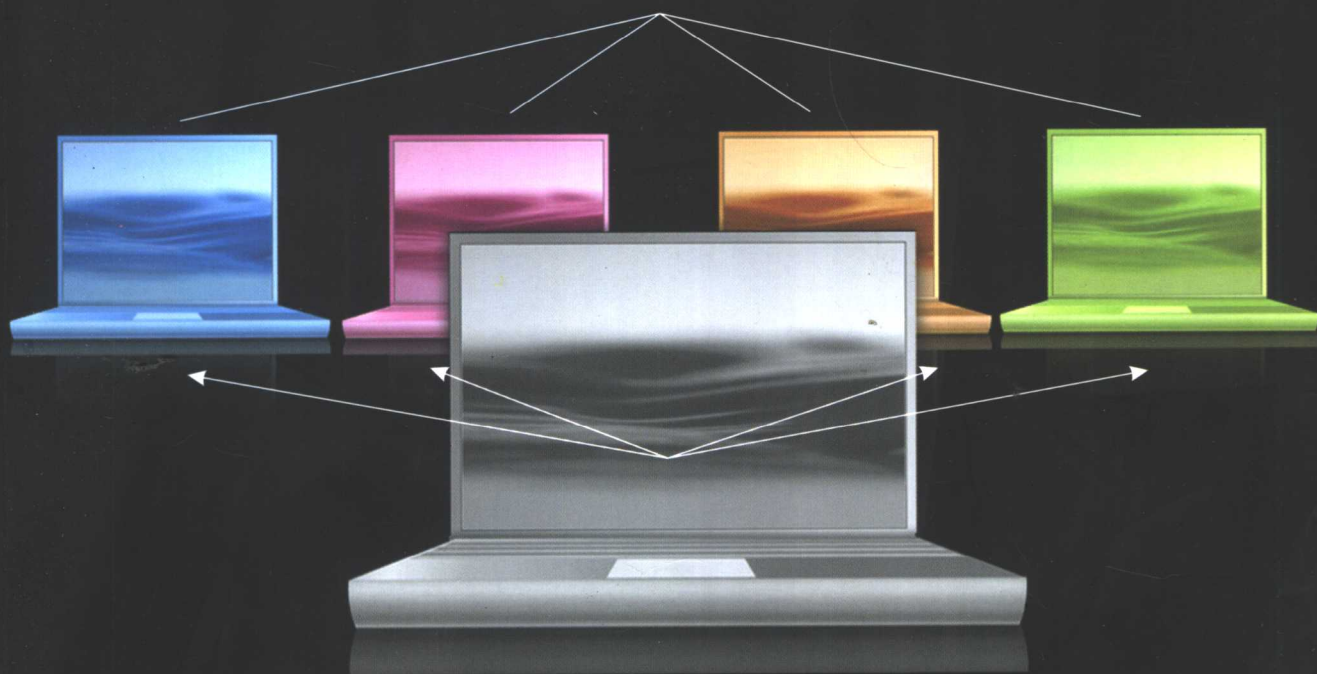


陕西省“三五”人才基金资助

OpenGL

图形程序开发实务

薛惠锋 吴慧欣 解丹蕊 编著



西北工业大学出版社

陕西省“三五”人才基金资助

OpenGL 图形程序开发实务

薛惠锋 吴慧欣 解丹蕊 编著



西北工业大学出版社

【内容简介】 OpenGL 作为一种图形工业标准,在实际工作中应用广泛并占有非常重要的地位。本书采用在 Delphi 的编程环境讲述 OpenGL 的功能原理,主要介绍了计算机图形学的发展及开发现状、OpenGL 的有关理论、OpenGL 的特点及工作原理、OpenGL 的基础知识及高级应用技巧,最后结合实例系统地讲述了如何综合利用 OpenGL 技术绘制复杂的三维真实感图形。

本书深入浅出、结构严谨、内容翔实,创新性、实用性很强,可供学习计算机图形学的本科生、研究生、科研人员及图形图像开发人员阅读参考。

图书在版编目(CIP)数据

OpenGL 图形程序开发实务/薛惠锋等编著. —西安:西北工业大学出版社,2005.6
ISBN 7-5612-1939-3

I. O… II. 薛… III. 图形软件,OpenGL-程序设计 IV. TP391.41

中国版本图书馆 CIP 数据核字(2005)第 043683 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路 127 号 邮编:710072 电话:(029)88493844 88491757

网 址:www.nwpup.com

印刷者:陕西向阳印务有限公司

开 本:787 mm×1 092 mm 1/16

印 张:19.25

字 数:460 千字

版 次:2005 年 8 月第 1 版 2005 年 8 月第 1 次印刷

定 价:29.00 元

前言

科学计算可视化、计算机动画和虚拟现实是近年来在计算机图形学领域内的三大热门研究方向,它们的技术核心都是三维真实感图形的绘制。当前,三维真实感图形技术已成功并广泛地应用在军事、航天、航空、医学、地质勘探、三维游戏和工业 CAD 设计等领域中,使人们能够在三维图形世界中直接对有形体的信息进行操作,并与计算机进行直接交流。这种技术已经把人和机器的力量以一种直觉而自然的方式加以统一,这种革命性的变化无疑将极大地提高人们的工作质量和工作效率。因此,积极推动计算机图形学领域的研究具有十分重要的理论和现实意义。

OpenGL 是 Open Graphics Library 的缩写,它是 SGI 公司开发的一套高性能图形处理系统,是图形硬件的软件界面,意为开放的图形库。通过 OpenGL,程序员可以创建交互式的应用程序,实现具有逼真效果的三维图形图像,从而在要求高度模拟真实世界的诸多领域中可以大显身手。由于其开放性和高度的可重用性,目前已成为业界标准。OpenGL 被设计成独立于硬件、独立于窗口系统,在运行各种操作系统的各种计算机上都可采用,并能在网络环境下以客户/服务器模式工作,是专业图形处理、科学计算等高端应用领域的标准图形库。

OpenGL 作为一种图形工业标准,几乎所有的可视化应用程序开发工具都支持 OpenGL 的集成,如 Delphi,VC++,VB,Java 等。然而,目前介绍 OpenGL 原理的可参考资源几乎都是基于 C 或 VC++ 的,这对于广大熟悉 Delphi 编程环境的开发人员来说,无疑是一种遗憾。众所周知,Delphi 以其简洁易用的可视化开发环境、高效率的编译器、功能强大的 Pascal 语言在可视化开发工具领域中占有非常重要的地位。基于上述因素考虑,本书采用在 Delphi 的编程环境讲述 OpenGL 的功能原理。

本书共由 16 章组成。第 1 章简单介绍了计算机图形学的发展、图形图像的开发现状;第 2 章和第 3 章介绍了 Delphi 编程基础和面向对象的有关理论;第 4 章介绍了 OpenGL 图形库的基本概况、OpenGL 的特点和工作原理;第 5 章介绍了实体建模基础,讲述了 OpenGL 的基本绘图命令,包括点实体、线实体和面实体的描述;第 6 章介绍了 OpenGL 中的视图变换理论,包括模型变换、视图变换、投影变换、视区变换以及矩阵堆栈操作等内容;第 7 章和第 8 章介绍了 OpenGL 中的颜色

和光照;第9章至第13章分别介绍了 OpenGL 的高级应用技巧,包括混和、反走样、雾、显示列表、位图、字体、纹理映射、帧缓存和动画等;第14章介绍了求值程序和非均匀有理 B 样条的原理;第15章解释了交互式 OpenGL 工作原理,包括选择机制和反馈机制;第16章作为总结,通过几个实例系统地讲述了如何综合利用 OpenGL 技术绘制复杂的三维真实感图形。

本书深入浅出、结构严谨、内容翔实。书中提供的大量源代码都在 Delphi 7.0 环境下编译通过。本书不仅可以作为大专院校本科生、研究生学习计算机图形学的教程,而且对于科研院所的研究人员、计算机图形图像的开发人员都具有极高的参考价值。

参加本书编著的人员有薛惠锋、吴慧欣、解丹蕊等,此外夏小野、高雄、徐荣跃和白林飞也参加了本书的整理工作。由于水平有限,书中难免存在疏漏和错误之处,敬请各位读者朋友批评指正。

编著者

2005 年 4 月

目 录

第一章 图形图像开发现状	1
1.1 蓬勃发展的计算机图形学	1
1.2 OpenGL 引领图形图像开发的潮流	3
1.3 开发工具的选择	4
第二章 Delphi 语言基础	6
2.1 Delphi 的集成开发环境	6
2.1.1 主窗体	6
2.1.2 对象树状视图	8
2.1.3 对象观察器	8
2.1.4 窗体设计器	9
2.1.5 代码编辑器.....	10
2.2 Delphi 语言的基本概念	12
2.2.1 保留字.....	12
2.2.2 标识符.....	12
2.2.3 指令符.....	13
2.3 数据类型.....	14
2.3.1 简单数据类型.....	14
2.3.2 字符串类型.....	16
2.3.3 结构类型.....	18
2.3.4 指针类型.....	20
2.3.5 变体类型.....	22
2.3.6 强制类型转换和类型约定	22
2.4 变量和常量.....	23
2.4.1 变量.....	23
2.4.2 常量.....	23

2.5 运算符和表达式	24
2.5.1 赋值运算符	24
2.5.2 比较运算符	24
2.5.3 逻辑表达式	24
2.5.4 算术运算符	24
2.5.5 位运算符	25
2.5.6 加减运算符	26
2.5.7 指针运算符	26
2.5.8 集合运算符、类运算符和 & 运算符	26
2.6 语句	27
2.6.1 语句的书写规则	27
2.6.2 注释语句	27
2.6.3 赋值语句	27
2.6.4 流程控制语句	28
2.6.5 循环语句	29
2.7 过程与函数	31
2.7.1 过程和函数的声明	31
2.7.2 常用函数与过程	32
2.7.3 过程和函数的参数	34
2.7.4 过程和函数的调用	35
2.7.5 作用域	38
第三章 面向对象编程	39
3.1 类与对象	39
3.2 类的声明	40
3.3 类的成员	41
3.3.1 数据域	41
3.3.2 类的属性	41
3.3.3 类的方法	43
3.4 OOP 的三个特性	44
3.4.1 数据封装	44
3.4.2 继承	45
3.4.3 多态性	46
3.5 类操作符	47
3.6 对象	48
3.6.1 对象的创建	48
3.6.2 对象的释放	49

3.7 异常处理	49
3.7.1 运行期异常基类.....	50
3.7.2 自定义异常.....	50
3.7.3 异常的触发.....	50
3.7.4 异常处理的捕捉与处理.....	51
第四章 OpenGL 概览	54
4.1 OpenGL 的发展史.....	54
4.2 OpenGL 的基本特点.....	55
4.3 OpenGL 的体系结构.....	56
4.4 OpenGL 的工作流程.....	57
4.5 OpenGL 的函数库.....	58
4.6 OpenGL 的基本图形功能.....	59
4.7 OpenGL 的未来与展望.....	60
第五章 实体建模基础	62
5.1 绘图前的准备工作.....	62
5.1.1 清除窗口和缓存.....	62
5.1.2 指定颜色.....	63
5.1.3 完成绘图.....	64
5.1.4 OpenGL 程序基本结构	65
5.2 OpenGL 数据类型和函数形式	66
5.3 基本几何图元的描述.....	67
5.3.1 齐次坐标.....	67
5.3.2 点的描述.....	67
5.3.3 线的描述.....	68
5.3.4 多边形的描述.....	70
5.4 使用 glBegin()和 glEnd()	72
5.5 法线向量.....	76
5.5.1 法线向量计算.....	76
5.5.2 法线向量定义.....	77
5.6 绘制基本图元实例.....	78
第六章 视图变换	89
6.1 图形变换的数学基础.....	89
6.1.1 平移变换.....	90
6.1.2 缩放变换.....	90

6.1.3 绕坐标轴的旋转变换	90
6.2 从三维空间到二维平面	91
6.2.1 照相机类比法	91
6.2.2 三维图形显示流程	92
6.2.3 绘制立方体	93
6.3 视图变换和模型变换	96
6.3.1 视图变换	96
6.3.2 模型变换	98
6.4 投影变换	102
6.4.1 透视投影	102
6.4.2 正交投影	104
6.5 视区变换	105
6.5.1 定义视区	106
6.5.2 视区变换后的 z 坐标	107
6.5.3 矩阵堆栈操作	107
6.6 附加裁剪平面	111
6.7 视图变换实例	112
第七章 OpenGL 中的颜色	119
7.1 颜色的感知	119
7.1.1 可见光谱与光度学参量	119
7.1.2 颜色的视觉生理和心理规律	120
7.2 计算机中的颜色	121
7.2.1 三基色原理	121
7.2.2 RGB 色立体	122
7.3 RGBA 模式与颜色索引模式	123
7.3.1 RGBA 模式	123
7.3.2 颜色索引模式	125
7.3.3 不同颜色模式的选择原则	127
7.3.4 两种颜色模式的转换	127
7.4 指定阴影模型	127
7.5 颜色使用实例	129
第八章 光照	133
8.1 真实世界与 OpenGL 光照	134
8.1.1 光照分量	135
8.1.2 材质颜色	135

8.1.3 光照与材质属性值	136
8.2 光照球体实例分析	136
8.2.1 为物体顶点定义法线向量	138
8.2.2 创建、定位和激活光源	138
8.2.3 选择光照模型	138
8.2.4 定义物体的材质属性	138
8.3 创建光源	139
8.3.1 颜色	140
8.3.2 定位与衰减	140
8.3.3 聚光灯	141
8.3.4 多光源	142
8.3.5 控制光源位置和方向	143
8.4 选择光照模型	144
8.4.1 全局环境光	144
8.4.2 局部和无穷远视点	144
8.4.3 双面光照	145
8.4.4 激活光源	145
8.5 定义材质属性	145
8.5.1 漫射和环境反射	146
8.5.2 镜面反射	147
8.5.3 发射光颜色	147
8.5.4 改变材质属性	147
8.6 光照的数学计算	155
8.6.1 材质的发射光	156
8.6.2 放大的全局环境光	156
8.6.3 光源的贡献	156
8.6.4 衰减因子	156
8.6.5 聚光灯效果	156
8.6.6 环境光项	157
8.6.7 散射光项	157
8.6.8 镜面反射光项	157
8.6.9 分离镜面颜色	157
8.7 颜色索引模式下的光照	158
第九章 混合、反走样、雾	159
9.1 混合	159
9.1.1 源因子与目的因子	159

9.1.2 混合的应用	161
9.1.3 使用深度缓存的混合	162
9.2 反走样	166
9.2.1 点和直线的反走样	168
9.2.2 多边形的反走样	171
9.3 雾	171
9.3.1 雾的使用	172
9.3.2 雾方程	172
第十章 显示列表	178
10.1 显示列表的基本概念	178
10.2 显示列表的创建与执行	179
10.3 显示列表的嵌套使用	182
10.4 显示列表的管理	182
10.5 多个显示列表的操作	183
10.6 利用显示列表改变封装模式	185
第十一章 位图、字符和图像	187
11.1 位图	187
11.1.1 光栅位置	188
11.1.2 绘制位图	189
11.2 字符	190
11.2.1 字符的分类	190
11.2.2 字符和显示列表	191
11.2.3 一个完整字符集的定义和使用	192
11.3 图像	196
11.3.1 像素读写	197
11.3.2 图像的缩放	198
11.3.3 像素拷贝	199
第十二章 纹理映射	201
12.1 纹理映射的基本步骤	202
12.2 指定纹理	202
12.2.1 一维纹理的定义	203
12.2.2 二维纹理的定义	203
12.3 纹理控制	206
12.3.1 纹理的过滤	206

12.3.2 细节等级.....	207
12.3.3 映射方式	211
12.4 纹理坐标.....	211
12.4.1 指定纹理坐标.....	212
12.4.2 重复和截取纹理.....	212
12.4.3 纹理坐标的自动生成.....	214
第十三章 帧缓存与动画.....	219
13.1 帧缓存.....	219
13.1.1 帧缓存的分类及其用途.....	219
13.1.2 帧缓存的基本操作.....	220
13.2 像素段的测试与操作.....	222
13.2.1 剪取测试.....	222
13.2.2 Alpha 测试	222
13.2.3 模板测试.....	223
13.2.4 深度测试.....	225
13.2.5 混合、抖动与逻辑操作	226
13.3 积累缓存.....	228
13.3.1 场景反走样.....	228
13.3.2 运动模糊度.....	229
13.3.3 景深控制.....	229
13.4 双缓存动画.....	232
第十四章 求值程序和非均匀有理 B 样条	240
14.1 求值程序.....	240
14.1.1 一维求值程序.....	241
14.1.2 二维求值程序.....	245
14.2 NURBS 曲线和曲面	252
14.2.1 简单的 NURBS 范例.....	252
14.2.2 曲面的修剪.....	255
第十五章 选择和反馈.....	257
15.1 选择.....	257
15.1.1 选择模式执行步骤.....	257
15.1.2 创建命名堆栈.....	258
15.1.3 选中记录.....	259
15.1.4 选择应用例程.....	260

15.1.5 拾取.....	263
15.1.6 使用选择机制的一些提示.....	267
15.2 反馈.....	269
15.2.1 反馈模式的执行步骤.....	269
15.2.2 反馈数组.....	270
15.2.3 反馈应用实例.....	270
第十六章 OpenGL 高级应用技巧	274
16.1 跳动的火焰.....	274
16.2 三维树木.....	281
16.3 纹理综合应用.....	285
参考文献.....	293

第一章 图形图像开发现状

【引言】

科学计算可视化、计算机动画和虚拟现实技术是当代计算机图形学的三个热点问题,而这三个热点问题的核心都是三维真实感图形的绘制。随着计算机图形学应用领域的不断扩大,出现了多种图形图像开发工具和图形库工业标准,本章简要地介绍了这一发展现状和趋势。

本章主要包括以下内容:

- 计算机图形学的发展概况,介绍了其发展历程,主要的研究内容,以及广泛的应用领域。
- OpenGL 图形工业标准,简单介绍了 OpenGL 的概念、原理及其在图形图像开发中的重要地位,OpenGL 已经成为引领图形图像开发潮流的主要开发工具。
- 开发工具的选择,讨论了目前应用较为广泛的几种开发工具的特点以及利用 Delphi+OpenGL 作为图形图像开发工具的优势。

1.1 蓬勃发展的计算机图形学

1950 年,第一台图形显示器作为美国麻省理工学院(MIT)旋风 I 号(Whirlwind I)计算机的附件诞生了。该显示器用一个类似于示波器的阴极射线管(CRT)来显示一些简单的图形。1958 年,美国 Calcomp 公司由联机的数字记录仪发展成为滚筒式绘图仪,GerBer 公司把数控机床发展成为平板式绘图仪。在 20 世纪 50 年代,只有电子管计算机,主要应用于科学计算,为这些计算机配置的图形设备仅具有输出功能。计算机图形学处于准备和酝酿时期,并称之为“被动式”图形学。到 20 世纪 50 年代末,MIT 的林肯实验室在“旋风”计算机上开发 SAGE 空中防御体系,第一次使用了具有指挥和控制功能的 CRT 显示器,操作者可以用笔在屏幕上指出被确定的目标。与此同时,类似的技术在设计和生产过程中也陆续得到了应用,它预示着交互式计算机图形学的诞生。

1962 年,MIT 林肯实验室的 Ivan E. Sutherland 发表了一篇题为“Sketchpad:一个人机交互通信的图形系统”的博士论文,他在论文中首次使用了“计算机图形学”(Computer Graphics)这个术语,证明了交互计算机图形学是一个可行的、有用的研究领域,从而确定了计算机图形学作为一个崭新的科学分支的独立地位。

计算机图形学作为计算机科学与技术学科的一个独立分支已经历了近 40 年的发展历程。一方面,作为一个学科,计算机图形学在图形基础算法、图形软件与图形硬件三方面取得了长足的进步,成为当代几乎所有科学和工程技术领域用来加强信息理解和传递的技术和工具。另一方面,计算机图形学的硬件和软件本身已发展成为一个巨大的产业,1996 年总产值达 500 亿美元,预计到 2006 年将达到近 2 000 亿美元。因此,当前全世界从事计算机图形学研究、应

用和产业的队伍十分庞大,这也是为什么每年参加 IG-GRAPH 年会的人数多达 3~4 万人的理由。

计算机图形学主要是研究图形(图像)的计算机生成,其研究方向众多。在图形基础研究方面可归纳为两个主要方向,即建模(Modeling)技术(又称“造型技术”)和绘制(Rendering)技术。建模技术又可分为两大分支,即计算机辅助几何设计和自然景物建模。计算机辅助几何设计追求建模的精确度、可靠性和建模的速度;自然景物建模追求建模的逼真度和速度。

计算机图形学中的绘制技术是指基于光栅图形显示技术的“真实感图形”绘制技术,包括各种光照模型、明暗(Shading)处理和纹理生成等内容。绘制技术追求的是真实感(逼真度)和绘制速度。

综合上述两大研究方向的追求目标可以看出,计算机图形学研究水平的高低就是反映在“真实感”和“速度”的高低以及两者的结合上,也就是既要逼真地反映客观世界的对象,又能高速地、通常又称“实时”地绘制它们。众所周知,“真实感”与“实时性”是一对尖锐的矛盾,如何解决这一矛盾是当代计算机图形学工作者奋斗的目标。

随着计算机图形学的迅速发展,其应用领域也越来越广泛,在工程应用的许多领域中,计算机仿真技术的实现使得人们可以不必再到真实试验场,便可以获得许多系统模型。涉及图形图像处理的领域如下:

(1)计算机辅助设计与制造:CAD/CAM 是计算机图形学在工业界最广泛、最活跃的应用领域,对所设计的系统、产品和工程的相关图形进行人-机交互设计和修改,经过反复的迭代设计,便可利用结果数据输出零件表、材料单、加工流程和工艺卡,或者是数据加工代码。

(2)可视化:科学技术的迅猛发展,数据量的与日俱增使得人们对数据的分析和处理变得越来越困难,人们难以从数据海洋中得到最有用的数据,找到数据的变化规律,提取数据最本质的特征。但是,如果能将这些数据用图形形式表示出来,常常会使问题迎刃而解。目前,科学计算可视化广泛应用于医学、流体力学、有限元分析、气象分析当中。

(3)图形实时绘制与自然景物仿真:运用计算机图形学有关原理可在计算机中重现真实世界场景,即可以模拟真实物体的物理属性,包括物体的形状、光学性质、表面的纹理和粗糙程度,以及物体间的相对位置、遮挡关系等等。

(4)计算机动画:随着计算机图形学和计算机硬件的不断发展,人们已经不满足于仅仅生成高质量的静态场景,于是计算机动画就应运而生。事实上计算机动画也只是生成一幅幅静态的图像,但是每一幅都是对前一幅做一小部分修改,这样,当这些画面连续播放时,整个场景就动起来了。

(5)用户接口:用户接口是人们使用计算机的第一观感,一个友好的图形化的用户界面能够大大提高软件的易用性。在 DOS 时代,计算机的易用性很差,编写一个图形化的界面要花费大量的劳动,过去软件中有 60%的程序是用来处理与用户接口有关的问题和功能的。进入 20 世纪 80 年代后,随着 Xwindow 标准的面世,苹果公司图形化操作系统的推出,特别是微软公司 Windows 操作系统的普及,标志着图形学已经全面融入计算机的方方面面。如今在任何一台普通计算机上都可以看到图形学在用户接口方面的应用。操作系统和应用软件中的图形、动画比比皆是,程序直观易用。很多软件几乎可以不看任何说明书,而根据它的图形、或动画界面的指示进行操作。

1.2 OpenGL 引领图形图像开发的潮流

我们生活在一个充满三维物体的三维世界中,为了使计算机能精确地再现这些物体,就必须能在三维空间描绘这些物体;我们又生活在一个充满信息的世界中,能否尽快地理解并运用这些信息将直接影响事业的成败,所以我们需要用一种最直接的形式来表示这些信息。

最近几年计算机图形学的发展使得三维表现技术得以形成,这些三维表现技术使我们能够再现三维世界中的物体,能够用三维形体来表示复杂的信息,这种技术就是可视化(Visualization)技术。可视化技术使人们能够在三维图形世界中直接对具有形体的信息进行操作,和计算机直接交流。这种技术已经把人和机器的力量以一种直觉而自然的方式加以统一,这种革命性的变化无疑将极大地提高人们的工作效率。可视化技术赋予人们一种仿真的、三维的并且具有实时交互的能力,这样人们可以在三维图形世界中用以前不可想像的手段来获取信息或发挥自己创造性的思维。机械工程师可以从二维平面图中得以解放直接进入三维世界,从而很快得到自己设计的三维机械零件模型。医生可以从病人的三维扫描图像分析病人的病情。军事指挥员可以面对用三维图形技术生成的战场地形,指挥具有真实感的三维飞机、军舰、坦克向目标开进并分析战斗方案的效果。

更令人惊奇的是目前正在发展的虚拟现实技术,它能使人们进入一个三维的、多媒体的虚拟世界,人们可以游历远古时代的城堡,也可以遨游浩瀚的太空。所有这些都依赖于计算机图形学、计算机可视化技术的发展。人们对计算机可视化技术的研究已经经历了一个很长的过程,而且形成了许多可视化工具,其中 SGI 公司推出的 GL 三维图形库表现突出,易于使用而且功能强大。利用 GL 开发出来的三维应用软件颇受许多专业技术人员的喜爱,这些三维应用软件已涉及建筑、产品设计、医学、地球科学、流体力学等领域。随着计算机技术的继续发展,GL 已经进一步发展成为 OpenGL,OpenGL 已被认为是高性能图形和交互式视景处理的标准,目前包括 AT&T 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 公司在内的几家在计算机市场占主导地位的大公司都采用了 OpenGL 图形标准。

值得一提的是,由于 Microsoft 公司在 Windows NT 中提供 OpenGL 图形标准,OpenGL 将在微机中广泛应用,尤其是 OpenGL 三维图形加速卡和微机图形工作站的推出,人们可以在微机上实现三维图形应用,如 CAD 设计、仿真模拟、三维游戏等,从而更有机会、更方便地使用 OpenGL 及其应用软件来建立自己的三维图形世界。

OpenGL 是 Open Graphics Library 的缩写,它是 SGI 公司开发的一套高性能图形处理系统,是图形硬件的软件界面,意为开放的图形库。通过 OpenGL 程序员可以创建交互式的应用程序,实现具有逼真效果的三维图形图像,从而在要求高度模拟真实世界的诸多领域中大显身手。由于其开放性和高度的可重用性,目前已成为业界标准。OpenGL 被设计成独立于硬件,独立于窗口系统,在运行各种操作系统的计算机上都可用,并能在网络环境下以客户/服务器模式工作,是专业图形处理、科学计算等高端应用领域的标准图形库。

长期以来,从事三维图形开发的技术人员都不得不在自己的程序中编写矩阵变换、外部设备访问等函数,为调制这些与自己的软件开发目标关系并不十分密切的函数费脑筋,而 OpenGL 正是提供一种直观的编程环境,通过提供一系列函数大大地简化了三维图形程序。例如:

(1)OpenGL 提供一系列的三维图形单元供开发者调用。

(2)OpenGL 提供一系列的图形变换函数。

(3)OpenGL 提供一系列的外部设备访问函数,使开发者可以方便地访问鼠标、键盘、空间球、数据手套等。这种直观的三维图形开发环境体现了 OpenGL 的技术优势,这也是许多三维图形开发者热衷于 OpenGL 的缘故所在。

许多计算机公司已经把 OpenGL 集成到各种窗口和操作系统中,其中操作系统包括 UNIX,Windows NT,DOS 等,窗口系统有 X 窗口、Windows 等。为了实现一个完整功能的图形处理系统,设计一个与 OpenGL 相关的系统结构为:其最底层是图形硬件,第二层为操作系统,第三层为窗口系统,第四层为 OpenGL,第五层为应用软件。OpenGL 是网络透明的,在客户-服务器(Client - Server)体系结构中,OpenGL 允许本地和远程绘图。所以在网络系统中,OpenGL 在 X 窗口、Windows 或其他窗口系统下都可以以一个独立的图形窗口出现。

OpenGL 作为一个性能优越的图形应用程序设计界面(API)而适合于广泛的计算环境,从个人计算机到工作站和超级计算机,OpenGL 都能实现高性能的三维图形功能。由于许多在计算机界具有领导地位的计算机公司纷纷采用 OpenGL 作为三维图形应用程序设计界面,所以 OpenGL 应用程序具有广泛的移植性。因此,OpenGL 已成为目前的三维图形开发标准,是从事三维图形开发工作的技术人员所必须掌握的开发工具。

1.3 开发工具的选择

选择一个高效的开发工具将会有事半功倍的效果,因此选择一种高效的开发工具显得尤为重要。OpenGL 作为一种图形工业标准,几乎所有的应用程序开发工具都支持 OpenGL 的集成,如 Delphi,VC++,C++ Builder,Java 等程序设计语言。本书选择 Delphi+OpenGL 来开发图形图像应用程序,是基于以下几个方面考虑的。

首先是要有一个简洁易用的可视化开发环境。总的来说,Delphi 的编辑器和其他工具的编辑器类似,但它的 CodeInsight 技术却省去了许多输入工作的麻烦。这一技术是建立在编译器信息之上的,而不是基于像 Visual Basic 等使用的类型库,因此应用范围更为广泛。Delphi还支持在调试时随意放置和停靠窗口并把这一状态保存为命令的桌面设置。由此,Delphi 的 IDE 实现了对调试功能的良好支持。Delphi 的窗体设计器的与众不同之处在于它是建立在一个真正面向对象的框架结构基础之上的。这样,你对基类所做的改变都将会传递给所有的派生类。这里涉及的一项关键技术就是 VFI(Visual Form Inheritance),即可视化窗体继承。VFI 技术使你能够动态地继承当前项目或对象库中的任何其他窗体。一旦基窗体发生改变,派生的窗体会立即予以更新。

其次是编译器的效率。快速的编译器可以使你逐步递进地开发软件,经常地修改源代码、重新编译、测试、再修改、再编译、再测试……形成这样一个良好的开发循环。如果编译速度很慢,开发者就不得不分批地修改代码,每次编译前进行多处修改以适应一个低效率的循环过程。提高运行效率,节约运行时间,生成的二进制代码更为短小,其优越性是不言而喻的。

也许 Pascal 编译器最显著的特点就是速度快,而 Delphi 正是建立在这种编译器的基础之上的。事实上,它可能是针对 Windows 的最快的高级语言本地代码编译器。以往速度很慢的 C++编译器近年来取得了很大的进步,增加了链接和各种缓存策略,尤其是在 Visual C++