

Jean-Marc NASR

François MANCHON

F A I T E S  
V O S J E U X

*sur*

P B  
Z O O



---

**FAITES VOS JEUX  
SUR PB 700**



## CHEZ LE MEME EDITEUR

- CANAL - *Parler L.S.E. et apprendre à l'utiliser* - 160 p. ; 1983, (coll. Micro-ordinateurs).
- DELANNOY - *Initiation à la programmation* - 192 p. ; 1984.
- DELANNOY - *Apprendre à programmer en Basic* - 272 p. ; 1984.
- DE GEETER - *FORTH pour micros* - 192 p. ; 1984, (coll. Micro-ordinateurs).
- HERNANDEZ - *PASCAL par l'exemple* - 152 p. ; 1983, (coll. Micro-ordinateurs).
- KRUTCH - *Expériences d'intelligence artificielle en Basic* - 128 p. ; 1984, (coll. Micro-ordinateurs).
- MONTEIL - *Premiers pas en LOGO* - 104 p. ; 1984, (coll. Micro-ordinateurs).
- SAGUEZ - *Télécommande avec votre micro-ordinateur* - 136 p. ; 1983, (coll. Micro-ordinateurs).
- TEJA - *Apprenez à parler à votre ordinateur* - 168 p. ; 1983, (coll. Micro-ordinateurs).
- VULDY - *Graphisme 3D sur votre micro-ordinateur* - 128 p. ; 1985, (coll. Micro-ordinateurs).

# FAITES VOS JEUX SUR PB 700

par

**Jean-Marc NASR  
François MANCHON**



61, boulevard Saint-Germain – 75005 Paris  
1985

Si vous désirez être tenu au courant de nos publications, il vous suffit d'adresser votre carte de visite au :

Service «Presse», Éditions EYROLLES  
61, Boulevard Saint-Germain,  
75240 PARIS CEDEX 05,

en précisant les domaines qui vous intéressent.  
Vous recevrez régulièrement un avis de parution  
des nouveautés en vente chez votre librairie  
habituel.

«La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les «copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective» et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite» (alinéa 1<sup>er</sup> de l'article 40).»

«Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal.»

© Éditions EYROLLES, 1984

# Avant-propos

*Ce livre donne un aperçu des possibilités ludiques du PB 700 dont nous avons cherché à tirer le plus grand parti des aptitudes graphiques.*

*Ainsi, dans un micro-ordinateur qui tiendrait dans une poche, vous pouvez vous adonner à vos jeux préférés allant des classiques Master Mind et Othello — où le PB 700 sera un partenaire coriace — aux plus modernes "escalier infernal" et "Rallye" que nous vous laisserons découvrir.*

*Les quatre premiers chapitres sont consacrés aux jeux: nous vous en présentons les listings expliqués et commentés.*

*Le dernier chapitre est plutôt orienté vers les graphismes et propose deux outils de dessin.*

*Faites vos jeux sur PB 700... et bon divertissement!*

LES AUTEURS

VII

# Table des matières

<b>Avant-propos</b>	VII
<b>1. Réflexion</b>	1
Master Mind	1
Bataille navale	4
Pendu	16
Othello	19
Solitaire	33
<b>2. Hasard</b>	39
Courses de puces	40
Loterie	43
Jeu de Yam	49
<b>3. Adresse</b>	59
Mur de briques	59
Ping-Pong	62
Tir forain	65
Tir balistique	69
Rallye	74

<b>4. Aventure</b>	79
Champ de mines	84
Escalier infernal	88
Mission suicide	92
Vol spatial	96
Labyrinthe	
<b>5. Graphismes</b>	101
Dessin	102
Police de caractères	104

# 1

## Jeux de réflexion

### ***Master Mind***

"Faites vos jeux sur PB 700" ne serait pas complet sans le célèbre et indispensable MASTER MIND. Ce jeu permet d'allier logique et intuition.

Rappelons-en brièvement les règles : vous devez découvrir une combinaison de quatre caractères. Vous disposez de six couleurs "R" rouge; "B" bleu; "J" jaune; "V" vert; "N" noir et "O" orange. Le caractère " " (touche SPC) est possible. La combinaison est composée aléatoirement par le PB 700. Il existe 2 401 possibilités différentes.

Vous proposez une combinaison : l'ordinateur vous indique le nombre de bonnes couleurs, et le nombre de couleurs bien placées.

Vous avez sept coups pour déterminer la combinaison. Si vous ne la trouvez pas, le micro-ordinateur vous la livre à la fin du jeu.

Installez-vous bien confortablement, concentrez-vous, déjouez les pièges du PB 700 et vous deviendrez le MASTER MIND.

## **Structure du programme**

- 1-13 : Présentation.
- 15-30 : Construction de la combinaison secrète.
- 60-70 : Entrée au clavier.
- 75-80 : Affichage de la combinaison proposée.
- 100-135 : Recherche des lettres bien placées.
- 140-175 : Recherche des lettres mal placées.
- 185 : Affichage du nombre de lettres mal et bien placées.
- 190 : Le joueur a-t-il gagné?  
Notez le DRAW qui dessine un accent sur le "e" de gagné.
- 195 : Retourner en 50 si le jeu n'est pas fini.
- 200-210 : Conclusion et retour au début.

## **Variables utilisées**

- C\$ : Ensemble des sept lettres possibles.
- K\$ : Combinaison secrète.
- N : Nombre de coups d'essai.
- E\$ : Combinaison proposée au clavier.
- P\$ : Copie altérable de K\$.
- B, M : Nombre de bien et mal placés.

## **MASTERMIND**

```
1 REM MASTERMIND
4 C$=" BONJUR"
10 CLS :PRINT :PRINT "----MASTERMIND
-----";C$ .
13 FOR I=1 TO 500:NEXT I
15 CLS :K$=""
20 FOR I=1 TO 4
```

```

25 K$=K$+MID$(C$, 1+INT(RND*2), 1)
30 NEXT I
35 DRAW(87,0)-(87,31)
40 N=0
50 REM Boucle de jeu
55 N=N+1
60 LOCATE 0,0:PRINT "???? M B"
65 LOCATE 0,0:INPUT "",E$
70 IF LEN(E$)<>4 THEN 60
75 IF N<4 THEN LOCATE 0,N ELSE LOCATE
    11,N-4
80 PRINT E$;
100 REM Tests combinaison proposee
105 P$=K$
110 B=0:REM Rech. bien Places
120 FOR I=1 TO 4
125 IF MID$(E$,I,1)<>MID$(P$,I,1) THEN
    135
130 B=B+1:E$=LEFT$(E$,I-1)+"*"+RIGHT$(E$,4-I)
133 P$=LEFT$(P$,I-1)+"*"+RIGHT$(P$,4-I)
    )
135 NEXT I
140 M=0:REM Rech. mal Places
145 FOR I=1 TO 4
150 L$=MID$(E$,I,1)
155 FOR J=1 TO 4
160 IF I=J THEN 170
165 IF L$<>MID$(P$,J,1) THEN 170
167 P$=LEFT$(P$,J-1)+"-"+RIGHT$(P$,4-J)
    )
169 M=M+1:GOTO 125
170 NEXT J
175 NEXT I
185 PRINT M;B;
190 IF B=4 THEN BEEP 1:LOCATE 0,0:PRIN
    T "Gagne ! ":"DRAW(36,1)-(37,0):GOT
    O 210

```

---

```
195 IF NK2 THEN 50
200 REM Fin du jeu
205 LOCATE 0,0:PRINT K$;" Perdu";:BEEP

210 IF INKEY$="" THEN 210 ELSE 15
```

### **Bataille navale**



Nous avons voulu rénover l'intérêt de ce jeu, un classique sur les micro-ordinateurs en lui donnant trois caractéristiques particulières.

Tout d'abord nous avons voulu profiter de la taille relativement importante de l'écran pour supprimer totalement l'usage du papier au cours du jeu : toutes les informations utiles sont résumées sur l'écran sous une forme graphique agréable.

---

Ensuite, vous pourrez constater que l'ordinateur, s'il tire généralement au hasard (mais jamais deux fois au même endroit !), par contre ne lache pas sa proie une fois touchée et l'achève sans pitié !

Enfin, dans le but d'accélérer le jeu, la machine détecte d'elle-même si ses coups tombent dans l'eau. C'est uniquement dans le cas contraire qu'elle vous demandera une précision : si votre bateau est touché ou coulé. Naturellement, rien ne vous empêche de tricher en lui mentant délibérément, mais ne vous étonnez pas, si elle s'en aperçoit, de voir apparaître un chapelet d'injures à l'écran !

### **Comment jouer**

**1.** Avant la bataille, le programme vous demande de placer vos bateaux sur votre grille de jeu. Le programme vous indique le nombre de cases du bateau à placer et vous demande :

Origine?

Répondez alors par les coordonnées de la case où vous voulez placer une extrémité du bateau :

**Exemple:**

Origine?23

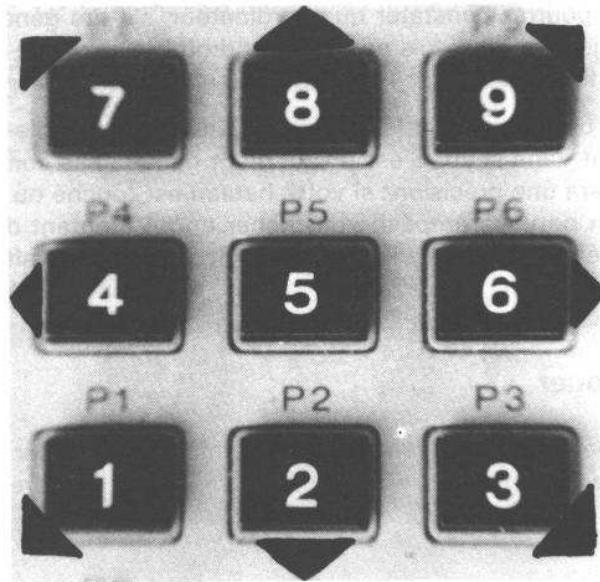
signifie : colonne 2, ligne 3.

Ensuite, répondez à la question :

Direction?

par un chiffre de l'ensemble 2, 4, 6, 8 conformément à la grille page suivante :

Le programme vérifie que vous n'essayez pas de placer deux bateaux l'un sur l'autre. Par contre, il ne vérifie pas que les bateaux ne se touchent pas.



Cela dit, évitez de le faire, car la règle du jeu l'interdit, et si le programme s'en aperçoit plus tard, il vous abreuvera d'insultes !

**2.** Lorsque tous vos bateaux sont en place, patientez quelques instants pour que l'ordinateur place les siens. Un "BEEP" vous préviendra que le jeu peut commencer.

**3.** La bataille. C'est vous qui tirez le premier. Vous répondez à la question :

?

A vous

en spécifiant une seule case de la même manière qu'en 1.

Vos tirs seront indiqués à l'écran, qu'ils aient fait mouche ou qu'ils tombent à l'eau.

Bonne chance... et méfiez-vous, le programme est très efficace.

## **Structure du programme**

- 1-10 : Initialisations. Présentation.
- 15-40 : Construction du damier du joueur.
- 45-70 : Construction du damier de la machine.
- 75-85 : Initialisations de certaines variables.
- 105-145 : Tir du joueur.
- 155-160 : Au tour de la machine.
- 170-185 : La machine joue "au hasard" (mais jamais deux fois au même endroit).
- 200-220 : Affichage du Tir de la machine et détection des coups dans l'eau.
- 222-275 : Le coup n'est pas dans l'eau, la machine demande si c'est touché ou coulé.
- 315-385 : Fin de la partie.
- 400-620 : La machine a touché un bateau adverse.
  - 407-480 : Elle recherche son orientation.
  - 450-480 : Une fois l'orientation connue, elle progresse dans un sens tant que ses tirs font mouche.
  - 485-520 : Si un de ces tirs est à l'eau, la machine change de sens.
  - 600-620 : Si la machine détecte une tricherie, elle réagit et considère le bateau comme coulé.
- 700-850 : Le joueur a fait mouche, le bateau est-il touché ou coulé ?
- 1000-1135 : Sous-programme demandant au joueur de placer un bateau de longueur "L" cases.  
Ce sous-programme vérifie que le joueur n'essaye pas de placer deux bateaux l'un sur l'autre, ou un bateau en dehors du damier. Le cas échéant, les lignes 1100 à 1135 effacent ce qui a été dessiné par erreur.
- 1200-1299 : Dessin d'une grille à l'abscisse "Z".
- 1300-1520 : Sous-programme permettant au programme de placer un bateau de "L" cases, en vérifiant que les bateaux ne se touchent pas.

## Variables utilisées

D\$(8,8)\*1 : Damier du jeu machine.

“ ” → case vide (l'eau)

“ X ” → élément du bateau

“ M ” → élément de bateau touché (Mort).

Z: Paramètre utilisé par le sous-programme en 1200, qui localise la grille 8 × 8 à tracer.

L: Longueur du bateau que l'on place.

X, Y, X0, Y0, X1, Y1: Coordonnées d'une case de D\$.

DX, DY, AX, AY: Déplacements autour d'une case (X,Y).

A, B: Coordonnées “DRAW” du centre d'une case (X,Y).

O, CS: Variables d'entrée:

O = 10\*X+Y

C\$ = “T” (-ouché) ou “C” (-oulé).

M, P: Décompte des éléments de bateaux. Le jeu prend fin quand l'un des deux arrive à zéro.

T, U, V: Variables booléennes (indicateurs 0/1):

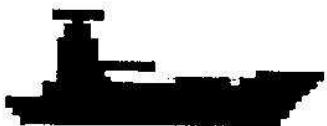
- quand T=0, on tire au hasard, sinon on achève un bateau touché,
- quand T=U=1, on connaît l'orientation du bateau précédemment touché,
- quand T=U=V=1, on bute, on doit faire marche arrière.

G: Nombre de fois où la machine a détecté une tricherie de la part du joueur.

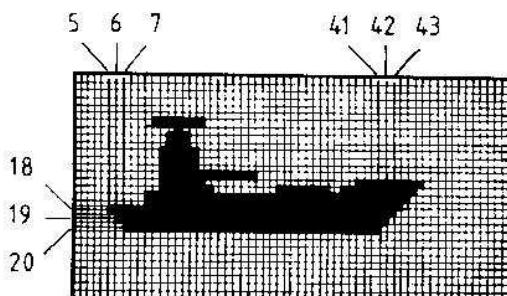
## Sous-programme de présentation

Le sous-programme en ligne 2000 est chargé de dessiner les deux navires de guerre qui présentent le jeu. Nous allons l'expliquer spécialement, car c'est un excellent exercice graphique, qui montre comment il est possible de réaliser des dessins assez élaborés avec un programme simple.

Un modèle de bateau a tout d'abord été dessiné sur du papier millimétré (par une personne ne connaissant pas la programmation, d'ailleurs !). En voici une reproduction :



L'original ainsi réalisé a été codifié par les coordonnées DRAW des extrémités des lignes horizontales qui les composent. Ce sont ces nombres qui sont en DATA.



Il suffit donc de lire les DATA quatre par quatre pour tirer les traits horizontaux qui reproduiront le dessin.

Voyons maintenant comment dessiner deux bateaux.

Si vous regardez attentivement la boucle FOR-NEXT des lignes 2010 à 2030, vous verrez que la ligne 2020 dessine un trait à l'échelle 1,5 et que la ligne 2025 dessine le même trait à l'échelle 1, et de plus "renversé" par l'opération 150-X, et "surélevé" par l'opération Y-5.

Des manipulations simples comme celle-là peuvent permettre des effets variés, dont certains surprenants comme celui-ci : remplacez la ligne 2020 par 2020 DRAW(1.5\*X,30-Y)-(1.5\*Z,30-T) puis faites RUN...

## BATAILLE NAUALE

```
1 REM BATAILLE NAUALE
5 REM JUIN 84
7 GOSUB 2000
10 CLEAR :DIM D$(8,8)*1
15 REM Damier joueur
17 CLS
20 Z=120:GOSUB 1200
30 L=4:GOSUB 1000
35 L=3:GOSUB 1000:GOSUB 1000
40 L=1:FOR J=1 TO 4:GOSUB 1000:NEXT J
45 REM----Damier machine
50 LOCATE 0,0:PRINT "Un";TAB(12),"ins
tant","SUP."
55 Z=72:GOSUB 1200
60 L=4:GOSUB 1300
65 L=3:GOSUB 1300:GOSUB 1300
70 L=1:FOR J=1 TO 4:GOSUB 1300:NEXT J
75 REM---Initialisations
80 T=0:M=14:P=14:G=0
85 BEEP 1
99 REM-----
100 REM---Boucle de Jeu
101 REM-----
105 LOCATE 0,3:PRINT "A vous ";
107 LOCATE 0,0:PRINT TAB(8),TAB(8),TAB
(8)
110 LOCATE 0,0:INPUT Q
115 X=INT(Q/10):Y=Q MOD 10
120 IF (X-1)*(X-8)>0 THEN 100
125 IF (Y-1)*(Y-8)>0 THEN 100
130 A=4*X+70:B=4*Y-2
132 LOCATE 0,1
135 IF D$(X,Y)="" THEN DRAW(A,B):PRINT
"Dans","l eau!";
```

---

```
140 IF D$(X,Y)="X" THEN GOSUB 700
145 IF P=0 THEN 350
149 REM-----
150 REM---Jeu de la machine
151 REM-----
155 LOCATE 0,0:PRINT "A moi    "
157 PRINT TAB(8),TAB(8),TAB(8);
160 IF T=1 THEN 400
170 REM--strategie au hasard
175 X=1+INT(8*RND):Y=1+INT(8*RND)
180 A=4*X+118:B=4*Y-2
185 IF POINT(A,B)=1 THEN 175
200 REM---Tir de la machine
205 LOCATE 0,1:PRINT "Tir:",10*X+Y
210 IF POINT(A,B-1)=1 THEN 222
215 PRINT "A l'eau!":DRAW(A,B)
217 IF T*U=1 THEN U=1
220 GOTO 100
222 REM---Aisuillage
225 DRAWC(A,B+1)-(A+1,B)-(A,B-1)-(A-1,
   B)
227 LOCATE 0,3:PRINT "T ou C ?";
230 LOCATE 0,0:PRINT TAB(7):LOCATE 0,0
   :INPUT C$
235 IF C$<>"C" THEN IF C$<>"T" THEN 23
   0
240 DRAW(A,B)
245 M=M-1:IF M=0 THEN 315
250 IF C$="T" THEN 260
255 REM---Coule
258 T=0:PRINT "Coule !":GOTO 100
260 REM--touche
265 X0=X:Y0=Y
270 U=T:T=1:U=0
275 GOTO 100
315 REM---Machine sage
320 LOCATE 0,0:PRINT TAB(8),"J ai      "
   , "sage ",TAB(8);
```

```

325 GOTO 385
350 REM----Joueur gagne
355 LOCATE 0,0:PRINT "Vous      ", "avez"
      , "gagne...", TAB(8);
370 IF G=0 THEN 385
375 FOR I=1 TO 200:NEXT I
380 LOCATE 0,0:PRINT "Mais      ", "vous
      avez", "triche  ", G; " fois !";
385 IF INKEY$="" THEN 385 ELSE 10
399 REM-----
400 REM----Strategie Particuliere
401 REM--onacheve un bateau touche
402 REM-----
405 IF U=1 THEN 450
407 Q=0
410 REM--recherche orientation
412 Q=Q+1:IF Q>4 THEN 600
415 W=DY:DY=-DX:DX=W:REM rotation
420 X=X0+DX:Y=Y0+DY
425 IF (X-1)*(X-8)>0 THEN 410
430 IF (Y-1)*(Y-8)>0 THEN 410
435 A=4*X+118:B=4*Y-2
440 IF POINT(A,B)=1 THEN 410
445 GOTO 200
450 REM--direction (DX,DY) connue
455 IF U=1 THEN 485
460 X=X0+DX:Y=Y0+DY
465 A=4*X+118:B=4*Y-2
470 IF (X-1)*(X-8)>0 THEN 485
475 IF (Y-1)*(Y-8)>0 THEN 485
477 IF POINT(A,B)=1 THEN 485
480 GOTO 200
485 REM--marche arriere
487 DX=-DX:DY=-DY:X=X0:Y=Y0
490 X=X+DX:Y=Y+DY
495 A=4*X+118:B=4*Y-2
500 IF (X-1)*(X-8)>0 THEN 600
502 IF (Y-1)*(Y-8)>0 THEN 600

```

```
505 IF POINT(A+1,B+1)*POINT(A,B)=1 THEN
    N 490
515 IF POINT(A,B)=1 THEN 600
520 GOTO 200
600 REM---Erreur ou tricherie du Joue
ur!
605 LOCATE 0,0:PRINT "Escroc !","Trich
eur","Vendu !","Truand !";
610 T=0:G=G+1:BEEP 1:BEEP 1:BEEP 1:BEE
P 1
615 LOCATE 0,0:PRINT TAB(8),TAB(8),TAB
(8)
620 GOTO 170
700 REM---Bateau touche ou coule par
joueur
705 DRAW(A+1,B+1)-(A-1,B-1):DRAW(A-1,B
+1)-(A+1,B-1)
710 D$(X,Y)="M":P=P-1
715 FOR I=1 TO 4
725 W=AY:AY=-AX:AX=W
730 X1=X+AX:Y1=Y+AY
735 IF (X1-1)*(X1-8)>0 THEN 755
740 IF (Y1-1)*(Y1-8)>0 THEN 755
745 IF D$(X1,Y1)="M" THEN 770
750 IF D$(X1,Y1)="X" THEN 830
755 NEXT I
760 PRINT "Coule !";GOTO 840
770 REM--Bateau de plus d'une case
772 S=0
775 X1=X1+AX:Y1=Y1+AY
780 IF (X1-1)*(X1-8)>0 THEN 800
785 IF (Y1-1)*(Y1-8)>0 THEN 800
790 IF D$(X1,Y1)="X" THEN 830
795 IF D$(X1,Y1)="M" THEN 775
800 REM Changement de sens
805 IF S=1 THEN 760
810 S=1:AX=-AX:AY=-AY
820 GOTO 775
```

---

```
830 PRINT "Touche !"
840 IF INKEY$="" THEN 840
850 RETURN
999 REM
1000 REM----Placement bateaux joueur
1010 LOCATE 0,0:PRINT L;" case";:IF L>1
    THEN PRINT "s" ELSE PRINT
1015 INPUT "Origine";O
1020 X=INT(O/10):Y=O MOD 10
1022 IF L=1 THEN 1035
1025 LOCATE 0,2:INPUT "Direction";D
1030 IF (D-2)*(D-4)*(D-6)*(D-8)<>0 THEN
    1020
1032 DX=SGN((5-D)*(D-8)*(D-2)):DY=SGN((
    5-D)*(D-4)*(D-6))
1035 FOR I=1 TO L
1040 IF (X-1)*(X-8)>0 THEN 1100
1045 IF (Y-1)*(Y-8)>0 THEN 1100
1050 A=4*X+118:B=4*Y-2
1055 IF POINT(A,B-1)=1 THEN 1100
1060 REM--Trace d'un element de bateau
1065 DRAW(A-1,B-1)-(A-1,B+1)-(A+1,B+1)-
    -(A+1,B-1)-(A-1,B-1)
1070 X=X+DX:Y=Y+DY
1080 NEXT I
1090 LOCATE 0,0:PRINT TAB(14),TAB(14),T
    AB(14)
1095 RETURN
1100 REM----Erreur de l'utilisateur
1105 IF I=1 THEN 1000
1110 FOR J=I-1 TO 1 STEP -1
1115 X=X-DX:Y=Y-DY
1120 A=4*X+118:B=4*Y-2
1125 DRAW(A-1,B-1)-(A-1,B+1)-(A+1,B+1)-
    -(A+1,B-1)-(A-1,B-1)
1130 NEXT J
1135 GOTO 1000
```

```

1200 REM---Damiers 8*8
1205 FOR I=0 TO 8
1210 DRAW(Z+4*I,0)-(Z+4*I,31)
1225 DRAW(Z,4*I)-(Z+32,4*I)
1230 NEXT I
1235 RETURN
1299 REM
1300 REM----Placement bateaux machine
1310 X0=1+INT(RND*8):Y0=1+INT(RND*8)
1315 DX=INT(3*RND)-1
1320 IF DX=0 THEN DY=SGN(.5-RND) ELSE D
   Y=0
1330 FOR I=0 TO L-1
1335 X=X0+I*DX:Y=Y0+I*DY
1340 IF (X-1)*(X-8)>0 THEN 1300
1345 IF (Y-1)*(Y-8)>0 THEN 1300
1350 IF D$(X,Y)="X" THEN 1300
1360 AX=DX:AY=DY
1365 FOR K=1 TO 4
1370 W=AY:AY=-AX:AX=W
1375 X1=X+AX:Y1=Y+AY
1380 IF (X1-1)*(X1-8)>0 THEN 1400
1385 IF (Y1-1)*(Y1-8)>0 THEN 1400
1390 IF D$(X1,Y1)="X" THEN 1300
1400 NEXT K
1410 NEXT I
1500 REM---Bateau OK.
1505 FOR I=0 TO L-1
1510 D$(X0+I*DX,Y0+I*DY)="X"
1515 NEXT I
1520 RETURN
1999 REM-----
2000 REM----Presentation
2001 REM-----
2005 CLS
2010 FOR I=1 TO 18
2015 READ X,Y,Z,T

```

```
2020 DRAW(1.5*X,Y)-(1.5*Z,T)
2025 DRAW(150-X,Y-5)-(150-Z,T-5)
2030 NEXT I
2050 LOCATE 1,3:PRINT "*BATAILLE NAVAL
E*";
2090 IF INKEY$="" THEN 2090 ELSE RETURN

2100 REM COQUE
2105 DATA7,20,41,20,6,19,42,19,4,18,43,
18,10,17,44,17,10,16,45,16
2110 REM SUPERSTRUCTURES
2115 DATA12,15,18,15,28,15,34,15,37,15,
46,15
2120 DATA12,14,17,14,38,14,48,14,12,13,
24,13
2125 DATA12,12,16,12,12,11,16,11,12,10,
16,10,13,9,15,9,13,8,15,8
2130 DATA14,7,14,7,11,6,17,6
```

## *Jeu du pendu*

Il ne s'agit pas d'éviter le lynchage d'un innocent dans l'ouest américain mais de la traditionnelle recherche d'un mot mystérieux (introduit dans la machine par un tiers), jeu si répandu dans nos écoles et lycées.

Le PB 700 autorise les mots de 3 à 32 lettres. Il affiche la première et la dernière lettres. Chaque caractère à découvrir est représenté par un tiret.

Vous proposez une lettre : si elle rentre effectivement dans la composition du mot inconnu, vous la verrez apparaître à sa place dans le mot ; sinon, c'est un pas de plus vers la potence.

Vous avez droit à dix erreurs pour aboutir. Si vous ne trouvez pas, le micro-ordinateur vous fournira la réponse à la fin du jeu.

Et maintenant, à vous de jouer !

---

## Structure du programme

- 1-15 : Présentation.
- 20-26 : Saisie du mot secret.
- 27-32 : Copie du mot saisi dans une structure plus appropriée.
- 34-38 : Initialisations.
- 40-64 : Affichage de la partie connue du mot et interrogation.
- 70-80 : Recherche de la lettre dans le mot secret.
- 90-106 : Tracé d'un trait du gibet.
- 108-110 : Définition du gibet: liste de 11 couples définissant des segments de droite. La tête qui est un rond est définie à part (lignes 94-96).
- 112-122 : Le joueur a perdu... ou gagné.

## Variables utilisées

- I\$ : Limite à 60 le nombre de lettres du mot secret.
- M\$ : Est une copie lettre à lettre de I\$.
- T\$ : Est ce qui est affiché au joueur de M\$.
- A, B : Sont des booléens:
  - A=1 signifie: la lettre proposée est dans le mot.
  - B=1 signifie: T\$<>M\$ (soit: le mot n'est pas entièrement trouvé).

### P E N D U

```
1 REM
15 CLEAR :CLS :LOCATE 0,1:PRINT "|||||
JEU DU PENDU>>>""
20 REM ENTREE DU MOT
22 DIM I$(1)*60
24 INPUT "MOT : ",I$(1)
26 L=LEN(I$(1)):IF L<3 THEN 24
27 CLS
```

---

```
28 DIM M$(L)*1,T$(L)*1
30 FOR I=1 TO L:M$(I)=MID$(I$(1),I,1)
    :NEXT I
32 ERASE I$
34 FOR I=1 TO L-1:T$(I)=CHR$(95):NEXT
    I
36 T$(1)=M$(1):T$(L)=M$(L)
38 B=1:RESTORE
40 REM AFFICHAGE
42 LOCATE 0,2
44 FOR I=1 TO L
46 IF I=16 THEN PRINT
48 PRINT T$(I);
50 NEXT I
55 IF B=0 THEN 200
58 LOCATE 5,0:PRINT "LETTRE :"
60 REM ENTREE DU JOUEUR
62 LOCATE 14,0:PRINT "":LOCATE 14,0
    :INPUT "",C$
64 IF LEN(C$)<>1 THEN 62
70 REM RECHERCHE LETTRE DANS MOT
72 A=0:B=0
74 FOR I=2 TO L-1
76 IF M$(I)=C$ THEN T$(I)=C$:A=i
78 IF T$(I)<>M$(I) THEN B=1
80 NEXT I
84 IF A=1 THEN 40
90 REM CONSTRUCTION DU GIBET...
92 N=N+1:BEEP :IF N<>2 THEN 100
94 REM TETE
96 DRAW(149,11)-(149,12)-(150,13)-(15
    1,12)-(151,11)
98 GOTO 60
100 REM TRAIT AUTRE QUE LA TETE
102 READ X,Y,Z,T
104 DRAW(X,Y)-(Z,T)
106 IF N<>11 THEN 60
```

---

```
108 DATA130,31,159,31,130,31,130,1,130
    ,21,140,31
109 DATA130,1,150,1,130,10,139,1,150,1
    ,150,10,145,14,155,14
110 DATA150,14,150,18,150,18,145,23,15
    0,18,155,23
112 REM PENDAISON !
114 LOCATE 0,2
116 FOR I=1 TO L
118 IF I=16 THEN PRINT
120 PRINT M$(I);:BEEP
122 NEXT I
200 REM FIN DU JEU
202 LOCATE 4,0
204 IF B=0 THEN PRINT "***BRAVO***":BE
    EP 1 ELSE PRINT " AARGH.. "
215 IF INKEY$="" THEN 215 ELSE 1
```

## Othello

Ce jeu est devenu un grand classique sur micro-ordinateur. C'est pourquoi nous n'avons pas résisté à la tentation de vous le proposer.

De plus, ce programme étant assez long, il présente quelques particularités que nous développerons par la suite.

Avant de vous expliquer le fonctionnement nous allons d'abord vous présenter la règle du jeu.

### Règle du jeu

Othello se joue à deux personnes sur un échiquier  $8 \times 8$ .

Chaque personne joue à son tour en posant un pion sur l'échiquier.

Le but du jeu est de posséder plus de pions que l'adversaire en fin de partie.



*Point principal:* Tout pion encadré par deux pions adverses passe à l'ennemi en changeant de couleur.

Ce point va dicter votre manière de jouer durant toute la partie.

*Il découle de cette règle des obligations importantes*

1. On ne peut jouer une case que si on prend un ou plusieurs pions adverses.

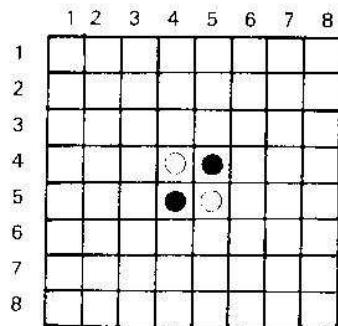
Il est donc impossible de jouer sur une case loin de tout autre pion.

2. La position de départ est imposée: c'est la suivante et les blancs commencent. (voir schéma page ci-après)

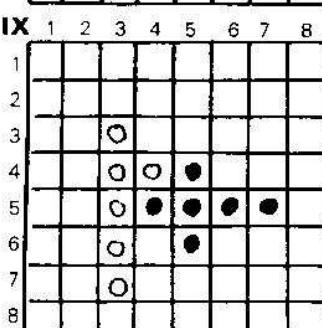
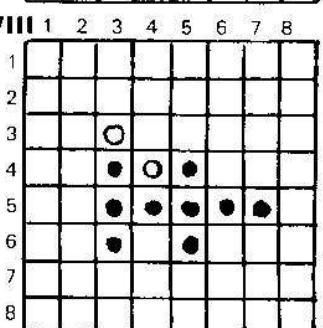
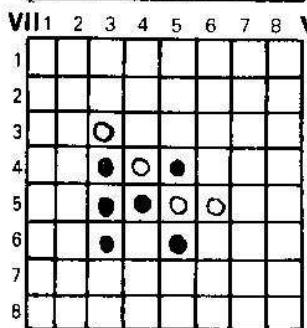
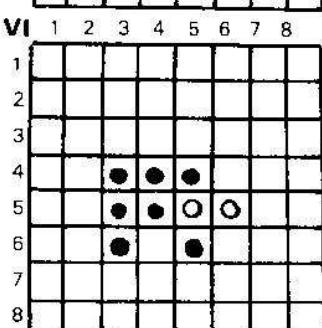
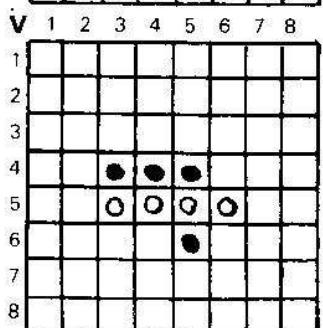
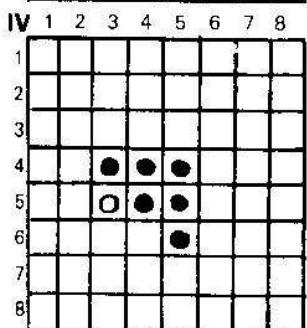
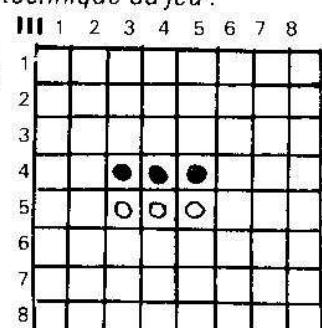
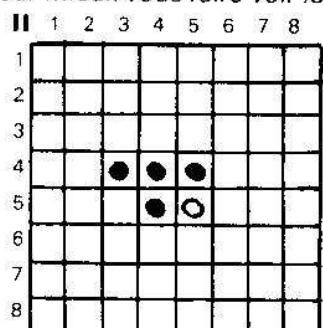
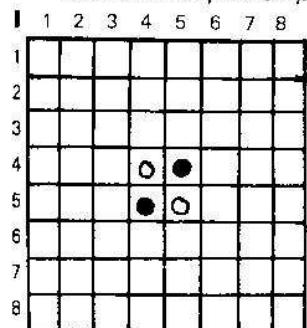
Suivant l'obligation n° 1, les blancs ne peuvent jouer qu'aux cases suivantes:

4.3, 3.4, 6.5, 5.6

*position  
de départ*



*Voici une séquence pour mieux vous faire voir la technique du jeu :*



## **Commentaires sur la séquence de jeu**

- I. Position de départ.
- II. Les noirs jouent et encadrent le blanc placé en 4.4 par deux noirs placés en 5.4 et 3.4 (récemment joué)  $\Leftrightarrow$  le blanc devient noir.
- III. Les blancs jouent en 3.5 et encadrent le noir en 4.5  $\Leftrightarrow$  le noir devient blanc.
- IV. Les noirs jouent en 5.6 et encadrent du même coup :
  - a) le blanc placé en 5.5,
  - b) le blanc placé en 4.5Les noirs gagnent deux pions.
- V. Les blancs jouent 6.5 et prennent deux noirs (4.5 et 5.5).
- VI. Voir IV.
- VII. Les blancs jouent en 3.3 et encadrent le noir (4.4).
- VIII. Les noirs jouent 7.5 et encadrent deux blancs.
- IX. Les blancs jouent en 3.7 et encadrent trois noirs.

Après le neuvième coup les noirs ont six pions et les blancs ont six pions. Il y a égalité.

La partie est terminée quand l'un des trois événements ci-dessous se produit :

1. Toutes les cases sont remplies.
2. Ni les blancs ni les noirs ne peuvent prendre des pions.
3. L'un des adversaires n'a plus de pions.

Vous verrez très vite que certaines cases de l'échiquier sont stratégiquement fondamentales. Nous ne vous les avons pas indiquées. Il est bien plus divertissant de les découvrir par vous-même.

## **Présentation du programme**

Celui-ci comprend quatre grandes parties que nous présenterons de manière indépendante :

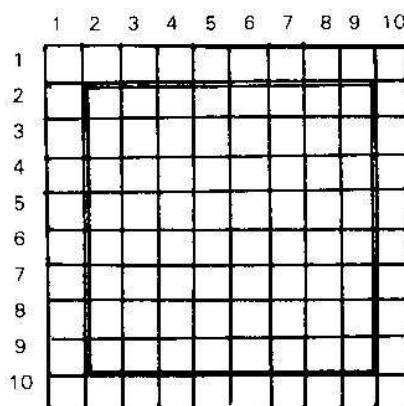
1. Stratégie de l'ordinateur.

2. Stratégie du joueur.
3. Pose d'un pion sur l'échiquier.
4. Calcul des résultats.

## Préliminaires

L'échiquier est représenté en mémoire par un tableau nommé F\$, il est déclaré à la ligne 20 par DIM F\$(10,10)\*1.

Ce tableau est utilisé comme suit:



La partie centrale sert à "poser" les pions.

La partie périphérique sert de "tampon" pour les pions posés sur le bord de l'échiquier.

Le signe \* 1 placé après DIM F\$(10,10) signifie que les cases du tableau ne contiendront qu'un caractère.

La mémoire A\$ contient votre pion.

La mémoire D\$ contient le pion de l'ordinateur.

### 1. 1 à 270: Initialisation du jeu

Cette partie du programme met toutes les mémoires dans l'état initial. A la ligne 220, le programme vous demande si vous voulez commencer et en

---

fonction de votre choix, il attribue le pion blanc ou le pion noir à l'ordinateur.

#### *2. 290 à 550: Stratégie ordinateur*

Celui-ci joue de la manière suivante : Il examine chaque case jouable et lui attribue une note en fonction de sa position stratégique sur l'échiquier.

Ensuite, il évalue le nombre de pions adverses qu'il prend en jouant cette case. Il additionne ce nombre à la note stratégique ci-dessus.

La case ayant la note globale la plus élevée est alors jouée.

Dans la partie préliminaire, les lignes 60 → 210 servent à remplir un tableau (nommé G(8,8)) avec les notes des cases de l'échiquier.

*Particularités :*

- Un compteur est mis en route à la ligne 342, il permet de contrôler le nombre de cases que l'ordinateur a déjà testé.
- Lorsque l'ordinateur a sélectionné une case, il "regarde" dans les huit directions autour de lui pour détecter le nombre total de pions adverses vulnérables. Ces huit directions sont répertoriées dans un tableau B(8,2).
- A la ligne 525, l'ordinateur examine le cas où aucune case ne prend de pions à l'adversaire. Si cela arrive, il ne peut pas jouer.

#### *3. 590 à 860: Stratégie joueur*

La stratégie joueur est bien plus simple. Cette partie sert simplement à contrôler les erreurs que vous pouvez faire en tapant votre jeu au clavier.

La ligne 700 sert de message d'erreur en ce cas.

#### *4. 3000 à 3180: Pose d'un pion sur l'échiquier*

Ceci est un sous-programme qui "dessine" sur votre écran la position courante du jeu.

En ligne 3000, il identifie l'appel du programme principal (joueur ou ordinateur). Il "sait" donc quel genre de pion il devra tracer.

**3010-3090:** Il trace des pions blancs.

**3100-3180:** Il trace des pions noirs.

### **900 à 1000: Calcul des résultats**

Ici, la machine compte sur le tableau F\$ le nombre de pions blancs et le nombre de pions noirs. En fonction de leur appartenance à l'ordinateur ou à vous, elle décrète la victoire.

### **Particularités**

**2000-2200:** Ceci est un sous-programme traçant l'échiquier sur l'écran et affichant la position de départ vue précédemment.

**9000-9010:** Temporisation utilisée fréquemment.

## **Comment jouer avec votre ordinateur.**

C'est quand même la partie la plus intéressante. Voici le mode d'emploi.

1. A la question (ligne 240) "que choisissez-vous?" vous devrez répondre:

"." pour un pion blanc et  
"0" pour un pion noir.

Si vous avez choisi . vous commencez.

2. Quand vous devrez jouer, l'ordinateur vous posera la question suivante : "Votre jeu?". Vous devrez alors entrer votre choix de la manière suivante:

numéro de colonne . numéro de ligne  
(de 1 à 8) (de 1 à 8)

**Exemple:** Votre jeu : 3.4



Vous savez maintenant presque tout sur Othello. A vous de vous mesurer à l'ordinateur. Il est possible de le battre en jouant très finement. Mais n'oubliez pas que celui-ci ne pardonne aucune erreur.

Amusez-vous bien !

## **Recensement des variables utilisées**

F\$ : Tableau représentant l'échiquier.

G : Tableau contenant les notes.

B : Tableau contenant les directions.  
 I, J, V, R : Compteurs de boucles.  
 E 4 : Indicateur d'impossibilité de jouer.  
 A\$ : Pion du joueur.  
 D\$ : Pion de l'ordinateur.  
 CT : Compteur de cases testées par l'ordinateur.  
 S : Compteur de coups joués.  
 N1 : Note statique extraite du tableau G.  
 PC : Nombre de pions maximum pris à une case.  
 Z, Y : Nombre de pions courant pris à une case.  
 N 2 : Tampon contenant PC en cas de modification.  
 L,M : Coordonnées d'une case proche.  
 P! : Note générale.  
 X : Indicateur d'appel du sous-programme de tracé.  
 C : Jeu du joueur.  
 I,J : Dans la partie "stratégie joueur" contient l'abscisse et l'ordonnée.  
 I1,J1 : Mémoires contenant l'abscisse et l'ordonnée pour le sous-programme.  
 GO : Contient le nombre de pions de l'ordinateur en fin de partie.  
 GJ : Contient le nombre de pions du joueur en fin de partie.  
 OS : Contient la réponse à "Rejouez-vous ?".  
 Q,W : Coordonnées de tracé d'un pion correspondant au tableau F\$(I,J).

### O T H E L L O

```

1 CLS :BEEP
2 LOCATE 0,1:PRINT "***** OTHELLO **"
****"
6 DRAW(42,5)-(110,5)-(110,17)-(42,17)
)-(42,5)
8 REM
9 REM      INITIALISATION
10 CLEAR

```

```

20 DIM F$(10,10)*1
30 DIM G(8,8)
40 DIM B(8,2),P(8),PI(8)
42 FOR I=2 TO 3
43 FOR J=2 TO 9
44 F$(I,J)="+":F$(I+6,J)="+"
45 NEXT J
46 NEXT I
47 FOR I=4 TO 7
48 F$(I,2)="+":F$(I,8)="+":F$(I,3)="+
":F$(I,9)="+"
49 NEXT I
50 F$(5,5)="0":F$(5,6)=".":F$(6,6)="0
":F$(6,5)=".":BEEP
55 E4=0
60 FOR I=1 TO 8
70 FOR J=1 TO 8
80 READ G(I,J)
90 NEXT J
100 NEXT I
120 FOR I=1 TO 8
130 FOR J=1 TO 2
140 READ B(I,J)
150 NEXT J
160 NEXT I
170 DATA10,1,4,3,3,4,1,10,1,-1,1,1,1,1
,-1,1
180 DATA4,1,2,3,3,2,1,4,3,1,3,0,0,3,1,
3,3,1,3,0,0,3,1,3,4,1,2,3,3,2,1,4
200 DATA1,-1,1,1,1,1,-1,1,10,1,4,3,3,4
,1,10
210 DATA-1,-1,-1,0,-1,1,0,-1,0,1,1,-1,
1,0,1,1
219 CLS :BEEP
220 PRINT "Pions (. ) contre Pions (
0)."
230 PRINT "Les (. ) commencent.":GOSUB
9000:GOSUB 9000:GOSUB 9000

```

```
240 INPUT "Que choisissez-vous? (.)?"  
      (0)?",A$  
250 IF A$=".," THEN D$="G" ELSE D$=".,"  
260 CLS :GOSUB 2000  
270 IF A$=".," THEN 610  
280 REN  
281 REN  
282 REN  
290 REM STRATEGIE CRDINATEUR  
300 LOCATE 0,0:PRINT "Patience ", "Je  
cherche ":CT=64  
310 IF S=60 THEN 900  
320 N1=0:PC=0:S=S+1:Z=0:N2=0  
322 FOR R=1 TO 8  
324 P!(R)=0  
325 NEXT R  
330 FOR I=2 TO 9  
340 FOR J=2 TO 9  
342 LOCATE 10,0:PRINT USING"##";CT  
345 CT=CT-1  
350 IF F$(I,J)><"" THEN 510  
360 FOR K=1 TO 8  
370 L=I+B(K,1):M=J+B(K,2)  
380 IF F$(L,M)<>A$ THEN 450  
385 Y=0  
390 PC=PC+1  
400 Y=Y+1  
410 I=I+B(K,1):M=M+B(K,2)  
420 IF F$(L,M)=A$ THEN 390  
430 IF F$(L,M)="" THEN PC=PC-Y:GOTO 45  
0  
440 P(K)=Y  
450 NEXT K  
455 IF PC=0 THEN 510  
460 N1=G(I-1,J-1)  
470 IF N1+PC<=Z THEN 480  
471 Z=N1+PC:I1=I:J1=J:N2=PC  
472 FOR R=1 TO 8
```

```
423 P!(R)=P(R)
424 NEXT R
480 PC=0
485 FOR R=1 TO 8
486 P(R)=0
487 NEXT R
510 NEXT J
520 NEXT I
525 IF N2<0 THEN 540
530 LOCATE 0,0:PRINT "Je ne veux","Pas
jouer ":"E4=1:S=S-1:GOSUB 3000:GOTO 590
540 LOCATE 0,0:PRINT "Je Joue en   ",
I1-1;";J1-1;"  ":"GOSUB 9000
550 X=10:GOSUB 3000
586 REM
587 REM
588 REM
589 REM STRATEGIE JOUEUR
590 IF S=61 THEN 900
610 PC=0:S=S+1:LOCATE 0,0:PRINT "Coup
n. ";S;" "
612 FOR R=1 TO 8
613 P!(R)=0
614 NEXT R
620 LOCATE 0,1:INPUT "Votre Jeu:",C
622 IF C<>0 THEN 630
624 IF E4=1 THEN LOCATE 0,2:PRINT "Jeu
bloque":GOSUB 9000:GOTO 900
626 S=S-1:GOTO 300
630 I=1+INTC:J=1+FRACC*10
632 IF I<2 THEN 700
634 IF J<2 THEN 700
636 IF I>9 THEN 700
638 IF J>9 THEN 700
640 IF F$(I,J)><"" THEN 700
690 GOTO ?10
700 LOCATE 0,0:PRINT "Impossible":GOSU
B 9000:LOCATE 10,1:PRINT "  ":GOTO 620
```

```
710 FOR K=1 TO 8
720 L=I+B(K,1):M=J+B(K,2)
740 IF F$(L,M)<>D$ THEN 790
750 P!(K)=P!(K)+1:PC=PC+1
760 IF F$(L+B(K,1),M+B(K,2))="" THEN P
C=PC-P!(K):P!(K)=0:GOTO 790
765 IF F$(L+B(K,1),M+B(K,2))=A$ THEN 7
90
770 L=L+B(K,1):M=M+B(K,2):GOTO 750
790 NEXT K
800 IF PC=0 THEN 700
810 I1=I:J1=J
820 X=0:GOSUB 3000
860 GOTO 300
865 REM
866 REM
867 REM
868 REM RESULTATS
900 GO=0:GJ=0:LOCATE 0,0:PRINT "RESULT
AT: "
910 FOR I=2 TO 9
920 FOR J=2 TO 9
925 IF F$(I,J)=A$ THEN GJ=GJ+1
930 IF F$(I,J)=D$ THEN GO=GO+1
940 NEXT J
950 NEXT I
960 IF GJ>GO THEN PRINT "VOUS GAGNEZ!
",GJ;" A";GO;" D":GOSUB 9000:GOTO 1000
980 IF GO>GJ THEN PRINT "VOUS PERDEZ!
",GJ;" A";GO;" D":GOSUB 9000:GOTO 1000
990 PRINT "PARTIE NULLE","EGALITE A";G
J:GOSUB 9000:GOSUB 3000
1000 IF INKEY$="" THEN 1000
1005 CLS :INPUT "VOULEZ VOUS REJOUER?",O$
1010 IF O$><"N" THEN RESTORE 170:GOTO 1
1020 END
```

```
1996 REM
1997 REM
1998 REM
1999 REM AFFICHAGE INITIA!
2000 CLS
2010 FOR I=0 TO 31 STEP 4
2020 DRAW(118,I)-(150,I)
2030 NEXT I
2040 FOR I=118 TO 150 STEP 4
2050 DRAW(I,0)-(I,31)
2060 NEXT I
2130 FOR I=5 TO 6
2140 FOR J=5 TO 6
2150 IF F$(I,J)><". " THEN 2180
2160 Q=4*I+112:W=4*J-6
2170 DRAW(Q,W)
2180 IF F$(I,J)><"0" THEN 2210
2190 Q=4*I+111:W=4*J-5
2200 DRAW(Q,W)-(Q+2,W)-(Q+2,W-2)-(Q,W-2)
)-(Q,W)
2210 NEXT J
2220 NFXT I
2230 RETURN
2996 REM
2997 REM
2998 REM
2999 REM CORRECTION
3000 IF X=10 THEN 3005
3002 IF A$="." THEN 3100 ELSE 3010
3005 IF D$="." THEN 3100
3010 Q=4*I1+111:W=4*J1-5
3015 F$(I1,J1)="0"
3020 DRAW(Q,W)-(Q+2,W)-(Q+2,W-2)-(Q,W-2)
)-(Q,W):BEEP
3025 FOR R=1 TO 8
3026 IF P!(R)><0 THEN 3030
3027 IF F$(I1+B(R,1),J1+B(R,2))><"+" TH
EN 3085
```

```

3028 F$(I1+B(R,1),J1+B(R,2))=""
3029 GOTO 3085
3030 FOR T=1 TO P!(R)
3040 I1=I1+B(R,1):J1=J1+B(R,2)
3045 F$(I1,J1)="0"
3050 Q=4*I1+112:W=4*J1-6
3060 DRAWC(Q,W)
3070 DRAW(Q-1,W+1)-(Q+1,W+1)-(Q+1,W-1)-
(Q-1,W-1)-(Q-1,W+1):BEEP
3080 NEXT T
3082 I1=I1-(P!(R)*B(R,1)):J1=J1-(P!(R)*
B(R,2))
3085 NEXT R
3090 GOTO 3180
3100 Q=4*I1+112:W=4*J1-6
3105 F$(I1,J1)=". "
3110 DRAW(Q,W):BEEP 1
3115 FOR R=1 TO 8
3116 IF P!(R)<>0 THEN 3120
3117 IF F$(I1+B(R,1),J1+B(R,2))<> "+" TH
FN 3175
3118 F$(I1+B(R,1),J1+B(R,2))=""
3119 GOTO 3175
3120 FOR T=1 TO P!(R)
3130 I1=I1+B(R,1):J1=J1+B(R,2)
3135 F$(I1,J1)=". "
3140 Q=4*I1+111:W=4*J1-5
3150 DRAWC(Q,W)-(Q+2,W)-(Q+2,W-2)-(Q,W-
2)-(Q,W)
3160 DRAW(Q+1,W-1):BEEP 1
3170 NEXT T
3172 I1=I1-(P!(R)*B(R,1)):J1=J1-(P!(R)*
B(R,2))
3175 NEXT R
3180 RETURN
9000 FOR U=1 TO 100:NEXT U
9010 RETURN

```

## VARIABLES DECLAREES UTILISEES

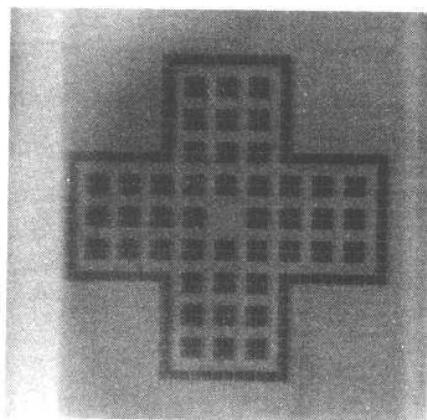
---

```
F$( ) G( ) B( ) P( )
P!( ) E4    PC   I1
J1     CT    N1   N2
```

### **Solitaire**



Un livre de jeux sur micro-ordinateur ne pouvait pas être complet sans un jeu de solitaire que vous connaissez sûrement pour être assez difficile à réussir. Nous n'avons donc pas résisté à la tentation de vous le présenter.



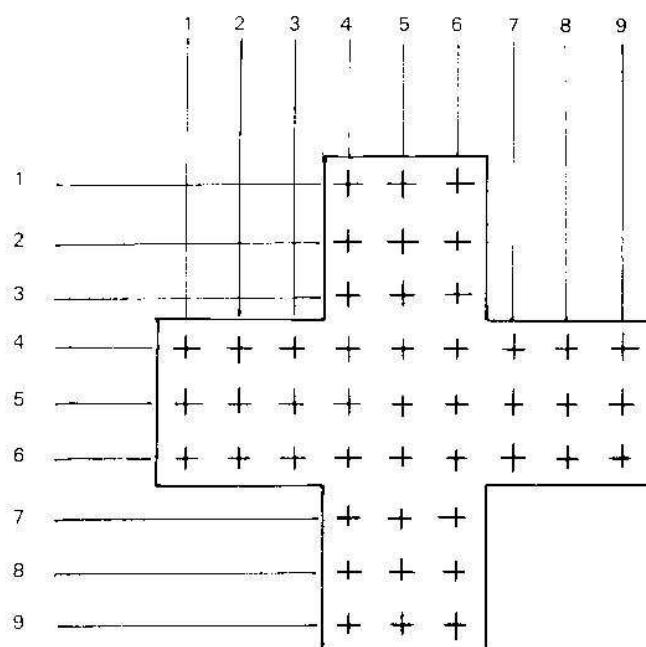
## Présentons d'abord la règle du jeu

Le Solitaire se joue sur un damier en forme de croix. Il y a 44 pions et une case vide.

Le but du jeu est de ne laisser qu'un seul pion sur le damier en fin de partie.

Pour cela, on supprime les pions en les "mangeant" comme aux Dames.  
La case vide permet d'entamer la partie.

Le programme vous demande tout d'abord quel pion vous voulez enlever pour créer une case vide. Pour désigner un pion sur le damier, il faut vous servir des coordonnées rectangulaires ci-après :



Ainsi la case centrale a pour coordonnées :

5 en horizontal  
5 en vertical

A la question "1<sup>er</sup> pion ?" vous répondrez donc 55 pour enlever le pion central.

Puis quand vous commencerez à jouer le programme vous demandera :

### Déplacement

vous indiquerez alors le lieu origine du pion preneur et la case vide vers laquelle il se dirige en prenant un pion.

Ainsi, si comme l'indique la flèche sur le dessin ci-dessus, vous désirez manger le pion situé en 54 par le pion situé en 53, vous écrirez :

53-55

et le programme vous fera jouer.

Soyez persévérant, c'est très difficile !!

### Description du programme

- 10-50 : Initialisation.
- 60-190 : Dessin du damier.
- 200-260 : Création de la case vide.
- 270-410 : Boucle de jeu.
- 500-540 : Résultat en cas d'insuccès.
- 900-950 : Correction en cas d'erreur.
- 1000-1010: Sous-programme d'effacement d'un pion.
- 1500-1520: Résultat en cas de succès.

### S O L I T A I R E

```
10 CLS :CLEAR :T=44
20 LOCATE 0,1:PRINT "**** SOLITAIRE *
****";
```

```

30 DRAW(0,5)-(159,5)
40 DRAW(0,17)-(159,17)
50 FOR I=0 TO 120:NEXT I
60 CLS
70 DRAW(120,10)-(129,10)-(129,1)-(140
,1)-(140,10)-(149,10)-(149,21)
80 DRAW(149,21)-(140,21)-(140,30)-(12
9,30)-(129,21)-(120,21)-(120,10)
90 FOR I=122 TO 146 STEP 3
100 FOR J=12 TO 18 STEP 3
110 DRAW(I,J)-(I+1,J)-(I+1,J+1)-(I,J+1
)
120 NEXT J
130 NEXT I
140 FOR I=131 TO 137 STEP 3
150 FOR J=3 TO 9 STEP 3
160 DRAW(I,J)-(I+1,J)-(I+1,J+1)-(I,J+1
)
170 DRAW(I,J+18)-(I+1,J+18)-(I+1,J+19)
-(I,J+19)
180 NEXT J
190 NEXT I
200 LOCATE 0,0:PRINT "1er Pion:";:INPU
T X
210 I=INT(X/10):J=FRAC(X/10)*10
220 Q=3*I+119:W=3*J
230 IF POINT(Q,W)><0 THEN 250
240 LOCATE 10,0:PRINT " ";:GOTO 200
250 GOSUB 1000
260 Z=1
270 LOCATE 0,0:PRINT "Coup n.";Z;" "
280 LOCATE 0,1:PRINT "Deplacement:",""
":LOCATE 0,2:INPUT D$
285 IF D$="" THEN 500
290 X=VAL(MID$(D$,1,2)):Y=VAL(MID$(D$,
4,2))
300 ID=INT(X/10):JD=FRAC(X/10)*10:IA=I
NT(Y/10):JA=FRAC(Y/10)*10

```

```

310 Q=3*ID+119:W=3*JD
320 IF POINT(Q,W)=0 THEN GOSUB 900
330 Q=3*IA+119:W=3*JA
340 IF POINT(Q,W)=1 THEN GOSUB 900
350 IM=(IA+ID)/2:JM=(JA+JD)/2
360 Q=3*IM+119:W=3*JM
370 IF POINT(Q,W)=0 THEN GOSUB 900
380 T=T-1:Z=Z+1
390 GOSUB 1000
391 Q=3*ID+119:W=3*JD
392 GOSUB 1000
395 Q=3*IA+119:W=3*JA
396 DRAW(Q,W)-(Q+1,W)-(Q+1,W+1)-(Q,W+1)
>
400 IF T=1 THEN 1500
410 GOTO 270
500 LOCATE 0,0:PRINT TAB(10),TAB(10),T
AB(10);
520 LOCATE 0,0:PRINT "Perdu!!," Il re
ste ",T;" Pions":BEEP :BEEP
530 LOCATE 0,3:INPUT "ReJouez-vous":0$
540 IF 0$="N" THEN END ELSE 10
900 REM Message d erreur
910 BEEP :LOCATE 0,3:PRINT "IMPOSSIBLE"
";
920 LOCATE 0,2:PRINT "
930 FOR B=1 TO 50:NEXT B
940 LOCATE 0,3:PRINT "
950 GOTO 280
1000 DRAWC(Q,W)-(Q+1,W)-(Q+1,W+1)-(Q,W+
1)
1010 RETURN
1500 CLS :LOCATE 0,0:PRINT "VOUS AVEZ R
EUSSI!!":BEEP :BEEP
1510 LOCATE 0,1:INPUT "REJOUEZ-VOUS":0$
1520 IF 0$="N" THEN END ELSE 10

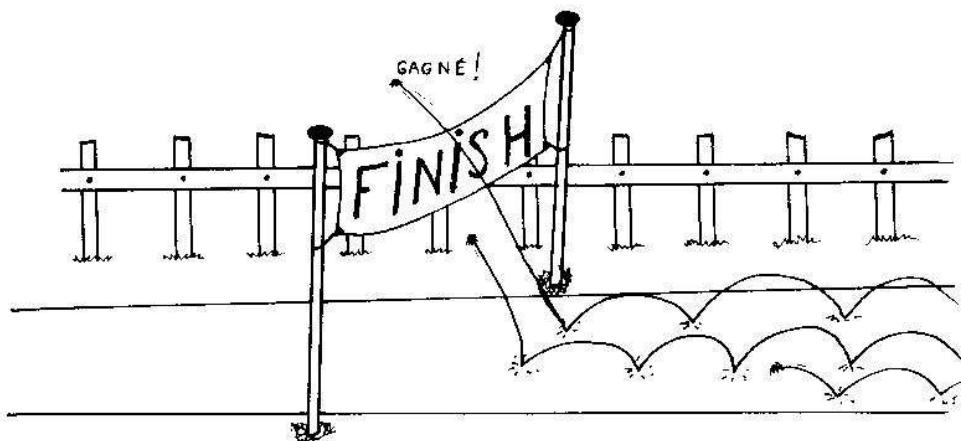
```

2

## Jeux de hasard



## **Course de puces**



Ceci s'adresse aux adeptes des jeux de pur hasard. Nous vous proposons une variante du tiercé, la course de puces, adaptée aux dimensions de l'écran.

Le principe en est simple : Vous disposez d'un capital de départ de 50 F. Vous en jouez une partie sur une des huit puces. Vous donnez le départ et la course est lancée.

Si vous gagnez, votre mise est multipliée par sept, sinon vous perdez votre mise.

Choisissez la bonne puce ou vous connaîtrez les pires angoisses.

### **Structure du programme**

- 1-20 : Présentation hilarante.
- 25-45 : Pari.
- 50-99 : Prêts au départ ? Partez !
- 100-105 : Les puces sortent de leurs Box.
- 110 : Origine de la boucle de course.
- 110-130 : Boucle : une puce au hasard est avancée.

140-170 : Une puce a atteint l'arrivée ! Résultat du pari...

500 Sous-programme affichant le champ de course.

## Variables utilisées

P!(8) : Abscisses des huit puces.  
T : Total de la richesse du joueur.  
M : Mise.  
N : Puce choisie.  
K : Numéro de la puce qui avance.  
P, Q : Copie de ses coordonnées DRAW.  
D\$ et A\$ : Sortes de DATA pour écrire verticalement (départ et arrivée de la course).

```
C O U R S E   D E   P U C E S
1 REM
10 CLS :CLEAR :T=50:DIM P!(8)
14 DRAW(0,6)-(151,6)
15 LOCATE 0,1:PRINT "...PUCES DE COURS
E.."
16 DRAW(0,16)-(151,16)
20 FOR I=1 TO 100:DRAW(RND*160,RND*32
):NEXT I
25 REM JEU
27 CLS :IF T>0 THEN 30
28 PRINT "...VOUS ETES RUINE!":END
30 PRINT "Vous avez";T;" F"
35 INPUT "Quelle est votre mise ";
M:M=INTM
37 IF M*(M-T)>0 THEN 30
40 INPUT "Sur quelle puce ";N
45 IF (N-1)*(N-8)>0 THEN 40
50 GOSUB 500:REM (TERRAIN)
60 LOCATE 0,0:PRINT "Mise:",M,"Puce";
N
```

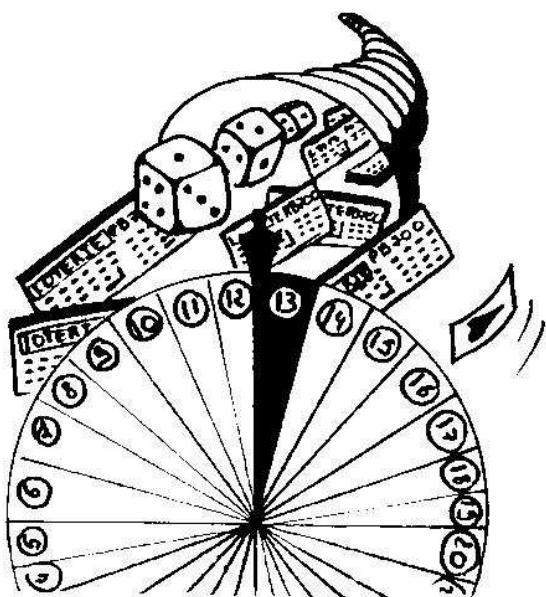
```

70 LOCATE 0,3:PRINT TAB(5);
75 IF INKEY$<>"" THEN 100
80 LOCATE 0,3:PRINT "PRET?":GOTO 70
99 REM
100 REM POSITION DE DEPART
105 FOR I=1 TO 8:P!(I)=57:NEXT I
109 REM -----
110 REM COURSE
111 REM -----
115 K=1+INT(RND*8)
120 Q=P!(K)+6:P=P!(K):Y=4*K-3
125 DRAW(P,Y)-(Q,Y):DRAWC(P,Y)-(Q-1,Y)
127 P!(K)=Q
130 IF Q<151 THEN 110
139 REM -----
140 REM FIN DE LA COURSE
141 REM -----
145 LOCATE 0,0:PRINT "La    ","Puce  ","
No. ";K,"sagne";
150 IF INKEY$="" THEN 150
160 IF K=N THEN T=T+2*M ELSE T=T-M
170 GOTO 25
499 REM -----
500 REM AFFICHAGE DU TERRAIN
501 REM -----
510 CLS
515 D$="DEP":A$="ARR"
520 FOR I=0 TO 2
530 LOCATE 6,I:PRINT MID$(D$,I+1,1)
535 LOCATE 19,I:PRINT MID$(A$,I+1,1):
540 NEXT I
545 REM Points
550 LOCATE 6,3:PRINT ".";
555 DRAW(155,30)-(156,30)-(156,29)-(15
5,29)
560 REM Traits verticux
565 DRAW(48,0)-(48,31):DRAW(56,0)-(56,
31)

```

```
570 DRAW(152,0)-(152,31);DRAW(159,0)-(159,31)
575 REM Traits horizontaux
580 FOR I=3 TO 31 STEP 4
585 DRAW(56,I)-(152,I)
590 NEXT I
599 RETURN
```

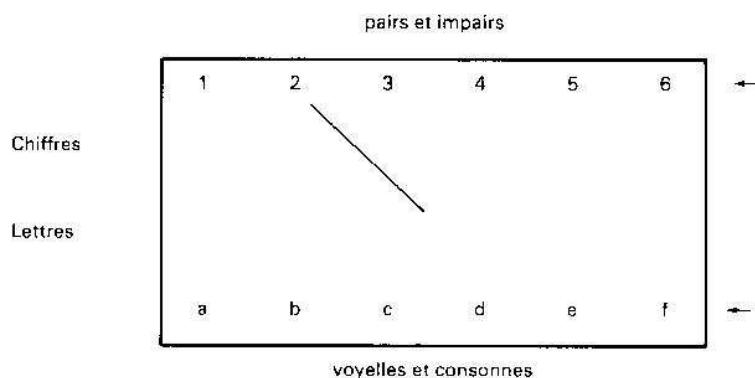
### Loterie



Vous voici maintenant avec une somme de 50 F entre les mains devant un tapis vert d'un genre un peu particulier où vous allez pouvoir faire fortune.

En effet, ce jeu est une version modifiée de la roulette de casino et de la loterie de fête foraine.

Le tapis vert est constitué de la manière suivante:



Vous devez miser au début de chaque manche de la manière suivante:

Sur la première ligne de votre écran, vous verrez s'afficher votre Avoir (= à 50 F en début de partie). Puis vous verrez s'afficher:

- l? vous entrerez alors votre mise sur les LETTRES.
- c? vous entrerez alors votre mise sur les CHIFFRES.
- k? vous entrerez alors votre mise sur les CONSONNES.
- v? vous entrerez alors votre mise sur les VOYELLES.
- p? vous entrerez alors votre mise sur les PAIRS.
- i? vous entrerez alors votre mise sur les IMPAIRS.

Si vous ne voulez rien miser dans l'une de ces rubriques, affichez Ø à la question.

Une fois votre mise effectuée (sans dépasser votre avoir bien entendu auquel cas vous vous ferez violentement invectiver par l'honnête PB 700 croupier !) l'aiguille commencera à tourner pour s'arrêter sur l'une des douze figures.

Vos gains seront alors calculés comme suit:

	MISES					
	i	k	v	c	p	i
1	Ø	Ø	Ø	2	Ø	4
2	Ø	Ø	Ø	2	4	Ø
3	Ø	Ø	Ø	2	Ø	4
F	4	Ø	Ø	Ø	2	4
I	5	Ø	Ø	Ø	2	Ø
G	5	Ø	Ø	Ø	2	Ø
U	6	Ø	Ø	Ø	2	4
R	a	2	Ø	10	Ø	Ø
E	b	2	5	Ø	Ø	Ø
S	c	2	5	Ø	Ø	Ø
	d	2	5	Ø	Ø	Ø
	e	2	Ø	10	Ø	Ø
	f	2	5	Ø	Ø	Ø

On indique dans les cases, le multiplicateur de la mise.

**Exemple :**

$$\begin{array}{l} \text{Si vous avez misé :} \\ \quad l = 10 \qquad \qquad k = 15 \qquad \qquad v = \emptyset \\ \quad c = 5 \qquad \qquad p = 10 \qquad \qquad i = \emptyset \end{array}$$

et que l'aiguille s'arrête sur 4 .

$$\begin{array}{lll} \text{vous aurez: } & l = \emptyset & c = c \times 2 \Leftrightarrow c = 10 \\ & k = \emptyset & p = p \times 4 \qquad p = 40 \\ & v = \emptyset & i = \emptyset \\ & & \Leftrightarrow \text{RAPPORT} = 50 \end{array}$$

## Description du programme

10-80 : Initialisation des variables.  
90-110 : Affichage de la roulette.  
120-210 : Demande des mises.  
250-305 : La roue tourne !  
310-520 : Calcul des gains.  
530-560 : Affichage des résultats.  
600-fin : Sous-programmes de tests de la mise.

## Variables utilisées

Q, W: Point fixe de l'aiguille.  
A !(12) et B !(12): Tableaux contenant les coordonnées des différentes figures sur l'écran.  
T: Total des rapports.  
L, C, K, V, P, I: Mémoires contenant les mises.  
E: Nombre aléatoire de "sauts" fait par l'aiguille.  
O\$: Contient la réponse à "Rejouez-vous".

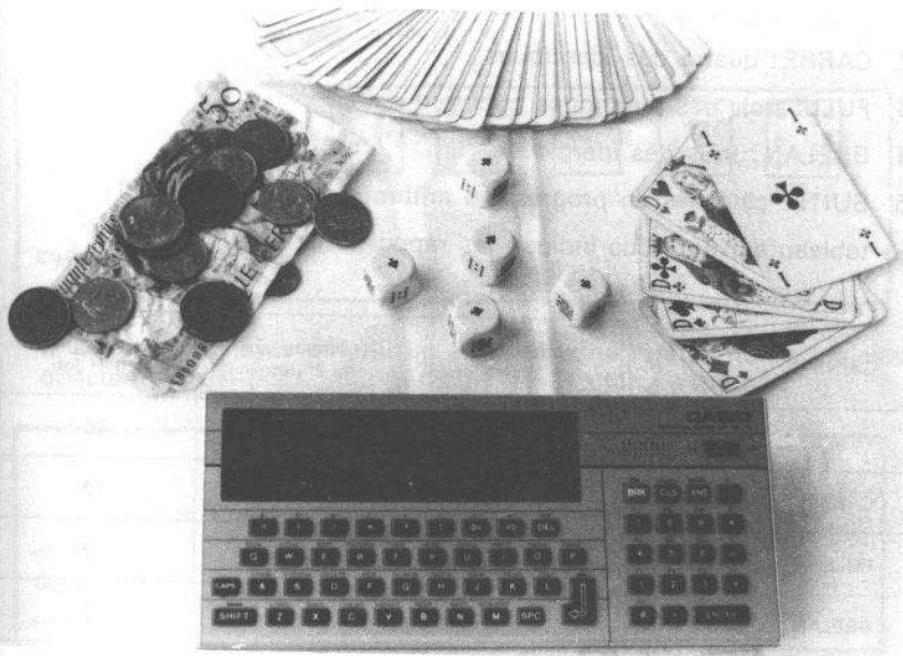
## LOTERIE

```
10 CLS :CLEAR :Q=127:W=15:DIM A!(12):  
DIM B!(12)  
20 LOCATE 6,1:PRINT "LOTERIE"  
30 DRAW(0,4)-(159,4)-(159,5)-(0,5):DR  
AW(0,17)-(159,17)-(159,18)-(0,18)  
35 LOCATE 4,3:PRINT "$$$$$$$$$$";  
40 FOR I=1 TO 12  
50 READ A!(I),B!(I)  
60 NEXT I  
70 DATA108,7,116,7,124,7,132,7,140,7,  
148,7,148,24,140,24,132,24,124,24,116,24
```

```
    75 DATA108,24
    80 T=50
    90 CLS
    95 LOCATE 13,0:PRINT "123456":LOCATE
13,3:PRINT "abcdef";
    97 DRAW(99,0)-(99,31)-(100,0)-(100,31)
)-(101,0)-(101,31)
    98 DRAW(157,0)-(157,31)-(158,31)-(158
,0)-(159,0)-(159,31)
    100 DRAW(Q,W)-(108,7)
    110 LOCATE 0,0:PRINT "Avoir":;:PRINT U
SING"#####";T
    115 L=0:C=0:K=0:U=0:P=0:I=0:REM mises
    120 LOCATE 0,1:INPUT "1",L
    125 GOSUB 600
    130 LOCATE 0,2:INPUT "c",C
    135 GOSUB 600
    160 LOCATE 4,1:INPUT "k",K
    165 GOSUB 600
    170 LOCATE 8,1:INPUT "u",U
    175 GOSUB 600
    180 LOCATE 4,2:INPUT "p",P
    185 GOSUB 600
    190 LOCATE 8,2:INPUT "i",I
    195 GOSUB 600
    200 IF L+C+K+U+P+I>T THEN GOSUB 1000:G
OTO 115
    210 T=T-L-C-K-U-P-I
    250 E=INT(RND*23)+2
    260 FOR X=1 TO E
    265 BEEP 1
    270 IF X>=13 THEN J=X-12 ELSE J=X
    280 F=A!(J):G=B!(J)
    290 DRAW(Q,W)-(F,G)
    300 IF (X-1)*(X-13)<0 THEN DRAWC(Q,W)
-(A!(J-1),B!(J-1))
    303 DRAWC(Q,W)-(108,24)
    305 NEXT X
```

```
310 REM resultats
320 BEEP :BEEP
330 IF G=7 THEN DRAW(F+5,0)-(F+5,G)-(F
-5,G)-(F-5,0):GOTO 360
340 DRAW(F-5,G)-(F+5,G)-(F+5,31)-(F-5,
31)-(F-5,G)
360 IF G=24 THEN 400
370 C=C*2:L=0:U=0:K=0
380 IF (F-116)*(F-132)*(F-148)=0 THEN
P=P*4:I=0 ELSE I=I*4:P=0
390 GOTO 500
400 L=L*2:C=0:I=0:P=0
410 IF (F-108)*(F-140)=0 THEN U=U*10:K
=0 ELSE K=K*5:U=0
500 FOR X=0 TO 2
510 LOCATE 0,X:PRINT TAB(12);
520 NEXT X
530 LOCATE 0,0:PRINT "Resultats:","RaP
Port:";L+C+U+K+P+I
535 LOCATE 0,2:PRINT "Avoir:";T;
540 T=T+L+C+K+U+P+I:IF T=0 THEN LOCATE
0,3:PRINT "Faillite!";:BEEP :GOTO 545
544 GOTO 550
545 BEEP :BEEP 1:IF INKEY$="" THEN 545
ELSE CLS :END
550 LOCATE 0,3:PRINT "Encore"::INPUT O
$
560 IF O$="N" THEN END ELSE 90
600 LOCATE 6,0:PRINT USING"#####";T-L-C
-K-U-P-I;
610 RETURN
1000 FOR X=1 TO 9 STEP 4
1010 FOR Y=1 TO 2
1020 LOCATE X,Y:PRINT " ";
1030 NEXT Y:NEXT X
1035 LOCATE 6,0:PRINT USING"#####";T
1036 BEEP :LOCATE 0,3:PRINT "TRICHEUR!!
";:FOR H=1 TO 10:BEEP :NEXT H
1037 BEEP :BEEP :BEEP :BEEP :LOCATE 0,3
:PRINT " ";
1040 RETURN
```

*Jeu de Yam*



Ce jeu est avec OTHELLO et BATAILLE NAVALE un des plus longs de notre livre. Cependant c'est aussi l'un des plus divertissant (si vous aimez gagner de l'argent fictivement bien sûr).

Le Yam est un jeu de dés d'origine chinoise que nous avons transformé pour le PB 700 en une sorte de poker sur dés.

### Voici la règle du jeu

- Vous avez cinq dés et une somme de 50 F pour commencer la partie.
- A chaque manche vous misez une somme inférieure ou égale à votre avoir et vous tentez de réaliser une figure de poker. Celles-ci, plus ou moins difficiles à obtenir, rapportent plus ou moins d'argent.
- Pour réaliser cette figure, vous pouvez rejeter un ou plusieurs dés qui ne vous plaisent pas. Mais vous ne pouvez pas le faire plus de deux fois. Notez aussi que vous pouvez ne rien avoir à changer.

Les différentes figures exécutables sont les suivantes:

1. YAM: la plus difficile : cinq dés identiques.
2. CARRÉ: quatre dés identiques.
3. FULL: trois dés identiques plus une paire.
4. BRELAN: trois dés identiques.
5. SUITE: cinq dés en progression arithmétique (5, 4, 3, 2, 1).

Le tableau suivant vous indique les rapports obtenus:

GAINS	obtenu au 1 <sup>er</sup> Jet	obtenu au 2 <sup>e</sup> Jet	obtenu au 3 <sup>e</sup> Jet
YAM	60	40	20
CARRÉ	30	20	10
FULL	15	10	5
BRELAN	6	4	2
SUITE	24	16	8

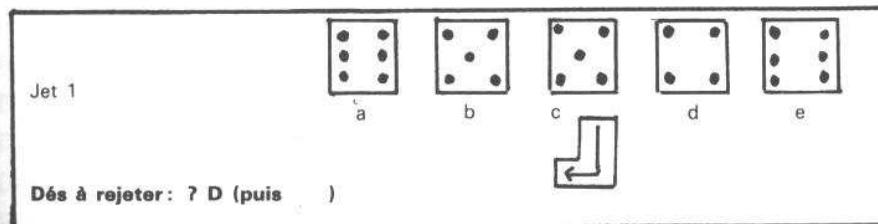
Les chiffres indiqués représentent le nombre de fois la mise.

**Exemple:** Si vous faites une suite au 2<sup>e</sup> jet et que vous avez misé 30 F, votre gain sera de  $30 \times 16 = 480$  F.

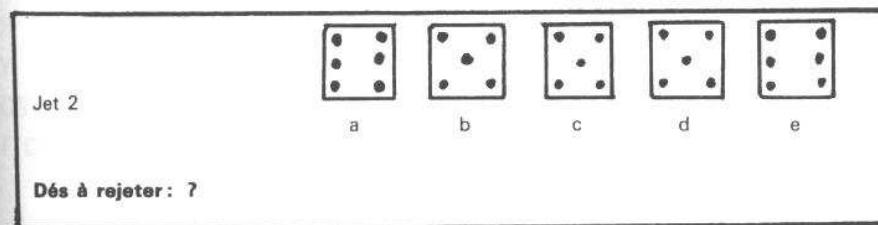
*Notez bien:* Chaque figure ne peut être effectuée qu'une seule fois. Si vous faites un deuxième FULL, celui-ci sera considéré comme un brelan.

Pour vous montrer comment utiliser le programme, nous vous avons exposé un exemple de partie.

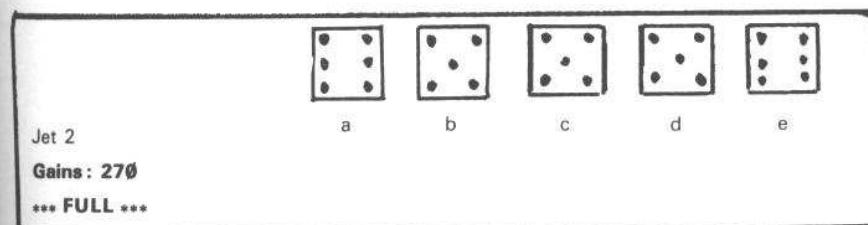
**Exemple de partie:** Vous avez 50 F, vous misez 30 F.



— *1<sup>er</sup> Écran:* Vous obtenez au 1<sup>er</sup> Jet 2 six, 2 cinq et 1 quatre. Vous visez le Full et vous changez le dé D.



— *2<sup>e</sup> Écran:* Vous avez obtenu un Full. Vous ne rejetez rien en faisant directement .



- 3<sup>e</sup> Écran: PB 700 vous indique vos gains.

Jet 2	a	b	c	d	e
Gains : 270					
Rejouez-vous ?	<input type="checkbox"/>				

- 4<sup>e</sup> Écran: Si vous voulez rejouer, vous faites directement  sinon vous écrivez N puis .

Total = 350 F
Mise = —

5<sup>e</sup> Écran: Vous avez voulu rejouer: on vous indique votre avoir et on vous demande votre mise pour la partie suivante.

### Description du programme

- 10-50 : Initialisation.
- 50-90 : Demande de la mise et affichage du total.
- 100-260 : Affichage des dés du jet 1.
- 270-340 : Questions sur le changement des dés.
- 500-590 : Sous-programme de changement de la valeur d'un dé.
- 600-990 : Calcul des résultats.
- 1000-1010: Calcul aléatoire de la valeur d'un dé.
- 1100-1110: Dessin d'un 1 sur un dé.
- 1200-1220: Dessin d'un 2 sur un dé.
- 1300-1320: Dessin d'un 3 sur un dé.
- 1400-1450: Dessin d'un 4 sur un dé.
- 1500-1520: Dessin d'un 5 sur un dé.

1600-1650: Dessin d'un 6 sur un dé.

1700-1710: Dessin d'un point.

## Variables utilisées

H1, H2, H3, H4, H5: Indicateurs de figure effectuée.

T : Contient le total en francs.

M : Mise.

N : Garde la valeur de la mise jusqu'au calcul du gain.

I : Nombre de jets.

A : Valeur du premier dé.

B : Valeur du 2<sup>e</sup> dé.

C : Valeur du 3<sup>e</sup> dé.

D : Valeur du 4<sup>e</sup> dé.

E : Valeur du 5<sup>e</sup> dé.

R\$ : Dés à rejouer.

Z\$ : Contient l'un des éléments de R\$ pour le changer.

P : Position d'un dé.

X : Valeur aléatoire calculée d'un dé.

Y : Indicateur de boucle.

O\$ : Réponse à la question "rejouez-vous?"

F, G : Données du programme de tracé d'un point.

J E U   D E   Y A M

```
10 CLS :CLEAR :H1=0:H2=0:H3=0:H4=0:H5  
=0  
20 LOCATE 0,1:PRINT CHR$(232);CHR$(23  
3);CHR$(234);CHR$(235);" JEU DE YAM ";  
30 PRINT CHR$(235);CHR$(234);CHR$(233  
);CHR$(232)  
40 DRAW(36,6)-(124,6)-(124,16)-(36,16  
)-(36,6)  
50 FOR U=1 TO 100:NEXT U
```

```

60 T=50
70 Y=0:CLS :PRINT "Vous avez";T;"F":L
OCATE 0,2:INPUT "Mise:",M
80 IF M*(M-T)>0 THEN 70
90 T=T-M:N=M:CLS
100 I=1
110 REM Debut du jeu
120 LOCATE 0,1:PRINT "Jet";I
130 LOCATE 5,0:PRINT "a b c d e"
140 FOR B=48 TO 144 STEP 24
150 DRAW(B,0)-(B,11)-(B+11,11)-(B+11,0)
)-(B,0)
160 NEXT B
170 GOSUB 1000:A=X
180 P=0:GOSUB 1000+A*100
190 GOSUB 1000:B=X
200 P=24:GOSUB 1000+B*100
210 GOSUB 1000:C=X
220 P=48:GOSUB 1000+C*100
230 GOSUB 1000:D=X
240 P=72:GOSUB 1000+D*100
250 GOSUB 1000:E=X
260 P=96:GOSUB 1000+E*100
270 IF I=3 THEN 600
280 LOCATE 0,2:INPUT "Des a rejouer:",R$
290 IF R$="" THEN 600
300 I=I+1:LOCATE 0,2:PRINT TAB(19);:LO
CATE 3,1:PRINT " "
310 FOR U=1 TO LEN(R$)
320 Z$=MID$(R$,U,1)
330 GOSUB 500
340 NEXT U
350 LOCATE 3,1:PRINT I:GOTO 270
500 IF Z$>"A" THEN 510
505 P=0:GOSUB 1000:A=X:GOTO 550
510 IF Z$>"B" THEN 520
515 P=24:GOSUB 1000:B=X:GOTO 550

```

```
520 IF Z$>"C" THEN 530
525 P=48:GOSUB 1000:C=X:GOTO 550
530 IF Z$<>"D" THEN 540
535 P=72:GOSUB 1000:D=X:GOTO 550
540 P=96:GOSUB 1000:E=X
550 DRAWC(50+P,2)-(50+P,9)-(57+P,9)-(5
7+P,2)-(56+P,2)-(56+P,8)-(51+P,8)-(51+P,
2)
560 DRAWC(53+P,5)-(53+P,6)-(54+P,6)-(5
4+P,5)
580 GOSUB 1000+X*100
590 RETURN
600 BEEP :BEEP :LOCATE 0,2:PRINT "Dern
ier jet! "
610 Y=Y+1
620 IF A>B THEN U=A:A=B:B=U
630 IF B>C THEN U=B:B=C:C=U
640 IF C>D THEN U=C:C=D:D=U
650 IF D>E THEN U=D:D=E:E=U
655 IF Y<4 THEN 610
660 IF A>E THEN 700
662 LOCATE 0,3:PRINT "****YAM***";
665 IF H1=1 THEN 890
670 M=M*20:H1=1
680 GOTO 900
700 IF A>D THEN 720
702 LOCATE 0,3:PRINT "***CARRE***";
705 IF H2=1 THEN 890
710 M=M*10:H2=1
715 GOTO 900
720 IF B>E THEN 730
725 GOTO 702
730 IF A>C THEN 750
735 IF D>E THEN 745
737 LOCATE 0,3:PRINT "***FULL***";
740 IF H3><0 THEN 745
742 M=M*5:H3=1:GOTO 900
```

```

745 IF H4<>0 THEN 890
748 M=M*3:H4=1:LOCATE 0,3:PRINT "##BRE
LAN##";:GOTO 900
750 IF B<>D THEN 770
752 LOCATE 0,3:PRINT "##BRELAN##";
755 IF H4><0 THEN 890
760 M=M*3:H4=1
765 GOTO 900
770 IF C><E THEN 820
775 IF A><B THEN 790
777 LOCATE 0,3:PRINT "##FULL##";
780 IF H3><0 THEN 790
785 M=M*5:H3=1:GOTO 900
790 IF H4><0 THEN 890
800 M=M*3:H4=1:LOCATE 0,3:PRINT "##BRE
LAN##";
810 GOTO 900
820 IF A<>B+1 THEN 890
830 IF B><C+1 THEN 890
840 IF C><D+1 THEN 890
850 IF D><E+1 THEN 890
855 LOCATE 0,3:PRINT "##SUITE##";
860 IF H5><0 THEN 890
870 M=M*8:H5=1:GOTO 900
890 M=0
900 IF I=1 THEN M=M*3
905 FOR U=1 TO 70:NEXT U
910 IF I=2 THEN M=M*2
920 T=T+M
930 IF T=0 THEN CLS :PRINT "Desole...."
      Vous etes ruine!!";:END
940 IF M>=N THEN LOCATE 0,2:PRINT "Gai
ns:";M-N;"F ";GOTO 980
950 LOCATE 0,2:PRINT "Pertes:";N-M;"F
"
980 LOCATE 0,3:INPUT "Re jouez-vous?",0
$
```

```
990 IF O$="N" THEN END ELSE 70
1000 X=INT(RND*6)+1
1010 RETURN
1100 F=53+P:G=5:GOSUB 1700
1110 RETURN
1200 F=50+P:G=2:GOSUB 1700
1210 F=56+P:G=8:GOSUB 1700
1220 RETURN
1300 GOSUB 1100
1310 GOSUB 1200
1320 RETURN
1400 FOR F=50+P TO 56+P STEP 6
1410 FOR G=2 TO 8 STEP 6
1420 GOSUB 1700
1430 NEXT G
1440 NEXT F
1450 RETURN
1500 GOSUB 1400
1510 GOSUB 1100
1520 RETURN
1600 FOR F=50+P TO 56+P STEP 6
1610 FOR G=2 TO 8 STEP 3
1620 GOSUB 1700
1630 NEXT G
1640 NEXT F
1650 RETURN
1700 DRAW(F,G)-(F+1,G)-(F+1,G+1)-(F,G+1)
)
1710 RETURN
```

# 3

## Jeux d'adresse

### **Mur de briques**

Attention à la balle ! Vous n'en avez que cinq pour une partie...

Le but de ce jeu de café bien connu est de rattraper la balle dans la raquette pour la renvoyer vers le mur afin qu'elle y détruise le plus de briques possible. Chaque brique vous fera gagner des points, et si le mur est totalement détruit, le programme le reconstruit entièrement, pour que vous puissiez continuer à vous livrer à votre fureur destructrice.

Utilisez les touches + et / pour déplacer votre raquette vers le bas ou vers le haut.

### **Structure du programme**

10-35 : Affichage du terrain.

40-70 : Initialisation des diverses variables.

80 : Attente du service.  
100-120 : Scrutation du clavier.  
140-150 : Mouvement de la raquette.  
200-225 : Mouvement de la balle.  
250-260 : La balle rencontre-t-elle un obstacle ?  
265-280 : Rebond aléatoire contre la raquette ou le fond du terrain.  
300-330 : La balle rencontre-t-elle une brique ? Si oui, effacement de la brique, rebond et affichage du score augmenté.  
350-370 : Le mur est entièrement détruit. On le reconstruit.  
400-500 : Le joueur a perdu une balle. S'il lui en reste, retour au service, sinon le jeu est fini.

### Variables utilisées

X, Y : Coordonnée DRAW de la balle.  
E, F : Vitesse de la balle.  
A, B : Coordonnées LOCATE de la balle.  
R : Position DRAW de l'extrémité inférieure de la raquette.  
S, M, N : Score, nombre de balles restantes, nombre de blocs non détruits.

### M U R   D E   B R I Q U E S

```
1 REM MUR DE BRIQUES
10 REM Terrain
15 CLS
20 FOR I=0 TO 3
25 LOCATE 15,I:PRINT CHR$(135);CHR$(1
35);CHR$(135);CHR$(135);
30 NEXT I
35 DRAW(32,0)-(32,31)-(159,31)-(159,0
)
```

```

40 REM Variables
45 R=13:REM Position raquette
57 DRAW(37,R)-(37,R+4)
60 S=0:M=5:REM score, nb. balles
62 N=16:REM Nb. blocs
65 X=38:Y=15:REM balle
70 F=SGN(.5-RND)*5:E=2*INT(2+3*RND)
80 IF INKEY$="" THEN 80
100 REM Boucle de Jeu
105 IF INKEY$="" THEN 200
110 IF INKEY$="/" THEN R=R-4:GOTO 140
115 IF INKEY$ "+" THEN R=R+4:GOTO 140
120 GOTO 200
140 REM Mouvement raquette
145 DRAWC(37,0)-(37,30)
150 DRAW(37,R)-(37,R+4)
200 REM Mouvement balle
205 IF Y=0 THEN F=ABS(F) ELSE IF Y=30
    THEN F=-ABS(F)
210 DRAWC(X,Y)
215 X=X+E:Y=Y+F
220 DRAW(X,Y)
225 IF X>38 THEN IF X<120 THEN 100
250 REM Choc ?
255 IF X>38 THEN IF X<158 THEN 300
260 IF POINT(X+SGN(E),Y)=0 THEN 400
265 REM rebond/raquette ou fond terrain
270 F=SGN(RND-.5)*5
275 E=-SGN(E)*2*INT(2+3*RND)
280 GOTO 100
300 REM Choc bloc ?
305 IF POINT(X+SGN(E),Y)=0 THEN 100
310 A=INT(X/8):B=INT(Y/8)
315 LOCATE A,B:PRINT " ";
317 DRAW(120,31)-(152,31)
320 S=S+(A-14)

```

```
325 LOCATE 0,1:PRINT USING"##";S
327 E=-E
330 N=N-1:IF N>0 THEN 100
350 REM Mur detruit !
355 FOR I=0 TO 3
360 LOCATE 15,I:PRINT CHR$(135);CHR$(1
    35);CHR$(135);CHR$(135);
365 NEXT I
370 GOTO 62
400 REM Perte balle
402 DRAWC(X,Y):M=M-1
404 LOCATE 0,2:PRINT M
415 IF M>0 THEN 65
420 LOCATE 5,0:PRINT "*GAME OVER*"
490 IF INKEY$<>"" THEN 490
500 IF INKEY$="" THEN 500 ELSE 10
```

## Ping-pong

Voici une version PB 700 du Ping-Pong sur écran, jeu d'adresse et de rapidité extrêmement répandu sur les ordinateurs personnels.

L'adaptation au PB 700 pose naturellement des problèmes particuliers puisque l'écran à cristaux liquides n'est ni aussi grand ni aussi lumineux qu'un écran cathodique. D'autre part, l'absence de manettes de jeu oblige, pour manœuvrer les raquettes, à recourir au clavier.

Le joueur de gauche utilisera les touches Q et Z pour déplacer sa raquette respectivement vers le haut ou vers le bas. Le joueur de droite fera de même avec les touches / et +. Si ces touches ne vous conviennent pas, une très simple modification des lignes 115 à 130 du programme vous permettra d'adapter le jeu à vos goûts.

La programmation en BASIC fait évidemment que ce jeu ne peut prétendre rivaliser avec des logiciels écrits en langage machine, mais c'est tout de même un excellent exemple des possibilités surprenantes du PB 700. En une soixantaine de lignes, on peut jouer au tennis, et de plus, la petitesse

de l'écran se transformant en avantage, on peut l'emporter... dans sa poche !

### Structure du programme

- 1-7 : Présentation.
- 10-30 : Dessin du terrain.
- 35-55 : Initialisation.
- 60-65 : Positionnement initial des raquettes.
- 105-135 : Scrutation du clavier.
- 140-150 : Mouvement des raquettes à l'écran.
- 200-220 : Affichage de la balle.
- 225-270 : Rebond aléatoire contre une raquette.
- 300-340 : Marquage d'un point et service.
- 400-500 : Acclamation du vainqueur.

### Variables utilisées

A, B: Ordonnées des deux raquettes, leurs abscisses étant en permanence 19 et 141.

E, F: Vitesse de la balle.

$E = \pm 5$  au service.  
 $\pm 5,10$  ou 15 ensuite (E doit diviser 120).  
 $F = \pm 5$  (F doit diviser 30).

S, T : Scores de A et B.

D : Discriminateur: vaut A ou B.

P : Discriminateur: vaut 19 ou 141.

P I N G   P O N G

```
1 REM
5 CLS :PRINT TAB(25);"PING-PONG"
```

```

    7 FOR I=1 TO 500:NEXT I
10 REM Terrain
15 CLS
20 DRAW(9,0)-(9,31)-(151,31)-(151,0)
30 LOCATE 0,0:PRINT "A":LOCATE 19,0:P
RINT "B"
35 REM Variables
40 A=13:B=13:REM Raquettes
45 S=0:T=0:REM Scores
50 X=20:Y=15:REM Balle
55 E=5:F=SGN(.5-RND)*5:REM Vitesse
60 DRAW(19,A)-(19,A+4)
65 DRAW(141,B)-(141,B+4)
70 IF INKEY$="" THEN 70
100 REM Boucle de Jeu
105 IF INKEY$="" THEN 200
115 IF INKEY$="Q" THEN P=19:A=A-4:D=A:
GOTO 140
120 IF INKEY$="Z" THEN P=19:A=A+4:D=A:
GOTO 140
125 IF INKEY$="/" THEN P=141:B=B-4:D=B
:GOTO 140
130 IF INKEY$="+" THEN P=141:B=B+4:D=B
:GOTO 140
135 GOTO 200
140 REM Mouvement des raquettes
145 DRAWC(P,0)-(P,30)
150 DRAW(P,D)-(P,D+4)
200 REM Mouvement de la balle
205 IF Y*(Y-30)=0 THEN F=-F
210 DRAWC(X,Y)
215 X=X+E:Y=Y+F
220 DRAW(X,Y)
225 IF (X-20)*(X-140)<0 THEN 100
250 REM Choc contre raquette ?
255 IF X<99 THEN D=A ELSE D=B
260 IF (Y-D)*(Y-D-4)>0 THEN 300
265 F=SGN(.5-RND)*5

```

```

267 E=-SGN(E)*5*INT(2.5*RND+1)
270 GOTO 100
300 REM Point !
305 BEEP 1
310 IF X>99 THEN S=S+1 ELSE T=T+1
313 DRAWC(X,Y)
315 LOCATE 0,1:PRINT USING"#";S
320 LOCATE 19,1:PRINT USING"#";T
325 IF (S-9)*(T-9)=0 THEN 400
329 REM Remise en Jeu
330 IF D=A THEN E=-5:X=140 ELSE E=5:X=
20
335 IF INKEY$<>"" THEN 335
340 IF INKEY$="" THEN 340 ELSE 100
400 REM Fin du Jeu
405 BEEP 1:BEEP 1:CLS
410 IF S>T THEN PRINT "A"; ELSE PRINT
"B";
420 PRINT " vainqueur";S;" a";T;:DRAWC
115,1-(114,0)
500 IF INKEY$="" THEN 500 ELSE 10

```

### Tir forain

Vous êtes maintenant devant un stand de tir, une 22 long Rifle en forme de PB 700 dans la main et vous devez, sous peine de paraître ridicule, détruire 15 cibles qui vont apparaître devant vous.

Quatre niveaux de difficulté vous sont offert:

- *Niveau 1*: Vous disposez de 24 balles.
- *Niveau 2*: Vous disposez de 21 balles.
- *Niveau 3*: Vous disposez de 18 balles.
- *Niveau 4*: Vous ne disposez que de 15 balles.

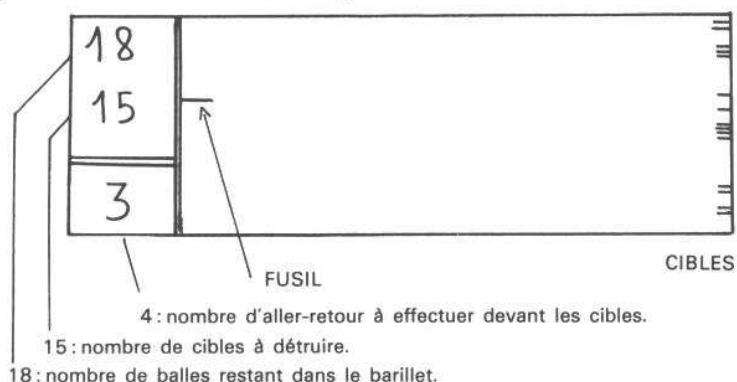
Pour 15 cibles



Pour corser un peu l'épreuve, vous n'aurez qu'un temps limité à 4 passages devant les cibles. En effet, les cibles sont fixes et vous vous déplacez.

Pour tirer, appuyez sur n'importe quelle touche noire.

**Exemple d'écran en début de jeu au niveau 3 :**



- Appuyez sur n'importe quelle touche noire pour recommencer une partie.

### Description du programme

10-23 : Mise en forme du titre.  
 30-60 : Détermination du niveau et initialisation des variables.  
 70-110 : Placement aléatoire des cibles.  
 120-150 : Dessin du champ de tir.  
 165-180 : Boucle de jeu.  
 190-330 : Tir et examen du résultat.  
 340-1510: Résultat de la partie.

### Mémoires utilisées

A\$ : Contient l'ordre de tir.  
 N : Niveau.  
 L : Nombre de tirs autorisés.  
 Z : Nombre de cibles.  
 X : Emplacement des cibles.  
 R : Nombre d'aller-retours effectués.  
 P : Position du fusil.  
 Q : Nombre initial de tirs autorisés.  
 O\$ : Contient la réponse à la question "Rejouez-vous ?"

T I R F O R A I N

```

10 CLS :CLEAR
20 PRINT TAB(25);"TIR FORAIN"
22 DRAW(0,5)-(159,5):DRAW(0,17)-(159,
17)
  
```

```
23 DRAW(0,11)-(40,11):DRAW(120,11)-(1
59,11)
30 LOCATE 0,3:INPUT "Niveau(1..4)";N
40 IF N*(N-4)>0 THEN 30
50 L=-3*N+27:Q=L
60 Z=15:P=INT(RND*32):R=4:J=1
70 CLS
80 FOR I=1 TO 15
90 X=INT(RND*32)
100 IF POINT(159,X)=1 THEN 90 ELSE DRA
W(157,X)-(159,X)
110 NEXT I
120 DRAW(24,0)-(24,31)-(25,31)-(25,0):
DRAW(26,P)-(28,P)
130 LOCATE 0,0:PRINT USING"##";L:LOCAT
E 0,1:PRINT USING"##";Z
140 LOCATE 0,3:PRINT USING"##";R;
150 DRAW(0,19)-(24,19)-(24,20)-(0,20)
165 A$=INKEY$
168 IF A$><"" THEN 250
170 IF P*(31-P)=0 THEN 200
180 P=P+J:DRAWC(26,P-J)-(28,P-J):DRAW(
26,P)-(28,P)
190 GOTO 165
200 J=-J:R=R-1:IF R<0 THEN 1000
210 LOCATE 0,3:PRINT USING"##";R;
220 GOTO 180
250 L=L-1
260 DRAW(29,P)-(158,P)
270 IF POINT(159,P)<>0 THEN 300
280 DRAWC(29,P)-(158,P):LOCATE 0,0:PRI
NT USING"##";L
290 IF L=0 THEN 1000 ELSE 165
300 BEEP :DRAWC(29,P)-(159,P):Z=Z-1
310 IF L*Z=0 THEN 1000
320 LOCATE 0,1:PRINT USING"##";Z
330 LOCATE 0,0:PRINT USING"##";L:GOTO
165
```

```

1000 CLS :PRINT "RESULTATS:",  

1005 IF Z=0 THEN 1100  

1010 IF L=0 THEN 1200  

1020 IF R=-1 THEN 1300  

1100 PRINT "Destruction totale en";Q-L  

;" tir";  

1105 IF Q-L>1 THEN PRINT "s." ELSE PRIN  

T ".."  

1110 GOTO 1500  

1200 PRINT "Vous visez mal!","Il reste"  

;Z;" cible(s)"  

1210 GOTO 1500  

1300 PRINT "Vous etes tres lent!";15-Z;  

" cibles detruites":GOTO 1500  

1500 INPUT "ReJouez-vous:",O$  

1510 IF O$="N" THEN END ELSE 10

```

### **Tir balistique**

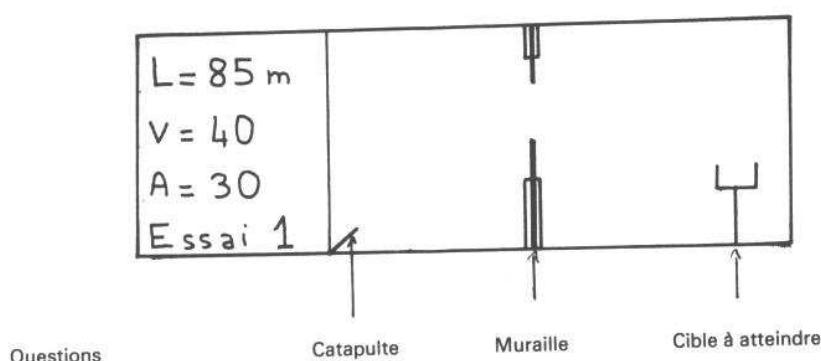


Ce jeu de tir est plus difficile que le Tir Forain précédemment décrit. En effet, il s'agit pour vous de détruire un objectif placé derrière une muraille. Vous disposez pour cela d'une catapulte et de trois boulets par cible.

Votre "PB 700 catapulte" vous demande la vitesse initiale de votre projectile et l'angle de tir.

A vous de vous montrer bon artilleur. Soyez persévérant car ce n'est pas très facile.

Au début de la partie, votre écran se présente de la manière suivante :



A chaque tir, plusieurs paramètres sont générés aléatoirement :

- La position de la muraille.
- La largeur du trou dans la muraille.
- La hauteur de la cible.

Le reste est constant.

### Description du programme

- 10-110 : Initialisation.
- 120 : Génération aléatoire des paramètres précédents.
- 130-180 : Dessin du champ de tir.
- 190-220 : Questions sur le tir.
- 230-280 : Déplacement du boulet jusqu'à la muraille.
- 290-360 : Passage de la muraille.
- 370-420 : Déplacement du boulet jusqu'à la cible.

430-600 : Affichage du résultat.

9000-9010: Temporisation.

## Mémoires utilisées

X : Hauteur de la muraille.

L: Largeur du trou de la muraille.

H: Hauteur de la cible.

Z : Position de la muraille.

V : Vitesse initiale.

A : Angle de tir.

La position du boulet est donné par la formule de balistique :

$$y = \frac{g x^2}{2v_0^2 \cos^2 A} + x \tan A + y_0 \quad \text{ou } g = 9,81 \text{ m/s}^2$$

$$W = \frac{g}{2v_0^2 \cos^2 A} \quad (\text{terme en facteur de } x^2)$$

T : Terme en facteur de x.

R : yo

I : Abscisse du boulet.

Y : Ordonnée du boulet.

Q : Nombre d'essais dans le tir.

O\$ : Réponse à la question (Rejouez-vous?).

## T I R      B A L I S T I Q U E

```
10 CLS :CLEAR :Q=1
20 LOCATE 0,1:PRINT "    TIR BALISTIQU
E"
30 FOR I=0 TO 2 STEP 2
40 FOR J=0 TO 19
50 LOCATE J,I:PRINT CHR$(135);
60 NEXT J:NEXT I
```

```

70 GOSUB 9000
80 CLS
90 PRINT "3 essais par cible. Voici votre champ de tir:"
100 GOSUB 9000
110 CLS
120 X=INT(RND*10)+15:L=INT(RND*4)+3:H=INT(RND*10):Z=INT(RND*31)+100
130 DRAW(57,0)-(57,31)-(159,31)-(159,0)
)
140 DRAW(144,31)-(144,31-H)-(142,31-H)
-(146,31-H)-(146,28-H)
150 DRAW(142,31-H)-(142,28-H)
160 DRAW(Z,31)-(Z,31-X):DRAW(Z,0)-(Z,3-X-L)
170 DRAW(Z-1,31)-(Z-1,31-(X/2)):DRAW(Z+1,31)-(Z+1,31-(X/2))
180 DRAW(58,30)-(62,26)
190 LOCATE 0,0:PRINT "L=85m"
200 LOCATE 0,1:INPUT "U=",U:LOCATE 0,2
:INPUT "A=",A
210 LOCATE 0,3:PRINT "Essai";Q;
220 W=5/U^2/(COS(A))^2:T=TAN(A)+114*W:
R=57^2*W+31+57*TAN(A)
230 FOR I=58 TO Z-3 STEP 3
240 Y=W*I^2-I*T+R
250 IF Y<31 THEN 270
260 BEEP :LOCATE 0,0:PRINT "PAF! ":GO
TO 430
270 DRAW(I,Y):DRAWC(I-3,C)
280 C=Y:NEXT I
290 FOR I=Z-3 TO Z+1
300 Y=W*I^2-I*T+R
310 IF I>Z THEN 350
320 IF Y>31-X-L THEN 330
325 BEEP :LOCATE 0,0:PRINT "PAF! ":GO
TO 430

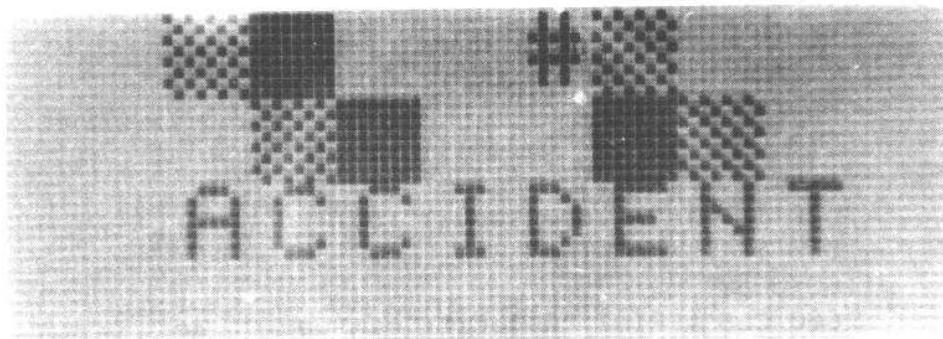
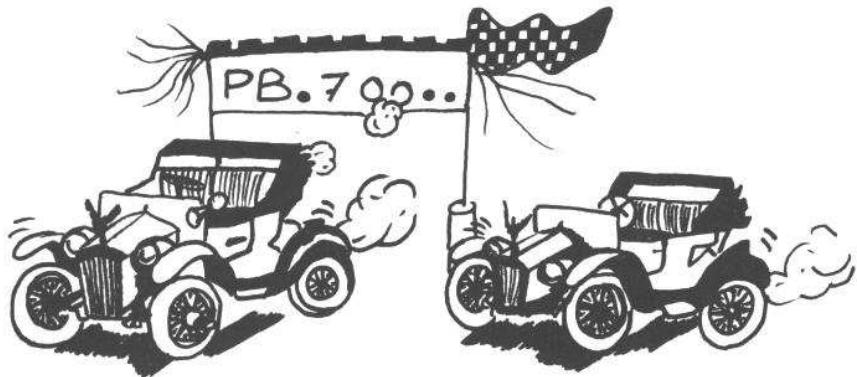
```

```
330 IF Y<31-X THEN 350
340 GOTO 325
350 DRAW(I,Y):DRAWC(I-1,C)
360 C=Y:NEXT I
370 FOR I=2+1 TO 159 STEP 3
380 Y=W*I^2-T*I+R
390 IF (I-142)*(I-147)>0 THEN 410
400 IF (Y-30+H)*(Y-26+H)<=0 THEN 500
410 DRAW(I,Y):DRAWC(I-3,C)
420 C=Y:NEXT I
430 LOCATE 0,3:PRINT "Rate!! ";:GOSUB
9000
435 IF Q<3 THEN Q=Q+1 ELSE 600
440 LOCATE 0,0:PRINT "      ", ","
      ", "
      ", ";
450 GOTO 190
500 BEEP 1:BEEP 1:LOCATE 0,0:PRINT "To
uche ":GOSUB 9000
510 CLS :PRINT "Objectif atteint","a 1
essai";Q;" .",
520 INPUT "ReJouez-vous?",O$
530 IF O$="N" THEN END ELSE CLEAR :Q=
1:GOTO 80
600 CLS :PRINT "Objectif manque      de
finitivement!":GOTO 520
9000 FOR I=1 TO 200:NEXT I
9010 RETURN
```

num  
eb  
etns  
oonl

fnscr

## Rallye



Imaginez-vous maintenant confortablement installé dans le cockpit d'un PB 700 en forme de Formule 1. Vous devez vous déplacer sur un circuit de dimension olympique et atteindre l'arrivée en faisant le moins d'accidents possible. Votre accélérateur est collé au plancher. Vous ne pouvez donc que tourner à droite ou à gauche pour éviter les obstacles.

- Pour aller à droite, vous appuyez de manière continue sur [enter].
- Pour aller à gauche, vous appuyez de manière continue sur [J].

Vous avez le choix entre 10 niveaux de difficulté :

- *Niveau 1* : Facile : la route est très large au début et va en devenant de plus en plus étroite.

- Niveau 10: La route est très étroite du début jusqu'à la fin.
- Niveau 2 → 9 : Difficulté croissante.
- Appuyez sur n'importe quelle touche noire pour commencer la partie.
- Une tonalité aiguë vous annonce un rétrécissement de la chaussée.

### Description du programme

1-15 : Initialisation et demande du niveau.  
 17-29 : Préparation au départ.  
 30-180 : Boucle de jeu.  
 190-200 : Rétrécissement de la route.  
 290-320 : Résultats.  
 1000-1080: Dessin du tronçon de route à l'arrivée.

### Variables utilisées

D : Difficulté et largeur initiale de la route.  
 G : Temps que durera une phase de largeur de la route.  
 I : Position de la voiture.  
 K : Position du bord gauche de la route.  
 A\$ : Scrutation du clavier (changement de direction).  
 X : Nombre d'accidents.

### R A L L Y E

```

1 CLS
2 LOCATE 5,0:PRINT CHR$(254);" RALLY
E ";CHR$(254),
5 DRAW(0,8)-(159,8)
15 LOCATE 0,2:PRINT "Niveau: (0..10):"
::INPUT D
  
```

```

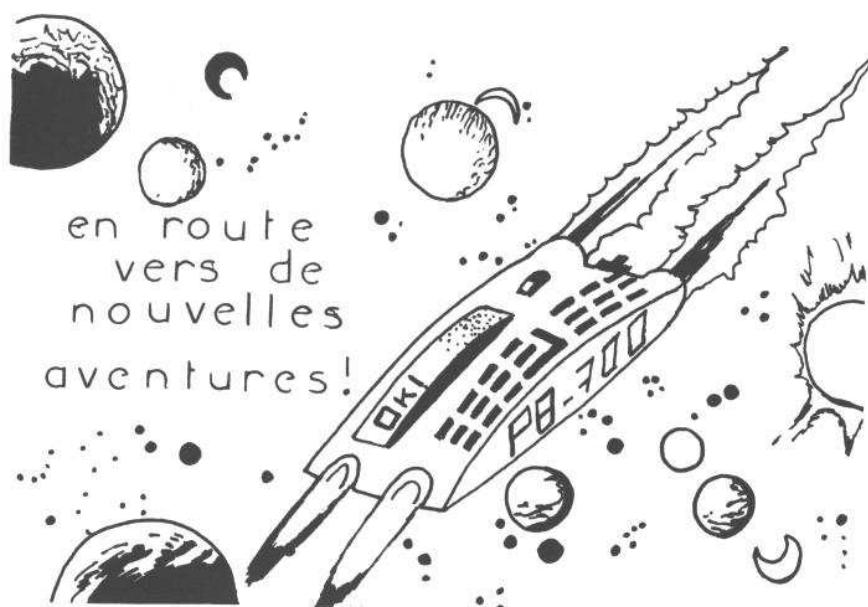
17 D=14-INTD
20 I=10:K=10-INT(D/2):X=0
25 G=INT(105/(D-3))
26 CLS :LOCATE 0,0:PRINT "ATTENTION A
U DEPART!":LOCATE 10,1:PRINT "#"
27 DRAW(0,8)-(159,8)
29 IF INKEY$="" THEN 29
30 FOR T=1 TO G
40 A$=INKEY$
50 IF ASC(A$)=13 THEN I=I-1
60 IF ASC(A$)=23 THEN I=I+1
70 LOCATE K,3:PRINT CHR$(254);CHR$(13
5);TAB(K+D);CHR$(135);CHR$(254)
80 IF POINT(8*I,2)=1 THEN 100
90 LOCATE I,0:PRINT "#":GOTO 150
100 BEEP :BEEP :I=W+INT(D/2)
105 LOCATE I+1,0:PRINT "PAF!"
110 LOCATE I,0:PRINT "#":X=X+1
140 IF INKEY$="" THEN 140
150 L=INT(RND*3)-1
160 W=K:K=K+L:IF K<0 THEN K=0
170 IF K+D>16 THEN K=16-D
180 NEXT T
190 D=D-1:IF D=3 THEN GOSUB 1000:GOTO
290
200 BEEP 1:BEEP 1:BEEP 1:GOTO 30
290 CLS :LOCATE 3,3:PRINT "!!ARRIUEE!!
"
300 LOCATE 3,3:PRINT X;" ACCIDENT";
305 IF X>1 THEN PRINT "S." ELSE PRINT
"
"
310 IF INKEY$="" THEN 310
320 X=0:GOTO 1
1000 FOR T=1 TO 3
1005 IF I-1-T<0 THEN I=1+T
1010 LOCATE I-1-T,3:PRINT CHR$(135);TAB
(I+1+T);CHR$(135)

```

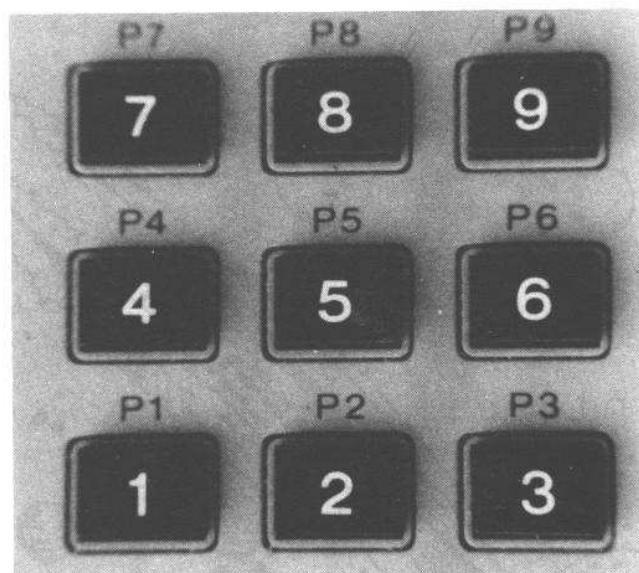
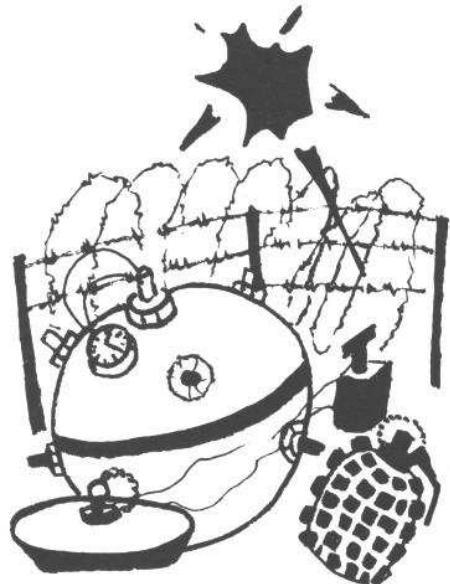
```
1020 LOCATE I,0:PRINT "#"  
1030 NEXT T  
1040 FOR T=1 TO 15  
1050 LOCATE I-4,3:PRINT CHR$(135);TAB(I  
+4);CHR$(135)  
1060 LOCATE I,0:PRINT "#"  
1070 NEXT T  
1080 RETURN
```

# 4

## Jeux d'aventures



*Champ de mines*



Nous sommes en 1917, vous êtes soldat sur le front et vous devez absolument effectuer une reconnaissance en territoire ennemi. Pour cela, vous devez franchir un champ truffé de mines.

Mais votre état major, qui tient beaucoup à vous, vous a donné un appareil détectant les mines si elles sont à un pas de vous !. Le seul problème est que ce merveilleux détecteur ne peut pas vous dire dans quelle direction vous risquez la mort. Il s'agit donc d'être prudent.

La partie droite de l'écran de votre PB 700 représente le champ de mines et le point en haut à gauche: c'est vous !!

A l'initialisation, le programme vous demande un niveau de difficulté. Le niveau minimum est Ø mais il n'est pas limité vers le haut:

En effet: Ø → 5 mines  
1 → 1Ø mines  
2 → 15 mines                dans le champ  
3 → 2Ø mines  
4 → 25 mines  
etc...

Nous n'avons jamais réussi à dépasser le niveau 5 (3Ø mines). A vous d'essayer donc.

Vous pouvez vous diriger dans 4 directions avec les touches suivantes:

2 vers le bas  
6 vers la droite  
8 vers le haut  
4 vers la gauche.

Dès qu'une mine est en vue (à un saut de vous, ne l'oubliez pas !) le sigle ALERTE ROUGE apparaît sur la gauche de votre écran.

Le nombre de mines est affiché en bas à gauche.

Si vous arrivez vivant au but, l'ordinateur placera les mines sur l'écran pour que vous puissiez voir à quoi vous avez échappé.

Vous recommencez en appuyant sur n'importe quelle touche noire.

## Description du programme

10-40 : Initialisation et demande du niveau.  
50-170 : Génération aléatoire des mines dans un tableau A !(8,24).  
180-200 : Dessin de l'écran.  
210-340 : Boucle de jeu.  
900-910 : Fin de l'alerte rouge.  
1000-1020: Début de l'alerte rouge.  
2000-2080: EXPLOSION.  
3000-3011: Placement des mines en cas de succès.  
3012-3300: Félicitations.

## Mémoires utilisées

N : Niveau.  
M : Nombre de mines.  
A !(10,26): Tableau représentant le champ de mines.  
X,Y : Abscisse et ordonnée d'une mine.  
I,J : Coordonnées d'adressage du tableau.  
Z\$ : Scrutation de l'écran (déplacement).  
C : Indicateur d'alerte.  
O\$ : Contient la réponse à "Rejouez-vous?".

\*\*\*\* C H A M P D E M I N E S \*\*\*\*

```
10 CLS :CLEAR
20 LOCATE 0,1:PRINT "** CHAMP DE MINE
S **";
30 DRAW(0,17)-(159,17)-(159,18)-(0,18
)
35 DRAW(0,4)-(159,4)-(159,5)-(0,5)
40 LOCATE 0,3:INPUT "Niveau:",N
50 M=5*N+5
```

```
60 CLS :LOCATE 0,0:PRINT "Placement d
es mines","Veuillez Patienter."
70 DIM A!(10,26)
80 FOR I=1 TO M
85 LOCATE 9,3:PRINT USING"##";M-I+1;
90 X=INT(RND*8)+2:Y=INT(RND*22)+4
100 A!(X,Y)=2
110 FOR K=-1 TO 1 STEP 2
115 IF A!(X+K,Y)=2 THEN 125
120 A!(X+K,Y)=1
125 NEXT K
130 FOR K=-1 TO 1 STEP 2
135 IF A!(X,Y+K)=2 THEN 145
140 A!(X,Y+K)=1
145 NEXT K
170 NEXT I
180 BEEP 1:CLS
190 DRAW(62,0)-(62,31)-(63,31)-(63,0)
200 LOCATE 1,3:PRINT "M=";M;
220 I=2:J=2:C=1
230 X=4*I+57:Y=4*I-7
240 IF A!(I,J)=0 THEN IF C=1 THEN 900
ELSE 260
250 IF A!(I,J)=1 THEN 1000 ELSE 2000
260 DRAW(X,Y)-(X+1,Y)-(X+1,Y+1)-(X,Y+1)
)
275 IF J=25 THEN BEEP 1:BEEP 1:GOTO 30
00
290 Z$=INKEY$
295 IF Z$="" THEN 290
300 IF Z$="8" THEN I=I-1
310 IF Z$="6" THEN J=J+1
320 IF Z$="2" THEN I=I+1
330 IF Z$="4" THEN J=J-1
340 GOTO 230
900 LOCATE 0,0:PRINT " _____", " _____"
```

```
910 C=0:GOTO 260
1000 LOCATE 0,0:PRINT "*ALERTE","*ROUGE
"
1020 BEEP :C=1:GOTO 260
2000 DRAW(X,Y)-(X+1,Y)-(X+1,Y+1)-(X,Y+1
)
2005 LOCATE 0,2:PRINT "*!BOUM!"
2010 FOR H=1 TO 30
2030 DRAW(X,Y)-(INT(RND*75)+65, INT(RND*
32))
2040 NEXT H
2045 FOR T=1 TO 200:NEXT T
2050 CLS
2060 PRINT "Vous venez de sautersur une
mine...","Condoleances..."
2070 LOCATE 0,3:INPUT "ReJouez-vous";O$
2080 IF O$="N" THEN END ELSE 10
3000 FOR I=2 TO 9
3001 FOR J=4 TO 25
3002 IF A!(I,J)><2 THEN 3010
3003 X=4*j+57:Y=4*i-7
3004 DRAW(X,Y)
3010 NEXT J
3011 NEXT I
3012 IF INKEY$="" THEN 3012
3015 CLS
3020 PRINT "Vous etes Passe!!! Bravo."
3030 GOTO 2070
```

## *L'escalier infernal*

Ce n'est pas parce qu'il ne comporte que quatre marches qu'il est aisé d'atteindre son sommet.

---

Pour passer d'une marche à l'autre, il vous faudra tout d'abord attendre l'apparition (imprévisible !) d'un trou dans le plafond...

Parvenu à un niveau intermédiaire, méfiez-vous des trappes qui peuvent s'ouvrir sous vos pieds, ainsi que des boulets de canon qui peuvent balayer tout un étage. Si l'un d'entre eux vous atteint, vous perdez une vie et vous n'en avez que cinq...

Vous avez trois commandes à votre disposition : et pour vous déplacer sans changer d'étage et pour sauter à l'étage supérieur... à condition bien sûr que cela soit possible : si vous faites une chute, vous perdez deux points. Mais si vous parvenez enfin au sommet, vous gagnez vingt points.

Sachez être attentif, si vous ne voulez pas finir avec un score négatif.

## Structure du programme

- 1-35 : Initialisations.
- 50-65 : Affichage initial du terrain.
- 95 : Position initiale du joueur.
- 105-130 : Mouvement des planchers.  
La ligne 120 règle les probabilités d'apparition des trous et des boulets : respectivement 0,3 et 0,05.
- 150-165 : Scrutation du clavier.
- 170-195 : Mouvement horizontal du joueur.
- 200-275 : Tests:
  - 220-240 : Chute à travers une trappe.
  - 250-275 : Choc contre un boulet (et fin du jeu si V=0).
- 300-335 : Tentative de saut à l'étage supérieur.
- 350-385 : Le joueur est arrivé en haut. Il marque 20 points et... retourne en bas.

## Variables utilisées

P\$(4)\*15: 4 "étages", comportant 15 caractères: soit un plafond CHR\$(134), soit un trou CHR\$( ), soit un boulet CHR\$(236).

S,V : Score et décompte des vies.

X,Y : Coordonnées "DRAW" du joueur.

A,B : Coordonnées "LOCATE" du joueur.

D : Déplacement horizontal.

## ESCALIER INFERNAL

```
1 REM
2 CLS :LOCATE 1,2:PRINT "ESCALIER I
NFERNAL"
5 REM init
10 CLEAR :DIM P$(4)*15
15 FOR I=1 TO 4
20 FOR J=1 TO 15
22 P$(I)=P$(I)+CHR$(148)
25 NEXT J
30 NEXT I
35 S=0:U=5:REM Score $ vies
50 REM Terrain
55 CLS :DRAW(31,0)-(31,32)-(152,32)-
152,0)
65 FOR I=1 TO 4:LOCATE 4,I-1:PRINT P$(
I);:NEXT I
95 X=43:Y=31
99 REM
100 REM Boucle de jeu
101 REM-----
105 REM Mouvement planchers
110 I=1+INT(4*RND)
115 R=RND
120 C$=CHR$(134-58*SGN(.3-R)+44*SGN(R-
.95))
```

```
125 P$(I)=C$+LEFT$(P$(I),14)
130 LOCATE 4,I-1:PRINT P$(I);
145 REM
150 T=ASC(INKEY$)
155 IF T=29 THEN D=-8:GOTO 175 180
160 IF T=28 THEN D=8:GOTO 175 180
165 IF T=13 THEN 300 305
170 D=0
175 REM Mouvement Joueur
180 DRAWC(X,Y)-(X,Y-5)
185 X=X+D
187 IF X>147 THEN X=35
188 IF X<35 THEN X=147
190 DRAW(X,Y)-(X,Y-5)
195 IF Y<0 THEN 350 355
199 REM
200 REM Tests
201 REM-----
205 A=INT(X/8):B=INT(Y/8)
207 IF B=3 THEN 240
215 IF MID$(P$(B+2),A-3,1)=CHR$(148) THEN 240
220 REM Chute!
225 LOCATE A,B:PRINT "!";
230 S=S-2:LOCATE 0,0:PRINT USING"####";
S
235 Y=Y+8:GOTO 170
240 IF ASC(MID$(P$(B+1),A-3,1))<>236 THEN 100 100
250 REM Paf!
255 LOCATE 0,2:PRINT "AIE":BEEP 1
260 LOCATE A,B:PRINT CHR$(254);
262 LOCATE 0,2:PRINT TAB(3)
265 U=U-1:IF U>0 THEN D=8:GOTO 175 180
270 LOCATE 0,1:PRINT "VOUS","ETES","MO
RT";
275 IF INKEY$="" THEN 275 ELSE 10
```

```
— 299 REM
— 300 REM Sute!
305 A=INT(X/8):B=INT(Y/8)
- 310 LOCATE A,B:PRINT "^";
320 IF MID$(P$(B+1),A-3,1)="" THEN D=
8:Y=Y-8:GOTO 175 190
330 LOCATE 0,2:PRINT "PAF":BEEP 1
332 S=S-3:LOCATE 0,0:PRINT USING"###";
S
335 LOCATE 0,2:PRINT TAB(3):D=0:GOTO 1
25 190
350 REM Arrive en haut!
355 BEEP :BEEP :BEEP :BEEP 1
360 S=S+100
380 LOCATE 0,0:PRINT USING"###";S
385 GOTO 95
```

## **Mission suicide**

2384: Déjà vingt-quatre ans... vingt-quatre ans que les roboïdes nous ont déclaré la guerre. Le conflit atomique a pu être évité. Mais l'humanité encourt un danger imminent; nos services secrets savent que huit missiles nucléaires doivent bientôt décoller et provoquer LA CATASTROPHE, L'APOCALYPSE TERRIENNE...

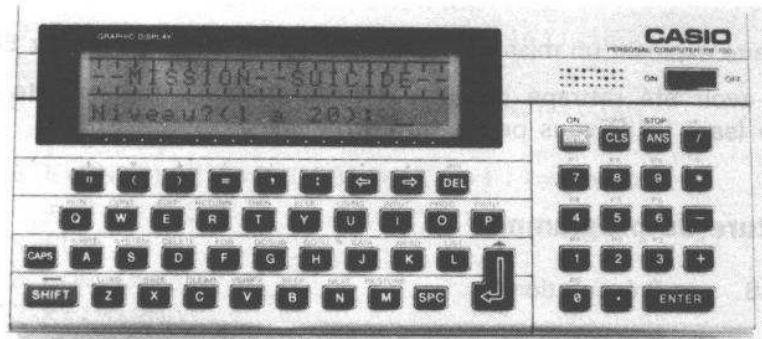
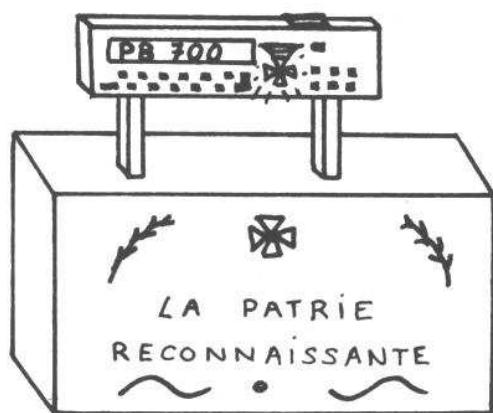
Vous avez été choisi, parmi les sept cents gardiens de la paix intergalactique pour sauver la civilisation.

Votre objectif: détruire les huits missiles avant leur envol.

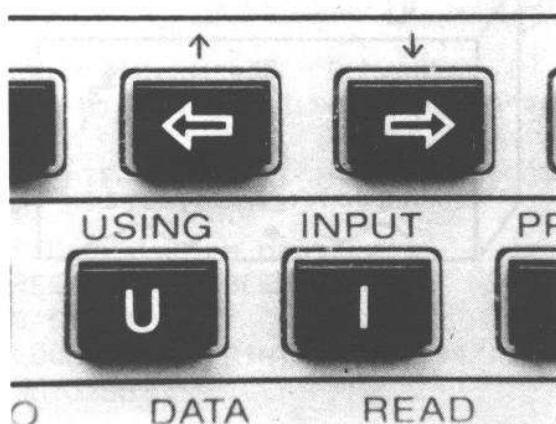
Vous avez été déposé dans leur base de départ mais attention: une patrouille ennemie vous a repéré.

A vous de mener à bien votre mission avant d'être pris...

Votre sacrifice ne sera pas oublié...



Pour vous déplacer, utilisez les touches  $\leftarrow$ ,  $\rightarrow$ ,  $\uparrow$  et  $\downarrow$ , ces deux dernières devant être shiftées.



Pour faire exploser un missile, placez-vous dessus et appuyez sur ENTER.

Après avoir été rattrapé par la patrouille, l'appui sur n'importe quelle touche (sauf BRK) vous permettra de rejouer.

### Structure du programme

- 1-8 : Présentation.
- 10-18 : Initialisations.
- 20-28 : Positionnement des objectifs (missiles).
- 30-50 : Scrutation du clavier.
- 50-54 : Tir !
- 55-58 : Affichage de l'espion.
- 60-72 : Mouvement de la patrouille.
- 80-85 : Test de rencontre espion-patrouille.
- 100-200 : Explosion finale.

---

## Variables utilisées

X,Y : Coordonnées DRAW de l'espion.  
E,F : Copie intermédiaire.  
T,U : Coordonnées DRAW de la tête de la patrouille.  
Q,R : Coordonnées DRAW de la queue de la patrouille.  
V,W : Coordonnées du vecteur espion → tête de patrouille puis nouvelle position de la tête de patrouille.  
D : Distance espion → tête de patrouille.  
B,C : Coordonnées PRINT de l'espion.

### M I S S I O N     S U I C I D E

```
1 REM
5 CLS
6 FOR I=1 TO 20:PRINT CHR$(145)::NEXT I
7 PRINT "--MISSION--SUICIDE--";
8 FOR I=1 TO 20:PRINT CHR$(144)::NEXT I
10 INPUT "Niveau?(1 à 20)": ",N:N=(3*N+35)/19
16 E=RND*160:F=RND*31:X=E:Y=F:T=158:U=15:Q=159:R=U
18 CLS
20 FOR I=1 TO 8
22 K=10*RND
24 FOR J=1 TO K:PRINT " ";:NEXT J
26 PRINT CHR$(244);
28 NEXT I
30 REM MUT. ESPION
40 A=ASC(INKEY$)
42 IF A=28 THEN X=E+5:GOTO 55
44 IF A=29 THEN X=E-5:GOTO 55
46 IF A=30 THEN Y=F-5:GOTO 55
48 IF A=31 THEN Y=F+5:GOTO 55
50 IF A><13 THEN GOTO 55
```

```

51 BEEP 1:B=E/8:C=F/8
52 IF B*(B-20)>=0 THEN 55
53 IF C*(C-4)>=0 THEN 55
54 LOCATE B,C:PRINT " ";
55 REM TRACE ESPION
56 DRAWC(E,F)-(E+1,F)-(E+1,F+1)-(E,F+
1):DRAW(X,Y)-(X+1,Y)-(X+1,Y+1)-(X,
Y+1)
58 E=X:F=Y
60 REM MUT. PATR.
65 U=E-T:W=F-U
66 D=ABSU+ABSW
68 U=T+N*U/D:W=U+N*W/D
70 DRAWC(T,U)-(Q,R):DRAW(U,W)-(T,U)
72 Q=T:R=U:T=U:U=W
80 REM TEST RENCONTRE
82 D=ABS(E-T)+ABS(F-U)
85 IF D>=2 THEN 30
100 REM PAF!!
120 B=INT(E/8):C=INT(F/8):IF B*(B-19)<
=0 THEN IF C*(C-3)<=0 THEN 150
130 B=9:C=2
150 BEEP
200 LOCATE B,C:PRINT CHR$(254); "BOUM!" ;
300 IF INKEY$><"" THEN 300
305 IF INKEY$="" THEN 305 ELSE 1

```

### ***Simulateur de vol spatial***

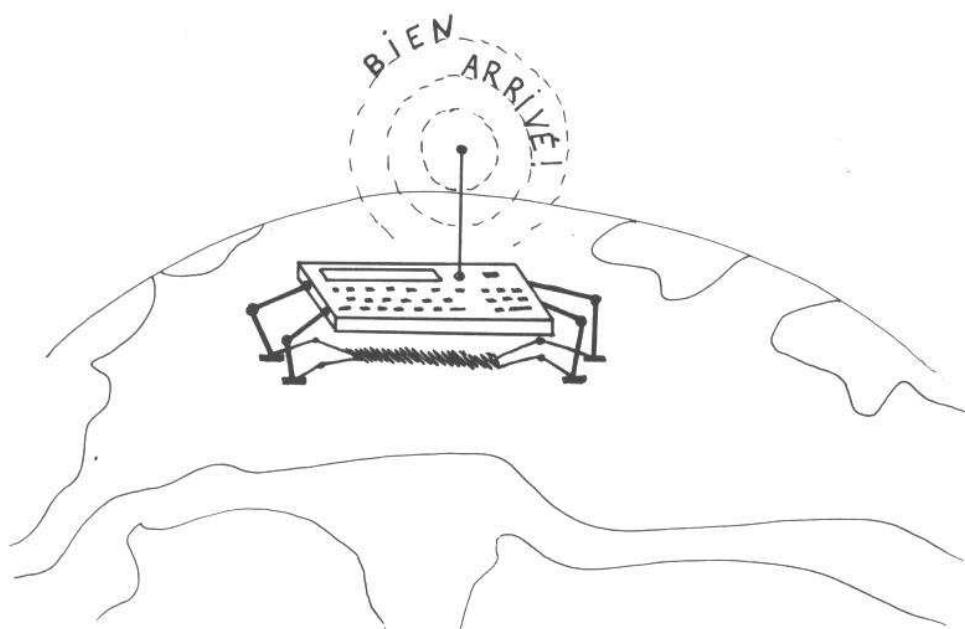
Nous avons transformé le PB 700 en fusée pour vous permettre de voyager dans l'espace, activité fort peu banalisée à l'heure actuelle.

Puisque la lune reste le siège de tant de rêves et de phantasmes c'est là-bas que vous devez atterrir.

Vous êtes seul aux commandes de votre appareil. Les paramètres permettant le bon alunissage sont la puissance des moteurs (P), la quantité de carburant (C) et bien sûr la vitesse (V). Cette dernière doit être inférieure à 3 m/s à l'alunissage. Sur l'écran est également repérée l'altitude de la fusée (ALT.).

La position de départ est générée aléatoirement par le micro-ordinateur.

Alors, enfilez vite votre costume de voyageur de l'espace et bon alunissage !



### Structure du programme

- 1-6 : Présentation.
- 10-19 : Dessin du décor (tableau de bord).
- 20-24 : Initialisations.
- 30-38 : Lecture du clavier.

- 50-58 : Calcul du mouvement et de la consommation.  
 60-72 : Affichage des paramètres du mouvement au tableau de bord.  
 80-92 : Arrivée au sol ?  
     Panne sèche ?  
     Vitesse exagérée ?  
     En cas de panne sèche, le GOTO 50 de la ligne 90 ignore le clavier.  
 100-120 : Crash ou atterrissage.

### Variables utilisées

Z,V : Altitude, vitesse.  
 C,P : Carburant, puissance moteurs: en ligne 57, l'accélération est prise égale à :

$$\frac{72 - P}{12} = 6 - \frac{P}{12}$$

pesanteur lunaire      composante due aux moteurs

L E M

```

1 REM----SIMULATEUR D"ALUNISSAGE
2 CLS :LOCATE 4,1:PRINT "^ L E M ^"
  "
4 FOR I=20 TO 120 STEP 20
6 DRAW(I,20)-(I+10,18)-(I+10,22)-(I,
  20)
8 NEXT I
9 FOR I=1 TO 500:NEXT I
10 REM----DECOR
12 CLS
  
```

```
14 PRINT "ALT: M U: M/S";CHR$(237);
16 PRINT "MOT: " ;CHR$(37);"\u00C:
           \u00U";CHR$(223);
18 LOCATE 19,2:PRINT CHR$(236);
19 DRAW(152,0)-(152,31)
20 REM----INITIALISATION
22 Z=RND*100:U=RND*5
24 C=2500:P=50
30 REM----SCRUTATION DU CLAVIER
32 IF INKEY$= "+" THEN P=P+1:GOTO 50
34 IF INKEY$= "-" THEN P=P-1:GOTO 50
36 IF INKEY$= "^" THEN P=P+10:GOTO 50
38 IF INKEY$= "!" THEN P=P-10
50 REM----CALCUL DU MOUVEMENT
53 IF P>100 THEN P=100
55 IF P<0 THEN P=0
57 Z=Z+U:U=U+6-P/12:IF Z>141 THEN Z=1
41
58 C=C-P*P/100:IF C<0 THEN C=0:P=0
60 REM----AFFICHAGE ETAT
62 LOCATE 4,0:PRINT USING"###";141-Z
64 LOCATE 11,0:PRINT USING"##.#";U
66 LOCATE 4,1:PRINT USING"###";P
68 LOCATE 13,1:PRINT USING"###";C
70 LOCATE 0,2:PRINT TAB(19)
75 DRAW(Z,20)-(Z+10,18)-(Z+10,22)-(Z,
20)
80 REM----TESTS
82 LOCATE 0,3:PRINT TAB(19);
84 IF Z=141 THEN 100
86 IF U>5 THEN IF Z>100 THEN LOCATE 0
,3:PRINT "WARNING: VITESSE":BEEP
1
88 IF C<500 THEN LOCATE 0,3:PRINT "WA
RNING: CARBURANT":BEEP 1
90 IF C=0 THEN 50
```

```
92 GOTO 30
100 REM-----ARRIVEE AU SOL
102 BEEP :BEEP :LOCATE 3,3
104 IF U>1 THEN PRINT "CRASH !"::LOCAT
E 18,2:PRINT CHR$(254)
106 IF U<1 THEN PRINT "BIEN ARRIVE !
";
120 IF INKEY$="" THEN 120 ELSE 10
```

## *Labyrinthe*



Vous ne vous êtes sûrement jamais demandé combien Thésée à peiné dans le Labyrinthe du roi MINOS pour échapper au Minotaure. Vous allez main-

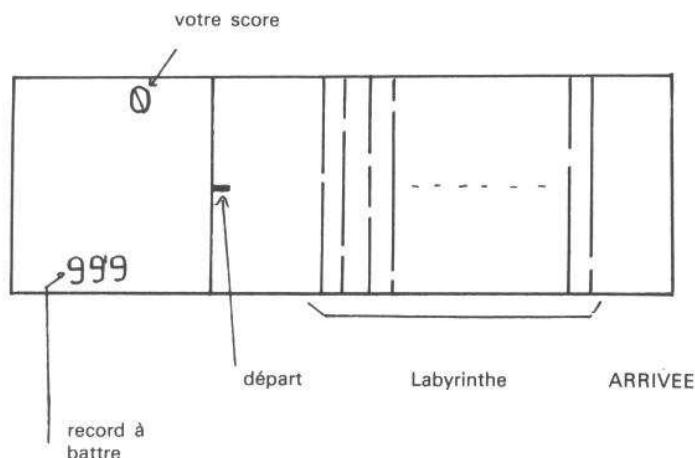
tenant pourvoir le faire, non pas avec un fil d'Ariane mais avec un PB 700 muni de deux touches de direction.

Surtout n'oubliez pas que vous êtes poursuivi et que vous devez coûte que coûte aller de l'avant.

Le but du jeu est de traverser le Labyrinthe le plus vite possible. Il faut donc prendre le chemin le plus court et éviter absolument de heurter un mur (ce qui vous gratifierai d'une pénalité de dix points).

## Description

1. Avant de commencer à jouer pour la première fois, il faut ABSOLUMENT initialiser le record (oublier provoquerait une UL\_ERROR). Pour cela faire PR=999 ↴
2. Lorsque vous lancerez le programme, votre écran se présentera de la manière suivante :



Pour commencer à avancer, appuyez sur n'importe quelle touche noire.  
Pour vous diriger, vous avez deux touches :

+ pour descendre

/ pour monter

Dès que vous lâchez l'une de ces touches, vous allez tout droit.

Arrivé au but, appuyez sur n'importe quelle touche noire pour connaître votre score.

### Description du programme

- 1-15 : Initialisation.
- 16-70 : Dessin aléatoire du Labyrinthe.
- 75-140 : Boucle de jeu.
- 150-350 : Résultats.

### Mémoires utilisées

- X,Y : Tracé du Labyrinthe puis position du point joueur.
- A\$ : Scrutation du clavier.
- C : Nombre de chocs sur les murs.
- D : Score du joueur.
- PR : Record (à initialiser à la première partie).

-----L A B Y R I N T H E-----

```
i CLS :C=0:D=0
2 LOCATE 5,1:PRINT "LABYRINTHE"
3 DRAW(0,5)-(159,5):DRAW(0,17)-(159,
17):DRAW(0,11)-(30,11):DRAW(130,11)-(159
,11)
8 FOR U=1 TO 200:NEXT U
9 CLS
15 DRAW(35,0)-(35,31)
16 DRAW(35,0)-(159,0)-(159,31)-(35,31
):DRAW(34,15)-(36,15)
20 FOR I=45 TO 145 STEP 5
30 X=INT(RND*29)+1:Y=INT(RND*29)+1
40 DRAW(I,0)-(I,31):DRAWC(I,X):DRAWC(
I,Y)
```

```
50 NEXT I
60 X=36:Y=15
70 DRAW(X,Y)
75 LOCATE 0,0:PRINT USING "###";D:LOCATE 0,3:PRINT USING "###";PR;
80 IF INKEY$="" THEN 80
85 DRAW(X,Y)
87 D=D+1:LOCATE 0,0:PRINT USING "###";
D
90 A$=INKEY$
100 IF A$="/" THEN Y=Y-1:GOTO 85
110 IF A$="/" THEN Y=Y+1:GOTO 85
120 X=X+1:IF X=159 THEN 200
122 IF X MOD 5<>0 THEN 85
125 IF X<15 THEN 85
130 IF POINT(X,Y)=0 THEN 85
140 BEEP :BEEP :DRAWC(X-1,Y)-(X-2,Y):D=D+10:DRAW(X-2,Y):X=X-2:C=C+1
150 IF INKEY$="" THEN 150 ELSE 85
200 IF INKEY$="" THEN 200
205 CLS :PRINT "ARRIUEE!","Vous avez heurte","";C;" mur";
206 IF C>1 THEN PRINT "s." ELSE PRINT "."
210 IF INKEY$="" THEN 210
300 CLS
310 IF DKPR THEN PR=D:PRINT "RECORD BATU!!","Distance Parcourue:",D;"m":GOTO 350
320 PRINT "Distance Parcourue:",D;"m":GOTO 350
350 IF INKEY$="" THEN 350 ELSE 1
```

# 5

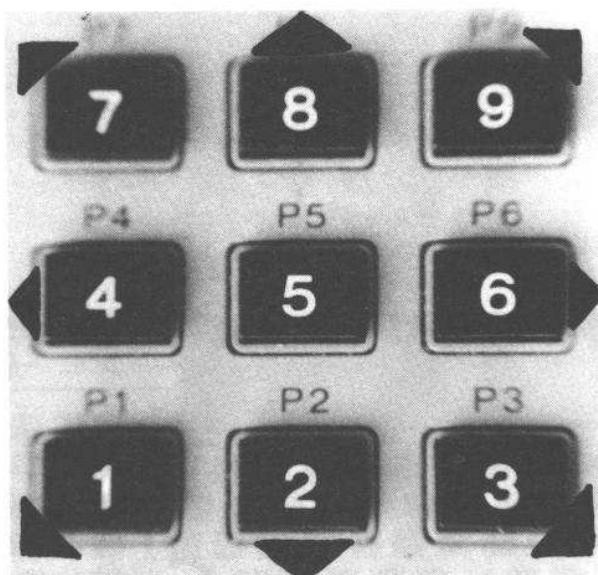
## Graphismes



## Dessin

Ce programme a pour but de vous permettre de développer les possibilités graphiques de l'écran du PB 700, sans vous enliser dans de fastidieux calculs de nombre de points...

Une fois le programme lancé par RUN, l'écran est effacé et un point clignote au centre. Vous avez alors le loisir de déplacer ce point suivant huit directions à l'aide du clavier numérique.



Ce point en se déplaçant laisse normalement une trace derrière lui. Vous pouvez changer cela en appuyant sur la touche ". ". Vous verrez alors apparaître un chiffre en haut à gauche de l'écran. Ce chiffre prend trois valeurs :

- Ø : le point se déplace sans laisser de trace.
- 1 : le point se déplace en laissant un trait noir derrière lui.
- 2 : le point se déplace en effaçant l'écran sur son passage.

## Structure du programme

- 1-25 : Initialisations.
- 50-70 : Différents aiguillages en fonction du clavier.
- 70-90 : Mise en place des paramètres du mouvement.
- 100-490 : Clignotement du dernier point tracé et retour à la scrutation du clavier.

Sous-programme en ligne 500:

- 510 : Le sous-programme ne s'achève que lorsque l'utilisateur relâche le clavier.
- 520-600 : Mouvement du point et retour en 500.

## Variables utilisées

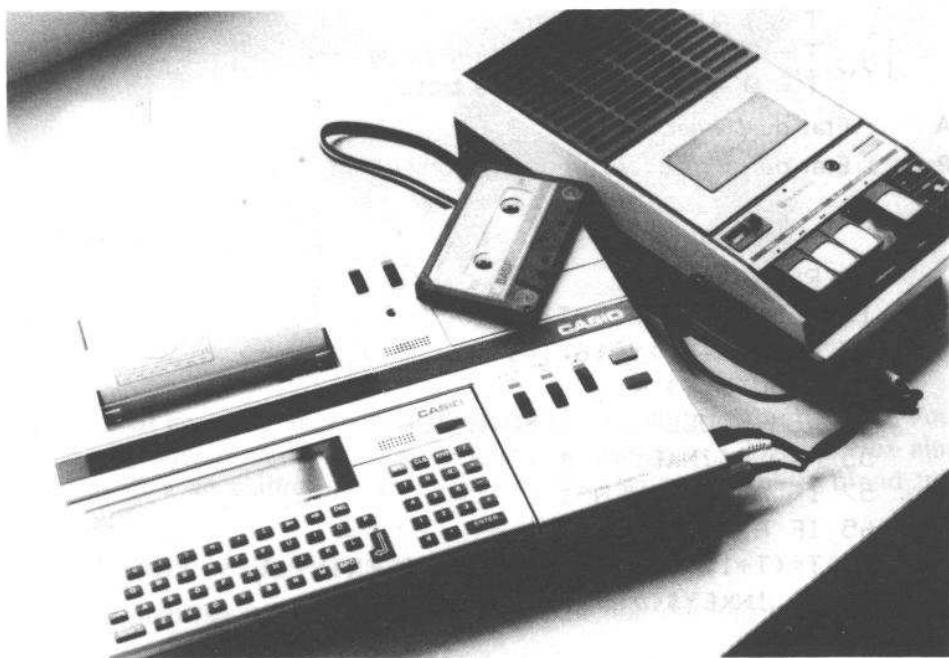
- X, Y : Coordonnées DRAW du curseur.
- D, E : Vitesse du curseur.
- T : Paramètre indiquant si le curseur doit:
  - T = 1 → Laisser une trace.
  - T = 2 → Effacer tout là où il passe.
  - T = Ø → Ne laisser aucune trace.
- A : État du clavier.
- P : État du point où se trouve le curseur.

### D E S S I N

```
1 REM
10 REM INITIALISATIONS
20 CLS :X=80:Y=16:T=1
25 DRAW(X,Y)
50 REM BOUCLE DE SCRUTATION
55 A=ASC(INKEY$)-48:D=0:E=0
57 IF A=-37 THEN 1
65 IF A<-2 THEN 70
67 T=(T+1) MOD 3:LOCATE 0,0:PRINT T
68 IF INKEY$<>"" THEN 68 ELSE 50
70 REM
71 IF (A-1)*(A-4)*(A-7)=0 THEN D=-1
```

```
75 IF (A-3)*(A-6)*(A-9)=0 THEN D=1
80 IF (A-1)*(A-2)*(A-3)=0 THEN E=1
85 IF (A-7)*(A-8)*(A-9)=0 THEN E=-1
90 IF A*(A-10)<0 THEN GOSUB 500
100 P=POINT(X,Y):DRAWC(X,Y):DRAW(X,Y)
110 IF P=0 THEN DRAWC(X,Y)
490 GOTO 50
499 REM
500 REM MOUVEMENT
501 REM
510 IF INKEY$="" THEN RETURN
520 X=X+D:Y=Y+E
530 P=POINT(X,Y):DRAW(X,Y)
540 IF T=2 THEN DRAWC(X,Y)
550 IF T=0 THEN IF P=0 THEN DRAWC(X,Y)
600 GOTO 500
```

### **Police de caractères**



Ce programme n'est pas un jeu à proprement parler. C'est une routine permettant d'écrire sur l'écran de votre PB 700 avec des caractères plus gros.

Deux tailles sont à votre disposition :

- *Taille 1*: On peut écrire deux lignes de dix caractères.
- *Taille 2*: On peut écrire une ligne de cinq caractères.

Au lancement, le programme vous demande la taille désirée.

### a) Si vous choisissez 1

- Vous verrez apparaître un ? à un endroit défini et vous pourrez écrire dix caractères au plus: c'est la première ligne.
- Puis vous faites ↵ et vous verrez apparaître un deuxième ? et vous pourrez encore écrire dix caractères: c'est la deuxième ligne.
- Vous faites alors ↵ et le programme est lancé.
- Si vous ne voulez rien écrire sur la première ligne, faites directement ↵ à la question correspondante.

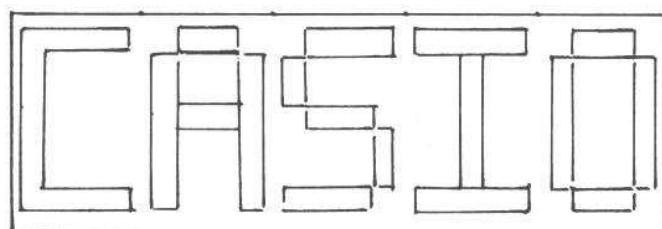
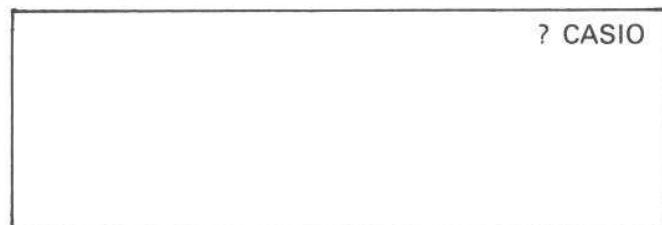
**NB:** Si vous voulez écrire en plus gros un caractère non accessible au clavier, suivez la méthode ci-après :

- Répondez par un espace (SPC) au point d'interrogation, le programme vous demande alors le code du caractère. Vous pouvez ainsi remplir toute une ligne de ce genre de caractères.
- Si vous ne désirez pas une ligne de dix caractères mais seulement de 3 ou 4, répondez Ø à la question "code (l) =" et le programme sera lancé.
- Dès que vous entendez la tonalité sonore, le programme est terminé. Appuyez alors sur n'importe quelle touche pour recommencer.

### b) Si vous choisissez 2

La méthode est la même que précédemment mais une ligne de cinq caractères remplit l'écran.

**Exemple :**



### Description du programme

- 10-50 : Initialisation.
- 55-260 : Question sur la taille des caractères (taille 1).
- 270-360 : Écriture de la ligne 1.
- 370-440 : Écriture de la ligne 2.
- 450-600 : Question sur la chaîne de caractères (taille 2).
- 610-710 : Écriture de la ligne.
- 1000-1010: Fin du programme.

```
P O L I C E S   D   E C R I T U R E
10 CLS
20 LOCATE 1,1:PRINT "POLICES D ECRITU
RE"
30 DRAW(0,18)-(160,18)-(160,4)-(0,4)
40 LOCATE 0,3:INPUT "Taille(1 ou 2):"
,R
```

```
50 IF R=2 THEN 500
55 REM lettres moyennes
60 CLS
70 LOCATE 9,0: INPUT AA$
80 IF AA$="" THEN 170
90 IF AA$>" " THEN 170
100 CLS :AA$=""
110 FOR I=1 TO 10
120 PRINT "Code(";I;")::";:INPUT W
130 IF W=0 THEN CLS :GOTO 170
140 AA$=AA$+CHR$(W)
150 NEXT I
160 CLS
170 LOCATE 9,2: INPUT BB$
180 IF BB$="" THEN 260
190 IF BB$>" " THEN 260
200 CLS :BB$=""
210 FOR I=1 TO 10
220 PRINT "Code(";I;")::";:INPUT W
230 IF W=0 THEN 260
240 BB$=BB$+CHR$(W)
250 NEXT I
260 CLS
265 REM écriture
270 LOCATE 10,0:PRINT AA$:LOCATE 10,2:
PRINT BB$
280 K=LEN(AA$):L=LEN(BB$)
290 IF K=0 THEN 370
300 FOR I=80 TO 79+8*K
310 FOR J=0 TO 7
320 IF POINT(I,J)=0 THEN 350
330 X=2*I-160:Y=2*j
340 DRAW(X,Y)-(X+1,Y)-(X+1,Y+1)-(X,Y+1)
>:DRAWC(I,J)
350 NEXT J
360 NEXT I
370 IF BB$="" THEN GOTO 1000
```

```
380 FOR I=80 TO 79+8*L
390 FOR J=16 TO 23
400 IF POINT(I,J)=0 THEN 430
410 X=2*I-160:Y=2*J-16
420 DRAW(X,Y)-(X+1,Y)-(X+1,Y+1)-(X,Y+1)
):DRAWC(I,J)
430 NEXT J
440 NEXT I
450 GOTO 1000
495 REM grandes lettres
500 CLS
510 LOCATE 14,0:INPUT A$
520 IF A$="" THEN 1000
530 IF A$>"" THEN LOCATE 14,0:PRINT
":GOTO 610
540 CLS :A$=""
550 FOR U=1 TO 5
560 PRINT "Code(";U;")";::INPUT W
570 IF W=0 THEN 600
580 A$=A$+CHR$(W)
590 NEXT U
600 CLS :LOCATE 15,0:PRINT A$
605 REM ecriture
610 K=LEN(A$)
620 FOR I=120 TO 119+8*K
630 FOR J=0 TO 7
640 IF POINT(I,J)=0 THEN 700
650 DRAWC(I,J)
660 X=4*I-480:Y=4*J
670 FOR T=0 TO 3
680 DRAW(X,Y+T)-(X+3,Y+T)
690 NEXT T
700 NEXT J
710 NEXT I
1000 BEEP
1010 IF INKEY$="" THEN 1010 ELSE 10
```

**Imprimerie de la Manutention à Mayenne**

Dépôt légal: janvier 1985

Nº d'Éditeur: 4198

Avec le PB 700 est née une nouvelle ère de l'informatique sur micro ordinateurs. En effet, vous avez maintenant un véritable petit écran graphique «dans la poche» et cela va vous permettre de découvrir tout au long de ce livre, des jeux beaucoup plus passionnantes et interactifs que ceux que l'on pouvait programmer auparavant sur une machine de poche.

Le temps où un papier et un crayon étaient nécessaires pour jouer avec votre Pocket-computer est maintenant révolu. Que vous soyez joueur d'Othello ou fanatique de Bataille navale, que vous aimiez l'argent gagné facilement au poker ou aux dés, que vous ayez une âme de destructeur de briques, de cosmonaute ou même de pilote de course, ouvrez vite ce livre et tout en vous divertissant, découvrez facilement le basic en explorant les vingt programmes qui vous sont commentés de telle manière que l'informatique elle-même deviendra un jeu.