



中山大學  
SUN YAT-SEN UNIVERSITY

# 本科生实验报告

实验课程: 计算机图形学

专业名称: 计算机科学与技术

学生姓名: \_\_\_\_\_

学生学号: \_\_\_\_\_

教学班级: 计科 3 班

实验成绩: \_\_\_\_\_

报告时间: 2023 年 11 月 14 日

## 1 环境搭建

### 1.1 Visual Studio 安装

本次实验使用的 VS 为 2022 版本。因版本问题，我们还需要下载 VS 2019 工具集。



fig 1: VS 2019 工具集下载

## 1.2 Qt 5.13.0 安装



fig 2: Qt 5.13.0 下载安装

### 1.3 Qt VS Tools 扩展安装

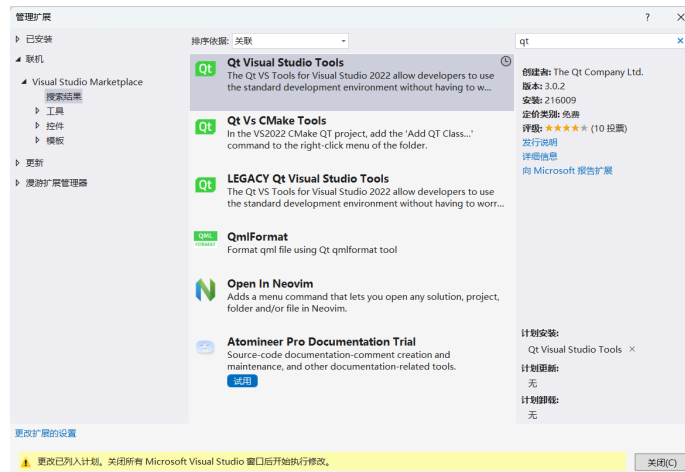


fig 3: Qt VS Tools 下载安装

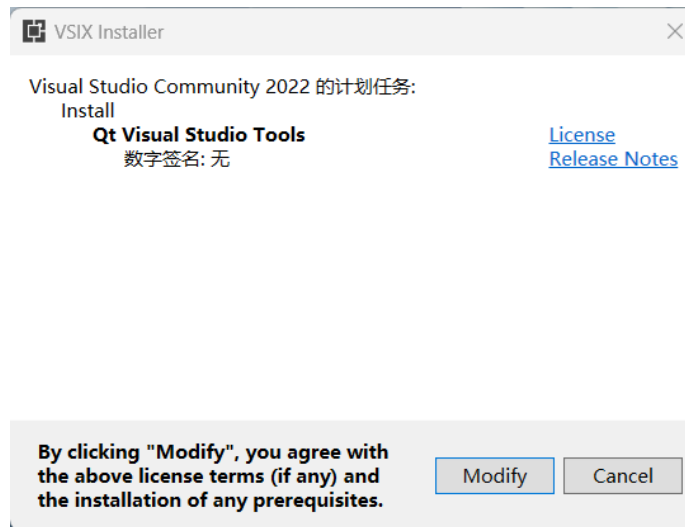


fig 4: 安装完成后关闭 VS 以应用 Qt 扩展

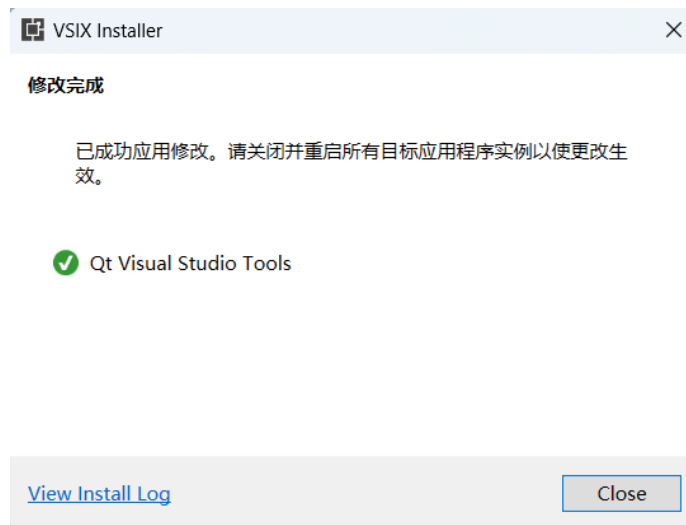


fig 5: 应用 Qt 扩展

## 1.4 配置 Qt 插件

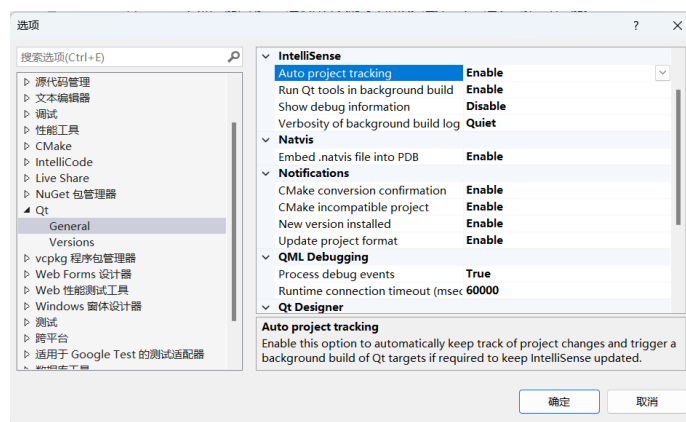


fig 6: 进入扩展中的 Qt VS Tools 点开 Qt Options

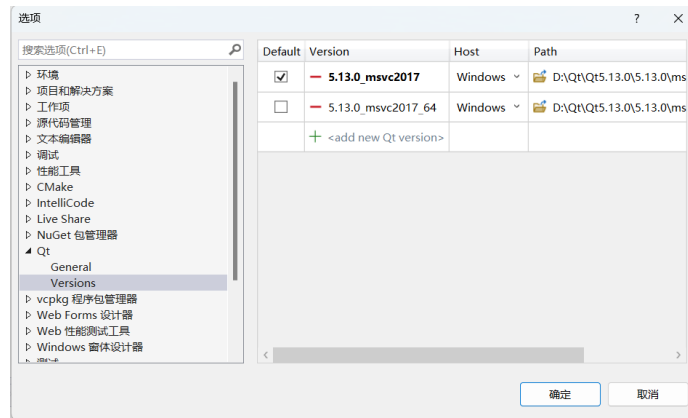


fig 7: 添加 Qt 编辑器的安装路径

## 1.5 环境配置

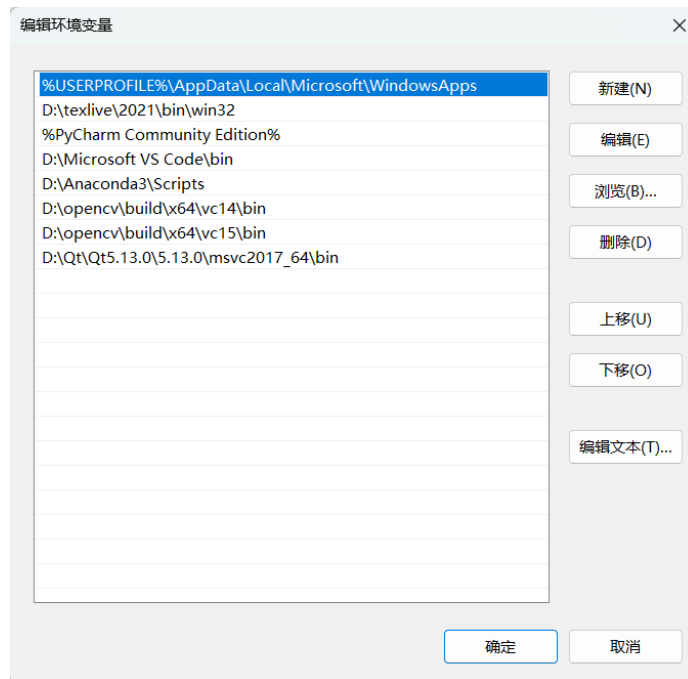
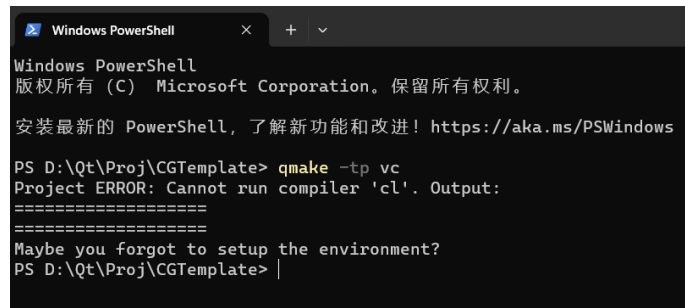


fig 8: 新建系统环境变量

启动 cmd 进入 Demo 文件夹



```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows

PS D:\Qt\Proj\CGTemplate> qmake -tpvc
Project ERROR: Cannot run compiler 'cl'. Output:
=====
Maybe you forgot to setup the environment?
PS D:\Qt\Proj\CGTemplate> |
```

fig 9: 键入命令: qmake -tpvc 生成 VS 工程文件

为解决出现的报错，我们还需要将'cl.exe'加入系统环境变量，其在本机中的路径为

```
1 D:\Visual Studio\Community\VC\Tools\MSVC\14.37.32822\
  bin\Hostx64\x64
```

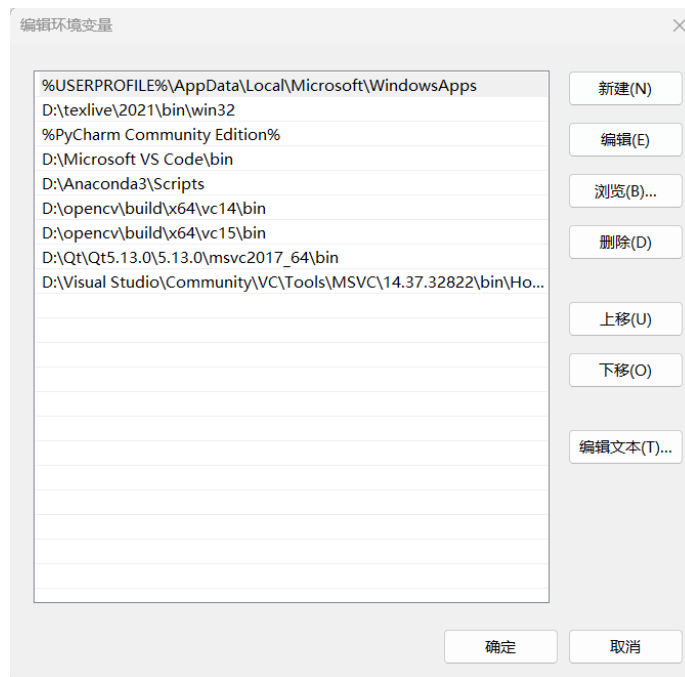
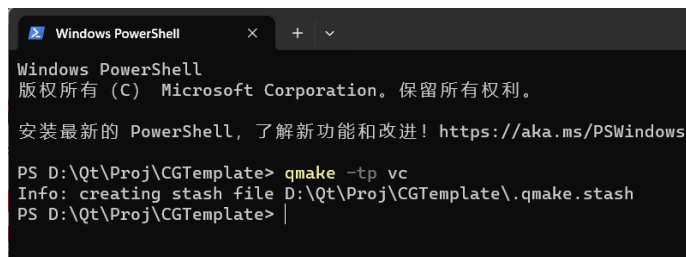


fig 10: 将'cl.exe'加入系统环境变量



```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

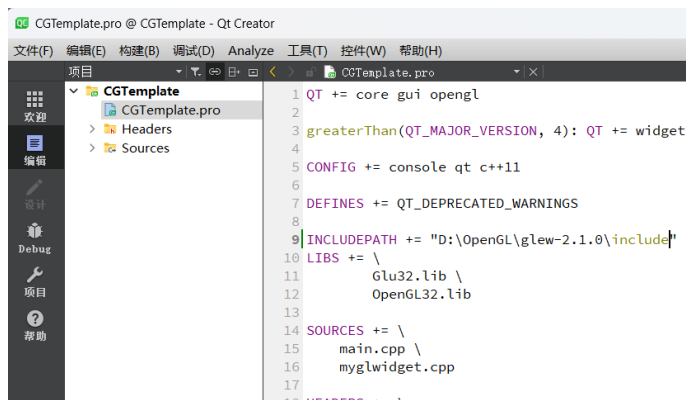
PS D:\Qt\Proj\CGTemplate> qmake -tp vc
Info: creating stash file D:\Qt\Proj\CGTemplate\qmake.stash
PS D:\Qt\Proj\CGTemplate> |
```

fig 11: 再次键入命令：qmake -tpvc 生成 VS 工程文件

## 1.6 Demo 运行

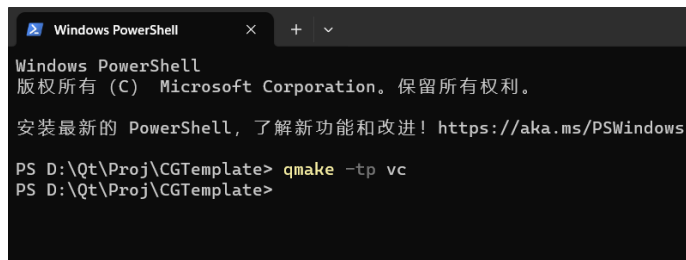
由于头文件 GL/glew.h 的缺失，我们需要先配置 GL/glew.h。在 Qt 中打开 CGTemplate.pro 文件，添加以下代码：

```
1 INCLUDEPATH += "D:\OpenGL\glew-2.1.0\include"
```



```
CGTemplate.pro @ CGTemplate - Qt Creator
文件(F) 编辑(E) 构建(B) 调试(D) Analyze 工具(T) 控件(W) 帮助(H)
项目
  CGTemplate
    CGTemplate.pro
    Headers
    Sources
编辑
设计
Debug
项目
帮助
1 QT += core gui opengl
2
3 greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
4
5 CONFIG += console qt c++11
6
7 DEFINES += QT_DEPRECATED_WARNINGS
8
9 INCLUDEPATH += "D:\OpenGL\glew-2.1.0\include"
10 LIBS += \
11     Glu32.lib \
12     OpenGL32.lib
13
14 SOURCES += \
15     main.cpp \
16     myglwidget.cpp
17
```

fig 12: 配置 GL/glew.h



```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS D:\Qt\Proj\CGTemplate> qmake -tp vc
PS D:\Qt\Proj\CGTemplate>
```

fig 13: 再次键入命令：qmake -tpvc 生成 VS 工程文件



在 VS 中打开并运行，得到 Demo 的输出结果。

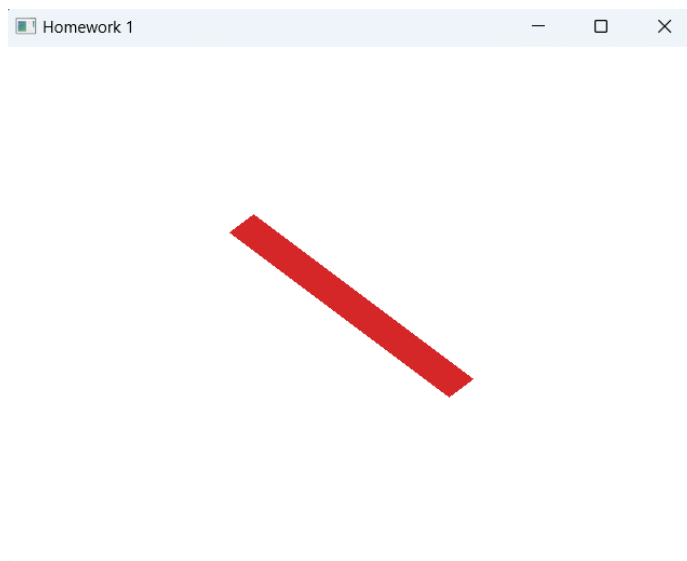


fig 14: Demo 输出结果

至此，环境配置的工作全部完成。

## 2 二维平面绘图

### 2.1 结果展示

绘制平面姓名首字母。

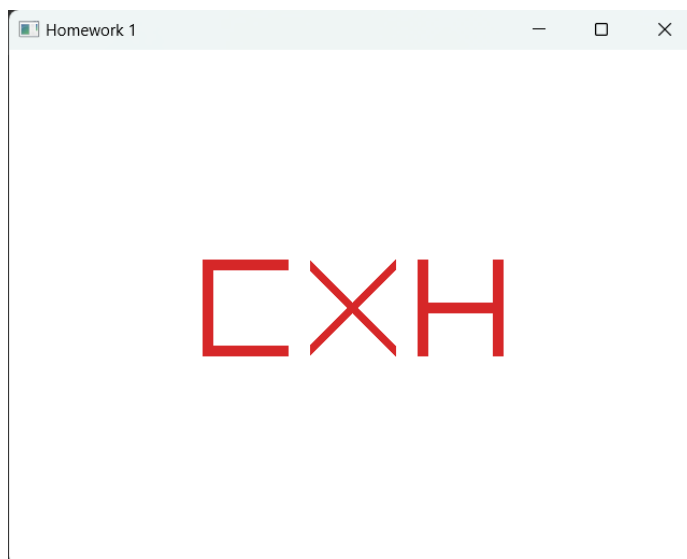


fig 15: 输出结果

## 2.2 实验讨论

### 2.2.1 绘制开销

使用基本图元进行姓名首字母绘制的图例如下：

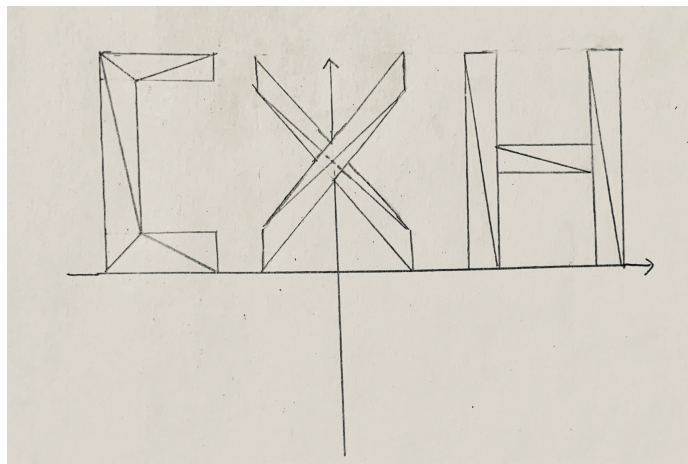


fig 16: 绘制图例

可以推出，使用 `GL_TRIANGLES` 函数构造基本图元需要调用 48 次 `glVertex` 函数，使用 `GL_TRIANGLE_STRIP` 函数构造基本图元需要调用 28 次 `glVertex` 函数，使用 `GL_QUAD_STRIP` 函数构造基本图元也需要调用 28 次

glVertex 函数。

因此，GL\_TRIANGLES 函数构造基本图元的开销最大。

### 2.2.2 视角比较

以下过程均使用  $(0, 1, 0)$  作为视角的上方向 (up)。

**Orthogonal 投影方式产生的图像：**

(1) 从  $(0, 0, d)$  看向原点  $(0, 0, 0)$ ；



fig 17: 从  $(0, 0, 1)$  看向原点  $(0, 0, 0)$

(2) 从  $(0, 0.5*d, d)$  看向原点  $(0, 0, 0)$ ：

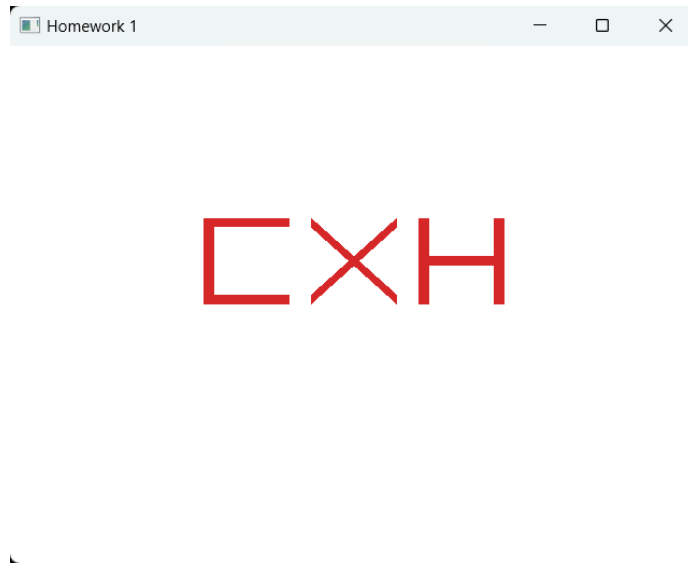


fig 18: 从  $(0,1,2)$  看向原点  $(0, 0, 0)$

可见，Orthogonal 投影方式产生的图像，视角由  $(0,0,d)$  变换到  $(0,0.5*d,d)$  后图形变得更为扁平。

**Perspective 投影方式产生的图像：**

- (1) 从  $(0,0,d)$  看向原点  $(0, 0, 0)$ ;

fig 19: 从  $(0,0,200)$  看向原点  $(0, 0, 0)$ 

(2) 从  $(0,0.5*d,d)$  看向原点  $(0, 0, 0)$ :

fig 20: 从  $(0,100,200)$  看向原点  $(0, 0, 0)$ 

可见，Perspective 投影方式产生的图像，视角由  $(0,0,d)$  变换到  $(0,0.5*d,d)$  后可以明显感觉到透视感。

### 3 三维空间绘图

#### 3.1 结果展示

绘制三维空间姓名首字母。

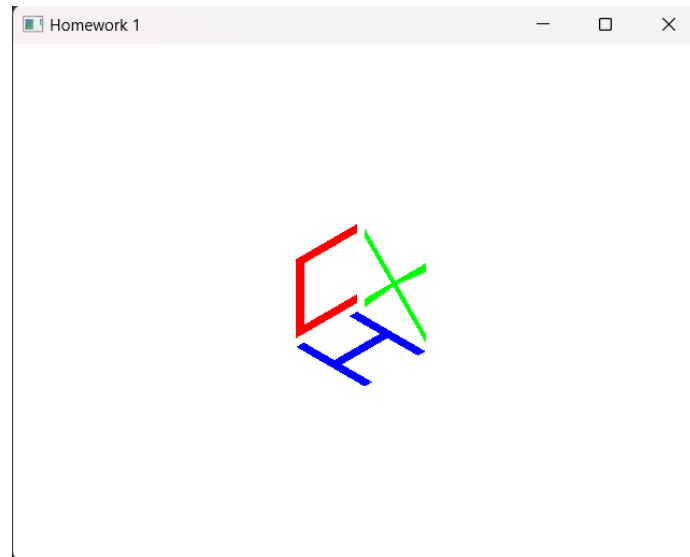


fig 21: 输出结果

#### 3.2 实验讨论

绕  $z$  轴旋转，不同参数的旋转效果：

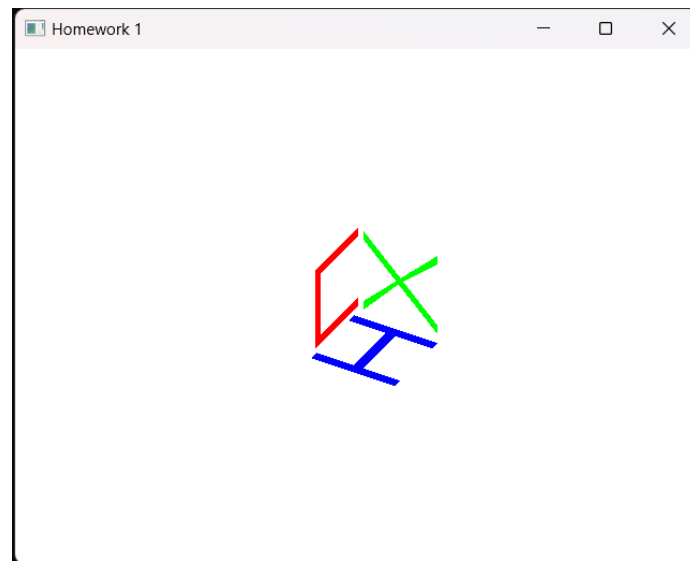


fig 22: 旋转 15 度

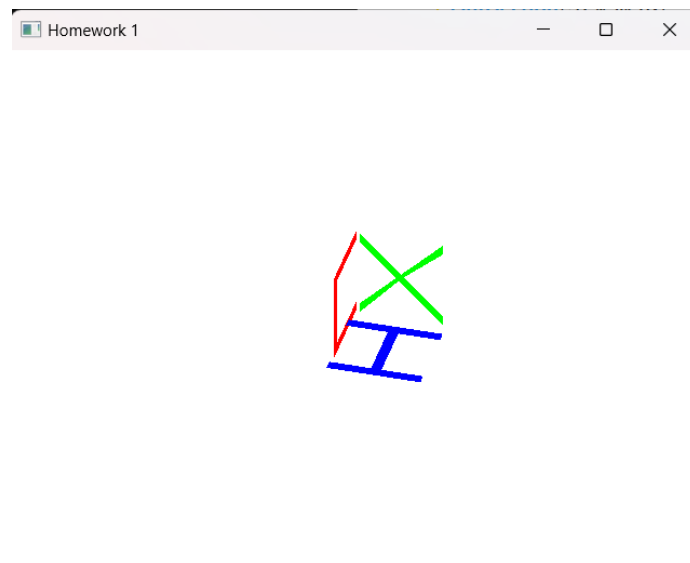


fig 23: 旋转 30 度

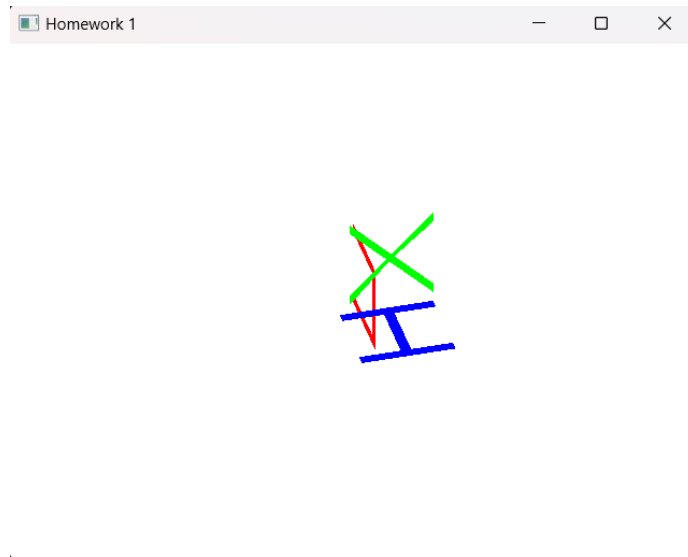


fig 24: 旋转 60 度

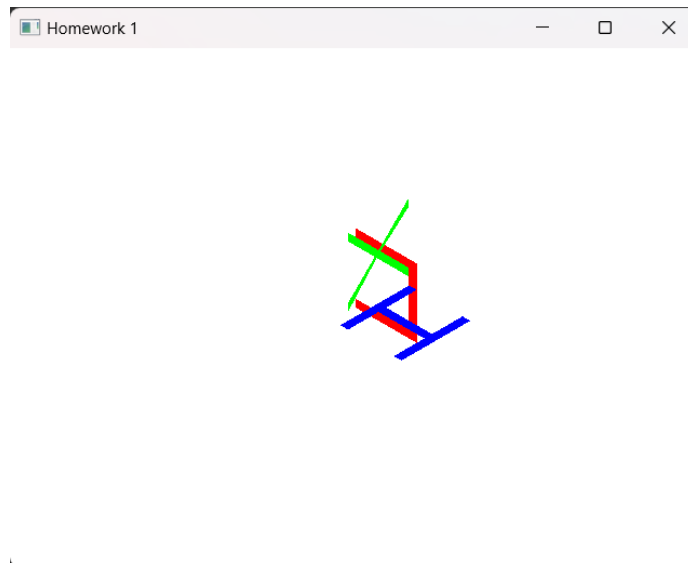


fig 25: 旋转 90 度



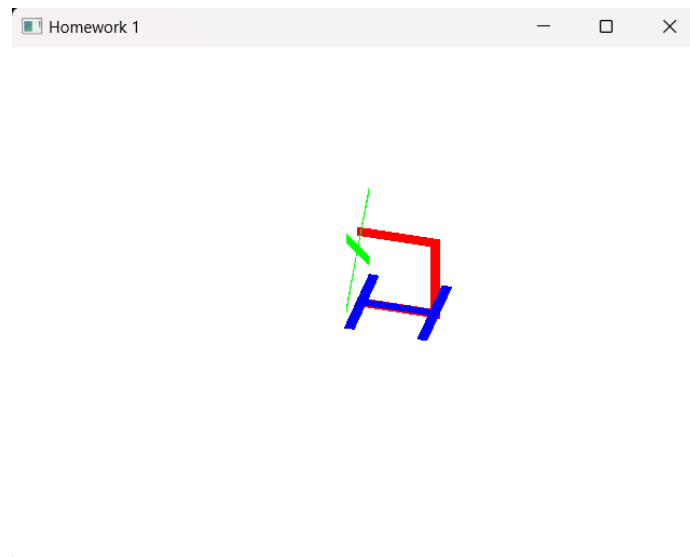


fig 26: 旋转 120 度

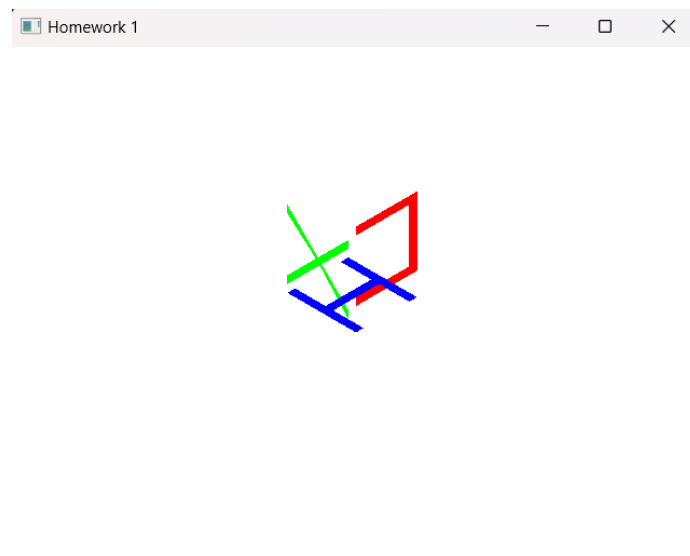


fig 27: 旋转 180 度