



a cura di Raffaello De Masi

True Basic

Nel 1963 un gruppo di ricercatori, capeggiati da John G. Kenemy e Thomas E. Kurtz decise di affrontare in maniera estesa ed articolata il problema di rendere facile agli studenti la programmazione. Questa è cosa nota, ed è altrettanto noto che da questa decisione nacque poi il Basic. Meno nota, invece, è la cronistoria che portò alla definitiva redazione del primo linguaggio.

La decisione di sviluppare un linguaggio estremamente semplice da usare, nuovo, e del tutto avulso dalle esigenze della macchina, non era una idea né nuova né originale; lo stesso Fortran, a rigore, è linguaggio dalla elevata evoluzione e poco legato a manipolazione diretta delle risorse interne della macchina.

D'altro canto si era giunti ad un periodo della evoluzione delle macchine tale che il mondo informatico era in notevole fermento, alla ricerca di nuove idee, grazie anche alle notevoli possibilità date dalle macchine della seconda e terza generazione e dalla nascita di un componente, l'integrato (che poi si evolverà nel microprocessore), che lasciava intravedere nuove e più potenti applicazioni del «cervello elettronico» come, sfortunatamente, ebbe a definirlo Von Neumann.

Il gruppo di ricercatori del Dartmouth College non si applicò, all'inizio, alla produzione di un linguaggio vero e proprio; il principio che animò la ricerca fu quello di individuare una nuova strada per l'uso dei computer. La cosa più interessante fu che al gruppo di ricerca parteciparono numerosi studenti, particolarmente abili nel campo dell'analisi; il primo parto di questo gruppo fu il primo vero sistema di time-sharing completamente funzionale, anche se, forse un poco pomposamente, gli stessi componenti del gruppo definirono la loro opera come il primo vero esponente dell'informatica interattiva.

Parallelamente a questo progetto,

che, per la verità, non ebbe gran fortuna, i due progettaronò un nuovo linguaggio: questo nuovo nato, forse un po' immodestamente, si proponeva vari ed ambiziosi fini: essere facile da usare, essere avulso da profonde conoscenze della componentistica della macchina, ed essere utile a diversi scopi. Le premesse erano, come dicevamo, alquanto ambiziose, visto che ben altri linguaggi, ormai ben consolidati nell'uso (parliamo di Fortran e Cobol) erano divenuti di uso comune. I risultati di tali progetti sono fin troppo noti; Basic è il linguaggio di gran lunga più usato nel mondo, e se è vero quanto ricordo di aver letto da qualche parte, che esistono circa 12000 titoli di libri sull'argomento, non c'è bisogno di ulteriori commenti.

Poco dopo il 1970 due innovazioni principali, nel campo della potenzialità delle macchine e nei concetti principali di programmazione, diedero una svolta rapida all'architettura dei più diffusi linguaggi: faceva timidamente capolino la prima grafica, e quasi contemporaneamente, cominciarono a farsi largo i principi della programmazione strutturata. Alla prima esigenza il Basic si adattò immediatamente, dotandosi di istruzioni più sofisticate ed efficienti; alla seconda fece, nei primi tempi, una certa resistenza, da una par-

te per certe sue caratteristiche intrinseche, dall'altra probabilmente per un malinteso senso della semplicità.

Il tempo ha ulteriormente giocato a favore di un prodotto che, non siamo noi a scoprirlo, aveva tutti i numeri per figurare elegantemente a fianco degli idiomi mostri sacri della programmazione. Negli ultimi anni, grazie a linguaggi come lo ZBASIC della Zedcor o il True Basic che stiamo esaminando, questo idioma è divenuto veramente universale ed intercambiabile, visto che di essi sono state sviluppate versioni specifiche per singole macchine. Il True Basic, che trattiamo in questa puntata, è una implementazione del Basic che, sebbene del tutto conforme allo standard ANSI, è moderno e flessibile essendo dotato dei più perfezionati mezzi propri della strutturazione, pur restando del tutto aderente alla sua originale filosofia: facilità di acquisizione anche da parte di utenti del tutto ignari delle più comuni tecniche di programmazione, interattività estrema, ma con notevole riduzione dei disagi che questa soluzione pur porta, utilizzo semplice e sempre trasparente, potenza elevata tanto da renderlo adatto a qualsivoglia applicazione. Vediamo se è vero.

Il True Basic

Caratteristiche generali del linguaggio.

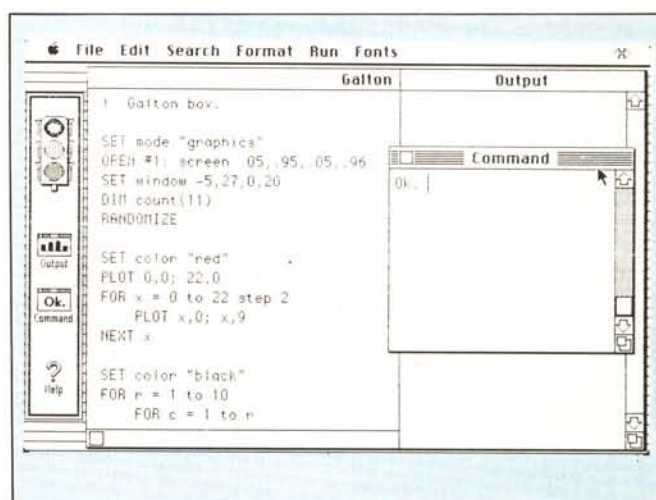
Al contrario di quanto oggi abbastanza in voga nei «salotti» del Basic, True rimane ancora fondamentalmente un linguaggio interpretato: non possiede, cioè un compilatore del tipo «built-in», anche se ammette una soluzione salomonica di «compilazione dell'ultima ora», trasformando il programma in una forma pseudocompilata. La mano dei creatori del Basic, che hanno redatto questo linguaggio si sente potente, come, ad esempio, nella presenza del LET, che qui è obbligatorio. D'altro canto onore al merito di



chi ha creato un'epoca e che desidera difendere, a ragione, qualcosa che gli è sfuggita di mano, ma che è pur sua in tutti i sensi.

Ma procediamo con ordine. Al lancio del linguaggio, a sinistra appare la finestra di redazione del programma, che si presenta simile alle window cui ci aveva abituato il MSBasic, ma, al contrario di questo, purtroppo, non è ridimensionabile. A destra compaiono le finestre di output e di command, anche esse non ridimensionabili, anche esse dalla funzione piuttosto simile a quella del Basic Microsoft, tranne che per la presenza di un «OK» in odore di Forth che sostituisce il prompt dopo l'esecuzione del comando immediato. Ogni programma soggiace a certe piccole regole di redazione, che certi nuovi idiomi ci avevano fatto dimenticare: è necessario che in un listato, ad esempio, ci sia un END (che deve essere unico); accanto all'END è invece possibile sistemare tanti STOP quanti ne si desiderano. Ancora molto standard sono le regole per la definizione delle variabili e dei loro nomi; il nome non può superare la lunghezza di 32 caratteri, ma può contenere anche caratteri speciali e, cosa molto pratica, il linguaggio non tenta, a tutti i costi, di decifrare nei nomi parole riservate (cosa che ha creato non pochi problemi agli utenti di vecchie versioni di altre realizzazioni). Ancora, e questo in contrasto con le regole ANSI, ma in ossequio con le direttive iniziali imposte al Dartmouth, il linguaggio non fa distinzione tra caratteri maiuscoli e minuscoli, anche se poi provvede a convertire in maiuscolo tutte le keyword. Ma accanto a queste caratteristiche molto tradizionali (ormai il LET è considerato opzionale, se non superfluo, da tutti i costruttori) si vede subito un substrato moderno e potente, efficiente e facile da utilizzare. La versione per Macintosh, che qui proviamo, ha un editor efficiente e rapido, del tutto simile al Mac Write, e come questo, di rapido apprendimento; la linea di programma può essere lunga fino a 32000 caratteri (è il doppio della lunghezza di questo articolo; ma a che cosa serve?). In ossequio alle più recenti regole (ed anche questo inversamente a quanto previsto dall'ANSI) i numeri di riga sono facoltativi (anche se esiste una opzione che consente, con molta facilità, di rinumerare immediatamente il testo del programma secondo le nostre esigenze). Le stringhe hanno una lunghezza massima di 32000 caratteri, mentre (nello spirito

Caricamento di un programma e visualizzazione delle finestre di output e command, tutte ridimensionabili.



della facilità d'uso del Basic) le variabili numeriche sono assunte sempre nella massima precisione possibile (precisione che varia da macchina a macchina). Gli operatori logici sono solo 3 (AND, OR, e NOT), ma la possibilità di creare procedure personalizzate consente di superare facilmente questa limitazione.

Accanto a tali operatori esistono vere e proprie funzioni logiche del tipo: **KEY INPUT** (del tutto analogo all'INKEYS)

END DATA, molto utile in quanto evidenzia l'esaurimento di valori in una serie READ-DATA

MORE DATA, al contrario del precedente, restituisce [True] se esistono, ancora, nel corpo del programma DATA da leggere, e viceversa [False] in caso contrario.

END #(espr) [True] se il puntatore al file nominato è alla fine del file assegnato ad (espr), [False] se è il contrario.

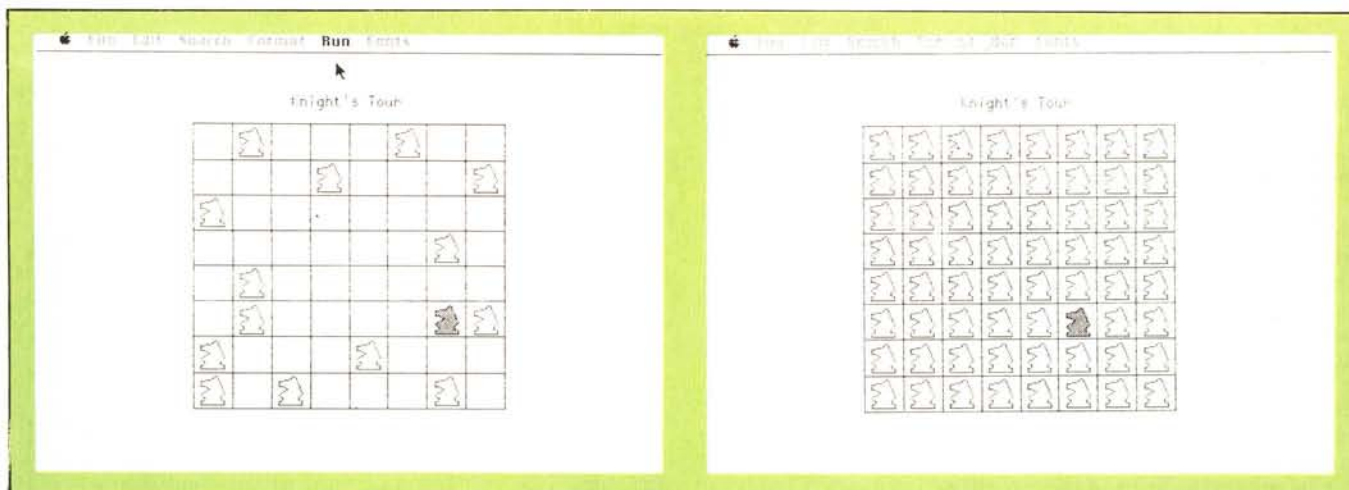
MORE #(espr) l'inverso del precedente.

Funzioni matematiche e di stringa

True Basic supporta le più richieste funzioni matematiche: ABS (valore assoluto dell'argomento), DIVIDE (che consente di eseguire la divisione conservando immediatamente risultato e resto), EPS (il valore più piccolo generabile dalla macchina), EXP, INT, LOG, LOG(2) (logaritmo in base 2), LOG(10), MAX, MAXNUM (il numero più grande generabile dalla macchina), MIN, MOD, REMAINDER (resto della divisione), RND, ROUND (arrotondamento), ROUND(A,B) (numero A arrotondato a B cifre decimali), SGN, SQR e TRUNCATE. Allo

stesso modo è molto ricco di funzioni trigonometriche; ma prima di passare a descriverle occorre fare una precisazione.

Agli albori della microinformatica di massa (sto parlando di una decina di anni fa, mica tanto! Erano i tempi delle TI 56 e 57, degli Apple II con il loro Integer Basic prima versione, si parlava di programmazione sintetica sulle piccole calcolatrici della HP, MC non esisteva ancora, e Marinacci era un buono di cuore, sempre pronto a dire una parola gentile al prossimo), tutte le macchine, grandi e piccole, avevano la possibilità di scegliere il tipo di misura degli angoli preferiti. Molte macchine ammettevano anche quattro opzioni diverse: DEG (gradi da 0 a 360), RAD (radianti, da 0 a 2π), GRAD (gradi quattrocentesimali, da 0 a 400) e BRAD (da 0 a 256) e, una volta settate in tale misura, la conservavano fino a nuova opzione. I linguaggi ammettevano uno statement, anzi per essere precisi una dichiarazione, che consentiva di lavorare nella unità di misura desiderata. Molto comoda per chi era, come la maggior parte di noi, abituato a «pensare» in gradi, la possibilità è andata, col passare del tempo, nella maggior parte dei casi perduta e sono rimasti, come unità di misura i soli radianti. Ora se questo ha notevole valore formale e scientifico, impone d'altro canto all'utente non smaliziato un piccolo sforzo di adattamento ad una serie di piccole, ma talora fastidiose trasformazioni che magari potevano essere evitate. Dei Basic più recenti solo il True Basic conserva, ancora, la possibilità di lavorare in DEG, tramite una opzione caratteristica, da dichiarare prima dell'uso di una



```

! Knight's tour. Needs more than 128k.
SET window -3,13,0,11
DIM move(8,2) ! Legal moves
DIM board(-1 to 10,-1 to 10) ! Board status

CALL initdraw ! Board & knight
CALL initial ! Tables
RANDOMIZE

LET x, y = 1 ! Initial position
FOR t = 2 to 64
  CALL nextmove(x,y,newx,newy) ! Find new position
  BOX SHOW k$ at x*.1,y*.1 ! Shadow knight to old
  BOX SHOW k$ at newx*.1,newy*.1 ! Knight to new
  PRASE .2 ! Otherwise too fast
  LET x = newx ! Now the position
  LET y = newy
NEXT t

SUB initial ! Set up tables
  RMT READ move ! Legal (x,y) changes
  DATA 2,1,2,-1,1,2,1,-2,-1,2,-1,-2,-2,1,-2,-1
  FOR x = -1 to 10 ! Mark positions off board
    FOR y = -1 to 0
      LET board(x,y), board(y,x) = -1
    NEXT y
    FOR y = 9 to 10
      LET board(x,y), board(y,x) = -1
    NEXT y
  NEXT x
  FOR x = 1 to 8 ! Moves possible from here
    FOR y = 1 to 8
      LET count = 0
      FOR a = 1 to 8 ! Try 8 legal possibilities
        LET x1 = x+move(a,1)
        LET y1 = y+move(a,2)
        IF board(x1,y1)>=0 then LET count = count + 1 ! On board
      NEXT a
      LET board(x,y) = count ! Possible moves
    NEXT y
  NEXT x
END SUB

SUB nextmove(x,y,x2,y2) ! Find hardest to get to
  DIM opt(8,2) ! Optimal moves
  LET board(x,y) = -1 ! Can't move there anymore
  LET bmin = 9 ! For minium calculation
  FOR a = 1 to 8 ! 8 possibles
    LET x1 = x+move(a,1)
    LET y1 = y+move(a,2)
    LET b = board(x1,y1) - 1 ! Now one less possible move
    LET board(x1,y1) = b
    IF b <= 0 and b<bmin then ! Possible move, new low
      LET bmin = b
      LET new = 1
      LET opt(new,1) = x1
      LET opt(new,2) = y1
    ELSEIF b=0 and b = bmin then ! Tie
      LET new = new + 1
      LET opt(new,1) = x1
      LET opt(new,2) = y1
    END IF
  NEXT a
  LET na = int(new*rad+1) ! Pick at random
  LET x2 = opt(na,1)
  LET y2 = opt(na,2)
END SUB

SUB initdraw ! Draw board and knight
  PLOT TEXT, at 3,5,9,5: "Knight's Tour"
  FOR x = 1 to 9 ! Board
    PLOT x,1,x,9
    PLOT 1,x,9,x
  NEXT x
  SET color "black" ! Outline of knight
  DRAW knight with shift(1,1)
  BOX KEEP 1,1,1,9,1,1,1,9 in k$ ! Save for show
  SET color "red" ! Red knight
  FLOOD 1,5,1,5
  BOX KEEP 1,1,1,9,1,1,1,9 in k$ ! Save it also
END SUB

PICTURE knight ! Picture of knight
  PLOT .2,1;.8,1;.8,2;
  PLOT .7,25;.7,3;.8,4;.65,7;.6,9;.55,9;
  PLOT .5,82;.2,75;.2,6;.3,6;.4,55;
  PLOT .25,45;.2,37;.3,3;.3,25;.2,2;.2,1
END PICTURE

```

Risoluzione e listato del problema di Knight (è possibile e, nel caso, quale è il numero minimo di mosse necessarie perché un cavallo copra tutte le caselle di una scacchiera, senza mai passare sullo stesso punto?).

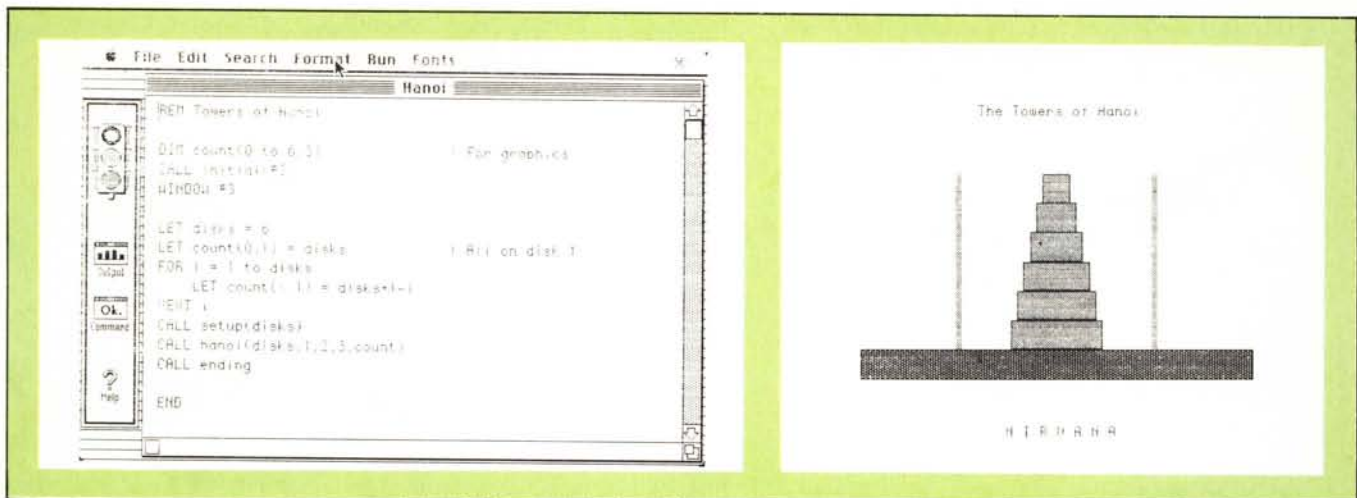
qualunque funzione coincidente gli angoli: Option Angle Degree. Non è una possibilità peregrina, visto che il Basic, per sua filosofia, impone l'adattamento del linguaggio alle esigenze, e perché no, alle abitudini dell'utente e non viceversa.

Come dicevamo True Basic suppor-

ta le più diffuse ed utili funzioni trigonometriche; una ci ha colpito particolarmente, per essere un po' inusuale in un linguaggio Basic: ANGLE(p1,p2) restituisce l'angolo compreso tra l'asse delle X ed il punto p1,p2.

Anche la manipolazione delle stringhe è abbastanza standard; anche qui,

comunque non mancano certe raffinatezze, come LTRIMS (argomento) che «taglia» tutti gli spazi bianchi superflui in una frase, fornita come argomento; ancora, esiste una funzione non molto diffusa, LCASE\$, che esegue il contrario della più universale UCASE\$.



Procedura ricorsiva utilizzata, manco a dirlo, per la risoluzione del problema delle torre di Hanoi; si noti, nel listato, la chiamata a librerie esterne; il problema, per una torre con 6 dischi viene risolto, compresa l'animazione grafica ed il caricamento delle librerie, in 38 secondi.

Loop e strutture di controllo

Anche in questo settore, accanto al massimo della standardizzazione richiesta dalla dichiarazione di completa aderenza all'Ansi, True Basic non manca di numerose particolarità interessanti. Il classico costrutto FOR prevede, come opzione; l'EXIT FOR che va usato nella forma:

```
IF evento EXIT FOR
```

che consente una «escape» anche in caso di loop non completato. Ancora, la struttura DO è qui costruita in modo particolare, ma molto potente: esistono infatti quattro possibilità diverse di loop:

```
DO
----
----
----
LOOP
-----
DO WHILE condizione
----
----
----
LOOP
-----
DO
----
----
----
LOOP UNTIL condizione
-----
DO WHILE condizione
----
----
----
LOOP WHILE condizione
```

che, per giunta, ammettono anche l'opzione EXIT DO, del tutto simile all'EXIT FOR. Si tratta di strutture molto potenti e ben studiate, che non lasciano scoperto alcun caso di ciclo.

Altrettanto tradizionale per un Basic è il costrutto IF, anche se, in questo caso, altri linguaggi sono riusciti a far bene altrettanto. Accanto al classico IF monolinea, con o senza l'ELSE, si affianca l'IF multilinea, che richiede come partitore finale END IF. Ma non basta; è ammessa la struttura di IF ad albero, con il costrutto ELSEIF, anch'esso multilinea, anche se in questo caso, può essere presa in considerazione la possibilità di usare il costrutto SELECT CASE, del tutto simile al CASE OF del Pascal.

Strutture di I/O

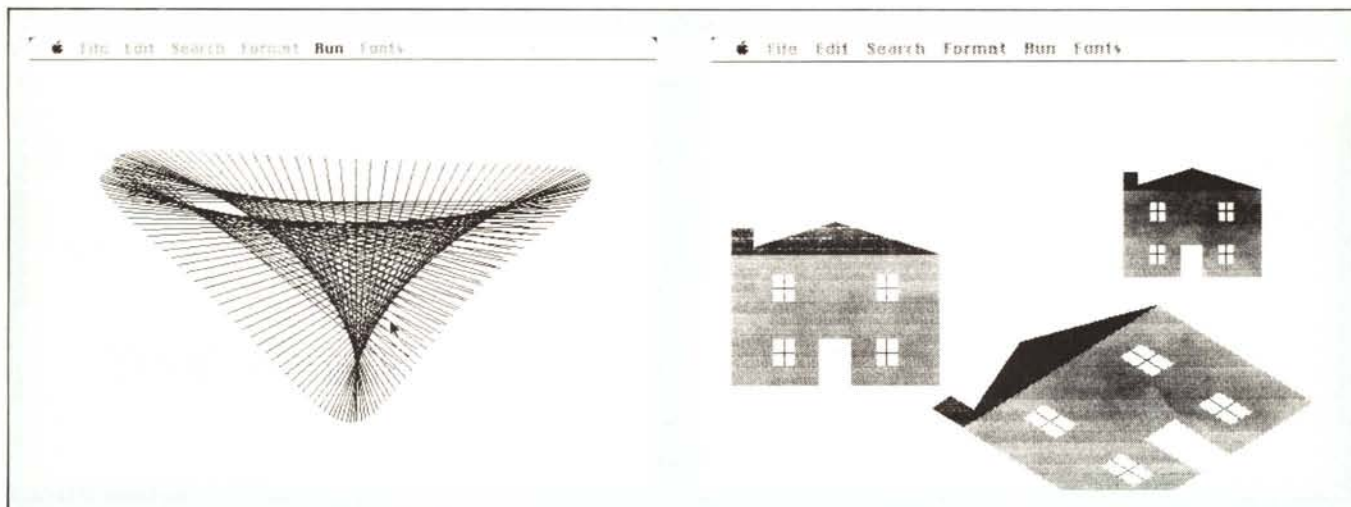
In questo caso le novità e l'originalità del True Basic sono molto meno evidenti: sono supportate tutte le forme più comuni di input e di stampa, compreso un Print Using abbastanza raffinato specie nella manipolazione di numeri. Anche nell'input compaiono i più comuni costrutti, come input multipli, lettura di singoli tasti con un comando GET KEY, più efficiente del solito INKEYS. Il cursore può essere posizionato sullo schermo in maniera abbastanza simile al Locate della Microsoft, mediante il comando SET CURSOR, cui fa ad riscontro un ASK CURSOR che restituisce in coordinate di carattere la locazione del cursore. È ancora possibile «spegnere» il cursore con il comando SET CURSOR.

READ, DATA e RESTORE vengono trattati nel modo classico, senza particolari caratteristiche. Ma sempre a proposito di strutture di input occorre segnalare certe particolarità di grande utilità, che raramente si trovano in linguaggi più efficienti e «potenti».

True Basic possiede numerose utility per manipolare array e matrici; è

possibile ridimensionare nel corso del programma una matrice lasciando libera la memoria superflua; è consentita inoltre l'allocazione dinamica delle matrici, vale a dire che le dimensioni delle matrici possono essere definite tramite una variabile. Due istruzioni particolari, MAT PRINT e MAT PRINT USING, consentono tramite un solo comando, di stampare tutto il contenuto di una matrice, eventualmente (USING) utilizzando una particolare formattazione. Allo stesso modo MAT INPUT consente, tramite un solo comando, di «saturare» una matrice tramite una sola riga di input (i valori vengono immessi su una sola riga, e viene in ogni caso eseguito un controllo per verificare che siano «coperti» tutti posti disponibili). Ancora più specializzati sono le istruzioni MAT LINE INPUT e MAT READ, il primo di ovvio significato, il secondo capace di leggere valori da strutture DATA e di riempire, con essi una matrice (l'allocazione di valori viene fatta da sinistra a destra e dall'alto in basso). È possibile addirittura assegnare una matrice ad un'altra matrice, ed eseguire operazioni aritmetiche su matrici tramite un solo ordine, così come è possibile moltiplicare una matrice per un vettore. Pensate che sia finita? Neppure per sogno!! È possibile eseguire l'inversione e la trasposizione di matrice, oltre ad altre numerose (più di una ventina) funzioni, come calcolo del determinante di una matrice quadrata, prodotto e rapporto di vettori, calcolo del valore minimo e massimo, ecc.

Sempre a proposito di input/output, la manipolazione dei file viene eseguita in maniera abbastanza simile a quella Microsoft, che poi non è altro



Un po' di grafica non guasta, anche se l'output su carta non fa giustizia della spettacolarità e della velocità d'esecuzione.

che quella, ancora una volta, adottata dall'ANSI. True Basic prevede, comunque, oltre a file sequenziali e random (che vengono qui chiamati come file di testo e file di record), anche una terza forma, i byte file, che manipolano una qualunque forma di dato; il linguaggio legge poi questi dati come semplici sequenze di byte. La cosa è utile quando si debbano leggere file creati da sistemi diversi o si desidera accedere direttamente alla struttura interna di un file, come, ad esempio un programma applicativo.

Sempre a proposito di I/O True Basic consente un discreto controllo del-

lo speaker della macchina: il tutto è regolato da tre istruzioni, piuttosto efficienti, PLAY, SOUND e PAUSE, il primo dei quali, soprattutto, possiede una notevole estensione di opzioni, che ne fanno un tool molto raffinato, se non nel campo della musica, in quello degli effetti speciali.

Funzioni e subroutine

True Basic offre, oltre alle strutture precedentemente viste, due tool strutturati fondamentali, le subroutine e le funzioni. In questo non si differenzia

molto da altri Basic, ma la somiglianza è solo apparente; oltre la facciata, queste due strutture si rivelano molto più potenti di quello che effettivamente sono.

Una funzione, per True Basic è un blocco di istruzioni che può accettare ogni numero di parametri, e restituisce un singolo valore numerico o di stringa. Una subroutine, invece, è un blocco di istruzioni che accetta qualsiasi numero di parametri, e che non restituisce valori.

Ambedue, funzioni e subroutine, possono essere interne ed esterne. Nel primo caso esse sono fisicamente pre-

True Basic ed il Toolbox

Il toolbox di Mac è stato da sempre croce e delizia dei programmatori in Basic su Mac. Il Microsoft Basic, nella sua versione 1 non ne prevedeva affatto l'accesso e anche nella versione 3 è possibile utilizzare solo parzialmente le routine presenti nelle viscere delle Ram. La colpa sta nel linguaggio in sé, che non possiede strutture, come tipi, puntatori, ecc., capaci di passare parametri alle utility di Rom. Recentemente, comunque, anche i Basic hanno avvicinato questo mondo sotterraneo, ed in questo anche il True non sfigura. Siamo ancora distanti dalla potenza dello ZBasic, che consente un accesso virtualmente illimitato al toolbox, ma questo pacchetto ci ha riservato molte piacevoli sorprese.

True accede a gran parte delle risorse interne della macchina; le elenchiamo di seguito: Variabili globali. Routine di Graphport, Manipolazione del cursore, Grafica di linee ed aree (ivi compresi

ovali, rettangoli, rettangoli smussi, poligoni generici), Grafica di testo, Colore (inteso, però come di back e foreground), e, più in generale, grafica manipolabile attraverso passaggio di variabili. Restano ancora precluse le vie della manipolazione dei record e dei type, dominio del Pascal (anche se ZBasic, l'enfant terrible di Dortmund, ci arriva in altro modo). Mancano altresì le utility di manipolazione di menu e di button, caratteristiche del mondo Mac.

Accanto al True Basic di base (ci si perdoni il gioco di parole), la casa offre una lunga serie di librerie a supporto, molto ben organizzate, che aggiungono alla già elevata versatilità e potenza del linguaggio quella marcia in più che non guasta. Le librerie sono presenti in tutte le versioni dei linguaggi, anche se, ovviamente, possono esserci differenze tra le diverse implementazioni. Ne indichiamo qualche esempio:

- **PC Basic Converter:** consente, in ambiente MS-DOS, di tradurre programmi, redatti in GWBasic e Basica, in linguaggio True. Particolarmente efficiente, il pacchetto si rivela intelligente, eseguendo anche una parziale interpretazione del listato e tentando, ove possibile, una strutturazione del nuovo programma.

- **Communication Support:** consente di connettere due porte COM per il trasferimento di dati fino a 19200 baud.

- **Form Management Library:** r=libreria di routine precostituite per la manipolazione di operazione di I/O (molto interessante una estesa gamma di PRINT USING).

- **Hercules Graphic Support,** per la più avanzata gestione di questa scheda.

- **Sorting and Searching:** libreria di routine finalizzate.

- **Advances String Library:** idem, finalizzato alla manipolazione di stringhe.

- **3-D Graphics,** finalizzate, tra l'altro, alle proiezioni prospettiche e parallele.

senti, nel programma, prima dell'END, nel secondo caso lo seguono. La differenza non è solo formale. Le unità esterne possono essere fisicamente non presenti nel corpo del programma, ma risiedere sul disco come librerie esterne; si tratta di vere e proprie unità di programmazione, indipendenti dal resto del programma, anche se ad esso afferenti, e, come vedremo, sono di grande aiuto nel lavoro del programmatore.

Le funzioni, almeno quelle interne, non sono molto dissimili dalle DEF FN del classico Basic. Alle subroutine si accede, invece, tramite una chiamata, CALL, che fornisce, anche, alla subroutine stessa le variabili. Nei più vecchi Basic tutte le variabili erano considerate come globali, vale a dire che intervenivano in qualunque parte del programma modificando acconciamente il loro valore; nelle funzioni e subroutine di True Basic le variabili possono essere condivise (shared) col programma principale ma possono altresì essere locali, vale a dire che è possibile il loro uso e definizione nel solo ambito della definizione o del subprogramma. Fin qui ancora niente di diverso, tranne forse qualche perfezionamento nell'uso delle variabili. Ma True Basic consente diverse cose in più: un programma principale consente di agganciarsi ad un altro programma, in una forma molto simile al CHAIN, ma senza le complessità della definizione delle variabili comuni, proprie di altri linguaggi. Ma la cosa più interessante è la possibilità di costruirsi file di libreria. Un programmatore attento e scrupoloso potrà prepararsi una serie di subroutine o funzioni utili, per poi richiamarsele al momento più opportuno. Si tratta dello stesso principio che ha animato il «C» e che, alla fin fine ne ha fatto la sua fortuna. Le librerie sono rappresentate da textfile presenti sul media che vengono chiamate tramite l'istruzione

LIBRARY «nome _ del _ file»

Al lancio del programma principale, il file viene caricato e quanto in esso contenuto viene trasformato in funzioni o subroutine esterne, in tutto e per tutto simili a quelle presenti. Ovvio il vantaggio per il programmatore di poter disporre, all'inizio del lavoro, di un sacco di tool precostituiti, da non dover ribattere alla tastiera ogni volta.

Le subroutine e le funzioni sono, udite udite, utilizzabili, in True Basic ricorsivamente; la ricorsione è ammessa anche nelle sue forme più complesse, alla faccia della debolezza del Basic e di chi ne parla male!

Grafica

Chi scrive, fin da quando ha lasciato il suo vecchio HP 87 per passare a nuovi mezzi, ha sempre rimpianto la grafica della sua vecchia macchina; non nascondo che mi sono avvicinato a questo True Basic desideroso di controllare quanto la pubblicità prometteva circa una grafica superiore alla concorrenza; devo dire che pur non essendo ancora lontano dalla grafica del Basic ANSI Hewlett-Packard, True Basic possiede una marcia un più rispetto agli standard più diffusi, primo tra tutti il Microsoft. Tra le altre cose è possibile definire il campo di disegno «clippando» una certa area, che è possibile ridefinire in coordinate a piacere; oltre i soliti RATIO, PLOT, BOX, ecc. esistono comandi più raffinati, come PLOT AREA, che riempie una regione col colore corrente, PLOT TEXT, che stampa caratteri tenendo conto delle coordinate predefinite, MAT PLOT, che consente di plottare famiglie di punti conservati in una matrice, GET POINT che interroga l'utente sulla posizione di un punto, PICTURE, che consente di creare disegni predefiniti e di richiamarli alla bisogna, WINDOW, che consente di utilizzare finestre diverse sullo schermo, ecc.

Non siamo ancora al top, altro occorrerà fare per raggiungere davvero una elevata potenza grafica, ma siamo sulla buona strada.

Deskscene

Custom Desktop Installer
di Bruce Horn

V. 1.0 - 1986

PBI Software

Data la lunghezza dell'articolo su True Basic, solo qualche riga per presentare un programmino piccolo piccolo che consente di personalizzare la scrivania. Il programma non ha alcuna funzione pratica, ma



Conclusioni

True Basic si presenta con tutte le carte in regola per affrontare i più complessi problemi programmatori senza alcun complesso d'inferiorità nei confronti degli altri linguaggi.

Questo linguaggio è strutturato, potente, veloce, poco ingombrante, efficace e dotato di opzioni (come tutta la manipolazione delle matrici) addirittura assenti nei prodotti della migliore concorrenza.

Un lungo capitolo del manuale è dedicato alla compatibilità con altri Basic, cosa sempre possibile (anche se non sempre è possibile il contrario).

Il manuale che viene dato in dotazione è chiaro e preciso (non per niente è stato scritto dai creatori del Basic stesso) ed è conforme allo standard ANSI, in modo da non creare problemi di portabilità.

Non male per un linguaggio la cui unica pretesa iniziale era quella di avviare l'utente, che non aveva mai toccato una tastiera, allo studio della programmazione.

True Basic è tutto questo ed altro, costa poco, e può essere dotato di un compilatore, da acquistare a parte, efficiente e rapido, anche se un po' costoso, che doterà i programmi di quella velocità che è stata sempre rimproverata al Basic da santoni di altri linguaggi, che non riescono poi a vedere i difetti ben più gravi delle loro creature.

consente solo di sostituire la scrivania con un disegno Mac Paint di nostra fattura e scelta. L'installazione avviene in maniera pulita e rapida, anche se l'operazione non è proprio sparpagnina in fatto di memoria. L'installazione, inoltre, può essere temporanea, vale a dire solo per la seduta in cui viene eseguita, o permanente, in quest'ultimo caso ulteriormente finalizzabile tramite l'assegnazione di un suo buffer temporaneo di locazione, che consente operazioni più veloci sul programma principale (che risulta appena rallentato dalla presenza di uno sfondo così complesso da sottoporre a restore continuamente).

Abbiamo utilizzato per la prova un disegno di una mela (manco a dirlo), ottenuto con un digitizzatore, ed i risultati sono quelli che vedete. Probabilmente a causa delle dimensioni del disegno, in termini di K, il programma si è rifiutato di eseguire una memorizzazione temporanea.

Altra utilizzazione, se non quella di puro diletto, non abbiamo trovato per questo programmino.

MC

LA COMPUTER CENTER PRESENTA LE NOVITÀ HARDWARE DELL'AUTUNNO '87 PER GLI AMIGA 500, 1000, E 2000

AMEGADRIVE - Hard Disk e Drive controller con orologio in tempo reale per l'Amiga 500, 1000 e 2000 in standard SCSI. Utilizza drive con capacità formattata da 20 a 800 Mb, l'adozione del protocollo SCSI ne fanno il più moderno e veloce Hard Disk del mercato. Il controller, autoconfigurante si inserisce sul bus laterale del computer ed è munito di connettore passante per permettere il contemporaneo inserimento a cascata di altre periferiche come espansioni di memoria, ecc. Possibilità di effettuare sul drive fino a 4 partizioni a capacità variabile sulla scheda controller è presente un integrato con batteria tampone, che assicura il mantenimento dell'orario anche a computer spento o anche a scheda momentaneamente disinserita dal computer. Altre periferiche SCSI come HARD DISK, WORM, CD-ROM, FOTOSCANNER, ecc.

AMEGA DRIVE 20 Mb CD-ROM, FOSCANNER, ECC.

AMEGA DRIVE 20 Mb
PER ALTRE CAPACITÀ CHIEDERE

L. 1.290.000

A-2090 Hard disk controller ST-506/SCSI e drive ST-506 da 20 Mb per l'amiga 2000 - si inserisce all'interno del computer

L. 1.090.000

HD 2000 Hard Disk da 20 Mb e controller in standard ST-506 per l'amiga 2000 - necessita della scheda Janus - si può partizionare sia in ambiente MS-DOS che in amiga DOS.

PER ALTRE CAPACITÀ CHIEDERE

L. 590.000

HD 2000 CARD come il modello HD 2000, ma montato su scheda. Non occupa pertanto lo scot che rimane libero per un drive da 3"5 o 5"1/4 L. 690.000
PER ALTRE CAPACITÀ CHIEDERE

I DRIVE

A DRIVE drive esteso da 3"5 - per qualsiasi modello amiga. Completo di connettore passante per collegare fino a 4 unità. Costruito completamente in tecnologia TTL-CMOS non necessita di alimentazione esterna - il cabinet metallico di colore light beige amiga assicura il più alto grado di affidabilità contro disturbi R.F. Meccanica interna firmata NEC per garantire la massima velocità di trasferimento dati (256K bit/sel) e di silenziosità di funzionamento. Dimensioni 10 x 3 x 21 cm

L. 270.000

A DRIVE TOWER Compatto sistema comprendente 3 drive esterni per qualsiasi modello amiga - specialmente indicato per le software house in quanto permette anche di effettuare 3 copie contemporanee di dischetti. Dimensioni 10 x 10 x 21

L. 590.000

AD 2000 drive aggiuntivo interno per l'amiga 2000, con frontale colore grigio amiga 2000. Completo di dettagliate istruzioni di montaggio

L. 219.000

ESPANSIONI DI MEMORIA

DRIVEADAPTOR permette di collegare all'amiga qualsiasi drive esterno da 3"1/2 - 1Mb o 5"1/4 - 360Kb Shugart compatibili 500, 1000 e 2000 L. 89.000

- AMEGA BOARD - Espansione di memoria di 8 Mb per Amiga si collega nel collettore di espansione laterale del computer, duplicando per permettere il contemporaneo inserimento di altre periferiche. Autoconfigurabile non necessita di software per l'installazione. La versione base viene fornita con 2 Mb di memoria e 3 slot liberi per inserire le AMEGA CARD da 2 Mb ciascuna. **AMIGA BOARD L. 790.000 - AMEGA CARD da 2 Mb L. 460.000 - AMINTERAM** Espansione di memoria da 512 Kb per l'Amiga 500 si inserisce nell'apposito scomparto del computer. Munita di orologio in tempo reale con batteria tampone per il mantenimento dell'ora anche a computer spento. Comprende una serie di potenti utilities e vanno da una RAM disk speciale (non si cancella durante Reset o Guru) ha una "cache" per velocizzare i floppy disk L. 179.000 **KIC KROM** kit di montaggio per AMIGA 1000 che consente di installare il KICKSTART direttamente su ROM e di incrementare la memoria RAM di 256Kb. Completo di dettagliate istruzioni di montaggio. L. 99.000 **RAMINT** Kit di montaggio per AMIGA 1000 che consente di incrementare la memoria interna di 512kb. Completo di dettagliate istruzioni di montaggio L. 169.000 **KITSERVICE** Installazione di uno dei 2 Kit sopra descritti effettuato direttamente nei nostri laboratori L. 150.000 con il più potente MC-68010 L. 59.000

A MEM Kit di espansione da 512 Kb per le prime versioni amiga 2000 L. 75.000

A-2052 espansione di memoria da 2 Mb per l'amiga 2000 - si inserisce all'interno nell'apposito slot. L. 590.000

I PROCESSORI E COPROCESSORI

BANG 10 Kit di sostituzione della Cpu originale MC - 6800 con il più potente MC 68010 L. 49.000

BANG 1081 Kit di sostituzione della CPU con una scheda comprendente l'MC-68010 e coprocessore matematico e 80 bit 68881 con clock a 12 MHz L. 499.000

BANG 2081 Kit di sostituzione della CPU MC-68.000 con una scheda comprendente il full 32 bit 68020 e coprocessore 68881 con clock a 12 MHz L. 899.000

Per versioni con clock a 16 e 20 MHz chiedere

Attenzione gli articoli della serie bang trasformano gli amiga 500, 1000 e 2000 nei più veloci computer del mercato. I premi sono stati tenuti volutamente bassi in modo da permetterne la massima diffusione. Sono di semplice installazione e non richiedono alcuna saldatura o dissaldatura di componenti e sono pertanto alla portata di tutti. A tutti gli acquirenti verrà inviata tutta la documentazione Motorola. Siamo inoltre interessati ad entrare in contatto con esperti software ed hardware con conoscenza linguaggio macchina, e, famiglia 68000, protocollo SCSI, sistemi operativi amiga DOS, UNIX OS-9.

I COMPUTER

... se ti hanno convinto a voltare pagina fallo alla grande con....

I PROFESSIONALS

Amiga 500 completo di mouse e manuali espand. 1 Mb con orologio e batteria tampone (aminteram) L. 990.000

Amiga 500 + Aminteram per monitor a colori Commodore 1081/1084 o Philips 8833 L. 1.549.000

Amiga 2000 con monitor 1081/1084 o 8833 + 2Mb RAM A-2052 + Hard Disk da 20 Mb e controller A-2090 L. 3.990.000

Amiga 2000 con monitor 1081/1084 o 8833 + 2 Mb RAM A-2052 + Janus XT con drive da 5"1/4 + 2 Drive interno da 3"5 AD 2000 + hard disk da 20 Mb e controller HD 2000 card L. 4.990.000

Attenzione i computer, le periferiche Commodore e i prodotti Computer Center sono muniti di garanzia ufficiale Commodore Italia e Computer Center valida un anno dalla data di acquisto.

La direzione della Computer Center augura a tutti un felice periodo natalizio e coglie l'occasione per ringraziare tutti coloro che ci hanno aiutato a crescere professionalmente collaborando o preferendo i nostri prodotti.

VENDITA SOLO PER CORRISPONDENZA

**COMPUTER
CENTER**

ATTENZIONE I PREZZI SOVRAESPONDI SONO IVA COMPRESA

PER INFORMAZIONI E/O ORDINAZIONI:

VIA FORZE ARMATE, 260 20152 MILANO - TEL. 02-4890213