

Costruzione di Interfacce Lezione 20 Qt + OpenGL

cignoni@isti.cnr.it
<http://vcg.isti.cnr.it/~cignoni>

❖ Primo tentativo

- ❖ si prende il tutorial hello e si sostituisce il pushbutton con un QGLWidget

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

2

```
/*.....  
**  
** Qt GL 1  
**.....*/  
  
#include <qapplication.h>  
#include <qgl.h>  
  
int main( int argc, char **argv )  
{  
    QApplication a( argc, argv );  
  
    QGLWidget hello(0);  
    hello.resize( 100, 30 );  
  
    a.setMainWidget( &hello );  
    hello.show();  
    return a.exec();  
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

3

❖ Compila ma non fa nulla di interessante.

- ❖ QGLWidget e' pensata per essere la classe base da cui si deriva una classe in cui specializziamo
 - ❖ paintGL()
 - ❖ resizeGL()
 - ❖ initializeGL()
- ❖ sostituendole con le nostre implementazioni

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

4

subclassiamo

- ❖ prima specializzazione poco interessante
 - ❖ resize setta solo il viewport
 - ❖ initialize setta solo il colore del clear
 - ❖ paint si limita a pulire lo schermo.
- ❖ Non c'è bisogno di fare swapbuffer.
 - ❖ viene fatto automaticamente da qt una volta finito il paint
 - ❖ QGLWidget e' abbastanza ad alto livello
 - ❖ QGLContext provvede un controllo maggiore di quel che succede.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

5

subclassiamo

```
/*.....  
**  
** Qt GL 2  
**.....*/  
  
#include <qapplication.h>  
#include <qgl.h>  
  
class myGLWidget : public QGLWidget  
{ public:  
    myGLWidget( QWidget *parent=0, const char *name="0" ): QGLWidget(parent, name) {}  
    void initializeGL() {  
        // Set up the rendering context, define display lists etc.:  
        glClearColor( 0.0, 1.0, 1.0, 0.0 );  
    }  
    void resizeGL( int w, int h ) {  
        // setup viewport, projection etc.:  
        glViewport( 0, 0, (GLint)w, (GLint)h );  
    }  
    void paintGL() {  
        glClear( GL_COLOR_BUFFER_BIT );  
    }  
};
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

6

subclassiamo

```
int main( int argc, char **argv )
{
    QApplication a( argc, argv );

    myGLWidget hello(0);
    hello.resize( 100, 30 );

    a.setMainWidget( &hello );
    hello.show();
    return a.exec();
}
```

Ripuliamo

- ❖ La definizione della classe e la sua implementazione spostiamola in due file a parte
- ❖ CIGLWidget.h e CIGLWidget.cpp
- ❖ prepariamo anche la classe a slot e a tutto il resto aggiungendo la macro Q_OBJECT
- ❖ Il file verra' processato dal MOC

CIGLWidget.h

```
#ifndef CIGLWIDGET_H
#define CIGLWIDGET_H

#include <qgl.h>

class CIGLWidget : public QGLWidget
{
    Q_OBJECT
public:
    CIGLWidget( QWidget *parent=0, const char *name=0 );
    void initializeGL();
    void resizeGL( int w, int h );
    void paintGL();
};

#endif
```

CIGLWidget.cpp

```
#include "CIGLWidget.h"

CIGLWidget::CIGLWidget( QWidget *parent, const char *name ) :
    QGLWidget(parent, name)
{
}

void CIGLWidget::initializeGL()
{
    // Set up the rendering context, define display lists etc.:
    glClearColor( 0.0, 1.0, 1.0, 0.0 );
}

void CIGLWidget::resizeGL( int w, int h )
{
    // setup viewport, projection etc.:
    glViewport( 0, 0, (GLint)w, (GLint)h );
}

void CIGLWidget::paintGL()
{
    glClear(GL_COLOR_BUFFER_BIT);
}
```

main.cpp

```
#include <qapplication.h>
#include "CIGLWidget.h"

int main( int argc, char **argv )
{
    QApplication a( argc, argv );

    CIGLWidget hello(0);
    hello.resize( 100, 30 );

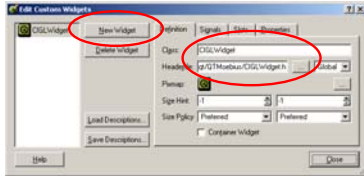
    a.setMainWidget( &hello );
    hello.show();
    return a.exec();
}
```

Designer

- ❖ Aprire Designer
- ❖ nuovo progetto
- ❖ nuovo mainwindow
 - ❖ nome MainForm
 - ❖ caption QTMoeibus
- ❖ nuovo main.cpp
- ❖ salvare

Aggiungere Custom Widget

- ❖ Copiare CIGLWidget.h e.cpp nella dir del progetto
- ❖ menu tool->custom->Edit Custom Widget



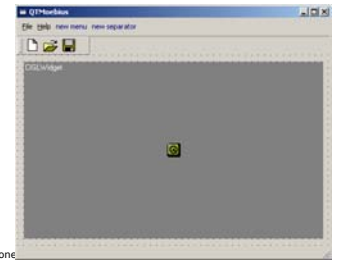
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

13

Usare il custom widget

- ❖ Aggiungere il CIGLWidget al form
- ❖ deselezionare e fare layout H



29 Oct 2003

Costruzione

CIGLWidget

- ❖ I file che implementano la CIGLWidget devono essere aggiunti al progetto di designer.
- ❖ Devono essere nel .pro perche' il .h va processato dal moc
- ❖ project->add file
- ❖ salvare

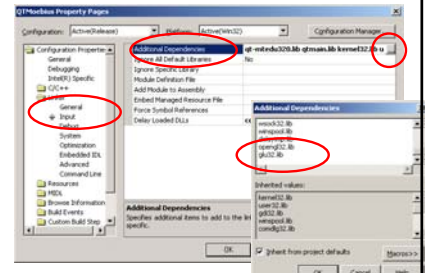
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

15

.net

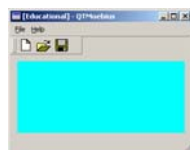
- ❖ Importare il progetto .pro nel .net
- ❖ controllare che ci siano tutti file
- ❖ aggiungere le lib opengl



29 Oct 2003

Compilare

- ❖ Dovreste vedere qualcosa di simile a questo...



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

17

Aggiungiamo moebius

- ❖ Aggiungiamo la solita roba di moebius
 - ❖ Strutturata in una classe CIMoebius
- ❖ Cambiamo la paint e la initialize
- ❖ Aggiungiamo due membri alla classe CIWidget
 - ❖ **SideNum** il numero di lati dell'anello
 - ❖ **Twist**, quanto si avvolge l'anello
- ❖ Aggiungiamo anche due slot per modificarli
- ❖ aggiungiamo nel.cpp della CIWidget una globale per tenere l'anello di moebius

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

18

La classe CIMoebius

- ❖ Incapsula un anello di moebius
- ❖ ha un metodo per generare un anello con un
 - ❖ dato numero di lati,
 - ❖ con un certo twist
 - ❖ di un certo raggio
- ❖ ha un membro interno pubblico di tipo mesh che ha un metodo per disegnarsi.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

19

Nuova CIGLWidget

```
/* *****  
** Definition of CIGLWidget class  
***** */  
#ifndef CIGLWIDGET_H  
#define CIGLWIDGET_H  
#include <qgl.h>  
  
class CIGLWidget : public QGLWidget  
{  
    Q_OBJECT  
public:  
    CIGLWidget( QWidget *parent=0, const char *name=0 );  
    void initializeGL();  
    void resizeGL( int w, int h );  
    void paintGL();  
public slots:  
    void setSideNum( int n ) {SideNum=n; initializeGL(); update();}  
    void setTwist( int n ) {Twist=n; initializeGL(); update();}  
  
private:  
    int SideNum;  
    int Twist;  
};  
  
#endif  
29 Oct 2003
```

Costruzione di Interfacce - Paolo Cignoni

20

CIWidget.cpp initialize

```
CIMoebius pp;  
void CIGLWidget::initializeGL()  
{  
    glClearColor( 0.0, 1.0, 1.0, 0.0 );  
    CIMoebius::myShape s;  
    s.BuildNGon( SideNum, 1 );  
    pp.GenerateMoebiusRing( s, 36, 3, Twist*360/SideNum );  
    float spec[4]={.4, .4, .4, 1};  
    float shin = 1;  
    glMaterialfv( GL_FRONT_AND_BACK, GL_SPECULAR, spec );  
    glMaterialf( GL_FRONT_AND_BACK, GL_SHININESS, shin );  
  
    glColorMaterial( GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE );  
    glEnable( GL_COLOR_MATERIAL );  
  
    glColorMaterial( GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE );  
    glEnable( GL_COLOR_MATERIAL );  
  
    glEnable( GL_LIGHTING );  
    glEnable( GL_NORMALIZE );  
    glEnable( GL_DEPTH_TEST );  
    glShadeModel( GL_SMOOTH );  
}
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

21

Costruttore paint e resize

```
CIGLWidget::CIGLWidget( QWidget *parent, const char *name ) :  
    QGLWidget( parent, name )  
{  
    SideNum=3;  
    Twist=1;  
}  
void CIGLWidget::resizeGL( int w, int h )  
{  
    // setup viewport, projection etc.:  
    glMatrixMode( GL_PROJECTION );  
    glLoadIdentity();  
    float ratio=(float)w/(float)h;  
    gluPerspective( 45, ratio, 3, 20 );  
    glViewport( 0, 0, (GLsizei) w, (GLsizei) h );  
    glMatrixMode( GL_MODELVIEW );  
}  
void CIGLWidget::paintGL()  
{  
    glMatrixMode( GL_MODELVIEW );  
    glClear( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );  
    glLoadIdentity();  
    gluLookAt( 2, 5, 12, 0, 0, 0, 1, 0 );  
    glColor3f( 1, .4, .4 );  
    pp.m.Draw();  
}
```

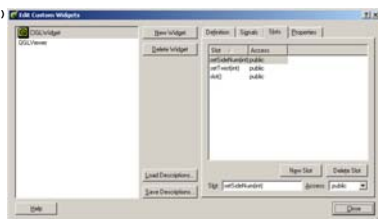
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

22

designer

- ❖ Comuniciamo l'esistenza dei due slot a designer
- ❖ tools->custom->edit custom widget
- ❖ scegliere il tab slot e aggiungere nome e tipi dei due slot
 - ❖ setSideNum(int)
 - ❖ setTwist(int)



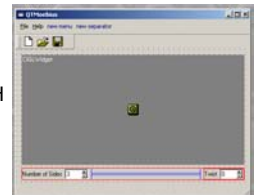
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

23

Modifichiamo l'interfaccia

- ❖ aggiungiamo due spinbox e colleghiamoli agli slot del GLWidget
- ❖ metteteci nel mezzo uno spacer
- ❖ selezionati tutti layout H
- ❖ non selezionate nulla e layout V



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

24

