对话框中,单击 C++ →"预处理器",在预处理器定义栏的最后面 加上"GLUT BUILDING LIB"。 (5) 编写第一个 OpenGL 程序。 #include <gl/glut.h> void Initial (void) { //设置窗口背景色为白色 glClearColor(1.0f,1.0f,1.0f,1.0f); //指定设定投影参数 glMatrixMode(GL PROJECTION); //设置投影参数 gluOrtho2D(0.0,200,0.0,150.0); } void myDisplay (void) { glClear(GL COLOR BUFFER BIT); //用当前背景色填充窗口 glColor3f(0.0f,0.0f,0.0f); //设置当前的绘图颜色为黑色 glRectf(50.0f,100.0f,150.0f,50.0f); //绘制一个矩形 //清空 OpenGL 命令缓冲区,执行 OpenGL 命令 glFlush(); } int main(int argc, char * argv[]) { glutInit(&argc, argv); glutInitDisplayMode(GLUT RGB|GLUT SINGLE); //初始化窗口的显示模式 glutInitWindowPosition(100, 100); //设置窗口的位置 glutInitWindowSize(400, 400); //设置窗口的大小 glutCreateWindow("第一个 OpenGL 程序"); //创建窗口 //设置当前窗口的显示回调函数 glutDisplayFunc(&myDisplay); //完成窗口初始化 Initial(); //启动主 GLUT 事件处理循环 glutMainLoop(); return 0; }

}

程序运行结果如图 3-7 所示。

3.3.2 OpenGL 程序的结构

在安装了 VC++. NET 2008 并设置完编程环 境之后,我们就可以正式编写 OpenGL 程序了。下 面以 3.3.1 节的 OpenGL 源程序为例,说明利用 OpenGL 绘制图形的步骤。

1. 包含 OpenGL 头文件

使用 OpenGL 绘制图形程序,首先要引入 OpenGL 核心库以及其他需要使用的库的头文件。

此外,由于 OpenGL 没有包含窗口系统,因此需要通过头文件来引入窗口系统。可以使用 GLUT 库函数实现窗口管理,此时就不需要包含 gl.h和 glu.h,因为 GLUT 保证了这两



图 3-7 程序运行结果

第3章 OpenGL编程环境介绍

者被正确包含。即使用下面的方式在引入窗口系统的同时也能实现对 gl.h 和 glu.h 的 包含。

#include <gl/glut.h>

3.3.1节的程序就是采用这种方式。当然如果程序中还使用了其他的 C 或 C++ 程序 库,也需要引入相应的头文件,如 stdio.h。

2. 使用 GLUT 库函数实现窗口管理

要使用 GLUT 库首先需要进行初始化,对命令行的参数进行处理,相应的语句是:

glutInit(&argc,argv);

接着需要用 glutCreateWindow 函数创建一个窗口,该函数的参数是一个保存了窗口标题名称的字符串,如:

```
glutCreateWindow("第一个 OpenGL 程序");
```

在创建窗口之前,还要用 glutInitDisplayMode 函数设定窗口的显示模式,包括缓存和颜色模型等。该函数的参数是一些符号常量的组合,例如 GLUT_RGB|GLUT_SINGLE 表示的含义是窗口使用单缓存即 RGB 颜色模型,即:

glutInitDisplayMode(GLUT RGB|GLUT SINGLE);

另外,在创建窗口之前还需要设定窗口的位置和大小,即:

```
glutInitWindowPosition(100,100);
glutInitWindowSize(400,400);
```

其中位置以屏幕左上角为原点,用整数表示坐标,单位为像素。窗口大小也以整数表示宽度 和高度,单位为像素。上面的程序定义了一个距离屏幕左边界100个像素,距离屏幕上边界 100个像素,宽度为400个像素,高度为300个像素的窗口。

做完初始化工作之后,就需要定义窗口中显示的内容了。通常 OpenGL 将显示内容定 义在一个不带任何参数的函数内,例如上面例子中的 myDisplay 函数,并通过 glutDisplayFunc 函数将 myDisplay 指定为当前窗口的显示内容函数,即:

```
glutDisplayFunc(&myDisplay);
```

实际上,显示在窗口内的图形因为屏幕的刷新会不断地被重绘,而 glutDisplayFunc 函数指定 myDisplay 函数作为每次窗口重绘时调用的函数,因此也把 myDisplay 函数称为显示回调函数。不过这时窗口仍然不会显示在屏幕上,需要调用如下函数:

glutMainLoop();

该函数使窗口框架运行起来,使设置的显示回调函数开始工作,直到用户终止程序为止。

3. 利用 OpenGL 绘制图形

接下来该说一下显示回调函数 myDisplay 以及初始化函数 Initial 的组成了。在实现了窗口管理之后,我们需要调用 OpenGL 函数来实现图形的绘制。

```
glClearColor(1.0f,1.0f,1.0f,1.0f);
```

29

这条语句的作用是指定窗口的背景色为白色。函数的前3个参数为指定颜色的红、绿、蓝三 个颜色的分量,每种颜色成分使用0.0~1.0之间的任意有效浮点数来表示颜色值。红绿蓝 三个颜色分量如果都是1.0,则混合颜色为白色,如果三个分量都是0.0,则混合颜色为黑 色,三个颜色分量取不同的值会产生不同的颜色。函数的最后一个参数是颜色的 Alpha 成 分,它用于指定颜色混合后的特殊效果,如透明效果。Alpha 值的范围也是0.0~1.0,其中 Alpha 值为0.0表示对象是完全透明的,Alpha 值为1.0表示对象完全不透明。

指定了窗口的背景色之后,在每次绘制图形前需要用到如下函数:

glClear(GL COLOR BUFFER BIT);

该函数用 glClearColor 函数中指定的值设定颜色缓冲区的值,即将窗口中每个像素的颜色 设置为背景色。除了设定背景色外,还可以设置绘制图形所用的颜色,即:

glColor3f(0.0f,0.0f,0.0f);

该函数的参数和 glClearColor 函数类似。在绘制函数之前,还有一个问题需要解决。由于 OpenGL 按照三维方式来处理图形,即二维矩形也是在一个三维坐标系中创建的。单显示 器是一个二维显示设备,因此需要一个投影变换将三维图形(即二维矩形)投影到显示器的 二维空间中。这里使用如下语句:

```
glMatrixMode(GL_PROJECTION);
gluOrtho2D(0.0,200,0.0,150.0);
```

这两个函数用来设置投影参数。其中 gluOrtho2D 函数指定使用正投影将一个 x 坐标在 0. 0~200.0, y 坐标在 0.0~150.0 范围内的矩形坐标区域投影到显示器窗口中,任何在整个 矩形坐标区域内绘制的图形都可以显示在窗口中,任何坐标范围不在该区域内的图形都不 能显示。接着调用如下函数:

```
glRectf(50.0f,100.0f,150.0f,50.0f);
```

该函数绘制一个左上角在(50.0,100.0),右下角在(150.0,50.0)的矩形。在 myDisplay 函数的最后,调用了如下函数:

glFlush();

该函数的作用是清空 OpenGL 命令缓冲区,强制执行命令缓冲区中的所有 OpenGL 函数。

习题

- 3-1 上机熟悉 Visual C++. NET 2008 开发环境。
- 3-2 OpenGL 有什么特点?
- 3-3 OpenGL 有哪些功能?
- 3-4 上机设置 OpenGL 编程环境。
- 3-5 上机调试 3.3.1节 OpenGL 代码,熟悉 OpenGL 程序结构。