

ORIC

ATMOS

VOS PROGRAMMIMES

 collection
micro
monde



cedic
nathan 

GESTION
DE FICHES

ORIC ATMOS

**VOS PROGRAMMES
BASIC ET LANGAGE MACHINE**

Michel Bussac
Gil Espèche



dirigée par Serge Pouts-Lajus

 **cedic
nathan**

32, bd St-Germain, 75005 Paris

- Minitel, votre guide pratique — *Jacques David*.
- ZX-Spectrum, votre micro-ordinateur — *Serge Pouts-Lajus*.
- Oric-Atmos, votre micro-ordinateur — *Michel Bussac*.
- MO5, votre micro-ordinateur — *Serge Pouts-Lajus*.
- Dictionnaire micro-informatique — *Éric Duceau, Christophe Doë*.
- Portatifs, les nouveaux micro-ordinateurs : *Pierre Raguènes, Gérard Sitbon*.
- MO5-TO7-TO7-70, vos programmes — *Serge Pouts-Lajus, Pierre Champeaux*.
- La micro en 100 questions. — *Bruno Delatour*.
- Mac Intosh, votre micro-ordinateur — *Jean-Baptiste Touchard*.
- Basic, votre méthode — *Serge Pouts-Lajus*.
- Choisir son micro-ordinateur — *Yvon Dargery*.

Couverture : studio Bercy
Maquette : Alain Dufourcq et Agathe Genot
Illustrations : Walter Lalonde et Annicka Boyriven
Photographies : Fabien Robineau

Ce volume porte la référence
ISBN 2-7124-1510-8

Toute reproduction, même partielle, de cet ouvrage est interdite. Une copie ou reproduction par quelque procédé que ce soit, photographie, photocopie, microfilm, bande magnétique, disque ou autre, constitue une contrefaçon passible des peines prévues par la loi du 11 mars 1957 sur la protection des droits d'auteur.

© CEDIC 1984

CEDIC, 32, boulevard Saint-Germain, 75005 - PARIS

SOMMAIRE

Avant-propos	5
Réflexion	7
Kim des lettres	7
Mots brouillés	13
Lecture flash	18
Dé automatique	22
J'apprends les opérations	26
Action	29
Champ de force	29
Descente à skis	34
Chenille et labyrinthe	39
Vingt-quatre Heures du Mans	44
Champ de météores	49
Serpent	53
Sisyphé	58
Mission bombardement	65
Création	73
Générateur de caractères	73
Télécran : éditeur de dessin LORES	78
Editeur de dessin HRES	82
Boîte à musique : éditeur musical	87
Editeur de DATA code machine	92
Conversion de base numérique	97
Gestion	101
Impôts	101
Crédit	106
Carnet électronique d'adresses	111

Annexes	118
Caractères graphiques et attributs.....	118
Gestion de la mémoire écran	120
Topographie de la mémoire vive d'Oric-Atmos	125
Code ASCII	126

AVANT-PROPOS

Ce recueil s'adresse à tous les possesseurs d'ORIC 1 ou d'ATMOS, débutants ou programmeurs confirmés. Nous l'avons voulu aussi varié que possible pour permettre à chacun d'y trouver ce qu'il cherche : des programmes faciles à comprendre pour les débuts en Basic, des programmes utiles pour la vie quotidienne, des jeux en code machine pour ceux qui cherchent à aller toujours plus loin, des outils pour compléter votre panoplie de programmeur.

Nous avons également cherché à tirer parti des possibilités graphiques et musicales de ces deux micro-ordinateurs avec des éditeurs de dessins et un synthétiseur. La gestion des fichiers n'est pas oubliée pas plus que la sauvegarde de blocs de mémoire écran. Mais c'est par la présentation des listings du code machine que nous souhaitons aider ceux d'entre vous qui veulent étudier et comprendre les codes hexadécimaux placés en DATA des jeux d'action.

Pour ces jeux, nous avons choisi d'utiliser le moins possible d'instructions d'affichage BASIC afin de gagner le maximum de temps à l'intérieur des boucles et d'obtenir un gain de rapidité pour ce langage. Nous utilisons donc principalement l'instruction POKE, avec laquelle on peut tout faire. Pour la même raison, nous avons choisi d'introduire dans nos programmes de petites routines en langage machine pour les mouvements mettant en œuvre de nombreux objets. C'est aussi afin d'assurer la compatibilité Oric I-Atmos que nous avons opté pour ces deux solutions.

Au-delà de l'amusement que nous souhaitons vous procurer, nous espérons que la lecture de ce livre libérera votre inspiration. On peut tout faire avec un ordinateur : il suffit d'un zeste d'imagination, de patience et de pas mal de logique. Mais cela vous le savez...Bon courage, donc.

Les auteurs.

KIM DES LETTRES



On n'entraîne jamais assez sa mémoire. Voici un jeu qui amusera grands et petits. Il teste la capacité de stockage immédiat de l'information. Un groupe de lettres va croissant; il faut le restituer sans erreur. Facile, dites-vous. Tout dépend de la longueur de la chaîne de lettres. Et puis songez que vous avez le choix entre trois

vitesses d'apparition : longue, moyenne, courte. Alors, réfléchissez à deux fois avant de jeter un défi à vos amis ou vos enfants !

Déroulement

Avec le titre s'affiche une règle du jeu et le choix du temps d'affichage. Puis le jeu débute par l'affichage d'une première lettre qu'il faut retaper ensuite. Si la réponse est juste, le jeu continue en ajoutant à la première une seconde lettre, tirée aléatoirement. Dans le cas contraire, un bilan s'affiche avec la possibilité de recommencer une partie.

```
5 '  
6 '  
7 '  
8 ' KIM DES LETTRES  
9 ' *****  
10 '  
11 '  
12 '  
20 GOSUB 30000:GOSUB 31000  
99 '  
100 'Progr. Principal  
101 '  
110 GOSUB10000  
120 FOR I=1 TO 25:PRINT:NEXT I  
140 PRINTAB$  
150 FOR I=1 TO T:NEXT I  
170 FOR I=1 TO 25:PRINT:NEXT I  
200 PRINT"Quel etait le Groupe de lettres?"  
210 INPUT SA$:GOSUB 11000  
230 IF HT=0 THEN 270  
240 GOSUB 12000:GOSUB 15000  
260 GOTO 110:GOSUB 14000  
270 GOSUB 14000:GOSUB 20000  
290 IF A#="0" THEN 300 ELSE 900  
300 SA#="" AB#="" C=0 :GOTO 20  
999 '  
900 'Fin  
901 '  
905 CLS:PAPER7:PRINT CHR$(17)  
910 PLOT12,13,CHR$(10)+CHR$(18)  
+CHR$(1)+"AU-REVOIR"+CHR$(23)
```

```
+CHR$(1)+"AU-REVOIR"+CHR$(23)  
920 PLOT12,14,CHR$(10)+CHR$(18)  
930 PRINT CHR$(6)  
940 END  
9999 '  
10000 'Aleatoire  
10001 '  
10010 A=INT(RND(1)*26)+65  
10020 A#=CHR$(A)  
10030 AB#=AB#+A#  
10040 RETURN  
11000 '  
11001 'Analyse de rePonse  
11002 '  
11005 IF SA#=AB# GOTO 11040  
11010 HT=0:RETURN  
11040 HT=1:RETURN  
11050 '  
12000 'bonne rePonse  
12001 '  
11500 GET A#  
12010 GET A#:RETURN  
13999 '  
14000 'mauv. reP.  
14001 '  
14005 FOR I= 1 TO 20:PRINT:NEXT I  
14030 PRINT"Ce n'est Pas cela..."  
14040 FOR I=1 TO 10:PRINT:NEXT I  
14070 RETURN  
14999 '  
15000 'score  
15001 '  
15010 C=C+1:RETURN  
19999 '  
20000 'Suite  
20001 '  
20010 PRINT"Ta rePonse etait: "  
20030 PRINTSA#  
20040 PRINT  
20050 PRINT"Et la bonne rePonse: "  
20060 PRINTAB#  
20070 PRINT  
20090 PRINT  
20090 PRINT"Tu t'es rapPele un ensemble de:"
```

```

19583 PRINTC)" lettres."
20100 PRINT"Veux-tu continuer ?"
20110 PRINT"(Oui ou Non)"
20120 GET A$:RETURN
30000 '
30010 ' Titre
30011 '
30015 CLS:PAPER7: POKE #26A,10
30020 PLOT12,13,CHR$(14)+CHR$(18)+CHR$(1)
+"KIM DES LETTRES"+CHR$(23)
30030 PLOT12,14,CHR$(14)+CHR$(18)+CHR$(1)
+"KIM DES LETTRES"+CHR$(23)
30040 GET A$:CLS:PAPER2:INK1:RETURN
31000 CLS:PRINT"Je vais te presenter des "
31005 PRINT"lettres dont tu devras "
31010 PRINT"te souvenir."
31020 PRINT:PRINT"Tu tacheras de les ";
31030 PRINT"retaper ensuite."
31040 PRINT:PRINT:
31050 PRINT"Veux-tu qu'elles restent
affichees: "
31060 PRINT"1- Peu"
31070 PRINT"2- moyennement"
31080 PRINT"3- long temps"
31090 GET T
31100 T=T*200
31110 FOR I=1 TO 12:PRINT:NEXT I
31140 PRINT"D'accord ! Allons-y..."
31150 A$=""
31160 FOR I=1 TO 12:PRINT:NEXT I
31190 FOR I=1 TO 2000:NEXT I:RETURN

```

Commentaires

20 Sous-programmes de titre et règle du jeu.

110 Appel du sous-programme de tirage aléatoire de la chaîne de lettres.

120 et 170 Boucles de saut de ligne pour effet de scrolling (défilement du texte).

140 Affichage chaîne.

210 Entrée de la réponse et envoi du sous-programme d'analyse de réponse.

260 Retour à la question suivante et envoi du sous-programme de comptage du nombre de lettres proposé.

270 Appel du sous-programme de mauvaise réponse, suivi de celui du bilan.

290 Sortie du sous-programme de relance du jeu (20100-20120).

300 Réinitialisation des variables.

910-920 Caractère en double hauteur fixé par CHR\$(10); couleur de forme rouge [CHR\$(1)], fond vert [CHR\$(18)] et effacement à droite [CHR\$(23)].

930 Rétablissement du curseur supprimé en 30015.

10010 Tirage aléatoire dans l'intervalle des 26 lettres de l'alphabet.

10030 Concaténation de la nouvelle lettre avec la chaîne déjà formée.

11010 HT=0 est le drapeau de réponse fausse.

11020 HT=1 est celui de la réponse juste.

11508 GET permet une attente du programme jusqu'à ce qu'une touche soit frappée. Très pratique pour « tourner une page ».

30015 POKE #26A,10 - suppression curseur.

31100 Calcul du temps d'affichage par allongement de la boucle de temporisation en ligne 150 (variable T).

Se reporter en annexe pour l'affichage des caractères en double hauteur.

L'effet de scrolling obtenu par une répétition de PRINT dans une boucle est réutilisable (120-170).

Adaptation de ce jeu pour le transformer en « Kim des chiffres » :

Bien entendu, remplacer le mot « lettres » par « nombres » partout où il apparaît. Pour le reste, voici les lignes d'instruction qui changent :

```
210 INPUT SA : GOSUB 11000
```

```
215 SA$=STR$(SA) :IF ASC(SA$)=32 THEN SA$  
=MID$(SA$,2)
```

```
10010 A$=STR$(INT(RND(1)*10))
```

```
10015 IF ASC(A$)=32 THEN A$=MID$(A$,2)
```

Note

STR\$ transforme une valeur numérique en chaîne alphanumérique. Mais le premier caractère de la chaîne comporte le signe si le nombre est négatif et reste blanc si le nombre est positif. C'est la raison pour laquelle, on teste la présence de ce blanc (ASCII 32) afin de l'enlever par MID\$.

Pour varier les plaisirs, offrons à ce programme un titre clignotant, en double hauteur :

```
30020 PLOT 3,17,CHR$(14)+CHR$(20)+CHR$(1)+« KIM DES  
NOMBRES »+CHR$(23)
```

```
30030 PLOT 3,18,CHR$(14)+CHR$(20)+CHR$(1)+« KIM DES  
NOMBRES »+CHR$(23)
```

MOTS BROUILLÉS

Voici un jeu d'entraînement au décodage. L'ordinateur choisit au hasard un mot dans une liste fournie en DATA. Il brouille l'ordre des lettres et l'affiche à l'écran. A vous de taper le mot que vous croyez identifier. C'est un excellent exercice de vocabulaire et de déduction pour les enfants. De bonnes soirées familiales en perspectives où les jeunes risquent d'avoir le dernier...mot !

Déroulement

Après RUN, le titre s'affiche avec la possibilité d'accéder à la règle du jeu. « Je vais penser à un mot que je te présenterai en brouillant les lettres. A toi de deviner ce mot. Si tu donnes ta langue au chat, tape + et je te ferai une suggestion qui te mettra certainement sur la piste. ». Après avoir appuyé sur une touche, le jeu est lancé. « Voici le mot brouillé :GERILRL.

Quelle est ta réponse ? »

Il s'agit de trouver le mot avec un minimum d'essais et d'aides. Le jeu est terminé, lorsque la liste de mots est épuisée.

Structure

20-60	Initialisations
110-190	Titre
200-490	Programme principal
10000-10100	Sous-programme d'affichage de la règle
11000-11020	Tirage aléatoire
12000-13020	Analyse de réponse
15000-15040	Score
16000-16050	Aide
18000-18040	Recommencer une partie
18500-18620	Suppression de mot mal utilisé
20000-fin	Données

```

8 '
9 '
10 ' MOTS BROUILLES
11 '
15 CLS
20 'initialisations
30 TL=20:BL=TL
40 DIM FL$(TL),HN$(TL),B$(20)
50 FOR I=1 TO TL
60 READ FL$(I),HN$(I)
70 NEXT I
100 ' titre
101 '
110 PRINT@12,12;"Mots brouilles"
120 PRINT:PRINT:PRINT
130 FOR I= 1 TO 200:NEXT I
140 PRINT "Voulez-vous la regle du jeu ?"
150 PRINT"O/N'"
170 GOSUB 19000
180 IF A$="O" THEN 190 ELSE 200
190 GOSUB 10000
199 '
200 'debut jeu
210 GOSUB 11000
220 C$=FL$(R):L=LEN(C$)
230 T$=""
250 FOR I= 1 TO L
260 B$(I)="#":NEXT I
280 FOR I=1 TO L
290 D$=MID$(C$,I,1)
300 J=INT(RND(1)*L)+1
310 IFB$(J)="#"THEN330
320 GOTO 300
330 B$(J)=D$
340 NEXT I
350 FOR I=1 TO L
360 T$=T$+B$(I):NEXT I
380 IF T$=C$ THEN 240
410 PRINT:PRINT:PRINT"Voici le mot brouille :"
420 PRINT:PRINTT$
430 PRINT:PRINT
440 PRINT"Quelle est ta reponse ?"
450 PRINT" Aide= '+'"
460 INPUT S$

```

```

470 GOSUB 12000
480 IF HT=1 THEN 15000
490 GOTO 410
1000 ' regle
10000 REM Regle
10010 CLS:PRINT@12,1;"Regle du jeu"
10020 PRINT:PRINT:PRINT
10030 PRINT"Je vais Penser a un mot"
10040 PRINT"que Je te Presenterai en"
10050 PRINT"brouillant les lettres."
10060 PRINT:PRINT:PRINT"A toi de deviner le mot."
10070 PRINT:PRINT:PRINT"Si tu donnes ta langue"
10080 PRINT"au chat, Tape '+' et "
10090 PRINT"Je te ferai une suggestion."
10095 PRINT"qui te mettra certainement"
10096 PRINT"sur la Piste."
10097 PRINT:PRINT"TaPe sur une touche"
10098 GET A$
10100 GET A$:CLS:RETURN
11000 'aleatoire
11010 R=INT(RND(1)*BL)+1
11020 RETURN
12000 'analyse reponse
12010 IF S$="+" THEN GOSUB 16000
12020 TR=TR+1
12025 IF TR>3 THEN 18000
12030 IF S$=C$ THEN 12100
12040 GOTO 13000
12100 HT=1
12110 PRINT"Bravo !"
12120 RETURN
13000 'mauvaise reponse
13010 PRINT"Essaye ";
13015 PRINT"encore une fois"
13016 PRINT"      Essai : "TR
13020 RETURN
15000 'score
15010 HT=0
15020 PRINT"Tu as trouve le mot en "
15030 IF TR=1 THEN 1570
15040 PRINT TR;"essais"
15050 GOTO 18000
15070 GOTO 18010
16000 'aide

```

```

16010 PRINT
16020 PRINT HN$(R)
16030 HN$(R)="Pas d'autres suggestions!"
16040 PRINT
16050 RETURN
18000 'recommencer
18010 PRINT"Veux-tu un autre mot(O/N)";
18020 GOSUB 19000
18030 IF A$="O" THEN 18100
18040 CLS:PRINT"Au-revoir"
18050 END
18100 S$="":TR=0:HT=0
18500 'suppression mot utilise
18510 FL$(R)=FL$(BL)
18520 HN$(R)=HN$(BL)
18530 BL=BL-1
18540 IF BL=0 THEN 18600
18550 GOTO 210
18600 PRINT"Excuse-moi mais tu as epuise"
18610 PRINT"mon stock de mots..."
18620 CLS:GOTO 18040
19000 'test clavier
19010 A$=KEY$
19020 IF A$=""THEN 19010
19030 RETURN
20000 DATA BONJOUR,SALUT,BEAU
20010 DATA ADMIRABLE,ADIEU,AU-REVOIR
20030 DATA AUTOROUTE,MOTORWAY,LOGICIEL
20040 DATA PROGRAMME,ADORER,AIMER
20050 DATA DEVORER,ENGLOUTIR,PERDRE
20060 DATA GASPILLER,DORMIR,SOMMEILLER
20070 DATA CONGELATEUR,FRIGIDAIRE
20100 DATA AEROPLANE,AVION,PISTE
20110 DATA CIRCUIT,CALCULER,COMPTER
20120 DATA DIRIGER,ORGANISER,CONDUIRE
20130 DATA PILOTER,HUITRE,MOULE
20140 DATA FRITE,CHIPS,CREUSER
20150 DATA APPROFONDIR,RIRE,RIGOLER
20160 DATA GRILLER,BRULER

```

Commentaires

30 TL Variable de dimensionnement de liste.
 BL Pointeur de liste en cours de partie, au fur et à mesure de la suppression des mots déjà utilisés.

40 Dimensionnement des tableaux de mots et de leur équivalent.

220 C\$ est le mot tiré au hasard dans le tableau FL\$.

260 B\$ Initialisation du tableau servant à brouiller un mot. Le signe dièse servant à tester si la case ne contient pas de lettres (ligne 310).

280-340 Boucle de brouillage de l'ordre des lettres d'un mot. D\$ est affecté d'une lettre du mot. Un nombre aléatoire est tiré en 300. La lettre est affectée à une case prise aléatoirement dans le tableau B\$. La ligne 300 filtre l'affectation pour éviter les superpositions de lettres.

350-360 Boucle de chaînage du mot brouillé.

460 S\$ est la réponse.

480 HT Drapeau de bonne ou mauvaise réponse.

18500-18550 Boucle de suppression du mot utilisé. Le pointeur initialisé à 20 au départ est décrémenté de 1 à chaque passage. Auparavant (en 18510-520), le dernier mot de la liste a pris la place de celui qui vient de servir. De même, pour le mot de la liste des équivalents (HN\$). Lorsque BL vaut 0, le jeu est fini.

20000 Pour la saisie des DATA, prendre garde à faire suivre chaque mot de son équivalent.

Bloc-notes

La routine de brouillage de listes de lettres dans un tableau est réutilisable (250-260); de même, la gestion des listes de DATA avec suppression des mots au fur et à mesure qu'ils ont servi (18500 et suivantes).



De nos jours, le besoin de lire vite et bien se fait de plus en plus sentir. La masse d'informations à absorber visuellement est si diverse et si considérable qu'il convient d'entraîner le regard et l'esprit à une telle activité. L'ordinateur offre dans ce domaine un outil intéressant puisqu'il permet l'affichage rapide de mots et la possibilité de contrôler si ces mots ont été identifiés. Le jeu que voici utilise cette possibilité dans le but d'entraîner les jeunes lecteurs à un repérage rapide. Il ne s'agit pas de lecture proprement dite mais d'entraînement à la lecture, un peu comme on fait des gammes.

Déroulement

Après le titre et la règle du jeu, vous choisissez votre vitesse d'apparition des mots. Ensuite, vous sélectionnez votre mot-cible. Le jeu peut alors commencer. Des mots apparaissent au hasard sur votre écran. A vous d'identifier le mot-cible. Chaque fois que vous croyez le voir, appuyez sur la touche RETURN. Si c'est juste vous marquez un point. Dans le cas contraire, votre stock de réponses fausses augmente d'un point.

```

1 '
2 '
3 'LECTURE FLASH
4 '
5 FOR I=1 TO 4:POKE#BBA3+I,#20:NEXT
6 POKE#26A,10:CLS:PAPER0:INK3
7 PLOT12,13,CHR$(14)+CHR$(18)+CHR$(1)
  +"LECTURE FLASH"+CHR$(16)
8 PLOT12,14,CHR$(14)+CHR$(18)+CHR$(1)
  +"LECTURE FLASH"+CHR$(16)
9 GET A$:CLS
15 RJ=0:RF=0
20 GOSUB1000:GOSUB 2000:GOSUB3000
46 '
47 'Vitesse
48 '
50 PRINT@2,6;"Pour attraper le mot"
60 PRINT@2,7;"choisi, appuie sur ENTREE"
80 PRINT@2,8;"Duree d'affichage":INPUTV1
90 V1=V1*30:CLS:GOSUB1000
99 '
100 'Boucle du jeu
101 '
110 READ ND:FOR I=1 TO ND:READ M#
120 X=INT(RND(1)*30)+1
130 Y=INT(RND(1)*15)+7
155 PLOTX,Y,M#
160 FOR J=1 TO V1:NEXTJ
170 I$=KEY$
180 IF I$=CHR$(13) AND M#=MT# THEN 185
      ELSE 190

```

```

185 RJ=RJ+1:GOSUB1000:GOTO 200
190 IF I#=CHR$(13) THEN RF=RF+1:GOSUB 1000
200 PLOT X,Y,"      "
210 NEXT I
220 CLS:PRINT@5,5:"RePonses justes: "RJ
230 PRINT@5,7:"RePonses fausses: "RF
240 PRINT:PRINT:
250 PRINT"Veux-tu continuer(O/N)"
260 A#=KEY$:IF A#=""THEN260
270 IF A#="O"THEN280 ELSE 290
280 RESTORE:GOTO 10
290 CLS:END
999 '
1000 'Score
1001 '
1010 PRINT@28,2:"JUSTES: "RJ
1020 PRINT@28,3:"FAUX: "RF
1030 RETURN
1999 '
2000 'menu
2001 '
2010 PRINT@ 5,10:"Veux-tu ?"
2020 PRINT@5,12:"1-Jouer"
2030 PRINT@5,14:"2-arreter"
2040 PRINT@5,16:"Ton choix";
2050 A#=KEY$:IF A#=""THEN 2050
2060 A=VAL(A#):IF A=1 THEN RETURN
2070 CLS:END
2999 '
3000 'Choix mot temoin
3001 '
3005 CLS:GOSUB1000
3010 PRINT@5,10:"Choisis l'un de ces "
3020 PRINT@5,11:"grouPes de lettres"
3030 T$(1)="RSRS":T$(2)="STSTTS"
3040 T$(3)="PLRPL":T$(4)="CVCV"
3050 FOR I=1 TO 4
3060 PRINTTAB(10),I;"-":T$(I)
3070 NEXT I
3080 PRINT:PRINT:
3090 PRINTTAB(5)"Numero choisi";
3100 R#=KEY$:IF R#=""THEN3100
3110 R=VAL(R#):IF R>4 THEN 3100
3120 AT#=T$(R)

```

```

3130 CLS:GOSUB 1000:RETURN
5999
6000 DATA 30,LSST,RSRS,STSTTS,PLRPL
6010 DATA CVCV,VCCV,LLCT,RRRS,PLPPL
6020 DATA STSTTS,MRTVC,RSRS,CVCV,CVCV
6030 DATA SSTTS,STSTTS,PLRPL,XXVX
6040 DATA RSRS,RSRS,STSTTS,MSRS,PPLRP
6050 DATA CVVC,RSRS,STSTTS,VCCV,MSRS
6060 DATA STSTTS,PLRPL

```

Commentaires

5 Suppression du CAPS, en haut et à droite.

6 Suppression du bip clavier et curseur.

7-8 Titre clignotant en double hauteur.

110 ND Nombre de DATA. Ce paramètre permet d'augmenter ou de diminuer la durée du jeu qui est fonction du nombre de données affichées.

120-130 Calcul aléatoire des coordonnées d'affichage d'un mot.

155 Affichage d'un mot.

160 Durée d'affichage paramétrée par V1.

170-185 Test de bonne réponse.

190 Test de mauvaise réponse.

200 Effacement du mot.

210 Fin de boucle centrale.

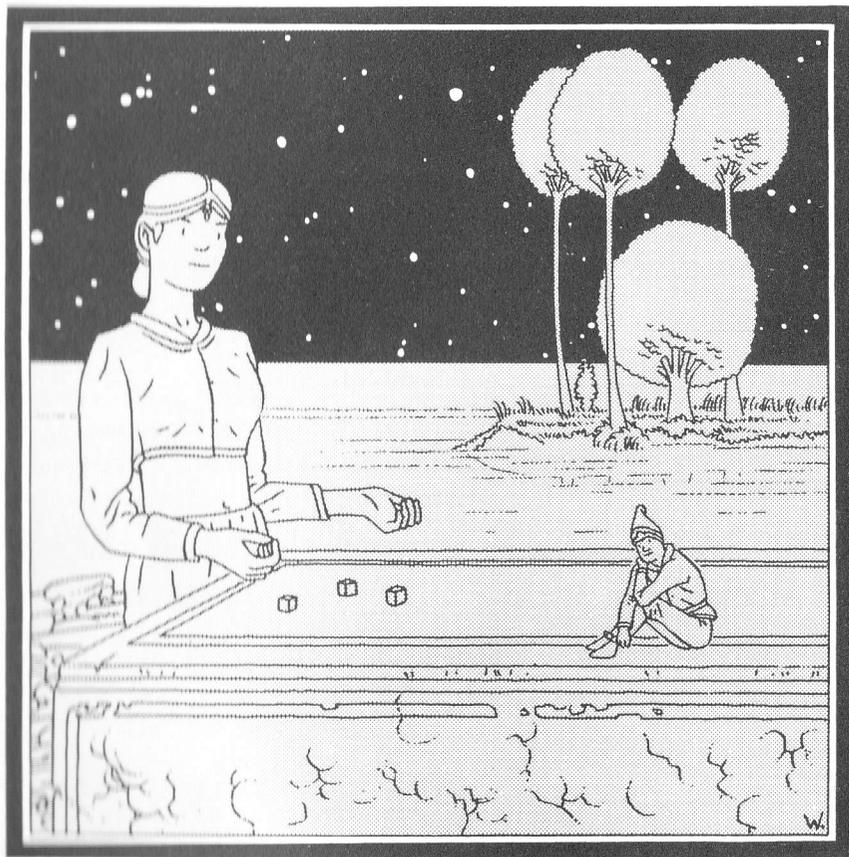
220-230 Affichage du bilan.

280 Remise à zéro du pointeur de DATA.

1000-1030 Sous-programme d'affichage du score.

3030-3040 Tableau des mots-témoins. Vous pouvez les changer à votre guise à condition d'en disséminer quelques-uns dans les DATA.

6000 DATA interchangeable.



Si vous aimez jouer contre le hasard, si vous êtes un fana du 421 ou autre jeu de dé, ce programme devrait vous intéresser. Cette fois c'est l'ordinateur qui jette le dé. Vous n'avez qu'à appuyer sur une touche, il s'exécute dans les plus brefs délais. A vous de parier sur le chiffre qui va sortir. A partir de là, il vous est possible d'échafauder vos martingales ou toute autre théorie fondée sur la probabilité. Après un bon entraînement sur ce simulateur, faites sauter la banque !

Déroulement

Après RUN, le titre du programme apparaît. En appuyant sur une touche, vous entrez directement dans le jeu. Un dé s'affiche sur la partie gauche de l'écran. « Appuyez sur une touche pour jeter la dé ». Dès que c'est fait, les combinaisons défilent jusqu'à ce que retentisse le *ping* indiquant que le jeu est fait. Jusqu'à ce moment, vous pouvez parier seul ou à plusieurs. Ce jeu se prête à toutes les règles que vous pourrez inventer pour transformer votre ordinateur en tapis vert de casino et vous en croupier-directeur de la banque...

```

1 REM
3 REM  DE AUTOMATIQUE
4 REM
10 REM
20 PAPER 4:INK 3
30 CLS:HIRE$
40 POKE #26A,10:GOSUB 1000
45 GET A$
46 CLS:HIRE$
50 GOSUB 430
54 '
56 REM
60 POKE #26A,10
70 PRINT" Appuyez sur une touche..."
80 PRINT" Pour jeter le DE ."
100 IF KEY$="" THEN 100
130 T=INT(RND(1)*3)+5
140 FOR I= 1 TO T
160 A=1
180 GOSUB 260
190 R=INT(RND(1)*6)+1
200 A=A+R
210 GOSUB 260
220 NEXT I
240 PING
250 GOTO 100
255 REM
256 ' Affichages
257 REM
260 GOSUB 590
270 ON R GOTO 280,300,330,350,380,400
    
```

```

280 CURSET 73,80,1
290 CHAR 95 ,1,A
295 RETURN
300 CURSET 50,60,1
305 CHAR 95 ,1,A
310 CURSET 98,100,1
320 CHAR 95 ,1,A
325 RETURN
330 GOSUB 280
340 GOTO 300
350 CURSET 50,100,1
355 CHAR 95 ,1,A
360 CURSET 98,60,1
365 CHAR 95 ,1,A
370 GOTO 300
380 GOSUB 350
390 GOTO 280
400 CURSET 73,60,1
405 CHAR 95,1,A
410 CURSET 73,100,1
415 CHAR 95 ,1,A
420 GOTO 350
429 REM
430 REM Initialisations
431 REM
440 R=1
450 CURSET 50,50,0
460 FILL 70,10,127
480 PAPER 4
490 INK 3
500 SOUND 4,5,15
510 GOTO 260
590 SOUND 4,4,15
600 PLAY 0,1,0,0
90 WAIT 1
620 PLAY 0,0,0,0
630 WAIT 140
640 RETURN
999 REM
1000 REM titre
1001 REM
1002 PAPER 4
1005 AB="DE AUTOMATIQUE"
1010 FOR W=1 TO 15

```

```

1020 CURSET 20,50+W*3,3
1030 GOSUB 1500
1040 NEXT W
1050 RETURN
1500 REM
1510 FOR I=1 TO LEN(A$)
1520 CHAR ASC(MID$(A$,I)),0,1
1530 DRAW W,0,3
1540 NEXT I
1550 RETURN

```

Commentaires

3Ø Mode HIRES (haute résolution) : vous disposez de trois lignes de texte en bas de l'écran.

4Ø et 6Ø Suppression du bip clavier et du curseur.

10ØØ Début de boucle de jeu : tirages aléatoires et affichages des combinaisons.

13Ø Tirage aléatoire du nombre de tirs.

14Ø Boucle d'affichage des combinaisons qui s'affichent pendant que le dé « roule ».

19Ø Tirage proprement dit.

27Ø Aiguillage vers les dessins des différentes « faces » du dé en fonction du nombre aléatoire R.

28Ø CURSET positionne le curseur en mode HIRES.

29Ø CHAR permet de placer des caractères à l'endroit positionné par CURSET. 95 est le code ASCII du caractère alterné, 1 détermine une écriture inversée, A initialisé à 1 en 16Ø et à Ø en 2ØØØ détermine l'effet « couleur de premier plan » et « couleur de fond » qui permet l'effacement entre deux tirages. Jusqu'en 42Ø sont dessinées les différentes faces du dé.

45Ø Positionnement de la fenêtre qui représente la face du dé.

46Ø FILL dessine une fenêtre carrée blanche.

48Ø et 49Ø Coloration du dé (fond bleu, encre jaune) qui est ensuite inversée par CHAR (fond jaune, encre bleue, cf. 29ØØ).

J'APPRENDS LES OPÉRATIONS

Un ordinateur à la maison peut être un répétiteur plein de patience pour vos enfants. Pourquoi ne pas leur proposer de réviser leurs opérations avec Oric ?

Déroulement

Voici un programme d'apprentissage de l'addition. Son fonctionnement est si simple que vos enfants pourront l'utiliser seuls. La présentation de l'opération est la plus courante. Ils rentreront leurs résultats chiffre par chiffre. Le curseur se placera au bon endroit.

```
1 REM
2 REM
3 REM  J'APPRENDS LES OPERATIONS
4 REM
10 /Initialisations
20 RJ=0:PRINT:PRINT
30 CLS:PAPER4:INK6
40 GOSUB2000
50 PRINT"Combien veux-tu d'additions?";
60 INPUT NA:PRINT
70 PRINT"Combien de chiffres (1,2,3)";
80 INPUT NCH:PRINT
90 PRINT"Tapé sup ESPACE      Pour commencer"
100 X=RND(1):IF KEY$<>" "THEN 100
110 REM
120 REM  Boucle Principale
121 REM
130 CLS
140 FOR I=1 TO NA
150 CLS
160 N1=INT(RND(1)*(10^NCH-1))+1
170 N2=INT(RND(1)*(10^NCH-1))+1
180 PRINT@0,6) " "N1
190 PRINT@0,8) "+" :PRINTN2
200 PRINT@9,10) "----"
210 PRINT@0,12) "= " :GOSUB 1000
```

```
220 PRINT:PRINT
230 IF VAL(B$)=N1 +N2 THEN 240 ELSE 250
240 RJ=RJ+1:PRINT"BIEN":GETA$:GOTO 260
250 PRINT"NON":GET A$
260 NEXT
299 REM
300 REMScore
301 REM
310 CLS:PRINT:PRINT:PRINT RJ;
   "RePonseS Justes"
320 PRINT:PRINT"Pour ";NA; " additions"
400 PRINT:PRINT:PRINT:PRINT
   "Veux-tu continuer(O/N)"
410 A$=KEY$:IF A$=""THEN410
420 IF A$="O" THEN 10 ELSE 430
430 CLS:PRINT"A bientôt"
999 REM
1000 REM Affichage rePonse
1001 REM
1010 B$=""
1020 A$=KEY$:IF A$<>" "THEN1020
1030 B$=""
1040 A$=KEY$:IF A$<>" "THEN 1040
1050 A$=KEY$:IF A$=""THEN 1050
1060 IF A$=CHR$(13)THEN RETURN
1070 IF ASC(A$)<48 OR ASC(A$)>57 THEN 1050
1080 B$=A$+B$
1090 PRINT CHR$(9)+A$+CHR$(8)+CHR$(8)
   +CHR$(8);
1100 GOTO 1040
2000 PRINT@5,12;"ADDITIONS"
2010 GET A$
2020 RETURN
```

Commentaires

20 Initialisation du compteur de réponses justes.

50 Initialisation de la boucle principale (nombre d'opérations).

100 X=RND(1) Réinitialisation de random tant que le clavier n'est pas sollicité.

16Ø-17Ø Tirage aléatoire des deux opérateurs.

18Ø-22Ø Affichage de l'opération.

23Ø-25Ø Test de validité de la réponse.

1ØØØ-1Ø8Ø Gestion du curseur pour l'affichage des réponses de la droite vers la gauche.

1Ø8Ø Concaténation de la réponse sous forme de chaîne.

1Ø9Ø Affichage de la réponse avec recul à gauche.

Bloc-notes

La combinaison de RND(1) avec une lecture du clavier par KEY\$ est à retenir pour le renouvellement de l'aléatoire.

Sur Oric 1, PLOT remplace PRINT@, avec obligation d'utiliser STR\$ pour l'affichage des variables numériques.

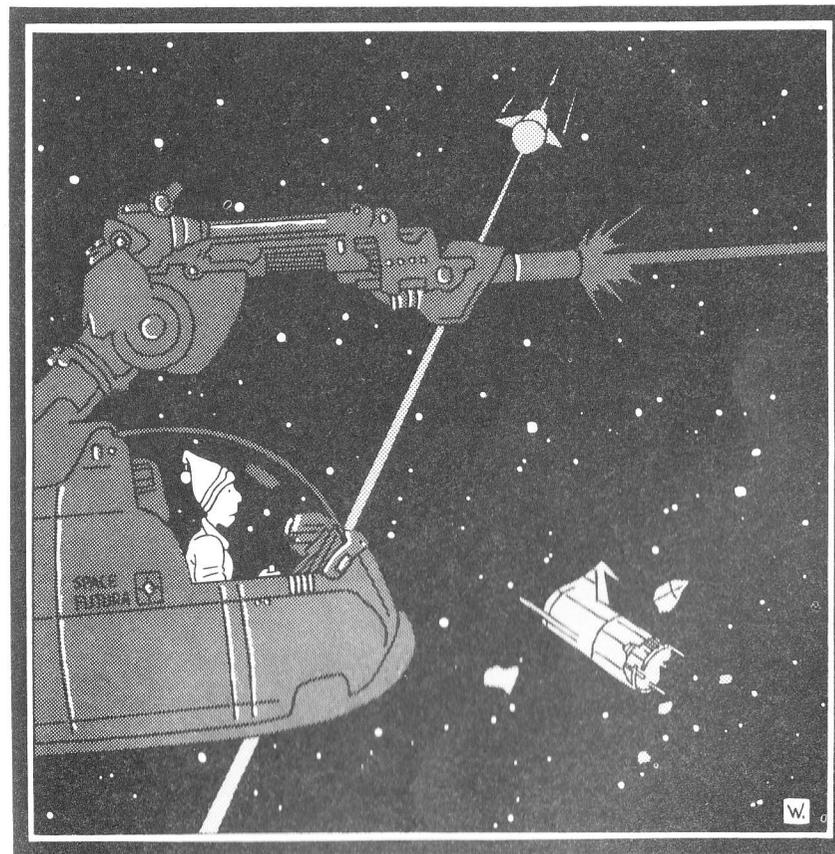
Exemple :

PLOT 1Ø,1Ø,STR\$(A). Si A=8, affiche huit aux coordonnées 1Ø,1Ø. Pour les chaînes de caractères, il n'y a pas besoin de STR\$.

Exemple : PLOT 1Ø,1Ø, « Bonjour » affiche « Bonjour » en 1Ø,1Ø.

ACTION

CHAMP DE FORCE



L'artillerie lourde, ça vous connaît ? Ajoutez-y beaucoup d'adresse et de la stratégie. Cela vous intéresse ? Alors tapez CHAMP DE FORCE : vous ne le regretterez pas.

Déroulement

Au-dessus de vous défile une armada de martiens belliqueux autant que verts. Votre mission consiste à les désintégrer à l'aide de votre canon à protons. Seulement voilà...une barrière magnétique, ultra-résistante, s'interpose entre vous et l'objectif. Il vous faudra donc détruire, morceau par morceau, ce champ de force qui ne cesse de tourner sur lui-même et dont les brèches se déplacent sans arrêt. Coup d'œil et astuce sont les seuls partenaires sur lesquels vous devrez compter.

Pour vous déplacer, vous disposez de deux touches : < pour aller vers la gauche, > vers la droite. La touche Z vous permettra de tirer.

Le temps vous est mesuré pour mener à bien cette mission. Agissez vite si vous voulez réussir à détruire toute l'armée en temps voulu. Le bonus s'additionnera à votre score. Mais attention, votre prochaine mission n'en sera que plus dure. Cela ne saurait vous impressionner, n'est-ce pas ? Gare aux traînants qui laisseront le bonus arriver à zéro.

Structure

1φ-2φ	Chargement des codes machine et des caractères
3φ-6φ	Initialisations
1φφ-17φ	Boucle centrale
2φφ-22φ	Déplacement à droite du vaisseau
3φφ-32φ	Déplacement à gauche du vaisseau
5φφ-55φ	Martien touché
6φφ	Perdu
1φφφ-121φ	Langage machine
2φφφ-2φ7φ	Caractères

```
1 REM
2 REM
3 REM CHAMP DE FORCE
4 REM
5 TEXT:CLS:POKE#26A,10:PAPER0:INK2
10 FORR=0T040:READA:POKE#400+R,A:NEXT
20 FORR=0T063:READA:POKE#B508+R,A:NEXT
```

```
29 REM INITIALISATIONS
30 AB=#BE2A:SC=0:AM=#BC9A:TC=0
35 DOKE4,AM:DOKE6,AM-1
40 AV=#BFCC:T=0
45 POKEAV,33:DOKE0,AB:DOKE2,AB-1
50 FORR=AB-1T0AB+37:POKER,34:NEXT
55 FORR=AMT0AM+35STEP5:POKER,40:NEXT
60 DOKE#276,20000
70 PRINT@0,0;"SCORE:";SC:PRINT@23,0;"BONUS:"
100 BO=DEEK(#276):IFBO>60000THEN600
102 PRINT@30,0;BO
105 IFPEEK(#208)=#94THEN200
110 IFPEEK(#208)=#8CTHEN300
115 IFT=1THEN135
120 IFPEEK(#208)=#AATHENT=1:AT=AV-40:SHOOT
130 IFT=0THENCALL#400:GOTO100
135 IFAT<#BBF8THENPOKEAT,32:T=0:GOTO130
140 POKEAT,32:CALL#400:AT=AT-40
150 TS=PEEK(AT)
160 IFTS>33ANDTS<38THENPOKEAT,TS+1:T=0:
EXPLODE:GOTO100
165 IFTS=40THEN500
170 POKEAT,39:GOTO100
200 IFAV=#BFDETHEN130
210 POKEAV,32:AV=AV+1:POKEAV,33
220 GOTO130
300 IFAV=#BFBATHEN130
310 POKEAV,32:AV=AV-1:POKEAV,33
320 GOTO130
500 SC=SC+100:PRINT@7,0;SC
510 T=0:EXPLODE8POKEAT,32:TC=TC+1
520 IFTC<8THEN100
530 SC=SC+DEEK(#276):PRINT@7,0;SC
540 AB=AB-40
550 TC=0:POKEAV,32:GOTO35
600 PRINT@31,0;"0":PLOT18,10,"PERDU":END
1000 DATA#A0,#00 'LDY#50
1010 DATA#B1,#02 'LDA(#02),Y
1020 DATA#8D,#FF,#04 'STA#4FF
1030 DATA#B1,#00 'LDA(#00),Y
1040 DATA#91,#02 'STA(#02),Y
1050 DATA#C8 'INY
1060 DATA#C0,#26 'CPY##26
1070 DATA#D0,#F7 'BNE#407
```

```

1080 DATA#AD,#FF,#04 'LDA#4FF
1090 DATA#91,#02 'STA#( #02 ),Y
1110 DATA#A0,#25 'LDY##25
1120 DATA#B1,#04 'LDA( #04 ),Y
1130 DATA#8D,#FF,#04 'STA#4FF
1140 DATA#B1,#06 'LDA( #06 ),Y
1150 DATA#91,#04 'STA( #04 ),Y
1160 DATA#88 'DEY
1170 DATA#D0,#F9 'BNE#41C
1190 DATA#AD,#FF,#04 'LDA#4FF
1200 DATA#91,#04 'STA( #04 ),Y
1210 DATA#60 'RTS
2000 DATA30,12,12,45,33,43,63,30
2010 DATA63,63,63,63,63,63,63,63
2020 DATA63,63,47,63,61,55,43,1
2030 DATA63,47,63,45,7,18,0,0
2040 DATA63,58,14,20,0,8,0,0
2050 DATA0,0,0,0,0,0,0,0
2060 DATA8,4,8,6,24,14,28,0
2070 DATA30,51,63,18,33,33,18,18

```

Commentaires

- 5 POKE #26A,10 Suppression du bip et du curseur.
- 10-20 Chargement du langage machine et des caractères.
- 30-45 Initialisation de variables.
- 50-55 Affichage du mur et des martiens.
- 60 Mise à 20 000 du bonus.
- 70 Affichage du score et du bonus.
- 100 Début de boucle centrale : teste si le bonus est égal à 0.
- 102 Affichage du bonus.
- 105-120 Test clavier.
- 130 Si aucun missile n'est envoyé, on déplace mur et martiens et on fait le retour de boucle.
- 135 Si le missile est arrivé en haut de l'écran, on remet le témoin de tir à zéro.

140-170 Déplacement du missile, du mur et des martiens : test pour savoir si le missile a atteint son objectif.

200-220 Déplacement du vaisseau et test d'arrivée en bout d'écran.

300-320 Idem.

500-520 Martien touché, on ajoute 100 au score et on efface la victime. S'il reste encore des martiens on retourne dans la boucle principale.

530-550 Tous les martiens sont éliminés, le bonus est additionné au score : le champ de force monte.

600 Perdu.

1000-2070 DATA du langage machine et des caractères.

Bloc-notes

Il n'y a pas de routine de décompte du bonus à proprement parler, il se fait tout seul. Pour cela, on utilise des adresses #276 et #277 qui forment, à elles deux, un compteur interne à la machine. Pour l'utiliser, il suffit de taper DOKE #276, X. A partir de ce moment, ce qui se trouve en #276 se décomptera tout seul et pourra être testé avec DEEK (#276).

DESCENTE A SKIS

L'air est vif et la montagne belle. Quelques mouvements d'échauffement pour vaincre la morsure du froid. A vos pieds, à une distance indéterminée, la vallée où flotte une légère brume. Clac ! les fixations bloquent vos chaussures montantes. Vous vous élanchez d'un coup de rein. Les skis déchirent une neige de soie.



Déroulement

A vous de choisir votre piste : de la bleue (1) à la noire (5), selon votre force. Une fois engagé, impossible de revenir. Ne surestimez

pas votre dextérité. Touche > pour obliquer à droite, touche < pour virer à gauche. Vous dévalez, dans un chuintement de vos « planches », entre deux rangées de sapins. Gare à ne pas en rencontrer un ! La moindre collision vous met immédiatement sur la civière, ambiance SAMU garantie. Combien de temps pourrez-vous résister à l'ivresse de la descente ?

Structure

1ø	Regénération des caractères
1øø-13ø	Définition des caractères
14ø-21ø	DATA caractères
25ø-38ø	Définition des variables
4øø-45ø	Boucle centrale de jeu
3øøø-311ø	Choix des pistes
4øøø-5ø7ø	Perdu

```
1 REM
2 REM
3 REM      * DESCENTE A SKI *
4 REM
5 REM
10 CALL#F8D0
20 GOSUB30,10
30 CLS
100 FORR=0T063
110 READA
120 POKE#B508+R,A
130 NEXT
132 REM
133 REM      * CARACTERES      *
134 REM
140 DATA0,28,14,31,15,1,2,13
150 DATA0,16,41,18,44,8,16,32
160 DATA2,2,19,19,23,26,18,2
170 DATA16,16,50,50,58,38,18,16
180 DATA0,2,37,16,13,4,2,1
190 DATA20,14,28,63,60,32,16,44
200 DATA0,1,15,1,31,1,63,1
210 DATA0,32,56,32,60,32,63,32
```

```

212 REM
213 REM      * INITIALISATION *
214 REM
250 CLS:PAPER7:INK0:POKE#26A,10
260 SK(1)=#2221:SK(2)=#2423:SK(3)=#2625
270 FORR=1T05
280 E#="":LE=1+2*R
290 FORE=1TOLE:E#=E#+ " ":NEXTE
300 P$(R)="'( "+E#+"'('
310 NEXTR
340 CLS
350 FORR=1T026
360 PRINTTAB(15);P$(D)
370 NEXT
380 PP=15:PS=#BCAC:DS=0:PLAY0,7,0,0
391 REM
392 REM      *      JEU      *
393 REM
400 AZ=INT(RND(1)*3)
401 FORR=1T03
420 DP=AZ-1+PP
430 IF DP=2 OR DP=23THEN445
440 PP=PP+AZ-1
445 DOKEPS,#2020
450 PRINTTAB(PP);P$(D)
460 DS=0:B=5
470 IF PEEK(#208)=#94 THEN DS=1:B=3
480 IF PEEK(#208)=#8C THEN DS=-1:B=3
490 PS=PS+DS
499 REM
500 IFDEEK(PS)<>#2020THEN 4000
510 DOKEPS,SK(2+DS)
520 SOUND4,B,5
530 GOTO400
3001 REM
3002 REM      * CHOIX NIVEAU *
3003 REM
3010 PAPER 7:INK0
3020 CLS:PRINT
3030 PRINT SPC(10);"DESCENTE A SKI"
3040 FOR I=1 TO 5:PRINT:NEXT I
3050 PRINT"Vous etes sur une Piste de ski"
3060 PRINT"Quelle Pente choisissez-vous"
3070 FOR I=1 TO 10:PRINT:NEXT I

```

```

3080 PRINT"De 1 (Piste bleue)
3090 PRINT" a 5 (Piste noire)
3095 GETD$:D=VAL(D$)
3100 IF D<1 OR D>5 THEN 3095
3110 D=6-D
3120 RETURN
3991 REM
3992 REM      * FIN DE JEU *
3993 REM
4000 CLS:EXPLODE
4005 FORR=10T014:PLOT20,R,17:PLOT21,R,23:NEXT
4006 FORR=18T022:PLOTR,12,17:NEXT
4007 PLOT23,12,23
4010 FORR=1T03
4015 PLAY7,0,0,0
4020 MUSIC1,3,4,10:WAIT10:MUSIC1,2,12,10
4030 WAIT10:MUSIC1,3,4,10:WAIT10
4040 PLAY0,0,0,0:WAIT100
4050 NEXT
4500 CLS
4510 PRINT"Vous etes dans les sapins!"
5000 PRINT CHR$(17)
5010 PRINT"Une autre descente (O/N)?"
5015 CALL#F8D0:WAIT50
5020 GETA$:GETA$
5030 IF A$="O" THEN RUN
5040 INK 0
5050 PAPER 7
5070 END

```

Commentaires

10 Sur Oric 1, la ligne 10 doit s'écrire : CALL #F89B. Faire la même modification en 5015. L'adresse #F8D0 sur Atmos (#F89B sur Oric 1) recharge le clavier standard lorsqu'on l'appelle à l'aide d'un CALL.

20 Appel du sous-programme de choix des pistes.

100 Boucle de définition des caractères.

120 La boucle lit les DATA et les met en mémoire à l'aide d'un POKE. L'adresse du POKE est calculé ainsi : #B400 + 8 * code ASCII du premier caractère à modifier (ASC(X)).

25) Pour éliminer le bip clavier et le curseur, il suffit de taper POKE #26A,1. L'avantage sur PRINT CHR\$(6) et CHR\$(13) est qu'il est définitif.

26) SK(1) correspond au graphisme du skieur tournant à gauche. La valeur qui suit correspond à la valeur du code ASCII du caractère concerné. Comme il faut deux caractères pour dessiner un skieur, il y a deux codes ASCII qui sont #22 et #21, soit respectivement 34 et 33 en décimal. Comme on utilise DOKE au lieu de deux POKE, #22 et #21 deviennent #2221. DOKE adresse #2221 = POKE adresse #21 et POKE adresse + 1, #22.

4) AZ est une fonction aléatoire créée à l'aide de RND(1) : elle tire au hasard une valeur comprise entre 0 et 2 qui déterminera dans quel sens évoluera la piste de ski.

47) Les touches de contrôle sont testées à l'aide de PEEK(#208) qui renvoie une valeur en fonction de la touche enfoncée. Les touches de commandes choisies sont < pour aller à gauche et > pour aller à droite, mais elles peuvent être modifiées.

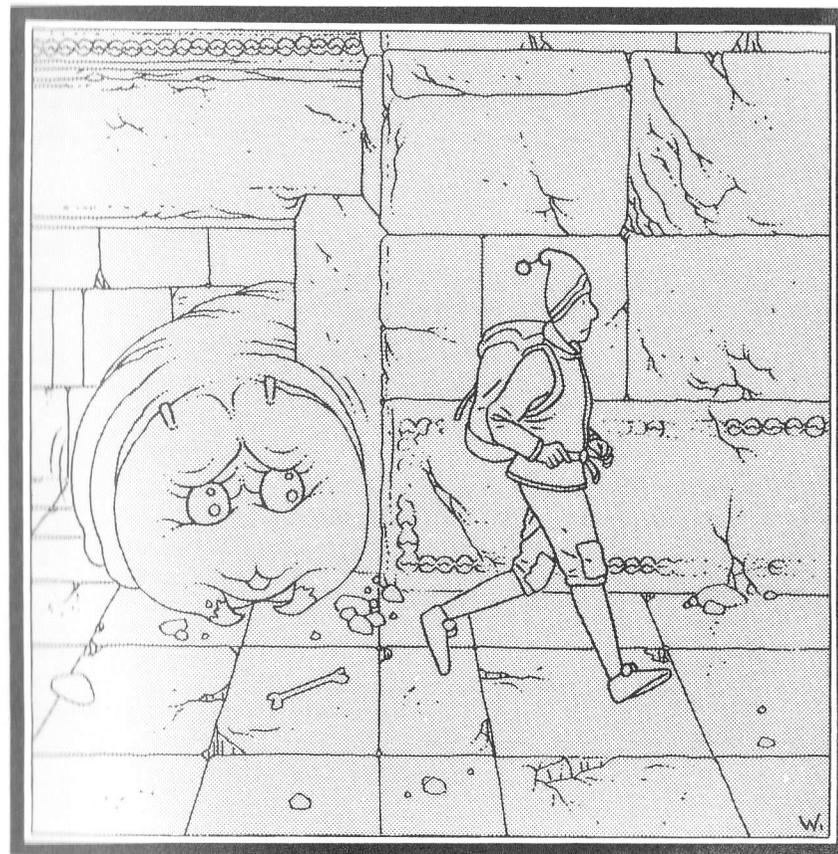
5) DEEK(PS) regarde sur l'écran à l'emplacement du skieur et détermine s'il y a eu une collision. Si aucune collision n'a eu lieu, la valeur renvoyée sera #202. En effet, #20 (32 en décimal) correspond au code ASCII d'un espace, c'est-à-dire rien sur l'écran.

Bloc-notes

CALL#F8D (#F89B sur Oric 1) pourra resservir dans tous les programmes où l'on aura besoin de reprendre les caractères standards.

L'emploi de TAB sur Oric 1 est délicat : en raison d'une erreur du Basic dans la ROM, toutes les valeurs en TAB inférieures à 13 ne sont pas prises en compte. Il faut donc marquer + 13 à l'intérieur du TAB. Exemple : TAB(N)=TAB(13+N) sur ORIC 1.

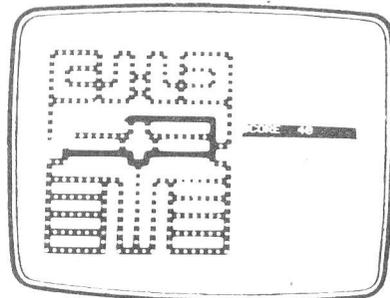
CHENILLE ET LABYRINTHE



Horreur et damnation, vous voilà enfermé dans un labyrinthe sans issue, condamné à finir vos pauvres jours dans ce réduit. A moins que... à moins que l'ingurgitation forcée de pilules égrenées sur votre chemin ne vous offre un sursis inespéré mais bien illusoire. Car à mesure que le temps s'écoule, votre corps s'allonge sans relâche, paralysant peu à peu vos mouvements. Seul moyen de retrouver votre taille svelte : absorber toutes les pilules se trouvant dans le labyrinthe. Vous entrez alors dans un autre labyrinthe où le temps s'écoule encore plus vite, votre embonpoint augmentera d'autant et la lutte pour la sveltesse par hyper-absorption n'en sera que plus démente. Quel cruel destin pour une créature issue du hasard !

Déroulement

Le curieux et terrifiant dessin d'un labyrinthe emprisonnant une chenille apparaît sous vos yeux émerveillés. Inutile de vous réjouir, la chenille c'est vous ! Misérable créature. Vous voici bel et bien piégé. Pour mouvoir votre embarrassante personne, quatre touches sont à votre disposition : pour aller à gauche <, à droite >, en haut A, en bas Z. Vous engloutissez tout ce qui se trouve sur votre passage, y compris vous-même, mais là ça fait mal : un *shoot* retentissant annonce à tout votre entourage que vous avez commis la *faute* fatale ! Sur la droite de l'écran, vous aurez la maigre consolation de prendre connaissance de votre performance. Un point par pilule absorbée. Qui sera le roi de la pilule ?



```
0 ' CHENILLE
1 TEXT:POKE#26A,10:CLS
2 H=#AE:B=#AA:D=#94:G=#8C
3 Q=101
4 SC=0:P=0:TB=1
5 GOSUB900
60 L=100
65 GOSUB2000:'Affiche labyrinthe
66 '
70 FORR=0TO5:POKE#BDB6+R,97:NEXT:AQ=#BDB5:
   AT=#BDEC
80 POKEA,103:POKEAT,100
82 '
83 'Boucle principale
84 '
85 IFTP<0THENTP=L:GOTO140
90 POKEA,32
```

```
100 IFPEEK(AQ-1)=97THENAQ=AQ-1:Q=1:GOTO135
110 IFPEEK(AQ+1)=97THENAQ=AQ+1:Q=2:GOTO135
120 IFPEEK(AQ-40)=97THENAQ=AQ-40:Q=3:GOTO135
130 IFPEEK(AQ+40)=97THENAQ=AQ+40:Q=4
135 IFQ=101THENQ=105ELSEQ=101
136 POKEA,Q+Q
140 A=AT
150 T=PEEK(#208)
155 TP=TP-1
160 IFT=H THENAT=AT-40:T=1:GOTO200
170 IFT=B THENAT=AT+40:T=2:GOTO200
180 IFT=D THENAT=AT+1:T=3:GOTO200
190 IFT=G THENAT=AT-1:T=4:GOTO200
195 GOTO150
200 TR=PEEK(AT):IFTR>64THEN500
210 IFTR=33THENSC=SC+1:P=P+1:
   PLOT33,10,STR$(SC):PLAY1,0,1,200:
   SOUND1,500,0
220 IFTR=64THENAT=A:GOTO150
230 POKEAT,97+T:POKEA,97
250 IFF=308THEN600
260 GOTO85
265 '
266 'Perdu
267 '
500 PAPER5:PAPER0:PAPER5:PAPER0:EXPLODE
510 POKE#BD93,3:PLOT12,12,"PERDU":POKE#BD99,6
520 IFPEEK(#208)<>#38THEN520
530 GETA#:END
595 '
596 'Tableau suivant
597 '
600 WAIT150:CLS:TB=TB+1:
   A$="TABLEAU No "+STR$(TB):
610 PLOT10,12,A$:WAIT200:CLS:P=0
620 L=L-10:GOTO65
697 '
698 ' data
699 END
900 FORR=#B508TO#B50F:READA:POKEA,A:NEXT
910 FORR=0TO103
920 READA:POKE#B708+R,A:NEXT
930 FORR=0TO7:POKE#B600+R,63:NEXT
950 RETURN
```

```

998 ' data
999 '
1000 DATA0,0,0,12,12,0,0,0,0,
      12,30,63,63,30,12,0
1010 DATA0,12,12,30,30,63,30,0,0,
      30,63,30,30,12,12,0
1020 DATA0,16,30,63,63,30,16,0,
      0,2,15,63,63,15,2,0
1030 DATA0,0,16,56,60,18,1,0,
      0,32,18,15,7,2,0,0
1040 DATA0,12,30,12,8,16,32,0,
      0,1,2,4,12,30,12,0
1050 DATA0,1,18,60,56,16,0,0,
      0,0,2,7,15,18,32,0
1060 DATA0,12,30,12,4,2,1,0,0,
      32,16,8,12,30,12,0
2000 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
2001 PRINT"@!!!!!!!!@!!!!@!!!!@!!!!!!!"
2002 PRINT"@!@@@@@!@!@!@!@!@@@@@!"
2003 PRINT"@!@!!!!!@!@!@!@!!!!!@!"
2004 PRINT"@!@!@@@!@!@! !@!@!@!@!"
2005 PRINT"@!@!!!!!!!@!@!@!!!!!!!"
2006 PRINT"@!@@@@@!@@@!!!@@@!@@@@@!"
2007 PRINT"@!!!!!!!!!!!!@!!!!!!!"
2008 PRINT"@!@@@@@@@@@@@@@@@@@@@@@@@@@!"
2009 PRINT"@!@!!!!!!!"
2010 PRINT"@!@!@@@@@@@!@!@@@@@@@!@!"
2011 PRINT"@!@!!!!!!!"
2012 PRINT"@!!!@@@@@!@@@!@@@@@@!!!"
2013 PRINT"@@@!!!!!!!!"
2014 PRINT"@!!!@@@@@@@!!!@@@@@@@!!!"
2015 PRINT"@!@@@@@@@@@@@!@@@@@@@@@@@!"
2016 PRINT"@!!!!!!!!!!!!@!!!!!!!"
2017 PRINT"@!@@@@@@@!@!@!@!@@@@@@@!"
2018 PRINT"@!!!!!!!!!!!!@!@!@!!!!!!"
2019 PRINT"@!@@@@@@@!@!@!@!@@@@@@@!"
2020 PRINT"@!!!!!!!!!!!!@!@!@!!!!!!"
2021 PRINT"@!@@@@@@@!@!@!@!@@@@@@@!"
2022 PRINT"@!!!!!!!!!!!!@!@!@!!!!!!"
2023 PRINT"@!@@@@@@@!@!@!@!@@@@@@@!"
2024 PRINT"@!!!!!!!!!!!!@!!!!!!!"
2025 PRINT"@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@"
2030 PLOT27,10,"SCORE:"
2040 FORR=#BB80TO#BBAS:POKER,32:NEXT

```

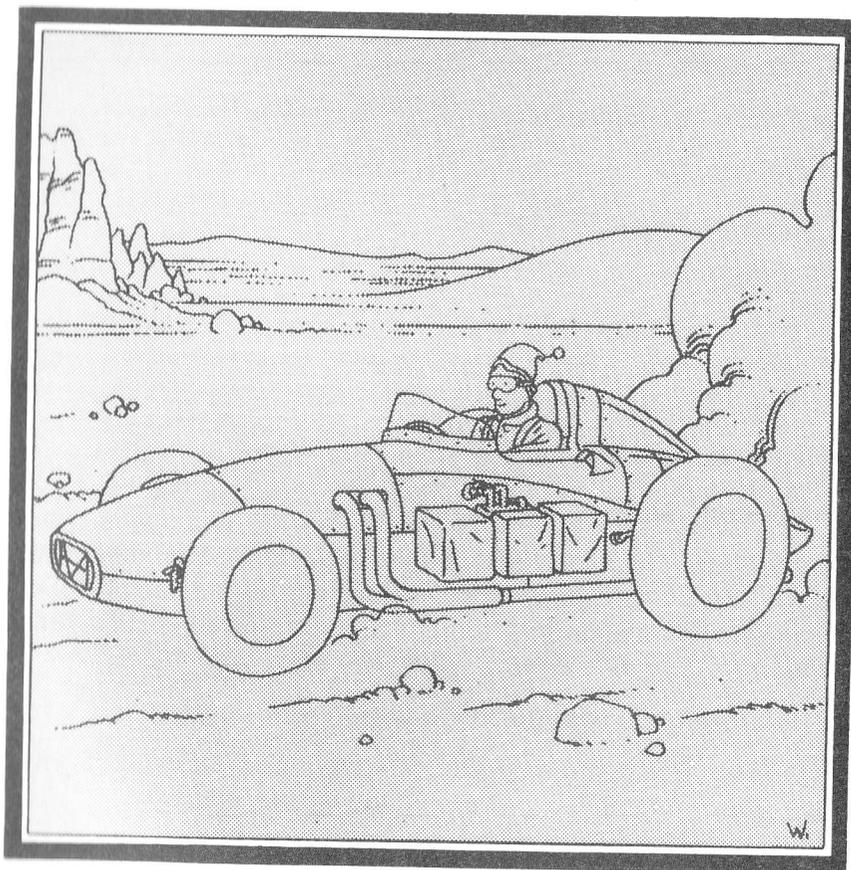
```

2050 INK6:PAPEP0
2060 RETURN

```

Commentaires

- 1 POKE #26A,1Ø Élimine le curseur et le bip clavier.
- 2 Affectation d'une touche à chaque déplacement.
- 5 Définition des caractères.
- 65 Affichage du labyrinthe.
- 7Ø Affichage de la chenille.
- 8Ø Affichage de la tête et de la queue.
- 85 Début de boucle centrale : si PT < Ø, la chenille s'allonge d'un maillon.
- 9Ø Efface la queue.
- 1ØØ-136 Réaffiche la queue à la place du dernier maillon.
- 15Ø-195 Test clavier.
- 2ØØ-22Ø Test écran : on regarde sur l'écran ce que la chenille va manger. S'il n'y a rien, on avance simplement; s'il y a une pilule, on la mange et on incrémente le score; s'il y a un bout de chenille, on a perdu.
- 23Ø Réaffichage de la tête dans sa nouvelle position.
- 25Ø On teste le nombre de pilules mangées sur les tableaux. Si elles sont toutes avalées, on passe au tableau suivant.
- 26Ø Retour de boucle.
- 5ØØ-53Ø Perdu.
- 6ØØ-62Ø Tableau suivant : on raccourcit le délai au bout duquel la chenille s'allonge. Retour à un nouveau tableau.
- 9ØØ-95Ø Lecture des DATA et initialisation du caractère.
- 1ØØØ-1Ø6Ø DATA des caractères.
- 2ØØØ-2Ø25 Définition du labyrinthe.
- 2Ø4Ø Effacement du CAPS.



Soleil radieux, odeur d'essence, tonnerre déchirant la rumeur de la foule. Phares trouant la nuit dans un jaillissement d'eau de pluie. Pilote porté en triomphe, fontaine de champagne. Qui n'a jamais souhaité en être ? Cette course d'endurance est l'une des plus belles qui soient. Cette simulation vous en donnera l'avant-goût. Et qui sait, au printemps prochain... En attendant, prenez le départ au volant de votre Orié-Atmos. Soyez sûr qu'il mettra vos nerfs et vos réflexes à l'épreuve.

Déroulement

Dès le départ, votre formule 1 dévore la piste dans un vrombissement infernal. Au bout de la première ligne droite, déjà les premiers virages. Sachez anticiper les circonvolutions du circuit qui défile à une vitesse endiablée. Touche > pour virer à droite, touche < pour obliquer à gauche. Attention, à mesure que la course se déroule, la piste se rétrécit alors que votre véhicule accélère ! Il s'agit donc bien d'une course d'endurance dans laquelle le moindre accrochage vous sera fatal. Qui parcourra la plus longue distance ?

```

0 '24 HEURES DU MANS
1 DATA#A9,#BA,#85,#02,#A9,#92,#85,#00
2 DATA#A9,#BF,#85,#01,#85,#03,#A2,#19
3 DATA#A0,#00,#B1,#00,#91,#02,#C8,#C0
4 DATA#26,#D0,#F7,#38,#A5,#00,#85,#02
5 DATA#E9,#28,#85,#00,#A5,#01,#85,#03
6 DATA#E9,#00,#85,#01,#CA,#D0,#E1,#A0
7 DATA#00,#A9,#20,#99,#D2,#BB,#C8,#C0
8 DATA#26,#D0,#F8,#60
9 '      CHARGE CODE MACHINE
10 FORR=0TO#3B:READA:POKE#400+R,A:NEXT
11 DATA#01,#0D,#0F,#0D,#01,#1D,#1F,#1D
12 DATA#20,#2C,#3C,#2C,#20,#2E,#3E,#2E
13 '
14 'CHARGE CARACTERES
15 '
16 FORR=0TO15:READA:POKE#B508+R,A:NEXT
17 FORR=0TO7:POKE#B518+R,59:POKE#B520+R,55:
NEXT
18 '
19 '      DEFINI 3 LARGEURS DE ROUTE
20 '
21 R$(1)="#          $"
22 R$(2)="#          $"
23 R$(3)="#          $"
24 '
25 'INITIALISATION
26 '
27 CLS:POKE#26A,10:TEXT:PAPER0:INK6
28 FORR=0TO3:POKE#BBA7-R,32:NEXT
29 RT=12:N=0:L=1
    
```

```

100 KM=0:EXPLODE
110 FORR=1T026:PLOT12,R,R#(1):NEXT
120 DOKE#BE10,#2221:V=#BE10
127 '
128 ' BOUCLE PRINCIPALE
129 '
130 AZ=INT(RND(1)*3)
140 IFAZ=1THEN180
150 IFAZ=0THEN200
160 IFRT=29THEN200
170 RT=RT+1:GOTO200
180 IFRT=2THEN200
190 RT=RT-1
200 DOKEY,#2020
210 CALL#400
220 IFPEEK(#208)=#94THENV=V+1
230 IFPEEK(#208)=#80THENV=V-1
240 IF DEEK(V)<>#2020THEN300
250 DOKEY,#2221
260 PLOTRT,1,R#(L)
270 PLAY0,1,3,35
280 KM=KM+1:P=P+1:IFP=100THEN400
290 GOTO130
297 '
298 ' PERDU
299 '
300 EXPLODE
310 FORR=0T010:PAPER7:PAPER0:NEXT:INK3
315 WAIT100
320 CLS:PLOT16,15,"PERDU"
330 PRINT"VOUS AVEZ PARCOURU ":KM/100:
    "KILOMETRE."
340 END
400 P=0:N=N+1:IFN>2THEN430
410 L=L+1:RT=RT+1
420 GOTO200
430 DOKEY,#2020:V=V-40
440 GOTO130

```

Code machine

```

1400-43E
0400: A9 BA LDA #BA
0402: 85 02 STA #02

```

```

0404: A9 92 LDA #92
0406: 85 00 STA #00
0408: A9 BF LDA #BF
040A: 85 01 STA #01
040C: 85 03 STA #03
040E: A2 19 LDX #19
0410: A0 00 LDY #00
0412: B1 00 LDA ($00),Y
0414: 91 02 STA ($02),Y
0416: C9 INY
0417: C0 26 CPY #26
0419: D0 F7 BNE $0412
041B: 38 SEC
041C: A5 00 LDA #00
041E: 85 02 STA #02
0420: E9 28 SBC #28
0422: 85 00 STA #00
0424: A5 01 LDA #01
0426: 85 03 STA #03
0428: E9 00 SBC #00
042A: 85 01 STA #01
042C: CA DEX
042D: D0 E1 BNE $0410
042F: A0 00 LDY #00
0431: A9 20 LDA #20
0433: 99 D2 BB STA $BBD2,Y
0436: C9 INY
0437: C0 26 CPY #26
0439: D0 F8 BNE $0433
043B: 60 RTS

```

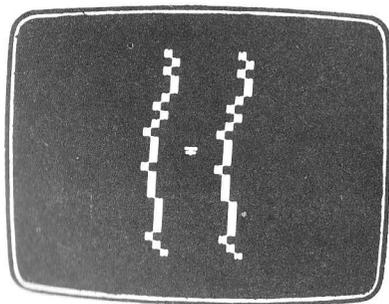
Commentaires

- 1) Met les codes machine en mémoire.
- 2) et 3) Définition des graphismes.
- 4)-6) Mise en tableau des différentes configurations de la route.
- 7) Le POKE #26A,1) élimine le curseur et le bip clavier.
- 8) Elimination du CAPS en haut et à droite.
- 11) Affichage de la ligne droite de départ.

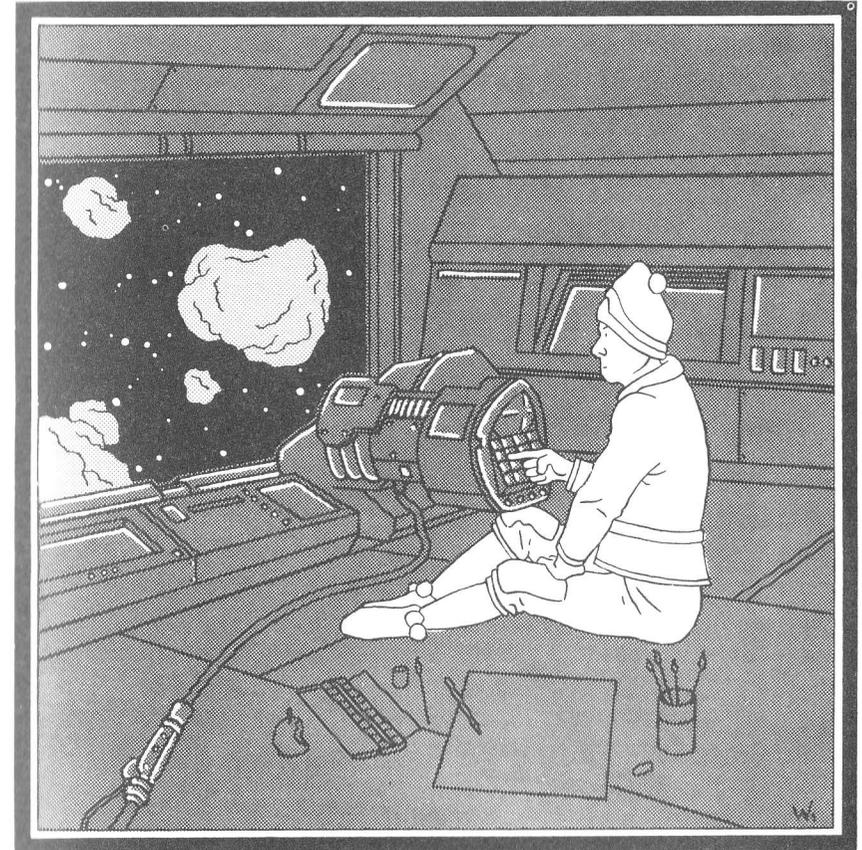
- 12) Positionnement de la voiture sur la ligne de départ.
- 13) Tirage d'une valeur aléatoire pour faire évoluer la route.
- 20) Effacement de la voiture.
- 21) Défilement de la route.
- 22-25) Réaffichage de la voiture dans sa nouvelle position.
- 26) Affichage d'une nouvelle portion de route.
- 27) Bruit du moteur.
- 28) Décompte des kilomètres parcourus et évolution des difficultés.
- 30-34) Perdu et bilan.
- 40) Nouvelles difficultés : rétrécissement de la route et montée de la voiture vers le haut de l'écran.

Bloc-notes

La routine de défilement de l'écran, donnée en DATA en début de programme, peut resservir. En voici le mode d'emploi : insérez les DATA et la boucle de la ligne 1) dans votre programme. Une fois les codes machine mis en mémoire, il suffit de faire CALL#4) pour faire défiler l'écran d'un caractère vers le bas.



CHAMP DE MÉTÉORES



Aux commandes de votre vaisseau, le signal d'alerte vient de se déclencher : une zone de météorites est en vue. Impossible de la contourner. Diable ! Aussi vous trouvez-vous projeté dans un chaos de pierres venant de toutes parts, frôlant votre frêle esquif, du moins tant que votre vigilance ne sera pas prise en défaut. Ceux qui reviendront vivants pourront témoigner...

Déroulement

Après RUN, le vrombissement des réacteurs se fait entendre. L'écran se couvre d'une horde aussi sauvage que minérale ! Et plus on avance et plus il y en a ! Le chemin à frayer est de plus en plus étroit et périlleux. Pour guider votre vaisseau et lui éviter toute fatale collision, vous disposez de deux touches : > pour aller à droite, < pour obliquer à gauche. Dès qu'une pierre percute le vaisseau, votre mission s'arrête là, dans un coin perdu de l'espace. Terrible destin. Il ne vous reste plus qu'à faire le bilan de vos points. Maigre consolation...

Structure

1-8	Langage machine
10	Chargement des codes machine
11-14	Caractères
20	Chargement des caractères
70-120	Initialisation des variables
230-290	Boucle centrale
300-400	Perdu et bilan

```
0 ' CHAMP DE METEORITES
1 DATA#A9,#BA,#85,#02,#A9,#92,#85,#00
2 DATA#A9,#BF,#85,#01,#85,#03,#A2,#19
3 DATA#A0,#00,#B1,#00,#91,#02,#C8,#C0
4 DATA#26,#D0,#F7,#38,#A5,#00,#85,#02
5 DATA#E9,#28,#85,#00,#A5,#01,#85,#03
6 DATA#E9,#00,#95,#01,#CA,#D0,#E1,#A0
7 DATA#00,#A9,#20,#99,#D2,#BB,#C9,#C0
8 DATA#26,#D0,#F8,#60
9 ' CHARGE LANGAGE MACHINE
10 FORR=0TO#3B:READA:POKE#400+R,A:NEXT
11 DATA1,1,7,31,57,1,3,7
12 DATA32,32,56,62,39,32,48,56
13 DATA6,7,31,15,31,31,13,0
14 DATA0,56,62,60,60,60,48,32
17 '
18 'DEFINITION DES CARACTERES
19 '
20 FORR=0TO31:READA:POKE#2508+R,A:NEXT
67 '
```

```
68 ' INITIALISATION
69 '
70 CLS:POKE#26A,10:TEXT:PAPER0:INK6
80 FORR=0TO3:POKE#BBA7-R,32:NEXT
90 N=1:PLAY0,1,0,0
100 KM=0
110 DOKE#BE10,#2221:V=#BE10
120 FORR=31TO10STEP-1:SOUND4,R,15:WAIT4:NEXT
127 '
128 ' BOUCLE PRINCIPALE
129 '
130 AZ=INT(RND(1)*N)
140 FORR=0TOAZ:DOKE#BBFA
+INT(RND(1)*37),#2423:NEXT
200 DOKEY,#2020
210 CALL#400
220 IFPEEK(#208)=#94THENV=V+1
230 IFPEEK(#208)=#9CTHENV=V-1
240 IF DEEK(V)<>#2020THEN300
250 DOKEY,#2221
290 KM=KM+1:P=P+1:IFP=100THEN400
290 GOTO130
297 '
298 ' PERDU
299 '
300 EXPLODE
310 FORR=0TO10:PAPER7:PAPER0:NEXT:INK3
315 WAIT100
320 CLS:PLOT16,15,"PERDU"
330 PRINT"VOUS AVEZ PARCOURU ";
KM/100;"KILOMETRE."
340 END
400 P=0:IFN=2THEN430
410 N=N+1
420 GOTO200
430 DOKEY,#2020:V=V-40
440 GOTO130
```

Commentaires

- 10 Met les codes machine en mémoire.
- 20 Définition des graphismes.

- 7) Le POKE #26A, 1) élimine le curseur et le bip clavier.
- 8) Elimination du CAPS en haut et à droite.
- 11) Positionnement du vaisseau.
- 12) Post-combustion des réacteurs.
- 13) Tirage d'une valeur aléatoire pour placer les météorites.
- 20) Effacement du vaisseau.
- 21) Défilement de l'espace.
- 22-25) Réaffichage du vaisseau dans sa nouvelle position.
- 28) Décompte des kilomètres parcourus et évolution des difficultés.
- 30-34) Perdu et bilan.
- 40) Nouvelles difficultés : prolifération des météorites et montée du vaisseau vers le haut de l'écran.

Bloc-notes

Ce programme est un exemple de réutilisation de la routine en langage machine du programme 24 Heures du Mans.

SERPENT



La jungle grouille de boas et autres anacondas lovés autour du bois d'ébène, telles d'inoffensives lianes. Quand ils déroulent leurs anneaux, il est déjà trop tard : vous êtes prisonnier, au bord de l'asphyxie. Il ne vous reste plus qu'à prier... Tarzan.

Mais non ! Tout cela n'était qu'un mauvais rêve. Vous êtes bien assis, devant votre Oric-Atmos, en train de taper une simulation de serpent vorace. Un jeu drôle et plein de circonvolutions.

Déroulement

Le titre du jeu s'affiche au centre de l'écran. Si vous appuyez sur une touche, le corps de votre bestiole apparaît et simultanément un chiffre, à une position quelconque de l'écran. L'animal affamé n'a de cesse que d'aller manger le chiffre. Dès que vous sollicitez une flèche de direction, la tête du serpent apparaît et vous pouvez le guider vers sa « nourriture ». Parallèlement, en haut et à gauche, apparaît le nombre des chiffres dévorés; en bas et au centre, le nombre de déplacements d'un caractère que vous faites pour chaque « victime ». Ce nombre est réinitialisé à 79 à chaque festin de votre horrible reptile, mais vous ne pouvez dépasser 80 pour chaque déplacement.

Le serpent s'allonge d'autant d'anneaux que le montant du chiffre qu'il engloutit. Si, dans sa goinfrie, votre serpent heurte les limites de son territoire, il perd ses anneaux, un bruit de casserole le poursuit dans ses déplacements. S'il en vient à se marcher sur la queue, il explose et le jeu s'interrompt immédiatement. Qui pilotera, jusqu'à saturation de l'écran, le serpent le plus long dans les méandres de ses propres anneaux aura gagné. Merci à O. Rozenkranc, auteur d'une version TO7 de cette aventure.

Structure

10-90	Initialisations
100-175	Déplacements
175-200	Affichages
200-250	Tests
300-330	Tirages aléatoires
400-450	Affichage de la tête

```
1 REM
2 REM
3 REM SERPENT
4 REM
5 REM Initialisations
6 FORR=0TO7:READA:POKE#B558+R,A:NEXT
7 FORR=0TO7:READA:POKE#B778+R,A:NEXT
10 TEXT:POKE#26A,10:CLS:PAPER0:INK3
20 CLEAR
30 DIM T(40,24)
```

```
40 X0=3:Y0=11:X=8:Y=11:TE$=""
50 SC=5:F=0:SC$=STR$(SC)
65 PLOT12,12,"SERPENT"
70 S=RND(1):IF KEY$="" THEN 70
75 CLS:PLOT34,0,SC$
80 INK3:FOR X=3 TO 7:T(X,11)=2
85 PLOTX,11,"+":NEXT X
90 GOSUB300
97 REM
98 REMDéplacement corps
99 REM
100 IF KEY$="" THEN 100
110 A=PEEK(#208)
125 IF F=1THEN SHOOT
130 IFA=#SC THEN XX=-1:YY=0:C=1
135 IF A=#94 THEN XX=1:YY=0:C=2
140 IF A=#AE THEN XX=0:YY=-1:C=3
145 IFA=#AA THEN XX=0:YY=1:C=4
147 CH$=""
150 IF X+XX<2 OR XX+X>38 THEN 155 ELSE 160
155 F=1:GOTO 210
160 IF Y+YY<1 OR YY+Y>23 THEN 165 ELSE 170
165 F=1:GOTO 210
170 F=0:IF T(X+XX,Y+YY)>0 THEN 400
175 T(X,Y)=C:PLOTX,Y,CH$
180 X=XX+X:Y=Y+YY:DEP=DEP+1
190 PLOT X,Y,TE$
192 PAS=PAS+1:PLOT19,24,STR$(80-PAS)
194 IF PAS=80 THEN 195 ELSE 198
195 GOTO 400
198 IF T(X,Y)<0 THEN 199 ELSE 205
199 U=0:UM=Y:SC=SC+Y:PLOT34,0,STR$(SC):PAS=0
200 GOSUB 300
205 IF U<UM THEN U=U+1:GOTO110
210 PLOTX0,Y0," "
215 C0=T(X0,Y0):T(X0,Y0)=0
220 IF C0=0 THEN 400
230 IF C0=1 THEN X0=X0-1:Y0=Y0
235 IF C0=2 THEN X0=X0+1:Y0=Y0
240 IF C0=3 THEN X0=X0:Y0=Y0-1
245 IF C0=4 THEN X0=X0:Y0=Y0+1
250 GOTO110
299 REM
300 REM Alea
```

```

301 REM
310 XC=INT(RND(1)*36)+2
320 YC=INT(RND(1)*20)+2
325 IF T(XC,YC)=0 THEN 330 ELSE 310
330 V=INT(RND(1)*8)+1
340 N$=MID$(STR$(V),2)
350 PLOT XC,YC,N$
360 T(XC,YC)=-1
370 RETURN
399 REM
400 REM
410 FOR I=0 TO 10
420 FOR J=1 TO 7
430 PLOTX,Y,TE$
440 NEXT NEXT
445 EXPLODE
450 PLOT X,Y,TE$
460 END
497 REM
498 REM CARACTERE
499 REM
500 DATA0,0,12,30,30,12,0,0
510 DATA0,0,12,18,18,12,0,0

```

Commentaires

6-7 Chargement des caractères en DATA.

10 Suppression du curseur et du bip, couleurs de fond et d'encre.

30 Tableau des positions du serpent.

40 Initialisations des variables « corps » et « tête » du serpent et position de début de jeu.

50 Initialisation du score.

65 Titre.

70 Relance du tirage aléatoire.

80-85 Affichage de la première position du serpent.

100-145 Déplacement à l'aide des touches de direction.

125 F est le drapeau de sortie d'écran.

150-165 Test de sortie d'écran.

170 Test du serpent se marchant sur la queue.

180 DEP est le nombre de déplacements.

190 Affichage de la tête.

192 Affichage du nombre de déplacements.

194 Test d'excès du nombre de déplacements. Vous pouvez renforcer la difficulté du jeu en diminuant ce paramètre.

210 Effacement d'un anneau de queue.

215 Tableau de position de la queue.

220 Test de sortie en cas de serpent se marchant sur la queue.

230-245 Déplacement du dernier anneau.

310-320 Tirage aléatoire de la position du nombre V.

330 Tirage de V, le nombre à « manger ».

340-350 Affichage de V. STR\$ créant un blanc devant l'équivalent chaîne d'une valeur numérique, il faut extraire ce blanc par MID\$ avant l'affichage avec PLOT.

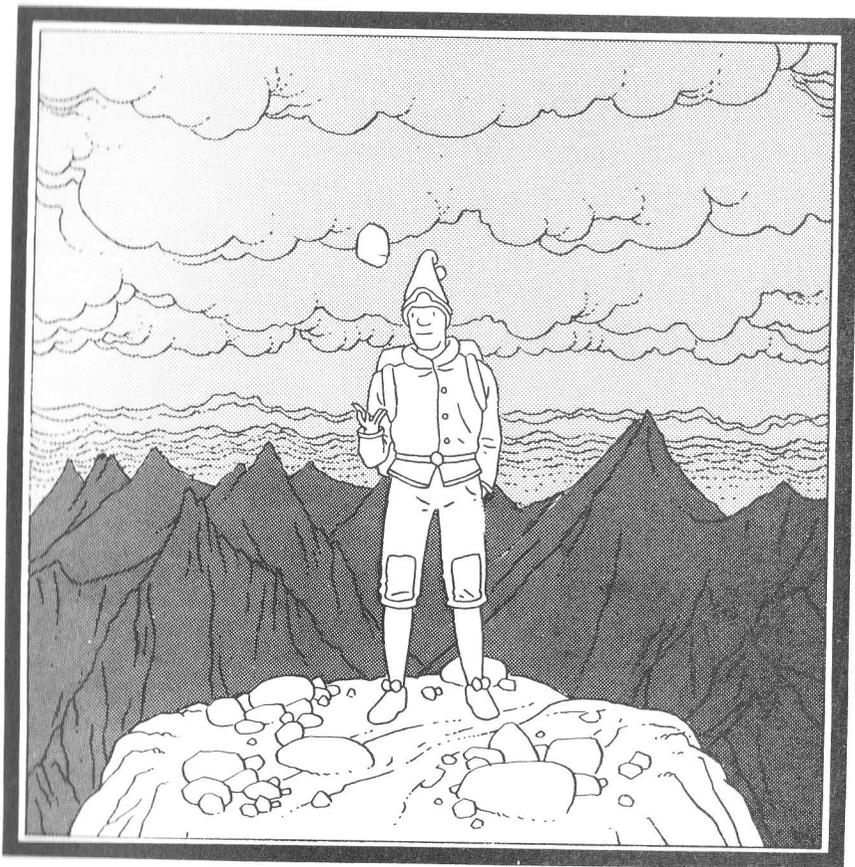
400-460 Sortie du jeu en cas de mauvaise manœuvre.

Bloc-notes

La gestion du déplacement du corps du serpent 100-250 peut être adaptée à d'autres jeux.

Ce jeu met en évidence l'utilisation de PLOT qui n'affiche que des chaînes de caractères, d'où la nécessité de transformer les valeurs numériques en chaînes par STR\$ (Ex : 192).

En ligne 70, un truc pour le renouvellement de l'aléatoire. Au lieu de demander un nombre et de relancer RND dans une boucle de la longueur de ce nombre, on peut utiliser KEY\$ lors d'un test clavier obligatoire. Tant qu'on n'appuie pas sur une touche, la randomisation a lieu.



Vous connaissez l'histoire de ce personnage condamné par les dieux ? Sa peine consistait à rouler une pierre au sommet de la montagne puis, arrivé au but, à la voir dégringoler pour recommencer. Ici, les pierres sont des super-balles en caoutchouc. Notre Sisyphus (vous) a tellement hâte de jouer avec qu'à chaque fois il trébuche et les fait tomber. L'ennui est qu'elles rebondissent indéfiniment et qu'il est condamné à aller les chercher sous cette pluie meurtrière. Quel rapport, direz-vous, avec le personnage antique ? Aucun apparemment, si ce n'est l'absurdité qui les accable tous deux. Combien de balles pourrez-vous faire rebondir sans en prendre une sur la tête ?

Déroulement

Sur l'écran bleu de l'éternité un petit bonhomme s'agite curieusement. Son manège consiste à transporter inlassablement des balles d'un bout à l'autre de son domaine. Le maladroit perd régulièrement son fardeau. Et voilà que les damnées balles rebondissent à qui mieux mieux. L'individu n'en persiste pas moins à se déplacer sous votre contrôle : vers la droite avec la touche >, vers la gauche avec la touche <.

Il prend la balle à droite et va la porter à gauche en ne cessant de maugréer dès qu'il la perd. A quoi sert d'insulter les dieux si tel est votre destin...

Comment connaître votre score ? En prenant une balle sur la tête, bien sûr.

Structure

10-38	Code machine
40-43	Caractères
49-196	Initialisations
200-240	Boucle centrale
300-390	Sous-programme de déplacement du personnage
400-460	Sous-programme de balle perdue
600-690	Perdu
700-740	Une balle de plus

```

1 REM
3 REM
7 REM
8 REM   SISYPHE
9 REM
10 DATA#AE, #00, #60, #BD
11 DATA#00, #65, #85, #00
12 DATA#BD, #00, #66, #85
13 DATA#01, #BD, #00, #67
14 DATA#A8, #A9, #20, #91
15 DATA#00, #BD, #00, #68
16 DATA#10, #32, #29, #7F
17 DATA#F0, #26, #BD, #00
18 DATA#65, #18, #69, #28

```

```

19 DATA#9D,#00,#65,#85
20 DATA#00,#BD,#00,#66
21 DATA#69,#00,#9D,#00
22 DATA#66,#85,#01,#DE
23 DATA#00,#68,#4C,#6B
24 DATA#40,#BD,#00,#68
25 DATA#09,#80,#9D,#00
26 DATA#68,#4C,#1E,#40
27 DATA#BD,#00,#68,#29
28 DATA#7F,#9D,#00,#68
29 DATA#29,#7F,#DD,#00
30 DATA#69,#F0,#E6,#BD
31 DATA#00,#65,#38,#E9
32 DATA#28,#9D,#00,#65
33 DATA#85,#00,#BD,#00
34 DATA#66,#E9,#00,#9D
35 DATA#00,#66,#85,#01
36 DATA#FE,#00,#68,#A9
37 DATA#21,#91,#00,#CA
38 DATA#10,#91,#60
39 REM CARACTERE
40 DATA0,0,12,30,30,12,0,0
41 DATA12,30,12,63,12,30,33,33
42 DATA0,0,12,30,30,12,0,0
43 DATA63,0,0,0,0,0,0,0
46 REM
47 REM INITIALISATION
48 REM
49 IFPEEK(#B508)=0THEN170
50 FORR=0TO#72:READA:POKE#4000+R,A:NEXT
60 FORR=0TO31:READA:POKE#B508+R,A:NEXT
170 DF=15:NB=0:POKE#26A,10
175 CLS:POKE#BFB5,33:PAPER6:INK0
180 GOSUB700
190 HB=#2220:AB=#BF92:DOKEAB,HB
195 FORR=0TO36:POKE#BFBA+R,36:NEXT
196 POKE#BFB9,4
197 REM
198 REM BOUCLE CENTRALE
199 REM
200 IFPEEK(AB+1)<>34THEN600
210 IFPEEK(#208)=#94THEN300
220 IFPEEK(#208)=#8CTHEN350
225 WAIT4

```

```

230 CALL#4000
240 GOTO200
297 REM
298 REM SOUS PROGRAMME
299 REM
300 IFXB=35THEN330
305 IFDEEK(AB)=#2220THENHB=#2220
310 DOKEAB,#2020
315 XB=XB+1:AB=AB+1
320 DOKEAB,HB:GOTO230
330 IFHB=#2223THEN230
340 HB=#2223:SC=SC+10:GOTO320
350 IFXB=0THEN380
355 IFDEEK(AB)=#2220THENHB=#2220
360 DOKEAB,#2020
365 XB=XB-1:AB=AB-1
370 DOKEAB,HB:GOTO230
380 IFHB=#2220THEN230
390 HB=#2220:SC=SC+20:PLOT4,22,"DAMNED"
400 DF=DF-1:NB=NB+1:POKE#BFB5,33
405 DF=ABS(DF)
410 GOSUB700
420 DOKEAB,HB
430 FORR=10TO5STEP-9:SOUND1,R,15:NEXT
440 FORR=5TO200STEP20:SOUND1,R,15:NEXT
450 SOUND1,0,0
460 PLOTXB,22,"":GOTO230
600 PLOTXB,22,"DIANTRE"
610 FORR=500TO2000STEP40:SOUND1,R,15:NEXT:
SOUND1,0,0
630 PLOTXB,22,""
640 CLS
660 PRINT" VOTRE SCORE EST:";SC
670 PRINT:PRINT:PRINT:PRINT
675 PRINT"VOUS AVEZ PORTE ";NB
676 PRINT:PRINT" SUPER BABALLE(S)"
680 IFPEEK(#208)=#38THEN680
690 RUN
700 POKE#6900+NB,DF+INT(RND(1)*10)
710 POKE#6800+NB,0
720 POKE#6000,NB:POKE#6500+NB,148
725 POKE#6600+NB,191
730 POKE#6700+NB,INT(RND(1)*31)
740 RETURN

```

Code machine

```

14000-4072
4000:  AE  00  60  LDX $6000
4003:  BD  00  65  LDA $6500,X
4006:  85  00          STA $00
4008:  BD  00  66  LDA $6600,X
400B:  85  01          STA $01
400D:  BD  00  67  LDA $6700,X
4010:  A8          TAY
4011:  A9  20          LDA #$20
4013:  91  00          STA ($00),Y
4015:  BD  00  68  LDA $6800,X
4018:  10  32          BPL $404C
401A:  29  7F          AND #$7F
401C:  F0  26          BEQ $4044
401E:  BD  00  65  LDA $6500,X
4021:  18          CLC
4022:  69  28          ADC #$28
4024:  9D  00  65  STA $6500,X
4027:  85  00          STA $00
4029:  BD  00  66  LDA $6600,X
402C:  69  00          ADC #$00
402E:  9D  00  66  STA $6600,X
4031:  85  01          STA $01
4033:  DE  00  68  DEC $6800,X
4036:  4C  6B  40  JMP $406B
4039:  BD  00  68  LDA $6800,X
403C:  09  80          ORA #$80
403E:  9D  00  68  STA $6800,X
4041:  4C  1E  40  JMP $401E
4044:  BD  00  68  LDA $6800,X
4047:  29  7F          AND #$7F
4049:  9D  00  68  STA $6800,X
404C:  29  7F          AND #$7F
404E:  DD  00  69  CMP $6900,X
4051:  F0  E6          BEQ $4039
4053:  BD  00  65  LDA $6500,X
4056:  38          SEC
4057:  E9  28          SBC #$28
4059:  9D  00  65  STA $6500,X
405C:  85  00          STA $00
405E:  BD  00  66  LDA $6600,X
4061:  E9  00          SBC #$00

```

```

4063:  9D  00  66  STA $6600,X
4066:  85  01          STA $01
4068:  FE  00  68  INC $6800,X
406B:  A9  21          LDA #$21
406D:  91  00          STA ($00),Y
406F:  CA          DEX
4070:  10  91          BPL $4003
4072:  60          RTS

```

Commentaires

49 On teste si le langage machine est déjà en mémoire. Si oui, on saute directement à l'initialisation. Si non, on charge les codes machine et les caractères.

50-60 Chargement des codes machine et des caractères.

180 On saute à la routine qui rajoute une balle sur l'écran.

190-196 On affiche le sol, le bonhomme et on réinitialise des variables.

200 On teste si le bonhomme a pris une balle sur la tête.

210-220 Test clavier.

230 On fait bouger toutes les balles d'une case.

240 Retour de boucle.

300 On teste si le bonhomme est arrivé sur le bord droit de l'écran.

305 On teste si une balle est tombée sur une balle du bonhomme et dans ce cas, il la perd.

310 On efface le bonhomme.

315 On le fait bouger d'une case à droite.

320 On le réaffiche et on retourne dans la boucle centrale.

330-340 On teste si le bonhomme a pris la balle. Si oui, on rajoute dix points au score.

350 On teste si le bonhomme a touché le bord gauche de l'écran. S'il a une balle à la main, il la perd. Le score augmente de 20. La balle perdue va rebondir sur l'écran.

360-370 On fait bouger le bonhomme vers la gauche.

400 La nouvelle balle rebondira moins haut que les précédentes (DF=DF-1).

600-690 Page perdue et retour à une nouvelle partie.

700-740 Nouvelle balle :

700 De #6900 jusqu'à #69FF on stockera pour chaque balle à quelle hauteur elle va rebondir.

710 De #6800 jusqu'à #68FF on stockera pour chaque balle, à quelle hauteur elle se trouve et dans quel sens elle bouge(montée ou descente). Chaque nouvelle balle sera initialisée à zéro.

720 On stocke en #6000 le nombre de balles qu'il y a sur l'écran. De #6500 jusqu'à #65FF, on stockera pour chaque balle son adresse basse sur l'écran. De #6600 jusqu'à #66FF, on stockera pour chaque balle son adresse haute sur l'écran. Chaque nouvelle balle est initialisée à #BF92 (#92=148 ; #BF=191).

730 De #6700 jusqu'à #67FF, on stockera pour chaque balle ses coordonnées en Y (numéro de la colonne où elle se trouve).

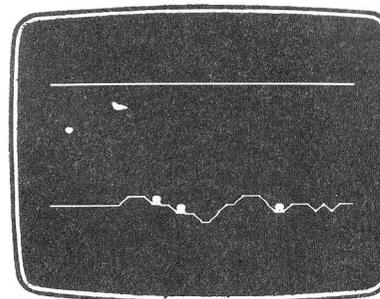
MISSION BOMBARDEMENT

La guerre fait rage quelque part sur une planète inconnue. Engoncé dans le cockpit de votre chasseur-bombardier, vous voyez le relief défilé vertigineusement sous vos ailes. Votre mission : voler en rase-mottes et détruire les stations radar ennemies qui prolifèrent dans cette zone montagneuse. Mission aisée si l'ennemi n'avait pas semé quelques mines volantes qui, au moindre choc, vous transforment en torche vivante. Si vous n'avez pas la vocation de kamikaze, inutile de vous porter volontaire pour une mission que seul un individu aux nerfs d'acier pourra mener à bien. Contentez-vous de taper le programme et laissez jouer les professionnels !

Déroulement

Dès le décollage, les puissantes tuyères de votre chasseur-bombardier hurlent à vos oreilles. Tenez le manche d'une main ferme car les premiers dangers apparaissent déjà. Les montagnes montent, par endroits, si haut qu'il est difficile — voire impossible — de se glisser entre elles et votre plafond de vol. De plus, les mines volantes surgissent curieusement où on ne les attend pas. En considérant que les radars sont toujours derrière le flanc des montagnes, cette mission, si facile d'apparence, s'avère un défi à vos réflexes.

Pour orienter votre assiette de vol, vous disposez de quatre commandes : < pour ralentir, > pour accélérer, A pour prendre de l'altitude et Z pour descendre. La barre d'espace vous permet de larguer vos bombes. Combien d'objectifs arriverez-vous à détruire avant de mourir glorieusement au combat ?



```

1 REM
2 REM
3 REM MISSION BOMBARDEMENT
4 REM
5 REM
7 REM
8 REM Lan9a9e machine
9 REM
10 DATA#A2,#00,#A0,#00
11 DATA#B1,#00,#91,#02
12 DATA#C8,#C0,#26,#D0
13 DATA#F7,#A9,#20,#91
14 DATA#02,#18,#A5,#02
15 DATA#69,#28,#85,#02
16 DATA#48,#A5,#03,#69
17 DATA#00,#85,#03,#68
18 DATA#18,#69,#01,#85
19 DATA#00,#A5,#03,#69
20 DATA#00,#85,#01,#E8
21 DATA#E0,#1C,#D0,#D2
22 DATA#60,#A0,#01,#A9
23 DATA#20,#91,#04,#88
24 DATA#10,#FB,#20,#00
25 DATA#04,#AD,#08,#02
26 DATA#A2,#01,#C9,#94
27 DATA#F0,#0E,#C9,#8C
28 DATA#F0,#19,#A2,#28
29 DATA#C9,#AE,#F0,#13
30 DATA#C9,#AA,#D0,#1E
31 DATA#8A,#18,#65,#04
32 DATA#85,#04,#A5,#05
33 DATA#69,#00,#85,#05
34 DATA#4C,#72,#04,#86
35 DATA#06,#A5,#04,#38
36 DATA#E5,#06,#85,#04
37 DATA#A5,#05,#E9,#00
38 DATA#85,#05,#A0,#01
39 DATA#B1,#04,#C9,#20
40 DATA#D0,#0E,#88,#10
41 DATA#F7,#A0,#01,#B9
42 DATA#FD,#04,#91,#04
43 DATA#88,#10,#F8,#60
44 DATA#A9,#01,#8D,#FC
45 DATA#04,#60

```

```

46 REM
47 REM *****
48 REM * CARACTERE *
49 REM *****
50 DATA0,48,56,60,31,31,31,7
51 DATA0,0,0,0,48,60,63,32
52 DATA30,45,45,45,30,18,33,63
53 DATA1,1,2,4,8,16,32,32
54 DATA63,0,0,0,0,0,0,0
55 DATA32,32,16,8,4,2,1,1
56 DATA0,0,0,0,0,0,0,63
57 DATA0,30,33,63,18,12,0,0
58 DATA45,24,61,35,51,62,37,35
59 DATA0,0,16,56,24,4,2,1
60 CLS:POKE#26A,10
65 PLAY0,7,0,0
97 REM *****
98 REM * CHARGE CODES MACHINE *
99 REM *****
100 FORR=0TO#8D
110 READA
120 POKE#400+R,A
130 NEXT
137 REM *****
138 REM * CHARGE CARACTERES *
139 REM *****
140 FORR=0TO79
150 READA
160 POKE#8508+R,A
170 NEXT
175 PAPER0:DOKE#4FD,#2221:AC=#25:SC=0
190 FORR=0TO3:POKE#BBA4+R,32:NEXT
195 FORR=0TO38:POKE#BB00+R,39:
    POKE#BE28+R,37:SOUND4,R,5:NEXT
200 DOKE4,#BD18:M=#BE4E:HM=15:POKE#4FC,0
210 DOKE0,#BBA9:DOKE2,#BBA8:POKE#BBF6,39
212 IF T=1THENPOKE#T,32:GOTO220
215 IFPEEK(#208)=#84THEN700
220 CALL#431
225 IFPEEK(#4FC)=1THEN800
227 IFT=1THEN710
230 AZ=INT(RND(1)*5)
235 IFAZ=4THEN600
240 IFAZ=1THEN300

```

```

245 IFAZ=2THEN400
250 IFAZ=3THEN500
255 IFAC=#26ORAC=#23THENM=M+40:HM=HM+1
260 POKEM,37:AC=#25
270 GOTO210
300 IFHM-1<4THEN255
320 IFAC=#24 OR AC=#25 THEN M=M-40:HM=HM-1
330 POKEM,36:AC=#24
340 GOTO210
400 IFHM+1>20THEN255
410 IFAC=#26ORAC=#23THEN M=M+40:HM=HM+1
420 POKEM,38:AC=#26
430 GOTO210
500 IFAC=#26THEN520
510 GOTO255
520 POKEM,35:AC=#23
530 GOTO210
600 IF INT(RND(1)*2)=0THEN630
610 HB=INT(RND(1)*HM)
620 POKE#BC1E+40*HB,40
630 GOTO255
700 T=1:MT=DEEK(4):GOTO210
710 MT=MT+40
720 IFDEEK(MT-1)<>#2020THEN740
730 POKEMT,42:GOTO230
740 IFPEEK(MT)=35THENSC=SC+1
750 T=0:POKEMT,41:SHOOT
760 PAPER7:PAPER0
770 PLAY0,7,0,0:SOUND4,6,5
780 GOTO230
800 FORR=0TO10:PAPER7:PAPER0:SOUND4,3*R,15:
NEXT
810 EXPLODE
820 WAIT50
830 CLS:PRINT"VOUS AVEZ DETRUIT ";SC;"CIBLE."
840 PRINT:PRINT:PRINT:PRINT
850 PRINT:PRINTSPC(10)"PERDU"

```

Code machine

1400-480

```

0400: A2 00 LDX #000
0402: A0 00 LDY #000
0404: B1 00 LDA (#00),Y

```

```

0406: 91 02 STA (#02),Y
0408: C8 INY
0409: C0 26 CPY #26
040B: D0 F7 BNE #0404
040D: A9 20 LDA #20
040F: 91 02 STA (#02),Y
0411: 18 CLC
0412: A5 02 LDA #02
0414: 69 28 ADC #28
0416: 85 02 STA #02
0418: 48 PHA
0419: A5 03 LDA #03
041B: 69 00 ADC #00
041D: 85 03 STA #03
041F: 68 PLA
0420: 18 CLC
0421: 69 01 ADC #01
0423: 85 00 STA #00
0425: A5 03 LDA #03
0427: 69 00 ADC #00
0429: 85 01 STA #01
042B: E8 INX
042C: E0 1C CPX #1C
042E: D0 D2 BNE #0402
0430: 60 RTS
0431: A0 01 LDY #01
0433: A9 20 LDA #20
0435: 91 04 STA (#04),Y
0437: 88 DEY
0438: 10 FB BPL #0435
043A: 20 00 04 JSR #0400
043D: A0 08 02 LDA #0208
0440: A2 01 LDX #01
0442: C9 94 CMP #94
0444: F0 0E BEQ #0454
0446: C9 8C CMP #8C
0448: F0 19 BEQ #0463

044A: A2 28 LDX #28
044C: C9 AE CMP #AE
044E: F0 13 BEQ #0463
0450: C9 AA CMP #AA
0452: D0 1E BNE #0472
0454: 8A TXA

```

```

0455: 18          CLC
0456: 65 04       ADC #04
0458: 85 04       STA #04
045A: A5 05       LDA #05
045C: 69 00       ADC #00
045E: 85 05       STA #05
0460: 4C 72 04    JMP #0472
0463: 86 06       STX #06
0465: A5 04       LDA #04
0467: 38          SEC
0468: E5 06       SBC #06
046A: 85 04       STA #04
046C: A5 05       LDA #05
046E: E9 00       SBC #00
0470: 85 05       STA #05

0472: A0 01       LDY #01
0474: B1 04       LDA (#04),Y
0476: C9 20       CMP #20
0478: D0 0E       BNE #0488
047A: 88          DEY
047B: 10 F7       BPL #0474
047D: A0 01       LDY #01
047F: B9 FD 04    LDA #04FD,Y
0482: 91 04       STA (#04),Y
0484: 88          DEY
0485: 10 F8       BPL #047F
0487: 60          RTS
0488: A9 01       LDA #01
048A: 8D FC 04    STA #04FC
048D: 60          RTS

```

Commentaires

190 Effacement du CAPS.

195 Affichage du couloir de départ et bruit des moteurs.

200 Mise en place de tables à l'usage de la routine en langage machine.

210 Début de la boucle centrale : initialisation de vecteurs pour le déplacement horizontal de l'écran.

212 Test de tirs.

215 Test clavier de tirs.

220 Appel de la sous-routine de défilement d'écran et de déplacement de l'avion.

225 Test de collision.

227 Second test de tir.

230-250 Fonction aléatoire déterminant l'évolution du jeu.

255-260 Montagne plate.

300-340 Flanc de montagne montant.

400-430 Flanc descendant.

500-530 Apparition d'un radar derrière un flanc descendant.

600-630 Apparition d'une mine volante.

700-730 Gestion du tir.

Bloc-notes

La routine de langage machine peut être insérée dans un jeu. Pour cela, tapez la première série de DATA donnée en début de programme plus le bout de routine allant de 100 à 130 inclus, chargé de placer la routine en mémoire.

Cette routine peut s'utiliser de deux façons :

— en faisant DOKE 0,#BBA9 : DOKE2,#BBA8 : CALL #400, on obtient un déplacement de l'écran de droite à gauche.

— en faisant les deux mêmes DOKE mais en appelant la routine en #431, on obtient un déplacement de l'écran de la droite vers la gauche plus l'affichage de deux caractères dont la valeur doit être fixée au début du jeu en #4FD et #4FE. De plus, utilisée de cette manière, cette routine fait la gestion du clavier et déplace les caractères en fonction des touches enfoncées.

GÉNÉRATEUR DE CARACTÈRES



Vous désirez faire un alphabet arabe, grec, chinois ou simplement écrire en anglaises ou en italiques ou encore faire du graphisme en mode TEXT : voici l'utilitaire que vous attendiez. Il répondra à toutes vos exigences esthétiques.

Déroulement

Un cadre se dessine définissant le contour d'un caractère de six sur huit. En haut à gauche se place le curseur que l'on peut déplacer avec :

- A pour monter,
- Z pour descendre,
- < pour aller à gauche,
- > pour aller à droite,

Lorsqu'on déplace le curseur, il ne laisse pas de trace. Si l'on veut « baisser le crayon » il faut appuyer sur le SHIFT de droite en même temps qu'on déplace le curseur.

Une fois le caractère défini, il faut appuyer sur SHIFT : le message « Désirez-vous mémoriser ce caractère (O/N) » apparaît. Si OUI, il demande le caractère à programmer et dans quel jeu de caractères (standard ou using). Après réponse, il y a mémorisation du caractère. L'ordinateur attend qu'on appuie sur une touche. Après quoi, il demande si on veut définir un autre caractère.

Si on a répondu NON, l'ordinateur vous offre la possibilité de remettre en mémoire les caractères standards.

Remarque

Ce programme débute en 400 de manière à pouvoir être couplé au programme de dessin en HIRÉS. Si vous êtes l'heureux possesseur d'Atmos, l'instruction CLOAD, J vous permettra de le faire même si les deux programmes ont été tapés séparément.

```
387 REM
388 REM      GENERATEUR DE CARACTERES
389 REM
397 REM
398 REM      DESSIN DU CADRE
399 REM
400 CLS
410 FORR=0T07
420 POKE#BC7F+R,127
430 POKE#BDE7+R,127
440 NEXTR
```

```
450 FORR=0T09
460 POKE#BC7F+R#40,127
470 POKE#BC86+R#40,127
480 NEXT
490 POKE#25A,10
497 REM
498 REM      INITIALISATION
499 REM
500 INK1:PAPER0
510 X=0:Y=0:AD=#BCA8
515 POKEAD,160
519 REM
520 REM      BOUCLE CENTRALE
521 REM
522 GETA$:IFA$="!"THEN800
525 IFPEEK(#209)=167THEN530
528 POKEAD,32
530 IFPEEK(#208)=#AETHEN600
540 IFPEEK(#208)=#AETHEN650
550 IFPEEK(#208)=#CTHEN700
560 IFPEEK(#208)=#94THEN750
580 IFA$="+ "THEN905
590 POKEAD,160
595 GOTO522
597 REM
598 REM      DEPLACEMENT CURSEUR
599 REM
600 IFY=0THEN522
610 Y=Y-1:AD=AD-40:GOTO590
650 IFY=7THEN522
660 Y=Y+1:AD=AD+40:GOTO590
700 IFX=0THEN522
710 X=X-1:AD=AD-1:GOTO590
750 IFX=5THEN522
760 X=X+1:AD=AD+1:GOTO590
797 REM
798 REM      CALCUL OCTETS CORRESPONDANT
799 REM
800 INK2:PRINT"DESIREZ VOUS MEMORISER
      CE CARACTERE ? ... (O/N)":GETA$
802 IFA$="N"THEN 400
805 PRINT"CARACTERE A PROGRAMER?":GETA$
810 PRINTA$
815 PRINT"STANDARD OU USER ?...(S/U)"
```

```

820 GETB$
830 IFB$="S" THEN B=#B400:GOTO850
840 IFB$="U" THEN B=#B800:GOTO850
845 GOTO820
850 A=B+(ASC(A$)*8)
855 X=#BCAS
860 FORT=0T07
870 FORR=0T05:A(R)=(PEEK(X+R)-32)/128:NEXTR
875 X=X+40
880 V=A(0)*32+A(1)*16+A(2)*8+A(3)
      *4+A(4)*2+A(5)
885 POKEA+T,V:NEXTT
890 GETA$:CLS:PRINT
      "AUTRE CARACTERE ?...(O/N)":GETA$
900 IFA$="O" THEN 400
905 IFPEEK(#208)<>#38 THEN 905
906 WAIT100
910 PRINT"DESIREZ VOUS REPENDRE
      LES CARACTERES STANDARDS ?...(O/N)"
920 GETA$:IFA$<>"O" THEN END
925 WAIT100
930 PRINT"CONFIRMATION...(O/N)":GETA$
940 IFA$="O" THEN CALL#F800

```

Commentaires

41Ø-44Ø Affichage des deux bords horizontaux du cadre.

45Ø-48Ø Affichage des deux bords verticaux du cadre.

51Ø-515 Affichage du curseur et définition de ses coordonnées.

522 Début de boucle : arrêt du clavier. Tant qu'une touche n'a pas été enfoncée, la boucle est bloquée. Si on appuie sur SHIFT, il y a mise en mémoire du caractère.

525 Si le SHIFT droit est enfoncé, le CURSET laisse une trace en se déplaçant.

528 Effacement du curseur dans sa dernière position.

53Ø-58Ø Test clavier.

59Ø Affichage du curseur.

595 Retour de boucle.

6ØØ-76Ø Calcul de la nouvelle position du curseur et test de bord de cadre.

8ØØ-845 Procédure.

85Ø Calcul de l'adresse en mémoire du caractère programmé.

855 X est l'adresse du bord intérieur en haut et à gauche du cadre.

86Ø-88Ø — 87Ø Transformation ligne par ligne du contenu de la case écran à l'intérieur du cadre en un tableau pour lequel :

— une case blanche = 1

— une case noire = Ø

— 88Ø Somme de toutes les valeurs du tableau en tenant compte de leur poids (coefficient) :

$$V=A(\emptyset)2^0+A(1)^1+A(2)^2+\dots+A(5)^5$$

Transforme un motif de six points en un octet en Ø et 63 (Ø et #3F), $\Lambda(X)$ étant égal à Ø ou à 1.

89Ø-94Ø Autre caractère ? Sinon caractère standard et fin.

Bloc-notes

Pour sauvegarder des caractères définis comme ci-dessus, il faut taper :

CSAVE « nom des caractères », A#B4ØØ,E#B7FF (standard) ou
 CSAVE « nom des caractères », A#B8ØØ,E#BB7F (using).

Sur Oric 1, la ligne 94Ø s'écrit :... THEN CALL #F89B.

TÉLÉCRAN : ÉDITEUR DE DESSIN LORES

Voici un programme qui va solliciter votre créativité ou celle de vos jeunes enfants. Il fait appel aux possibilités graphiques d'Oric en mode LORES (basse résolution). Il vous permet de dessiner sur l'écran, en guidant votre curseur à la manière du jeu bien connu qu'on appelle télécran. Le tracé, de huit points de large, n'autorise que des motifs de type géométrique. En revanche, l'utilisation des six couleurs de votre machine vous permet de réaliser des motifs intéressants. Vous pourrez, bien entendu, enregistrer vos essais sur cassette. Insistons sur l'intérêt formateur pour les enfants de ce type de programme qui développe chez eux la coordination et la représentation spatiale.

Déroulement

Vous disposez de touches programmées pour manœuvrer le curseur : touches de 0 à 7 pour choisir vos couleurs sur la palette d'Oric, touches de déplacement dans les quatre directions pour tracer, DEL pour effacer, ESPACE pour se déplacer sans traçage, S pour enregistrer votre dessin sur cassette, L pour charger un dessin déjà enregistré.

```
1 '
2 '
5 ' TELECRAN
6 '
7 GOSUB 3000
10 CLS:LORES0
15 PRINTCHR$(17)
20 X=19:Y=13:P=7
30 GOSUB 2000
50 C=SCRN(X,Y)
60 PLOT X,Y,16:WAIT 10
70 PLOT X,Y,23:WAIT 10
90 PLOT X,Y,C
100 A#="KEY":IF A#="" THEN 50
110 A=ASC(A#)
```

```
130 IF A=8 THEN 800
140 IF A=9 THEN 900
150 IF A=10 THEN 1000
160 IF A=11 THEN 1100
170 IF A=32 THEN 1200
180 IF A=127 THEN 1300
190 IFA>47ANDAK<56THENP=A-48:B=0:GOSUB2000
200 IF A#="S" THEN 4000
210 IF A#="L" THEN 4100
300 GOTO 50
800 X=X-1
910 IF X<1 THEN X=1:GOTO 50
920 GOSUB 2000
930 GOTO 50
900 X=X+1
910 IF X>38 THEN X=38:GOTO 50
920 GOSUB 2000
930 GOTO 50
1000 Y=Y+1
1010 IF Y>26 THEN Y=26:GOTO 50
1020 GOSUB 2000
1030 GOTO 50
1100 Y=Y-1
1110 IF Y<0 THEN Y=0:GOTO 50
1120 GOSUB 2000
1130 GOTO 50
1200 B=1:GOTO 50
1300 B=0:P=0:GOSUB 2000:GOTO 50
2000 IF B=1 THEN RETURN
2010 PLOT X,Y,16+P
2020 RETURN
3000 'Presentation
3001 '
3002 CLS:PAPER 5:INK 2
3005 PLOT11,11,12:PLOT13,11,"TELECRAN"
3020 PRINT"Commandes"
3030 PRINT"Touches 0 a 7 : choix couleurs"
3040 PRINT"Touches <- -> deplacement"
3050 PRINT"DEL Pour effacer"
3060 PRINT"E Pour effacer tout l'ecran"
3070 PRINT"Barre ESPACE Pour se deplacer"
3075 PRINT"sans effacer le dessin"
3080 PRINT" S Pour sauvegarder le dessin "
3095 PRINT" sur cassette"
```

```

3090 PRINT" L Pour charger un dessin "
3095 PRINT" a partir d'une cassette"
3100 GET A#:CLS:RETURN
4000 '
4005 'operations sur fichier
4006 '
4010 INPUT"NOM DU DESSIN":N#
4020 CSAVE N#,A#BB80,E#BFDf,S
4030 RETURN
4100 INPUT"NOM DU DESSIN":N#
4110 CLOAD N#,A#BB80,E#BFDf,S
4120 RETURN

```

Commentaires

15 Effacement du curseur.

50 SCRN fournit le code ASCII du caractère situé à la position définie par les coordonnées x,y. Ce code ASCII permettra d'utiliser l'instruction PLOT pour dessiner les carrés successifs formant votre dessin (80).

80 PLOT permet d'afficher en un endroit donné un caractère. Si ce caractère est unique, il peut être défini par son code ASCII. Si ce code ASCII est inférieur à 32, il est considéré comme un ATTRIBUT. Ces attributs ou code escape sont, par exemple, les couleurs d'encre (de 1 à 7) ou bien les couleurs de fond (de 16 à 23) qui nous intéressent ici. En effet, en 190, A prend le numéro de la touche définissant les couleurs, puis P cette valeur dans l'intervalle 0-32 du code escape, enfin P est augmenté de 16 en 2010 pour obtenir le code des couleurs de fond de nos petits carrés.

60-70 Premier carré clignotant noir-blanc.

130-160 Déplacement du curseur : selon la touche sollicitée, il se déplace d'un caractère. Incrémentation de X ou Y dans les instructions 800,900,1000,1100.

4000 fin CSAVE pour sauver les blocs de mémoire écran constitués par votre dessin. Indiquez au programme les adresses de début et de fin de bloc.

Bloc-notes

La gestion du déplacement du curseur sur l'écran est adaptable à n'importe quel programme (100-160 et sous-programmes correspondants).

De même, le sous-programme de sauvegarde des blocs de mémoire écran. A ce sujet, se reporter à l'annexe 3 : la gestion de la mémoire écran.

La haute résolution est l'une des possibilités dont peut s'enorgueillir Oric-Atmos. Peu de micros de sa gamme peuvent rivaliser avec lui sur ce terrain. Pourtant, il est difficile de concevoir des jeux au graphisme acceptable en l'absence de résolution suffisante. Voici un programme qui offre à votre créativité l'outil qu'elle attendait pour donner libre cours aux fantasmes visuels les plus saugrenus. L'écran devient une toile où le trait est fin et précis. Et la palette des couleurs de votre micro n'attend qu'un geste pour aviver vos productions. Bien entendu, la postérité connaîtra cette œuvre puisqu'il ne suffira que d'une cassette pour en garder la copie intacte. Plus concrètement, vos dessins pourront être mis à contribution dans un programme de votre invention, un jeu d'aventure, par exemple, si vous avez dessiné le repaire sinistre du Dr Mabuse...



Déroulement

Pour dessiner, vous déplacez simplement le curseur à l'aide des touches :

- A pour monter,
- Z pour descendre,
- < pour aller à gauche,
- > pour aller à droite,

Le crayon est baissé tant que vous n'appuyez pas sur le SHIFT de droite. Si vous appuyez sur cette touche, le crayon se lève. Il n'y a plus de tracé. Cette possibilité peut également vous servir à effacer des traits sur lesquels vous repassez, à la manière d'une gomme.

Si vous appuyez sur le SHIFT de gauche, le curseur se déplace rapidement, ce qui est pratique pour tracer de longues lignes droites.

L'appui sur SHIFT entraîne la sortie du programme et l'effacement du curseur.

Pour définir une couleur en mode dessin, il suffit d'appuyer sur C. L'ordinateur demande alors quelle couleur est choisie puis si l'on désire l'imprimer en encre ou en fond. Dans le premier cas, seul le point est allumé à la couleur désirée. Dans le second, le fond sera à la couleur indiquée du curseur jusqu'à la fin de l'écran.

L'appui sur SHIFT + provoque le clignotement du curseur qui se déplace alors sans effacer le dessin : on est passé dans la procédure enregistrement. Le curseur sert à définir les bornes supérieures et inférieures du dessin en vue d'une sauvegarde sur cassette. Pour définir le début, on place le curseur au-dessus du dessin puis on appuie sur D; de même pour la partie inférieure (on place le curseur sous le dessin) avec un appui sur F.

En cas d'erreur, il suffit d'appuyer sur DEL pour retourner à l'éditeur.

Pour enregistrer le dessin sur cassette, il faut appuyer sur E. Si aucune borne n'a été définie, on sauve tout l'écran, soit 8 K.

Rappel des commandes :

— Mode dessin (curseur fixe) :

- SHIFT droit + touche de déplacement = déplacement du curseur seul ou effacement.
- SHIFT gauche + touche de déplacement = dessin rapide :
- C = couleurs :
- SHIFT + = passage en mode sauvegarde.
- SHIFT / = sortie du programme.

— Mode sauvegarde (curseur clignotant) :

- D mémorise l'adresse du début de la partie à sauvegarder.
- F mémorise l'adresse de fin de la partie.
- E passe à la procédure d'enregistrement.
- DEL = retour en mode dessin.

```
1 REM    DESSIN HIRES
2 REM
3 REM
4 REM
5 REM    MODE DESSIN
6 REM
9 REM    INITIALISATION
10 HIRES:POKE#26A,10
20 CURSET120,100,1:X=120:Y=100
25 GETA$:IFA$="!"THENCURSETX,Y,0:END
26 IF A$="+"THEN200
27 IFA$="C"THEN360
28 REM    GESTION CLAVIER ET ECRAN
29 REM
30 T=PEEK(#208)
40 IFPEEK(#209)=#A7 THEN CURSETX,Y,2
50 IFY=199THEN70
60 IFT=#AATHENY=Y+1
70 IFX=239THEN90
80 IFT=#94THENX=X+1
90 IFX=0THEN110
100 IFT=#8CTHENX=X-1
110 IFY=0THEN130
120 IFT=#AETHENY=Y-1
130 CURSETX,Y,1
140 IFPEEK(#209)=#A4THEN30
150 GOTO25
187 REM
188 REM MODE SAUVEGARDE
189 REM
190 D=#A000:F=#BF3F
200 T=PEEK(#208)
210 CURSETX,Y,2
220 IFY=199THEN240
230 IFT=#AATHENY=Y+1
```

```
240 IFX=239THEN260
250 IFT=#94THENX=X+1
260 IFX=0THEN280
270 IFT=#8CTHENX=X-1
280 IFY=0THEN300
290 IFT=#AETHENY=Y-1
300 CURSETX,Y,2
310 IF T=#B9THEN340
315 IFT=#AD THEN25
320 IFT=#99THEN345
325 IFT=#9ETHEN350
330 GOTO200
340 D=INT((Y*40)+(X/6))+#A000:GOTO200
345 F=INT((Y*40)+(X/6))+#A000:GOTO200
350 INPUT"NOM DU DESSIN":N$
355 CURSETX,Y,0:CSAVEN$,AD,EF:GOTO25
360 C=INT((Y*40)+(X/6))+#A000
365 INPUT"QUELLE COULEUR (DE 0 A 7)":CL
370 IF CL<0 OR CL>7THEN610
375 INPUT"FOND OU ENCRE (F/E)":FE$
380 IFFE$="E"THEN395
385 IFFE$="F"THENCL=CL+16:GOTO395
390 GOTO630
395 POKEC,CL:GOTO25
```

Commentaires

- 10 Suppression du bip clavier et du curseur.
- 20 Affiche le curseur en mode HIRES et définit ses coordonnées.
- 25 Attend qu'une touche soit enfoncée. Si c'est le SHIFT/, alors fin et effacement du curseur HIRES.
- 40 Si le SHIFT de droite est enfoncé, on efface le curseur dans sa dernière position : déplacement du curseur sans trace.
- 50-120 Teste les touches enfoncées. Si c'est Z, le curseur descend, à droite(>), à gauche(<), en haut (A).
- 130 Affiche le curseur dans sa nouvelle position.
- 140 Si le SHIFT de gauche est enfoncé, on reboucle directement au clavier sans passer par la pause clavier en ligne 25 (boucle en ?) : déplacement du curseur à grande vitesse.

- 15) Reboucle sur la pause clavier en 25.
- 19) Définition des bornes de sauvegarde.
- 2) Test clavier et affichage.
- 21) On inverse le curseur : s'il est noir, il devient blanc et vice-versa.
- 22) On teste si le curseur est arrivé sur un bord de l'écran. Si c'est le cas, on l'empêche de sortir.
- 23) On fait évoluer les coordonnées du curseur en fonction de la touche enfoncée. La séquence 22)-23) est répétée quatre fois entre 22) et 29), une fois pour chaque touche.
- 3) On réinverse le curseur dans sa nouvelle position.
- 31)-325 Test des 4 fonctions du mode sauvegarde.
- 33) Retour de boucle.
- 34) Calcul de l'adresse sur l'écran du curseur en fonction de ses coordonnées en X et Y afin de déterminer la borne supérieure de la partie à sauvegarder sur cassette. Même chose en 345 pour déterminer la borne inférieure.
- 35) On entre le nom du dessin à sauver.
- 355 On efface le curseur et on sauve le bout d'écran concerné. On retourne ensuite en mode dessin.
- 36) On détermine l'adresse du curseur à partir de ses coordonnées en X et Y pour mettre à sa place un ordre de couleur.
- 375 Détermine s'il s'agit de couleur d'encre ou de fond.
- 395 On affiche la couleur sur l'écran et on retourne au mode dessin.

Bloc-notes

Si on désire sauvegarder une page écran en haute résolution, il faut taper :

CSAVE « nom de la page », A#A), E#BF3F.

BOITE À MUSIQUE = ÉDITEUR MUSICAL



Qui n'a pas rêvé devant une vitrine d'instruments de musique ? Une de ces devantures où rutilent de fabuleux synthétiseurs, l'instrument à produire des sons le plus riche en possibilités de notre époque. Sans prétendre rivaliser avec les appareils spécialisés, vous pouvez obtenir des résultats tout à fait honorables grâce au synthétiseur et au haut-parleur présents dans votre Oric-Atmos.

Ce synthétiseur est un Général Instrument 8912 à trois canaux. Similaire à celui des appareils de jeux électroniques, il permet de

produire des notes musicales depuis les fréquences subsoniques jusqu'aux fréquences supersoniques : vous pouvez effectuer des tests d'écoute à votre entourage ou exécuter des morceaux de musique classique ou de rock.

Mais si vous disposez d'un connecteur DIN, vous pouvez brancher ATMOS sur votre chaîne Hi-Fi pour donner un concert improvisé grâce au programme qui suit.

Mode d'emploi

« Boîte à musique » vous permet de jouer des notes et de les voir s'afficher sur l'écran. Vous obtiendrez les tons et les demi-tons. Mais en cas de fausse manœuvre, il affiche aussi vos « couacs » ! Faisons connaissance avec votre nouveau clavier, tel qu'il est redéfini par le programme.

La première rangée du clavier, en commençant par le bas (Z à M), joue les notes.

La deuxième (S,D,G,H) joue les dièses.

Les flèches -> et <- vous permettent de changer d'octave. Vous pouvez ainsi couvrir, de façon continue, six octaves.

La touche RETURN vous offre le choix entre deux enveloppes : piano ou orgue.

Les flèches (haut) et (bas) augmentent ou diminuent la durée des notes.

Enfin, pour arrêter votre concert : touche ESC.

Amusez-vous bien !

Structure

20	Suppression du bip clavier
30-90	Redéfinition du clavier
100	Initialisation des variables
200-980	Scrutation du clavier
1000-1100	Musique
12000	Fin, présentation et mode d'emploi

```

2 '
3 'BOITE A NOTES
4 '
10 GOSUB12000
20 PRINTCHR$(6)
25 'initialisations
30 DIMNO$(100)
40 DIM NO(100)
45 FOR N=1 TO 100:NO(N)=-1:NO$(N)="COUAC"
46 NEXT N
50 DATA 90,1,DO,83,2,DO#,88,3,RE,68,4,RE#
55 DATA 67,5,MI,86,6,FA
60 DATA 71,7,FA#,66,8,SOL,72,9,SOL#,78
65 DATA 10,LA,74,11,LA#,77,12,SI
70 FOR I=1 TO 12
80 READ A:READ NO(A):READ NO$(A)
90 NEXT I
100 O=3:C=1:E=1:D=500
110 CLS:PAPER 4:INK 1
200 GET A#
205 ' tests clavier
210 A=ASC(A#)
220 IF A<8 THEN 200:IF A>13
AND A<48 THEN 200
230 IF A>55 AND A<66 THEN 200:
IF A>90 THEN 200
240 IF A=47 THEN CLS:END
300 IF A>65 AND A<91 THEN GOTO 1000
400 IF A=8 THEN 410 ELSE 500
410 IF O=0 THEN 200 ELSE O=O-1
500 IF A=09 THEN 510 ELSE 600
510 IF O=6 THEN 200 ELSE O=O+1
600 IF A=13 THEN 610 ELSE 700
610 IF E=1 THEN E=2 ELSE E=1
700 IF A=11 THEN 710 ELSE 800
710 IF D=32700 THEN 200 ELSE D=80+D
800 IF A= 10 THEN 810 ELSE 900
810 IF D=80 THEN 200 ELSE D=D-80
900 IF A>=48 AND A<=55 THEN 910 ELSE 900
910 IF A=48 THEN 200 ELSE C=VAL(A#)
980 GOTO 200
990 ' execution musique
1000 IF NO(A)<0 THEN 1001 ELSE 1005

```

```

1001 PING:WAIT 50:PRINT NO$(A):GOTO 200
1005 MUSIC 1,0,NO(A),0
1010 MUSIC 2,0,NO(A),0
1020 MUSIC 3,0,NO(A),0
1030 PLAY C,0,E,D
1038 CLS
1039 PLOT 16,13,10:PLOT 17,13,NO$(A)
1040 PLOT 16,14,10:PLOT 17,14,NO$(A)
1050 PLOT 37,26,STR$(O)
1100 GOTO 200
11999 'titre
12000 CLS:PAPER 4:INK 1
12005 PLOT 12,12,"BOITE A NOTES"
12010 PRINT01,14;"Commandes":PRINT
12020 PRINT"/ Pour arreter"
12030 PRINT"-> et <- augmentation"
12040 PRINT"et diminution octave"
12050 PRINT"RETURN change enveloppe"
12060 PRINT"fleche HAUT et BAS : duree"
12070 GET A$:RETURN

```

Commentaires

- 10 Appel du sous-programme de titre et mode d'emploi.
- 20 Suppression du bip clavier. Peut également s'obtenir avec PRINT CHR\$(6).
- NO\$ est le tableau des notes : NO\$(1) = DO, NO\$(2) = DO#.
- NO est celui des numéros de note : NO(1) = 1.
- 45 NO(N)=1 : cette instruction évite une sortie en erreur. En effet, en appuyant sur une touche du clavier autre que celle qui donne un son ou le modifie, l'instruction MUSIC de la ligne 1005 sortirait en erreur car elle n'accepte pas 0 en troisième paramètre. On teste donc ces touches en leur affectant une valeur négative qui, en ligne 1000, donnera le son *ping* et écrira le message « COUAC » à l'écran avant de ramener l'exécution en ligne 2000 pour l'attente d'une autre touche.
- 50-90 Remplissage du tableau de correspondance entre le code ASCII, la touche, le numéro de la note, le nom de la note.
- 100 Initialisation des variables : remise à zéro des paramètres des

instructions MUSIC et PLAY. O pour octave, C pour canal, E pour enveloppe et D pour durée.

200 Saisie du clavier au vol et retour en 200 tant qu'on n'appuie sur aucune touche.

220-230 Filtrage permettant de sélectionner les touches désirées pour jouer. Pour cela, on a recours au code ASCII dont on élimine les parties indésirables en renvoyant le programme à la saisie (l. 200).

240 Test d'arrêt par la touche ESC.

300 Test du clavier alphabétique.

400-410 Diminution d'octave par sollicitation de <-.

500-510 Augmentation d'octave par sollicitation de ->.

600-610 Changement de l'enveloppe par RETURN.

700-810 Augmentation et diminution de la durée.

900-910 Test clavier numérique avec renvoi en 200 après transformation en valeur numérique de A\$ par la fonction VAL ou directement, selon le cas.

1000-1001 Affichage « COUAC » et retour.

1005-1038 Exécution de la note selon les paramètres sélectionnés.

1039-1050 Affichage du nom de la note, en double hauteur au milieu de l'écran et de l'octave choisi en bas et à droite. Le troisième paramètre de PLOT permet d'afficher la note en double hauteur. Il est indispensable d'écrire la ligne 1040, répétition de la ligne 1030. Ce point est traité en détail dans le « bloc-notes » du programme KIM DES LETTRES.

Bloc-notes

Suppression du dé clic de clavier par deux instructions :

— PRINT CHR\$(6)

— POKE #26A,10.

Rappelons qu'en mode direct le même effet s'obtient par CTRL F.

ÉDITEUR DE DATA CODE MACHINE

Ce programme traduit le contenu d'une plage mémoire en DATA basic. Si vous souhaitez transcrire en DATA des caractères reprogrammés ou une routine en langage machine, sans avoir à la retaper à la main, il peut vous être d'un grand secours.

Déroulement

Après avoir tapé le programme, faites RUN pour transcrire les DATA en langage machine. Après quoi, le programme en langage machine se trouvera entre les adresses #400 et #489. Vous le sauvez sous forme de langage machine en tapant CSAVE« »A#400,E#489. Si toutefois vous souhaitez sauvegarder le programme en langage BASIC, il suffit de taper CSAVE“ “. Mais il faut savoir qu'après RUN, il ne vous sera plus d'aucune utilité.

Mode d'emploi

Commencer par déterminer le nombre d'octets à mettre en DATA. Cela s'obtient en faisant POKE 5, nombre d'octets. Ensuite déterminer l'adresse du premier octet à mettre en DATA. Pour cela faire DOKE #427, adresse. Puis taper DOKE 0, numéro première ligne de DATA et DOKE 2, adresse du BASIC. S'il n'y a aucun programme BASIC en mémoire, cette adresse sera #501. Par contre, s'il y en a un et que vous souhaitez le conserver, vous aurez l'adresse du BASIC en tapant PRINT DEEK (#9C). Il suffit ensuite de taper CALL#400 et vous aurez instantanément en mémoire les DATA par ligne de 8 et numérotés de 1 en 1, à partir du numéro que vous aurez précisé.

```
1 DATA#A2,#00,#A0,#02,#A5,#00,#91,#02
2 DATA#18,#69,#01,#85,#00,#A5,#01,#C8
3 DATA#91,#02,#69,#00,#85,#01,#A0,#04
4 DATA#A9,#91,#91,#02,#C8,#A9,#08,#85
5 DATA#04,#A9,#23,#91,#02,#C8,#BD,#00
6 DATA#04,#48,#4A,#4A,#4A,#4A,#20,#83
7 DATA#04,#91,#02,#C8,#68,#29,#0F,#20
8 DATA#83,#04,#91,#02,#C8,#E8,#E4,#05
9 DATA#F0,#0C,#C6,#04,#F0,#08,#A9,#2C
10 DATA#91,#02,#C8,#4C,#21,#04,#A9,#00
11 DATA#91,#02,#C8,#98,#A0,#00,#18,#65
12 DATA#02,#91,#02,#85,#06,#A5,#03,#69
```

```
13 DATA#00,#C8,#91,#02,#85,#03,#A5,#06
14 DATA#85,#02,#E4,#05,#D0,#94,#A0,#00
15 DATA#A9,#00,#91,#02,#C8,#91,#02,#A5
16 DATA#02,#18,#69,#02,#85,#9C,#A5,#03
17 DATA#85,#9D,#60,#F8,#C9,#0A,#69,#30
18 DATA#D8,#60
20 FORR=0TO#89:READA:POKE#400+P,A:NEXT
```

```
I400-489
0400: A2 00 LDX #000
0402: A0 02 LDY #002
0404: A5 00 LDA #000
0406: 91 02 STA ($02),Y
0408: 18 CLC
0409: 69 01 ADC #001
040B: 85 00 STA #000
040D: A5 01 LDA #001
040F: C8 INY
0410: 91 02 STA ($02),Y
0412: 69 00 ADC #000
0414: 85 01 STA #001
0416: A0 04 LDY #004
0418: A9 91 LDA #091
041A: 91 02 STA ($02),Y
041C: C8 INY
041D: A9 08 LDA #008
041F: 85 04 STA #004
0421: A9 23 LDA #023
0423: 91 02 STA ($02),Y
0425: C8 INY
0426: BD 00 B5 LDA #B500,X
0429: 48 PHA
042A: 4A LSR
042B: 4A LSR
042C: 4A LSR
042D: 4A LSR
042E: 20 93 04 JSR #0483
0431: 91 02 STA ($02),Y
0433: C8 INY
0434: 68 PLA
0435: 29 0F AND #00F
0437: 20 83 04 JSR #0483
```

```

043A: 91 02 STA ($02),Y
043C: C8 INY
043D: E8 INX
043E: E4 05 CPX $05
0440: F0 0C BEQ $044E
0442: C6 04 DEC $04
0444: F0 08 BEQ $044E
0446: A9 2C LDA #$2C
0448: 91 02 STA ($02),Y
044A: C8 INY
044B: 4C 21 04 JMP $0421
044E: A9 00 LDA #$00
0450: 91 02 STA ($02),Y
0452: C8 INY
0453: 98 TYA
0454: A0 00 LDY #$00
0456: 18 CLC
0457: 65 02 ADC $02
0459: 91 02 STA ($02),Y
045B: 85 06 STA $06
045D: A5 03 LDA $03
045F: 69 00 ADC #$00
0461: C8 INY
0462: 91 02 STA ($02),Y
0464: 85 03 STA $03
0466: A5 06 LDA $06
0468: 85 02 STA $02
046A: E4 05 CPX $05
046C: D0 94 BNE $0402
046E: A0 00 LDY #$00
0470: A9 00 LDA #$00
0472: 91 02 STA ($02),Y
0474: C8 INY
0475: 91 02 STA ($02),Y
0477: A5 02 LDA $02
0479: 18 CLC
047A: 69 02 ADC #$02
047C: 85 9C STA $9C
047E: A5 03 LDA $03
0480: 85 9D STA $9D
0482: 60 RTS
0483: F8 SED
0484: C9 0A CMP #$0A
0486: 69 30 ADC #$30

```

```

0488: D8 CLD
0489: 60 RTS

```

Commentaires

Avant d'étudier le programme, il est nécessaire de se pencher sur le fonctionnement de l'interpréteur BASIC.

Pour chaque ligne de BASIC, l'interpréteur la mémorise comme suit :

— dans les deux premiers octets, on trouvera l'adresse de la prochaine ligne;

— dans les deux suivants, le numéro de la ligne codé en hexadécimal;

— dans les octets suivants, la ligne elle-même. On trouvera dans cette ligne des codes hexadécimaux correspondant à une instruction Basic, suivis de nombres, s'il y a lieu, codés en ASCII. Une ligne de DATA sera donc composée de cette façon :

*Octets 0 et 1 : adresse de la prochaine ligne.

*Octets 2 et 3 : numéro de la ligne codé en HEXA.

*Octet 4 : code de l'instruction DATA : #91.

*Octet 5 : code ASCII du # (#23).

*Octets 6 et 7 : DATA hexadécimal codé ASCII sur 2 octets.

*Octet 8 : code ASCII de la virgule (#2C).

Le motif d'octets 5 à 8 se répète autant de fois qu'il y a de DATA dans la ligne, soit huit fois, sauf pour le dernier où l'on ne trouve pas de virgule à la fin. A la fin de chaque ligne, l'interpréteur pose un 0 sur le dernier octet pour indiquer que la ligne est finie.

400-414 On extrait le numéro de la première ligne stocké en 0 et 1 pour le mettre dans les octets 2 et 3 de la ligne en passant par l'accumulateur. On en profite pour y ajouter 1 en prévision de la ligne suivante.

416-41A On met dans l'octet 4 le code de l'instruction DATA, soit 91.

41C à 423 On place dans l'octet 5 le code ASCII du dièse (23).

426 On charge dans l'accumulateur le premier octet à transcrire en DATA pris dans une table pointée par X.

429 On sauvegarde cette valeur dans la pile.

42A à 42D On extrait les 4 bits de poids fort de l'octet.

42E On transcrit la valeur hexadécimal obtenue en code ASCII.

431 On stocke ce code ASCII dans l'octet 6 de la ligne.

434 On reprend dans la pile l'octet sauvegardé en ligne 429.

435 On en extrait les 4 bits de poids faible.

437 On transforme l'octet obtenu en code ASCII.

43A On stocke le code ASCII obtenu dans l'octet 7.

43D On incrémente X en vue de transcrire le prochain octet de la table.

43E-440 On regarde si on a transcrit tous les octets de la table. Si OUI, on va en 44E.

442-444 On teste si on a posé 8 DATA sur la ligne. Si OUI, on va en 44E.

446-44B On pose le code ASCII d'une virgule sur l'octet suivant et on reboucle en 421.

44E On pose un 0 sur le dernier octet de la ligne.

453-468 On calcule l'adresse de la prochaine ligne que l'on va stocker ensuite dans les octets 0 et 1 de la ligne précédente.

46A-46C On teste si on transcrit tous les octets de la table. Si OUI, on va en 46E, si NON, on reboucle en 402 où on débutera une nouvelle ligne.

46E Si l'on est arrivé ici, c'est que tous les DATA ont été transcrits. A ce moment-là, on doit indiquer à l'interpréteur que le fichier BASIC s'arrête ici. Cela se fait en mettant 2 zéros après la dernière ligne au lieu d'une adresse.

477-482 On remet en place le pointeur de fin de fichier BASIC qui se trouve en 9C et 9D.

482 Retour au moniteur BASIC.

483-489 Sous-programme de conversion HEXA-ASCII.

CONVERSION DE BASE NUMÉRIQUE

Il peut être utile d'avoir recours à un programme pour s'éviter de fastidieux calculs. Voici un exemple de routine qui vous rendra de précieux services. Il vous convertit à la demande n'importe quel chiffre dans n'importe quelle base.

Déroulement

A la question « NOMBRE A CONVERTIR », vous tapez une valeur suivie de RETURN. Puis vous précisez la base dans laquelle il est exprimé. Enfin, dans quelle base vous souhaitez le convertir. Si le chiffre indiqué n'existe pas dans la base d'origine, l'ordinateur vous le précise poliment. Dans le cas contraire, il vous indique sa valeur en décimal, le nombre de caractères qu'il occupe dans la nouvelle base et, enfin, le chiffre converti, le tout dans un temps record !

Structure

10-14	Déclaration de tableaux pour les calculs
20-80	Entrée des données et contrôle de validité
100-150	Calcul du chiffre en décimal et du nombre de caractère dans la nouvelle base
200-206	Calcul du chiffre dans la nouvelle base
210-214	Affichage du chiffre

```

0 '
1 '  CONVERSION DE BASE
2 '
10 DIM A(250)
11 DIM Z(250)
12 DIM X(250)
13 DIM A$(250)
14 DIM V(250)
15 CLS
16 '
17 'ENTRÉE DES DONNEES
18 '
20 INPUT "NOMBRE A CONVERTIR ?":N#
20 INPUT "DANS QUELLE BASE ?":B
21 IF B<2 THEN30

```

```

32 B= INT (B)
40 INPUT "NOUVELLE BASE      ?":NB
41 IF NB<2 THEN 40
42 NB= INT (NB)
50 L= LEN (N$)
60 FORI= 1 TO L
70 A$(I)= MID$(N$,I,1)
71 A(I)= VAL (A$(I))
72 IF A(I)=0 AND ASC (A$(I))<>#30
   THENA(I)=ASC(A$(I))-55
73 IF A (I)>=BTHENPRINTN$:" N'EXISTE PAS EN
   BASE":B:GOTO220
80 NEXT
95 '
96 '   CALCUL
97 '
100 FORI= 1 TO L
105 N= L-I
110 V(I)=A(I)*(B^N)
120 Z=Z+V(I)
130 NEXT
140 PRINT"EN DECIMAL:":Z
150 M= INT (LOG (Z)/LOG (NB) +1)
151 PRINT"LE NOUVEAU NOMBRE COMPORTE":
   INT(M):"CHIFFRES"
152 PRINT:PRINT
200 FORI= 1 TO M
201 P=INT((NB^(M-I))+0,5)
202 Z (I)= Z/P:X (I)= INT (Z (I))
203 Z= (Z (I)-X (I))*P
204 NEXT
205 Z (M)= INT (Z (M)*100+0,5)/100
206 X (M)= INT (Z (M))
210 FORI= 1 TO M
211 IF X (I)<10 THEN AF$(I)=STR$(X (I)):GOTO213
212 AF$(I)=" "+CHR$(55+X (I))
213 PRINTAF$(I)
214 NEXT
220 PRINT:PRINT:PRINT:PRINT
221 PRINT"ON RECOMMENCE (O/N) ?":GETP$
222 IF R$ ="O" THEN RUN
230 END

```

Commentaires

50 Toutes les entrées de données se font à l'aide d'un INPUT.

31 On vérifie que la base est supérieure ou égale à deux.

50-80 On dimensionne le chiffre à convertir pour qu'il soit compréhensible par la machine et on vérifie qu'il existe dans la base indiquée. Pour cela, on le décompose caractère par caractère et on convertit chacun d'eux en une valeur que l'ordinateur comprend. Exemple : 2F se décompose en 2 = 2 et F = 15 car pour des bases supérieures à dix, il n'existe pas de caractères correspondant à un chiffre supérieur à 10. Pour pouvoir l'exprimer, on le remplace par des lettres de l'alphabet. Par exemple : A = 10, B = 11, etc... Si le chiffre existe dans la base indiquée, on est en mesure de commencer les calculs.

100-130 Boucle qui additionne tous les coefficients établis plus haut, en tenant compte de la base, pour en extraire la valeur décimale.

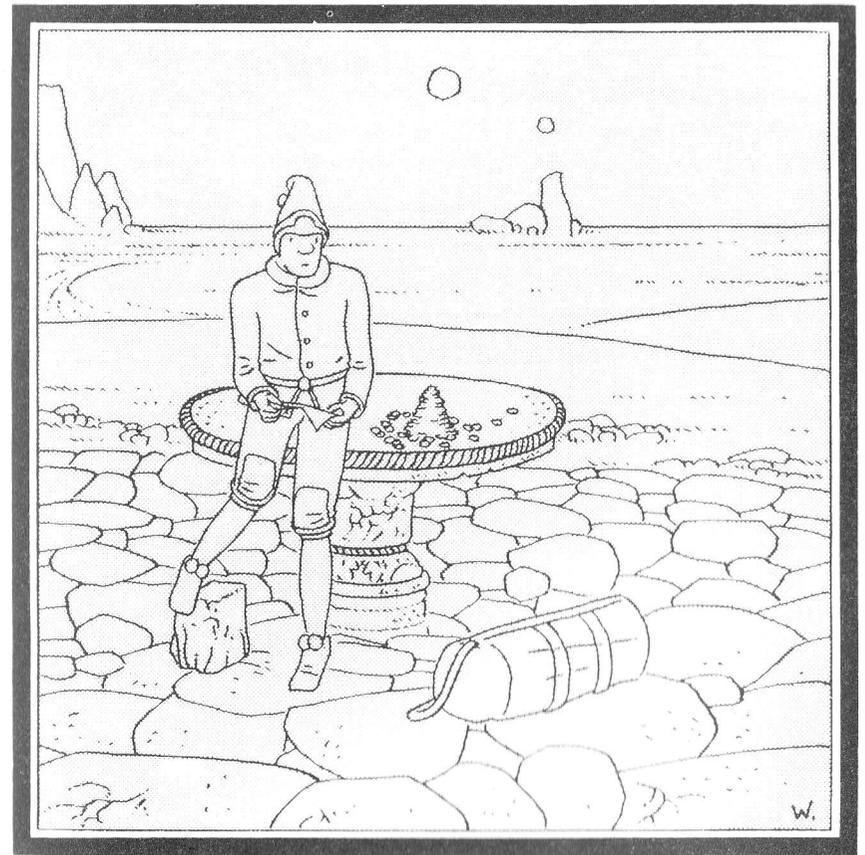
140 On affiche la valeur décimale.

150 A l'aide des LOG de la nouvelle base et de la valeur décimale, on établit le nombre de caractères que comportera le chiffre (LOG de base n = LOG décimal / LOG décimal de n).

200-204 On fractionne le chiffre décimal en puissances de la nouvelle base afin de déterminer les caractères qui composeront le nouveau chiffre.

210-214 Affichage du nouveau chiffre caractère par caractère.

IMPÔTS



Acceptez-vous de dire à Oric la vérité, rien que la vérité, toute la vérité sur votre prochaine déclaration ? Lui, en tout cas, s'engage à effectuer des calculs transparents. Il est également à votre service pour vous permettre de prévoir ce que vous auriez à payer si vous aviez plus d'enfants, si vous viviez en union libre plutôt que sous les liens sacrés du mariage... A moins qu'un abattement pour frais professionnels bien chiffrés ne vous rende la pilule moins amère. Oric se pliera à toutes vos combinaisons et se montrera un partenaire plus compréhensif que votre percepteur. Remerciements à J. Delcourt, auteur de la version TO7 de ce programme.

Déroulement

Après RUN un dialogue commence entre vous et votre « feuille électronique d'impôt » : préfiguration de ce qui nous attend dans un proche avenir ? Possible. En attendant, vous admettez qu'il était difficile de prévoir absolument tous les cas dans un programme court. Un choix a été fait dans certains cas comme ceux des revenus non salariaux, des gardes d'enfants. De plus, les réglementations changent : les dispositions concernant la « majoration exceptionnelle » n'ont pas été intégrées. En dehors de ces détails, ce programme suit la norme de votre déclaration annuelle.

Structure

10-300	Calcul du revenu brut
500-650	Déductions
700-2240	Parts et calcul de l'impôt
3000-3070	Exonérations et tranches

```
1 '
2 '
3 ' IMPOTS
4 '
5 '
10 'Revenus imposables
20 CLS:PAPER2:INK0
30 PLOT17,1,"IMPOTS"
40 PRINT:PRINT:PRINT:PRINT
50 PRINT"Revenus salariaux:"
60 INPUT"Revenus du chef de famille:";S(1)
70 INPUT"Revenus du conjoint:";S(2)
80 INPUT"Revenus des Pers. a charge:";S(3)
90 FOR I=1 TO 3:D(I)=0.1*S(I)
91 IF D(I)<1800 AND D(I)>0 THEN D(I)=1800
92 IF D(I)>50900 THEN D(I)=50900
95 NEXT I
100 PRINT:PRINT "Frais PProfessionnels"
110 PRINT"SONT :":PRINT
120 FOR I=1 TO 3:PRINT " ";D(I):NEXT I
130 PRINT"Desirez-vous deduire des"
135 PRINT"frais Professionnels"
140 A# = KEY$: IF A#="" THEN 140
```

```
145 IF A#<>"O" AND A#<>"N" THEN 140
146 IF A#="" THEN 150 ELSE 210
150 PRINT:INPUT"Frais du chef de famille":
    D(1)
160 INPUT"D( conjoint: ";D(2)
180 INPUT"Des Personnes a charge";D(3)
210 FOR I=1 TO 3
220 REV=REV+0.8*(S(I)-D(I))
230 NEXT I
240 PRINT
250 PRINT"Autres revenus ";
260 INPUT RV2
270 REV=REV+RV2
280 PRINT:PRINT:
290 PRINT"Votre revenu brut ";
295 PRINT"est ";REV:PRINT"Francs"
300 GET A#
499 '
500 ' Deductions
501 '
510 CLS:PAPER 3:INK2
515 PRINT:PRINT"Deductions diverses"
520 PRINT:INPUT"Frais de garde ";FG
530 IF REV>165580 THEN FG=0
550 PRINT
560 INPUT" Habitation ";HA
570 PRINT
580 INPUT" Versements aux oeuvres";VO
590 IF VO<>0 AND VO<>0.01*REV
    THEN VO=0.01*REV
600 PRINT:PRINT:PRINT:PRINT
620 REV=REV-FG-HA-VO
630 PRINT"Votre revenu net imposable est de"
640 PRINTREV:PRINT"Francs"
650 GET A#
699 '
700 "Nombre de Part
701 '
1000 CLS:PAPER6:INK1
1010 PRINT:PRINT "Etes-vous..."
1020 PRINT:PRINT"Celibataire, divorce
    ou veuf..1"
1030 PRINT:PRINT"marie..2"
1050 PRINT:PRINT"Votre reponse:";
```

```

1060 INPUT R
1065 PRINT
1070 INPUT "Nombre d'enfants a charge ";N1
1075 PRINT
1080 PRINT "Nombre d'autres Personnes"
1085 INPUT "a charge";N2
1090 NP=R+.5*N1+.5*N2
1100 IF R=1 AND N1>0 THEN NP=NP+.5
1110 IF N1>2 THEN NP=NP+.5
1120 PRINT:PRINT "Votre nombre de Parts est:"
1130 PRINT " ";NP
1999 '
2000 'calcul de l'impot
2001 '
2020 FOR I=1 TO 10:READ LIM(I):NEXT I
2030 IF REV<LIM(2*NP-1)THEN 2220
2040 DIM TR(13)
2050 FOR I=1 TO 12:READ TR(I):NEXT I
2060 QF=REV/NP
2100 FOR I=1 TO 12
2110 IF QF<TR(I)THEN 2200
2120 PAYE=PAYE+(TR(I+1)-TR(I))*I/20
2130 NEXT I:STOP
2200 PAYE=PAYE+(QF-TR(I))*(I-1)/20
2210 PAYE=NP*PAYE
2220 PRINT:PRINT "Votre impot est de
2240 END
2999 '
3000 ' Exoneration
3010 DATA 25210,25280,28500,34950,41400
3020 DATA 47860,54310,60760,567210,73670
3025 '
3026 ' Tranches
3027 '
3040 DATA 12620,13190,15640,24740,31810
3050 DATA 39970,48360,55790,92970,127860
3070 DATA 151250,172040

```

Commentaires

Prenez votre feuille de déclaration pour une meilleure compréhension des calculs.

50-80 Entrée des revenus.

90 Tableau de la déduction de 10% pour chaque revenu.

91-92 Plafonds.

220 Calcul du nouveau revenu REV après déduction éventuelle de frais professionnels.

270-295 Calcul du revenu brut avec ajout éventuel d'autres revenus.

620 Calcul du revenu après déductions diverses.

640 Affichage du revenu imposable.

2020 Remplissage du tableau d'exonérations.

2030 Test entre revenus et exonérations.

2040-2110 Tableau des tranches et test du quotient familial — tranches.

2200-2220 Calcul de l'impôt.

3000-3070 Plafonds des exonérations et tranches. Pour réactualiser votre programme, il suffit de taper les données en remplacement de celles proposées par ce programme.

CRÉDIT

Vous souhaitez acheter un appartement, faire construire une maison, aménager votre résidence secondaire ou aider un ami à monter le financement de son voilier ? Il vous faut beaucoup d'argent. Le crédit s'impose. Mais lequel ? Comment emprunter ? Qui dit prêt d'argent, suppose intérêts, délai de remboursement, amortissement. Pour vous aider à y voir clair et vite dans vos prévisions, voici un programme qui effectuera les calculs sous tous les angles d'approche de votre problème personnel. Il vous suffira de lui donner les éléments de base pour qu'il réponde instantanément aux questions que vous vous posez. Les calculs de ce programme de J. Delcourt (que nous remercions) concernent le cas le plus courant d'un emprunt à mensualités constantes.

Déroulement

Un menu initial vous propose différents calculs :

- le nombre de mensualités,
- le montant d'une mensualité,
- le montant de l'emprunt,
- le taux d'intérêt.

Après avoir donné le numéro de votre choix, vous entrez les données réclamées par le programme. Quatre nombres interviennent :

- la somme empruntée ou capital (variable C),
- le taux d'intérêt annuel (I),
- le montant de chaque mensualité (M),
- le nombre de mensualités (N). Si vous empruntez sur quinze ans, le nombre N vaut 15×12 , soit 180.

Ce programme s'efforce de traiter les quatre problèmes qui se posent à l'emprunteur.

Problème 1 : j'ai besoin d'une certaine somme C, le taux d'intérêt I m'est imposé, et je peux rembourser M francs par mois : pendant combien de mois devrais-je rembourser ?

Problème 2 : j'ai besoin de la somme C, le taux d'intérêt I est donné et je souhaiterais avoir fini de rembourser en N mois : combien devrais-je payer par mois ?

Problème 3 : je suis prêt à payer M francs par mois, pendant N mois. Le taux I est donné : combien puis-je emprunter ?

Problème 4 : j'ai besoin d'une certaine somme C, et l'on me propose de verser M francs par mois, pendant N mois : quel taux d'intérêt me fera-t-on payer ?

Après chaque calcul est donnée la table d'amortissement de votre emprunt : pour chaque échéance sont déterminées la part de la mensualité M qui relève du remboursement du capital (le principal) et la part « perdue » qui relève des intérêts; vous verrez ainsi que cette dernière diminue de plus en plus.

Structure

10-130	Menu et aiguillage des choix
150-190	Problème 1
200-299	Problème 2
300-396	Problème 3
400-540	Problème 4
1000-2000	Amortissement
3000-3020	Sous-programme de changement de page

```
1 '  
2 '  
3 ' CREDIT  
4 '  
5 '  
10 CLS:PAPER2:INK0  
20 PLOT17,1,"CREDIT"  
30 FOR I=1 TO 6:PRINT:NEXT I  
40 PRINT"Souhaitez-vous calculer :"  
50 PRINT:PRINT  
60 PRINT TAB(5)"1-Le nombre de mensualites"  
65 PRINT:PRINT  
70 PRINT TAB(5)"2-Le montant d'une mensualite"  
75 PRINT:PRINT  
80 PRINT TAB(5)"3-Le montant de l'emPrunt"
```

```

85 PRINT:PRINT
90 PRINT TAB(5)"4-Le taux d'interet "
95 PRINT:PRINT
100 PRINT"Votre choix :";
110 A$=KEY$:IF A$=""THEN110
120 A=VAL(A$):IF A<1 AND A>4 THEN 110
130 ON A GOTO 150,200,300,400
149 '
150 'Nbre mensualites
155 CLS:PAPER5:INK0:PRINT
157 PRINT"Taux d'interet annuel"
158 INPUT"en Pourcentage :";I
160 PRINT:INPUT"Montant de l'emPrunt :";C
165 PRINT:INPUT"Montant d'une mensualite :";M
170 I=I/100:IM=(1+I)^(1/12)-1
175 N=-LOG(1-IM*C/M)/LOG(1+IM)
180 N=INT(N)
182 PRINT:PRINT
185 PRINT"Vous devrez rembourser en ";N;" mois"
190 GOSUB 3000:GOTO 1000
199 '
200 ' Montant mensualite
201 '
210 CLS:PAPER2:INK0
220 PRINT"Taux d'interet annuel"
230 INPUT"en Pourcentage :";I
240 PRINT:INPUT"Montant de l'emPrunt :";C
250 PRINT:INPUT"Nombre de remboursements ";N
260 I=I/100:IM=(1+I)^(1/12)-1
270 M=C*IM/(1-(1+IM)^(-N))
280 PRINT:PRINT
290 PRINT". Vous devrez rembourser:";
295 PRINT M ;:PRINT"francs Par mois"
296 GOSUB 3000:GOTO 1000
299 '
300 ' Montant emPrunt
301 '
310 CLS:PAPER6:INK4
320 PRINT"Taux d'interet annuel"
330 INPUT"en Pourcentage :";I
340 INPUT"Montant d'une mensualite ";M
350 INPUT"Nombre de mensualites ";N
360 I=I/100:IM=(1+I)^(1/12)-1
370 C=M*(1-(1+IM)^(-N))/IM

```

```

390 PRINT:PRINT:PRINT
390 PRINT"Vous Pouvez emPrunter:";PRINT
395 PRINT C;:PRINT"francs"
396 GOSUB 3000:GOTO 1000
399 '
400 'taux d'interet
401 '
410 CLS:PAPER3:INK4:PRINT
420 INPUT"Montant de l'emPrunt:";C
430 PRINT:INPUT"Montant d'une mensualite:";M
440 INPUT"Nombre de mensualites ";N
450 K=C/M
460 I = 1/K-K(N*N): 'Valeur approchee
470 Z =(1+I )^(-N)
480 F =(1-Z )/I -K
490 FF =N*Z /(1+I )*(I )+(Z -1)/(I *I )
495 I =I -F /FF
500 IF ABS(F)>10^-5 THEN 470
510 I=(1+I )^12-1:I=100*I:IM=I
520 PRINT:PRINT:PRINT"Le taux d'interet"
530 PRINT"est : "; I;
540 PRINT" %":GOSUB 3000:GOTO 1000
999 '
1000 ' Amortissement
1001 '
1010 CLS:PAPER0:INK3
1020 PRINT:PRINT
1030 C1=C
1040 Y=6
1050 PRINT:PRINT
1060 FOR K=1 TO N
1070 IT=C1*IM:PPAL=M-IT
1080 PRINT@1,Y;K):PRINT@10,Y:IT:PPAL
1090 C1=C1-PPAL :Y=Y+1
1095 IF Y=22 THEN GOSUB 3000:Y=6:CLS
2000 NEXT K
2010 GOSUB 3000:GOTO 10
2999 '
3000 'Tourne Page
3001 '
3010 PLOT15,24,"APPuisez sur une touche"
3020 GET A$:RETURN

```

17) IM est l'équivalent mensuel de l'intérêt annuel I.

17) Le nombre d'échéances N est arrondi à l'entier immédiatement inférieur. Il faut donc s'attendre à payer une dernière mensualité un peu plus forte que les autres.

27) Quelle formule lie les quatre données C, M, N et I ? Si on appelle C_k la fraction de capital restant à rembourser après la k-ième échéance, on a $C_0=C$ et $C_n=0$. Quel est maintenant le lien entre C_k et C_{k+1} ? Ce calcul est fait dans le sous-programme 1000 à chaque passage dans la boucle : la somme versée mensuellement M sert à payer les intérêts sur la valeur résiduelle C_k et à rembourser une partie du capital. On a donc :

$$C_{k+1} = C_k - (M - IM/k)$$

Ce qui peut aussi s'écrire :

$$C_{k+1} - M/IM = (1 + IM)(C_k - M/IM)$$

On en déduit donc :

$$C_n - M/IM = (1 + IM)^N (C_0 - M/IM)$$

Il ne reste plus qu'à poser $C_n=0$ pour obtenir la formule :

$$M = C \cdot IM / -(1 + IM)^{-N}$$

48)-49) La méthode de Newton a été appliquée à cette fonction notée F. La fonction dérivée est notée FF.



Le nombre de vos relations augmente ? Vous souhaitez conserver leurs noms, adresses, numéros de téléphone en lieu sûr ? Ce memento électronique vous rendra service. Que lui demanderez-vous ? Un nom, une liste de noms. Il vous proposera instantanément adresses, téléphones et tout autre renseignement que vous auriez pu y adjoindre. Ce carnet sera stocké sur votre cassette, dans un fichier. Vous pourrez le consulter et le renouveler régulièrement.

Déroulement

Un menu s'affiche et vous propose de :

- remplir un carnet,
- lire un carnet,
- consulter une fiche,
- ajouter une fiche,
- modifier une fiche,
- fin de travail.

Le choix fait, vous entrez dans une procédure dont vous ne pouvez sortir qu'en revenant au menu principal ci-dessus.

Tenez, remplissons ensemble votre carnet. Si c'est la première fois, le programme vous demande le nombre de noms que vous souhaitez y inscrire. Il doit, en effet, prévoir la place pour les ranger dans sa mémoire. Vous n'avez plus maintenant qu'à compléter les différentes rubriques de votre memento (nom, prénom, adresse, téléphone, etc...). Plus tard, lorsque vous souhaitez ajouter de nouveaux noms à un carnet déjà rempli, vous choisirez directement la procédure « ajouter une fiche ».

Un trou de mémoire vous amène à « interroger » votre memento électronique ? Choisissez la procédure « lire un carnet ». Les fiches apparaîtront dans l'ordre où vous les avez rentrées. Si vous connaissez le nom, choisissez plutôt la consultation par fiche pour accéder aux renseignements souhaités.

Un ami de perdu, dix de retrouvés ! De temps en temps, une remise à jour s'imposera. Les procédures « ajouter » ou « modifier » vous le permettront.

Structure

20-110	Titre et menu
150-199	Création du carnet
210-299	Lecture du carnet en mémoire vive
300-370	Consultation par nom
400-599	Ajouts et modifications
600-640	Message d'enregistrement
1000-1999	Enregistrement du carnet sur cassette
2000-2260	Lecture du carnet sur cassette

```
1 '  
2 ' CARNET D'ADRESSES  
3 '  
10 '  
20 CLS:PAPER7:INK0  
30 PLOT7,1,CHR$(10)+CHR$(18)  
+CHR$(1)+"CARNET D'ADRESSES"+CHR$(23)  
31 PLOT7,2,CHR$(10)+CHR$(18)  
+CHR$(1)+"CARNET D'ADRESSES"+CHR$(23)  
35 FOR I=1 TO 4:PRINT:NEXT  
40 PRINT:PRINT TAB(08)"1-Remplir un carnet"  
50 PRINT:PRINT TAB(08)"2-Lire un carnet"  
60 PRINT:PRINT TAB(08)"3-Consulter une fiche"  
70 PRINT:PRINT TAB(08)"4-Ajouter une fiche"  
72 PRINT:PRINT TAB(08)"5-Modifier une fiche"  
75 PRINT:PRINT TAB(08)"6-Fin de travail"  
80 PRINT:PRINT:PRINT"Votre choix":  
90 A$=KEY$:IF A$="" THEN 90 ELSE A=VAL(A$)  
100 IF ASC(A$)<49 AND ASC(A$)>54 THEN 90  
110 ON A GOTO 150,200,300,400,500,146  
140 '  
146 CLS:PRINT"Operations terminees"  
147 END:  
149 '  
150 ' Creation carnet  
151 '  
153 CLEAR :DIM NOM$(100,4)  
154 CLS:PAPER4:INK3  
155 INPUT"Nombre de noms a enregistrer":N  
160 FOR I=1 TO N  
170 PRINT"Numero":I:"(nom Prénom adresse tel"  
175 INPUT"NOM: ";NOM$(I,1)  
176 PRINT:INPUT"PRENOM: ";NOM$(I,2)  
177 PRINT:INPUT"ADRESSE: ";NOM$(I,3)  
178 PRINT:INPUT"TEL: ";NOM$(I,4)  
190 NEXT I  
191 GOSUB 600  
194 IF A$="0" THEN GOSUB 1000  
195 GOTO 20  
199 '  
200 ' lecture carnet  
201 '  
205 CLS:PAPER2:INK0  
210 PRINT"Ce carnet est-il : "
```

```

215 PRINT:PRINT"1-sur cassette?"
216 PRINT:PRINT"2-en memoire vive?"
217 PRINT:PRINT "Votre reponse: ";
218 A$=KEY$:IF A$="" THEN 218 ELSE A=VAL(A$)
219 ON A GOTO 2000,230
225 '
226 'Affichage
227 '
230 FOR I=1 TO N
235 PRINT"Numero":I
240 PRINT NOM$(I,1)+" "+NOM$(I,2)
242 PRINT NOM$(I,3)
243 PRINT NOM$(I,4)
250 NEXT I
255 PRINT
260 PRINT:PRINT"Appuyez sur une touche"
265 PRINT"Pour revenir au menu"
270 GET A$:GOTO 20
299 '
300 ' Consultation
301 '
305 CLS:PAPER 6:INK4
310 INPUT"Quel nom desirez-vous ";N$
320 FOR I=1 TO N
325 IF N$=NOM$(I,1)THEN 355
330 NEXT I
350 PRINT"Ce nom ne figure pas "
352 PRINT"dans le carnet":GOTO 360
354 PRINT NOM$(I,1)+" "+NOM$(I,2)
355 PRINT NOM$(I,3):PRINTNOM$(I,4)
360 GET A$
370 GOTO 20
399 '
400 'Ajout de fiche
401 '
405 CLS:PRINT"Entrez la nouvelle fiche"
410 N=N+1
420 PRINT"Numero":N;"(nom, prenom, adresse,
tel)"
430 PRINT:INPUT"nom: ";NOM$(N,1)
440 PRINT:INPUT"PRENOM: ";NOM$(N,2)
450 PRINT:INPUT"ADRESSE: ";NOM$(N,3)
460 PRINT:INPUT"TEL: ";NOM$(N,4)
470 PRINT:PRINT"Une autre fiche (O/N)"

```

```

480 A$=KEY$:IF A$=""THEN 480
485 IF A$<>"O" AND A$<>"N"THEN 480
490 IF A$="O" THEN 405 ELSE GOSUB 600
495 IF A$="O"THEN GOSUB 1000
496 GOTO 20
499 '
500 'Modification
501 '
510 CLS:PAPER3:INK4
520 INPUT"Quel nom souhaitez-vous":N$
530 FOR I=1 TO N
540 IF N$=NOM$(I,1) THEN 570
545 NEXT I
550 PRINT:PRINT"Ce nom ne figure pas "
551 PRINT"dans le carnet."
560 GET A$:GOTO20
570 PRINT:PRINTNOM$(I,1):PRINTNOM4(I,2)
580 PRINTNOM$(I,3):PRINTNOM$(I,4)
585 PRINT"Entrez les nouvelles donnees"
587 PRINT:INPUT"Adresse":NOM$(I,3)
588 PRINT:INPUT"Tel":NOM$(I,4)
590 GOSUB 600
595 IF A$="O" THEN GOSUB1000
597 GOTO 20
599 '
600 'Message
601 '
610 PRINT"Voulez-vous enregistrer "
615 PRINT"Ce nouveau carnet (O/N)?"
620 A$=KEY$:IF A$=""THEN620
630 IF A$<>"O" AND A$<>"N"THEN 620
640 RETURN
999 '
1000 'Enregistrement
1001 '
1010 PRINT"Quand le magnetophone est pret"
1020 PRINT"a enregistrer, "
1030 PRINT"appuyez sur une touche"
1035 PRINT"du clavier."
1040 GET A$
1050 STORE NOM$,"CARNET"
1060 PRINT:PRINT:
1070 RETURN
1999 '

```

```

2000 'Lecture d'un carnet enregistre
2001 '
2005 CLEAR:CLS:PAPER 5:INK2
2010 DIM NOM$(100,4)
2015 INPUT"Nombre de noms a lire":N
2020 PRINT"Appuyez sur une touche"
2030 PRINT"du clavier"
2040 GET A#
2050 RECALL NOM$,"CARNET"
2060 GOTO 20

```

Commentaires

3) Noter l'utilisation de PRINT @, instruction qui ne figure pas dans le vocabulaire d'Oric 1. Sur ce modèle on peut obtenir l'affichage à un endroit quelconque de l'écran avec PLOT X,Y « chaîne de caractères ». Il est également possible de jouer avec les PRINT et les espaces pour obtenir un menu centré.

9) L'utilisation de KEY\$ évite d'avoir à valider un choix par RETURN. On y gagne en vitesse d'exécution. La ligne 10) filtre les réponses indésirables. Elle n'accepte aucune réponse dont le code ASCII ne soit pas situé dans l'intervalle 49-56, c'est-à-dire entre les valeurs numériques 1 et 6 correspondant au choix du menu.

11) ON A GOTO... permet un aiguillage vers les différentes procédures.

16) Déclaration du tableau à deux dimensions permettant de stocker les renseignements.

20) 219 Lecture soit du tableau enregistré sur cassette, soit du tableau déjà en mémoire vive.

23) 25) Affichage des rubriques de chaque fiche.

105) Instruction d'enregistrement. Voir le « bloc-notes » ci-dessous.

2005 CLEAR supprime tout tableau NOM\$ en mémoire et permet de déclarer un nouveau tableau du même nom (l. 2010) prêt à recevoir les données en provenance de la cassette.

2020 Déclaration du tableau récepteur.

225) Instruction de lecture ou « rappel » des données stockées sur la cassette.

Bloc-notes

Atmos offre une nouvelle possibilité par rapport à Oric 1 : celle d'enregistrer ou de lire des tables de données sur cassette ou disquette. Rappelons qu'un tableau est une série de données de même type, rangées dans un certain ordre et étiquetées par un indice.

L'enregistrement est réalisé par STORE, STORE X, « bidule », S (S est optionnel). X est le nom du tableau de données à sauvegarder, tel qu'il a été déclaré par DIM dans le programme. « Bidule » est le nom que vous donnez à ce tableau avant de l'enregistrer sur la bande magnétique.

La marche à suivre est identique à celle de CSAVE avec la même possibilité de vitesse lente ou rapide selon la présence ou l'absence de l'indice S.

RECALL permet de réaliser la lecture d'un tableau déjà enregistré. Le processus est identique à celui du chargement par CLOAD. Auparavant, il convient de définir un nouveau tableau de même dimension ou plus grand afin de recueillir les données. C'est ce qui a été fait en ligne 2020 après un CLEAR qui évite une sortie en DD ERROR (redéfinition de tableau dans le même programme).

PRINT @ — Autre particularité d'ATMOS, l'existence de cette instruction d'affichage à n'importe quel endroit de l'écran dont on fournit les coordonnées. L'instruction est formulée PRINT @ X,Y;« liste d'éléments » ou PRINT @ X,Y; Z.

Les deux premières valeurs donnent les coordonnées, la troisième concerne le ou les éléments à afficher qui peuvent être des nombres ou des chaînes de caractères. Attention au ; séparant ces paramètres des précédents. Les valeurs de X doivent être comprises entre 0 et 39 (nombre de colonnes en mode TEXT); celles d'Y entre 0 et 27 (nombre de lignes). Si ces valeurs ne sont pas respectées, le message ILLEGAL QUANTITY apparaît.

Pour enrichir le programme, vous pouvez créer une rubrique « renseignements supplémentaires » en ligne 180.

ANNEXE 1

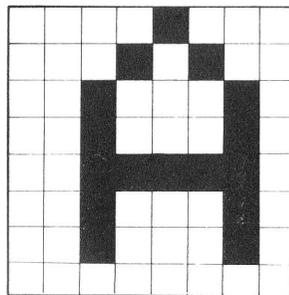
CARACTÈRES GRAPHIQUES ET ATTRIBUTS

Deux informations sont nécessaires pour définir la coloration d'un caractère sur l'écran :

- la couleur des points de premier plan,
- la couleur des points du fond.

Ces couleurs sont obtenues par des codes qui doivent figurer sur la même ligne d'écran que le caractère et à sa gauche.

Chaque code ainsi écrit occupe lui-même un rectangle 6*8 qui apparaît à l'écran sous la forme d'un pavé uniformément coloré.



« Oric » apparaît en rouge sur fond vert et « Atmos » s'inscrit en noir sur fond vert également car seule la couleur de forme a été redéfinie entre les deux mots.

Les emplacements contenant les trois codes ainsi que la fin de la ligne sont ici en vert.

Pour obtenir cet effet, il faut connaître les codes proposés dans le tableau ci-après.

En mode TEXT, il y a trois façons de positionner ces codes.

1 - Par PRINT

1Ø PRINT

CHR\$(146);CHR\$(129);« ORIC »;CHR\$(128);« ATMOS »

On peut remplacer ; par +

2 - Par PLOT

1Ø PLOT

3,15,CHR\$(18)+CHR\$()+« ORIC »+CHR\$()+« IMPORTANT »

Tableau des codes de couleurs

Par	FORME		FOND	
	PRINT	PLOT	PRINT	PLOT
Noir	128	Ø	144	16
Rouge	129	1	145	17
Vert	13Ø	2	146	18
Jaune	131	3	147	19
Bleu	132	4	148	2Ø
Magenta	133	5	149	21
Cyan	134	6	15Ø	22
Blanc	135	7	151	23

Caractères fixes ou clignotants, en simple ou double hauteur.

	PRINT	PLOT
Simple hauteur fixe	136	8
Double hauteur fixe	138	1Ø
Simple hauteur clignotant	14Ø	12
Double hauteur clignotant	142	14

Attention

En double hauteur, la partie supérieure doit toujours être écrite sur une ligne d'écran de numéro impair.

LA GESTION DE LA MÉMOIRE ÉCRAN

Qu'est-ce qu'une mémoire écran ?

Vu du BASIC, nous avons l'habitude de considérer celle-ci comme une feuille sur laquelle on viendrait écrire avec des instructions telles PRINT ou PLOT. Nous allons voir ce qu'il en est réellement. La mémoire est comparable à une succession de cases numérotées de 0 à 65 535 (sur Oric 1) et pouvant contenir un chiffre. Ce numéro entre 0 et 65 535 est appelé une adresse. C'est elle qui détermine sur quelle case mémoire on travaille. Le chiffre contenu dans l'adresse est codé en binaire (base 2). Chaque caractère qui compose un nombre binaire (0 ou 1) est appelé « bit ». Il ne peut être qu'égal à 0 ou à 1. Pour des raisons de commodité, tous les nombres contenus dans la mémoire seront codés sur huit bits, par exemple, 01011001 ou 11100100. Ainsi chacun de ces nombres sera appelé un « octet ». Il sera compris entre 0 et 255 et pourra ainsi prendre 256 valeurs différentes (2^8).

Comme il est pénible de manipuler des nombres binaires, l'habitude a été prise de les exprimer dans des bases plus compréhensibles telles que le décimal (base 10) ou l'hexadécimal (base 16). Comme l'hexadécimal est le plus commode, nous n'utiliserons principalement que cette base pour tout ce qui est adressage et affichage. Ainsi allons-nous la voir plus en détail.

En décimal, nous avons à notre disposition dix caractères qui nous permettent de coder tous les nombres que l'on désire. Ces caractères sont :

0 1 2 3 4 5 6 7 8 9

En base 16, nous disposons de 16 caractères qui sont :

0 1 2 3 4 5 6 7 8 9 A B C D E F

Ainsi, la valeur de A est 10

B = 11 D = 13 F = 15
C = 12 E = 14

16 en décimal s'exprime 10 en hexadécimal.

17 en décimal s'exprime 11 en hexadécimal.

Pour indiquer à la machine qu'un nombre est codé en hexadécimal, on placera un # devant.

1 = # 1 6 = # 6 11 = # B
2 = # 2 7 = # 7 12 = # C
3 = # 3 8 = # 8 13 = # D
4 = # 4 9 = # 9 14 = # E
5 = # 5 10 = # A 15 = # F
16 = # 10 17 = # 11 (etc.)

Des tables existent pour faire les conversions hexadécimal/décimal ou vice-versa. Mais vous pouvez aussi le faire à l'aide de votre Oric en tapant PRINT #NB pour obtenir la valeur décimale de NB, et PRINT HEX\$(NB) pour obtenir la valeur hexadécimale de NB ou encore en utilisant notre utilitaire de conversion de base numérique.

Pour conserver les informations qui sont portées à l'écran, l'ordinateur a besoin d'utiliser une partie de sa mémoire. Les octets de ce morceau de mémoire contiennent ce que l'on voit sur l'écran. La fonction d'affichage est ensuite réalisée par un circuit spécialisé appelé processeur graphique. C'est lui qui vient extraire le contenu de la mémoire écran et le transformer en image sur l'écran par l'intermédiaire de la prise Péritel.

Sur Oric, il y a deux modes d'affichage, le mode TEXT et le mode HIRES. Nous laissons à d'autres ouvrages le soin d'expliquer le mode HIRES (haute résolution) pour nous intéresser à la création et l'affichage de caractères en mode TEXT.

Affichage de caractères

En mode TEXT, chaque octet contenu dans la mémoire écran représente un caractère. Pour afficher une lettre à l'écran, il suffit donc de placer le code du caractère à une adresse écran donnée. Le code qui fait correspondre un caractère à chaque octet s'appelle le code ASCII (voir annexe).

C'est ensuite le générateur de caractères de l'Oric qui va aller prendre en « mémoire caractère » le graphisme de la lettre correspondant.

Comment est mémorisé un graphisme de caractère ?

Un graphisme de caractère tient dans une matrice 6 x 8. Il est représenté par huit octets de la manière suivante :

D⁷ D⁶ D⁵ D⁴ D³ D² D¹ D⁰

Nous avons vu qu'un octet était un groupe de huit bits. Nous savons que le graphisme est un ensemble de points. Lorsque l'on considère un point, on constate qu'il peut être allumé ou éteint; il ne peut prendre que deux états. Un bit ne pouvant prendre que deux états (0 ou 1), on peut comparer un point et un bit. Si maintenant on regroupe les points par barres de huit, chaque barre sera similaire à un octet.

En posant la règle :

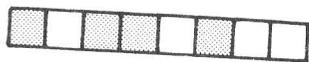
bit = 0 = point éteint,
bit = 1 = point allumé,

nous sommes en mesure de composer des graphismes.

Par exemple :

11 ou # B
1 0 1 1

4 ou # 4
0 1 0 0



Un graphisme de caractères sera donc huit octets composés de la sorte et empilés de manière à former une matrice.

Remarque : sur Oric, seuls les six premiers bits (en partant de la droite) sont portés à l'écran. Les deux derniers n'ont donc pas d'importance et n'ont aucune signification en mode TEXT.

Comment déterminer l'adresse du graphisme d'un caractère

Les huit octets relatifs à un graphisme se suivent et sont regroupés dans un espace mémoire prévu à cet effet.

Sur Oric 1, cet espace mémoire part de #B400 et va jusqu'à #BB7F; sur Atmos de #BB80 à #BFE0. Les graphismes y sont rangés dans l'ordre du code ASCII. Chaque graphisme prenant huit

octets, pour obtenir l'adresse du premier octet d'un caractère, il suffit de taper :

PRINT HEX\$ (#B400 + (A8)).

« A » étant le code ASCII du caractère. Les sept autres octets qui définissent le graphisme sont aux sept adresses suivantes, par exemple,

HEX\$ (#B400 + 32 8) = #BN500.

32 étant le code ASCII d'un espace, on pourra constater que dans l'espace mémoire allant de #B500 à #B507, on ne trouve que des zéros (FOR R = 0 TO 7 :PRINT PEEK (#B500 + R) :NEXT R).

Si maintenant on tape POKE #B500,1, on pourra constater que l'écran se couvre de points. Pourquoi ? Parce que l'on vient de remplacer un des zéros qui composaient le caractère d'espace par un 1 :

1 = 0 0 0 0 0 0 0 1 =

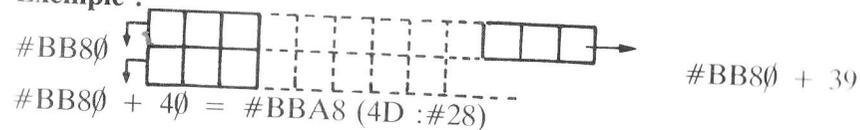
L'écran vide étant couvert de caractères d'espace, cela explique le fait qu'il soit maintenant couvert de points. Nous venons donc de modifier un caractère, et de la même façon, nous pourrions modifier tous les caractères que nous désirons, au gré de notre fantaisie.

Comment afficher un caractère sur l'écran

Nous avons vu plus haut que le contenu de l'écran était stocké dans un espace mémoire appelé « mémoire écran ». Sur Oric 1, cet espace mémoire part de #BB80 et va jusqu'à #BFE0. Pour afficher un caractère, il suffit donc de taper POKE AD,ASC, où AD est une adresse comprise entre #BB80 et #BFE0, et où ASC est le code ASCII du caractère à afficher.

L'écran est rangé de façon linéaire, c'est-à-dire que lorsque l'on arrive au bout d'une ligne, on passe à la suivante, juste en-dessous.

Exemple :



Chaque ligne est composée de quarante caractères.

Pour vous aider, tapez la ligne :

FOR R = 0 TO 79 :POKE #BB80 + R, 65 :NEXT

65 est le code ASCII du « A ». Votre compréhension sera immédiate.

Rappel : POKE AD,DO place la donnée DO à l'adresse AD. PEEK(AD) renvoie la donnée qui se trouve à l'adresse AD.

Après ces quelques explications, vous voilà en mesure de taper les programmes qui suivent et de les comprendre. Pour les routines en assembleur, il est impossible de donner des explications complètes en quelques pages. Si le langage machine vous a séduit, nous vous conseillons de faire l'achat d'un ouvrage spécialisé (références ci-dessous). Toutefois, nous vous avons mis le listing correspondant à chaque programme utilisant des routines en langage machine. Ainsi vous pourrez, si vous le désirez, l'étudier et le comprendre. Ceci s'adresse, bien sûr, à ceux qui possèdent un livre traitant de l'assembleur. Nous espérons que ces quelques programmes vous apporteront du plaisir et vous aideront à mieux cerner les problèmes qui se posent lors de la conception d'un programme.

Quelques ouvrages de référence pour vos débuts en Assembleur :

Le guide pratique de l'ORIC-ATMOS

Du Basic au langage machine, M. Bussac et R. Lagoutte, éditions Cedic Nathan.

La programmation du 6502, Rodney Zakcs, éditions Sybex.

L'assembleur facile du 6502 et du 6510, V. Monteil, éditions Eyrolles.

ANNEXE 3

Topographie de la mémoire vive d'ORIC-ATMOS

Adresses en mode texte

MODE TEXT		MODE HIRES	
16 K FFFF	ROM	16 K FFFF	ROM
C0000		C0000	
40000	Zone inoccupée	40000	Zone inoccupée
3FE00		BFE00	ÉCRAN
3B800	Caractères alternés	BB800	Caractères alternés (2 ^e clavier)
38000	Caractères standards	B8000	Caractères standards (1 ^{er} clavier)
34000	Zone utilisée en BASIC si GRAB est exécuté	B4000	Programmes BASIC
18000	Programmes BASIC	98000	
05000	PAGE 4	05000	PAGE 4
04000	PAGE 3 Adresses entrée/sortie	04000	PAGE 3 Adresses entrée/sortie
03000	PAGE 2 Variables en mode RUN	03000	PAGE 2 Variables en mode RUN
02000	PAGE 1 Pile	02000	PAGE 1 Pile
01000	PAGE 0 Utilisée	01000	PAGE 0 Utilisée
00000		00000	

ANNEXE 4

Table des codes ASCII

Codes 0-31

Code	Caractère	Code CTRL
0	Nul	
1	Copie	CTRL-A
2		
3	Break	CTRL-C
4	Impression en double ligne	CTRL-D
5		
6	Déclat de touche	CTRL-F
7	Clochette (Ping)	CTRL-G
8	Espacement arrière (Curseur vers la gauche)	CTRL-H
9	Curseur vers la droite	CTRL-I
10	Changement de ligne (Curseur vers le bas)	CTRL-J
11	Curseur vers le haut	CTRL-K
12	Effaçage écran	CTRL-L
13	[RETURN]	CTRL-M
14	Effaçage ligne	CTRL-N
15	Invalidation écran	CTRL-O
16		
17	Curseur	CTRL-Q
18		
19	Écran	CTRL-S
20	Capitales (majuscules)	CTRL-T
21		
22		
23		
24	Annulation ligne	CTRL-X
25		
26		
27	[esc] (changement de code)	
28		
29		
30		
31		
32	Espacement	

CODE	CARACTÈRE STANDARD	CARACTÈRE ALTERNÉ	CODE	CARACTÈRE STANDARD	CARACTÈRE ALTERNÉ
33	!	33	71	G	71
34	"	34	72	H	72
35	#	35	73	I	73
36	\$	36	74	J	74
37	%	37	75	K	75
38	&	38	76	L	76
39	'	39	77	M	77
40	(40	78	N	78
41)	41	79	O	79
42	*	42	80	P	80
43	+	43	81	Q	81
44	,	44	82	R	82
45	-	45	83	S	83
46	.	46	84	T	84
47	:	47	85	U	85
48	0	48	86	V	86
49	1	49	87	W	87
50	2	50	88	X	88
51	3	51	89	Y	89
52	4	52	90	Z	90
53	5	53	91	[91
54	6	54	92	,	92
55	7	55	93]	93
56	8	56	94	^	94
57	9	57	95	_	95
58	:	58	96	(96
59	;	59	97	a	97
60	<	60	98	b	98
61	=	61	99	c	99
62	>	62	100	d	100
63	?	63	101	e	101
64	@	64	102	f	102
65	A	65	103	g	103
66	B	66	104	h	104
67	C	67	105	i	105
68	D	68	106	j	106
69	E	69	107	k	107
70	F	70	108	l	108

Code effet

- 128 premier plan noir (texte/graphiques)
- 129 premier plan rouge (texte/graphiques)
- 130 premier plan vert (texte/graphiques)
- 131 premier plan jaune (texte/graphiques)
- 132 premier plan bleu (texte/graphiques)
- 133 premier plan magenta (texte/graphiques)
- 134 premier plan cyan (texte/graphiques)
- 135 premier plan blanc (texte/graphiques)
- 136 premier plan noir (texte/graphiques)
- 137 caractères graphiques
- 138 caractères en hauteur double (texte)
- 139 caractères en hauteur double (graphiques)
- 140 caractères clignotants (texte)
- 141 caractères clignotants (graphiques)
- 142 caractères clignotants en hauteur double (texte)
- 143 caractères clignotants en hauteur double (graphique)
- 144 fond noir
- 145 fond rouge
- 146 fond vert
- 147 fond jaune
- 148 fond bleu
- 149 fond magenta
- 150 fond cyan
- 151 fond blanc

ORIC-ATMOS, VOS PROGRAMMES

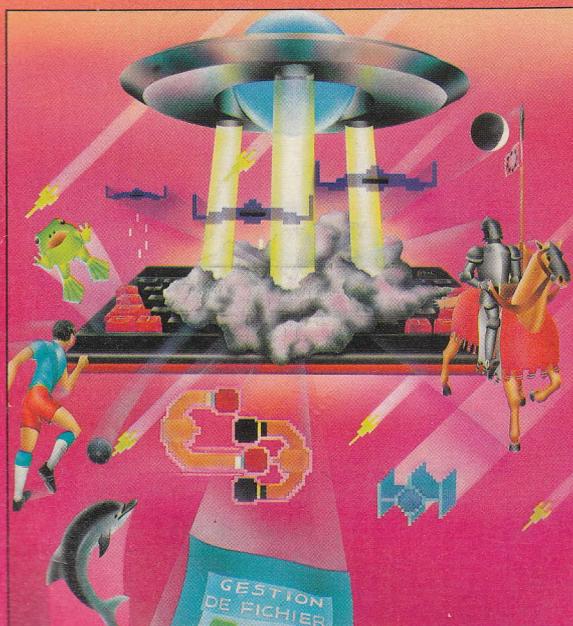
BASIC + LANGAGE MACHINE

Michel Bussac - Gil Espeche

Des jeux de réflexes pour les amateurs de sensations fortes, des spectacles visuels et sonores, des jeux de hasard et de mémoire, des jeux pour apprendre et des jeux pour créer.

Ce livre propose aux utilisateurs de ORIC-1 et ORIC-ATMOS, 26 programmes variés et originaux, faciles à transcrire.

Aux amateurs de programmation, il offre quelques incursions dans le langage machine pour des réalisations particulièrement spectaculaires.



* *

291307

ISBN - 2 - 7124 - 1510 - 8