

浙江大学

面向对象程序设计

大 程 序 报 告



姓名：_____ 学号：_____ 电话：_____

指导老师：李际军

2021~2022 秋冬学期 2021 年 12 月 31 日

报告撰写注意事项

- 1) 图文并茂。文字通顺，语言流畅，无错别字。
- 2) 书写格式规范，排版良好，内容完整。
- 3) 存在拼凑、剽窃等现象一律认定为抄袭；0 分
- 4) 蓝色文字为说明，在最后提交的终稿版本，请删除这些文字。

目 录

1	大程序简介	4
1.1	选题背景及意义	4
1.2	目标要求	4
1.3	术语说明	4
2	功能需求分析.....	4
3	程序开发设计.....	5
3.1	总体架构和类体系设计	5
3.2	类体系设计	5
3.3	主要类内部函数功能设计描述	5
3.4	源代码文件组织设计	8
4	部署运行和使用说明	10
4.1	编译安装	10
4.2	运行测试	10
4.3	使用操作	11
5	总结和参考文献资料	12
5.1	编码规范	12
5.2	收获感言	12
5.3	参考文献	12

MeshWare 大程序设计

1 大程序简介

本次大程序设计为三维模型处理软件 MeshWare。

1.1 选题背景及意义

在计算机图形学中，三维网格处理是一个重要的部分；借鉴成熟的开源三维图像处理软件，构建一个简化版本的类似软件，有助于我们认知 Qt、OpenGL 等泛用 C++ 库，并深化我们对类体系的认识。

1.2 目标要求

1. 完成常见三维模型的读取和显示
2. 完成放大、缩小、旋转等基本图形功能
3. 完成基本常用的菜单功能

1.3 术语说明

无。

2 功能需求分析

2.1 基本程序功能

1. 基本文件操作
2. 基本操作的快捷键及图标工具栏
3. 帮助

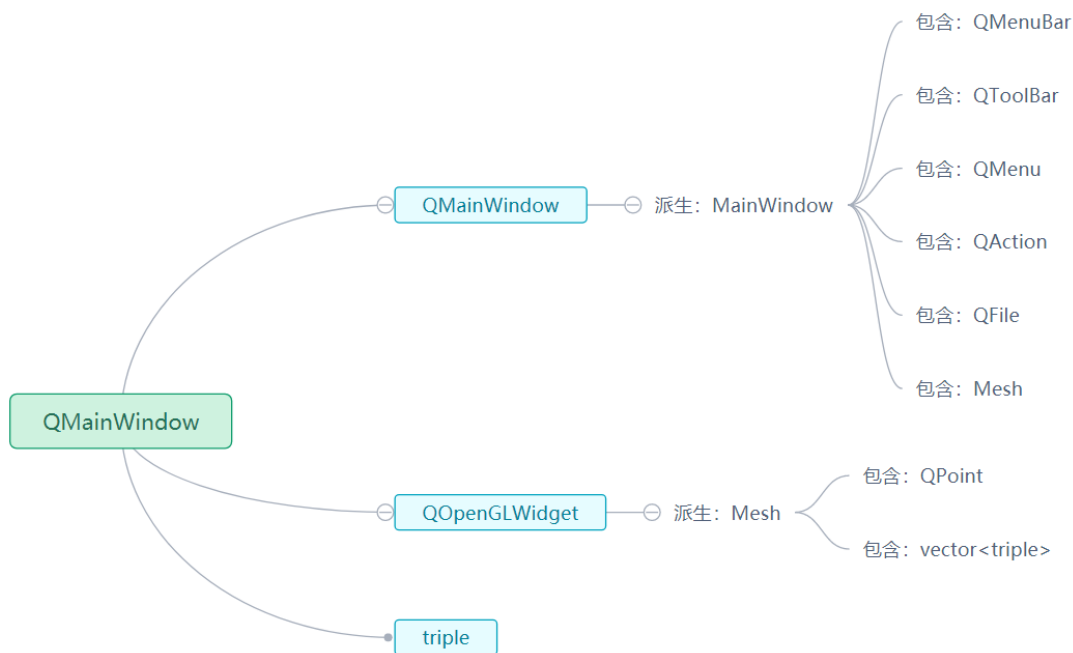
2.2 图像处理功能

1. 放大、缩小
2. 旋转

3 程序开发设计

3.1 总体架构和类体系设计

通过 `main()` 调用 `MainWindow`, 通过 `MainWindow` 调用 `Mesh`。类体系如下:



3.2 功能算法类体系设计

按照需要定义了三个类, 作为泛型对象, 在其中采用 `vector` 等 STL 单元存储数据。

3.3 主要类内部函数功能设计描述

1. MainWindow 类

该类由 `QMainWindow` 派生。

i) FileLoadManagement()

原型: `bool FileLoadManagement();`

功能: 在 Load 操作被 trigger 的情况下, 完成 .ply 文件读取和显示

参数: 无

返回值：返回操作成功与否

局部变量：

QString path: 存储打开的文件路径

String str: path 的 String 形式, 通过 substr() 方法获得文件扩展名进行判断

算法: 通过 getOpenFileName() 获得文件路径; 为空则返回; 不为空则检测扩展名; 不为 .ply 则输出信息; 否则打开该文件, 并调用 DisplayMesh()

ii) HelpHelpManagement()

原型: bool HelpHelpManagement();

功能: 在 Help 操作被 trigger 的情况下, 给出帮助提示

参数: 无

返回值: 返回操作成功与否

局部变量: 无

算法: 调用 SendMsg()

iii) 构造函数

原型: MainWindow(QWidget *parent = Q_NULLPTR);

功能: 完成构造

参数: QWidget* parent: 母窗口

返回值: this

局部变量: 无

算法: 调用 ui; 构造选项栏和操作; 构造图像按钮

2. Mesh 类

该类由 QWidget 派生。

i) draw()

原型: void draw();

功能: 读取 vector<triple>, 完成点云画图

参数: 无

返回值: 无

局部变量: 无

算法描述: 按照目前的旋转角度和点向量, 调用 OpenGL 图形库, 画

出点云

ii) mousePressEvent ()

原型: void mousePressEvent (QMouseEvent* event);

功能: 完成鼠标点击的处理

参数: event: 事件信息

返回值: 无

局部变量: 无

算法描述: 直接记录目前鼠标点击事件发生的坐标

iii) mouseMoveEvent ()

原型: void mouseMoveEvent (QMouseEvent* event);

功能: 完成鼠标移动的处理

参数: event: 事件信息

返回值: 无

局部变量: 无

算法描述: 在同时有鼠标按下时, 直接记录目前鼠标移动事件发生的坐标, 对坐标系旋转角度进行操作, 并重新画图

iv) wheelEvent ()

原型: void wheelEvent (QMouseEvent* event);

功能: 完成鼠标滚轮事件的处理

参数: event: 事件信息

返回值: 无

局部变量: 无

算法描述: 按照鼠标滚轮的偏移量增减放大倍数 zoom (当 zoom 不为正数则置为 1), 并重新画图

v) initializeGL ()

原型: void initializeGL ();

功能: 初始化画图窗口

参数: 无

返回值: 无

局部变量: 无

算法描述：置背景色为黑色

vi) `resizeGL()`

原型： `void resizeGL(int width, int height);`

功能：初始化画图窗口大小

参数： `width`、`height`：窗口大小参数

返回值：无

局部变量：无

算法描述：修正窗口大小并创建透视模型

vii) `paintGL()`

原型： `void paintGL();`

功能：画出点云图

参数：无

返回值：无

局部变量：无

算法描述：清除缓冲区，调用 `draw()`

viii) 构造函数

原型： `Mesh(QFile* PLY, QWidget* parent = 0);`

功能：初始化旋转角度、放大倍数等画图参数，并将文件数据读入到类中存储

参数： `PLY`：读取的文件指针；`parent`：母窗口

返回值： `this`

局部变量：

`QString le`： `PLY` 的逐行数据

`double x, y, z`：从文件读出的点坐标

算法描述：逐行读取 `PLY` 文件，读到 `end_header` 后开始读入 `x, y, z`，构造出 `triple` 并放入 `vector` 存储；将旋转角度和放大倍数初始化

3.4 源代码文件组织设计

<文件目录结构>

1) 文件函数结构

3. `main.cpp`

这里是 `main` 函数。声明一个 `QApplication` 类对象；声明一个 `MainWindow` 类对象，并进行 `show()` 操作来生成主窗口。

4. `MainWindow.h`

`MainWindow.cpp`

`MainWindowRuntime.h`

`MainWindowRuntime.cpp`

在 `.h` 文件中放置了 `MainWindow` 类的类定义和相关组件和函数声明。

具体实现在两个 `.cpp` 文件中。其中 `MainWindow.cpp` 进行主窗口初始化，`MainWindowRuntime.cpp` 执行主窗口运行时函数调用。

5. `Debug.h`

`Debug.cpp`

原意是用来作为调试时的运行库，后来因为工作计划仓促，实际上变成了生成 `QMessageBox` 来给出弹出窗口的运行库。

6. `triple.h`

这里放置了一个 `triple` 类的声明，用来表示一个点的 `x, y, z` 坐标，以方便 `std::vector` 存储。

7. `Widget.h`

`Widget.cpp`

这里是 `Mesh` 类的声明及相关函数的定义及实现。它完成读取 `.ply` 文件并显示的过程及完成旋转等图形基本操作。

2) 多文件构成机制

头文件使用预编译命令 `#pragma once` 来防止重复包含头文件导致声明冲突。因此没有使用 `#define #ifndef #endif` 方式进行头文件保护。

每一个 `.h` 文件都有一个对应 `.cpp` 文件进行定义，使用的全局函数较少，无全局变量。

一般，`.cpp` 文件包含 `.h` 文件。`.h` 文件中有相互包含。

在 `main.cpp` 中通过预编译命令

```
#pragma comment(lib, "opengl32.lib")
```

```
#pragma comment(lib, "glu32.lib")
```

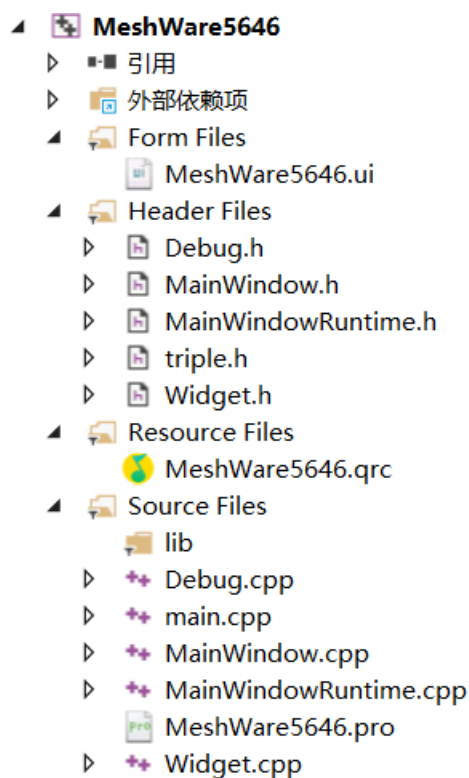
```
#pragma comment(lib, "glut32.lib")
```

进行库声明。

4 部署运行和使用说明

4.1 编译安装

环境使用 VS2017, Qt5.12+OpenGL。项目需要的文件为：



除此之外还需要 texture 文件夹放在源码同一目录下进行编译。生成 exe 即可；预编译好的可执行文件可在 MeshWare5646/Release 文件夹中找到。

4.2 运行测试

1. 文件读取功能测试

在文件读取过程中，容易出现读取文件不正确的情况，主要是以下两种：在读取文件时退出；读取不正确的文件。这两种情况都会导致 QFile 打开不正确，从而使程序崩溃退出。

经过修改，通过特判空字符串、将 QString 转换成 std::string 并查最后的四个字符（文件扩展名）就行判断避免了此类问题。

2. 画图功能测试

一开始，我的写法是在 paintGL 函数内进行文件读取和画图。在测试中发现，由于 update 会调用 paintGL 函数，在后续画图时就会导致失效的文件指针，从

而程序崩溃退出。之后我在构造函数内将文件信息导入 **Mesh** 类，避免了此类问题。

4.3 使用操作

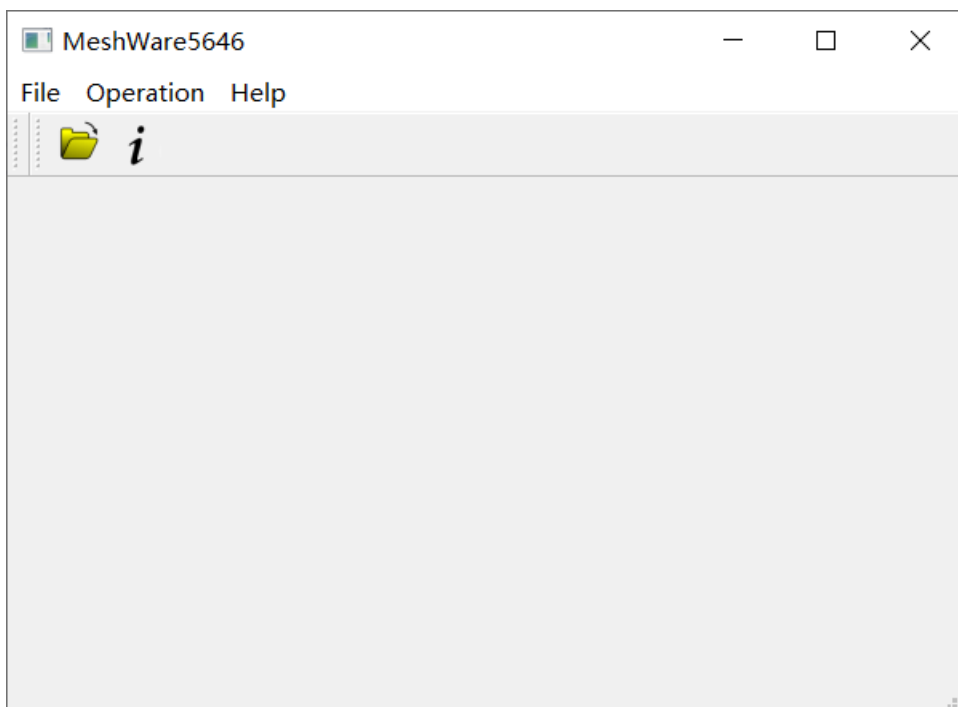
通过 **File/Load (Ctrl+I)** 打开文件对话框，进行读取文件操作。注意：目前文件读取只支持 **.ply** 文件，且正常读取需要为 **ascii** 编码。

通过 **Help/Help (Ctrl+H)** 查看（很摆烂的）帮助。也可以通过两个图标按钮进行上述操作。

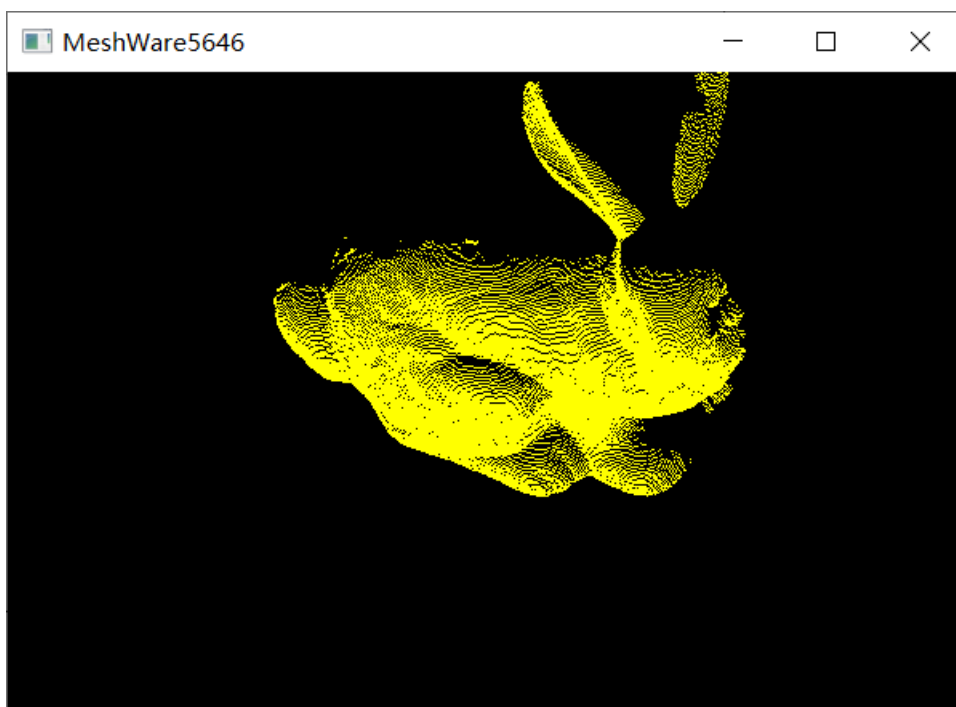
在查看图形文件时：鼠标滚轮可控制放大/缩小，按住鼠标左键/右键进行旋转。

在目录下，我提供了一些可用的 **example**。

使用前需要将 **Qt** 的相关 **dll** 文件添加到环境变量中。



主窗口



三维点云示例

5 总结和参考文献资料

5.1 编码规范

编码语言使用 GNU C++11。编码规范按照 VS 自动格式已经过校正，严格遵守课程要求的编码规范。

5.2 收获感言

本次大作业让我明白了在较短时间内从头开始研究一个大库并进行程序设计是相当难的。我在参考 Qt 文档等调试了数天后才画出了第一个三维图形，之后的时间也都用来研究实现基本图形功能，也就没有完全实现好不同文件格式的读取。

总之，我的大框架经历了多次调整，导致编程思维较混乱，最后实现的功能也较少，以后在进行大型编程时应该尽可能确定一个较好的框架。

5.3 参考文献

Qt 框架：

1. [Qt5.9 中工具栏类 QToolBar 的用法_净无邪博客-CSDN 博客](#)
2. [Qt 创建的窗口一闪而过 \(打开立刻关闭\)的一种解决方案_清楚](#)

[xc 的博客-CSDN 博客](#)

3. [使用 QFile 进行文件操作_未来无限的博客-CSDN 博客](#)
4. [Qt 学习之路 10--Qt 标准对话框（消息提示对话框和文件对话框）
一念之间、-CSDN 博客](#)

OpenGL 控件：

1. [QT 关于鼠标滚轮事件_keen-CSDN 博客](#)
2. [Qt 中 OpenGL 的四个重要事件\(initializeGL\(\) resizeGL\(\) paintGL\(\)
pdateGL\(\)\)调用规则_AplusX-CSDN 博客](#)
3. [Qt 环境下利用 OpenGL 显示三维模型_我要上火星（停止更新，
转为私人笔记）-CSDN 博客](#)
4. [OpenGL\(3\)之 Qt 窗口\(QOpenGLWidget\) Joey 博客-CSDN 博客](#)