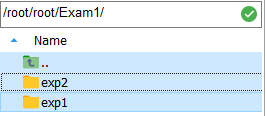
实验报告一

姓名：胡力杭

学号：2019K8009926002

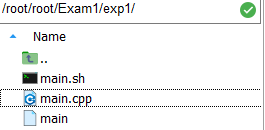
**代码明细**：

整体实验的代码沿用了老师上课讲过的示例代码，就细节部分进行了改动和补充。

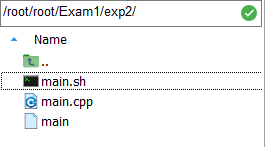


代码和执行脚本放置于/root/Exam1/目录，exp1为平面多边形区域填充实验，exp2为交互控制绘制实验。

exp1目录包含main.cpp、可执行文件main、执行脚本main.sh



exp2目录包含main.cpp、可执行文件main、执行脚本main.sh。



**环境配置：**

无

**程序编译及运行命令：**

exp1：

使用sh main.sh命令进行编译运行

在main.sh中包含编译命令和执行命令，

编译：g++ main.cpp -lglut -lGLU -lGL -o main

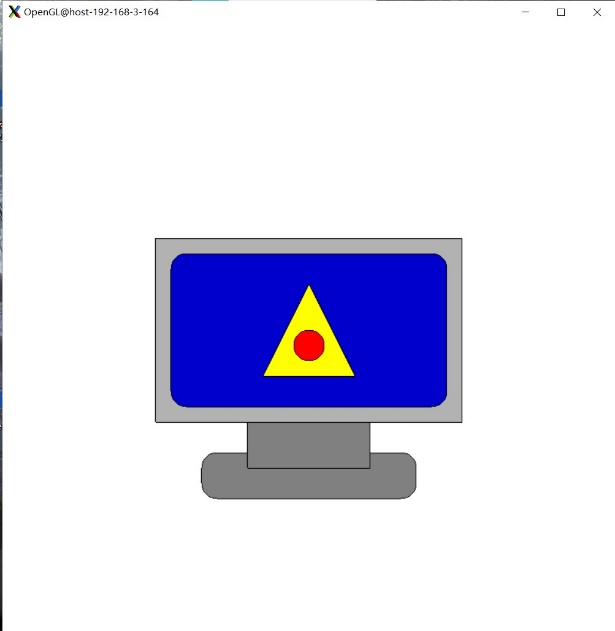
运行：./main

exp2：编译和运行命令完全相同

**实验效果（效果展示+操作说明）：**

exp1：

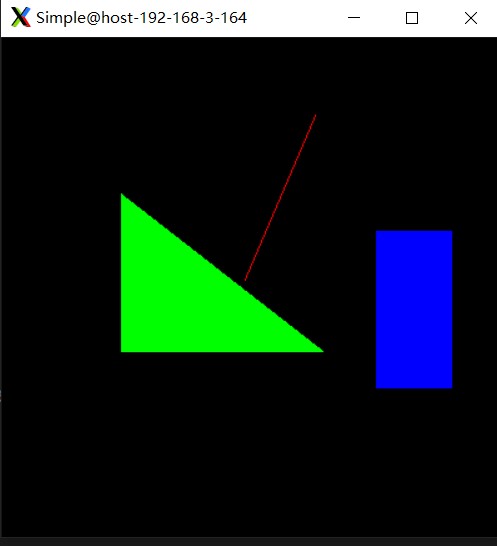
绘制图案：



因为这个电脑是直接画出来的，所以不需要任何的操作

exp2：

鼠标右键开启菜单，设置绘制形状、绘制颜色，鼠标左键拖动绘制图案：



操作说明：

进入界面之后，点击右键，第一个子目录是颜色菜单，第二个子目录是形状菜单，第三个条目是清屏，第四个条目是绘制图像。想要绘制图形时，选择自己需要的颜色和形状，用鼠标左键进行绘制。最多可以绘制16个图形，然后点击清屏可以清除图像。然后再点击画图（draw mode），又可以开始绘制图形。需要注意的是，当进行清屏操作之后，必须点击一次draw mode才能继续绘制。（这是因为使用了mod变量来控制绘图，每次清屏之后mod置为1，只有点击draw mode之后mod才会置零）

注意到，在vs上运行的时候，点击清屏并不会立刻清除图形，而是在下一次绘制的时候清除。但是在mobax上运行的时候，清屏可以正常使用。可能是编译器的差别。但因为作业在mobax上提交，这个问题没有去细究。