

Tutorial Allegro!

(le basi della libreria allegro utilizzata sotto Windows con MinGW)

**Versione 0.17 (molto incompleta)
del 22/maggio/2007**

> www.valarsoft.com/ramuso
> www.valarsoft.com

Capitolo 1:

Installazione e primi passi

Cos'è la libreria Allegro?

Iniziamo col dire che una libreria (in generale) è una raccolta di funzioni (o classi), già pronte, che espandono le capacità di base del nostro linguaggio (nel caso specifico c++). La libreria allegro è una libreria specifica per la programmazione di videogiochi 2D, più nel dettaglio la libreria allegro gestisce la grafica, il sonoro e l'input output in maniera estremamente semplice. Per usare la libreria allegro occorre installarla assieme ad un compilatore C/C++. Useremo, nel nostro tutorial il porting per windows del linguaggio c/c++ GNU usato sotto linux, il MinGW.

Installazione di MinGW

In questo tutorial si userà la versione 3.4.2 del compilatore MinGW (vedi la sezione "dove scaricare il software" per procurarlo). Il sistema operativo usato si suppone sia WindowsXP.

Supponiamo di avere il file "MinGW-3.4.2.exe", per installare MinGW basta un doppio click sul file ed il gioco è fatto.

E' consigliabile installare il minGW nella cartella **C:\mingw** (in questo tutorial si supporrà di usare sempre la suddetta cartella).

Dopo l'installazione del MinGW, per un suo corretto utilizzo bisogna settare alcune variabili d'ambiente, bisogna cioè far capire al nostro sistema operativo (windows) dove si trovano gli eseguibili ed in quale cartella abbiamo installato il MinGW.

Per settare le variabili d'ambiente bisogna andare su:

Risorse del computer->(tasto destro del mouse)Proprietà->Avanzate->Variabili D'ambiente

Dalla finestra variabili d'ambiente andare nel riquadro **variabili di sistema** nel quale si deve accodare al valore della variabile "path" la stringa "**;c:\mingw\bin**", per farlo bisogna selezionare, dalla lista, la variabile **path** e cliccare su **Modifica**.

Cliccare quindi su **Nuovo** ed inserire in "nome variabile" la stringa **MINGDIR** ed in "valore variabile" la stringa **c:\mingw**. Cicare quindi su **Ok**.

Per testare il corretto funzionamento del compilatore aprire una finestra di Ms Dos (Prompt di Ms Dos), e scrivere in essa il comando **gcc**, se il prompt vi risponde con "**gcc: no input file**" allora l'installazione di MinGW è andata a buon fine.

Installazione di Allegro

In questo tutorial si userà la versione 4.2.0 della libreria allegro (vedi la sezione “dove scaricare il software” per procurarlo). Il sistema operativo usato si suppone sia WindowsXP.

Supponiamo di avere il file “allegro-mingw-4.2.0.zip”, per installare la libreria allegro basta scompattare l’archivio .zip ed inserire tutto il contenuto (le tre cartelle ed il file reame.txt) così ottenuto nella cartella **c:\mingw** (sovrascrivendo eventuali cartelle o file omonimi). Per essere sicuri che tutto funzioni bene occorre anche copiare il file **alleg42.dll** (presente nella cartella bin di allegro) in **c:\windows\system32**.

Dove scaricare il software

Il sito ufficiale del compilatore MinGW è www.mingw.org, il sito ufficiale della libreria allegro è www.allegro.cc, tutto il materiale può essere reperito in questi siti.

Nel nostro tutorial si userà comunque le versione 3.4.2 di minGW e la versione 4.2.0 della libreria Allegro, se non volete faticare a cercarli all’interno del sito ufficiale di allegro allora scaricateli dal sito www.valarsoft.com/ramuso secondo le seguenti indicazioni:

- **Il MinGW può essere scaricato dal sito:**
www.valarsoft.com/ramuso (sezione download->Programmazione->C/C++)
- **La libreria allegro (versione 4.2) può essere scaricata dal sito:**
www.valarsoft.com/ramuso (sezione download->Programmazione->C/C++)
- **Tutti i listati delle varie lezioni del tutorial sono scaricabili dal sito:**
www.valarsoft.com/ramuso (sezione download->Programmazione->Tutorial)

Prima lezione (Aprire uno schermo con allegro)

Il listato 1 - "es01.cpp" (manca ancora la spiegazione) va compilato scrivendo sotto ms dos la riga di comando:

g++ es01.cpp -oes01.exe -lalleg -mwindows

l'opzione "-mwindows" può anche essere omessa.

```
// Tutorial Allegro - www.valarsoft.com/ramuso
//
// Listato 1 – primi passi
// Esempio di come aprire uno schermo tramite allegro e stampare
// in esso la frase "Hello World!".
// Si userà inoltre la funzione keypress (che aspetta la pressione di un tasto)
// per uscire dal programma.

#include <allegro.h> // In ogni programma allegro si deve includere il file allegro.h

int main()
{
    allegro_init(); // Inizializza la libreria allegro, funzione obbligata che setta correttamente
                   // tutto ciò che serve ad allegro per funzionare

    install_keyboard(); // inizializza le funzioni di allegro relative alla tastiera
                       // da mettere quando occorre usare la tastiera

    if(set_gfx_mode(GFX_AUTODETECT,640,480,0,0)) return 1;
    // Cambia la modalità video in 640x480 pixel
    // Nel caso non riesco ad aprire lo schermo il programma esce con un codice di errore

    textprintf_ex(screen,font,0,0,makecol32(255,255,255),-1,"Hello World!");
    // Stampa a video la scritta "Hello World!"
    // screen è la variabile globale che rappresenta lo schermo dove stampare la frase
    // font rapresenta il carattere da usare (in questo caso quello di default)
    // 0,0 sono le coordinarte (in pixel) dello schermo dove stampare la frase (0,0 si trova in alto a sinistra)
    // makecol32(255,255,255) dice alla funzione textprintf_ex di stampare la frase di colore bianco

    readkey(); // Aspetta la pressione di un tasto

    return 0; // Esce dal programma senza segnalare errori
} END_OF_MAIN() // Questa macro va richiamata alla chiusura della parentesi della funzione main
                // ed è specifica di allegro.
```

Ma vediamo come è gestito lo schermo da allegro.
Esso è una matrice di pixel

Seconda lezione (Iniziamo con la grafica)

Il listato 2 - "es02.cpp" (manca ancora la spiegazione) va compilato scrivendo sotto ms dos la riga di comando:

g++ es02.cpp -oes02.exe -lalleg -mwindows

l'opzione "-mwindows" può anche essere omessa.

```
// Tutorial Allegro - www.valarsoft.com/ramuso
//
// Listato 2 – primi passi con la grafica e la tastiera
// Esempio di una pallina che rimbalza sullo schermo
// Si userà inoltre l'array key[] per gestire l'uscita dal programma.

#include <allegro.h> // In ogni programma allegro si deve includere il file allegro.h

int main()
{
    int x,y; // Coordinate della pallina
    int dx,dy; // Direzione della pallina lungo x e y

    allegro_init(); // Inizializza la libreria allegro, funzione obbligata che setta correttamente
                    // tutto ciò che serve ad allegro per funzionare

    install_keyboard(); // inizializza le funzioni di allegro relative alla tastiera
                       // da mettere quando occorre la tastiera

    set_color_depth(32); // Setta lo schermo a 32 bit di colore per pixel
                        // (modalità truecolor a 16 milioni di colori)

    if(set_gfx_mode(GFX_AUTODETECT,640,480,0,0)) return 1;
    // Cambia la modalità video in 640x480 pixel
    // Nel caso non riesco ad aprire lo schermo il programma esce con un codice di errore
    // dopo aver chiamato la funzione set_gfx_mode() allegro crea le seguenti variabili:
    // screen che non è altro che una matrice di pixel (punti colorati) che rappresenta lo schermo
    // (è una variabile di tipo BITMAP)
    // SCREEN_W che è un'intero che rappresenta la larghezza del video
    // SCREEN_H che è un'intero che rappresenta l'altezza del video

    x=100; y=100; // Coordinate iniziali della pallina
    dx=2; dy=2; // Direzione iniziale della pallina
                // (si muove in diagonale dall'alto verso il basso e da sinistra a destra)

    // Ciclo principale del programma
    while(!key[KEY_ESC]) // Finche non premo il tasto Esc il programma continua a girare
    {
        // L'array key[] contiene una "casella" per ogni tasto della tastiera, quando premo un tasto
        // il relativo valore al tasto premuto diventa true (vero), quando il tasto viene rilasciato
        // il relativo valore al tasto premuto diventa false (falso)

        clear_to_color(screen,makecol32(0,0,0)); // Cancella lo schermo colorandolo di nero
        // screen è la variabile globale che rappresenta lo schermo da cancellare
        // makecol32(0,0,0) dice alla funzione clear_to_color
        // di cancellare lo schermo colorandolo tutto di nero
    }
}
```

```

circlefill(screen,x,y,5,makecol32(255,0,0)); // Stampa a video la pallina (cerchio pieno)
// screen è la variabile globale che rappresenta lo schermo dove disegnare la pallina
// x e y sono le coordinate dove stampare il cerchio pieno che rappresenta la pallina
// (ricordiamo che 0,0 si trova in alto a sinistra rispetto allo schermo)
// 5 rappresenta il raggio del cerchio pieno (la grandezza della pallina)
// makecol32(255,0,0) dice alla funzione circlefill di stampare un cerchio pieno di colore rosso

vsync(); // Attende la sincronia verticale prima di ridisegnare lo schermo
// serve per fare ridisegnare lo schermo ad una velocità adeguata all'occhio umano
// se si prova a togliere questa funzione, la pallina sarà troppo veloce

x+=dx; y+=dy; // Muove la pallina
if(x<0 || x>SCREEN_W) dx=-dx; // Controlla che la pallina abbia urtato uno dei lati dello schermo
if(y<0 || y>SCREEN_H) dy=-dy; // se la pallina urta uno dei lati, si inverte la direzione
// di moto della pallina
}

return 0; // Esce dal programma senza segnalare errori
} END_OF_MAIN() // Questa macro va richiamata alla chiusura della parentesi della funzione main
// ed è specifica di allegro.

```

Questo esempio funziona perfettamente, ma presenta un difetto di fondo, se aumentiamo il numero di oggetti da disegnare sullo schermo, si verifica uno strano lampeggiamento dello schermo. Per evitare questo si dovrà ricorrere alla tecnica del doppio buffer video.

Capitolo 2: Doppio buffer

Terza lezione (Doppio buffer video)

Il listato 3 - "es03.cpp" (manca ancora la spiegazione) va compilato scrivendo sotto ms dos la riga di comando:

g++ es03.cpp -oes03.exe -lalleg -mwindows

l'opzione "-mwindows" può anche essere omessa.

```
// Tutorial Allegro - www.valarsoft.com/ramuso
//
// Listato 3 – doppio buffer video
// Esempio di una pallina che rimbalza sullo schermo
// con doppio buffer per evitare "lampeggiamenti"

#include <allegro.h>

int main()
{
    int x,y; // Coordinate della pallina
    int dx,dy; // Direzione della pallina lungo x e y
    BITMAP *schermo_nascosto; // Questa variabile di tipo puntatore a BITMAP
                                // rappresenta uno schermo nascosto della stessa dimensioni
                                // dello schermo visibile

    allegro_init();
    install_keyboard();

    set_color_depth(32);
    if(set_gfx_mode(GFX_AUTODETECT,640,480,0,0)) return 1;
    // Cambia la modalità video in 640x480 pixel
    // dopo aver chiamato la funzione set_gfx_mode() allegro crea le seguenti variabili:
    // screen è lo schermo visibile
    // SCREEN_W è un'intero che rappresenta la larghezza dello schermo visibile
    // SCREEN_H è un'intero che rappresenta l'altezza dello schermo visibile

    // Schermo nascosto
    schermo_nascosto=create_bitmap_ex(32,SCREEN_W,SCREEN_H); // Crea lo schermo nascosto
    // con lo stesso numero di colori per pixel dello schermo visibile (32)
    // e delle stesse dimensioni dello schermo visibile (SCREEN_W,SCREEN_H)

    x=100; y=100; // Coordinate iniziali della pallina
    dx=3; dy=3; // Direzione iniziale della pallina (ora è un pò più veloce)

    // Ciclo principale del programma
    while(!key[KEY_ESC]) // Finche non premo il tasto Esc il programma continua a girare
    {
        // Agisco adesso solo sullo schermo nascosto
        clear_to_color(schermo_nascosto,makecol32(0,0,0)); // Cancella lo schermo nascosto
        // schermo_nascosto è la variabile che rappresenta lo schermo nascosto
        // makecol32(0,0,0) dice alla funzione clear_to_color di cancellare lo schermo
        // colorandolo tutto di nero
    }
}
```

```

circlefill(schermo_nascosto,x,y,5,makecol32(255,255,0)); // Stampa nello schermo nascosto la pallina
// schermo_nascosto è la variabile globale che rappresenta lo schermo dove disegnare la pallina
// x e y sono le coordinate dove stampare il cerchio pieno che rappresenta la pallina
// 5 rappresenta il raggio del cerchio pieno (la grandezza della pallina)
// makecol32(255,255,0) dice alla funzione circlefill di stampare un cerchio pieno di colore giallo

// Stampa il contenuto dello schermo_nascosto in quello vero che è visibile
blit(schermo_nascosto,screen,0,0,0,0,SCREEN_W,SCREEN_H);
// blit fa questo: prende lo schermo_nascosto e lo sovrascrive a screen (che è lo schermo visibile)
// gli altri parametri servono per copiare solo una porzione dello schermo (per adesso sono tutti 0)
// infine si devono indicare le dimensioni dello schermo (SCREEN_W,SCREEN_H).
// Così facendo si evitano sgradevoli "lampeggiamenti" dello schermo
// e l'animazione risulta più fluida

vsync(); // Attende la sincronia verticale prima di ridisegnare lo schermo

x+=dx; y+=dy; // Muove la pallina
if(x<0 || x>SCREEN_W) dx=-dx; // Controlla che la pallina abbia urtato uno dei lati dello schermo
if(y<0 || y>SCREEN_H) dy=-dy; // se la pallina urta uno dei lati, si
                                // inverte la direzione di moto della pallina
}

return 0;
} END_OF_MAIN()

```