

浙江大学

面向对象程序设计

大 程 序 报 告



姓名： 叶泽凯 学号： 3200104207 电话： 18658705712

指导老师： 李际军

2021~2022 秋冬学期 2021 年 12 月 21 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

<b>1</b>	<b>大程序简介 .....</b>	<b>4</b>
1.1	选题背景及意义 .....	4
1.2	目标要求 .....	4
1.3	术语说明 .....	4
<b>2</b>	<b>功能需求分析.....</b>	<b>4</b>
<b>3</b>	<b>程序开发设计.....</b>	<b>5</b>
3.1	总体架构和类体系设计 .....	5
3.2	类体系设计 .....	6
3.3	主要类内部函数功能设计描述 .....	7
3.4	源代码文件组织设计 .....	8
<b>4</b>	<b>部署运行和使用说明 .....</b>	<b>11</b>
4.1	编译安装 .....	11
4.2	运行测试 .....	15
4.3	使用操作 .....	16
<b>5</b>	<b>总结和参考文献资料 .....</b>	<b>20</b>
5.1	编码规范 .....	20
5.2	收获感言 .....	20
5.3	参考文献 .....	20

# Mashware4017 大程序设计

## 1 大程序简介

### 1.1 选题背景及意义

MeshLab 是一个开源、可移植和可扩展的三维几何处理系统，主要用于交互处理和结构化编辑三维三角形网格。本次大程序设计旨在实现一个小规模的 MeshLab，以更深入地了解 C++ 的特性，以及对图形学、项目构建等有初步了解。

### 1.2 目标要求

- 1) 提供美观、合理、符合传统习惯的界面
- 2) 包含文件、编辑、显示设置、帮助等基本菜单体系
- 3) 支持对网格的加密和稀疏等高级操作

### 1.3 术语说明

术语	说明
OpenGL	用于渲染 2D、3D 矢量图形的跨语言、跨平台的应用程序编程接口 (API)
Qt	一个用于创建图形化用户接口的窗口工具链
Visual Studio 2019	C++ 编程 IDE，项目构建平台
VCGLib	Visualization and Computer Graphics Library，可视化与计算机图形学库，用于三角网络和四面体网络的控制、处理和 OpenGL 的显示
VBO	顶点缓存对象，将绘制图形的信息投喂给 VBO，再由 VBO 传递给显卡
VAO	顶点数组对象，保存对 VBO 内信息利用方式，并通知显卡

## 2 功能需求分析

①主要功能列举如下表

功能	类
文件操作窗口	QMainWindow
模型读取	vcg::tri::io::Importer
模型保存	vcg::tri::io::Exporter
模型移动、翻转、大小改变	矩阵实现

模型稀疏	vcg::tri::Clustering
模型加密	vcg::tri::Refine
模型平滑	vcg::tri::Smooth
帮助窗口	QMainWindow
状态栏	QMainWindow

②需要的主要库列举如下

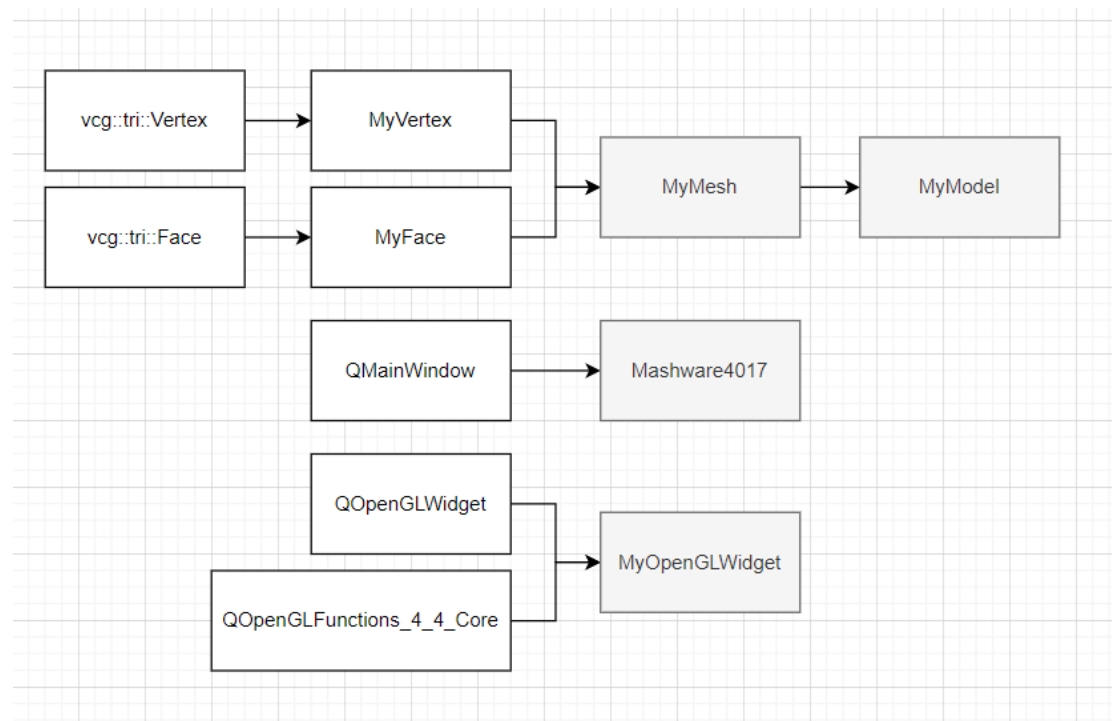
类	备注
vcglib	网格操作数学库
QMainWindow	UI 设计与响应功能库
QOpenGLFunctions_4_4_Core	Qt 的 OpenGL 库

③性能要求

- 1) 图像操作不卡顿
- 2) 能打开大小合适的 3D 图像文件

## 3 程序开发设计

### 3.1 总体架构和类体系设计



总体架构类图如上所示，所有的类包含均为公有继承，深色部分为我自己实现的部分

## 3.2 功能算法类体系设计

*焦点关注功能的类体系设计，尤其是自己简化的思路*

### 1) 总体

本项目的功能类体系，为 mymodel 类和 myvcg 类

主要功能参考了网上的资料以及 vcg 自带的 app 示例。

其中光滑 (smooth)，稀疏 (cluster)，加密 (refine) 的主要代码也是参考了

①trimesh\_smooth ②trimesh\_clustering ③trimesh\_refine 三个样例文件

### 2) mymodel 类

mymodel 主要实现对模型的抽象与视角的切换

其中视角的切换未应用摄像机概念，直接通过矩阵的乘法，对点的坐标进行改变  
矩阵的定义如下，分别对应着旋转和大小的改变

<pre>float RotateX[9] = {     1.0f, +0.0f, 0.0f,     0.0f, +cos1, sin1,     0.0f, -sin1, cos1 };  float RotateY[9] = {     cos1, 0.0f, -sin1,     0.0f, 1.0f, +0.0f,     sin1, 0.0f, +cos1 };  float RotateZ[9] = {     +cos1, sin1, 0.0f,     -sin1, cos1, 0.0f,     +0.0f, 0.0f, 1.0f };  float RotateX_[9] = {     1.0f, +0.0f, +0.0f,     0.0f, cos1, -sin1,     0.0f, +sin1, cos1 };</pre>	<pre>float RotateY_[9] = {     cos1, 0.0f, +sin1,     +0.0f, 1.0f, +0.0f,     -sin1, 0.0f, cos1 };  float RotateZ_[9] = {     cos1, -sin1, 0.0f,     +sin1, cos1, 0.0f,     +0.0f, +0.0f, 1.0f };  float Bigger[9] = {     1.25f, 0.0f, 0.0f,     0.0f, 1.25f, 0.0f,     0.0f, 0.0f, 1.25f };  float Smaller[9] = {     0.8f, 0.0f, 0.0f,     0.0f, 0.8f, 0.0f,     0.0f, 0.0f, 0.8f };</pre>
---	---

### 3) myvcg 类

对 vcg 的自带类进行继承，继承关系如下

<pre>struct MyUsedTypes : public vcg::UsedTypes &lt;vcg::Use&lt;MyVertex&gt;::AsVertexType, vcg::Use&lt;MyFace&gt;::AsFaceType&gt; {};  class MyVertex : public vcg::Vertex&lt; MyUsedTypes, vcg::vertex::VFAAdj, vcg::vertex::Coord3f, vcg::vertex::TexCoord2f, vcg::vertex::Normal3f, vcg::vertex::Color4b, vcg::vertex::BitFlags &gt; {};  class MyFace : public vcg::Face&lt; MyUsedTypes, vcg::face::InfoOcf, vcg::face::Normal3f, vcg::face::FFAdjOcf, vcg::face::VFAAdj, vcg::face::VertexRef, vcg::face::BitFlags &gt; {};  class MyMesh : public vcg::tri::TriMesh&lt;std::vector&lt;MyVertex&gt;, vcg::face::vector_ocf&lt;MyFace&gt; &gt;</pre>
--

### 4) 动作实现

每次动作被捕捉后，传入 mymodel，mymodel 做出相应反应，若涉及网格的变换

抑或读取/保存，进一步调用 myvcg 类，否则将最新计算出的点对投喂给 VBO，调用 update()函数，进行更新。

### 3.3 主要类内部函数功能设计描述

函数原型：Update()

功能描述：更新 model 的绘制坐标

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：遍历点集(x,y,z)，与大小矩阵和旋转矩阵依次相乘后，再加上位移向量，得到呈现在画板上的矩阵

函数原型：Draw()

功能描述：绘制 model

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：先清空画布，载入 VAO，设置绘制模式为线模式/填充模式，再进行绘制

函数原型：Extract(bool hav)

功能描述：提取 model 点集

参数描述：hav，是否是第一次调用

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：如果已经被调用过了，清理原来 vertex 和 vertexw 的内存。遍历图像包含的面，将面的三个角的(x,y,z)坐标依次加入 vertex

函数原型：ViewFill()

功能描述：切换为填充模式

参数描述：无

返回值描述：无

重要局部变量定义：无

重要局部变量用途描述：无

函数算法描述：将 mode 切换为 ModeFill 模式，然后进行更新

函数原型：ViewLine ()

功能描述：切换为线模式

参数描述：无

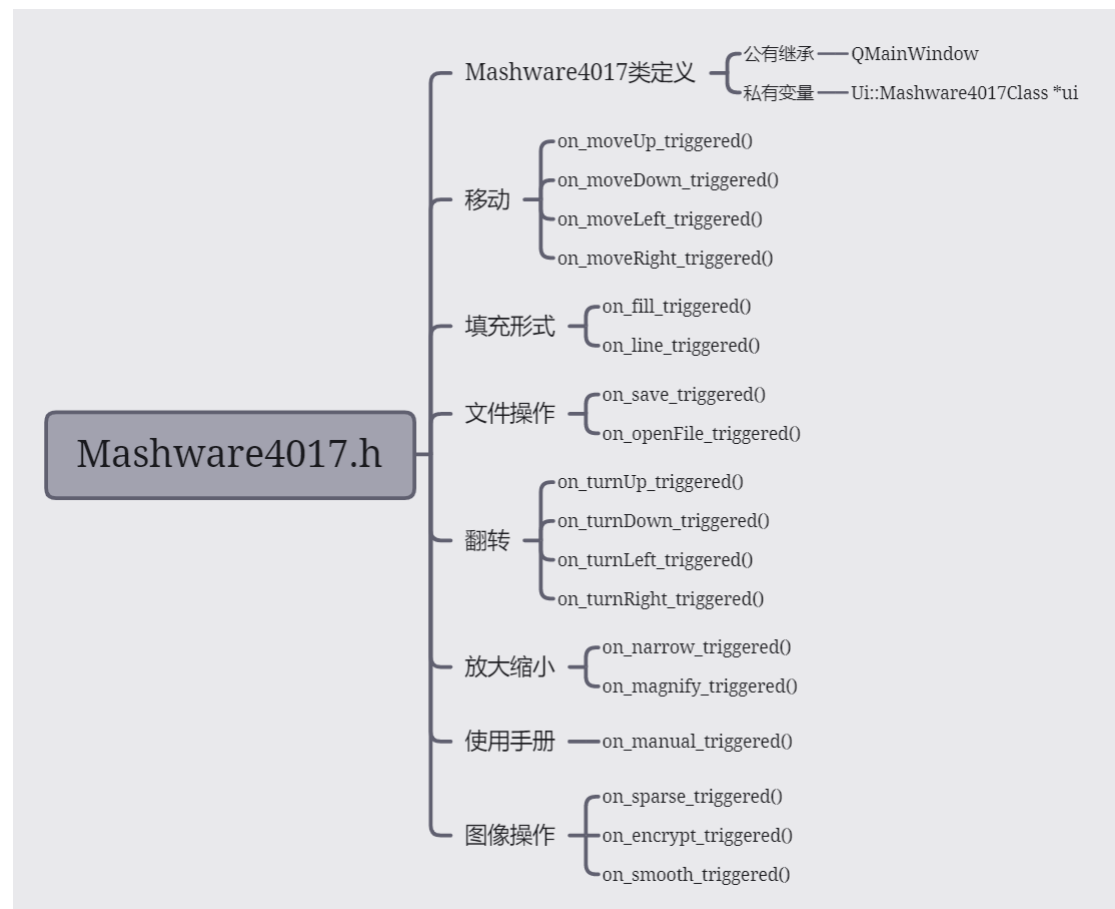
返回值描述：无  
 重要局部变量定义：无  
 重要局部变量用途描述：无  
 函数算法描述：将 mode 切换为 ModeLine 模式，然后进行更新

### 3.4 源代码文件组织设计

<文件目录结构>

1) 文件函数结构

①Mashware4017.h

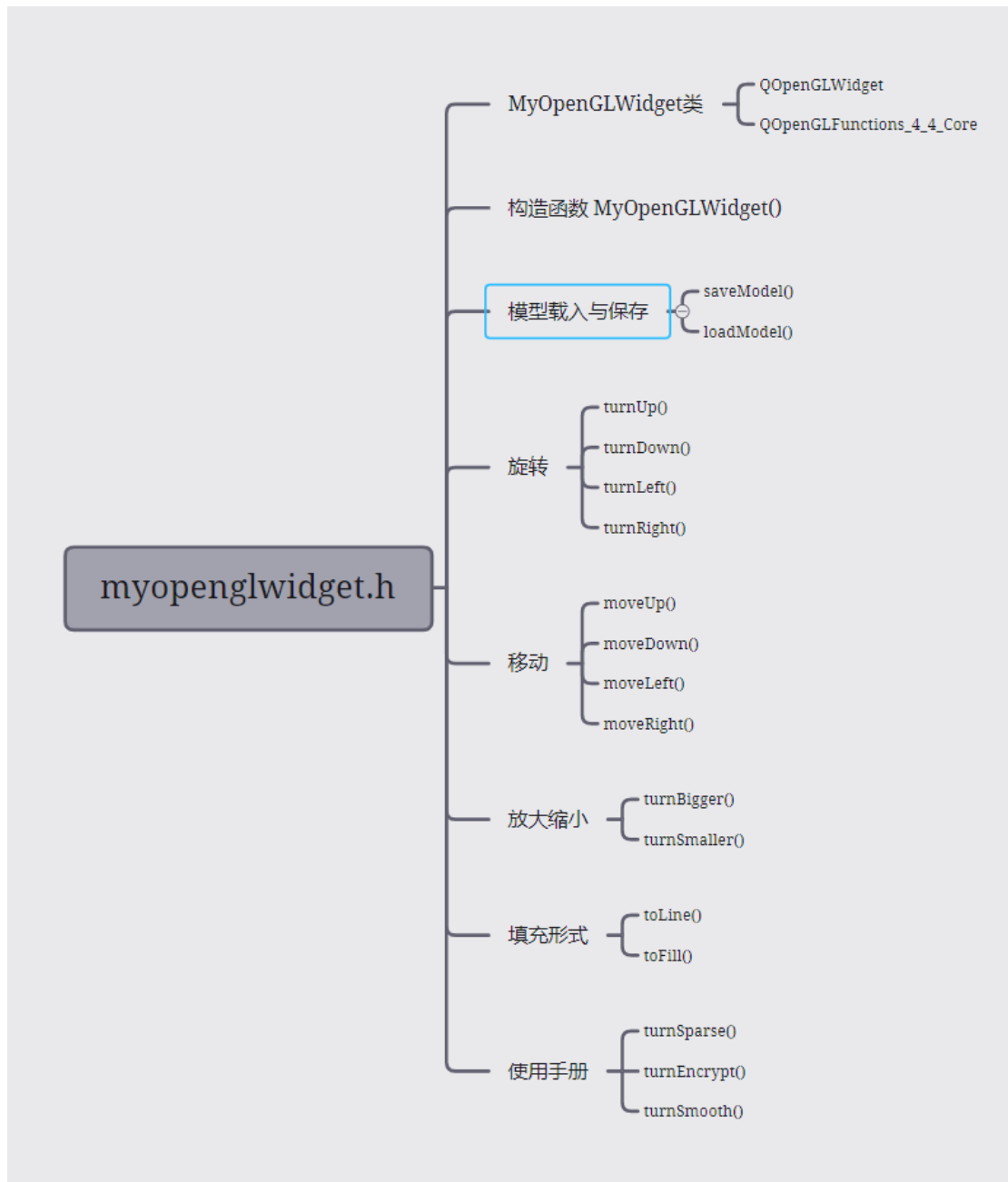


②Mashware4017.cpp

定义了 Mashware4017.h 中声明的所有函数

③myopenglwidget.h

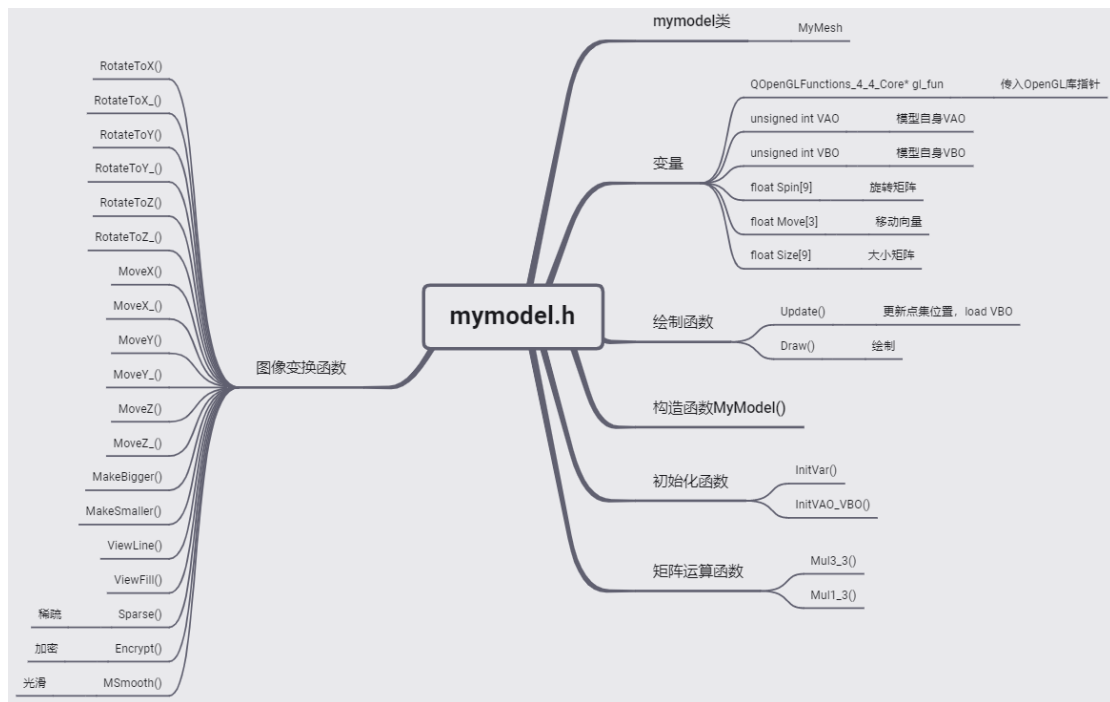




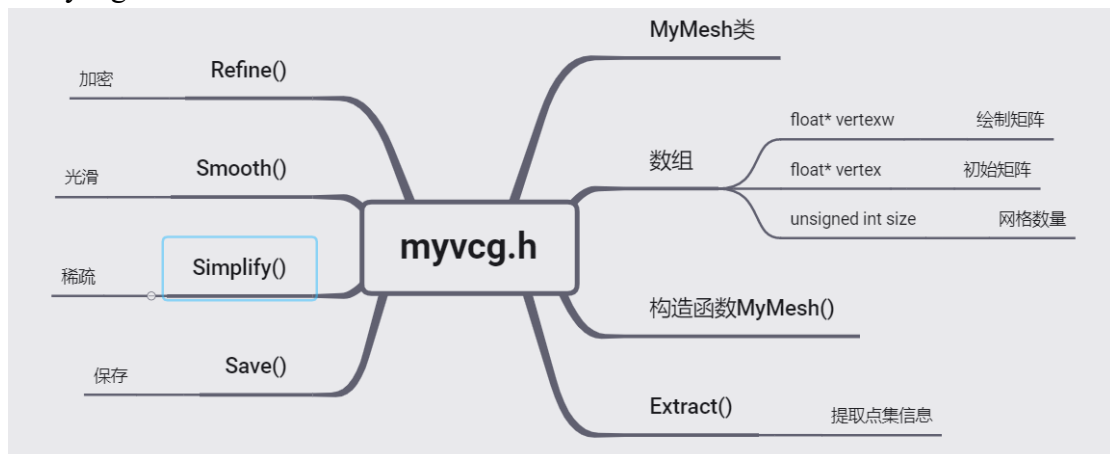
④myopenglwidget.cpp

定义了 myopenglwidget.h 中声明的所有函数

⑤mymodel.h



## ⑥myvcg.h



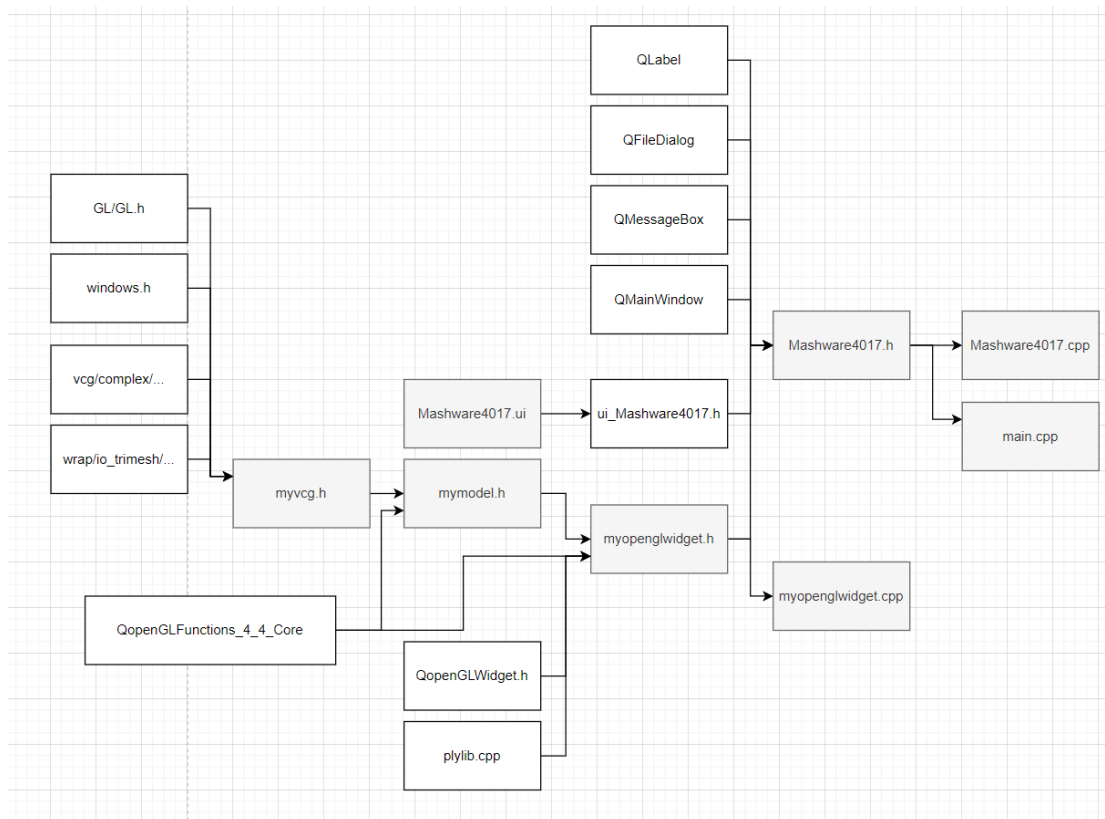
## 2) 多文件构成机制

### ①总的文件架构如下

```

Mashware
├── Mashware4017.ui (ui 设计)
├── Mashware4017.h + Mashware4017.cpp
├── myopenglwidget.h + myopenglwidget.cpp
├── mymodel.h (包含实现)
└── myvcg.h (包含实现)
    
```

### ②文件关系可由下述思维导图展示





备注：有颜色部分为我自己实现的代码；由于篇幅限制，忽略了少部分文件。

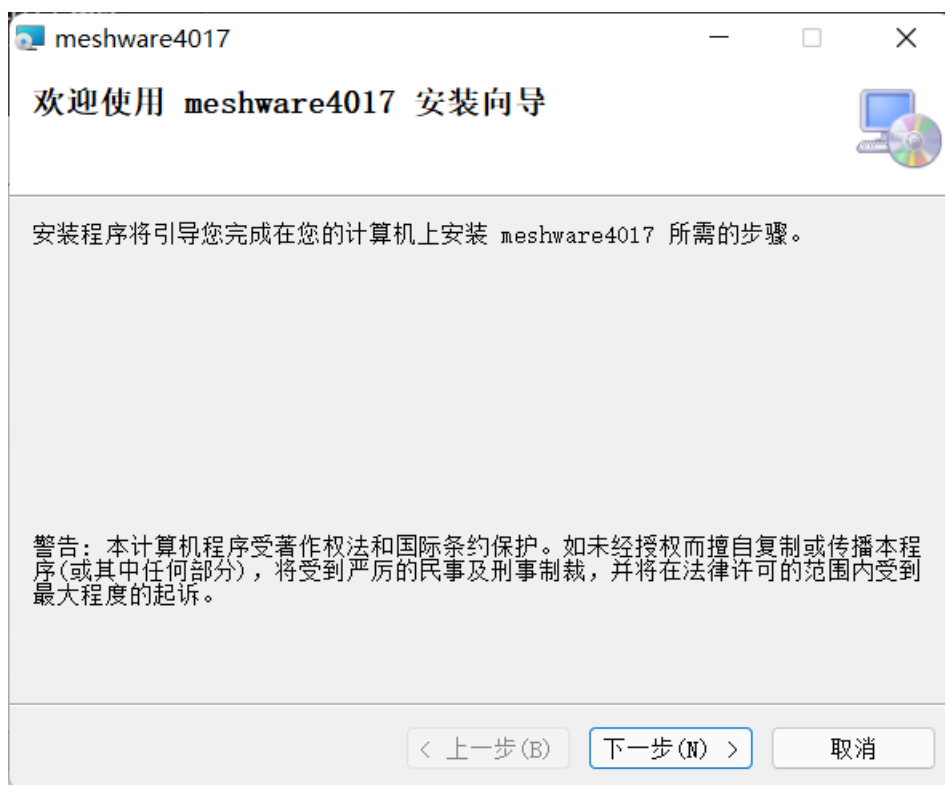
## 4 部署运行和使用说明

### 4.1 编译安装

①运行压缩包内的 meshware4017.ms

 meshware4017.msi	2021/12/21 22:15	Windows Install...	14,263 KB
 setup.exe	2021/12/21 22:15	应用程序	551 KB

② 点 击 下 一 步



③若想要查看源码，则可以将文件夹目录更改为你想保存的目录（若使用默认目录，则在保存文件的时候会出现权限错误）

安装选项，选择只有我或者任何人

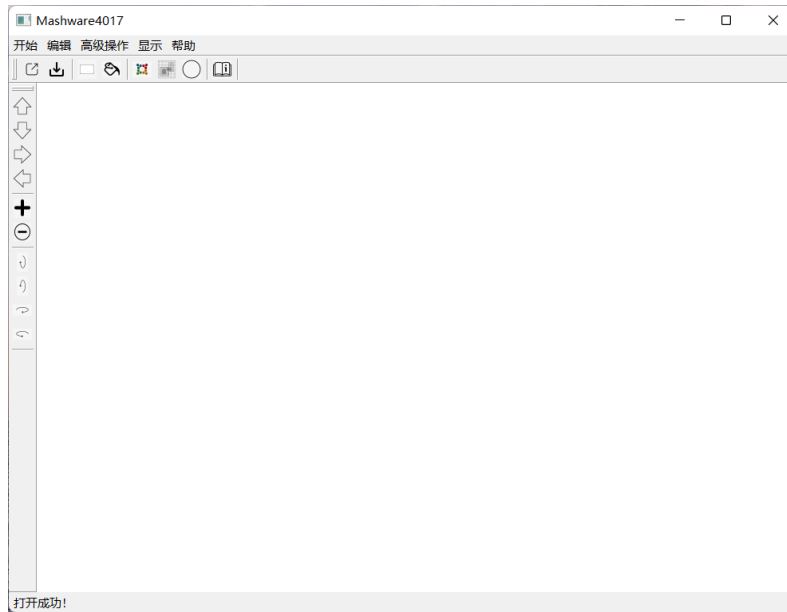


④安装完毕

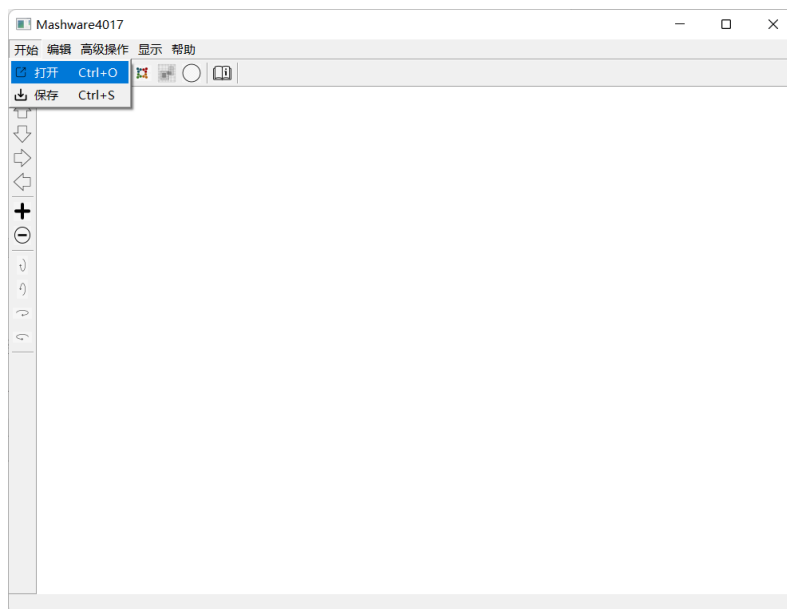
此时在桌面会生成一个可执行文件



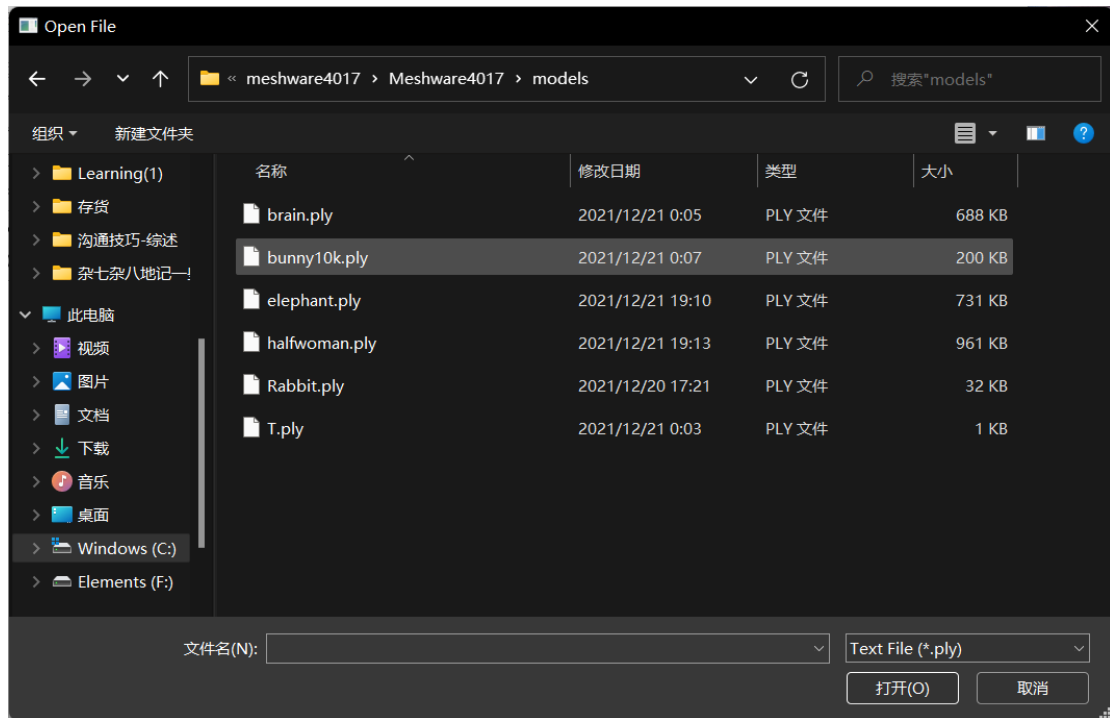
### ⑤双击打开



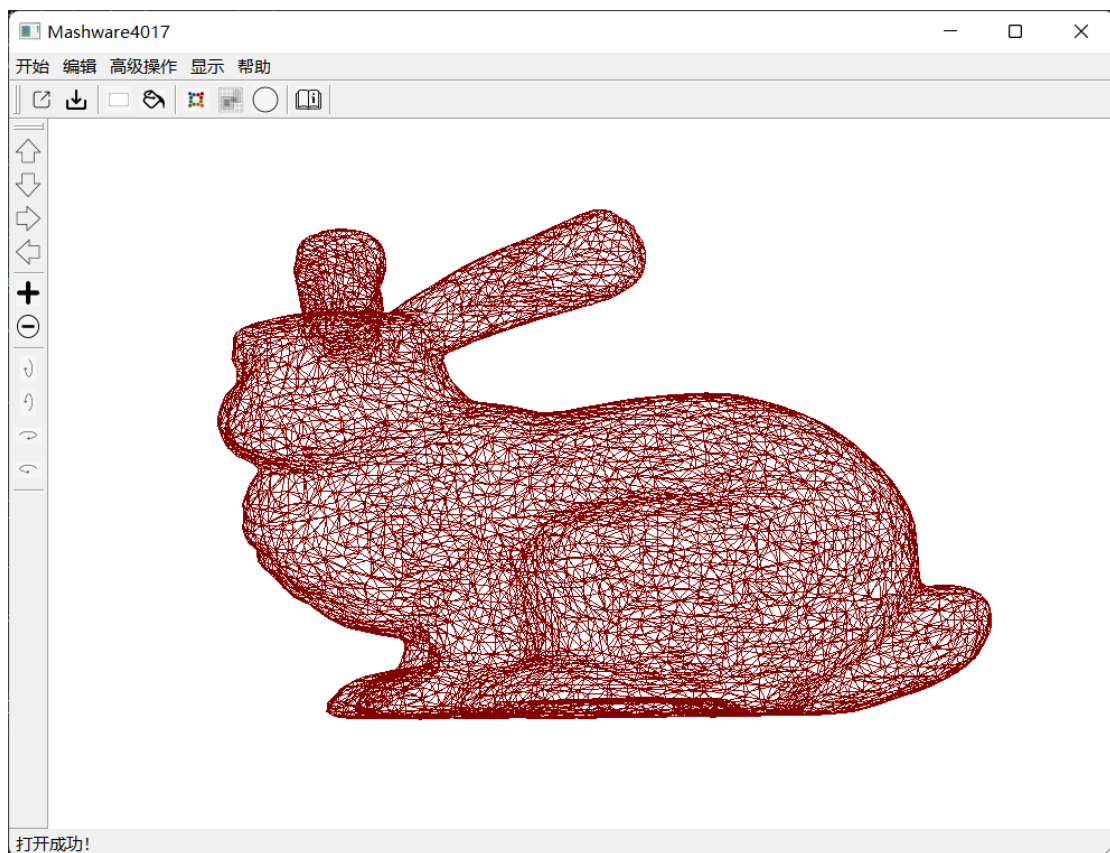
### ⑥打开文件



### ⑦选择任意模型打开

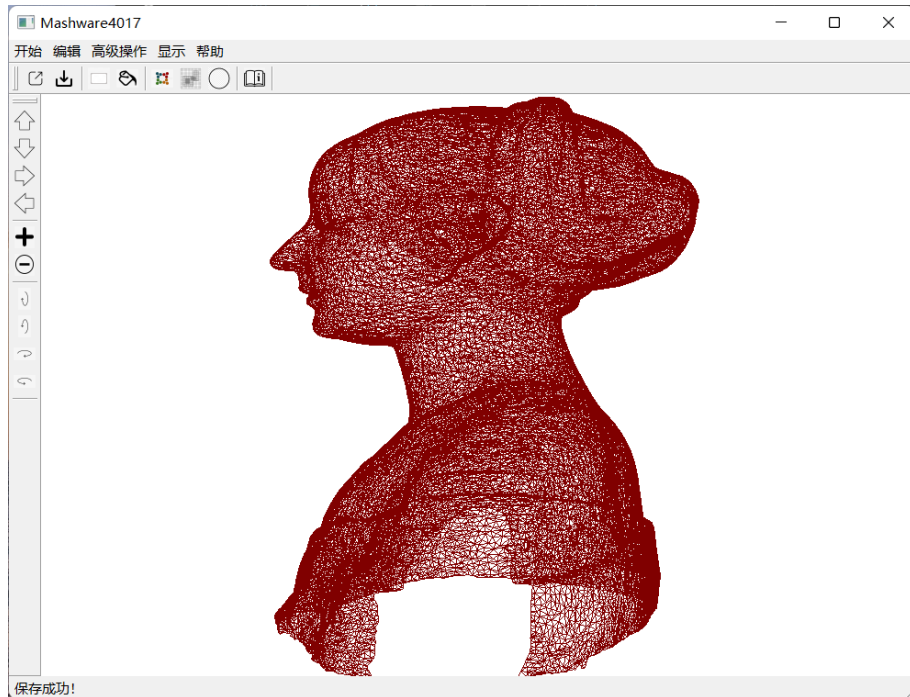


⑧以 bunny10k.ply 为例



进行操作（注意：稀疏操作目前仍存在算法不兼容的情况，可能会有弹窗提示或者会有图像消失的情形）（同时加密操作存在次数限制）

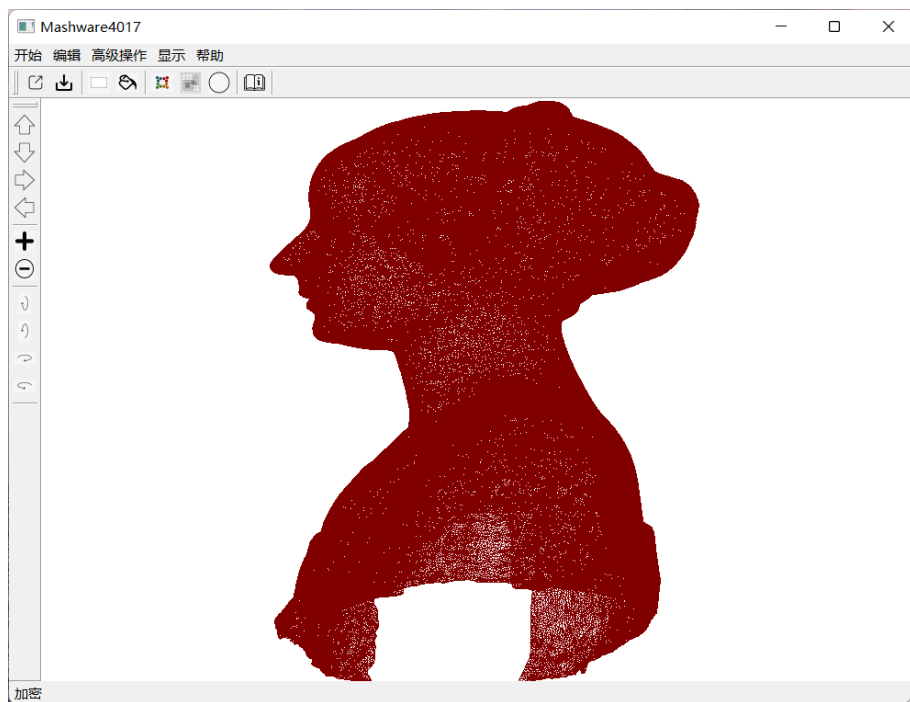
## 4.2 运行测试



打开一个某个文件  
按 e 发现无法加密  
定位至 Refine 函数

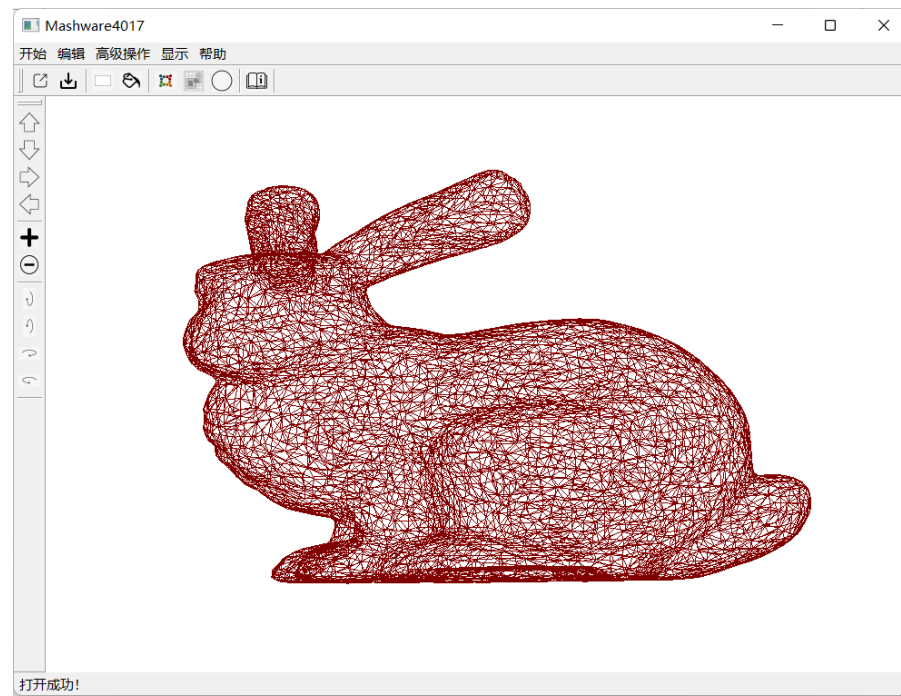
```
164 | vcg::tri::Refine<MyMesh, vcg::tri::MidPoint<MyMesh>>(*this, vcg::tri::MidPoint<MyMesh>(this), 0.1);
```

发现参数设置 0.1 可能过大  
将参数调整到 0.0，可以进行进一步加密

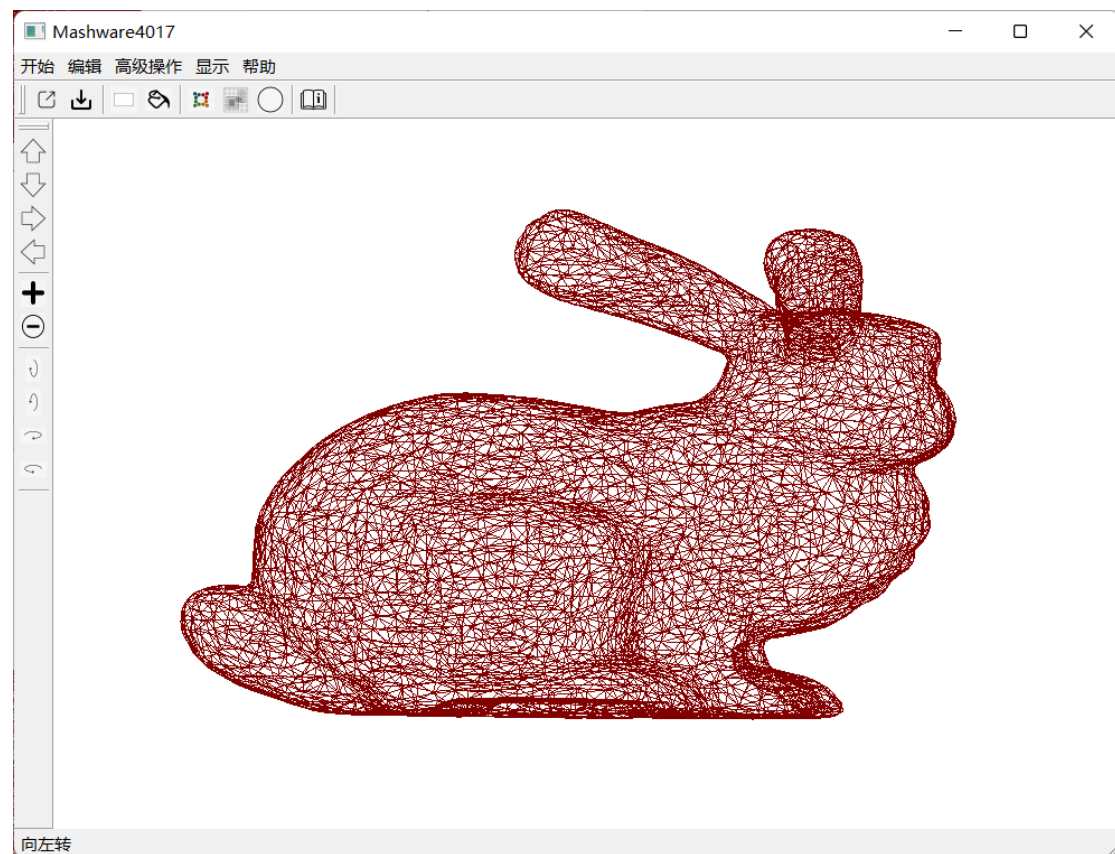


## 4.3 使用操作

打开图像后

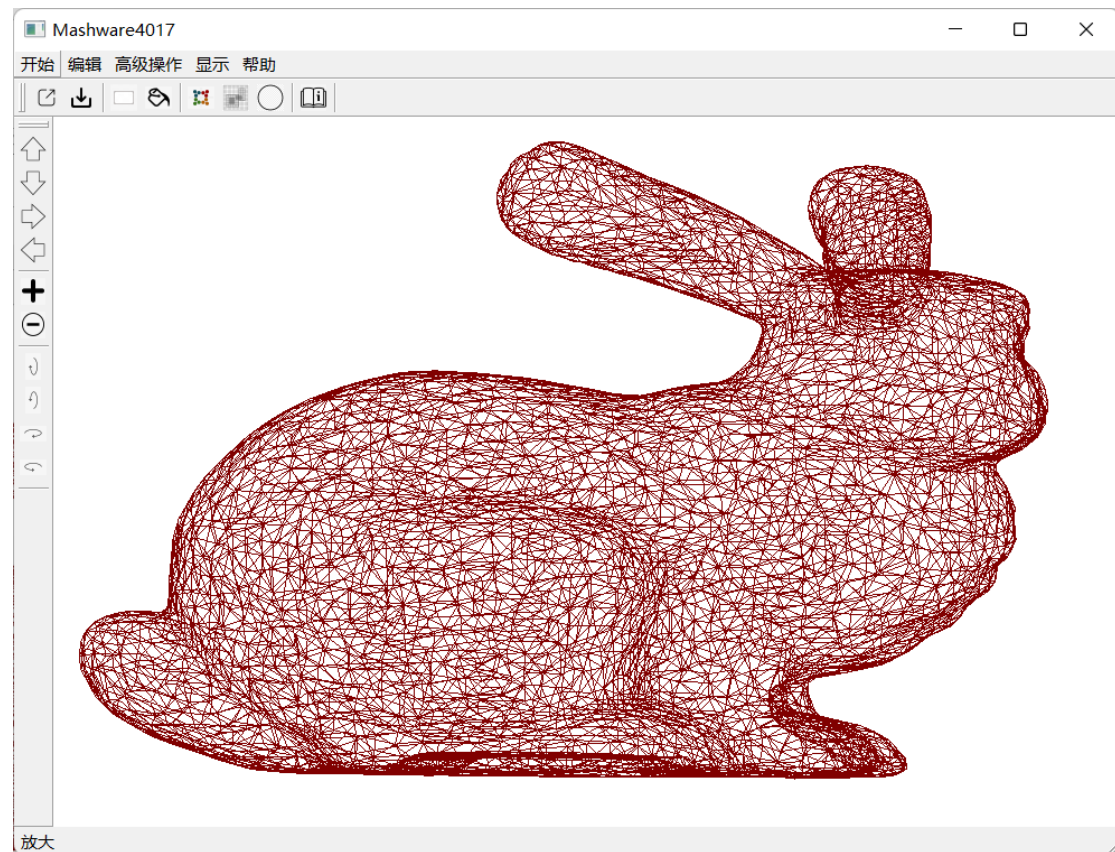


可点击工具栏图标/菜单栏项目/快捷键，对图像进行操作  
如：旋转

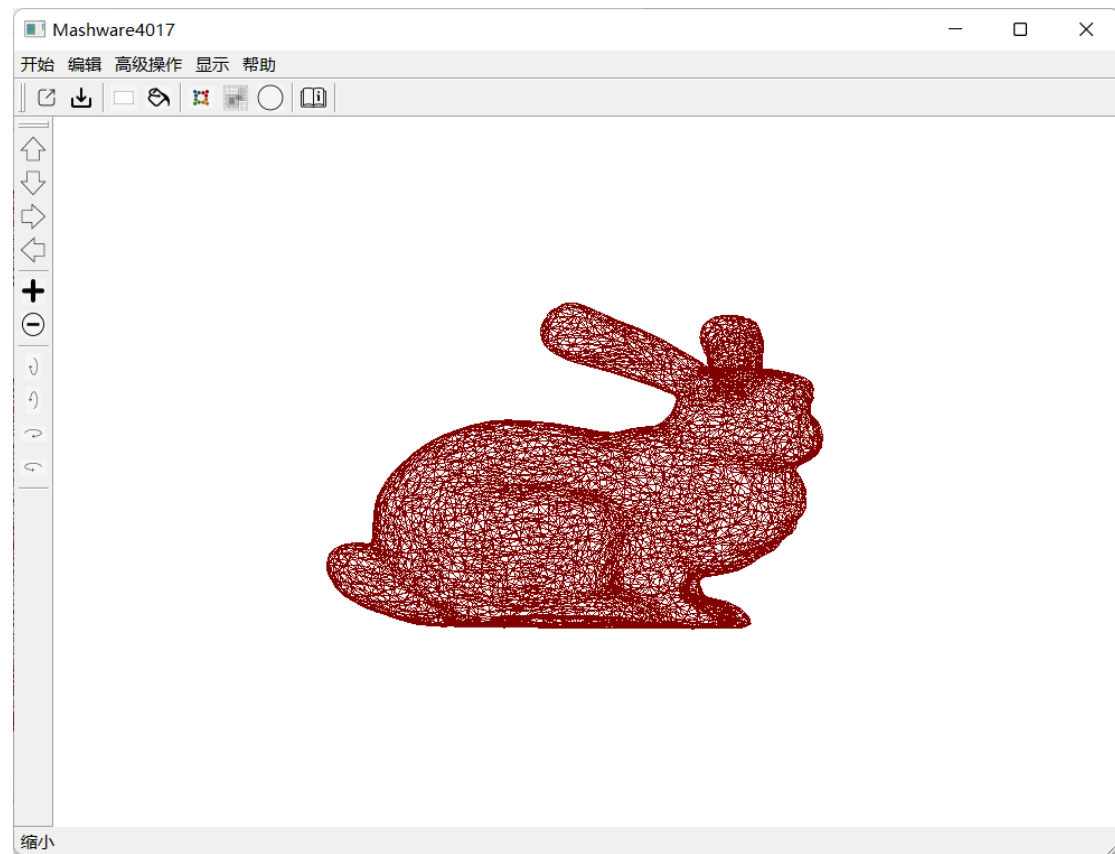




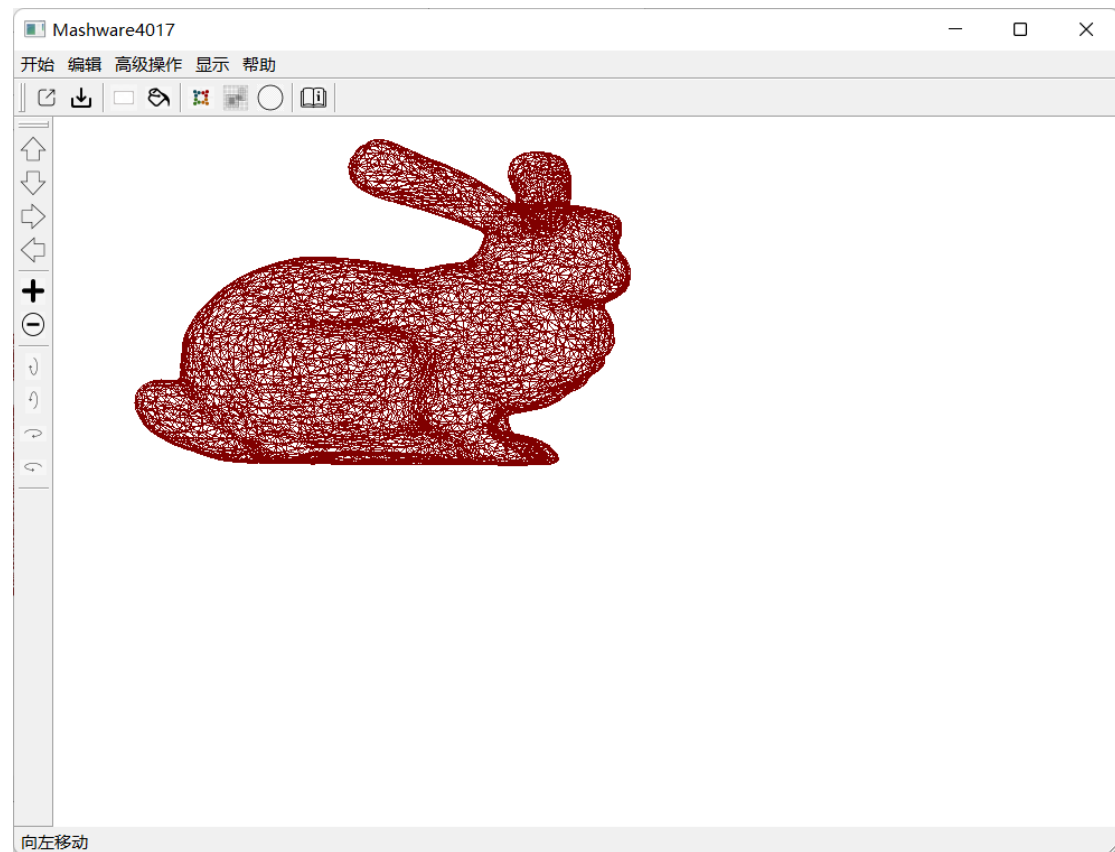
放大



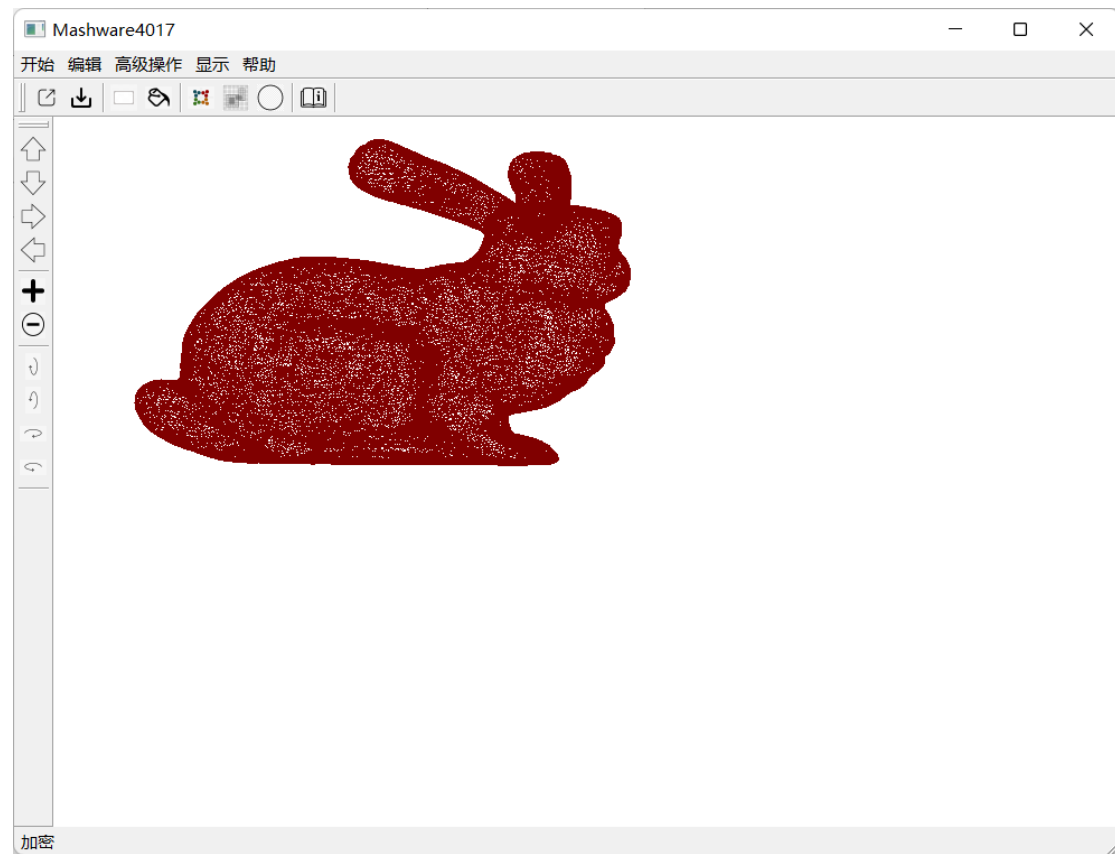
缩小



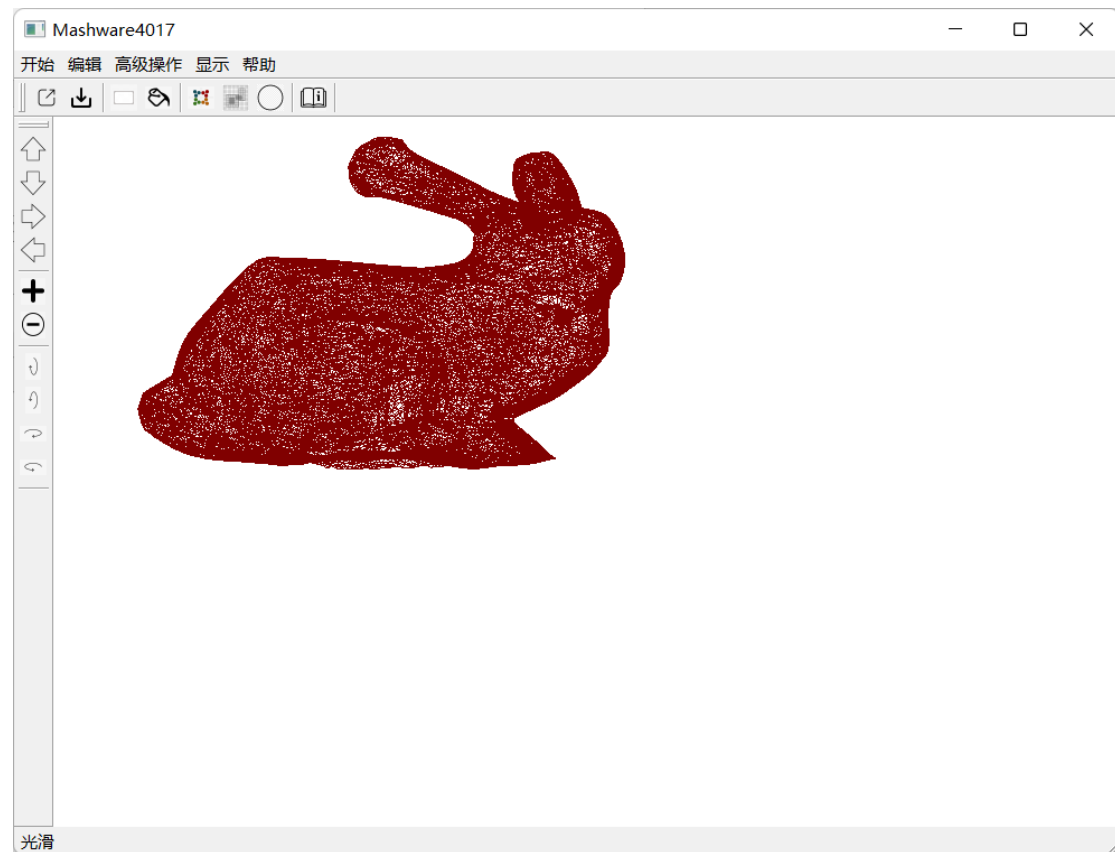
## 移动



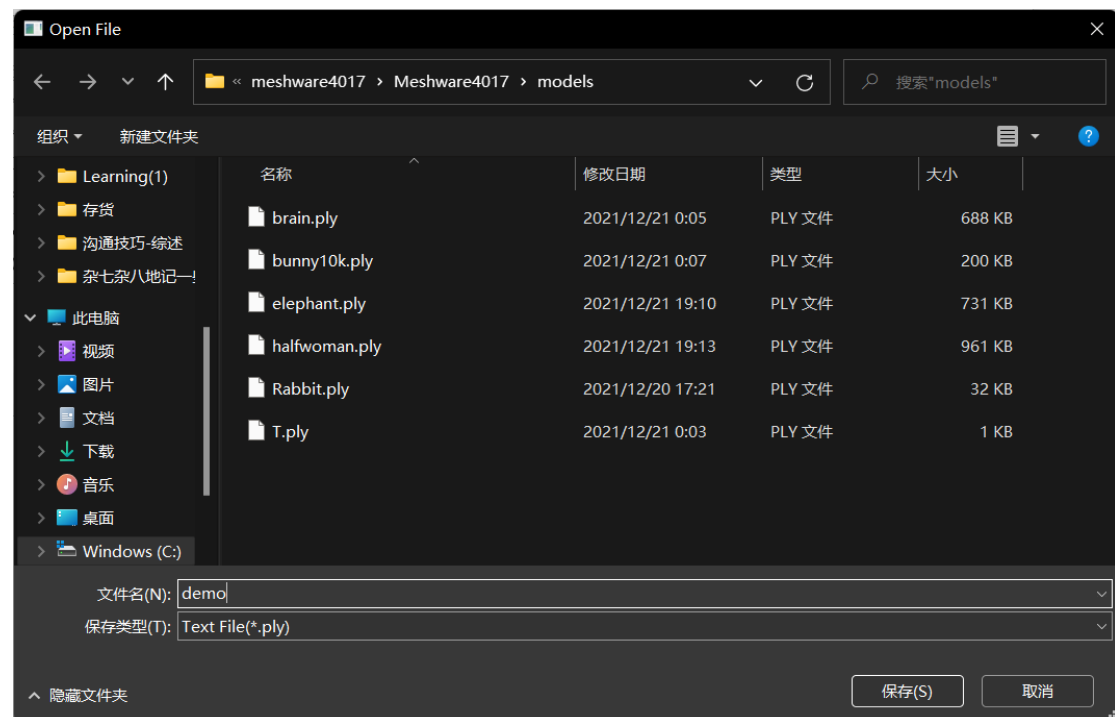
## 加密



## 光滑化



左下角的菜单栏会实时提示当前的操作  
点击保存



即可成功保存图像（如果发生权限错误，请更换安装时的目录）

## 5 总结和参考文献资料

### 5.1 编码规范

①本次代码，myvcg.h 和 mymodel.h 都包含了函数的定义，这是参考了 vcglib 的实现形式，可以方便迁移与环境的配置。

②myopenglwidget.h 中的函数均使用小驼峰命名法，mymodel.h 和 myvcg.h 里的函数均使用大驼峰命名法

③对具有特殊含义的变量，都进行了宏定义

### 5.2 收获感言

这次实验是我第一次从头开始一个人构建项目，在过程中通过网上搜集资料，咨询有相关经验的同学一些技术问题，磕磕绊绊花了一个多星期终于完成了这个项目。当然这个项目有不少差强人意的地方，比如稀疏和加密并不能兼容等等。但我从中仍然收获了许多：比如显卡的运作逻辑，opengl 规范，Qt 的使用，visual studio 的项目管理，vcglib 的使用。更重要的，就是资料的搜集能力以及官网文档的阅读真的很重要。十分有幸能够有这次动手利用现有库构建项目的机会！

### 5.3 参考文献

- [1]<https://github.com/cnr-isti-vclab/vcglib>
- [2]<https://github.com/cnr-isti-vclab/meshlab>
- [3][https://blog.csdn.net/qq\\_31804159/article/details/103228706](https://blog.csdn.net/qq_31804159/article/details/103228706)
- [4]<https://www.cnblogs.com/liangliangh/p/4165228.html>
- [5]<http://vcg.isti.cnr.it/vcglib>
- [6]<https://www.bilibili.com/video/BV1MJ411u7Bc?p=20>