

```
/* *****  
** $Id: qt/gltxobj.cpp 3.1.2 edited Nov 8 2002 $  
**  
** Copyright (C) 1992-2002 Trolltech AS. All rights reserved.  
**  
** This file is part of an example program for Qt. This example  
** program may be used, distributed and modified without limitation.  
**  
** This is a simple QGLWidget demonstrating the use of QImage for textures.  
**  
*****/  
  
#include <iostream>  
#include <iomanip>  
#include <fstream>  
#include <string>  
  
#include <GL/gl.h>  
#include <GL/glu.h>  
#include <GL/glext.h>  
  
#include <QImage>  
#include <QTimer>  
#include <QEvent>  
#include <QString>  
#include <QFileDialog>  
#include <QMouseEvent>  
  
#include "gltxobj.h"  
  
const int redrawWait = 50;  
//const int redrawWait = 0;  
  
GLTexobj::GLTexobj( QWidget* parent ) :  
    QGLWidget( parent ),  
    img(64,64,QImage::Format_RGB32)  
{  
    // create a GLTexobj widget  
    timer = new QTimer( this );  
    connect( timer, SIGNAL(timeout()), SLOT(update()) );  
    timer->setInterval(redrawWait);  
    timer->setSingleShot(FALSE);  
    timer->start();  
}  
  
GLTexobj::~GLTexobj()  
{  
    // release allocated resources  
    makeCurrent();  
}  
  
void GLTexobj::initializeGL()  
{  
    // set up the OpenGL rendering state, and define display list  
    std::cerr << "GL: " << glGetString(GL_VERSION) << std::endl;  
  
    glGenTextures(1,&texName);  
    imageBind();  
  
    // specify clear color; (0.3,0.4,0.6,1.0) is a nice "slate blue"
```

```
glClearColor( 0.0, 0.0, 0.0, 0.0 );

glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
glEnable(GL_BLEND);
}

void GLTexobj::resizeGL( int w, int h )
{
    // set up the OpenGL view port, matrix mode, etc.
    glViewport(0,0,w,h);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0,w,0.0,h);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void GLTexobj::paintGL()
{
    // the actual openGL commands for drawing the texobj are performed here
    glDrawBuffer(GL_BACK);
    glClear(GL_COLOR_BUFFER_BIT);
    glColor4f(1.0,1.0,1.0,1.0);

    glEnable(GL_TEXTURE_2D);
    glMatrixMode(GL_TEXTURE);
    glLoadIdentity();
    // can translate texture here

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    // render
    glBegin(GL_QUADS);
    {
        // with convertToGLFormat
/*
        glTexCoord2f(0,0);
        glVertex2i(0,0);

        glTexCoord2f(1,0);
        glVertex2i(width(),0);

        glTexCoord2f(1,1);
        glVertex2i(width(),height());

        glTexCoord2f(0,1);
        glVertex2i(0,height());
*/
        // w/out convertToGLFormat
        glTexCoord2f(0,1);
        glVertex2i(0,0);

        glTexCoord2f(1,1);
        glVertex2i(width(),0);

        glTexCoord2f(1,0);
        glVertex2i(width(),height());
    }
}
```

```
    glTexCoord2f(0,0);
    glVertex2i(0,height());
}
glEnd();

// disable TU0
glDisable(GL_TEXTURE_2D);
}

void GLTexobj::mouseMoveEvent(QMouseEvent* e)
{
//std::cerr << "(" << e->x() << "," << e->y() << ")" << std::endl;

    e->accept();
    updateGL();
}

void GLTexobj::update()
{
//std::cerr << "updating..." << std::endl;

    updateGL();
}

void GLTexobj::imageOpen()
{
    QString qfilename = QFileDialog::getOpenFileName(this,
                                                    tr("Open Image"),
                                                    "./",
                                                    tr("Image Files (*.png *.jpg *
.bmp *.ppm)"));

    if(!qfilename.isEmpty()) img.load(qfilename);
//img = QGLWidget::convertToGLFormat(img);

    imageBind();

    updateGL();
}

void GLTexobj::imageBind()
{
    glBindTexture(GL_TEXTURE_2D,texName);
    glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_WRAP_S,GL_CLAMP);
    glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_WRAP_T,GL_CLAMP);
    glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MAG_FILTER,GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D,GL_TEXTURE_MIN_FILTER,GL_NEAREST);
    glTexImage2D(GL_TEXTURE_2D,                // target
                0,                            // level
                GL_RGBA,                      // internalFormat
                img.width(),img.height(),     // width,height
                0,                            // border
                //GL_RGBA,                    // format, w/    convertToGLFormat
                GL_BGRA,                      // format, w/out convertToGLFormat
                GL_UNSIGNED_BYTE,            // type
                img.bits());                // texels
}
```

```
glTexEnvi(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE);  
}
```

image to texture

```
#include <iostream>

#include <QApplication>
#include <QMenuBar>
#include <QMenu>
#include <QVBoxLayout>

#include "glwinobj.h"
#include "gltexobj.h"

GLObjectWindow::GLObjectWindow( QWidget* parent, Qt::WindowFlags f ) :
    QWidget( parent, f )
{
    // Create an OpenGL widget: (doubleBuffer | rgba | depth) set globally
    GLTexobj* c = new GLTexobj(this);
    std::cout << "doubleBuffer: " << c->format().doubleBuffer() << " "
              << "rgba: " << c->format().rgba() << " "
              << "depth: " << c->format().depth() << " "
              << std::endl;
    // c->setMouseTracking(true);

    // create the file menu
    QMenu *file = new QMenu("File",this);
    file->addAction("Open...",c,SLOT(imageOpen()),Qt::CTRL+Qt::Key_O);
    // doesn't seem to be needed in Qt 4.6
    file->addAction("Quit",QApplication::instance(),SLOT(quit()),
                  Qt::CTRL+Qt::Key_Q);

    // create a menu bar
    QMenuBar *m = new QMenuBar(this);
    m->addSeparator();
    m->addMenu(file);

    // top level layout (with 0,0 border)
    QVBoxLayout* vlayout = new QVBoxLayout(this);

    // no border, no margin
    vlayout->setSpacing(0);
    vlayout->setMargin(0);

    // add menu bar and GL window
    vlayout->setMenuBar(m);
    vlayout->addWidget(c);
}
```

```
/* *****  
** $Id: qt/main.cpp 3.1.2 edited Nov 8 2002 $  
**  
** Copyright (C) 1992-2000 Trolltech AS. All rights reserved.  
**  
** This file is part of an example program for Qt. This example  
** program may be used, distributed and modified without limitation.  
**  
*****/  
//  
// Qt OpenGL example: Texture  
//  
// File: main.cpp  
//  
// The main() function  
//  
#include <QApplication>  
#include <QGLFormat>  
  
#include "glwinobj.h"  
  
int main(int argc, char **argv)  
{  
    QApplication::setColorSpec(QApplication::CustomColor);  
    QApplication app(argc,argv);  
  
    if(!QGLFormat::hasOpenGL()) {  
        qWarning( "This system has no OpenGL support. Exiting." );  
        return -1;  
    }  
  
    // Create OpenGL format  
    QGLFormat f;  
    f.setDoubleBuffer(TRUE); f.setRgba(TRUE); f.setDepth(TRUE);  
    QGLFormat::setDefaultFormat(f);  
  
    GLObjectWindow* w = new GLObjectWindow;  
  
    // set size...  
    w->resize( 400, 350 );  
    //w->resize( 1280, 1024 );  
    w->show();  
    // ... or go full screen  
    //w->showFullScreen();  
  
    int result = app.exec();  
    delete w;  
    return result;  
}
```