

## Costruzione di Interfacce Lezione 24 GI Selection; MFC: menu e toolbar

[cignoni@iei.pi.cnr.it](mailto:cignoni@iei.pi.cnr.it)  
<http://vcg.iei.pi.cnr.it/~cignoni>

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

1

## Moebius8

- ❖ Partiamo nuovamente da zero
  - ❖ App Mfc, questa volta **SDI**
  - ❖ Cambiamo I nomi alle classi...
  - ❖ Deriviamo la classe vista da **CFormView**...
  - ❖ Appena fatto il progetto aggiungiamo al progetto una classe derivata genericamente da Cview e chiamiamola CGLView (menu *project->addclass*)
  - ❖ Sistemiamo include e tipi in modo che tutti conoscano i tipi che servono...

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

2

## CFormView

- ❖ Al solito vogliamo fare un'app con un formview (finestra tipo dialogo) da una parte e opengl dall'altra
- ❖ Questa volta tra le classi non c'è la CChildFrame derivata da CMDIChildWindow
- ❖ Useremo la CMainFrame derivata da CFrameWnd

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

3

## CFormView

- ❖ Una form view è una view che contiene control (bottoni, combobox, listbox, tree ecc)
- ❖ I controlli sono piazzati secondo una risorsa dialog-template
- ❖ Ad ogni controllo nel template si può associare un oggetto, membro della classe che viene creato/piazzato/distrutto automaticamente, ma i cui eventi devono essere gestiti dall'utente

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

4

## Doppia View

- ❖ Obiettivo un'applicazione con da una parte opengl e dall'altra un dialogo.
- ❖ Si usa la classe CSplitterWnd
- ❖ Si modifica la classe CMainFrame (il frame della finestra principale in un app sdi)

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

5

## CSplitterWnd

- ❖ Facciamo override totale della funzione CMainFrame::OnCreateClient

```
BOOL CMainFrame::OnCreateClient(LPCREATESTRUCT lpcs, CCreateContext* pContext)
{
    m_SplitWnd.CreateStatic(this,1,2);
    if(!m_SplitWnd.CreateView(0,0,RUNTIME_CLASS(CMBView),
        CSize(300,100),pContext)){
        return FALSE;
    };
    if(!m_SplitWnd.CreateView(0,1,RUNTIME_CLASS(CMBGLView),
        CSize(100,100),pContext)){
        return FALSE;
    }
    m_fview = (CMBView *) m_SplitWnd.GetPane(0,0);
    m_glview = (CMBGLView *) m_SplitWnd.GetPane(0,1);
    m_hSplitterCreated=true;
    return true;
}
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

6

## CSplitterWnd

- ❖ In questo modo alla creazione di una nuova finestra per il doc invece di fare una sola view e di attaccarcela crea un oggetto di tipo CSplitterWnd e ci attacca due view di due tipi diversi.
- ❖ Adesso adattiamo la seconda finestra ad opengl come abbiamo visto nella scorsa lezione:

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

7

## Aggiungiamo opengl

- ❖ Per comodità trovate tutti i pezzi di codice che servono in un txt chiamato MFCOpenGL.txt
  - ❖ Include opengl
  - ❖ Membri nella classe view
    - ❖ OnCreate
    - ❖ OnSize
    - ❖ OnDestroy
    - ❖ OnEraseBkgnd
    - ❖ SetGL
    - ❖ InitGL

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

8

## Facciamo un po' di ordine.

- ❖ Bug texture
- ❖ Troppe cose hard coded
- ❖ Menu
- ❖ Toolbar
- ❖ Gestione animazione

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

9

## Moebius Ristrutturata

- ❖ Adesso l'anello è strutturato in blocchettini
- ❖ ogni blocchettino è una singola mesh.
- ❖ Scopo:
  - ❖ poter piazzare I vari personaggi dove voglio
  - ❖ spiegarvi la selection...

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

10

## glSelection

- ❖ Meccanismo fornito da opengl per capire cosa fa a finire in una data porzione dello schermo.
- ❖ Quindi per capire, con precisione, cosa si trova sotto il mouse
- ❖ Sfrutta la pipeline di rendering
- ❖ Non fa rasterizzazione

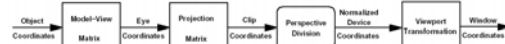
6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

11

## glSelection

- ❖ In pratica si definisce una nuova matrice di proiezione che inquadra solo la regione scelta
- ❖ Si rimanda tutta la geometria della scena, dando ad ogni entità un nome
- ❖ La geometria attraversa solo la prima parte della pipeline (no rasterization)
- ❖ I nomi di quello non viene clippato tutto fuori dal volume di vista vengono salvati in un buffer.



6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

12

## glSelection in pratica

- ❖ Per scoprire cosa passa nel punto x,y occorre:
- ❖ Passare alla modalità di rendering selection
- ❖ `glRenderMode(GL_SELECT);`
- ❖
- ❖ Inizializzare lo stack dei nomi
- ❖ `glInitNames();`  
`// LoadName() won't work with no names on the stack`  
`glLoadName(-1);`
- ❖ Preparare un buffer dove opengl metterà i nomi delle entità che cadono nella zona d'interesse
- ❖ `static unsigned int selectBuf[16384];`
- ❖ `glSelectBuffer(16384, selectBuf);`

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

13

## glSelection

- ❖ Modificare la matrice di proiezione in modo che mi clippi quello che mi interessa
- ❖ Occorre PRE-moltiplicarla per una matrice fatta apposta e relativa al punto che vogliamo e al viewport corrente:

```
int viewport[4];
glGetIntegerv(GL_VIEWPORT,viewport);
glMatrixMode(GL_PROJECTION);
double mp[16];
glGetDoublev(GL_PROJECTION_MATRIX ,mp);
glPushMatrix();
glLoadIdentity();
gluPickMatrix(x, y, 4, 4, viewport);
glMultMatrixd(mp);
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

14

## glSelection

- ❖ Rendering con i nomi per oggetti che voglio scegliere
- ❖ `glLoadName(int)`
- ❖ Nota è uno stack
- ❖ `glPushName()`
- ❖ `glPopName()`

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

15

## glSelect

- ❖ Dopo il rendering si rimette tutto a posto
- ❖ `glMatrixMode(GL_PROJECTION);`
- ❖ `glPopMatrix();`
- ❖ `glMatrixMode(GL_MODELVIEW);`
- ❖ `int hits = glRenderMode(GL_RENDER);`
- ❖ E nel buffer che avevamo preparato opengl ha messo una serie di record con la seguente struttura:
  - ❖ int: Numero nomi nello stack
  - ❖ int,int: Minima e max depth
  - ❖ int ... int Lista nomi sullo stack

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

16

## Parsing del selection buffer

- ❖ Se non abbiamo usato lo stack (se non abbiamo fatto `glPushName`) i record sono tutti lunghi uguali e parsarli è facile:

```
vector< pair<double,unsigned int> > H;
for(int ii=0;ii<hits;ii++){
    TRACE("%ui %ui %ui %ui\n",
          selectBuf[ii*4 ],selectBuf[ii*4+1],
          selectBuf[ii*4+2],selectBuf[ii*4+3]);
    H.push_back( make_pair(
                  selectBuf[ii*4+1]/4294967295.0,
                  selectBuf[ii*4+3])
                );
}
sort(H.begin(),H.end());
TRACE("\n Closest is %i\n",H[0].second);
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

17

## In pratica

- ❖ Non fare la pick direttamente nell'handler del mouse
- ❖ Nel 90% dei casi va, ma potrebbe non essere disponibile il contesto opengl
- ❖ Salvarsi posizione del mouse;

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

18

## gluPickMatrix

```
int MoebiusStrip::PickElement(int x, int y, int& SideInd, int& BlockInd)
{
    long hits;
    static unsigned int selectBuf[16384];
    glSelectBuffer(16384, sBuf);
    glRenderMode(GL_SELECT);
    glInitNames(); glPushName(-1);
    int viewport[4]; glGetIntegerv(GL_VIEWPORT, viewport);
    double mp[16]; glGetDoublev(GL_PROJECTION_MATRIX, mp);
    glMatrixMode(GL_PROJECTION);
    glPushMatrix();
    glLoadIdentity();
    gluPickMatrix(x, viewport[3]-y, 4, 4, viewport);
    glMultMatrixd(mp);
    Draw(); // scene!
    glPopMatrix();
    glMatrixMode(GL_PROJECTION);
    glPopMatrix();
    glMatrixMode(GL_MODELVIEW);
    hits = glRenderMode(GL_RENDER);
    if (hits <= 0) return 0;
    vector< pair<double, unsigned int> > H;
    for (int ii=0; ii<hits; ii++)
        H.push_back( make_pair(sBuf[ii*4+1]/4294967295.0, sBuf[ii*4+3]));
    sort(H.begin(), H.end());
    Ind2sec(H[0].second, SideInd, BlockInd);
    return H[0].second;
}
```

6 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni

19

## GISelection in azione

- ❖ Per provare abbiamo legato, in Moebius8 al ctrl click del mouse la selezione e coloritura in verde di una sezione dell'anello.

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

20

## Aggiungiamo Bottoni e menu

- ❖ Di solito le entry dei menu costituiscono il modo base per fare qualcosa in un interfaccia
- ❖ I bottoni delle toolbar costituiscono un modo più rapido come gli hotkey.
- ❖ Quindi prima si fa I menu, poi di quello che si considera utile si fa anche I bottoni nelle toolbar.

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

21

## Aggiungere un menu

- ❖ Aggiungiamo la possibilità di fermare l'animazione.
- ❖ Aggiungiamo tra le risorse un menu con due animation con due voci
  - ❖ Play
  - ❖ Stop
- ❖ Notate che ci sono due risorse
  - ❖ Una per quando l'app non ha documenti aperti
  - ❖ Una per quando ha un doc



6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

22

## Gestori menu

- ❖ Aggiungiamo un booleano nel doc
- ❖ bool m\_bplay;
- ❖ E sempre nel doc I gestori degli eventi:
- ❖ void CMBDoc::OnAnimationStop()
  - ❖ {
    - ❖ m\_bplay=false;
- ❖ void CMBDoc::OnAnimationPlay()
  - ❖ {
    - ❖ m\_bplay=true;

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

23

## Gestione update

- ❖ Per ogni voce dei menu' il framework chiede come deve essere (abilitato/marcato ecc)
- ❖ Basta aggiungere l'handler del messaggio OnUpdate...
- ❖ void CMBDoc::OnUpdateAnimationStop(CCmdUI \*pCmdUI)
  - ❖ { pCmdUI->Enable(m\_bplay); }
- ❖ void CMBDoc::OnUpdateAnimationPlay(CCmdUI \*pCmdUI)
  - ❖ { pCmdUI->Enable(!m\_bplay); }
- ❖ In questo modo quando l'animazione gira si puo' solo premere stop e viceversa

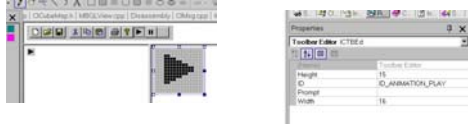
6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

24

## Toolbar

- ❖ Per aggiungere i corrispondenti bottoni, basta fare attenzione ad usare lo stesso id di risorsa



6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

25

## Gestione Animazione

- ❖ Rimane solo da gestire effettivamente l'azione di stop e play, ma questo non riguarda l'interfaccia ma come è stata strutturata la mia scena... (poco e male)
  - ❖ Aggiungo un campo nel doc che tiene quanto tempo è passato CMBDoc::ElapsedSec;
  - ❖ Si aggiorna solo tramite una funzione Doc::UpdateTime() da chiamare il più spesso possibile e che controlla se m\_bplay è true;
  - ❖ Si chiama UpdateTime() nella onidle e nella mousemove (durante il dragging non l'app non è idle)
  - ❖ La vista lo legge, per capire come e dove disegnare tutto.

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

26