

**JEAN-FRANÇOIS
SEHAN**

L'ORIC A L'AFFICHE

20 PROGRAMMES EN BASIC



EDITIONS DU P.S.I.

PROGRAMMES

PROGRAMMES

L'ORIC
A L'AFFICHE
20 PROGRAMMES EN BASIC

La collection « **PROGRAMMES** » est constituée actuellement de vingt titres. Certains disposent d'une disquette pour Apple II (*).

- (*) Bibliothèque scientifique en Pascal — Hervé Haut
- (*) Mathématiques et statistiques — Hervé Haut
- (*) Modèles pratiques de décision — tomes 1 et 2 — Jean-Pierre Blanger
- Jeux, trucs et comptes pour PET/CBM — Michel Benelfoul
- Jeux, trucs et comptes pour TRS-80 — Michel Benelfoul et Denis Chauvin
- Jeux, trucs et comptes pour T07 — Michel Benelfoul
- Jeux, trucs et comptes pour TI-99/4A — Michel Benelfoul
- Jeux, trucs et comptes pour Commodore 64 — Michel Benelfoul et Cyril Cambien
- Etudes pour ZX-81 — tomes 1 et 2 — Jean-François Séhan
- Variations pour PC-1211 — Jean-François Séhan
- Récréations pour TI-57 — tomes 1 et 2 — Jacques Deconchat
- Collège, poquettes et maths — Jacques Deconchat
- 102 Programmes pour Sinclair et Timex
- 102 programmes pour Commodore 64 — Jacques Deconchat
- 102 programmes pour TI-99/4A — Jacques Deconchat
- ZX 81 et ZX Spectrum-Timex 1 000, 1 500 et 2 000 — Jacques Deconchat
- Le TI-99/4A à l'affiche — Jean-François Séhan
- Le vic à l'affiche — Jean-François Séhan
- Suites pour PC-1500 — Jean-François Séhan
- L'Oric à l'Affiche — Jean-François Séhan

Autres ouvrages relatifs à l'Oric :

- 52 programmes Oric-1 — Jacques Boigontier
- Collection « ... POUR TOUS »
- Oric-1 pour tous — Jacques Boigontier et Sophie Brébion
- Collection « ... POUR TOUS »
- Le Basic Microsoft — Ken Knecht
- Collection « LANGAGES »
- La découverte de l'Oric — par Daniel-Jean David
- Collection « MATERIELS » — série verte

RAPPELS

Les séries :

En fait, il faudrait parler de niveaux, puisque la couleur attachée à chaque ouvrage permet de situer la « force » de celui-ci selon le code suivant :

Série VERTE : ouvrage d'initiation ne nécessitant que des connaissances de base.

Série BLEUE : suppose une connaissance élémentaire du sujet traité.

Série ROUGE : ouvrage d'approfondissement, niveau de complexité moyen.

Série NOIRE : ouvrage d'approfondissement, niveau de complexité élevé.

Les collections :

Les ouvrages d'Édition du PSI, actuellement au nombre de 100, sont répartis en collections :

« **LANGAGES** », « **MATERIELS** », « **PROGRAMMES** », « **GUIDES PRATIQUES** », « **MEMENTOS** », « **UTILISATIONS DE L'ORDINATEUR** », « **LOGIGUIDES** » et pour l'initiation, outre quelques livres hors collection, « ... POUR TOUS ».

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 donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© Editions du P.S.I. Parc industriel nord, Bâtiment 9, 77200 Torcy Marne-la-Vallée 1984
ISBN : 2-86595-130-8

**JEAN-FRANÇOIS
SEHAN**

**L'ORIC
A L'AFFICHE
20 PROGRAMMES EN BASIC**



EDITIONS DU P.S.I.

1984

L'ORIC A L'AFFICHE

Du même auteur :

- Variations pour PC-1211.
- Etudes pour ZX-81 - tome I.
- Etudes pour ZX-81 - tome II.
- Suites pour PC-1500.
- Le TI-99/4A à l'affiche.
- Clefs pour ZX-81.
- Le VIC à l'affiche.
- Clefs pour ZX-SPECTRUM.

SOMMAIRE

Chapitre 1 : JEUX D'ADRESSE

Grenouille	8
Incendie	15
Autoroute	22
Slalom	27
Chenille	32
Circuit 24	38
Canadair	46
Réflexe	52

Chapitre 2 : JEUX DE REFLEXION

Chien de berger	60
Missionnaires et cannibales	66
Parités	72
Champ de carottes	79
Piranhas	84
Awari	90
Chasse au trésor	98
Tours de Hanoi	103
Taquin	109
Ayez du caractère	114

Chapitre 3 : JEUX DE HASARD

Jackpot	120
Baccara	127

AVERTISSEMENT

Si vous voulez utiliser correctement les programmes de ce livre, lisez attentivement les conseils suivants :

- Pour éviter tout problème au moment du lancement du programme par la commande RUN, appuyez sur le bouton du RESET sous la machine. Ceci afin de s'assurer que l'on est bien dans le mode TEXT, que les jeux de caractères sont bien ceux d'origine et qu'aucun POKE ne viendra troubler la bonne marche du programme. Un POKE#306, #1000 par exemple (pour accélérer la répétition des touches du clavier) ralentit de 30 % l'exécution du programme et bloque la machine lors de l'utilisation de l'instruction WAIT.
- Si vous n'obtenez pas les mêmes résultats que sur les exemples, vérifiez bien toutes les lignes et surtout celles contenant des DATA.
- Pour gagner en rapidité d'exécution dans les jeux d'adresse, toutes les définitions de caractères graphiques sont placées en fin de listing bien qu'elles soient utilisées en priorité. Dans le même souci, vous pouvez trouver certaines instructions NEXT non suivies d'une variable (gain de 20 à 30 %).
- Dès que les programmes sont entrés dans l'ORIC, sauvez-les sur votre magnétophone. Pour les programmes très longs, il est vivement conseillé d'effectuer une première sauvegarde au milieu de leur frappe. N'attendez pas qu'ils soient entièrement entrés et testés... une coupure de courant est si vite arrivée !

CHAPITRE 1

JEUX D'ADRESSE

GRENOUILLE

Si vous croyez que la vie d'une grenouille est de tout repos, vous allez vite changer d'idée. Car l'héroïne de ce programme s'est endormie sur son nénuphar et rêve qu'elle se trouve au bord d'une route et d'une rivière, qu'elle doit traverser pour regagner les rochers salutaires de l'autre rive.

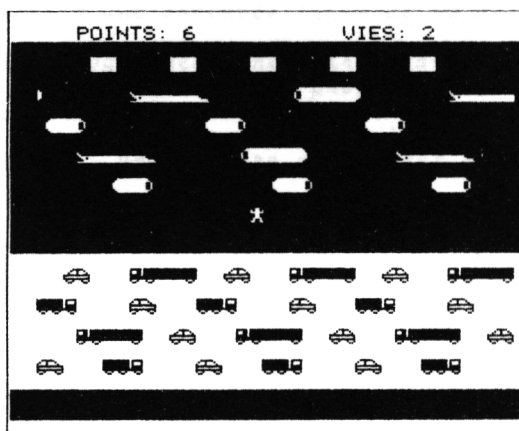
Après avoir lancé le programme, vous pouvez manoeuvrer votre grenouille latéralement avec les flèches de déplacement à gauche et à droite. Vous devez la faire sauter en utilisant la barre d'espacement. Attention à bien viser entre les véhicules qui défilent impitoyablement, sinon vous perdrez à coup sûr une des trois vies qui vous sont allouées à chaque début de partie.

Ouf ! vous êtes sur le bord de la rivière. Il vous faut maintenant emprunter un tronc d'arbre, ou un crocodile, pour aller sauter sur l'un des rochers. Mais alors la pauvre grenouille se retrouve à nouveau sur le bord de la route.

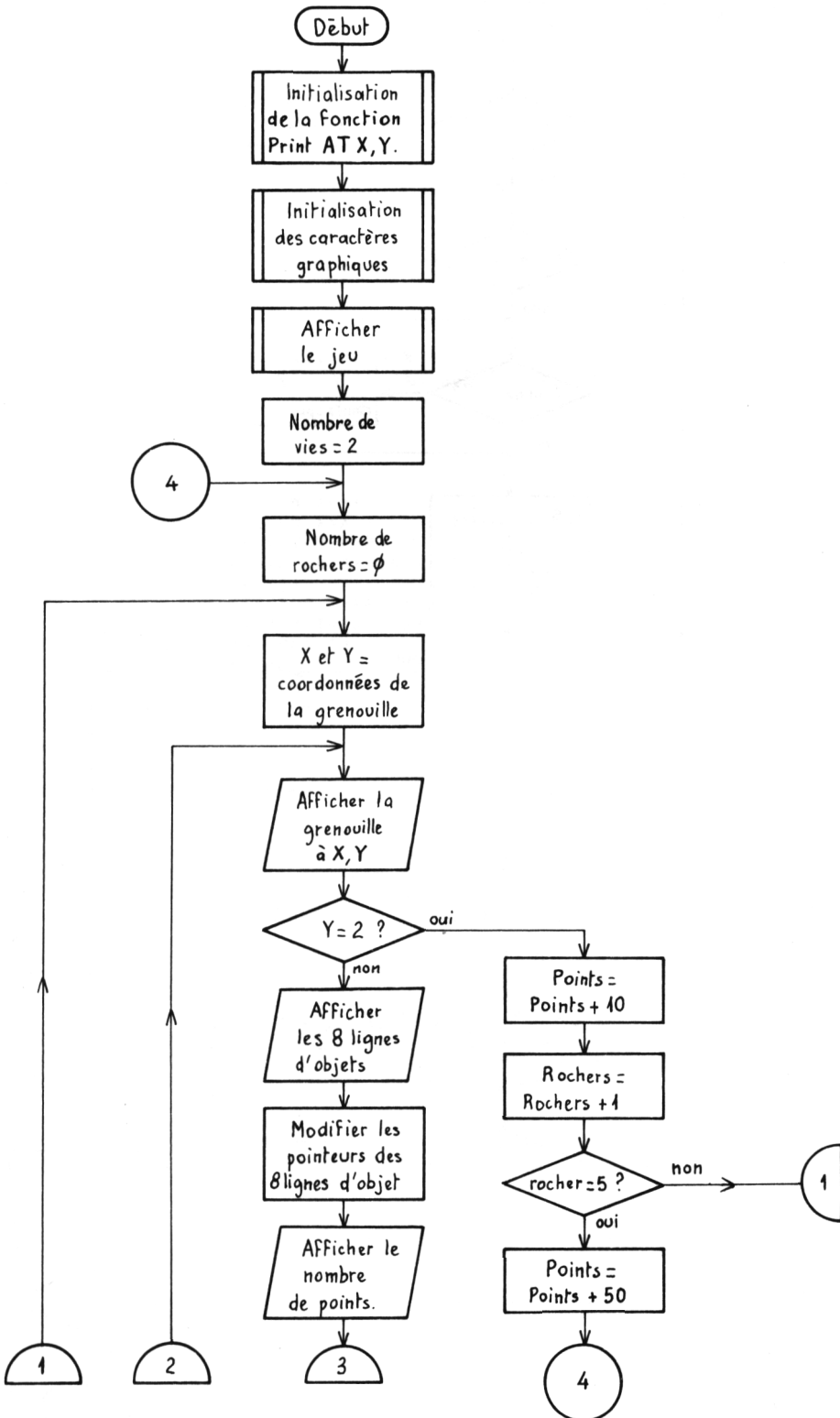
Son rêve ne cessera que lorsqu'elle aura regagné successivement les cinq rochers.

Alors, à vos bonds, prêt, partez !...

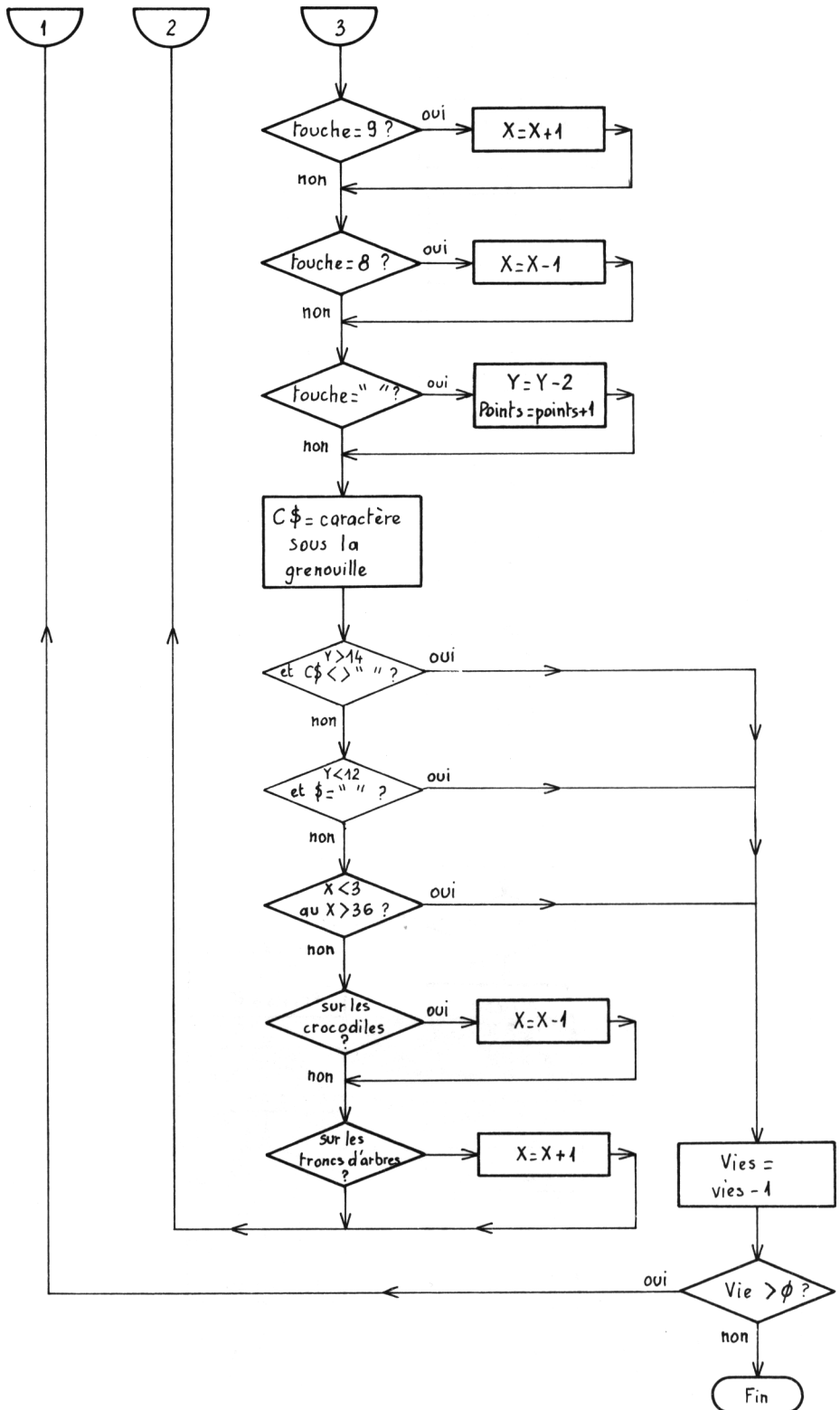
A la fin de chaque partie, relevez votre score (en haut à gauche de l'écran) et enfoncez n'importe quelle touche pour un autre jeu.



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE



```

10 REM GRENOUILLE
20 REM -----
30 GOSUB 1000
40 V=2: PLOT 24,0,CHR$(1)+"VIES: 2"
50 T=0
90 X=18:Y=24:C$=" "
100 PLOT X,Y,107:AX=X:AY=Y
110 A=2:FOR I=1 TO 8
120 !1,I*2+A,MID$(L$(I),D(I),36);
130 IF I=4 THEN A=6
140 NEXT I
150 IF Y=2 THEN 2500
500 FOR I=1 TO 7 STEP 2
510 D(I)=D(I)+1:IF D(I)=25 THEN D(I)=1
520 D(I+1)=D(I+1)-1:IF D(I+1)=0 THEN D(I+1)=24
530 NEXT I
540 PLOT 5,0,"POINTS:"+STR$(P)
580 REM -----
590 REM DEPLACEMENTS
600 A$=KEY$
610 IF A$=CHR$(9) THEN X=X+1
620 IF A$=CHR$(8) THEN X=X-1
630 IF A$=" " THEN Y=Y-2:P=P+1
640 IF AY=14 OR AY=12 OR AY=24 THEN PLOT AX,AY," "
650 C$=CHR$(SCRN(X,Y))
660 IF Y>14 AND C$<>" " THEN 2000
670 IF Y<12 AND C$=" " THEN 2000
680 IF X<3 OR X>36 THEN 2000
690 IF AY<>Y THEN 100
700 IF Y=4 OR Y=8 THEN X=X-1
710 IF Y=6 OR Y=10 THEN X=X+1
720 GOTO 100
980 REM -----
990 REM INITIALISATION
1000 FOR I=#400 TO #419
1010 READ A:POKE I,A
1020 NEXT I:DOKE#2F5,#400
1030 DATA #20,#96,#D9,#AC,#F8,#02,#C8,#8C,#69,#2,#A5,#1F
1040 DATA #A4,#20,#85,#12,#84,#13,#A9,#3B,#20,#0B,#CF,#4C,#61,#CB
1050 FOR I=0 TO 95+16+40
1060 READ A:POKE 46856+I,A
1070 NEXT I
1080 DATA 15,18,34,63,32,63,36,24,48,8,4,63,1,63,9,6
1090 DATA 3,4,8,63,32,63,18,12,56,36,34,63,1,63,9,6
1100 DATA 62,38,38,62,63,63,36,24,31,17,17,31,31,63,9,6
1110 DATA 3,7,9,9,9,9,7,3,0,32,48,48,48,48,32,0
1120 DATA 48,56,36,36,36,36,56,48,0,1,3,3,3,3,1,0
1130 DATA 12,45,63,12,12,30,18,51,63,63,63,63,63,63,63
1140 DATA 63,63,63,63,63,63,18,12,63,63,63,63,63,63,0,0
1150 DATA 0,0,32,48,31,13,7,63,0,0,0,0,63,63,63,63,0,0,0,0,32,48,62,63
1160 DATA 0,0,1,3,62,44,56,63,0,0,0,0,1,3,31,63
1200 L$(1)="qllllh      oppppq      qllllh      oppppq      gllllh
"
1210 L$(2)="jlli      jlli      jlli      jlli      jlli
"
1220 L$(3)=L$(1)
1230 L$(4)=L$(2)
1240 L$(5)="emnm cd      emnm cd      emnm cd      emnm cd      emnm
cd
"
1250 L$(6)="mmf - ab      mmf      ab      mmf      ab      mmf      ab      mmf
ab
"

```

L'ORIC A L'AFFICHE



```

1260 L$(7)=L$(5)
1270 L$(8)=L$(6)
1300 FOR I=1 TO 7 STEP 4
1310 D(I)=1:D(I+1)=30
1320 D(I+2)=5:D(I+3)=25
1330 NEXT I
1380 REM -----
1390 REM COULEURS
1400 PAPER 0:INK 7
1410 POKE 618,10:CLS
1420 FOR I=1 TO 25
1430 READ A:READ B:PLOT 0,I,A+16:PLOT 1,I,B
1440 NEXT I
1450 DATA 7,0,7,0,4,3,4,3,4,3,4,2,4,2,4,3,4,3,4,2,4,2,1,3
1460 DATA 1,3,1,3,0,3,0,3,0,3,0,4,0,4,0,3,0,3,0,4,0,4,2,1,2,1
1470 PLOT 2,2,"      11      11      11      11      11"
1480 RETURN
1980 REM -----
1990 REM PERDU
2000 EXPLODE
2010 V=V-1
2020 IF V<0 THEN ZAP:GET A$:RUN
2030 PLOT 25,0,"VIES:"+STR$(V)
2040 GOTO 90
2480 REM -----
2490 REM GAGNE
2500 PING
2510 T=T+1:P=P+10
2520 IF T=5 THEN GOSUB 1470:P=P+50:GOTO 50
2530 PLOT X-1,Y,"      "
2540 GOTO 90

```



L'ORIC A L'AFFICHE

LE PROGRAMME

Lignes 30 à 90 : initialisation

Cette première partie du programme appelle le sous-programme en 1000 qui exécute les tâches suivantes :

- 1- Définition du programme en Assembleur PRINT AT X,Y. Il s'agit du programme décrit à la page 141 du manuel de l'ORIC (lignes 1000 à 1040).
- 2- Définition des caractères graphiques contenus dans les datas (lignes 1050 à 1160).
- 3- Définition du tableau L\$ contenant les objets à déplacer sur l'écran et les pointeurs correspondants (tableau D, lignes 1200 à 1330).
- 4- Placer les codes des couleurs pour le papier et l'encre dans chaque ligne de l'écran (lignes 1400 à 1480).

Les variables X et Y donnent la position de départ de la grenouille et C\$ le caractère sur lequel elle est posée.

Lignes 100 à 540 : le jeu

Après l'affichage de la grenouille et la sauvegarde de ses coordonnées dans AX et AY, on place sur l'écran les chaînes de caractères du tableau L\$ à partir des pointeurs D() en utilisant la fonction PRINT AT.

Si la grenouille est déjà sur la ligne 2 de l'écran (ligne 150), on va en 2500 pour le comptage des rochers.

Les lignes 500 à 530 modifient le tableau des pointeurs D pour le déplacement des objets.

Lignes 600 à 720 : déplacement

En fonction de la touche qui vient d'être appuyée, on modifie les variables X et Y. On place dans C\$ le caractère qui correspond à ces nouvelles coordonnées.

Si C\$ est différent d'un espace et que la grenouille se trouve sur la route (ligne 660), on va en 2000 pour la phase "PERDU". Idem quand C\$ est égal à un blanc et que la grenouille est sur la rivière (ligne 670).

Si la grenouille dépasse les limites de l'écran (en ligne 680), on va aussi en 2000.

La grenouille devant suivre le cours de l'eau, on la déplace à droite ou à gauche suivant qu'elle se trouve sur un tronc d'arbre ou un crocodile (lignes 700 et 710).

Lignes 2000 à 2540 : perdu et gagné

Si l'on a perdu une vie, on décrémente le compteur V, et l'on revient au jeu uniquement si V est encore supérieur à 0.

Si l'on a gagné, on incrémente le compteur de rochers T de 1 et le compteur de points de 10. Si T est inférieur à 5 on continue le jeu, sinon on recommence une partie en incrémentant le compteur P de 50 points.

L'ORIC A L'AFFICHE

6/1980

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

INCENDIE

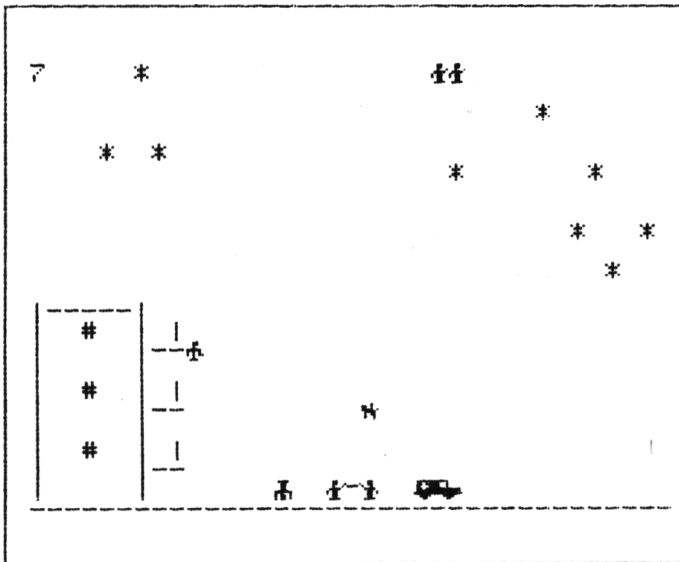
Après le lancement du programme, un décor se dessine lentement sur l'écran.

Dans le ciel des étoiles, et au milieu de l'écran, une maison, une ambulance et deux pompiers. Ces derniers doivent rattraper tous les personnages (1, puis 2, puis 3 jusqu'à 10) qui tombent de la maison en feu et les mettre dans l'ambulance.

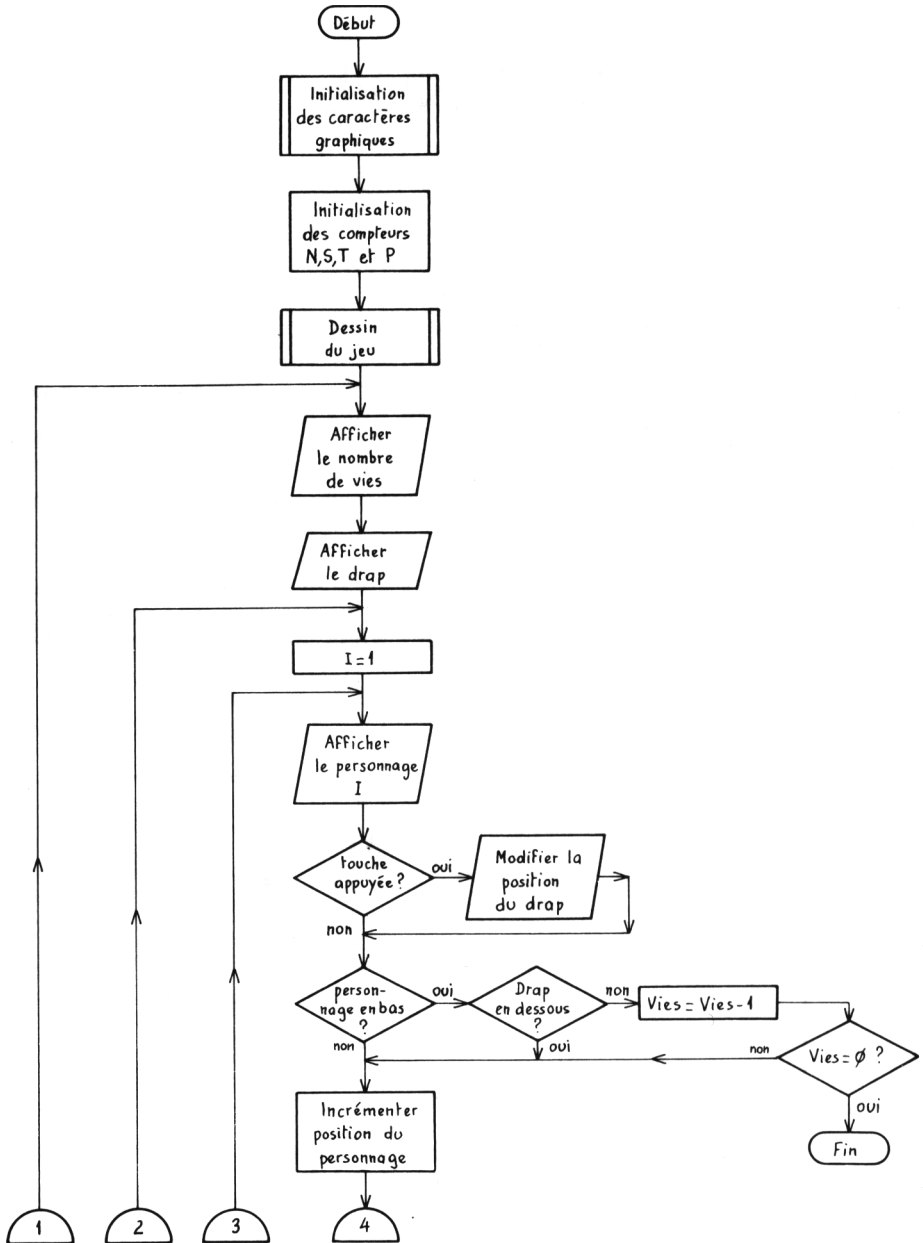
Vous les manœuvrez avec les touches Q (vers la gauche) et P (vers la droite).

Vous gagnez un point quand l'un des personnages atteint l'ambulance. Par contre vous perdez une des trois vies qui vous sont données au départ quand l'un des bonhommes s'écrase au sol. De plus, au bout de 50 points, vous gagnez une nouvelle vie.

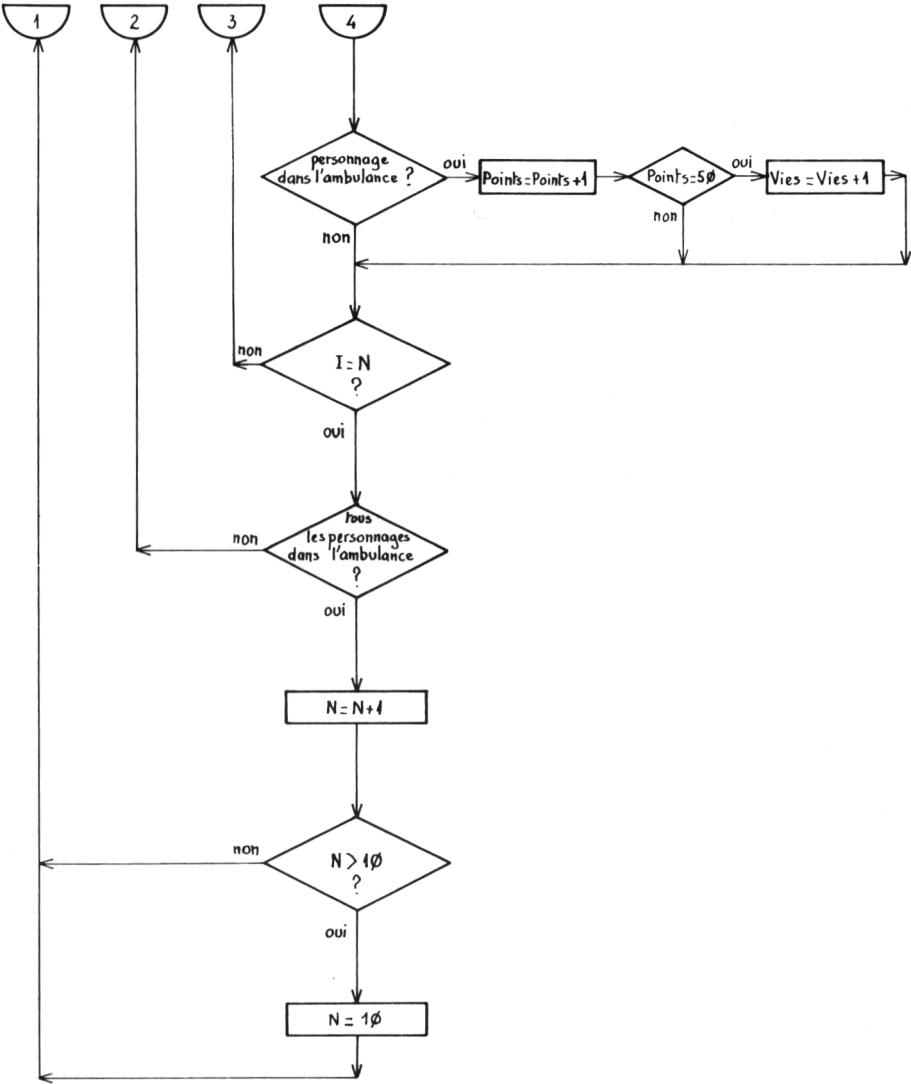
De quoi peut-être faire naître une vocation !



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE





```

10 REM INCENDIE
20 REM -----
30 DIM X(14),Y(14),C(14)
40 GOSUB 1000
50 N=1:P=1:S=2:T=0
100 CLS:PAPER 0:INK 1
110 GOSUB 3000
120 IF KEY$="" THEN 120
170 REM -----
180 REM LE JEU
190 GOSUB 2000
200 PLOT 10,21,A$(P)
210 FOR I=1 TO N
220 P(I)=0
230 NEXT I
240 I=0
250 REPEAT:I=I+1
260 IF I=1 THEN 300
270 IF P(I-1)=3 THEN PLOT 8,13+INT(RND(1)+.5)*3,"a"
280 C$=KEY$:P=P+(C$="P")*(P<3)-(C$="Q")*(P>1)
290 IF P(I-1)<6 THEN 500
300 IF P(I)>14 THEN 500
310 PLOT 10,21,A$(P)
320 IF P(I)=0 THEN PLOT 8,13," ":PLOT 8,16," ":GOTO 340
330 PLOT X(P(I)),Y(P(I))," "
340 P(I)=P(I)+1
350 IF P(I)<>15 THEN 380
360 T=T+1:IF T/50=INT(T/50) THEN S=S+1:GOSUB 2000
370 PLOT 0,0,STR$(T):GOTO 500
380 PLOT X(P(I)),Y(P(I)),CHR$(C(P(I))+96)
390 FOR J=1 TO 12-N
400 C$=KEY$:IF J<6 THEN P=P-(C$="Q")*(P>1)+(C$="P")*(P<3):PLOT 10,2
1,A$(P)
410 NEXT J
420 IF Y(P(I))=20 THEN PING:PLAY 0,0,0,0:IF SCRN(X(P(I)),21)=32 THE
N 700
500 UNTIL I=N
505 J=0
510 REPEAT:J=J+1
520 IF P(J)<15 THEN PULL:GOTO 240
530 UNTIL J=N
550 N=N+1:IF N>10 THEN N=10
560 GOTO 190
680 REM -----
690 REM PERDU
700 PLOT X(P(I)),20," "
710 PLOT X(P(I)),21,"d"
720 SHOOT:S=S-1
730 WAIT 25:GOSUB 2000
740 P(I)=16
750 IF S>-1 THEN 500
760 IF KEY$="" THEN 760
770 PULL:GOTO 50
980 REM -----
990 REM GRAPHIQUES
1000 DATA 12,12,8,62,42,43,8,28,0,4,52,61,30,22,18,2
1010 DATA 4,5,55,62,11,9,8,0,30,12,12,12,63,41,45,45
1020 DATA 12,12,9,30,44,44,12,30,0,33,30,0,0,0,0,12,12,36,30,13,1
3,12,30
1025 DATA 31,27,17,27,31,63,15,6,63,57,57,63,63,63,3,1,0,0,0,60,60,
62,48,32
1030 FOR I=0 TO 79

```





```

1040 READ A:POKE 46856+I,A
1050 NEXT I
1060 DATA 10,14,1,10,17,2,11,20,3,12,17,2,13,14,1,14,17,2,15,20,3
1070 DATA 16,17,2,17,14,1,18,17,2,19,20,3,20,17,2,21,14,3,22,17,2
1080 FOR I=1 TO 14
1090 READ X(I),Y(I),C(I)
1100 NEXT I
1110 A$(1)="efg      "
1120 A$(2)="      efg  "
1130 A$(3)="      efg"
1140 RETURN
1980 REM -----
1990 REM AFFICHE LES e
2000 PLOT 24,0,"      "
2010 IF S<=0 THEN 2050
2020 FOR J=1 TO S
2030 PLOT 23+J,0,"e"
2040 NEXT J
2050 RETURN
2980 REM -----
2990 REM DESSIN DU JEU
3000 POKE 616,12:PRINT
3020 PRINT "|-----|"
3030 PRINT "|  #  | |"
3040 PRINT "|      |--"
3050 PRINT "|      |"
3060 PRINT "|  #  | |"
3070 PRINT "|      |--"
3080 PRINT "|      |"
3090 PRINT "|  #  | |"
3100 PRINT "|      |--"
3110 PRINT "|      |             hij"
3120 PRINT "-----"
3130 FOR I=1 TO 10
3140 PLOT RND(1)*35+2,RND(1)*12,CHR$(2)+"*"
3150 NEXT I
3500 RETURN

```



L'ORIC A L'AFFICHE

LE PROGRAMME

Lignes 30 à 120 : initialisation

Pour connaître la position des personnages et les caractères graphiques qui s'y rapportent, on dimensionne trois tableaux X, Y et C. Ceux-ci sont remplis par le sous-programme 1000. C'est aussi ce sous-programme qui définit les caractères graphiques.

Après l'effacement de l'écran, on affiche le jeu par le sous-programme en 3000.

Lignes 190 à 560 : le jeu

Le sous-programme en 200 affiche des personnages en haut de l'écran pour indiquer au joueur le nombre de vies restantes.

Chaque personnage a une position bien déterminée contenue dans le tableau P. Celui-ci est donc remis à zéro aux lignes 210 à 230.

Comme on sort des boucles par des gotos lors du jeu, on utilise des fonctions REPEAT/UNTIL plutôt que des fonctions FOR/NEXT.

La variable N donnant le nombre de personnages à récupérer, on effectue cette boucle de 1 à N le nombre de fois qu'il faut pour qu'ils soient dans l'ambulance.

Les personnages apparaissent au balcon uniquement si le précédent a déjà couvert trois positions (ligne 270), et ils sautent uniquement quand le précédent en est à six positions (ligne 290).

Le caractère saisi par KEY\$ en ligne 280 permet de déterminer la position du drap (variable P). On utilise pour cela une expression logique pour modifier la valeur de P (ligne 280). Pour permettre de bouger plus rapidement le drap, on répète cette ligne en 400.

Quand un personnage atteint la position 15 (l'ambulance), on incrémente le compteur de points T (ligne 360). Si T est divisible par 50, on ajoute une vie (compteur S).

Dans le cas où un personnage arrive sur le sol, on teste directement sur l'écran par la fonction SCRN(X,Y) la présence du drap (ligne 420). Si le caractère trouvé à cet endroit est 32 (un blanc), on va à la ligne 700 pour la fin du jeu.

Les lignes 505 à 530 testent si tous les personnages sont dans l'ambulance. Si ce n'est pas le cas, on sort du REPEAT/UNTIL par une instruction PULL.

Tous les personnages étant arrivés à bon port, on incrémente le compteur N pour agrandir la famille au prochain tour (ligne 550).

Lignes 3000 à 3500 : dessin du jeu

Le POKE de la ligne 3000 indique au Basic que le prochain PRINT doit être affiché à la ligne 12 de l'écran.

L'ORIC A L'AFFICHE

6/1980

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

AUTOROUTE

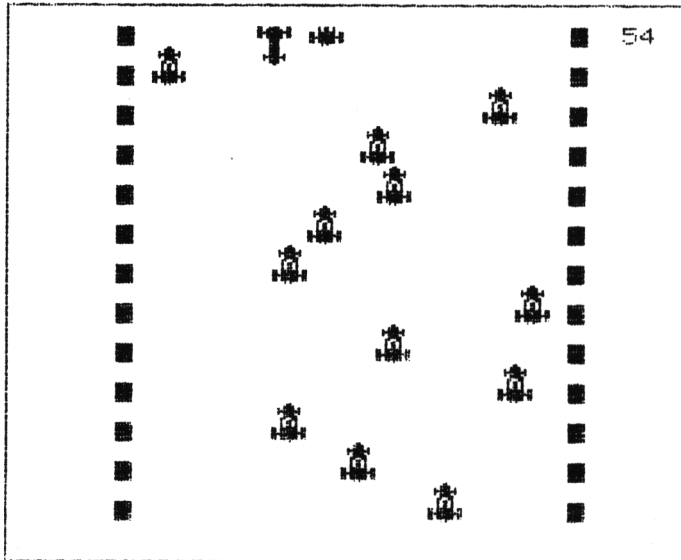
Aidez ce fou du volant qui a pris l'autoroute sur la voie inverse à faire le plus de kilomètres possibles sans accidents.

Pour lui faire éviter tout ce qui surgit en face de lui, pilotez sa voiture à l'aide des touches Q et P (directions gauche-droite).

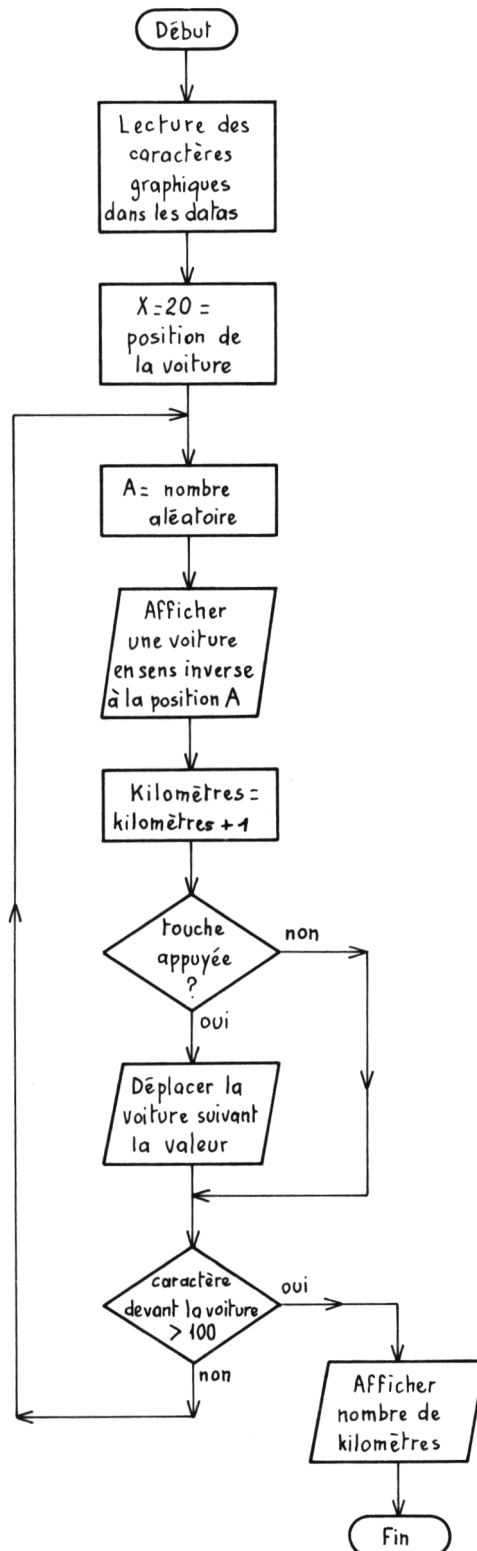
Si par malheur un accident se produisait quand même, la voiture effectuera une belle sortie de route.

Le programme donne alors le nombre de kilomètres parcourus sans accident.

En tapant sur une touche du clavier, vous retrouverez le conducteur imprudent qui a plus que jamais besoin de vous.



L'ORIC A L'AFFICHE



```

10 REM AUTOROUTE
20 REM -----
30 PAPER 0:INK 2:CLS
40 DATA 51,55,63,55,55,3,3,3,51,59,63,59,59,48,48,48
50 DATA 3,3,3,11,15,11,3,1,48,48,48,52,60,52,48,32
60 DATA 1,3,11,15,11,3,6,5,32,48,52,60,52,56,24,40
70 DATA 5,5,55,55,63,55,55,2,40,40,59,59,63,59,59,32
80 DATA 63,63,63,63,63,63,63,63
100 FOR I=0 TO 71
110 READ A:POKE 46856+I,A
120 NEXT I
130 PRINT CHR$(17);CHR$(6)
150 X=20:T=0
160 MUSIC 1,0,1,8:PLAY 7,0,4,100
180 REM -----
190 REM LE JEU
200 A=INT(RND(1)*22)+13
210 POKE 616,24:POKE 617,1:PRINT:PRINT
220 PRINT TAB(A+6);CHR$(27);"D";"ef"
230 PRINT " ";CHR$(27);"Bi";:PRINT TAB(A);CHR$(27);"Dgh";
240 PRINT TAB(47-A);CHR$(27);"Bi":PRINT
250 AS=KEY$
260 X=X+(AS="Q")-(AS="P")
270 T=T+1
300 PLOT 34,0,STR$(T)
310 PLOT X,0,CHR$(1)+"ab"
320 PLOT X,1,CHR$(1)+"cd"
330 IF SCRN(X+1,2)>100 OR SCRN(X+2,2)>100 THEN 500
400 GOTO 200
480 REM -----
490 REM FIN ...
500 EXPLODE:WAIT 20
505 X=X+1
510 FOR I=0 TO 5
520 PLOT X-I,I,32:PLOT X-I-1,I+1,"a"
530 PLOT X+I+1,I,32:PLOT X+I+2,I+1,"b"
540 PLOT X-I,I+1,32:PLOT X-I-1,I+2,"c"
550 PLOT X+I+1,I+1,32:PLOT X+I+2,I+2,"d"
560 SHOOT:WAIT 15
570 NEXT I
580 CLS
590 PLOT 10,11,CHR$(10)+CHR$(7)+"SCORE: "+STR$(T)
600 PLOT 10,12,CHR$(10)+CHR$(5)+"SCORE: "+STR$(T)
610 IF KEY$="" THEN 610
620 CLS
630 GOTO 150

```

L'ORIC A L'AFFICHE

LE PROGRAMME

Ligne 30 à 160 : initialisation

On place à partir du caractère "a" la représentation graphique des voitures ainsi que celle du bord de la route.

La variable X donne la position de départ de la voiture du joueur et T le nombre de kilomètres parcourus.

Lignes 200 à 400 : le jeu

Le nombre aléatoire contenu par la variable A donne l'emplacement de la voiture en sens inverse. Celle-ci est affichée simplement par une fonction PRINT ce qui a pour effet de faire monter l'écran de deux lignes vers le haut. Les deux instructions POKE de la ligne 210 donnent au Basic l'emplacement sur l'écran de la prochaine instruction PRINT (simulation d'un PRINT AT X,Y).

La ligne 250 teste le sens de déplacement du joueur. Si un déplacement est effectué, on modifie la valeur de la variable X avant d'afficher à l'écran le dessin de la voiture (lignes 310 et 320).

Si des caractères supérieurs à 100 (caractères des voitures) sont en face du joueur (testés par SCRN(,)), on va directement à la fin du jeu (ligne 500).

Lignes 500 à 630 : fin de partie

Des caractères supérieurs à 100 se trouvant devant la voiture du joueur, on simule un accident en l'affichant dans tous les sens. Puis l'on signale le nombre de kilomètres parcourus avant de revenir à une autre partie.

L'ORIC A L'AFFICHE

6/1980

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

SLALOM

Quoi de plus propice qu'une bonne piste de ski pour exercer ses talents d'agilité et d'endurance ?

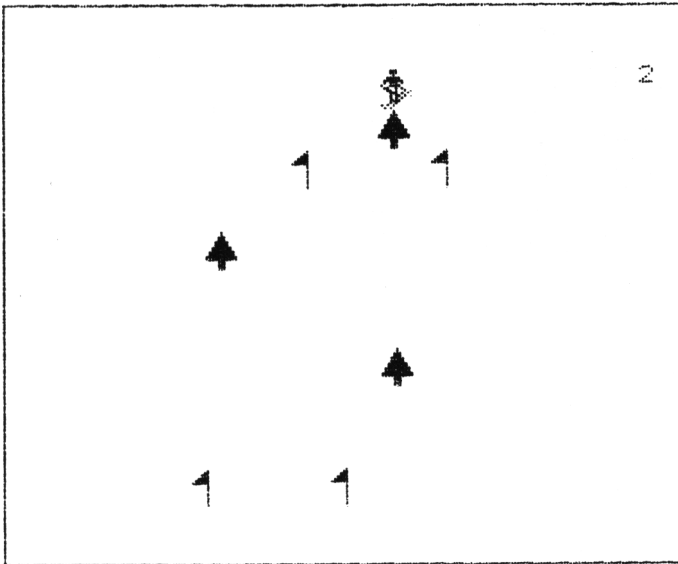
En lançant le programme, vous vous retrouverez directement parmi les sapins pour le slalom le plus long du monde.

Le skieur se manoeuvre à l'aide des touches de déplacement vers la gauche et vers la droite. Le nombre de points gagnés s'inscrit selon les conditions suivantes :

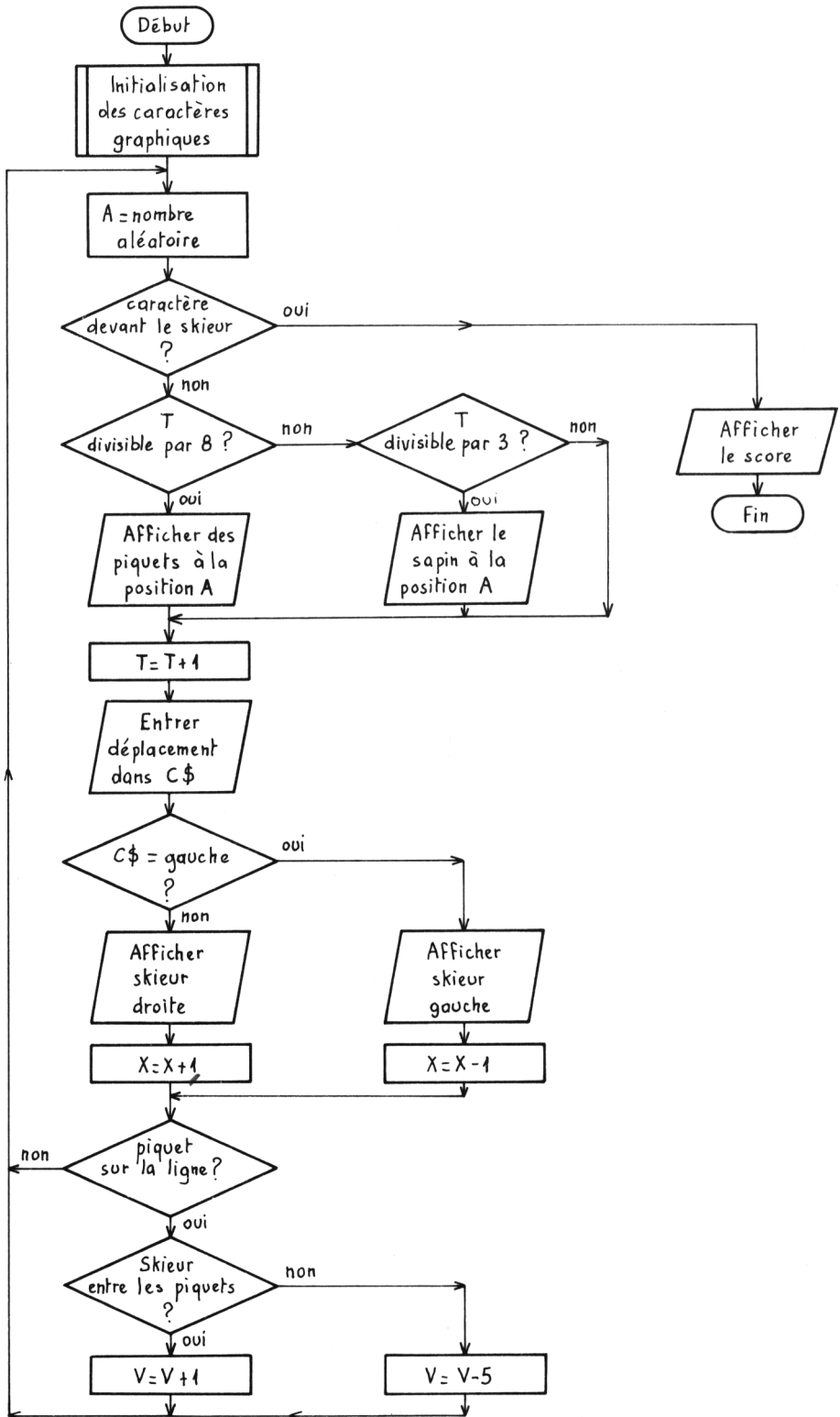
- une porte passée vaut 1 point ;
- une porte ratée fait perdre 5 points.

Le score final s'affiche sur la piste quand le skieur heurte un obstacle (sapins ou piquets).

Alors, prêts pour une descente ?



L'ORIC A L'AFFICHE





```

10 REM SLALOM
20 REM -----
30 GOSUB 1000
40 E$=CHR$(27):B$=E$+"W":V$=E$+"B":R$=E$+"A":N$=E$+"@":FB$=E$+"G"
50 CLS:INK 0:PAPER 7
60 X=20:T=0:C=8:V=0:S=3
70 DOKE 616,282:PRINT
80 REM -----
90 REM LE JEU
100 A=INT(RND(1)*15)+17
110 IF SCRN(X,2)>97 OR SCRN(X+1,2)>97 THEN 300
120 IF T/8<>INT(T/8) THEN 160
130 PRINT TAB(A);R$;" m m";N$
140 PRINT B$;FB$;CHR$(A+40);N$;TAB(A);R$;" n n";N$
150 GOTO 200
160 IF T/3<>INT(T/3) THEN 190
170 PRINT TAB(A);V$;"1j";N$:PRINT TAB(A);V$;"k1";N$
180 GOTO 200
190 PRINT:PRINT
200 T=T+1:PLOT 34,0,STR$(V)
210 C$=KEY$:IF C$<>" " THEN C=ASC(C$)
220 IF C=8 THEN PLOT X,0,"ab":PLOT X,1,"cd":IFX>6 THEN X=X-1
230 IF C=9 THEN PLOT X,0,"ef":PLOT X,1,"gh":IFX<30 THEN X=X+1
250 P=SCRN(1,1):IF P=32 THEN 270
260 IF X>P-51 AND X<P-43 THEN V=V+1 ELSE V=V-5
270 GOTO 100
280 REM -----
290 REM PERDU ...
300 SHOOT
310 PLOT 20,7,CHR$(10)+CHR$(4)+"SCORE:"+STR$(V)
320 PLOT 20,8,CHR$(10)+" SCORE:"+STR$(V)
330 GET AS
970 GOTO 50
980 REM -----
990 REM GRAPHIQUES
1000 DATA 3,3,1,7,15,11,19,12,32,32,0,48,56,40,48,40
1010 DATA 6,3,2,2,3,2,21,10,37,42,53,40,48,32,0,0
1020 DATA 1,1,0,3,7,5,3,5,48,48,32,56,60,52,50,52
1030 DATA 41,21,43,5,3,1,0,0,24,48,16,16,48,16,42,20
1040 DATA 1,1,3,3,7,7,15,15,0,0,32,32,48,48,56,56
1050 DATA 31,31,63,63,3,3,3,3,60,60,62,62,32,32,32,32
1060 DATA 1,3,7,15,31,63,1,1,1,1,1,1,1,1,1,1
1070 FOR I=0 TO 111
1080 READ A:POKE 46856+I,A
1090 NEXT I
1100 RETURN

```



LE PROGRAMME

Lignes 30 à 70 : initialisation

Après la définition des caractères graphiques, on place dans des chaînes de caractères les codes couleur de papier et d'encre nécessaires dans l'élaboration du jeu (voir liste des variables). Puis la ligne 60 initialise à 20 la position de départ du skieur sur l'écran (variable X).

Le POKE de la ligne 70 indique au Basic que la prochaine instruction PRINT doit commencer à la ligne 26. De ce fait, toutes les instructions PRINT qui vont suivre effectueront automatiquement un scroll sur l'écran puisque c'est la dernière ligne.

Lignes 100 à 270 : le jeu

Le nombre aléatoire contenu dans A donne la position sur l'écran des prochains piquets ou sapins (ligne 100).

Avant de les afficher, on teste par la fonction SCRN si l'écran est vide devant le skieur. Si ce n'est pas le cas, on va en 300 pour la fin du jeu.

Le compteur T indique au programme le nombre de lignes déjà passées. Si celui-ci est divisible par 8 on affiche à l'écran deux piquets à partir de la position donnée par A (lignes 120 à 150). Dans la première colonne de cette ligne, on affiche un caractère de code A+40 pour connaître par la suite la position des piquets. Ce caractère étant écrit en blanc sur fond blanc, il reste invisible sur l'écran.

Par contre, si le compteur T est divisible par 3, on effectue la même opération, mais cette fois-ci pour un sapin (lignes 160 à 180).

Suivant le code du caractère saisi dans la variable C\$, on affiche à la position X un skieur vers la droite ou vers la gauche et on incrémente ou l'on décrémente X de la même façon.

La ligne 250 teste le premier caractère de la ligne pour détecter la présence des piquets.

Si c'est le cas, on compare ce caractère avec la position du skieur.

En dehors des limites des piquets, on décrémente de 5 le compteur de points V, sinon on l'incrémente de 1.

L'ORIC A L'AFFICHE

DATE		NOM DU PROGRAMME : SLALOM					PAGE 1 / 1
IDENTIFICATION des VARIABLES							
NOM SYMBOLIQUE	TYPE (*)	ORIGINE				LONG.	OBSERVATIONS
		Clavier	Programme	Cassette	Disquette		
A, _ _ _	Variable	X					Nombre aléatoire : position des piquets du sapin
B, \$, _ _	Constante	X					Couleur : papier blanc
C, _ _ _	Variable	X					Sens du déplacement : 8 pour gauche, 9 pour droite
C, \$, _ _	Variable	X					Dernière touche enfoncée
E, \$, _ _	Constante	X					Code escape
F, B, \$, _	Constante	X					Couleur : encre blanche
I, _ _ _	Indice	X					Boucle FOR/NEXT
N, \$, _ _	Constante	X					Couleur : encre noire
P, _ _ _	Variable	X					Caractère sur le bord de l'écran : indique la présence des piquets
R, \$, _ _	Constante	X					Couleur : encre rouge
T, _ _ _	Compteur	X					Nombre de lignes déjà passées
V, _ _ _	Compteur	X					Nombre de points
V, \$, _ _	Constante	X					Couleur : encre verte
X, _ _ _	Variable	X					Position du skieur
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

CHENILLE

Vous devez constituer une chenille en avalant tous les éléments apparaissant sur l'écran.

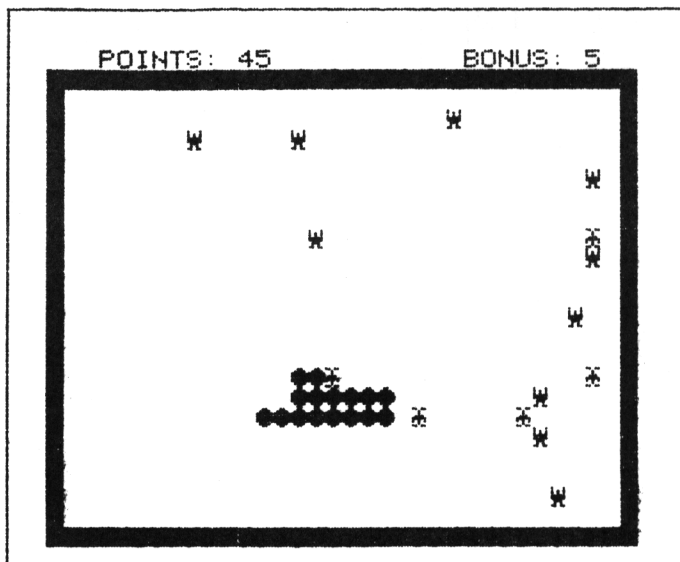
Ils sont de deux sortes : les bons qui donnent un bonus (ceux qui ont de grandes oreilles), et les autres qui en font perdre un (ceux qui ont les bras levés).

Le compte des bonus s'affiche en haut à droite de l'écran : il faut bien sûr avoir ce compteur supérieur à 0 pour prétendre manger un "anti-bonus".

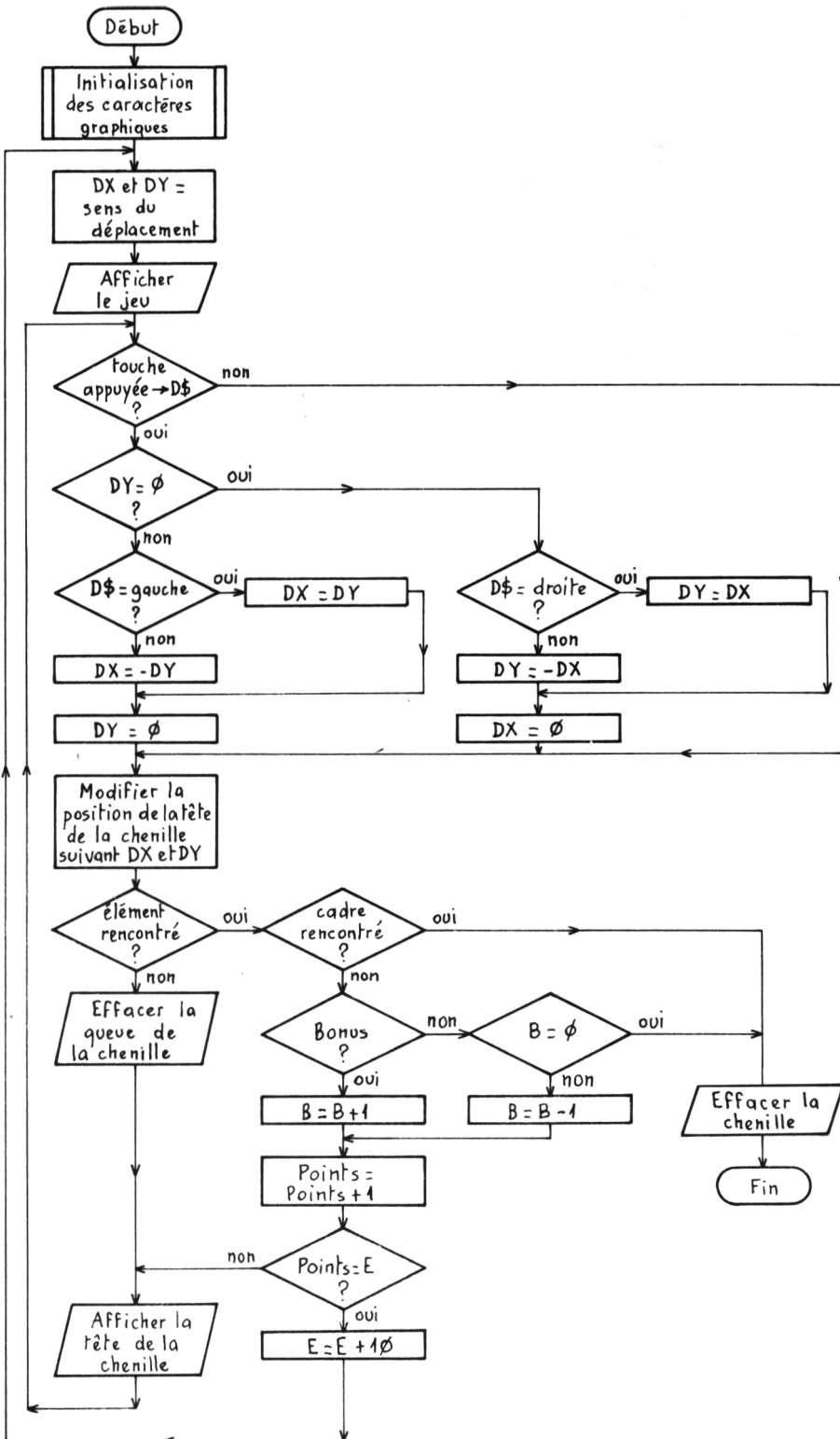
La chenille s'allonge à chaque élément qu'elle avale. Elle se déplace toute seule, le joueur devant modifier sa trajectoire avec les flèches gauche et droite du clavier.

Mais attention à ne pas heurter le bord du tableau ou à enrouler entièrement la chenille sur elle-même.

A chaque tableau réussi succède un nouveau tableau comportant un nombre de plus en plus important d'éléments. Ne vous découragez surtout pas au premier tableau : la dextérité s'acquiert très vite.



L'ORIC A L'AFFICHE



```

10 REM CHENILLE
20 REM -----
30 GOSUB 1000
40 DIM X(201),Y(201)
50 F=0: TT=0
80 REM -----
90 REM DESSIN DU JEU
100 P=1:Q=1:T=0:E=F+10:IF E>200 THEN E=200
110 DX=0:DY=-1:X(1)=17:Y(1)=12:B=0
120 INK 0:PAPER 3:CLS
130 FOR I=2 TO 35
140 PLOT 1,2,160:PLOT 1,25,160
150 IF I<26 THEN PLOT 2,I,160:PLOT 35,I,160
160 NEXT I
170 PLOT X(1),Y(1),97
180 FOR I=1 TO F
190 X=INT(RND(1)*30)+4:Y=INT(RND(1)*22)+2
200 IF SCRNX(X,Y)<>32 THEN 190
210 IF I>F/2 THEN PLOT X,Y,99 ELSE PLOT X,Y,100
220 NEXT I
230 PLOT 4,1,CHR$(1)+"POINTS: "
240 PLOT 25,1,CHR$(1)+"BONUS: "
250 PLOT 12,1,STR$(TT):PLOT 32,1,STR$(B)
260 GET D$:GOTO 310
280 REM -----
290 REM LE JEU
300 D$=KEY$:IF D$="" THEN 380
310 IF DY=0 THEN 350
320 IF D$=CHR$(8) THEN DX=DY:GOTO 340
330 DX=-DY
340 DY=0:GOTO 380
350 IF D$=CHR$(9) THEN DY=DX:GOTO 370
360 DY=-DX
370 DX=0
380 PLOT X(P),Y(P),98
390 AP=P:P=P+1:IF P=202 THEN P=1
400 X(P)=X(AP)+DX:Y(P)=Y(AP)+DY
410 IF SCRNX(X(P),Y(P))<>32 THEN 500
420 PLOT X(Q),Y(Q),32
430 Q=Q+1:IF Q=202 THEN Q=1
440 PLOT X(P),Y(P),97
450 GOTO 300
480 REM -----
490 REM ELEMENT RENCONRE ?
500 A=SCRNX(X(P),Y(P))
510 IF A=160 OR A=98 THEN 800
520 IF A=99 THEN R=R+1:GOTO 550
530 IF R=0 THEN 800
540 R=R-1
550 T=T+1:TT=TT+1
560 PLOT 12,1,STR$(TT):PLOT 32,1,STR$(B)+" "
570 IF T=F THEN ZAP:GOTO 100
580 PING:GOTO 440
780 REM -----
790 REM PERDU
800 PLOT 12,1,STR$(TT):PLOT 32,1,STR$(B):SHOOT
810 PLOT X(Q),Y(Q),32
820 Q=Q+1:IF Q=202 THEN Q=1
830 IF Q>P THEN 810
840 GET A$

```

L'ORIC A L'AFFICHE



```
850 GOTO 50
980 REM -----
990 REM GRAPHIQUES
1000 FOR I=0 TO 31
1010 READ A:POKE 46856+I,A
1020 NEXT I
1030 RETURN
1100 DATA 12,22,39,63,59,39,30,12,12,30,63,63,63,63,30,12
1110 DATA 54,8,8,28,62,42,8,54,34,42,42,62,62,28,20,54
```



LE PROGRAMME

Lignes 30 à 50 : initialisation

Après la définition des caractères graphiques par le sous-programme en 1000, on dimensionne deux tableaux X et Y pour connaître à tout moment la position des éléments qui composent la chenille.

Le compteur E donne le nombre d'éléments à éliminer, tandis que TT compte le nombre de points de toutes les parties.

Lignes 100 à 260 : dessin du jeu

Après l'effacement de l'écran et le dessin du cadre (lignes 120 à 160), on affiche aléatoirement sur l'écran les deux sortes d'éléments (code 99 et 100).

Lignes 300 à 450 : le jeu

Si une touche est appuyée, on modifie en conséquence les incréments de direction DX et DY (lignes 300 et 370). Puis l'on affiche à la place de la tête de la chenille, pointée par la variable P (code 97), un élément du corps (code 98).

Après l'incrément de P, on modifie le contenu de X(P) et Y(P) avec les incréments DX et DY (ligne 400).

Si un élément est rencontré (testé par SCRN en ligne 410), on va en 500 pour identification.

Dans le cas contraire, on efface le dernier élément de la chenille pointé par la variable Q. Cette dernière variable est incrémentée dans la limite des tableaux X et Y.

Pour terminer cette partie du programme, on affiche la tête de la chenille aux coordonnées pointées par P.

Lignes 500 à 580 : élément rencontré

Si l'élément rencontré est le bord ou la chenille (code 160 et 98), on va directement en ligne 800 pour la fin du jeu.

Par contre, s'il s'agit d'un bonus, on incrémente le compteur B.

Puisqu'il ne reste plus que les "anti-bonus" à tester, on va en ligne 800 pour la fin du jeu si le compteur B est à zéro.

Dans l'autre cas, on décrémente la variable B.

La variable T indique le nombre d'éléments déjà éliminés. On la compare donc à E pour connaître le moment de fin d'un tableau. Si ces deux variables sont égales, on va en ligne 100 où le compteur E est incrémenté de 10 pour le nouveau tableau (dans la limite de 200 éléments).

DATE		NOM DU PROGRAMME :					PAGE
		CHENILLE					1 / 1
IDENTIFICATION des VARIABLES							
NOM SYMBOLIQUE	TYPE (*)	ORIGINE				LONG.	OBSERVATIONS
		Clavier	Programme	Cassette	Disquette		
A ₁ _ _	Variable	X					Code du caractère rencontré par la chenille
A ₁ P ₁ _	Variable	X					Ancienne valeur de P
B ₁ _ _	Compteur	X					Nombre de bonus
D ₁ \$ ₁ _	Variable	X					Dernière touche enfoncée
D ₁ X ₁ _	Variable	X					} Sens du déplacement en X et Y de la chenille
D ₁ Y ₁ _	Variable	X					
E ₁ _ _	Compteur	X					Nombre d'éléments à éliminer
I ₁ _ _	Indice	X					Boucle FOR/NEXT
P ₁ _ _	Pointeur	X					Pointeur de la tête de la chenille dans les tableaux X et Y
Q ₁ _ _	Pointeur	X					Pointeur de la queue de la chenille dans les tableaux X et Y
T ₁ _ _	Compteur	X					Nombre de points du tableau
T ₁ T ₁ _	Compteur	X					Total des points de la partie
X ₁ _ _	Variable	X					} Nombres aléatoires : coordonnées des bonus et anti-bonus
Y ₁ _ _	Variable	X					
X ₁ (_)	Tableau	X					} Coordonnées de chaque élément de la chenille
Y ₁ (_)	Tableau	X					
_ _ _							
_ _ _							
_ _ _							
_ _ _							
_ _ _							
_ _ _							
_ _ _							

6/1980

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

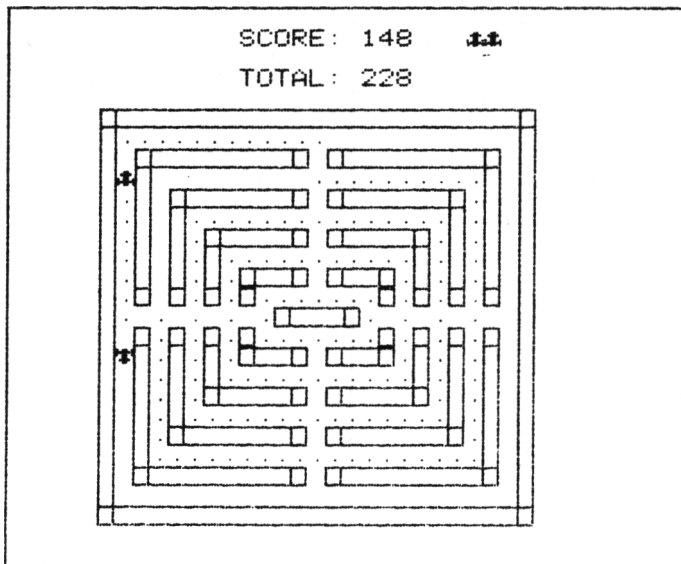
CIRCUIT 24

Evitez-vous la voiture infernale de l'ordinateur qui roule en sens inverse, tout en passant sur chaque portion du circuit?

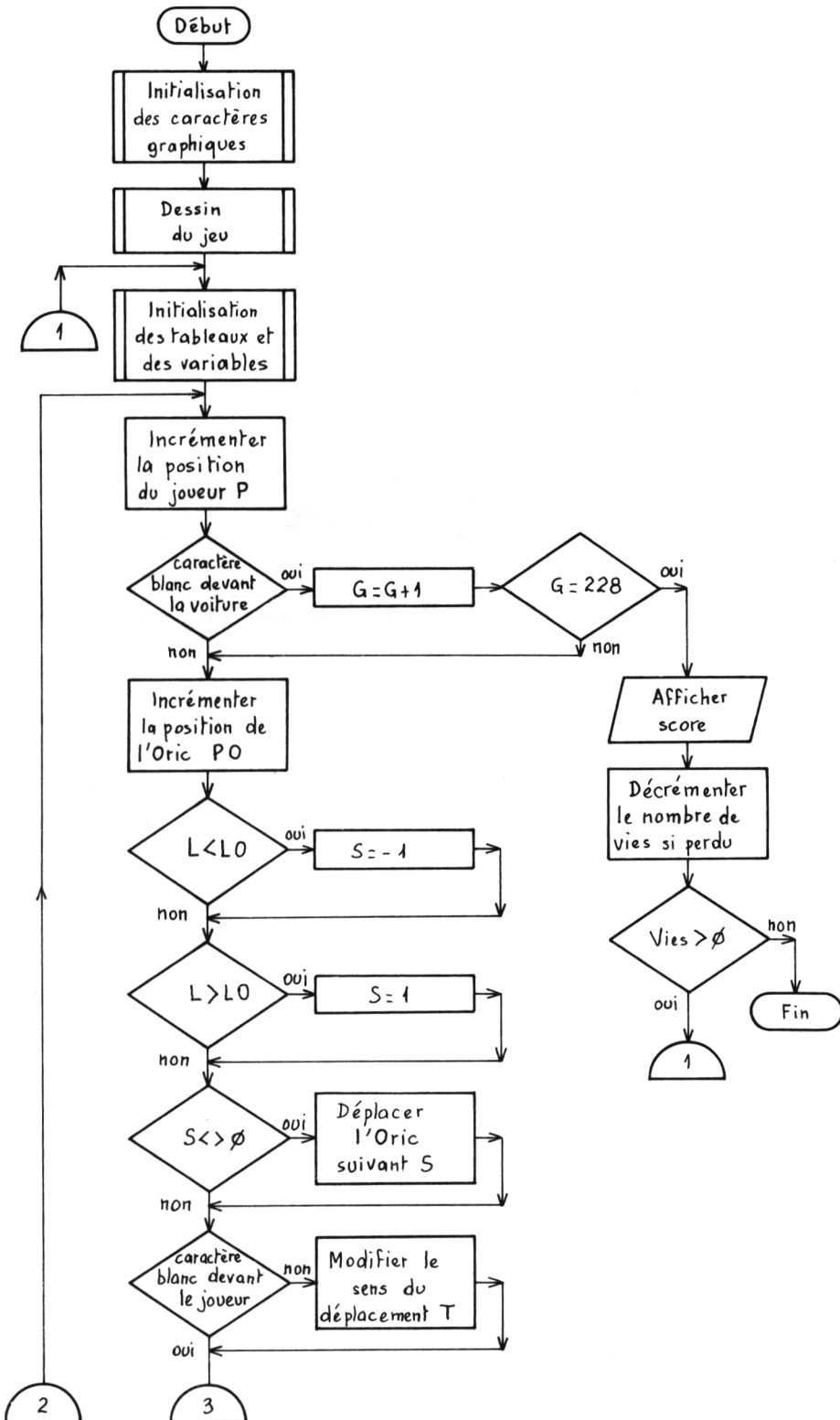
Deux commandes sont nécessaires à votre déplacement : la touche P pour vous diriger vers le centre et la touche Q pour aller vers l'extérieur.

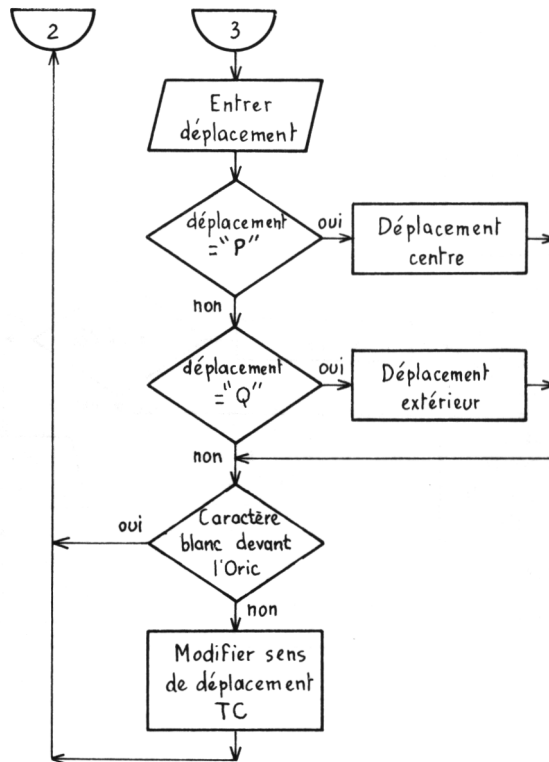
Dès le début de la course, trois parties gratuites vous sont allouées. Chaque déplacement donne un point. Les points sont affichés en haut de l'écran à chaque déplacement vers le centre ou vers l'extérieur. Si vous passez sur toutes les portions du circuit, une nouvelle piste s'affiche avec le total général et le jeu reprend de plus belle.

Bonne chance... et pas d'imprudence !



L'ORIC A L'AFFICHE







```

10 REM CIRCUIT 24
20 REM -----
30 GOSUB 2000
50 FOR I=1 TO PG:PLOT 25+I,0,"a"  ":NEXT I
90 A=A1
100 POKE P,46:P=P+T
110 IF PEEK(P)=32 THEN POKE P,46:G=G+1:IF G=228 THEN 3000
120 POKE P,V(C):POKE PO,A:PO=PO+TC
130 A1=PEEK(PO):POKE PO,V(CO)
140 IF L<LO THEN S=-1
150 IF L>LO THEN S=1
160 IFS<>0ANDPEEK(PO+T(CO+S))=32AND(PEEK(PO+T(CO))=32ORPEEK(PO+TC)=
46)THEN900
170 IF PEEK(P+T)=V(CO) THEN 3000
180 IF PEEK(P+T)=32 OR PEEK(P+T)=46 THEN 210
190 C=C+1:IF C=6 THEN C=2
200 T=T(C)
210 BS=KEY$:IF BS<>" " THEN AS=BS
220 IF AS="P" THEN 700
230 IF AS="Q" THEN 800
240 IF PEEK(PO+TC)=32 OR PEEK(PO+TC)=46 THEN 90
250 CO=CO-1:IF CO=1 THEN CO=5
260 TC=T(CO)
270 GOTO 100
680 REM -----
690 REM DEPLACEMENT CENTRE
700 IF PEEK(P+T(C+1))<>32 THEN 240
710 POKE P,46
720 P=P+T(C+1)+T(C+1)
730 L=L-1:IF L<1 THEN L=1
750 PLOT 19,0,STR$(G)
760 AS="":GOTO 240
780 REM -----
790 REM DEPLACEMENT EXTERIEUR
800 IF PEEK(P+T(C-1))<>32 THEN 240
810 POKE P,46
820 P=P+T(C-1)+T(C-1)
830 L=L+1
840 PLOT 19,0,STR$(G)
850 AS="":GOTO 240
880 REM -----
890 REM DEPLACEMENT ORIC
900 POKE PO,A1
910 PO=PO+T(CO+S)+T(CO+S)
920 LO=LO+S:S=0
930 A1=PEEK(PO)
940 GOTO 170
1000 STOP
1980 REM -----
1990 REM GRAPHIQUES
2000 DATA 0,12,30,12,45,63,45,0,0,7,18,63,63,18,7,0
2010 DATA 0,0,0,0,0,0,0,0,0,56,18,63,63,18,56,0
2020 DATA 0,45,63,45,12,30,12,0
2030 DATA 63,0,0,0,0,0,0,63,33,33,33,33,33,33,33
2040 DATA 63,33,33,33,33,33,33,63
2050 FOR I=0 TO 63
2060 READ A:POKE 46856+I,A
2070 NEXT I
2080 CLS:PAPER 1:INK 0
2110 FOR K=1 TO 12 STEP 2
2120 FOR I=4 TO 30-K*2
2140 PLOT I+K,K+3,"f"

```



L'ORIC A L'AFFICHE

LE PROGRAMME

Lignes 30 à 270 : le jeu

Après l'initialisation des graphiques et du dessin du jeu (sous-programme en 2000), on affiche en haut de l'écran le nombre de parties gratuites.

La ligne 100 place un point sur la position du joueur et l'incrémente (variable P).

Si la nouvelle position comporte un espace (ASCII 32), on incrémente le compteur de points G.

Si G = 228, on va en 3000 pour une nouvelle partie.

On répète la même séquence pour la voiture de l'ordinateur (variable PO).

Les variables L et LO donnant le numéro de piste (L=5 pour la piste extérieure et L=1 pour la piste intérieure), on place dans S le sens du déplacement de l'ordinateur (S=-1 pour l'intérieur et S=1 pour l'extérieur).

Dans le cas où S est différent de zéro et qu'une case blanche est adjacente, on va en 900 pour le déplacement de l'ORIC.

Si la voiture du joueur (PEEK(P)) rencontre celle de l'ordinateur (V(CO)) en ligne 170, on saute à la fin de partie (ligne 3000).

La touche appuyée (stockée dans B\$) est gardée en mémoire dans la variable A\$ jusqu'au prochain carrefour. Les lignes 220 et 230 dispatchent alors sur les parties concernées du programme.

Lignes 700 à 760 : déplacement centre

La touche P étant appuyée, on déplace la voiture du joueur vers le centre dans le cas où il existe une case blanche adjacente (ASCII 32).

Puis l'on incrémente la variable P et l'on décrémente le compteur L pour que l'ordinateur sache sur quelle piste on se trouve.

La ligne 750 affiche le nombre de points acquis.

Lignes 800 à 850 : déplacement extérieur

La touche Q étant appuyée, on déplace la voiture du joueur vers l'extérieur dans le cas où il existe une case blanche adjacente (ASCII 32).

Puis l'on incrémente la variable P et le compteur L pour que l'ordinateur sache sur quelle piste on se trouve.

La ligne 840 affiche le nombre de points acquis.

Lignes 900 à 940 : déplacement de l'ORIC

La variable S (sens du déplacement de la voiture du joueur) étant différente de zéro, on déplace la voiture du joueur vers l'extérieur (si S=1) ou vers l'intérieur (si S=-1).

Puis l'on incrémente la variable PO et le compteur LO pour que l'ordinateur sache sur quelle piste il se trouve.

L'ORIC A L'AFFICHE

Lignes 2000 à 2320 : graphiques et dessin du jeu

Après la lecture des caractères graphiques provenant des datas, on dessine sur l'écran le circuit 24.

Lignes 2400 à 2500 : initialisation

Les variables P et PO donnent la position exacte de la voiture du joueur et celle de l'ordinateur dans la mémoire de l'écran.

Le tableau T donne le nombre de cases-mémoire de déplacement pour les mouvements des voitures et le tableau V les caractères graphiques qui s'y rapportent.

Lignes 3000 à 3160 : fin de partie

Si la voiture du joueur a rencontré celle de l'ordinateur ou si toutes les portions du circuit ont été couvertes, on affiche le score total et l'on efface les points pour une nouvelle partie.

Lignes 3200 à 3230 : fin du jeu

Comme il ne reste plus aucune partie gratuite (variable PG), on affiche un nouveau jeu dès qu'une touche est enfoncée.

L'ORIC A L'AFFICHE

DATE		NOM DU PROGRAMME : CIRCUIT 24				PAGE 1 / 1	
IDENTIFICATION des VARIABLES							
NOM SYMBOLIQUE	TYPE (*)	ORIGINE				LONG.	OBSERVATIONS
		Clavier	Programme	Cassette	Disquette		
A ₁ _ _ _	Variable	X					Contient le caractère qui effacera la voiture de l'ORIC au prochain déplacement
_ _ _							
A ₁ 1 _ _	Variable	X					Caractère sur l'écran avant déplacement de l'ORIC
A ₁ \$ _ _	Variable	X					Caractère correspondant au déplacement.
B ₁ \$ _ _	Variable	X					Dernière touche enfoncée
C ₁ _ _ _	Variable	X					Sens du déplacement du joueur
C ₁ 0 _ _	Variable	X					Sens du déplacement de l'ORIC
G ₁ _ _ _	Compteur	X					Nombre de points dans une partie
I ₁ _ _ _	Indice	X					Boucle FOR/NEXT
K ₁ _ _ _	Indice	X					Boucle FOR/NEXT
L ₁ _ _ _	Variable	X					} Numéro de la piste du joueur } L=1 : piste intérieure } L=5 : piste extérieure
_ _ _							
L ₁ 0 _ _	Variable	X					Numéro de la piste de l'ORIC
P ₁ _ _ _	Variable	X					Position du joueur
P ₁ G ₁ _	Compteur	X					Parties gratuites
P ₁ 0 _ _	Variable	X					Position de l'ORIC
S ₁ _ _ _	Variable	X					Différence entre le joueur et l'ORIC
T ₁ _ _ _	Variable	X					Valeur du déplacement du joueur
T ₁ 0 _ _	Variable	X					Valeur du déplacement de l'ORIC
T ₁ P ₁ _	Compteur	X					Total des points
T ₁ (_)	Tableau	X					Direction du déplacement
V ₁ (_)	Tableau	X					Tableau des pointeurs des caractères graphiques

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (C) Editions du P.S.I.

CANADAIR

Qui n'a jamais rêvé d'être seul maître à bord d'un avion ?
Voilà l'occasion d'exercer vos talents de pilote.

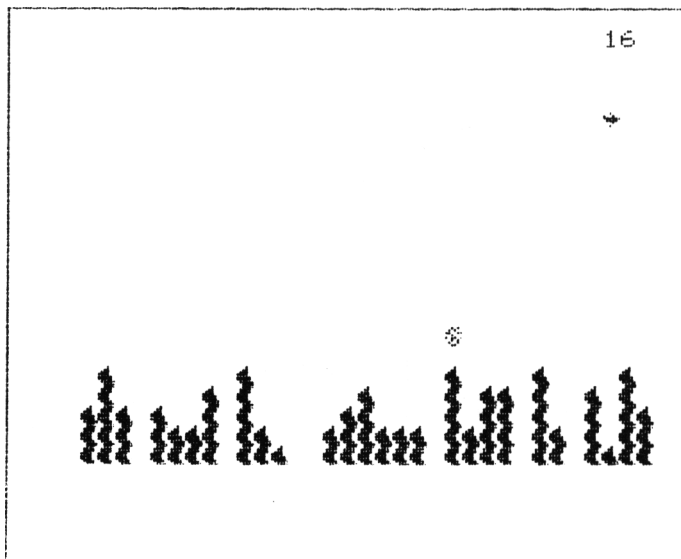
Un feu de forêt vient d'être annoncé et le canadair peut
jeter son eau sur toutes les flammes dessinées sur l'écran.

Au premier tableau, le canadair va et vient d'abord lentement,
puis sa vitesse est doublée définitivement.

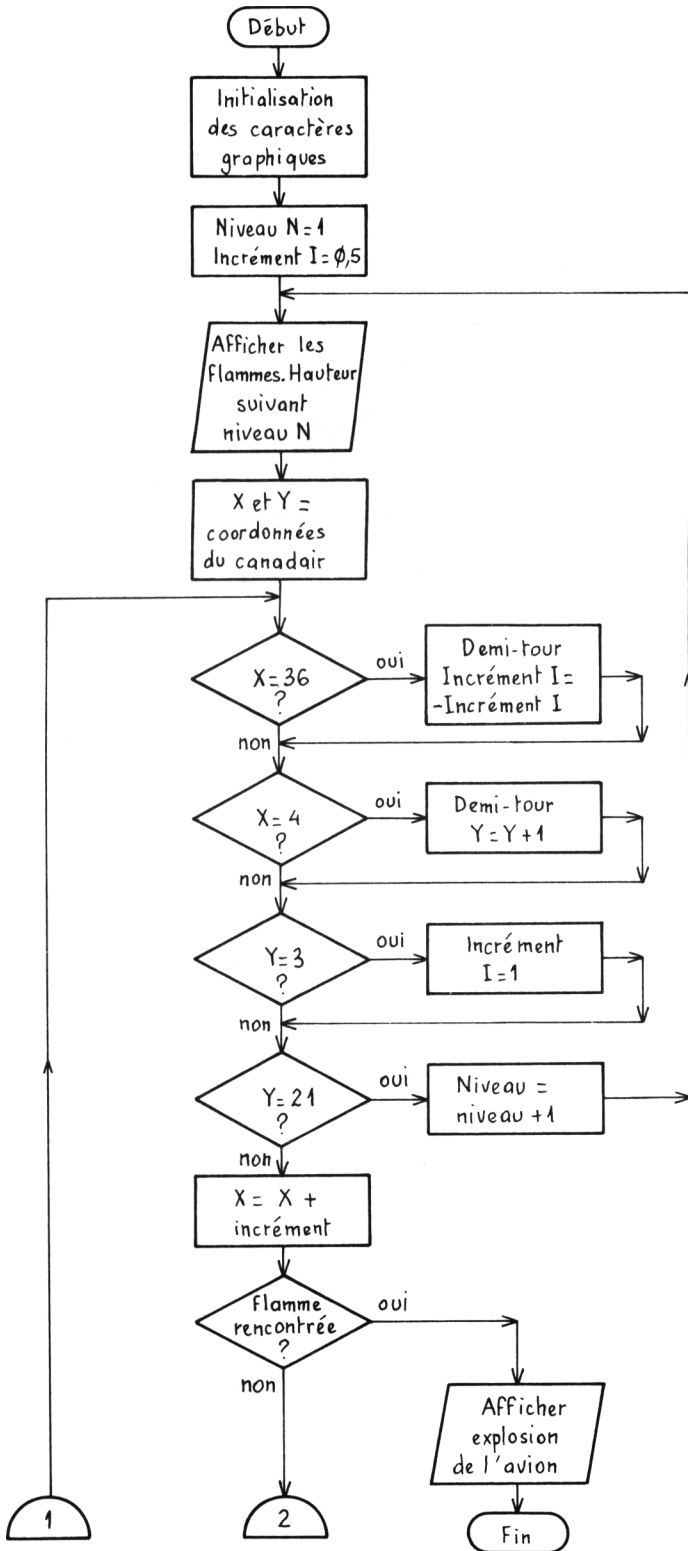
Le largage de l'eau se fait par la barre d'espace ainsi que
les changements de tableaux quand toutes les flammes sont
éteintes, et les remises en jeu.

Attention : aux tableaux suivants, les flammes montent de plus
en plus haut et nécessitent une bonne réflexion stratégique
pendant la descente.

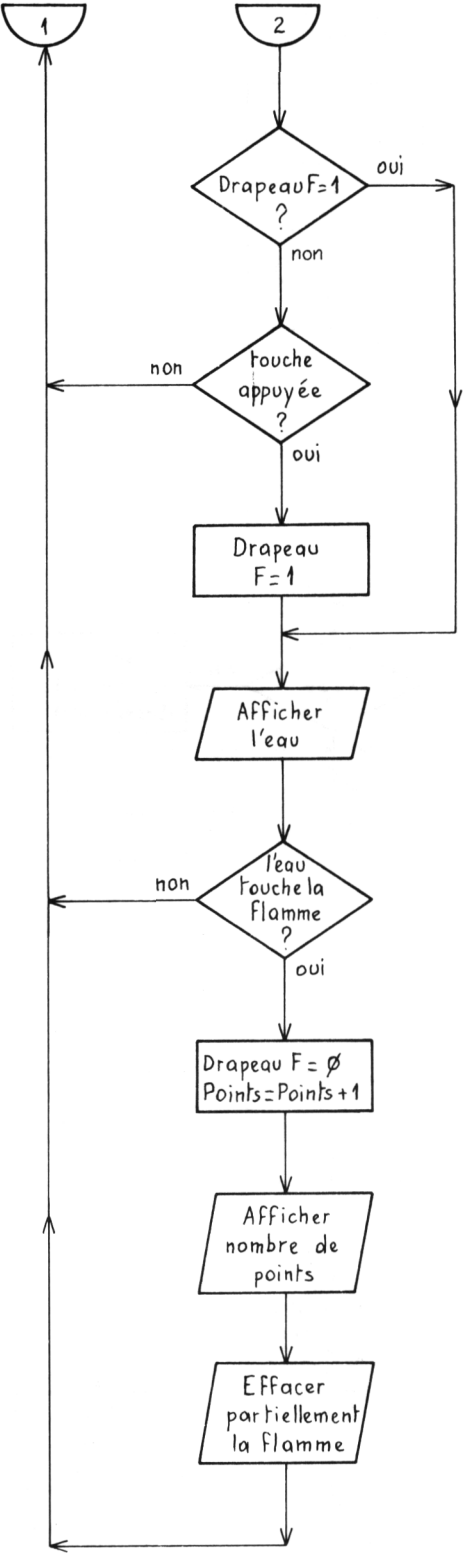
Quand l'avion heurte une flamme, il s'écrase et repart à
zéro.



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE





```

10 REM CANADAIR
20 REM -----
30 TEXT:CLS:PAPER 0
40 DATA 4,12,28,60,62,62,31,15,15,30,30,60,60,60,30,15,0,8,36,63,30
   ,4,8,0
50 DATA 0,4,9,63,30,8,4,0,10,20,51,8,45,10,18,12
60 FOR I=0 TO 39
70 READ C:POKE 46856+I,C
80 NEXT I
90 F=0:T=0:N=1
100 P=.5
110 I=P
120 CLS:INK 1
130 MUSIC 1,0,1,8:PLAY 0,0,0,0
180 REM -----
190 REM LE JEU
200 FOR X=4 TO 36
210 A=INT(RND(1)*N*4)
220 PLOT X,20-A,"a"
230 FOR Y=21-A TO 21
240 PLOT X,Y,"b"
250 NEXT Y
260 NEXT X
310 X=4
320 Y=0
340 IF X>=36 THEN I=-P
350 IF X<=4 THEN I=P:PLOT X,Y," ":Y=Y+1:PLOT 0,Y,3
360 IF Y=3 THEN P=1
370 IF Y=21 THEN 1200
380 X=X+I
390 IF SCRN(X,Y)=97 THEN 1000
400 IF I>0 THEN PLOT X-1,Y,"c":GOTO 420
410 PLOT X,Y,"d"
420 PLAY 1,1,1,100:PLAY 0,0,0,0
430 IF F=1 THEN 520
440 IF KEY$="" THEN 340
450 F=1
500 L=Y+1
510 C=X
520 PLOT C,L,32
530 L=L+1
540 IF L>=22 THEN F=0:GOTO 340
550 IF SCRN(C,L)=97 THEN 580
560 PLOT C,L,"e"
570 GOTO 340
580 F=0:T=T+1:PLOT 33,0,STR$(T)
590 PLAY 1,1,4,100:MUSIC 1,0,1,8
600 FOR J=L TO L+2
610 IF J<22 THEN PLOT C,J," "
620 NEXT J:L=L+2
630 IF L>=21 THEN 340
640 PLOT C,L+1,"a"
650 GOTO 340
980 REM -----
990 REM PERDU
1000 IF X>=38 THEN X=37
1010 FOR I=Y TO 20
1020 PLOT X-1,I-1," "
1030 PLOT X-1,I,"c\/":PLOT X,I+1,"/\ "
1040 ZAP:WAIT 5:EXPLODE:WAIT 5
1050 NEXT I
1060 IF KEY$="" THEN 1060
1070 GOTO 90
1200 SHOOT
1210 IF KEY$="" THEN 1210
1220 N=N+1:IF N=5 THEN N=4
1230 GOTO 110

```



L'ORIC A L'AFFICHE

LE PROGRAMME

Ligne 30 à 130 : initialisation

Après la définition des caractères graphiques, on initialise les divers compteurs (voir liste des variables).

Puis l'on donne un ordre **MUSIC** pour le bruit du moteur de l'avion.

Lignes 200 à 320 : dessin du jeu

La variable **N** donnant le niveau du jeu, on affiche à partir du bas de l'écran des flammes plus ou moins hautes suivant le nombre aléatoire à lui-même fonction de **N**.

Les variables **X** et **Y** définissent la position du canadair.

Lignes 340 à 650 : le jeu

Si le canadair arrive à droite de l'écran ($X \geq 36$), on change de direction en modifiant la valeur de l'incrément **I** (ligne 340).

On effectue l'opération inverse s'il arrive à gauche ($X \leq 4$ en ligne 350) et l'on descend d'une ligne.

Quand on arrive sur la troisième ligne ($Y=3$), on change l'incrément de base pour aller plus vite (de 0.5 à 1).

La 21ème ligne étant la dernière, on incrémente le niveau du jeu **N**, et l'on commence une nouvelle partie (ligne 370).

Une fois la position du canadair modifiée (ligne 380), on teste sur l'écran le caractère qui s'y trouve. Si celui-ci est différent d'un espace, on va en ligne 1000 pour la fin du jeu.

Suivant la valeur de **I** (positive ou négative), on affiche un canadair vers la gauche ou vers la droite (lignes 400 et 410).

Le drapeau **F** indique que l'eau est déjà en train de descendre. On va alors directement en ligne 520.

Dans l'autre cas, on teste pour savoir si l'on doit jeter de l'eau ou non.

Les lignes 500 à 570 affichent sur l'écran l'eau qui descend.

Dès que celle-ci touche la flamme (testée en ligne 550), on efface une partie (lignes 580 à 650).

Lignes 1000 à 1230 : perdu et gagné

Si le canadair a touché une flamme, on l'affiche de sa position jusqu'au bas de l'écran à grand renfort de **ZAP** et de **EXPLODE**.

Mais si c'est gagné, on incrémente le niveau du jeu **N** avant de reprendre une autre partie.

L'ORIC A L'AFFICHE

6/1980

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

RÉFLEXE

Avez-vous de bons réflexes ?

Le jeu suivant vous aidera à le savoir.

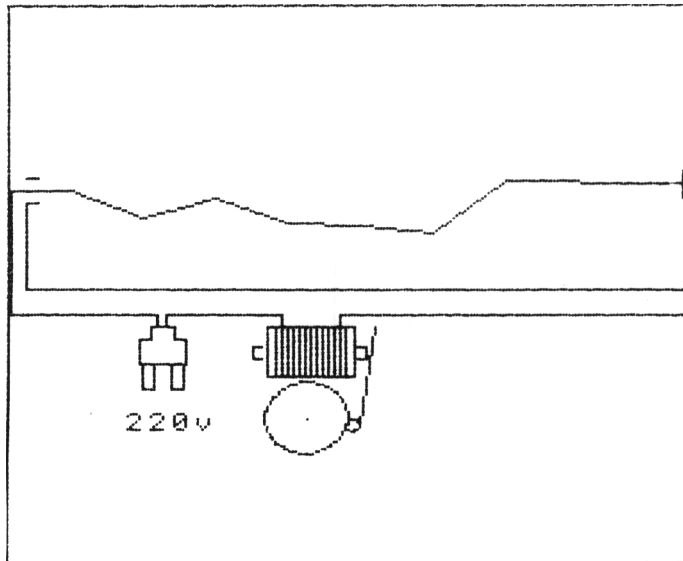
Il s'agit de guider un anneau le long d'un fil électrique qui dessine des courbes aléatoires plus ou moins compliquées.

Pour cela, vous disposez des flèches de déplacement vers le haut et vers le bas.

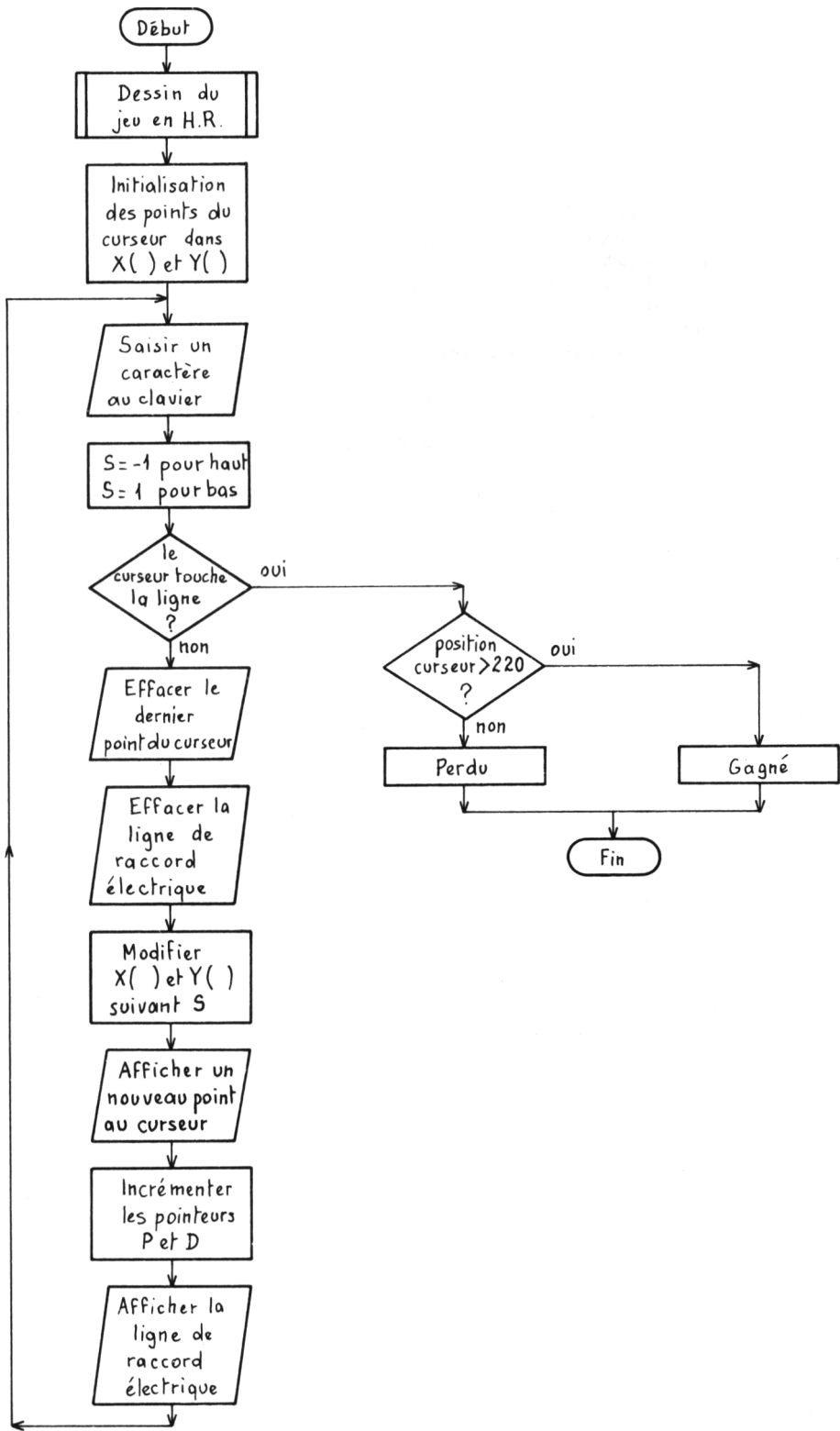
Faites partir l'anneau en tapant sur une des deux flèches et rectifiez sans arrêt son glissement alors qu'il a tendance à revenir à l'horizontale.

Une sonnerie intempestive se déclenche quand vous heurtez le fil et quand vous gagnez.

Ce programme est à utiliser de préférence les jours de fête pour tester la lucidité de tous les joyeux convives...



L'ORIC A L'AFFICHE





```

10 REM REFLEXES
20 REM -----
30 HIRES
40 RESTORE:GOSUB 1000
100 CURSET 0,90,1
110 DRAW 20,0,1
120 X=20:Y=90
130 S=1
140 X=X+25
150 IF RND(1)>0.5 THEN S=S*-1
160 IY=RND(1)*25*S
170 IF Y+IY>110 OR Y+IY<10 THEN 150
180 DRAW 25,IY,1
190 Y=Y+IY
200 IF X>149 THEN 300
210 GOTO 140
300 DRAW 50,90-Y,1
310 DRAW 17,0,1
320 CURMOV 0,-5,1:DRAW 0,11,1
380 REM -----
390 REM INIT. CURSEUR
400 X=5:Y=85:P=5:D=1
410 FOR I=1 TO 5
420 X(I)=X:Y(I)=Y
430 CURSET X(I),Y(I),1
440 CURSET X(I),Y(I)+10,1
450 X=X+1
460 NEXT I
470 GET A$:IF A$="" THEN 470
480 REM -----
490 REM DEPLACEMENT
500 A$=KEY$
510 S=1*(A$=CHR$(11))-1*(A$=CHR$(10))
520 IF POINT(X(P)+1,Y(P))=-1 THEN 700
530 IF POINT(X(P)+1,Y(P)+9)=-1 THEN 700
540 CURSET X(D),Y(D),0:CURSET X(D),Y(D)+10,0
550 CURSET X(D),130,0:DRAW 0,Y(D)-120,0
560 X(D)=X(P)+1:Y(D)=Y(P)+S
570 CURSET X(D),Y(D),1:CURSET X(D),Y(D)+10,1
580 P=P+1:IF P=6 THEN P=1
590 D=D+1:IF D=6 THEN D=1
600 CURSET X(D),130,1:DRAW 0,Y(D)-120,1
610 GOTO 500
680 REM -----
690 REM SONNETTE
700 IF X(P)>220 THEN 760
710 FOR I=1 TO 15
720 PING:WAIT 15
730 NEXT I
740 GET A$:IF A$="" THEN 730
750 GOTO 30
760 FOR I=1 TO 15
770 SHOOT:WAIT 15
780 NEXT I
790 GET A$:IF A$="" THEN 790
800 GOTO 30
980 REM -----
990 REM DESSIN DU JEU
1000 FOR I=1 TO 7
1010 READ N,X,Y
1020 CURSET X,Y,1
1030 FOR J=1 TO N

```





```

1040 READ X,Y: DRAW X,Y,1
1050 NEXT J,I
1060 CURSET 102,182,1: CIRCLE 15,1
1070 FOR I=91 TO 115 STEP 2
1080 CURSET I,145,1: DRAW 0,20,1
1090 NEXT I
1100 CURSET 125,145,1: DRAW -5,40,1
1110 CURMOV -2,0,0: CIRCLE 3,1
1120 CURSET 40,180,0: CHAR 50,0,1: CURSET 48,180,0: CHAR 50,0,1
1130 CURSET 56,180,0: CHAR 48,0,1: CURSET 64,180,0: CHAR 118,0,1
1140 RETURN
1180 REM -----
1190 REM GRAPHIQUES
1200 DATA 13,0,90,0,50,50,0,0,5,3,0,0,-5,40,0
1210 DATA 0,5,20,0,0,-5,120,0,0,-10,-228,0,0,-35
1220 DATA 8,48,145,0,5,-4,0,0,10,16,0,0,-10,-4,0,0,-5,-8,0
1230 DATA 3,45,160,0,10,4,0,0,-10
1240 DATA 3,55,160,0,10,4,0,0,-10
1250 DATA 4,88,145,0,20,30,0,0,-20,-30,0
1260 DATA 3,86,153,-3,0,0,5,3,0
1270 DATA 3,119,153,3,0,0,5,-3,0
    
```



LE PROGRAMME

Lignes 30 à 320 : initialisation

Après le passage en haute résolution (commande **HIRES**), on dessine sur l'écran le jeu à l'aide des datas en fin de programme (sous-programme en 1000).

Comme on utilise plusieurs fois ce sous-programme, on effectue d'abord une commande **RESTORE**.

Les lignes 100 à 320 dessinent la ligne brisée en utilisant des nombres aléatoires pour corser le jeu.

Lignes 400 à 470 : initialisation du curseur

Chaque point du curseur est placé dans les tableaux **X** et **Y** pour en connaître à tout moment sa position. La variable **P** pointe le premier de ces points et **D** le dernier.

Lignes 500 à 610 : déplacement

La touche appuyée étant placée dans **AS**, on détermine par une fonction logique la valeur de **S** (ligne 510).

Les lignes 520 et 530 testent l'écran à côté du curseur pour savoir si la ligne brisée est touchée.

Après l'effacement du dernier point du curseur (à **X(D),Y(D)**) et de la ligne de raccord électrique, on définit le dernier point comme étant maintenant le premier, et on modifie en conséquence les tableaux **X()** et **Y()**.

Il ne reste plus qu'à afficher le nouveau point, à incrémenter les pointeurs **P** et **D** et à afficher la ligne de raccord électrique.

Lignes 700 à 800 : sonnette

Si l'on a gagné (**X(P) = 220**), on envoie quinze **SHOOT**, sinon on envoie quinze **PING** pour la sonnette.

Lignes 1000 à 1270 : dessin du jeu

Les traits qui composent le dessin sont effectués en plusieurs temps.

Le premier **DATA** (variable **N**) donne le nombre de traits à tracer, les deux suivants le point de départ pour **CURSET** (variable **X** et **Y**), et les **N** fois 2 datas qui suivent les coordonnées des instructions **DRAW**.

L'ORIC A L'AFFICHE

[illegible]

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

CHAPITRE 2

JEUX DE RÉFLEXION

CHIEN DE BERGER

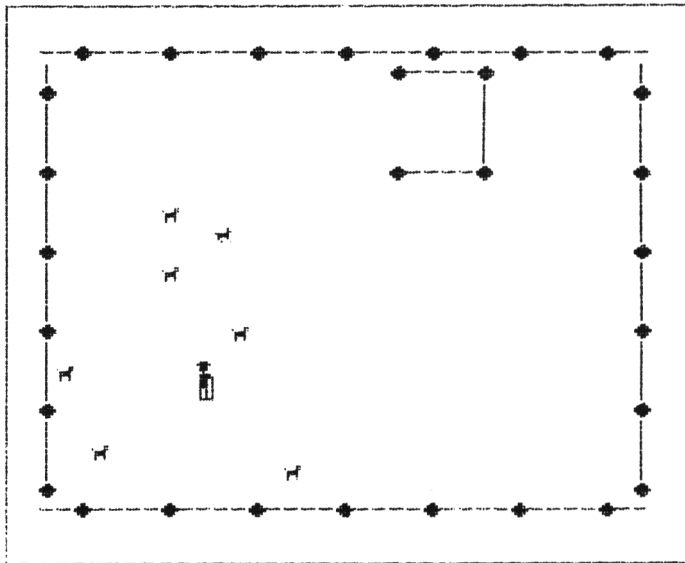
Ce jeu a pour inspiration les fameuses compétitions de chiens de berger en Ecosse.

Le gagnant est celui qui ramène le plus vite possible tout un troupeau de moutons dans son enclos.

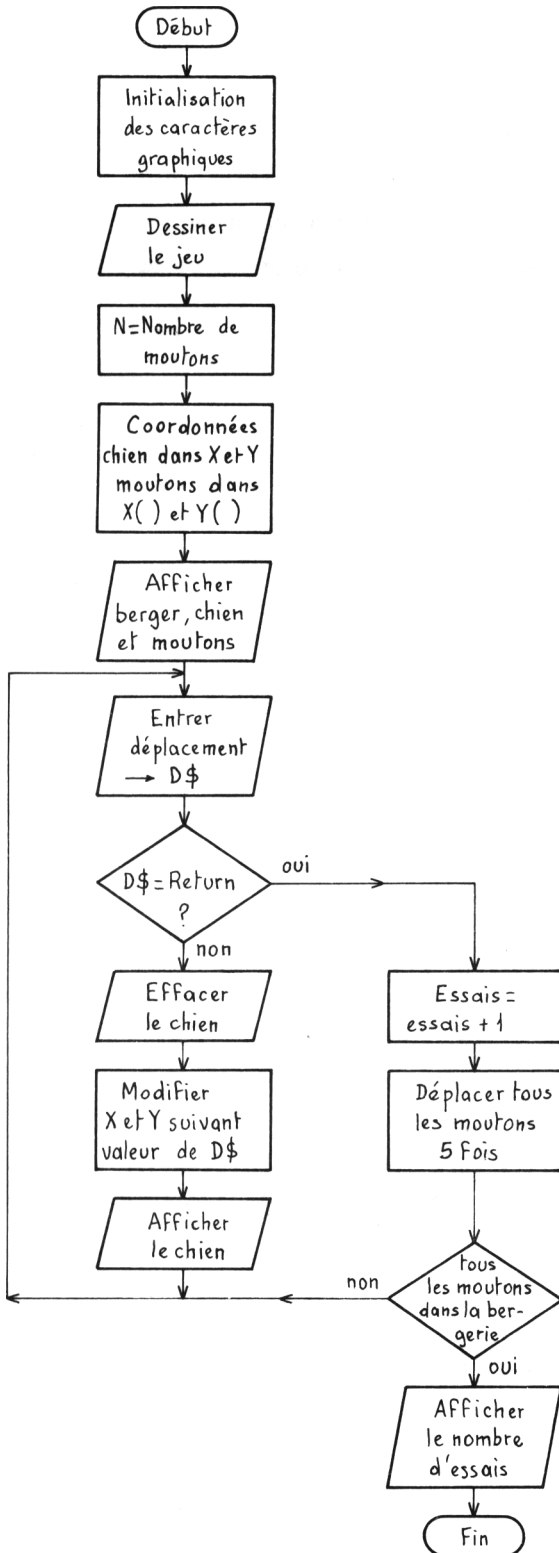
Ici, le chien de berger peut diriger de cinq à dix moutons, selon les parties. Vous devez le déplacer avec les quatre flèches et valider le mouvement en tapant sur RETURN.

A ce moment, tous les moutons bougent et seuls ceux qui sont vraiment près de l'enclos y pénètrent définitivement. Les autres s'éparpillent de plus en plus dans le pâturage et le pauvre chien n'a plus qu'à les rabattre à nouveau vers l'enclos.

Bien sûr, il s'agit d'utiliser le moins de coups possible. Le nombre d'essais s'affiche d'ailleurs dès que tous les moutons sont dans l'enclos.



L'ORIC A L'AFFICHE





```

10 REM CHIEN DE BERGER
20 REM -----
30 PAPER 0:INK 2:CLS
40 F=0:E=0
50 DATA 0,0,2,3,62,30,18,9,0,0,3,35,30,30,18,18,0,12,30,63,63,30,12
,0
60 DATA 28,62,28,28,8,30,31,29,29,29,29,21,21,21,21,31
70 FOR I=0 TO 39
80 READ A:POKE 46856+I,A
90 NEXT I
100 DIM X(11),Y(11)
180 REM -----
190 REM DESSIN DU JEU
200 FOR I=3 TO 37
210 IF I/5=INT(I/5)THENPLOTI,0,"c":PLOTI,23,"c"ELSEPLOTI,0,"-":PLOTI
,23,"-"
220 IF I>24 THEN 250
230 IF I/4=INT(I/4) THEN PLOT3,I-2,"c":PLOT37,I-2,"c":GOTO 250
240 PLOT 3,I-2,"I":PLOT 37,I-2,"I"
250 NEXT I
280 REM -----
290 REM AFFICHE BERGER, CHIEN, MOUTONS
300 X=INT(RND(1)*13)+4
310 Y=INT(RND(1)*20)+2
320 PLOT X,Y,"d":PLOT X,Y+1,"e"
330 V=INT(RND(1)*10)+18
340 W=INT(RND(1)*15)+1
360 PLOT V,W,"c----c"
370 FOR I=W+1 TO W+4:PLOT V+5,I,"I":NEXT I
380 PLOT V,W+5,"c----c"
390 W=W+1
400 N=5+INT(RND(1)*6)
410 FOR I=1 TO N+1
420 X(I)=INT(RND(1)*14)+4
430 Y(I)=INT(RND(1)*22)+1
440 IF SCRN(X(I),Y(I))<>32 THEN 420
450 IF I<=N THEN PLOT X(I),Y(I),"b":GOTO 470
460 X=X(I):Y=Y(I):PLOT X(I),Y(I),"a"
470 NEXT I
580 REM -----
590 REM DEPLACE CHIEN, MOUTONS
600 D$=KEY$:IF D$="" THEN 600
610 IF D$=CHR$(13) THEN E=E+1:GOTO 700
620 PLOT X,Y," "
630 IF D$=CHR$(8) THEN IF SCRN(X-1,Y)=32 THEN X=X-1
640 IF D$=CHR$(9) THEN IF SCRN(X+1,Y)=32 THEN X=X+1
650 IF D$=CHR$(11) THEN IF SCRN(X,Y-1)=32 THEN Y=Y-1
660 IF D$=CHR$(10) THEN IF SCRN(X,Y+1)=32 THEN Y=Y+1
670 PLOT X,Y,"a"
680 GOTO 600
700 FOR J=1 TO 5
710 F=0
720 FOR I=1 TO N
730 PLOT X(I),Y(I)," "
740 IF X(I)<V OR X(I)>V+4 THEN 800
750 IF Y(I)<W OR Y(I)>W+3 THEN 800
760 F=F+1
770 IF SCRN(X(I)+1,Y(I))<>32 THEN 910
780 X(I)=X(I)+1

```





```
790 GOTO 910
800 AX=X(I):AY=Y(I)
810 IF Y<Y(I) THEN Y(I)=Y(I)+1
820 IF Y>Y(I) THEN Y(I)=Y(I)-1
830 IF Y(I)=AY AND Y(I)=1 THEN Y(I)=Y(I)+1
840 IF Y(I)=AY AND Y(I)=22 THEN Y(I)=Y(I)-1
850 IF X(I)=AX AND X(I)=4 THEN X(I)=X(I)+1:GOTO 890
860 IF X(I)=AX AND X(I)=36 THEN X(I)=X(I)-1:GOTO 890
870 IF X<X(I) THEN X(I)=X(I)+1
880 IF X>X(I) THEN X(I)=X(I)-1
890 IF SCRN(X(I),Y(I))=32 THEN 910
900 X(I)=AX:Y(I)=AY
910 PLOT X(I),Y(I),"b"
920 NEXT I
930 IF F=N THEN 1000
940 NEXT J
950 PING
960 GOTO 600
980 REM -----
990 REM FIN
1000 PLOT 14,22,"NB D'ESSAIS:"+STR$(E)
1010 SHOOT
1020 IF KEYS="" THEN 1020
1030 RUN
```



LE PROGRAMME

Lignes 30 à 100 : initialisation

Après la définition des caractères graphiques des datas des lignes 50 et 60, on dimensionne deux tableaux **X** et **Y** pour connaître à tout moment la position des moutons.

Le compteur **F** donne le nombre de moutons déjà dans la bergerie, tandis que **E** compte le nombre d'essais à chaque partie.

Lignes 200 à 250 : dessin du jeu

Cette partie du programme affiche à l'écran la clôture du pâturage avec les caractères graphiques définis dans l'initialisation.

Lignes 300 à 470 : affiche berger, chien et moutons

Ces trois éléments sont affichés sur l'écran à des positions aléatoires. Les coordonnées du chien sont sauvegardées dans les variables **X** et **Y** et celles des moutons dans les tableaux **X()** et **Y()**. C'est la variable **N** qui détermine le nombre de moutons (entre 5 et 10).

La bergerie est aussi déplacée à chaque partie. Ses coordonnées sont conservées dans les variables **V** et **W**.

Lignes 600 à 680 : déplacement du chien

La touche appuyée étant stockée dans la variable **DS**, on modifie en conséquence les variables **X** et **Y** si aucun élément sur l'écran ne s'y oppose (testé par **SCRN**).

Si cette touche correspond au "RETURN" du clavier (code 13), on va en 700 pour le déplacement des moutons.

Lignes 700 à 960 : déplacement des moutons

Chaque mouton est déplacé à tour de rôle, cinq fois de suite, dans la direction qui l'éloigne le plus du chien.

S'il se trouve dans la bergerie (coordonnées **V** et **W** on incrémente le compteur **F**. Le mouton ne peut alors plus en sortir.

Si le compteur **F** est égal à **N**, on va directement à la ligne 1000 pour la fin du jeu.

Dans le cas contraire, on revient en ligne 600 pour un nouvel essai.

DATE		NOM DU PROGRAMME : CHIEN DE BERGER					PAGE 1 / 1	
IDENTIFICATION des VARIABLES								
NOM SYMBOLIQUE	TYPE (*)	ORIGINE				LONG.	OBSERVATIONS	
		Clavier	Programme	Cassette	Disquette			
A ₁ X ₁ _ _	Variable	X					} Anciennes coordonnées d'un mouton avant déplacement	
A ₁ Y ₁ _ _	Variable	X						
D ₁ \$ ₁ _ _	Variable	X					Dernière touche appuyée	
E ₁ _ _ _	Compteur	X					Nombre d'essais	
F ₁ _ _ _	Compteur	X					Nombre de moutons dans la bergerie	
I ₁ _ _ _	Indice	X					Boucle FOR/NEXT	
J ₁ _ _ _	Indice	X					Boucle FOR/NEXT	
N ₁ _ _ _	Compteur	X					Nombre de moutons	
V ₁ _ _ _	Variable	X					} Coordonnées de la bergerie	
W ₁ _ _ _	Variable	X						
X ₁ _ _ _	Variable	X					} Coordonnées du chien	
Y ₁ _ _ _	Variable	X						
X ₁ (_) ₁	Tableau	X					} Coordonnées des moutons	
Y ₁ (_) ₁	Tableau	X						
_ _ _ _								
_ _ _ _								
_ _ _ _								
_ _ _ _								
_ _ _ _								
_ _ _ _								
_ _ _ _								
_ _ _ _								
_ _ _ _								
_ _ _ _								

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

MISSIONNAIRES ET CANNIBALES

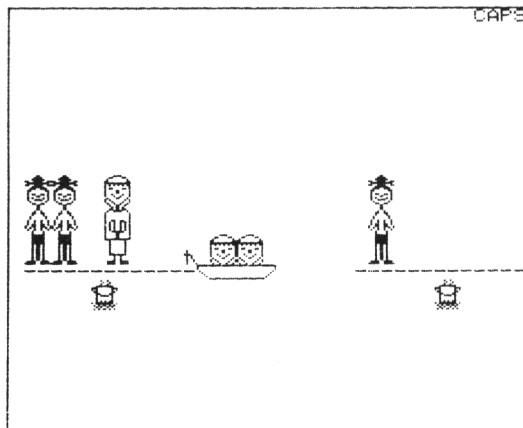
Ne vous fiez pas à la mine angélique des six personnages qui vont apparaître sur votre écran. En effet, les trois cannibales guettent le premier moment d'inattention des trois missionnaires pour les précipiter dans leurs marmites bouillantes.

Pourrez-vous les faire tous passer sur l'autre rive en sachant que les missionnaires ne doivent jamais être en minorité sur l'une des deux berges ?

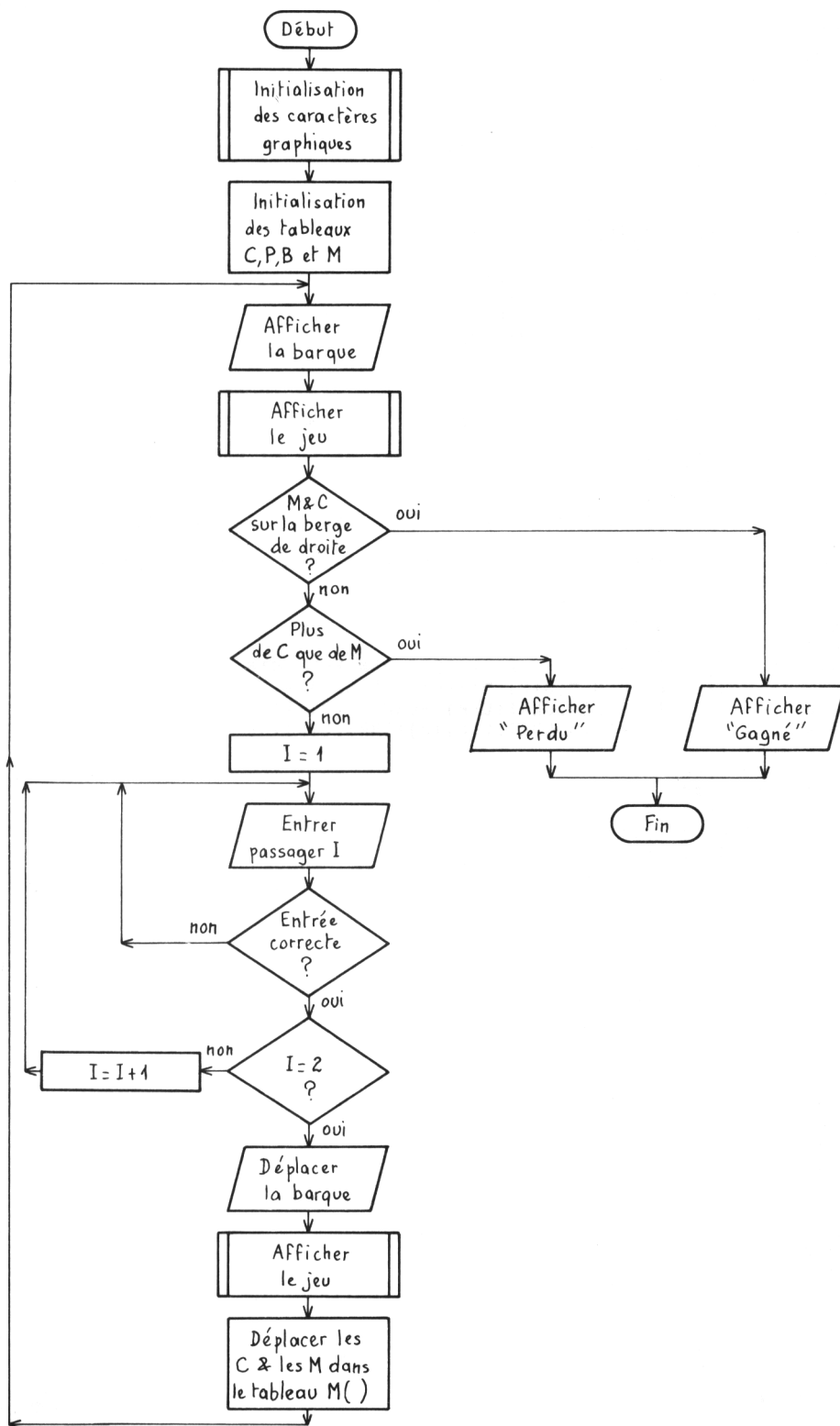
Vous avez à votre disposition une barque qui ne peut prendre plus de deux passagers par traversée. A chaque voyage, l'ordinateur vous demande le nombre de cannibales ou de missionnaires à installer dans la barque (tapez 1 pour cannibale, 2 pour missionnaire et 0 s'il n'y a pas de deuxième passager). Puis la barque déposera ses voyageurs sur l'autre rive et devra en ramener au moins un sur la première.

Attention : dès que les cannibales deviennent plus nombreux que les missionnaires sur l'une des rives, ils entraînent automatiquement leurs prisonniers dans leurs marmites. Dans ce cas, tapez sur n'importe quelle touche et vos personnages ressusciteront pour un nouvel essai.

A vous de trouver la meilleure tactique.



L'ORIC A L'AFFICHE





```

10 REM MISSIONNAIRES
20 REM -----
30 GOSUB 1000
40 C(1)=94:C(2)=106:C(3)=94:C(4)=106
50 P(1)=-1:P(2)=5
60 P(3)=25:P(4)=31
70 B=1:B(1)=14:B(2)=20
80 M(1)=3:M(2)=3:M(3)=0:M(4)=0
100 INK 0:PAPER 1:CLS
110 PLOT 1,16,"-----"
120 PLOT 26,16,"-----"
130 PLOT 6,17,"{1":PLOT 6,18,"}" +CHR$(126)
140 PLOT 32,17,"{1":PLOT 32,18,"}" +CHR$(126)
280 REM -----
290 REM LE JEU
300 PLOT B(8),14,"      "
310 IF B=1 THEN PLOT B(8)-1,15,"y      " ELSE PLOT B(8),15,"
z"
320 PLOT B(8),16,"vwvwx"
330 GOSUB 900
340 IF M(3)+M(4)=6 THEN 600
350 IF (M(1)>M(2) AND M(2)<>0) OR (M(3)>M(4) AND M(4)<>0) THEN 700
390 FOR I=1 TO 2
400 PLOT 5,22,"PASSAGER"+STR$(I)+CHR$(0)+"CAN.=1/MIS.=2 ?"
410 GET P:IF P<0 OR P>2 THEN 410
420 IF I=1 AND P=0 THEN 410
430 IF P=0 THEN 470
440 IF B=1 THEN 441 ELSE 445
441 IF M(P)>0 THEN M(P)=M(P)-1:GOTO 450 ELSE 410
445 IF M(B+P)>0 THEN M(B+P)=M(B+P)-1 ELSE 410
450 PLOT B(8)-1+2*I,14,CHR$(C(P))+CHR$(C(P)+1)
460 PLOT B(8)-1+2*I,15,CHR$(C(P)+2)+CHR$(C(P)+3)
470 NEXT I
480 PLOT 3,22,"      "
490 GOSUB 900
500 PLOT B(8),14,"      "
510 PLOT B(8)-1,15,"      "
520 PLOT B(8),16,"      "
530 B=B+1:IF B=3 THEN B=1
540 IF B=2 THEN M(3)=3-M(1):M(4)=3-M(2) ELSE M(1)=3-M(3):M(2)=3-M(4)
550 GOTO 300
580 REM -----
590 REM GAGNE
600 PLOT 4,22,CHR$(12)+"VOUS AVEZ ECHAPPE A LA MARMITTE"
610 GET A$
620 GOTO 70
680 REM -----
690 REM PERDU
700 PLOT 4,22,CHR$(12)+"VOUS AVEZ FINI DANS LA MARMITTE"
710 GET A$
720 GOTO 70
880 REM -----
890 REM AFFICHE LE JEU
900 FOR I=1 TO 4
910 FOR J=1 TO 3
920 G=C(I)
930 FOR K=10 TO 15
940 IF J<=M(I) THEN PLOT P(I)+J*2,K,CHR$(G)+CHR$(G+1):GOTO 960
950 PLOT P(I)+J*2,K,"      "
960 G=G+2:NEXT K
970 NEXT J:NEXT I

```





```

980 RETURN
985 REM -----
990 REM GRAPHIQUES
1000 DATA 1,3,7,51,31,51,7,14,32,48,56,51,62,51,56,28
1010 DATA 24,16,22,17,20,19,8,6,6,2,26,34,10,50,4,24
1020 DATA 1,1,1,2,6,4,8,8,32,32,32,16,24,8,4,4
1030 DATA 18,18,22,38,42,18,4,4,18,18,26,25,21,18,8,8
1040 DATA 7,7,7,7,7,5,5,5,56,56,56,56,56,40,40,40
1050 DATA 5,5,5,5,5,5,30,62,40,40,40,40,40,40,30,31
1060 DATA 3,12,16,48,63,42,32,44,48,12,2,3,63,43,1,13
1070 DATA 33,33,36,34,17,40,36,18,33,33,9,17,34,5,9,18
1080 DATA 9,5,13,19,32,32,32,32,36,40,44,50,1,1,1,1
1090 DATA 38,42,42,42,42,42,39,33,25,21,21,21,21,57,33
1100 DATA 33,33,63,8,8,8,8,8,33,33,63,4,4,4,4,4
1110 DATA 8,8,15,2,2,2,30,30,4,4,60,16,16,16,30,30
1120 DATA 63,32,32,16,8,4,2,1,63,0,0,0,0,0,0,63
1130 DATA 63,1,1,2,4,8,16,32,16,16,56,20,20,18,18,17
1140 DATA 2,2,7,10,10,18,18,34
1150 DATA 2,1,2,15,56,40,24,12,16,8,16,60,7,5,6,12
1160 DATA 11,8,8,10,47,21,10,21,52,4,4,44,61,42,20,42
1200 FOR I=0 TO 263
1210 READ A:POKE 46832+I,A
1220 NEXT I
1230 RETURN

```



LE PROGRAMME

Lignes 30 à 140 : initialisation

Après la définition des caractères graphiques (sous-programme en 1000), on définit les tableaux **C**, **P**, **B** et **M** pour connaître, à tout moment, la composition du jeu sur chaque berge (**M**), leurs positions sur l'écran (**P**), les caractères graphiques qui les composent (**C**) et la position sur l'écran de la barque (**B**).

Lignes 300 à 550 : le jeu

La variable **B** donnant le numéro de berge (**B**=1 gauche, **B**=2 droite), on affiche la barque à **B(B)**.

Puis les personnages sont affichés par le sous-programme en 900, en fonction de **M()** et à la position **P()**.

Dès la ligne 340, on vérifie s'ils sont tous sur la berge de droite. Si c'est le cas, on va en ligne 600 pour un message de félicitations.

La ligne 350 teste si le nombre de cannibales n'est pas supérieur à celui des missionnaires. C'est en ligne 700 que l'on va dans ce cas.

Puis c'est la saisie des passagers, effectuée par une boucle **FOR/NEXT**. On vérifie, bien évidemment, qu'il y ait au moins un passager (ligne 420) et qu'il existe bien sur la berge les voyageurs que l'on veut embarquer (lignes 441 et 445).

Ces vérifications étant faites, on affiche leurs têtes sur la barque (lignes 450 et 460).

Pour finir cette partie du programme, on affiche à nouveau les personnages (sous-programme en 900), on efface la barque et on réajuste le tableau **M**, suivant le numéro de la berge.

Lignes 900 à 980 : affiche le jeu

En fonction des tableaux **M**, **P**, et **C**, on affiche à l'écran les six personnages sur les deux berges. La variable **G** pointe le premier des caractères graphiques.

L'ORIC A L'AFFICHE

[illegible]

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

PARITÉS

Dès que vous aurez rentré le nombre de joueurs (jusqu'à 5) et leurs prénoms, dix-huit paires de figures diverses (carte, téléphone, pièce de monnaie...) se dessinent faces cachées sur votre écran.

Elles se présentent sur six lignes de six colonnes, le but du jeu étant de retrouver les dix-huit paires mélangées.

Pour cela, il faut retourner, chacun à son tour, deux cartes de façon à situer les positions des paires pour les reconstituer.

L'ORIC demande les coordonnées des cartes sous forme du numéro de ligne, puis du numéro de colonne.

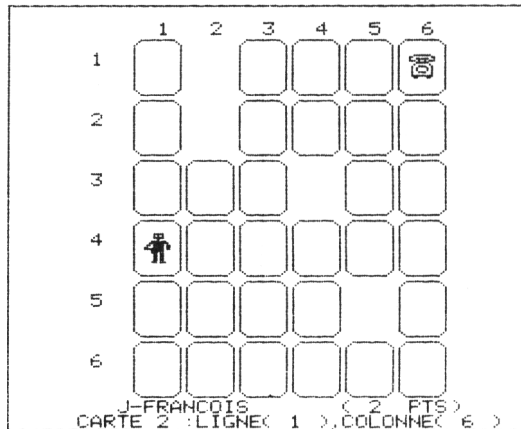
Il refuse les cartes déjà mises en paire et celles dont les coordonnées sont erronées.

Dans le cas où l'un des joueurs trouve une paire, les cartes sont effacées de l'écran.

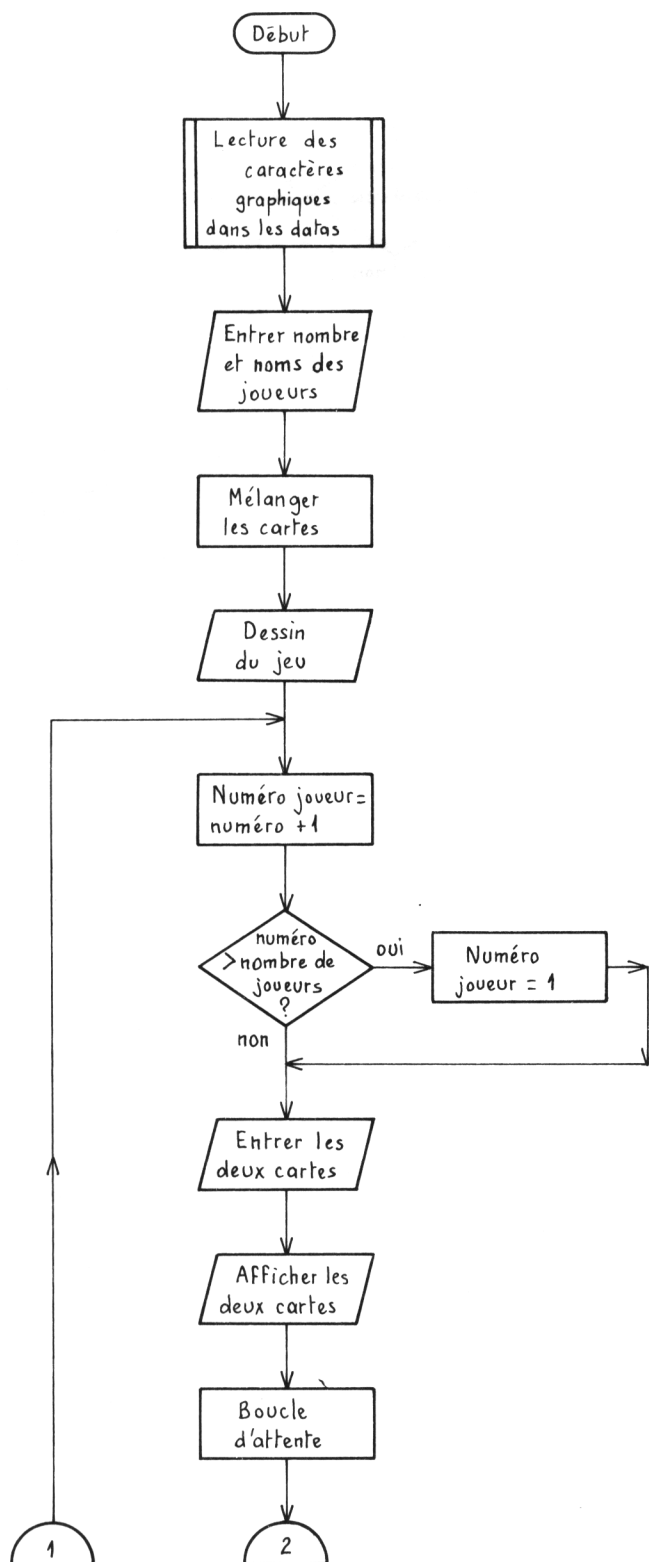
Chaque paire trouvée fait obtenir un point et le gagnant peut rejouer.

Quand toute la grille est vide, les résultats de tous les joueurs s'affichent.

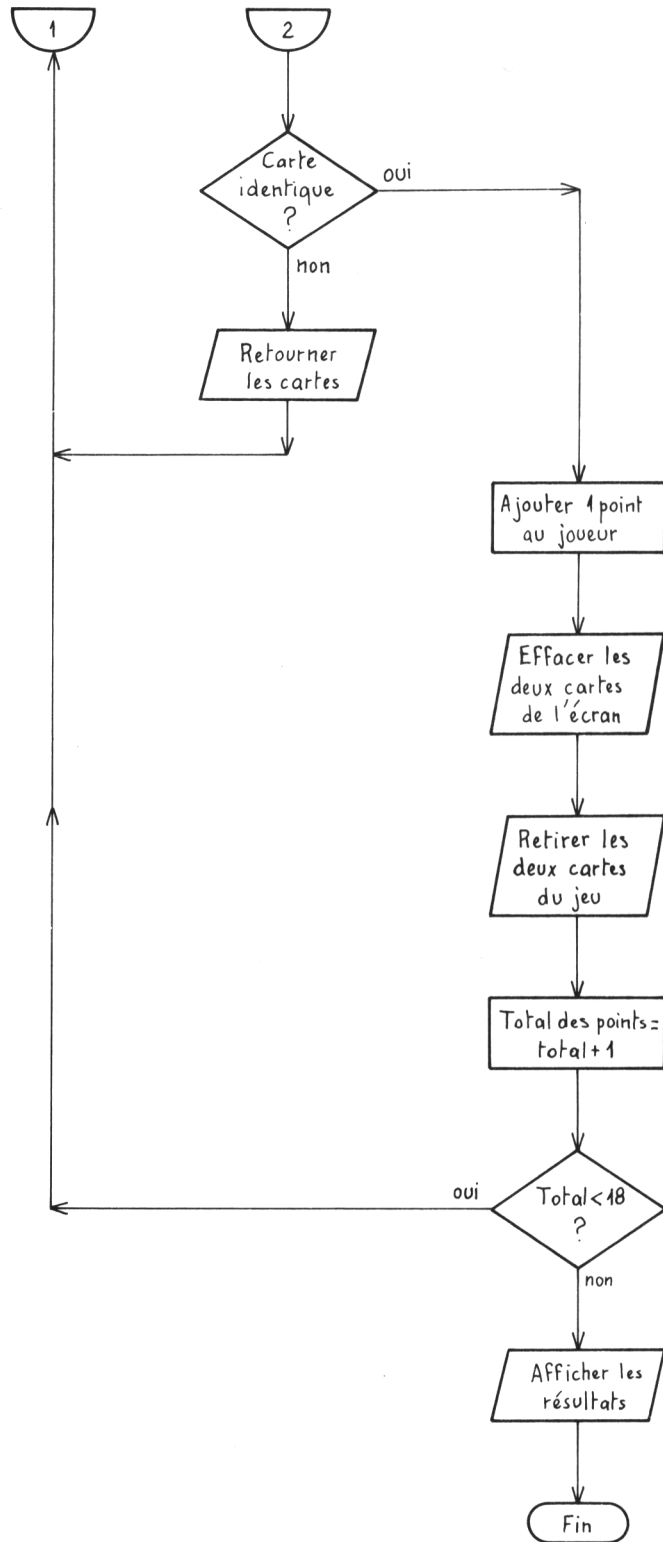
Ce jeu, apparemment simple, demande cependant une grande patience et une grande attention.



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE





```

10 REM PARITES
20 REM -----
30 GOSUB 2000
40 INK 5:PAPER 0:CLS
50 INPUT"NB DE JOUEURS ";JN
60 IF JN<2 OR JN>5 THEN PRINT"ENTRE 2 ET 5 !!":GOTO 50
70 FOR I=1 TO JN
80 PRINT:PRINT"NOM DU JOUEUR ";I;:INPUT N$(I)
90 N$(I)=LEFT$(N$(I)+",15)
100 NEXT I
180 REM -----
190 REM INITIALISATION
200 K=40
210 FOR I=1 TO 3
220 FOR J=1 TO 6
230 C(I,J)=K:C(I+3,J)=K
240 K=K+4
250 NEXT J
260 NEXT I
290 CLS:PLOT 10,10,"MELANGE DES CARTES:"
300 FOR I=1 TO 120
310 L1=INT(RND(1)*6)+1
320 L2=INT(RND(1)*6)+1
330 C1=INT(RND(1)*6)+1
340 C2=INT(RND(1)*6)+1
350 A=C(L1,C1):C(L1,C1)=C(L2,C2):C(L2,C2)=A
360 PLOT 30,10,STR$(120-I)+" "
370 NEXT I
380 REM -----
390 REM AFFICHE LES CARTES
400 CLS
410 FOR I=1 TO 24:PLOT 8,I,9:PLOT 33,I,8:NEXT I
420 FOR I=1 TO 6:PLOT 6+I*4,0,STR$(I):NEXT I
430 FOR J=0 TO 5
440 PLOT 5,J*4+2,STR$(J+1)+CHR$(5)
450 FOR I=0 TO 5
460 FOR K=1 TO 4
470 PLOT 9+I*4,J*4+K,A$(K)
480 NEXT K,I,J
580 REM -----
590 REM LE JEU
600 NJ=NJ+1:IF NJ>JN THEN NJ=1
610 PLOT 7,25,CHR$(4)+N$(NJ)+" (" +STR$(P(NJ))+CHR$(4)+" PTS)
620 FOR I=1 TO 2
630 PLOT 4,26,CHR$(1)+"CARTE"+STR$(I)+CHR$(1)+" :LIGNE(   ),COLONNE(
)"
640 GETA$:Y(I)=VAL(A$):IF Y(I)<10RY(I)>6 THEN 640 ELSE PLOT20,26,STR$(Y(I))
650 GETA$:X(I)=VAL(A$):IF X(I)<10RX(I)>6 THEN 630 ELSE PLOT33,26,STR$(X(I))
660 IF SCRNI(5+X(I)*4,Y(I)*4)=5 THEN 630
670 IF I=2 AND X(1)=X(2) AND Y(1)=Y(2) THEN 630
680 C=C(X(I),Y(I))
690 PLOT 10+(X(I)-1)*4,2+(Y(I)-1)*4,CHR$(C)+CHR$(C+2)
700 PLOT 10+(X(I)-1)*4,3+(Y(I)-1)*4,CHR$(C+1)+CHR$(C+3)
710 NEXT I
720 WAIT 300
730 IF C(X(1),Y(1))=C(X(2),Y(2)) THEN 800
740 FOR I=1 TO 2
750 PLOT 10+(X(I)-1)*4,2+(Y(I)-1)*4,CHR$(5)+CHR$(5)
760 PLOT 10+(X(I)-1)*4,3+(Y(I)-1)*4,CHR$(5)+CHR$(5)
770 NEXT I
780 GOTO 600

```





```

800 P(NJ)=P(NJ)+1
810 FOR I=1 TO 2:FOR J=1 TO 4
820 PLOT 9+(X(I)-1)*4,J+(Y(I)-1)*4,CHR$(5)+CHR$(5)+CHR$(5)+CHR$(5)
830 NEXT J,I
840 TP=TP+1
850 IF TP<18 THEN 610
880 REM -----
890 REM RESULTATS
900 CLS
910 PRINT TAB(25); "RESULTATS":PRINT:PRINT:PRINT:PRINT
920 PRINT "NOM"; TAB(35); "POINTS":PRINT:PRINT
930 FOR I=1 TO JN
940 PRINT N$(I); TAB(25); P(I):PRINT
950 P(I)=0
960 NEXT I
970 INPUT AS
980 TP=0:GOTO 200
1980 REM -----
1990 REM GRAPHIQUES
2000 DATA 0,3,4,8,16,16,16,16,16,16,16,16,8,4,3,0
2010 DATA 0,48,8,4,2,2,2,2,2,2,2,4,8,48,0
2020 DATA 0,63,0,0,0,0,0,0,0,0,0,0,0,63,0
2030 DATA 16,16,16,16,16,16,16,16,2,2,2,2,2,2,2
2040 DATA 3,2,3,0,7,15,27,51,63,7,7,3,3,3,7
2050 DATA 56,40,56,32,60,62,59,59,59,56,24,24,24,24,28
2060 DATA 0,1,15,8,8,63,32,39,39,32,32,39,36,36,32,63
2070 DATA 0,32,48,16,17,61,5,37,37,5,5,39,36,36,4,60
2080 DATA 0,3,5,8,16,16,32,49,33,17,16,8,5,3,0,0
2090 DATA 0,32,16,8,12,20,34,6,2,4,4,8,16,32,0,0
2100 DATA 0,3,4,8,18,22,42,34,34,34,18,18,8,4,3,0
2110 DATA 0,48,8,4,2,58,33,49,33,33,2,2,4,8,48,0
2120 DATA 0,15,48,39,58,7,8,19,20,20,19,16,31,12,0,0
2130 DATA 0,60,3,57,23,56,4,50,10,10,50,2,62,12,0,0
2140 DATA 63,32,42,46,36,32,34,35,33,32,42,46,36,32,63,0
2150 DATA 62,2,42,58,18,2,34,34,2,2,42,58,18,2,62,0
2160 DATA 63,17,17,63,4,4,63,17,17,63,4,4,63,17,17,63
2170 DATA 63,4,4,63,17,17,63,4,4,63,17,17,63,4,4,63
2180 DATA 63,32,36,32,32,36,32,63,32,32,32,32,36,32,63,0
2190 DATA 63,1,9,1,1,9,1,63,1,9,1,1,1,1,63,0
2200 DATA 1,3,5,9,19,33,37,33,41,34,37,42,48,63,0,0
2210 DATA 63,1,21,1,21,1,21,1,63,2,20,40,16,32,0,0
2220 DATA 17,32,36,24,45,5,47,17,41,5,5,1,1,1,1,3
2230 DATA 8,40,17,34,13,41,62,52,50,41,36,42,32,32,48
2240 DATA 0,0,1,2,12,30,63,63,30,12,0,0,1,1,0,0
2250 DATA 15,49,2,4,4,8,8,16,16,16,24,60,62,62,60,24
2260 DATA 1,1,3,2,3,2,3,2,7,2,7,15,5,11,20,56
2270 DATA 0,0,32,32,32,32,32,32,48,32,48,56,16,40,20,14
2280 DATA 0,0,0,1,3,7,63,31,15,31,63,7,3,1,0,0
2290 DATA 0,0,0,0,32,48,62,60,56,60,62,48,32,0,0,0
2300 DATA 0,0,0,0,1,11,63,31,15,15,15,31,63,15,0,0
2310 DATA 0,0,0,32,48,60,63,62,62,60,62,63,60,48,0,0
2320 DATA 0,0,0,0,0,0,3,7,31,63,63,63,31,15,0,0
2330 DATA 0,0,3,4,10,51,51,33,32,32,48,48,32,0,0,0
2340 DATA 1,3,7,15,63,16,20,22,16,17,17,20,18,9,4,3
2350 DATA 0,32,48,56,62,4,22,54,6,6,6,22,38,13,23,37
2380 DATA 0,0,0,12,30,63,63,63,63,31,15,7,3,1,0,0
2390 DATA 0,0,0,24,60,62,62,62,62,60,56,48,32,0,0,0
2400 DATA 0,0,0,1,3,7,3,9,29,63,29,9,3,7,0,0
2410 DATA 0,0,0,0,32,48,32,8,28,62,28,8,32,48,0,0
2990 CLS:PAPER 0:INK 1:PLOT 10,10,"LECTURE DES DATAs:"
3000 FOR I=0 TO 639:READ A:POKE 47360+I,A:PLOT 30,10,STR$(639-I)+"
":NEXT
3010 A$(1)=CHR$(32)+CHR$(36)+CHR$(36)+CHR$(34)
3020 A$(2)=CHR$(38)+CHR$(5)+CHR$(5)+CHR$(39)
3030 A$(3)=A$(2)
3040 A$(4)=CHR$(33)+CHR$(37)+CHR$(37)+CHR$(35)
3050 RETURN

```



LE PROGRAMME

Lignes 30 à 370 : initialisation

On place dans la totalité du deuxième jeu de caractères les représentations graphiques des cartes du jeu et, dans le tableau A\$, le dessin des cartes (sous-programme en 2000).

Puis l'on saisit au clavier le nombre et le nom des joueurs.

Le numéro des caractères graphiques de chaque carte étant placé dans le tableau C(,), on mélange ce dernier en utilisant les quatre nombres aléatoires C1, C2, L1 et L2.

Lignes 400 à 480 : affichage du jeu

Comme tous les caractères graphiques sont placés dans le deuxième jeu, on place sur toute la colonne 8 des attributs 9 et sur la colonne 33 des attributs 8.

Lignes 600 à 780 : le jeu

Après la saisie dans la variable A\$, les numéros de lignes et de colonnes sont placés respectivement dans les tableaux Y() et X().

Dans le cas où ces deux cartes sont identiques, on saute directement à la ligne 800. Sinon, on les retourne après une petite boucle d'attente.

Lignes 800 à 850 : cas des cartes identiques

On incrémente le total des points du joueur (tableau P), puis l'on efface de l'écran les cartes désignées.

Si le total des cartes effacées est inférieur à 18, on continue normalement le jeu mais en gardant le même joueur (ligne 610 au lieu de 600).

Lignes 900 à 980 : résultats

On affiche le nom et le nombre de points de chaque joueur (tableaux N\$ et P).

DATE	NOM DU PROGRAMME : PARITES					PAGE 1 / 1
IDENTIFICATION des VARIABLES						
NOM SYMBOLIQUE	TYPE (*)	ORIGINE				OBSERVATIONS
		Clavier	Programme	Cassette	Disquette	
A, , ,	Variable	X				Utilisée pour le mélange des cartes
A,\$, ,	Variable	X				Caractère saisi au clavier
A,\$,(,)	Tableau	X				Dessin des cartes
C, , ,	Variable	X				Pointeur des caractères graphiques pour l'affichage des cartes
C,(,)	Tableau	X				Tableau des cartes
C,1, ,	Variable	X				} Nombres aléatoires pour le mélange des cartes
C,2, ,	Variable	X				
I, , ,	Indice	X				Boucle FOR/NEXT
J, , ,	Indice	X				Boucle FOR/NEXT
J,N, ,	Compteur	X				Nombre de joueurs
K, , ,	Variable	X				Pointeur des caractères graphiques des cartes
L,1, ,	Variable	X				} Nombres aléatoires pour le mélange des cartes
L,2, ,	Variable	X				
N,\$,(,)	Tableau	X				Noms des joueurs
N,J, ,	Compteur	X				Numéro du joueur
P,(,)	Tableau	X				Total des points de chaque joueur
T,P, ,	Compteur	X				Total des points
X,(,)	Tableau	X				} Coordonnées des cartes choisies
Y,(,)	Tableau	X				
, , , ,						
, , , ,						
, , , ,						
, , , ,						

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

CHAMP DE CAROTTES

C'est un jour d'hiver bien froid et Maître Lapin ne voit plus qu'une carotte dans le fond du champ. Il va essayer d'aller la déterrer, tout en sachant que les villageois ont tendu plein de pièges sous l'herbe.

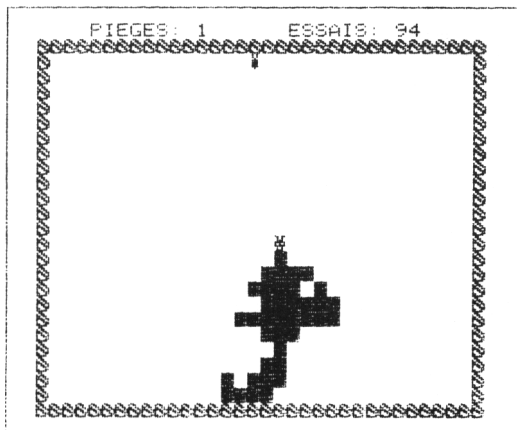
Heureusement qu'il porte sur lui son fidèle détecteur électronique...

Le détecteur inscrit, en haut à gauche de l'écran, le nombre des pièges qui entourent le lapin, en comptant aussi tous ceux qui composent la bordure du champ.

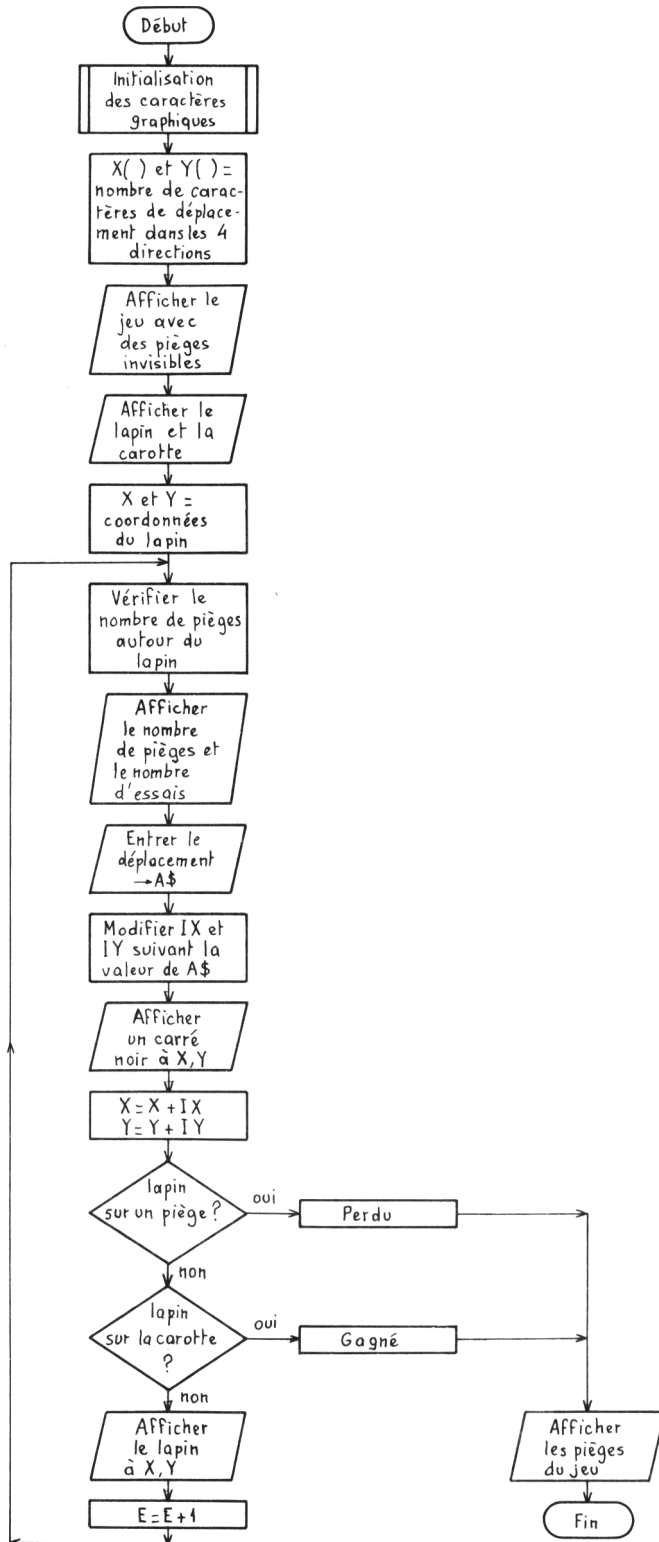
Vous n'avez plus qu'à guider logiquement votre lapin en utilisant les quatre touches de déplacement.

Que vous ayez évité ou non les pièges, vous découvrirez, à chaque fin de jeu, leur plan précis avec votre nombre d'essais parfois impressionnant.

Si votre lapin, toujours affamé, veut repartir à la recherche d'une autre carotte, appuyez sur n'importe quelle touche.



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE

```

10 REM CHAMP DE CAROTTES
20 REM -----
30 GOSUB 1000
40 X(1)=1:X(2)=-1:X(3)=0:X(4)=0-
50 Y(1)=0:Y(2)=0:Y(3)=1:Y(4)=-1
60 CLS:PAPER 3:INK 0
200 FOR I=1 TO 100
210 X=INT(RND(1)*32)+3
220 Y=INT(RND(1)*21)+3
230 IF SCRN(X,Y)<>32 THEN 210
240 PLOT X,Y,100
250 NEXT I
280 REM -----
290 REM AFFICHE LE JEU
300 FOR I=2 TO 35
310 PLOT I,1,98:PLOT I,25,98
320 IF I>25 THEN 340
330 PLOT 2,I,98:PLOT 35,I,98
340 NEXT I
350 PLOT 18,2,99
360 X=18:Y=24:PLOT X,Y,97
370 E=0
380 REM -----
390 REM NB DE PIEGES
400 P=0
410 FOR I=1 TO 4
420 A=SCRN(X+X(I),Y+Y(I)):IF A=100 OR A=98 THEN P=P+1
430 NEXT I
440 PLOT 5,0,CHR$(4)+"PIEGES:"+CHR$(1)+MID$(STR$(P),2)
450 PLOT 20,0,CHR$(4)+"ESSAIS:"+CHR$(1)+MID$(STR$(E),2)
480 REM -----
490 REM DEPLACEMENTS
500 GET AS
510 IF AS=CHR$(9) THEN IX=1:IY=0:GOTO 600
520 IF AS=CHR$(8) THEN IX=-1:IY=0:GOTO 600
530 IF AS=CHR$(10) THEN IX=0:IY=1:GOTO 600
540 IF AS=CHR$(11) THEN IX=0:IY=-1:GOTO 600
550 GOTO 500
600 PLOT X,Y,101
610 X=X+IX:Y=Y+IY
620 IF SCRN(X,Y)=98 OR SCRN(X,Y)=100 THEN 700
630 IF SCRN(X,Y)=99 THEN 710
640 PLOT X,Y,97
650 E=E+1
660 GOTO 400
680 REM -----
690 REM FIN DE PARTIE
700 SHOOT:WAIT 100:GOTO 800
710 ZAP:WAIT 100
800 FOR I=3 TO 35
810 FOR J=3 TO 23
820 IF SCRN(I,J)=100 THEN PLOT I,J,98
830 NEXT J,I
840 GET AS
850 GOTO 60
980 REM -----
990 REM GRAPHIQUES
1000 DATA 54,20,28,62,42,62,20,28,24,44,54,59,45,39,51,62
1010 DATA 20,20,8,28,28,28,28,8,0,0,0,0,0,0,0
1015 DATA 63,63,63,63,63,63,63,63
1020 FOR I=0 TO 39
1030 READ A:POKE 46856+I,A
1040 NEXT I
1100 RETURN

```

LE PROGRAMME

Lignes 30 à 250 : initialisation

Après l'initialisation des caractères graphiques (sous-programme en 1000), on place dans les tableaux **X()** et **Y()** le nombre de caractères de déplacement correspondant aux quatre directions.

Puis on affiche à l'écran des pièges à des positions aléatoires (100 pièges au total). On se sert pour cela du caractère graphique de code 100 correspondant à un espace blanc. Ils existent donc bien, mais ils sont invisibles sur l'écran.

Lignes 300 à 360 : affiche le jeu

Cette partie du programme affiche simplement des pièges au bord de l'écran (visibles cette fois, code 98), ainsi que le lapin et la carotte.

Les variables **X** et **Y** donnent la position exacte du lapin.

Lignes 400 à 450 : nombre de pièges

Pour connaître le nombre de pièges autour du lapin, il suffit de tester directement sur l'écran, à l'aide des tableaux **X** et **Y**, la présence des codes 100 et 98.

Lignes 500 à 660 : déplacements

Le caractère saisi au clavier étant stocké dans la variable **AS**, on modifie, suivant sa valeur, les variables **IX** et **IY**.

Avant d'afficher le lapin, on place un carré noir sur son ancienne position.

La ligne 620 teste si le lapin n'est pas sur un piège (code 100 pour les invisibles, code 98 pour ceux du bord).

La ligne 630, elle, teste la présence éventuelle de la carotte.

Lignes 700 à 850 : fin de partie

Quelle que soit l'issue du jeu, on affiche à l'écran tous les pièges jusqu'alors invisibles en testant toutes les positions de l'écran par une fonction **SCRN**.

L'ORIC A L'AFFICHE

0861/9

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

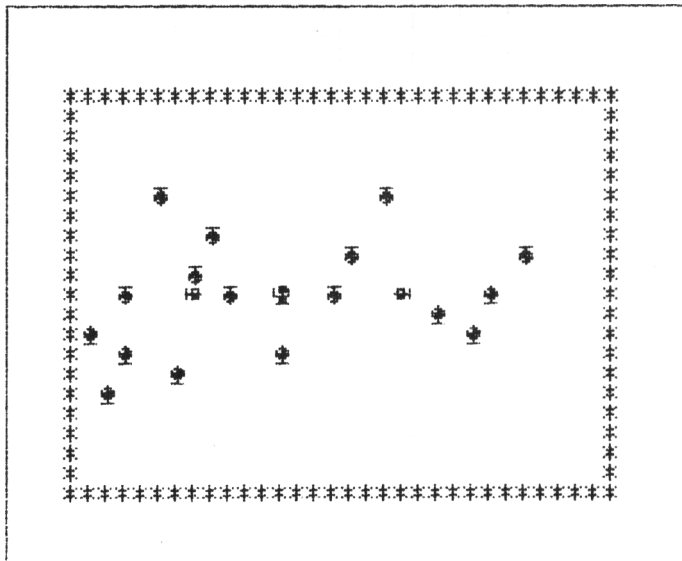
PIRANHAS

Ne paniquez pas si vous voulez sauver cet imprudent explorateur tombé dans l'Amazonie au milieu des piranhas.

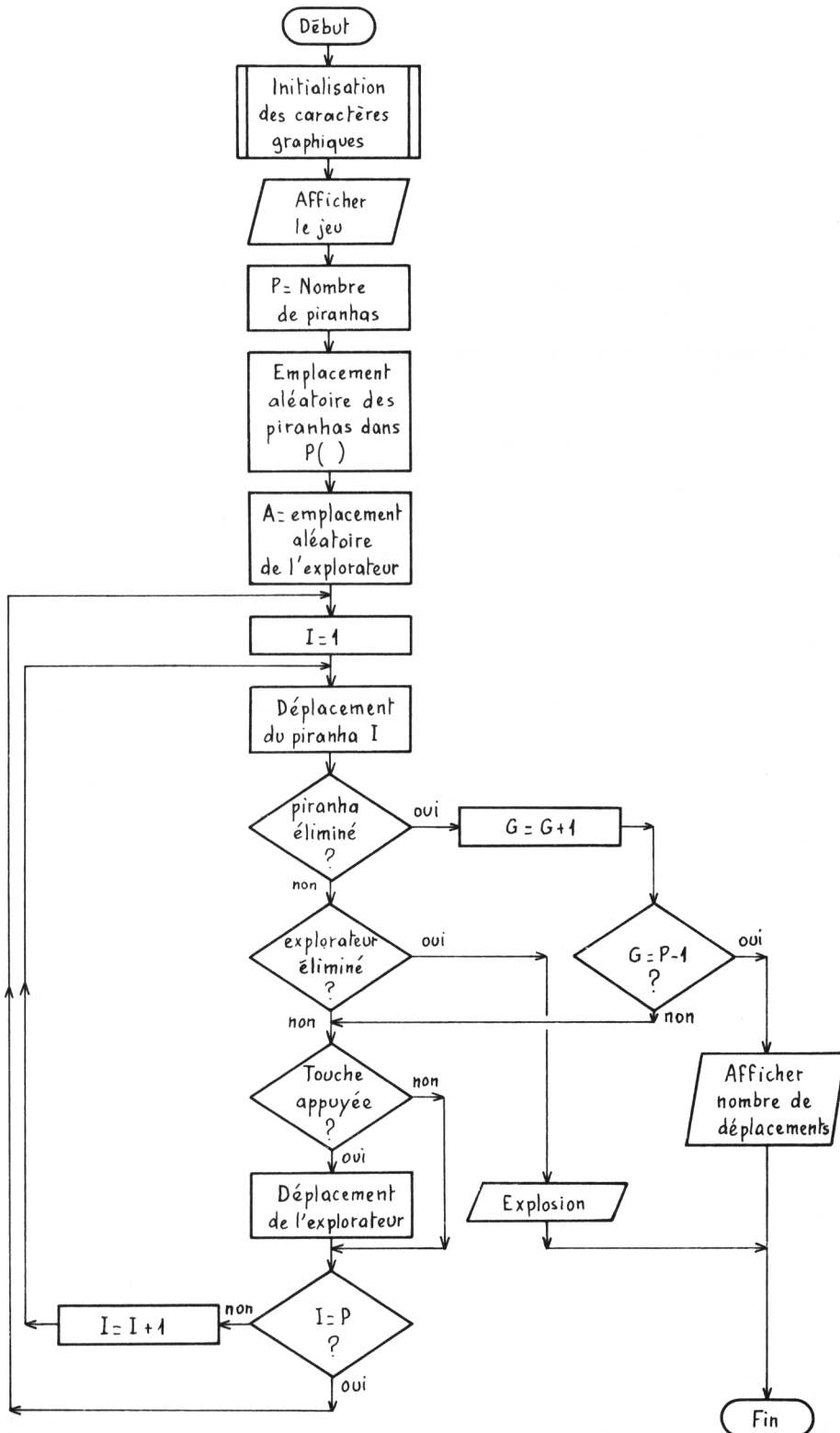
Sa seule chance est que ces féroces poissons s'entre-dévorent lorsqu'ils se retrouvent côte à côte.

Tout en déplaçant l'explorateur dans les quatre directions avec les quatre flèches du clavier, vous imaginerez sûrement une tactique pour forcer les piranhas à se rencontrer jusqu'au dernier.

Vous devrez alors être prêt pour une autre traversée de l'Amazonie, dès que vous aurez appuyé sur n'importe quelle touche.



L'ORIC A L'AFFICHE




```

10 REM PIRANHAS
20 REM -----
30 GOSUB 2000
40 G=0:D=0
50 INK 1:PAPER 6:CLS
60 FOR I=3 TO 34
70 PLOT I,3,42:PLOT I,23,42
80 IF I<24 THEN PLOT 3,I,42:PLOT 34,I,42
90 NEXT I
100 P=INT(RND(1)*11)+20
110 FOR I=1 TO P
120 P(I)=#BB80+INT(RND(1)*30)+INT(RND(1)*19)*40+205
140 POKE P(I),97
150 NEXT I
160 A=#BB80+INT(RND(1)*30)+INT(RND(1)*19)*40+205
170 POKE A,101
180 GET A$
280 REM -----
290 REM DEPLACEMENT PIRANHAS
300 FOR I=1 TO P
310 IF P(I)=0 THEN 400
320 POKE P(I),32
330 FOR J=1 TO 4
340 IF PEEK(P(I)+T(J))=101 THEN 800
350 IF PEEK(P(I)+T(J))>96 THEN 380
360 NEXT J
370 GOTO 410
380 P(I)=0:G=G+1:PING
390 IF G=P-1 THEN 900
400 GOTO 600
410 IF INT((A-#BB80)/40)>INT((P(I)-#BB80)/40) THEN K=2:GOTO 450
420 IF INT((A-#BB80)/40)<INT((P(I)-#BB80)/40) THEN K=4:GOTO 450
430 IF A-1=P(I) THEN K=1:GOTO 450
440 K=3
450 IF PEEK(P(I)+T(K))=42 THEN 600
460 P(I)=P(I)+T(K)
580 REM -----
590 REM DEPLACEMENT JOUEUR
600 C$=KEY$:IF C$="" THEN 700
610 C=ASC(C$)
620 FOR J=1 TO 4
630 IF C=C(J) THEN 650
640 NEXT J:GOTO 700
650 IF PEEK(A+T(J))=42 THEN 700
660 IF PEEK(A+T(J))>96 THEN 800
670 POKE A,32
680 A=A+T(J)
690 POKE A,101:D=D+1
700 POKE P(I),G(K)
720 NEXT I
730 GOTO 300
780 REM -----
790 REM PERDU
800 POKE A,32
810 EXPLODE
820 GET A$
830 GOTO 40
880 REM -----
890 REM GAGNE
900 PING:WAIT 60:SHOOT:WAIT 60:ZAP
910 PAPER 0:PLOT 20,25,"DEPLACEMENT:"+STR$(D)
920 GET A$

```



```
930 GOTO 40
1900 STOP
1980 REM -----
1990 REM INITIALISATION
2000 DATA 0,46,42,63,46,46,0,0,0,29,21,63,29,29,0,0
2010 DATA 31,4,14,31,31,23,14,4,4,14,23,31,31,14,4,31
2020 DATA 14,47,47,37,63,14,14,31
2030 FOR I=0 TO 39
2040 READ A:POKE 46856+I,A
2050 NEXT I
2060 DIM P(30)
2070 T(1)=1:T(2)=40:T(3)=-1:T(4)=-40
2080 C(1)=9:C(2)=10:C(3)=8:C(4)=11
2090 G(1)=97:G(2)=99:G(3)=98:G(4)=100
2100 RETURN
```



LE PROGRAMME

Lignes 30 à 180 : initialisation

Après l'initialisation des caractères graphiques, on place dans le tableau **T()** les directions de déplacement des piranhas, et dans le tableau **C()** les valeurs ASCII des touches de déplacement (sous-programme en 2000).

Le nombre de piranhas est déterminé par la variable **P** (nombre aléatoire).

On remplit le tableau **P()** avec les coordonnées des piranhas, **A** donnant celle de l'explorateur.

Lignes 300 à 460 : déplacement des piranhas

Chaque piranha est déplacé sur l'écran dans le sens qui le rapproche le plus de l'explorateur, sauf s'il trouve un congénère sur une case adjacente. Dans ce cas, il s'empresse de l'avaler (lignes 350 et 380 à 400).

Lignes 600 à 730 : déplacement de l'explorateur

On déplace l'explorateur suivant la touche appuyée (saisie par **KEY\$**).

Lignes 800 à 930 : fin de partie

Suivant l'issue de la partie, l'ORIC fait entendre une explosion (perdu !) ou une cloche (gagné !).

Si la partie est gagnée, il affiche en plus le nombre de déplacements.

L'ORIC A L'AFFICHE

[illegible]

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

AWARI

AWARI est un combat stratégique entre l'ordinateur et vous.

L'ORIC vous demande d'entrer votre prénom, puis il dessine le jeu sur l'écran de cette façon :

EXEMPLE 1 :

	3	3	3	3	3	3	
0	PARTIE 1						0
	3	3	3	3	3	3	

Tout au cours d'une partie, l'ordinateur affichera votre prénom, puis un message, soit :

PRENOM
JE JOUE EN

ou :

PRENOM
QUEL EST VOTRE MOUVEMENT ?

L'ordinateur joue sur les cases du haut et le joueur sur les cases du bas.

Il y a trois pions au départ dans chaque case. Le jeu consiste à déplacer les trois pions d'une case, de un en un, dans les cases suivantes et dans le sens inverse des aiguilles d'une montre :

EXEMPLE 2 :
MOUVEMENT

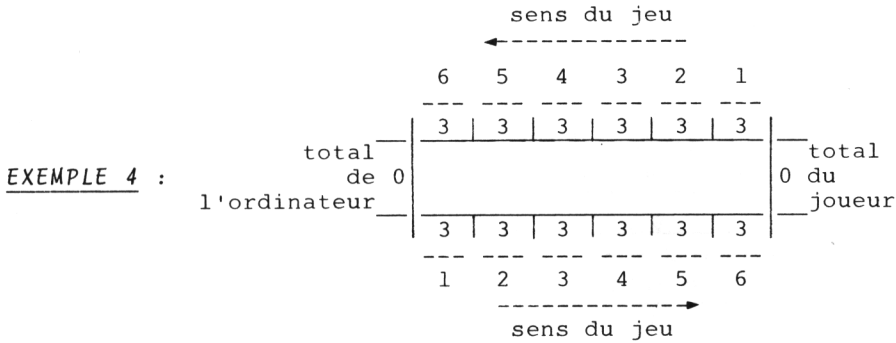
	3	3	3	3	3	3	
0	PARTIE 1						0 +1
	3	3	3	3	3	3	
				-3	+1	+1	

EXEMPLE 3 :
RESULTAT

	3	3	3	3	3	3	
0	PARTIE 1						1
	3	3	3	0	4	4	

L'ORIC A L'AFFICHE

Il conviendra de considérer les cases sous les numéros suivants :



EXEMPLE 4 :

Ainsi, dans l'exemple 2, la case mouvementée était la case 4 :

PRENOM

QUEL EST VOTRE MOUVEMENT ?

Taper 4.

Si le dernier pion placé s'arrête dans une case TOTAL, le joueur doit rejouer. Si le dernier pion s'arrête avant ou après cette case, c'est au joueur adverse de jouer. Dans l'exemple 2, c'est au joueur (et non à l'ordinateur) de rejouer.

La partie s'arrête quand toutes les cases de l'un des deux joueurs sont vides.

L'ordinateur signale alors qui a gagné. Le nombre de parties gagnées reste d'ailleurs affiché en permanence sur l'écran.

Puis le jeu initial se redessine, vous invitant à refaire une partie...

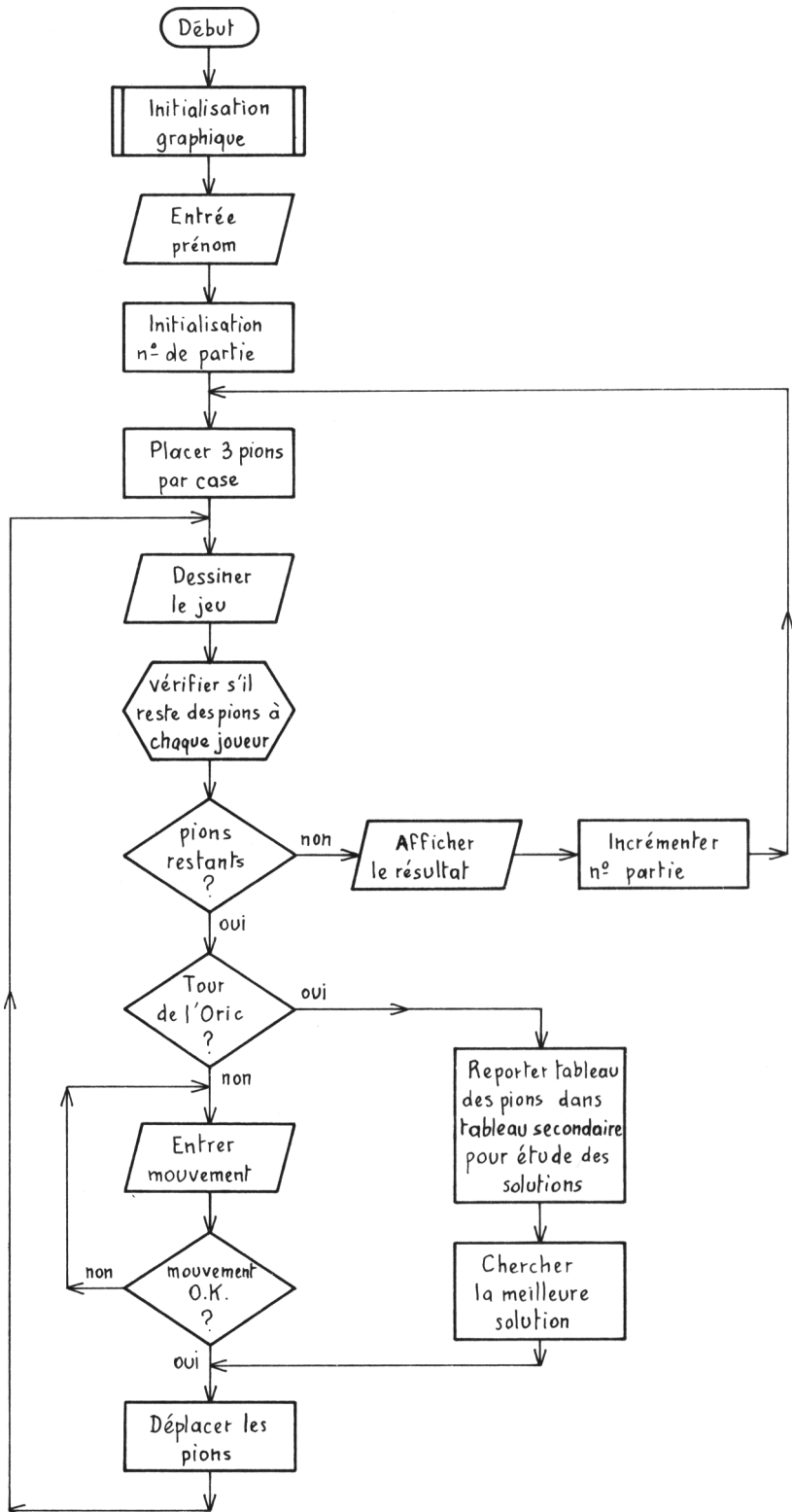
AWARI

1	0	2	1	4	5	0	1
5	1	PARTIE : 2				1	1
1	1	5	4	4	4	0	1

J-FRANCOIS
QUEL EST VOTRE MOUVEMENT ?

ORIC: 0
J-FRANCOIS: 1

L'ORIC A L'AFFICHE



```

10 REM AWARI
20 REM -----
30 GOSUB 3000
40 DIM B(14),K(14),G(14),H(14)
50 CLS:PAPER 4:INK 1
60 INPUT "VOTRE PRENOM ";P$
70 PA=1:T1=0:T2=0
80 E=INT(RND(1)+.5)
90 CLS:PRINT
100 PRINT TAB(27);CHR$(4);CHR$(27)+"JAWARI";CHR$(4)
180 REM -----
190 REM AFFICHE LE JEU
200 FOR I=1 TO 13:B(I)=3:NEXT I
210 B(7)=0:B(14)=0
220 FOR I=1 TO 37
230 PLOT I,10,"a":PLOT I,12,"a"
240 NEXT I
250 PLOT 3,19,P$
260 PLOT 0,11,LEFT$(STR$(B(14))+" ",3)+CHR$(1)+"b"      PARTIE :
+STR$(PA)
270 PLOT 33,11,CHR$(1)+"b"+STR$(B(7))
280 FOR J=13 TO 8 STEP-1
290 PLOT 3+(13-J)*5,9,CHR$(1)+"b"+LEFT$(STR$(B(J))+" ",3)
300 NEXT J
310 PLOT 33,9,CHR$(1)+"b"
320 FOR J=1 TO 6
330 PLOT 3+(J-1)*5,13,CHR$(1)+"b"+LEFT$(STR$(B(J))+" ",3)
340 NEXT J
350 PLOT 33,13,CHR$(1)+"b"
380 REM -----
390 REM LE JEU
400 B=0
410 FOR I=1 TO 6
420 IF B(I)<>0 THEN B=1
430 NEXT I:IF B=0 THEN 1700
440 FOR I=8 TO 13
450 IF B(I)<>0 THEN B=1
460 NEXT I
470 IF B=0 THEN 1700
480 IF E=1 THEN 900
490 PLOT 3,20,"QUEL EST VOTRE MOUVEMENT ? "
500 GET A$:A=VAL(A$)
510 IF A<1 OR A>6 THEN 500
520 IF B(A)=0 THEN 500
530 PLOT 30,20,STR$(A):GOSUB 740
540 IF E=1 AND A=14 THEN 260
550 IF A=7 THEN 260
560 IF E=1 THEN E=0:GOTO 260
570 E=1:GOTO 260
680 REM -----
690 REM DEPLACEMENT DES PIONS
700 A=J
710 FOR Z=1 TO 14
720 B(Z)=G(Z)
730 NEXT Z
740 P=B(A):B(A)=0
750 FOR I=1 TO P
760 A=A+1:IF A=15 THEN A=1
770 B(A)=B(A)+1
780 NEXT I
790 RETURN
880 REM -----

```



```
890 REM COUP DE L'ORIC
900 PLOT 3,20,"JE JOUE EN
910 FOR J=1 TO 14
920 G(J)=B(J)
930 NEXT J
940 IF B(13)=1 THEN 1500
950 IF B(13)>=3 THEN 1500
960 IF B(12)>=3 THEN 1510
970 IF B(11)>=4 THEN 1520
980 FOR J=13 TO 8 STEP-1
990 IF G(J)=0 THEN 1070
1000 GOSUB 700
1010 IF A=14 THEN 1550
1020 H(J)=A
1030 FOR U=9 TO 13
1040 Z=U+B(U)
1050 IF Z=14 THEN K(J)=K(J)+1
1060 NEXT U
1070 NEXT J
1080 Z=0
1090 IF G(12)>=5 THEN 1510
1100 IF G(8)=1 THEN 1540
1110 IF G(9)=1 THEN 1530
1120 FOR J=8 TO 13
1130 U=K(J)
1140 IF U>Z THEN GOSUB 1560
1150 NEXT J
1160 IF Z>0 THEN 1280
1170 Z=8
1180 FOR J=8 TO 12
1190 U=H(J)
1200 IF U>Z AND U>0 THEN GOSUB 1560
1210 NEXT J
1220 IF A>7 THEN GOTO 1280
1230 Z=7
1240 FOR J=8 TO 13
1250 U=H(J)
1260 IF U<Z AND U>0 THEN GOSUB 1560
1270 NEXT J
1280 FOR U=8 TO 13
1290 K(U)=0
1300 H(U)=0
1310 NEXT U
1320 C=A-7
1330 PLOT 14,20,STR$(C)
1340 J=A
1350 GOSUB 700
1360 GOTO 540
1480 REM -----
1490 REM COUP CHOISI PAR L'ORIC
1500 A=13:GOTO 1280
1510 A=12:GOTO 1280
1520 A=11:GOTO 1280
1530 A=9:GOTO 1280
1540 A=8:GOTO 1280
1550 A=J:GOTO 1280
1560 Z=U: A=J
1570 RETURN
1680 REM -----
1690 REM RESULTATS
1700 IF B(14)>B(7) THEN 1800
1710 IF B(14)<B(7) THEN 1750
```

L'ORIC A L'AFFICHE

```
1720 PLOT 3,20,"NOUS SOMMES EXAEQUO      "  
1730 E=INT(RND(1)+.5)  
1740 GOTO 1850  
1750 PLOT 3,20,"VOUS AVEZ GAGNE          "  
1760 T1=T1+1  
1770 F=1  
1780 GOTO 1850  
1800 PLOT 3,20,"J'AI GAGNE              "  
1810 T2=T2+1  
1820 E=0  
1850 PA=PA+1  
1860 PLOT 3,25,"ORIC:"+STR$(T2)+CHR$(1)+"      "+PS+": "+STR$(T1)  
1870 GET A$  
1880 PLOT 37,11," "  
1890 GOTO 200  
2980 REM -----  
2990 REM GRAPHIQUES  
3000 FOR I=0 TO 15  
3010 READ A:POKE 46856+I,A  
3020 NEXT I  
3030 RETURN  
3100 DATA 0,0,0,63,63,0,0,0,12,12,12,12,12,12,12,12
```

LE PROGRAMME

Lignes 30 à 350 : initialisation

Après l'appel du sous-programme de définition des caractères graphiques (ligne 3000), le programme saisit le prénom du joueur et affiche à l'écran le dessin du jeu.

La variable E détermine le tour du joueur si sa valeur est 0, et le tour de l'ORIC si sa valeur est 1.

Au premier tour, cette variable contient un nombre aléatoire, mais, dès la deuxième partie, c'est toujours le perdant qui commence.

Lignes 380 à 480 : test de fin de partie

Le jeu doit s'arrêter quand un des joueurs n'a plus de pions dans son terrain.

Dans ces lignes, on vérifie à chaque tour cette éventualité. Si c'est le cas, on passe à la ligne 1700.

En ligne 480, le programme se sépare en deux, suivant la valeur de E.

Lignes 490 à 530 : E=0, tour du joueur

Dès l'entrée du mouvement, on vérifie sa validité. En cas de mauvaise réponse, comme il n'est pas possible de passer son tour, la question est de nouveau posée jusqu'à l'obtention d'une réponse correcte.

L'appel du sous-programme de la ligne 740 déplace les pions des cases concernées.

Lignes 880 à 1570 : E=1, tour de l'ORIC

Pour déterminer son choix, l'ORIC utilise trois tableaux pour comparer les solutions possibles et les plus avantageuses.

La solution trouvée est placée dans A comme pour le joueur, ce qui permet d'utiliser le sous-programme de déplacement des pions des lignes 700 à 790.

Lignes 700 à 790 : déplacement des pions

On déplace les pions de la case pointée par A en les distribuant dans les cases suivantes, en considérant que la case 15 (inexistante) correspond à la case 1.

Lignes 1700 à 1890 : résultats

Dès que l'un des joueurs n'a plus de points, on affiche à l'écran le nombre de parties gagnées par chacun ainsi qu'un message de félicitations.

L'ORIC A L'AFFICHE

DATE		NOM DU PROGRAMME : AWARI					PAGE 1 / 1
IDENTIFICATION des VARIABLES							
NOM SYMBOLIQUE	TYPE (*)	ORIGINE				LONG.	OBSERVATIONS
		Clavier	Programme	Cassette	Disquette		
A ₁ _ _	Variable	X					Votre mouvement
A ₁ \$ _ _	Variable	X					Votre mouvement
B ₁ _ _	Aiguillage	X					Si B=Ø : fin de partie
B ₁ (_)	Tableau	X					Nombre de pions dans chaque case
C ₁ _ _	Variable	X					Numéro de la case jouée par l'ORIC
E ₁ _ _	Aiguillage	X					Si E=Ø : tour du joueur Si E=1 : tour de l'ORIC
G ₁ (_)	Tableau	X					Sauvegarde du tableau B()
H ₁ (_)	Tableau	X					Nombre de pions dans chaque case après déplacement
I ₁ _ _	Indice	X					Boucle FOR/NEXT
J ₁ _ _	Indice	X					Boucle FOR/NEXT
K ₁ (_)	Tableau	X					Coefficient de chaque case
P ₁ _ _	Compteur	X					Nombre de pions à déplacer
P ₁ \$ _ _	Variable	X					Votre prénom
P ₁ A ₁ _	Compteur	X					Numéro de la partie
T ₁ 1 _ _	Compteur	X					Nombre de parties gagnées par le joueur
T ₁ 2 _ _	Compteur	X					Nombre de parties gagnées par l'ORIC
U ₁ _ _	Variable	X					Coefficient de la case étudiée
Z ₁ _ _	Indice	X					Boucle FOR/NEXT
_ _ _ _							
_ _ _ _							
_ _ _ _							
_ _ _ _							

4/1982

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

CHASSE AU TRÉSOR

Vous n'avez pas de chance au jeu ?

Alors voilà une chasse au trésor où la ruse et non le pur hasard vous sera utile pour gagner.

Cette chasse au trésor se joue à deux. Une grille de 18 sur 10 numéros s'inscrit devant vous. Il suffit de déplacer les deux chasseurs avec les quatre flèches du clavier, selon l'itinéraire qui rapportera le plus de points (verticalement et horizontalement).

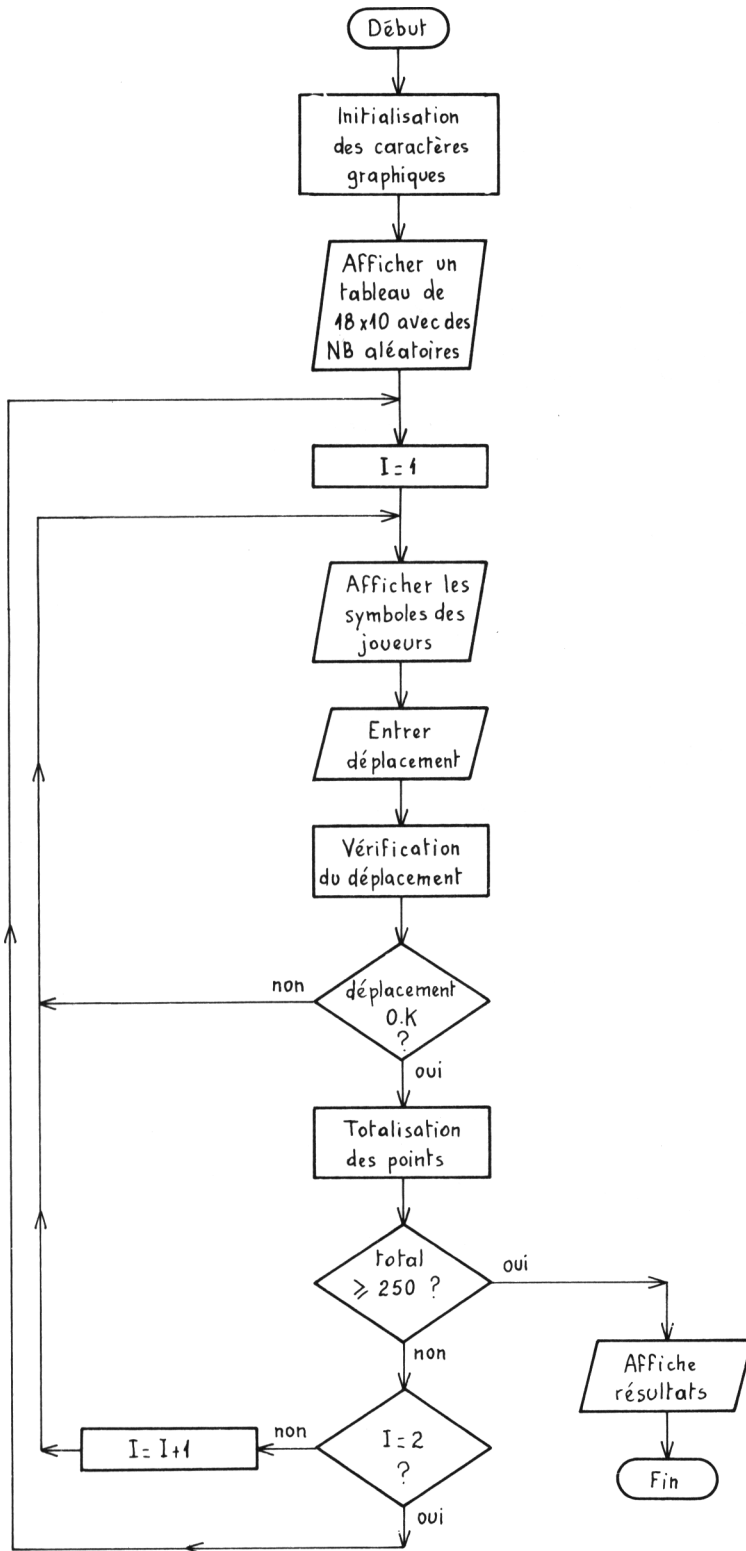
Attention, ne détruisez pas toute une zone sans prévoir un possible retour en arrière fructueux...

La partie s'arrête quand l'un des deux joueurs a obtenu au moins 250 points et les résultats s'affichent alors. Pour repartir, il faudra appuyer sur n'importe quelle touche.

Maintenant, bonne chasse !

0	0	0	0	0	0	0	0	6	6	6	7	0	0	0	0	0	0
0	0	0	5	9	1	1	0	0	5	5	5	0	9	9	0	0	0
0	5	5	7	3	6	7	0	5	7	9	0	0	2	9	8	9	0
0	3	5	1	7	8	0	0	9	8	6	Ψ	0	5	0	3	0	0
0	7	2	4	6	1	3	0	0	0	0	0	0	9	3	4	4	0
0	1	5	9	2	0	0	0	0	2	4	0	4	8	7	2	8	0
0	5	5	8	5	0	5	0	0	♣	3	0	6	3	4	8	8	0
0	4	1	5	0	0	8	3	5	7	7	0	6	9	7	8	8	0
0	7	9	7	1	0	9	0	3	2	4	0	2	8	3	0	0	0
0	0	0	0	0	0	9	2	3	3	4	0	0	0	0	0	0	0
C'EST Ψ QUI JOUE ♣: 150 ♠: 135																	

L'ORIC A L'AFFICHE





```

10 REM CHASSE AU TRESOR
20 REM -----
30 DATA 45,45,41,63,12,12,12,30,12,12,8,63,45,63,12,30,45,45,41,63,
12,12,12,30
40 FOR I=0 TO 23
50 READ A:POKE 46856+I,A
60 NEXT I
70 C$(1)=CHR$(4):C$(2)=CHR$(5):C$=CHR$(2)
80 X(1)=9:X(2)=10
90 Y(1)=6:Y(2)=5
130 PAPER 0:INK 2:CLS
140 FOR I=1 TO 10
150 FOR J=1 TO 18
160 PLOT J*2,I*2,48+INT(RND(1)*10)
170 NEXT J,I
180 PLOT 0,22,"C'EST QUI JOUE "+C$(1)+"b: "+C$(2)+"b: "
280 REM -----
290 REM LE JEU
300 FOR I=1 TO 2
310 PLOT X(1)*2-1,Y(1)*2,C$(1)+CHR$(96+I)+C$
320 PLOT X(2)*2-1,Y(2)*2,C$(2)+CHR$(97+I)+C$
325 PLOT 5,22,C$(I)+"a"+C$
330 AX=X(I):AY=Y(I)
340 GET AS:A=ASC(AS)
350 IF A=8 AND X(I)>1 THEN X(I)=X(I)-1:GOTO 400
360 IF A=9 AND X(I)<18 THEN X(I)=X(I)+1:GOTO 400
370 IF A=11 AND Y(I)>1 THEN Y(I)=Y(I)-1:GOTO 400
380 IF A=10 AND Y(I)<10 THEN Y(I)=Y(I)+1:GOTO 400
390 GOTO 340
400 PLOT AX*2-1,AY*2,C$+"0"
410 T(I)=T(I)+SCRN(X(I)*2,Y(I)*2)-48
420 PLOT 12+I*8,22,STR$(T(I))
430 IF T(I)>=250 THEN 500
440 NEXT I
450 GOTO 300
480 REM -----
490 REM RESULTATS
500 CLS
510 IF T(1)>T(2) THEN I=1 ELSE I=2
520 PLOT 15,9,CHR$(10)+C$(1)+CHR$(96+I)+": "+STR$(T(1))
530 PLOT 15,10,CHR$(10)+C$(1)+CHR$(96+I)+": "+STR$(T(1))
540 PLOT 15,13,CHR$(10)+C$(2)+CHR$(97+I)+": "+STR$(T(2))
550 PLOT 15,14,CHR$(10)+C$(2)+CHR$(97+I)+": "+STR$(T(2))
560 PLOT 11,18,"C'EST "+C$(I)+"a"+C$+"QUI GAGNE"
570 GET AS
580 RUN

```



L'ORIC A L'AFFICHE

LE PROGRAMME

Lignes 30 à 180 : initialisation

Après l'initialisation des caractères graphiques, on place dans les tableaux **X()** et **Y()** les coordonnées de départ des deux joueurs et, dans **C\$()**, les codes des couleurs.

On affiche à l'écran, par deux boucles imbriquées, une grille de nombres aléatoires de 18 sur 18.

Lignes 300 à 450 : le jeu

Après l'affichage des figures, on modifie les coordonnées du joueur en cours suivant la valeur du caractère saisi au clavier.

Puis on ajoute au total des points du joueur le nombre sur lequel on vient de se placer.

On récupère cette valeur en effectuant un **SCRN(,)** sur l'écran. Le total est affiché au bas de l'écran.

On vérifie, après chaque tour, si le joueur ne dépasse pas les 250 points requis pour gagner (tableau **T()**).

Lignes 500 à 580 : fin de partie

Un des joueurs ayant dépassé 250 points, on affiche à l'écran les résultats de la partie en précisant le gagnant.

L'ORIC A L'AFFICHE

0861/9

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

TOUR DE HANOÏ

Face à ces tours de Hanoï, vous devrez faire preuve de stratégie, de réflexion et de beaucoup de patience.

Il s'agit de faire passer, sur la tour de droite et dans le même ordre, un certain nombre de disques empilés sur la tour de gauche, en vous aidant de la tour du milieu.

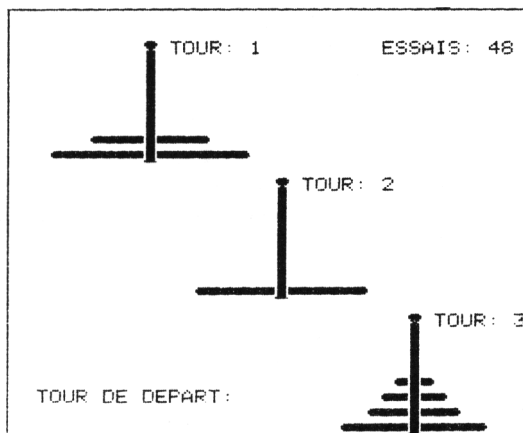
Vous pouvez choisir le nombre de disques à manoeuvrer entre trois et sept.

L'ORIC demande, en bas de l'écran, la tour de départ du mouvement, puis la tour d'arrivée : entrer 1, 2 ou 3, numéros des tours de gauche à droite.

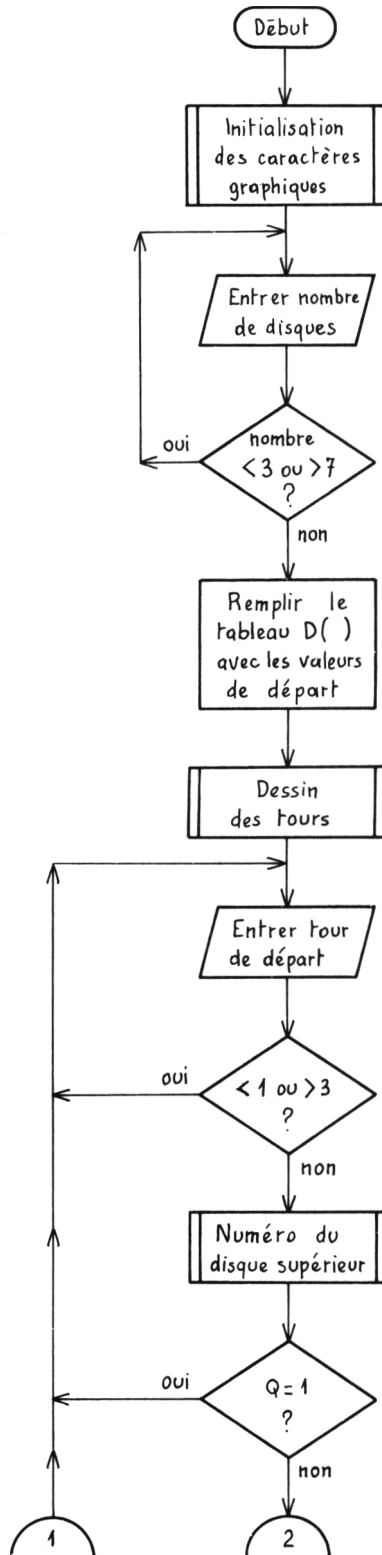
Il refuse les ordres impossibles à exécuter : on ne peut poser un disque sur un autre moins large. Par contre, un disque peut aller de la tour 1 à la tour 3, puis revenir à la tour 2.

Il affiche aussi, tout au long du jeu, le nombre d'essais.

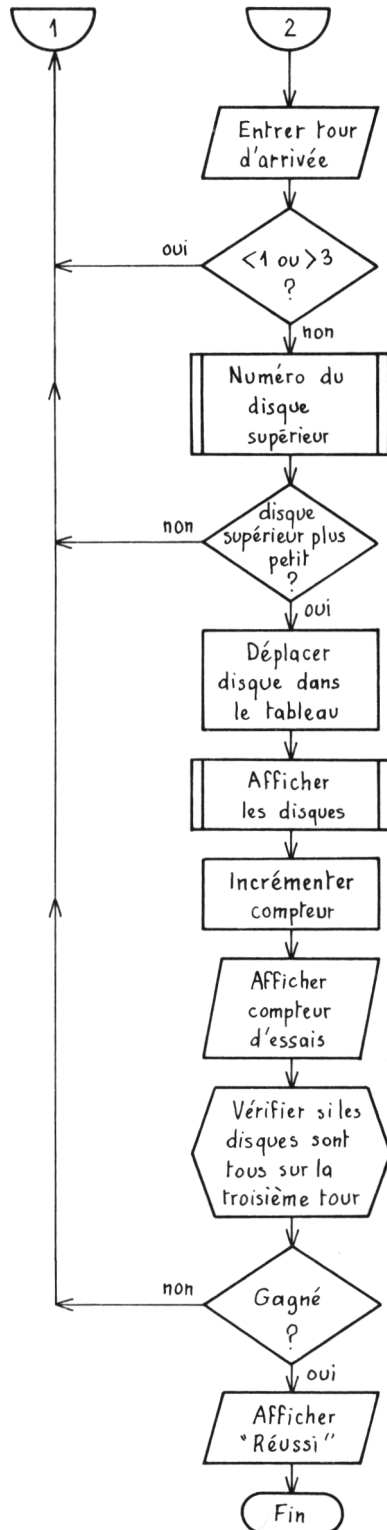
Petit conseil : il vaut mieux, pour vos nerfs, commencer par trois disques !



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE



```

10 REM TOURS DE HANOI
20 REM -----
30 GOSUB 2000
40 INK 1:PAPER 0:CLS
50 INPUT "NOMBRE DE DISQUES ";N
60 IF N<3 OR N>7 THEN 40
70 CLS
80 FOR I=1 TO 7
90 IF I<=N THEN D(1,I)=8-I
100 NEXT I
110 GOSUB 1500
120 REM -----
130 REM LE JEU
200 PLOT 1,24,CHR$(4)+"TOUR DE DEPART: "+CHR$(1)
210 GET TD:IF TD<1 OR TD>3 THEN 210
220 J=TD:GOSUB 800:IF Q=1 THEN 200
230 DD=1
240 PLOT 1,24,CHR$(4)+"TOUR D'ARRIVEE: "+CHR$(1)
250 GET TA:IF TA<1 OR TA>3 THEN 250
260 IF TA=TD THEN 200
270 J=TA:GOSUB 800:IF Q=1 THEN 290
280 IF D(TD,DD)>D(TA,I) THEN 200
290 DA=I+1
300 D(TA,DA)=D(TD,DD):D(TD,DD)=0
310 I=TD:J=DD:GOSUB 1000
320 I=TA:J=DA:GOSUB 1000
330 E=E+1
340 PLOT 27,1,CHR$(6)+"ESSAIS: "+STR$(E)
350 FOR I=1 TO N
360 IF D(3,I)=0 THEN 200
370 NEXT I
380 PLOT 2,24,"REUSSI EN"+STR$(E)+CHR$(1)+"COUPS"
390 GET A$
400 RUN
410 REM -----
420 REM NUMERO DISQUE SUPERIEUR
800 FOR I=7 TO 1 STEP -1
810 IF D(J,I)<>0 THEN Q=0:GOTO 830
820 NEXT I:Q=1
830 RETURN
980 REM -----
990 REM AFFICHE LES DISQUES
1000 FOR K=7 TO 1 STEP-1
1010 IF D(I,J)>K THEN PLOT I*10-K,I*9-J,"e":GOTO 1030
1020 IF D(I,J)=K THEN PLOT I*10-K,I*9-J,"d" ELSE PLOT I*10-K,I*9-J,
" "
1030 NEXT K
1040 FOR K=1 TO 7
1050 IF D(I,J)=K THEN PLOT K+(I)*10,I*9-J,"c":GOTO 1070
1060 IF D(I,J)>K THEN PLOT K+I*10,I*9-J,"e" ELSE PLOT K+I*10,I*9-J,
" "
1070 NEXT K
1080 RETURN
1480 REM -----
1490 REM DESSIN DU JEU
1500 FOR I=1 TO 3
1510 PLOT I*10,(I-1)*9+1,"a"+CHR$(5)+"TOUR: "+STR$(I)
1520 FOR J=1 TO 7
1530 GOSUB 1000
1540 IF J>1 THEN PLOT I*10,I*9-J,"b" ELSE PLOT I*10,I*9-J,"f"
1550 NEXT J
1560 NEXT I
1570 RETURN
1980 REM -----
1990 REM GRAPHIQUES
2000 DATA 30,63,63,30,12,30,30,30,30,30,30,30,30,30,30
2010 DATA 0,0,62,63,63,62,0,0,0,31,63,63,31,0,0
2020 DATA 0,0,63,63,63,63,0,0,30,30,30,30,30,30,63
2030 FOR I=0 TO 47
2040 READ A:POKE 46856+I,A
2050 NEXT I
2100 RETURN

```

L'ORIC A L'AFFICHE

LE PROGRAMME

Lignes 30 à 110 : initialisation

Après l'initialisation des caractères graphiques par le sous-programme en 2000, on saisit le nombre de disques à jouer entre trois et sept.

On remplit le tableau de départ en mettant tous les disques sur la première tour, le disque le plus grand étant placé à la position 1. Puis l'on fait appel au sous-programme en 1500 pour le dessin du jeu.

Lignes 200 à 340 : déplacement des disques

On commence le jeu en saisissant le numéro de la tour de départ (on déplace, dans ce cas, le disque supérieur).

Le sous-programme en 800 trouve le numéro de ce disque. Si cette tour ne comporte aucun disque, on met le drapeau à 1 (variable Q). Au retour du sous-programme, si Q=1, on revient à la saisie de la tour de départ.

On utilise le même principe pour la tour d'arrivée. Au retour du sous-programme en 800, on teste si le disque du dessus n'est pas plus petit. Si c'est le cas, on revient à la saisie de la tour de départ.

Lignes 350 à 400 : fin du jeu

On teste sur la troisième tour si tous les disques sont déjà placés. Si ce n'est pas le cas, on retourne à la ligne 200 pour le dessin des disques. Dans le cas contraire, on affiche le message "REUSSI EN X COUPS".

Lignes 1000 à 1080 : dessin des disques

Ces lignes dessinent sur l'écran, avec des caractères semi-graphiques, le disque pointé par les variables I et J dans le tableau D().

Lignes 1500 à 1570 : dessin des tours

Cette portion de programme affiche à l'écran le jeu de départ en utilisant le sous-programme en 1000.

L'ORIC A L'AFFICHE

6/1980

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

TAQUIN

Le taquin ou puzzle à 15 a été inventé par Sam Lloyd aux Etats-Unis, dans les années 1880. Il se compose d'un plateau et de quinze plaquettes numérotées de 1 à 15, que l'on dispose en désordre (plus de vingt mille milliards de possibilités). Le vide, sur le plateau, permet de bouger les plaquettes pour les remettre dans leur ordre croissant.

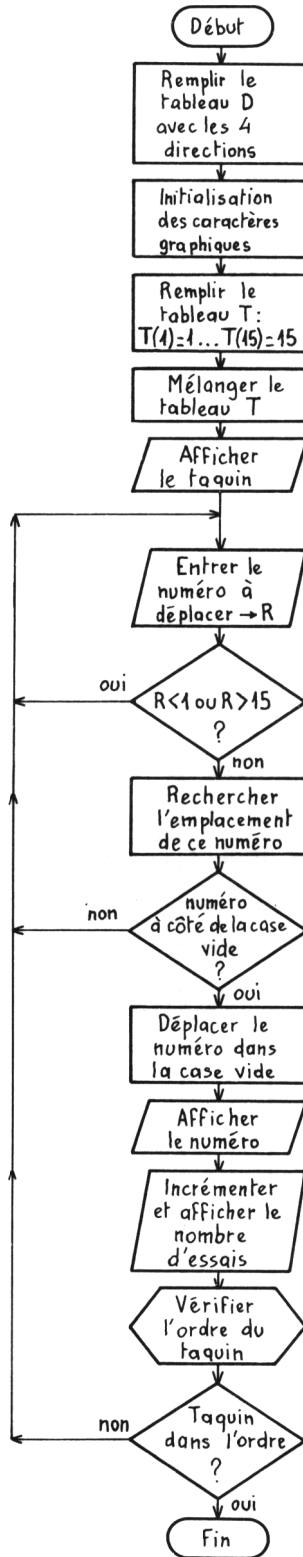
Lancez le programme en tapant **RUN**. Après quelques instants, le temps de mélanger le taquin, le plateau s'affiche à l'écran.

Seules les plaquettes voisines de la case vide peuvent se déplacer.

Il ne s'agit pas du jeu original de Sam Lloyd. En conséquence, toutes les positions de départ permettent une reconstitution de l'ensemble du jeu.

TAQUIN			
1	2	3	4
5	6	7	8
10	13	9	
12	14	15	11
QUEL NUMERO ? 9			
ESSAIS: 23			

L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE

8	1	4
5	7	6
2		3

8	1	4
5		6
2	7	3

```

10 REM TAQUIN
20 REM -----
30 DIM T(16)
40 D(1)=-4:D(2)=4:D(3)=1:D(4)=-1
50 DATA 0,0,0,0,0,0,63,1,1,1,1,1,1,1,63,1,1,1,1,1,1
60 FOR I=0 TO 23
70 READ A:POKE 46856+I,A
80 NEXT I
90 INK 0:PAPER 1:CLS:PRINT
100 PRINT TAB(27);CHR$(4);CHR$(27);"JTAQUIN";CHR$(4):PRINT
180 REM -----
190 REM MELANGE DU TAQUIN
200 FOR I=1 TO 15:T(I)=I:NEXT I
210 V=16:E=0
220 FOR I=1 TO 128
230 A=INT(RND(1)*4)+1
240 IF V+D(A)<1 OR V+D(A)>16 THEN 230
250 IF INT(V/4)=V/4 AND A=3 THEN 230
260 IF INT(V/4)=(V-1)/4 AND A=4 THEN 230
270 T(V)=T(V+D(A))
280 T(V+D(A))=0
290 V=V+D(A)
300 NEXT I
380 REM -----
390 REM DESSIN DU TAQUIN
400 PRINT"          aaaaaaaaaaaaaaaaaa"
410 FOR I=1 TO 4
420 PRINT"      c      c      c      c      c"
430 PRINT"      c      c      c      c      c"
440 PRINT"      caaaabaaaabaaaabaaaab"
450 NEXT I
460 FOR I=0 TO 15
470 IF T(I+1)=0 THEN 510
480 IF T(I+1)<10 THEN 500
490 PLOT9+5*(I-INT(I/4)*4),5+INT(I/4)*3,STR$(T(I+1))+CHR$(0):GOTO 5
500 PLOT 10+5*(I-INT(I/4)*4),5+INT(I/4)*3,STR$(T(I+1))+CHR$(0)
510 NEXT I
580 REM -----
590 REM LE JEU
600 RS="":PLOT 10,20,CHR$(3)+"QUEL NUMERO ?      "
610 GET AS:IF AS=CHR$(13) THEN 630
620 RS=RS+AS:PLOT 25+LEN(RS),20,AS:GOTO 610
630 R=VAL(RS)
640 IF R<1 OR R>15 THEN 600
650 FOR I=1 TO 16
660 IF R=T(I) THEN 680
670 NEXT I
680 R=I
690 FOR I=1 TO 4
700 IF R+D(I)>16 OR R+D(I)<1 THEN 720
710 IF T(R+D(I))=0 THEN 740
720 NEXT I
730 GOTO 600
740 T(R+D(I))=T(R)
750 T(R)=0
760 R=R-1
800 PLOT9+5*(R-INT(R/4)*4),5+INT(R/4)*3,"      "
810 J=R+D(I)
820 IF T(J+1)<10 THEN 840
830 PLOT9+5*(J-INT(J/4)*4),5+INT(J/4)*3,STR$(T(J+1))+CHR$(0):GOTO 8
840 PLOT 10+5*(J-INT(J/4)*4),5+INT(J/4)*3,STR$(T(J+1))+CHR$(0)
850 F=F+1:PLOT 13,24,"ESSAIS:"+STR$(F)
880 REM -----
890 REM FIN DU JEU ?
900 FOR I=1 TO 15
910 IF T(I)<>I THEN 600
920 NEXT I
930 PLOT 9,20,"O.K VOUS AVEZ TROUVE"
940 GET AS
950 GOTO 90

```

8	1	4
5	6	
2	7	3

0	-	
4	9	5
4	2	4

LE PROGRAMME

Lignes 30 à 300 : initialisation et mélange

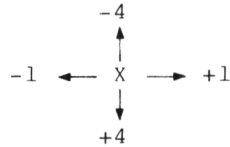
Le tableau T correspond aux seize cases du taquin. On y place donc les numéros de 1 à 15 (ligne 200).

Le tableau D correspond au déplacement des plaquettes.

Tableau T()

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	/

Tableau D()



On mélange les numéros en permutant deux cases désignées par un nombre aléatoire et le pointeur de la case vide (variable V).

Cent vingt huit permutations sont nécessaires pour mélanger correctement le taquin.

Lignes 400 à 510 : dessin du taquin

On affiche le jeu à l'écran dans le nouvel ordre (ou plutôt désordre !). Les chiffres sont placés sur l'écran directement par des plots (lignes 460 à 510).

Lignes 600 à 850 : le jeu

On déplace le numéro tapé par le joueur en vérifiant qu'il n'est pas inférieur à 1 ou supérieur à 15 et qu'il est bien placé à côté de la case vide. Le déplacement est calculé avec le tableau D. Comme pour le dessin du jeu, ces nombres sont affichés par des plots.

Lignes 900 à 950 : fin du jeu

A chaque essai, on vérifie qu'il reste des plaquettes dans le désordre. Si c'est le cas, on retourne à la ligne 600 pour un nouvel essai.

L'ORIC A L'AFFICHE

[illegible]

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

AYEZ DU CARACTÈRE

Ce programme permet d'obtenir les valeurs décimales d'une figure dessinée sur l'écran pour les utiliser par la suite dans vos programmes, à la place du premier ou du deuxième jeu de caractères.

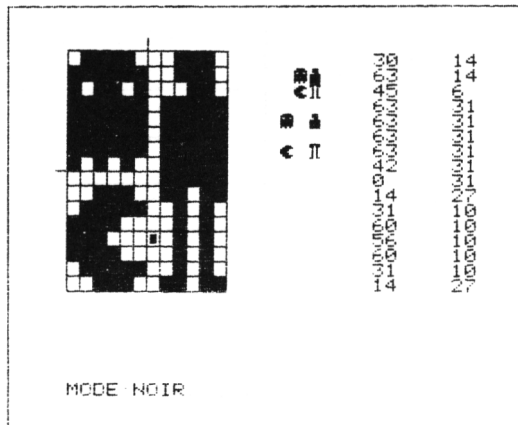
En appuyant sur la touche **M**, vous pouvez changer de mode et obtenir les fonctions suivantes :

- noir = écrire en noir ;
- blanc = écrire en blanc (c'est-à-dire le fond de l'écran) ;
- déplacement = rien (permet de se positionner sur l'écran sans changer la valeur de ce qui y est déjà).

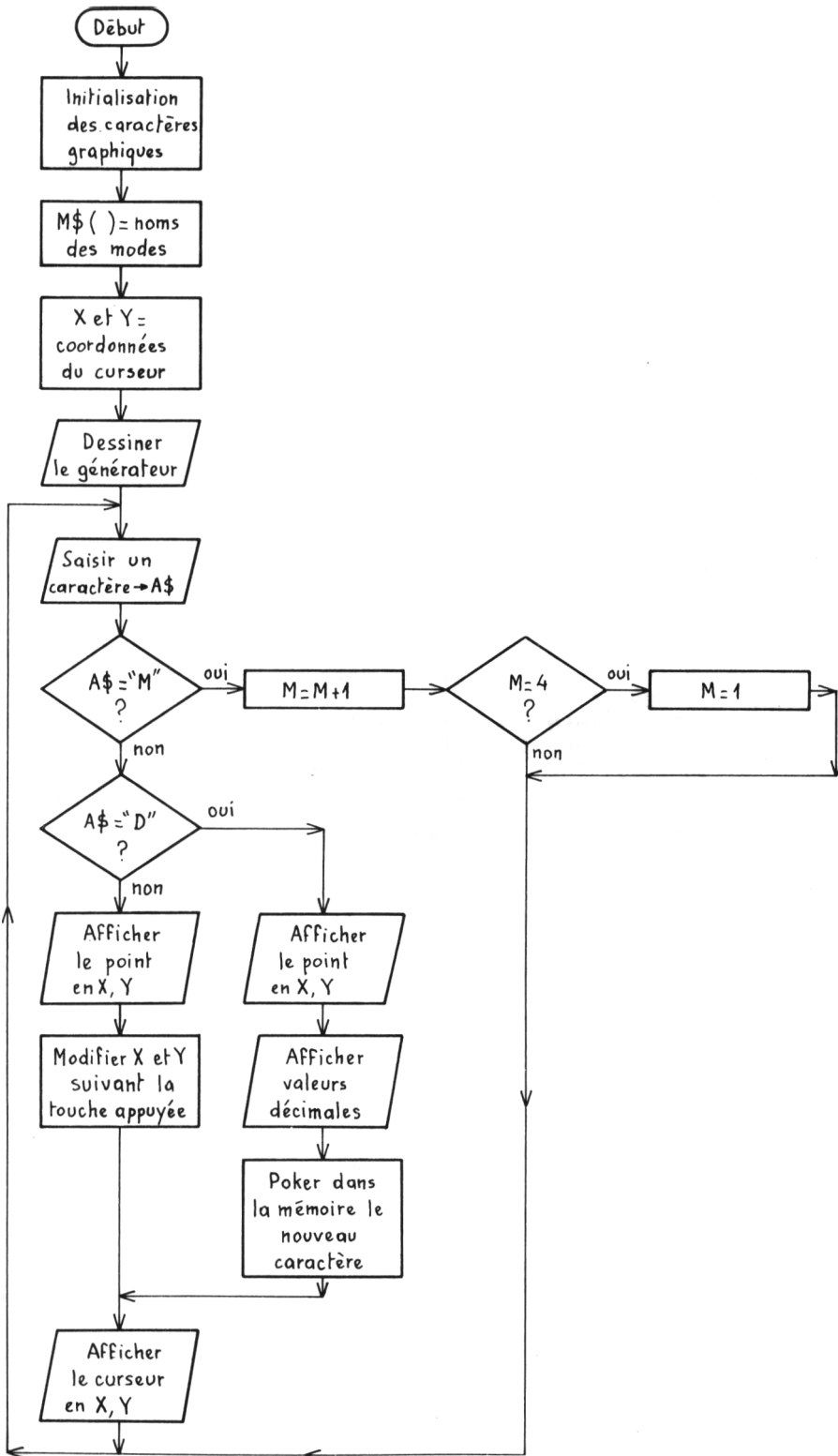
La touche **D** entraîne le calcul des valeurs décimales de la figure (elles s'affichent à droite de l'écran) et les caractères dans leur taille normale d'utilisation.

Pour se déplacer sur la grille, il faut utiliser les quatre flèches du clavier.

Ceci permet de créer rapidement quatre caractères graphiques pour vos programmes en n'utilisant ni papier ni crayon !



L'ORIC A L'AFFICHE





```

10 REM AYEZ DU CARACTERE
20 REM -----
30 DATA 63,63,63,63,63,63,63,63,1,1,1,1,1,1,1,63,0,0,0,0,0,0,0,1
40 DATA 1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,63,1,29,29,29,29,29,1,63
60 FOR I=0 TO 47
70 READ A:POKE 46856+I,A
80 NEXT I
90 M$(1)="DEPLACEMENT":M$(2)="NOIR"           ":M$(3)="BLANC"
100 C=98:X=7:Y=10
180 REM -----
190 REM DESSIN GENERATEUR
200 CLS:INK 0:PAPER 1:PRINT:PRINT
210 PRINT "ceeeeebeeeeee"
220 FOR I=1 TO 16
230 IF I=8 THEN PRINT"bbbbbbbbbbbbbb":GOTO 250
240 PRINT"dbbbbbbbbbbbbbbb"
250 NEXT I
260 PLOT X,Y,"f"
270 PLOT 19,4,"gi":PLOT 19,5,"hj"
280 PLOT 18,7,"g i":PLOT 18,9,"h j"
300 GOTO 1000
380 REM -----
390 REM DEPLACEMENTS
400 GET A$:IF A$="" THEN 400
410 IF A$="M" THEN 1000
420 IF A$="D" THEN PLOT X,Y,C:GOTO 2000
430 PLOT X,Y,CHR$(C)
440 IF A$=CHR$(8) THEN X=X-1:IF X=1 THEN X=13
450 IF A$=CHR$(9) THEN X=X+1:IF X=14 THEN X=2
460 IF A$=CHR$(11) THEN Y=Y-1:IF Y=2 THEN Y=18
470 IF A$=CHR$(10) THEN Y=Y+1:IF Y=19 THEN Y=3
480 IF M=2 THEN C=97:GOTO 500
490 IF M=3 THEN C=98 ELSE C=SCRN(X,Y)
500 PLOT X,Y,"f"
510 GOTO 400
980 REM -----
990 REM CHANGEMENT DE MODE
1000 M=M+1:IF M=4 THEN M=1
1010 PLOT 2,25,"MODE:"+M$(M)
1020 GOTO 400
1980 REM -----
1990 REM VALEURS DECIMALES
2000 FOR I=3 TO 18
2010 PLOT 25,I," "
2020 NEXT I
2030 FOR I=3 TO 18
2040 FOR J=1 TO 7 STEP 6
2050 D=0
2060 FOR K=5 TO 0 STEP -1
2070 IF SCRN(J+6-K,I)=97 THEN D=D+2^K
2080 NEXT K
2090 PLOT 23+J,I,STR$(D)
2100 IF J=1 THEN POKE 46901+I,D:GOTO 2110
2105 POKE 46917+I,D
2110 NEXT J:NEXT I
2120 GOTO 500

```



LE PROGRAMME

Lignes 30 à 300 : initialisation

Après l'initialisation des caractères graphiques, on place dans le tableau **M\$** les noms des trois modes (noir, blanc et déplacement).

Les variables **X** et **Y** donnent les coordonnées du point de départ.

Les lignes 200 à 300 affichent à l'écran, en caractères graphiques, le dessin de l'utilitaire, puis l'on saute directement à la ligne 1000 pour initialiser le mode (déplacement pour le début).

Lignes 400 à 510 : déplacement du curseur

Cette partie du programme teste le clavier et déplace le curseur sur l'écran en laissant derrière lui une trace de couleur blanche ou noire, suivant la fonction en cours, ou remplace les anciennes couleurs s'il s'agit du mode déplacement.

La valeur 97 indique un point noir et la valeur 98 un point blanc.

Lignes 1000 à 1020 : changement de mode

On incrémente simplement la variable **M** pour des valeurs allant de 1 à 3.

Lignes 2000 à 2120 : décimal

On calcule, à partir de l'écran (testé par **SCRN(,)**), les valeurs décimales des quatre caractères graphiques.

Ces valeurs, placées dans la variable **D**, sont affichées à l'écran puis stockées dans la mémoire par l'instruction **POKE** dans le premier jeu de caractères.

L'ORIC A L'AFFICHE

0861/9

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

CHAPITRE 3

JEUX DE HASARD

JACKPOT

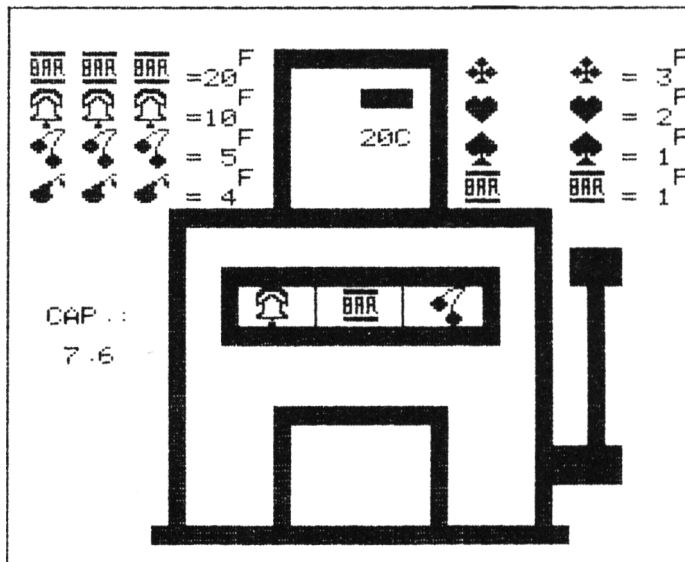
Voilà un jeu qui peut paraître tout d'abord passif puisque le joueur intervient très peu. Mais attention pour le joueur acharné : il risque d'en perdre le sommeil !

Après avoir tapé RUN, le jack-pot se dessine avec les combinaisons gagnantes.

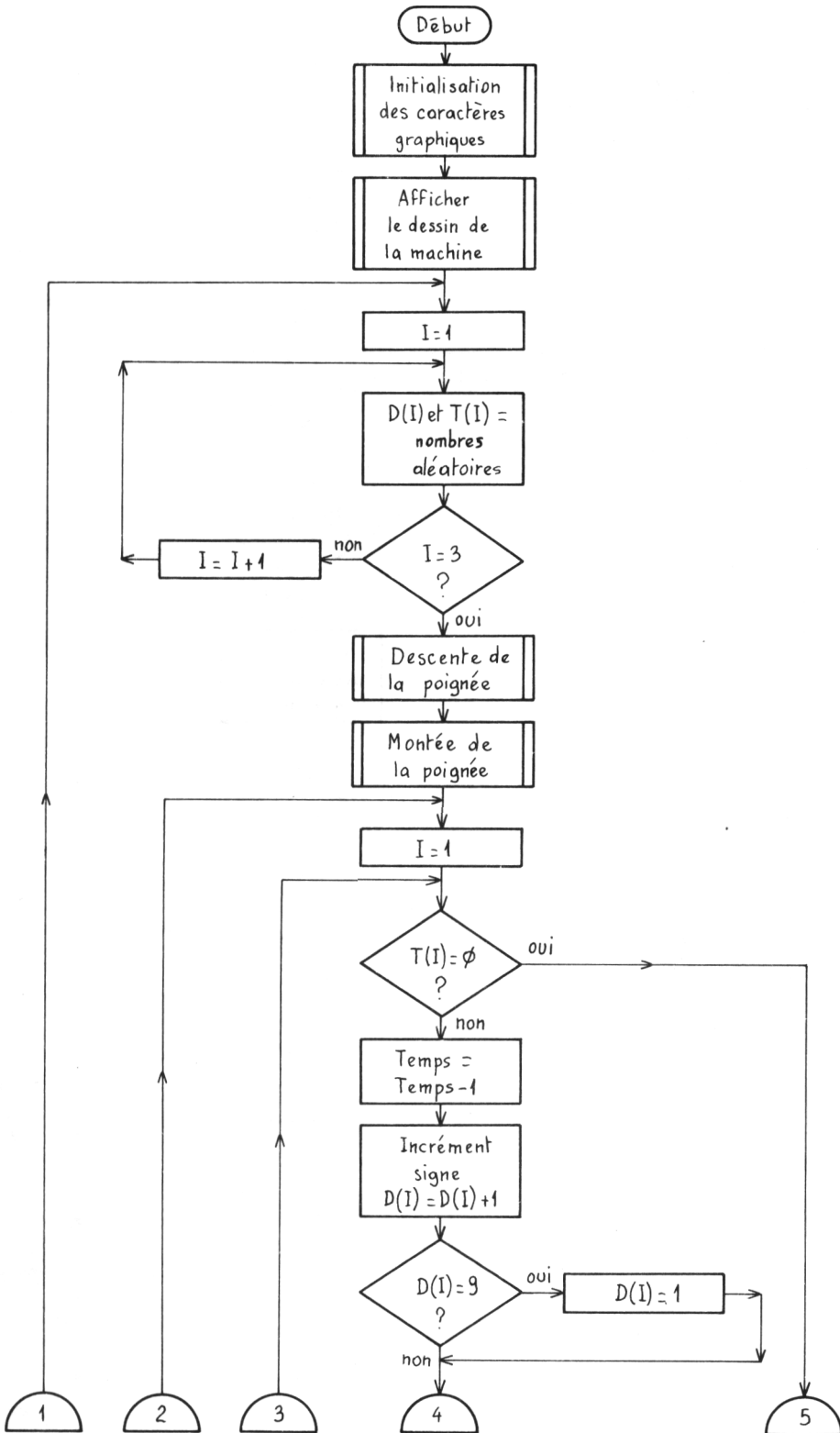
Vous appuyez sur n'importe quelle touche et c'est parti. Quand les symboles d'immobilisent, si vous n'avez rien gagné, votre capital se décrémente de 0,20 Franc. Il suffit d'appuyer à nouveau sur n'importe quelle touche pour relancer le jeu.

Quand une combinaison est gagnante, votre gain s'affiche en clignotant en bas du jack-pot et votre capital augmente d'autant.

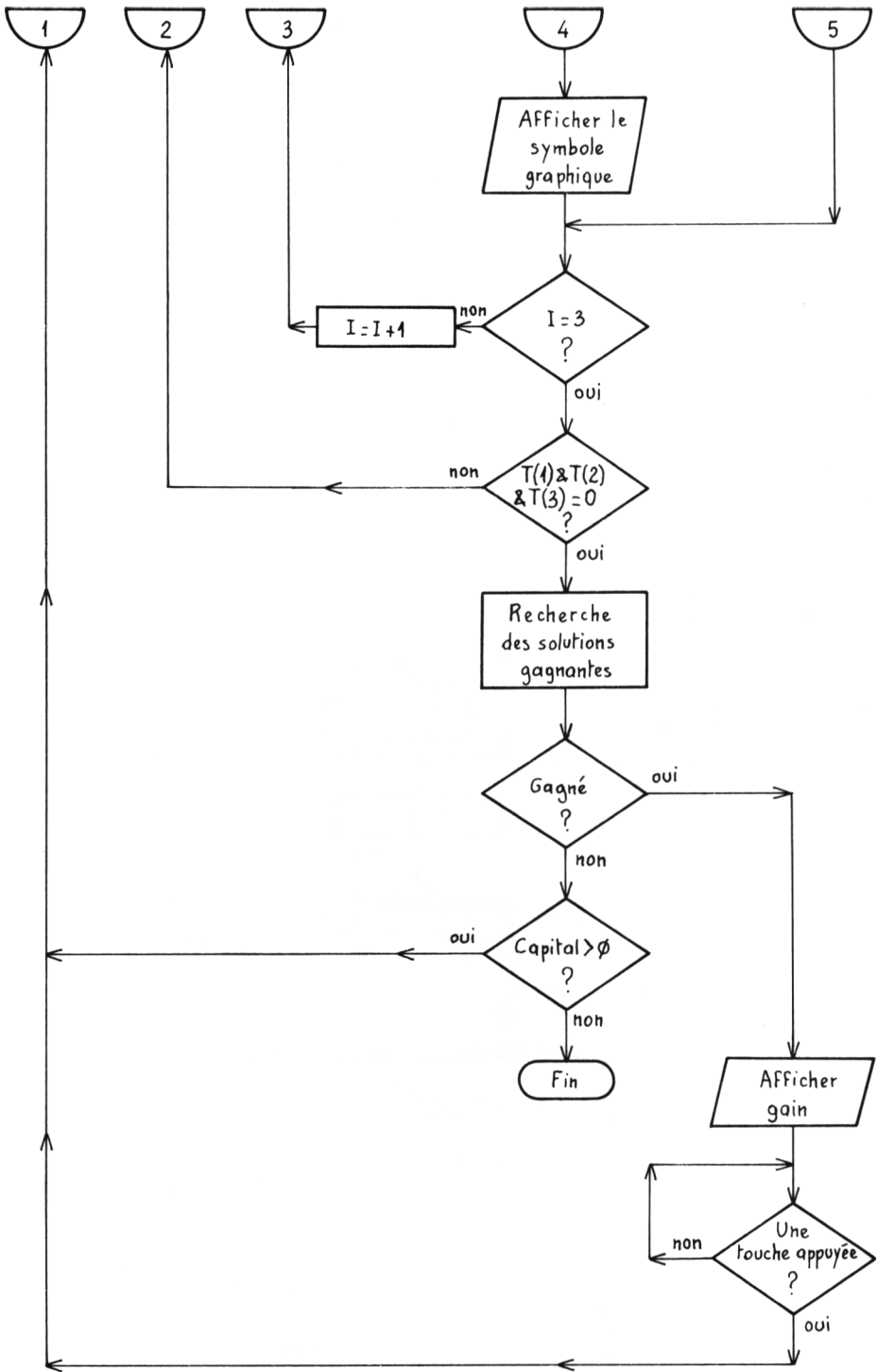
Patience, patience, ce jack-pot, digne de ses ancêtres, peut vous prendre beaucoup de temps ...



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE



```

10 REM JACK-POT
20 REM -----
30 C$:PAPER 0:INK 7
40 GOSUB 2000
50 C$=CHR$(96):GOSUB 3000
60 C=10
180 REM -----
190 REM LE JEU
200 FOR I=1 TO 3
210 D(I)=INT(RND(1)*8)+1
220 T(I)=INT(RND(1)*10)+5
230 NEXT I
240 GOSUB 1000:GOSUB 1500
250 FOR I=1 TO 3
260 IF T(I)=0 THEN 320
270 T(I)=T(I)-1
280 D(I)=D(I)+INT(RND(1)*2)+1:IF D(I)>=9 THEN D(I)=1
290 Z=INT(RND(1)*7)+1
300 PLOT I*5+8,13,CHR$(Z)+CHR$(87+D(I)*4)+CHR$(88+D(I)*4)+CHR$(7)
310 PLOT I*5+8,14,CHR$(Z)+CHR$(89+D(I)*4)+CHR$(90+D(I)*4)+CHR$(7)
320 NEXT
330 IF T(1) OR T(2) OR T(3) THEN 250
380 REM -----
390 REM GAINS ?
400 C=INT((C-0.195)*10)/10:CAP=C
410 D=D(1)*100+D(2)*10+D(3)
420 D1=D(1)*100+D(3)
430 IF D=888 THEN C=C+20
440 IF D=777 THEN C=C+10
450 IF D=555 THEN C=C+5
460 IF D=666 THEN C=C+4
470 IF D1=303 THEN C=C+3
480 IF D1=202 THEN C=C+2
490 IF D1=404 THEN C=C+1
500 IF D1=808 THEN C=C+1
510 PLOT 2,16,+STR$(C)+CHR$(7)+" "
520 IF C>CAP THEN 600
530 ZAP:GET AS
540 IF C=0 THEN 800
550 GOTO 200
580 REM -----
590 REM GAGNE
600 PLOT 17,21,CHR$(12)+CHR$(1)+MID$(STR$(C-CAP),2)+CHR$(7)+CHR$(8)
610 PING:GET AS
620 PLOT 17,21," "
630 GOTO 200
780 REM -----
790 REM PERDU
800 PLOT 17,21,"MERCI":PLOT 16,22,"POUR LES":PLOT 17,23,"10 FR$"
810 EXPLODE:GET AS
820 PLOT 17,21," ":PLOT 16,22," ":PLOT 17,23," "
830 GOTO 60
980 REM -----
990 REM DESC POIGNEE
1000 FOR I=11 TO 17
1010 PLOT 32,I," ":PLOT 32,I+1,"<<<":PLOT 32,I+2,"<<<"
1020 NEXT I
1030 RETURN
1480 REM -----
1490 REM MONT. POIGNEE
1500 FOR I=18 TO 11 STEP -1
1510 PLOT 32,I+2,"< ":PLOT 32,I+1,"<<<":PLOT 32,I,"<<<"
1520 NEXT I

```


LE PROGRAMME

Lignes 30 à 60 : initialisation

Après la définition des caractères graphiques (sous-programme en 2000) et le dessin de la machine (sous-programme en 3000), le programme initialise à 10 Francs le capital (variable C).

Lignes 200 à 330 : le jeu

Le jeu a besoin des deux tableaux suivants :

- D() donne le symbole de départ à afficher. Les symboles suivants apparaissent dans l'ordre défini dans le sous-programme en 2000.
- T() donne le temps, c'est-à-dire le nombre de symboles qui vont défiler sur la machine.

Dès que ces deux tableaux sont définis, on manoeuvre la poignée en utilisant les sous-programmes en 1000 et 1500 (respectivement descente et montée), puis on affiche les 3 symboles à partir de D() pour un temps T().

Lignes 400 à 550 : gains ?

- On commence par décrémenter le capital C de 20 centimes ; puis on place dans CAP le capital restant.
- En additionnant D(1) à D(3), on obtient un chiffre qui sert de comparaison avec les solutions gagnantes (variable D). La variable D1 ne prend en considération que les deux extrémités de la combinaison.
- On compare D et D1 aux solutions gagnantes et on incrémente le capital (variable C).

Quand C est supérieur à CAP, on gagne quelque chose. On va alors à la ligne 600 pour afficher le montant des gains ; sinon on retourne à un nouveau jeu.

Lignes 600 à 630 : gains !

On affiche en clignotant le gain tant que l'on n'appuie pas sur une touche.

L'ORIC A L'AFFICHE

[illegible]

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

BACCARA

Si vous rêvez de l'ambiance feutrée mais exaltante des casinos, entrez dans le jeu et osez manipuler des sommes fabuleuses.

Un crédit de 10 000 Francs vous est attribué au début de la partie. La banque - votre adversaire - tire quatre cartes d'un sabot composé de 8 jeux de cinquante deux cartes. Toutes les cartes conservent leurs valeurs nominales de l'as (1) au neuf ; les figures et le dix ne valent rien. Pour obtenir la valeur totale d'un jeu, on additionne les valeurs de chaque carte en ne retenant que le chiffre des unités.

A chaque donne, l'ordinateur affiche les cartes ; de même, tout au long du jeu, il indique en bas de l'écran les mouvements en cours ainsi que votre mise et vos gains.

La banque donne tout d'abord au ponté - c'est vous - deux des quatre cartes et garde les deux autres pour elle.

Vous regardez vos cartes et vous vous trouvez alors devant quatre possibilités pour parler :

- vous avez 8 ou 9, c'est-à-dire une main naturelle : c'est au banquier de parler ;
- vous avez 6 ou 7 : vous ne prenez pas d'autre carte ;
- vous avez 5 : vous choisissez ou non de prendre une autre carte ;
- vous avez entre 0 et 4 : vous prenez obligatoirement une autre carte.

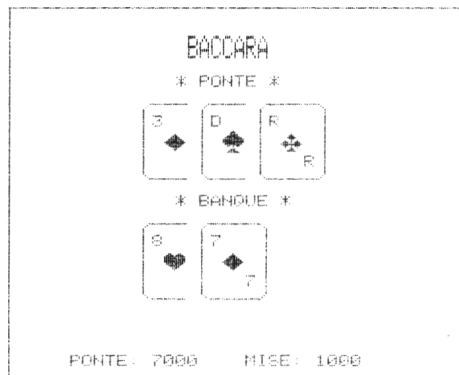
Seule la 3ème carte que vous aurez pu tirer est visible du banquier.

Puis le banquier parle. Si vous avez 8 ou 9, il faut comparer les jeux. Sinon le banquier a le choix de prendre ou non une carte (choix déterminé par un tableau dans le programme).

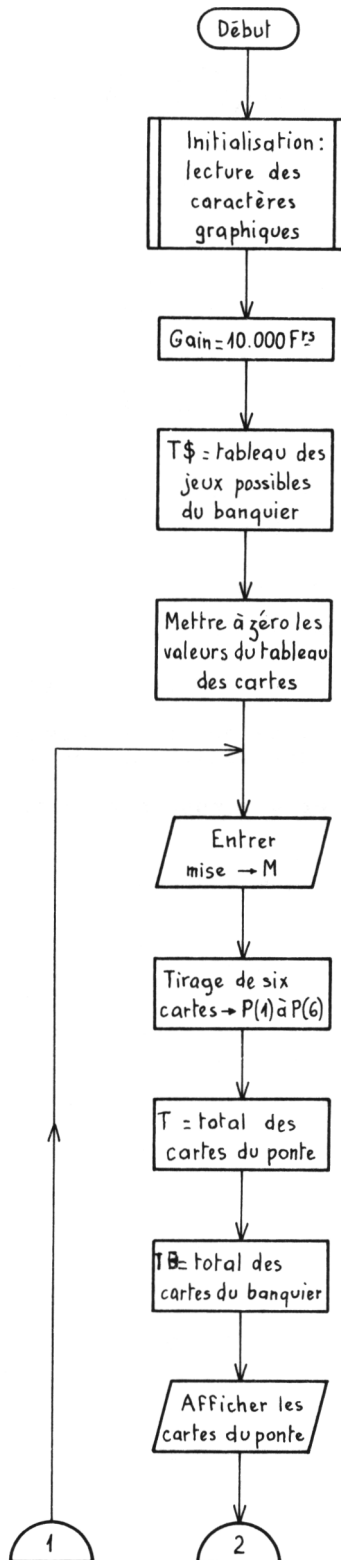
A ce moment là, toutes les cartes sont retournées ; le joueur le plus près de 9 gagne la partie.

En tapant sur n'importe quelle touche vous obtenez vos gains et vous pouvez miser pour un autre tour.

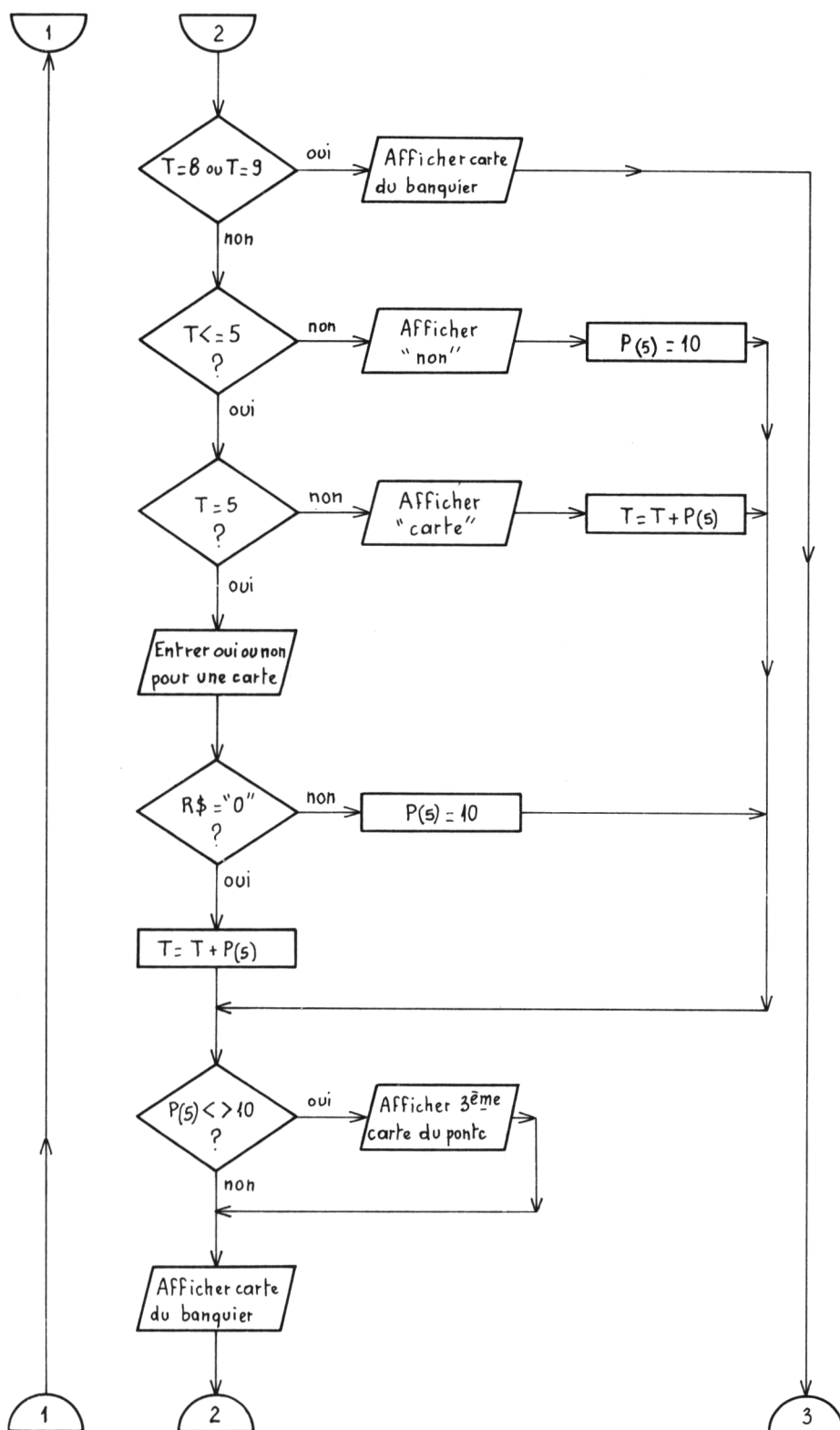
Le jeu reprend jusqu'à disparition de tout votre capital ...



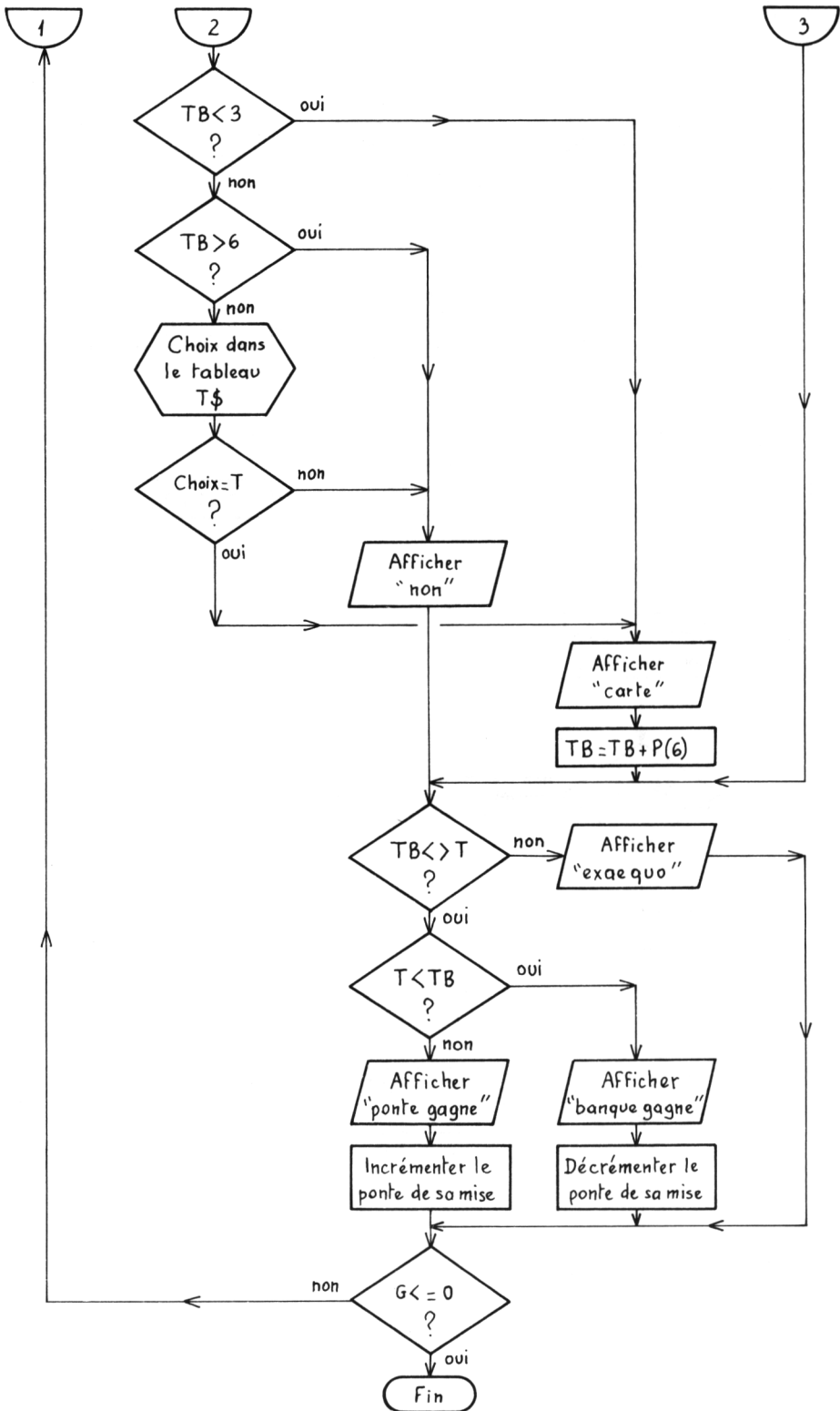
L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE



L'ORIC A L'AFFICHE

```

10 REM BACCARA
20 REM -----
30 GOSUB 2000:CLS
50 PRINT:PRINT TAB(26)+CHR$(4)+CHR$(27)+"JBACCARA"+CHR$(4)
200 JS="123456789XVDR"
210 G=10000
220 DIM J(13,4)
230 TS(3)="TTTTTTTTRTT"
240 TS(4)="RRTTTTTTRRT"
250 TS(5)="RRRRTTTTTRT"
260 TS(6)="RRRRRTTTRR"
270 INK 1:PAPER 4
280 PLOT 10,10,"JE MELANGE LES CARTES"
290 FOR I=1 TO 13
300 FOR K=1 TO 4
310 J(I,K)=0
320 NEXT K
330 NEXT I
340 NC=0
350 REM -----
360 REM MISE DU PONTE
370 CLS
380 PRINT "MISE (MAXI ";G;" FRS) ";
390 INPUT M
400 IF (M>G)+(M<=0) THEN 370
410 REM -----
420 REM TIRAGE DE 6 CARTES
430 FOR I=1 TO 6
440 P(I)=0
450 F=INT(RND(1)*13)+1
460 C=INT(RND(1)*4)+1
470 IF J(F,C)>=8 THEN 450
480 J(F,C)=J(F,C)+1
490 IF F>9 THEN 510
500 P(I)=F
510 S(I,1)=F
520 S(I,2)=C
530 NEXT I
540 T=P(1)+P(3)
550 IF T<10 THEN 570
560 T=T-10
570 TB=P(2)+P(4)
580 IF TB<10 THEN 620
590 TB=TB-10
600 REM -----
610 REM AFFICHAGE DES CARTES
620 CLS
630 PRINT:PRINT TAB(26)+CHR$(4)+CHR$(27)+"JBACCARA"+CHR$(4)
640 PLOT 14,4,"* PONTE *"
650 PLOT 14,13,"* BANQUE *"
660 PLOT 5,25,"PONTE:"+STR$(G)+CHR$(5)+" MISE:"+STR$(M)
670 X=10:Y=6:I=1:GOSUB 1600
680 X=15:I=3:GOSUB 1600
690 N=0
700 IF I<>9 AND T<>8 THEN 800
710 N=1:GOTO 1060
780 REM -----
790 REM CHOIX DU PONTE
800 IF T<=5 THEN 850
810 P(5)=10
820 MS="*PONTE NON*"
830 GOSUB 1540

```

L'ORIC A L'AFFICHE

```

840 GOTO 1020
850 IF T=5 THEN 920
860 M$="*PONTE CARTE*"
870 GOSUB 1540
880 T=T+P(5)
890 IF T<10 THEN 980
900 T=T-10
910 GOTO 980
920 PLOT 12,23,CHR$(7)+"CARTE (O/N) ?"
940 GET A$
950 IF A$="O" THEN 880
970 P(5)=10
980 PLOT 10,23,"
1000 REM -----
1010 REM CHOIX DE LA BANQUE
1020 IF P(5)=10 THEN 1060
1030 X=20:I=5:GOSUB 1600
1060 X=10:Y=15:I=2:GOSUB 1600
1100 X=15:I=4:GOSUB 1600
1130 IF N=1 THEN 1300
1140 IF TB<3 THEN 1200
1150 IF TB>6 THEN 1170
1160 IF MID$(TS(TB),P(5)+1,1)="T" THEN 1200
1170 M$="*BANQUE NON*"
1180 GOSUB 1540
1190 GOTO 1300
1200 M$="*BANQUE CARTE*"
1210 GOSUB 1540
1220 X=20:I=6:GOSUB 1600
1250 TB=TB+P(6)
1260 IF TB<10 THEN 1300
1270 TB=TB-10
1280 REM -----
1290 REM RESULTATS
1300 IF T<>TB THEN 1340
1310 M$="*EXAEQUO*"
1320 GOSUB 1540
1330 GOTO 1420
1340 IF T>TB THEN 1390
1350 M$="*BANQUE GAGNE*"
1360 GOSUB 1540
1370 G=G-M
1380 GOTO 1420
1390 M$="*PONTE GAGNE"+STR$(M)+CHR$(5)+"FRS*"
1400 GOSUB 1540
1410 G=G+M
1420 NC=NC+1
1430 GET A$
1440 IF G<=0 THEN 1470
1450 IF NC>=204 THEN 270
1460 GOTO 370
1470 CLS
1480 PRINT"MERCI POUR LES DIX MILLE FRANCS":PRINT
1490 PRINT"LE CASINO ACCEPTE LES CHEQUES"
1500 END
1520 REM -----
1530 REM MESSAGE
1540 PLOT 17-LEN(M$)/2,23,CHR$(3)+CHR$(12)+M$+CHR$(8)
1550 PING:WAIT 500
1560 PLOT 0,23,"
1570 RETURN
1580 REM -----

```

L'ORIC A L'AFFICHE

```

1590 REM AFFICHE LES CARTES
1600 FOR J=0 TO 7
1610 PLOT X,J+Y,LS(J+1)
1620 NEXT J
1630 K=93+S(I,2)*4
1640 PLOT X+2,Y+1,MID$(JS,S(I,1),1)
1650 PLOT X+5,Y+4,MID$(JS,S(I,1),1)
1660 IF S(I,2)>2 THEN Z=0 ELSE Z=1
1670 PLOT X+2,Y+2,CHR$(Z)+CHR$(K)+CHR$(K+1)
1680 PLOT X+2,Y+3,CHR$(Z)+CHR$(K+2)+CHR$(K+3)
1690 RETURN
1980 REM -----
1990 REM GRAPHIQUES
2000 FOR I=0 TO 191
2010 READ A:POKE 46856+I,A
2020 NEXT I
2030 LS(1)=CHR$(7)+"quuuus":LS(2)=CHR$(7)+"w      "+CHR$(7)+"x"
2040 LS(3)=LS(2):LS(4)=LS(2):LS(5)=LS(2)
2050 LS(6)=CHR$(7)+"rvvvvt"
2060 RETURN
2100 DATA 0,0,0,1,3,7,15,31,0,0,0,0,32,48,56,60
2110 DATA 63,31,15,7,3,1,0,0,62,60,56,48,32,0,0,0
2120 DATA 0,0,0,12,30,63,63,63,0,0,0,24,60,62,62,62
2130 DATA 63,31,15,7,3,1,0,0,62,60,56,48,32,0,0,0
2140 DATA 0,0,0,1,3,7,3,9,0,0,0,0,32,48,32,8
2150 DATA 29,63,29,9,3,7,0,0,28,62,28,8,32,48,0,0
2160 DATA 0,1,3,7,15,31,63,63,0,0,32,48,56,60,62,62
2170 DATA 63,31,13,1,3,7,15,0,62,60,24,0,32,48,56,0
2180 DATA 0,3,4,8,16,16,16,16,16,16,16,16,16,8,4,3,0
2190 DATA 0,48,8,4,2,2,2,2,2,2,2,2,4,8,48,0
2200 DATA 0,63,0,0,0,0,0,0,0,0,0,0,0,63,0
2210 DATA 16,16,16,16,16,16,16,16,2,2,2,2,2,2,2,2

```


LE PROGRAMME

Lignes 30 à 340 : initialisation

Après la définition des caractères graphiques des quatre couleurs (sous-programme en 2000), on place dans le tableau T\$ les décisions que prendra le banquier. Chaque liste du tableau contient onze informations. Les dix premières correspondent à la valeur de la carte tirée par le ponté (entre 0 et 9). La onzième précise que le ponté n'a pas pris de carte.

La lettre T conseille au banquier de tirer, alors que R l'incite à rester. Chaque ligne correspond au jeu du banquier. Seules sont représentées les solutions pour les valeurs comprises entre 3 et 6 puisque le banquier tire toujours une carte quand il a moins de 3 points et refuse quand il en a plus de 6.

Le tableau J correspond au sabot. Il est remis à zéro quand les cartes sont toutes tirées.

Lignes 370 à 400 : mise au ponté

On saisit la mise du ponté à condition qu'elle ne soit pas supérieure au capital et qu'elle ne soit pas nulle.

Lignes 430 à 590 : tirage des cartes

On tire 6 cartes dans le sabot, F correspondant à la figure de la carte et C à la couleur. On incrémente le tableau J de 1 ; celui-ci ne doit jamais dépasser 8 puisqu'il n'y a que 8 jeux de 52 cartes.

Lignes 800 à 980 : choix du ponté

Après l'affichage des cartes du ponté, celui-ci peut choisir une carte uniquement si le total de ses points est égal à 5.

Dans les autres cas, on suit les règles exposées dans l'introduction de ce chapitre.

Lignes 1020 à 1270 : choix du banquier

Le banquier tire ou non une carte suivant les données du tableau T\$ (voir les explications des lignes 30 à 340).

Lignes 1300 à 1500 : résultat et fin

Pour les résultats, il suffit de comparer les jeux du ponté et du banquier (variable T et TB). On les affiche à l'écran ainsi que les gains du ponté. Si ce dernier a toujours de l'argent, on continue la partie.

Dans le cas où le nombre de cartes déjà tirées dépasse 204, on remet le tableau J à zéro avant une nouvelle partie.

L'ORIC A L'AFFICHE

DATE		NOM DU PROGRAMME : BACCARA				PAGE	
1/1							
IDENTIFICATION des VARIABLES							
NOM SYMBOLIQUE	TYPE (*)	ORIGINE				LONG.	OBSERVATIONS
		Clavier	Programme	Cassette	Disquette		
A, \$, _ , _	Variable	X					Choix du pont
B, _ , _ , _	Drapeau	X					Drapeau dans l'instruction CALL KEY
C, _ , _ , _	Variable	X					Nombre aléatoire donnant la couleur
F, _ , _ , _	Variable	X					Nombre aléatoire donnant la figure
G, _ , _ , _	Compteur	X					Gains du pont
I, _ , _ , _	Indice	X					Boucle FOR/NEXT
J, _ , _ , _	Indice	X					Boucle FOR/NEXT
J, \$, _ , _	Constante	X					Liste des cartes
J, (, _ , _)	Tableau	X					Cartes du jeu : 13 cartes de 4 couleurs
K, _ , _ , _	Indice	X					Boucle FOR/NEXT
L, \$, (, _)	Tableau	X					Dessin des cartes
M, _ , _ , _	Variable	X					Mise du pont
M, \$, _ , _	Variable	X					Message à afficher
N, _ , _ , _	Aiguillage	X					N=1 si main naturelle
N, C, _ , _	Compteur	X					Nombre de cartes utilisées
P, (, _ , _)	Tableau	X					Valeur des cartes de la partie
S, (, _ , _)	Tableau	X					Cartes en jeu
T, _ , _ , _	Compteur	X					Total des cartes du pont
T, \$, (, _)	Tableau	X					Liste des choix du banquier : T : conseille de tirer 1 carte R : incite à rester à 2 cartes
T, B, _ , _	Compteur	X					Total des cartes du banquier
X, _ , _ , _	Variable	X					} Coordonnées des cartes sur l'écran
Y, _ , _ , _	Variable	X					

(*) Légende : aiguillage, compteur, constante, index, indice, tableau, variable. Reproduction interdite (c) Editions du P.S.I.

Achévé d'imprimer en février 1984
sur les presses de l'imprimerie Laballery et C^{ie}
58500 Clamecy
Dépôt légal : janvier 1984

N° d'impression : 401031
N° d'édition : 86595-130-1
ISBN : 2-86595-130-8



90 FF

**JEAN-FRANÇOIS
SEHAN**

L'ORIC A L'AFFICHE

Possesseurs de l'ORIC, ce livre s'adresse à vous.
Jean-François Sehan vous propose une sélection de
20 programmes de jeux d'adresse, de réflexion et de hasard,
utilisant les possibilités graphiques et sonores de votre ORIC
(slalom, missionnaires et cannibales, baccara...).

Dans un but pédagogique, chaque programme
est accompagné d'un organigramme, d'une liste de variables
et d'une explication de chaque ligne Basic
pour l'adaptation éventuelle à d'autres ordinateurs.

Editions du P.S.I.

Boîte Postale 86

F-77402 Lagny/Marne cedex

ISBN 2.86595.130.8 - Imprimé en France

JEAN-FRANÇOIS SEHAN
L'ORIC A L'AFFICHE

PROGRAMMES

