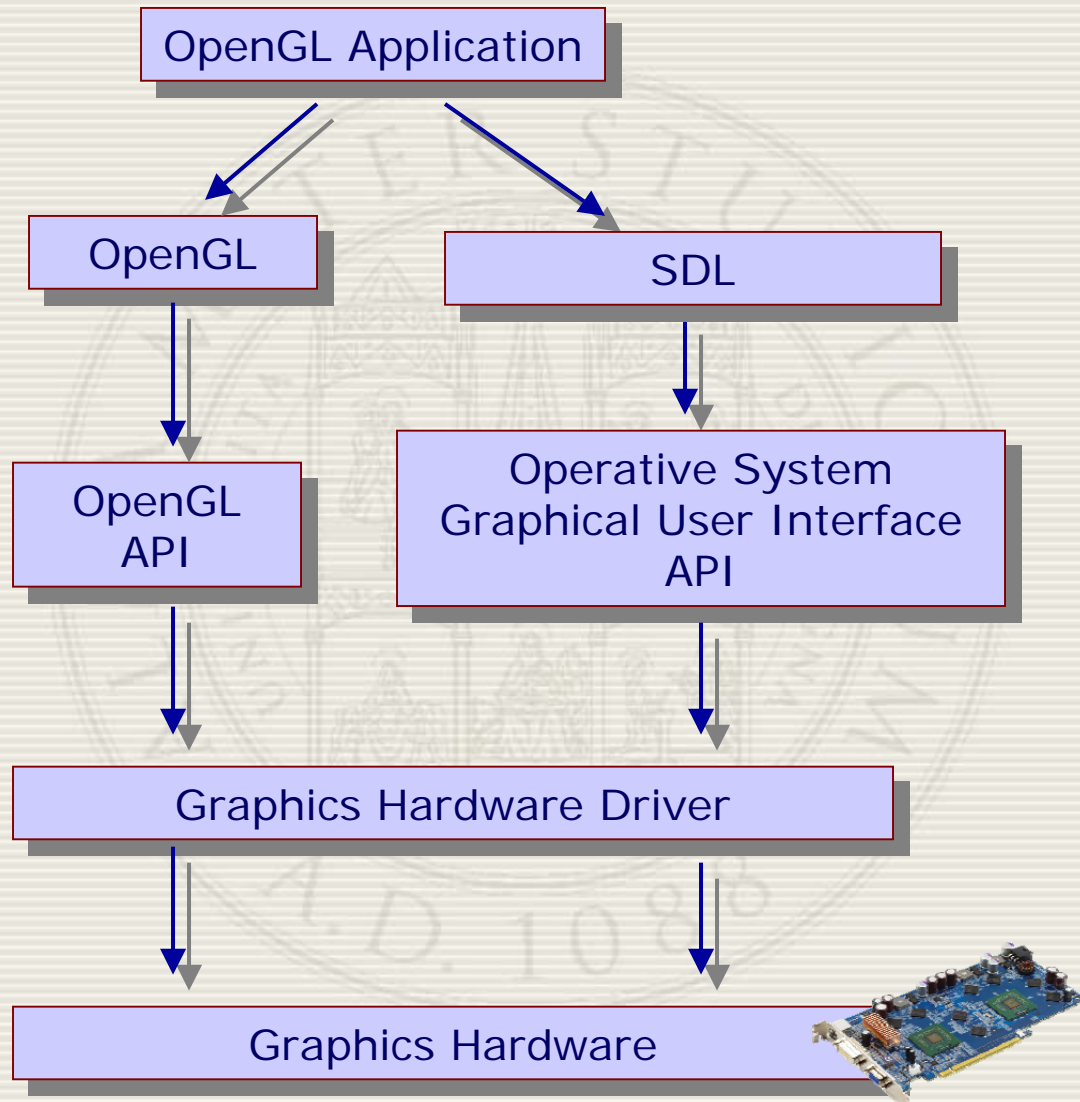


Libreria Grafica OpenGL + SDL2



Struttura Applicazione OpenGL+SDL



SDL: Inizio e Fine

Dopo aver inizializzato SDL, è necessario creare una `SDL_Window` per OpenGL. Questo viene fatto mediante

```
SDL_CreateWindow("Title", x, y, w, h,  
                SDL_WINDOW_OPENGL);
```

L'esempio seguente crea un'area di disegno e ritorna un puntatore a `SDL_Window`;

Esempio:

```
SDL_Window *p_win;  
.....  
p_win = SDL_CreateWindow("Title",0,0,640,480,SDL_WINDOW_OPENGL);  
if ( p_win == NULL) {  
    fprintf(stderr, "Error setting window: %s\n", SDL_GetError());  
    /* Fail! */  
}
```

SDL con OpenGL

La libreria SDL dà la possibilità di creare e usare il contesto OpenGL su differenti piattaforme (Linux/X11, Win32, MacOS, iOS, Android).

Ciò significa poter utilizzare audio, gestione eventi, thread e gestione tempo nelle applicazioni OpenGL, come fanno anche le GLUT.

Inizializzazione delle SDL

Rispetto ad inizializzare le SDL senza OpenGL ci sono tre differenze:

- Si deve passare `SDL_WINDOW_OPENGL` alla `SDL_SetCreateWindow()`
- Si devono specificare gli attributi `GL` (come dimensione depthbuffer, dimensione framebuffer, ecc.) usando la `SDL_GL_SetAttribute()`
- Se si vuole usare il double buffer lo si deve specificare come un attributo `GL`.

SDL con OpenGL: esempio

```
/* First, initialize SDL's video subsystem */
if( SDL_Init( SDL_INIT_VIDEO ) < 0 ) {
    fprintf( stderr, "Video initialization failed: %s\n", SDL_GetError( ) );
    quit_tutorial( 1 );
}

/*Now, we want to setup our requested window attributes for our OpenGL
window*/

SDL_GL_SetAttribute( SDL_GL_RED_SIZE, 5 );
SDL_GL_SetAttribute( SDL_GL_GREEN_SIZE, 5 );
SDL_GL_SetAttribute( SDL_GL_BLUE_SIZE, 5 );
SDL_GL_SetAttribute( SDL_GL_DEPTH_SIZE, 16 );
SDL_GL_SetAttribute( SDL_GL_DOUBLEBUFFER, 24 );

/*We want to request that SDL provide us with an OpenGL window*/
flags = SDL_WINDOW_OPENGL | SDL_FULLSCREEN;
if( SDL_SetVideoMode( width, height, bpp, flags ) == 0 ) {
    fprintf( stderr, "Video mode set failed: %s\n", SDL_GetError( ) );
    quit_tutorial( 1 );
}
```

SDL con OpenGL: disegno

Attenzione: se si sta usando il doublebuffer, si deve usare

`SDL_GL_SwapBuffers()`

per scambiare i buffer e aggiornare il display.

Concludendo:

a parte l'inizializzazione, usare le OpenGL con le SDL è la stessa cosa di usare le OpenGL con un qualunque altro API come per esempio GLUT.

Semplice Applicazione SDL+OpenGL: header

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

#include <GL/gl.h>
#include <GL/glu.h>
#include <stdlib.h>
#include <SDL2/SDL.h>
#include <SDL2/SDL_image.h>
#include <SDL2/SDL_ttf.h>
```

Semplice Applicazione SDL+OpenGL: main

```
int main(int argc, char **argv) {
    SDL_Event event;
    SDL_Init(SDL_INIT_VIDEO);
    SDL_SetVideoMode(640, 480, 0, SDL_OPENGL);

    int done = 0;
    while ( ! done ) { /* Loop, drawing and checking events */
        if (SDL_PollEvent(&event) {
            switch(event.type){
                case SDL_WINDOWEVENT:
                    if(e.window.event==SDL_WINDOW_EVENT_EXPOSE)
                        myDrawGLScene();
                    break;
                case SDL_QUIT : done = 1;  break ;
                case SDL_KEYDOWN :
                    if ( event.key.keysym.sym == SDLK_ESCAPE )
                        done = 1;
                    break;
            }
            else myTimer();
        }
        ...
    }
}
```


Semplice Applicazione SDL+OpenGL: timer per ridisegnare

```
void myTimer()
{
    Uint32 timeNow;

    timeNow=SDL_GetTicks();
    /*si vuol disegnare una volta ogni tot secondi*/
    if (timeNow-timeLast > elapsed)
    {
        redraw( );
        timelast=timeNow;
    }
    else
        ....
}
```

```
void myTimer()
{
    SDL_Delay(elapsed);
    redraw( );
}
```

Semplice Applicazione SDL+OpenGL: la parte che disegna

```
void redraw(){  
    /*ci automandiamo un messaggio che ci fara'  
    ridisegnare la finestra*/  
    SDL_Event e;  
    e.type=SDL_WINDOWEVENT;  
    e.window.event=SDL_WINDOWEVENT_EXPOSE;  
    SDL_PushEvent(&e);  
}
```

```
void myDrawGLScene()  
{  
    glClear(GL_COLOR_BUFFER_BIT);  
  
    /* disegna tutto facendo chiamate alle GL*/  
  
    glFinish(); /* aspetta che sia tutto finito */  
    SDL_GL_SwapBuffers();  
}
```

Esempio

Esempio: sdlsimple_orig.c
(dir opengl/gl_start/)

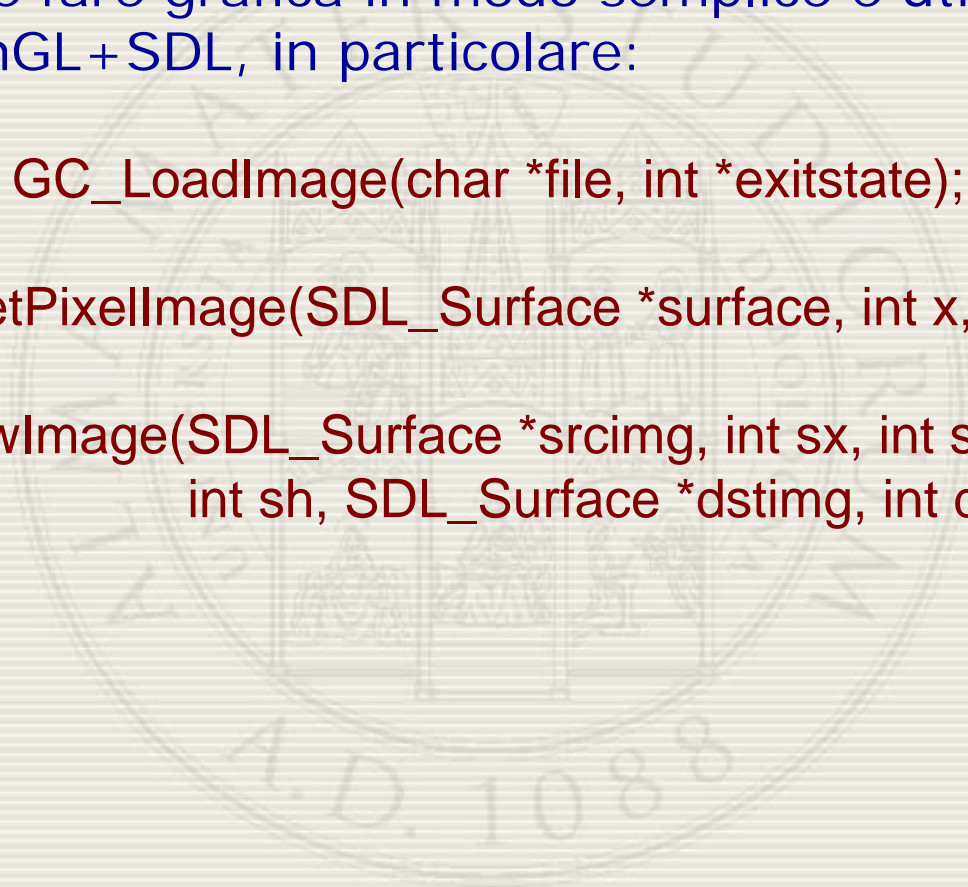
Libreria GCGraLib2

La piccola libreria GCGraLib2 vista per interfacciarsi con le librerie SDL e fare grafica in modo semplice è utile anche usando OpenGL+SDL, in particolare:

```
SDL_Surface* GC_LoadImage(char *file, int *exitstate);
```

```
Uint32 GC_GetPixellImage(SDL_Surface *surface, int x, int y);
```

```
void GC_DrawImage(SDL_Surface *srcimg, int sx, int sy, int sw,  
                  int sh, SDL_Surface *dstimg, int dx, int dy);
```



Texture

Vediamo come caricare una immagine nella texture memory per farla processare come OpenGL texture image:

```
int LoadTexture(int textbind,char *filename){
int res;
SDL_Surface *s;
s = GCLoadImage(filename,res);
if (!s || !res)
return 0;
glBindTexture(GL_TEXTURE_2D, textbind);
gluBuild2DMipmaps(GL_TEXTURE_2D, GL_RGB, s->w, s->h, GL_RGB,
GL_UNSIGNED_BYTE, s->pixels);
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR );
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
GL_LINEAR_MIPMAP_LINEAR );
return 1;
}
```

Testo

Vediamo una funzione che crea una immagine testo in SDL e la processa come OpenGL texture image:

```
void SDL_GL_DrawText(TTF_Font *font,char fgR, char fgG, char fgB,  
    char fgA,char bgR, char bgG, char bgB, char bgA, char *text, int x,  
    int y,enum textquality quality)
```

dove:

```
enum textquality{ solid, shaded, blended};
```

Esempio: perps_sdl_gl.c
(dir SDLprg2_gl/)