学号 E51814014 专业 数字媒体技术 姓名 吴振龙

实验日期 **11.25**  教师签字 成绩

实验报告

【实验名称】 基本的OpenGL编程

【实验目的】

1. 熟悉OpenGL的应用工具包GLUT的安装和使用。

2. 掌握基本的OpenGL编程方法。

3. 掌握显示窗口的显示方法。

【实验内容】

1. 练习基本的OpenGL编程方法。

（1）配置OpenGL，能在VC下通过编译。

（2）阅读并运行simple.c，熟悉最简单的OpenGL程序结构。

（3）阅读并运行改进的simple2.c，掌握OpenGL程序的通用结构。给simple2.c中的函数加注释。

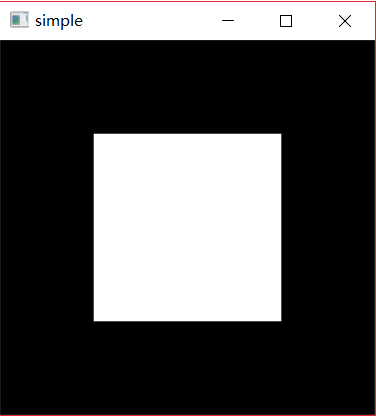
2. 可尝试做如下修改：

（a）改变窗口的位置，尺寸，标题；

（b）改变窗口背景色和矩形颜色；

（c）改变矩形的尺寸和其在窗口中的位置，还可以只显示矩形的一部分（通过设置正投影参数（glOrtho函数的参数）和矩形四个顶点的坐标来完成）。

3.学会查文档，自学。

Simple.c程序运行截图：

Simple2.c运行结果与上图相同，

改变glVertex2f()函数的参数修改图形的顶点坐标，修改glClearColor()参数的(R,G,B,a)来改变背景色，修改glColor3f()参数改变图形的填充颜色，

glVertex2f(-0.3, -0.5)

glVertex2f(-0.5, 0.5)

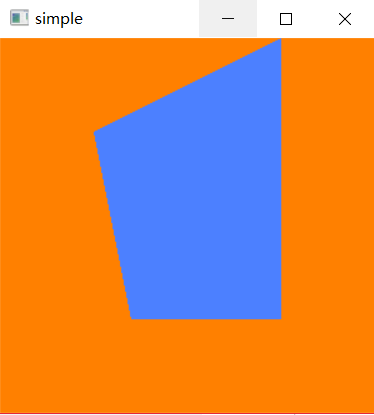
glVertex2f(0.5, 1)

glVertex2f(0.5, -0.5)

glClearColor(1.0, 0.5, 0.0, 0.5)

glColor3f(0.3, 0.5, 1.0)

效果如下：



【小结或讨论】

1、设置图形显示模式

|  |
| --- |
| void glutInitDisplayMode(unsigned int mode); |

设置图形显示模式，参数mode可选值有:

* GLUT\_RGBA:当未指明GLUT\_RGBA或GLUT\_INDEX时, 是使用的默认模式. 表示欲创建RGBA模式的窗口;
* GLUT\_RGB：与GLUT\_RGBA作用类似；
* GLUT\_INDEX：颜色索引模式；
* GLUT\_SINGLE：只使用单缓存；
* GLUT\_DOUBLE：使用双缓存。以避免把计算机动画作图的过程表现出来，实现平滑的显示动画；
* GLUT\_ACCUM：让窗口使用累加的缓存；
* GLUT\_ALPHA：让颜色缓存区使用alpha组件；
* GLUT\_DEPTH：使用深度缓存；
* GLUT\_STENCIL：使用模板缓存；
* GLUT\_MULTISAMPLE：让窗口支持多例程；
* GLUT\_STEREO：让窗口支持立体；
* GLUT\_LUMINACE：luminance是亮度的意思。在多数OpenGL平台上，不支持该参数。

2、设置窗口大小、位置和名称

|  |
| --- |
| void glutInitWindowSize(int width, int height);  void glutInitWindowPosition(int x, int y);  int glutInitWindow(char\* name) |

位置以屏幕左上为原点，像素为单位。

3、指定顶点

|  |
| --- |
| glVertex\* ()系列  void glVertex2d()  void glVertex2f()  void glVertex3f()  void glVertex3fv() |

函数名中的数字代表参数的个数，字母代表参数类型，其中v表示指针类型。

4、设置当前颜色

|  |
| --- |
| void glColor3f(GLfloatred, GLfloatgreen,GLfloat blue); |

5、设置背景颜色

|  |
| --- |
| void glClearColor(GLfloat red, GLfloat green, GLfloat blue, GLfloat alpha); |

6、清除屏幕颜色

|  |
| --- |
| void glClear(GLbitfield mask); |

mask 对指定需要清除的缓存进行按位或屏蔽操作，mask可选值如下：

|  |  |
| --- | --- |
| **mask** | **说明** |
| GL\_COLOR\_BUFFER\_BIT | 指定当前被激活为写操作的颜色缓存 |
| GL\_DEPTH\_BUFFER\_BIT | 指定深度缓存 |
| GL\_ACCUM\_BUFFER\_BIT | 指定累加缓存 |
| GL\_STENCIL\_BUFFER\_BIT | 指定模板缓存 |

7、开始一段gl程序

|  |
| --- |
| void glBegin(Glenum mode); |

mode可选参数有：

* GL\_POINTS：把每一个顶点作为一个点进行处理，顶点n即定义了点n，共绘制N个点
* GL\_LINES：把每一个顶点作为一个独立的线段，顶点2n－1和2n之间共定义了n条线段，总共绘制N/2条线段
* GL\_LINE\_STRIP：绘制从第一个顶点到最后一个顶点依次相连的一组线段，第n和n＋1个顶点定义了线段n，总共绘制n－1条线段
* GL\_LINE\_LOOP：绘制从第一个顶点到最后一个顶点依次相连的一组线段，然后最后一个顶点和第一个顶点相连，第n和n＋1个顶点定义了线段n，总共绘制n条线段
* GL\_TRIANGLES：把每个顶点作为一个独立的三角形，顶点3n－2、3n－1和3n定义了第n个三角形，总共绘制N/3个三角形
* GL\_TRIANGLE\_STRIP：绘制一组相连的三角形，对于奇数n，顶点n、n＋1和n＋2定义了第n个三角形；对于偶数n，顶点n＋1、n和n＋2定义了第n个三角形，总共绘制N-2个三角形
* GL\_TRIANGLE\_FAN：绘制一组相连的三角形，三角形是由第一个顶点及其后给定的顶点确定，顶点1、n＋1和n＋2定义了第n个三角形，总共绘制N-2个三角形
* GL\_QUADS：绘制由四个顶点组成的一组单独的四边形。顶点4n－3、4n－2、4n－1和4n定义了第n个四边形。总共绘制N/4个四边形
* GL\_QUAD\_STRIP：绘制一组相连的四边形。每个四边形是由一对顶点及其后给定的一对顶点共同确定的。顶点2n－1、2n、2n＋2和2n＋1定义了第n个四边形，总共绘制N/2-1个四边形
* GL\_POLYGON：绘制一个凸多边形。顶点1到n定义了这个多边形。

8、清空缓存区

|  |
| --- |
| void glFlush(void); |

glFlush()清空缓冲区，将指令送往缓硬件立即执行，但是它是将命令传送完毕之后立即返回，不会等待指令执行完毕。这些指令会在有限时间内执行完毕。

个人理解：将gl绘图结果送到硬件显示。