

Scuola di videogame

Il discorso che andiamo ad affrontare in questa puntata si riferisce al progetto di una routine per il movimento dei «nemici» in uno «spara e fuggi». Alcune considerazioni sono valide in generale, ma effettueremo una distinzione fra tre categorie che illustreremo appresso

Cominciamo dall'ipotesi più banale, ovvero dal caso semplice di un determinato «set» di nemici (vi ricordo che sotto questo nome consideriamo ogni tipo di ostacolo che il giocatore incontrerà durante il suo tragitto), che agiscono secondo schemi prefissati e con un ordine prestabilito. In tal caso la routine è costituita da un insieme di algoritmi per il movimento (ad esempio una decina) e da una tabella che determina i tempi e le modalità di attacco dei nemici. Facciamo un esempio: un gioco «spaziale» a scrolling orizzontale verso sinistra (vedi foto 1) dispone di una tabella che contiene una serie di eventi, collegati allo scrolling, ovvero, ogni qualvolta si giunge ad una posizione della coordinata X contenuta anche nella suddetta tabella, si attiva una nuova routine di movimento e quindi dei nuovi sprite con relative animazioni, che in genere è meglio definire con tabelle di contorno. Supponiamo quindi (foto 2) che l'attuale posizione dello scrolling faccia attivare l'ingresso in campo di sette nuovi nemici (in foto solo 4); evidentemente nel caso del Commodore 64 (a meno di «moltiplicazioni») il nuovo ingresso deve coincidere con una «uscita», pena «sfarfallamenti» indesiderati. Andiamo a considerare alcuni aspetti della routine. I problemi da risolvere sono:

- inializzazione;
- movimento;
- esclusione dei «morti»;
- uscita;
- gestione ordinata degli sprite disponibili.

Nel caso del gioco delle foto 1 e 2 (che possiamo far rientrare nella «categoria 1») le cose sono notevolmente semplificate (in genere) perché il fondale è ininfluente, o meglio non fa da vincolo e quindi le routine di movimento sono semplici sequenze di posizioni da assumere (o funzioni matematiche che restituiscono dette posizioni).

Il problema della inializzazione è strettamente collegato con la gestione ordinata degli sprite, che tuttavia, per la categoria 1, è da considerarsi semplice. Sostanzialmente si tratta di eliminare gli sprite che sono in campo se servono per la «formazione» in ingresso (ad esempio con una esplosione), ma in genere il problema è ancora più sempli-

ce, in quanto possiamo fare in modo che sullo schermo non compaiano nuovi sprite se quelli disponibili sono già tutti utilizzati. Dal momento che non dobbiamo preoccuparci di questo problema, possiamo dedicarci alla struttura della tabella; essa deve comprendere informazioni sulla posizione di partenza e sulla routine da attivare per il successivo movimento (consideriamo la tabella costituita da informazioni relative ad un solo nemico per volta), più la posizione di controllo dello scrolling. Se vogliamo ottenere l'effetto della foto 2 (che purtroppo è statica!) ognuno dei sette nemici deve essere attivato ad una posizione dello scrolling molto vicina a quella del precedente ed utilizzando la stessa routine di movimento (che quindi deve essere «multi-utente»), facendo attenzione a calcolare nel frattempo anche la nuova posizione di partenza (visto che lo schermo si è spostato) se vogliamo che i nemici «entrino» tutti dallo stesso punto. Se vogliamo che un certo nemico sia formato da più sprite messi assieme basta attivare il numero di sprite necessari (ad esempio 4) nello stesso momento, quindi alla stessa posizione dello scrolling, e posizionarli nel modo giusto, ovvero secondo l'effetto che vogliamo ottenere. Quest'ultima possibilità ci impone una regola nel progetto delle routine di movimento; esse infatti dovranno gestire un massimo di 7 sprite (1 è del giocatore... e non abbiamo contato i «colpi») e quindi 14 coordinate indipendenti, che tuttavia seguiranno sempre lo stesso schema, in modo da fornire lo stesso spostamento (vi faccio presente che le routine non devono fornire coordinate precise, ma solo le direzioni di spostamento) se vengono attivati più sprite contemporaneamente. Ciò permette anche di ottenere l'effetto «serpente», se gli sprite sono parzialmente sovrapposti e sono attivati a breve distanza. Non è da escludere ad ogni modo una routine che fornisce spostamenti casuali, che ovviamente è da utilizzarsi solo nel caso di sprite singoli.

Facciamo uno schema di riepilogo per la tabella:

- posizione dello scrolling;
- posizione di start (x, y);
- routine da utilizzare;
- animazione da utilizzare (con sotto-



Foto 1



Foto 2

tabella);

ripetuta per ogni sprite-nemico.

Veniamo alle routine di movimento e supponiamo che il loro sistema di definizione degli spostamenti sia costituito da un insieme di tabelle (quindi niente funzioni matematiche). Una volta attivate esse dovranno fornire una direzione di spostamento per ogni ciclo raster e per ogni sprite che la utilizza. L'inizializzazione consiste nell'azzeramento del puntatore alla tabella. In conseguenza dell'attivazione esse restituiranno (per il relativo sprite), in appositi registri, la direzione di spostamento, tra le otto + una (spostamento nullo) disponibili. Tengo a precisarvi che gli sprite subiscono anche un decremento della loro coordinata X per ogni spostamento dello scrolling. Volendo fare le cose un po' più sofisticate possiamo anche indicare, oltre alla direzione, la quantità di spostamento, al fine di ottenere effetti di accelerazione e viceversa. Le tabelle di spostamento possono essere lunghe a piacere e magari cicliche, per ottenere ad esempio, una continua rotazione intorno ad un punto, con possibilità di definire la posizione di restart per il ciclo, in modo da separare la fase di ingresso e quella di loop (l'ingresso dall'alto e la rotazione ciclica si ottengono definendo la posizione di restart coincidente con l'inizio della fase circolare).

Supponiamo quindi di avere 10 (diciamo subito che in genere sono poche) di

queste routine, ognuna con una determinata traiettoria. La loro struttura è del tipo di quella seguente:

- attendi l'attivazione;
- se attivata per la prima volta «reset»;
- deposita lo spostamento per ogni sprite attivo nei relativi registri;
- incrementa i puntatori (attivi) alla tabella;
- «reset» in caso di «end-tabella».

Esistono poi 7 registri per indicare lo spostamento e la quantità dello stesso (ad esempio «nord-3 spazi»).

È evidente che le traiettorie dovranno essere ben collaudate, ovvero non dovranno provocare strani movimenti, ma questi sono problemi scontati. Associate alle routine di movimento abbiamo le routine di animazione e controllo dell'uscita dallo schermo o «morte». In sostanza esse devono prelevare i registri di spostamento solo nel caso il nemico sia ancora vivo e devono effettuare l'animazione che per semplicità supponiamo ciclica (ad esempio con 4 fotogrammi), ma che volendo potrebbe anche essere in funzione della direzione (visto che le informazioni ci sono). In caso di uscita, questa routine deve disabilitare lo spostamento e rendere disponibile lo sprite. In caso di «morte», si deve attivare la sequenza di esplosione e, una volta terminata, rendere disponibile lo sprite.

Sintetizziamo tutto il discorso con un

esempio chiarificatore (speriamo). Il nostro fondale è in scrolling e la nostra astronave vola liberamente. Arrivati ad una determinata posizione la routine di controllo della tabella degli eventi verifica una attivazione e quindi imposta la routine indicata nei dati, la posizione di partenza e l'animazione da utilizzare. Fatto questo la palla passa alle altre routine; innanzi tutto viene scelto uno sprite tra quelli a disposizione, poi la routine di spostamento effettua il suo dovere e segue la routine di animazione e controllo «morte», che utilizza la tabella indicatagli dalla inizializzazione. Chi stabilisce la morte è la routine delle collisioni che in questa sede non trattiamo. Di contorno ci sono ancora subroutine che si occupano di gestire il chip video, quindi di trasformare le informazioni in modo da inserirle nei registri opportuni.

Abbiamo quindi concluso il caso della «categoria 1». In conclusione di questa parte vorrei darvi un consiglio, valido in generale per qualsiasi progetto; se dovette realizzare insiemi di dati complessi e suscettibili ad errori di compilazione, vi conviene «costruirvi» un editor specifico; il tempo speso vi verrà largamente ricompensato in seguito.

I «più attenti» avranno capito che le restanti foto si riferiscono alle categorie 2 e 3 (rispettivamente, foto 3 e 4). La categoria 2 è quella degli spara e fuggi a scorrimento verticale (ma volendo pos-



Foto 3



Foto 4

siamo farli anche a scorrimento orizzontale) con vista dall'alto e con fondale interattivo. Quest'ultima caratteristica ci complica notevolmente le cose, in quanto la traiettoria deve essere realizzata con un algoritmo e non con una semplice sequenza di spostamenti.

Ciò modifica solo la routine che determina la traiettoria, ma non il sistema di attivazione; al limite possiamo effettuare anche un ibrido tra i due metodi di definizione della traiettoria, in quanto non è escluso che vi siano nemici «volanti» e che quindi non vengono vincolati dal terreno. Anche la categoria 3 (vista laterale con fondale interattivo) utilizza lo stesso metodo, ma gli algoritmi sono notevolmente differenti, visto che in questi giochi si tratta soprattutto di «saltare» eventuali «buche» o «muri», mentre per la categoria 2 basta cambiare direzione se si incontra un «muro». Tuttavia in entrambi i casi occorrono routine di controllo del fondale che ci costringono a realizzare tabelle per specificare dove si può passare e dove ci si deve fermare.

Una totale rivoluzione la possiamo effettuare se decidiamo di inserire nemici «intelligenti». Dovremo realizzare algoritmi capaci di individuare le mosse dell'avversario (giocatore) e quindi di controbattere, magari effettuando «finte». Un esempio lampante sono i giochi di lotta («International Karate»). In aggiunta agli algoritmi di movimento vincolato vanno quindi costruite routine di ricerca della posizione (sia dell'avversario che di eventuali «proiettili») e scelta della mossa da eseguire (quest'ultima al limite potrebbe anche essere fatta casualmente). È chiaro che il nemico non

deve essere invincibile. L'ingresso in campo di un nemico di questo genere potrebbe essere predeterminato come nei casi precedenti o, volendo, anche casuale (senza esagerare). Se si riesce ad effettuare un giusto miscuglio delle

tecniche descritte il risultato è senza dubbio più soddisfacente (possiamo, in fase di gioco, differenziare i nemici in stupidi e ingegnosi e tutto diventa più divertente). Adesso però devo salutarvi... a presto.

Megaposta

«Carissimo Marco Pesce, sono un ragazzo di 20 anni e frequento il secondo anno di ingegneria. Ritengo di averti già sommerso di informazioni inutili quindi passo al dunque della mia missiva. La prima cosa da dire è che non sono un assiduo lettore di questa rivista, ma interessandomi di videogame sono stato colpito dalla tua iniziativa (un po' in ritardo forse, ma purtroppo l'ultima volta che ho comprato MC ancora non c'era traccia del tuo operato!). La cosa che mi piace di più è il tipo di V.G. scelto, che non rientra assolutamente tra i più diffusi (almeno stando a quanto ho capito). Io sono dell'idea che un gioco al computer deve essere considerato un'opera d'arte e in quanto tale deve offrire qualcosa di inconsueto per essere degno di nota. E sembra proprio che il tuo caso sia da considerarsi perlomeno eccentrico. Ecco, questa era la cosa più importante che volevo dirti. C'è anche un'altra cosa che voglio dirti: quando pensi che sarà pronto?...».

Massimiliano Vetriani, Roma

del megagame... Vogliamo fare una previsione? Penso proprio che per il numero di marzo potremo cominciare le nostre battaglie intergalattiche... grazie dei complimenti.

Caro maestro (Marco Pesce), è da molto che attendo una tua svolta per la scuola di videogame, nel senso che ormai sarebbe ora di passare al computer del momento (il piccolo grande Amiga); mi sembra che anche tu avevi accennato a una cosa del genere. Credimi, a sentire i pareri di molti sembra proprio che riscuoteresti un successo...».

Luca Cesaroni, Milano

A parte il fatto che non ho capito di chi sono questi «pareri», comunque mi trovo in perfetta armonia con te. In effetti hai ragione, e del resto è molto che mi occupo di questo computer (in privato) senza mai aver accennato nulla su queste pagine... ancora poche battute per il 64 e poi vedremo cosa si può fare.

Finalmente qualcuno che si preoccupa

MC

GLI HARD DISK

FLASHBANK

Hard Disk e DMA controller su scheda per A2000 e Zorro Big Blue. Velocità trasferimento dati: 7,5 Mbit/sec. Si inserisce su slot a 100 pin. Autocboot con sistemi operativi 1.2 e 1.3. Autoconfging. Formattato con Fast File System.

FLASHBANK 32Mb 40ms	L. 890000
FLASHBANK 60Mb 40ms	L. 990000
FLASHBANK 60Mb 25ms	L. 1240000

Modulo A2090 Autoboot

Modulo che rende autoboot i controller Commodore A2090. Si inserisce su uno slot a 100 pin. L. 129000

MULTIBRAIN

Hard Disk e DMA controller per A500/1000. Autoboot. Autoconfging. Formattato con FFS. Espansione opzionale RAM da 2 a 8 MB.

MULTIBRAIN 20 Mb 40ms	L. 890000
MULTIBRAIN 40 Mb 40ms	L. 1090000
MULTIBRAIN 40 Mb 25ms	L. 1390000
Modulo RAM 2Mb L. 690000 - 4Mb L. 1190000 - 8Mb	L. 2440000

IMPACT A2000 GVP

HD controller SCSI piú Espansione RAM 2Mb per A2000 o ZBB con autoboot. O Kb L. 440000 - 2Mb L. 990000

IMPACT HC2000

Come sopra, ma senza RAM, con la possibilità di montare l'hard disk direttamente su scheda. L. 410000

TOP CARD HARDITAL HD CONTROLLER IN STANDARD SCSI

Costruito completamente in tecnologia VLSI A 2000 o ZBB

HARD DISK per controller SCSI 20 MB 3.5" 40ms miniscibe	L. 640000
HARD DISK per controller SCSI 47 MB 3.5" 28ms seagate	L. 890000
HARD DISK per controller SCSI 80 MB 3.5" 11ms quantum	L. 1690000
HARD DISK per controller SCSI 105 MB 3.5" 11ms quantum	L. 1890000

A2090A Commodore

HD controller piú Harddisk da 20 Mb per A2000. Autoboot. L. 1090000

A590 Commodore

HD controller piú hardisk da 20 Mb con espansione Ram da 0 a 2Mb autoboot per A500. L. 920000

HD2000card

Controller ed HardDisk da su scheda per AMSTSRAD, IBM/XT o A2000 con Janus. HD 2000CARD 32MB L. 690000, 40MB L. 740000

Janus XT

Emulatore IBM/XT per A2000 + drive da 5,1/4 L. 540000

Janus AT

Emulatore IBM/AT per A2000 + drive da 5,1/4 con garanzia Commodore Italia L. 840000

LE ESPANSIONI DI MEMORIA AMEGABOARD

Espansione di memoria per A500/1000 da 2 a 8Mb. Esterna. Autoconfigurante. Si installa sul connettore laterale. Munita di connettore passante per altre periferiche completa di LED e di interruttore per il diserimento senza disconnetterla dal computer. Dimensioni 21 x 10 x 4,7 cm. L. 890000

AMINTERAM

Espansione di memoria per A500 da 512 Kb. Si inserisce nell'apposito slot del computer. Con orologio e batteria tampone. L. 189000

SUPEROTTO HARDITAL

Espansione da 0-2-4-8Mb sulla stessa scheda per A2000 o ZBB. Completa di display con indicazione della memoria disponibile e di led di autoconfgurazione. Zerì wait state.

2Mb L. 740000, 4Mb L. 1240000, 8Mb	L. 2240000.
------------------------------------	-------------

A2058 Commodore

Espansione da 2 a 8 Mb per A2000. 2 Mb L. 1040000

KICKROM 1.3 A1000

Kickstart 1.3 su eeprom senza saldature per A1000 con orologio tampone. Si inserisce sul connettore laterale del computer. Completo di connettore passante. L. 1490000

KICKROM 1.3 A500/A2000

Kickstart 1.3 su Eprom interno per A500/2000. Completo di deviatore per Kickstart 1.2. L. 890000

I DRIVE

ADRIVE

Drive da 3,5 esterno per A500/1000/2000. Con interruttore per il disinserimento e di connettore passante. L. 199000

ADRIVE TOWER

Come sopra, ma triplo nello stesso contenitore. L. 490000

ADRIVE 2000

Drive interno da 3,5" per A2000. L. 169000

ACCELERATORI-PROCESSORI-COPROCESSORI

ATTENZIONE!!! I PREZZI SOTTOINDICATI COMPREDONO LE SCHEDE ACCELERATRICI SENZA PROCESSORI E COPROCESSORI CHE SONO INDICATI A PARTE. QUESTO PER LASCIARE IL MASSIMO GRADO DI LIBERTÀ ALL'UTENTE.

HURRICANE

Scheda acceleratrice per A1000/A2000. Hurricane A1000 L. 499000 - Hurricane A2000 L. 799000.

BANG

Scheda acceletrice per A1000/A500. L. 340000

HURRICANE MEMORY 1-4Mb

Espansione di memoria 32 bit per Hurricane. Hurricane Men 1MbL. 1190000

ADAPTER 030

Adattatore per 68030 per Hurricane e Bang. L. 24.000

PROCESSORI: 68010 L. 49000 - 68020 L. 290000 - 68030 L. 590000

COPROCESSORI: 68881 - 12MHz L. 240000 - 16MHz L. 290000

25MHz L. 740000 - 68882 16MHz L. 390000 - 25MHz L. 990000

A26260

Scheda acceleratrice contenente 68020 a 14,3MHz + 68881 a 16MHz + MMU68851 + RAM a 32 bit da 2 Mb L. 2190000

I DIGITALIZZATORI AUDIO VIDEO

GENLOCK CARD A2301 Commodore

Scheda Genlock semiprofessionale per Amiga 2000. L. 390000

FLICKER FIXER

Scheda da inserire nello slot video dell'A2000 ed elimina il flicker. L. 690000. Flicker Fixer + Monitor Multisync NEC 2A L. 1390000

ZORRO BIG BLUE

Chassis metallico per A500/1000 comprendente mainboard con 3slot 100 pin A2000 piú 3 slot XT, 3 slot At compatibili piú 1 slot CPU a 86 pin per schede con 68020/68881 (Hurricane, A2620). Completo di alimentatore switching da 180 W. Predisposizione per 2 drive da 3,5", 1 - drive da 5,1/4" 1 hard disk da 3,5" o 5,1/4" e digiboard. L. 470000

MODULO DRIVE

1 o 2 drive da 3,5" -880Kb. L. 189000

ATTENZIONE!!! NELLO ZORRO BIG BLUE SI POSSONO MONTARE TUTTE LE SCHEDE PER L'AMIGA 2000 (JANUS XT/AT, AMEGADRIVE, A2058, SUPEROTTO, IMPACT, A2620, A2090, ecc.)

I MONITORS

NEC MULTISYNC 2A L. 990.000

COMMODORE 1084

Monitor HiRes per A500, A1000, A2000. L. 520000

PHILIPS 8833

Monitor stereo per Amiga o PC L. 520000

STAMPANTI

STAR LC 10 L. 420000

STAR LC 10 color L. 490000

STAR LC 15 136 colonne L. 790000

STAR LC 24-10 L. 640000

Stampante a 24 aghi 150 cps NLQ.

I COMPUTER

AMIGA 500

Completo di mouse e manuali con garanzia Commodore Italia 1 anno. L. 850000

Come sopra ma con espansione da 1Mb e orologio L. 990000

AMIGA 2000

CON MONITOR E SECONDO DRIVE DA 3,5" con garanzia Commodore Italia 1 anno L. 2450000

DISCHETTI DA 3,5"

1 L. 1900 - 10 L. 1600 - 100 L. 1250

SONO INOLTRE DISPONIBILI TUTTI I COMPUTER E LE PERIFERICHE AMSTRAD. CHIEDERE.

DISPONIBILI ANCHE TUTTI I COMPATIBILI XT, AT E PS2. CHIEDERE

PER INFORMAZIONI E/O ORDINAZIONI



VIA FORZE ARMATE 260 20152 MILANO

TELEFONO 02-4890213

VENDITA SOLO PER CORRISPONDENZA

TUTTI I PREZZI SONO IVA COMPRESA

FLASHBANK



- HARD DISK CARD PER A 2000 E ZORRO BIG BLUE
- AUTOBOOT CON EPROM
- FULL AUTOCONFIG
- CAPACITÀ: 20-32-40-60 Mb

MULTI BRAIN

- HARD DISK E CONTROLLER PER A 500 / A 1000
- AUTOBOOT CON EPROM
- AUTOCONFIG
- ESPANSIONE OPZIONALE DA 2 A 8 Mb
- 1 POSTO PER DRIVE DA 3,5
- MODULO OPZIONALE CON DISPLAY DELLA CAPACITÀ DI MEMORIA, CAPACITÀ HARD DISC E TRACCE DISC DRIVE
- CAPACITÀ: 20-32-40-60 Mb

SUPER 8

- ESPANSIONE DI MEMORIA PER A 2000 E ZORRO BIG BLUE
- AUTO CONFIGURANTE
- ZERO WAIT STATE
- DISPLAY CON INDICAZIONE DELLA CAPACITÀ INSTALLATA
- CAPACITÀ: 0-2-4-8 Mb

ZORRO BIG BLUE



- 3 SLOT A 100 PIN A 2000 COMPATIBILI
- 3 SLOT IBM XT COMPATIBILI
- 3 SLOT IBM AT COMPATIBILI
- 1 SLOT A 86 PIN PER 68020/68881
- 2 POSTI PER 2 DRIVE DA 3,5"
- 1 POSTO PER 1 DRIVE DA 5,1/4
- 1 POSTO PER HARD DISC
- ALIMENTATORE SWITCHING



VIA TORTONA, 12

20144 MILANO

Tel. 02 - 8376887