

a cura di Andrea de Prisco



■ Non è immediato calcolare a quanti MIPS (cfr. MC n. 68 Appunti di Informatica) corre un Amiga, ma per quello che ha combinato il lettore Paolo Russo col suo programma che non esitiamo a pubblicare (subito) nella rubrica del Software Amiga, siamo certi che corre ad almeno un MICS (attenzione alla C in luogo della P). Ovviamente l'acronimo è spudoratamente inventato al volo per iscrivere questa introduzione, e «starebbe a significare» MI-lioni di C-ellule al S-econdo. Cellule? Sì, il programma presentato parla del ben noto auto-gioco (il computer gioca da solo e noi stiamo a guardare e/o a divertirci) LIFE già ampiamente trattato dal buon Corrado in IntelliGIOCHI (numeri 37 e seguenti... ai quali vi rimandiamo se non conoscete di già il problema). Buon divertimento! ■

Life

di Paolo Russo - Muggia (TS)

Il tema, Life, è un classico dell'informatica che resterà tale forse in eterno; nonostante ciò, e nonostante il fatto che un programma di questo genere per Amiga non sia ancora apparso sulle pagine di MC, ritengo che non avrebbe molto senso proporre per un computer così innovativo e potente un software che non avesse qualche caratteristica capace di distinguerlo da tutti gli altri programmi riguardanti Life scritti negli ultimi anni. Il programma che vi presento possiede una velocità di calcolo fenomenale, al punto da farmi ritenere che non sia mai esistito per un qualunque altro personal un programma di Life veloce anche solo la metà di questo. Il programma possiede tutte le funzioni di base che un buon Life credo debba avere, consentendo di ingrandire zone del piano, di richiamare configurazioni di cellule preventivamente definite e di studiare le proprietà statistiche delle grosse popolazioni; naturalmente è stata adotta-

ta la tecnica dei menu a discesa e del mouse, assolutamente obbligatoria su un computer come l'Amiga.

In realtà il programma è una dimostrazione della potenza del Blitter ed è stato scritto come esercitazione sull'uso dello stesso: ritengo quindi che tale programma possa interessare chiunque possieda un Amiga e desideri sfruttarne a fondo l'hardware.

Disgraziatamente la lunghezza del listato ne impedisce comunque la pubblicazione.

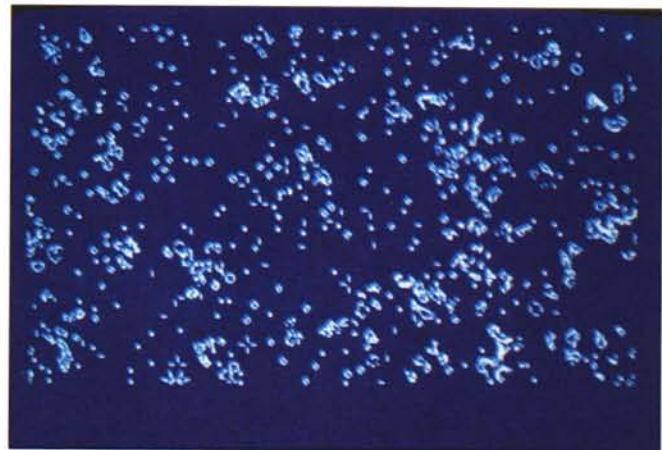
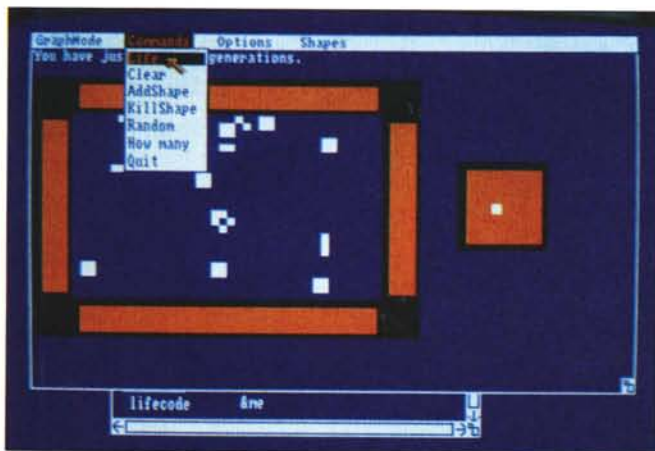
Devo dire che il classico Life, visto dall'alto della velocità consentita dal programma, ha proprio tutto un altro aspetto; le timide diffusioni di cellule diventano vere esplosioni; è facile notare dei fenomeni difficilmente percepibili a bassa velocità o studiarne agevolmente altri che richiederebbero giornate intere di calcolo; ho persino effettuato qualche piccola scoperta che penso potrà interessare l'eclettico Giustozzi (e chiunque altro si sia occupato di Life).

Sono certo che l'argomento «uso del Blitter in pratica» riscuoterebbe un discreto successo presso tutte quelle

persone che hanno acquistato un'Amiga per la potenza del suo hardware (e per quale altro motivo si può scegliere un Amiga?) e si trovano adesso nei guai perché, pur essendo buoni programmatori, non sanno come fare per utilizzare tale potenza. Ma facciamo un passo indietro...

Che cosa determina il successo di un computer? Un nutrito complesso di cause, naturalmente, ma le due principali hanno un nome: hardware e software. Molti considerano quest'ultimo il fattore predominante, citando l'Apple o l'IBM come esempi di computer la cui fortuna derivò e deriva tuttora dall'abbondanza del software; non ho mai udito da queste stesse persone la spiegazione di come queste macchine si siano affermate quando ancora il software non l'avevano, e perché mai la gente abbia scritto più software per

È disponibile, presso la redazione, il disco con il programma pubblicato in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 265.



Selezionando dal Menu a Discesa il comando LIFE... vedremo «partire» le nostre cellule.

queste macchine che per le altre. Si potrebbe rispondere che il buon hardware stimola il buon software, e questo è vero, ma non è tutto. In realtà il successo di un computer dipende molto dal suo rapporto prestazioni/prezzo, dove per prestazioni si intende ciò che la macchina può dare in pratica; la qual cosa dipende dall'hardware ma anche da come esso viene sfruttato. Da cosa a sua volta dipende il maggiore o minore sfruttamento dell'hardware? Evidentemente dal grado di diffusione delle informazioni che descrivono come usare al meglio tale hardware: in altre parole, i trucchi del mestiere. Infatti l'Apple si è affermato in un periodo in cui era l'unico personal, e se la gente parlava di personal parlava principalmente di Apple.

Presto si fece viva tutta una schiera di persone che illustrarono come superare quella tale limitazione, come ottenere quel tale risultato, come inserire testo nella pagina grafica, come bloccare un listato con dei POKE ben piazzati e tante altre cosette interessanti scoperte o inventate da quegli strani esseri metà uomini metà byte chiamati hacker. Tutti questi apporti hanno contribuito non solo alla creazione di buoni programmi da parte delle software house, costituite in buona parte dagli stessi hacker, ma anche alla soddisfazione di tutti i programmatori hobbisti che apprendevano, sfruttavano e tramandavano i trucchi più efficaci. «Cosa ha a che vedere tutto questo con l'Amiga?» si chiederà qualcuno spazientito; ebbene, è presto detto: che fine hanno fatto gli amigosi hacker? La Commodore sembra aver

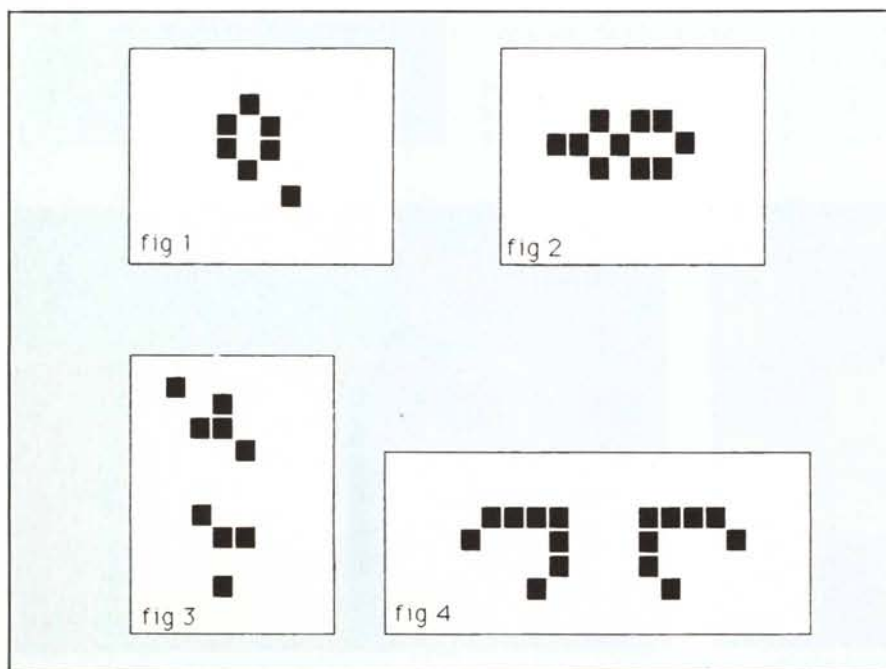
suddiviso tutti gli acquirenti di Amiga in «users» e «developers»: questi ultimi sono le software house che possono permettersi il lusso di procurarsi l'indispensabile serie completa dei manuali sull'Amiga, anche se forse non possono permettersi i lunghi mesi di tempo necessari al loro studio anche superficiale (tremila pagine non sono poche). Gli user sono tutti gli altri, che a quanto sembra nelle intenzioni della Commodore non dovrebbero mai toccare linguaggi di programmazione, fatta eccezione per la possibilità di trastullarsi con l'AmigaBasic che, non consentendo di accedere alle vere risorse del computer se non assai superficialmente, è una specie di linguaggio giocattolo. Posso prevedere il seguente itinerario per il programmatore medio: acquisto di uno o due manuali, loro lettura, incomprendimento dovuta ai continui rimandi agli altri manuali nonché alla macchinosità intrinseca, scoraggiamento, accantonamento dei manuali, delusione ed eventualmente acquisto di un ST, tiè!

Posso citare il mio caso. Abituato da sempre a smanettare in Basic e soprattutto Assembler su Spectrum e QL, appassionato di grafica in tempo reale, fermamente deciso a trarre il massimo da ogni possibile hardware, ho scelto un Amiga convinto di procurarmi un biglietto per il paradiso. Per prima cosa ho espresso il desiderio di vedere i famosi quattromilantantasei colori sullo schermo; inevitabile disappunto: il modo HAM non è attivabile da Basic nemmeno con i POKE più audaci, il manuale AmigaBasic non sa nemmeno cosa sia. Passo quin-

di al mio caro Assembler e trascorro una settimana tentando di capire cos'è una libreria, come la si utilizza, cosa significano tutti quei CALLLIB, XLIB e LINKLIB che infestano i listati, prima di accorgermi che si tratta di macro, e cosa sia un linker e quali librerie debbano essere linkate. Infine mi rendo conto che per predisporre uno schermo in modo HAM sono necessarie trentacinque (!) chiamate alla ROM; su qualunque altro computer sarebbe bastato pokare qualche locazione. È curioso notare come gli autori dei firmware siano convinti che il programmatore non debba perdere tempo e pazienza studiando come far funzionare l'hardware, che deve invece essere gestito dal S.O., e non si curino delle difficoltà di utilizzo dell'assai più complesso firmware. Il mio programma HAM1 è una delirante e interminabile sequenza di XLIB, CALLLIB e passaggi di parametri in registri; edito, assemblato, corretto, riassembleo, linko e l'inevitabile risultato è uno splendido Guru Meditation. Perdo del tutto la pazienza, decido di abolire il linker e creo una piccola libreria di macro che mi consente di richiamare le routine di sistema come fossero procedure di un linguaggio evoluto; il listato magicamente si dimezza e funziona persino.

Poi decido di usare il Blitter; è stato spesso detto che la grafica dell'Amiga è innovativa per la sua velocità più che per la risoluzione o i colori. Studio quindi l'Hardware Manual che spiega come usare il Blitter, dilungandosi in particolare modo sulle cose semplici e sorvolando o tacendo del tutto sui dettagli più importanti (in quale direzione avviene lo shift? Si tratta di uno shift che introduce zeri? Il riporto proveniente dall'ultima word finisce perso o viene scritto in un'ulteriore word? Il mascheramento della sorgente A avviene prima o dopo lo shift? Il fill hardware si applica alla sorgente A o al risultato? E potrei continuare). «Non fa niente», mi dico, «se riesco ad attivare il Blitter tutte queste cose le scoprirò da solo», come infatti è avvenuto. Per sapere come usare OwnBlitter(), DisownBlitter() e WaitBlit() devo ricorrere al ROM Kernel Manual: Libraries and Devices, che non dice molto sul come aprire la libreria che contiene tali routine. Basta avere il ROM Kernel Manual: Exec per chiarire ogni dubbio in proposito.

Ma se il Blitter blitta, lo fa nella pagina grafica; dove si trovano i bitplane della pagina grafica del Basic? Niente paura: la funzione WINDOW(7) ritorna un puntatore alle strutture di Intuition, quindi è sufficiente consultare il relativo manuale...



Se vi siete messi le mani nei capelli, niente paura... era tutto calcolato! Con l'augurio di ritornare presto sul tema, per spiegarvi meglio tutto, passiamo ora al programma.

Il programma LIFE

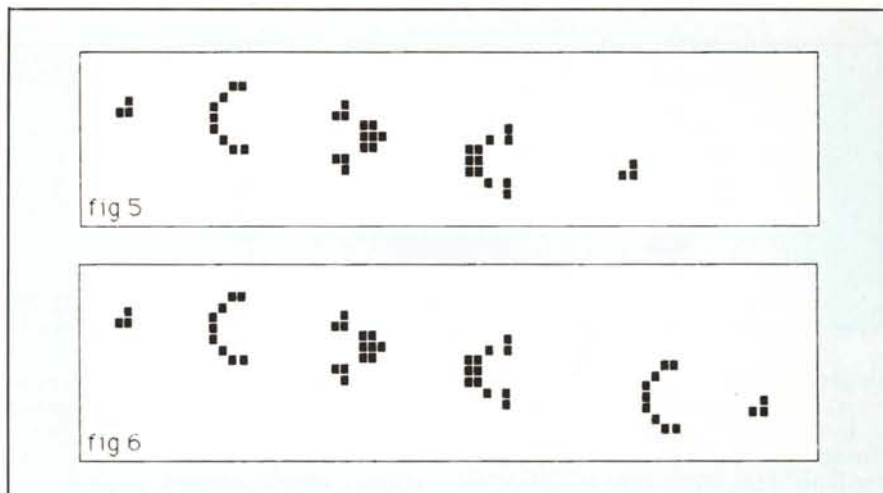
Come già preannunciato, questo non è un Life come tutti gli altri: possiede alcune caratteristiche che lo rendono interessante. Procediamo con ordine: quanto può essere veloce un programma di Life? Se realizzato in Basic potrebbe a occhio e croce ricalcolare un centinaio di cellule al secondo; in Pascal siamo attorno al migliaio, credo; in C siamo, penso, nel range $1000 \div 10000$; in Assembler ritengo si possa arrivare a 50000 (a suo tempo ho ottenuto 25000 con uno Spectrum, l'Amiga dovrebbe almeno raddoppiare questa cifra). È possibile spingersi oltre? Non molto, già a 50000 cellule/s il povero 68000 ha solo 20 μ s di tempo per cellula: abbastanza per eseguire una trentina di istruzioni, non di più. Ma allora come fa il qui presente programma a ricalcolare un milione di cellule al secondo, cioè una cellula in 1 μ s?

Il trucco c'è: basta usare il Blitter per calcolare la nuova generazione. «Un momento», penserà qualcuno, «come può un coprocessore grafico eseguire un calcolo numerico e per giunta venti volte più in fretta del 68000?». In effetti il Blitter, oltre al tracciamento di linee, può eseguire la seguente operazione, utile per le animazioni: prelevare tre zone rettangolari di memoria, eseguire su di esse una operazione logica a tre variabili e scrivere il risultato in una quarta zona rettangolare delle stesse dimensioni di ognuno dei tre operandi. Per «zona rettangolare» si intende una serie di n word (16 bit) consecutive, poi un intervallo lungo k word, poi altre n word, un altro intervallo lungo k e così via per m volte.

Il nome «zona rettangolare» deriva dal fatto che se disegnate un rettangolo pieno sullo schermo i pixel alterati, in ogni singolo piano di bit, si trovano in una serie di word di memoria disposte come sopra indicato. Il Blitter consente di gestire queste zone come degli array (bidimensionali) di singoli bit, più che di word; se indichiamo con A, B e C le tre zone «operando», con D la zona «risultato» e con x(i,j) il generico bit della zona X (A, B, C o D), allora si può dire che il Blitter esegue l'operazione

$$d(i,j) = LF(a(i,j), b(i,j), c(i,j))$$

in parallelo per ogni i e per ogni j. Con LF() ho indicato una qualunque



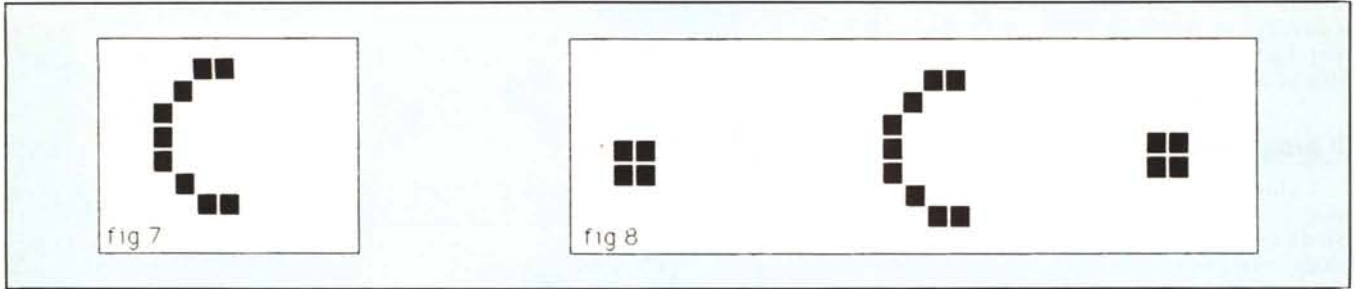
funzione logica di tre variabili (consultate se necessario MC Microfacile, nn. 41-42), la cui tabella di verità può essere scelta dal programmatore. Guarda caso, il piano di Life può essere pensato come un array di bit, e lo stato di ogni bit dipende dallo stato dello stesso bit e di altri otto confinanti alla generazione precedente; si tratta quindi di una funzione logica di nove variabili, non direttamente calcolabile dal Blitter, ma scomponibile in una sequenza di operazioni a tre variabili direttamente eseguibili. Ho trascorso un pomeriggio a minimizzare il numero di operazioni richieste e sono riuscito a ridurle a dieci. A questo punto bastava soltanto (!) pilotare convenientemente il Blitter. Accidenti al sistema operativo, le risorse del sistema bisogna strappargliele a viva forza! Mi pareva quasi di sentirlo urlare quando mi sono appropriato del Blitter invocando OwnBlitter() (cosa questa scongiatissima dai manuali, perché né il S.O. né gli altri eventuali task possono più usare il Blitter; la procedura ufficiale consigliata è di una complessità aberrante, di una scomodità atroce e non è neppure spiegata bene).

Devo dire che mi ero sempre chiesto perché mai per l'Amiga non ci fosse molto software capace di sfruttare a fondo l'hardware; ora lo so, il perfido firmware con i suoi manuali (ROM Kernel vari) ha di certo disgustato gli hacker più audaci.

Il programma può gestire 320×200 cellule a 16 generazioni al secondo, o una popolazione più piccola (160×100 , 80×50) a velocità maggiore; ho quindi risolto il mistero della sorte del «piccolo mostro a cinque cellule» che Conway ha studiato per quattrocentottanta generazioni senza capirci niente e posso assicurarvi che dopo circa milleduecento generazioni il tutto si stabilizza dopo aver lanciato sei alianti in tre direzioni diverse e do-

po aver invaso un'area di circa 150×90 . In genere il piano di Life viene gestito come la superficie di un toroide: tradizione vuole che sia così, Blitter non vuole. In questo programma ciò che esce da destra rientra a sinistra, ma una posizione più in basso e ciò che esce dall'alto o dal basso non rientra proprio. Un'altra caratteristica non standard (ma opzionale, e dalla storia curiosa: in origine era un bug) consente di trasformare il bordo inferiore in un mare turbolento da cui si alzano fiamme di cellule e alianti (stupiti? L'emersione di alianti dal caos è assai frequente, al contrario delle «astronavi leggere», veramente rarissime). Esistono due schermi; quello principale contiene i menu e la finestra di editing, all'interno della quale si può osservare una zona del piano di Life opportunamente ingrandita, mentre quello secondario è il piano di Life vero e proprio.

Si può passare allo schermo secondario premendo F2 e tornare a quello principale con F1; si possono disegnare configurazioni di cellule lavorando indifferentemente in entrambi gli schermi, ma se si usa una risoluzione elevata è più comoda la finestra di editing dello schermo principale. Per tracciare una singola cellula basta puntare e clickare con il mouse come in un qualunque tool grafico; è naturalmente possibile definire come «pennello» qualcosa di più complesso di una singola cellula. Il menu Shapes contiene appunto alcune configurazioni predefinite; è interessante il fatto che tale menu è ampliabile con i comandi AddShape e KillShape, che consentono di definire nuove configurazioni. Le shape possono essere ruotate e invertite. Premendo la barra dello spazio il programma calcola una singola generazione; il comando Life al contrario seleziona automaticamente lo schermo secondario e continua a



mostrare nuove generazioni finché non si clicca il mouse.

Il comando Random genera una popolazione casuale di data densità mentre HowMany conta tutte le cellule del piano.

Devo dire che mi capita spesso, dopo aver studiato per un po' una certa popolazione, di fermare il comando Life clickando il mouse e di sentirmi dire dal programma che ho osservato qualcosa come ottomila generazioni; ah, quando penso al povero Conway ed al suo PDP-7...

Alcune scoperte

Grazie al mio programma ho effettuato diverse scoperte, alcune delle quali piuttosto interessanti. Innanzitutto lo studio statistico delle popolazioni casuali basato sulla densità non ha molto senso: non è infatti tanto importante sapere quante cellule si trovano in una data popolazione bensì come sono disposte. In altri termini il «piccolo mostro a cinque cellule» dallo sviluppo imprevedibile non è affatto un caso isolato; piccole popolazioni di aspetto assai simile evolvono in modi sorprendentemente diversi. Se in una popolazione moribonda si trova anche un solo «mostro», non necessariamente a cinque cellule, il suo destino può mutare radicalmente. Ad ogni modo sembra che la tendenza alla stabilizzazione sia maggiore di quella alla proliferazione; date ad una popolazione abbastanza tempo per decadere in una serie di forme stabili e ciò prima o poi avverrà. La densità asintotica è quasi sempre attorno al 3%.

Non è vero che gli alianti a cinque cellule sono di due tipi diversi, come affermato in MC numero 39; si tratta sempre dello stesso tipo numero che cambia forma ciclicamente.

L'emersione di alianti dal caos è statisticamente un fenomeno molto probabile; le astronavi leggere sono invece rarissime.

Ho scoperto il destino del «piccolo mostro» che Conway studiò per quattrocentoottanta generazioni; si stabi-

lizza dopo 1100 ÷ 1200 generazioni (no, non ho passato un pomeriggio intero a studiarlo; bastano quaranta secondi usando la risoluzione 160 × 100, oppure ottanta secondi passando alla risoluzione doppia) e dopo aver lanciato sei alianti.

Ho scoperto una serie di configurazioni interessanti: quella di figura 1 si evolve in un quadrifoglio, quella di figura 2 in uno strano oscillatore di periodo tre che ho battezzato «sole» (sempre che non ci abbia già pensato qualcun altro); la configurazione di figura 3 evolve in una lunga e curiosa serie di immagini alcune delle quali sembrano facce che fanno smorfie. Inoltre due astronavi leggere disposte come in figura 4 collidono trasformandosi in alianti.

Ho compreso il principio base di funzionamento del cannone ad alianti: accludo quindi un cannone che ne spara due alla volta e uno che ne spara tre (fig. 5 e fig. 6); l'estensione a n alianti è alquanto semplice. Il segreto sta nella struttura di figura 7, forse scoperta dal gruppo del MIT per puro caso, il cui comportamento è indubbiamente originale: si muove verso sinistra come una medusa, crea un favo e si volta verso destra, quindi si dirige a destra e vi deposita un altro favo, voltandosi nel contempo a sinistra; a questo punto tutto è tornato come prima, a parte i due favi, e la «medusa» si dirige nuovamente verso sinistra per crearvi un altro favo, ma entra in collisione con quello creato precedentemente e si distrugge. Come renderla stabile? Si può pensare di eliminare i favi appena vengono creati, facendoli collidere con qualcosa di stabile; in effetti ciò è possibile (fig. 8) e la struttura risultante diventa un oscillatore di periodo trenta. Domanda: è possibile disporre due «meduse» in modo che i favi che producono collidano l'un con l'altro dando vita ad un aliante? Risposta: sì, per la precisione uno dei due favi deve collidere con la parte terminale della «medusa» che sta per trasformarsi nell'altro favo, con uno sfasamento temporale tra le due «meduse» di cinque generazioni.

Domanda: come sono riusciti gli studiosi di Life a determinare l'esatto sfasamento spaziale e temporale tra le due «meduse» in modo da produrre l'aliante? Risposta: o hanno avuto una fortuna sfacciata o hanno provato tutte le combinazioni possibili. Quindi il cannone ad alianti non è altro che una coppia di «meduse» che non si assomigliano perché sfasate di cinque generazioni (venti, se consideriamo che sono anche rivolte l'una contro l'altra e che il tempo di «rivoltamento» ammonta a quindici generazioni), più due quadrati il cui compito è la distruzione dei favi generati verso l'esterno. Il mio doppio cannone ad alianti è quindi costituito da tre «meduse» che, associandosi a due a due, creano alianti e dai soliti due quadrati che eliminano i favi esterni. Naturalmente il triplo cannone è formato da quattro «meduse».

Ovviamente tutti i quadrati possono essere sostituiti da triplette di cellule disposte a elle, tanto alla prima generazione diventano quadrati.

Un'ultima curiosità: il comportamento del mio programma ai confini del piano non è standard, per motivi tecnici: in particolare ciò che esce dall'alto o dal basso sparisce. Ho notato che quando un'astronave leggera urtava il bordo superiore o inferiore diventava un aliante e mi sono rammaricato al pensiero che questo curioso fenomeno non avvenisse con un Life standard. In seguito tuttavia ho pensato che si poteva ricreare l'effetto con uno scontro simmetrico: infatti quando due configurazioni speculari che distano un numero dispari di posizioni l'una dall'altra entrano in collisione tutte le cellule disposte sull'asse di simmetria restano morte, perché per simmetria il loro numero di vicini vivi è sempre pari e quindi diverso da tre.

Dal momento che il confine superiore del piano simulato dal mio programma può essere visto come un'immaginaria fila di cellule perennemente morte, ne consegue che qualunque cosa urti contro un tale confine sperimenta qualcosa che è indistinguibile negli effetti da un urto con una configurazione simmetrica.

A.A. AMIGA

periferiche offresi

PRODUZIONE

D-Smart II drive esterno da 3.5" compatibile con tutta la gamma Amiga; possibilità di configurare il proprio sistema all'accensione.

Half-2-One espansione di memoria da 512 Kb con orologio per Amiga 500; montaggio interno.

MIDI-X interfaccia MIDI per Amiga 1000 e Amiga 500/2000

Kit hard disk (per A-2000 con scheda MS-DOS) kit per il montaggio dell'hard disk all'interno della Amiga 2000 senza rinunciare all'uso del secondo drive da 3.5" e senza compromettere l'uso degli slot.

Kit drive 3.5" (per Amiga 2000) drive da 3.5" da montare internamente.

Kit espansione 512 Kb (per Amiga 2000) kit di 16 chip da 256 Kbit con istruzioni per espandere di 512 Kb la memoria dell'Amiga 2000.

Cavo per TV-Monitor permette il collegamento di un comune TV-Monitor alla uscita RGB dell'Amiga.

IMPORTAZIONE

StarBoard2 2 Mb espansione di memoria da 2 Mb auto-configurante per Amiga 1000; permette l'installazione del modulo Multifunction. Disponibile anche in altri formati.

Multifunction modulo da installare sulla StarBoard2; comprende orologio con batteria tampone, controllo di parità, zoccolo per coprocessore matematico (68881), funzione di RAM disk protetta.

MouseTime orologio con batteria tampone per Amiga 1000 dotato di rimando della porta giochi.

Kit di montaggio per 68010 kit per la sostituzione del microprocessore 68000.

KickStart eliminator kit per il montaggio del KickStart su ROM; aumenta la RAM di 256 Kb.

FutureSound digitalizzatore audio prodotto dalla Applied Vision.

PerfectSound digitalizzatore audio prodotto dalla Sun Rize.

Digi-view digitalizzatore di immagini.

Genlock permette la miscelazione di immagini esterne con immagini generate da Amiga.

Tavoletta grafica Easy1 tavoletta grafica funzionante con qualsiasi software; può essere usata al posto del mouse.

Monitor Philips RGB analogico monitor a colori.

Monitor Eizo 3010 a fosfori persistenti monitor monocromatico a fosfori bianchi persistenti; elimina il flicker.

Jitter Rid schermo antiriflesso da applicare al monitor; diminuisce l'effetto del flicker.

Plotter Roland DXY 990 plotter formato A3 a otto colori con funzione di digitizer.

Plotter Roland DXY 980 plotter formato A3 a otto colori.

Stampanti Epson linea completa di stampanti dalle più economiche a matrice d'aghi, alla sofisticatissima laser.

BAESSO GRAFICA

Disponibili in stock
tutti i modelli Amiga

BYTEC

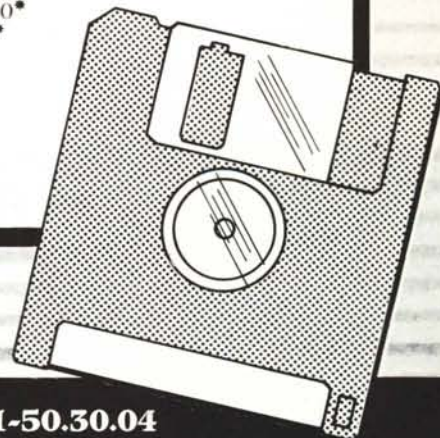
Azienda importatrice e produttrice di periferiche per Amiga 500-1000-2000, a prezzi imbattibili!!!

Ecco alcuni esempi:

| | | | |
|-----------------------------------|-----------|---|-----------|
| D-smart II singolo | £ 298.000 | PerfectSound | £ 180.000 |
| D-smart II doppio | £ 485.000 | Tavoletta grafica Easy1 | £ 840.000 |
| Half-2-one | £ 210.000 | Monitor Philips RGB analogico | £ 420.000 |
| MIDI-X per Amiga 500-1000-2000 | £ 94.000 | Monitor Eizo 3010 a fosfori persistenti | £ 220.000 |
| Kit hard disk per Amiga 2000 | £ 890.000 | Jitter Rid | £ 39.000 |
| Espansione 512 Kb per Amiga 2000 | £ 98.000 | Stampante Epson LX 800* | |
| Cavo video per TV-monitor | £ 35.000 | Stampante Epson EX 800* | |
| Cavo stampante per Amiga 500-2000 | £ 15.000 | Stampante Epson GQ 3500* | |
| StarBoard2 2 Mb | £ 990.000 | Kit drive 3.5" per Amiga 2000* | |
| StarBoard2 1 Mb | £ 710.000 | Kit di montaggio per 68010* | |
| StarBoard2 512 Kb | £ 630.000 | KickStart eliminator* | |
| Upper Deck 1 Mb | £ 310.000 | Digi-view | |
| Upper Deck 0 Kb | £ 120.000 | Genlock* | |
| Multifunction | £ 190.000 | Plotter Roland DXY 990* | |
| Kit 16 chip 256 Kbit (512 Kb) | £ 98.000 | Plotter Roland DXY 980* | |
| FutureSound | £ 300.000 | *Telefonare | |

TUTTI I PREZZI SONO IVA INCLUSA

Richiedete il nostro catalogo, vi invieremo in omaggio
un microdisk contenente preziose utility per il vostro Amiga



BYTEC - Via S. Secondo, 95 - 10128 Torino - Tel. (011) 59.25.51-50.30.04

Problema.

Un Personal che studi, lavori, organizzi, si diverta?

Non c'è casa senza Personal,
non c'è ufficio senza Personal, non c'è gioco senza Personal, non c'è scuola senza Personal.
Ecco perchè oggi un Personal deve saper fare mille cose ed essere vicino ed esauriente a
mille esigenze. Da quelle del professionista più impegnato a quelle
dello studente più attuale, dai piani casa della donna d'oggi, alla giovane creatività di tutte le età.

Soluzione.

Personal Acer 500: lo standard per la casa.



Personal Acer 500 **Multitech** rappresenta oggi il computer standard per la casa. Un computer che avvantaggia qualsiasi professione... compresa quella di padre. Vivere insieme un Personal Acer 500, infatti, è l'ottima occasione di instaurare un dialogo con i figli aperto e attuale. Per invitarli a studiare meglio, a imparare le lingue, a svolgere professionalmente il ruolo di studente.

Le Soluzioni SHR

L'informatica dal volto umano
Società del Gruppo FERRUZZI