

SCIENCE & VIE

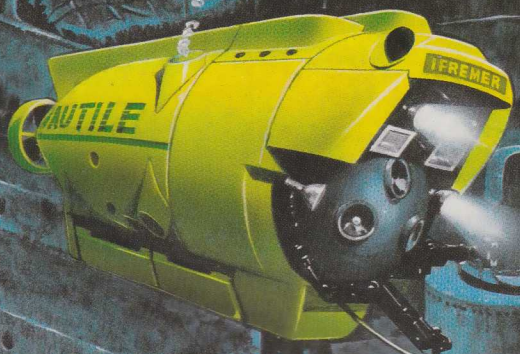
MENSUEL

N° 842 NOVEMBRE 1987

LE RETOUR DU TITANIC

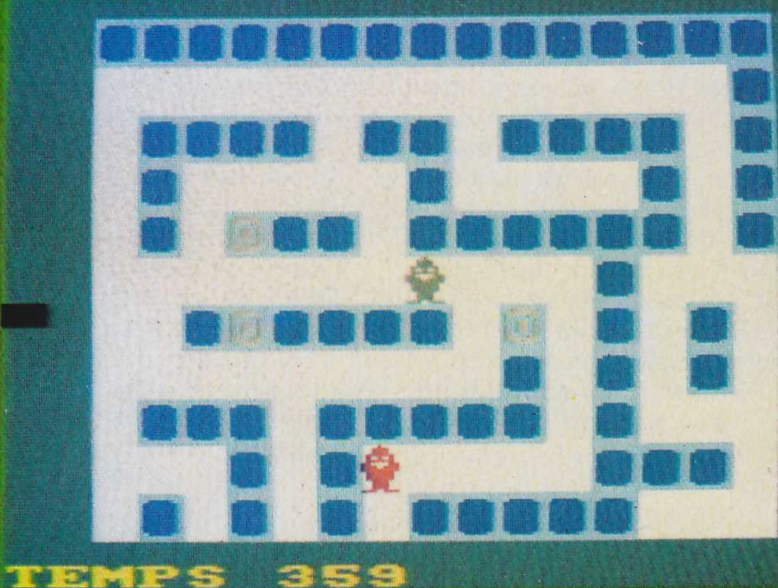
**NUCLÉAIRE :
QUAND EDF
SE TROMPE
DE ROBINET**

**L'AVION
INVISIBLE**



M 2578 - 842 - 17,00 F





Panique au Pôle

INFORMATIQUE AMUSANTE

Voici un jeu d'animation inspiré d'un jeu d'arcades : deux pingouins travaillent au pôle, à classer des glaçons. Arrivent, un jour, trois glaçons renfermant des pépites d'or. Pour profiter de ce trésor, les pingouins doivent aligner les trois glaçons contenant les pépites.

L'un des pingouins, celui que commande le joueur, a fait cette découverte le premier et fera la première tentative, pendant que le second, rouge de colère, entre dans la réserve de glace et commence à tout casser. Le joueur devra donc piloter son pingouin de manière à lui faire aligner les glaçons en dépit du désordre semé par le second pingouin. Enfin, si le pingouin en colère parvient à attraper l'autre, la partie sera perdue.

Ces quelques règles une fois fixées, passons à l'écriture du programme. Nous utiliserons ici le mode 0 de manière à profiter de l'intégralité de la palette couleurs de l'Amstrad : ce changement de mode est assuré par la ligne 10 du programme.

Ensuite, nous ouvrirons, à la ligne 50, un tableau de mémorisa-

tion du jeu; celui-ci aura pour fonction de repérer en permanence la position de chaque glaçon durant le déroulement de la partie. Les principales variables seront ensuite initialisées des lignes 90 à 140. HJ et VJ correspondront à la position du pingouin du joueur, DR à la direction du déplacement souhaité, HC, VC et DC assureront les mêmes fonctions pour le pingouin en colère, TP sera utilisé pour le décompte du temps de jeu disponible, FIN, SC et SND permettront respectivement de contrôler la fin de la partie, le score et certaines illustrations sonores. En ce qui concerne le son, diverses enveloppes seront fixées des lignes 180 à 210. Viennent ensuite de nombreuses lignes de données mémorisées à l'aide de l'instruction DATA (lignes 250 à 570). Ces lignes auront pour but de positionner l'ensemble des blocs de glace sur l'écran en début de partie. Les symboles graphiques nécessaires au jeu seront ensuite redéfinis par les lignes 610 à 670.

L'ensemble des données nécessaires étant ainsi mémorisées, le jeu pourra donc commencer. En

premier lieu, les données mémorisées plus haut sous forme de DATA permettant de représenter trois situations de départ différentes, un tableau de jeu sera tiré au hasard puis présenté sur l'écran. Ce tirage et son affichage seront assurés par les lignes 710 à 820.

Ensuite le programme s'intéressera aux déplacements du joueur. Pour cela la direction souhaitée pour le déplacement sera prise en compte grâce aux instructions INKEY placées des lignes 880 à 910. En fonction des touches enfoncées, le pingouin sera déplacé sur l'écran et, si sa route est barrée par un glaçon, la sous-routine 1390 sera utilisée pour savoir si celui-ci

doit être détruit ou poussé. Enfin, le programme contrôlera si la partie est terminée ou non. Les lignes 1100 à 1150 seront utilisées à cet effet. Nous analyserons ici la valeur prise par la variable FIN ; si elle vaut 1, le joueur aura perdu, si elle vaut 2 il aura gagné. Notons que tant que FIN vaudra 0 le jeu rebouclera grâce à la présence de la ligne 1190.

Diverses sous-routines sont également utilisées pour le déroulement de ce jeu ; passons donc maintenant à leur écriture.

La première d'entre elles, placée des lignes 1280 à 1370, concerne la mise en place des glaçons contenant une pépite en début de jeu.

Leur positionnement sera aléatoire et commandé par les variables HD et VD. Cependant, afin d'éviter de détruire l'effet de labyrinthe présenté par le jeu, nous vérifierons, à la ligne 1320, si la position ainsi choisie pour placer une pépite correspond bien à celle d'un glaçon déjà positionné. Cette sous-routine rebouclera trois fois, une fois par pépite, et, chaque fois, la case correspondante du tableau de mémorisation du jeu prendra la valeur 2. Ceci permettra par la suite au programme de distinguer les glaçons "standard" de ceux "de valeur".

La seconde sous-routine, placée des lignes 1410 à 1870, a pour mission de contrôler le déplacement

```

10 MODE 0:BORDER 9:PAPER 8:CLS:CLEAR
20 REM *****
30 REM * TABLEAU DE MEMORISATION DE L'ETAT DU JEU *
40 REM *****
50 DIM J(15,11)
60 REM *****
70 REM * INITIALISATION DES VARIABLES PRINCIPALES *
80 REM *****
90 LET HJ=8:LET VJ=6:LET DR=0
100 LET HC=8:LET VC=10:LET DC=4
110 LET TP=360
120 LET FIN=0
130 LET SC=0
140 LET SND=0
150 REM *****
160 REM * DETERMINATION DES ENVELOPPES SONORES *
170 REM *****
180 ENV 1,1,11,1,2,-4,1,3,-1,1
190 ENV 2,3,1,1,4,3,2,1,-15,1
200 ENV 3,1,15,1,10,-1,1,5,-1,1
210 ENT 1,1,1,1,10,-2,1,1,0,1
220 REM *****
230 REM * DONNES DU POSITIONNEMENT DES BLOCS DE CLACE *
240 REM *****
250 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
260 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
270 DATA 0,1,1,1,1,0,1,1,0,1,1,1,1,0,1
280 DATA 0,1,0,0,0,0,0,0,1,0,0,0,0,1,0,1
290 DATA 0,1,0,1,1,1,0,1,1,1,1,1,1,0,1
300 DATA 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
310 DATA 0,0,1,1,1,1,1,0,1,0,1,0,1,0,1,0
320 DATA 0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0
330 DATA 0,1,1,1,0,1,1,1,1,1,0,1,0,0,0,0
340 DATA 0,0,0,1,0,1,0,0,0,0,0,1,1,1,0,1,0
350 DATA 0,1,0,1,0,1,0,1,1,1,1,1,1,0,0,0
360 DATA 0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1
370 DATA 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
380 DATA 0,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1
390 DATA 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1
400 DATA 0,1,0,1,1,1,1,0,1,1,1,1,1,0,1,0
410 DATA 0,1,0,1,0,0,0,0,0,0,0,0,0,1,0,1
420 DATA 0,1,0,1,1,1,1,1,1,1,0,1,0,1,0,1
430 DATA 0,1,0,0,0,0,0,0,0,0,0,0,1,0,1,0
440 DATA 0,1,0,1,1,1,0,1,1,1,1,1,1,0,1,0
450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
460 DATA 1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1
470 DATA 0,0,0,0,1,0,1,0,1,1,1,1,1,1,1,0
480 DATA 1,1,1,0,1,0,1,0,0,0,0,0,0,0,0,0
490 DATA 0,0,0,0,1,0,1,0,1,1,1,1,1,1,1,1
500 DATA 1,1,1,0,0,0,1,0,1,0,0,0,0,0,0,0
510 DATA 0,0,0,0,1,1,1,0,1,0,1,1,1,1,1,1
520 DATA 1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
530 DATA 1,0,1,0,1,1,1,0,1,1,1,1,1,1,0,1
540 DATA 1,0,1,0,1,0,1,0,0,0,0,0,0,0,1
550 DATA 1,0,1,0,1,0,1,0,1,1,1,1,1,1,1,1
560 DATA 1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
570 DATA 0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1
580 REM *****
590 REM * DETERMINATION DES SYMBOLES GRAPHIQUES *
600 REM *****
610 SYMBOL AFTER 127
620 SYMBOL 128,0,0,60,126,126,126,126,126
630 SYMBOL 129,126,126,126,126,126,126,60,0,0
640 SYMBOL 130,0,0,60,126,102,66,90,90
650 SYMBOL 131,90,90,66,102,126,60,0,0
660 SYMBOL 132,16,56,56,124,84,124,198,238
670 SYMBOL 133,254,124,124,124,58,58,58,238
680 REM *****
690 REM * TIRAGE ALEATOIRE DU TABLEAU DE JEU *
700 REM *****
710 LET TB=INT(RND*3)
720 IF TB=0 THEN RESTORE 250
730 IF TB=1 THEN RESTORE 360
740 IF TB=2 THEN RESTORE 470
750 PEN 8
760 FOR V=1 TO 11
770 FOR H=1 TO 15
780 READ X:LET J(H,V)=X
790 PAPER 4:LOCATE H+2,2*V:PRINT " ":LOCATE H+2,(2*V)+1:PRINT " "
800 IF X=1 THEN LOCATE H+2,2*V:PAPER 2:PRINT CHR$(128):LOCATE H+2,(2*V)+1:PRINT CHR$(129):SOUND 1,24,0,0,1,0,0:SOUND 2,2,0,0,1,0,0
810 NEXT H
820 NEXT V
830 GOSUB 1280
840 REM *****
850 REM * PRISE EN COMPTE DES DEPLACEMENTS DU JOUEUR *
860 REM *****
870 LET DR=0
880 IF INKEY(0)=0 THEN LET DR=1
890 IF INKEY(2)=0 THEN LET DR=3
900 IF INKEY(1)=0 THEN LET DR=2
910 IF INKEY(8)=0 THEN LET DR=4
920 IF DR<>0 THEN LOCATE HJ+2,2*VJ:PAPER 4:PRINT " ":LOCATE HJ+2,(2*VJ)+1:PRINT " "
930 IF DR=1 AND VJ>1 THEN LET VJ=VJ-1
940 LET HG=HJ:LET VG=VJ
950 IF DR=1 AND J(HJ,VJ)<>0 THEN LET VJ=VJ+1:IF INKEY(9)=0 THEN GOSUB 1390
960 IF DR=3 AND VJ<11 THEN LET VJ=VJ+1
970 LET HG=HJ:LET VG=VJ
980 IF DR=3 AND J(HJ,VJ)<>0 THEN LET VJ=VJ-1:IF INKEY(9)=0 THEN GOSUB 1390
990 IF DR=4 AND HJ>1 THEN LET HJ=HJ-1
1000 LET HG=HJ:LET VG=VJ
1010 IF DR=4 AND J(HJ,VJ)<>0 THEN LET HJ=HJ+1:IF INKEY(9)=0 THEN GOSUB 1390
1020 IF DR=2 AND HJ<15 THEN LET HJ=HJ+1
1030 LET HG=HJ:LET VG=VJ
1040 IF DR=2 AND J(HJ,VJ)<>0 THEN LET HJ=HJ-1:IF INKEY(9)=0 THEN GOSUB 1390
1050 PAPER 4:PEN 9:LOCATE HJ+2,2*VJ:PRINT CHR$(132):LOCATE HJ+2,(2*VJ)+1:PRINT CHR$(133)
1060 GOSUB 1910
1070 REM *****
1080 REM * CONTROLE DE FIN DE LA PARTIE *
1090 REM *****
1100 IF FIN<>0 THEN LOCATE 5,8:PRINT " "
1110 IF FIN=1 THEN LOCATE 5,10:PRINT " PERDU !!! "
1120 IF FIN=2 THEN LOCATE 5,10:PRINT " BRAVO "
1130 IF FIN<>0 THEN LOCATE 5,11:PRINT " "
1140 IF FIN=2 THEN LOCATE 1,25:PRINT "SCORE:";(TP*100)+SC
1150 IF FIN<>0 THEN LOCATE 1,2:PRINT " TAPER 'P' S.V.P "
1160 GOTO 1230
1160 REM *****
1170 REM * REBOUCLAGE DU JEU *
1180 REM *****
1190 GOTO 870
1200 REM *****
1210 REM * DEMANDE DE NOUVELLE PARTIE *
1220 REM *****
1230 IF UPPER$(INKEY$)<>"P" THEN GOTO 1230
1240 GOTO 10
1250 REM *****
1260 REM * SOUS-ROUTINE DE MISE EN PLACE DES PEPITES *
1270 REM *****
1280 PAPER 2:PEN 14
1290 FOR I=1 TO 3
1300 LET HD=INT (RND*13)+2
1310 LET VD=INT (RND*9)+2
1320 IF J(HD,VD)<>1 THEN GOTO 1300
1330 LOCATE HD+2,2*VD:PRINT CHR$(130):LOCATE HD+2,(2*VD)+1:PRINT CHR$(131)
1340 SOUND 1,0,0,0,2,0,15
1350 LET J(HD,VD)=2

```

des glaçons. Dans chaque cas, la direction choisie sera prise en compte et, pour chaque déplacement, le programme commencera par contrôler, dans le tableau de mémorisation de la partie, si le déplacement est possible, puis modifiera son contenu en conséquence. Notons enfin que cette sous-routine comporte deux parties. La première s'adresse aux glaçons standard, la seconde aux glaçons contenant une pépite.

Une troisième sous-routine assure les déplacements du pingouin en colère. Celle-ci est placée des lignes 1910 à 2100. La direction de son déplacement sera tirée au hasard et pourra être modifiée cha-

que fois qu'il heurtera ou poussera un glaçon. De même, cette sous-routine fera régulièrement appel à la précédente de manière à ce que le pingouin en colère soit également en mesure de pousser les glaçons.

Enfin, la dernière sous-routine a pour mission de contrôler l'alignement, horizontal ou vertical, des glaçons à pépites. Placée des lignes 2140 à 2280, elle effectuera ses contrôles grâce aux informations contenues dans le tableau de mémorisation du jeu. Si les trois glaçons sont effectivement alignés, la valeur 2 sera donnée à FIN.

L'utilisation de ce jeu est extrêmement simple. Une fois le pro-

gramme entièrement frappé, RUN sera demandé. L'écran affichera alors le labyrinthe de début de partie. Les déplacements du pingouin seront obtenus en utilisant les quatre flèches de déplacement du curseur. Pour pousser un glaçon, il faudra appuyer simultanément sur la flèche correspondant à la direction souhaitée et sur la touche centrale marquée COPY.

Notons enfin que si un autre glaçon bloque le glaçon à pousser, ce dernier sera détruit sauf s'il contient une pépite. Dans ce dernier cas, la touche COPY restera sans effet. En fin de jeu, une nouvelle partie pourra être demandée en tapant "P". **Henri-Pierre Penel**

```

1360 NEXT I
1370 RETURN
1380 REM *****
1390 REM * SOUS-ROUTINE DE DEPLACEMENT BLOCS DE CLACE *
1400 REM *****
1410 IF DR=1 AND VG=1 AND J(HG, VG)=1 THEN LET J(HG, VG)=0: PAPER
ER 4: LOCATE HG+2, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT "
": RETURN
1420 IF DR=3 AND VG=11 AND J(HG, VG)=1 THEN LET J(HG, VG)=0: PAPER
4: LOCATE HG+2, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT "
": RETURN
1430 IF DR=2 AND HG=15 AND J(HG, VG)=1 THEN LET J(HG, VG)=0: PAPER
4: LOCATE HG+2, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT "
": RETURN
1440 IF DR=4 AND HG=1 AND J(HG, VG)=1 THEN LET J(HG, VG)=0: PAPER
ER 4: LOCATE HG+2, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT "
": RETURN
1450 IF SND=0 THEN SOUND 1, 20, 10, 6, 0, 1, 0: LET SC=SC+20
1460 IF J(HG, VG)<>1 THEN GOTO 1690
1470 LET GL=0
1480 IF DR<>1 THEN GOTO 1530
1490 IF J(HG, VG-1)=0 THEN LET J(HG, VG)=0: PAPER 4: LOCATE HG+2
, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT " ": PAPER 2: PEN 6
: LET VG=VG-1
: LOCATE HG+2, 2*VG: PRINT CHR$(128): LOCATE HG+2, (2*VG)+1: PRINT
CHR$(129): LET GL=1: IF VG>1 THEN GOTO 1490
1500 LET J(HG, VG)=1: SOUND 2, 0, 0, 0, 3, 0, 1
1510 IF VG=1 THEN RETURN
1520 IF J(HG, VG-1)<>0 AND VG>1 AND GL=0 THEN LET J(HG, VG)=0:
PAPER 4: LOCATE HG+2, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRIN
T " "
1530 IF DR<>3 THEN GOTO 1580
1540 IF J(HG, VG+1)=0 THEN LET J(HG, VG)=0: PAPER 4: LOCATE HG+2
, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT " ": LET VG=VG+1: P
APER 2: PEN 6
: LOCATE HG+2, 2*VG: PRINT CHR$(128): LOCATE HG+2, (2*VG)+1: PRINT
CHR$(129): LET GL=1: IF VG<11 THEN GOTO 1540
1550 LET J(HG, VG)=1: SOUND 2, 0, 0, 0, 3, 0, 1
1560 IF VG=11 THEN RETURN
1570 IF J(HG, VG+1)<>0 AND VG<11 AND GL=0 THEN LET J(HG, VG)=0:
PAPER 4: LOCATE HG+2, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRI
NT " "
1580 IF DR<>2 THEN GOTO 1630
1590 IF J(HG+1, VG)=0 THEN LET J(HG, VG)=0: PAPER 4: LOCATE HG+2
, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT " ": LET HG=HG+1: P
APER 2: PEN 6
: LOCATE HG+2, 2*VG: PRINT CHR$(128): LOCATE HG+2, (2*VG)+1: PRINT
CHR$(129): LET GL=1: IF HG<15 THEN GOTO 1590
1600 LET J(HG, VG)=1: SOUND 2, 0, 0, 0, 3, 0, 1
1610 IF HG=15 THEN RETURN
1620 IF J(HG+1, VG)<>0 AND HG<15 AND GL=0 THEN LET J(HG, VG)=0:
PAPER 4: LOCATE HG+2, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRI
NT " "
1630 IF DR<>4 THEN GOTO 1680
1640 IF J(HG-1, VG)=0 THEN LET J(HG, VG)=0: PAPER 4: LOCATE HG+2
, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT " ": LET HG=HG-1: P
APER 2: PEN 6
: LOCATE HG+2, 2*VG: PRINT CHR$(128): LOCATE HG+2, (2*VG)+1: PRINT
CHR$(129): LET GL=1: IF HG>1 THEN GOTO 1640
1650 LET J(HG, VG)=1: SOUND 2, 0, 0, 0, 3, 0, 1
1660 IF HG=1 THEN RETURN
1670 IF J(HG-1, VG)<>0 AND HG>1 AND GL=0 THEN LET J(HG, VG)=0:
PAPER 4: LOCATE HG+2, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRIN
T " "
1680 RETURN
1690 IF J(HG, VG)<>2 THEN RETURN
1700 LET GL=0
1710 IF DR<>1 THEN GOTO 1750
1720 IF J(HG, VG-1)=0 THEN LET J(HG, VG)=0: PAPER 4: LOCATE HG+2
, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT " ": LET VG=VG-1: P
APER 2: PEN 1
4: LOCATE HG+2, 2*VG: PRINT CHR$(130): LOCATE HG+2, (2*VG)+1: PRIN
T CHR$(131): LET GL=1: IF VG>1 THEN GOTO 1720

```

```

1730 LET J(HG, VG)=2: SOUND 2, 0, 0, 0, 3, 0, 1: GOSUB 2140
1740 IF VG=1 THEN RETURN
1750 IF DR<>3 THEN GOTO 1790
1760 IF J(HG, VG+1)=0 THEN LET J(HG, VG)=0: PAPER 4: LOCATE HG+2
, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT " ": LET VG=VG+1: P
APER 2: PEN 1
4: LOCATE HG+2, 2*VG: PRINT CHR$(130): LOCATE HG+2, (2*VG)+1: PRIN
T CHR$(131): LET GL=1: IF VG<11 THEN GOTO 1760
1770 LET J(HG, VG)=2: SOUND 2, 0, 0, 0, 3, 0, 1: GOSUB 2140
1780 IF VG=11 THEN RETURN
1790 IF DR<>2 THEN GOTO 1830
1800 IF J(HG+1, VG)=0 THEN LET J(HG, VG)=0: PAPER 4: LOCATE HG+2
, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT " ": LET HG=HG+1: P
APER 2: PEN 1
4: LOCATE HG+2, 2*VG: PRINT CHR$(130): LOCATE HG+2, (2*VG)+1: PRIN
T CHR$(131): LET GL=1: IF HG<15 THEN GOTO 1800
1810 LET J(HG, VG)=2: SOUND 2, 0, 0, 0, 3, 0, 1: GOSUB 2140
1820 IF HG=15 THEN RETURN
1830 IF DR<>4 THEN RETURN
1840 IF J(HG-1, VG)=0 THEN LET J(HG, VG)=0: PAPER 4: LOCATE HG+2
, 2*VG: PRINT " ": LOCATE HG+2, (2*VG)+1: PRINT " ": LET HG=HG-1: P
APER 2: PEN 1
4: LOCATE HG+2, 2*VG: PRINT CHR$(130): LOCATE HG+2, (2*VG)+1: PRIN
T CHR$(131): LET GL=1: IF HG>1 THEN GOTO 1840
1850 LET J(HG, VG)=2: SOUND 2, 0, 0, 0, 3, 0, 1: GOSUB 2140
1860 IF HG=1 THEN RETURN
1870 RETURN
1880 REM *****
1890 REM * SOUS-ROUTINE DE DEPLACEMENT DU PINGOUIN FOU *
1900 REM *****
1910 GOTO 1980
1920 LET SND=1
1930 LET DR=DC: IF DC=1 THEN LET HG=HC: LET VG=VC-1: GOSUB 1390
1940 LET DR=DC: IF DC=3 THEN LET HG=HC: LET VG=VC+1: GOSUB 1390
1950 LET DR=DC: IF DC=2 THEN LET HG=HC+1: LET VG=VC: GOSUB 1390
1960 LET DR=DC: IF DC=4 THEN LET HG=HC-1: LET VG=VC: GOSUB 1390
1970 LET DC=INT(RND*4)+1: LET SND=0
1980 IF DC=1 AND (VC<4 OR J(HC, VC-1)<>0) THEN GOTO 1920
1990 IF DC=3 AND (VC>8 OR J(HC, VC+1)<>0) THEN GOTO 1920
2000 IF DC=2 AND (HC>12 OR J(HC+1, VC)<>0) THEN GOTO 1920
2010 IF DC=4 AND (HC<4 OR J(HC-1, VC)<>0) THEN GOTO 1920
2020 PAPER 4: LOCATE HC+2, 2*VC: PRINT " ": LOCATE HC+2, (2*VC)+1
: PRINT " "
2030 IF DC=1 THEN LET VC=VC-1
2040 IF DC=3 THEN LET VC=VC+1
2050 IF DC=2 THEN LET HC=HC+1
2060 IF DC=4 THEN LET HC=HC-1
2070 PAPER 4: PEN 3: LOCATE HC+2, 2*VC: PRINT CHR$(132): LOCATE H
C+2, (2*VC)+1: PRINT CHR$(133)
2080 LET TP=TP-1: PAPER 8: PEN 1: LOCATE 1, 25: PRINT "TEMPS": TP:
" "
2090 IF TP=0 OR (HC=HJ AND VC=VJ) THEN LET FIN =1
2100 RETURN
2110 REM *****
2120 REM * SOUS-ROUTINE CONTROLE ALIGNEMENT DES PEPITES *
2130 REM *****
2140 IF HG>2 THEN LET DH=HG-2
2150 IF HG<14 THEN LET FH=HG
2160 IF HG<=2 THEN LET DH=1
2170 IF HG>=14 THEN LET FH=13
2180 IF VG>2 THEN LET DV=VG-2
2190 IF VG<=9 THEN LET FV=VG
2200 IF VG>=9 THEN LET FV=9
2210 IF VG<=2 THEN LET DV=1
2220 FOR I=DH TO FH
2230 FOR J=DV TO FV
2240 IF J(1, J)=2 AND J(1+1, J)=2 AND J(1+2, J)=2 THEN LET FIN=
2
2250 IF J(1, J)=2 AND J(1, J+1)=2 AND J(1, J+2)=2 THEN LET FIN=
2
2260 NEXT J
2270 NEXT I
2280 RETURN

```