

Geometric Modeling

2013 Spring

Programming Assignment #0 Environment Setup

What you need

- ▶ C++

Install Qt 4.8.4 first, and then Visual Studio Add-in.

- ▶ Compiler

- ▶ Microsoft Visual Studio 2008
- ▶ Visual Studio Add-in 1.1.11 for Qt4

→ combine Qt compiler with VS

- ▶ Library

- ▶ CGAL 3.8 (require Boost library)
- ▶ Boost 1.47
- ▶ Qt 4.8.4 for Windows VS 2008
- ▶ libQGLViewer 2.3.17

→ maintain 3D mesh

→ develop GUI

→ nice OpenGL based viewer

- ▶ 3D Models

- ▶ Basic wavefront file (.obj) without texture and normal information

Configuration

▶ Include Path

- ▶ CGAL : `[CGAL PATH]\include`
- ▶ gmp : `[CGAL PATH]\auxiliary\gmp\include`
- ▶ Boost : `[Boost PATH]\boost_I_47\boost`
- ▶ libQGLViewer : `[QGLViewer PATH]\QGLViewer`
- ▶ Qt : `$(QTDIR)*`

▶ Linking Path

- ▶ CGAL : `[CGAL BUILT PATH]\lib`
- ▶ gmp : `[CGAL PATH]\auxiliary\gmp\lib`
- ▶ Boost : `[Boost PATH]\boost_I_47\lib`
- ▶ libQGLViewer : `[QGLViewer PATH]\lib`
- ▶ Qt : `$(QTDIR)\lib`

Configuration

▶ Additional Library Dependencies:

▶ CGAL :

- ▶ Debug : CGAL-vc90-mt-gd.lib
- ▶ Release : CGAL-vc90-mt.lib

▶ Qt :

- ▶ Debug : qtmaind.lib QtCored4.lib QtGui4.lib QtXml4.lib QtOpenGLd4.lib
- ▶ Release : qtmain.lib QtCore4.lib QtGui4.lib QtXml4.lib QtOpenGL4.lib

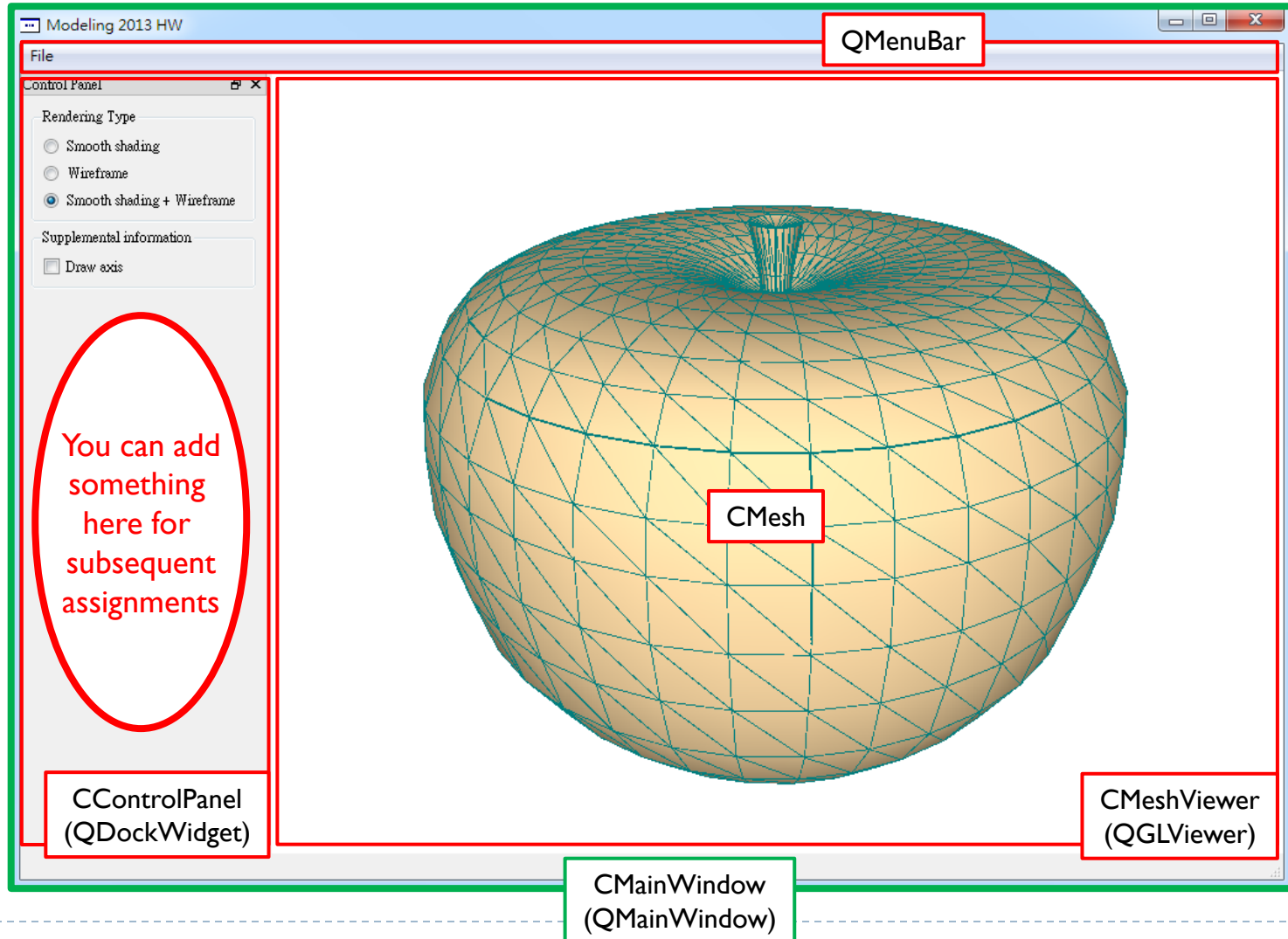
▶ libQGLViewer :

- ▶ Debug : QGLViewerd2.lib
- ▶ Release : QGLViewer2.lib

▶ OpenGL :

- ▶ opengl32.lib glu32.lib

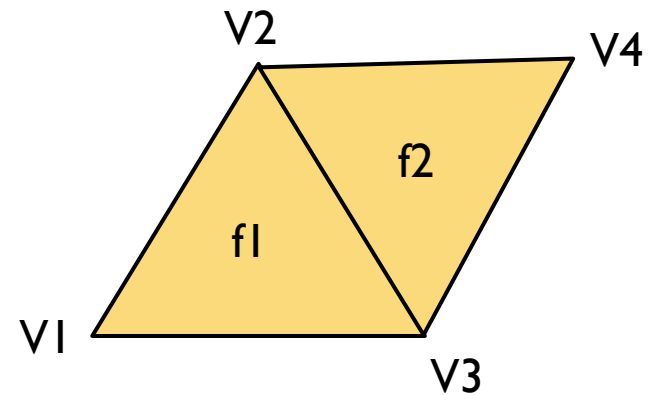
The Example User Interface



Wavefront File Format (.obj)

Example.obj

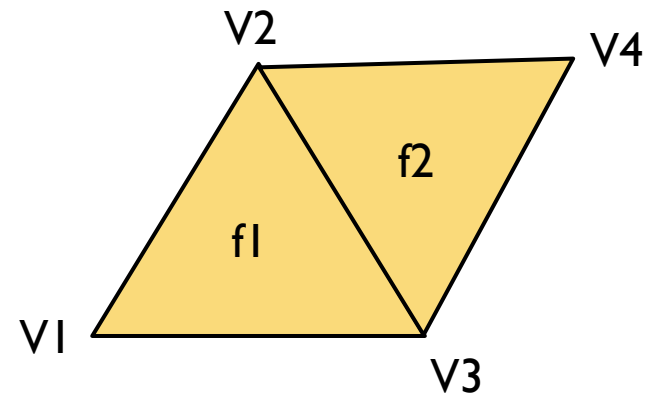
```
# List of vertices.  
v      x1 y1 z1  
v      x2 y2 z2  
v      x3 y3 z3  
v      x4 y4 z4  
  
# Face definitions.  
f      1 3 2  
f      2 3 4
```



Wavefront File Format (.obj)

Example.obj

```
# List of vertices.  
v      x1 y1 z1  
v      x2 y2 z2  
v      x3 y3 z3  
v      x4 y4 z4  
  
# List of texture coordinates.  
vt     u1 v1  
vt     u2 v2  
  
# List of normals.  
vn     nx1 ny1 nz1  
vn     nx2 ny2 nz2  
  
# Face definitions (vertex/texture-coordinate/normal).  
f      1/1/1 3/1/2 2/1/2  
f      2/2/2 3/1/2 4/2/2
```



Read Wavefront File Format (.obj)

MeshIO.cpp

```
// Postcondition: 'hds' is a valid polyhedral surface.
CGAL::Polyhedron_incremental_builder_3<HDS> builder(hds, true);

// Read data from the file.
infile.open(m_aFilename);
builder.begin_surface(nVertexNum, nTriangleNum);
while(!infile.eof())
{
    infile.getline(aBuf, 200, '\n');

    if(aBuf[0] == 'v' && aBuf[1] == ' ')
    {
        float fX, fY, fZ;
        sscanf(aBuf, "v %f %f %f", &fX, &fY, &fZ);
        builder.add_vertex(Point(fX, fY, fZ));
    }

    if(aBuf[0] == 'f' && aBuf[1] == ' ')
    {
        int nIndex1, nIndex2, nIndex3;
        sscanf(aBuf, "f %d %d %d", &nIndex1, &nIndex2, &nIndex3);
        builder.begin_facet();
        builder.add_vertex_to_facet(nIndex1 - 1);
        builder.add_vertex_to_facet(nIndex2 - 1);
        builder.add_vertex_to_facet(nIndex3 - 1);
        builder.end_facet();
    }
}
builder.end_surface();
infile.close();
```


Good Luck!
