# GAMES301 作业框架指南

## 安装说明

GAMES301提供两个作业框架，分别是c++的“Surface\_Framework\_Cmake”、Matlab的“matlab\_template”。Matlab框架无需配置，使用Matlab软件打开即可正常运行，在此仅介绍c++框架安装。

（1）在安装c++作业框架前，需要自行安装如下软件：

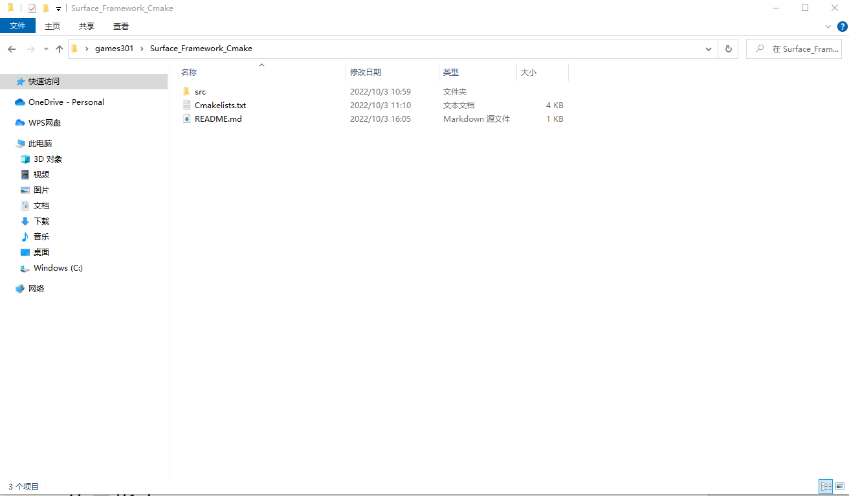
Cmake 建议3.7版本以上

Qt 建议5.4版本以上

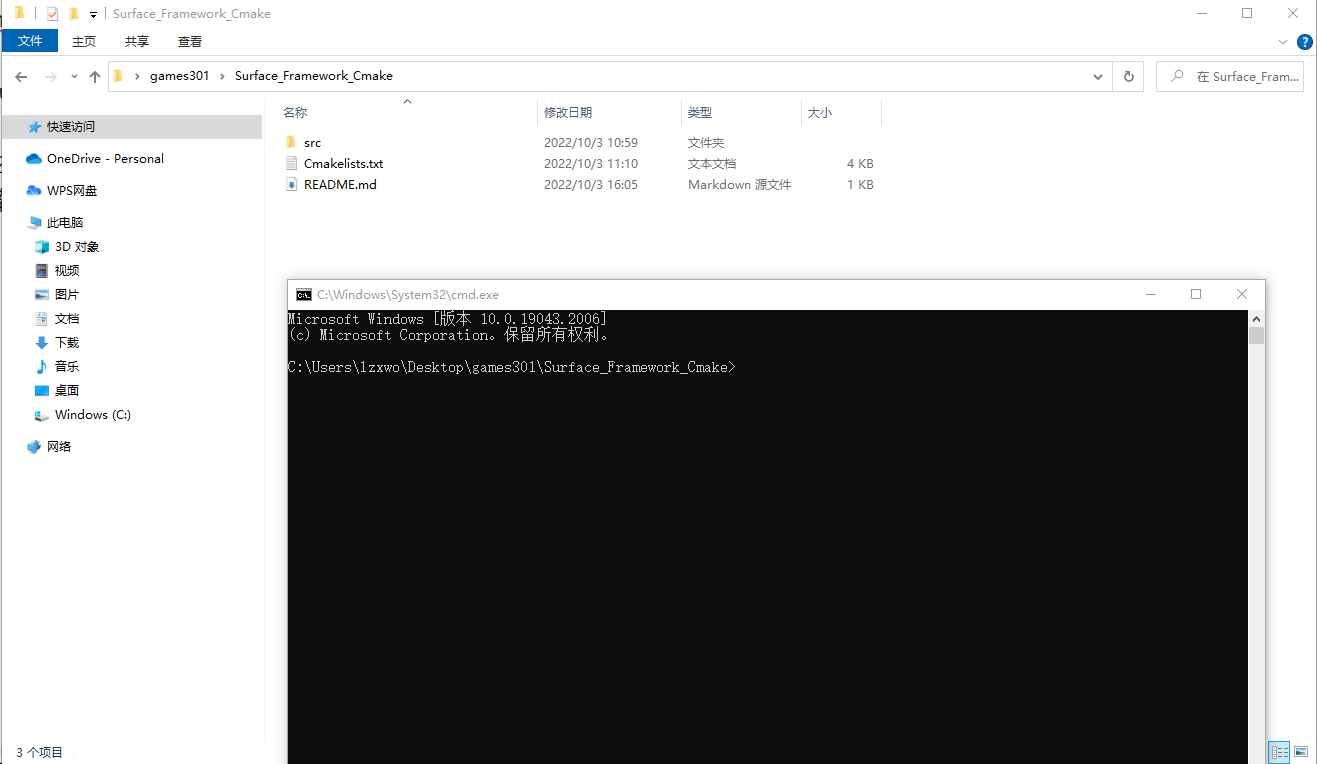
Visual studio 建议VS2017以上

（2）c++框架安装教程：

下载“Surface\_Framework\_Cmake.zip”，解压，进入“Surface\_Framework\_Cmake”目录：



在命令行窗口进入当前目录下

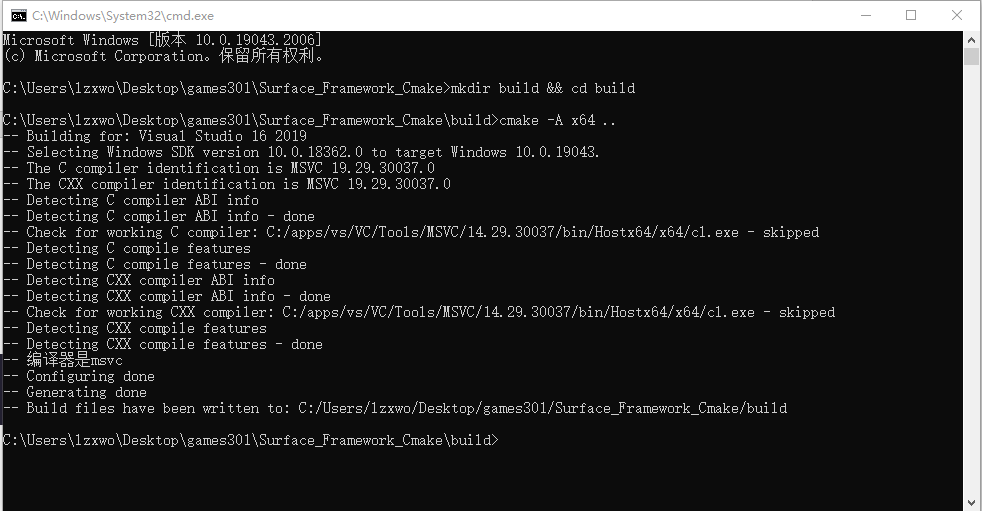


输入：

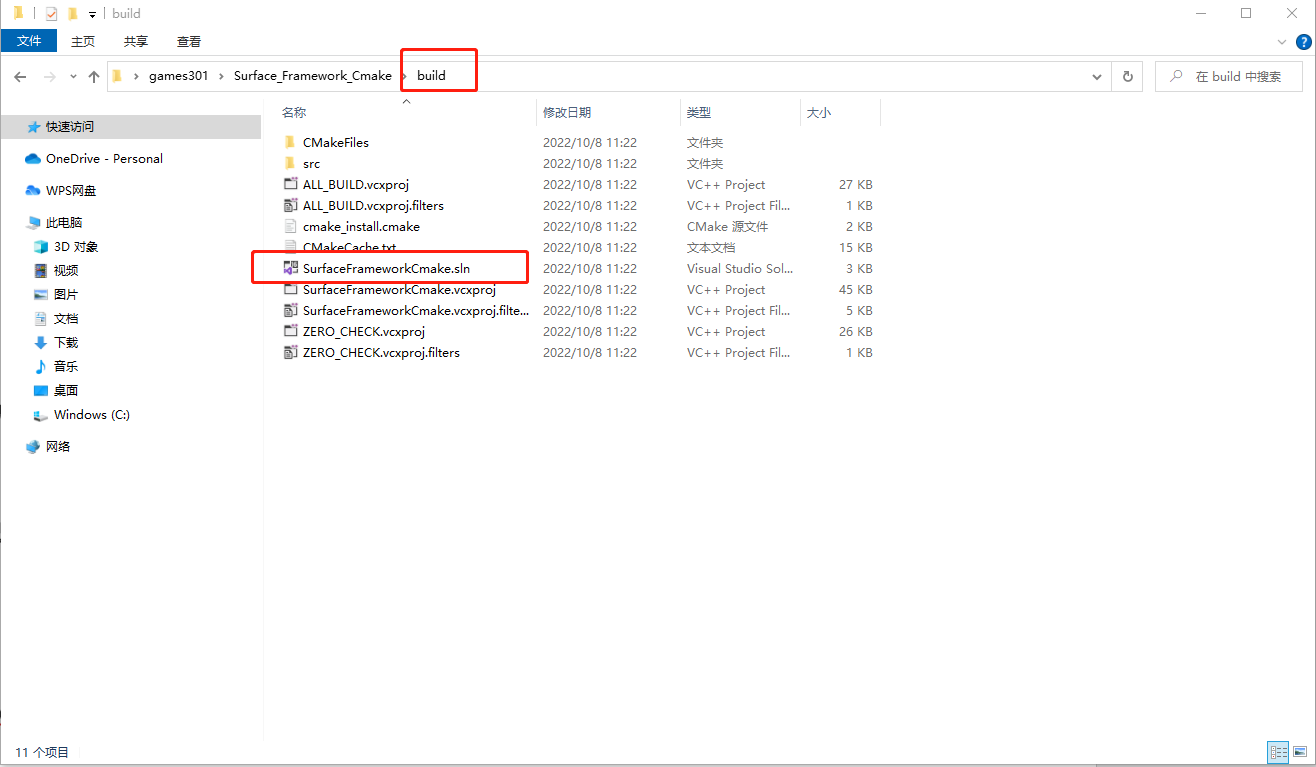
mkdir build && cd build

cmake -A x64 ..

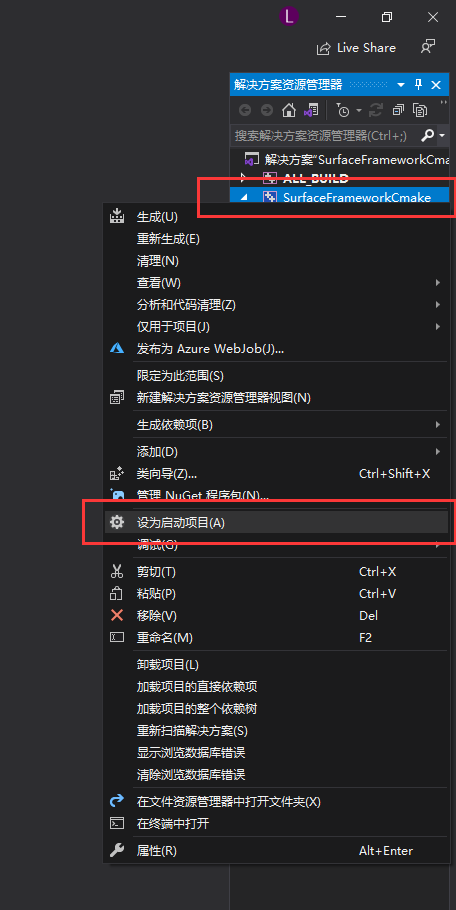
运行正常结束后：



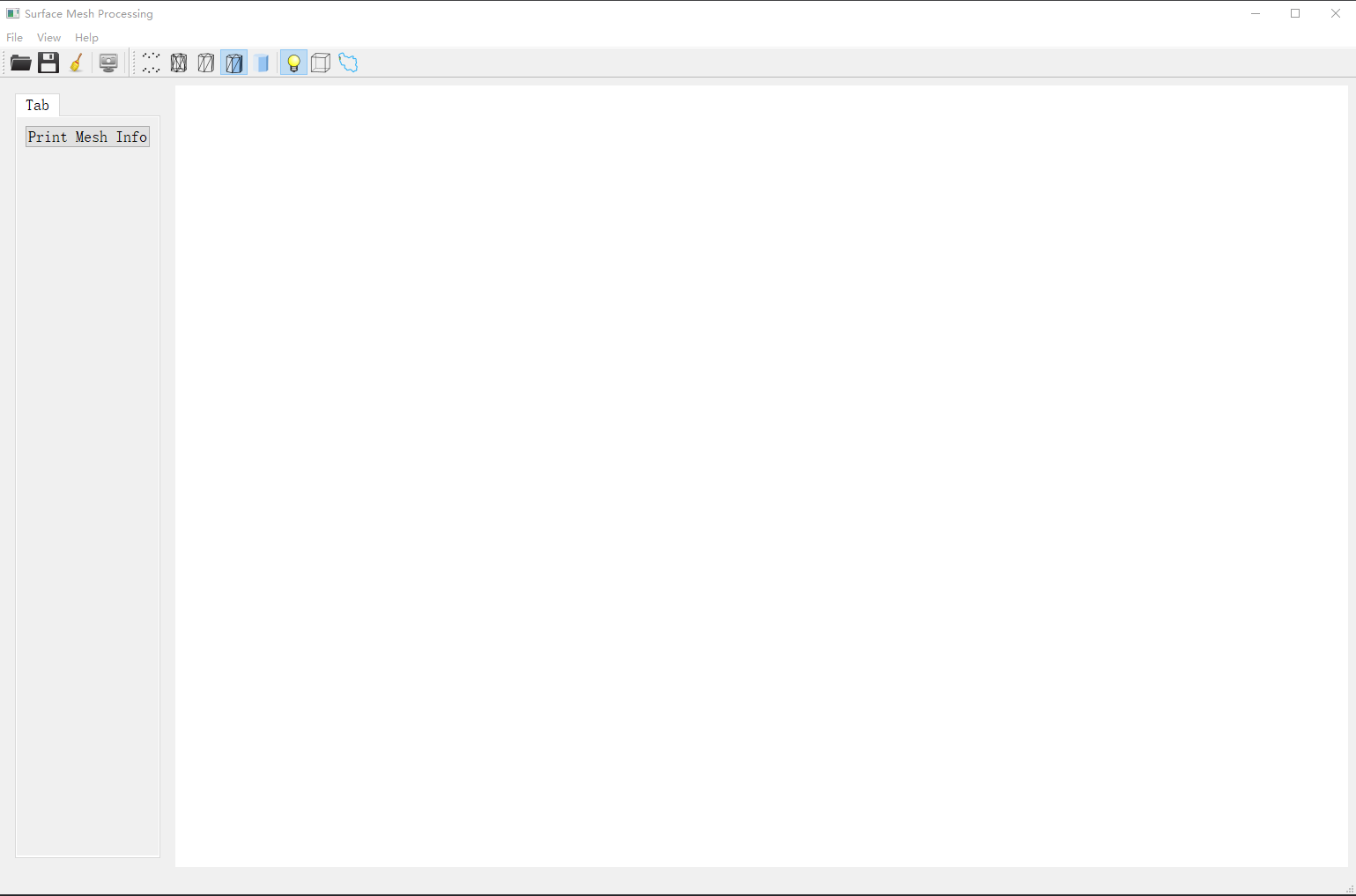
打开生成的build文件夹，点击sln文件：



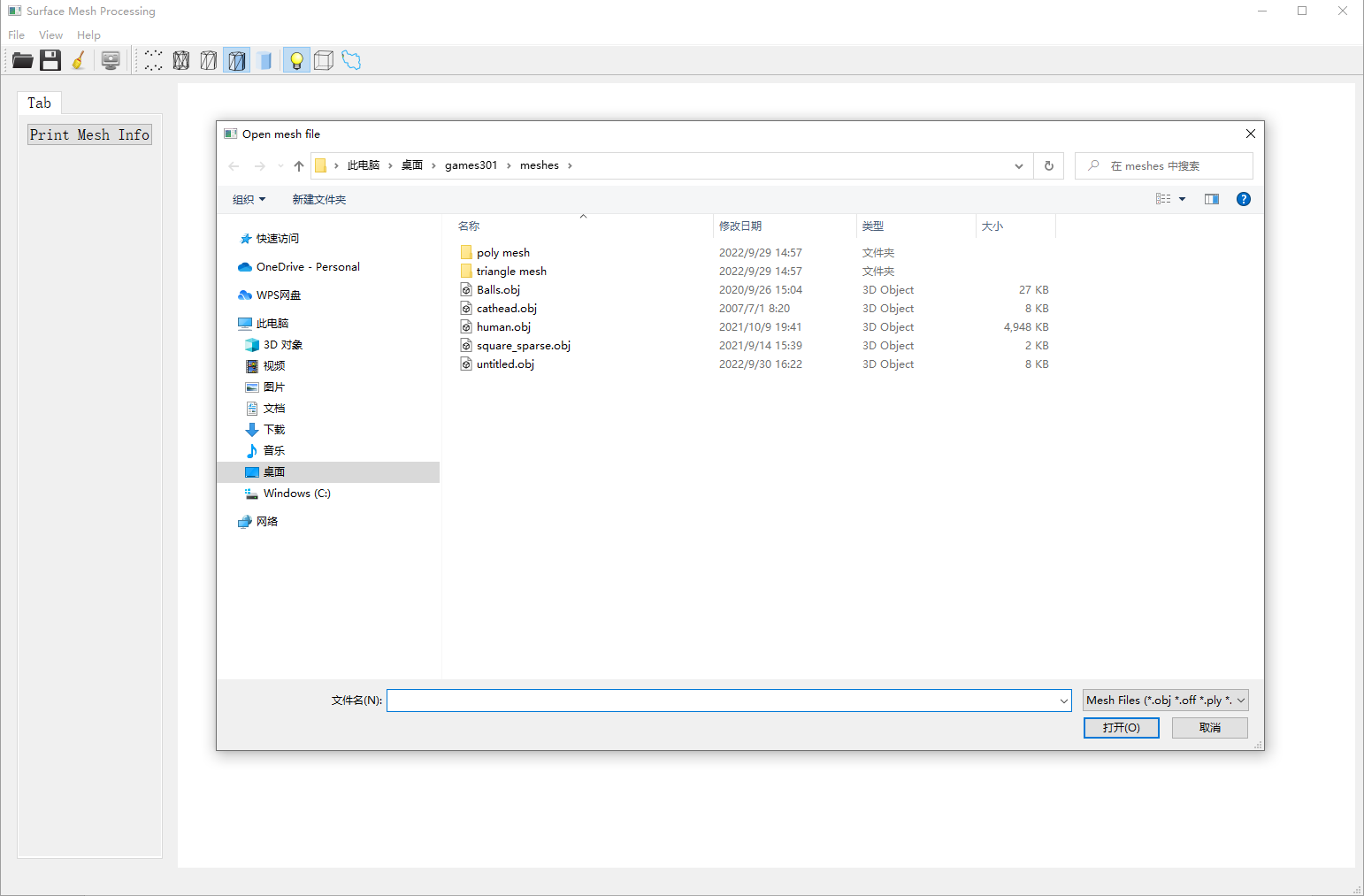
在VS软件里，选择SurfaceFrameworkCmake项目作为启动项目，点击运行

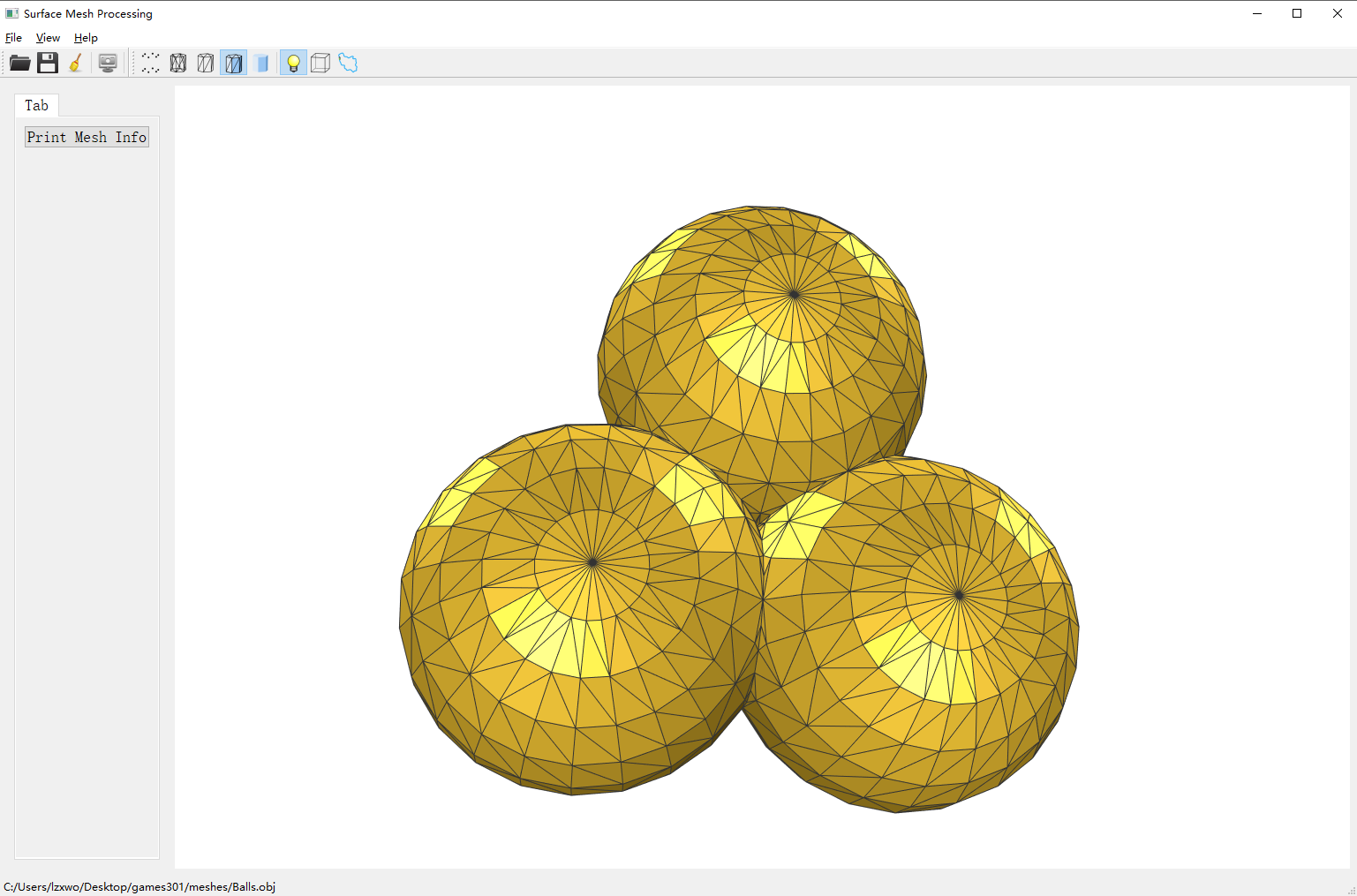


运行界面如下：



通过open mesh file按钮打开网格查看效果：





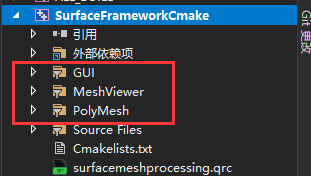
## 框架介绍

了解此框架需要三个基础：

C++：以 C++ Primer 为参考书，至少了解到类、容器、泛型算法

Qt：了解Qt的信号与槽机制，了解按钮、文本框等基本GUI组件、鼠标的输入事件

OpenGL: 了解C++中以gl为代表的函数，达到绘制基本图元、赋予颜色的水平。

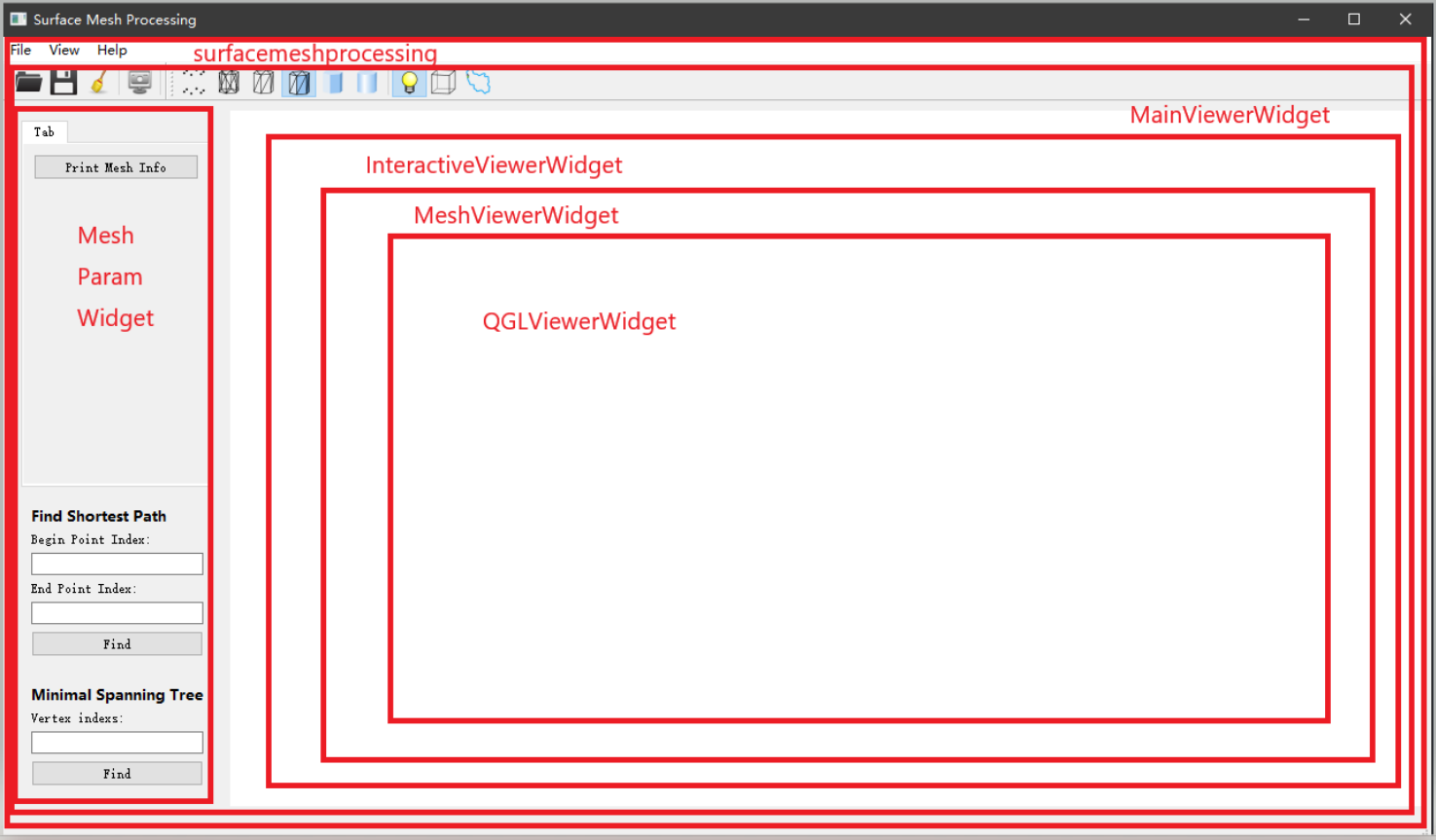


GUI中的文件设置程序界面的顶栏，侧边栏

MeshViewer中的文件与程序界面的中部OpenGL窗口有关，主要是展示模型

PolyMEsh中的文件定了网格结构

Qt程序界面布局如下所示：



以绘制网格上两点的之前的最短路径算法为例，我们把整个过程拆分为如下三步：

1. 从GUI界面接受起点、终点顶点的Index。

2. 发送到某槽函数，在Mesh上计算最短路径。

3. 在GUI界面绘制最短路径。

Step 1

在MeshParamViewer窗口中添加 QLineEdit 、 QButton 、 QLabel 等组件，接受输入顶点值和触发信号。

将值随着信号发送到MainViewerWidget中的 你定义的 槽函数中。

Step 2

在MainViewerWidget中定义一个槽函数，接受来自Step1的信号与值。在该函数中，在类成员 mesh 上计算最短路径。将最短路径存放到容易取得的位置。调用update()，更新网格绘制。

Step 3

更新网格绘制，会依次调用 update()->paintGL()->DrawScene()->DrawSceneMesh()。我们需要关注和修改的函数是MainViewerWidget中的 DrawSceneMesh ，仿照其中的几类DrawMode，写绘制最短路径的DrawMode

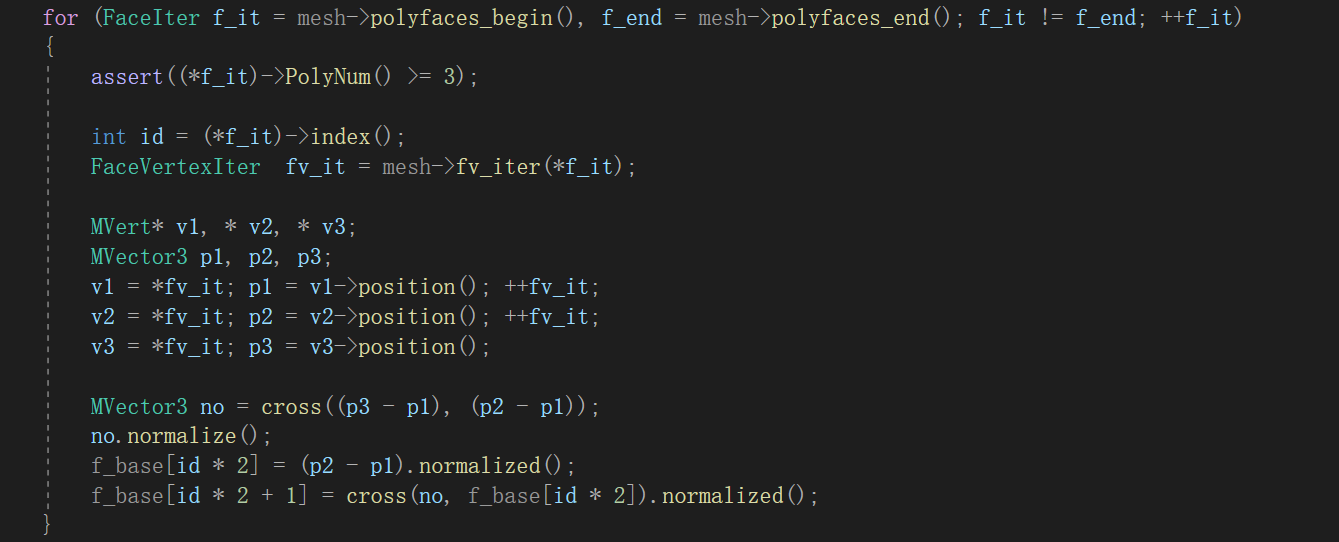
## PolyMesh使用指南

PolyMesh是一个用于表示和操作多边形网格数据结构的c++库，用户可以根据应用需要自己定制网格类型，可以提供自定义的用于表示点、边和面的数据结构或者可以方便地使用PolyMesh中预定义的结构。

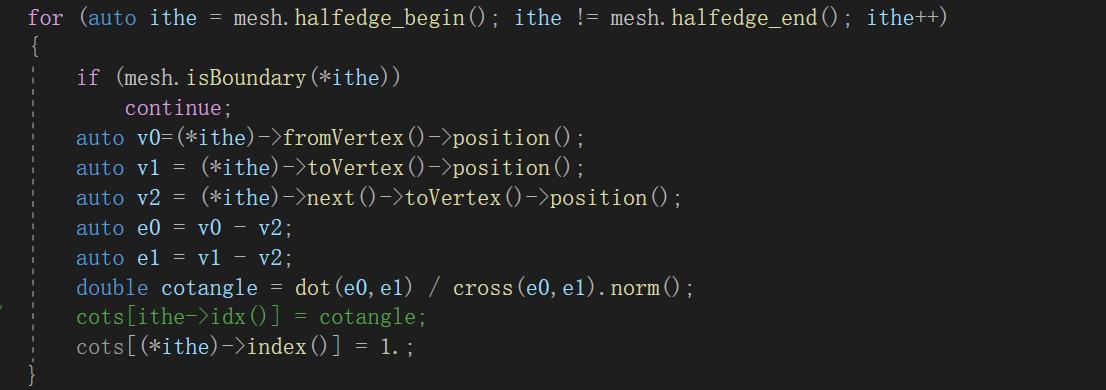
PolyMesh使用半边数据结构存储和管理网格元素（点、边、 面）和它们之间的连接关系。

PolyMesh的常见操作有遍历网格顶点、遍历网格边、遍历网格面等，如下图所示。具体示例可以参考：傅孝明老师的Github上，提供的使用PolyMesh做数字几何处理作业的示例代码：<https://github.com/USTC-GCL-F/AMMesh>

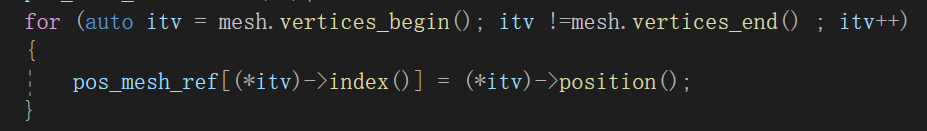
遍历网格面：



遍历半边：



遍历点：



## 其他说明

第三方库OpenMesh也可以进行网格操作，我们也提供使用openmesh库的作业框架，如果想使用openmesh库的同学，可以从傅孝明老师主页（https://ustc-gcl-f.github.io/code/index.html#sec\_surface\_framework）中下载使用第三方库的框架：

