

南京工程学院

实 验 报 告

课 程 名 称 虚拟现实 2020

实验项目名称 基本 OpenGL 应用程序生成

实验学生班级 数嵌 172

实验学生姓名 朱广锋

学 号 202170638

同组学生姓名 无

实 验 时 间 2020.5.18

实 验 地 点

一、 实验主题

利用 VC 集成开发环境，配置 OpenGL 库函数， 建立基本的 OpenGL 应用程序。

二、 实验准备

1. 打开 Visual Studio 并且设置好工作目录；

2. 下载 OpenGL 安装包所需文件

(<http://d.download.csdn.net/down/2560229/ssagnn23>), 主要包括 GL.H GLAUX.H GLU.H glut.h

GLAUX.LIB GLU32.LIB glut32.lib glut.lib OPENGL32.LIB
glaux.dll glu32.dll glut32.dll glut.dll opengl32.dll

3. 复制并配置 OpenGL 库函数到指定的目录 (.h、.lib、.dll 分别放到 MSVC include、lib 和系统 Path 路径如 System32), 检查复制后是否文件已经存在于指定目录下。

三、 主要数据源、库函数、变量、涉及函数及其解释

```
// #include <GL/glut.h>
#include <cstdlib>
void display(void)
{
    /* 清除颜色缓存 */
    glClear(GL_COLOR_BUFFER_BIT);
    /* 绘制一个白色多边形，指定四个顶点的坐标 */
    glColor3f(1.0, 1.0, 1.0);
    glBegin(GL_POLYGON);
    glVertex3f(0.25, 0.25, 0.0);
    glVertex3f(0.75, 0.25, 0.0);
    glVertex3f(0.75, 0.75, 0.0);
```

```

    glVertex3f(0.25, 0.75, 0.0);
    glEnd();
    /* 立即开始处理保存在缓冲区中的OpenGL函数调用。*/
    glFlush();
}
void init(void)
{
    /* 制定清除颜色 */
    glClearColor(0.0, 0.0, 0.0, 0.0);
    /* 设置投影变换方式 */
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0.0, 1.0, 0.0, 1.0, -1.0, 1.0);
}
/*指定窗口的初始大小和位置以及显示模式（单缓存和RGBA模式）
*打开一个标题为hello的窗口
*调用init函数.
*注册显示函数.
* 进入主循环并处理事件*/
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(250, 250);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("hello");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}

```

函数解释：

glutInit(&argc, argv)：使用命令行参数初始化GL库

glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)：初始化显示模式为单缓冲，设置RGB颜色空间

glutInitWindowSize(250, 250)：设置窗口大小为250x250

glutInitWindowPosition(100, 100)：设置窗口位置100,100

glutCreateWindow("hello")：创建标题为hello的窗口

init()：初始化投影方式和视角

glutDisplayFunc(display)：设置显示绘制回调函数

glutMainLoop()：进入opengl消息循环

<p>glClear(GL_COLOR_BUFFER_BIT): 清除颜色缓存</p> <p>glColor3f(1.0, 1.0, 1.0): 指定绘制颜色为白色</p> <p>glBegin(GL_POLYGON): 开始绘制多边形, 将glEnd之前的顶点作为多边形顶点</p> <p>glEnd(): 结束绘制</p> <p>glVertex3f(0.25, 0.25, 0.0): 设置顶点</p> <p>glFlush(): 刷新缓冲</p>

四、 实验任务

任务 1: 建立基本的 OpenGL 应用程序

在 Visual Studio 菜单栏上选择文件->新建->项目, 选择 Win32 应用程序, 输入项目名称 HelloOpenGL

点击“OK”后, 在后续的对话框中选中 Empty project, 然后点击 Finish。

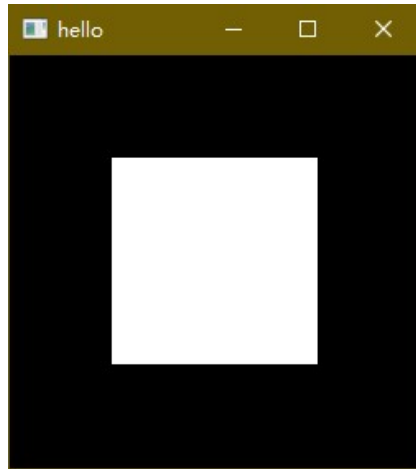
右击项目名 HelloOpenGL, 选择属性(property), 再选择链接器(Linker)中的输入选项(Input), 附加依赖项(Additional Dependencies): opengl32.lib glu32.lib glaux.lib

右击项目名 HelloOpenGL 下的源文件, 选择添加(ADD), 再选择新建项 (New Item), 在弹出来的对话框中选择 C++ File(.cpp), 输入文件名 FirstOpenGL.cpp, 然后点击添加(ADD), 检查源文件是否建立成功。

任务 2: 在 FirstOpenGL.cpp 中输入并测试代码。

五、 主要总结

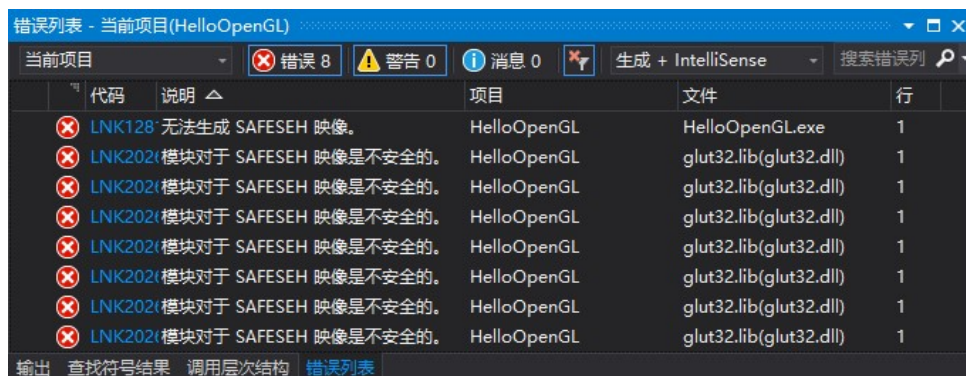
运行结果:



中途遇到错误:

1. 源代码没有引用 `glut.h` 导致编译出错, 添加引用即可(见首行注释部分)。
2. 项目无法生成 SAFESEH 映像。

SafeSEH 是防止攻击者通过覆盖堆栈上的异常处理函数句柄，从而控制程序执行流程的攻击的保护机制。我没有找到出现无法生成映像的具体原因，因为此错误与实验内容无关，所以最后决定关闭映像生成/SAFESEH:NO 来解决问题。



总结：

第一次实验涉及了 OpenGL 库的安装和简单使用，借助实验用到的函数名称便可以理解 OpenGL 库的基本运行流程，并懂得这些函数的含义和大致使用方法，同时可以了解到 OpenGL 基本的一些元素，如颜色、顶点等。代码的逻辑为，初始化并设定好窗口、视角以及回调函数等，进入消息循环展示结果。初始化部分的一些函数之间存在着依赖关系，即不能任意调整其先后顺序。OpenGL 本身是一个状态机，我们只需事先设定好逻辑、管理好输入输出即可方便的切换其状态，来实现各种复杂的效果。

教师评阅：

评阅项目及内容	得分
1. 考勤（10 分）	
2. 实验完成情况（50 分）	
3. 报告撰写内容（40 分）	
合 计	
成绩评定	