

Costruzione di Interfacce Lezione 18 Qt Tutorial 2

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

GUI Applications

- ❖ Le applicazioni *classiche* hanno un'interfaccia abbastanza standard
 - ❖ dialog based
 - ❖ l'applicazione e' principalmente un dialog che permette di settare un po' di opzioni e poi terminare
 - ❖ clienti ftp, programmi stateless, ecc.
 - ❖ main window
 - ❖ l'applicazione e' strutturata con un'area centrale dove vengono visualizzati/editati i dati e con una barra di menu, una o piu' toolbar e una status bar
 - ❖ documento centriche

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

2

GUI Applications

- ❖ le applicazioni *main window* sono basate su almeno due template
 - ❖ Single Document Interface (SDI)
 - ❖ l'app puo' lavorare su un solo doc alla volta
 - ❖ wordpad/mediaplayer
 - ❖ Multiple Document Interface (MDI)
 - ❖ si puo' lavorare su piu' doc contemporaneamente, tenuti in finestre separate in uno workspace strutturato come un piccolo desktop
 - ❖ PaintShopPro/ Acrobat Reader ecc.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

3

QT

- ❖ supporta entrambi i generi di app.
- ❖ ogni widget puo' essere fatto il widget principale dell'app
- ❖ perfetto per dialog based app.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

4

The main window

- ❖ QMainWindow
 - ❖ e' un widget che fa da framework per le app basate sul concetto di documento
 - ❖ la parte in alto del widget e' occupata da una barra di menu
 - ❖ sotto ci sono le toolbar (docked o floating)
 - ❖ in basso alla finestra c'e' una status bar
 - ❖ la parte centrale del widget e' destinata al documento

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

5

Menu

- ❖ QPopupMenu
 - ❖ possono essere
 - ❖ standalone: menu contestuali
 - ❖ attaccati ad una barra
 - ❖ attaccati ad un item di un altro menu
 - ❖ ogni item si puo' connettere ad uno slot

```
QPopupMenu *fileMenu = new QPopupMenu( this );
fileMenu->insertItem("&New", this, SLOT(newFile()), CTRL+Key_N );
fileMenu->insertItem("&Open...", this, SLOT(open()), CTRL+Key_O );
fileMenu->insertSeparator();
fileMenu->insertItem("E&xit", qApp, SLOT(quit()), CTRL+Key_Q );
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

6

Menu Bar

- ❖ QMenuBar
- ❖ implementa una generica barra dei menu a cui si attaccano i popupmenu
 - ❖ appare dove vuole il SO
 - ❖ nei mac in cima allo schermo
 - ❖ win e linux in cima alla mainwindow

```
QMenuBar *bar = new QMenuBar( this );
bar->insertItem( "File", fileMenu );
bar->insertItem( "Edit", editMenu );
bar->insertItem( "Help", helpMenu );
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

7

Toolbar

- ❖ QPushButton
 - ❖ bottone con un'icona e opz del testo
- ❖ QToolBar
 - ❖ un'app può avere un qualunque numero di toolbar
 - ❖ possono contenere sia bottoni che altri widget (combobox, spinbox)

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

8

Actions

- ❖ Un'interfaccia offre differenti modi per fare la stessa cosa, e.g. save tramite:
 - ❖ menu
 - ❖ bottone
 - ❖ ctrl+s
- ❖ QAction incapsula questo concetto permettendo di definire un'azione in un posto solo

```
QAction *saveAct = new QAction( "Save", saveIcon, "Save",
CTRL+Key_S, this );
connect( saveAct, SIGNAL(activated()), this, SLOT(save()) );
saveAct->addTo( fileMenu );
saveAct->addTo( toolbar );
```

- ❖ lo stato dei vari item viene mantenuto consistente
- ❖ possibilità di gestire gruppi (stile radiobutton)

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

9

Help

- ❖ tre tipi di modi di fornire informazioni all'utente
- ❖ help standard
 - ❖ html based, l'engine di browsing e' l'assistant di qt stesso che oltre a far vedere html permette di farci ricerche.
- ❖ tooltips
 - ❖ piccole win solo testo, di solito associate a bottoni di toolbar che non hanno testo che le descrive. Appaiono automaticamente
 - ❖ QToolTip::add(quitButton, "Leave the application");

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

10

Help 2

- ❖ balloon
 - ❖ quello che appare con "guida rapida" che serve a descrivere un po' piu' dettagliatamente gli elementi dell'interfaccia:

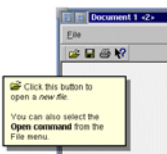
```
const char * fileOpenText =
"q<img source=\"fileopen\"> "
"Click this button to open a <em>new file</em>. <br>"
"You can also select the <b>Open</b> command "
"from the <b>File</b> menu.</p>";
QMimeSourceFactory::defaultFactory()->setPixmap( "fileopen",
fileOpenAction->iconSet().pixmap() );
fileOpenAction->setWhatsThis( fileOpenText );
```

- ❖ QMimeSourceFactory fornisce un'interfaccia astratta per memorizzare dati come immagini, suoni che hanno un mimetype.
- ❖ le cose memorizzate dentro la default factory sono accessibili da tutta l'applicazione

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

11



L'area centrale

- ❖ L'area centrale di una QMainWindow può contenere qualsiasi widget.
- ❖ E.g. in un text editor si può usare un QTextEdit come widget centrale:

```
QTextEdit *editor = new QTextEdit( mainWindow );
mainWindow->setCentralWidget( editor );
```
- ❖ Multiple document interface (MDI) e' realizzata usando come central widget la classe Qworkspace

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

12

MDI

- ❖ figli di QWorkspace possono essere widgets of qualunque tipo.
- ❖ I figli di QWorkspace sono visualizzati con frame simile a quello dei widget top level
- ❖ *Hide, show, minimize* funzionano per i child di QWorkspace come per i widget top level
- ❖ idem per posizionamento di finestre ecc.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

13

Drawing non in 3d

- ❖ QImage supporta input, output e manipolazione di immagini
- ❖ supporta formati BMP, GIF★, JPEG, MNG, PNG, PNM, XBM, and XPM.
- ❖ la maggior parte dei widget possono mostrare immagini (bottoni, label, menu items ecc)

```
QPushButton *button =
    new QPushButton( "&Find Address", parent );
button->setIconSet( IconSet( QImage( "find.bmp" ) ) );
```

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

14

QPainter

- ❖ Classe che astrae l'accesso agli strumenti di disegno su di un particolare contesto
- ❖ funzionalita' classiche di base
 - ❖ disegno punti, linee, trasformazioni, testo
 - ❖ Origine top left
 - ❖ bottom right e' w-1 e h-1
- ❖ Usato dai widget per ridisegnarsi tramite override del metodo
 - ❖ void myClass::paintEvent(QPaintEvent *)

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

15

paint devices

- ❖ **QPainter** funziona su generici "paint device."
 - ❖ i **QWidget** sono anche essi stessi paint device
 - ❖ **QPixmap** essenzialmente un "off-screen widget." usato per fare "double buffering."
 - ❖ **QPicture** e' una rappresentazione vector dell'immagine che puo' essere scaled, rotated, and sheared bene. Supportata SVG XML format
 - ❖ **QPrinter** su Windows, i paint commands sono mandati al Windows print engine; su Unix si fa via PostScript.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

16

Canvas

- ❖ classe che fornisce un interfaccia di piu' alto livello della QPainter per disegnare
- ❖ un QCanvas contiene QCanvasItem
 - ❖ sprite, poligoni, ellissi linee ecc.
 - ❖ I QCanvasItem possono essere interattivi (eg mossi dall'utente)

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

17

Layout

- ❖ QT mette a disposizione gli strumenti per evitare il piazzamento fisso dei widget
- ❖ I widget vengono disposti e scalati all'interno di altri widget adattivamente.
- ❖ Piu' robusto a
 - ❖ internazionalizzazione
 - ❖ scaling font
- ❖ Layout manager standard
 - ❖ **QHBoxLayout**, **QVBoxLayout**, and **QGridLayout**.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

18

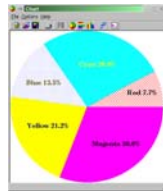
Layout 2

- ❖ Si può personalizzare il meccanismo:
 1. settando minimum size, maximum size, o fixed size per qualche child widgets.
 2. Aggiungere stretch items o spacer items (oggetti che riempiono gli spazi vuoti in un layout)
 3. cambiare le size policies; Child widgets possono essere obbligati a expand, contract, mantenere la propria size, etc.
 4. cambiare le size hints. preferred size and preferred minimum size based on the contents.
 5. settare stretch factors. permettono di stabilire proporzioni tra widgets, e.g. del nuovo spazio ottenuto da un resize, 2/3 a widget A e 1/3 a widget B



il tutorial 2

- ❖ Chart, un'applicazione vera e propria



Riferimenti

- ❖ nell'assistant
 - ❖ il whitepaper su qt
 - ❖ tutorial #2
 - ❖ Actions
- ❖ il codice prendetelo direttamente dalla dir examples/chart di QT