

很早就想学 opengl 的，一直没时间，今天心血来潮初步了解了下 opengl。

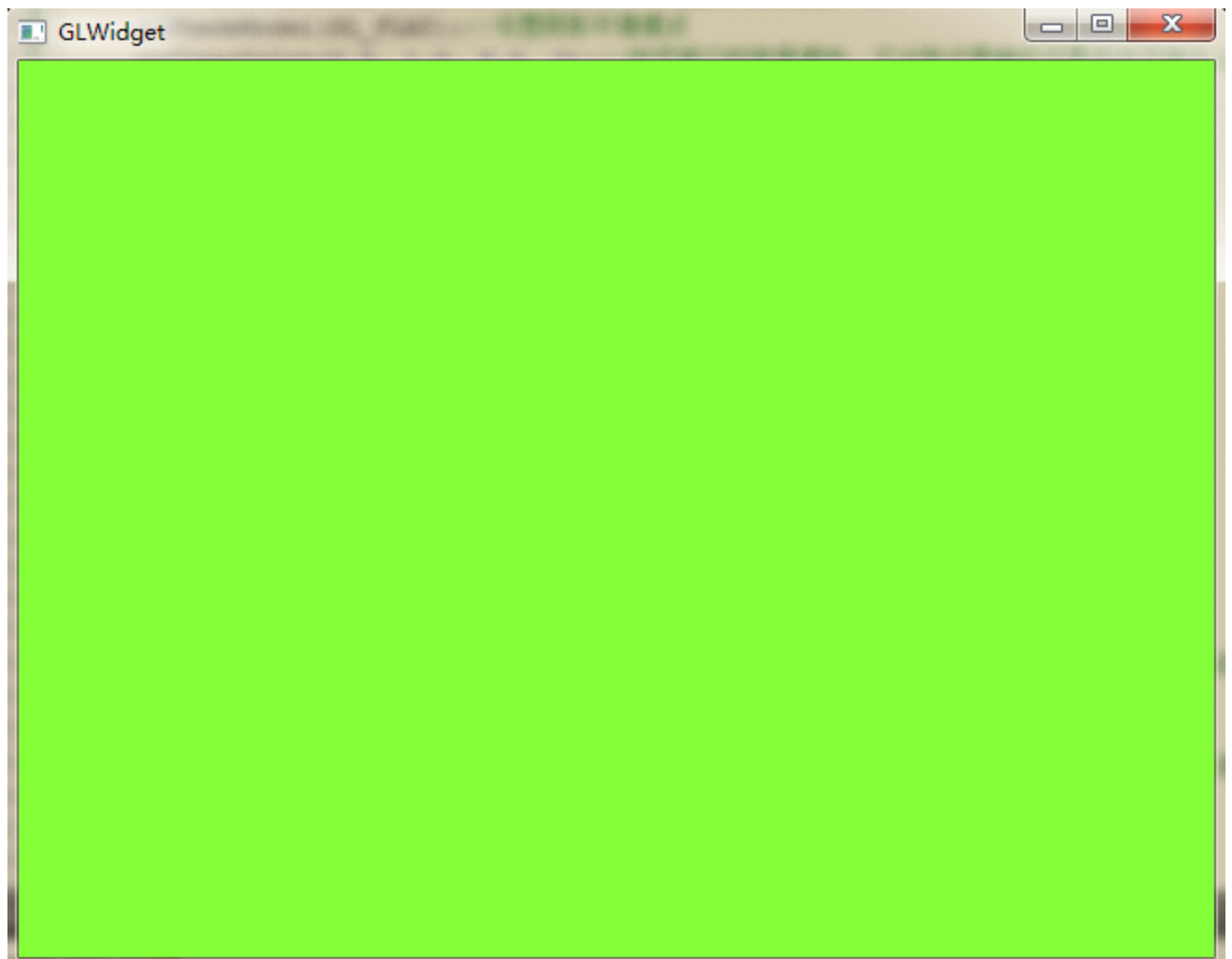
Opengl 是对2D 和3D 图形支持很好，有非常多的优化函数，因为 opengl 的主要目标是图形功能函数上，所以它对图形界面的支持并不完善。当然了，作为一个跨平台的开源库，这是能理解的，因为每个平台开发界面用的工具不同，如果它只支持其一，那么私心也太重了，还不如节省精力放在核心的代码上。

Qt 的 UI 函数很多，它的 Opengl 封装很好，将 opengl 的函数和 Qt 的界面结合了起来，这主要是通过 Qt 中 QGLWidget 这个类来实现。所以一般我们开发 opengl，且要用到 Qt 的界面，这些类都可以从 QGLWidget 继承过来。

本次试验是按照网络上 NeHe 写的最著名的 opengl 英文教程：<http://nehe.gamedev.net/> 和对应的中文教程：<http://www.owlei.com/DancingWind/> 的第一讲来学习的。当然也参考了网上对 Nehe 教程改编的 Qt 版：<http://wenku.baidu.com/view/835b29fd941ea76e58fa04a2.html>

本次试验的目的很简单，只是显示一个窗口，可以通过 F1键值来切换全屏显示和普通屏显示，并当按下 ESE 键时退出程序。窗口的颜色背景和透视效果(其实该试验都没用上)等用 opengl 来实现，主要是为后面的学习写了个框架，其实这里主要是重写了3个函数：initializeGL();paintGL();resizeGL();这3个函数都是 QGLWidget 内部的虚函数。

程序运行时效果如下：



程序代码如下：

glwidget.h:



```
#ifndef GLWIDGET_H
#define GLWIDGET_H
#include<QtOpenGL>#include<QWidget>

namespace Ui {
class GLWidget;
}

class GLWidget : public QGLWidget
{
    Q_OBJECT

public:
    explicit GLWidget(QGLWidget *parent = 0);
```

```

~GLWidget();

protected:
    void initializeGL();
    void paintGL();
    void resizeGL(int width, int height);
    void keyPressEvent(QKeyEvent *e);
    bool fullscreen;

private:
    Ui::GLWidget*ui;
};

#endif // GLWIDGET_H

```



glwidget.cpp:



```

#include "glwidget.h"#include"ui_glwidget.h"
#include<QtGui>#include<QtCore>#include<QtOpenGL>
GLWidget::GLWidget(QGLWidget*parent) :
    QGLWidget(parent),
    ui(new Ui::GLWidget)
{
    ui->setupUi(this);
    fullscreen= false;
}

//这是对虚函数，这里是重写该函数
void GLWidget::initializeGL()
{
    setGeometry(300, 150, 640, 480); //设置窗口初始位置和大小
    glShadeModel(GL_FLAT); //设置阴影平滑模式
    glClearColor(0.5, 1.0, 0.2, 0); //改变窗口的背景颜色，不过我这里貌似设置后并没有什么效果
    glClearDepth(1.0); //设置深度缓存
    glEnable(GL_DEPTH_TEST); //允许深度测试
    glDepthFunc(GL_LEQUAL); //设置深度测试类型
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST); //进行透视校正
}

void GLWidget::paintGL()

```

```

{
    //glClearColor() 函数在这里就是对 initializeGL() 函数中设置的颜色和缓存深度等起作用
    glClearColor(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity(); //重置当前的模型观察矩阵? 不是很理解!
}

//该程序是设置 opengl 场景透视图, 程序中至少被执行一次(程序启动时).
void GLWidget::resizeGL(int width, int height)
{
    if(0 == height)
        height= 1; //防止一条边为0
    glViewport(0, 0, (GLint)width, (GLint)height); //重置当前视口, 本身不是重置窗口的, 只不过是这里被 Qt 给封装好了
    glMatrixMode(GL_PROJECTION); //选择投影矩阵
    glLoadIdentity(); //重置选择好的投影矩阵
    // gluPerspective(45.0, (GLfloat)width/(GLfloat)height, 0.1, 100.0); //建立透
    视投影矩阵
    //glMatirxMode(GL_MODELVIEW); //以下2句和上面出现的解释一样
    glLoadIdentity();
}

void GLWidget::keyPressEvent(QKeyEvent *e)
{
    switch(e->key())
    {
        //F1键为全屏和普通屏显示切换键
        case Qt::Key_F1:
            fullscreen= !fullscreen;
            if(fullscreen)
                showFullScreen();
            else
            {
                setGeometry(300, 150, 640, 480);
                showNormal();
            }
            updateGL();
            break;
        //Ese 为退出程序键
        case Qt::Key_Escape:
            close();
    }
}

GLWidget::~GLWidget()

```

```
{  
    delete ui;  
}
```



程序里面有很多 opengl 的函数暂时还用不到，因为图形学领域不熟，有些函数功能暂时也不理解，以后慢慢会懂的。

编译时，gluPerspective，glMatirxMode 处如下错误：

C:\Users\Administrator\Desktop\tornadomeet\opengl\opengl_nehe_01\glwidget.cpp:46: error: C3861: “gluPerspective” :找不到标识符。

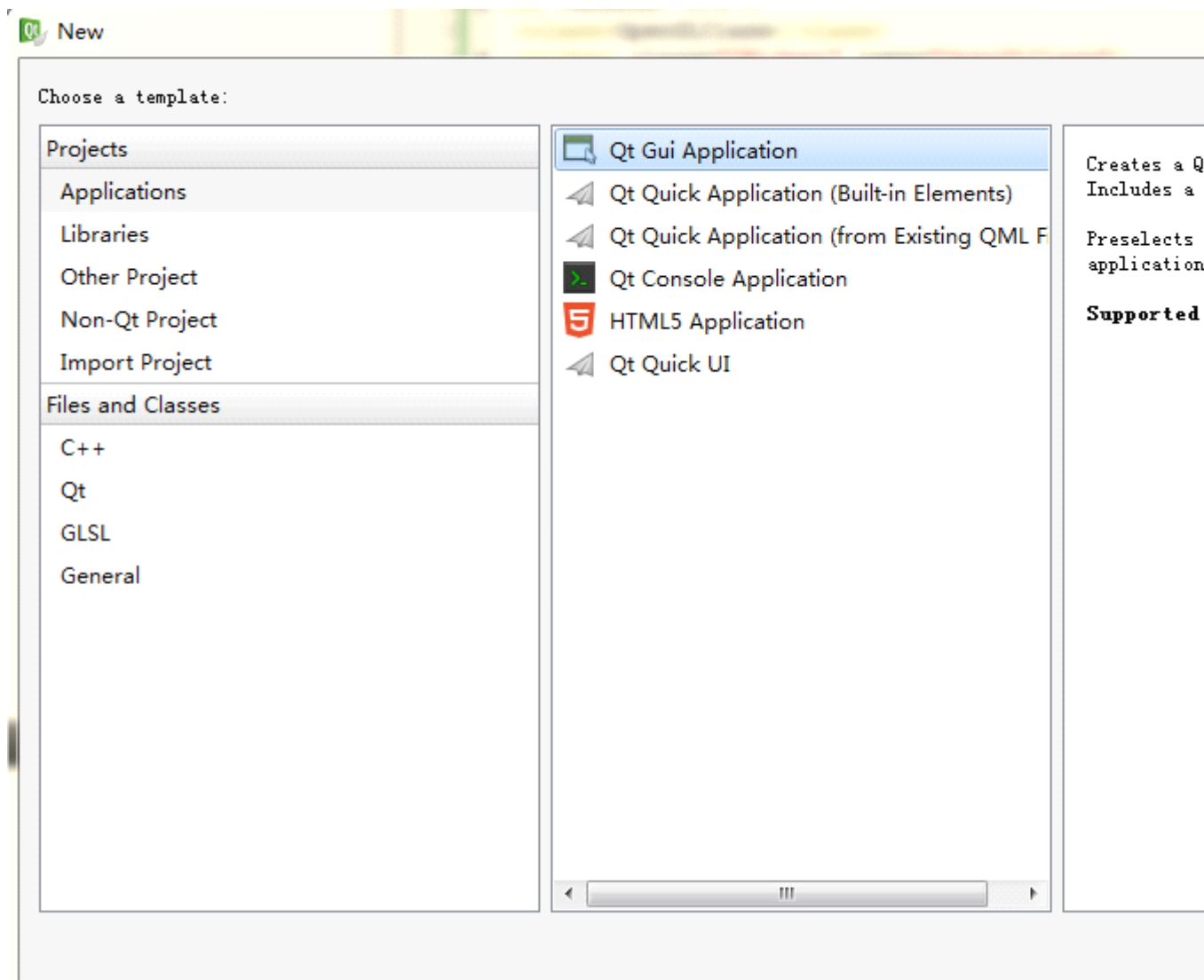
网上不少人都遇到过，貌似是 Qt4.8 以后的高级版本都不支持 glu 开头的函数了，是 Nokia 弄的 bug，还是有什么原因故意这么弄的？据说 Qt4.7.4 还支持。对于我这种初学者，反正还没用到那个函数，可以不管，实在要用到时切换到 4.7.4 也行，还有，5.0 据说 9 月份就要发行了，很期待。

附录一

因为是初学，可能有些同学不懂改怎样建立于 opengl 有关的工程，是用 QtGUI 工程，还是 Qt 控制台工程，或者是 Qt 空工程？网上貌似还没有找到比较好的介绍。下面我把建立该工程的过程记下来(也许步骤不对，不过我这次就是这么弄的)。

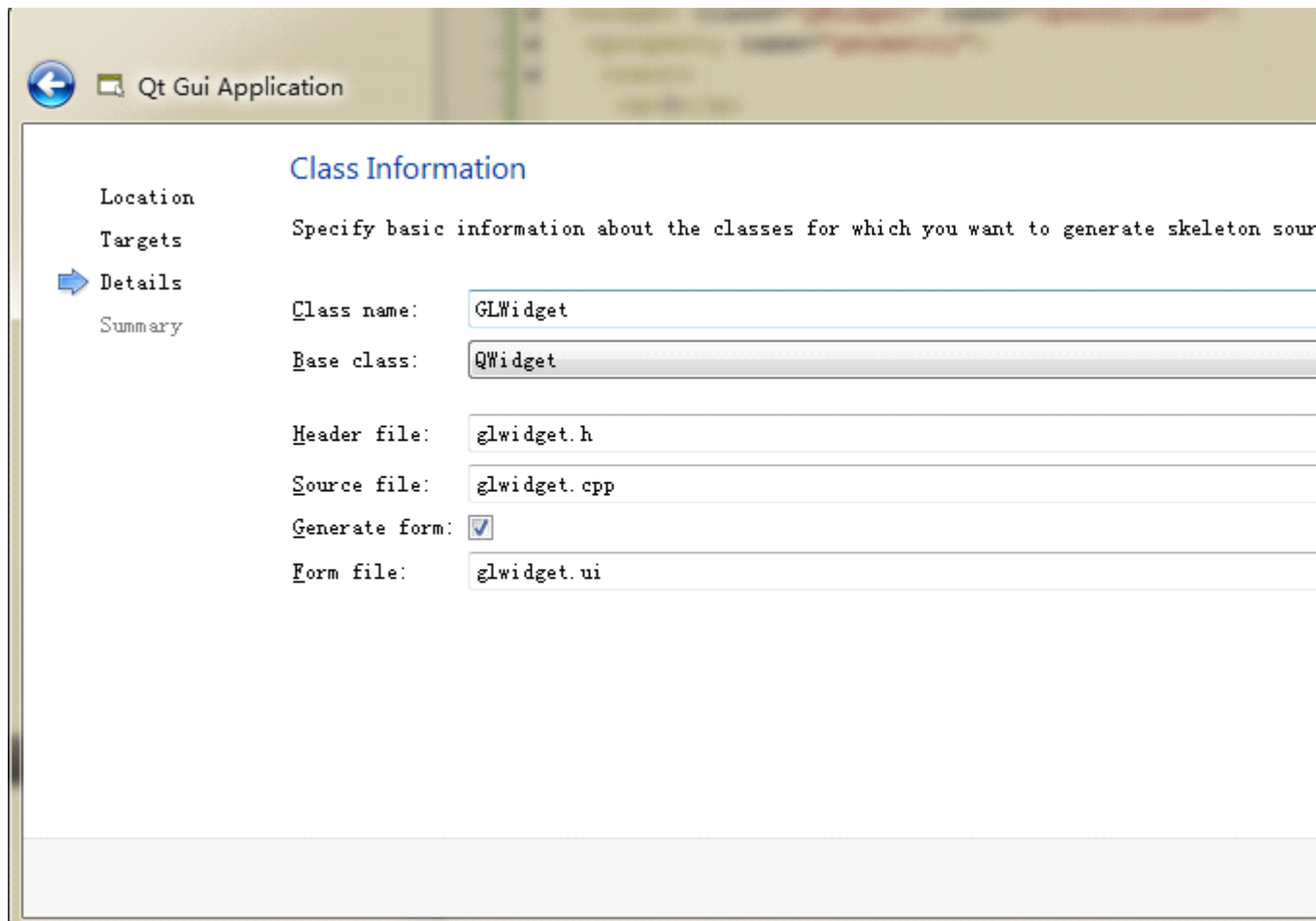
开发环境：windows+Qt4.8.2+QtCreator2.5.1

1. 打开 QtCreator 软件，进入 File-->New File or Project，左边 Projects 选择 Application，中间选择 Qt Gui Application，如下图所示：



2. 单击 choose，进入下一个界面，自己填入工程名和目录(不能包含空格和中文路径)。

3. 单击 next，进入下一界面，继续单击 next，进入 Class Information 界面。类的名字自己设置，注意 Base class(基类)一栏选择 QWidget，不要选择 QMainWindow 或者 QDialog，貌似我那2种试过，在本程序都不行。如下图所示：



4. 单击 next 进入下一界面，单击 Finish 完成操作。

5. 打开 glwidget.h 文件，因为生成的文件该类是从 QWidget 类继承过来的，而我们需要从 QGLWidget 中继承过来，所以把代码更改后如下：

```
#ifndef GLWIDGET_H
#define GLWIDGET_H

#include<QtOpenGL/QtOpenGL>
#include<QWidget>

namespace Ui {
class GLWidget;
}
```

```

class GLWidget : public QGLWidget
{
    Q_OBJECT

public:
    explicit GLWidget(QGLWidget *parent = 0);

    ~GLWidget();

private:
    Ui::GLWidget*ui;
};

#endif // GLWIDGET_H

```



6. 同理，glwidget.cpp 文件也更改后如下：

```

#include "glwidget.h"
#include "ui_glwidget.h"

GLWidget::GLWidget(QGLWidget*parent) :

    QGLWidget(parent),

    ui(new Ui::GLWidget)
{

```



```
    ui->setupUi(this);  
  
}  
  
GLWidget::~GLWidget()  
{  
  
    delete ui;  
  
}
```



附录二

实验工程 code 下载。