

NOUVEAUX JEUX sur ATMOS

C. Kosniowski



Edimicro

NOUVEAUX JEUX SUR ATMOS

par
Czes KOSNIOWSKI



Adaptation pour ATMOS réalisée par Michel BIRADES.

Traduction autorisée de l'ouvrage publié en langue anglaise.
Original title : « Games to play on your Oric-1 ».
Copyright © 1983 Czes Kosniowski

First published in English 1983 by : Shiva Publishing Ltd.

All rights reserved. No part of
this publication may be reproduced,
stored in a retrieval system, or
transmitted, in any form or by any
means, electronic, mechanical, recording
or otherwise, without the prior
permission of the publishers.

Imprimé en France. Droits mondiaux réservés.

« La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1^{er} de l'article 40). » « Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal. »

© 1984 F.D.S./Edimicro pour la traduction française

© 1984 F.D.S./Edimicro pour l'adaptation à l'ATMOS

ISBN 2-904457-18-6

EDIMICRO
DÉPARTEMENT ÉDITIONS DE F.D.S. SARL
121/127, avenue d'Italie, 75013 Paris

Table des matières

Préface	9
1. La chasse au trésor	11
2. Ne criez pas au loup	15
3. Alunissage	20
4. L'oiseau et la chenille	24
5. Les vingt-cinq carrés	29
6. Test d'intelligence	34
7. Hypnose	39
8. Les danseurs	41
9. Show graphique	44
10. Le Nim carré	48
11. Oh ! Susana	52
12. Extinction	54
13. Courbes	57
14. Cascade	60
15. La Cucaracha	63
16. Horloge puzzle	66
17. Les fenêtres	71
18. Surfaces	74
19. Le Danube bleu	77
20. Le serpent	79
21. Poursuite spatiale	83
22. Epelez-le	86
23. Aide au dessin	89
24. Rebondissement	92
25. Spirales	95
26. Knees up Mother Brown	97
27. Jeu de mémoire	101
28. Le bourreau	106
29. Découvrez les nombres	110
30. Ne soyez pas gourmand	114
Annexe 1 : Conservation d'une image sur bande	118
Annexe 2 : Renumérotation des lignes	119
Annexe 3 : Les notes musicales	124

NOTE DE L'ÉDITEUR

Tous les jeux du présent volume, originellement conçus et écrits pour l'ORIC-1, ont été adaptés à l'ATMOS et dûment testés sur cette machine. Les trente jeux fonctionnent donc tous sur ATMOS. La plupart d'entre eux tournent également sur ORIC-1. Seuls les six jeux suivants ne tournent pas sur ORIC-1 : Jeux n^{os} : 1 - 10 - 16 - 17 - 18 - 27.

Préface

Vous découvrirez dans ce livre de nombreux programmes nouveaux et captivants adaptés à votre *Oric*. Certains sont des jeux, d'autres de pures distractions (utilisant le son et les graphiques). Les jeux sont de différentes natures. *Les danseurs*, par exemple, est un « jeu d'arcade » qui exige de vous une réponse rapide et précise. Avec quelques jeux, vous avez le temps d'exercer votre esprit. Par exemple, dans *Ne criez pas au loup*, vous êtes pourchassé par une meute de loups. Vous pouvez les duper si vous réfléchissez à vos mouvements. Et rien ne presse ; le déplacement des loups n'intervient qu'après votre propre mouvement. Les jeux du troisième type ressemblent à des puzzles dans lesquels vous avez à remettre en ordre les divers éléments.

Les listings sont issus de programmes exécutables, sans erreur. Chaque programme est accompagné d'informations qui en décrivent l'objet général. Habituellement, pour donner une idée de ce que vous pouvez attendre du programme, un exemple d'écran est fourni. De plus, de courtes descriptions du programme et de ses différents sous-programmes sont données pour vous aider à en comprendre le fonctionnement.

Chaque programme a été écrit dans un style qui devrait vous permettre de l'appréhender dans son ensemble. Le mot BASIC GOTO n'est utilisé dans aucun programme. En effet, un emploi excessif de ce mot tend à rendre un programme difficile à comprendre, et conduit à ce que l'on appelle couramment une programmation « spaghetti ». Il est relativement facile d'éviter l'emploi de GOTO sur l'*Oric*, les programmes le prouvent à l'évidence.

Toutes les instructions GOSUB sont du type GOSUB BLANK où la variable BLANK reçoit une valeur dans les premières lignes du programme. Cela facilite la renumérotation des lignes. Un court programme de renumérotation est donné en annexe, vous pouvez l'ajouter à la fin de vos programmes si vous le désirez.

Soyez attentif lorsque vous tapez un programme, et tout particulièrement aux signes deux-points et point-virgule, au chiffre 1 et à la lettre I et enfin au chiffre 0 et à la lettre O. Lorsque vous avez fini de taper un programme, sauvegardez le (CSAVE) avant de l'exécuter (RUN), au cas où vous auriez fait une erreur de frappe. Tous les programmes sont garantis sans erreur ; ainsi, si une erreur de syntaxe (SYNTAX ERROR) se produit, comparez soigneusement votre programme au listing de ce livre. Notez que les espaces présents dans les listings répondent à un souci de clarté ; il n'est pas nécessaire de les taper.

Nous avons pris la peine de nous assurer que tous les programmes sont « robustes ». En conséquence, ils ne devraient pas se bloquer si vous tapez sur une mauvaise touche. Par exemple, si le programme attend un chiffre et que vous introduisiez par erreur une lettre, il ne se bloquera pas.

Les programmes ont été entièrement testés aussi bien par Ann, Kora et Inga que

par de nombreux visiteurs intègres. Je les remercie tous pour avoir été des « cobayes » et pour avoir fait de nombreuses remarques pertinentes. Je remercie aussi Kathleen McCaffrey qui a réussi à transformer mes brouillons en un texte lisible à l'aide d'un ordinateur ACT SIRIUS 1.

Czes Kosniowski
Newcastle, juin 1983

1. La chasse au trésor

Voici un jeu simple dans lequel vous avez à trouver l'emplacement d'un trésor caché. De nombreux buissons sont clairement répartis sur votre écran. Le trésor est caché sous l'un d'eux. Vous êtes au buisson qui clignote et pouvez vous déplacer de buisson en buisson en utilisant la barre d'espacement ; votre *Atmos* vous indiquera approximativement quelle distance vous sépare du trésor. Approximativement, parce que le nombre affiché est la partie entière de cette distance.

Pour vous embrouiller, l'*Atmos* affichera parfois un point d'interrogation au lieu de la distance. En outre, si celle-ci est supérieure ou égale à 10, la lettre X est affichée.

Il n'y a pas de limite au nombre de tentatives autorisées, mais essayez de trouver le trésor en trois coups environ.

CHASSE AU TRÉSOR

#	?	?	?	2	#	?	#	#	#	#	#	#	#
2	1	#	1	2	#	4	#	#	#	#	#	#	#
2	?	1	#	#	#	4	#	#	#	#	#	#	#
?	2	#	#	#	#	4	#	#	#	#	#	#	?
3	#	#	#	#	#	5	#	#	#	#	?	X	?
#	#	#	#	#	#	5	#	#	#	#	9	#	#
#	#	#	#	#	#	#	#	#	#	#	?	#	#
#	#	#	#	#	#	#	#	#	#	#	?	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	#	#	#	#	#
#	#	#	#	#	#	#	#	#	?	#	X	?	?

Utilisez les touches curseur pour vous déplacer
Utilisez la <barre d'espacement> pour voir.

```

10 REM *****
20 REM * LA CHASSE AU TRESOR *
30 REM *****
40 REM
50 VISU=500:INIT=1000 :INFO=2000
60 PING:WAIT 20:PING
100 REM * THE MAIN PROGRAM *
110 GOSUB INFO:GOSUB INIT
120 REPEAT
130 : PLOT X,Y,12:GET K$:KK=ASC(K$):PLOT X,Y,8
140 : IF KK=32 THEN GOSUB VISU
150 : IF KK=8 AND X>4 THEN X=X-2
160 : IF KK=9 AND X<30 THEN X=X+2
170 : IF KK=10 AND Y<22 THEN Y=Y+2
180 : IF KK=11 AND Y>2 THEN Y=Y-2
190 UNTIL STATE
200 SHOOT:WAIT 40:PING:WAIT 40:PING
210 PRINT CHR$(14)+E$+"L";SPC(11);"BIEN JOUE"
220 PRINT E$+"A"+"      Autre partie? O ou N"
230 PRINT CHR$(14)+CHR$(11);:PING
240 REPEAT:GET K$:UNTIL K$="O" OR K$="N"
250 IF K$="O" THEN RUN
260 CLS:PING:PRINT "A la prochaine.":POKE #26A,3:END
500 REM * VISU
510 IF SCRNX(X+1,Y)<>35 THEN RETURN
520 DIST=INT(SQR((MM-X)*(MM-X)+(NN-Y)*(NN-Y))/2)
530 A$=CHR$(48+DIST)
540 IF DIST>9 THEN A$="X"
550 IF DIST>0 AND RND(1)<.5 THEN A$="?"
560 PLOT X+1,Y,A$:STATE=STATE+DIST=0
570 RETURN
1000 REM * Initialisation *
1010 CLS:PRINT SPC(10);"LA CHASSE AU TRESOR":PRINT
1020 FOR I=1 TO 11
1030 : PRINT SPC(2);
1040 : FOR J=0 TO 13
1050 : PRINT CHR$(27)+"H#";
1060 : NEXT
1070 : PRINT:PRINT
1080 NEXT
1090 PRINT E$+"A      Curseurs-> Mouvement"
1100 PRINT:PRINT E$+"D      Distance->Barre d'espacement";
1110 PRINT CHR$(11);CHR$(11);CHR$(11)
1120 X=16:Y=12 'INITIAL POSITION
1130 MM=2*INT(RND(9)*14+1)+2:NN=2*INT(RND(9)*11+1)
1140 STATE=FALSE
1150 RETURN
2000 REM * Information *
2010 POKE #26A,10 'Annule le bruit du curseur et de la barre d'espacement
2020 INK 0:PAPER 3:CLS:PRINT
2030 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
2040 PRINT D$:SPC(5);"LA CHASSE AU TRESOR":PRINT D$

```

```

2050 PRINT:PRINT
2060 PRINT D$;SPC(5);"Par Czes Kosniowski":PRINT D$
2070 PRINT:PRINT E$+"A Trouvez le tresor cache."
2080 PRINT "En pressant la barre d'espace on vous donnera ";
2090 PRINT "la distance approximative du tresor.Parfois ";
2100 PRINT "un ? apparait pour vous embrouiller."
2110 PRINT "Un X signifie que la distance est superieure";
2120 PRINT" ou egale a 10.      BONNE CHANCE"
2130 PRINT:PRINT E$+"B"+E$+"L      Pressez 0 pour continuer."
2140 REPEAT:K$=KEY$:K=RND(1):UNTIL K$="0"
2145 INIT=1000
2150 RETURN

```

PROGRAMME PRINCIPAL

La partie principale du programme attend que vous pressiez une touche. Si vous actionnez un curseur (valeurs 8 à 11 du code ASCII), alors la position du buisson clignotant est modifiée. Ceci est réalisé en utilisant `PLOT x, y, 8` qui annule le clignotement, et `PLOT x, y, 12` qui le provoque. Remarquez que la position $(x + 2, y)$ contient déjà le caractère de contrôle `CHR$(27) + CHR$(8)` qui annule aussi le clignotement et par conséquent seul le buisson de coordonnées $(x + 1, y)$ clignote. Si la barre d'espacement (valeur 32 du code ASCII) est pressée, alors le sous-programme `VISU` vous affiche la distance qui vous sépare du trésor. Lorsque la distance est zéro, la valeur de la variable `STATE` est vraie (`TRUE`) et le programme sort de la boucle. Divers effets de son et messages sont alors émis.

SOUS-PROGRAMMES

500 VISUalisation

Il calcule la distance (`DIST`) qui vous sépare du trésor. Si cette distance est supérieure ou égale à 10, alors le nombre est transformé en `X`, c'est pourquoi, environ une fois sur deux, la distance n'est pas révélée. Lorsque cette distance est nulle, la valeur de `STATE` devient vraie (`TRUE`) ce qui conduit le programme vers les étapes finales.

1000 INITIALisation

Il provoque l'affichage initial des buissons. La position du trésor est calculée aléatoirement. Remarquez que `PRINT CHR$(27) + « H # »`; provoque l'affichage d'un blanc suivi du symbole `#`. Le blanc contient le caractère de contrôle `CHR$(27) + CHR$(8)` qui est utilisé pour arrêter le clignotement de tous les caractères situés à droite de cette position.

2000 INFOrmation

Un texte qui décrit sommairement ce jeu est contenu dans ce sous-programme.

Remarque :

Lorsque vous allumez votre *Oric* et que vous demandez un nombre aléatoire, vous obtenez la même séquence de nombres. Pour donner un caractère de hasard à cette séquence, une ligne a été placée dans ce programme qui permet à l'ordinateur de choisir (et d'ignorer) des nombres aléatoires jusqu'à ce que le joueur presse la touche Y. Vous trouverez une telle ligne dans la plupart des programmes de ce livre.

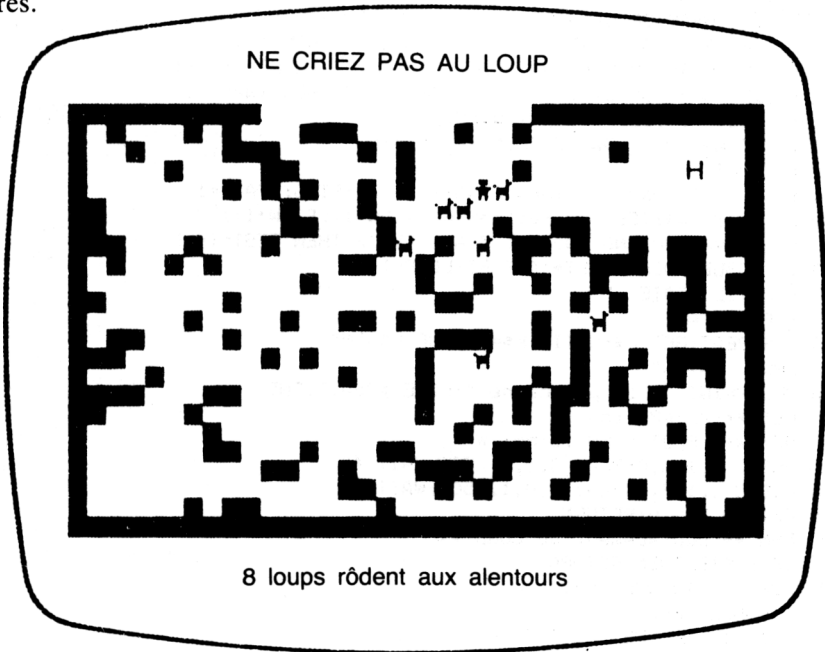
2. Ne criez pas au loup

Imaginez que vous êtes en pleine forêt et qu'il se fait tard. Vous devez rentrer chez vous (H sur l'écran). De nombreux loups sont dans les parages prêts à vous mordre s'ils viennent près de vous. Pouvez-vous regagner votre maison sans dommage ?

Pour vous déplacer, utilisez les curseurs et les touches suivantes (qui permettent les déplacements en diagonale) :

1 en haut à gauche \ en haut à droite
Z en bas à gauche / en bas à droite

Grâce à ces touches vous pouvez vous faufiler entre les arbres. Prenez votre temps et réfléchissez à tous vos mouvements. Si vous êtes poursuivi par les loups — ne vous découragez pas — vous pouvez essayer de déjouer leurs intentions. Les loups sont assez lourdauds et sont souvent coincés par les arbres et les coins. Ils ne peuvent pas se déplacer en diagonale entre les arbres.



A chaque fois que vous réussissez à rentrer chez vous, vous pouvez jouer à nouveau — avec un loup de plus.

Les loups restent cachés pendant vos premiers mouvements — ils se déplacent cependant autour de vous. Prenez garde. Et ne criez pas au loup.

```
10 REM *****
20 REM * NE CRIEZ PAS AU LOUP *
30 REM *****
40 REM
50 APPARI=500:LDEPL=600:DUR=700:DEPL=800
60 IHIT=1000:CHR=1500:INFO=2000
70 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:GOSUB CHR
120 REPEAT
130 : GOSUB IHIT
140 : REPEAT
150 : PRINT H$:REPEAT:UNTIL KEY$=""
160 : REPEAT
170 : GET G$:G=ASC(G$)
180 : ANS=FALSE
190 : IF G=49 OR G=90 OR G=47 OR G=92 THEN ANS=TRUE
200 : IF G>7 AND G<12 THEN ANS=TRUE
210 : UNTIL ANS
220 : XN=X:YN=Y
230 : IF (G=49 OR G=90 OR G=8) AND XN>2 THEN XN=XN-1
240 : IF (G=92 OR G=47 OR G=9) AND XN<35 THEN XN=XN+1
250 : IF (G=90 OR G=47 OR G=10) AND YN<23 THEN YN=YN+1
260 : IF (G=49 OR G=92 OR G=11) AND YN>3 THEN YN=YN-1
270 : PLOT X,Y,32:WW=SCRN(XN,YN)
280 : IF WW=32 OR WW=72 THEN X=XN:Y=YN
290 : PLOT X,Y,Z
300 : IF X=33 AND Y=5 THEN PING:WAIT 20:PING:STATE=1
310 : PRINT W$:TEST=FALSE:XY=SCRN(A(NW+1),B(NW+1))
320 : IF NW<=WD AND RND(1)<.3 AND XY=32 THEN TEST=TRUE
330 : GOSUB LDEPL-TEST*(APPARI-LDEPL)
340 : UNTIL STATE
350 : PRINT A$
360 : REPEAT:GET K$:UNTIL K$="O" OR K$="N"
370 UNTIL K$="N"
380 CLS:PRINT "A la prochaine.":POKE #26A,3:END
500 REM ** APPARITION **
510 NW=NW+1:PLOT A(NW),B(NW),W
520 MUSIC 1,1,6,0:PLAY 3,0,1,2500:WAIT 40
530 MUSIC 1,1,3,0:PLAY 3,0,1,2500:WAIT 50
540 PLAY 0,0,0,0:RETURN
600 REM ** Déplacement du Loup **
610 IF STATE THEN RETURN
620 FOR I=0 TO WOLVES-1
630 : GOSUB DEPL
640 : IF I<=NW THEN PLOT A(I),B(I),32
```

```

650 : A(I)=A:B(I)=B
660 : IF I<=NW THEN GOSUB DUR
670 NEXT
680 RETURN
700 REM ** DUR **
710 PLOT A,B,W
720 IF A<>X OR B<>Y OR STATE THEN RETURN
730 PLOT X,Y,"DUR":ZAP:STATE=2
740 RETURN
800 REM ** DEPL **
810 XX=(A(I)>X)-(A(I)<X):YY=(B(I)>Y)-(B(I)<Y):S=0
820 IF YY=0 THEN S=1:YY=1
830 IF XX=0 THEN S=2:XX=1
840 TEST=FALSE:T=-1
850 REPEAT
860 : T=T+1:A=A(I)+U(S,T)*XX:B=B(I)+V(S,T)*YY
870 : SS=SCRN(A,B):TT=SCRN(A(I),B):UU=SCRN(A,B(I))
880 : IF SS<W AND (TT<W OR UU<W) THEN TEST=TRUE
890 UNTIL TEST OR T>7
900 RETURN
1000 REM ** Initialisation de l'ecran **
1010 PAPER 3:INK 0:PRINT FRE(9):CLS
1020 WOLVES=WOLVES+(WOLVES<20)*(STATE=1)
1030 STATE=0
1040 NW=-1 'Nombre de loups
1050 FOR X=2 TO 36
1060 : PLOT X,2,160:PLOT X,24,160
1070 NEXT
1080 PLOT 9,2,"NE CRIEZ PAS AU LOUP"
1090 FOR Y=3 TO 23
1100 : PLOT 2,Y,160:PLOT 36,Y,160
1110 : FOR X=2 TO 35
1120 : IF RND(9)>.75 THEN PLOT X,Y,160
1130 NEXT:NEXT
1140 PLOT 5,26,"Il y a"
1150 PLOT 16,26,MID$(STR$(WOLVES),2)
1160 PLOT 19,26,"loups alentour"
1170 FOR Y=3 TO 7:FOR X=3 TO 6
1180 : PLOT 29+X,Y,32:PLOT X,16+Y,32
1190 NEXT:NEXT
1200 PLOT 33,5,72:PLOT 4,21,37
1210 X=4:Y=21:Z=37
1220 REM Position des loups
1230 FOR I=0 TO WOLVES-1
1240 : REPEAT
1250 : A(I)=INT(RND(1)*34+2):B(I)=INT(RND(1)*21+3)
1260 : UNTIL SCRN(A(I),B(I))=32 AND B(I)-A(I)<17
1270 NEXT
1280 RETURN
1500 REM ** CHR **
1510 H$=CHR$(30)+E$+"L"+E$+"A" VOTRE COUP"
1520 W$=CHR$(30)+CHR$(14)
1530 A$=CHR$(30)+E$+"L"+E$+"D" Autre partie O ou N "

```



```

1540 W=38:DIM A(20),B(20)
1550 START=#B400+37*8:AA=#A000
1560 POKE AA,2:IF PEEK(AA)<>2 THEN START=START-#B000
1570 FOR I=START TO START+7:READ R:POKE I,R:NEXT
1580 START=START+8
1590 FOR I=START TO START+7:READ R:POKE I,R:NEXT
1600 STATE=0
1610 FOR S=0 TO 2:FOR T=0 TO 8
1620 : READ U(S,T),V(S,T)
1630 NEXT:NEXT
1640 RETURN
1700 DATA 14,14,4,31,14,14,10,10
1710 DATA 2,3,35,30,30,30,18,18
1720 DATA 1,1,1,0,0,1,1,-1,-1,1,0,-1,-1,0,0,0,0
1730 DATA 1,0,1,-1,1,1,0,-1,0,1,-1,-1,1,-1,0,0,0
1740 DATA 0,1,1,1,-1,1,1,0,-1,0,1,-1,-1,-1,0,-1,0,0
2000 REM ** INFORMATION **
2010 E$=" "+CHR$(27):D$=E$+CHR$(4)+"J"
2020 PAPER 6:INK 0:CLS:PRINT
2030 PRINT D$;SPC(6);"NE CRIEZ PAS AU LOUP":PRINT:PRINT D$
2040 PRINT:PRINT D$;SPC(6);"par Czes Kosniowski":PRINT D$
2050 PRINT:PRINT"Vous etes en pleine foret et il y a des loups";
2060 PRINT" dans les parages. Essayez de rentrer chez ";
2080 PRINT "Vous pouvez vous faufiler entre les arbres,";
2090 PRINT "les loups,non."
2100 PRINT:PRINT "CONTROLES:"
2110 PRINT "Curseurs pour aller a gauche, a droite, en haut, a gauche"
2120 PRINT " 1 pour Haut-Gauche \ pour Haut-Droite"
2130 PRINT " Z pour Bas-Gauche / pour Bas-droite"
2150 PRINT "combien de loups voulez vous au debut?"
2160 PRINT "Donnez un nombre entre 2 et 9:"
2170 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$>"1" AND G$<:"
2180 WOLVES=VAL(G$)
2190 POKE #26A,2 'Curseurs sans effet
2200 RETURN

```

PROGRAMME PRINCIPAL

La partie principale du programme attend que vous pressiez une touche. Si c'est l'une des 8 clés permises qui est pressée, le programme procède à votre déplacement et en conséquence à celui des loups.

SOUS-PROGRAMMES

500 APPARItion

Il détermine le moment où un loup apparaît. Il y a 30 % de chance qu'un loup apparaisse pendant un déplacement. Remarquez que quand un loup apparaît, les autres loups ne se déplacent pas. Cela donne une chance au joueur de s'échapper.

600 LDEPLacement

Ce sous-programme, avec le sous-programme DEPL, déplace les loups (même ceux qui sont invisibles). Il contrôle aussi si un loup vous a atteint. Les loups invisibles ne nuisent pas tant qu'ils ne sont pas devenus visibles.

700 DUR

C'est un sous-programme de LDEPL. Son but est d'afficher DUR! lorsqu'un loup vous a atteint.

800 DEPL

Un simple calcul est fait pour savoir comment déplacer les loups. Si c'est possible, chaque loup se rapproche de vous; cependant, s'il n'y a pas de position plus proche il reculera d'une case. Ainsi, un loup peut parfois faire un mouvement de va-et-vient.

1000 INITialisation

Il crée la disposition de la forêt, au hasard, et détermine la position initiale des loups.

1500 CHR

Les caractères spéciaux (vous et les loups) sont créés grâce à ce sous-programme. De plus, ce sous-programme initialise quelques constantes (nombre de loups, etc.)

2000 INFOrmation

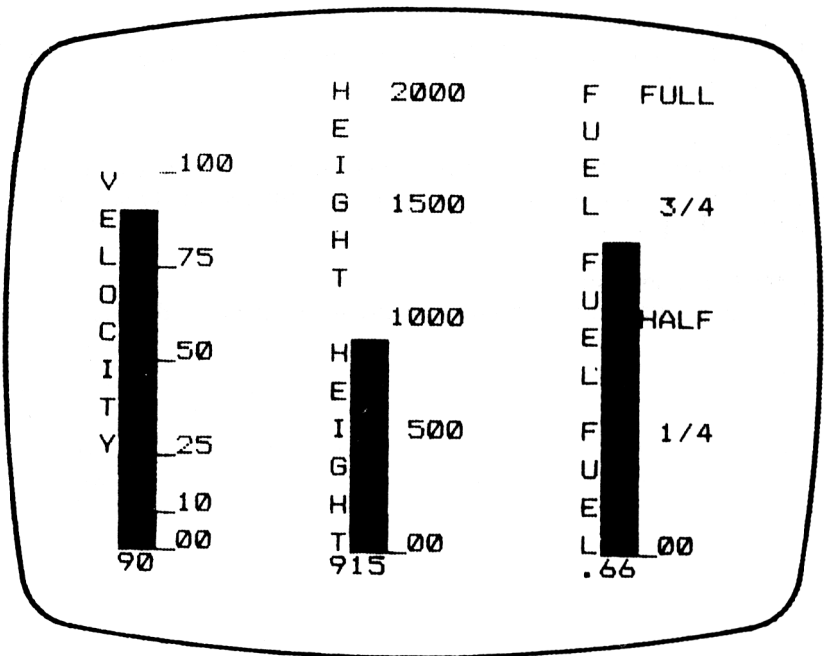
Ce sous-programme contient une brève description du jeu.

3. Alunissage

Vous êtes aux commandes d'un vaisseau lunaire. Il vient de se séparer de la fusée porteuse et est en phase d'approche. Seules les rétro-fusées (barre d'espacement) peuvent vous ralentir. Les réserves de carburant sont faibles ; il ne faut donc pas presser la barre d'espacement tout le temps.

Le vaisseau a été construit avec des matériaux importés à bon marché, il est donc très sensible à la vitesse. C'est pourquoi, si vous dépassez la vitesse de 100, votre vaisseau explose !

Essayez de vous poser à une vitesse inférieure ou égale à 10. Votre *Atmos* vous fournit un graphique (accompagné de sons) sur votre vitesse, votre altitude et votre réserve de carburant.



```

--
10 REM *****
20 REM * ALUNISSAGE *
30 REM *****
40 REM
50 CALCCARB=500:AFFI=600:INSTRUCT=1000:IHIT=1500
60 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INSTRUCT:GOSUB IHIT
120 REPEAT
130 : VELOCITY=VELOCITY+ACCEL
140 : SOUND 4,125-VELOCITY/4,7
150 : IF VELOCITY<0 THEN SOUND 5,200-VELOCITY,7
160 : Y1=INT(VELOCITY/5):Y2=INT(VELOCITY-5*Y1):X=7
170 : GOSUB AFFI
180 : PLOT 7,26,MID$(STR$(INT(VELOCITY)),2)+ " "
190 : HEIGHT=HEIGHT-VELOCITY/100
200 : Y1=INT(HEIGHT/8):Y2=INT(HEIGHT-8*Y1):X=19
210 : GOSUB AFFI
220 : PLOT 18,26,MID$(STR$(INT(HEIGHT*10)),2)+ " "
230 : IF KEY$<>" " THEN GOSUB CALCCARB
240 UNTIL HEIGHT<.1 OR VELOCITY>100
250 EXPLODE
260 FOR I=0 TO INT(VELOCITY/10)
270 : PAPER 1:WAIT 10:PAPER 4:WAIT 10:PAPER 7:WAIT 10
280 NEXT
290 PRINT:PRINT CHR$(14);CHR$(10);CHR$(14);
300 PRINT CHR$(11);CHR$(4);
310 IF VE>100 THEN PRINT W1$
320 IF VE<=100 AND VE>20 THEN PRINT SPC(10);W2$
330 IF VE<=20 AND VE>10 THEN PRINT SPC(13);W3$
340 IF VELOCITY<=10 THEN PRINT SPC(12);W4$
350 WAIT 500:PRINT
360 PRINT E$+"N" Autre partie O ou N";SPC(8)
370 PRINT CHR$(4)
380 REPEAT:GET K$:UNTIL K$="O" OR K$="N"
390 IF K$="O" THEN RUN
400 CLS:PRINT "A la prochaine":POKE #26A,3:END
500 REM ** CALCul du CARBurant **
510 IF FUEL<1 THEN RETURN
520 FUEL=FUEL-F
530 Y1=INT(FUEL/8):Y2=INT(FUEL-8*Y1):X=32
540 IF Y1<0 THEN RETURN
550 VELOCITY=VELOCITY-BRAKES:GOSUB AFFI
560 PLOT 31,26,MID$(STR$(INT(FUEL/2)/100),2)+ " "
570 RETURN
600 REM ** AFFIchage **
610 IF Y1<0 OR Y1>24 THEN RETURN
620 FOR I=0 TO 1
630 : PLOT X+I,24-Y1,32:PLOT X+I,25-Y1,32+Y2
640 : IF Y1>0 THEN PLOT X+I,26-Y1,A$
650 NEXT
660 RETURN
1000 REM ** INSTRUCTIONS **

```

```

1010 CLS:PAPER 4:INK 7:E$=" "+CHR$(27)
1020 D$=E$+"J"+CHR$(4):PRINT
1030 PRINT D$+E$+"C";SPC(9);"ALUNISSAGE"
1040 PRINT D$:PRINT:PRINT
1050 PRINT D$+E$+"C";SPC(5);"De Czes Kosniowski"
1060 PRINT D$:PRINT:PRINT
1070 PRINT "Vous etes aux commandes d'un vaisseau lunaire.";
1080 PRINT "Il est en chute libre et seules les retro-";
1090 PRINT "fusees (la barre d'espace) vous ralentissent."
1100 PRINT "Ne depassez pas la vitesse de 100."
1110 PRINT:PRINT "Essayez d'atterrir a une vitesse <10."
1120 PRINT:PRINT;SPC(10);"BONNE CHANCE"
1130 PRINT:PRINT"      Pressez 0 pour continuer."
1140 POKE #26A,10 ' Curseurs sans effet
1150 REPEAT:K$=KEY$:K=RND(1):UNTIL K$="0"
1160 RETURN
1500 REM ** IHIT **
1510 CLS:PAPER 7:INK 0:PING
1520 START=#B400+33*8;AA=#A000
1530 POKE AA,2:IF PEEK(AA)<>2 THEN START=START-#B000
1540 FOR I=0 TO 63:POKE START+I,127:NEXT
1550 FOR I=0 TO 6:FOR J=0 TO 6-I
1560 : POKE START+J+I*8,0
1570 NEXT:NEXT
1580 FOR I=1 TO 25
1590 : PLOT 6,I,MID$("      V I T E S S E V I T",I,1)
1600 : PLOT 18,I,MID$("A L T I D U D E A L T I",I,1)
1610 : PLOT 31,I,MID$("C A R B U R A N T C A R B",I,1)
1620 NEXT
1630 FOR I=0 TO 25
1640 : PLOT 1,I,1:PLOT 13,I,4:PLOT 26,I,2
1650 NEXT
1660 A$="(:B$="!"
1670 FOR I=0 TO 1
1680 : FOR J=1 TO 25
1690 : PLOT 19+I,J,A$:PLOT 32+I,J,A$
1700 : NEXT
1710 : PLOT 1,24+I,5:PLOT 7+I,24,A$:PLOT 7+I,25,A$
1720 NEXT
1730 PLOT 21,1,"2000":PLOT 21,7,"1500"
1740 PLOT 21,13,"1000":PLOT 21,19," 500"
1750 PLOT 21,25,B$+"00"
1760 PLOT 34,1,"PLEIN":PLOT 34,7," 3/4"
1770 PLOT 34,13,"1/2 ":PLOT 34,19," 1/4"
1780 PLOT 34,25,B$+"00"
1790 PLOT 9,5,B$+"100":PLOT 9,10,B$+"75"
1800 PLOT 9,15,B$+"50":PLOT 9,20,B$+"25"
1810 PLOT 9,23,B$+"10":PLOT 9,25,B$+"00"
1820 PRINT E$ "à ":PLOT 31,26,"FULL"
1830 HEIGHT=199:FUEL=199:ACCEL=.9:BRAKES=1.8
1840,F=.9+.4*RND(1):VELOCITY=10
1850 W1$=E$+"N Trop rapide, vaisseau EXPLOSE "
1860 W2$=E$+"NCRASH "

```

1870 W3\$=E\$+"NPAS MAL"
1880 W4\$=E\$+"NBIEN JOUE"
1890 PLAY 0,7,0,0
1900 RETURN

PROGRAMME PRINCIPAL

A chaque cycle, votre vitesse et votre altitude sont affichées — aussi bien numériquement que grâce à un graphique. Ce graphique est réalisé grâce au sous-programme AFFichage. Si une touche est pressée, du carburant est alors consommé (s'il en reste) grâce au sous-programme CALCCARBurant.

SOUS-PROGRAMMES

500 CALCCARBurant

Il est utilisé pour calculer et afficher la quantité de carburant restant. L'affichage graphique utilise le sous-programme AFFichage.

600 AFFichage

Il affiche simplement le caractère spécial relatif à la vitesse, à la hauteur ou au carburant.

1000 INSTRUCTIONS

Ce sous-programme contient les brèves instructions du jeu.

1500 INITIalisation

Les caractères spéciaux sont créés ici. Il n'y a aucune fantaisie dans ces caractères, simplement un rectangle d'une certaine hauteur. Les caractères ont les codes ASCII de 32 à 39 et vont du blanc jusqu'à une barre pleine. Lorsque ces caractères ont été créés, l'affichage initial sur l'écran est réalisé.

Remarque :

Après l'exécution du programme vous pouvez presser la touche RESET de l'Atmos, les caractères standards reprendront leur place.

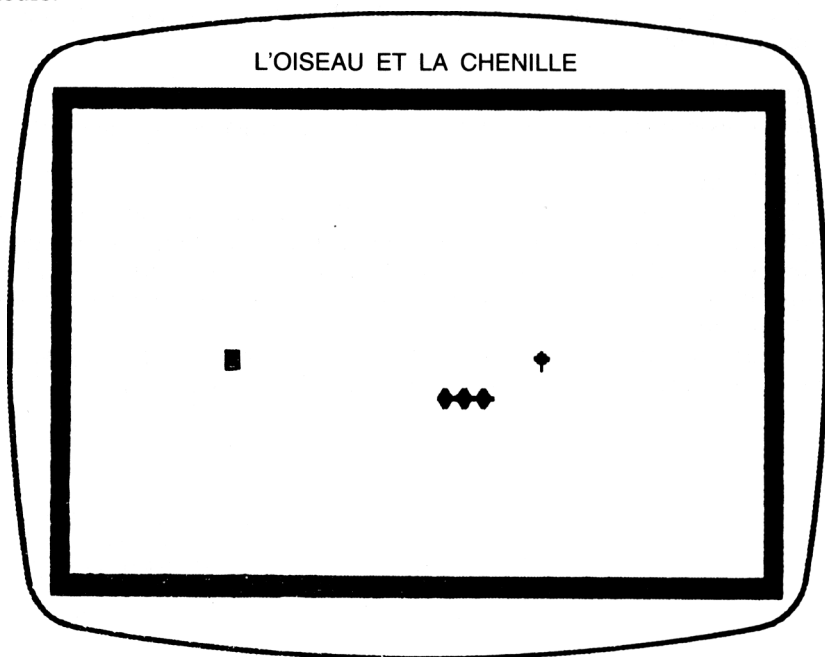
4. L'oiseau et la chenille

Dans ce jeu, vous êtes une chenille affamée qui rampe sur l'écran. Il y a de la nourriture non loin de vous, mais pourrez-vous l'attraper à temps ? La chenille ne peut pas s'arrêter, mais vous pouvez modifier la direction de son mouvement en utilisant les curseurs.

Attention ! La chenille ne doit pas heurter le mur (le bord de l'écran) car, dans ce cas, elle s'écraserait. Elle ne doit pas, non plus, rebrousser subitement chemin car elle s'autodétruirait et le jeu cesserait. Ainsi, si la chenille se déplace vers la gauche, ne pressez pas le curseur « droit ». Au lieu de cela, pressez d'abord l'un des curseurs « haut » ou « bas ».

Il y a un danger de plus autour de vous. Un oiseau affamé (contrôlé par l'*Oric*) vole dans les parages à la recherche de la nourriture et de la chenille.

Dès qu'un morceau de nourriture est mangé, un autre morceau apparaît ailleurs.



Regardez quelle est la quantité de nourriture que la chenille peut manger avant de s'écraser, de s'autodétruire, ou encore d'être mangée par l'oiseau.

```

10 REM *****
20 REM * L'OISEAU ET LA CHENILLE *
30 REM *****
40 REM
50 OISE=500:DIRECT=700:NOUR=900:INIT=1000
60 COMMENC=1500:CHR=1700:DESCRIP=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB DESCRIP:GOSUB COMMENC
115 PRINT">>> ON COMMENCE"
120 REPEAT
130 : GOSUB INIT:GOSUB NOUR:STATE=FALSE
140 : REPEAT
150 : K$=KEY$:IF K$ <>"" THEN GOSUB DIRECT
160 : PLOT P(2),Q(2),32
170 : FOR I=2 TO 1 STEP -1
180 : P(I)=P(I-1):Q(I)=Q(I-1)
190 : NEXT
200 : P(0)=P(0)+DP:Q(0)=Q(0)+DQ
210 : PQ=SCRN(P(0),Q(0))
220 : IF PQ<>32 AND PQ<>C THEN STATE=PQ
230 : IF PQ=C THEN ZAP:WAIT 30:GOSUB NOUR:SUM=SUM+1
240 : FOR I=0 TO 2:PLOT P(I),Q(I),R:NEXT
250 : GOSUB OISE
260 : IF X=A AND Y=B THEN ZAP:WAIT 30 :GOSUB NOUR
270 : FOR I=0 TO 2
280 : IF X=P(I) AND Y=Q(I) THEN STATE=1
290 : NEXT
300 : UNTIL STATE
305 PRINT STATE
310 : EXPLODE
320 :FOR I=1 TO 3:PAPER 1:WAIT 20:PAPER 7:WAIT 20:NEXT
330 : PAPER 0:PRINT B$
340 : W=- (STATE=1)- (STATE=223)-2*(STATE=160)
350 PRINTA$(W)
360 : PRINTCHR$(27)+"T"+E$+"GLa chenille a mange ";SUM;
370 : IF SUM=1 THEN PRINT"morceau" "+C$ C$"
380 : IF SUM<>1 THEN PRINT"morceaux" "+C$"
390 : PRINTCHR$(27)+"R"+E$+"G";
400 PRINT" Une autre partie? O ou N";SPC(10)
410 : REPEAT:GETG$:UNTIL G$="O" OR G$="N"
420 UNTIL G$<>"O"
430 CLS:PRINT"A BIENTOT":INK 7:POKE #26A,3:END
500 REM ** MOUVEMENTS DE L'OISEAU **
510 XX=SGN(A-X):YY=SGN(B-Y)
520 IF RND(1)<.3 THEN XX=SGN(P(0)-X):YY=SGN(Q(0)-Y)
530 IF XX*YY <> 0 AND RND(1)<.5 THEN XX=0
540 IF XX*YY<>0 THEN YY=0

```



```

550 PLOT X,Y,32:X=X+XX:Y=Y+YY:PLOT X,Y,Z
560 RETURN
700 REM ** DIRECTION **
710 KK=ASC(K$)
720 IF KK<8 OR KK>11 THEN RETURN
730 DP=(KK=8)-(KK=9)
740 DQ=(KK=11)-(KK=10)
750 RETURN
900 REM ** NOURRITURE **
910 REPEAT
920 : A=INT(RND(1)*36+2):B=INT(RND(1)*25+1)
930 UNTIL SCRN(A,B)=32
940 PLOT A,B,C
950 RETURN
1000 REM ** INITIALISATION **
1010 PAPER 0:INK 2:PRINT FRE(9):CLS
1020 FOR X=2 TO 38
1030 : PLOT X,0,160:PLOT X,26,160
1040 NEXT
1050 FOR Y=1 TO 25
1060 : PLOT 2,Y,160:PLOT 38,Y,160
1070 NEXT
1080 REM POSITION DE L'OISEAU
1090 X=INT(RND(1)*36+2):Y=3:Z=223
1100 PLOT X,Y,Z
1110 REM CHENILLE
1120 P(0)=30:P(1)=31:P(2)=32
1130 Q(0)=22:Q(1)=22:Q(2)=22
1140 FOR I=0 TO 2:PLOT P(I),Q(I),R:NEXT
1150 DP=-1:DQ=0 'Direction
1160 C=38:R=37:SUM=0
1170 RETURN
1500 REM ** COMMENCEMENT **
1510 START=#B400+37*8:AA=#A000
1520 POKE AA,2:IF PEEK(AA)<>2 THEN START=START-#8000
1530 GOSUB CHR
1540 START=START+8:GOSUB CHR
1550 START=START+456:GOSUB CHR
1560 B$=CHR$(27)+"Q"+CHR$(27)+"G "
1570 C$=" "+CHR$(14)
1580 A$(0)=B$+"La chenille s'est mangee elle-meme"+C$
1590 A$(1)=B$+"L'oiseau a mange la chenille"+C$
1600 A$(2)=B$+"La chenille a heurte le mur"+C$
1610 B$=CHR$(30)+B$+" L'OISEAU ET LA CHENILLE "+C$
1620 RETURN
1700 REM CHR
1710 FOR I=START TO START +7:READ X:POKE I,X:NEXT
1720 RETURN
1730 DATA 12,12,30,63,63,30,12,12
1740 DATA 4,14,31,31,14,4,4,4
1750 DATA 63,45,63,63,51,63,33,63
2000 REM ** DESCRIPTION **
2010 INK 0:PAPER 6:CLS:PRINT

```

```

2020 POKE #26A,2
2030 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
2040 PRINTD$;SPC(6);"L'OISEAU ET LA CHENILLE":PRINTD$
2050 PRINT:PRINT
2060 PRINTD$;SPC(7);"par Czes Konowski"
2070 PRINTD$;PRINT
2080 PRINT"Guidez la chenille vers la nourriture"
2090 PRINT"Mais, évitez les murs et l'oiseau!"
2120 PRINT:PRINT"Pour changer de direction, utilisez les fleches"
2140 PRINT:PRINT E$+"L":SPC(8);"Appuyez sur 0 pour commencer"
2150 REPEAT:G$=KEY$:G=RND(1):UNTIL G$="0"
2160 RETURN

```

PROGRAMME PRINCIPAL

Le programme principal déplace sur l'écran la chenille et l'oiseau. Un test est réalisé pour voir si la chenille a heurté le mur, s'est autodétruite ou a mangé un morceau de nourriture. Un test similaire est fait pour savoir si l'oiseau a mangé la chenille ou de la nourriture.

SOUS-PROGRAMMES

500 OISEau

Il contrôle le mouvement de l'oiseau. A chaque occasion, l'ordinateur décide si l'oiseau ira vers la chenille (30 % des cas) ou vers la nourriture (70 % des cas). Ayant choisi son but, l'oiseau se déplace d'une case vers celui-ci.

700 DIREcTion

Après avoir testé le clavier, ce sous-programme calcule la direction que doit prendre la chenille. Seuls les curseurs ont un effet.

900 NOURriture

Lorsqu'il n'y a plus de nourriture en jeu, ce sous-programme sélectionne la position d'un nouveau morceau. Un test est fait pour s'assurer que la nourriture apparaît dans un endroit vide (hors du mur, de l'oiseau ou de la chenille).

1000 INITialisation

La première image qui apparaît sur l'écran est créée par ce sous-programme. Il crée le mur d'enceinte et positionne l'oiseau et la chenille sur l'écran.

1500 COMMENCement

Le programme principal nécessite trois caractères spéciaux : l'oiseau, la nourriture et la chenille. L'oiseau remplace le caractère ASCII 223 (qui est l'inverse-vidéo du caractère de code 95). Les dessins de la nourriture et de la chenille remplacent les caractères de code ASCII 37 et 38. La chenille est formée de trois losanges mis bout à bout.

1700 CHR

Ce court sous-programme lit les DATA pour les caractères spéciaux et les implante (POKE) aux bons emplacements.

2000 DEScriptiOn

Il contient une brève description du programme.

5. Les vingt-cinq carrés

C'est un jeu du type puzzle dans lequel l'*Atmos* mélange les vingt-cinq lettres de l'alphabet de A à Y. Le but est de les remettre en ordre dans la position ci-dessous :

```

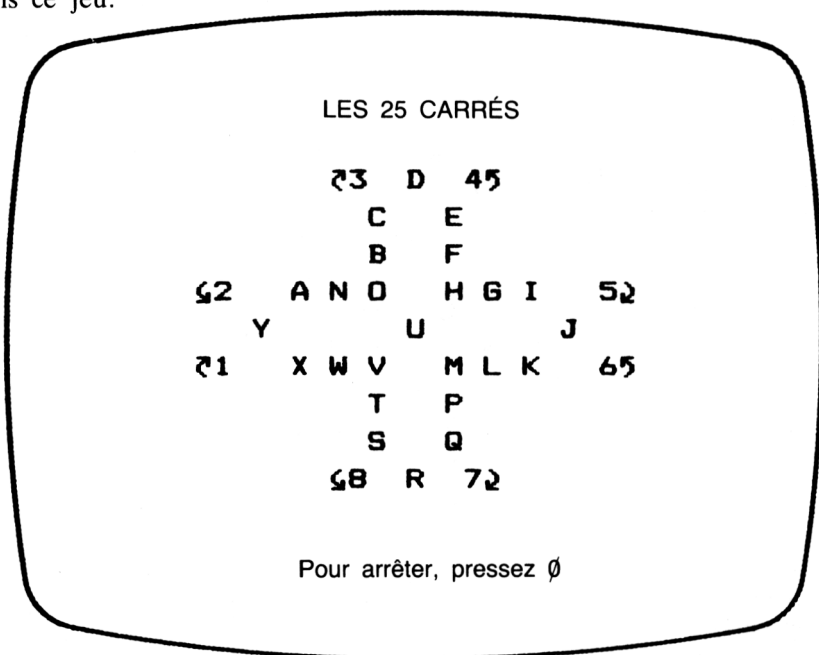
      E
    D   F
    C   G
  X Y B   H I J
W       A       K
  V U T   N M L
    S   O
    R   P
      Q
  
```

Seuls certains groupes de lettres peuvent être déplacés à tous moments. Les groupes sont formés de 8 lettres qui se trouvent dans la partie haute, droite ou gauche de la figure. Les déplacements sont effectués en pressant les touches 1 à 8. Chacune des touches impaires déplace un groupe de lettres dans le sens des aiguilles d'une montre alors que les touches paires les déplacent dans le sens inverse. Par exemple, presser la touche 1 déplace les 8 lettres de la gauche (TUVWXYBA sur le dessin) dans le sens des aiguilles d'une montre. La touche $\emptyset$ est utilisée si l'on abandonne le jeu.

```

      E
    D   F
    C   G
  X Y B   H I J
W       A       K
  V U T   N M L
    S   O
    R   P
      Q
  
```

Ce jeu a plusieurs niveaux de difficulté. Au niveau le plus facile (niveau 1) seuls deux mouvements sont faits par l'ordinateur. Voyez si vous pouvez réussir avec ces deux mouvements. Aux niveaux plus élevés, le jeu peut s'avérer assez difficile et provocant. La réflexion importe plus que la vitesse dans ce jeu.



```

10 REM *****
20 REM * VINGT-CINQ CARRÉS *
30 REM *****
40 REM
50 MOUV=500:AFFI=1000:INITIAL=1500:DESCRIP=2000
60 PING:WAIT 20:PING
100 REM ** BOUCLE PRINCIPALE **
110 GOSUB DESCRIP:GOSUB INITIAL
120 GOSUB AFFI:STATE=FALSE
130 REPEAT
140 : ANSWER=FALSE
150 : REPEAT:GET G$:UNTIL G$>"/" AND G$<"9"
160 : GOSUB MOUV+50*VAL(G$):IF G$<>"0" THEN GOSUB AFFI
170 : REM CHECK FOR END
180 : I=0
190 : REPEAT
200 : I=I+1:IF A$(I)<>64+I THEN ANSWER=TRUE
210 : IF I>24 THEN ANSWER=TRUE:STATE=1
220 : UNTIL ANSWER
230 UNTIL STATE
240 PRINT CHR$(30)
  
```

```

250 REM FIN
260 IF STATE=1 THEN PRINT W$:PING:WAIT 30:PING
270 PLOT 10,3,"Autre partie 0 or N"
280 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
290 IF G$="O" THEN RUN
300 CLS:PRINT "A bientot.":POKE #26A,3:END
500 REM ** MOUV 0 - Quitter **
510 STATE=2
520 ANSWER=TRUE
530 RETURN
540 REM Simplement pour remplir l'espace **
550 REM ** MOUV 1 - Gauche direct **
560 B%=A%(25)
570 FOR I=25 TO 21 STEP -1:A%(I)=A%(I-1):NEXT
580 A%(20)=A%(1):A%(1)=A%(2):A%(2)=B%
590 RETURN
600 REM ** MOUV 2 - Gauche direct **
610 B%=A%(20)
620 FOR I=20 TO 24:A%(I)=A%(I+1):NEXT
630 A%(25)=A%(2):A%(2)=A%(1):A%(1)=B%
640 RETURN
650 REM ** MOUV 3 - haut indirect **
660 B%=A%(8)
670 FOR I=8 TO 2 STEP -1:A%(I)=A%(I-1):NEXT
680 A%(1)=B%
690 RETURN
700 REM ** MOUV 4 - Haut direct **
710 B%=A%(1)
720 FOR I=1 TO 7:A%(I)=A%(I+1):NEXT
730 A%(8)=B%
740 RETURN
750 REM ** MOUV 5 - Droit indirect **
760 B%=A%(14)
770 FOR I=14 TO 9 STEP -1:A%(I)=A%(I-1):NEXT
780 A%(8)=A%(1):A%(1)=B%
790 RETURN
800 REM ** MOUV 6 - Droit direct **
810 B%=A%(8)
820 FOR I=8 TO 13:A%(I)=A%(I+1):NEXT
830 A%(14)=A%(1):A%(1)=B%
840 RETURN
850 REM ** MOUV 7 - Bas indirect **
860 B%=A%(20)
870 FOR I=20 TO 15 STEP -1:A%(I)=A%(I-1):NEXT
880 A%(14)=A%(1):A%(1)=B%
890 RETURN
900 REM ** MOUV 8 - Bas direct **
910 B%=A%(14)
920 FOR I=14 TO 19:A%(I)=A%(I+1):NEXT
930 A%(20)=A%(1):A%(1)=B%
940 RETURN
1000 REM ** AFFichage **
1010 FOR I=1 TO 25

```

```

1020 : PLOT X%(I),Y%(I),A%(I)
1030 : PLOT X%(I),Y%(I)+1,A%(I)
1040 NEXT
1050 RETURN
1500 REM ** INITIALisation **
1510 CLS:PAPER 0:INK 7
1520 DIM A%(25),X%(25),Y%(25)
1530 FOR I=1 TO 25
1540 : A%(I)=64+I
1550 : READ X%(I),Y%(I)
1560 NEXT
1570 FOR I=5 TO 22:PLOT 0,I,10:NEXT
1580 PLOT 0,1,10:PLOT 0,2,10:PLOT 0,3,2:PLOT 1,3,12
1590 REM MIXING
1600 FOR J=1 TO ZZ:GOSUB MOUV+50*INT(8*RND(9)+1):NEXT
1610 W$=" "
1620 W$=E$+"A"+CHR$(4)+W$+"BIEN JOUE "+W$+CHR$(4)
1630 PRINT:PRINT V$:PRINT E$+"A"
1640 START=#B400+90*8:AA=#A000
1650 POKE AA,2:IF PEEK(AA)<>2 THEN START=START-#B000
1660 FOR I=0 TO 3
1670 : START=START+8
1680 : FOR J=0 TO 7:READ R:POKE START+J,R:NEXT
1690 NEXT
1700 FOR I=1 TO 8
1710 : READ U,V,X$
1720 : PLOT U-1,V,4:PLOT U-1,V+1,4
1730 : PLOT U,V,X$:PLOT U,V+1,X$
1740 : PLOT U+3,V,7:PLOT U+3,V+1,7
1750 NEXT
1760 PLOT 0,24,4:PLOT 9,24,"Pour abandonner pressez 0"
1770 RETURN
1780 REM DONNEES
1790 DATA 19,13,17,11,17,9,17,7,19,5,21,7,21,9,21,11
1800 DATA 23,11,25,11,27,13,25,15,23,15,21,15
1810 DATA 21,17,21,19,19,21,17,19,17,17,17,15
1820 DATA 15,15,13,15,11,13,13,11,15,11
1830 REM DONNEES-CARACTERES
1840 DATA 30,6,10,18,16,16,8,4
1850 DATA 4,8,16,16,18,10,6,30
1860 DATA 30,24,20,18,2,2,4,4
1870 DATA 4,4,2,2,18,20,24,30
1880 REM DONNEES POUR L'AFFICHAGE
1890 DATA 8,15,[ 1,8,11,\ 2,15,5,[ 3,22,5,4]
1900 DATA 29,11,5↑,29,15,6],22,21,7↑,15,21,\ 8
2000 REM ** DESCRIPTION **
2010 CLS:PAPER 0:INK 7:POKE #26A,3
2020 E$=" " +CHR$(27):D$=E$+CHR$(4)+"J"
2030 V$=E$+"A"+D$+" Vingt cinq carres. " +D$
2040 PLOT 1,2,1
2050 PRINT:PRINT V$:PRINT:PRINT:PRINT
2060 PRINT D$;SPC(7);"de Czes Kosniowski"
2070 PRINT D$:PRINT

```

```

2080 PRINT "Reordonnez les 25 lettres dans la"
2090 PRINT "position correcte, comme ci-dessous."
2100 PRINT SPC(17);"E":PRINT SPC(16);"D F"
2110 PRINT SPC(16);"C G":PRINT SPC(14);"XYB HIJ"
2120 PRINT SPC(13);"W A K":PRINT SPC(14);"VUT NML"
2130 PRINT SPC(16);"S O":PRINT SPC(16);"R P"
2140 PRINT SPC(17);"Q"
2150 PRINT:PRINT "Utilisez les touches 1 a 9 pour "
2160 PRINT "les mouvements."
2170 PRINT E$+"DDifficulte? 1 a 9: "
2180 REPEAT:G$=KEY$:GG=RND(1):UNTIL VAL(G$)>0
2190 ZZ=3*VAL(G$)-1:POKE #26A,2
2200 RETURN

```

PROGRAMME PRINCIPAL

La partie principale de ce programme attend que vous pressiez une des touches 1 à 8. Le mouvement approprié est ensuite effectué et un test regarde si les lettres sont dans leur position correcte.

SOUS-PROGRAMMES

500 MOUVement

Les lignes 500 à 540 contiennent les instructions pour quitter le jeu. Les autres déplacements 1 à 8 sont séparés par des intervalles de 50 entre les lignes 550 et 940. Les déplacements sont appelés par l'instruction GOSUB MOVE + 50 *VAL (G\$) où G\$ est la touche pressée.

1000 AFFichage

Il place les lettres dans leur position.

1500 INITIALisation

Les 25 lettres sont rangées suivant leurs codes ASCII (65 à 89) dans les variables entières A % (I). Les positions des lettres sur l'écran sont rangées dans les variables entières X % (I) et Y % (I). Quatre caractères spéciaux sont créés. Ce sont quatre flèches courbes.

2000 DESCRIPTION

Une brève description du jeu est donnée dans ce sous-programme.

6. Test d'intelligence

De nombreuses questions dans ces tests sont basées sur des séquences de nombres et/ou de lettres. Par exemple,

19, 16, 13, 10, ...

O, K, H, E, ...

Vous devez donner le terme suivant de la séquence. Votre *Atmos* peut être utilisé pour créer toutes sortes de séquences qui vous testeront et vous amuseront. Pour chaque séquence, essayez de trouver la bonne réponse. Rien ne presse, l'ordinateur attend patiemment lorsque vous réfléchissez.

Après dix séquences votre score vous sera donné. Ne prenez pas le jeu trop au sérieux, le but est de vous amuser.

TEST D'INTELLIGENCE SEQUENCE 3

O S W A E I M

Quelle est la prochaine lettre ? P

Non, essayez encore

quelle est la prochaine lettre ? R

La réponse est Q

Pressez Y pour continuer

```

10 REM *****
20 REM * TEST D'INTELLIGENCE *
30 REM *****
40 REM
50 SEQUENCE=500:INFO=1000
60 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL *
110 GOSUB INFO
120 FOR ZZ=1 TO 10
130 : REM NOMBRES ALEATOIRES
140 : A=INT(RND(1)*10+1):N=INT(RND(1)*4)
150 : B=INT(RND(1)*10):C=1+2*(RND(1)<.5)
160 : REM ECRAN
170 : CLS:PRINT " ";I$:PRINT
180 : PRINT ,,, " SEQUENCE ";ZZ:PRINT
190 : PRINT CHR$(4):PRINT E$+"J ";
200 : GOSUB SEQUENCE+100*INT(RND(1)*5):PRINT
210 : PRINT CHR$(4):PRINT:PRINT
220 : FOR JJ=1 TO 2
230 : INPUT "Quel est le terme suivant";Y$:PRINT
240 : IF Y$=X$ THEN PING:JJ=3:PRINT W$:SC=SC+1
250 : IF JJ=1 THEN SHOOT:PRINT E$+"BNon - essayez encore"
260 : IF JJ=2 THEN SHOOT:PRINT E$+"BLa reponse est ";X$
270 : PRINT
280 : NEXT
290 : PRINT:PRINT E$+"L"+E$+"D Pressez 0 pour continuer"
300 : REPEAT:GET G$:UNTIL G$="0"
310 NEXT
320 CLS:PRINT:PRINT D$+I$:PRINT D$:PRINT
330 PRINT " Votre score est sur 10 de ";SC
340 FOR I=1 TO SC:PING:WAIT 20:NEXT
350 PRINT:PRINT
360 PRINT E$+"L"+E$+"D Autre partie? 0 ou N"
370 REPEAT:GET G$:UNTIL G$="0" OR G$="N"
380 IF G$="0" THEN RUN
390 CLS:PRINT "A bientot. ":END
500 REM * Arithmetique *
510 IF C<0 THEN B=B+80+INT(RND(1)*10)
520 FOR I=1 TO 6:PRINT A*I*C+B;" ";:NEXT
530 X$=MID$(STR$(A*I*C+B),2)
540 PRINT:RETURN
600 REM * lettres-arithemetiques *
610 FOR I=N TO N+6
620 W=A*C*I+B:W=W+26*INT(W/26):PRINT CHR$(W);" ";
630 NEXT
640 W=A*C*I+B:W=W+26*INT(W/26):X$=CHR$(W):PRINT
650 RETURN
700 REM * Arithmetique double *
710 W=80+INT(RND(1)*10)
720 FOR I=1 TO 6
730 PRINT W;" ";
740 W=W+C*A:C=-C:A=A+B+1
750 NEXT

```

```

760 PRINT:X$=MID$(STR$(W),2)
770 RETURN
800 REM * Arithmetique croissante *
810 W=85+INT(RND(1)*10)
820 D=1+2*(RND(1)<.5)
830 FOR I=1 TO 6
840 PRINT W;" ";
850 W=W+C*A:C=C*D:A=A+I
860 NEXT
870 X$=MID$(STR$(W),2):PRINT
880 RETURN
900 REM * Arithmetique mixte *LIST1100-1200
910 FOR I=N TO N+6
920 W=A*C*I+B:W=W+S-26*INT(W/26):Y$=CHR$(W)
930 IF 2*INT(I/2)=I THEN Y$=MID$(STR$(W-S+1),2)
940 PRINT Y$+" ";
950 NEXT
960 W=A*C*I+B:W=W+S-26*INT(W/26):X$=CHR$(W)
970 IF 2*INT(I/2)=I THEN X$=MID$(STR$(W-S+1),2)
980 PRINT
990 RETURN
1000 REM ** INFOrmation **
1010 INK 7:PAPER 0:CLS
1020 E$=" "+CHR$(27):S=65:D$=E$+"J"+CHR$(4)
1030 I$=E$+"A TEST D' INTELLIGENCE"
1040 PRINT:PRINT D$+I$:PRINT D$:PRINT:PRINT
1050 PRINT D$;SPC(7);"de Czes Kosniowski":PRINT D$
1060 PRINT:PRINT "Une sequence de nombres et/ou de lettres ";
1070 PRINT "vous est donnee.Utilisez votreintelligence ";
1080 PRINT "pour decouvrir le terme suivant. Tapez ";
1090 PRINT "le et faites <return>":PRINT:PRINT
1100 PRINT E$+"L"+E$+"B Pressez O pour commencer"
1110 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$="O"
1120 W$=E$+"BBien joue"
1130 RETURN

```

PROGRAMME PRINCIPAL

Le programme principal est constitué d'une boucle qui est exécutée 10 fois. A chaque passage des nombres aléatoires et une séquence type (une parmi cinq) sont sélectionnés. Ils sont utilisés pour créer une séquence aléatoire de lettres et/ou de nombres.

Après l'affichage de la séquence, l'ordinateur attend une réponse. Après l'introduction de cette dernière, il donne la bonne réponse.

SOUS-PROGRAMMES

500 SEQUENCE

Il est constitué de cinq séquences type décrites plus loin. Elles sont appelées par l'instruction `GOSUB SEQUENCE + 100 *INT (RND (I) *5)`.

500 *Séquences de nombres en progression arithmétique*

Exemples : 2, 4, 6, 8, 10, 12, ...

1, 7, 13, 19, 25, ...

21, 17, 13, 9, 5, ...

Dans une séquence arithmétique la différence entre deux termes consécutifs est constante. Dans les exemples précédents, les différences sont de 2 pour la première séquence, de 6 pour la seconde et de -4 pour la troisième.

600 *Séquence arithmétique de lettres*

Exemples : A, C, E, G, I, ...

U, X, A, D, G, ...

O, N, M, L, K, ...

R, Z, H, P, X, F, ...

Une séquence arithmétique de lettres est semblable à une séquence arithmétique de nombres, ces derniers étant remplacés par des lettres de l'alphabet. La séquence est donc constituée de lettres choisies à intervalle régulier. Lorsque on atteint la fin de l'alphabet, on revient alors au début de celui-ci comme le montrent les exemples deux et quatre.

700 *Deux séquences arithmétiques*

Exemples : 85, 91, 80, 96, 75, 101, ...

80, 72, 85, 67, 90, 62, ...

91, 101, 112, 125, 141, 161, ...

Ces séquences sont constituées de deux séquences arithmétiques dont les éléments alternent. L'une est croissante alors que l'autre est décroissante.

800 *Séquences arithmétiques croissantes.*

Exemples : 87, 97, 108, 121, 137, 157, ...

90, 87, 91, 85, 94, 81, ...

88, 83, 77, 69, 58, 43, ...

Dans ces séquences la différence entre deux termes consécutifs (qu'ils soient positifs ou négatifs) augmente d'une quantité croissante. Par exemple, dans la troisième séquence les différences sont de 5, 6, 8, 11, 15 qui augmentent donc respectivement de 1, 2, 3 et 4.

900 *Séquences arithmétiques mixtes*

Exemples : 14, P, 18, T, 22, X, 26, ...

22, S, 16, M, 10, G, 4, ...

J, 12, N, 16, R, 20, V, ...

Ces séquences sont des combinaisons entre des séquences arithmétiques de nombres et de lettres.

1000 INFOrmation

Il contient une brève description du programme.

7. Hypnose

Ce programme est un simple divertissement — ce n'est pas un jeu — mettez-vous tranquillement devant votre écran et regardez.

```
10 REM *****
20 REM * HYPNOSE *
30 REM *****
40 REM
50 INFO=500
60 PING:WAIT 20:PING
100 REM * PROGRAMME PRINCIPAL **
110 GOSUB INFO
120 FOR G=0 TO 40 STEP 10
130 : HIRES:POKE #26A,10:PLAY 2,0,0,0
140 : FOR Y=198 TO 0 STEP -6
150 : CURSET 1.2*Y,Y,3:FILL 200-Y,1,17
160 : SOUND 2,250-Y*G,8
170 : NEXT
180 : HIRES:POKE #26A,10:EXPLODE
190 : FOR I=0 TO 3:PAPER 1:WAIT 10:PAPER 0:WAIT 5:NEXT
200 : PLAY 2,0,0,0
210 : FOR Y=0 TO 199 STEP 6
220 : CURSET 1.2*Y,Y,3:FILL 200-Y,1,16+RND(1)*8
230 : SOUND 2,250-Y*G,8
240 : NEXT
250 : WAIT 10
260 NEXT
270 PLAY 0,0,0,0:WAIT 50:E$=" "+CHR$(27)
280 PRINT:PRINT E$+"B"+E$+"L Autre partie? O ou N"
290 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
300 TEXT:IF G$="O" THEN RUN
310 CLS:PRINT "A bientot. ":POKE #26A,3:END
500 REM ** INfOrMation **
510 INK 6:PAPER 0:CLS:POKE #26A,10:PRINT
520 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
530 PRINT D$:SPC(11):"HYPNOSE ":PRINT D$:PRINT
540 PRINT:PRINT D$:SPC(6):"de Czes Kosniowski
550 PRINT D$:PRINT:WAIT 50:PRINT "RELAXEZ VOUS";
560 FOR I=0 TO 23:PRINT ".":WAIT 10:NEXT:PRINT
570 PRINT:PRINT "Vous allez vous faire":WAIT 100
580 PRINT E$+"LHYPNOTISER.":WAIT 400
590 RETURN
```

PROGRAMME PRINCIPAL

Le programme principal remplit l'écran en haute résolution (HIRES) de deux façons. Tout d'abord avec des rectangles rouges de taille croissante, puis avec des rectangles d'autres couleurs de taille décroissante en surimpression. Le son accompagne cet effet hypnotique alors qu'une explosion vous fait sursauter.

SOUS-PROGRAMMES

500 INFOrmation

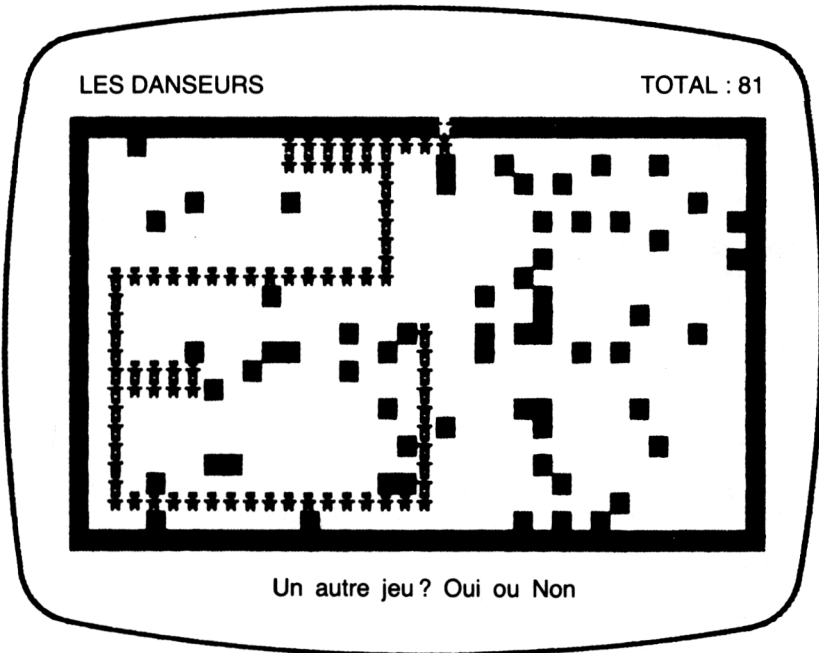
Il donne une courte description du programme.

8. Les danseurs

Nous avons déjà tous fait des chaînes en papier, en pliant, découpant puis dépliant du papier. Essayez de faire une longue chaîne de danseurs sur votre *Atmos*.

Au début un homme apparaît au centre de l'écran. Un instant plus tard un autre homme apparaît juste en dessous de lui. Ce procédé est ensuite répété. En utilisant les curseurs vous pouvez changer la direction d'apparition des hommes. Ce procédé s'arrête si vous touchez un obstacle (l'obstacle peut être un homme déjà sur l'écran).

Le nombre d'hommes sur l'écran est affiché en haut à droite.




```

10 REM *****
20 REM * LES DANSEURS *
30 REM *****
40 REM
50 TCH=500:OBSTACLE=600
60 INITI=1000:COMMENC=1500:DESCRIP=2000
70 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB DESCRIP:GOSUB COMMENC
120 REPEAT
130 : GOSUB INITI
140 : REPEAT
150 : K$=KEY$:IF K$(<)"* THEN GOSUB TCH
160 : X=X+DX:Y=Y+DY:STATE=SCRN(X,Y)<>C
170 : PLOT X,Y,Z:AMOUNT=AMOUNT+1
180 : PLOT 35,1,MID$(STR$(AMOUNT),2)
190 : GOSUB OBSTACLE
200 : UNTIL STATE
210 :PLOT X,Y,128+Z:EXPLODE:WAIT 40
220 : PLOT 10,25,"Autre partie? O ou N"
230 : REPEAT:GET G$:UNTIL G$="O" OR G$="N"
240 UNTIL G$="N"
250 CLS:PRINT "A bientot. ": POKE #26A,3:END
500 REM ** TouCHE **
510 K=ASC(K$):IF K<8 OR K>11 THEN RETURN
520 DX=(K=8)-(K=9):DY=(K=11)-(K=10)
530 RETURN
600 REM ** OBSTACLE **
610 IF RND(1)<.2 THEN RETURN
620 REPEAT
630 : A=INT(RND(1)*34+2):B=INT(RND(1)*21+3)
640 UNTIL SCRN(A,B)=C
650 PLOT A,B,160
660 RETURN
1000 REM ** INITIialisation **
1010 PAPER 6:INK 0:CLS
1020 FOR X=2 TO 36:PLOT X,2,160:PLOT X,24,160:NEXT
1030 FOR Y=2 TO 24:PLOT 2,Y,160:PLOT 36,Y,160:NEXT
1040 X=19:Y=13:DX=0:DY=1:PLOT X,Y,Z:STATE=FALSE
1050 PLOT X,Y,Z:STATE=FALSE:W=20:AMOUNT=1
1060 PLOT 2,25,4:PLOT 3,25,12
1070 PLOT 2,1,"LES DANSEURS"
1080 PLOT 27,1," TOTAL: 1"
1090 RETURN
1500 REM ** INITIALisation **
1510 START=#B400+38*8:AA=#A000
1520 POKE AA,2:IF PEEK(AA)<>2 THEN START=START-#8000
1530 FOR I=START TO START+7:READ R:POKE I,R:NEXT
1540 Z=38:C=32
1550 RETURN
1560 DATA 14,14,4,31,14,14,10,10
2000 REM ** DESCRIPTION **
2010 CLS:PAPER 1:INK 7:PRINT

```

```

2020 POKE #26A,2 'Curseurs sans effet
2030 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
2040 PRINT D$;SPC(10);"LES DANSEURS":PRINT D$:PRINT:PRINT
2050 PRINT D$;SPC(7);"de Czes Kosniowski":PRINT D$
2060 PRINT:PRINT "Faites une chaine de danseurs.Utilisez";
2070 PRINT "les curseurs pour changer de direction":PRINT
2080 PRINT "Mais - faites attention aux obstacles!"
2090 PRINT:PRINT E$+"L";SPC(7);"Pressez 0 pour jouer."
2100 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$="0"
2110 RETURN

```

PROGRAMME PRINCIPAL

Le programme principal dessine un homme et regarde s'il a heurté un obstacle.

SOUS-PROGRAMMES

500 TouCHE

Il est utilisé lorsque une touche est pressée. Lorsqu'un curseur est utilisé l'ordinateur calcule la direction que l'homme doit prendre.

600 OBSTACLE

Il crée un obstacle (approximativement dans 20 % des cas).

1000 INITIalisation

Un mur d'enceinte et un homme de coordonnées (19, 13) sont dessinés.

1500 COMMENCement

Un caractère spécial, l'homme, est créé. Il remplace le caractère de code ASCII 38.

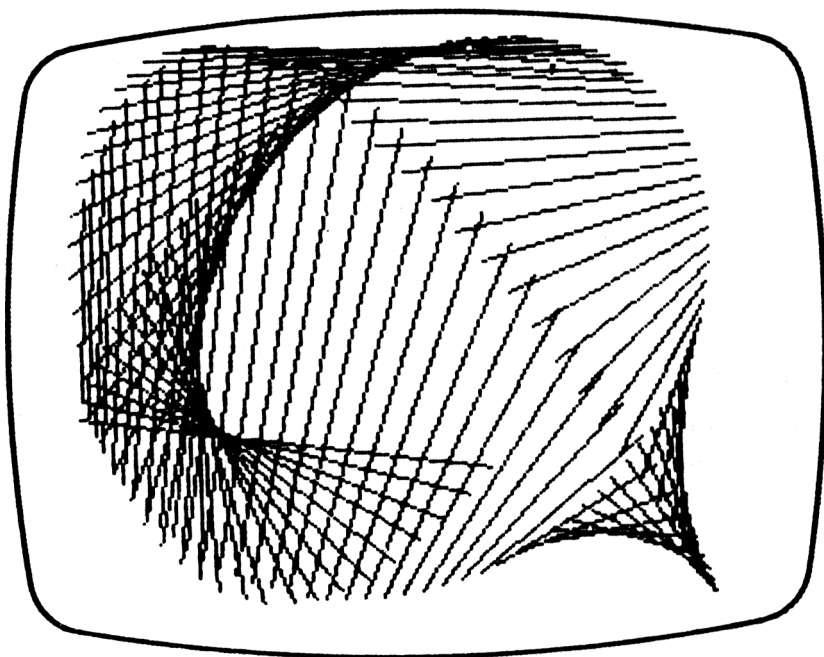
2000 DESCRiption

Il contient une courte description du programme.

9. Show graphique

Vous avez certainement déjà vu en démonstration un ordinateur puissant et coûteux exhibant ses possibilités graphiques. Certains des dessins sont très intéressants et peuvent être très facilement réalisés sur votre *Atmos*. Ce programme va vous montrer 5 différents types de dessins. Chaque dessin utilise des nombres aléatoires ; ainsi, vous ne verrez jamais deux fois le même dessin.

Reportez-vous à l'annexe 1 pour savoir comment sauvegarder vos écrans en haute résolution (HIRES) grâce à votre lecteur de cassettes.



```

10 REM *****
20 REM * SHOW GRAPHIQUE *
30 REM *****
40 REM
50 INC=100:GRAPH=400:CONTI=1000:ARCS=1100
60 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 INK 6:PAPER 0:CLS
120 REPEAT
130 : TEXT:CLS:PRINT:Z=FRE(1)
140 : E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
150 : XMID=119:YMID=99:DEF FNR(X)=INT(RND(1)*X):PF=2*PI
160 : PRINT D$;SPC(8);"SHOW GRAPHIQUE":PRINT D$
170 : PRINT:PRINT:PRINT D$;SPC(6);
180 : PRINT "de Czes Kosniowski":PRINT D$:PRINT
190 : PRINT "1. ENROULEMENT":PRINT
200 : PRINT "2. ETOILE":PRINT
210 : PRINT "3. OEIL":PRINT
220 : PRINT "4. CHARDON":PRINT
230 : PRINT "5. MOIRE":PRINT
240 : PRINT "0. SORTIE":PRINT
250 : PRINT "Entrez le nombre de votre choix"
260 : REPEAT:G$=KEY$:GG=RND(1):UNTIL G$>="0" AND G$<"6"
270 : G=VAL(G$):HIRES:POKE #26A,2
280 : IF G>0 THEN GOSUB GRAPH+INC*G:GOSUB CONTI
290 UNTIL G=0
300 TEXT:CLS:PRINT "A bientot. ":POKE #26A,3:END
500 REM ** ENROULEMENT **
510 A=.5-RND(1):B=.5-RND(1):C=.5-RND(1):D=.5-RND(1)
520 PRINT SPC(14);"ENROULEMENT":SH=15:XMID=XMID-SH
530 FOR I=1 TO 25 STEP .25
540 : X=XMID*SIN(A*I):Y=YMID*COS(B*I)
550 : CURSET X+XMID+SH,Y+YMID,1
560 : DRAW 1+XMID*SIN(C*I)-X,1+YMID*COS(D*I)-Y,1
570 NEXT
580 XMID=XMID+SH
590 RETURN
600 REM ** ETOILE **
610 PRINT SPC(16);"ETOILE":N=3+FNR(7):ANGLE=PF/N
620 W=YMID*RND(1):START=PI*(2*INT(N/2)-N-1)/2
630 FOR I=YMID TO 0 STEP -5:XX=I*COS(START)
640 : YZ=I*SIN(START):CURSET XX+XMID,YZ+YMID,1
650 : FOR J=START+ANGLE TO START+PF+ANGLE STEP ANGLE
660 : UX=W*COS(J-ANGLE/2):VZ=W*SIN(J-ANGLE/2)
670 : DRAW UX-XX,VZ-YZ,1:XZ=UX:YZ=VZ:UX=I*COS(J)
680 : VZ=I*SIN(J):DRAW UX-XX,VZ-YZ,1:XZ=UX:YZ=VZ
690 NEXT:NEXT:RETURN
700 REM ** OEIL **
710 PRINT SPC(16);"OEIL"
720 A=10+FNR(40):B=50+FNR(40):N=10+FNR(40)
730 GOSUB ARCS
740 A=(A+B)/2:B=(B+YMID)/2
750 FOR I=0 TO PF STEP PI/N

```

```

760 : CURSET XMID+A*COS(I),YMID-A*SIN(I),1
770 : DRAW (B-A)*COS(I),(B+A)*SIN(I),1
780 NEXT
790 RETURN
800 REM ** CHARDON **
810 PRINT SPC(14);"CHARDON"
820 A=10+FNR(40):B=50+FNR(40):N=10+FNR(40)
830 GOSUB ARCS
840 A=(A+B)/2:B=(B+YMID)/2
850 FOR I=0 TO PI STEP PI/N
860 : CURSET XMID-A*COS(I),YMID+A*SIN(I),1
870 : DRAW (B+A)*COS(I),(B-A)*SIN(I),1
880 NEXT
890 RETURN
900 REM ** MOIRE **
910 PRINT SPC(15);"MOIRE"
920 XMAX=224:YMAX=199:K=1+5*RND(1)
930 FOR I=0 TO XMAX STEP K
940 : CURSET 15,0,1:DRAW I,YMAX,1
950 NEXT
960 FOR I=YMAX TO 0 STEP -K
970 : CURSET 15,0,1:DRAW XMAX,I,1
980 NEXT
990 RETURN
1000 REM ** CONTINUE **
1010 PING:FOR I=1 TO 6:INK I:WAIT 75:NEXT
1020 E$=" "+CHR$(27)
1030 PRINT E$+"B"+E$+"L Pressez Y pour continuer"
1040 REPEAT:UNTIL KEY$=""
1050 REPEAT:GET G$:UNTIL G$="Y"
1060 HIRES:POKE #26A,2:RETURN
1100 REM ** ARCS **
1110 FOR I=0 TO PP STEP PP/N
1120 : CURSET XMID,YMID,1
1130 : DRAW A*COS(PP-I),A*SIN(PP-I),1
1140 : CURSET XMID,YMID,1
1150 : DRAW B*COS(PI/N+I),B*SIN(PI/N+I),1
1160 NEXT
1170 RETURN

```

PROGRAMME

Il demande à l'opérateur quel type de dessin il veut créer.

SOUS-PROGRAMMES

400 GRAPHique

Il n'y a pas de sous-programme à ce numéro de ligne. Les véritables sous-programmes commencent à la ligne 500 et sont séparés par un intervalle de 100 lignes. Ils sont appelés par `GOSUB GRAPH + INC G` ou `INC = 100` et `G` est un nombre compris entre 1 et 5 donné par l'opérateur. Les éléments de ce sous-programme sont donnés maintenant.

500 ENROUlement

Il fait tourner des lignes droites autour de l'écran.

600 STAR

Il dessine une étoile dont le nombre de branches, compris entre 3 et 9, est aléatoire. L'ordinateur remplit ensuite cette région de l'écran.

700 OEIL

Des lignes se déploient en partie en éventail depuis le centre de l'écran. Le résultat ressemble à un œil.

800 CHARDON

Il ressemble à OEIL.

900 MOIRE

Des lignes droites issues du coin en haut à gauche remplissent petit à petit tout l'écran.

1000 CONTInuer

Ce sous-programme change la couleur du dessin et vous demande si vous voulez continuer.

1100 ARCS

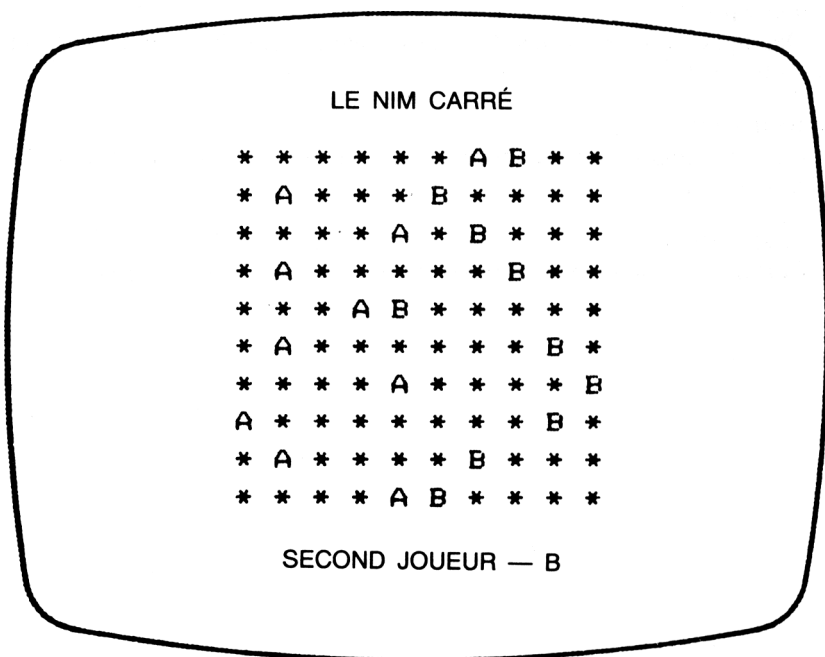
Certains des sous-programmes de GRAPH font appel à ARCS.

10. Le Nim carré

Le *Nim carré* est un jeu pour deux personnes. Comme le nom le laisse supposer, il s'agit d'une variante du jeu de *Nim*. Au cas où vous l'auriez oublié, le jeu de *Nim* est ce jeu où il vous est présenté des piles de bâtons. Chaque joueur enlève à son tour des bâtons d'une pile. Le joueur qui retire le dernier bâton gagne.

Dans le *Nim carré*, nous avons un tableau carré (un peu comme un échiquier). Il y a sur chaque ligne un A et un B. Une personne est le joueur A, l'autre le B.

Chaque joueur à son tour choisit une ligne et déplace sa lettre sur cette ligne sans sauter par-dessus celle de son adversaire. Le jeu se termine lorsque les lettres A et B de chaque ligne sont l'une à côté de l'autre. Le joueur ayant provoqué cette situation gagne la partie.



Les lettres qui clignent peuvent être déplacées en utilisant le curseur droit ou le gauche. Si vous voulez déplacer une autre lettre, utilisez les curseurs haut ou bas. Les curseurs haut ou bas n'auront un effet que si vous n'avez pas encore déplacé votre lettre. Lorsque vous l'avez déplacée appuyez sur Y pour que votre adversaire puisse jouer. Pour que la lettre Y ait un effet, il faut avoir déplacé une lettre.

```

10 REM *****
20 REM * NIM CARRE *
30 REM *****
40 REM
50 CLIGN=500:BIT=20:IHITI=1000:INFO=2000
60 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:GOSUB IHITI
120 REPEAT
130 : PLOT 8+2*X,5+2*Y,8:ZAP
140 : C=-(NOT-C):ANSWER=FALSE
150 : GOSUB CLIGN+BIT
160 : PLOT 10,25,X$(C):T=0
170 : REPEAT
180 : GET G$:G=ASC(G$)
190 : PLOT 8+2*X,5+2*Y,8
200 : IF G=8 AND C=0 AND X>0 THEN A(Y)=A(Y)-1:T=T-1
210 : IF G=8 AND C=1 AND X>A(Y)+1 THEN B(Y)=B(Y)-1:T=T-1
220 : IF G=9 AND C=0 AND X<B(Y)-1 THEN A(Y)=A(Y)+1:T=T+1
230 : IF G=9 AND C=1 AND X<9 THEN B(Y)=B(Y)+1:T=T+1
240 : IF G=8 OR G=9 THEN GOSUB CLIGN
250 : IF G=10 AND Y<9 AND T=0 THEN Y=Y+1
260 : IF G=11 AND Y>0 AND T=0 THEN Y=Y-1
270 IF G$="0" AND T<>0 THEN ANSWER=TRUE
280 : GOSUB CLIGN+BIT
290 : UNTIL ANSWER
300 : REM TEST DE VICTOIRE
310 : STATE=TRUE
320 : FOR I=0 TO 9:STATE=STATE*(A(I)=B(I)-1):NEXT
330 UNTIL STATE
340 PING:WAIT 30:ZAP:WAIT 30:PING
350 PLOT 9,25,12:PLOT 23+C,25," A GAGNE."
360 PLOT 7,3,CHR$(12)+CHR$(4)+"Autre partie? 0 ou N"
370 REPEAT:GET G$:UNTIL G$="0" OR G$="N"
380 IF G$="0" THEN RUN
390 CLS:POKE #26A,3:PRINT "A bientot. ":END
500 REM ** CLIGNotemnt **
510 PLOT 9+2*X,5+2*Y,42
520 X=C*B(Y)-(C=0)*A(Y)
530 PLOT 8+2*X,5+2*Y,12:PLOT 10+2*X,5+2*Y,8
540 PLOT 9+2*X,5+2*Y,65+C
550 RETURN
1000 REM ** IHIInitialisation **

```



```

1010 PAPER 6:INK 0:CLS:PRINT
1020 PRINT S$:PRINT D$:PRINT:PRINT
1030 FOR I=0 TO 9
1040 : PRINT SPC(6)
1050 : FOR J=0 TO 9:PRINT CHR$(136)+"*";NEXT
1060 : PRINT:PRINT:A(I)=INT(RND(1)*5)
1070 : B(I)=INT(RND(1)*(9-A(I))+A(I)+1)
1080 : PLOT 9+2*A(I),5+2*I,65:PLOT 9+2*B(I),5+2*I,66
1090 NEXT
1100 X$(0)="LE JOUEUR 'A' " :X$(1)="LE JOUEUR 'B' "
1110 C=1:Y=0:PLOT 8,25,4
1120 RETURN
2000 REM ** INFORMATION **
2010 CLS:PRINT:POKE #26A,2:PAPER 6:INK 0
2020 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
2030 S$=D$+"          NIM CARRE"
2040 PRINT S$:PRINT D$:PRINT:PRINT
2050 PRINT D$+"          de Czes Kosniowski":PRINT D$:PRINT
2060 PRINT "C'est un jeu pour deux personnes, A & B.";
2070 PRINT "Chaque joueur, a son tour, choisi uneligne ";
2080 PRINT "et deplace sa lettre ou il vout dans ";
2090 PRINT "cette ligne, sans sauter par dessus ";
2100 PRINT "celle de son adversaire. Le jeu se";
2110 PRINT " termine lorque les lettres sont, sur  chaque ligne";
2120 PRINT " l'une a cote de l'autre. Le joueur qui cree ";
2130 PRINT "cette situation  gagne.":PRINT
2140 PRINT "La lettre qui clignote peut etre "
2150 PRINT "deplacee a droite ou a gauche avec "
2160 PRINT "les curseurs. Si vous voulez deplacer"
2170 PRINT "une autre lettre, utilisez les autres curseurs.":PRINT
2180 PRINT"Lorsque vous etes prêts, tapez 0"
2190 PRINT E$+"L"+E$+"D";SPC(6);"Tapez 0 pour jouer."
2200 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$="0"
2210 RETURN

```

PROGRAMME PRINCIPAL

La boucle principale répond à la touche qui a été pressée. La valeur de T détermine si une lettre a été ou non déplacée. RÉPONSE devient vraie (TRUE) lorsque T est différent de zéro et lorsque Y a été pressée.

SOUS-PROGRAMMES

500 CLIGNotement

Il est utilisé pour faire clignoter la bonne position sur l'écran. Il y a un autre sous-programme inséré dans celui-ci — on y accède par l'instruction GOSUB FLASH + BIT, c'est-à-dire GOSUB 520.

1000 INITialisation

Il affiche le tableau et positionne au hasard les A et les B dans chaque ligne.

2000 INFOrmation

Une courte description du jeu est contenue dans ce sous-programme.

11. Oh! Susanna

Avec ce programme l'*Atmos* joue une mélodie et crée un dessin aléatoire en couleur. Pressez Q si vous voulez quitter le programme.

```
10 REM *****
20 REM * OH! SUSANNA *
30 REM *****
40 REM
50 INFO=500
60 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO
120 REPEAT
130 : PLOT FNR(37),FNR(27),CHR$(FNR(7)+1)+CHR$(32+FNR(M))
140 : I=I+1:IF I>58 THEN I=0:J=J+1:IF J>5 THEN J=1
150 : B=INT((A(I)-1)/12)
160 : IF A(I)>0 THEN MUSIC 1,J+B,A(I)-12*B,0:PLAY 3,0,1,L
170 : WAIT 5*(I)
180 UNTIL KEY$="Q"
190 PLAY 0,0,0,0
200 PRINT E$;"B";E$;"L   Autre partie? O ou N";SPC(10)
210 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
220 IF G$="O" THEN RUN
230 CLS:PRINT "A bientôt.":POKE #26A,3:INK 7:END
500 REM ** INFOrmation **
510 INK 5:PAPER 0:CLS:DEF FNR(X)=INT(RND(1)*X)
520 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4):PRINT
530 PRINT D$;SPC(10);"OH! SUSANNA":PRINT D$:PRINT:PRINT
540 PRINT D$;SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
550 PRINT "Regardez votre ordinateur qui cree  "
560 PRINT "un dessin au hasard et joue une      melodie."
570 PRINT"Tapez O pour demarrer";
580 REPEAT:GET G$:UNTIL G$="O"
590 PRINT:PRINT:PRINT "Tapez Q pour arreter.";
600 DIM A(58),T(58):FOR I=0 TO 58:READ A(I),T(I):NEXT
610 IF T(58)<>6 THEN PRINT "ERROR IN DATA!":END
620 CLS:POKE #26A,10:LORES 1:L=2500:M=95
630 RETURN
700 DATA 6,1,8,1,10,2,13,1,13,3,15,1,13,2,10,2
710 DATA 6,3,8,1,10,2,10,2,8,2,6,2,8,4,0,2,6,1
```

```

720 DATA 8,1,10,2,13,3,13,1,15,1,13,2,10,2,6,3
730 DATA 8,1,10,2,10,2,8,2,8,2,6,6,0,2,11,4,11,4
740 DATA 15,2,15,5,15,1,13,2,13,2,10,2,6,2,8,4
750 DATA 0,2,6,1,8,1,10,2,13,2,13,3,15,1,13,2
760 DATA 10,2,6,3,8,1,10,2,10,2,8,2,8,2,6,6,0,6

```

PROGRAMME PRINCIPAL

Le programme principal joue une note et affiche un caractère aléatoire issu de l'ensemble des caractères. La mélodie est jouée à chaque fois sur une gamme différente. Voir pour la musique en annexe.

SOUS-PROGRAMME

500 INFORMATION

Il contient une brève description du programme.

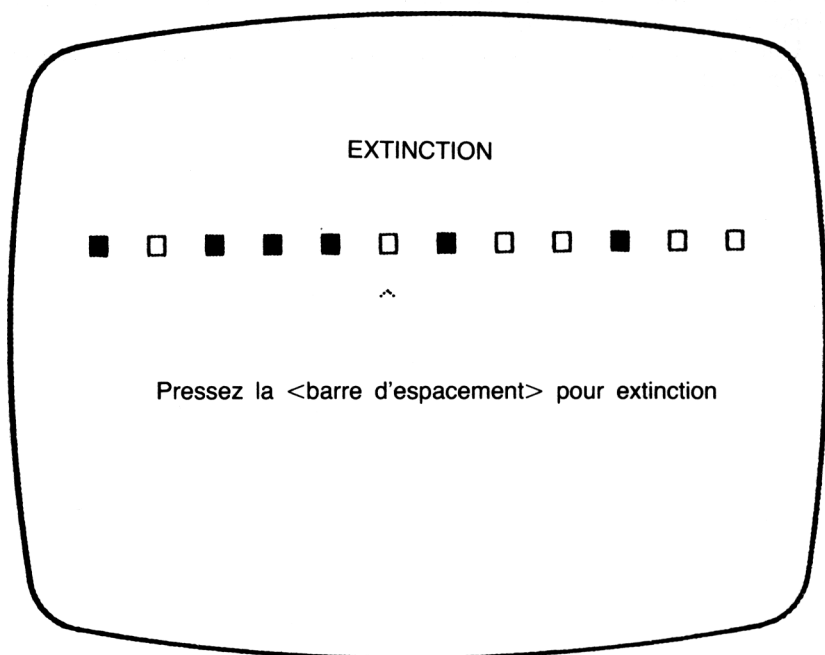
Après cette description les données (DATA) sont lues. Ces données sont des notes (rangées dans la variable A(I)) et des intervalles de temps pour le rythme de la musique (rangés dans la variable T(I)).

Une ligne de test d'erreur a été introduite pour s'assurer que toutes les données sont là.

12. Extinction

Il y a douze lampes sur l'écran qui sont éteintes ou allumées. Au début seules une ou deux lampes sont allumées. Chaque lampe se déplace de gauche à droite. Lorsqu'elles arrivent en bout de ligne, elles reviennent au début. L'objet du jeu est d'éteindre toutes les lampes en appuyant sur une touche quelconque.

Si la lampe dans la position repérée par ^ est allumée et si vous appuyez sur une touche, elle s'éteint. Si elle est éteinte, elle s'allume. Ainsi, si vous ratez votre coup, il y aura une lampe allumée de plus. Essayez donc de les éteindre toutes...



```

10 REM *****
20 REM * EXTINCTION *
30 REM *****
40 REM
50 IHITI=400:INFO=600
60 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:STATE=FALSE:GOSUB IHITI
120 REPEAT
130 : FOR K=0 TO 10
140 : WAIT ZZ:PRINT A$+CHR$(11)
150 : IF A$=B$ THEN STATE=TRUE
160 : A$=RIGHT$(A$,4)+LEFT$(A$,44)
170 : N=ASC(MID$(A$,23,1))
180 : IF NOT STATE AND KEY$<>" " THEN N=71-(N-64 AND 7)
190 : A$=LEFT$(A$,22)+CHR$(N)+MID$(A$,24)
200 : NEXT
210 : WAIT RND(1)*20+1
220 UNTIL STATE
230 PING:WAIT 30:PING:PRINT:PRINT SPC(14);"BIEN JOUE"
240 PRINT:PRINT E$+"E"+E$+"L Autre partie? O ou N"
250 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
260 IF G$="O" THEN RUN
270 CLS:PRINT "A bientot.":POKE #26A,3:END
400 REM ** Initialisation **
410 PAPER 0:INK 6:CLS:PRINT CHR$(4)
420 PRINT E$+"J";SPC(12);"EXTINCTION":PRINT:PRINT CHR$(4)
430 FOR I=0 TO 4:PRINT:NEXT
440 PLOT 19,12,94
450 POKE# 26A,2 'Curseur sans effet
460 FOR I=0 TO 11:B$=B$+E$+"F"+CHR$(255):NEXT
470 A$=LEFT$(B$,22)+"A"+MID$(B$,24)
480 A$=LEFT$(A$,14)+"A"+MID$(A$,16)
490 A$=LEFT$(A$,2)+"A"+MID$(A$,4)
500 PLOT 4,16,"Pressez la <barre d'espace> pour
510 PLOT 4,18,"eteindre les lampes."
520 RETURN
600 REM ** INfOrmatIOn **
610 PAPER 0:INK 7:CLS
620 E$=" "+CHR$(27):PRINT CHR$(4)
630 PRINT E$+"J";SPC(9);"EXTINCTION":PRINT:PRINT
640 PRINT:PRINT E$+"J";SPC(6);"de Czes Kosniowski"
650 PRINT:PRINT CHR$(4):PRINT:PRINT
660 PRINT "Pressez la barre d'espace pour changer"
670 PRINT "la couleur des lampes.":PRINT
680 PRINT "Essayez de les rendre cyan (bleue).":PRINT
690 PRINT "A quelle vitesse voulez vous que les ":PRINT
700 PRINT "lampes bougent? 0 (rapide) à 9 (lent)";
710 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$<" " AND G$>"/"
720 ZZ=VAL(G$)+1
730 RETURN

```

PROGRAMME PRINCIPAL

Les lampes sont rangées dans la chaîne A\$. Chaque lampe occupe quatre caractères. L'effet de mouvement est réalisé en déplaçant les quatre derniers caractères de A\$ (qui représentent la lampe de droite) au début de A\$. Si vous appuyez sur une touche, l'état de la lampe en sixième position est changé. Cette lampe est allumée si la valeur de N est 65 ou 70. L'expression $N = 71 - (N - 64 \text{ AND } 7)$ change 70 en 65 et vice versa. La vitesse à laquelle les lampes se déplacent est contrôlée par deux instructions WAIT. L'une est déterminée par le joueur, l'autre au hasard.

SOUS-PROGRAMMES

400 INITialisation

Il crée la chaîne des lampes A\$, trois sont allumées. La chaîne B\$ est celle où toutes les lampes sont éteintes.

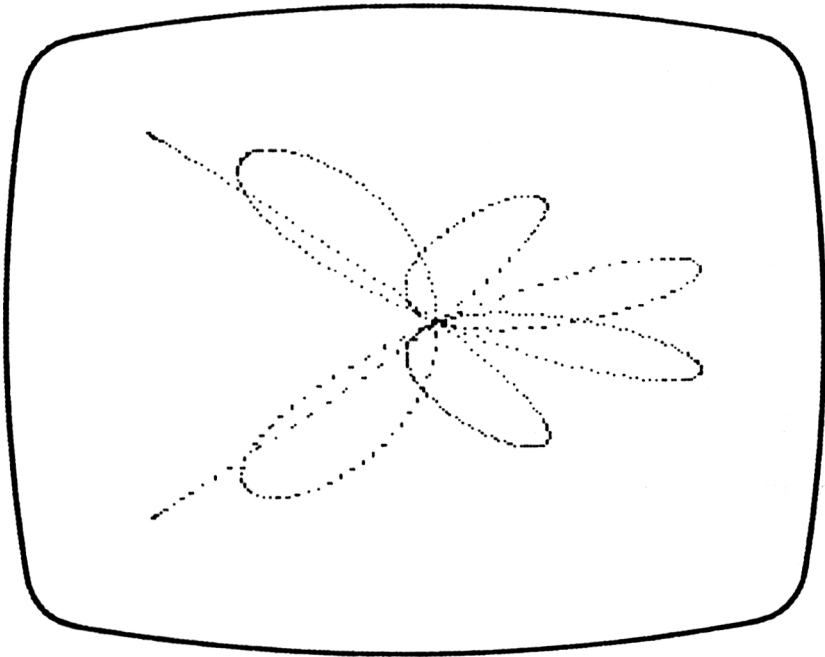
600 INFOrmation

Il contient une brève description du jeu. Le joueur décide de la vitesse de déplacement des lampes.

13. Courbes

Grâce à ce programme, vous pouvez réaliser des courbes spectaculaires. En termes techniques, l'ordinateur ne dessinera pas le graphe de fonctions polaires — il y aura un léger gauchissement. Le degré de gauchissement dépend de la valeur de A et de B que vous rentrerez lorsqu'on vous le demandera.

Reportez-vous à l'annexe 1 pour savoir comment sauvegarder vos dessins sur cassette.




```

10 REM *****
20 REM * COURBES *
30 REM *****
40 REM
50 PL=500
60 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 INK 6:PAPER 0:CLS:FOR I=0 TO 6:READ A$(I):NEXT
120 REPEAT
130 : TEXT:CLS:PRINT:ZZ=FRE(1)
140 : E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
150 : XMID=119:YMID=99:PF=2*PI
160 : XRATIO=XMID*.6:YRATIO=YMID*.9
170 : PRINT D$;SPC(12);"COURBES":PRINT D$
180 : PRINT:PRINT:PRINT D$;SPC(6);"de Czes Kosniowski"
190 : PRINT D$:PRINT:PRINT "O SORTIE"
200 : FOR I=0 TO 6:PRINT I+1;" R = ";A$(I):NEXT
210 : PRINT:PRINT "Rentrez le numero de l'equation."
220 : REPEAT:GET G$:UNTIL G$>="0" AND G$<"8"
230 : G=VAL(G$)-1:IF G>=0 THEN GOSUB PL
240 UNTIL G<0
250 CLS:PRINT "A bientot.":END
500 REM * AFFICHAGE **
510 PRINT SPC(17-LEN(A$(G)))/2;A$(G)
520 IF G=0 THEN DEF FNA(Z)=1
530 IF G=1 THEN DEF FNA(Z)=SIN(2*Z)
540 IF G=2 THEN DEF FNA(Z)=SIN(7*Z)
550 IF G=3 THEN DEF FNA(Z)=(1+2*COS(Z))/3
560 IF G=4 THEN DEF FNA(Z)=(1+COS(Z))/2
570 IF G=5 THEN DEF FNA(Z)=(1+SIN(2*Z))/2
580 IF G=6 THEN DEF FNA(Z)=(1+2*COS(2*Z))/3
590 PRINT:PRINT "Tapez les valeurs de A et B puis "
600 PRINT "tapez (return).Pour un affichage standard";
610 PRINT "A=1 & B=1.":PRINT:INPUT "Valeur de A";A
620 INPUT "Valeur de B";B
630 HIRES:POKE #26A,2
640 PRINT SPC(13-LEN(A$(G)))/2;A$(G);" A=";A;" B=";B
650 FOR Z=0 TO PF+.01 STEP .01
660 : X=XMID+XRATIO*COS(A*Z)*FNA(Z)
670 : Y=YMID+YRATIO*SIN(B*Z)*FNA(Z)
680 : CURSET X,Y,1
690 NEXT
700 PING:FOR I=1 TO 6:INK I:WAIT 75:NEXT
710 E$=" "+CHR$(27)
720 PRINT E$+"B"+E$+"L Tapez 0 pour continuer"
730 REPEAT:UNTIL KEY$=""
740 REPEAT:GET G$:UNTIL G$="0"
750 RETURN
800 DATA 1,SIN(2*Z),SIN(7*Z),1+2*COS(Z),1+COS(Z)
810 DATA 1+SIN(2*Z),1+2*COS(2*Z)

```

PROGRAMME PRINCIPAL

Il affiche les différents choix qui vous sont offerts ; essayez-les tous. Avec l'option 1 ($A = 1$ et $B = 1$) le résultat sera un cercle.

SOUS-PROGRAMMES

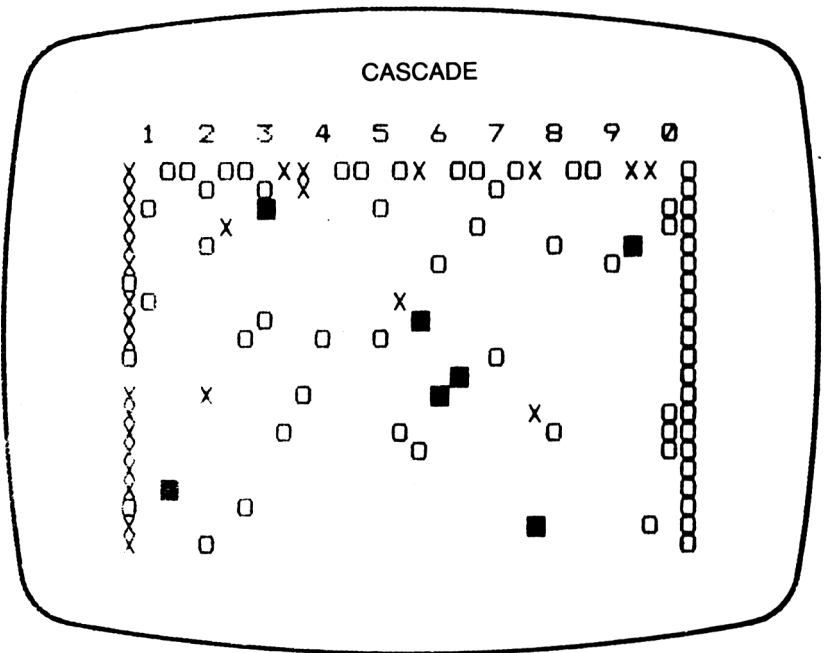
500 PL

Ce sous-programme affiche le graphe choisi. De plus, il attend que vous pressiez Y qui vous redonnera la liste des possibilités offertes.

14. Cascade

Dans *Cascade*, vous devez attraper une balle lancée du haut de l'écran vers le bas. Une balle peut être lancée d'une quelconque des dix positions en pressant une des touches comprises entre 0 et 9.

Différents obstacles se présentent devant votre balle. Si la balle heurte l'un d'eux, elle peut être réfléchiée, déviée d'un angle droit ou même absorbée. Un O dévie la balle, un X la réfléchit tandis qu'un carré rouge l'absorbe. Les obstacles peuvent changer après un choc.



```

10 REM *****
20 REM * CASCADE *
30 REM *****
40 REM
50 CHANGE=500: IHITI=1000: INFO=2000
60 PING: WAIT 20: PING
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO: GOSUB IHITI
120 REPEAT
130 : REPEAT: GET G$: UNTIL G$>"/" AND G$<": "
140 : X=2+3*VAL(G$)-30*(G$="O")
150 : Y=4:A=0:B=1
160 : PLOT X,Y,Z
170 : STATE=FALSE
180 : REPEAT
190 : U=X+A: V=Y+B: W=SCRN(U,V)
200 : IF U<4 OR U>33 OR V<4 THEN SHOOT: STATE=2
210 : IF V>24 THEN PING: WAIT 20: PING: STATE=1
220 : IF W>70 THEN GOSUB CHANGE
230 : PLOT X,Y,32: X=X+A: Y=Y+B
240 : IF STATE<2 THEN PLOT X,Y,Z
250 : UNTIL STATE
260 UNTIL STATE=1
270 PLOT 8,26,S$
280 REPEAT: GET G$: UNTIL G$="O" OR G$="N"
290 IF G$="O" THEN RUN
300 CLS: POKE #26A,3: PRINT "A bientot.": END
500 REM ** CHANGEMENT **
510 IF W=255 THEN ZAP: STATE=2
520 IF W=88 THEN A=-A: B=-B: PING
530 IF W=79 THEN C=-B: B=A: A=C: PING
540 PLOT U,V,A(FNR(P-3))
550 RETURN
1000 REM ** INITIALISATION **
1010 DEF FNR(X)=INT(RND(1)*X)
1020 PAPER 0: INK 6: CLS: Z=126
1030 PRINT E$+"B"; SPC(12); "CASCADE": PRINT
1040 PRINT E$+"B ";
1050 FOR I=1 TO 9: PRINT CHR$(48+I); SPC(2); : NEXT
1060 PRINT "O"
1070 P=7: FOR I=0 TO P-1: READ A(I): NEXT
1080 FOR J=4 TO 33
1090 : IF 3*INT(J/3)<>J-2 THEN PLOT J,4,A(1+FNR(2))
1100 NEXT
1110 FOR I=5 TO 24
1120 : PLOT 4,I,A(1)
1130 : FOR J=4 TO 32 STEP 6
1140 : IF RND(1)<.5 THEN PLOT J+FNR(6),I,A(FNR(P))
1150 : NEXT
1160 : PLOT 33,I,A(2)
1170 NEXT
1180 FOR J=5 TO 32 STEP 3: PLOT J,5+FNR(17),A(2): NEXT
1190 S$=CHR$(2)+CHR$(12)+"Autre partie? O ou N"

```

```

1200 RETURN
1210 DATA 255,88,79,79,32,32,32
2000 REM ** INFORMATION **
2010 PAPER 0:INK 6:CLS:POKE #26A,2:PRINT
2020 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
2030 PRINT D$;SPC(12);"CASCADE":PRINT D$:PRINT:PRINT
2040 PRINT D$;SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
2050 PRINT "La balle doit atteindre le bas de "
2060 PRINT "l'écran. Pressez une touche entre 0 et";
2070 PRINT "9 pour liberer une balle.Les obstacles";
2080 PRINT "peuvent absorber, devier ou reflechir"
2090 PRINT "votre balle.":PRINT
2100 PRINT "Un carre rouge absorbe la balle. Un 0"
2110 PRINT "la devie et un X la reflechit.":PRINT
2120 PRINT E$+"B"+E$+"L Tapez 0 pour commencer."
2130 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$="0"
2140 RETURN

```

PROGRAMME PRINCIPAL

La balle se déplace jusqu'au moment où elle rencontre un obstacle après quoi la direction du mouvement peut changer ou la balle peut avoir été absorbée.

Si STATE est faux (FALSE), la balle est active. La valeur de STATE devient vraie (NOT FALSE) si la balle a soit été absorbée, soit atteint le haut de l'écran.

Si STATE = 1 le jeu se termine (sauf si vous voulez rejouer).

Si STATE = 2 c'est que votre balle a été absorbée, faites un autre essai.

SOUS-PROGRAMMES

500 CHANGement

Il est utilisé lorsque la balle rencontre un obstacle.

Il change la direction du déplacement et la nature de l'obstacle rencontré.

1000 INITIalisation

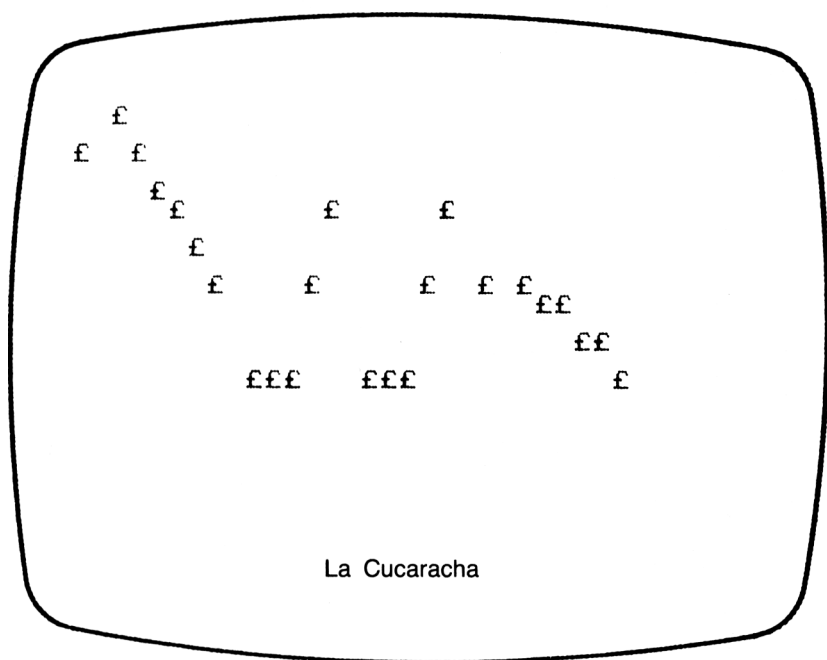
Ce sous-programme crée la disposition aléatoire des obstacles.

2000 INFOrmation

Ce sous-programme donne une brève description du jeu.

15. La Cucaracha

C'est une simple mélodie accompagnée de graphiques qui décrivent les notes.



```

10 REM *****
20 REM * LA CUCARACHA *
30 REM *****
40 REM
50 INFO=500
60 PING:WAIT 20:PING
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:J=2:K=0
120 REPEAT
130 : I=I+1:PLOT 14,26,"La Cucaracha"
140 : IF I>MAX THEN I=0:J=J+1:CLS:INK J:IF J>5 THEN J=2
150 : B=INT((A(I)-1)/12)
160 : K=K+1:IF A(I)=0 AND K>27 THEN K=0:CLS
170 : PLOT K,24-A(I),95+63*(A(I)=0)
180 IF A(I)>0 THEN MUSIC 1,J+B,A(I)-12*B,0:PLAY 3,0,1,L
190 : WAIT 5*T(I)
200 UNTIL KEY$="Q"
210 PLAY 0,0,0,0
220 PRINT E$;"B";E$;"L Autre melodie? O ou N";SPC(10)
230 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
240 IF G$="O" THEN RUN
250 CLS:PRINT "A bientot.":POKE #26A,3:INK 7:END
500 REM ** INfOrMation **
510 INK 5:PAPER 0:CLS:DEF FNR(X)=INT(RND(1)*X)
520 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4):PRINT
530 PRINT D$:SPC(9);"LA CUCARACHA":PRINT D$:PRINT:PRINT
540 PRINT D$:SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
550 PRINT "Une simple melodie.....":PRINT
560 PRINT "Tapez O pour commencer.."
570 REPEAT:GET G$:UNTIL G$="O"
580 PRINT:PRINT:PRINT "Tapez Q pour sortir";
590 DIM A(100),T(100):MAX=-1
600 REPEAT:MAX=MAX+1:READ A(MAX),T(MAX):UNTIL A(MAX)=-1
610 IF MAX<>87 THEN PRINT "ERREUR DE DONNEES":END
620 MAX=MAX-1:CLS:POKE #26A,10:INK 6:L=2500
630 RETURN
700 DATA 8,1,8,1,8,1,13,3,17,1,0,1
710 DATA 8,1,8,1,8,1,13,3,17,5,0,1
720 DATA 13,1,0,1
730 DATA 13,1,12,1,12,1,10,1,10,1,8,4,0,1
740 DATA 8,1,8,1,8,1,12,3,15,1,0,1
750 DATA 8,1,8,1,8,1,12,3,15,5,0,1
760 DATA 20,1,0,1
770 DATA 22,1,20,1,18,1,17,1,15,1,13,4,0,1
780 DATA 8,1,8,1,8,1,13,3,17,1,0,1
790 DATA 8,1,8,1,8,1,13,3,17,5,0,1
800 DATA 13,1,0,1
810 DATA 13,1,12,1,12,1,10,1,10,1,8,4,0,1
820 DATA 8,1,8,1,8,1,12,3,15,1,0,1
830 DATA 8,1,8,1,8,1,12,3,15,5,0,1
840 DATA 20,1,0,1
850 DATA 22,1,20,1,18,1,17,1,15,1,13,2
860 DATA 13,1,13,1,13,1,0,6
870 DATA -1,-1

```

PROGRAMME PRINCIPAL

A chaque fois qu'une note est jouée, un caractère approprié est dessiné sur l'écran. Les notes sont rangées dans la variable A(I) et l'intervalle de temps qui les sépare dans T(I). Pressez Q pour arrêter le programme. Reportez-vous à l'annexe 4 pour les notes de musique.

SOUS-PROGRAMME

500 INFORMATION

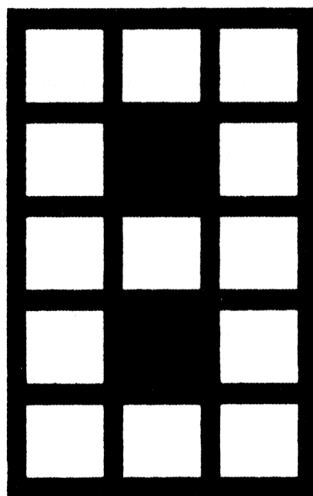
Il contient les renseignements. Les notes et les intervalles de temps sont lus dans les variables A(I) et T(I).

Une ligne qui teste les erreurs éventuelles de données est incluse dans ce sous-programme.

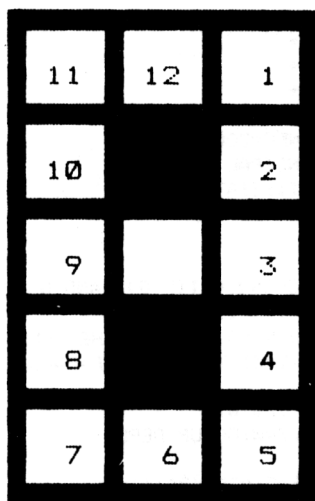
16. Horloge puzzle

Le principe de ce jeu inventé par Czes Kosniowski et John Ewing est celui d'un puzzle dont les pièces peuvent « glisser ».

Il y a sur un tableau douze cases, numérotées de 1 à 12, et un espace vide (voir ci-dessous).

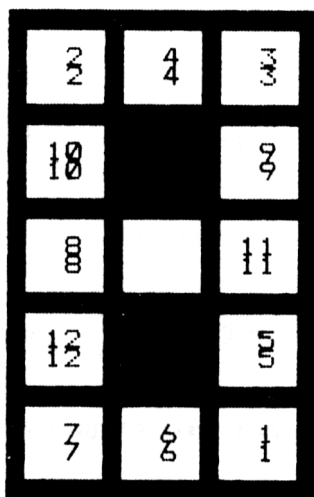


Le but est de déplacer les chiffres contenus dans les cases pour qu'ils apparaissent comme sur une horloge.



Les curseurs sont utilisés pour déplacer l'espace vide sur le tableau.

Le puzzle est très amusant, provoque parfois la fureur ; il n'est certainement pas facile !



11	12	1
10	HORLOGE	2
9	HORLOGE	3
8	HORLOGE	4
7	6	5

```

10 REM *****
20 REM * HORLOGE PUZZLE *
30 REM *****
40 REM
50 DEPL=500: IHITI=1000: INFO=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO: GOSUB IHITI: STATE=FALSE
120 REPEAT
130 : ANSWER=FALSE
140 : REPEAT
150 : REPEAT: GET G$: G=ASC(G$): UNTIL G>7 AND G<12 OR G=81
160 : U=X+(G=8)-(G=9): V=Y+(G=11)-(G=10)
170 : IF U>=0 AND U<3 AND V>=0 AND V<5 THEN ANSWER=TRUE
180 : IF U=1 AND V=1 OR V=3 THEN ANSWER=FALSE
190 UNTIL ANSWER
200 : W=5*U+V: P$(Z)=P$(W): P$(W)=" "
210 : GOSUB DEPL: X=U: Y=V: Z=W: GOSUB DEPL
220 : STATE=TRUE
230 : FOR I=0 TO 14: STATE=STATE*(P$(I)=Q$(I)): NEXT
240 : IF G=81 THEN STATE=2
250 UNTIL STATE
260 IF STATE<2 THEN PING: WAIT 30: PING:
270 PLOT 24,1,CHR$(12)+CHR$(2)+"Autre partie?"
280 PLOT 26,3,CHR$(12)+CHR$(2)+"O ou N"
290 REPEAT: GET G$: UNTIL G$="O" OR G$="N"
300 IF G$="O" THEN RUN
310 CLS: POKE #26A,3: PRINT "A bientot.": END
500 REM ** DEPLACEMENT **
510 PLOT 7+5*X, 2+5*Y, P$(Z): PLOT 7+5*X, 3+5*Y, P$(Z)
520 RETURN
1000 REM ** IHIInitialisation **
1010 E$=CHR$(27): F$=CHR$(255)
1020 PAPER 0: INK 7: CLS
1030 B$=" "+E$+"D"+F$: X$=E$+"B"+B$
1040 A$=B$+X$+X$+X$: Y$=B$
1050 FOR I=0 TO 4: Y$=Y$+F$: NEXT
1060 C$=B$+E$+"B"+Y$+X$
1070 FOR I=0 TO 14: B$=B$+F$: NEXT
1080 REM IMPRESSION TABLEAU
1090 PRINT CHR$(30)+B$
1100 FOR I=0 TO 4: FOR J=1 TO 5
1110 : IF 2*INT(I/2)=I AND J<5 THEN PRINT A$
1120 : IF 2*INT(I/2)<>I AND J<5 THEN PRINT C$
1130 : IF J=5 THEN PRINT B$
1140 NEXT: NEXT
1150 R$=CHR$(1): PLOT 25,6,R$+"11 12 1"
1160 : PLOT 25,9,R$+"10 HEURES 2": PLOT 25,12,R$+" 9 HEURES 3"
1170 PLOT 25,15,R$+" 8 HEURE 4"
1180 PLOT 25,18,R$+" 7 6 5"
1190 REM LES PIECES
1200 DIM P$(14), Q$(14)
1210 FOR I=1 TO 9: P$(I)=" "+CHR$(48+I): NEXT
1220 FOR I=0 TO 2: P$(10+I)="1"+CHR$(48+I): NEXT

```

```

1230 P$(0)="  ":P$(13)=F$+F$
1240 FOR I=0 TO 14:READ K:Q$(I)=P$(K):NEXT
1250 FOR I=0 TO 14:P$(I)=Q$(I):NEXT
1260 REM MELANGE
1270 FOR I=1 TO 10
1280 : K=INT(RND(1)*6)+9*INT(RND(1)*2)
1290 : L=INT(RND(1)*6)+9*INT(RND(1)*2)
1300 : R=P$(K):P$(K)=P$(L):P$(L)=R$
1310 NEXT
1320 REM AFFICHAGE DES PIECES
1330 FOR X=0 TO 2:FOR Y=0 TO 4
1340 : Z=5*X+Y:GOSUB DEPL
1350 NEXT:NEXT
1360 X=1:Y=2:Z=7
1370 RETURN
1800 DATA 11,10,9,8,7,12,13,0,13,6,1,2,3,4,5,
2000 REM ** INFOrmation **
2010 E$=" "+CHR$(27):PAPER 0:INK 7:CLS:PRINT CHR$(4)
2020 PRINT E$+"J";SPC(8);"HORLOGE PUZZLE":PRINT:PRINT
2030 PRINT:PRINT E$+"J";SPC(8);"de Czes Kosniowski"
2040 PRINT CHR$(4):PRINT
2050 PRINT "Remettez les nombres de l'horloge dans";
2060 PRINT "leur position exacte."
2070 PRINT "Utilisez les curseurs pour deplacer la";
2080 PRINT "case vide.";E$+"ATapez Q pour sortir":PRINT:PRINT
2090 PRINT E$+"L"+E$+"BTapez O pour jouer"
2100 POKE #26A,2'Curseur sans effet
2110 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$="O"
2120 RETURN

```

PROGRAMME PRINCIPAL

Le programme principal est constitué de deux parties. La première attend que vous pressiez un des curseurs (code ASCII 8 à 11) ou la lettre Q (code ASCII 81). Si vous pressez un des curseurs et si ce dernier indique une position correcte pour la case vide, alors le déplacement est réalisé. Si vous pressez Q le jeu s'arrête. La seconde partie teste les nombres pour savoir s'ils sont dans la position correcte.

SOUS-PROGRAMMES

500 DEPLACEMENT

Il affiche simplement les nombres (ou l'espace vide) aux positions exactes.

1000 INITIalisation

C'est un sous-programme assez long qui engendre l'affichage de l'horloge. De plus, l'ordinateur mélange les pièces pour le jeu.

Remarque :

Si vous décidez de changer le nombre de fois que l'ordinateur mélange les pièces, soyez sûr que ce nombre soit encore pair. Si ce nombre est impair, le puzzle devient impossible à résoudre.

2000 INFOrmations

Ce sous-programme contient une brève description du jeu.

17. Les fenêtres

Neuf fenêtres sont numérotées de 1 à 9. Les rideaux en sont ouverts. Une femme passe derrière ces fenêtres une lampe à la main. Elle va de la fenêtre 1 à 9 puis revient à la fenêtre 1 et ainsi de suite. Si vous pressez le numéro de la fenêtre où se trouve la femme, alors les rideaux se ferment. Le but du jeu est de fermer tous les rideaux.

Remarque :

Il y a une petite astuce dans le programme qui vous permet de tricher — les détails ne seront pas donnés ici...

111 222
111 222

444 555
444 555

777
777
777

999
999
999

LES FENÊTRES

Pressez la bonne touche au bon moment

```

10 REM *****
20 REM * LES FENETRES *
30 REM *****
40 REM
50 BLANC=500: IHIT=1000: INFO=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO: GOSUB IHIT
120 REPEAT
130 : PLOT X(C),Y,Z(C)
140 : C=C+1: IF C>9 THEN C=1: WAIT 1+RND(1)*20
150 : PLOT X(C),Y,W(C): WAIT ZZ
160 : IF KEY$=CHR$(Z(C)) THEN GOSUB BLANC
170 UNTIL STATE
180 PING: WAIT 20: PING
190 PRINT: PRINT: PRINT E$+"L   Autre partie? 0 ou N"
200 REPEAT: GET G$: UNTIL G$="0" OR G$="N"
210 IF G$="0" THEN RUN
220 POKE #26A,3: CLS: INK 7: PRINT "A la prochaine.": END
500 REM ** BLANC **
510 Z(C)=32
520 FOR I=-1 TO 1: PLOT X(C)-1,Y+I,"   ": NEXT
530 W(C)=32: MM=MM+1: IF MM>8 THEN STATE=TRUE
540 RETURN
1000 REM ** INITIALISATION **
1010 PAPER 0: INK 2: CLS
1020 PRINT: PRINT: PRINT:
1030 FOR I=1 TO 9: X(I)=4*I-1: Z(I)=48+I: W(I)=255: NEXT
1040 FOR J=0 TO 2: FOR I=1 TO 9
1050 : FOR K=0 TO 2: PRINT CHR$(Z(I));: NEXT
1060 : PRINT " ";
1070 NEXT: PRINT: NEXT
1080 Y=5: Z=255: C=1: MM=0
1090 POKE #26A,2 'Curseurs sans effet
1100 PLOT 13,11,CHR$(4)+"LES FENETRES"
1110 PLOT 0,13,"Pressez la bonne touche au bon moment"
1120 STATE=FALSE
1130 RETURN
2000 REM ** INFORMATION **
2010 PAPER 0: INK 7: CLS: PRINT CHR$(4)
2020 E$=" "+CHR$(27)
2030 PRINT E$+"J";SPC(9);"LES FENETRES": PRINT: PRINT
2040 PRINT: PRINT E$+"J";SPC(5);"par Czes Kosniowski"
2050 PRINT CHR$(4): PRINT: PRINT
2060 PRINT "Fermez les rideaux en"
2070 PRINT "pressant la bonne touche"
2080 PRINT " au bon moment."
2090 PRINT: PRINT "A quelle vitesse voulez vous jouer?"
2100 PRINT "0 (rapide) a 9 (lent) ";
2110 REPEAT: G$=KEY$: GG=RND(1): UNTIL G$>"/" AND G$<":
2120 ZZ=VAL(G$)+1
2125 INIT=1000
2130 RETURN

```

PROGRAMME PRINCIPAL

La boucle principale regarde si vous tapez la bonne touche.

SOUS-PROGRAMME

500 BLANC

Il dessine les rideaux des fenêtres, *i.e.* des blancs remplacent l'affichage. Le rideau C est fermé si $W(C) = 32$. Il est ouvert si $W(C) = 255$. La raison est que CHR\$ (32) est un carré blanc alors que CHRS (255) est un carré de couleur.

1000 INITialisation

La variable X(I) contient la position de la fenêtre I. La variable W(I) contient l'information sur le fait que le rideau de la fenêtre I est ouvert ou non.

2000 INFOrmation

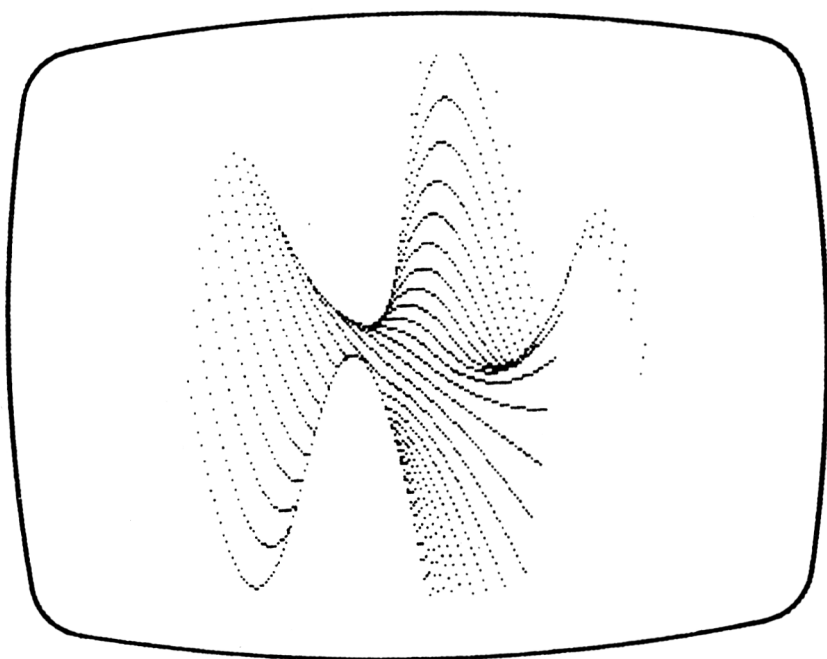
Ce sous-programme affiche une brève description du jeu. Il demande de plus au joueur de choisir la vitesse du jeu.

18. Surfaces

C'est un court programme qui réalise des dessins de surfaces en trois dimensions d'une grande netteté.

Remarque :

Le temps global est assez long mais le résultat vaut la peine d'attendre. Si vous le désirez, vous pouvez garder le dessin sur cassette afin de ne pas avoir à tout recommencer à chaque fois. Voir l'annexe 1 pour plus de détails.



```

10 REM *****
20 REM * SURFACES *
30 REM *****
40 REM
50 PL=500
100 REM ** PROGRAMME PRINCIPAL **
110 INK 6:PAPER 0:CLS:FOR I=0 TO 5:READ A$(I):NEXT
120 REPEAT
130 : TEXT:CLS:PRINT:ZZ=FRE(1)
140 : E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
150 : XMID=119:YMID=99:PP=2*PI:S=SQR(2)/2:SS=XMID*S
160 : PRINT D$;SPC(11);"SURFACES":PRINT D$
170 PRINT:PRINT
180 : PRINT D$;SPC(6);"de Czes Kosniowski"
190 PRINT D$:PRINT
200 : FOR I=0 TO 5:PRINT I:A$(I):NEXT
210 : PLOT 28,11,"ou":PLOT 24,13,"R=SQR(X*X+Y*Y)"
220 : PRINT:PRINT "Donnez votre selection. ";
230 : REPEAT:GET G$:UNTIL G$>="0" AND G$<"7"
240 : G=VAL(G$):IF G>0 THEN GOSUB PL
250 UNTIL G=0
260 CLS:PRINT "A bientot.":END
500 REM ** Plot (affichage) **
510 Hires:POKE #26A,2
520 PRINT,,A$(G)
530 FOR A=-SS TO SS+5*S
540 : MAX=-YMID:TT=SS+A-10*S*INT((A+ABS(A))/10/S)
550 : FOR B=-TT TO TT+S*4 STEP 10*S
560 : X=S*(A+B):Y=S*(B-A):Z=B:R=SQR(X*X+Y*Y)
570 : IF G=1 THEN Z=12*COS(R/5)+B
580 : IF G=2 THEN Z=YMID*EXP(-R*R/600)+B
590 : IF G=3 AND R<>0 THEN Z=2*XMID*SIN(R/5)/R+B
600 : IF G=4 THEN Z=12*COS(X/10)*COS(Y/10)+B
610 : IF G=5 AND R<>0 THEN Z=X*Y*(X-Y)*(X+Y)/(1000*R)+B
620 : FB=1:IF Z<MAX THEN FB=3 ELSE MAX=Z
630 : U=XMID+A:V=YMID-Z*S
640 : IF V>1 AND V<2*YMID THEN CURSET U,V,FB
650 : NEXT
660 NEXT
670 PING:FOR I=1 TO 6:INK I:WAIT 75:NEXT:E$=" "+CHR$(27)
680 PRINT E$+"B"+E$+"LTapez Y pour continuer"
690 REPEAT:UNTIL KEY$=""
700 REPEAT:GET G$:UNTIL G$="Y"
710 RETURN
800 DATA FIN ,Z = COS(R),Z = EXP(-R*R)
810 DATA Z = SIN(R)/R,Z = COS(X)*COS(Y)
820 DATA La selle de singe

```

PROGRAMME PRINCIPAL

Le but principal du programme y est présenté. On vous demande quelle surface vous voulez dessiner.

SOUS-PROGRAMME

500 PL

Il affiche (PLOT) la surface désirée. L'effet de perspective du dessin est obtenu grace aux variables A,B ainsi que X et Y. Les points cachés ne sont pas imprimés ; ceci est réalisé en comparant la valeur du point à afficher avec MAX.

19. Le Danube bleu

C'est une simple mélodie en référence à Strauss. Reportez-vous à l'annexe 4 pour les notes de musique.

```
10 REM *****
20 REM * DANUBE BLEU *
30 REM *****
40 REM
50 INFO=500
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:J=3:K=0
120 REPEAT
130 : PLOT 6+A(I),26-A(I),32
140 : I=I+1:IF I>MAX THEN I=0:J=J+1:IF J>5 THEN J=3
150 : B=INT((A(I)-1)/12)
160 : PLOT 6+A(I),26-A(I),14+J
170 : IF A(I)>0 THEN MUSIC 1,J+B,A(I)-12*B,0:PLAY 1,0,1,L
180 : WAIT B*T(I)
190 UNTIL KEY$="Q"
200 PLAY 0,0,0,0
210 PRINT E$;"B";E$;"L Autre partie? O ou N";SPC(10)
220 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
230 IF G$="O" THEN RUN
240 CLS:PRINT "A bientôt.":POKE #26A,3:INK 7:END
500 REM ** INFOrmation **
510 INK 5:PAPER 0:CLS:DEF FNR(X)=INT(RND(1)*X)
520 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4):PRINT
530 PRINT D$:SPC(8);"DANUBE BLEU":PRINT D$:PRINT:PRINT
540 PRINT D$:SPC(6);"De Czes Kosniowski":PRINT D$:PRINT
550 PRINT "Une simple melodie.....":PRINT
560 PRINT "Tapez O pour commencer.";
570 REPEAT:GET G$:UNTIL G$="O"
580 PRINT:PRINT:PRINT "Tapez Q pour sortir.";
590 DIM A(100),T(100):MAX=-1
600 REPEAT:MAX=MAX+1:READ A(MAX),T(MAX):UNTIL A(MAX)=-1
610 IF MAX<>69 THEN PRINT "Erreur dans les DATA":END
620 MAX=MAX-1:CLS:POKE #26A,10:INK 6:L=2500
630 PRINT:PRINT D$:SPC(8);"DANUBE BLEU":PRINT D$
640 RETURN
```

```

700 DATA 1,2,1,2,5,2,8,2,8,4,20,2,20,4,17,2,17,4
710 DATA 1,2,1,2,5,2,8,2,8,4,20,2,20,4,18,2,18,4
720 DATA 3,2,3,2,6,2,10,2,10,4,22,2,22,4,18,2,18,4
730 DATA 3,2,3,2,6,2,10,2,10,4,22,2,22,4,17,2,17,4
740 DATA 1,2,1,2,5,2,8,2,13,4,25,2,25,4,20,2,20,4
750 DATA 1,2,1,2,5,2,8,2,13,4,25,2,25,4,22,2,22,4
760 DATA 3,2,3,2,6,2,10,2,10,7,7,2,8,2,17,8
770 DATA 13,2,5,2,5,4,3,2,10,4,8,2,1,6
780 DATA -1,-1

```

PROGRAMME PRINCIPAL

Il joue chaque note pendant le temps voulu. En même temps, la hauteur du son est représentée graphiquement. A chaque fois que la mélodie est jouée, la clef est changée.

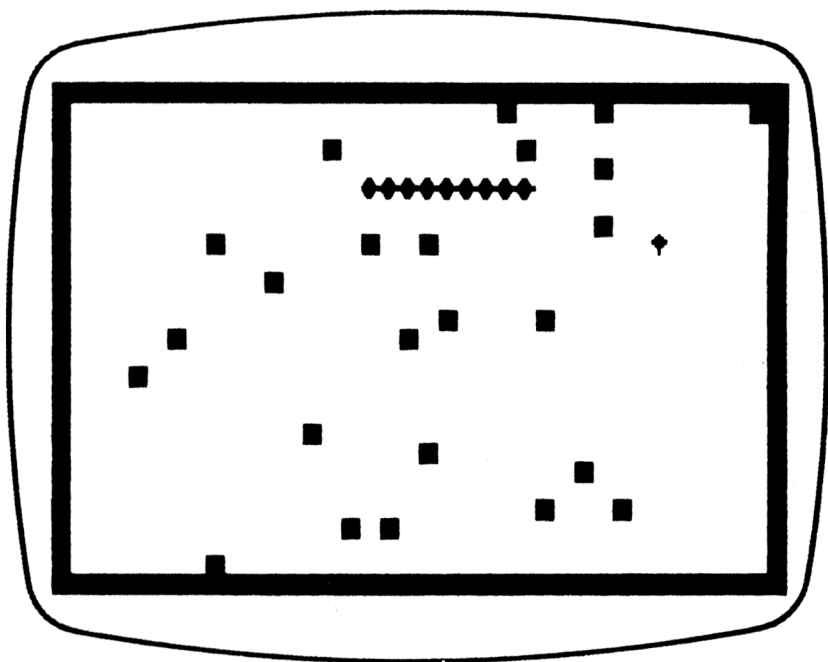
SOUS-PROGRAMME

500 INFORMATION

Les renseignements sont donnés ici. Lorsqu'ils sont terminés le programme lit les données nécessaires, c'est-à-dire les notes et les intervalles de temps.

20. Le serpent

Ce programme est un jeu d'action dans lequel vous devez contrôler une créature (un serpent) en utilisant les curseurs. Essayez de faire manger la nourriture par le serpent mais évitez les rochers et le mur (le bord de l'écran). De plus, ne vous détruisez pas vous-même. Chaque fois que le serpent mange un morceau, il devient plus long et se déplace d'une façon plus posée. Le succès est atteint lorsque le serpent est long de dix unités. C'est assez prenant que d'essayer de nourrir ce serpent en pleine croissance mais toujours affamé.



```

10 REM *****
20 REM * LE SERPENT *
30 REM *****
40 REM
50 TEST=400:DIRECT=500:NOUR=600
60 IHITI=1000:CAR=1200:INFO=2000
100 REM ** PROGRAMME PRINCIPALE **
110 GOSUB INFO:GOSUB IHITI:GOSUB NOUR
120 REPEAT
130 : K$=KEY$:IF K$<>"" THEN GOSUB DIRECT
140 : FOR I=L TO 1 STEP -1:P(I)=P(I-1):Q(I)=Q(I-1):NEXT
150 : P(0)=P(0)+DP:Q(0)=Q(0)+DQ:PQ=SCRN(P(0),Q(0))
160 : PLOT P(0),Q(0),R:PLOT P(L),Q(L),32
170 : IF PQ<>32 THEN GOSUB TEST
180 UNTIL STATE OR L>10
190 IF L>10 THEN PLOT 15,0,"BIEN JOUE":PING:WAIT 10:PING
200 IF STATE THEN PLOT 17,0,"AIE!!":PAPER 1
210 WAIT 20:PAPER 0:PRINT
220 PRINT E$+"L":SPC(8):"Autre partie? O ou N":SPC(10)
230 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
240 IF G$="O" THEN RUN
250 CLS:PRINT "A bientot.":POKE #26A,3:INK 7:END
400 REM ** TEST **
410 IF PQ=R THEN SHOOT:STATE=1
420 IF PQ=W THEN EXPLODE:STATE=2
430 IF PQ=C THEN ZAP:GOSUB NOUR
440 RETURN
500 REM ** DIRECTION **
510 KK=ASC(K$):IF KK<3 OR KK>11 THEN RETURN
520 DP=(KK=8)-(KK=9):DQ=(KK=11)-(KK=10)
530 RETURN
600 REM ** NOURriture **
610 FOR I=0 TO 3
620 REPEAT:A=FNR(36)+2:B=FNR(25)+1:UNTIL SCR(N,A,B)=32
630 IF I=0 THEN PLOT A,B,C:L=L+1
640 IF I>0 THEN PLOT A,B,W
650 NEXT
660 RETURN
1000 REM ** INITIALISATION **
1010 PAPER 0:INK 2:CLS:R=37:DIM P(20),Q(20):W=255
1020 FOR X=2 TO 38:PLOT X,0,W:PLOT X,26,W:NEXT
1030 FOR Y=1 TO 25:PLOT 2,Y,W:PLOT 38,Y,W:NEXT
1040 P(0)=30:P(1)=31:P(2)=32:Q(0)=22:Q(1)=22:Q(2)=22
1050 L=1:FOR I=0 TO L:PLOT P(I),Q(I),R:NEXT
1060 DP=-1:DQ=0:C=38:START=#B400+R*8:AA=#A000
1070 POKE AA,2:IF PEEK(AA)<>2 THEN START=START-#B000
1080 GOSUB CAR:START=START+8:GOSUB CAR
1090 STATE=0
1100 RETURN
1200 REM ** CARactere **
1210 FOR I=START TO START+7:READ X:POKE I,X:NEXT
1220 RETURN
1300 DATA 12,12,30,63,63,30,12,12

```

```

1310 DATA 21,21,14,14,4,4,4,4
2000 REM ** INFORMATION **
2010 INK 6:PAPER 0:CLS:DEF FNR(X)=INT(RND(1)*X)
2020 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4):PRINT
2030 PRINT D$;SPC(8);"LE SERPENT":PRINT D$:PRINT:PRINT
2040 PRINT D$;SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
2050 PRINT "Guidez le serpent vers la nourriture.":PRINT
2060 PRINT "Mais evitez les obstacles et le mur."
2070 PRINT "De plus ne vous autodetruisez pas!":PRINT
2080 PRINT "Utilisez les curseurs pour changer de"
2090 PRINT"direction.":PRINT
2100 PRINT "Tapez 0 pour partir.....";
2110 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$="0"
2120 POKE #26A,2
2130 RETURN

```

LE PROGRAMME PRINCIPAL

Le serpent se déplace jusqu'à ce qu'il rencontre un obstacle. En utilisant les curseurs, vous pouvez changer la direction de son déplacement. Le programme principal balaye le clavier et change la direction du serpent si on lui en donne l'ordre. Il regarde aussi si le serpent a heurté un obstacle.

LES SOUS-PROGRAMMES

400 TEST

Si la tête du serpent arrive dans un endroit qui n'est pas vide (code ASCII 32), ce sous-programme est utilisé. Le serpent peut soit heurter le mur ou un rocher (si PQ = W), soit s'autodétruire (si PQ = R) ou enfin manger de la nourriture (si PQ = C).

500 DIRECTION

Ce sous-programme entre en jeu si une touche est pressée. Si l'un des curseurs est actionné, la direction correspondante du mouvement est calculée.

600 NOURriture

Ce sous-programme place sur l'écran non seulement la nourriture mais aussi deux rochers.

1000 INITialisation

Il dessine le mur d'enceinte et crée deux caractères spéciaux (pour le serpent et la nourriture).

1200 CARactère

C'est un court sous-programme qui lit les données et les plante aux bons endroits pour des caractères spéciaux.

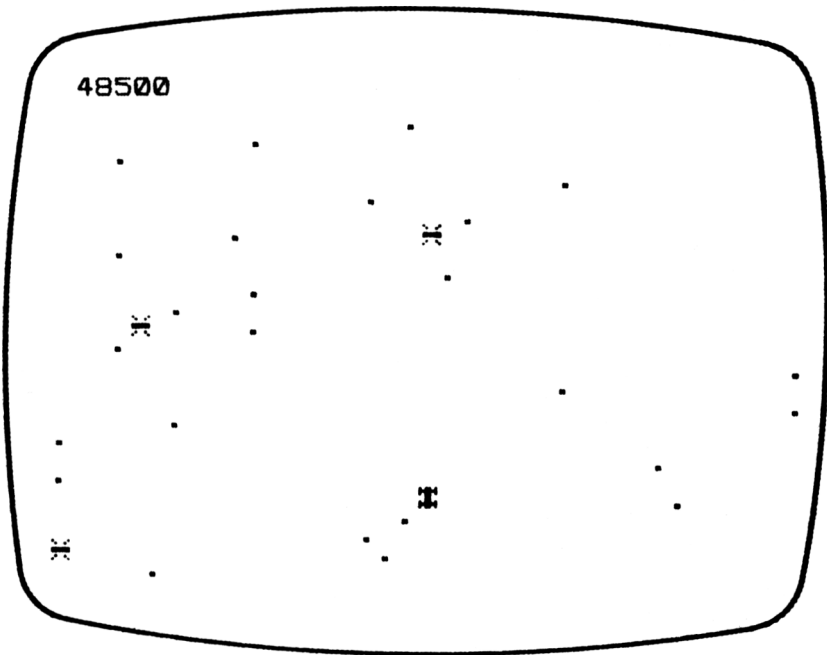
2000 INFOrmation

Ce sous-programme contient une brève description du jeu.

21. Poursuite spatiale

Vous êtes aux commandes d'un vaisseau spatial qui plonge dans le vide interplanétaire. Vous manquez de carburant et vous êtes obligé de diriger votre vaisseau vers les sources d'énergie qui passent parfois non loin de vous. Ces sources d'énergie ressemblent à des astérisques.

Une récente explosion de planète a laissée de nombreux débris dans l'espace, vous devez les éviter. Votre vaisseau dispose d'une grande puissance qui vous permet de le diriger avec les curseurs gauche et droit.



```

10 REM *****
20 REM * POURSUITE SPATIALE *
30 REM*****
40 REM
50 CHOC=500: IHITI=1000: INSTRUCT=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INSTRUCT:GOSUB IHITI
120 FOR I=1 TO 26:PRINT:NEXT: REPEAT
130 : PLOT 39*RND(1), 26, 38
140 : FOR I=1 TO 20
150 : PLOT 39*RND(1), 26, 46:PLOT X, Y, 32:PRINT
160 : IF X>0 AND PEEK(769)=223 THEN X=X-1
170 : IF X<38 AND PEEK(769)=127 THEN X=X+1
180 : XY=SCRN(X,Y):IF XY=46 OR XY=38 THEN GOSUB CHOC
190 : PLOT X,Y,37
200 : NEXT
210 UNTIL COUNT>=MAX
220 CLS
230 PRINT CHR$(4)
240 PRINT E$+"J"+E$+"B";SPC(8);"VOTRE SCORE: ";SC
250 PRINT:PRINT:PRINT:WAIT 200
260 PRINT E$+"N"+E$+"A Autre partie? 0 ou N "
270 PRINT CHR$(4)
280 REPEAT:GET K$:UNTIL K$="0" OR K$="N"
290 IF K$="0" THEN RUN
300 POKE #26A,3:CLS:PRINT "A bientot.":END
500 REM ** CHOC **
510 IF XY=46 THEN EXPLODE:PAPER 1:WAIT 20:COUNT=COUNT+1
520 POKE TP+39,48+MAX-COUNT
530 IF XY=38 THEN PING:PAPER 2:WAIT 20:PING:SC=SC+250
540 PAPER 3:WAIT 20:PAPER 0
550 IF RND(1)<.5 THEN PLOT 39*RND(1),0,38
560 FOR I=2 TO LEN(STR$(SC))
570 : POKE TP+I-1,48+VAL(MID$(STR$(SC),I,1))
580 NEXT
590 RETURN
1000 REM ** INITIALisation **
1010 X=18:Y=0:COUNT=0:SC=0:AA=#A000:Q=0
1020 POKE AA,2:IF PEEK(AA)<>2 THEN Q=#B000
1030 START=46080+37*B-Q:TP=#BB80-Q
1040 FOR I=0 TO 7:READ J:POKE START+I,J:NEXT
1050 FOR I=0 TO 7:READ J:POKE START+8+I,J:NEXT
1060 FOR I=36 TO 39:POKE TP+I,32:NEXT
1070 POKE TP+39,48+MAX:INK 7:PAPER 0:CLS
1080 RETURN
1100 DATA 45,63,45,12,12,45,63,45
1110 DATA 33,18,12,63,63,12,18,33
2000 REM ** INSTRUCTIONS **
2010 E$=" "+CHR$(27):CLS:POKE #26A,10
2020 INK 7:PAPER 1:CLS:PRINT CHR$(4)
2030 PRINT E$+"J"+E$+"A";SPC(4);"POURSUIVE SPATIALE"
2040 PRINT:PRINT:PRINT:WAIT 20
2050 PRINT E$+"J"+E$+"D de Czes Kosniowski"

```

```

2060 PRINT CHR$(4):PRINT
2070 PRINT:PRINT "Pilotez votre vaisseau spatial vers "
2080 PRINT "les sources d'énergie - mais faites"
2090 PRINT "attention aux débris cosmiques....."
2100 PRINT:PRINT "Utilisez les curseurs gauche et droit"
2110 PRINT "pour vous déplacer.    BONNE CHANCE"
2120 PRINT:PRINT
2130 PRINT E$+"àNombre de chocs autorisés?"
2140 PRINT E$+"àTapez un nombre entre 1 et 9."
2150 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$>"0" AND G$<":"
2160 MAX=VAL(G$):PING:WAIT 30:PING
2170 RETURN

```

PROGRAMME PRINCIPAL

Les lignes montent sur l'écran une par une grâce à la commande PRINT. Des débris sont déposés sur chaque ligne créée par PLOT 39*RND(1), 26, 38.

PEEK (769) lit le clavier. Si le curseur gauche est pressé, il a la valeur 223 (les autres touches ont aussi la même valeur). Le curseur droit a la valeur 127. L'utilisation de PEEK (769) est plus rapide que celle de de KEYS.

Tout ceci continue jusqu'au moment où le vaisseau rencontre un débris cosmique.

SOUS-PROGRAMMES

500 CHOC

Ce sous-programme entre en jeu si le vaisseau heurte un objet. Il modifie le nombre de «crashes» ou de sources d'énergie déjà utilisées en fonction de l'objet rencontré. Le score est affiché sur la première ligne de l'écran, généralement utilisée pour les messages.

1000 INITIalisation

Il initialise les compteurs et crée deux caractères spéciaux : le vaisseau spatial et les sources d'énergie.

2000 INSTRUCTIONS

Il donne une courte description du jeu. Il vous demande aussi combien de «crashes» vous vous permettez avant la fin du jeu.

22. Epelez-le

Dans ce jeu, un mot clignote un court moment sur l'écran. Regardez-le attentivement car vous allez devoir l'épeler. Pour vous aider, vous pouvez choisir la vitesse de disparition du mot de l'écran.

EPELEZ-LE

MOT 8

Tapez le mot et pressez <return>

? TIMONNIER

Non, la réponse est TIMONIER

Pressez Y pour continuer

```

10 REM *****
20 REM * EPELEZ LE *
30 REM *****
40 REM
50 EPEL=500: IHITI=1000: INFO=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO: GOSUB IHITI
120 FOR I=1 TO 10
130 : A$=R$(INT(RND(1)*MAX))
140 : GOSUB EPEL: PING: WAIT 50: PING: PRINT: PRINT
150 : PRINT SPC(16-LEN(A$)/2); D$+A$
160 : PRINT D$: WAIT T: GOSUB EPEL: PRINT
170 : PRINT: PRINT "Tapez le mot puis <return>"
180 : PRINT: PRINT SPC(13);: INPUT B$
190 : PRINT: PRINT: PRINT: PRINT
200 : IF A$=B$ THEN PING: SC=SC+1: PLOT 14,12,"BIEN VU"
210 : IF A$<>B$ THEN SHOOT: PLOT 8,12,N$+A$
220 : PRINT SPC(9); "Tapez 0 pour continuer";
230 : REPEAT: GET G$: UNTIL G$="0"
240 NEXT
250 GOSUB EPEL: PLOT 15,3," "
260 PRINT SPC(6); "Vous avez obtenu "; SC; " sur 10."
270 PRINT: PRINT E$+"L"+E$+"B Autre partie? 0 ou N"
280 REPEAT: GET G$: UNTIL G$="0" OR G$="N"
290 IF G$="0" THEN RUN
300 CLS: PRINT "A bientôt.": END
500 REM ** EPEllation **
510 CLS: PRINT: PRINT S$: PRINT D$+E$+"A": PRINT: PRINT
520 PLOT 15,3, CHR$(8)+"MOT"+STR$(I)
530 RETURN
1000 REM ** IHITIialisation **
1010 MAX=-1: DIM R$(100)
1020 REPEAT: MAX=MAX+1: READ R$(MAX): UNTIL R$(MAX)="*"
1030 N$="Non, la reponse est "
1040 RETURN
1100 DATA TABLE, CHAISE, LIVRE, OREILLER, CUBE, TAPIS, ILE
1110 DATA VERRE, VOLUME, TORCHE, CUISINE, BOUTEILLE, TISSU
1120 DATA DORMIR, COURIR, MANQUE, VOLEUR, TIMONIER, PERSONNE
1130 DATA PLASTIQUE, BANANE, GANT, BALLE, DEBUT, DEPUTE
1140 DATA VIANDE, RESINE, MISERE, DIRE, FLEUR, TABLEAU
1150 DATA PLATEAU, BOITE, DIABLE, JOLIE, ENFIN, EGLISE
1160 DATA COUPABLE, SOLDAT, LUNDI, MARDI, MERCREDI
1170 DATA CHAMBRE, SALUT, SOLEIL, PAPIER, CHAINE, VINGT
1180 DATA JEUDI, VERTICAL, SELECTION, CIRQUE, THEATRE
1190 DATA MAGIQUE, LETTRE, JUNGJE, INNOCENT, INJURE
1200 DATA INFLUENCE, INCIDENT, BOMBE, ATOME, GRAND
1210 DATA CHEMISE, VALISE, TRIVIAL, DEBUT, GENERAL
1220 DATA PEUPLE, TRANQUILLE, PROPRIETE, POLICE, POISON
1230 DATA *
2000 REM ** INFORMATION **
2010 PAPER 0: INK 7: E$=" "+CHR$(27): D$=E$+"J"+CHR$(4)
2020 S$=D$+E$+"A EPELEZ LE"
2030 GOSUB EPEL: PLOT 15,3," "

```

```

2040 PRINT D$+E$+"A      de Czes Kosniowski"
2050 PRINT D$:PRINT:PRINT:PRINT
2060 PRINT "Un mot va clignoter sur l'ecran."
2070 PRINT "Regardez le bien puis epelez le...."
2080 PRINT:PRINT "Quelle vitesse? 1 (rapide) a 9 (lent)";
2090 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$>"0" AND G$<":
2100 T=10*VAL(G$)
2110 PRINT:PRINT:PRINT SPC(10);"O.K. - soyez pret."
2120 RETURN

```

PROGRAMME PRINCIPAL

Un mot est choisi au hasard parmi ceux contenus dans les données. Deux PINGS après, le mot est affiché sur l'écran (en caractères de hauteur double). L'ordinateur attend alors que vous tapiez le mot. Ceci est suivi d'un commentaire approprié. Après dix questions, votre score est affiché.

SOUS-PROGRAMMES

500 EPELez

Il est utilisé pour nettoyer l'écran après l'affichage d'un mot.

1000 INITIalisation

Les mots sont rangés dans les variables chaîne R\$(I). Vous pouvez, si vous le désirez, ajouter des mots à la liste. Cependant, si le nombre total de mots dépasse 100, changez la DIMENSION à la ligne 100.

2000 INFOrmation

Il contient une brève description du programme. Il est aussi utilisé pour déterminer la vitesse de disparition des mots de l'écran.

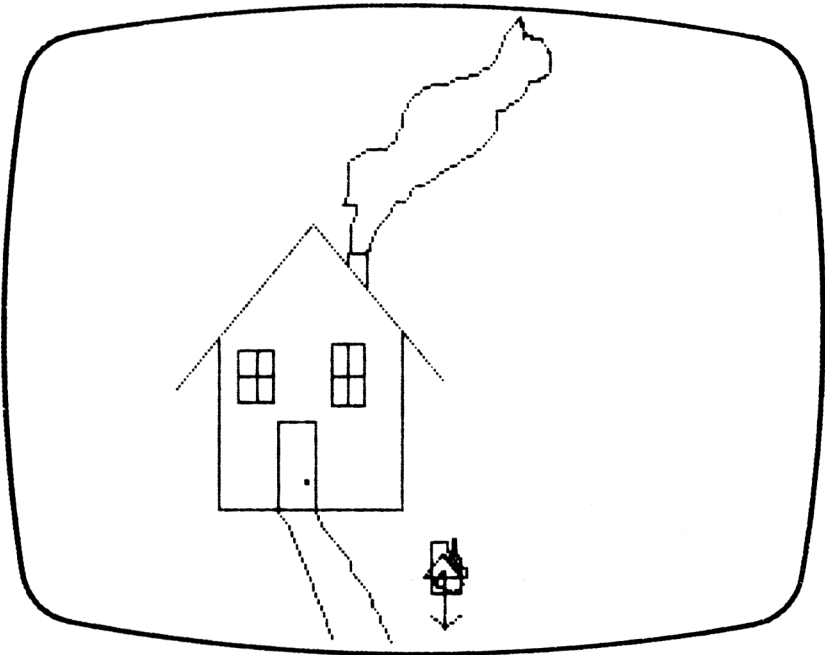
23. Aide au dessin

Exercez vos capacités artistiques grâce à ce programme d'aide au dessin, simple mais néanmoins efficace. Vous pouvez dessiner des lignes horizontales, verticales ou en diagonales. Les lignes que vous dessinez peuvent être visibles ou invisibles, selon votre choix.

Utilisez les curseurs pour tracer les lignes verticales et horizontales. Pour dessiner en diagonale, utilisez les touches suivantes :

- I pour en haut à gauche
- \ pour en haut à droite
- Z pour en bas à gauche
- / pour en bas à droite

En tapant la lettre I vous changez la nature (visible ou invisible) de la ligne que vous tracez. Pour quitter le programme, tapez Q. Si vous voulez



garder un dessin sur cassette, vous devrez ajouter quelques lignes supplémentaires au programme. Reportez-vous à l'annexe I pour plus de détails.

```

10 REM *****
20 REM * AIDE AU DESSIN *
30 REM *****
40 REM
50 IHITI=400:INFO=600
100 REM ** PROGRAMME **
110 GOSUB INFO:GOSUB IHITI
120 REPEAT
130 : CURSET X,Y,2
140 : GET G$:G=ASC(G$)
150 : A=(G=8)-(G=9)-(G=92)+(G=49)-(G=47)+(G=90)
160 : B=(G=11)-(G=10)+(G=92)+(G=49)-(G=47)-(G=90)
170 : CURSET X,Y,0 'efface la position actuelle
180 : IF X+A>239 OR X+A<0 THEN A=0:PING
190 : IF Y+B>199 OR Y+B<0 THEN B=0:PING
200 : U=X:V=Y:X=X+A:Y=Y+B
210 : IF G=73 THEN FB=-NOT-FB
220 : IF NOT ((A=0) AND (B=0)) THEN CURSET U,V,FB
230 : Z=FRE(1)
240 UNTIL G=81
250 TEXT:E$=" "+CHR$(27)
260 PRINT E$+"B"+E$+"LAutre partie? O ou N":PRINT
270 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
280 IF G$="O" THEN RUN
290 CLS:PRINT "a bientôt.":POKE #26A,3:END
400 REM ** Initialisation **
410 X=119:Y=99:FB=1:HIRES
420 POKE #26A,10
430 PRINT "Utilisez les curseurs et 1,\,Z ou /."
440 PRINT:PRINT "I - Dessin on/off. Q - Sortie.";
450 RETURN
600 REM ** Information **
610 INK 6:PAPER 0:CLS:PRINT
620 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
630 PRINT D$:SPC(8);"AIDE AU DESSIN":PRINT D$:PRINT:PRINT
640 PRINT D$:SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
650 PRINT "Faites un dessin grace a ce programme.":PRINT
660 PRINT "Utilisez les curseurs pour aller en"
670 PRINT "haut, en bas, a droite ou a gauche."
680 PRINT "Utilisez aussi pour les diagonales,le"
690 PRINT "1, le \, le Z et le / .":PRINT
700 PRINT "Pour faire une ligne invisible, tapez"
710 PRINT "I, tapez le a nouveau pour redessiner."
720 PRINT "Tapez Q pour sortir du programme.":PRINT
730 PRINT E$+"B"+E$+"LTapez Q pour commencer."
740 REPEAT:GET G$:UNTIL G$="O"
750 RETURN

```

PROGRAMME PRINCIPAL

Le programme principal attend que vous pressiez une touche. Si l'une des touches de *mouvement* est pressée, la direction de celui-ci (A, B) est calculée. Si le point (X + A, Y + B) est toujours sur l'écran, il est affiché par PLOT. Si vous tapez I, la valeur de FB passe de 1 à 0 ou de 0 à 1. Ceci détermine si le point affiché est visible ou non.

SOUS-PROGRAMMES

400 INITIALisation

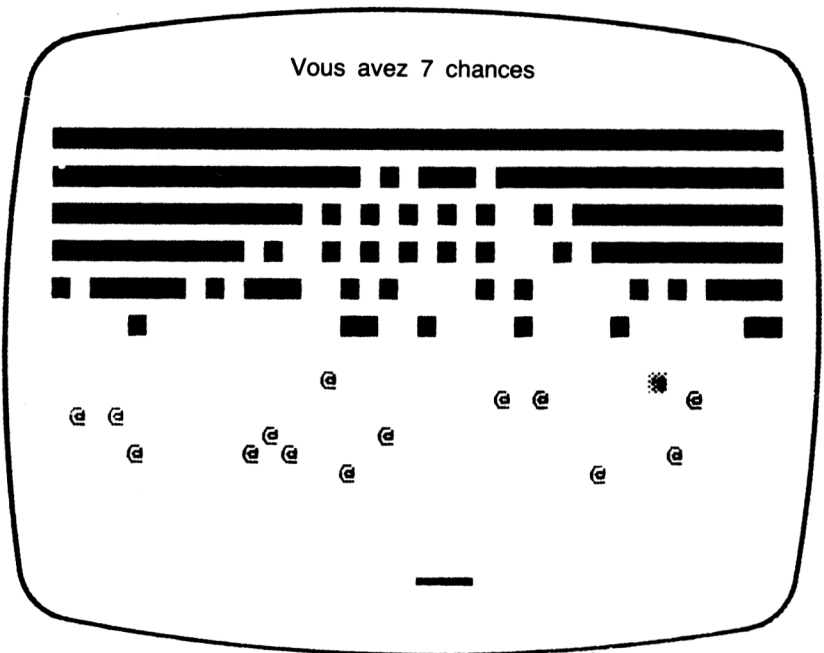
Il initialise les variables et donne des indications sur le rôle de certaines touches du clavier.

600 INFOrmation

Il donne une courte description du programme.

24. Rebondissement

Vous pouvez faire rebondir une balle sur votre raquette, cette balle parcourra alors l'écran. Vous déplacez votre raquette avec les curseurs gauche et droit. L'écran est initialisé avec plusieurs bandes de couleurs constituées de carrés. A chaque fois que la balle rencontre un de ceux-ci, il disparaît. Le but du jeu est d'atteindre avec la balle le sommet de l'écran. La balle rebondit sur les bords de l'écran, sur les carrés de couleur et sur votre raquette. Pour donner un peu de piquant au jeu, des obstacles, qui disparaissent lentement, ont été ajoutés.



```

10 REM *****
20 REM * REBONDISSEMENT *
30 REM *****
40 REM
50 PERD=500:FIN=600:IHITI=1000:INFO=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:GOSUB IHITI
120 REPEAT
130 : PLOT X,Y,32:X=X+DX:Y=Y+DY:PLOT X,Y,126
140 : IF X<3 OR X>37 THEN DX=-DX:PING
150 : V=Y+DY:W=SCRN(X,V)
160 : IF W>32 THEN DY=-DY:PING
170 : IF W=127 THEN PLOT X,V,32
180 : IF V>25 AND W<33 THEN GOSUB PERD
190 : IF V<1 THEN GOSUB FIN
200 : IF Z>0 AND PEEK(769)=223 THEN Z=Z-1:PLOT Z,26,B$
210 : IF Z<35 AND PEEK(769)=127 THEN Z=Z+1:PLOT Z,26,B$
220 UNTIL STATE<1
230 PRINT E$;"B";E$;"L   Autre partie? O ou N"
240 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
250 TEXT:IF G$="O" THEN RUN
260 CLS:PRINT "A bientot.":POKE #26A,3:END
500 REM ** PERDu **
510 EXPLODE:STATE=STATE-1:DY=-DY:WAIT 100
520 PLOT X,Y,32:PLOT Z,26,"   "
530 PLOT 19,0,MID$(STR$(STATE),2)+ " coups.  "
540 IF STATE=1 THEN PLOT 27,0,".  "
550 X=10+FNR(20):Z=X:DX=2*FNR(2)-1
560 PLOT X,Y,126:PLOT Z,26,B$
570 RETURN
600 REM ** FIN **
610 PING:WAIT 50:PING:WAIT 50:PING
620 PLOT13,1,CHR$(6)+CHR$(12)+"BIEN JOUE"
630 WAIT 100:STATE=-1
640 RETURN
1000 REM ** INITIALisation **
1010 CLS:INK 7:PRINT SPC(7);"Vous avez ";
1020 PRINT MID$(STR$(STATE),2);" coups."
1040 A$="":FOR I=1 TO 37:A$=A$+CHR$(255):NEXT
1050 PRINT:FOR I=1 TO SKILL:PRINT CHR$(127+I)+A$:PRINT:NEXT
1060 PLOT 0,26,9:Y=25:X=10+FNR(20):Z=X
1070 B$=CHR$(9)+"###  "
1080 DX=2*FNR(2)-1:DY=-1:PLOT X,Y,126:PLOT Z,26,B$
1090 FOR I=1 TO 15
1100 : PLOT 2+FNR(34),2*SKILL+3+FNR(6),96
1110 NEXT
1120 RETURN
2000 REM ** INFORMATION **
2010 INK 5:PAPER 0:CLS:DEF FNR(X)=INT(RND(1)*X)
2020 E$="  "+CHR$(27):D$=E$+"J"+CHR$(4):PRINT
2030 PRINT D$:SPC(9);"REBONDISSEMENT":PRINT D$:PRINT:PRINT
2040 PRINT D$:SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
2050 PRINT "Utilisez les curseurs droit et gauche"

```

```

2060 PRINT "pour attraper la balle. Les obstacles"
2070 PRINT "disparaissent lentement tant que la "
2080 PRINT "balle n'a pas atteint le sommet.":PRINT
2090 PRINT "Choisissez votre niveau de difficulté.";
2100 PRINT "1 (facile) a 4 (difficile).";
2110 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$>"0" AND G$<"5"
2120 SKILL=2+VAL(G$):STATE=13-SKILL:POKE #26A,10
2130 RETURN

```

LE PROGRAMME PRINCIPAL

Le programme principal déplace la balle et regarde quel obstacle, s'il y en a un, elle rencontre. PEEK (769) lit le clavier. S'il prend la valeur 223 c'est que le curseur gauche a été pressé, la raquette est alors déplacée vers la gauche. S'il vaut 127, elle est déplacée vers la droite.

LES SOUS-PROGRAMMES

500 PERDu

Si la balle passe à côté de la raquette, ce sous-programme émet le son approprié et remet à jour les données.

600 FIN

Si la balle a atteint le sommet de l'écran, ce sous-programme émet le son approprié et affiche BIEN JOUE.

1000 INITIALisation

Il affiche les lignes composées de carrés et les obstacles aléatoires.

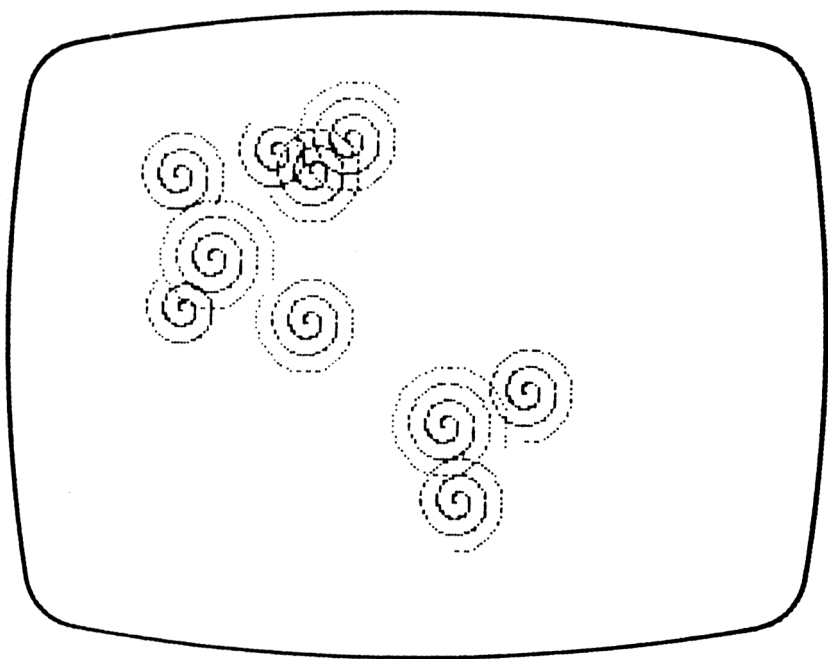
2000 INFORMATION

Ce sous-programme contient une brève description du jeu.

25. Spirales

C'est un court programme qui dessine des spirales sur votre écran en mode haute résolution (HIRES). Après l'initialisation des variables, plusieurs spirales aléatoires sont dessinées.

Pour dessiner une spirale, vous devez afficher des points de la forme :
 $(X_{MID} + Z \cdot \cos(Z), Y_{MID} + Z \cdot \sin(Z))$
pour des valeurs de Z comprises entre 0 et $5 \cdot \pi$. Le point (X_{MID}, Y_{MID}) est le centre de la spirale.



```

10 REM *****
20 REM * SPIRALES *
30 REM *****
40 REM
100 REM ** PROGRAMME PRINCIPAL **
110 INK 6:PAPER 0:CLS:DEF FNR(X)=INT(RND(1)*X)
120 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4):PRINT
130 PRINT D$;SPC(10);"SPIRALES":PRINT D$:PRINT:PRINT
140 PRINT D$;SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
150 PRINT "Des spirales colorees aleatoires..."
200 REM ** Affichage (plot) **
210 WAIT 200:HIRES:POKE #26A,2
220 PRINT SPC(13);"SPIRALES"
230 FOR Y=0 TO 199:CURSET 0,Y,3:FILL 1,1,FNR(7)+1:NEXT
240 FOR I=0 TO 9
250 : XMID=FNR(175)+40:Y MID=FNR(148)+26
260 : FOR Z=0 TO 15+FNR(10) STEP.1
270 : CURSET XMID+Z*COS(Z),Y MID+Z*SIN(Z),1
280 : NEXT
290 NEXT
300 PING
310 FOR I=0 TO 5
320 : FOR Y=0 TO 199:CURSET 0,Y,1:FILL 1,1,FNR(7)+1:NEXT
330 NEXT
340 WAIT 50:PING:FOR I=1 TO 6:INK I:WAIT 75:NEXT
350 E$=" "+CHR$(27)
360 PRINT E$+"B"+E$+"L Autre partie? O ou N"
370 REPEAT:G$=KEY$:UNTIL G$="O" OR G$="N"
380 TEXT:IF G$="O" THEN RUN
390 CLS:PRINT "A bientot.":END

```

26. Knees up Mother Brown!

C'est une chanson anglaise bien connue. Si vous n'en connaissez pas encore les paroles, cela ne va pas durer. A chaque fois qu'une note est jouée, le mot ou la syllabe correspondant est affiché. Si vous connaissez les paroles, vous pouvez regarder l'homme faisant ses exercices, sautant en mesure avec la musique. La chanson est jouée avec des clés différentes.

KNEES UP MOTHER BROWN!

I've just been to a ding-dong
down dear old Brixton way
Old Mother Brown the Pearly Queen's
a hundred years today
Oh! what a celebration!
was proper lah-di-dah!
Until they roll'd the carpet up
and shouted Nah then Ma
Ooh! knees up Mother Brown
knees up Mother Brown
Under the table you must go
Ee-i-ee-i-ee-i-oh!
If I catch you bending
I'll saw your leg right off
So knees up knees up
Don't let the breeze up
Knees up Mother Brown!




```

10 REM *****
20 REM * KNEES UP MOTHER BROWN! *
30 REM *****
40 REM
50 SELECT=300:INFO=500:CHR=800
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO
120 REPEAT
130 : PLOT 35,Y," " :PLOT 35,Y+1," "
140 : I=I+1:IF I>MAX THEN GOSUB SELECT
150 : B=INT((A(I)-1)/12):Y=25-A(I):K=NOT K
160 : PRINT C$(I);
170 : PLOT 35,Y,CHR$(J-1)+A$(-K)
180 : PLOT 35,Y+1,CHR$(J-1)+B$(-K)
190 : IF A(I)>0 THEN MUSIC 1,J+B,A(I)-12*B,0:PLAY 1,0,1,L
200 : WAIT 7*T(I)
210 UNTIL KEY$="Q"
220 PLAY 0,0,0,0:PRINT
230 PRINT E$;"B";E$;"Une autre fois? O ou N";SPC(10)
240 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
250 IF G$="O" THEN RUN
260 CLS:PRINT "A bientot.":POKE #26A,3:INK 7:END
300 REM ** SELECTION **
310 I=0:CLS:PRINT:PRINT D$;SPC(4);K$:PRINT D$:PRINT:PRINT
320 J=J+1: IF J>5 THEN J=3
330 RETURN
500 REM ** INFORMATION **
510 INK 5:PAPER 0:CLS:DEF FNR(X)=INT(RND(1)*X)
520 E$=" " +CHR$(27):D$=E$+"J"+CHR$(4):PRINT
530 K$="KNEES UP MOTHER BROWN!"
540 PRINT D$;SPC(4);K$:PRINT D$:PRINT:PRINT
550 PRINT D$;SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
560 PRINT "Une simple melodie.....":PRINT
570 PRINT "Et quelques mots.....":PRINT
580 PRINT "Tapez Q pour commencer....";
590 REPEAT:GET G$:UNTIL G$="O"
600 PRINT:PRINT:PRINT "Tapez Q pour sortir. ":PRINT:PRINT
610 FOR J=0 TO 1
620 : A$(J)=CHR$(38-J)+CHR$(96-J)
630 : B$(J)=CHR$(124-J)+CHR$(126-J)
640 NEXT
650 PRINT D$+A$(0)+" " +A$(1):PRINT D$
660 PRINT D$+B$(0)+" " +B$(1):PRINT D$
670 START=#B400+37*8:AA=#A000
680 POKE AA,2:IF PEEK (AA)<>2 THEN START=START-#B000
690 GOSUB CHR
700 START=START+8:GOSUB CHR:START=START+57*8:GOSUB CHR
710 START=START+8:GOSUB CHR:START=START+27*8:GOSUB CHR
720 FOR J=0 TO 2:START=START+8:GOSUB CHR:NEXT
730 DIM A(120),T(120),C$(120):MAX=-1:WMAX=-1
740 REPEAT:MAX=MAX+1:READ A(MAX),T(MAX):UNTIL A(MAX)=-1
750 REPEAT:WMAX=WMAX+1:READ C$(WMAX):UNTIL C$(WMAX)="-1"
760 IF MAX<>WMAX THEN PRINT "Erreur de donnees":END

```

```

770 MAX=MAX-1:CLS:POKE #26A,10:INK 6
780 J=2:K=0:I=MAX:Y=13:L=2500
790 RETURN
800 REM ** CHR **
810 FOR I=START TO START+7:READ R:POKE I,R:NEXT
820 RETURN
1000 REM ** DONNEES (DATA) **
1010 REM DATA - Caracteres
1020 DATA 3,3,3,1,1,7,11,19
1030 DATA 3,3,35,17,9,7,3,3
1040 DATA 32,32,32,0,0,48,40,36
1050 DATA 32,32,34,4,8,48,32,32
1060 DATA 35,3,2,2,2,2,2,2
1070 DATA 3,3,2,4,8,16,32,0
1080 DATA 34,32,32,32,32,32,32,32
1090 DATA 32,32,32,16,8,4,2,0
1100 REM DATA - notes
1110 DATA 8,2,10,2,8,2,7,2,8,2,13,2,13,4,5,2
1120 DATA 6,2,5,2,4,2,5,2,8,6,5,2,8,2,10,2,8,2,6,2
1130 DATA 5,2,6,2,8,2,13,2,15,2,13,2,12,2,10,2
1140 DATA 8,6,8,2,10,2,8,2,7,2,8,2,13,2,13,4,13,2
1150 DATA 12,2,7,2,8,2,10,2,12,6,8,2,3,2,5,2,7,2,8,2
1160 DATA 10,2,12,2,13,2,10,2,15,2,13,2,12,2,10,2
1170 DATA 8,2,0,4,8,2
1180 DATA 5,2,8,2,8,2,8,2,8,8,6,2,10,2,10,2,10,2,10,8
1190 DATA 2,1,12,1,12,2,12,2,12,2,10,2,9,2,8,4
1200 DATA 12,2,12,2,12,2,12,2,10,2,9,2,8,4
1210 DATA 5,2,8,2,8,2,8,2,8,2,8,4,14,2,15,2,10,2,10,2
1220 DATA 10,2,10,6,11,2,12,2,12,2,10,2,8,2,12,2,12,1
1230 DATA 12,1,10,2,8,2,8,2,8,2,10,2,12,2,13,6,8,2,0,8
1240 DATA -1,-1
1250 REM DATA - PAROLES
1260 DATA I've ,just ,been ,to ,a ,ding-,"dong"
1270 DATA down ,dear ,old ,Brix,ton ,,"way"
1280 DATA Old ,Mot,her ,Brown ,the ,Pearl,y ,,"Queen's"
1290 DATA a ,hun,dred ,years ,to,"day"
1300 DATA Oh! ,what ,a ,cel,ebra,"tion!"
1310 DATA was ,prop,er ,lah-,di-,"dah!"
1320 DATA Un,til ,they ,roll'd ,the ,car,pet ,,"up"
1330 DATA and ,shou,ted ,Nah ,then ,,"Ma"
1340 DATA Ooh! ,knees ,up ,Moth,er ,,"Brown"
1350 DATA knees ,up ,Moth,er ,,"Brown"
1360 DATA Un,der ,the ,tab,le you ,must ,,"go"
1370 DATA Ee-,i-,ee-,i-,ee-,i-,"oh!"
1380 DATA If ,I ,catch ,you ,bend,"ing"
1390 DATA I'll ,saw ,your ,leg ,right ,,"off"
1400 DATA So ,knees ,up ,knees ,,"up"
1410 DATA Don't ,let ,the ,breeze ,,"up"
1420 DATA Knees ,up ,Moth,er ,Brown! , , ,
1430 DATA -1

```

LE PROGRAMME PRINCIPAL

Il joue chaque note et affiche sur l'écran les paroles et l'homme. Tapez Q pour sortir du programme. L'homme est fait de quatre caractères, deux contenus dans AS(I) et deux autres dans BS(I). Il a deux apparences suivant que I = 0 ou 1. Les notes sont contenues dans A(I) et les intervalles de temps dans T(I). Les paroles sont rangées dans CS(I). Reportez vous à l'annexe 3 pour plus de détails sur la musique.

LES SOUS-PROGRAMMES

300 DELECT

Lorsque la chanson est finie, l'écran est rafraîchi. Une nouvelle clé est choisie pour la musique.

500 INFOrmation

Il contient une brève description du programme. Huit caractères spéciaux sont créés. Lorsqu'ils sont utilisés par groupe de quatre, un homme de grande taille apparaît. Il peut apparaître sous deux états.

Après cela, les données (DATA) des notes et des paroles sont lues. Soyez très attentifs lorsque vous tapez les données des paroles, chaque espace blanc est important. Un blanc de plus ou de moins et tout le programme devient inopérant. Pour chaque DATA, le dernier élément a été mis entre guillemets pour s'assurer qu'il y a le bon nombre de blancs. Dans deux DATA, il y a une virgule en fin de ligne ; cela signifie seulement que la donnée suivante est vide.

800 CHR

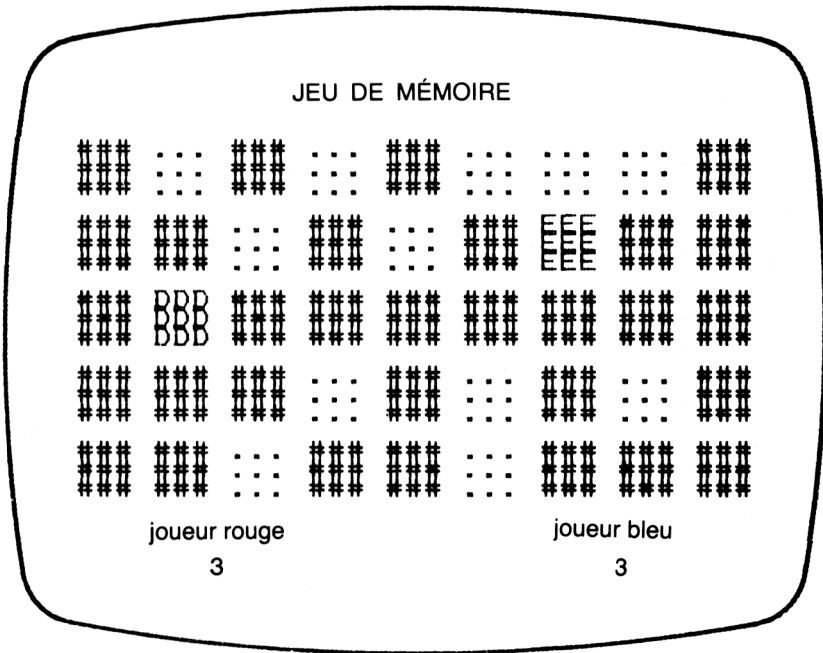
Ce sous-programme lit les données des caractères spéciaux et les POKE aux bons endroits.

27. Jeu de mémoire

C'est un programme qui peut aiguïser votre mémoire, un jeu pour deux dont les principaux éléments sont 22 paires de cartes repérées par les lettres comprises entre A et V. Les lettres sont invisibles. A votre tour, vous pouvez retourner deux cartes et regarder leur lettre. Si elles sont semblables, vous les gardez et vous pouvez en retourner deux autres. Autrement, les cartes sont retournées faces cachées et c'est à l'autre joueur de tenter sa chance. Le jeu continue jusqu'au moment où il n'y a plus de carte à retourner. Le joueur qui a le plus de cartes a gagné.

En fait il y a 45 cartes, 22 paires et une carte seule repérée par le symbole @. Cette carte ne peut être retirée du jeu.

Une carte qui clignote peut être retournée en tapant Y. Utilisez les curseurs pour changer de carte clignotante.



```

10 REM *****
20 REM * JEU DE MEMOIRE *
30 REM *****
40 REM
50 CHANGE=500:DEPL=600:OUI=700:VISU=800:RETO=900
60 IHITI=1000:INFO=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:GOSUB IHITI
120 REPEAT
130 : PLOT X(M),24,8:PING:WAIT 30:PING
140 : M=-NOT-M:PLOT X(M),24,12
150 : REPEAT
160 : ZZ=12:GOSUB CHANGE:WW=0
170 : REPEAT
180 : REPEAT:UNTIL KEY$=""
190 : REPEAT
200 : GET G$:G=ASC(G$)
210 : UNTIL G>7 AND G<12 OR G=79
220 : IF G>7 AND G<12 THEN GOSUB DEPL
230 : IF G=79 THEN GOSUB VISU
240 : UNTIL WW>1
250 : STATE=TRUE:B$="####"
260 :IF P$(0)=P$(1) THEN GOSUB OUI
270 : GOSUB RETO : UNTIL STATE
290 UNTIL SC(0)+SC(1)>21
300 CLS:PRINT:PRINT SPC(10);N$:PRINT D$
310 PING:PRINT:PRINT "SCORES:":PRINT:PRINT
320 WAIT 50:PING:WAIT 50:PING
330 PRINT E$;"AJOUEUR ROUGE ";SC(0):PRINT:PRINT
340 PRINT E$;"DJOUEUR BLEU ";SC(1):PRINT:PRINT:WAIT 99
350 PRINT E$;"B";E$;"L Autre partie? O ou N"
360 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
370 IF G$="O" THEN RUN
380 CLS:PRINT "A bientot.":INK 7:POKE #26A,3:END
500 REM ** CHANGement **
510 FOR I=0 TO 2
520 : PLOT 4*X+2,4*Y+4+I,ZZ
530 NEXT
540 RETURN
600 REM ** DEPLacement **
610 ZZ=8:GOSUB CHANGE
620 IF X>0 AND G=8 THEN X=X-1
630 IF X<8 AND G=9 THEN X=X+1
640 IF Y>0 AND G=11 THEN Y=Y-1
650 IF Y<4 AND G=10 THEN Y=Y+1
660 ZZ=12:GOSUB CHANGE
670 RETURN
700 REM ** OUI **
710 PING:SC(M)=SC(M)+1
720 PLOT 20*M+8,26,STR$(SC(M))
730 STATE=FALSE:B$="..."
740 IF SC(0)+SC(1)>21 THEN STATE=TRUE
750 RETURN

```

```

800 REM ** VISU **
810 IF SCRN(4*X+3,4*Y+5)<>35 THEN RETURN
820 FOR I=0 TO 2
830 : PLOT 4*X+3,4*Y+4+I,A$(X+9*Y)
840 NEXT
850 A(WW)=X:B(WW)=Y:P$(WW)=A$(X+9*Y)
860 WW=WW+1
870 RETURN
900 REM ** RETOurnement **
910 ZZ=8:GOSUB CHANGE:WAIT 100
920 IF STATE THEN SHOOT:WAIT 300
930 FOR J=0 TO 1:FOR I=0 TO 2
940 : PLOT 4*A(J)+3,4*B(J)+4+I,B$
950 NEXT:NEXT
960 PING
970 RETURN
1000 REM ** IHITialisation **
1010 CLS:INK 5:PING
1020 PRINT:PRINT SPC(10);N$:PRINT D$:PRINT
1030 FOR I=0 TO 4
1040 : FOR J=0 TO 2
1050 : PRINT " ";CHR$(8);
1060 : FOR K=0 TO 8
1070 : PRINT CHR$(27)+"H###";
1080 : NEXT
1090 : PRINT
1100 : NEXT
1110 : PRINT
1120 NEXT
1130 PRINT E$;"H";E$;"AJQUEUR ROUGE ";
1140 PRINT E$;"H";E$;"DJOUEUR BLEU "
1150 PLOT 8,26,STR$(0):PLOT 28,26,STR$(0)
1160 DIM A$(44)
1170 FOR I=0 TO 22
1180 : FOR J=0 TO 2
1190 : IF I>0 THEN A$(2*I-1)=A$(2*I-1)+CHR$(64+I)
1200 : A$(2*I)=A$(2*I)+CHR$(64+I)
1210 : NEXT
1220 NEXT
1230 M=0:IF RND(1)<.5 THEN M=1
1240 X(0)=2:X(1)=20
1250 X=FNR(9):Y=FNR(5)
1260 REM MELANGE
1270 FOR I=0 TO 43
1280 : PP=FNR(45-I)+I
1290 : TT$=A$(I):A$(I)=A$(PP):A$(PP)=TT$
1300 NEXT
1310 RETURN
2000 REM ** INfORMation **
2010 INK 2:PAPER 0:CLS:DEF FNR(X)=INT(RND(1)*X)
2020 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4):PRINT
2030 N$=D$+"JEU DE MEMOIRE"
2040 PRINT SPC(7);N$:PRINT D$:PRINT:PRINT

```

```

2050 PRINT D$;SPC(6);"de Czes Kosniowski":PRINT D$:PRINT
2060 PRINT "C'est un jeu pour deux. A votre tour"
2070 PRINT "vous pouvez retourner deux cartes. Si"
2080 PRINT "elles sont semblables, vous les "
2090 PRINT "gardez et continuez. Sinon, les cartes";
2100 PRINT "sont retournees et c'est le tour de"
2110 PRINT "l'autre joueur.":PRINT:WAIT 99
2120 PRINT "La carte qui clignote peut etre "
2130 PRINT "retournee. Utilisez les curseurs pour"
2140 PRINT "la deplacer. Puis tapez 0 pour la"
2150 PRINT "retourner.":PRINT:WAIT 99
2160 PRINT "Decidez maintenant qui sera le joueur"
2170 PRINT "ROUGE et qui sera le BLEU.":PRINT
2180 PRINT:PRINT:PRINT "Tapez 0 pour commencer.";
2190 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$="0"
2200 POKE #26A,2
2210 RETURN

```

LE PROGRAMME PRINCIPAL

Il consiste en des boucles REPEAT/UNTIL. La première permet l'alternance entre les deux joueurs et continue jusqu'à épuisement des cartes. La seconde continue jusqu'à ce qu'une paire de cartes différentes apparaisse. La troisième permet aux joueurs de sélectionner deux cartes.

LES SOUS-PROGRAMMES

500 CHANGement

Ce sous-programme provoque ou arrête le clignotement d'une carte suivant que $zz = 8$ ou 12 .

600 DEPLacement

Il modifie la position de la carte clignotante.

700 OUI

Si les deux cartes retournées sont semblables, ce sous-programme est utilisé pour les enlever du jeu.

800 VISUalisation

Ce sous-programme révèle les cartes aux joueurs.

900 RETOurnement

Ce sous-programme retourne les cartes vers la position face cachée.

1000 INITIalisation

L'affichage est créé par ce sous-programme. Le tableau de variables A\$(I) attribue une lettre à chaque carte. Le contenu de ce tableau est ensuite rendu aléatoire.

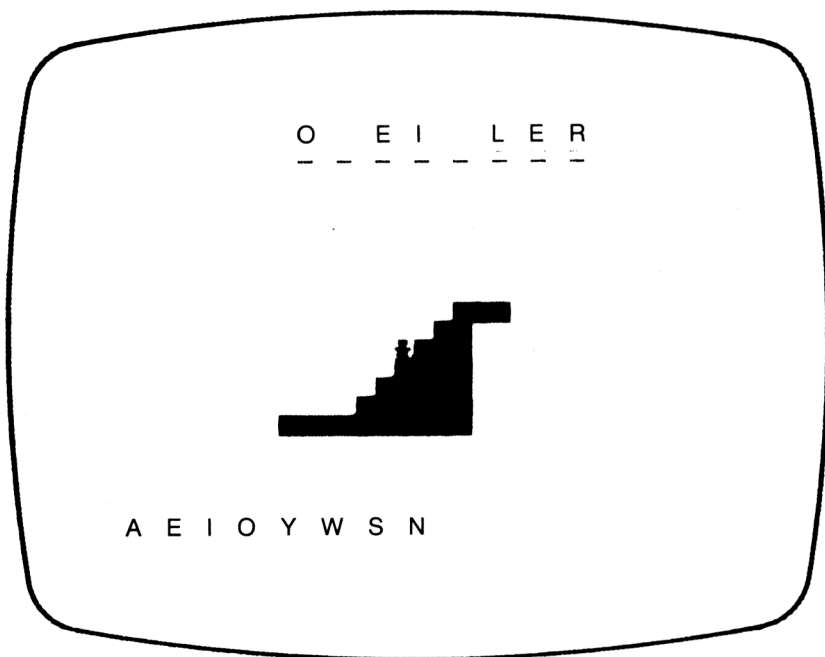
2000 INFOrmation

Il contient une brève description du jeu.

28. Le bourreau

C'est le traditionnel jeu du pendu mais sans la pendaison ! A la place, un homme est forcé de monter une série de marches... Comme d'habitude, vous devez découvrir le mot sélectionné par l'ordinateur. Il vous indique le nombre de lettres de ce mot. A chaque fois que vous entrez une lettre, l'ordinateur vous dit si le mot contient celle-ci ou non. Si oui, la lettre est placée dans le mot, plusieurs fois s'il contient cette lettre plus d'une fois. Si vous entrez une mauvaise lettre, l'homme monte d'une marche. Au sommet de l'escalier se trouve une planche, après...

Ne rentrez pas la même lettre deux fois, l'homme avancerait d'une marche vers son destin.



```

10 REM *****
20 REM * LE PENDU *
30 REM *****
40 REM
50 CHOIX=500:OUI=600:NON=700:FAUX=800
60 IHITI=1000:INFO=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:GOSUB IHITI
120 REPEAT
130 : GOSUB CHOIX
140 : REPEAT
150 : REPEAT:GET G$:UNTIL G$>"a" AND G$<"*"
160 : PRINT G$;" ";BLANK=TRUE
170 : FOR I=1 TO L
180 : IF MID$(B$,I,1)=G$ THEN GOSUB OUI
190 : NEXT
200 : IF BLANK THEN GOSUB NON
210 : UNTIL STATE=L OR NUMBER>9
220 : PRINT:PRINT
230 : IF STATE=L THEN PING:WAIT 10:PING:PLOT 1,5,12
240 : IF NUMBER>9 THEN GOSUB FAUX
250 : PRINT E$;"B";E$;"L Autre partie? O ou N"
260 : REPEAT:GET G$:UNTIL G$="O" OR G$="N"
270 UNTIL G$="N"
280 CLS:PRINT "A bientot.":INK 7:END
500 REM ** CHOIX **
510 STATE=0:NUMBER=0:CLS:FOR I=0 TO 21:PRINT:NEXT
520 B$=R$(FNR(MAX)):L=LEN(B$)
530 PLOT 0,1,10:PLOT 1,1,6:PLOT 0,2,10:PLOT 1,2,6
540 PLOT 0,3,9:PLOT 1,3,3:B=INT(17-L)
550 FOR I=B+2 TO B+2*L STEP 2:PLOT I,3,44:NEXT
560 PLOT 0,5,2:PLOT 1,5,0:PLOT 14,5,"BIEN JOUE"
570 FOR I=1 TO 6:PLOT 20-I,10+I,LEFT$(C$,I):NEXT
580 PLOT 20,11,LEFT$(C$,2):PLOT 10,17,C$
590 PLOT X(0),Y(0),37:RETURN
600 REM ** OUI **
610 IF SCRNB(B+2*I,1)>32 THEN SHOOT:WAIT 20:RETURN
620 PING:PLOT B+2*I,1,G$:PLOT B+2*I,2,G$
630 STATE=STATE+1:BLANK=FALSE
640 RETURN
700 REM ** NON **
710 PLOT X(NUMBER),Y(NUMBER),32
720 ZAP:NUMBER=NUMBER+1
730 PLOT X(NUMBER),Y(NUMBER),37
740 RETURN
800 REM ** FAUX **
810 FOR I=1 TO L
820 : G$=MID$(B$,I,1):GOSUB OUI
830 : PLOT 22,9+I,32:PLOT 22,10+I,37
840 NEXT
850 PLOT 21,10+L,"AIE!!":EXPLODE:WAIT 50
860 RETURN
1000 REM ** Initialisation **

```

```

1010 MAX=-1: DIM R$(100)
1020 REPEAT: MAX=MAX+1: READ R$(MAX): UNTIL R$(MAX)="*"
1030 FOR I=0 TO 10: READ X(I), Y(I): NEXT
1040 START=#B400+37*8: AA=#A000
1050 POKE AA, 2: IF PEEK(AA)<>2 THEN START=START-#8000
1060 FOR I=START TO START+7: READ R: POKE I, R: NEXT
1070 RETURN
1100 REM ** DONNEES **
1110 DATA TABLE, CHAISE, LIVRE, OREILLER, CUBE, HYENE, POIGNET
1120 DATA VERRE, CUVETTE, TORCHE, CUISINE, TELEVISION, BOUTEILLE
1130 DATA HARICOT, SALUT, NOMMER, QUELLE, COUREUR, DOUZE, ZEBRE
1140 DATA VOLEUR, SOLEIL, FINIR, DEVINER, CORNET, PAPILLON
1150 DATA VAINCRE, FLEURIR, COCHER, ROCHER, ORANGE, NOIRCIR
1160 DATA NULLITE, BLANCHIR, BEURRE, TRICHEUR, TABLEAU
1170 DATA VIANDE, TRAITER, THEATRE, PROGRAMME, VIOLET, GRISAILLE
1180 DATA GRAMMAIRE, ECOLE, SIGNE, POSTE, COTE, EGLISE, LICORNE
1190 DATA CHAMBRE, TISSU, MANQUER, TEST, AOUT, MINUTE, HEURE
1200 DATA ANIMAL, POUSSIERE, DEBUT, MANUFACTURE, ONCLE, BATEAU
1210 DATA ZOOM, VINGT, SQASH, TRICHER, MAGNIFIQUE, FEVRIER
1220 DATA DIAMANT, JANVIER, MARS, AVRIL, JUIN, JUILLET
1230 DATA SEPTEMBRE, OCTOBRE, NOVEMBRE, DECEMBRE, COUSIN
1240 DATA DESSINER, TOMBER, NOTICE, OFFRIR, TRAVAIL, CALENDRIER
1250 DATA POUDRE, SPHINX, ZYGOMATIQUE, VOLUME, ANCRE, DEBRANCHER
1280 REM place pour plus de mots...
1290 DATA *
1300 DATA 12, 16, 13, 16, 14, 15, 15, 14
1310 DATA 16, 13, 17, 12, 18, 11, 19, 10
1320 DATA 20, 10, 21, 10, 22, 10
1330 DATA 14, 14, 4, 31, 14, 14, 10, 10
2000 REM ** INFORMATION **
2010 INK 5: PAPER 0: CLS: DEF FNR(X)=INT(RND(1)*X)
2020 E$=" "+CHR$(27): D$=E$+"J"+CHR$(4): PRINT
2030 PRINT D$; SPC(12); "LE PENDU ": PRINT D$: PRINT: PR"NT
2040 PRINT D$; SPC(6); "de Czes Kosniowski": PRINT D$: PR"NT
2050 PRINT "Trouvez le mot avant que l'homme "
2060 PRINT "parvienne au sommet et tombe.....": PRINT
2070 PRINT "A chaque mauvaise lettre l'homme monte";
2080 PRINT "d'une marche...": PRINT
2090 PRINT "Tapez 0 pour commencer.";
2100 REPEAT: G$=KEY$: GG=RND(1): UNTIL G$="0"
2110 PRINT: PRINT: PRINT "On y va...";
2120 FOR I=0 TO 9: C$=C$+CHR$(255): NEXT
2130 RETURN

```

PROGRAMME PRINCIPAL

Le programme principal attend que vous pressiez une touche (une des touches A à Z). Cette lettre est imprimée au bas de l'écran. Une vérification est faite pour voir si cette lettre est contenue dans le mot choisi par l'ordinateur. Si c'est le cas, le sous-programme OUI est appelé, sinon le sous-programme NON est utilisé. Le processus complet se poursuit jusqu'à ce que 10 erreurs aient été commises ou que le mot ait été trouvé correctement.

SOUS-PROGRAMMES

500 CHOIX

Il choisit un mot au hasard et crée l'affichage approprié sur l'écran. La variable c\$ est formée de dix caractères (CHR\$(255)) et est utilisée pour créer les escaliers.

600 OUI

Si la lettre choisie par le joueur existe dans le mot, ce sous-programme est appelé pour afficher cette lettre. Remarquez que, si la lettre a déjà été choisie précédemment, cela vous est compté comme une erreur.

700 NON

Le choix d'une mauvaise lettre (ou une lettre déjà choisie) appelle ce sous-programme. Il fait gravir à l'homme une marche d'escalier.

800 FAUX

Si le mot n'est pas trouvé, l'homme tombe dans le précipice et l'ordinateur émet le son SPLAT.

1000 INITIalisation

Il lit les mots qui se trouvent en mémoire. Vous pouvez ajouter des mots ou changer très facilement ceux contenus dans les données (DATA).

Le chemin suivi par l'homme pour grimper les escaliers est stocké dans les variables x(i), y(i).

Un caractère spécial, figurant l'homme, est nécessaire.

2000 INFOrmation

Ce sous-programme contient une brève description du jeu.

29. Trouvez les nombres

Dans ce jeu l'ordinateur choisit des nombres. Vous avez à les trouver en utilisant les informations fournies par l'ordinateur. A chaque essai il vous sera dit combien de nombres corrects vous avez deviné. Un signal de couleur cyan est utilisé pour indiquer un nombre correct et bien positionné. Un signal de couleur magenta indique un nombre correct mais mal positionné.

Vous êtes autorisé à dire combien de nombres l'ordinateur doit choisir. Vous pouvez également dire quel peut être le nombre maximum. Par exemple, vous pouvez demander à l'ordinateur de choisir trois nombres entre 1 et 2. Cela donne un jeu plutôt facile. A l'autre extrême, vous pouvez demander à l'ordinateur de choisir neuf nombres entre 1 et 9. Cela sera beaucoup plus difficile.

DECOUVREZ LES NOMBRES

1	2	2	1	1	1	■	■	■	✶	✶	
1	1	2	2	2	2	■	■	■	✶	✶	
2	2	2	1	1	1	■	■	■	■	✶	✶
2	1	2	1	1	2	■	■	■	■	✶	✶
2	1	2	1	2	1	■	■	■	■	✶	✶
2	1	2	2	1	1	■	■	■	■	■	■

BIEN JOUE — Vous l'avez fait en 6
Une autre partie ? O ou N

```

10 REM *****
20 REM * DECOUVREZ LES NOMBRES *
30 REM *****
40 REM
50 CHECK=500:ANSWER=700:SETUP=1000:INFO=2000
100 REM ** PROGRAMME PRINCIPAL **
110 GOSUB INFO:GOSUB SETUP
120 REPEAT
130 : FOR I=1 TO NUMBER:C$(I)=A$(I):NEXT
140 : PLOT 1,0,T$:PING:REPEAT:UNTIL KEY$=""
150 : FOR I=1 TO NUMBER
160 : REPEAT:GET G$:UNTIL G$>="0" AND G$<CHR$(49+RANGE)
170 : IF G$="0" THEN STATE=TRUE:I=NUMBER
180 : PRINTG$+" ";B$(I)=G$
190 : NEXT
200 : PING:PLOT 1,0,S$
210 : IF NOT STATE THEN GOSUB CHECK
220 : PRINT:PRINT:TRIES=TRIES+1
230 UNTIL STATE
240 IF STATE<0 THEN GOSUB ANSWER
250 IF STATE=1 THEN PRINT W$:TRIES
260 WAIT 50:PRINT:PRINT E$+"L      Une autre partie? o ou n"
270 REPEAT:GET G$:UNTIL G$="0" OR G$="N"
280 IF G$="0" THEN RUN
290 CLS:INK 7:PRINT" A BIENTOT":END
500 REM ** CHECK **
510 X$="":Y$=""
520 FOR I=1 TO NUMBER
530 : IF C$(I)=B$(I) THEN X$=X$+"° ":B$(I)="B":C$(I)="C"
540 NEXT
550 FOR I=1 TO NUMBER:FOR J=1 TO NUMBER
560 : IF C$(I)=B$(J) THEN Y$=Y$+"9 ":B$(J)="B":C$(I)="C"
570 NEXT:NEXT
580 PRINT D$+F$+"F"+F$+"I"+" "+X$+D$+F$+"E"+Y$+D$
590 IF LEN(X$)=2*NUMBER THEN STATE=1:PING:WAIT 30:PING
600 RETURN
700 REM ** ANSWER **
710 PRINT"Etait-ce difficile?":SHOOT
720 PRINT:PRINT"La reponse est :":PRINT
730 FOR I=1 TO NUMBER:PRINT A$(I)+" ";:NEXT
740 PRINT
750 RETURN
1000 REM ** SET UP **
1010 PAPER 0:INK 2:CLS
1020 FOR I=1 TO NUMBER
1030 : A$(I)=CHR$(49+INT(RND(1)*RANGE))
1040 NEXT
1050 T$=CHR$(12)+"Tapez "+MID$(STR$(NUMBER),2)
1060 T$=T$+" nombres, entre 1 et "+MID$(STR$(RANGE),2)
1070 FOR I=0 TO 33:S$=S$+" ":NEXT
1080 STATE=FALSE:PRINT:PRINT:PRINT N$:PRINT
1090 W$="BIEN JOUE - vous avez trouve en "
1100 RETURN

```

```

2000 REM ** INFORMATION **
2010 PAPER 0:INK 7:CLS
2020 F$=CHR$(27):E$=" "+F$:D$=CHR$(8)
2030 N$=E$+"A" TROUVEZ LES NOMBRES"
2040 PRINTCHR$(4):PRINT E$+"J"+N$:PRINT:PRINT
2050 PRINT:PRINT E$+"J"+E$+"B" par Czes Kosniowski"
2060 PRINT CHR$(4):PRINT:PRINT
2070 PRINT"L'ordinateur va choisir quelques"
2080 PRINT"nombres que vous devrez deviner"
2085 PRINT"a partir des figures affichees."
2090 PRINT"Le symbole "+F$+"I"+F$+"F"+F$+"H"+F$+"Gapparaît";
2110 PRINT" pour chaque nombre dans la bonne position."
2120 PRINT"Le symbole "+F$+"I"+F$+"E9"+F$+"H"+F$+"G";
2130 PRINT" apparaît pour un bon nombre dans la mauvaise position"
2160 PRINT:PRINT"Si vous voulez arreter et avoir la solution, tapez 0"
2170 PRINT:PRINT
2180 PRINT"Combien de nombres voulez-vous que"
2190 PRINT"l'ordinateur choisisse? de 3 a 9: ";
2200 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$>"2" AND G$<":
2210 NUMBER=VAL(G$):PRINT G$
2220 PRINT:PRINT"Choisissez la grandeur maximale des nombres. de 2 a 9: ";
2240 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$>"1" AND G$<":
2250 RANGE=VAL(G$):PRINT G$
2260 PRINT:PRINT "O.K. Nous y allons! ";:WAIT 100
2270 RETURN

```

PROGRAMME PRINCIPAL

La boucle principale attend à chaque essai que vous ayez fini de rentrer votre séquence de nombres. Ces nombres sont affichés et un test est fait, utilisant le sous-programme TEST, pour voir combien d'entre eux sont corrects. Le processus se poursuit jusqu'à ce que la séquence correcte ait été trouvée ou que le chiffre 0 soit pressé. En pressant le chiffre 0, vous pouvez abandonner.

SOUS-PROGRAMMES

500 TEST

Ce sous-programme fait d'abord un test pour voir combien de nombres sont à la fois corrects et en bon ordre. Il fait ensuite un autre test pour voir combien des nombres restant sont corrects mais mal positionnés.

700 REPONSE

Si 0 est pressé, le programme principal est stoppé et ce sous-programme affiche alors la bonne réponse.

1000 INITIalisation

Ce sous-programme choisit aléatoirement les nombres que vous aurez à deviner.

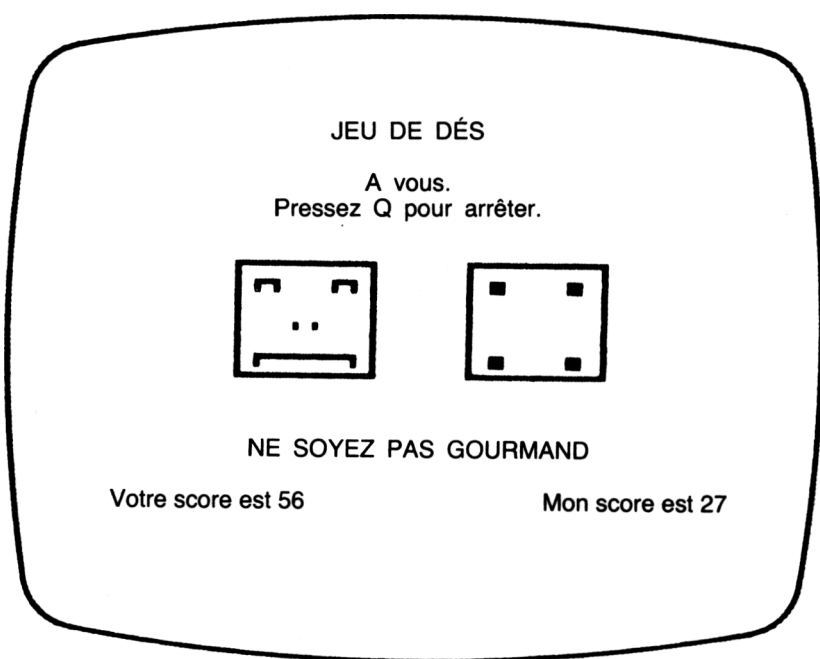
2000 INFOrmation

Ce sous-programme contient une brève description du jeu. Il demande également combien de nombres (NUMBER) l'ordinateur doit choisir et le maximum (RANGE) de ces nombres.

30. Ne soyez pas gourmand

Si vous aimez les jeux de dés, en voici un pour vous. Il contient une bonne part de chance mais exige aussi un peu d'astuce. Ce jeu utilise deux dés qui ressemblent à des dés ordinaires excepté qu'il n'y a pas de nombre 1. Il est remplacé par une tête d'aspect lamentable.

Vous jouez contre l'ordinateur. Le but du jeu est d'être le premier à atteindre un score supérieur à 100. L'ordinateur et vous lancez les dés à tour de rôle. Chaque tour peut comprendre plusieurs lancers (au moins un mais autant que vous voulez). Vous pouvez vous arrêter de lancer les dés quand vous voulez en pressant Q. Si vous obtenez une tête triste, votre tour est stoppé et votre score pour ce tour est nul. Si vous obtenez des tristes têtes sur les deux dés à la fois, votre tour est également stoppé mais c'est votre score



général qui se trouve ramené à 0. Si aucune tête n'apparaît, vous avez le choix entre lancer les dés à nouveau ou passer la main à l'ordinateur. Dans ce dernier cas, le score de votre tour est ajouté à votre total général. L'ordinateur procède de la même manière : après quoi, c'est à nouveau votre tour. L'astuce est de décider quand stopper de jouer — rappelez-vous simplement : *ne soyez pas gourmand*.

Nota :

Les deux dés sont lancés aléatoirement et l'ordinateur ne triche pas...

```

40 REM
50 YOU=300:ME=500:DICE=700:ADDUP=800:CARRY=900:INFO=1000
100 REM ** THE MAIN PROGRAM **
110 GOSUB INFO:ANSWER=TRUE
120 REPEAT
130 : IF ANSWER THEN GOSUB YOU
140 : IF ANSWER THEN GOSUB ME
150 UNTIL NOT ANSWER
160 FOR I=0 TO 2:PING:WAIT 30:NEXT
170 PRINT:PRINT E$;"B";E$;"L   AUTRE PARTIE? O OU N"
180 REPEAT:GET G$: UNTIL G$="O" OR G$="N"
190 IF G$="O" THEN RUN
200 CLS:PRINT"A BIENTOT": POKE#26A,3:INK7 :END
300 REM ** YOU**
310 STATE=FALSE:M=0:N=0:P=2:SC=YSC:T=0
320 PLOT 14,4,"C'est a vous"
330 REPEAT
340 :PLOT 10,6,"Appuyez sur Q pour arreter"
350 : REPEAT: UNTIL KEY$=""
360 :GOSUB DICE:GOSUB ADDUP
370 : PLOT10,6,"
380 :PRINT H$+"      Votre score est "+STR$(T)
390 : IF NOT STATE THEN GOSUB CARRY
400 UNTIL STATE
410 YSC=SC+T
420 WAIT 50:PRINT H$:PRINT:PRINT J$
430 PLOT 1,20,"Votre score est "+STR$(YSC)+"   ":WAIT 100
440 IF YSC>100 THEN ANSWER=FALSE:PRINT H$:SPC(14);W$
450 PLOT 14,4,"      ":WAIT 50
460 RETURN
500 REM ** ME **
510 : STATE=FALSE:M=0:N=1:P=FNR(20)+15:SC=MSC:T=0
520 PLOT 15,4,"A mon tour"
530 REPEAT
540 : GOSUB DICE:GOSUB ADDUP
550 :PRINT H$+"      Mon score est "+STR$(T)
560 : STATE=STATE OR (RND(1)<.5+2*(SC+T<YSC))
570 : IF SC+T>100 THEN STATE=TRUE
580 : IF NOT STATE THEN PRINT:PRINTSPC(10);Y$
590 : WAIT 250:PRINTCHR$(11)+J$
600 UNTIL STATE
610 MSC=SC+T:WAIT 50:PRINT H$:PRINT:PRINT J$

```

```

620 PLOT 20,20,CHR$(5)+"Mon score est de"+STR$(MSC)+" "
630 IF MSC>100 THEN ANSWER=FALSE:PRINT H$+J$:SPC(15);I$
640 PLOT 15,4," "
650 RETURN
700 REM ** DICE **
710 PLOT 14,2,"LES DES ROULENT"
720 REPEAT
730 : K=FNR(6):L=FNR(6):M=M+N
740 : FOR I=0 TO 2
750 PLOT X,Y+2*I,B$(K,I):PLOTX+Z,Y+2*I,B$(L,I)
760 : NEXT
770 UNTIL KEY$="0" AND P<5 OR M>P
780 PLOT 14,2," "
790 RETURN
800 REM ** ADD UP**
810 IF K=0 OR L=0 THEN ZAP:T=0:STATE=TRUE
820 IF K=0 AND L=0 THEN EXPLODE:SC=0
830 S=0:IFK>0 AND L>0 THEN S=K+L+2: PING
840 T=T+S
850 RETURN
900 REM ** CARRY ON **
910 PRINT:PRINT$+"LVoulez-vous relancer les des? O ou N"
920 REPEAT:GET G$:UNTIL G$="O" OR G$="N"
930 PRINT CHR$(11)+J$
940 IF G$="N" THEN STATE=TRUE:RETURN
950 RETURN
1000 REM ** INFORMATION **
1010 INK3 : PAPER0:CLS:DEF FNR(X)=INT(RND(1)*X)
1020 E$=" " +CHR$(27):D$=E$+"J"+CHR$(4):PRINT
1030 K$="NE SOYEZ PAS GOURMAND!"
1040 PRINTD$:SPC(7);K$:PRINTD$:PRINT:PRINT
1050 PRINTD$:SPC(6);"de Czes Kosniowski":PRINTD$:PRINT
1060 PRINT"Ce jeu utilise deux des. Essayez"
1070 PRINT"d'etre le premier a obtenir un score superieur a 100":PRINT
1080 PRINT"Un coup consiste en une serie de lances de des"
1090 PRINT"Vous pouvez arreter quand vous le voulez,";
1100 PRINT" MAIS vous etes oblige d'arreter si vous obtenez un '1'"
1110 PRINT"(repere par une figure.)"
1120 PRINT"Si vous obtenez une figure, votre score pour ce coup sera de 0"
1130 PRINT"Si vous obtenez 2 figures, votre score total est ramene a 0":PRINT
1140 PRINT"Appuyez sur 0 pour commencer"
1150 REPEAT:G$=KEY$:GG=RND(1):UNTIL G$="0"
1160 A$="<,,,,4":B$="5 5":C$="-,,,,,%"
1170 B$(0,0)="<4 <4":B$(0,1)=" 00 ":B$(0,2)="<,,4"
1180 B$(1,0)=" \ ":B$(1,1)=" ":B$(1,2)=" \ "
1190 B$(2,0)=" \ \ ":B$(2,1)=" \ \ ":B$(2,2)=" \ \ "
1200 B$(3,0)=" \ \ \ ":B$(3,1)=" \ \ \ ":B$(3,2)=" \ \ \ "
1210 B$(4,0)=" \ \ \ \ ":B$(4,1)=" \ \ \ \ ":B$(4,2)=" \ \ \ \ "
1220 B$(5,0)=" \ \ \ \ \ ":B$(5,1)=" \ \ \ \ \ ":B$(5,2)=" \ \ \ \ \ "
1230 X=9:Y=9:Z=12:CLS:POKE #26A,2:INK5
1240 FOR I=Y-2 TO Y+6:PLOT 0,I,9:NEXT
1250 PLOT 0,4,1:PLOT 0,6,3
1260 PLOT X-1,Y-1,A$:PLOTX-1+Z,Y-1,A$

```

```

1270 FOR I=Y TO Y+4:PLOT X-1,I,B$:PLOT X-1+Z,I,B$:NEXT
1280 PLOT X-1,Y+5,C$:PLOT X-1+Z,Y+5,C$
1290 J$=CHR$(14):H$=CHR$(30)+J$:W$="VOUS AVEZ GAGNE"
1300 I$="J'AI GAGNE":Y$="JE RELANCE LES DES"
1310 PLOT 9,17,CHR$(6)+K$
1320 RETURN

```

PROGRAMME PRINCIPAL

Le programme principal est une courte boucle qui se termine quand la valeur de ANSWER est fausse (FALSE). C'est le cas quand l'un des joueurs (vous ou l'ordinateur) atteint un score supérieur à 100.

SOUS-PROGRAMMES

300 VOUS

Ce sous-programme est utilisé quand c'est votre tour de jouer. Il comprend une boucle qui s'achève quand vous décidez de stopper votre tour ou quand vous êtes forcé de le stopper à cause de l'apparition d'une tête.

500 MOI

Ce sous-programme est semblable au précédent sauf qu'il comprend la stratégie suivie par l'ordinateur.

700 DES

Le lancer de dés est exécuté ici. Comme vous pouvez aisément le vérifier, celui-ci est totalement aléatoire et sans tricherie.

800 SCORE

Le score de chaque lancer est calculé par ce sous-programme.

900 CONTINUE

Ce sous-programme vous demande si vous voulez continuer de lancer les dés.

1000 INFORMATION

Il contient une brève description du programme, suivie de données pour les tableaux. Les variables A\$, B\$ et C\$ sont utilisées pour dessiner un carré. Les variables B\$(I, J) contiennent les schémas qui peuvent apparaître sur les dés. Par exemple B\$(0,0), B\$(0,1) et B\$(0,2) ensemble produisent la tête alors que B\$(3,0), B\$(3,1) et B\$(3,2) engendrent un quatre. Remarquez que toutes ces variables sont affichées en utilisant les caractères alternés.

Annexe 1

Sauvegarde de l'écran sur cassette

Vous pouvez un jour vouloir sauvegarder sur cassette un dessin que vous avez créé sur l'écran, spécialement s'il est en haute résolution graphique et a mis un certain temps à se réaliser. Vous pouvez le faire facilement en suivant la méthode qui suit.

Supposons que votre *Oric* est en mode HIRES et que votre dessin est prêt. Tapez alors ce qui suit (ceci dépend de la nature de votre *Oric*, 16k ou 48k).

```
CSAVE «NOM», A # A000, E # BF3F (modèle 48k)
```

```
CSAVE «NOM», A # 2000, E # 3F3F (modèle 16k)
```

Puis enfoncez les touches PLAY et RECORD de votre magnétophone et enfin tapez RETURN sur votre *Oric*.

Pour visualiser ensuite votre dessin, rembobinez d'abord votre cassette et ajustez le volume de sortie de votre magnétophone. Assurez-vous que l'*Oric* est en mode HIRES. Tapez alors ce qui suit :

```
CLOAD «NOM», A # A000, E # BF3F (modèle 48k)
```

```
CLOAD «NOM», A # 2000, E # 3F3F (modèle 16k)
```

Faites alors RETURN sur l'*Oric* et PLAY sur le magnétophone. Le dessin devrait apparaître rapidement sur l'écran. Vous pouvez joindre les quelques lignes de sauvegarde qui suivent à vos programmes. Insérez dans ce cas dans le programme la commande GOSUB 50000.

Tapez les quelques lignes qui suivent.

```
50000 REM ** SAVE UTILITY **
50010 PRINT:PRINT
50020 PRINT "Do you want the save this picture?"
50030 PRINT "Y or N : ";
50040 REPEAT:GET G$:UNTIL G$="Y" OR G$="N"
50050 IF G$="N" THEN RETURN
50060 PRINT:PRINT
50070 PRINT "Press RECORD and PLAY on your recorder";
50080 PRINT CHR$(140);SPC(10);"Press Y when ready";
50090 REPEAT:GET G$:UNTIL G$="Y"
50100 CSAVE "PICTURE",A#A000,E#BF3F
50110 REM Change details above for 16K model
50120 PRINT:PRINT
50130 PRINT CHR$(140);SPC(8);"Turn your recorder off"
50140 PRINT CHR$(140);SPC(11);"and press Y ";
50150 REPEAT:GET G$:UNTIL G$="Y"
50160 PRINT:PRINT
50170 RETURN
```

Annexe 2

Utilitaire de renumérotation

Lorsque vous écrivez ou modifiez vos programmes, il vous arrive de vouloir renuméroter les lignes. Suit un court programme capable de renuméroter un bloc quelconque de lignes. Cependant, il n'aura pas d'effet sur les numéros de ligne appelés dans le programme. En d'autres termes, les expressions GOTO 999, GOSUB 237, etc., ne seront pas changées. Pour contourner le problème (ce qui est fait dans ce livre), utilisez des expressions de la forme GOSUB BLANK où la variable BLANK reçoit une valeur dans les premières lignes du programme. Dans ce cas, après la renumérotation, vous n'aurez à changer que la valeur de BLANK et le programme marchera.

Revenons au programme de renumérotation, tapez-le puis sauvegardez-le sous un nom, par exemple RENUMÉROTATION. Si ensuite vous tapez un programme, chargez d'abord RENUMÉROTATION puis tapez votre programme.

Pour utiliser le programme de renumérotation, tapez RUN 60000 puis RETURN. L'ordinateur vous demande d'introduire quatre nombres :

début, fin, numéro de la première ligne, incrément.

Début et *fin* sont les numéros de la première et de la dernière ligne que vous voulez renuméroter. *Numéro de la première ligne* est le nouveau numéro de la première ligne et *incrément* est la différence entre deux lignes consécutives (habituellement 10). Tapez simplement les quatre nombres séparés par des virgules puis RETURN. Le programme renumérottera alors les lignes comme vous l'avez demandé.

```
60000 REM ** RENUMBER **
60010 PRINT "Type in 4 numbers, separated by commas"
60020 PRINT "Block start, finish, New line no., inc"
60030 INPUT X,Y,L,I
60040 S=1283
60050 REPEAT
60060 : IF DEEK(S)>X THEN DOKE S,L:L=L+I
60070 : S=DEEK(S-2)+2
60080 UNTIL DEEK(S)>Y OR DEEK(S)=60000
```

Attention : soyez attentif lorsque vous utilisez ce programme de renumérotation. Vous pouvez détruire votre programme par manque de bon sens. Assurez-vous que les lignes que vous renumérotez resteront toujours comprises entre les autres lignes. Par exemple, regardez le programme suivant.

```
10 REM SAMPLE PROGRAM
12 FOR I=1 TO 10
14 PRINT I
16 PING
18 WAIT 30
20 NEXT
```

Tapez RUN 60000 et entrez les nombres suivants :

12, 16, 10, 10

Cela va renuméroter les lignes 12 à 16, les nouveaux numéros commençants à 10 avec un incrément de 10. Le résultat est :

```
10 REM SAMPLE PROGRAM
10 FOR I=1 TO 10
20 PRINT I
30 PING
18 WAIT 30
20 NEXT
```

Etrange, mais le programme marche encore. Vous pouvez retrouver le programme initial en utilisant RUN 60000 et en entrant les nombres :

10, 100, 10, 2

Revenez au programme initial et essayez RUN 60000 avec les nombres

10, 20, 10, 0

Le résultat est :

```
10 REM SAMPLE PROGRAM
10 FOR I=1 TO 10
10 PRINT I
10 PING
10 WAIT 30
10 NEXT
```

Une nouvelle fois : étrange, mais le programme marche encore. Enfin, revenez au programme initial et tapez RUN 60000 avec les nombres suivants :

10, 20, 50, -10

ce qui donne :

```
50 REM SAMPLE PROGRAM
40 FOR I=1 TO 10
30 PRINT I
20 PING
10 WAIT 30
0 NEXT
```

Ce programme marche encore, mais essayez de supprimer la ligne 10 ! Il peut être très amusant de créer des programmes à l'aspect bizarre. Les problèmes apparaissent avec les instructions GOSUB et GOTO. Voici un exemple :

```
10 REM ANOTHER SAMPLE PROGRAM
12 GOSUB 20
14 PING
16 WAIT 30
18 END
20 PRINT "HELLO"
22 RETURN
```

Renumérotez ce programme (RUN 60000) avec les nombres :

10, 18, 100, 200

Le résultat est ci-dessous. Ce programme ne marchera pas.

```
100 REM ANOTHER SAMPLE PROGRAM
300 GOSUB 20
500 PING
700 WAIT 30
900 END
20 PRINT "HELLO"
22 RETURN
```


Enfin, voici le programme de Chasse au trésor renuméroté d'une façon originale. Le programme marche mais vous ne pourrez le modifier, éditer une ligne est difficile (si vous voulez renuméroter vos programmes de cette façon, renumérotez d'abord les lignes de numéros élevés puis continuez en décroissant).

```

0 REM *****
0 REM * TREASURE HUNT *
0 REM *****
0 REM
0 VIEW=1:SETUP=2:INFO=3
0 REM * THE MAIN PROGRAM *
0 GOSUB INFO:GOSUB SETUP
0 REPEAT
0 : PLOT X,Y,12:GET K$:KK=ASC(K$):PLOT X,Y,8
0 : IF KK=32 THEN GOSUB VIEW
0 : IF KK=8 AND X>4 THEN X=X-2
0 : IF KK=9 AND X<30 THEN X=X+2
0 : IF KK=10 AND Y<22 THEN Y=Y+2
0 : IF KK=11 AND Y>2 THEN Y=Y-2
0 UNTIL STATE
0 SHOOT:WAIT 40:PING:WAIT 40:PING
0 PRINT CHR$(14)+E$+"L";SPC(11);"WELL DONE"
0 PRINT E$+"A"+"      Another go? Y or N"
0 PRINT CHR$(14)+CHR$(11);:PING
0 REPEAT:GET K$:UNTIL K$="Y" OR K$="N"
0 IF K$="Y" THEN RUN
0 CLS:PING:PRINT "Bye for now.":POKE #26A,3:END
1 REM * VIEW *
1 IF SCRN(X+1,Y)<>35 THEN RETURN
1 DIST=INT(SQR((MM-X)*(MM-X)+(NN-Y)*(NN-Y))/2)
1 A$=CHR$(48+DIST)
1 IF DIST>9 THEN A$="X"
1 IF DIST>0 AND RND(1)<.5 THEN A$="?"
1 PLOT X+1,Y,A$:STATE=DIST=0
1 RETURN
2 REM * SETUP *
2 CLS:PRINT SPC(11);"TREASURE HUNT":PRINT
2 FOR I=1 TO 11
2 : PRINT SPC(3);
2 : FOR J=0 TO 13

```

```

2 : PRINT CHR$(27)+"H#";
2 : NEXT
2 : PRINT:PRINT
2 NEXT
2 PRINT E$+"A    Use cursor keys to move."
2 PRINT:PRINT E$+"D    Press <Space bar> to view.";
2 PRINT CHR$(11);CHR$(11);CHR$(11)
2 X=16:Y=12 'INITIAL POSITION
2 MM=2*INT(RND(9)*14+1)+2:NN=2*INT(RND(9)*11+1)
2 STATE=FALSE
2 RETURN
3 REM * INFORMATION *
3 POKE #26A,10 'Disables cursor and key click
3 INK 0:PAPER 3:CLS:PRINT
3 E$=" "+CHR$(27):D$=E$+"J"+CHR$(4)
3 PRINT D$;SPC(9);"TREASURE HUNT":PRINT D$
3 PRINT:PRINT
3 PRINT D$;SPC(6);"by Czes Kosniowski":PRINT D$
3 PRINT:PRINT E$+"A Find the hidden treasure."
3 PRINT "By pressing the <space bar> you will"
3 PRINT "be told how far away the treasure is"
3 PRINT "approximately. Sometimes a ? appears"
3 PRINT "to confuse you. An X means that the"
3 PRINT "distance is 10 or more.    GOOD LUCK"
3 PRINT:PRINT E$+"B"+E$+"L . Press Y to continue."
3 REPEAT:K$=KEY$:K=RND(1):UNTIL K$="Y"
3 RETURN

```

Annexe 3

Notes de musique

Les commandes MUSIC nécessitent quatre nombres sur l'Oric.

Canal, Octave, Note, Volume

Note peut prendre une valeur quelconque entre 1 et 12; les autres nombres entraînent un message d'erreur. Ces nombres correspondent approximativement aux notes suivantes :

1 = do	2 = do#	3 = ré	4 = ré#	5 = mi	6 = fa
7 = fa#	8 = sol	9 = sol#	10 = la	11 = la#	12 = si

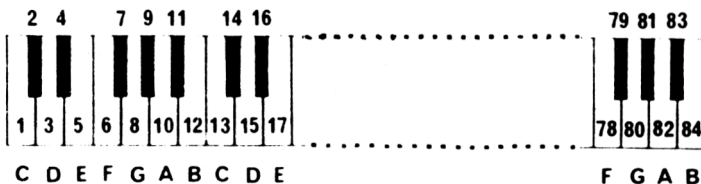
Pour jouer un air sur l'Oric, les données doivent avoir été introduites, généralement dans des intructions DATA. Les choses sont assez faciles si vous restez sur le même octave. Cependant, vous voulez souvent passer d'un octave à l'autre. L'utilisation des DATA pour indiquer l'octave voulu n'est pas nécessaire et prendrait trop de place. Cette annexe donne une méthode simple pour combiner les valeurs de l'octave et de la note en un seul nombre.

L'idée est d'utiliser les nombres 1 à 12 pour le premier octave, les nombres 13 à 24 pour le second et ainsi de suite. Ainsi un nombre, noté R, donne à la fois la valeur de l'octave et de la note. Pour retrouver la valeur OCT de l'octave et celle NN de la note, utilisez les lignes qui suivent.

$$OCT = INT(R-1)/12$$

$$NN = R - 12 * OCT$$

Avec ce schéma vous pouvez imaginer un clavier de piano. Entrer les notes devient ainsi facile.



Bien sûr, nous n'avons pas tout dit. Habituellement, nous mettons la commande de volume à 0 et nous utilisons un ordre tel que :

PLAY 3, 0, 1, 2500 : WAIT T

Ceci contrôle la façon dont le son est produit (son mode) et le temps T pendant lequel la note est jouée. De cette façon, nos instructions DATA devraient contenir deux nombres R et T. La valeur de R indique à l'*Oric* quelle est la note et son octave alors que T indique sa durée.

Ce procédé donnera une musique acceptable ; vous pouvez bien sûr être beaucoup plus aventureux...

LA BIBLIOTHÈQUE EDIMICRO

● *Collection « Ordinateurs personnels »*

ATMOS/ORIC 1

Kosniowsky	Nouveaux Jeux sur ATMOS
Chane-Hune, Darbois	Jeux Graphiques sur ATMOS
Bayvejiel	Guide de l'Oric
Chane-Hune, Darbois	Jeux sur ORIC
Viguiier	Premiers pas en programmation sur Oric

COMMODORE 64

Fleurier, Meiller	Jeux sur Commodore 64 : Jeux d'adresse et de hasard
Fleurier, Meiller	Jeux sur Commodore 64 : Jeux d'action et de réflexion

ELECTRON

Bennani, Chaieb	Graphisme et sons sur Electron
-----------------------	--------------------------------

M05

Bieber, Perbost, Renucci	Tout sur le M05
Perbost, Renucci	Jeux sur M05

SPECTRUM

Bridge, Carnell	Aventures sur Spectrum
Hurley	Jeux Graphiques sur Spectrum

T07

— ouvrages

Bieber, Perbost, Renucci	Guide du T07
Perbost, Renucci	Jeux sur T07

— logiciels sur cassette

Perbost, Renucci	4 jeux avec manette pour T07
Perbost, Renucci	6 jeux d'action et de réflexion pour T07

VG 5000 PHILIPS

Amsler, Bardon	Guide du VG 5000 Philips
Amsler, Villemaud	Jeux sur VG 5000 Philips

PHILIPS C7420 VIDEOPAC +

Bardon, de Merly	Jeux sur Philips C7420 Vidéopac +
------------------------	--------------------------------------

● **Collection « Ordinateurs professionnels »**

APPLE

de Merly	Guide de l'Apple Tome 1 : l'Apple standard Tome 2 : les extensions Tome 3 : les applications
Bonnet, Dinh	Multiplan sur Apple

IBM PC

Bonnet, Dinh	Multiplan sur IBM PC
--------------------	-----------------------------

● **Collection « langages »**

Gaucherand, Lamoitier	Fichiers en Basic par l'Exemple
-----------------------------	--

● **« Intérêt général »**

de Merly	Ordinateur Familial : que choisir ?
Lamoitier	Le Traducteur Micro
Bonnet, Dinh	Mémento Multiplan
Darnis, Van Thong	Graphisme et CAO

EXTRAITS DE PRESSE

GUIDE DE L'APPLE

« Un des meilleurs sur la place. »

Le Figo.

« Le livre que nous attendions : complet, clair et pratique. »

Apple France.

« Toutes les étapes deviennent plus claires. »

Golden.

« Exposés clairs, à la portée de ceux qui ont des connaissances élémentaires en informatique. »

Minis et Micros.

MULTIPLAN SUR APPLE

« Nul doute qu'avec un tel outil vous gagnerez en temps et en efficacité. »

Microsoft France.

« Une bonne introduction... Un bon guide... Un bon investissement. »

Science et Vie Micro.

GUIDE DU TO 7

« Ce guide a le privilège d'être présenté de façon claire, dans un style parfaitement accessible à tous. »

Micro 7.

« Un manuel de référence absolue. »

Le Figo.

JEUX SUR TO 7

« De nombreux conseils et "trucs" pour une programmation rapide et efficace. »

Décision informatique.

« Un excellent livre d'approche, plein de renseignements utiles. »

Micro 7.

GUIDE DE L'ORIC

« Tout ce qu'il faut savoir pour tirer le meilleur parti des possibilités — en particulier sonores et graphiques — de l'ORIC. »

Microsystèmes.

« Un best-seller. »

Livres Hebdo.

PREMIERS PAS EN PROGRAMMATION SUR ORIC

« Les exemples sont clairs et bien choisis, la méthode simple et rationnelle. »

MICR'ORIC, revue des utilisateurs de l'ORIC.

Nombreux autres titres à paraître. Catalogue sur simple demande.

EDIMICRO 121-127, avenue d'Italie, 75013 Paris

Achévé d'imprimer en septembre 1984
sur les presses de l'imprimerie Laballery et C^{ie}
58500 Clamecy
Dépôt légal : septembre 1984
N° d'imprimeur : 406023

NOUVEAUX JEUX sur ATMOS

C. Kosniowski

Serez-vous capable :

- d'échapper à une meute de loup ?
- de chanter des airs de musique populaire ?
- de dresser un serpent ?
- de piloter un module lunaire ?

Voici **30 PROGRAMMES DE JEUX VARIÉS** : jeux de pur divertissement **qui utilisent le son et la couleur**, puzzles, jeux d'arcade, problèmes de réflexion.

Vous trouverez en annexe une section très utile :

- programme de **SAUVEGARDE** de l'écran
- programme utilitaire de **RENUMEROTATION**
- **MUSIQUE** sur Atmos

Les programmes sont clairement décrits et commentés.
De courtes descriptions et des représentations de l'écran sont fournies.

ISBN : 2-904457-18-6

Prix : 88 F.



Edimicro

121-127 Avenue d'Italie 75013 Paris

Edimicro

NOVEL AXES FOR ATOMS

C. KOSNIOWSKI