

Corso di Informatica

Modulo T2

B2 – Animazione

M. Malatesta B2-Animazione-18

1
07/03/2012

Prerequisiti

- Tecnica della programmazione elementare
- Elementi di geometria analitica
- Cenni di programmazione concorrente

M. Malatesta B2-Animazione-18

2
07/03/2012

Introduzione

In questa Unità vediamo come nasce l'animazione delle immagini, tecnica di cui studiamo anche la realizzazione informatica attraverso un meccanismo di *temporizzazione*.

Accenniamo anche al principale problema connesso con l'animazione che è lo **sfarfallio**, un fenomeno fastidioso che compromette la visualizzazione del movimento e si descrivono alcune tecniche per diminuirlo o eliminarlo.

M. Malatesta B2-Animazione-18

3
07/03/2012

Cos'è l'animazione

Per **animazione** si intende (in ambito **televisivo** e **cinematografico**), tutto ciò che concerne audiovisivi prodotti mediante disegni animati o immagini *in movimento apparente*.

M. Malatesta B2-Animazione-18

4
07/03/2012

Come si ottiene l'animazione

L'illusione dell'animazione si produce mediante una successione di pose, *più rapida del tempo di percezione dell'occhio umano* (0.1 secondi, per cui la frequenza è di 10 **Hz**, ovvero 10 **fps**, *fotogrammi al secondo*).

Le pose possono raffigurare attori in recita (**cinematografia**), disegni o fotografie di pupazzi (**cartoni animati**, **effetti speciali**).

Tecniche rudimentali furono usate da Newton e Leonardo, fino a giungere alla realizzazione di pellicole cinematografiche.

M. Malatesta B2-Animazione-18

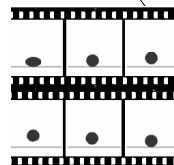
5
07/03/2012

Come si ottiene l'animazione

Per ottenere l'effetto di animazione il valore minimo accettabile di **fps** era inizialmente 12, ma poiché dava immagini con effetto lampeggiante, il valore attualmente considerato 16.

Valori standard di *fps*:

- 24 per il cinema;
- 25 per i sistemi televisivi europei PAL;
- 4 per le moviole.



M. Malatesta B2-Animazione-18

6
07/03/2012

Rappresentare l'animazione

Le tecniche per rappresentare l'animazione sono:

- **metodo elementare**, consiste nel pensare di codificare separatamente ogni fotogramma come immagine fissa, ma lo spazio di memoria richiesto sarebbe enorme (650 MB, un intero CD per un minuto di proiezione, ...)
- **metodi di compressione**, che definiscono standard per la rappresentazione digitale di audio, video e altre tipologie di contenuti multimediali, in modo da economizzare la memoria necessaria:
 - **MPEG** (*Moving Picture Experts Group*)
 - **AVI** (© Microsoft) (*Audio Video Interleave*)

M. Malatesta B2-Animazione-18

7
07/03/2012

Applicazioni animate

Per realizzare un'applicazione con disegni animati sono necessari alcuni elementi:

- una serie di disegni (fotogrammi) già predisposta, ad esempio in un array;
- un oggetto **timer** che, a scadenze di tempo prefissate, faccia transitare il successivo fotogramma, emettendo un apposito **segnale di sincronismo**;
- un linguaggio che preveda una modalità grafica in cui visualizzare immagini.

M. Malatesta B2-Animazione-18

8
07/03/2012

Applicazioni animate

Algoritmo Animazione

```
Intero delay, // tempo di transizione
    prec=0; // ultima immagine visualizzata
Immagine Im[5] = {i0, i1, i2, i3, i4}; // array di immagini
Inizio
    InizializzaTimer (delay);
    AvviaTimer();
    Ripeti
        Se (clock()) // arriva evento dal timer
            Disegna (Im[prec]); // disegna il fotogramma
            Se (prec < 4) prec ++;
            Altrimenti prec=0;
        Fintantochè (true);
Fine
```

In questo semplice esempio, si crea effetto di animazione su 5 immagini.

M. Malatesta B2-Animazione-18

9
07/03/2012

Lo sfarfallio

L'effetto di animazione consiste nel fatto che la funzione *Disegna()* in realtà cancella lo schermo e vi ridisegna l'immagine successiva, generalmente diversa.

Durante l'effetto di animazione di un disegno, si può osservare il fenomeno, detto **sfarfallio**, che consiste in *piccoli flash* ed effetto di *spezzettatura* dell'immagine.

Le cause dello sfarfallio sono dovute al tempo per il *ridisegno*, che prevede:

- **tempo di accesso** alla **memoria video virtuale** (la memoria che contiene le immagini da inviare al monitor, distinta dalla RAM ed accessibile direttamente dalla CPU per esigenze di velocità);
- **tempo di invio** dell'immagine al video
- **tempo di composizione** a video dell'immagine.

M. Malatesta B2-Animazione-18

10
07/03/2012

Tecniche per ridurre lo sfarfallio

Le tecniche per ridurre lo sfarfallio sono molte, per cui ne elenchiamo solo due, tra le più usate:

- **clipping**
- **double buffering**

Ovviamente, nel caso di schermate di solo testo, data la rapidità con cui i caratteri sono riprodotti sullo schermo, il problema dello sfarfallio non si pone, data la velocità con cui è possibile cancellare l'intera schermata.

M. Malatesta B2-Animazione-18

11
07/03/2012

Il *clipping*

Poiché l'animazione si basa sul ridisegno di immagini, si può pensare, ad ogni ridisegno, di *ridurre l'area da ridisegnare*, in modo da rendere più veloce l'azione di ri-disegno. Si tratta di una tecnica molto semplice che, in genere, *riduce ma non elimina lo sfarfallio*.

Il **rettangolo di clipping di default** è l'intero *canvas* (lo schermo).

Tutte le funzioni grafiche agiscono nella finestra di visualizzazione che si dice anche **rettangolo di clipping**.

La parte che va *al di fuori di questo rettangolo non viene disegnata*.

Il **clipping** è il processo che determina quali parti di un disegno appartengono all'area di visualizzazione.

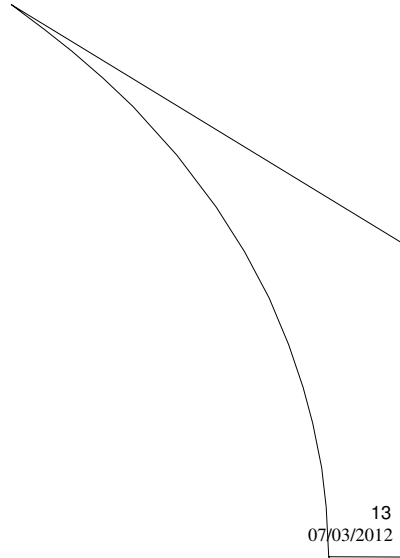
M. Malatesta B2-Animazione-18

12
07/03/2012

Il *clipping*

Possiamo avere *clipping* di:

- punti
- linee
- poligoni



M. Malatesta B2-Animazione-18

13
07/03/2012

Il *clipping* di punti

Il *clipping* di un punto $P(x, y)$ consente di stabilire se $P(x, y)$ si trova o no all'interno del *rettangolo di clipping* ($r.xmin, r.ymin, r.xmax, r.ymax$)

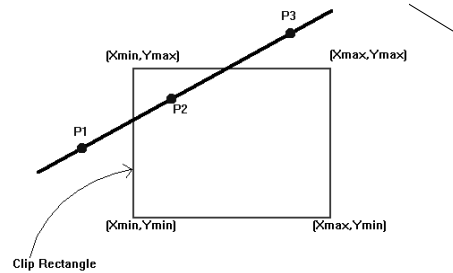
Ciò richiede che la seguente funzione restituisca **Vero**.

Booleano clipP (Intero x, Intero y)

Inizio

```
clipP = (x <= r.xmax And  
        x >= r.xmin And  
        y <= r.ymax And  
        y >= r.ymin);
```

Fine



M. Malatesta B2-Animazione-18

14
07/03/2012

Il *clipping* di linee

Il *clipping* di una linea è molto importante in quanto evita che frammenti della linea compaiano in parti dello schermo in cui non vorremmo, come, ad esempio:

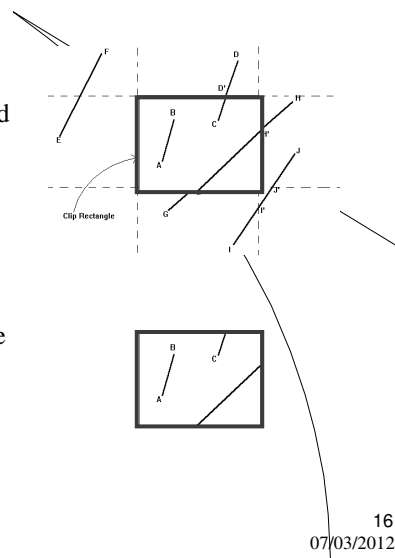
- una linea che si interrompe sul bordo destro dello schermo e riappare su quello sinistro;
- una linea che esce dal fondo dello schermo

Il *clipping* di linee

In questo esempio la finestra di visualizzazione (in alto) è l'intero *canvas* ed è anche il *rettangolo di clipping di default*.

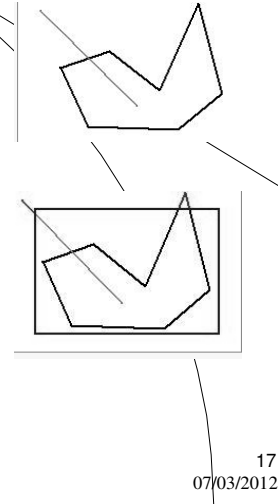
Il rettangolo rosso è il *rettangolo di clipping*.

Effettuando il *clipping* delle linee, si ottiene la figura riquadrata in basso.



Il *clipping* di poligoni

Se, ad esempio, il disegno è composto da una linea e da un poligono (figura in alto), il *rettangolo di clipping* (colorato nella figura in basso) riduce l'area eliminando le parti che sono esterne ad esso.



M. Malatesta B2-Animazione-18

17
07/03/2012

Il *clipping* di poligoni

Il *clipping* dei poligoni presenta 4 casi:

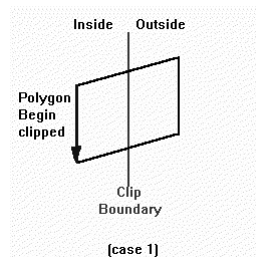
1. tutto il poligono è nella regione visibile (si salva il punto finale);
2. il poligono esce dalla regione visibile (si salva l'intersezione);
3. il poligono è tutto nella regione non visibile (non si salva nulla);
4. il poligono rientra nella regione visibile (si salva il punto di intersezione e il punto finale).

M. Malatesta B2-Animazione-18

18
07/03/2012

Il *clipping* di poligoni

1. Tutto il poligono è nella regione visibile (si salva il punto finale)

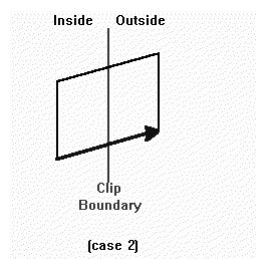


M. Malatesta B2-Animazione-18

19
07/03/2012

Il *clipping* di poligoni

2. Il poligono esce dalla regione visibile (si salva l'intersezione)

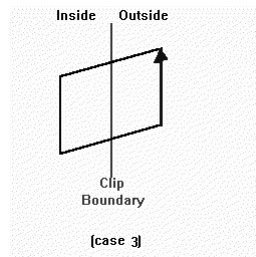


M. Malatesta B2-Animazione-18

20
07/03/2012

Il *clipping* di poligoni

3. Il poligono è tutto nella regione non visibile (non si salva nulla)

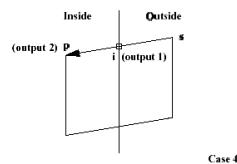


M. Malatesta B2-Animazione-18

21
07/03/2012

Il *clipping* di poligoni

4. Il poligono rientra nella regione visibile (si salva il punto di intersezione e il punto finale)



M. Malatesta B2-Animazione-18

22
07/03/2012

Il *clipping* sulle immagini fisse

Il *clipping* non si usa solo per l'animazione, ma anche per le immagini fisse.

Se si vuole fare una modifica, tramite un editor grafico, all'immagine di un particolare di un quadro, si pratica lo zoom sul particolare, per cui è inutile ridisegnare tutta l'immagine, cosicché il programma procederà più velocemente.

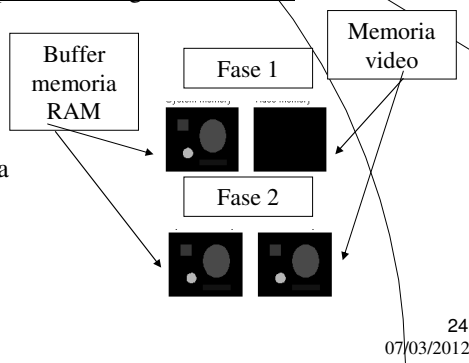
M. Malatesta B2-Animazione-18

23
07/03/2012

Double buffering

Quando il *clipping* non è sufficiente a ridurre lo sfarfallio, si ricorre al **double buffering** che consiste nel formare l'immagine (il fotogramma) in un'area di memoria (buffer) esterna al video, la quale viene visualizzata solo quando è ultimata, mentre nel buffer si predispone l'immagine successiva.

Questa tecnica, abbastanza sofisticata, offre buoni risultati, e in genere elimina completamente lo sfarfallio, tuttavia richiede una notevole elaborazione addizionale da parte del processore in quanto si devono gestire due buffer (2 “superfici” sulle quali disegnare).



M. Malatesta B2-Animazione-18

24
07/03/2012

Double buffering

La tecnica della doppia bufferizzazione può essere esemplificata con il semplice **problema del giardiniere**.

Il giardiniere, per innaffiare un grande giardino, può seguire due strade:

- utilizzare un innaffiatore, riempiendolo ogni volta dal rubinetto e portandolo poi nella zona da innaffiare;
- utilizzare due innaffiatori, in modo che mentre ne stia usando uno, l'altro si stia riempiendo; in questo modo, i tempi si accorciano perché quando il giardiniere torna al rubinetto, trova sempre un innaffiatore già pieno.

M. Malatesta B2-Animazione-18

25
07/03/2012

Double buffering

La tecnica si implementa con uno schema del tipo seguente:

- inizializzare un'immagine e un contesto grafico per il *buffer1* fuori schermo;
- creare un oggetto immagine e il contesto grafico ad esso associato, sul quale eseguire le successive istruzioni;
- al termine della procedura, visualizzare l'immagine presente nel *buffer1*;
- rilasciare la memoria occupata dal nuovo contesto grafico e dall'immagine;
- ripetere i passi precedenti sul *buffer2*

M. Malatesta B2-Animazione-18

26
07/03/2012

Argomenti

- Cos'è l'animazione
- Come si ottiene l'animazione
- Rappresentare l'animazione
- Applicazioni animate
- Lo sfarfallio
- Tecniche per ridurre lo sfarfallio
- Il *clipping*
- Il *clipping* di punti
- Il *clipping* di linee
- Il *clipping* di poligoni
- Il *clipping* sulle immagini fisse
- *Double buffering*

M. Malatesta B2-Animazione-18

27
07/03/2012

Altre fonti di informazione

- A.Lorenzi, A.Rizzi – Il linguaggio Java, ed. ATLAS
- R.Pasin, E. Di Pietro – Informatica industriale, ed. Tramontana
- R. Crandall - Soluzione di problemi scientifici, ed. Franco Angeli
- P.Camagni, R.Nicolassy – Java, ed. Hoepli

M. Malatesta B2-Animazione-18

28
07/03/2012