

A fianco dei consueti programmi scritti in linguaggio Basic residente del TI-99/4A, iniziamo con questo mese la pubblicazione di programmi in Extended Basic. Speriamo così di soddisfare il numero crescente di richieste che ci sono giunte in proposito. Vogliamo subito precisare che questo nuovo assetto della rubrica non produrrà un restringimento dello spazio dedicato ai programmi in TI Basic. Si tratta invece di un ampliamento per poter soddisfare le richieste di molti lettori. In questa filosofia è anche nostra intenzione, per i prossimi mesi, di pubblicare programmi e notizie che riguardano anche le altre periferiche ed espansioni del TI-99/4A.

Tutto questo sottintende l'invio di materiale da parte vostra.

Dopo parecchi mesi nei quali in questa rubrica ci siamo occupati esclusivamente di giochi, in questo numero vogliamo proporvi tre programmi di utilità, riguardanti la grafica definita dall'utente. Su questo argomento ci sembra esemplare il giudizio del lettore Roberto Prat, autore di uno dei programmi che vi presentiamo, il quale scrive: "Credo che tutti i possessori del TI-99 concordino sul fatto che la composizione di disegni o grafici sia sempre un'operazione laboriosa e assai poco divertente. A differenza di altri personal il nostro Texas non contempla neanche un simbolo grafico ottenibile direttamente dalla tastiera, e ciò rende sempre e comunque necessario il ricorso al sottoprogramma CHAR con cui si ridefinisce nel modo desiderato un qualsiasi carattere ASCII. La nuova sagoma viene precisata attraverso una stringa esadecimale che costituisce uno dei parametri del sottoprogramma CHAR e che deve essere composta dal programmatore. È questa un'operazione che, a seconda della familiarità che l'utente ha con i numeri binari ed esadecimali, può risultare ostica o solamente noiosa, ma che comunque, per il suo carattere ripetitivo, è sicuramente più congeniale alla "filosofia" del calcolatore che non a quella umana. Come annotazione personale posso dire che, pur avendo una certa dimestichezza con la conversione binario-esadecimale, ho ben presto rinunciato a fare "a mano" grandi disegni, più che altro per il tempo che questi richiedevano, tempo ulteriormente esteso dall'inevitabile presenza di errori".

Ecco a voi allora tre diverse soluzioni al problema formulato.

I primi due programmi sono degli editor di caratteri grafici per il TI Basic. Sebbene a prima vista sembri che i due programmi facciano la stessa cosa, in realtà è diversa la filosofia per cui sono stati pensati; quindi abbiamo preferito inserirli entrambi anziché preferire l'uno o l'altro penalizzando gli usi specifici. Nel primo caso l'utente ha la possibilità di definire CONTEMPORANEA-

MENTE un reticolo di dimensioni massime di 24 per 32 pixel (pixel è il singolo puntino che il computer può accendere sullo schermo), ovvero può definire da 1 a 12 caratteri grafici disposti in un rettangolo di dimensioni 3 per 4. Nel secondo programma invece è possibile definire un solo carattere grafico per volta, in compenso si ha a disposizione una "lavagnetta elettronica" di dimensioni 14 per 14 su cui disporre a piacimento i caratteri via via definiti realizzando così disegni grandi e complicati. È chiaro allora che la prima soluzione risulterà assai più veloce e pratica da usare quando bisogna realizzare piccole figure, come omini, automobili, aeroplani, eccetera, da usare eventualmente nell'ambito di disegni più grandi. Questi potranno essere disegnati "in un colpo" solo anche se occupano più di un carattere. Nel caso in cui il disegno da realizzare sia assai vasto, come nel caso di un paesaggio, converrà utilizzare il secondo programma che permette di tenere sempre davanti agli occhi il risultato d'insieme.

Infine il programma per l'Extended Basic, per rimanere in tema con tanto parlare di grafica, non poteva che essere un editor per le sprite, ovvero le figure mobili di dimensioni 64 x 64 pixel che rendono questo linguaggio tanto adatto alla realizzazione di giochi.

GRAFICA CON IL TI-99/4A

di Roberto Prat - Roma

Il disegno che l'utente vuole ottenere dal calcolatore viene composto direttamente sullo schermo per mezzo di un cursore i cui movimenti vengono comandati dai quattro tasti con le frecce (E, S, D, X) nonché, per gli spostamenti obliqui, dai tasti W, R, C, Z. A seconda che tali tasti siano premuti unitamente a CTRL o a FCTN il cursore si muoverà rispettivamente annerendo le caselle di un sottostante reticolo (e quindi tracciando linee), o lasciando le caselle del loro colore; questa seconda modalità permette anche l'eventuale cancellazione di caselle annerite in precedenza (correzioni). In questo caso il cursore non esegue le traslazioni diagonali. Vediamo comunque in dettaglio come si adopera il programma.

Dopo il RUN appare l'intestazione che permane fino alla pressione di un tasto; poi si forma un reticolo formato dalle $32 \times 24 = 768$ aree di carattere che il TI-99 può visualizzare. Il reticolo è diviso in 12 caselle quadrate di 8×8 caratteri, disposte in tre file orizzontali. Nell'angolo in alto a sinistra appare il cursore (quadrato nero lampeggiante) che può essere spostato sul reticolo premendo i tasti come descritto in

```

100 REM   COMPOSIZIONE GRAFICA
110 REM   **DI ROBERTO PRAT
120 CALL SCREEN(10)
130 CALL CLEAR
140 PRINT "#### GRAFICA SUL TI-99 ####"
150 FOR I=1 TO 20
160 PRINT
170 NEXT I
180 CALL CHAR(120,"FFFFFFFFFFFFFF")
190 CALL CHAR(128,"FFFFFFFFFFFFFF")
200 CALL HCHAR(5,14,120,3)
210 CALL HCHAR(6,14,120,3)
220 CALL HCHAR(7,14,120,8)
230 CALL HCHAR(8,10,120,12)
240 CALL HCHAR(9,10,120,12)
250 CALL HCHAR(10,11,120,12)
260 CALL HCHAR(11,12,120,11)
270 CALL HCHAR(12,15,120,7)
280 CALL HCHAR(13,15,120,5)
290 CALL HCHAR(14,15,120,5)
300 CALL HCHAR(15,15,120,5)
310 CALL HCHAR(16,16,120,4)
320 CALL HCHAR(17,16,120,5)
330 CALL HCHAR(18,17,120,4)
340 CALL HCHAR(19,18,120,2)
350 CALL HCHAR(8,15,128,3)
360 CALL VCHAR(9,16,128,6)
370 CALL VCHAR(10,16,128,5)
380 CALL COLOR(12,5,1)
390 PRINT "PREMERE UN TASTO"
400 CALL KEY(5,KEY,ST)
410 IF ST=0 THEN 400
420 OPTION BASE 1
430 DIM SAG(24,32)
440 CALL CHAR(112,"FFB18181818181FF")
450 CALL CHAR(111,"FF1818181818181FF")
460 CALL CHAR(142,"007E7E7E7E7E00")
470 CALL CLEAR
480 CALL HCHAR(1,1,112,768)
490 CALL VCHAR(1,8,111,24)
500 CALL VCHAR(1,16,111,24)
510 CALL VCHAR(1,24,111,24)
520 CALL HCHAR(8,1,111,32)
530 CALL HCHAR(16,1,111,32)
540 CALL COLOR(11,15,16)
550 CALL COLOR(10,5,16)
560 X=1
570 Y=1
580 CALL SCREEN(3)
590 CALL COLOR(2,2,15)
600 CALL KEY(5,KEY,ST)
610 CALL VCHAR(Y,X,42)
620 CALL VCHAR(Y,X,112)
630 IF ST=0 THEN 600
640 IF KEY=13 THEN 3170
650 IF KEY=11 THEN 640
660 ON KEY-6 GO TO 820,670,700,730,760
670 IF X=1 THEN 600
680 X=X-1
690 GO TO 780
700 IF X=32 THEN 600
710 X=X+1
720 GO TO 780
730 IF Y=24 THEN 600
740 Y=Y+1
750 GO TO 780
760 IF Y=1 THEN 600
770 Y=Y-1
780 CALL VCHAR(Y,X,42)
790 CALL VCHAR(Y,X,112)
800 SAG(Y,X)=0
810 GO TO 600
820 CALL KEY(5,KEY,ST)
830 IF ST=0 THEN 820
840 IF KEY<>132 THEN 880
850 IF X=32 THEN 820
860 X=X+1
870 GO TO 1230
880 IF KEY<>133 THEN 920
890 IF Y=1 THEN 820
900 Y=Y-1
910 GO TO 1230
920 IF KEY<>147 THEN 960
    
```

```

930 IF X=1 THEN 820
940 X=X-1
950 GO TO 1230
960 IF KEY<>152 THEN 1000
970 IF Y=24 THEN 820
980 Y=Y+1
990 GO TO 1230
1000 IF KEY<>131 THEN 1060
1010 IF X=32 THEN 820
1020 IF Y=24 THEN 820
1030 X=X+1
1040 Y=Y+1
1050 GO TO 1230
1060 IF KEY<>151 THEN 1120
1070 IF X=1 THEN 820
1080 IF Y=1 THEN 820
1090 X=X-1
1100 Y=Y-1
1110 GO TO 1230
1120 IF KEY<>154 THEN 1180
1130 IF X=1 THEN 820
1140 IF Y=24 THEN 820
1150 X=X-1
1160 Y=Y+1
1170 GO TO 1230
1180 IF KEY<>146 THEN 1260
1190 IF X=32 THEN 820
1200 IF Y=1 THEN 820
1210 X=X+1
1220 Y=Y-1
1230 CALL VCHAR(Y,X,42)
1240 SAG(Y,X)=1
1250 GO TO 820
1260 REM
1270 IF KEY<>13 THEN 640
1280 IF KEY<>145 THEN 560
1290 CALL SCREEN(5)
1300 DIM E$(16,12),S$(STRING$(16))
1310 REM
1320 CALL KEY(5,KEY,ST)
1330 IF ST=0 THEN 1320
1340 IF ST=-1 THEN 1320
1350 CALL SCREEN(9)
1360 IF KEY<>148 THEN 1400
1370 A=1
1380 D=1
1390 GO TO 1830
1400 IF KEY<>153 THEN 1440
1410 A=1
1420 D=2
1430 GO TO 1830
1440 IF KEY<>149 THEN 1480
1450 A=1
1460 D=3
1470 GO TO 1830
1480 IF KEY<>137 THEN 1520
1490 A=1
1500 D=4
1510 GO TO 1830
1520 IF KEY<>135 THEN 1560
1530 A=2
1540 D=1
1550 GO TO 1830
1560 IF KEY<>136 THEN 1600
1570 A=2
1580 D=2
1590 GO TO 1830
1600 IF KEY<>138 THEN 1640
1610 A=2
1620 D=3
1630 GO TO 1830
1640 IF KEY<>139 THEN 1680
1650 A=2
1660 D=4
1670 GO TO 1830
1680 IF KEY<>130 THEN 1720
1690 A=3
1700 D=1
1710 GO TO 1830
1720 IF KEY<>142 THEN 1760
1730 A=3
1740 D=2
1750 GO TO 1830

1760 IF KEY<>141 THEN 1800
1770 A=3
1780 D=3
1790 GO TO 1830
1800 IF KEY<>128 THEN 1830
1810 A=3
1820 D=4
1830 CALL KEY(5,KEY,ST)
1840 IF ST=0 THEN 1830
1850 IF ST=-1 THEN 1830
1860 CALL SCREEN(5)
1870 IF KEY<>148 THEN 1910
1880 B=1
1890 E=1
1900 GO TO 2360
1910 IF KEY<>153 THEN 1950
1920 B=1
1930 E=2
1940 GO TO 2360
1950 IF KEY<>149 THEN 1990
1960 B=1
1970 E=3
1980 GO TO 2360
1990 IF KEY<>137 THEN 2030
2000 B=1
2010 E=4
2020 GO TO 2360
2030 IF KEY<>135 THEN 2070
2040 B=2
2050 E=1
2060 GO TO 2360
2070 IF KEY<>136 THEN 2110
2080 B=2
2090 E=2
2100 GO TO 2360
2110 IF KEY<>138 THEN 2150
2120 B=2
2130 E=3
2140 GO TO 2360
2150 IF KEY<>139 THEN 2190
2160 B=2
2170 E=4
2180 GO TO 2360
2190 IF KEY<>130 THEN 2230
2200 B=3
2210 E=1
2220 GO TO 2360
2230 IF KEY<>142 THEN 2270
2240 B=3
2250 E=2
2260 GO TO 2360
2270 IF KEY<>141 THEN 2310
2280 B=3
2290 E=3
2300 GO TO 2360
2310 IF KEY<>128 THEN 1930
2320 B=3
2330 E=4
2340 REM
2350 REM
2360 L=1
2370 FOR Y=A TO B
2380 FOR X=D TO E
2390 L=X+4*(Y-1)
2400 FOR V=1 TO 8
2410 FOR K=1 TO 2
2420 N=0
2430 FOR U=1 TO 4
2440 N=N+2*(4-U)*SAG(V+8*(Y-1),U+8*(X-1)
+4*(K-1))
2450 NEXT U
2460 H=K+2*(V-1)
2470 IF N>9 THEN 2500
2480 E$(H,L)=STR$(N)
2490 GO TO 2620
2500 ON N-9 GO TO 2510,2530,2550,
2570,2590,2610
2510 E$(H,L)="A"
2520 GO TO 2620
2530 E$(H,L)="E"
2540 GO TO 2620
2550 E$(H,L)="C"
2560 GO TO 2620

2570 E$(H,L)="D"
2580 GO TO 2620
2590 E$(H,L)="E"
2600 GO TO 2620
2610 E$(H,L)="F"
2620 NEXT K
2630 NEXT V
2640 NEXT X
2650 NEXT Y
2660 CALL SCREEN(3)
2670 CALL CLEAR
2680 REM
2690 FOR L=D+(A-1)*4 TO 4*(B-1)+E
2700 FOR K=1 TO 16
2710 STRING$(L)=STRING$(L)&E$(K,L)
2720 NEXT K
2730 CALL SOUND(200,1319,4)
2740 CALL SOUND(200,1047,4)
2750 CALL SOUND(200,1319,4)
2760 CALL SOUND(200,1047,4)
2770 PRINT "SAGOMA";L;"=";STRING$(L)
2780 NEXT L
2790 FOR J=1 TO 4
2800 PRINT
2810 NEXT J
2820 FOR Y=1 TO 3
2830 FOR X=1 TO 4
2840 G=X+4*(Y-1)
2850 IF STRING$(G)<>" THEN 2870
2860 STRING$(G)="0000000000000000"
2870 CALL CHAR(G+97,STRING$(G))
2880 CALL VCHAR(20+Y,13+X,X+97+4*(Y-1))
2890 NEXT X
2900 NEXT Y
2910 CALL COLOR(9,5,16)
2920 CALL COLOR(10,5,16)
2930 PRINT "          PREMERE"
2940 PRINT "1 PER RI-INCINCIARE"
2941 PRINT "2 PER DISEGNARE SUL RETICOLO"
2942 PRINT "3 DISEGNARE SENZA RETICOLO"
2950 FOR I=1 TO 12
2960 STRING$(I)=" "
2970 NEXT I
2980 CALL KEY(1,KEY,ST)
2990 IF ST=0 THEN 2980
3000 IF KEY=8 THEN 3160
3010 IF KEY=7 THEN 3060
3020 CALL CLEAR
3030 CALL SCREEN(9)
3040 CALL COLOR(2,5,5)
3050 GO TO 3070
3060 CALL VCHAR(1,1,12,768)
3070 FOR K=1 TO 24
3080 FOR L=1 TO 32
3090 IF SAG(K,L)=0 THEN 3110
3100 CALL VCHAR(K,L,42)
3110 NEXT L
3120 NEXT K
3130 IF KEY=7 THEN 600
3140 CALL KEY(4,KEY,ST)
3150 IF ST=0 THEN 3140
3160 CALL SCREEN(12)
3170 FOR I=1 TO 24
3180 FOR J=1 TO 32
3190 SAG(I,J)=0
3200 NEXT J
3210 NEXT I
3220 CALL SCREEN(3)
3230 GO TO 480

```

precedenza. Si ha quindi a disposizione qualcosa di simile ad un foglio di 32×24 quadretti su cui è possibile comporre qualunque disegno (chi ha vocazioni artistiche può limitarsi ad impiegare il programma per trasformare il televisore in una sorta di lavagna elettronica...).

È importante ricordarsi che il computer "annerisce" o "cancella" non la casella in cui si trova il cursore, ma quella in cui va a portarsi in seguito alla pressione di uno dei tasti E, S, D, X insieme al tasto CTRL o al tasto FCTN.

Al termine del disegno se ne può chiedere la conversione in stringhe esadecimali premendo CTRL Q. Il cambiamento di colore dello sfondo da verde a blu segnala che il comando è stato recepito. A questo punto il calcolatore è in attesa di precisazioni riguardo al numero di caselle che deve analizzare: non è detto, infatti, che il disegno occupi tutte e dodici le caselle dello schermo e sarebbe un'inutile perdita di tempo fargli codificare stringhe di soli zeri. Le caselle del reticolo sono perciò poste in corrispondenza con i dodici tasti

| | | | | | | | | |
|---|---|---|---|--|---|---|---|---|
| T | Y | U | I | Ripartizione dello schermo in caselle. Ogni casella è formata da 8×8 aree di carattere. | Y | U | I | |
| G | H | J | K | | G | H | J | K |
| B | N | M | . | | B | N | M | . |

e la pressione di due di essi determina l'esplorazione di tutte e sole le caselle comprese fra quella associata al primo e quella associata al secondo. Ad esempio, premendo Y e J si ottiene la codifica delle quattro caselle Y U H J; premendo due volte T si ottiene la codifica della sola casella in alto a sinistra; premendo T e ., si ottiene la codifica di tutto lo schermo; ecc. Entrambi i tasti devono essere premuti insieme a CTRL e l'accettazione dei comandi è accompagnata ancora da cambiamenti di colore dello sfondo. Dopo un tempo che varia da alcuni secondi a un minuto e mezzo, a seconda del numero di caselle da scandire, sparisce il reticolo ed appaiono le stringhe associate alle sagome disegnate nelle singole caselle. Inoltre viene riprodotto il disegno a grandezza naturale, cioè come appare fornendo ad altrettanti sottoprogrammi CHAR le stringhe esadecimali ottenute. Se il programmatore è soddisfatto del risultato non deve far altro che annotarsi le stringhe esadecimali per utilizzarle nel disegno che sta componendo; in caso contrario un menu gli fornisce tre alternative: ricominciare (tasto 1); rivedere il disegno sul reticolo per procedere ad eventuali modifiche (tasto 2); rivedere il disegno su uno sfondo omogeneo (tasto 3). Se sceglie quest'ultima opzione, in seguito potrà soltanto far ricominciare il programma premendo un tasto qualsiasi.

Analisi del listato

Data la particolare leggibilità e semplicità di funzionamento di "Grafica con il TI 99", abbiamo preferito non modificarne il listato. Infatti sarebbe stato possibile ottenere alcune funzionalità del programma con un numero di istruzioni assai minore di quanto è stato fatto dall'autore, ma ciò ne avrebbe compromesso di molto la leggibilità. Poiché lo scopo di questa rubrica non dovrebbe essere solo quello di fornirvi programmi già belli e fatti, ma anche idee e consigli e soprattutto spunti per la programmazione, abbiamo preferito preservare l'aspetto della leggibilità. Vi proponiamo quindi una breve analisi del listato, per facilitarne la comprensione, specie a chi voglia effettuare delle modifiche per adattare il programma alle proprie esigenze. Prima di concludere vorremmo attirare la vostra attenzione sull'ottimo uso che l'autore ha fatto dell'istruzione CALL KEY più volte nel corso del programma, ad esempio selezionando di volta in volta l'unità di tastiera più opportuna al suo scopo.

| | | |
|------|------|---|
| 100 | 380 | Schermata iniziale e presentazione. Caricando il programma, se avete fretta, potete evitare di caricare questa parte. |
| 390 | 410 | Ciclo attesa. |
| 420 | 430 | Dimensionamento matrice. |
| 440 | 550 | Disegno reticolo. |
| 560 | 570 | Inizializza x,y coordinate cursore. |
| 600 | 630 | Attesa comando e lampeggio cursore. |
| 640 | | Se viene premuto ENTER cancella tutto e ricomincia. |
| 650 | | Se è stato premuto CTRL... |
| 660 | | salta a seconda della pressione di FCTN e la freccia. |
| 670 | 770 | Aggiorna i valori di y,x secondo 660. |
| 780 | 800 | Spegne sullo schermo e in memoria il punto. |
| 810 | | Chiude il ciclo. |
| 820 | 1220 | Se è stato premuto CTRL e una freccia aggiorna le coordinate y,x. |
| 1230 | 1250 | Accende il punto sullo schermo e in memoria. |
| 1280 | | Se non è stato premuto CTRL Q riprende il ciclo attesa comandi. |
| 1320 | 1820 | Mette nelle variabili A e D le coordinate della prima casella da codificare a seconda del tasto premuto. |
| 1830 | 2330 | Mette in B ed E le coordinate dell'ultima casella da codificare, a seconda del tasto premuto. |
| 2350 | 2650 | Costruzione della matrice dei caratteri esadecimali. |
| 2440 | | Calcola il valore decimale corrispondente ad una quartina di punti. |
| 2470 | 2610 | Se il valore ottenuto è maggiore di 9, sostituisce al numero decimale la cifra esadecimale. |

| | | |
|------|------|--|
| 2680 | | Inizio routine di presentazione. |
| 2690 | 2710 | Giustappone i caratteri esadecimali così da ottenere le stringhe relative ai caratteri definiti. |
| 2770 | | Stampa la stringa. |
| 2820 | 2900 | Stampa le 12 sagome generate. |
| 2910 | 2990 | Attesa nuovi comandi. |
| 3000 | | Se premuto 1, cancella tutto e ricomincia. |
| 3010 | | Se premuto 2, disegna la sagoma con il reticolo. |
| 3060 | | Ridisegna il reticolo. |
| 3070 | 3120 | Ridisegna le sagome sul reticolo o sullo schermo bianco. |
| 3130 | 3230 | Se era stato premuto 2, ritorna in attesa di comandi per modificare il disegno, altrimenti attende che venga premuto un tasto qualunque per cancellare tutto e ricominciare da capo. |

DEF CHAR

di Roberto Sassoli - Ferrara

Dopo aver dato il RUN sul video compariranno 2 riquadri, uno più piccolo e scuro destinato alla definizione del carattere, l'altro più grande e chiaro destinato alla disposizione dei caratteri già creati a mo' di mosaico. Una freccia che punta alla scritta "DEF CHAR" indica che si è in fase di definizione di carattere. Analogamente al programma precedente l'idea è quella di annerire, in una griglia 8×8 , quei quadretti in corrispondenza dei quali si vogliono accendere i pixel del carattere da generare. Per far ciò ci sposteremo nell'area scura mediante i soliti tasti E,S,D,X e, una volta posizionato il cursore (questa volta non lampeggiante), premeremo il tasto 1; a questo punto la casella rimarrà annerita e il cursore si riporterà nella casella in alto a sinistra. Se volessimo annerire proprio quest'ultima casella basterà semplicemente premere il tasto 1. Notate che il cursore può passare sui quadretti già definiti senza cancellarli. In caso di errori si potrà procedere in due modi: o cancellare tutto il carattere in corso di definizione (tasto 2), oppure cancellare solo i quadretti sbagliati, sovrapponendosi con il cursore e premendo lo zero.

Una volta che il carattere sarà pronto si dovrà premere -SHIFT 1- e il computer passerà alla fase successiva, che è quella di calcolo della stringa esadecimale corrispondente al carattere. Durante questa fase la freccia punterà sulla dicitura "EX-CHAR". Dopo alcuni secondi comparirà sul video la stringa esadecimale desiderata e il carattere appena generato andrà a porsi in alto a destra nel quadrato più grande per la fase di posizionamento. Anche qui i tasti E,S,D,X serviranno per muovere il carattere all'interno dell'area disponibile; il fat-

to di passare sopra altri caratteri già disposti non altererà questi ultimi. Quando avremo posto il carattere nella posizione desiderata premeremo il tasto I. Dopo questa operazione il computer tornerà di nuovo alla fase di definizione di un carattere e il ciclo si ripeterà.

SPRITE EDITOR

di Aristide Torrelli - Roma

Come probabilmente i nostri lettori già sapranno, le sprite possono essere di quattro formati: due normali, ed i loro rispettivi ingrandimenti che ne raddoppiano le dimensioni. Tali formati vengono selezionati tramite l'istruzione CALL MAGNIFY. Il primo formato normale occupa una sola area carattere, il secondo ne occupa quattro, disposte in un quadrato 2×2 . Le sprite vengono definite mediante le solite stringhe esadecimali, alla stessa maniera della grafica definita dall'utente; le stringhe esadecimali vanno inserite mediante CALL CHAR nell'area di memoria riservata ai caratteri ASCII. In fase di chiamata della procedura SPRITE, andrà poi inserito tra i parametri di definizione proprio il codice ASCII relativo alla stringa. Mentre per le sprite di formato piccolo, ad un solo carattere, il discorso termina qui, per le sprite del formato più grande sono necessari quattro caratteri, ovvero quattro stringhe esadecimali di sedici caratteri ciascuna. L'Extended Basic permette di definire quattro caratteri ASCII consecutivi mediante una sola CALL CHAR avente come parametri il codice del primo carattere e una stringa esadecimale lunga sessantaquattro caratteri. È importante che il primo codice ASCII sia un numero divisibile per quattro. Nella CALL SPRITE sarà sufficiente specificare tale codice ed, in seguito, il formato dello sprite (3 oppure 4).

Sperando di non avere annoiato con questa introduzione quelli di voi già esperti in materia, eccovi un programmino che risparmia la fatica di calcolare con carta e matita la stringa dei sessantaquattro caratteri esadecimali.

Il programma funziona in maniera analoga ai due precedenti. Dopo il RUN compare il reticolo 16×16 , con in alto a sinistra il cursore; questo può essere spostato mediante i tasti E,S,D,X e, sebbene questa possibilità non venga segnalata sul video per motivi di spazio, mediante i tasti W,R,Z,C per gli spostamenti in diagonale. Mediante i tasti zero ed uno è possibile modificare l'effetto del passaggio del cursore sui quadretti. Se è stato premuto I, il cursore "annerisce" ogni quadretto su cui si viene a trovare, se è stato premuto O, il

```

10 REM *****DEF CHAR****
20 REM *****DI*****
30 REM *****SASSOLI****
40 REM *****ROBERTO****
50 REM *****FERRARAT****
60 DIM A(64)
70 HG=92
80 CALL CHAR(41,"01010101010101")
90 CALL CHAR(42,"80E08080808080")
100 CALL CHAR(43,"00000000000000FF")
110 CALL CHAR(44,"FF")
120 CALL CHAR(40,"FFFFFFFFFFFFFFFF")
130 CALL CHAR(45,"00001C7F3E1C08")
140 CALL CLEAR
150 CALL COLOR(1,5,1)
160 CALL CHAR(33,"FFFFFFFFFFFFFFFF")
170 B$="DEF. COD. POS."
180 X$="CHAR CHAR CHAR"
190 FOR I=1 TO LEN(B$)
200 JK=ASC(SEG$(B$,I,1))
210 KJ=ASC(SEG$(X$,I,1))
220 CALL HCHAR(23,I+6,JK)
230 CALL HCHAR(24,I+6,KJ)
240 NEXT I
250 CALL HCHAR(2,12,43,15)
260 CALL VCHAR(3,11,41,15)
270 CALL VCHAR(3,27,42,15)
280 CALL HCHAR(18,12,44,15)
290 CALL HCHAR(2,2,33,8)
300 CALL HCHAR(3,2,3,8)
310 CALL HCHAR(4,2,33,8)
320 CALL HCHAR(5,2,3,8)
330 CALL HCHAR(6,2,33,8)
340 CALL HCHAR(7,2,33,8)
350 CALL HCHAR(8,2,33,8)
360 CALL HCHAR(9,2,3,8)
370 FOR I=1 TO 64
380 A(I)=0
390 NEXT I
400 CALL HCHAR(22,2,32,28)
410 CALL HCHAR(22,8,43)
420 FS=0
430 A$=""
440 C=2
450 R=2
460 CALL KEY(O,K,S)
470 CALL HCHAR(R,C,40)
480 IF (S=0)+(S=-1) THEN 460
490 IF K=83 THEN 570
500 IF K=68 THEN 620
510 IF K=69 THEN 670
520 IF K=88 THEN 720
530 IF K=50 THEN 290
540 IF K=48 THEN 810
550 IF K=33 THEN 850
560 IF K=49 THEN 770 ELSE 460
570 IF C=2 THEN 460
580 CALL HCHAR(R,C,X)
590 CALL GCHAR(R,C-1,X)
600 C=C-1
610 GOTO 460
620 IF C=9 THEN 460
630 CALL HCHAR(R,C,X)
640 CALL GCHAR(R,C+1,X)
650 C=C+1
660 GOTO 460
670 IF R=2 THEN 460
680 CALL HCHAR(R,C,X)
690 CALL GCHAR(R-1,C,X)
700 R=R-1
710 GOTO 460
720 IF R=9 THEN 460
730 CALL HCHAR(R,C,X)
740 CALL GCHAR(R+1,C,X)
750 R=R+1
760 GOTO 460
770 CALL HCHAR(R,C,40)
780 SD=R*8+C-17
790 A(SD)=1
800 GOTO 440
810 SD=R*8+C-17
820 CALL HCHAR(R,C,33)
830 A(SD)=0
840 GOTO 440
850 IF A(1)=1 THEN 880
860 CALL HCHAR(2,2,33)
870 GOTO 890
880 CALL HCHAR(2,2,40)
890 CALL HCHAR(22,2,32,28)
900 CALL HCHAR(22,15,45)
910 FOR I=1 TO 16
920 N=0
930 FOR L=3 TO 0 STEP -1
940 FS=FS+1
950 N=N+(2^L)*A(FS)
960 NEXT L
970 IF N>9 THEN 1000
980 C$=STR$(N)
990 GOTO 1130
1000 H=N-9
1010 ON H GOTO 1020,1040,1060,
1080,1100,1120
1020 C$="A"
1030 GOTO 1130
1040 C$="B"
1050 GOTO 1130
1060 C$="C"
1070 GOTO 1130
1080 C$="D"
1090 GOTO 1130
1100 C$="E"
1110 GOTO 1130
1120 C$="F"
1130 A$=A$&C$
1140 NEXT I
1150 D$="COD. CHAR="&A$
1160 FOR I=1 TO LEN(D$)
1170 F=ASC(SEG$(D$,I,1))
1180 CALL HCHAR(20,I+1,F)
1190 NEXT I
1200 HG=HG+1
1210 X=32
1220 CALL CHAR(HG,A$)
1230 C=12
1240 R=3
1250 CALL HCHAR(22,2,32,28)
1260 CALL HCHAR(22,2,1,45)
1270 CALL KEY(O,K,S)
1280 CALL HCHAR(R,C,HG)
1290 IF S=0 THEN 1270
1300 IF K=83 THEN 1350
1310 IF K=68 THEN 1400
1320 IF K=69 THEN 1450
1330 IF K=49 THEN 290
1340 IF K=88 THEN 1500 ELSE 1270
1350 IF C=12 THEN 1270
1360 CALL HCHAR(R,C,X)
1370 CALL GCHAR(R,C-1,X)
1380 C=C-1
1390 GOTO 1270
1400 IF C=26 THEN 1270
1410 CALL HCHAR(R,C,X)
1420 CALL GCHAR(R,C+1,X)
1430 C=C+1
1440 GOTO 1270
1450 IF R=3 THEN 1270
1460 CALL HCHAR(R,C,X)
1470 CALL GCHAR(R-1,C,X)
1480 R=R-1
1490 GOTO 1270
1500 IF R=17 THEN 1270
1510 CALL HCHAR(R,C,X)
1520 CALL GCHAR(R+1,C,X)
1530 R=R+1
1540 GOTO 1270

```

cursore lascia quella casella bianca ovvero la cancella se era stata già annerita in precedenza. Tale fatto viene segnalato dal colore del cursore stesso: se esso è completamente nero l'effetto è quello di annerire, se presenta un quadratino bianco al centro, l'effetto è quello di cancellare.

Quando sarete soddisfatti della sagoma da voi creata, premendo il tasto P passerete alla fase successiva. Dopo alcuni secondi,

necessari per il calcolo della stringa esadecimale, sul video compare la sprite appena definita. Premendo il tasto M potrete passare al formato ingrandito (che occupa un'area 4 x 4) e viceversa; premendo il tasto C la sprite assumerà di volta in volta un colore diverso; premendo il tasto E il programma tornerà in modalità definizione di sagoma e potrete apportare delle modifiche. Infine la pressione del tasto L, quando sa-

rete soddisfatti del risultato ottenuto, produrrà la stampa della stringa esadecimale e dei codici relativi al colore ed al fattore ingrandimento. Dopo tale operazione non è possibile riottenere sul reticolo la vecchia sagoma, ma è possibile solo uscire dal programma oppure ricominciare da capo per definire una nuova sagoma.

MC

```

100 REM SPRITE EDITOR
110 REM DI A. TORRELLI
120 CAR=100
130 M1=3 :: M2=4
140 DIM Z(16,16)
150 PROX(3)=4 :: PROX(4)=3
160 CO=7
170 CALL CHAR(100,"01010101010101FF")
180 CALL CHAR(101,"F::FFFFFFFFFFFF")
190 CALL CHAR(102,"FFFFFF3C3C3C3FFFF")
200 CALL COLOR(9,9,16)
210 CALL CLEAR
220 DISPLAY AT(1,5):"EDITORE DI SPRITE"
230 FOR R=1 TO 16
240 CALL HCHAR(R+4,2,100,16)
250 NEXT R
260 CALL MAGNIFY(1)
270 IF K=69 THEN 290
280 CALL SCREEN(8)
290 CALL DELSPRITE(ALL)
300 CALL SPRITE(#15,102,14,32,8)
310 DISPLAY AT(21,1):KPT$( " ",112)
320 DISPLAY AT(22,2):"E=SU X=GIU' S=SIN. D=DESTRA"
330 DISPLAY AT(23,2):"PREMI 1-ON      0-OFF"
340 DISPLAY AT(24,2):"PREMI P PER AVERE LO SPRITE"
350 R=1 :: C=1
360 CALL KEY(0,K,S):: IF S=0 THEN 360
370 IF K=48 THEN CAR=100
380 IF K=49 THEN CAR=101
390 IF K=83 THEN C=C-1 :: GOTO 480
400 IF K=68 THEN C=C+1 :: GOTO 480
410 IF K=69 THEN R=R-1 :: GOTO 480
420 IF K=88 THEN R=R+1 :: GOTO 480
430 IF K=82 THEN R=R-1 :: C=C+1 :: GOTO 480
440 IF K=87 THEN R=R-1 :: C=C-1 :: GOTO 480
450 IF K=90 THEN R=R+1 :: C=C-1 :: GOTO 480
460 IF K=67 THEN R=R+1 :: C=C+1 :: GOTO 480
470 IF K=80 THEN 560
480 IF C<1 THEN C=16
490 IF C>16 THEN C=1
500 IF R<1 THEN R=16
510 IF R>16 THEN R=1
520 CALL LOCATE(#15,(R*8)+25,C*8+1)
530 CALL HCHAR(R+4,C+1,CAR)
540 CALL SOUND(20,2000,0)
550 GOTO 360
560 CALL DELSPRITE(ALL)
570 DISPLAY AT(21,1):KPT$( " ",112)
580 DISPLAY AT(22,2):"      ATTENDI PER FAVORE."
590 FOR R=1 TO 16
600 FOR C=1 TO 16
610 CALL GCHAR(R+4,C+1,CAR)
620 CA=CA-100
630 Z(R,C)=CA
640 NEXT C
650 NEXT R
660 ESA$="0123456789ABCDEF" :: D$=""
670 FOR R=1 TO 16
680 BASSO=Z(R,5)*8+Z(R,6)*4+Z(R,7)*2+Z(R,8)+1
690 ALTO=Z(R,1)*8+Z(R,2)*4+Z(R,3)*2+Z(R,4)+1
700 D$=D$&SEG$(ESA$,ALTO,1)&SEG$(ESA$,BASSO,1)
710 NEXT R
720 FOR R=1 TO 16
730 BASSO=Z(R,13)*8+Z(R,14)*4+Z(R,15)*2+Z(R,16)+1
740 ALTO=Z(R,9)*8+Z(R,10)*4+Z(R,11)*2+Z(R,12)+1
750 D$=D$&SEG$(ESA$,ALTO,1)&SEG$(ESA$,BASSO,1)
760 NEXT R
770 CALL CHAR(120,D$)
780 CALL MAGNIFY(3)
790 CALL SPRITE(#1,120,CO,80,170)
800 DISPLAY AT(21,2):"C COLORE  M MAGNIFY  E EDIT"
810 DISPLAY AT(22,2):"A RIFAI  F FINE  "
820 DISPLAY AT(23,8):"L LISTA LA STRINGA"
830 CALL KEY(0,K,S)
840 IF K=76 THEN 970
850 IF K=70 THEN 960
860 IF K=65 THEN 100
870 IF K=69 THEN 260
880 IF K=77 THEN 910
890 IF K=67 THEN 1110
900 GOTO E30
910 CALL MAGNIFY(PROX(M2))
920 M1=PROX(M2)
930 M2=M1
940 FOR D=1 TO 50 :: NEXT D
950 GOTO 830
960 CALL DELSPRITE(ALL):: CALL CLEAR :: END
970 REM STAMPA STRINGA
980 CALL SCREEN(8):: CALL DELSPRITE(ALL)
990 CALL CLEAR
1000 PRINT "      ECCO LA STRINGA" :: PRINT ::
1010 PRINT
1020 PRINT D$: :
1030 PRINT "MAGNIFY=";M1,"COLORE";CO
1040 PRINT :: PRINT :: PRINT :: PRINT :: PRINT
1050 PRINT RPT$("-",78)
1060 DISPLAY AT(24,3):"      A RIFAI  F FINE"
1070 CALL KEY(0,K,ST):: IF ST=0 THEN 1070
1080 IF K=70 THEN CALL DELSPRITE(ALL):: END
1090 IF K=65 THEN 100
1100 GOTO 1070
1110 CO=CO+1 :: IF CO>16 THEN CO=2
1120 CALL COLOR(#1,CO)
1130 FOR U=1 TO 50 :: NEXT U :: GOTO 830

```