

JEUX SUR VG 5000 PHILIPS

B. Amsler, O. Villemaud



Edimicro

**JEUX GRAPHIQUES
SUR VG 5000
PHILIPS**

Collection animée par Benoît de MERLY.

© F.D.S./Edimicro 1984
Première édition

Imprimé en France, Droits mondiaux réservés.

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

ISBN : 2-904457-26-7

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

JEUX GRAPHIQUES SUR VG 5000 PHILIPS

par
Benoît AMSLER
Olivier VILLEMAUD

Collection « Ordinateurs personnels »

 **Edimicro**

THE
SCHOOL
OF
THE

SOMMAIRE

1. TECHNIQUES DE PROGRAMMATION DES JEUX	11
1.1 Qu'est-ce qu'un jeu ?	11
1.1.1 Les différents types de jeux	11
1.1.2 Les jeux de réflexion	11
1.1.3 Les jeux d'action	12
1.1.4 Qualités d'un bon jeu	12
1.2 Comment programmer un jeu ?	13
1.2.1 Introduction	13
1.2.2 Qualités d'un bon programme de jeu	14
1.2.3 Mise au point	21
1.2.4 Programmation des jeux de réflexion	21
1.2.5 Programmation des jeux d'action	22
1.2.6 Utilisation de manches à balai	24
1.3 Exemple de jeu d'action : le Squash	25
1.3.1 Présentation	25
1.3.2 Organigramme et structure du programme	27
1.3.3 Liste des instructions	30
1.4 Exemple de jeu d'action : Bombardier	33
1.4.1 Présentation	33
1.4.2 Organigramme et structure du programme	33
1.4.3 Liste des instructions	38
1.5 Optimisation de la vitesse d'un jeu	41
2. JEUX D'ACTION À UN JOUEUR	43
2.1 Stock-car	43
2.1.1 Présentation	43
2.1.2 Organigramme et structure du programme	44
2.1.3 Liste des instructions	48

2.2	La chenille infernale	53
2.2.1	Présentation	53
2.2.2	Organigramme et structure du programme	53
2.2.3	Liste des instructions	57
2.3	Envahisseurs	61
2.3.1	Présentation	61
2.3.2	Organigramme et structure du programme	61
2.3.3	Liste des instructions	65
2.4	Survie	70
2.4.1	Présentation	70
2.4.2	Organigramme et structure du programme	70
2.4.3	Liste des instructions	74
2.5	Collimateur	79
2.5.1	Présentation	79
2.5.2	Organigramme et structure du programme	79
2.5.3	Liste des instructions	82
2.6	Labyrinthe	86
2.6.1	Présentation	86
2.6.2	Organigramme et structure du programme	86
2.6.3	Liste des instructions	90
3.	JEUX D'ACTION À DEUX JOUEURS	97
3.1	Face-à-face	97
3.1.1	Présentation	97
3.1.2	Organigramme et structure du programme	98
3.1.3	Liste des instructions	100
3.2	Pilotes de course	103
3.2.1	Présentation	103
3.2.2	Organigramme et structure du programme	103
3.2.3	Liste des instructions	107
3.3	Bataille d'hélicoptères	112
3.3.1	Présentation	112
3.3.2	Organigramme et structure du programme	112
3.3.3	Liste des instructions	115

4. JEUX DE RÉFLEXION À UN JOUEUR	119
4.1 Radar	119
4.1.1 Présentation	119
4.1.2 Organigramme et structure du programme	120
4.1.3 Liste des instructions	122
4.2 Taquin	125
4.2.1 Présentation	125
4.2.2 Organigramme et structure du programme	125
4.2.3 Liste des instructions	129
4.3 Puzzle	133
4.3.1 Présentation	133
4.3.2 Organigramme et structure du programme	133
4.3.3 Liste des instructions	136
4.4 Cavalier	140
4.4.1 Présentation	140
4.4.2 Organigramme et structure du programme	140
4.4.3 Liste des instructions	143
4.5 Tic-tac-toe	146
4.5.1 Présentation	146
4.5.2 Organigramme et structure du programme	146
4.5.3 Liste des instructions	149
4.6 Bataille navale	154
4.6.1 Présentation	154
4.6.2 Organigramme et structure du programme	154
4.6.3 Liste des instructions	157
4.7 Karo	164
4.7.1 Présentation	164
4.7.2 Organigramme et structure du programme	165
4.7.3 Liste des instructions	167
4.8 Poker	173
4.8.1 Présentation	173
4.8.2 Organigramme et structure du programme	174
4.8.3 Liste des instructions	177

5. JEUX DE RÉFLEXION À DEUX JOUEURS	185
5.1 Carré	185
5.1.1 Présentation	185
5.1.2 Organigramme et structure du programme	185
5.1.3 Liste des instructions	187
5.2 Réactions en chaîne	190
5.2.1 Présentation	190
5.2.2 Organigramme et structure du programme	191
5.2.3 Liste des instructions	193
6. JEUX D'AVENTURE : FANTÔME	199
6.1 Présentation	199
6.2 Organigramme et structure du programme	201
6.3 Liste des instructions	206

Techniques de programmation des jeux

1.1 QU'EST-CE QU'UN JEU ?

1.1.1 Les différents types de jeux

Ce livre présente de nombreux types de jeux où les joueurs vont tirer sur des pipes, lancer des missiles, piloter des engins de combat, trouver un trésor caché, réordonner un carré de lettres... Pour répondre aux goûts et à la curiosité du lecteur, nous proposons des jeux de hasard, d'action, d'adresse et de réflexion.

1.1.2 Les jeux de réflexion

Qu'il s'agisse du jeu de Taquin ou du jeu de Tic-tac-toe, l'utilisateur sera amené à jouer seul contre l'ordinateur, ou avec celui-ci comme partenaire. Il pourra prendre le temps qu'il juge nécessaire à sa réflexion.

1.1.3 Les jeux d'action

Beaucoup plus démonstratifs, les jeux d'action supposent normalement un consensus quasi-général entre les joueurs. Ils font appel à l'habileté et aux réflexes des joueurs, face à divers monstres et autres phénomènes sortis de l'imagination du programmeur.

1.1.4 Qualités d'un bon jeu

Il est fort difficile de définir exactement ce qu'est un bon jeu. Certaines personnes peuvent trouver un jeu génial alors que d'autres ne le regarderont même pas. La longueur du jeu et la complexité de l'algorithme ne rentrent pas forcément en ligne de compte. Bien qu'il n'existe pas de loi, il est cependant possible d'énoncer certains critères sans lesquels un jeu ne peut prétendre recueillir l'enthousiasme des joueurs. Ces critères sont les suivants :

- Le jeu doit faire intervenir l'habileté, les réflexes (action), l'"intelligence" (réflexion), la chance ou l'intuition (aventure) ...
- Il ne doit être ni trop simple, ni trop complexe. Le joueur ne doit pas gagner ou perdre constamment.
- Il doit être adapté à l'ordinateur (pas plus de deux joueurs).
- Il doit avoir une belle présentation, utiliser au maximum les possibilités graphiques et sonores, être parfaitement au point, et protégé contre les erreurs de manipulation du joueur.
- Il doit tourner rapidement pour ne pas lasser l'attention du joueur.

1.2 COMMENT PROGRAMMER UN JEU ?

1.2.1 Introduction

Que vous ayez eu l'idée d'un jeu, ou que vous souhaitiez reprendre un jeu célèbre, vous avez intérêt à ne pas vous mettre directement devant votre ordinateur pour taper des instructions au fur et à mesure que vous réfléchissez au jeu. Dans la plupart des cas, pour un jeu suffisamment complexe, vous obtiendrez un programme illisible, rempli d'erreurs, très difficile à mettre au point et incompréhensible, même par vous.

Pour éviter ces défauts, nous vous conseillons de mettre votre jeu sur le papier, c'est-à-dire de faire son analyse :

- décrivez-le complètement, dans ses moindres détails
- écrivez les règles du jeu
- dessinez le - ou les - décors
- dessinez le - ou les - objets en mouvement, dans le cas d'un jeu d'action.

Réfléchissez à la manière dont les diverses parties des règles vont s'enchaîner et s'entrecouper. Etudiez la part du joueur et de la machine, etc... Essayez en outre d'entrevoir toutes les étapes du fonctionnement du jeu. Tout ceci forme le cahier des charges du jeu.

A la fin de cette étape, vous devez être capable de simuler complètement le travail de la machine. Tracez alors l'organigramme fonctionnel du programme. Commencez à réfléchir à la façon d'implémenter le cahier des charges en machine. En somme, réfléchissez à la programmation.

1.2.2 Qualités d'un bon programme de jeu

* Introduction

Tout programme en cours de mise au point, qu'il s'agisse ou non d'un jeu, doit avoir certaines qualités. Il doit être entre autres :

- modulaire et bien découpé,
- lisible et clairement présenté,
- structuré et programmé de façon correcte.

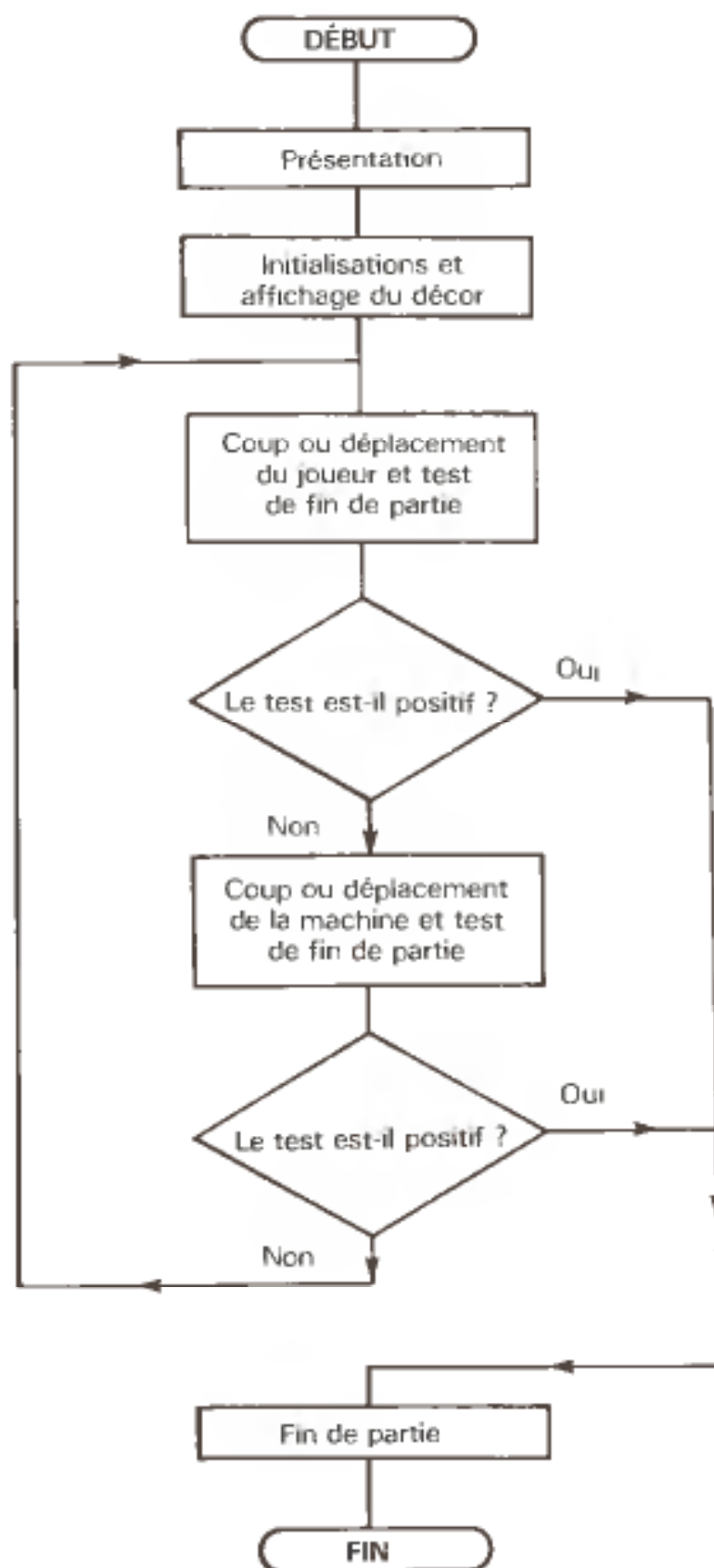
* Modularité d'un programme

Dire qu'un programme est modulaire, c'est aussi dire qu'il est découpé en unités autonomes qui ont chacune une tâche bien définie. Par exemple, pour passer d'un jeu de Squash à un jeu de Casse-briques, il suffit d'ajouter le module "gestion de briques" qui sera, en pratique, un sous-programme clairement défini. Nous pouvons donner comme autres exemples :

- l'évaluation du coup du joueur
- l'évaluation du coup de la machine
- le déplacement du mobile dirigé par le joueur
- l'apparition de nouveaux objets sur l'écran
- la présentation du jeu
- la création du décor
- le traitement d'une fin de partie.

Prévoyez pour chacun de ces modules les arguments dont ils ont besoin en entrée et les paramètres de retour.

La plupart de nos jeux adoptent les règles de numérotation suivantes :



Lignes	10 à	499	Programme principal
Lignes	500 à	2999	Sous-programmes
Lignes	5000 à	5199	Présentation
Lignes	5200 à	5999	Initialisations, tracé du décor
Lignes 10000 et suivantes			Fin de partie

Nous utiliserons par exemple très souvent la structure de programme indiquée (voir page 15)

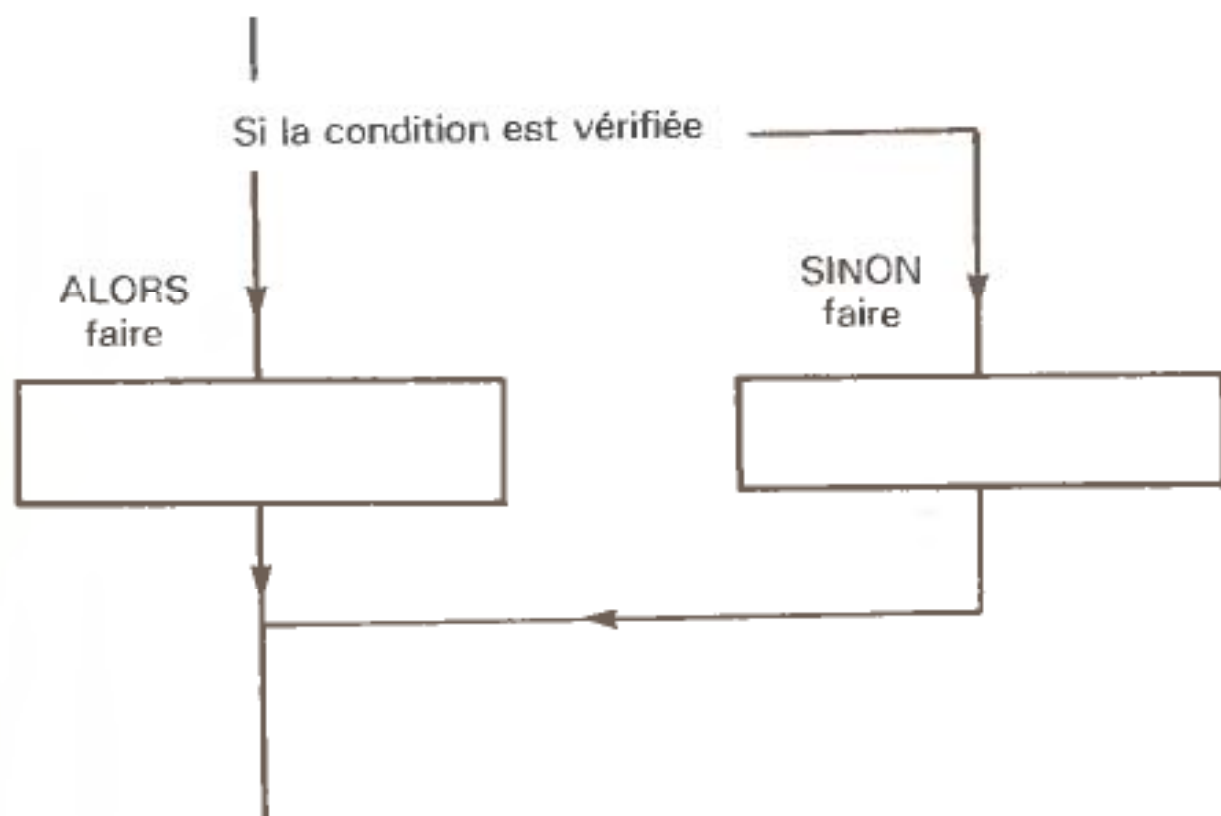
* Lisibilité d'un programme

Pour rendre un programme lisible, nous vous conseillons de mettre de nombreuses lignes de remarques ou de commentaires. Pour notre part, nous attribuons à chaque module un intervalle de numéros de lignes, et nous commencerons nos programmes en tapant d'abord les lignes de remarques qui forment alors le "squelette" du programme.

* Structuration d'un programme

Lorsque vous avez décomposé votre programme en modules, et que vous avez tapé le squelette des commentaires, vous devez vous attaquer à la programmation proprement dite. Il y a en fait deux manières de programmer : la programmation dite "spaghetti", remplie de branchements GOTO qui vont dans tous les sens (et rendent le programme incompréhensible), et la programmation structurée dans laquelle les modules sont décomposés en blocs et sous-blocs. Ces modules n'utilisent finalement que les quatre types de structure suivants:

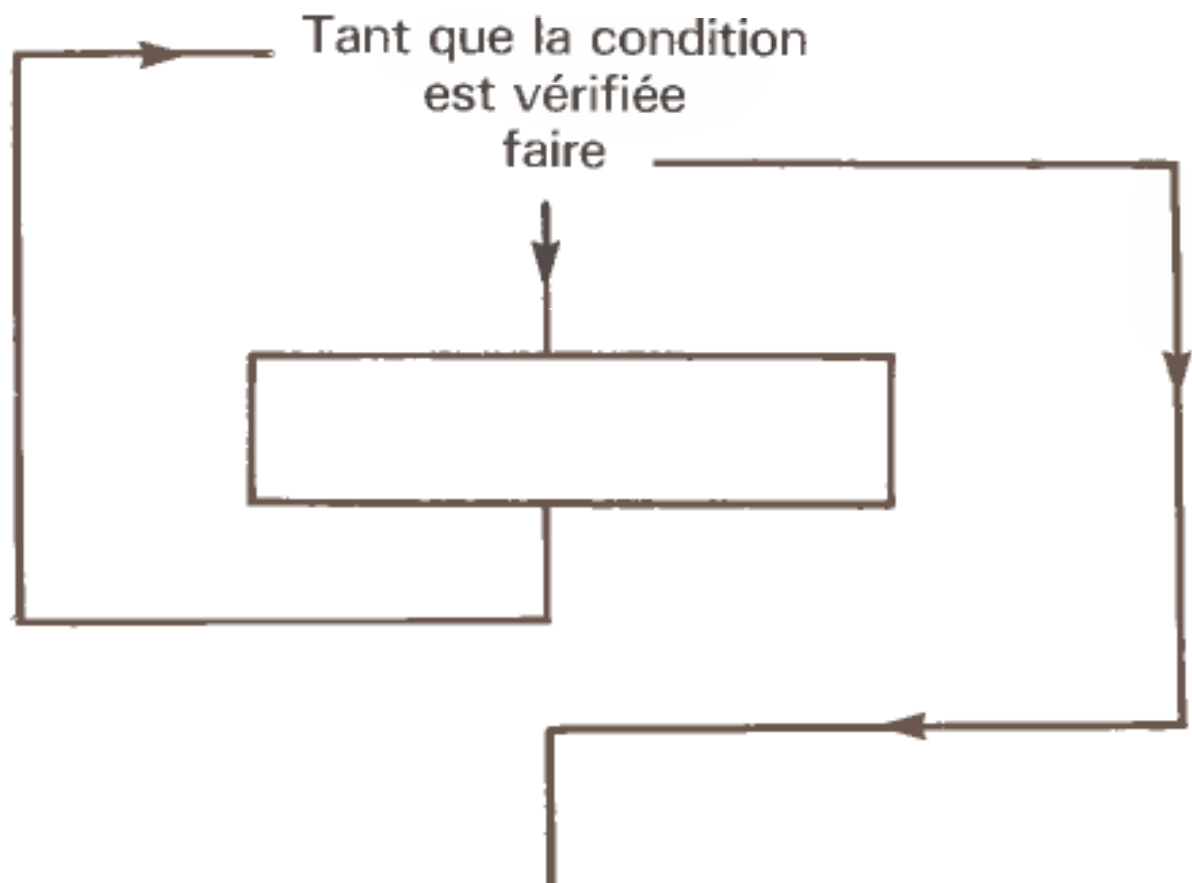
- Le test



Cette structure est réalisée en BASIC par un emploi simultané des instructions IF...THEN et GOTO de la façon suivante :

```
100 IF condition non vérifiée THEN GOTO 150
110 REM condition vérifiée
  :
  :
140 GOTO 200
150 REM condition non vérifiée
  :
  :
200 REM suite
```


- La boucle TANT...QUE



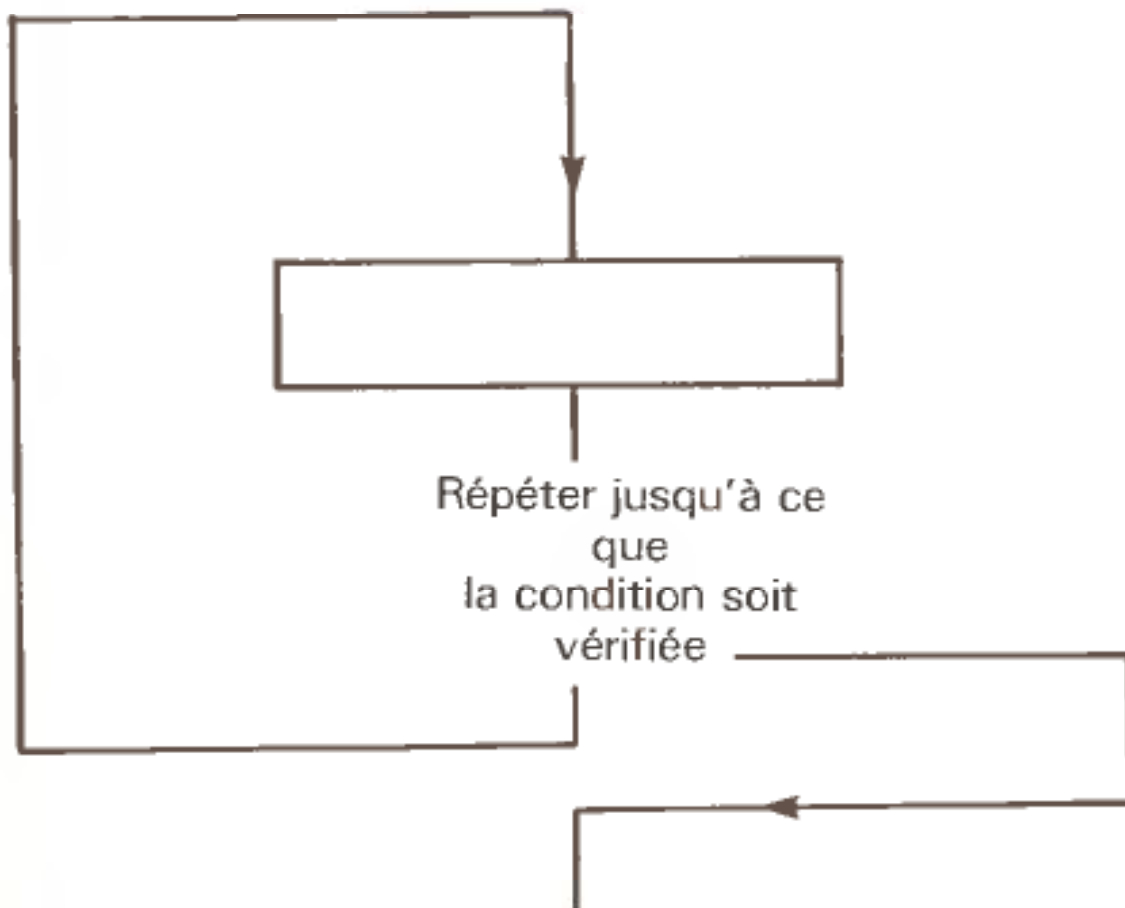
Bien qu'il n'existe pas d'instruction appropriée, cette structure peut être réalisée en BASIC par un emploi simultané des instructions IF...THEN et GOTO de la façon suivante :

```

100 IF condition non vérifiée THEN GOTO 150
110 REM condition vérifiée
    :
    :
140 GOTO 100
150 REM suite

```

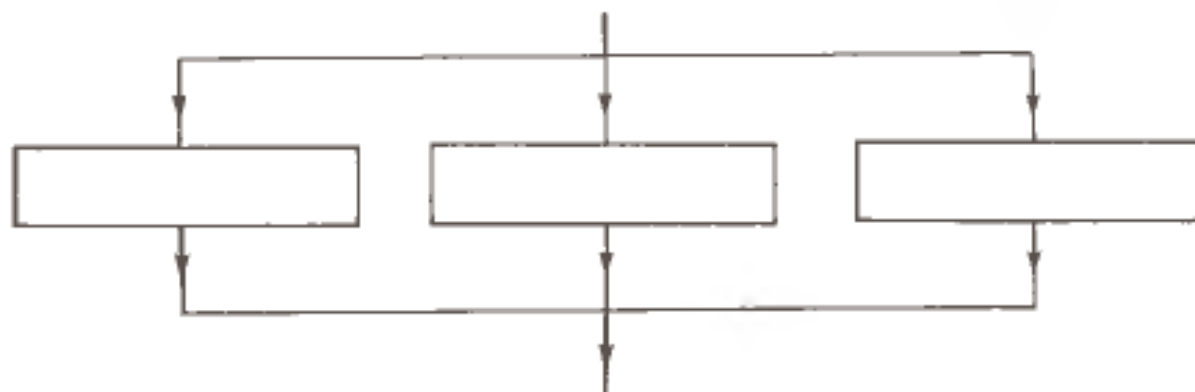
- La boucle REPETER...JUSQU'A CE QUE



Bien qu'il n'existe pas d'instruction appropriée, cette structure peut être réalisée en BASIC par un emploi simultané des instructions IF...THEN et GOTO de la façon suivante :

```
100 REM condition non vérifiée
   :
   :
   :
140 IF condition non vérifiée, THEN GOTO 100
150 REM suite car condition vérifiée
```

- Le choix multiple



Cette structure se programme en BASIC au moyen de l'instruction ON...GOTO. Nous l'utiliserons dans les fins de partie de nos jeux.

* Présentation du jeu

Une belle présentation, ainsi que l'utilisation des possibilités graphiques et sonores de la machine, rendent les jeux beaucoup plus attrayants. Il convient donc de les soigner, ce que permet le BASIC de votre VG5000. La possibilité de définir ses propres caractères est un facteur très important pour la vraisemblance des jeux : création de voitures, monstres, balles, animaux...

* Protection d'un programme

Il est bon d'insister sur le fait que tout programme doit être protégé contre les erreurs de manipulation des joueurs, qu'elles soient conscientes ou non. Toutes les manipulations au clavier - ou avec les manches à balai - peuvent conduire à "planter" le programme : en faisant sortir un mobile de l'écran, en tapant une lettre quand un chiffre est demandé, etc...

Cela provoque l'apparition sur l'écran d'un message d'erreur et vous devrez alors relancer le programme. Dans tous les cas, ces erreurs peuvent être évitées en contrôlant chaque entrée de variable et chaque déplacement de mobile : il faut protéger les programmes pour minimiser les risques d'erreurs. Bien entendu, tous ces tests de protection ralentissent les programmes, mais c'est préférable à une erreur d'exécution.

1.2.3 Mise au point

En théorie, cette étape est superflue. Mais la pratique démontre qu'une bonne analyse réduit considérablement sa longueur. Malheureusement, un long programme contient inmanquablement des erreurs de divers types. Les erreurs les plus graves sont les erreurs d'algorithme dues à une mauvaise analyse ; dans ce cas, il faut revoir la conception du jeu.

Les autres erreurs sont dues, pour la plupart, à un mauvais codage de l'organigramme. Il est alors d'autant plus facile de les trouver que le programme est modulaire, c'est-à-dire découpé en blocs. Dans ce cas, un examen attentif du module où l'erreur s'est produite doit permettre de la corriger. Dans le cas contraire, les instructions STOP et CONT permettent de stopper puis de reprendre l'exécution du programme, en laissant ainsi l'accès aux diverses variables du programme.

1.2.4 Programmation des jeux de réflexion

* Introduction

La qualité d'un jeu de réflexion dépend moins de sa vitesse d'exécution que de sa capacité à faire penser le (ou les) joueur(s). Toutefois, dire que la vitesse ne joue aucun rôle est à notre avis exagéré. Attendre plus de dix minutes la réponse de la machine est trop long, et même exaspérant !

* Organisation d'un programme de jeu de réflexion

Elle varie beaucoup selon que l'ordinateur tient le rôle d'arbitre ou celui de joueur actif.

Dans le premier cas, la machine dessinera simplement le terrain de jeu et comptera les points ; le programme ne sera pas compliqué.

Dans le second cas, il faut créer un sous-programme spécial qui fait jouer la machine. Comme il est impossible d'analyser toutes les possibilités de jeu sur tous les coups (une infinité de cas), il faut le plus souvent se limiter à un certain nombre d'étapes.

Le jeu de Tic-tac-toe que nous vous proposons au chapitre 4 (jeux de réflexion) évalue toutes les possibilités de jeu sur un coup. Cela vous laisse une possibilité de gain. Une évaluation sur davantage de coups fait en général appel à des algorithmes "d'intelligence artificielle".

1.2.5 Programmation des jeux d'action

* Introduction

La qualité d'un jeu d'action résulte de sa capacité à déplacer rapidement des mobiles sur l'écran. Pour créer l'illusion, la méthode couramment employée consiste à visualiser, à un emplacement donné, un caractère graphique (standard ou redéfini au goût du programmeur), à laisser s'écouler un délai d'attente, à supprimer le mobile de l'écran en le remplaçant par un blanc, et enfin à réafficher l'objet à un autre endroit, après calcul du déplacement.

La vitesse de déplacement découle donc de la vitesse d'affichage de l'ordinateur et de la durée d'attente entre l'effacement et le nouvel affichage.

Le problème consiste très souvent à ne pas donner l'impression d'un mouvement trop saccadé au joueur. Pour cela, le déplacement ne doit pas être exagéré ; il faut tout de même tenir compte de la vitesse de travail de la machine utilisée, et chercher le compromis entre vitesse et propreté du déplacement des objets.

De plus, un jeu d'action comporte en général plusieurs objets à déplacer, ce qui allonge la durée de l'affichage et limite la complexité des jeux, pour éviter une trop grande lenteur. Le calcul des nouvelles coordonnées des mobiles dépend en général d'éventuelles collisions, explosions, sorties de l'écran, apparitions de nouveaux objets, etc. Le programme doit donc tenir compte de tout.

* Organisation d'un programme de jeu d'action

Le programme d'un jeu d'action doit être organisé pour limiter au maximum la boucle principale, et n'en sortir que pour entreprendre une action précise. Par exemple, dans le jeu de Squash, la boucle principale déplace la balle, et la sortie de cette boucle a lieu lors de la rencontre d'un mur ou de la raquette.

Les sous-programmes souvent appelés devront être les plus courts, en reportant si nécessaire certains traitements de cas spéciaux dans des sous-programmes de second niveau.

Les portions du programme faisant la présentation du jeu, le tracé des décors et les fins de partie auront des numéros de ligne élevés.

1.2.6 Utilisation de manches à balai

Les manches à balai (joysticks en anglais) sont fréquemment utilisés pour les jeux vidéo et les jeux sur ordinateurs domestiques. En plus des quatre touches de déplacement du curseur, situées en bas et à droite du clavier - qui permettent de simuler un manche à balai - votre VG5000 vous offre la possibilité de disposer de deux joysticks, par l'intermédiaire d'une interface séparée. Dans les fonctions décrites ci-après, les joysticks seront repérés de la manière suivante :

- 0 manche à balai de gauche
- 1 manche à balai de droite
- 2 touche de déplacement du curseur

Les trois fonctions suivantes permettent de connaître l'état des manches à balai. Nous avons précisé entre parenthèses la correspondance entre joysticks et touches de déplacement du curseur.

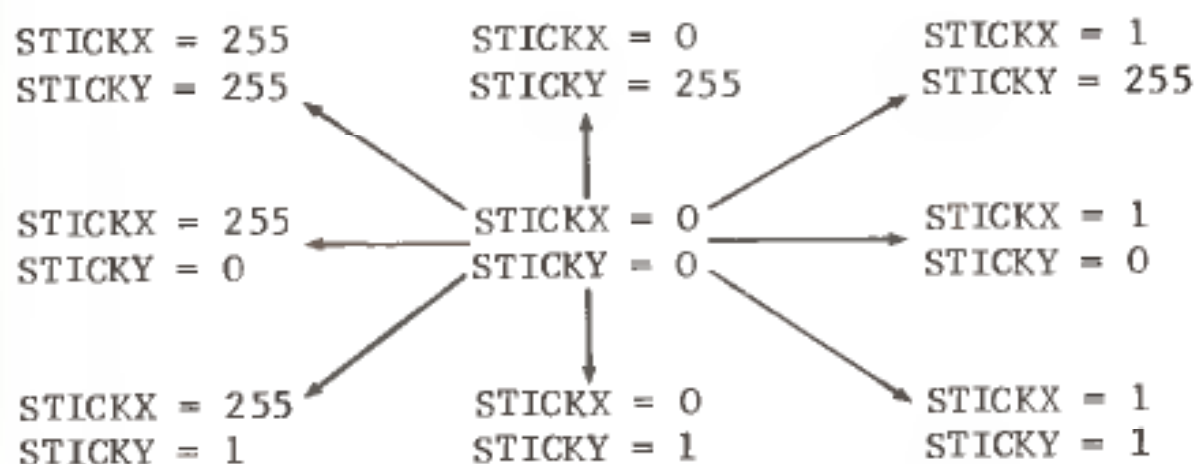
- STICKX (numéro manche à balai) qui indique la position horizontale du manche à balai

- 1 manche poussé vers la droite
(touche curseur droite pressée)
- 255 manche poussé vers la gauche
(touche curseur gauche pressée)
- 0 si le manche est au milieu
(aucune de ces deux touches n'est pressée)

- STICKY (numéro manche à balai) qui indique la position verticale du manche à balai

- 255 manche poussé vers le haut
(touche curseur haut pressée)
- 1 manche poussé vers le bas
(touche curseur bas pressée)
- 0 si le manche est au milieu
(aucune de ces deux touches n'est pressée)

Faisons un dessin des huit positions d'un manche à balai :



Nous les emploierons pour une manipulation dans un seul sens (horizontal ou vertical), ce qui est suffisant dans la plupart des jeux.

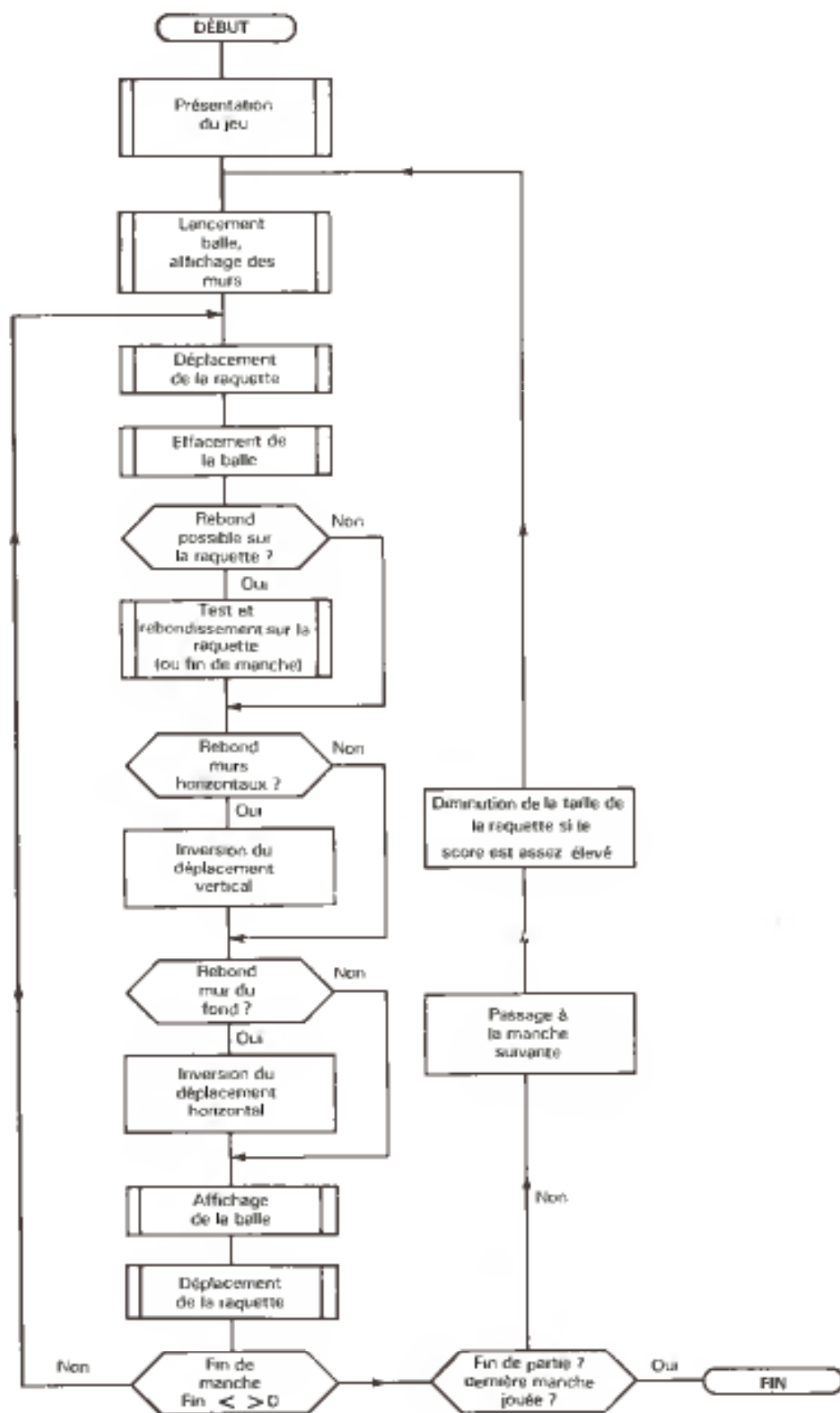
- ACTION (numéro manche à balai) qui indique si le bouton-poussoir du manche concerné est pressé (valeur 1) ou non (valeur 0).

Construisons maintenant un jeu de SQUASH qui comporte un seul objet en mouvement. Nous aborderons ensuite le jeu BOMBARDIER dans lequel deux objets sont en mouvement.

1.3 EXEMPLE DE JEU D'ACTION : LE SQUASH

1.3.1 Présentation

Le jeu de Squash consiste à faire rebondir une balle sur les murs d'une pièce, à l'aide d'une raquette qui peut se déplacer verticalement le long d'une paroi. La balle se déplace sous des angles différents selon l'endroit où elle rebondit.



Au milieu de la raquette, le rebond est normal. Au bord, le rebond est aléatoire. Après chaque rebond, la vitesse de la balle change aléatoirement.

Le jeu s'arrête lorsque la balle passe derrière la raquette. Trois balles sont disponibles (une par manche). Le déplacement de la raquette est obtenu à l'aide des touches de déplacement du curseur.

1.3.2 Organigramme et structure du programme

Lignes 90 à 240 Boucle principale

Cette boucle gère le déplacement de la balle ainsi que son rebond sur les murs. Le rebond sur la raquette est géré par l'appel d'un sous-programme. Le déplacement de la balle est réalisé :

- en effaçant la balle,
- en testant les rebonds,
- en réaffichant la balle.

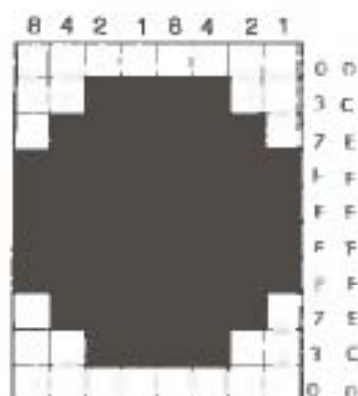
La balle se déplace de plus d'une case à la fois pour accélérer le jeu.

Dans le cas du rebond sur un mur horizontal, le déplacement vertical est inversé. Dans le cas du rebond sur le mur vertical du fond, le déplacement horizontal est inversé.

90	Positionnement à 0 de l'indicateur de fin de partie
100	Appel du sous-programme de déplacement de la raquette
120	Appel du sous-programme d'affichage de la raquette (fond/fond donne un effacement)
130 à 140	Test de rebond de la balle sur la raquette

	150 à	160	Rebond sur le mur du haut
	170 à	180	Rebond sur le mur du bas
	190 à	200	Rebond sur le mur du fond
	210		Avance de la balle
	220		Affichage de la balle (en noir)
	230 à	240	Test de fin de manche
Lignes	500 à	540	Sous-programme d'affichage de la balle

L'affichage de la balle utilise le caractère graphique suivant :



Lignes	600 à	690	Sous-programme de déplacement de la raquette
--------	-------	-----	--

Le déplacement de la raquette est géré de façon à créer l'illusion d'un mouvement continu. Pour cela, à chaque mouvement, l'une des extrémités de la raquette est effacée et rallumée de l'autre côté de la raquette.

Lignes	700 à	830	Sous-programme de rebond sur la raquette
--------	-------	-----	--

Ce sous-programme teste si la balle touche la raquette (YB situé dans la raquette, c'est-à-dire se trouvant dans l'intervalle YR-ER, YR+ER). Dans le cas contraire, il provoque une fin de manche. Dans le cas favorable, il crée un rebond aléatoire si la balle arrive sur une extrémité de la raquette ou un rebond normal si la balle arrive au centre de la raquette.

Lignes 1000 à 1120 Fin de manche

Ce bloc affiche le score du joueur, demande l'ordre de continuer et diminue la taille de la raquette si le score du joueur est élevé.

Lignes 5000 à 5100 Présentation du jeu

Le mot SQUASH est affiché en double hauteur et double largeur, ainsi qu'en vidéo inverse (jaune sur fond turquoise).

Lignes 5200 à 5400 Tracé des murs, de la raquette et lancement de la balle

Les murs horizontaux sont tracés par deux instructions DELIM. Le mur vertical nécessite une boucle d'affichage du caractère plein (code 127).

Lignes 10000 à 10040 Fin de partie

Le programme demande si vous voulez rejouer, et relance une nouvelle partie si votre réponse est positive.

PRINCIPALES VARIABLES :

XB, YB	Position de la balle
XR, YR	Position de la raquette
HB, VB	Déplacement de la balle
VR	Déplacement de la raquette
TR	Taille de la raquette

1.3.3 Liste des instructions

```
10 REM *****
20 REM * Squash *
30 REM *****
40 CLEAR 2000
50 GOSUB 5000
60 REM Afficher cadre, raquette, balle
70 GOSUB 5200
80 REM Programme principal
90 FIN=0
100 GOSUB 600
110 REM Déplacement de la balle
120 CB=CF:GOSUB 500
130 IF XB+HB>XR GOTO 150
140 GOSUB 700:GOTO 230
150 IF YB+VB>1 GOTO 170
160 IF YB>1 THEN YB=1:VB=-VB:PLAY"T101A"
:GOTO 230
170 IF YB+VB<23 GOTO 190
180 IF YB<23 THEN YB=23:VB=-VB:PLAY"T101
A":GOTO 230
190 IF XB+HB<39 GOTO 210
200 XB=38:HB=-HB:PLAY "T101A":GOTO 230
210 XB=XB+HB:YB=YB+VB
220 CB=0:GOSUB 500
230 IF FIN<>0 GOTO 1000
240 GOTO 100
500 REM Affichage de la balle
510 ET CB,0,0:CURSORX XB:CURSORY YB
520 PRINT CHR$(126)
530 DISPLAY 10
540 RETURN
600 REM Déplacement de la raquette
610 IF STICKY(2)=0 THEN RETURN
620 IF STICKY(2)=1 THEN VR=+1
630 IF STICKY(2)=255 THEN VR=-1
640 IF YR+ER*VR=1 THEN RETURN
650 IF YR+ER*VR=23 THEN RETURN
660 TX CF,0,0:CURSORX XR:CURSORY YR-ER*V
R:PRINT CHR$(127)
```

```

670 YR=YR+VR
680 TX 1,0,0:CURSORX XR:CURSORY YR+ER*VR
:PRINT CHR$(127)
690 RETURN
700 REM Test de rebond sur la raquette
710 IF YB<YR-ER THEN FIN=1:RETURN
720 IF YB>YR+ER THEN FIN=1:RETURN
730 REM Rebond de la balle
740 SR=SR+1:PLAY "t10o3a4":HB=INT(RND(1)
*6+1)
750 IF (YB=YR-ER) OR (YB=YR+ER) GOTO 780
760 CB=0:GOSUB 500:RETURN
770 REM Rebond aléatoire
780 ON INT(3*RND(1)+1) GOTO 790, 800, 81
0
790 VB=3:IF RND(1)<0.5 THEN VB=-VB:GOTO
820
800 VB=2:IF RND(1)<0.5 THEN VB=-VB:GOTO
820
810 VB=1:IF RND(1)<0.5 THEN VB=-VB
820 CB=0:GOSUB 500
830 RETURN
1000 REM Fin de manche
1010 PLAY "T1001G0FFFEEDDCCBBAA"
1020 TX 1,0,0:CURSORX 10:CURSORY 8
1030 PRINT "score: ";SR
1040 TX 1,0,0:CURSORX 10:CURSORY 10
1050 PRINT "fin de la manche No ";4-NB
1060 NB=NB-1
1070 IF NB=0 GOTO 10000
1080 PRINT :PRINT "RET pour la balle sui
vante":INPUT R$
1090 IF SR<35-10*NB GOTO 60
1100 REM Changement de la taille de la r
aquette
1110 IF TR>=3 THEN TR=TR-2
1120 GOTO 60
5000 REM Présentation
5010 NB=3:SR=0:TR=5
5020 INIT 4,4:TX 3,7,0
5030 CURSORX 10:CURSORY 10
5040 PRINT "s q u a s h "

```

```
5050 CURSORX 10:CURSORY 11
5060 PRINT "s q u a s h "
5070 TX 0,0,0:PRINT
5080 PRINT "RET pour la première balle":
INPUT R$
5090 SETET 126,"003c7effffff7e3c00"
5100 RETURN
5200 REM Lancement de la balle et tracé
du cadre
5210 CF=7:INIT CF,7:TX 4,0,0:PAGE
5220 CURSORX 1:CURSORY 0:DELIM4,4,0
5230 CURSORX 1:CURSORY 24:DELIM4,4,0
5240 FOR Y=1 TO 24
5250 CURSORX 39:CURSORY Y:PRINT CHR$(127
)
5260 NEXT Y
5270 REM Initialisations de la raquette
5280 XR=7:YR=12:VR=0:ER=(TR-1)/2
5290 TX 1,0,0:CURSORY YR-ER
5300 FOR I=1 TO TR
5310 CURSORX XR:PRINT CHR$(127)
5320 NEXT I
5330 REM Initialisations de la balle
5340 XB=38:YB=INT(RND(1)*16+4):HB=-INT(R
ND(1)*3+1)
5350 ON INT(3*RND(1)+1) GOTO 5360, 5370,
5380
5360 VB=3:IF RND(1)<0.5 THEN VB=-VB:GOTO
5390
5370 VB=2:IF RND(1)<0.5 THEN VB=-VB:GOTO
5390
5380 VB=1:IF RND(1)<0.5 THEN VB=-VB
5390 CB=0:GOSUB 500
5400 RETURN
10000 REM Fin de partie
10010 TX 0,0,0:PRINT:PRINT "Fin de la
partie"
10020 PRINT:PRINT "Voulez-vous rejouer"
;CHR$(4);:INPUT R$
10030 IF R$="0" OR R$="o" THEN RUN
10040 INIT6,6:END
```

1.4 EXEMPLE DE JEU D'ACTION : BOMBARDIER

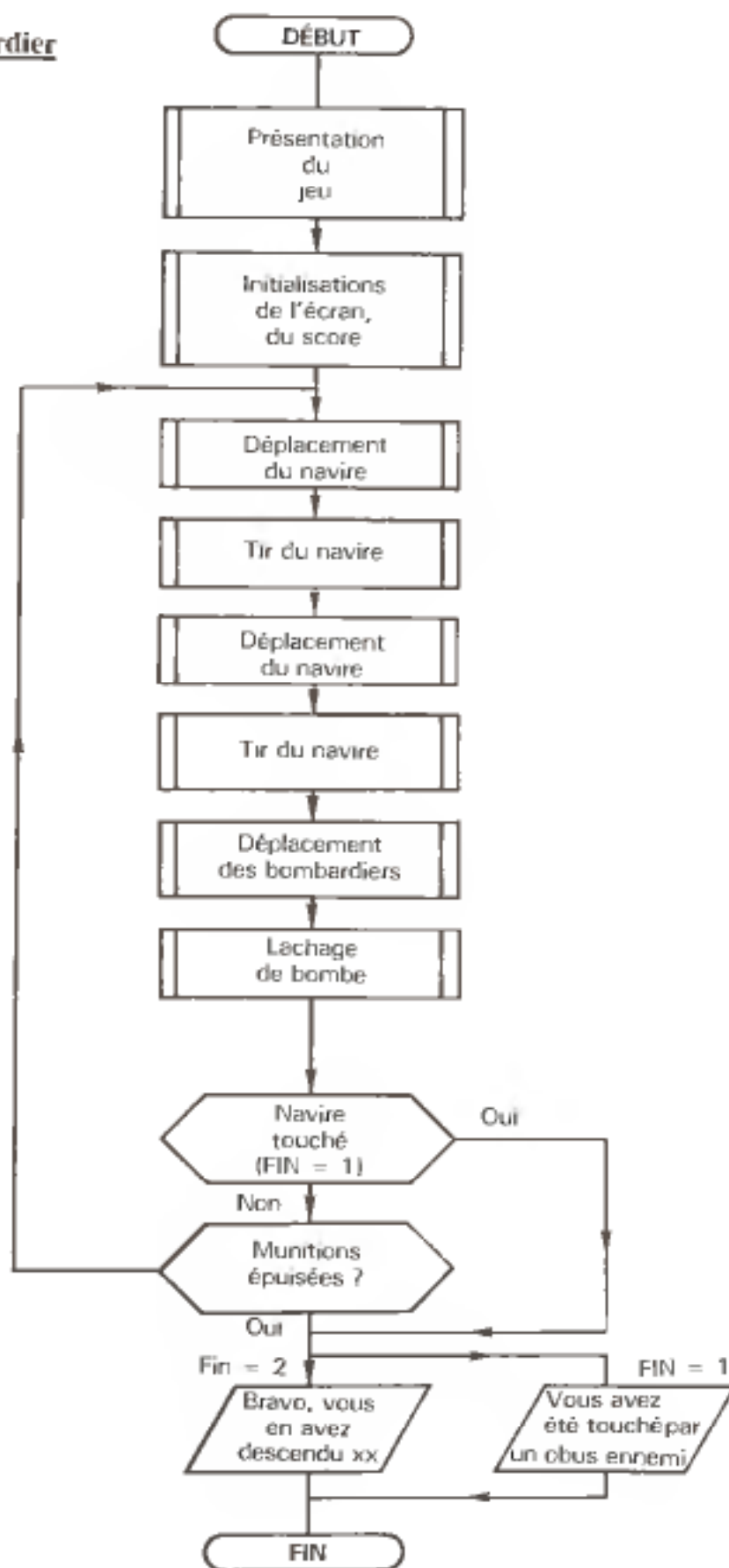
1.4.1 Présentation

Depuis plusieurs années vous êtes le commandant de bord du CINGLANT. Votre croiseur porte bien son nom, car vous avez burlingué avec lui sur toutes les mers du globe et il vous a toujours ramené sain et sauf à bon port. Ce navire, vous l'aimez... Même si l'amirauté l'a récemment modernisé en y installant une batterie anti-aérienne de 30 missiles. Cahin-caha, vous acceptez ces modifications car vous savez que des attaques aériennes sont possibles. Soudain, votre radar de bord détecte un écho qui se transforme très vite en une multitude de points. Submergé par le nombre, vous allez tout de même combattre avec courage. Mais pourrez-vous résister longtemps contre les bombardiers ?...

Heureusement pour vous, ils attaquent à tour de rôle.

1.4.2 Organigramme et structure du programme

Alors que le SQUASH ne gère principalement qu'un objet (la balle), ce programme dirige le mouvement de deux objets qui sont le vaisseau et l'avion, avec, en outre, les balles et obus. Pour bien vous rendre compte de l'organisation du programme, nous vous conseillons de taper successivement le "squelette" du jeu (commentaires et lignes RETURN pour les sous-programmes), puis la boucle principale, le sous-programme de présentation et le sous-programme de déplacement du navire. A cette étape, le navire doit suivre les ordres du manche à balai de la main droite. Ajoutez ensuite le déplacement des avions puis le

Bombardier

tir du navire et des avions, et enfin le module de fin de partie.

Lignes	60 à	120	Boucle principale
	60 à	90	Trois appels des sous-programmes de déplacement du vaisseau et de tir
	100 à	120	Test de fin de partie

Lignes	500 à	600	Déplacement du croiseur Ce sous-programme : - efface le croiseur, - lit les touches de déplacement du curseur - réaffiche le croiseur à sa nouvelle place sur l'écran
--------	-------	-----	---

Il est très utilisé dans plusieurs jeux de cet ouvrage.

Lignes	1000 à	1240	Tir du croiseur et test touché
--------	--------	------	--------------------------------

Lignes	1040 à	1070	Apparition d'un missile
--------	--------	------	-------------------------

Lignes	1080 à	1130	Avance du missile
--------	--------	------	-------------------

Lignes	1140 à	1160	Test touché
--------	--------	------	-------------

Lignes	1170 à	1210	Avion abattu - Appel d'un nouvel avion
--------	--------	------	--

Lignes	1220 à	1240	Affichage du missile à sa nouvelle position
--------	--------	------	---

Lignes	1500 à	1560	Avance et affichage de l'avion avec test de sortie d'écran et apparition dans ce cas d'un nouvel avion
--------	--------	------	--

Lignes	1600 à 1650	Apparition d'un nouvel avion par tirage aléatoire de X entre 0 et 39, Y entre 0 et 18.
Lignes	1700 à 1850	Tir de l'avion et test touché Ce sous-programme est construit de façon identique au tir du croiseur, à la différence près que l'obus "suit" l'avion
Lignes	5000 à 5150	Présentation du jeu et définition des caractères spéciaux
Lignes	5200 à 5310	Décor et mobiles (avion, navire)
	5210 à 5240	Décor
	5250 à 5300	Mobiles
Lignes	10000 à 10130	Fin de partie et affichage du message correspondant à la cause de fin.

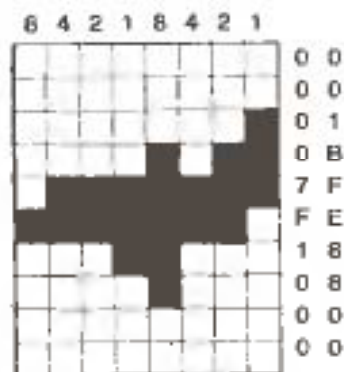
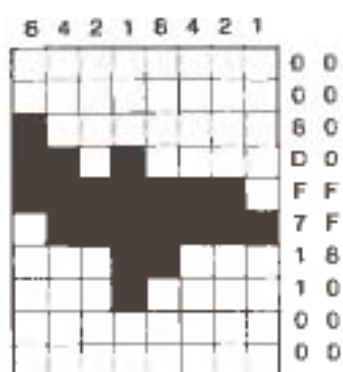
PRINCIPALES VARIABLES

XC, YC	Coordonnées du croiseur
DC	Déplacement horizontal du croiseur
TB	Tir d'une balle en cours
XB, YB	Coordonnées de la balle
TIR	Nombre de munitions
XA, Ya	Coordonnées de l'avion
DA	Déplacement horizontal de l'avion
TA	Tir de l'avion en cours
XO, YO	Coordonnées de l'obus

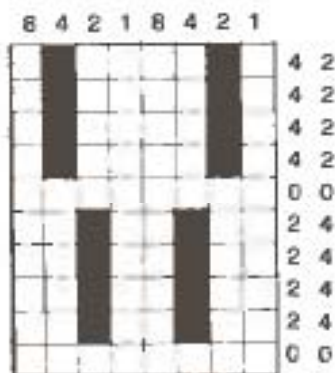
CARACTERES SPECIAUX

Nous avons défini les cinq caractères suivants :

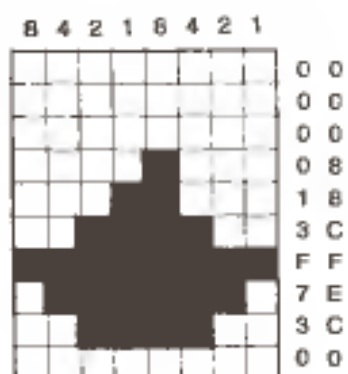
Avion droit, avion gauche, obus, missile, navire.



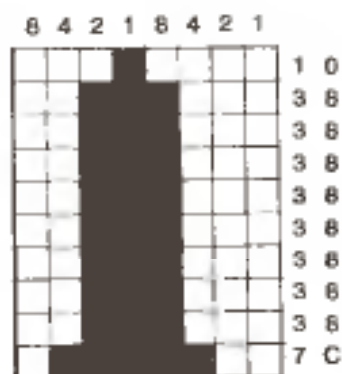
AVION



OBUS



NAVIRE



BALLE

1.4.3 Liste des instructions

```
10 REM *****
20 REM * BOMBARDIER *
30 REM *****
40 GOSUB 5000
50 GOSUB 5210:FIN=0
60 GOSUB 500:GOSUB 1000
70 GOSUB 500:GOSUB 1000
80 GOSUB 500:GOSUB 1000
90 GOSUB 1500:GOSUB 1700
100 IF FIN<>0 GOTO 10000
110 IF TIR<>0 GOTO 60
120 FIN=2:GOTO 10000
500 REM Déplacement du croiseur
510 EG 6,6,0:CURSORX XC:CURSOR YC
520 PRINT CHR$(CN);
530 IF STICKX(2)=1 THEN DC=+1
540 IF STICKX(2)=255 THEN DC=-1
550 XC=XC+DC
560 IF XC<0 THEN XC=0
570 IF XC>39 THEN XC=39
580 EG 1,6,0:CURSORX XC:CURSOR YC
590 PRINT CHR$(CN);
600 RETURN
1000 REM Tir du croiseur et test touché
1010 IF TB=1 GOTO 1080
1020 IF ACTION(2)=0 THEN RETURN
1040 REM Tir demandé
1050 XB=XC:YB=YC-1:TB=1
1060 EG 1,6,0:CURSORX XB:CURSOR YB
1070 PRINT CHR$(126);:RETURN
1080 REM Avance du missile
1090 EG 6,6,0:CURSORX XB:CURSOR YB
1100 PRINT CHR$(126);
1110 YB=YB-1
1120 IF YB>0 THEN 1150
1130 TB=0:TIR=TIR-1:RETURN
1140 REM Test touché
1150 IF YB<>YA GOTO 1220
```

```
1160 IF XB<>XA GOTO 1220
1170 CURSORX XA:CURSORY YA:EG 6,6,0
1180 PRINT CHR$(123);
1190 SR=SR+1:TB=0:TIR=TIR-1:PLAY"o1aaa"
1200 GOSUB 1600
1210 RETURN
1220 EG 1,6,0:CURSORX XB:CURSORY YB
1230 PRINT CHR$(126);
1240 RETURN
1500 REM Affichage de l'avion
1510 CURSORX XA:CURSORY YA:EG 6,6,0
1520 PRINT CHR$(123);:XA=XA+DA
1530 IF XA>39 OR XA<0 THEN GOSUB 1600
1540 CURSORX XA:CURSORY YA:EG 0,6,0
1550 PRINT CHR$(AV);:DISPLAY 10
1560 RETURN
1600 REM Apparition d'un nouvel avion
1610 XA=39*INT(RND(1)+0.5)
1620 YA=INT(RND(1)*18)
1630 DA=SGN(38-XA):AV=123
1640 IF DA>0 THEN AV=122
1650 RETURN
1700 REM Tir des avions
1710 IF TA=1 GOTO 1750
1720 XO=XA:YO=YA+1:TA=1:DO=DA
1730 EG 0,6,0:CURSORX XO:CURSORY YO
1740 PRINT CHR$(124);:RETURN
1750 REM Avance des obus
1760 EG 6,6,0:CURSORX XO:CURSORY YO
1770 PRINT CHR$(124);
1780 YO=YO+1:IF YO<19 THEN XO=XO+DO
1790 IF XO<0 THEN XO=0
1800 IF XO>39 THEN XO=39
1810 IF YO>19 THEN TA=0:RETURN
1820 IF YO<19 THEN 1850
1830 IF XO<>XC GOTO 1850
1840 FIN=1:PLAY "01T10GFEBAAA":RETURN
1850 EG 0,6,0:CURSORX XO:CURSORY YO
1860 PRINT CHR$(124);
1870 RETURN
5000 REM Présentation
5010 INIT 6,6:TX 0,7,0
```

```
5020 CURSORX 9:CURSORY 10
5030 PRINT "B O M B A R D I E R "
5040 CURSORX 9:CURSORY 11
5050 PRINT "B O M B A R D I E R "
5060 TX 0,0,0:PRINT
5070 PRINT SPC(9);"RET pour commencer"
5080 INPUT R$
5090 REM Caractères graphiques
5100 SETEG 122,"000080D0FE7F18100000"
5110 SETEG 123,"0000010B7FFE18080000"
5120 SETEG 124,"42424242002424242400"
5130 SETEG 125,"00000008183CFF7E3C00"
5140 SETEG 126,"1038383838383838387C"
5150 RETURN
5200 REM Tracé du décor et des mobiles
5210 INIT 6,6
5220 CURSORX 0:CURSORY 20:DELIM 4,4,0
5230 CURSORX 0:CURSORY 21:DELIM 4,4,0
5240 CURSORX 0:CURSORY 22:DELIM 4,4,0
5250 REM Affichage des mobiles
5260 TA=0:TIR=30:DA=1:YA=0:AV=122
5270 GOSUB 1500
5280 XC=21:YC=19:CN=125:TB=0
5290 CURSORX XC:CURSORY YC:EG 1,6,0
5300 PRINT CHR$(CN);
5310 RETURN
10000 REM Fin de partie
10010 INIT 6,6:TX 0,0,0
10020 ON FIN-1 GOTO 10060
10030 PRINT "Vous avez été touché par un
missile"
10040 PRINT "ennemi"
10050 GOTO 10070
10060 PRINT "Plus de munitions."
10070 PRINT "Bravo, vous avez abattu ";
10080 PRINT SR;" bombardier";
10090 IF SR>1 THEN PRINT "s"
10100 CURSORX 1:CURSORY 20
10110 INPUT "Une autre partie";R$
10120 IF R$="o" OR R$="O" THEN RUN
10130 INIT 6,6:END
```

1.5 OPTIMISATION DE LA VITESSE D'UN JEU

Pour optimiser la vitesse d'exécution d'un jeu ou de tout programme il existe quelques règles simples :

- supprimer les lignes de remarques et de commentaires
- resserrer au maximum les instructions, en enlevant les espaces inutiles (tout espace est traité par l'interpréteur, ce qui prend du temps)
- mettre au début les parties du programme qui sont employées le plus souvent (faibles numéros de lignes)
- placer les sous-programmes peu utilisés à la fin du programme (forts numéros de lignes)
- utiliser des variables au lieu de constantes
- éviter au maximum l'emploi de tableaux

Nous n'emploierons pas forcément ces techniques dans les jeux qui suivent. En effet, le programme perd en lisibilité, et tel n'est pas l'objectif visé dans ce livre d'initiation.



Jeux d'action à un joueur

2.1 STOCK-CAR

2.1.1 Présentation

La ville de Busopolis est en pleine effervescence car la célèbre course de stock-car va bientôt commencer. Cette année, vous avez décidé de tenter votre chance et de risquer votre vie pour un peu de gloire. Au volant de votre bolide rouge, vous allez foncer et entrer en contact avec le maximum de balises BEGA (balises se dématérialisant au contact de votre pare-choc). Chaque balise renversée vous rapporte un point et le contact de son générateur avec votre pare-choc la dématérialise. La tâche ne sera pas facile, car votre voiture est équipée d'un émetteur vers lequel se dirige une voiture-robot bourrée d'explosifs ; votre seul avantage par rapport à celle-ci réside dans votre possibilité de vous déplacer de 2 pistes à chaque croisement alors que la voiture-robot ne peut bouger que d'une seule piste à la fois. Si vous réussissez à dématérialiser toutes les balises, il en apparaîtra une nouvelle série.

Pour changer de piste, vous utiliserez le manche à balai de la main droite. Vous ne disposez que de trois voitures. Ne les gêchez pas !!

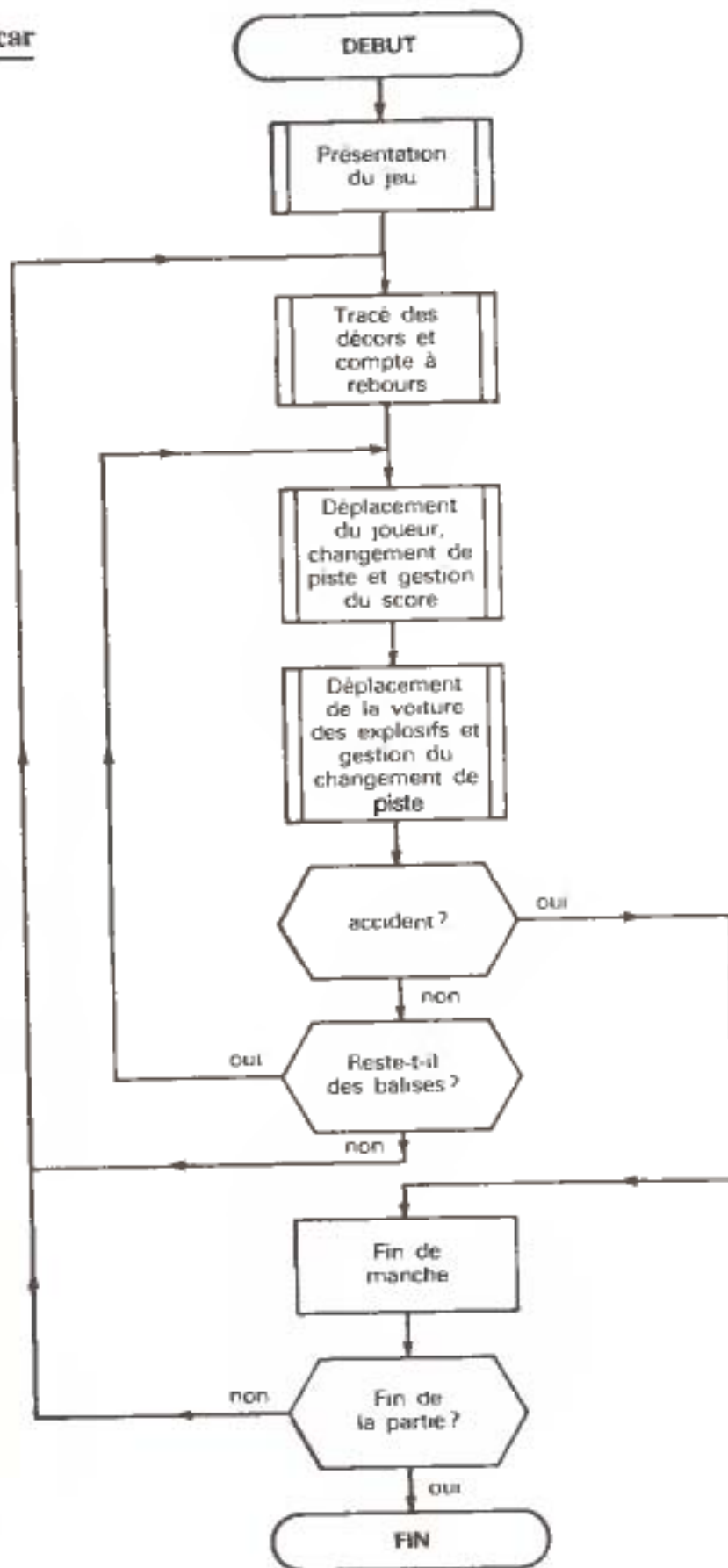
2.1.2 Organigramme et étude du programme

STRUCTURE

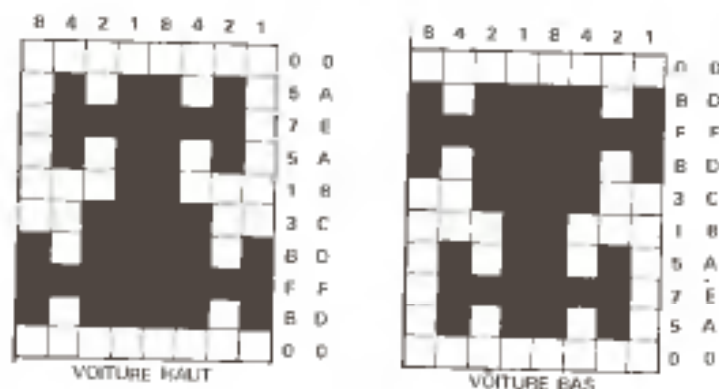
La structure du programme STOCK-CAR est assez classique :

Lignes	10 à 40	Présentation et initialisations
Ligne	50	Initialisation de la manche, tracé des décors
Lignes	60 à 110	Boucle principale avec test d'accident (à la ligne 90) et test de rencontre de toutes les balises (à la ligne 100)
Lignes	800 à 840	Affichage d'une voiture
Lignes	1000 à 1599	Sous-programme de déplacement de la voiture du joueur
	1010	Calcul des extrémités de la piste où se trouve le joueur
	1020	Effacement de la voiture
	1030	Déplacement de la voiture
	1040 à 1070	Test de présence de la voiture dans un virage
	1080 à 1140	Test de changement de piste
	1150 à 1170	Virage gauche haut
	1180 à 1200	Virage gauche bas
	1210 à 1230	Virage droite haut
	1240 à 1260	Virage droite bas
	1270 à 1340	Changement vertical de piste
	1350 à 1420	Changement horizontal de piste
	1430 à 1480	Gestion du score
	1490 à 1500	Affichage de la voiture

Stockcar



Lignes	2000 à	2370	Sous-programme de déplacement de la voiture explosifs
	2020		Effacement de la voiture
	2030		Déplacement de la voiture
	2040 à	2070	Test de présence de la voiture dans un virage
	2080 à	2100	Test de changement de piste
	2110 à	2130	Virage gauche haut
	2140 à	2160	Virage gauche bas
	2170 à	2190	Virage droite haut
	2200 à	2220	Virage droite bas
	2230 à,	2290	Changement de piste de la voiture explosifs
	2300 à	2340	Remise en place d'une éventuelle balise
	2350 à	2360	Affichage de la voiture explosifs
Lignes	3000 à	3130	Module de gestion de fin d'une manche - affichage de BOUM !! - test de fin de partie
Lignes	5000 à	5150	Présentation du jeu et définition des caractères graphiques
Lignes	5200 à	5570	Tracé du décor
	5220 à	5260	Affichage des balises électriques
	5270 à	5390	Tracé des murs de la piste
	5400 à	5420	Tracé du bloc central
	5430 à	5510	Zones libres
	5520 à	5540	Voiture du joueur
	5550 à	5570	Voiture explosifs
Lignes	5800 à	5860	Remplissage d'une zone
Lignes	5900 à	5960	Affichage du compte à rebours
Lignes	10000 à	10110	Fin de partie avec affichage du score



2.1.3 Liste des instructions

```

10 REM *****
20 REM * Stock-car *
30 REM *****
40 GOSUB 5000:SR=0:NV=3
50 GOSUB 5200:GOSUB 5900
60 GOSUB 1000
70 GOSUB 2000
80 REM Test de fin de manche
90 IF XE=XJ AND YE=YJ GOTO 3000
100 IF SM>=108 THEN 50
110 GOTO 60
800 REM Affichage d'une voiture
810 ET CL,0,0
820 CURSORX X+9 : CURSORY Y+1
830 PRINT CHR$(CD)
840 RETURN
1000 REM Déplacer la voiture du joueur
1010 AL=2*PJ:BE=21-AL
1020 X=XJ:Y=YJ:CL=0:CD=CJ:GOSUB 800
1030 XJ=XJ+HJ:YJ=YJ+VJ
1040 IF XJ=AL AND YJ=AL GOTO 1150
1050 IF XJ=AL AND YJ=BE GOTO 1180
1060 IF XJ=BE AND YJ=AL GOTO 1210
1070 IF XJ=BE AND YJ=BE GOTO 1240
1080 REM Test de changement de piste
1090 IF HJ=0 GOTO 1120
1100 IF XJ>9 AND XJ<12 GOTO 1270
1110 GOTO 1430
1120 IF VJ=0 GOTO 1430

```

```
1130 IF YJ>9 AND YJ<12 GOTO 1350
1140 GOTO 1430
1150 REM Virage haut gauche
1160 HJ=0:VJ=1:CJ=124
1170 GOTO 1430
1180 REM Virage bas gauche
1190 HJ=1:VJ=0:CJ=126
1200 GOTO 1430
1210 REM Virage haut droite
1220 HJ=-1:VJ=0:CJ=125
1230 GOTO 1430
1240 REM Virage bas droite
1250 HJ=0:VJ=-1:CJ=123
1260 GOTO 1430
1270 REM Changement vertical de piste
1280 IF STICKY(2)=1 THEN DP=-HJ
1290 IF STICKY(2)=255 THEN DP=HJ
1300 IF PJ+DP<1 THEN DP=0
1310 IF PJ+DP>3 THEN DP=0
1320 PJ=PJ+DP
1330 YJ=YJ-2*DP*HJ
1340 GOTO 1430
1350 REM Changement horizontal de piste
1360 DP=0
1370 IF STICKX(2)=1 THEN DP=VJ
1380 IF STICKX(2)=255 THEN DP=-VJ
1390 IF PJ+DP<1 THEN DP=0
1400 IF PJ+DP>3 THEN DP=0
1410 PJ=PJ+DP
1420 XJ=XJ+2*DP*VJ
1430 REM Score
1440 IF PISTE(XJ,YJ)<>2 GOTO 1490
1450 SR=SR+1:SM=SM+1:PISTE(XJ,YJ)=0
1460 TX 7,0,0:SR$=STR$(SR):SOUND10,10,2
1470 CURSORX 22-LEN(SR$):CURSORY 12
1480 PRINT RIGHT$(SR$,LEN(SR$)-1)
1490 X=XJ:Y=YJ:CL=1:CD=CJ
1500 GOSUB 800
1999 RETURN
2000 REM Déplacer la voiture robot
2010 AL=2*PE:BE=21-AL
2020 X=XE:Y=YE:CL=0:CD=CE:GOSUB 800
```



```
2030 AXE=XE:AYE=YE:XE=XE+HE:YE=YE+VE
2040 IF XE=AL AND YE=AL GOTO 2110
2050 IF XE=AL AND YE=BE GOTO 2140
2060 IF XE=BE AND YE=AL GOTO 2170
2070 IF XE=BE AND YE=BE GOTO 2200
2080 REM Test de changement de piste
2090 IF XE=9 OR YE=9 GOTO 2230
2100 GOTO 2300
2110 REM Virage haut gauche
2120 HE=1:VE=0:CE=126
2130 GOTO 2300
2140 REM Virage bas gauche
2150 HE=0:VE=-1:CE=123
2160 GOTO 2300
2170 REM Virage haut droite
2180 HE=0:VE=1:CE=124
2190 GOTO 2300
2200 REM Virage bas droite
2210 HE=-1:VE=0:CE=125
2220 GOTO 2300
2230 REM Changement de piste
2240 DP=SGN(PJ-PE)
2250 IF PE+DP<1 THEN DP=0
2260 IF PE+DP>3 THEN DP=0
2270 PE=PE+DP
2280 IF VE=0 THEN YE=YE+2*DP*HE
2290 IF HE=0 THEN XE=XE-2*DP*VE
2300 REM Remise en place des balises
2310 IF PISTE(AXE,AYE)<>2 GOTO 2350
2320 CURSORX 9+AXE:CURSORY AYE+1
2330 TX 4,0,0
2340 PRINT "*"
2350 X=XE:Y=YE:CL=5:CD=CE
2360 GOSUB 800
2370 RETURN
3000 REM Fin de manche
3010 NV=NV-1:SOUND 5,255,255
3020 INIT3,3:TX0,7,0
3030 CURSORX10:CURSORY 10
3040 PRINT "B O U M !!!!!!"
3050 CURSORX 10:CURSORY 11
3060 PRINT "B O U M !!!!!!"
```

```

3070 FOR I=1TO5000:NEXTI
3080 IF NV=0 GOTO 10000
3090 SM=0
3100 TX0,0,0:CURSORX 5:CURSORY 18
3110 PRINT"RET pour la manche suivante"
3120 INPUT R$
3130 INIT0,2:GOTO 50
5000 REM Présentation
5010 INIT 3,3:TX1,7,0
5020 CURSORX 6:CURSORY 8
5030 PRINT "s t o c k - c a r "
5040 CURSORX 6:CURSORY 9
5050 PRINT "s t o c k - c a r "
5060 PRINT :PRINT :PRINT :TX0,0,0
5070 PRINT "RET pour commencer ";
5080 INPUT R$
5090 SETET 123,"005A7E5A183CBDFFB00"
5100 SETET 124,"00BDFFB03C185A7E5A00"
5110 SETET 125,"000EE24FFFFFF4FE20E00"
5120 SETET 126,"00E047F2FFFFFF247E000"
5130 INIT0,2
5140 DIM PISTE(20,20)
5150 RETURN
5200 REM Tracé du circuit
5210 TX 4,0,0:PAGE
5220 REM Bornes électriques
5230 FOR I=1 TO 20:FOR J=1 TO 20
5240 CURSORX I+9:CURSORY J+1
5250 PRINT "*"
5260 PISTE(I,J)=2:NEXTJ,I
5270 REM Tracé de la piste et
5280 REM murs internes
5290 FOR K=0TO3
5300 AL=1+2*K:BE=21-AL
5310 CZ=3:J1=AL:J2=J1:I1=AL:I2=BE
5320 GOSUB 5800
5330 CZ=3:J1=BE:J2=J1:I1=AL:I2=BE
5340 GOSUB 5800
5350 CZ=3:J1=AL:J2=BE:I1=AL:I2=I1
5360 GOSUB 5800
5370 CZ=3:J1=AL:J2=BE:I1=BE:I2=I1
5380 GOSUB 5800

```

```
5390 NEXT K
5400 REM Bloc central
5410 CZ=3:J1=8:J2=13:I1=8:I2=13
5420 GOSUB 5800
5430 REM Effacement étoiles en trop
5440 CZ=0:J1=9:J2=12:I1=2:I2=6
5450 GOSUB 5800
5460 CZ=0:J1=9:J2=12:I1=15:I2=19
5470 GOSUB 5800
5480 CZ=0:J1=2:J2=6:I1=9:I2=12
5490 GOSUB 5800
5500 CZ=0:J1=15:J2=19:I1=9:I2=12
5510 GOSUB 5800
5520 REM Voiture joueur
5530 XJ=2:YJ=19:CJ=126:FJ=1:VJ=0:HJ=1
5540 X=XJ:Y=YJ:CL=1:CD=CJ:GOSUB 800
5550 REM Voiture robot
5560 XE=19:YE=10:CE=124:PE=1:VE=1:HE=0
5570 X=XE:Y=YE:CL=5:CD=CE:GOSUB 800
5580 RETURN
5800 REM Remplissage zone I1I2,J1J2
5810 TX CZ,0,0
5820 FOR I=I1 TO I2:FOR J=J1 TO J2
5830 PISTE(I,J)=CZ-2
5840 CURSORX I+9:CURSORY J+1
5850 PRINT CHR$(127):NEXTJ:NEXT I
5860 RETURN
5900 REM Copmte à rebours
5910 FOR I=5 TO 0 STEP-1
5920 CURSORX 20:CURSORY 12
5930 TX1.0,0:PRINT RIGHT$(STR$(I),1)
5940 FOR J=1 TO 100:NEXT J
5950 NEXT I
5960 RETURN
10000 REM Fin de partie
10010 INIT 7,7:TX 4,0,0
10020 PRINT "Vos trois voitures sont à l
a ferraille."
10030 PRINT :PRINT
10040 PRINT "Votre score est de ";SR
10050 CURSORX 1:CURSORY 20
10060 PRINT "Voulez vous rejouer ";
```

```
10070 INPUT R$
10080 IF R$="0" OR R$="c" THEN RUN
10090 INIT 6,6:END
```

2.2 LA CHENILLE INFERNALE

2.2.1 Présentation

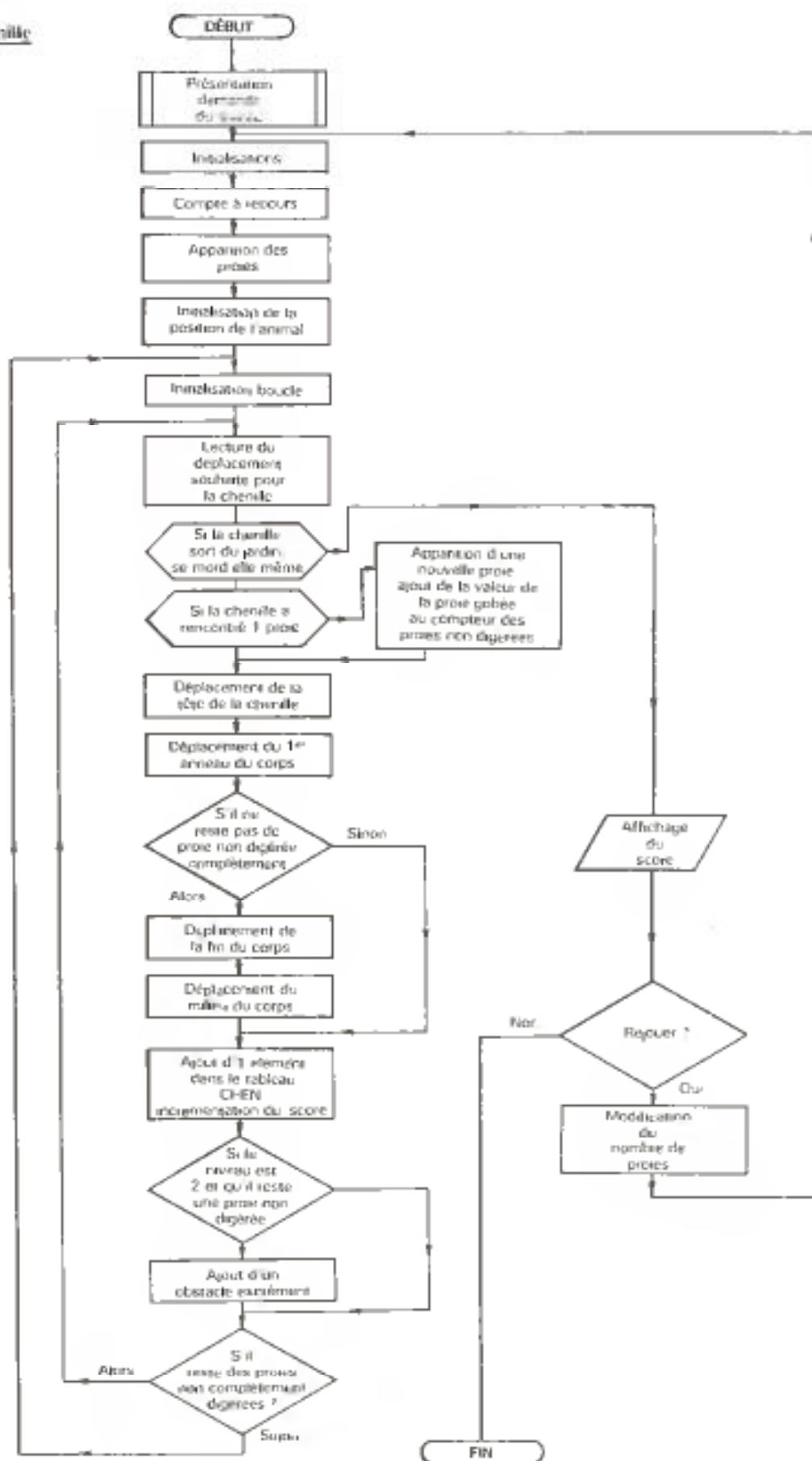
Bien avant de pouvoir être papillon, il vous a fallu passer par le stade de la chenille. Souvenez-vous de l'époque où vous n'étiez encore qu'un bébé chenille "Basicum Philipis" en quête de nourriture. Maintenant que vous êtes père de famille, il vous faut prendre en main le destin d'un autre bébé chenille. En vous aidant du manche à balai, vous le conduirez vers des tas de nourriture plus ou moins riches. A chaque fois que vous mangerez un point, la chenille grandira d'un anneau. Si vous réussissez à créer 254 anneaux, vous aurez gagné en amenant la chenille à l'état de papillon. Bien évidemment, il ne faudra pas que le bébé sorte de l'écran ou qu'il se mange lui-même, car ce serait la mort à coup sûr.

Attention, il vous est possible de jouer ce jeu de manière plus sophistiquée en utilisant le deuxième niveau de jeu : la chenille laissera des obstacles (... des excréments !) après chaque tas de nourriture avalé...

2.2.2 Organigramme et étude du programme

Ce programme est assez complexe. En premier lieu, la chenille se déplace (de tout son corps) ; il faut donc gérer ce mouvement et conserver en mémoire les coordonnées de tous les éléments de la chenille dans un tableau que nous avons appelé CHEN. Le déplacement en mémoire de la chenille consistera donc à déplacer tous les éléments du tableau.

Chenille



En second lieu il faut prévoir un mécanisme d'allongement du corps durant la digestion des proies, et la création d'excréments si le niveau est 2.

STRUCTURE

Lignes	10 à	110	Présentation, initialisations
Lignes	120 à	170	Compte à rebours
Lignes	180 à	210	Proies
Lignes	220 à	290	Initialisation de la chenille
Lignes	300 à	350	Lecture des touches de déplacement du curseur pour faire la tête de la chenille
Lignes	360 à	460	Test sortie, proie, obstacle
Lignes	470 à	490	Déplacement de la tête
Lignes	500 à	520	Déplacement du début du corps
Lignes	530 à	570	Effacement de la fin du corps
Lignes	580 à	660	Déplacement du milieu du corps
Lignes	670 à	740	Ajout d'excréments si le niveau est 2
Lignes	3000 à	3080	Tirage nouvelle proie
Lignes	5000 à	5290	Présentation et initialisations
Lignes	10000 à	10150	Fin de partie

REMARQUE :

Pour utiliser ce jeu avec un joystick il suffit de remplacer le chiffre 2 utilisé dans STICKX, STICKY par 0 (ou 1).

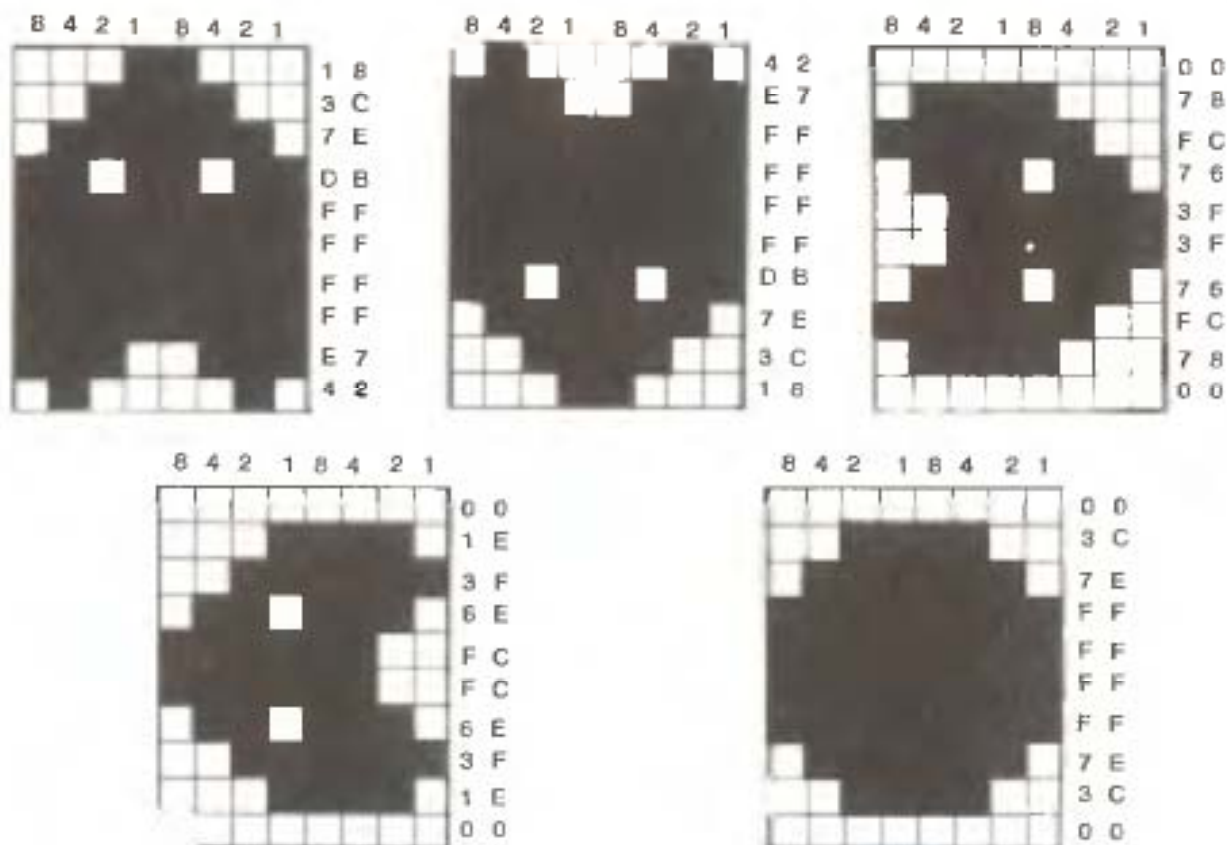
PRINCIPALES VARIABLES

ECRAN (x,y)	Copie de l'écran gérée par le programme et qui contient les valeurs 0 si la case de l'écran est vide 1 à 9 selon la valeur de la proie située sur l'écran 10 si la case de l'écran contient un obstacle
CHEN	Tableau des coordonnées du corps
X,Y	Coordonnées de la tête de la chenille
H,V	Déplacements de la tête de la chenille
PP	Indicateur de proie
CP	Compteur des proies non digérées
SR	Score
NOUR	Nombre de proies sur l'écran

CARACTERES SPECIAUX

Nous utilisons cinq caractères :

- quatre têtes de chenille
- un corps



2.2.3 Liste des instructions

```

10 REM *****
20 REM * LA CHENILLE INFERNALE *
30 REM *****
40 GOSUB 5000
50 DIM ECRAN(40,24),CHEN(255,2)
60 NOUR=20:INIT 6,6:TX 4,0,0
70 CURSORY 10:PRINT "patientez sup..."
80 FOR I=0 TO 40:FOR J=0 TO 24
90 ECRAN(I,J)=0:NEXT J:NEXT I
100 FOR I=1 TO 255:FOR J=0 TO 2
110 CHEN(I,J)=0:NEXT J:NEXT I
120 REM compte à rebours
130 INIT 6,6:TX 1,0,0
140 FOR I=5 TO 0 STEP-1
150 CURSORX 17:CURSORY 11:PRINT I
160 FOR J=1 TO 500:NEXT J
170 NEXT I
180 REM tracé des proies
190 INIT 7,3:PAGE
200 FOR I=1 TO NOUR:TX 4,0,0:GOSUB 3000:
NEXT I
210 TX 4,0,0:GOSUB 3000
220 REM initialisation chenille
230 X=20:Y=11:H=0:V=1:SR=1:TT=121
240 ET 1,0,0:CURSORX X
250 CURSORY Y:PRINT CHR$(TT)
260 CURSORY Y-1:CURSORX X
270 PRINT CHR$(126)
280 CHEN(1,1)=X:CHEN(1,2)=Y-1
290 CP=1:PP=0
300 REM déplacement chenille
310 IF STICKX(2)=1 THEN H=1:V=0:TT=122:G
OTO 350
320 IF STICKX(2)=255 THEN H=-1:V=0:TT=12
3:GOTO 350
330 IF STICKY(2)=1 THEN H=0:V=1:TT=121:G
OTO 350
340 IF STICKY(2)=255 THEN H=0:V=-1:TT=12
0

```



```
350 X=X+H:Y=Y+V
360 REM sortie, obstacle
370 IF X<1 OR X>39 GOTO 10000
380 IF Y<0 OR Y>24 GOTO 10000
390 IF ECRAN(X,Y)>9 GOTO 10000
400 IF SR>254 THEN INIT 6,6:GOTO 10040
410 REM pas de proie
420 IF ECRAN(X,Y)=0 GOTO 470
430 REM proie
440 CP=CP+ECRAN(X,Y)-1:PP=1
450 REM nouvelle proie
460 TX 5,0,0:GOSUB 3000
470 REM déplacement tête
480 ET1,0,0:CURSORX X:CURSOR Y
490 PRINT CHR$(TT):ECRAN(X,Y)=10
500 REM déplacement du début du corps
510 CURSORX X-H:CURSOR Y-V:PRINT CHR$(1
26)
520 IF PP=1 THEN 640
530 REM effacement fin du corps
540 XC=CHEN(1,1):YC=CHEN(1,2)
550 TX 7,0,0:CURSORX XC:CURSOR YC
560 PRINT CHR$(127)
570 ECRAN (XC,YC)=0
580 REM déplacement milieu du corps
590 FOR I=1 TO SR-1
600 CHEN(I,1)=CHEN(I+1,1)
610 CHEN(I,2)=CHEN(I+1,2)
620 NEXT I
630 GOTO 650
640 SR=SR+1
650 CHEN(SR,1)=X-H:CHEN(SR,2)=Y-V
660 CP=CP-1
670 REM niveau
680 IF NIVEAU<>2 OR PP<>1 GOTO 730
690 TX 4,0,0:CURSORX XC:CURSOR YC
710 PRINT CHR$(127)
720 ECRAN (XC,YC)=10
730 IF CP=0 GOTO 290
740 GOTO 310
3000 REM affichage d'une proie
3010 OX=INT(RND(1)*37+1)
```

```
3020 OY=INT(RND(1)*24)
3030 CURSORX OX:CURSORY OY
3040 IF ECRAN(OX,OY)<>0 GOTO 3000
3050 N=INT(RND(1)*8+1)
3060 PRINT RIGHT$(STR$(N),1)
3070 ECRAN(OX,OY)=N
3080 RETURN
5000 REM présentation
5010 INIT 6,6:TX 4,7,0
5020 CURSORX 8:CURSORY 6
5030 PRINT "o h e n i l l e "
5040 CURSORX 8:CURSORY 7
5050 PRINT "o h e n i l l e "
5060 CURSORX 1:CURSORY 13
5070 TX0,0,0
5080 PRINT "Le but de ce jeu est de fair
e avancer"
5090 PRINT "la chenille pour manger les
5100 PRINT "affichées sur l'écran"
5110 PRINT
5120 PRINT "Pour commencer, appuyer sur
RET"
5130 REM Choix du niveau
5140 INPUT R$
5150 INIT 6,6:CURSORY 1:TX1,0,0
5160 PRINT "Quel niveau choisissez-vous
?"
5170 PRINT "1: la chenille emmène ses pr
oies et"
5180 PRINT "    son corps s'allonge de la
valeur"
5190 PRINT "    de la proie."
5200 PRINT "2: la chenille laisse en plu
s un"
5210 PRINT "    obstacle."
5220 PRINT :INPUT "Choix";NIVEAU
5230 IF NIVEAU<1 OR NIVEAU>2 GOTO 5150
5240 SETET 120,"183C7EDBFFFFFFFFFE742"
5250 SETET 121,"42E7FFFFFFFFFDB7E3C18"
5260 SETET 122,"0078FC763F3F76FC7800"
5270 SETET 123,"001E3F6EFCFC6E3F1E00"
5280 SETET 126,"003C7EFFFFFFFFF7E3C00"
```

```
5290 RETURN
10000 REM fin de partie
10010 INIT 6,6:TX 0,0,0:CURSORX 1:CURSOR
Y 1
10020 PRINT "vous avez perdu"
10030 REM score
10040 TX 0,0,0:CURSORX 1:CURSOR Y 10
10050 PRINT "votre score est de: ";SR-1
10060 IF SR<254 GOTO 10080
10070 PRINT "Bravo, c'est le score maxim
al"
10080 CURSORX 1:CURSOR Y 20
10090 PRINT "voulez-vous rejouer?"
10100 INPUT R$
10110 IF R$="0" OR R$="o" GOTO 10130
10120 INIT 6,6:END
10130 IF SR>40 THEN NOUR=SR/2:GOTO 80
10140 NOUR=20:GOTO 80
```

2.3 ENVAHISSEURS

2.3.1 Présentation

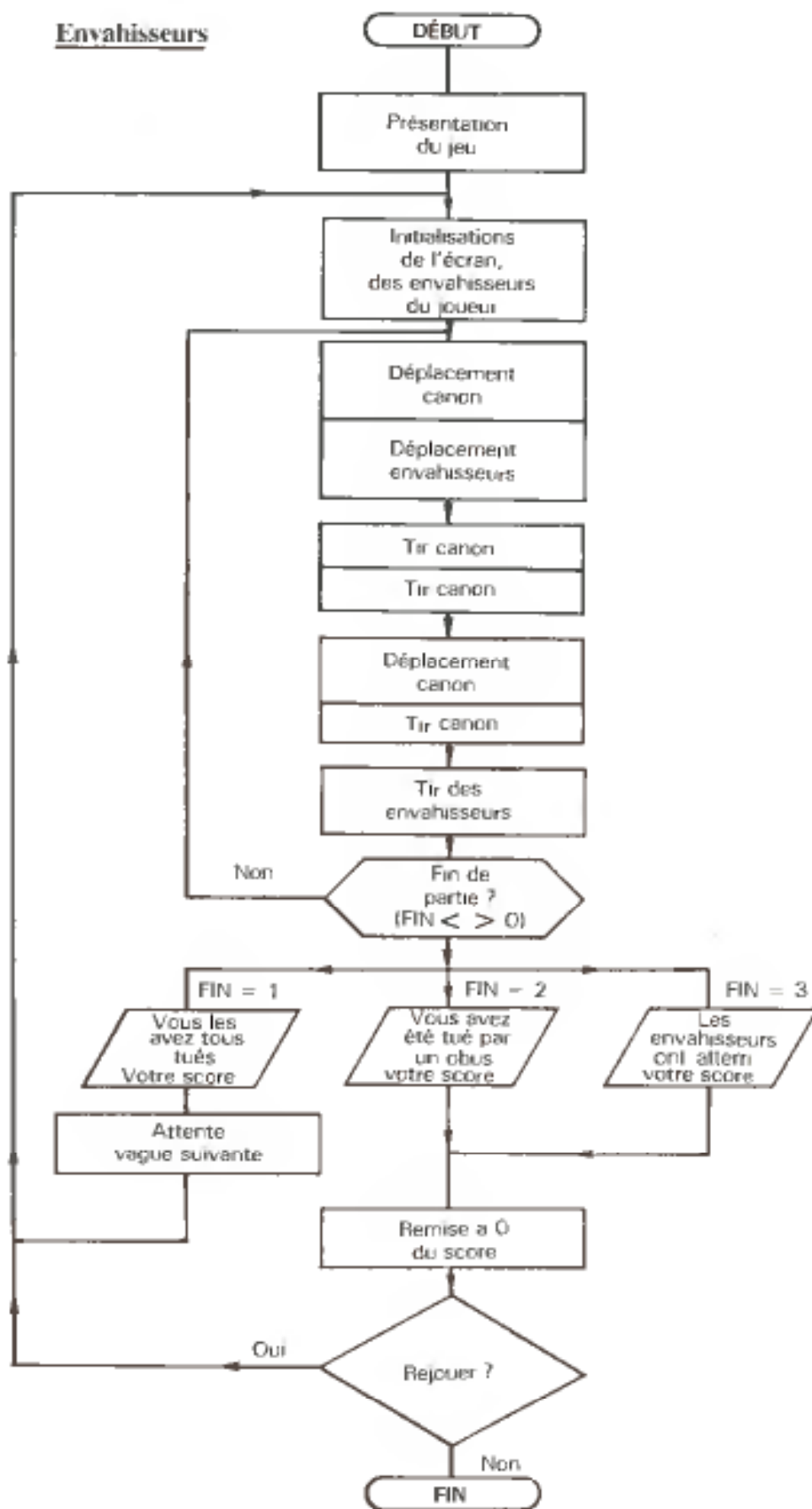
Cela fait aujourd'hui douze ans que vous avez été enlevé par des créatures d'outre-espace. De longues années pendant lesquelles vous avez été durement maltraité. Aujourd'hui, vous avez tué votre gardien et vous vous êtes enfui avec son canon ionique (arme qui se recharge automatiquement). Votre espoir de fuite est nul, mais vous pouvez leur faire payer leur violence en les abattant. Détruisez la première vague et il en apparaîtra une seconde, puis une troisième ... Devant votre hargne, les créatures n'auront que les deux choix suivants : essayer d'atterrir pour vous capturer vivant, ou vous tuer en larguant des bombes à fragmentation. Faites-leur payer le prix de vos souffrances !

2.3.2 Organigramme et étude du programme

Ce programme comporte plusieurs sous-programmes "communs" avec le jeu "Bombardier" décrit précédemment. Nous ne reprendrons donc pas leur analyse en détail.

STRUCTURE

Lignes	40 à	60	Appel des sous-programmes de présentation du jeu et d'initialisations
Lignes	70 à	140	Boucle de jeu
Lignes	200 à	290	Déplacement du canon



Lignes	300 à	600	Tir du canon et test envahisseur touché
Lignes	700 à	800	Affichage des envahisseurs
Lignes	900 à	1000	Déplacement des envahisseurs
Lignes	1100 à	1350	Tir des envahisseurs et test canon touché
Lignes	5000 à	5190	Présentation du jeu
Lignes	5200 à	5470	Initialisations
Lignes	10000 à	10140	Fin de partie

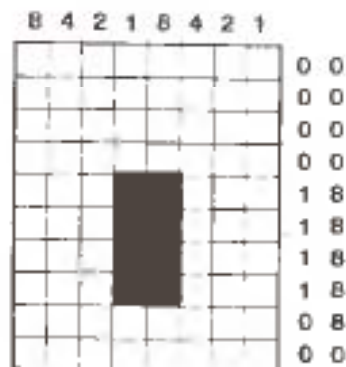
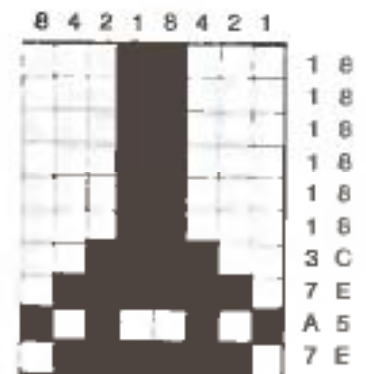
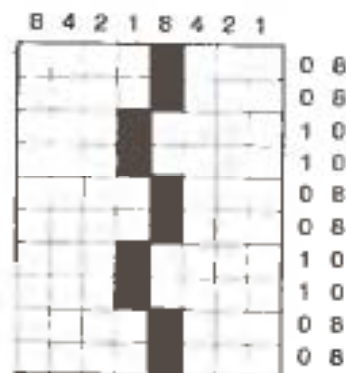
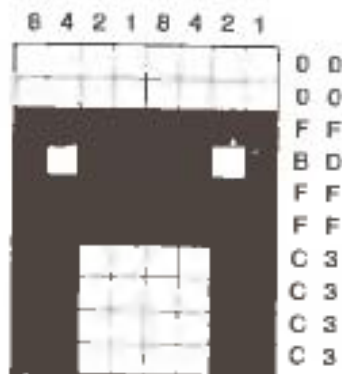
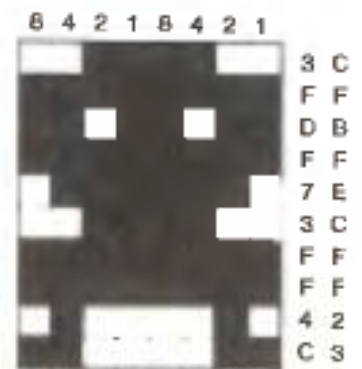
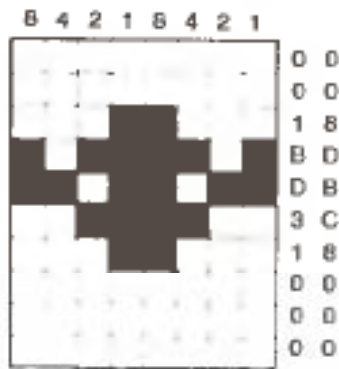
PRINCIPALES VARIABLES

XC, YC	Coordonnées du joueur
XB, YB	Coordonnées de la balle tirée par le joueur
XE, YE	Coordonnées des envahisseurs
XO, YO	Coordonnées des obus successifs
FIN	Indicateur de fin de partie.

CARACTÈRES SPECIAUX

Nous utilisons sept caractères :

- quatre envahisseurs
- un canon
- un type de balle
- un type d'obus



2.3.3 Liste des instructions

```
10 REM *****
20 REM * ENVAHISSEUR *
30 REM *****
40 GOSUB 5000: REM Présentation
50 GOSUB 5210: REM Initialisations
60 FIN=0:STORE
70 REM Boucle de jeu
80 GOSUB 200
90 GOSUB 900:GOSUB 1100
100 GOSUB 200:GOSUB 300
110 SCREEN
120 IF NE=0 THEN FIN=1
130 IF FIN<>0 GOTO 10000
140 GOTO 80
200 REM Déplacement canon
210 EG 0,0,0:CURSORX XC:CURSOR YC
220 PRINT CHR$(CN);
230 IF STICKX(2)=1 THEN XC=XC+1
240 IF STICKX(2)=255 THEN XC=XC-1
250 IF XC<0 THEN XC=0
260 IF XC>39 THEN XC=39
270 EG 1,0,0:CURSORX XC:CURSOR YC
280 PRINT CHR$(CN);
290 RETURN
300 REM Tir du canon et test touché
310 IF TB=1 GOTO 370
320 IF ACTION(2)=0 THEN RETURN
330 REM Tir demandé
340 XB=XC:YB=YC-1:TB=1
350 EG 1,0,0:CURSORX XB:CURSOR YB
360 PRINT CHR$(126);:RETURN
370 REM Avance de la balle
380 FOR I=1 TO 3
390 EG 0,0,0:CURSORX XB:CURSOR YB
400 PRINT CHR$(126);
410 YB=YB-1:IF YB =0 THEN TB=0:RETURN
420 REM Détection de collision
```



```
430 REM avec les envahisseurs
440 IF YB<YE OR YB>YE+3 GOTO 560
450 IF XB<XE OR XB>XE+7 GOTO 560
460 I=(YB-YE)*8+(XB-XE)+1
470 IF ENV(I)=32 THEN RETURN
480 REM Envahisseur touché
490 ENV(I)=32:SOUND 7,100
500 TB=0:NE=NE-1:SR=SR+1
510 L=(I-1)/8:C=INT((XB-XE)/2)
520 LIG(L)=LIG(L)-1:COL(C)=COL(C)-1
530 CURSORX XB:CURSORY YB
540 TX 0,0,0:PRINT CHR$(127)
550 RETURN
560 REM Avance simple de la balle
570 EG 1,0,0:CURSORX XB:CURSORY YB
580 PRINT CHR$(126);
590 NEXT I
600 RETURN
700 REM Affichage des envahisseurs
710 EG CE,0,0
720 FOR I=0 TO 3
730 IF LIG(I)=0 GOTO 790
740 CURSORY YE+I
750 CURSORX XE
760 FOR J=1 TO 8
770 PRINT CHR$(ENV(I*8+J));
780 NEXT J
790 NEXT I
800 RETURN
900 REM Déplacement des envahisseurs
910 CE=0:GOSUB 700
920 XE=XE+DE
930 IF XE<33 AND XE>-1 GOTO 990
940 YE=YE+1:DE=-DE:XE=XE+DE
950 IF YE<13 THEN YE=YE+1
960 FOR I=0 TO 3
970 IF YE>16+I AND LIG(3-I)<>0 THEN FIN
980 NEXT I
990 CE=3:GOSUB 700
1000 RETURN
1100 REM Tir des envahisseurs
1110 IF TE=1 GOTO 1220
```

```

1120 IF YE>15 THEN RETURN
1130 REM Tir
1140 YO=YE+4
1150 C=INT(RND(1)*3.99)
1160 IF COL(C)=0 AND NE>0 GOTO 1150
1170 XO=XE+C*2
1180 TE=1
1190 EG 7,0,0:CURSORX XO:CURSORY YO
1200 PRINT CHR$(124);
1210 RETURN
1220 REM Avance de l'obus
1230 EG 0,0,0:CURSORX XO:CURSORY YO
1240 PRINT CHR$(124);
1250 IF YE>15 THEN TE=0:RETURN
1260 YO=YO+3
1270 IF YO<19 GOTO 1330
1280 IF YO>22 THEN TE=0:RETURN
1290 IF XO<>XC THEN TE=0:RETURN
1300 REM Touché
1310 FIN=2:RETURN
1320 REM Avance simple de l'obus
1330 EG 7,0,0:CURSORX XO:CURSORY YO
1340 PRINT CHR$(124);
1350 RETURN
5000 REM Présentation
5010 INIT 6,6:TX 1,7,0
5020 CURSORX 6:CURSORY 10
5030 PRINT "e n v a h i s s e u r s "
5040 CURSORX 6:CURSORY 11
5050 PRINT "e n v a h i s s e u r s "
5060 TX 4,0,0:PRINT
5070 PRINT "RET pour commencer"
5080 INPUT R$
5090 DIM ENV(32),XO(8),YO(8)
5100 DIM COL(3),LIG(3)
5110 SETEG 032,"0000000000000000000000"
5120 SETEG 120,"187edbfff7e3c18244281"
5130 SETEG 121,"000018bddb3c18000000"
5140 SETEG 122,"3cfffdbfff7e3cffffff42c3"
5150 SETEG 123,"0000ffbdffffc3c3c3c3"
5160 SETEG 124,"08081010080810100808"
5170 SETEG 125,"181818181818183c7ea57e"

```

```
5180 SETEG 126,"00000000181818180000"  
5190 RETURN  
5200 REM Initialisations  
5210 INIT 0,0  
5220 CURSORX 0:CURSORY 20  
5230 DELIM 3,3,0  
5240 CURSORX 0:CURSORY 21  
5250 DELIM 3,3,0  
5260 CURSORX 0:CURSORY 22  
5270 DELIM 3,3,0  
5280 REM Affichage des envahisseurs  
5290 XE=0:YE=0:TE=0:DE=1:NE=16  
5300 FOR I=1 TO 32:ENV(I)=32:NEXT I  
5310 FOR I=1 TO 8STEP 2  
5320 ENV(I)=120:NEXT I  
5330 FOR I=9 TO 15 STEP 2  
5340 ENV(I)=121:NEXT I  
5350 FOR I=17 TO 23 STEP 2  
5360 ENV(I)=122:NEXT I  
5370 FOR I=25 TO 31 STEP 2  
5380 ENV(I)=123:NEXT I  
5390 CE=3:GOSUB 700  
5400 FOR I=0 TO 3  
5410 COL(I)=4:LIG(I)=4  
5420 NEXT I  
5430 REM Affichage initial joueur  
5440 XC=20:YC=19:CN=125:TB=0  
5450 CURSORX XC:CURSORY YC:EG 1,0,0  
5460 PRINT CHR$(CN);  
5470 RETURN  
10000 REM Fin de partie  
10010 INIT 6,6:TX 0,0,0  
10020 ON FIN GOTO 10040, 10070, 10090  
10030 GOTO 10100  
10040 PRINT "Bravo, vous les avez TOUS tués"  
10050 PRINT:PRINT "Votre score est donc de: ";SR  
10060 FOR I=1 TO 10000:NEXT I:GOTO 50  
10070 INIT 7,7:PRINT "Vous avez été tué par un  
obus ennemi...Si, Si!!!"  
10080 GOTO 10100  
10090 PRINT "Les envahisseurs ont atterri, vous  
êtes mort..."
```

```
10100 PRINT :PRINT "Votre score est de: ";SR
10110 CURSORX 1:CURSORY 20:SR=0
10120 INPUT"Voulez-vous rejouer ";R$
10130 IF R$="0" OR R$="o" THEN 50
10140 INIT 6,6:END
```

2.4 SURVIE

2.4.1 Présentation

Vous êtes dans un terrain fermé, à l'intérieur duquel vous dirigez un char. Un missile sillonne le terrain en laissant derrière lui une traînée mortelle. Des bombes surgissent près de vous, pour éclater peu après...

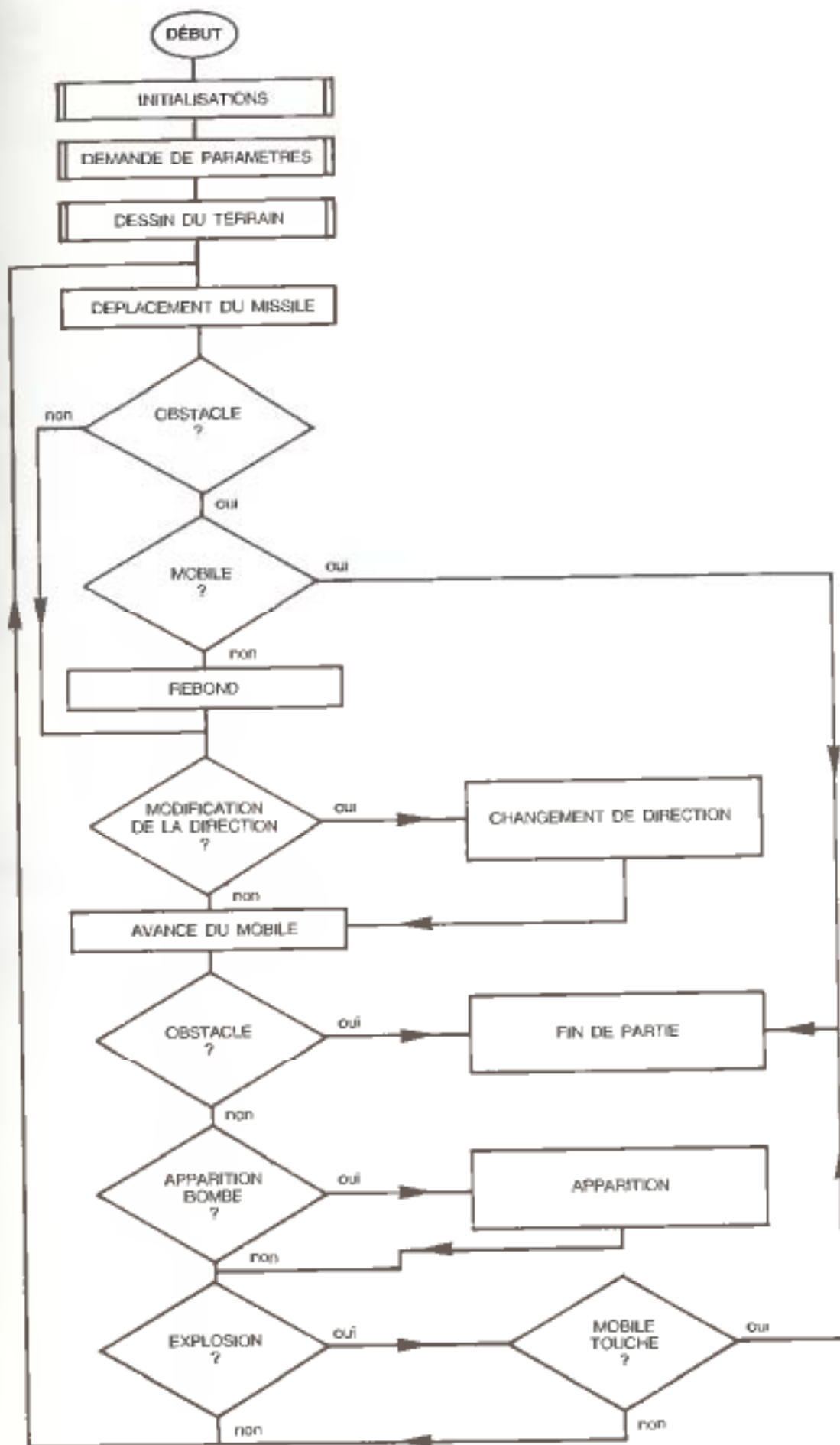
Votre but : survivre le plus longtemps possible.

Vous vous déplacez à l'aide des quatre flèches. Vous placez au début du jeu 2 murs verticaux et 2 murs horizontaux destinés à vous protéger et sur lesquels le missile rebondit.

2.4.2 Organigramme et étude du programme

STRUCTURE

Lignes	40 à	90	Appels des sous-programmes de mode d'emploi, de définition des caractères spéciaux, d'initialisation, de tracé du cadre et des murs
Lignes	100 à	140	Affichages initiaux du joueur et du missile
Lignes	150 à	530	Boucle de jeu
	160 à	250	Déplacement du missile avec rebond sur les murs
	260 à	290	Affichage du missile
	300 à	310	Rencontre missile - joueur



	320		Incrémentation du score
	330 à	350	Effacement du joueur sur l'écran
	360 à	440	Déplacement du joueur à l'aide des flèches
	450 à	460	Test de rencontre d'un obstacle
	470 à	490	Affichage du joueur
	500 à	520	Gestion des bombes
	530		Retour au début de boucle
Lignes	600 à	680	Sous-programme d'apparition d'une bombe
Lignes	700 à	800	Sous-programme d'explosion d'une bombe
Lignes	5000 à	5340	Présentation
	5000 à	5050	Présentation du jeu
	5060 à	5190	Mode d'emploi
	5200 à	5340	Entrée des positions des murs et de la vitesse
Lignes	5400 à	5560	Tracé des cadres
Lignes	5600 à	5720	Initialisations
Lignes	5800 à	5850	Définition des caractères spéciaux
Lignes	5900 à	6020	Tracé des murs
Lignes	10000 à	10110	Fin de partie

PRINCIPALES VARIABLES

NT	Variable définissant le temps de survie (s'incrémente de 1 à chaque tour)
P (42,23)	Tableau contenant l'état de l'écran. Sa valeur est : - 0 si la case est vide

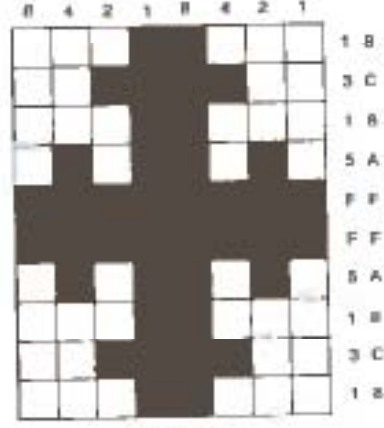
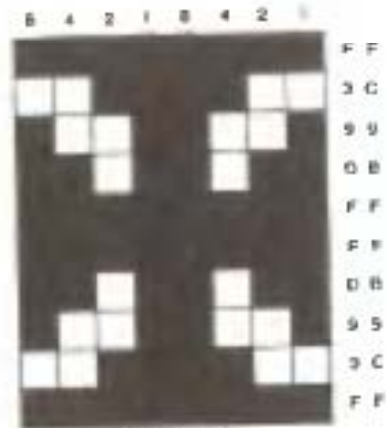
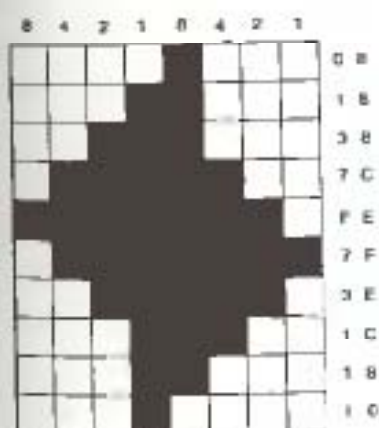
- 1 si la case fait partie d'un mur vertical
- 2 si la case fait partie d'un mur horizontal
- 3 si elle est occupée par un missile
- 4 si elle est occupée par une bombe
- 7 s'il s'agit d'une intersection de murs

XJ, YJ
 XM, YM
 XB, YB
 AM, BM

Coordonnées du joueur
 Coordonnées du missile
 Coordonnées de la dernière bombe
 Pas qui définit l'avance du missile

CARACTERES SPECIAUX

Trois caractères spéciaux ont été définis pour le missile, le char du joueur et les bombes :



2.4.3 Liste des instructions

```
10 REM *****
20 REM SURVIE *
30 REM *****
40 GOSUB 5000: REM Mode d'emploi
50 GOSUB 5810: REM Caractères spéciaux
60 GOSUB 5600: REM Initialisations
70 GOSUB 5400: REM Tracé du cadre
80 X=X1:Y=Y1:GOSUB 5900:REM Tracé mur
90 X=X2:Y=Y2:GOSUB 5900:REM Tracé mur
100 REM Affichage joueur et missile
110 CURSORX XJ:CURSORY YJ
120 ET 0,0,0:PRINT CHR$(JOU);
130 CURSORX XM:CURSORY YM
140 ET 1,0,0:PRINT CHR$(MIS);
150 REM Boucle de jeu
160 REM Déplacement du missile
170 ZZ=0
180 REM Rencontre d'un mur horizontal?
190 IF P(XM+AM,YM+BM)=1THEN AM=-AM:ZZ=1
200 REM Rencontre d'un mur vertical?
210 IF P(XM+AM,YM+BM)=2THEN BM=-BM:ZZ=1
220 REM Rencontre d'un coin entre murs?
230 IF P(XM+AM,YM+BM)=7THEN AM=-AM:BM=-BM
240 XM=XM+AM:YM=YM+BM
250 IF ZZ=1 THEN GOSUB 6030
260 REM Affichage du missile
270 CURSORX XM:CURSORY YM
280 ET 1,0,0:PRINT CHR$(MIS);
290 P(XM,YM)=3
300 REM Rencontre missile - joueur
310 IF XM=XJ AND YM=YJ GOTO 10000
320 NT=NT+1
330 REM Effacement du joueur
340 CURSORX XJ:CURSORY YJ
350 ET 3,0,0:PRINT CHR$(127);
360 REM Déplacement du joueur
```

```

370 FOR I=1 TO V
380 A=KEY(0)
390 NEXT I
400 IF A=7 THEN AJ=1:BJ=0
410 IF A=8 THEN AJ=-1:BJ=0
420 IF A=9 THEN BJ=-1:AJ=0
430 IF A=10 THEN BJ=1:AJ=0
440 XJ=XJ+AJ:YJ=YJ+BJ
450 REM Test rencontre d'un obstacle
460 IF P(XJ,YJ)<>0 THEN 10000
470 REM Affichage du joueur
480 ET 0,0,0:CURSORX XJ:CURSORY YJ
490 PRINT CHR$(JOU);
500 REM Bombes
510 IF INT(NT/30)=NT/30 THEN GOSUB 600:GOTO 150
520 IF INT((NT-15)/30)=(NT-15)/30 THEN GOSUB 700
530 GOTO 150
600 REM Apparition d'une bombe
610 X=INT(RND(1)*4)-2:Y=INT(RND(1)*4)-2
620 D=2:IF NT>500 THEN D=1
630 XB=XJ+X+D:YB=YJ+Y+D
640 IF P(XB,YB)<>0 THEN 610
650 IF XB<3 OR XB>36 OR YB<2 OR YB>20 THEN 610
660 ET 1,0,0:CURSORX XB:CURSORY YB
670 PRINT CHR$(113);:P(XB,YB)=4
680 RETURN
700 REM Explosion d'une bombe
710 ET 1,0,0
720 FOR I=XB-1 TO XB+1
730 FOR J=YB-1 TO YB+1
740 CURSORX I:CURSORY J
750 IF I=XJ AND J=YJ THEN 10000
760 IF P(I,J)<>0 AND P(I,J)<>4 THEN 780
770 PRINT CHR$(113);:P(I,J)=4
780 NEXT J:NEXT I
790 SOUND 112,6,24
800 RETURN
5000 REM Présentation - Mode d'emploi
5010 INIT 3,1:TX 4,7,0
5020 CURSORX 8:CURSORY 0
5030 PRINT "S u r v i e "
5040 CURSORX 8:CURSORY 1

```

```
5050 PRINT "S u r v i e "
5060 TX 0,0,0:CURSOR 3
5070 PRINT "Vous êtes dans un terrain fermé, à"
5080 PRINT "l'intérieur duquel vous dirigez un char"
5090 PRINT "Un missile sillonne le terrain en lais"
5100 PRINT "sant derrière lui une trainée mortelle"
5110 PRINT "Des bombes surgissent près de vous"
5120 PRINT "pour éclater peu après."
5130 PRINT "Votre but : survivre le plus longtemps"
5140 PRINT "possible.":PRINT
5150 PRINT "Vous dirigez votre char à l'aide des 4"
5160 PRINT "flèches. Vous placez au départ 4 murs"
5170 PRINT "destinés à vous protéger: 2 verticaux"
5180 PRINT "et 2 horizontaux."
5190 PRINT:PRINT "Choisissez leurs positions:"
5200 PRINT "(Entrez 2 coordonnées séparées par une"
5210 PRINT "virgule)":PRINT
5220 CURSOR 22
5230 PRINT "2 murs verticaux: 14<x<35 ";
5240 INPUT X1,X2
5250 IF X1<15 OR X2<15 OR X1>34 OR X2>34 GOTO 5220
5260 CURSOR 23
5270 PRINT "2 murs horizontaux: 4<y<18 ";
5280 INPUT Y1,Y2
5290 IF Y1<5 OR Y2<5 OR Y1>17 OR Y2>17 GOTO 5260
5300 CURSOR 24
5310 PRINT:PRINT "Choisissez votre vitesse:"
5320 INPUT "1<v<200 (1 est la plus rapide)";V
5330 IF V<1 OR V>200 GOTO 5300
5340 RETURN
5400 REM Tracé du cadre
5410 INIT 3,1 : ET 4,0,0
5420 FOR I=1 TO 21
5430 CURSOR I
5440 CURSORX 2:PRINT CHR$(110);
5450 CURSORX 37:PRINT CHR$(110);
5460 P(2,I)=1:P(37,I)=1
5470 NEXT I
5480 FOR I=2 TO 37
5490 P(I,1)=2:P(I,21)=2:NEXT I
5500 CURSOR 1:CURSORX 2:DELIM 4,4,0
5510 CURSOR 21:CURSORX 2:DELIM 4,4,0
```

```

5520 CURSORY 1:CURSORX 38:DELIM 3,3,0
5530 CURSORY 21:CURSORX 38:DELIM 3,3,0
5540 P(2,1)=7:P(2,21)=7
5550 P(37,1)=7:P(37,21)=7
5560 RETURN
5600 REM initialisations
5610 DIM P(42,23)
5620 FOR I=1 TO 40:FOR J=1 TO 22
5630 P(I,J)=0
5640 NEXT I
5650 MUR=110:JOU=111:MIS=112:BOMB=113
5660 NT=16:XJ=35:YJ=18
5670 XM=INT(RND(1)*10)+3
5680 IF XM=X1 OR XM=X2 GOTO 5670
5690 YM=INT(RND(1)*8)+2
5700 IF YM=Y1 OR YM=Y2 GOTO 5690
5710 AM=1:BM=1:P(XM,YM)=3
5720 RETURN
5800 REM Caractères spéciaux
5810 SETET 110,"FFFFFFFFFFFFFFFFFFFFFF"
5820 SETET 111,"FF3C99DBFFFFFFDB9933FF"
5830 SETET 112,"0818387CFE7F3E1C1810"
5840 SETET 113,"183C185AFFFF5A183C18"
5850 RETURN
5900 REM Tracé des murs
5910 ET 4,0,0
5920 FOR I=5 TO 15
5930 P(X,I)=P(X,I)+1
5940 IF P(X,I)>2 THEN P(X,I)=7
5950 P(I,Y)=P(I,Y)+2
5960 IF P(I,Y)>2 THEN P(I,Y)=7
5970 CURSORX X:CURSORY I
5980 PRINT CHR$(MUR);
5990 CURSORX I:CURSORY Y
6000 PRINT CHR$(MUR);
6010 NEXT I
6020 RETURN
6030 T=RND(1)*100
6040 IF (P(XM+1,YM)=0 OR P(XM+1,YM)=4) AND T>90
    THEN XM=XM+1
6050 RETURN
10000 REM Fin de partie

```

```
10010 SOUND 122,16,55
10020 PLAY "AAAEEDDDDDAACCCCEEDDDDDDDDD"
10030 INIT 2,0:TX 0,0,0
10040 PRINT "Vous êtes mort ! ! !"
10050 PRINT:PRINT
10060 PRINT "Votre temps de survie est de ";NT
10070 PRINT:PRINT
10080 INPUT "Autre partie ";R$
10090 IF R$="0" OR R$="o" THEN RUN
10100 INIT 6,6:PRINT "Au revoir !"
10110 END
```

2.5 COLLIMATEUR

2.5.1 Présentation

L'action se passe au 21ème siècle dans une partie de l'espace galactique. Vous êtes seul à bord de votre patrouilleur lorsque vous surprenez un groupe de pirates interstellaires. Malheureusement pour eux, leurs armes sont sans effet sur votre blindage et il ne leur reste que la stratégie du sauve-qui-peut. Abattez-les sans sommation, et poursuivez-les aussi longtemps que votre carburant le permet (équivalence de 30 tirs). Pour en abattre un, il faut placer votre collimateur (flèches curseur) sur le vaisseau ennemi et faire feu (barre d'espace). N'ayez aucun scrupule et massacrez ces vermines...

2.5.2 Organigramme et étude du programme

Ce programme réalise des accès directs en mémoire d'écran pour un affichage plus rapide des barres constituant le collimateur. Pour chaque caractère affiché, il accède à 2 octets en mémoire d'écran : le code et les attributs du caractère (coordonnées X et Y) :

L'octet de code est à l'adresse $16384+2*X+80*Y$

L'octet contenant les attributs suit en mémoire l'octet de code.

Le code utilisé est 127 (caractère plein), les attributs sont la couleur rouge sur fond noir (=1).

STRUCTURE

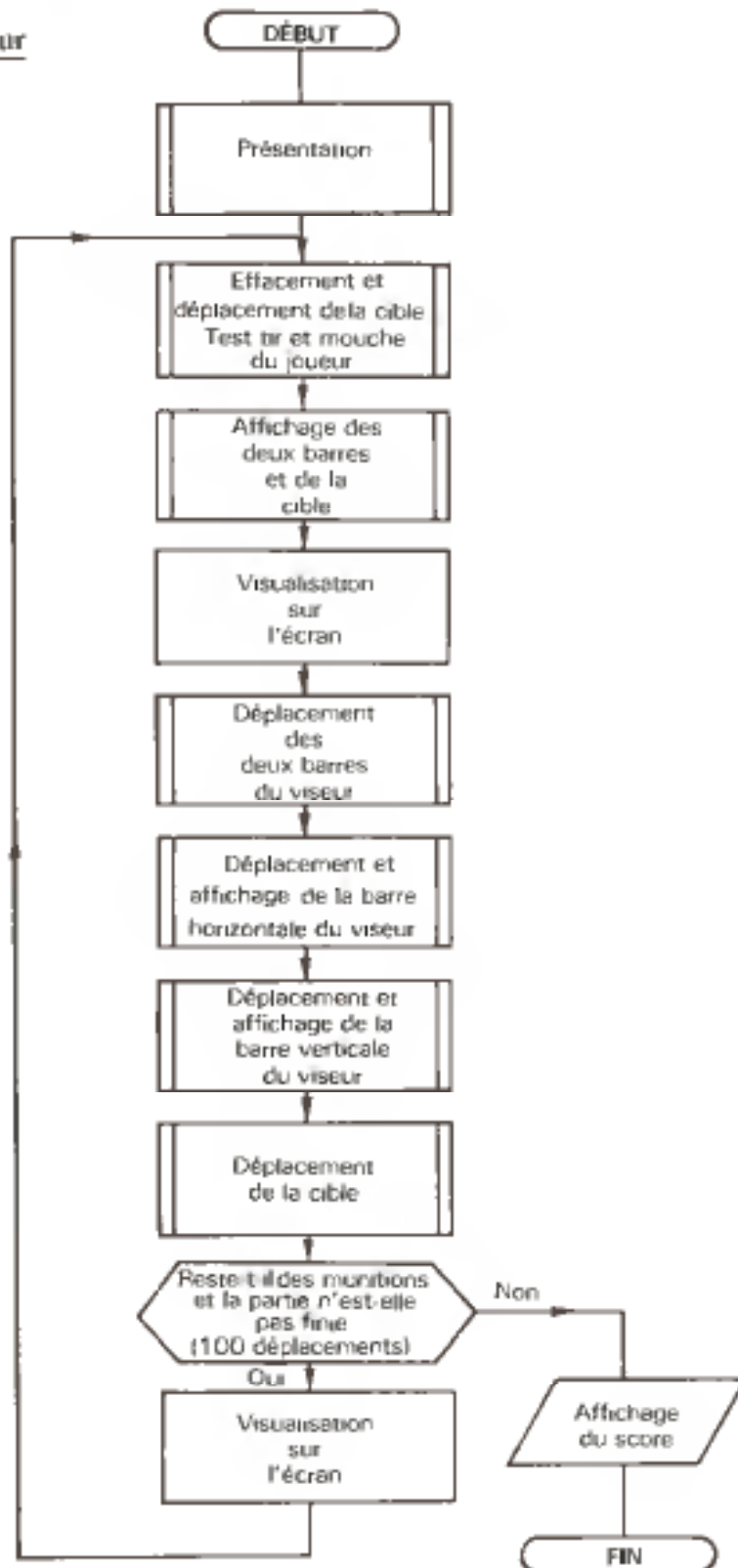
Lignes	10 à	50	Initialisations
Lignes	50 à	140	Boucle principale et test de munitions
Lignes	200 à	270	Tir et test de choc
Lignes	280 à	390	Message de victoire et affichage du score
Lignes	500 à	570	Déplacement barre verticale
Lignes	800 à	870	Déplacement barre horizontale
Lignes	1000 à	1070	Tracé barre horizontale
Lignes	1100 à	1160	Effacement barre horizontale
Lignes	2000 à	2070	Tracé d'une barre verticale
Lignes	2100 à	2160	Effacement barre verticale
Lignes	2300 à	2370	Déplacement de la cible
Lignes	2500 à	2660	Affichage cible
Ligne	2700		Effacement cible
Lignes	5000 à	5150	Présentation
Lignes	10000 à	10150	Fin de partie

REMARQUE : Remplacez 2 par 0 (ou 1) dans les instructions STICKX, STICKY et ACTION. Vous pourrez ainsi utiliser le manche à balai de gauche (droite) avec ce jeu

PRINCIPALES VARIABLES

XB, YB	Coordonnées des barres du viseur
HB, VB	Déplacements des barres du viseur
XC, YC	Coordonnées de la cible
HC, LC	Déplacements de la cible
SY	Sens de déplacement horizontal de la cible
SR	Score du joueur
NOGUPS	Nombre de tirs réalisés
NMUNIT	Nombre de munitions disponibles

Collimateur



2.5.3 Liste des instructions

```
10 REM *****
20 REM * Collimateur *
30 REM *****
40 GOSUB 5000
50 FOR N=1 TO 100
60 DISPLAY1
70 GOSUB 2700:GOSUB 1100:GOSUB 2100
80 GOSUB 500:GOSUB 800
90 GOSUB 1000:GOSUB 2000
100 GOSUB 2300:GOSUB 2500
110 GOSUB 200
120 IF NCOUPS=NMUNIT GOTO 10000
130 TP=TP+1:NEXT N
140 GOTO 10000
200 REM Test de tir et de choc
210 TIR=KEY(0)
220 IF TIR<>32 THEN RETURN
230 NCOUPS=NCOUPS+1
240 IF XB<XC+1 THEN RETURN
250 IF YB<YC+1 THEN RETURN
260 IF XB>XC+LC-1 THEN RETURN
270 IF YB>YC+HC-1 THEN RETURN
280 SR=SR+1
290 INIT 3,3:TX 1,0,0:CUSDRY 10
300 CURSORX 4:PRINT "Félicitations, "
310 PRINT "votre score est de ";SR; " ci
ble";
320 IF SR>1 THEN PRINT "s";
330 PRINT " ennemie";
340 IF SR>1 THEN PRINT "s"
350 FOR I=1 TO 3000:NEXT I
360 XC=INT(RND(1)*33)
370 YC=INT(RND(1)*16)
380 INIT 0,0
390 RETURN
500 REM Déplacer la barre verticale
510 VB=0
520 IF STICKY(2)=1 THEN VB=2
```

```
530 IF STICKY(2)=255 THEN VB=-2
540 YB=YB+VB
550 IF YB<0 THEN YB=0
560 IF YB>23 THEN YB=23
570 RETURN
800 REM Déplacer la barre horizontale
810 HB=0
820 IF STICKX(2)=1 THEN HB=2
830 IF STICKX(2)=255 THEN HB=-2
840 XB=XB+HB
850 IF XB<0 THEN XB=0
860 IF XB>39 THEN XB=39
870 RETURN
1000 REM Tracer la barre horizontale
1010 PMC=16384+80*YB
1020 FOR X=0 TO 39
1030 POKE PMC,127
1040 POKE PMC+1,1
1050 PMC=PMC+2
1060 NEXT X
1070 RETURN
1100 REM Effacer la barre horizontale
1110 PMC=16384+80*YB
1120 FOR X=0 TO 39
1130 POKE PMC+1,0
1140 PMC=PMC+2
1150 NEXT X
1160 RETURN
2000 REM Tracer la barre verticale
2010 PMC=16384+2*XB
2020 FOR Y=0 TO 23
2030 POKE PMC,127
2040 POKE PMC+1,1
2050 PMC=PMC+80
2060 NEXT Y
2070 RETURN
2100 REM Effacer la barre verticale
2110 PMC=16384+2*XB
2120 FOR Y=0 TO 23
2130 POKE PMC+1,0
2140 PMC=PMC+80
2150 NEXT Y
```

```
2160 RETURN
2300 REM Déplacement de la cible
2310 XC=XC+INT(RND(1)*3)
2320 YC=YC+INT(RND(1)*3)*SY
2330 IF XC<2 THEN XC=30
2340 IF XC>33 THEN XC=4
2350 IF YC<0 THEN YC=0:SY=-SY
2360 IF YC>20 THEN YC=20:SY=-SY
2370 RETURN
2500 REM Affichage de la cible
2510 PMC=16384+80*YC+2*XC
2520 FOR P=PMC TO PMC+80*HC STEP80
2530 POKE P,127:POKE P+1,3:NEXT P
2540 IF XC+LC>39 GOTO 2590
2550 PMC=PMC+2*LC
2560 FOR P=PMC TO PMC+80*HC STEP80
2570 POKE P,127:POKE P+1,3
2580 NEXT P
2590 PMC=16384+80*YC+2*XC
2600 FOR P=PMC TO PMC+2*LC STEP 2
2610 POKE P,127:POKE P+1,3:NEXT P
2620 IF YC+HC>39 GOTO 2660
2630 PMC=PMC+80*HC
2640 FOR P=PMC TO PMC+2*LC STEP 2
2650 POKE P,127:POKE P+1,3:NEXT P
2660 RETURN
2700 REM Effacement de la cible
2710 PMC=16384+80*YC+2*XC
2720 FOR P=PMC TO PMC+80*HC STEP80
2730 POKE P,127:POKE P+1,0:NEXT P
2740 IF XC+LC>39 GOTO 2790
2750 PMC=PMC+2*LC
2760 FOR P=PMC TO PMC+80*HC STEP80
2770 POKE P,127:POKE P+1,0
2780 NEXT P
2790 PMC=16384+80*YC+2*XC
2800 FOR P=PMC TO PMC+2*LC STEP 2
2810 POKE P,127:POKE P+1,0:NEXT P
2820 IF YC+HC>39 GOTO 2860
2830 PMC=PMC+80*HC
2840 FOR P=PMC TO PMC+2*LC STEP 2
2850 POKE P,127:POKE P+1,0:NEXT P
```

```
2860 RETURN
5000 REM Présentation
5010 INIT 4,4:TX 2,5,0
5020 CURSORX 8:CURSORY 8
5030 PRINT "C o l l i m a t e u r "
5040 CURSORX 8:CURSORY 9
5050 PRINT "C o l l i m a t e u r "
5060 TX0,0,0:PRINT :PRINT
5070 PRINT "RET pour commencer"
5080 INPUT R$
5090 REM Initialisations
5100 XB=20:YB=9:NCOUPS=0:NMUNIT=10
5110 XC=INT(RND(1)*33)
5120 YC=INT(RND(1)*16)
5130 INIT 0,0:GOSUB 1000:GOSUB 2000
5140 SY=1:LC=6:HC=3:GOSUB 2500
5150 RETURN
10000 REM fin de partie
10010 REM Plus de munitions
10020 INIT3,3:CURSORX 4:CURSORY 8
10030 TX 0,0,0
10040 IF NCOUPS=NMUNIT GOTO 10060
10050 PRINT "Temps écoulé":GOTO 10070
10060 PRINT "Vous n'avez plus de munitions"
10070 PRINT :PRINT
10080 PRINT "Votre score est de ";SR;" cible";:IF SR>1 THEN PRINT "s";
10090 PRINT " ennemie";:IF SR>1 THEN PRINT "s"
10100 CURSORX 1:CURSORY 20
10110 PRINT "Une autre partie (o/n)";
10120 INPUT R$
10130 IF R$="0" OR R$="o" THEN RUN
10140 INIT 6,6
10150 END
```

2.6 LABYRINTHE

2.6.1 Présentation

Voici un jeu qui va vous permettre de tester votre mémoire visuelle et votre rapidité de réaction.

Vous vous trouvez, sans la moindre visibilité, à l'intérieur d'un labyrinthe dans lequel sont cachés trois trésors dont vous connaissez l'emplacement mais non le chemin qui y mène.

Votre but est d'aller chercher ces trésors le plus rapidement possible en tenant compte des règles suivantes :

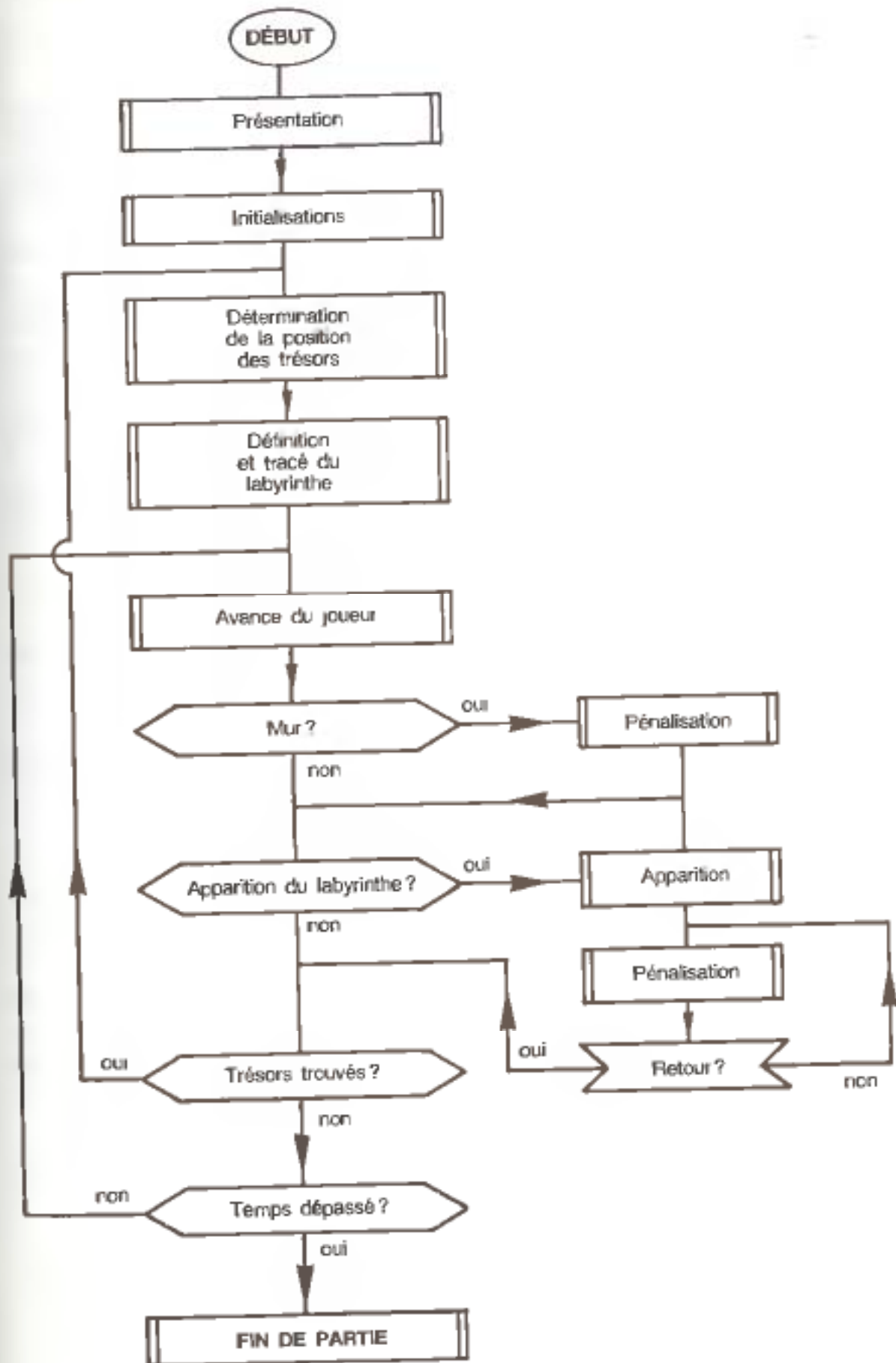
- chaque fois que vous rencontrez un mur, une pénalité vous est comptée.
- vous pouvez avoir accès au plan du labyrinthe pendant le temps que vous souhaitez, et à tout moment, en sachant toutefois que cela vous cause une grosse perte de temps.

A partir de là, c'est à vous de combiner rapidité et efficacité pour vous enrichir le plus possible.

Dès que vous avez atteint les trois trésors, et s'il vous reste du temps, un autre labyrinthe est créé et le jeu continue.

2.6.2 Organigramme et étude du programme

Lignes	40 à	80	Appel des sous-programmes (présentation, initialisation, tracé...)
--------	------	----	--



Lignes	90 à	130	Placement du joueur
Lignes	140 à	150	Boucle de jeu
Lignes	200 à	440	Déplacement du joueur, et test de réflexion sur les murs.
	210		Lecture du clavier - Affichage des trésors.
	220 à	250	Déplacement du joueur selon les touches curseur
	260		Si le joueur appuie sur RET, les murs sont affichés
	270		Pas de touche appuyée
	280 à	440	Test de collision sur les murs
Lignes	500 à	640	Apparition du labyrinthe (sur RET)
Lignes	700 à	790	Mise en place des trésors (tirage au sort)
Lignes	800 à	890	Affichache des trésors
	Lignes 900 à	1000	Test de découverte d'un trésor
Lignes	1100 à	1230	Tracé du cadre
Lignes	1300 à	1470	Délimitation des murs et des fermetures
Lignes	1500 à	1690	Tracé du labyrinthe
Lignes	1700 à	1770	Algorithme associé au tracé du labyrinthe
Lignes	1800 à	1920	Choix d'un parcours entre le joueur et le trésor
Lignes	5000 à	5310	Présentation, mode d'emploi, caractères spéciaux

Lignes 6000 à 6090 Initialisation des variables

Lignes 10000 à 10100 Fin de partie

PRINCIPALES VARIABLES

CO, CC	Déterminent les couleurs du labyrinthe.
P	Tableau servant à l'algorithme de tracé du labyrinthe.
V, H	Chaque case est caractérisée par son mur vertical de droite et son mur horizontal du bas. Ces 2 tableaux enregistrent s'il y a ou non des murs.
VD, HD	Ont la même fonction que V et H mais concernent seulement les murs dans lesquels le joueur est rentré.
LC	Variable servant à définir si, lors du tracé, les murs seront visibles ou non.
XJ, YJ	Coordonnées du joueur.
T	Tableau contenant les coordonnées des trésors.

CARACTERES SPECIAUX

Nous utilisons ici les caractères spéciaux du jeu "Fantôme" que vous trouverez en page 205 du présent volume.

2.6.3 Liste des instructions

```

10 REM *****
20 REM * Labyrinthe *
30 REM *****
40 GOSUB 5000
50 GOSUB 6000
60 GOSUB 700
70 GOSUB 1100:GOSUB 1300:GOSUB 1880:GOSU
B 1500
80 GOSUB 800
90 ET 1,0,0
100 LC=0
110 CURSORX 25:CURSORY 22:PRINT CHR$(125
);
120 XJ=10:YJ=10
130 FOR I=0 TO 11:V(10,I)=1:V(0,I)=1:H(I
,10)=1:H(I,0)=1:NEXT I
140 GOSUB 200
150 GOTO 140
200 REM Déplacement du joueur
210 AA=KEY(0):MM=0:GOSUB 800:TE=TE+1
220 IF AA=7 THEN AJ=1-V(XJ,YJ):BJ=0:ZE=0
:ZZ=0
230 IF AA=8 THEN AJ=V(XJ-1,YJ)-1:BJ=0:ZE
=-1:ZZ=0
240 IF AA=9 THEN BJ=H(XJ,YJ-1)-1:AJ=0:ZE
=0:ZZ=-1
250 IF AA=10 THEN BJ=1-H(XJ,YJ):AJ=0:ZE=
0:ZZ=0
260 IF AA=13 THEN GOSUB 500:RETURN
270 IF AA=0 THEN RETURN
280 R=110:I=XJ+ZE:J=YJ+ZZ:CC=1
290 IF AJ<>0 OR BJ<>0 THEN 340
300 TE=TE+100:SOUND 20,4,15:MM=1:IF AA<9
THEN VD(I,J)=1
310 IF AA>8 THEN HD(I,J)=1
320 HH(I,J)=HD(I,J):VV(I,J)=VD(I,J)
330 GOSUB 1540:GOTO 410
340 IF VD(XJ,YJ)=1 THEN R=111

```

```
350 IF HD(XJ,YJ)=1 THEN R=112
360 IF HD(XJ,YJ)=1 AND VD(XJ,YJ)=1 THEN
R=113
370 ET CC,0,0
380 CURSORX 2*XJ+5:CURSORY 2*YJ+2:PRINT
CHR$(R);
390 XJ=XJ+AJ:YJ=YJ+BJ
400 GOSUB 900:IF LJ=0 GOTO 60
410 ET 1,0,0
420 CURSORX 2*XJ+5:CURSORY 2*YJ+2:PRINT
CHR$(125);
430 IF TE>3000 THEN 10000
440 RETURN
500 REM Apparition du labyrinthe
510 CC=0:TE=TE+300
520 GOSUB 1500
530 CURSORX 2*XJ+5:CURSORY 2*YJ+2
540 ET 1,0,0:PRINT CHR$(125);
550 AA=KEY(0):TE=TE+2:IF AA=0 GOTO 550
560 MM=1:CC=1
570 FOR I=1 TO 10:FOR J=1 TO 10
580 VV(I,J)=VD(I,J):HH(I,J)=HD(I,J)
590 GOSUB 1540
600 NEXT J:NEXT I
610 GOSUB 800
620 CURSORX 2*XJ+5:CURSORY 2*YJ+2
630 ET 1,0,0:PRINT CHR$(125);
640 RETURN
700 REM Mise en place des trésors
710 FOR K=1 TO 3
720 W=1:Q=1
730 X=INT(RND(1)*4):IF X=4 THEN X=3
740 Y=INT(RND(1)*4):IF Y=4 THEN Y=3
750 IF K=2 THEN W=7
760 IF K=3 THEN Q=7
770 T(K,1)=W+X:T(K,2)=Q+Y
780 NEXT K
790 RETURN
800 REM Affichage des trésors
810 ET 4,0,0
820 FOR K=1 TO 3
830 I=T(K,1):J=T(K,2):IF I=0 THEN 800
```

```
840 CURSORX 2*I+5:CURSORY 2*J+1:PRINT CH
R$(124);
850 CURSORX 2*I+4:PRINT CHR$(124);
860 CURSORY 2*J+2:PRINT CHR$(124);
870 CURSORX 2*I+4:PRINT CHR$(124);
880 NEXT K
890 RETURN
900 REM Test de découverte du trésor
910 FOR L=1 TO 3
920 IF XJ<>T(L,1) OR YJ<>T(L,2) GOTO 940
930 PLAY "AAARREEERRAAARREEEEE":TR=TR+1
:T(L,1)=0:T(L,2)=0:TT=1
940 NEXT L
950 LJ=0
960 FOR L=1 TO 3
970 LJ=LJ+T(L,1)
980 NEXT L
990 IF LJ<>0 THEN RETURN
1000 PLAY "AAAEERAAAAEERAADDDDDRRREEECCD
RRRAAAAAAAA":GOSUB 6020:RETURN
1100 REM Tracé du cadre
1110 INIT 7,1:ET 0,0,0
1120 FOR I=5 TO 26
1130 CURSORX I:CURSORY 2
1140 PRINT CHR$(124);
1150 CURSORX I:CURSORY 23
1160 PRINT CHR$(124);
1170 NEXT I
1180 FOR I=3 TO 22
1190 CURSORY I
1200 CURSORX 5:PRINT CHR$(124);
1210 CURSORX 26:PRINT CHR$(124);
1220 NEXT I
1230 RETURN
1300 REM Délimitation des murs
1310 FOR I=1 TO 10
1320 FOR J=1 TO 10
1330 IF P(I,J)=0 THEN KK=KK+1:GOSUB 1360
1340 NEXT J
1350 NEXT I
1360 REM Mise en place des fermetures
1370 A=I:B=J:P(I,J)=1
```

```
1380 X=INT(RND(1)*4):IF X=4 THEN X=3
1390 E=0:Z=0
1400 IF X=0 AND A<10 THEN V(A,B)=0:E=1:A
=A+1
1410 IF X=1 AND A>1 THEN A=A-1:V(A,B)=0:
E=-1
1420 IF X=2 AND B>1 THEN B=B-1:H(A,B)=0:
Z=-1
1430 IF X=3 AND B<10 THEN H(A,B)=0:B=B+1
:Z=1
1440 IF E=0 AND Z=0 THEN 1380
1450 IF P(A,B)=KK THEN GOSUB 1710:IF TT=
0 THEN RETURN
1460 P(A,B)=KK
1470 GOTO 1380
1500 REM Tracé du labyrinthe
1510 FOR I=1 TO 10
1520 FOR J=1 TO 10
1530 HH(I,J)=H(I,J):VV(I,J)=V(I,J)
1540 A=111:B=112:C=110:C1=C0:C2=C0
1550 C3=C0:IF LC=1 THEN GOTO 1600
1560 IF I<1 OR J<1 THEN RETURN
1570 IF VV(I,J)=1 THEN C1=CC:C=111:C3=CC
1580 IF HH(I,J)=1 THEN C2=CC:C=112:C3=CC
1590 IF VV(I,J)=1 AND HH(I,J)=1 THEN C3=
CC:C=113
1600 ET C1,0,0:CURSORX 2*I+5
1610 CURSORY 2*J+1:PRINT CHR$(A);
1620 CURSORX 2*I+4:PRINT CHR$(110);
1630 ET C2,0,0:CURSORX 2*I+4
1640 CURSORY 2*J+2:PRINT CHR$(B);
1650 ET C3,0,0
1660 CURSORX 2*I+5:PRINT CHR$(C);
1670 IF MM=1 THEN RETURN
1680 NEXT J:NEXT I
1690 RETURN
1700 REM Algorithme de tracé-labyrinthe
1710 A=A-E:B=B-Z
1720 IF P(A+1,B)=0 AND A<11 THEN A=A+1:T
T=1:V(A-1,B)=0:RETURN
1730 IF P(A-1,B)=0 AND A>0 THEN A=A-1:TT
=1:V(A,B)=0:RETURN
```

```

1740 IF P(A,B-1)=0 AND B>0 THEN B=B-1:TT
=1:H(A,B)=0:RETURN
1750 IF P(A,B+1)=0 AND B<11 THEN B=B+1:T
T=1:H(A,B-1)=0:RETURN
1760 TT=0
1770 RETURN
1800 REM Choix d'un parcours
1810 I=10:J=10:IF U=10 OR V=10 THEN 1850
1820 X=INT(RND(1)*2)
1830 IF X=0 THEN I=I-1:V(I,J)=0:GOTO 1850
1840 J=J-1:H(I,J)=0
1850 IF I=U THEN FOR K=V TO J-1:H(U,K)=0
:NEXT K:RETURN
1860 IF J=V THEN FOR K=U TO I-1:V(K,V)=0
:NEXT K:RETURN
1870 GOTO 1820
1880 FOR L=1 TO 3
1890 U=T(L,1):V=T(L,2)
1900 GOSUB 1800
1910 NEXT L
1920 RETURN
5000 REM Présentation
5010 INIT 7,0:TX 4,7,0
5020 CURSORX 8:CURSORY 0
5030 PRINT "L a b y r i n t h e "
5040 CURSORX 8:CURSORY 1
5050 PRINT "L a b y r i n t h e "
5060 TX 0,0,0:PRINT :PRINT
5070 REM Mode d'emploi
5080 PLAY "AAARRRCCDDEEEEERRAAARARAREEEEE
EEE"
5090 PRINT "Vous êtes dans un labyrinthe
dans"
5100 PRINT "lequel vous devez atteindre
trois"
5110 PRINT "trésors.":PRINT
5120 PRINT "Votre temps est limité, vous
ne voyez"
5130 PRINT "pas les murs et vous êtes pé
nalisés"
5140 PRINT "quand vous leur rentrez deda
ns.":PRINT

```

```
5150 PRINT "Vous manoeuvrez grâce aux 4
flèches de"
5160 PRINT "direction. Si vous appuyez s
ur RET, le"
5170 PRINT "labyrinthe apparait. Vous ét
es alors"
5180 PRINT "pénalisé. Dès que vous avez
atteint"
5190 PRINT "les 3 trésors, un autre laby
rinthe est"
5200 PRINT "constitué. A vous de ramener
le plus "
5210 PRINT "de trésors possible !"
5220 PRINT :PRINT "RET pour commencer"
5230 INPUT C$
5240 REM Caractères spéciaux
5250 SETET 110,"0000000000000000000000"
5260 SETET 111,"01010101010101010101"
5270 SETET 112,"00000000000000000000FF"
5280 SETET 113,"01010101010101010101FF"
5290 SETET 124,"FFFFFFFFFFFFFFFFFFFF"
5300 SETET 125,"1C361C087F5D5D1C3663"
5310 RETURN
6000 REM Initialisation des variables
6010 DIM P(12,12),V(11,11),H(11,11),VD(1
1,11),HD(11,11),T(3,2)
6020 LC=1
6030 FOR I=1 TO 10:FOR J=1 TO 10
6040 P(I,J)=0:V(I,J)=1:H(I,J)=1:VD(I,J)=
0:HD(I,J)=0
6050 NEXT J:NEXT I
6060 FOR I=0 TO 11
6070 P(0,I)=1:P(I,0)=1:P(11,I)=1:P(I,11)
=1
6080 NEXT I
6090 RETURN
10000 REM FIN DE PARTIE
10010 INIT 6,4:TX 0,0,0
10020 PLAY "AAARREEEEEERRDDDRRRRCCCAAARAR
ARAREEEEEEEEEEE"
10030 PRINT "Votre temps est dépassé !"
```

```
10040 PRINT :PRINT "vous avez ramené ";T  
R;" trésor";:IF TR>1 THEN PRINT "s"  
10050 PRINT :PRINT  
10060 PRINT "Une autre partie (OUI/NON)"  
;  
10070 INPUT NP$:NP$=LEFT$(NP$,1)  
10080 IF NP$="O" OR NP$="o" THEN RUN  
10090 PRINT :PRINT "AU REVOIR !"  
10100 END
```



Jeux d'action à deux joueurs

3.1 FACE-À-FACE

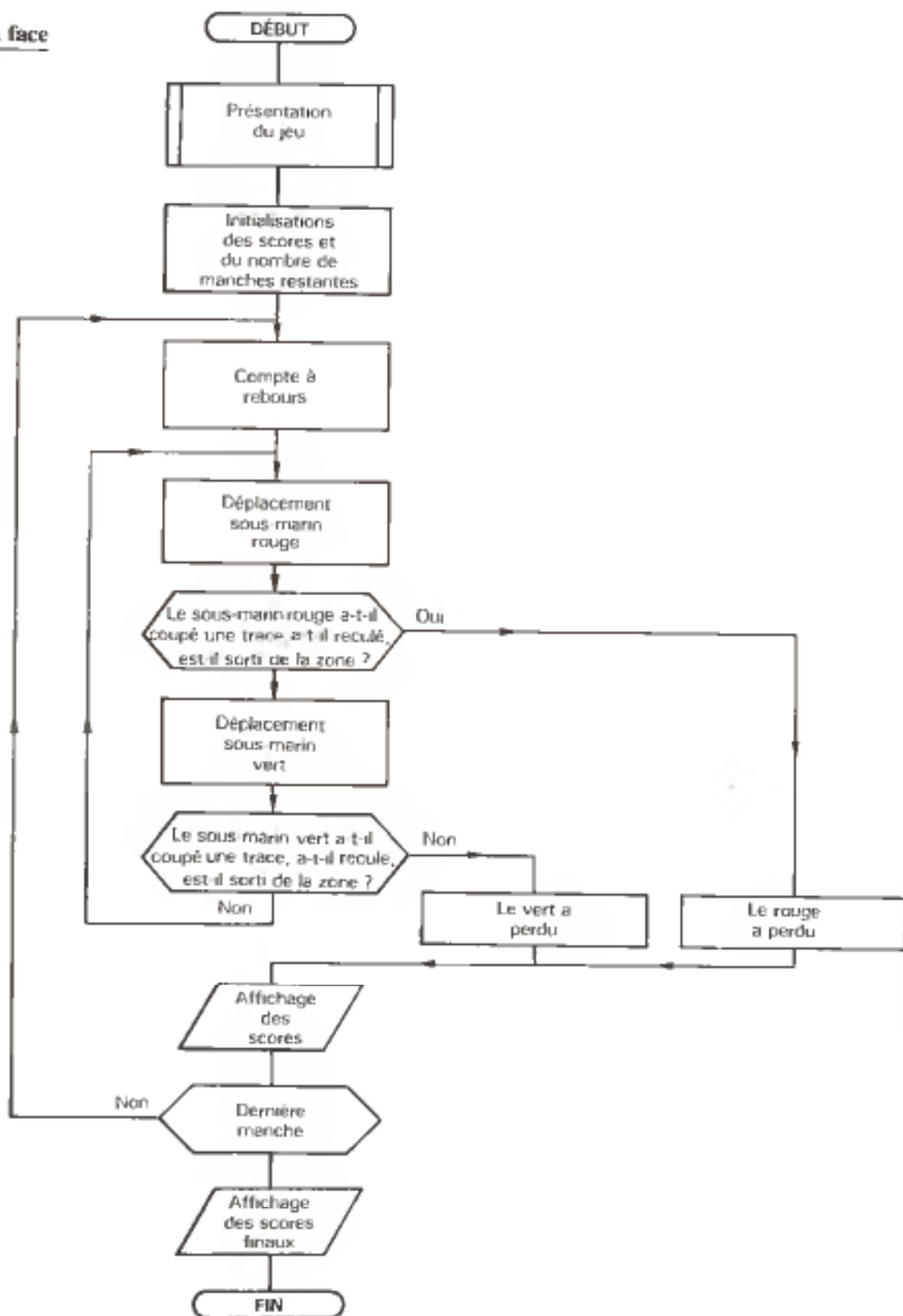
3.1.1 Présentation

Ce jeu oppose deux joueurs dans un espace limité à l'écran. Chaque concurrent est aux commandes d'un sous-marin qui largue des mines sur son passage. Les deux sous-marins laissent donc derrière eux une traînée sur laquelle ils ne doivent pas repasser sous peine d'explosion. L'évolution des sous-marins est réalisée grâce aux manches à balai ; elle est limitée par des falaises sous-marines qui bornent la zone de combat. Devant le remplissage constant de cette zone, chaque commandant doit constamment faire appel à sa ruse et à son habileté pour conduire son adversaire à l'explosion.

Le combat s'arrêtera lorsque dix sous-marins auront été détruits.

3.1.2 Organigramme et étude du programme

Face à face



La logique de ce programme consiste à avoir constamment dans un tableau une copie de la situation de l'écran et à y ranger, au fur et à mesure du déplacement des sous-marins, la trace de ces derniers. Le test de fin de partie se fait en lisant le tableau.

STRUCTURE

Lignes	10 à 120	Initialisations
Lignes	130 à 180	Compte à rebours
Lignes	190 à 280	Initialisation des positions des sous-marins
Lignes	290 à 450	Déplacements sous-marin rouge
Lignes	460 à 630	Déplacements sous-marin vert
Lignes	1000 à 1130	Fin de manche
Lignes	1140 à 1250	Fin de partie
Lignes	5000 à 5100	Présentation.

PRINCIPALES VARIABLES

X0, Y0	Coordonnées du sous-marin rouge
H0, V0	Déplacements du sous-marin rouge
X1, Y1	Coordonnées du sous-marin vert
H1, V1	Déplacements du sous-marin vert
ECRAN (x,y)	Copie de l'écran géré par le programme et qui contient les valeurs 0 si la case de l'écran est rouge 1 si la case est verte 2 si la case est vide
SV	Score du sous-marin vert,
SR	Score du sous-marin rouge
NMANCH	Nombre de manches restant à jouer dans la partie.

3.1.3 Liste des instructions

```
10 REM *****
20 REM * Face à face *
30 REM *****
40 GOSUB 5000
50 INIT 6,6
60 CURSORX 1:CURSORY 12
70 NMANCH=10:SR=0:SV=0
80 DIM ECRAN(40,22)
90 FOR I=1 TO 40
100 FOR J=1 TO 22
110 ECRAN(I,J)=2
120 NEXT J:NEXT I
130 REM Compte à rebours
140 INIT 6,6:TX 1,0,0
150 FOR I=5 TO 0 STEP -1
160 CURSORX 17:CURSORY 11:PRINT I
170 FOR J=1 TO 500:NEXT J
180 NEXT I
190 REM Initialisations des
200 REM positions des sous-marins
210 INIT 6,6:PAGE
220 X0=35:Y0=11:H0=0:V0=-1
230 TX 1,0,0:CURSORX X0
240 CURSORY Y0:PRINT CHR$(127);
250 X1=4:Y1=11:H1=0:V1=-1
260 TX 2,0,0:CURSORX X1
270 CURSORY Y1:PRINT CHR$(127);
280 ROUGE=0:VERT=1
290 REM Déplacement du sous-marin rouge
300 IF STICKX(1)=1 GOTO 350
310 IF STICKX(1)=255 GOTO 360
320 IF STICKY(1)=1 GOTO 370
330 IF STICKY(1)=255 GOTO 380
340 GOTO 390
350 H0=+1:V0=0:GOTO 390
360 H0=-1:V0=0:GOTO 390
370 H0=0:V0=+1:GOTO 390
380 H0=0:V0=-1
```

```
390 X0=X0+H0:Y0=Y0+V0
400 IF X0<1 OR X0>39 GOTO 1040
410 IF Y0<1 OR Y0>20 GOTO 1040
420 IF ECRAN(X0,Y0)<>2 GOTO 1040
430 ECRAN(X0,Y0)=ROUGE
440 CURSORX X0:CURSORY Y0:TX 1,0,0
450 PRINT CHR$(127);
460 REM Déplacement du sous-marin vert
470 IF STICKX(0)=1 GOTO 520
480 IF STICKX(0)=255 GOTO 530
490 IF STICKY(0)=1 GOTO 540
500 IF STICKY(0)=255 GOTO 550
510 GOTO 560
520 H1=+1:V1=0:GOTO 560
530 H1=-1:V1=0:GOTO 560
540 H1=0:V1=+1:GOTO 560
550 H1=0:V1=-1
560 X1=X1+H1:Y1=Y1+V1
570 IF X1<1 OR X1>39 GOTO 1010
580 IF Y1<1 OR Y1>20 GOTO 1010
590 IF ECRAN(X1,Y1)<>2 GOTO 1010
600 ECRAN(X1,Y1)=VERT
610 CURSORX X1:CURSORY Y1:TX 2,0,0
620 PRINT CHR$(127);
630 GOTO 290
1000 REM Fin de manche
1010 CURSORX 1:CURSORY 1
1020 TX 2,0,0:PRINT "Le vert a perdu"
1030 SR=SR+1:GOTO 1070
1040 CURSORX 1:CURSORY 1
1050 TX 1,0,0:PRINT "Le rouge a perdu"
1060 SV=SV+1
1070 REM Score
1080 TX 0,0,0:CURSORX 1:CURSORY 20
1090 PRINT "Rouge";SR
1100 CURSORX 1:CURSORY 21
1110 PRINT "Vert ";SV
1120 NMANCH=NMANCH-1
1130 IF NMANCH<>0 THEN 90
1140 INIT 2,2
1150 TX 1,0,0
1160 PRINT "Fin de la partie"
```

```
1170 CURSORX 1:CURSORY 10:TX 1,0,0
1180 PRINT "Rouge";SR
1190 CURSORX 1:CURSORY 11:TX 0,0,0
1200 PRINT "Vert ";SV
1210 CURSORX 1:CURSORY 20:TX 0,0,0
1220 PRINT "Une autre partie"
1230 INPUT R$
1240 IF R$="o" OR R$="O" THEN RUN
1250 INIT 6,6:END
5000 REM Présentation
5010 INIT 2,2:TX 1,7,0
5020 CURSORX 4:CURSORY 8
5030 PRINT "F A C E   à   F A C E "
5040 CURSORX 4:CURSORY 9
5050 PRINT "F A C E   à   F A C E "
5060 TX 0,0,0
5070 CURSORX 1:CURSORY 12
5080 PRINT "Pour commencer, enfoncez la
touche RET"
5090 INPUT R$
5100 RETURN
```

3.2 PILOTES DE COURSE

3.2.1 Présentation

Après avoir ingéré beaucoup de boissons alcoolisées, vous faites le pari (stupide !) avec l'un de vos amis de remonter le périphérique en sens inverse à cinq heures de l'après-midi. Combien de temps resterez-vous en piste ? Vous rebondissez sur chaque bord (haut, bas, gauche, droite) de la piste et de nombreuses voitures viennent en sens inverse.

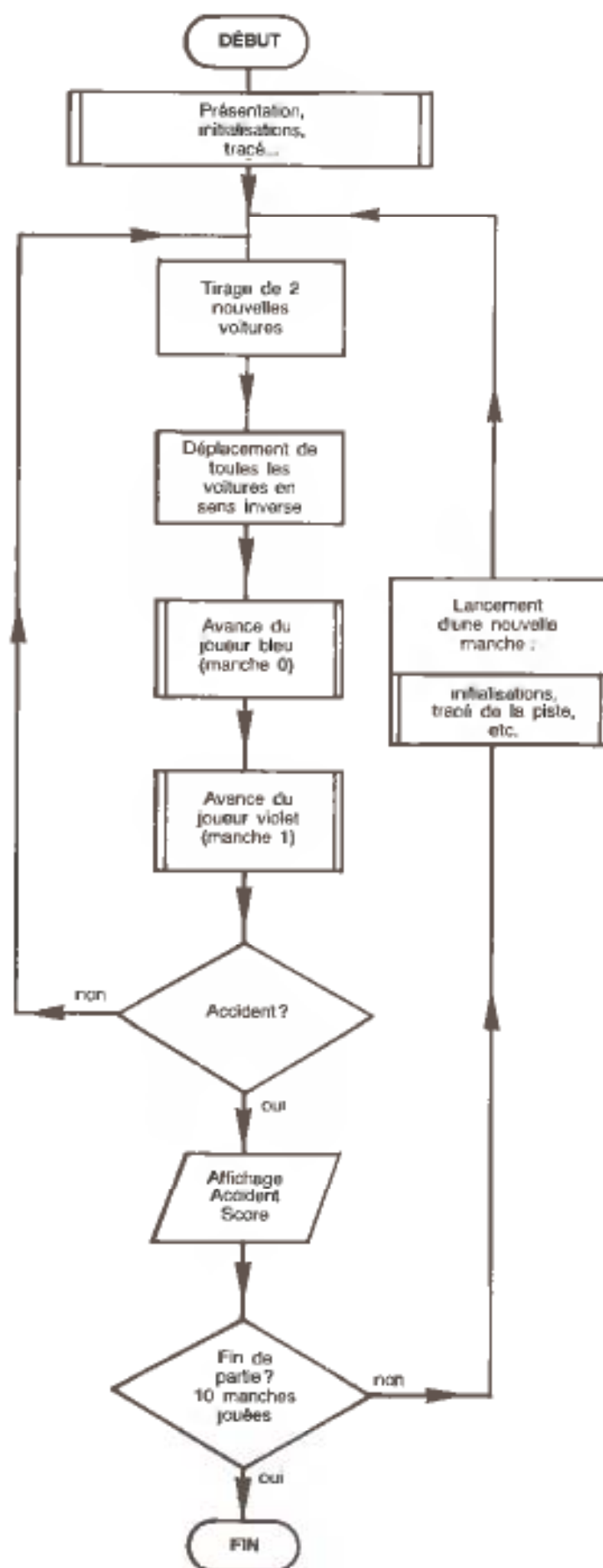
La partie se joue en 10 manches qui s'arrêtent à chaque accident. Vous dirigez vos voitures à l'aide des joysticks et le "chauffard de l'année" est désigné à la fin du jeu !

Bonne route !

3.2.2 Organigramme et étude du programme

STRUCTURE

Lignes	10 à	140	Appel des sous-programmes de présentation, d'initialisation, de tracé, etc... et affichage des voitures des deux joueurs
Lignes	150 à	470	Boucle de jeu
	170 à	300	Tirage des 2 nouvelles voitures venant en sens inverse
	310 à	410	Déplacement de toutes les voitures venant en sens inverse
	420 à	450	Avance des voitures des joueurs (appel deux fois du sous-programme situé aux lignes 500 à 720)



460		En cas d'accident, aller en fin de manche
470		Retour au début de la boucle
Lignes	500 à 720	Avance d'un joueur (utilisé pour les 2 joueurs) - effacement de la voiture - déplacement à l'aide d'un joystick - tests d'accidents - affichage de la voiture

REMARQUE : Deux tests d'accident sont nécessaires pour éviter que la voiture du joueur ne croise une voiture venant en sens inverse sans avoir d'accident.

Lignes	1000 à 1280	Fin de manche, affichage du score et test de fin de partie
Lignes	5000 à 5210	Présentation du jeu
Lignes	5300 à 5370	Définition des caractères spéciaux
Lignes	5400 à 5520	Tracé de la piste par des instructions
Lignes	5600 à 5640	Dimensionnement des tableaux
Lignes	5650 à 5790	Initialisations des variables
Lignes	10000 à 10150	Fin de partie et affichage du message correspondant

PRINCIPALES VARIABLES

P (40,25) Indique les limites de la piste, les voitures venant en sens inverse...

VO (21) Ordonnées)
 M (21) Abcisses) des voitures venant en
 T (21) : Codes) sens inverse
 C (21) : Couleurs)

CARACTERES SPECIAUX

Nous utilisons cinq caractères spéciaux pour ce jeu :

- un pour la voiture des joueurs
- quatre pour les voitures venant en sens inverse.

Voitures des joueurs



code 115

Voitures venant en sens inverse (codes 111 à 114)



3.2.3 Liste des instructions

```

10 REM *****
20 REM * Pilotes de course *
30 REM *****
40 GOSUB 5000:REM Présentation
50 GOSUB 5310:REM Caractères spéciaux
60 GOSUB 5600:REM Dim tableaux
70 NMANCH=10:REM Nombre de manches
80 SC(0)=0:SC(1)=0:REM Score
90 GOSUB 5650:REM Initialisations
100 GOSUB 5400:REM Tracé de la piste
110 FOR I=0 TO 1
120 CURSORX X(I):CURSOR Y(I)
130 ET 4+I,2,0:PRINT CHR$(JOU);" ";
140 NEXT I
150 REM Tour de jeu
160 PLAY "O1T1A"
170 REM Tirage des voitures
180 M(NB)=2*INT(RND(1)*(38-2*L)/2)+L
190 VO(NB)=1:IF P(M(NB),1)=1 GOTO 180
200 C(NB)=INT(RND(1)*4):T(NB)=111+C(NB)
210 CURSORX M(NB):CURSOR Y 1
220 ET C(NB),2,0:PRINT CHR$(T(NB));" ";
230 M(NB+1)=2*INT(RND(1)*(38-2*L)/2)+L
240 VO(NB+1)=1:IF P(M(NB+1),1)=1 GOTO 230
250 C(NB+1)=INT(RND(1)*4)
260 T(NB+1)=111+C(NB+1)
270 CURSORX M(NB+1):CURSOR Y 1
280 ET C(NB+1),2,0:PRINT CHR$(T(NB+1));
290 PRINT " ";
300 NB=NB+2:IF NB=22 THEN NB=0
310 REM Déplacement des voitures
320 FOR I=0 TO 21:IF VO(I)=0 GOTO 410
330 CURSORX M(I):CURSOR Y VO(I)
340 ET 7,2,0:PRINT CHR$(110);" ";
350 VO(I)=VO(I)+2
360 IF P(M(I),VO(I))=1 THEN VO(I)=0:GOTO
  410
370 P(M(I),VO(I))=0

```

```

380 CURSORX M(I):CURSORY VO(I)
390 ET C(I),2,0:PRINT CHR$(T(I));" ";
400 P(M(I),VO(I))=1:P(M(I),VO(I)-2)=0
410 NEXT I
420 REM Avance des joueurs
430 FOR I=0 TO 1
440 GOSUB 500
450 NEXT I
460 IF S(0)=1 OR S(1)=1 GOTO 1000
470 GOTO 150
500 REM Avance d'un joueur
510 REM Test d'accident
520 IF P(X(I),Y(I))=1 THEN S(I)=1
530 CURSORX X(I):CURSORY Y(I)
540 ET 7,2,0:PRINT CHR$(110);" ";
550 REM Déplacement de la voiture
560 IF STICKX(I)=1 THEN W(I)=2
570 IF STICKX(I)=255 THEN W(I)=-2
580 IF STICKY(I)=1 THEN Z(I)=2
590 IF STICKY(I)=255 THEN Z(I)=-2
600 X(I)=X(I)+W(I):Y(I)=Y(I)+Z(I)
610 REM Tests de rebonds sur les bords
620 IF Y(I)<1 THEN Y(I)=3:Z(I)=-Z(I)
630 IF Y(I)>19 THEN Y(I)=17:Z(I)=-Z(I)
640 IF X(I)<=L+2 THEN W(I)=-W(I):X(I)=L+
2
650 IF X(I)>=36-L THEN W(I)=-W(I):X(I)=3
6-L
660 REM Test d'accident
670 IF P(X(I),Y(I))=1 THEN S(I)=1
680 IF S(I)=1 THEN RETURN
690 REM Affichage de la voiture
700 CURSORX X(I):CURSORY Y(I)
710 ET 4+I,2,0:PRINT CHR$(JOU);" ";
720 RETURN
1000 REM Fin de manche
1010 PLAY "01T1GFEDCBA"
1020 INIT 1,1:TX 0,0,0
1030 CURSORX 1:CURSORY 5
1040 IF S(0)=1 AND S(1)=1 GOTO 1100
1050 IF S(1)=1 GOTO 1080
1060 PRINT "Le joueur bleu a eu un accident"

```

```
1070 SC(0)=SC(0)+1:GOTO 1120
1080 PRINT "Le joueur violet a eu un acc
ident"
1090 SC(1)=SC(1)+1:GOTO 1120
1100 PRINT "Les deux joueurs ont eu un a
ccident"
1110 FOR I=0 TO 1:SC(I)=SC(I)+1:NEXT I
1120 REM Affichage du score
1130 CURSORX 5:CURSORY 12
1140 TX 4,0,0:PRINT "bleu : ";SC(0);
1150 PRINT " accident";
1160 IF SC(0)>1 THEN PRINT "s"
1170 CURSORX 5:CURSORY 13
1180 TX 5,0,0:PRINT "violet: ";SC(1);
1190 PRINT " accident";
1200 IF SC(1)>1 THEN PRINT "s"
1210 REM Test de fin de partie
1220 NMANCH=NMANCH-1
1230 IF NMANCH=0 GOTO 10000
1240 REM Relancer une autre manche
1250 FOR I=0 TO 1
1260 M(I)=0:Z(I)=0
1270 NEXT I
1280 GOTO 90
5000 REM Présentation
5010 INIT 3,3:TX 4,7,0
5020 CURSORX 8:CURSORY 4
5030 PRINT "Ppillootteess ddee"
5040 CURSORX 8:CURSORY 5
5050 PRINT "Ppillootteess ddee"
5060 CURSORX 12:CURSORY 8
5070 PRINT "ccouurrssee"
5080 CURSORX 12:CURSORY 9
5090 PRINT "ccouurrssee"
5100 TX 1,0,0
5110 CURSORX 2:CURSORY 12
5120 PRINT "Restez le plus longtemps en
piste en"
5130 PRINT "évitant les voitures venant
en sens"
5140 PRINT "inverse et en rebondissant h
abilement"
```

```
5150 PRINT "sur les bords de la piste"
5160 PRINT :PRINT " Le vaincu est celui
des 2 joueurs qui"
5170 PRINT "a un accident en premier. La
partie se"
5180 PRINT "joue en 10 manches."
5190 PRINT :PRINT "Bonne route!!":PRINT
5200 INPUT "RET pour commencer";R$
5210 RETURN
5300 REM Caractères spéciaux
5310 SETET 110,"FFFFFFFFFFFFFFFFFFFFF"
5320 SETET 111,"00BDFFB3C185A7E5A00"
5330 SETET 112,"0081FFFF7E3CBDF993C"
5340 SETET 113,"18BDFFB3C3CBDF993C"
5350 SETET 114,"00BDF993C3C5A7E5A18"
5360 SETET 115,"005A7E5A183CBDFFB00"
5370 RETURN
5400 REM Tracé de la piste
5410 INIT 7,2:TX 0,0,0
5420 FOR Y=0 TO 24
5430 CURSORX 0:CURSOR Y
5440 DELIM 2,2,0
5450 CURSORX L:CURSOR Y
5460 IF Y<20 THEN DELIM 2,7,0
5470 IF Y>19 THEN DELIM 1,7,0
5480 CURSORX 38-L:CURSOR Y
5490 IF Y<20 THEN DELIM 2,2,0
5500 IF Y>19 THEN DELIM 1,2,0
5510 NEXT Y
5520 RETURN
5600 REM Dimensionnement
5610 DIM P(40,25)
5620 DIM VO(21),M(21),T(21),C(21)
5630 DIM X(1),Y(1),S(1),NT(1),W(1),Z(1)
5640 RETURN
5650 REM Initialisations
5660 V=10:L=5:JOU=115:NB=0
5670 X(0)=21:Y(0)=17:X(1)=11:Y(1)=17
5680 S(0)=0:S(1)=0
5690 FOR I=0 TO 40:FOR J=0 TO 24
5700 P(I,J)=0
5710 NEXTJ:NEXT I
```

```
5720 FOR J=0 TO 24:FOR I=0 TO L
5730 P(I,J)=1:P(38-I,J)=1
5740 NEXT I:NEXT J
5750 FOR I=0 TO 40:P(I,25)=1:NEXT I
5760 FOR I=0 TO 21
5770 VO(I)=0:M(I)=0:T(I)=0:C(I)=0
5780 NEXT I
5790 RETURN
10000 REM Fin de partie
10010 IF SC(0)>SC(1) THEN J$="bleu"
10020 IF SC(0)<SC(1) THEN J$="violet"
10030 TX 0,0,0
10040 CURSORX 1:CURSORY 15
10050 IF SC(0)=SC(1) GOTO 10090
10060 PRINT"Le joueur ";J$;" est élu ";
10070 PRINT"chauffard":PRINT"de l'année
!!!!"
10080 GOTO 10110
10090 PRINT "Vous êtes aussi mauvais con
ducteurs"
10100 PRINT "l'un que l'autre!!!!"
10110 CURSORX 1:CURSORY 22
10120 INPUT "Une autre partie ";R$
10130 IF R$="D" OR R$="o" GOTO 70
10140 INIT 6,6:PRINT "Au revoir !"
10150 END
```

3.3 BATAILLE D'HÉLICOPTÈRES

3.3.1 Présentation

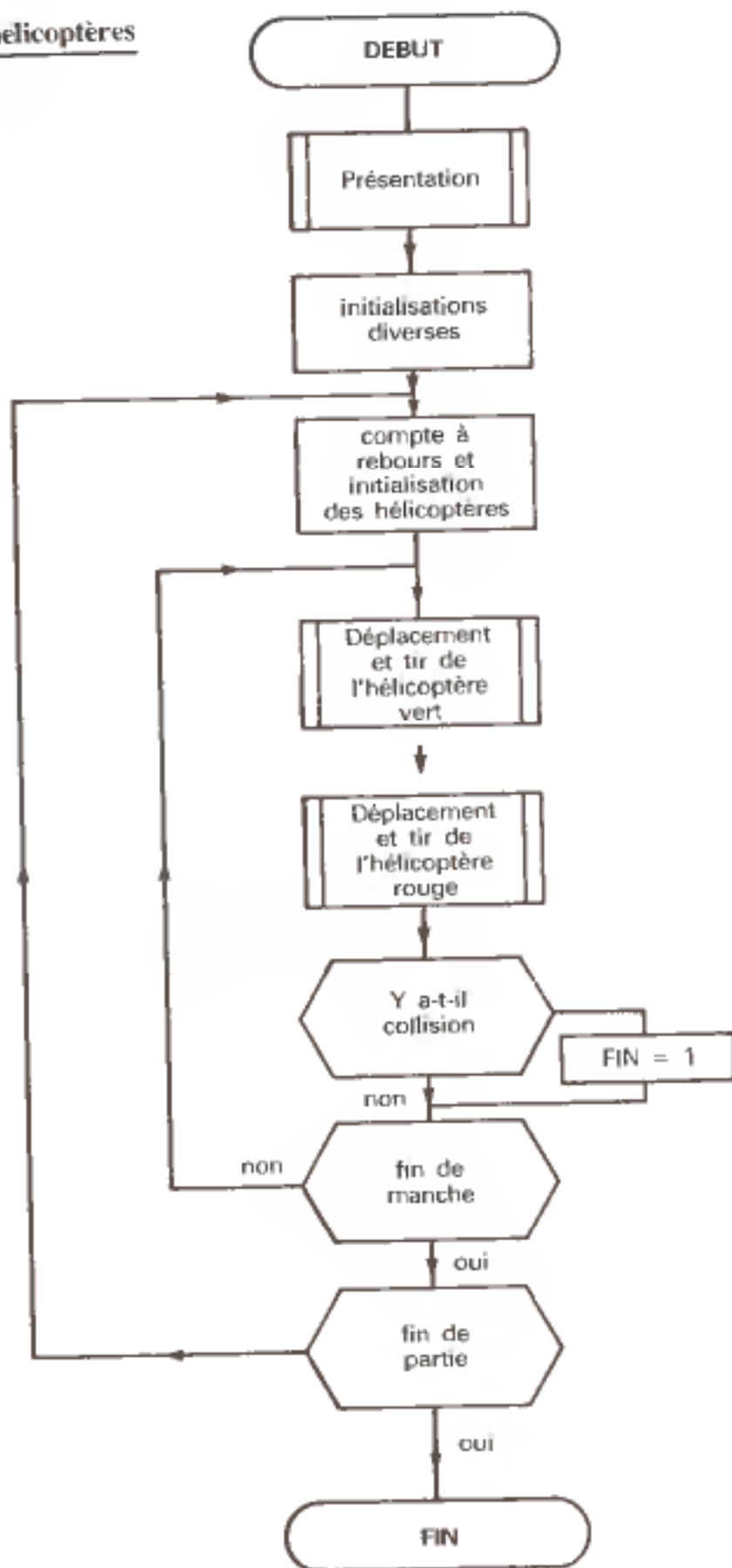
Nous sommes en 1942 et la guerre est à son paroxysme. Le sort de vos patries respectives dépend du résultat de la bataille que vous vous livrez au-dessus de la frontière aux commandes de vos hélicoptères. Le pilote qui sortira vainqueur de cet affrontement en dix épisodes apportera la paix à son pays !

3.3.2 Organigramme et étude du programme

STRUCTURE

Lignes	10 à	70	Initialisations
Lignes	80 à	130	Compte à rebours
Lignes	140 à	200	Initialisation des positions des hélicoptères
Lignes	210 à	250	Déplacement hélicoptère vert
Lignes	260 à	300	Déplacement hélicoptère rouge
Lignes	310		Appel du sous-programme de déplacement du nuage
Lignes	330 à	380	Test de collision
Lignes	400 à	790	Déplacement et tir d'un hélicoptère

Bataille d'hélicoptères

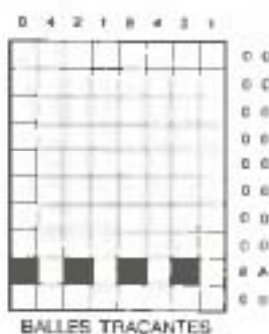
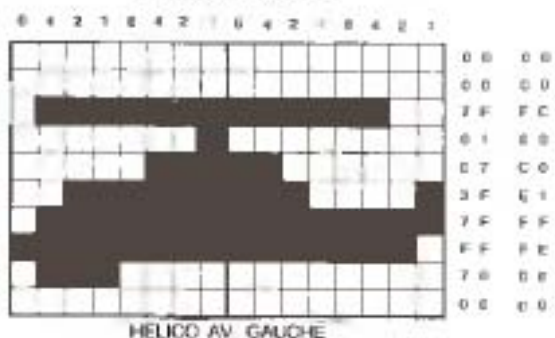
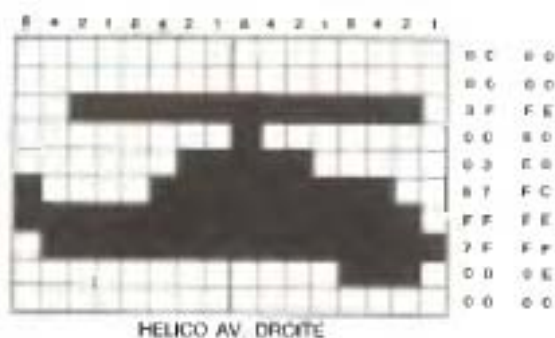


Lignes	1000 à 1090	Affichage d'un hélicoptère
Lignes	1100 à 1190	Déplacement du nuage
Lignes	1200 à 1370	Affichage du nuage
Lignes	5000 à 5160	Présentation du jeu et définition des caractères spéciaux
Lignes	10000 à 10280	Fins de manche et de partie

CARACTERES SPECIAUX

Le programme utilise les cinq caractères spéciaux suivants :

- quatre pour les hélicoptères
- un pour les balles.



3.3.3 Liste des instructions

```
10 REM *****
20 REM * Bataille d'hélicoptères *
30 REM *****
40 GOSUB 5000
50 INIT 6,6
60 CURSORX 1:CURSORY 12
70 NMANCH=10:SR=0:SV=0:YN=10:VN=10
80 REM compte à rebours
90 INIT 6,6:TX 1,0,0
100 FOR I=5 TO 0 STEP-1
110 CURSORX 17:CURSORY 11:PRINT I
120 FOR J=1 TO 500:NEXT J
130 NEXT I
140 REM initialisations des
150 REM positions des hélicoptères
160 INIT 0,0:PAGE:PRINT CHR$(150)
170 STORE
180 D0=1:D1=-1
190 X0=1:Y0=11:H0=-1:V0=1:T0=0:C0=2
200 X1=35:Y1=11:H1=1:V1=1:T1=0:C1=1
210 REM déplacement hélicoptère vert
220 S=S0:HB=H0:VB=V0:T=T0:XA=X1:YA=Y1
230 X=X0:Y=Y0:M=0:C=C0:DX=D0:GOSUB 400
240 S0=S:H0=HB:V0=VB:T0=T:X0=X:Y0=Y
250 D0=DX
260 REM déplacement hélicoptère rouge
270 S=S1:HB=H1:VB=V1:T=T1:XA=X0:YA=Y0
280 X=X1:Y=Y1:M=1:C=C1:DX=D1:GOSUB 400
290 S1=S:H1=HB:V1=VB:T1=T:X1=X:Y1=Y
300 D1=DX
310 GOSUB 1100
320 SCREEN
330 REM test de collision
340 IF Y1<>Y0 GOTO 370
350 IF ABS(X1-X0)>1 GOTO 370
360 FIN=1
370 IF FIN=0 GOTO 220
```

```

380 GOTO 10000
400 REM déplacement hélicoptère
410 IF T=0 THEN 440
420 POKE 16384+2*HB+80*VB,124+128
430 POKE 16385+2*HB+80*VB,0
440 C=0:GOSUB 1000
450 DY=0:DX=SGN(DX)
460 IF STICKX(M)=1 THEN DX=1
470 IF STICKX(M)=255 THEN DX=-1
480 IF STICKY(M)=1 THEN DY=1
490 IF STICKY(M)=255 THEN DY=-1
500 X=X+DX:Y=Y+DY
510 IF X<1 THEN X=1
520 IF X>36 THEN X=36
530 IF Y<0 THEN Y=0
540 IF Y>21 THEN Y=21
550 C=M+1:GOSUB 1000
560 REM tir des hélicoptères
570 IF T=1 THEN 640
580 IF ACTION(M)=0 THEN RETURN
590 S=SGN(DX):IF S=0 THEN RETURN
600 T=1:S=2*S
610 VB=Y:HB=X-(S=2)*3-1
620 GOTO 700
630 REM avance de la balle et test
640 HB=HB+S
650 IF HB<XA-1 OR HB>XA+2 THEN 690
660 IF VB<>YA THEN 690
670 T=0:IF FIN<>0 THEN FIN=3:GOTO 690
680 FIN=M+2
690 IF HB<1 OR HB>39 THEN T=0:RETURN
700 POKE 16384+2*HB+80*VB,124+128
710 POKE 16385+2*HB+80*VB,C
720 RETURN
1000 REM affichage des hélicoptères
1010 IF DX<0 GOTO 1040
1020 K1=120+128:K2=121+128
1030 GOTO 1050
1040 K1=122+128:K2=123+128
1050 POKE 16384+2*X+80*Y,K1
1060 POKE 16385+2*X+80*Y,C
1070 POKE 16386+2*X+80*Y,K2

```

```

1080 POKE 16387+2*X+80*Y,C
1090 RETURN
1100 REM Déplacement du nuage
1110 TX 0,0,0:GOSUB 1200
1120 XN=XN+(INT(RND(1)*2.99)-1)/2
1130 YN=YN+(INT(RND(1)*2.99)-1)/2
1140 IF XN<1 THEN XN=1
1150 IF YN<0 THEN YN=0
1160 IF XN>28 THEN XN=28
1170 IF YN>16 THEN YN=16
1180 TX 7,0,0:GOSUB 1200
1190 RETURN
1200 REM Affichage d'un nuage
1210 CURSORX XN+4:CURSORY YN
1220 PRINTCHR$(127)+CHR$(127)+CHR$(127)
1230 CURSORX XN+2
1240 FOR I=1TO8:PRINT CHR$(127);:NEXT I
1250 PRINT
1260 FOR J=1 TO 2
1270 CURSORX XN
1280 FOR I=1TO11:PRINT CHR$(127);:NEXT I
1290 PRINT
1300 NEXT J
1310 CURSORX XN+1
1320 FOR I= 1TO9:PRINT CHR$(127);:NEXT I
1330 PRINT:CURSORX XN+2
1340 FOR I= 1TO6:PRINT CHR$(127);:NEXT I
1350 PRINT:CURSORX XN+5
1360 FOR I= 1TO3:PRINT CHR$(127);:NEXT I
1370 RETURN
5000 REM présentation
5010 INIT 2,2:TX 0,3,0
5020 CURSORX 4:CURSORY 8
5030 PRINT "H E L I C O P T E R E S "
5040 CURSORX 4:CURSORY 9
5050 PRINT "H E L I C O P T E R E S "
5060 TX 0,0,0
5070 CURSORX 1:CURSORY 12
5080 PRINT "pour le départ, enfoncez la touche
RET"
5090 INPUT R$
5100 SETET 127,"FFFFFFFFFFFFFFFFFFFFFF"

```

```
5110 SETET 120,"00003F000387FF7F0000"  
5120 SETET 121,"0000FE80E0FCFEFF0E00"  
5130 SETET 122,"00007F01073F7FFF7000"  
5140 SETET 123,"0000FC00C0E1FFFE0000"  
5150 SETET 124,"00000000000000006A00"  
5160 RETURN  
10000 REM fin =1 collision  
10010 REM fin =2 rouge vainqueur  
10020 REM fin =3 vert vainqueur  
10030 REM fin =4 tous abattus  
10040 DISPLAY 20  
10050 TX 3,0,0:CURSORX 1:CURSORY 10  
10060 ON FIN-1 GOTO 10090, 10110, 10130  
10070 PRINT "collision en vol"  
10080 GOTO 10150  
10090 PRINT "le vert a perdu"  
10100 SR=SR+1:GOTO 10150  
10110 PRINT "le rouge a perdu"  
10120 SV=SV+1:GOTO 10150  
10130 PRINT "les deux balles ont fait mouche"  
10140 SV=SV+1:SR=SR+1  
10150 NMANCH=NMANCH-1:FIN=0  
10160 CURSORX 1:CURSORY 15:TX 1,0,0  
10170 PRINT "rouge";SR  
10180 CURSORX 1:CURSORY 16:TX2,0,0  
10190 PRINT "vert ";SV  
10200 FOR I=1 TO 5000:NEXT I  
10210 IF NMANCH>0 THEN 80  
10220 TX 6,0,0:CURSORX 2:CURSORY 6  
10230 PRINT "fin de partie"  
10240 CURSORX 1:CURSORY 20:TX 4,0,0  
10250 PRINT "voulez-vous rejouer";  
10260 INPUT R$  
10270 IF R$="o" OR R$="O" GOTO 40  
10280 INIT 6,6:END
```

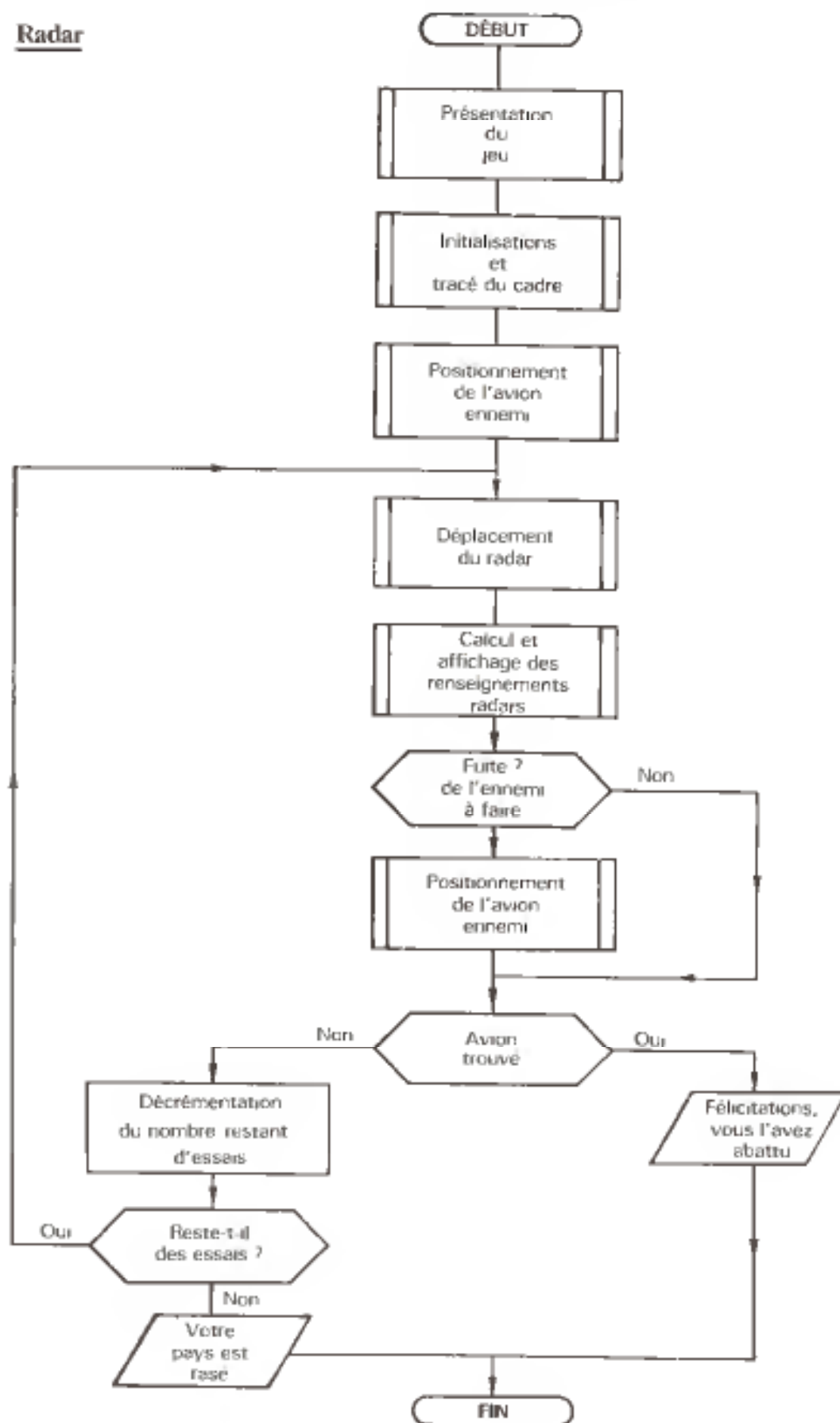
Jeux de réflexion à un joueur

4.1 RADAR

4.1.1 Présentation

A la tête de votre petite unité IMC (Logistique Mobile de Combat), vous avez pris position dans la région de Calinie Inférieure. Celle-ci pose depuis longtemps un problème à votre état-major, car ses nombreuses montagnes facilitent les incursions de l'aviation ennemie, qui ne peut être localisée par des radars conventionnels. L'ennemi peut ainsi ravitailler ses unités sans grand risque d'interception par votre chasse. Le but de votre unité consiste à localiser l'appareil ennemi et à transmettre ses coordonnées à la chasse qui se chargera de le descendre. Pour cela, positionnez votre unité (le curseur) à l'emplacement de votre choix avec les flèches et appuyez sur la barre d'espace. Des renseignements sur la position de l'appareil ennemi s'afficheront alors à gauche de votre écran de contrôle. Votre tâche ne sera pas facilitée : l'adversaire vous détecte en trois coups et s'enfuit alors. De plus, votre intendance, par trop sous-équipée, ne permet pas plus de 50 essais de recherche.

4.1.2 Organigramme et étude du programme

Radar

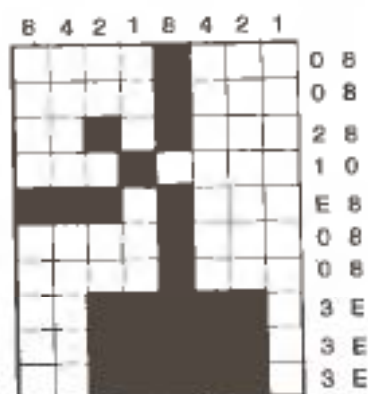
Lignes	10 à	50	Initialisations
Lignes	60 à	110	Boucle principale avec gestion de la fuite de l'avion ennemi
Lignes	120 à	170	Positionnement de l'avion
Lignes	180 à	340	Déplacement du curseur
Lignes	320 à	510	Calcul de l'objectif
Lignes	5000 à	5100	Présentation
Lignes	5110 à	5200	Tracé et initialisations
Lignes	10000 à	10080	Fin de partie et affichage des messages correspondant aux divers cas

PRINCIPALES VARIABLES

C, L	Positionnement du curseur
CA, LA	Positionnement de l'avion
NO, SO, NE, SE	Résultats des détecteurs

CARACTERE SPECIAL

Nous utilisons comme radar le caractère suivant :



4.1.3 Liste des instructions

```
10 REM *****
20 REM RADAR
30 REM *****
40 GOSUB 5000:GOSUB 5110:DISPLAY1
50 C=2:L=7:GOSUB 120
60 FOR I=1 TO 30
70 GOSUB 180:GOSUB 350
80 IF C=CA AND L=LA GOTO 10040
90 FUITE=FUITE-1:SOUND 150,10,2
100 IF FUITE=0 THEN GOSUB 120
110 NEXT I:GOTO 10000
120 REM Mise en place de l'avion
130 CA=INT(RND(1)*10)+1
140 LA=INT(RND(1)*10)+1
150 IF CA=11 OR LA=11 THEN 130
160 FUITE=4:SOUND 150,1,10
170 RETURN
180 REM Déplacement du curseur
190 CURSORX 10+2*(C-1):CURSORY 2*L-1
200 ET 4,0,1:PRINT CHR$(126)
220 CURSORX 10+2*(C-1):CURSORY 2*L-1
230 ET 0,0,0:PRINT CHR$(126)
240 IF STICKX(2)=1 THEN C=C+1
250 IF STICKX(2)=255 THEN C=C-1
260 IF C>10 THEN C=10
270 IF C<1 THEN C=1
280 IF STICKY(2)=1 THEN L=L+1
290 IF STICKY(2)=255 THEN L=L-1
300 IF L>10 THEN L=10
310 IF L<1 THEN L=1
320 IF ACTION(2)=1 THEN RETURN
340 GOTO 180
350 REM Distance de l'objectif
360 N1=0:S1=0:N2=0:S2=0
370 IF CA<C THEN 410
380 REM Joueur à gauche de l'avion
390 IF LA<L THEN N2=1:GOTO 430
400 S2=1:GOTO 430
```

```
410 IF L<LA THEN N1=1:GOTO 430
420 S1=1
430 REM Visualisation
440 IF C=CA AND L>LA THEN N1=1:N2=1
450 IF C=CA AND L<LA THEN S1=1:S2=1
460 IF L=LA AND C<CA THEN N2=1:S2=1
470 IF L=LA AND C>CA THEN N1=1:S1=1
480 TX 4,0,0
490 CURSORX 1:CURSORY 2:PRINT "no:";N1
500 CURSORX 1:CURSORY 4:PRINT "ne:";N2
510 CURSORX 1:CURSORY 6:PRINT "so:";S1
520 CURSORX 1:CURSORY 8:PRINT "se:";S2
530 RETURN
5000 REM Présentation
5010 INIT 2,2:TX 4,7,0
5020 CURSORX 4:CURSORY 10
5030 PRINT "r a d a r "
5040 CURSORX 4:CURSORY 11
5050 PRINT "r a d a r "
5060 TX 0,0,0:PRINT :PRINT
5070 PRINT "RET pour commencer";
5080 INPUT R$
5090 RETURN
5110 REM Tracé et initialisations
5120 INIT 0,0:TX 1,0,0:PAGE
5130 FOR I=0 TO 20
5140 CURSORX 9:CURSORY I
5150 PRINT CHR$(127)
5160 CURSORX 29:CURSORY I
5170 PRINT CHR$(127)
5180 NEXT I
5190 FOR I=10 TO 20
5200 CURSORX I:CURSORY 0
5210 PRINT CHR$(127);
5220 CURSORX I:CURSORY 20
5230 PRINT CHR$(127);
5240 NEXT I
5250 SETET 126,"08082010E808083E3E3E"
5260 RETURN
10000 REM Test de fin de partie
10010 INIT 7,7:TX0,0,0:PLAY"010FEDAAA"
10020 PRINT "Votre pays est rasé"
```

```
10030 GOTO 10080
10040 REM Gagné
10050 INIT 2,2:TX0,0,0:PLAY"01T1ABCDEFG"
10060 PRINT "Félicitations, vous l'avez
abattu."
10080 PRINT :PRINT "Une autre partie"
10090 INPUT R$
10100 IF R$="o" OR R$="O" THEN RUN
10110 INIT 6,6:END
```

4.2 TAQUIN

4.2.1 Présentation

Taquin est un jeu de réflexion qui est tout à fait recommandé pour meubler vos longues soirées d'hiver. Le jeu se joue dans un cadre où se trouvent 16 carrés mobiles. Parmi ces carrés, quinze ont une lettre (a à o) inscrite sur leur face. Un espace vide remplace le seizième carré, permettant ainsi de bouger les carrés contigus (gauche, droite, haut, bas).

L'ordinateur positionnera au hasard les 16 carrés, vous laissant le loisir de les remettre dans l'ordre suivant :

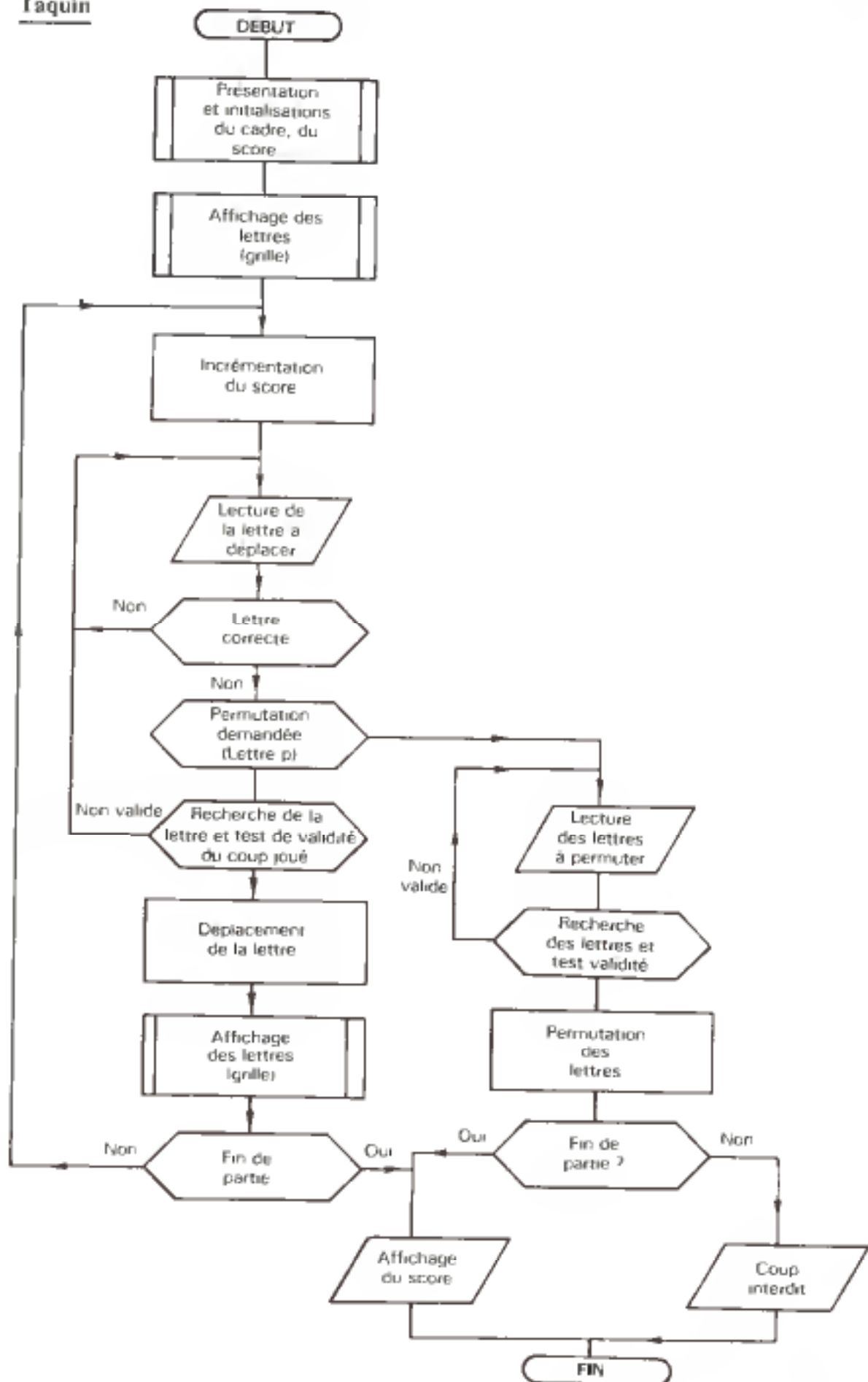
A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	.

Si vous n'êtes pas en mesure d'ordonner cette série, l'ordinateur vous donnera (une fois seulement, et obligatoirement comme fin de partie) la possibilité d'échanger deux lettres entre elles.

4.2.2 Organigramme et étude du programme

Lignes	60 à 280	Boucle principale
	60	Incrémentation du nombre de coups joués
	70 à 110	Lecture de la lettre à bouger
	120	Testent une permutation éventuelle demandée

Taquin



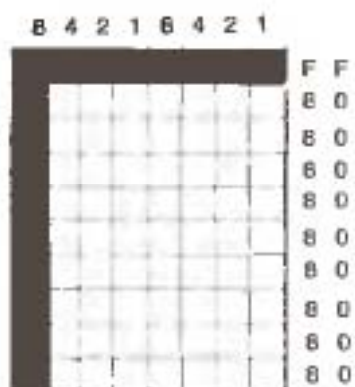
	130 à 170	Recherche de la lettre et de la case vide
	180 à 200	Testent la validité du coup
	210 à 230	Coup interdit
	240 à 250	Déplacement de la lettre
	260 à 280	Affichage du taquin et du test de fin de partie
Lignes	500 à 650	Permutation de 2 lettres
	510 à 550	Saisie des lettres
	560 à 620	Recherche des lettres, affichage du taquin
	630	Test de fin de partie
	640 à 650	Coup interdit
Lignes	700 à 800	Affichage du taquin
Lignes	900 à 960	Test de victoire
Lignes	5000 à 5540	
	5000 à 5090	Présentation
	5100 à 5200	Contenu du taquin
	5210 à 5300	Définition des caractères spéciaux
	5310 à 5510	Tracé de la grille
	5520 à 5540	Initialisation du score
Lignes	10000 à 10080	Fin de partie

PRINCIPALES VARIABLES

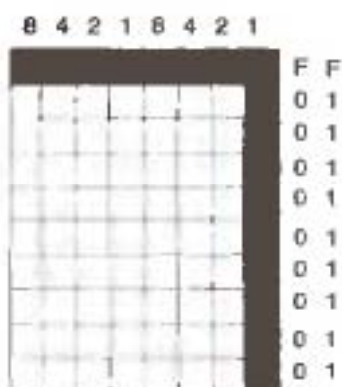
L\$, L1\$, L2\$ Lettres à déplacer ou à permuter

CARACTERES SPECIAUX

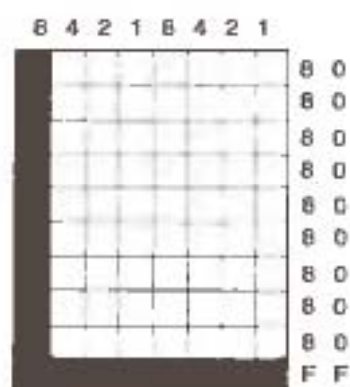
Pour tracer la grille, nous utilisons les huit caractères suivants :



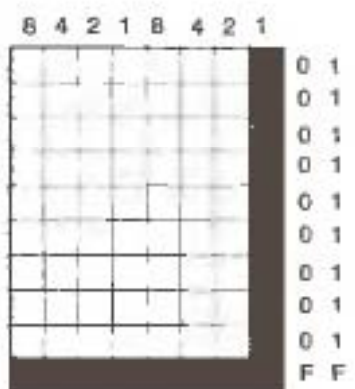
110



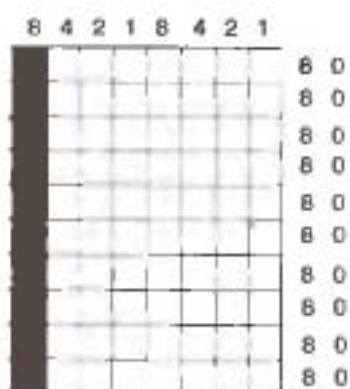
111



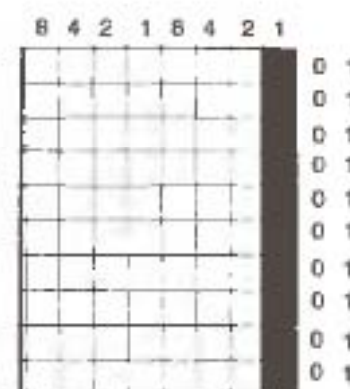
112



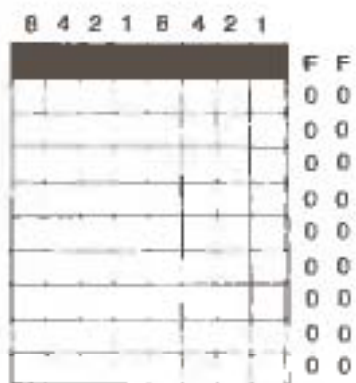
113



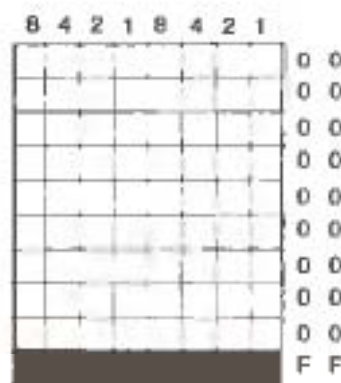
123



124



125



126

4.2.3 Liste des instructions

```
10 REM *****
20 REM Taquin
30 REM *****
40 INIT 6:GOSUB 5000
50 GOSUB 700
60 PAGE:SR=SR+1
70 REM Déplacement
80 CURSORX 1:CURSORY 17
90 INPUT "Lettre à déplacer";L$
100 IF ASC(L$)<97 GOTO 80
110 IF ASC(L$)>112 GOTO 80
120 IF ASC(L$)=112 GOTO 500
130 REM Recherche de la lettre,du blanc
140 FOR I=1 TO 16
150 IF C$(I)=L$ THEN A1=I
160 IF C$(I)=" " THEN A2=I
170 NEXT I
180 REM TEST DE VALIDITE DU COUP
190 IF ABS(A1-A2)=1 THEN 240
200 IF ABS(A1-A2)=4 THEN 240
210 PRINT "Coup interdit"
220 CURSORX 1:CURSORY 18 :FOR I=1 TO 10:
SOUND 100,1:NEXT I
230 PRINT "          ":GOTO 80
240 C$(A1)=" "
250 C$(A2)=L$
260 GOSUB 700:GOSUB 900
270 IF FIN=1 GOTO 10000
280 GOTO 60
500 REM PERMUTATION DE 2 LETTRES
510 INPUT "Lettres à échanger";L1$,L2$
520 IF ASC(L1$)<97 GOTO 510
530 IF ASC(L1$)>111 GOTO 510
540 IF ASC(L2$)<97 GOTO 510
550 IF ASC(L2$)>111 GOTO 510
560 FOR I=1 TO 16
570 IF C$(I)=L1$ THEN A1=I
580 IF C$(I)=L2$ THEN A2=I
```



```
590 NEXT I
600 C$(A1)=L2$
610 C$(A2)=L1$
620 GOSUB 700
630 GOSUB 900:IF FIN=1 GOTO 10000
640 PRINT "Coup interdit":SOUND 200,200:
GOTO 10050
650 GOTO 10040
700 REM Affichage du taquin
710 X=12:Y=5:TX 4,0,0
720 FOR J=1 TO 4
730 CURSOR Y
740 FOR I=1 TO 4
750 CURSORX X+I*3
760 PRINT C$(I+(J-1)*4);
770 NEXT I
780 Y=Y+3
790 NEXT J
800 RETURN
900 REM TEST DE FIN DE PARTIE
910 FIN=1:CURSOR Y 21:CURSORX 2
920 FOR I=1 TO 15
930 IF ASC(C$(I))<>(I+96) THEN FIN=0
940 NEXT I
950 IF ASC(C$(16))<>32 THEN FIN=0
960 RETURN
5000 REM Présentation
5010 CURSORX 10:CURSOR Y 10
5020 TX 4,7,0
5030 PRINT "T a q u i n "
5040 CURSORX 10:CURSOR Y 11
5050 PRINT "T a q u i n "
5060 TX 1,0,0:PLAY "AAABAAABBCDEDDDE"
5070 CURSORX 1:CURSOR Y 20
5080 PRINT "RET pour commencer"
5090 INPUT R$
5100 REM Contenu du taquin
5110 DIM C$(16):ERR=0
5120 FOR I=1 TO 15
5130 L$=CHR$(INT(RND(1)*15)+97)
5140 FOR J=1 TO I
5150 IF L$=C$(J) THEN ERR=1
```

```
5160 NEXT J
5170 IF ERR=1 THEN ERR=0:GOTO 5130
5180 C$(I)=L$:ERR=0
5190 NEXT I
5200 C$(16)=" "
5210 REM Caractères spéciaux
5220 SETET 110,"FF80808080808080808080"
5230 SETET 111,"FF010101010101010101"
5240 SETET 112,"80808080808080808080FF"
5250 SETET 113,"010101010101010101FF"
5260 SETET 123,"80808080808080808080"
5270 SETET 124,"000000000000000000FF"
5280 SETET 125,"01010101010101010101"
5290 SETET 126,"FF000000000000000000"
5300 SETET 032,"00000000000000000000"
5310 REM Tracé de la grille
5320 INIT 6:ET 1,0,0:CURSORX 4
5330 FOR J=1 TO 4
5340 CURSORX 14
5350 FOR I=1 TO 4
5360 PRINT CHR$(110);CHR$(126);
5370 PRINT CHR$(111);
5380 NEXT I
5390 PRINT :CURSORX 14
5400 FOR I=1 TO 4
5410 PRINT CHR$(123);CHR$(32);
5420 PRINT CHR$(125);
5430 NEXT I
5440 PRINT :CURSORX 14
5450 FOR I=1 TO 4
5460 PRINT CHR$(112);CHR$(124);
5470 PRINT CHR$(113);
5480 NEXT I
5490 PRINT
5500 NEXT J
5510 TX 4,0,0
5520 REM Score
5530 SR=0
5540 RETURN
10000 REM Fin de partie
10010 CURSORX 1:CURSORX 18
10020 PRINT "Vous avez gagné en ";SR;
```

```
10030 PRINT " coups":PLAY "EDEDEDAAA"  
10040 CURSORX 1:CURSORY 20  
10050 PRINT "Voulez-vous rejouer o/n";  
10060 INPUT R$  
10070 IF R$="o" OR R$="O" THEN RUN  
10080 INIT 6,6:END
```

4.3 PUZZLE

4.3.1 Présentation

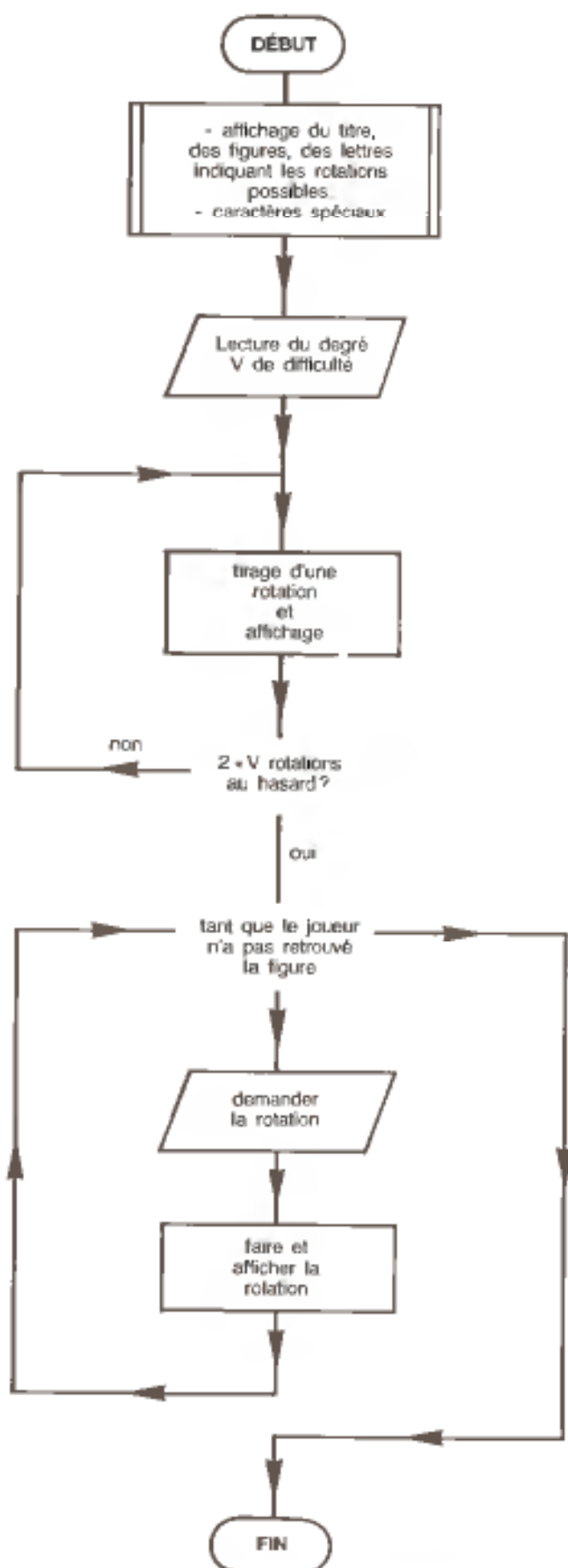
Dans votre château de TRANSYLVANIE, vous venez de terminer votre nouveau monstre (robotisé bien sûr). Malheureusement pour vous, votre chef d'oeuvre s'est retrouvé par terre à la suite de la tempête qui a détruit vos ateliers (le vent a pu varier entre les forces 1 et 5 et les dégâts sont proportionnels).

Douze types de rotations sont possibles pour remettre en place les diverses pièces de votre robot.

4.3.2 Organigramme et étude du programme

STRUCTURE

Lignes	40 à	50	Initialisations
Ligne	60		Appel du sous-programme d'affichage du titre, de définition des caractères spéciaux, de génération et de tracé des figures...
Lignes	80 à	190	Lecture du degré de difficulté voulu
Lignes	200 à	310	Modification de la figure de gauche
	220 à	250	Tirage de la rotation
	260 à	310	Rotation effectuée

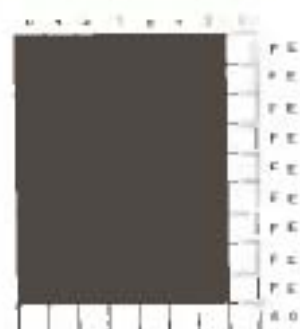
PUZZLE

Lignes	320 à 330	à 470	Lecture de la rotation souhaitée (coup du joueur)
	480 à 570	à 660	Effectuer la rotation Test de reconstitution du puzzle
Lignes	1000 à 1100		Génération de la figure (modifier ce sous-programme change le puzzle)
Lignes	1200 à 1290		Affichage de la figure
Lignes	1300 à 1380		Affichage de la rotation
Lignes	5000 à 5000	à 5050	Affichages Affichage du titre du jeu
	5060 à 5110	à 5100	Définition des caractères spéciaux
	5110 à 5210	à 5200	Affichage des figures
	5210 à 5290	à 5290	Affichage des lettres indiquant les rotations

CARACTÈRES SPECIAUX

Nous avons défini les quatre caractères spéciaux suivants :

PUZZLE



4.3.3 Liste des instructions

```
10 REM *****
20 REM * PUZZLE *
30 REM *****
40 DIM X(20),Y(20)
50 H1=0
60 GOSUB 5000
70 REM Boucle principale
80 TX 1,0,0
90 CURSORX 1:CURSORY 20
100 INPUT"Choix du degré de difficulté (
1a5)";D$
110 FOR D=1 TO 500:NEXT D
120 CURSORY 20:PRINT CHR$(30)
130 V=VAL(D$)
140 IF V>0 AND V< 6 GOTO 200
150 CURSORX 10:CURSORY 24
160 PRINT "Choix impossible";
170 FOR D=1 TO 500:NEXT D
180 CURSORY 23:PRINT CHR$(30);
190 GOTO 80
200 REM Modification figure de gauche
210 FOR II=1 TO 2*V
220 REM Tirage rotation à effectuer
230 K=INT(RND(1)*12)+1
240 C=INT((K-1)/3)
250 K=K+C
260 REM Rotation de la figure de gauche
270 C=Y(K):Y(K)=Y(K+1)
280 Y(K+1)=Y(K+5):Y(K+5)=Y(K+4)
290 Y(K+4)=C
300 GOSUB 1320
310 NEXT II
320 REM Boucle de jeu
330 REM Lecture de la rotation à faire
340 TX 1,0,0:CURSORX 10:CURSORY 20
350 PRINT "Quel est votre choix?";
360 TX 4,0,1:PRINT CHR$(127);
```

```
370 IF KEY(0)=0 GOTO 370
380 TX 1,0,0:PRINT CHR$(2);
390 K=KEY(0)-96
400 IF K>0 AND K<13 GOTO 490
410 REM Coup impossible
420 CURSORY 24:PRINT CHR$(30);
430 CURSORX 20:CURSORY 24
440 PRINT "Choix impossible!";
450 FOR DD=1 TO 1000:NEXT DD
460 CURSORY 24 :PRINT CHR$(30);
470 GOTO 340
480 REM Effectuer la rotation demandée
490 PRINT CHR$(K+64);
500 C=INT((K-1)/3)
510 K=K+C
520 C=Y(K):Y(K)=Y(K+1)
530 Y(K+1)=Y(K+5):Y(K+5)=Y(K+4)
540 Y(K+4)=C
550 H1=0
560 GOSUB 1320
570 REM Test figure reconstituée
580 I1=1
590 IF X(I1)=Y(I1) GOTO 620
600 CURSORY 24:PRINT CHR$(30);
610 GOTO 340
620 I1=I1+1
630 IF I1<21 GOTO 590
640 CURSORY 20:PRINT CHR$(30);
650 TX 1,0,0
660 CURSORX 10:CURSORY 20:PRINT "Bravo,
vous avez gagné";:END
1000 REM Génération de la figure
1010 IF K=1 OR K=5 OR K=8 THEN X(K)=1111
: RETURN
1020 IF K=2 THEN X(K)=1222:RETURN
1030 IF K=6 OR K=13 OR K=16 OR K=19 THEN
X(K)=1211:RETURN
1040 IF K=4 THEN X(K)=1121:RETURN
1050 IF K=3 OR K=10 OR K=12 THEN X(K)=22
12:RETURN
1060 IF K=7 OR K=18 THEN X(K)=2221:RETUR
N
```



```
1070 IF K=9 OR K=14 THEN X(K)=1212:RETURN
1080 IF K=11 OR K=15 THEN X(K)=2222:RETU
RN
1090 IF K=17 THEN X(K)=1112:RETURN
1100 X(K)=2122:RETURN
1110 REM
1120 J=2*INT((K-1)/4)
1130 I=2*(K-2*J)+2
1140 J=J+5
1150 RETURN
1200 REM Tracé de la figure
1210 H=VAL(MID$(A$,2,1))-1:ET H,0,H1
1220 CURSORX I:CURSORY J:PRINT CHR$(50)
1230 H=VAL(MID$(A$,3,1))-1:ET H,0,H1
1240 CURSORX I+1:CURSORY J:PRINT CHR$(51
)
1250 H=VAL(MID$(A$,4,1))-1:ET H,0,H1
1260 CURSORX I:CURSORY J+1:PRINT CHR$(52
)
1270 H=VAL(MID$(A$,5,1))-1:ET H,0,H1
1280 CURSORX I+1:CURSORY J+1:PRINT CHR$(
53)
1290 RETURN
1300 REM Affichage de la rotation
1310 CURSORY 24:PRINT CHR$(30);
1320 HH=K
1330 FOR JJ=0 TO 5
1340 IF JJ=2 OR JJ=3 GOTO 1370
1350 K=HH+JJ:GOSUB 1120
1360 A$=STR$(Y(K)):GOSUB 1210
1370 NEXT JJ:K=HH
1380 RETURN
5000 REM Affichage du titre du jeu
5010 INIT 3,6:TX 4,7,0
5020 CURSORX 14:CURSORY2
5030 PRINT "P u z z l e "
5040 CURSORX 14:CURSORY3
5050 PRINT "P u z z l e "
5060 REM Caractères spéciaux
5070 SETET 50,"007F7F7F7F7F7F7F7F7F7F"
5080 SETET 51,"00FEFEFEFEFEFEFEFEFE"
5090 SETET 52,"7F7F7F7F7F7F7F7F7F7F00"
```

```
5100 SETET 53,"FEFEFEFEFEFEFEFEFE00"  
5110 REM Tracé des deux figures et du  
5120 REM rectangle central  
5130 TX 0,0,0  
5140 FOR K=1 TO 20  
5150 GOSUB 1010:Y(K)=X(K)  
5160 GOSUB 1120:A#=STR$(X(K))  
5170 GOSUB 1210:I=I+24:GOSUB 1210  
5180 A#=STR$(1111)  
5190 I=I-12:GOSUB 1210  
5200 NEXT K  
5210 REM Afficher les lettres indiquant  
5220 REM les douze rotations possibles  
5230 TX 1,0,0  
5240 FOR J2=0 TO 3  
5250 FOR I2=0 TO 2  
5260 II=3*J2+I2+65  
5270 CURSORX 2*I2+17:CURSORY 2*J2+6:PRIN  
T CHR$(II)  
5280 NEXT I2:NEXTJ2  
5290 RETURN
```

4.4 CAVALIER

4.4.1 Présentation

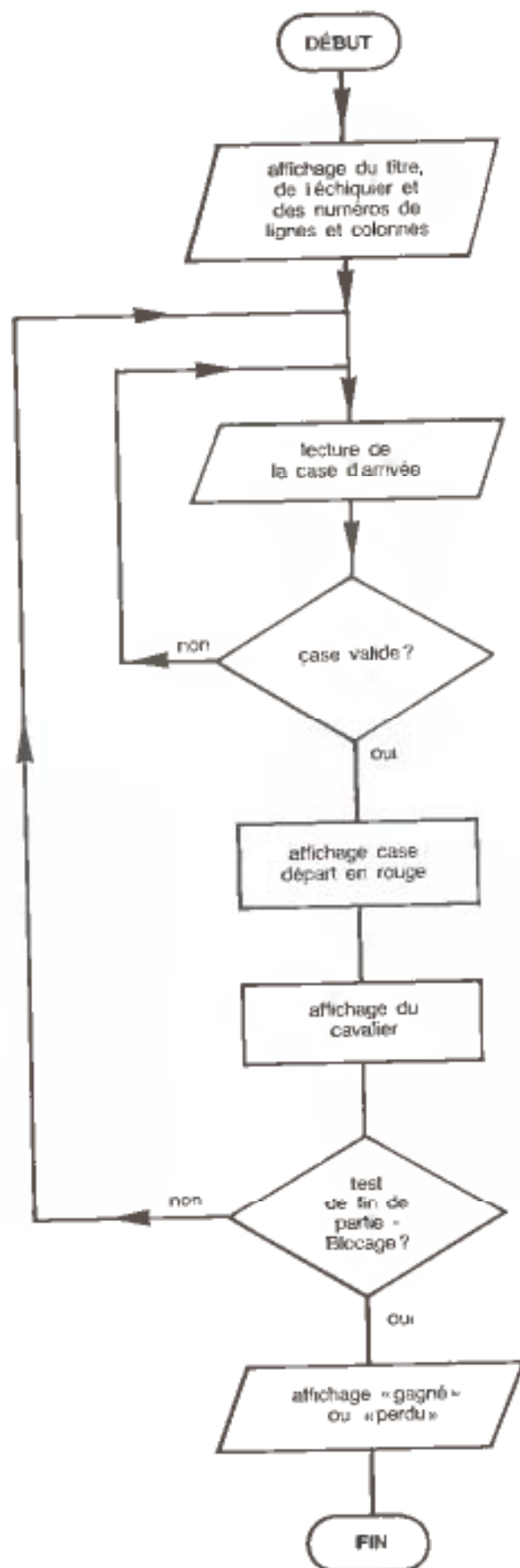
Les échecs fournissent un nombre colossal de problèmes, indépendants du jeu lui-même. De nombreux problèmes tiennent au déplacement en L du cavalier. Il s'agit pour le joueur, en partant d'un endroit quelconque de l'échiquier, de déplacer le cavalier de telle manière que celui-ci passe une fois - et une seule - par toutes les cases de l'échiquier.

4.4.2 Organigramme et étude du programme

STRUCTURE

Lignes	40 à	90	Affichage du titre
Lignes	100 à	140	Définition des caractères spéciaux
Lignes	150 à	250	Initialisation et affichage de l'échiquier
Lignes	260 à	290	Affichage des lettres indiquant les colonnes
Lignes	300 à	330	Affichage des numéros de la ligne
Lignes	360 à	530	Demande de la case d'arrivée et test de validité
Lignes	540 à	620	Coup joué

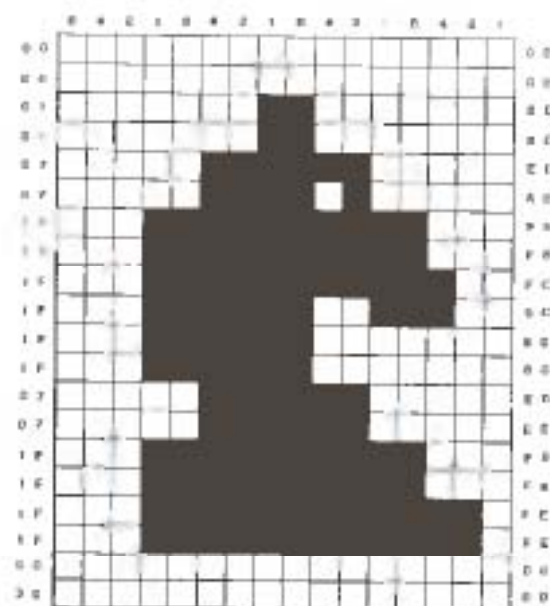
CAVALIER



- Lignes 630 à 730 Test de fin de partie
- Lignes 740 à 780 Fin de partie et affichage du message correspondant (gagné/perdu)
- Lignes 790 à 880 Affichage d'une case de l'échiquier
- Lignes 890 à 940 Affichage "case choisie"
- Lignes 1000 à 1090 Affichage du cavalier

CARACTERES SPECIAUX

Nous utilisons les huit caractères suivants :



CAVALIER



CASE D'ECHEQUIER

4.4.3 Liste des instructions

```
10 REM *****
20 REM * Cavalier *
30 REM *****
40 INIT 3,1:TX 4,7,0
50 CURSORX 12:CURSORY 0
60 PRINT "CCaavvaalliieerr"
70 CURSORX 12:CURSORY 1
80 PRINT "CCaavvaalliieerr"
90 TX 0,0,0
100 REM Définition caractères spéciaux
110 SETET 110,"0000010107071F1F1F1F"
115 SETET 111,"1F1F07071F1F1F1F0000"
120 SETET 112,"00008080E0A0F8F8FC9C"
125 SETET 113,"8080E0E0F8F8FEFE0000"
130 SETET 120,"007F7F7F7F7F7F7F7F"
135 SETET 121,"7F7F7F7F7F7F7F7F00"
140 SETET 122,"00FEFEFEFEFEFEFEFEFE"
145 SETET 123,"FEFEFEFEFEFEFEFEFE00"
150 REM Initialisations et affichage
160 REM de l'échiquier
170 DIM X(8,8)
180 FOR I=0 TO 7
190 FOR J=0 TO 7
200 X(I,J)=1
210 K=I+J-2*INT((I+J)/2)
220 IF K=0 THEN ET 0,0,0
230 IF K=1 THEN ET 7,0,0
240 II=2*I:JJ=2*J:GOSUB 800
250 NEXT J:NEXT I
260 REM Affichage des lettres
270 REM indiquant les colonnes
280 CURSORX 13:CURSORY 3:TX1,0,0
290 PRINT "A B C D E F G H ":CURSORY 5
300 REM Affichage des numéros de ligne
310 FOR I=1 TO 8
320 CURSORX 10:PRINT STR$(I):PRINT
330 NEXT I
340 REM Début du jeu
```

```
350 C=0
360 REM Demande de la case d'arrivée
370 CURSORX 22:CURSORY 23:TX1,0,0
380 INPUT "Votre choix ";X$
390 CURSORY 21:PRINT CHR$(30)
400 IF LEN(X$)<>2 GOTO 450
410 A$=LEFT$(X$,1) :H=ASC(A$)-65
420 A$=RIGHT$(X$,1):V=VAL(A$)-1
430 IF H>=0 AND H<8 AND V>=0 AND V<8 GOT
0 500
440 REM Choix impossible
450 CURSORY 22:CURSORX 23
460 TX1,0,0:PRINT "Choix impossible";
470 FOR D=1 TO 2000:NEXT D
480 CURSORY 22:PRINT CHR$(30);
490 GOTO 360
500 IF C=0 GOTO 570
510 IF ABS(H-I)=2 AND ABS(V-J)=1GOTO 550
520 IF ABS(H-I)=1 AND ABS(V-J)=2GOTO 550
530 GOTO 450
540 REM Choix possible
550 IF X(H,V)=0 GOTO 900
560 REM Ancienne position devient rouge
570 ET 1,0,0
580 IFC<>0THEN II=2*I:JJ=2*J:GOSUB 800
590 REM Nouvelle position devient bleue
600 ET 4,0,0
610 II=2*H:JJ=2*V:GOSUB 1000
620 X(H,V)=0:C=C+1:ET 5,0,0
630 REM Test de fin de partie
640 S=0:I=H:J=V
650 IF 1<I AND 0<J THEN S=S+X(I-2,J-1)
660 IF 1<I AND J<7 THEN S=S+X(I-2,J+1)
670 IF 0<I AND J<6 THEN S=S+X(I-1,J+2)
680 IF 0<I AND 1<J THEN S=S+X(I-1,J-2)
690 IF I<6 AND 0<J THEN S=S+X(I+2,J-1)
700 IF I<6 AND J<7 THEN S=S+X(I+2,J+1)
710 IF I<7 AND J<6 THEN S=S+X(I+1,J+2)
720 IF I<7 AND 1<J THEN S=S+X(I+1,J-2)
730 IF S<>0 GOTO 360
740 REM Blocage du jeu
750 TX0,0,0:CURSORX 14:CURSORY 22
```

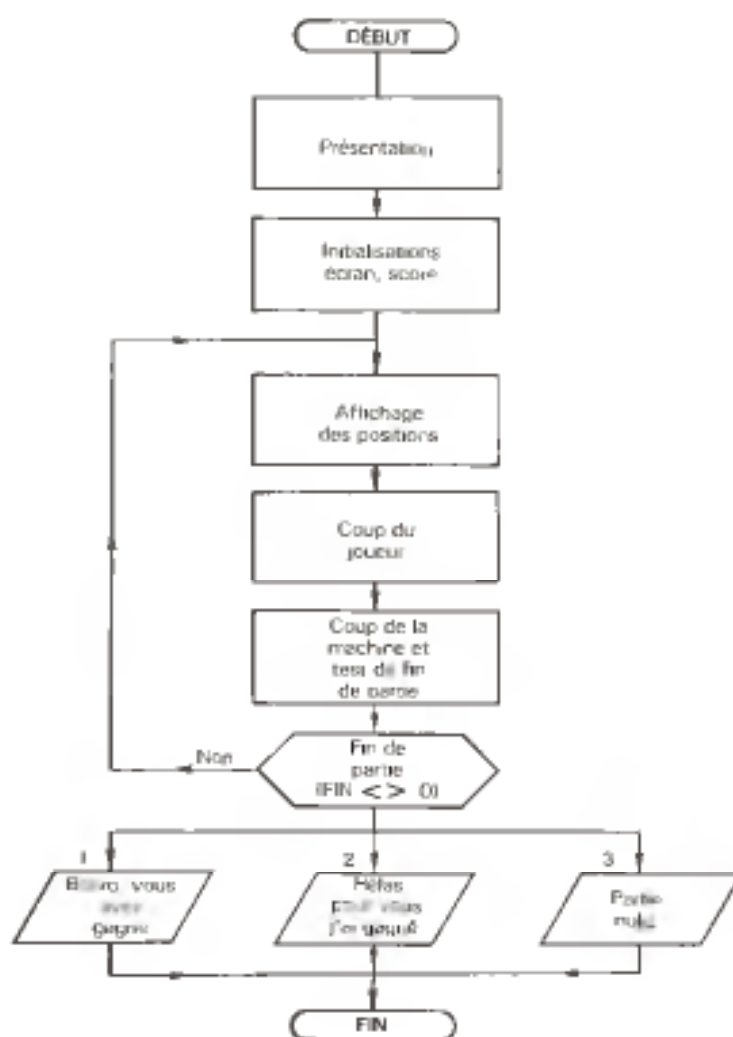
```
760 IF C= 64 THEN PRINT "Gagné"
770 IF C<>64 THEN PRINT "Perdu"
780 END
790 REM Affichage d'une case échiquier
800 CURSORX II+12:CURSORY JJ+4
810 PRINT CHR$(120);
820 CURSORX II+13:CURSORY JJ+4
830 PRINT CHR$(122);
840 CURSORX II+12:CURSORY JJ+5
850 PRINT CHR$(121);
860 CURSORX II+13:CURSORY JJ+5
870 PRINT CHR$(123);
880 RETURN
890 REM Affichage "déjà choisi"
900 CURSORX 23:CURSORY 24
910 PRINT "déjà choisi";
920 FOR D=1 TO 2000:NEXT D
930 CURSORY 24 :PRINT CHR$(30);
940 GOTO 360
1000 REM Affichage du cavalier
1010 CURSORX II+12:CURSORY JJ+4
1020 PRINT CHR$(110);
1030 CURSORX II+13:CURSORY JJ+4
1040 PRINT CHR$(112);
1050 CURSORX II+12:CURSORY JJ+5
1060 PRINT CHR$(111);
1070 CURSORX II+13:CURSORY JJ+5
1080 PRINT CHR$(113);
1090 RETURN
```

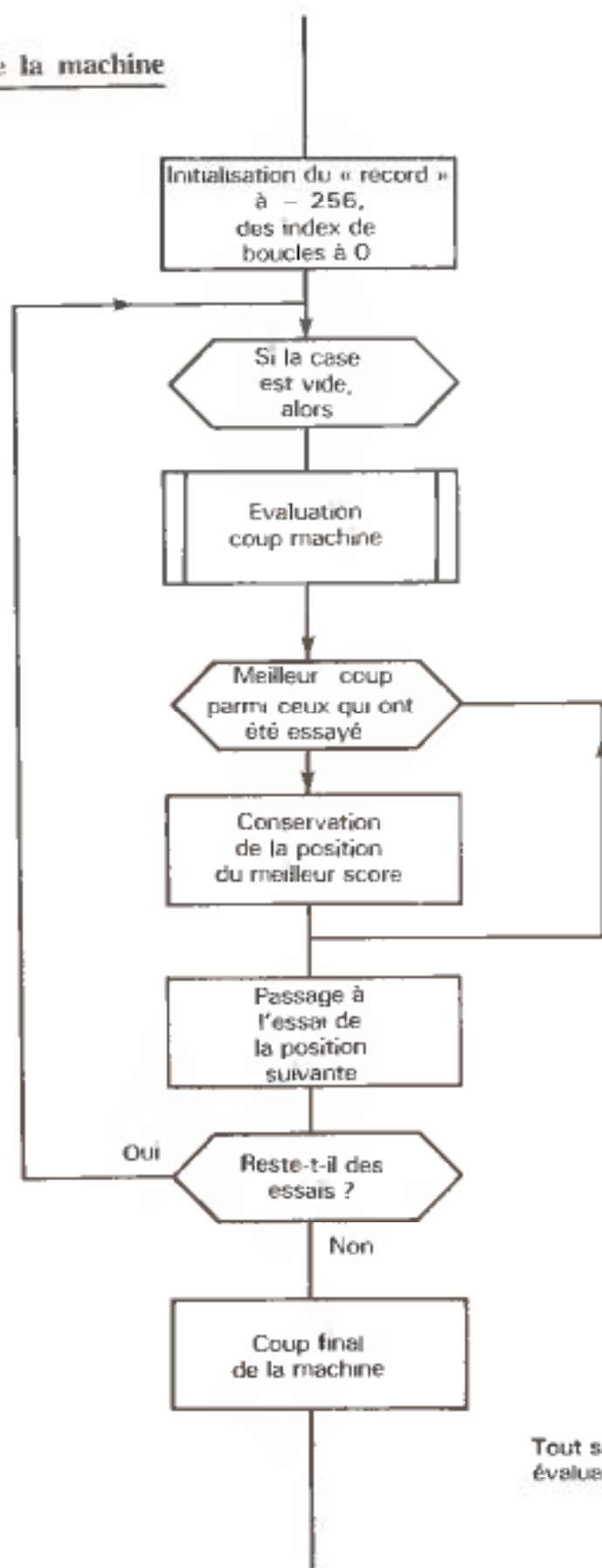

4.5 TIC-TAC-TOE

4.5.1 Présentation

Tic-Tac-Toe est un jeu de réflexion, qui se joue au clavier, entre un homme et une machine. Ne vous inquiétez pas : un adversaire, si bon soit-il, peut commettre des erreurs. Le jeu se déroule sur une grille de 3 * 3 cases où chaque joueur positionne un jeton à tour de rôle dans une case vide. Le but du jeu est d'aligner trois points identiques dans l'une des quatre orientations suivantes : horizontale, verticale, et les deux diagonales. Vous utiliserez vous-même des jetons marqués X, et la machine des jetons marqués O.

4.5.2 Organigramme et étude du programme



Tic-Tac-Toe (2) : Coup de la machine

Tout se passe dans
évaluation machine

Ce programme, qui fait jouer l'homme contre la machine, utilise donc un sous-programme de détermination du coup de la machine qui est situé aux lignes 1000 à 1100. Comme vous le montre le second organigramme ci-dessus, il n'est rien d'autre qu'une simple boucle qui essaye tous les coups possibles pour la machine. Votre ordinateur appelle chaque fois un sous-programme d'évaluation de la grille, qui fournit une note. Ce sous-programme d'évaluation teste automatiquement les fins de partie. Il contient tout l'algorithme du jeu de la machine.

Lignes	70 à	170	Coup du joueur
Lignes	180 à	250	Coup de la machine (appel du sous-programme)
Lignes	1000 à	1110	Essai de chaque coup par la machine pour trouver le meilleur
Lignes	300 à	990	Sous-programme d'évaluation des coups de la machine et des tests de fin de partie. Il est la clef du coup de l'ordinateur.
	320		Analyse par ligne
	400 à	480	Analyse par colonne
	490 à	580	Test 1ère diagonale
	590 à	680	Test 2ème diagonale
	690 à	740	Résultats
Lignes	1500 à	1550	Affichage des positions
Lignes	5000 à	5110	Présentation
Lignes	5200 à	5499	Initialisation et dessin de la grille et des chiffres
Lignes	10000 à	10160	Fin de partie

PRINCIPALES VARIABLES

LJ, LC	Position (ligne, colonne) du joueur
CC, LC	Index de boucle du sous-programme d'évaluation des coups de la machine
JO, OD, A, B	Résultats intermédiaires durant l'évaluation des coups de la machine
ECRAN (i,j)	Copie des coups précédents : 1 coup joueur 2 coups machine

4.5.3 Liste des instructions

```

10 REM *****
20 REM Tic Tac Toe
30 REM *****
40 GOSUB 5000:GOSUB 5200
50 REM Coup du joueur
70 GOSUB 1500
80 CURSORX 1:CURSORY 20:PLAY "O2A"
90 INPUT "Votre coup :ligne+col.";R$
100 IF LEN(R$)<>2 THEN 80
110 LJ=VAL(LEFT$(R$,1))-1
120 IF LJ<0 OR LJ>2 THEN 80
130 CJ=VAL(RIGHT$(R$,1))-1
140 IF CJ<0 OR CJ>2 THEN 80
150 IF ECRAN(CJ,LJ)=0 THEN 180
160 PRINT "Case déjà occupée"
170 PLAY"O1T1AAA":GOTO 70
180 REM Passage au coup du VG 5000
190 CURSORX 1:CURSORY 21
200 PRINT "          "
210 ECRAN(CJ,LJ)=1
220 GOSUB 1500
230 GOSUB 1000
240 IF FIN<>0 GOTO 10000
250 GOTO 70

```

```
300 REM Evaluation des coups
310 FOR I=0 TO 3:A(I)=0:B(I)=0:NEXT I
320 GOSUB 800
400 REM Analyse par colonne
410 FOR CC=0 TO 2:JO=0:OD=0
420 FOR LC=0 TO 2
430 IF ECRAN(LC,CC)=1 THEN JO=JO+1
440 IF ECRAN(LC,CC)=2 THEN OD=OD+1
450 NEXT LC
460 GOSUB 900
480 NEXT CC
490 REM Test diagonale
500 OD=0:JO=0
510 FOR CC=0 TO 2
520 IF ECRAN(CC,CC)=0 GOTO 560
530 IF ECRAN(CC,CC)=2 GOTO 550
540 OD=OD+1:GOTO 560
550 JO=JO+1
560 NEXT CC
570 IF JO=0 THEN A(OD)=A(OD)+1
580 IF OD=0 THEN B(JO)=B(JO)+1
590 REM Test seconde diagonale
600 OD=0:JO=0
610 FOR CC=0 TO 2
620 IF ECRAN(2-CC,CC)=0 GOTO 660
630 IF ECRAN(2-CC,CC)=2 GOTO 650
640 OD=OD+1:GOTO 660
650 JO=JO+1
660 NEXT CC
670 IF JO=0 THEN A(OD)=A(OD)+1
680 IF OD=0 THEN B(JO)=B(JO)+1
690 REM Resultats
700 IF A(3)=1 THEN FIN=1:RETURN
710 IF B(3)=1 THEN FIN=2:RETURN
720 R=-63*A(2)+31*B(2)
730 R=R-15*A(1)+7*B(1)
740 RETURN
800 REM Analyse par ligne
810 FOR CC=0 TO 2:JO=0:OD=0
820 FOR LC=0 TO 2
830 IF ECRAN(CC,LC)=1 THEN JO=JO+1
840 IF ECRAN(CC,LC)=2 THEN OD=OD+1
```

```

850 NEXT LC:GOSUB 900
860 NEXT CC
870 RETURN
900 REM Evaluation coup machine
910 IF JO<>0 GOTO 930
920 B(OD)=B(OD)+1
930 IF OD<>0 THEN RETURN
940 A(JO)=A(JO)+1
950 RETURN
1000 REM Essai de chaque coup
1010 S=-256:TEST=1
1020 FOR L=0 TO 2:FOR C=0 TO 2
1030 IF ECRAN(C,L)<>0 GOTO 1080
1040 TEST=0:ECRAN(C,L)=2:GOSUB 300
1050 IF FIN<>0 THEN RETURN
1060 IF R>S THEN S=R:CM=C:LM=L
1070 ECRAN(C,L)=0
1080 NEXT C:NEXT L
1090 ECRAN(CM,LM)=2
1100 IF TEST=1 THEN FIN=3
1110 RETURN
1500 REM Affichage des positions
1510 FOR LC=0 TO 2:FOR CC=0 TO 2
1520 CURSORX 15+CC*4:CURSORY 6+LC*4
1530 PRINT B$(ECRAN(CC,LC));
1540 NEXT CC:NEXT LC
1550 RETURN
5000 REM Présentation
5010 INIT 7:TX 1,7,0
5020 CURSORX 9:CURSORY 8
5030 PRINT "T I C   T A C   T O E "
5040 CURSORX 9:CURSORY 9
5050 PRINT "T I C   T A C   T O E "
5060 CURSORX 1:CURSORY 20:TX 0,0,0
5070 PRINT "RETURN pour commencer";
5090 INPUT R$
5100 INIT 3,3
5110 RETURN
5200 REM Dessin des initialisations
5210 DIM ECRAN(2,2),B$(2)
5220 DIM A(4),B(4)
5230 INIT 3:TX 0,0,0:PAGE

```

```
5240 CURSORX 14:CURSORY 8
5250 FOR I=0 TO 10:PRINT CHR$(127);:NEXT
  I
5260 CURSORX 14:CURSORY 12
5270 FOR I=0 TO 10:PRINT CHR$(127);:NEXT
  I
5280 FOR I=0 TO 10
5290 CURSORX 17:CURSORY 5+I
5300 PRINT CHR$(127)
5310 NEXT I
5320 FOR I=0 TO 10
5330 CURSORX 21:CURSORY 5+I
5340 PRINT CHR$(127)
5350 NEXT I
5360 FOR I=1 TO 3:CURSORX 10+I*4
5370 CURSORY 4:PRINT I;:NEXT I
5380 FOR I=1 TO 3:CURSORY 2+I*4
5390 CURSORX 10:PRINT I*10:NEXT I
5400 REM
5410 FOR I=0 TO 2
5420 READ B$(I)
5430 NEXT I
5440 DATA " ", "X", "0"
5450 FOR C=0 TO 2
5460 FOR L=0 TO 2
5470 ECRAN(C,L)=0
5480 NEXT L
5490 NEXT C
5500 RETURN
10000 REM Fin de partie
10010 CURSORX 1:CURSORY 20:TX 0,0,0
10020 PRINT "
  "
10050 CURSORX 1:CURSORY 20:TX 0,0,0
10060 ON FIN-1 GOTO 10090, 10120
10070 PRINT "Bravo, vous avez gagné !":P
LAY "01T1ABCDEFG"
10080 GOTO 10130
10090 PRINT "Hélas pour vous, j'ai gagné
  !":PLAY "01T10GFEDAAA"
10100 GOSUB 1500:CURSORX 1:CURSORY 21
10110 GOTO 10130
```

```
10120 PRINT "Partie nulle !":PLAY "AAA"  
10130 REM Autre partie  
10140 INPUT "Une autre partie ";R$  
10150 IF R$="0" OR R$="o" THEN RUN  
10160 INIT 6,6:END
```


4.6 BATAILLE NAVALE

4.6.1 Présentation

La bataille navale est un jeu bien connu et auquel vous avez sûrement déjà joué. Il consiste à repérer et à couler une flotte ennemie en un minimum de temps. Cette flotte est composée de 10 navires de tailles différentes :

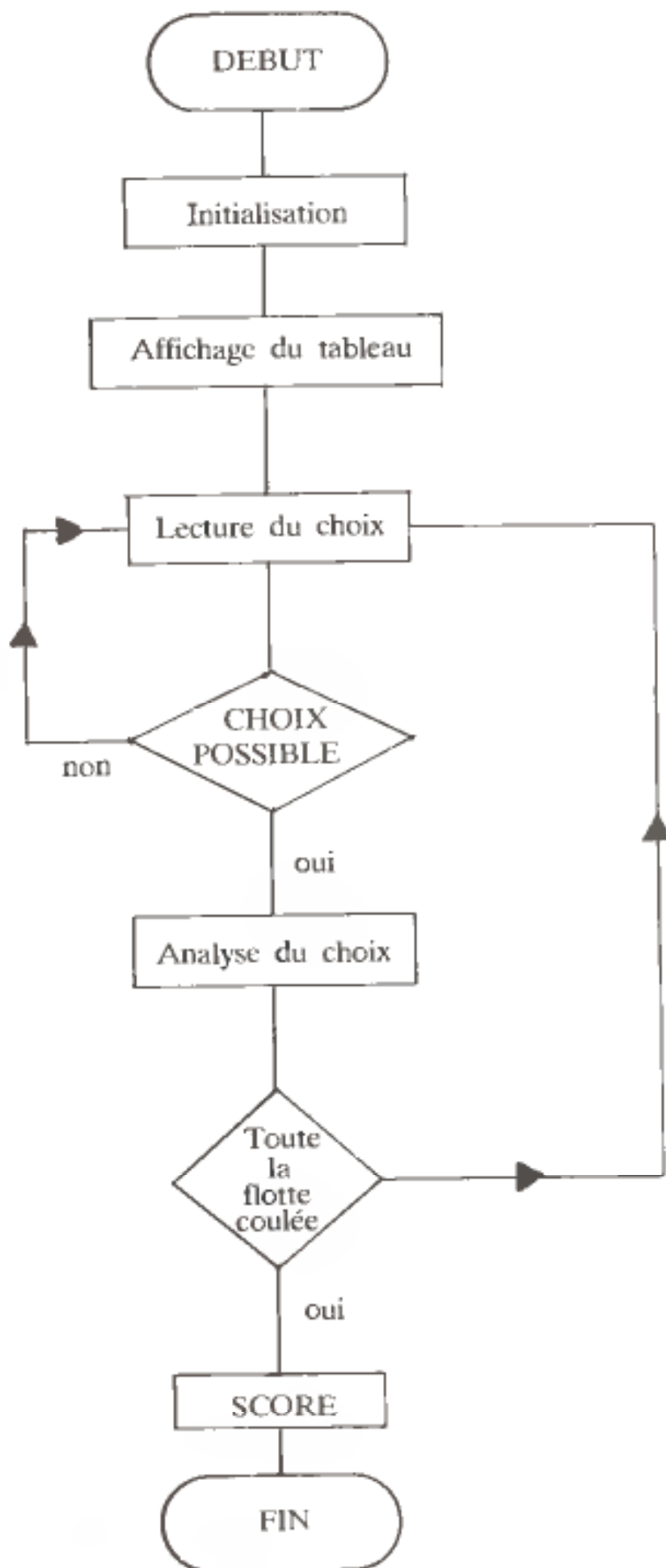
- 1 navire de 4 cases
- 2 navires de 3 cases
- 3 navires de 2 cases
- 4 navires de 1 case

Ce programme répartit de manière totalement aléatoire les 10 navires qui peuvent être disposés soit horizontalement, soit verticalement. L'analyse de votre coup est détaillée : selon le cas, votre VG 5000 répondra "déjà choisi", "raté", "touché", "coulé". En fin de partie, votre ordinateur vous donnera votre score, qui sera bien entendu proportionnel au nombre de coups nécessaires pour détruire la flotte ennemie.

4.6.2 Organigramme et étude du programme

STRUCTURE DU PROGRAMME

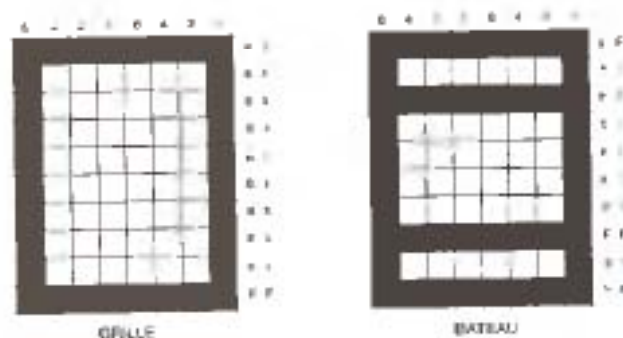
- Lignes 50 à 70 Définition des caractères spéciaux
- Lignes 80 à 150 Affichage de la légende
- Lignes 160 à 310 Tracé des exemples de navires
- Lignes 320 à 390 Tracé de la grille



- Lignes 400 à 460 Affichage des lettres indiquant les colonnes
- Lignes 470 à 510 Affichage des numéros de lignes
- Lignes 520 à 580 Tirage des bateaux (appel du sous-programme)
- Lignes 590 à 640 Boucle de jeu
- Lignes 650 à 860 Test de fin de partie et affichage du message correspondant au nombre de coups joués
- Lignes 870 à 1350 Sous-programme traitant le coup du joueur et affichant le message correspondant (raté, touché, coulé)
- Lignes 1360 à 1480 Lecture du coup du joueur avec test de validité
- Lignes 1490 à 1660 Tirage de la position du bateau 4X
- Lignes 1670 à 1730 Tirage des positions, bateaux 3X
- Lignes 1740 à 1800 Tirage des positions, bateaux 2X
- Lignes 1810 à 1940 Tirage des positions, bateaux 1X
- Lignes 1950 à 2030 Sous-programme utilisé pour le tirage des bateaux 3X
- Lignes 2040 à 2140 Sous-programme utilisé pour le tirage des bateaux 3X
- Lignes 2150 à 2250 Sous-programme utilisé pour le tirage des bateaux 2X
- Lignes 2260 à 2360 Sous-programme utilisé pour le tirage des bateaux 2X

CARACTERES SPECIAUX

Nous avons défini les deux caractères suivants :



4.6.3 Liste des instructions

```

10 REM *****
20 REM * Bataille navale *
30 REM *****
40 DIM T(14,18)
50 REM Caractères spéciaux
60 SETET 120,"FF8181818181818181FF"
70 SETET 121,"FF81FF8181818181FF81FF"
80 REM Affichage de la légende
90 INIT 6,3:TX0,0,0
100 CURSORX 26:CURSORY 3:PRINT"Flotte"
110 CURSORX 26:CURSORY 4:PRINT"ennemie"
120 CURSORX 26:CURSORY 6:PRINT"1X"
130 CURSORX 26:CURSORY 10:PRINT"2X"
140 CURSORX 26:CURSORY 14:PRINT"3X"
150 CURSORX 26:CURSORY 18:PRINT"4X"
160 REM Tracé des exemples de navires
170 ET1,0,0
180 FOR J=1 TO 4
190 CURSORX 25+J:CURSORY 7
200 PRINT CHR$(121)
210 NEXT J
220 FOR J=1 TO 3
230 CURSORX 25+J:CURSORY 11

```

```
240 PRINT CHR$(121)
250 NEXT J
260 FOR J=1 TO 2
270 CURSORX 25+J:CURSORY 15
280 PRINT CHR$(121)
290 NEXT J
300 CURSORX 26:CURSORY 19
310 PRINT CHR$(121)
320 REM Tracé de la grille
330 ET4,0,0
340 FOR I=3 TO 19
350 FOR K=10 TO 22
360 CURSORY I:CURSORX K
370 PRINT CHR$(120)
380 NEXT K
390 NEXT I
400 REM Affichage des lettres indiquant
410 REM les colonnes
420 TX0,0,0
430 FOR J=0 TO 12
440 CURSORY 2:CURSORX 10+J
450 PRINT CHR$(65+J)
460 NEXT J
470 REM Affichage des numéros de lignes
480 FOR J=1 TO 17
490 CURSORX 7:CURSORY J+2
500 PRINT J
510 NEXT J
520 REM Début du jeu
530 REM Tirage des bateaux
540 GOSUB 1500:REM 4X
550 GOSUB 1680:REM 3X
560 GOSUB 1750:REM 2X
570 GOSUB 1820:REM 1X
580 A=0:B=0
590 REM Boucle de jeu
600 FOR S=1 TO 1000:NEXT S
610 CURSORX 1:CURSORY 22:PRINT CHR$(30)
620 GOSUB 1380
630 IF YY=1 THEN YY=0 :GOTO 600
640 GOSUB 880
650 REM Test de fin de partie
```

```
660 IF A<>20 THEN GOTO 600
670 TX0,0,0:PRINT CHR$(31)
680 CURSORX 18:CURSORY 10
690 PRINT "Score"
700 CURSORX 15:CURSORY 12
710 PRINT "Coups joués:";B
720 IF B>79 GOTO 760
730 CURSORX 10:CURSORY 16
740 PRINT "Bravo, vous êtes très fort"
750 GOTO 860
760 IF B>99 GOTO 800
770 CURSORX 10:CURSORY 16
780 PRINT "Vous avez du flair...."
790 GOTO 860
800 IF B>=120 GOTO 840
810 CURSORX 10:CURSORY 16
820 PRINT "Vous n'êtes pas trop nul"
830 GOTO 860
840 CURSORX 10:CURSORY 16
850 PRINT "Ne vous découragez pas"
860 CURSORY 23:END
870 REM Messages de réponse au joueur
880 IF T(I,J)=6 THEN T(I,J)=8:CURSORX 17
:CURSORY 24:PRINT"déjà choisi";:B=B+1
890 IF T(I,J)=7 THEN T(I,J)=8:CURSORX 17
:CURSORY 24:PRINT"déjà choisi";:B=B+1
900 IF T(I,J)=8 THEN CURSORX 17:CURSORY
24:PRINT"déjà choisi";:B=B+1
910 IF T(I,J)<>0 AND T(I,J)<>5 GOTO 970
920 REM T(I,J)=0 ou T(I,J)=5
930 T(I,J)=8:B=B+1
940 CURSORX 17:CURSORY 24:PRINT"raté";
950 CURSORX I+9:CURSORY J+2
960 TX4,0,0:PRINT CHR$(127)
970 IF T(I,J)<>1 GOTO 1010
980 REM T(I,J)=1
990 T(I,J)=7:A=A+1:B=B+1
1000 CURSORX 17:CURSORY 24:PRINT"Coulé";
1010 IF T(I,J)=21 THEN GOSUB 1120:B=B+1
1020 IF T(I,J)=22 THEN GOSUB 1160:B=B+1
1030 IF T(I,J)=23 THEN GOSUB 1200:B=B+1
1040 IF T(I,J)=31 THEN GOSUB 1240:B=B+1
```

```
1050 IF T(I,J)=32 THEN GOSUB 1280:B=B+1
1060 IF T(I,J)=4 THEN GOSUB 1320:B=B+1
1070 IF T(I,J)<>6AND T(I,J)<>7 GOTO 1110
1080 REM T(I,J)=6 ou T(I,J)=7
1090 CURSORX I+9:CURSORY J+2
1100 ET 1,0,0:PRINT CHR$(121)
1110 RETURN
1120 T(I,J)=6 :A=A+1 :Z1=Z1+1
1130 IF Z1=2 THEN T(I,J)=7:CURSORX 17:CURSORY 24:PRINT "Coulé";
1140 IFZ1<>2 THEN CURSORX 17:CURSORY 24:PRINT "Touché";
1150 RETURN
1160 T(I,J)=6:A=A+1:Z2=Z2+1:
1170 IFZ2=2THEN T(I,J)=7:CURSORX 17:CURSORY 24:PRINT "Coulé";
1180 IFZ2<>2THEN CURSORX 17:CURSORY 24:PRINT "Touché";
1190 RETURN
1200 T(I,J)=6 :A=A+1 :Z3=Z3+1
1210 IFZ3=2 THEN T(I,J)=7:CURSORX 17:CURSORY 24:PRINT"Coulé";
1220 IFZ3<>2 THEN CURSORX 17:CURSORY 24:PRINT "Touché";
1230 RETURN
1240 T(I,J)=6:A=A+1:Z4=Z4+1
1250 IFZ4=3 THEN T(I,J)=7:CURSORX 17:CURSORY 24:PRINT "Coulé";
1260 IFZ4<>3 THEN CURSORX 17:CURSORY 24:PRINT "Touché";
1270 RETURN
1280 T(I,J)=6: A=A+1:Z5=Z5+1
1290 IFZ5=3THEN T(I,J)=7:CURSORX 17:CURSORY 24:PRINT"coulé";
1300 IFZ5<>3 THEN CURSORX 17:CURSORY 24:PRINT "touché";
1310 RETURN
1320 T(I,J)=6 :A=A+1 :Z6= Z6+1
1330 IFZ6=4 THEN T(I,J)=7: CURSORX 17:CURSORY 24:PRINT "coulé";
1340 IFZ6<>4 THEN CURSORX 17:CURSORY 24:PRINT "touché";
```

```
1350 RETURN
1360 REM Lecture du coup du joueur
1370 REM avec test de validité du coup
1380 TX0,0,0:CURSORX 1:CURSORY 22:INPUT
"Quel est votre choix ";A$
1390 IF LEN(A$)<>3 AND LEN(A$)<>2 THEN
  YY=1: GOTO 1460
1400 I=ASC(A$)-64
1410 IF LEN(A$)=3 THEN J=VAL(RIGHT$(A$,2
))
1420 IF LEN(A$)=2 THEN J=VAL(RIGHT$(A$,1
))
1430 IF LEN(A$)=3 OR LEN(A$)=2 THEN GOTO
1440
1440 IF J<1 OR J>17 THEN YY=1
1450 IF I<1 OR I>13 THEN YY=1
1460 IF YY=1 THEN CURSORX 16:CURSORY 24:
PRINT "Choix impossible";
1470 A$="XX"
1480 RETURN
1490 REM Tirage du bateau 4X
1500 Q=INT(RND(1)*140)+1
1510 I=Q-10*INT(Q/10)
1520 IF I=0 THEN I=10
1530 J=(Q-1)/(10)+1:W=INT(RND(1)*2)+1
1540 IF W<>1 GOTO 1600
1550 FOR K=-1 TO 4
1560 T(I+K,J-1)=5:T(I+K,J+1)=5
1570 T(I+K,J)=4
1580 NEXT K
1590 T(I-1,J)=5:T(I+4,J)=5
1600 IF W<>2 THEN RETURN
1610 FOR K=-1 TO 4
1620 T(I-1,J+K)=5:T(I+1,J+K)=5
1630 T(I,J+K)=4
1640 NEXT K
1650 T(I,J-1)=5:T(I,J+4)=5
1660 RETURN
1670 REM Tirage des bateaux 3X
1680 FOR L=1 TO 2
1690 H=INT(RND(1)*2)+1
1700 IF H=1 THEN GOSUB 1950
```



```
1710 IFH<>1 THEN GOSUB 2040
1720 NEXT L
1730 RETURN
1740 REM Tirage des bateaux 2X
1750 FOR L=1 TO 3
1760 Y=INT(RND(1)*2)+1
1770 IF Y=1 THEN GOSUB 2150
1780 IF Y=2 THEN GOSUB 2260
1790 NEXT L
1800 RETURN
1810 REM Tirage des bateaux 1X
1820 FOR L=1 TO 4
1830 U=INT(RND(1)*221)+1
1840 I=U-13*INT(U/13)
1850 IF I=0 THEN I=13
1860 J=(U-I)/(13)+1
1870 IF T(I,J)<>0 THEN GOTO 1830
1880 FOR K=-1 TO 1
1890 T(I+K,J-1)=5:T(I+K,J+1)=5
1900 T(I+K,J)=1
1910 NEXT K
1920 T(I-1,J)=5:T(I+1,J)=5
1930 NEXT L
1940 RETURN
1950 O=INT(RND(1)*165)+1:I=O-11*INT(O/11)
:IF I=0 THEN I=11
1960 J=(O-I)/(11)+1
1970 IF T(I,J)<>0 OR T(I+1,J)<>0 OR T(I+
2,J)<>0 THEN GOTO 1950
1980 FOR K=-1 TO 3
1990 T(I+K,J-1)=5:T(I+K,J+1)=5
2000 T(I+K,J)=30+L
2010 NEXT K
2020 T(I-1,J)=5:T(I+3,J)=5
2030 RETURN
2040 O=INT(RND(1)*165)+1
2050 I=O-11*INT(O/11)
2060 IF I=0 THEN I=11
2070 J=(O-I)/(11)+1
2080 IF T(I,J)<>0 OR T(I,J+1)<>0 OR T(I,
J+2)<>0 THEN GOTO 2040
2090 FOR K=-1 TO 3
```

```
2100 T(I-1,J+K)=5:T(I+1,J+K)=5
2110 T(I,J+K)=30+L
2120 NEXT K
2130 T(I,J-1)=5:T(I,J+3)=5
2140 RETURN
2150 D=INT(RND(1)*192)+1
2160 I=D-12*INT(D/12)
2170 IF I=0 THEN I=12
2180 J=(D-I)/(12)+1
2190 IF T(I,J)<>0 OR T(I+1,J)<>0 THEN G
OTO 2150
2200 FOR K=-1 TO 2
2210 T(I+K,J-1)=5:T(I+K,J+1)=5
2220 T(I+K,J)=20+L
2230 NEXT K
2240 T(I-1,J)=5:T(I+2,J)=5
2250 RETURN
2260 D=INT(RND(1)*192)+1
2270 I=D-12*INT(D/12)
2280 IF I=0 THEN I=12
2290 J=(D-I)/(12)+1
2300 IF T(I,J)<>0 OR T(I,J+1)<>0 THEN G
OTO 2260
2310 FOR K=-1 TO 2
2320 T(I-1,J+K)=5:T(I+1,J+K)=5
2330 T(I,J+K)=20+L
2340 NEXT K
2350 T(I,J-1)=5:T(I,J+2)=5
2360 RETURN
```

4.7 KARO

4.7.1 Présentation

KARO est un jeu de réflexion original. Vous avez devant vous un quadrillage de neuf cases, repérées horizontalement par 1, 2, 3 et verticalement par A, B, C, dont certaines sont allumées et les autres éteintes.

But du jeu : reconstituer cette figure en allumant des cases selon certains principes.

* Si vous appuyez sur la case centrale (B2), la croix entière s'allume (A2, B2, C2, B1, B3).

* Si vous appuyez sur un coin, le carré adjacent s'allume. Par exemple, A1 allume A1, A2, B1, B2.

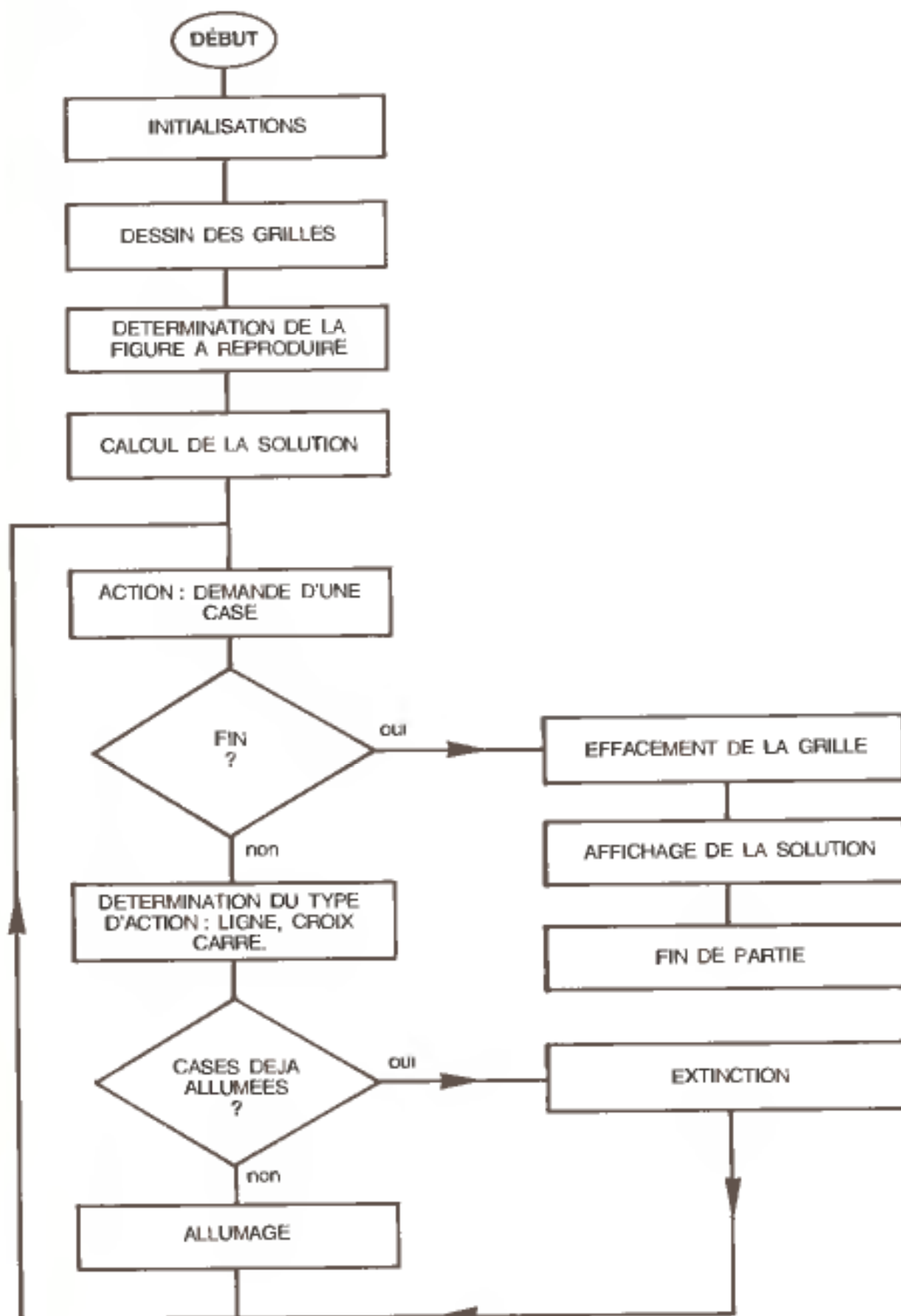
* Si vous appuyez sur le centre de l'un des côtés, vous allumez tout le côté correspondant. Par exemple, A2 allume A1, A2 et A3.

* Si vous activez une case déjà allumée, elle s'éteint.

Vous le verrez, c'est bien moins facile qu'il n'y paraît. Vous pourrez aussi choisir vous-même la figure à reproduire.

Mais l'intérêt de ce jeu ne s'arrête pas là : il contient lui-même un module donnant, pour chaque combinaison, la solution optimale. Un véritable petit problème mathématique que vous pourrez chercher à résoudre... Essayez de comprendre la manière dont ce module a été conçu ! Bravo à qui trouvera la solution... Mais attendez tout de même d'être devenu suffisamment fort pour résoudre cette énigme.

4.7.2 Organigramme et étude du programme



Lignes	60 à	80	Boucle principale
Lignes	100 à	270	Sous-programme de tracé des grilles
Lignes	300 à	390	Sous-programme de modification d'une case (allume la case si elle est éteinte et réciproquement)
Lignes	400 à	590	Choix d'une case par le joueur
Lignes	600 à	790	Sous-programme de création de la figure à reproduire (par le joueur ou aléatoire)
	630 à	720	Figure aléatoire
	730 à	790	Figure créée
Lignes	800 à	960	Sous-programme exécutant l'action correspondant à la case jouée
Lignes	1000 à	1110	Calcul de la solution
Lignes	1200 à	1290	Initialisation du tableau solution
Lignes	5000 à	5230	Présentation, mode d'emploi, choix de la figure
Lignes	6000 à	6140	Initialisations et décors
	6000 à	6030	Initialisation diverses
	6040 à	6060	Tracé des grilles
	6070 à	6090	Initialisation du tableau définissant l'état des cases
	6100		Appel du sous-programmes d'initialisation du tableau solution. Appel du sous-programme de définition de la grille à reproduire

6110 à 6140	Initialisation du tableau définissant l'état des cases pour la seconde grille
Lignes 10000 à 10250	Fin de partie, affichage de la solution

PRINCIPALES VARIABLES

SO (9,9)	Tableau permettant de calculer la solution. A vous de comprendre comment il fonctionne.
C (3,3)	Etat des cases de la grille (allumées ou éteintes)
A, B	Coordonnées de l'origine de la grille 1, 0 première grille 20, 10 seconde grille
NC	Nombre de coups joués

4.7.3 Liste des instructions

```

10 REM *****
20 REM * KARO *
30 REM *****
40 GOSUB 5000
50 GOSUB 6000
60 GOSUB 400:GOSUB 800
70 NC=NC+1
80 GOTO 60
100 REM Tracé du cadre
110 RESTORE
120 ET 0,0,0
130 FOR I=1 TO 19:FOR J=1 TO 13 STEP4
140 CURSORX A+I:CURSORY B+J
150 PRINT CHR$(110);
160 NEXTJ:NEXTI
170 FOR I=1 TO 13:FOR J=1 TO 19 STEP6
180 CURSORX A+J:CURSORY I+B

```

```
190 PRINT CHR$(110);
200 NEXTJ:NEXT I
210 TX 0,0,0
220 CURSORX B:FOR I=3 TO 15 STEP6
230 READ R$:CURSORX A+I:PRINT R$;
240 NEXT I
250 FOR I=3 TO 11 STEP4:READ R$:CURSORX
A:CURSORX B+I:PRINT R$;
260 NEXT I
270 RETURN
300 REM Allumage d'une case
310 IF C(X,Y)=1 THEN C(X,Y)=0:C=4:GOTO 3
30
320 C=1:C(X,Y)=1
330 ET C,0,0
340 FOR I=0 TO 4:FOR J=0 TO 2
350 CURSORX 2+(X-1)*6+I+A
360 CURSORX 2+(Y-1)*4+J+B
370 PRINT CHR$(110);
380 NEXT J:NEXT I
390 RETURN
400 REM Choix d'une case par le joueur
410 TX 0,0,0
420 CURSORX 1 :CURSORX 20:TT$="F"
430 IF MW=1 THEN TT$="S"
440 PRINT "POUR ARRETER : ";TT$
450 CURSORX 25:CURSORX 0
460 PRINT "Case ?";
470 CURSORX 25:CURSORX 1
480 INPUT CA$
490 IF CA$="F" AND MW<>1 THEN 10010
500 YY$=LEFT$(CA$,1):Y=1
510 IF YY$="B" THEN Y=2
520 IF YY$="C" THEN Y=3
530 X=VAL(RIGHT$(CA$,1))
540 CURSORX 25:CURSORX 1
550 PRINT " ";
560 IF CA$="S" AND MW=1 THEN RETURN
570 IF Y=1 AND YY$<>"A" THEN 400
580 IF X>3 OR X=0 THEN 400
590 RETURN
600 REM Choix de la figure
```

```
610 IF FI$="N" GOTO 630
620 GOTO 730
630 REM Figure aléatoire
640 A=1:B=0
650 FOR X=1 TO 3
660 FOR Y=1 TO 3
670 NB=INT(RND(1)*2):IF NB=2 THEN NB=1
680 IF NB=1 THEN GOSUB 300:GOSUB 1000
690 NEXT Y
700 NEXT X
710 GOSUB 1070
720 RETURN
730 REM Création de la figure
740 A=1:B=0:WW=1
750 FOR U=1 TO 9
760 GOSUB 400:IF CA$="S" THEN 780
770 GOSUB 300:GOSUB 1000:NEXT U
780 GOSUB 1070:WW=0
790 RETURN
800 REM Action relative à la case jouée
810 IF X=2 AND Y<>2 THEN GOSUB 870
820 IF X=2 AND Y=2 THEN GOSUB 890
830 IF Y=2 AND X<>2 THEN GOSUB 880
840 IF X=2 OR Y=2 THEN RETURN
850 GOSUB 930
860 RETURN
870 FOR X=1 TO 3:GOSUB 300:NEXT X:X=2:RE
TURN
880 FOR Y=1 TO 3:GOSUB 300:NEXT Y:Y=2:RE
TURN
890 FOR X=1 TO 3:GOSUB 300:NEXT X
900 Y=1:X=2:GOSUB 300
910 Y=3:GOSUB 300
920 RETURN
930 GOSUB 300:XX=X:X=2:GOSUB 300
940 Y=2:GOSUB 300
950 X=XX:GOSUB 300
960 RETURN
1000 REM Calcul de la solution
1010 J=3*(Y-1)+X
1020 FOR I=1 TO 9
1030 SO(I,J)=SO(I,J)+SO(I,KJ)
```



```
1040 NEXT I
1050 KJ=J
1060 RETURN
1070 REM CALCUL DE LA SOLUTION
1080 FOR I=1 TO 9
1090 SO(I,9)=SGN(SO(I,KJ)-2*INT(SO(I,KJ)
/2))
1100 NEXT I
1110 RETURN
1200 REM Initialisation du tableau sol.
1210 FOR I=1 TO 9:FOR J=1 TO 9
1220 X=INT((I-1)/3):Y=INT((J-1)/3)
1230 U=(X-Y)^2+(I-J-3*X+3*Y)^2
1240 K=3*ABS(SGN(I-5))-2*((-1)^SGN(I-2*I
NT(I/2)))
1250 U=INT(U):K=INT(K)
1260 IF U<>K AND U<>K+3 THEN SO(J,I)=1
1270 NEXT J:NEXT I
1280 RETURN
1290 DATA 1,2,3,A,B,C
5000 REM Présentation
5010 INIT 3,0:TX 4,7,0
5020 CURSORX 15:CURSORY 0
5030 PRINT "K a r o "
5040 CURSORX 15:CURSORY 1
5050 PRINT "K a r o "
5060 TX 0,0,0:PRINT :PRINT
5070 PRINT "But du jeu: reconstituer une
  figure"
5080 PRINT "placée dans une grille de ne
uf cases"
5090 PRINT "avec les principes suivants"
5100 PRINT :PRINT
5110 PRINT "B2 active A2 B2 C2 B1 B3"
5120 PRINT "A1 active A1 A2 B1 B2"
5130 PRINT "A3 active A2 A3 B2 B3"
5140 PRINT "C1 active C1 C2 B1 B2"
5150 PRINT "C3 active C2 C3 B2 B3"
5160 PRINT "A2 active A1 A2 A3"
5170 PRINT "C2 active C1 C2 C3"
5180 PRINT "B1 active A1 B1 C1"
5190 PRINT "B3 active A3 B3 C3"
```

```

5200 PRINT :PRINT :PRINT
5210 INPUT "Voulez-vous créer votre figur
e (O/N) ";FI$
5220 IF FI$("<>")"0" AND FI$("<>")"N" GOTO 5210
5230 RETURN
6000 REM Initialisations
6010 DIM SO(9,9),C(3,3)
6020 INIT 7,2
6030 A=1:B=0:GOSUB 100
6040 A=20:B=10:GOSUB 100
6050 SETET 110."FFFFFFFFFFFFFFFFFFFFFFF"
6060 A=1:B=0
6070 FOR X=1 TO 3:FOR Y=1 TO 3
6080 C(X,Y)=1:GOSUB 300
6090 NEXT Y:NEXT X
6100 GOSUB 1200:GOSUB 600:REM Choix figu
re
6110 A=20:B=10
6120 FOR X=1 TO 3:FOR Y=1 TO 3
6130 C(X,Y)=1:GOSUB 300
6140 NEXT Y:NEXT X
6150 RETURN
10000 REM Fin de partie
10010 PLAY "AAABADDDDDAAEEEEFFG"
10020 FOR X=1 TO 3:FOR Y=1 TO 3
10030 C(X,Y)=1:GOSUB 300
10040 NEXT Y:NEXT X
10050 REM Affichage de la solution
10060 TX 0,0,0
10070 CURSORX 25:CURSORY 0:PRINT "
";
10080 CURSORX 25:CURSORY 1:PRINT "
";
10090 FOR U=1 TO 9
10100 TX 0,0,0
10110 Y=INT((U-1)/3)+1:X=U-3*(Y-1)
10120 CA$="C":IF Y=1 THEN CA$="A"
10130 IF Y=2 THEN CA$="B"
10140 CA$=CA$+RIGHT$(STR$(X),1)
10150 IF SO(U,9)=1 THEN CURSORX 25:CURSO
RY U-1:PRINT CA$+"
";:PLAY "AB":G
OSUB 800

```

```
10160 NEXT U
10170 CURSORX 1:CURSORY 0:TX 0,0,0
10180 PRINT "Nombre de coups :";NC;
10190 CURSORX 1:CURSORY 1
10200 PRINT "Autre partie (O/N)";
10210 INPUT NP$
10220 NP$=LEFT$(NP$,1)
10230 IF NP$="0" OR NP$="o" THEN RUN
10240 CURSORX 1:CURSORY 2:PRINT "Au revo
ir !"
10250 END
```

4.8 POKER

4.8.1 Présentation

Les règles de ce jeu très connu sont simples. Au début vous disposez de 100 francs. Après que vous avez misé une somme comprise entre 1 et 10 francs, l'ordinateur vous distribue cinq cartes parmi trente-deux. Vous pouvez alors en changer autant que vous voudrez, mais en une seule fois.

- Si vous obtenez une QUINTE ROYALE (cinq cartes de même couleur : 10, valet, dame, roi, as), vous gagnez mille fois votre mise.
- Si vous obtenez une QUINTE FLUSH (cinq cartes de la même couleur et qui se suivent) vous gagnez cent fois votre mise.
- Si vous obtenez un CARRE (quatre cartes portant le même numéro) vous gagnez quarante fois votre mise.
- Si vous obtenez un FULL (brelan et paire), vous gagnez vingt fois votre mise.
- Si vous obtenez une COULEUR, (cinq cartes de la même couleur), vous gagnez dix fois votre mise.
- Si vous obtenez une SUITE (cinq cartes qui se suivent), vous gagnez cinq fois votre mise.
- Si vous obtenez un BRELAN (trois cartes portant le même numéro), vous gagnez trois fois votre mise.
- Si vous obtenez deux paires, vous êtes remboursé.

- Si vous obtenez une PAIRE (deux cartes portant le même numéro), ou moins, ou rien du tout, vous perdez votre mise.

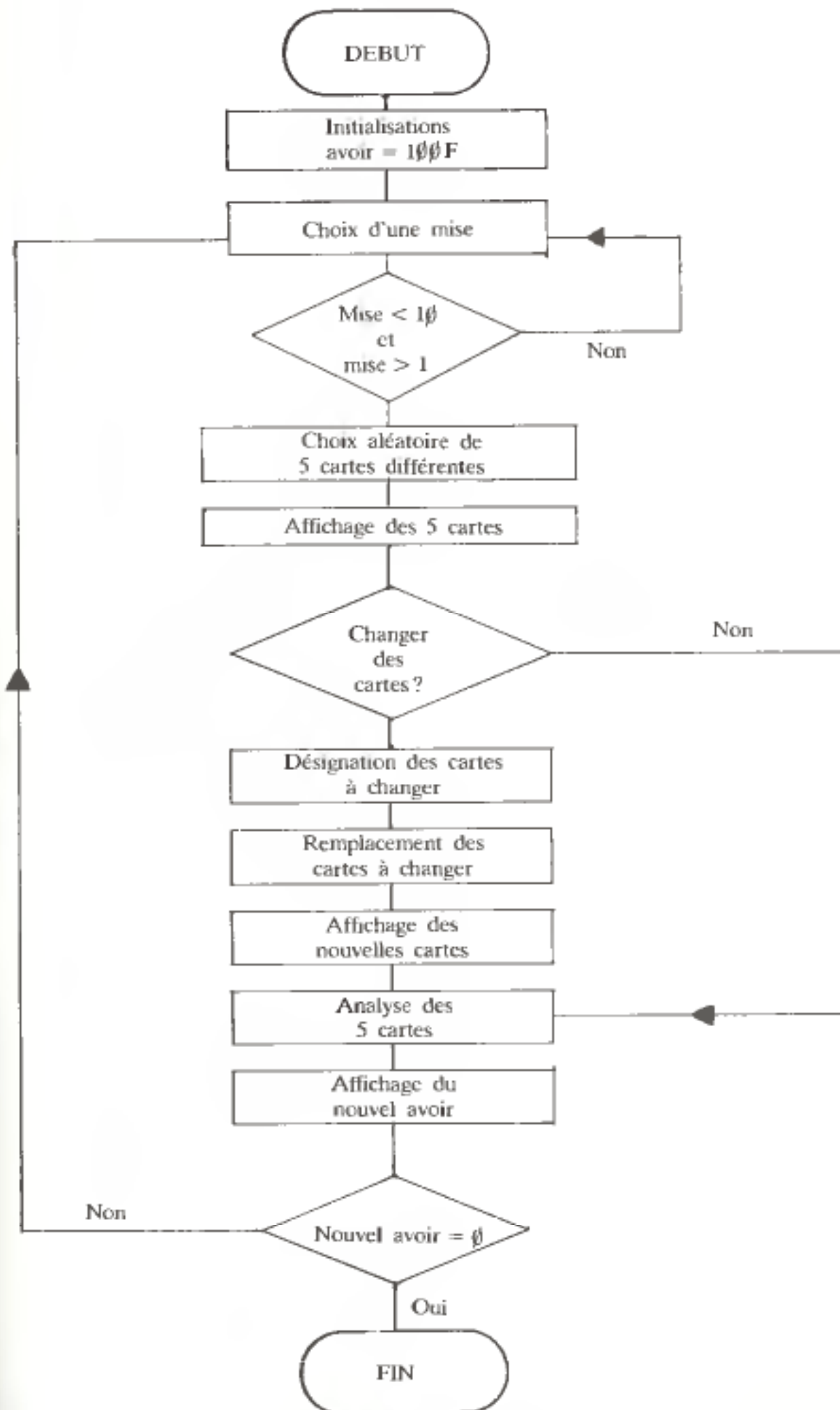
L'ordinateur affiche alors votre nouvel avoir. Selon le cas, il l'augmente ou le diminue. Une nouvelle partie est ensuite lancée.

Le jeu s'arrête lorsque votre avoir s'annule.

4.8.2 Organigramme et étude du programme

STRUCTURE

Lignes	70 à 320	Affichage des gains possibles
Ligne	330	Test de fin de partie
Lignes	340 à 390	Fin de partie
Lignes	400 à 470	Lecture de la mise du joueur et test de validité
Lignes	480 à 540	Définition des caractères spéciaux
Lignes	550 à 730	Tracé des "cadres" des cartes remplies (dos des cartes)
Lignes	740 à 820	Affichage des numéros des cartes
Lignes	830 à 900	Tirage et affichage des cartes
Lignes	910-960	Demande de changement de cartes
Ligne	970	Appel du sous-programme de calcul du résultat
Ligne	980	Retour en 90 = début de la boucle de jeu
Lignes	990 à 1010	Tirage de la valeur d'une carte (sous-programme)



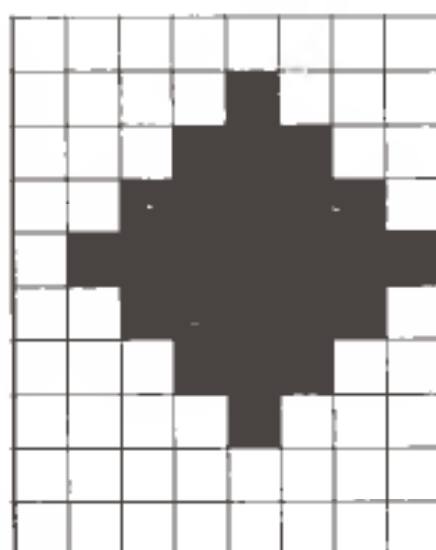
Lignes 1020 à 1390 Affichage des cartes (sous-programme)

Lignes 1400 à 1790 Sous-programme de changement de cartes

Lignes 1800 à 2180 Calcul et affichage du résultat

CARACTERES SPECIAUX

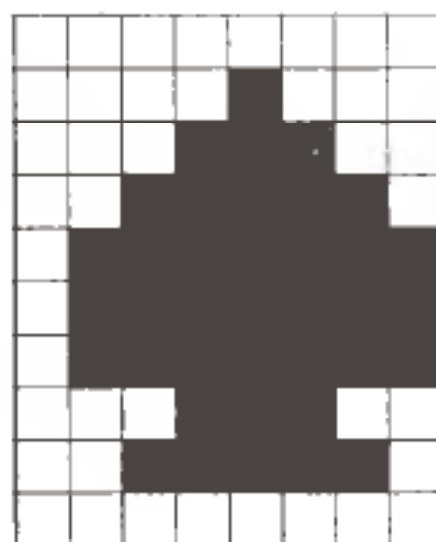
Les quatre caractères suivants (trèfle, carreau, coeur, pique) ont été définis :



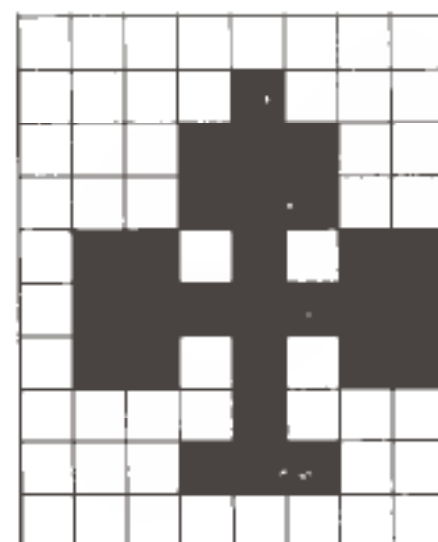
CARREAU



CŒUR



PIQUE



TREFLE

4.8.3 Liste des instructions

```
10 REM *****
20 REM * Poker *
30 REM *****
40 INIT2,2
50 DIM A(5),B(5),T(5)
60 S1=100
70 REM Affichage des gains possibles
80 CURSORX 34:CURSORY 1:PRINT"Gain"
90 CURSORX 34:CURSORY 3:PRINT STR$(S1)
100 CURSORX 14:CURSORY 1:PRINT"Poker"
110 CURSORX 1:CURSORY 4
120 PRINT"Pour une mise de 1F, ";
130 PRINT"vous gagnez:"
140 CURSORX 1:CURSORY 6
150 PRINT"Quinte royale 1000F"
160 CURSORX 1:CURSORY 8
170 PRINT"Quinte flush 100F"
180 CURSORX 1:CURSORY 10
190 PRINT"Carré 40F"
200 CURSORX 1:CURSORY 12
210 PRINT"Full 20F"
220 CURSORX 1:CURSORY 14
230 PRINT"Couleur 10F"
240 CURSORX 1:CURSORY 16
250 PRINT"Suite 5F"
260 CURSORX 1:CURSORY 18
270 PRINT"Brelan 3F"
280 CURSORX 1:CURSORY 20
290 PRINT"2 paires 1F"
300 CURSORX 1:CURSORY 22
310 PRINT"1 paire 0F"
320 RESTORE 790:FOR F=1 TO1000:NEXTF
330 IF S1<>0 GOTO 410
340 REM Avoir nul = fin de partie
350 FOR H=0 TO 1000:NEXT H
360 PRINT CHR$(31)
370 CURSORX 16:CURSORY 12
380 PRINT "Au revoir"
```



```
390 FOR T=0 TO 1000:NEXT T:END
400 REM Entrée de la mise du joueur
410 CURSORX 1:CURSORY 23
420 INPUT"Quelle est votre mise";M$
430 M1=VAL(M$)
440 IF M1>0 AND M1<11AND S1>=M1GOTO 470
450 CURSORX 1:CURSORY 23
460 PRINT CHR$(30):GOTO 410
470 FOR T=0 TO 1000:NEXTT:S1=S1-M1
480 REM Définition caractères spéciaux
490 SETET 83,"00081C3E7F7F7F1C3E00"
500 SETET 81,"00081C3E7F3E1C080000"
510 SETET 82,"00081C1C6B7F6B081C00"
520 SETET 80,"0000081C3E7F7F7F1C00"
530 SETET 84,"2669A9292929292926"
540 SETET 79,"55AA55AA55AA55AA55AA"
550 PRINT CHR$(31)
560 TX0,0,0
570 FOR F=3 TO 36 STEP 7
580 CURSORX F:CURSORY 7:PRINT "-----"
590 CURSORX F:CURSORY 15
600 FOR S=1 TO 5:PRINT CHR$(126);:NEXTS
610 NEXT F
620 FOR T=8 TO 14
630 CURSORX 2:CURSORY T
640 PRINT CHR$(125);
650 NEXT T
660 FOR T=8 TO 14
670 CURSORX 36:CURSORY T
680 PRINT CHR$(123);
690 NEXT T
700 FOR P=8 TO 29 STEP 7:FOR T=8 TO14
710 CURSORX P:CURSORY T
720 PRINT CHR$(123);CHR$(125);
730 NEXT T:NEXT P
740 ET1,0,0
750 FOR J=8 TO 14 :FOR S=3 TO 31 STEP7
760 CURSORX S:CURSORY J:PRINT"00000";
770 NEXT S,J
780 FOR GL=0 TO1000:NEXTGL
790 DATA 49,50,51,52,53
800 TX0,0,0: FOR J=5 TO 33 STEP7:READX
```

```
810 CURSORX J:CURSORY 16:PRINT CHR$(X)
820 NEXT J
830 K=0:GOSUB 990
840 K=1:GOSUB 990:IF T(1)=T(0) GOTO 840
850 K=2:GOSUB 990:IF T(2)=T(1) OR T(2)=T
(0) GOTO 850
860 K=3:GOSUB 990:IF T(3)=T(2) OR T(3)=T
(1) OR T(3)=T(0) GOTO 860
870 K=4:GOSUB 990:IF T(4)=T(3) OR T(4)=T
(2) OR T(4)=T(1) OR T(4)=T(0) GOTO 870
880 FOR J=8 TO 14:FOR S=3 TO 31 STEP 7:CUR
SORX S:CURSORY J:PRINT CHR$(32);CHR$(32)
;CHR$(32);CHR$(32);CHR$(32)
890 NEXTS:NEXTJ
900 GOSUB 1020
910 TX0,0,0:CURSORX 2:CURSORY 20
920 INPUT"Changez-vous des cartes (O/N)
";X$
930 IF X$="O" THEN GOSUB 1420:RESTORE 10
20:GOSUB 1020
940 IF X$="O" OR X$="N" GOTO 970
950 CURSORX 2:CURSORY 20
960 PRINT CHR$(30):GOTO 920
970 GOSUB 1810
980 GOTO 90
990 I=INT(RND(1)*8)+1:J=INT(RND(1)*4)+1
1000 T(K)=I+10*J
1010 RETURN
1020 DATA 5,12,19,26,33
1030 FOR J=0 TO 4: B=INT(T(J)/10):READY
1040 IF B=1 OR B=2 THEN ET1,0,0
1050 IF B<>1 AND B<>2 THEN ET0,0,0
1060 CURSORX Y:CURSORY 11:PRINT CHR$(B+7
9);:NEXT J
1070 FOR J=0 TO 4
1080 V=79+INT(T(J)/10):W=3+(7*J):Z=W+4
1090 IF V=80 OR V=81 THEN ET1,0,0:CURSOR
X W:CURSORY 9:PRINT CHR$(V):CURSORX W:CU
RSORY 13:PRINT CHR$(V)
1100 IF V=80 OR V=81 THEN ET1,0,0:CURSOR
X Z:CURSORY 9:PRINT CHR$(V):CURSORX Z:CU
RSORY 13:PRINT CHR$(V)
```

```
1110 IF V=82 OR V=83 THEN ET0,0,0:CURSOR
X W:CURSOR Y 9:PRINT CHR$(V):CURSOR X W:CU
RSOR Y 13:PRINT CHR$(V)
1120 IF V=82 OR V=83 THEN ET0,0,0:CURSOR
X Z:CURSOR Y 9:PRINT CHR$(V):CURSOR X Z:CU
RSOR Y 13:PRINT CHR$(V)
1130 C=T(J)-10*INT(T(J)/10)
1140 IF C<4 AND (V=82 OR V=83) THEN TX0,
0,0:CURSOR X W:CURSOR Y 14:PRINT CHR$(54+C
):CURSOR X W:CURSOR Y 8:PRINT CHR$(54+C)
1150 IF C<4 AND (V=82 OR V=83) THEN TX0,
0,0:CURSOR X Z:CURSOR Y 8:PRINT CHR$(54+C)
:CURSOR X Z:CURSOR Y 14:PRINT CHR$(54+C)
1160 IF C<4 AND (V=80 OR V=81) THEN TX1,
0,0:CURSOR X W:CURSOR Y 14:PRINT CHR$(54+C
):CURSOR X W:CURSOR Y 8:PRINT CHR$(54+C)
1170 IF C<4 AND (V=80 OR V=81) THEN TX1,
0,0:CURSOR X Z:CURSOR Y 8:PRINT CHR$(54+C)
:CURSOR X Z:CURSOR Y 14:PRINT CHR$(54+C)
1180 IF C=4 AND (V=80 OR V=81) THEN ET1,0
,0:CURSOR X W:CURSOR Y 8:PRINT CHR$(84):CU
RSOR X W:CURSOR Y 14:PRINT CHR$(84)
1190 IF C=4 AND (V=80 OR V=81) THEN ET1,
0,0:CURSOR X Z:CURSOR Y 8:PRINT CHR$(84):C
URSOR X Z:CURSOR Y 14:PRINT CHR$(84)
1200 IF C=4 AND (V=82 OR V=83) THEN ET0,
0,0:CURSOR X Z:CURSOR Y 8:PRINT CHR$(84):C
URSOR X Z:CURSOR Y 14:PRINT CHR$(84)
1210 IF C=4 AND (V=82 OR V=83) THEN ET0,0
,0:CURSOR X W:CURSOR Y 8:PRINT CHR$(84):CU
RSOR X W:CURSOR Y 14:PRINT CHR$(84)
1220 IF C=5 AND (V=80 OR V=81) THEN TX1,0
,0:CURSOR X W:CURSOR Y 8:PRINT CHR$(86):CU
RSOR X W:CURSOR Y 14:PRINT CHR$(86)
1230 IF C=5 AND (V=80 OR V=81) THEN TX1,
0,0:CURSOR X Z:CURSOR Y 8:PRINT CHR$(86):C
URSOR X Z:CURSOR Y 14:PRINT CHR$(86)
1240 IF C=5 AND (V=82 OR V=83) THEN TX0,0
,0:CURSOR X W:CURSOR Y 8:PRINT CHR$(86):CU
RSOR X W:CURSOR Y 14:PRINT CHR$(86)
1250 IF C=5 AND (V=82 OR V=83) THEN TX0,
0,0:CURSOR X Z:CURSOR Y 8:PRINT CHR$(86):C
URSOR X Z:CURSOR Y 14:PRINT CHR$(86)
```

```
1260 IF C=6 AND(V=80 OR V=81) THEN TX1,0
,0:CURSORX W:CURSOR Y 8:PRINT CHR$(68):CU
RSORX W:CURSOR Y 14:PRINT CHR$(68)
1270 IF C=6 AND (V=80 OR V=81) THEN TX1,
0,0:CURSORX Z:CURSOR Y 8:PRINT CHR$(68):C
URSORX Z:CURSOR Y 14:PRINT CHR$(68)
1280 IF C=6 AND(V=82 OR V=83) THEN TX0,0
,0:CURSORX W:CURSOR Y 8:PRINT CHR$(68):CU
RSORX W:CURSOR Y 14:PRINT CHR$(68)
1290 IF C=6 AND (V=82 OR V=83) THEN TX0,
0,0:CURSORX Z:CURSOR Y 8:PRINT CHR$(68):C
URSORX Z:CURSOR Y 14:PRINT CHR$(68)
1300 IF C=7 AND(V=80 OR V=81) THEN TX1,0
,0:CURSORX W:CURSOR Y 8:PRINT CHR$(82):CU
RSORX W:CURSOR Y 14:PRINT CHR$(82)
1310 IF C=7 AND (V=80 OR V=81) THEN TX1,
0,0:CURSORX Z:CURSOR Y 8:PRINT CHR$(82):C
URSORX Z:CURSOR Y 14:PRINT CHR$(82)
1320 IF C=7 AND(V=82 OR V=83) THEN TX0,0
,0:CURSORX W:CURSOR Y 8:PRINT CHR$(82):CU
RSORX W:CURSOR Y 14:PRINT CHR$(82)
1330 IF C=7 AND (V=82 OR V=83) THEN TX0,
0,0:CURSORX Z:CURSOR Y 8:PRINT CHR$(82):C
URSORX Z:CURSOR Y 14:PRINT CHR$(82)
1340 IF C=8 AND(V=80 OR V=81) THEN TX1,0
,0:CURSORX W:CURSOR Y 8:PRINT CHR$(49):CU
RSORX W:CURSOR Y 14:PRINT CHR$(49)
1350 IF C=8 AND (V=80 OR V=81) THEN TX1,
0,0:CURSORX Z:CURSOR Y 8:PRINT CHR$(49):C
URSORX Z:CURSOR Y 14:PRINT CHR$(49)
1360 IF C=8 AND(V=82 OR V=83) THEN TX0,0
,0:CURSORX W:CURSOR Y 8:PRINT CHR$(49):CU
RSORX W:CURSOR Y 14:PRINT CHR$(49)
1370 IF C=8 AND (V=82 OR V=83) THEN TX0,
0,0:CURSORX Z:CURSOR Y 8:PRINT CHR$(49);;
CURSORX Z:CURSOR Y 14:PRINT CHR$(49);
1380 NEXT J
1390 RETURN
1400 REM Changement de cartes
1410 CURSORX 1:CURSOR Y 22:TX 0,0,0
1420 PRINT"Pour changer les cartes 5,2,3
faites"
```

```
1430 INPUT "523";A$
1440 C=LEN(A$):I=1
1450 IF I=C+1 AND C<>0 GOTO 1490
1460 W=VAL(MID$(A$,I,1)):I=I+1
1470 IF W<1 OR W>5 THEN CURSORX 1:CURSOR
Y 20:PRINT CHR$(30):GOTO 1420
1480 GOTO 1450
1490 K=LEN(A$)
1500 FOR I=1 TO K
1510 Y$=MID$(A$,I,1)
1520 J=ASC(Y$)-49
1530 FOR H=8 TO 14
1540 CURSORX 7*J+3:CURSORY H
1550 ET 1,0,0:PRINT "00000";
1560 NEXT H
1570 NEXT I
1580 FOR P=1 TO 500:NEXT P
1590 K=LEN(A$)
1600 FOR Q=1 TO K
1610 Y$=MID$(A$,Q,1)
1620 J=ASC(Y$)-49
1630 G=T(J)
1640 L1=(J+1)-INT((J+1)/5)*5
1650 L2=(J+2)-INT((J+2)/5)*5
1660 L3=(J+3)-INT((J+3)/5)*5
1670 L4=(J+4)-INT((J+4)/5)*5
1680 T(J)=(INT(RND(1)*8)+1)+10*(INT(RND(
1)*4)+1)
1690 IF T(J)=G OR T(J)=T(L1) OR T(J)=T(L
2)OR T(J)=T(L3)OR T(J)=T(L4)THEN GOTO 16
80
1700 NEXT Q
1710 K=LEN(A$)
1720 FOR I=1 TO K
1730 Y$=MID$(A$,I,1):J=ASC(Y$)-49
1740 FOR H=8 TO 14
1750 CURSORX 7*J+3:CURSORY H
1760 TX 2,0,0:PRINT "      "
1770 NEXT H
1780 NEXT I
1790 RETURN
1800 REM Calcul et affichage resultat
```

```
1810 FOR S=0 TO 4
1820 FOR H=0 TO 4
1830 L1=(H+1)-INT((H+1)/5)*5
1840 L2=(H+2)-INT((H+2)/5)*5
1850 L3=(H+3)-INT((H+3)/5)*5
1860 L4=(H+4)-INT((H+4)/5)*5
1870 IF T(H)<T(L1) AND T(H)<T(L2) AND T(
H)<T(L3) AND T(H)<T(L4) THEN B(S)=T(H):T
(H)=100:H=5
1880 NEXTH:NEXTS
1890 FOR J=0 TO 4:T(J)=B(J)-INT(B(J)/10)
*10:NEXT J
1900 FOR S=0 TO 4
1910 FOR H=0 TO 4
1920 L1=(H+1)-INT((H+1)/5)*5
1930 L2=(H+2)-INT((H+2)/5)*5
1940 L3=(H+3)-INT((H+3)/5)*5
1950 L4=(H+4)-INT((H+4)/5)*5
1960 IF T(H)<=T(L1) AND T(H)<=T(L2) AND
T(H)<=T(L3) AND T(H)<=T(L4) THEN A(S)=T(
H):T(H)=100:H=5
1970 NEXTH:NEXTS
1980 X1=10000*A(0)+1000*A(1)+100*A(2)+10
*A(3)+A(4)
1990 TX0,0,0
2000 CURSORX 1:CURSORY 20
2010 PRINT CHR$(30);
2020 IF (INT(B(0)/10)=INT(B(4)/10)) AND(
X1=45678) THEN S1=S1+1000*M1:CURSORX 20:
CURSORY 20:PRINT"Royale":GOTO 2150
2030 IF INT(B(0)/10)=INT(B(4)/10) AND (X
1=12345 OR X1=23456 OR X1=34567) THEN SA
=1
2040 IF SA=1 THEN SA=0:S1=S1+100*M1:CURS
ORX 20:CURSORY 20:PRINT"Flush":GOTO 2150
2050 IF A(0)=A(3) OR A(1)=A(4) THEN S1=S
1+40*M1:CURSORX 20:CURSORY 20:PRINT"Carr
é":GOTO 2150
2060 IF (A(0)=A(2) AND A(3)=A(4)) OR (A(
0)=A(1) AND A(2)=A(4)) THEN S1=S1+20*M1:
CURSORX 20:CURSORY 20:PRINT"Full":GOTO 2
150
```

```
2070 IF INT(B(0)/10)=INT(B(4)/10) THEN S
1=S1+10*M1:CURSORX 28:CURSORY 20:PRINT"c
ouleur":GOTO 2150
2080 IF A(0)=A(2) OR A(1)=A(3) OR A(2)=A
(4) THEN S1=S1+3*M1:CURSORX 28:CURSORY 2
0:PRINT"Brelan":GOTO 2150
2090 IF X1=12345 OR X1=23456 OR X1=3456
7 OR X1=45678 THEN S1=S1+5*M1:CURSORX 28
:CURSORY 20:PRINT"Suite":GOTO 2150
2100 IF (A(0)=A(1) AND (A(2)=A(3) OR A(3
)=A(4))) OR (A(1)=A(2) AND A(3)=A(4)) TH
EN SB=1
2110 IF SB=1 THEN SB=0:S1=S1+M1:CURSORX
28:CURSORY 20:PRINT"2 Paires":GOTO 2150
2120 IF A(0)=A(1) OR A(1)=A(2) OR A(2)=A
(3) OR A(3)=A(4) THEN CURSORX 28:CURSORY
20:PRINT"1 Paire":GOTO 2150
2130 CURSORX 28:CURSORY 20
2140 PRINT"Perdu";
2150 FOR D=1 TO 2000:NEXTD
2160 CURSORX 1:CURSORY 1
2170 PRINT CHR$(30);
2180 RETURN
```



**Jeux
de réflexion
à deux joueurs**

5.1 CARRÉ

5.1.1 Présentation

Ce jeu est un petit jeu de réflexion auquel vous pourrez vous amuser avec vos amis. Le VG 5000 affiche un tableau composé de quinze colonnes ayant quinze lignes chacune. Le principe en est fort simple : chaque joueur choisit à son tour un des carrés encore affichés à l'écran. Ensuite, l'ordinateur efface tous les carrés situés en bas et à droite de celui-ci. Le perdant est celui qui désigne le carré en haut à gauche.

5.1.2 Organigramme et étude du programme

STRUCTURE

Ce jeu de conception fort simple vous montre combien il est facile de faire des jeux intéressants avec un programme très simple.

Lignes 40 à 60 Initialisations

Lignes	70 à	100	Affichage du titre
Lignes	110 à	180	Tracé des carrés
Lignes	190 à	220	Affichage des lettres désignant les colonnes
Lignes	230 à	290	Affichage des chiffres désignant les lignes
Ligne	300		Test de première partie
Lignes	310 à	360	Lecture des noms des joueurs
Ligne	370		Premier joueur va jouer
Lignes	380 à	430	Demande du coup joué
Lignes	440 à	480	Test de coup impossible
Lignes	490 à	500	Test de coup interdit (carré déjà noirci)
Lignes	510 à	590	Coup accepté - Le VG 5000 noircit les carrés correspondants
Lignes	600 à	610	Test de fin de partie
Lignes	620 à	650	Passage au joueur suivant
Lignes	660 à	730	Affichage "coup impossible" ou "coup interdit"
Lignes	1000 à	1190	Fin de partie

VARIABLES IMPORTANTES

K	n° du joueur
PP	Test de première partie (PP = 0)
X (15,15)	Etat des carrés (1 = noir, 0 = bleu)

CARACTÈRES SPECIAUX

Nous utilisons le caractère spécial suivant pour les carrés :

	8	4	2	1	8	4	2	1	
1									0 0
2									7 E
3									7 E
4									7 E
5									7 E
6									7 E
7									7 E
8									7 E
9									7 E
10									0 0

5.1.3 Liste des instructions

```

10 REM *****
20 REM * Carré *
30 REM *****
40 DIM X(15,15)
50 SETET 65,"007E7E7E7E7E7E7E00"
60 INIT 3,6:TX 1,7,0
70 CURSORX 12:CURSORY 1
80 PRINT "C a r r é s "
90 CURSORX 12:CURSORY 2
100 PRINT "C a r r é s "
110 REM Tracé des carrés
120 ET 4,0,0
130 FOR J=1 TO 15
140 FOR I=1 TO 15
150 CURSORX I+10:CURSORY J+4
160 PRINT CHR$(65)
170 X(I,J)=0
    
```

```
180 NEXT I: NEXT J
190 REM Lettres désignant les colonnes
200 TX 1,0,0
210 CURSORX 11: CURSORY 4
220 PRINT "ABCDEFGHIJKLMNO"
230 REM Chiffres désignant les lignes
240 FOR I=1 TO 15
250 IF I<10 THEN T=9: GOTO 270
260 T=8
270 CURSORX T: CURSORY I+4
280 PRINT STR$(I)
290 NEXT I
300 IF PP=1 GOTO 390
310 REM Lecture du nom des joueurs
320 CURSORX 10: CURSORY 22
330 INPUT "Nom du joueur 1"; X$
340 CURSORX 10: CURSORY 23
350 INPUT "Nom du joueur 2"; Y$
360 CURSORY 22: PRINT CHR$(30)
370 K=1
380 REM Demande du coup joué
390 TX 1,0,0
400 CURSORX 5: CURSORY 23
410 IF K=1 THEN PRINT X$;: GOTO 430
420 PRINT Y$;
430 INPUT " Quel est votre choix"; A$
440 REM Tests de coup impossible
450 IF LEN(A$)<>2 AND LEN(A$)<>3 THEN
    C=1: GOTO 680
460 I=ASC(MID$(A$,1,1))-64
470 J=VAL(MID$(A$,2,2))
480 IF I<1 OR I>15 OR J<1 OR J>15 THEN C
=1: GOTO 680
490 REM Test coup interdit
500 IF X(I,J)=1 THEN C=2: GOTO 680
510 REM Coup accepté
520 ET 0,0,0
530 FOR JJ=J TO 15
540 FOR II=I TO 15
550 IF X(II,JJ)=1 GOTO 590
560 REM Effacer les carrés sur l'écran
570 CURSORX II+10: CURSORY JJ+4
```

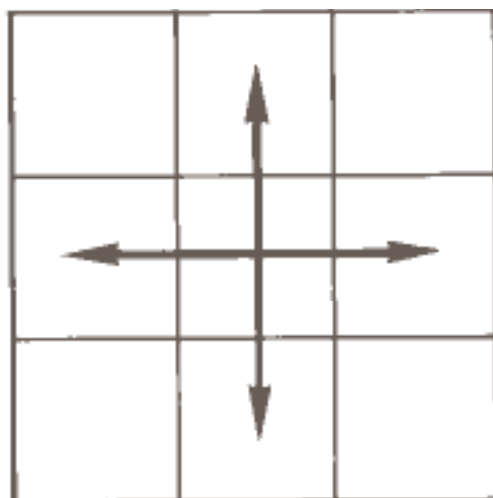
```
580 PRINT CHR$(65):X(II,JJ)=1
590 NEXT II:NEXT JJ
600 REM Test de fin de partie
610 IF X(1,1)=1 GOTO 1000
620 REM Passage au joueur suivant
630 K=K+1:IF K=3 THEN K=1
640 CURSOR Y 23:PRINT CHR$(30)
650 GOTO 390
660 REM Choix impossible ou interdit
670 TX 1,0,0:CURSOR Y 23
680 CURSOR X 10:CURSOR Y 24
690 TX 1,0,0
700 PRINT "Choix interdit";
710 FOR D=1 TO 1000:NEXT D
720 CURSOR Y 23:PRINT CHR$(30)
730 GOTO 390
1000 REM Fin de partie
1010 CURSOR X 11:CURSOR Y 5
1020 ET 0,0,0:PRINT CHR$(65)
1030 TX 1,0,0:CURSOR Y 23
1040 PRINT CHR$(30):REM effacer ligne 23
1050 CURSOR X 10:CURSOR Y 23
1060 IF K=1 THEN PRINT Y$;:GOTO 1000
1070 PRINT X$;
1080 PRINT " a gagné";
1090 FOR D=1 TO 2000:NEXT D
1100 CURSOR Y 23:PRINT CHR$(30)
1110 TX 5,0,0
1120 CURSOR X 10:CURSOR Y 23
1130 INPUT "Voulez-vous rejouer";B$
1140 CURSOR Y 23:PRINT CHR$(30)
1150 K=K+1:IF K=3 THEN K=1
1160 PP=1
1170 IF B$="0" GOTO 60
1180 PRINT CHR$(31)
1190 END
```

5.2 RÉACTIONS EN CHAÎNE

5.2.1 Présentation

Ce jeu de réflexion se joue à deux joueurs au clavier. Le but de chaque joueur est d'éliminer son adversaire sur un terrain de jeu composé de cinq colonnes de cinq cases, contenant chacune trois particules au maximum. Deux sortes de particules existent : les rouges et les vertes ; elles ne peuvent pas cohabiter dans une même case.

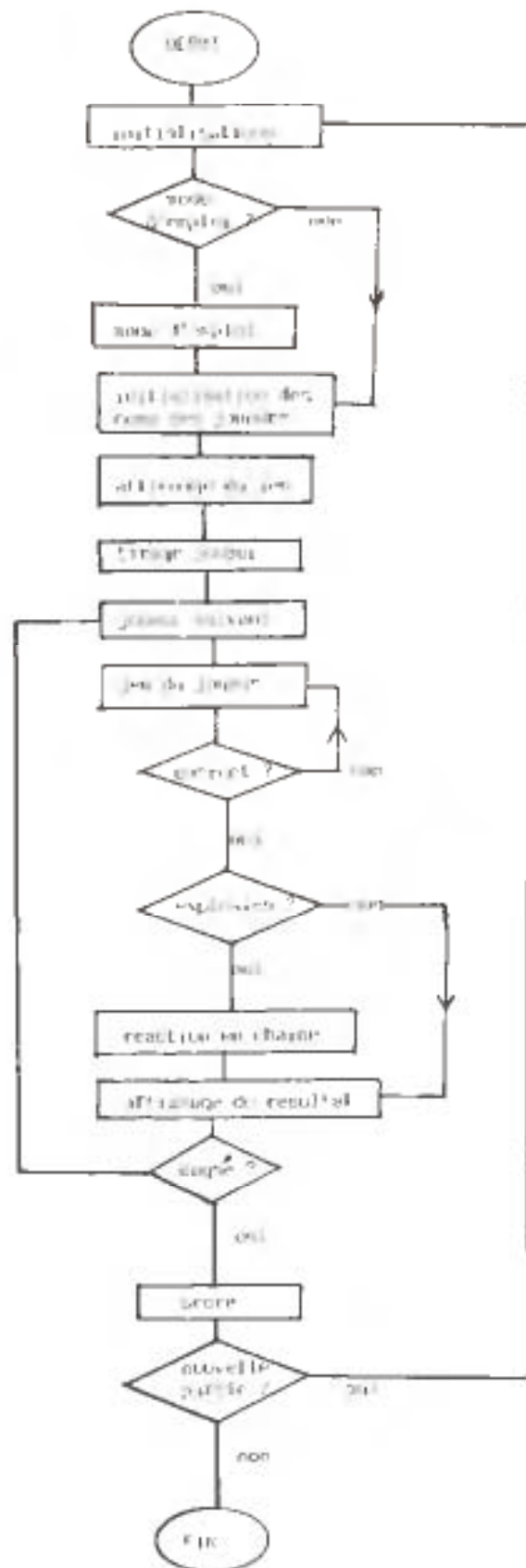
Lorsqu'une case contient trois particules, elle explose en projetant une particule dans chacune de ses cases voisines (haut, bas, gauche, droite).



Les particules de ces quatre cases deviennent alors de la même couleur que celle qui a explosé (qui est alors vide). A leur tour, ces cases voisines peuvent exploser et provoquer ainsi une réaction en chaîne.

Le gagnant est celui qui, par une suite d'explosions a éliminé l'adversaire. En pratique on désigne la case jouée par ses numéros de colonne et de ligne. L'ordinateur se charge du reste.

5.2.2 Organigramme et étude du programme



Lignes	10 à	130	Initialisations, appel du mode d'emploi et du tracé de la grille
	70		Définition de la fonction AR qui permet d'arrondir les résultats.
Lignes	500 à	770	Choix d'une case - Boucles de jeu
	500 à	590	Saisie des coordonnées de la case
	600 à	630	Ajout de la particule dans la case
	640 à	660	Test de validité du coup
	670 à	720	Coup impossible
	730 à	770	Test d'explosion de la case
	740		Appel du sous-programme de contamination des cases voisines, en cas d'explosion.
Lignes	800 à	940	Contamination des cases voisines
Lignes	1000 à	1090	Explosion d'une case
Lignes	1100 à	1150	Changement éventuel de couleurs des cases voisines
Lignes	1500 à	1630	Tracé de la grille vide
Lignes	1800 à	2030	Affichage de la grille résultat
Lignes	5000 à	5250	Mode d'emploi
Lignes	10000 à	10110	Fin de partie

PRINCIPALES VARIABLES

R (6,6)	Il décrit l'état des cases et contient le nombre de particules contenues dans chaque case, multiplié par leur couleur (1 = rouge, 10 = vert)
EX	Indique s'il y a eu ou non une explosion (1 ou 0)

5.2.3 Liste des instructions

```

10 REM *****
20 REM Réactions en chaine *
30 REM *****
40 PLAY "ABCDEABCDEEEAAAAAA"
50 GOSUB 5000: REM Mode d'emploi
60 GOSUB 1500: REM Tracé du terrain
70 DEFFN AR(X)=INT((X-INT(X))*2)+INT(X)
80 DIM R(6,6)
90 FOR I=0 TO 6
100 FOR J=0 TO 6
110 R(I,J)=0
120 NEXTJ:NEXT I
130 C$=A$:NC=0
500 REM Choix d'une case
510 TX 0,0,0:NC=NC+1
520 CURSORX 1:CURSORY 22
530 PRINT "                ":PRINT "
                    "
540 CURSORX 1:CURSORY 22:PRINT "JOUEUR "
;:PRINT C$:PLAY "02ABC"
550 PRINT "CASE :COLONNE ?";:INPUT I$
560 CURSORX 1:CURSORY 23:PRINT "LIGNE
      ";:INPUT J$
570 I=VAL(LEFT$(I$,1))
580 J=VAL(LEFT$(J$,1))
590 IF I<1 OR J<1 OR I>5 OR J>5 GOTO 520
600 REM Ajout 1 particule dans la case
610 IF C$=A$ THEN C=1:C$=B$:GOTO 630
620 C=10:C$=A$

```



```
630 R(I,J)=R(I,J)+C
640 REM Test de validité du coup joué
650 IF R(I,J)>10 AND INT(R(I,J)/10)<>R(I
.J)/10 THEN 680
660 GOTO 740
670 REM Coup impossible
680 R(I,J)=R(I,J)-C:CURSORX 1:CURSORY22
690 PRINT "Coup impossible"
700 FOR I=1 TO 20 :PLAY "AE":NEXT I
710 IF C$=A$ THEN C$=B$ : GOTO 500
720 C$=A$:GOTO 500
730 REM Test d'explosion de la case
740 IF (R(I,J)>=3 AND R(I,J)<10) OR R(I,
J)>=30 THEN GOSUB 800:GOSUB 1000
750 Z=0:T=0:GOSUB 1800
760 IF (Z=0 OR T=0)AND NC<>1 GOTO 10000
770 GOTO 500
800 REM Contamination cases voisines
810 RT=1
820 X=INT(R(I,J)/(3*C))
830 K=I-1:L=J:GOSUB 1100
840 IF K>0 THEN GOSUB 1830
850 K=I+1:GOSUB 1100
860 IF K<6 THEN GOSUB 1830
870 K=I:L=J-1:GOSUB 1100
880 IF L>0 THEN GOSUB 1830
890 L=J+1:GOSUB 1100
900 IF L<6 THEN GOSUB 1830
910 R(I,J)=R(I,J)-3*C*X
920 R(I,J)=FNAR(R(I,J))
930 K=I:L=J:GOSUB 1830
940 RETURN
1000 REM Explosion d'une case
1010 EX=0:SS=0:TT=0
1020 FOR I=1 TO 5:FOR J=1 TO 5
1030 IF R(I,J)>9 THEN SS=SS+1
1040 IF R(I,J)<10 AND R(I,J)>0 THEN TT=T
T+1
1050 IF (R(I,J)>=3 AND R(I,J)<10) OR R(I
.J)>=30 THEN EX=1:GOSUB 800
1060 NEXT J:NEXT I
1070 IF SS=0 OR TT=0 THEN GOSUB 10000
```

```
1080 IF EX=1 THEN 1000
1090 RETURN
1100 REM Cases voisines
1110 IF C=1 THEN IF R(K,L)>9 THEN R(K,L)
=R(K,L)/10
1120 IF C=10 THEN IF R(K,L)<10 THEN R(K,
L)=R(K,L)*10
1130 R(K,L)=FNAR(R(K,L))
1140 R(K,L)=R(K,L)+(X*C)
1150 RETURN
1500 REM Tracé de la grille
1510 SETET 110,"FFFFFFFFFFFFFFFFFFFFF"
1520 INIT 7,7:ET 0,0,0
1530 REM Lignes verticales
1540 FOR K=1 TO 21
1550 FOR L=2 TO 37 STEP 7
1560 CURSORX L:CURSORY K
1570 PRINT CHR$(110);
1580 NEXTL:NEXTK
1590 REM Lignes horizontales
1600 FOR L=1 TO 21 STEP 4
1610 CURSORY L:CURSORX 2:DELIM 0,0,0:CUR
SORX 38:DELIM 7,7,0
1620 NEXT L
1630 RETURN
1800 REM Affiche la grille résultat
1810 RT=0
1820 FOR K=1 TO 5:FOR L=1 TO 5
1830 R(K,L)=FNAR(R(K,L))
1840 D=1
1850 IF R(K,L)>9 THEN D=2
1860 IF R(K,L)=0 THEN D=7
1870 IF D=1 THEN Z=Z+1
1880 IF D=2 THEN T=T+1
1890 Y=R(K,L)/(D+(8*(D-1)))-1
1900 Y=FNAR(Y)
1910 IF D=7 OR Y>2 THEN Y=2
1920 FOR N=0 TO 2
1930 CURSORX 3+(7*(K-1)):CURSORY 4*L-N
1940 IF N>Y THEN D=7
1950 ET D,0,0
1960 FOR W=1 TO 6
```

```
1970 PRINT CHR$(110);
1980 NEXTW
1990 NEXTN
2000 IF RT=1 THEN RETURN
2010 NEXTL
2020 NEXT K
2030 RETURN
5000 REM MODE D'EMPLOI
5010 INIT 6,1:TX 4,3,0:PAGE
5020 CURSORX 0:CURSORY 0:PRINT "R é a o
t i o n s   e n   c h a i n e s "
5030 CURSORX 0:CURSORY 1:PRINT "R é a o
t i o n s   e n   c h a i n e s "
5040 TX 0,0,0
5050 PRINT "La partie se joue à deux jou
eurs, dis-"
5060 PRINT "posant chacun d'une couleur.
Le terrain"
5070 PRINT "est composé de 25 cases pouv
ant conte-"
5080 PRINT "nir chacune jusqu'à trois p
articules."
5090 PRINT "Les joueurs déposent à tou
r de rôle"
5100 PRINT "des particules dans une ca
se vide ou"
5110 PRINT "dans une case contenant déjà
des parti"
5120 PRINT "-cules de la même couleur.
Une case"
5130 PRINT "pleine explose automatiqueme
nt et qua-"
5140 PRINT "tre particules sont projetée
s dans les"
5150 PRINT "quatre cases adjacentes. Cel
les-ci de-"
5160 PRINT "viennent alors de la coule
ur de la"
5170 PRINT "case qui vient d'exploser."
5180 PRINT "Le vainqueur est le joueur q
ui prend"
```

```
5190 PRINT "toutes les particules de son
  adversaire"
5200 PRINT "Entrer le nom des deux joueu
rs"
5210 INPUT "Joueur 1 ":A$
5220 INPUT "Joueur 2 ":B$
5230 PRINT :PRINT "Pour commencer, appuy
er sur RET"
5240 INPUT C$
5250 RETURN
10000 REM Fin de partie
10010 INIT 3,0 :TX 0,0,0
10020 C$=A$
10030 IF TT=0 OR Z=0 THEN C$=B$
10040 PRINT "La partie est terminée !"
10050 PLAY "AAABAAABCCEEEFFG"
10060 PRINT "Le joueur ";:PRINT C$;:PRIN
T " a gagné !           BRAVO!!!"
10070 PRINT "Nouvelle partie ";
10080 INPUT NP$
10090 IF NP$="0" OR NP$="o" THEN RUN
10100 PRINT "Au revoir"
10110 END
```



Jeu d'aventure : Fantôme

6.1 PRÉSENTATION

Voici un véritable jeu d'aventure.

Vous y jouez le rôle d'un guerrier ayant pour mission de retrouver un trésor caché au plus profond d'un labyrinthe infernal. Ce trésor, qu'il vous faudra localiser, est gardé par un fantôme impitoyable, endormi depuis des siècles dans l'attente que l'on vienne le défier.

Pour explorer le labyrinthe, vous choisissez une base de départ - case sur laquelle vous serez toujours en sécurité - invisible au fantôme. A partir de là, vous pouvez vous déplacer de case en case dans toutes les directions (à l'aide des flèches). Deux cases peuvent être séparées par un mur. Vous aurez intérêt à reconnaître le terrain, de façon à éliminer les voies d'accès possibles à votre base sans réveiller le fantôme. Il est très dangereux de fuir à l'aveuglette.

Si vous passez à moins de trois cases du trésor, vous réveillez le fantôme qui dormait dessus. Celui-ci vous pourchassera alors inlassablement.

Au départ, vous avez huit points de force, c'est à dire que le fantôme avancera d'une case à chaque fois que vous en aurez parcouru huit.

Si vous rencontrez un mur, votre tour est fini et le fantôme avance dans votre direction. Si vous êtes sur votre base, il vous sent mais ne peut pas vous atteindre ; il y restera donc jusqu'à ce que vous sortiez.

Attention, il ignore les murs et peut avancer en diagonale, ce qui lui fait gagner beaucoup de temps sur vous !

Votre but : atteindre la case du trésor pour rapporter celui-ci à votre base.

Si le fantôme vous atteint, il vous attaque et vous vous retrouvez sur votre case de base en ayant perdu une partie de votre force. Après la première attaque, vous ne pourrez avancer que de six cases à chaque coup. Après la deuxième, de quatre. Après la troisième vous êtes mort !

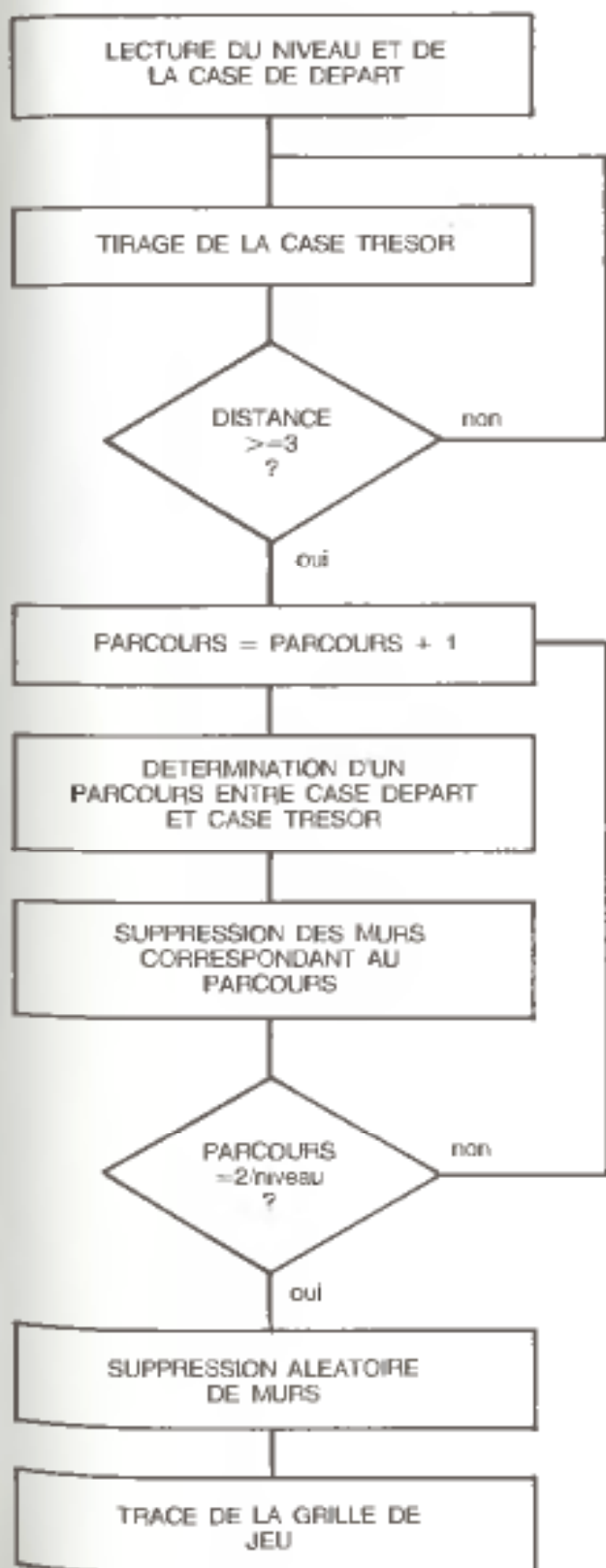
Dès que vous aurez atteint la case du trésor, vous ne pourrez plus avancer que de quatre cases par coup, alourdi par le poids du trésor que vous ramenez. Si le fantôme vous rattrape alors, c'est la mort...

Si vous désirez n'utilisez qu'une partie de votre potentiel de déplacement (par exemple pour attirer le fantôme dans une direction), appuyez sur la touche "RETURN".

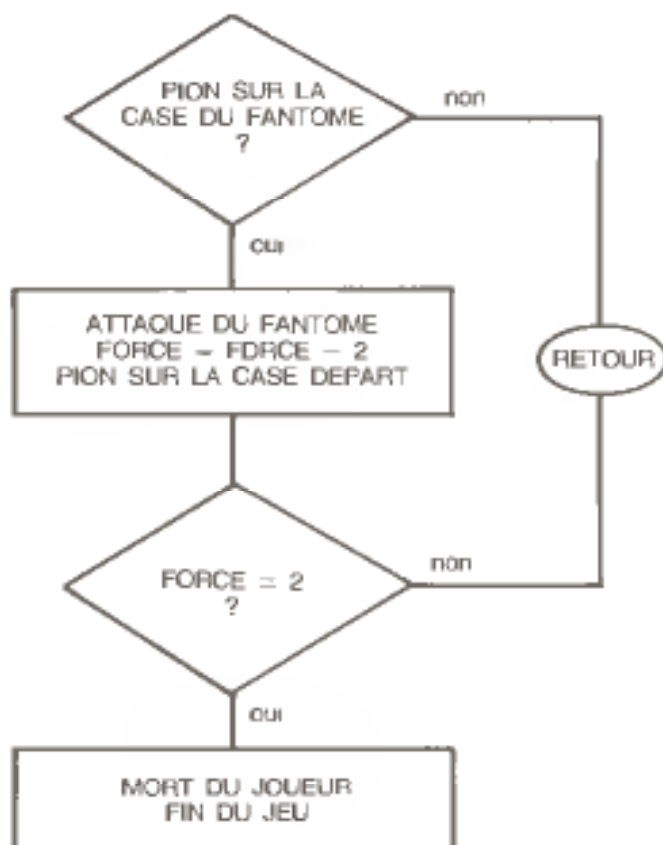
Ce n'est qu'à force de ruse et de sang-froid que vous parviendrez à votre but !

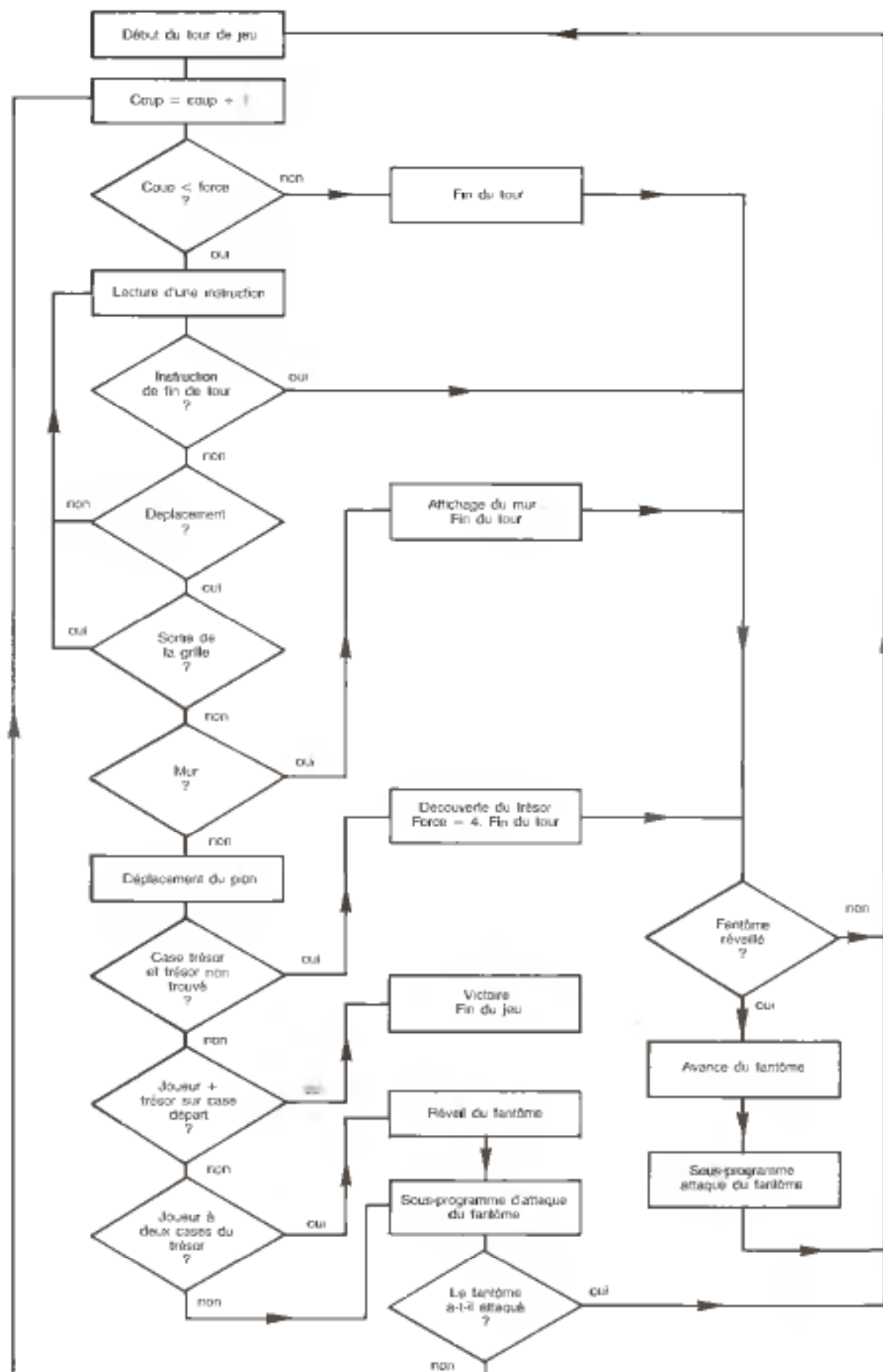
6.2 ORGANIGRAMME ET ÉTUDE DU PROGRAMME

INITIALISATIONS



SOUS-PROGRAMME D'ATTAQUE DU FANTOME





STRUCTURE DU PROGRAMME

Lignes	40 à	90	Appel des sous-programmes de mode d'emploi, d'initialisation des variables, des caractères graphiques, de tracé du labyrinthe, de positionnement du trésor...
Lignes	100 à	190	Affichages initiaux
Lignes	200 à	230	Boucle principale
Lignes	300 à	570	Déplacement du joueur (test contrôlant les diverses situations possibles)
Lignes	600 à	670	Tour de jeu (plusieurs déplacements successifs) et appel du sous-programme de test de réveil du fantôme
Lignes	700 à	790	Test de réveil du fantôme (joueur proche)
Lignes	800 à	890	Avance du fantôme (poursuite du joueur)
Lignes	1000 à	1110	Test de découverte du trésor par le joueur
Lignes	1200 à	1320	Attaque du joueur par le fantôme
Lignes	2000 à	2080	Positionnement initial du trésor dans le labyrinthe

Lignes	2500 à 2780	Règles du jeu
Lignes	3000 à 3280	Mode d'emploi (présentation, règles du jeu, choix de la case de base et du niveau)
Lignes	4000 à 4170	Construction du labyrinthe
Lignes	5000 à 5330	Tracé du labyrinthe et du cadre
Lignes	6000 à 6180	Initialisations des variables et des caractères spéciaux
Lignes	7000 à 7070	Algorithme lié à la construction du labyrinthe
Lignes	7100 à 7210	Choix d'un trajet libre entre le joueur et le trésor
Lignes	10000 à 10130	Fin de partie

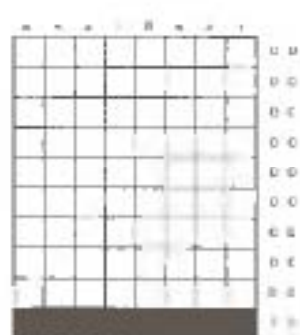
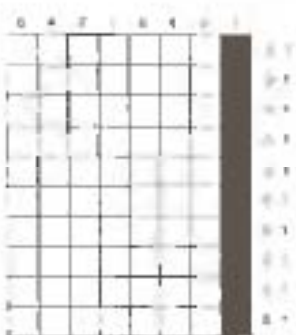
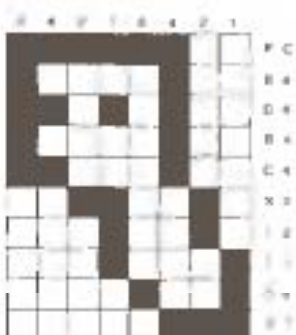
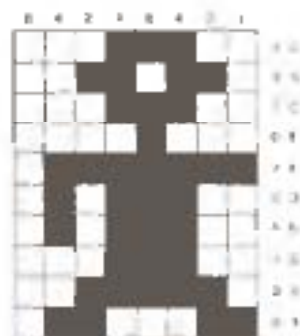
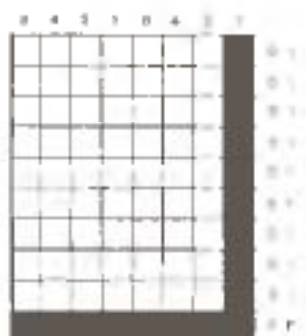
PRINCIPALES VARIABLES

FO	Force du joueur (c'est le nombre de cases qu'il peut parcourir en un tour de jeu)
CO, CC	Définissent les couleurs du labyrinthe
LC	Détermine si le labyrinthe doit être ou non apparent (vous pouvez essayer de faire LC = 0 pour visualiser le labyrinthe)
P	Tableau intervenant dans l'algorithme de tracé du labyrinthe
V, H	Tableaux définissant, pour chaque case, la présence ou non d'un mur vertical à droite (V) et d'un mur horizontal en bas (H)
VD, HD	Ont la même fonction que V et H mais concernent les murs découverts par le joueur

- FT Détermine si le fantôme est réveillé
- MM Détermine si le joueur est ou non rentré dans un mur
- XJ, YJ, XB, YB, Coordonnées du joueur, de la case de base, du fantôme et du trésor
- XF, YF, XT, YT
- JC Détermine si le joueur est sur la case de base
- PP Détermine si le joueur est mort ou victorieux

CARACTERES SPECIAUX

Nous utilisons les six caractères spéciaux suivants (joueur, fantôme, trésor, labyrinthe) :



6.3 LISTE DES INSTRUCTIONS

```

10 REM *****
20 REM * Fantôme *
30 REM *****
40 GOSUB 3000: REM Mode d'emploi
50 GOSUB 6000: REM Initialisations
60 GOSUB 2000: REM Position du trésor
70 GOSUB 4000: REM Construction labyr.
80 GOSUB 7100: REM Choix du trajet libre
90 GOSUB 5000: REM Tracé du labyrinthe
100 LC=0:ET 1,0,0:CURSORX 2*XJ+5
110 CURSORY 2*YJ+2:PRINT CHR$(125);
120 FOR I=1 TO 11
130 V(10,I)=1:V(0,I)=1
140 H(I,10)=1:H(I,0)=1
150 NEXT I
160 CURSORX 2*XB+4:CURSORY 2*YB+1
170 ET 0,0,0:PRINT CHR$(124);
180 CURSORX 1:CURSORY 0:TX 0,0,0
190 PRINT "Interruption du tour de jeu:R
ET";
200 GOSUB 600:REM Tour de jeu
210 IF FT=0 GOTO 200
220 GOSUB 800:REM Avance du fantôme
230 GOTO 200
300 REM Déplacement du joueur
310 AA=KEY(0):MM=0:IF AA=0 GOTO 310
320 IF AA=13 THEN RETURN
330 IF AA=7 THEN AJ=1-V(XJ,YJ):BJ=0:ZE=0
:ZZ=0
340 IF AA=8 THEN AJ=V(XJ-1,YJ)-1:BJ=0:ZE
=-1:ZZ=0
350 IF AA=9 THEN BJ=H(XJ,YJ-1)-1:AJ=0:ZE
=0:ZZ=-1
360 IF AA=10 THEN BJ=1-H(XJ,YJ):AJ=0:ZE=
0:ZZ=0
380 R=110:I=XJ+ZE:J=YJ+ZZ:CC=1
390 IF AJ<>0 OR BJ<>0 THEN 440
400 SOUND 20,4,15:MM=1:IF AA<9 THEN VD(I
,J)=1

```

```

410 IF AA>8 THEN HD(I,J)=1
420 HH(I,J)=HD(I,J):VV(I,J)=VD(I,J)
430 GOSUB 5150:GOTO 550
440 IF VD(XJ,YJ)=1 THEN R=111
450 IF HD(XJ,YJ)=1 THEN R=112
460 IF HD(XJ,YJ)=1 AND VD(XJ,YJ)=1 THEN
R=113
470 CURSORX 2*XJ+5:CURSORY 2*YJ+2
480 ET CC,0,0:PRINT CHR$(R);
490 IF BH=1 THEN RETURN
500 JC=0
510 XJ=XJ+AJ:YJ=YJ+BJ:IF XJ=XB AND YJ=YB
THEN JC=1
520 IF JC=1 AND XT=0 THEN GOTO 10000
530 IF XJ=XF AND YJ=YF AND JC=0 THEN GOS
UB 1200
540 IF XJ=XT AND YJ=YT THEN GOSUB 1000
550 CURSORX 2*XJ+5:CURSORY 2*YJ+2
560 ET 1,0,0:PRINT CHR$(125);
570 RETURN
600 REM Tour de jeu (F0 coups)
610 FOR L=1 TO F0
620 GOSUB 300:IF FT=0 THEN GOSUB 700
630 IF A=13 THEN RETURN
640 IF MM=1 THEN 660
650 NEXT L
660 IF FT=1 THEN PLAY "EERREERREERREE"
670 RETURN
700 REM Test de réveil du fantôme
710 IF ABS(XJ-XF)>3 OR ABS(YJ-YF)>3 THEN
RETURN
720 FT=1:PLAY "AAAABBBBBBBBCCCCCCCCCCCC"
730 CURSORX 1:CURSORY 1:TX 0,0,0
740 PRINT "Le fantôme se réveille!";
750 FOR BB=1 TO 50:SOUND 12,2,12
760 NEXT BB
770 CURSORX 1:CURSORY 1
780 PRINT " ";
790 RETURN
800 REM Avance du fantôme
810 IF NIV>1 GOTO 840
820 ET 7,0,0:CURSORX 2*XF+5

```

```
830 CURSORY 2*YF+2:PRINT CHR$(126);
840 IF XF>XJ THEN XF=XF-1
850 IF XF<XJ THEN XF=XF+1
860 IF YF<YJ THEN YF=YF+1
870 IF YF>YJ THEN YF=YF-1
880 CURSORX 1:CURSORY 1:TX 0,0,0
890 PRINT "Le fantôme avance!";
900 FOR BB=1 TO 25: SOUND 100,2,12
910 NEXT BB
920 CURSORX 1:CURSORY 1
930 PRINT " ";
940 IF XF=XJ AND YF=YJ AND JC=0 THEN GOS
UB 1200
950 IF NIV>1 GOTO 980
960 ET 4,0,0:CURSORX 2*XF+5
970 CURSORY 2*YF+2:PRINT CHR$(126);
980 RETURN
1000 REM Test découverte du trésor
1010 F0=4
1020 XT=0:YT=0
1030 PLAY "ARARARARARRRAE EEECCDDDDDDDD"
1040 CURSORX 2*XJ+5:CURSORY 2*YJ+2
1050 ET 4,0,0:PRINT CHR$(127);
1060 FOR BB=1 TO 50: SOUND 200,2,3
1070 NEXT BB
1080 CURSORX 2*XJ+5:CURSORY 2*YJ+2
1090 PRINT CHR$(125);
1100 MM=1
1110 RETURN
1200 REM Attaque du fantôme
1210 IF F0=4 THEN PP=1:GOTO 10000
1220 PLAY "AABBBCCCCDDDD"
1230 CURSORX 2*XF+5:CURSORY 2*YF+2
1240 ET 4,0,0:PRINT CHR$(126);
1250 FOR BB=1 TO 40: SOUND 35,2,24
1260 NEXT BB
1270 F0=F0-2
1280 GH=1:CC=1:GOSUB 440:GH=0
1290 XJ=XB:YJ=YB:JC=1
1300 CURSORX 2*XJ+5:CURSORY 2*YJ+2
1310 PRINT CHR$(125);
1320 RETURN
```

```
2000 REM Tirage position du trésor
2010 X=INT(RND(1)*20)-10:XT=XB+X
2020 IF ABS(X)<4 GOTO 2010
2030 IF XT<1 OR XT>10 GOTO 2010
2040 Y=INT(RND(1)*10)-5:YT=YB+Y
2050 IF ABS(Y)<4 GOTO 2040
2060 IF YT<1 OR YT>10 GOTO 2040
2070 XF=XT:YF=YT
2080 RETURN
2500 REM Règles du jeu
2510 CURSORX 1:CURSORY 5
2520 PRINT "Vous êtes dans un labyrinthe
dans"
2530 PRINT "lequel vous devez découvrir
un trésor"
2540 PRINT "gardé par un fantôme. Vous a
vancez "
2550 PRINT "grâce aux quatre flèches de
direction."
2560 PRINT
2570 PRINT "Si vous passez à moins de 4
cases du"
2580 PRINT "trésor, vous réveillez le fa
ntôme"
2590 PRINT "endormi dessus. Il vous pour
suivra"
2600 PRINT "alors impitoyablement."
2610 PRINT
2620 PRINT "Vous avez le droit d'avancer
de huit"
2630 PRINT "cases; puis le fantôme se dé
place."
2640 PRINT "S'il vous atteint, vous rega
gnez votre"
2650 PRINT "case de base, et vous ne pou
rrez alors"
2660 PRINT "plus avancer que de 6 cases
à la fois":PRINT
2670 CURSORX 1:CURSORY 21
2680 PRINT "RET pour la suite de la régl
e"
2690 INPUT R$:INIT 3,0:TX 0,0,0
```



```
2700 PRINT "Attention, le fantôme peut t
raverser"
2710 PRINT "les murs. Au bout de 3 attaq
ues, vous"
2720 PRINT "êtes mort.":PRINT
2730 PRINT "Votre but est de ramener le
trésor à"
2740 PRINT "votre base. Si vous rentrez
dans un"
2750 PRINT "mur, le fantôme avance."
2760 PRINT "Sur votre case de base, vous
êtes"
2770 PRINT "invisible pour le fantôme"
2780 RETURN
3000 REM Mode d'emploi
3010 REM Présentation
3020 INIT 3,0:TX 0,7,0
3030 CURSORX 0:CURSORY 0
3040 PRINT "F a n t ô m e "
3050 CURSORX 8:CURSORY 1
3060 PRINT "F a n t ô m e "
3070 TX 0,0,0:CURSORY 5
3080 PRINT "Voulez-vous les règles du";
3090 INPUT " jeu";R$
3100 IF R$="0" OR R$="o" THEN GOSUB 2500
3110 REM Coordonnées de la case de base
3120 PRINT :PRINT
3130 PRINT "Entrez les coordonnées de vo
tre base"
3140 INPUT "1<X<10";XB
3150 IF XB<1 OR XB>10 THEN 3140
3160 INPUT "1<Y<10";YB
3170 IF YB<1 OR YB>10 THEN 3160
3180 XJ=XB:YJ=YB
3190 REM Choix du niveau
3200 PRINT :PRINT
3210 PRINT "Quel niveau voulez-vous ?"
3220 PRINT
3230 PRINT"1: murs et fantôme visibles"
3240 PRINT"2: murs visibles"
3250 PRINT "3: rien de visible"
3260 PRINT :INPUT "votre choix";NIV
```

```

3270 IF NIV<1 OR NIV>3 GOTO 3260
3280 RETURN
4000 REM Délimitation des murs
4010 FOR I=1 TO 10
4020 FOR J=1 TO 10
4030 IF P(I,J)=0 THEN KK=KK+1:GOSUB 4060
4040 NEXT J
4050 NEXT I
4060 REM Tirage des murs
4070 A=I:B=J:P(I,J)=1
4080 X=INT(RND(1)*4):IF X=4 THEN X=3
4090 E=0:Z=0
4100 IF X=0 AND A<10 THEN V(A,B)=0:E=1:A
=A+1
4110 IF X=1 AND A>1 THEN A=A-1:V(A,B)=0:
E=-1
4120 IF X=2 AND B>1 THEN B=B-1:H(A,B)=0:
Z=-1
4130 IF X=3 AND B<10 THEN H(A,B)=0:B=B+1
:Z=1
4140 IF E=0 AND Z=0 THEN 4080
4150 IF P(A,B)=KK THEN GOSUB 7010:IF TT=
0 THEN RETURN
4160 P(A,B)=KK
4170 GOTO 4080
5000 REM Tracé du cadre
5010 INIT 7,1:ET 0,0,0
5020 FOR I=5 TO 26
5030 CURSORX I:CURSORY 2:PRINT CHR$(124)
;
5040 CURSORX I:CURSORY 23:PRINT CHR$(124
);
5050 NEXT I
5060 FOR I=3 TO 22
5070 CURSORY I
5080 CURSORX 5:PRINT CHR$(124);
5090 CURSORX 26:PRINT CHR$(124);
5100 NEXT I
5110 REM Tracé du labyrinthe
5120 FOR I=1 TO 10
5130 FOR J=1 TO 10
5140 HH(I,J)=H(I,J):VV(I,J)=V(I,J)

```

```

5150 A=111:B=112:C=110:C1=C0:C2=C0:C3=C0
5160 IF LC=1 THEN GOTO 5210
5170 IF I<1 OR J<1 THEN RETURN
5180 IF VV(I,J)=1 THEN C1=CC:C=111:C3=CC
5190 IF HH(I,J)=1 THEN C2=CC:C=112:C3=CC
5200 IF VV(I,J)=1 AND HH(I,J)=1 THEN C3=
CC:C=113
5210 ET C1,0,0
5220 CURSORX 2*I+5:CURSORY 2*J+1
5230 PRINT CHR$(A);
5240 CURSORX 2*I+4:PRINT CHR$(110);
5250 IF I=XB AND J=YB THEN CURSORX 2*I+4
:ET 1,0,0:PRINT CHR$(124);
5260 ET C2,0,0
5270 CURSORX 2*I+4:CURSORY 2*J+2:PRINT C
HR$(B);
5280 ET C3,0,0
5290 CURSORX 2*I+5:PRINT CHR$(C);
5300 IF MM=1 THEN RETURN
5310 NEXT J
5320 NEXT I
5330 RETURN
6000 REM Initialisations des variables
6010 C0=6:CC=0:FO=8:LC=0:IF NIV=3 THEN L
C=1
6020 DIM P(12,12),V(11,11),H(11,11),VD(1
1,11),HD(11,11)
6030 FOR I=1 TO 10:FOR J=1 TO 10
6040 P(I,J)=0:V(I,J)=1:H(I,J)=1:VD(I,J)=
0:HD(I,J)=0
6050 NEXT J:NEXT I
6060 FOR I=0 TO 11
6070 P(0,I)=1:P(I,0)=1:P(11,I)=1:P(I,11)
=1
6080 NEXT I
6090 REM Caractères spéciaux
6100 SETET 110,"00000000000000000000"
6110 SETET 111,"01010101010101010101"
6120 SETET 112,"00000000000000000000FF"
6130 SETET 113,"01010101010101010101FF"
6140 SETET 124,"FFFFFFFFFFFFFFFFFFFFFF"
6150 SETET 125,"1C361C087F5D5D1C3663"

```

```

6160 SETET 126,"FC84D484C43212110907"
6170 SETET 127,"FEEEC66C38101010387C"
6180 RETURN
7000 REM Algorithme du labyrinthe
7010 A=A-E:B=B-Z
7020 IF P(A+1,B)=0 AND A<11 THEN A=A+1:T
T=1:V(A-1,B)=0:RETURN
7030 IF P(A-1,B)=0 AND A>0 THEN A=A-1:TT
=1:V(A,B)=0:RETURN
7040 IF P(A,B-1)=0 AND B>0 THEN B=B-1:TT
=1:H(A,B)=0:RETURN
7050 IF P(A,B+1)=0 AND B<11 THEN B=B+1:T
T=1:H(A,B-1)=0:RETURN
7060 TT=0
7070 RETURN
7100 REM CHOIX D'UN TRAJET
7110 I=XB:J=YB
7120 KL=I:KM=XT-1:BB=1:B1=-1:NN=1:B2=-1:
IF XB>XT THEN BB=-1:B1=0:KL=XT:KM=I-1
7130 KZ=YT:KW=J-1
7140 IF YB>YT THEN NN=-1:B2=0:KZ=J:KW=YT
-1
7150 IF XT=XB OR YT=YB THEN 7190
7160 X=INT(RND(1)*2)
7170 IF X=0 THEN I=I+BB:V(I+B1,J)=0:GOTO
7190
7180 J=J+NN:H(I,J+B2)=0
7190 IF I=XT THEN FOR K=KZ TO KW:H(XT,K)
=0:NEXT K:RETURN
7200 IF J=YT THEN FOR K=KL TO KM:V(K,YT)
=0:NEXT K:RETURN
7210 GOTO 7160
10000 REM Fin de partie
10010 INIT 3,4:TX 0,0,0
10020 IF PP=0 GOTO 10060
10030 PRINT "VOUS ETES MORT ! ! !"
10040 PLAY "AAAAARRRRBBBBBBBRRRRCCCCC"
10050 GOTO 10090
10060 PRINT "BRAVO , VOUS AVEZ GAGNE !"
10070 PLAY "AAARRRBBRRCCCCRRRDDAAABCR"
10080 PLAY"CRCRCRCRCRCRCRAAAAAAAAAAAAAA"
10090 PRINT "Nouvelle partie ":

```

```
10100 INPUT R$
10110 IF R$="0" OR R$="c" THEN RUN
10120 PRINT :PRINT "Au revoir !"
10130 END
```

LA BIBLIOTHÈQUE EDIMICRO

• *Collection « Ordinateurs personnels »*

ATMOS/ORIC 1

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

COMMODORE 64

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

ELECTRON

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

MOS

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

SPECTRUM

Bridge, Carnell	Aventures sur Spectrum
Hurley	Jeux Graphiques sur Spectrum

TO7

— ouvrages	
Bieber, Perbost, Renucci	Guide du TO7
Perbost, Renucci	Jeux sur TO7
— logiciels sur cassette	
Perbost, Renucci	4 jeux avec manette pour TO7
Perbost, Renucci	6 jeux d'action et de réflexion pour TO7

VG 5000 PHILIPS

Amster, Bardon	Guide du VG 5000 Philips
Amster, Villemaud	Jeux graphiques sur VG 5000 Philips

PHILIPS C7420 VIDEOPAC +

Bardon, de Merly	Jeux sur Philips C7420
	Vidéopac +

• *Collection « Ordinateurs professionnels »*

APPLE

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

IBM PC

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

• *Collection « langages »*

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

• *« Intérêt général »*

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

Nombreux autres titres à paraître. Catalogue sur simple demande.

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

EXTRAITS DE PRESSE

GUIDE DE L'APPLE

« Un des meilleurs sur la place. »

Le Figaro.

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

Apple France.

« Toutes les étapes deviennent plus claires. »

Goiden.

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

Minis et Micros.

MULTIPLAN SUR APPLE

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

Microsoft France.

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

Science et Vie Micro.

GUIDE DU TO 7

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

Micro 7.

« Un manuel de référence absolue. »

Le Figaro.

JEUX SUR TO 7

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

Décision informatique.

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

Micro 7.

GUIDE DE L'ORIC

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

Microsystèmes.

« Un best-sellar. »

Livres Hebdo.

PREMIERS PAS EN PROGRAMMATION SUR ORIC

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

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

JEUX SUR VG 5000 PHILIPS

B. Amsler, O. Villemaud

JEUX D'ACTION : Squash, Bombardier, Chenille, Survie, Stock-car, Envahisseurs, Pilote de course, Collimateur, Course-poursuite, Réactions en chaîne, ...

JEUX DE RÉFLEXION : Karo, Labyrinthe, Radar, Tic-tac-toe, Taquin, Boîte Noire, Mémocarte, Poker, Bataille Navale, ...

JEU D'AVENTURE : Fantôme.

Tous les programmes sont prêts à l'emploi : vous pourrez ainsi vous constituer, à très peu de frais, une véritable bibliothèque de jeux passionnants et diversifiés.

Si vous voulez aller plus loin, étudiez et modifiez vous-même les programmes que nous proposons.

Chaque jeu est complètement décrit : règle du jeu, organigramme et étude du programme, définition des caractères graphiques spéciaux, liste des instructions.

Le premier chapitre vous donne tous les conseils utiles pour créer vos propres jeux.

Ce livre a été réalisé en étroite collaboration avec Philips.

ISBN : 2-904457-26-7

88 F.



Edimicro

121-127 Avenue d'Italie 75013 Paris