

Spad. in Abb. Postale Gr. 11/70%

# Radio Elettronica & Computer

12 programmi  
per C64  
e C128

Anno XVIII - N.10 - Dicembre 1989 - L. 8.500

**VIDEOGIOCHI**  
TUTTE LE NOVITA'  
DAL MERCATO  
MONDIALE

**GRAFICA**  
NUOVE IDEE,  
GESTIONE DAL BASIC  
E ALTRO ANCORA

**SUONI**  
EFFETTI SPECIALI NEI  
TUOI PROGRAMMI

**AMIGA**  
ANCORA ESPANSIONE  
E ACCELERAZIONE

**UTILITY**  
COME VINCERE  
IL DUBBIO

Trasferimento  
automatico  
dei programmi  
da cassetta a disco

Gruppo Editoriale  
**JCE**

**TUTORIAL**  
Calcolo istantaneo  
dell'autopolizza



Tassa postale di campione allegato

Il mensile con disco programmi per C64 e C128

# COMMO DISK

Sped. in Abb. Postale Gr. III/70%

Anno III - Dicembre 1989 - N. 37 - L. 13.000

**ESPANSIONE**  
Il basic pronto  
da compilare

**PROGRAMMAZIONE**  
Gestione ottimale  
dell'interfaccia  
utente

**DRIVE**  
Directory  
facile  
nei tuoi  
programmi

**ADVENTURE**  
Usa i magici  
poteri di Feud!

**ANIMAZIONI**  
Effetti  
e supertesti  
in azione



**è in edicola**

Gruppo Editoriale  
**JCE**

# SE PAPA' NON MI COMPRA AMIGA 500 CON I SUOI 4096 COLORI, DIVENTO NERA.



"Per anni ho avuto le mani impegnate con matite e colori, nella convinzione che artisti non si



nasce ma si diventa. Poi, da un giorno all'altro e

senza sforzo, il disegno non ha più avuto segreti per me. Non è una favola, è la realtà di

**Amiga 500.**

Con **Amiga 500** carta e



matite non mi servono più, perché posso disegnare direttamente sullo schermo, posso modificare i co-



lori (ne ho ben 4096 a disposizione), posso ingrandire, ridurre, lavorare in prospettiva e in tridimensionale. Una soddisfazione così grande me la sono tolta a un prezzo che non mi ha mandato in rosso. Se poi un giorno volessi tradire il disegno per la musica, o la regia, o la narrativa, con

**Amiga 500** non avrei che l'imbarazzo della scelta. In più **Amiga 500** è Commodore, cioè un capolavoro di prestazioni, di sicurezza e di affidabilità. Adesso che vi ho dipinto una realtà così rosa, correte su-

bito a compravi **Ami-**

**go 500** nel più vicino Commodore Point".

Per informazioni, dalle 14.00 alle 18.00

Hot-Line **Ami-**  **go 500** nel più vicino Commodore Point".

Per sapere qual è il  più vicino a casa vostra, telefonate  alla 02/76123.1.

**Commodore**



ONLY  **AMIGA**  
MAKES IT POSSIBLE.

**Direttore responsabile**  
Paolo Romani

**Direttore Editoriale**  
**Area Informatica**  
Marinella Zetti

**Art director**  
Sergio Sironi

**Caporedattore**  
Fernando Zanini

**Responsabile grafico**  
**Desktop Publishing**  
Adelio Barcella

**Impaginazione elettronica**  
Denise De Matteis

**Segretaria di redazione**  
Alessandra Marini

**Collaboratori**  
Paolo Gussoni, Giorgio Caironi

**Testi, Programmi, Fotografie e Disegni**  
Riproduzione vietata. Copyright.  
Qualsiasi genere di materiale inviato in Redazione,  
anche se non pubblicato non verrà in nessun caso  
restituito.

**RadioELETTRONICA&COMPUTER**  
Rivista mensile, una copia L. 8.500, numeri arretrati  
lire 13.000 cadauno.  
Pubblicazione mensile registrata presso il  
Tribunale di Monza n. 679 del 28/11/88.

**Fotolito:** Bassoli - Milano.

**Stampa:** GEMM Grafica srl, Paderno Dugnano  
(MI).

**Difusione:** Concessionario esclusivo per l'Italia  
A.&G. Marco SpA, Via Forzezza 27 - 20126  
Milano. Spedizione in abb. post. gruppo III/70.

**Abbonamenti:** Annuale L. 64.000, estero  
L. 130.000.

RadioELETTRONICA & COMPUTER è titolare in  
esclusiva per l'Italia dei testi e dei progetti di Radio Plans  
e Electronique Pratique, periodici  
del gruppo Sociétés Parisiennes d'Édition.



**Gruppo Editoriale JCE srl**  
*Sede legale, Direzione, Redazione,  
Amministrazione*

Via Ferri 6 - 20092 Cinisello Balsamo (MI)  
Tel. 02/669025.1 - Telex 362376 JCE MIL I -  
Telefax 61.27.620 - 66.010.353

**Direzione Amministrativa:** Walter Buzzavo

**Pubblicità e Marketing**  
Gruppo Editoriale JCE - Divisione Pubblicità  
Via Ferri 6 - 20092 Cinisello Balsamo (MI)  
Tel. 02/66025.1

**Direttore Pubblicità:** Giuseppe Tiani

**Responsabile Marketing:** Daniela Morandi

**Concessionario esclusivo per Roma,  
Lazio e centro sud:**  
UNION MEDIA srl - Via C. Fracassini, 18  
00198 Roma - Tel. 06/3215434 (13 linee R.A.)  
Telex 630206 UNION I - Telefax 06/3215678

**Abbonamenti:** Le richieste di informazioni sugli  
abbonamenti in corso si ricevono per telefono tutti i giorni  
lavorativi dalle ore 9 alle 12. Tel. 02/66025311 - 66025338

I versamenti vanno indirizzati a:  
Gruppo Editoriale JCE srl, Via Ferri 6  
20092 Cinisello Balsamo (MI), mediante l' emissione di  
assegno circolare, cartolina vaglia o utilizzando il  
c.c.p. n. 351205. Per i cambi di indirizzo allegare  
alla comunicazione l'importo di L. 3.000, anche in  
francobolli,  
e indicare insieme al nuovo anche il vecchio indirizzo.



**Elettronica  
& Computer**

## 24 ALLA SCOPERTA DEI PIXEL

*Dedicato ai principianti del Basic, questo articolo cerca di far luce sui problemi relativi alla gestione dell'alta risoluzione del Commodore 64*



**PROGRAMMI SU CASSETTA**

## 13 SINCERO COME UN COMPUTER

*Il programma che insegna, questa volta si chiama Renault e vi dice in un attimo e con esattezza il costo della polizza assicurativa obbligatoria della vostra auto.*

## 18 IN SOGNO ANCORA SULL'ISOLA DEL TESORO

*Logo. In questa puntata si parlerà ancora di avventura che cercheremo di perfezionare razionalizzando la struttura e la logica del programma*

## 31 L'INSOSTENIBILE PESANTEZZA DEL DUBBIO

*Con l'utility che pubblichiamo, non solo riuscirete a prendere decisioni con cognizione di causa, ma imparerete anche un metodo d'analisi applicabile a qualsiasi problema*

## 36 ATTENTI A QUESTI DUE

*Due soli tips questa volta. Ma che tips! Le due mega routine proposte passeranno sicuramente alla storia perché dopo averle viste all'opera vi chiederete se non state sognando*

# SOMMARIO

N° 10 - Dicembre 1989

## 40 LO SPRINT FINALE

*Ultima e densissima puntata di Ultra Basic. Sono di scena le potentissime istruzioni per la gestione dell'alta risoluzione, del Sid e delle innumerevoli funzioni di aiuto alla programmazione*

## 46 GRAFICA CON L'ASSEMBLER

*In questa puntata studiamo nei particolari il programma Finestra Hardware visto sui due numeri precedenti analizzandone, routine per routine, il cuore in linguaggio macchina*

## 52 IL CHIP CHE FA CRASH!

*Questa volta vi insegnamo come applicare qualsiasi tipo di effetto sonoro ai vostri programmi con l'aiuto dell'ottimo chip Sid del C54*

**AMIGA**

## 58 L'URAGANO SUL MARE DI RAM

*Questo mese la nostra attenzione è rivolta a due formidabili schede per Amiga 2000: l'espansione di memoria Hardital SuperOtto e la scheda Hurricane dotata del potente microprocessore 68020 con coprocessore matematico 68881*

### Rubriche:

**Software news**  
pag. 6

**Cosa, Come, Quanto**  
pag. 64

**Posta**  
pag. 66

### Caricate così i programmi della cassetta allegata

*Riavvolgete il nastro e digitate **LOAD** seguito da **RETURN** sulla tastiera del C64 e **PLAY** sul registratore. Verrà caricato il programma di presentazione con il menù dei programmi. Digitate **RUN** seguito dalla pressione del tasto **RETURN**. Terminata la presentazione, per caricare uno qualsiasi dei programmi è sufficiente digitare: **LOAD "NOME PROGRAMMA"** seguito dalla pressione del tasto **RETURN**.*

Associato al

**CSST**

Consorzio  
Stampa  
Specializzata  
Tecnica

**USPI**

Mensile associato  
all'USPI  
Unione Stampa  
Periodica Italiana

## Geos 2.0

Com'è noto presso tutti gli utenti di Commodore 64, il sistema operativo a finestre e icone non è un'esclusiva di Macintosh e di Amiga: anche il piccolo C64 dispone di un simile sistema operativo. Avete capito che vi stiamo parlando di Geos. Non intendiamo, tuttavia, parlarvi di uno dei potenti pacchetti Geos, bensì della nuova versione del sistema operativo stesso: Geos 2.0.

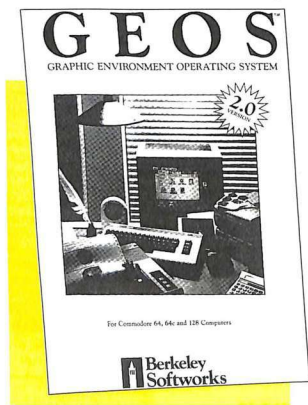
Spendiamo due parole per spiegare cosa significa disporre di un nuovo sistema operativo per un computer. Il sistema operativo standard del C64 (si chiama Kernel ed è contenuto in una Rom da 8 Kb) è molto semplice: all'accensione il computer si dispone a lavorare in Basic, e per caricare un qualsiasi programma basta usare il comando Load. La gestione dello schermo e di qual-

pa fosse fisso non sarebbe possibile creare stampanti più potenti dello standard. Per controllare in modo efficace una stampante è opportuno creare le routine di controllo specifiche per quel modello. È altresì ovvio che se ogni programma applicativo dovesse avere le proprie routine di controllo per ogni tipo di stampante, il lavoro da svolgere per creare un programma completo sarebbe davvero grande. Un sistema operativo completo e potente come quello di Macintosh o di Amiga (e non come l'antiquato Ms-Dos) dispone dei cosiddetti driver (routine di controllo) per tutte le stampanti più diffuse, e l'utente non deve far altro che scegliere una volta per tutte il driver per la sua stampante. Quando un software applicativo deve stampare una pagina non si deve preoccupare di quali codici inviare alla stampante per controllarla correttamente, gli basta invece caricare il driver messaggi a disposizione dal sistema operativo.

La stessa cosa è valida per la gestione delle finestre video: se il sistema operativo comprende le routine di gestione delle finestre e dei menù a discesa (Amiga, per esempio, dispone della potentissima interfaccia grafica Intuition), ogni programma può utilizzare tali routine e non c'è bisogno che ogni programmatore si scriva le proprie.

Il vantaggio naturalmente non è solo per il programmatore, anzi, è soprattutto per l'utente, per due diversi motivi. Innanzitutto poiché queste routine sono caratteristiche del sistema operativo, sono comuni a tutti i programmi, pertanto diventa molto facile imparare a disporre di tutto il software in circolazione senza dover ricorrere continuamente al manuale, inoltre man mano che il sistema operativo evolve e queste routine vengono aggiornate, automaticamente si rinnova il proprio software. Per esempio Geos 2.0 dispone di nuovi driver per stampanti e tali driver, possono essere utilizzati anche con i programmi Geos più datati.

Questo è solo un aspetto dei vantaggi che comporta un sistema operativo più avanzato. In realtà Geos è molto di più, poiché rappresenta un nuovo modo di utilizzare il C64, più moderno ed efficiente. La struttura a finestre, icone e menù a discesa, si sta sempre più diffondendo nel mondo dell'informatica, ed è destinata non solo agli utenti meno esperti per facilitarne il lavoro con un sistema più intuitivo, ma anche agli utenti più



siasi periferica resta affidata al programma. Kernel può solo controllare l'accesso al registratore o ai drive, ma interviene pochissimo durante l'esecuzione di un programma.

Un sistema operativo più complesso, invece, può raccogliere tutte le operazioni più comuni e fornire a tutti i programmi il proprio supporto. Spieghiamoci più dettagliatamente con un esempio. Quasi tutti i programmi applicativi richiedono l'accesso alla stampante. Le stampanti non sono tutte uguali, infatti, se il set di comandi di stam-

esperti per permettere loro di sfruttare al meglio le possibilità offerte da un computer.

Il C64, in realtà, presenta alcuni problemi perché un sistema operativo molto complesso richiede un microprocessore molto potente e floppy disk molto veloci, o meglio uno hard disk. Proprio per questo motivo riteniamo che Geos sia un prodotto eccezionale, infatti i programmatori della Berkeley Softworks sono riusciti a creare un sistema operativo del tipo appena descritto, che sfrutta in modo eccezionale il C64, nascondendone alcune carenze.

A questo scopo è stato necessario scrivere routine ottimizzate per la gestione delle finestre e soprattutto, sono state riscritte interamente le routine che controllano l'accesso ai floppy disk drive. Geos fa un uso molto intenso dei floppy disk, ma, grazie alle sue particolari routine, il C64 non subisce particolari rallentamenti. Dopo questa lunga descrizione generica delle possibilità di Geos passiamo a fornirvi qualche particolare sulla versione 2.0. Esistono due versioni di Geos 2.0, una è stata progettata specificamente per il C128, mentre l'altra può essere utilizzata dal C64 oppure dal C128 in modo 64.

La versione per C128 è più completa e permette di sfruttare al meglio alcune possibilità offerte dal più potente computer, soprattutto consente di utilizzare il modo a 80 colonne, preziosissimo per il word processing se si ama il vero Wysiwyg. L'impostazione generale comunque è identica.

Geos 2.0 è un pacchetto software di dimensioni inconsuete per il C64: nella scatola troviamo ben quattro dischetti, un grosso manuale, un foglio di acetato per utilizzare GeoPaint (il programma grafico della serie Geos) in modo ricalco e il foglio di registrazione dell'acquisto.

Una novità rispetto alla versione precedente di Geos consiste nel fatto che insieme al sistema operativo vengono venduti diversi programmi, e in particolare tutto il necessario per la grafica e il word processing. Inoltre quest'ultimo è davvero completo: oltre al word processor vero e proprio, la versione 2.1 di GeoWrite, troviamo GeoSpell, GeoLaser, GeoMerge, TextGrabber e Paint Drivers. La grafica, invece, è servita da GeoPaint. Questo significa che acquistando Geos vi trovate con un completo sistema di Desktop Publishing. Su un disco

si trova il software per il collegamento con Quantum Link, un Bbs a disposizione degli utenti Geos. In realtà Q-Link si trova negli Stati Uniti, pertanto non è di molto interesse per gli utenti europei. Non possiamo descrivervi nei dettagli Geos 2.0, pertanto ci limiteremo a farvi notare alcune novità della nuova versione. Il desktop è stato modificato: notiamo con piacere l'avvento del colore per le icone, che non è solo un'innovazione estetica, ma anche funzionale, poiché si può assegnare a ogni tipo di file un diverso colore. Nella parte in alto, a sinistra di desktop, ora c'è un orologio che può essere regolato da menù. Altri miglioramenti riguardano la gestione dei floppy disk: ora, dopo aver rimosso un file, è possibile recuperarlo selezionando l'opzione Undo Delete (questa opzione è davvero preziosa se siete distratti e vi capita di cancellare file importanti!), inoltre si può cancellare tutto il contenuto di un disco senza riformattarlo interamente come si procede con il comando DOS "N:NOMEDISCO". Davvero notevole il numero dei driver per le stampanti: ben 30! L'altro settore di Geos, che ha subito considerevoli modifiche, è il sistema di gestione dei dischi. Ora è possibile lavorare con ben tre device logici, chiamati con spreco di fantasia A, B e C. Ciascuno di questi device può essere assegnato a una Ram Disk, un 1541, un 1571 o (finalmente!) a un 1581. Naturalmente la Ram Disk è disponibile solo se si possiede l'espansione di memoria da 256 Kilobyte a 1764.

Non ci stancheremo mai di sottolineare l'importanza di un buon manuale per un programma applicativo: il manuale di Geos 2.0 è la cigliolina sulla torta, poiché si tratta di una completa guida all'apprendimento e all'uso di Geos. Non fatevi spaventare dalla mole (oltre 300 pagine) perché Geos è un sistema molto intuitivo. Il manuale serve per imparare a sfruttare sempre più a fondo il proprio computer dotato di questo moderno sistema operativo.

Le conclusioni per programmi come questi sono più che scontate: già in passato eravamo entusiasti della potenza di Geos, un sistema operativo capace di scrollare di dosso dal C64 l'aria da giocattolo per conferirgli un aspetto serio e molto funzionale. Oggi Geos è ancora più potente e più efficiente, molte pecche delle precedenti versioni sono state cancellate e siamo di fronte a uno dei migliori prodotti mai creati per C64 e C128. Certa-

mente il costo di Geos 2.0 è molto elevato, decisamente inconsueto per gli utenti C64, tuttavia siamo certi che chi deciderà di acquistarlo non sprecherà una lira.

## Giochi

### • Indiana Jones and the last crusade

Già qualche tempo fa la Us Gold ci ha annunciato, con grande clamore, l'uscita sul mercato del nuovo programma della Lucasfilm Games: "Indiana Jones and the last crusade". Ora finalmente abbiamo avuto il piacere di vedere il programma, sia nella versione per Amiga, sia per C64. Come ormai è consuetudine per la nostra rivista, la descrizione del programma vale per entrambe le versioni, infatti ormai quasi ogni gioco nasce contemporaneamente per tutti i computer più diffusi sul mercato a opera della stessa software house, pertanto è inutile soffermarsi sulle differenze tra le varie versioni. Indiana Jones è un classico programma arcade multischermo che richiama molto da vicino la storia dell'omonimo film che sta mettendo successi in tutto il mondo. Voi controllate il mitico Indy, e il vostro scopo consiste nel superare quattro diverse prove. La prima prova è relativamente semplice: dovete percorrere le caverne in modo da trovare la Croce di Coronado.

Effettuato il recupero dovete cercare la fuga sul tetto di un treno. Una difficoltà consiste nel buio delle caverne: per vederli dovete procurarvi delle torce che troverete lungo il percorso.



Al secondo livello vi trovate nelle catacombe e dovete recuperare lo Scudo del Crociato. La difficoltà consiste nel trova-

re i passaggi giusti in modo da non perdersi nei cunicoli delle catacombe. Quando avete recuperato lo Scudo dovete fuggire scavalcando le mura del castello, facendo attenzione a evitare i fulmini perché, naturalmente, la notte è buia e tempestosa. Al terzo livello abbandonate le profondità della terra e vi portate sul dirigibile alla ricerca del Diario del Graal che vostro padre ha perduto. Per circolare avete bisogno dei lasciapassare, ma attenzione perché essi hanno breve durata e se rimanete senza suona l'allarme.

L'ultimo livello è il più drammatico: vostro padre, il dottor Jones Senior, è stato colpito e per salvarlo, dovete assolutamente raggiungere il Santo Graal entro un certo tempo. E proprio il tempo il vostro nemico in quest'ultima prova: se siete abbastanza freddi e coraggiosi potete farcela!

La descrizione del programma è senza dubbio invitante. Il programma, del resto, è sicuramente un prodotto di qualità: ottimo il sonoro, per quanto non sia nulla di eccezionale, ottima la grafica con lo scroll del fondale di gioco e una buona realizzazione della figura e dell'animazione dei diversi personaggi e ottima anche l'idea portante, anche se non è altro che la trascrizione del film; peraltro molto fedele.

### • Apb (All Points Bulletin)

"Comunicato a tutte le postazioni: spingi a fondo l'acceleratore e lancia la tua automobile a tutto gas attraverso le strade cittadine in una serie di allegri e spericolati inseguimenti." Così esordisce l'opuscolo che accompagna questo programma della Tengen. In effetti si tratta di una scanzonata versione delle vicende di una pattuglia automobilistica della polizia americana, quasi una parodia dei numerosi telefilm che spesso appaiono in televisione. (Chi non conosce "Chips" o l'infalibile tenente Hooker?)

I poliziotti in questione siete proprio voi e dovete fare del vostro meglio per arrestare un certo numero di criminali ogni giorno in modo da aumentare lo stipendio e non essere licenziati. Pattugliate le vie cittadine a bordo della vostra affidabile auto; per arrestare i malviventi è necessario azionare la sirena e speronare l'auto sulla quale essi cercano di scappare; questo metodo brutale, ma efficace può però presentare alcuni non trascurabili inconvenienti: presi dalla



foga dell'inseguimento potreste fare qualche vittima innocente fra i passanti; in questo caso vi verrà assegnato un punto di demerito. Se, giunti alla centrale, riuscirete a far confessare i criminali catturati prima che ritorni il capo non ci sarà più bisogno, per quel giorno, di continuare il pattugliamento, poiché vi verrà abbonato il restante numero di malviventi da arrestare. Potrete far uso dei soldi che avrete guadagnato con il vostro duro lavoro comprando accessori per l'auto o carburante. Se le entrate risultano scarse, dovrete far affidamento sulla fortuna raccogliendo i portafogli che di stratti passanti hanno smarrito.

A volte il comando centrale può comunicare la presenza nella vostra zona di criminali particolarmente pericolosi; se vorrete continuare a fare il vostro mestiere dovrete arrestarli a tutti i costi. Apb è un programma particolarmente divertente grazie soprattutto all'originalità con cui è trattato l'argomento: non più superuomini che con la loro forza sgominano in un sol colpo intere bande, ma un simpatico poliziotto che deve fare continuamente i conti più con il proprio portafoglio che con la criminalità.

La giocabilità è sostenuta dalle numerose trovate che sono disseminate per le strade in cui si svolge l'azione (negozi di accessori, portafogli, pompa di benzina, negozio per acquistare tempo supplementare eccetera). La grafica è semplice ma efficace.

#### • Dynamite Dux

Proseguiamo con Dynamite Dux, un arcade della Sega Enterprises a dir poco fantasioso. Ci troviamo in una stravagante città di papere. Il malvagio Achacha ha rapito Lucy e, Bin e Pin (solo i nomi sono tutto un programma!) vogliono liberarla. L'impresa si dimostra più difficile del previsto poiché Achacha ha molti amici che tentano in tutti i modi di ostacolare i nostri due eroi: maiali lottatori di sumo, cocodrilli pugili, gatti pattinatori, cani, topi e chi più né ha più né metta. In questo scenario da "Libro della giungla" di Walt Disney i nostri poveri Bin e Pin dovranno farsi largo a suon di sganassoni, incrementando la loro energia sbafandosi i dolci e gli hamburger che, ormai non ci stupiamo più di niente, sono disseminati in mezzo alla strada. Durante il cammino è facile incontrare ogni tipo d'arma, dai semplici idranti ai micidiali ba-

zooka. Più che per la dinamica di gioco, già vista in molti altri giochi da questo punto di vista anche più interessanti, Dynamite Dux colpisce per la fantasia dello scenario e la simpatia dei personaggi. L'azione può essere ingaggiata anche da un singolo giocatore, ma è sicuramente in doppio che il gioco offre le migliori chance di divertirsi. La grafica è ottima, basti osservare lo sprite del fuoco il cui volto umano diventa sempre più provato man mano che subisce i colpi dei paperi. Lo stile da cartoon del gioco è sottolineato dai fumetti che descrivono l'impatto dei pugni inferti dai protagonisti ai poveri malcapitati.

#### • Xenon 2 (Megablast)

Proseguiamo con un gioco esclusivamente per Amiga. Si tratta di una classica avventu-



*Dinamite Dux. I simpatici personaggi del gioco in azione.*

ra spaziale in cui il vostro Megablast, avveniristica navicella spaziale, vaga non nella solita galassia misteriosa, ma nientemeno, che nel tempo. Per spiegare il significato della vostra missione è necessario conoscere alcuni antecedenti: i periferici Xeniti sono da tempo in guerra con il vostro sistema solare e pur di non perdere, sono disposti ad autodistruggersi assieme a tutto il resto dell'universo; per ottenere ciò hanno pensato bene di innescare cinque bombe nell'arco della storia, dai tempi preistorici a quelli ultramoderni. Inutile dire che il vostro compito è quello di superare le difese nemiche, individuare ogni ordigno e renderlo innocuo. Ogni livello di gioco corrisponde a un preciso periodo della storia; alla fine di ogni schema vi imbatterete in un mostruoso essere che funge da guardiano della bomba. Esistono dei modi particolari per distruggerlo ed è compito vostro scoprirli. Se fosse tutto qui

Xenon 2 non sarebbe altro che un'ordinario gioco spaziale; invece, esso offre una varietà di trovate e di possibilità di gioco da far invidia anche al più completo programma del genere. Elenchiamone alcune. Innanzitutto è possibile raccogliere durante il volo una serie di simboli che permettono l'attivazione di armi particolari (proiettili laterali e smart bomb che distruggono tutto ciò che appare sullo schermo, potenza di fuoco aumentata per un certo periodo di tempo, integrazione della potenza protettiva degli scudi in dotazione alla navicella). Ma l'aspetto sicuramente più interessante consiste nella possibilità di compiere delle vere e proprie compravendite con le attrezzature in dotazione. Infatti, a metà di ogni livello, è collocato un negozio nella cui vetrina potete trovare gli strumenti più indicati per superare nel migliore dei modi il punto in cui vi trovate. In fondo alla vetrina del negozio appare l'entità del vostro conto in banca. Nel caso i liquidi non bastassero all'acquisto dell'ultimo modello di mina spaziale o di laser, sarà sufficiente rivendere a metà



La grafica stupefacente di Xenon II



prezzo parte dell'attrezzatura di cui già disponete. Il negoziante è una losca figura, per giunta alieno, ma è necessario fidarsi poiché senza la sua mercanzia è molto arduo arrivare indenni alla fine del percorso; inoltre egli (o esso, fate voi) è sempre gentilmente disposto, sotto pagamento naturalmente, a consigliarvi l'arma più indicata per la particolare situazione in cui vi trovate.

Si potrebbe parlare di molti altri aspetti interessanti del gioco, come la varietà di esseri ostili che popolano il paesaggio e che

presentano ognuno una caratteristica particolare che richiede un'appropriata tecnica per eliminarli; oppure si potrebbe continuare nella descrizione delle opportunità di gioco (acceleratore, elettropalla, autofire, possibilità di riprendere il gioco da dove si è stati distrutti e non dall'inizio), ma bisogna lasciare spazio anche agli altri programmi.

Ci limiteremo a dire che la grafica è ottima (basti pensare allo scroll in secondo piano che si muove a velocità diversa da quella del paesaggio) e che la musica sfrutta le potenzialità sonore di Amiga in tutte le sue sfumature (il brano di accompagnamento presenta addirittura delle parti ottenute con una voce digitalizzata. Xenon 2 (Megablast) è una produzione Image Works.

## Ultimissime novità software

### • Activision

Con l'avvicinarsi dei campionati mondiali i giochi di calcio stanno veramente proliferando. La Activision ci informa dell'uscita di Fighting Soccer. Si tratta di un programma che vi permette di competere con 11 squadre nazionali. Direttamente dalle sale giochi la Activision presenta la conversione del coin-op Wonderboy. Si tratta di un arcade molto ricco: il protagonista è il ragazzo delle meraviglie (wonderboy) alle prese con il fiammeggiante drago Meka. Finché il drago non è sconfitto la pace non può tornare a Wonderland. Durante l'avventura Wonderbot incontra una lunga serie di mostri pericolosi: anaconde, vampiri, scheletri sono all'ordine del giorno. Non potevano mancare i vari tesori da raccogliere per rinforzarsi e proteggersi, pozioni per recuperare le vite perse e scarponi alati per volare. La bellezza del gioco sta essenzialmente nella ricchezza delle armi, dei tesori e delle situazioni. Per gli appassionati di simulazione di volo ecco Bomber, un simulatore di un moderno caccia bombardiere: è possibile scegliere tra il Tornado, il classico F15, lo svedese Saab Viggen, il sovietico Mig 27 o il fantasma F4. Si tratta di un programma che unisce l'azione di combattimento alla simulazione. Anche questo programma appare ricco di opzioni, dalla scelta dell'aereo, alla scelta delle armi. Come Indiana Jones, anche Ghostbusters II è destinato a essere il titolo di un videogioco, oltre che di un film. Il

Xenon 2 non sarebbe altro che un'ordinario gioco spaziale; invece, esso offre una varietà di trovate e di possibilità di gioco da far invidia anche al più completo programma del genere. Elenchiamone alcune. Innanzitutto è possibile raccogliere durante il volo una serie di simboli che permettono l'attivazione di armi particolari (proiettili laterali e smart bomb che distruggono tutto ciò che appare sullo schermo, potenza di fuoco aumentata per un certo periodo di tempo, integrazione della potenza protettiva degli scudi in dotazione alla navicella). Ma l'aspetto sicuramente più interessante consiste nella possibilità di compiere delle vere e proprie compravendite con le attrezzature in dotazione. Infatti, a metà di ogni livello, è collocato un negozio nella cui vetrina potete trovare gli strumenti più indicati per superare nel migliore dei modi il punto in cui vi trovate. In fondo alla vetrina del negozio appare l'entità del vostro conto in banca. Nel caso i liquidi non bastassero all'acquisto dell'ultimo modello di mina spaziale o di laser, sarà sufficiente rivendere a metà



La grafica stupefacente di Xenon II



prezzo parte dell'attrezzatura di cui già disponete. Il negoziante è una losca figura, per giunta alieno, ma è necessario fidarsi poiché senza la sua mercanzia è molto arduo arrivare indenni alla fine del percorso; inoltre egli (o esso, fate voi) è sempre gentilmente disposto, sotto pagamento naturalmente, a consigliarvi l'arma più indicata per la particolare situazione in cui vi trovate.

Si potrebbe parlare di molti altri aspetti interessanti del gioco, come la varietà di esseri ostili che popolano il paesaggio e che

presentano ognuno una caratteristica particolare che richiede un'appropriata tecnica per eliminarli; oppure si potrebbe continuare nella descrizione delle opportunità di gioco (acceleratore, elettropalla, autofire, possibilità di riprendere il gioco da dove si è stati distrutti e non dall'inizio), ma bisogna lasciare spazio anche agli altri programmi.

Ci limiteremo a dire che la grafica è ottima (basti pensare allo scroll in secondo piano che si muove a velocità diversa da quella del paesaggio) e che la musica sfrutta le potenzialità sonore di Amiga in tutte le sue sfumature (il brano di accompagnamento presenta addirittura delle parti ottenute con una voce digitalizzata. Xenon 2 (Megablast) è una produzione Image Works.

## Ultimissime novità software

### • Activision

Con l'avvicinarsi dei campionati mondiali i giochi di calcio stanno veramente proliferando. La Activision ci informa dell'uscita di Fighting Soccer. Si tratta di un programma che vi permette di competere con 11 squadre nazionali. Direttamente dalle sale giochi la Activision presenta la conversione del coin-op Wonderboy. Si tratta di un arcade molto ricco: il protagonista è il ragazzo delle meraviglie (wonderboy) alle prese con il fiammeggiante drago Meka. Finché il drago non è sconfitto la pace non può tornare a Wonderland. Durante l'avventura Wonderbot incontra una lunga serie di mostri pericolosi: anaconde, vampiri, scheletri sono all'ordine del giorno. Non potevano mancare i vari tesori da raccogliere per rinforzarsi e proteggersi, pozioni per recuperare le vite perse e scarponi alati per volare. La bellezza del gioco sta essenzialmente nella ricchezza delle armi, dei tesori e delle situazioni. Per gli appassionati di simulazione di volo ecco Bomber, un simulatore di un moderno caccia bombardiere: è possibile scegliere tra il Tornado, il classico F15, lo svedese Saab Viggen, il sovietico Mig 27 o il fantasma F4. Si tratta di un programma che unisce l'azione di combattimento alla simulazione. Anche questo programma appare ricco di opzioni, dalla scelta dell'aereo, alla scelta delle armi. Come Indiana Jones, anche Ghostbusters II è destinato a essere il titolo di un videogioco, oltre che di un film. Il

programma dovrebbe essere pubblicato poco prima dell'uscita di questa rivista, contemporaneamente alla prima del film. Con Ghostbusters II appartenete alla nota squadra di acchiappafantasma e dovete affrontare le tre fasi della lotta alla nuova invasione di New York City. La Activision pensa che Ghostbusters II possa ripetere il successo del primo Ghostbusters che ha venduto oltre due milioni di copie. Galaxy Force è la nuova conversione Activision di un famoso coin-op. La versione casalinga di Galaxy Force dovrebbe mantenere inalterate le caratteristiche del coin-op: il classico shoot'em up con cinque livelli di gioco, numerosissimi nemici da disintegrare e soprattutto un'elevatissima giocabilità. Se siete fanatici dell'autofire Galaxy Force è per voi.

#### • Una compilation dalla Domark

La Domark ha recentemente immesso sul mercato la compilation della saga di "Guerre Stellari Parl", ovvero la raccolta dei tre programmi: "Star Wars", "The Empire Strikes Back" e "Return of the Jedi". Il primo, Star Wars, nacque due anni fa. Prendete il ruolo di Luke Skywalker in "Guerre Stellari" e dovete distruggere le forze di Darth Fener. The Empire Strikes Back è lo stadio successivo della vicenda: questa volta dovete guidare Luke contro gli Imperial Walkers. Return of the Jedi completa la storia. Dovete accompagnare la Principessa Leila attraverso la foresta di Endor fino al villaggio di Ewokrs e poi dovete raggiungere il reattore centrale di Deathstar.

#### • The software toolworks

Dopo tanti arcade passiamo a un programma di simulazione piuttosto complesso: The software toolworks presenta Life & Death, un incredibile programma di simulazione di chirurgo! Si tratta naturalmente di un programma destinato a chi sopporta la vista del sangue e di un tavolo operatorio: voi interpretate la parte di un famoso chirurgo in un ospedale e dovete svolgere un completo lavoro di cura: ordinare radiografie, ultrasuoni o esami del sangue, leggere i rapporti medici e se necessario impugnare il bisturi e... la vita del paziente è nelle vostre mani. In sala operatoria avete a disposizione oltre a tutti gli strumenti necessari, anche un monitor delle pulsazioni cardiache e delle altre funzioni vitali. Un pro-

gramma notevole: aspettiamo di vederlo per darvi istruzioni più dettagliate.

#### • Empire Software

Torniamo ancora al calcio: la Empire Software ha prodotto Gazzas Super Soccer. Per i meno informati di football britannico ricordiamo che Paul Gascoigne detto Gazza è un possente giocatore del Tottenham Hot Spurs, la plurititolata squadra londinese.

Gazza è uno dei calciatori più popolari in Inghilterra e stando alle dichiarazioni della Empire Gazzas Super Soccer è destinato a diventare uno dei più popolari programmi di calcio. Non abbiamo ancora visto il programma, tuttavia dalle dichiarazioni della Empire possiamo capire che si tratta di un programma piuttosto originale, che lascia molte possibilità di controllo al giocatore.

Contrariamente alla moda vigente inoltre



*Fighting Soccer. Due situazioni di gioco e la scheda del campionato.*

non esiste lo scrolling del campo, bensì un particolare sistema di raffigurazione del campo che dovrebbe privilegiare la tecnica e gli schemi agli effetti grafici. Vi parleremo sicuramente più dettagliatamente di questo programma non appena potremo vederlo: vi possiamo anticipare che probabilmente in Italia uscirà la versione italiana del gioco.

#### • Microprose

Terminiamo le nostre anticipazioni con un programma della software house regina delle simulazioni: la Microprose ha prodotto una simulazione di guerra a terra con carri armati: Tank Platoon. Conoscendo di fama questa software house ci aspettiamo un prodotto molto accurato e valido.

**Gianni Arioli  
Massimiliano Del Rio**

# ELECTRONICS PERFORMANCE

- PERMUTE
- RIPARAZIONI
- ASSISTENZA

OFFERTE SPECIALI • VENDITA PER CORRISPONDENZA • SCONTI RISERVATI AI RIVENDITORI

ARTICOLO	PREZZO	ARTICOLO	PREZZO
Amiga 500 con mouse e 3 dischi	739.000 *	Joystick Flashfire con 3 spari manuali	10.000
Commodore 64 New + Registratore + Penna Ottica, ecc.	339.000 *	Joystick Flashfire con 3 spari man + autofire	15.000
Commodore 128 + Registratore + Penna Ottica	350.000 *	Joystick Flashfire con autofire trasparente	19.000
Disk Drive compatibile x 64/128k	250.000 *	Joystick Flashfire con Microswitches B.	29.000
Disk Drive 1541 II	295.000 *	Joystick Albatros microswitches	49.000
Drive esterno x Amiga 500/2000 multi/discconn.	165.000 *	Joystick Flashfire x C16	15.000
Stampante Riteman x C 64/128K	390.000 *	Joystick Flashfire x C16	49.000
Stampante per Commodore 64/128k	325.000 *	Joystick x Amstrad con sdoppiatore D.	24.000
Stampante Commodore 1230 x C64 Amiga e PC	490.000 *	Joystick x Amstrad con sdoppiatore D Microsw	32.000
Stampante Epson x Amiga e Pc	395.000 *	Joystick Digitale a sfioramento	45.000
Stampante Star Lc 10 Colore x Amiga e Pc	495.000 *	Joystick WIZ Master	35.000
Monitor Monocromatico con audio (universale)	165.000 *	Joystick Sinclair Plus 2	24.000
Monitor colore Commodore 1802 x C64 e Amiga/Am	330.000 *	Paddles Controllers (coppia)	18.000
Monitor colore 1084 C x C64 Amiga, PC	520.000 *	Adattatore x Joystick C16	5.000
Monitor Philips x C64 Amiga PC	460.000 *	Mouse x C64 128K	60.000
TV Monitor Universale	580.000 *	Mouse x Amiga 500	95.000
Registratore Compat. x C64/128k	50.000 *	Mouse Elettronico X C64 128K Amiga	45.000
Registratore Commodore x C64/128k	68.000 *	Penna ottica x C64/128k cassetta o disco	35.000
Registratore 1531 x C16	68.000 *	Geos x C64/128K	49.000
Adattatore Registratore da C16 e C64	25.000	Regolatore Elettronico di testine reg.	19.000
Adattatore Duplicatore x registratore C16	18.000	Fast Disk (velocizzatore + copiatore C64)	45.000
Duplicatore x Registratore C64	18.000	Final Turbo IV New	79.000
Modulatore x Amiga 500	45.000 *	Power Cartridge	120.000
Espansione di Memoria 512K Amiga 500	250.000 *	Isepic Tape (sprotettore e + copiatore C64)	65.000
Videodigitalizzatore Amiga 500	200.000	Reflex Backup C64	49.000
Videon (digitalizzatore Amiga)	390.000	Turbo Dos	90.000
Video Gen Lock (miscelatore di immagini Amiga)	450.000 *	Speed Dos 1541 II	90.000
Interfaccia Musicale "Midi" x Amiga 500	120.000	Digitalizzatore x C64	60.000
Copricomputer Plexiglas Amiga 500	19.500	CartucciaCpm x C64	100.000
Copricomputer Plexiglas 128k	18.000	Rom x Stampante Mps 803	48.000
Copricomputer Plexiglas C64 New	16.000	Portacassette "Posso" 15 Pz.	15.000
Copricomputer Plexiglas C64 C16 Vic 20	14.000	Portadischetti 5" 1/4 "Posso"	38.000
Modulatore x Vic 20	35.000 *	Portadischetti 3" 1/2 "Posso"	38.000
Cavo Start x Amiga 500 Tv e Monitor	27.000 *	Portadischetti 3" 1/2 (cont 10 Pz.)	4.500
Cavo Start C64/128k/C16/Vic 20	19.000 *	Portadischetti 3" 1/2 (cont 25 Pz.)	18.000
Cavo Monitor C64/128K/C16	25.000	Portadischetti 3" 1/2 (cont 40 Pz.)	24.000
Cavo Monitor per Amiga 500	40.000	Portadischetti 3" 1/2 (cont 80 Pz.)	28.000
Cavo Monitor Colore x PC	59.000	Portadischetti 5" 1/4 (cont 10 Pz.)	4.500
Cavo Monitor videocomposto x PC	12.000	Portadischetti 5" 1/4 (cont 50 Pz.)	24.000
Cavo Monitor videocomposto x C64	14.000	Portadischetti 5" 1/4 (cont 100 Pz.)	28.000
Cavo TV/Computer mt. 1,5	6.500	Portacassette a valigetta 16 Posti	14.000
Cavo TV/Computer mt. 3	12.000	Portacassette a valigetta 36 Posti	18.000
Filtro antidiurbo x Computer	19.000	Portacassette Componibili (multibox)	3.500
Cavo 40/80 colonne x 128k anche Skart	25.000	Dischetti 3" 1/2 Df Dd Conf. 50 Pz. (1 Mega)	CAD 2.000
Cavo Seriale x Drive e Stampante	16.000	Dischetti 3" 1/2 Df Dd Conf. 20 Pz. (1 Mega)	CAD 2.500
Cavo Centronics	25.000	Dischetti 3" 1/2 Df Dd Sciolti (1 Mega)	CAD 3.000
Alimentatore Universale	28.000	Dischetti 5" 1/4 Df Dd Conf. 10 Pz.	CAD 1.000
Caricatore Batterie e Alimentatore	39.000	Dischetti 5" 1/4 Df Dd xc PC	CAD 1.500
Alimentatore x Atari 2600	35.000 *	Dischetti 5" 1/4 Df Dd 3M	CAD 6.000
Alimentatore x Commodore C16	35.000 *	Tagliadischetti in Acciaio	16.000
Alimentatore Commodore 64	45.000 *	Tagliadischetti in Plastica	12.000
Cassette Vergini x Computer da C 10 e C 120	da 1.000	Kit Puliscitistine x Drive	15.000
Videoregistratori e cassette video	da 1.000	Giochi Disco x C 64/128/Amiga/PC/Msx ecc. da L.	10.000
Kit Pulisci testine x Videoregistratore	19.000	Giochi Cassette x C64/MSX/C16/Atari ecc. da L.	9.000
	Telefonare	Disponibili integrati e ricambi per computer	Telefonare

\* I PREZZI INDICATI CON ASTERISCO SI INTENDONO + IVA 18%

## ELECTRONICS PERFORMANCE

Via S. Fruttuoso, 16/A - Monza (S. Fruttuoso) - Tel 039/744164



*Il programma che insegna, questa volta, si chiama Rc-auto e vi dice in un attimo e con esattezza il costo della polizza assicurativa obbligatoria della vostra auto. Chi conosce il Basic può listare il programma, apportare modifiche o scoprire i complessi meccanismi che gestiscono il programma e determinano il premio assicurativo*

## Sincero come un computer



Questo programma permette di calcolare, facilmente e in pochissimo tempo, l'esatto costo di una polizza Rc-auto (obbligatoria per tutti gli automobilisti) in base all'ultimo, generalizzato, ritocco apportato recentemente ai tariffari dalle compagnie di assicurazione che operano in Italia. La mediazione politico-tecnica del Cip (Comitato interministeriale prezzi) tra Ania (il raggruppamento delle va-



rie imprese assicuratrici) e associazioni degli utenti ha portato ad aumenti più contenuti rispetto alle richieste iniziali delle compagnie, ma non certo limitati: si va da un minimo del +3,6% a un massimo del +8,7% rispetto alle quote dei premi precedentemente a listino, con i consueti distinguo su zone di immatricolazione delle auto, cavalli fiscali tassati e classe di merito del guidatore (e/o dell'auto).

Un esempio: per una Fiat Uno 45 (potenza fiscale 12 cavalli) registrata con targa Roma si pagava, prima, sulla classe settima (quella in cui vengono inserite d'ufficio le automobili di nuova immatricolazione) un premio annuo di lire 444.100. Adesso occorre sborsare la cifra di lire 468.900 (cioè l'aumento relativo è del 5,6 per cento).

Qualsiasi automobilista sarà interessa-



to a sapere dal computer, oltre che dall'agente assicuratore di fiducia (o di sfiducia!), il giusto ammontare del premio da corrispondere annualmente per la Rc-auto, anche al fine di fare raffronti e verifiche: visto che i valori imposti per legge (quelli che anche il computer fornisce su video) devono intendersi come massimali, sarà facile verificare se un eventuale sconto promesso è stato mantenuto o se, addirittura, anche solo per sbaglio di calcolo, si è pagato più del massimale contemplato e imposto dalla legge.

Rc-Auto è insomma un utile strumento di lavoro ideale non solo per chiunque vo-

glia fare chiarezza su una materia tanto particolare come quella delle tariffe di assicurazione Rc-auto, ma anche per tutti coloro che, per ragioni professionali, devono determinare l'esatto costo del relativo premio massimale (gli stessi agenti assicuratori, per esempio).

D'altra parte con questo programma il numero di dati da inserire è limitato a tre specifiche soltanto perché, a fare tutti i calcoli, ci pensa il computer, che si preoccupa anche di verificare la veridicità delle informazioni ricevute via tastiera, comunicando eventuali possibili sviste dell'utente.

Non solo: nella memoria del computer è già compresa la lista aggiornata di tutte le varie centinaia di prezzi in vigore. L'utente non deve fare altro che leggere su video il risultato finale delle operazioni di conteggio e ricerca, espresso in lire.

Relativamente alle zone di immatricolazione delle auto sono previsti, oltre a tutte le province d'Italia, anche casi particolari come, per esempio, la Repubblica di San Marino, la Città del Vaticano o il Corpo diplomatico.

Tramite un'unica pagina grafica multicolor si può controllare con facilità lo svolgimento di tutte le routine di elaborazione, con monitoraggio effettuato da un display che permette una lettura istantanea e, soprattutto, facilità di interpretazione senza possibilità di errori.

La carica sul Commodore 64 si effettua con i consueti comandi di loading, ovvero LOAD™ oppure LOAD™RC-AUTO™ o, anche con il pratico metodo di pressione contemporanea dei tasti Commodore e Run/Stop, che causa tra l'altro l'autostart immediato a fine carica, evitando la necessità del comando Run attivatore.

In memoria Ram il programma occupa uno spazio di poco più di 5 Kb, che ammonta di circa 2,4 Kb a lancio avvenuto.

## Uso del programma

Terminata l'operazione di caricamento e trascorsi pochi attimi da quando il programma viene fatto partire, compare la pagina-monitor a fondo e bordo di colore nero, che rimane invariata nella sua parte grafica e strutturale durante tutte le fasi di esecuzione delle routine.



Nella zona inferiore dello schermo sono presenti la scritta "RC-AUTO" e il data-set di copyright (in colore blu reverse).

A centro video si trova invece, simpaticamente, il disegno di un pocket-computer con tanto di mini-display funzionante (a caratteri verdi su fondo nero), che è quello di riferimento per tutte le operazioni d'immissione dei dati richiesti e di proiezione dei responsi (siano questi messaggi-guida o risultati di elaborazione).

Anche se non esplicitamente precisato su video, s'intende che l'uso del programma Rc-auto non comporta, comunque e da chiunque venga usato, alcuna responsabilità per chi lo ha realizzato, prodotto e pubblicato: occorrerà, pertanto, controllare sempre attentamente ogni operazione che porta poi al calcolo dell'importo di un premio assicurativo Rc-auto.

Il display ha come prima videata il seguente messaggio introduttivo:

CIAO, UTENTE, IO  
SONO IL COMPUTER  
DEL TUO COMPUTER (!)

MI ATTIVERO  
AUTOMATICAMENTE  
TRA 7 SECONDI

sta a significare che occorre attendere pochi secondi prima dell'attivazione vera e propria del software in quanto il computer, nel mentre, carica velocemente in memoria i dati che dovrà poi usare o meno a seconda dei vari casi che sarà chiamato a risolvere.

Dopo detto periodo di tempo compare la seconda videata:

PROGRAMMA RC-AUTO  
IN FUNZIONE

PER INIZIARE PREMI  
IL TASTO RETURN

A questo punto, premuto il tasto Return per iniziare effettivamente, è possibile vedere subito la terza videata che chiede il primo input da specificare:

INNANZITUTTO DOVRESTI  
SPECIFICARE LE LETTERE  
DI IDENTIFICAZIONE



DELLA TARGA  
DELL'AUTOMOBILE

È chiaro che viene usato un linguaggio di comunicazione confidenziale, diretto, anche per facilitare il dialogo utente-computer.

Pertarga dell'automobile s'intende quella della provincia di immatricolazione del veicolo: per esempio si inserirà Mi per Milano, Na per Napoli, Aq per L'Aquila, Ss per Sassari, Ao per Aosta e così via. È importante ricordare che non sono ammessi errori, anche perché in questo caso a fine elaborazione ci sarebbe dal computer un invito a ripetere tutti gli input.

Nel caso di Roma occorre specificare la targa così com'è, cioè Roma, senza abbreviare.

Sono previsti input particolari: specificare Scv per auto della Città del Vaticano, Cd per il Corpo diplomatico, Ee per gli escursionisti esteri, Ext per tutte le targhe estere, Rsm per la Repubblica di San Marino, o anche Smom, Afi o Ftase se interessa (non ci si dilunga qui con lunghe spiegazioni circa i significati delle ultime tre sigle, gli interessati possono comunque usarle perché previste dal software).

È disponibile un campo di una sigla massimo di cinque lettere e, nella maggior





## TUTORIAL

parte dei casi, basteranno le due identificative della provincia di immatricolazione. Sono abilitati all'input tutti i tasti dalla A alla Z e, su video, è possibile leggere l'immissione.

La conferma della specifica va data premendo il tasto Return avendo cura, però, di correggere prima eventuali errori con il tasto Inst/Del.

Il computer non accetta conferme di input nulli (che non hanno, cioè, alcun carattere numerico o letterario specificato).

Per il secondo input viene proiettata la quarta videata:

ORA VORREI SAPERE  
IL NUMERO DEI  
CAVALLI FISCALI  
DELL'AUTOMOBILE

A questo punto la specifica è di tipo numerico e, infatti, sono abilitati i soli dieci tasti dall'uno allo zero (fila in alto della tastiera).

Il campo massimo disponibile è di una cifra di tre numeri (da 0 a 999).

Per sapere quanti sono i cavalli fiscali dell'auto di riferimento si possono controllare i documenti di immatricolazione, oppure delle riviste specializzate che solitamente pubblicano, nella parte finale, i listini di tutte le auto in produzione e commercio. Attenzione a non confondere mai i cavalli fiscali (quelli che interessano in questa sede) con i cavalli-motore (che invece indicano la potenza di una macchina). Solitamente i primi sono sempre quantitativamente di molte volte inferiori ai secondi citati: per esempio una Lancia Delta Hf integrale ha 20 cavalli fiscali e ben 181 cavalli di potenza; una Ferrari F40 vale 26 cavalli per il fisco, ma è un mostro da ben 478 cavalli-motore! Diventa più facile sbagliarsi con la Fiat 126, con un motore da dieci cavalli fiscali e soli 25 dell'altro tipo. La legge prevede cinque classificazioni nella fattispecie, a cui corrispondono ovviamente altrettante fasce di prezzo Rc-auto: è comunque il computer che pensa automaticamente al pescaggio del settore giusto, di caso in caso. Anche per questo secondo input occorre dare conferma con il tasto Return, dopo avere corretto eventuali errori con Inst/Del.

L'ultimo dato che viene chiesto all'uten-

te è relativo alla classificazione bonus malus sugli automobilisti e relativi veicoli: infatti le compagnie di assicurazione, in base al fattore di più o meno comprovata pericolosità dei loro clienti, li suddividono in "buoni" e "cattivi", distinguendo sette categorie. La migliore è la prima e ci si arriva dopo anni di guida senza incidenti o danni causati e rimborsati, mentre la peggiore (e più frequentata...) è guarda caso la settima, quella che a parità di condizioni prevede i premi (per l'assicurazione) più alti.

La quinta videata, con il messaggio:

TI CHIEDO INFINE  
IN CHE CLASSE  
DI RIFERIMENTO  
È CONSIDERATO  
L'AUTOMOBILISTA

invita proprio a specificare, con un numero da 1 a 7, la classe di appartenenza.

Sono, ovviamente, anche in questo caso abilitati i tasti dall'1 allo 0 (8, 9 e 0 non vanno comunque usati), così come la conferma va data premendo Return (dopo eventuali correzioni operate con Inst/Del).

La classificazione può essere ricercata controllando la polizza assicurativa o contattando il proprio agente.

Inserire valori casuali comporta ovviamente proiezione di responsi non veritieri, per cui è bene operare solo conoscendo alla perfezione zona di immatricolazione dell'auto, cavalli fiscali della stessa ed esatta classificazione bonus malus (come da polizza).

Ricordare sempre che, più crescono i cavalli fiscali e il numero di classe, più cresce il premio da pagare per assicurarsi: inoltre, anche la targa della vettura ha motivo di comparire tra i dati da specificare, perché anch'essa influisce sulla determinazione del costo Rc. Un esempio: nelle province di Bologna, Firenze, Genova, La Spezia, Lucca, Massa e Pistoia i premi sono, a parità di condizioni, mediamente il doppio (in lire) che ad Agrigento, Ragusa e Siracusa, città valutate come a bassissimo rischio di sinistri. Una Autobianchi Y10 Fi-re paga, su classe prima bonus malus, solo lire 165.900 all'anno ad Agrigento e ben lire 331.900 all'anno a Bologna. La sesta videata, con il messaggio:



## STO ELABORANDO I DATI INSERITI

TRA POCHI SECONDI  
POTRAI VEDERE  
IL RISULTATO  
DEL MIO LAVORO

avverte che il computer esegue l'elaborazione dei dati inseriti e calcola l'importo del premio assicurativo Rc-auto risultante.

Nel caso prima esposto della Y10 Fire su Bologna appare, dunque, la seguente videata finale:

IN QUESTO CASO  
IL PREMIO DELLA RC-AUTO  
E' DI LIRE 331.900

PER UN'ALTRA  
ELABORAZIONE PREMI  
IL TASTO RETURN

premando il tasto Return si può di seguito tornare alla videata iniziale ed effettuare un'altra elaborazione.

Una precisazione doverosa a questo punto: i valori ottenibili sono importi massimi consentiti dalla legge per chi si assicura con un massimale di un miliardo (massimale, peraltro, scelto da quasi tutti gli automobilisti) e con riferimento a classi bonus malus. In caso di polizze di diversa tipologia devono, pertanto, essere modificati.

Se, per qualsiasi motivo, volutamente o per svista, si inserisce e si conferma anche un solo input sbagliato o fuori dal campo di accettazione previsto, scatta un automatismo che interrompe l'elaborazione finale e, al posto dell'ultima videata, fa comparire questa alternativa:

CI SONO DATI  
NON CORRETTAMENTE  
SPECIFICATI

PER RIPETERE  
GLI INPUT PREMI  
IL TASTO 3RETURN+

In questo caso occorre, ovviamente, ripetere da capo le fasi di immissione dei dati che verranno di nuovo richiesti.

Eccovi un esempio di uso pratico del pro-



gramma: si vuole conoscere il premio di assicurazione Rc per un'auto Seat Ibiza 1.5 Glx da immatricolare con targa Pescara per un guidatore di classe quarta. Per prima cosa si specifica Pe come lettere di identificazione (targa), poi si indicano i cavalli fiscali, che sono 16 e infine la classe 4. Dopo pochi istanti la videata finale proietta il costo-premio, calcolato in lire 444.400.

Infine sarà gradita una curiosità riguardante l'argomento Rc-auto.

Si tratta di estremi: un pessimo o nuovo guidatore (classe settima) a bordo di una Lamborghini Countach 5000 quattrottole Lp immatricolata a Firenze pagherà un premio annuale di un milione 184.500 lire, mentre un automobilista-modello (classe prima) su una Innocenti 500 L immatricolata a Ragusa pagherà solo, annualmente, lire 114.400 (insomma il 90% in meno).

Il programma Rc-auto, insomma, aiuterà tutti gli automobilisti ad andare d'accordo con il proprio agente assicuratore e, speriamo, insegnerà qualcosa a tutti coloro che non conoscono i problemi della piccola amministrazione casalinga, perché troppo giovani o perché non se ne sono mai occupati, come desideriamo che accada spesso per i programmi di questa rubrica.

**Daniele Malavasi**



C64

DIDATTICA

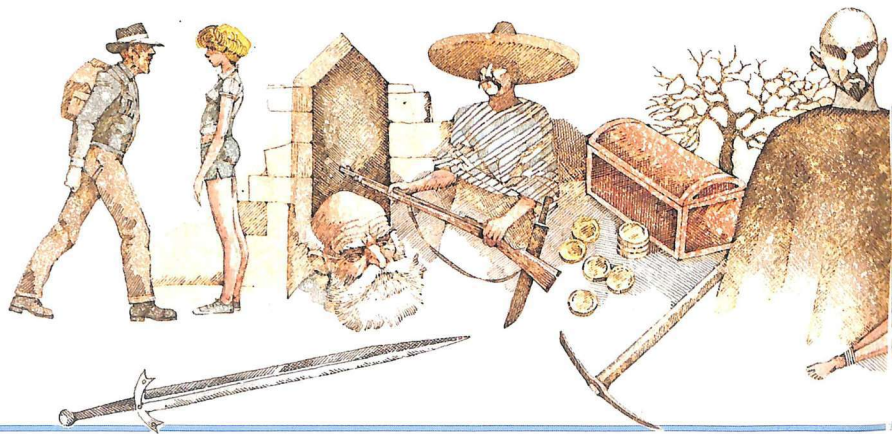
# In sogno, ancora sull'isola del tesoro

*Si conclude su questo numero il lungo e amatissimo corso di programmazione Logo. In questa puntata si parlerà ancora di avventura (quella del numero scorso), che cercheremo di perfezionare razionalizzando la struttura e la logica del programma*

Ultima puntata

## CAPITOLO DICIASSETTESIMO

Se il programma che avete digitato seguendo il numero scorso funziona in tutte le sue parti, potete iniziare a telefonare ai vostri amici per vedere come se la cavano nel ruolo di naufraghi. Prima di invitarli bisognerà rimediare a un grosso difetto che il programma ha ancora: come avrete sicuramente notato, basta un minimo errore di digitatura per bloccarlo. Certo, dal momento che tutti i comandi so-





no procedure Logo si può continuare ugualmente a giocare, ma quando vi impottereste del tesoro il computer non potrà accorgersi della vostra vittoria.

Per ovviare a questo inconveniente svilupperemo la procedura Interpreta che avevamo lasciato in sospenso: essa dovrà esaminare il comando inserito controllando che sia corretto.

A questo scopo ci servirà sicuramente una lista delle parole ammesse nel gioco.

AS "VOCABOLARIO (INVENTARIO GUARDA NORD EST SUD OVEST)  
PRENDI LASCIA  
CONCHIGLIA STATUETTA SPADA TESORO]

Per prima cosa faremo passare il comando attraverso una procedura che lo ripulisce di tutte quelle parole che non sono contenute nel vocabolario.

Per esempio, la frase <<prendi la maledetta conchiglia>> dovrà essere trasformata in <<prendi conchiglia>>, mentre <<per fa-

vore, vai verso ovest>> diventerà semplicemente ovest.

La procedura Ripulisci, scritta qui sotto, assomiglia decisamente a quella Rimuovi che abbiamo già visto.

PER RIPULISCI :COM  
SE VUOTO? :COM RI []  
SE NON AP? PRI :COM :VOCABOLARIO RI RIPULISCI MP :COM  
RI INPRI PRI :COM RIPULISCI MP :COM  
FINE

La procedura Interpreta comincerà quindi così:

PER INTERPRETA :COMANDO  
AS "COMANDO RIPULISCI :COMANDO

Adesso basterà analizzare il comando e farlo accettare solo se è scritto correttamente.

• **Primo caso:** il comando è una lista vuota. Forse è stato premuto inavvertitamente il tasto Return, o forse c'è stato un errore di digitatura; comunque

non abbiamo niente da fare.

SE VUOTO? :COMANDO ST "EH? RI []

• **Secondo caso:** il comando consiste di una sola parola.

Allora bisogna che la parola sia una delle seguenti: Inventario, Guarda, Nord, Sud, Est, Ovest. Anche se non sono propriamente dei verbi, le chiameremo parole Intransitive, Definite Così:

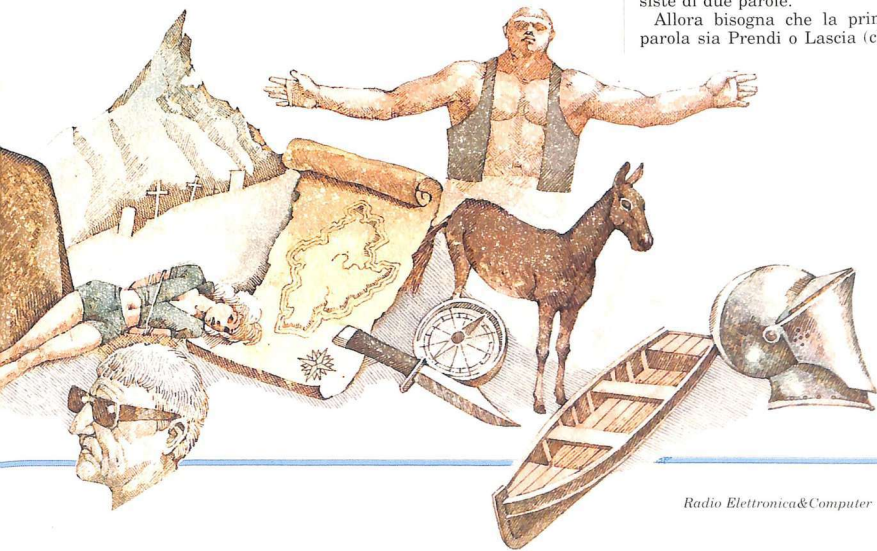
PER INTRANS? :PAROLA  
RI AP? :PAROLA (INVENTARIO GUARDA NORD SUD EST OVEST)  
FINE

Affinché un comando lungo una parola sia valido, occorre che quella parola sia Intrans?. Usiamo qui la primitiva Tutti-veri? che permette di unire due condizioni dopo il Se.

SE TUTTIVERI? (CONTA :COMANDO) =1  
(INTRANS? PRI :COMANDO)  
ALLORA RI :COMANDO

• **Terzo caso:** il comando consiste di due parole.

Allora bisogna che la prima parola sia Prendi o Lascia (che





C64

## DIDATTICA

*Ecco finalmente il testo completo del nostro programma di avventura.*

```

PER START
PT SFOTESTO 0 COLTESTO 1
.DEPOSITA 53272 23
INIZIALIZZA
INTRODUZIONE
GIOCO
FINE

```

```

PER INIZIALIZZA
AS "DOVESONO 1

```

```

AS "MAPPA [(2 0 4 5)
[0 3 1 0]
[0 0 0 2]
[1 0 0 6]
[0 1 0 0]
[0 4 0 0]]

```

```

AS "DESCRIZIONI [(SEI IN UNA INTRICATISSIMA JUNGLA.)
[SEI DI FRONTE ALL'IMPONENTE TEMPIO DEL DIO
DELL'ISOLA.] [SEI IN UNA OSCURA CAVERNA SULLE PENDICI
DEL VULCANO.] [SEI NEL BEL MEZZO DEL VILLAGGIO DEI
CANNIBALI.] [SEI SU UNA SPIAGGIA SOLEGGIATA.] [SEI NEI
PRESSI DI UN ARSENALE ABBANDONATO DAI PIRATI.]]

```

```

AS "OGGETTI [(5 CONCHIGLIA)
[4 STATUETTA]
[6 SPADA]
[3 TESORO]]

```

```

AS "CONDIVITA [(]
[C'E'? STATUETTA (-1)]
[C'E'? SPADA (-1)]
[C'E'? CONCHIGLIA (-1)]
[ ]
[ ]

```

```

AS "MESSAGGIATURA [(] [HAI RIPORTATO AL TEMPIO LA SACRA
STATUETTA RUBATA SECOLI FA DAI CANNIBALI. IL DIO È
RICONSCENTE.] [L'ORSO CHE ABITA LA CAVERNA TI ATTACCA,
MA TU LO TRAFIGGI CON UN PRECISO COLPO DI SPADA.] [LA
CONCHIGLIA CHE HAI CON TE È IL MAGICO SIMBOLO DELLA
TRIBÙ, PERSO DA GENERAZIONI. I CANNIBALI TI
PROCLAMANO LORO RE.] [ ] [ ]

```

```

AS "MESSAGGIUMORIE [(] [HAI PROFANATO IL TEMPIO DEL
POTENTISSIMO DIO, CHE TI FULMINA CON LE SUE SAETTE.]
[ L'ORSO CHE ABITA LA CAVERNA TI ATTACCA FEROCEMENTE.
NON HAI ARMI PER DIFENDERTI.] [ALCUNE DECINE DI
CANNIBALI TI SAITANO ADDOSSO E TI METTONO IN PADELLA.]
[ ] [ ]

```

```

AS "VOCABOLARIO [AIUTO INVENTARIO GUARDA NORD EST SUD
QUEST PRENDI LASCIA CONCHIGLIA STATUETTA SPADA TESORO]
FINE

```

```

PER INTRODUZIONE
SB [TI SVEGLI ALL'IMPROVISO.]
SB [TI SENTI FRASTORNATO E RAMMENTI IL TERRIBILE
NAUFRAGIO DELLA SCORSA NOTTE.]
SB [SARÀ PPOPRIO IL CASO DI ESPLOREARE L'ISOLA:]
SB [SAI PER CERTO CHE DA QUALCHE PARTE SU DI ESSA SI
CELA UN INESTIMABILE TESORO.]
GUARDA
SB [(BATTI "AIUTO" PER QUALCHE SPIEGAZIONE)]
FINE

```

```

PER GIOCO
ST [E ADESSO?]
ESEGUI INTERPRETA LR
SE FINITO? ALLORA HAIVINTO
GIOCO
FINE

```

```

PER NORD
SPOSTATI 1
FINE

```

```

PER EST
SPOSTATI 2
FINE

```

```

PER SUD
SPOSTATI 3
FINE

```

```

PER OVEST
SPOSTATI 4
FINE

```

```

PER SPOSTATI :DIR
LOCALE "NUOVOLUOGO
AS "NUOVOLUOGO ELEMENTO :DIR ( ELEMENTO :DOVESONO :MAPPA
)

```

```

SE "NUOVOLUOGO = 0 ST [NON SI PUÒ ANDARE DA QUELLA
PARTE!] STOP
ST "OK.
AS "DOVESONO :NUOVOLUOGO
GUARDA
SE "E'PERICOLO? ALLORA ESEGUIPERICOLO
FINE

```

```

PER GUARDA
SB ELEMENTO :DOVESONO :DESCRIZIONI
ST "VEDI:
STAMPAOGGETTI :DOVESONO
FINE

```

```

PER INVENTARIO
ST [HAI CON TE:]
STAMPAOGGETTI (-1)
FINE

```

```

PER STAMPAOGGETTI :LUOGO
LOCALE "TROVATI
AS "TROVATI COSAC'E' :LUOGO :OGGETTI [(]
SE VUOTO? :TROVATI ST [UN BEL NIENTE!] STOP
ST :TROVATI
FINE

```

```

PER COSAC'E' :LUOGO :TABELLA :LISTA
SE VUOTO? :TABELLA RI :LISTA
LOCALE "COPIA
AS "COPIA PRI :TABELLA
SE PRI :COPIA = :LUOGO AS "LISTA ( FR :LISTA LUT
:COPIA )
RI COSAC'E' :LUOGO MP :TABELLA :LISTA
FINE

```

```

PER PRENDI :OGG
SE NON C'E'? :OGG :DOVESONO ST [NON C'E'!] STOP
ST "OK.
SPOSTA :OGG :DOVESONO (-1)
FINE

```

```

PER LASCIA :OGG
SE NON C'E'? :OGG (-1) ST [NON CE L'HAI!] STOP
ST "OK.
SPOSTA :OGG (-1) :DOVESONO
FINE

```

```

PER C'E'? :OGG :LUOGO
RI AP? ( LISTA :LUOGO :OGG ) :OGGETTI
FINE

```

```

PER SPOSTA :OGG :DA :A
AS "OGGETTI SOSTI ( LISTA :A :OGG ) ( LISTA :DA :OGG )
:OGGETTI
FINE

```

```

PER SOSTI :NUOVO :VECCHIO :LISTA

```



```

SE VUOTO? :LISTA RI :LISTA
SE PRI :LISTA = :VECCHIO RI INFRI :NUOVO MP :LISTA
RI INFRI ( PRI :LISTA ) ( SOSTI :NUOVO :VECCHIO MP
:LISTA )
FINE

PER CONCHIGLIA
RI "CONCHIGLIA
FINE

PER STATUETTA
RI "STATUETTA
FINE

PER SPADA
RI "SPADA
FINE

PER TESORO
RI "TESORO
FINE

PER C'E'PERICOLO?
RI NON VUOTO? ELEMENTO :DOVESONO :CONDVITA
FINE

PER ESEQUIPERICOLO
SE SOPRAVVIVI? VIVI ALTRIMENTI MUORI
FINE
*fp0
PER SOPRAVVIVI?
RI ESEGUI ELEMENTO :DOVESONO :CONDVITA
FINE

PER VIVI
SB ELEMENTO :DOVESONO :MESSAGGIVITA
ANNULLAPERIC
FINE

PER ANNULLAPERIC
AS "CONDVITA SOSTI {} ( ELEMENTO :DOVESONO :CONDVITA )
:CONDVITA
FINE

PER MUORI
SB ELEMENTO :DOVESONO :MESSAGGIMORTE
SFOTESTO 2
ST [SEI MORTO?]
ST [PREMI UN TASTO PER RICOMINCIARE.]
ASPETTASTO
START
FINE

PER ASPETTASTO
RIPETI 10000 [SE TASTO? STOP]
FINE

PER FINITO?
RI C'E'? TESORO (-1)
FINE

PER HAIVINTO
ST [COMPLIMENTI, HAI VINTO!]
    
```

```

PUNTOCAPO
FINE

PER INTERPRETA :COMANDO
AS "COMANDO RIPULISCI :COMANDO
SE VUOTO? :COMANDO ST "HI? RI {}
SE TUTTIVERI? ( CONTA :COMANDO ) = 1 ( INTRANS? PRI
:COMANDO ) ALLORA RI :COMANDO
SE ( TUTTIVERI? ( CONTA :COMANDO ) = 2 TRANS? PRI
:COMANDO OGGETTO? ULT :COMANDO ) ALLORA RI :COMANDO
ST [NON CAPISCO.]
RI {}
FINE
PER RIPULISCI :COM
SE VUOTO? :COM RI {}
SE NON AP? PRI :COM :VOCABOLARIO RI RIPULISCI MP :COM
RI INFRI PRI :COM RIPULISCI MP :COM
FINE

PER INTRANS? :PAROLA
RI AP? :PAROLA [INVENTARIO GUARDA NORD SUD EST OVEST
AIUTO]
FINE

PER TRANS? :PAROLA
RI AP? :PAROLA [PRENDI LASCIA]
FINE

PER OGGETTO? :PAROLA
RI AP? :PAROLA [CONCHIGLIA STATUETTA SPADA TESORO]
FINE

PER AIUTO
PT SFOTESTO 6
SB [DAMMI PURE DEL TU.]
SB [PUOI FARMI LE RICHIESTE CHE CREDI IN ITALIANO
CORRENTE, ANCHE SE PURTROPPO NON CONOSCO MOLTE PAROLE.]
SB [SONO IN GRADO DI FARTI MUOVERE LUNGO I QUATTRO
PUNTI CARDINALI, DI PRENDERE E LASCIARE UN OGGETTO, DI
GUARDARE LA SCENA (E RIFERIRE) E DI FARE L'INVENTARIO
DEGLI OGGETTI CHE STAI PORTANDO.]
SB [INOLTRE, BATTENDO "AIUTO" POTRAI SEMPRE RILEGGERTI
QUESTA PAGINA.]
SB [PREMI QUALSIASI TASTO PER CONTINUARE.]
ASPETTASTO
PT SFOTESTO 0
GUARDA
FINE

PER SB :LISTA
SE VUOTO? :LISTA ST {} STOP
SE ( CONTA PRI :LISTA ) * ( PRI CURSORE ) > 36 ST {}
SCRIVI PRI :LISTA
SPAZIO
SB MP :LISTA
FINE

PER SPAZIO
SCRIVI CARATTERE 32
FINE

AS "AUTOINIZIO [START]
    
```

sono verbi TranSitivi), e che la seconda (e ultima) sia il nome di un Oggetto).

```

PER TRANS? :PAROLA
RI AP? :PAROLA [PRENDI LASCIA]
FINE
    
```

```

PER OGGETTO? :PAROLA
RI AP? :PAROLA [CONCHIGLIA STATUETTA
    
```

```

SPADA TESORO]
FINE
    
```

Usiamo di nuovo Tuttiveri?, questa volta con le parentesi tonde perché gli diamo tre argomenti invece di due.

```

SE ( TUTTIVERI? ( CONTA :COMANDO ) = 2
TRANS? PRI :COMANDO
    
```

```

OGGETTO? ULT :COMANDO )
ALLORA RI :COMANDO
    
```

• **Quarto caso:** tutte le altre possibilità. Se il comando è vuoto, o se è più lungo di due parole, o ancora se è sgrammaticato (come <<Nord Conchiglia>> o <<Prendi>>), non abbiamo proprio niente da eseguire. Ci limi-



tiamo quindi a stampare un messaggio di protesta e riportiamo la lista vuota.

ST [NON CAPISCO.]  
RI [ ]

Per riassumere, ecco qua la procedura Interpreta finalmente completa.

PER INTERPRETA :COMANDO  
AS "COMANDO RIPULISCI :COMANDO  
SE VUOTO? :COMANDO ST "EH? RI [ ]  
SE TUTTIVERI? (CONTA :COMANDO) = 1  
( INTRANS? PRI :COMANDO )  
ALLORA RI :COMANDO  
SE ( TUTTIVERI? (CONTA :COMANDO) = 2  
TRANS? PRI :COMANDO  
OGGETTO? ULT :COMANDO )  
ALLORA RI :COMANDO  
ST [NON CAPISCO.]  
RI [ ]  
FINE

### Un po' di aiuto

Ora l'avventura è veramente pronta per essere giocata. Conservatela accuratamente, ma prima non dimenticatevi di digitarla:

AS "AUTOINIZIO [START]

Con questa istruzione il programma partirà da solo subito dopo il recupero. Per chiarimenti consultate l'appendice A.

Nei prossimi paragrafi discuteremo brevemente alcuni possibili perfezionamenti. Non dimentichiamoci intanto che questo programma può servire a costruire qualunque avventura che segua gli stessi schemi: basterà modificare le varie liste create in Inizializza, le procedure con il nome degli oggetti e il predicato Oggetto?. Le limitazioni riguardano soprattutto le poche azioni possibili e il rigido schema dei pericoli.

Aggiungere nuove possibilità di azione non è necessariamente

complicato: basta definire a dovere nuove procedure. Per esempio, se nel gioco abbiamo una lampada che deve essere strofinata per fare apparire il genio, possiamo definire una

PER STROFINA :OGGETTO  
SE NON :OGGETTO = LAMPADA ST [NON  
SUCCEDA NIENTE!] STOP  
eccetera

Dobbiamo poi ricordarci di inserire la parola Strofinata sia nel Vocabolario sia nella lista del predicato Trans?.

In ogni caso, sarà sempre bene preparare una pagina di aiuto per il giocatore, per informarlo riguardo alle parole che gli è concesso usare. Possiamo proprio definire appositamente una procedura Aiuto che possa essere chiamata in qualsiasi momento.

PER AIUTO  
PT  
ST [DAMMI PURE DEL TU.]  
ST [PUOI FARMI LE RICHIESTE CHE CREDI  
IN ITALIANO CORRENTE,  
ANCHE SE PURTROPPO NON CONOSCO  
MOLTE PAROLE.]  
ST [SONO IN GRADO DI FARTI MUOVERE  
LUNGO I QUATTRO  
PUNTI CARDINALI, DI PRENDERE E LASCIARE  
UN OGGETTO,  
DI GUARDARE LA SCENA (E RIFERIRE) E DI

FARE  
L'INVENTARIO DEGLI OGGETTI CHE STAI PORTANDO.]  
ST [INOLTRE, BATTENDO "AIUTO"  
POTRAI SEMPRE RILEGGERTI QUESTA PAGINA.]  
ST [PREMI QUALSIASI TASTO PER CONTINUARE.]  
ASPETTASTO  
PT  
GUARDA  
FINE

Inserite questa procedura nel programma; aggiungete poi la parola Aiuto a Vocabolario e alla lista di Intrans?.

Inoltre, per informare il giocatore dell'esistenza di questa procedura, si può aggiungere in coda a Introduzione la linea:

ST [(BATTI "AIUTO" PER QUALCHE SPIEGAZIONE)]

In questo modo il gioco permetterà un approccio molto più amichevole.

### Suggerimenti

Per concludere, vediamo alcuni piccoli accorgimenti che possono servire a rendere i vostri programmi più gradevoli e, per così dire, più professionali.

Uno stratagemma sempre d'effetto è quello di modificare il colore dello schermo o quello del testo al momento giusto. Per farlo





basta utilizzare le primitive Sfo-  
testo e Coltesto con un argomen-  
to numerico da zero a 15.

Inoltre, avrete notato che pre-  
mendo contemporaneamente i  
tasti Shift e Commodore (in bas-  
so a sinistra sulla tastiera) i ca-  
ratteri sullo schermo vengono  
trasformati da maiuscoli in mi-  
nuscoli e viceversa. Se le lettere  
minuscole vi sembrano più ele-  
ganti, potete inserire nel pro-  
gramma l'istruzione:

DEPOSITA 53272 23

Attenti a non sbagliare gli ar-  
gomenti, però: .Deposità è una  
primitiva che serve a scrivere un  
valore (in questo caso 23) diret-  
tamente nella memoria del com-  
puter (in questo caso nella cella  
di memoria il numero 53272).

Depositando numeri a casac-  
cio si rischia di danneggiare il  
funzionamento del sistema, e po-  
treste essere costretti a spenge-  
re il computer perdendo così  
quello che avete in memoria.  
Non potete causare danni per-  
manenti, comunque.

Una caratteristica un po' fasti-  
diosa del programma è forse il  
fatto che non sa andare a capo  
stampando i messaggi. Per ov-  
viare a ciò possiamo scrivere una  
procedura che stampi una paro-

la alla volta controllando prima  
la sua lunghezza. Usiamo qui la  
primitiva Cursore, che riporta  
una lista di due elementi: il pri-  
mo è la colonna dello schermo in  
cui si trova il cursore (cioè dove  
verrà stampato il prossimo ca-  
rattere), mentre il secondo è la  
riga. Sommando Pri Cursore al-  
la lunghezza della prossima pa-  
rola otterremo un certo numero;  
se questo numero è maggiore di  
36 (lo schermo ha 40 colonne)  
stamperemo la lista vuota per  
andare a capo. Sb è l'abbrevia-  
zione di Stampa Bene.

```
PER SB :LISTA
SE VUOTO? :LISTA ST [] STOP
SE ( CONTA PRI :LISTA ) + ( PRI CURSORE ) >
36 ST []
SCRIVI PRI :LISTA
SPAZIO
SB MP :LISTA
FINE
```

```
PER SPAZIO
SCRIVI CARATTERE 32
FINE
```

Abbiamo usato altre due pri-  
mitive: la prima è Scrivi, che

esegue la stessa cosa di Stampa,  
ma non va a capo quando ha fini-  
to; l'altra è Carattere, che accet-  
ta come argomento un numero e  
riporta il carattere corrispon-  
dente secondo una tabella chia-  
mata codice Ascii.

Per esempio, il Carattere nu-  
mero 65 è la lettera A, il numero  
66 è la lettera B, e così via.

In questo codice compaiono  
tutti i caratteri stampabili, e vi è  
incluso anche lo spazio, al nume-  
ro 32. Così, anche se il Logo solit-  
amente ignora gli spazi (o me-  
glio, li usa come separatori tra  
parole) possiamo fargliene stam-  
pare con Scrivi Carattere 32.

Naturalmente, per usare la  
procedura Sb nel programma do-  
vremo chiamarla al posto di St  
tutte le volte che occorre. Buon  
lavoro!

## Esercizi

Suona un po' strano che quan-  
do il nostro eroe riporta la con-  
chiglia ai cannibali questi lo fac-  
ciano re ma non se la riprenda-  
no; lo stesso vale per la statuet-  
ta riportata al tempio.

Trovate il modo di far sparire  
al momento giusto questi ogget-  
ti (per esempio, mettendoli nella  
locazione 0, da dove sicuramente  
non ricompariranno). Come ab-  
biamo suggerito poco fa, associa-  
te a ogni luogo dell'avventura un  
disegno, da chiamare nella pro-  
cedura Guarda.

Poiché avrete dei problemi di  
memoria se cercate di eseguirlo  
da programma con le dovute pro-  
cedure, vi conviene conservare i  
disegni su disco e recuperarli  
con Conserdis (consultate l'ap-  
pendice A).

Per il nome dei disegni, potete  
usare i valori di Dovesono (1, 2,  
3 eccetera) o, meglio, qualcosa  
come Paesaggio1, Paesaggio2  
eccetera ottenuti con Parola Pa-  
esaggio :Dovesono.

**Luigi Ambrosi**





**C64**

GRAFICA

# Alla scoperta dei pixel

Una delle caratteristiche più interessanti dei computer è la gestione della grafica in alta risoluzione, prerogativa anche del C64, in merito alla quale, il Manuale d'uso non fa alcun cenno. Ne consegue che il neoutente, che voglia impadronirsi dell'argomento, è costretto a ricorrere a pubblicazioni o riviste specializzate, come questa per esempio, che in passato, fra tanti altri, ha pubblicato anche un esteso corso di grafica. Forse è bene che una rivista faccia ogni tanto un


**C64E**

*Dedicato ai principianti del Basic, questo articolo cerca di far luce sui problemi relativi alla gestione dell'alta risoluzione del Commodore 64*

passo indietro, riprendendo nozioni già esaminate per mettere in grado chi è alle prime armi di comprendere il contenuto degli articoli più avanzati. Provvediamo allora, previo qualche accenno alla bassa risoluzione, a evidenziare gli elementi basilari necessari per gestire l'alta risoluzione. Accesso il computer (siamo in bassa risoluzione), digitiamo:

```
10 FORK=1024TO2023:POKEK,1
```

ove abbiamo utilizzato Pokek,1 per inserire in K la lettera A, il cui codice video è 1; poi, per rallentare e rendere visibili le operazioni, introduciamo:

```
20 FORTM=0TO50:NEXT:NEXT
```

in fine:

```
30 GOTO 30
```

per impedire che compaia il Ready sullo schermo. Al Run vediamo lo sfondo riempirsi di lettere A da sinistra a destra e dall'alto in basso. Quando le A hanno completato una riga passano a quella inferiore. Tutti gli spazi in cui è comparsa la A fanno capo a locazioni numerate. La numerazione, se non disponiamo diversamente con particolari interventi che qui non menzioniamo, viene assegnata automaticamente dal computer. Parte con il numero 1024 per il primo spazio in alto a sinistra e termina con 2023 in fondo a destra. È in funzione di questo comportamento del computer che abbiamo costruito la linea dieci. Ognuna delle predet-

te locazioni comprende otto byte. Se osserviamo bene lo spazio in cui è inserita una lettera, vediamo che è fatto di otto linee orizzontali sovrapposte. Ogni lineetta rappresenta un byte. Lo sfondo dello schermo in bassa risoluzione è paragonabile a un foglio di carta a quadretti, grandi all'incirca come quelli dei quaderni delle classi di prima elementare, sul quale vi sia rappresentato un rettangolo con il lato maggiore di 40 quadretti e quello minore di 25, in tutto mille quadretti. Disegnare su tale foglio di carta il contorno di una figura annerendo mano mano un quadretto, conduce a un risultato piuttosto grossolano.

Abbiamo bisogno di quadretti molto più piccoli, come quelli, per esempio, della carta millimetrata. Ciò possiamo ottenerlo con l'alta risoluzione, ovvero, per usare una terminologia spesso ricorrente, entrando in pagina grafica. Per pagina grafica si intende sempre il medesimo sfondo del video, gestito da ben ottomila locazioni, la cui numerazione, se lasciata agli automatismi del computer, corre da 8192 a 16191. Ognuna di queste locazioni rappresenta un byte, ogni bit del quale può accendere o spegnere un suo corrispondente punto sullo sfondo. Ne consegue che, mentre in bassa risoluzione lo sfondo è paragonabile a un rettangolo con una superficie di mille grossi quadretti, in pagina grafica lo stesso rettangolo risulta di 320 punti per 200, pari in totale a 64mila piccoli punti. Come facciamo a trasferire il governo dello sfondo del video dalle locazioni 1024-2023 a quelle 8192-16191? È semplicissimo. Premiamo Run/Stop-Restore e digitiamo New, quindi:



**C64E**

## GRAFICA

10 PRINTCHR\$(147)  
20 POKE53265,PEEK(53265)OR32  
30 POKE53272,PEEK(53272)OR8

Non ci dilunghiamo a spiegare il funzionamento del metodo, altrimenti andremmo lontano dal proposito di conoscere solo come si debba operare. Dato il Run, la linea dieci pulisce lo schermo da ogni eventuale scritta, le altre due linee ci trasferiscono nella pagina grafica. Ma, non appena in pagina grafica, ci troviamo di fronte a uno sfondo di aspetto strano, costituito da una serie di linee e punti luminosi o neri per un video in bianco e nero, colorati o neri per un video a colori. In alto a sinistra traspaiono alcuni quadretti sotto i quali lampeggia il cursore. Su una pagina di tal fatta, priva di uniformità, non possiamo scrivere niente. Le linee e i punti luminosi sono bit accesi, gli altri sono bit spenti. È facile intuire che se li spegnessimo tutti avremmo una pagina completamente nera sulla quale poter scrivere come con il gesso su una lavagna. Agiremo quindi sulle locazioni per azzerare tutti i byte. Facciamolo al rallentatore per vedere cosa succede. Premiamo RunStop/Restore per far tornare il video in situazione normale, poi con List richiamiamo le linee precedenti e aggiungiamo:

40 FORK=8192TO16191  
50 POKEK,0  
60 FORTM=0TO500:NEXT  
70 NEXT

La linea 60 agisce da rallentatore. Diamo il Run. Appare nuovamente la pagina grafica costituita da linee o punti luminosi e neri. Inizia anche l'opera di spegnimento dei bit o punti accesi, che si svolge mettendo in evidenza il diverso modo di procedere della numerazione delle locazioni in pagina grafica rispetto alla bassa risoluzione. Grazie al rallentatore vediamo che si fa nero un primo trattino in alto a sinistra, poi, uno alla volta, sette trattini immediatamente sottostanti, formando un gruppo di otto trattini neri (quelli già neri restano ovviamente tali). Dopo, diventano neri, sempre uno alla volta, otto trattini affiancati ai primi, e così via. Poiché ogni trattino dipende da una locazione, ciò vuol dire che la numerazione delle locazioni in pagina grafica parte dall'angolo superiore sinistro, procede scendendo di otto posti, per risalire e scendere in continuazione di altri otto fino al termine dello sfondo. È importante tener presente questo fatto se vogliamo riuscire a disegnare. Con i soliti RunStop/Restore e List torniamo al nostro programma e togliamo la linea 60 del rallentatore semplicemente digitando il suo numero seguito da Return. Diamo nuovamente il Run e lasciamo, se già non lo sono, che vengano spenti tutti i punti e lo sfondo sia interamente nero. Solo in alto a sinistra traspariranno alcuni quadretti corrispondenti alla parola Ready. Ora è come se avessimo di fronte una lavagna. Cominciamo con il far apparire un punto. Dato che le locazioni corrono da 8192 a 16191, quella di mezzo sarà contraddistinta dal numero 12192. Decidiamo di accendere il bit quattro del byte corrispondente a tale locazione. Poiché il conteggio della posizione dei bit va da destra a sinistra cominciando da zero, il bit quattro sarà quello di posizione





C64E

tre, e possiamo accenderlo inserendo nella locazione il valore 2^3=8. Premiamo Run-Stop/Restore e List e aggiungiamo al programma la linea:

```
80 POKE12192,8
```

Dopo il Run ci sarà un'attesa piuttosto lunga, in quanto anche se avevamo già posto a zero tutti i bit della pagina grafica, che si ripresenta subito nera, tuttavia occorre aspettare che si compia il ciclo For-Next delle linee 40-70, ciclo che lasciamo in essere per le operazioni future. Comunque, dopo qualche tempo apparirà un punto chiaro in mezzo allo schermo. È il nostro primo disegno. In luogo del punto vogliamo una lineetta? Run-Stop/Restore e List (che d'ora innanzi daremo per scontati quando varieremo il programma) e rettifichiamo la linea 80 inserendo nel byte della locazione 12192 il massimo valore assegnabile, cioè 255, che accende, ovvero pone a uno, tutti i suoi bit:

```
80 POKE12192,255
```

Dato il Run, altra attesa. Il ciclo For-Next scorre tutta la pagina cancellando il punto acceso, al posto del quale dopo un po' apparirà la lineetta. Proviamo a ottenere una riga orizzontale che attraversi lo schermo a metà. Scriveremo in luogo della linea precedente:

```
80 FORK=12032TO12351STEP8:POKEK,255:NEXT
```

La linea 12032 è una locazione a sinistra dello schermo, la 12351 è a destra alla stessa altezza. Però abbiamo dovuto tener presente che c'è un salto di otto numeri tra locazioni affiancate, il che ci ha costretti a ricorrere allo Step 8. Al Run aspetteremo nuovamente con pazienza che venga cancellata la lineetta, poi sullo schermo avremo la nostra linea orizzontale. Il Basic è lentissimo, e noi non vogliamo invecchiare davanti al computer. Sostituiamo il ciclo delle linee 40-70, di cui ormai abbiamo capito la funzione, con una routine in linguaggio macchina che produca gli stessi effetti.

Un programma in Lm è in genere più lungo del corrispondente in Basic, ma è sempre impagabilmente più veloce. Non chiariremo la struttura della routine in Lm per non al-

lontanarci dall'argomento e ci limiteremo a trascrivere le seguenti linee dopo aver cancellato quelle da 40 a 70:

```
40 GOSUB1000
990 END
1000 FORK=49152TO49176:READA:
POKEK,A:NEXT
1010 SYS49152:RETURN
1020 DATA169,32,133,255,169,0,133,
254,169,0,168,145
1030 DATA254,200,208,251,230,255,
165,255,201,64,208,240,96
```

Se non ci sono stati errori di copiatura, il programma funzionerà, e siccome contiene sempre la richiesta di disegnare una linea orizzontale, al Run, grazie al Lm, la vedremo scomparire immediatamente per poi rifarsi subito dopo. Poiché il discorso è diretto ai principianti, questa è un'occasione per mostrare loro la differente velocità di esecuzione tra Basic e Lm. Le cose si complicano se proviamo a ottenere una riga centrale verticale. Ricordiamo ancora una volta quanto abbiamo visto al rallentatore. La numerazione procede dall'alto verso il basso aumentando di un'unità per otto volte consecutive, risale e scende per 40 volte fino al limite destro dello sfondo, quindi prosegue con lo stesso sistema formando 25 strisce ciascuna di 40 blocchetti, ognuno di otto linee. Perciò fra una locazione di una striscia e quella ordinatamente corrispondente della striscia successiva esiste un salto di 320 unità (40 x 8 = 320). Abbiamo inoltre 40 colonne costituite ognuna da 25 gruppi di otto byte. Scegliamo la locazione 8352 che è a circa metà dello sfondo sulla prima linea in alto. Scendendo verso il basso, dopo la prima striscia di 320 locazioni affiancate a gruppi di otto, ne abbiamo altre 24, e 24 x 320 fa 7680. Pertanto digitiamo:

```
50 FORK=0TO7680STEP320
60 FORH=0TO7
70 POKE8352+H+K,8
80 NEXT:NEXT
```

Alla linea 70 abbiamo messo il numero otto per accendere in ogni bit soltanto il bit di posizione tre, e al Run ecco formarsi la linea verticale richiesta. Attenzione, il ciclo contenuto nelle linee 50-80 funziona solo se si parte da un byte che sia il primo di un gruppo di



**C64**

## GRAFICA

otto. Se proviamo a partire da 8354 invece che da 8352, ci accorgiamo che per ottenere una linea verticale dovremmo modificare e complicare il programma. Ma, allora, è possibile che si debbano superare tante difficoltà per disegnare in pagina grafica, ossia in alta risoluzione? Se indicassimo con uno zero l'angolo inferiore sinistro del rettangolo che corrisponde allo sfondo e numerassimo da 0 a 319 i punti della base e da 0 a 199 quelli verticali sul lato sinistro, qualsiasi punto P potrebbe essere facilmente individuato mediante coordinate, cioè con un'ascissa  $x$  e un'ordinata  $y$ . Per chi non avesse studiato la geometria analitica o non avesse mai giocato a battaglia navale, diciamo che  $x$  rappresenta la distanza dal lato sinistro e  $y$  quella dalla base. Dobbiamo trovare un collegamento tra un punto determinato da tali coordinate e il relativo byte designato dalla particolare numerazione delle locazioni in pagina grafica. Procediamo allora in questo modo. Se un punto P ha come coordinate  $X$  e  $Y$ , la parte intera risultante da  $X/8$  sarà la colonna in cui si trova P. Perciò scriviamo:

$$2000 C = \text{INT}(X/8)$$

dove C rappresenta il numero d'ordine della colonna in cui si trova P. La differenza fra  $X$  e  $C \times 8$  indicherà la posizione di P in quella colonna, ossia del bit nel relativo byte, a partire da sinistra. Indichiamo con B tale posizione e scriviamo:

$$2010 B = X - (C * 8)$$

Ora stabiliamo a quale altezza, ossia in quale linea si trova il punto P di ordinata Y. La numerazione delle locazioni va per il computer dall'alto al basso. Noi vogliamo invece partire dal basso. Tenuto presente che i byte rappresentati sullo sfondo dello schermo sono raggruppati verticalmente a otto alla volta, cioè in strisce di otto linee ciascuna, con:

$$2020 S = \text{INT}((199 - Y) / 8)$$

troviamo la striscia S in cui è presente P. Poiché la striscia è di otto linee, con:

$$2030 R = 199 - Y - S * 8$$

otteniamo il numero R della linea cercata.

Siamo finalmente in possesso di tutti i dati occorrenti al computer per risalire dalle coordinate X e Y al numero che contrassegna in pagina grafica il byte o locazione in cui si trova il bit che gestisce P. Se scriviamo:

$$2040 BY = 8192 + S * 320 + C * 8 + R$$

abbiamo che By è il numero del byte in questione. Per accedere in esso il bit relativo a P, dobbiamo eseguire una Poke. Quando abbiamo calcolato  $B = X - (C * 8)$ , abbiamo visto che B era la posizione di P nel byte a partire da sinistra. Poiché nei byte le posizioni dei bit si contano a partire da destra e vanno da zero a sette, la posizione valida per il computer è data da  $7 - B$ . Dovremmo perciò fare  $\text{Poke By}, 2^{(7 - B)}$ . Però si usa ricorrere a Peek con  $\text{Or} 2^{(7 - B)}$  affinché, se il computer deve calcolare più punti e questi risultano appartenere allo stesso byte, non vengano cancellati quelli man mano già disegnati. Ma è anche consigliabile usare  $\text{Or} 2^{\text{Int}(7 -$





B+0.5) se i calcoli relativi a X e Y dipendono da un ciclo For-Next seguito da uno Step riduttore, per esempio, Step.1. Per quanto ci occorre, digitiamo:

```
2050 POKEBY,PEEK(BY)OR2^(7-B)
```

e concludiamo, poiché ciò che abbiamo predisposto è una piccola routine richiamabile mediante Gosub, con:

```
2060 RETURN
```

Non ci resta che mettere alla prova il nostro lavoro. Cancelliamo le linee da 50 a 80, che ormai non ci servono, e vediamo se con il sistema approntato riusciamo a far apparire un punto, un segmento inclinato e un cerchio. Digitiamo (le Rem esplicative possono essere tralasciate):

```
50 REM "PUNTO"
60 X=100:Y=150
70 GOSUB2000
80 REM "SEGMENTO"
90 Y=100
100 FORK=70TO220:X=K:Y=Y+.4
110 GOSUB2000:NEXT
120 REM "CERCHIO DI RAGGIO 50 E
COORDINATE DEL CENTRO 220,70"
130 FORK=-50TO50
140 Y1=SQR(50^2-K*K):X=K+220:
Y=Y1+70:GOSUB2000
150 Y=Y1-70:GOSUB2000
160 NEXT
```

Eventuali errori segnalabili provocano l'apparizione di quadretti luminosi o colorati, a seconda del video in uso. In tal caso digitando in modo diretto, alla cieca, perché i caratteri non compaiono:

```
POKE53265,27:POKE53272,21
```

si torna al modo testo e possiamo leggere l'indicazione d'errore che permette la correzione. Ormai dovremmo aver imparato a disegnare.

Sostituendo alle linee 50-160 altri programmi, possiamo far apparire ciò che vogliamo, anche, sebbene occorra molta pazienza, un paesaggio che sarà in bianco e nero perché qui non abbiamo trattato del colore. I concetti e i metodi sopra esposti, pur se basilari, sono solo introduttivi all'uso dell'alta ri-

soluzione. In quanto tali, dovrebbero invogliare ad approfondire l'argomento.

Al riguardo segnaliamo che nei mesi da dicembre 1986 a settembre 1987 è stato pubblicato su questa rivista un corso completo di grafica, che tratta anche l'utilizzo del colore e la gestione degli sprite. Nella cassetta allegata vi sono alcuni esempi di costruzioni geometriche elementari e di grafici rappresentanti equazioni cartesiane. L'esame del listato non dovrebbe essere di difficile comprensione. Tuttavia riteniamo di dover segnalare quanto segue:

- gli assi cartesiani, graduati ogni dieci pixel, anziché con un lentissimo programma Basic, sono realizzati nelle linee da 1040 a 1320 in Lm allo scopo di abbreviare le attese e dimostrare l'enorme differenza di velocità di esecuzione fra il linguaggio macchina e il Basic;

- l'uso degli assi cartesiani, che permette l'introduzione di valori negativi per X e Y, ha costretto alla variazione dei valori di scala, per cui i cicli For-Next sono seguiti da Step.1, che rallenta, ma rende più preciso il grafico;

- le linee 2045 e 2046 impediscono non solo l'uscita dai valori della numerazione della pagina grafica, ma anche di imbattersi in valori che producano Illegal Quantity;

- le linee 585 e 970 consentono nel corso dell'esecuzione o al termine del grafico delle rette e delle parabole di ritornare subito al menù premendo F1, senza attendere che sia completato il ciclo For-Next;

- la linea 2040 ottiene By mediante la funzione Int al fine di evitare arrotondamenti del computer che possano provocare l'accensione di bit estranei ai risultati desiderati;

- per evidenziare i valori accettabili affinché i grafici di rette e parabole possano essere visualizzati, suggeriamo di iniziare provando a introdurre i seguenti coefficienti:

per le Rette: A=2,B=1; A=-1,B=2; A=0,B=5; A=-0.8,B=0;  
per le Parabole: A=1,B=0; C=0; A=-1,B=4,C=0; A=0.4,B=4,C=2.

Gustavo Fontanella

**Caro lettore,**

a partire dal prossimo numero **"Radio Elettronica & Computer"** cambierà. La testata sarà più sintetica e tratterà solo del software su cassetta (peraltro molto sostanzioso), che manterrà le caratteristiche di sempre. Le istruzioni per l'uso dei programmi su cassetta saranno più curate, ricche di illustrazioni e quindi più facili da apprendere, anche per i principianti. Tutte le rubriche che non fanno riferimento ai programmi su cassetta compariranno però su **"TuttoCOMMODORE"**, una delle tre testate di area Commodore pubblicate dal Gruppo Editoriale JCE. **"Radio Elettronica e Computer"** uscirà in edicola nei mesi di Febbraio, Aprile, Giugno, Settembre e Dicembre.

Salutiamo dunque tutti gli utenti Amiga che d'ora in avanti troveranno la loro rubrica sul nuovo **"TuttoCOMMODORE"** e cogliamo l'occasione per augurare a tutti gli affezionati di R.E.&C

**Buon Natale e Felice AnnoNuovo!**

La redazione  
**Fernando Zanini**

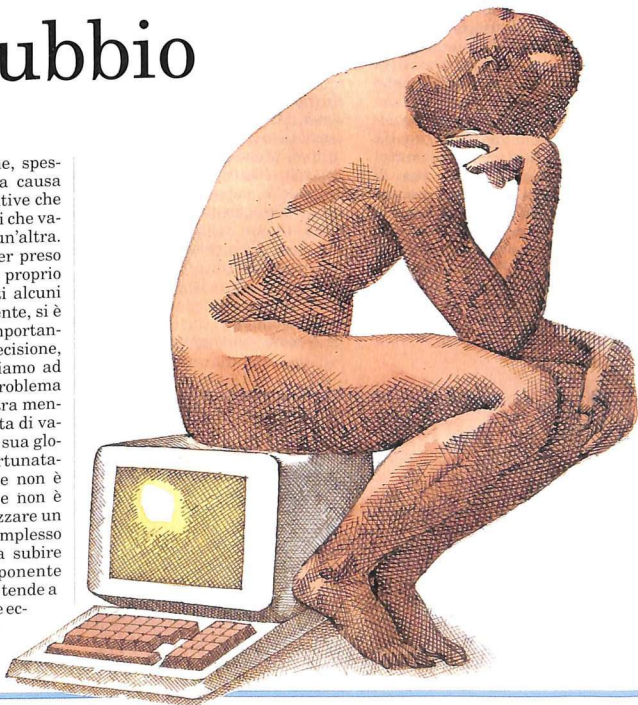




*Quando si tratta di prendere una decisione importante è utilissimo avvalersi della logica infallibile del computer. Con l'utility che pubblichiamo, non solo riuscirete a prendere decisioni con cognizione di causa, ma imparerete anche un metodo d'analisi applicabile a qualsiasi problema*

# L'insostenibile pesantezza del dubbio

Prendere una decisione, spesso, può essere difficile a causa della quantità di alternative che si presentano e dei fattori che valorizzano una scelta o un'altra. Spesso ci si trova ad aver preso una decisione sbagliata proprio perché si sono trascurati alcuni fattori o, più semplicemente, si è valutata male la loro importanza. Nel prendere una decisione, anche importante, tendiamo ad affidare l'analisi del problema esclusivamente alla nostra mente e alla capacità di questa di valutare la questione nella sua globalità. Purtroppo (o fortunatamente?) la nostra mente non è mai del tutto obiettiva e non è sempre in grado di analizzare un problema articolato e complesso in modo integrale senza subire l'influenza della componente emotiva che, non di rado, tende a trascurare o a valorizzare eccessivamente taluni fattori. Noi esseri umani prendiamo troppo spes-







so decisioni in base a ciò che ci suggeriscono le emozioni, più o meno conscie. Molti di noi affidano le loro decisioni ad altri, o meglio, alle emozioni degli altri, perché ritengono che la propria emotività influisca troppo sull'esame del problema. Taluni cercano nell'opinione altrui semplicemente dei consigli nella speranza che il confronto delle varie interpretazioni del problema porti elementi chiarificatori. Nella maggior parte dei casi prendiamo decisioni, anche importanti, basando la nostra valutazione del problema solo su un fattore, magari quello che ci sta più a cuore o che ci ha colpiti di più, ma trascuriamo tanti altri fattori, singolarmente poco significativi, che insieme giustificerebbero una decisione opposta a quella presa.

Il discorso fatto sembrerebbe dare conto a tutti coloro che prendono decisioni in base a quello che sentono dentro. In realtà ciò che si è detto vuole solo mettere in guardia su questo tipo di procedimento verso una decisione e non bollare il buon senso umano come consigliere incapace. Infatti, una decisione presa secondo quello che ci si sente dentro significa lasciare al buon senso e all'intuito (che sembra essere una sorta di super intelligenza inconscia) il compito di trovare

una soluzione senza trascurare sia le necessità oggettive sia quelle soggettive di chi è direttamente coinvolto nel problema e che richiedono una soluzione personalizzata. L'intuito, tuttavia, pur essendo in grado di prendere ottime decisioni personalizzate non è molto affidabile.

La sua affidabilità, inoltre, varia da persona a persona e da problema a problema. Più il problema è ricco di alternative e fattori più è alto il rischio d'errore.

### La soluzione

Come abbiamo visto, pertanto, ogni problema implica una decisione. Il problema pone sempre un certo numero di alternative tra cui scegliere e ogni alternativa è valorizzata da un certo numero di fattori. Il programma che pubblichiamo si chiama Decision Maker e permette di potenziare le capacità del vostro intuito di prendere una decisione, non solo obiettiva e conveniente, ma anche personalizzata. Per chiarire meglio quanto abbiamo detto finora vediamo qualche esempio. Cominciamo con un problema semplice come quello di dover scegliere il tipo di stampante per il Commodore 64. Supponiamo dunque che il signor X, professore universitario, abbia

intenzione di comprare una periferica per stampare i testi che egli scrive ogni giorno con Easy Script e che, da una prima analisi del problema, abbia trovato che la stampante che fa per lui si trova fra un ristretto numero di quelle disponibili sul mercato, cioè, per semplificare il problema, che le alternative siano solo due: una stampante ad aghi in bianco e nero o una a margherita.

Il signor X può affidare la sua decisione al rivenditore. In questo caso il nostro personaggio non tornerà a casa con la stampante più adatta al suo problema ma quella più adatta al problema del rivenditore che è, molto probabilmente, vendere la stampante più costosa tra le due.

Se il signor X è un tipo accorto cercherà, invece, consiglio presso qualche amico che se ne intende. Anche in questo caso, però, pur essendo l'amico colui che per definizione tenta di fare il nostro bene, la decisione non sarà sicuramente la più adatta al signor X, sempre che l'amico non sia così speciale e eccezionale da riuscire a immedesimarsi completamente nel signor X.

Ammettendo, invece, che il signor X sia convinto che far da sé è la soluzione migliore, egli acquisterà la stampante che sente più adatta per le sue necessità dopo aver capito bene le differenze tecniche tra l'una e l'altra periferica. Forse questa è la soluzione migliore, ma richiede maggior sforzo. Il signor X avrà assimilato tutti i fattori del problema, il suo intuito avrà valutato ogni alternativa in rapporto a ogni fattore e, alla fine, avrà suggerito una soluzione. In particolare, il primo passaggio è stato quello di individuare con precisione le alternative possibili. Il signor X ha scartato subito le stampanti laser perché troppo costose. Egli ha scartato anche stampanti a getto di inchiostro o termiche perché costose e



Figura 1.  
Schermata  
principale del  
programma  
con le tredici  
funzioni  
disponibili



Fattore	Importanza
1	80
Esteriorità	100
Economicità	85
Comodità	75
Prestazioni	

Fattore numerico

Figura 2. La lista di tutti i fattori inseriti e la votazione relativa

non specifiche per la stampa di testi. Poi ha scartato quelle ad aghi a colori per le stesse ragioni. Alla fine di questa selezione, però, il signor X, si è ritrovato con il dubbio: una stampante ad aghi monocromatica, che costa poco e permette anche di effettuare la stampa in modo grafico, oppure una stampante a margherita, veloce e studiata apposta per stampare testi, ma più costosa?

Il signor X, dopo aver interpellato il suo intuito, tornerà a casa con una stampante a margherita e ne rimarrà soddisfatto, malgrado il suo amico esperto ritenga che ormai le stampanti a margherita sono state completamente sostituite da quelle ad aghi, più versatili e moderne. Il nostro personaggio sarà comunque soddisfatto perché la decisione presa sarà la più adatta per lui, mentre la visione del problema da parte dell'amico non tiene conto delle necessità particolari del signor X, nonché la loro grande importanza: stampare testi molto lunghi in breve tempo e in modo nitido.

Ora, però, è necessario vedere nei particolari come, l'intuito del signor X di fronte al dubbio, è giunto alla soluzione vista, perché lo stesso procedimento viene effettuato dal programma.

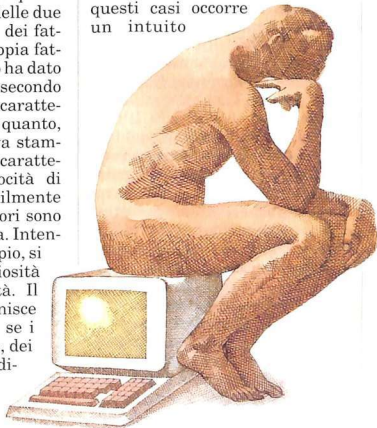
L'intuito del signor X ha preso in considerazione le due alterna-

tive e le ha messe in rapporto con i seguenti fattori: economicità, velocità di stampa, qualità di stampa, versatilità, silenziosità, praticità e semplicità d'uso.

A ogni fattore l'intuito ha dato un valore assoluto dipendente dalle necessità soggettive del signor X. In pratica ha stabilito quanto importante fosse ogni fattore per il signor X (e solo per lui). Sicuramente i fattori velocità di stampa e qualità di stampa hanno avuto la massima importanza. Il passo successivo è stato quello di confrontare ciascuna delle due alternative con ciascuno dei fattori. In pratica a ogni coppia fattore/alternativa, l'intuito ha dato un voto da zero a cento secondo quanto l'alternativa era caratterizzata dal fattore, cioè quanto, per esempio, l'alternativa stampante a margherita era caratterizzata dal fattore velocità di stampa. Come probabilmente avrete notato tutti i fattori sono espressi in forma positiva. Intendiamo dire che, per esempio, si è usato il termine silenziosità piuttosto che rumorosità. Il programma, infatti, fornisce risultati attendibili solo se i fattori sono, per così dire, dei pregi per le alternative disponibili. Ma torniamo al parallelo con l'intuito del signor X. Al ter-

mine della procedura descritta sopra, ciascuna alternativa ha avuto un punteggio.

L'alternativa con punteggio maggiore ha rappresentato la decisione da prendere. Il problema d'esempio era semplice e, forse, non era nemmeno necessario ricorrere all'intuito per risolverlo, anche perché i fattori del problema erano molto eterogenei e quelli più importanti facilmente individuabili. Provate, però, a pensare a problemi come quello dell'acquisto di un'automobile nuova o quello della scelta di due offerte di lavoro. Se, chi deve decidere, ha necessità chiare e particolari, probabilmente anche i fattori del problema risultano di facile valutazione, sempre che le alternative siano ben definite nelle loro caratteristiche. Scegliere un auto, però, è comunque un problema molto difficile poiché le alternative sono tante e, a prima vista, tutte equivalenti. Nel caso della scelta tra due proposte di lavoro, poi, al gran numero di fattori che entrano in gioco, va aggiunta l'influenza emotiva che una scelta del genere implica. In questi casi occorre un intuito





veramente eccezionale, capace di valutare moltissimi fattori e alternative senza farsi influenzare troppo dalle spinte emotive. Ecco che in questi casi risulta estremamente utile una pianificazione esterna. Intendiamo dire che sarebbe utilissimo disporre su un foglio le varie alternative, i fattori e i valori di ognuno di essi in rapporto alle alternative stesse e poi, calcolatore alla mano, elaborare il tutto per vedere per quale alternativa risulta più conveniente optare.

Il programma Decision Maker che trovate sulla cassetta allegata a questo numero di *Radio Elettronica & Computer* effettua per voi tutte le elaborazioni necessarie dopo avervi aiutato a pianificare il problema con il sistema illustrato prima, che, fra le altre cose, viene adottato dai grandi uomini d'affari prima di prendere importanti decisioni e insegnato nelle qualificatissime scuole americane per apprendisti manager.

### Il programma

Il programma può essere caricato in memoria dalla cassetta con LOAD"DECISION MAKER" o dal disco su cui lo avete trasferito con il programma Dsave V2 (che trovate anch'esso sulla cassetta) con LOAD"DECISION MAKER", 8,1. Subito dopo il lancio del programma, che potete effettuare digitando Run e premendo di seguito il tasto Return, compare una schermata introduttiva che scompare dopo breve tempo lasciando il posto a quella che vedete riprodotta in **figura 1**.

### •Alternative/Aggiungi

Questa funzione permette di inserire le alternative del problema. Il numero delle alternative che potete inserire è limitato a dieci, ma dovete tener conto che un problema con dieci alternative

apparentemente equivalenti è ben raro: forse nemmeno quello accennato sopra e relativo alla scelta di un'auto nuova è così ricco di alternative. Oltre a questo dovete tenere conto che quando selezionate questa opzione, cioè quando cominciate a usare il programma, dovete avere già operato una selezione delle alternative. Nel caso di problemi complessi come potrebbe essere quello appena citato dell'auto nuova, cercate almeno di suddividere le alternative in gruppi di dieci in modo da potere effettuare delle eliminatorie alla fine delle analisi di tutti i gruppi (vedi anche paragrafo Consigli).

### •Alternative/Cambia-Cancella-Lista

Le due funzioni che seguono servono per effettuare dei cambiamenti nell'archivio. Premendo il tasto 2 e, di seguito, il tasto Return, vi viene data l'opportunità di cambiare il contenuto dell'archivio Alternative. Nella parte alta dello schermo vengono mostrate le tre opzioni disponibili:

- cambia: attiva un input che richiede un'alternativa da sostituire a quella in oggetto;
- annulla: permette di togliere dall'archivio l'alternativa in oggetto;
- fine: torna alla schermata principale di **figura 1**.

Per passare all'alternativa successiva dovete premere il tasto Return.

Premendo il tasto 3 dal menù principale viene visualizzata semplicemente una lista di tutte le alternative in archivio.

### •Fattori/Aggiungi

Il passaggio più importante, co-

me avrete sicuramente capito, è quello di inserire i fattori. Il numero dei fattori è limitato a 11 per costringere l'utente a effettuare una semplificazione del problema dove è più necessario. Lo sforzo di semplificazione permette, in un certo senso, di purificare il problema, eliminando quei fattori che influiscono poco sull'analisi ma che l'emotività di chi deve decidere spinge a sopravvalutare. Le riflessioni imposte dal programma in questa fase hanno anche lo scopo inverso di fare emergere quei fattori del problema che, per le stesse ragioni citate sopra, potrebbero essere ignorati o sottovalutati. Generalmente i fattori per un problema di media complessità possono essere ridotti a circa sei. L'utilità del programma si rivela proprio in questa fase quando, cioè, l'utente si fa conscio degli estremi del problema, cosa che non sarebbe sicuramente avvenuta senza uno sforzo così mirato.

### •Fattori/Cambia-Cancella

La quinta funzione è del tutto analoga alla seconda vista prima.



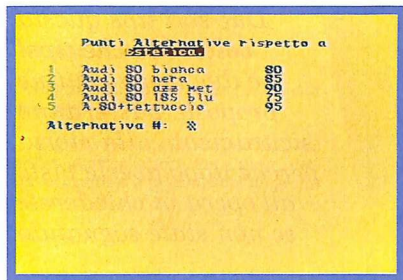


Figura 3.  
Elenco delle  
alternative e il  
fattore in esame

### • Fattori/Lista-Valori

La sesta funzione mostra la lista dei fattori immessi e permette di assegnare a ognuno di essi un valore. Questo valore rappresenta l'importanza che un certo fattore ha per l'utente. Per esempio se il fattore in questione è gradevolezza colore (è bene che i fattori abbiano una accezione positiva il più spiccata possibile) in rapporto al problema di quale auto comprare fra alcune disponibili dal concessionario di fiducia, si deve specificare il grado di importanza che ha il colore per chi deve scegliere. Ogni valore deve essere compreso fra zero e cento. Inoltre, almeno un fattore deve valere cento, come fattore più importante. Per questa ragione risulta comodo scegliere, per prima cosa, quale tra i fattori elencati è il più importante, valutarlo cento e, quindi, procedere inserendo gli altri valori.

§Per inserire un valore bisogna digitare il numero progressivo del fattore e premere, di seguito, il tasto Return. Fatto questo basta inserire il valore opportuno e premere ancora Return. La figura 2 riporta un elenco di fattori d'esempio.

### • Punti/Alternative

La settima opzione visualizza ancora l'elenco delle alternative

(figura 3) e, nella parte alta del video, uno dei fattori immessi dall'utente. In questa fase del programma occorre specificare un valore per ogni alternativa in rapporto al fattore visualizzato.

Per dare un valore a una certa alternativa basta digitare il numero progressivo che la individua, premere Return, digitare il valore e ancora Return. Il valore, in pratica, deve rappresentare quanto l'alternativa in questione è caratterizzata dal fattore.

Nel caso dell'automobile accennato sopra, l'auto X, essendo di un colore che piace molto all'utente, avrà una valutazione molto alta rispetto all'auto Y che, pur avendo un'ottimo motore, ha un colore sgradevole.

In rapporto al fattore prestazioni, invece, sarà l'auto Y ad avere votazione maggiore. Anche in questo caso i valori devono essere compresi fra zero e cento.

### • Elaborazione/Risultati

L'opzione otto comunica al programma che tutto è pronto per l'elaborazione.

A questo punto sul video compare ancora l'elenco delle alternative e una votazione per ciascuna di esse. L'alternativa con votazione maggiore è quella per cui l'utente ha maggior convenienza a optare.

### • Elaborazione/Stampa

È possibile ottenere un stampato di tutto il procedimento selezionando l'opzione numero nove.

### • File Dati

Le successive tre funzioni permettono di salvare l'archivio di alternative e fattori su disco o su cassetta, oppure di visualizzare la directory di un disco (opzione 12). Il problema salvato su disco o nastro può, naturalmente, essere richiamato successivamente, modificato ed elaborato ancora.

### Consigli

Il programma Decision Maker potrebbe apparire di capacità limitata perché può archiviare solo dieci alternative e 11 fattori.

In realtà ogni problema, anche molto complesso, può essere suddiviso in sottoproblemi.

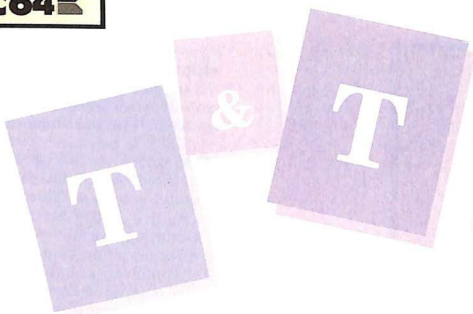
Per continuare con l'esempio automobilistico supponiamo di dover scegliere un'auto e di non avere nemmeno un'idea generale di quale sia il modello più adatto. Nel mare di alternative, quindi, andando per esclusione, potremo isolare una trentina di modelli possibili.

Poi le divideremo in gruppi secondo i fattori generali che le caratterizzano (economicità, prestazioni, valore estetico eccetera). In questo modo, non solo avremo cominciato a chiarirci le idee, ma avremo ottenuto dei gruppi di alternative su cui operare con l'aiuto del computer.

Per ogni gruppo metteremo in esame fattori più specifici e (se fosse necessario), opereremo ulteriori suddivisioni.

Alla fine otterremo un ristretto numero di alternative molto caratterizzate, fra cui sarà facilissimo scegliere con l'aiuto del programma e potremo prendere facilmente una decisione.

Raffaele Zanini



*Due soli tips questa volta. Ma che tips! Le due mega routine proposte passeranno sicuramente alla storia perché dopo averle viste all'opera vi chiederete se non state sognando*

## Attenti a questi due

Con l'ultimo numero dell'anno vi proponiamo due super tips indimenticabili e indispensabili per chi vuole il massimo dal buon vecchio C64.

### Provic

Definire in poche parole questa incredibile utility è un'impresa impossibile. Vi basti sapere che quadruplica la potenza del Vic II, il chip che gestisce la grafica del C64. Più precisamente, Provic vi metterà a disposizione ben 4 Vic II con le stesse caratteristiche e potenzialità dell'originale. A ogni pseudo Vic potrete assegnare una porzione di dimensioni variabili dello schermo e all'interno di questa potrete attivare un modo grafico qualsiasi indipendentemente da quelli attivi nelle altre porzioni di schermo controllate dagli altri pseudo Vic. Ecco un esempio di ciò che sarete in grado di fare:

- avere 32 sprite contemporaneamente sullo schermo;
- avere una finestra di hi-res sul modo testo;
- attivare contemporaneamente più set di caratteri e utilizzarli insieme.

Per utilizzare Provic dovete caricarlo dalla cassetta con LOAD "PROVIC" e dare il consueto Run. Il **listato 1** è relativo a un esempio di ciò che si può fare con questo eccezionale mega-tip. Studiatelo con attenzione insieme alla descrizione che segue.

Per attivare Provic dovete usare l'istruzione: sys 52544.

Per disattivare Provic dovete usare l'istruzione: sys 52970 oppure premere contemporaneamente Run Stop e Restore. Indirizzi base dei pseudo Vic, che d'ora in poi chiameremo Pvic:

- Pvic 1: 52992
- Pvic 2: 53056
- Pvic 3: 53120
- Pvic 4: 53184

### Funzione dei registri di ogni Pvic

Il numero del registro rappresenta l'offset da aggiungere all'indirizzo base del Pvic per trovare l'indirizzo del registro stesso.



- Registro 0: ascissa sprite 0.
- Registro 1: ordinata sprite 0.
- Registri dal 2 al 15: ascissa e ordinata degli sprite 1 - 7.

• Registro 16: msb (most significant bit) per l'ascissa degli sprite. Per ulteriori dettagli consultate il manuale del programmatore del C64.

- Registro 17:

- bit 0 - 2: controllano lo scrolling verticale
- bit 3: permette di selezionare la visualizzazione su 24 o 25 linee
- bit 4: attiva/ disattiva lo screen blanking
- bit 5: attiva/ disattiva l'alta risoluzione
- bit 6: attiva/ disattiva l'extended color mode
- bit 7: ottavo bit del registro di confronto con la posizione del raster

• Registro 18: è il cosiddetto raster compare register. Il contenuto di questo registro viene confrontato con la posizione del raster e se i due valori coincidono allora viene generata un'interruzione. In pratica questo registro serve per specificare la porzione di schermo controllata dal Pvic. Assicuratevi che sia sempre Pvic 1 < Pvic 2 < Pvic 3 < Pvic 4 (con Pvic n si indica il valore del registro 18 del Pvic n).

Il Pvic 1 controlla la porzione di schermo che va fino alla linea specificata nel registro 18 del Pvic 2.

Il Pvic 2 a sua volta controlla la porzione di schermo che va dalla linea indicata nel suo registro 18 e quella indicata nel registro 18 del Pvic 3. Un discorso analogo vale per i Pvic rimanenti. La situazione sembra abbastanza complicata ma in realtà non lo è. Tutto verrà comunque chiarito molto meglio dal demo.

• Registri 19 e 20: contengono l'ascissa e l'ordinata dell'impulso della penna ottica.

• Registro 21: registro di attivazione degli sprite. Ogni bit di questo registro controlla uno sprite e se è posto a 1 lo sprite corrispondente è attivato.

### Tavola 1.

Tasto	con Shift	con Commodore
A	print	print#
B	and	or
C	chr\$	asc
D	read	data
E	get	end
F	for	next
G	gosub	return
H	to	step
I	input	input#
J	goto	on
K	dim	restore
L	load	save
M	mid\$	len
N	int	rnd
O	open	close
P	poke	peek
Q	tab()	spc()
R	right\$	left\$
S	str\$	val
T	if	then
U	tan	sqr
V	verify	cmd
W	def	fn
X	list	fre
Y	sin	cos
Z	run	sys

- Registro 22:

- bit 0 - 2: controllano lo scrolling orizzontale dello schermo

- bit 3: attiva/ disattiva la visualizzazione su 38 o 40 colonne

- bit 4: attiva/ disattiva il modo multicolore per l'alta risoluzione e il modo testo

- bit 5: video chip reset

- bit 6 - 7: inutilizzati

• Registro 23: controlla l'espansione degli sprite in direzione verticale. Ogni bit di questo registro è associato a uno sprite e se è posto a 1 lo sprite corrispondente viene espanso in direzione verticale raddoppiando la propria altezza.

• Registro 24: controlla il banco di lavoro del Vic II, il chip che gestisce la grafica del Commodore 64.

- bit 0: inutilizzato

## Listato 1

```

1 rem -----
2 rem - - - - -
3 rem - demo provic - -
4 rem - - - - -
5 rem -----
6 :
99 rem
100 rem inizializza i pvic
101 rem
110 pl=52990; p2=53056; p3=53120; p4=53184; rem indirizzi base pvic
120 pokepl=21,255;pokepl+24,22;pokepl+47,1;pokepl+27,255;pokep2+21,255
130 pokep2+17,59;pokep2+24,24;pokep2+32,7;pokep2+47,1;pokep2+27,255
140 pokep3+21,255;pokep3+32,8;pokep3+47,1;pokep3+27,255;pokep4+32,5
150 pokep4+21,255;pokep4+24,22;pokep4+47,1;pokep4+18,230;pokep4+27,255
199 rem
200 rem legge dati sprite
201 rem
210 restore
220 fora=0to126
230 read wert
240 poke$32+a,wert
250 nexta
299 rem
300 rem
301 rem
310 print"<CLEAR><WHITE><DOWN><RS ON><RIGHT 7>P R O V I C<SPC 3>6 4<SPC 3>D E M O"
320 print"<DOWN 2> Caratteri CARATTERI <DOWN>:"
330 fora=0to7:print"<BLUE>+++++<WHITE>";inexta
335 fora=0to14:poke1289+a,34;nexta
340 print"      hires
350 print"<DOWN>      caratteri grafici
360 print"<DOWN> QWERTYUIO<SHIFT @><SHIFT *><SHIFT ^><SHIFT ~><LKHJFSAZXCVBNM<CMDR Q><CMDR W><CMDR E><CMDR R><CMDR T><CMDR F><CMDR G><CMDR Z><CMDR X><CMDR C><CMDR V><CMDR B>"
370 print"<DOWN 5>      * caratteri Minuscoll *UP 3":printtab(6);
390 sys52544:rem provic-64 attivato
399 rem
400 rem grafica hi-res
401 rem
410 fora=10110to12360;pokea,0;nexta
420 fora=0to6<SHIFT *>step.08
430 x=3+a/.05;y=77-1*int(a)-9*cos(a/.7)
440 av=8192+320*int(y/8)+(y and 7)+8*int(x/8)
450 pokeav,peek(av) or 2*(7-(x and 7))
460 nexta
470 la$="** ciao ciao ciao ** demo demo demo demo ____ "
480 la$=la$+**** P R O V I C<SHIFT /SPC 3>6 4 ____ "
490 la$=la$+left$(la$,25);r=53266
499 rem
500 rem
501 rem
510 rem setta sprite
511 rem
520 fora=0to7
530 pokepl+2*a,30+24*a+7*rnd(1);pokepl+2*a+1,60+6*rnd(1)
540 pokepl+39*a, rnd(1)*16;pokepl+48*a,13.5+rnd(1)
550 pokep2+2*a,30+24*a+7*rnd(1);pokep2+12*a+1,110+6*rnd(1)
560 pokep2+39*a, rnd(1)*16;pokep2+48*a,13.5+rnd(1)
570 pokep3+2*a,30+24*a+7*rnd(1);pokep3+2*a+1,160+6*rnd(1)
580 pokep3+39*a, rnd(1)*16;pokep3+48*a,13.5+rnd(1)
590 pokep4+2*a,30+24*a+7*rnd(1);pokep4+2*a+1,210+6*rnd(1)
600 pokep4+39*a, rnd(1)*16;pokep4+48*a,13.5+rnd(1)
610 nexta
619 :
620 :
621 :
625 forlp=ltolen(la$)-25
630 lz=lz-1;filz=0;thenpokep4+22,lz or 8:fora=0to9;nexta:goto630
640 printtab(6);wait53265,128;wait53266,64;poke53206,15;printlf0:print"<UP 2>"
660 lz=1;lf$=mid$(la$,lp,25)
670 nextlp
680 geta$lf0$=""then500
690 sys52970:rem disattiva provic
999 rem
1000 rem dati sprite
1001 rem
1002 data0,0,0,0,126,0,1,255,128,7,255,224,15,255,240,15,253,240,31,255,248
1003 data31,255,248,63,255,252,63,255,252,63,243,252,63,252,0,63,255,252,63
1004 data255,252,31,255,248,31,255,248,15,255,240,15,255,240,7,255,240,15,255
1005 data126,0,126,0,0,0,0,0,126,0,1,255,128,7,255,224,15,255,240,15,253,240,31,255
1006 data240,31,255,248,11,255,248,63,255,240,63,255,0,63,240,0,63,832,0,63
1007 data255,0,63,255,224,31,255,248,31,255,248,15,255,240,15,255,240,7,255
1008 data224,1,255,128,0,126,0

```



- bit 1 - 3: indicano la posizione (offset) dell'immagine dei caratteri nel banco  
- bit 4 - 7: indicano la posizione (offset) della memoria video all'interno del banco

• Registro 25 e 26: controllano l'abilitazione delle sorgenti di interrupt e segnalano l'arrivo di una richiesta di interrupt.

• Registro 27: controlla la priorità degli sprite rispetto ai caratteri. Ogni bit del registro è associato a uno sprite e se è settato lo sprite corrispondente avrà priorità inferiore rispetto ai caratteri.

• Registro 28: attiva/ disattiva il modo multicolore per gli sprite. Anche in questo caso ogni bit controlla uno sprite.

• Registro 29: attiva/ disattiva l'espansione in direzione orizzontale degli sprite.

• Registro 30: registro di collisione fra sprite.

• Registro 31: registro di collisione fra sprite e carattere.

• Registro 32: contiene il colore del bordo della porzione di schermo controllata dal Pvic.

• Registro 33: contiene il colore del fondo della porzione di schermo controllata dal Pvic. xInoltre contiene il colore 0 per l'alta risoluzione, gli sprite e i caratteri multicolor.

• Registro 34: contiene il colore 1 del modo multicolor.

• Registro 35: contiene il colore 2 del modo multicolor.

• Registro 36: contiene il colore 3 del modo multicolor.

• Registri 37 e 38: contengono il colore 0 e 1 degli sprite multicolor.

• Registri 39 - 46: contengono il colore degli sprite.

• Registri 48 - 55: contengono i puntatori ai blocchi sprite.

## Listato 2

```

1 rem -----
2 rem - - - - -
3 rem - 64 word -
4 rem - - - - -
5 rem -----
6 :
100 if peek(peek(56)*256)>120 thenpoke56,peek(56)-1:clr
110 hi=peek(56):base=hi*256
120 printchr$(147)"reading data"
130 forad=0 to 211:read by
140 poke base+ad,by:next
150 :
200 pokebase+26,hi:pokebase=81,hi
210 poke base+123,hi:pokebase=133,hi
220 :
230 printchr$(147)tab(15)""**64word**":print
250 print"per attivare/ disattivare: sys ",base:print:print
260 printchr$(145);chr$(145):
270 data 120,173,143,2,201,32
280 data 208,12,169,220,141,143
290 data 2,169,72,141,144,2
300 data 88,96,169,32,141,143
310 data 2,169,0,141,144,2
320 data 88,96,165,212,208,117
330 data 173,141,2,201,3,176
340 data 110,201,0,240,106,169
350 data 194,133,245,169,235,133
360 data 246,165,215,201,193,144
370 data 98,201,219,176,91,56
380 data 233,193,174,141,2,224
390 data 2,208,3,24,105,26
400 data 170,189,159,0,162,0
410 data 134,198,170,160,158,132
420 data 34,160,160,132,35,160
430 data 0,10,240,16,202,16
440 data 12,230,34,208,2,230
450 data 35,177,34,16,246,48
460 data 241,200,177,34,48,17
470 data 8,142,211,0,230,198
480 data 166,198,157,119,2,174
490 data 211,0,40,208,234,230
500 data 198,166,198,41,127,157
510 data 119,2,230,198,169,20
520 data 141,119,2,76,72,235
530 data 76,224,234
570 data 153,175,199,135,161,129
580 data 141,164,133,137,134,147
590 data 202,181,159,151,163,201
600 data 196,139,192,149,150,155
610 data 191,138,152,176,198,131,128,130
620 data 142,169,132,145,140,148
630 data 195,187,160,194,186,200
640 data 197,167,186,157,165,184
650 data 190,158,0

```

## 64 word

Anche questo tip è estremamente potente e vi aiuterà moltissimo nella realizzazione dei programmi. 64 word vi permetterà infatti di scrivere qualsiasi istruzione Basic con la semplice pressione di una combinazione di tasti (Commodore o Shift più un tasto). Per utilizzare 64 word dovete copiare il **listato 2** e dare il Run.

Le istruzioni per attivare e disattivare la routine vengono visualizzate dalla routine stessa. La **tavola 1** riassume l'elenco delle combinazioni di tasti necessarie per ottenere ogni istruzione Basic.

Daniele Maggio





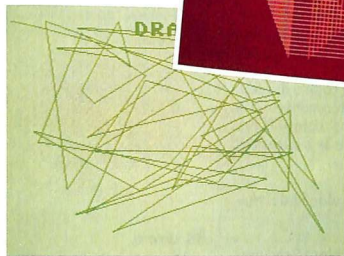
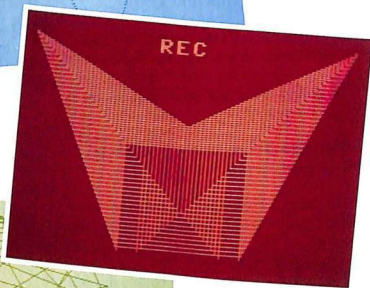
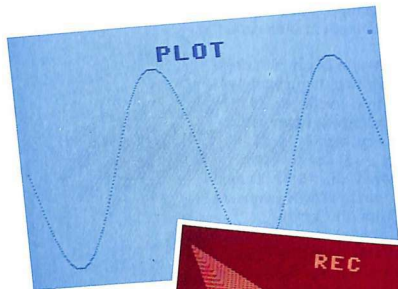
# Lo sprint finale

*Ultima e densissima puntata di Ultra Basic. Sono di scena le potentissime istruzioni per la gestione dell'alta risoluzione e del sid e le innumerevoli funzioni di aiuto alla programmazione*

Nelle precedenti puntate si è parlato di grafica ma non si è mai accennato all'alta risoluzione. Questo potrebbe avervi fatto sospettare che Ultra Basic non possieda istruzioni per la gestione dell'hi-res. Ebbene, in quest'ultima puntata verrà spazzato via ogni dubbio di questo genere. Oltre che di alta risoluzione parleremo di musica e di funzioni d'aiuto alla programmazione, e vedrete che anche in questi campi Ultra Basic dà il massimo e pone in ombra ogni altra espansione. Iniziamo ora con le ultime istruzioni.

## Alta risoluzione

• **Hires:** attiva l'alta risoluzione cancellando la pagina grafica. La sintassi è: hires l, f dove l rappresenta il colore di linea (cioè il colore con cui saranno visualizzati i punti) e f quello di fondo dell'alta risoluzione.



• **Graphic:** attiva l'alta risoluzione senza cancellare la pagina grafica.

La sintassi è: graphic (l, f). La parentesi in questo caso sta a indicare che i due parametri sono opzionali. Essi hanno lo stesso significato degli omonimi parametri dell'istruzione precedente.

• **Hicol:** questa è una istruzione particolarmente utile per creare

effetti spettacolari in alta risoluzione. Permette infatti di cambiare il colore di linea e di fondo agendo localmente sui singoli punti (il colore di fondo di un punto è quello della matrice di 8 x 8 punti che lo contiene).

La sintassi è hicol l, f e il significato dei parametri è lo stesso degli omonimi parametri dell'istruzione Hires. Ecco un esempio di applicazione:



```
10 hires 0,6
20 fort=0 to 10:hicol t.6:for y=0 to
10:draw 1,1+y+8*t,320,1+y+8*t,
1:next:next1000 goto 1000
```

Come noterete lanciando l'esempio, l'istruzione Hicol agisce sui punti acceduti dalle istruzioni che la seguono nel listato.

- **Nrm:** riattiva il modo testo.

- **Plot:** setta un punto della pagina grafica. La sintassi è: plot x, y, m. I primi due parametri sono le coordinate del punto mentre l'ultimo parametro ha il seguente significato:

```
m=0: cancella il punto
m=1: setta il punto
m=2: inverte il punto
```

- **Draw:** traccia una linea. La sintassi è: draw x1, y1, x2, y2, m. I primi quattro parametri rappresentano le coordinate degli estremi del segmento mentre m ha lo stesso significato dell'omonimo parametro dell'istruzione Plot.

- **Rec:** traccia un rettangolo. La sintassi è: rec x1, y1, x2, y2, m. I primi quattro parametri rappresentano rispettivamente le coordinate dell'angolo superiore sinistro e inferiore destro del rettangolo. Il parametro m ha il solito significato.

- **Box:** disegna un rettangolo pieno. La sintassi è identica a quella dell'istruzione precedente.

- **Circle:** disegna un'ellisse. La sintassi è: circle xc, yc, xr, yr, m. I primi due parametri rappresentano le coordinate del centro dell'ellisse. I due parametri successivi sono rispettivamente il semiasse orizzontale e verticale dell'ellisse (per disegnare un cerchio basta fare in modo che sia  $xr=yr$ ). L'ultimo parametro ha il solito significato.

## Tavola 1

- 0 joystick fermo
- 1 joystick verso l'alto
- 2 joystick in diagonale verso l'alto a destra
- 3 joystick verso destra
- 4 joystick in diagonale verso il basso a destra
- 5 joystick verso il basso
- 6 joystick in diagonale verso il basso a sinistra
- 7 joystick verso sinistra
- 8 joystick in diagonale verso l'alto a sinistra

*Nota: se viene premuto il tasto Fire i valori sono aumentati di 128.*

- **Arc:** è la classica istruzione che va vista all'opera per capire come funziona. La sintassi è: arc xc, yc, xr, yr, init, fine, p, m. Ecco un esempio d'uso:

```
10 hires 0,1
20 arc 150,90,70,80,270,450,12,1
30 arc 160,100,90,90,0,360,72,1
40 getkey a$
```

L'istruzione permette di unire fra loro dei punti di un'ellisse. I primi quattro parametri hanno lo stesso significato degli omonimi parametri dell'istruzione Circle. I parametri init e fine individuano il primo e l'ultimo punto da unire. Il parametro p rappresenta la distanza, in gradi, fra due punti consecutivi. L'ultimo parametro ha il solito significato.

- **Paint:** riempie una regione chiusa di cui viene indicato un punto interno. La sintassi è: paint x, y, m.

- **Sshape:** carica in una variabile una porzione rettangolare della pagina grafica. La sintassi è: sshape a\$, x, y, l, a. Il primo parametro deve essere una variabile stringa. In essa verrà caricata la porzione di alta risoluzione. Gli altri quattro parametri permettono di specificare l'area da salvare: x e y sono le coordinate dell'angolo superiore sinistro della zona

mentre l e a sono rispettivamente la larghezza e l'altezza in pixel.

La quantità di caratteri necessari per salvare una determinata regione della pagina grafica è data da:  $l^*a/8+4$ . Se questo valore supera 255 verrà segnalato l'errore string too long.

- **Gshape:** effettua l'operazione contraria a quella svolta dall'istruzione precedente, cioè visualizza il contenuto della stringa in una determinata posizione all'interno della pagina grafica. La sintassi è: gshape a\$, x, y, m. Il primo parametro è la stringa contenente la regione da visualizzare. I due parametri successivi rappresentano le coordinate del punto da cui iniziare a copiare. L'ultimo parametro ha il solito significato. Ecco un esempio d'uso di queste ultime due istruzioni:

```
10 hires 0,1
20 nchar:sshape x$,0,0,24,24
30 gshape x$,100,170,1
40 getkey a$
```

- **Sblock:** copia una regione rettangolare della pagina grafica in uno sprite. La sintassi è: sblock b, d, x, y. I primi due parametri hanno lo stesso significato degli omonimi parametri dell'istruzione Create 1. Gli ultimi due parametri rappresentano le coordinate dell'angolo superiore sinistro del-



la regione da copiare nello sprite. Le dimensioni della regione sono fisse e sono uguali a quelle di uno sprite.

• **Gblock:** copia il contenuto di uno sprite nella pagina grafica.

La sintassi è: gblock b, d, x, y, gx, gy, m. I primi quattro parametri permettono di specificare il blocco sprite e la posizione in cui copiarlo nella pagina grafica.

I parametri gx e gy controllano l'espansione dello sprite rispettivamente in direzione orizzontale e verticale. Ecco i valori che possono assumere:

- 1= sprite non espanso
- 2= sprite espanso

L'ultimo parametro ha il solito significato. Ecco un esempio d'uso di questa istruzione:

```
10 hires 0,1:nchar
20 sblock 13,0,0,0
30 hires 0,1
40 gblock 13,0,100,100,2,1,1
```

• **Text:** visualizza una stringa nella pagina grafica. La sintassi è: text x, y, a\$, gx, gy, a, m. I primi due parametri rappresentano le coordinate da cui parte la visualizzazione della stringa. A\$ è la stringa da visualizzare.

I due parametri successivi rappresentano rispettivamente la larghezza e l'altezza dei caratteri in multipli delle dimensioni normali. Il parametro a rappresenta la distanza in punti fra due caratteri consecutivi. L'ultimo parametro ha il solito significato.

• **Window:** apre una fine-

stra in alta risoluzione nel modo testo. La sintassi è: window (x, y). I due parametri, opzionali, rappresentano la prima e l'ultima linea della pagina grafica da visualizzare in modo testo. Per chiudere la finestra si deve usare l'istruzione senza parametri.

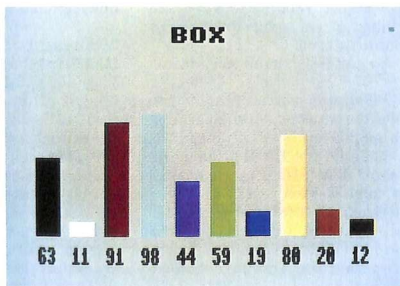
• **Revers:** inverte il contenuto della pagina grafica.

• **Scnclr:** inizializza la pagina grafica e la memoria del colore.

• **grol:** cambia il colore di linea e di fondo di una regione della pagina grafica. La sintassi è: grol x, y, l, a, l, f. I primi quattro parametri rappresentano l'ampiezza, in caratteri, della regione da trattare: i primi due parametri sono le coordinate dell'angolo superiore sinistro mentre i due parametri successivi sono la larghezza e l'altezza. Gli ultimi due parametri sono il nuovo colore di linea e di fondo per la regione indicata. Ecco un esempio d'uso:

```
10 hires 0,1:nchar
20 text 10,160,"demo per
grol",2,2,16,1
30 loop
40 grol 33*rnd(0),18*rnd(0)
,8,8,16*rnd(0), 16*rnd(0)50
endloop
```

• **Test:** è una funzione che per-



mette di conoscere lo stato di un punto della pagina grafica.

La sintassi è: test (x,y). I due parametri sono le coordinate del punto da testare. La funzione a ritorna in uno dei seguenti valori:

- 1: il punto è settato
- 0: il punto non è settato

Nelle figure 1, 2 e 3 sono riportate alcune delle schermate prodotte dal demo grafica demo registrato sulla cassetta acclusa a questo numero di *Radio elettronica & Computer*. Il demo comprende numerosissimi esempi di uso delle istruzioni grafiche e dovrebbe eliminare ogni dubbio sul loro comportamento.

## Musica

• **Vol:** fissa il volume. La sintassi è: vol n dove n, che rappresenta il volume e deve essere un intero fra zero e 15

• **Envelope:** fissa i parametri a, d, s e r per una delle tre voci. La sintassi è: envelope v, a, d, s, r. Il primo parametro, che rappresenta la voce, deve essere un intero fra uno e tre.

• **Wave:** fissa la forma d'onda di una voce.

La sintassi è: wave v, %xxxxxxx. Il primo parametro rappresenta la voce. Il secondo parametro deve essere un numero binario. Ecco il significato di ogni bit:

- 0: rumore bianco
- 1: dente di sega
- 2: testbit
- 3: sincronizzazione
- 4: impulso ad ampiezza variabile
- 5: triangolare
- 6: modulazione
- 7: gatebit

• **Sound:** fissa i valori di fre-



## Tavola 2

<i>until without repeat</i>	31
<i>end loop without loop</i>	32
<i>missing end loop</i>	33
<i>missing bend</i>	34
<i>proc not found</i>	35
<i>end proc without exec</i>	36
<i>can't resume</i>	37
<i>out of range</i>	38
<i>illegal string lenght</i>	39
<i>wrong string</i>	40
<i>missing endcase</i>	41
<i>unknown variable</i>	42

quenza e lunghezza del tono di una voce. La sintassi è: sound v, f (.). Il primo parametro è la voce. Il secondo parametro la frequenza (0-65535) e l'ultimo parametro, opzionale, è la lunghezza (0-65535=max 30 secondi).

- **Pulse:** fissa l'ampiezza dell'impulso per l'onda a impulso variabile. La sintassi è: pulse v, x.

Il primo parametro è la voce e il secondo l'ampiezza dell'impulso (0-4095).

- **Clrsid:** inizializza il sid.

- **Music:** prepara una serie di note. La sintassi è: music v, a\$.

Il primo parametro è la voce mentre il secondo rappresenta la stringa di note, da specificare nella forma indicata sul manuale dell'utente.

- **Play:** suona le note indicate con l'istruzione precedente. La sintassi è: play m. Il parametro m può assumere i seguenti valori:

m=0: smette di suonare

m=1: inizia a suonare

m=2: suona le note da interrupt

- **Beep:** emette un beep. La sintassi è: beep n dove n è il numero di beep da emettere.

Sulla cassetta acclusa a *Radio*

*Elettronica & Computer* si trova il programma dimostrativo musica demo che mostra alcuni esempi d'uso di quest'ultimo gruppo di istruzioni.

### Aiuto alla programmazione

- **Swap:** scambia il contenuto di due variabili (dello stesso tipo, naturalmente). La sintassi è: swap v1, v2.

- **Doke:** effettua un doppio poka. La sintassi è: doke i, v. L'istruzione poka nella locazione i il byte basso di v e nella locazione i+1 il byte alto di v.

- **Bit on:** setta un bit di una determinata locazione. La sintassi è: bit on i, nb. Il primo parametro è l'indirizzo della locazione, mentre il secondo è il bit da settare (deve essere un intero fra uno e otto).

- **Bit off:** pone a zero un bit di una determinata locazione. La sintassi è identica a quella dell'istruzione precedente.

- **Kill:** resetta il computer.

- **Break off:** disattiva il tasto Run/Stop.

- **Break on:** riattiva il tasto Run/Stop.

- **Fix:** fissa la locazione su cui avrà effetto l'istruzione Vary (descritta in seguito). La sintassi è: fix i dove i è l'indirizzo della locazione.

- **Vary:** permette di alternare, da interrupt, due valori diversi nella locazione indicata con Fix, la sintassi è: vary v1, v2, l.

I primi due parametri sono i valori da alternare nella locazione indicata con Fix. L'ultimo parametro è la velocità con cui devono essere alternati i valori:

0=velocità massima  
255=velocità minima

Per disattivare questo meccanismo dovete usare Vary senza parametri.

- **Pause:** pone in pausa il computer. La sintassi è: pause n dove n è la durata in secondi della pausa.

- **Commands:** visualizza l'elenco delle istruzioni messe a disposizione dall'espansione.

- **Comshift off:** disattiva i tasti Commodore e Shift.

- **Comshift on:** riattiva i tasti Commodore e Shift.

- **Rapid on:** attiva il repeat automatico dei tasti.

- **Rapid off:** disattiva il repeat automatico.

- **Screen off:** rende invisibile, senza cancellarlo, il contenuto dello schermo.

- **Screen on:** riporta lo schermo alla normalità.

- **Local:** permette di definire



## ESPANSIONI

delle variabili locali. La sintassi è: locale, a\$, c\$, x%,.... Le variabili a, a\$, c\$, x% ecc sono dichiarate locali e saranno visibili sino a quando non verranno dichiarate altre variabili globali con lo stesso nome. Le variabili locali si sovrappongono alle, eventuali, variabili globali con lo stesso nome.

- **Global:** permette di definire delle variabili globali. La sintassi è identica a quella dell'istruzione precedente.

- **Go 64:** disattiva l'espansione. Dato questo comando, dovete premere y per confermare la decisione e n per annullarla.

- **Lomem:** fissa il limite inferiore della Ram Basic. La sintassi è: lomem i dove i è l'indirizzo della prima locazione che sarà disponibile per i programmi Basic.

- **Himem:** fissa il limite superiore della Ram Basic. La sintassi è: himem i dove i rappresenta l'indirizzo dell'ultima locazione disponibile per i programmi Basic.

- **Cleol:** cancella la linea dalla posizione del cursore fino alla fine dello schermo.

- **Bckgnd:** attiva il modo extended color mode e fissa i colori utilizzabili. La sintassi è: bckgnd c1, c2, c3, c4. I parametri rappresentano i quattro colori utilizzabili

nel modo grafico attivato. Per disattivare l'extended color mode si deve usare l'istruzione senza parametri.

- **[ ]:** permette di inserire delle linee di commento. Ecco un esempio d'uso:

```
10 [[ inizio commento
20 questo
30 è un testo
40 di esempio ]]
50 [[inizio ciclo]] fort=0 to 100
60 print t
70 [[fine ciclo]] next
```

### Funzioni

- **Ypos:** ritorna la riga in cui si trova il cursore.

- **Penx:** ritorna l'ascissa del punto indicato con la penna ottica.

- **Peny:** ritorna l'ordinata del punto indicato con la penna ottica

- **Inkey:** ritorna il numero del tasto funzione premuto.

- **Errln:** ritorna il numero della linea Basic che ha causato l'errore segnalato dall'interprete.

- **Errn:** ritorna il codice dell'errore riscontrato.

- **Err\$:** ritorna la stringa d'errore corrispondente a un determi-

nato codice d'errore. La sintassi è: err\$(n) dove n è il codice dell'errore.

- **Ascii:** ritorna il codice ascii corrispondente a un determinato codice video. La sintassi è: ascii(n) dove n è un codice video.

- **Bsc:** ritorna il codice video corrispondente a un determinato codice Ascii.

- **Dec:** ritorna l'equivalente decimale di un numero espresso in notazione binaria o esadecimale. La sintassi è: dec(a\$) dove a\$ può essere o una stringa di cifre binarie o una stringa di cifre esadecimali. Ecco qualche esempio d'uso:

```
print dec ("00000011")
print dec ("aa")
print dec ("aab1")
```

Ricordatevi che la stringa di cifre binarie deve sempre essere lunga otto caratteri mentre quella di cifre esadecimali può essere di zero, uno, due, tre o quattro caratteri.

- **Hex\$:** ritorna la stringa di cifre esadecimali corrispondenti a un numero decimale. La sintassi è: hex\$(n) dove n è il valore decimale da convertire.

- **Bin\$:** ritorna la stringa di cifre binarie corrispondente a un particolare valore decimale. La sintassi è: bin\$(n) dove n è un valore decimale.

- **Row:** ritorna tutti i caratteri compresi in un certo range di codici Ascii.

La sintassi è: row(a, b) dove a e b rappresentano il primo e l'ultimo codice Ascii del range.

Ecco un esempio d'uso (visualizza tutti i caratteri compresi fra la a e la z):

```
print row(asc("a"),asc("z"))
```

## Tavola 3

\$0000 - \$0400: pagina zero più pagine 1-3. Riservate  
 \$0400 - \$0800: posizione normale della mem. video  
 \$0800 - \$8000: Ram Basic  
 \$8000 - \$c000: Ultra Basic  
 \$c000 - \$ca00: spazio libero  
 \$ca00 - \$cc00: spazio riservato all'espansione  
 \$cc00 - \$d000: Ram colore (hi-res o modo testo)  
 \$d000 - \$e000: spazio riservato all'espansione  
 \$e000 - \$ffff: pagina grafica oppure dati per caratteri prog



- **Dup:** ritorna il multiplo intero di una stringa. La sintassi è: dup(a\$, n). Ecco un esempio d'uso:

```
print dup("*. ",10)
```

- **Insert:** inserisce una stringa all'interno di un'altra stringa a partire da una determinata posizione e ritorna la stringa risultante.

La sintassi è: insert(a\$, p, b\$). Il primo parametro è la stringa da inserire, il secondo parametro è la posizione in cui deve avvenire l'inserimento e b\$ è la stringa in cui avviene l'inserimento. Ecco un esempio:

```
b$="123456789"
print insert("xx",3,b$)
```

- **Inst:** sostituisce una stringa una serie di caratteri di un'altra stringa. In pratica questa istruzione opera come la precedente con la sola differenza che la stringa viene sostituita a partire da una certa posizione ai caratteri della stringa destinazione. La sintassi è identica a quella dell'istruzione precedente. Ecco un esempio:

```
a$=dup("a",255)
b$=dup("x",255)
print inst(a$,1,b$)
```

se si cerca di eseguire questa sequenza di istruzioni sostituendo Insert a Inst verrà visualizzato un messaggio d'errore.

- **Place:** ritorna la posizione di una stringa all'interno di un'altra stringa.

La sintassi è: place (a\$, b\$ [,c]). Il primo parametro è la stringa da cercare mentre b\$ è la stringa in cui viene effettuata la ricerca. Il terzo parametro, opzionale, indica la posizione da cui deve iniziare la ricerca (per default questo valore è uno). Ecco qualche esempio:

```
a$="qwerty"
print place ("q",a$,3)
print place ("q",a$)
```

- **Using:** è utilissima per formattare l'output di valori numerici. Ecco un esempio d'uso:

```
10 loop
20 x=100*rnd(0)
30 print using(###,##,x)
40 end loop
```

I caratteri # e il segno di punto (.) permettono di decidere il formato con cui saranno visualizzati i valori. Ogni # indica una cifra e il punto (.) indica la posizione del punto decimale.

- **By:** posiziona il cursore. La sintassi è: by (c, l) dove c è la colonna e l la linea in cui spostare il cursore. Ecco un esempio d'uso:

```
print by (10,10);"ciao"
```

- **Deek:** equivale a un doppio Peek. La sintassi è: deek(i). Il valore ritornato è: peek(i)+256\*peek(i+1).

- **Ceek:** esegue una Peek nella Ram sotto la Rom. La sintassi è uguale a quella dell'istruzione Peek. Ecco un esempio:

```
print peek(60000),ceek(60000)
```

- **Check:** testa lo stato di una periferica. La sintassi è: check(n) dove n è il codice della device da testare. Ecco un esempio d'uso:

```
10 loop:exit if check(8)=1
20 print"Il drive 8 non è collegato correttamente"
30 getkey a$:end loop
40 print "ok"
```

- **Root:** ritorna la radice n-esima di un numero.

La sintassi è: root(r, v). Il primo parametro è l'indice della radice mentre il secondo è il radicante.

Entrambi i valori possono essere interi o reali.

- **Round:** arrotonda un valore a una certa cifra decimale. La sintassi è: round(v, c). Il primo parametro è il valore da arrotondare e il secondo è la cifra a cui effettuare l'arrotondamento.

- **Eval:** valuta un'espressione numerica contenuta in una variabile stringa. La sintassi è: eval(a\$) dove a\$ è la stringa contenente l'espressione. Ecco un esempio:

```
print eval("3*2*cos(1.5)+2")
```

Questa funzione non può essere utilizzata in modo diretto.

- **Exor:** ritorna l'oreclusivo fra due valori. La sintassi è: exor (x, y).

- **Joy:** ritorna lo stato di un joystick. La sintassi è: joy (n) dove n è la porta da leggere (1-2). I valori che la funzione a cui può ritornare sono indicati nella **tavola 1**.

- **\$:** permette di specificare un numero in notazione esadecimale. Ecco un esempio:

```
sys $aeafd
poke $002e,0
```

- **Segno di percentuale (%):** permette di specificare un numero in notazione binaria. Ecco qualche esempio:

```
poke 1024, %10101010
poke $0015, %11001101
b=%000011101
```

Nella **tavola 2** sono riportati i nuovi messaggi d'errore che si aggiungono a quelli dell'interprete standard. Nella **tavola 3** viene, invece, riportata la mappa della memoria quando l'espansione è attiva.

Daniele Maggio



*In questa puntata studiamo nei particolari il programma Finestra Hardware visto sui due numeri precedenti analizzandone, routine per routine, il cuore in linguaggio macchina*

# Grafica con l'Assembler

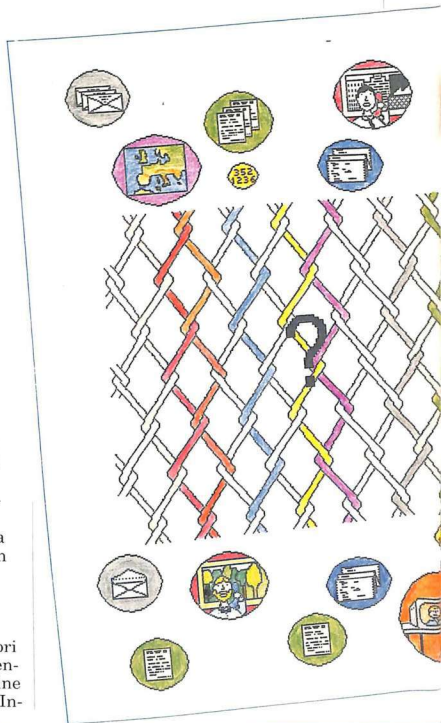
Nelle ultime due puntate (numeri 8 e 9 di *R.E.&C* 1989), abbiamo visto come funziona il programma. Sul numero scorso abbiamo pubblicato anche cinque interessanti programmi dimostrativi per capire come utilizzare, da Basic, le preziose routine di questa piccola espansione. In questa puntata forniremo le informazioni necessarie per utilizzare le stesse routine in Assembler. Anche il programmatore Assembler alle prime armi riuscirà a gestire il pacchetto di routine dando ai propri programmi un tocco di professionalità. Per chi volesse rinfrescarsi la memoria su ciò che abbiamo visto sul numero scorso, abbiamo preparato un listato Basic che permette di aprire una finestra e disegnarci dentro (**listato 1**).

Commento al ciclo principale (RASTER allocato a \$C1E9).

Cominciamo, dunque, l'analisi della routine di gestione degli interrupt, in pratica il cuore del programma.

```
AGA3R:      jmp $FEBC
```

È una routine che ripristina i valori dei registri interni del 6510, precedentemente salvati nello stack. La routine termina con un Rti, (Return From In-





errupt). La routine Raster salta a \$FEBC ogni 21 righe schermo.

**RASTER:** lda \$D019  
sta \$D019  
and #\$01  
beq AGA3R

Resetta la locazione \$D019, altrimenti non possono avvenire più interrupt. Se il bit uno è a zero, significa che non è avvenuto un interrupt raster e, allora, esce dalla routine saltando a \$FEBC.

Si osservi che, per azzerare il bit interessato, bisogna porre a uno lo stesso bit. Questo fatto che può sembrare una stranezza è, invece, una comodità.

ldx TEMP1  
lda TABY,x  
sta TM3  
sta \$D001  
sta \$D003  
sta \$D005  
sta \$D007  
sta \$D009  
sta \$D00B

Il registro X viene caricato con il valore della locazione Temp1 (\$CD39). Questa locazione contiene il numero di riga di sprite. La prima volta che avviene un interrupt questa locazione contiene zero, ciò sta a indicare che si deve disegnare la prima riga di sprite.

A Taby (\$CD7E) comincia una tabella che contiene l'ordinata delle varie righe di sprite. La locazione Taby+0 contiene, cioè, l'ordinata di inizio della finestra. Normalmente nella Taby sono contenuti valori che differiscono l'uno dall'altro di \$15 unità, valore pari al numero di linee di uno sprite (se si usano sprite espansi i valori

di Taby differiscono di \$2A unità). La tabella in questione è generata dalla routine Sety, che somma un certo parametro (Flagexpy) alla posizione Y. Il valore dell'ordinata viene salvato nella locazione Tm3 (\$CD3C), che sarà usata in seguito. Inoltre, viene settata la nuova ordinata degli sprite.

E da notare che il pannello elettronico ha, per il momento, un'ordinata inferiore al valore della locazione Tm3. In pratica il Vic II non si è ancora accorto della modifica e se ne accorgerà solo quando la posizione del pannello coinciderà con il valore scritto in \$D001-\$D003 eccetera.

lda TABBL,x  
tay  
cpx #\$00  
beq BASE

Viene prelevato dalla Tabbl (\$CD8B) il valore del blocco del primo sprite per la riga in questione. Il valore viene temporaneamente posto nel registro Y del 6510.

Il valore del blocco degli altri sprite di quella riga, viene ottenuto sommando uno al registro Y.

Viene poi controllato il valore del registro X. Ricordiamo che X contiene il numero di riga di sprite. Se è la prima riga della finestra non è necessario aspettare il sincronismo con il pannello elettronico e, quindi, salta direttamente a Base (\$C21E), dove vengono pokati i valori dei blocchi.

VG1: lda TM3  
AW02: cmp \$D012  
bne  
AW02

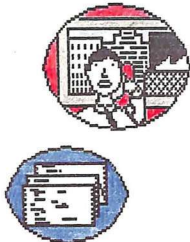
Aspetta che la posizione del pannello elettronico coincida con l'ordinata

## Tavola 1.

### Locazioni e tabelle

TABBASIC	SCD19	
TEMPSHB	SCD1E	
TMOPY	SCD1F	
TNSX	SCD20	
TNMY	SCD21	
TEMPXL	SCD22	
TEMPXH	SCD23	
TEMPY	SCD24	
FLAGCOL	SCD25	
FLAGPRI	SCD26	
FLAGXPY	SCD27	
FLASEXPX	SCD28	
COLORE	SCD29	
POKXL	SCD2A	
POKXH	SCD2B	
POGY		SCD2C
BLOCCO	SCD2D	
NGMY	SCD2E	
NGMAX	SCD2F	
MEMCHAR	SCD30	
MEMSCR	SCD31	
BANKOV	SCD32	
FLAGDOT	SCD33	
DOTY	SCD34	
DOTX	SCD35	
FLAGN	SCD36	
FLAGINT	SCD37	
MEMIO	SCD38	
TEMP1	SCD39	
SINCRAS1	SCD3A	
SINCRAS2	SCD3B	
TM5		SCD3C
TM4		SCD3D
TM3		SCD3E
TM6		SCD3F
TMT		SCD40
TMB		SCD41
SCRWIN	SCD42	
YSC		SCD43
XSC		SCD44
TY2		SCD45
TM2		SCD46
TY1		SCD47
TX1		SCD48
TIME		SCD49
TINC1	SCD4A	
FLAGXY	SCD4B	
TRES1	SCD4C	
TRES2	SCD4D	
TML1X	SCD4E	





della riga di sprite. Questo fatto è di fondamentale importanza perché, se i puntatori alle animazioni vengono modificati prima del previsto, succede quanto segue: il Vic II non ha ancora azzerato il suo contatore interno a sei bit, perché sta ancora leggendo i dati della riga precedente. Viene modificato il puntatore alle animazioni, così il Vic II va a leggere in una locazione che non è quella richiesta. Infatti, la locazione da dove il Vic II legge i dati degli sprite è così formata: 14 bit, di cui i bit da zero a cinque sono il contatore interno, mentre i bit da sei a 13 sono il valore del blocco. Facciamo un esempio.

Il blocco per la riga precedente vale uno (%01), mentre il blocco per la riga attuale vale due (%10). Supponiamo allora che il contatore interno sia arrivato a contare fino a \$3F (%011111), il Vic II sta allora leggendo i dati dalla locazione \$01\*\$40+\$3F=\$7F (%01 011111).

Se "blocco" cambia e da uno diventa due, si ha la seguente situazione

\$02\*\$40+\$3F=\$BF (%10 011111).

Cioè la prima riga del secondo sprite diventa l'ultima. Ovviamente è il caos e il disegno risulterà confuso. Questo fatto si verifica ugualmente se la finestra si trova nella posizione  $y=nx8+6$ .

BASE: sty \$07F8

- iny
- sty \$07F9
- iny
- sty \$07FA
- iny
- sty \$07FB
- iny
- sty \$07FC
- iny
- sty \$07FD

Il registro Y che contiene il blocco del primo sprite viene salvato in \$07F8.

Se viene spostata la memoria video il programma si occupa di modificare la locazione che contiene i puntatori alle animazioni. I puntatori alle animazioni occupano gli otto byte seguenti alle mille locazioni della memoria video. Una riga di sprite possiede una memoria contigua e questo è necessario per ragioni di velocità nella scrittura dei blocchi.

Sempre per motivi di velocità vengono scritti i blocchi di sei sprite, anche se la finestra in quel momento ne sta usando solo tre. In questo modo si risparmia tempo perché non sono necessari cicli di controllo. Ovviamente questo è un difetto, perché rimangono liberi un numero minore di sprite per altri scopi (sprite 6-7). Fortunatamente, si possono modificare i registri di uno sprite senza che questo compaia sullo schermo.

JMP ADDPAR

Terminata la scrittura dei puntatori alle animazioni il programma salta alla routine Ad-dpar (\$CDDC), la quale contiene un Jmp Carryras (\$C238), in pratica ritorna alla locazione seguente il Jmp Addpar. Questo Jmp serve se si vuole inserire una propria routine tra un inter-

Listato 1

Esempio di programma per aprire una finestra e disegnarci qualcosa

```

10 WI=49152 ; poiché tutte le routine hanno come
punto di chiamata la locazione 49152 ;
20 SYS WI,2:88,56 ; posizione x= 88: posizione y=56
30 SYS WI,0:7,0,0,0,0 ; colore giallo, nessuna espansione,
Sopra schermo, high resolution ;
40 SYS WI,1:5,200 ; window larga 6*24=144 pixel, alta
6*21=126 pixel, e posta alla
locazione 200*64=12800 ;
50 SYS WI,4:85 ; riempie con un retino
60 SYS WI,5:1 ; disegna cornice
70 SYS WI,8:0,0,144,126,0 ; cancella la diagonale della
finestra
80 SYS WI,9:144,0,0,126,0 ; come sopra

```



rupt raster e l'altro. Infatti, basta scrivere il programma voluto a partire dalla locazione (\$CDDC), avendo l'accortezza di non modificare il registro X del 6510 e di far terminare il programma con un `Jmp Carryras`:

```
CARRYRAS: ldy TABRAST +$01,x
           cpy   #$FE
           bcs   NM1
           cpx   NMAXY
           bcs   ESCI1
```

Il registro X contiene ancora il valore della locazione `Temp1`. Viene letto un dato dalla locazione `Tabrast` (\$CD98) +X+1. `Tabrast` indica la locazione di inizio della tabella più importante del programma. Infatti, essa contiene i valori di riga schermo a cui far avvenire i prossimi interrupt raster.

Il programma genera la tabella con la routine `Sety` e, se la prima riga di sprite si trova fuori quadro, cioè oltre la posizione \$FA (250), mette in `Tabrast` il valore \$FE. In questo modo, il programma capisce che da quella riga di sprite in poi, non è più necessario visualizzare gli sprite poiché questi sono totalmente fuori dallo schermo.

Se si verifica questo fatto salta alla locazione `Nm1` (\$C26E). Inoltre, controlla che la riga di sprite, di cui ha appena settato la posizione e i puntatori alle animazioni, non sia l'ultima riga della finestra. Se è proprio l'ultima riga salta alla routine `Esci1` (\$C253). Se, invece, la finestra ha altri sprite in lunghezza avviene che:

```
           inc   TEMP1
AGAIN2:   ldx   SINRAST1
AGA2R:   dex
         bne   AGA2R
         sty   $D012
         jmp   $FEBC
```

Prima di terminare la routine di interrupt è necessario eseguire altri compiti. In primo luogo va incrementata la locazione `Temp1`, cioè bisogna aumentare di uno il numero di riga di sprite. Deve, poi, essere settata la linea di qua-

## Tavola 2.

### Tabelle usate dal programma

TABHIGH	\$CD4F
TABML	\$CD56
TABR	\$CD5A
TABC1	\$CD5D
TABXCH	\$CD65
TABXH	\$CD77
TBY	\$CD7E
TABBL	\$CD8B
TABRAST	\$CD98
TABS1	\$CDA6
TABYC	\$CDB4
TABJUMP	\$CDB8
ADDPAR	\$CDDC
CARRYRAS	\$C238

dro per il prossimo interrupt, cosa che viene fatta salvando il registro Y in \$D012. Vi è poi un ciclo di ritardo, inserito per migliorare il sincronismo. La routine termina con il già descritto

### JMP \$FEBC

La richiesta di interrupt è stata soddisfatta e il 6510 riprende il compito che stava eseguendo prima che il pennello elettronico raggiungeva una delle posizioni contenute nella `Tabrast`. Se, invece, è appena stata disegnata l'ultima riga della finestra e la finestra è tutta contenuta nello schermo (registro X = `NMAXY`), succede che:

### ESCI1: clc

```
         lda   TM3
         adc   FLAGEXPY
         bcc   ETRT2
         lda   #$FF
         sta   $D012
         lda   high(LASTRA)
         sta   $0315
         lda   low(LASTRA)
         sta   $0314
         jmp   $FEBC
```

### ETRT2:

Viene preparato un interrupt raster alla riga `Tm3+$15` (se lo sprite non è espanso, altrimenti +\$2A). Se, invece, la posizione risultante è maggiore di \$FF, viene settata la riga \$FF.

Quest'ultimo interrupt viene indirizzato a una diversa routine, allocata a `Lastra` (\$C277).



## LINGUAGGIO MACCHINA

```
NM1: lda   #$FE
ET2:  cmp   $D012
      bne  ET2
      beq  ESCI
```

Viene chiamata se il prossimo sprite si trova fuori quadro. Si aspetta, cioè, che il pennello elettronico sia fuori quadro, dopo di che si fa terminare l'interrupt (ESCI §C289).

```
jsr   SETSP
jsr   SETX
jsr   SETY
jmp   ESC
lda   TMOVXY
```

La locazione Tempspb (§CD1E) viene posta a uno dalla routine Zero, cioè quando vengono modificati i parametri degli sprite che formano la finestra (colore, priorità e multicolor routine Setsp §C16B, espansione oppure distanza tra gli sprite Setx e Sety). In questo modo i parametri vengono settati quando il pennello è fuori quadro, diminuendo i tremolii. La routine Esc §C2CD è l'ultima parte della routine di interrupt (vedi dopo).

### Listato 2

*Esempio di gestione delle routine del programma in linguaggio macchina. Il programmino è l'equivalente del programma in Basic che traccia due diagonali di una finestra*

```
lda $000          sta POSY
sta FLAGDOL      lda $01
sta FLAGPR3     sta TMOVXY
lda $318
sta FLAGEXFX    lda $555
lda $315        sta $02
sta FLAGEXBY   lda $500
lda $507        sta TRES1
sta CCLORE     jsr CLEAR
jsr SETSP
jsr SETSP       lda $501
                sta FLAGDOT
lda $564        jsr FRAME
sta bloccb
lda $504
sta NSAVX
lda $505
sta NSAVY
jsr NK
jsr NK
lda $500
sta POSXH
lda $500
sta POSXL
lda $560
```

```
KKBS1:  lda   TMOVXY
        bne  ESC
        lda  #$00
        sta  TMOVXY
        lda  POSY
        cmp  TEMPY
        beq  KK1
        jsr  SETY
        lda  POSXH
        cmp  TEMPXH
        beq  KK2
        jsr  SETX
        jmp  ESC
KK2:    jmp  TEMPXL
        lda  POSXL
        cmp  ESC
        beq  ESC
        jsr  SETX$
        idx  FLAGINT
```

```
LAstra:  lda   $D019
        sta  $d019
        lda  TINT1+$01
        sta  $0314
        lda  TINT2+$01
        sta  $0315
```

La routine Lastra termina la gestione della finestra ogni quadro video. Viene ripristinato il vecchio vettore di interrupt, in modo che punti a Raster.

```
ESCI:   lda   TEMPSPB
        beq  KKBS1
        lda  #$00
        sta  TEMPSPB
```

Come per la locazione Tempspb la Tmovxy §CD1F, viene posta a uno ogni volta che è chiamata la routine Due, cioè ogni volta che la posizione cambia. Vi sono due locazioni che contengono la posizione x, una per il byte basso (PO-SXL §CD2A) e una per il byte alto (PO-SXH §CD2B). La posizione y è, invece, contenuta in un solo byte Posy §CD2C.

Ogni volta che sono chiamate le routine Setx o Sety, i valori delle posizioni sono ricopiati in altre locazioni (vedi Tempy, Tempxl, Tempxh).

In questo modo il programma, confrontando il valore della posizione attuale con il valore delle posizioni temporanee (Tempy eccetera), può sapere se



la posizione della finestra è cambiata. In caso affermativo provvede ad aggiornare la posizione, altrimenti termina la routine di interrupt saltando a Esc \$C2CD.

```
ESC:      ldx  FLAGINT
          lda  #$00
          sta  TEMP1
          sta  FLAGINT
          lda  TABRAST
          sta  $D012
          cpx  #$00
          beq  GOINT
          jmp  $FEBC
GOINT:    jmp  $EA31
```

Viene azzerato il numero di riga di sprite e la locazione Flagint \$CD37, che indica se durante il quadro sono state chiamate le routine di movimento (Setx e Sety). Se è avvenuto qualche movimento, non chiama la normale routine di interrupt allocata a \$EA31 perché, altrimenti, andrebbe perso troppo tempo. Esce dall'interrupt con il noto Jump \$FEBC. Se, invece, la finestra è rimasta nella posizione in cui si trovava il quadro precedente, salta a \$EA31 e termina l'interrupt.

La routine \$EA31 si occupa di scandire la tastiera, di aggiornare l'orologio interno e di muovere il cursore.

Prima di terminare, cioè prima di controllare se sono avvenuti movimenti, viene preparato anche l'interrupt raster per il nuovo quadro. La linea a cui avverrà l'interrupt è, ovviamente, presa dalla locazione Tabrast. Le tavole 1, 2 e 3 riportano l'elenco delle routine fondamentali e alcune tabelle delle locazioni attraverso cui vengono passati i parametri.

Uso di alcune routine da linguaggio macchina (mettere a uno la locazione Flagon).

- **Attiva finestra:** blocco, larghezza e lunghezza nelle locazioni blocco, Nmaxx, Nmaxy e salto a (XX \$C003).

- **Chiude finestra:** salto a Close1 (\$C2E8).

- **Settare banco del Vic II:** banco,

### Tavola 3.

#### Indirizzi routine che interpretano i vari comandi

BSETSP	\$CB81
BSTART	\$CB83
BPOS	\$CB83
BCLEAR	\$CC15
BFRAME	\$CBF2
BSETCHAR	\$CC4D
BLOT	\$CC36
BINVERT	\$CC12
BLINEXY	\$CCD9
BCLOSE	\$CBFF
BWRAP	\$CC75
BMOVE	\$CCB6
BSAVE	\$CABD
BLOAD	\$CAF1

memoria generatore caratteri e memoria video nelle locazioni Bancov, Memchar, Memser e saltare a Setbanco (\$C312).

- **Modifica caratteristiche:** multi-color, priorità, espansione y e x, colore nelle locazioni Flagcol, Flagpri, Flagexy, Flagexpx, Colore e salto a Setsp (\$C16B).

- **Plotta un punto:** colore, ordinata, ascissa nelle locazione Flagdot Doty, Dotx e salto a Plotts (\$C49A).

- **Traccia:** linea dal valore contenuto in Ty1 Tx1 a Ty2 Tx2, colore in Flagdot: saltare a Stlinexy(\$C877).

- **Cornice:** colore in Flagdot e salto a Frame (\$C708).

- **Riempie con un valore:** valore in \$02, locazione \$CD4D a zero e salto a Clear (\$C39E).

- **Stampa carattere:** numero del carattere in (\$FD-\$FE), sovrapposizione o meno in (\$C583) ordinata e ascissa in Doty e Dotx, salto a Setchar (\$C534).

- **Cambia posizione:** posizione X in Posxl e Posxh posizione Y in Posy, contenuto della locazione Tmovxy diverso da zero (non occorre nessun salto, basta che la finestra sia attiva). Il **listato 2** è un esempio di quanto detto finora.

Nicola Chiminelli  
(continua)

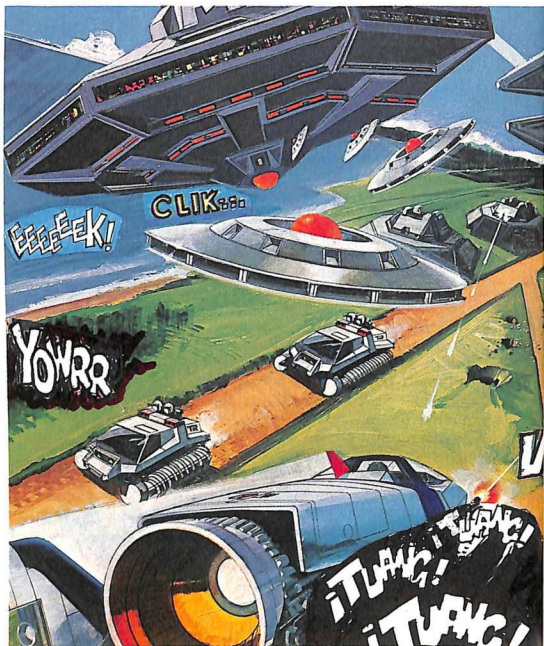


C64

SUONI

*Questa volta vi insegnamo come applicare qualsiasi tipo di effetto sonoro ai vostri programmi con l'aiuto dell'ottimo chip Sid del C64. Le istruzioni per l'uso sono autoguidate e anche i meno esperti potranno effettuare le operazioni necessarie senza problemi*

CLING!



## Il chip che fa crash!

Da molto tempo sono presenti sul mercato i più disparati programmi per la creazione di musiche ed effetti sonori personalizzati. Quello che si riesce ad ottenere da questi programmi è veramente notevole, si riescono a creare una moltitudine di effetti originalissimi che nei nostri program-

mi farebbero veramente una strepitosa figura. Il problema è anche l'unica grande lacuna di queste utility è che nella stragrande maggioranza dei casi permettono di creare effetti belli e complessi ma inutilizzabili separatamente o in un altro programma.

Ora, grazie al programma di questo mese e allo stupendo Chip Sid del Commodore 64, riusciremo a colmare questa lacuna. Chiunque potrà inserire nei propri programmi, in una qualunque zona della memoria, l'effetto sonoro desiderato.

Il programma si chiama Sid 2 ed è predisposto per lavorare congiuntamente al precedente programma Sid, ma questo non pregiudica assolutamente la sua funzionalità anche senza di esso.



### Caricamento

Dopo aver acceso il computer, riavvolto completamente la cassetta con i programmi e azzerato il contagiri del registratore, si digiti direttamente Load seguito dal tasto Return, in questo modo avrete la possibilità di rilevare tutti gli indirizzi dei vari programmi della cassetta semplicemente sottraendo quattro numeri al contagiri del registratore ogni volta che sul video appare la scritta Found seguita dal nome del programma. Terminata quest'operazione è possibile caricare separatamente ognuno dei programmi presenti.

Una volta che il computer avrà caricato Sid 2 si potrà digitare il consueto Run seguito dal tasto Return. Sullo schermo apparirà un messaggio lampeggiante dove si

avvertirà l'operatore che il programma è utilizzabile solo per ogni singolo effetto sonoro, una volta salvato il quale, è obbligatorio ripetere la procedura dall'inizio ricaricando il programma Sid 2. Questa particolarità è data dal fatto che voi potreste aver bisogno di salvare un effetto sonoro proprio nella zona di memoria in cui il programma convertitore sta lavorando.

### Funzionamento

Una volta apparsa la presentazione è sufficiente premere un tasto per passare nella fase preliminare dove compaiono alcune note e sarà presente anche la possibilità di scegliere una tra le tre stupende musiche di sottofondo contenute nel programma, semplicemente premendo i tasti da F1 a F3. Il tasto F7 per chi non desidera alcuna musica. Dopo quest'ultima operazione potrete finalmente passare al programma principale che vi apparirà sotto forma di una serie di domande alle quali dovrete rispondere in funzione dell'effetto sonoro desiderato e possibilmente già sperimentato con il programma Sid, comparso sul numero di settembre 89, o comunque con un qualsiasi programma creatore di effetti sonori. La parte di inserimento dei vostri valori è divisa in cinque fasi.

#### • Prima fase (voce 1)

Il computer vi richiederà nove valori che voi dovrete inserire nel seguente ordine:

- 1- Frequenza (0-65535) : frequenza della nota.
- 2- Pulsazione (0-100 per cento) : ampiezza dell'onda pulsante, da usare solo con la forma d'onda pulsante.



3- Forma d'onda (T=1,S=2,P=3,N=4): triangolare, dente di sega, pulsante e rumore.

4- Sincronismo: inserendo il numero zero o premendo il tasto Return si annulla la funzione, digitando invece il numero uno si sincronizzerà la voce 1 con la voce 3.

5- Modulazione ad anello: ha le stesse particolarità del sincronismo, ma se selezionata permette la modulazione ad anello della voce 1 con la voce 3.

6- Attacco (0-15) : Tempo che l'effetto impiega per raggiungere il volume massimo.

7- Caduta (0-15): Tempo di caduta dell'effetto sonoro al livello di sostegno.

8- Sostegno (0-100 per cento): percentuale del volume totale al quale l'effetto sonoro fa riferimento.

Questa opzione potrebbe risultare superflua al lettore inesperto, il quale si chiederebbe il motivo della presenza di un volume secondario quando è già presente un volume principale. La risposta sta nel fatto che il Sostegno è selezionabile singolarmente per ciascuna voce, da questo deriva il pregio di poter alzare e abbassare il volume di ogni singola voce mantenendo inalterato quello delle altre due voci.

9- Stacco (0-15) : è il tempo che impiega l'effetto sonoro ad azzerarsi.

Per avere un'idea sui tempi delle ultime quattro opzioni è possibile riferirsi alla tabella valori dell'involuppo contenuta nel programma scorso Sid.

#### •Seconda fase (voce 2)

Identica alla prima fase ma con i valori riferiti alla voce 2. Modulazione ad anello e sincronismo tra le voci 1 e 2.

#### •Terza fase (voce 3)

Sempre identica alla prima ma riferita alla voce 3. Modulazione ad anello e sincronismo tra le voci 2 e 3.

#### •Quarta fase

In quest'ultima fase di inserimento sono presenti le più peculiari caratte-

ristiche del programma, e precisamente:

1- Frequenza filtro : frequenza di taglio del filtro programmabile dall'operatore.  
2- Risonanza : risonanza dell'effetto sonoro in una gamma di valori da 0 a 15.

3- Filtro 1 : premere il numero zero o il tasto Return per non attivarlo, oppure premere il numero uno per filtrare l'effetto sonoro prodotto dalla voce 1. Questo parametro fa riferimento alla frequenza di taglio selezionata in precedenza, la quale fa a sua volta riferimento alla frequenza selezionata per l'effetto sonoro, sempre della voce 1.

4- Filtro 2 : ha le stesse caratteristiche del Filtro 1, ma si riferisce all'effetto prodotto dalla voce 2.

5- Filtro 3 : stesso discorso del Filtro 2, ma riferito alla voce 3.

6- Modo Filtro (1-2-3) : questa funzione deve essere necessariamente impostata per permettere ai parametri di filtro selezionati in precedenza di funzionare. Ogni volta che viene attivata, influenza contemporaneamente tutte le voci del Sid che sono state impostate per essere filtrate. Può essere indirizzata in tre differenti modi:

a) Tasto numero 1 (Passa Basso) : se impostato permette il passaggio solo delle frequenze più basse di quella di taglio selezionata.

b) Tasto numero 2 (Passa Banda) : ora le frequenze che potranno passare sono comprese attorno al valore selezionato.

c) Tasto numero 3 (Passa Alto) : in quest'ultimo caso invece potranno passare solo le frequenze superiori a quella di taglio inserita all'inizio.

7- V3 Osc. : questa è una di quelle opzioni per cui è necessario possedere il precedente programma Sid, ma in caso contrario conviene andare per tentativi inserendo uno dei tre valori possibili.

8- V3 Env. : anche quest'altro necessita dello stesso programma per essere sfruttato adeguatamente.

9- Interruttore Canale 3 : è consigliabile selezionare quest'opzione solo se in congiunzione con il sincronismo o la modulazione ad anello per evitare noiosi disturbi. Infatti, nella maggior parte dei casi, la



voci 3 è usata per influenzare altre voci, nel caso venga disattivata essa compie comunque la sua funzione, ma evita di disturbare il campione sonoro d'uscita.

10. Volume (0-15) : volume principale dell'intero sistema. È consigliabile selezionare valori tra il dieci e il 15 per una buona udibilità dell'effetto.

#### •Quinta fase

In questa parte il programma compie tutte quelle operazioni che porteranno alla formazione di un listato in codice macchina che genera l'effetto sonoro precedentemente selezionato. Prima di tutto ciò il computer chiede se tutti i dati che sono stati inseriti precedentemente sono corretti e vi farà sentire l'effetto sonoro impostato. Nel caso riscontrate qualche imprecisione, o il campione sonoro non è come ve lo aspettavate, allora alla domanda del computer "Tutto ok (s/n)?" dovrete rispondere negativamente premendo il tasto N.

Se invece l'effetto risponde alle vostre esigenze potrete premere il tasto S e il computer continuerà l'elaborazione dei dati chiedendovi l'indirizzo di memoria dal quale intendete salvare il codice macchina, e questa sarà l'ultima richiesta che il computer vi farà, ma anche la più importante perché determina la zona di memoria entro la quale si andrà a collocare il vostro sottoprogramma. Evitate di inserire il numero 49152 come indirizzo di partenza, perché in questo modo, andrete a salvare l'effetto esattamente sulla routine che lo gestisce causando un inevitabile blocco.

Una buona posizione potrebbe essere la 49500. Le successive schermate vi chiariranno le idee sull'utilizzo dell'effetto sonoro generato da Sid 2. Terminate queste schermate di aiuto potrete salvare l'effetto su registratore o drive.

#### Applicazioni

Se avete già fatto un programma e volete aggiungergli degli effetti sonori le operazioni da compiere sono le seguenti:

1) caricate il vostro programma e senza dare il Run digitate direttamente la seguente linea Basic:



PRINTPEEK(45)+PEEK(46)\*256

Sul video dovrebbe apparire un numero compreso tra 2049 e 40960. Scrivete il numero che apparirà su un foglio; questo valore vi darà un'idea della lunghezza del programma. Sapere dove termina la memoria riservata al programma è indispensabile per poter porre le routine in linguaggio macchina degli effetti sonori in una zona libera della memoria.

Il numero che avete sul video indica l'ultima locazione occupata dal programma appena caricato, ma non tiene conto dell'area che il programma occuperà con le variabili e le matrici. Se conoscete a fondo il programma saprete, grosso modo, quanto spazio lasciare libero.

In media dovete lasciare un migliaio di byte liberi dopo un programma senza matrici per essere sicuri. Se il programma fa uso di matrici la lunghezza reale del programma aumenterà notevolmente. Bisogna tener conto inoltre che per utilizzare gli effetti sonori generati da Sid 2 sono necessarie altre linee Basic da aggiungere al programma già scritto. La **tavola 1** elenca le istruzioni Basic da dare per attivare e disattivare l'effetto sonoro. La lettera B che compare nel testo rappresenta l'indirizzo di memoria dal quale avete in precedenza deciso di salvare il vostro sottoprogramma.

La locazione più opportuna per la disposizione in memoria dell'effetto sonoro (o delle catene di effetti sonori) è il più vicino possibile alla fine del programma Basic che essa accessoria. Purtroppo l'ultima locazione del programma, come spiegato sopra, non è facile da individuare per un pro-





grammatore non molto esperto, che deve andare un po' per tentativi. Disporre un effetto sonoro troppo lontano dal programma principale significa aumentare inutilmente le dimensioni della versione finale e completa del vostro programma. Infatti il programma definitivo occuperà la memoria da 2049 alla fine della routine dell'effetto sonoro, pur lasciando inutilizzata la memoria intermedia.

Se desiderate avere disponibili molti effetti sonori nello stesso programma esso sarà ovviamente allungato dalle stesse linee Basic per attivare ognuno di essi.

Un appunto importante è di ricordarsi, ogni volta che si richiede di scrivere il numero su un foglio di carta, il punto in cui il computer lo ha richiesto per evitare una successiva confusione. Per esempio al punto numero 1 si richiede di scrivere un valore, accanto al quale conviene anche segnarsi il numero 1, per ricordarsi che il numero è stato scritto a quel punto.

2) Spegnete e accendete il computer e caricate Sid 2, digitate Run seguito dal tasto Return.

3) Ora create un qualsiasi effetto sonoro rispondendo alle domande che vi farà il computer. Quando vi verrà chiesto a che punto della memoria intendete salvare l'effetto, digitate il numero che ritenete più idoneo (per andare sul sicuro potete di-

gitare 36864, ma avrete un programma finale lunghissimo) e scrivete anche questo sul foglio. Questo numero rappresenta la base di richiamo dell'effetto sonoro che in **tavola 1** trovate indicata con la lettera B. Fatto questo seguite le istruzioni sullo schermo e appena appare la scritta Ready, salvate l'effetto su disco o nastro con un comune Save "nome", 1(8), 1.

È importante ricordarsi di inserire anche il secondo "1" perché in caso contrario il computer salverà l'effetto a partire dalla locazione di Default 2049, ignorando l'indirizzo di partenza che avevate imparato. Ora che avete tutti e due i programmi salvati su registratore o drive, passiamo alla parte conclusiva del lavoro: unirli in un unico file.

4) Senza spegnere il computer caricate il programma principale fatto da voi e aggiungete, al termine del caricamento, le linee Basic necessarie al richiamo delle routine sonore (vedi punto 1).

5) Salvate per precauzione il programma modificato, e senza ovviamente spegnere il computer, caricate il programma che ha generato Sid 2.

6) Verificate pure, dando il Run al vostro programma, il corretto funzionamento delle routine sonore e se qualcosa non funziona provate a controllare l'esattezza

## Tavola 1.

Elenco delle istruzioni Basic da dare per attivare e disattivare l'effetto sonoro. B è l'indirizzo specificato durante la creazione dell'effetto

### • Per attivare, digitate:

POKE B+80,PEEK (B+80)OR 1 per impostare la voce 1.  
POKE B+115,PEEK (B+115)OR1 per impostare la voce 2.  
POKE B+150,PEEK (B+150)OR1 per impostare la voce 3.  
SYS B+49 da inserire sempre.

### • Per disattivare:

POKE B+80,PEEK (B+80)AND 254 per fare passare allo stacco la voce 1.  
POKE B+115,PEEK (B+115)AND 254 per fare passare allo stacco la voce 2.  
POKE B+150,PEEK (B+150)AND 254 per fare passare allo stacco la voce 3.  
SYS B+49 da inserire sempre.



delle linee di richiamo inserite. Prima di dare nuovamente il Run ricordate di caricare nuovamente le routine sonore generate da Sid 2: manipolando il programma Basic, a volte, viene alterata tutta la memoria e, nel nostro caso, le nostre routine sonore. Se anche ora qualche cosa andasse storto, vorrà dire che gli effetti sonori sono stati salvati troppo vicino al programma e si è verificata una sovrapposizione di dati, perciò dovrete ripetere l'operazione dal punto 2.

7) Se tutto funziona correttamente potrete salvare l'effetto su periferica digitando il consueto Save "nome",8(1),1 e premo il tasto Return.

Naturalmente ogni volta che il programma sarà caricato le routine sonore saranno già presenti al suo interno, e perciò direttamente utilizzabili.

### Particolarità

Vedremo ora come inserire più effetti sonori nello stesso programma.

La procedura da seguire è piuttosto semplice, in quanto, essendo l'effetto sonoro composto da un codice macchina non vi è necessità di inserire spazi tra l'uno e l'altro, in quanto il linguaggio macchina non usa variabili in modo diretto e perciò non necessita di spazio dopo un eventuale programma.

Da questo fatto emerge la possibilità di concatenare uno svariato numero di effetti sonori semplicemente salvandoli uno dietro l'altro alla distanza di sicurezza di 200 byte, senza alcun pericolo di incorrere in sovrapposizioni; vediamo un esempio diretto: se abbiamo salvato il primo effetto alla locazione 4000, il secondo andrà a sua volta salvato alla locazione 4200, il terzo alla locazione 4400 e così via... Alla fine vi troverete con molti programmini su periferica, ciascuno dei quali occuperà meno di 200 byte, e per averli tutti disponibili nello stesso programma basta seguire la procedura di prima con la differenza che al punto 5 il programma generato da Sid 2 da caricare non è uno, ma bensì tutti quegli effetti che sono distanti tra loro 200 byte, avendo l'accortezza di caricare per ultimo l'effetto che usa le locazioni più alte, in quanto, tutto ciò che c'è in memoria oltre

l'ultimo programma caricato, viene automaticamente perso quando noi impartiamo il Save. Se noi infatti carichiamo un effetto a partire dalla locazione 4400 e successivamente un altro che parte dalla locazione 4200, tutto ciò che si trova oltre questo ultimo numero viene perso e perciò anche l'effetto caricato dalla locazione 4400, ovviamente tutto questo avviene quando impartiamo il Save.

L'uso ottimale del programma Sid 2 consiste nell'inserire dai cinque ai 20 effetti sonori in modo da beneficiare dell'effettivo risparmio di memoria che ne deriva.

Tutto ciò che riguarda l'inserimento degli effetti sonori dovrebbe ora essere stato chiarito, ma rimane ancora una nota di merito a favore di questa utility: infatti, una volta che l'effetto sonoro è stato abilitato dal vostro programma, questo continuerà a sentirsi finché non lo disabilitate e, inoltre, grazie all'interrupt che viene impostato da ogni singolo effetto sonoro, il programma continua ad andare avanti indipendentemente, come nei programmi più sofisticati!

Nella speranza di avervi chiarito tutte le fasi della creazione del vostro programma sonorizzato vi suggeriamo di scatenare senza ritengo la vostra fantasia, perché la parte più interessante di tutta l'operazione, cioè il programma che dovrà gestire gli effetti sonori e gli effetti sonori stessi, dipende completamente da voi.

Coloro che invece di utilizzare il drive usano ancora il registratore saranno abbastanza penalizzati dalla scarsa agilità delle operazioni. Un ottimo accorgimento per questi utenti è quello di scrivere su carta ogni più piccola informazione.

**Franco Valter**

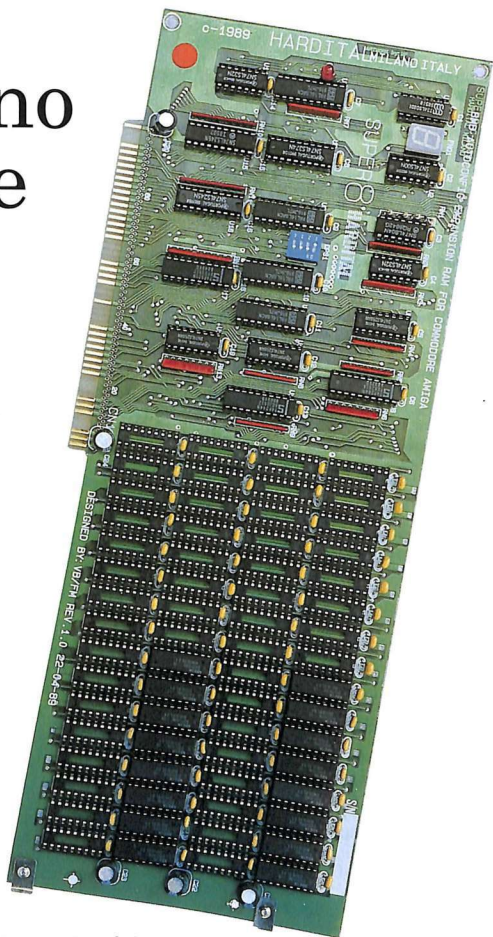
CLING!



AMIGA È...

# L'uragano sul mare di Ram

*Questo mese la  
nostra attenzione è  
rivolta a due  
formidabili schede  
per Amiga 2000:  
l'espansione di  
memoria Hardital  
SuperOtto e la  
scheda Hurricane  
dotata del potente  
microprocessore  
68020 con  
coprocessore  
matematico 68881*



Questa volta dedichiamo la rubrica a due accessori che rendono l'Amiga uno strumento decisamente professionale e allo stesso tempo molto economico.

#### **Hardital SuperOtto**

Questa scheda è una raffinata espansione di memoria, utilizzabile da 2 a 8 Mb.

La prima occhiata ha messo in risalto la precisa realizzazione e l'ordine con cui è stata disegnata la scheda, che è fittissima di componenti. Ogni Mb di Ram è realizzato con otto chip da un Mbit e, poiché tutti i chip Ram sono montati su uno zoccolino,



la maggior parte della scheda è occupata dai 64 zoccolini per le Ram. La parte restante della scheda raccoglie alcuni chip di controllo, un display a Led, quattro dip-switch e un Led di accensione.

Il montaggio della scheda è facilissimo: si toglie il "guscio" di Amiga 2000, si inserisce la scheda in uno qualsiasi degli slot di espansione, si richiude. Non potrebbe essere più facile.

Se si acquista la scheda priva dei chip Ram, oppure si desidera aggiungere altra Ram alla scheda, prima di procedere con l'installazione è necessario provvedere all'inserimento di otto chip Ram per ogni Mb di memoria. È possibile utilizzare SuperOtto Hardital con 2, 4, 6 oppure 8 Mb. In qualsiasi caso si deve procedere attentamente all'inserimento di tutti i chip negli zoccoli prestando attenzione, soprattutto, a non invertire nessun piedino e a rispettare la direzione di inserimento. A questo punto si deve configurare la scheda agendo sui quattro dip-switch secondo una tabella contenuta nelle istruzioni di montaggio. In qualunque caso, il display a Led montato sulla scheda mostra la quantità di memoria inserita sulla scheda. Non ci sembra un particolare fondamentale poiché, per sapere quanta memoria è montata, basta contare i banchi di otto chip presenti: si tratta più che altro di una decorazione.

Terminato il rapidissimo montaggio della scheda abbiamo acceso Amiga e abbiamo ammirato sulla barra menù dello schermo di Workbench la scritta "3009315 bytes free" che ci ha confermato l'avvenuta configurazione dell'espansione di memoria.

Abbiamo provato la SuperOtto Hardital con numerosi programmi e non abbiamo osservato alcuna incompatibilità. Abbiamo anche provato ad avviare numerosi programmi contemporaneamente, mettendo a dura prova la capacità di Amiga di gestire il multitasking ma, nonostante il video fosse saturo di schermi e finestre e l'esecuzione dei vari task fosse ormai molto rallentata, Amiga non ha avuto alcuna difficoltà a proseguire il proprio lavoro.

Accertato, dunque, che la SuperOtto Hardital è un'ottima scheda di espansione memoria, si pone una domanda: a chi serve un'espansione simile, considerando che

costa più di un'Amiga 500?

Non pensiamo che l'utilizzo principale possa essere un uso più spinto del multitasking, perché la velocità di Amiga risente moltissimo dell'apertura di numerosi task contemporaneamente. Piuttosto, una simile espansione di memoria può essere utilizzata molto efficacemente come Ram disk accelerando in modo molto significativo il funzionamento di diversi programmi. In particolare, l'uso di compilatori è agevolato in modo davvero notevole dall'uso della Ram disk, infatti, la lentezza dei compilatori e dei linker è da addebitarsi in grandissima parte al tempo richiesto per il caricamento in memoria e il salvataggio dei vari file.

Se si dispone della versione 1.3 del sistema operativo è possibile utilizzare invece della Ram disk il device "CARD:" che è sostanzialmente una Ram disk, ma presenta la pregevole caratteristica di non cancellarsi in caso di reset del sistema. Un'espansione di memoria è essenziale anche per chi pensa di utilizzare in modo serio programmi complessi e potenti che sono soffocati in un solo Mb di memoria come, per esempio, Sculpt 4D. Certamente la spesa è considerevole, tuttavia, Amiga 2000 con 3 Mb di memoria diventa uno strumento davvero potente.

## Hurricane

La Ronin Research and Development produce già da tempo un'efficiente espansione per Amiga 1000 e 2000 dal nome forse troppo altisonante: Hurricane (uragano). Il pacchetto consiste in una scheda, un disco con del software e un manuale di installazione e uso.

La scheda Hurricane è relativamente piccola. Consiste in un circuito stampato da inserire in uno slot Amiga sul quale sono montati l'imponente Motorola 68020, il suo assistente 68881, il clock per quest'ultimo, che funziona asincrono rispetto al microprocessore, un poco di circuiteria di controllo e un fitto connettore per l'espansione di memoria a 32 bit. Il microprocessore 68020 è, in un certo senso, il fratello maggiore del 68000 che equipaggia Amiga standard. Una differenza fondamentale tra i due microprocessori consiste nel fatto che il 68000, pur aven-



## AMIGA E...

do registri interni a 32 bit, ha il bus dati a 16 bit, ovvero comunica con il mondo esterno come un 16 bit, mentre il 68020 è un vero 32 bit, ovvero lavora e trasferisce dati sempre tramite parole" di 32 bit. Questo fatto, che rappresenta il punto di forza del 68020, è anche il punto debole della scheda Hurricane: poiché Amiga è stato progettato intorno al 68000, il suo bus dati è a 16 bit, quindi, quando Hurricane deve accedere alla memoria o ai chip custom deve convertire i dati da 32 a 16 bit, rinunciando al vantaggio di cui dispone. Il 68020 montato sulla Hurricane, inoltre, è stato progettato per funzionare a 14,32 MHz, ovvero esattamente il doppio del clock del 68000 di Amiga. Poiché il bus dati di Amiga lavora a 7,16 MHz, ogni volta che il 68020 deve accedere alla Ram o ai chip custom è costretto a dimezzare la propria velocità. È possibile rimediare a entrambi questi problemi in modo molto efficace e semplice, ma anche molto costoso, acquistando l'espansione di memoria Hurricane a 32 bit. Tale espansione si collega direttamente alla scheda Hurricane grazie all'apposito connettore e fornisce al 68020 una Ram a 32 bit con accesso a 14,32 MHz. Solo in questo modo il 68020 può lavorare alla massima velocità e può essere sfruttato completamente.

La maggiore velocità di clock e il bus più esteso non sono le sole caratteristiche che fanno del 68020 un microprocessore più potente del 68000. Il 68020 ha anche un set di istruzioni più esteso rispetto al fratello. Per esempio, è in grado di eseguire moltiplicazioni, divisioni e calcolo di modulo a 32 bit con un'unica istruzione al posto delle 20 o 30 istruzioni richieste dal 68000, il che consente un notevole risparmio di tempo. Naturalmente per poter sfruttare questa caratteristica del microprocessore è necessario che il software sia scritto appositamente per il 68020.

Nel floppy disk di Hurricane, tuttavia, si trova un programmino chiamato PatchMath020 che provvede a modificare i programmi creati per 68000 sostituendo, appunto, le numerose istruzioni di cui quest'ultimo necessita per effettuare una moltiplicazione con l'istruzione corrispondente del 68020, seguita da un Rts (ritorno da subroutine). È evidente che un programma modificato con PatchMath020

non può più funzionare con il 68000.

Un'altra caratteristica fondamentale del 68020 è la memoria cache, che è motivo sia di prestazioni eccezionali con alcuni programmi, sia di noiosissime Guru Meditation con altri programmi. La memoria cache consiste nella possibilità del microprocessore di ricordare 64 istruzioni, il che gli consente di effettuare cicli di istruzioni senza dover ricorrere continuamente alla Ram. In questo modo molti programmi possono girare più velocemente grazie al ridotto accesso alla Ram, tuttavia, alcuni programmi scritti per 68000 hanno delle istruzioni auto-modificanti, generalmente introdotte per protezione anti-copia. Chiaramente questo tipo di programmazione è causa di sicuro crash se utilizzata con memoria cache, infatti, il programma modifica il codice su Ram, ma il microprocessore segue le istruzioni che ormai si trovano in cache. Per rimediare a questi problemi sul floppy disk si trovano i due programmi CacheOn e CacheOff che rispettivamente abilitano e disabilitano la memoria cache. In teoria, inserendo nella startup-sequence dei programmi incriminati il comando CacheOff si risolverebbe il problema, tuttavia, molti giochi partono in autostart pertanto, l'operazione non è possibile e il crash del sistema è sicuro.

Passiamo al primo ministro di Sua Maestà il 68020. Si tratta naturalmente del coprocessore matematico che, però, non è venduto insieme alla scheda, ma deve essere acquistato separatamente. È possibile installare sia il 68881, sia il più potente 68882 e poiché il coprocessore funziona asincrono rispetto al 68020 è possibile scegliere il clock desiderato tra i diversi disponibili: 16, 20 oppure 24 MHz, avendo la sola accortezza di inserire il cristallo di quarzo che fornisce il clock desiderato nell'apposito zoccolino.

A cosa serve un coprocessore matematico? Può migliorare le prestazioni del vostro computer in modo drastico, così come può essere quasi inutile, a seconda dei programmi che pensate di utilizzare. Un coprocessore matematico è in grado di eseguire con un'unica istruzione il calcolo in virgola mobile delle quattro operazioni e delle funzioni trascendenti, come le funzioni trigonometriche, l'esponenziale o il

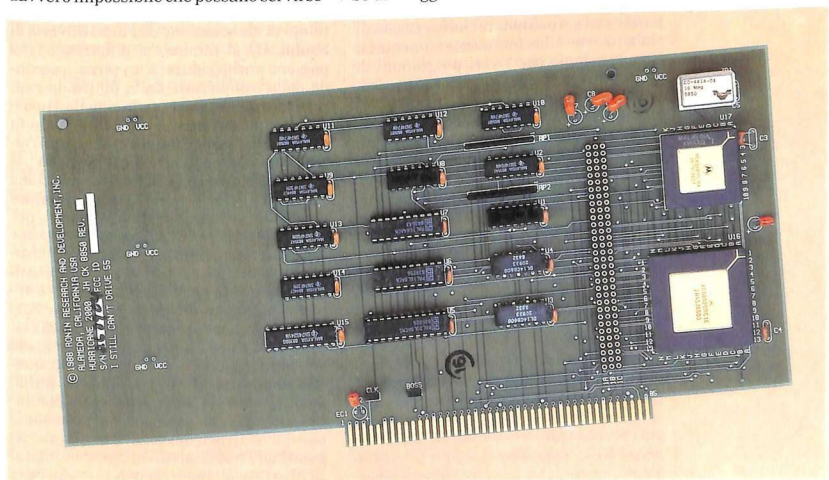


logaritmo. Questo significa che un programma che fa largo uso di queste funzioni può girare molto più velocemente se viene assistito da un coprocessore matematico: addirittura può impiegare un decimo del tempo richiesto dal solo microprocessore. Purtroppo, la situazione non è così semplice poiché, per poter fruttare un coprocessore matematico, un programma deve essere scritto appositamente, altrimenti non si sogna neppure di utilizzarlo. Le routine matematiche contenute nelle librerie di sistema di Workbench 1.3 sono in grado di accorgersi se è installato un coprocessore matematico e quindi di utilizzarlo, pertanto i programmi che fanno uso di tali librerie possono sfruttare in parte il 68881 (oppure il 68882). Per avere il meglio dal nostro 68881, però, è necessario diporre di un programma creato in modo specifico per essere utilizzato con un coprocessore matematico. Non ci sono in commercio numerosi programmi che prevedono l'uso del coprocessore matematico, anche perché le schede con il 68020 e il 68881 sono costose e poco diffuse. Molti programmi, inoltre, non hanno nessun bisogno di un coprocessore matematico, ed è davvero impossibile che possano servirse-

ne. Si tratta naturalmente dei word processor, dei database e di altri programmi a "basso contenuto di matematica". I programmi che invece possono davvero beneficiare della scheda Hurricane sono principalmente i programmi di ray-tracing, i Cad e i compilatori di linguaggi utilizzati per programmi di matematica, fisica e ingegneria.

In particolare i due più diffusi programmi di ray-tracing, Sculpt 4D e Turbo Silver, sono in grado di utilizzare il 68020 e il coprocessore matematico.

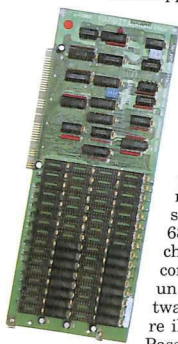
Dopo questa descrizione generale passiamo al montaggio della scheda. L'operazione è molto semplice, ma anche molto delicata. La scheda Hurricane è disponibile in due diverse versioni per Amiga 1000 e Amiga 2000. In entrambi i casi si deve aprire il computer e si deve estrarre (stavamo per dire esautorare) il buon vecchio 68000. Se si desidera tenere il 68000 (non ci pare proprio il caso di buttarlo via!) è necessario fare molta attenzione a sfilarlo dal proprio zoccolo senza stortare i piedini, inoltre, sia maneggiando il 68000, sia maneggiando la scheda bisogna evitare di indurre elettricità statica che potrebbe danneggiare i chip. Dopo aver estratto





## AMIGA È...

il 68000 si deve inserire la scheda, che nel caso di Amiga 1000 deve essere inserita direttamente nello zoccolino del microprocessore, mentre Amiga 2000 dispone di un apposito slot Cpu dove Hurricane



trova comodo alloggio. Se si acquista il 68881, prima di inserire la scheda nel computer è necessario inserire sia il coprocessore, sia il quarzo per il clock nei loro zoccolini, sempre con delicatezza e attenzione infinita. Il montaggio richiede pochi minuti: sarebbe stato meglio, comunque, se non fosse necessario disinserire il 68000, almeno per Amiga 2000 che dispone dello slot Cpu. Ancora meglio sarebbe disporre di un interruttore o un sistema software per attivare il 68000 oppure il 68020, a scelta dell'utente. Passiamo finalmente a vedere il 68020 in azione. Abbiamo provato

a utilizzare Amiga con Hurricane con alcuni programmi non scritti appositamente per il 68020. Amiga funziona regolarmente e non ci si accorge quasi del diverso microprocessore. Osservando più attentamente si può notare che la velocità di elaborazione è discretamente aumentata. Non pensiamo che i vari programmi di benchmark offerti per testare la Hurricane, dei quali parleremo in seguito, siano significativi più di tanto, pertanto ci azzardiamo a dire che le prestazioni di Amiga in termini di tempo aumentano dal 50 al 90 per cento. Da una parte il risultato è buono, tuttavia non ci sembra che questo aumento in prestazioni valga la spesa.

Abbiamo poi provato la Hurricane con A-Max, l'emulatore di Macintosh provato lo scorso numero di *Radio Elettronica & Computer*. Abbiamo fatto girare tutti i programmi più conosciuti per Macintosh, da Word 4 a PageMaker 1.2, a Excel senza incontrare alcun problema e osservando un aumento di prestazioni simile a quello osservato in AmigaDos. Siamo poi passati a QuickBasic, il programma Microsoft che fornisce un interprete Basic del tutto compatibile con AmigaBasic che permette di compilare i propri programmi. Sapevamo che tale programma dispo-

ne, in fase di compilazione, dell'opzione 68020 e 68881, pertanto abbiamo provato la compilazione che avrebbe dovuto sfruttare al meglio Hurricane. Il risultato è stato davvero eccezionale: non solo i programmi hanno girato perfettamente, ma il seguente programma ha girato in soli sette secondi:

```
b=1.42123
FOR i=1 TO 100000
a=SIN(b)
NEXT i
```

Inutile aggiungere che si tratta di un risultato di tutto rispetto: senza Hurricane quel programma avrebbe girato in diversi minuti. Passiamo ora ai programmi scritti appositamente per la magica coppia 68020-68881. Abbiamo provato i due più famosi programmi di ray-tracing, ovvero Sculpt-Animate 4D e Turbo Silver 3.0. Com'è noto questo tipo di programmi richiede tempi di elaborazione molto lunghi poiché, necessariamente, un'immagine ray-tracing richiede il calcolo del passaggio dei raggi luminosi, il che si traduce in un'infinità di funzioni trigonometriche. È proprio nel calcolo di queste ultime che il 68881 dà il meglio di sé e così i tempi di elaborazione di Turbo Silver e di Sculpt 4D al minimo si dimezzano, ma possono anche ridursi a un terzo, a seconda della complessità della figura da rappresentare. Certamente la scheda Hurricane è uno strumento utilissimo a chi fa grande uso di programmi ray-tracing. L'altro campo in cui la scheda Hurricane potrebbe fare miracoli è la programmazione scientifica. Ci riferiamo ai programmatori (generalmente di Fortran e Pascal, più raramente di C) che utilizzano il computer con programmi propri di carattere scientifico-matematico. Questo tipo di utente di Amiga è piuttosto trascurato poiché è vero che esiste un ottimo compilatore Fortran per Amiga (prodotto dalla Absoft), ed è pur vero che di tale compilatore esiste la versione per 68020-68881, tuttavia quest'ultima versione è irripetibile in Italia, mentre la versione per 68000 è disponibile solo tramite la cosiddetta "importazione parallela" (leggi -pirati-). Per quanto riguarda il Pascal esiste l'ottimo compilatore Mcc Pascal, ma esso non prevede l'uso



con 68020 e 68881. Per il C le cose vanno meglio poiché ormai sono disponibilissimi i compilatori che possono sfruttare i due preziosi chip della scheda Hurricane e con tali compilatori si ottengono risultati eccezionali. Purtroppo, però, il C è un linguaggio di livello troppo basso per poter essere utilizzato efficacemente nel campo della programmazione scientifica, dove il Fortran regna ancora incontrastato.

Vediamo ora il risultato di alcuni benchmark, ovvero di programmi creati appositamente per testare la validità della scheda. Il valore di questi programmi è naturalmente molto relativo ma, comunque, dovrebbero fornire un'idea più precisa sul valore di una scheda.

Sul disco fornito insieme alla Hurricane sono disponibili due benchmark e un programma di prova. Ciascun programma è stato compilato in tre versioni: con le routine standard leee in una versione fast floating point e in una versione che prevede l'uso del coprocessore matematico. Il benchmark Savage serve per testare l'abilità di un computer nel calcolo delle funzioni trascendenti, quindi, c'è da aspettarsi una prestazione eccezionale da parte del coprocessore matematico. Questo si verifica puntualmente: la seguente tabella mostra i tempi di esecuzione, in secondi, del programma e ci pare molto esplicita.

	68000	Hurricane	Hurricane con mem. 32 bit
IEEE	47,14	1,44	0,82
FFP	9,92	5,68	3,44
68881	-	0,40	0,38

È importante notare che la memoria a 32 bit raddoppia la velocità di Hurricane, a meno che sia previsto un uso intenso del coprocessore matematico, nel qual caso l'accesso alla memoria è più limitato.

Il secondo benchmark è il Sieve e non prevede calcoli in virgola mobile, pertanto non ha nessun peso la presenza del 68881. Sieve gira in 47 secondi su Amiga con 68000, in 20,3 secondi su Amiga con Hurricane e in soli 12,96 secondi su Amiga con Hurricane e memoria a 32 bit. L'altro programma utilizzato per testare la Hurricane è un generatore di insieme di

Mandelbrot e come tale utilizza molto il calcolo delle quattro operazioni in virgola mobile. Anche in questo caso riportiamo la tabella con i tempi di esecuzione.

	68000	Hurricane	Hurricane con mem. 32 bit
IEEE	65,7	33,8	15,2
FFP	36,3	30,0	16,8
68881	-	03,1	02,9

Appare evidente ancora una volta il grande salto di qualità portato dal coprocessore matematico ogni qualvolta sia possibile servirsene. Passiamo ad analizzare l'ultimo punto rimasto in sospeso: la compatibilità tra la scheda Hurricane e i programmi Amiga creati per il 68000. Per quanto riguarda i programmi applicativi non abbiamo avuto nessun problema: abbiamo effettuato diverse prove e tutto ha girato perfettamente. Nel caso in cui sorgano problemi dovuti alla memoria cache del 68020 è possibile servirsi dell'istruzione Cache Off per risolverli, mentre, qualora si incorra nell'altra possibile causa di incompatibilità, ovvero la violazione di privilegio causata nel 68020 dall'uso dell'istruzione Move Sr, basta lanciare il programma TrapMoveSr e lavorare tranquilli. Abbiamo, invece, avuto diversi problemi nell'uso dei giochi: alcuni di essi si rifiutano di partire, evidentemente bloccati dalla memoria cache, mentre altri funzionano, ma la grafica risulta notevolmente alterata. In questi casi non è possibile utilizzare le istruzioni CacheOff o TrapMoveSr poiché i giochi solitamente partono in autostart senza servirsene della Startup-sequence, pertanto non è possibile utilizzarli. Sarebbe stato meglio poter disporre di un interruttore per selezionare il vecchio 68000 in caso di bisogno.

**Gianni Arioli**

*L'espansione Hardital  
SuperOtto e la scheda Hurricane  
sono disponibili presso  
Computer Center, via Forze  
Armate 260 20152 Milano  
Tel. 02/4890213*



# Cosa, Come, Quanto?

Notizie, novità e prezzi da tutto il mondo

## Nintendo

Alcuni anni fa (sembra davvero un'eternità, ma è passato un solo decennio, anno più, anno meno) fu lanciata sul mercato una novità assoluta: i giochi elettronici. Prima nelle sale giochi e nei bar, poi nelle case, i videogiochi iniziarono a diffondersi sempre più capillarmente. Si trattava di videogiochi molto semplici, spesso in bianco e nero, un classico esempio è il tennis giocato con due palette controllate da due paddle. A questa rivoluzione ne seguì ben presto un'altra, ancora più eclatante: il personal computer. In brevissimo tempo è parso evidente che il personal computer, o meglio l'home computer, era lo strumento più adatto per i videogiochi: si passava da una macchina specializzata e quindi limitata a un apparecchio più versatile. Il seguito lo conoscete benissimo: nel giro di pochissimi anni sono stati venduti milioni di home computer e il mercato è sempre in ascesa.

Di macchine specializzate per i videogiochi non se n'è più sentito parlare, finché la Nintendo non ha deciso di entrare in competizione con gli affermatissimi C64 e compagni con il proprio videosistema. Il sistema di base Nintendo si chiama Control Deck ed è composto di una consolle, due pulsantiere e un videogioco. La consolle è una scatola nuda, nel senso che vi si trovano solo due tasti, la spia di accensione, le prese per le pulsantiere e per il televisore. Non sembra certamente un computer, anche se in realtà si tratta proprio di un elaboratore basato su un veloce microprocessore a otto bit.

Le pulsantiere costituiscono l'interfaccia tra la consolle e il giocatore, e sono state studiate appositamente per ottenere il migliore controllo dei videogiochi.

Le pulsantiere raccolgono quattro pulsanti, di cui due sono piccoli e servono per il controllo delle opzioni dei giochi, mentre gli altri due sono più grandi e permettono di controllare lo sparo, il cambio, il salto o qualsiasi altra caratteristica del gioco. Non poteva mancare un controllo bidirezionale tipo joystick. Il videogioco è un classico della Nintendo: il mitico Super Mario Bros. È possibile espandere il siste-

ma Nintendo con tre interessanti accessori: la classica pistola ottica per mirare direttamente allo schermo, l'incredibile Rob, un piccolo robot programmabile, e inoltre agli inizi del 1990 farà la sua apparizione il Power Glove (guanto di potenza) che permette di interfacciare in modo estremamente efficace il giocatore con la macchina. Non abbiamo ancora potuto ammirare il Power Glove, ma ci pare un'idea decisamente originale ed efficace. Per il sistema Nintendo sono già disponibili numerosi titoli di videogiochi, pertanto non resta che aspettare come reagirà il mercato a questa novità.

Negli Stati Uniti il videosistema Nintendo è già molto diffuso: a nostro avviso in Italia l'enorme disponibilità di software a prezzi davvero stracciati per i computer più diffusi costituirà un serio ostacolo alla diffusione di un sistema il cui software non può essere copiato con facilità.

## Mk6 la nuova cartuccia per C64

La Nuova Newel distribuisce una nuova versione della più potente cartuccia per Commodore 64 attualmente sul mercato. Alla acclamatissima Mk5 segue la Mk6 (in figura).

La nuova versione offre le stesse eccezionali funzioni della precedente ma dispone di opzioni ottimizzate. Fra queste il turbo loading, che permette di caricare 202 blocchi di programma in sei secondi, generatore di vite infinite, mo-



# Cosa, Come, Quanto?

Notizie, novità e prezzi da tutto il mondo

nitor per linguaggio macchina, ricerca automatica degli sprite in memoria, freezing di tutta la memoria e tante altre utilissime funzioni.

*MK6 è un prodotto della Datel Electronics ed è disponibile presso la Nuova Newel, via Mac Mahon, 75 - Milano - Tel. 02/323492.*

## Ab Video

In questo senso può risultare preziosa la recente iniziativa assunta dalla Ab Video, una società che distribuisce videocassette. Se questa iniziativa avrà successo sarà possibile noleggiare sia il sistema Nintendo, sia i videogiochi, presso i videonegozianti che aderiranno alla proposta Ab Video. Si parla di 30mila lire settimanali per il noleggio del sistema con un videogioco, e diecimila lire settimanali per il noleggio del solo videogioco. L'idea è sicuramente buona e aspettiamo di vederne il seguito.

## Computer Center

La Computer Center di Milano propone alcune interessanti offerte per gli utenti Amiga. Ci sembra particolarmente interessante Flashbank, un hard disk su sche-

da per Amiga 2000 che consente l'auto boot. Flashbank è formattato con il Fast File System che consente una veloce scrittura e lettura dei dati: la velocità dichiarata è di 7.5 Mbit al secondo.

Flashbank è disponibile in tre modelli dal prezzo molto invitante: il modello da 32 Mb costa 690mila lire, mentre i due modelli da 60 Mb con tempo di accesso di 40 oppure 25 millisecondi costano rispettivamente 990mila e 1 milione 240mila lire.

## Bbs

Già da alcuni anni è operativa una rete telematica di banche dati dedicate a due modelli di computer: i Commodore Amiga e gli Ibm compatibili.

Chi dispone di un modem può accedere al patrimonio di queste banche dati tramite la Bbs Rozzano. È sufficiente comporre il numero 02/89202186. La Bbs Rozzano, che si trova nell'omonima cittadina alle porte di Milano, è funzionante ventiquattrore su ventiquattro. È possibile collegarsi a 300, 1200 e anche 2400 baud con formato N81 ovvero nessuna parità, otto bit di dati, un bit di stop.

Giochi nuovi Amiga: Shinobi, Shufflepuch, Paperboy, Altered beast, Dragon sea, Strideri, Rally cross, Frindish fredy (alcuni occupano più di un disco)

## Per trasferire i programmi di RE&C

Molti lettori hanno incontrato difficoltà nell'eseguire le operazioni di trasferimento dei programmi da nastro a disco. L'utility Dsave richiedeva infatti di specificare due indirizzi che definivano la zona occupata dal programma da trasferire. La sequenza di operazioni non era sempre uguale per tutti i programmi, perciò molti lettori inesperti non hanno potuto operare tutti i trasferimenti desiderati. In questo numero è disponibile sulla cassetta allegata una nuova utility denominata Dsave v2. Dopo il caricamento del solito menu all'inizio della cassetta, tutti coloro che vorranno trasferire i programmi sul disco dovranno caricare e lanciare Dsave v2. Il menu offre tre possibilità:

- 1 - La cassetta verrà letta e il primo programma incontrato caricato. A questo punto viene chiesta conferma per il trasferimento sul disco, dipendendo si passerà al caricamento del successivo programma sulla cassetta e così via.
- 2 - Scegliendo la seconda opzione, invece, verranno salvati su disco tutti i programmi automaticamente, senza selezioni da parte dell'utente. In questo caso bisogna ricordare che difficilmente tutti i programmi sulla cassetta di RE&C staranno su una sola facciata del disco su cui vengono trasferiti, perciò a un certo punto il trasferimento si bloccherà e il drive segnalerà errore per mancanza di spazio. Vi consigliamo quindi di utilizzare l'opzione 1 anche se volete trasferire tutti i programmi della cassetta.
- 3 - Questa opzione consente di visionare la directory del disco.

## Programmare con due drive

Leggo da molto tempo *Radio Elettronica e Computer* e, grazie agli articoli che vi trovo pubblicati, ho imparato quasi tutto sul Basic e comincio a programmare con successo qualcosa in linguaggio macchina. Da qualche tempo, però, cerco informazioni sulla gestione del secondo drive (quello con numero di device 9). Tutto quello che so è che per indirizzare le operazioni di loading al secondo drive, in Basic, è sufficiente specificare `LOAD"NOME",9,1` invece che `LOAD"NOME",8,1`. Io, però, sto cercando un sistema per fare in modo che un mio programma multi-load possa essere caricato indifferentemente sia dal drive numero 8 sia dal numero 9. So che è possibile perché possiedo alcuni programmi scritti da altri che sono in grado di farlo, ma non posso esaminarli essendo in Basic compilato o in linguaggio macchina.

**Luciano Poggi**  
Seregno (Mi)

*I programmi multi-load, per chi non lo sapesse, sono costituiti da due o più file. Il primo di essi è il file di boot. Generalmente, per caricare un programma multi-load occorre caricare e lanciare il file di boot che prevede a caricare il file (o i file) che costituiscono il resto del programma. Al termine del caricamento multiplo il programma*

*entra in esecuzione regolarmente. Tutti i problemi relativi a questo tipo di programma possono definitivamente essere risolti con un programma di compatto e un pizzico di abilità tecnica.*

*In alcuni casi, però, il procedimento di multi-load si ripete anche durante l'esecuzione del programma, per esempio quando una certa zona di memoria deve essere condivisa da due o più segmenti di programma in linguaggio macchina o anche solo da file di dati, a seconda della funzione svolta.*

*I programmi di boot in Basic contengono spesso una o più linee di programma del tipo:*

```
50 LOAD "FILE 2",8,1
```

*Un simile programma di boot, se lanciato dal drive numero 9, restituisce un messaggio d'errore del tipo "device not present" o "file not found". Il problema, tuttavia, può essere aggirato modificando la linea in questo modo:*

```
50 DV=PEEK(186):LOAD "FILE 2",8,1
```

*La locazione 186 contiene il numero dell'ultimo drive utilizzato. Quando l'istruzione viene eseguita il computer conosce già il valore dell'ultimo drive utilizzato perché, sicuramente, è già stato impartito il comando di Load per caricare il boot.*

*Tenete solo presente che la linea:*

```
50 LOAD"FILE 2",PEEK(186),1
```

*non può funzionare perché, quando il computer trova l'istruzione Load, azzerla la locazione di memoria, prima, quindi, dell'istruzione Peek, invalidando l'operazione.*

*Se possedete dei programmi multi-load in linguaggio macchina e volete modificarli per fare in modo che indirizzino indifferentemente al drive 8 o 9, dovete munirvi di un monitor con l'opzione di ricerca. Dovete cercare in memoria la sequenza di byte*

```
A9 02 A2 08 A0 02
```

*che corrisponde a istruzioni macchina equivalenti a Open 2,8,2. Trovata la sequenza dovete sostituire al valore 08 il valore BA. Questo cambiamento impone al programma di servirsi dell'ultimo drive utilizzato, ovvero quello da cui avete lanciato il primo file (quello di boot), risolvendo ogni problema. L'unico accorgimento è quello di assicurarsi che in memoria non vi sia più di una serie di valori corrispondente a quella suggerita e in caso contrario, operare ancora la modifica. Altre routine di chiamata al device 8, infatti, potrebbero fallire invisibilmente riportando l'attenzione del computer su quel drive generando errori e malfunzionamenti.*

# Tutto **COMMODORE**

Sped. in Abb. Postale Gr. III/70%

Anno II - Numero 29 - Dicembre 1989 - L. 13.000

## Grafica

**TUTTO PER  
FARE GRAFICA  
AL MEGLIO**



### TESTI

19 set  
di caratteri  
giganti

### GRAFICI

Crearli,  
colorarli,  
stamparli

### DISEGNI

Presentare  
software  
e stampare

Gruppo Editoriale  
**JCE**

*è in edicola*

# LA DIFESA AD OLTRANZA

...QUANDO TRA IL DISCHETTO E LA SUA UNITÀ DISCO

NON ESISTE PIÙ RAPPORTO

Le norme Ansi/Ecma stabiliscono che il dischetto debba avere un "clipping level" non inferiore al 40%. Ciò permette al dischetto di colloquiare con il drive in condizioni di sicurezza.

Ma cosa succede se il drive esce dalle sue norme di sicurezza? Per esempio se si stira o degrada per invecchiamento o opera in condizioni ambientali critiche? Solamente un dischetto di caratteristiche superiori, con un "clipping level" del 65%, può continuare il rapporto con la sua unità a disco. Ecco perché il SIXTY FIVE difende ad oltranza i tuoi dati.



è un prodotto  
**datamatic**  
TESTA MINIE CALCOLABRE

VIA AGORDAT, 34  
20127 MILANO  
Tel. (02) 2871131 (8 linee r.a.)  
Telex 315377 SADATA I

VIA CITTÀ DI CASCIA, 29  
00191 ROMA  
Tel. (06) 3273581 (3 linee r.a.)  
FAX (06) 3283894

C.SO MONCALIERI, 269/E  
10133 TORINO  
Tel. (011) 6367171 (3 linee r.a.)  
FAX (011) 6367035

