

## Scuola di videogame «C-64 vs Amiga»

*In questa puntata dovevamo concludere il lungo cammino del megagame con la presentazione del gioco definitivo, abbinata a una sorta di manuale di istruzioni. Sfortunatamente ci sono stati alcuni ritardi e quindi per questo mese dobbiamo accontentarci della scuola di videogame, che ospita una specie di confronto tra le possibilità del nostro piccolo e quelle della sorella Amiga; il tutto con la manifesta intenzione di cominciare a trattare anche quest'ultima come oggetto di studi per la nostra scuola. Inoltre vi presenterò un piccolo editor di videogiochi per Amiga*

Cominciamo con una descrizione sommaria, specifica per i nostri interessi, delle caratteristiche di Amiga; processore a 16 bit con clock a 7 MHz, RAM da 512 kbyte (di base), risoluzione video ottimale per videogiochi di 320x200 pixel in 16 o 32 colori (ne ha anche molte altre che però... non ci interessano), chip sonoro a 4 voci a campionamento ad 8 bit con suddivisione dei canali stereofonica, chip BLITTER per spostamenti veloci di zone RAM (soprattutto grafica), chip COPPER per la gestione del display video con elevata precisione di intervento sul pennello elettronico, 8 sprite hardware con 4 colori ciascuno (oppure 4 con 16 colori) con risoluzione di 16 X illimitati pixel, disk drive incorporato veloce (tranne che con le directory...) da 880 kbyte e due porte joystick. Caratteristiche così allettanti non potevano lasciare inerte la massa di programmatori di videogame, che infatti in questi ultimi anni ha tirato fuori da questo computer giochi di qualità sempre più elevata (anche se ogni tanto sembra di tornare ai ... vecchi tempi, quando tutto si muoveva con «terribili scatti»).

Ed ora uno sguardo al C-64; processore ad 8 bit con clock di 1 MHz, RAM da 64 kbyte, risoluzione video ottimale per videogiochi di 160x200 pixel con 4 colori, chip sonoro a 3 voci monofonico ma molto versatile, interruzioni hardware abbinata al pennello elettronico per una discreta gestione del display, 8 sprite hardware con 4 colori e risoluzione di 12x21 pixel, disk drive (esterno) non molto veloce da 170 kbyte, due porte joystick.

Nonostante la netta superiorità di Amiga su C-64 quest'ultimo non si è ancora deciso a cedergli definitivamente il passo. La spiegazione di ciò è ovviamente nel fatto che il C-64 esiste da tantissimo tempo ed ha avuto una diffusione incredibile. Inoltre lo si è studiato talmente a fondo da raggirare gran parte di quelle che inizialmente erano limitazioni, ma questa è un po' una conseguenza della prima motivazione. Il C-64 è quindi diventato una specie di standard per i videogiochi ed è quindi

ancora oggetto di interesse di molte software house. L'Amiga esiste già da un bel po' di tempo ormai ed è una fortuna che non abbia fatto la fine di alcuni suoi colleghi, aventi caratteristiche hardware alla pari se non superiori (senza fare nomi). Ne è passato di tempo da quando sono stati presentati i primi giochini stile C-64, che a me personalmente mi mandavano in bestia; tutta quella potenza sprecata e nessuno se ne rendeva conto, o meglio nessuno voleva. Per vostra fortuna non voglio tirare di nuovo in ballo la storia di Amiga, che ormai risulterebbe decisamente obsoleta.

### Un editor di videogame

Con la scusa del confronto vi presenterò una sorta di editor di videogiochi per Amiga, realizzato dal sottoscritto per produrre un giochino molto divertente (secondo il mio parere), che può essere utilizzato per la «costruzione» di qualunque spara e fuggi terrestre (tipo G'n'G... tanto per cambiare) a scorrimento orizzontale.

L'idea di partenza è quindi un giochino avente protagonista un personaggio umano (o simili...) che deve a tutti i costi avanzare lungo un territorio pieno zeppo di mostri e pericoli di ogni tipo, tanto intenzionati a ridurlo senza un briciolo di energia. Un limite di tempo costringe il nostro eroe a non soffermarsi troppo ad osservare il panorama. Tanti tanti punti se si ammazzano i cattivini.

La struttura base di un gioco-programma del genere è la stessa qualunque sia il computer sul quale viene realizzato: le differenze si riscontrano nel lato estetico e nel modo di gestire le risorse hardware della macchina. Per C-64 esiste una serie incredibile di videogiochi di questo tipo, e a suo tempo anch'io ne realizzai uno, che però non ho mai completato. Per Amiga la situazione si sta facendo molto simile. Il punto di partenza comune è la grafica, visto che il gioco è già bello e strutturato, stabilito che la qualità artistica di questa dipende in buona parte dall'abili-

*È disponibile, presso la redazione, il disco con il programma presentato in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 263.*

tà dei disegnatori, ci occuperemo della filosofia di gestione dal punto di vista software. Il C-64, come abbiamo più volte ripetuto, possiede un sistema di scrolling parte hardware e parte software e non ci soffermeremo a discuterne. L'Amiga ne possiede uno totalmente hardware, ma che necessita per forza di cose di una parte software. Tra l'altro, inizialmente nessuno si è mai preoccupato di utilizzare questa risorsa a fondo e infatti si vedeva. Il sistema utilizzato nel nostro pseudo-editor di videogiochi è in buona parte software per mia scelta (si poteva comunque fare di meglio). Esso si basa su un concetto abbastanza semplice, quello del fondale composto da «mattonelle» ovvero da una combinazione opportuna di rettangolini disegnati in modo da essere «componibili». Ciò per economizzare la RAM a disposizione, anche perché se è vero che l'Amiga ha 512k è anche vero che la grafica bitmap se la succhia come se niente fosse. A proposito l'Amiga lavora sempre come il rispettivo modo Hires del C-64; se su quest'ultimo, detto modo occupa 10k, sull'Amiga, in 16 colori, ne usa 32k e quando occorre gestire 10 schermi del genere...

Il fondale del nostro gioco è quindi composto da tante mattonelle una accanto all'altra che «scrollano» grazie al blitter, che ad ogni passata sposta di due pixel a sinistra (se lo scrolling è richiesto) tutto lo schermo e quando i

pixel diventano 16 (ampiezza di una mattonella) aggiunge 8 mattonelle in verticale a destra dello schermo (che sono ampie 24 pixel l'una per un totale di 196 pixel... restano 8 pixel verticali per scritte varie). In realtà il sistema di scrolling è un tantino più complicato ed ora gli daremo uno sguardo più approfondito. Per evitare sfarfallamenti dell'immagine il gioco utilizza due pagine di 320x200 pixel 16 colori l'una, che vengono scambiate visivamente ogni volta che quella nascosta è stata ridisegnata completamente. Esistono inoltre altre due pagine che vengono ridisegnate più lentamente e che effettuano un lavoro invisibile, ma estremamente utile. L'area di gioco effettivamente riempita dal

fondale è di 304x192 pixel. Il blitter preleva un'area di 304x192 pixel da una delle due pagine sempre nascoste. Quest'ultima pagina è completamente piena per un'area di 320x192 pixel. L'altra pagina sempre nascosta viene disegnata, sempre dal blitter, in 8 passate; il blitter quindi prende di volta in volta un pezzo dalla pagina sempre nascosta completamente disegnata e lo stampa nell'altra pagina, ma spostato di 16 pixel a sinistra e in più aggiunge per ognuna delle 8 passate una mattonella a destra (in colonna). Il risultato di ciò è che al termine delle 8 passate la pagina che stiamo riempiendo con questo sistema è completamente riempita per un'area di 320x192 pixel e contiene un

1 ▶

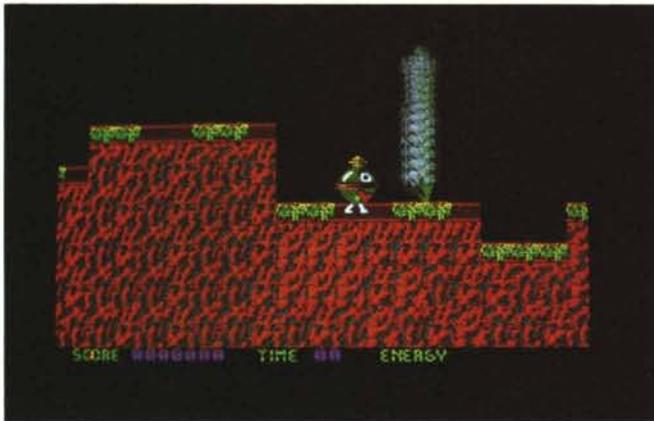


2 ▼



3 ▼



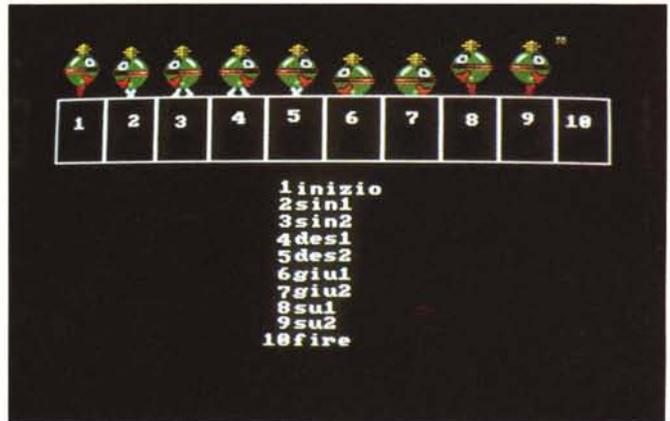


4

disegno simile a quello della pagina di partenza, ma spostato di 16 pixel e i 16 pixel che si sono eliminati a sinistra sono stati sostituiti a destra con una serie di 8 mattonelle.

E ancora non è finita. Nel frattempo il blitter deve, ad ognuna di queste passate, stampare un'area di 304x192 pixel sulla pagina invisibile che diventerà subito dopo visibile, di volta in volta prelevando due pixel più a destra questa fetta di grafica, dalla pagina mai visibile e completa. Potrà sembrare il sistema più complicato possibile, ma il suo scopo è quello di distribuire gli sforzi del blitter nel modo più uniforme possibile tra le 8 passate che compongono il ciclo. Oltre a questo ingrato compito il blitter è sottoposto anche alla stampa dei fotogrammi del giocatore e di quelli dei nemici.

Veniamo a come vengono realizzati il giocatore e i nemici. Sul C-64 avremmo risolto il problema dedicando uno o più sprite al giocatore e i restanti ai nemici, magari effettuando delle «moltiplicazioni». In ambiente Amiga possiamo utilizzare sprite, ma non è l'unica alternativa; è possibile utilizzare il blitter per stampare porzioni di grafica con una modalità particolare che permette di rendere immune da stampa zone particolari, che ad esempio vogliamo trasparenti; quindi in un disegno contenuto in un'area di 32x40 pixel, che rappresenta ad esempio una sfera, bisogna stabilire quali sono i pixel che non contengono il disegno e che quindi devono lasciare inalterato il fondale, al fine di evitare i tanto antiestetici «contorni neri». L'utilizzo del blitter non è dei più semplici, ma una volta realizzata una routine, anche complicata come in questo caso, basterà chiamarla ed il gioco è fatto. Ho utilizzato il blitter in quanto gli sprite sono troppo piccoli e hanno pochi colori. Qualcuno tra voi si sarà accorto che non ho detto come si fa a ripristinare il



5

fondale una volta che uno «blit-sprite» è stato stampato... non occorre ripristinarlo in quanto con la tecnica di scrolling che vi ho descritto sopra, il fondale viene di volta in volta ridisegnato da capo, cancellando eventuali «blit-sprite» indesiderati.

Il numero complessivo di personaggi che possono essere contemporaneamente sullo schermo (nel nostro caso) è 3, più uno blit-sprite per il «proiettile» lanciato dal giocatore. Visto che uno dei tre personaggi è il giocatore, gli altri due sono i cattivoni.

Le routine di movimento del giocatore e dei nemici sono uguali sia nel caso del C-64 che in quello dell'Amiga. Ancora diverse sono invece le routine sonore, ma solo per il modo di gestire l'audio.

Visto che l'Amiga funziona più o meno come un registratore a 4 piste basterebbe campionare una musicchetta e qualche effetto sonoro per risolvere il problema... in modo discreto. Ed è proprio così che ho fatto nel mio caso, ovvero ho messo una musicchetta campionata in loop come sottofondo musicale (il sistema, lo riconosco, è tra i più pigri possibili) e ho usato due effetti campionati per il giocatore (quando spara e quando viene colpito). Con il C-64 dovevamo realizzare una routine di gestione del chip SID o utilizzare una già preconfezionata (ad esempio l'editor musicale del sottoscritto). Il gioco è sostanzialmente tutto qui; 5 livelli diversi nella grafica e nei nemici una schermata grafica gradevole come premio finale e buon divertimento al giocatore.

### Una mano al programmatore

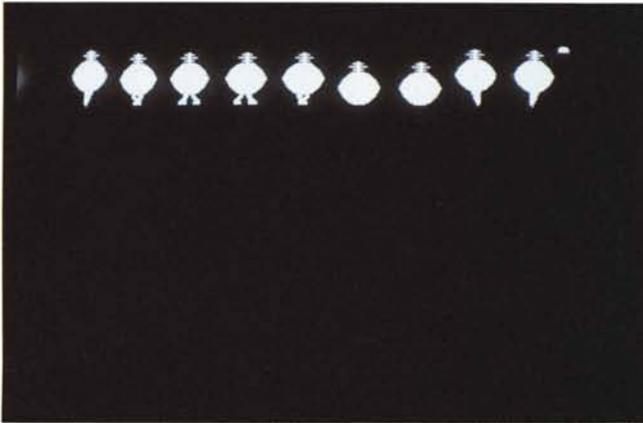
Ed ora è il momento di dare istruzioni a coloro che vorranno cimentarsi nella realizzazione del videogioco personalizzato con l'ausilio del mio pseudo-editor. Innanzi tutto si chiama «pseudo-editor»

in quanto è costituito sostanzialmente da una serie di utility e non da un vero e proprio programma stile S.E.U.C.K. Quello che l'utente «pseudo-programmatore» dovrà realizzare è il seguente materiale:

- 5 schermate contenenti le mattonelle;
- 1 schermata con i movimenti del giocatore;
- 10 schermate con quelli dei nemici;
- 5 blocchi dati per la distribuzione delle mattonelle;
- 5 blocchi dati per gli schemi di attacco dei nemici;
- una musica campionata;
- due effetti sonori;
- una schermata finale.

Ognuno di questi prodotti deve essere realizzato in un certo modo e con un determinato programma. Un solo avvertimento: l'impresa richiede un po' di abilità e quindi non è proprio alla portata di tutti, ma seguite attentamente le mie istruzioni e giudicate da soli se la cosa fa per voi. Magari se il vostro elaborato lo ritenete di qualità, potete inviarmelo in redazione: chissà che non... venga ricompensato.

Tutta la grafica va disegnata con il Deluxe Paint e dovrà rispettare le dimensioni che vi dirò tra un po'. Ogni schermata dovrà subire un trattamento particolare con apposite utility, ovviamente incluse nel pacchetto. Le cinque schermate relative alle mattonelle devono essere del tipo di quella in foto 8, ovvero ogni singola mattonella va inserita in una casella di 16x24 pixel, facendo attenzione a non creare «disallineamenti». Ogni schermata è valida per un solo livello. Potete, volendo, cambiare anche le scritte che vedete sotto (score, time, energy). Non variate troppo lo schermo colore delle schermate demo (pena problemi con le scritte). Realizzata la prima schermata con le mattonelle, salvatela e «convertitela» con il pro-



6



7

gramma «Convertitore» che trovate sul dischetto. Salvatela con il nome «01pic». Le successive schermate dei... successivi livelli devono essere chiamate «11pic», «21pic», «31pic», e «41pic». La «51pic» è lo screen finale.

Ed ora un elenco dei nomi da dare ai file:

**musica:** file della musica campionata

**suonoaio:** file del suono giocatore ferito

**suonotie:** file del suono giocatore che spara

**02pic:** schermata con movimenti del giocatore (vedi foto 4)

**01mas:** schermata con ombre del giocatore (foto 5).

File che variano con il livello:

**01pic:** schermata con mattonella (primo livello).

**03pic:** scr. sprite dei nemici1 (primo livello)

**04pic:** scr. sprite dei nemici2 (primo livello)

**02mas:** scr. ombre dei nemici1 (primo livello)

**03mas:** scr. ombre dei nemici2 (primo livello)

**0data:** dati delle mattonelle

**0nem:** dati dei nemici

per i successivi livelli basta sostituire la prima cifra del nome con i numeri 1, 2, 3 e 4;

**51pic:** schermata finale.

Tutte le schermate sono in 16 colori. La schermata con i movimenti del giocatore (che è rappresentato da un'area grafica di 32x40 pixel) deve contenere i movimenti come in foto 4, ovviamente cambiando a piacere il disegno. Anche detta schermata va convertita con il programma «Convertitore». L'ultimo fotogramma è il proiettile che spara il giocatore (a gittata limitata). Per realizzare lo screen con le ombre (foto 5) occorre catturare (siamo in ambiente

DPaint) con un brush tutti i fotogrammi e renderli del colore grigio scuro (l'ultimo) con l'opzione «Color» quindi stamparli nella stessa posizione e salvare lo screen, per poi convertirlo con il programma «maschera» e salvarlo con il nome «01mas». Anche gli screen dei nemici vanno realizzati con la stessa modalità e con le ombre, solo che i fotogrammi di animazione sono solo 2 (più che sufficienti nel nostro caso); uno è quello della schermata di foto 6 e l'altro quello della schermata di foto 7 (si alternano all'infinito con una certa frequenza). Le loro dimensioni sono le stesse del giocatore.

Per realizzare le mappe (0data e 0nem) occorre il programma «Edtabelle». Prima di addentrarci nella spiegazione del suo funzionamento voglio dirvi quali sono gli algoritmi di movimento dei nemici. Essi non sparan mai, ma possono:

- cadere dall'alto e scoppiare;
- cadere dall'alto e rimbalzare;
- camminare a destra (evitando ostacoli);
- camminare a sinistra (evitando ostacoli);
- restare fermi.

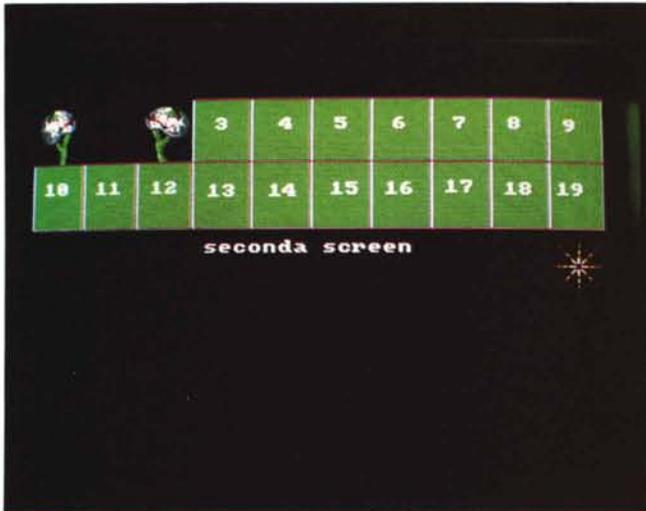
Solo uno a scelta di questi algoritmi può essere attribuito a ciascun nemico. I nemici entrano in azione in funzione della posizione dello scrolling ed è in base ad essa che verranno programmati.

Quando un nemico viene colpito dal proiettile del giocatore esso esplosa con il fotogramma a forma di stella delle foto 6 e 7.

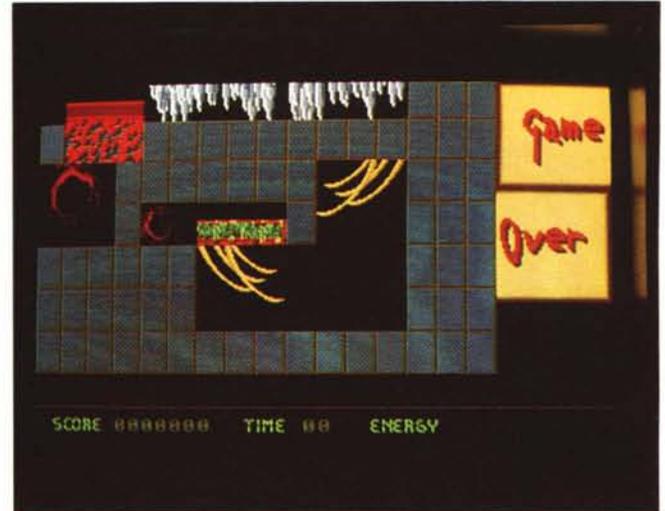
Il giocatore muore quando finisce l'energia o il tempo.

E veniamo all'editor di tabelle «Edtabelle». Innanzi tutto va caricato con l'AmigaBasic. Subito dopo il caricamento ci viene chiesto il nome della schermata che contiene le mattonelle (nel caso del primo «quadro» il nome è «01pic») quindi il nome della tabella-

mattonelle (se per caso ne avevamo già iniziata una) e quello della tabella-nemici. Cominciamo con la realizzazione della tabella-mattonelle. A questo punto occorre improvvisarsi equilibristi (sto scherzando) per riuscire a capirci qualcosa tra tutte le schermate che possono visualizzarsi. Innanzi tutto occorre rendere visibili i gadget del Workbench che permettono lo scambio tra gli schermi (quelli in alto a destra) e quindi scambiare il Workbench con lo screen sottostante. In questo dovremo comporre il nostro «mosaico», servendoci della schermata sottostante, contenente il disegno delle mattonelle. Per passare da uno screen ad un altro è sempre necessario servirsi dei gadget di Intuition quindi fate attenzione perché negli screen sottostanti il Workbench questi non sono visibili (occorre andare a memoria). Ma vediamo come si compone il mosaico. Il puntatore del mouse controlla una mattonella base che può essere depositata sul terreno semplicemente premendo il tasto «P». Per cambiarla basta andare nello screen delle mattonelle e premere il tasto «M» su quella desiderata (per poi tornare al «foglio da disegno»). Ovviamente è anche possibile scegliere di «tirare giù» lo schermo «a tendina», per vedere sia le mattonelle che il foglio da disegno (vedi foto 9). La metà a destra dello schermo è quella effettivamente disegnabile, mentre quella a sinistra serve per avere un sistema di riferimento. Premendo la barra spazio si memorizza la mezza schermata a destra e verrà effettuato uno scrolling grosso di mezza pagina. A questo punto potete continuare a disegnare. È anche possibile effettuare uno scrolling manuale per le correzioni o per visualizzare una particolare zona della mappa-tabella; basta servirsi dei tasti del tastierino numerico «4» per sinistra e «6» per destra con salti di mezzo schermo. C'è anche una funzione di



8



9

GOTO alla pagina numero tal-dei-tali; si preme «G» e si inserisce il numero desiderato. Ci sono notevoli possibilità di rendere più rapida la composizione della mappa, prelevando fette più grandi di una mattonella (gruppi di mattonelle) sia dallo screen che le contiene che dalla mappa stessa (una specie di funzione brush del DPaint). Per fare ciò è necessario servirsi dei tasti:

«1» - per indicare la mattonella di start;  
 «2» - per indicare quella di end (prelevando dalla mappa);  
 «3» - per indicare quella di end (prelevando dallo screen mattonelle).

Per stampare questo gruppo di mattonelle basta premere il tasto «I» (dopo aver ovviamente «mirato» nel punto desiderato). Con il tasto «O» si preleva la mattonella base direttamente dallo screen della mappa.

In qualunque momento è possibile salvare («SAVE») la tabella premendo «shift»+«S» e quindi rispondendo alla richiesta del nome.

Terminata la composizione (circa 15 schermate di ampiezza orizzontale), occorre definire quali tra le mattonelle dovranno risultare «solide», ovvero impenetrabili, sia da parte del giocatore che da quella dei nemici. È chiaro che già in fase di disegno si deve studiare più o meno quali saranno dette mattonelle. Esse impediscono di essere attraversate e quindi ci si può camminare sopra (con gli sprite!) e possono limitare l'avanzata e costringere a «salti». È importante non disporre più di 3 mattonelle solide in verticale, pena un'impossibilità di salto da parte degli sprite. Vediamo come si definiscono; bisogna entrare in questa modalità premendo «shift»+«M», quindi occorre andare sullo screen delle mattonelle e puntarle con il mouse, premendo il tasto «M»

per confermare. Terminata la scelta si preme lo spazio e inizia il processo di «solidificazione», che durerà un bel po' (circa 15 secondi per mattonella solida). Ad ogni mattonella completata si udirà un «beep».

Terminata questa operazione potete salvare finalmente il blocco definitivo della tabella-mappa. Se per caso avete solidificato mattonelle sbagliate, basterà entrare nel modo di disattivazione della solidificazione, premendo «shift»+«N» e eseguendo lo stesso tipo di selezione. Quando una mattonella viene definita come solida essa si deforma per indicare che viene definita come solida essa si deforma.

Passiamo ora all'edit del movimento dei nemici. Per entrare in questa modalità occorre premere «shift»+«Q». A questo punto fate «scrollare» lo schermo di una mattonella alla volta con il tasto «A» fin quando non arrivate alla posizione da voi scelta per far apparire il primo nemico; premete il tasto «I» e il programma vi chiederà quale deve essere il tipo di movimento (da 0 a 4). I numeri corrispondono a:

- 0 - nemico che cade e si «rompe»
- 1 - nemico che cade e rimbalza
- 2 - nemico che avanza verso sinistra
- 3 - nemico che avanza verso destra
- 4 - nemico che resta fermo nella posizione di partenza.

Scelto il tipo di movimento il programma ci chiede di posizionare il giocatore sullo schermo, quindi dobbiamo tornare alla mappa e puntare il mouse, tenendo conto che la megamattonella che state spostando è delle dimensioni di un nemico.

Segue la richiesta che stabilisce quale sarà lo sprite di animazione (e dove-

te farvi un elenco di dove sono posizionati nelle vostre schermate; i numeri si contano da sinistra a destra, come in foto 6 e 7). Siamo giunti alla conclusione del blocco dati per il primo nemico; ora basta ripetere l'operazione per tante volte durante tutto il percorso-mappa. Tenete comunque presente che se sullo schermo dovrebbero esserci più di 2 nemici, il primo di questi che è stato attivato verrà fatto «esplodere» (quindi non esagerate con gli attacchi). Per salvare questa tabella basta premere «shift»+«S» e impostare il nome. La collisione giocatore-nemico comporta un decremento continuo di energia; la collisione proiettile-nemico comporta un incremento del punteggio.

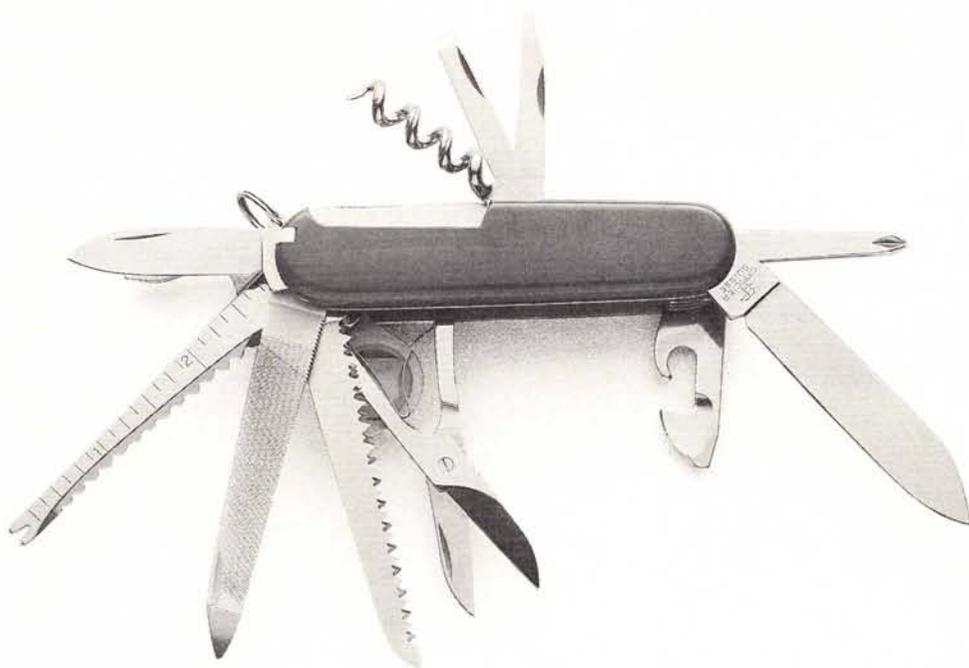
Mancano a questo punto solo informazioni su i file sonori.

La musica campionata deve avere una lunghezza di 26690 campioni e su sampling rate di 8363. Il «suonoio» è lungo 3778 e ha lo stesso sampling rate. Il «suonotie» è lungo 1794 e ha un sampling rate di 9800.

Sul dischetto entrano a mala pena tutti i dati delle schermate e delle tabelle, quindi bisogna farne una copia e ripulirla delle utility. La startup-sequenza è già predisposta per far avviare il gioco in autostart.

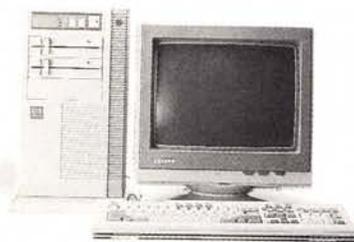
Per usare i programmi-utility occorre caricare il Workbench e fermarlo con «CTRL»+«D» per poi inserire il dischetto contenente i programmi e avviarli da CLI. Tutte le schermate e le «ombre» devono essere salvate sullo stesso dischetto contenente le utility e poi copiate su quello del gioco, pena una parziale ma dannosa cancellatura delle stesse. Vi auguro di cuore buon divertimento.

MC



# INTEGRATI PER ECCELLENZA

La maggior parte dei produttori hanno l'abitudine di mettere assieme un'assortimento di parti di computer e di annunciare al tempo stesso che sono "produttori di sistemi". E puo' andargli bene, qualche volta. Non sempre. Le schede possono essere strambe, arrestando ai produttori e di conseguenza a voi, una somma di problemi di incompatibilita'. La CST Corporation si e' costruita una eccellente reputazione sulla base di una propria industria produttiva, che fabbrica ogni parte dai PCB al complesso PC systems.



I nostri sistemi coprono la gamma dei 286, 386 e le unita' da ufficio 486 oltre al laptop NEAT286 per gli uffici del futuro. Con il nostro controllo della produzione della componentistica dei sistemi, siamo in grado di testare i prodotti completamente e meglio della concorrenza.

Di conseguenza, siamo in grado di mettere d'accordo reliability bene come la compatibilita'.

I sistemi CST sono progettati sistematicamente, e cosi' eseguono il lavoro come vi aspettate.

Senza errori.

Practical solutions from

## CST

**Head office:**

Chung Shing Technics Corporation  
12F, No. 123, Sec.2, Chung Hsiao E. RD.  
Taipei, Taiwan, Republic of China  
Tel: 886-2-322-5088, Fax: 886-2-341-0577  
Tlx: 16656 KINC

**Factory:**

B1, No. 447, Sec. 2, Chung Hwa Rd.  
Taipei, Taiwan, Republic of China  
Tel: 886-2-303-6421 - 2  
Fax: 886-2-305-5945

**Australia branch:**

CST Computers (Australia) Pty. Ltd.  
1 Highgate Street, Auburn  
NSW 2144, Australia  
Tel: 61-2-748-3711  
Fax: 61-2-748-4714

**Hong Kong branch:**

CST Computers (H.K.) Ltd.  
Room 1804, Star Centre  
443-451 Castle Peak Rd.  
Kwai Chung, N.T., Hong Kong  
Tel: 852-481-7299, 852-481-7314  
Fax: 852-481-7469

**Thailand branch:**

Chung Shing Technics Corporation  
4F, Boonmitr Bldg., 138 Silom Rd.  
Bangkok 10500, Thailand  
Tel: 66-2-235-0421 - 4  
Fax: 66-2-236-4734, 66-2-236-0207  
Tlx: 87237 CONSOL TH

**U.S.A. branch:**

Chung Shing Technics Corporation  
2009 Dewberry Court  
Westlake Village, CA 91361, U.S.A.  
Tel: 1-805-496-8053  
Fax: 1-805-497-3606

**West Germany branch:**

CST Computer Vertrieb GmbH.  
7515 Linkenheim  
Gewerbering 1, West Germany  
Tel: 49-724-74055/8  
Fax: 49-724-74060