

<<CAD应用程序开发详解>>

图书基本信息

书名：<<CAD应用程序开发详解>>

13位ISBN编号：9787121179709

10位ISBN编号：7121179709

出版时间：2012-9

出版时间：电子工业出版社

作者：王清辉，李静蓉 编著

页数：421

字数：691000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<CAD应用程序开发详解>>

前言

笔者长期从事计算机辅助设计（CAD）系统的研究与开发工作，将自己从事CAD系统设计与开发的经验提炼和总结并汇集成书，一直是笔者的心愿。

2003年，编著了《Visual C++ CAD应用程序开发技术》一书。

该书出版后，读者发来电邮100余封，给予了积极评价，并交流了在各自专业领域从事CAD开发的一些技术问题。

同时，该书所附带光盘中的程序也在国内软件开发的主要论坛被广泛转载，成为国内OpenGL及CAD开发的主要参考资料之一，并被国内科技论文作为参考文献广泛引用。

随着近年来CAD技术的发展，以及近年来与读者的交流与体验，笔者感觉有必要在原书的基础上，对该书的内容进行重要的扩充与修改，并增加更多新的内容。

恰逢电子工业出版社来我校组织约稿，遂将上述想法付诸实施。

CAD软件开发是工程软件开发中的一个重要领域。

开发一个三维CAD软件所涉及的知识面很多，如何规划和展开软件的开发工作是系统开发成功与否的关键问题之一。

目前，单纯讲解OpenGL技术、Visual C++使用，以及面向对象的编程技术的出版物较多，本书的特点在于，从系统开发的角度，将开发一个基于OpenGL显示的三维CAD软件所涉及的各项技术环节逐步展现给读者，将计算机图形学、OpenGL、面向对象的软件开发技术三个方面的知识予以融会贯通，将许多技术难点、抽象的概念融入具体的开发实例中讲解，着重讲解系统开发过程中的实际问题，易于读者领会，具有新颖性和强的实用性。

在讲解中，以一个三维CAD原形系统STLViewer的开发为主线，贯穿全书，并提供详细的代码注解。

Visual C++是Microsoft公司迄今开发的功能最为强大的软件开发工具，是新一代CAD软件的主要开发平台。

随着面向对象程序设计技术广泛应用、Visual C++优秀的开发环境、Microsoft Foundation Class（MFC）以及MFC程序框架、Windows操作系统对OpenGL的支持等，为在Windows系统上开发三维CAD软件提供了极大的方便。

读者通过本书的学习，将能够循序渐进地了解 and 掌握使用Visual C++开发三维图形及CAD应用软件的相关技术。

通过对应用实例的具体剖析，希望能使读者从软件开发的思维方法上对面向对象的程序设计技术有更深入的了解。

本书内容共分10章，分别介绍基于MFC的总体程序框架的分析与设计；功能模块的划分以及相关DLL库的开发与使用；CAD的基础几何类库的开发；在MFC环境下使用OpenGL进行图形绘制；开发面向CAD应用程序的OpenGL通用绘图类；使用面向对象技术设计CAD软件的几何内核；曲面的数学描述及OpenGL绘制；CAD软件的图形交互；及软件界面设计等。

本书所附带的光盘中，根据每章讲述的内容提供了全部的实现代码。

本书的代码适用于Visual C++ 2005以上版本。

本书的内容是作者长期从事CAD系统设计与开发的一些经验的提炼和总结。

在内容或方法上若有疏漏和不妥之处，恳请各位读者给予指正。

参加本书编写工作的还有华南理工大学的熊巍、阎汉生、黄仲辉、张小明等，在此向他们表示感谢。

作者

<<CAD应用程序开发详解>>

内容概要

本书系统地阐述了在MFC与OpenGL API的集成开发环境下，用面向对象的技术开发三维CAD软件的有关知识与方法，包括：总体程序框架的分析与设计、功能模块的划分，以及相关DLL库的开发与使用、CAD基础几何类库的开发、在MFC环境下使用OpenGL进行图形绘制、开发面向CAD应用程序的OpenGL通用绘图类、使用面向对象技术设计CAD软件的几何内核、CAD软件的图形交互、软件界面设计等。在介绍过程中，以一个完整的三维CAD软件（STLViewer）为例，将其开发过程贯穿于全书各章节，并附有完整的Visual C++实现代码。全书面向开发实例进行分析与介绍，讲解透彻、易于理解。读者通过对本书的阅读和学习，能够掌握使用Visual C++进行具有一定复杂程度的软件的设计与实现方法。本书适合从事图形和CAD软件开发的技术人员，以及具有一定Visual C++基础的大专院校相关专业学生阅读。

书籍目录

第1章 三维CAD软件的系统框架分析

1.1 关于微软基础类MFC

1.2 MFC应用程序的文档/视图结构

1.2.1 文档/视图结构概述

1.2.2 文档与多个视图的关联

1.2.3 文档模板及主要组成类

1.3 实例分析：三维CAD模型浏览软件STLViewer

1.4 STLViewer的程序框架

1.5 STLViewer中类的层次设计及软件模块结构划分

1.6 建立和使用动态链接库

1.6.1 动态链接库的基本概念

1.6.2 基于MFC的动态链接库

1.6.3 查看执行程序EXE与DLL库的层次关系

附：本章相关程序使用说明

第2章 矢量计算工具与几何计算类库设计

2.1 矢量运算概述

2.1.1 点

2.1.2 矢量

2.1.3 齐次坐标与齐次变换矩阵

2.2 设计点、矢量和齐次变换矩阵类

2.2.1 点类CPoint3D

2.2.2 矢量类CVector3D

2.2.3 变换矩阵类CMatrix3D

2.3 三维图形的几何变换

2.3.1 三维齐次变换矩阵

2.3.2 平移变换

2.3.3 旋转变换/绕空间任意轴的旋转变换函数的实现

2.3.4 几何缩放

2.3.5 对称变换/沿空间任意平面的对称变换函数的实现

2.4 矢量计算及其工具函数集设计

2.4.1 判断两个空间矢量是否平行

2.4.2 计算两个矢量的夹角

2.4.3 计算三角面片的面积

2.4.4 正交投影及点到直线之间的距离计算

2.4.5 直线求交

2.4.6 创建过平面三个点的圆

2.5 设计几何基本工具库GeomCalc.dll

2.5.1 GeomCalc.dll中的输出类与输出函数

2.5.2 创建几何基本工具库GeomCalc.dll的步骤

2.5.3 使用GeomCalc.dll

2.6 有关源程序代码

2.6.1 文件CadBase.h

2.6.2 文件CadBase.cpp

2.6.3 文件CadBase1.cpp

附：本章相关程序及使用说明

<<CAD应用程序开发详解>>

第3章 创建基于MFC的OpenGL Windows程序

3.1 OpenGL概述

3.2 在Windows环境下使用OpenGL

3.2.1 OpenGL的函数库

3.2.2 获取与安装OpenGL

3.2.3 OpenGL与GDI

3.2.4 渲染场境

3.2.5 像素格式

3.3 OpenGL MFC应用程序创建实例

3.3.1 创建一个应用程序框架

3.3.2 修改视类CGLView

3.3.3 使用OpenGL的双缓存技术为应用程序增加动画效果

3.4 程序清单

3.4.1 文件GLView.h

3.4.2 文件GLView.cpp

附：本章相关程序使用说明

第4章 设计封装OpenGL功能的C++类

4.1 封装OpenGL的C++类的设计

4.2 照相机类GCamera的设计

4.2.1 视点坐标系和视图变换

4.2.2 投影变换与视景物

4.2.3 视口变换

4.2.4 设计照相机类GCamera

4.3 类COpenGLDC

4.4 修改类CGLView

4.5 运行应用程序

4.6 源程序清单

4.6.1 类GCamera的声明代码

4.6.2 类GCamera的实现代码

4.6.3 类COpenGLDC的声明代码

4.6.4 类COpenGLDC的实现代码

4.6.5 类CGLView的声明代码

4.6.6 类CGLView的实现代码

附：本章相关程序使用说明

第5章 设计基于OpenGL的CAD图形工具库

5.1 创建动态链接库glContext.dll

5.2 照相机类GCamera的功能增强

5.2.1 选择典型的观察视图

5.2.2 景物平移

5.2.3 景物缩放

5.2.4 使用OpenGL的选择模式

5.3 模型真实感渲染与OpenGL光照设置

5.3.1 光照的组成

5.3.2 定义OpenGL的光源

5.3.3 物体材质

5.3.4 单面光照与双面光照

5.4 类COpenGLDC功能的增强

<<CAD应用程序开发详解>>

- 5.4.1 实现和Windows窗口的关联
- 5.4.2 有关光源的操作
- 5.4.3 有关颜色的操作
- 5.4.4 图形绘制函数
- 5.4.5 选择模式
- 5.5 增加类CGLView中的功能
- 5.6 glContext类的输出和调用
- 5.7 源程序清单
 - 5.7.1 文件Camera.h (类GCamera)
 - 5.7.2 文件Camera.cpp (类GCamera)
 - 5.7.3 文件OpenGLDC.h (类COpenGLDC、CGLView)
 - 5.7.4 文件OpenGLDC.cpp (类OpenGLDC)
 - 5.7.5 文件GLView.cpp (类CGLView)
- 附：本章相关程序使用说明
- 第6章 设计几何内核模块
 - 6.1 CAD实体模型的常用表示方法
 - 6.1.1 边界表示法与体素构造法简介
 - 6.1.2 多边形网格实体建模的概念
 - 6.1.3 立体光造型 (STL) 文件格式
 - 6.2 几何对象类的设计
 - 6.2.1 类的层次设计
 - 6.2.2 几何对象基本类CEntity
 - 6.2.3 三角形面片对象类CTriChip
 - 6.2.4 STL几何模型类CSTLModel
 - 6.2.5 高级几何模型类CPart
 - 6.3 串行化 (Serialize) 实现文档存取功能
 - 6.3.1 为什么要使用串行化
 - 6.3.2 CArchive类
 - 6.3.3 串行化类的设计步骤
 - 6.3.4 CObArray的Serialize()函数
 - 6.3.5 应用程序的文档串行化实例剖析
 - 6.4 利用多态性实现CAD模型的绘制
 - 6.4.1 虚拟函数与多态性
 - 6.4.2 利用多态性统一CAD元素的绘制接口
 - 6.4.3 纯虚拟函数
 - 6.4.4 实现CPart模型的OpenGL显示
 - 6.5 建立几何内核库GeomKernel.dll
 - 6.6 程序清单
 - 6.6.1 文件Entity.h
 - 6.6.2 文件Entity.cpp
 - 附：本章相关程序使用说明
- 第7章 CAD应用程序的模块化实现
 - 7.1 网格模型可视化程序STLViewer的解决方案
 - 7.1.1 网格模型可视化程序STLViewer的模块结构
 - 7.1.2 创建应用程序的解决方案
 - 7.1.3 为解决方案配置目录与路径
 - 7.2 设计应用程序框架

<<CAD应用程序开发详解>>

- 7.2.1 增加界面资源
- 7.2.2 修改框架类CMainFrame
- 7.2.3 修改文档类CSTLViewerDoc
- 7.2.4 修改视图类CSTLViewerView
- 7.3 运行STLViewer.exe
 - 7.3.1 输入STL模型
 - 7.3.2 存储STLViewer自己的文档 (*.mdl)
 - 7.3.3 模型显示缩放
 - 7.3.4 使用键盘平移场景
 - 7.3.5 模型视图切换
 - 7.3.6 模型着色模式切换
- 7.4 源程序清单
 - 7.4.1 文件MainFrm.h
 - 7.4.2 文件MainFrm.cpp
 - 7.4.3 文件STLViewerDoc.h
 - 7.4.4 文件STLViewerDoc.cpp
 - 7.4.5 文件STLViewerView.h
 - 7.4.6 文件STLViewerView.cpp
- 附：本章相关程序使用说明
- 第8章 曲线和曲面的OpenGL绘制及类设计
 - 8.1 关于曲面造型技术
 - 8.2 曲线和曲面的参数表示
 - 8.2.1 曲线的非参数表示
 - 8.2.2 曲线的参数表示
 - 8.2.3 插值与逼近
 - 8.3 曲线的样条描述
 - 8.3.1 参数样条曲线的基本概念
 - 8.3.2 Bezier曲线、曲面的数学表示
 - 8.3.3 使用OpenGL绘制Bezier曲线
 - 8.3.4 使用OpenGL绘制Bezier曲面
 - 8.3.5 NURBS曲线的数学表示
 - 8.3.6 使用OpenGL绘制NURBS曲线
 - 8.3.7 使用OpenGL绘制NURBS曲面
 - 8.4 在应用程序中集成曲线和曲面功能
 - 8.4.1 添加Bezier曲线功能
 - 8.4.2 添加Bezier曲面功能
 -
- 第9章 CAD应用程序的界面设计
- 第10章 三给CAD图形交互功能的实现
- 参考文献

章节摘录

版权页：插图：GCamera的功能类似于与一个照相机，可称之为照相机类。

GCamera中定义了视口大小、投影变换和一个视点的位置和观察方向。

这如同使用一架相机取景，使用者需要确定相机的位置和角度、取景范围，最后才将景物投影于胶片上的过程。

这些操作实际上定义了OpenGL中一系列的变换。

在三维空间中，由于变换是图形制作和显示环节中最关键的问题之一，物体摆放的位置、方向以及动画的实现都依靠变换来实现。

通过变换，OpenGL将三维对象投影到二维屏幕上。

同时，OpenGL中的变换还使得用户可以对图形进行平移、旋转和缩放。

这些变换分别通过对视点变换、模型变换、投影变换和视口变换等操作来最终实现。

GCamera就是这样一个定义和操作变换的类。

具体到CAD应用中，视图的放大、缩小、旋转、平移以及在三维空间中的导航（Navigation）的实现，都归结于对变换的操作。

COpenGLDC作为一个封装OpenGL功能的C++类，封装了在MFC下OpenGL的环境设置，即一个渲染场境（Rendering Context），以及图形绘制的相关函数。

它的内容包括：OpenGL和Windows窗口的关联、光照和颜色、取景操作、场景绘制等几方面。

当然，可以根据不同的应用需要在类COpenGLDC中增加新的内容，这一点在本书第5章中还将具体论述。

COpenGLDC中包含了一个GCamera的对象，用于取景操作。

对OpenGL的操作可在类COpenGLDC中实现。

因而在MFC的窗口类插入一个COpenGLDC的对象，并使它与窗口关联，就可以使用COpenGLDC进行对OpenGL的操作和图形绘制。

和上一章中创建的CGLView不同的是，这里的CGLView把与OpenGL相关的代码分离了出来，在类中包括了一个COpenGLDC的对象，与OpenGL相关的操作通过对COpenGLDC的调用予以实现。

作为一个视类，CGLView本身的代码将集中于处理和分发用户和视图窗口的交互信息，如对视图的放大、缩小、视角的变换、旋转、平移以及鼠标的拖动和物体的捡取等操作。

这样划分的优点在于代码更具有独立性，能够提高代码的重复利用率，也便于对类进行维护和功能扩充。

下面将具体介绍这几个类的设计与实现，通过这几个类的设计，也可以对OpenGL的一些相关概念有一个深入的了解。

<<CAD应用程序开发详解>>

编辑推荐

《CAD应用程序开发详解:Visual C++与OpenGL综合应用》适合从事图形和CAD软件开发的技术人员, 以及具有一定Visual C++基础的大专院校相关专业学生阅读。

<<CAD应用程序开发详解>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>