1. 计算三角网格中每个三角形的面积 (10分)

此函数已经被声明为： Scalar calc\_facet\_area(const FacetHandle& \_fh); //给定一个三角面//片，计算它的面积

Tips: 可以根据程序输出结果判定是否计算正确，例如，输入模型cow.off，它的所有三角形面片中，面积最大的三角形面片的面积为：The maximal area of the mesh is: 0.0494426

1. 计算三角网格中每个顶点(vertex)的法向 (20分)

此函数已经被声明为：inline Normal calc\_normal(const VertexHandle& \_vh);///计算顶点的法向值

Tips: 注意法向要归一化。

1. 用OpenGL把三角网格中每个三角形面片的法向画出来（在每个三角面片的重心处画） (20分)

此函数已经被声明为：bool ogl\_writer2(bool \_orient = true, bool \_smooth = false);

Tips: 可以参照函数bool ogl\_writer(bool \_orient = true, bool \_smooth = false)的来实现，这时只需要在每个三角形面片中画出法向即可（法向用Line表示，也可以在绘制出的直线上加上箭头（代表方向）。按键盘上”m”可以查看画出面片法向后的结果。）

1. 实现一种三角网格去噪算法（例如Laplacian smoothing （全局或者局部Laplacian Smoothing），双边滤波 (bilateral mesh denoising “Shachar Fleishman, Iddo Drori, Daniel Cohen-Or: Bilateral mesh denoising. ACM Trans. Graph. 22(3): 950-953 (2003)” 或者 引导双边滤波(guided mesh normal filtering, “Wangyu Zhang, Bailin Deng, Juyong Zhang, Sofien Bouaziz, Ligang Liu: Guided Mesh Normal Filtering. Comput. Graph. Forum 34(7): 23-34 (2015)”), L0方法”Lei He, Scott Schaefer: Mesh denoising via L0 minimization. ACM Trans. Graph. 32(4): 64:1-64:8 (2013)”等等）(30分)

此函数声明为：void Laplacian\_Smoothing();//////////////////实现一种三角网格去噪算法

Tips: 可以参照已经实现的两个去噪算法；按键盘上”b”可以查看去噪结果。

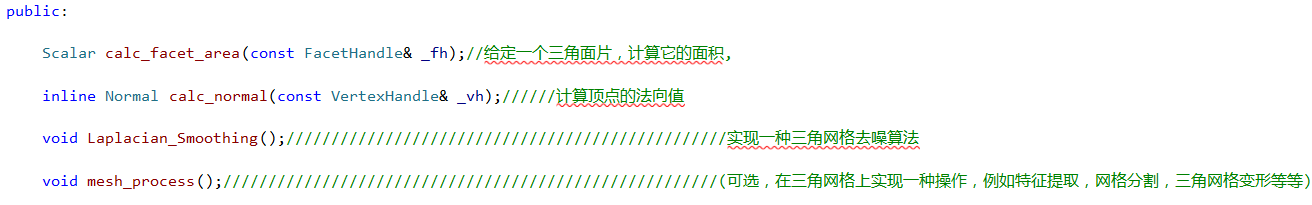
1. 实现一种基于三角网格的操作（例如，特征点/边提取，给三角网格添加随机高斯噪声，求三角网格每个顶点的高斯曲率，三角网格简化，三角网格四边形化，平面参数化、平面投影操作等等） (20分)

此函数声明为: void mesh\_process();

以下为提示信息

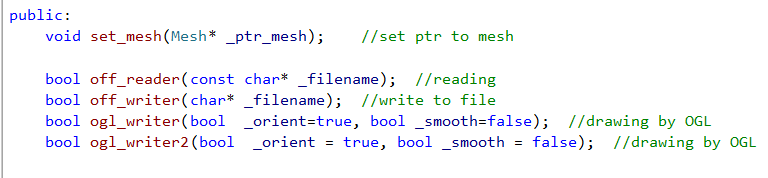
1. 其中1. 2. 4. 5的函数声明在 mesh/extension/ExKernelT.h (截图如下)

函数实现请在mesh/extension/ExkernelT.cpp 中完成 (务必)



1. 其中3的函数声明在read\_write/read\_write.h

函数实现请在read\_write/read\_write.cpp 中完成 (务必)



1. 请把整个代码看明白，里面实现了各种基本操作，例如求三角网格中每个三角形的重心、求每个三角面片的法向等等。
2. 程序运行截图如下：

|  |
| --- |
|  |

1. 鼠标点击右键，会出现一个操作界面，里面有一些灯光和其他绘制效果。

|  |
| --- |
|  |