

**SUPER-CONCOURS**  
**voir P.37**

LE MAGAZINE DE LA MICRO PASSION

**MICRO**  
**v.o**

**ENVAHISSEUR**  
**COMPR**  
**MSX 3D**  
**UTIL-1**  
**SOFT-AIDE**  
**MICRO MORSE**  
**THOMAVE**  
**LEM SX**  
**FM VOICING**

**LE**  
**CAHIER**  
**DU**  
**LOGICIEL**

**CAHIER DES AS**

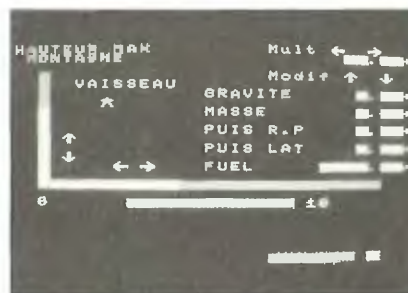
**N°6**

## EDITO

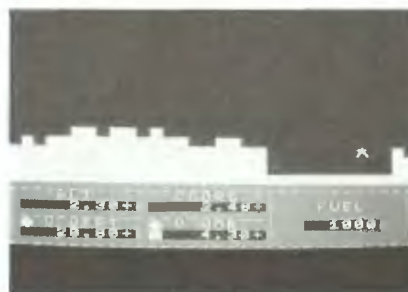
Le super-concours court toujours au grand dam du facteur qui doit maintenant faire plusieurs voyages pour nous livrer votre courrier. La fête n'est d'ailleurs pas finie pour lui puisque le super-concours n'est pas encore terminé. Mais dépêchez-vous quand même car, comme le prévoyait le règlement, la date de clôture des envois est très proche. Les programmes que nous recevrons après le 1er avril ne pourront plus être sélectionnés pour gagner l'un des huit lots qui restent à attribuer. Le « Cahier du Logiciel » continuera cependant toujours aussi tarhadziboumant. Il y aura comme avant deux programmes gagnants, dont l'un pour le « Cahier des As ».

Nous devons, hélas !, évoquer une malheureuse affaire. Parmi les gagnants du dernier numéro, un petit malin a réussi à faire passer un programme déjà paru ailleurs. Heureusement, vous veillez. Et, s'il n'est déjà pas facile de nous blouser (mais on ne peut pas tout connaître), il l'est encore plus de vous tromper. Bien évidemment, comme le règlement le précise, il ne recevra pas ce qui aurait dû lui revenir.

Un petit rappel pour ceux qui envoient des routines en langage machine. Indiquez-nous clairement les adresses de début et de fin de chaque routine (et ce qu'elles font).



*Hors-concours, le programme LEM-SX réalisé par l'un de nos collaborateurs. Il s'agit d'une adaptation d'un programme pour TO 7 paru dans Micro 7 n° 13. Le but du jeu : faire atterrir son vaisseau sur la piste.*



### FICHE DE PARTICIPATION AU CONCOURS À RETOURNER AVEC VOS LOGICIELS.

Machine: .....	Nom: .....
Extensions: .....	Prénom: .....
Cassette <input type="checkbox"/> Disquette <input type="checkbox"/>	Adresse: .....
Langage: .....	Code Postal: ..... Ville: .....
Programme: _____	Tel: ..... Date: / /86

N'hésitez pas à nous envoyer vos programmes sur support magnétique. Nous vous expédions en échange une disquette ou une cassette vierge, correspondant à votre expédition. Ces disquettes et cassettes vous sont offertes par SCOTCH.

Langage : Basic



**DAMIEN POUPARD  
GAGNE  
UN SONY HIT BIT  
HB 75 F  
AVEC DRIVE  
HBD 50**

**Damien Poupard**

**A treize ans et demi, Damien est un assez bon élève de classe de quatrième. Au départ, il voulait créer un programme de jeu qu'il n'a pas trouvé assez original. Il s'est donc fait un peu aider pour ce logiciel qui est parti d'une idée bien à lui. Damien programme depuis deux ans mais il se sent plutôt une âme d'artiste et envisage une carrière dans la musique, le cinéma ou l'écriture...**

Les mots croisés sont un sport national. Mais avez-vous déjà envisagé de faire des mots croisés en trois dimensions. Cela paraît impossible. C'est pourtant ce que permet de faire ce programme. Imaginez un cube composé de cinq petits cubes par côté (soit 125 cases). Dans chaque case on peut mettre une lettre ou rien. Une lettre appartient ainsi à trois mots ; un horizontal, un vertical et un profondeur.

Le problème est qu'il est difficile d'afficher la grille entière en perspective, de manière que tout soit lisible. On se contentera donc de ne voir que les trois plans (face, côté et dessus) qui passent par la lettre sur laquelle est pointé le curseur. Ce curseur peut être déplacé dans les trois dimensions : horizontalement, verticalement ou en profondeur. Et dans chaque dimension en avant ou en arrière (flèche gauche et droite). Un pointeur indique la dimension choisie. Le changement de dimension s'effectue avec la touche « return ».

Le programme permet trois modes de travail : création ou modification d'une grille, création ou modification des définitions et jeu. Chacun de ces modes peut être sélectionné au départ. Lorsque l'on a fini de rentrer une grille on passe du mode création de grille au mode création de définitions sans revenir au menu grâce à la touche « ! ».

D'autres touches ont des fonctions particulières (elles sont paramétrées en début d'initialisation et vous pouvez changer leurs valeurs).

ESCAPE sort du programme et permet de sauver votre travail.

SELECT permet de rentrer une définition (seulement dans le mode adéquat).

RETURN, outre le changement de dimension, garde sa signification habituelle de validation.

« \* » permet de rentrer une case noire dans la grille (mode création de grille). « & » donne la grille solution et « » le mot sur lequel le curseur pointe (dans le mode jeu).

Sur ce, je vais prendre deux aspirines et je vous conseille de laisser la boîte à proximité de la main. ■

**Damien POUPARD**

NDLR : Des rumeurs courent à propos de la création d'une grille à quatre dimensions par notre ami Dupin. On ne l'a pas revu depuis...

```

1 'SAVE 'MX3DB.BAS
10 GOSUB 1620 'PRESENTATION
20 CLEAR 5000;DEFINT A-Z:KEYOFF;COLOR 15
,1,1;SCREEN 1
30 ON STOP GOSUB 110:STOP ON
40 '
50 GOSUB 1990 'INIT PROG
60 GOSUB 1360 'LOAD
70 GOSUB 1300 'AFF GRILLES
80 GOSUB 120 'TRAITEMENT
90 CLS:IF NJ<>3 THEN GOSUB 1510 'SAVE
100 LOCATE 0,20:PRINT "ON RECOMMENCE : 0"
;:LOCATE 16:R$=INPUT$(1):IF R$<>"N" THEN
60
110 SCREEN 0:KEY ON:END
120 'TRAITEMENT -----
130 LOCATE 13,PT#8:PRINT "-->";
140 LOCATE C#2+2,L+2:R$=INPUT$(1)
150 IF R$=CD$ THEN LOCATE 13,PT#8:PRINT
" ";:PT=(PT+1)MOD 3:GOTO 120
160 IF R$=E$ THEN 300
170 GOSUB 660 'ETEIND CURSEURS
180 IF R$=FG$ THEN ON PT+1 GOSUB 310,380
,450:GOTO 280
190 IF R$=FD$ THEN ON PT+1 GOSUB 340,410
,480:GOTO 280
200 ON NJ GOTO 210,230,250
210 IF R$=DE$ THEN GOSUB 890:NJ=2:GOTO 2
80
220 GOTO 270
230 IF R$=SEL$ THEN GOSUB 740:GOTO 280
240 BEEP:GOTO 280
250 IF R$=SM$ THEN GOSUB 1230:GOTO 280
    
```

```

260 IF R$=50$ THEN GOSUB 1210:GOTO 280
270 GOSUB 520 'UN CARAC
280 GOSUB 680 'RALUME CURSEURS
290 GOTO 120
300 RETURN
310 ' GAUCHE-----
320 IF C=0 THEN BEEP ELSE C=C-1
330 GOTO 360
340 ' DROITE-----
350 IF C=4 THEN BEEP ELSE C=C+1
360 LOCATE 28,16:PRINT CHR$(145+C);:GOSU
B 610
370 RETURN
380 ' HAUT-----
390 IF L=0 THEN BEEP ELSE L=L-1
400 GOTO 430
410 ' BAS-----
420 IF L=4 THEN BEEP ELSE L=L+1
430 LOCATE 28,0:PRINT CHR$(149+L);:GOSUB
640
440 RETURN
450 ' AVANT-----
460 IF P=0 THEN BEEP ELSE P=P-1
470 GOTO 500
480 ' ARRIERE-----
490 IF P=4 THEN BEEP ELSE P=P+1
500 LOCATE 28,8:PRINT CHR$(97+P);:GOSUB
590
510 RETURN
520 ' 1 CARAC DANS GRILLE-----
530 IF G$(0,P,L,C)=CN$ OR R$=CN$ THEN IF
NJ=3 THEN PRINT CL$;:GOTO 580 ELSE 550
540 IF (R$<"A" OR R$>"Z") AND R$<>" "
THEN BEEP:GOTO 580
550 G$(0,P,L,C)=R$
560 GOSUB 660.
570 ON PT+1 GOSUB 340,410,480
580 RETURN
590 ' AFFICHE FACE-----
600 FOR XL=0 TO 4:FOR XC=0 TO 4:LOCATE X
C#2+2,XL+2:PRINT G$(0,P,XL,XC);:NEXT:NEX
T:RETURN
610 ' AFFICHE COTE-----
620 FOR XL=0 TO 4:FOR XP=0 TO 4:LOCATE X
P#2+2,XL+10:PRINT G$(0,XP,XL,C);:NEXT:NE
XT:RETURN
630 RETURN
640 ' AFF DESSUS-----
650 FOR XP=0 TO 4:FOR XC=0 TO 4:LOCATE X
    
```

```

C82+2,XP+18:PRINT G$(0,XP,L,XC);:NEXT:NE
XT:RETURN
660 ' APRES CURSEURS----
670 LOCATE C82+2,L+2:PRINT G$(0,P,L,C);:
LOCATE P82+2,L+10:PRINT G$(0,P,L,C);:LOC
ATE C82+2,P+18:PRINT G$(0,P,L,C);:RETURN
680 ' 3 CURSEURS -----
690 IF NJ>1 THEN FOR I=0 TO 2:MC=DP(I,P,
L,C):GOSUB710:NEXT
700 LOCATE P82+2,L+10:PRINT CHR$(255);:L
OCATE C82+2,P+18:PRINT CHR$(255);:RETURN
710 ' AFFIC LIGNES DEF-----
720 LOCATE 14,I88+1:PRINT LM(MC);"lettre
s";
730 FOR J=0 TO 4:LOCATE 14,I88+2+J:PRINT
SPC(15);:LOCATE 14,I88+2+J:PRINT MID$(D
F$(MC),J815+1,15);:NEXT:RETURN
740 ' ENTRE DEFINITIONS-----
750 MC=DP(PT,P,L,C):LM=LM(MC):DM=DM(MC):
IF MC<2 THEN BEEP:GOTO 770
760 GOSUB 780 'SAISIE DEF
770 RETURN
780 ' SAISIE DEF -----
790 W$="":X=0
800 LOCATE 14+X MOD 15,X/15+PT88+2:R$=IN
PUT$(1)
810 IF R$=CHR$(13) THEN 870
820 IF R$=F$ THEN IF X>0 THEN PRINT " "
;:X=X-1:W$=MID$(W$,1,X):GOTO 800 ELSE BE
EP:GOTO 800
830 IF X>69 THEN BEEP:GOTO 800
840 IF INSTR(VA$,R$)=0 THEN BEEP:GOTO 80
0
850 PRINT R$;:X=X+1:W$=W$+R$
860 GOTO 800
870 IF W$<>" THEN DF$(MC)=W$
880 RETURN
890 ' CHERCHE LES NOMS-----
HORIZONTAL
900 MX=2
910 FOR P=0 TO 4:GOSUB 590
920 FOR L=0 TO 4:C=0
930 IF G$(0,P,L,C)=CN$ THEN DP(0,P,L,C)=
1:IF C<4 THEN C=C+1:GOTO930 ELSE 980
940 DM=C:LM=1
950 IF C<4 THEN C=C+1:IF G$(0,P,L,C)<>CN
$ THEN LM=LM+1:GOTO950 ELSE DP(0,P,L,C)=
1
960 IF LM>1 THEN MX=MX+1:LM(MX)=LM:DM(M
X)=DM:FOR I=DM TODM+LM-1:DP(0,P,L,I)=MX:L
OCATE I82+2,L+2:PRINT USING"@";MXMOD10;:
NEXT
970 IF C<4 THEN 930
980 NEXT:NEXT
990 ' VERTICAL -----
1000 FOR C=0 TO 4:GOSUB 610
1010 FOR P=0 TO 4:L=0
1020 IF G$(0,P,L,C)=CN$ THEN DP(1,P,L,C)
=1:IF L<4 THEN L=L+1:GOTO1020 ELSE 1070
1030 DM=L:LM=1
1040 IF L<4 THEN L=L+1:IF G$(0,P,L,C)<>C
N$ THEN LM=LM+1:GOTO1040 ELSE DP(1,P,L,C
)=1

```

```

1050 IF LM>1 THEN MX=MX+1:LM(MX)=LM:DM(M
X)=DM:FOR I=DM TODM+LM-1:LOCATE P82+2,I+
10:PRINT USING"@";MXMOD10;:DP(1,P,L,C)=M
X:NEXT
1060 IF L<4 THEN 1020
1070 NEXT:NEXT
1080 ' PROFONDEUR -----
1090 FOR L=0 TO 4:GOSUB 640
1100 FOR C=0 TO 4:P=0
1110 IF G$(0,P,L,C)=CN$ THEN DP(2,P,L,C)
=1:IF P<4 THEN P=P+1:GOTO1110 ELSE 1160
1120 DM=P:LM=1
1130 IF P<4 THEN P=P+1:IF G$(0,P,L,C)<>C
N$ THEN LM=LM+1:GOTO1130 ELSE DP(2,P,L,C
)=1
1140 IF LM>1 THEN MX=MX+1:LM(MX)=LM:DM(M
X)=DM:FOR I=DM TODM+LM-1:LOCATE C82+2,I+
18:PRINT USING"@";MXMOD10;:DP(2,I,L,C)=M
X:NEXT
1150 IF P<4 THEN 1110
1160 NEXT:NEXT
1170 P=0:L=0:C=0:GOSUB 590:GOSUB610:GOSU
B640
1180 RETURN
1190 ' STOCK GRILLE-----
1200 FOR Z=0 TO4:FOR Y=0 TO 4:FOR X=0 TO
4:G$(1,Z,Y,X)=G$(0,Z,Y,X):NEXT:NEXT:NEX
T:GOSUB590:GOSUB610:GOSUB640:RETURN
1210 ' SOLUTION-----
1220 FOR Z=0 TO4:FOR Y=0 TO 4:FOR X=0 TO
4:G$(0,Z,Y,X)=G$(1,Z,Y,X):NEXT:NEXT:NEX
T:GOSUB590:GOSUB610:GOSUB640:RETURN
1230 'SOLUTION UN MOT-----
1240 MC=DP(PT,P,L,C):DM=DM(MC):FM=DM+LM(
MC)-1:DN PT GOTO 1260,1270
1250 FOR I=DM TO FM:G$(0,P,L,I)=G$(1,P,L
,I):NEXT:GOTO1280
1260 FOR I=DM TO FM:G$(0,P,I,C)=G$(1,P,I
,C):NEXT:GOTO1280
1270 FOR I=DM TO FM:G$(0,I,L,C)=G$(1,I,L
,C):NEXT:GOTO1280
1280 GOSUB590:GOSUB610:GOSUB640
1290 RETURN
1300 ' AFF GRILLES-----
1310 CLS
1320 FOR I=0 TO 2:FOR J=0 TO 4:LOCATEJ82
+2,I88:PRINT CHR$(145+J+48*(I=1));:NEXT:
LOCATE 17:PRINTHV$(I)TAB(28);CHR$(49+48*
I);
1330 PRINT T$:FOR L=0 TO 4:PRINT CHR$(49
+L-48*(I=2))U$:NEXT:PRINT T$;:NEXT
1340 PT=0:P=0:L=0:C=0:GOSUB590:GOSUB610:
GOSUB640:GOSUB680
1350 RETURN
1360 ' LOAD -----
1370 CLS:PRINT "MOTS CROISES 3 DIMENSION
S"
1380 LOCATE 2,4:PRINT "CREER UNE GRILLE"
:LOCATE 2,6:PRINT "MODIFIER SANS DEFINIT
IONS":LOCATE 2,8:PRINT "MODIFIER LES DEF
INITIONS":LOCATE2,10:PRINT "JOUER":NJ=0
1390 LOCATE 2,NJ82+4:R$=INPUT$(1):IF R$=
FD$ THEN NJ=(NJ+1)MOD4:GOTO 1390 ELSE IF

```

```

R$<>CD$ THEN 1390
1400 W$=SPACE$(5*5*5):IF NJ=3 THEN G=1 E
LSE G=0
1410 IF NJ=0 THEN LOCATE 0,12:PRINT "INI
TIALISATION...":GOTO1450
1420 LOCATE 0,12:PRINT "NOM FICHER : "NF
$;:LOCATE 13:LINEINPUT NF$:PRINT "CHARGE
"NF$
1430 OPEN NF$ FOR INPUT AS 1
1440 LINEINPUT #1,W$
1450 X=0:FOR P=0 TO4:FOR L=0 TO 4:FOR C=
0 TO 4:X=X+1:G$(6,P,L,C)=MID$(W$,X,1):IF
G=1 THEN IF G$(1,P,L,C)=CN$ THEN G$(0,P
,L,C)=CN$ ELSE G$(0,P,L,C)=" "
1460 NEXT:NEXT:NEXT:IF NJ<2 THEN 1490
1470 FOR I=0 TO 2:FOR P=0 TO 4:FOR L=0 T
O 4:FOR C=0 TO 4:INPUT#1,DP(1,P,L,C):NEX
T:NEXT:NEXT:NEXT
1480 INPUT#1,MX:FOR MC=1 TO MX:INPUT#1,D
F$(MC):INPUT#1,DM(MC):INPUT#1,LM(MC):NEX
T
1490 BEEP:BEEP:CLOSE
1500 RETURN
1510 ' SAVE -----
1520 PRINT "SAUVE FICHER : 0";:LOCATE
17:R$=INPUT$(1):IF R$="N" THEN 1610
1530 LOCATE 0,3:PRINT "NOM FICHER : "NF$
;:LOCATE 13:LINEINPUT NF$:PRINT "SAUVE
"NF$
1540 PRINT "MAGNETO ET RETURN";:R$=INPUT
$(1):PRINT:PRINT "SAUVE...."
1550 OPEN NF$ FOR OUTPUT AS 1
1560 W$="":FOR P=0 TO 4:FOR L=0 TO 4:FOR
C=0 TO 4:W$=W$+G$(0,P,L,C):NEXT:NEXT:NE
XT
1570 PRINT #1,W$:IF NJ<2 THEN 1600
1580 FOR I=0 TO 2:FOR P=0 TO 4:FOR L=0 T
O 4:FOR C=0 TO 4:PRINT#1,DP(1,P,L,C):NEX
T:NEXT:NEXT:NEXT
1590 PRINT#1,MX:FOR MC=1 TO MX:PRINT#1,D
F$(MC):PRINT#1,DM(MC):PRINT#1,LM(MC):NEX
T
1600 PRINT"TERMINE...":BEEP:BEEP:CLOSE 1
1610 RETURN
1620 ' PRESENTATION ----
1630 SCREEN 2:COLOR 2:OPEN "GRP:" FOR OU
TPUT AS 1
1640 RESTORE 2270:D=824:FORJ=0 TO 4: REA
D W,W$:FOR I=0 TO 7:VPOKE D,VAL("&H"+MID
$(W$,I82+1,2)):VPOKE D+8192,&HC1:D=D+1:N
EXT:D=D+8:NEXT
1650 PLAY "M5000 S1 LBT160":RESTORE 1790
:READ W$:PLAY W$:READ W$:PLAY W$:READ W$
:PLAY W$
1660 C=88:C2=C/2:LINE (88,8815)-STEP(C,
-C),2,BF:COLOR 4:LINE-STEP(C2,C2):LINE-S
TEP(0,C):LINE-STEP(-C2,0):LINE-STEP(0,-C
-C2):PAINT STEP(1,2):COLOR 6:LINE (88,8
815)-STEP(C2,C2):LINE-STEP(C,0):LINE-STE
P(-C2,-C2):LINE-STEP(-C,0):PAINTSTEP(2,1
)
1670 FORI=0 TO 4:PRESET(32,I816+40):COLO
R 4:PRINT#1,I+1;:PRESET(144+I88,28+I88):

```

```

COLOR 6:PRINT#1,CHR$(97+1):NEXT
1680 READ W$:PLAY W$
1690 PRESET (72,72):COLOR15:PRINT #1,"M$
X 3D";
1700 CLOSE
1710 '-----PRELUDE--(BACH)---
1720 READ W$:IF W$="X" THEN RESTORE 1790
:READ W$
1730 PLAY W$
1740 R$=INKEY$:IF R$(">)" THEN 1760
1750 GOTO 1720
1760 RETURN
1770 '
1780 '
1790 DATA 036056D04B05D046BD
1800 DATA 036056D04B05D046BD
1810 DATA 03605FD04B05D046BD
1820 DATA 03605FD04B05D046BD
1830 DATA 03605EC04605C04E6C
1840 DATA 03605EC04605C04E6C
1850 DATA 03605C04AF+ADF+C
1860 DATA 03605C04AF+ADF+C
1870 DATA 03604B6D603B04D036
1880 DATA 03F+04B6D603B04D03F+
1890 DATA 03E056E04B05E046BE
1900 DATA 04D056E04B05E046BD
1910 DATA 04C+056E04A05C+04E6C+
1920 DATA 03A056EC+E04A05C+046
1930 DATA 04D05F+D04A05D04F+AD
1940 DATA 04C05FD04A05D04FAC
1950 DATA 03B05FD046BD603B
1960 DATA 03605FD04B05D046BF
1970 DATA 04C05EC04605C04E6C
1980 DATA X
1990 ' INITIALISATION-----
2000 NF$="MX1":RESTORE 2230
2010 F$=CHR$(29) 'FLECHE GAUCHE
2020 FD$=CHR$(28) 'FLECHE DROITE
2030 ES$=CHR$(27) 'ESCAPE
FIN PROGRAMME
2040 SE$=CHR$(24) 'SELECT
ENTRER DEFINITION
2050 CD$=CHR$(13) 'RETURN
2060 CN$="@" 'CASE NOIRE
2070 DE$="!" '
PASSE A L'ENTREE DES DEF
2080 SD$="&" '
SOLUTION GRILLE
2090 SM$="&" '
SOLUTION UN MOT
2100 A$=CHR$(1):M$=A$+"M"+A$+"R":T$=" "+
A$+"X"+M$+M$+M$+M$+M$+M$+M$+M$+M$+"Y"
2110 W$=A$+"V ":U$=M$+M$+M$+M$+M$+M$+M$+M$+"V"
2120 W$=A$+"M"+A$+"O":T1$=" "+A$+"Z"+M$+
M$+M$+M$+M$+M$+M$+M$+M$+M$+"W"+A$+CHR$(91)
2130 HV$(2)="PROFONDEUR":HV$(0)="HORIZON
TAL":HV$(1)="VERTICAL"
2140 VA$=" ,.:;-'QMERTYUIOPASDFGHJKLZXCV
BNM"
2150 DC=VDP(3)*64:READ W$:FOR I=0 TO 31:
VPOKE DC+I,VAL("&H"+MID$(W$,I#2+1,2)):NE
XT
2160 DM=VDP(4)*2048

```

```

2170 READ W:IF W=-1 THEN 2190
2180 D=DM+W#8:READ W$:FOR I=0 TO 7:VPOKE
D+I,VAL("&H"+MID$(W$,I#2+1,2)):NEXT:GOT
O 2170
2190 ' TABLEAUX -----
2200 DIM G$(1,4,4,4)
2210 DIM DF$(100),DN(100),LM(100),DP(2,4
,4,4):DF$(0)="--":DF$(1)=CN$:LM(0)=1:LM
(1)=0:MX=2
2220 RETURN
2230 ' COULEURS -----
2240 DATA 212121212121F1F141414141616161
61F1F1A1F1F1F1F1F1F1F1F1F1F1F1F1F1F1F1
2250 ' REDEFINITION CARACTERES---
2260 DATA 42,007E7E7E7E7E7E00
2270 DATA 145,3B1010101010103800
2280 DATA 146,7C2B2B2B2B2B7C00
2290 DATA 147,FE5454545454FE00
2300 DATA 148,FF51514A4A4AFF00
2310 DATA 149,FE44442B2B107C00
2320 DATA -1

```

**EXEMPLE  
DE  
GRILLE**

FACE: 1

A R M E E  
U \* E \* L  
T O R D U  
R I O \* D  
E L E V A

FACE: 2

N O E \* P  
\* R U S A  
R \* T E L  
U N \* M A  
T I T A N

FACE: 3

T I R E E  
R O \* T I  
I N \* R M  
\* \* R E \*  
E T A \* O

FACE: 4

R \* \* I L  
\* T A \* \*  
M I \* M \*  
O R E S \*  
T \* \* I D

FACE: 5

E T A M E  
V I S A T  
A D \* E \*  
S \* M \* R  
E C U L E

**HORIZONTAL**

VERTICAL: a

1/ 1 - ARMEE: GROUPEMENT UNIFORME  
3/ 1 - TORDU: TOURNE EN SENS CONTRAIR  
E  
4/ 1 - RIO : BRAVO DANS UN WESTERN  
5/ 1 - ELEVA: HISSA

VERTICAL: b

1/ 1 - NOE : LE PREMIER QUI BUT TROP  
2/ 1 - RUSA : SE MONTRA PLUS FIN  
3/ 1 - TEL : IDENTIQUE  
4/ 1 - UN : ARTICLE  
4/ 2 - MA : POSSESSIF  
5/ 1 - TITAN: FILS D OURANOS

VERTICAL: c

1/ 1 - TIREE: SORTIE  
2/ 1 - RO : LETTRE GRECQUE  
2/ 2 - TI : SYMBOLE CHIMIQUE  
3/ 1 - IN : DANS LE VENT  
3/ 2 - RM : DU RHUM SANS CORPS  
4/ 1 - RE : NOTE  
5/ 1 - ETA : LETTRE GRECQUE

VERTICAL: d

1/ 1 - IL : PERSONNEL  
2/ 1 - TA : POSSESSIF  
3/ 1 - MI : NOTE  
4/ 1 - ORES : AVEC D ET AVEC DEJA  
5/ 1 - ID : LE MEME EN PLUS COURT

VERTICAL: e

1/ 1 - ETAME: RECOUVRE D UNE COUCHE M  
ETALLIQUE  
2/ 1 - VISAT: APPOSAT UN TAMPON  
3/ 1 - AD : VERS LATIN  
5/ 1 - ECULE: USE JUSQU A LA CORDE

**VERTICAL**

PROFONDEUR: I

a/ 1 - AUTRE: DIFFERENT  
b/ 1 - OIL : CARBURANT ANGLAIS  
c/ 1 - MEROE: ANCIENNE CAPITALE DE NU  
BIE  
e/ 1 - ELUDA: FIT MINE DE RIEN

PROFONDEUR: II

a/ 1 - RUT : PERIODE DE RAPPORTS  
b/ 1 - OR : CONJONCTION  
b/ 2 - NI : CONJONCTION  
c/ 1 - EUT : S APPROPRIA  
d/ 1 - SEMA : DISSEMINA  
e/ 1 - PALAN: TREUIL

PROFONDEUR: III

a/ 1 - TRI : CHOIX SELECTIF  
b/ 1 - ION : ATOME  
c/ 1 - RA : BRILLE SUR LE NIL

d/ 1 - ETR : AUXILIAIRE PRECIEUX  
 e/ 1 - EIM : VIEILLE AMIE TOUTE RETO  
 URNEE

PROFONDEUR: IV

a/ 1 - MOT : MIEUX VAUT NE PAS Y ETR  
 E PRIS  
 b/ 1 - TIR : SON CHAMP EST DANGEREUX  
 d/ 1 - MSI : PARTI NEOFASCISTE ITALI  
 EN

PROFONDEUR: V

a/ 1 - EVASE: ELARGI AU COL  
 b/ 1 - TID : PARLE EN VERLAN  
 c/ 1 - AS : FAIT SOUVENT LE PLI  
 c/ 2 - MU : MIS EN MOUVEMENT  
 d/ 1 - MAE : PRENOM FEMININ  
 e/ 1 - ET : CONJONCTION  
 e/ 2 - RE : NOTE

PROFONDEUR

HORIZONTAL: 1

I/ 1 - ANTRE: BAUGE OU CAVERNE  
 II/ 1 - ROI: BOURG LA REINE

III/ 1 - MER: ELLE DANSE AU FOND DES GO  
 LFES CLAIRS

IV/ 1 - EIM: LE CONTRAIRE D UNE VIEILL  
 E COPINE

V/ 1 - EPELE: MOT A MOT

HORIZONTAL: 2

II/ 1 - ROTI: CUIT AU FOUR  
 III/ 1 - EU: PARTICIPE PASSE  
 III/ 2 - AS: BONNE CARTE  
 IV/ 1 - ST: UN POUR CHAQUE JOUR  
 V/ 1 - LAI: POEME

HORIZONTAL: 3

I/ 1 - TRIMA: TRAVAILLA DUR  
 II/ 1 - NID: ABRI  
 III/ 1 - RT: AUX LIMITES DU RESSORT  
 IV/ 1 - DERME: COUCHE SUPERFICIELLE  
 V/ 1 - ULM: VICTOIRE NAPOLEONNIENNE

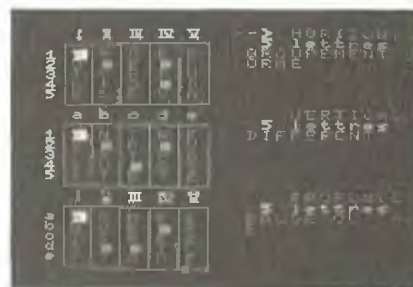
HORIZONTAL: 4

I/ 1 - RU: FRANCHI D UN BOND  
 I/ 2 - OS: STRUCTURE INTERNE  
 II/ 1 - IN: PREFIXE INCLUSIF  
 III/ 1 - REM: MESURE DE RADIOACTIVITE

IV/ 1 - MES: POSSESSIF  
 V/ 1 - DA: HOBBY QUAND IL EST DOUBLE

HORIZONTAL: 5

I/ 1 - ETETE: COMME DANTON OU LOUIS X  
 VI  
 II/ 1 - LIT: PREND CONNAISSANCE  
 III/ 1 - ETA: GRECQUE  
 IV/ 1 - VA: SE REND  
 IV/ 2 - IL: PERSONNEL  
 V/ 1 - ANODE: POLE NEGATIF



# THOMAVE

Langage : Basic



**MAXIME GAILLARD  
 GAGNE  
 UNE IMPRIMANTE  
 EPSON LX80**

**Maxime Gaillard  
 Trente-cinq ans, marié, trois enfants.  
 Programme depuis trois ans sur ma-  
 tériel Thomson, pour son plaisir. Ju-  
 riste, il utilise l'informatique sur plus  
 gros systèmes dans le cadre de sa  
 profession.**

**L**es passionnés de jeux d'aven-  
 ture vont être aux anges : ils vont  
 pouvoir créer eux-mêmes leur  
 propre circuit. Thomave est en effet un  
 module de création de jeux en mode  
 texte. Il permet d'intégrer une centaine  
 de situations et de les relier entre elles  
 grâce à plusieurs sortes de branche-  
 ments tout en gérant les points de vie.  
 L'option création du menu est la pre-  
 mière à considérer : pour chaque situ-  
 ation, à partir de la première, le pro-  
 gramme demande successivement  
 l'énoncé de la situation (six lignes ma-  
 ximum. Si l'énoncé comprend des sé-

parateurs comme des virgules il faut  
 ouvrir les guillemets), l'évolution des  
 points de vie (positive, nulle ou néga-  
 tive) et le type de branchement sur la  
 situation suivante. Il existe quatre types  
 de branchement :

- 1 - branchement direct, c'est-à-dire in-  
 conditionnel sur une situation dont on  
 indique le numéro ;
- 2 - branchement au choix : après avoir  
 indiqué le nombre de choix (3 maxi-  
 mum), il faut entrer l'énoncé de chaque  
 alternative (une ligne maximum) puis le  
 numéro de la situation de branche-  
 ment correspondant à chacune ;
- 3 - branchement aléatoire. Dans ce  
 cas, indiquer le pourcentage de chan-  
 ces de réussite (de 1 à 99) puis les nu-  
 méros des situations correspondant à  
 la réussite ou à l'échec ;
- 4 - branchement conditionnel : l'accès  
 à certaines situations (ouverture d'une  
 porte par exemple) peut n'être autorisé  
 que si l'on a déjà vécu une autre situa-

tion. Il faut donc indiquer le numéro de  
 cette situation préalable puis les bran-  
 chements subséquents selon que  
 cette situation préalable a été vécue ou  
 non.

Certaines situations (mort ou réussite)  
 ne nécessitent aucun branchement :  
 l'énoncé doit alors impérativement  
 s'achever par « fin ». Il est recommandé  
 de faire boucler cette situation en lui af-  
 fectant un branchement direct sur elle-  
 même. Il est conseillé d'imprimer le jeu  
 en cours de création pour avoir l'en-  
 semble des situations sous les yeux.  
 Les autres options ne présentent au-  
 cune difficulté. Il suffit de suivre les in-  
 dications portées à l'écran. A vous  
 maintenant et, pour une fois, vous se-  
 rez les « Maîtres du jeu ».

**Maxime GAILLARD**



```

8 'MODULE DE CREATION DE JEUX D'AVENTURE
10 '     MENU 1
12 '
14 CLEAR16000:DEFINT A-Z:DIM A$(100),Q$(
3,100),G(100),R(3,100),AL(100),B(100),PV
(100),P(100),V(100),C(100):GOSUB 376
16 CONSOLE 0,24:CLS:SCREEN4,0,0
18 ATTRB 1,1:LOCATE3,2:PRINT"MODULE 'AVE
NTURES'"
20 PRINT:ATTRB0,0:COLOR2:PRINT"VOULEZ-VO
US:"
22 RESTORE 34
24 FORI=1TOS
26 READ D$:COLOR0,2:LOCATE 5,7+1*3,0:PRI
NT CHR$(I+48):;COLOR2,0:PRINT" ";D$:PRIN
T
28 NEXT
30 R$=INPUT$(1):R=VAL(R$):IF R<1 OR R>5
THEN 30
32 ON R GOTO 38,68,142,142,198
34 DATA CREER UN FICHER,MODIFIER UN FIC
HIER,CHARGER UN FICHER,SAUVEGARDER UN F
ICHER,JOUER
36 '
38 '     CREATION
40 '
42 S=100:I1=1
44 FOR I=1 TO 5
46 CONSOLE 0,24:COLOR4,0:CLS:GOSUB 248
48 LOCATE 0,24,0:COLOR4,0:PRINT"ARRETER:
";:COLOR 0,4:PRINT"A";
50 COLOR 4,0:PRINT" POURSUIVRE: ";:COLO
R 0,4:PRINT"P";
52 COLOR 4,0:PRINT" IMPRIMER: ";:COLOR
0,4:PRINT"I";
54 R$=INPUT$(1):IF R$<>"A" AND R$<>"P"AN
D R$<>"I" THEN 54
56 IF R$="P" THEN 58 ELSE IF R$="A" THEN
S1=I:GOTO16:ELSE IF R$="I" THEN GOSUB 39
6:GOTO54
58 NEXT I
60 S1=S:GOTO16
62 '
64 '     MENU 2
66 '
68 S=100:CONSOLE 0,24:CLS:ATTRB 1,1:LOCA
TE 8,3:COLOR 5:PRINT"MODIFICATION"
70 ATTRB0,0:RESTORE 82
72 FOR I=1 TO 5
74 READ D$:LOCATE 2,1*3+7,0:COLOR0,4:PRI
NT CHR$(I+48):;COLOR4,0:PRINT" ";D$
76 NEXT I
78 R$=INPUT$(1):R=VAL(R$):IF R<1 OR R>5
THEN 78
80 CLS:ON R GOTO 90,116,132,450,16
82 DATA RECHERCHE DE LA SITUATION A MODI
FIER,MODIFICATION D'UNE SITUATION ISOLEE
,POURSUITE DE CREATION D'UN FICHER,VERI
FICATION DES BRANCHEMENTS,RETOUR AU MENU
PRINCIPAL
84 '
86 '     RECHERCHE
88 '

```

```

90 CLS
92 LOCATE 0,24,0:COLOR4,0:PRINT"ARRETER:
";:COLOR 0,4:PRINT"A";
94 COLOR 4,0:PRINT" POURSUIVRE: ";:COLO
R 0,4:PRINT"P";
96 COLOR 4,0:PRINT" IMPRIMER: ";:COLOR
0,4:PRINT"I";:CONSOLE0,23
98 LOCATE0,10:COLOR 5,0:INPUT"SITUATION
DE DEPART ";I1
100 FOR I=1 TO 5:GOSUB 276
102 R$=INPUT$(1):IF R$<>"A" AND R$<>"P"AN
D R$<>"I" THEN 102
104 IF R$="A" THEN I=5 ELSE IF R$="I" TH
EN GOSUB 396:GOTO102
106 NEXT I
108 GOTO 68
110 '
112 '     MODIFICATION ISOLEE
114 '
116 CONSOLE0,24:CLS:COLOR4:LOCATE 0,10:I
NPUT"NO DE LA SITUATION A MODIFIER ";:I
118 CLS:GOSUB 248
120 LOCATE 13,24:COLOR 5:PRINT"D'ACCORD?
(O/N)";
122 R$=INPUT$(1):IF R$="N" THEN 116
124 GOTO 68
126 '
128 '     POURSUITE DE CREATION
130 '
132 LOCATE0,10:COLOR4:INPUT"NO DE LA DER
NIERE SITUATION CREE ";:I1:I1=I1+1
134 GOTO 44
136 '
138 '     CHARGEMENT
140 '
142 CLS:COLOR 4:LOCATE0,10:INPUT"NO DU
FICHER";N$
144 PRINT:PRINT" PERIPHERIQUE PRET?
(O/N)"
146 R$=INPUT$(1):IF R$<>"O" THEN 16
148 SCREEN,,4:IF R=4 THEN 174
150 OPEN"1",#1,N$:INPUT#1,S:S1=5
152 FORI=1 TO 5
154 INPUT#1,A$(I),PV(I),B(I),AL(I)
156 ON B(I) GOSUB 160,162,164,166
158 NEXT I:CLOSE:SCREEN,,0:GOTO16
160 INPUT#1,R(1,I):RETURN
162 FOR J=1 TO AL(I):INPUT#1,Q$(J,I),R(J
,I):NEXT:RETURN
164 INPUT#1,C(I),R(1,I),R(2,I):RETURN
166 INPUT#1,P(I),R(1,I),R(2,I):RETURN
168 '
170 '     SAUVEGARDE
172 '
174 OPEN"0",#1,N$:PRINT#1,S1
176 FORI=1 TO 51
178 PRINT#1,A$(I),PV(I),B(I),AL(I)
180 ON B(I) GOSUB 184,186,188,190
182 NEXT I:CLOSE:SCREEN,,0:GOTO16
184 PRINT#1,R(1,I):RETURN
186 FOR J=1 TO AL(I):PRINT#1,Q$(J,I),R(J
,I):NEXT:RETURN
188 PRINT#1,C(I),R(1,I),R(2,I):RETURN

```

```

190 PRINT#1,P(I),R(1,I),R(2,I):RETURN
192 '
194 '     JOUER
196 '
198 CLS:SCREEN4,0,0:PV=50:N=1
200 CLS:LOCATE0,0,0:A$=A$(N):GOSUB 430:P
V=PV+PV(N):IF PV<0 THEN PV=0:GOTO206
202 PRINT:IF RIGHT$(A$,3)="FIN" THEN 214
204 COLOR 5:PRINT:PRINT"POINTS DE VIE:";
PV:PRINT
206 COLOR 4:IF PV=0 THEN PRINT" VOUS E
TES MORT":GOTO214
208 ON B(N) GOSUB 220,224,230,236
210 N=R(RS,N):V(N)=1
212 IF B(N)<>0 THEN 200
214 LOCATE 9,24:PRINT"UN AUTRE ESSAI? (O
/N)";
216 R$=INPUT$(1):IF R$<>"N" THEN FOR K=1
TO 100:V(K)=0:NEXT:GOTO 198 ELSE 16
218 '
220 LOCATE9,24:COLOR1:PRINT"APPUYEZ SUR
UNE TOUCHE";
222 R$=INPUT$(1):RS=1:COLOR4:RETURN
224 FOR I=1 TO AL(N):COLOR0,4:PRINT CHR$
(I+48):;COLOR 4,0:PRINT" "+Q$(I,N):PRINT
:NEXT
226 R$=INPUT$(1):RS=VAL(R$):IF RS<1 OR R
S>AL(N) THEN 226
228 RETURN
230 LOCATE 9,24:COLOR 1:PRINT"APPUYEZ SU
R UNE TOUCHE";
232 R$=INPUT$(1):IF V(C(N))=1 THEN RS=1
ELSE RS=2
234 COLOR4:RETURN
236 LOCATE 9,24:COLOR 1:PRINT"APPUYEZ SU
R UNE TOUCHE";
238 R$=INPUT$(1):IF P(N)>INT(RND#100) TH
EN RS=1 ELSE RS=2
240 COLOR4:RETURN
242 '
244 '     ROUTINE DE CREATION
246 '
248 COLOR 4:PRINT USING EN$;I;:INPUT A$(
I)
250 COLOR 5:LOCATE 0,8:PRINT PV$;:INPUT
PV(I)
252 COLOR 4:LOCATE 0,10:PRINT TB$:PRINT
254 FOR K=1 TO 4
256 COLOR0,4:PRINT CHR$(48+K):;COLOR 4,0
:PRINT TB$(K):PRINT
258 NEXT K
260 R$=INPUT$(1):R=VAL(R$):IF R=0 OR R>4
THEN 260
262 COLOR 4,0:CONSOLE 9,24:CLS
264 COLOR1,0:LOCATE0,10,0:PRINT"BRANCHEM
ENT ";TB$(R):PRINT:COLOR4
266 ON R GOSUB 308,320,342,360:B(I)=R
268 RETURN
270 '
272 '     ROUTINE DE RECHERCHE
274 '
276 CLS:COLOR4:PRINT USING EN$;I;:PRINT
A$(I):PRINT

```

```

278 COLOR 5:PRINT PV$;PV(I):COLOR4:PRINT
280 PRINT"BRANCHEMENT ";TB$(B(I)):PRINT
282 ON B(I) GOSUB 286,288,292,298
284 RETURN
286 PRINT SB$;R(1,I):RETURN
288 FOR K=1 TO AL(I):COLOR 4:PRINT USING
C$;K$:PRINT Q$(K,I)
290 COLOR 2:PRINT SB$;R(K,I):PRINT:NEXT
K:RETURN
292 PRINT"NO DE LA SITUATION CONDITIONEL
LE:";C(I):PRINT:PRINT SB$
294 PRINT"SI PASSAGE PAR SIT. COND. :";R
(1,I):PRINT
296 PRINT TAB(22)"SINON :";R(2,I):RETURN
298 PRINT"1ere SITUATION DE BRANCHEMENT:
";R(1,I)
300 PRINT" % DE CHANCES (0-100):
";P(I):PRINT
302 PRINT"2eme SITUATION DE BRANCHEMENT:
";R(2,I)
304 RETURN
306 '
308 ' BRANCHEMENT DIRECT
310 '
312 AL(I)=1
314 PRINT SB$;:INPUT R(1,I)
316 RETURN
318 '
320 ' BRANCHEMENT SUR QUESTIONS
322 '
324 LOCATE 0,12
326 PRINT NC$;:INPUT AL(I)
328 CONSOLE 12,24:CLS :CONSOLE24,24:LOCA
TE0,12
330 FOR K=1 TO AL(I)
332 COLOR 4:PRINT USING C$;K$:INPUT Q$(K,
I)
334 COLOR 2:PRINT SB$;:INPUT R(K,I):PRIN
T
336 NEXT K
338 CONSOLE 12,24:RETURN
340 '
342 ' BRANCHEMENT CONDITIONEL
344 '
346 AL(I)=2
348 PRINT"NO DE LA SITUATION CONDITIONEL
LE";:INPUT C(I)

```

```

350 PRINT:PRINT SB$
352 PRINT"SI PASSAGE PAR SIT. COND. :";
INPUT R(1,I):PRINT
354 PRINT TAB(22)"SINON :";:INPUT R(2,I)
356 RETURN
358 '
360 ' BRANCHEMENT ALEATOIRE
362 '
364 AL(I)=2
366 PRINT"1ere SITUATION DE BRANCHEMENT:
";:INPUT R(1,I)
368 PRINT" % DE CHANCES (0-100):
";:INPUT P(I):PRINT
370 PRINT"2eme SITUATION DE BRANCHEMENT:
";:INPUT R(2,I)
372 RETURN
374 '
376 ' INITIALISATIONS
378 '
380 EN$="Enonce de la situation No ###:"
382 PV$="Evolution des points de vie(-0+
):"
384 TB$="TYPE DE BRANCHEMENT : "
386 TB$(1)=" Direct":TB$(2)=" Sur choix "
:TB$(3)=" Conditionel":TB$(4)=" Aleatoir
e"
388 SB$="Situation de branchements:"
390 NC$="Nombre de choix"
392 C$=" Choix No # : "
394 RETURN
396 '
398 ' IMPRESSION
400 '
402 OPEN"O",#2,"LPRT:(40)"
404 PRINT #2,I;A$(I)
406 PRINT #2,"PV.(+0-)";PV(I):PRINT #2,
" "
408 PRINT #2,"BRANCHEMENT";TB$(B(I))
410 ON B(I) GOSUB 416,418,420,422
412 PRINT #2,"-----"
-----"
414 CLOSE:RETURN
416 PRINT #2,"==>",R(1,I):RETURN
418 FOR K=1 TO AL(I):PRINT #2,K;Q$(K,I):
PRINT #2,"==>",R(K,I):NEXT K:RETURN
420 PRINT #2,C(I);"REALISEE==>";R(1,I):
PRINT #2,"NON REALISEE==>";R(2,I):RETUR

```

```

N
422 PRINT #2,P(I);"%==>",R(1,I):PRINT #
2,"SINON ==>",R(2,I):RETURN
424 '
426 '
428 ' TRAITEMENT DES CHAINES
430 '
432 LA=LEN(A$):IF LA<39 THEN PRINT A$:GO
TO 446
434 FOR I=40 TO 1 STEP -1
436 IF ASC(MID$(A$,I,1))<64 THEN B$=LEFT
$(A$,I):U=I:1
438 NEXT
440 PRINT B$;
442 A$=RIGHT$(A$,LA-U)
444 GOTO 432
446 RETURN
448 '
450 ' VERIFICATION DES BRANCHEMENTS
452 '
454 MN$="SITUATION### BRANCHEE PAR CHOIX
# DE###"
456 CLS:COLOR 4
458 PRINT"SITUATIONS BRANCHEES MAIS NON
CREES : "
460 PRINT:COLOR 2:T=0
462 FOR I=1 TO S:FOR J=1 TO AL(I)
464 IF B(R(J,I))>0 OR A$(I)=""THEN 468
466 PRINT USING MN$;R(J,I);J:I:T=T+1
468 NEXT:NEXT
470 PRINT"TOTAL";T:PRINT
472 COLOR 4:PRINT"SITUATIONS CREES MAIS
NON BRANCHEES : "
474 PRINT:COLOR2:T=0
476 FOR I=2 TO S
478 P=0:IF A$(I)=""THEN 490
480 FOR K=1 TO S:FOR J=1 TO AL(K)
482 IF R(J,K)=I THEN P=1:J=AL(K):K=S
484 NEXT:NEXT
486 IF P=1 THEN 490
488 PRINT"SITUATION";I:T=T+1
490 NEXT
492 PRINT"TOTAL";T
494 LOCATE 0,24:COLOR 1
496 PRINT"POUR REVENIR AU MENU,PRESSEZ U
NE TOUCHE";
498 R$=INPUT$(1):GOTO68

```



**COMPR**

Langage : Langage machine + Basic



**PHILIPPE LHOSTE  
GAGNE  
UNE IMPRIMANTE  
EPSON LX80**



**Philippe Lhoste**

**Vingt-quatre ans. Philippe a une maîtrise en informatique et prépare actuellement son DEA dans cette discipline. L'informatique reste en plus une source de loisirs pour lui ainsi que les jeux de rôle.**

Les jeux d'aventure réclament beaucoup d'écrans graphiques. Un écran graphique prend 33 secteurs sur disquette. Il est très intéressant de comprimer les écrans afin de pouvoir en stocker davantage sur une seule disquette. Le compresseur d'image permet de réduire la place de 20 à 85 %, entre 6 et 25 secteurs (les limites théoriques sont de 2 à 31 secteurs selon la complexité de l'image). Le principe de la compression est basé sur le fait que la plupart des images créées par les programmes spécialisés ont de grandes plages de couleurs uniformes. Le programme parcourt l'écran par colonnes de 7 pixels (un octet), et incrémente un compteur tant qu'il rencontre des octets identiques, ou que le compteur n'arrive pas à 255. On code une telle suite par un zéro (repère) puis le nombre d'octets et l'octet lui-même. S'il n'y a qu'un ou deux octets, on ne les code pas. Mais un octet à zéro (00) doit être codé 80 (toujours noir) pour éviter de le confondre avec le code des suites. Enfin, comme la plupart des palettes des programmes graphiques sont constituées de pseudo-couleurs composées de couleurs alternées, le programme examine successivement les lignes impaires puis les lignes paires.

La routine DECOMPR (qui permet de décompresser un écran) peut être facilement isolée des suivantes, et si on remplace alors les appels de sous-programmes par leur code, on a une routine courte et rapide, utilisable pour décompresser des images.

Un menu en Basic a été écrit pour donner un exemple d'utilisation du compresseur et pour faciliter son usage. Il permet l'utilisation du compresseur et la manipulation d'images normales ou compressées. Il est facile d'utilisation. Précisons seulement que pour copier une image compressée par son intermédiaire, il faut la décompresser (D), la recompresser (C) et sauver l'image compressée (E).

Quand une image est affichée, on revient au menu en tapant sur une touche quelconque. On peut ne compresser qu'une partie de l'image. Pour cela déplacez le cadre clignotant avec les touches I, J, K, L (avec ou sans ctrl) et validez par RETURN. Attention à ne pas sortir de l'écran et à ne pas croiser les faisceaux sous peine d'effets bizarres ou même de plantage.

Enfin, on peut manipuler les images avec des opérateurs bouléens et des constantes (en décimal, mais on peut

ajouter des routines de conversion de l'hexadécimal ou du binaire), ou avec d'autres pages graphiques. Je rappelle que le bit 7 agit sur la couleur, et que les bits sont en ordre inverse des pixels. Je rappelle aussi que AND force les bits à zéro, OR force les bits à un et EOR inverse les bits à un. LDA remplit le cadre avec une couleur (avec constante) ou recopie l'autre page graphique. Cette recopie peut être faite de façon plus souple avec l'option déplacement. On commence par définir le cadre comme avant, on peut changer de page ou pas, puis déplacer le cadre. RETURN valide la recopie; ESC l'annule. ■

**Philippe LHOSTE**

\$9200-0000000000000000	0	\$9348-0004080C1014181C	112
\$9208-8080808080808080	1024	\$9350-0105090D1115191D	120
\$9210-0000000000000000	0	\$9358-0105090D1115191D	120
\$9218-8080808080808080	1024	\$9360-02060A0E12161A1E	128
\$9220-0000000000000000	0	\$9368-02060A0E12161A1E	128
\$9228-8080808080808080	1024	\$9370-03070B0F13171B1F	136
\$9230-0000000000000000	0	\$9378-03070B0F13171B1F	136
\$9238-8080808080808080	1024	\$9380-CAC98CBB8B9CCCB	1364
\$9240-2828282828282828	320	\$9388-A940C5E6F0034A85	1110
\$9248-ABABABABABABABAB	1344	\$9390-E660A9008518A960	917
\$9250-2828282828282828	320	\$9398-8519208893A90085	775
\$9258-ABABABABABABABAB	1344	\$93A0-CFABAAB1189506E6	1131
\$9260-2828282828282828	320	\$93AB-18D002E619EBE004	949
\$9268-ABABABABABABABAB	1344	\$93B0-00F1E60BE609A606	1098
\$9270-2828282828282828	320	\$93B8-861EA607E8861FD0	942
\$9278-ABABABABABABABAB	1344	\$93C0-1DE8E8E4099016E6	1126
\$9280-5050505050505050	640	\$93CB-1EA41EC408900CC6	782
\$9288-D0D0D0D0D0D0D0D0	1664	\$93D0-1FA61FE407D015A4	856
\$9290-5050505050505050	640	\$93D8-06841EA61FAB8D00	722
\$9298-D0D0D0D0D0D0D0D0	1664	\$93E0-928526BDC09205E6	1079
\$92A0-5050505050505050	640	\$93E8-85279860A8686868	900
\$92AB-D0D0D0D0D0D0D0D0	1664	\$93F0-F004982048946020	779
\$92B0-5050505050505050	640	\$93F8-9293A90048B118D0	943
\$92BB-D0D0D0D0D0D0D0D0	1664	\$9400-14E618D002E619B1	916
\$92C0-0004080C1014181C	112	\$9408-1885CFE618D002E6	1058
\$92C8-0004080C1014181C	112	\$9410-19B118C6CFA41E91	970
\$92D0-0105090D1115191D	120	\$9418-2620C193A4CFD0F3	1232
\$92D8-0105090D1115191D	120	\$9420-E618D0D09E619D0D5	1355
\$92E0-02060A0E12161A1E	128	\$9428-209293A9FF48A41E	1015
\$92E8-02060A0E12161A1E	128	\$9430-B12620C193A41ED1	990
\$92F0-03070B0F13171B1F	136	\$9438-26D00AA4CF0CFEF0	1313
\$92F8-03070B0F13171B1F	136	\$9440-04E6CFD0ED204894	1141
\$9300-0004080C1014181C	112	\$9448-4C2E94E6CFA4CF00	1270
\$9308-0004080C1014181C	112	\$9450-03901C4BA900A891	729
\$9310-0105090D1115191D	120	\$9458-18E618D002E619A5	908
\$9318-0105090D1115191D	120	\$9460-CF9118E618D002E6	1070
\$9320-02060A0E12161A1E	128	\$9468-1968C884CFD006C9	1083
\$9328-02060A0E12161A1E	128	\$9470-009002A980A00091	812
\$9330-03070B0F13171B1F	136	\$9478-18E618D002E619C6	941
\$9338-03070B0F13171B1F	136	\$9480-CFD0F460A9008506	1063
\$9340-0004080C1014181C	112	\$9488-8507A9278508A9BF	849
		\$9490-850920889320D094	845
		\$9498-20D094AD00C010F5	1014
		\$94A0-8D10C0A203DDB493	1014
		\$94A8-F01CCA10FBA203D8	1120
		\$94B0-8093F017CA10FBC9	1205
		\$94B8-BDD0DAA20385069D	1076
		\$94C0-0060CA10F860D606	878
		\$94C8-4C9594F6064C9594	998
		\$94D0-C606A60720FC9A46	975
		\$94D8-0920FC94E606A406	847
		\$94E0-A90120E994A408A9	924
		\$94E8-4085D8A60920DE93	989
		\$94F0-B12645D89126CAE4	1113
		\$94FB-07D0F26020DE93A4	1118
		\$9500-08B126497F912688	742
		\$9508-C406D0F560209293	1076
		\$9510-A90048A41EB12649	723

\$9518-FF912620C1934C13 905  
 \$9520-95209293A90048A4 879  
 \$9528-1EA5274849608527 647  
 \$9530-B12685D8688527B1 1017  
 \$9538-2605D8912620C193 814  
 \$9540-4C2795209293C608 795  
 \$9548-C60920D09420D094 983  
 \$9550-AD00C010F58D10C0 975  
 \$9558-C9C9D006C607C609 1028  
 \$9560-D0EBC9C8D006E607 1295  
 \$9568-E609D0DEC9CAD006 1286  
 \$9570-C606C608D0D4C9CC 1235  
 \$9578-D006E606E608D0CA 1098  
 \$9580-C98DD00620BB944C 999  
 \$9588-F793C998D0BC60FF 1497

**PROGRAMME  
MENU**

1 REM

COMPRESSEUR D'IMAGES  
 PHILIPPE LHOSTE  
 LE 06/10/85

10 REM

INITIALISATIONS: ADRESSES

20 DECOMPR = 37879:COMPR = 37928:CADRE =  
 38020:M1ANIP = 38157:M2ANIP = 38177:M3A  
 NIP = 38211:AD = 24576

30 DNERR GOTO 5000

40 HIMEM: 37376:PAGE = 1: POKE 230,32:D  
 ISQUE = 1:D\$ = CHR\$(4): PRINT CHR\$(2  
 1)

45 REM

CHARGEMENT DE COMPR  
 SI PAS DEJA EN MEMOIRE

50 IF PEEK (DE) + PEEK (CO) + PEEK (C  
 CA) + PEEK (M1) + PEEK (M2) + PEEK (M  
 3) < > 329 THEN PRINT D\$"BLDADCOMPR"

90 REM

DEBUT: MENU PRINCIPAL

100 TEXT : HOME : INVERSE : HTAB 10: PR  
 INT "COMPRESSEUR D'IMAGES": HTAB 12: PRI  
 NT "PHILIPPE LHOSTE": NORMAL

110 PRINT : PRINT " </> CHANGER  
 DE PAGE. PAGE:": PRINT " <\*> EFFACER L  
 A PAGE.": PRINT " <ESPACE> VOIR L'IMAGE"  
 : PRINT " <ESC> QUITTER LE PROGRAMME.":  
 PRINT " <#> CHANGER DE DISQUE. DISQUE:  
 ": PRINT

120 PRINT " <A> APPELER UNE IMAGE (33 S

ECTEURS).": PRINT " <B> BSAVE L'IMAGE (3  
 3 SECTEURS).": PRINT " <C> COMPRESSER L'  
 IMAGE.": PRINT " <D> DECOMPRESSER UNE IM  
 AGE.": PRINT " <E> ENREGISTRER UNE IMAGE  
 COMPRESSEE."

130 PRINT : PRINT " <M> MANIPULER LES I  
 MAGES.": GOSUB 1000: GOSUB 1500  
 140 VTAB 20: HTAB 10: PRINT "CHOIX? ";;  
 GET R\$:X = FRE (0): REM  
 FRE(0) POUR LIMITER LES CHAINES

145 REM

DETERMINATION CHDIX

150 IF R\$ = "/" THEN PA = 3 - PA: POKE  
 230,PA \* 32: GOSUB 1000: GOTO 140: REM  
 CHANGEMENT DE PAGE ET AFFICHAGE

160 IF R\$ = "\*" THEN CALL - 3086: GOT  
 O 140: REM

EFFACE LA PAGE HGR COURANTE  
 (DEFINIE EN 230)

170 IF R\$ = " " THEN GOSUB 2000: GET R  
 \$: GOTO 100

180 IF R\$ = CHR\$(27) THEN 990

185 IF R\$ = "\*" THEN DI = 3 - DI: GOSUB  
 1500: GOTO 140: REM  
 CHANGEMENT DISQUE ET AFFICHAGE

190 IF R\$ < "A" THEN 140

200 ON ASC (R\$) - 64 GOTO 10000,20000,  
 30000,40000,50000: REM  
 COMMANDES ABCDE

210 IF R\$ = "M" THEN 60000

220 GOTO 140: REM

MAUVAISE TOUCHE, ATTEND UNE AUTRE

980 REM

SOUS-PROGRAMMES

990 HOME : END : REM

FIN DE PROGRAMME

1000 VTAB 5: HTAB 30: INVERSE : PRINT P  
 A: NORMAL : RETURN : REM  
 AFFICHE NUM. PAGE COURANTE

1500 VTAB 9: HTAB 34: INVERSE : PRINT D  
 I: NORMAL : RETURN : REM  
 AFFICHE DISQUE DE TRAVAIL

2000 POKE - 16304,0: POKE - 16302,0:  
 POKE - 16297,0: POKE - 16301 + PA,0: R  
 ETURN : REM  
 AFFICHE ECRAN HGR PLEINE PAGE

2999 REM

ROUTINE DEMANDANT NOM DE FICHIER

3000 PRINT " PAGE ";; INVERSE : PRINT P  
 A: NORMAL : REM  
 RAPPELLE PAGE POUR EVITER ERREURS

3010 PRINT : PRINT "DONNER LE NOM";: IF  
 R\$ = "D" OR R\$ = "E" THEN PRINT " (SAN  
 S LE SUFFIXE .C)": REM  
 SI CHARGE OU SAUVE IMAGE COMPR.

3020 PRINT : PRINT "(? POUR CATALOGUE,  
 RETURN POUR ANNULER)": PRINT  
 3030 INPUT NM\$: IF NM\$ = "?" THEN PRIN  
 T D\$"CATALOG,D"DI: GOTO 3030  
 3040 IF NM\$ = "" THEN POP : GOTO 100:  
 REM  
 PDP CAR SORTIE SS-PGM PAR GOTO

3050 RETURN

5000 TEXT : HOME : PRINT "### ERREUR ##  
 #": POKE 216,0: GET R\$: RUN : REM  
 TRAITEMENT DES ERREURS  
 ASSEZ RUDIMENTAIRE...

9990 REM

TRAITEMENT DES CHOIX ABCDEM

9995 REM

-----  
 10000 HOME : INVERSE : PRINT "APPELER U  
 NE IMAGE (33 SECTEURS)":: NORMAL  
 10010 GOSUB 3000: REM  
 DEMANDE NOM IMAGE

10030 PRINT D\$"BLOAD"NM\$,A\$\*PA \* 2000"  
 ,D"DI: GOSUB 2000: GET R\$: REM  
 CHARGE ET AFFICHE

10040 GOTO 100: REM  
 APRES TOUCHE, REVIENT AU MENU

19995 REM

-----  
 20000 HOME : INVERSE : PRINT "SAUVER L'  
 IMAGE (33 SECTEURS)":: NORMAL

20010 GOSUB 3000

20030 GOSUB 2000: PRINT D\$"BSAVE"NM\$,A  
 \$\*PA \* 2000",L\$1FFB,D"DI

20040 GOTO 100

29995 REM

-----  
 30000 GOSUB 2000: CALL CA: CALL CO:L =  
 PEEK (24) + 256 \* PEEK (25) - AD: REM  
 AFFICHE PAGE, DEMANDE CADRE,  
 COMPRESSE ET CALCULE LONGUEUR

30010 TEXT : HOME : INVERSE : PRINT "CO  
 MPRESSION DE L'IMAGE": NORMAL

30020 PRINT : PRINT "LONGUEUR: "L" OCTE  
 TS, " INT (L / 256 + 1.9)" SECTEURS."

30030 GET R\$: GOTO 100

39995 REM



```
40000 HOME : INVERSE : PRINT "CHARGER U
NE IMAGE COMPRESSEE:"; NORMAL
40010 GOSUB 3000
40020 PRINT D$"BLOAD"NM$.C,A$6000,D"DI
40030 GOSUB 2000: CALL DE: GET R$: REM
DECOMPRESSE ET AFFICHE
```

```
40040 GOTO 100
49995 REM
```

```
50000 HOME : INVERSE : PRINT "SAUVER UN
E IMAGE COMPRESSEE:"; NORMAL
50010 GOSUB 3000
50020 GOSUB 2000: PRINT D$"BSAVE"NM$.C
,A$6000,L"L",D"DI
50030 GOTO 100
59995 REM
```

```
60000 HOME : INVERSE : PRINT "MANIPULAT
IONS D'IMAGES": NORMAL
```

```
60010 PRINT : PRINT " <1> OPERATION AVE
C CONSTANTE.": PRINT " <2> OPERATION AVE
C PAGE ": INVERSE : PRINT 3 - PA: NORMA
L : PRINT " <3> DEPLACEMENT DE LA FENETR
E.": PRINT " <ESC> RETOUR AU MENU": PRIN
T
```

```
60020 GET R$: IF R$ = CHR$(27) THEN 1
00: REM
CHR$(27)=ESC
```

```
60030 IF R$ < "1" OR R$ > "3" THEN 6002
0
```

```
60040 PRINT R$: ON ASC (R$) - 48 GOTO
61000,62000,63000
60499 REM
MENU COMMUN
```

```
60500 PRINT : PRINT "OPERATIONS:"; PRIN
T " <1> EOR": PRINT " <2> AND": PRINT "
<3> OR": PRINT " <4> LDA": PRINT
```

```
60510 GET R$: IF R$ < "1" OR R$ > "4" T
HEN 60510
```

```
60520 PRINT R$:R = ASC (R$) - 48: RETU
RN
```

```
60990 REM
OPERATION AVEC CONSTANTE
```

```
61000 GOSUB 60500: POKE M1 + 10,73 * (R
= 1) + 41 * (R = 2) + 9 * (R = 3) + 169
* (R = 4): REM
POKE INSTRUCTION EN FONCTION DE R
```

```
61010 PRINT : INPUT "MASQUE (0 A 255)?
";M0: POKE M1 + 11,M0
```

```
61020 GOSUB 2000: CALL CA: CALL M1: GET
R$: REM
```

```
AFFICHE PAGE, DEMANDE CADRE
ET FAIT MANIPULATION
```

```
61030 GOTO 100
61990 REM
OPERATION AVEC AUTRE PAGE
```

```
62000 GOSUB 60500: POKE M2 + 24,69 * (R
= 1) + 37 * (R = 2) + 5 * (R = 3) + 165
* (R = 4): REM
```

```
POKE INSTRUCTION EN FONCTION DE R
```

```
62010 GOSUB 2000: CALL CA: CALL M2: GET
R$
```

```
62020 GOTO 100
```

```
62990 REM
DEPLACEMENT CADRE
```

```
63000 PRINT : PRINT "SELECTION CADRE (T
APER RETURN) ";: GET R$: PRINT
```

```
63010 GOSUB 2000: CALL CA: CALL CO: POK
E - 16300,0: POKE - 16303,0: REM
```

```
COMPRESSE IMAGE DANS CADRE
ET REVIENT A PAGE TEXTE
```

```
63020 PRINT "/ POUR CHANGER DE PAGE, RE
TURN SINON";: GET R$
```

```
63030 IF R$ = "/" THEN PA = 3 - PA: POK
E 230,PA * 32
```

```
63040 GOSUB 2000: CALL M3: GET R$
63050 GOTO 100
```

COMPR a été écrit en assembleur BIG-MAC MERLIN. Tout le monde ne possédant pas cet assembleur (aux deux sens du mot), nous avons décidé de le publier en code machine (Hexadécimal). Si vous vous sentez sûr de vous, vous pouvez le rentrer directement avec le moniteur (CALL-151). Nous vous proposons un petit programme (ELM) qui permet de rentrer les octets en hexadécimal par groupes de huit. Le programme fait la somme de ces huit octets et vous demande de taper celle qui est inscrite sur le listing (dernière colonne). Si le chiffre ne correspond pas, vous avez commis une erreur dans cette ligne. Il faut la recommencer. Vous pouvez arrêter le programme par CTRL-C et sauver par BSAVE «COMPR».A9200,L038F. L'adresse de début peut être rentrée en hexadécimal sous la forme 9200. Vous avez en prime les routines de conversion hexa/décimal et décimal/héxa.

```
10 REM SAVE ELM
20 HOME
30 REM -----
40 INPUT "ADRESSE DE DEBUT ";X$
50 IF LEFT$(X$,1) = "$" THEN X$ = M1
0$(X$,2,25): GOSUB 210:D = X: PRINT D:
GOTO 65
60 D = VAL (X$):X = D: GOSUB 180: PRINT
"$" RIGHT$( "000" + X$,4)
65 VTAB 22: HTAB 7: PRINT ".....
...."
70 FOR D = D TO 65535 STEP 8
80 VTAB 23:X = D: GOSUB 180: PRINT "$"
RIGHT$( "000" + X$,4) "-";
90 INPUT " ";L$
100 IF LEN (L$) < > 16 THEN PRINT C
HR$(7);: GOTO 80
110 VTAB 23: HTAB 26: INPUT "SOMME: ";S
1:S = 0
120 FOR B = 1 TO 15 STEP 2:X$ = MID$(
L$,B,2): GOSUB 200: POKE (B - 1) / 2 + D
,X:S = S + X: NEXT B
130 IF S < > S1 THEN PRINT CHR$(7);
: GOTO 80
140 PRINT
150 NEXT
160 END
170 REM ---- DECIM --> HEXA
180 X = X + 65536 * (X < 0):X$ = "":J =
16: FOR I = 0 TO 1:A = X - J * INT (X /
J):X = INT (X / J):I = (X = 0):X$ = C
HR$(48 + A + 7 * (A > 9)) + X$: NEXT
190 RETURN
200 REM ---- HEXA --> DECIM
210 J = 1:X = 0: FOR I = LEN (X$) TO 1
STEP - 1:A$ = MID$(X$,I,1):X = X + J
* (VAL (A$) + (A$ > = "A" AND A$ < =
"F")) * (ASC (A$) - 55):J = J * 16: NEX
T
220 RETURN
```





TEXAS  
INSTRUMENTS

# MICRO MORSE

Langage : Basic Etendu



**JEAN BOULNOIS  
GAGNE  
UN MONITEUR  
EUREKA MC14**

## Jean Boulnois

Cinquante-huit ans, agent technique électricien à la recherche d'un emploi. Il programme depuis deux ans, sans doute sous l'influence de son fils, analyste informaticien.

Le but de ce programme est d'apprendre l'alphabet morse (lettres et chiffres). La lecture seule est possible et pas l'émission. Quatre sous-programmes sont proposés à votre sagacité : l'apprentissage (la frappe d'une touche entraîne l'apparition à l'écran du code morse par points et traits et, en même temps, l'émission sonore concordante), un contrôle des connaissances lettre par lettre sans limitation du temps de réponse (il faut répondre à l'émission sonore en frappant la touche correspondante), un exercice de reconnaissance des lettres et des chiffres par séries de douze et enfin un dernier exercice lors duquel on a la possibilité de régler la vitesse d'émission des sons. ■

Jean, Charles BOULNOIS

```
10 CALL CLEAR
20 DIM A$(90),A1$(21),A2$(12):: RANDOMIZ
E :: CALL SCREEN(2)
30 IMAGE "####"
40 !
50 ! MORSE, CHIFFRES, ALPHABET
60 !
70 A$(48)="44444"
80 A$(49)="14444"
90 A$(50)="11444"
100 A$(51)="11144"
110 A$(52)="11114"
120 A$(53)="11111"
130 A$(54)="41111"
140 A$(55)="44111"
150 A$(56)="44411"
160 A$(57)="44441"
170 A$(65)="14"
180 A$(66)="4111"
190 A$(67)="4141"
200 A$(68)="411"
210 A$(69)="1"
```

```
220 A$(70)="1141"
230 A$(71)="441"
240 A$(72)="1111"
250 A$(73)="11"
260 A$(74)="1444"
270 A$(75)="414"
280 A$(76)="1411"
290 A$(77)="44"
300 A$(78)="41"
310 A$(79)="444"
320 A$(80)="1441"
330 A$(81)="4414"
340 A$(82)="141"
350 A$(83)="111"
360 A$(84)="4"
370 A$(85)="114"
380 A$(86)="1114"
390 A$(87)="144"
400 A$(88)="4114"
410 A$(89)="4144"
420 A$(90)="4411"
430 !
440 ! PRESENTATION
450 !
460 CALL CHAR(136,"7F070000000000")
470 CALL CHAR(137,"FFFF3F03000000")
480 CALL CHAR(138,"FFFFFFFF3F030000")
490 CALL CHAR(139,"FFFFFFFF3F03")
500 CALL CHAR(140,"00FCFFFFFFFF")
510 CALL CHAR(141,"000000FCFFFFFFFF")
520 CALL CHAR(142,"0000000000FCFFFF")
530 CALL CHAR(143,"000000000000E0FE")
540 CALL COLOR(14,7,2):: XA=1
550 FOR I=1 TO 11 :: IF I>=5 THEN XA=XA+4
560 FOR J=XA TO 32 STEP 4 :: CALL HCHAR(I,J,136):: CALL HCHAR(I,J+1,137):: CALL HCHAR(I,J+2,138):: CALL HCHAR(I,J+3,139):: NEXT J :: NEXT I
570 XA=0 :: FOR I=7 TO 24 :: XA=XA+4 :: IF XA>=32 THEN XA=32
580 FOR J=1 TO XA STEP 4 :: CALL HCHAR(I,J,140):: CALL HCHAR(I,J+1,141):: CALL HCHAR(I,J+2,142):: CALL HCHAR(I,J+3,143)
590 NEXT J :: NEXT I
600 X=50 :: AA=1
610 DATA "MICRO MORSE"
```

```
620 DATA "CE PROGRAMME PERMET D'APPRENDRE LA PHONETIQUE DE L'ALPHABET MORSE...D'EN CONTROLER SA CONNAISSANCE...SA RAPIDITE DE SAISIE"
630 DATA "...C'EST AUSSI UN BON EXERCICE POUR LA MAITRISE DU CLAVIER ... BON COURAGE !..."
640 DATA " "
650 DATA "0"
660 READ B$
670 IF B$="0" THEN B30
680 FOR I=1 TO LEN(B$)
690 CALL SPRITE(#AA,ASC(SEG$(B$,I,1)),12,96,256,-7.5,-28)
700 IF AY THEN B00
710 B=ASC(SEG$(B$,I,1))
720 FOR J=1 TO LEN(A$(B))
730 A=VAL(SEG$(A$(B),J,1))
740 CALL SOUND(A*X,1800,0,-1.5)
750 NEXT J :: IF I>=2 THEN B70
760 CALL COINC(#AA,41,57,17,V):: IF V=0 THEN B70
770 CALL MOTION(#AA,0,0):: GOTD B00
780 CALL COINC(#AA,#AA-1,19,V):: IF V=0 THEN B70
790 CALL MOTION(#AA,0,0)
800 AA=AA+1 :: IF AA=19 THEN AA=1
810 NEXT I :: IF AY=1 THEN B60
820 FOR I=1 TO 300 :: NEXT I :: CALL DEL SPRITE(ALL):: AY=1 :: GOTD B60
830 CALL DELSPRITE(ALL)
840 !
850 !
860 !
870 FOR I=1 TO 100 :: NEXT I :: CALL MANNIFY(2):: CALL CHAR(96,RPT$("F",16)):: CALL CHAR(128,"FF003C2424E700FF")
880 !
890 ! MENU
900 !
910 CALL CD
920 DISPLAY AT(4,1):"1 APPRENDRE L'ALPHABET"
930 DISPLAY AT(7,1):"2 LECTURE PHONIQUE D'UN"
940 DISPLAY AT(8,4):"CARACTERE ET COMPARAISON"
950 DISPLAY AT(9,4):"PAR FRAPPE DU CARAC"
```

```

TERE"
960 DISPLAY AT(12,1):"3 LECTURE PHONIQUE
E D'UN"
970 DISPLAY AT(13,4):"GROUPE DE CARACTER
ES ET"
980 DISPLAY AT(14,4):"COMPARAISON DE LA
FRAPPE"
990 DISPLAY AT(17,1):"4 IDENTIQUE A 3 P
LUS CHOIX"
1000 DISPLAY AT(18,4):"DE LA VITESSE D'E
MISSION"
1010 DISPLAY AT(22,1):"PRESSEZ AU CHOIX
1-2-3-4"
1020 CALL COL
1030 CALL KEY(0,K,S):: IF K<49 OR K>52 T
HEN 1030
1040 CALL CD
1050 ON K-48 GOTO 1090,1200,1490,1670
1060 !
1070 ! S/P1
1080 !
1090 DISPLAY AT(5,4):"APPRENONS L'ALPHAB
ET"
1100 DISPLAY AT(6,4):"-----
--"
1110 DISPLAY AT(18,1):"PRESSEZ UNE TOUCH
E DE[A--Z]"
1120 DISPLAY AT(20,1):"DU DE [ 0--9 ]"
1130 DISPLAY AT(22,1):"POUR QUITTER S/P1
: (ENTER)"
1140 CALL COL
1150 GOSUB 1880 :: DISPLAY AT(12,1):: CA
LL SPRITE(#1,K,16,81,48)
1160 SP=1 :: KN=K :: L=12 :: CALL COL1 :
: GOSUB 2440 :: GOTO 1150
1170 !
1180 ! S/P2
1190 !
1200 A1$(3)="MAINTENANT J'EMETS, DE FACON
"
1210 A1$(5)="ALEATOIRE, UN CARACTERE. "
1220 A1$(7)="A VOUS DE L'IDENTIFIER EN"
1230 A1$(9)="FRAPPANT LA TOUCHE CORRES-"
1240 A1$(11)="-PONDANTE ! LA COMPARAISON
"
1250 A1$(13)="ET LE DECOMPTE DE CONCORDA
N-"
1260 A1$(15)="-CE S'AFFICHERONT !"
1270 A1$(17)=" "
1280 A1$(19)=" "
1290 A1$(21)="POUR CONTINUER (FCTN
6)"
1300 GOSUB 1930
1310 CALL CD
1320 DISPLAY AT(16,22):"^"
1330 DISPLAY AT(17,22):"I"
1340 DISPLAY AT(18,1):"POUR CONTINUER PR
ESS I"
1350 DISPLAY AT(22,1):"POUR QUITTER S/P2
(ENTER)"
1360 CP1,CP2=0 :: SP=2
1370 CALL CDI :: CALL COL
1380 L=20 :: CALL COL2 :: CALL DELSPRITE
(ALL)
1390 GOSUB 2280
1400 GOSUB 2440
1410 GOSUB 1880
1420 CALL SPRITE(#1,K,16,81,57,#2,KN,6,8
1,185):: CALL COL1
1430 CP1=CP1+1 :: DISPLAY AT(7,16)SIZE(4
):USING 30:CP1
1440 IF K=KN THEN CP2=CP2+1 :: DISPLAY A
T(7,10)SIZE(4):USING 30:CP2
1450 GOSUB 1880 :: IF K<>KN THEN 1450 EL
SE 1380
1460 !
1470 ! S/F3
1480 !
1490 A1$(3)="A PRESENT J'EMETS DE FACON"
1500 A1$(5)="ALEATOIRE UN GROUPE DE DOUZ
E"
1510 A1$(7)="CARACTERES SEPARES CHACUN"
1520 A1$(9)="D'UN BLANC. IDENTIFIEZ CE CA
-"
1530 A1$(11)="-RACTERE PAR LA TOUCHE COR
-"
1540 A1$(13)="-RESPONDANTE...ATTENTION!
JE"
1550 A1$(15)="N'ATTENDS PAS VOTRE REPONS
E"
1560 A1$(17)="POUR CONTINUER D' EMETTRE
!"
1570 A1$(19)=" "
1580 A1$(21)="POUR COMMENCER (FCTN
6)"
1590 GOSUB 1930
1600 SP=3 :: VT=150
1610 GOSUB 2350
1620 GOSUB 1980
1630 GOSUB 2100 :: GOTO 1620
1640 !
1650 ! S/P4
1660 !
1670 A1$(3)="VOUS CONNAISSEZ MAINTENANT"
1680 A1$(5)="LE MORSE SUR LE BOUT DES ..
"
1690 A1$(7)="DOIGTS...ALDRS ESSAYEZ DE +
"
1700 A1$(9)="EN + VITE !VOUS POUVEZ ENTR
E"
1710 A1$(11)="CHAQUE EMISSION DE GROUPE"
1720 A1$(13)="MODIFIER LA VITESSE PAR LE
S"
1730 A1$(15)="TOUCHES + DU -"
1740 A1$(17)=" "
1750 A1$(19)=" "
1760 A1$(21)="POUR CONTINUER (FCTN
6)"
1770 GOSUB 1930
1780 SP=4 :: VT=100 :: VTU=1
1790 GOSUB 2350
1800 DISPLAY AT(16,1):"POUR VITESSE (+ 0
U -)"
1810 CALL SPRITE(#1,ASC(STR$(VTU)),12,11
7,225)
1820 GOSUB 2100
1830 GOSUB 1980
1840 GOSUB 2100 :: GOTO 1830
1850 !
1860 !
1870 !
1880 CALL KEY(0,K,S):: IF K=13 THEN CALL
DELSPRITE(ALL):: GOTO 910 ELSE IF (K<48
OR K>57)AND(K<65 OR K>90)THEN 1880
1890 RETURN
1900 !
1910 !
1920 !
1930 FOR I=3 TO 21 STEP 2 :: DISPLAY AT(
I,1):A1$(I):: NEXT I :: CALL COL
1940 CALL KEY(0,K,S):: IF K<>12 THEN 194
0 ELSE RETURN
1950 !
1960 !
1970 !
1980 FOR H=1 TO 12 :: GOSUB 2280 :: A2$(
H)=CHR$(KN):: GOSUB 2440
1990 FOR TPS=1 TO VT :: CALL KEY(0,KT,S)
2000 IF (KT<48 OR KT>57)AND(KT<65 OR KT)
90)THEN 2010 ELSE 2030
2010 NEXT TPS
2020 DISPLAY AT(12,H)SIZE(1):CHR$(96)::
GOTO 2040
2030 DISPLAY AT(12,H)SIZE(1):CHR$(KT)
2040 IF KT=KN THEN CP2=CP2+1
2050 CP1=CP1+1
2060 NEXT H :: RETURN
2070 !
2080 !
2090 !
2100 CALL KEY(0,K,S)
2110 IF K=13 THEN CALL COL2 :: CALL DELS
PRITE(#1):: GOTO 910
2120 IF K<>6 THEN 2160
2130 DISPLAY AT(7,10)SIZE(4):USING 30:CP
2
2140 DISPLAY AT(7,16)SIZE(4):USING 30:CP
1
2150 FOR I=1 TO 12 :: DISPLAY AT(12,I+16
):A2$(I):: NEXT I
2160 IF SP=4 AND K=43 OR K=45 THEN 2170
ELSE 2210
2170 IF (VT)=100 AND K=45)OR(VT<=20 AND
K=43)THEN 2100
2180 IF K=43 THEN VT=VT-10 :: VTU=VTU+1
2190 IF K=45 THEN VT=VT+10 :: VTU=VTU-1
2200 CALL SPRITE(#1,ASC(STR$(VTU)),12,11
7,225)
2210 IF K<>12 THEN 2100
2220 CALL COL2 :: DISPLAY AT(12,1)SIZE(1
2):: DISPLAY AT(12,17)SIZE(12):: CALL CO
L1
2230 FOR I=1 TO 100 :: NEXT I
2240 RETURN
2250 !
2260 !
2270 !
2280 R=INT(RND#4)+1

```

```

2290 IF R=1 THEN KN=INT(RND*(57-48+1))+4
8
2300 IF R=2 AND R<=4 THEN KN=INT(RND*(9
0-65+1))+65
2310 RETURN
2320 !
2330 !
2340 !
2350 CALL CD :: DISPLAY AT(18,1):"POUR C
OMPARER (FCTN 8)"
2360 DISPLAY AT(20,1):"POUR CONTINUER
(FCTN 6)"
2370 DISPLAY AT(22,1):"POUR QUITTER
(ENTER)"
2380 CALL CDI :: CALL COL :: CALL COL1
2390 CP1,CP2=0
2400 RETURN
2410 !
2420 !
2430 !
2440 X=70 :: CL=8
2450 FOR I=1 TO LEN(A$(KN))
2460 A=VAL(SEG$(A$(KN),I,1))
2470 IF A=1 THEN B$=CHR$(32)&CHR$(96)
2480 IF A=4 THEN B$=CHR$(32)&RPT$(CHR$(9

```

```

6),2)
2490 CALL SOUND(A*X,1800,0,-1,0)
2500 IF SP=3 OR SP=4 THEN 2540
2510 DISPLAY AT(L,CL):B$
2520 IF A=1 THEN CL=CL+2
2530 IF A=4 THEN CL=CL+3
2540 NEXT I
2550 RETURN
2560 !
2570 !
2580 ! CADRE
2590 SUB CD
2600 CALL CLEAR :: CALL CHARSET :: CALL
COLOR(13,1,1)
2610 CALL HCHAR(1,1,128,32)
2620 CALL HCHAR(24,1,128,32)
2630 CALL VCHAR(2,1,128,22)
2640 CALL VCHAR(2,32,128,22)
2650 SUBEND
2660 !
2670 ! CADRAGE INTERIEUR
2680 !
2690 SUB CDI
2700 CALL HCHAR(5,2,128,30):: CALL HCHAR
(9,2,128,30):: CALL HCHAR(14,2,128,30)

```

```

2710 CALL VCHAR(6,11,128,3):: CALL VCHAR
(2,16,128,12):: CALL VCHAR(2,17,128,12):
: CALL VCHAR(6,22,128,3)
2720 DISPLAY AT(3,5)SIZE(4):"VOUS" :: DI
SPLAY AT(3,18):"ORDINATEUR"
2730 DISPLAY AT(7,1)SIZE(7):"VALIDES" ::
DISPLAY AT(7,23):"EMIS"
2740 SUBEND
2750 !
2760 !
2770 !
2780 SUB COL :: CALL COLOR(13,10,5)
2790 FOR I=1 TO 8 :: CALL COLOR(I,10,1):
: NEXT I
2800 SUBEND
2810 !
2820 !
2830 !
2840 SUB COL1 :: CALL COLOR(9,6,1)
2850 SUBEND
2860 !
2870 !
2880 !
2890 SUB COL2 :: CALL COLOR(9,1,1)
2900 SUBEND

```

## AMSTRAD

# ENVAHISSEUR

Langage : L.M. + Basic



**SYLVAIN KOUBDJANIAN  
GAGNE  
UN MONITEUR  
EUREKA MC14**

**Sylvain Koubdjanian**  
Vingt-trois ans, célibataire, Sylvain est technicien de maintenance. Il programme depuis trois ans et puise ses loisirs dans l'informatique, l'électronique et la robotique domestique.

Il s'agit d'un jeu ne présentant aucune difficulté de compréhension. Les déplacements s'effectuent par les touches «0» et «ENTER» du pavé numérique et le tir par la barre d'espace. La saisie est un peu plus délicate. Il va falloir suivre scrupuleusement les instructions suivantes : tout d'abord, et après chaque «reset», il est impératif de taper : MEMORY &5FFF, ceci afin de réserver les adresses supérieures aux données en binaire. Voici ensuite la procédure.

- 1 - Tapez le petit programme intitulé «ELM» (Ecriture en Langage Machine).
- 2 - Sauvegardez : SAVE «ELM».
- 3 - RUN «ELM». Il faudra commencer par le programme intitulé «CODES». Une adresse de départ vous est demandée. Dans le cas présent, ce sera 7530 (hex). La saisie s'effectue par groupes de huit octets. La somme des huit octets saisis vous est alors demandée : pour l'adresse de départ 7530 (hex), la réponse sera 630. Si vous vous êtes trompé lors de la frappe des codes, un message d'erreur s'affichera et vous n'aurez qu'à recommencer. La procédure se répète de façon identique et automatique pour toutes les adresses suivantes.
- 4 - Vous exécutez exactement la même opération avec le programme «TABLE», dont la première adresse est 6000 (hex).
- 5 - Tapez NEW.
- 6 - Entrez le programme principal

«SPACE INVADERS».

7 - Sauvegardez impérativement le tout en observant la progression suivante : SPEED WRITE 1 «RETURN» SAVE «SPACE INVADERS» «RETURN» SAVE «CODES»,B,&7530,2100 «RETURN» SAVE «TABLE»,B,&6000,1200 «RETURN».

Si vous êtes prudent, vous sauvegarderez de temps à autre votre travail, et de toute façon, avant chaque RUN. Après tous ces efforts, vous aurez bien mérité d'abattre quelques extra-terrestres envahissants. ■

**Sylvain KOUBDJANIAN**

```

i ' save "elm
1000 'Entree des codes-----
1010 INPUT "adresse de depart (hex)";a$a=VA
L("&"a$)
1020 PRINT:PRINT "adresse courante ";HEX$(a)

```

```

:PRINT "-)";
1030 LINE INPUT a$:IF LEN(a$)<>16 THEN PRINT
CHR$(7)::PRINT "Refaites l'entree !!!":GOTO
1020
1040 INPUT "somme " :s1:s=0
1050 FOR b=1 TO 15 STEP 2:POKE a,VAL("%"+MID
$(a$,b,2)):s=s+PEEK(a):a=a+1:NEXT b
1060 IF s<>s1 THEN PRINT CHR$(7)::a=a-8:PRIN
T "erreur!!!!!"
1070 GOTO 1020

```

## CODES

```

7530 CD0B7DC921300106 630
7538 06AF772310FC0E00 617
7540 CD34760E05CD3476 769
7548 3E0A326001AF3204 448
7550 0132079132100132 176
7558 0D01322001325F01 243
7560 3E3C3201014FCD70 570
7568 7B3E023200013EFE 554
7570 3211013E14321501 222
7578 CDF7BCDB77ACD98 1445
7580 7B2150612216012A 432
7588 1601235E06000E00 172
7590 1600C5D5CD0279D1 969
7598 C13A1501FE00200E 573
75A0 ED5FE616C6143215 873
75A8 013E013210013A20 221
75B0 01FE32C82A1601BE 760
75B8 3B0523232216010C 200
75C0 3E01B9200CC5D5CD 907
75C8 4377CD2077D1C10E 958
75D0 00043E01B82009C5 489
75D8 D5C8CD7AD1C10600 1153
75E0 143E04BA2009D5C5 723
75E8 CDE976C1D116001D 1009
75F0 2016C5D5CDF57A21 1069
75F8 7463CDAABC1C12A 1222
7600 1601235EAF321401 398
7608 3A08B5C87F28063E 68E
7610 03325F01C9C50601 554
7618 0E010D20FD10F9C1 771
7620 C392753A60014FC6 890
7628 05326001E5C5CD34 835
7630 76C1E1C90603CD51 1032
7638 760C0CCD517605CD 756
7640 51760D0DCD51760C 641
7648 05CD517605CD5176 818
7650 C9C53E0BCD927CC1 1139
7658 C90603CD76760C0C 675
7660 CD767605CD76760D 900
7668 0DCD76760C05CD76 794
7670 7605CD76760C9C5AF 1137
7678 CD927CC1C9213001 951
7680 0606113CC07EFE00 661
7688 20072313131310F5 392

```

```

7690 C9ED534001C5E55E 1108
7698 CB23160021706119 527
76A0 5E23562A40010607 335
76A8 0E031A7723130D20 261
76B0 F9C501FD0709C110 925
76B8 EFE1C1ED58400113 1069
76C0 1313ED5340012310 474
76C8 CCC921340106057E 628
76D0 FE0A3811D60A77F5 925
76D8 7DFE32CC2376F105 1032
76E0 C82B3418EA2B10E7 843
76E8 C93A1001FE00CB3A 788
76F0 11013C3CFE962013 593
76F8 4F060821C060CD4E 697
7700 7C3EFE321101AF32 733
7708 1001C93211014F06 371
7710 08219060CD4E7C21 721
7718 8863CDAABC31F7B 1147
7720 3A0701320A01ED4B 439
7728 0801ED430B01ED5F 657
7730 4FCD6777ED4B0B01 830
7738 ED430B013A0A0132 432
7740 0701C93A0D01320A 341
7748 01ED4B0E01ED430B 643
7750 013A01014FCD6777 567
7758 ED4B0B01ED430E01 643
7760 3A0A01320D01C93A 392
7768 0A01FE00204FCD96 731
7770 77ED430B0105050D 458
7778 CD87770C0CCD8777 942
7780 3E01320A011845C5 414
7788 3E0ACD927CC1C9C5 1138
7790 AFC927CC1C97921 1198
7798 62621E327EC60BB9 796
77A0 30072B2B1D20F5F1 688
77A8 C9798E300218F3ED 1066
77B0 5FE60786C6024F23 780
77B8 7EC60B47C9ED4B0B 930
77C0 0104ED430B0178FE 695
77C8 C7CAE278ED4B0B01 1071
77D0 CD177CFE002037ED 930
77D8 4B0B01050DC5CD17 530
77E0 7CC1FE0020280C0C 667
77E8 CD177CFE00201FED 906
77F0 4B0B01CD87770D05 564
77F8 CD87770CCD8F770C 950
7800 CDB7770505CCD8F77 936
7808 0D0CDB8F77C9FE08 956
7810 CAA378FE09200921 822
7818 150134CDFE7918CF 885
7820 FE0638CBFE0A28C7 1022
7828 216A63CDAABC3E1C 891
7830 CD5ABB3E07CD5ABB 1033
7838 3E0FCD5ABB3E1ACD 852
7840 5ABB3E0A06FF0EFF 879
7848 0D20FD10F93D20F4 900
7850 3E1CCD5ABB3E07CD 846
7858 5ABBAFCD5ABBAFCD 1314

```

```

7860 5ABB3A6001D60530 699
7868 0AF1F1F1F13E0232 1088
7870 5F01C93260014FCD 728
7878 5976CDE278ED4B08 1078
7880 0104ED430B01CDE2 752
7888 78ED4B0E0104ED43 755
7890 0B01CDE278AF3207 795
7898 01320D013A01014F 204
78A0 C3707BED4B0B010D 767
78A8 0D1E05C5D5AFCD92 984
78B0 7CD1C10C1D20F40D 856
78B8 0DED43020116021E 374
78C0 03ED4B020104ED43 826
78C8 0201C5D5AFCD927C 1063
78D0 D1C10D1D20F41520 773
78D8 E6040C0CAFCD927C 908
78E0 1B00ED4B0B0105CD 558
78E8 BF77050DCD8F770C 759
78F0 0CCD8F7705CD8F77 951
78F8 0D0DCD8F77AF320A 728
7900 01C93A0401FE012B 560
7908 473A04B5C87FC821 877
7910 6063CDAABC3A0101 818
7918 C6054F06BF1607D5 721
7920 C5CD177CC1C5FE01 1194
7928 300D3E09CD927CC1 800
7930 D1051520EA180EFE 793
7938 0430EF0688ED4305 790
7940 01C1F11857ED4305 855
7948 013E013204011830 191
7950 ED4B050178C60747 714
7958 AFC927CED4B0501 968
7960 0578FE072016C3FE 889
7968 79ED4B05011608D5 682
7970 C5AFCD927CC1D104 1253
7978 1520F4C9ED430501 808
7980 ED4B0501C5CD177C 867
7988 C1FE00287DFE08CA 1076
7990 147AFE04CA337AFE 1029
7998 0ACA4A7AFE043011 731
79A0 ED44C60421340186 727
79A8 77C5CDCA76CD7D76 1289
79B0 C12101621E3278D6 739
79B8 08BE380623231D20 391
79C0 F8C978BE3004D608 1033
79C8 18F2792BD60C3006 710
79D0 7EC60BB93009BE38 823
79D8 062378D60818DD79 749
79E0 BE300218F44E36FF 895
79E8 234636FFC5217E63 869
79F0 CDAABCC121C060CD 1282
79F8 4E7C21200134CD69 630
7A00 7921150135AF3204 458
7A08 01C9ED4B05013E09 591
7A10 CD927CC978C60747 1072
7A18 160A3A05011E033D 190
7A20 4FC5D5AFCD927CD1 1348
7A28 C10C1D20F4051520 568

```

7A30	E918CBCDFE792133	1124
7A3B	013434CDA76CD7D	960
7A40	76CD25BD3A1101C3	820
7A4B	F876ED4B050103ED	924
7A50	430501211501341B	204
7A5B	A579320201320301	393
7A60	06BFC0997A3A0201	738
7A6B	C6123203014F06BF	546
7A70	CD997A160806B33A	753
7A7B	02014F1E18CDA97A	632
7A80	051520F3160C3A02	395
7A8B	01C60C924FAFB2B2	871
7A90	5FCDA97A051520EE	887
7A9B	C9160C1E06CDA97A	767
7AA0	3A03014F051520F3	442
7AA8	C9C5D53E08CD927C	1156
7AB0	D1C10C1D20F3C93A	977
7AB8	5E01FE5AD00E10CD	882
7AC0	597A0E44CD597A0E	723
7ACB	78CD597AC93AEBB4	1210
7AD0	CB7720153AECB4CB	1052
7ADB	7FCB3A0101FE00CB	841
7AE0	3D3201014FCD707B	632
7AEB	C93A0101FE96C83C	925
7AF0	3201011BEF210062	446
7AFB	06323A00014F7EFE	574
7B00	FF280581FE98303F	946
7B08	232310F221006206	465
7B10	327EFEFF28028177	975
7B1B	232310F5CD987B3A	869
7B20	0401FE00C81E08ED	734
7B2B	4B0501C5D5CD177C	843
7B30	FE00D1C12005041D	726
7B3B	20F1C9FE0530F7FE	1282
7B40	04CA337AC39C79CD	1056
7B4B	D97B21016206327E	654
7B50	C60B77FEC02009F1	1053
7B5B	F1F13E01325F01C9	892
7B60	232310EB3A0001ED	617
7B6B	44320001CD987BC9	800
7B70	21B0CF0600CB39F5	879
7B7B	09F1381711F0600E	696
7B80	0706061A77132310	234
7B8B	FAC501FA0709C10D	920
7B90	20EFC911206118E7	873
7B9B	DD2100621E142100	435
7BA0	60229061CDBE7B1E	919
7BAB	14213060229061CD	677
7BB0	BE7B1E0A21606022	612
7BB8	9061CDBE7BC9DD7E	1307
7BC0	00DD4601DD23DD23	804
7BCB	FEFF28094F2A9061	920
7BD0	D5CD4E7CD11D20E6	1120
7BDB	C9DD2100621E32DD	854
7BE0	7E00DD4601DD23DD	895
7BEB	23FEFF28094F21C0	897
7BF0	60D5CD4E7CD11D20	986
7BF8	E6C92100623A5E01	715

7C00	470E05AF1E0A7723	459
7C08	7023C60C1D20F77B	785
7C10	C610470D20EDC9CD	973
7C18	677C7E382506040E	470
7C20	0017CB191710FA79	661
7C2B	B71F1F1F1F47CB4F	660
7C30	2009CB97CB502007	717
7C3B	C8BFC9CBD718F5CB	1437
7C40	CFC906040E001717	478
7C4B	CB1910FA18D9E5CD	1169
7C50	677CD10E0B06061A	496
7C5B	77132310FAC501FA	887
7C60	0709C10D20EFC9AF	869
7C6B	CB38CB1FCB38CB1F	986
7C70	CB38CB1FB71F1F1F	769
7C7B	1F1F21B0B7041150	555
7C80	001910FD1100083C	379
7C8B	471910FDCB39F509	879
7C90	F1C9F5AFCB381FCB	1355
7C9B	381FCB381FB71F1F	622
7CA0	1F1F1F21B0B70411	506
7CAB	50001910FD11000B	399
7CB0	3C471910FDF1CB39	926
7CBB	F509F15F7E382606	816
7CC0	040E00CB38CB11CB	703
7CCB	2110F859CB59200D	723
7CD0	CBA9CB6B200BCB99	1081
7CDB	E655B177C9CBE918	1272
7CE0	F1CBD918F306040E	952
7CEB	00CB21CB38CB1110	734
7CF0	F859CB51200DCBA1	1030
7CFB	CB63200CB91E6AA	1093
7D00	B118D8CBE11BF1CB	1313
7D0B	D118F3210060062B	651
7D10	0E00CD4E7C213060	598
7D1B	06480E00CD4E7C21	532
7D20	60600E00066BCD4E	599
7D2B	7C219060068B0E00	553
7D30	CD4E7C9000000000	608
7D3B	0000000000000000	0
7D40	0000000000000000	0
7D4B	0000000000000000	0
7D50	0000000000000000	0
7D5B	0000000000000000	0
7D60	0000000000000000	0

**TABLES**

6000	0000C0C000000040	448
600B	40B080000040C0C0	768
6010	80000000C0C00000	512
601B	000040B000000000	192
6020	80400000004040B0	448
602B	8000000000000000	128
6030	0000040B00000000	12

603B	0C0C000000004040B	40
6040	080000040C0C0800	44
604B	0000040B00000000	12
6050	0804000000004040B	28
605B	0800000000000000	8
6060	004400008B000000	204
606B	8B440000008BCCCC	748
6070	4400008B8BCC4400	612
607B	00CCCCCCCC000000	816
6080	CCCC000000008B44	612
608B	000000CC8B44CC00	612
6090	0000000000000000	0
609B	0000000000003030	96
60A0	0000001010002000	64
60AB	0030303030000000	192
60B0	3030000000000000	96
60BB	0000000000000000	0
60C0	0000000000000000	0
60CB	0000000000000000	0
60D0	0000000000000000	0
60DB	0000000000000000	0
60E0	0000000000000000	0
60EB	0000000000000000	0
60F0	0000140000000000	20
60FB	1400000000007C2B	184
6100	00000014FCBC0000	460
610B	007CFCFC2B00007C	792
6110	FCFC2B00143C3C3C	744
611B	3C00000000000000	60
6120	0000002B00000000	40
612B	002B0000000014BC	248
6130	000000007CFC2B00	416
613B	0014FCFCBC000014	732
6140	FCFCBC00003C3C3C	872
614B	3C2B000000000000	100
6150	0E401A2024102D0B	241
615B	3304000000000000	55
6160	0000000000000000	0
616B	0000000000000000	0
6170	70628B62A062B862	984
617B	D062EB6200631863	858
6180	30634B6300000000	318
618B	0000000000000000	0
6190	6060000000000000	192
619B	0000000000000000	0
61A0	0000000000000000	0
61AB	0000000000000000	0
61B0	0000000000000000	0
61BB	0000000000000000	0
61C0	0000000000000000	0
61CB	0000000000000000	0
61D0	0000000000000000	0
61DB	0000000000000000	0
61E0	0000000000000000	0
61EB	0000000000000000	0
61F0	0000000000000000	0
61FB	0000000000000000	0
6200	185B245B30583C58	520





```

6208 48585458FFFF6C58 1038
6210 7858845818682468 696
6218 30683C6848685468 680
6220 60686C6878688468 872
6228 1878FF0730783C78 754
6230 48785478FF0FFF0F 936
6238 7878B4781888FF0F 922
6240 3088FF0F48885488 882
6248 FF17FF1FFF0FB488 1102
6250 1898FF1FFF0FFF27 1026
6258 4898FF07FF1FFF27 1066
6260 FF27FF1F00000000 580
6268 0000000000000000 0
6270 55FF00AA00AAAA55 935
6278 AAAAAAAFF0AAAA 1275
6280 00AA55FF00000000 510
6288 00AA0055AA0000AA 595
6290 0000AA0000AA0000 340
6298 AA0055FF00000000 510
62A0 55FF00AA00AA0000 680
62A8 AA00FF00550000AA 680
62B0 0000FFFFAA000000 680
62B8 55FF00AA00AA0000 680
62C0 AA00FF000000AAAA 765
62C8 00AA55FF00000000 510
62D0 00550000FF005555 510
62D8 00AA5500FFFFAA00 935
62E0 5500005500000000 170
62E8 FFFFAAAA0000FFFF 1360
62F0 000000AA0000AAAA 510
62F8 00AA55FF00000000 510
6300 00FF00550000AA00 510
6308 00FFFF00AA00AAAA 1020
6310 00AA55FF00000000 510
6318 FFFFAA0000AA0055 935
6320 0000AA0055000055 340
6328 0000550000000000 85
6330 55FF00AA00AAAA00 850
6338 AA55FF00AA00AAAA 1020
6340 00AA55FF00000000 510
6348 55FF00AA00AAAA00 850
6350 AA55FFAA0000AA00 850
6358 550055AA00000000 340
6360 8400011E00000F0A 188
6368 0000820202000005 139
6370 00CB0000820000EB 562
6378 03000F0500FFB100 407
6380 06CB00000F100000 237
6388 810001BC00000C14 302
6390 0002000000000000 2
6398 0000000000000000 0
63A0 0000000000000000 0
63A8 0000000000000000 0
63B0 0000000000000000 0
63B8 0000000000000000 0
63C0 0000000000000000 0
63C8 0000000000000000 0
63D0 0000000000000000 0

```

```

63D8 0000000000000000 0
63E0 0000000000000000 0
63E8 0000000000000000 0
63F0 0000000000000000 0
63F8 0000000000000000 0
6400 0000000000000000 0
6408 0000000000000000 0
6410 0000000000000000 0
6418 0000000000000000 0
6420 0000000000000000 0
6428 0000000000000000 0
6430 0000000000000000 0
6438 0000000000000000 0
6440 0000000000000000 0
6448 0000000000000000 0
6450 0000000000000000 0
6458 0000000000000000 0
6460 0000000000000000 0
6468 0000000000000000 0
6470 0000000000000000 0
6478 0000000000000000 0
6480 0000000000000000 0
6488 0000000000000000 0
6490 0000000000000000 0
6498 0000000000000000 0
64A0 0000000000000000 0
64A8 0000000000000000 0

```

## SPACE INVADERS

```

10 MEMORY &5FFF
20 LOAD "!CODES"
30 LOAD "!TABLE"
40 DELETE -40
50 "SPACE INVADER
60 " (c) KOUBDJANIAN Sylvain 1985
70 ENT 1,10,1,1:ENV 2,1,15,10,15,-1,7
80 ENT 2,1,5,1,10,-1,20,10,1,1,5,-1,1
90 ENT 8,10,-10,1,1,0,10:ENT 6,1,0,8,B,-20,1
100 "ENTREE DES OPTIONS
110 GOSUB 600
120 PEN 1:PRINT SPC(8);"SPACE INVADER":PEN 2
:LDCATE 1,6:PRINT"Selectionnez votre niveau
de jeu":PEN 3
130 LOCATE 1,10:PRINT"Difficulte du jeu ( 0-
9 ) :";GOSUB 630
140 "Pokes correspondants
150 "Tir des monstres
160 RESTORE 180:FOR B=0 TO 9:READ C,D:IF C=A
THEN 170 ELSE NEXT B
170 POKE &75C1,D
180 DATA 0,8,1,6,2,4,3,4,4,3,5,2,6,2,7,1,8,1
,9,1
190 P=16+(A\2)*8
200 "Avance des monstres:table &6150
210 RESTORE 220:C=13-A:FOR B=&6159 TO &6150

```

```

STEP -2:READ D:POKE B,C:C=C&2:POKE B-1,D:NEX
T B
220 DATA 51,45,36,26,14
230 "Vitesse mystery
240 POKE &75E2,(19-A)\2)*8:IF A>7 THEN POKE
&75E2,4
250 PRINT:PRINT"Vitesse du jeu ( 0-9 ) :";:6
OSUB 630:A=9-A
260 "Vitesse du jeu:pokes
270 POKE &7617,A+1:IF A=0 THEN POKE &7619,1
ELSE POKE &7619,255
280 "Vitesse de la base
290 PRINT:PRINT"Vitesse de la base ( 0-9 ) :
";:6OSUB 630:A=10-A:POKE &75D3,A
300 "Presentation des monstres
310 GOSUB 610:PEN 13:PRINT" ENVAHISSEURS"
320 CALL 3000:RESTORE 330:FOR A=6 TO 18 STE
P 4:READ B:LOCATE 6,A:PEN 12:PRINT B;"PTS.":
NEXT A
330 DATA 30,20,10,200
340 LOCATE 1,23:PRINT"< TOUCHE >":CALL &B006
350 "Jeu commence
360 POKE &15E,P:in=TIME:GOSUB 610:PEN 13:LOC
ATE 10,1:PRINT"SCORE:";:CALL 30004
370 A=PEEK(&15F)+1:ON A GOTO 380,420,450,530
380 "Efface le bas de l'ecran
390 FOR b=2 TO 25:LOCATE 1,b:PRINT "
";:NEXT b
400 IF PEEK(&15E)=104 THEN 410 ELSE POKE &15
E,PEEK(&15E)+8
410 CALL 30029:GOTO 370
420 "Base envahie
430 PEN 14:LOCATE 1,5:PRINT"Vous etes envahi
!!!";
440 FOR a=1 TO 2000:NEXT a:GOTO 470
450 "Base detruite
460 INK 0,13:PEN 14:LOCATE 1,5:PRINT"Vous et
es mort !!!":GOTO 440
470 "Affichage des resultats
480 GOSUB 600:A=PEEK(&131)*10000+PEEK(&132)*1
000+PEEK(&133)*100+PEEK(&134)*10
490 PEN 1:PRINT"Vous avez tenu pendant";:PEN
2:PRINT INT((TIME-in)/300);:PEN 1:PRINT"sec
ondes."
500 LOCATE 1,10:PEN 3:PRINT"VOTRE SCORE EST
DE ";:PEN 3:PRINT A
510 PEN 1:LOCATE 1,22:PRINT"< Presser un '.
' >"
520 IF INKEY#(">")." THEN 520
530 "Menu final
540 GOSUB 600:PEN 1:PRINT"Desirez-vous : "
550 PEN 2:LOCATE 1,8:PRINT"- 1) Refaire une
partie au meme niveau.":PRINT:PRINT"- 2) Ref
aire une partie en changeant le nive
au de jeu."
560 PRINT:PRINT"- 3) Arrêter."
570 a%=INKEY%:IF a#("<" OR a#(">" OR a#("&0" OR a#("&9" THEN 570
580 ON VAL(A%) GOTO 360,50,590
590 MODE 1:INK 0,13:INK 1,0:PEN 1:BORDER 13:
END
600 MODE 1:INK 0,13:BORDER 13:INK 1,0:INK 2,
2:INK 3,26:RETURN
610 MODE 0:INK 0,0:BORDER 0:INK 1,6:INK 2,8:
INK 3,20:INK 4,26:INK 6,24:INK 7,0:INK 8,18
620 INK 9,26:INK 10,14:INK 11,23:INK 12,19:INK
13,11:INK 15,17:INK 14,2,21:RETURN
630 A%=INKEY%:IF A#("<" OR A#("&0" OR A#("&9" THEN 630
640 A=VAL(A%):PRINT A:RETURN

```

## PROGRAMME DE LISTAGE DES CODES MACHINES

```

1 'save "llm
10 'Liste des codes-----
20 INPUT "adresse de depart (hex)";a$:a=VAL(
"%"+a$)
30 INPUT "Longueur      ";b
35 b=a+b:PRINT "adresse de fin  ";HEX$(b)
37 PRINT
40 PRINT RIGHT$("000"+HEX$(a),4);" ";
45 s=0
50 FOR i=0 TO 7
60 x=PEEK(a):s=s+x
70 a=a+1
80 PRINT RIGHT$("0"+HEX$(x),2);
90 NEXT
100 PRINT USING " ####";s
110 IF a<b THEN 40

```



## ORIC Atmos

# UTIL-1

Langage : Basic



**PHILIPPE NAVEZ  
ET EMMANUEL ROSSI  
GAGNENT  
UN MONITEUR  
EUREKA MC14**

### Emmanuel Rossi

Prépare sa deuxième année de DEUST Informatique et gestion. Deux passions dans la vie : l'informatique et la lecture. Emmanuel programme sur tout objet possédant un micro-processeur et un clavier : FX 602P, Micral 9050, Oric, Amstrad PCW 8256, HP 250, etc.

### Philippe Navez

Vingt et un ans. Philippe a un DUT d'informatique, il est programmeur-analyste.

Util 1 vous propose trois nouvelles commandes à intégrer à votre cher Oric : !GET, !DRAW et !SORT. !GET compile un écran en mode HIRES ou TEXT à partir d'une adresse mémoire et permet un gain de 40 % en taille. !DRAW affiche un écran préalablement mémorisé par l'ordre !GET. !SORT trie un tableau de variables, que ces variables soient réelles,

entières ou une chaîne de caractères. Le mode précis d'utilisation de ces nouveaux ordres est décrit à l'écran et un programme de démonstration vous est gracieusement offert pour que vous puissiez vous faire une idée de leurs possibilités. ■

### Philippe NAVEZ et Emmanuel ROSSI

```

100 REM
101 REM
102 REM=====      UTIL 1      =====
==
103 REM
104 REM      !GET, !DRAW, !SORT
105 REM
106 REM      PAR Philippe NAVEZ & Emmanuel ROS
SI
107 REM
108 REM=====
==
190 REM
191 REM      IMPLANTATION PROGRAMME MACHINE
192 REM
195 CLS:PAPER0:INK3:DOKE#A6,#9400
196 PRINT:PRINT"PATIENCE IMPLANTATION DU LM

```

```

.. "
197 DIMA$(14):FORI=0TO14:READA$(I):NEXT
200 OI=(PEEK(##FFFE)=40)
202 IF OI THENZ1=#E6CA:Z2=#E04 ELSE Z1=#E7
6A:Z2=#E93D
204 I=#9450:CALLZ1
205 S=0
210 READA$:IFA$="FIN"THEN250
220 IFLEFT$(A$,1)<>"#"THENA=VAL("#"+A$):POK
EI,A:S=S+A:I=I+1:GOTO210
230 IFS<>VAL(A$)THENPRINT"ERREUR DATAS SOMM
E=",A$:CALLZ2:END
240 GOTO205
250 CALLZ2
260 REM
261 REM      MODIFICATIONS SI ATMOS
262 REM
270 IF OI THEN 295
280 READA,B:IFA=0THEN295
285 DOKEA,B:GOTO280
290 REM
291 REM      DESCRIPTION DES FONCTIONS
292 REM
295 V$=CHR$(130):POKE618,2
300 CLS:PRINTCHR$(129)"      Description des
fonctions"
310 PRINT:PRINTV$"!GET Ad I,Var]"

```

```

315 PRINT"Compile un ecran en mode HIRES ou
TEXT";
320 PRINT"a partir de l'adresse memoire Ad.
"
325 PRINT"Si une variable est indiquee,l'ad
resse";
330 PRINT"de fin de compilation y sera stoc
kee."
340 PRINT"On note en general un gain de 40%
en":PRINT"taille"
350 PRINT"ATTENTION:La variable eventuelle
doit"
360 PRINT"etre reelle,d'autre part il faut
faire";
370 PRINT"attention a ne pas compiler une i
mage"
380 PRINT"sur cet utilitaire."
390 PRINT:PRINTV$!"DRAW Ad"
400 PRINT"Affiche un ecran prealablement me
mori-";
410 PRINT"se par l'ordre !GET."
450 PRINT:PRINTV$!"SORT Var(0)"
460 PRINT"Trie le tableau de variables a un
e"
470 PRINT"dimension Var. Le '(0)' est obli-
"
480 PRINT"gatoire. Le tri peut se faire sur
des"
490 PRINT"variables reelles, entieres ou ch
aines";
500 PRINT"de caracteres."
510 PRINT"REMARQUE : Trie aussi les variabl
es"
520 PRINT"non dimensionnees."
600 GETA#:GOTO3000
790 REM
791 REM DATAS DEMO TRI CHAINES
792 REM
800 DATAORDINATEUR,"MICRO VO",CASSETTE,"VIV
E L'INFORMATIQUE","ORIC ATMOS",ORIC
810 DATAAMSTRAD,PROGRAMME,SINCLAIR,BASIC,AT
ARI,LANGUAGE,DISQUETTE,COMMODORE
820 DATAFIN
990 REM
991 REM DATAS PROGRAMME EN LM
992 REM
1000 DATAA9,5F,A0,94,8D,F5,2,8C,F6,2,#544
1010 DATAEA,EA,EA,EA,60,A0,0,84,18,20,#564
1020 DATAEB,0,C9,BE,80,3,4C,FA,95,C9,#5E6
1030 DATAAC,D0,3,4C,15,97,A9,53,20,DB,#46E
1040 DATACF,A9,D2,20,DB,CF,A9,54,20,DB,#60C
1050 DATACF,4C,97,94,20,FC,D0,48,AS,B4,#5D3
1060 DATA30,6,AS,B5,30,2,68,60,4C,B6,#35C
1070 DATACE,20,FC,D0,B5,4A,84,4B,38,AS,#535
1080 DATAB6,E9,3,85,5C,AS,87,E9,0,B5,#54D
1090 DATA5D,A0,0,B1,5C,C9,1,D0,76,7B,#492
1100 DATACB,B1,5C,85,55,CB,B1,5C,85,54,#55D
1110 DATAA5,B4,30,25,AS,B5,30,A0,A9,FF,#520
1120 DATA85,61,A9,5,85,62,A9,CB,8D,BE,#537
1130 DATA95,8D,BF,95,8D,C0,95,A9,8B,8D,#616
1140 DATAC9,95,8D,CA,95,8D,CB,95,4C,2E,#5B1
1150 DATA95,A9,0,85,61,A9,2,85,62,A9,#45F
1160 DATAEA,8D,BE,95,8D,BF,95,8D,C0,95,#68D
1170 DATABD,C9,95,8D,CA,95,8D,CB,95,4C,#610
1180 DATA2E,95,A9,1,85,61,A9,3,85,62,#3E6
1190 DATAA9,CB,8D,BE,95,A9,8B,8D,C9,95,#66D
1200 DATAA9,EA,8D,BF,95,8D,C0,95,8D,CA,#6AD
1210 DATA95,8D,CB,95,4C,2E,95,4C,A0,D2,#54F
1220 DATA5B,60,AS,4A,85,5E,AS,4B,85,5F,#45E
1230 DATA38,AS,54,E9,1,85,56,AS,55,E9,#4D9
1240 DATA0,30,E9,85,57,D0,6,AS,56,C9,#4BF
1250 DATA2,90,DF,A9,0,85,60,AS,61,30,#435
1260 DATA20,F0,3B,A0,5,B1,5E,99,58,0,#3F0
1270 DATAB8,10,FB,CB,C4,58,F0,6C,C4,5B,#5EF
1280 DATAF0,4B,B1,59,D1,5C,90,62,F0,EF,#640
1290 DATA4C,B2,95,AS,5E,A4,5F,20,73,DE,#50A
1300 DATA1B,AS,5E,65,62,4B,AS,5F,69,0,#397
1310 DATAAB,6B,20,34,DF,30,45,F0,43,4C,#437
1320 DATAB2,95,1B,AS,5E,65,62,85,59,AS,#4AC
1330 DATA5F,69,0,85,5A,A0,0,B1,5E,D1,#427
1340 DATA59,90,2B,D0,9,CB,B1,5E,D1,59,#4AE
1350 DATA90,22,F0,20,A9,1,85,60,AA,62,#457
1360 DATAB8,B1,5E,4B,CB,CB,CB,CB,B1,#678
1370 DATA5E,AA,6B,91,5E,8B,8B,8B,8B,#507
1380 DATA8A,91,5E,8B,10,E7,1B,AS,5E,65,#478
1390 DATA62,85,5E,90,2,E6,5F,AS,56,D0,#4E7
1400 DATA2,C6,57,C6,56,F0,3,4C,51,95,#460
1410 DATAA5,57,F0,3,4C,51,95,AS,60,F0,#316
1420 DATA3,4C,2E,95,58,60,20,E2,0,20,#2EC
1430 DATA9D,E7,20,33,97,91,33,E6,33,90,#4DB
1440 DATA2,E6,34,AS,33,69,2,85,1B,AS,#3A1
1450 DATA34,69,0,85,19,4C,23,96,E6,1D,#343
1460 DATADO,2,E6,1E,20,8D,96,A0,0,B1,#46A
1470 DATA1D,85,1F,20,5E,96,D0,EC,20,6A,#41B
1480 DATA96,A9,5,1B,65,1D,85,1D,90,2,#312
1490 DATAE6,1E,A0,0,84,26,B1,1D,C5,1F,#400
1500 DATADO,7,E6,26,CB,C0,FF,D0,F3,20,#64D
1510 DATA7B,96,1B,AS,1D,65,26,85,1D,90,#3A8
1520 DATA2,E6,1E,4C,20,96,A0,4,B1,1D,#37A
1530 DATAC5,1F,D0,3,8B,D0,F7,60,A0,0,#506
1540 DATAA5,1D,3B,E5,F6,91,1B,AS,1E,E5,#526
1550 DATAF7,CB,91,1B,60,A0,2,AS,26,91,#4C6
1560 DATA1B,1B,AS,1B,69,3,85,1B,90,2,#2BB
1570 DATAE6,19,60,AS,1E,C9,C0,D0,F9,6B,#5DC
1580 DATAB8,A0,0,AS,1B,91,33,CB,AS,19,#40F
1590 DATA91,33,AS,F6,85,1D,AS,F7,85,1E,#540
1600 DATA4C,C0,96,E6,1D,D0,2,E6,1E,20,#49B
1610 DATAF5,96,AS,1F,A0,0,91,1B,E6,1B,#496
1620 DATADO,2,E6,19,A0,0,B1,1D,85,1F,#3E3
1630 DATA20,5E,96,D0,E0,A9,5,1B,65,1D,#40C
1640 DATA85,1D,90,2,E6,1E,A0,0,84,26,#382
1650 DATAB1,1D,C5,1F,D0,7,E6,26,CB,C0,#51D
1660 DATAFF,D0,F3,1B,AS,1D,65,26,85,1D,#4C9
1670 DATA90,2,E6,1E,4C,84,96,AS,1E,C9,#4BB
1680 DATACO,D0,37,68,6B,20,E8,0,C9,2C,#494
1690 DATADO,2E,A4,1B,AS,19,20,D5,DB,20,#465
1700 DATAE2,0,20,86,94,AA,4C,AS,DE,20,#485
1710 DATAE2,0,20,9D,E7,20,33,97,D1,33,#474
1720 DATAF0,3,4C,72,DA,1B,E6,33,90,2,#44E
1730 DATAE6,34,8,7B,20,53,97,2B,60,A0,#3CC
1740 DATA0,84,1D,84,F6,A9,A0,B5,1E,B5,#4BC
1750 DATAF7,AD,C0,2,29,1,D0,C,A2,80,#48E
1760 DATAB6,1D,8B,F6,A2,BB,8B,F7,86,1E,#59D
1770 DATA60,B1,33,85,2B,CB,B1,33,85,2C,#451
1780 DATA1B,AS,33,69,2,85,1B,AS,34,69,#33A
1790 DATA0,85,19,A0,0,B1,1B,1B,65,F6,#37A
1800 DATAB5,F3,CB,B1,1B,65,F7,85,F4,CB,#6A6
1810 DATAB1,1B,85,26,A9,3,65,1B,85,1B,#33A
1820 DATA90,2,E6,19,AS,1D,C5,F3,D0,46,#521
1830 DATAA5,1E,C5,F4,D0,40,A0,0,B1,2B,#508
1840 DATAB5,1F,91,1D,C0,5,F0,4,CB,4C,#41F
1850 DATA9A,97,1B,A9,5,65,1D,85,1D,90,#3AB
1860 DATA2,E6,1E,A0,0,AS,1F,CA,26,F0,#444
1870 DATA6,91,1D,CB,4C,B3,97,1B,AS,26,#3F5
1880 DATA65,1D,85,1D,90,2,E6,1E,E6,2B,#3CB
1890 DATADO,2,E6,2C,20,EC,97,4C,69,97,#4D3
1900 DATAA0,0,B1,2B,91,1D,E6,2B,D0,2,#40D
1910 DATAE6,2C,E6,1D,D0,2,E6,1E,20,EC,#4F7
1920 DATA97,4C,8B,97,AS,1E,C9,C0,D0,2,#520
1930 DATA6B,6B,60,#130
1940 DATAFIN
1990 REM
1991 REM DATAS ATMOS
1992 REM
2000 DATA#9477,#D067
2002 DATA#947C,#D067
2004 DATA#9481,#D067
2006 DATA#9487,#D18B
2008 DATA#9495,#CF12
2010 DATA#9498,#D18B
2012 DATA#952A,#D336
2014 DATA#957A,#DE7B
2016 DATA#9589,#DF4C
2018 DATA#95FE,#E853
2019 DATA#9709,#DF40
2020 DATA#9713,#DEAD
2022 DATA#9719,#E853
2024 DATA#9723,#DAFD
2026 DATA0,0
2990 REM
2991 REM DEMONSTRATION GRAPHIQUE
2992 REM
3000 CALL#9450
3100 HIRES:POKE61B,2
3110 PRINTV$!"DEMONSTRATION DE L'INSTRUCTION
!GET":WAIT200
3115 CLS:PRINT"Faisons d'abord 2 jolis dess
ins"
3120 FORI=0TO199STEP2:CURSET0,1,3:DRAW239,1
99-!#2,2:NEXT
3130 CURSET120,30,0:FORI=1TO30STEP2:CIRCLEI
,1:NEXT
3140 CURSET120,169,0:FORI=1TO30STEP2:CIRCLE
I,1:NEXT
3150 CLS:PRINT"Memorisation du 1er dessin"
3160 D=#3000:GOSUB3500
3200 GETR#
3210 HIRES:L=160:X1=200:X2=40
3213 POKE61B,2
3215 PRINT"Arretez le dessin par une touche
"
3230 FORI=31TO10STEP-2:INKI/6+1
3240 FORC=0TOLSTEP1:CURSETX1-C,10,3
3245 IFKEY<>"#THEN3300
3250 DRAWC,L-C,1:CURSETX2,C+10,3
3260 DRAWC,L-C,1:CURSETC+X2,X1-30,3
3270 DRAWL-C,-C,1:CURSETX2,C+10,3
3280 DRAWL-C,-C,1:NEXT:NEXT
3300 CLS:PRINT"Memorisation du 2eme dessin"
3310 D=#5000:GOSUB3500
3320 GETR#HIRES:POKE61B,2
3330 CLS:PRINTV$!"DEMONSTRATION DE L'INSTRUC
TION !DRAW":PRINT
3340 PRINT"TAPEZ UNE TOUCHE POUR ARRETER";:
WAIT250
3350 !DRAW#3000:!DRAW#5000
3360 IFKEY#=""THEN3350
3400 CLS:PRINT"Tapez une touche":GETR#GOT
D4000
3500 !GET D,A:CLS:F=A-D
3510 PRINT"PLACE OCCUPEE : ";P;"OCTETS EN ME
NOIRE"
3520 PRINT"AU LIEU DE 8159 OCTETS"
3530 PRINT"PLACE GAGNEE : ";8159-P;"OCTETS"

```

```

3540 RETURN
3990 REM
3991 REM      DEMONSTRATION DU TRI
3992 REM
4000 TEXT:CLS:POKE61B,2:PRINTV$* DEMONSTRATI
TION DU TRI DE TABLEAUX*
4005 M$=CHR$(133):POKE#BBA4,0
4010 PRINTCHR$(129)* FONCTION !SORT*
4020 N=14:DIMAX(N),A(N)
4025 NT$=V$+"Tableau non trie ":"TT$=V$+"Ta
bleau trie ":"
4030 PRINT:PRINT:PRINTM$*TRI D'ENTIER":PRI

```

```

NT:PRINT$:PRINT
4040 FORI=0TON:AZ(I)=INT(RND(1)*2000)+1:PRI
NTAZ(I),:NEXT
4050 !SORT AZ(0)
4060 PRINT:PRINT:PRINTTT$:PRINT
4070 FORI=0TON:PRINTAZ(I),:NEXT
4080 PRINT:PRINT:PRINT:PRINT* APPUYEZ SUR
UNE TOUCHE*:GETRR$
4100 CLS:PRINT:PRINTM$*TRI DE REELS*:PRINT:
PRINTNT$:PRINT
4110 FORI=0TON:A(I)=RND(1)*1000+1:PRINTA(I)
,:NEXT
4120 !SORT A(0)

```

```

4130 PRINT:PRINT:PRINTTT$:PRINT
4140 FORI=0TON:PRINTA(I),:NEXT
4200 PRINT:PRINT:PRINT:PRINT* APPUYEZ SUR
UNE TOUCHE*:GETRR$
4210 CLS:PRINT:PRINTM$*TRI DE CHAINES*:PRIN
T:PRINTNT$:PRINT
4215 FORI=0TON:PRINTA$(I),:NEXT
4220 !SORT A$(0)
4230 PRINT:PRINT:PRINTTT$:PRINT
4240 FORI=0TON:PRINTA$(I),:NEXT
4250 PRINT:POKE61B,3

```

## ZX Spectrum

# SOFT-AIDE

Langage : Basic



**CHRISTOPHER POUSE  
GAGNE  
UN MONITEUR  
EUREKA MC14**

### Christopher Pouze

Vingt ans. Christopher, après deux ans d'apprentissage de la photo, travaille actuellement en free lance. Programme depuis plus d'un an sur Spectrum. Ses autres loisirs : le cinéma et la musique.

**M**ini Toolkit essentiellement axé sur la présentation visuelle et audio des programmes en Basic, Soft-Aide vous permettra d'améliorer vos productions, surtout si vous avez un tempérament plus technicien qu'artiste. Quatorze options différentes vous sont proposées. Commencez par l'option 13, un échantillonnage des différents effets qui peuvent être obtenus : mots en 3D, impression lente, effacements de l'écran, bruitages, scrolls... Le mode d'emploi est indiqué au fur et à mesure. Au début, lancez le programme par un RUN, mais attention si par la suite vous faites un BREAK, relancez toujours le programme par un RUN 15, sinon vous risquez de réinitialiser le Spectrum et de perdre le programme. ■

**Christopher POUSE**

```

1 REM *****
7 REM *
B REM *****
9 REM
      CHARGEMENT DES CARACTERES
10 BORDER 0: PAPER 0: INK 4: B

```

RIGHT 1: 60 TO 9000

```

11 REM *****
12 REM * DEBUT DU PROGRAMME *
13 REM *****

```

14 REM

```

*****
IMPORTANT,SI VOUS FAITES
*BREAK* RELANCER TOUJOURS
LE PROGRAMME PAR *RUN 15*
*****

```

```

15 INK 7: CLS : POKE 23606,0:
POKE 23607,249: LET PINK=50136:
LET DUR=50138: LET SON=50132: LE
T GAUCHE=50000: LET DROITE=50051
: LET HAUT=50090: LET BAS=50111
20 PDKE 23609,60: POKE 23658,8
25 LET P=10: 60 SUB 8000: BORD
ER 1

```

```

27 PAPER 1: CLS : LET I=6: LET
A$="CHRISTOPHER POVSE PRESENTE"
: LET X=2: LET Y=3: 60 SUB 7000
30 LET P=1: LET I=7: LET B=1:
LET V1=3.5: LET V2=4: LET V3=140
: LET E1=0: LET E2=70: LET A$="S
OFT-AIDE": 60 SUB 154: PRINT AT
21,0;" "

```

```

31 INK 2: PLOT 12B,30: PLOT 13
6,20: DRAW 7,10,20000: INK 7
32 PRINT #0;AT 0,2; INK 6;"LOG
ICIEL ECRIT POUR MICRO VD"

```

33 RESTORE 35: FOR J=1 TO 32:  
READ K

```

34 BEEP .07,K-28: BEEP .04,K-2
8: BEEP .07,K-40: BEEP .04,K-40
35 DATA 29,41,53,65,29,65,53,4
1,31,43,55,67,31,67,55,43,29,41,
53,65,29,65,53,41,70,58,46,34,70
,34,34,46,70

```

```

36 NEXT J
37 FOR N=1 TO 96: 60 SUB 6450:
BEEP .02,N/2: NEXT N: 60 SUB 64
00: LET C1=0: LET C2=16: 60 SUB
6200: 60 SUB 7600: 60 SUB 6300:
60 TO 40

```

```

40 INK 4: CLS : LET P=71: 60 S
UB 8000

```

```

45 BORDER 0: BRIGHT 1: PAPER 0
: CLS : INK 4: PRINT PAPER 2; IN
K 7;"CHRISTOPHER POVSE      SDF
T-AIDE": LET Z=3

```

```

50 PRINT INK 7;"1.MOTS EN 3D"
"2.PRINT LENT""3.LISTE DE POKE
S""4.BEEP CLAVIER""5.EFFACEMENT
TS ECRAN""6.BRUITAGES"

```

```

51 PRINT INK 7;"7.BRUITAGE 2"
"8.SCROLLS""9.MULTI PAPER""10.
ECRAN BICOLORE""11.BINAIRE-DECI
MAL""12.DECIMAL-BINAIRE""13.DE
MONSTRATION""14.SAUVEGARDE DE S
OFT-AIDE"

```

```

52 PRINT AT 19,9; PAPER 1; INK
2;"S"; INK 7;"SELECTIONNE";

```

```

AT 20,9; INK 2;""ENTER""; INK
7;"EXECUTE": PRINT AT 2,0; OVER
1; INK 2;"
"
54 PAUSE 0: LET M$=INKEY$
56 IF M$("<"S" AND M$("<"CHR# 13
THEN GO TO 54
58 IF M$=CHR# 13 THEN GO SUB 6
150: CLS : GO TO (Z#100)-200
60 PRINT AT 2,0; OVER 1; INK 7
;"
": LET Z=Z+1; IF Z=17 THEN LET
Z=3
61 GO TO 52
99 GO TO 99
100 REM #####
110 REM # MOTS EN 3D #
120 REM #####
125 LET P=10: GO SUB 8000: LET
P=71: GO SUB 8000
130 PRINT PAPER 2; INK 7;"SOFT-
AIDE MOTS EN 3D"
135 PRINT ""CETTE ROUTINE PERM
ET D'ECRIRE ""DES MOTS EN 3D,
TROIS LARGEURS""DE LETTRE SONT
POSSIBLES:"; INK 3;""-1=PETITE(
9 LETTRES MAXI)""-2=MOYENNE(7 L
ETTRES MAXI)""-3=LARGE(6 LETTRE
S MAXI)"
136 INPUT "PAPER(0-7)?";P: IF P
<0 OR P>7 THEN GO TO 136
137 PAPER P: CLS
141 INPUT "INK(0-7)?";I: IF I<0
OR I>7 THEN GO TO 141
142 INPUT "BRIGHT(0-1)?";B: IF
B<0 OR B>1 THEN GO TO 142
143 INPUT "LARGEUR LETTRE(1,2,3
)?";L: IF L<1 OR L>3 THEN GO TO
143
144 LET E1=0
145 INPUT "ESPACEMENT DU HAUT(
=0)?";E2: IF E2<0 OR E2>130 THEN
GO TO 145
147 INPUT "CARACTERES MAGNETIC(
0-N)?";R$: IF R$="N" THEN POKE 23
606,0: POKE 23607,60
148 INPUT "TEXTE(6,7,9)?";A$: I
F L=1 AND LEN A$>9 THEN GO TO 14
8
149 IF L=2 AND LEN A$>7 THEN GO
TO 148
150 IF L=3 AND LEN A$>6 THEN GO
TO 148
151 IF L=1 THEN LET V1=3.5: LET
V2=4: LET V3=140
152 IF L=2 THEN LET V1=4: LET V
2=4: LET V3=135
153 IF L=3 THEN LET V1=5: LET V
2=5: LET V3=140
154 BRIGHT B: INK I: LET A=LEN
A$: PRINT BRIGHT I; INK P;AT 21,

```

```

0;A$: FOR F=5 TO 10: FOR G=6 TO
2 STEP -1: POKE PINK,G: POKE DUR
,F: RANDOMIZE USR SON: NEXT G: N
EXT F
155 POKE 23606,0: POKE 23607,24
9
160 FOR F=0 TO 8#A-1: FOR N=0 T
O 7: IF POINT (F,N)=0 THEN GO TO
179
170 PLOT F#V1+E1,N#V2+V3-E2: DR
AW 4,0: DRAW 0,4: DRAW -4,0: DR
AW 0,-3: DRAW 3,0: DRAW 0,2: DR
AW -2,0: DRAW 0,-1: DRAW 2,0: DR
AW -2,-2
175 DRAW 5,5: DRAW 0,4: DRAW 0,
-4: DRAW 4,0: DRAW 0,4: DRAW 0,-
4: DRAW -5,-5
177 DRAW 0,4: DRAW 5,5: DRAW -4
,0: DRAW -5,-5
179 NEXT N: NEXT F
180 IF A$="SOFT-AIDE" OR A$="DE
MO" THEN RETURN
185 INPUT "ECRIRE ENCORE(0-N)?"
;R$: IF R$="N" THEN GO TO 190
187 IF R$="0" THEN INPUT "EFFAC
ER L'ECRAN(0-N)?";R$: IF R$="0"
THEN CLS : GO TO 100
189 IF R$="N" THEN GO TO 140
190 INPUT "SAUVEGARDE SCREEN$(0
-N)?";R$: IF R$="N" THEN GO TO 1
94
192 INPUT "NDM DE L'ECRAN";S$:
IF LEN S$>10 THEN GO TO 192
193 PRINT #0;AT 0,7; FLASH 1;"P
RESSER UNE TOUCHE": POKE 23736,1
81: SAVE S$SCREEN$
194 INPUT "COPIE IMPRIMANTE(0-N
)?";R$: IF R$="0" THEN COPY
196 CLS : LET P=55: GO SUB 8000
: LET P=71: GO SUB 7800: GO TO 4
5
200 REM
201 REM PRINT LENT
202 REM
205 LET P=20: GO SUB 7800: LET
P=71: GO SUB 7800
210 PRINT PAPER 2; INK 7;"SOFT-
AIDE PRINT LENT"
215 LET I=4: LET X=3: LET Y=0:
LET A$="CETTE ROUTINE PERMET D'A
FFICHER": GO SUB 7000
220 LET A$="DES MESSAGES LETTRE
PAR LETTRE.": LET I=4: LET X=5:
LET Y=0: GO SUB 7000
230 INK 2: PLOT 10,100: DRAW 23
5,0: DRAW 0,-43: DRAW -235,0: DR
AW 0,43: INK 4
235 PRINT INK 7;AT 10,2;"10 LET
A$="VOTRE MESSAGE";AT 11,2;"20
FOR K=1 TO LEN(A$);AT 12,2;"30
PRINT A$(K TO K);;PAUSE 5";AT 1

```

```

3,2;"40 NEXT K"
250 GO TO 5000
300 REM
301 REM POKE EN VRACS
302 REM
304 LET P=60: GO SUB 8000: LET
P=71: GO SUB 8100
305 PRINT INK 7; PAPER 2;"SOFT-
AIDE LISTE DE POKES"
310 PRINT INK 7;""-POKE 2365
8,8 AUTO CAPS LOCK""-POKE 2373
6,181 SUPRIME LES ""start ta
pe"";A PLACER AVANT LES ""SAVE""
""-POKE 23562,2:POKE 23561,20
AUGMENTE LA SENSIBILITE DU
CLAVIER"
320 PRINT INK 7;""-PRINT 65536-
USR 7962 DONNE LA MEMOIRE DISPO
NIBLE DU SPECTRUM"
350 GO TO 5000
400 REM
401 REM BEEP CLAVIER
402 REM
404 LET P=40: GO SUB 8000: LET
P=71: GO SUB 7800
405 PRINT INK 7; PAPER 2;"SOFT-
AIDE BEEP CLAVIER"
410 PRINT ""CE POKE PERMET DE
CHANGER LE ""CLICK PRODUIT PAR
LES TOUCHES""DU SPECTRUM."
420 PRINT AT 11,0; INK 7; PAPER
1;" POKE 23609,X
"" INK 6;"X DETERMINE LA LON
GUEUR DU CLICK""X VARIANT DE 0
A 255
"
450 GO TO 5000
500 REM
501 REM EFFACEMENT ECRAN
502 REM
504 LET P=30: GO SUB 8000: LET
P=0: GO SUB 7800
505 PRINT PAPER 2; INK 7;"SOFT-
AIDE EFFACEMENTS ECRAN"
510 PRINT ""CETTE ROUTINE PERM
ET DE CHANGER""D'ECRAN DE FACON
ORIGINALE"
515 PRINT "POUR L'UTILISER DANS
VOS SOFTS""IL FAUT D'ABORD LA
SAUVEGARDER""SOUS FORME DE BYTE
S:"
520 PRINT AT 10,4; INK 7;"FAITE
S BREAK PUIS:";AT 11,4; INK 2;"S
AVE"BYTES"CODE 50000,146.";AT 12
,4; INK 7;"POUR LA CHARGER FAIRE
:";AT 13,4; INK 2;"CLEAR 49999:L
OAD""CODE"
525 INK 5: PLOT 24,104: DRAW 21
9,0: DRAW 0,-48: DRAW -219,0: DR
AW 0,48: INK 4
530 PRINT ""POUR UTILISER CETT
E ROUTINE""REGARDER LES SOUS PR

```

```

06GRAMMES""D'EFFACEMENTS CONTENU
S DANS CE""LOGICIEL SITUES AUX
LIGNES 7800""8000,8100,...ETC"
540 GO TO 5000
600 REM
601 REM BRUITAGES
602 REM
604 LET P=30: GO SUB 8100: LET
P=71: GO SUB 8000
610 PRINT INK 7; PAPER 2;"SOFT
AIDE          BRUITAGES"
620 PRINT ""CETTE ROUTINE PERM
ET D'OBTENIR""DES EFFETS SONORE
S ORIGINAUX"
630 PRINT "POUR L'UTILISER DANS
VOS SOFTS""IL FAUT D'ABORD LA
SAUVEGARDER""SOUS FORME DE BYTE
S:"
640 PRINT AT 10,4; INK 7;"FAITE
S BREAK PUIS:";AT 11,4; INK 2;"S
AVE~BYTES~CODE 50000,146.";AT 12
,4; INK 7;"POUR LA CHARGER FAIRE
:";AT 13,4; INK 2;"CLEAR 49999:L
OAD~CODE"
650 INK 5: PLOT 24,104: DRAW 21
9,0: DRAW 0,-48: DRAW -219,0: DR
AW 0,48: INK 4
660 PRINT ""POUR UTILISER CETT
E ROUTINE""REGARDER LES SOUS PR
OGRAMMES""DE BRUITAGES CONTENUS
DANS CE""LOGICIEL SITUES AUX L
IGNES 6000""6100,7600,...ETC"
670 GO TO 5000
700 REM
702 REM BRUITAGE 2
704 REM
705 LET P=10: GO SUB 8100: LET
P=71: GO SUB 7800
710 PRINT INK 7; PAPER 2;"SOFT-
AIDE          BRUITAGE 2"
720 PRINT ""CETTE ROUTINE COMPL
ETE LA          ROUTINE DE BRUITAGE
1,ELLE NE          CONTIENT QU'UN SEUL
SON."
730 PRINT "POUR L'UTILISER DANS
VOS SOFTS""IL FAUT D'ABORD LA
SAUVEGARDER""SOUS FORME DE BYTE
S:"
740 PRINT AT 10,4; INK 7;"FAITE
S BREAK PUIS:";AT 11,4; INK 2;"S
AVE~BYTES~CODE 50300,29";AT 12,4
; INK 7;"POUR LA CHARGER FAIRE:"
;AT 13,4; INK 2;"CLEAR 50299:LOA
D~CODE"
750 INK 5: PLOT 24,104: DRAW 21
9,0: DRAW 0,-48: DRAW -219,0: DR
AW 0,48: INK 4
760 PRINT ""POUR UTILISER CETT
E ROUTINE""TAPER SIMPLEMENT:";
INK 7;"RANDOMIZE USR 50300"

```

```

770 GO TO 5000
800 REM
802 REM SCROLLS
803 REM
805 LET P=10: GO SUB 8100: LET
P=71: GO SUB 8000
810 PRINT INK 7; PAPER 2;"SOFT-
AIDE          SCROLLS"
820 PRINT ""CETTE ROUTINE PERME
T DE FAIRE          SCROLLER L'ECRAN SO
IT VERS LA          GAUCHE,SOIT VERS LA
DROITE"
830 PRINT "POUR L'UTILISER DANS
VOS SOFTS""IL FAUT D'ABORD LA
SAUVEGARDER""SOUS FORME DE BYTE
S:"
840 PRINT AT 10,4; INK 7;"FAITE
S BREAK PUIS:";AT 11,4; INK 2;"S
AVE~BYTES~CODE 50400,77";AT 12,4
; INK 7;"POUR LA CHARGER FAIRE:"
;AT 13,4; INK 2;"CLEAR 50399:LOA
D~CODE"
850 INK 5: PLOT 24,104: DRAW 21
9,0: DRAW 0,-48: DRAW -219,0: DR
AW 0,48: INK 4
860 PRINT ""UTILISATION:""; I
NK 7;"SCROLL A DROITE-RAND USR 5
0430""SCROLL A GAUCHE-RAND USR
50454"
870 GO TO 5000
901 REM MULTI-PAPER
903 REM
905 LET P=20: GO SUB 8000: LET
P=71: GO SUB 8100
910 PRINT INK 7; PAPER 2;"SOFT
AIDE          MULTI PAPER"
920 PRINT ""CETTE ROUTINE PERME
T DE CHANGER DE PAPER SANS ALTER
ER LE          CONTENU DE L'ECRAN"
930 PRINT "POUR L'UTILISER DANS
VOS SOFTS""IL FAUT D'ABORD LA
SAUVEGARDER""SOUS FORME DE BYTE
S:"
940 PRINT AT 10,4; INK 7;"FAITE
S BREAK PUIS:";AT 11,4; INK 2;"S
AVE~BYTES~CODE 50200,25";AT 12,4
; INK 7;"POUR LA CHARGER FAIRE:"
;AT 13,4; INK 2;"CLEAR 50199:LOA
D~CODE"
950 INK 5: PLOT 24,104: DRAW 21
9,0: DRAW 0,-48: DRAW -219,0: DR
AW 0,48: INK 4
960 PRINT ""EMPLDI:TAPER"; IN
K 7;"FOR N=1 TO 96"; INK 4;"(=96
CHANGEMENTS)"; INK 7;"RANDOMIZ
E USR 50200:NEXT N"; INK 4;"ON
PEUT RALENTIR L'EXECUTION EN""A
JOUTANT UN""PAUSE""AVANT NEXT N"
970 GO TO 5000
1000 REM

```

```

1002 REM ECRAN BICOLORE
1004 REM
1005 LET P=10: GO SUB 8000: LET
P=71: GO SUB 7800
1010 PRINT INK 7; PAPER 2;"SOFT-
AIDE          ECRAN BICOLORE"
1020 PRINT ""CETTE ROUTINE PERME
T D'OBTENIR UN PAPER BICOLORE S
ANS ALTERER          LE CONTENU DE L'ECR
AN"
1030 PRINT "POUR L'UTILISER DANS
VOS SOFTS""IL FAUT D'ABORD LA
SAUVEGARDER""SOUS FORME DE BYTE
S:"
1040 PRINT AT 10,4; INK 7;"FAITE
S BREAK PUIS:";AT 11,4; INK 2;"S
AVE~BYTES~CODE 50400,77";AT 12,4
; INK 7;"POUR LA CHARGER FAIRE:"
;AT 13,4; INK 2;"CLEAR 50399:LOA
D~CODE"
1050 INK 5: PLOT 24,104: DRAW 21
9,0: DRAW 0,-48: DRAW -219,0: DR
AW 0,48: INK 4
1060 PRINT ""UTILISATION:TAPER"
"; INK 7;"POKE 50401,C1:POKE 504
23,C2:""POKE 50426,C2:RAND USR
50400"; INK 4;"C1 ET C2 ETANT L
ES CODES DES""COULEURS VOULUES(
INK#8)""EX BLEU=(1#8)=8,VERT=(4
#8)=32"
1070 GO TO 5000
1100 REM
1102 REM BINAIRE-DECIMAL
1104 REM
1107 LET P=10: GO SUB 7800: LET
P=71: GO SUB 8100
1110 PRINT INK 7; PAPER 2;"SOFT-
AIDE          BINAIRE-DECIMAL"
1120 PRINT ""CETTE ROUTINE CONV
ERTI TOUT          NOMBRE BINAIRE EN N
OMBRE DECIMAL""CE QUI PERMET DE
DEFINIR""FACILEMENT DES UDS A
VEC DES""NOMBRES DECIMAUX"
1130 LET DEC=0: INPUT AT 11,0;AT
0,0; INK 2;"NOMBRE BINAIRE?""
LINE A$: LET X=LEN A$: FOR N=X T
O 1 STEP -1
1135 PRINT AT 11,0;"
    ""
    "
1140 LET DEC=DEC+VAL (A$(N))*2^(
X-N): NEXT N
1150 LET D$=STR$ DEC: LET L=LEN
D$: LET X=INT (32-X)/2: PRINT AT
11,0; INK 2;"VALEUR DECIMALE D
E ";A$:";";AT 13,X; INK 7;D$
1160 PRINT AT 17,3;"UNE AUTRE CO
NVERSION(O-N)": LET K$=INKEY$
1165 IF K$="0" THEN CLS : GO TO
1110

```

```

1170 IF K$="N" THEN PRINT AT 17,
3;"          ": 6
0 TO 5000
1180 GO TO 1160
1200 REM
1202 REM DECIMAL-BINAIRE
1204 REM
1205 LET P=10: 60 SUB 8000: LET
P=71: 60 SUB 7800
1210 PRINT INK 7; PAPER 2;"SOFT-
AIDE          DECIMAL-BINAIRE"
1220 PRINT "'CETTE ROUTINE CONV
ERTI TOUT'"NOMBRE DECIMAL EN N
OMBRE BINAIRE"
1230 INPUT AT 11,0;AT 0,0; INK 2
;"NOMBRE DECIMAL?"X: LET DEC=X
: LET A$=""
1235 IF INT (DEC/2)=(DEC/2) THEN
60 TO 1245
1237 LET A$="1"+A$
1239 LET DEC=DEC-1
1241 IF DEC=0 THEN GO TO 1251
1242 GO TO 1247
1243 GO TO 1247
1245 LET A$="0"+A$
1247 LET DEC=DEC/2
1249 GO TO 1235
1251: LET L=LEN A$: LET L=INT (3
2-L)/2: PRINT AT 11,0; INK 2;"VA
LEUR BINAIRE DE ";X;"":AT 13,L;
INK 7;A$
1260 PRINT AT 17,3;"UNE AUTRE CO
NVERSION(D-N)": LET K$=INKEY$
1265 IF K$="0" THEN CLS : 60 TO
1210
1270 IF K$="N" THEN PRINT AT 17,
3;"          ": 6
0 TO 5000
1280 GO TO 1260
1290 REM
1292 REM DEMONSTRATION
1295 REM
1300 LET P=10: 60 SUB 8100: LET
P=71: 60 SUB 7800: LET V1=3.5: L
ET V2=4: LET V3=140: LET E2=0: L
ET E1=60: LET I=1: 60 SUB 9900:
LET V1=4: LET V2=4: LET V3=135:
LET E2=50: LET E1=40: LET I=4: 6
0 SUB 9900: LET V1=5: LET V2=5:
LET E2=120: LET E1=25: LET I=2:
60 SUB 9900
1310 FOR I=1 TO 96: 60 SUB 6450:
BEEP .02,I/4: NEXT I: FOR I=1 T
O 3: 60 SUB 6400: NEXT I
1320 GO SUB 6350
1330 GO SUB 7100: 60 SUB 7600: 6
0 SUB 7200: 60 SUB 7650: 60 SUB
7300: 60 SUB 6150: 60 SUB 7400:
60 SUB 6000: LET P=71: 60 SUB 78
00: 60 SUB 7500: 60 SUB 6060: LE

```

```

T P=71: 60 SUB 8000: 60 SUB 6100
: 60 SUB 6060
1340 GO TO 45
1400 REM
1402 REM SAUVEGARDE
1403 REM
1405 CLS : LET P=10: 60 SUB 8000
: LET P=71: 60 SUB 8100
1410 CLS : PRINT AT 0,0; INK 7;
PAPER 2;"SOFT-AIDE          S
AUVEGARDE"
1415 INK 4: PRINT "'AVANT DE RE
ALISER LA SAUVEGARDE IL FAUT ENL
EVER LA LIGNE 10 ET LES LIGNES
8900 A 9895,PUIS          TAPER:"
1420 PRINT AT 9,2; INK 7;"10 CLE
AR 49999:LOAD          CODE:          L
OAD          CODE"
1425 INK 2: PLOT 14,107: DRAW 22
0,0: DRAW 0,-23: DRAW -220,0: DR
AW 0,23
1430 INK 4: PRINT AT 13,0;"AVEZ
VOUS REALISER LES MODIFI- -CATI
ONS(D/N)": FLASH 1;"?": IF INKEY
$="N" THEN CLS : LIST 10
1440 IF INKEY$("<")="0" THEN GO TO 1
430
1445 PRINT AT 13,0; INK 5;" DE
MAREZ LE MAGNETOPHONE ET PR
ESSER UNE TOUCHE": PAUSE 0: PAUS
E 0
1450 PRINT AT 13,0; INK 5;"
SAUVEGARDE EN COURS
          ": PO
KE 23736,181: SAVE "SOFT-AIDE" L
INE 0: POKE 23736,181: SAVE "SOF
T-CODE"CODE 50000,480: POKE 2373
6,181: SAVE "MAGNETICS"CODE 6400
0,770
1460 PRINT AT 13,0; INK 5;" RE
MBOBINNER LA K7 POUR LA VE
RIFICATION,PUIS PRESSER UNE TO
UCHE": PAUSE 0
1465 PRINT AT 13,0; INK 5;"
VERIFICATION EN COURS          "; PA
PER 0;"
          ": VERIFY **: VERIFY **CO
DE : VERIFY **CODE
1470 PRINT AT 13,0; INK 5;"
VERIFICATION REUSSIE          ": PA
USE 100: RETURN
4000 GO TO 4000
4900 REM
4910 REM COPY---RETOUR AU MENU
4920 REM
5000 PRINT #0;AT 1,0; INK 5;"
          M"RETOURNE AU MENU          ":
PAUSE 30: PRINT #0;AT 1,0;"
          ": PR
INT #0;AT 1,0; INK 5;"          "

```

```

C*POUR COPIER          *: PAUSE 3
0
5010 LET W$=INKEY$
5020 IF W$="M" THEN LET SA=0: CL
S : LET P=10: 60 SUB 7800: LET P
=71: 60 SUB 8000: 60 TO 45
5030 IF W$="C" THEN LET SA=0: CO
PY
5040 GO TO 5000
6000 REM
6010 REM BRUIT 1
6015 REM
6020 FOR G=1 TO 3
6025 FOR F=6 TO 2 STEP -1
6030 POKE PINK,F
6035 RANDOMIZE USR SON
6040 NEXT F
6045 NEXT G
6050 RETURN
6055 REM
6060 REM BRUIT 2
6065 REM
6070 FOR F=5 TO 10
6075 FOR G=6 TO 2 STEP -1
6080 POKE PINK,G
6085 POKE DUR,F
6090 RANDOMIZE USR SON
6095 NEXT G: NEXT F
6096 RETURN
6100 REM
6105 REM BRUIT 3
6110 REM
6115 POKE DUR,1
6120 POKE PINK,2
6122 FOR G=1 TO 5
6125 FOR F=1 TO 10
6130 RANDOMIZE USR SON
6135 NEXT G: NEXT F
6140 RETURN
6150 REM
6155 REM BRUIT 4
6160 REM
6170 FOR F=1 TO 5
6175 RANDOMIZE USR SON
6180 NEXT F
6190 RETURN
6200 REM
6201 REM ECRAN BICOLORE
6202 REM
6210 POKE 50401,C1: REM C1=COULE
UR DU HAUT
6220 POKE 50423,C2: REM C2=COULE
UR DU BAS
6230 POKE 50426,C2
6240 RANDOMIZE USR 50400
6250 RETURN
6300 REM
6301 REM SCROLL DROITE
6302 REM
6310 RANDOMIZE USR 50430

```

```

6320 RETURN
6350 REM
6351 REM SCROLL GAUCHE
6352 REM
6355 RANDOMIZE USR 50454
6360 RETURN
6400 REM
6401 REM SON 2
6402 REM
6410 RANDOMIZE USR 50300
6420 RETURN
6450 REM
6451 REM COULEUR ECRAN
6452 REM
6460 RANDOMIZE USR 50200
6470 RETURN
7000 REM
7001 REM ROUTINE PRINT LENT
7002 REM
7010 FOR K=1 TO LEN A$
7020 PRINT INK I;AT X,Y;A$(K TO
K);
7030 BEEP .05,-40
7040 LET Y=Y+1
7050 NEXT K
7060 RETURN
7100 REM
7110 REM LIGNES !!!!
7120 REM
7130 FOR G=0 TO 5
7140 LET X=0
7150 FOR F=0 TO 16
7155 POKE 23296,X
7156 LET X=X+8
7160 IF X>56 THEN LET X=0
7165 RANDOMIZE USR GAUCHE: RANDO
MIZE USR DROITE
7170 NEXT F
7175 NEXT G
7180 RETURN
7200 REM
7210 REM LIGNES==
7220 REM
7230 FOR G=0 TO 5
7235 LET X=0
7240 FOR F=1 TO 13
7245 POKE 23296,X
7250 LET X=X+8
7255 IF X>56 THEN LET X=0
7260 RANDOMIZE USR HAUT: RANDOMI
ZE USR BAS
7265 NEXT F
7270 NEXT G
7280 RETURN
7300 REM
7305 REM EFFACEMENT ( )
7310 REM
7320 LET X=79
7325 POKE 23296,X
7330 FOR F=1 TO 16
7335 RANDOMIZE USR HAUT: RANDOMI

```

```

ZE USR GAUCHE
7340 NEXT F
7345 PAUSE 10
7350 RETURN
7400 REM
7405 REM EFFACEMENT ( )
7410 REM
7420 LET X=20
7425 POKE 23296,X
7430 FOR F=0 TO 16
7435 RANDOMIZE USR BAS: RANDOMIZ
E USR GAUCHE
7440 NEXT F
7445 PAUSE 10
7450 RETURN
7500 REM
7505 REM DRAPEAU
7510 REM
7520 LET X=0
7525 FOR F=0 TO 60
7530 LET X=X+8
7535 IF X>128 THEN LET X=0
7540 POKE 23296,X
7545 RANDOMIZE USR HAUT: RANDOMI
ZE USR BAS: RANDOMIZE USR GAUCHE
: RANDOMIZE USR DROITE
7550 NEXT F
7560 RETURN
7600 REM
7605 REM BRUIT 5
7610 REM
7620 FOR F=12 TO 2 STEP -1
7625 POKE PINK,F
7630 RANDOMIZE USR SON
7640 NEXT F
7645 RETURN
7646 REM
7647 REM BRUIT 6
7649 REM
7650 FOR F=8 TO 2 STEP -1
7655 POKE PINK,F
7660 RANDOMIZE USR SON
7665 NEXT F
7670 RETURN
7800 REM
7810 REM EFFACEMENT - -
7815 REM
7820 POKE 23296,P
7825 FOR F=0 TO 16
7830 RANDOMIZE USR GAUCHE: RANDO
MIZE USR DROITE
7840 NEXT F
7850 NEXT F
7997 REM
7998 REM ROUTINE EFFACEMENT +
7999 REM
8000 POKE 23296,P: FOR F=1 TO 16
: RANDOMIZE USR GAUCHE: RANDOMIZ
E USR DROITE: RANDOMIZE USR HAUT
: RANDOMIZE USR BAS: NEXT F: RET
URN

```

```

8010 FOR F=1 TO 16
8020 RANDOMIZE USR GAUCHE: RANDO
MIZE USR DROITE: RANDOMIZE USR H
AUT: RANDOMIZE USR BAS
8030 NEXT F
8040 RETURN
8100 REM
8101 REM EFFACEMENT =
8105 POKE 23296,P: REM DETERMINE
LA COULEUR DU PAPIER(10=BLEU,20
=ROUGE,30=MAGENTA...ETC)
8110 FOR F=1 TO 16: REM APPELLE L
A ROUTINE 16 FOIS POUR EFFACER C
OMPLETEMENT L'ECRAN
8120 RANDOMIZE USR HAUT: RANDOMI
ZE USR BAS
8130 NEXT F
8140 RETURN
8200 REM
8201 REM ECRAN BICOLORE
8203 REM
8900 REM *****
8910 REM $LECTURE CODES MACHINE$
8920 REM *****
9000 CLEAR 49999: RESTORE 9112
9001 PRINT AT 0,12; INK 7; PAPER
2;"ATTENDEZ"
9005 PRINT AT 11,3; FLASH 1;"ENT
REE DES CODES MACHINES": FOR A=5
0000 TO 50146
9010 READ B
9020 POKE A,B
9030 NEXT A
9040 LET X=0: FOR F=50000 TO 501
46
9050 LET X=X+PEEK F
9060 NEXT F
9070 IF X<>10232 THEN GO TO 9890
9072 RESTORE 9170: LET T=0: FOR
N=50200 TO 50225: READ A: POKE N
,A: LET T=T+A: NEXT N: IF T<>242
4 THEN GO TO 9890
9075 RESTORE 9180: LET T=0: FOR
N=50300 TO 50329: READ A: POKE N
,A: LET T=T+A: NEXT N: IF T<>314
2 THEN GO TO 9890
9080 LET S=0: RESTORE 9200: FOR
I=50400 TO 50477: READ N: POKE I
,N: LET S=S+N: NEXT I: IF S<>872
3 THEN GO TO 9890
9095 REM *****
9100 REM $ DATAS CODES MACHINE $
9110 REM *****
9112 DATA 33,0,88,58,0,91,14,24,
6,15,35,94,43,115,35,16,249,119,
35,35,35,35,35,35,35,35,35,35,
35,35,35,35,35,13,32,226,201
9115 DATA 17,0,88,33,0,0,1,224,2
,237,176,201
9120 DATA 33,255,90,58,0,91,14,2
4,6,15,43,94,35,115,43,16,249,11

```



```

9,43,43,43,43,43,43,43,43,43,43,
43,43,43,43,43,43,43,13,32,226,2
01
9130 DATA 33,32,88,17,0,88,1,128
,1,237,176,58,0,91,6,32,18,19,16
,252,201
9150 DATA 33,223,90,17,255,90,1,
96,1,237,184,58,0,91,6,32,18,27,
16,252,201
9160 DATA 6,255,33,0,6,17,10,0,4
3,205,181,3,16,250,201
9170 DATA 33,0,88,1,3,0,126,198,
1,230,7,95,126,198,8,230,56,131,
119,35,16,240,13,32,237,201
9180 DATA 243,17,16,208,38,10,58
,72,92,31,31,31,14,254,238,16,23
7,121,67,16,254,37,32,244,28,21,
32,232,251,201,201
9200 DATA 22,40,33,255,87,6,12,1
97,6,32,35,126,230,199,130,119,1
6,248,193,16,242,122,254,32,200,
22,32,24,232,201,6,0,197,33,255,
63,6,192,197,175,6,32,35,203,30,
16,251,193,16,244,193,16,235,201
,6,0,197,33,0,88,6,192,197,175,6
,32,43,203,22,16,251,193,16,244,
193,16,235,201
9700 CLEAR 63999 : REM
      INITIALISATION DES
      CARACTERES MAGNETIQUES.
9701 PRINT AT 0,12; INK 7; PAPER
2;"ATTENDEZ"
9702 PRINT AT 11,1; FLASH 1;"ENT
REE DES LETTRES MAGNETIQUES"
9705 FOR X=0 TO 767

```

```

9710 LET Y=PEEK (15616+X)
9715 POKE 64000+X,Y
9720 NEXT X
9725 RESTORE 9830
9730 LET V=0
9735 FOR X=64128 TO 64207
9740 READ NUM
9745 LET V=V+NUM
9750 POKE X,NUM
9755 NEXT X
9760 IF V<>4044 THEN GO TO 1000
9765 LET V1=0
9770 RESTORE 9855
9775 FOR X=64264 TO 64471
9780 READ NUM
9785 LET V1=V1+NUM
9790 POKE X,NUM
9795 NEXT X
9800 IF V1<>12576 THEN GO TO 100
0
9802 GO TO 15
9805 REM
9810 REM NOMBRES DE 0 A 9
9815 REM
9825 REM
9830 DATA 0,126,98,82,74,70,126,
0,0,24,8,8,12,12,12,0,0,126,2,2,
126,64,126,0,0,124,4,60,6,6,126,
0,0,96,96,96,102,126,102,0
9835 DATA 0,126,64,64,126,2,126,
0,0,120,72,64,126,66,126,0,0,126
,66,6,28,16,16,0,0,60,36,36,126,
102,126,0,0,62,34,62,6,6,6,0
9840 REM
9845 REM LETTRES DE A a Z

```

```

9850 REM
9855 DATA 0,62,34,34,62,98,98,0,
0,124,100,100,126,98,126,0,0,126
,70,64,64,66,126,0,0,124,98,98,9
8,98,124,0,0,124,64,124,64,96,12
4,0
9860 DATA 0,126,96,96,124,32,32,
0,0,126,66,64,78,70,126,0,0,66,6
6,126,98,98,98,0,0,8,8,8,24,24,2
4,0,0,6,6,6,70,70,126,0
9865 DATA 0,98,100,120,126,102,1
02,0,0,32,32,32,96,96,126,0,0,11
0,86,70,70,70,86,0,0,98,82,74,70
,66,90,0,0,126,66,66,66,66,126,0
9870 DATA 0,124,68,124,96,96,96,
0,0,124,68,68,70,70,126,0,0,124,
68,124,102,102,102,0,0,126,64,12
6,2,98,126,0,0,126,24,24,24,24,2
4,0
9885 DATA 0,70,70,70,70,70,126,0
,0,98,98,98,100,104,112,0,0,106,
106,106,74,74,126,0,0,102,36,24,
36,102,102,0,0,66,66,102,24,24,2
4,0,0,126,4,8,16,38,126,0
9890 PRINT INK 2; PAPER 6; BRIGH
T 1; FLASH 1; AT 13,5;"ERREUR DAN
S LES DATAS": BEEP 2,-10: STOP
9895 REM
9900 LET P=0: LET B=1: LET A$="D
END": GO SUB 154: PRINT AT 21,0;
" ": RETURN
9999 REM CHRISTOPHER POVSE VOUS
      REMERCIE D'AVOIR TAPER
      SON PROGRAMME JUSQU'A
      LA FIN.

```

MSX

# LEM SX

Langage : Basic

Certains se diront qu'ils ont déjà vu ce programme quelque part. Il s'agit en effet de la transposition d'un programme Thomson paru dans le n° 13 de Micro 7 (ça remonte à loïn). Ceux qui ont ce numéro pourront constater qu'il y a une grande ressemblance entre les deux programmes. Les différences essentielles venant de la taille de l'écran, qui est plus petit sur MSX (32 x 24 au lieu de 40 x 25) et de la manière de gérer celui-ci. L'ins-

truction DEFGR\$ de Thomson n'existant pas, je l'ai simulée en transformant le jeu de caractères dans la VRAM (voir Images, Imagine... numéros précédents). La seule différence est que j'utilise un mode écran bâtarde (une sorte de SCREEN 1 étendu dans lequel chaque caractère peut avoir plusieurs couleurs). Les caractères sont redéfinis avec leurs couleurs.

Le Jeu : au lancement du programme, vous avez le choix entre : jouer avec

les paramètres tels qu'ils sont, un rappel des règles, le changement des paramètres ou la fin du programme. Le choix s'effectue avec le joystick (celui du clavier ou un autre). Vous devez poser votre vaisseau sur la piste d'atterrissage (en rouge et blanc). La gravitation de la planète vous attire vers le sol. Pour freiner la chute, poussez le joystick. La réaction dépend de la puissance du réacteur principal. Il en est de même pour les réacteurs latéraux. Le

tableau de bord vous indique la vitesse verticale (négative lorsque l'on remonte), l'altitude, la vitesse horizontale (positive lorsque l'on va vers la droite, négative vers la gauche), les coordonnées horizontales de votre véhicule par rapport au milieu de la piste. Pour que l'atterrissage soit correct, il faut se poser sur la piste, que la vitesse latérale soit inférieure à 1 et la vitesse de chute inférieure à 5. Surveillez le tableau de bord qui est plus précis que l'écran. Attention aux montagnes. La position de

```

1 ' SAVE "LEM-SX.BAS
10 DEFINT A-U:DEFSNG V-Z:KEY OFF
20 SCREEN 1:COLOR 15,4,0:WIDTH 3
2:VDP(0)=VDP(0) OR 2:VDP(1)=VDP(
1) AND &HE7:VDP(3)=&HDF:VDP(4)=0
30 GOTO 1000
100 ' ATTERISSAGE -----
110 ZM=PL/2+PG-.5
120 X=RND(1)*22+3:Y=0:AX=X:AY=Y:
FU=FR:LF=FU*.1:FY=YP:FX=XP
130 XV=RND(1)*10-5:YV=RND(1)
140 LOCATE X,Y:PRINT GR$(0);
150 ' BOUCLE JEU -----
160 BEEP:IF FU>0 THEN PT=STICK(M
A) ELSE :BEEP:FU=0:GOTO 250
205 IF PT=1 THEN YV=YV-.5*YP/(M
V+FU*VF):FU=FU-FY ELSE IF PT=3 TH
EN XV=XV+.5*XP/(MV+FU*VF):FU=FU-
FX ELSE IF PT=7 THEN XV=XV-.5*XP
/(MV+FU*VF):FU=FU-FX
250 YV=YV+.5*YG:Y=Y+YV*ZE:X=X+XV
*ZE
255 LOCATE 1,20:PRINT GR$(11-(YV
>YS));:LOCATE T1,20: PRINT GR$(1
1-(ABS(XV)>XS));
260 LOCATE 1,21:PRINT USING IM%;
YV%;:LOCATE T1:PRINT USING IM%;XV
;
270 LOCATE 1,18:PRINTUSING IM%;M
B-Y%;:LOCATE T1:PRINT USING IM%;X
-ZM-.5%;:LOCATE23,20:PRINT USING
I2%;FU%;:IF FUKLF THEN BEEP
300 IF Y>=MB THEN GOTO 500 ELSE
IF X<=E OR X>=ED OR Y<=EH THEN LOC
ATE AX,AY:PRINT " ";:GOTO 400
330 LOCATE AX,AY:PRINT " ";
340 LOCATE X,Y:PRINT GR$(0);
350 AX=X:AY=Y
355 IF AY>=MT(X) THEN 500
400 GOTO 150
500 OK=0
510 IF YV>YS THEN OK=OK-1
520 IF ABS(XV)>XS THEN OK=OK-1
530 IF AX<PG OR AX>PG+PL-1 THEN
OK=OK-1
590 IF OK=0 THEN GOSUB 3900 ELSE
GOSUB 3800

```

la piste et la hauteur des montagnes sont calculées par l'ordinateur en fonction des paramètres que vous aurez choisis.

Changement des paramètres. On peut modifier au joystick :

- la hauteur maximale des montagnes ;
- la largeur de la piste ;
- la gravité ;
- la masse du vaisseau ;
- la puissance du réacteur principal ;
- la puissance des réacteurs latéraux ;
- la quantité de fuel ; le chiffre du haut

```

980 LOCATE 30,23:PRINT GR$(9);
990 IF STRIG(MA)=0 THEN 990
999 RETURN
1000 ' JEU-----
1010 CLS:LOCATE0,0,0
1020 GOSUB 30000
1090 '
1600 CLS
1610 LOCATE 10,0:PRINT "### LEM
SX ###"
1615 LOCATE 16,2:PRINT GR$(0);
1617 LOCATE 12,4:PRINT "Choisiss
ez"
1620 LOCATE 2,18:PRINT GR$(3) " "
GR$(4) " "GR$(5) " "GR$(6);TAB(10)
"= Direction@manette"
1630 LOCATE 5,21:PRINT GR$(9);TA
B(10)"= Bouton"
1640 '
1700 LOCATE 13, 8:PRINT "ON JOUE
"
1705 LOCATE 22,10:PRINT "REGLER"
1710 LOCATE 20,11:PRINT "PARAMET
RES"
1715 LOCATE 6,10:PRINT "MODE"
1717 LOCATE 5,11:PRINT "EMPLOI"
1720 LOCATE 11,14:PRINT "FIN PRO
GRAMME"
1722 LOCATE 16,10:PRINT GR$(3);
1723 LOCATE 14,11:PRINT GR$(6);
1724 LOCATE 18,11:PRINT GR$(4);
1725 LOCATE 16,12:PRINT GR$(5);
1727 FOR MA=0 TO 2:C1=STICK(MA);
IF C1=0 THEN NEXT:GOTO 1727
1745 IF C1=1 THEN GOSUB 3000 ELS
E IF C1=3 THEN GOSUB 2000 ELSE I
F C1=5 THEN C=1 ELSE IF C1=7 THE
N GOSUB 6200
1900 IF C1<>5 THEN 1090
1910 CLS:SCREEN 1:COLOR 15,1
1999 END
2000 ' PARAMETRES -----
2110 CLS:LOCATE 20,EB:PRINT "ACC
EPTÉ ";GR$(9);
2115 LOCATE 5,4:PRINT "VAISSEAU
":LOCATE 7,6:PRINT GR$(0);
2120 LOCATE 4,10:PRINTGR$(3):LOC

```

de l'écran sert de multiplicateur, afin de pouvoir faire varier les chiffres par dizaines, centaines, milliers... Attention le fuel pèse lourd ! Plus un réacteur est puissant, plus il consomme...

Vous pouvez évidemment améliorer ce programme en y ajoutant des sprites, des étoiles dans le ciel ou ce que vous voudrez. ■

François DUPIN

```

ATE 4,12:PRINT GR$(5);:LOCATE 0,
0:PRINT "HAUTEUR MAX":PRINT" MON
TAGNE ";
2125 LOCATE 1,17:PRINT MB-MH;:FO
R I=0 TO MB-MH:LOCATE 2,MB-I:PRI
NT GR$(1);:NEXT I:IF MH>3 THEN F
OR I=3 TO MH-1:LOCATE 2,I:PRINT
"@";:NEXT I
2130 FOR PT=0 TO -1 STEP -1
2140 PT=STRIG(MA)
2150 C1=STICK(MA):IF C1=0 THEN 2
190
2160 IF C1=1 AND MH>3 THEN MH=MH
-1:LOCATE 2,MH:PRINT GR$(1);
2170 IF C1=5 AND MH<MB-3 THEN LO
CATE 2,MH:PRINT "@";:MH=MH+1
2180 LOCATE 1,17:PRINT MB-MH;
2190 NEXT PT
2200 '
2210 LOCATE 8,13:PRINT GR$(6) "
"GR$(4);:LOCATE 9,17:PRINT "Larg
eur@piste";
2220 LOCATE 22,17:PRINT PL;:FOR
I=1 TO PL:LOCATE I+2,MB:PRINT GR
$(2);:NEXT I:IF PL<26 THEN FOR I
=PL+1 TO 26:LOCATE I+2,MB:PRINT
"@";:NEXT I
2230 FOR PT=0 TO -1 STEP -1
2240 PT=STRIG(MA)
2250 C1=STICK(MA):IF C1=0 THEN 2
285
2260 IF C1=3 AND PL<ED-5 THEN PL
=PL+1:LOCATE PL+2,MB:PRINT GR$(2
);
2270 IF C1=7 AND PL>1 THEN LOCAT
E PL+2,MB:PRINT "@";:PL=PL-1
2280 LOCATE 22,17:PRINT PL;
2285 Z=RND(1)
2290 NEXT PT
2300 ' PARAMETRES-----
2305 LOCATE 20,0:PRINT "Mult "GR
$(6) " "GR$(4)
2307 LOCATE 20,3:PRINT "Modif "G
R$(3) " "GR$(5)
2310 T$="GRAVITE":L=5:ZS=20:ZI=1
:ZB=Y6:ZP=.01:GOSUB 5000:Y6=ZB
2320 T$="MASSE":L=7:ZS=MS:ZI=MI:

```

```

ZB=MV:ZP=1:GOSUB 5000:MV=ZB
2330 YP=5*Y6#MV:IF YP>HV THEN YP
=HV
2332 T$="PUIS R.P.":L=9:ZS=HV:ZI
=0:ZB=YP:ZP=1:GOSUB 5000:YP=ZB
2340 XP=MV:T$="PUIS LAT.":L=11:Z
S=HV:ZI=0:ZB=XP:ZP=.1:GOSUB 5000
:XP=ZB
2350 FR=1000:T$="FUEL":L=13:ZS=H
V:ZI=0:ZB=FR:ZP=10:GOSUB 5000:FR
=ZB
2999 RETURN
3000 ' PARTIE -----
3010 CLS
3050 GOSUB 20000
3060 GOSUB 10000
3080 GOSUB 100
3099 RETURN
3800 ' BDM -----
3802 IF X<EG THEN X=EG
3803 IF X>=ED THEN X=ED-1
3805 LOCATE AX,AY:PRINT " ";:SOU
ND 0,0:SOUND 6,15:SOUND 7,7:FOR
I=B TO 10:SOUND I,16:NEXT
3810 SOUND 13,0:SOUND 12,90:FOR
K=0 TO 5:FOR W=0 TO15:LOCATE X,Y
:PRINT GR$(7+(K MOD2));:COLOR ,,
W:NEXT:NEXT:COLOR,,1
3850 LOCATE 0,0:PRINT "VITESSE D
E CHUTE ";:IF YV>YS THEN PRINT "
TROP GRANDE" ELSE PRINT "CORRECT
E":OK=OK+1
3860 LOCATE0,1:PRINT "VITESSE LA
TERALE ";:IF ABS(XV)>XS THEN PRI
NT "TROP GRANDE" ELSE PRINT "COR
RECTE":OK=OK+1
3870 LOCATE 0,2:PRINT "ATERRISSA
GE ";:IF AX<PG OR AX>PG+PL-1 THE
N PRINT "HORS PISTE" ELSE PRINT
"SUR PISTE":OK=OK+1
3880 LOCATE 0,5:PRINT "VOUS AVEZ
FAIT UN TROU DE ";:LOCATE12,6:P
RINTUSING I3$;MV$(XV#XV+YV#YV);:
PRINT"CHR$(253)
3899 RETURN
3900 ' OK-----
3910 LOCATE 5,2:PRINT"*** BIENVE
NUE ***"
3920 PLAY "S1M2000D2T64L16CEGCFD
04B6G#EFDC"
3999 RETURN
5000 ' SP/PARAMETRES-----
5010 LOCATE 15,L:PRINT T$;
5015 IF STRIG(MA) THEN 5015
5030 FOR PT=0 TO -1 STEP-1
5040 PT=STRIG(MA)
5050 C1=STICK(MA)
5054 IF C1=7 THEN ZP=ZP#10:IF ZP

```

```

>1000 THEN ZP=1000
5056 IF C1=3 THEN ZP=ZP/10: IF Z
P<.01 THEN ZP=.01
5060 IF C1=1 THEN ZB=ZB+ZP:IF ZB
>ZS THEN ZB=ZS:BEEP
5070 IF C1=5 THEN ZB=ZB-ZP:IF ZB
<ZI THEN ZB=ZI:BEEP
5080 LOCATE 23,L:PRINT USING IM$
;ZB;
5085 LOCATE 23,1:PRINT USING IM$
;ZP;
5090 NEXT
5099 RETURN
6200 ' MODE D'EMPLOI -----
6230 CLS:PRINT "
Le but du jeu est de poser un
vaisseau spacial sur une planete
"
6235 PRINT:PRINT "
La hauteur des montagnes, la
largeur de la piste d'atterris-
sage, la gravitation, la masse
du vaisseau et la quantite de
fuel disponible peuvent etre
modifiees.
6250 PRINT:PRINT "
La vitesse maximale de chute a
l'atterrissage est de ";YS
6260 PRINT "
La vitesse maximale laterale a
l'atterrissage est de ";XS
6270 PRINT:PRINT :PRINT:PRINT "
Toutes les manoeuvres
s'effectuent par
l'intermediaire d'une manette
et du bouton"
6290 LOCATE 16,23:PRINT GR$(9);
6295 IF STRIG(MA)=0 THEN 6295
6299 RETURN
9260 PRINT "
La vitesse maximale laterale a
l'atterrissage est de ";XS
10000 ' CONSTRUIT MONTAGNES----
10010 HI=MB-(RND(1))*(MB-MH-3))
10030 PG=RND(1)*(ED-PL)
11000 FOR I=EG TO ED
11010 IF I>=PG AND I<PG+PL THEN
MT(I)=MB:GOTO 11100
11030 HD=INT(RND(1)*3)-1:HI=HI+H
D
11040 IF HI<MH OR HI>MB-1 THEN H
I=HI-2#HD
11050 MT(I)=HI
11060 FOR J=MB TO HI STEP -1
11070 LOCATE I,J
11080 PRINT GR$(1);
11090 NEXT
11100 NEXT

```

```

11200 FOR I=PG TO PG+PL-1
11210 LOCATE I,MB
11220 PRINT GR$(2);
11230 NEXT I
11999 RETURN
20000 ' TABLEAU DE BORD -----
20001 LOCATE 0,16:PRINT "
: @@@ALTE@@: @ECOORDE@: @@@@@@@@@@:
: @@@@@@@@@@: @@@@@@@@@@: @@@FUEL@@:
: ";
20002 PRINT "
: @@@@@@@@@@: @@@@@@@@@@:
: @@@VEVERTE: @@@VEHDRE@: @@@@@@@@@@:
: @@@@@@@@@@: @@@@@@@@@@: @@@@@@@@@@:
";
20099 RETURN
30000 ' INITIALISATION -----
30010 DIM GR$(20)
31000 EG=0:ED=31:DIM MT(ED)
31010 EH=0:EB=23
31020 MB=15:MH=10
31030 T1=11:T2=22:T3=24
31040 HV=9999
31050 MS=9999:MI=1:MV=1
31060 PL=10:Y6=1.63
31070 ZE=1/20:VF=1/1000
31080 Y6=9.81/6:YP=5*Y6#MV:XP=MV
:YS=5:XS=1:FR=1000
31100 IM$=" ####.##+":I2$=" ####
#":I3$=" #####,.## "
40040 ' LM DANS TABL NUMERIQUE-
40042 FOR I=0 TO 8:READ A$,B$:X=
VAL("&H"+A$)+VAL("&H"+B$)*256:L0
%(I)=X+65536!*(X>32767):NEXT
40043 FOR I=0 TO 8:READ A$,B$:X=
VAL("&H"+A$)+VAL("&H"+B$)*256:L1
%(I)=X+65536!*(X>32767):NEXT
40090 'UN JEU DE CARACTERE-----
40100 DEFUSR=VARPTR(LO%(0)):LO%(
3)=0:I=USR(0):LO%(3)=2048:I=USR(
0):LO%(3)=4096:I=USR(0)
40120 ' DEUX JEUX DE COULEURS--
40210 DEFUSR1=VARPTR(L1%(0)):L1%(
1)=%H70:L1%(3)=8192:L1%(5)=2048
:J=USR1(0)
40220 DEFUSR1=VARPTR(L1%(0)):L1%(
1)=%H74:L1%(3)=8192+4096:L1%(5)
=2048:J=USR1(0)
40230 DEFUSR1=VARPTR(L1%(0)):L1%(
1)=%HF8:L1%(3)=8192+ASC("0")*8:
L1%(5)=(ASC("9")-ASC("0")+1)*8:J
=USR1(0)
40235 DEFUSR1=VARPTR(L1%(0)):L1%(
1)=%HF0:L1%(3)=8192+ASC(" ")*8+
4096:L1%(5)=(ASC("9")-ASC(" ") +1
)*8:J=USR1(0)

```

```

40250 DEFUSR1=VARPTR(L1%(0)):L1%
(1)=%HF8:L1%(3)=B192+ASC("-")%B+
4096:L1%(5)=B:J=USR1(0)
40260 DEFUSR1=VARPTR(L1%(0)):L1%
(1)=%H30:L1%(3)=B192+ASC("+")%B+
4096:L1%(5)=B:J=USR1(0)
40300 ' REDEF CARACT -ET COUL--
40310 DM=0:J=0
40320 READ W: IF W=-1 THEN 40399
40330 GR$(J)=CHR$(W):J=J+1:D=DM+
W*B:READ W$:FOR I=0 TO 7:VPOKE D
+I,VAL("%H"+MID$(W$,I*2+1,2)):NE
XT:READ W$:FOR I=0 TO 7:FOR K=0
TO 2:VPOKE D+I+B192+K*2048,VAL("
&H"+MID$(W$,I*2+1,2)):NEXT:NE
XT:GOTO 40320
40399 '
49999 RETURN
50160 ' TRANSFERT RAM->VRAM ---
50170 DATA E5,21 '
50180 DATA BF,1B ' AD. CAR ROM
50190 DATA 00,11 '
50200 DATA 00,00 ' AD. VRAM
50210 DATA 00,01 '
50220 DATA 00,08 ' LONGUEUR
50230 DATA CD,5C,00 'APPEL BIOS

```

```

50240 DATA E1,C9,00
50250 ' REMPLIT VRAM -----
50260 DATA E5,3E
50270 DATA 00,00 ' CARACTERE
50280 DATA 00,21 ' HL
50290 DATA 00,00 ' ADRESSE VRAM
50300 DATA 00,01 ' BC
50310 DATA 00,00 ' LONGUEUR
50320 DATA CD,56,00 'APPEL BIOS
50330 DATA E1,C9,00
60070 ' REDEFINITION CARACTERES
60075 ' VEHICULE
60080 DATA 128,187ECB7E3C6642C3
60085 DATA 40404B4040B0B0B0
60088 ' MONTAGNE
60090 DATA 232,7AC45C7DC7BAD5C2
60095 DATA ABABABABABABABAB
60097 ' PISTE
60100 DATA 224,FFFFCCCC3333FFFF
60105 DATA FBFBFBFBFBFBFBFB
60107 ' FLECHE HAUT
60110 DATA 136,183C5A99918181B18
60115 DATA F0F0F0F0F0F0F0F0
60117 ' FLECHE DROITE
60120 DATA 137,0B0402FFFF02040B
60125 DATA F0F0F0F0F0F0F0F0

```

```

60127 ' FLECHE BAS
60130 DATA 138,1B181818995A3C18
60135 DATA F0F0F0F0F0F0F0F0
60137 ' FLECHE GAUCHE
60140 DATA 139,102040FFFF402010
60145 DATA F0F0F0F0F0F0F0F0
60147 ' BOUM 1
60150 DATA 140,B2A159BE39D4A299
60155 DATA B0B0B0B0B0B0B0B0
60157 ' BOUM 2
60160 DATA 141,B2A159BE39D4A299
60165 DATA 0B0B0B0B0B0B0B0B
60167 ' BOUTON
60170 DATA 142,FFC3A59999A5C3FF
60175 DATA F6F6F6F6F6F6F6F6
60180 ' FOND TABLEAU DE BORD
60200 DATA 64,0000000000000000
60210 DATA 7474747474747474
60215 ' VOYANT VERT
60220 DATA 248,183C7EFFFF7E3C18
60230 DATA 2424242424242424
60235 ' VOYANT ROUGE
60240 DATA 249,183C7EFFFF7E3C18
60250 DATA 8484848484848484
60999 DATA -1

```

## YAMAHA

# FM VOICING

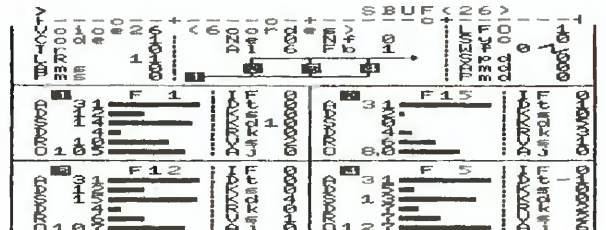
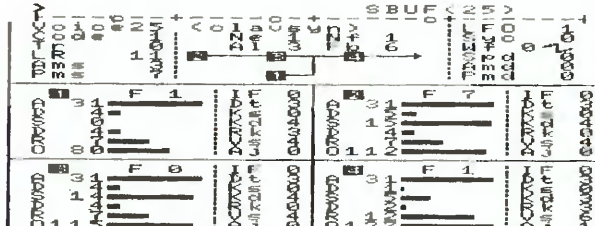
Langage : FM voicing program  
 Machine : Yamaha CX5M ou YIS503F  
 +SFK-01

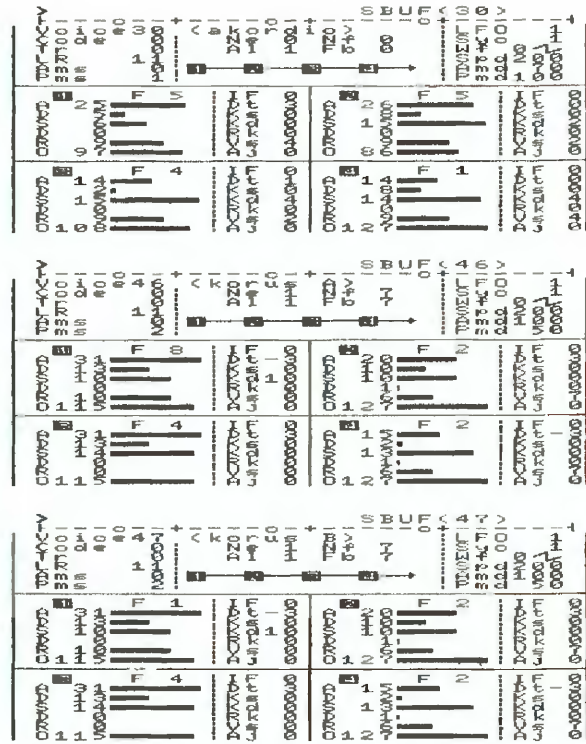
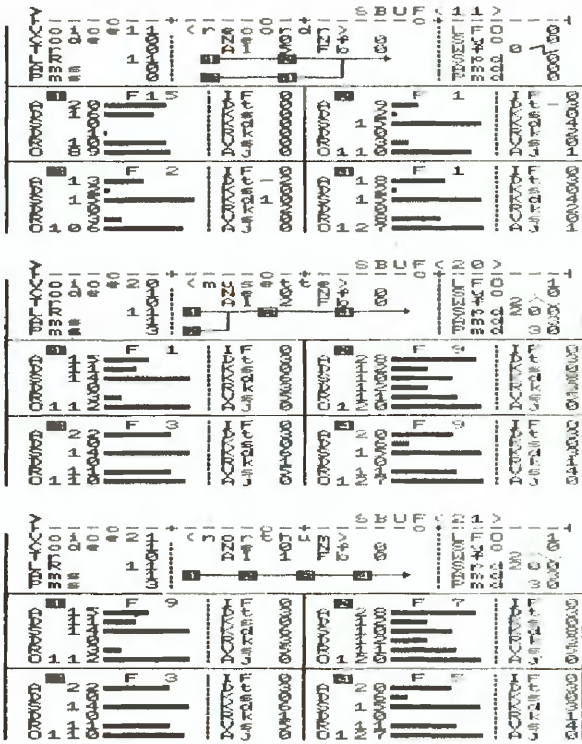
**D**u bleu, du bleu, du bleu (musique Jonasz), c'est pas du billard. J'ai l'blouse, doc. Si vous ne nous envoyez pas plus de sons, nous on laisse béton. Je sais, cette foutue machine n'est pas si facile que ça à programmer, surtout sur cassette.

En plus, les bons préfèrent vendre leurs sonneries. Mais, enfin quoi ! Un effort, par Belzébuth. On doit quand même être un certain nombre à penser que tout le monde y gagne à faire connaître nos trucs. Pour le moment, Jean-Pierre Maingam nous sauve de la

déroute, de banlieue évidemment, avec une collection de sons variés, et non pas à varier, comme me souffle Berthe.

Colette et Berthe MAVALET





# CAHIER DES AS

=  C64

## COMMUS

Langage : L.M. + Basic



**DOMINICO MANFREDI  
GAGNE  
UN SONY HIT BIT  
HB 75 F  
AVEC DRIVE  
HBD 50**

**Dominico Manfredi**  
Vingt-cinq ans. Pour lui, l'informatique est un loisir dont il aurait souhaité faire une profession. Il est actuellement technicien en dépannage de machines de tri postal.

**C**et utilitaire ajoute de nouveaux mots-clés au Basic pour gérer les sons. Les commandes créées par Commus peuvent être utilisées dans un programme Basic quelconque. En voici la liste : 1 - SONOFF : arrêt du son sur les trois voix. 2 - ADSR v,w,x,y,z : définition de l'enveloppe

avec v = voix de 1 à 3, w = attaque de 0 à 15, x = délai de 0 à 15, y = sustain de 0 à 15, z = relâche de 0 à 15. 3 - VOLUME v : définition du niveau sonore avec v de 0 à 15. 4 - WAVE v,f,ph,pb : définition de la forme d'onde avec v = voix de 1 à 3, f = forme d'onde de 1 à 6 (1 = triangle, 2 = scie,

3 = bruit, 4 = rectangle, 5 = rectangle triangle, 6 = rectangle scie), ph = largeur d'impulsion partie haute de 0 à 15 et pb = largeur d'impulsion partie basse de 0 à 255. Les largeurs d'impulsion ne sont à définir que si l'on prend une forme d'onde comportant un signal rectangulaire, c'est-à-dire si f = 4, 5 ou 6. 5 - MUSIC v, fh, fb : définition d'une note avec v = voix de 1 à 3, fh = fréquence de la note partie haute

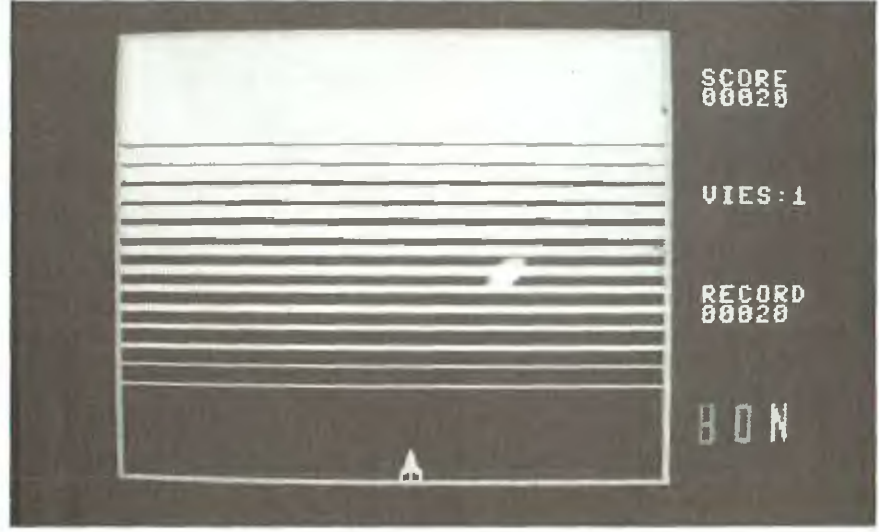
```
10 CS=0:FORI=49152T050014:READA:POKEI,A:
CS=CS+A:NEXTI
20 IFCS<>100215THENSTOP
30 SYS 49152:NEW
49152 DATA 32,68,229,162,0,189,41,195
49160 DATA 157,48,4,189,67,195,157,128
49168 DATA 4,169,1,157,48,216,157,128
49176 DATA 216,232,224,26,208,231,24,162
49184 DATA 6,160,0,32,10,229,169,69
49192 DATA 141,4,3,169,192,141,5,3
49200 DATA 169,8,141,6,3,169,193,141
49208 DATA 7,3,169,61,141,8,3,169
49216 DATA 193,141,9,3,96,166,122,160
49224 DATA 4,132,15,189,0,2,16,7
49232 DATA 201,255,240,62,232,208,244,20
1
49240 DATA 32,240,55,133,8,201,34,240
49248 DATA 85,36,15,112,45,201,63,208
49256 DATA 4,169,153,208,37,201,48,144
49264 DATA 4,201,60,144,29,132,113,160
49272 DATA 0,132,11,136,134,122,202,200
49280 DATA 232,189,0,2,56,249,158,160
49288 DATA 240,245,201,128,208,47,5,11
49296 DATA 164,113,232,200,153,251,1,201
49304 DATA 0,240,56,56,233,58,240,4
49312 DATA 201,73,208,2,133,15,56,233
49320 DATA 85,208,160,133,8,189,0,2
49328 DATA 240,224,197,8,240,220,200,153
49336 DATA 251,1,232,208,240,166,122,230
49344 DATA 11,200,185,157,160,16,250,185
49352 DATA 158,160,208,181,240,15,189,0
49360 DATA 2,16,189,153,253,1,198,123
49368 DATA 169,255,133,122,96,160,0,185
49376 DATA 253,194,208,2,200,232,189,0
```

de 0 à 255 et fb = fréquence de la note partie basse de 0 à 255. 6 -VOFF v : arrêt du son sur une voix avec v de 1 à 3. Pour avoir un aperçu des possibilités de cet utilitaire, vous trouverez à la suite une démonstration. Elle ne peut fonctionner que si vous avez lancé au préalable le premier programme. ■

Dominico MANFREDI

```
49384 DATA 2,56,249,253,194,240,245,201
49392 DATA 128,208,4,5,11,208,153,166
49400 DATA 122,230,11,200,185,252,194,16
49408 DATA 250,185,253,194,208,224,240,1
98
49416 DATA 16,15,36,15,48,11,201,255
49424 DATA 240,7,201,204,176,6,76,36
49432 DATA 167,76,243,166,56,233,203,170
49440 DATA 132,73,160,255,202,240,8,200
49448 DATA 185,253,194,16,250,48,245,200
49456 DATA 185,253,194,48,5,32,71,171
49464 DATA 208,245,76,239,166,32,115,0
49472 DATA 32,70,193,76,174,167,201,204
49480 DATA 144,4,201,210,144,6,32,121
49488 DATA 0,76,237,167,56,233,204,10
49496 DATA 170,189,28,195,72,189,27,195
49504 DATA 72,76,115,0,162,25,169,0
49512 DATA 157,0,212,202,208,248,96,32
49520 DATA 158,183,224,1,240,11,224,2
49528 DATA 240,16,224,3,240,21,76,182
49536 DATA 193,173,4,212,41,254,141,4
49544 DATA 212,96,173,11,212,41,254,141
49552 DATA 11,212,96,173,18,212,41,254
49560 DATA 141,18,212,96,32,158,183,224
49568 DATA 16,176,19,142,24,212,96,32
49576 DATA 158,183,224,1,240,13,224,2
49584 DATA 240,16,224,3,240,19,162,14
49592 DATA 32,55,164,160,0,132,2,76
49600 DATA 205,193,160,7,132,2,76,205
49608 DATA 193,160,14,132,2,32,253,174
49616 DATA 32,158,183,224,16,176,223,138
49624 DATA 10,10,10,10,164,2,153,5
49632 DATA 212,32,253,174,32,158,183,224
49640 DATA 16,176,203,138,141,39,195,164
```

```
49648 DATA 2,185,5,212,13,39,195,153
49656 DATA 5,212,32,253,174,32,158,183
49664 DATA 224,16,176,178,138,10,10,10
49672 DATA 10,164,2,153,6,212,32,253
49680 DATA 174,32,158,183,224,16,176,158
49688 DATA 138,141,39,195,164,2,185,6
49696 DATA 212,13,39,195,153,6,212,96
49704 DATA 32,158,183,224,1,240,11,224
49712 DATA 2,240,14,224,3,240,17,76
49720 DATA 182,193,160,0,132,2,76,76
49728 DATA 194,160,7,132,2,76,76,194
49736 DATA 160,14,132,2,32,253,174,32
49744 DATA 158,183,224,1,240,23,224,2
49752 DATA 240,27,224,3,240,31,224,4
49760 DATA 240,35,224,5,240,41,224,6
49768 DATA 240,47,76,182,193,164,2,169
49776 DATA 17,153,4,212,96,164,2,169
49784 DATA 33,153,4,212,96,164,2,169
49792 DATA 129,153,4,212,96,164,2,169
49800 DATA 65,153,4,212,76,160,194,164
49808 DATA 2,169,81,153,4,212,76,160
49816 DATA 194,164,2,169,97,153,4,212
49824 DATA 32,253,174,32,158,183,224,16
49832 DATA 144,3,76,182,193,164,2,138
49840 DATA 153,3,212,32,253,174,32,158
49848 DATA 183,164,2,138,153,2,212,96
49856 DATA 32,158,183,224,1,240,11,224
49864 DATA 2,240,14,224,3,240,17,76
49872 DATA 182,193,160,0,132,2,76,228
49880 DATA 194,160,7,132,2,76,228,194
49888 DATA 160,14,132,2,32,253,174,32
49896 DATA 158,183,138,164,2,153,1,212
49904 DATA 32,253,174,32,158,183,138,164
49912 DATA 2,153,0,212,96,65,68,83
49920 DATA 210,86,79,70,198,86,79,76
49928 DATA 85,77,197,87,65,86,197,77
49936 DATA 85,83,73,195,83,79,78,79
49944 DATA 70,198,0,166,193,110,193,155
49952 DATA 193,39,194,191,194,99,193,0
49960 DATA 0,42,42,42,32,3,15,13
49968 DATA 13,1,14,4,5,19,32,32
49976 DATA 19,15,14,15,18,5,19,32
49984 DATA 42,42,42,32,32,16,1,18
49992 DATA 32,32,13,1,14,6,18,5
50000 DATA 4,9,32,4,15,13,9,14
50008 DATA 9,3,15,32,32,13,9
```



**PROGRAMME DE DÉMONSTRATION**

```
1 REM DEMO1
2 PRINT CHR$(147)"DEMO1"
5 SONOFF
7 X=1:VOLUME 15
10 ADSR 1,5,9,10,15:MUSIC 1,15,10
20 FOR Z=0 TO 20
25 WAVE 1,X:X=X+1
30 IF X=3 THEN X=1
35 FOR A=0 TO 100:NEXT A,Z
40 VOFF 1
```



```

50 REM DEMO2
55 PRINT"DEMO2"
60 ADSR 1,5,10,10,10:ADSR 2,5,15,9,1
65 FOR X=10 TO 90:MUSIC 1,X,X
67 MUSIC 2,X,0:WAVE 2,3
70 WAVE 1,2
75 FOR A=0 TO 50:NEXT A,X
80 VOFF 1:MUSIC 2,20,0
85 FOR X=15 TO 0 STEP-1
90 VOLUME X
95 FOR A=0 TO 400:NEXT A,X
100 VOFF 2
105 REM DEMO3
110 PRINT"DEMO3"
115 VOLUME 15:ADSR 3,0,0,15,0
120 WAVE 3,1
125 FOR Y=60 TO 5 STEP-1
130 FOR X=1 TO 255 STEP Y
135 MUSIC 3,X,0
140 NEXT X,Y
145 VOFF 3
150 REM DEMO4
155 PRINT"DEMO4"
160 ADSR 1,0,12,0,0:WAVE 1,3
165 FOR X=80 TO 15 STEP -1
170 MUSIC 1,X,0:VOFF1:WAVE1,3
175 FOR A=0 TO 20:NEXT A,X

```

# BONG

Machine : C 64  
Langage : L.M. + Basic

**P**our illustrer le programme précédent et en guise de décontraction, un jeu plein de rebondissement dont le but est d'atteindre des plaques mouvantes avant qu'elles n'atteignent le bas de l'écran (joystick dans le port 2) et dont l'intérêt principal réside dans la sonorisation : vous serez surpris par la qualité de la musique de fond. Après avoir tapé le programme, il est impératif de le sauvegarder avant de le lancer ; en effet le chargement des dessins de sprites détruit le programme Basic. ■

Dominico MANFREDI

```

2 Y=2482 :CS=0 :Z=25
4 PRINT CHR$(147):POKE 646,1
6 POKE 214,10:PRINT:PRINT TAB(7) "OCTETS
A CHARGER :";Y
8 FOR I=49152 TO 51633:READA:POKE I,A:CS=CS+
A
10 IF Y=1000 OR Y=100 OR Y=10 THEN Z=Z+1
12 Y=Y-1:POKE 214,10:PRINT:PRINT TAB(7)

```

```

Y
14 NEXT I
16 IF CS<>257780 THEN STOP
18 SYS 49152
20 DATA 169,0,141,32,208,141,33,208
22 DATA 32,48,196,169,15,141,134,2
24 DATA 32,68,229,162,0,189,2,200
26 DATA 157,211,4,232,224,18,208,245
28 DATA 162,0,189,20,200,157,80,5
30 DATA 189,28,200,157,240,5,189,36
32 DATA 200,157,24,6,232,224,8,208
34 DATA 233,162,0,189,44,200,157,158
36 DATA 7,232,224,28,208,245,32,167
38 DATA 196,165,251,201,250,144,250,173
40 DATA 0,220,41,16,208,249,120,169
42 DATA 49,141,20,3,169,234,141,21
44 DATA 3,88,32,129,255,32,48,196
46 DATA 169,0,141,32,208,141,33,208
48 DATA 32,44,197,173,22,208,9,16
50 DATA 141,22,208,169,4,141,34,208
52 DATA 169,12,141,35,208,169,81,133
54 DATA 251,169,4,133,252,169,0,133
56 DATA 2,166,2,189,240,200,133,38
58 DATA 32,243,195,230,2,165,2,201
60 DATA 9,208,238,162,29,169,44,157
62 DATA 40,4,169,45,157,152,7,202
64 DATA 208,243,169,40,160,4,162,46
66 DATA 32,17,196,169,70,160,4,162
68 DATA 47,32,17,196,162,0,189,249
70 DATA 200,157,152,4,189,5,201,157
72 DATA 192,4,157,120,6,189,11,201
74 DATA 157,136,5,189,255,200,157,80
76 DATA 6,169,7,157,152,216,157,192
78 DATA 216,157,136,217,157,80,218,157
80 DATA 120,218,232,224,6,208,207,162
82 DATA 0,189,28,200,157,64,7,189
84 DATA 36,200,157,104,7,189,17,201
86 DATA 157,64,219,157,104,219,232,224
88 DATA 8,208,230,162,0,169,0,157
90 DATA 0,10,157,0,11,202,208,245
92 DATA 162,0,189,25,201,157,128,10
94 DATA 232,224,35,208,245,162,0,189
96 DATA 60,201,157,192,10,189,107,201
98 DATA 157,0,11,189,136,201,157,64
100 DATA 11,189,78,201,157,128,11,232
102 DATA 224,29,208,227,162,0,160,40
104 DATA 152,157,248,7,232,200,224,7
106 DATA 208,246,162,0,189,165,201,157
108 DATA 41,208,232,224,4,208,245,169
110 DATA 52,141,21,208,169,150,141,4
112 DATA 208,169,228,141,5,208,169,96
114 DATA 141,8,208,169,192,141,10,208
116 DATA 169,96,141,9,208,141,11,208
118 DATA 169,2,133,248,133,191,133,38
120 DATA 133,40,32,151,195,32,151,195
122 DATA 160,2,32,158,196,166,248,240
124 DATA 7,162,0,134,248,76,172,193
126 DATA 162,2,134,248,166,248,181,247
128 DATA 240,40,201,1,240,27,201,2
130 DATA 240,26,189,8,208,201,28,240
132 DATA 48,189,9,208,201,234,240,15
134 DATA 222,8,208,254,9,208,76,56
136 DATA 194,76,255,193,76,36,194,76
138 DATA 117,194,189,8,208,201,255,240
140 DATA 23,189,9,208,201,57,240,9
142 DATA 254,8,208,222,9,208,76,56
144 DATA 194,169,1,149,247,76,172,193
146 DATA 169,2,149,247,76,172,193,189
148 DATA 8,208,201,255,240,16,189,9
150 DATA 208,201,234,240,202,254,8,208
152 DATA 254,9,208,76,56,194,169,3
154 DATA 149,247,76,172,193,169,0,149
156 DATA 247,76,172,193,189,8,208,201
158 DATA 28,240,242,189,9,208,201,57
160 DATA 240,228,222,8,208,222,9,208
162 DATA 173,30,208,133,190,165,190,41
164 DATA 24,201,24,240,11,165,190,41
166 DATA 40,201,40,240,10,76,146,193
168 DATA 162,0,134,248,76,91,194,162
170 DATA 2,134,248,173,21,208,41,247
172 DATA 141,21,208,32,115,195,181,247
174 DATA 240,142,201,1,240,175,201,2
176 DATA 240,171,76,248,193,166,248,240
178 DATA 19,165,38,240,60,173,21,208
180 DATA 41,223,141,21,208,169,0,133
182 DATA 38,76,156,194,165,40,240,41
184 DATA 173,21,208,41,239,141,21,208
186 DATA 169,0,133,40,169,0,141,20
188 DATA 212,169,11,141,19,212,169,10
190 DATA 141,15,212,169,128,141,18,212
192 DATA 169,129,141,18,212,198,191,240
194 DATA 3,76,146,193,206,141,5,173
196 DATA 141,5,201,48,240,3,76,103
198 DATA 193,169,240,141,20,212,169,0
200 DATA 141,19,212,169,17,141,18,212
202 DATA 169,52,133,34,169,1,133,35
204 DATA 160,2,32,158,196,166,34,230
206 DATA 35,165,35,201,255,240,14,202
208 DATA 208,245,206,34,208,165,35,141
210 DATA 15,212,76,224,194,169,1,133
212 DATA 35,198,34,165,34,201,5,240
214 DATA 3,76,224,194,169,16,141,18
216 DATA 212,162,0,142,34,208,142,21
218 DATA 208,189,169,201,157,83,7,169
220 DATA 1,157,83,219,232,224,9,208
222 DATA 240,173,0,220,41,16,208,249
224 DATA 162,11,169,32,157,82,7,202
226 DATA 208,248,169,4,141,34,208,169
228 DATA 53,141,141,5,160,1,185,191
230 DATA 4,217,119,6,240,5,176,11
232 DATA 76,102,195,192,4,240,4,200
234 DATA 76,70,195,160,4,185,191,4
236 DATA 153,119,6,136,208,247,160,4
238 DATA 169,48,153,191,4,136,208,248
240 DATA 76,103,193,138,72,162,4,189
242 DATA 191,4,201,57,240,6,254,191
244 DATA 4,104,170,96,202,240,8,169
246 DATA 48,157,192,4,76,119,195,169
248 DATA 48,141,192,4,76,115,195,173
250 DATA 0,220,41,31,201,27,240,7
252 DATA 201,23,240,16,76,190,195,173
254 DATA 4,208,201,32,144,16,206,4
256 DATA 208,76,190,195,173,4,208,201
258 DATA 251,176,3,238,4,208,169,8
260 DATA 44,21,208,208,18,173,0,220
262 DATA 41,16,208,29,173,4,208,141

```

264 DATA 6,208,169,228,141,7,208,173  
 266 DATA 21,208,9,8,141,21,208,173  
 268 DATA 7,208,201,57,240,4,206,7  
 270 DATA 208,96,173,21,208,41,247,141  
 272 DATA 21,208,96,162,2,160,0,165  
 274 DATA 38,145,251,200,192,29,208,249  
 276 DATA 165,251,24,105,40,133,251,165  
 278 DATA 252,105,0,133,252,202,208,229  
 280 DATA 96,133,251,132,252,134,2,162  
 282 DATA 23,160,0,165,2,145,251,165  
 284 DATA 251,24,105,40,133,251,165,252  
 286 DATA 105,0,133,252,202,208,234,96  
 288 DATA 173,14,220,41,254,141,14,220  
 290 DATA 165,1,41,251,133,1,169,0  
 292 DATA 133,251,169,216,133,252,162,4  
 294 DATA 160,0,169,0,145,251,136,208  
 296 DATA 251,230,252,202,208,242,169,0  
 298 DATA 133,251,133,253,169,208,133,252  
 300 DATA 169,48,133,254,162,4,160,0  
 302 DATA 177,251,145,253,136,208,249,230  
 304 DATA 252,230,254,202,208,240,162,0  
 306 DATA 189,72,200,157,216,48,232,224  
 308 DATA 168,208,245,165,1,9,4,133  
 310 DATA 1,173,14,220,9,1,141,14  
 312 DATA 220,173,24,208,41,240,9,12  
 314 DATA 141,24,208,96,160,32,206,255  
 316 DATA 207,208,251,136,208,248,96,120  
 318 DATA 169,202,141,20,3,169,196,141  
 320 DATA 21,3,173,17,208,41,127,141  
 322 DATA 17,208,169,1,141,26,208,133  
 324 DATA 2,169,32,141,18,208,133,251  
 326 DATA 88,96,169,1,44,25,208,208  
 328 DATA 3,76,49,234,198,2,208,12  
 330 DATA 169,2,133,2,165,251,201,250  
 332 DATA 176,2,230,251,169,1,141,25  
 334 DATA 208,169,3,141,20,3,169,197  
 336 DATA 141,21,3,165,251,141,18,208  
 338 DATA 169,0,141,33,208,104,168,104  
 340 DATA 170,104,64,169,1,44,25,208  
 342 DATA 208,3,76,49,234,169,16,141  
 344 DATA 18,208,169,202,141,20,3,169  
 346 DATA 196,141,21,3,169,1,141,25  
 348 DATA 208,169,15,141,33,208,104,168  
 350 DATA 104,170,104,64,120,169,95,141  
 352 DATA 20,3,169,197,141,21,3,169  
 354 DATA 79,141,5,212,169,48,141,6  
 356 DATA 212,169,13,141,12,212,169,128  
 358 DATA 141,13,212,169,128,141,9,212  
 360 DATA 169,0,141,10,212,32,216,197  
 362 DATA 169,15,141,24,212,88,96,198  
 364 DATA 2,208,23,169,7,133,2,173  
 366 DATA 252,7,141,255,7,162,0,189  
 368 DATA 253,7,157,252,7,232,224,3  
 370 DATA 208,245,198,171,165,171,240,3  
 372 DATA 76,173,197,169,16,141,4,212  
 374 DATA 169,17,141,4,212,164,163,185  
 376 DATA 229,197,240,19,133,171,185,84  
 378 DATA 198,141,1,212,185,194,198,141  
 380 DATA 0,212,230,163,76,173,197,32  
 382 DATA 216,197,76,122,197,198,253,165  
 384 DATA 253,240,3,76,49,234,169,80  
 386 DATA 141,11,212,169,81,141,11,212  
 388 DATA 164,254,185,48,199,133,253,185

390 DATA 118,199,141,8,212,185,188,199  
 392 DATA 141,7,212,230,254,76,49,234  
 394 DATA 169,5,133,171,133,253,169,0  
 396 DATA 133,163,133,254,96,40,12,12  
 398 DATA 12,12,12,12,12,12,12,12  
 400 DATA 12,12,12,12,12,12,12,12  
 402 DATA 12,12,12,12,12,12,12,12  
 404 DATA 12,12,12,12,12,12,12,12  
 406 DATA 12,12,12,12,12,12,12,12  
 408 DATA 12,12,12,12,12,12,12,12  
 410 DATA 12,12,12,12,12,12,12,12  
 412 DATA 12,12,12,12,12,12,12,12  
 414 DATA 12,12,12,12,12,12,12,12  
 416 DATA 12,12,12,12,12,12,12,12  
 418 DATA 12,12,12,12,12,12,12,12  
 420 DATA 24,24,24,24,24,24,24,24  
 422 DATA 24,24,12,12,12,12,12,12  
 424 DATA 12,12,36,0,0,34,39,41  
 426 DATA 26,39,23,20,19,17,20,26  
 428 DATA 34,23,41,39,46,55,34,32  
 430 DATA 29,26,31,39,46,41,34,26  
 432 DATA 34,43,34,46,34,52,34,55  
 434 DATA 34,39,31,23,31,39,31,41  
 436 DATA 31,46,31,52,31,34,27,26  
 438 DATA 27,34,41,31,26,23,26,31  
 440 DATA 41,27,26,23,26,27,20,19  
 442 DATA 20,23,19,15,13,13,15,20  
 444 DATA 15,13,15,13,20,15,20,17  
 446 DATA 20,15,20,17,20,18,20,17  
 448 DATA 20,27,27,23,39,39,31,52  
 450 DATA 52,46,41,39,34,39,41,31  
 452 DATA 26,0,0,207,18,101,20,18  
 454 DATA 59,178,137,103,178,20,207,59  
 456 DATA 101,18,118,65,207,219,69,20  
 458 DATA 3,18,118,101,207,20,207,219  
 460 DATA 207,118,207,39,207,65,207,18  
 462 DATA 3,59,3,18,3,101,3,118  
 464 DATA 3,39,3,207,160,20,160,207  
 466 DATA 101,3,20,59,20,3,101,160  
 468 DATA 20,59,20,160,178,137,178,59  
 470 DATA 137,129,208,10,129,178,129,10  
 472 DATA 129,208,178,129,178,103,178,129  
 474 DATA 178,103,178,112,178,103,178,160  
 476 DATA 160,59,18,18,3,39,39,118  
 478 DATA 101,18,207,18,101,3,20,0  
 480 DATA 40,24,24,24,24,24,24,24  
 482 DATA 24,24,24,24,24,24,24,24  
 484 DATA 24,24,24,24,24,24,24,24  
 486 DATA 24,24,24,24,24,24,24,24  
 488 DATA 24,24,24,24,24,24,24,24  
 490 DATA 24,24,24,24,24,24,12,12  
 492 DATA 12,12,12,12,12,12,12,12  
 494 DATA 12,12,12,12,12,12,12,12  
 496 DATA 24,24,24,36,12,24,0,8  
 498 DATA 17,13,13,6,0,13,11,9  
 500 DATA 13,6,0,8,17,15,13,13  
 502 DATA 11,7,15,13,13,11,10,13  
 504 DATA 17,11,13,15,10,11,13,11  
 506 DATA 15,7,9,10,5,9,8,7  
 508 DATA 6,6,5,5,6,10,13,13  
 510 DATA 11,10,9,11,15,13,13,11  
 512 DATA 13,15,20,18,17,15,13,15  
 514 DATA 7,10,15,20,0,180,103,10

516 DATA 208,232,0,208,158,196,10,133  
 518 DATA 0,180,103,129,208,10,158,193  
 520 DATA 129,208,10,158,89,208,103,158  
 522 DATA 10,129,89,158,208,158,129,193  
 524 DATA 196,89,45,56,180,193,232,133  
 526 DATA 207,45,232,89,208,10,158,89  
 528 DATA 196,158,129,208,10,158,10,129  
 530 DATA 178,112,103,129,208,129,193,89  
 532 DATA 129,178,13,1,14,6,18,5  
 534 DATA 4,9,32,32,4,15,13,9  
 536 DATA 14,9,3,15,16,18,5,19  
 538 DATA 5,14,20,5,27,32,28,32  
 540 DATA 29,32,30,32,31,32,33,32  
 542 DATA 34,32,35,32,1,16,16,21  
 544 DATA 25,5,26,32,19,21,18,32  
 546 DATA 12,5,32,2,15,21,20,15  
 548 DATA 14,32,4,5,32,20,9,18  
 550 DATA 254,231,231,231,231,231,254  
 552 DATA 126,255,231,231,231,231,231  
 554 DATA 227,227,243,243,243,251,251,251  
 556 DATA 126,255,231,231,231,224,224,239  
 558 DATA 255,231,231,231,231,231,255,254  
 560 DATA 0,0,0,0,0,0,0,0  
 562 DATA 231,231,231,231,231,231,255,126  
 564 DATA 239,239,239,231,231,231,227,227  
 566 DATA 239,231,231,231,231,231,255,126  
 568 DATA 85,85,85,85,85,85,85,85  
 570 DATA 85,85,85,85,85,85,85,0  
 572 DATA 85,85,85,85,85,85,0,0  
 574 DATA 85,85,85,85,85,0,0,0  
 576 DATA 85,85,85,0,0,0,0,0  
 578 DATA 85,85,85,0,0,0,0,0  
 580 DATA 85,85,0,0,0,0,0,0  
 582 DATA 85,0,0,0,0,0,0,0  
 584 DATA 170,170,85,85,85,85,85,85  
 586 DATA 0,0,0,0,0,170,170  
 588 DATA 10,10,10,10,10,10,10,10  
 590 DATA 160,160,160,160,160,160,160,160  
 592 DATA 36,36,37,38,39,40,41,42  
 594 DATA 43,19,3,15,18,5,32,18  
 596 DATA 5,3,15,18,4,48,48,48  
 598 DATA 48,48,32,22,9,5,19,58  
 600 DATA 53,5,5,2,2,4,4,6  
 602 DATA 6,3,0,0,3,0,0,7  
 604 DATA 128,0,7,128,0,7,128,0  
 606 DATA 15,192,0,15,192,0,31,224  
 608 DATA 0,63,240,0,59,112,0,51  
 610 DATA 48,0,51,48,3,0,0,3  
 612 DATA 0,0,3,0,0,3,0,0  
 614 DATA 3,0,0,3,0,0,0,0  
 616 DATA 0,0,0,0,0,0,0,0  
 618 DATA 0,0,255,192,0,255,192,0  
 620 DATA 0,0,0,0,0,0,0,0  
 622 DATA 0,0,0,255,192,0,255,192  
 624 DATA 0,255,192,0,255,192,0,255  
 626 DATA 192,0,255,192,0,255,192,0  
 628 DATA 255,192,0,255,192,0,255,192  
 630 DATA 0,0,0,0,0,0,255,192  
 632 DATA 0,255,192,0,255,192,0,255  
 634 DATA 192,0,255,192,0,255,192,0  
 636 DATA 0,0,0,0,0,3,1,1  
 638 DATA 1,7,1,13,5,32,15,22  
 640 DATA 5,18

