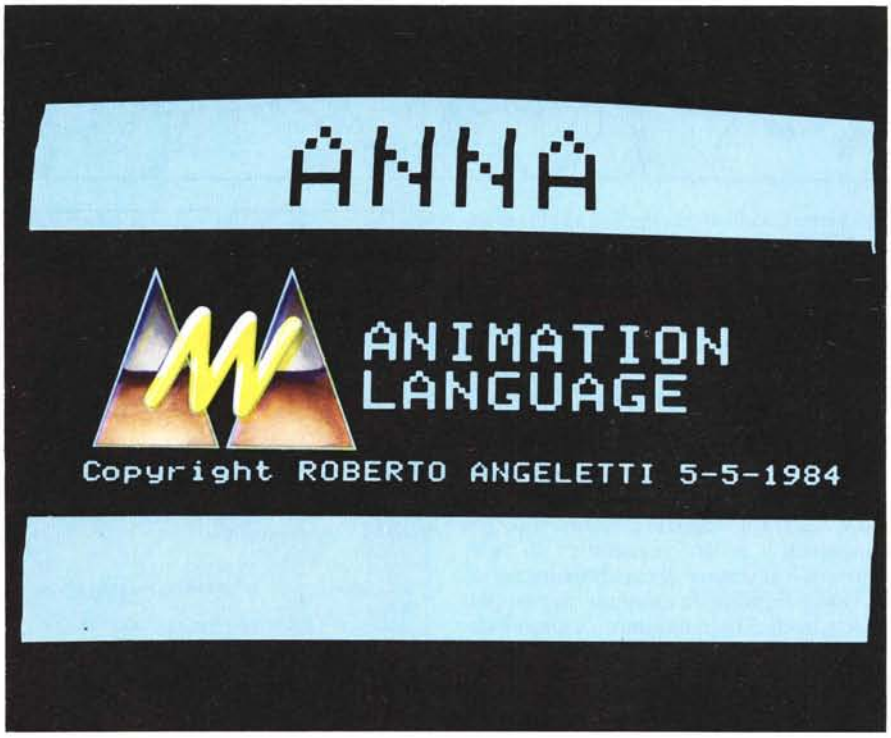


ANNA Animation Language

Per Apple II e MC-Tablet

di Roberto Angeletti

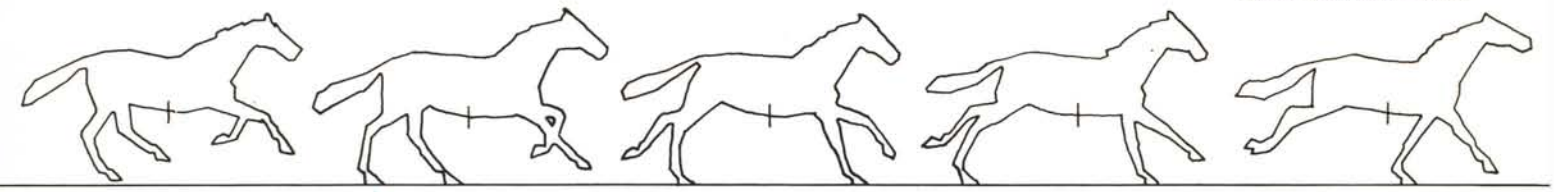
Quello dell'animazione è senz'altro uno dei problemi più stimolanti nel campo della computer graphic. Più o meno tutti siamo rimasti affascinati dalle sequenze dinamiche che hanno fatto da fiore all'occhiello di molti film dell'ultima generazione, quella elettronica; altre sequenze ci vengono mostrate ormai a getto continuo dalle reti televisive. Tuttavia, l'uso che oggi si fa delle sequenze computer generated è ancora restrittivo. Per fare un paragone storico possiamo ricordare che quando cominciò, agli inizi del secolo, a diffondersi la tecnica cinematografica, essa veniva usata per impressionare gli spettatori con trucchi stucchevoli che non avevano altro fine se non quello di spingerne la diffusione. Soltanto successivamente il cinema è maturato fino a diventare il vero e proprio mezzo espressivo che conosciamo. Per la computer art siamo, per così dire, ancora in quella fase post-pionieristica più o meno lunga che ha lo scopo di saturare ed abituare prima di diventare cultura. In questa fase di "diffusione alle masse" e di riduzione dei costi, così come è possibile farsi un film in casa, con il videotape, è divenuto possibile creare degli effetti dinamici con computer "casalinghi".

Tra i sistemi a basso costo è indispensabile effettuare una scelta secondo una caratteristica essenziale che è quella dell'esistenza o meno di due separati picture buffer, o più semplicemente pagine grafiche. Infatti è noto che, per avere una simulazione dinamica efficiente, ogni "quadro" (o fotogramma) deve essere percepito nella sua completezza e non può avvenire "allo scoperto" nessuna operazione di creazione o cancellazione di immagine, pena un ben percepibile sfarfallìo che può risultare fastidioso e che, comunque, scopre il gioco. Con due pagine grafiche si disegna in una mentre si mostra l'altra e, quando è finita l'operazione, si scambiano istantaneamente e si ripete l'operazione per una nuova immagine. In questo modo la fluidità della sequenza non è condizionata troppo dal mezzo che si usa.

Ognuna di queste immagini avrà un tempo "fisico" di costruzione, al di sotto del quale è impossibile andare; ad esempio, nel simulare la rotazione di un oggetto sul proprio asse, il tempo di calcolo della vista prospettica cresce con la complessità dell'oggetto ed è di gran lunga superiore a quello necessario alla visualizzazione vera e propria; dunque, usando sistemi a basso costo è impensabile ottenere in tempo reale o per lo meno in un tempo accettabile la serie delle viste necessarie, specie se queste sono piuttosto elaborate. Sempre per quanto riguarda la prospettiva, operazioni che bruciano tempo sono quella della soluzione dell'intersezione col piano anteriore (operazione che deve avvenire quando una parte dell'oggetto è alle nostre spalle, e che, altrimenti, verrebbe rappresentata ugualmente, però ribaltata) e quella, tipica, delle hidden line, o linee nascoste per le quali a volte sono necessarie addirittura svariate decine di minuti. A questi due problemi più generali si aggiunge, per quanto riguarda l'Apple II, quello del troncamento dei segmenti parzialmente fuori dallo schermo, che va risolto dal programmatore, non essendo già previsto nel set di routine in ROM. Tenuto conto di tutto ciò, è chiaro che per ottenere i tempi brevi necessari alla dinamica si deve svincolare la fase di visualizzazione da quella di calcolo dell'immagine virtuale. Per fare ciò, l'insieme dei quadri deve essere messo da parte per poter essere richiamato sullo schermo in rapida successione. Ma in che modo, o meglio dove va immagazzinato? La prima soluzione che viene in mente è quella di registrare su disco l'intero contenuto delle pagine grafiche, ma questa, si scopre subito, non è una via efficiente e, quindi, praticabile; innanzitutto, perché occorrerebbe una quantità di memoria enorme (su un disco si possono memorizzare solo fino a 15 immagini), e poi perché il tempo di accesso di una pagina è troppo lungo. Scartata questa ipotesi, ce n'è un'altra: quella di costruire dei vettori in cui sono contenute le istruzioni per ottenere quel disegno. Questa è una buona soluzione, ma bisogna stare attenti all'impiego della memoria.

Facciamo un po' i conti, supponendo di voler memorizzare un segmento. Possiamo costruire due vettori distinti, uno per le coordinate X e l'altro per le Y di ciascun punto, oppure possiamo servirci di un vettore unico che contenga alternate le X e le Y in successione; questa seconda soluzione ci permette di risparmiare memoria. Un ulteriore notevole vantaggio si apporterà usando gli interi, che operano su due soli byte ad elemento, contro i 5 dei reali. Tirando le somme, un segmento avrà bisogno di 4 elementi, ovvero di 8 byte. Supponiamo, però, che nel disegno non siano contenuti solo segmenti, ma anche punti isolati e spezzate; potremo, allora, avere bisogno di un altro vettore, che ci indichi il tipo di elemento; oppure possiamo inserire quest'ulteriore informazione sempre nello stesso vettore unico. Ma eccoci ora ad una considerazione sostanziale: due byte ci consentono di memorizzare valori che vanno da -32768 a +32767 ed è, quindi, facile constatare che una gamma così vasta è inutile, dato che il campo di visualizzazione Apple va da 0 a 279 per la X e da 0 a 191 per Y; sono sufficienti, perciò, due soli byte, così come già avviene nella notazione interna esadecimale.

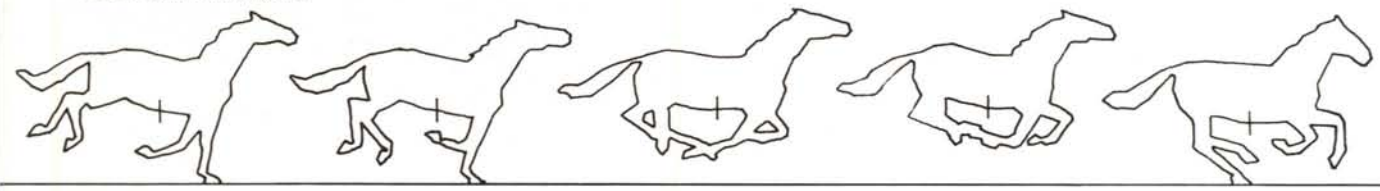
Facciamo qualche richiamo sulla rappresentazione binaria di un numero, dato che ciò è essenziale per comprendere quello che segue; senza entrare troppo nel dettaglio, ci basti sapere che un numero come



255 viene codificato come 11111111, mentre 280 corrisponde a 100011000; dato che un byte può contenere fino a 8 di questi elementi binari, un numero superiore a 255 avrà bisogno di almeno due byte. Per quanto riguarda la X, uno dei due byte necessari, quello del riporto (o Most Significant Byte) rimane, anche con il valore massimo, praticamente vuoto, usando solo un bit della sua rappresentazione binaria. Tutti gli altri bit possono essere a nostra disposizione per contenere qualunque altra cosa noi vogliamo, compreso, per l'appunto, il nostro segnalatore di tipo. Tornando al vettore di cui abbiamo parlato, esso è formato da elementi raggruppati in settori di 3 byte ognuno, l'ultimo indicante la coordinata Y, il primo ed il secondo quella X, nelle sue parti meno e più significanti. È proprio nel secondo byte, cioè quello di centro, che è contenuto il codice del "tipo" per quel determinato elemento; questo codice è, naturalmente, arbitrario e, quindi, segue una nostra convenzione. Il caso più semplice è quello di un punto isolato con coordinata X inferiore a 255; il bit più a destra è a zero, come, del resto, tutti gli altri. Con la X maggiore di 255, il bit va a 1, mentre gli altri restano a 0. Dato che, nel caso in cui vogliamo tracciare un segmento tra due punti, basta indicare che il secondo punto va collegato con il precedente, riusciremo in questo modo a tracciare anche delle spezzate senza ripetere ogni volta tutti e due gli estremi; per codificare tutto questo è sufficiente che nel byte, che chiameremo da ora in poi HX, cioè High X o parte alta di X, sia messo ad uno il secondo bit da destra; avremo così finora 4 possibilità, o status del byte: 00, punto isolato, minore di 255; 01, punto, maggiore di 255; 10, segmento, minore; 11, segmento, maggiore. Ciò significa che dovremo, per codificare, inserire con dei POKE i valori 0, 1, 2, 3 e che, viceversa, per decodificare, dovremo "sbirciare" nel byte con un PEEK ed eseguire i relativi HPLOT o HPLOT TO. Con lo stesso sistema possiamo continuare per altri tipi di elementi, e così facciamo per le shape e le corrispondenti istruzioni DRAW e XDRAW. Tutti i valori nei diversi casi sono indicati nella tabella:

bin	hex	dec	istruzione	
00000000	$0	0	HPLOT	X<255
00000001	$1	1	HPLOT	X>255
00000010	$2	2	HPLOT TO	X<255
00000011	$3	3	HPLOT TO	X>255
00000100	$4	4	DRAW	X<255
00000101	$5	5	DRAW	X>255
00000110	$6	6	XDRAW	X<255
00000111	$7	7	XDRAW	X>255

```
0  GOTO 2240
1 E% = 0:NF% = NF% + 1: POKE DT,NF%: RETURN
2 E% = E% + 1: POKE DT + NF%,E%:SD% = SD% + 1:H% = X% / 256:L% = X% - H% *
    256: POKE SD%,L%:SD% = SD% + 1: POKE SD%,H% + IS%:SD% = SD% + 1: POKE
    SD%,Y%: RETURN
4 SD% = SD% + 1: POKE SD%,RZ:SD% = SD% + 1: POKE SD%,SL:SD% = SD% + 1: POKE
    SD%,N: RETURN
5 :
6  REM          "ANNA" CODING ROUTINES
7 :
8 :
9 :
10  HPLOT X%,Y%: RETURN
20  HPLOT  TO X%,Y%: RETURN
40  GOSUB 100: DRAW N AT X%,Y%: RETURN
60  GOSUB 100: XDRAW N AT X%,Y%: RETURN
61 :
62 :
63 :
64  REM          DINAMIC SLIDE-SHOW
65 :
100 LC = LC + 1: ROT=  PEEK (LC):LC = LC + 1: SCALE=  PEEK (LC):LC = LC +
    1:N =  PEEK (LC): RETURN
180  HGR : POKE  - 16302,0: HCOLOR= 3:PG = 0
190 LC = DT + 100
200  FOR FT = 1 TO  PEEK (DT)
210 PG =  NOT PG: POKE 230,32: IF PG THEN  POKE 230,64
220  CALL 62450
230  FOR TF = 1 TO  PEEK (DT + FT)
240 LC = LC + 1:L =  PEEK (LC):LC = LC + 1:H =  PEEK (LC):SS% =  INT (H /
    2) <  > (H / 2):X% = L + SS% * 256:LC = LC + 1:Y% =  PEEK (LC)
250 H = H + 1: ON H GOSUB 10,10,20,20,40,40,60,60
260  NEXT
270  POKE  - 16300 + PG,0
280  NEXT
290  GET C$: IF C$ = "R" THEN 190
300  GOTO 1080
301 :
302 :
303 :
304  REM          LIMITS
305 :
510  IF X% < 0 THEN X% = 0
520  IF X% > 279 THEN X% = 279
530  IF Y% < 0 THEN Y% = 0
540  IF Y% > 190 THEN Y% = 190
550  RETURN
551 :
552 :
553 :
554  REM          LETTURA TAVOLETTA
555 :
560  POKE 779,100: CALL 768:P0 = (Z0 -  FN PK(12)) * PZ:S0 =  SIN (P0):C0
    =  COS (P0): POKE 779,101: CALL 768:P1 = ( FN PK(12) - Z1) * PY - P0
    :X% = 150 * ( COS (P1) - C0):Y% = 150 * (S0 +  SIN (P1))
570 X% = X% + 144:Y% = Y% - 63
580  RETURN
1001 :
1002 :
1003 :
1004  REM          MENU
1005 :
1080  TEXT
1090  HOME : PRINT W$; CHR$ (CH);"          S L I D E  -  C O D E R          "
    ; CHR$ (CH);: PRINT W$
1120  VTAB 4: PRINT "          A N N A   ANIMATION LANGUAGE    ": PRINT W$: VTAB
    8: HTAB 5: FOR E = 1 TO 32: PRINT  CHR$ (CH);: NEXT : PRINT : FOR E =
    1 TO 13: HTAB 5: PRINT  CHR$ (CH) SPC( 30) CHR$ (CH): NEXT : HTAB 5: FOR
    E = 1 TO 32: PRINT  CHR$ (CH);: NEXT : PRINT
1130  VTAB 8: HTAB 12: PRINT "MENU"
1140  PRINT : HTAB 8: PRINT "1) LOAD Shape da disco"
1150  PRINT : HTAB 8: PRINT "2) LOAD SLIDE DA DISCO"
1160  PRINT : HTAB 8: PRINT "3) SAVE SLIDE SU DISCO"
1170  PRINT : HTAB 8: PRINT "4) DEF. DELLA SLIDE"
1180  PRINT : HTAB 8: PRINT "5) SHOW DELLA SLIDE-TABLE"
1190  PRINT : HTAB 8: PRINT "6) Fine Programma"
1200  GET RI$:RI =  VAL (RI$): ON RI GOTO 1290,1210,1220,1370,180,2420: IF
    RI > 6 OR RI < 1 THEN 1090
1201 :
1202 :
1203 :
1204  REM          LOAD SLIDE-VECTOR
1205 :
1210  HOME : INVERSE : PRINT "          LOAD SLIDE-TABLE                ": NORMAL
    : PRINT : INPUT "NOME DEL FILE : ";NF$: PRINT : PRINT D$"BLOADSEQ-"NF
```



```
    $:SD% = ( PEEK (43616) +  PEEK (43617) * 256) - 1:SD% = SD% + DT:NF% =
     PEEK (DT): GOTO 1080
1211 :
1212 :
1213 :
1214  REM       SAVE SLIDE-VECTOR
1215 :
1220  HOME : INVERSE : PRINT "         SAVE SLIDE-TABLE              ": NORMAL
    : PRINT : INPUT "NOME DEL FILE : ";NF$
1230  ONERR  GOTO 1250
1240  PRINT  CHR$ (4);"VERIFYSEQ-"NF$: POKE 216,0: PRINT "IL FILE "NF$" G
    IA' ESISTE !!": PRINT "VUOI CANCELLARLO (S/N) ? ";: GET C$: IF C$ <  >
    "S" AND C$ <  > "Y" THEN 1220                                    :
1250  POKE 216,0
1260  PRINT
1270  PRINT  CHR$ (4)"BSAVESEQ-"NF$",A"DT",L"(SD% - DT) + 1
1280  GOTO 1080
1281 :
1282 :
1283 :
1284  REM       LOAD SHAPE-TABLE
1285 :
1290  HOME : INVERSE : PRINT "         LOAD SHAPE                    ": NORMAL
    : PRINT : INPUT "NOME DEL FILE : ";NF$
1300  PRINT  CHR$ (4)"BLOADSH-"NF$",A"AD
1310 ST =  PEEK (43616) +  PEEK (43617) * 256:NS =  PEEK (AD):S1% = 1
1320  GOTO 1080
1321 :
1322 :
1323 :
1324  REM       COMANDI
1325 :
1330  HOME : VTAB 22
1350  INVERSE : PRINT " SHAPE                  ";: NORMAL : PRINT " ";: INVERSE
    : PRINT "   PLOT          ";: NORMAL : PRINT "/  N.SH   <- ->  ROT
     P      HPLOT   D  DRAW   S   Z  SCALE   SHIFT  HPLOT TOX  XDRAW
    Erase   Text   Next   Menu";
1360  RETURN
1361 :
1362 :
1364  REM       DEFINIZIONE SLIDE
1365 :
1370  POKE  - 16304,0: POKE  - 16300,0: POKE  - 16297,0: POKE  - 16301,0:
    TC = 0
1390  POKE 230,32: GOSUB 1
1400 SL = 1:RZ = 0:N = 1
1410  GOSUB 1330
1420 KC =  PEEK ( - 16384): POKE  - 16368,0: REM       LETTURA TASTIERA
1430  IF KC = 175 THEN 1570: REM '/'
1440  IF KC = 136 THEN 1610: REM '<-'
1450  IF KC = 149 THEN 1630: REM '->'
1460  IF KC = 218 THEN 1640: REM 'Z'
1470  IF KC = 211 THEN 1660: REM 'S'
1480  IF KC = 197 THEN 1680: REM 'E'
1490  IF KC = 212 THEN 1690: REM 'T'
1500  IF KC = 205 THEN 1080: REM 'M'
1510  IF KC = 196 THEN 1730: REM DRAW
1511  IF KC = 216 THEN 1740: REM XDRAW
1512  IF KC = 208 THEN 1770: REM HPLOT
1513  IF KC = 206 THEN 1750: REM 'N'
1514  IF  PEEK (SW) < 128 THEN 1780: REM  PULSANTE TAVOLETTA
1520  ROT= RZ: SCALE= SL
1530  GOSUB 560: GOSUB 510
1531 :
1535  IF  NOT S1% THEN  POKE 232,177: POKE 233,3: SCALE= 1: ROT= 0: HCOLOR=
    3: XDRAW 1 AT X%,Y%: XDRAW 1 AT X%,Y%: POKE 232,0: POKE 233,112: SCALE=
    SC: ROT= RT: GOTO 1420
1540  HCOLOR= 3: XDRAW N AT X%,Y%
1550  HCOLOR= 0: XDRAW N AT X%,Y%
1560  GOTO 1420
1561 :
1562 :
1563 :
1564  REM       NUMERO SHAPE
1565 :
1570  HOME : VTAB 21: PRINT "SHAPE N.": POKE  - 16301,0:TC = 0
1580 RZ = 0:SL = 1: VTAB 21: HTAB 10: PRINT N: VTAB 21: HTAB 10: INPUT ""
    ;N$:N =  VAL (N$)
1590  IF N >  PEEK (AD) THEN N =  PEEK (AD)
1600  VTAB 21: HTAB 10: PRINT N: GOTO 1420
1601 :
1602 :
1603 :
1604  REM       ROTAZ.ANTIORARIA
1605 :
```

(continua a pagina 96)

Banda (ricavata dalle istantanee di E. Muybridge) che andava inserita nello Zootropio.

Come si vede, i bit interessati sono solo tre e, quindi, ci sarebbe altro spazio per altri ipotetici tipi. Il byte HX è diventato, praticamente, una "parola" o pseudocodice, che servirà ad indicarci l'uso che dovremo fare dei valori X e Y.

Comincia a prendere forma a questo punto la possibilità di un metalinguaggio, cioè di un sistema di codifica e decodifica di istruzioni. Vedremo successivamente gli ulteriori sviluppi, che ci porteranno passo-passo alla creazione di nuovi comandi da aggiungere al Basic Applesoft, eseguibili sia all'interno di un programma, che direttamente da tastiera. Ma, per il momento, torniamo di nuovo alle shape, per le quali influiscono altri parametri, oltre alle coordinate: il numero, la scala e la rotazione. Si tratta di tre altri valori compresi tra 0 e 255 che decidiamo di contenere nello stesso vettore; unica difficoltà è quella di slittare di tre posizioni gli elementi successivi, ma ciò è presto fatto, dato che basta incrementare di tre il puntatore di lettura-scrittura del vettore negli ultimi quattro casi della tabella, cioè nei casi in cui debba essere disegnata una shape. Abbiamo così altri tre byte liberi prima dell'elemento successivo e in essi memorizziamo, nell'ordine, ROT, SCALE e numero. Ricapitolando, il vettore unico sarà formato secondo il seguente schema:
da $6065 in poi

Low X
High X + istruzione
Y
ROT
SCALE
N. Shape
Low X
High X + istruzione
Y

e così via. È ovvio che i tre ulteriori elementi per le shape non occorrono quando queste non vengono usate e che, allora, il vettore unico diventa piu "serrato".

Alla testa del vettore poniamo una serie

Questo programma è disponibile su cassetta presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 126.

di indicatori: il primo dice quante immagini, o SLIDE, sono contenute, cioè quanti sottoinsiemi esistono nel vettore; gli indicatori seguenti dicono quanti elementi ci sono nel primo sottoinsieme, quanti nel secondo e così via. Essi ci permetteranno di riconoscere dei gruppi di elementi grafici che saranno i vari "fotogrammi" di una sequenza. Successivamente le immagini verrano mostrate dalla numero 1 all'ultima; vedremo prossimamente come sia possibile, tuttavia, mescolare l'ordine di presentazione.

Blocchiamoci a questo punto, per non appesantire troppo il discorso e dedichiamoci ad un primo divertente esempio di tutto il sistema, con l'aiuto della tavoletta grafica di MC ed il programma "SHAPE-TABLET", pubblicato sul numero 22 della rivista.

Lo "Zootropio"

Torniamo all'accenno storico che facevamo all'inizio a proposito dei primordi della cinematografia. Quando, dopo il 1870, si riuscirono a creare delle macchine fotografiche istantanee, si pensò subito ad una loro applicazione per studiare il movimento. Nel 1878 Etienne Marey e Eadweard Muybridge giunsero separatamente, quanto simultaneamente, alla creazione del "fucile fotografico", uno strano e poco tranquillizzante aggeggio, che serviva per scattare una sequenza di immagini ad animali che correvano. Si trattava del primo

Lo Zootropio che veniva offerto agli abbonati del settimanale francese "L'Illustration" nel 1879. Facendo ruotare l'apparecchio, si ricreava l'illusione del movimento.

esemplare di cinepresa della storia. Divenne di moda un oggetto, chiamato "Zootropio", che serviva per rivedere il movimento creato da strisce di carta come quella che abbiamo riprodotta, che rappresentavano, appunto, l'animale nelle varie posizioni assunte durante la corsa. Vediamo come possiamo creare il nostro Zootropio elettronico.

Per chi possiede il disco originale "SHA-

(segue da pagina 95)

```
1610 RZ = RZ - 1: IF RZ =  - 1 THEN RZ = 64
1620  GOTO 1420
1621 :
1622 :
1623 :
1624  REM       ROTAZ.ORARIA
1625 :
1630 RZ = RZ + 1: GOTO 1420
1631 :
1632 :
1633 :
1634  REM     INGRANDIMENTO
1635 :
1640 SL = SL + 1: IF SL > 255 THEN SL = 255
1650  GOTO 1420
1651 :
1652 :
1653 :
1654  REM       RIMPICCOLIMENTO
1655 :
1660 SL = SL - 1: IF SL < 1 THEN SL = 1
1670  GOTO 1420
1671 :
1672 :
1673 :
1674  REM       ERASE SCREEN
1675 :
1680  HOME : POKE  - 16301,0:TC = 0: VTAB 21: INPUT "VUOI CANCELLARE LO S
    CHERMO (S/N) ? ";C$: IF C$ = "S" THEN  CALL 62450
1681 :
1682 :
1683 :
1684  REM       RIGHE TESTO
1685 :
1690  IF TC THEN  POKE  - 16301,0:TC = 0: GOTO 1420
1700  POKE  - 16302,0:TC = 1
1710  GOSUB 1330
1720  GOTO 1420
1721 :
1722 :
1723 :
1724  REM       DRAW SHAPE
1725 :
1730  HCOLOR= 3: DRAW N AT X%,Y%:IS% = 4: GOSUB 2: GOSUB 4: GOTO 1570
1731 :
1732 :
1733 :
1734  REM       XDRAW SHAPE
1735 :
1740  HCOLOR= 3: XDRAW N AT X%,Y%:IS% = 6: GOSUB 2: GOSUB 4: GOTO 1570
1741 :
1742 :
1743 :
1744  REM       NEXT SLIDE
1745 :
1750  GOSUB 1: GOTO 1410
1751 :
1752 :
1753 :
1754  REM       HPLOT
1755 :
1770 X1% = X%:Y1% = Y%: GOSUB 560: GOSUB 510:IS% = 0: GOSUB 2: HCOLOR= 3:
    HPLOT X%,Y%: GOTO 1420
1771 :
1772 :
1773 :
1774  REM       HPLOT TO
1775 :
1780  GOSUB 560: GOSUB 510:IS% = 2: GOSUB 2: HCOLOR= 3: HPLOT X1%,Y1% TO
    X%,Y%:X1% = X%:Y1% = Y%: GOTO 1420
2231 :
2232 :
2235 :
2236  REM       INIZIALIZZAZIONI
2237 :
2238 :
2240 DT = 24576:SD% = DT + 100:D$ =  CHR$ (4)
2250  RO1= 0: SCALE= 1
2260 AD = 28672: REM   $7000
2270 HD = 8192: REM   $2000
2310 BE$ =  CHR$ (7):CH = 255: REM SENZA MINUSCOLE CH=64
2320  FOR E = 1 TO 40:W$ = W$ +  CHR$ (CH): NEXT
2330  POKE 232,0: POKE 233,112
2350  DEF  FN PK(I) =  PEEK (I) + 256 *  PEEK (I + 1)
2360 Z0 =  FN PK(797):Z1 =  FN PK(799)
2370 SW = 49251:V0 =  FN PK(801):V1 =  FN PK(803)
2380 PY = 3.14159 / V1:PZ = 3.14159 / V0
2410  GOTO 1080
2420  END
2500 :
2501  REM  AVVERTENZA:
2502  REM  PER IL CORRETTO FUNZIONAMENTO
2503  REM  DEL PROGRAMMA VANNO ELIMINATI
2504  REM  TUTTI I 'REM'
3001 :
3002 :
```

```
3003 :
3004 :
3005 :
3006  REM ---------------------------------
3007  REM          SLIDE - CODER
3008  REM    ANNA - ANIMATION LANGUAGE
3009  REM
4001  REM      MC  MICROCOMPUTER
4002  REM
4003  REM
4004  REM      ROBERTO ANGELETTI
4005  REM
4006  REM    COPYRIGHT  5-5-1984
4007  REM ---------------------------------
```

PE-TABLET" le cose sono più semplici, poiché su di esso è già presente il file "SHZOOTROPE", costituito dalle dieci sagome del cavallo. Comunque, il file può essere creato in questo modo:

1) Si ingrandisce la striscia fino a portare ogni disegno ad una lunghezza di circa 10 cm. Per questa operazione si suggeriscono i seguenti sistemi:

a) si usa un pantografo

b) si fa una diapositiva della pagina e la si proietta

c) si usa un episcopio (che non è un parente vescovo, bensì un proiettore a specchio)

d) si fotocopia ripetutamente la pagina usando un fattore di ingrandimento per ogni passaggio

e) si chiede ad un amico con una "buona mano" di aiutarci.

2) Ottenute con un sistema qualsiasi le sagome nella scala adatta, si mette la prima sulla tavoletta. Si inserisce il programma "SHAPE-TABLET" e si sceglie l'opzione "a vettore".

Cominciando da un punto in mezzo alla pancia, si segue il contorno, approssimandolo con dei segmenti. Finita la prima figura, si passa alla seconda ripartendo dallo stesso punto. Bisogna stare attenti soprattutto all'orientamento del disegno, altrimenti potremmo avere dei risultati deludenti.

Slide-Coder

Il programma pubblicato questo mese serve per generare automaticamente il vettore unico di cui abbiamo parlato, codificando una serie di elementi grafici fino a formare un archivio di disegni. Un tale sistema ha applicazioni che sconfinano da quella di cui ci stiamo occupando direttamente, cioè l'animazione; in sostanza il campo d'uso è vasto e potrebbe essere paragonato a quello di un alfabeto simbolico, come quello egiziano.

Torniamo al cavallo. Una volta inserito il programma e caricata la routine "PADDLE. CODE", si sceglie l'opzione "LOAD shape" che carica a partire da $7000 il file delle sagome ed inizializza i puntatori con POKE 232,0 e POKE 233,112. Si sceglie, poi, "Def. della SLIDE", premendo il numero 4. Apparirà la pagina grafica 1 "sporca" dell'immagine presente in memoria, per cui occorrerà pulirla scegliendo Erase nella lista dei comandi, presente in basso nello schermo. Brevemente, vediamo nella tabella quali sono questi comandi.

Riguardano le Shape-Table:

carattere premuto	operazione
/	cambia il numero della shape visualizzata
← →	rotazioni antioraria e oraria
Z	aumenta il fattore di scala
S	lo diminuisce
D	esegue un DRAW
X	esegue un XDRAW

Riguardano i segmenti:

P	HPLOT (posizionamento) Shift o pulsante Tablet HPLOT TO

Riguardano il sistema:

E	cancella la pagina grafica
T	Fa apparire o scomparire le righe di testo
N	Sposta il puntatore su una nuova slide
M	Torna al Menu

Rispondiamo S alla richiesta di sicurezza per il clear screen e premiamo Return. Vedremo, a questo punto, la prima delle sagome lampeggiare alla posizione in cui si trova il braccio della Tablet.

L'animazione che vogliamo ottenere farà apparire il cavallo da sinistra, attraversare lo schermo e uscire da destra. Come Step tra un'immagine e l'altra seguiremo, per comodità, la quadrettatura della Tablet. Si sceglie una riga lungo la quale "scorrerà" la pancia del cavallo e si dovranno ripetere le seguenti operazioni:
Si pone il braccio della Tablet sul punto e si preme D. Apparirà la scitta "SHAPE N." con il cursore lampeggiante sull'1. Si scriva, ovviamente, "2" e si prema Return. Si inserisca, quindi, "N". Abbiamo così codificato nel vettore, a partire dalla locazione $6000, il primo "fotogramma". Si sposta il braccio di un quadretto e si ripete la sequenza di tasti descritta, chiaramente incrementando il numero della shape. Quando si è arrivati a 10, si ricomincia da 1 e si continua a camminare lungo la riga. Conviene non cancellare lo schermo, in modo da avere un riferimento con le sagome precedenti.

In sostanza, per ogni immagine la sequenza è questa:

1) numero shape <RET>
2) spostare il braccetto
3) Next
4) Draw

eccetto la prima volta, in cui non occorrono i punti 1 e 3, in quanto vengono stabiliti di default. Nel caso noi volessimo far "correre" il cavallo sempre sullo stesso punto dello schermo, basterà non eseguire l'operazione 2. È molto importante essere scrupolosi nell'eseguire queste operazioni in maniera ordinata, altrimenti l'animazione potrebbe venire incongruente in qualche fotogramma.

Quando abbiamo raggiunto il penultimo quadretto, possiamo tornare al Menu e salvare su disco il file che avrà il nome da noi scelto preceduto da "SEQ-" che sta per Sequenza. Dopo di ciò possiamo vedere il frutto della nostra fatica con "SHOW della Slide Table", che esegue l'animazione vera e propria.

Per quanto riguarda la velocità, che ci apparirà senz'altro deludente, c'è da precisare che si è usata una routine in Basic, per ragioni, diciamo, didattiche.

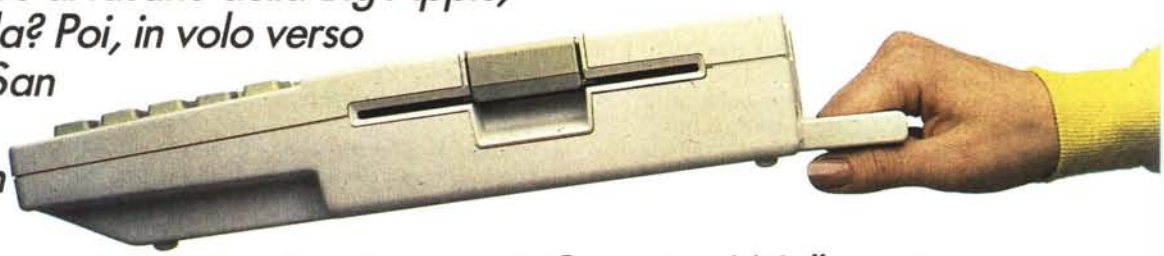
Consigliamo, a questo riguardo, una lettura attenta dei REMark del listato, per comprendere le varie fasi di codifica. Le principali routine sono la 1, 2 e 4, che costruiscono il vettore; da 10 a 300 c'è, invece, il loop per la restituzione della sequenza. Prossimamente pubblicheremo delle routine in assembler, che fanno parte del linguaggio d'animazione di cui abbiamo solo accennato, con il quale si ottiene una rapidità di 6-8 fotogrammi al secondo. Svilupperemo il discorso sulla codifica di immagini generate automaticamente dal computer, come ad esempio per la grafica tridimensionale e la simulazione del movimento di un osservatore intorno ad un oggetto o su una superficie. Vedremo, infine, come è possibile far generare le cosiddette "immagini di interpolazione", quelle, cioè, che servono a dare la continuità ad un insieme discreto di disegni, come, per esempio, nei cartoni animati. Un tale strumento, opportunamente sviluppato, sarebbe senz'altro utile a dei disegnatori satirici che avessero bisogno di creare delle vignette animate. MC

Bibliografia:
A. Stecchina "Hi-Res Move", Bit luglio-agosto '82
F. Petroni "Animazioni con il computer", MCmicrocomputer n. 19 maggio '83
V. Togliasco "Danza al Terminale", Scienze Digest novembre '83

Corri all'edicola e vola in California.

Compra Applicando: puoi vincere un viaggio a Silicon Valley - California. O un Apple IIc.

Fra le pagine di Applicando ora in edicola trovi un biglietto, e forse è proprio quello vincente. Se lo è, buon viaggio! Ti aspetta infatti un fantastico soggiorno di 8 giorni in California passando per New York. Come rinunciare al fascino della Big Apple, la grande mela? Poi, in volo verso l'altra costa: San Francisco. E mentre corri in tram su e giù per la città, pregusta la prossima tappa: si, Cupertino! Nella mitica Silicon Valley sarai accolto dallo staff Apple, e potrai vedere nascere i personal computer Apple. Se invece il tuo biglietto non è quello fortunato, niente paura: infatti puoi partecipare all'estrazione di un Apple IIc, il nuovo personal computer completo, compatto con grafica ad altissima risoluzione. Presto, corri in edicola, la California ti aspetta!



I ♥ YOU

La rivista per i computer Apple.

Qui di seguito abbiamo elencato tutti coloro che sono in grado di eguagliare l'Ampex 210 in Editing, Emulazioni ed Ergonomia.

Esatto, nessuno.

Il che non è una grande sorpresa, perchè far meglio del terminale conversazionale Ampex 210 non è proprio possibile. È completo: dispone di ben 14 emulazioni residenti. È comodo: il suo schermo da 14 pollici è inclinabile e orientabile. È bello: ha colorazione verde o ambra, senza sovrapprezzo. Oltre all'affidabilità che deriva dalla trentennale esperienza Ampex in fatto di periferiche di computer.

Rivolgetevi quindi al più vicino ufficio vendite. E chiedete dell'Ampex 210. Confrontatelo con i prodotti della concorrenza. E non stupitevi quando scoprirete che ogni confronto è impossibile.

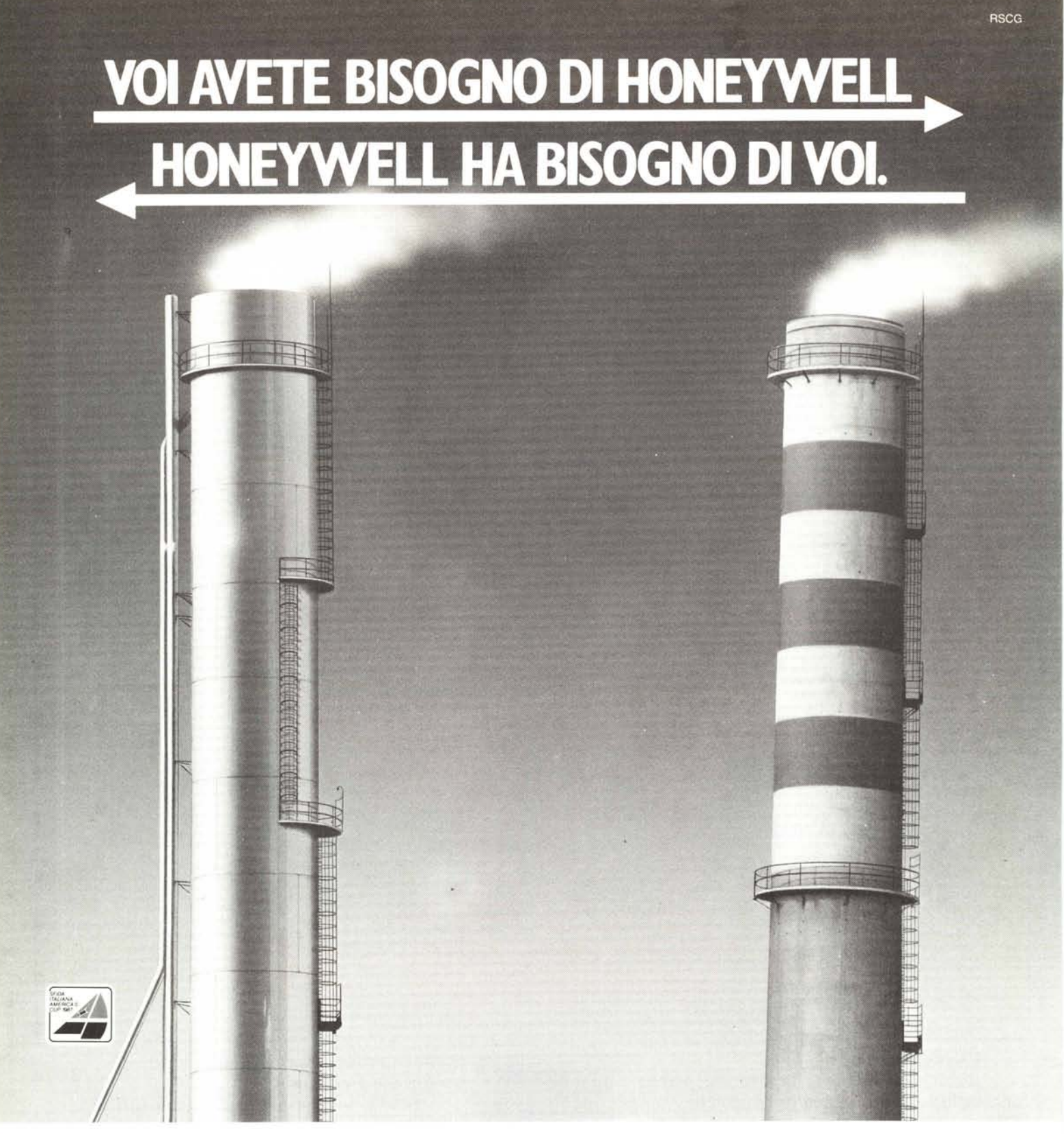
AMPEX

Ampex Corporation • One of The Signal Companies

Ampex Italiana S.p.A., Via Riccardo Gigante 4, 00143-Roma. Casella Postale 10720, Roma-EUR. Tel. 06/55461 Telex: 680243
Ampex Italiana S.p.A., Via Cristoforo Colombo 49, 20090 Trezzano sul Naviglio, Milano. Tel. 02/4459551 Telex: 321246

VOI AVETE BISOGNO DI HONEYWELL

HONEYWELL HA BISOGNO DI VOI.



Le ciminiere sono mute, indicano ma non spiegano. Da sole non dicono nulla sulla realtà produttiva che alla base le anima e le rende diverse. Anche la vostra azienda, alla base, è unica e originale: i vostri uomini, i vostri mezzi, i vostri prodotti, i vostri problemi. Honeywell è riuscita a realizzare sistemi completi per l'elaborazione dati, non guardando da lontano le ciminiere, ma entrando nel vivo del vostro lavoro. Anzi, ha sviluppato la propria attività in funzione delle vostre esigenze specifiche ed è proprio a voi che si rivolge. Pianificare, gestire, calcolare e risolvere: un contributo in termini di informatica che vi era dovuto e che Honeywell è felice di rendervi. Sarebbe un vero peccato se due aziende che hanno tanto da dirsi e da fare insieme non si incontrassero.

Conoscere e risolvere insieme.

Honeywell

Honeywell Information Systems Italia

Per informazioni scrivere a Honeywell I.S.I. Sviluppo Commerciale, Via Vida, 11 - 20127 Milano