

# Costruzione di Interfacce

## Lezione 6

### Esercitazione Trasformazioni

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

## Skeleton SDL

```
#ifdef WIN32
#define WIN32_LEAN_AND_MEAN
#include <windows.h>
#endif

#include <GL/gl.h>
#include <GL/glu.h>
#include <stdlib.h>
#include <SDL.h>

void DrawGLScene()
{
    glClear(GL_COLOR_BUFFER_BIT);
    SDL_GL_SwapBuffers();
}

void myGLReshapeFunc(GLsizei w, GLsizei h)
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    float ratio=(float)w/(float)h;
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_MODELVIEW);
}
-
```

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

2

## Skeleton SDL

```
int main(int argc, char **argv)
{
    if ( SDL_Init(SDL_INIT_VIDEO) < 0 ) {
        fprintf(stderr, "Unable to initialize SDL: %s\n", SDL_GetError());
        exit(1);
    }

    if ( SDL_SetVideoMode(640, 480, 0, SDL_OPENGL | SDL_RESIZABLE) == NULL ) {
        fprintf(stderr, "Unable to create OpenGL screen: %s\n", SDL_GetError());
        SDL_Quit();
        exit(1);
    }

    /* Loop, drawing and checking events */
    int done = 0;
    myGLReshapeFunc(640,480);
    while ( ! done )
    {
        SDL_Event event;
        SDL_WaitEvent(&event);
        switch(event.type)
        {
            case SDL_VIDEOEXPOSE : DrawGLScene(); break;
            case SDL_QUIT : done = 1; break;
            case SDL_KEYDOWN : if ( event.key.keysym.sym == SDLK_ESCAPE ) done = 1; break;
            case SDL_VIDEORESIZE :
                SDL_SetVideoMode(event.resize.w, event.resize.h, 0, SDL_OPENGL | SDL_RESIZABLE);
                myGLReshapeFunc(event.resize.w, event.resize.h);
                break;
        }
    }

    SDL_Quit();
    return 1;
}
10 Ott 2003
```

Costruzione di Interfacce - Paolo Cignoni

3

## Obiettivo

- ❖ Costruire una serie di anelli di moebius fatti di blocchettini:
1. Costruire un blocchettino e definire il suo sistema di riferimento
  2. Usarlo per costruire un anello nello sistema di riferimento dell'anello
  3. Piazzare i vari anelli nel sistema di riferimento del mondo

10 Ott 2003

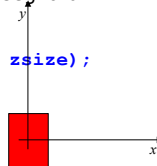
Costruzione di Interfacce - Paolo Cignoni

4

## Aggiungiamo un cubo

- ❖ Sistema di riferimento del blocchetto:
  - ❖ Centrato nell'origine.
  - ❖ Scriviamo una funzione che disegna un blocchetto in wireframe

`DrawBlock(float xsize, ysize, zsize);`



10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

5

## DrawBlock

- ❖ Disegno i 12 edge di un box3d

```
void DrawBlock(float xsz, float ysz, float zsz)
{
    glBegin(GL_LINES);
    glVertex3f(-xsz, ysz, zsz); glVertex3f( xsz, ysz, zsz);
    glVertex3f(-xsz, -ysz, zsz); glVertex3f( xsz, -ysz, zsz);
    glVertex3f( xsz, ysz, zsz); glVertex3f( xsz, -ysz, zsz);
    glVertex3f(-xsz, ysz, zsz); glVertex3f(-xsz, -ysz, zsz);

    glVertex3f( xsz, ysz, -zsz); glVertex3f( xsz, ysz, zsz);
    glVertex3f(-xsz, ysz, -zsz); glVertex3f(-xsz, ysz, zsz);
    glVertex3f( xsz, -ysz, -zsz); glVertex3f( xsz, -ysz, zsz);
    glVertex3f(-xsz, -ysz, -zsz); glVertex3f(-xsz, -ysz, zsz);

    glVertex3f(-xsz, ysz, -zsz); glVertex3f( xsz, ysz, -zsz);
    glVertex3f(-xsz, -ysz, -zsz); glVertex3f( xsz, -ysz, -zsz);
    glVertex3f( xsz, ysz, -zsz); glVertex3f( xsz, -ysz, -zsz);
    glVertex3f(-xsz, ysz, -zsz); glVertex3f(-xsz, -ysz, -zsz);
    glEnd();
}
10 Ott 2003
```

Costruzione di Interfacce - Paolo Cignoni

6

## Testiamo la nostra funzione

### **Mai fare tutto assieme**

- ❖ Cerchiamo di visualizzare subito il nostro blocchetto
- ❖ In questo caso il nostro sistema di riferimento di mondo e' quello del blocchetto stesso.
- ❖ Quindi basta settare la trasf di projection e la trasf da world a camera:
  - ❖ gluPerspective e glulookat

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

7

## Visualizzare il blocchetto

- ❖ Matrice di proiezione
  - ❖ definisce il volume di vista in spazio di camera, ha bisogno:
    - ❖ dell'aspect ratio della finestra (tutto il volume di vista viene mappato nella finestra quindi deve avere le stesse proporzioni)
    - ❖ Range di distanze entro cui visualizzo gli oggetti
- ❖ Lo setto nella Reshape

```
void myGLReshapeFunc(GLsizei w, GLsizei h)
{
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    float ratio=(float)w/(float)h;
    gluPerspective(45,ratio,3,10);
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_MODELVIEW);
}
```

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

8

## Visualizzo il blocchetto

- ❖ Matrice di modellazione
  - ❖ Passa da spazio di mondo a spazio di camera. Ha bisogno di sapere, in spazio di mondo:
    - ❖ Da dove, in che direzione e con che orientamento
    - ❖ Si sta guardando la nostra scena.
- ❖ La settiamo nella drawScene e si assume di guardare nell'origine

```
void DrawGLScene()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(2,2,5,0,0,0,0,1,0);
    DrawBlock(1,1,1);
    DrawBlock(1.5,.5,-.5);
    SDL_GL_SwapBuffers();
}
```

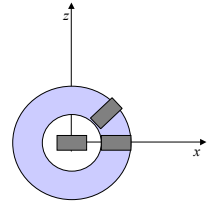
10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

9

## Sistema di riferimento dell'anello

- ❖ Centrato sull'origine
- ❖ Sul piano xz
- ❖ Si costruisce con una serie di blocchetti che vanno rototraslati prima di essere disegnati



10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

10

## Ruotiamo il cubo

- ❖ Rototraslare il cubo significa
  - ❖ Risposta semplice
    - ❖ faccio una glRotate e una glTranslate prima di disegnare il cubo
  - ❖ Risposta vera
    - ❖ Cambio il sistema di riferimento della camera rispetto al sistema di riferimento del cubo.
  - ❖ Nota:
    - ❖ glRotate e glTranslate **modificano** la matrice corrente, quindi uso glPushMatrix e glPopMatrix per far trovare ad ogni cubetto rototraslatato sempre lo stesso stato iniziale

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

11

## Composizione di trasformazioni

- ❖ Fare
  - ❖ glRotate
  - ❖ glTranslate
- ❖ E' ben diverso di
  - ❖ glTranslate
  - ❖ glRotate
- ❖ Poiché le matrici si compongono per post moltiplicazione l'ordine sembra quello inverso...
- ❖ Trucco: per visualizzare il risultato partire dall'ultima trasformazione e applicarle all'indietro...

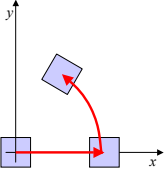
10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

12

## Rototraslazione del cubo

```
glRotatef(i*20,0,0,1);  
glTranslatef(1,0,0);
```



10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

13

## Proviamo a disegnare un anello

- ❖ Scriviamo una funzione abbastanza generica per fare un anello di blocchetti

```
void DrawRing(int step, float r)  
{  
    float angle=360.0f/step;  
    for(int i=0;i<step;++i)  
    {  
        glPushMatrix();  
        glRotatef(i*angle,0,1,0);  
        glTranslatef(r,0,0);  
        DrawBlock(1.5,.5,.5);  
        glPopMatrix();  
    }  
}
```

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

14

## Proviamo

- ❖ Cambiamo anche l'inquadratura per prendere tutto l'anello.
  - ❖ Nota state attenti al range della gluPerspective!

```
void DrawGLScene()  
{  
    glClearColor(GL_COLOR_BUFFER_BIT);  
    glLoadIdentity();  
    gluLookAt(2,5,10,0,0,0,0,1,0);  
  
    DrawRing(18,3);  
  
    SDL_GL_SwapBuffers();  
}
```

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

15

## Esercizio 2

- ❖ Trasformare la striscia in un anello di Moebius.
  - ❖ hint: basta solo una rotazione in piu'
- ❖ Basta ruotare il cubo prima di iniziare la rototraslazione che lo mette nella propria posizione all'interno dell'anello

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

16

## Moebius Ring

```
void DrawRing(int step, float r, int twist)  
{  
    float angle=360.0f/step;  
    float angletwist=(180.0f*twist)/step;  
    for(int i=0;i<step;++i)  
    {  
        glPushMatrix();  
        glRotatef(i*angle,0,1,0);  
        glTranslatef(r,0,0);  
        glRotatef(i*angletwist,0,0,1);  
        DrawBlock(1.5,.5,.5);  
        glPopMatrix();  
    }  
}
```

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

17

## Animazione

- ❖ Come si fa animazione in un'applicazione event driven?
- ❖ Vietato disegnare continuamente un frame dopo l'altro senza mai uscire dalla callback di display!!
- ❖ Si deve disegnare un frame e guardare se non c'è altro di importante da fare prima.

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

18

## Idle Processing

- ❖ Tutte le idee che non esplicitano il loop degli eventi, mettono a disposizione una callback (evento) che scatta scattare quando non hanno nulla da fare (idle processing)
- ❖ Le varie app vengono così chiamate ogni qual volta il so non ha altro da fare.
- ❖ **NOTA:** Non si ha alcuna garanzia di QUANDO si viene chiamati...

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

19

## Animating

- ❖ Tutti i calcoli relativi alle posizioni, orientamento durante le animazioni DEVONO essere temporizzati.
- ❖ Per rendere il cambio di fotogramma stabile, si usa la tecnica del double buffering;
- ❖ Si disegna la scena in un buffer nascosto e si mostra solo quando abbiamo finito.

```
SDL_GL_SwapBuffers();
```

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

20

## Animazione

- ❖ **Temporizzazione:**
  - ❖ **Fondamentale.**
  - ❖ Meccanismo base:
    - ❖ Esprimo tutti i movimenti che in unita assolute (e.g. m/sec, gradi/sec ecc)
  - ❖ Per ogni frame
    - ❖ Guardo quanto tempo e' passato
    - ❖ Aggiorno le variabili di stato (e.g. posizione e orinetamento) in funzione di quanto tempo e' passato.
  - ❖ Usare i timer e' un po' meno affidabile.

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

21

## Animazione in SDL

- ❖ Cambiamo la funzione che preleva gli eventi in una non bloccante.
- ❖ Aggiungiamo nella funzione di disegno delle variabili statiche per tener traccia del tempo che passa.

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

22

## Moebius Ring

```
while (! done)
{
    SDL_Event event;
    ret=SDL_PollEvent(&event);
    if(ret)
        switch(event.type)
        {
            case SDL_QUIT : done = 1;
                          break;
            case SDL_KEYDOWN : if ( event.key.keysym.sym == SDLK_ESCAPE )
                              done = 1;
                              break;
            case SDL_VIDEORESIZE :
                          SDL_SetVideoMode(event.resize.w,event.resize.h, 0,
                          SDL_OPENGL | SDL_RESIZABLE);
                          myGLReshapeFunc(event.resize.w,event.resize.h);
                          break;
        }
    else DrawGLScene();
}
```

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

23

## Moebius Ring

```
void DrawGLScene()
{
    static LastT=clock();
    static float CurAngle=0;

    int CurT=clock();
    float DeltaT=(CurT-LastT)/1000.0; // elapsed time in secs.
    if(DeltaT < 0.01) return;
    LastT=CurT;
    float AngularSpeed = 90; // in degree/sec

    CurAngle+=AngularSpeed*DeltaT;
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    gluLookAt(2,5,12,0,0,0,1,0);

    glRotatf(CurAngle,0,1,0);
    DrawRing(18,5,1);
    SDL_GL_SwapBuffers();
}
```

10 Ott 2003

Costruzione di Interfacce - Paolo Cignoni

24

## Sempre piu' difficile

- ❖ Aggiungiamo nell'ordine
  - ❖ L'anello che e' formato da una doppia striscia
  - ❖ Una pallina che rotola sull'anello
  - ❖ L'anello che ruota su se stesso

## Esercizio

- ❖ Aggiungete una serie di anelli di moebius e spostate la camera in modo che si vedano tutti.
- ❖ Tutti gli anelli devono ruotare intorno al proprio asse