

software

C-64

a cura di Tommaso Pantuso

STRANGE BASIC

di Fabrizio Fantoccoli - Arona (NO)
Gianluca Finistauri
Sesto Calende (VA)

Presentazione dello Strange Basic

Il programma Strange Basic è stato creato per darvi la possibilità di accedere a tutta la potenza del vostro computer Commodore 64. Ciò è reso possibile grazie ad un set di 51 nuovi comandi Basic supplementari che si aggiungono a quelli del Basic standard. Questi nuovi comandi appartengono a otto diversi gruppi qui di seguito elencati e descritti:

comandi di AUSILIO ALLA PROGRAMMAZIONE, come ad esempio JJUMP e JFKEY, che vi permettono di programmare in Basic in modo più veloce ed efficace;

comandi per la GESTIONE DEI TESTI, come ad esempio JSCL e JBACK, che vi permettono di ottenere particolari effetti nell'output del testo;

comandi di CONVERSIONE NUMERICA, che vi permettono un migliore uso della memoria in base binaria, esadecimale e decimale;

comandi per la GESTIONE DELLO SCHERMO, come ad esempio JRUST per il trattamento della scansione dell'immagine;

comandi per la GRAFICA, come JCBOX e JCHAR, che vi permettono di produrre facilmente disegni e scritte sul video;

comandi MUSICALI, come ad esempio JPLAY e JFILT, che vi permettono di comporre musiche e creare stupendi effetti sonori;

comandi per EFFETTI SPECIALI per creare particolari immagini ed effetti;

comandi per la VELOCIZZAZIONE DEL REGISTRATORE.

Particolari e convenzioni dello Strange Basic

Per facilitare l'uso, i comandi dello Strange Basic iniziano tutti con la lettera «J» e sono formati da cinque caratteri. I parametri da introdurre sono composti da numeri o da stringhe alfanumeriche e sono tra loro divisi da virgole. L'omissione o l'errata disposizione di tali parametri darà origine ad un «SINTAX ERROR».

Tutti i comandi possono essere usati sia in modo diretto che da programma.

Le convenzioni usate nella seguente descrizione sono:

1) i termini scritti a lettere maiuscole devono essere battuti esattamente come sono presentati;

2) i termini scritti a lettere minuscole o tra virgolette indicano rispettivamente le variabili numeriche e alfanumeriche;

3) la scritta [RETURN] significa che occorre premere il tasto RETURN;

4) tutti i testi scritti tra parentesi indicano la pressione del tasto corrispondente, ad esempio [CLR/HOME], premendo il quale il cursore si porta nell'angolo superiore sinistro.

Ausili alla programmazione

Lo Strange Basic mette a vostra disposizione alcuni comandi che semplificano e velocizzano la programmazione in Basic del vostro Commodore 64:

— JPAUS n

esegue una pausa la cui lunghezza è determinata dal parametro numerico 'n'; se 'n' = 728 la pausa è di un secondo. Premendo [RETURN] si interrompe la pausa.

— JJUMP n

esegue un goto calcolato dove 'n' è una qualsiasi espressione numerica.

— JOLDD

recupera un programma cancellato con NEW.

— JLOCA a,b,c

trasferisce la zona di memoria compresa tra 'a' e 'b' a partire da 'c'.

— JBYTE b,n,t

se 't' = 0 questo comando spegne il BIT 'n' del BYTE 'b';

se 't' = 1 questo comando accende il BIT 'n' del BYTE 'b'.

— JPSAV «nome programma», d,l,i,f
salva una zona di memoria sulla periferica 'd' con indirizzo secondario 'l' compresa tra 'i' ed 'f'.

— JPLDA «nome programma», d,l,i
carica un programma ponendolo a partire da 'i'; 'l' deve essere uguale a 0.

— JPVER «nome programma», d,l,i
verifica una zona di memoria a partire da 'i'.

— JINVT a,b

inverte la zona di memoria compresa tra 'a' e 'b'. (EORFF, immagine speculare del BYTE).

— JFILL a,b,c

riempie la zona di memoria compresa tra 'a' e 'b' con il valore di 'c'.

— JPUSH n

attende la pressione di un tasto il cui codice ASCII è uguale ad 'n'.

Se 'n' = 0 aspetta la pressione di un tasto qualsiasi.

— JDATI l

sposta il puntatore delle linee DATA a partire dalla linea 'l'.

— JFKEY n

se 'n' = 1 attiva i seguenti tasti funzione:

F1 = RUN + RETURN

F3 = LIST + RETURN

F5 = LOAD "\$", 8 + RETURN

F7 = SAVE

se 'n' è diverso da 1 i tasti funzione vengono disattivati.

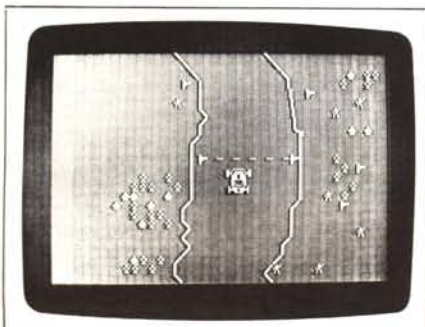
— JCOMN

lista tutti i comandi aggiunti dallo Strange Basic.

Per chi vuole il listato

I listati dei due programmi sono molto lunghi. In conseguenza di ciò, si è ritenuto opportuno non pubblicarli, sia perché avrebbero occupato troppo spazio sulla rivista sottraendone ad altri argomenti, sia perché una digitazione senza errori di listati così lunghi appare poco probabile. Chi è interessato ai programmi può ordinare secondo il solito sistema, il disco o la cassetta in redazione. È anche possibile «pescare» direttamente (e gratuitamente) i programmi 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.

Questi due programmi sono disponibili su disco presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 229.



— JCOLD
 esegue una partenza a freddo; simile a SYS 64738, ma senza uscire dallo Strange Basic.

Gestione dei testi

Questi comandi permettono di migliorare al massimo l'output del testo, ottimizzando la scelta della posizione delle scritte e dei dati sul video.

— JSCRR v,r,n, «parole da introdurre»
 fa scorrere a destra una riga dello schermo compresa tra 0-24, specificata nella variabile 'r', di un certo numero di caratteri 'n', introducendo la parola tra le virgolette. La velocità è specificata in 'v'.

— JSCRL v,r,n, «parole da introdurre»
 è identico al comando precedente, tranne la direzione dello scroll che avviene verso la sinistra.

— JSCRU v,c,n, «parole da introdurre»
 fa scorrere verso l'alto una colonna dello schermo compresa tra 0-39, specificata nella variabile 'c', di un certo numero di caratteri 'n', introducendo la parola tra le virgolette. La velocità è specificata in 'v'.

— JSCRD v,c,n, «parole da introdurre»
 è identico al comando precedente, tranne la direzione dello scroll che avviene verso il basso.

— JSTAI x,y
 posiziona il cursore sullo schermo nella posizione specificata in 'x' ed 'y'.

— JBACK v,n,t,r
 fa scorrere tutti i caratteri presenti sullo schermo (in bassa risoluzione) dove:

'v' è la velocità compresa tra 0-255;
 'n' è il numero di riga o di colonna;
 't' è il tipo di scroll: 't' = 0 scroll a destra; 't' = 1 scroll a sinistra; 't' = 2 scroll alto; 't' = 3 scroll basso.

Se 't' = 3 lo scroll è verso il basso, ma la prima riga di schermo rimane invariata.

'r' indica la locazione d'inizio che contiene i codici dei caratteri da inserire con lo scroll; se 'r' = 0 non viene introdotto alcun carattere.

— JBLNK n
 se 'n' = 1 predispone il computer a operare in stato di BLANK;

se 'n' = 0 ritorna al normale funzionamento.

Nota:
 il computer quando è in stato di BLANK lavora un poco più velocemente, essendo esclusa la gestione dello schermo.

— JEXTN a,b,c
 mette i caratteri nel modo a colore fondo esteso.

'a' contiene il colore dei caratteri battuti con le SHIFT

'b' contiene il colore dei caratteri in REVERSE

'c' contiene il colore dei caratteri in REVERSE + SHIFT.

— JCOLR b,f
 colora il bordo con il colore contenuto in 'b' e lo sfondo con il colore contenuto in 'f'.

Conversione numerica

Questi comandi permettono di convertire i valori numerici nei vari codici per un migliore utilizzo della memoria del vostro

CBM 64, ad esempio per programmare la memoria carattere o per creare sprite.

— JSHOW n
 questo comando mostra il contenuto del BYTE 'n' in codice BINARIO, ESADECIMALE e DECIMALE.

— JINBY b,a\$

immette nei byte a partire da 'b' i valori in codice binario indicati nella variabile 'a\$'. Questa variabile deve contenere i caratteri 0 e 1 (bit spento od acceso) e deve essere lunga otto caratteri o un multiplo di otto, a seconda se vogliamo definire solo il byte 'b' o anche quelli successivi.

GENERALITÀ SUI COMANDI AGGIUNTIVI DELLO STRANGE BASIC

| N. | NOME | INDIRIZZO DI PARTENZA | DESCRIZIONE |
|----|-------|-----------------------|---|
| 1 | JSCRR | \$9870 | Fa scorrere a destra una linea di N caratteri introducendo un eventuale testo |
| 2 | JSCRL | \$9860 | Fa scorrere a sinistra una linea di N caratteri introducendo un eventuale testo |
| 3 | JSCRU | \$9A4A | Fa scorrere verso l'alto una linea di N caratteri introducendo un eventuale testo |
| 4 | JSCDW | \$9AD5 | Fa scorrere verso il basso una linea di N caratteri introducendo un eventuale testo |
| 5 | JSTAI | \$9AEA | Sposta il cursore nella pagina a bassa risoluzione ad una data coordinata |
| 6 | JBACK | \$9D00 | Fa scorrere tutti i caratteri della pagina a bassa risoluzione nei quattro sensi |
| 7 | JPAUS | \$9DEE | Esegue una pausa di N/278 secondi |
| 8 | JSHOW | \$9E20 | Mostra il contenuto di un byte in codice binario, esadecimale e decimale |
| 9 | JJUMP | \$9E5B | Esegue un GOTO «calcolato», si può introdurre una funzione per il numero di linea |
| 10 | JOLDD | \$9E68 | Recupera un programma Basic se cancellato con NEW o JCOLD |
| 11 | JLOCA | \$9295 | Copia una zona della memoria da un punto all'altro |
| 12 | JBYTE | \$92FC | Permette di agire sui BIT di ogni byte senza usare calcoli logici |
| 13 | JMEMC | \$93BC | Trasferisce la memoria carattere nella RAM rendendola programmabile |
| 14 | JINBY | \$93D3 | Immette in una serie di byte un valore in codice binario |
| 15 | JBLNK | \$942A | Mette in blank il video |
| 16 | JEXTN | \$9450 | Permette la possibilità di usare caratteri multicolor |
| 17 | JCOLR | \$9483 | Inserisce i colori di schermo e sfondo |
| 18 | JPSAV | \$9497 | Salva una determinata porzione di memoria |
| 19 | JPLDA | \$94BF | Carica una determinata porzione di memoria |
| 20 | JPVER | \$94B8 | Verifica una determinata porzione di memoria |
| 21 | JINVT | \$94FE | Inverte una zona di memoria eseguendo un EOR con \$FF per ogni byte |
| 22 | JFILL | \$9520 | Riempie una zona di memoria con un valore prefissato |
| 23 | JHIRS | \$95FD | Cambia nel modo ad alta risoluzione normale (320*200 pixel) |
| 24 | JLOWR | \$95B5 | Torna al modo di bassa risoluzione grafica |
| 25 | JHICL | \$95E0 | Pulisce lo schermo in alta risoluzione grafica |
| 26 | JMCOL | \$9614 | Cambia nel modo ad alta risoluzione multicolor (160*200 pixel, 4 colori) |
| 27 | JPUSH | \$9657 | Blocca un programma in esecuzione fino alla pressione di un determinato tasto |
| 28 | JPLOT | \$C000 | Traccia un punto in alta risoluzione normale |
| 29 | JCPLT | \$C09A | Traccia un punto in alta risoluzione multicolor |
| 30 | JBOXY | \$C26D | Disegna un rettangolo pieno in alta risoluzione normale |
| 31 | JCBOX | \$C275 | Disegna un rettangolo pieno in alta ris. multicolor |
| 32 | JNREC | \$C330 | Disegna un rettangolo vuoto in alta ris. normale |
| 33 | JCREC | \$C3B6 | Disegna un rettangolo vuoto in alta ris. multicolor |
| 34 | JCHAR | \$C3FA | Scriva un carattere qualunque in alta ris. multicolor o normale |
| 35 | J-SAV | \$C542 | Salva un programma su cassetta con TURBO-TAPE (10 volte più veloce) |
| 36 | J-LDA | \$C549 | Carica un programma su cassetta con TURBO-TAPE (10 volte più veloce) |
| 37 | J-VER | \$C54F | Verifica un programma su cassette con TURBO-TAPE (10 volte più veloce) |
| 38 | JDFSP | \$C86A | Definisce e attiva uno SPRITE servendosi di una zona di memoria a piacere |
| 39 | JSPOF | \$C920 | Disattiva uno SPRITE |
| 40 | JROTA | \$C945 | Attiva la rotazione continua di un carattere in una delle quattro direzioni |
| 41 | JFERM | \$C9B7 | Disattiva la rotazione di un carattere, attivata con il comando JROTA |
| 42 | JADSR | \$CAB9 | Predispone i valori di inviluppo per una voce del SID |
| 43 | JFILT | \$CB4D | Predispone e attiva i filtri di una voce del SID |
| 44 | JPLAY | \$999A | Suona una nota con una voce tenendo conto dei valori fissati con JADSR e JFILT |
| 45 | JMSOF | \$C555 | Disattiva una voce |
| 46 | JDATI | \$C58A | Sposta il puntatore ad una certa linea per la lettura dei «DATA» |
| 47 | JRUST | \$CC25 | Divide lo schermo in tre parti con diverse caratteristiche agendo sulla scansione |
| 48 | JRSOF | \$CD40 | Disattiva il comando JRUST riportando il video alle normali condizioni |
| 49 | JFKEY | \$CE15 | Attiva quattro tasti funzione con utili comandi |
| 50 | JCOMN | \$CE40 | Elenca i comandi aggiuntivi dello STRANGE BASIC |
| 51 | JCOLD | \$97B5 | Esegue una partenza a freddo dello STRANGE BASIC |
| 52 | JRAMC | \$9A27 | Definisce i caratteri con ciò che vi è in memoria RAM a partire da \$3000 |

Gestione dello schermo

Questi comandi permettono di controllare la scansione dell'immagine suddividendo lo schermo in più zone con diverse caratteristiche (colore, risoluzione...).

JRUST l1,l2,l3,b1,b2,b3,s1,s2,s3,m1,m2,m3 agisce sulla scansione dell'immagine dividendo lo schermo in tre parti:

la prima parte va da 'l3' a 'l1' (colore di bordo 'b1', colore di sfondo 's1', modo 'm1');

la seconda parte va da 'l2' a 'l1' (colore di bordo 'b2', colore di sfondo 's2', modo 'm2');

la terza parte va da 'l2' a 'l3' (colore di bordo 'b3', colore di sfondo 's3', modo 'm3');

Modo: 0 = normale; 1 = alta risoluzione; 2 = alta risoluzione multicolor.

Distanziare le linee l1,l2,l3 di almeno 40 linee.

— JRJOF

disattiva il comando JRUST.

Grafica

Questi sono potenti comandi grafici che vi permettono di realizzare in modo molto semplice dei disegni in alta risoluzione.

— JHIREs a,b

Mette in alta risoluzione non multicolor, dove 'a' è il colore dello sfondo e 'b' è quello dei pixel (del disegno).

— JLOWR

Torna in bassa risoluzione.

— JHICL

Pulisce lo schermo in alta risoluzione.

— JMCOL a,b,c

Mette in alta risoluzione multicolor con i colori specificati in 'a', 'b', 'c'.

Si può tranquillamente passare da JHIRS a JMCOL e viceversa.

— JPLOT x,y,c

Se c = 0 spegne il punto di coordinate x,y in alta risoluzione non multicolor.

Se c = 1 accende il punto.

— JCPLT x,y,c

Accende un punto in alta risoluzione multicolor con c = 1,2,3 che corrispondono al colore.

Se c = 0 lo spegne.

— JBOXY x,y,lx,ly,t

Disegna un rettangolo pieno in alta risoluzione normale, che comincia dal punto x,y (angoli superiori sinistri) ed ha il lato orizzontale lungo 'lx' e quello verticale 'ly'. 't' è il colore (t = 0 spegne, t = 1 accende).

— JCBOY x,y,lx,ly,t

Come il comando precedente ma in alta risoluzione multicolor.

('t' = 0 spegne, t = 1,2,3 accende con il colore corrispondente).

— JNREC x,y,lx,ly,t

Disegna un rettangolo in alta risoluzione normale.

I parametri sono gli stessi di JBOXY.

— JCREC x,y,lx,ly,t

Come il precedente, ma in alta risoluzione multicolor. I parametri sono gli stessi di JBOXY.

— JCHAR x,y,co,t,m

Permette di scrivere un carattere in alta risoluzione (m=0 alta risoluzione normale, m=1 alta risoluzione multicolor), a partire dalle coordinate x,y con 'co' che è il codice

schermo del carattere e 't' che è il colore del carattere.

— JDFSP n,b,x,y,m,p,c

Definisce uno sprite dove:

'n' è il numero dello sprite (0-7);

'b' è il blocco dei dati dello sprite;

'x,y' indicano la posizione dello sprite;

'm' = 0 risoluzione normale, 'm' = 1 multicolor;

'p' = 0 priorità sprite, 'p' = 1 priorità caratteri;

'c' colore dello sprite.

Se lo sprite deve essere visualizzato in alta risoluzione, per ovvi motivi di memoria, è indispensabile inserire i dati a partire da 32768+1024 fino a 36864.

I numeri dei blocchi vanno da 16 a 63.

Se per esempio volessimo porre i dati nel blocco 20 e visualizzare lo sprite in alta risoluzione, i dati andrebbero posti a partire da 32768+20*64=34048.

— JSPOF n

Strange Basic

Indirizzo di partenza con intestazione: \$CF16 #53014

Indirizzo di partenza senza intestazione: \$9268 #37480

-VETTORI MODIFICATI:

0302-0303: \$91DA #37338 WARM START (\$A483)
 0308-0309: \$9200 #37376 CHARACTER DISPATCH (\$A7E7)
 0314-0315: \$CF6A #53098 IRQ (\$EA31)

Usati in pagina zero i bytes \$14, \$15, \$33, \$34, \$37, \$38, \$FB, \$FC, \$FD, \$FE...

-ORGANIZZAZIONE DELLA MEMORIA:

34811 BYTES FREE
 Memoria occupata dal programma: \$9000-\$9FFF / \$C000-CFFF
 Top of memory: \$BFFF
 Memoria per la pagina grafica ad alta risoluzione: \$A000-\$B3F3 (\$40960-\$48959)
 Memoria colore alta risoluzione: \$8000-\$83E7

BYTES SIGNIFICATIVI:

-BYTES USATI E MODIFICATI DALLE ROUTINES DELLO "STRANGE BASIC":

\$9920-\$992F Dati per la scansione video (JRUST)
 \$9765-\$977F Dati inseriti con i comandi musicali (JFILT, JPLAY, ecc.)
 \$9F08-\$9FFF Dati transienti di vario uso
 \$9FF6 Numero di caratteri per lo scorrimento (JSCRR, ecc.)
 \$9FB0-\$9FC1 Caratteri da inserire per lo scorrimento
 \$CFD6 Transiente per il controllo dei tasti funzione (JFKEY)
 \$CFD7 Puntatore per la scansione (JRUST)
 \$CFD8 Transienti per la rotazione caratteri verso l'alto e il basso
 \$CFD9 Contatore per la ricerca dei caratteri da ruotare
 \$CFDA Byte indicatore delle routines di IRQ attivate:

| | BIT | FUNZIONE | |
|--------|-----|-------------------------------|---------|
| | 0 | Controllo scansione | (JRUST) |
| \$CFDA | 1 | Controllo tasti funzione | (JFKEY) |
| | 2 | Controllo rotazione caratteri | (JROTA) |

Gli altri bit del byte \$CFDA non sono usati.

\$CFDB-\$CFDF Tipo di rotazione per i cinque caratteri ruotabili:

| VALORE BYTE | ROTAZIONE |
|-------------|-------------|
| \$00 | Destra |
| \$01 | Sinistra |
| \$02 | Alto |
| \$03 | Basso |
| \$FF | Disattivata |

\$CFE0-\$CFE4 Codice dei caratteri per la rotazione (JROTA)
 \$CFE6-\$CFED Velocità di rotazione dei caratteri per il conto alla rovescia
 \$CFEB-\$CFE5 Velocità di rotazione dei 5 caratteri fissa
 \$CFF0-\$CFF9 Puntatori alle locazioni di memoria dei bytes da ruotare (byte basso e alto della locazione)
 \$CFFC Contatore per l'inserimento dei caratteri da ruotare (JROTA)
 \$CFFD Indicatore di stato:
 \$01 Attivata la memoria carattere programmabile (JMEMC1)
 \$02 Disattivata la memoria carattere programmabile (JMEMC0)

\$CFFE Indicatore di stato per il modo grafico:
 \$FF alta risoluzione normale (JHIRS)
 \$00 bassa risoluzione (JLOWR)
 \$01 alta risoluzione multicolor (JMCOL)

\$CFFF Indica il valore del byte \$D018 prima di entrare in HI-RES

-TAVOLE DI DATI NON MODIFICATE DAI COMANDI AD USO DELLE ROUTINES DEL PROGRAMMA:

\$9000-\$90CF Tavola dei comandi aggiuntivi dello STRANGE BASIC
 \$90D0-\$90E7 Dati dell'intestazione iniziale dello STRANGE BASIC
 \$9100-\$918F Puntatori alle locazioni delle routines dei comandi dello STRANGE BASIC (byte basso e byte alto)
 \$9190-\$91D9 Tabella dei quattro per il confronto con la tavola dei comandi
 \$96DE-\$9700 Dati dell'intestazione iniziale dello STRANGE BASIC
 \$C1A6-\$C1ED Tavola dati per individuare il byte per eseguire il PLOT
 \$CD95-\$CDA4 Dati per JFKEY
 \$CFA0-\$CFCF Dati dell'intestazione iniziale dello STRANGE BASIC

Disabilita (spegne) lo sprite 'n'.
 — JMEMC n
 Se n=1 trasferisce la memoria carattere in \$3000;
 se n=0 ripristina la memoria.

Musica ed effetti speciali

Questi sono comandi che vi permettono di sfruttare al meglio e con poca fatica le straordinarie doti musicali del vostro Commodore 64. Permettono di creare semplici melodie o effetti «specialissimi» grazie all'uso dei filtri.

— JADSR v,o,a,d,s,r,m,f
 Seleziona i valori di ADSR (Attack, Decay, Sustain, Release) (ovvero l'involuppo) per una voce e deve sempre essere usato prima di suonare una nota.

'v' = voce (0-1-2);
 'o' = forma d'onda (0 = rumore, 1 = rett., 2 = dente sega, 3 = triang.);
 'a' = Attack (0-15);
 'd' = Decay (0-15);
 's' = Sustain (0-15);
 'r' = Release (0-15);
 'm' = modo (0 = normale, 1 = ring simulato, 2 = Sincronize);

'f' = frequenza del tocco simulato.
 — JFILT fq,rs,fl,f2,f3,m

Serve a selezionare i filtri dove:
 'fq' = frequenza di taglio del filtro (*17 in Hertz);

'rs' = risonanza (0-15);
 'fl' = 1-attiva filtro per voce 1, 0-disattiva;

'f2' = 1-attiva filtro per voce 2, 0-disattiva;

'f3' = 1-attiva filtro per voce 3, 0-disattiva;

'm' = modo filtro (0 = no filtro, 1 = passa basso, 2 = passa banda, 3 = passa alto).

Per attivare più modi di filtro differenti si sommano assieme i valori: 1+2=3 - passa basso + passa banda.

— JPLAY v,f,vol.
 Permette di suonare una nota 'f' con volume 'vol' e voce 'v'.

— JMSOF n
 Disattiva la voce 'n'.

Effetti speciali

Questi comandi permettono di arricchire i vostri programmi di effetti speciali quali caratteri rotanti in diverse direzioni o tremolanti. Sono comandi molto utili anche per la realizzazione di giochi.

— JROTA c,v,t
 Fa ruotare il carattere di codice schermo 'c' con velocità 'v' (0-254) in queste direzioni:

't'=0 ruota a destra;
 't'=1 ruota a sinistra;
 't'=2 ruota in alto;
 't'=3 ruota in basso.

Prima di eseguire questo comando, bisogna digitare JMEMC 1, per spostare la memoria. Si possono eseguire al massimo 5 rotazioni contemporaneamente.

— JFERM n
 Ferma la rotazione del carattere di codice schermo 'n'.

Velocizzazione del registratore

Questi comandi permettono di salvare, verificare e rileggere dal registratore i vostri programmi con una velocità circa 10 volte maggiore quella normale.

— J-SAV
 Salva un programma su registratore in iper-velocità.

— J-VER
 Verifica un programma registrato su nastro in iper-velocità.

— J-LDA
 Carica dal registratore un programma in iper-velocità.

Brevi note tecniche ed informazioni

Il programma è allocato da \$9000 a \$9FFF e dopo la ROM da \$C000 a \$CFFF.

A \$9200 c'è la routine di interpretazione dei comandi, da \$9000 a \$9200 ci sono tutti i dati relativi ai comandi ed i vettori per il salto alle varie subroutine.

A \$9500 ci sono i dati transienti usati da

svariati comandi che non agiscono sull'IRQ (JSCRL, JBSCCK, ecc...).

Una qualsiasi azione in queste zone di memoria comprometterebbe sicuramente il funzionamento del programma.

Quando si è in alta risoluzione viene commutato il terzo blocco di memoria, la schermata in alta risoluzione viene posta sotto la ROM Basic (\$A000, \$BFFF) e i dati del colore vanno posti a \$8000.

In questo caso i dati degli sprite vanno messi nella zona di memoria compresa tra \$8400 e \$8FFF (62 sprite).

I vettori che vengono variati sono:
 0302 - 0303 che punta a \$91DA (Warm Start);

0308 - 0309 che punta a \$9200 (Character Dispatch);

0314 - 0315 che punta a \$CF6A IRQ.

Per fare partire il programma dopo un SYS 64738 si può fare:

SYS 53014 (CF16) con intestazione e presentazione.

SYS 37480 (9268) senza presentazione.

Infine lo Strange Basic è resistente al RUN/STOP e RESTORE.

Il Castello di Dracula

di Carlo Arlotti
 Viserba di Rimini (FO)

Il fine ultimo di questa avventura, è quello di riuscire a trovare ed uccidere il Conte Dracula. Essa inizia all'esterno da dove cominceremo subito a cercare tutte le possibili vie di accesso al castello (cosa non troppo semplice). Una volta riusciti ad entrare, si devono trovare gli oggetti necessari per il compimento del gioco.

L'avventura è stata descritta interamente in Basic per cui una volta entrati in possesso della stessa si consiglia effettuare una versione compilata per rendere il gioco più veloce e divertente.

Le persone che hanno una discreta esperienza di Basic possono facilmente modificare luoghi, oggetti, spostamenti in modo tale da crearsi una avventura «su misura». Il gioco gira perfettamente anche su registratore semplicemente variando un paio di righe.

Subito dopo che si è mandato in esecuzione il programma principale (DRACULA)

esso richiede il nome del file che contiene i dati relativi all'inizio del gioco: il nome del file di inizio è «BOOT». Cambiando il nome del file, è possibile riprendere un gioco precedentemente salvato.

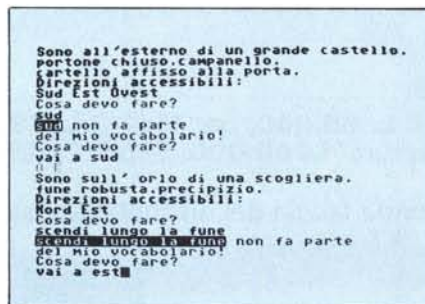
Il corpo dell'avventura è composto da due pezzi: quello principale ed un file sequenziale che contiene tutti i dati necessari per lo svolgimento del gioco. Per poter creare quest'ultimo è sufficiente mandare in esecuzione il programma «DATI DRACULA» che provvederà a creare automaticamente il file sequenziale Boot sia che si usi il registratore che il drive. L'interprete dei comandi accetta solo verbi in seconda persona (per intenderci prendi, guarda, sali, ecc.), seguiti dall'oggetto.

Lista comandi

| | | |
|------------|--------------|---------------|
| Accendi, | Impiccati, | Spegni, |
| Alza, | Immergiti, | Salta, |
| Ascolta, | Inventario, | Sega, |
| Abbassa, | Leggi, | Sali, |
| Asciuga, | Lascia, | Scendi, |
| Aiuto, | Mangia, | Taglia, |
| Annega, | Muori, | Vai in alto, |
| Apri, | Prendi, | Vai in basso, |
| Cerca, | Risali, | Est, |
| Esci, | Scuoti, | Ovest, |
| Esamina, | Salva gioco, | Nord, |
| Entra, | Suona, | Sud. |
| Guarda, | Sposta, | |
| Inserisci, | Scava, | |

Abbreviazioni


A per Vai in alto, B per Vai in basso, E per est, O per ovest, N per nord, S per sud, O per guarda e I per inventario.



Hard & soft

LA NIWA 

PUÒ ESSERE LA TUA MIGLIORE AMIGA® Distributore autorizzato COMMODORE

In regalo a tutti gli acquirenti di un PC 
la tessera del NIWA  AMIGA CLUB.

 AMIGA costa £ 1.950.000 IVA comp.
consegna GRATIS IN TUTTA ITALIA.

Tutto il software disponibile
e l'hardware novità.

Inoltre la NIWA vi propone per il vostro C/64-C/128:

| | | |
|--|----|------------|
| Floppy disk "Memorette" 5 1/4 ssdd 100% error free | cd | L. 1.150 |
| Floppy disk bulk 3 1/2 dsdd 100% error free | da | L. 3.000 |
| Allinea testine Cartridge | | L. 32.000 |
| Allinea testine con turbotape e turbo 202 | | L. 39.000 |
| MPS 802 New Graphic CON MONTAGGIO GRATUITO rende 100% compatibile la tua MPS 802 con i programmi di grafica | | L. 80.000 |
| O.M.A. Non permettere che i tuoi programmi originali si ROVININO. Con O.M.A., puoi fare una copia di sicurezza in un unico file (!) ricassettabile del tuo software su disco o su nastro | | L. 99.000 |
| HACKER Cartridge: trasferisce il 99% del tuo software protetto da nastro e da disco a disco in soli 4 minuti senza bisogno di conoscenza Linguaggio. | | L. 80.000 |
| HACKER-TAPE: permette di ricassettare qualsiasi tipo di programma precedentemente trattato con HACKER, senza nessun problema di blocchi, leggendo in turbo da disco e scrivendo in turbo su nastro | | L. 45.000 |
| OFFERTA: HACKER + HACHER TAPE | | L. 99.000 |
| Speeddos per C64 L. 65.000 per C128 L. 85.000, per 1541 C L. 79.000, Fast load reset L. 35.000, Isepic L. 50.000, Capture L. 99.000, Super Cartridge L. 99.000, Super Freere 3 L. 99.000 | | |
| Double side kit per scrivere sulla seconda faccia del dischetto senza più forarlo - disinseribile. | | L. 10.000 |
| Commodore 64 | | L. 369.000 |