

7 GAMES PER CBM 64 e 128
7 GAMES PER C16 e PLUS 4

GO GAMES

mensile d'informatica e video-games - n. 6 - febbraio 1986 - L. 8000

7 VIDEO-GAMES
PER CBM 64 E 128

1) KOREA - 3) JET MAN - 5) PUGILATO - 7) SCARABEO

2) NIDIATA - 4) DON RODRIGO - 6) STAR MAN

7 VIDEO-GAMES
PER C16 E PLUS 4

1) DECATHLON - 3) FIRST POSITION - 5) COMBAT

2) BIORITMO - 4) FILETTO - 6) SPACE IN - 7) SKRAMBLE



GO GAMES

Mensile di informatica
e video giochi

Anno II
N. 6 - Febbraio '86

EDITORE:
Editions Fermont s.r.l.
20121 Milano

REDAZIONE:
Via Cialdini, 11
20161 Milano
Tel. 02/6453775/6

FOTOLITO:
Fotolito RVM s.n.c.
Via Puricelli, 4
20147 Milano

COMPOSIZIONE:
Nuovo Gruppo Grafico s.r.l.
Frazione Venina, 7
20090 Assago (Mi)

STAMPA:
A.G.E.L. s.r.l.
Viale dei Kennedy, 92
20027 Rescaldina

DISTRIBUZIONE:
MePe
Via G. Carcano, 32
20141 Milano

DIRETTORE RESPONSABILE:
Amilcare Medici

Fotografie di Stefano Monti

Numeri arretrati: Ogni numero arretrato £. 8.000 più £. 3.000 di spese postali - Versamento da effettuare sul c/c postale n. 37332202 intestato a EDITIONS FERMONT, Via Cialdini, 11 20161 Milano

ATTENZIONE

CBM 64

Per il CBM 64 ti proponiamo un nuovo sistema di caricamento che ti permette di scegliere il gioco che vuoi caricare e di posizionare il nastro con l'avanzamento veloce (F.FWD) subito prima del gioco da te prescelto, quindi di procedere al caricamento normale. Con questo sistema eviti di dover passare tutto il nastro per cercare il programma che ti interessa.

Le operazioni da fare sono:

- 1) Digita Load e premi Return.
- 2) Attenti che sul video compaia la presentazione.
- 3) Premi Stop sul registratore.
- 4) Dopo qualche secondo apparirà una schermata con l'elenco dei giochi preceduto da un numero e la scritta «Programma N°» col cursore che lampeggia.
- 5) Inserisci il N° corrispondente al programma desiderato e premi «Return».

6) Comparirà la scritta «premi F.FWD» quindi il registratore si fermerà subito prima del programma da te scelto. A questo punto premi «STOP» e successivamente premi «PLAY».

AVVERTIMENTO: se lo schermo si riempirà di righe colorate significa che il caricamento procede regolarmente. Se non escono le righe torna indietro all'inizio del gioco e premi nuovamente Play.

C16 / PLUS 4

Ecco le istruzioni per il caricamento dei programmi: Avvolgere completamente la cassetta dalla parte che si desidera caricare. Quindi digitare LOAD & RETURN e far iniziare il caricamento. Quando ricompare il cursore digitare RUN & RETURN ed attendere. La prima volta che si caricano i programmi conviene azzerare il contatore del registratore alla fine dell'avvolgimento e scrivere il numero dell'inizio del gioco in modo che in un tempo successivo si conosce l'esatto inizio del gioco.

ATTENZIONE: poiché da questo numero i videogiochi sono stati turbizzati, se l'Azimut non è perfettamente allineato, possono verificarsi problemi in fase di caricamento. In questo caso, è necessario riprovare.

l'uomo e il computer 4

Il mese scorso abbiamo parlato degli automi e della loro evoluzione come fantocci spettacolari e meccanici che simulavano forme e movimenti animali o umani. Ma volutamente abbiamo rimandato a questa quarta puntata l'aspetto forse più rilevante: l'automa o il robot (chiamatelo come vi pare) è costruito per sostituire l'uomo in alcune sue specifiche funzioni.

L'automa è cioè una macchina, può essere piacevole vederlo muoversi e scimmiettare dei comportamenti umani ma è molto più importante se queste sue azioni sono finalizzate ad uno scopo pratico. Non è perciò un giocattolo, deve essere un utensile semplice o complesso, una specie di protesi che subentra nel momento in cui si dimostra più efficiente dell'uomo.

Con queste premesse è però molto difficile definire che cosa sia un automa. Qualsiasi apparecchio meccanico o elettrico che ci aiuta nel nostro lavoro manuale o mentale è dunque un robot? Certamente no: l'automobile che trasporta degli oggetti sostituendo le nostre braccia e gambe e la calcolatrice da tasca che esegue velocemente delle complesse operazioni aritmetiche non sono robot. Allora perché alcuni robot industriali che non fanno altro che spostare dei pezzi meccanici in una catena di montaggio ed eventualmente compiere altre semplici operazioni di assemblaggio sono chiamati appunto robot?

È un errore poiché la distinzione tra **servomeccanico** (di qualsiasi tipo) e **robot** non si basa su quello che effettivamente compie quel congegno bensì sul « come » la compie.

Ritorniamo alla storia, al 1920 in cui ci eravamo lasciati il mese scorso con l'accenno alla commedia R.U.R. di Karel Capek. R.U.R. è

l'acronimo (cioè un nome formato dalle iniziali di più parole) di Rossum Universal Robot.

Questo Rossum, un ipotetico scienziato, ha scoperto il modo di costruire degli esseri simili all'uomo ma molto più specifici. Sono lavoratori instancabili e fortissimi ma non posseggono degli « optionals » tipicamente umani come i sentimenti, la coscienza, la fantasia...

Sono macchine biologicamente perfette che, sobbarcandosi tutte le faticose incombenze per la nostra sopravvivenza, dovrebbero permettere all'uomo di dedicarsi solo al pensiero. Ma il perfezionismo dei loro creatori le migliora a tal punto che questi robot si ribellano e vogliono sostituirsi all'uomo:

ALQUIST - I Robot non sono la vita. I Robot sono macchine.

SECONDO ROBOT - Signore, eravamo macchine; ma con la paura e il dolore siamo diventati...

ALQUIST - Cosa?

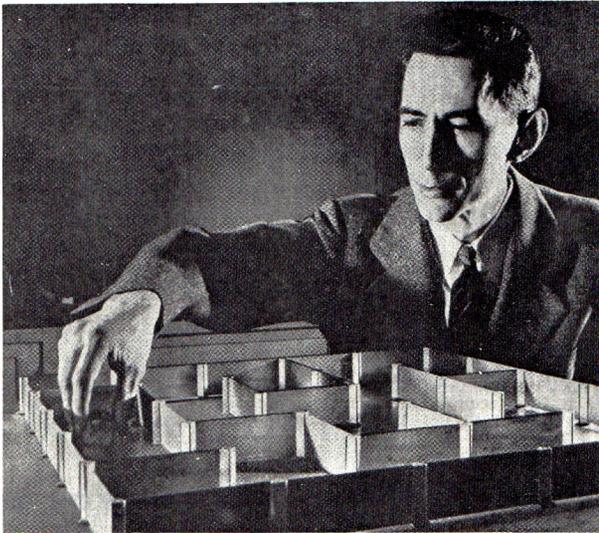
SECONDO ROBOT - Siamo diventati anime.

QUARTO ROBOT - C'è qualcosa che lotta con noi. Vi sono attimi in cui qualcosa entra in noi. Ci vengono idee che non hanno origine da noi.

Queste ultime frasi (1), anche se in modo molto teatrale ed un po' esagerato, ci possono essere d'aiuto per scoprire la differenza tra « macchina » e « robot ». Non si può infatti parlare di anima riferendosi ad un oggetto artificiale, d'altro canto usare il termine intelligenza può sembrare azzardato se si riferisce a quegli utensili, a quelle protesi di cui parlavamo sopra, ma lo è molto meno nei confronti

di altri apparecchi che possono reagire in modi differenziati a seconda degli stimoli, degli « inputs » che vengono loro dati. Ritornando agli esempi precedenti un'automobile va guidata, una calcolatrice va manipolata per avere delle risposte, invece il vero e proprio robot dovrebbe avere, entro certi limiti, la possibilità di elaborare da sé le sue reazioni confrontandosi con l'ambiente esterno ed ottimizzando le scelte di comportamento.

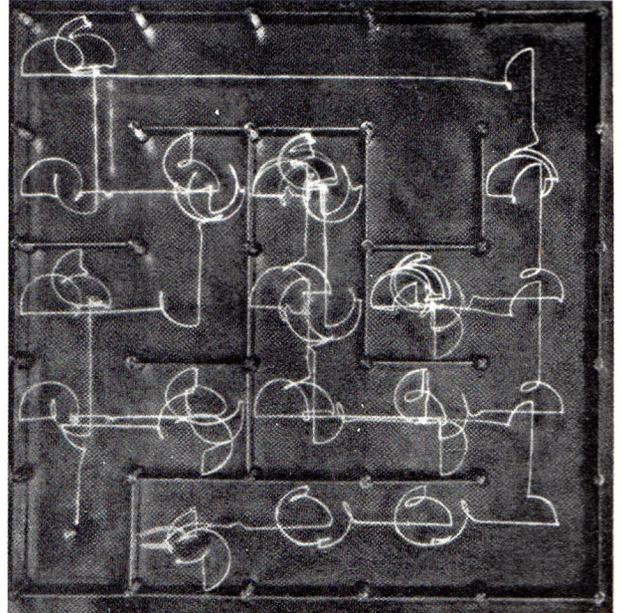
Se fare riferimento all'anima è assurdo e parlare di intelligenza può sembrare azzardato, il basarsi invece su un comportamento « reattivo » - e non pedissequamente passivo - fonda la ricerca e la sperimentazione nel campo dei robot.



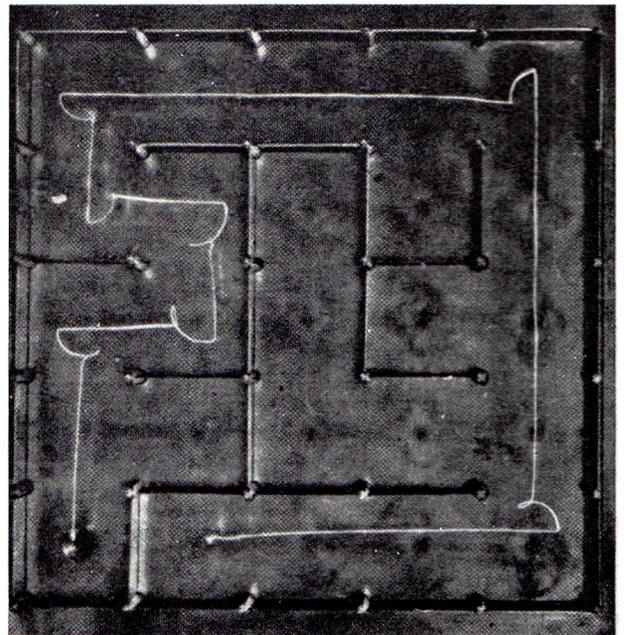
Claude Shannon e il suo labirinto con un topolino elettromeccanico

Nella figura 1 vediamo uno dei precursori della moderna informatica, il matematico americano Claude Shannon che mostra uno dei primi studi sull'apprendimento da parte di una macchina. Nel labirinto un topolino artificiale cerca la sua meta procedendo per tentativi ed errori. Questo topo è costruito con una calamita che viene mossa sotto il piano da un'altra calamita; i suoi baffetti di rame sono un « contattore » cioè un interruttore a contatto che fa mutare la direzione di spostamento della calamita guida. Quando il topo ha raggiunto il « formaggio » ovvero il suo obiettivo nel labirinto, la sua « memoria » (...uno dei primi el-

boratori elettronici) gli permette di ripetere il percorso senza mai sbagliare una volta (nella figura 2 il primo tentativo, nella 3 il secondo e successivi tentativi fintantoché il labirinto non viene modificato).



Il primo tentennante percorso del topo



ed il secondo in cui raggiunge la meta con sicurezza

È questa dunque una macchina che impara... e da lì a farla reagire in modi differenziati il passo è breve (il solito cugino che sa tutto di

BASIC ci direbbe che basta instaurare un ciclo di IF... THEN...).

Per imparare sono quindi necessari quelli che noi chiamiamo « sensi »: il tatto (i baffi del topolino), l'udito, la vista... per quest'ultima la tecnologia di trent'anni fa metteva a disposizione dei succedanei artificiali come le cellule fotoelettriche.

Ecco allora che nel 1950 Grey Walter, neurologo inglese, progetta e costruisce due tartarughe « elettroniche »: ELMER (Electronic Mechanical Robot) ed ELSIE (Electro Light Sensitive Internal External).

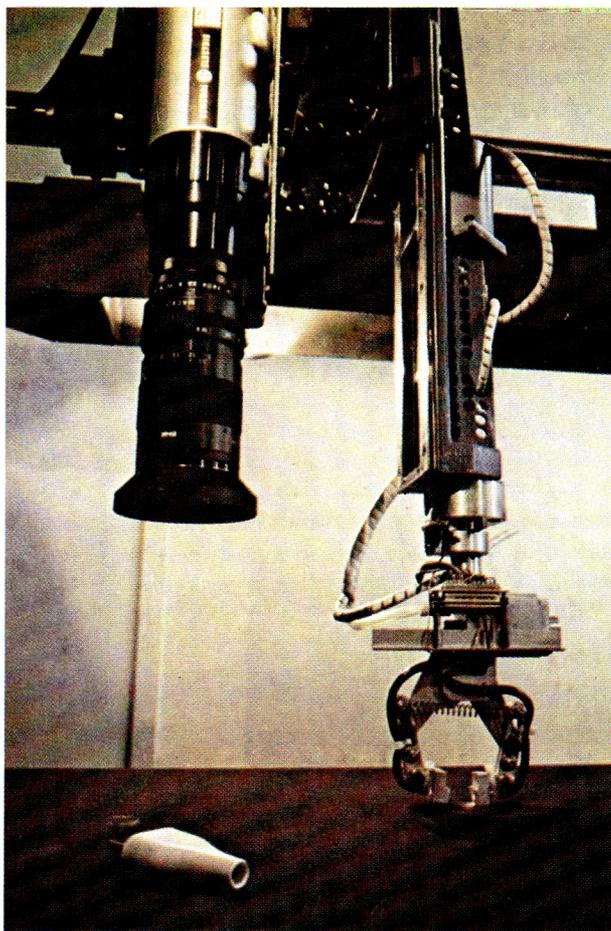
Elmer ed Elsie si muovevano per la casa di Walter evitando gli ostacoli grazie alle loro fotocellule ma facevano ancor di più: all'occorrenza ogni tartaruga cercava il cibo. Quando i suoi accumulatori erano prossimi all'esaurimento la cellula fotoelettrica la dirigeva verso una potente fonte luminosa sotto la quale era predisposta una succulenta presa di corrente per la ricarica. Non raggiungendo questa presa la tartaruga sarebbe morta. « Un'altra somiglianza con gli animali: quando non hanno più la forza di procurarsi il cibo, gli animali muoiono: e così fanno le mie tartarughe. Praticamente, ciò che può impedire loro di raggiungere la nicchia luminosa, è solo un ostacolo continuo, uno scalino ad esempio, o un tappeto di pelliccia dove affondino, oppure un cumulo di ostacoli a forma di labirinto in cui si smarrirebbero. »(2)

Il topo artificiale vive nel labirinto mentre la sua collega tartaruga fotosensibile muore: il vecchio concetto di **robot faber** cioè di robot esecutore si sta lentamente trasformando producendo i primi tentativi di **robot sapiens**.

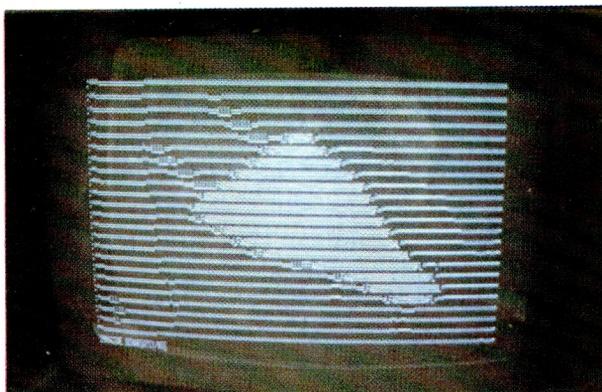
Per questo è di grande aiuto la **cibernetica**, la « la scienza del controllo e della comunicazione nell'animale e nella macchina ». Studi teorici sul funzionamento del nostro sistema nervoso portano infatti scienziati come Norbert Wiener e William Ross Ashby a elaborare dei semplici modelli di comportamento trasferibili a delle macchine. « Abbiamo visto nel caso della fotografia e processi analoghi che è possibile conservare un messaggio nella forma di un'alterazione permanente di certi elementi di immagazzinaggio. Nel reinserire l'informazione nel

sistema, è necessario far sì che queste alterazioni influiscano sui messaggi che percorrono il sistema. Uno dei modi più semplici di ottenere ciò è di avere, quali elementi alterabili di immagazzinaggio, parti che intervengano normalmente nella trasmissione dei messaggi, e di natura tale che il cambiamento delle loro caratteristiche dovuto all'immagazzinamento influisca sul modo in cui esse trasporteranno il messaggio in futuro. »(3)

Si comincia a parlare di « intelligenza artificiale », un'espressione da una parte ambigua (che cos'è, in fondo, l'intelligenza?) e forse contraddittoria (può una macchina pensare?). Ritorniamo sull'argomento in una prossima puntata, rimaniamo per ora un gradino più sotto, non entriamo nel « cervello » ma soffermiamoci negli organi nervosi periferici. Con gli stessi termini della neurologia possiamo parlare di « sensori » e « motori ».



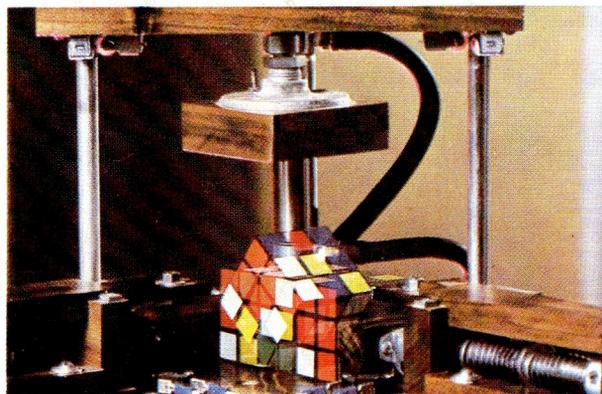
L'occhio del robot...



...e quello che vede

In questo ambito i robot di oggi si sono molto perfezionati. Sono in grado di vedere (figure 4 e 5) con delle telecamere e riconoscere la forma e l'orientamento di alcuni oggetti.

La « vista » ne permette la manipolazione sostituendo così le mani dell'uomo non tanto nei lavori di precisione (avete mai sentito parlare di robot orafo?) quanto nel trattamento di oggetti pericolosi (materiali radioattivi, per esempio). Il robot è instancabile, non si distrae e quindi non può avere incidenti sul lavoro ... a meno che non manchi la corrente. Ci sono tuttavia anche dei robot abilissimi e delicatissimi nel trattare gli oggetti. Nella figura 6 vediamo in azione **Robbie**, costruito da un gruppo di studenti di ingegneria dell'università dell'Illinois. Abbinato a un computer, riesce a risolvere il cubo di Rubik in soli due decimi di secondo (partendo da una qualsiasi configurazione) ma gli stantuffi che muovono le facce del rompicapo sono più lenti - anche se molto precisi - e ricompongono il cubo in 5 o 6 minuti. Il fatto curioso è che questo robot è



Le « mani » di Robbie al lavoro

stato realizzato da persone che non sono mai riuscite a riassetare il diabolico cubo ... è bastato scrivere un programma basato sulle istruzioni copiate da un libro.

Il robot, sempre guidato dall'uomo ma con una discreta autonomia, può essere presente e operare in luoghi inaccessibili all'uomo: « Voyager 1 e Voyager 2 furono lanciati in agosto e settembre 1977 con direzione Giove e Saturno. È difficile dire se fossero robot o no. È certo che la loro capacità di navigazione era notevole (orientamento in base alle stelle, ...) e non poteva essere diversamente visto i tempi di comunicazione con la terra, troppo lunghi per permettere un comando remoto. »(4)

Un'altra famiglia molto prolifica è quella dei robot casalinghi o domestici. Dagli anni '50 ad oggi sono stati fatti molti tentativi per costruire il maggiordomo perfetto: problemi di costo e di limitazioni nel loro operare li hanno però costretti nella categoria dei gadgets. Soltanto oggi, rinunciando a far loro pulire la moquette e lavare i vetri e riscoprendoli invece come « personal su rotelle » con le peculiarità di elaboratori che sanno anche colloquiare, la nuova generazione di **home-robot** può presupporre ulteriori sviluppi come futuri nostri compagni utili ma anche, un poco, intelligenti.

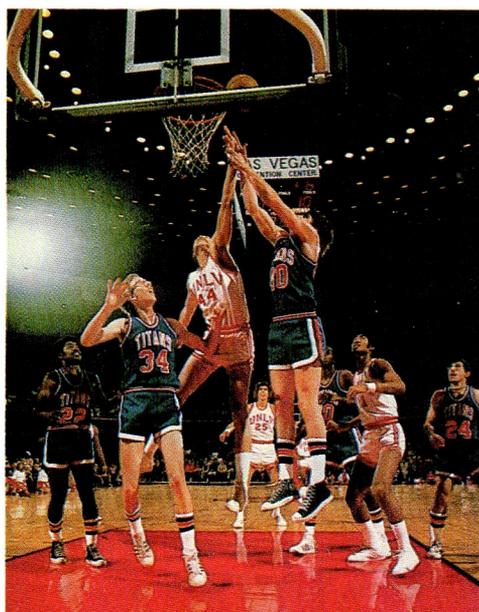
Bye bye a tutti, anzi byte byte e arrivederci al prossimo mese.

Aldo Spinelli

- (1) Karel Capek, **R.U.R.**, Einaudi, Torino 1971
- (2) Grey Walter in **Scienza e Vita**, Aprile 1951
- (3) Giuseppina e Maria Gini, **ROBOT controllo, programmazione, interazione con l'ambiente**, clup, Milano 1983

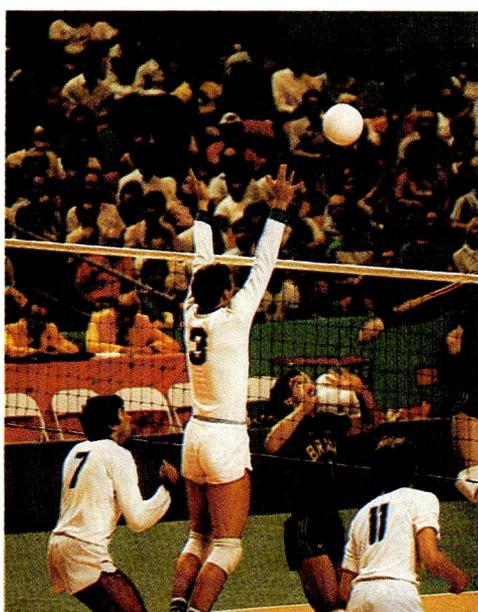
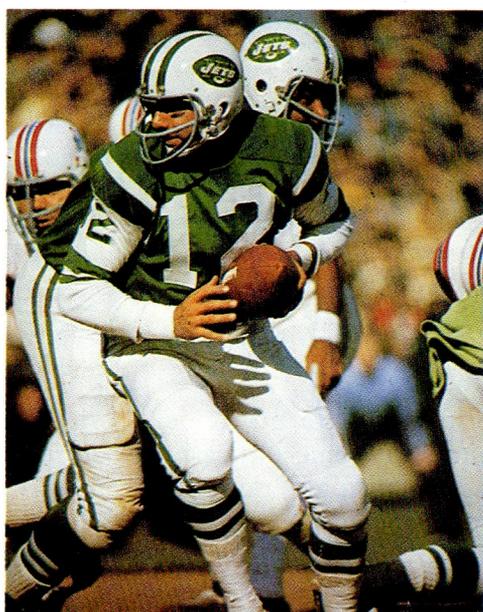


CALCIO



PALLACANESTRO

**FOOTBALL
AMERICANO**



PALLAVOLO

i nostri magnifici supergiochi

DECATHLON (Parte 2)

C 16 e PLUS 4 - Joystick in porta 2
Parola chiave: STADIO



Nel numero precedente di Go Games vi avevamo presentato la prima parte delle Olimpiadi: ecco ora la seconda.

Nella prima parte vi siete cimentati nelle gare di velocità ora dovrete gareggiare:

salto in lungo
lancio del peso
lancio del giavellotto

All'inizio del gioco, per iniziare le competizioni, dovete premere il tasto del joystick, quindi inizierà la competizione con il salto in lungo. Spostate velocemente il joystick a destra e a sinistra per correre, quando vedrete la pedana premete il pulsante fino a che non avrete raggiunto i gradi voluti. Raggiunti i gradi desiderati (si consiglia di arrivare a 45 gradi) lasciate il pulsante di fuoco e comincerete a saltare. Nelle due gare seguenti il procedimento è identico, tranne che saltare lancerete il peso o il giavellotto.

Avete 3 tentativi per ogni specialità.

BIORITMO

C 16 e PLUS 4
Parola chiave: CICLO

Bioritmo mostra un modello fisico, emotivo e intellettuale che indica i tuoi giorni "alti" e "bassi" per ogni periodo di tempo, di 39 giorni. Bioritmo cerca di predire i giorni "buoni" e "cattivi" nel passato, presente e futuro.

Lo stato fisico, emotivo ed intellettuale di ogni persona segue periodi fissi.



I simboli ed i periodi, come appaiono sul grafico, sono così definiti:

Pallino vuoto - stato fisico 23 giorni
 Pallino pieno - stato emotivo 28 giorni
 Asterisco - stato intellettivo 33 giorni

Questi cicli formano un'onda; i giorni critici significativamente avvengono durante il passaggio fra giorni "alti" e "bassi"; la teoria del bioritmo suggerisce che durante i giorni critici si è più portati agli errori, agli imprevisti ed agli abbattimenti fisici.

I giorni "critici" sono indicati quando l'onda passa per l'asse centrale.

Il programma è stato concepito per l'uso con la stampante, ma voi potete senza dubbio visualizzare il vostro bioritmo sul video.

Per iniziare il gioco, inserite il joystick in porta uno e premete "spazio" dopo alcuni secondi inizierà la gara. Il pannello di guida vi fornirà delle indicazioni: in alto a sinistra il tempo che vi rimane, in basso (sempre a sinistra) il tempo del giro, sopra il volante il record, in alto a destra il vostro punteggio e in basso la velocità con la marcia inserita.

Per cambiare dovete premere il pulsante del joystick, per aumentare la velocità sposterlo in avanti, per rallentare sposterlo indietro. Naturalmente per curvare lo si deve spostare a destra o a sinistra.

FIRST POSITION

C 16 e PLUS 4 - Joystick in porta 1
Parola chiave: AUTODROMO

Finalmente siamo riusciti a realizzare nella versione per il vostro computer di questo favoloso gioco da bar.

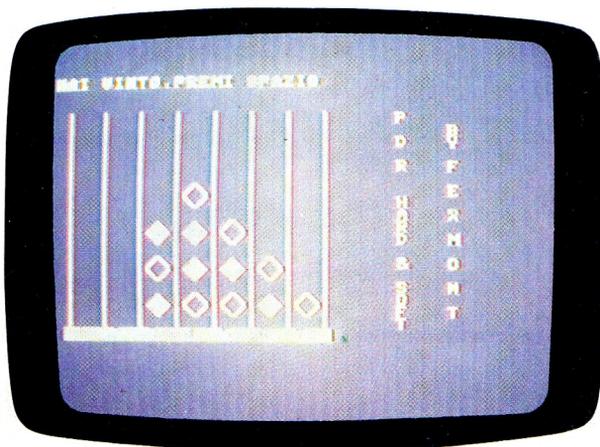
Dovete guidare la macchina di Formula Uno su un percorso tutto curve e chicane imprevedibili; ci saranno anche delle macchine che vi daranno molto filo da torcere se vorrete superarle.

FILETTO

C 16 e PLUS 4 - Tastiera
Parola chiave: PEDINA

Sfidate il vostro computer o un vostro amico in questo gioco, ma, attenzione: dovrete pensare molto. Si tratta di un gioco di società molto conosciuto. In una griglia dovrete cercare di fare "filetto" (in orizzontale, verticale o diagonale) mettendo vicine 4 pedine del vostro colore.

Potete giocare contro un vostro amico o contro un computer.



In questo caso potrete vedere le percentuali di scelte del computer per ogni colonna.

Il computer è stato programmato per rispondere sempre con la miglior mossa, quindi vi sarà molto difficile batterlo.

La vostra scelta viene effettuata premendo il numero corrispondente alla colonna, il computer farà il resto.

SPACE IN

C 16 e PLUS 4 - Joystick in porta 1

Parola chiave: INVADERS

F1 - pausa

F2 - fine pausa



Ecco la versione per il vostro computer del più famoso gioco spaziale creato per i video giochi da bar.

Siete al comando di una nave spaziale fornita di laser, il vostro compito è quello di fermare un'invasione aliena.

Gli invasori sono disposti in squadre, terminata la prima ce ne sarà subito una seconda pronta ad attaccarvi.

Il livello di gioco viene scelto mediante joystick e se per caso durante il gioco sarete stanchi di sparare potete premere il tasto F1 per fare una pausa o mangiare la merenda. Per ripartire premete F2.

SKRAMBLE

C 16 e PLUS 4 - Joystick in porta 2

Parola chiave: CAVERNA



Sei stato invitato da un illustre professore di geologia a scoprire una nuova caverna sotterranea. Ad un certo punto trovate dei resti di una civiltà passata. Vi consultate e decidete di proseguire. Non l'avete mai fatto!!

Andando avanti trovate una base aliena fornita di tutte le migliori armi per distruggere la terra. Dovete cercare di distruggerle tutte colpendo anche i barili di benzina per rifornire la vostra astronave.

Il viaggio si snoda in sei tipi diversi di schermi e di difficoltà. Cercate di sopravvivere e di portare a termine la vostra missione, altrimenti questo sarà l'ultimo numero di go games che vedrete!! (Scherziamo).

COMBAT

C 16 e PLUS 4 - Joystick in porta
Parola chiave: SPAZIO



Sei in una navicella spaziale LP2718 ed all'improvviso la spia dell'allarme rosso si illumina: è in arrivo uno storno di alieni. Tramite i laser in tuo possesso devi distruggere il maggior numero di alieni stando però attento a non scontrarti con loro.

Quindi devi stare attento al suono d'allarme. Quando lo senti devi spostarti finché non si spegne.

Nei laser è inserito un dispositivo di sicurezza che impedisce di far salire troppo la temperatura. Questo dispositivo è visualizzato in basso a destra: cerca di tenerlo d'occhio.

Per scegliere il livello di gioco devi spostare il Joystick.

KOREA

CBM 64 - Joystick in porta 2
Parola chiave: MISSIONE



Guidate il vostro eroe in un'entusiasmante missione di guerra che lo impegnerà contro nemici spietati.

Solo la vostra abilità potrà consentire lo sfruttamento di tutte le sue doti.

Il vostro compito è quello di annientare il nemico, assalendo le sue postazioni fortificate in cui si è rifugiato.

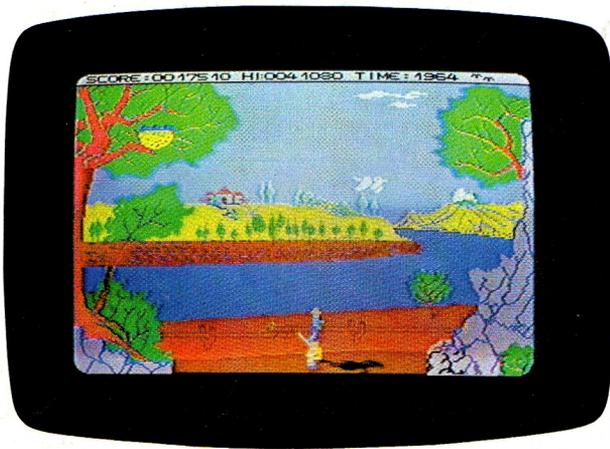
Il vostro eroe ha disponibili diversi tipi di armi (mitra, arma bianca, bombe a mano). Per selezionare, premere la barra spaziatrice mentre giocate.

NIDIATA

CBM 64 - Joystick in porta 2
Parola chiave: UCCELLO

In "Nidiata" dovete aiutare un simpatico uccello a costruire il proprio nido. Per farlo, dovete innanzitutto guidarlo nella raccolta dei ramoscelli mentre cadono dalle piante; dovranno essere colti al volo premendo Fire

al momento giusto e depositati nel nido ripremendo Fire.



JET MAN

CBM 64 - Joystick in porta 2
Parola chiave: PIANTA

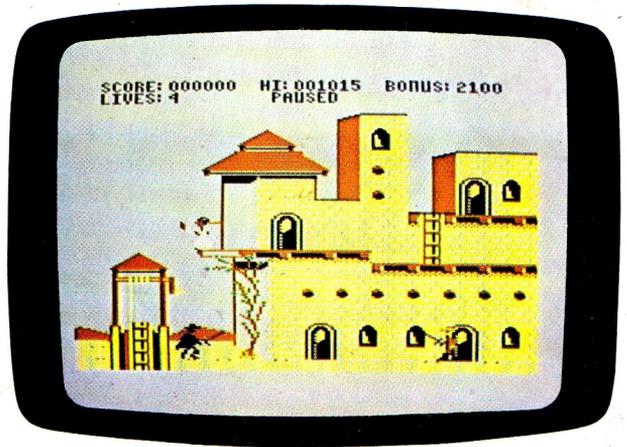


Jet Man dovrà curare la crescita delle proprie piante chiuse nella serra cosmica, destreggiandosi fra tante difficoltà.

Con lui dovrete raccogliere le pianticelle dalla parte superiore della serra e portarle nei vari ripiani, facendo tanti viaggi quanti saranno necessari alla crescita di tutte le piante.

DON RODRIGO

CBM 64 - Joystick in porta 1
Parola chiave: ZETA



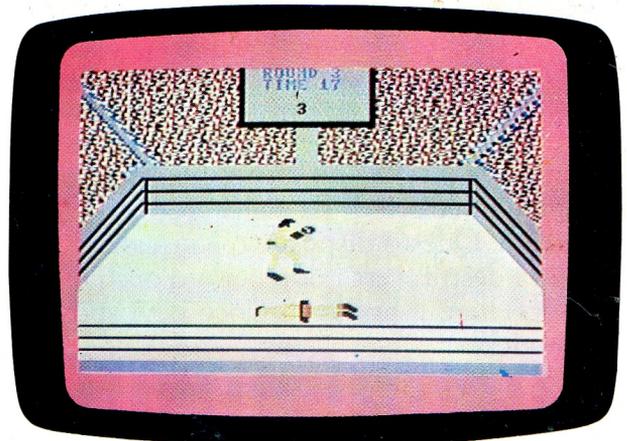
Il leggendario Don Rodrigo, nobile Messicano, in una notte si trasforma e diventa un giustiziere mascherato.

Dovrete guidarlo nella lotta contro i fuorilegge che hanno rapito la sua amata.

Il compito certamente non è dei più facili ma la sua spada lo aiuterà.

PUGILATO

CBM 64 - Joystick in porta 1 (per un giocatore)
Parola chiave: RING



Se siete appassionati di pugilato, con questo gioco potrete vivere le fasi entusiasmanti di un favoloso incontro.

Si può giocare contro il computer oppure in due giocatori.

Alla fine di ogni ripresa sarà visualizzata la situazione generale dell'incontro con le condizioni fisiche dei pugili.

Per iniziare le varie riprese premere Fire.

Comandi: quando compare la schermata iniziale si potrà optare per una dimostrazione oppure no, premendo F1 o F3 rispettivamente.

F1 - seleziona numero riprese

F3 - numero giocatori

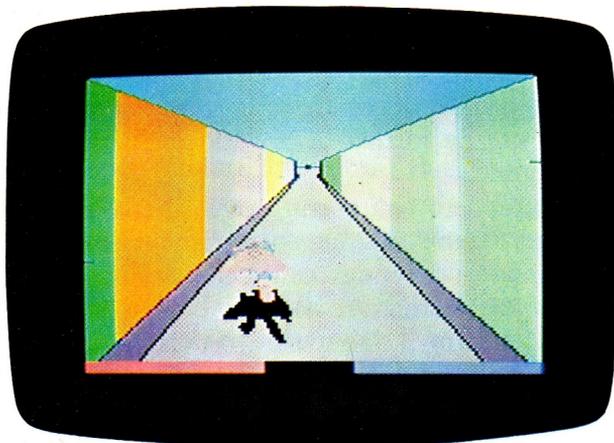
F5 - livello difficoltà

F7 - inizio combattimento

STAR MAN

CBM 64 - Joystick in porta 2 (per un giocatore)

Parola chiave: METROPOLI



Stupendo gioco di abilità.

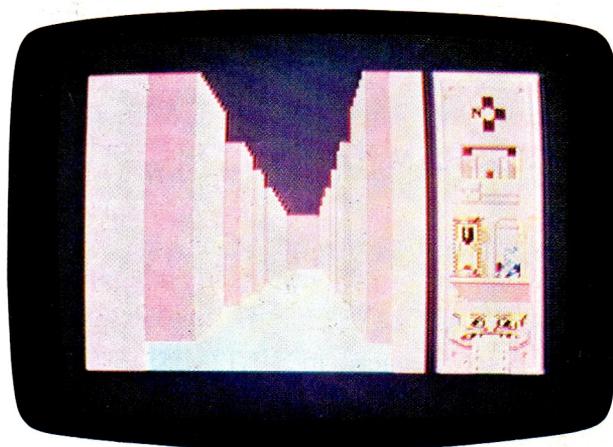
Dovrete guidare Star Man nella difesa della metropoli contro il suo più acerrimo nemico. Disponi di una potente pistola laser, il tuo nemico dispone di scudi protettivi che faranno deviare i raggi laser rendendo il compito sempre più arduo.

Non ci resta che augurarvi buona fortuna.

SCARABEO

CBM 64 - Joystick in porta 2

Parola chiave: FARAONE



Un astronauta, facendo un fantastico viaggio a ritroso nei secoli, si ritrova ai tempi dei Faraoni, all'interno di una grandiosa piramide, dove i molteplici piani sono formati da labirinti con innumerevoli sorprese.

Per superare ogni fase, dovrete dare la caccia alle varie mummie e mostri che vi si annidano.

Comandi: spingendo la leva del joystick verso destra, sinistra o avanti esplorate il labirinto

– spingendo la leva in dietro si visualizza la piantina del labirinto

– premendo Fire si visualizzano le figure catturate

– per ripartire dopo la schermata del punteggio finale, premere la lettera S.



**BUON
DIVERTIMENTO**

impariamo a usare il computer 4

LA PROGRAMMAZIONE

Come potete leggere in **ULTIMISSIME**, oggi il LOGO è finalmente utilizzabile sul nostro C 64.

Il senso e lo spirito di questo linguaggio ci può far meglio comprendere le possibilità del nostro « amico da tavola ». Per questo anche nella **PROGRAMMAZIONE** parliamo di LOGO e dei suoi concetti per arrivare, la prossima puntata, ai diagrammi di flusso per « ragionare » in BASIC.

Il nostro lettore che sino ad oggi si è cimentato nella costruzione di piccoli programmi in basic, senz'altro si sarà accorto che questo linguaggio non è quanto di meglio per potersi divertire un po' con la grafica.

In effetti dobbiamo pensare che venti anni fa, quando venne inventato negli USA il basic, il videoterminale era un prodotto molto sofisticato, riservato all'uso di pochi tecnici di laboratorio mentre era molto più diffuso l'uso della telescrivente come ultima interfaccia verso l'utenza.

Chi ha avuto esperienza di queste macchine elettromeccaniche (da alcuni chiamate affettuosamente « trebbiatrici » per il caratteristico rumore che producevano durante il funzionamento) conosce molto bene la scarsità di caratteri che esse riuscivano a riprodurre sulla carta; in più con il basic ebbe inizio l'epoca dei terminali remoti. Il periodo cioè in cui parte degli utilizzatori di computer interagivano con esso a tempo parziale ed in luoghi lontani dal centro di elaborazione vero e proprio. Tutto

ciò limitava di fatto la possibilità per l'utente di attivare complessi programmi grafici: come quelli ad esempio che gestiscono gli sprites sul **COMMODORE 64**.

In questi anni però anche il basic ha subito una notevole evoluzione: l'avvento dei micro-computers ha via via rese possibili delle variazioni del linguaggio aprendo le porte ad applicazioni sempre più sofisticate ed, in un certo senso, universali. Oggi si può usare il basic per costruire programmi gestionali, grafici, educazionali, ed in alcuni casi esso può divenire un valido strumento di lavoro per il programmatore e l'utente finale.

Comunque permangono in esso le fondamentali caratteristiche del basic originario di venti anni fa che gli impediscono di soddisfare con la necessaria immediatezza i desideri più elementari di chi, per la prima volta si avvicina al computer.

Immaginiamo di essere una persona che da poche ore ha iniziato il proprio dialogo con il computer.

Può succedere che costui voglia mettersi a disegnare alcune figure geometriche elementari: ovviamente si rende conto che non potrà in questo caso utilizzare la riga e la matita ma dovrà istruire l'elaboratore a tracciare delle linee e stabilire degli angoli sino alla loro completa costruzione.

Già questo primo passo lo mette in difficoltà poichè si vede costretto a seguire una via logica differente dalla usuale, vieppiù la difficoltà aumenta quando egli deve scrivere un programma in basic che esegua ad esempio la figura **QUADRATO**.

In BASIC non si possono tendere le linee: esse vengono realizzate mettendo in fila il carattere « - » in orizzontale ed il carattere « I » in verticale,

```

10 REM PROGRAMMA "QUADRATO"
20 PRINT "  _____  "
25 PRINT "I          I"
30 PRINT "I          I"
35 PRINT "I          I"
40 PRINT "I_____I"
50 END

```

con un risultato né bello né quadrato. Qualche miglioramento lo si può ottenere nella composizione di grafici: impostando alcune piccole routines come questa:

```

10 REM ROUTINE GRAFICA 1
20 FOR X=1 TO 20
30 PRINT TAB(X*X/10); "*"
40 NEXT X
50 END

```

oppure

```

10 REM ROUTINE GRAFICA 2
20 FOR X=1 TO 20
30 PRINT "X:"; TAB(ABS(3*X-31)+3); "*"
40 PRINT "Y:"; TAB(X+3); "#"
50 NEXT X
60 END

```

Prendendo spunto da queste due routines, se volessimo realizzare dei piccoli istogrammi orizzontali, potremmo utilizzare l'algoritmo che piazza su di una riga tanti caratteri giusto in proporzione alla grandezza di un numero (X) variabile a piacere:

```

100 PRINT " INSERISCI NUMERO
"; INPUT X
110 PRINT A; TAB(5); X;
120 FOR K=1 TO X/10
130 PRINT "#";
140 LET A=A+1 : NEXT K

```

in questo caso A rappresenta la scala di incremento dell'asse verticale.

Tutta questa prassi è troppo cervellotica per chi vuole semplicemente tentare un primo approccio con l'elaboratore; in più tutto ciò manca di umanità: non c'è dialogo con l'elaboratore se non a livello di comunicazioni di errore, peraltro non esaltanti.

L'ideale sarebbe di poter istruire non un impalpabile linguaggio, piuttosto un simpatico personaggio a cui poter dire: «ora fai così: vai dritto per 10 passi, poi giri a sinistra di quarantacinque gradi e ti fermi», trovando poi disegnate sullo schermo tutte le azioni del nostro interlocutore.

Si tratterebbe in poche parole di saltare da una situazione di mera attività programmatoria ad un diverso ambiente: molto più umano, semplice e divertente. Questo ambiente esiste e si chiama LOGO.

LOGO è stato realizzato dal Mit statunitense dopo anni di ricerche svolte al fine di realizzare un dispositivo software che rendesse davvero semplice ed amichevole il colloquio tra utente ed elaboratore personale.

LOGO è particolarmente orientato all'uso dell'elaboratore da parte dei bambini: risulta comunque evidente che questa non è una vera e propria discriminazione verso il pubblico degli adulti quanto l'espressione della filosofia che ha mosso i ricercatori americani nel proprio lavoro. D'altronde abbiamo visto poc'anzi che le difficoltà per disegnare un quadrato in basic sono relative più alle caratteristiche del linguaggio che all'età dell'utente.

LOGO permette di impostare il tipo di colloquio che prima avevamo ipotizzato: difatti per disegnare un quadrato basta istruire una simpatica (ed immaginaria) tartaruga affinché essa si muova sullo schermo in modo da costruire la figura desiderata, impostando una piccola serie di comandi la cui semplicità risulta evidente:

```

CHIAMIAMO LA TARTARUGA SULLO
SCHERMO
LE INDICHIAMO LA DIREZIONE IN CUI
ESSA DEVE PROCEDERE
DI QUANTO DEVE AVANZARE
DI QUANTI GRADI (90) DEVE POI
SVOLTARE A DESTRA O A SINISTRA
LE INDICHIAMO QUINDI CHE DOVRÀ
RIPETERE 4 VOLTE LE PRECEDENTI
ISTRUZIONI

```

alla fine la tartaruga avrà disegnato sullo schermo un quadrato. Potremo allora chiamare tutto ciò semplicemente «QUADRATO» e successivamente richiamare la procedura così definita senza stare ogni qualvolta a ripetere l'intero insieme di istruzioni.

Volendo realizzare una scacchiera non faremo altro che chiedere alla tartaruga di posizionare l'uno accanto all'altro tanti quadratini quanti ne servono a coprire l'intero piano.

All'interno di LOGO vi sono comandi semplici e di facile uso; non si deve però cadere nell'errore di pensare che LOGO sia banale; tutt'altro. Destra, sinistra, avanti e indietro, procedi con o senza scrivere, cambia il colore delle righe o di fondo sono le istruzioni, non dimentichiamolo, che utilizziamo normalmente nel nostro linguaggio naturale nella vita di ogni giorno.

Ma partendo proprio dalla esperienza quotidiana ci rendiamo conto di quanti errori, di quante incomprensioni si generano ogniqualvolta non esprimiamo i concetti e/o le azioni da eseguire con la necessaria precisione formale.

Anche LOGO soggiace a questa ferrea disciplina: proviamo a invertire l'ordine di alcuni comandi alla tartaruga. Essa non disegnerà più il tanto sospirato quadrato, bensì se ne andrà inutilmente a spasso per lo schermo disegnando scarabocchi.

È assolutamente necessario che noi per primi si abbia chiaro in mente lo sviluppo delle azioni da far compiere: si tratta in altre parole di scomporre un lavoro complesso, un piccolo insieme di azioni elementari che la tartaruga riconosce e sa compiere.

Ma siamo noi abituati a razionalizzare così bene i nostri pensieri? Non sempre.

In nostro aiuto giunge un modello grafico per descrivere le azioni complesse: il **diagramma di flusso**. Nelle prossime puntate vedremo come utilizzare questo modello per impostare le istruzioni alla tartaruga, ma non solo, in quanto il diagramma di flusso è uno strumento indispensabile per la corretta programmazione del computer in qualsiasi linguaggio, basic compreso.

impariamo a programmare in assembler

3

commodore 64 - commodore 64 - commodore 64 - commodore 64 - commodore

Per rappresentare i numeri con il sistema BCD si codifica separatamente ogni digit decimale e si

usano tutti i bit necessari per rappresentare esattamente e in modo completo il numero desidera-

POKE

rivista di informatica e video-games

**RIVISTA DI VIDEOGAMES
PER
48 K SINCLAIR e CBM 64
È IN EDICOLA
IL 1° DI OGNI MESE
NON PERDETELA!**

**7 VIDEO-GAMES per CBM 64
7 VIDEO-GAMES per SPECTRUM 48 K**

to. Per esempio, nel caso in cui il numero sia quello espresso nel sistema decimale con 90, in BCD sarà necessario codificare il 9 e lo 0, per cui — dovendo rappresentare i numeri da 0, a 9 — saranno necessari quattro bit ($9_{(10)} = 1001_{(2)}$) cioè un nibble. Ne consegue che il numero indicato nel sistema decimale con 90 sarà rappresentato in BCD con 10010000 dove, evidentemente, si è co-

dificato 9 in 1001 e 0 in 0000.

Utilizzando un nibble per ogni numero da codificare, appare subito chiaro che si dovrà, in seguito, affrontare il problema delle addizioni e delle sottrazioni in quanto, con 4 bit, si possono rappresentare 16 numeri mentre noi ne utilizzeremo solo 10:

CODICE	SIMBOLO BCD	CODICE	SIMBOLO BCD
0000	0	1000	8
0001	1	1001	9
0010	2	1010	non utilizzato
0011	3	1011	”
0100	4	1100	”
0101	5	1101	”
0110	6	1110	”
0111	7	1111	”

Con un byte si possono rappresentare due digit BCD: questo sistema viene denominato BCD impaccato.

Tipicamente, all'inizio della rappresentazione, si utilizzeranno uno o più nibble per indicare il numero totale dei digit BCD utilizzati. Un altro byte o nibble sarà necessario per indicare la posizione della virgola, quindi seguiranno i nibble dei digit BCD.

Per chiarirvi quanto sopra, poniamo, per esempio, di dover rappresentare con il sistema BCD multibyte il numero che nel sistema decimale è indicato con +123. Come prima cosa, usiamo un nibble per indicare che questo numero è formato da tre digit — 0011 — quindi utilizziamo un nibble per indicare il segno + positivo — 0000 — (se il segno fosse — negativo utilizzeremo 0001) ed infine faremo seguire i nibble che codificano rispettivamente $1_{(10)} = 0001_{(2)}$; $2_{(10)} = 0010_{(2)}$ e $3_{(10)} = 0011$. Per concludere questo esempio, il numero interno +123 sarà codificato in BCD come: 00110000000100100011; poiché questo numero è formato da cinque nibble ed in nostro microprocessore è un 8 bit, il nostro numero in BCD risulterà composto da 3 byte e cioè:

000000110000000100100011

Ora poniamo di rappresentare un numero decimale, per esempio +1,23:

1° nibble per indicare il numero dei digit: 0011

2° nibble per indicare che la virgola sta a sinistra

del secondo digit: 0010

3° nibble per il segno: 0000

4° nibble per il numero 1: 0001

5° nibble per il numero 2: 0010

6° nibble per il numero 3: 0011

Il numero +123 diventa in BCD:

001100100000000100100011.

In questa sede non approfondiremo ulteriormente il codice BCD, in quanto il suo utilizzo è strettamente legato a problemi esclusivamente di carattere contabile; ci limiteremo ad osservare che il sistema BCD presenta l'enorme vantaggio di farci pervenire a risultati esatti in modo assoluto, ma l'altrettanto enorme svantaggio di utilizzare una quantità notevole di memoria.

Vorrei concludere dicendo che quella qui esposta è solo una delle tante convenzioni possibili e che mi riservo di rispondere personalmente a coloro che mi scriveranno perché particolarmente interessati all'argomento.

Abbiamo già constatato che il codice BCD è in grado di rappresentare anche numeri decimali; ma per la ragione appena considerata, proviamo a rappresentare numeri decimali usando un formato a lunghezza fissa. Allo scopo di usare un formato fisso e nel contempo di non sprecare bit, è stata ideata la rappresentazione A VIRGOLA MOBILE che avrà come principio di base la "normalizzazione" di tutti i numeri.

Poniamo ad esempio di dover rappresentare il nu-

mero che nel sistema decimale è indicato con 0,000123: utilizzeremo tre zeri — che si possono definire inutili in quanto il loro uso è quello di indicare la posizione del punto decimale — e normalizzeremo il numero come $0,123 \times 10^{-3}$ dove 0,123 è la MANTISSA NORMALIZZATA e -3 è l'esponente (in pratica, il numero 0,000123 è stato normalizzato eliminando tutti gli zeri non significativi alla sua sinistra ed aggiungendo l'esponente).

Consideriamo ora un ulteriore esempio in cui il numero sia: 88,1. La sua normalizzazione porge $0,881 \times 10^{+2}$, cioè, generalizzando quanto visto finora, possiamo dire che un numero viene normalizzato secondo la formula:

$M \times 10^E$ dove M è la mantissa ed E è l'esponente, con M maggiore o uguale a 0,1 e minore di 1:

$$0,1 \leq M < 1 \text{ oppure } 10^{-1} \leq M < 10^0$$

Analogamente nella rappresentazione binaria:

$$2^{-1} \leq M < 2^0 \text{ (leggibile anche } 0,5 \leq M < 1)$$

dove M è IL VALORE ASSOLUTO DELLA MANTISSA (trascurando il segno).

Per esempio: 634,01 è normalizzato come: $0,63401 \times 2^3$. La mantissa è 63401 e l'esponente è 3.

Esaurito il principio della rappresentazione, esaminiamo ora il formato effettivo.

La rappresentazione tipica A VIRGOLA MOBILE è la seguente:

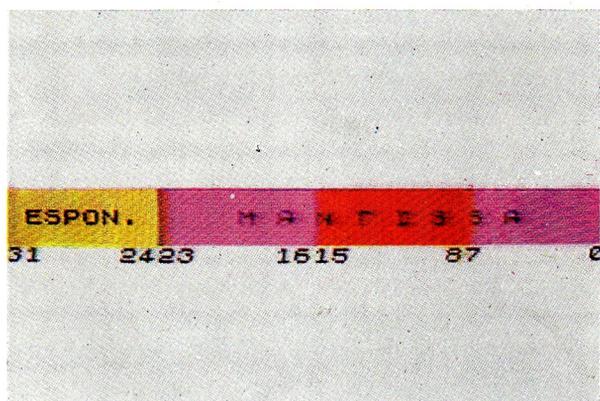


Fig. 1

In figura 1 è rappresentato un esempio in cui vengono utilizzati quattro byte per un totale di 32 bit. Il primo byte di sinistra è utilizzato per rappresentare l'esponente, i rimanenti tre byte sono

occupati dalla rappresentazione della mantissa. Sia la mantissa che l'esponente saranno indicati mediante la rappresentazione in complemento a due (vedi numero precedente).

Il massimo esponente rappresentabile sarà: $127 - 2^{127}$ — e il minore — $128 - 2^{128}$ —. Tre byte sono utilizzati dalla mantissa dove, poiché il primo byte della rappresentazione in complemento a due indica il segno, avremo 23 bit per la rappresentazione della grandezza della mantissa.

Diciamo subito che è possibile utilizzare tre byte oppure più byte per rappresentare un numero a virgola mobile, ma che la rappresentazione a quattro byte è senza dubbio il compromesso più ragionevole tra precisione, grandezza dei numeri, utilizzazione della memoria ed efficienza nelle operazioni aritmetiche.

Così, con la trattazione della rappresentazione a virgola mobile, abbiamo esaurito i problemi associati alla rappresentazione di numeri e passiamo ad affrontare la rappresentazione interna di dati alfanumerici.

La rappresentazione dei DATI ALFANUMERICI — cioè di caratteri — è completamente diretta in quanto tutti i caratteri sono codificati in un codice di 8 bit.

I codici generalmente utilizzati sono due: il Codice ASCII ed il Codice EBCDIC. Chiariamo subito che il Codice EBCDIC è una variante dell'ASCII utilizzato dalla IBM, quindi non ci riguarda direttamente, poiché in questa sede noi trattiamo del microprocessore 6510 montato sul COMMODORE 64.

L'ASCII è universalmente utilizzato nella parola dei microprocessori.

Se esaminiamo la codifica ASCII ci rendiamo subito conto che tutti i caratteri sono codificati in 7 bit; infatti, 7 bit consentono 128 codici possibili e noi abbiamo 26 lettere dell'alfabeto maiuscole, 26 lettere minuscole, 10 simboli numerici e circa 20 simboli addizionali speciali, per cui 7 bit sono largamente sufficienti.

L'ottavo bit, quando è utilizzato, è detto BIT DI PARITÀ.

Quella della parità è una tecnica che verifica l'accidentale cambiamento nei contenuti di una parola. Esiste una PARITÀ PARI e una PARITÀ DISPARI. Farò due esempi per chiarire quanto sopra: si calcoli quale dovrà essere il bit di parità

STREPITOSO È IN EDICOLA

TUTTO CBM 64

TUTTO CBM 64

TUTTO CBM 64

8 video games



di "0010011" impiegando la parità pari. Se noi calcoliamo il totale di "1" presenti, vediamo che sono tre, cioè in quantità dispari; a questo punto il bit di parità pari dovrà necessariamente essere un 1, in modo tale da ottenere un totale di "1" pari, cioè quattro. Avremo quindi la rappresentazione 10010011 dove l'uno di sinistra è il bit di parità ed il numero 001011 identifica il carattere.

Ovviamente nel caso in cui si operi con PARITÀ DISPARI — considerando sempre l'esempio di partenza — si scriverà 0 al posto dell'1 nell'ottavo bit per ottenere una rappresentazione valida (cioè con il numero totale di "1" dispari).

A questo punto abbiamo esaurito quella che è stata definita la rappresentazione interna e cioè il modo di rappresentare il programma e i dati all'interno del calcolatore.

Ora esamineremo le possibili rappresentazioni esterne.

Per rappresentazione esterna si intende il modo in cui l'informazione è presentata all'utente, generalmente al programmatore. Esistono essenzialmente tre formati possibili:

- 1) BINARIO
- 2) OTTALE ED ESADECIMALE
- 3) RAPPRESENTAZIONE SIMBOLICA.

1) BINARIO

Talvolta sarebbe desiderabile mostrare l'informazione interna direttamente nel suo formato binario e questo avviene in modo diretto solo su alcune macchine fornite di un pannello recante (nel caso di microprocessore a 8 bit) 8 LED dove, a LED acceso, si ha un "1", mentre a LED spento uno 0.

2) OTTALE ED ESADECIMALE

L'ottale e l'esadecimale codificano rispettivamente tre e quattro bit binari in un unico simbolo.

Nel sistema OTTALE qualsiasi numero rappresentato nel sistema binario con tre bit viene codificato da un unico simbolo compreso da 0 a 7. Per esempio il numero binario 00100100 sarà codificato in ottale in questo modo:

00 100 100 e diverrà 044 in ottale.

↓ ↓ ↓
0 4 4

Oppure il numero 321 in ottale:

3 2 1
↓ ↓ ↓
011 010 001

rappresenta il numero 11010001 in binario.

Ecco di seguito la tabella dei simboli ottali:

BINARIO	OTTALE
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

La rappresentazione ottale veniva applicata dai vecchi computer che utilizzavano internamente numeri di bit varianti da 8 a 64 circa.

Oggi, con il dominio di microprocessori a 8 bit, si utilizza una rappresentazione decisamente più pratica, quella ESADECIMALE.

Nella rappresentazione esadecimale, si codificano quattro bit come digit esadecimale. I simboli esadecimali sono i numeri da 0 a 9 seguiti dalle lettere A, B, C, D, E ed F..

TABELLA CONVERSIONE
DECIMALE - BINARIO - ESADECIMALE

DECIMALE	BINARIO	ESADECIMALE
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Consideriamo ora il seguente esempio:

Il numero in sistema binario 11000010 sarà così rappresentato:

$$\begin{array}{cc} 1100 & 0010 \\ \downarrow & \downarrow \\ C & 2 \end{array}$$

e diventerà C2 in esadecimale.

Invece, il numero indicato in esadecimale con B1 sarà:

$$\begin{array}{cc} B & 1 \\ 1011 & 0001 \end{array}$$

cioè 10110001 in binario.

La rappresentazione esadecimale offre il vantaggio di codificare i bit in soli due digit, quindi risulta più facile da visualizzare e da memorizzare ed è più veloce da rappresentare.

Ogni volta, però, che l'informazione nella memoria ha un significato — cioè rappresenta un testo o dei numeri — allora la rappresentazione esa-

decimale diventa scomoda.

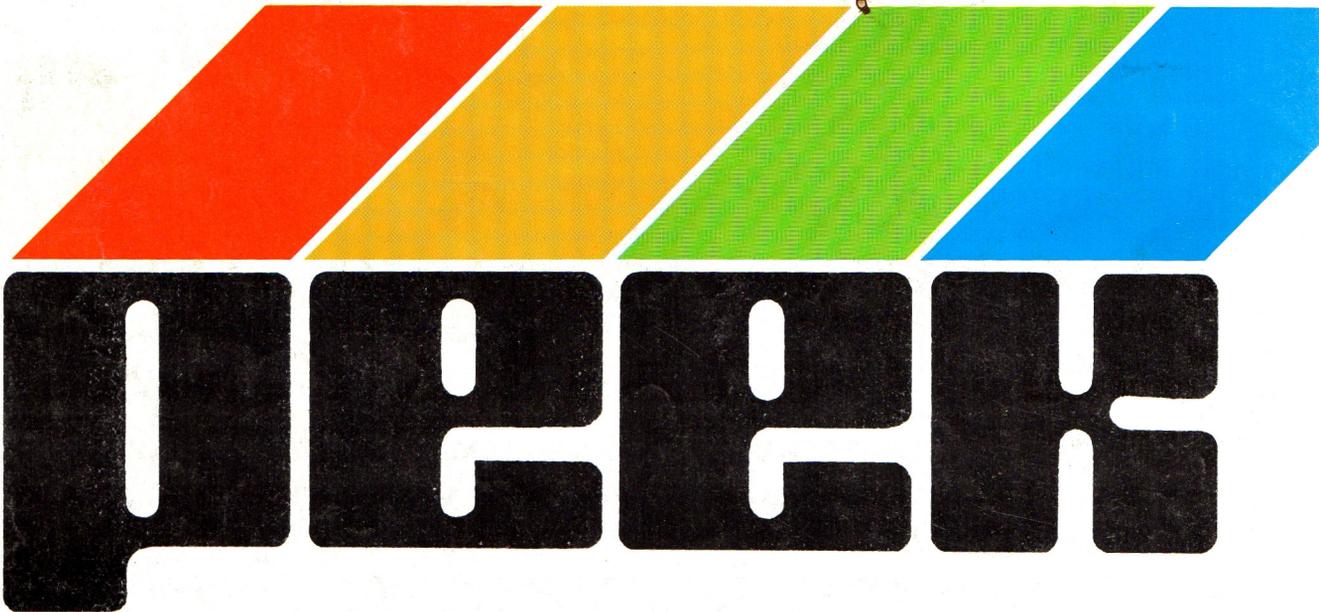
In questi casi è meglio ricorrere alla RAPPRESENTAZIONE SIMBOLICA.

3) LA RAPPRESENTAZIONE SIMBOLICA.

La rappresentazione simbolica dell'informazione è la più naturale per l'utente umano. Normalmente essa si serve di una tastiera alfanumerica per l'input e di un display CRT o di una stampante per l'output.

Nella rappresentazione simbolica i numeri decimali sono rappresentati nel sistema decimale, le lettere come lettere, il testo come testo e non come insieme di codici esadecimale o — peggio ancora — di codici binari.

Siamo finalmente giunti alla fine della trattazione delle rappresentazioni interne ed esterne possibili. Nel prossimo articolo esamineremo il microprocessore effettivo che manipolerà l'informazione. Arrivederci al prossimo mese!



7 VIDEO-GAMES PER VIC 20

7 VIDEO-GAMES PER CBM 64

IL BEL PAESE

UN GIOCO AL MESE

anno 1 n. 1 UN

anno 1 n. 2 UN

anno 1 n. 3

dicembre 1985 6.000

26 15 91 46

8 23 44 60

TOMBOLA
dell' Anno

Edizione Farnoni s.r.l. - Milano
DISTRIBUZIONE: Messaggerie Periodici
64 D. Corneo, 32 - 20141 Milano
Pubblicazione registrata al Tribunale di Milano
il 17/9/85 n. 407

Spedizione in abbonamento postale - Gruppo 0170

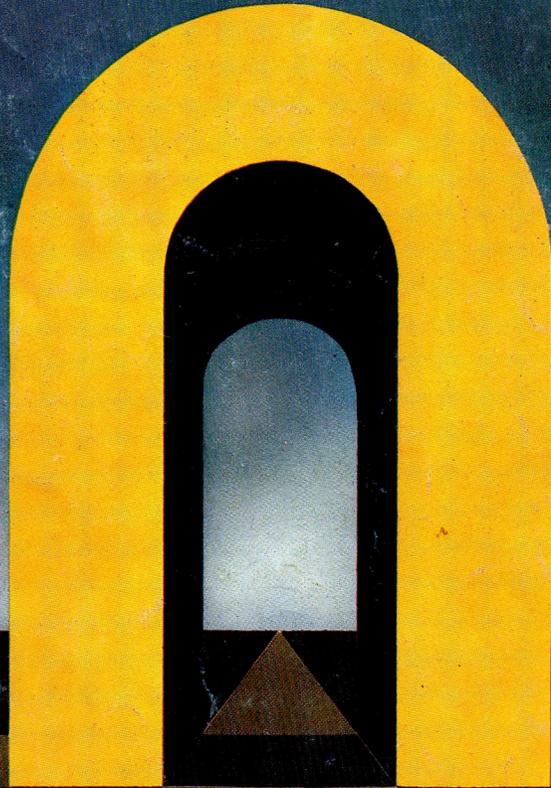
IL BEL PAESE, un gioco al mese, è un instant game, un gioco da tavolo per adulti basato sulla simulazione di avvenimenti di attualità, sulla presenza di personaggi e situazioni legate alla realtà, sullo sviluppo di meccanismi che cercano, né più né meno, di riprodurre le regole del gioco della vita. Ogni mese IL BEL PAESE sarà in edicola con una proposta attuale e sempre diversa, per far divertire tutti i suoi lettori.



STREPITOSO È IN EDICOLA

DIALETTI

ADVENTURES GAMES AND NEWS



COMMODORE 64