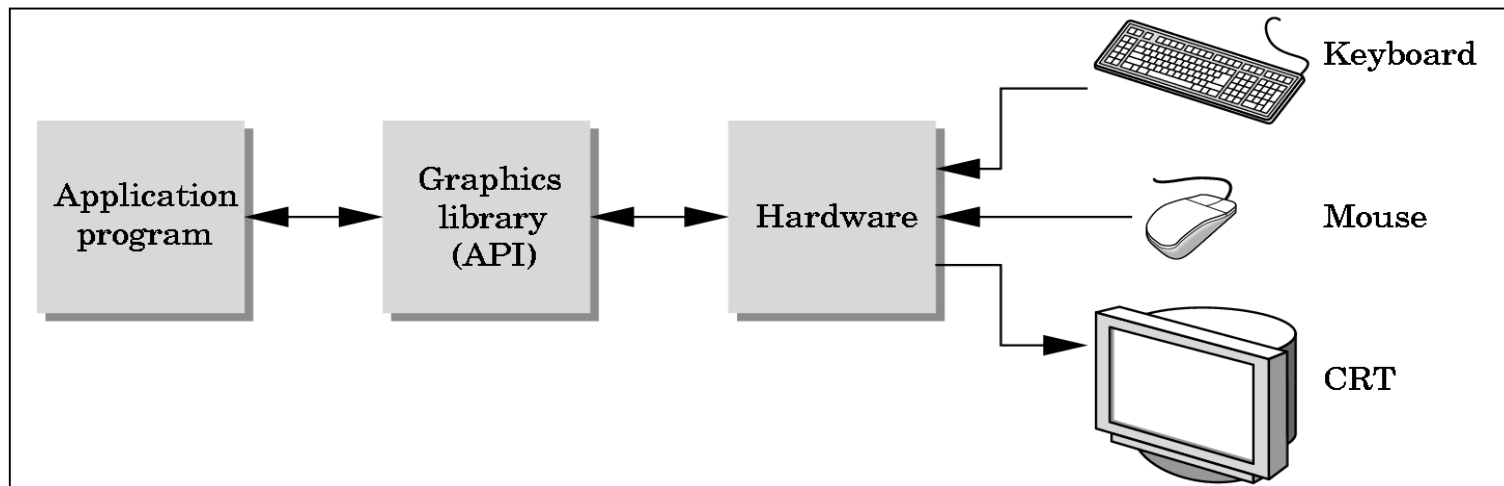
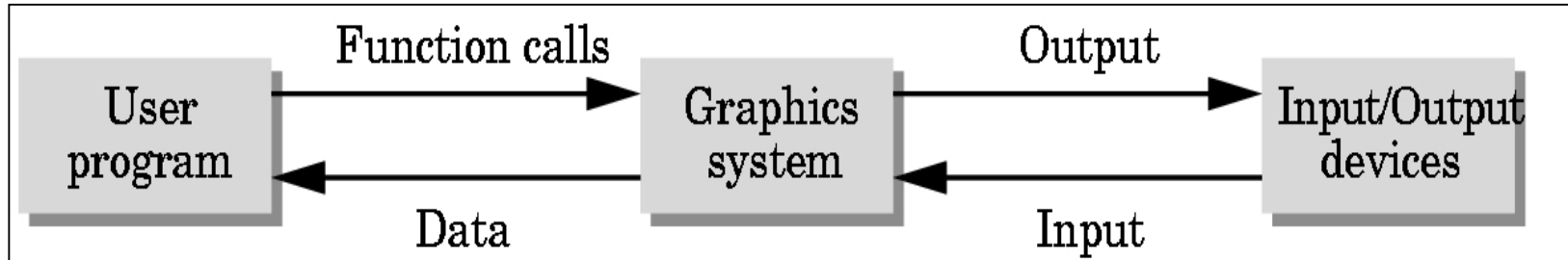


Introduzione a OpenGL

- La libreria OpenGL è stata sviluppata da Silicon Graphics;
- API tra Software e Hardware;
- 150 comandi;
- Hardware indipendente;



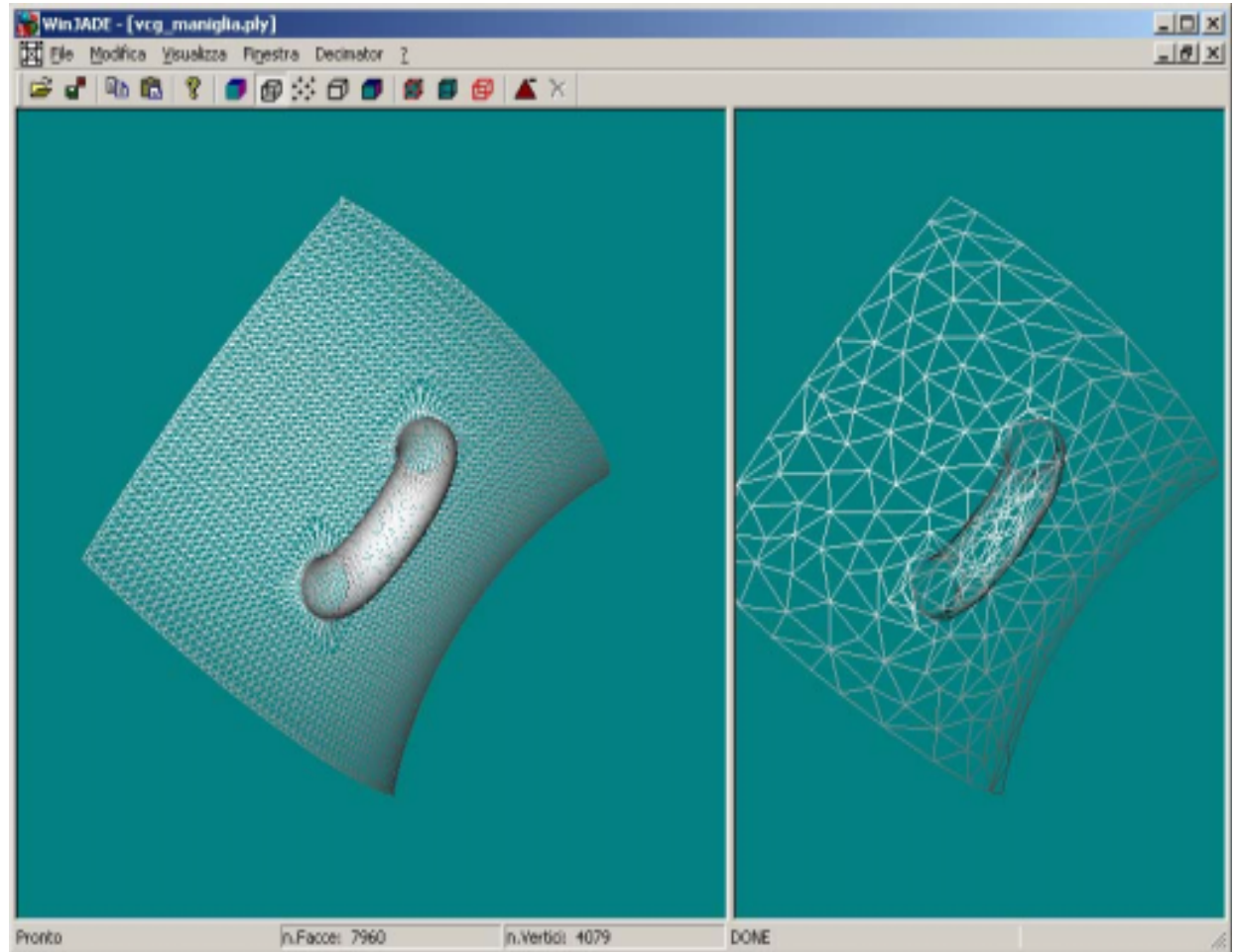
Introduzione a OpenGL (2)



Un software grafico riceve in input dati che vengono elaborati e quindi visualizzati tramite opportune chiamate di funzioni alla libreria (OpenGL / Window System).

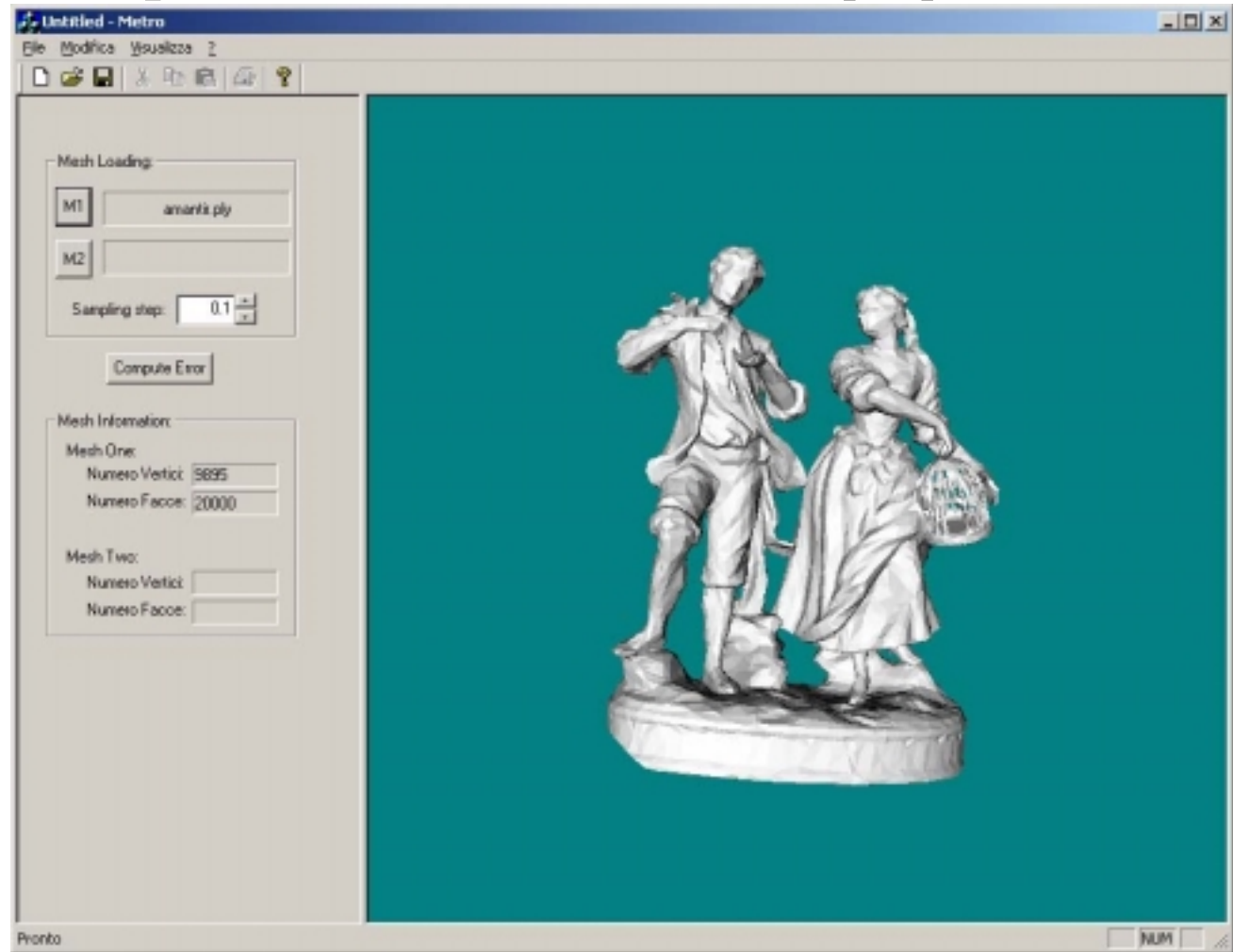
Esempio software:

WinJADE:
semplificatore di
Mesh
tridimensionali
sviluppato presso
il **CNR** di Pisa,
**Istituto di
Elaborazione
dell'informazione:
Visual
Computing
Group.**



Esempio software (2):

METRO:
misura
l'errore tra
mesh
originale e
mesh
semplificata.

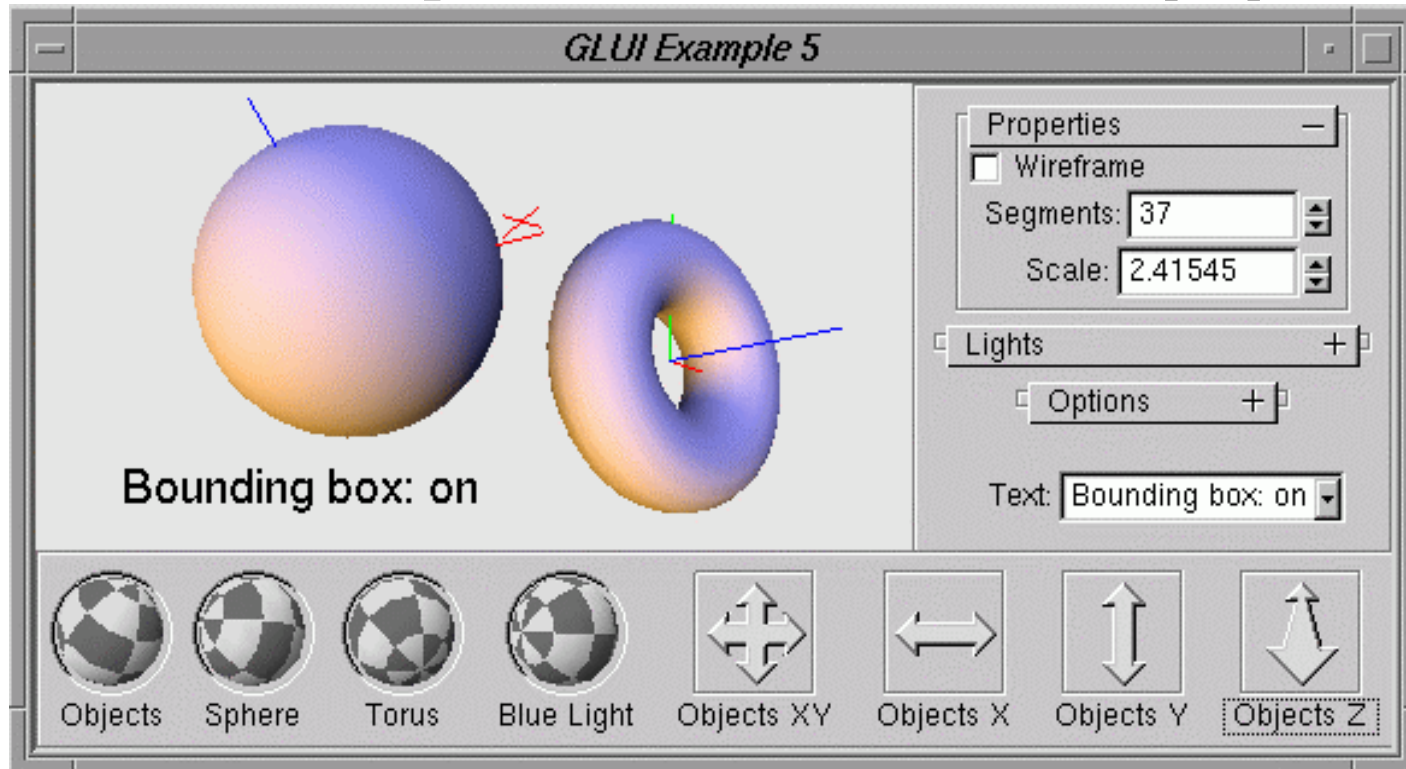


Esempio software (3):

Output di Metro:
mesh semplificata+
texture Mapping
dell'errore

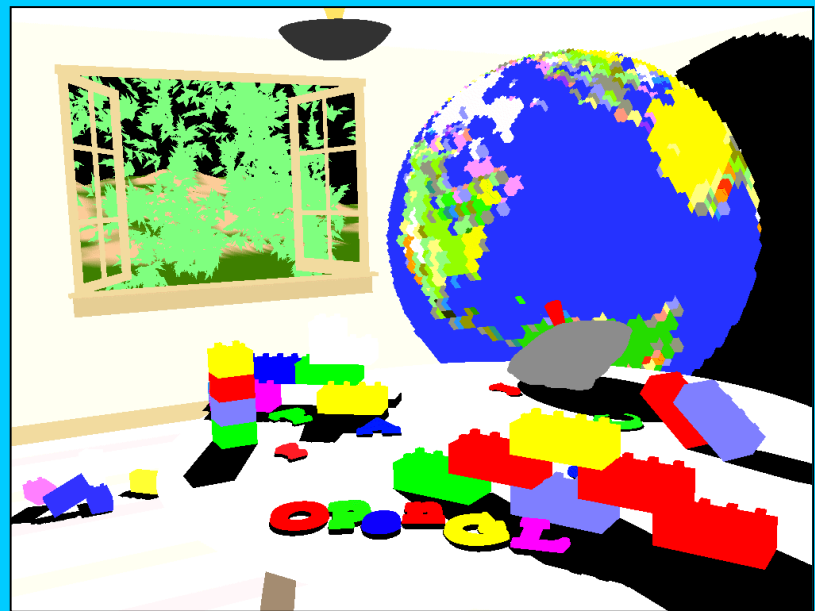
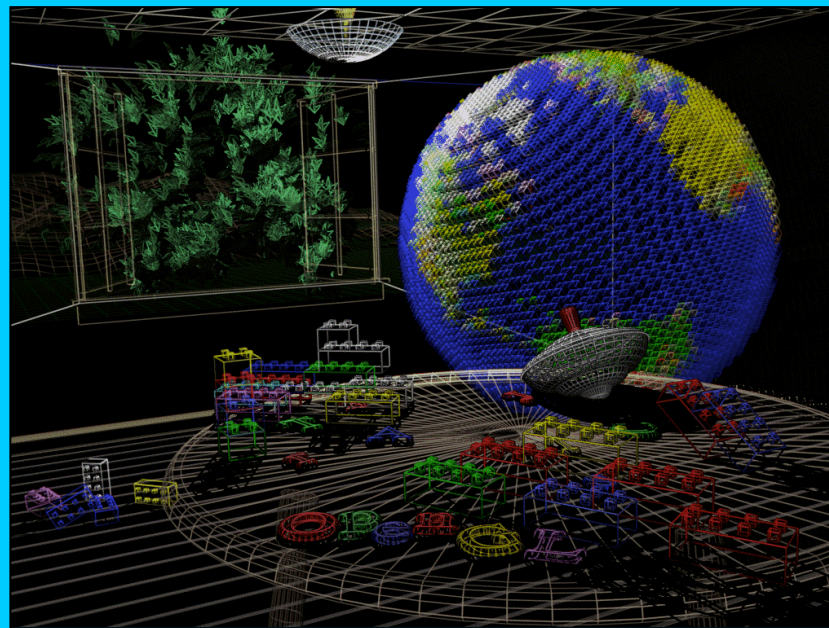


Esempio software (4):

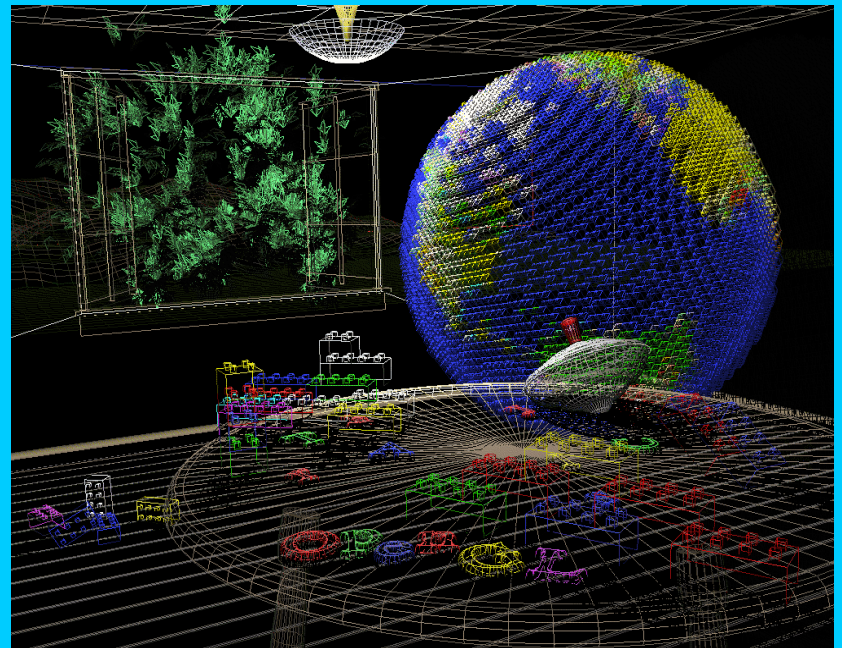
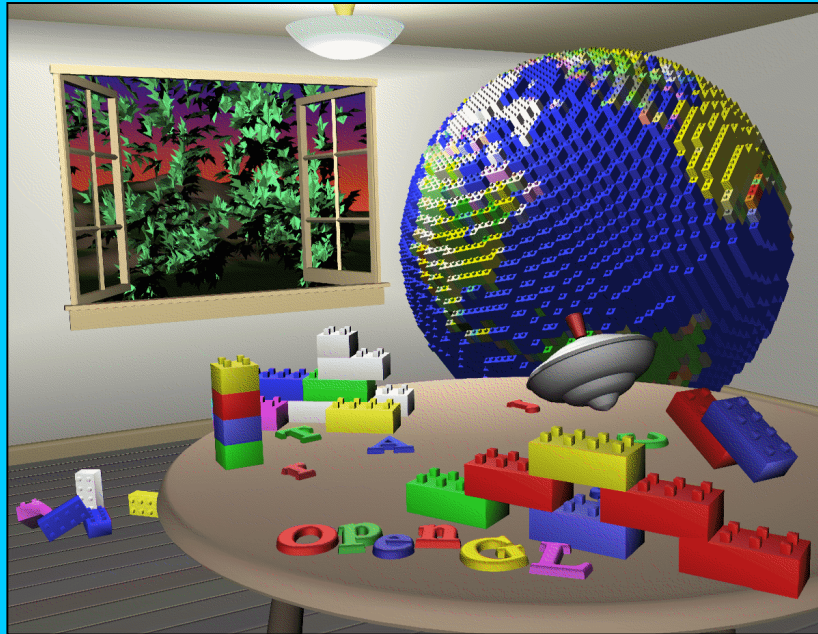


OpenGL + GLUT + GLUI

Esempi



Esempi (2)



Introduzione a OpenGL (3)

OpenGL è una macchina a STATI.

Ogni comando permette di passare da uno stato all'altro.

Esempi di STATO possono essere:

- Il colore di visualizzazione;
- La posizione di un punto;
- L'impostazione di una luce;
- Il refresh del video;
- La rotazione di un oggetto;
- ecc...

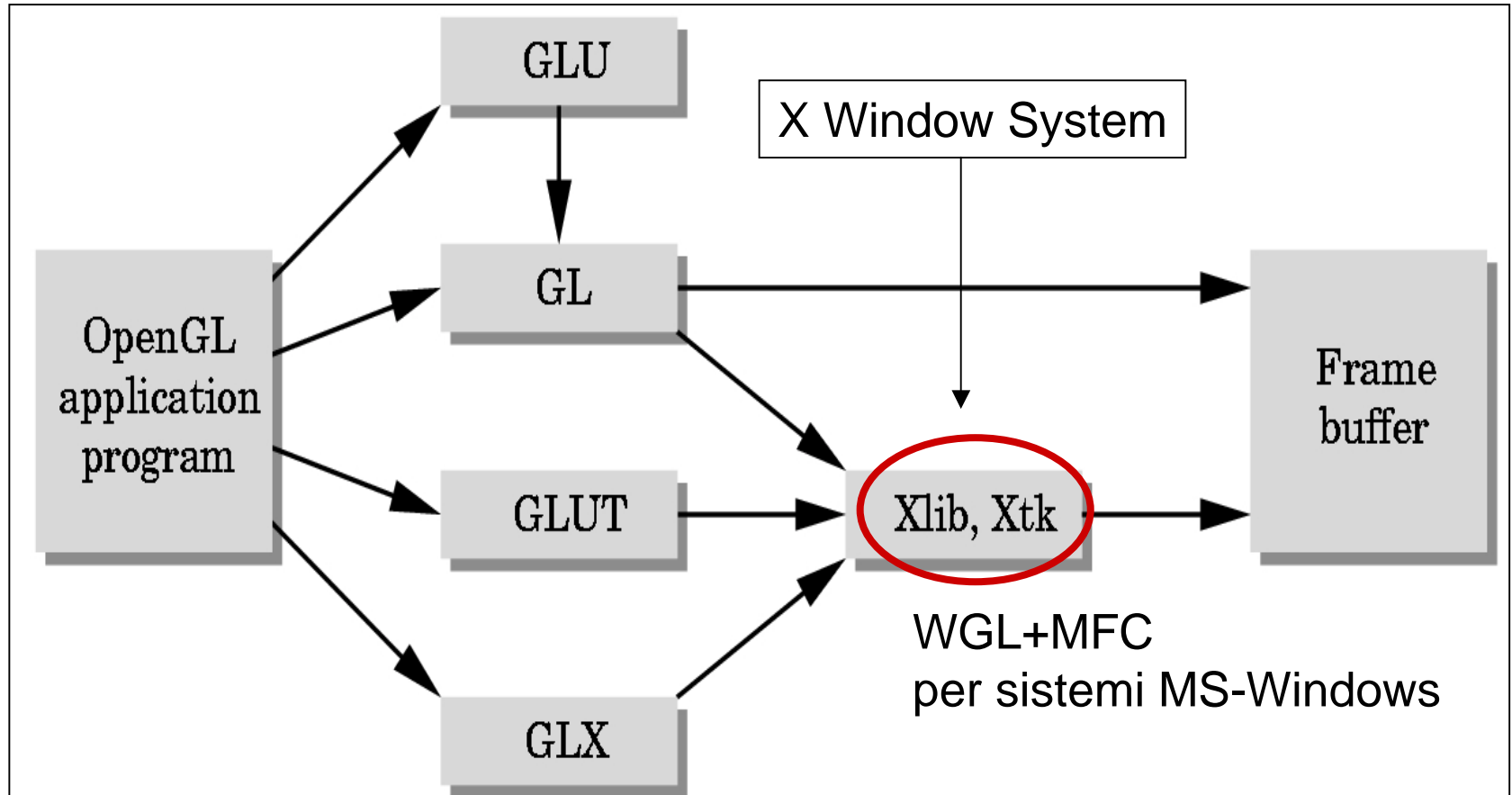
Introduzione a OpenGL (4)

GLUT (OpenGL Utility Toolkit): permette di utilizzare le chiamate al window system consentendo la creazione di finestre, menù, ...;

GLUI (GLUT-based User Interface Library): libreria C++ che completa le GLUT con la creazione di oggetti interfaccia come: bottoni, checkbox, spinbutton, textfield, ...;

OpenGL: la libreria di visualizzazione grafica di base.

Introduzione a OpenGL (5)



OpenGL (1)

I comandi di OpenGL:

- prefisso **gl** per i comandi;
- prefisso **GL_** costanti predefinite;
- **gl_____2[3]{sifdv}** comandi per visualizzazione dati, impostazione colori, ecc... ;

Tipi numerici:

- int → **GLint**;
 - float → **GLfloat**;
 - double → **GLdouble**;
- ecc ...

Es:

```
glClearColor(0.0,0.0,0.0,0.0);  
glClear(GL_COLOR_BUFFER_BIT);  
glColor3f(1.0,1.0,1.0);  
glVertex3f(0.25,0.75,0.0);
```

OpenGL (2): main.cpp

Struttura del main.cpp:

```
#include <LIBRERIE .h>
#include "Libreria di oggetti 2D-3D.h"

int [void] main(... , ...){
    FASE INIZIALIZZAZIONE/CREAZIONE;
    REGISTRAZIONE callback functions;
    START ciclo di eventi;
}
```

← Creata da VOI!

OpenGL (3): main.cpp

```
#include <LIBRERIE .h>
#include <Librerie oggetti 2D-3D.h>
void InitGLUT(void)
{Finestre, menu, mouse, reshape, display, ...}
void InitGLUI(void)
{bottoni, checkbox, textfield, spinner, spaceball}
void InitGL(void)
{
sistemi di riferimento, colori, visualizzazione, luci,...
}
int [void] main(int argc, char *argv[]){
    glutInit(&argc, argv);
    InitGLUT();
    InitGLUI();
    InitGL();
    glutMainLoop();
}
```

Inizializzazione/Creazione
Registrazione

Ciclo eventi

OpenGL (4): main.cpp

```
glutInit(&argc, argv);
```

Inizializza la libreria GLUT e si collega con il window system.

```
glutMainLoop();
```

Avvia il ciclo che catturerà gli eventi da elaborare.

gli eventi vengono gestiti richiamando le funzioni di callback registrate in fase di inizializzazione.

Configurazione

VISUAL C++:

- includere gli opportuni header nel file .cpp: `#include<GL/glut.h>;`
- Dal menù principale selezionare:
- PROJECT → *Settings* (ALT+F7);
- Selezionare il tag: LINK per le opzioni di *linking*;
- Alla voce Objects/Library modules inserire nell'ordine:

[glui32.lib] glu32.lib glut32.lib opengl32.lib

NB: Verificare che la voce *settings* for sulla sx sia posizionata su **All Configuration**.

Configurazione (2)

LINUX:

- In `/usr/include/GL/` inserire il file header: `glut.h`.

Devono essere già presenti `gl.h`, `glu.h`.

- Inserire in `/usr/include/` il file header: `glui.h`
- Inserire in `/usr/lib/` il file di libreria: `libglui.a`

Dovrebbero altresì essere presenti i seguenti file di libreria:

- 1) `libglut.so` oppure `libglut.a`
- 2) `libglut.so.3`
- 3) `libglut.so.3.6`

Esempio di makefile: linkare prima le glui delle glut !!!

```
LIB = -lglui -lglut -IGLU -IGL [-L/usr/X11R6/lib] -IXmu -IXi -IXext -LX11 -lm  
CC = g++ (oppure gcc,cc)
```

```
triang: triang.cpp  
      $(CC) $(LIB) triang.cpp -o triangle
```

Configurazione (3)

Librerie:

- OpenGL → `#include <GL/gl.h>`
- GLU → `#include <GL/glu.h>`: trasformazioni della finestra di visualizzazione, proiezioni, superfici e oggetti 3D.
- GLUT → `#include <GL/glut.h>`: interazioni dispositivi I/O, menu, oggetti 3D,
- GLUT → `#include <glui.h>`: libreria C++ di oggetti per la composizione di un'interfaccia GUI (bottoni, check box, spin button, ...).

NB: `#include <GL/glut.h>` comprende anche `gl.h` e `glu.h`