

Costruzione di Interfacce Lezione 28 Riordino e Scene Graph III

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

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

1

Introduzione

- ❖ Riordino e merge dei due progetti screen saver e editor in un'unica soluzione.
- ❖ Scene Graph design
- ❖ Adattare lo scene graph

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

2

Riordiniamo

- ❖ Facciamo un'unica Soluzione con due progetti dentro, uno per l'editor e uno per il saver:
 - ❖ Partite dall'editor
 - ❖ Spostare le directory e i file
 - ❖ Mettete la sln nella root
 - ❖ lasciate il vcproj dove volete che vi faccia le dir debug/release (sotto editor)
 - ❖ cancellate e riaggiungete il progetto
 - ❖ Cancellate e riaggiungete tutti i file che si trovano in path relativi diversi
 - ❖ Aggiustate tutti i path degli include
- ❖ Aggiungete lo screensaver

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

3

Riordiniamo

- ❖ Cerchiamo di isolare quello che serve per disegnare l'anello dal resto della vista/doc
 - ❖ Alla fine solo il codice condiviso tra saver e editor è composto da:
 - ❖ Tutte le classi sotto vcg e sotto code
 - ❖ La classe moebiusstrip e le altre classi accessorie (pallina e movimento) derivate da CSGNode.

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

4

Sistemiamo lo screen Saver

- ❖ Nello Screen saver dobbiamo
 - ❖ Aggiungere un oggetto globale
 - ❖ CSGMb Scene;
 - ❖ Creare una funz di inizializzazione

```
void CCISaverApp::Init(CString &name)
{
    Scene.Clear();
    if(name!="" && (00==name.Right(3).CompareNoCase("xml")))
    {
        XmlDoc xml;
        if(xml.read(LPCSTR (name) ))
        {
            Scene.root.XMLRead(xml.root(), &Scene);
            Scene.m->Generate();
        }
    }
}
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

5

OnPaint

```
void CGLWnd::OnPaint()
{
    CPaintDC dc(this); // device context for painting
    SetGL();
    glEnable(GL_LIGHTING);
    glEnable(GL_NORMALIZE);
    glEnable(GL_DEPTH_TEST);
    glColor3f(1,1,1);

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60,aspect, .4,25);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt(0,0,10,0,0,0,1,0); // World To Camera Transformation
    Scene.root.gDraw(float(clock())/1000.0f);

    SwapBuffers(m_hDC);
}
```

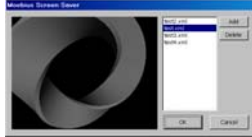
13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

6

Screen saver

- ❖ Modifichiamo l'interfaccia dello screen saver
- ❖ Togliamo l'editbox e aggiungiamo un listbox



- ❖ La configurazione dello screensaver consiste:
 - ❖ un elenco di nomi di file (pathname assoluti!) generati con l'editor.
 - ❖ Un indice che dice qual'è quello corrente

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

7

Interfaccia screen saver

- ❖ Aggiungo I seguenti membri alla classe CCISaverDialog

```
vector<CString> m_names;  
int m_select;
```

```
CListBox m_listSaver;
```

- ❖ e alla classe CCISaverApp

```
vector<CString> names;
```

```
int Starting;
```

- ❖ Leggiamo/Salviamo tutti questi dati nel registro

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

8

Read e write Settings

```
void CCISaverApp::ReadSetting()  
{  
    names.clear();  
    int i=0;  
    bool ended=false;  
  
    while(!ended)  
    {  
        CString tmp;  
        CString buf; buf.Format("SaverName%03i",i);  
        tmp=GetProfileString("CISaver", buf, "");  
        if(tmp!=""){  
            names.push_back(tmp);  
            ++i;  
        }  
        else ended = true;  
    }  
    Starting=GetProfileInt("CISaver", "Starting", 0);  
    if(names.empty()) Init(CString());  
    else Init(names[Starting]);  
}
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

9

WriteSetting

```
void CCISaverApp::WriteSetting()  
{  
    for(int i=0;i<names.size();++i)  
    {  
        CString buf; buf.Format("SaverName%03i",i);  
        WriteProfileString("CISaver", buf, names[i]);  
    }  
    CString buf; buf.Format("SaverName%03i",i);  
    WriteProfileString("CISaver", buf, "");  
    WriteProfileInt("CISaver", "Starting", Starting);  
}
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

10

DoSaverSetting

```
void CCISaverApp::DoSaverSetting(void)  
{  
    CCISaverDlg dlg;  
    CString tok,cmdln=_lpCmdLine;  
    ReadSetting();  
    dlg.m_names=names;  
    dlg.m_select=Starting;  
    m_pMainWnd = &dlg;  
    if (dlg.DoModal() == IDOK)  
    {  
        names=dlg.m_names;  
        Starting=dlg.m_select;  
        WriteSetting();  
    }  
}
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

11

Screen saver

- ❖ Quando parte con il dialogo di configurazione l'app crea un oggetto dialogo, ne riempie alcune strutture con I parametri letti dal registro e passa il controllo al dialog (doModal)
- ❖ Alla fine se l'utente ha premuto ok si copiano le strutture dati del dialog in quelle dell'app, si salvano nel registro e si esce.

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

12

On InitDialog

```
BOOL CCISaverDlg::OnInitDialog()
{
    CDialog::OnInitDialog();
    SetIcon(m_hIcon, TRUE);           // Set big icon
    SetIcon(m_hIcon, FALSE);        // Set small icon

    // Creazione finestra opengl
    CRect rect;
    GetDlgItem(IDC_IMAGE)->GetWindowRect(&rect);
    ScreenToClient(&rect);
    m_glwnd.myCreate(WS_EX_TOPMOST, WS_VISIBLE|WS_CHILD, rect, this, IDC_IMAGE);
    // init del list box
    for(unsigned i=0;i<m_names.size();++i){
        CString buf=m_names[i];
        int lastSlash=buf.ReverseFind('\\');
        m_listSaver.AddString(buf.Right(buf.GetLength()-lastSlash-1));
    }
    m_listSaver.SetCurSel(m_select);
    return TRUE; // return TRUE unless you set the focus to a control
}
```

13 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni

13

Gestione interfaccia

- ❖ Nel dialogo dobbiamo gestire come eventi:
 - ❖ Aggiunta/cancellazione di un nome dal listbox
 - ❖ Selezione di un elemento nel box.
 - ❖ Nota l'elenco dei nomi dei file è tenuto in due posti
 - ❖ Nel list box (in forma abbreviata)
 - ❖ Nel vettore m_names

13 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni

14

Add

```
void CCISaverDlg::OnBnClickedButtonBrowse()
{
    CFileDialog fd(true, "*.txt", "test.txt");
    if(fd.DoModal()==IDOK) {
        int pos=m_listSaver.FindStringExact(0,fd.GetFileName());
        if(pos!=-1) AfxMessageBox("File già scelto!");
        else{
            m_names.push_back(fd.GetPathName());
            int pos=m_listSaver.AddString(fd.GetFileName());
            assert((pos+1)==m_names.size());
            m_listSaver.SetCurSel(pos);
            m_select=pos;
            theApp.Init(m_names[pos]);
        }
    }
};
```

13 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni

15

Delete e update

```
void CCISaverDlg::OnBnClickedButtonDelete()
{
    int pos=m_listSaver.GetCurSel();
    m_names.erase(m_names.begin()+pos);
    m_listSaver.DeleteString(pos);
}

void CCISaverDlg::OnLbnSelChangeListSaver()
{
    m_select=m_listSaver.GetCurSel();
    theApp.Init(m_names[m_select]);
}
```

13 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni

16

Finiamo di pulire doc/view

- ❖ Aggiungiamo allo scene graph
 - ❖ Nodo Light
 - ❖ Nodi Trasformazioni
 - ❖ Nodo Texture
- ❖ Come per le matrici hanno lo scope dei gruppi.
 - ❖ Nota in alcune situazioni potrebbe convenire fare gruppi che non chiudano lo scope per luci e texture.

13 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni

17

```
void CSGGroup::glDraw(const float DocTime)
{
    glPushMatrix();
    if(SaveLightState) glPushAttrib(GL_LIGHTING_BIT);
    iterator i;
    for(i=Sons.begin();i!=Sons.end();++i)
        (*i)->glDraw(DocTime);
    if(SaveLightState) glPopAttrib();
    glPopMatrix();
}
```

13 Dicembre 2002 Costruzione di Interfacce - Paolo Cignoni

18

Nodo Light

```
class CSGLight :public CSGNode
{
public:
    CSGLight(int id=0);
    int Id;
    Point4f Position;
    Point4f AmbientColor;
    Point4f DiffuseColor;
    Point4f SpecularColor;

    virtual void glDraw(const float DocTime);
    virtual void XMLWrite(FILE *fp);
    virtual void XMLRead(Xml &xml, CSG *Base);
};
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

19

Nodo Light

```
void CSGLight::glDraw(const float DocTime)
{
    glEnable(GL_LIGHT0+Id);
    glLightfv(GL_LIGHT0+Id, GL_DIFFUSE, DiffuseColor.V());
    glLightfv(GL_LIGHT0+Id, GL_AMBIENT, AmbientColor.V());
    glLightfv(GL_LIGHT0+Id, GL_SPECULAR, SpecularColor.V());
    glLightfv(GL_LIGHT0+Id, GL_POSITION, Position.V());
}
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

20

Nodo Light

```
class CSG
{
private:
    int LightCounter;
public:
    CSG(void);
    ~CSG(void);
    CSGGroup root;
    virtual CSGNode *Allocate(const string &classname);
    virtual void Clear();
    int Pick(int x, int y, const float &DocTime);
};

void CSG::Clear()
{
    root.Clear();
    LightCounter=0;
}
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

21

Nodo Light

```
CSGNode *CSG::Allocate(const string &classname)
{
    CSGNode *pt=0;
    if(classname=="CSGLight") pt= new CSGLight(LightCounter++);
    if(classname=="CSGGroup") pt= new CSGGroup;
    if(classname=="CSGRotation") pt= new CSGRotation;
    if(classname=="CSGTranslation") pt= new CSGTranslation;
    if(classname=="CSGTransformation") pt= new
        CSGTransformation;
    if(classname=="CSGAnimRotation") pt= new CSGAnimRotation;
    if(classname=="CSGAnimZPrecession") pt= new
        CSGAnimZPrecession;
    return pt;
}
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

22

Nodo Light: Read e Write

```
void CSGLight::XMLWrite(FILE *fp)
{
    fprintf(fp, "<CSGLight>");
    fprintf(fp, "DC0= %f DC1= %f DC2= %f DC3= %f\n",
        DiffuseColor[0], DiffuseColor[1], DiffuseColor[2], DiffuseColor[3]);
    fprintf(fp, "AC0= %f AC1= %f AC2= %f AC3= %f\n",
        AmbientColor[0], AmbientColor[1], AmbientColor[2], AmbientColor[3]);
    fprintf(fp, "SC0= %f SC1= %f SC2= %f SC3= %f\n",
        SpecularColor[0], SpecularColor[1], SpecularColor[2], SpecularColor[3]);
    fprintf(fp, "P0= %f P1= %f P2= %f P3= %f\n",
        Position[0], Position[1], Position[2], Position[3]);
    fprintf(fp, ">");
}

void CSGLight::XMLRead(Xml &xml, CSG *Base)
{
    assert(xml.id == "CSGLight");
    DiffuseColor[0] = (float)atof(xml["DC0"].c_str()); DiffuseColor[1] = (float)atof(xml["DC1"].c_str());
    AmbientColor[0] = (float)atof(xml["AC0"].c_str()); AmbientColor[1] = (float)atof(xml["AC1"].c_str());
    SpecularColor[0] = (float)atof(xml["SC0"].c_str()); SpecularColor[1] = (float)atof(xml["SC1"].c_str());
    Position[0] = (float)atof(xml["P0"].c_str()); Position[1] = (float)atof(xml["P1"].c_str());
    DiffuseColor[2] = (float)atof(xml["DC2"].c_str()); DiffuseColor[3] = (float)atof(xml["DC3"].c_str());
    AmbientColor[2] = (float)atof(xml["AC2"].c_str()); AmbientColor[3] = (float)atof(xml["AC3"].c_str());
    SpecularColor[2] = (float)atof(xml["SC2"].c_str()); SpecularColor[3] = (float)atof(xml["SC3"].c_str());
    Position[2] = (float)atof(xml["P2"].c_str()); Position[3] = (float)atof(xml["P3"].c_str());
}
```

13 Dicembre 2002

Costruzione di Interfacce - Paolo Cignoni

23