

说明：

- 1、报告中的任务书、进度表由指导教师在课程设计开始前填写并发给每个学生；四、五两项（中英文摘要）由学生在完成综合设计后填写。
- 2、学生成绩由指导教师根据学生的设计情况给出各项分值及总评成绩。
- 3、指导教师评语一栏由指导教师就学生在整个设计期间的平时表现、设计完成情况、报告的质量及答辩情况，给出客观、全面的评价。
- 4、所有学生必须参加课程设计的答辩环节，凡不参加答辩者，其成绩一律按不及格处理。答辩小组成员应由2人及以上教师组成。
- 5、报告正文字数一般应不少于3000字，也可由指导教师根据本门综合设计的情况另行规定。
- 6、平时表现成绩低于6分的学生，取消答辩资格，其本项综合设计成绩按不及格处理。

计算机科学系课程设计任务书

学生姓名	王雅宁	学号	14102080210	专业班级	2008级软件工程2班
设计题目	旋转立方体的设计				
内容概要:	画出一个正方体, 并将它旋转起来。				
文献资料:	<ol style="list-style-type: none">1. 谭浩强 C 程序设计 (M) (第三版) 清华大学出版社 20102. 严蔚敏.吴伟民 数据结构[M] 高等教育 20043. 钱能 C++程序设计教程 (M) (第二版) 清华大学出版社 20094. 郑丽 Visual Basic 程序设计清华大学出版社 2009-10-29				
设计要求:	<ol style="list-style-type: none">1. 用 C/C++或 VB实现;2. 友好性: 界面要友好;3. 可读性: 源程序代码清晰、有层次;4. 当鼠标离屏幕中心的位置远近可以决定立方体旋转速度的快慢。				
工作期限:	设计工作自 2010 年 5 月 31 日至 2010 年 6 月 11 日止。				

指导教师: _____

系主任: _____

日期: 2010 年 5 月 28 日

计算机科学系课程设计进度安排表

专业：软件工程

姓名：王雅宁

学号：14102080210

日期	内 容	备注
5月31日 ~ 6月1日	分析题目，查阅相关资料	
6月2日 ~ 6月3日	拟定整体方案，编写程序	
6月4日 ~ 6月6日	程序编写	
6月7日 ~ 6月8日	调试程序	
6月9日 ~ 6月10日	整理报告	
6月11日 ~ 6月11日	答辩	
月 日 ~ 月 日		
月 日 ~ 月 日		
月 日 ~ 月 日		
月 日 ~ 月 日		
月 日 ~ 月 日		
月 日 ~ 月 日		
月 日 ~ 月 日		
月 日 ~ 月 日		
月 日 ~ 月 日		
月 日 ~ 月 日		

指导教师签名： _____

2010年5月28日

指导教师评语

指导教师：_____（签字）

日期：2010年6月10日

答辩记录表

学生姓名：_____ 学号：_____ 班级：_____

答辩地点：_____

答辩内容记录：

	合计 分值	各项 分值	评分标准	实际 得分	合计 得分	备注
答辩 成绩	25	10	在规定时间内能就所设计的内容进行阐述，言简意明，重点突出，论点正确，条理清晰。			
		15	在规定时间内能准确、完整、流利地回答教师所提出的问题。			

答辩小组成员（签字）：

_____ 2010 年 6 月 11 日

成绩评定表

学生姓名：_____ 学号：_____ 班级：_____

类别	合计分值	各项分值	评分标准	实际得分	合计得分	备注
平时表现	10	10	按时参加设计指导，无违反纪律情况。			
完成情况	30	20	按设计任务书的要求完成了全部任务，能完整演示其设计内容，符合要求。			
		10	能对其设计内容进行详细、完整的介绍，并能就指导教师提出的问题进行正确的回答。			
报告质量	35	10	报告文字通顺，内容翔实，论述充分、完整，立论正确，结构严谨合理；报告字数符合相关要求，工整规范，整齐划一。			
		5	课题背景介绍清楚，综述分析充分。			
		5	设计方案合理、可行，论证严谨，逻辑性强，具有说服力。			
		5	符号统一；图表完备、符合规范要求。			
		5	能对整个设计过程进行全面的总结，得出有价值的结论或结果。			
		5	参考文献数量在2篇以上，格式符合要求，在正文中正确引用。			
答辩情况	25	10	在规定时间内能就所设计的内容进行阐述，言简意明，重点突出，论点正确，条理清晰。			
		15	在规定时间内能准确、完整、流利地回答教师所提出的问题。			

总评成绩：_____ 分

指导教师：_____ (签字)

日期：2010年6月15日

摘 要

软件工程专业经过两年的课程学习,已经积累了相关高级语言程序设计的基本知识。画出正方体并使之旋转,这一题目考察了程序设计自顶而下、逐步细化的相关基本思想。题目难点在于实现可视化界面的交互、以及图像的旋转过程。分析题目可以得出程序可以大致分为两个方面的内容:

- 1.画出一个正方体。
- 2.使正方体旋转起来。

经过查找相关资料,本程序采用功能强大的图形库 **OPENGL** 并调用一系列 **WINDOWS API** 采用 **C/C++**语言在 **Microsoft Visual C++6.0** 环境下编写。很好的解决了用 **C/C++**编写图形界面的难点问题。通过 **Win32** 平台的 **OPENGL API** 实现了逼真的三维旋转效果

关键词: 正方体; 旋转; 可视化; **OPENGL**

Abstract

Software Engineering after two years course of study, has accumulated a relevant high-level language programming basics. Draw the cube and make it spin, this subject examines the design top-down process, gradually refining the basic idea related.

Subject difficulty is to achieve interactive visual interface, and image rotation process. Analysis procedures can be drawn subject can be broadly divided into two aspects:

1. Draw a cube.
2. To make a cube spin.

After find relevant information, this program features a powerful graphics library OPENGL and call a WINDOWS API using C / C + + language in Microsoft Visual C + +6.0 environment preparation.

Good solution using C / C + + graphical interface written and difficult problems. OPENGL API through the Win32 platform to achieve a realistic three-dimensional rotation effect

Keywords : cube; spin; visual; OpenGL

录

目 录	I
第一章 课题背景.....	
1.1 引言	
背景及目的	
1.2 题目概述	
1.2.1 问题及技术要求.....	
1.2.2 理论依据及工作内容.....	
第二章 设计简介及设计方案论述.....	
2.1 设计简介	
2.2 OPENGL 图形库简介.....	
2.2.1 OPENGL 历史简介.....	
2.2.2 OPENGL 特点.....	
2.2.3 OPENGL 工作顺序.....	
2.3 OPENGL 简单编程方法.....	
2.3.1 OPENGL 基本语法.....	
2.3.2 OPENGL 状态机制.....	
2.3.3 OPENGL 基本结构.....	
2.4 程序所使用的 OPENGL 库函数	
2.5 OPENGL 及 WINDOWS 坐标系.....	
2.6 WINDOWS 下配置 OPENGL 环境	
2.7 WINDOWS 消息机制.....	
2.8 大体设计方案	
第三章 详细设计.....	
3.1 OPENGL 的绘制工作.....	
3.1.1 OPENGL 的初始化工作.....	
3.1.2 OPENGL 的主体绘制工作.....	
3.2 鼠标控制正方体的旋转.....	11
3.3 重置 OPENGL 窗口.....	
3.4 WINDOWS 中 OPENGL 窗体设置及创建.....	
3.5 完成 WINDOWS 主程序.....	
第四章 设计结果及分析.....	
4.1 设计结果分析.....	
4.2 进一步美化程序.....	
总 结	
致 谢	
参考文献	
附录	

课题背景

引言

随着专业课程学习的深入，相关高级语言程序设计的知识已经有所积累，如何更好的掌握并应用已经成为当务之急。通过软件工程课程设计的过程能够学习程序设计的基本思想以及整体流程。

背景及目的

基本高级语言程序设计以及相关专业基础课程已经学习完毕，在学习过程中遇到的一些难点问题，为了综合以前学过的课程知识并融会贯通。可以通过课程设计的独立完成来通过相关途径解决，从而能够更好的掌握程序设计的思想，提升编程能力。

1.2 题目概述

1.2.1 问题及技术要求

本题：画出一个正方体，并将它旋转起来。主要应实现以下 点基本要求：

1. 用 C/C++或 VB实现；
2. 友好性：界面要友好；
3. 可读性：源程序代码清晰、有层次；
4. 当鼠标离屏幕中心的位置远近可以决定立方体旋转速度的快慢。

1.2.2 理论依据及工作内容

题目难点在于实现可视化界面的交互、以及图像的旋转过程。分析题目可以得出程序可以大致分为两个方面的工作内容：1.画出一个正方体。2.使正方体旋转起来。通过在互联网上查找资料，我决定使用 OPENGL 图形库来解决 C/C++编写图形界面困难的问题。

设计简介及设计方案论述

设计简介

为了实现本程序的两大功能，计划采用 **OPENGL** 图形库并调用一系列 **WINDOWS API** 采用 **C/C++** 语言编写。

首先，应熟悉 **OPENGL** 在 **WIN32** 平台下的相关 **API**，以及其余 **WINDOWS** 窗口交互的相关接口方法，来构建窗口的内容。

其次，熟悉了解 **OPENGL** 库函数在窗体中实现绘图（既绘制正方体），实现正方体旋转功能的相关函数。

最后，实现 **OPENGL** 与 **WINDOWS** 的交互的过程，完成程序及注释。

2.2 OPENGL 图形库简介

历史简介

OpenGL 是 **SGI (Silicon Graphics Inc.)** 公司对 **IRIS GL** 进行改进，扩展可移植性，形成的一个跨平台开放式图形编程接口，利用它可以很方便地开发出有多种特殊视觉效果(如光照、纹理、透明、阴影)的三维图形。由于 **OpenGL** 采用类似 **C** 语言的语法，所以对于学过 **C** 语言的人来说使用起来特别方便。

2.2.2 OPENGL 特点

从程序开发人员的角度来看，**OpenGL** 是一组绘图命令的 **API** 集合。利用这些 **API** 能够方便地描述二维和三维几何物体，并控制这些物体按某种方式绘制到显示缓冲区中。**OpenGL** 的 **API** 集合提供了物体描述、平移、旋转、缩放、光照、纹理、材质、象素、位图、文字、交互以及提高显示性能等方面的功能，基本涵盖了开发二、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** 具有以下几个突出特点：

- (1)应用广泛
- (2)跨平台性
- (3)高质量和高性能
- (4)出色的编程特性
- (5)网络透明性

OpenGL 的工作顺序就是一个从定义几何要素到把象素段写入帧缓冲区的过程。在屏幕上显示图象的主要步骤是以下 3 步：

- 1) 构造几何要素(点、线、多边形、图像、位图), 创建对象的数学描述。在三维空间放置对象, 选择有利的观察点。
- 2) 计算对象的颜色, 这些颜色可能直接定义, 或由光照条件及纹理间接给出。
- 3) 光栅化, 把对象的数学描述和颜色信息转换到屏幕的象素。

简单编程方法

2.3.1 OPENGL 基本语法

OpenGL 基本函数均使用 `gl` 作为函数名的前缀, 如 `glClearColor()`; 实用函数则使用 `glu` 作为函数名的前缀, 如 `gluSphere()`。OpenGL 基本常量的名字以 `GL_` 开头, 如 `GL_LINE_LOOP`; 实用常量的名字以 `GLU_` 开头, 如 `GLU_FILL`。一些函数如 `glColor*()`(定义颜色值), 函数名后可以接不同的后缀以支持不同的数据类型和格式。如 `glColor3b(□)`、`glColor3d(□)`、`glColor3f(□)`和 `glColor3bv(□)`等, 这几个函数在功能上是相似的, 只是适用于不同的数据类型和格式, 其中 3 表示该函数带有三个参数, `b`、`d`、`f` 分别表示参数的类型是字节型、双精度浮点型和单精度浮点型, `v` 则表示这些参数是以向量(数组)形式出现的。

OpenGL 还定义了一些特殊的类型名, 如 `GLfloat`, `GLvoid`。它们其实就是 C 中的 `float` 和 `void`。在 `gl.h` 文件中可以看到以下定义:

```
typedef float GLfloat;
```

```
typedef void GLvoid;
```

```
□□
```

一些基本的数据类型都有类似的定义。

2.3.2 OPENGL 状态机制

OpenGL

置，这些状态或模式在重新改变它们之前一直有效。例如，当前颜色就是一个状态变量，在这个状态改变之前，绘制的每个象素都将使用该颜色，直到当前颜色被设置为其它颜色为止。OpenGL 中大量地使用了这种状态机制，如颜色模式、投影模式、单双显示缓存区的设置、背景色的设置、光源的位置和特性等等。

基本结构

OpenGL 程序的基本结构可分为三个部分：

第一部分是初始化部分，主要是设置一些 OpenGL 的状态开关，如颜色模式(RGBA 或 ALPHA 等)的选择，是否作光照处理(若有的话，还需设置光源的特性)，深度检验，裁剪等等。这些状态一般都用函数 `glEnable()`，`glDisable()`来设置，() 中为相应的状态。

第二部分设置观察坐标系下的取景模式和取景框位置及大小。主要利用了三个函数：

1. 函数 `void glViewport(left, top, right, bottom)`: 设置在屏幕上的窗口大小，四个参数描述屏幕窗口四个边界坐标(以象素表示)；
2. 函数 `void glOrtho(left, right, bottom, top, near, far)`: 设置投影方式为正交投影(平行投影)，其取景体积(观察体)是一个各面均为矩形的六面体；
3. 函数 `void gluPerspective(fovy, aspect, zNear, zFar)`: 设置投影方式为透视投影，其取景体积(观察体)是一个截头锥体(棱台)，在这个体积内的物体投影到锥体的顶点。

第三部分是 OpenGL 的主要部分，使用 OpenGL 的库函数构造几何物体对象的数学描述，包括点线面的位置和拓扑关系，几何变换，光照处理等等。

程序所使用的 OPENGL 库函数

1. OpenGL 函数库分为以下几类：

- | | |
|---|---------|
| (1)OpenGL 核心库: <code>gl</code> | 115 个函数 |
| (2)OpenGL 编程辅助库: <code>aux</code> | 31 个函数 |
| (3)OpenGL 实用程序工具包 (OpenGL utility toolkit, GLUT): <code>glut</code> | 30 多个函数 |
| (4)Windows 专用库: <code>wgl</code> | 6 个函数 |

(5)OpenGL 实用库(GLU)

利用较低层 OpenGL 命令编写一些执行特殊任务的例程，如纹理映射、坐标变换、NURBS 曲线曲面等。GLU 库函数前缀都用 `glu`。

(6)OpenGL 的 X Window 系统扩充(GLX)

在使用 X Window 系统的机器上，提供一种建立 OpenGL 现场(context)，并把它与可绘(draw able)窗口关联起来的方法。GLX 作为 OpenGL 的附件提

GLX 库函数前缀为 glx。

(7)OpenGL Programming Guide 辅助库(Auxiliary Library)

这个库建立了一系列简单而又较完整的编程例子，例如初始化窗口、监控输入，以及绘制一些三维几何体等函数。辅助库函数用前缀 aux。在 Windows95/98 中，相关的库以动态链接库的形式存在，opengl32.lib、glu32.lib、glaux.lib 分别表示 OpenGL 库，实用库和辅助库，和它们相应的头文件是 gl.h、glu.h 及 glaux.h。

2. 基本常用函数

初始化 (glutInit)

创建窗口 (glutCreateWindow)

设定窗口的显示模式 (glutInitDisplayMode)

指定窗口的位置和大小 (glutInitWindowPosition 和 glutInitWindowSize)

指定窗口的显示内容函数 (glutDisplayFunc)

运行框架 (glutMainLoop)

刷新窗口的缓冲区

```
glClear(GL_COLOR_BUFFER_BIT);
```

设定投影参数(将三维图形投影到显示窗口)

```
glMatrixMode(GL_PROJECTION);
```

及 WINDOWS 坐标系

OPENGL 的三维坐标系如图 2-1 所示：

XOY 平面为屏幕所在

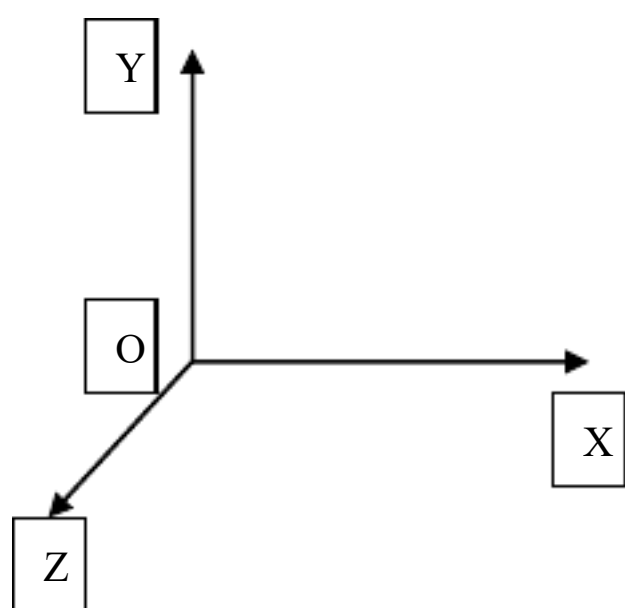


图 2-1 OPENGL 三维坐标系

WINDOWS 的窗体坐标如图 2-2 所示：

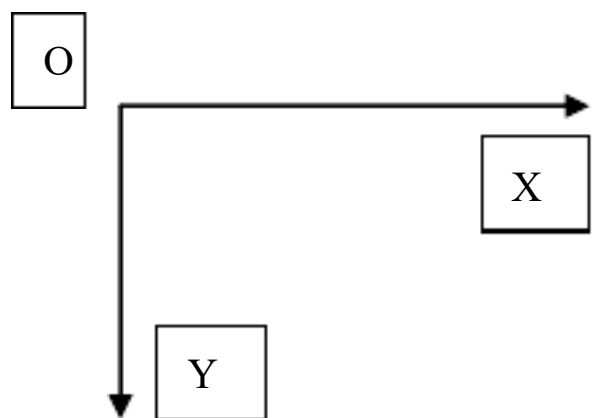


图 2-2 WINDOWS 窗体二维坐标系

2.6 WINDOWS 下配置 OPENGL 环境

第一步：选择一个编译环境

现在 Windows 系统的主流编译环境有 Visual Studio, Borland C++ Builder, Dev-C++等，它们都是支持 OpenGL 的。但这里我选择 VC++ 6.0 作为 OpenGL 的环境。

第二步：安装 GLUT 工具包

GLUT 不是 OpenGL 所必须的，但它会给我们的学习带来一定的方便，推荐安装。Windows 环境下的 GLUT 下载地址：（大小约为 150k）
<http://www.opengl.org/resources/libraries/glut/glutdlls37beta.zip>
 Windows 环境下安装 GLUT 的步骤：

1、将下载的压缩包解开，将得到 5 个文件

2、以我的安装目录为例：

（1）“ ”文件夹”。把解压得到的 glut.h 放到这个 GL 文件夹里。没有 GL 文件夹可以自己建一个，一般都有的。

（2）“ ”文件夹”。把解压得到的 glut.lib 和 glut32.lib 放到静态函数库所在文件夹，即 lib 文件夹。

（3）把解压得到的 glut.dll 和 glut32.dll 放到操作系统目录下面的 system32 文件夹内。（典型的位置为： ）这是非常重要的动态链接库设置！

2.7 WINDOWS 消息机制

消息机制是 Windows 应用程序的核心，在 Windows 中发生的一切都可以用消息来表示，消息用于告诉操作系统发生了什么，所有的 Windows 应用程序都是消息驱动的，除了 WM_COMMAND 消息，所有以 WM_ 为前缀的消息都是标准的 Windows 消息，如窗口、鼠

标移动、窗口大小改变等，程序启动或退出甚至每一段固定的时间都会产生标准 Windows 消息。

Windows 消息控制中心一般是三层结构，其顶端就是 Windows 内核。Windows 内核维护着一个消息队列，第二级控制中心从这个消息队列中获取属于自己管辖的消息，后做出处理，有些消息直接处理掉，有些还要发送给下一级窗体(Window)或控件(Control)。第二级控制中心一般是各 Windows 应用程序的 Application 对象。第三级控制中心就是 Windows 窗体对象，每一个窗体都有一个默认的窗体过程，这个过程负责处理各种接收到的消息。

1、消息的组成：

一个消息由一个消息名称 (UINT)，和两个参数 (WPARAM, LPARAM)。当用户进行了输入或是窗口的状态发生改变时系统都会发送消息到某一个窗口。例如当菜单转中之后会有 WM_COMMAND 消息发送，WPARAM 的高字中 (HIWORD(wParam)) 是命令的 ID 号，对菜单来讲就是菜单 ID。当然用户也可以定义自己的消息名称，也可以利用自定义消息来发送通知和传送数据。

2、谁将收到消息：

一个消息必须由一个窗口接收。在窗口的过程 (WNDPROC) 中可以对消息进行分析，对自己感兴趣的消息进行处理。例如你希望对菜单选择进行处理那么你可以定义对 WM_COMMAND 进行处理的代码，如果希望在窗口中进行图形输出就必须对 WM_PAINT 进行处理。

3、未处理的消息到那里去了：

MS 为窗口编写了默认的窗口过程，这个窗口过程将负责处理那些你不处理消息。正因为有了这个默认窗口过程我们才可以利用 Windows 的窗口进行开发而不必过多关注窗口各种消息的处理。例如窗口在被拖动时会有很多消息发送，而我们都可以不予理睬让系统自己去处理。

4、窗口句柄：

说到消息就不能不说窗口句柄，系统通过窗口句柄来在整个系统中唯一标识一个窗口，发送一个消息时必须指定一个窗口句柄表明该消息由那个窗口接收。而每个窗口都会有自己的窗口过程，所以用户的输入就会被正确的处理。例如有两个窗口共用一个窗口过程代码，你在窗口一上按下鼠标时消息就会通过窗口一的句柄被发送到窗口一而不是窗口二。

一个消息从产生到被一个窗口响应，其中有 5 个步骤：

- 1) 系统中发生了某个事件。
- 2) Windows 把这个事件翻译为消息，然后把它放到消息队列中。
- 3) 应用程序从消息队列中接收到这个消息，把它存放在 TMsg 记录中。
- 4) 应用程序把消息传递给一个适当的窗口的窗口过程。
- 5) 窗口过程响应这个消息并进行处理。

步骤 3 和 4 构成了应用程序的消息循环。消息循环往往是 Windows 应用程序的核心，因为消息循环使一个应用程序能够响应外部的事件。消息循环的任务就是从消息队列中检索消息，然后把消息传递给适当的窗口。如果消息队列中没有消息，Windows 就允许其他应用程序处理它们的消息。Windows 操作系统最大的特点就是其图形化的操作界面，其图形化界面是建立在其消息处理机制这个基础之上的

2.8 大体设计方案

综上所述，在本程序中。首先使用相关的 WINDOWS API 创建窗口，其次实现程序的相应功能：

1. 使用 OPENGL 相应库函数画出正方体。
2. 使之旋转起来。
3. 在窗体中响应鼠标事件，控制正方体的旋转。

程序流程图如图 2-3 所示：

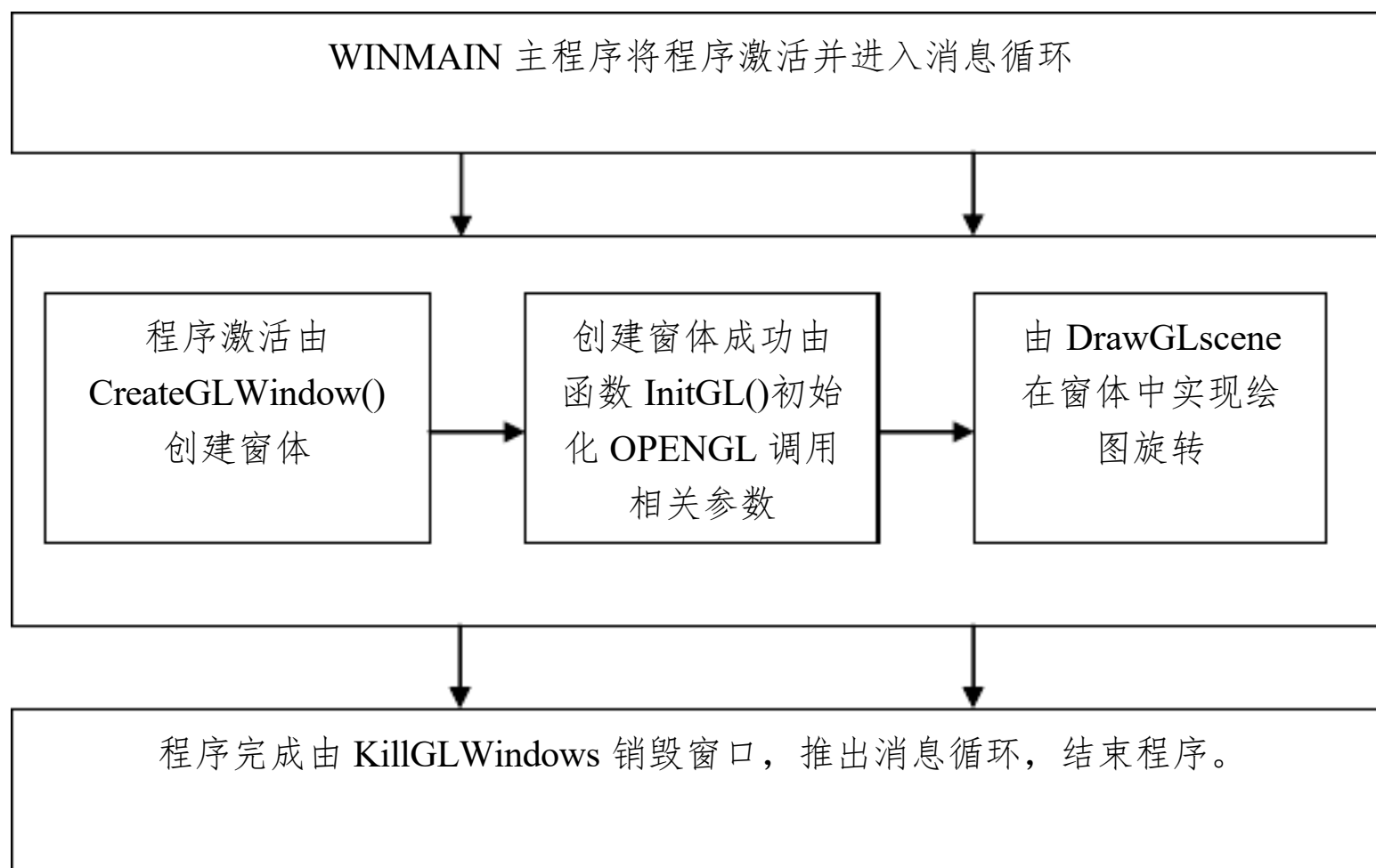


图 2-3 程序流程图

程序的大体内容以及机构已经构造成型，功能部分还需要进一步的细化。显然在消息循环中的内容还远不止这些，还需加入响应以下事件：

1. 当窗口大小发生变化时，重置窗口。
2. 响应鼠标事件，并改变相对应的参数值，来改变旋转速度。
3. 响应窗口中断事件，如关闭事件、屏保事件。

第三章 详细设计

3.1 OPENGL 的绘制工作

上一章我们提到完成 OPENGL 的绘制工作主要有两个主要方面的工作需要完成：

1. OPENGL 的初始化。
2. OPENGL 的绘制工作，其中包括了是正方体旋转。

下面我们分别来完成。

3.1.1 OPENGL 的初始化工作

通过查找资料 OPENGL 的初始化工作由如下库函数过程来完成

```
glShadeModel(GL_SMOOTH); // 启用阴影平滑
启用 smooth shading( 阴影平滑)。阴影平滑通过多边形精细的混合色彩，并对外部光进行平滑。
```

```
glClearColor(0.0f, 0.0f, 0.0f, 0.5f); // 黑色背景
色彩值的范围从 0.0f 到 1.0f。0.0f 代表最黑的情况，1.0f 就是最亮的情况。
```

glClearColor 后的第一个参数是 Red Intensity(红色分量), 第二个是绿色, 第三个是蓝色。最大值也是 1.0f, 代表特定颜色分量的最亮情况。

接下来的三个函数必须做的是关于 depth buffer(深度缓存)的。将深度缓存设想为屏幕后面的层。深度缓存不断的对物体进入屏幕内部有多深进行跟踪。我们本节的程序其实没有真正使用深度缓存，但几乎所有在屏幕上显示 3D 场景 OpenGL 程序都使用深度缓存。它的排序决定那个物体先画。这样您就不会将一个圆形后面的正方形画到圆形上来。深度缓存是 OpenGL 十分重要的部分。

```
glClearDepth(1.0f); // 设置深度缓存
glEnable(GL_DEPTH_TEST); // 启用深度测试
glDepthFunc(GL_LEQUAL); // 所作深度测试的类型
```

接下来是透视图修饰使透视图好看些：

```
glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST); // 真正精细的透视修正
```

3.1.2 OPENGL 的主体绘制工作

首先使用 `glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);` 清除屏幕和深度缓存

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/358014050117007010>