

AMSTRAD

JEUX EN ASSEMBLEUR

ERIC RAVIS



DANS LA MÊME COLLECTION

Amstrad jeux d'action, P. Monsaut
Amstrad premiers programmes, R. Zaks
Amstrad 56 programmes, S.R. Trost
Amstrad exploré, J. Braga
Amstrad programmation en assembleur, G. Fagot-Barraly
Amstrad guide du graphisme, J. Wynford
Amstrad CP/M 2.2, A. d'Hardancourt
Amstrad astrologie, numérologie, biorythmes, P. Bourgault
Amstrad graphisme en trois dimensions, T. Lachant-Robert
Amstrad Multiplan, Amstrad
Amstrad CP/M plus, A. d'Hardancourt
Amstrad Astrocalc, G. Blanc/P. Destrebecq
Amstrad gagnez aux courses, J.-C. Despoine
Amstrad créer de nouvelles instructions, J.-C. Despoine
Amstrad Locoscript, B. Le Du
Amstrad mise au point des programmes BASIC, C. Vivier/Y. Jacob
Amstrad Logo, A. d'Hardancourt (à paraître)
Amstrad programmes en langage machine, S. Webb (à paraître)
Amstrad guide du DOS, Amstrad (à paraître)
Amstrad introduction à la programmation en assembleur du Z80, A. d'Hardancourt (à paraître)
Amstrad systèmes d'exploitation, Amstrad (à paraître)
Amstrad routines en assembleur, J.-C. Despoine (à paraître)
Amstrad mieux programmer en assembleur, T. Lachant-Robert (à paraître)
Amstrad jeux de réflexion, G. Fagot-Barraly (à paraître)

ERIC RAVIS

AMSTRAD

JEUX EN ASSEMBLEUR



Paris • Berkeley • Düsseldorf

Illustrations : Goulven Gallais

Tous les efforts ont été faits pour fournir dans ce livre une information complète et exacte. Néanmoins, SYBEX n'assume de responsabilités ni pour son utilisation, ni pour les contrefaçons de brevets ou atteintes aux droits de tierces personnes qui pourraient résulter de cette utilisation.

Copyright version originale © 1986, Sybex

ISBN 2-7361-0205-7

Tous droits réservés. Toute reproduction même partielle, par quelque procédé que ce soit, est interdite sans autorisation préalable. Une copie par xérographie, photographie, film, bande magnétique ou autre, constitue une contrefaçon passible des peines prévues par la loi sur la protection des droits d'auteur.

SOMMAIRE

Sauvetage	9
Mines	13
Mission	17
Galleries	21
Labyrinthe	25
Défense	29
Attaque	33
Panique	37
Ascenseurs	43
Football	47
Formule 1	51
Piège	57
Slalom	61
S.O.S.	65
Xéros	71
Poste avancé	77
Astéroïdes	85
Arcturus	91

Avant-Propos

Les jeux présentés dans ce livre utilisent des fonctions propres aux micro-ordinateurs Amstrad : son, couleur, graphisme, etc. Pour cette raison, ces programmes ne fonctionnent pas sur les micro-ordinateurs qui ne se conforment pas à cette norme.

Au-delà du jeu lui-même, l'étude de ces programmes vous enseignera de nombreuses techniques de programmation qui vous seront très utiles pour développer vos propres programmes.

NOTES CONCERNANT LES PROGRAMMES

Les programmes doivent être tapés tels qu'ils sont reproduits. Ils doivent être enregistrés sur cassette (ou sur disquette) pour éviter d'avoir à les retaper à chaque utilisation. Utilisez pour cela l'instruction

SAVE " < NOM > "

en remplaçant < NOM > par le nom du programme à enregistrer.

Exemple : pour enregistrer le programme BASE, tapez

SAVE "BASE"

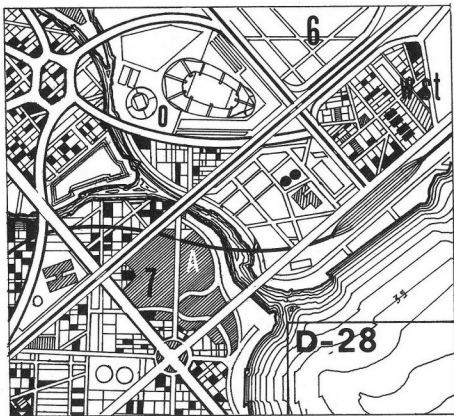
et pressez la touche RETURN après avoir appuyé sur les touches LECTURE et ENREGISTREMENT du magnétophone. Pour recharger un programme, rembobinez la cassette jusqu'au début du programme et tapez LOAD " < NOM > ". Si vous voulez, par exemple, recharger le programme BASE, tapez

LOAD "BASE"

et pressez la touche RETURN.

Afin de réduire au maximum les risques d'erreurs, les listings des programmes ont été reproduits par photographie. Si vos programmes ne fonctionnent pas, vérifiez-les attentivement en considérant plus particulièrement les points suivants :

- lignes oubliées,
- numéros de lignes erronés,
- erreurs de frappe (attention en particulier à ne pas confondre la lettre O et le chiffre 0 ou la lettre I et le chiffre 1).



SAUVETAGE

A l'aide des touches du curseur, manœuvrez votre vaisseau afin de sauver les habitants des villes qui s'enlisent de plus en plus rapidement.

Routines utilisées :

- &BB60 met dans le registre A le code ASCII du caractère de l'écran à la position courante du curseur.
- &BC50 déplace une partie de l'écran d'un caractère. Si B=0, le déplacement est vers le bas, sinon il est vers le haut. A contient le code de l'encre de la ligne qui va apparaître, H contient la première colonne, L contient la dernière ligne, D contient la dernière colonne, E contient la dernière ligne.

```
10 REM *****
20 REM ^SAUVETAGE^
30 REM ^*****
40 MEMORY 10000:PAPER 0
50 SYMBOL 254,255,255,195,195,195,195,255,255
60 SYMBOL 255,0,0,0,0,16,56,16,40
70 ON BREAK GOSUB 80:GOTO 100
80 MODE 1:PEN 3
90 END
100 GOSUB 460
110 MA=100:SC=0:TA=0
120 CLS:GOSUB 540
130 GOSUB 630
140 X=10:Y=3:X1=1:Y1=1:CO=0
150 CO=CO+1:IF CO<MA THEN 250
160 CO=0
170 LOCATE X,Y:PRINT CHR$(32);
```

```

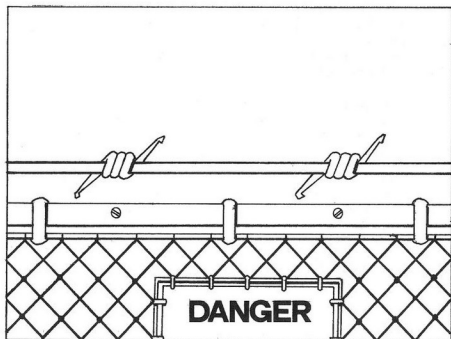
180 CALL 40007:PEN 1
190 LOCATE X,Y:PRINT CHR$(235);
200 S=S+1:IF S<21 THEN 250
210 LOCATE X,Y:PRINT CHR$(32);
220 LOCATE 13,5:PRINT"APPUYER SUR 'FO'"
230 IF INKEY(15)=-1 THEN 230
240 GOTO 100
250 X=X-INKEY(8)+INKEY(1)
260 IF X=0 OR X=41 THEN X=X1
270 Y=Y-INKEY(0)+INKEY(2)
280 IF Y=2 OR Y=26 THEN Y=Y1
290 IF X=X1 AND Y=Y1 THEN 150
300 LOCATE X,Y:CALL 40000:A=PEEK(39999)
310 IF A=32 THEN GOSUB 390:GOTO 150
320 IF A=254 THEN X=X1:Y=Y1:GOTO 150
330 GOSUB 390
340 SC=SC+1:SOUND 1,200:GOSUB 540
350 N=N+1:IF N<NF THEN 150
360 IF MA>10 THEN MA=MA-10
370 TA=TA+1
380 GOTO 120
390 LOCATE X1,Y1
400 PRINT CHR$(32);
410 PEN 1
420 LOCATE X,Y
430 PRINT CHR$(235);
440 X1=X:Y1=Y
450 RETURN
460 MODE 1
470 BORDER 0
480 INK 0,10
490 INK 1,6
500 INK 2,18
510 INK 3,0
520 GOSUB 820
530 RETURN
540 PEN 2
550 LOCATE 11,1
560 PRINT RIGHT$(STR$(10000+SC),4)
570 LOCATE 20,1
580 PRINT RIGHT$(STR$(100+TA),2)
590 IF SC>RE THEN RE=SC
600 LOCATE 27,1

```

```

610 PRINT RIGHT$(STR$(10000+RE),4)
620 RETURN
630 PEN 3
640 FOR X=1 TO 20
650 I=INT(RND(1)*15)+7
660 FOR Y=I TO 25
670 LOCATE X*2-1,Y
680 PRINT CHR$(254)CHR$(254);
690 NEXT Y
700 NEXT X
710 PEN 2:N=0:S=0
720 NF=10+INT(RND(1)*5)
730 FOR I=1 TO NF
740 Y=24:A=INT(RND(1)*40)+1
750 LOCATE A,Y:CALL 40000
760 B=PEEK(39999)
770 IF B=254 THEN Y=Y-1:GOTO 750
780 IF B<>32 THEN 740
790 PRINT CHR$(255)
800 NEXT I
810 RETURN
820 RESTORE 860
830 FOR T=40000 TO 40018
840 READ A:POKE T,A:NEXT T
850 RETURN
860 DATA &cd
870 DATA &60
880 DATA &bb
890 DATA &32
900 DATA &3f
910 DATA &9c
920 DATA &c9
930 DATA &af
940 DATA &47
950 DATA &21
960 DATA &03
970 DATA &00
980 DATA &11
990 DATA &18
1000 DATA &27
1010 DATA &cd
1020 DATA &50
1030 DATA &bc
1040 DATA &c9

```



MINES

Vous devez traverser un champ de mines. Mais attention ! vous êtes chronométré. En début de partie, l'ordinateur affiche les zones minées mais celles-ci seront invisibles par la suite. En appuyant sur la touche < COPY > , elles réapparaîtront mais cela vous fera perdre du temps. Pour vous déplacer, utilisez les touches du curseur.

```
10 REM*****
20 REM^MINES^
30 REM*****
40 MODE 1
50 MEMORY 9000:GOSUB 780
60 BORDER 0
70 INK 0,25:INK 1,0:INK 2,9:INK 3,2
80 SYMBOL 250,7,7,7,7,1,1,127,127
90 SYMBOL 251,1,1,6,6,24,24,24,24
100 SYMBOL 252,224,224,224,224,128,128,254,254
110 SYMBOL 253,128,128,96,96,24,24,24,24
120 PEN 1
130 FOR T=1 TO 40
140 LOCATE T,3
150 PRINT CHR$(225)
160 NEXT T
170 FOR T=1 TO 40
180 LOCATE T,24
190 PRINT CHR$(225)
200 NEXT T
210 K=TIME:PEN 2
220 GOSUB 520:GOSUB 550
230 TE=20
240 GOSUB 640:GOSUB 580
```

```

250 K=TIME:X=1:Y=12
260 GOTO 430
270 GOSUB 520:A=X:B=Y
280 IF INKEY(9)=0 THEN INK 3,2:TE=10:GOSUB 580
290 X=X+INKEY(1)-INKEY(8)
300 Y=Y+INKEY(2)-INKEY(0)
310 IF X=A AND Y=B THEN 270
320 IF X=0 THEN X=1:GOTO 270
330 IF Y=3 OR Y=23 THEN Y=B:GOTO 270
340 SOUND 1,200,4
350 LOCATE A,B:PRINT CHR$(32);CHR$(32);
360 LOCATE A,B+1:PRINT CHR$(32);CHR$(32);
370 E=0
380 FOR C=X TO X+1
390 FOR D=Y TO Y+1
400 GOSUB 750
410 NEXT D:NEXT C
420 IF E=1 THEN SOUND 1,2000:SOUND 1,3000:GOTO 49
0
430 PEN 2
440 LOCATE X,Y:PRINT CHR$(250);CHR$(252);
450 LOCATE X,Y+1:PRINT CHR$(251);CHR$(253);
460 IF X<39 THEN 270
470 SOUND 1,100:SOUND 1,300:SOUND 1,150
480 GOSUB 550
490 INK 3,2
500 LOCATE 13,3:PRINT "APPUYER SUR 'FO'"
510 IF INKEY(15)=-1 THEN 510 ELSE CLS:GOTO 120
520 SC=INT((TIME-K)/10)
530 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)
540 RETURN
550 IF(SC<RE OR RE=0)AND SC>0 THEN RE=SC
560 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
570 RETURN
580 PEN 2:FOR T=TE TO 0 STEP-1
590 LOCATE 20,1:PRINT RIGHT$(STR$(100+T),2)
600 FOR I=1 TO 300:NEXT I
610 NEXT T
620 INK 3,25
630 RETURN
640 PEN 3
650 FOR T=5 TO 35 STEP 5
660 FOR I=0 TO 2

```

```
670 D=4+I*7+INT(RND(1)*5)
680 LOCATE T,D
690 PRINT CHR$(204);CHR$(205);
700 LOCATE T,D+1
710 PRINT CHR$(205);CHR$(204);
720 NEXT I
730 NEXT T
740 RETURN
750 LOCATE C,D:CALL 10000
760 IF PEEK(39000)<>32 THEN E=1
770 RETURN
780 FOR T=10000 TO 10006
790 READ A
800 POKE T,A
810 NEXT T
820 RETURN
830 DATA 205,96,187,50,88,152,201
```



MISSION

Votre mission : franchir vingt tableaux de météorites mais le champ dans lequel vous vous déplacez diminue à mesure que vous progressez dans les tableaux. Pour manœuvrer votre vaisseau, utilisez les touches du curseur droite et gauche.

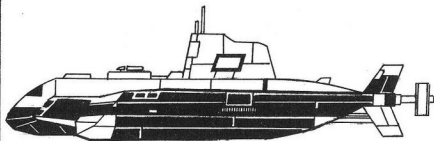
```
10 REM*****
20 REM*MISSION*
30 REM*****
40 GOSUB 390
50 MODE 1:MEMORY 9000
60 BORDER 0
70 INK 0,0:INK 1,24
80 INK 2,23:INK 3,6
90 X=20:SC=0:M=0:P=41:N=0:TA=6
100 L=1:C=40
110 SC=SC+1:N=N+1
120 IF N<100 THEN 170
130 N=0:TA=TA+1:IF TA=25 THEN SOUND 1,100:SOUND 1,
300:SOUND 1,200:FOR T=1 TO 5000:NEXT T:CLS:LOCATE
12,5:PRINT "MISSION ACCOMPLIE":
GOTO 370
140 INK 3,TA:M=M+1:P=P-1:C=C-2:L=L+1
150 IF X=M THEN X=M+1
160 IF X=P THEN X=P-1
170 PEN 2:LOCATE X,25:PRINT CHR$(239);
180 LOCATE X,24:CALL 10008:IF PEEK(39000)<>32 THEN
270
```

```

190 PEN 3:B=INT(RND(1)*C)+L
200 LOCATE B,3:PRINT CHR$(202)
210 CALL 10000
220 A=X:X=X-INKEY(8)+INKEY(1)
230 IF A=X THEN 110
240 IF X=P OR X=M THEN X=A:GOTO 110
250 PEN 2:LOCATE A,25:PRINT CHR$(32);
260 LOCATE X,24:CALL 10008:IF PEEK(39000)=32 THEN
110
270 SOUND 1,2000:SOUND 1,2000
280 FOR T=1 TO 100
290 LOCATE X,25:PRINT CHR$(238);
300 LOCATE X,25:PRINT CHR$(32);
310 NEXT T
320 CLS:PEN 1
330 LOCATE 16,5:PRINT "SCORE :";SC
340 IF SC>RE THEN RE=SC
350 LOCATE 16,7:PRINT "RECORD :";RE
360 LOCATE 16,9:PRINT "TABLEAU:";TA-5
370 LOCATE 13,11:PRINT "APPUYER SUR 'FO'"
380 IF INKEY(15)=0 THEN 50 ELSE 380
390 RESTORE 450
400 FOR T=10000 TO 10014
410 READ A
420 POKE T,A
430 NEXT T
440 RETURN
450 DATA 62,0,6,0,205,77,188,201,205,96,187,50,88,152,2
01

```


6-20.B



GALERIES

Traversez un maximum de galeries avec votre sous-marin en évitant les mines qui deviennent de plus en plus nombreuses. En appuyant sur la touche < COPY > , si votre score n'est pas nul, vous pourrez détruire les obstacles qui se trouvent juste devant votre sous-marin. Pour vous déplacer, utilisez les touches du curseur haute et droite, le poids du sous-marin se chargeant de vous faire descendre.

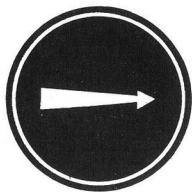
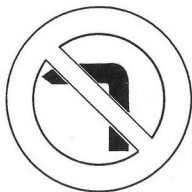
```
10 REM*****
20 REM^GALERIES^
30 REM*****
40 GOSUB 590
50 MODE 1:MEMORY 9000
60 BORDER 0
70 INK 0,1:INK 1,24:INK 2,23:INK 3,6
80 TA=1
90 GOSUB 380
100 SC=0
110 X=1:Y=12
120 GOSUB 310
130 A=X:B=Y
140 IF INKEY(9)=0 AND SC>0 THEN SC=SC-1:SOUND 1,20
:LOCATE A+2,B:PRINT CHR$(32);CHR$(32):GOSUB 310:FOR
T=1 TO 100:NEXT T
150 IF INKEY(1)=0 THEN X=X+1
160 IF INKEY(0)=0 THEN Y=Y-1
170 E=E+1:IF E>35 THEN E=0:Y=Y+1
180 IF X=A AND Y=B THEN 130
190 LOCATE A,B:PRINT CHR$(32);CHR$(32):E=0
200 G=55214+Y*80+X*2
210 IF PEEK(G)=0 AND PEEK(G+2)=0 THEN 270
220 LOCATE X,Y:PRINT CHR$(206);CHR$(206)
```

```

230 SOUND 1,1000:SOUND 1,2000:SOUND 1,1500:SOUND 1
,1000
240 IF SC>RE THEN RE=SC:GOSUB 310
250 LOCATE 13,5:PRINT "APPUYER SUR 'F0'"
260 IF INKEY(15)=0 THEN 50 ELSE 260
270 LOCATE X,Y:PEN 2
280 PRINT CHR$(231);CHR$(202)
290 IF X=39 THEN SC=SC+2:SOUND 1,200:SOUND 1,200:T
A=TA+1:GOSUB 440:GOTO 110
300 GOTO 130
310 LOCATE 11,1:PEN 1
320 PRINT RIGHT$(STR$(SC+10000),4)
330 LOCATE 20,1
340 PRINT RIGHT$(STR$(TA+10000),2)
350 LOCATE 27,1
360 PRINT RIGHT$(STR$(RE+10000),4)
370 RETURN
380 RESTORE 390
390 DATA 255,249,249,249,249,249,249,255
400 FOR T=39000 TO 39007
410 READ A
420 POKE T,A
430 NEXT T
440 CLS
450 CALL 10000
460 C=10:L=5
470 FOR X1=1 TO 40
480 FOR Y1=C TO C+L
490 LOCATE X1,Y1:PRINT CHR$(32)
500 NEXT Y1
510 IF RND(1)<0.5 AND C>4 THEN C=C-1 ELSE IF C+L<2
3 THEN C=C+1
520 NEXT X1
530 PEN 1
540 FOR T=1 TO TA
550 X2=INT(RND(1)*36+4):Y2=INT(RND(1)*18+3)
560 IF PEEK(55214+Y2*80+X2*2)=0 THEN LOCATE X2,Y2:
PRINT CHR$(186) ELSE 550
570 NEXT T
580 RETURN
590 RESTORE 650
600 FOR T=10000 TO 10036
610 READ A

```

620 POKE T,A
630 NEXT T
640 RETURN
650 DATA 33,160,192,1,88,152,229,205,39,39,225,35,124,
254,199,32,242,125,254,208
660 DATA 32,237,201,17,0,8,122,245,10,119,3,25,241,61,3
2,247,201



LABYRINTHE

Descendez à travers le labyrinthe le plus rapidement possible. Pour vous déplacer, utilisez les touches du curseur gauche et droite.

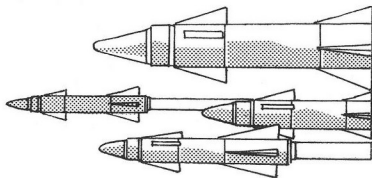
```
10 REM*****
20 REM*LABYRINTHE*
30 REM*****
40 GOSUB 690
50 MODE 1
60 B$=CHR$(233)+STRING$(37,CHR$(32))+CHR$(233)
70 GOSUB 400
80 BORDER 0
90 INK 0,0:INK 1,24:INK 2,23:INK 3,6
100 X=20:K=TIME:N=0
110 GOSUB 340
120 GOSUB 460
130 GOTO 170
140 PEN 2:LOCATE X,3:PRINT CHR$(250);CHR$(252)
150 LOCATE X,4:PRINT CHR$(251);CHR$(253)
160 RETURN
170 GOSUB 140
180 IF PEEK(51518+X*2)<>0 OR PEEK(51520+X*2)<>0 TH
EN 270
190 LOCATE 1,21:GOSUB 520
200 FOR T=1 TO 4:PEN 3
210 IF N>17 THEN 240
220 LOCATE 1,22:PRINT CHR$(233)
230 LOCATE 39,22:PRINT CHR$(233)
240 CALL 20000:GOSUB 140
```

```

250 NEXT T
260 GOTO 180
270 A=X:X=X-INKEY(8)+INKEY(1)
280 GOSUB 340
290 IF X=A THEN 180
300 IF X<2 OR X>37 THEN X=A
310 LOCATE A,3:PRINT CHR$(32);CHR$(32)
320 LOCATE A,4:PRINT CHR$(32);CHR$(32)
330 GOTO 170
340 SC=INT((TIME-K)/20)
350 LOCATE 11,1:PEN 1
360 PRINT RIGHT$(STR$(SC+10000),4)
370 LOCATE 27,1
380 PRINT RIGHT$(STR$(RE+10000),4)
390 RETURN
400 MEMORY 10000
410 SYMBOL 250,7,7,7,7,1,1,127,127
420 SYMBOL 251,1,1,6,6,24,24,24,24
430 SYMBOL 252,224,224,224,224,128,128,254,254
440 SYMBOL 253,128,128,96,96,24,24,24,24
450 RETURN
460 LOCATE 1,5
470 FOR T=1 TO 5
480 GOSUB 520
490 PRINT B$;B$;B$;
500 NEXT T
510 RETURN
520 SOUND 1,200
530 N=N+1:IF N<19 THEN 640
540 IF N<23 THEN RETURN
550 PEN 1
560 IF SC<RE OR RE=0 THEN RE=SC:GOSUB 370
570 RESTORE 580
580 DATA 200,150,200,100,200,100,100
590 FOR T=1 TO 7:READ A
600 SOUND 1,A
610 NEXT T
620 LOCATE 13,6:PRINT "APPUYER SUR 'FO'"
630 IF INKEY(15)=0 THEN 50 ELSE 630
640 PEN 3:A$=STRING$(39,CHR$(233))
650 A=INT(RND(1)^10+5):B=INT(RND(1)^10+20)
660 MID$(A$,A,2)=CHR$(32)+CHR$(32):MID$(A$,B,2)=CHR$(3
2)+CHR$(32)

```

```
670 PRINT A$;  
680 RETURN  
690 RESTORE 740  
700 FOR T=20000 TO 20013  
710 READ A  
720 POKE T,A:NEXT T  
730 RETURN  
740 DATA 6,1,62,0,33,2,0,17,24,39,205,80,188,201
```



DÉFENSE

Défendez votre planète en évitant que l'envahisseur ne s'y pose. Après chaque tir, vous devez attendre que votre base redevienne blanche pour tirer à nouveau. Plus les envahisseurs sont touchés haut, plus ils rapportent de points. Pour manœuvrer votre base, utilisez les touches du curseur gauche et droite et la touche <Z> pour tirer.

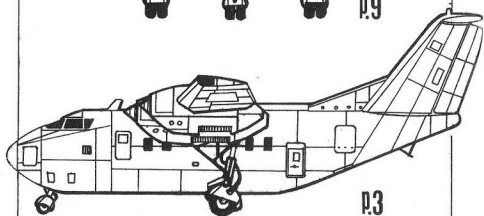
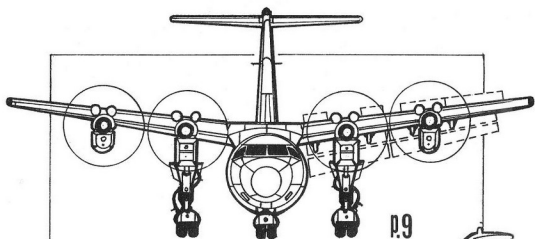
```
10 REM*****
20 REM^DEFENSE^
30 REM*****
40 MODE 1
50 NE=8
60 GOSUB 570
70 BORDER 0
80 INK 0,0:INK 1,6:INK 2,18:INK 3,26
90 MEMORY 10000
100 SYMBOL 250,24,60,60,60,60,126,219,0
110 SYMBOL 251,3,15,25,63,63,99,224,64
120 SYMBOL 252,192,240,152,252,252,198,7,2
130 CLS:X=20:Y1=0:POKE 39203,0:SC=0
140 L=0
150 POKE 39204,NE
160 FOR T=1 TO NE
170 Y(T)=-T^2
180 POKE 39049+T,256-T^2
190 NEXT T
200 LOCATE 1,21:PEN 1
210 FOR T=1 TO 120
220 PRINT CHR$(143);
230 NEXT T
240 GOSUB 390
250 IF L=0 THEN PEN 3 ELSE L=L-1:PEN 1
```

```

260 LOCATE X,20:PRINT CHR$(250);
270 IF Y1=0 THEN 320
280 IF PEEK(48911+X1*2+Y1*80)<>0 THEN GOSUB 480:G
OTO 320
290 LOCATE X1,Y1:PRINT CHR$(32):Y1=Y1-2:LOCATE X1,Y
1:PEN 3:PRINT CHR$(149)
300 IF PEEK(48911+X1*2+Y1*80)<>0 THEN GOSUB 480:G
OTO 320
310 IF Y1=4 THEN LOCATE X1,Y1:PRINT CHR$(32):Y1=0
320 IF INKEY(71)=0 AND L=0 THEN SOUND 1,100:Y1=20:
X1=X:L=14
330 PEN 2:CALL 15000:IF PEEK(39203)=1 THEN 440
340 A=X:X=X+INKEY(1)*2-INKEY(8)*2
350 IF X=A THEN 250
360 IF X=2 OR X=40 THEN X=A:GOTO 250
370 LOCATE A,20:PRINT CHR$(32);
380 GOTO 250
390 PEN 3
400 LOCATE 11,1:PRINT RIGHT$(STR$(SC+10000),4)
410 IF SC>RE THEN RE=SC
420 LOCATE 27,1:PRINT RIGHT$(STR$(RE+10000),4)
430 RETURN
440 SOUND 1,2000:SOUND 1,1000:SOUND 1,2000:SOUND 1
,1500
450 LOCATE 13,5:PEN 1
460 PRINT "APPUYER SUR 'FO'"
470 IF INKEY(15)=-1 THEN 470 ELSE 130
480 SOUND 1,2000
490 LOCATE X1,Y1-2:PRINT CHR$(32);CHR$(32)
500 FOR T=1 TO NE
510 IF X1=PEEK(38999+T) AND Y1-2=PEEK(39049+T) THE
N N=T
520 NEXT T
530 POKE 39049+N,250
540 SC=SC+22-Y1:GOSUB 390
550 LOCATE X1,Y1:PRINT CHR$(32):Y1=0
560 RETURN
570 RESTORE 620
580 FOR T=15000 TO 15207
590 READ A
600 POKE T,A:NEXT T
610 RETURN
620 DATA 58,36,153,79,6,0,33,87,152,9,126,50,33,153,33,

```

187,152,9,126,50
630 DATA 34,153,33,137,152,9,126,50,32,153,254,200,56,
7,60,50,32,153,195,75
640 DATA 59,254,0,32,30,50,34,153,62,4,50,32,153,62,2,5
0,33,153,42,32
650 DATA 153,205,117,187,62,251,205,90,187,60,205,90,1
87,24,104,42,32,153,205,117
660 DATA 187,62,32,205,90,187,205,90,187,58,34,153,254
,1,40,29,58,33,153,254
670 DATA 38,40,7,60,60,50,33,153,24,42,62,1,50,34,153,5
8,32,153,60,60
680 DATA 50,32,153,24,27,58,33,153,254,2,40,7,61,61,50,
33,153,24,13,62
690 DATA 0,50,34,153,58,32,153,60,60,50,32,153,42,32,15
3,205,117,187,62,251
700 DATA 205,90,187,60,205,90,187,58,32,153,254,20,32,
5,62,1,50,35,153,33
710 DATA 87,152,9,58,33,153,119,33,137,152,9,58,32,153,
119,33,187,152,9,58
720 DATA 34,153,119,13,194,158,58,201



ATTAQUE

Détruisez un maximum de tanks à l'aide de votre bombardier. Au départ, vous disposez de cinq bombes mais vous ne pouvez en transporter qu'une seule à la fois. Pour décoller, appuyez sur la touche F1, et sur la touche F2 pour larguer la bombe. Si vous détruisez au moins quatre tanks, vous bénéficiez d'un bonus de cinq bombes.

```
10 REM*****
20 REM^ATTAQUE^
30 REM*****
40 MODE 1
50 BORDER 0
60 INK 0,1:INK 1,2:INK 3,6:INK 2,26
70 MEMORY 10000
80 SYMBOL 250,0,0,0,7,31,127,255,127
90 SYMBOL 251,1,3,7,15,255,255,255,254
100 SYMBOL 249,128,192,224,240,255,255,255,127
110 SYMBOL 253,0,0,0,224,248,254,255,254
120 SYMBOL 254,15,255,255,255,255,255,255,255
130 A$(0)=CHR$(32)+CHR$(249)+CHR$(253)
140 A$(1)=CHR$(250)+CHR$(251)+CHR$(32)
150 B$(0)=CHR$(32)+CHR$(189)+CHR$(189)
160 B$(1)=CHR$(189)+CHR$(189)+CHR$(32)
170 C$=CHR$(32)+CHR$(32)
180 SE=1:X=38:TE=0:T2=1:NB=5:GOSUB 490
190 N=0:SC=0
200 GOSUB 420
210 GOSUB 450:GOSUB 520
220 PEN 2:LOCATE X,8:PRINT A$(SE)
230 IF INKEY(13)=0 THEN TE=1
```

```

240 IF INKEY(14)=0 AND T2=1 AND TE=1 THEN T1=9:X1
=X:T2=0:NB=NB-1:GOSUB 490
250 IF T1>0 THEN LOCATE X1,T1-2:PRINT CHR$(32):LOC
ATE X1,T1:PRINT CHR$(252):T1=T1+2 ELSE 330
260 IF PEEK(59230+X1*2+T1*80)<>0 AND T1<>25 THEN
SOUND 1,1000:LOCATE X1,T1-2:PRINT CHR$(32):T1=0
270 IF T1=25 AND ABS(X1-A-1)<2 THEN SOUND 1,200:S
C=SC+10:N=N+1:GOSUB 450:LOCATE A-1,23:PRINT C$:C$:T
1=0:GOSUB 420
280 IF T1=25 THEN LOCATE X1,T1-2:PRINT CHR$(32):T1
=0:SOUND 1,2000
290 IF T1>0 OR NB<>0 THEN 330
300 IF N>3 THEN N=0:NB=5:GOSUB 490:GOTO 330
310 LOCATE 13,5:PRINT "APPUYER SUR 'FO'"
320 IF INKEY(15)=-1 THEN 320 ELSE CLS:GOTO 180
330 IF S1=1 THEN A=A-1 ELSE A=A+1
340 IF A=1 AND S1=1 THEN LOCATE A+1,23:PRINT C$:A
=38
350 IF A=38 AND S1=0 THEN LOCATE A,23:PRINT C$:A=
1
360 LOCATE A,23:PRINT B$(S1);
370 IF TE=0 THEN FOR T=1 TO 30:NEXT T:GOTO 230
380 IF SE=1 THEN X=X-1 ELSE X=X+1
390 IF X=1 THEN SE=0
400 IF X=38 THEN SE=1:TE=0:T2=1
410 GOTO 220
420 S1=INT(RND(1)*2)
430 IF S1=0 THEN A=1 ELSE A=38
440 RETURN
450 PEN 1
460 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)
470 IF SC>RE THEN RE=SC
480 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
490 PEN 1:LOCATE 20,1:PRINT RIGHT$(STR$(100+NB),2)
500 PEN 2
510 RETURN
520 PEN 3
530 FOR T=9 TO 20
540 LOCATE 46-T+T MOD 2,T:PRINT CHR$(274)
550 FOR I=47-T TO 40
560 LOCATE I,T:PRINT CHR$(143)
570 NEXT I
580 NEXT T

```

```
590 LOCATE 1,24
600 FOR T=1 TO 40:PRINT CHR$(254);
610 NEXT T
620 FOR T=1 TO 40:PRINT CHR$(143);
630 NEXT T
640 RETURN
```



PANIQUE

Une nouvelle race d'envahisseurs est arrivée : les araignées. Tuez-les à l'aide de votre base avant qu'elles n'atteignent le sol. Mais méfiez-vous : elles descendent de plus en plus rapidement. Déplacez votre base à l'aide des touches "<" et ">" et tirez avec la touche <Z>.

```
10 REM*****
20 REM*PANIQUE*
30 REM*****
40 MODE 1
50 SYMBOL 255,24,60,126,126,60,126,231,195
60 GOSUB 620
70 MEMORY 15000
80 BORDER 0
90 INK 0,0:INK 1,24:INK 3,6
100 RESTORE 220
110 FOR T=0 TO 4
120 READ D
130 FOR I=0 TO 7
140 READ A
150 IF D=1 THEN B=A AND 240:C=(A AND 15)*16
160 IF D=2 THEN B=(A AND 240)/16:C=A AND 15
170 IF D=3 THEN B=17*(A AND 240)/16:C=17*(A AND 1
5)
180 POKE 39168+T*16+I,B
190 POKE 39176+T*16+I,C
200 NEXT I
210 NEXT T
220 DATA 1,16,8,0,0,0,0,0
230 DATA 2,90,189,189,153,165,66,129,66
240 DATA 1,0,0,0,0,0,0,0
250 DATA 3,24,60,24,60,24,60,24,24
```

```

260 DATA 2,90,189,189,153,66,36,24,36
270 CLS
280 PEN 3:INK 2,18
290 X=20:VI=7:POKE 40286,VI
300 RESTORE 310
310 DATA 100,10,19,300,20,11,700,40,20,1500,80,15,1500
,80
320 SC=0
330 LOCATE 1,3:PRINT STRING$(40,140)
340 GOSUB 570
350 FOR T=0 TO 10
360 POKE 35002+T*3,250-INT(RND(1)*50)
370 NEXT T
380 POKE 39000,0:POKE 39001,10
390 POKE 39002,0:POKE 39006,1
400 READ M:READ P
410 CALL 40079
420 IF PEEK(39006)=1 THEN 460
430 SOUND 1,20,4
440 SC=SC+P:GOSUB 580
450 IF SC=M THEN VI=VI-1:POKE 40286,VI:READ A:INK
2,A:READ M:READ P
460 T=T+1:IF T MOD 2=1 THEN 500
470 IF INKEY(39)=0 AND X>1 THEN LOCATE X,23:PRINT
CHR$(32):X=X-1
480 IF INKEY(31)=0 AND X<40 THEN LOCATE X,23:PRINT
CHR$(32):X=X+1
490 LOCATE X,23:PRINT CHR$(255)
500 IF INKEY(71)=0 AND PEEK(39002)=0 THEN POKE 390
02,1:POKE 39003,X
510 IF PEEK(39000)=0 THEN 410
520 SOUND 1,2000:SOUND 1,2500:SOUND 1,1500
530 FOR I=1 TO 500:NEXT I
540 LOCATE 13,7:PEN 3
550 PRINT "APPUYER SUR 'FO'"
560 IF INKEY(15)=-1 THEN 560 ELSE 270
570 PEN 1
580 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)
590 IF SC>RE THEN RE=SC
600 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
610 RETURN
620 RESTORE 40000
630 FOR T=40000 TO 40442

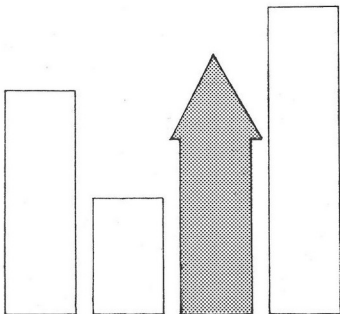
```

```

640 READ A
650 POKE T,A
660 NEXT T
670 RETURN
40000 DATA 22,153,205,73,156,205,73,156,201,229,1,8,8,
26,119,28,124,254,248,56
40020 DATA 9,213,17,176,63,183,237,82,209,124,129,103,
16,235,225,35,201,123,230,7
40040 DATA 245,123,203,63,203,63,203,63,95,6,192,74,10
7,38,0,203,37,203,37,133
40060 DATA 111,203,37,41,41,41,9,193,120,183,40,6,17,0,
8,25,16,253,201,58
40080 DATA 90,152,183,40,67,254,1,32,21,60,50,90,152,5
8,91,152,203,39,214,2
40100 DATA 50,93,152,62,168,50,92,152,24,30,237,91,92,
152,205,101,156,30,32,205
40120 DATA 64,156,58,92,152,214,4,50,92,152,254,20,32,
6,175,50,90,152,24,12
40140 DATA 237,91,92,152,205,101,156,30,48,205,64,156,
62,1,50,94,152,33,184,136
40160 DATA 34,160,140,6,11,197,42,160,140,17,136,144,1
,3,0,237,176,58,138,144
40180 DATA 254,1,32,75,237,91,136,144,205,101,156,126,
254,17,32,63,175,50,90,152
40200 DATA 237,91,136,144,205,101,156,30,32,205,64,15
6,237,95,230,64,198,170,50,138
40220 DATA 144,58,136,144,203,63,203,63,203,63,71,58,1
37,144,87,30,24,197,213,205
40240 DATA 101,156,30,32,205,64,156,209,123,198,8,95,1
93,16,238,175,50,94,152,33
40260 DATA 136,144,237,91,160,140,1,3,0,237,176,237,83
,160,140,193,16,143,58,89
40280 DATA 152,60,50,89,152,254,6,216,175,50,89,152,33
,184,136,34,160,140,6,11
40300 DATA 197,58,96,152,254,64,62,16,40,2,62,64,50,96,
152,42,160,140,17,136
40320 DATA 144,1,3,0,237,176,58,138,144,254,2,56,6,60,5
0,138,144,24,81,183
40340 DATA 32,35,60,50,138,144,62,24,50,136,144,237,95
,214,36,254,37,48,250,203
40360 DATA 39,198,4,50,137,144,237,91,136,144,205,101,
156,126,183,32,230,237,91,136
40380 DATA 144,205,101,156,30,0,205,64,156,58,136,144,

```

198,2,50,136,144,254,176,32
40400 DATA 5,62,1,50,88,152,237,91,136,144,205,101,156
,58,96,152,95,205,64,156
40420 DATA 33,136,144,237,91,160,140,1,3,0,237,176,237
,83,160,140,193,16,1,201
40440 DATA 195,108,157



ASCENSEURS

Dans ce jeu, vous devrez traverser une série d'ascenseurs. Pour vous déplacer, utilisez la touche < COPY > qui ne sera valide que si vous êtes sur un ascenseur ou un étage. Un conseil : si votre prochain seuil est un ascenseur qui descend, appuyez sur la touche < COPY > lorsque vos bras seront au même niveau que cet ascenseur. Le sens de votre déplacement change automatiquement lorsque vous parvenez à une extrémité de l'écran. Le jeu s'arrête lorsque vous touchez le haut ou le bas de l'écran.

```
10 REM*****
20 REM^ASCENSEURS^
30 REM*****
40 MODE 1
50 GOSUB 660
60 MEMORY 15000
70 BORDER 0
80 INK 0,0:INK 1,18:INK 2,2:INK 3,6
90 RESTORE 180
100 FOR T=0 TO 5
110 FOR I=0 TO 7
120 READ A
130 IF T>3 THEN B=A AND 240:C=(A AND 15)*16 ELSE
B=(A AND 240)/16:C=A AND 15
140 POKE 39168+T*16+I,B
150 POKE 39176+T*16+I,C
160 NEXT I
170 NEXT T
180 DATA 7,7,7,7,1,1,127,127
190 DATA 224,224,224,224,128,128,254,254
200 DATA 1,1,6,6,24,24,24,24
```

```

210 DATA 128,128,96,96,24,24,24,24
220 DATA 255,255,255,255,0,0,0,0
230 DATA 0,0,0,0,0,0,0,0
240 CLS:SC=0:X=0:Y=12:TE=0
250 CO=240:SE=2
260 POKE 34998,Y:POKE 34999,X
270 FOR T=0 TO 27
280 POKE 35000+T*3,5+INT(RND(1)*6)+(T MOD 2)*10
290 POKE 35001+T*3,2+INT(T/2)*2-4*(T>9)-4*(T>17)
300 POKE 35002+T*3,((T-(T MOD 2))/2) MOD 2
310 NEXT T
320 PEN 1
330 A$=CHR$(143)+CHR$(143)
340 LOCATE 1,14:PRINT A$
350 LOCATE 39,14:PRINT A$
360 LOCATE 13,14:PRINT A$;A$
370 LOCATE 25,14:PRINT A$;A$
380 GOSUB 610
390 CALL 40067
400 Y=PEEK(34998)
410 TE=PEEK(34997)
420 IF Y<4 OR Y>22 THEN 530
430 A=X:N=0
440 IF INKEY(9)=0 THEN X=X+SE:FOR T=1 TO 100-N:NE
XT T:GOTO 460
450 N=N+1:IF N<20 THEN 440
460 IF X=A THEN 390
470 IF TE=1 THEN X=A:GOTO 390
480 IF X=-2 OR X=40 THEN SE=-SE:X=A+SE
490 SC=SC+1:GOSUB 610
500 TE=1:CALL 40024
510 SOUND 1,200
520 POKE 34999,X:GOTO 390
530 FOR T=0 TO 3
540 SOUND 1,2000:CALL 40031
550 FOR I=1 TO 200:NEXT I
560 SOUND 1,1000:CALL 40024
570 FOR I=1 TO 200:NEXT I
580 NEXT T
590 LOCATE 13,5:PRINT "APPUYER SUR 'FO'"
600 IF INKEY(15)=-1 THEN 600 ELSE 240
610 PEN 3
620 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)

```



```

630 IF SC>RE THEN RE=SC
640 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
650 RETURN
660 RESTORE 730
670 FOR T=40000 TO 40247
680 READ A
690 POKE T,A
700 NEXT T
710 RETURN
720 REM^AFFICHE
730 DATA 22,153,205,73,156,205,73,156,201,229,1,8,8,26,
119,28,124,129,103,16
740 DATA 248,225,35,201
750 REM^SUITE
760 DATA 17,80,80,205,102,156,201,17,32,0,205,102,156,
201,42,182,136,205,26,188
770 DATA 213,205,64,156,205,64,156,42,182,136,45,205,2
6,188,209,90,205,64,156,205
780 DATA 64,156,201,205,95,156,33,184,136,34,160,140,6
,28,197,42,160,140,17,136
790 DATA 144,1,3,0,237,176,42,136,144,205,26,188,30,80,
205,64,156,30,80,205
800 DATA 64,156,58,138,144,254,1,40,18,58,136,144,254,
2,32,5,62,23,50,136
810 DATA 144,33,136,144,53,24,16,58,136,144,254,22,32,
5,62,1,50,136,144,33
820 DATA 136,144,52,42,136,144,205,26,188,30,64,205,64
,156,30,64,205,64,156,33
830 DATA 136,144,237,91,160,140,1,3,0,237,176,237,83,1
60,140,193,16,152,62,0
840 DATA 50,181,136,42,182,136,205,26,188,126,254,240,
32,9,33,182,136,53,205,95
850 DATA 156,24,40,42,182,136,44,205,26,188,126,254,24
0,40,28,42,182,136,44,44
860 DATA 205,26,188,126,254,240,40,5,62,1,50,181,136,2
05,88,156,33,182,136,52
870 DATA 205,95,156,201

```



FOOTBALL

Marquez un maximum de buts en évitant toute l'équipe adverse. Dix interceptions du ballon par l'ennemi et vous aurez perdu. Vous pouvez déplacer le ballon lorsque celui-ci est au repos en utilisant les touches curseur gauche et droite et tirer avec la touche < COPY > .

```
10 REM*****
20 REM^FOOTBALL^
30 REM^*****
40 MODE 1
50 GOSUB 710
60 MEMORY 15000
70 BORDER 9
80 INK 0,9:INK 1,1:INK 2,0:INK 3,26
90 RESTORE 170
100 FOR I=0 TO 4
110 FOR T=0 TO 7
120 READ A
130 POKE 39000+T*I^16,A AND 240
140 POKE 39008+T*I^16,(A AND 15)^16
150 NEXT T
160 NEXT I
170 DATA 7,7,7,7,1,1,127,127
180 DATA 224,224,224,224,128,128,254,254
190 DATA 1,1,6,6,24,24,24,24
200 DATA 128,128,96,96,24,24,24,24
210 DATA 0,0,0,0,0,0,0,0
220 FOR I=0 TO 4
230 FOR T=0 TO 10
```

```

240 READ A
250 POKE 35000+I+T^5,A
260 NEXT T
270 NEXT I
280 DATA 3,7,7,10,13,13,16,16,18,18,20
290 DATA 19,14,24,19,9,29,12,26,11,27,19
300 DATA 0,0,1,1,1,0,0,1,1,0,0
310 DATA 24,18,26,33,18,34,18,32,18,28,30
320 DATA 14,12,20,5,4,20,6,20,10,20,8
330 CLS:N=0:SC=0
340 MOVE 0,318:DRAW 639,318,3
350 MOVE 219,318
360 DRAW 219,360:DRAW 419,360:DRAW 419,318
370 MOVE 192,318
380 DRAW 172,248:DRAW 466,248:DRAW 446,318
390 MOVE 100,318
400 DRAW 40,152:DRAW 598,152:DRAW 538,318
410 GOSUB 650
420 X=20:Y=25
430 LOCATE X,Y:PRINT CHR$(231)
440 CALL 40043
450 IF INKEY(9)=0 AND Y=25 THEN LOCATE X,Y:PRINT C
HR$(32):Y=23
460 IF PEEK(48910+X^2+Y^80)<>0 THEN 590
470 IF Y<25 THEN LOCATE X,Y:PRINT CHR$(32):Y=Y-2:LO
CATE X,Y:PRINT CHR$(231)
480 IF PEEK(48910+X^2+Y^80)<>0 THEN 590
490 IF Y<>5 THEN 550
500 SOUND 1,200
510 SC=SC+1:GOSUB 650
520 LOCATE X,Y:PRINT CHR$(32)
530 X=20:Y=25
540 LOCATE X,Y:PRINT CHR$(231)
550 A=X:X=X+INKEY(1)-INKEY(8)
560 IF A=X THEN 440
570 IF Y<>25 OR X=14 OR X=27 THEN X=A:GOTO 440
580 LOCATE A,Y:PRINT CHR$(32):GOTO 430
590 N=N+1:GOSUB 650
600 SOUND 1,2000
610 LOCATE X,Y:PRINT CHR$(32)
620 IF N<>10 THEN 420
630 LOCATE 13,7:PRINT "APPUYER SUR '0'"
640 IF INKEY(15)=-1 THEN 640 ELSE 330

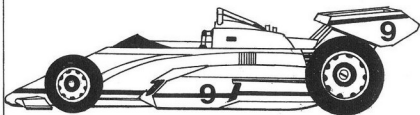
```

```

650 PEN 2
660 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)
670 IF SC>RE THEN RE=SC
680 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
690 LOCATE 20,1:PRINT RIGHT$(STR$(100+N),2)
700 RETURN
710 RESTORE 780
720 FOR T=40000 TO 40176
730 READ A
740 POKE T,A
750 NEXT T
760 RETURN
770 REM^AFFICH
780 DATA 62,0,6,8,79,229,33,88,152,87,25,235,225,229,21
3,205,96,156,209,104
790 DATA 103,25,235,225,35,229,205,96,156,225,35,201,1
20,245,26,119,19,9,241,61
800 DATA 32,247,201
810 REM^SUITE
820 DATA 33,184,136,34,160,140,6,11,197,42,160,140,17,
136,144,1,5,0,237,176
830 DATA 1,64,64,17,64,64,205,206,156,58,137,144,33,13
9,144,190,32,7,62,1
840 DATA 50,138,144,24,11,33,140,144,190,32,5,62,0,50,1
38,144,33,137,144,58
850 DATA 138,144,254,0,32,3,52,24,1,53,17,0,16,1,32,48,2
05,206,156,33
860 DATA 136,144,237,91,160,140,1,5,0,237,176,237,83,1
60,140,193,16,166,201,42
870 DATA 136,144,197,229,213,205,26,188,209,213,205,6
4,156,209,90,205,64,156,225,35
880 DATA 205,26,188,209,213,205,64,156,209,90,205,64,1
56,201

```

FI 708



FORMULE 1

Réalisez le meilleur temps à bord de votre Formule 1 en évitant les autres voitures. Si vous en heurtez une, votre vitesse deviendra nulle. Pour déplacer votre bolide, utilisez les touches curseur gauche et droite, pour accélérer servez-vous de la touche <Z> et de la touche <X> pour freiner. Bonne route !

```
10 REMAAAAAAAAA
20 REM^FORMULE 1^
30 REMAAAAAAAAA
40 MODE 1
50 GOSUB 730
60 MEMORY 15000
70 BORDER 0
80 INK 0,0:INK 1,26:INK 2,2:INK 3,9
90 RESTORE 190
100 FOR T=0 TO 9
110 FOR I=0 TO 7
120 IF I=0 AND T=4 THEN RESTORE 190
130 READ A
140 IF T<4 THEN B=A AND 240:C=(A AND 15)^16 ELSE
B=(A AND 240)/16:C=A AND 15
150 POKE 39168+T^16+I,B
160 POKE 39176+T^16+I,C
170 NEXT I
180 NEXT T
190 DATA 192,192,222,222,204,222,255,255
200 DATA 0,0,0,0,60,24,254,159
210 DATA 255,255,222,204,222,222,192,192
220 DATA 159,254,24,60,0,0,0,0
```

```

230 DATA 0,0,0,0,0,0,0
240 DATA 0,0,0,0,0,0,0
250 CLS:X=5:Y=10:VI=0
260 N=0
270 POKE 34998,Y:POKE 34999,X
280 FOR T=0 TO 4
290 POKE 35000+T^2,25
300 POKE 35001+T^2,T^8
310 NEXT T
320 PEN 3
330 LOCATE 1,1
340 FOR T=1 TO 5
350 PRINT STRING$(200,CHR$(143));
360 NEXT T
370 PEN 1
380 LOCATE 1,7
390 PRINT STRING$(40,CHR$(131));
400 FOR T=1 TO 2
410 PRINT STRING$(160,CHR$(32));
420 NEXT T
430 PRINT STRING$(40,CHR$(140))
440 GOSUB 610:K=TIME:GOSUB 640
450 CALL 40031
460 IF INKEY(71)=0 THEN VI=VI+10
470 IF INKEY(63)=0 THEN VI=VI-20
480 D=PEEK(34997)
490 IF D<4 THEN N=N+D+2 ELSE N=N+D-254
500 IF N>999 THEN 680
510 IF VI>360 THEN VI=360
520 IF VI<0 THEN VI=0
530 GOSUB 640
540 SOUND 1,4000-9*VI:IF VI<144 THEN POKE 34997,2
54+INT(VI/72) ELSE POKE 34997,INT(VI/72)-2
550 CALL 40067
560 IF PEEK(34995)=1 THEN SOUND 129,1:SOUND 2,200:
VI=0
570 A=Y:Y=Y+INKEY(1)-INKEY(8)
580 IF A=Y THEN 460
590 IF Y=6 OR Y=14 THEN Y=A:GOTO 460
600 CALL 40024:POKE 34998,Y:GOTO 450
610 IF SC<RE OR RE=0 THEN RE=SC
620 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
630 RETURN

```



```

640 SC=INT((TIME-K)/10)
650 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)
660 LOCATE 19,20:PRINT RIGHT$(STR$(10000+VI),4)
670 RETURN
680 SOUND 129,1
690 SOUND 1,200:SOUND 1,100:SOUND 1,150
700 GOSUB 610
710 LOCATE 13,5:PRINT "APPUYER SUR 'FO'"
720 IF INKEY(15)=-1 THEN 720 ELSE 250
730 RESTORE 800
740 FOR T=40000 TO 40284
750 READ A
760 POKE T,A
770 NEXT T
780 RETURN
790 REM^AFFICHE
800 DATA 22,153,205,73,156,205,73,156,201,229,1,8,8,26,
119,28,124,129,103,16
810 DATA 248,225,35,201
820 REM^SUITE
830 DATA 17,128,128,205,102,156,201, 17,0,32,205,102,1
56,201,42,182,136,205, 26,188
840 DATA 213,205,64,156,205,64,156,42,182,136,44,205,2
6,188,209,90,205,64,156,205
850 DATA 64,156,201,62,0,50,179,136,33,184,136,34,160,
140,6,5,197,42,160,140
860 DATA 17,136,144,1,2,0,237,176,58,136,144,254,25,32,
5,205,43,157,24,68
870 DATA 42,136,144,205,26,188,30,128,205,64,156,205,6
4,156,42,136,144,44,205,26
880 DATA 188,30,128,205,64,156,205,64,156,205,43,157,5
8,136,144,254,25,40,29,42
890 DATA 136,144,205,26,188,30,64,205,64,156,205,64,15
6,42,136,144,44,205,26,188
900 DATA 30,96,205,64,156,205,64,156,33,136,144,237,91
,160,140,1,2,0,237,176
910 DATA 237,83,160,140,193,16,145,42,182,136,205,26,1
88,62,0,190,40,24,17,2
920 DATA 56,25,190,40,17,42,182,136,44,205,26,188,190,
40,7,17,2,0,25,190
930 DATA 32,5,62,1,50,179,136,205,95,156,201,58,137,14
4,33,181,136,150,50,137
940 DATA 144,254,39,56,35,254,50,48,12,214,38,50,137,1

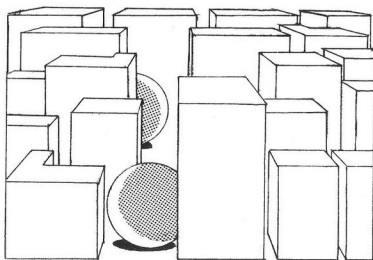
```

44,62,25,50,136,144,24

950 DATA 19,198,38,50,137,144,237,95,230,7,254,7,32,1,6

1,198,7,50,136,144

960 DATA 201



PIÈGE

Vous vous êtes laissé prendre au piège ; enfermé dans l'une des pièces les plus sombres du château, vous devez éviter les êtres supranaturels qui la hantent et qui apparaissent de plus en plus nombreux sur l'écran. Pour vous déplacer, utilisez les touches curseur gauche et droite.

```
10 REM*****
20 REM^PIEGE^
30 REM*****
40 MODE 1
50 GOSUB 690
60 RESTORE 220
70 FOR I=0 TO 1
80 FOR T=0 TO 7
90 READ A
100 POKE 39000+T*I^16,A AND 240
110 POKE 39008+T*I^16,(A AND 15)^16
120 NEXT T
130 NEXT I
140 NB=15:POKE 34999,NB
150 CLS
160 FOR T=0 TO NB-1
170 POKE 35000+T^4,INT(RND(1)^15)+5
180 POKE 35001+T^4,2
190 POKE 35002+T^4,2+INT(255/NB)^T
200 POKE 35003+T^4,2+INT(RND(1)^2)
210 NEXT T
```

```

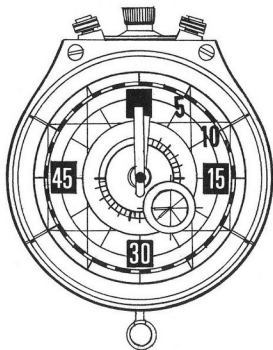
220 DATA 24,60,102,195,195,102,60,24
230 DATA 0,0,0,0,0,0,0
240 MEMORY 15000
250 BORDER 0
260 INK 0,0:INK 1,24:INK 2,18:INK 3,6
270 SC=0:X=20:N=0:K=TIME
280 GOSUB 630
290 PEN 3
300 LOCATE 2,3
310 FOR T=1 TO 38
320 PRINT CHR$(143);
330 NEXT T
340 FOR T=4 TO 23
350 LOCATE 2,T:PRINT CHR$(143)
360 LOCATE 39,T:PRINT CHR$(143)
370 NEXT T
380 LOCATE 2,23
390 FOR T=1 TO 38
400 PRINT CHR$(143);
410 NEXT T
420 PEN 2:K=TIME
430 LOCATE X,13:PRINT CHR$(249)
440 SC=SC+1:GOSUB 640
450 CALL 40043
460 IF PEEK(50110+X^2)<>3 THEN 530
470 A=X:X=X+INKEY(1)-INKEY(8)
480 IF A=X THEN 440
490 IF X=2 OR X=39 THEN X=A:GOTO 440
500 LOCATE A,13:PRINT CHR$(32)
510 IF PEEK(50110+X^2)<>0 THEN 530
520 GOTO 430
530 FOR T=1 TO 4
540 SOUND 1,2000
550 LOCATE X,13:PRINT CHR$(249)
560 FOR I=1 TO 200:NEXT I
570 SOUND 1,1000
580 LOCATE X,13:PRINT CHR$(32)
590 FOR I=1 TO 200:NEXT I
600 NEXT T
610 LOCATE 13,7:PRINT "APPUYER SUR 'F0'"
620 IF INKEY(15)=-1 THEN 620 ELSE 150
630 PEN 2
640 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)

```

```

650 IF TIME-K>200 THEN K=TIME:SOUND 1,200
660 IF SC>RE THEN RE=SC
670 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
680 RETURN
690 RESTORE 760
700 FOR T=40000 TO 40230
710 READ A
720 POKE T,A
730 NEXT T
740 RETURN
750 REM^AFFICH
760 DATA 62,0,6,8,79,229,33,88,152,87,25,235,225,229,21
3,205,96,156,209,104
770 DATA 103,25,235,225,35,229,205,96,156,225,35,201,1
20,245,26,119,19,9,241,61
780 DATA 32,247,201
790 REM^SUITE
800 DATA 33,184,136,34,160,140,58,183,136,71,197,42,16
0,140,17,136,144,1,4,0
810 DATA 237,176,58,138,144,254,2,56,7,60,50,138,144,1
95,16,157,254,0,32,17
820 DATA 60,50,138,144,42,136,144,205,26,188,30,0,205,
64,156,24,108,42,136,144
830 DATA 205,26,188,30,16,205,64,156,58,136,144,254,21
,32,9,58,139,144,60,50
840 DATA 139,144,24,11,254,3,32,7,58,139,144,61,50,139,
144,58,137,144,254,2
850 DATA 32,10,58,139,144,222,2,50,139,144,24,12,254,3
7,32,8,58,139,144,198
860 DATA 2,50,139,144,42,136,144,58,139,144,254,0,32,2,
36,44,254,2,32,2
870 DATA 37,44,254,1,32,2,36,45,254,3,32,2,37,45,34,136,
144,205,26,188
880 DATA 30,0,205,64,156,33,136,144,237,91,160,140,1,4,
0,237,176,237,83,160
890 DATA 140,193,16,1,201,195,117,156

```



SLALOM

Réalisez le meilleur temps dans cette descente à ski. Chaque porte ratée entraînera une pénalité de temps. De même, changer de sens vous fera perdre du temps. Pour manœuvrer votre skieur, utilisez les touches curseur gauche et droite.

```
10 REM*****
20 REM^SLALOM^
30 REM*****
40 MODE 1
50 GOSUB 670
60 MEMORY 15000
70 BORDER 0
80 INK 0,26:INK 1,2:INK 2,6:INK 3,0
90 PEN 3
100 RESTORE 190
110 FOR T=0 TO 11
120 FOR I=0 TO 7
130 READ A
140 IF T>7 THEN B=A AND 240:C=(A AND 15)*16 ELSE
B=(A AND 240)/16:C=A AND 15
150 POKE 39168+T*16+I,B
160 POKE 39176+T*16+I,C
170 NEXT I
180 NEXT T
190 DATA 1,1,1,0,3,5,5,9
200 DATA 192,192,192,128,224,208,208,200
210 DATA 7,99,26,6,1,0,0,0
220 DATA 32,160,96,25,134,100,24,0
```

```

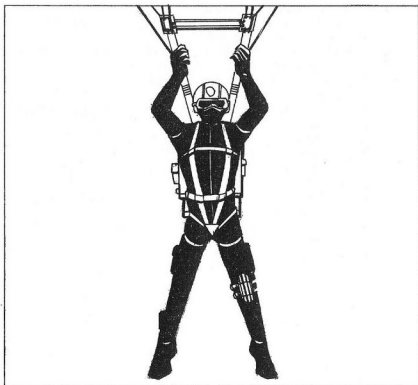
230 DATA 3,3,3,1,7,11,11,19
240 DATA 128,128,128,0,192,160,160,144
250 DATA 4,5,6,24,97,38,24,0
260 DATA 224,198,88,96,128,0,0,0
270 DATA 16,24,28,30,31,16,16,16
280 DATA 16,16,16,16,16,16,56,16
290 DATA 0,0,0,0,0,0,0,0
300 DATA 0,0,0,0,0,0,0,0
310 CLS:X=20:Y=7:SE=-1
320 RESTORE 410
330 POKE 40086,4
340 POKE 34998,Y:POKE 34999,X
350 FOR I=0 TO 3
360 FOR T=0 TO 1
370 READ A
380 POKE 35000+I*2+T,A
390 NEXT T
400 NEXT I
410 DATA 8,17,8,24,18,10,18,17
420 K=TIME:GOSUB 640:SC=0:GOSUB 620
430 CALL 40038:CALL 40074
440 IF INKEY(8)=-1 AND INKEY(1)=-1 THEN 440
450 K=TIME:N=0
460 GOSUB 610
470 IF SE=0 THEN 540
480 CALL 40074:N=N+1:IF N=402 THEN 570
490 IF N=390 THEN POKE 40086,2:LOCATE PEEK(35005)
+1,PEEK(35004)+1:PRINT CHR$(32):LOCATE PEEK(35007)+1
,PEEK(35006)+1:PRINT CHR$(32)
500 IF PEEK(34996)=1 THEN SOUND 1,2000:K=K-50
510 CALL 40024:X=X+SE:POKE 34999,X
520 IF SE=1 THEN CALL 40031 ELSE CALL 40038
530 IF X=1 OR X=37 THEN SE=0
540 IF INKEY(8)=0 AND X>1 THEN SE=-1:K=K-10
550 IF INKEY(1)=0 AND X<37 THEN SE=1:K=K-10
560 GOTO 460
570 SOUND 1,200:SOUND 1,100:SOUND 1,150
580 GOSUB 640
590 LOCATE 13,3:PRINT "APPUYER SUR 'F0'"
600 IF INKEY(15)=-1 THEN 600 ELSE 310
610 SC=INT((TIME-K)/10)
620 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)
630 RETURN

```

```

640 IF SC<RE OR RE=0 THEN RE=SC
650 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
660 RETURN
670 RESTORE 740
680 FOR T=40000 TO 40227
690 READ A
700 POKE T,A
710 NEXT T
720 RETURN
730 REM^AFFICHE
740 DATA 22,153,205,73,156,205,73,156,201,229,1,8,8,26,
119,28,124,129,103,16
750 DATA 248,225,35,201
760 REM^SUITE
770 DATA 17,182,192,205,109,156,201,17,32,0,205,109,15
6,201,17,96,64,205,109,156
780 DATA 201,42,182,136,205,26,188,213,205,64,156,205,
64,156,42,182,136,45,205,26
790 DATA 188,209,90,205,64,156,205,64,156,201,62,0,50,
180,136,33,184,136,34,160
800 DATA 140,6,4,197,42,160,140,17,136,144,1,2,0,237,17
6,42,136,144,205,26
810 DATA 188,30,192,205,64,156,58,136,144,254,3,32,38,
62,24,50,136,144,193,197
820 DATA 120,230,1,254,0,32,14,237,95,230,15,198,9,50,1
37,144,50,181,136,24
830 DATA 45,58,181,136,198,7,50,137,144,24,35,254,7,32,
31,193,197,33,183,136
840 DATA 120,230,1,254,0,32,8,58,137,144,190,56,13,24,6
,58,137,144,190,48
850 DATA 5,62,1,50,180,136,33,136,144,53,42,136,144,20
5,26,188,30,128,205,64
860 DATA 156,33,136,144,237,91,160,140,1,2,0,237,176,2
37,83,160,140,193,16,1
870 DATA 201,195,151,156

```



S.O.S.

Récupérez les parachutistes à l'aide de votre barque, mais attention, celle-ci est entraînée vers la berge. Le deuxième chiffre en partant de la gauche est le nombre de parachutistes que contient votre barque. Si celui-ci est à 5, le bord de l'écran devient vert et vous ne pouvez plus récupérer de parachutiste jusqu'à ce que vous ayez vidé votre barque en regagnant la berge. Elle peut être déplacée à l'aide des touches "<" et ">". Le jeu s'arrête si vous manquez dix parachutistes.

```
10 REM*****
20 REM*S.O.S*
30 REM*****
40 MODE 1:MEMORY 15000
50 GOSUB 780
60 BORDER 0
70 INK 0,0:INK 1,26:INK 2,1:INK 3,9
80 RESTORE 210
90 FOR T=0 TO 6
100 D=1
110 IF T>4 THEN D=3
120 FOR I=0 TO 7
130 READ A
140 IF D=1 THEN B=A AND 240:C=(A AND 15)*16
150 IF D=2 THEN B=(A AND 240)/16:C=A AND 15
160 IF D=3 THEN B=17*(A AND 240)/16:C=17*(A AND 15)
170 POKE 39168+T*16+I,B
180 POKE 39176+T*16+I,C
190 NEXT I
200 NEXT T
210 DATA 0,0,0,1,7,15,31,21
```

```

220 DATA 0,0,0,240,252,254,255,85
230 DATA 8,4,2,1,0,0,0
240 DATA 2,4,8,80,224,64,160,0
250 DATA 0,0,0,0,0,0,0
260 DATA 0,0,0,0,224,127,63,0
270 DATA 0,0,0,0,7,254,252,0
280 CLS
290 SC=0
300 PEN 2
310 LOCATE 1,23
320 FOR T=1 TO 120
330 PRINT CHR$(143);NEXT T
340 PEN 3
350 FOR I=22 TO 25
360 FOR T=35 TO 40
370 LOCATE T,I:PRINT CHR$(143);
380 NEXT T
390 NEXT I
400 N=0:M=0:GOSUB 700:X=64
410 GOSUB 740
420 E=33
430 FOR T=0 TO 19
440 POKE 35002+T*3,254-T
450 NEXT T
460 POKE 39007,E:Z=0
470 POKE 39003,X
480 IF INKEY(39)=0 AND X>4 THEN CALL 40002:X=X-3
490 IF INKEY(31)=0 AND X<63 THEN CALL 40002:X=X+2
500 IF X<64 THEN CALL 40002:X=X+1
510 POKE 39003,X:CALL 40004
520 IF X=64 AND N<>0 THEN N=0:SOUND 1,20:GOSUB 72
0:BORDER 0
530 CALL 40000
540 IF PEEK(39005)=0 THEN 470
550 Z=Z+1
560 Y=PEEK(39006)
570 IF ABS(Y-X-1)<4 AND N<>5 THEN SOUND 1,200:SC=S
C+1:N=N+1:GOSUB 700:GOSUB 680:IF Z=20 THEN 670 EL
SE 470
580 SOUND 1,2000
590 M=M+1:GOSUB 720
600 IF M<10 THEN IF Z=20 THEN 670 ELSE 470
610 SOUND 1,150:SOUND 1,50:SOUND 1,100

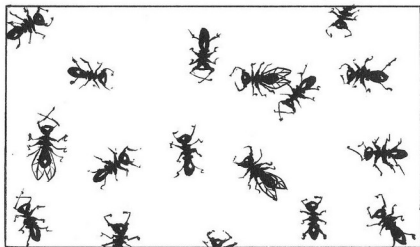
```

```

620 GOSUB 740
630 FOR T=1 TO 100:NEXT T
640 PEN 1
650 LOCATE 13,7:PRINT "APPUYER SUR 'FO'"
660 IF INKEY(15)=-1 THEN 660 ELSE 280
670 IF E>17 THEN E=E-8
675 GOTO 430
680 IF N=5 THEN BORDER 9
690 RETURN
700 PEN 2
710 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)
720 LOCATE 18,1:PRINT N;" ";M
730 RETURN
740 PEN 2
750 IF SC>RE THEN RE=SC
760 LOCATE 27,1:PRINT RIGHT$(STR$(10000+RE),4)
770 RETURN
780 RESTORE 840
790 FOR T=40000 TO 40423
800 READ A
810 POKE T,A
820 NEXT T
830 RETURN
840 DATA 24,119,24,18,58,91,152,87,30,168,205,143,156,
30,80,205,106,156,205,106
850 DATA 156,201,58,91,152,87,30,168,205,143,156,30,64
,205,106,156,30,64,205,106
860 DATA 156,201,22,153,205,115,156,205,115,156,201,2
29,1,8,8,26,119,28,124,254
870 DATA 248,56,9,213,17,176,63,183,237,82,209,124,129
,103,16,235,225,35,201,123
880 DATA 230,7,245,123,203,63,203,63,203,63,95,6,192,7
4,107,38,0,203,37,203
890 DATA 37,133,111,203,37,41,41,41,9,193,120,183,40,6,
17,0,8,25,16,253
900 DATA 201,175,50,93,152,58,92,152,60,50,92,152,230,
1,183,32,5,17,0,15
910 DATA 24,3,17,15,0,33,144,254,6,34,122,119,35,123,11
9,35,16,248,62,1
920 DATA 50,90,152,58,89,152,60,50,89,152,87,58,95,152,
186,48,7,175,50,89
930 DATA 152,50,90,152,33,184,136,34,160,140,6,20,197,
42,160,140,17,136,144,1

```

940 DATA 3,0,237,176,58,138,144,254,3,56,19,58,90,152,1
 83,32,7,58,138,144
 950 DATA 60,50,138,144,205,225,157,195,202,157,183,32,
 20,60,50,138,144,237,95,230
 960 DATA 31,198,16,50,137,144,62,16,50,136,144,24,66,2
 54,2,32,6,205,225,157
 970 DATA 195,202,157,237,91,136,144,205,143,156,229,3
 0,64,205,106,156,30,64,205,106
 980 DATA 156,225,17,80,0,25,30,64,205,106,156,30,64,20
 5,106,156,58,136,144,254
 990 DATA 158,32,16,62,2,50,138,144,50,93,152,58,137,14
 4,50,94,152,24,75,237
 1000 DATA 95,50,88,152,230,1,183,40,7,58,137,144,254,5,
 48,4,33,137,144,52
 1010 DATA 58,88,152,254,64,56,7,58,137,144,254,64,56,4,
 33,137,144,53,58,136
 1020 DATA 144,60,50,136,144,237,91,136,144,205,143,15
 6,229,30,0,205,106,156,205,106
 1030 DATA 156,225,17,80,0,25,30,32,205,106,156,205,106
 ,156,33,136,144,237,91,160
 1040 DATA 140,1,3,0,237,176,237,83,160,140,193,16,1,20
 1,195,0,157,6,255,197
 1050 DATA 193,16,252,201



XEROS

Vous devez détruire le générateur alimentant les Xéros, terribles fourmis guerrières, en un temps le plus faible possible. Pour ce faire, évitez les projectiles qu'elles envoient tout en provoquant une fissure dans le mur qui abrite le générateur. Pour déplacer votre base, utilisez les touches "<" et ">", et la touche <Z> pour tirer.

```
10 REM*****
20 REM^XEROS^
30 REM*****
40 MODE 1:BORDER 6:INK 0,6:INK 1,6
50 INK 2,21:INK 3,26
60 PEN 1:MEMORY 15000
70 FOR T=38970 TO 38975
80 POKE T,0
90 NEXT T
100 FOR T=0 TO 9
110 LOCATE 1,1:PRINT CHR$(48+T)
120 FOR I=0 TO 1
130 FOR Z=0 TO 7
140 POKE 39168+T^16+I^8+Z,PEEK(49152+Z^2048+I)
150 NEXT Z
160 NEXT I
170 NEXT T
180 CLS
190 GOSUB 950
200 BORDER 0:INK 0,0
210 RESTORE 320
220 FOR T=10 TO 14
```

```

230 FOR I=0 TO 7
240 READ A
250 B=(A AND 240)/16:C=A AND 15
260 IF T=10 OR T=12 THEN B=17^(A AND 240)/16:C=17
   *(A AND 15)
270 IF T=14 THEN B=A AND 240:C=(A AND 15)*16
280 POKE 39168+T*16+I,B
290 POKE 39176+T*16+I,C
300 NEXT I
310 NEXT T
320 DATA 0,126,126,126,0,231,231,231
330 DATA 0,0,0,0,0,0,0
340 DATA 24,60,126,126,60,126,231,195
350 DATA 90,219,60,255,255,60,219,90
360 DATA 0,0,24,60,60,24,0,0
370 CLS
380 FOR T=38990 TO 38995
390 POKE T,0
400 NEXT T
410 FOR T=39000 TO 39080
420 POKE T,1
430 NEXT T
440 POKE 39039,0:POKE 39080,0
450 POKE 38996,24:POKE 38997,20
460 POKE 38986,2
470 FOR T=0 TO 9
480 POKE 35002+T*5,250-T*5
490 NEXT T
500 LOCATE 1,25:PRINT STRING$(4,233)
510 LOCATE 37,25:PRINT STRING$(4,233);
520 LOCATE 1,3:PRINT STRING$(40,143)
530 FOR T=0 TO 11
540 FOR I=0 TO 7
550 POKE 49346+T*I*2048,RND(1)*255
560 NEXT I
570 NEXT T
580 GOSUB 900
590 CALL 40046
600 IF PEEK(38984)=0 THEN 720
610 FOR T=1 TO 2000:NEXT T
620 FOR T=1 TO 20
630 SOUND 1,RND(1)*256,5
640 FOR I=0 TO 26

```

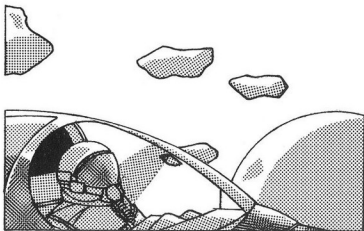
```

650 BORDER I:INK 0,I
660 NEXT I
670 NEXT T
680 BORDER 0:INK 0,0
690 LOCATE 18,3:PRINT STRING$(6,32)
700 GOSUB 820
710 GOTO 790
720 X=PEEK(38997)+1:PEN 2
730 SOUND 1,200,180,,,1
740 FOR T=1 TO 100
750 LOCATE X,25:PRINT CHR$(137)
760 LOCATE X,25:PRINT CHR$(134)
770 NEXT T
780 LOCATE X,25:PRINT CHR$(32)
790 PEN 1
800 LOCATE 13,11:PRINT"APPUYER SUR 'FO'"
810 IF INKEY(15)=-1 THEN 810 ELSE 370
820 T=0
830 IF PEEK(38970+T)<PEEK(38990+T) AND W<>0 THEN 9
00
840 IF PEEK(38970+T)>PEEK(38990+T) THEN 860
850 IF T<>4 THEN T=T+1:GOTO 830
860 FOR T=0 TO 5
870 POKE 38970+T,PEEK(38990+T)
880 NEXT T
890 W=1
900 LOCATE 25,1
910 FOR T=0 TO 5
920 PRINT CHR$(48+PEEK(38970+T));
930 NEXT T
940 RETURN
950 RESTORE 1020
960 FOR T=40000 TO 40745
970 READ A
980 POKE T,A
990 NEXT T
1000 RETURN
1010 REM^AFFICHE
1020 DATA 22,153,205,73,156,205,73,156,201,229,1,8,8,2
6,119,28,124,129,103,16
1030 DATA 248,225,35,201
1040 REM^POS
1050 DATA 203,34,6,192,74,123,107,38,0,203,37,203,37,1
33,111,203,37,41,41,41

```

1060 DATA 9,201
 1070 DATA 237,91,84,152,205,88,156,30,192,205,64,156,5
 8,86,152,60,50,86,152,254
 1080 DATA 100,56,82,175,50,86,152,17, 87,152,33,88,152
 ,1,81,0,237,176,58,87
 1090 DATA 152,50,127,152,58,128,152,50,168,152,30,4,33
 ,127,152,205,179,156,30,5
 1100 DATA 33,168,152,205,179,156,195,251,158,22,39,22
 9,213,126,183,40,10,205,88,156
 1110 DATA 30,160,205,64,156,24,8,205,88,156,30,176,205
 ,64,156,209,225,43,21,122
 1120 DATA 254,255,32,223,201,58,76,152,60,50,76,152,25
 4,5,56,68,175,50,76,152
 1130 DATA 62,39,17,4,21,205,248,156,62,31,17,35,20,205,
 248,156,24,46,205,30
 1140 DATA 187,40,40,122,50,24,157,123,50,11,157,237,91
 ,84,152,122,254,1,40,23
 1150 DATA 213,205,88,156,30,176,205,64,156,209,21,237,
 83,84,152,205,88,156,30,192
 1160 DATA 205,64,156,201,17,83,152,26,60,18,254,10,32,
 8,175,18,27,123,254,77
 1170 DATA 32,241,17,0,10,205,88,156,17,78,152,213,26,2
 03,39,203,39,203,39,203
 1180 DATA 39,95,205,64,156,209,19,123,254,84,32,235,58
 ,77,152,60,50,77,152,254
 1190 DATA 7,218,29,158,175,50,77,152,33,184,136,34,160
 ,140,6,10,197,42,160,140
 1200 DATA 17,136,144,1,5,0,237,176,58,138,144,254,3,56,
 6,60,50,138,144,24
 1210 DATA 127,183,32,37,60,50,138,144,237,95,230,3,198
 ,7,50,136,144,237,95,230
 1220 DATA 31,198,4,50,139,144,237,95,230,1,50,140,144,
 183,40,2,62,39,50,137
 1230 DATA 144,237,91,136,144,205,88,156,30,176,205,64,
 156,33,139,144,58,137,144,190
 1240 DATA 32,5,62,3,50,140,144,58,140,144,183,32,9,58,1
 37,144,60,50,137,144
 1250 DATA 24,34,254,1,32,9,58,137,144,61,50,137,144,24,
 21,58,136,144,254,24
 1260 DATA 32,7,62,245,50,138,144,24,19,58,136,144,60,5
 0,136,144,237,91,136,144
 1270 DATA 205,88,156,30,208,205,64,156,33,136,144,237,
 91,160,140,1,5,0,237,176

1280 DATA 237,83,160,140,193,5,40,3,195,110,157,237,91
 ,84,152,205,88,156,126,254
 1290 DATA 5,32,5,175,50,72,152,201,58,73,152,60,50,73,1
 52,254,6,218,251,158
 1300 DATA 175,50,73,152,58,74,152,254,2,32,3,195,219,1
 58,237,91,74,152,205,88
 1310 DATA 156,30,176,205,64,156,58,74,152,61,50,74,152
 ,237,91,74,152,205,88,156
 1320 DATA 126,254,5,32,16,205,25,159,30,176,205,64,156
 ,62,2,50,74,152,195,251
 1330 DATA 158,58,74,152,254,2,32,18,205,25,159,58,75,1
 52,254,17,56,83,254,23
 1340 DATA 48,79,50,72,152,201,237,91,74,152,205,88,156
 ,30,224,205,64,156,58,74
 1350 DATA 152,254,5,32,26,58,75,152,198,129,50,175,158
 ,50,182,158,58,129,152,183
 1360 DATA 40,71,175,50,129,152,205,4, 159,24,62,254,4,
 32,58,58,75,152,198,88
 1370 DATA 50,205,158,50,212,158,58,88,152,183,40,41,17
 5,50,88,152,205,4,159,24
 1380 DATA 32,62,71,205,30,187,40,25,58,84,152,61,50,74,
 152,58,85,152,50,75
 1390 DATA 152,237,91,74,152,205,88,156,30,224,205,64,1
 56,62,67,205,30,187,192,195
 1400 DATA 110,156,33,16,159,205,170,188,62,2,50,74,152
 ,201,1,0,0,30,0,0
 1410 DATA 15,5,0,33,32,159,205,170,188,201,1,0,0,250,0,
 0,12,10,0,32



POSTE AVANCÉ

Une nouvelle mission vous a été confiée : détruire toutes les réserves de fuel que vous rencontrerez afin d'empêcher l'ennemi de parvenir sur votre territoire. La zone que vous devrez couvrir est divisée en dix secteurs. Sur votre route, vous rencontrerez également des laboratoires de recherche que vous pourrez détruire mais, surtout, ne tirez pas sur les bases qui répliquent irrémédiablement. Pour manœuvrer votre vaisseau, utilisez les touches "<" et ">", et la touche "<Z>" pour tirer. Si vous parvenez à la fin du dixième secteur, un bonus vous sera accordé suivant le pourcentage de réserves de fuel détruites.

```
10 REM*****
20 REM*POSTE AVANCE*
30 REM*****
40 GOSUB 1190
50 MODE 1:PEN 1:MEMORY 15000
60 BORDER 24:INK 0,24:INK 1,24
70 FOR T=0 TO 9
80 LOCATE 1,1:PRINT CHR$(48+T)
90 FOR I=0 TO 1
100 FOR J=0 TO 7
110 POKE 39168+T*16+I*8+J,PEEK(49152+J*2048+I)
120 NEXT J
130 NEXT I
140 NEXT T
150 CLS
160 BORDER 0:INK 0,0
170 FOR T=39000 TO 39005
180 POKE T,0
190 NEXT T
200 INK 2,2:INK 3,6
210 RESTORE 390
```

```

220 FOR T=10 TO 19
230 IF T<17 THEN READ D
240 FOR I=0 TO 7
250 IF T>16 THEN READ D
260 READ A
270 IF D=1 THEN B=A AND 240:C=(A AND 15)*16
280 IF D=2 THEN B=(A AND 240)/16:C=A AND 15
290 IF D=3 THEN B=17*(A AND 240)/16:C=17*(A AND 1
5)
300 POKE 39168+T*16+I,B
310 POKE 39176+T*16+I,C
320 NEXT I
330 NEXT T
340 CALL 40052:PEN 3
350 LOCATE 11,8:PRINT ":FUEL.....20 PTS"
360 LOCATE 11,10:PRINT ":LABORATOIRE...50 PTS"
370 LOCATE 11,12:PRINT ":BASE"
380 FOR I=1 TO 10000:NEXT I
390 DATA 2,192,192,240,243,255,255,255,63
400 DATA 2,0,0,0,192,240,252,255,252
410 DATA 2,0,0,0,0,0,0,0,0
420 DATA 2,0,0,240,96,240,240,240,96
430 DATA 3,255,255,0,0,0,0,0,0
440 DATA 1,0,0,0,0,0,0,0,0
450 DATA 1,24,60,126,126,60,126,231,195
460 DATA 1,24,2,126,2,231,2,195,2,102,1,60,1,36,1,219
470 DATA 1,24,2,255,1,255,2,126,1,24,2,24,1,60,2,60
480 DATA 1,255,2,255,3,255,1,255,2,255,3,255,1,255,2,25
5
490 CLS:PEN 1
500 FOR T=39026 TO 39029
510 POKE T,0
520 NEXT T
530 FOR T=38990 TO 38995
540 POKE T,0
550 NEXT T
560 LOCATE 6,3:PRINT STRING$(30,210)
570 LOCATE 6,24:PRINT STRING$(30,208)
580 FOR T=4 TO 23
590 LOCATE 5,T:PRINT CHR$(209)
600 LOCATE 36,T:PRINT CHR$(211)
610 NEXT T
620 POKE 39008,10:POKE 39009,8

```

```

630 POKE 39010,23:POKE 39021,0
640 FOR T=0 TO 28
650 POKE 35001+T^2,10+T^2
660 POKE 35000+T^2,21+T MOD 2
670 NEXT T
680 POKE 39014,22
690 POKE 39016,0:POKE 39019,23
700 FOR T=0 TO 9
710 POKE 35100+T^3,24
720 NEXT T
730 LOCATE 11,1:PRINT "000000"
740 LOCATE 25,1
750 FOR T=0 TO 5
760 PRINT CHR$(PEEK(39000+T)+48);
770 NEXT T
780 LOCATE 7,25:PRINT "SECTEUR: 0"
790 LOCATE 1,1
800 POKE 39024,0
810 CALL 40046
820 IF PEEK(39024)<10 THEN 940
830 A=PEEK(39026)+PEEK(39027)^256
840 B=PEEK(39028)+PEEK(39029)^256
850 C=INT(B/A^100)
860 PEN 3:LOCATE 15,5:PRINT"BONUS :";C;"%"
870 PEN 1
880 IF C=0 THEN 930
890 FOR T=0 TO C
900 CALL 40050:SOUND 1,40,5
910 FOR I=1 TO 50:NEXT I
920 NEXT T
930 GOTO 1070
940 IF PEEK(39021)=2 THEN 1010
950 FOR T=PEEK(39010)-1 TO PEEK(39008) STEP -1
960 LOCATE 10,T:PRINT CHR$(196);
970 NEXT T
980 FOR T=PEEK(39010)-1 TO PEEK(39008) STEP-1
990 LOCATE 10,T:PRINT CHR$(32);
1000 NEXT T
1010 FOR T=1 TO 80
1020 SOUND 1,INT((RND(1)^100)+130),2,,,,1
1030 FOR I=0 TO 10
1040 NEXT I
1050 CALL 40048

```

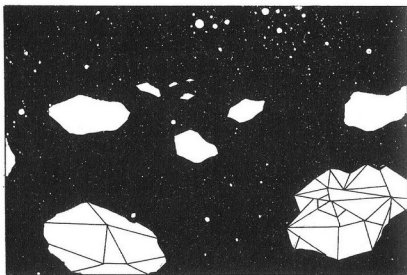
```

1060 NEXT T
1070 T=0
1080 IF PEEK(38990+T)>PEEK(39000+T) THEN 1110
1090 IF PEEK(38990+T)<PEEK(39000+T) THEN 1160
1100 T=T+1:IF T<6 THEN 1080
1110 LOCATE 25,1
1120 FOR T=0 TO 5
1130 POKE 39000+T,PEEK(38990+T)
1140 PRINT CHR$(PEEK(39000+T)+48);
1150 NEXT T
1160 PEN 3
1170 LOCATE 13,7:PRINT "APPUYER SUR 'F0'"
1180 IF INKEY(15)=-1 THEN 1180 ELSE 490
1190 RESTORE 1260
1200 FOR T=40000 TO 41005
1210 READ A
1220 POKE T,A
1230 NEXT T
1240 RETURN
1250 REM*AFFICHE
1260 DATA 22,153,205,73,156,205,73,156,201,229,1,8,8,2
6,119,28,124,129,103,16
1270 DATA 248,225,35,201
1280 REM*POS
1290 DATA 203,34,6,192,74,123,107,38,0,203,37,203,37,1
33,111,203,37,41,41,41
1300 DATA 9,201
1310 REM*SUITE
1320 DATA 24,86,24,49,24,37,17,7,9,205,88,156,17,16,154
,205,66,156,17,9
1330 DATA 9,205,88,156,17,32,154,205,66,156,17,11,9,20
5,88,156,17,0,154,205
1340 DATA 66,156,201,205,252,159,205,252,159,205,15,1
60,201,237,91,96,152,205,88,156
1350 DATA 6,4,197,229,6,8,237,95,230,1,183,40,2,237,95,
119,17,0,8,25
1360 DATA 16,240,225,35,193,16,231,201,205,203,156,24,
16,237,91,96,152,205,88,156
1370 DATA 30,160,205,64,156,205,64,156,201,22,20,33,10
0,152,205,115,157,56,63,62
1380 DATA 39,17,53,4,205,247,156,62,31,17,52,21,205,24
7,156,24,45,205,30,187
1390 DATA 40,39,122,50,8,157,123,50,31,157,58,96,152,2

```

54,4,40,24,237,91,96
 1400 DATA 152,205,88,156,30,192,205,64,156,30,192,205,
 64,156,33,96,152,53,205,203
 1410 DATA 156,201,22,8,33,101,152,205,115,157,56,108,5
 8,98,152,254,23,40,37,237
 1420 DATA 91,98,152,205,88,156,30,192,205,64,156,58,98
 ,152,60,50,98,152,254,23
 1430 DATA 40,14,237,91,98,152,205,88,156,30,208,205,64
 ,156,24,64,62,71,205,30
 1440 DATA 187,40,57,175,50,100,152,62,9,50,99,152,58,9
 6,152,60,50,98,152,24
 1450 DATA 217,52,126,186,216,175,119,201,33,129,157,2
 05,170,188,201,1,10,0,200,0
 1460 DATA 1,12,5,0,33,145,157,205,170,188,201,1,10,0,30
 ,0,0,12,10,0
 1470 DATA 22,6,33,103,152,205,115,157,48,3,195,44,159,
 22,160,33,113,152,205,115
 1480 DATA 157,56,38,58,112,152,60,50,112,152,254,10,48
 ,27,203,39,198,15,87,30
 1490 DATA 24,205,88,156,58,112,152,203,39,203,39,203,3
 9,203,39,95,22,154,205,64
 1500 DATA 156,58,104,152,60,50,104,152,230,1,183,32,36
 ,58,106,152,183,32,19,237
 1510 DATA 95,254,90,48,13,58,102,152,254,7,56,6,61,50,1
 02,152,24,11,58,102
 1520 DATA 152,254,22,48,243,60,50,102,152,33,184,136,3
 4,160,140,6,29,197,42,160
 1530 DATA 140,17,136,144,1,2,0,237,176,58,137,144,254,
 19,32,6,58,136,144,50
 1540 DATA 105,152,237,91,136,144,205,90,156,30,192,20
 5,64,156,58,137,144,61,254,10
 1550 DATA 48,8,58,102,152,50,136,144,62,67,50,137,144,
 237,91,136,144,205,90,156
 1560 DATA 30,224,205,64,156,33,136,144,237,91,160,140,
 1,2,0,237,176,237,83,160
 1570 DATA 140,193,16,169,33,28,137,34,160,140,175,50,1
 06,152,6,10,197,42,160,140
 1580 DATA 17,136,144,1,3,0,237,176,58,136,144,254,23,4
 8,14,237,91,136,144,205
 1590 DATA 90,156,30,192,205,64,156,24,82,58,106,152,18
 3,32,123,58,104,152,230,1
 1600 DATA 183,40,115,58,102,152,254,22,40,108,6,255,16
 ,254,237,95,254,12,48,98

1610 DATA 6,255,16,254,237,95,254,60,56,4,62,2,24,18,6,
 255,16,254,237,95
 1620 DATA 230,1,183,40,7,42,114,152,35,34,114,152,50,1
 38,144,62,68,50,137,144
 1630 DATA 58,102,152,61,50,136,144,33,106,152,52,58,13
 7,144,61,50,137,144,254,10
 1640 DATA 32,5,62,23,50,136,144,58,136,144,254,23,48,2
 4,237,91,136,144,205,90
 1650 DATA 156,58,138,144,203,39,203,39,203,39,203,39,9
 5,22,154,205,66,156,33,136
 1660 DATA 144,237,91,160,140,1,3,0,237,176,237,83,160,
 140,193,16,2,24,3,195
 1670 DATA 114,158,58,105,152,87,58,98,152,186,56,24,25
 4,23,40,20,237,91,98,152
 1680 DATA 205,88,156,30,192,205,64,156,62,23,50,98,152
 ,205,122,157,33,28,137,6
 1690 DATA 10,197,229,126,254,23,40,99,87,58,98,152,186
 ,32,92,35,126,254,17,56
 1700 DATA 86,254,19,48,82,35,126,183,32,21,62,1,50,109,
 152,225,229,94,35,86
 1710 DATA 205,90,156,17,0,154,205,66,156,24,56,254,1,4
 0,4,6,5,24,9,42
 1720 DATA 116,152,35,34,116,152,6,2,197,205,252,159,19
 3,16,249,205,15,160,225,229
 1730 DATA 94,35,86,205,90,156,35,30,192,205,64,156,205
 ,138,157,225,229,35,35,62
 1740 DATA 3,119,62,23,50,98,152,225,193,35,35,35,16,14
 3,62,67,205,30,187,192
 1750 DATA 58,112,152,254,10,208,58,109,152,254,1,200,2
 37,91,96,152,205,88,156,17
 1760 DATA 0,56,25,126,254,3,32,13,35,35,126,254,15,32,6
 ,35,126,254,12,40
 1770 DATA 6,62,2,50,109,152,201,195,219,156,17,82,152,
 26,60,18,254,10,32,8
 1780 DATA 175,18,27,123,254,77,32,241,201,17,0,10,205,
 88,156,17,78,152,213,26
 1790 DATA 203,39,203,39,203,39,203,39,95,205,64,156,20
 9,19,123,254,84,32,235,201



ASTÉROIDES

Vous pénétrez dans l'espace intersidéral et devez détruire les astéroïdes qui vous menacent. Pour tourner à gauche, utilisez la touche < B > , pour tourner à droite la touche < M > ; la touche < N > pour avancer, et la touche < Z > pour tirer.

```
10 REM*****
20 REM^ASTEROIDES^
30 REM*****
40 MODE 1:MEMORY 15000
50 GOSUB 670
60 BORDER 0
70 INK 0,0:INK 1,24:INK 3,6
80 RESTORE 220
90 FOR T=0 TO 10
100 D=2
110 IF T>4 THEN D=3
120 IF T>6 THEN D=1
130 FOR I=0 TO 7
140 READ A
150 IF D=1 THEN B=A AND 240:C=(A AND 15)*16
160 IF D=2 THEN B=(A AND 240)/16:C=A AND 15
170 IF D=3 THEN B=17^(A AND 240)/16:C=17*(A AND 15)
180 POKE 39168+T*16+I,B
190 POKE 39176+T*16+I,C
200 NEXT I
210 NEXT T
220 DATA 24,24,0,0,60,126,231,195
230 DATA 3,7,14,204,204,14,7,3
```

```

240 DATA 195,231,126,60,0,0,24,24
250 DATA 192,224,112,51,51,112,224,192
260 DATA 0,0,0,0,0,0,0
270 DATA 0,0,0,255,255,0,0,0
280 DATA 24,24,24,24,24,24,24,24
290 DATA 24,24,60,126,126,252,248,32
300 DATA 6,15,63,127,127,30,0,0
310 DATA 24,124,254,254,255,255,56,16
320 DATA 224,248,124,62,254,255,126,60
330 CLS:INK 2,2
340 POKE 39000,96:POKE 39001,40
350 POKE 39002,16:POKE 39016,0
360 POKE 39005,0
370 K=10:POKE 39020,K
380 SC=0:N=0
390 GOSUB 600:GOSUB 630
400 FOR T=35000 TO 35005+K*6
410 POKE T,0
420 NEXT T
430 FOR I=0 TO K-1
440 POKE 35000+I*6,200
450 POKE 35005+I*6,250-I*20
460 NEXT I
470 PEN 3
480 CALL 40079
490 IF PEEK(39016)=1 THEN SOUND 1,20,4:SC=SC+10:GO
SUB 610:GOTO 480
500 N=N+1:SOUND 1,2000,10:GOSUB 650
510 IF N<>5 THEN 480
520 INK 2,0
530 SOUND 1,400:SOUND 1,100:SOUND 1,200
540 FOR I=1 TO 300:NEXT I
550 FOR I=1 TO 100:NEXT I
560 GOSUB 630
570 PEN 3
580 LOCATE 13,7:PRINT "APPUYER SUR 'F0'"
590 IF INKEY(15)=-1 THEN 590 ELSE 330
600 PEN 3
610 LOCATE 11,1:PRINT RIGHT$(STR$(10000+SC),4)
620 RETURN
630 IF SC>RE THEN RE=SC
640 LOCATE 26,1:PRINT RIGHT$(STR$(10000+RE),4)
650 LOCATE 19,1:PRINT N

```

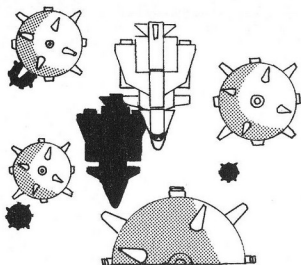
```

660 RETURN
670 RESTORE 40000
680 FOR T=40000 TO 41040
690 READ A
700 POKE T,A
710 NEXT T
720 RETURN
40000 DATA 22,153,205,73,156,205,73,156,201,229,1,8,8,
26,119,28,124,254,248,56
40020 DATA 9,213,17,176,63,183,237,82,209,124,129,103,
16,235,225,35,201,123,230,7
40040 DATA 245,123,203,63,203,63,203,63,95,6,192,74,10
7,38,0,203,37,203,37,133
40060 DATA 111,203,37,41,41,41,9,193,120,183,40,6,17,0,
8,25,16,253,201,22
40080 DATA 3,33,91,152,205,237,159,218,103,157,58,102,
152,183,40,4,61,50,102,152
40100 DATA 62,54,205,30,187,40,24,58,102,152,183,32,18
,62,10,50,102,152,58,90
40120 DATA 152,198,16,254,64,56,1,175,50,90,152,62,38,
205,30,187,40,25,58,102
40140 DATA 152,183,32,19,62,10,50,102,152,58,90,152,19
8,240,254,64,56,2,62,48
40160 DATA 50,90,152,62,46,205,30,187,40,125,6,192,17,
88,16,33,252,0,205,24
40180 DATA 157,6,78,17,89,0,33,255,16,205,24,157,6,16,1
7,88,192,33,4,32
40200 DATA 205,24,157,6,0,17,89,78,33,1,48,205,24,157,2
4,79,58,90,152,188
40220 DATA 192,120,50,78,157,122,50,74,157,123,50,71,1
57,50,83,157,50,88,157,50
40240 DATA 80,157,124,50,98,157,125,50,86,157,237,91,8
8,152,205,101,156,30,64,205
40260 DATA 64,156,58,88,152,254,16,32,5,62,186,50,88,1
52,58,88,152,198,248,50
40280 DATA 88,152,237,91,88,152,205,101,156,30,0,205,6
4,156,201,58,98,152,183,40
40300 DATA 4,61,50,98,152,218,74,158,58,93,152,183,40,
118,237,91,93,152,205,101
40320 DATA 156,30,64,205,64,156,58,97,152,183,40,18,58
,93,152,87,58,96,152,186
40340 DATA 32,24,58,99,152,50,93,152,24,16,58,94,152,8
7,58,96,152,186,32,6

```

40360 DATA 58,99,152,50,94,152,58,98,152,254,1,32,7,17
 5,50,93,152,195,74,158
 40380 DATA 58,97,152,183,40,13,58,95,152,87,58,93,152,
 130,50,93,152,24,11,58
 40400 DATA 95,152,87,58,94,152,130,50,94,152,237,91,93
 ,152,205,101,156,30,80,58
 40420 DATA 97,152,183,40,2,30,96,205,64,156,24,90,58,9
 8,152,183,32,84,62,71
 40440 DATA 205,30,187,32,8,6,3,197,193,16,252,24,69,58,
 88,152,50,93,152,58
 40460 DATA 89,152,50,94,152,38,192,17,16,1,1,0,252,205,
 244,159,38,78,17,0
 40480 DATA 0,1,16,255,205,244,159,38,16,17,192,1,1,32,4
 ,205,244,159,38,0
 40500 DATA 17,78,0,1,48,1,205,244,159,58,97,152,183,62,
 60,40,2,62,30,50
 40520 DATA 98,152,175,50,104,152,205,66,160,205,144,1
 59,22,4,33,101,152,205,237,159
 40540 DATA 218,127,159,33,184,136,34,160,140,58,108,1
 52,71,197,42,160,140,17,136,144
 40560 DATA 1,6,0,237,176,58,141,144,254,2,56,7,60,50,14
 1,144,195,108,159,183
 40580 DATA 32,127,60,50,141,144,237,95,50,100,152,254,
 64,48,29,230,1,183,40,3
 40600 DATA 175,24,2,62,78,50,137,144,58,100,152,230,31
 ,203,39,203,39,198,40,50
 40620 DATA 136,144,24,24,230,1,183,40,4,62,16,24,2,62,1
 92,50,136,144,58,100
 40640 DATA 152,230,63,198,7,50,137,144,58,100,152,230,
 3,50,138,144,58,100,152,230
 40660 DATA 15,254,7,62,255,48,2,62,1,50,140,144,58,100,
 152,230,31,254,8,48
 40680 DATA 4,62,252,24,18,254,16,48,4,62,254,24,10,254,
 24,48,4,62,2,24
 40700 DATA 2,62,4,50,139,144,195,108,159,237,91,136,14
 4,205,101,156,30,64,205,64
 40720 DATA 156,58,140,144,254,1,58,137,144,32,12,254,7
 8,32,2,62,255,60,50,137
 40740 DATA 144,24,9,183,32,2,62,79,61,50,137,144,58,13
 9,144,254,10,58,136,144
 40760 DATA 87,58,139,144,48,12,130,254,193,56,2,62,16,
 50,136,144,24,10,130,254
 40780 DATA 16,48,2,62,192,50,136,144,237,91,136,144,20

5,101,156,58,138,144,203,39
 40800 DATA 203,39,203,39,203,39,198,112,95,205,64,156,
 33,136,144,237,91,160,140,1
 40820 DATA 6,0,237,176,237,83,160,140,193,16,14,62,67,
 205,30,187,192,58,104,152
 40840 DATA 183,192,195,143,156,195,105,158,33,184,136
 ,34,160,140,58,108,152,71,197,120
 40860 DATA 50,107,152,42,160,140,17,136,144,1,6,0,237,
 176,58,141,144,254,1,40
 40880 DATA 8,6,40,197,193,16,252,24,32,6,4,197,237,91,1
 36,144,120,61,203,39
 40900 DATA 131,95,205,101,156,126,230,15,196,10,160,3
 5,126,230,15,196,10,160,193,16
 40920 DATA 226,33,136,144,237,91,160,140,1,6,0,237,176
 ,237,83,160,140,193,16,174
 40940 DATA 201,52,126,186,216,175,119,201,58,90,152,1
 85,192,120,50,95,152,123,50,96
 40960 DATA 152,122,50,97,152,124,50,99,152,201,197,12
 6,245,237,91,136,144,205,101,156
 40980 DATA 30,64,205,64,156,62,10,50,141,144,241,230,2
 40,62,2,40,18,237,91,93
 41000 DATA 152,205,101,156,30,64,205,64,156,175,50,93,
 152,62,1,50,104,152,58,107
 41020 DATA 152,50,105,152,193,201,237,91,88,152,205,1
 01,156,58,90,152,95,205,64,156
 41040 DATA 201



ARCTURUS

Sauvez l'étoile Arcturus en parvenant à la fin du sixième niveau de ce jeu. Des chenilles parcourent l'écran et se divisent en deux, produisant une bombe lorsque vous leur tirez dessus. Pour déplacer la base, utilisez les touches "<" et ">", et la touche <Z> pour tirer.

```
10 REM*****
20 REM*ARCTURUS*
30 REM*****
40 BORDER 18:INK 0,18:INK 1,18:PEN 1
50 MODE 1
60 FOR T=0 TO 9
70 LOCATE 1,1:PRINT CHR$(48+T)
80 FOR J=0 TO 1
90 FOR I=0 TO 7
100 POKE 39168+I+J*8+T*16,PEEK(49152+J+I*2048)
110 NEXT I
120 NEXT J
130 NEXT T
140 MODE 1
150 FOR T=39004 TO 39007:POKE T,0:NEXT T
160 GOSUB 1180
170 MEMORY 15000
180 BORDER 0:INK 0,0:INK 2,24:INK 3,1
190 RESTORE 300
200 FOR T=0 TO 7
210 FOR I=0 TO 7
220 READ A
230 IF T=0 THEN B=17*(A AND 240)/16:C=17*(A AND 15)
```

```

240 IF T=1 OR T=3 OR T=4 OR T=5 OR T=7 THEN B=A
    AND 240:C=(A AND 15)*16
250 IF T=2 OR T=6 THEN B=(A AND 240)/16:C=A AND 1
5
260 POKE 39328+T*16+I,B
270 POKE 39336+T*16+I,C
280 NEXT I
290 NEXT T
300 DATA 255,129,129,129,129,129,129,255
310 DATA 24,60,126,126,60,126,231,195
320 DATA 60,60,255,255,255,255,60,60
330 DATA 0,126,126,126,126,126,126,0
340 DATA 0,0,60,60,60,60,0,0
350 DATA 0,0,36,60,24,24,0,0
360 DATA 60,60,195,195,195,195,60,60
370 DATA 24,60,126,60,24,24,24,24
380 PEN 3:CLS
390 PRINT "NIVEAU? (1-4)"
400 NI=4
410 RESTORE 420
420 DATA 64,65,57,56
430 FOR T=0 TO 3
440 READ A
450 IF INKEY(A)=0 THEN NI=T
460 NEXT T
470 IF NI=4 THEN 410
480 CLS
490 FOR T=39000 TO 39003:POKE T,0:NEXT T
500 NB=2
510 GOTO 980
520 FOR T=2 TO 130 STEP 4
530 POKE 35001+T,0
540 POKE 35000+T,70
550 NEXT T
560 J=0:ZI=NI:IF NI>3 THEN ZI=3
570 POKE 37012,ZI*10
580 FOR T=0 TO 40*ZI-1 STEP 4
590 IF T MOD 40=0 THEN B=INT(RND(1)*2)^2:C=14+INT(
RND(1)*11):J=J+1:K=0
600 POKE 35000+T,3
610 POKE 35001+T,C
620 POKE 35002+T,B
630 POKE 35003+T,250-J*50-K

```



```

640 K=K+1
650 NEXT T
660 POKE 37014,NB
670 CALL 40046
680 LOCATE 2,3:PEN 1
690 FOR T=1 TO 38
700 PRINT CHR$(210);
710 NEXT T
720 LOCATE 2,25
730 FOR T=1 TO 38
740 PEN 1:IF T>2 AND T<37 THEN PEN 3
750 PRINT CHR$(208);
760 NEXT T
770 FOR T=4 TO 24
780 LOCATE 1,T:PRINT CHR$(209);
790 LOCATE 40,T:PRINT CHR$(211);
800 NEXT T
810 PEN 1
820 IF NI>3 THEN FOR T=1 TO 10:LOCATE 11,5+T:PRINT
  T CHR$(232):LOCATE 30,5+T:PRINT CHR$(232):NEXT T
830 IF NI>4 THEN FOR T=1 TO 10:LOCATE 17,10+T:PRINT
  T CHR$(232):LOCATE 24,10+T:PRINT CHR$(232):NEXT T
840 IF NI>5 THEN FOR T=1 TO 6:LOCATE 6,12+T:PRINT
  T CHR$(232):LOCATE 35,12+T:PRINT CHR$(232):NEXT T
850 CALL 40048:IF PEEK(37011)=0 THEN 980
860 SOUND 1,2000,100
870 FOR J=1 TO 3
880 FOR T=15 TO 0 STEP -1
890 BORDER T:INK 0,T
900 FOR I=1 TO 20:NEXT I
910 NEXT T
920 NEXT J
930 IF NB>1 THEN NB=NB-2:NI=NI-1:MODE 1:GOTO 510
940 GOSUB 1110:CALL 40050
950 LOCATE 13,10:PRINT "APPUYER SUR 'O'"
960 IF INKEY(15)=-1 THEN 960 ELSE MODE 1:GOTO 390
970 IF PEEK(37010)<>PEEK(37012) THEN END
980 IF NI=6 THEN SOUND 1,100:SOUND 1,200:SOUND 1,1
  50:LOCATE 8,7:PRINT "BRAVO:ARCTURUS EST SAUVEE":GO
  TO 940
990 IF NB<5 THEN NB=NB+1
1000 PEN 1
1010 NI=NI+1:MODE 1

```

```

1020 FOR T=1 TO 6
1030 SOUND 1,500,10
1040 LOCATE 16,7:PRINT "NIVEAU:";NI
1050 FOR I=1 TO 50:NEXT I
1060 SOUND 1,50,10
1070 LOCATE 16,7:PRINT STRING$(10,CHR$(32))
1080 FOR I=1 TO 50:NEXT I
1090 NEXT T
1100 MODE 1:GOTO 520
1110 T=0
1120 A=PEEK(39000+T):B=PEEK(39004+T)
1130 IF A>B THEN 1160
1140 IF A<B THEN RETURN
1150 T=T+1:IF T=4 THEN RETURN ELSE 1120
1160 FOR T=0 TO 3:POKE 39004+T,PEEK(39000+T):NEXT
T
1170 RETURN
1180 RESTORE 1250
1190 FOR T=40000 TO 40857
1200 READ A
1210 POKE T,A
1220 NEXT T
1230 RETURN
1240 REM^AFFICHE
1250 DATA 22,153,205,73,156,205,73,156,201,229,1,8,8,2
6,119,28,124,129,103,16
1260 DATA 248,225,35,201
1270 REM^POS
1280 DATA 203,34,6,192,74,123,107,38,0,203,37,203,37,1
33,111,203,37,41,41,41
1290 DATA 9,201
1300 REM^SUITE
1310 DATA 24,4,24,79,24,68,17,3,0,205,88,156,1,72,3,197,
30,160,205,64
1320 DATA 156,193,11,120,177,32,244,22,10,33,88,152,20
5,124,159,17,23,20,237,83
1330 DATA 103,152,205,67,159,33,0,0,34,108,152,34,146,
144,17,1,31,205,88,156
1340 DATA 58,150,144,245,30,176,205,64,156,35,241,61,3
2,245,22,26,33,92,152,205
1350 DATA 124,159,201,6,30,33,184,136,34,160,140,197,6
2,67,205,30,187,40,2,193
1360 DATA 201,58,146,144,33,148,144,190,32,2,193,201,5

```

8,147,144,183,40,2,193,201
 1370 DATA 193,197,42,160,140,17,136,144,1,4,0,237,176,
 58,139,144,183,40,22,60
 1380 DATA 50,139,144,183,32,22,237,91,136,144,205,88,1
 56,30,192,205,64,156,195,17
 1390 DATA 158,58,138,144,254,70,32,8,6,100,0,16,253,19
 5,17,158,237,91,136,144
 1400 DATA 205,88,156,30,160,205,64,156,58,138,144,254,
 4,56,3,195,204,157,33,138
 1410 DATA 144,126,254,2,56,9,58,137,144,254,1,32,2,53,5
 3,58,138,144,254,2
 1420 DATA 48,9,58,137,144,254,38,32,2,52,52,126,230,1,3
 2,8,58,136,144,254
 1430 DATA 3,32,1,52,126,230,1,40,8,58,136,144,254,23,32
 ,1,53,58,137,144
 1440 DATA 61,245,58,138,144,254,2,48,4,241,198,2,245,5
 8,136,144,60,245,58,138
 1450 DATA 144,230,1,32,4,241,214,2,245,241,95,241,87,2
 13,205,88,156,209,126,183
 1460 DATA 40,30,122,50,137,144,123,50,136,144,193,197,
 120,230,1,40,7,30,192,205
 1470 DATA 64,156,24,95,17,0,154,205,66,156,24,87,33,13
 8,144,52,52,58,138,144
 1480 DATA 254,4,56,75,53,53,53,53,24, 69,58,138,144,25
 4,68,56,32,58,136,144
 1490 DATA 254,23,32,7,62,70,50,138,144,24,48,60,50,136,
 144,237,91,136,144,205
 1500 DATA 88,156,30,240,205,64,156,24,30,60,50,138,144
 ,237,91,136,144,205,88,156
 1510 DATA 58,138,144,254,40,56,7,30,208,205,64,156,24,
 5,30,224,205,64,156,33
 1520 DATA 136,144,237,91,160,140,1,4,0,237,176,237,83,
 160,140,205,45,158,193,16
 1530 DATA 3,195,0,159,195,201,156,58,118,152,60,50,118
 ,152,254,8,216,205,164,158
 1540 DATA 175,50,118,152,58,108,152,183,32,36,62,71,20
 5,30,187,200,33,82,158,205
 1550 DATA 170,188,24,9,2,2,0,255,0,0,12,2,0,62,22,50,108
 ,152,58,104
 1560 DATA 152,50,109,152,24,28,237,91,108,152,205,88,1
 56,30,160,205,64,156,58,108
 1570 DATA 152,61,50,108,152,254,2,32,5,175,50,108,152,
 201,237,91,108,152,205,88

1580 DATA 156,17,16,154,205,66,156,205,164,158,201,23
 7,91,103,152,205,88,156,126,254
 1590 DATA 16,200,50,147,144,201,237,91,108,152,123,18
 3,40,25,29,205,88,156,126,183
 1600 DATA 32,17,17,80,0,25,30,160,205,64,156,175,50,10
 8,152,50,109,152,201,17
 1610 DATA 4,0,6,30,33,184,136,229,58, 108,152,190,32,3
 9,35,58,109,152,190,32
 1620 DATA 32,35,62,70,190,40,26,35,175,190,32,21,50,10
 8,152,50,109,152,43,62
 1630 DATA 3,190,56,9,62,20,119,205,80,159,225,24,4,225,
 25,16,206,201,205,149
 1640 DATA 158,58,104,152,254,3,40,23,62,39,205,30,187,
 40,7,205,50,159,21,205
 1650 DATA 67,159,237,91,103,152,122,254,36,40,14,62,31
 ,205,30,187,40,7,205,50
 1660 DATA 159,20,205,67,159,195,193,156,237,91,103,15
 2,205,88,156,30,160,205,64,156
 1670 DATA 237,91,103,152,201,237,83,103,152,205,88,15
 6,30,176,205,64,156,201,33,146
 1680 DATA 144,52,17,90,152,26,60,18,254,10,32,8,175,18,
 27,123,254,87,32,241
 1690 DATA 33,110,159,205,170,188,24,9,1,5,0,30,0,0,12,5,
 0,22,10,33
 1700 DATA 88,152,30,1,229,205,88,156,209,6,4,26,203,39,
 203,39,203,39,203,39
 1710 DATA 213,95,197,205,64,156,193,209,19,16,236,201

LA BIBLIOTHÈQUE SYBEX

OUVRAGES GÉNÉRAUX

VOTRE PREMIER ORDINATEUR *par Rodney Zaks.*

296 pages, Réf. 394

VOTRE ORDINATEUR ET VOUS *par Rodney Zaks.*

296 pages, Réf. 271

DU COMPOSANT AU SYSTÈME : une introduction aux microprocesseurs *par Rodney Zaks.*

636 pages, Réf. 0040

TECHNIQUES D'INTERFACE *aux microprocesseurs*

par Austin Lesca et Rodney Zaks.

460 pages, Réf. 0039

LEXIQUE INTERNATIONAL MICRO-ORDINATEURS, avec dictionnaire abrégé en 10 langues

192 pages, Réf. 234

GUIDE DES MICRO-ORDINATEURS A MOINS 3 000 F

par John Ponce.

144 pages, Réf. 322

LEXIQUE MICRO-INFORMATIQUE *par Pierre Le Beux.*

140 pages, Réf. 369

LA SOLUTION RS-232 *par Joe Campbell.*

208 pages, Réf. 0052

MINITEL ET MICRO-ORDINATEUR *par Pierrick Boursault.*

198 pages, Réf. 0119

ROBOTS : CONSTRUCTION, PROGRAMMATION

par Fernando Esteves.

400 pages, Réf. 0130

ALGORITHMES *par Pierre Beaufils et Wolfram Lutner.*

296 pages, Réf. 0149

BASIC

VOTRE PREMIER PROGRAMME BASIC *par Rodney Zaks.*

208 pages, Réf. 263

INTRODUCTION AU BASIC *par Pierre Le Beux.*

336 pages, Réf. 0035

LE BASIC PAR LA PRATIQUE : 60 exercices

par Jean-Pierre Laventier.

252 pages, Réf. 0095

LE BASIC POUR L'ENTREPRISE *par Xuan Tung Bui.*

204 pages, Réf. 253

PROGRAMMES EN BASIC, Mathématiques, Statistiques, Informatique *par Alan R. Miller.*

318 pages, Réf. 259

BASIC, PROGRAMMATION STRUCTURÉE

par Richard Matesian.

352 pages, Réf. 0129

JEUX D'ORDINATEUR EN BASIC *par David H. Ahl.*

192 pages, Réf. 246

NOUVEAUX JEUX D'ORDINATEUR EN BASIC

par David H. Ahl.

204 pages, Réf. 247

FICHIERS EN BASIC *par Alan Simpson.*

256 pages, Réf. 0102

TECHNIQUES DE PROGRAMMATION EN BASIC

par S. Crosmarie, M. Perron et D. Philippe.

152 pages, Réf. 0124

PASCAL

INTRODUCTION AU PASCAL *par Pierre Le Beux.*

496 pages, Réf. 0030

LE PASCAL PAR LA PRATIQUE

par Pierre Le Beux et Henri Tavernier.

562 pages, Réf. 361

LE GUIDE DU PASCAL *par Jacques Tbergmen.*

504 pages, Réf. 423

PROGRAMMES EN PASCAL pour Scientifiques et Ingénieurs *par Alan R. Miller.*

392 pages, Réf. 240

AUTRES LANGAGES

INTRODUCTION A ADA *par Pierre Le Beux.*

366 pages, Réf. 360

INTRODUCTION A C *par Bruce Hunter.*

312 pages, Réf. 0092

MICRO-ORDINATEURS

ALICE

JEUX EN BASIC POUR ALICE *par Pierre Monsaut.*

96 pages, Réf. 320

ALICE et ALICE 90, PREMIERS PROGRAMMES

par Rodney Zaks.

248 pages, Réf. 376

ALICE, 56 PROGRAMMES *par Stanley R. Trost.*

160 pages, Réf. 401

ALICE, GUIDE DE L'UTILISATEUR *par Norbert Rauloux.*

208 pages, Réf. 378

ALICE, PROGRAMMATION EN ASSEMBLEUR

par Georges Faugt-Barraly.

192 pages, Réf. 420

AMSTRAD

AMSTRAD, PREMIERS PROGRAMMES *par Rodney Zaks.*

248 pages, Réf. 0105

AMSTRAD, 56 PROGRAMMES *par Stanley R. Trost.*

160 pages, Réf. 0107

AMSTRAD, JEUX D'ACTION *par Pierre Monsaut.*

96 pages, Réf. 0108

AMSTRAD, PROGRAMMATION EN ASSEMBLEUR

par *GEORGES FAGOT-BARRALY*.

208 pages, Réf. 0136

AMSTRAD EXPLORÉ par *JOHN BRADA*.

192 pages, Réf. 0135

AMSTRAD, GUIDE DU GRAPHISME par *JAMES WYMFORD*.

208 pages, Réf. 0141

AMSTRAD CP/M 2.2 par *ANATOLE D'HARDENCOURT*.

248 pages, Réf. 0156

AMSTRAD ASTROLOGIE/NUMEROLOGIE/BIORYTHMES

par *PYENICK BOURDAULT*.

160 pages, Réf. 0167

AMSTRAD MULTIPLAN de *MICROSOFT*.

496 pages, Réf. 0111

AMSTRAD, CRÉER DE NOUVELLES INSTRUCTIONS

par *JEAN-CLAUDE DESPOND*.

144 pages, Réf. 0176

AMSTRAD ASTROCALC

par *GÉRARD BLANC ET PHILIPPE DESTREBECQ*.

168 pages, Réf. 0162

AMSTRAD, GRAPHISME EN TROIS DIMENSIONS

par *THOMAS LACHAND-ROBERT*.

240 pages, Réf. 0157

AMSTRAD, GUIDE DU BASIC ET DE L'AMSDOS

par *JEAN-LOUIS GIGECO ET MICHEL LAURENT*.

288 pages, Réf. 0159

AMSTRAD CP/M PLUS par *ANATOLE D'HARDENCOURT*.

208 pages, Réf. 0184

AMSTRAD, GAGNEZ AUX COURSES

par *JEAN-CLAUDE DESPOND*.

112 pages, Réf. 0197

AMSTRAD LOCOSCRIP par *BERNARD LE DÛ*.

144 pages, Réf. 0202

AMSTRAD, MISE AU POINT DES PROGRAMMES BASIC

par *CLAUDE VINER ET YVES JACOD*.

152 pages, Réf. 0166

APPLE / MACINTOSH

PROGRAMMEZ EN BASIC SUR APPLE II.

Tomes 1 et 2 par *LÉOPOLD LAURENT*.

208 pages, Réf. 333 et 380

APPLE II 66 PROGRAMMES BASIC par *STANLEY R. TROST*.

192 pages, Réf. 283

JEUX EN PASCAL SUR APPLE

par *DOUGLAS HERBERT ET JOSEPH T. KALASH*.

372 pages, Réf. 241

GUIDE DU BASIC APPLE II par *DOUGLAS HERBERT*.

272 pages, Réf. 0006

APPLE II, PREMIERS PROGRAMMES par *RODNEY ZAKS*.

248 pages, Réf. 373

MACINTOSH, GUIDE DE L'UTILISATEUR

par *JOSEPH CAGGIANO*.

208 pages, Réf. 396

APPLE INC, GUIDE DE L'UTILISATEUR

par *THOMAS BLACKADAR*.

160 pages, Réf. 0089

MULTIPLAN SUR MACINTOSH

par *GUYENEN HABASQUE*.

240 pages, Réf. 0099

INTRODUCTION A MAC PASCAL par *PIERRE LE BEUX*.

416 pages, Réf. 0145

MACINTOSH POUR LA PRESSE, L'ÉDITION ET

LA PUBLICITÉ par *BERNARD LE DÛ*.

160 pages, Réf. 0173

INTRODUCTION A MAC BASIC

par *PIERRE LEBEUX*.

440 pages, Réf. 0140

ATARI

JEUX EN BASIC SUR ATARI par *PAUL BUNN*.

96 pages, Réf. 282

ATARI, PREMIERS PROGRAMMES par *RODNEY ZAKS*.

248 pages, Réf. 387

ATARI, GUIDE DE L'UTILISATEUR par *THOMAS BLACKADAR*.

192 pages, Réf. 354

ATMOS

JEUX EN BASIC SUR ATMOS par *PIERRE MONSAUT*.

96 pages, Réf. 346

ATMOS, 56 PROGRAMMES par *STANLEY R. TROST*.

180 pages, Réf. 372

COMMODORE 64

JEUX EN BASIC SUR COMMODORE 64

par *PIERRE MONSAUT*.

96 pages, Réf. 0017

COMMODORE 64, PREMIERS PROGRAMMES

par *RODNEY ZAKS*.

248 pages, Réf. 342

GUIDE DU BASIC VIC 20, COMMODORE 64

par *DOUGLAS HERBERT*.

240 pages, Réf. 312

COMMODORE 64, GUIDE DE L'UTILISATEUR

par *J. KASCHER*.

144 pages, Réf. 314

COMMODORE 64, 66 PROGRAMMES

par *STANLEY R. TROST*.

192 pages, Réf. 319

COMMODORE 64, GUIDE DU GRAPHISME

par *CHARLES PLATT*.

372 pages, Réf. 0053

COMMODORE 64, JEUX D'ACTION par *ERIC RAVIS*.

96 pages, Réf. 403

COMMODORE 64, 1^{re} CONTACTS

par *MARTY DELONGHI ET CAROLINE EAPHAUT*.

208 pages, Réf. 390

COMMODORE 64, BASIC APPROFONDI

par GARY LUPPAM.
216 pages, Réf. 0100

DRAGON

JEUX EN BASIC SUR DRAGON par PIERRE MONSAUT.
96 pages, Réf. 324

EXL 100

EXL 100, JEUX D'ACTION par PIERRE MONSAUT.
96 pages, Réf. 0126

GOUPIL

PROGRAMMEZ VOS JEUX SUR GOUPIL

par FRANCIS ABELLA.
208 pages, Réf. 264

HECTOR

HECTOR JEUX D'ACTION par PIERRE MONSAUT.
96 pages, Réf. 388

IBM

IBM PC EXERCICES EN BASIC par JEAN-PIERRE LAMOTIER.
256 pages, Réf. 338

IBM PC GUIDE DE L'UTILISATEUR

par JOAN LASSELLE ET CAROL RAMSEY.
160 pages, Réf. 301

IBM PC 66 PROGRAMMES BASIC par STANLEY R. TROST.
192 pages, Réf. 359

GRAPHIQUES SUR IBM PC par NELSON FORD.
320 pages, Réf. 357

GUIDE DE PC DOS par RICHARD A. KING.
240 pages, Réf. 0013

LASER

LASER JEUX D'ACTION par PIERRE MONSAUT.
96 pages, Réf. 371

MO 5

MO 5 JEUX D'ACTION par PIERRE MONSAUT.
96 pages, Réf. 0067

MO 5, PREMIERS PROGRAMMES par RODNEY ZAKS.
248 pages, Réf. 370

MO 5, 56 PROGRAMMES par STANLEY R. TROST.
160 pages, Réf. 375

MO 5, PROGRAMMATION EN ASSEMBLEUR

par GEORGES FAGOT-BARRALY.
192 pages, Réf. 384

MO 5, DYNAMIQUE CINÉMATIQUE, MÉTHODE POUR LA PROGRAMMATION DES JEUX par DANIEL LEBRE.
272 pages, Réf. 0118

MO 5, STATIQUE, DYNAMIQUE, ELECTRONIQUE, PROGRAMMES DE PHYSIQUE EN BASIC

par BEAUFILS, LAMARCHE ET MUGGANI.
240 pages, Réf. 0148

MO 5, PROGRAMMES D'ELECTRONIQUE EN BASIC
par BEAUFILS, DELUSSEUX, DO, ROMANACIC.
312 pages, Réf. 0143

MO 5, OPTIQUE, THERMODYNAMIQUE, CHIMIE

par P. BEAUFILS, M. LAMARCHE, Y. MUGGANI.
224 pages, Réf. 0161

MO 5, PROGRAMMES D'ELECTRONIQUE APPLIQUÉE

par PIERRE BEAUFILS ET BERNARD DESPERRIER.
192 pages, Réf. 0194

MSX

MSX, JEUX D'ACTION par PIERRE MONSAUT.
96 pages, Réf. 411

MSX, INITIATION AU BASIC par RODNEY ZAKS.
248 pages, Réf. 410

MSX, 56 PROGRAMMES par STANLEY R. TROST.
160 pages, Réf. 0109

MSX, GUIDE DU GRAPHISME par MIKE SHAW.
192 pages, Réf. 0132

MSX, PROGRAMMES EN LANGAGE MACHINE
par STEVE WEBB.
112 pages, Réf. 0153

MSX, PROGRAMMATION EN ASSEMBLEUR
par GEORGES FAGOT-BARRALY.
216 pages, Réf. 0144

MSX, GUIDE DU BASIC par MICHEL LAURENT.
264 pages, Réf. 0155

MSX, JEUX EN ASSEMBLEUR par ERIC RAVES
112 pages, Réf. 0170

MSX, ROUTINES GRAPHIQUES EN ASSEMBLEUR
par STEVE WEBB.
98 pages, Réf. 0154

MSX, TECHNIQUES DE PROGRAMMATION
DES JEUX EN ASSEMBLEUR

par GEORGES FAGOT-BARRALY.
176 pages, Réf. 0178

MSX ASTROLOGIE/NUMEROLOGIE/BIDRYTHMES

par PIERRECK BOURGAULT.
157 pages, Réf. 0168

ORIC

JEUX EN BASIC SUR ORIC par PETER SHAW.
96 pages, Réf. 278

ORIC PREMIERS PROGRAMMES par RODNEY ZAKS.
248 pages, Réf. 344

SHARP

DÉCOUVREZ LE SHARP PC-1500 ET LE TRS-80 PC-2
par MICHEL LAMARCHE.
2 tomes, Réf. 261-262

SPECTRAVIDEO

SPECTRAVIDEO, JEUX D'ACTION par PIERRE MONSAUT.
96 pages, Réf. 377

SPECTRUM

PROGRAMMEZ EN BASIC SUR SPECTRUM
par S.M. GEE.
208 pages, Réf. 252

JEUX EN BASIC SUR SPECTRUM *par PETER SHAW*,
96 pages, Réf. 276
SPECTRUM, PREMIERS PROGRAMMES *par RODNEY ZAKS*,
248 pages, Réf. 381
SPECTRUM JEUX D'ACTION *par PIERRE MONSAUT*,
96 pages, Réf. 368

TI 99/4

PROGRAMMEZ VOS JEUX SUR TI 99/4
par FRANÇOIS ABELLA,
160 pages, Réf. 303

TO 7

JEUX EN BASIC SUR TO 7 *par PIERRE MONSAUT*,
96 pages, Réf. 0026
TO 7, PREMIERS PROGRAMMES *par RODNEY ZAKS*,
248 pages, Réf. 328
TO 7, PROGRAMMATION EN ASSEMBLEUR
par GEORGES FAGOT-BARNALY,
192 pages, Réf. 350
JEUX SUR TO 7 et MO 5 *par GEORGES FAGOT-BARNALY*,
168 pages, Réf. 0134
GESTION DE FICHIERS SUR TO 7 ET MO 5
par JEAN-PIERRE LHOIR,
136 pages, Réf. 0127

TO 7, 56 PROGRAMMES *par STANLEY R. TROST*,
160 pages, Réf. 374

TO 7 / MO 5, GUIDE DU BASIC
par JEAN-LOUIS GRECH ET MICHEL LAURENT,
288 pages, Réf. 0158

TO 7 / MO 5, GUIDE DU GRAPHISME
par MICHEL LAMARCHE ET YVES MUSSIANI,
240 pages, Réf. 0172

TO 7 / MO 5 ASTROLOGIE/NUMEROLOGIE/BIDRYTHMES
par PIERRE MONSAUT,
160 pages, Réf. 0169

TO 7 / MO 5, JEUX DE RÉFLEXION
par GEORGES FAGOT-BARNALY,
176 pages, Réf. 0201

TRS-80

PROGRAMMEZ EN BASIC SUR TRS-80
par LÉOPOLD LAURENT,
2 tomes, Réf. 366-251
JEUX EN BASIC SUR TRS-80 MC-10 *par PIERRE MONSAUT*,
96 pages, Réf. 323
JEUX EN BASIC SUR TRS-80 *par CHRIS PALMER*,
96 pages, Réf. 302
JEUX EN BASIC SUR TRS-80 COULEUR
par PIERRE MONSAUT,
96 pages, Réf. 325
TRS-80 MODÈLE 100, GUIDE DE L'UTILISATEUR
par ORSON KELLOGG,
112 pages, Réf. 300

TRS-80 COULEUR, PREMIERS PROGRAMMES
par RODNEY ZAKS,
248 pages, Réf. 414

TRS-80 COULEUR, 56 PROGRAMMES
par STANLEY R. TROST,
160 pages, Réf. 413

VIC 20

PROGRAMMEZ EN BASIC SUR VIC 20
par G. O. HAMANN,
2 tomes, Réf. 329-337
JEUX EN BASIC SUR VIC 20 *par ALASTAIR GOURLAY*,
96 pages, Réf. 277
VIC 20, PREMIERS PROGRAMMES *par RODNEY ZAKS*,
248 pages, Réf. 341
VIC 20 JEUX D'ACTION *par PIERRE MONSAUT*,
96 pages, Réf. 345

VG 5000

VG 5000, JEUX D'ACTION *par PIERRE MONSAUT*,
96 pages, Réf. 422
VG 5000, 56 PROGRAMMES *par STANLEY R. TROST*,
160 pages, Réf. 0128

ZX 81

ZX 81 GUIDE DE L'UTILISATEUR *par DOUGLAS HERGENT*,
208 pages, Réf. 351
ZX 81 56 PROGRAMMES BASIC *par STANLEY R. TROST*,
192 pages, Réf. 304
GUIDE DU BASIC ZX 81 *par DOUGLAS HERGENT*,
204 pages, Réf. 285
JEUX EN BASIC SUR ZX 81 *par MARK CHARLTON*,
96 pages, Réf. 275
ZX 81 PREMIERS PROGRAMMES *par RODNEY ZAKS*,
248 pages, Réf. 343

MICROPROCESSEURS

PROGRAMMATION DU Z80 *par RODNEY ZAKS*,
618 pages, Réf. 0058
APPLICATIONS DU Z80 *par JAMES W. COFFRON*,
304 pages, Réf. 0181
PROGRAMMATION DU 6502 *par RODNEY ZAKS*,
376 pages, Réf. 0031, 2ème édition
APPLICATIONS DU 6502 *par RODNEY ZAKS*,
288 pages, Réf. 332
PROGRAMMATION DU 6800
par DANIEL JEAN DAVID ET RODNEY ZAKS,
374 pages, Réf. 327
PROGRAMMATION DU 6809
par RODNEY ZAKS ET WILLIAM LABAK,
392 pages, Réf. 0139
PROGRAMMATION DU 8086/8088
par JAMES W. COFFRON,
304 pages, Réf. 0016

MISE EN OEUVRE DU 68000 *par C. VIEILLEFOND*,
352 pages, Réf. 0133

ASSEMBLEUR DU 8086/8088 *par FRANÇOIS RETOREAU*,
616 pages, Réf. 0093

SYSTÈMES D'EXPLOITATION

GUIDE DU CP/M AVEC MP/M *par RODNAY ZAKS*,
354 pages, Réf. 336

CP/M APPROFONDI *par ALAN R. MILLER*,
380 pages, Réf. 334

INTRODUCTION AU p-SYSTEM UCSD
par CHARLES W. GRANT ET JON BUTAH,
308 pages, Réf. 365

GUIDE DE MS-DOS *par RICHARD A. KING*,
360 pages, Réf. 0117

INTRODUCTION A UNIX *par JOHN D. HALAMKA*,
240 pages, Réf. 0098

GUIDE DE PRODOS
par PIERRE BEAUFILS ET WOLFRAM LUTHER,
248 pages, Réf. 0146

APPLICATIONS ET LOGICIELS

INTRODUCTION AU TRAITEMENT DE TEXTE
par HAL GLATZER,
228 pages, Réf. 243

INTRODUCTION A WORDSTAR *par ARTHUR NAMAN*,
200 pages, Réf. 0062

WORDSTAR APPLICATIONS *par JULIE ANNE ARCA*,
320 pages, Réf. 0005

VISICALC APPLICATIONS *par STANLEY R. TROST*,
304 pages, Réf. 258

VISICALC POUR L'ENTREPRISE *par DOMINIQUE HELLE*,
304 pages, Réf. 309

INTRODUCTION A dBASE II *par ALAN SIMPSON*,
280 pages, Réf. 0064

DE VISICALC A VISI ON *par JACQUES BOURDEU*,
256 pages, Réf. 321

MULTIPLAN POUR L'ENTREPRISE
par D. HELLE ET G. BOUSSAND,
304 pages, Réf. 0079

dBASE II APPLICATIONS *par CHRISTOPHE STEHLY*,
248 pages, Réf. 416

INTRODUCTION A LOTUS 1-2-3
par CHRIS GILBERT ET LAURIE WILLIAMS,
272 pages, Réf. 0106

INTRODUCTION A dBASE III *par ALAN SIMPSON*,
272 pages, Réf. 0131

LOTUS 1-2-3 POUR L'ENTREPRISE
par DOMINIQUE HELLE ET GUY BOUSSAND,
256 pages, Réf. 0147

LOTUS 1-2-3 PROGRAMMATION DES MACRO-
COMMANDES *par GOULVEN HABASQUE*,
144 pages, Réf. 0150 F

INTRODUCTION A WORDSTAR 2000
par DAVID KOLODNEY ET THOMAS BLACKADAR,
256 pages, Réf. 0160

LOGISTAT, ANALYSE STATISTIQUE DES DONNÉES
par FREDJ TEKAIA ET MICHELE BIDEI,
352 pages, Réf. 0115

ALGORITHMES *par PIERRE BEAUFILS ET WOLFRAM LUTHER*,
296 pages, Réf. 0149

***POUR UN CATALOGUE COMPLET
DE NOS PUBLICATIONS***

FRANCE

6-8, Impasse du Curé
75881 PARIS CEDEX 18
Tél. : (1) 42.03.95.95
Télex : 211801

U.S.A.

2344 Sixth Street
Berkeley, CA 94710
Tel. : (415) 848.8233
Telex : 336311

ALLEMAGNE

Vogelsanger. WEG 111
4000 Düsseldorf 30
Postfach N° 30.09.61
Tel. : (0211) 61 80 2-0
Telex : 08588163



Paris • Berkeley • Düsseldorf

Achevé d'imprimer le 10 avril 1996 sur les presses de l'imprimerie «La Source d'Or»
63200 Marsat - Dépôt légal : 2^e trimestre 1996 - Imprimeur n° 1995

Les 18 jeux présentés dans ce livre utilisent toutes les possibilités de votre micro-ordinateur : son, couleur, graphismes animés, etc. Pour cette raison, ces programmes ne fonctionnent que sur l'ordinateur pour lequel ils ont été conçus et sont difficilement adaptables à des matériels différents.

Au-delà du jeu lui-même, l'étude de ces programmes vous enseignera de nombreuses techniques de programmation qui vous seront très utiles pour développer vos propres programmes.

0205 0486 78 F



9 782736 102050