

iGameMeshView软件介绍及安装方法

目录

- 1、说明
- 2、环境安装
- 3、软件安装
- 4、使用方法

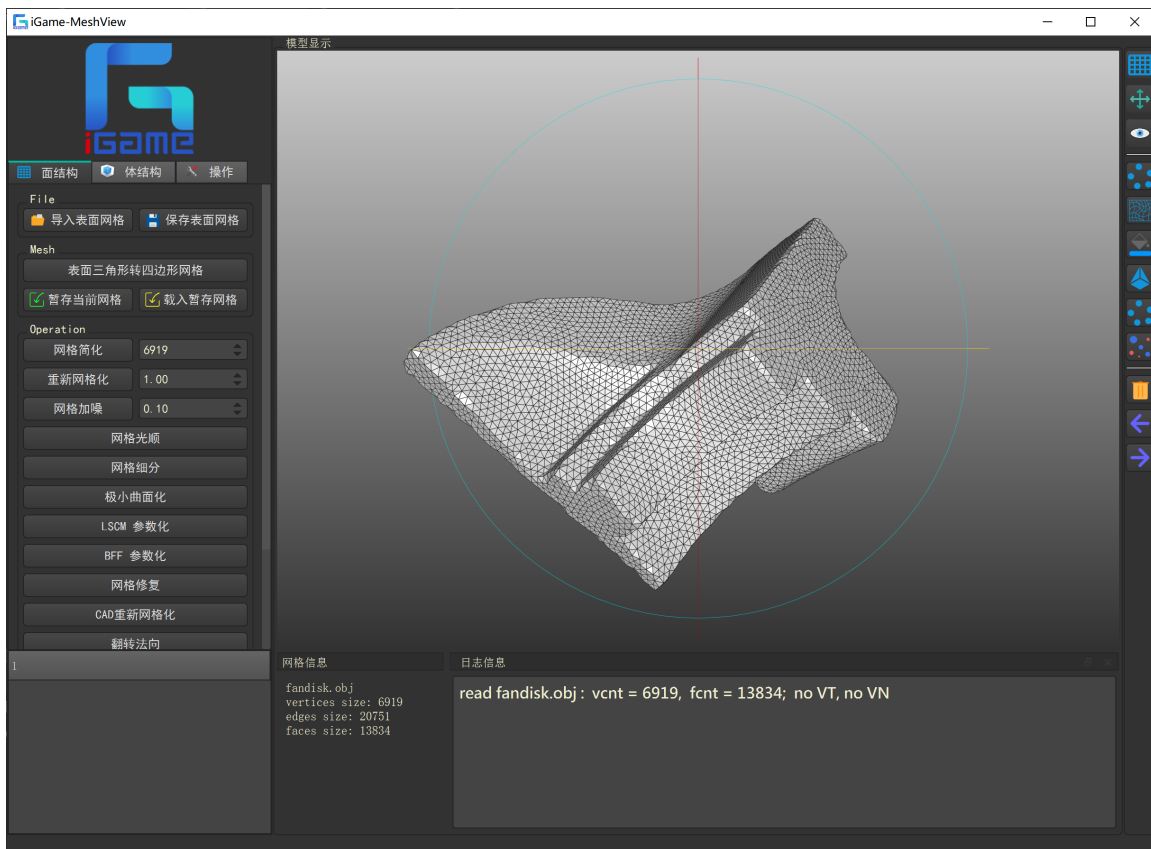
1、说明

iGameMeshView是iGame实验室制作的一个网格可视化与数字几何处理软件

主要功能为：

- 1、加载网格模型，并可视化
- 2、对网格模型进行几何处理

基于C++编程，使用Qt+OpenGL实现可视化



本框架的**核心数据结构**部分由实验室的谢津师兄编写

渲染可视化部分主要由实验室的郑伟伟和张盛编写

使用前提提示：

使用此软件框架需要有基本图形学知识和基础OpenGL、Qt知识，建议学习了GAMES101课程和learnOpenGL基础课（“模型加载”之前的章节）之后再使用此框架，到那个时候再安装软件也不迟，GAMES102和数字几何处理的课程作业可以使用此框架编写网格算法

（说实话，如果仅仅是拿它用来写网格算法，不是为了深入研究此软件的代码，其实是不需要OpenGL、Qt知识的）

2、环境安装

此框架使用cmake方式编译，支持Windows、linux、 MacOS全平台安装使用
安装框架前需要先安装：

- 1、一个C++的集成开发环境IDE，推荐：Visual Studio 2019 、CLion
- 2、 Qt

Windows用户

1、IDE的下载

Windows用户可以选择Visual Studio或者CLion，编译器选择MSVC（下载Visual Studio之后自带）

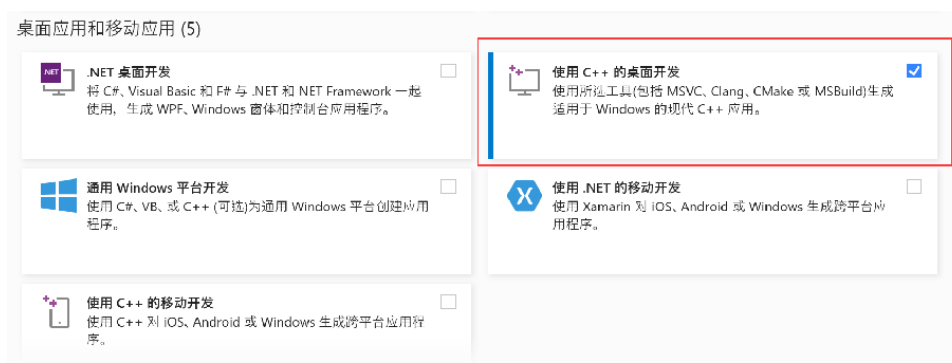
因为只有下载了VS才能用MSVC，所以要使用CLion的话，需要下载VS，然后编译器才能选择MSVC

VS下载：

官网下载地址：<https://visualstudio.microsoft.com/zh-hans/vs/>

VS2017/2019/2022任意版本都可以使用，community版和professional版都可以，自己挑一个

安装时只需勾选C++桌面开发，如图所示：



CLion下载：

官网：<https://www.jetbrains.com/clion/download/#section=windows>

CLion是收费的，但是学生可以免费使用，只需要有edu的邮箱就可以免费申请

CLion最近更新已经有中文界面了，如果打开还是英文界面，更新到最新版就可以了

2、Qt下载

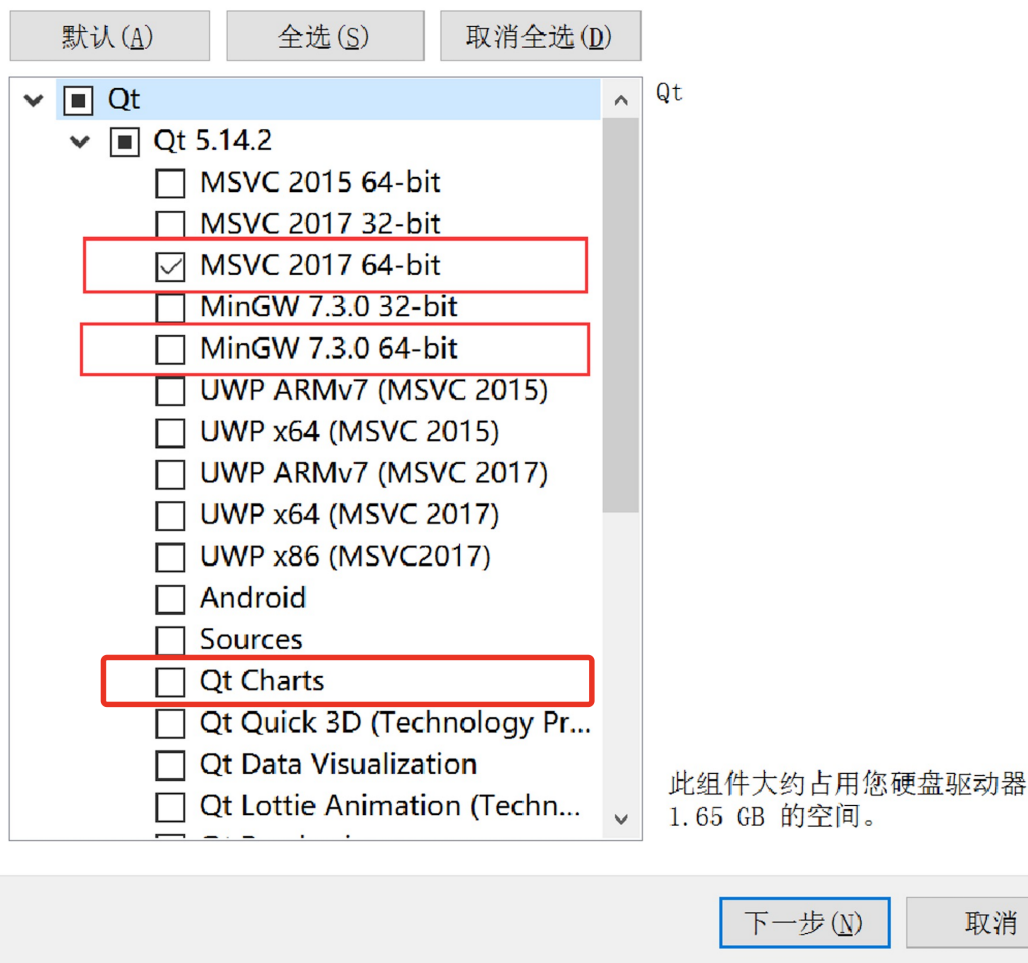
选择Qt5.14.2的Windows版本

下载地址：<https://download.qt.io/archive/qt/5.14/5.14.2/>

安装时只需勾选红框里的项目，MSVC版本必选，Qt Charts必选，MinGW版本可不选

选择组件

请选择要安装的组件。



Linux用户

1、IDE的下载

CLion下载：

官网：<https://www.jetbrains.com/clion/download/#section=linux>

CLion是收费的，但是学生可以免费使用，只需要有edu的邮箱就可以免费申请

CLion最近更新已经有中文界面了，如果打开还是英文界面，更新到最新版就可以了(或者去插件商店下载汉化语言包)

2、Qt下载

选择Qt5.14.2的linux版本

下载地址：<https://download.qt.io/archive/qt/5.14/5.14.2/>

安装gcc版本和Qt Charts，没有安装图，因为我没怎么用过linux

MacOS用户

注意！新版的Mac使用arm64架构的苹果自研芯片Apple Silicon（M1/M1X/M1U/M2芯片），注意留意你的芯片是Intel版的还是Apple Silicon版的，安装流程会有所不同

Intel版本

1、IDE的下载

CLion下载：（选择Intel版本）

官网：<https://www.jetbrains.com/clion/download/#section=mac>

CLion是收费的，但是学生可以免费使用，只需要有edu的邮箱就可以免费申请

CLion最近更新已经有中文界面了，如果打开还是英文界面，更新到最新版就可以了(或者去插件商店下载汉化语言包)

2、Qt下载

选择Qt5.14.2的mac版本

下载地址：<https://download.qt.io/archive/qt/5.14/5.14.2/>

安装apple clang版本和Qt Charts，没有安装图，因为我没用过Intel版的Mac

Apple Silicon版本

1、IDE的下载

需要先下Xcode，在应用商店下载，因为Xcode里有运行代码需要的SDK

然后是CLion下载：（选择Apple Silicon版本）

官网：<https://www.jetbrains.com/clion/download/#section=mac>

CLion是收费的，但是学生可以免费使用，只需要有edu的邮箱就可以免费申请

CLion最近更新已经有中文界面了，如果打开还是英文界面，更新到最新版就可以了(或者去插件商店下载汉化语言包)

2、Qt下载和安装

因为Qt在5.14版本以后，不再提供各平台的安装包，而是直接提供源码的形式让用户自行编译，所以就没有arm64版本的mac安装包了，需要我们自己用源码编译。再者，苹果自研芯片出的时间不久，网上这类相关的教材很少，所以安装起来非常麻烦，但还是有方法的

项目文档里还有一份pdf《Build Qt in arm64》，里面有详细的qt安装方法

3、软件安装

1、把iGameMeshView_lite文件夹放到一个没有中文路径的地方（不建议放C盘，会有些权限问题）

2、打开IDE，通过“打开文件夹”的方式打开iGameMeshView_lite项目

3、修改CMakeList.txt里的内容（只用把其中的Qt路径改成你自己安装的Qt的路径即可）

路径请使用反斜杠“/”（如图中所示），或者转义“\\”，不要使用“\”

linux用户修改此项：

```
#编译器
# linux平台的GCC编译器
if ("${CMAKE_CXX_COMPILER_ID}" STREQUAL "GNU")
    message(STATUS "is gcc")
    set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -fpermissive -O2 -pthread")
    #设置QT的目录
    set(QT_CMAKE_PATH '/opt/Qt5.14.2/5.14.2/gcc_64/lib/cmake')
    set(Qt5Core_DIR ${QT_CMAKE_PATH}/Qt5Core)
    set(Qt5Gui_DIR ${QT_CMAKE_PATH}/Qt5Gui)
    set(Qt5OpenGL_DIR ${QT_CMAKE_PATH}/Qt5OpenGL)
    set(Qt5Widgets_DIR ${QT_CMAKE_PATH}/Qt5Widgets)
```

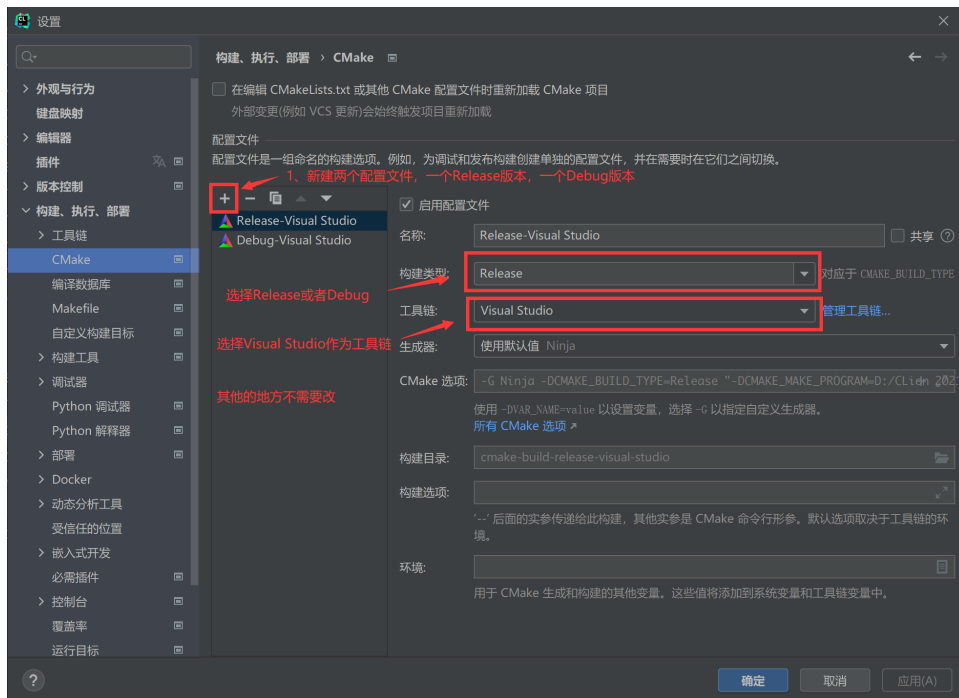
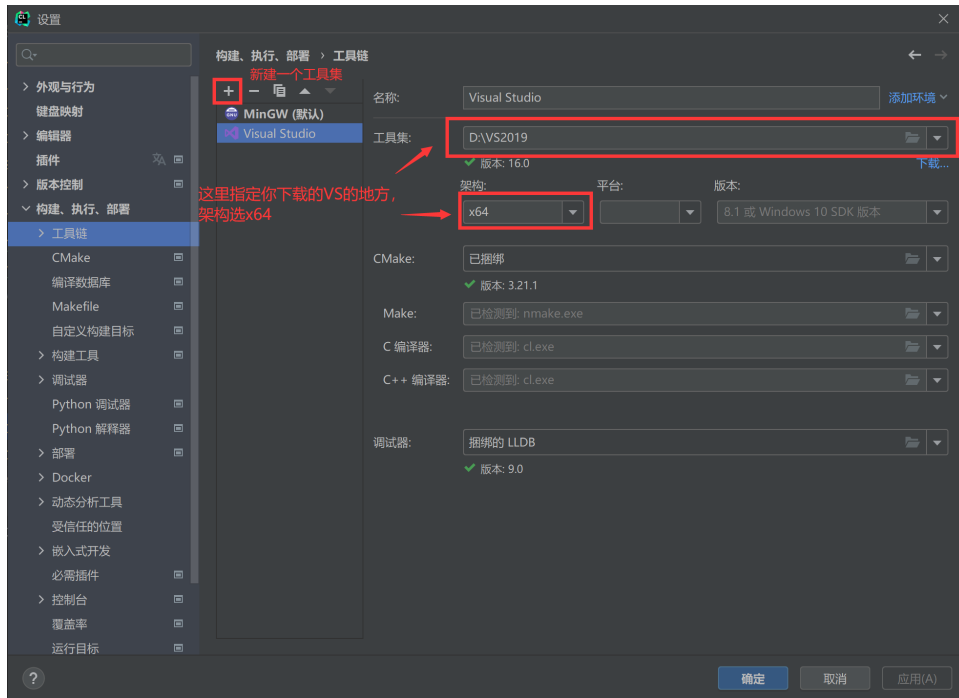
Windows用户修改此项：

```
# Windows平台的MSVC编译器
elseif ("${CMAKE_CXX_COMPILER_ID}" STREQUAL "MSVC")
    message(STATUS "The Compiler is MSVC")
    #设置QT的目录
    set(QT_CMAKE_PATH "E:/Qt/5.14.2/msvc2017_64/lib/cmake")
    set(Qt5Core_DIR ${QT_CMAKE_PATH}/Qt5Core)
    set(Qt5Gui_DIR ${QT_CMAKE_PATH}/Qt5Gui)
    set(Qt5OpenGL_DIR ${QT_CMAKE_PATH}/Qt5OpenGL)
    set(Qt5Widgets_DIR ${QT_CMAKE_PATH}/Qt5Widgets)
```

使用VS的同学可以无视以下步骤

你可以在CLion中使用MSVC作为你的编译器，前提是下载了VS的MSVC


如果你下载了CLion，并打算使用MSVC编译器，请在“设置”里的工具链和CMake中指定一下MSVC的路径，如下图所示：



MacOS用户修改此项：

```
# MacOS平台的Clang编译器
elseif ("${CMAKE_CXX_COMPILER_ID}" STREQUAL "AppleClang")
    message(STATUS "The Compiler is AppleClang")
    #设置QT的目录
    set(QT_CMAKE_PATH "/Users/halforange/Downloads/buildir/outDir/lib/cmake")
    set(Qt5Core_DIR ${QT_CMAKE_PATH}/Qt5Core)
    set(Qt5Gui_DIR ${QT_CMAKE_PATH}/Qt5Gui)
    set(Qt5OpenGL_DIR ${QT_CMAKE_PATH}/Qt5OpenGL)
    set(Qt5Widgets_DIR ${QT_CMAKE_PATH}/Qt5Widgets)
```

4、重新加载CMake项目，构建，运行

 从 CMake 收集信息后将刷新 C++ IntelliSense 信息。 [取消](#) - [设置](#)

4、疑难杂症

1、使用VS进行编译代码时，可能会发生编译错误的情况：

```
iGameMeshView_Lite\MeshKernel\Handle.h(9): error C2059: 语法错误：“=”
iGameMeshView_Lite\MeshKernel\Handle.h(9): error C2238: 意外的标记位于“;”之前
iGameMeshView_Lite\MeshKernel\Handle.h(10): error C2059: 语法错误：“return”
iGameMeshView_Lite\MeshKernel\Handle.h(10): error C2238: 意外的标记位于“;”之前
iGameMeshView_Lite\MeshKernel\Handle.h(14): error C2143: 语法错误：缺少“;”（在“&”的前面）
iGameMeshView_Lite\MeshKernel\Handle.h(14): error C4430: 缺少类型说明符 - 假定为 int。注意：C++ 不支持默认 int
iGameMeshView_Lite\MeshKernel\Handle.h(14): error C2801: “MeshKernel::operator =”必须是非静态成员
iGameMeshView_Lite\MeshKernel\Handle.h(14): error C2059: 语法错误：“{”
iGameMeshView_Lite\MeshKernel\Handle.h(14): error C2143: 语法错误：缺少“;”（在“{”的前面）
iGameMeshView_Lite\MeshKernel\Handle.h(14): error C2447: “{”：缺少函数标题(是否是老式的形式表?)
iGameMeshView_Lite\MeshKernel\Handle.h(20): error C2270: “is_valid”：非成员函数上不允许修饰符
iGameMeshView_Lite\MeshKernel\Handle.h(20): error C2065: “idx_”：未声明的标识符
iGameMeshView_Lite\MeshKernel\Handle.h(22): error C4430: 缺少类型说明符 - 假定为 int。注意：C++ 不支持默认 int
iGameMeshView_Lite\MeshKernel\Handle.h(22): error C2143: 语法错误：缺少“,”（在“&”的前面）
iGameMeshView_Lite\MeshKernel\Handle.h(22): error C2270: “<”：非成员函数上不允许修饰符
iGameMeshView_Lite\MeshKernel\Handle.h(22): error C2803: “operator <”必须至少有一个类类型的形参
iGameMeshView_Lite\MeshKernel\Handle.h(22): error C2805: 二进制“operator <”的参数太少
iGameMeshView_Lite\MeshKernel\Handle.h(22): error C2355: “this”：只能在非静态成员函数或非静态数据成员初始值设定项的内部引用
```

这主要是代码里的中文注释的原因，因为代码是UTF-8格式的，而VS是不支持UTF-8的（只支持UTF-8 with BOM），所以会导致编译错误

解决方法：在cmake中增加MSVC的utf-8 编译选项，这样可以避免UTF-8中文的问题：

#以UTF-8格式进行编译

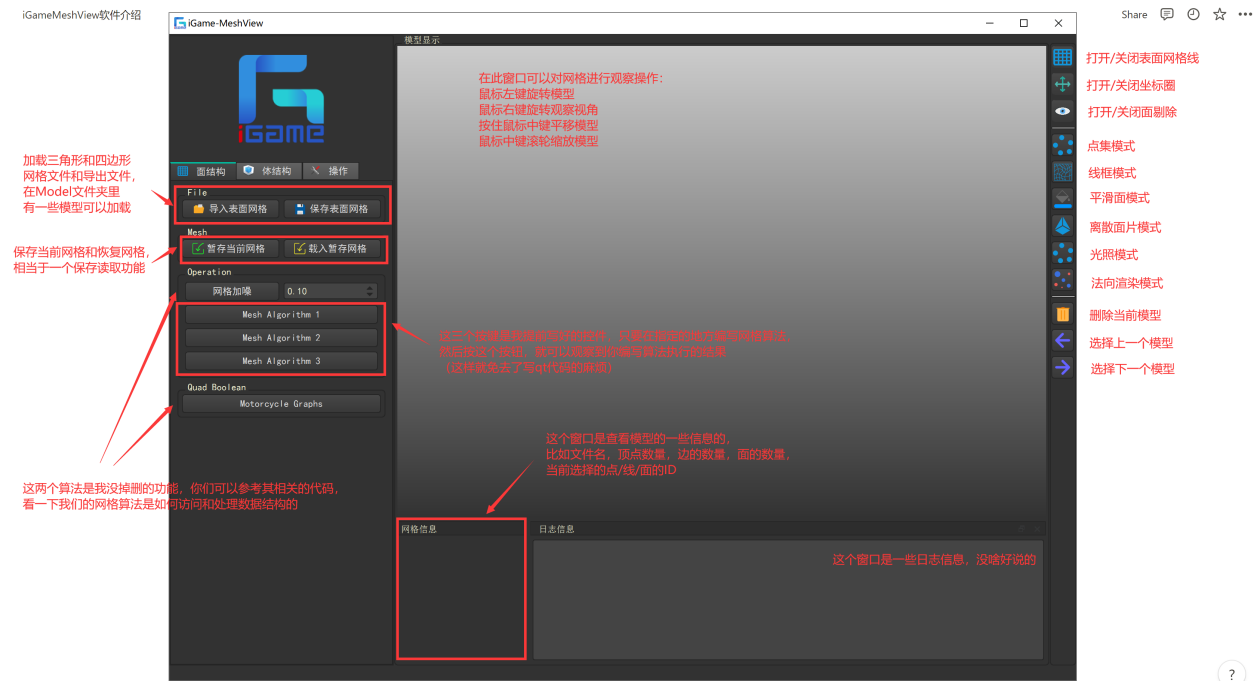
```
add_compile_options("$<$<C_COMPILER_ID:MSVC>:/utf-8>")
```

```
add_compile_options("$<$<CXX_COMPILER_ID:MSVC>:/utf-8>")
```

5、使用方法

界面使用方法

展示界面可能与实际界面稍有不同，但功能大差不差



我们在Model文件夹里放了一些模型，可以导入进来看看，你也可以自己找一些网格文件加载

此框架现在只支持加载以下类型的网格文件：

表面网格：三角形和四边形（或多边形），文件格式支持.obj、.off、.stl

体网格：四面体和六面体，文件格式仅支持.mesh

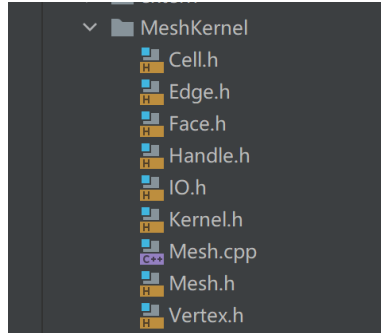
(碎碎念：其实鲁棒性不是很好，不一定都能加载，见谅。有能力的同学可以看看源码，优化一下)



目前观察模式已经分成了**编辑模式**和**观察模式**，其中编辑模式是对**模型**进行缩放和旋转，而观察模式是对**视角**进行缩放和旋转，在单模型可视化的情况下操作差别不大，在多模型的可视化的情况下操作会有所不同

6、核心代码学习

也就是MeshKernel里的几个文件，是谢师兄自己写的网格数据结构，这几个文件的代码是需要精读的，只有了解了其数据结构，才能编写自己的网格算法



这里推荐一下阅读顺序：

Kernel.h → Handle.h → Vertex.h → Edge.h → Face.h → Cell.h → Mesh.h → Mesh.cpp → IO.h

其中运用了C++面向对象和多态的思想，如果之前没有系统学过C++，阅读起来可能会比较吃力，不过精读此代码，可以提高你的C++能力，可以学习底层核心代码是如何一步一步构建起来的

学习完核心代码后，再结合GAMES102的课程，就可以试着编写自己的网格算法啦

祝您云路鹏程，前程似锦