

Costruzione di Interfacce Lezione 25 anatomia di uno screen saver

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

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

1

- ❖ Uno screen saver deve fare almeno
 - ❖ Screensaver
 - ❖ Preview
 - ❖ Dialogo di config
- ❖ Quale di queste tre cose deve fare è deciso in base alla command line con cui è invocato.
 - ❖ /c dialogo config
 - ❖ /p <HWND> preview come finestra figlia di <hwnd>
 - ❖ /s screen saver

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

2

InitInstance

- ❖ Per questo motivo aggiungiamo alla WinApp le seguenti funzioni
 - ❖ `void DoSaverSetting(void);`
 - ❖ `void DoSaverPreview(void);`
 - ❖ `void DoScreenSaver(void);`
 - ❖ `void ReadSetting();`
 - ❖ `void ReadSetting();`
- ❖ La `initinstance` deve parsare la commandline e decidere cosa far partire
- ❖ Il parsing della command line va fatto abbastanza robusto (puo' arrivare varie cose...)

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

3

InitInstace: Parsing

```
BOOL CCISaverApp::InitInstance()
{
    InitCommonControls(); CWinApp::InitInstance();
    AfxEnableControlContainer();
    int pos=0; CString tokem_ipCmdLine.Tokenise(" ",pos);
    if(!tok.CompareNoCase("/s") || !tok.CompareNoCase("--s")
        || !tok.CompareNoCase("s")) {
        DoScreenSaver(); // Run as screen saver
        return TRUE; // and continue the message loop
    } else if ( !tok.CompareNoCase("/c") || !tok.CompareNoCase("-c")
        || !tok.CompareNoCase("c")) {
        DoSaverSetting(); // Run modal config dialog
        return FALSE; // and Now just terminate
    } else if ( !tok.CompareNoCase("/p") || !tok.CompareNoCase("-p")
        || !tok.CompareNoCase("p")) {
        DoSaverPreview(); // Show the saver in a small win
        return TRUE; // and continue the message loop
    }
    return FALSE; // if no param terminate;
}
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

4

Strutturazione classi

- ❖ Ci servono due classi
 - ❖ Una che sappia crearsi in maniera abbastanza flessibile, abbia un contesto opengl e sappia disegnare lo screen saver
 - ❖ Una che sia capace di chiudersi non appena l'utente fa qualcosa
- ❖ Deriveremo la seconda dalla prima.
 - ❖ `CGLWnd` derivata da `CWnd`;
 - ❖ `CSaverWnd` derivata da `CGLWnd`

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

5

CGLWnd

- ❖ Facciamo aggiungere all'ide una classe mfc derivata da `cwnd`
- ❖ Come membri al solito mettiamo

```
HGLRC m_hRC; // Rendering device context of OpenGL
HDC m_hDC; // Current device context
float aspect;
BOOL SetupPixelFormat( HDC hDC );
void glInit();
inline BOOL SetGL()
```
- ❖ Presi dal solito file txt...

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

6

CGLWnd

- ❖ Dobbiamo rifare la create, per poter rispondere alle varie necessità di creazione.

```
BOOL CGLWnd::myCreate(DWORD dwExStyle, DWORD dwStyle,
                    const RECT& rect, CWnd* pParentWnd, UINT nID)
{
    TRACE("CGLWnd::myCreate\n");
    // Register a class with no cursor
    LPCTSTR lpszClassName= AfxRegisterWndClass(CS_HREDRAW|CS_VREDRAW,
        ::LoadCursor(
            AfxGetResourceHandle(), MAKEINTRESOURCE(IDC_NULL_CURSOR) )
        );

    return CreateEx(dwExStyle, lpszClassName, _T(""), dwStyle,
        rect.left, rect.top,
        rect.right - rect.left, rect.bottom - rect.top,
        pParentWnd->GetSafeHwnd(), NULL, NULL );
}
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

7

Window Classes

- ❖ Una *window class* è un insieme di attributi che il sistema usa come template per creare una finestra.
- ❖ Ogni finestra è membro di una *window class*
- ❖ Tutte le *window class* sono specifiche di un processo
- ❖ Ogni *window class* ha una procedura associata che processa i messaggi di tutte le finestre di quella classe
- ❖ Un processo deve registrare una *window class* prima di poter creare una finestra di quella classe
- ❖ Registrare una *window class* associa una window procedure, un'insieme di stili, ecc ad un nome.

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

8

Cursore nullo

- ❖ Nella finestra dello screen saver non ci deve essere il cursore quindi
- ❖ Creare una risorsa con un cursore vuoto
- ❖ Usarla nella creazione della window class

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

9

Timer

- ❖ Il nostro screen saver non puo'usare il meccanismo della onidle per l'animazione perchè quando è invocato nel preview non ho accesso al thread iniziale.

- ❖ Quindi si usano timer...

- ❖ Aggiungiamo tra i membri

```
❖ BOOL m_bHasTimer; //flag for timer in this child window
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

10

OnCreate

- ❖ Overridiamo anche la OnCreate, dove facciamo tutte le init di gl e del timer

```
int CGLWnd::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
    if (CWnd::OnCreate(lpCreateStruct) == -1) return -1;
    m_hDC = ::GetDC(m_hWnd);
    SetupPixelFormat(m_hDC);
    m_hRC = wglCreateContext(m_hDC);
    if(m_hRC==NULL){ AfxMessageBox("OpenGL contest fail");
        return -1;
    }
    if(!m_bHasTimer){ //set timer
        SetTimer(1, 10, NULL);
        m_bHasTimer = TRUE;
    }
    Invalidate();
    return 0;
}
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

11

OnSize

- ❖ La solita

```
void CGLWnd::OnSize(UINT nType, int cx, int cy)
{
    TRACE("CGLWnd: OnSize()\n");
    CRect rect;
    if(SetGL()){
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        aspect=(float)cx/(float)cy;
        glViewport(0, 0, (GLsizei) cx, (GLsizei) cy);
    }
}
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

12

OnPaint

- ❖ OnDraw è propria delle viste, qui siamo un po' più a basso livello. Si disegna nella onpaint.

```
void CGLWnd::OnPaint()
{
    CPaintDC dc(this); // device context for painting
    SetGL();
    glPushMatrix();
    glRotatef(30*clock()/1000.0f,0,0,1);
    glClearColor(.4,.3,.3,1);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3f(1,1,1);
    glBegin(GL_LINE_LOOP);
    glVertex2f(.3f,-.3f); glVertex2f(-.3f,-.3f);
    glVertex2f(-.3f,.3f); glVertex2f(.3f,.3f);
    glEnd();
    glPopMatrix();
    SwapBuffers(m_hDC);
}
6 Dicembre 2002      Costruzione di Interfacce - Paolo Cignoni      13
```

CSaverWnd

- ❖ Possiamo ora a scrivere la classe che deve incapsulare il fatto che la nostra finestra opengl
 - ❖ Parte massimizzata
 - ❖ Non appena l'utente fa qualcosa si distrugge
- ❖ Deriviamo CSaverWnd da CGLWnd
- ❖ E riscriviamo la Create

6 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni 14

- ❖ Crea la finestra grande quanto lo schermo, senza decorazioni, che stia sopra alle altre

```
BOOL CSaverWnd::Create()
{
    TRACE("CSaverWnd::Create()\n");

    CRect rect(0, 0, ::GetSystemMetrics(SM_CXSCREEN),
              ::GetSystemMetrics(SM_CYSCREEN) );

    return CGLWnd::myCreate(WS_EX_TOPMOST,
                            WS_VISIBLE|WS_POPUP, rect, NULL, 0);
}
6 Dicembre 2002      Costruzione di Interfacce - Paolo Cignoni      15
```

Messaggi da intercettare

- ❖ Per far terminare lo screen saver:
 - ❖ OnActivate
 - ❖ OnActivateApp
 - ❖ OnKeyDown
 - ❖ On{L/M/R}ButtonDown
 - ❖ OnMouseMove
 - ❖ OnSysKeyDown
- ❖ In risposta ai quali basta fare:

6 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni 16

Altri messaggi

```
void CSaverWnd::OnSysCommand(UINT nID, LPARAM lParam) {
    if ((nID == SC_SCREENSAVE) || (nID == SC_CLOSE))      return;
    CGLWnd::OnSysCommand(nID, lParam);
}

void CSaverWnd::OnDestroy() {
    PostQuitMessage(0);
    CGLWnd::OnDestroy();
}

BOOL CSaverWnd::OnSetCursor(CWnd* pWnd, UINT nHitTest, UINT message) {
    SetCursor(NULL);
    return TRUE; // per notificare che abbiamo gestito noi il messaggio
}
6 Dicembre 2002      Costruzione di Interfacce - Paolo Cignoni      17
```

L'app

- ❖ Si deve ora scrivere le funzioni che fanno partire tutto.

```
void CCISaverApp::DoScreenSaver(void)
{
    TRACE("CCISaverApp::DoScreenSaver()\n");
    m_psvwnd = new CSaverWnd;
    ReadSetting();
    m_psvwnd->Create();
    m_pMainWnd = m_psvwnd;
    TRACE("exiting CCISaverApp::DoScreenSaver(void)\n");
}
6 Dicembre 2002      Costruzione di Interfacce - Paolo Cignoni      18
```

Gestione della preview

```
void CCISaverApp::DoSaverPreview(void)
{
    CString tok,cmdln=m_lpCmdLine;
    int pos=0;
    tok=cmdln.Tokenize(" ",pos);
    tok=cmdln.Tokenize(" ",pos);
    TRACE("Second token in command line is '%s'\n",tok);
    CWnd* pParent = CWnd::FromHandle((HWND)atol(tok));
    ASSERT(pParent != NULL);
    CGLWnd* pWnd = new CGLWnd();
    ReadSetting();
    CRect rect;
    pParent->GetClientRect(&rect);
    pWnd->myCreate(NULL, WS_VISIBLE|WS_CHILD, rect, pParent,0);
    m_pMainWnd = pWnd;
}
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

19

Preview nel dialogo Settings

- ❖ Nel dialogo della classe CISaverDlg aggiungiamo una finestra di preview.
- ❖ Aggiungiamo un membro
CGLWnd m_glwnd;
- ❖ Nelle risorse, nel template del dialog aggiungere uno static text control della dimensione che ci interessa con id IDC_IMAGE
- ❖ Nella OnInitDialog aggiungiamo:

```
CRect rect;
GetDlgItem(IDC_IMAGE)->GetWindowRect(&rect);
ScreenToClient(&rect);
m_glwnd.myCreate(WS_EX_TOPMOST, WS_VISIBLE|WS_CHILD, rect,
this, IDC_IMAGE);
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

20

un po' di interfaccia...

- ❖ Mettiamo nel dialogo



- ❖ una variabile CString m_filename associata all'edit control
- ❖ l'handler della pressione del bottone ".."

```
void CCISaverDlg::OnBnClickedButtonBrowse()
{
    CFileDialog fd(true, "*.txt", "test.txt");
    if (fd.DoModal() == IDOK) {
        m_filename = fd.GetFileName();
        UpdateData(false);
    }
}
```

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

21

Note finali

- ❖ Per mettere l'estensione a *.scr basta modificare le opzioni del linker
- ❖ Per la stringa che si vede nella descrizione degli screen saver basta fare una risorsa stringa IDS_DESCRIPTION di valore 1

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

22

Salvare setting

- ❖ Di solito si usa il registro
- ❖ Conviene chiudere tutto in una classe e tenuta in winapp e passarla alla dialog.
- ❖ In CWinApp
 - ❖ **BOOL WriteProfileInt**(sezione, entry, val);
 - ❖ **BOOL WriteProfileString**(sezione, entry, str);
 - ❖ **Bool GetProfileString**(sezione, entry, default,buf,sizebuf)
 - ❖ **Int GetProfileInt** (sezione, entry, default);

6 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

23