

BOITE



A

POCHE

OUTILS
POUR

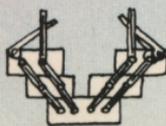
ORIC

ORIC I et ATMOS



TOME 2

**15 PROGRAMMES
EN BASIC**



michel martin

Boîte à outils
pour
ORIC
(ORIC 1 et ATMOS)

Boîte à outils pour ORIC, ORIC 1 ET ATMOS.

I.S.B.N. 2.87109.011.4

© s.a. SAPECA, Uccle (Belgique) 1984

Toute reproduction, non strictement réservée à l'usage du copiste, d'un extrait quelconque de ce livre par quelque procédé que ce soit, est interdite sans autorisation écrite de l'éditeur.

Les collections **MégaO** (Poche) sont éditées par SAPECA, 5 avenue de la Ferme Rose, 1180 Bruxelles (Roze Hoevelaan - 1180 Brussel). Le label **MégaO**, les titres des collections et la présentation des volumes sont déposés conformément à la loi. Distributeurs en **France** : PSI DIFFUSION, BP 86, 77402 Lagny-sur-Marne Cedex. Pour le **Canada** et les **USA** : S.C.E. Inc., 65 Avenue Hillside, Westmount - Québec H3 Z1 W1. Pour le **Benelux** : PSI Benelux, 5 Avenue de la Ferme Rose, 1180 Bruxelles.

Boîte à outils

pour

ORIC

(ORIC 1 et ATMOS)

15 programmes

par
Michel Martin



POCHE

1984

Avertissement

Ce livre constitue la suite de "Boîte à outils pour OM Tome 1". Les programmes présentés ici sont d'un niveau plus complexe. Ils gardent la forme structurée et hiérarchisée définie dans le tome 1.

Rappelons qu'un programme hiérarchisé possède des niveaux de complexité différents :

- . Un module de commande (Niveau hiérarchique élevé)
- . Souvent 3 modules principaux (Début, Traitement, Fin)
- . Des procédures élémentaires, souvent sollicitées (Niveau hiérarchique bas).

Le but de la hiérarchisation est de rendre un programme plus compréhensible et facilement adaptable à des modifications éventuelles.

Rappelons également qu'un programme structuré possède une base (ou structure) définie et logique, et que le but de la structuration est d'augmenter la lisibilité du programme et par la suite, sa maintenance.

Les programmes qui suivent possèdent souvent 3 modules principaux : Début, Traitement et Fin (structure A). Les modules Début et Fin ont la même fonction dans chaque programme. Pour cette raison, ils ne seront explicités que dans le premier programme.

Sommaire

	Page
Compatibilité ORIC 1 - ORIC ATMOS	7
JEU - D'adresse ou de rapidité	
- Atterrissage forcé	12
- Mur de briques	19
- Slalom ski	26
JEU - De réflexion	
- Awari	36
- Black box	46
- Grenouilles et crapauds	58
- Conjugaison des verbes	64
GRAPHIQUES ET MATHÉMATIQUES	
- Tracé de courbes $Y = F(x)$	72
- Tracé de courbes $Y(t), X(t)$	77
- Tracé de courbes $Y(t), X(t)$ imbriquées	82
- Snoopy	84
- Dessin Animé	89
- Le calendrier perpétuel	93
- Rubick's cube	97
- Classement par ordre croissant ou décroissant chaînes et nombres	106
UTILITAIRE	
- Désassembleur 6502	114

Compatibilité ORIC-1 ORIC ATMOS

Les 17 programmes proposés dans ce livre ont été développés sur ORIC 1. Ils sont compatibles ORIC ATMOS moyennant les modifications suivantes :

ATTERRISSAGE FORCE

Menu*

```
2010 :   GOSUB 1500:PRINT"Carburant : ";CA;
"Bombes vert.:";VG
2020 :   GOSUB 1500:PRINT"           ";
"Laser horiz.:";HL
2770 :   PRINT CHR$(11);CHR$(11);"   Carburant : ";
CA;"Bombes vert.:";VG
2870 :   PRINT"                               Laser";
"horiz.:";HL
```

MUR DE BRIQUES

Menu*

```
2270 :           FOR J=0 TO 35
2480 :   PRINT "Pressez une touche pour commencer"
3131 :   IF XB+VX<4 THEN VX=1:PLAY3,0,1,90
4040 :   PRINT CHR$(27);"L";CHR$(27);"Pressez une
touche pour commencer".

3370 :   PRINT"Pour commencer,frappez une
touche
3640 :   IF X1=2 OR X1=38 OR Y1=0 OR Y1=25 THEN
FIN=1:S=1:GOTO 3700
```

SLALOM SKI

```

..... 3370 :      PRINT"Pour commencer, frappez une
          touche"
          3640 :      IF X1=2 OR X1=38 OR Y1=0 OR Y1=25 THEN
          FIN=1:S=1:GOTO 3700

```

AWARI

Menu*

```

5130 :      IF E<>0 THEN POKE 48768+I*4,E+48
5140 :      POKE 48529+I*4,T(14-I)-10*E+48
5200 :      IF E<>0 THEN POKE 48768+I*4,E+48
5210 :      POKE 48769+I*4,T(I+2)-E*10+48
5250 :      IF E<>0 THEN POKE 48644,E+48
5260 :      POKE 48645,T(1)-E*10+48
5290 :      POKE 48672,E+48
5300 :      POKE 48673,T(8)-E*10+48

```

BLACK BOX

```

2276 :      GOSUB 1900:PRINT"T=Tir , P=Placement ,
          E=Effacement"
2550 :      PRINT CHR$(27);"H";CHR$(27);"AT=Tir";
CHR$(27);"H";CHR$(27);"F"
4120 :      PRINT CHR$(27);"H";CHR$(27);"FT=Tir ";
CHR$(27);"H";CHR$(27);"A";
4270 :      A$="F":GOSUB 1900:PRINT"T=Tir ,
          P=Placement , E=Effacement"
4430 :      PRINT" E=Effacement"
4570 :      A$="F":GOSUB 1900:PRINT"T=Tir ,
          P=Placement , E=Effacement"

```

GRENOUILLES ET CRAPAUDS

Menu*

```

2950 :      IF NG=NC AND JEU$(NC+1)=" " THEN NOFIN=0
ELSE NOFIN=1

```

CONJUGAISON DES VERBES

Menu*

```

2260 :      PRINT"CONJUGAISON DES VERBES : 1er
          GROUPE"

```

TRACE DE COURBES

$Y = F(x)$

Rien à modifier

TRACE DE COURBES

$Y(t), X(t)$

Menu*

TRACE DE COURBES

$Y(t), X(t)$ imbriquées

Menu*

SNOOPY

Rien à modifier

DESSIN ANIME

Rien à modifier

LE CALENDRIER PERPETUEL

Menu*

RUBICK'S CUBE

Rien à modifier

CLASSEMENT DE CHAINES

1080 : PRINT CHR\$(27); "A CLASSEMENT CROISSANT
ET DECROISSANT"

CLASSEMENT DE NOMBRES

1080 : PRINT CHR\$(27); "A CLASSEMENT CROISSANT
ET DECROISSANT"

DESASSEMBLEUR 6502

Menu*

Menu* - Certaines lignes du menu de présentation devront comporter un nombre de "blancs" moins important, ou des mots plus courts...

PREMIÈRE PARTIE

**JEUX D'ADRESSE
OU DE RAPIDITÉ**

Atterrissage forcé

PRESENTATION

Suite à une fuite de carburant, un avion doit se poser dans la forêt vierge s'il ne veut pas s'écraser.

Il doit se ménager un terrain d'atterrissage grâce au matériel laser qu'il possède à bord.

Ce matériel comprend :

- . Un laser de portée limitée mais d'autonomie infinie,
- . Un super laser vertical permettant 5 tirs,
- . Un laser horizontal permettant 4 tirs.

Pourra-t-il se poser à temps ?

ORGANIGRAMME DU JEU

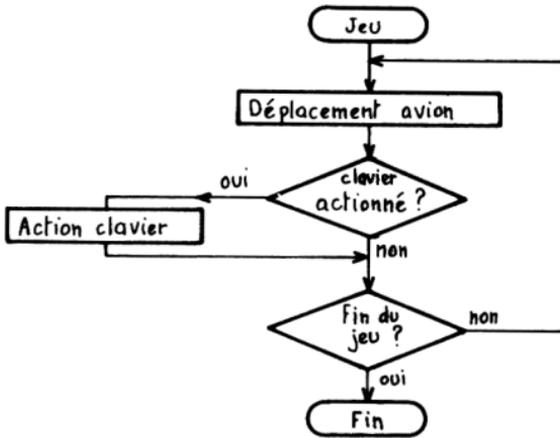
Le programme a la structure A :

Le module "Début" se décompose en :

- Affichage du menu (Présentation du jeu)
- Initialisation du jeu

Le module "Fin" commente le jeu et relance éventuellement une autre partie.

Le jeu a la structure suivante :



LE PROGRAMME

Il se décompose en 4 pavés :

Affichage du menu	Lignes 1000 à 1550	
Initialisation du jeu	Lignes 1600 à 2050	
	Module de commande	Lignes 1600 à 1670
	Définition avion	Lignes 1700 à 1780
	Définition terrain	Lignes 1800 à 1895
	Données de parcours	Lignes 1900 à 2050
Jeu	Lignes 2100 à 2890	
	Module de commande	Lignes 2100 à 2170
	Déplacement avion	Lignes 2200 à 2290
	Commande joueur	Lignes 2300 à 2390
	Avion bas	Lignes 2400 à 2480
	Avion haut	Lignes 2500 à 2580
	Tir	Lignes 2600 à 2690
	Bombe verticale	Lignes 2700 à 2790
	Laser horizontal	Lignes 2800 à 2885
Fin du jeu	Lignes 3000 à 3080	
	Commentaires sur la partie	
	Activation éventuelle d'une nouvelle partie.	

LES VARIABLES UTILISEES

A	Définition de l'avion et du terrain
A\$	Code couleur de ligne
B\$	Acquisition clavier
C\$	Une autre partie ?

CA	Carburant restant
FIN	Indicateur de fin de partie
HL	Laser horizontal
I	Index de boucles
VG	Bombes verticales
XP, YP	Positions de l'avion sur l'écran

```

10 : REM Atterrissage force
20 :
30 : GOSUB 1000 'Affichage du menu
40 : GOSUB 1600 'Initialisation du jeu
50 : GOSUB 2100 'Jeu
60 : GOSUB 3000 'Partie terminee
70 :
80 : END
90 : REM #####
1000 : REM Affichage du menu
1010 :
1020 : PAPER 0 : INK 5 : PRINT CHR$(17) 'Effacement du curseur
1030 : REM Titre
1040 :
1050 : PRINT CHR$(12)
1060 : PRINT CHR$(4);CHR$(27);"N";
1070 : PRINT CHR$(27);"A ATERRISSAGE FORCE";CHR$(4)
1080 :
1090 : REM Objet
1110 : PRINT:PRINT:PRINT
1120 : A$="D" 'Code couleur = Bleu
1130 : PRINT"OBJET :";
1140 : GOSUB 1500:PRINT"Jeu de rapidite."
1150 :
1160 : REM Job
1170 :
1180 : PRINT:PRINT
1185 : A$="F" 'Code couleur = Cyan
1190 : PRINT"JOB :";
1200 : GOSUB 1500:PRINT"Vous vous trouvez dans un"
1210 : GOSUB 1500:PRINT" avion et vous devez atterrir"
1220 : GOSUB 1500:PRINT" d'urgence faute de carburant."
1230 : GOSUB 1500:PRINT" Malheureusement,la foret vierge"
;

```

```

1780 : REM *****
1800 : REM Tirage aleatoire du terrain d'atterissage
1810 :
1815 : CLS : INK 6 'Couleur des obstacles
1820 : FOR I=1 TO 30
1830 :   A=INT(RND(1)*9) 'Hauteur du terrain
1840 :   FOR J=1 TO A
1850 :     PLOT I+3,23-J,CHR$(126)
1860 :     NEXT J
1870 :   NEXT I
1880 :
1890 : RETURN
1895 : REM *****
1900 : REM Donnees de parcours
1910 :
1920 : CA=1000 'Carburant restant
1930 : VG=4 'Bombes verticales
1940 : HL=5 'Laser horizontal
1950 :
1960 : FOR I=1 TO 24 'Positionne le curseur ligne 24
1970 :   PRINT
1980 : NEXT I
1990 :
2000 : A$="C" 'Code couleur = Jaune
2010 : GOSUB 1500:PRINT" Carburant : ";CA;" Bombes vert.: ";VG
2020 : GOSUB 1500:PRINT" Laser horiz.: ";HL
2030 :
2040 : RETURN
2050 : REM *****
2100 : REM Deroulement du jeu
2110 :
2120 : GOSUB 2200 'Deplacement de l'avion
2130 : A$=KEY$
2140 : IF A$(">") THEN GOSUB 2300 'Commande du joueur
2150 : IF FIN(">1") THEN GOTO 2100 'Boucle du jeu
2160 :
2170 : RETURN
2180 : REM *****
2200 : REM Deplacement de l'avion
2210 :
2220 : PLOT XP,YP," ":PLOT XP+1,YP," " 'Effacement de l'avion
2230 : IF XP=33 THEN XP=2 ELSE XP=XP+1 'Gestion position de l'avi
on
2240 : IF SCRIN(XP+1,YP)(">32") THEN FIN=1 'Collision

```

```

1240 : GOSUB 1500:PRINT"      s'etend au desous de vous.Pour"
1250 : GOSUB 1500:PRINT"      pouvoir atterrir,il faudra la"
1260 : GOSUB 1500:PRINT"      nettoyer.Pour cela,vous avez a"
1270 : GOSUB 1500:PRINT"      votre disposition 50 bombes ,"
1280 : GOSUB 1500:PRINT"      4 lasers verticaux et 5 "
1290 : GOSUB 1500:PRINT"      horizontaux actionnables par"
1300 : GOSUB 1500:PRINT"      les touches SPACE,C et X . Les"
1310 : GOSUB 1500:PRINT"      touches-fleches vers le bas et"
1320 : GOSUB 1500:PRINT"      vers le haut dirigent l'avion."
1330 :
1340 : REM Debut
1350 :
1360 : PRINT:PRINT
1370 : A$="C" 'Code couleur = Jaune
1380 : PRINT"DEBUT :";
1390 : GOSUB 1500:PRINT"Pour commencer , appuyez sur"
1400 : GOSUB 1500:PRINT"      une touche quelconque."
1410 : GET A$ 'Attente de l'appui sur une touche
1420 :
1430 : RETURN
1440 : REM *****
1450 : REM Couleur d'une ligne
1460 :
1470 : PRINT CHR$(27);"H";CHR$(27);A$;
1480 :
1490 : RETURN
1500 : REM *****
1510 : REM Initialisation du jeu
1520 :
1530 : GOSUB 1700 'Definition de l'avion
1540 : GOSUB 1800 'Tirage aleatoire du terrain d'atterrissage
1550 : GOSUB 1900 'Affichage des donnees de l'avion
1560 :
1570 : RETURN
1580 : REM *****
1590 : REM Definition de l'avion sur les lettres Y et Z
1600 :
1610 : FOR I=0 TO 15
1620 : READ A:POKE 46792+I,A
1630 : NEXT I
1640 DATA 0,0,0,0,32,49,63,63,0,0,0,0,0,48,62,62
1650 :
1660 : RETURN

```

```

2250 : PLOT XP,YP,"Y":PLOT XP+1,YP,"Z" 'Affichage de l'avion
2260 : IF XP=32 AND YP=21 THEN G=1:FIN=1 'Partie gagnee
2270 :
2280 : RETURN
2290 : REM *****
2300 : REM Commande du joueur
2310 :
2320 : IF ASC(A$)=10 THEN GOTO 2400 'Deplacement vers le bas
2330 : IF ASC(A$)=11 THEN GOTO 2500 'Deplacement vers le haut
2340 : IF A$=" " THEN GOTO 2600 'Demande de tir
2350 : IF A$="C" AND VG<>0 THEN GOTO 2700 'Demande de bombe verti
cale
2360 : IF A$="X" AND HL<>0 THEN GOTO 2800 'Demande de laser horiz
ontal
2370 :
2380 : RETURN
2390 : REM *****
2400 : REM Deplacement de l'avion vers le bas
2410 :
2415 : IF YP>20 THEN RETURN
2420 : PLOT XP,YP," ":PLOT XP+1,YP," " 'Effacement de l'avion
2430 : YP=YP+1 'L'avion descend
2440 : IF SCRN(XP,YP)<>32 OR SCRN(XP+1,YP)<>32 THEN FIN=1 'Collis
ion
2450 : PLOT XP,YP,"Y":PLOT XP+1,YP,"Z" 'Restitution de l'avion
2460 :
2470 : RETURN
2480 : REM *****
2500 : REM Deplacement de l'avion vers le haut
2510 :
2520 : IF YP<1 THEN RETURN
2530 : PLOT XP,YP," ":PLOT XP+1,YP," " 'Effacement de l'avion
2540 : YP=YP-1 'L'avion monte
2550 : PLOT XP,YP,"Y":PLOT XP+1,YP,"Z" 'Restitution de l'avion
2560 :
2570 : RETURN
2580 : REM *****
2600 : REM Demande de tir
2610 :
2620 : SHOOT 'Indique que la commande est prise en compte
2630 : FOR I=1 TO 8
2640 : IF YP+I<22 THEN PLOT XP+1,YP+I," "
2650 : NEXT I
2660 : CA=CA-30:IF CA<0 THEN FIN=1 'Gestion des ressources

```

```

2670 : PRINT CHR$(11);CHR$(11);" Carburant : ";CA;CHR$(10)
2680 :
2685 : RETURN
2690 : REM *****
2700 : REM Demande de bombe verticale
2710 :
2720 : EXPLODE 'Indique que la commande est prise en compte
2730 : FOR I=1 TO 8
2740 : IF YP+I<22 THEN PLOT XP+I,YP+I," ":PLOT XP,YP+I," "
2750 : NEXT I
2760 : CA=CA-80:VG=VG-1:IF CA<0 THEN FIN=1 'Gestion des ressource
s
2770 : PRINT CHR$(11);CHR$(11);" Carburant : ";CA;" Bombes ver
t.: ";VG;
2775 : PRINT CHR$(10)
2780 :
2785 : RETURN
2790 : REM *****
2800 : REM Demande de laser horizontal
2810 :
2815 : ZAP : IF HL=0 THEN RETURN
2820 : FOR I=2 TO 7
2830 : IF XP<30 THEN PLOT XP+I,YP," "
2840 : NEXT I
2850 : CA=CA-50:HL=HL-1:IF CA<0 THEN FIN=1 'Gestion des ressource
s
2860 : PRINT CHR$(11);CHR$(11);" Carburant : ";CA
2870 : PRINT" Laser horiz.: ";HL
2885 : RETURN
2890 : REM *****
3000 : REM Collision
3010 :
3020 : CLS : FIN=1 : PRINT CHR$(17)
3030 : IF G=1 THEN PRINT"Pas mal !!" ELSE PRINT"Ca devait arriver
!!!"
3040 : PRINT:PRINT:PRINT
3050 : INPUT"Une autre partie (O/N)";B$
3060 : IF B$="O" THEN RESTORE:FIN=0:XP=0:YP=0:GOTO 10
3070 :
3080 : RETURN

```

Mur de briques

PRESENTATION

Un mur de briques occupe la partie haute de l'écran. Le joueur déplace une raquette grâce aux touches-flèches droite et gauche. Le but du jeu est de détruire le mur de briques en perdant la balle 3 fois au maximum.

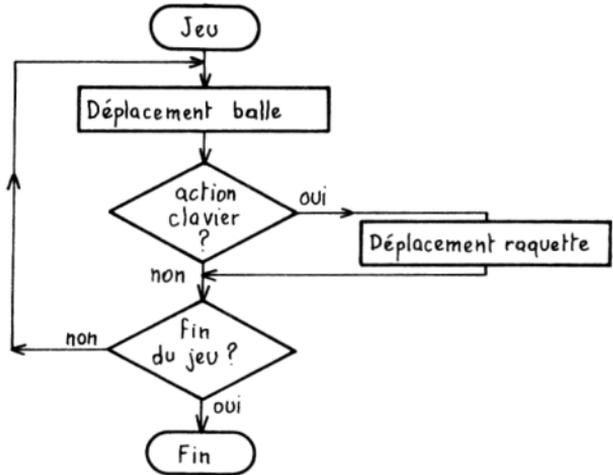
Le score est affiché en fin de partie.

ORGANIGRAMME DU JEU

Après la définition de la raquette, de la balle et des briques, le jeu commence. Il consiste en :

- . Déplacement de la balle,
- . Déplacement de la raquette,
- . Test de fin de partie.

Il se présente de la façon suivante :



LE PROGRAMME

Il se décompose en 3 pavés :

Initialisation

Lignes 1000 à 2550

Présentation

Lignes 1000 à 2000

Initialisation

Lignes 2000 à 2550

Jeu

Lignes 3000 à 4100

Module de commande

Lignes 3000 à 3060

Action balle

Lignes 3100 à 3360

Démolition brique

Lignes 3400 à 3490

Action joueur

Lignes 3600 à 3650

Déplacement raquette

Lignes 3700 à 3860

Fin de balle

Lignes 4000 à 4100

Fin du jeu

Lignes 5000 à 5090

LES VARIABLES UTILISEES

A\$

Code couleur de ligne

B\$

Code de déplacement raquette

BA

Nombre de balles utilisées

FIN

Indicateur de fin de partie

I, J

Index de boucles

SC

Score joueur

XB, YB

Position balle

XR, YR

Position raquette

VX, VY

Vecteur de déplacement balle

```

10 : REM Mur de briques
20 :
30 : GOSUB 1000 'Presentation du jeu
40 : GOSUB 2000 'Initialisation du jeu et jeu
50 : GOSUB 5000 'Fin du jeu
60 :
70 : END
80 : REM *****
1000 : REM Presentation du jeu
1010 :
1020 : INK 5 : PAPER 0 : PRINT CHR$(17)
1030 :
1040 : REM Titre
1050 :
1060 : PRINT CHR$(12)
1070 : PRINT CHR$(4);CHR$(27);"N";
1080 : PRINT CHR$(27);"A MUR DE BRIQUES";CHR$(4)
1090 :
1100 : REM Objet
1110 :
1120 : PRINT:PRINT:PRINT:PRINT
1130 : A$="D" 'Code couleur = Bleu
1140 : PRINT"OBJET : ";
1150 : GOSUB 1500:PRINT"Jeu de rapidite."
1160 :
1170 : REM Job
1180 :
1190 : PRINT:PRINT:PRINT:PRINT
1200 : A$="F" 'Code couleur = Cyan
1210 : PRINT"JOB : ";
1230 : GOSUB 1500:PRINT"Un mur de briques occupe la"
1240 : GOSUB 1500:PRINT" partie haute de l'ecran."
1250 : GOSUB 1500:PRINT" Une raquette peut se deplacer"
1260 : GOSUB 1500:PRINT" a gauche et a droite grace aux"
1270 : GOSUB 1500:PRINT" touches-fleches."
1280 : GOSUB 1500:PRINT" Le but du jeu est de detruire"
1290 : GOSUB 1500:PRINT" toutes les briques en faisant"
1300 : GOSUB 1500:PRINT" rebondir la balle sur la"
1310 : GOSUB 1500:PRINT" raquette."
1320 :

```

```
1330 : REM Debut
1340 :
1350 : PRINT:PRINT:PRINT:PRINT
1360 : A$="C" 'Code couleur = Jaune
1370 : PRINT"DEBUT : ";
1380 : GOSUB 1500:PRINT"Pour commencer , appuyez sur";
1390 : GOSUB 1500:PRINT" une touche quelconque."
1400 : GET A$ 'Attente de l'appui sur une touche
1410 :
1420 : RETURN
1430 : REM #####
1500 : REM Couleur d'une ligne
1510 :
1520 : PRINT CHR$(27);"H";CHR$(27);A$;
1530 :
1540 : RETURN
1550 : REM #####
2000 : REM Initialisation du jeu et jeu
2010 :
2020 : GOSUB 2100 'Initialisation du jeu
2030 : GOSUB 3000 'Jeu
2040 : GOSUB 4000 'Balle suivante
2050 : IF FIN=0 THEN GOTO 2030 ELSE RETURN
2060 :
2070 : REM #####
2100 : REM Initialisation du jeu
2110 :
2120 : REM Dessin d'ecran
2130 :
2140 : REM Definition de la brique et de la raquette
2150 :
2160 : FOR I=0 TO 15
2170 : READ A:POKE 46376+I,A
2180 : NEXT I
2190 DATA 0,0,42,20,42,20,42,0
2200 DATA 0,0,0,0,0,63,63,0
2210 :
2220 : REM Dessin des briques
2230 :
2240 : CLS
2250 : FOR I=1 TO 4
```

```

2260 : PRINT CHR$(27);"H";CHR$(27);CHR$(64+I);
2270 : FOR J=1 TO 38
2280 : PRINT"Z";
2290 : NEXT J
2300 : NEXT I
2310 :
2320 : REM Raquette
2330 :
2340 : XR=1:YR=20 'Position initiale
2350 : PLOT XR,YR,"&&&" 'Raquette
2360 :
2370 : REM Balle
2380 :
2390 : XB=3:YB=19 'Position initiale
2400 : VX=INT(RND(1)*3)-1:VY=-1 'Vecteur direction de la balle
2410 :
2420 : REM Debut de la partie et effichage du score
2430 :
2440 : FOR I=1 TO 19
2450 : PRINT
2460 : NEXT I
2465 : PRINT"          Balle 1":BA=1:PRINT
2470 : PRINT CHR$(27);"L";CHR$(27);"A";
2480 : PRINT"Appuyez sur une touche pour commencer"
2490 : GET A$
2500 : PRINT CHR$(11);CHR$(14) 'Effacement du commentaire
2510 : PRINT CHR$(11);:A$="B":GOSUB 1500
2520 : PRINT"          SCORE : ";SC
2530 :
2540 : RETURN
2550 : REM *****
3000 : REM Jeu
3010 :
3020 : GOSUB 3100 'Action sur la balle
3030 : B$=KEY$:IF B$(">)" THEN GOSUB 3600 'Action joueur
3040 : IF FIN=1 THEN RETURN ELSE GOTO 3000
3050 :
3060 : REM *****
3100 : REM Action sur la balle
3110 :
3120 : PLOT XB,YB," "
3130 : IF (YB+VY)<0 THEN VY=1:VX=INT(RND(1)*3)-1 'Haut de l'ecran

```

```

3131 : IF XB+VX<1 THEN VX=1:PLAY3,0,1,90
3132 : IF XB+VX=39 THEN VX=-1:PLAY3,0,1,90
3135 : IF SCRN(XB+VX,YB+VY)= 37 THEN GOSUB 3400 'Demolition brique
3140 : IF SCRN(XB+VX,YB+VY)=38 THEN GOSUB 3300 'Renvoi de la ball
3150 : IF YB+VY>20 THEN GOSUB 3500 'Perte de la balle
3155 : IF FIN=1 THEN RETURN
3160 : IF XB+VX<1 THEN VX=1:PLAY 3,0,1,90
3170 : IF XB+VX=39 THEN VX=-1:PLAY 3,0,1,90
3180 : XB=XB+VX:YB=YB+VY:PLOT XB,YB,"0"
3190 :
3195 : RETURN
3200 : REM *****
3300 : REM Renvoi de la balle
3310 :
3320 : VY=-1:VX=INT(RND(1)*3)-1
3330 : MUSIC 2,1,5,0:PLAY 3,0,1,200
3340 :
3350 : RETURN
3360 : REM *****
3400 : REM Demolition brique
3410 :
3420 : PLOT XB+VX,YB+VY," "
3430 : SC=SC+10*(4-YB-VY):PRINTCHR$(11);"          SCORE : ";S
C
3440 : VY=1:VX=INT(RND(1)*3)-1
3450 : IF SC>1300 THEN FIN=1 'Partie consideree comme gagnee
3460 : MUSIC 2,1,5,0:PLAY 3,0,1,200
3470 :
3480 : RETURN
3490 : REM *****
3500 : REM Balle perdue
3510 :
3520 : EXPLODE:FIN=1
3540 : RETURN
3550 : REM *****
3600 : REM Action du joueur
3610 :
3620 : IF ASC(B$)=8 THEN GOSUB 3700 'Deplacement vers la gauche
3630 : IF ASC(B$)=9 THEN GOSUB 3800 'Deplacement vers la droite

```

```

3640 :
3650 : RETURN
3660 : REM *****
3700 : REM Deplacement de la raquette vers la gauche
3710 :
3720 : IF XR=1 THEN RETURN
3730 : PLOT XR+1,YR," ":XR=XR-2:PLOT XR,YR,"&&"
3740 :
3750 : RETURN
3760 : REM *****
3800 : REM Deplacement de la raquette vers la droite
3810 :
3820 : IF XR=35 THEN RETURN
3830 : PLOT XR,YR," ":XR=XR+2:PLOT XR+1,YR,"&&"
3840 :
3850 : RETURN
3860 : REM *****
4000 : REM Fin de la balle
4010 :
4020 : BA=BA+1:IF 3A=4 THEN FIN=1:GOTO 4090 ELSE FIN=0
4030 : PRINT CHR$(11);CHR$(11);CHR$(11);" Balle ";BA
4040 : PRINT CHR$(27);"L";CHR$(27);"Appuyez sur une touche pour
commencer"
4050 : GET A$
4060 : PRINT CHR$(11);CHR$(14);CHR$(11);
4070 : XB=3;YB=19;VY=-1
4075 : PRINT:PRINT:PRINT
4080 :
4090 : RETURN
4100 : REM *****
5000 : REM Fin du jeu
5010 :
5020 : CLS
5030 : IF SC>1000 THEN PRINT"Pas trop mal."ELSE PRINT"Bof bof..."
5040 : IF BA=1 OR BA=2 THEN PRINT"Bravo!!"
5050 : PRINT CHR$(17)
5060 :
5070 : RETURN
5080 : REM *****
5090 :

```

Slalom ski

PRESENTATION

Un skieur animé d'un mouvement de translation continu est guidé par deux touches du clavier qui le font tourner dans le sens des aiguilles d'une montre ou dans le sens inverse.

Le but du jeu est de franchir les barrières d'un slalom inventé par ORIC. Ce jeu demande une rapidité de prise de décision, et une faculté d'improvisation certaine.

ORGANIGRAMME DU JEU

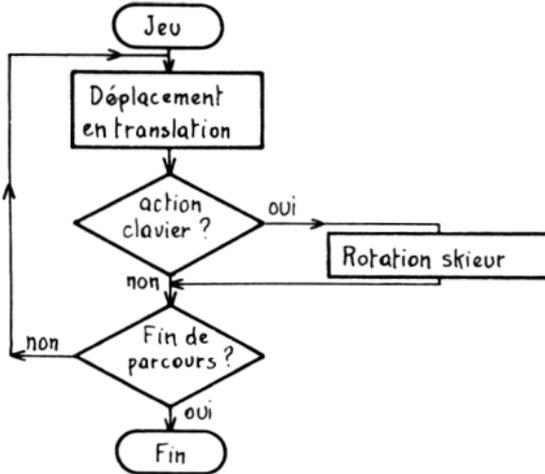
Le programme a la structure A.

Une grande partie du programme est consacrée à la définition des caractères graphiques du jeu : barrières et positions du skieur (5 positions de base).

Le jeu consiste :

- . à faire avancer le skieur (translation)
- . à le faire tourner par une action clavier
- . à tester la fin du parcours.

Ce qui correspond à l'organigramme suivant :



LE PROGRAMME

Il se décompose en 3 pavés :

Présentation et Initialisation

Lignes 1000 à 3420

Module de commande Lignes 1000 à 1060

Présentation du jeu Lignes 1100 à 1550

Initialisation Lignes 1600 à 3420

(Définition skieur, barrières et affichage du parcours).

Jeu

Lignes 3500 à 4170

Module de commande Lignes 3500 à 3570

Avance skieur Lignes 3600 à 4170

L'avance skieur comporte la translation et la rotation.

Fin de jeu

Lignes 5000 à 5150

Commentaires sur le jeu

LES VARIABLES UTILISEES

A\$	Code couleur de ligne
B\$	Code action clavier
I	Index de boucles
FIN	Indicateur de fin de partie
NB	Direction du skieur

S Indicateur de sortie de piste
 VX, VY Vecteurs de changement de direction skieur
 X Position d'un passage
 X1, X2, Y1, Y2 Coordonnées de la position skieur.

```

10 : REM Slalom ski
20 :
30 : GOSUB 1000 'Presentation et Initialisation
40 : GOSUB 3000 'Jeu
50 : GOSUB 5000 'La partie est terminee
60 : IF R$="o" THEN GOTO 40 'Une autre partie
70 :
80 : END
90 : REM *****
1000 : REM Presentation et Initialisation
1010 :
1020 : GOSUB 1100 'Presentation
1030 : GOSUB 1600 'Initialisation
1040 :
1050 : RETURN
1060 : REM *****
1100 : REM Presentation du jeu
1110 :
1120 : PAPER 0 : INK 6 : PRINT CHR$(17);CHR$(20)
1130 :
1140 : REM Titre
1150 :
1160 : PRINT CHR$(12)
1170 : PRINT CHR$(4);CHR$(27);"N";
1180 : PRINT CHR$(27);"A SLALOM SKI";CHR$(4)
1190 :
1200 : REM Objet
1210 :
1220 : PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
1230 : A$="D" 'Code couleur = Bleu
1240 : PRINT"OBJET :";
1250 : GOSUB 1500:PRINT"Jeu de rapidite."
1260 :

```

```

1270 : REM Job
1280 :
1290 : PRINT:PRINT:PRINT:PRINT
1300 : A$="F" 'Code couleur = Cyan
1310 : PRINT"JOB  :";
1320 : GOSUB 1500:PRINT"Vous vous trouvez sur une"
1330 : GOSUB 1500:PRINT" piste de ski et vous devez"
1340 : GOSUB 1500:PRINT" effectuer un slalom en vous"
1350 : GOSUB 1500:PRINT" servant des touches T et I"
1360 : GOSUB 1500:PRINT" effectuer une rotation dans le"
1370 : GOSUB 1500:PRINT" Trigo et Inverse Trigo."
1380 :
1390 : REM Debut
1400 :
1410 : PRINT:PRINT:PRINT:PRINT
1420 : A$="C" 'Code couleur = Jaune
1430 : PRINT"DEBUT :";
1440 : GOSUB 1500:PRINT"Pour commencer , appuyez sur"
1450 : GOSUB 1500:PRINT" une touche quelconque."
1460 : GET A$ 'Attente de l'appui sur une touche
1470 :
1480 : RETURN
1490 : REM *****
1500 : REM Couleur d'une ligne
1510 :
1520 : PRINT CHR$(27);"H";CHR$(27);A$;
1530 :
1540 : RETURN
1550 : REM *****
1600 : REM Initialisation
1610 :
1615 : CLS
1620 : REM Differentes positions du skieur
1630 :
1640 : FOR I=0 TO 127
1650 : READ A:POKE 46600+I,A
1660 : NEXT I
1670 :

```

```
1680 : REM Passages
1690 :
1700 :   FOR I=0 TO 7
1710 :     READ A:POKE 46728+I,A
1720 :     NEXT I
1730 :
1740 : REM Donnees skieur et passages
1750 :
1760 DATA 3,3,1,29,19,17,17,19
1770 DATA 32,32,0,0,0,48,12,4
1780 DATA 4,31,0,1,0,0,0,0
1790 DATA 36,52,20,63,4,0,0,0
1800 :
1810 DATA 3,3,1,1,9,7,27,51
1820 DATA 32,32,4,28,36,4,4,0
1830 DATA 20,24,20,2,1,0,0,0
1840 DATA 32,16,8,0,0,0,0,0
1850 :
1860 DATA 3,3,1,1,31,17,21,22
1870 DATA 32,32,0,0,60,4,20,52
1880 DATA 20,4,4,4,0,0,0,0
1890 DATA 20,16,16,16,0,0,0,0
1900 :
1910 DATA 3,3,49,45,35,35,35,7
1920 DATA 32,32,0,0,0,48,12,12
1930 DATA 8,16,32,1,2,0,0,0
1940 DATA 52,36,36,0,0,0,0,0
1950 :
1960 DATA 32,48,40,36,62,32,32,32
1970 :
1980 : REM Directions en X et en Y
1990 :
2000 :   FOR I=1 TO 5
2010 :     READ DX(I)
2020 :     NEXT I
2030 :
2040 :   FOR I=1 TO 5
2050 :     READ DY(I)
```

```

2060 :     NEXT I
2070 :
2080 DATA 1,1,0,-1,-1 'Directions en X
2090 DATA 0,1,1,1,0  'Directions en Y
2100 :
2110 : RETURN
2120 : REM *****
3000 : REM Initialisation du jeu et jeu
3010 :
3020 : GOSUB 3100 'Initialisation du jeu
3030 : GOSUB 3500 'Deroulement du jeu
3040 :
3050 : RETURN
3060 : REM *****
3100 : REM Initialisation du jeu
3110 :
3120 : REM Tirage des positions des passages
3130 :
3140 :     FOR I=1 TO 5
3150 :         X=INT(RND(1)*24)+9:TX(I)=X
3160 :     NEXT I
3170 :
3180 : REM Affichage du parcours
3190 :
3200 :     CLS
3210 :     FOR I=1 TO 5
3220 :         PLOT TX(I),(I-1)*5+1,"Q":PLOT TX(I),(I-1)*5+4,"Q"
3230 :     NEXT I
3240 :
3250 : REM Joueur au depart
3260 :
3270 :     X1=2:Y1=2:X2=3:Y2=3:VX=1:VY=0:NB=1:C$="A":D$="B":E$="C":
F$="D"
3280 :     PLOT X1,Y1,"A":PLOT X2,Y1,"B":PLOT X1,Y2,"C":PLOT X2,Y2,
"D"
3290 :
3300 : REM Commentaires
3310 :

```

```

3320 : PRINT CHR$(27);"L";CHR$(27);"C" slalom ski"
3330 : FOR I=1 TO 24
3340 : PRINT
3350 : NEXT I
3360 : PRINT CHR$(27);"L";CHR$(27);"D";
3370 : PRINT"pour commencer,appuyez sur une touche"
3380 : GET A$
3390 : PRINT CHR$(11);CHR$(14)
3400 :
3410 : RETURN
3420 : REM *****
3500 : REM Jeu
3510 :
3520 : GOSUB 3600 'Avance skieur
3530 : B$=KEY$:IF B$(">)" THEN GOSUB 3800 'Changement de direction

3540 : IF FIN(">") THEN GOTO 3500
3550 :
3560 : RETURN
3570 : REM *****
3600 : REM Avance skieur
3610 :
3620 : PLOT X1,Y1," ":PLOT X2,Y1," ":PLOT X1,Y2," ":PLOT X2,Y2,"
"
3630 : X1=X1+VX:X2=X2+VX:Y1=Y1+VY:Y2=Y2+VY
3640 : IF X1=0 OR X1=36 OR Y1=0 OR Y1=25 THEN FIN=1:S=1:GOTO 3700
3650 : IF SCRN(X1,Y1-1)=81 AND SCRN(X1,Y1+2)=81 THEN GA=GA+1:PING
3655 : IF X1=34 AND Y1=22 THEN FIN=1:S=0 'Partie finie
3660 : IF SCRN(X1,Y1)=81 OR SCRN(X1,Y2)=81 THEN EXPLODE
3670 : IF SCRN(X2,Y1)=81 OR SCRN(X2,Y2)=81 THEN EXPLODE
3680 : PLOT X1,Y1,C$:PLOT X2,Y1,D$:PLOT X1,Y2,E$:PLOT X2,Y2,F$
3690 :
3700 : RETURN
3710 : REM *****
3800 : REM Changement de direction
3810 :
3820 : IF B$="t" THEN GOSUB 3900 'Sens trigo
3830 : IF B$="i" THEN GOSUB 4000 'Sens Inverse Trigo
3840 :
3850 : RETURN

```

```

3860 : REM *****
3900 : REM Sens trigo
3910 :
3920 : IF NB= 1 THEN RETURN
3930 : NB=NB-1:GOTO 4100
3940 :
3950 : REM *****
4000 : REM Sens Inverse Trigo
4010 :
4020 : IF NB=5 THEN RETURN
4030 : NB=NB+1
4040 :
4050 : REM *****
4100 : REM Fin du changement de direction
4110 :
4120 : VX=DX(NB) : VY=DY(NB)
4130 : IF NB=5 THEN IN=65 ELSE IN=65+4*(NB-1)
4140 : C$=CHR$(IN):D$=CHR$(IN+1):E$=CHR$(IN+2):F$=CHR$(IN+3)
4150 :
4160 : RETURN
4170 : REM *****
5000 : REM La partie est terminee
5010 :
5020 : CLS
5030 : IF S<>1 THEN GOTO 5050
5040 : PRINT"alors comme ca, on sort des pistes !!":GOTO 5090
5050 : PRINT:PRINT:PRINT
5060 : PRINT"parcours effectue avec ";5-GA;" fautes(s)."
5070 : IF GA=5 THEN PRINT"pas mal." ELSE PRINT"minable !!!"
5080 :
5090 : PRINT:PRINT:PRINT
5100 : PRINT CHR$(17)
5110 : INPUT"une autre partie (o/n) ";R$
5120 : S=0:FIN=0:GA=0:PRINT CHR$(17)
5130 :
5140 : RETURN
5150 : REM *****

```


DEUXIÈME PARTIE
JEUX DE RÉFLEXION

Awari

PRESENTATION

Ce jeu de réflexion est actuellement très prisé. Il se déroule sur un parcours rectangulaire où vous occupez la partie basse et ORIC la partie haute. Ce parcours est divisé en cases. Chaque case contient un nombre. Au départ, ce nombre est 3 pour chaque case. A tour de rôle vous, puis ORIC, allez choisir une des cases de votre camp en la désignant par une flèche. Cette case sera forcée à \emptyset . Si son contenu était N, les N cases situées en aval (dans une rotation de sens trigonométrique) seront incrémentées de 1. Les cases latérales constituent les totaux de nombres déplacés. Si l'incrémentation finit sur une telle case, un nouveau coup peut être tenté.

Le but du jeu est double :

- . Totaliser le plus de points possibles dans SA case résultat
- . Annuler toutes les cases de SON camp

ORGANIGRAMME DU JEU

Le programme a la structure A.

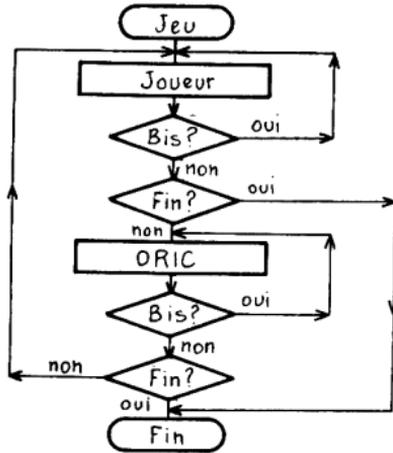
Le jeu se divise en 2 parties :

- . Joueur
- . ORIC

Après chaque jeu, 2 tests sont à faire :

- . Vous ou ORIC pouvez rejouer ?
- . La partie est finie ? (annulation de toutes les cases).

L'organigramme du jeu est donc le suivant :



Le jeu du joueur se décompose en :

- . Affichage de commentaires sur le jeu
- . Gestion de la flèche
- . Action sur les cases

Le jeu ORIC se décompose en :

- . Choix du meilleur coup
- . Affichage du coup joué
- . Action sur les cases

Il peut être intéressant d'expliciter la technique employée par ORIC pour sélectionner le meilleur coup :

ORIC calcule le maximum de points qu'il peut faire en une opération et joue ce coup là.

D'autres techniques intéressantes peuvent être employées :

- . Calculer le coup qui élimine le plus de points dans son camp
- . Calculer le coup qui donne le plus de points possibles dans le camp adverse

Si, au bout d'un moment, le jeu d'ORIC n'a plus de secrets pour vous, vous pouvez essayer de changer sa tactique de jeu, et même lui offrir la possibilité de se servir de plusieurs tactiques.

LE PROGRAMME

Il se décompose en 3 pavés :

<i>Présentation</i>	Lignes 1000 à 1550	
<i>Initialisation du jeu et jeu</i>	Lignes 2000 à 5350	
	Initialisation	Lignes 2000 à 2460
	Module de commande	Lignes 2500 à 2570
	Jeu utilisateur	Lignes 3000 à 3630
	Déplacement droite	Lignes 3200 à 3260
	Déplacement gauche	Lignes 3300 à 3360
	Jeu du joueur	Lignes 3400 à 3630
	Jeu ORIC	Lignes 4000 à 4280
	Calculs et affichage du coup joué	Lignes 5000 à 5350
<i>Fin du jeu</i>	Lignes 6000 à 6160	
	Commentaires sur la partie	

LES VARIABLES UTILISEES

A\$	Code couleur de ligne
A	Variable de calcul
B\$	Lettre d'acquisition clavier
C, D	Choix du meilleur coup
CASE	Numéro de case choisie
E	Variable d'affichage
GA	Nom du vainqueur
I, J	Index de boucle
JEU	Indicateur de jeu (joueur/ORIC)
PX	Position de la flèche
SO	Total des cases d'un des camps
T	Tableau de jeu

```

10 : REM AWARI
20 :
30 : GOSUB 1000 'Presentation du jeu
40 : GOSUB 2000 'Joueur contre ORIC
50 : GOSUB 6000 '1 des joueurs a gagne
60 :
70 : END
80 : REM *****
1000 : REM Presentation du jeu
1010 :
1020 : INK 6 : PAPER 0 : PRINT CHR$(17)
1030 :
1040 : REM Titre
1050 :
1060 : PRINT CHR$(12)
1070 : PRINT CHR$(4);CHR$(27);"N";
1080 : PRINT CHR$(27);"A AWARI";CHR$(4)
1090 :
1100 : REM Objet
1110 :
1120 : PRINT:PRINT:PRINT
1130 : A$="D" 'Code couleur = Bleu
1140 : PRINT"OBJET :";
1150 : GOSUB 1500:PRINT"Jeu de reflexion."
1160 :
1170 : REM Job
1180 :
1190 : PRINT:PRINT
1200 : A$="F" 'Code couleur = Cyan
1210 : PRINT"JOB :";
1220 : GOSUB 1500:PRINT"Le but du jeu est d'annuler"
1230 : GOSUB 1500:PRINT" le nombre de points dans son"
1240 : GOSUB 1500:PRINT" camp(Vous en bas,ORIC en haut)"
1250 : GOSUB 1500:PRINT" en tenant compte des regles : "
1260 : PRINT:GOSUB 1500:PRINT" 1.Le jeu d'une case conte
nant"
1270 : GOSUB 1500:PRINT" n pions provoque n additions"
;

```

```

1280 :   GOSUB 1500:PRINT"           d'un pion sur les n cases "
1290 :   GOSUB 1500:PRINT"           situees en aval(Sens trigo)."

```

```

2150 : FOR I=0 TO 15
2160 :   READ A:POKE 46792+I,A
2170 :   NEXT I
2180 :
2190 DATA 0,0,34,54,28,8,0,0
2200 DATA 0,0,8,28,54,34,0,0
2210 :
2220 : REM Affichage du jeu
2230 :
2240 : CLS
2250 : PRINT CHR$(27);"L";CHR$(27);"C           AWARI"
2260 : FOR I=1 TO 9
2270 :   PRINT
2280 : NEXT
2290 : FOR I=1 TO 11
2300 :   PRINT CHR$(27);"H";CHR$(27);"T"
2310 : NEXT I
2320 : FOR I=0 TO 1
2330 :   FOR J=0 TO 5
2340 :     PLOT 7+J*4,12+I*6,"I":PLOT 9+J*4,12+I*6,51
2350 :     NEXT J
2360 : NEXT I
2370 : PLOT 31,12,"I":PLOT 31,1,"I":PLOT 5,15,48:PLOT 33,15,48
2380 :
2390 : DIM T(14)
2400 : FOR I=1 TO 14
2410 :   T(I)=3
2420 : NEXT I
2430 : T(1)=0:T(8)=0 'Totaux ORIC et joueur
2440 :
2450 : RETURN
2460 : REM #####
2500 : REM Jeu
2510 :
2520 : GOSUB 3000 'Jeu joueur
2524 : IF FIN=1 THEN RETURN
2525 : IF BIS=1 THEN GOTO 2520
2530 : GOSUB 4000 'Jeu ORIC

```

```
2535 : IF BIS=1 THEN GOTO 2530
2540 : IF FIN=0 THEN GOTO 2500
2550 :
2560 : RETURN
2570 : REM *****
3000 : REM Jeu de l'utilisateur
3010 :
3020 : PLOT 9,20,"Z":PX=0:BIS=0:SO=0
3025 : PLOT 1,5,"A vous de jouer"
3030 : PLOT 1,22,"Servez vous des touches fleches droite"
3040 : PLOT 1,23,"et gauche pour positionner la case"
3050 : PLOT 1,24,"desiree.Appuyez sur T quand vous vous"
3060 : PLOT1,25,"serez positionne."
3070 : B$=KEY$:IF B$("<") THEN B=ASC(B$)
3080 : IF B=9 THEN GOSUB 3200 'Deplacement vers la droite
3090 : IF B=8 THEN GOSUB 3300 'Deplacement vers la gauche
3100 : IF B$="T" THEN GOSUB 3400
3110 : IF B$="T" THEN RETURN
3115 : GOTO 3070
3120 :
3130 : REM *****
3200 : REM Deplacement vers la droite
3210 :
3220 : IF PX=5 THEN RETURN
3230 : PLOT 9+PX*4,20," ":PX=PX+1:PLOT 9+PX*4,20,"Z":B=0
3240 :
3250 : RETURN
3260 : REM *****
3300 : REM Deplacement vers la gauche
3310 :
3320 : IF PX=0 THEN RETURN
3330 : PLOT 9+PX*4,20," ":PX=PX-1:PLOT 9+PX*4,20,"Z":B=0
3340 :
3350 : RETURN
3360 : REM *****
3400 : REM La fleche est positionnee
3410 :
```

```
3420 : REM Effacement des informations inscrites sur l'ecran
3430 :
3440 :   FOR I=1 TO 5
3450 :     PRINT CHR$(14)
3460 :   NEXT I
3470 :   FOR I=1 TO 5
3480 :     PRINT CHR$(11);
3490 :   NEXT I
3500 :   PLOT 9+PX*4,20," "
3510 :
3520 : REM Calcul et affichage du coup joue
3530 :
3540 :   JEU=0:GOSUB 5000
3550 :   PLOT 1,5,"
3560 :
3570 :   FOR I=2 TO 7
3580 :     SO=SO+T(I)
3590 :   NEXT I
3600 :   IF SO=0 THEN FIN=1:GA=1
3610 :
3620 : RETURN
3630 : REM *****
4000 : REM Jeu d'ORIC
4010 :
4020 :   MAX=-100:BIS=0:SO=0
4030 :   REM Apres calcul,MAX contiendra le maximum
4040 :   REM de points que peut faire ORIC en 1 coup
4050 :
4060 :   FOR I=9 TO 14
4070 :     C=0:D=0
4075 :     IF T(I)=0 THEN GOTO 4140
4080 :     FOR J=1 TO T(I)
4090 :       D=I+J
4100 :       IF D<15 AND D>8 THEN C=C-1
4110 :       IF D>1 AND D<8 THEN C=C+1
4120 :     NEXT J
4130 :     IF C>MAX THEN MAX=C:CASE=I
4140 :   NEXT I
```

```

4150 : PX=CASE-2
4160 :
4170 : PLOT 1,5,"Je joue en "
4190 : JEU=1:GOSUB 5000
4200 : PLOT 1,5,"          "
4210 :
4220 :   FOR I=9 TO 14
4230 :     SO=SO+T(I)
4240 :   NEXT I
4250 :   IF SO=0 THEN FIN=1:GA=0
4260 :
4270 : RETURN
4280 : REM *****
5000 : REM Calcul et affichage du coup joue
5010 :
5020 :   REM Calcul
5030 :
5035 :   A=PX+2:IF T(A)=0 THEN RETURN
5040 :   FOR I=1 TO T(A)
5050 :     IF A+I<=14 THEN T(A+I)=T(A+I)+1
5060 :     IF A+I>14 THEN T(A+I-14)=T(A+I-14)+1
5070 :   NEXT I
5074 :   IF A+I=9 OR A+I=16 THEN BIS=1
5075 :   T(A)=0
5076 :   IF JEU=1 THEN PLOT 9+(14-A)*4,10,"Y":WAIT 100
5080 :
5090 :   REM Affichage
5100 :
5110 :   FOR I=0 TO 5
5120 :     E=INT(T(14-I)/10)
5130 :     IF E<>0 THEN POKE 48529+I*4,E+48
5140 :     POKE 48530+I*4,T(14-I)-10*E+48
5160 :   NEXT I
5170 :
5180 :   FOR I=0 TO 5
5190 :     E=INT(T(I+2)/10)
5200 :     IF E<>0 THEN POKE 48769+I*4,E+48
5210 :     POKE 48770+I*4,T(I+2)-E*10+48

```

```
5220 : NEXT I
5230 :
5240 : E=INT(T(1)/10)
5250 : IF E<>0 THEN POKE 48646,E+48
5260 : POKE 48647,T(1)-E*10+48
5270 :
5280 : E=INT(T(8)/10)
5290 : POKE 48673,E+48
5300 : POKE 48674,T(8)-E*10+48
5310 :
5320 : IF JEU=1 THEN PLOT 9+(14-A)*4,10," "
5330 :
5340 : RETURN
5350 : REM *****
6000 : REM Partie terminee
6010 :
6020 : CLS
6030 : PRINT"Partie terminee"
6040 : PRINT:PRINT:PRINT
6050 : IF GA=1 THEN PRINT"Bravo";
6060 : IF GA=0 THEN PRINT"Bof";
6070 : PRINT" pour le jeu."
6080 : PRINT:PRINT
6090 : IF T(1)<T(8) THEN PRINT"Bravo";
6095 : IF T(1)>T(8) THEN PRINT"Bof";
6100 : PRINT" pour les points."
6110 : PRINT:PRINT
6120 : PRINT"ORIC: ";T(1);" Vous: ";T(8)
6130 : PRINT CHR$(17)
6140 :
6150 : RETURN
6160 : REM *****
```

Black box

PRESENTATION

Trois trous noirs sont placés dans une boîte quadrillée.

Ces trous noirs sont capables de dévier ou d'absorber la lumière issue d'un tube à effet laser.

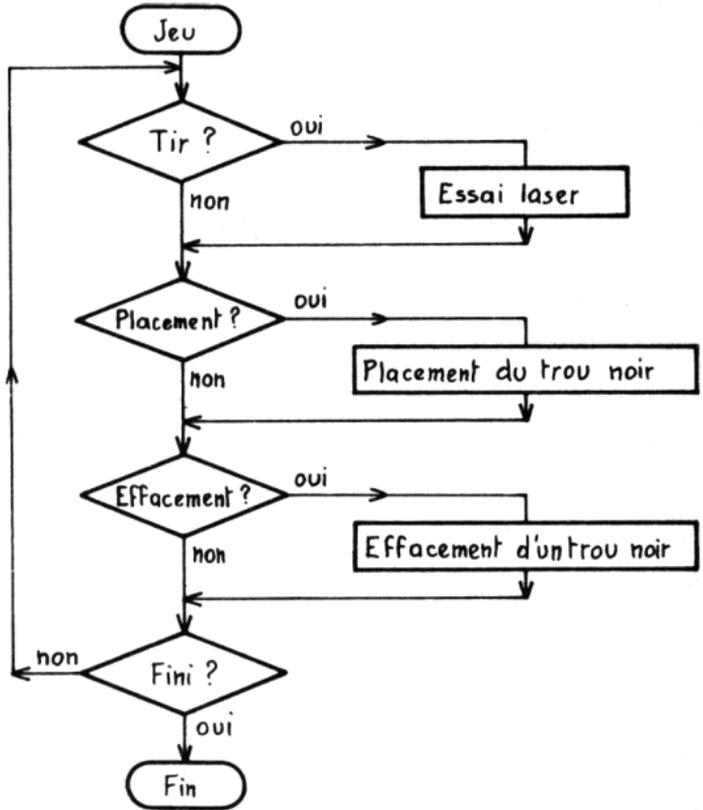
Il y aura déviation à 90° si le rayon passe dans une case située à côté d'un trou noir ; il y aura absorption si le rayon passe sur une case contenant un trou noir.

Le but du jeu est de positionner les trous noirs en un minimum de coups grâce à un laser orientable...

ORGANIGRAMME DU JEU

Le programme a la structure A.

Le jeu a la structure suivante :



LE PROGRAMME

Il se décompose en 3 pavés :

Initialisation

Lignes 1000 à 1950
 Définition des caract. Lignes 1100 à 1360
 Présentation Lignes 1400 à 1950

Jeu

Lignes 2000 à 4790
 Le jeu se décompose en 8 modules :
 Demande de tir Lignes 2500 à 2690
 Déplacement droite Lignes 2700 à 2790
 Déplacement gauche Lignes 2800 à 2890
 Déplacement bas Lignes 2900 à 2990
 Déplacement haut Lignes 3000 à 3090
 Essai laser Lignes 3100 à 3330
 Placement trou noir Lignes 4100 à 4320
 Effacement trou noir Lignes 4400 à 4600
 et six sous-modules :
 Déplac. laser gauche Lignes 3400 à 3480

Déplac. laser haut	Lignes 3500 à 3580
Déplac. laser bas	Lignes 3600 à 3680
Déplac. laser droite	Lignes 3700 à 3780
Repères sur grille	Lignes 3800 à 4020
Test de fin de partie	

Fin de partie

Lignes 5000 à 5080
 Tous les trous noirs ont été trouvés.
 Ce module affiche le nombre d'erreurs de placement et commente le jeu.

LES VARIABLES UTILISEES

A	Position du caractère entré
A\$	Code couleur de ligne
B	Position abscisse du caractère entré
BO	Indicateur d'absorption
B\$	Action joueur
C	Position du caractère de sortie
CO	Position colonne d'un trou noir
C\$	Action joueur déplacement
D	Position abscisse du caractère de sortie
DE	Sens de déplacement de la flèche
ER	Nombre d'erreurs
FIN	Indicateur de fin de partie
FF	Indicateur de sortie du laser
LE	Code lettre entrée/sortie
LI	Position ligne d'un trou noir
NF	Indicateur de partie finie/non finie
R	Tableau joueur
T	Tableau ORIC
XP	Abscisse laser
YP	Ordonnée laser

```

10 : REM Black box
20 :
30 : GOSUB 1000 'Initialisation & Presentation
40 : GOSUB 2000 'Jeu
50 : GOSUB 5000 'Fin de la partie
60 :
70 : END
80 : REM *****
1000 : REM Presentation et Initialisation
1010 :
1020 : GOSUB 1100 'Initialisation
1030 : GOSUB 1400 'Presentation
1040 :
1050 : RETURN
1060 : REM *****
1100 : REM Initialisation
1110 :
1120 : DIM T(4,9),R(4,9) 'Tableaux de jeu
1130 :
1140 : REM Remplissage tableau
1150 :
1160 : FOR I=1 TO 3
1170 : A=INT(RND(1)*2)+2:B=INT(RND(1)*6)+2
1180 : IF T(A,B)=1 THEN I=I-1 ELSE T(A,B)=1
1190 : NEXT I
1200 :
1210 : REM Definition des fleches
1220 :
1230 : FOR I=0 TO 39
1240 : READ A : POKE 46808+I,A
1250 : NEXT I
1260 DATA 0,4,12,24,24,12,4,0 'Vers la droite
1270 DATA 0,0,12,30,51,0,0,0 'Vers le haut
1280 DATA 0,0,51,30,12,0,0,0 'Vers le bas
1290 DATA 0,8,12,6,6,12,8,0 'Vers la droite
1295 DATA 8,8,8,8,8,0,0,0 'Trait vertical
1300 :
1310 : REM Position des indicateurs
1320 :
1330 : XP=2 : YP=7 'En bas a droite
1340 :
1350 : RETURN

```

```

1360 : REM *****
1400 : REM Presentation
1410 :
1420 : PAPER 0 : INK 5 : PRINT CHR$(17)
1430 : REM Titre
1440 :
1450 : PRINT CHR$(12)
1460 : PRINT CHR$(4);CHR$(27);"N";
1470 : PRINT CHR$(27);"A BLACK BOX";CHR$(4)
1480 :
1490 : REM Objet
1500 :
1510 : PRINT:PRINT:PRINT
1520 : A$="D" 'Code couleur = Bleu
1530 : PRINT"OBJET : ";
1540 : GOSUB 1900:PRINT"Jeu de reflexion."
1550 :
1560 : REM JOB
1570 :
1580 : PRINT:PRINT
1590 : A$="F" 'Code couleur = Cyan
1600 : PRINT"JOB : ";
1610 : GOSUB 1900:PRINT"Une boite noire renferme"
1620 : GOSUB 1900:PRINT" 3 trous noirs capables de"
1630 : GOSUB 1900:PRINT" devier ou d'absorber le rayon"
1640 : GOSUB 1900:PRINT" issu d'un tube a effet laser"
1650 : GOSUB 1900:PRINT" selon les 2 lois suivantes : "
1660 : GOSUB 1900:PRINT" \ \"
1670 : GOSUB 1900:PRINT" - - - - -"
1680 : GOSUB 1900:PRINT" - - - - - Deviation - - - - -"
1690 : GOSUB 1900:PRINT" [-- --^ ]"
1700 : GOSUB 1900:PRINT" 0 ou --^ [--]"
1710 : GOSUB 1900:PRINT" [-- --^ \ \"
1720 : GOSUB 1900:PRINT" - - - - - Absorbction - - - - -"
1730 : GOSUB 1900:PRINT" - - - - -"
1740 : GOSUB 1900:PRINT" ] ]"
1750 :
1760 : REM Debut
1770 :
1780 : PRINT:PRINT
1790 : A$="C" 'Code couleur = Jaune
1800 : PRINT"DEBUT : ";

```

```

1810 : GOSUB 1900:PRINT"Pour commencer , appuyez sur";
1820 : GOSUB 1900:PRINT"      une touche quelconque."
1830 : GET A$ 'Attente de l'appui sur une touche
1840 :
1850 : RETURN
1860 : REM *****
1900 : REM Couleur d'une ligne
1910 :
1920 : PRINT CHR$(27);"H";CHR$(27);A$;
1930 :
1940 : RETURN
1950 : REM *****
2000 : REM Jeu
2010 :
2020 : GOSUB 2100 'Initialisation du jeu
2030 : GOSUB 2400 'Deroulement du jeu
2040 :
2050 : RETURN
2060 : REM *****
2100 : REM Initialisation du jeu
2110 :
2120 : REM Titre
2130 :
2140 : CLS : INK 6
2150 : PRINT CHR$(27);"L";CHR$(27);"C                BLACK BOX"
2160 :
2170 : REM Affichage de la grille du jeu
2180 :
2190 : FOR I=1 TO 5
2200 :   FOR J=1 TO 10
2210 :     PLOT 8+2*I,6+2*J, "."
2220 :   NEXT J
2230 : NEXT I
2240 :
2250 : REM Affichage des commandes
2260 :
2270 : FOR I=1 TO 10
2271 :   PRINT CHR$(10)
2272 : NEXT I
2273 : A$="F" 'Code couleur = Cyan
2274 : GOSUB 1900:PRINT"  Commandes : "
2275 : PRINT

```

```

2276 :   GOSUB 1900:PRINT"   T=Tir , P=Placement , E=Effacement"
2290 :
2300 :   REM Position de la fleche
2310 :
2320 :   PLOT 11,17,"\"
2330 :   XP=2 : YP=7 'En bas a gauche
2340 :
2350 : RETURN
2360 : REM #####
2400 : REM Deroulement du jeu
2410 :
2420 :   GET B$
2430 :   IF B$="T" THEN GOSUB 2500
2435 FF=0:BO=0
2440 :   IF B$="P" THEN GOSUB 4100
2450 :   IF B$="E" THEN GOSUB 4400
2460 :   IF FIN<>1 THEN GOTO 2400 'Le jeu continue
2470 :
2480 : RETURN
2490 : REM #####
2500 : REM Demande de tir
2510 :
2520 :   REM Affichage de la commande
2530 :
2540 :   PRINT CHR$(11);CHR$(11)
2550 :   PRINTCHR$(27);"H";CHR$(27);"A   T=Tir";CHR$(27);"H";CHR$(27);"F"
2560 :   PLOT 0,3," Servez-vous des touches-fleches pour"
2570 :   PLOT 0,4," positionner l'origine du tir."
2580 :   PLOT 0,5," La touche T commande le tir."
2590 :
2600 :   REM Positionnement de l'origine du tir
2610 :
2620 :   GET C$
2630 :   IF ASC(C$)=9 THEN GOSUB 2700
2640 :   IF ASC(C$)=8 THEN GOSUB 2800
2650 :   IF ASC(C$)=10 THEN GOSUB 2900
2660 :   IF ASC(C$)=11 THEN GOSUB 3000
2670 :   IF C$<>"T" THEN GOTO 2620
2675 :   GOSUB 3100:RETURN
2680 :

```

```

2690 : REM *****
2700 : REM Deplacement vers la droite demande
2710 :
2720 : IF XP>9 OR YP=1 OR YP=4 OR YP=5 THEN RETURN
2730 : C=SCRN(XP#2+7,YP#2+3) 'Sauvegarde de la fleche
2740 : PLOT XP#2+7,YP#2+3," "
2750 : IF YP=3 THEN YP=YP-1:C=ASC("]")
2755 : IF YP=6 THEN YP=YP+1:C=ASC("\")
2760 : XP=XP+1:PLOT XP#2+7,YP#2+3,C
2770 :
2780 : RETURN
2790 : REM *****
2800 : REM Deplacement vers la gauche
2810 :
2820 : IF XP<3 OR YP=1 OR YP=4 OR YP=5 THEN RETURN
2830 : C=SCRN(XP#2+7,YP#2+3) 'Sauvegarde de la fleche
2840 : PLOT XP#2+7,YP#2+3," "
2845 : IF YP=3 THEN YP=YP-1:C=ASC("]")
2850 : IF YP=6 THEN YP=YP+1:C=ASC("\")
2860 : XP=XP-1:PLOT XP#2+7,YP#2+3,C
2870 :
2880 : RETURN
2890 : REM *****
2900 : REM Deplacement vers le bas
2910 :
2920 : IF YP>5 OR (XP<>1 AND XP<>2 AND XP<>10 AND XP<>11) THEN RE
TURN
2930 : C=SCRN(XP#2+7,YP#2+3) 'Sauvegarde de la fleche
2940 : PLOT XP#2+7,YP#2+3," "
2945 : IF XP=10 THEN XP=XP+1:C=ASC("[")
2950 : IF XP=2 THEN XP=XP-1:C=ASC("^")
2955 : IF XP=10 THEN XP=XP+1:C=ASC("[")
2960 : YP=YP+1:PLOT XP#2+7,YP#2+3,C
2970 :
2980 : RETURN
2990 : REM *****
3000 : REM Deplacement vers le haut
3010 :
3020 : IF YP<4 OR (XP<>1 AND XP<>2 AND XP<>10 AND XP<>11) THEN RE
TURN
3030 : C=SCRN(XP#2+7,YP#2+3) 'Sauvegarde de la fleche
3040 : PLOT XP#2+7,YP#2+3," "

```

```

3050 : IF XP=2 THEN XP=XP-1:C=ASC("^")
3055 : IF XP=10 THEN XP=XP+1:C=ASC("[")
3060 : YP=YP-1:PLOT XP*2+7,YP*2+3,C
3070 :
3080 : RETURN
3090 : REM *****
3100 : REM Demande d'essai laser
3110 :
3120 : REM Effacement des indications ecran
3130 :
3140 : FOR I=1 TO 21
3150 : PRINT CHR$(11);
3160 : NEXT I
3170 : FOR I=1 TO 3
3180 : PRINT CHR$(14)
3190 : NEXT I
3200 : FOR I=1 TO 17
3210 : PRINT CHR$(10);
3220 : NEXT I
3230 : A$="F":GOSUB 1900
3235 : PRINT CHR$(10);
3240 :
3250 : REM Calculs de la sortie du laser
3260 :
3270 : DE=SCRN(XP*2+7,YP*2+3)-90 'DE=Sens du deplacement
3280 : X=YP-2:Y=XP-1
3290 : ON DE GOSUB 3400,3500,3600,3700
3300 : IF FF<>1 AND BO<>1 THEN GOTO 3290 'Boucle de recherche
3310 : GOSUB 3800:RETURN
3320 :
3330 : REM *****
3400 : REM Deplacement du laser vers la gauche
3410 :
3420 : Y=Y-1:IF Y<=0 THEN Y=0:FF=1:RETURN
3430 : IF X<>1 THEN IF T(X-1,Y-1)=1 THEN DE=3 'Vers le bas
3440 : IF X<>4 THEN IF T(X+1,Y-1)=1 THEN DE=2 'Vers le haut
3450 : IF T(X,Y-1)=1 THEN BO=1 'Absorbtion
3460 :
3470 : RETURN
3480 : REM *****
3500 : REM Deplacement du laser vers le haut
3510 :

```

```

3520 : X=X-1:IF X<=0 THEN X=0:FF=1:RETURN
3530 : IF Y<>1 THEN IF T(X-1,Y-1)=1 THEN DE=4 'Vers la droite
3540 : IF Y<>9 THEN IF T(X-1,Y+1)=1 THEN DE=1
3550 : IF T(X-1,Y)=1 THEN BO=1 'Absorbtion
3560 :
3570 : RETURN
3580 : REM *****
3600 : REM Deplacement du laser vers le bas
3610 :
3620 : X=X+1:IF X>=4 THEN X=5:FF=1:RETURN
3630 : IF Y<>1 THEN IF T(X+1,Y-1)=1 THEN DE=4 'Vers la droite
3640 : IF Y<>9 THEN IF T(X+1,Y+1)=1 THEN DE=1 'Vers la gauche
3650 : IF T(X+1,Y)=1 THEN BO=1 'Absorbtion
3660 :
3670 : RETURN
3680 : REM *****
3700 : REM Deplacement du laser vers la droite
3710 :
3720 : Y=Y+1:IF Y>=9 THEN Y=10:FF=1:RETURN
3730 : IF X<>1 THEN IF T(X-1,Y+1)=1 THEN DE=3 'Vers le bas
3740 : IF X<>4 THEN IF T(X+1,Y+1)=1 THEN DE=2 'Vers le haut
3750 : IF T(X,Y+1)=1 THEN BO=1 'Absorbtion
3760 :
3770 : RETURN
3780 : REM *****
3800 : REM Placement des reperes Entree/Sortie sur la grille
3810 :
3820 : IN=X+2:X=Y+1:Y=IN
3830 : A=XP*2+7:B=YP*2+3:C=X*2+7:D=Y*2+3
3840 :
3850 : IF YP=7 THEN B=B+1 ELSE B=B-1
3860 : IF Y=7 THEN D=D+1 ELSE D=D-1
3870 : IF XP=11 THEN A=A+1:B=B+1
3880 : IF X=11 THEN C=C+1:D=D+1
3885 : IF XP=1 THEN B=B+1:A=A-1
3886 : IF X=1 THEN D=D+1:C=C-1
3890 :
3900 : REM Affichage de l'entree
3910 :
3920 : LE=LE+1 'Lettre + 1
3925 : IF LE=27 THEN FIN=1:NF=1:RETURN

```

```

3930 :   PLOT A,B,CHR$(LE+64)
3940 :
3950 :   REM Affichage eventuel de la sortie
3960 :
3970 :   IF B0=1 THEN B0=1:FF=0:RETURN
3980 :   PLOT C,D,CHR$(LE+96)
3990 :
4000 :   FF=0 : RETURN
4010 :
4020 : REM #####
4100 : REM Placement d'un trou noir
4110 :
4115 :   TN=TN+1:PRINT CHR$(11);
4120 :   PRINT CHR$(27);"H";CHR$(27);"F   T=Tir ";CHR$(27);"H";CHR$(
(27);"A";
4130 :   PRINT"P=Placement";CHR$(27);"H";CHR$(27);"F"
4140 :
4150 :   FOR I=1 TO 20
4160 :     PRINT CHR$(11);
4170 :   NEXT I
4180 :   PRINT CHR$(17);:INPUT"           Ligne";LI
4190 :   INPUT "           Colonne";CO:PRINT CHR$(17);
4200 :
4210 :   PLOT CO*2+9,LI*2+7,"@";R(LI,CO)=1
4220 :
4230 :   PRINTCHR$(11);CHR$(14);CHR$(11);CHR$(14)
4240 :   FOR I=1 TO 9
4250 :     PRINT CHR$(10)
4260 :   NEXT I
4270 :   A$="F":GOSUB 1900:PRINT"   T=Tir , P=Placement , E=Effacem
ent"
4280 :
4290 :   GOSUB 4700
4300 :
4310 : RETURN
4320 : REM #####
4400 : REM Effacement d'un trou noir
4410 :
4420 :   PRINT CHR$(11);"   T=Tir , P=Placement";CHR$(27);"H";CHR$(
27);"A";
4430 :   PRINT "E=Effacement"
4440 :

```

```

4450 : FOR I=1 TO 20
4460 :   PRINT CHR$(11);
4470 : NEXT I
4480 : PRINT CHR$(17);:INPUT"          Ligne";LI
4490 : INPUT "          Colonne";CO:PRINT CHR$(17);
4500 :
4510 : PLOT CO&2+9,LI&2+7," ":R(LI,CO)=0
4520 :
4530 : PRINTCHR$(11);CHR$(14);CHR$(11);CHR$(14)
4540 : FOR I=1 TO 9
4550 :   PRINT CHR$(10)
4560 : NEXT I
4570 : A$="F":GOSUB 1900:PRINT"  T=Tir , P=Placement , E=Effacement"
ent"
4575 : GOSUB 4700
4580 :
4590 : RETURN
4600 : REM #####
4700 : REM Test de fin de partie
4710 :
4720 : FOR I=1 TO 4
4730 :   FOR J=1 TO 9
4740 :     IF T(I,J)=R(I,J) THEN W=W+1
4750 :   NEXT J
4760 : NEXT I
4770 : IF W=36 THEN FIN=1:ER=TN-3
4780 : W=0:RETURN
4790 : REM #####
5000 : REM Fin de la partie .
5010 :
5020 : CLS
5030 : IF ER=0 AND NF=0 THEN PRINT"Parcours sans faute.Bravo!!."
5040 : IF ER(>)0 THEN PRINTER;"Erreurs de parcours."
5050 : IF NF=1 THEN PRINT"Partie non finie !! Pas de chance ."
5060 : PRINT CHR$(17) 'Restitution du curseur
5070 :
5080 : REM #####

```

Grenouilles et crapauds

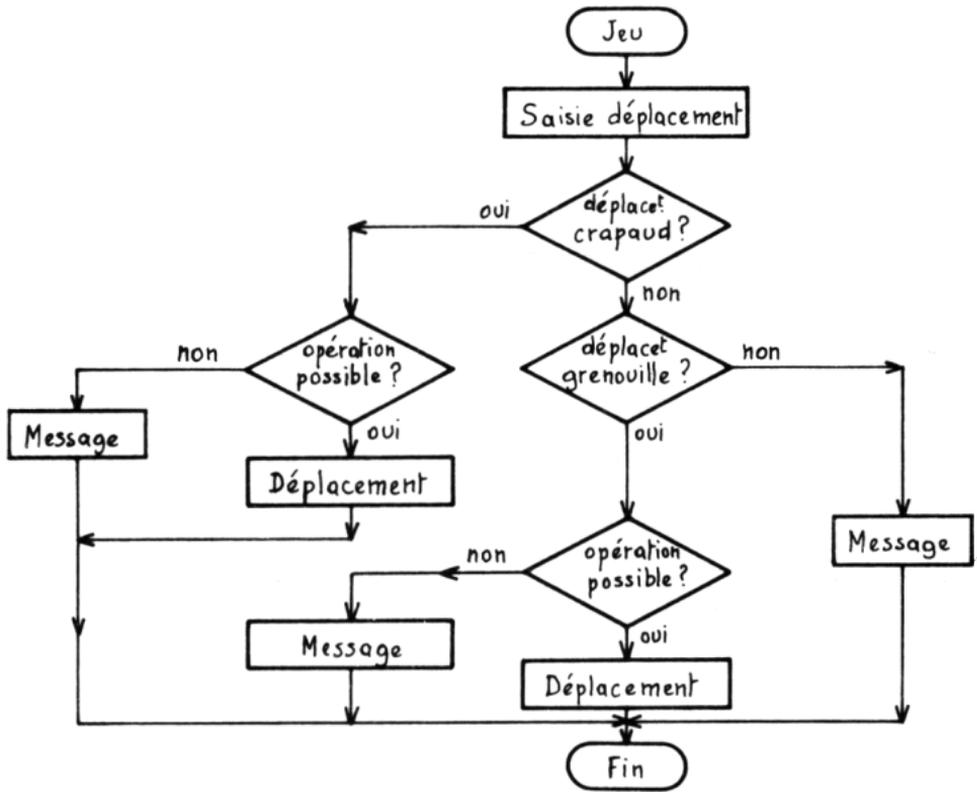
PRESENTATION

N crapauds se trouvent séparés de N grenouilles par une case vide. Les crapauds sont à gauche, les grenouilles à droite. Le but du jeu est d'amener tous les crapauds à droite, et toutes les grenouilles à gauche en respectant les 4 règles suivantes :

- . Les crapauds ne peuvent se déplacer que vers la droite
- . Les grenouilles ne peuvent se déplacer que vers la gauche
- . Un crapaud peut sauter par-dessus une grenouille si la case située à sa droite est libre
- . Une grenouille peut sauter par-dessus un crapaud si la case située à sa gauche est libre

ORGANIGRAMME DU JEU

Le jeu est initialisé par la saisie du nombre de crapauds. Le tableau de jeu est constitué et affiché, puis le jeu démarre.



LE PROGRAMME

Il se décompose en 3 pavés :

Initialisation

Lignes 1000 à 1800	
Module de commande	Lignes 1000 à 1050
Règle du jeu	Lignes 1100 à 1540
Initialisation du jeu	Lignes 1600 à 1800

Jeu

Lignes 1900 à 2970	
Module de commande	Lignes 1900 à 1950
Déplacement	Lignes 2000 à 2320
Affichage du jeu	Lignes 2400 à 2970

Fin du jeu

Lignes 3000 à 3040	
Commentaires sur le jeu	

LES VARIABLES UTILISEES

A\$	Code couleur de ligne
COUPS	Nombre de coups

DE\$	Code de déplacement
I	Index de boucles
JEU\$	Tableau de jeu
NC	Nombre de crapauds
NG	Nombre de grenouilles
NER	Nombre d'erreurs
PTR	Indicateur de déplacement

```

10 : REM Grenouilles et Crapauds
20 :
30 : GOSUB 1000 'Initialisation
40 : GOSUB 1900 'Traitement
50 : GOSUB 3000 'Fin
60 : END
70 : REM *****
1000 : REM Initialisation
1010 :
1020 : GOSUB 1100 'Regle du jeu
1030 : GOSUB 1600 'Dimension du jeu
1040 : RETURN
1050 : REM *****
1100 : REM Regle du jeu
1110 :
1120 : PAPER 0:INK 5
1130 : REM Titre
1140 : PRINT CHR$(12);CHR$(4)
1150 : PRINT CHR$(27);"N";CHR$(27);"A      GRENOUILLES ET CRAPAU
DS";
1160 : PRINT CHR$(4)
1170 : REM Objet
1180 :
1190 : PRINT:PRINT
1200 : A$="D":PRINT"OBJET :";

```

```

1210 : GOSUB 1500:PRINT"Jeu de reflexion"
1220 : REM Job
1230 :
1240 : PRINT
1250 : A$="F":PRINT"JOB  :";
1260 : GOSUB 1500:PRINT"Les crapauds sont definis par";
1270 : GOSUB 1500:PRINT"      la lettre C , les grenouilles";
1280 : GOSUB 1500:PRINT"      par la lettre G ."
1290 : GOSUB 1500:PRINT"      Le but du jeu est d'amener en";
1300 : GOSUB 1500:PRINT"      un minimum de coups les"
1310 : GOSUB 1500:PRINT"      crapauds a droite et les"
1320 : GOSUB 1500:PRINT"      grenouilles a gauche ."
1330 : GOSUB 1500:PRINT"      Restrictions : Un crapaud"
1340 : GOSUB 1500:PRINT"      peut se deplacer d'1 case "
1350 : GOSUB 1500:PRINT"      vers la droite si elle est "
1360 : GOSUB 1500:PRINT"      libre(DC).Il peut egalement"
1370 : GOSUB 1500:PRINT"      sauter par dessus une "
1380 : GOSUB 1500:PRINT"      grenouille si la case situee"
1390 : GOSUB 1500:PRINT"      a sa droite est libre(SC)."
1400 : GOSUB 1500:PRINT"      De meme pour une grenouille ,";
1410 : GOSUB 1500:PRINT"      mais vers la gauche(DG et SG)";
1420 : REM Debut
1430 :
1440 : PRINT
1450 : A$="C":PRINT"DEBUT :";
1460 : GOSUB 1500:PRINT"Pour commercer , appuyez sur"
1470 : GOSUB 1500:PRINT"      une touche quelconque ."
1480 : GET A$:RETURN
1490 : REM #####
1500 : REM Couleur d'une ligne
1510 :
1520 : PRINT CHR$(27);"H";CHR$(27);A$;
1530 : RETURN
1540 REM #####
1600 : REM Dimension du jeu
1610 :
1620 : CLS 'Effacement de l'ecran
1630 : PRINT:PRINT:PRINT:PRINT:PRINT

```

```

1640 : INPUT "Nombre de crapauds";NC
1650 : DIM JEU$(2*NC+1)
1660 : REM Constitution du tableau
1670 : FOR I=1 TO NC
1680 :   JEU$(I)="C"
1690 : NEXT I
1700 : JEU$(NC+1)=" "
1710 : FOR I=NC+2 TO NC*2+1
1720 :   JEU$(I)="G"
1730 : NEXT I
1740 : REM Affichage
1750 : PRINT:PRINT:PRINT
1760 : FOR I=1 TO NC*2+1
1770 :   PRINT JEU$(I);
1780 : NEXT I
1790 : RETURN
1800 : REM *****
1900 : REM Traitement
1910 :
1920 :   GOSUB 2000 'Deplacement
1930 :   GOSUB 2900 'Fin de partie
1940 :   IF NOFIN=1 THEN GOTO 1900
1950 : RETURN
2000 : REM Deplacement
2010 :
2020 :   INPUT " Deplacement";DE$:PTR=-1:COUPS=COUPS+1
2030 :   IF DE$(">DC" THEN GOTO 2090
2040 :   FOR I=1 TO NC*2
2050 :     IF JEU$(I)="C" THEN IF JEU$(I+1)=" " THEN PTR=I
2060 :   NEXT I
2070 :   IF PTR=-1 THEN GOSUB 2300:RETURN
2080 :   JEU$(PTR)=" ";JEU$(PTR+1)="C":GOSUB 2400:RETURN
2090 : IF DE$(">SC" THEN GOTO 2150
2100 :   FOR I=1 TO NC*2-1
2110 :     IF JEU$(I)="C" THEN IF JEU$(I+1)="G" THEN IF JEU$(I+2)=" " TH
EN PTR=I
2120 :   NEXT I
2130 :   IF PTR=-1 THEN GOSUB 2300:RETURN

```

```

2140 : JEU$(PTR)=" ":JEU$(PTR+2)="C":GOSUB2400:RETURN
2150 : IF DE$(>)"DG" THEN GOTO 2210
2160 : FOR I=NC*2+1 TO 2 STEP -1
2170 : IF JEU$(I)="G"THEN IF JEU$(I-1)=" "THEN PTR=I
2180 : NEXT I
2190 : IF PTR=-1 THEN GOSUB 2300:RETURN
2200 : JEU$(PTR)=" ":JEU$(PTR-1)="G":GOSUB 2400:RETURN
2210 : IF DE$(>)"SG" THEN GOTO 2300
2220 : FOR I=NC*2+1 TO 3 STEP -1
2230 : IF JEU$(I)="G"THEN IF JEU$(I-1)="C"THEN IF JEU$(I-2)=" "TH
EN PTR=I
2240 : NEXT I
2250 : IF PTR=-1 THEN GOSUB 2300:RETURN
2260 : JEU$(PTR)=" ":JEU$(PTR-2)="G":GOSUB 2400:RETURN
2300 : PRINT"Deplacement inaccepte"
2310 : NER=NER+1
2320 : RETURN
2330 : REM *****
2400 : REM Affichage du jeu
2410 :
2420 : FOR I=1 TO NC*2+1
2430 : PRINT JEU$(I);
2440 : NEXT I
2450 : RETURN
2900 : REM Fin de partie
2905 : NG=0 'RAZ Variable fin de jeu
2910 :
2920 : FOR I=1 TO NC
2930 : IF JEU$(I)="G" THEN NG=NG+1
2940 : NEXT I
2950 : IF NG=NC THEN IF JEU$(NC+1)=" " THEN NOFIN=0 ELSE NOFIN=1
2960 : RETURN
2970 : REM *****
3000 : REM Fin de partie : SCORE
3010 :
3020 : PRINT:PRINT"Partie terminee en ";COUPS;"coups ."
3030 : PRINT"avec ";NER;"erreur(s) ."
3040 : RETURN

```

Conjugaison des verbes

PRESENTATION

Même si la conjugaison des verbes du premier groupe n'a plus de secrets pour vous, vous feriez peut-être bien de jeter un coup d'oeil à ce didacticiel (logiciel à usage didactique).

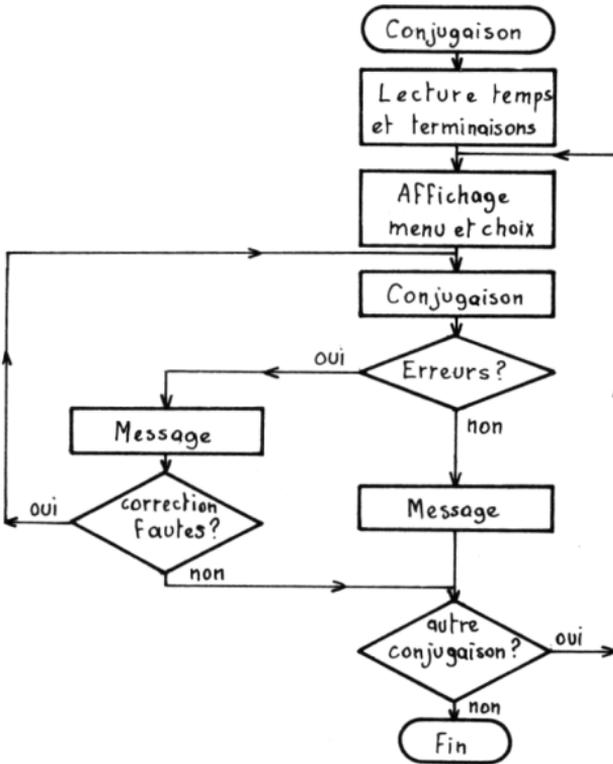
Après avoir entré le temps de conjugaison (Présent, Imparfait, Passé Simple, Futur simple, Plus-que-parfait, Passé antérieur, Futur antérieur, ou Passé composé), ORIC vous demande le verbe à conjuguer à l'infinif.

Vous conjuguez le verbe choisi, puis, ORIC vous indique les fautes s'il y a lieu (il vous invite alors à recommencer).

ORGANIGRAMME DU JEU

Le programme a la structure A.

la conjugaison a la structure suivante :



LE PROGRAMME

Il se décompose en 3 pavés :

Présentation	Lignes 1000 à 1550	
Initialisation et conjugaison	Lignes 2000 à 2995	
	Initialisation	Lignes 2000 à 2220
	Affichage menu	Lignes 2230 à 2400
	Verbe à conjuguer	Lignes 2410 à 2500
	Conjugaison	Lignes 2510 à 2680
	Recherche erreurs	Lignes 2700 à 2790
	Affich. résultats	Lignes 2800 à 2995
Fin de conjugaison	Lignes 3000 à 3060	

LES VARIABLES UTILISEES

A\$ Code couleur de ligne

C	Temps de la conjugaison
FX	Tableau des fautes
I, J	Variables de boucles
NF	Nombre de fautes
R\$	Réponse à "vous corrigez (O/N) ?"
RE\$	Conjugaison du verbe à une des personnes
T\$	Tableau des temps proposés
TE\$	Terminaisons de chaque temps
V\$	Verbe à conjuguer

```

10 : REM Conjugaison des verbes du 1er groupe
20 :
30 : GOSUB 1000 'Presentation du programme
40 : GOSUB 2000 'Conjugaison
50 : GOSUB 3000 'Fin de conjugaison
60 :
70 : END
80 : REM #####
1000 : REM Presentation
1010 :
1020 : PAPER 0 : INK 5 : PRINT CHR$(17) : DIM TE$(48)
1030 :
1040 : REM Titre
1050 :
1060 : PRINT CHR$(12)
1070 : PRINT CHR$(4);CHR$(27);"N";
1080 : PRINT CHR$(27);"A CONJUGAISON DU PREMIER GROUPE";CHR$(
4)
1090 :

```

```

1100 : REM Objet
1110 :
1120 : PRINT:PRINT:PRINT:PRINT:PRINT
1130 : A$="D" 'Code couleur = Bleu
1140 : PRINT"OBJET :";
1150 : GOSUB 1500:PRINT"Test de connaissances."
1160 :
1170 : REM Job
1180 :
1190 : PRINT:PRINT:PRINT:PRINT
1200 : A$="F" 'Code couleur = Cyan
1210 : PRINT"JOB  :";
1220 : GOSUB 1500:PRINT"Ce didacticiel permet de"
1230 : GOSUB 1500:PRINT"      tester vos connaissances"
1240 : GOSUB 1500:PRINT"      actuelles dans la conjugaison"
1250 : GOSUB 1500:PRINT"      des verbes du 1er groupe."
1260 : GOSUB 1500:PRINT"      En cas de faute,ORIC vous"
1270 : GOSUB 1500:PRINT"      corrigera."
1280 :
1290 : REM Debut
1300 :
1310 : PRINT:PRINT:PRINT:PRINT
1320 : A$="C" 'Code couleur = Jaune
1330 : PRINT"Debut :";
1340 : GOSUB 1500:PRINT"Pour commencer,appuyez sur"
1350 : GOSUB 1500:PRINT"      une touche quelconque."
1360 :
1370 : GET A$ 'Attente de l'appui sur une touche
1380 :
1390 : RETURN
1400 : REM #####
1500 : REM Couleur d'une ligne
1510 :
1520 : PRINT CHR$(27);"H";CHR$(27);A$;
1530 :
1540 : RETURN

```

```

1550 : REM *****
2000 : REM Conjugaison
2010 :
2020 :   CLS : INK 6 : PRINT CHR$(20)
2030 :
2040 :   REM Lecture des donnees
2050 :
2070 :     FOR I=1 TO 48
2080 :       READ A$:TE$(I)=A$
2090 :     NEXT I
2100 DATA "e","es","e","ons","ez","ent"
2110 DATA "ais","ais","ait","ions","iez","aient"
2120 DATA "ai","as","a","ames","ates","erent"
2130 DATA "erai","eras","era","erons","erez","eront"
2140 DATA "j'avais","tu avais","il avait"
2150 DATA "nous avions","vous aviez","ils avaient"
2160 DATA "j'eus","tu eus","il eut"
2170 DATA "nous eumes","vous eutes","ils eurent"
2180 DATA "j'aurai","tu auras","il aura"
2190 DATA "nous aurons","vous aurez","ils auront"
2200 DATA "j'ai","tu as","il a"
2210 DATA "nous avons","vous avez","ils ont"
2211 :
2212 :   FOR I=1 TO 6
2213 :     READ T$(I)
2214 :   NEXT
2215 DATA "Present","Imparfait","Passe simple","Futur simple"
2216 DATA "Plus que parfait","Passe anterieur","Futur anterieur"
2217 DATA "Passe compose"
2220 :
2230 :   REM Menu"
2240 :
2250 :     CLS:PRINT CHR$(27);"L";CHR$(27);"A";
2260 :     PRINT"CONJUGAISON DES VERBES DU 1er GROUPE"
2270 :     PRINT:PRINT:PRINT
2280 :     PRINT"Quel temps desirez vous conjuguer ?"
2290 :     PRINT:PRINT

```

```

2300 : PRINT:PRINT" - (1) Present"
2310 : PRINT:PRINT" - (2) Imparfait"
2320 : PRINT:PRINT" - (3) Passe simple"
2330 : PRINT:PRINT" - (4) Futur simple"
2340 : PRINT:PRINT" - (5) Plus que parfait"
2350 : PRINT:PRINT" - (6) Passe anterieur"
2360 : PRINT:PRINT" - (7) Futur anterieur"
2370 : PRINT:PRINT" - (8) Passe compose"
2380 : PRINT CHR$(17)
2390 : PRINT:INPUT"Entrez votre choix";C
2395 : IF C>8 OR C<1 THEN 2230
2400 :
2410 : REM Entree du verbe a conjuguer
2420 :
2430 : CLS
2440 : A$="B":GOSUB 1500
2450 : PRINT"Entrez le verbe a conjuguer : "
2460 : PRINT:INPUT V$
2470 : IF RIGHT$(V$,2) = "er" THEN GOTO 2500
2480 : PRINT:PRINT:A$="A":GOSUB 1500
2490 : PRINT"Ce n'est par un verbe du 1er groupe !!":GOTO 2440
2500 :
2510 : REM Conjugaison
2520 :
2580 : CLS:A$="A":GOSUB 1500
2590 : PRINT" Conjugaison de ";V$
2600 : GOSUB 1500:PRINT" Temps : ";T$(C)
2610 :
2620 : PRINT:PRINT:PRINT
2630 : FOR I=1 TO 6
2640 : IF I<4 THEN PRINT " "I"e Personne Sing ";
2650 : IF I>3 THEN PRINT " "I-3"e Personne Plur ";
2660 : INPUT RE$(I)
2670 : NEXT
2680 :
2700 : REM Recherche des eventuelles erreurs ...
2710 :

```

```

2720 :   FOR I=1 TO 6
2730 :       J=I+(C-1)*6:LT=LEN(TE$(J))
2740 :       IF C>4 THEN GOTO 2770
2750 :       IF RIGHT$(RE$(I),LT)=TE$(J) THEN FX(I)=0 ELSE FX(I)=1
2760 :       GOTO 2790 'Saut au next
2770 :       IF LEFT$(RE$(I),LT)=TE$(J) THEN A=1 ELSE A=0
2780 :       IF A=1 AND RIGHT$(RE$(I),1)="e" THEN FX(I)=0 ELSE FX(I)
2790 :       )=1
2800 :       NEXT I
2810 :       REM Affichage des resultats
2820 :       PRINT:PRINT:NF=0
2830 :       FOR I=1 TO 6
2840 :           IF FX(I)=1 THEN NF=NF+1
2850 :       NEXT I
2860 :       IF NF=0 THEN PRINT "Bravo,0 fautes.":GOTO 2950
2870 :       FOR I=1 TO 6
2880 :           IF FX(I)=1 THEN PRINT "Faute sur la ";
2890 :           IF FX(I)=1 AND I<4 THEN PRINT I;"e Pers Sing"
2900 :           IF FX(I)=1 AND I>3 THEN PRINT I-3;"e Pers Plur"
2910 :       NEXT I
2920 :       PRINT:PRINT
2930 :       IF NF=0 THEN GOTO 3000
2940 :       INPUT"Vous corrigez (O/N) ";R$
2950 :       IF R$="o" THEN GOTO 2580
2960 :       INPUT"Une autre conjugaison (O/N) ";R$
2970 :       IF R$="o" THEN GOTO 2240
2980 :       RETURN
2990 :
3000 : REM *****
3010 : REM Fin de la conjugaison
3020 : CLS
3030 : PLOT 10,10,"A la prochaine fois ..."
3040 : PRINT CHR$(17)
3050 :
3060 : RETURN

```

TROISIÈME PARTIE
GRAPHIQUES
ET MATHÉMATIQUES

Tracé de courbes

$y = F(x)$

PRESENTATION

Pour pouvoir représenter graphiquement une courbe à une variable, il faut connaître son domaine d'étude, exclure les points aberrants, trouver quelques tangentes, points d'inflexion et asymptotes. Bref, beaucoup d'opérations qui prennent du temps.

Le programme de tracé de courbes effectue pour vous une grande partie de ce travail. En effet, il lui suffit de connaître le domaine d'étude pour effectuer un tracé. Il calcule les échelles et affiche la courbe à l'écran, en haute résolution. Le domaine d'étude devra exclure les points aberrants.

ORGANIGRAMME DE L'UTILITAIRE

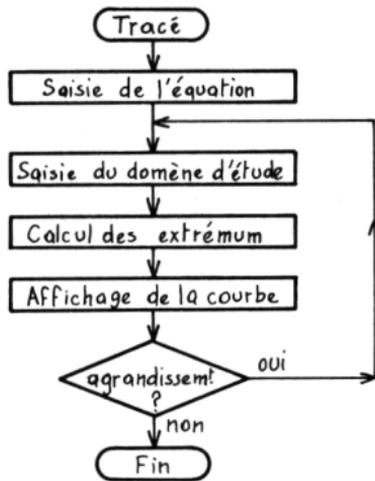
Le programme a la structure A.

L'équation est saisie en entrant une ligne de programme BASIC. Puis, le domaine d'étude est entré.

Les valeurs extrémales sont calculées, puis la courbe est tracée.

L'utilisateur peut obtenir un "agrandissement" de la courbe en restreignant son domaine d'étude.

L'organigramme suivant décrit ces opérations :



LE PROGRAMME

Il se décompose en 2 pavés :

Présentation

Lignes 1000 à 1550
Présentation du programme

Saisie Equation et tracé

Lignes 2000 à 2730	
Saisie équation	Lignes 2030 à 2160
Saisie domaine	Lignes 2210 à 2310
Calcul du pas	Lignes 2340 à 2580
Tracé	Lignes 2610 à 2700

LES VARIABLES UTILISEES

A\$	Code couleur de ligne
R\$	Réponse à la question "Agrandissement ?"
X	Abscisse courbe
X1	Borne minimum du domaine d'étude
X2	Borne maximum du domaine d'étude
XPAS	Pas en X
YPAS	Pas en Y

```

10 : REM Trace de courbes y = F (x)
20 :
30 : GOSUB 1000 'Presentation
40 : GOSUB 2000 'Saisie de l'equation et trace
50 :
60 : REM Fin du programme
70 : REM #####
1000 : REM Presentation
1010 :
1020 : PAPER 0 : INK 5 : PRINT CHR$(17)
1030 :
1040 : REM Titre
1050 :
1060 : PRINT CHR$(12)
1070 : PRINT CHR$(4);CHR$(27);"N";
1080 : PRINT CHR$(27);"A TRACE DE COURBES Y = F (X)";CHR$(4
)
1090 :
1100 : REM Objet
1110 :
1120 : PRINT:PRINT:PRINT:PRINT:PRINT
1130 : A$="D" 'Code couleur = Bleu
1140 : PRINT"OBJET :";
1150 : GOSUB 1500:PRINT"Representation graphique de"
1160 : GOSUB 1500:PRINT" courbes a une variable."
1170 :
1180 : REM Job
1190 :
1200 : PRINT:PRINT:PRINT
1210 : A$="F" 'Code couleur = Cyan
1220 : PRINT"JOB :";
1230 : GOSUB 1500:PRINT"Apres avoir entre l'equation"
1240 : GOSUB 1500:PRINT" et le domaine d'etude de"
1250 : GOSUB 1500:PRINT" la courbe,ORIC effectuera son"
1260 : GOSUB 1500:PRINT" trace en haute resolution."
1270 : PRINT:GOSUB 1500:PRINT" Il sera alors possible"
1280 : GOSUB 1500:PRINT" d'obtenir un 'agrandissement'"
1290 : GOSUB 1500:PRINT" de la courbe en restreignant"
1300 : GOSUB 1500:PRINT" son domaine d'etude."
1310 :
1320 : REM Debut
1330 :
1340 : PRINT:PRINT:PRINT
1350 : A$="C" 'Code couleur = Jaune
1360 : PRINT"DEBUT :";

```

```

1370 :   GOSUB 1500:PRINT"Pour commencer , appuyez sur"
1380 :   GOSUB 1500:PRINT"      une touche quelconque."
1390 :   GET A$ 'Attente de l'appui sur une touche
1400 :
1410 : RETURN
1420 : REM #####
1500 : REM Couleur d'une ligne
1510 :
1520 :   PRINT CHR$(27);"H";CHR$(27);A$;
1530 :
1540 : RETURN
1550 : REM #####
2000 : REM Saisie de l'equation et trace
2010 :
2020 :   CLS : INK 6 : PRINT CHR$(17)
2025 :
2030 :   REM Saisie de l'equation
2040 :
2050 :   PRINT CHR$(27);"L";CHR$(27);"A      SAISIE DE L'EQUAT
ION"
2060 :   PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
2070 :   PRINT"   Vous allez entrer l'equation"
2080 :   PRINT"   sous la forme :"
2090 :   PRINT
2100 :   PRINT"       2230 DEF FNA(X) = SIN(X)"
2110 :   PRINT:PRINT:PRINT
2120 :   PRINT"   Vous taperez ensuite :"
2130 :   PRINT:PRINT"       RUN 2200"
2140 :   PRINT:PRINT
2150 :
2160 :   STOP
2200 : REM L'equation est saisie
2210 : REM Saisie du domaine de definition
2220 :
2230 DEF FNA(X) = SIN(X)
2235 :   CLS
2240 :   PRINT CHR$(27);"L";CHR$(27);"A   SAISIE DU DOMAINE DE DE
FINITION"
2250 :   PRINT:PRINT:PRINT:PRINT:PRINT
2260 :   PRINT"   Vous allez entrer les valeurs"
2270 :   PRINT"   minimale Xmin et maximale Xmax"
2280 :   PRINT"   du domaine d'etude :"
2290 :   PRINT:PRINT:PRINT
2300 :   INPUT"       Xmin = ";X1
2310 :   PRINT:INPUT"       Xmax = ";X2
2320 :

```

```

2330 : PRINT:PRINT:PRINT:PRINT" Un moment s'il vous plait."
2340 : PRINT" Je suis en train de calculer."
2350 : PRINT CHR$(17)
2400 : REM Le domaine d'etude est entre
2410 :
2420 : XPAS=(X2-X1)/230 'Pas en ordonnee
2430 :
2440 : REM Recherche du pas en coordonnee
2450 :
2460 : REM Initialisation des variables de pas
2470 :
2480 : Y1 = 1E33 : Y2 = -1E33
2490 : REM Recherche du pas
2500 :
2510 : FOR X=X1 TO X2 STEP XPAS
2520 : A=FNA(X)
2530 : IF A<Y1 THEN Y1 = A : X3 = X
2540 : IF A>Y2 THEN Y2 = A : X4 = X
2550 : NEXT X
2560 :
2570 : YPAS = (Y1-Y2) / 180
2580 :
2590 : REM Trace
2600 :
2610 : HIRES
2620 : FOR X=X1 TO X2 STEP XPAS
2630 : Y=FNA(X)
2640 : CURSET INT((1/XPAS)*(X-X1)),INT((1/YPAS)*(Y-Y2)),1
2650 : NEXT X
2660 :
2670 : PRINT"Domaine d'etude : [ ";X1;" ";X2;"]"
2675 : GET A$
2680 : PRINT"Y max pour X="INT(X*100)/100"Y min pour X="INT(X3
*100)/100
2685 : GET A$
2690 : PRINT"Ymax = ";INT(Y2*100)/100" Ymin = ";INT(Y1*100)/10
0
2695 : GET A$
2700 : INPUT"Un agrandissement (D/N) ";R$
2710 : IF R$="D" THEN LORES 0 : GOTO 2200
2720 :
2730 : LORES 0 : END

```

Tracé de courbes $Y(t), X(t)$

PRESENTATION

Point n'est besoin d'être un mathématicien acharné pour se servir de ce programme. Le tracé des courbes $Y(t), X(t)$ ressemble beaucoup au tracé des courbes $Y = F(x)$ à ceci près qu'ordonnée et abscisse sont toutes deux fonctions de la variable t (souvent le temps).

Les courbes $Y = F(x)$ et $Y(t), X(t)$ s'étudient de façon similaire. La variable t fluctue dans un domaine de définition (ou d'étude) qui doit tenir compte des points aberrants de la fonction X et de la fonction Y .

Là aussi, il faut chercher quelques tangentes, points d'inflexions et asymptotes pour obtenir un tracé correct. ORIC va se charger des calculs fastidieux et vous demandera seulement le domaine d'étude. Il calculera les échelles en X et en Y et affichera la courbe à l'écran, en haute résolution.

ORGANIGRAMME DU TRACE

Le programme a la structure A.

Ici, deux équations sont saisies $X = f(t)$ et $Y = g(t)$.

Le domaine d'étude est entré.

Les valeurs extrémales sont calculées et déterminent les échelles en X et en Y . La courbe est tracée. L'utilisateur peut obtenir un agrandissement de la courbe en restreignant son domaine d'étude.

L'organigramme est le même que celui donné pour le tracé des courbes $Y = F(x)$.

LE PROGRAMME

Identique au tracé de courbes $Y = F(x)$.

LES VARIABLES UTILISEES

Identiques au tracé de courbes $Y = F(x)$, avec en plus...

TPAS	Pas de progression en t
Y1, Y2	Bornes minimum et maximum de la fonction Y (t).

Remarque :

Vous pouvez essayer les couples d'équations suivantes qui donnent d'assez beaux résultats avec 150 points et un domaine d'étude compris entre 0 et 6.5.

$\left\{ \begin{array}{l} (1+\cos(t)) * \sin(t) \\ (1+\cos(t)) * \cos(t) \end{array} \right.$	$\left\{ \begin{array}{l} (1-\sin(t)) * \cos(t) \\ (1+\cos(t)) * \cos(t) \end{array} \right.$
$\left\{ \begin{array}{l} (1+\cos(t)) * \cos(t) \\ (1+\cos(t)) * \sin(t) \end{array} \right.$	$\left\{ \begin{array}{l} (1-\cos(2*t)) * \sin(t) \\ (1-\cos(t)) * \cos(t) \end{array} \right.$
$\left\{ \begin{array}{l} (1+\cos(2*t)) * \sin(2*t) \\ (1+\cos(t)) * \cos(t) \end{array} \right.$	$\left\{ \begin{array}{l} (1+\cos(2*t)) * \cos(2*t) \\ (1+\sin(t/2)) * \sin(t) \end{array} \right.$

```

10 : REM Trace de courbes Y(t),X(t)
20 :
30 : GOSUB 1000 'Presentation
40 : GOSUB 2000 'Saisie de l'equation et trace
50 :
60 : REM Fin du programme
70 : REM #####
1000 : REM Presentation
1010 :
1020 : PAPER 0 : INK 5 : PRINT CHR$(17)
1030 :

```

```

1040 : REM Titre
1050 :
1060 : PRINT CHR$(12)
1070 : PRINT CHR$(4);CHR$(27);"N";
1080 : PRINT CHR$(27);"A TRACE DE COURBES Y(t),X(t)";CHR$(4
)
1090 :
1100 : REM Objet
1110 :
1120 : PRINT:PRINT:PRINT:PRINT:PRINT
1130 : A$="D" 'Code couleur = Bleu
1140 : PRINT"OBJET :";
1150 : GOSUB 1500:PRINT"Representation graphique de"
1160 : GOSUB 1500:PRINT" courbes Y(t),X(t) ."
1170 :
1180 : REM Job
1190 :
1200 : PRINT:PRINT:PRINT
1210 : A$="F" 'Code couleur = Cyan
1220 : PRINT"JOB :";
1230 : GOSUB 1500:PRINT"Apres avoir entre les"
1240 : GOSUB 1500:PRINT" equations et le domaine d'etude"
;
1250 : GOSUB 1500:PRINT" de la courbe,ORIC effectuera "
1260 : GOSUB 1500:PRINT" son trace en haute resolution."
1270 : PRINT:GOSUB 1500:PRINT" Il sera alors possible"
1280 : GOSUB 1500:PRINT" d'obtenir un 'agrandissement'"
1290 : GOSUB 1500:PRINT" de la courbe en restreignant"
1300 : GOSUB 1500:PRINT" son domaine d'etude."
1310 :
1320 : REM Debut
1330 :
1340 : PRINT:PRINT:PRINT
1350 : A$="C" 'Code couleur = Jaune
1360 : PRINT"DEBUT :";
1370 : GOSUB 1500:PRINT"Pour commencer , appuyez sur"
1380 : GOSUB 1500:PRINT" une touche quelconque."
1390 : GET A$ 'Attente de l'appui sur une touche
1400 :
1410 : RETURN
1420 : REM #####
1500 : REM Couleur d'une ligne
1510 :
1520 : PRINT CHR$(27);"H";CHR$(27);A$;
1530 :
1540 : RETURN

```

```

1550 : REM *****
2000 : REM Saisie des equations et trace
2010 :
2020 :   CLS : INK 6 : PRINT CHR$(17)
2025 :
2030 :   REM Saisie des equations
2040 :
2050 :     PRINT CHR$(27);"L";CHR$(27);"A           SAISIE DES EQUATI
DNS"
2060 :     PRINT:PRINT:PRINT:PRINT
2070 :     PRINT"   Vous allez entrer les equations"
2080 :     PRINT"   sous la forme :"
2090 :     PRINT
2100 :     GOSUB 1500:PRINT"       2230 DEF FNA(T) = SIN(T) et"
2101 :     GOSUB 1500:PRINT"       2231 DEF FNB(T) = COS(T)"
2102 :     PRINT:PRINT"   respectivement pour les equations"
2103 :     PRINT"       X(t) et Y(t)."

```

```

2420 : REM Calcul du pas en T
2430 :
2440 :   TPAS = (T2-T1) / 1000
2450 :
2460 : REM Recherche du pas en X et en Y
2470 :
2480 : REM Initialisation des variables
2490 :   X1=1E33 : X2=-1E33 : Y1=1E33 : Y2=-1E33
2500 :
2510 : REM Recherche des pas
2520 :
2530 :   FOR T=T1 TO T2 STEP TPAS
2540 :     A=FNA(T) : B=FNB(T)
2550 :     IF A<X1 THEN X1=A : X3=T
2560 :     IF A>X2 THEN X2=A : X4=T
2570 :     IF B<Y1 THEN Y1=B : Y3=T
2580 :     IF B>Y2 THEN Y2=B : Y4=T
2590 :   NEXT T
2600 :
2610 : REM Calcul des pas
2620 :
2630 :   XPAS = (X1-X2) / 235 : YPAS = (Y1-Y2) / 190
2999 : REM *****
3000 : REM Trace de la courbe
3005 :
3010 :   HIRES
3020 :   FOR T=T1 TO T2 STEP TPAS
3030 :     X=FNA(T)
3031 :     Y=FNB(T)
3040 :     CURSET INT((1/XPAS)*(X1-X)),INT((1/YPAS)*(Y-Y2)),1
3050 :   NEXT T
3060 :
3070 :   PRINT"Domaine d'etude : [ ";X1;",";X2;"]"
3075 :   GET A$
3080 :   PRINT"Y max pour X="INT(X4*100)/100"Y min pour X="INT(X3
#100)/100
3085 :   GET A$
3090 :   PRINT"Ymax = ";INT(Y2*100)/100" Ymin = ";INT(Y1*100)/10
0
3095 :   GET A$
3100 :   INPUT"Un agrandissement (0/N) ";R$
3110 :   IF R$="0" THEN LORES 0 : GOTO 2200
3120 :
3130 : LORES 0 : END

```

Tracé de courbes $Y(t), X(t)$ imbriquées

PRESENTATION

Ce programme est la suite logique du tracé de courbes $Y(t), X(t)$, et n'a qu'un but esthétique. Effectivement, il consiste à afficher sur l'écran une série de courbes paramétrées de type cardioïde imbriquées, ce qui a pour but de donner une "profondeur" à la courbe.

Seulement quelques lignes ont été changées :

- . Dans la saisie du domaine d'étude... Lignes 2200 à 2630
- . Dans le tracé des courbes à l'écran... Lignes 3000 à 3130

L'organigramme du tracé est identique à l'organigramme précédent "Tracé de courbes $Y(t), X(t)$ " et ne sera donc pas repris.

Seules les variables NC et NP sont supplémentaires. Elles désignent (respectivement) le nombre de cardioïdes imbriquées, et le nombre de points par cardioïde.

Le programme "Tracé de courbes $Y(t), X(t)$ imbriquées" n'est pas donné ici. En effet, il est très similaire au programme "Tracé de courbes $T(t), X(t)$ ". Les lignes, ci-contre, décrivent les modifications à apporter au second pour réaliser le premier.

```
15 : REM Imbriquées
2280 : PRINT "du domaine d'étude, puis le nombre de";
2281 : PRINT "cardioides imbriquées NC, et le"
2282 : PRINT "nombre de points par cartioides NP."
2290 : PRINT:PRINT
2311 : PRINT:INPUT"          NC = ";NC
2312 : PRINT:INPUT"          NP = ";NP
2440 : TPAS=(T2-T1)/NP
2540 : A=FNA(T)*NC:B=FNB(T)*NC
305 : FOR K=1 TO NC
3030 : X=FNA(T)*K
303 : Y=FNB(T)*K
3055 : NEXT K
```

Snoopy

PRESENTATION

Ce programme permet d'afficher SNOOPY sur l'écran, en haute résolution.

ORIC possède une mémoire d'écran qui est la réplique exacte de l'écran. Pour afficher un dessin en haute résolution, plusieurs méthodes peuvent être employées :

- . Un certain nombre de caractères basse résolution sont redéfinis afin de constituer le motif désiré.
- . Un programme BASIC se charge de transférer l'image à afficher de la mémoire vers la mémoire d'écran.
- . Comme ci-dessus, mais le programme est écrit en assembleur.

La première solution est à éliminer d'office car le nombre de caractères LORES 0 et LORES 1 est limité.

La deuxième solution est séduisante mais pas assez rapide (utilisation de BASIC).

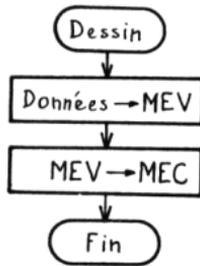
La troisième solution est celle développée ici. Elle offre un gros avantage : sa rapidité d'exécution...

ORGANIGRAMME DU DESSIN

Deux actions essentielles :

- . Le transfert des données "BASIC" vers la mémoire vive d'ORIC.
- . Le transfert de la mémoire vive vers la mémoire d'écran.

D'où l'organigramme du dessin :



LE PROGRAMME

Il se décompose en 2 pavés :

Transfert Basic vers Mémoire vive	Lignes 2000 à 2170	
	Informations à l'écran	Lignes 2000 à 2016
	Blanchiment MEV	Lignes 2020 à 2060
	Transfert	Lignes 2070 à 2170
Transfert MEV vers MEC	Lignes 2200 à 2790	
	Ecriture ASM	Lignes 2220 à 2290
	Message à l'écran	Lignes 2310 à 2350
	Transfert	Lignes 2400 à 2450
	Données SNOOPY	Lignes 2500 à 2580
	Données transfert	Lignes 2600 à 2790

LES VARIABLES UTILISEES

. VARIABLES BASIC :

A, B	Lecture des données
A\$	Attente de l'appui sur une touche
I, J	Index de boucles

. VARIABLES MEMOIRE :

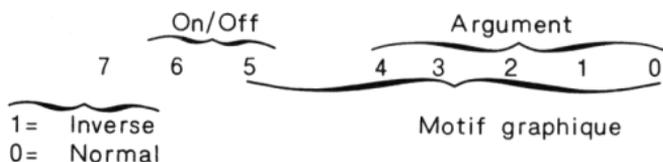
§ 08	Poids faible de début de mémoire vive
§ 09	Poids fort de début de mémoire vive
§ 0A	Poids faible de début de mémoire d'écran
§ 0B	Poids fort de début de mémoire d'écran

Avant de conclure sur ce programme, donnons la méthode d'affichage de dessins en mode HIRES :

Le mode HIRES étant validé, la mémoire d'écran peut être considérée

comme un tableau de 40 colonnes et de 199 lignes.

La mémoire d'écran est adressée entre # A 000 et # BFE 0. Un octet de la mémoire d'écran se décompose de la manière suivante :



Chaque octet envoyé à l'écran peut être :

- . Un motif graphique, ou
- . Un argument de contrôle d'attributs (Couleur, Clignotement...).

Les bits 5 et 6 font la différence entre motif graphique et argument

- . Ils sont tous les deux nuls, alors les bits 0 à 4 représentent un argument de contrôle d'attributs.
- . Un des deux n'est pas nul, alors les bits 0 à 5 représentent un motif graphique codé sur 6 points (car 6 bits).

Si un argument est positionné, alors il continue jusqu'à la fin de la ligne, sauf s'il est remis à 0 par un autre argument.

```

1000 : REM SNOOPY
1010 :
1020 : GOSUB 2000 'Transfert BASIC->MEV
1030 : GOSUB 2200 'Transfert MEV->MEC
1040 :
1050 : END
1060 : REM *****
2000 : REM Transfert Donnees -> MEV
2010 :
2011 : REM Informations
2012 :

```

```
2013 : PAPER 0:INK 6:CLS:PRINT CHR$(17)
2014 : PLOT0,12,2:PLOT1,12,12 'Vert clignotant
2015 : PLOT2,12,"Transfert BASIC -> MEV en cours ..."
2016 :
2020 : REM Blanchiment MEV
2025 :
2030 : FOR I=0 TO #500
2040 :     POKE #3200+I,64
2050 : NEXT
2060 :
2070 : REM Transfert
2080 :
2090 : FOR I=1 TO 30
2100 :     FOR J=1 TO 4
2110 :         READ A:B=#3214+J-1+((I-1)*40)
2120 :         POKE B,A
2130 :     NEXT J
2140 : NEXT I
2150 :
2160 : RETURN
2170 : REM *****
2200 : REM Transfert MEV -> MEC
2210 :
2220 : REM Ecriture du programme ASM
2230 :
2240 :     FOR I=0 TO 32
2250 :         READ A
2260 :         POKE#3100+I,A
2270 :     NEXT I
2280 :
2290 : REM Transfert termine
2300 :
2310 :     HIRES
2320 :     PRINT"Appuyez sur une touche pour"
2330 :     PRINT"declencher le transfert MEV->MEC"
2340 :     GET A$ 'Attente appui sur une touche
2350 :
2360 : REM *****
2400 : REM Transfert MEV -> MEC
```

```

2410 :
2420 :     CALL #3100
2430 :
2440 : RETURN
2450 : REM #####
2500 : REM Snoopy
2510 :
2520 DATA 64,64,78,64,64,64,33,32,64,65,64,32,64,66,64,80,65,63,64,
72
2530 DATA 68,64,80,72,72,64,64,72,72,64,65,56,80,56,75,56,80,80,83,
56
2540 DATA 80,64,83,56,72,49,35,48,79,60,71,80,64,90,83,64,64,90,83,
64
2550 DATA 64,84,48,64,64,68,80,64,64,72,80,64,64,72,80,64,64,72,72,
64
2560 DATA 64,72,66,64,64,72,65,64,64,72,83,64,64,68,83,64,64,68,83,
48
2570 DATA 64,48,79,48,67,48,83,56,66,79,67,68,67,65,63,52,67,63,32,
56
2580 :
2600 : REM Programme ASM MEV->MEC
2610 :
2620 DATA #A9,#00  'LDA #00
2630 DATA #85,#08  'STA $08
2640 DATA #85,#0A  'STA $0A
2650 DATA #A9,#32  'LDA #$32
2660 DATA #85,#09  'STA $09
2670 DATA #A9,#AF  'LDA #$AF
2680 DATA #85,#0B  'STA $0B
2690 DATA #A2,#05  'LDX #$05
2700 DATA #A0,#00  'LDY #$00
2710 DATA #B1,#08  'LDA ($08),Y
2720 DATA #91,#0A  'STA ($0A),Y
2730 DATA #8B      'DEY
2740 DATA #D0,#F9  'BNE $3104
2750 DATA #E6,#09  'INC $09
2760 DATA #E6,#0B  'INC $0B
2770 DATA #CA      'DEX
2780 DATA #D0,#F0  'BNE $3102
2790 DATA #60      'RTS

```

Dessin animé

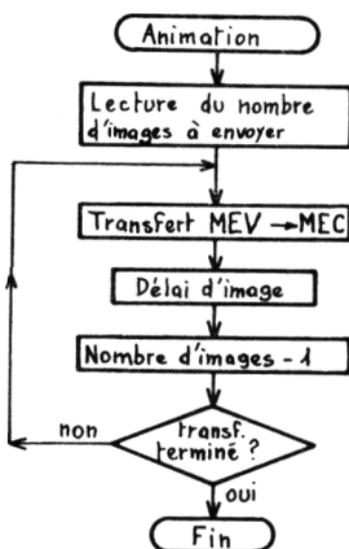
PRESENTATION

Le logiciel proposé ici s'inspire largement du transfert MEV → MEC précédent. Il est écrit en assembleur, et permet le transfert de N pages graphiques vers la mémoire d'écran. Chaque image est séparée d'un "délai image" et a une longueur de #4FF octets. La première image est située en #3200.

Le logiciel étudié précédemment permet le transfert d'une image graphique de la mémoire vers la mémoire d'écran. Pour faire un dessin animé, il suffit de faire défiler à l'écran N images par seconde et donc de faire N transferts par seconde, chaque transfert étant séparé d'un temps mort pendant lequel l'image reste sur l'écran. (Pour éviter un scintillement, ce temps mort devra être au moins 2 fois supérieur au temps de transfert MEV → MEC).

ORGANIGRAMME DU DESSIN ANIME

Le logiciel d'animation est donc fort simple :



LE PROGRAMME

Il se décompose en 2 pavés :

Transfert BASIC vers Mémoire vive

Lignes 1000 à 1400
Proc. de transfert
Initialisations
Envoi d'une image
Boucle d'attente
Boucle d'affichage

Lignes 1000 à 1080
Lignes 1100 à 1190
Lignes 1200 à 1290
Lignes 1300 à 1350
Lignes 1360 à 1370

Transfert MEV vers Mémoire d'écran

Lignes 2000 à 2110
Message d'écran
Dessin animé

Lignes 2020 à 2070
Lignes 2080

Remarque :

Le temps de pause entre deux images peut varier en modifiant les lignes 1300 et 1310. Le temps de pause minimal est obtenu par :

```

1300 DATA # A2, # 01      'LDX # $ 01
1310 DATA # A0, # 01      'LDY # $ 01
  
```

LES VARIABLES UTILISEES

VARIABLES BASIC :

A	Lecture des données
A\$	Attente de l'appui sur une touche

I

Index de boucles

. VARIABLES MEMOIRE :

§ 08	Poids faible de début de mémoire vive
§ 09	Poids fort de début de mémoire vive
§ 0A	Poids faible de début de mémoire d'écran
§ 0B	Poids fort de début de mémoire d'écran
§ 0C	Nombre d'images à envoyer

```

10 : REM Dessin anime
20 :
30 : GOSUB 1000 'Transfert BASIC->MEV
40 : GOSUB 2000 'Transfert MEV->MEC
50 :
60 : END
70 : REM #####
1000 : REM Transfert BASIC->MEV
1010 :
1020 : FOR I=0 TO 52
1030 : READ A
1040 : POKE#3000+I,A
1050 : NEXT I
1060 :
1070 : RETURN
1080 : REM #####
1100 DATA #A9,#10 'LDA #$10
1110 DATA #B5,#0C 'STA $0C
1120 DATA #A9,#00 'LDA #$00
1130 DATA #B5,#08 'STA $08
1140 DATA #A9,#32 'LDA #$32
1150 DATA #B5,#09 'STA $09
1160 DATA #A9,#00 'LDA #$00
1170 DATA #B5,#0A 'STA $0A
1180 DATA #A9,#A0 'LDA #$A0
1190 DATA #B5,#0B 'STA $0B
1200 DATA #A2,#04 'LDX #$04

```

```

1210 DATA #A0,#00      'LDY #$0
1220 DATA #B1,#0B      'LDA ($0B),Y
1230 DATA #91,#0A      'STA ($0A),Y
1240 DATA #88           'DEY
1250 DATA #D0,#F9      'BNE $3018
1260 DATA #E6,#09      'INC $09
1270 DATA #E6,#0B      'INC $0B
1280 DATA #CA           'DEX
1290 DATA #D0,#F0      'BNE $3016
1300 DATA #A2,#00      'LDX #$00
1310 DATA #A0,#00      'LDY #$00
1320 DATA #88           'DEY
1330 DATA #D0,#FD      'BNE $302A
1340 DATA #CA           'DEX
1350 DATA #D0,#F8      'BNE $ 302B
1360 DATA #C6,#0C      'DEC $0C
1370 DATA #D0,#DB      'BNE $300C
1380 DATA #60           'RTS
1390 :
1400 : REM *****
2000 : REM Transfert MEV->MEC
2010 :
2020 : PAPER 0:INK 6
2030 : HIRES:PRINT CHR$(17)
2040 : PRINT"Pourcommencer le transfert,"
2050 : PRINT"appuyez sur une touche du clavier."
2060 : GET A$ 'Attente de l'appui sur une touche
2070 :
2080 : CALL #3000 'Dessin anime
2090 :
2100 : RETURN
2110 : REM *****

```

Calendrier perpétuel

PRESENTATION

Pour des raisons diverses, vous pouvez vouloir connaître le nom du jour correspondant à une date donnée. Eh bien, rien n'est plus simple grâce à la formule de GAUSS. qui fera tout le travail pour vous.

Le programme "le calendrier perpétuel" se sert de cette formule.

Après avoir entré la date (jour, mois et année) pour laquelle vous voulez connaître le jour, ORIC applique ces données à la formule et vous donne immédiatement le jour correspondant.

ORGANIGRAMME DU CALENDRIER

Le programme a la structure A.

Il est difficile de faire plus simple au niveau traitement :

3 tâches séquentielles composent le calendrier :

- . Entrée de la date,
- . Calcul du jour,
- . Affichage du jour.

LE PROGRAMME

Il se décompose en 3 pavés :

**Présentation et
Initialisation**

Lignes 1000 à 1680

Module de commande Lignes 1000 à 1060

Présentation Lignes 1100 à 1550

Initialisation Lignes 1600 à 1680

Traitement

Lignes 2000 à 2140

Lecture de la date

Calcul du jour

Lignes 2000 à 2070

Lignes 2080 à 2140

*Affichage
des résultats*

Lignes 3000 à 3060

Le jour est affiché

LES VARIABLES UTILISEES

A\$	Code couleur de ligne
AN\$	Année considérée
JOUR	Jour considéré
J\$	Jour de la semaine
M, C, A	Variables de calcul
MOIS	Mois considéré
Z	Numéro du jour de la semaine

```

10 : REM Le Calendrier Perpetuel
20 :
30 : GOSUB 1000 'Initialisation
40 : GOSUB 2000 'Traitement
50 : GOSUB 3000 'Fin
60 :
70 : END
80 : REM *****
1000 : REM Initialisation
1010 :
1020 : GOSUB 1100 'Presentation
1030 : GOSUB 1600 'Initialisation
1040 : RETURN
1050 :
1060 : REM *****
1100 : REM Presentation
1110 :
1120 : PAPER 0:INK 5

```

```

1130 : REM Titre
1140 :
1150 : PRINT CHR$(12);CHR$(4)
1160 : PRINT CHR$(27);"N";CHR$(27);"A LE CALENDRIER PERPE
TUEL";
1170 : PRINT CHR$(4)
1180 :
1190 : REM Objet
1200 :
1210 : PRINT:PRINT:PRINT:PRINT
1220 : A$="D":PRINT"OBJET :";
1230 : GOSUB 1500:PRINT"Utilitaire."
1240 :
1250 : REM Job
1260 :
1270 : PRINT:PRINT:PRINT:PRINT
1280 : A$="F":PRINT"JOB :";
1290 : GOSUB 1500:PRINT"Ce calendrier permet de"
1300 : GOSUB 1500:PRINT" trouver le nom d'un jour ,"
1310 : GOSUB 1500:PRINT" connaissant Jour/Mois/Annee ."
1320 : GOSUB 1500:PRINT" Ce calendrier est etabli "
1330 : GOSUB 1500:PRINT" a partir d'une formule de GAUSS"
;
1340 :
1350 : REM Debut
1360 : PRINT:PRINT:PRINT
1370 : A$="C":PRINT"Debut :";
1380 : GOSUB 1500:PRINT"Pour commencer,appuyez"
1390 : GOSUB 1500:PRINT" sur une touche quelconque ."
1400 :
1410 : GET A$:RETURN
1420 : REM *****
1500 : REM Couleur d'une ligne
1510 :
1520 : PRINT CHR$(27);"H";CHR$(27);A$;
1530 :
1540 : RETURN

```

```

1550 : REM #####
1600 : REM Initialisation des variables
1610 :
1620 :   FOR I=0 TO 6
1630 :     READ J$(I)
1640 :   NEXT I
1650 : DATA Dimanche,Lundi,Mardi,Mercredi,Jeudi,Vendredi,Samedi
1660 :
1670 : RETURN
1680 : REM #####
2000 : REM Traitement
2010 :
2020 :   CLS:PRINT:PRINT:PRINT:PRINT
2030 :   INPUT"Jour ";JOUR
2040 :   PRINT
2050 :   INPUT"Mois ";MOIS
2060 :   PRINT
2070 :   INPUT"Annee ";AN$
2080 :   M=MOIS-2:IF MOIS<=2 THEN M=MOIS+10
2090 :   C=VAL(LEFT$(AN$,2)):A=VAL(RIGHT$(AN$,2))
2095 :   IF MOIS<=2 THEN A=A-1
2100 :   X=INT(2.6*M-0.199)+JOUR+A+INT(A/4)+INT(C/4)-2*C
2110 :   Z=INT(X)-7*INT(X/7)
2120 : RETURN
2130 :
2140 : REM #####
3000 : REM Affichage du resultat
3010 :   PRINT:PRINT:PRINT
3020 :   PRINT "Le ";JOUR;"/ ";MOIS;"/ ";AN$;" est un";
3030 :   PRINT CHR$(27);"L";CHR$(27);"A";J$(Z)
3040 : RETURN
3050 :
3060 : REM #####

```

Rubick's cube

PRESENTATION

Pour illustrer le mouvement d'objets dans l'espace, le Rubick's Cube est un exemple intéressant.

Deux mouvements sont étudiés dans ce programme :

- . Rotation du cube autour d'un axe vertical
- . Rotation du cube autour d'un axe horizontal

Trois des 6 faces du cube sont affichées à l'écran. Chaque face possède 9 facettes qui sont initialisées en début de programme. Les deux rotations possibles ne modifient pas la position des facettes sur chaque face. Elles permettent seulement de visualiser successivement toutes les faces.

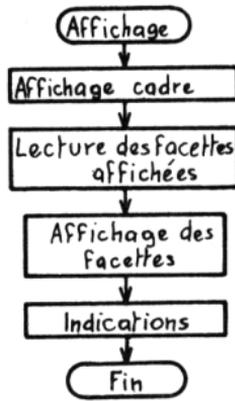
ORGANIGRAMME DU JEU

Après avoir entré la couleur (représentée par un nombre) de chaque facette, et après avoir initialisé les variables, le jeu peut commencer.

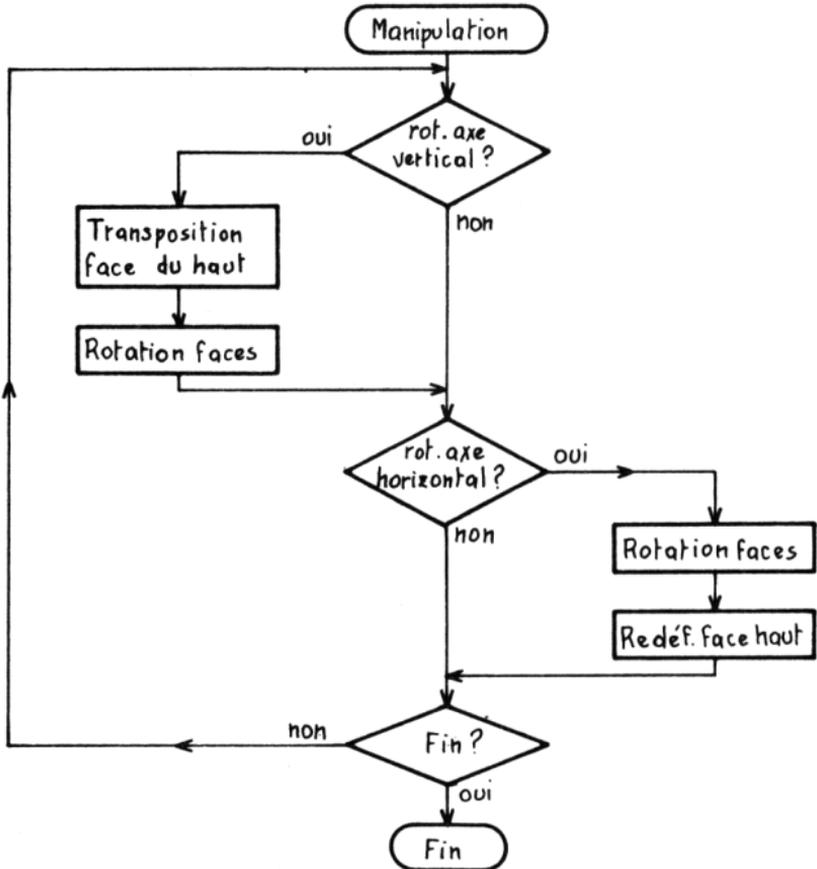
Il consiste en deux étapes essentielles :

- . Affichage
- . Manipulation

Affichage



Manipulation



LE PROGRAMME

Il se décompose en 2 pavés :

**Présentation et
Initialisation**

Lignes 1000 à 2980	
Module de commande	Lignes 1000 à 1060
Présentation	Lignes 1100 à 1650
Initialisation	Lignes 2000 à 2980

Manipulation cube

Lignes 3000 à 4420	
Module de commande	Lignes 3000 à 3070
Affichage cube	Lignes 3100 à 3580
Manipulation	Lignes 4000 à 4420
* Module de commande	Lignes 4000 à 4080
* Rotation axe vertical	Lignes 4110 à 4290
* Rotation axe horizontal	Lignes 4300 à 4420

LES VARIABLES UTILISEES

A\$	Code couleur de ligne
A, B	Variables de lecture des "DATAS"
CC	Tableau de couleur des facettes
I, J, K	Index de boucles
FA	Tableau des facettes affichées
FIN	Indicateur de fin de traitement
R\$	Saisie d'une commande de manipulation
T	Tableau représentant le cube
TF	Tableau des faces affichées
TP, TQ	Tableaux de travail sur une face

```

10 : REM RUBICK'S CUBE
20 :
30 : GOSUB 1000 'Presentation et Initialisation
40 : GOSUB 3000 'Manipulation du cube
50 :
60 : TEXT : END
70 : REM #####
1000 : REM Presentation et Initialisation
1010 :
1020 : GOSUB 1100 'Presentation
1030 : GOSUB 2000 'Initialisation
1040 :
1050 : RETURN
1060 : REM #####
1070 : PRINT CHR$(4);CHR$(27);"N";
1100 : REM Presentation
1110 :
1120 : PAPER 0 : INK 5 : PRINT CHR$(17)
1130 :
1140 : REM Titre
1150 :
1160 : CLS:PRINT CHR$(12)
1170 : PRINT CHR$(4);CHR$(27);"N";
1180 : PRINT CHR$(27);"A RUBICK'S CUBE";CHR$(4)
1190 :
1200 : REM Objet
1210 :
1220 : PRINT:PRINT:PRINT:PRINT:PRINT
1230 : A$="D" 'Code couleur = Bleu
1240 : PRINT"OBJET : ";
1250 : GOSUB 1600:PRINT"Manipulation du cube"
1260 : GOSUB 1600:PRINT" dans l'espace ."
1270 :
1280 : REM Job
1290 :
1300 : PRINT:PRINT:PRINT
1310 : A$="F" 'Code couleur = Cyan
1320 : PRINT"JOB : ";
1330 : GOSUB 1600:PRINT"Apres initialisation"
1340 : GOSUB 1600:PRINT" des facettes , le cube peut"
1350 : GOSUB 1600:PRINT" etre manipule dans tous les"

```

```

1360 :   GOSUB 1600:PRINT"      directions afin d'obtenir la"
1370 :   GOSUB 1600:PRINT"      reconstitution des faces ."
1380 :
1390 :   REM Debut
1400 :
1410 :   PRINT:PRINT:PRINT
1420 :   A$="C" 'Code couleur = Jaune
1430 :   PRINT"DEBUT : ";
1440 :   GOSUB 1600:PRINT"Pour commencer , appuyez"
1450 :   GOSUB 1600:PRINT"      sur une touche quelconque ."
1460 :
1470 :   GET A$ 'Attente de l'appui sur une touche
1480 :
1490 :   RETURN
1500 :   REM #####
1600 :   REM Couleur d'une ligne
1610 :
1620 :   PRINT CHR$(27);"H";CHR$(27);A$;
1630 :
1640 :   RETURN
1650 :   REM #####
2000 :   REM Initialisation des variables du jeu
2010 :
2020 :   CLS:INK 6:PRINT CHR$(17)
2025 :   DIM CC(54)
2030 :
2040 :   FOR I=0 TO 5
2050 :   CLS:A$="A" 'Code couleur = Jaune
2060 :   GOSUB 1600:PRINT"      Cube en position initiale"
2070 :
2080 :   PRINT
2090 :   A$="B":GOSUB 1600:PRINT"Face ";I+1;":"
2100 :   PRINT"Entrez la couleur de chaque facette "
2110 :   PRINT"selon le code indique en bas d'ecran."
2120 :   PLOT 4,24,"1=Rouge , 2=Vert , 3=Jaune"
2130 :   PLOT 4,25,"4=Bleu , 5=Magenta , 6=Cyan"
2140 :   FOR J=0 TO 2
2150 :   PRINT:PRINT
2160 :   A$="C":GOSUB 1600:PRINT"      Ligne ";J+1
2170 :   FOR K=0 TO 2
2180 :   A$="E":GOSUB 1600:PRINT"      Colonne ";K+1;

```

```

2190 : INPUT B
2200 : CC(I*9+J*3+K+1)=B
2210 : NEXT K
2220 : NEXT J
2230 : NEXT I
2240 :
2250 : REM Constitution du tableau Faces
2260 :
2270 : DIM T(3,3,6),FA(54)
2290 :
2300 : REM Faces 1 a 6
2305 : FOR K=1 TO 6
2310 : FOR I=1 TO 3
2320 : FOR J=1 TO 3
2330 : T(I,J,K)=(I-1)*3+J+(K-1)*9
2340 : NEXT J
2350 : NEXT I
2360 : NEXT K
2370 :
2870 :
2875 : REM Tableau des 3 faces affichees
2880 : FOR I=1 TO 3
2890 : FOR J=1 TO 3
2900 : TP(I,J)=T(I,J,5)
2910 : NEXT J
2920 : NEXT I
2930 :
2935 : REM Numero des 3 faces affichees et code de rotation (BI)
2940 : TF(1)=5:TF(2)=1:TF(3)=2
2950 :
2960 : RETURN
2970 :
2980 : REM *****
3000 : REM Manipulation du cube
3010 :
3020 : GOSUB 3100 'Affichage du cube
3030 : GOSUB 4000 'Manipulation
3040 : IF FIN<>1 THEN 3000
3050 :
3060 : RETURN

```

```

3070 : REM *****
3100 : REM Affichage du cube
3110 :
3120 : HIRES:INK 6
3130 : RESTORE
3150 :
3160 : REM Cadre
3170 :
3180 :   FOR I=1 TO 21
3190 :     READ A,B:CURSET A,B,1
3200 :     READ A,B:DRAW A,B,1
3210 :     NEXT I
3220 :
3230 : REM Facettes
3240 :
3241 :   FOR I=1 TO 3
3242 :     FOR J=1 TO 3
3243 :       FA((I-1)*3+J)=CC(TP(J,4-I))
3244 :       FA((I-1)*3+J+9)=CC(T(I,J,TF(2)))
3245 :       FA((I-1)*3+J+18)=CC(T(I,J,TF(3)))
3246 :     NEXT J
3247 :   NEXT I
3248 :
3250 :   IF TF(1)=5 THEN 3260
3251 :   FOR I=1 TO 3
3252 :     FOR J=1 TO 3
3253 :       FA((I-1)*3+J+9)=CC(T(4-I,4-J,TF(2)))
3254 :       FA((I-1)*3+J+18)=CC(T(4-I,4-J,TF(3)))
3255 :     NEXT J
3256 :   NEXT I
3257 :
3260 :   FOR I=1 TO 27
3270 :     READ C,D:CURSET C,D,0
3280 :     CHAR FA(I)+48,0,1
3285 :   NEXT I
3290 :
3300 : REM Indications
3310 :
3320 :   FOR I=1 TO 25
3330 :     READ A,B,C

```

```

3340 :      CURSET A,B,0:CHAR C,0,1
3350 :      NEXT
3360 :      CURSET 207,100,0:CHAR TF(1)+48,0,1
3361 :      CURSET 207,110,0:CHAR TF(2)+48,0,1
3362 :      CURSET 207,120,0:CHAR TF(3)+48,0,1
3363 :
3370 : RETURN
3380 : REM *****
3400 DATA 100,0,50,50,87,13,50,50,74,26,50,50,61,39,50,50
3410 DATA 100,0,-39,39,116,16,-39,39,133,33,-39,39,150,50,-39,39
3420 DATA 61,39,0,90,77,55,0,90,94,72,0,90,111,89,0,90
3430 DATA 124,76,0,90,137,63,0,90,150,50,0,90
3440 DATA 61,69,50,50,61,99,50,50,61,129,50,50
3450 DATA 111,119,39,-39,111,149,39,-39,111,179,39,-39
3460 :
3470 DATA 100,10,117,27,134,44,85,23,102,40,119,57,70,36,87,53,104,
70
3480 DATA 68,60,85,77,102,94,68,90,85,107,102,124,68,120,85,137,102
,154
3490 DATA 115,94,128,81,141,68,115,124,128,111,141,98,115,154,128,1
41
3500 DATA 141,128
3510 :
3520 DATA 165,85,70,171,85,97,177,85,99,183,85,101,189,85,115,195,8
5,58
3530 DATA 165,100,72,171,100,97,177,100,117,183,100,116,201,100,58
3540 DATA 165,110,71,171,110,97,177,110,117,183,110,99,189,110,104
3550 DATA 195,110,101,201,110,58
3560 DATA 165,120,68,171,120,114,177,120,111,183,120,105,189,120,11
6
3570 DATA 195,120,101,201,120,58
3580 : REM *****
4000 : REM Manipulation
4010 :
4020 :   A$="A":GOSUB 1600:PRINT"R1=Rotation axe vertical,R2=Rotati
on"
4030 :   GOSUB 1600:PRINT"axe horizontal,X=Sortie"
4040 :   INPUT "Choix ";R$
4050 :   IF R$="R1" THEN 4120
4060 :   IF R$="R2" THEN 4300

```

```

4070 :   IF R$="X" THEN FIN=1:RETURN
4080 :   GOTO 4000
4090 :   REM -----
4100 :   IF R$="R2" THEN 4300
4105 :   GOTO 4000
4110 :   REM -----
4120 :   REM Rotation autour d'un axe vertical
4130 :
4140 :   REM Transposition face du haut
4150 :   FOR I=1 TO 3
4160 :     FOR J=1 TO 3
4170 :       TQ(J,4-I)=TP(I,J)
4180 :     NEXT J
4190 :   NEXT I
4200 :   FOR I=1 TO 3
4210 :     FOR J=1 TO 3
4220 :       TP(I,J)=TQ(I,J)
4230 :     NEXT J
4240 :   NEXT I
4250 :   REM Rotation des faces Gauche et Droite
4260 :   TF(2)=TF(3):TF(3)=TF(3)+1
4270 :   IF TF(3)=5 THEN TF(3)=1
4280 :
4290 :   RETURN
4300 :   REM -----
4310 :   REM Rotation autour d'un axe horizontal
4320 :
4340 :   IF TF(1)=5 THEN TF(1)=6:TF(2)=4:TF(3)=2:GOTO 4360
4350 :   TF(1)=5:TF(2)=1:TF(3)=2
4360 :   FOR I=1 TO 3
4370 :     FOR J=1 TO 3
4380 :       TP(I,J)=T(I,J,TF(1))
4390 :     NEXT J
4400 :   NEXT I
4410 :
4420 :   RETURN

```

Classement par ordre croissant ou décroissant chaînes et nombres

PRESENTATION

Dans votre vie d'informaticien, vous serez sans doute amené à vouloir classer des données numériques (ou alphanumériques) par ordre croissant ou décroissant (respectivement alphabétique ou inverse alphabétique).

Le langage BASIC fait la différence entre caractères et nombres, c'est ainsi que deux programmes sont présentés :

- . Un pour le classement de chaînes de caractères,
- . Un pour le classement de nombres

Il existe un nombre impressionnant de méthodes de classement par ordre croissant et décroissant. Celle qui a été retenue ici offre deux avantages : sa rapidité et sa simplicité.

ORGANIGRAMME DU CLASSEMENT

Le programme a la structure A.

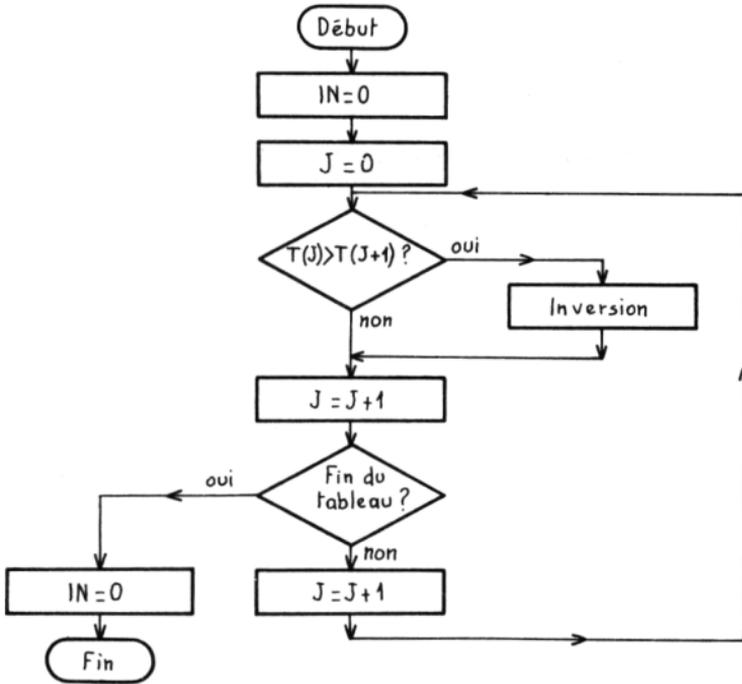
Examinons le cheminement du programme de classement par ordre croissant.

Le classement par ordre décroissant est très similaire.

Un tableau contient les entités à classer. La méthode consiste à balayer le tableau en "inversant" deux entités successives a et b si elles ne répondent pas à la condition ' $a > b$ ' ou ' $a < b$ ' (pour un classement décroissant respectivement croissant).

Le tableau sera balayé jusqu'à ce qu'il n'y ait plus aucune inversion dans un balayage complet.

L'organigramme suivant décrit cette méthode de classement.



Pour l'organigramme du classement par ordre décroissant, seul le test $T(J) > T(J+1)$ se transforme en $T(J) < T(J+1)$.

LE PROGRAMME

Il se décompose en 3 pavés :

Présentation

Lignes 1000 à 1550

Classement

Lignes 2000 à 2650

Acquisition des
données

Lignes 2000 à 2200

Classement

Lignes 2200 à 2650

Fin de classement

Lignes 3000 à 3100

Affichage des données classées

LES VARIABLES UTILISEES

A\$	Côde couleur de ligne puis variable de calcul
B\$	Chaîne des caractères à classer
C\$, D\$	Variables de calculs
I, J	Index de boucles
IN	Nombre de permutations effectuées
R\$	Sens du classement
T\$	Tableau contenant les chaînes à classer.

Pour le classement numérique, les variables **B\$**, **C\$**, **D\$**, **T\$** se changent en **B**, **C**, **D**, **T**.

Remarque :

Les programmes de classement de chaînes et de nombres sont différents. En effet, si nous ne faisons qu'un programme de classement, il opérerait sur des chaînes de caractères (puisque les chaînes peuvent aussi bien contenir de l'alphanumérique que du numérique). Malheureusement, la chaîne "21" est < à la chaîne "3" (d'un point de vue alphanumérique), ce qui est faux d'un point de vue numérique (on a en effet $21 > 3$!).

On est donc contraint de faire un programme de classement alphanumérique et un programme de classement numérique.

```

1 DIM T$(100)
10 : REM Classement par ordre Croissant et Decroissant
15 : REM de mots alphabetiques
20 :
30 : GOSUB 1000 'Presentation du programme
40 : GOSUB 2000 'Acquisition des donnees et classement
50 : GOSUB 3000 'Fin du programme
60 :
70 : END
80 : REM #####
1000 : REM Presentation du programme
1010 :
1020 : PAPER 0 : INK 6 : PRINT CHR$(17)
1030 :
1040 : REM Titre
1050 :
1060 : PRINT CHR$(12)
1070 : PRINT CHR$(4);CHR$(27);"N";
1080 : PRINT CHR$(27);"A CLASSEMENT CROISSANT ET DECROISSANT";
1090 : PRINT CHR$(4)
1100 :
1110 : REM Objet
1120 :
1130 : PRINT:PRINT:PRINT:PRINT:PRINT
1140 : A$="D" 'Code couleur = Bleu
1150 : PRINT"OBJET :";
1160 : GOSUB 1500:PRINT"Utilitaire mathematique."
1170 :
1180 : REM Job
1190 :
1200 : PRINT:PRINT:PRINT:PRINT
1210 : A$="F" 'Code couleur = Cyan
1220 : PRINT"JOB  :";
1230 : GOSUB 1500:PRINT"Ce programme se charge"
1240 : GOSUB 1500:PRINT"      d'effectuer un classement "
1250 : GOSUB 1500:PRINT"      croissant ou decroissant de"
1260 : GOSUB 1500:PRINT"      mots alphabetiques."
1280 :
1290 : REM Debut
1300 :
1310 : PRINT:PRINT:PRINT:PRINT

```

```
1320 : A$="C" 'Code couleur = Jaune
1330 : PRINT"DEBUT :";
1340 : GOSUB 1500:PRINT"Pour commencer , appuyez sur"
1350 : GOSUB 1500:PRINT"      une touche quelconque."
1360 :
1370 : GET A$ 'Attente de l'appui sur une touche
1380 :
1390 : RETURN
1400 : REM *****
1500 : REM Couleur d'une ligne
1510 :
1520 : PRINT CHR$(27);"H";CHR$(27);A$;
1530 :
1540 : RETURN
1550 : REM *****
1999 STOP
2000 : REM Acquisition des donnees et classement
2010 :
2020 : REM Acquisition des donnees
2030 :
2040 : CLS
2050 : PRINT CHR$(27);"L";CHR$(27);"C";
2060 : PRINT"      ACQUISITION DES DONNEES"
2070 : PRINT CHR$(17):PRINT:PRINT:PRINT
2080 : A$="B":GOSUB 1500
2090 : PRINT"Entrez les donnees a classer."
2100 : A$="B":GOSUB 1500
2110 : PRINT"La saisie se terminera par 9999 ."
2120 : PRINT:PRINT
2130 :
2140 : REPEAT
2150 : PRINT"      Donnee ";I;:INPUT B$
2160 : I=I+1:T$(I)=B$
2170 : IF INT(I/17)=I/17 THEN PAGE=1
2180 : UNTIL B$="9999" OR PAGE=1
2190 : IF PAGE=1 THEN PAGE=0:GOTO 2000
2200 :
2210 : REM Classement
2220 :
2230 : CLS:PRINT:PRINT
2240 : A$="C":GOSUB 1500
```

```

2250 : PRINT"Voulez-vous un classement croissant"
2260 : GOSUB 1500
2270 : INPUT"ou decroissant (C/D) ";R$
2280 :
2290 : IF R$="C" THEN CLAS=2500 ELSE CLAS=2600
2300 :
2310 : REPEAT
2320 :     IN=0
2330 :     FOR J=1 TO I-1
2340 :         C%=T$(J):D%=T$(J+1)
2350 :         GOSUB CLAS
2360 :     NEXT J
2370 : UNTIL IN=0
2380 :
2390 : RETURN
2400 : REM *****
2500 : REM Classement par ordre croissant
2510 :
2520 : IF C%>D% THEN T$(J)=D%:T$(J+1)=C%:IN=1
2530 :
2540 : RETURN
2550 : REM *****
2600 : REM Classement par ordre decroissant
2610 :
2620 : IF C%<D% THEN T$(J)=D%:T$(J+1)=C%:IN=1
2630 :
2640 : RETURN
2650 : REM *****
3000 : REM Fin du traitement
3010 :
3020 : CLS
3030 : PRINT CHR$(27);"L";CHR$(27);"C           DONNEES CLASSEES"
3040 : PRINT:PRINT:PRINT
3050 : FOR J=1 TO I
3060 :     A%=T$(J)
3070 :     IF A%<>"9999" THEN PRINT SPC(18);A%
3080 : NEXT J
3090 :
3100 : RETURN

```

Le programme de classement de nombres n'est pas donné ici. En effet, il est très similaire au programme de classement de mots alphabétiques. Les lignes qui suivent décrivent les modifications à apporter au second pour réaliser le premier.

```
1      : DIM T(100)
15     : REM de nombres
1260  : GOSUB 500:PRINT"      nombres."
2150  : PRINT"      Donnée ";I;:INPUT B
2160  : I=I+1:T(I)=B
2340  : C=T(J):D=T(J+1)
2520  : IF C > D THEN T(J)=D:T(J+1)=C:IN=1
2620  : IF C < D THEN T(J)=D:T(J+1)=C:IN=1
3060  : A=T(J)
3070  : IF A < 9999 THEN PRINT SPC(18);A
```

QUATRIÈME PARTIE
UTILITAIRE

Désassembleur 6502

PRESENTATION

Le langage d'assemblage (ou assembleur) est puissant et rapide. Son seul défaut est sa difficulté de mise en oeuvre...

Comme tout langage, il possède ses propres instructions. On les appelle "Codes-Opérateurs" (ou Codes-Op). Elles sont très spécifiques au microprocesseur utilisé.

Le 6502 possède 152 instructions qui sont détaillées ci-après.

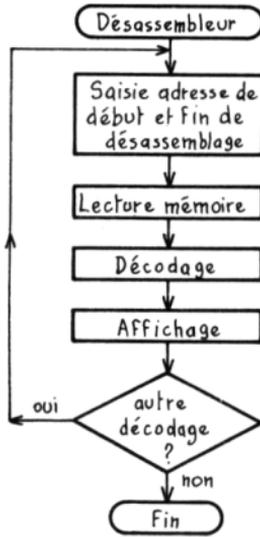
L'utilitaire de désassemblage proposé ici permet de convertir les données hexadécimales stockées en mémoire, en Codes-Opérateurs.

Le Code-Op peut être interprété plus facilement que le code hexa, car il reprend les 2 ou 3 premières lettres de l'action qu'il effectue. En vous reportant à la description de chaque instruction rencontrée, et, avec un peu de pratique, vous arriverez sans doute à comprendre les suites d'instructions écrites en mémoire.

ORGANIGRAMME DE L'UTILITAIRE

Le programme a la structure A.

L'utilitaire a la structure suivante :



LE PROGRAMME

Il se décompose en 2 pavés :

Présentation et Initialisation

Lignes 1000 à 2220	
Module de commande	Lignes 1000 à 1060
Présentation	Lignes 1100 à 1600
Initialisation	Lignes 1600 à 2220
(Lecture des codes-op et modes d'adressage)	

Désassemblage

Lignes 3000 à 4670	
Lecture bornes	Lignes 3000 à 3120
Désassemblage	Lignes 3130 à 3250
- Page 0, Y	Lignes 3500 à 4450
- Immédiat	Lignes 3600 à 3660
- Page 0	Lignes 3700 à 3760
- Absolu	Lignes 3800 à 3860
- Absolu, X	Lignes 3900 à 3960
- Absolu Y,	Lignes 4000 à 4060
- Indirect, X	Lignes 4100 à 4160
- Indirect, Y	Lignes 4200 à 4260
- Page 0, X	Lignes 4300 à 4360
- JMP indirect	Lignes 4400 à 4450
Calculs d'affichage	Lignes 4500 à 4670

LES VARIABLES UTILISEES

A\$	Code couleur de ligne
B	Variable intermédiaire lecture données
H2\$, H3\$	Codes hexa à afficher
I	Index de boucles
MA	Tableau des modes d'adressage
R\$	Variable réponse
T\$	Tableau des Codes-Op
CO, LA, CA, IN, H1, L, F2\$	Variables intermédiaires de désassemblage

```

10 : REM Desassembleur
20 :
30 : GOSUB 1000 'Presentation et Initialisation
40 : GOSUB 3000 'Desassemblage
50 :
60 : END
70 : REM *****
1000 : REM Presentation et Initialisation
1010 :
1020 : GOSUB 1100 'Presentation
1030 : GOSUB 1600 'Initialisation
1040 :
1050 : RETURN
1060 : REM *****
1100 : REM Presentation
1110 :
1120 : PAPER 0 : INK 5 : PRINT CHR$(17)
1130 : REM Titre
1140 : PRINT CHR$(12)
1150 : PRINT CHR$(4);CHR$(27);"N";
1160 : PRINT CHR$(27);"A          DESASSEMBLEUR";CHR$(4)
1170 :
1180 : REM Objet
1190 :
1200 : PRINT:PRINT:PRINT:PRINT:PRINT
1210 : A$="D" 'Code couleur = Bleu

```

```

1220 : PRINT"OBJET : ";
1230 : GOSUB 1500:PRINT"Utilitaire de desassemblage."
1240 :
1250 : REM Job
1260 :
1270 : PRINT:PRINT:PRINT
1280 : A$="F" 'Code coulNKur = Cyan
1290 : PRINT"JOB : ";
1300 : GOSUB 1500:PRINT"Ce programme permet de faire";
1310 : GOSUB 1500:PRINT"      la conversion du code"
1320 : GOSUB 1500:PRINT"      hexadecimal contenu en memoire";
1330 : GOSUB 1500:PRINT"      en code Assembleur ..."
1340 :
1350 : REM Debut
1360 :
1370 : PRINT:PRINT:PRINT
1380 : A$="C" 'Code couleur = Jaune
1390 : PRINT"DEBUT : ";
1400 : GOSUB 1500:PRINT"Pour commencer la conversion";
1410 : GOSUB 1500:PRINT"      appuyez sur une touche "
1420 : GOSUB 1500:PRINT"      quelconque ."
1430 : GET A$ 'Attente de l'appui sur une touche
1440 :
1450 : RETURN
1460 : REM *****
1500 : REM Module de couleur de ligne
1510 :
1520 : PRINT CHR$(27);"H";CHR$(27);A$;
1530 :
1540 : RETURN
1550 : REM *****
1600 : REM Initialisation des constantes du programme
1610 :
1620 : DIM T$(256),MA(256)
1630 :
1640 : REM Tableau des Codes-op
1650 : FOR I=0 TO 255
1660 :     READ B$:T$(I)=B$

```

```

1670 : NEXT I
1680 :
1690 : REM Tableau des Modes d'Adressage
1700 : FOR I=0 TO 255
1710 :   READ B:MA(I)=B
1720 : NEXT I
1730 :
1740 : REM Codes-op
1860 DATA "BRK","ORA",,,, "ORA","ASL",,"PHP","ORA","ASL A",,"ORA","
ASL",
1870 DATA "BPL","ORA",,,, "ORA","ASL",,"CLC","ORA",,,, "ORA","ASL",
1880 DATA "JSR","AND",,,, "BIT","AND","ROL",,"PLP","AND","ROL A",,"BI
T",,"AND"
1881 DATA "ROL",
1890 DATA "BMI","AND",,,, "AND","ROL",,"SEC","AND",,,, "AND","ROL",
1900 DATA "RTI","EOR",,,, "EOR","LSR",,"PHA","EOR","LSR A",,"JMP","E
OR",,"LSR",
1910 DATA "BVC","EOR",,,, "EOR","LSR",,"CLI","EOR",,,, "EOR","LSR",
1920 DATA "RTS","ADC",,,, "ADC","ROR",,"PLA","ADC","ROR A",,"JMP","A
DC",,"ROR",
1930 DATA "BVS","ADC",,,, "ADC","ROR",,"SEI","ADC",,,, "ADC","ROR",
1940 DATA ,"STA",,"STY","STA","STX",,"DEY",,"TXA",,"STY","STA","ST
X",
1950 DATA "BCC","STA",,"STY","STA","STX",,"TYA","STA","TXS",,"STA
",
1960 DATA "LDY","LDA","LDX",,"LDY","LDA","LDX",,"TAY","LDA","TAX",
1961 DATA "LDY","LDA","LDX",
1970 DATA "BCS","LDA",,"LDY","LDA","LDX",,"CLV","LDA","TSX",
1971 DATA "LDY","LDA","LDX",
1980 DATA "CPY","CMP",,"CPY","CMP","DEC",,"INY","CMP","DEX",
1981 DATA "CPY","CMP","DEC",
1990 DATA "BNE","CMP",,,, "CMP","DEC",,"CLD","CMP",,,, "CMP","DEC",
2000 DATA "CPX","SBC",,"CPX","SBC","INC",,"INX","SBC","NOP",,"CPX"
2001 DATA "SBC","INC",
2010 DATA "BEQ","SBC",,,, "SBC","INC",,"SED","SBC",,,, "SBC","INC",
2020 :
2030 : REM Mode d'adressage
2040 DATA 0,16,0,0,0,18,18,0,0,11,0,0,0,23,23,0

```

```

2050 DATA 19,16,0,0,0,18,18,0,0,25,0,0,0,24,24,0
2060 DATA 23,16,0,0,12,12,12,0,0,11,0,0,22,22,22,0
2070 DATA 19,17,0,0,0,18,18,0,0,25,0,0,0,24,24,0
2080 DATA 0,16,0,0,0,12,12,0,0,11,0,0,23,23,23,0
2090 DATA 19,17,0,0,0,18,18,0,0,25,0,0,0,24,24,0
2100 DATA 0,16,0,0,0,12,12,0,0,11,0,0,29,23,23,0
2110 DATA 19,17,0,0,0,18,18,0,0,25,0,0,0,24,24,0
2120 DATA 0,16,0,0,12,12,12,0,0,0,0,0,23,23,23,0
2130 DATA 19,17,0,0,18,18,10,0,0,25,0,0,0,24,0,0
2140 DATA 11,16,11,0,12,12,12,0,0,11,0,0,23,23,23,0
2150 DATA 19,17,0,0,18,18,10,0,0,25,0,0,24,26,25,0
2160 DATA 11,16,0,0,12,12,12,0,0,11,0,0,23,23,23,0
2170 DATA 19,17,0,0,0,18,18,0,0,25,0,0,0,24,24,0
2180 DATA 11,16,0,0,12,12,12,0,0,11,0,0,23,23,23,0
2190 DATA 19,17,0,0,0,18,18,0,0,15,0,0,0,23,23,0
2200 :
2210 : RETURN
2220 : REM *****
3000 : REM Desassemblage
3020 : CLS : INK 6 : PRINT CHR$(17)
3030 : PRINT"Entrez les adresses "
3040 : PRINT"de debut et de fin de desassemblage"
3050 : PRINT"en decimal ."
3060 : PRINT:PRINT
3070 : INPUT"      Debut : ";DEBUT
3080 : PRINT
3090 : INPUT"      Fin  : ";FIN
3100 : IF FIN <= DEBUT THEN 3000
3110 :
3120 : A=DEBUT
3130 : REPEAT
3140 :   CO=PEEK(A):LA=LEN(HEX$(A))
3150 :   CA=MA(CO):IN=CA-INT(CA/10)*10:H1=CO:L=LEN(HEX$(H1))
3160 :   F1%=RIGHT$(HEX$(A),LA-1):IF LEN(F1%)=4 THEN 3164
3161 :   FOR I=1 TO 4-LEN(F1%)
3162 :     PRINT "0";
3163 :   NEXT
3164 :   PRINT F1%;" ";

```

```

3165 : IF CD<>0 THEN F2%=RIGHT$(HEX$(H1),L-1) ELSE F2%="00"
3166 : IF LEN(F2%)=1 THEN PRINT "0";
3167 : PRINT F2%; " ";
3168 : IF IN=0 AND T$(CD)=" THEN 3180
3169 : IF IN=0 AND T$(CD)<>" THEN PRINT " ";T$(CD);:GOTO 3
180
3173 : IF CA=19 THEN IN=IN+1
3174 : IN=IN+1
3175 : ON IN GOSUB3500,3600,3700,3800,3900,4000,4100,4200,4300,
4400,4500
3180 : A=A+1:PRINT
3190 : UNTIL A>FIN
3200 : PRINT
3210 : INPUT"On continue (O/N) ";R$
3220 : IF R$="O" THEN GOTO 3000
3230 :
3240 : RETURN
3250 : REM *****
3500 : REM Page 0,Y
3510 :
3520 : GOSUB 4500
3530 : PRINT H2%; " ";T$(CD); " #";H2%;",Y";
3540 :
3550 : RETURN
3560 : REM -----
3600 : REM Immediat
3610 :
3620 : GOSUB 4500
3630 : PRINT H2%; " ";T$(CD); " #";H2%;
3640 :
3650 : RETURN
3660 : REM -----
3700 : REM Page 0
3710 :
3720 : GOSUB 4600
3730 : PRINT H2%; " ";T$(CD); " #";H2%;
3740 :
3750 : RETURN

```

```
3760 : REM -----
3800 : REM Absolu
3810 :
3820 : GOSUB 4600:GOSUB 4700
3830 : PRINT H2$;" ";H3$;" ";T$(CO);" "$;H3$;H2$;
3840 :
3850 : RETURN
3860 : REM -----
3900 : REM Absolu,X
3910 :
3920 : GOSUB 4600:GOSUB 4700
3930 : PRINT H2$;" ";H3$;" ";T$(CO);" "$;H3$;H2$;" ,X";
3940 :
3950 : RETURN
3960 : REM -----
4000 : REM Absolu,Y
4010 :
4020 : GOSUB 4600:GOSUB 4700
4030 : PRINT H2$;" ";H3$;" ";T$(CO);" "$;H3$;H2$;" ,Y";
4040 :
4050 : RETURN
4060 : REM -----
4100 : REM Indirect,X
4110 :
4120 : GOSUB 4600
4130 : PRINT H2$;" ";T$(CO);" ($";H2$;" ,X)";
4140 :
4150 : RETURN
4160 : REM -----
4200 : REM Indirect,Y
4210 :
4220 : GOSUB 4600
4230 : PRINT H2$;" ";T$(CO);" ($";H2$;" ),Y";
4240 :
4250 : RETURN
4260 : REM -----
4300 : REM Page 0,X
4310 :
```

```

4320 : GOSUB 4600
4330 : PRINT H2$;" ";T$(CO);" "$;H2$;" ,X";
4340 :
4350 : RETURN
4360 : REM -----
4400 : REM JMP Indirect
4410 :
4420 : GOSUB 4600:GOSUB 4700
4430 : PRINT H2$;" ";H3$;" ";T$(CO);" ($";H3$;H2$;" )";
4440 :
4450 : RETURN
4460 : REM -----
4500 : REM Relatif
4510 :
4520 : GOSUB 4600
4530 : PRINT H2$;" ";T$(CO);" ";
4540 : X$=HEX$(A+H2+1):PRINT"$"+RIGHT$(X$,LEN(X$)-1);
4550 :
4560 : RETURN
4570 : REM -----
4600 : REM Calcul de H2$
4610 :
4620 : H2=PEEK(A+1):A=A+1
4630 : L2=LEN(HEX$(H2))
4640 : H2$=RIGHT$(HEX$(H2),L2-1)
4645 : IF L2=1 THEN H2$="00"
4650 :
4660 : RETURN
4670 : REM -----
4700 : REM Calcul de H3$
4710 :
4720 : H3=PEEK(A+1):A=A+1
4730 : L3=LEN(HEX$(H3))
4740 : H3$=RIGHT$(HEX$(H3),L3-1)
4745 : IF L3=1 THEN H3$="00"
4750 :
4760 : RETURN
4770 : REM -----

```

JEU D'INSTRUCTIONS DU 6502 :

ADC	Addition avec retenue ; $A = A + M + C$ Affectés : N, V, Z, C
AND	ET logique ; $A = A \wedge M$ Affectés : N, Z
ASL	Décalage à gauche Affectés : N, Z, C
BCC	Branchement si pas de retenue ($C = 0$)
BCS	Branchement si retenue ($C = 1$)
BEQ	Branchement si zéro ($Z = 1$)
BIT	Test de bits (ET virtuel entre Accumulateur et mémoire spécifique). Affecté : Z
BMI	Branchement si négatif ($N = 1$)
BNE	Branchement si non égal à zéro ($Z = 0$)
BPL	Branchement si positif ou nul ($N = 0$)
BRK	Interruption Software
BVC	Branchement si pas de débordement ($V = 1$)
CLC	Annuler la retenue ; $C = 0$
CLD	Annuler le mode décimal ; $D = 0$
CLI	Autorise les interruptions ; $I = 0$
CLV	Annule l'indicateur de débordement ; $V = 0$
CMP	Compare Accumulateur et mémoire ; $A = A - M$ Affectés : N, Z, C
CPX	Compare X et mémoire ; $X = X - M$ Affectés : N, Z, C
CPY	Compare Y et mémoire ; $Y = Y - M$ Affectés : N, Z, C
DEC	Décrémente la mémoire ; $M = M - 1$ Affectés : N, Z
DEX	Décrémenter X ; $X = X - 1$ Affectés : N, Z
DEY	Décrémenter Y ; $Y = Y - 1$ Affectés : N, Z
EOR	OU exclusif entre Accumulateur et Mémoire Affectés : N, Z

INC	Incrémenter la mémoire ; $M = M + 1$ Affectés : N, Z
INX	Incrémenter X ; $X = X + 1$ Affectés : N, Z
INY	Incrémenter Y ; $Y = Y + 1$ Affectés : N, Z
JMP	Saut incondtionnel
JSR	Appel d'un sous-programme (PC est sauvegardé dans la pile)
LDA	Charger l'accumulateur ; $A = M$ Affectés : N, Z
LDX	Charger X ; $X = M$ Affectés : N, Z
LDY	Charger Y ; $Y = M$ Affectés : N, Z
LSR	Décalage à droite de l'accumulateur ou de la mémoire Affectés : N, Z, C
NOP	Pas d'opération
ORA	OU inclusif entre Accumulateur et mémoire Affectés : N, Z
PHA	Empiler A L'accumulateur est placé au sommet de la pile. Le pointeur de pile est mis à jour.
PHP	Empiler P Le registre d'état est placé au sommet de la pile Le pointeur de pile est mis à jour
PLA	Dépiler A Transfère vers A le contenu du sommet de la pile Le pointeur de pile est mis à jour. Affectés : N, Z
PLP	Dépiler vers P Transfère vers P le contenu du sommet de la pile Le pointeur de pile est mis à jour
ROL	Rotation à gauche de l'Accumulateur ou d'une mémoire Affectés : N, Z, C
ROR	Rotation à droite de l'Accumulateur ou d'une mémoire Affectés : N, Z, C
RTI	Retour d'interruption. Récupère PC et P sur la pile
RTS	Retour de sous-programme. Récupère PC sur la pile
SBC	Soustraction avec retenue ; $A = A - M - C$ Affectés : N, V, Z, C

SEC	Force à 1 bit de retenue ; C = 1
SED	Force à 1 le bit de Mode décimal ; D = 1
SEI	Inhibe les Interruptions ; I = 1
STA	Ranger l'accumulateur en mémoire ; M = A
STX	Ranger X en mémoire ; M = X
STY	Ranger Y en mémoire ; M = Y
TAX	Transfert de A dans X ; X = A Affectés : N, Z
TAY	Transfert de A dans Y ; Y = A Affectés : N, Z
TSX	Transfert de S dans X ; X = S Affectés : N, Z
TXA	Transfert de X dans A ; A = X Affectés : N, Z
TXS	Transfert de X dans S ; S = X
TYA	Transfert de Y dans A ; A = Y Affectés : N, Z

LES MODES D'ADRESSAGE

Syntaxe des Instructions Assembleur :

Les Instructions Assembleur du 6502 ont la forme suivante :

	Adresse	Code-Op	Opérande
avec :	Adresse	= Implantation du premier octet de l'instruction,	
	Code-Op	= Code opératoire de l'instruction ASM (un des 56 codes détaillés ci-avant),	
	Opérande	= Partie sur laquelle va porter l'instruction (adresse, donnée ou inexistante).	

Les modes d'adressage :

ADRESSAGE	FORMAT OPERANDE	OPERANDE EFFECTIF
Immédiat	## Valeur	Valeur
Absolu	\$ @	(@)
Page 0	\$ @	(@)
Page 0 Index X	\$ @ , X	(@ + (X))
Page 0 Index Y	\$ @ , Y	(@ + (Y))
Absolu Index X	\$ @ , X	(@ + (X))
Absolu Index Y	\$ @ , Y	(@ + (Y))
Relatif	\$ @	@
Indirect Préind X	(\$ @ , X)	((@ + (X)))
Indirect Postind Y	(\$@) , Y	((@) + (Y))
Indirect Absolu	(\$@)	((@))

Symboles utilisés :

@	Adresse
\$	La suite est exprimée en Hexadécimal
Valeur	Valeur Hexadécimale à deux Octets
(AD)	Contenu de la mémoire d'adresse AD

Imprimé en Belgique
sur les presses du Scorpion
D/1984/3548/5
N° d'édition 87109.11.1
I.S.B.N. 2.87109.011.4



POCHÉ

BOITE A OUTILS POUR ORIC ORIC 1 et ATMOS 20 PROGRAMMES EN BASIC

Qui n'a jamais regretté de ne pas avoir sous la main le tournevis ou la pince pour faire tel ou tel bricolage ? Les ordinateurs par leur aspect universel sont comme les boîtes à outils (et les auberges espagnoles...): on y trouve ce que l'on y met. Cette collection MegaO-Poché n'a pas la prétention de livrer le secret des grandes applications informatiques, mais bien de mettre à la disposition des utilisateurs des petits programmes tout prêts qui leur permettront de résoudre de nombreux problèmes de la vie quotidienne.



ISBN 2.87109.011.4





POÛCHE

BOITES A Outils POUR ORIC, TOMES 1 et 2

