

software

C-64

a cura di Tommaso Pantuso

NL-PRINTGRAF

Hardcopy della pagina grafica su stampante STAR NL-10

di Roberto Morassi - Pistoia

Con le stampanti grafiche quali la MPS 803, Star SG-10, Star NL-10 ecc., si effettua in genere la hardcopy della pagina grafica con il metodo bit-mapping; e così, che vengono stampate le schermate hi-res. Purtroppo tale metodo, almeno con le stampanti sopra indicate, presenta un difetto: la defor-

mazione del disegno rispetto a quello che appare sullo schermo. NL Printgraf è una utility in linguaggio macchina, progettata per la Star NL-10, che riproduce fedelmente e senza deformazioni la pagina grafica localizzata in una qualunque zona di memoria.

Una schermata grafica hi-res consiste in una matrice di 320×200 pixel, che in prima approssimazione possiamo considerare simmetrici (cioè la distanza orizzontale fra i pixel è uguale a quella verticale). Il rapporto fra i lati orizzontale e verticale dello schermo è quindi proprio di $320/200$, cioè 1,6.

Nella stampa bit-mapping, a densità normale, vengono stampate 60 colonne per pollice, e quindi la distanza fra le colonne è di $1/60$ di pollice; per contro, la distanza fra le singole righe (cioè fra gli aghi della stampante) è di $1/72$ di pollice. Ne consegue che una matrice di 320×200 punti non avrà un rapporto orizzontale/verticale uguale a quello dello schermo, ma superiore di circa il 20%, e risulterà allungata in senso orizzontale: così avremo un'ellisse al posto di un cerchio, un rettangolo al posto di un quadrato e via dicendo.

NL-PRINTGRAF

```

5 REM ** NL-PRINTGRAF - HARDCOPY PAGINA GRAFICA SU STAR NL-10 **
6 REM ** BY R. MORASSI- PISTOIA **
7
10 DATA 032,241,183,134,252,169,000,133
20 DATA 251,133,183,162,003,157,220,204
30 DATA 202,016,250,032,241,183,138,240
40 DATA 002,169,255,133,248,160,000,169
50 DATA 004,170,032,186,255,032,192,255
60 DATA 162,004,032,201,255,169,125,133
70 DATA 034,169,203,133,035,169,014,032
80 DATA 036,171,169,025,133,250,169,139
90 DATA 133,034,169,203,133,035,169,005
100 DATA 032,036,171,169,040,133,249,169
110 DATA 156,133,253,169,203,133,254,120
120 DATA 169,252,037,001,133,001,160,007
130 DATA 177,251,153,148,203,136,016,248
140 DATA 160,007,162,007,094,148,203,106
150 DATA 202,016,249,069,248,145,253,136
160 DATA 016,240,024,165,251,105,008,133
170 DATA 251,144,002,230,252,024,165,253
180 DATA 105,008,133,253,144,002,230,254
190 DATA 198,249,208,202,169,003,005,001
200 DATA 133,001,088,169,156,133,253,169
210 DATA 203,133,254,169,054,133,249,169
220 DATA 139,032,210,255,169,000,133,255
230 DATA 162,011,160,000,165,255,073,001
240 DATA 133,255,240,003,177,253,200,032
250 DATA 210,255,202,208,239,024,165,253
260 DATA 105,006,133,253,144,002,230,254
270 DATA 198,249,208,211,169,032,133,249
280 DATA 230,249,165,249,072,032,210,255
290 DATA 104,201,086,208,243,169,013,032
300 DATA 210,255,173,141,002,208,007,198
310 DATA 250,240,003,076,180,202,169,144
320 DATA 160,203,032,030,171,169,004,032
330 DATA 195,255,032,204,255,096,000,013
340 DATA 027,097,001,027,033,001,027,065
350 DATA 008,027,037,049,000,027,038,000
360 DATA 033,086,027,064,013,000
370
380 FORX=0TO285:READY:POKE51830+X,Y:A=A+Y:NEXT
390 IFA<>38433THENPRINT"DATA ERROR"
400 END
410
500 POKE43,118:POKE44,202:POKE45,148:POKE46,203
510 SAVE"NL-PRINTGRAF/M".8
520 SYS64738

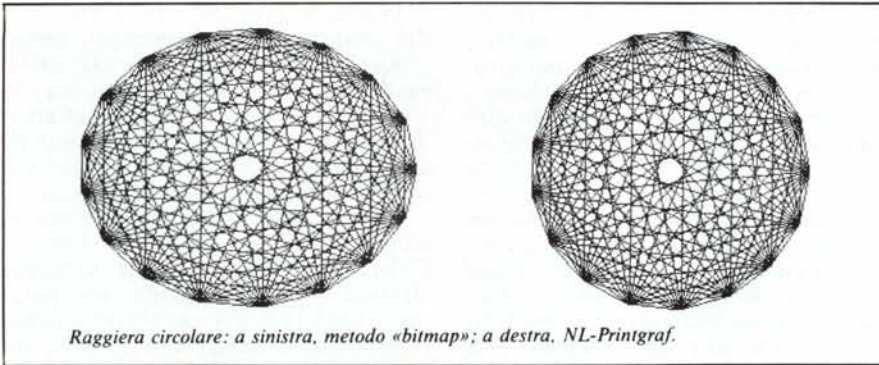
```



Immagine stampata col metodo «bitmap».



La stessa con NL-Printgraf.



Raggiera circolare: a sinistra, metodo «bitmap»; a destra, NL-Printgraf.

In NL-Printgraf ho usato il metodo «download characters» (caratteri ridefiniti) anziché il bit-mapping. Ciascuna «riga» di schermo, delle dimensioni di 320×8 pixel, viene copiata in un buffer di 320 byte, previa diagonalizzazione (lettura per colonne anziché per righe) di ciascuna matrice elementare di 8×8 pixel. Vengono poi aggiunti 4 byte uguali a zero, e con i 324 byte si costruiscono 54 caratteri ridefiniti (larghi 6 colonne ciascuno) che vengono inviati, con le opportune interpolazioni, alla memoria RAM della stampante (caratteri ASCII da 33 a 86) dopo avere attivato il «download mode» e un'interlinea di $8/72$ di pollice. I 54 caratteri vengono infine stampati, e l'operazione si ripete fino ad esaurire le 25 righe di schermo. Il ridimensionamento si ottiene stampando in modo «elite» con un «pitch» orizzontale di 12 caratteri per pollice, corrispondente a $6 \times 12 = 72$ colonne: la distanza fra le colonne risulta quindi

di $1/72$ di pollice, e cioè uguale a quella fra le righe.

Digitate NL-Printgraf, salvatelo e date il RUN. Se i DATA sono stati introdotti correttamente, senza segnalazione di errore, date poi RUN 500 per salvare il file in linguaggio macchina col nome «NL-Printgraf/M». Quando vorrete ricaricarlo in memoria, digitate semplicemente LOAD «NL-Printgraf/M», 8,1, seguito da NEW.

Il programma è allocato in memoria da \$CA76 e \$CB93 (dec. 51830-52115), e non è rilocabile senza modifiche. Il buffer per costruire la riga di stampa va da \$CB94 a \$CCDF. Dopo aver disegnato (o caricato da supporto magnetico) la pagina grafica, resettate il computer se necessario (usando un tasto di reset) e caricate NL-Printgraf/M. La sintassi da usare per la stampa è:

SYS 51830, P, N

dove i parametri hanno il seguente significato:

P = numero della pagina di memoria a cui iniziano gli 8 Kbyte della pagina grafica. Ecco ad esempio i valori da usare per alcune popolari utility o linguaggi:

SIMON'S BASIC: 224
KOALAPainter, PRINT SHOP
(Screen Magic), DOODLE:96
LOGO, ART STUDIO: 32
SCREEN GRAPHICS: 160

Come si vede, NL-Printgraf legge e stampa anche le pagine grafiche nascoste nella RAM «sotto» la ROM del Sistema Operativo.

N = modo di stampa. Con $N = 0$ si ha la stampa normale, con $N > 0$ quella invertita (vengono cioè stampati i punti che sullo schermo sono spenti, anziché quelli accesi). Quest'ultimo modo è comunque sconsigliabile a meno che lo schermo non sia esso stesso in «reverse», perché consuma molto il nastro e tende a surriscaldare la testina di stampa.

Nelle figure sono mostrati, per confronto, alcuni disegni stampati col metodo bit-mapping e con NL-Printgraf. Se si vuole una stampa più scura, si può attivare il «boldface» digitando POKE 52096, 17 (per riavere la stampa normale, POKE 52096, 1). Per interrompere una stampa in corso, infine, basta tenere premuto il tasto <SHIFT>.

Vita 3D

di Flavio Rizzardi
Sarmeola Di Rubano (PD)

Proprio quando sembrava che non ci fosse più niente da dire sul gioco Vita, scritto da Hector Conway e fatto conoscere da MCmicrocomputer e dagli articoli di Dewdney apparsi su «Le Scienze», ecco che arriva il cugino tridimensionale di Vita, creato da Carter Bays.

Per i pochi sfortunati mortali che non hanno idea di cosa significhi simulare con un calcolatore lo sviluppo di una colonia di cellule, sarà opportuno riassumere quanto ha scritto il già nominato Dewdney nell'ultimo numero di Le Scienze: «Il gioco di Conway si svolge su una griglia bidimensionale infinita di cellule quadrate. Ciascuna cellula ha otto adiacenti e può trovarsi nell'uno o nell'altro di due stati, viva o

morta. Da qualche parte un grande orologio segna lo scorrere del tempo e ad ogni istante certe cellule possono diventare vive ed altre morire; il destino di ogni cellula è determinato dal numero di adiacenti vivi. Se, per esempio, ad un certo istante una cellula viva ha meno di due adiacenti vivi o più di tre, all'istante successivo sarà morta; una cellula morta, invece, sarà rinata all'istante successivo se ha esattamente tre adiacenti vivi».

Nello spazio di Vita 3-d ogni cellula (non più quadrata ma sferica) viene ad avere ben 26 adiacenti, ma per il resto le regole di Conway non cambiano di molto.

Arriviamo ora alla versione scritta da me per il 64 (premetto che ho impiegato solo una settimana per scrivere il programma in Assembler, per cui se vi venisse l'idea di esaminarlo munitevi di almeno dieci compresse di Cibalgina — non sono mai stato molto ordinato, e la fretta non aiuta granché...).

Il programma occupa la zona di memoria da C000 e CFFF, e parte con SYS 49152. Dopo averlo fatto partire, chiede la versione di Vita alla quale vogliamo «giocare»: si cancella lo schermo e nell'angolo in alto a sinistra appare la scritta «VITA»; bisogna a questo punto inserire il codice desiderato (sempre di quattro cifre) seguendo la notazione di Bays.

Per chi non avesse letto l'ultimo numero di (RI)Creazioni al calcolatore, do la seguente spiegazione: ogni versione tridimensionale di Vita è identificata da un numero di quattro cifre, ad esempio 0191. La prima cifra indica il minimo di adiacenti vivi che una cellula viva deve avere per non essere «denutrita»; la seconda è il massimo di adiacenti oltre il quale si va incontro a soffocamento. Le altre due cifre indicano rispettivamente il numero minimo ed il massimo di genitori che una cellula morta deve avere per nascere. La versione preferita da Bays ha il codice 4555, ed è per questa versio-

ne che sono preparati i numerosi esempi riportati nell'articolo di Dewdney. Non appena si è inserita l'ultima cifra del codice, si passa al vivo del programma; nell'angolo in alto a destra appare la scritta «GEN. 000»: è il cronometro dell'Universo cellulare che, come si può vedere, è fermo alla generazione zero. Veniamo ora ai comandi, che sono veramente molti...

Con il tasto F1 si cancella lo schermo, che all'inizio è occupato da otto cellule disposte ai vertici del cubo-Universo. Per evitare una cancellazione involontaria, il tasto deve essere tenuto premuto finché il bordo dello schermo non smette di lampeggiare. Quando si fa partire il cronometro, F1 è disabilitato.

Oltre alle otto cellule ed al cronometro, all'inizio si può vedere sullo schermo anche uno strano cursore multiplo, costituito da una sfera bianca e dalle proiezioni di detta sfera su tre facce del cubo-Universo. Quando la sfera si sposta si spostano anche le sue tre ombre, permettendo di identificare con precisione la posizione del cursore nello spazio. I tasti che muovono il cursore sferico sono i seguenti: T-Y per lo spostamento orizzontale; F-V per lo spostamento verticale; R-G per quello in profondità. All'inizio i più inesperti correranno il grave rischio di annodarsi le dita, ma con la costanza si faranno grandi progressi...

Per chi vuole il listato

Il listato di questo programma è molto lungo. In conseguenza di ciò, si è ritenuto opportuno non pubblicarlo, sia perché avrebbe occupato troppo spazio sulla rivista sottraendone ad altri argomenti, sia perché una digitazione senza errori di un listato così lungo appare poco probabile. Chi è interessato al programma può ordinare secondo il solito sistema, il disco o la cassetta in redazione. È anche possibile «pescare» direttamente (e gratuitamente) il programma per via telematica, dal nostro servizio MC-Link; questo ovviamente vale per chi è attrezzato in tal senso. Ricordiamo che per ottenere una casella su MC-Link è sufficiente telefonare (con un modem e un programma di comunicazione) al numero 06/4510211.

Questo programma è disponibile su disco presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 195.

Per depositare nell'Universo una cellula viva, si preme il tasto + mentre, per uccidere una cellula nata per errore, si usa il tasto - (e qui non stiamo a discutere sulla moralità di questa soppressione...). Uno dei più grandi inconvenienti di Vita 3-d è il fatto che le cellule che stanno in fondo possono essere coperte dalle altre, ma nella mia versione il cubo-Universo può essere fatto «scorrere», cosicché «gli ultimi piani saranno i primi». I tasti per fare scorrere orizzontalmente il cubo-Universo sono W ed E, in verticale A e Z, in profondità Q ed S. Qualora non l'avessi detto prima, anche l'Universo cubico è chiuso su se stesso come quello bidimensionale: ogni cosa che esce dall'alto riappare in basso ecc... Per osservare il cubo sotto tutti i punti di vista si può anche compiere una rotazione sull'asse X (con il tasto M) o sull'asse Y (tasto U) o sull'asse Z (tasto O).

Una delle particolarità più importanti di questa versione di Vita è il fatto che, premendo i tasti da 0 a 9, si costruiscono sotto il cursore alcune configurazioni predefinite di cellule, che guarda caso sono le stesse nominate nell'articolo di Dewdney. Queste configurazioni funzionano solo con la versione 4555 di Vita, e sono per la precisione il «pedistallo» (tasto 1), la «croce» (2), il «gradino» (3), la «palla» (4), il «riflettore» (5), il «rotore» (6), il «cavallino rampante» (7) e l'«aliante» (che si ottiene con i tasti 8,9 o 0). Il pedistallo, la croce, il gradino e la palla sono stabili (cioè non mutano con il passare del tempo); il rotore si espande e si contrae a generazioni alternate; il cavallino ed il riflettore tornano alla forma originaria dopo quattro generazioni e l'aliante, sempre dopo quattro generazioni, si sposta in avanti di una casella per poi ricominciare da capo. Un aliante può così attraversare lo schermo da una parte all'altra e scontrarsi con strutture stabili (stile kamikaze) facendole esplodere. Una delle collisioni più curiose è quella con il riflettore, una struttura che ho inventato io, poiché dopo lo scontro l'aliante viene deviato di 90° rispetto alla sua traiettoria originaria.

Gli ultimi comandi sono il tasto «freccia a sinistra» che fa ritornare il controllo al Basic, nonché F3 che fa partire il cronometro del cubo-Universo dando vita e movimento alle configurazioni. Mentre funziona il cronometro tutti i tasti sono disabilitati; per fermarlo bisogna premere il tasto F7 e tenerlo premuto finché non riappare il cursore multiplo (alle volte bisogna attendere un bel po' di secondi) dopodiché si potrà provvedere a modificare qualche cellula qua e là, oppure a ruotare il cubo per portare alla luce i lati nascosti.

Prima di passare alla descrizione del programma in Assembler, consiglieri agli amanti dei caleidoscopi di mettere al centro del cubo-Universo in Vita 0919 un cubetto di otto cellule e di far partire il cronometro: alcune figure che si svilupperanno da quel cubetto iniziale sono veramente belle...

Il programma in linguaggio macchina va, come già detto, da C000 a CFFF, ma le mappe che vengono sfruttate dal programma occupano quasi tutta la RAM del nostro piccolo Commodore: tralasciando i byte da 0400 a 07FF e quelli da D800 a DBFF, che servono per i colori (l'output, come è facile notare, è in alta risoluzione multicolor) le mappe sono le seguenti: da 1000 a 1FFF c'è la matrice in tre dimensioni del cubo-Universo; da 2000 a 3FFF ci sono gli 8K della pagina grafica; da 4000 a 4FFF trovano posto i byte bassi della posizione di ogni cellula nella pagina grafica, e da 5000 a 5FFF ci sono i relativi byte alti; da 6000 a 7FFF ci sono altri 8K per una pagina grafica virtuale (l'immagine si forma qui lentamente e viene trasferita poi di colpo a 2000/3FFF per evitare una confusione paurosa, con cellule che compaiono e scompaiono troppo lentamente). Gli otto K da 8000 a 9FFF erano occupati dal monitor, per cui ho dovuto lasciarli liberi.

La mappa in tre dimensioni è organizzata nel seguente modo: se nella cella 1000 è contenuto il valore 01, significa che la cellula di coordinate X, Y e Z pari a 0F è viva (si tenga come origine degli assi la posizione del cursore all'inizio, e come asse delle X quello inclinato di 45°). Lo stato della cellula che all'inizio è sotto il cursore è determinato dal contenuto della cellula 1FFF, e in base a questo si può risalire alla posizione di tutte le cellule nella matrice. Quando comincia a lavorare il ciclo che determina lo stato di una cellula alla generazione successiva, il risultato dei calcoli (cioè il destino di ciascuna cellula) viene depositato nel bit 1 della locazione corrispondente.

Prima di ripetere il ciclo per la generazione successiva, un LSR trasforma quelle che Dewdney chiamava Nuovecellule in semplici Cellule.

Il procedimento è macchinoso e porta via un sacco di tempo, ma non avevo altri 4K liberi per la matrice Nuovecellule, ed in ogni caso anche così la velocità di elaborazione risulta accettabile.

Ricapitolando, la mappa 3-d può contenere i seguenti valori:

00: la cellula è morta e resterà morta anche alla generazione successiva.

01: la cellula è viva ma morirà all'istante successivo.

02: la cellula è morta ma rinascerà alla generazione successiva.



Una stessa generazione vista da diverse angolazioni.

03: la cellula è viva e lo sarà anche all'istante successivo.

Una particolare subroutine provvederà poi ad esaminare l'intera matrice ed a trasformare una sfilza di numeri in un ammasso di sferette azzurre. Per spiegare come faccia la routine a sapere esattamente in quale punto dello schermo deve piazzare la cellula, bisogna dare un'occhiata alle mappe da 4000 a 5FFF. Il contenuto di queste mappe è determinato una volta per tutte all'inizio del programma, quando un ciclo va a prelevare il contenuto di tre celle da una mini-mappa da 96 byte e li somma tra loro in tutte le possibili combinazioni. Alla fine del ciclo, in 40FF si trova il valore 00 e in 50FF c'è il valore 60 questo significa che la locazione dalla quale bisogna iniziare a disegnare la cellula mappata in 10FF è 6000.

Il ciclo che disegna una cellula è quello che rallenta maggiormente l'esecuzione del programma: una volta stabilito il punto da cui iniziare a disegnare si mascherano con degli AND i 32 byte destinati ad accogliere l'immagine della cellula (accorgimento necessario in quanto siamo in multicolor mode) e poi si trascrivono con degli OR i byte che definiscono la forma della cellula stessa. Per un inconveniente del ciclo che determina la posizione di ogni cellula nella mappa grafica, ho dovuto ripetere tutto quanto detto sopra per due: i blocchi di 32 byte che definiscono l'immagine di una

cellula sono infatti due, e nel secondo la figura della sferetta è spostata di quattro pixel in diagonale rispetto al primo.

Per quanto riguarda le mappe non c'è molto altro da dire: da CE00 a CEFF sono definiti i pattern per i quattro cursori, e da CDB0 a CDFE si trovano i codici per disegnare le cifre in alta risoluzione.

Il programma, che come ho detto inizia a C000, salta subito a CD34 perché ho dovuto fare un'aggiunta fuori programma, ma andiamo con ordine...

Da C003 a C050 prepara lo schermo in alta risoluzione.

Da C051 a C148 c'è il ciclo che prepara le mappe 4000/5FFF.

Da C274 a C2C3 si piazzano le otto cellule iniziali e si stampa «GEN. 000».

Da C2C6 a C348 si preparano gli sprite.

Da C349 a C4D8 c'è il ciclo principale per controllare la tastiera, per muovere il cursore e per mettere le cellule a posto. Se qualcuno crede che l'autorepeat della tastiera sia troppo rapido o troppo lento, non occorre essere dei maghi per cambiare il valore dell'LDY nel ciclo che genera il ritardo.

Da C51D a C882 ci sono alcune vangelhe di INC e di DEC che definiscono le configurazioni di base quali l'aliante o il rotore.

Da C891 a C99F si fa scorrere l'Universo nelle sei direzioni.

Da C9A2 a C9FD si prepara il ciclo principale di Vita.

Da CA00 a CAB6 si decidono i destini delle cellule...

Da CAB9 a CAE8 si controlla se una cellula è denutrita o sta soffocando.

Da CB9F a CCC4 l'Universo ruota sui tre assi.

Da CCB2 a CD33 una particolare routine, aggiunta all'ultimo momento, annulla una distorsione nell'Universo che si veniva a creare se una cellula passava dal lato destro al lato sinistro o viceversa.

Le principali subroutine sono le seguenti:

JSR C584 prepara delle locazioni per le configurazioni predefinite.

JSR C59E mette una cellula nella matrice 3-d (per configurazioni predefinite).

JSR C227 stampa il contenuto della matrice 3-d sullo schermo.

JSR C8C0 trasforma Nuovecellule in Cellule (cambia generazione).

JSR CB49 stampa il numero di generazioni (contenuto in CF18-CF1A).

JSR CB8B incrementa di una unità il contatore delle generazioni.

Tanto per chiarire ulteriormente le cose, aggiungono che nel ciclo principale la posizione del cursore nella mappa 3-d è contenuta nelle locazioni 61 e 62 della pagina zero; nel ciclo Vita vengono usate sei locazioni in pagina zero (da 61 a 66) per controllare lo stato degli adiacenti; il risultato di questo conteggio viene poi depositato nelle locazioni da 0800 a 081A.

Non pretendo affatto che la routine che ho scritto per contare il numero degli adiacenti vivi sia la migliore o la più rapida: dovendo controllare un cubo di tre cellule di lato ho preferito sezionarlo in tre quadrati di 3×3 e far controllare ciascuno di questi quadrati da una coppia di locazioni in pagina zero. Le locazioni 61-62 puntano ad esempio alla cellula centrale del primo quadrato di nove cellule; queste cellule verranno esaminate poi singolarmente utilizzando quello che credo si chiami «indirizzamento postindiretto indiretto», per dirla in parole povere LDA (61), Y, dove Y assume di volta in volta il valore EF, F0, F1 FF, 00, 01, 0F, 10 ed 11.

Prima di chiudere aggiungo un piccolo suggerimento: per accelerare la routine che controlla gli adiacenti si possono disabilitare le interruzioni, anche se bisogna stare attenti a riabilitarle prima del LDA CB che controlla la pressione di un tasto. 